MX180000A
Signal Quality Analyzer
Control Software
Operation Manual
Remote Control

28th Edition

- For safety and warning information, please read this manual before attempting to use the equipment.
- Additional safety and warning information is provided within the MP180A Signal Quality Analyzer Installation Guide, the MT1810A 4 Slot Chassis Installation Guide, and MX180000A Signal Quality Analyzer Control Software Operation Manual. Please also refer to them before using the equipment.
- Keep this manual with the equipment.

ANRITSU CORPORATION
Safety Symbols

To prevent the risk of personal injury or loss related to equipment malfunction, Anritsu Corporation uses the following safety symbols to indicate safety-related information. Ensure that you clearly understand the meanings of the symbols BEFORE using the equipment. Some or all of the following symbols may be used on all Anritsu equipment. In addition, there may be other labels attached to products that are not shown in the diagrams in this manual.

Symbols used in manual

⚠️ **DANGER**
This indicates a very dangerous procedure that could result in serious injury or death if not performed properly.

⚠️ **WARNING**
This indicates a hazardous procedure that could result in serious injury or death if not performed properly.

⚠️ **CAUTION**
This indicates a hazardous procedure or danger that could result in light-to-severe injury, or loss related to equipment malfunction, if proper precautions are not taken.

Safety Symbols Used on Equipment and in Manual

The following safety symbols are used inside or on the equipment near operation locations to provide information about safety items and operation precautions. Ensure that you clearly understand the meanings of the symbols and take the necessary precautions BEFORE using the equipment.

- This indicates a prohibited operation. The prohibited operation is indicated symbolically in or near the barred circle.
- This indicates an obligatory safety precaution. The obligatory operation is indicated symbolically in or near the circle.
- This indicates a warning or caution. The contents are indicated symbolically in or near the triangle.
- This indicates a note. The contents are described in the box.

These indicate that the marked part should be recycled.

MX180000A
Signal Quality Analyzer Control Software
Operation Manual Remote Control

27 November 2006 (First Edition)
6 April 2018 (28th Edition)

Copyright © 2006-2018, ANRITSU CORPORATION.
All rights reserved. No part of this manual may be reproduced without the prior written permission of the publisher.
The contents of this manual may be changed without prior notice.
Printed in Japan
Notes On Export Management

This product and its manuals may require an Export License/Approval by the Government of the product’s country of origin for re-export from your country.

Before re-exporting the product or manuals, please contact us to confirm whether they are export-controlled items or not.

When you dispose of export-controlled items, the products/manuals need to be broken/shredded so as not to be unlawfully used for military purpose.
About This Manual

A testing system combining an MP1800A Signal Quality Analyzer or MT1810A 4-Slot Chassis mainframe, module(s), and control software is called a Signal Quality Analyzer Series. The operation manuals of the Signal Quality Analyzer Series consist of separate documents for the installation guide, the mainframe, remote control operation, module(s), and control software, as shown below.

- **Installation Guide**
  Installation guide from module installation to the start of use. The Installation Guide varies depending on the mainframe used.

- **Mainframe Operation Manual**
  Describes basic operations of the mainframe. The Mainframe Operation Manual varies depending on the mainframe used.

- **Remote Control Operation Manual**
  Describes remote control using the GPIB interface and Ethernet interface.

- **Module Operation Manual**
  Operation manual for the module. The Module Operation Manual varies depending on the module(s) used.

- **Control Software Operation Manual**
  Operation manual of the software that controls the Signal Quality Analyzer Series.

---

Configuration of Signal Quality Analyzer Series Operation Manual

- indicates this document.
# Table of Contents

About This Manual.................................................. I

Chapter 1 Overview .............................................. 1-1
  1.1 Overview ..................................................... 1-2

Chapter 2 Connections ........................................... 2-1
  2.1 Connections .................................................. 2-2
  2.2 GPIB Interface ............................................... 2-5
  2.3 Ethernet Interface ........................................... 2-11

Chapter 3 SCPI Format............................................. 3-1
  3.1 SCPI Listener Input Format ................................. 3-2
  3.2 SCPI Talker Output Format ................................. 3-23
  3.3 Command Configuration .................................... 3-31
  3.4 Command Syntax ............................................ 3-32
  3.5 Command Combinations .................................. 3-33
  3.6 Parameters .................................................. 3-34
  3.7 Responses .................................................. 3-35

Chapter 4 Native Format ......................................... 4-1
  4.1 Native Listener Input Format ............................... 4-2
  4.2 Native Talker Output Format ............................... 4-6
  4.3 Command Combinations .................................. 4-10
  4.4 Parameters .................................................. 4-10
  4.5 Responses .................................................. 4-11

Chapter 5 IEEE488.2 Common Commands..... 5-1
  5.1 IEEE488.2 Common Commands ............................... 5-2
Chapter 6 Status Report......................................... 6-1
  6.1 Overview....................................................... 6-2
  6.2 IEEE488.2 Regulated Register............................ 6-3
  6.3 SCPI-Regulated Status Register.......................... 6-6
  6.4 How to Read and Write Status Register.................. 6-8
  6.5 How to Clear and Reset Status Register............... 6-9
  6.6 Device-Unique Status...................................... 6-10
  6.7 Status Commands........................................... 6-13

Chapter 7 SCPI Commands ................................. 7-1
  7.1 Common Commands........................................... 7-5
  7.2 Synthesizer Commands..................................... 7-61
  7.3 PPG Commands............................................. 7-78
  7.4 ED Commands.............................................. 7-209
  7.5 Optical Module Commands.................................. 7-359
  7.6 Automatic Measurement Commands...................... 7-368
  7.7 Stressed Eye Transmitter Commands..................... 7-490
  7.8 MUX Commands............................................ 7-504
  7.9 DEMUX Commands.......................................... 7-542
  7.10 Jitter Commands.......................................... 7-564
  7.11 28G/32G bit/s PPG Commands............................ 7-597
  7.12 28G/32G bit/s ED Commands.............................. 7-714

Chapter 8 Native Commands.............................. 8-1
  8.1 Common Commands.......................................... 8-2
  8.2 Synthesizer Commands..................................... 8-31
  8.3 PPG Commands............................................. 8-49
  8.4 ED Commands.............................................. 8-174
  8.5 Automatic Measurement Commands...................... 8-338

Chapter 9 Basic Operation of Device Messages9-1
  9.1 Basic Operation of Device Messages.................... 9-2
  9.2 Error Messages........................................... 9-4
Appendix A Command Compatibility List ...... A-1
Chapter 1 Overview

This chapter provides an overview of remote control by the MX180000A Signal Quality Analyzer Control Software (hereinafter, referred to as “MX180000A”).

1.1 Overview................................................................. 1-2
1.1 Overview

The MP1800A Signal Quality Analyzer (hereinafter, referred to as “MP1800A”) and the MT1810A 4-Slot Chassis (hereinafter, referred to as “MT1810A”) that are controlled by the MX180000A are capable of performing automatic measurements when connected to an external controller. Either the GPIB or Ethernet interface may be used for connection. In addition, the SCPI standard, which is becoming the global standard, is used for the remote control commands. Refer to SCPI 1999.0 (SCPI Consortium) for details on SCPI.

The general conditions for using the SCPI standard are described below:

- Commands for GPIB and Ethernet are standardized.
- These commands are not applied for functions executed within the Setup Utility (setting of GPIB/Ethernet, self-test, etc.).
- The above conditions can all be read by the command.

The GPIB/Ethernet commands of the MP1800A/MT1810A can also use some of the command definitions (Native) that are used with Anritsu’s Pulse Pattern Generators and/or Error Detectors (BERTS), as well as SCPI. This means that these commands can be executed with a single character string, differing from the tree structure of the SCPI commands.
Chapter 2 Connections

This chapter describes the connections of the equipment for using the remote control function and the settings for using the GPIB and Ethernet interfaces, which are MP1800A options.

2.1 Connections ................................................................. 2-2
  2.1.1 Connecting MP1800A........................................... 2-2
  2.1.2 Connecting MT1810A.......................................... 2-3
2.2 GPIB Interface ............................................................. 2-5
  2.2.1 GPIB interface function ..................................... 2-5
  2.2.2 Device message list.......................................... 2-6
  2.2.3 Bus commands................................................. 2-7
  2.2.4 Connecting GPIB cable .................................... 2-7
  2.2.5 Setting GPIB.................................................... 2-8
  2.2.6 System initialization....................................... 2-9
2.3 Ethernet Interface ....................................................... 2-11
  2.3.1 Device message list........................................ 2-11
  2.3.2 Setting Ethernet board.................................... 2-13
  2.3.3 Network connection and data flow ................. 2-15
2.1 Connections

This chapter describes the connections for using the remote control function for the MP1800A, which does not require a control PC for basic operations, and the MT1810A, which requires a control PC for basic operations.

2.1.1 Connecting MP1800A

The remote control function of the MP1800A is implemented by remotely controlling the MP1800A from a PC for remote control (remote control PC).

Use the GPIB or the 10/100 M Ethernet interface to connect the MP1800A and a remote control PC. An example is shown below.

Notes:

- Be sure to use the 100M full-duplex mode for the Ethernet connection.

- Direct connection without using hub is recommended for the Ethernet connection. Use a crossover cable for direct connection.

![Configuration for remote control of MP1800A from the remote control PC](image-url)
2.1.2 Connecting MT1810A

Two types of configurations are available for using the remote control function of the MT1810A: remote control of the MX180000A from the remote control PC, and remote control of the MX180000A within the same PC.

[1] Remote control of the MX180000A from the remote control PC
Remote control of the MX180000A can be achieved by using an Ethernet connection between the control PC and the remote control PC.

Use an Ethernet to connect the MT1810A and control PC, and use the GPIB or 10/100 M Ethernet interface to connect the remote control PC and the control PC. An example is shown below.

**Notes:**
- Use the 100M full duplex mode for the Ethernet connection.
- Direct connection without using any hub is recommended for the Ethernet connection. Use a crossover cable for direct connection.

![Figure 2.1.2-1 Configuration for remote control of the MX180000A from the remote control PC](image-url)
Remote control of the MX180000A within the same PC

Remote control of the MX180000A can be done within the control PC. An example is shown below.

**Note:**

For remote control within the same PC, set the destination IP address of the remote control application to 127.0.0.1 (PC loopback address).

---

**Figure 2.1.2-2** Configuration for remote control of MX180000A within the same PC
2.2 GPIB Interface

This chapter describes the GPIB interface functions and settings for using the GPIB interface of the MP1800A-001.

2.2.1 GPIB interface function

The MP1800A has device functions but no controller function. Therefore, its interface functions are as shown in the following table according to the IEEE488.2 standard.

<table>
<thead>
<tr>
<th>Code</th>
<th>Interface Function</th>
<th>IEEE488.2 Standard</th>
</tr>
</thead>
<tbody>
<tr>
<td>SH1</td>
<td>All functions for source handshake</td>
<td>All functions are standardly equipped</td>
</tr>
<tr>
<td>AH1</td>
<td>All functions for acceptor handshake</td>
<td>All functions are standardly equipped</td>
</tr>
<tr>
<td>T5</td>
<td>Basic talker functions</td>
<td>The device must have a subset T5, T6, TE5, or TE6.</td>
</tr>
<tr>
<td>L4</td>
<td>Basic listener functions</td>
<td>The device must have a subset L3, L4, LE3, or LE4.</td>
</tr>
<tr>
<td>SR1</td>
<td>All functions for service request</td>
<td>All functions are standardly equipped</td>
</tr>
<tr>
<td>RL1</td>
<td>All functions for remote/local</td>
<td>All remote/local functions of RL0 (no function) or RL1 (all functions)</td>
</tr>
<tr>
<td>PP0</td>
<td>No parallel poll function</td>
<td>PP0 (no function) or PP1 (all functions)</td>
</tr>
<tr>
<td>DC1</td>
<td>All functions for device clear</td>
<td>All functions are standardly equipped</td>
</tr>
<tr>
<td>DT1</td>
<td>All functions for device trigger</td>
<td>DT1 (all functions)</td>
</tr>
<tr>
<td>C1*1</td>
<td>Controller functions except parallel poll</td>
<td>C0 (no function), C4 and C5, or any of C7, C9, or C11</td>
</tr>
<tr>
<td>C2*2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C3*3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C4*4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C7*5</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*1: System controller
*2: IFC transmission, controller in charge
*3: REN transmission
*4: Response to SRQ
*5: Interface message transmission, reception and passing of the control, and passing of the control to itself
2.2.2 Device message list

Device messages are data messages that are transmitted and received between the remote control PC and the MP1800A via the system interface when the bus mode is the data mode (when the ATN line is "H"). Device messages consist of program messages and response messages.

Program messages are ASCII data messages transferred from the controller to the device. Response messages are data messages transferred from the device to the controller.

Program messages and response messages have the following types.

<table>
<thead>
<tr>
<th>Table 2.2.2-1  Device message</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Program Message</strong></td>
</tr>
<tr>
<td>(See Section 3.1.2)</td>
</tr>
</tbody>
</table>

- Program instruction
  - Device-unique command (See Chapters 7 and 8.)
  - IEEE488.2 common command (See Chapter 5.)
- Program query
  - Status message (See Chapter 6.)
  - Response message

The messages are exchanged via the I/O buffer of the device. The table below briefly describes the I/O buffer.

<table>
<thead>
<tr>
<th>Table 2.2.2-2  I/O buffer</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Input Buffer</strong></td>
</tr>
<tr>
<td>FIFO (First in First out) type memory area that temporarily stores DABs (program messages and query messages) before syntax analysis. The input buffer size of the MP1800A is 1 Kbytes.</td>
</tr>
</tbody>
</table>
2.2 GPIB Interface

2.2.3 Bus commands

Bus commands are used for internal communications of the interface transmitted while the bus mode is the command mode (when the ATN line is “L”).

The table below lists the bus commands.

<table>
<thead>
<tr>
<th>Bus Command</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>DCL (Device Clear)</td>
<td>Initializes message exchange of all devices connected to the GPIB bus.</td>
</tr>
<tr>
<td>SDC (Selected Device Clear)</td>
<td>Initializes message exchange of the addressed MP1800A. The operation is the same as the DCL.</td>
</tr>
<tr>
<td>IFC (Interface Clear)</td>
<td>Initializes the interface.</td>
</tr>
</tbody>
</table>

2.2.4 Connecting GPIB cable

Connect the GPIB cable to the GPIB connector on rear back panel of the MP1800A.

Systems using GPIB have the following restrictions:

- Number of connectable devices \( \leq 15 \) units
- Total cable length \( \leq 2 \times \text{Number of devices} \) (20 m, max.)
2.2.5 Setting GPIB

To use the GPIB as a remote interface, set the device to Local, and use the keys to perform the following settings on the Remote Control tab of the Setup Utility main window.

<table>
<thead>
<tr>
<th>Setting Detail</th>
<th>Setting Item</th>
<th>Setting Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Selection of the interface used for remote control</td>
<td>Active Interface</td>
<td>GPIB: Uses the GPIB interface for remote control.</td>
</tr>
<tr>
<td>Address setting</td>
<td>Address</td>
<td>0 to 30</td>
</tr>
<tr>
<td>Command processing mode</td>
<td>Performance</td>
<td>Normal/Enhanced</td>
</tr>
</tbody>
</table>

**Setting procedure:**
1. Start the Setup Utility, and click the Remote Control tab.
2. Select the GPIB in the Active Interface field (click a radio button).
3. Enter the GPIB address into the text box, and then click the **Apply** button.

See the *MP1800A Signal Quality Analyzer Installation Guide* for setting details.
2.2 GPIB Interface

2.2.6 System initialization

IEEE488.2 defines system initialization in three levels: bus initialization, message initialization, and device initialization.

Table 2.2.6-1  Device initialization

<table>
<thead>
<tr>
<th>Level</th>
<th>Initialization Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Bus initialization</td>
<td>Initializes all interface functions connected to the bus with IFC message from the controller.</td>
</tr>
<tr>
<td>2</td>
<td>Message initialization</td>
<td>Disables function to report completion of initialization of message exchange or operation of all devices on the GPIB with the GPIB bus command DCL or the device specified with the bus command SDC, to the controller.</td>
</tr>
<tr>
<td>3</td>
<td>Device initialization</td>
<td>Recovers the known state that is specific to the device with the *RST command regardless of the past use conditions.</td>
</tr>
</tbody>
</table>

2.2.6.1 Bus initialization

IFC  Initializes the bus with IFC statement.

Function  Activates the IFC line for about 100 μs to initialize interface functions of all devices connected to the GPIB bus line. Only the system controller can transmit IFC.

2.2.6.2 Message initialization

DCL, SDC  Message exchange initialization using the DCL/SDC bus command

DCL: Initializes message exchange for all devices on the GPIB.

SDC: Initializes message exchange for the specified device.

Function  Initializes message exchange for all devices on the GPIB or only the specified device. Initialize message exchange when change of the panel setting state is not required, but if the parts related to message exchange inside the device are in a state that is not suitable for control from the controller due to execution of other programs. When message exchange is initialized, new instructions can be transmitted from the controller.
2.2.6.3 Device initialization

*RST

Initializes the device with the *RST command.

Function

Resets the device-unique function to a known state, regardless of the past use history. For the MP1800A, the factory-shipped settings are restored.

2.2.6.4 Device state upon application startup

When the application of the MP1800A starts up, it enters the following state:

- The state in which the application was terminated is set.
- The input buffer and output queue are cleared.
- The syntax analyzer, execution controller, and response creator are reset.
- Messages can be transmitted and received only when the application is running.
2.3 Ethernet Interface

This chapter describes the settings for using the Ethernet interface of the MT1810A, MP1800A-002.

2.3.1 Device message list

Device messages are data messages that are transmitted and received between the controller and device via the system interface. Device messages consist of program messages and response messages.

Program messages are ASCII data messages transferred from the controller to the device. Response messages are data messages transferred from the device to the controller.

Program messages and response messages have the following types.

<table>
<thead>
<tr>
<th>Program message (See Section 3.1.2)</th>
<th>Response message (See Section 3.2.2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Program instruction</td>
<td>Program query</td>
</tr>
<tr>
<td>• Device-unique command (See Chapters 7 through 8.)</td>
<td>• Status message (See Chapter 6.)</td>
</tr>
<tr>
<td>• IEEE488.2 common command (See Chapter 5.)</td>
<td>• Response message</td>
</tr>
</tbody>
</table>

Figure 2.3.1-1 Device messages
The messages are exchanged via the I/O buffer of the device. The table below briefly describes the I/O buffer.

<table>
<thead>
<tr>
<th>Input Buffer</th>
<th>Output Queue</th>
</tr>
</thead>
<tbody>
<tr>
<td>FIFO (First in First out) type memory area that temporarily stores DABs (program messages and query messages) before syntax analysis. The input buffer size of the MP1800A is 1 Kbytes.</td>
<td>FIFO type queue memory area. All the DABs (response messages) outputted from the device to the controller are stored in this memory until the controller finishes reading them.</td>
</tr>
</tbody>
</table>

The MP1800A or MT1810A main unit must be connected to the TCP/IP network, for the Ethernet interface to be used. In this instance, the MP1800A or MT1810A works as the server.
2.3 Ethernet Interface

2.3.2 Setting Ethernet board

To use the Ethernet as a remote interface, set the device to Local, and use the keys to perform the following settings on the Remote Control tab of the Setup Utility main window.

<table>
<thead>
<tr>
<th>Window</th>
<th>Setting Detail</th>
<th>Setting Item</th>
<th>Setting Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Remote Control tab</td>
<td>Selection of the interface used for remote control</td>
<td>Active Interface</td>
<td>Ethernet: Uses the Ethernet interface for remote control.</td>
</tr>
<tr>
<td></td>
<td>Port number setting</td>
<td>Port Number</td>
<td>1024 to 5001</td>
</tr>
<tr>
<td>Network setting,</td>
<td>IP address</td>
<td>IP Address</td>
<td>0.0.0.0 to 255.255.255</td>
</tr>
<tr>
<td>property setting</td>
<td>Subnet mask</td>
<td>Sub Net Mask</td>
<td>0.0.0.0 to 255.255.255</td>
</tr>
<tr>
<td></td>
<td>Gateway</td>
<td>GateWay</td>
<td>0.0.0.0 to 255.255.255</td>
</tr>
</tbody>
</table>

Setting procedure:
1. Start the Setup Utility, and click the Remote Control tab.
2. Select the Ethernet in the Active Interface field (click a radio button).
3. Enter the IP Address, Subnet Mask, Gateway, and Port Number, and then click the Apply button.

![Setup Utility: Remote Control tab – Ethernet setting](image)

Notes:
- For the IP address of the remote interface, set the address other than “192.168.1.xxx”. The “192.168.1.xxx” address is used for the module address. If this address is set, the module may not operate properly.
For the network settings to be used for the remote interface, use “Setup Utility” of the MX180000A. If the IP address is set from the control panel of the operating system (Windows), the module may not operate properly.

IP address
In a network using TCP/IP, devices connected to the network are identified by IP addresses. An IP address must therefore be assigned to each device. An IP address is a 32-bit number, and expressed as four 8-bit portions separated by dots (called dot notation).
IP addresses include network information in addition to the device (host) information. The data lengths of the network part and host part of an IP address is defined depending on the network class. Class C has 24-bit network part and 8-bit host part, and up to 254 hosts can be connected. Classes A through E are available; however, only Classes A through C are normally used.

<table>
<thead>
<tr>
<th>Class</th>
<th>Network part length</th>
<th>Host part length</th>
<th>Number of hosts that can be assigned</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>8 bits</td>
<td>24 bits</td>
<td>16,777,214 units</td>
</tr>
<tr>
<td>B</td>
<td>16 bits</td>
<td>16 bits</td>
<td>65,534 units</td>
</tr>
<tr>
<td>C</td>
<td>24 bits</td>
<td>8 bits</td>
<td>254 units</td>
</tr>
</tbody>
</table>

Subnet mask
The subnet mask is used to indicate the network part in the IP address when the network is divided into subnets. The network part of the IP address above (including the extended subnet part) is indicated by “1”, and the host part is defined by “0”. If this setting is wrong, IP packets cannot be transmitted or received correctly to or from the connected network that uses subnets.

Gateway
A device called a gateway is used to connect networks. Gateways include dedicated devices such as routers. In a TCP/IP network, IP packets can be directly exchanged within the same network. To exchange IP packets among different networks (i.e., terminals that have IP addresses with different network parts), however, communication with a device connecting to other network connected to the gateway via the gateway is required.
2.3.3 Network connection and data flow

Connect the Ethernet cable to the MP1800A/MT1810A main unit to connect to the network. The communication with MP1800A/MT1810A is data communication via the TCP connection. For communication, creating a communication program (socket client) at the remote control PC side is required.

For the socket interface used for communication, see the operation manuals of the remote control PC, the network interface board installed, and the driver software.

Data communication
The data from the client is saved in the reception buffer. Flow control of TCP occurs when the internal buffer is full, and no command-level response may return to the client. Some applications may be abnormally terminated due to timeout. In this case, the connection with the client is not disconnected. In some cases, you need to protect application operations by re-transmission.

Figure 2.3.3-1 Data flow
This chapter describes the SCPI command system.

3.1 SCPI Listener Input Format ........................................ 3-2
   3.1.1 SCPI listener input program message format .. 3-2
   3.1.2 Functional elements of program messages ..... 3-5
   3.1.3 Program data format ....................................... 3-17
3.2 SCPI Talker Output Format ....................................... 3-23
   3.2.1 SCPI talker output response message format ........... 3-23
   3.2.2 Functional elements of response messages .. 3-24
3.3 Command Configuration ............................................ 3-31
3.4 Command Syntax ..................................................... 3-32
3.5 Command Combinations ........................................... 3-33
3.6 Parameters ............................................................ 3-34
3.7 Responses ............................................................. 3-35
3.1 SCPI Listener Input Format

This chapter describes the format of program messages received by the listener (MP1800A/MT1810A) from the talker (remote control PC).

The device-unique commands of the MP1800A/MT1810A comply with the SCPI, so the SCPI commands are used in the examples in this chapter.

3.1.1 SCPI listener input program message format

The following figure shows a sample program message, which sets the data output to ON and sets the test pattern to PRBS.
3.1 *SCPI Listener Input Format*

The program message format consists of a sequence of functional elements, which are the minimum level units to indicate a function. In the figure above, the words written in uppercase alphabetical characters enclosed within brackets (< >) indicate examples of functional elements. A functional element is further divided into coding elements. In the figure above, the words written in lowercase alphabetical characters enclosed within brackets indicate examples of coding elements.

The WRITE and READ commands have the following formats.

**WRITE @**

Outputs data to the MP1800A/MT1810A.

* Format

\[
\text{WRITE } \ @\text{Device number: Data}
\]

- Data → Mathematical expression
- Data → String expression
Example: The same as the previous page

WRITE @
Outputs the data to the device.

★ Format

WRITE @ Device number: Data

Data → Mathematical expression
String expression

Example: The same as the example on the previous page

WRITE @03: " : OUTP : DATA : OUTP ON ; : SOUR : PATT : TYPE PRBS"

Listener address (when the GPIB address of MP1800A/MT1810A is 3)

READ @
Substitutes the data inputted from the device into the variable.

★ Format

READ @ Device number: variable

The data input from the device (input connector setting) to the variable A$.

WRITE @03: " : OUTP : DATA : OUTP?" Queries the input connector setting.

READ @03: A$

Listener address (when the GPIB address of MP1800A/MT1810A is 3)
3.1.2 Functional elements of program messages

The MP1800A/MT1810A receives a program message by detecting the terminator at the end of the program message. The functional elements of program messages are described below.

3.1.2.1 <TERMINATED PROGRAM MESSAGE>

The following figure shows a <TERMINATED PROGRAM MESSAGE> that transmits two instructions.

![Diagram of program message structure](image)

**Figure 3.1.2.1-1  <TERMINATED PROGRAM MESSAGE>

Definition

<TERMINATED PROGRAM MESSAGE> is defined as follows:

- **<PROGRAM MESSAGE>**: (See Section 3.1.2.6.)
- **<PROGRAM MESSAGE TERMINATOR>**: (See Section 3.1.2.2.)

<TERMINATED PROGRAM MESSAGE> is a data message containing all the functional elements required for transmitting the message from the remote control PC to the MP1800A/MT1810A. A <PROGRAM MESSAGE TERMINATOR> is added to the end of a <PROGRAM MESSAGE> to complete transmission of the <PROGRAM MESSAGE>. 
3.1.2.2 **<PROGRAM MESSAGE TERMINATOR>**

*<PROGRAM MESSAGE TERMINATOR>* is placed to terminate a sequence of one or more *<PROGRAM MESSAGE UNIT>* elements. The definition of the *<PROGRAM MESSAGE TERMINATOR>* differs according to the used interface.

(1) For GPIB interface:

(2) For Ethernet interface:

NL Defined as a single ASCII code byte 0A (decimal number 10), i.e. the ASCII control character LF (Line Feed) that performs the carriage return operation bringing the print position to the same character position in the next line. It is also called NL (New Line) because it starts from a new line.

For line feed, the CR+LF code may be used instead of the LF code.

If the remote control PC runs on MS-DOS or Windows, line feed is done with “CR+LF”, while it is done with only “LF” for UNIX.

END Generates an EOI signal by setting the EOI (End-or-Identify) line of the GPIB control bus to TRUE (low level).
3.1.2.3 \textless\text{white space}\textgreater

\textless\text{white space}\textgreater is defined as follows:

\begin{itemize}
  \item \textless\text{White space Character}\textgreater is defined as a single ASCII code byte within the range of the ASCII code bytes 00 to 09 and 0B to 20 (decimal numbers 0 to 9 and 11 to 32). The range includes the ASCII control symbols and space signals except New Line. The MP1800A/MT1810A processes these ASCII symbols simply as spaces or just ignores them, instead of interpreting them as ASCII control symbols.
\end{itemize}
3.1.2.4  <PROGRAM MESSAGE>

The following figure shows a setting example, which activates the alarm when an error occurs and sets the test pattern to PRBS.

![Diagram showing setting example for activating alarm and setting test pattern to PRBS]

**Definition**

<PROGRAM MESSAGE> is defined as follows:

<PROGRAM MESSAGE UNIT>

<PROGRAM MESSAGE UNIT UNIT SEPARATOR>

<PROGRAM MESSAGE UNIT>

<PROGRAM MESSAGE UNIT UNIT SEPARATOR>

<PROGRAM MESSAGE UNIT>

<PROGRAM MESSAGE UNIT UNIT SEPARATOR>

<PROGRAM MESSAGE UNIT>

<PROGRAM MESSAGE UNIT UNIT SEPARATOR>

<PROGRAM MESSAGE UNIT>

<PROGRAM MESSAGE UNIT UNIT SEPARATOR>

<PROGRAM MESSAGE UNIT>

<PROGRAM MESSAGE UNIT UNIT SEPARATOR>

<PROGRAM MESSAGE UNIT>

<PROGRAM MESSAGE UNIT UNIT SEPARATOR>

<PROGRAM MESSAGE UNIT>

<PROGRAM MESSAGE UNIT UNIT SEPARATOR>

<PROGRAM MESSAGE UNIT>

<PROGRAM MESSAGE UNIT UNIT SEPARATOR>

<PROGRAM MESSAGE UNIT>

<PROGRAM MESSAGE UNIT UNIT SEPARATOR>

<PROGRAM MESSAGE UNIT>

<PROGRAM MESSAGE UNIT UNIT SEPARATOR>

<PROGRAM MESSAGE UNIT>

<PROGRAM MESSAGE UNIT UNIT SEPARATOR>

<PROGRAM MESSAGE UNIT>

<PROGRAM MESSAGE UNIT UNIT SEPARATOR>

<PROGRAM MESSAGE UNIT>

<PROGRAM MESSAGE UNIT UNIT SEPARATOR>

<PROGRAM MESSAGE UNIT>

<PROGRAM MESSAGE UNIT UNIT SEPARATOR>

<PROGRAM MESSAGE UNIT>

<PROGRAM MESSAGE UNIT UNIT SEPARATOR>

<PROGRAM MESSAGE UNIT>

<PROGRAM MESSAGE UNIT UNIT SEPARATOR>

<PROGRAM MESSAGE UNIT>

<PROGRAM MESSAGE UNIT UNIT SEPARATOR>

<PROGRAM MESSAGE UNIT>

<PROGRAM MESSAGE UNIT UNIT SEPARATOR>

<PROGRAM MESSAGE UNIT>

<PROGRAM MESSAGE UNIT UNIT SEPARATOR>

<PROGRAM MESSAGE UNIT>

<PROGRAM MESSAGE UNIT UNIT SEPARATOR>

<PROGRAM MESSAGE UNIT>

<PROGRAM MESSAGE UNIT UNIT SEPARATOR>

<PROGRAM MESSAGE UNIT>

<PROGRAM MESSAGE UNIT UNIT SEPARATOR>

<PROGRAM MESSAGE UNIT>

<PROGRAM MESSAGE UNIT UNIT SEPARATOR>

<PROGRAM MESSAGE UNIT>

<PROGRAM MESSAGE UNIT UNIT SEPARATOR>

<PROGRAM MESSAGE UNIT>

<PROGRAM MESSAGE UNIT UNIT SEPARATOR>

<PROGRAM MESSAGE UNIT>

<PROGRAM MESSAGE UNIT UNIT SEPARATOR>

<PROGRAM MESSAGE UNIT>

<PROGRAM MESSAGE UNIT UNIT SEPARATOR>

<PROGRAM MESSAGE UNIT>

<PROGRAM MESSAGE UNIT UNIT SEPARATOR>

<PROGRAM MESSAGE UNIT>

<PROGRAM MESSAGE UNIT UNIT SEPARATOR>

<PROGRAM MESSAGE UNIT>

<PROGRAM MESSAGE UNIT UNIT SEPARATOR>

<PROGRAM MESSAGE UNIT>

<PROGRAM MESSAGE UNIT UNIT SEPARATOR>

<PROGRAM MESSAGE UNIT>

<PROGRAM MESSAGE UNIT UNIT SEPARATOR>

<PROGRAM MESSAGE UNIT>

<PROGRAM MESSAGE UNIT UNIT SEPARATOR>

<PROGRAM MESSAGE UNIT>

<PROGRAM MESSAGE UNIT UNIT SEPARATOR>

<PROGRAM MESSAGE UNIT>

<PROGRAM MESSAGE UNIT UNIT SEPARATOR>

<PROGRAM MESSAGE UNIT>

<PROGRAM MESSAGE UNIT UNIT SEPARATOR>

<PROGRAM MESSAGE UNIT>

<PROGRAM MESSAGE UNIT UNIT SEPARATOR>

<PROGRAM MESSAGE UNIT>

<PROGRAM MESSAGE UNIT UNIT SEPARATOR>

<PROGRAM MESSAGE UNIT>

<PROGRAM MESSAGE UNIT UNIT SEPARATOR>

<PROGRAM MESSAGE UNIT>

<PROGRAM MESSAGE UNIT UNIT SEPARATOR>

<PROGRAM MESSAGE UNIT>

<PROGRAM MESSAGE UNIT UNIT SEPARATOR>

<PROGRAM MESSAGE UNIT>

<PROGRAM MESSAGE UNIT UNIT SEPARATOR>

<PROGRAM MESSAGE UNIT>

<PROGRAM MESSAGE UNIT UNIT SEPARATOR>

<PROGRAM MESSAGE UNIT>

<PROGRAM MESSAGE UNIT UNIT SEPARATOR>

<PROGRAM MESSAGE UNIT>

<PROGRAM MESSAGE UNIT UNIT SEPARATOR>

<PROGRAM MESSAGE UNIT>

<PROGRAM MESSAGE UNIT UNIT SEPARATOR>

<PROGRAM MESSAGE UNIT>

<PROGRAM MESSAGE UNIT UNIT SEPARATOR>

<PROGRAM MESSAGE UNIT>

<PROGRAM MESSAGE UNIT UNIT SEPARATOR>

<PROGRAM MESSAGE UNIT>

<PROGRAM MESSAGE UNIT UNIT SEPARATOR>

<PROGRAM MESSAGE UNIT>

<PROGRAM MESSAGE UNIT UNIT SEPARATOR>

<PROGRAM MESSAGE UNIT>

<PROGRAM MESSAGE UNIT UNIT SEPARATOR>

<PROGRAM MESSAGE UNIT>

<PROGRAM MESSAGE UNIT UNIT SEPARATOR>

<PROGRAM MESSAGE UNIT>

<PROGRAM MESSAGE UNIT UNIT SEPARATOR>

<PROGRAM MESSAGE UNIT>

<PROGRAM MESSAGE UNIT UNIT SEPARATOR>

<PROGRAM MESSAGE UNIT>

<PROGRAM MESSAGE UNIT UNIT SEPARATOR>

<PROGRAM MESSAGE UNIT>

<PROGRAM MESSAGE UNIT UNIT SEPARATOR>

<PROGRAM MESSAGE UNIT>

<PROGRAM MESSAGE UNIT UNIT SEPARATOR>

<PROGRAM MESSAGE UNIT>

<PROGRAM MESSAGE UNIT UNIT SEPARATOR>

<PROGRAM MESSAGE UNIT>

<PROGRAM MESSAGE UNIT UNIT SEPARATOR>

<PROGRAM MESSAGE UNIT>

<PROGRAM MESSAGE UNIT UNIT SEPARATOR>

<PROGRAM MESSAGE UNIT>

<PROGRAM MESSAGE UNIT UNIT SEPARATOR>

<PROGRAM MESSAGE UNIT>

<PROGRAM MESSAGE UNIT UNIT SEPARATOR>

<PROGRAM MESSAGE UNIT>

<PROGRAM MESSAGE UNIT UNIT SEPARATOR>

<PROGRAM MESSAGE UNIT>

<PROGRAM MESSAGE UNIT UNIT SEPARATOR>

<PROGRAM MESSAGE UNIT>

<PROGRAM MESSAGE UNIT UNIT SEPARATOR>

<PROGRAM MESSAGE UNIT>

<PROGRAM MESSAGE UNIT UNIT SEPARATOR>

<PROGRAM MESSAGE UNIT>

<PROGRAM MESSAGE UNIT UNIT SEPARATOR>

<PROGRAM MESSAGE UNIT>

<PROGRAM MESSAGE UNIT UNIT SEPARATOR>

<PROGRAM MESSAGE UNIT>

<PROGRAM MESSAGE UNIT UNIT SEPARATOR>

<PROGRAM MESSAGE UNIT>

<PROGRAM MESSAGE UNIT UNIT SEPARATOR>

<PROGRAM MESSAGE UNIT>

<PROGRAM MESSAGE UNIT UNIT SEPARATOR>

<PROGRAM MESSAGE UNIT>

<PROGRAM MESSAGE UNIT UNIT SEPARATOR>

<PROGRAM MESSAGE UNIT>

<PROGRAM MESSAGE UNIT UNIT SEPARATOR>

<PROGRAM MESSAGE UNIT>

<PROGRAM MESSAGE UNIT UNIT SEPARATOR>

<PROGRAM MESSAGE UNIT>

<PROGRAM MESSAGE UNIT UNIT SEPARATOR>

<PROGRAM MESSAGE UNIT>

<PROGRAM MESSAGE UNIT UNIT SEPARATOR>

<PROGRAM MESSAGE UNIT>

<PROGRAM MESSAGE UNIT UNIT SEPARATOR>

<PROGRAM MESSAGE UNIT>

<PROGRAM MESSAGE UNIT UNIT SEPARATOR>

<PROGRAM MESSAGE UNIT>

<PROGRAM MESSAGE UNIT UNIT SEPARATOR>

<PROGRAM MESSAGE UNIT>

<PROGRAM MESSAGE UNIT UNIT SEPARATOR>

<PROGRAM MESSAGE UNIT>

<PROGRAM MESSAGE UNIT UNIT SEPARATOR>

<PROGRAM MESSAGE UNIT>

<PROGRAM MESSAGE UNIT UNIT SEPARATOR>

<PROGRAM MESSAGE UNIT>

<PROGRAM MESSAGE UNIT UNIT SEPARATOR>

<PROGRAM MESSAGE UNIT>

<PROGRAM MESSAGE UNIT UNIT SEPARATOR>

<PROGRAM MESSAGE UNIT>

<PROGRAM MESSAGE UNIT UNIT SEPARATOR>

<PROGRAM MESSAGE UNIT>

<PROGRAM MESSAGE UNIT UNIT SEPARATOR>

<PROGRAM MESSAGE UNIT>

<PROGRAM MESSAGE UNIT UNIT SEPARATOR>

<PROGRAM MESSAGE UNIT>

<PROGRAM MESSAGE UNIT UNIT SEPARATOR>

<PROGRAM MESSAGE UNIT>

<PROGRAM MESSAGE UNIT UNIT SEPARATOR>

<PROGRAM MESSAGE UNIT>

<PROGRAM MESSAGE UNIT UNIT SEPARATOR>

<PROGRAM MESSAGE UNIT>

<PROGRAM MESSAGE UNIT UNIT SEPARATOR>

<PROGRAM MESSAGE UNIT>

<PROGRAM MESSAGE UNIT UNIT SEPARATOR>

<PROGRAM MESSAGE UNIT>

<PROGRAM MESSAGE UNIT UNIT SEPARATOR>

<PROGRAM MESSAGE UNIT>

<PROGRAM MESSAGE UNIT UNIT SEPARATOR>

<PROGRAM MESSAGE UNIT>

<PROGRAM MESSAGE UNIT UNIT SEPARATOR>

<PROGRAM MESSAGE UNIT>

<PROGRAM MESSAGE UNIT UNIT SEPARATOR>

<PROGRAM MESSAGE UNIT>

<PROGRAM MESSAGE UNIT UNIT SEPARATOR>

<PROGRAM MESSAGE UNIT>

<PROGRAM MESSAGE UNIT UNIT SEPARATOR>

<PROGRAM MESSAGE UNIT>

<PROGRAM MESSAGE UNIT UNIT SEPARATOR>
3.1.2.5 **<PROGRAM MESSAGE UNIT SEPARATOR>**

<PROGRAM MESSAGE UNIT SEPARATOR> is defined as follows:

```
<White space>
(See Section 3.1.2.3.)
```

<PROGRAM MESSAGE UNIT SEPARATOR> separates a sequence of two or more <PROGRAM MESSAGE UNIT> elements into <PROGRAM MESSAGE> elements. The MP1800A/MT1810A interprets a semicolon (;) as the separator of the <PROGRAM MESSAGE UNIT>. The <white space character> elements before and after the semicolon are therefore ignored. The <white space character> is useful, however, to make the program readable.

3.1.2.6 **<PROGRAM MESSAGE UNIT>**

<PROGRAM MESSAGE UNIT> is defined as follows:

```
<COMMAND MESSAGE UNIT>
(See Section 3.1.2.7.)
```

<PROGRAM MESSAGE UNIT> consists of <COMMAND MESSAGE UNIT>, a single command message received by the MP1800A/MT1810A, or <QUERY MESSAGE UNIT>, a single query message. Setting and query can be performed for the MP1800A/MT1810A in units of <PROGRAM MESSAGE UNIT>.
3.1.2.7  <COMMAND MESSAGE UNIT> and <QUERY MESSAGE UNIT>

(1)  <COMMAND MESSAGE UNIT>

Example: The following shows a date setting command.

```
SENS:MEAS:EAL:PER 0,0,25,40
```

Figure 3.1.2.7-1  <COMMAND MESSAGE UNIT>

<COMMAND MESSAGE UNIT> is defined as follows:

(2)  <QUERY MESSAGE UNIT>

Example: The following shows a query that queries the area to add a bit error.

```
SOUR:PATT:EADD:AREA? APAT,8
```

Figure 3.1.2.7-2  <QUERY MESSAGE UNIT>
3.1 SCPI Listener Input Format

<QUERY MESSAGE UNIT> is defined as follows:

For both <COMMAND MESSAGE UNIT> and <QUERY MESSAGE UNIT>, when program data follows the program header, one space must be inserted as a separator between them. The program header identifies the application, function, and operation of the program. If no program data is added, the header alone indicates the application, function, and operation for the MP1800A/MT1810A.

Among program headers, <COMMAND PROGRAM HEADER> is a command used to control the MP1800A/MT1810A from the remote control PC. <QUERY PROGRAM HEADER> is a query command to be transmitted from the remote control PC to the MP1800A/MT1810A in advance for the remote control PC to receive response messages from the MP1800A/MT1810A. A query indicator (?) is added to the end of the header.
### 3.1.2.8 <COMMAND PROGRAM HEADER>

<COMMAND PROGRAM HEADER> is defined as follows:

A <white space> can be inserted in front of each header.

- **<COMMAND PROGRAM HEADER>**
  Indicates the application, function, and operation of the program data executed by the MP1800A/MT1810A. If no program data is added, the header alone indicates the application, function, and operation for the MP1800A/MT1810A. <program mnemonic> expresses the meaning in ASCII code characters, and is generally just called a "mnemonic".

1. **<Instrument-Control Headers>** is defined as follows:

   - **<short form mnemonic>**
   - **<long form mnemonic>**

   <short form mnemonic>/<long form mnemonic> Correspond to the short form and long form of the SCPI commands, respectively. For the specifications of mnemonics, the specifications of <program mnemonic> described earlier are applied as they are.

   - **<Instrument-Control Header>**
     <Instrument-Control Header> is defined in the SCPI. The device-unique commands of the MP1800A/MT1810A comply with the SCPI, so the command format conforms to the SCPI.
3.1 SCPI Listener Input Format

- <numeric suffix>
  Defined as a single ASCII code byte, within the range of the ASCII code bytes 30 to 39 (decimal numbers 48 to 57 = numerical values 0 to 9).

(2) <common command program header> is defined as follows:

   «Program mnemonic»

- <common command program header>
  For <common command program header>, an asterisk (*) must be added before the <program mnemonic> (refer to Chapter 5 "IEEE488.2 Common Commands" for details).

(3) <Program mnemonic> is defined as follows:

   «upper/lower case alpha»

- <program mnemonic>
  A mnemonic must start with an uppercase or lowercase alphabetic character, followed by any combination of uppercase alphabetic characters ("A" to "Z"), lowercase alphabetic characters ("a" to "z"), underbar (_), and numbers ("0" to "9"). The maximum length of a mnemonic is twelve characters. Space must not be inserted between characters.

- <upper/lower case alpha>
  Defined as a single ASCII code byte, within the range of the ASCII code bytes 41 to 5A and 61 to 7A (decimal numbers 65 to 90, 97 to 122 = uppercase alphabetic characters A to Z, lowercase alphabetic characters a to z).
Chapter 3  SCPI Format

- <digit>
  Defined as a single ASCII code byte, within the range of the ASCII code bytes 30 to 39 (decimal numbers 48 to 57 = numerical values 0 to 9).
- ( )
  Indicates the ASCII code byte 5F (decimal number 95 = underbar). It is defined as a single ASCII code byte.
  Example: :SYSTem (The subsequent part is omitted.)

3.1.2.9 <QUERY PROGRAM HEADER>

<QUERY PROGRAM HEADER> is defined as follows. A <white space> can be inserted in front of each header.

(1) <Instrument-Control Headers> is defined as follows:

(2) <common query program header> is defined as follows:
3.1 SCPI Listener Input Format

- <QUERY PROGRAM HEADER>

<QUERY PROGRAM HEADER> is a query command transmitted from the remote control PC to the MP1800A/MT1810A in advance for the remote control PC to receive response messages from the MP1800A/MT1810A. A query indicator (?) must be added to the end of the header.

Example: 'SYSTem:DATE?

The format of <QUERY PROGRAM HEADER> above is the same as <COMMAND PROGRAM HEADER> except that the query indicator (?) is added to the end of the header. For details, refer to Section 3.1.2.8 "<COMMAND PROGRAM HEADER>.'

3.1.2.10 <PROGRAM HEADER SEPARATOR>

![Diagram of PROGRAM HEADER SEPARATOR]

Figure 3.1.2.10-1  <PROGRAM HEADER SEPARATOR>

Definition <PROGRAM HEADER SEPARATOR> is defined as follows:

- <White space>
  (See Section 3.1.2.3.)

<PROGRAM HEADER SEPARATOR> is used as a separator between <COMMAND PROGRAM HEADER> or <QUERY PROGRAM HEADER> and <PROGRAM DATA>. If there are two or more <white space character> elements between the program header and program data, only the first <white space character> is interpreted as a separator, and the others are ignored. The <white space character> is useful, however, to make the program readable. That is, only one header separator must exist between the header and data, which indicates the end of the program as well as the beginning of the program data.
3.1.2.11 <PROGRAM DATA SEPARATOR>

<PROGRAM DATA SEPARATOR> is defined as follows:

If <COMMAND PROGRAM HEADER> or <QUERY PROGRAM HEADER> has many parameters, <PROGRAM DATA SEPARATOR> is used to separate them. Using this data separator requires commas, but no <white space character>. <white space character> before and after the comma are ignored. The <white space character> is useful, however, to make the program readable.
3.13 Program data format

Among the terminated program message formats described above, this chapter presents an example of the format of <PROGRAM DATA> shown in the functional grammar diagram in Section 3.1.2.

![Figure 3.1.3-1 <PROGRAM DATA>](image)

The functional elements of <PROGRAM DATA> are used to transmit the parameters of the type related to the program header. The following table lists the program data used by the MP1800A/MT1810A.

<table>
<thead>
<tr>
<th>&lt;PROGRAM DATA&gt;</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;CHARACTER PROGRAM DATA&gt;</td>
<td>Indicates short alphabetic or alphanumerical data.</td>
</tr>
<tr>
<td>&lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
<td>Indicates a decimal numerical constant.</td>
</tr>
<tr>
<td>&lt;NON-DECIMAL NUMERIC PROGRAM DATA&gt;</td>
<td>Indicates an alphanumerical character other than a decimal.</td>
</tr>
<tr>
<td>&lt;BOOLEAN PROGRAM DATA&gt;</td>
<td>Indicates a theoretical value (defined in SCPI).</td>
</tr>
<tr>
<td>&lt;STRING PROGRAM DATA&gt;</td>
<td>Indicates a string enclosed within double quotation marks (&quot; &quot;) or single quotation marks (').</td>
</tr>
</tbody>
</table>

### 3.1.3.1 <CHARACTER PROGRAM DATA>

<CHARACTER PROGRAM DATA> indicates short alphabetic or alphanumerical data.

Example:

```
: SENSEs : PATTern : TYPE PRBS (Mnemonic data indicating PRBS)
: SENSEs : MEASure : EALarm : UNIT CLOCK
```

(Mnemonic data indicating the measurement cycle in Clock Count units.)

The character data is the same as those described "<program mnemonic>" in Section 3.1.2.8 (3).
3.1.3.2 **<DECIMAL NUMERIC PROGRAM DATA>**

<DECIMAL NUMERIC PROGRAM DATA> indicates a decimal numerical value, and is defined as follows:

```
<mantissa> <White space> <exponent>
```

<mantissa> is defined as follows:

```
<digit> . <optional digits> <optional digits> <digit>
```

<exponent> is defined as follows:

```
E/e <white space> <digit> + <digit> -
```

The MP1800A/MT1810A uses the decimal integer format.
- Integer format
- $\Delta$ represents a space.
- Indicates a decimal integer value.
- Zeros can be entered in the beginning. → 005
- No space can be inserted between the sign and the numerical value. → +5 (applicable), +5$\Delta$ (not applicable)
- Spaces can be inserted after the numerical value. → +5$\Delta$
- Positive sign (+) may be omitted. → +5, 5
- Commas cannot be used for separating digits. → 1,234 (not applicable)

Example: : SOURce : PATTern : PRBS : LENGth 7
3.1.3.3 <NON-DECIMAL NUMERIC PROGRAM DATA>

<NON-DECIMAL NUMERIC PROGRAM DATA> indicates a numerical value other than a decimal (e.g., binary, hexadecimal), and is defined as follows:

- <hexadecimal digit>: Hexadecimal format
  The hexadecimal format consists of #H and the subsequent numbers (0 to 9) and/or alphabetic characters (A to F). \( \Delta \) represents a space.
  - Spaces can be inserted following #H.
    
    #H1234 \( \rightarrow \) #H\( \Delta \)1234
    #H00AF \( \rightarrow \) #H\( \Delta \Delta \Delta \)00AF
  - Zeros can be omitted.
    #H00FF \( \rightarrow \) #HFF
    #H0000 \( \rightarrow \) #H0

Example:

: SOURce : PATTern : DREVerse : ADDRess #H0,#H1F

- <binary digit>: Binary format
  The binary format consists of #B and the subsequent 0s and/or 1s. \( \Delta \) represents a space.
  - Spaces can be inserted following #B.
    #B11011011 \( \rightarrow \) #B\( \Delta \)11011011
    #B00100100 \( \rightarrow \) #B\( \Delta \Delta \Delta \)00100100

*SCPI Format*
3.1.3.4  **<BOOLEAN PROGRAM DATA>**

<BOOLEAN PROGRAM DATA> is the program data that is defined in the SCPI, and indicates a theoretical value. As the values corresponding to True and False, ON and OFF of <CHARACTER PROGRAM DATA> and 1 and 0 of <DECIMAL NUMREIC PROGRAM DATA> are defined.

Example:

```
: SOURce : PATtern : EADDition : SET ON
: SOURce : PATtern : EADDition : SET 1
```
3.1.3.5 <STRING PROGRAM DATA>

<String Program Data> is character string data, enclosed within double quotation marks (" ") or single quotation marks (’ ’). If the character string includes a double (" ") or single quotation mark (’ ’), the same type of quotation mark must be described to enclose the string.

It is defined as follows:

1. <inserted'> is defined as a single ASCII symbol of a value 27 (decimal number, 39 = ').
2. <non-single quote char> is defined as a single ASCII symbol of a value other than 27 (decimal number, 39 = ').
3. <inserted"> is defined as a single ASCII symbol of a value 22 (decimal number, 34 = ").
4. <non-double quote char> is defined as a single ASCII symbol of a value other than 22 (decimal number, 34 = ").
<Example of description>
The following show description examples.

When a character string is enclosed within single quotation marks (‘ ’):
'calculate'
'remote"control’ (The double quotation mark (‘”) between single quotation marks is regarded as a string.)
"Jan. " "Feb. " ’ (The double quotation marks ("”) between single quotation marks are regarded as a string.)
'remote"control’ (The double quotation mark (‘”) between single quotation marks is regarded as a string.)

When a character string is enclosed within double quotation marks (" "):
"calculate"
"It’s a nice day. " (The single quotation mark (‘’) between double quotation marks is regarded as a string.)
" ‘Mar. "Apr. ’ " (The single quotation marks (‘’) and the double quotation mark (‘”) between outer double quotation marks are regarded as a string.)
"program" "data" (The double quotation marks ("”) between outer double quotation marks are regarded as a string.)

As shown above, use the same type of quotation marks for enclosing a character string.

Actual commands are described as follows:

Examples: 
:DISP:CUST:BUTT:ADD "EADDition" (String indicating the Error ADDition function button)
:CALC:OPTical:STATus? "TX:LASer" (String indicating whether optical output of the Module is normal)
3.2 SCPI Talker Output Format

This section describes the format of response messages returned from the talker (transmitter) to the listener (receiver).

3.2.1 SCPI talker output response message format

The following figure shows the response to a data output query command :OUTP:DATA:OUTP? and test pattern selection query commands :SOUR:PATT:TYPE? and :OUTPut:RCLock:SeLect?. The SCPI response has no header, so the response contains only data.

As in the case of program messages, the format of response messages consists of a sequence of functional elements, which are the minimum level units to indicate a function. In the figure above, the words written in uppercase alphabetical characters enclosed within brackets (< >) indicate examples of functional elements. A functional element is further divided into coding elements. In this figure, the words written in lowercase alphabetical characters enclosed within brackets (< >) indicate examples of coding elements. Syntax notations are therefore the same for both the talker and listener.
3.2.2 Functional elements of response messages

3.2.2.1 <TERMINATED RESPONSE MESSAGE>

The following figure shows a <TERMINATED RESPONSE MESSAGE> with two message units linked.

![Diagram of <TERMINATED RESPONSE MESSAGE> with two message units linked.]

**Figure 3.2.2.1-1 <TERMINATED RESPONSE MESSAGE>**

Definition <TERMINATED RESPONSE MESSAGE> is defined as follows:

<TERMINATED RESPONSE MESSAGE> is a data message containing all the functional elements required for transmitting the message from the talker (MP1800A/MT1810A) to the remote control PC. A <RESPONSE MESSAGE TERMINATOR> is added to the end of a <RESPONSE MESSAGE> to complete transmission of <RESPONSE MESSAGE>.
3.2.2.2 <RESPONSE MESSAGE TERMINATOR>

<RESPONSE MESSAGE TERMINATOR> is placed after the last <RESPONSE MESSAGE UNIT> to terminate a sequence of one or more <RESPONSE MESSAGE UNIT> elements. The definition of <RESPONSE MESSAGE TERMINATOR> differs according to the used interface.

(1) For GPIB interface:

```
NL —END
```

(2) For Ethernet interface:

```
NL
```

NL Defined as a single ASCII code byte 0A (decimal number 10), i.e. the ASCII control character LF (Line Feed) that performs the carriage return operation bringing the print position to the same character position in the next line. It is also called NL (New Line) because it starts from a new line.

For line feed, the CR+LF code may be used instead of the LF code.

If the remote control PC runs on MS-DOS or Windows, line feed is done with "CR+LF", while it is done with only "LF" for UNIX.

END Generates an EOI signal by setting the EOI (End-or-Identify) line of the GPIB control bus to TRUE (low level).

---

Example: A typical program that reads the state of the Synthesizer reference signal currently set is shown below.

```
10 WRITE @03:":OUTP:RCL:SEL?"
20 READ @03:A$
30 PRINT A$
40 END
```
3.2.2.3  <RESPONSE MESSAGE>

The following figure shows an example of a response to a query command that queries the selected input connector and a query command that queries the bit rate set for reception signals.

![Figure 3.2.2.3-1  <RESPONSE MESSAGE>](image)

**Figure 3.2.2.3-1  <RESPONSE MESSAGE>**

Definition

<RESPONSE MESSAGE> is defined as follows:

<RESPONSE MESSAGE> is a sequence of one or more <RESPONSE MESSAGE UNIT> elements. The <RESPONSE MESSAGE UNIT> element indicates a single message to be sent from the MP1800A/MT1810A to the remote control PC. The <RESPONSE MESSAGE UNIT SEPARATOR> element is used to separate two or more <RESPONSE MESSAGE UNIT> elements.

3.2.2.4  <RESPONSE MESSAGE UNIT SEPARATOR>

<RESPONSE MESSAGE UNIT SEPARATOR> is defined as follows:

<RESPONSE MESSAGE UNIT SEPARATOR> separates two or more <RESPONSE MESSAGE UNIT> elements with the <UNIT SEPARATOR> semicolon (;) when a sequence of them is output in one <RESPONSE MESSAGE>.
3.2.2.5  **<RESPONSE MESSAGE UNIT>**

<RESPONSE MESSAGE UNIT> of the MP1800A/MT1810A is a response message unit with no header, and returns only the measurement result data. <RESPONSE MESSAGE UNIT> is defined as follows:

![Diagram](image)

3.2.2.6  **<RESPONSE DATA SEPARATOR>**

<RESPONSE DATA SEPARATOR> is used to separate data when two or more <RESPONSE DATA> elements are output.

<RESPONSE DATA SEPARATOR> is defined as follows:

![Diagram](image)
3.2.2.7 <RESPONSE DATA>

The <RESPONSE DATA> elements used in the MP1800A/MT1810A are described below. The response data to be returned depends on the query message.

Table 3.2.2.7-1 Response data

<table>
<thead>
<tr>
<th>Element</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1) CHARACTER RESPONSE DATA &lt;response mnemonic&gt;</td>
<td>Indicates short mnemonic data.</td>
</tr>
<tr>
<td>Example: ABC</td>
<td></td>
</tr>
<tr>
<td>DEFG</td>
<td></td>
</tr>
</tbody>
</table>

| (2) NR1 NUMERIC RESPONSE DATA | Indicates a decimal integer. |
| Example: 123 |  |
| +123 |  |
| −1234 |  |

| (3) NR2 NUMERIC RESPONSE DATA | Indicates a fixed-point numerical value. |
| Example: 12.3 |  |
| +12.34 |  |
| −12.345 |  |

| (4) NR3 NUMERIC RESPONSE DATA | Indicates a real decimal number with an exponent. |
| Example: 1.23E+45 |  |
| −12.3E+45 |  |
### Table 3.2.7-1  Response data (Cont'd)

<table>
<thead>
<tr>
<th>Element</th>
<th>Function</th>
</tr>
</thead>
</table>
| (5) STRING RESPONSE DATA  
Example:  
"1234"  
"ABCD"  
"1234.5" | Indicates a character string enclosed within double quotation marks(" "). |

### Diagram for (5)

- " "  
  - "<inserted>"  
  - "<non-double quote char>"  

(6) ARBITRARY ASCII RESPONSE DATA  
Example:  
<ASCII Byte><ASCII Byte>NL^END | Transmits an ASCII data byte without separation, excluding NL characters. The message is terminated without exit as NL^END (or NL only) is inserted next to the last data.  
1) For GPIB interface:  
- <digit>  
  - NL  
  - ^END  

### Diagram for (6)

- <digit>  
  - NL  
  - ^END  
  - <ASCII data byte>  

2) For Ethernet interface:

- <ASCII data byte>  
  - NL
3.2.2.8 Syntax differences between listener input format and talker output format

The differences in syntax between the listener input format and the talker output format are as follows:

- Listener input format
  A listener input message possesses flexibility so that the MP1800A/MT1810A can easily receive program messages from the remote control PC.
  Easy-to-read programs can be created since uppercase and lowercase alphabetical are not identified (non case sensitive), and any <white spaces> can be added as desired to the separator or terminator in this format.

- Talker output format
  A talker output message is, on the other hand, transmitted strictly according to the syntax so that the remote control PC can easily accept response messages output from the MP1800A/MT1810A.
  There is only one response message for one function.

<table>
<thead>
<tr>
<th>Item</th>
<th>Listener Input Program Message</th>
<th>Talker Output Program Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>Characteristics</td>
<td>Flexible</td>
<td>Strict</td>
</tr>
<tr>
<td>Alphabetic characters</td>
<td>Uppercase and lowercase characters have the same meanings (non case sensitive).</td>
<td>Uppercase characters only</td>
</tr>
<tr>
<td>Before and after the NR3 exponent part E</td>
<td>$E_\Delta + , e^+E$ ($\Delta$ is 0 or more)</td>
<td>Uppercase E only</td>
</tr>
<tr>
<td>Positive sign (+) of the NR3 exponent part</td>
<td>Can be omitted</td>
<td>Cannot be omitted</td>
</tr>
<tr>
<td>&lt;white space&gt; ( : $\Delta$)</td>
<td>Two or more &lt;white space&gt; can be added before and after the separator and before the terminator.</td>
<td>Not used</td>
</tr>
<tr>
<td>Unit separator</td>
<td>$\Delta + ;$ (Δ is 0 or more)</td>
<td>Semicolon only</td>
</tr>
<tr>
<td>Blank before header</td>
<td>$\Delta + -$ (Δ is 0 or more)</td>
<td>Header only</td>
</tr>
<tr>
<td>Header separator</td>
<td>Header + $\Delta$ (Δ is 1 or more)</td>
<td>Header + one $20$</td>
</tr>
<tr>
<td>Data separator</td>
<td>$\Delta + , ;$ (Δ is 0 or more)</td>
<td>Comma only</td>
</tr>
<tr>
<td>Terminator</td>
<td>$\Delta + {\text{NL}, \text{EOI}, \text{or} \text{NL+EOI}}$ (Δ is 0 or more)</td>
<td>NL+EOI</td>
</tr>
</tbody>
</table>

*1: $\Delta$ indicates <white space>.
*2: ASCII code byte 20 (decimal number 32 = ASCII character SP, space)
*3: "NL+EOI" and "CR+NL+EOI" can be switched by the SYSTEM:TERMINation command. CR is defined as the ASCII code 0D.
3.3 Command Configuration

The SCPI commands have a hierarchic structure. The SCPI are grouped by the related functions, and each group forms a hierarchic structure called a subsystem. In this document, subsystems are expressed in a command tree, as shown below.

Table 3.3-1 Example of SCPI command tree

<table>
<thead>
<tr>
<th>:INPut</th>
<th>:CLOCk</th>
<th>:SELection</th>
<th>&lt;clock&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>:SELaction?</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>:RECoverey</td>
<td>&lt;string&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:RECoverey?</td>
<td></td>
</tr>
</tbody>
</table>

The same headers can exist in an SCPI command tree, and the position where the header exists corresponds to its function. Thus, a command must be described in full path to the header used.
3.4 Command Syntax

Table 3.4-1  Examples of SCPI commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>:INPut:CLOCk:SELection &lt;clock&gt;</td>
<td>A command always starts with a colon (:) and consists of headers connected by colons (:).</td>
</tr>
<tr>
<td>:INPut:CLOCk:SELection?</td>
<td>A command always starts with a colon (:) and consists of headers connected by colons (:).</td>
</tr>
<tr>
<td>:INPut:CLOCk:RECovery &lt;string&gt;</td>
<td>The headers are classified into short and long forms. The short form is an abbreviated form of the long form. The command is interpreted as the identical command, regardless whether it is described in the short form or in the long form. Short and long forms can also be used in combination. Although uppercase and lowercase characters are used to distinguish between the short and long forms (uppercase characters indicate the short form) in this document, they are not case sensitive when actually used.</td>
</tr>
<tr>
<td></td>
<td>Short form</td>
</tr>
<tr>
<td></td>
<td>Long + short form</td>
</tr>
<tr>
<td></td>
<td>&gt; :INPUT:CLOCK:SELECTION RECOVERED</td>
</tr>
<tr>
<td></td>
<td>&gt; :INP:CLOC:SEL REC</td>
</tr>
<tr>
<td></td>
<td>&gt; :Inp:CLOC:SELECTION REC</td>
</tr>
</tbody>
</table>

<Optional node>
Square brackets ([ ]) indicate optional nodes. A header enclosed within square brackets can be omitted; it is interpreted as being the same command, regardless of whether it is omitted.

Example:
:STATus:OPERation:[EVENt]?
When a header is not omitted  > :STATus:OPERation:EVENt?
When a header is omitted     > :STATus:OPERation?

<Header separator>
At least one space must be inserted between a command and parameter. Two or more parameters must be separated by commas (,).
3.5 Command Combinations

Commands can be combined using a semicolon (;), as shown in the examples below. The second command is referred to as the same level as the lowest hierarchy of the first command. Thus, the second command can be described in full path, as shown in Example 1, or described omitting the higher-level headers than "SELection", as shown in Example 2.

Example 1: >:INPut:CLOCk:SELection
               RECovered::INPut:CLOCk:RECovery" "OC_3"
Example 2: > :INPut:CLOCk:SELection RECovered:ReCOvery "OC_3"

Note:

Commands that handle some kind of binary data cannot be combined.

Example:

:SOURce:PATTern:BDATA:WHOLe
WRT, RED?
3.6 Parameters

The following table shows the parameter types used in the MP1800A/MT1810A. In this document, parameter types are indicated by the lowercase alphabetical characters within brackets (< >). The corresponding <PROGRAM DATA> types as defined in IEEE488.2 (or SCPI) are indicated by uppercase alphabetical.

Table 3.6-1  Description of parameter types

<table>
<thead>
<tr>
<th>Parameter Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;numeric&gt; &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
<td>Indicates a decimal integer.</td>
</tr>
<tr>
<td>&lt;numeric&gt; &lt;NON-DECIMAL NUMERIC PROGRAM DATA&gt;</td>
<td>Indicates a numeric value other than a decimal (binary, octal, etc.).</td>
</tr>
<tr>
<td>&lt;boolean&gt; &lt;BOOLEAN PROGRAM DATA&gt;</td>
<td>Indicates a logical value. OFF or 0 corresponds to False, and ON or 1 corresponds to True. Either 0/1 or OFF/ON can be used for setting, but responses to a query use 0/1.</td>
</tr>
<tr>
<td>&lt;string&gt;,&lt;display&gt;...etc &lt;STRING PROGRAM DATA&gt;</td>
<td>Indicates a character string. It is an ASCII character string enclosed within single quotation marks (’’) or double quotation marks (“”). Example: ‘SES_3:DM_6’ or &quot;SES_3:DM_6&quot;</td>
</tr>
<tr>
<td>&lt;brate&gt;,&lt;type&gt;...etc &lt;CHARACTER PROGRAM DATA&gt;</td>
<td>Indicates character data. It is expressed as a short string corresponding to the setting details.</td>
</tr>
<tr>
<td>&lt;bdata&gt;...etc &lt;ARBITRARY BLOCK PROGRAM DATA&gt;</td>
<td>Used to transmit 8-bit binary data.</td>
</tr>
</tbody>
</table>
### 3.7 Responses

The following table shows the response types used in the MP1800A/MT1810A. In this document, response types are indicated by the lowercase alphabetical characters enclosed within brackets (< >). The corresponding <RESPONSE DATA> types as defined in IEEE488.2 (or SCPI) are indicated by uppercase alphabetical.

**Table 3.7-1 Description of response types**

<table>
<thead>
<tr>
<th>Response Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;numeric&gt;</td>
<td>The number of digits of the response is variable, and the maximum number of digits of the numerical value range is the maximum number of digits of the response. No space must be inserted between the sign and numerical value.</td>
</tr>
<tr>
<td>&lt;NR1 NUMERIC RESPONSE DATA&gt;</td>
<td>&gt; :SYSTem:DATE? &lt; 2006,7,14</td>
</tr>
<tr>
<td>&lt;numeric&gt;</td>
<td>The number of digits of the response is variable, and the maximum number of digits of the numerical value (including decimal numbers) range is the maximum number of digits of the response. A space must not be inserted between the sign and numerical value.</td>
</tr>
<tr>
<td>&lt;NR2 NUMERIC RESPONSE DATA&gt;</td>
<td>&gt; :OUTPut:DATA:AMPLitude? &lt; 1.000</td>
</tr>
<tr>
<td>&lt;brate&gt;,&lt;type&gt;...etc</td>
<td>Returns the short form of the character.</td>
</tr>
<tr>
<td>&lt;string&gt;,&lt;display&gt;...etc</td>
<td>Returns a string enclosed within double quotation marks (&quot; &quot;). If there are short and long forms for the string of the corresponding program command, the short form is returned.</td>
</tr>
<tr>
<td>&lt;STRING RESPONSE DATA&gt;</td>
<td>&gt; :DISPlay:CUSTomize:BUTTon:NOW? &lt; 1, &quot;EADD&quot; (The short form of &quot;EADDition&quot;)</td>
</tr>
<tr>
<td>&lt;ARBITRARY ASCII RESPONSE DATA&gt;</td>
<td>Returns 7-bit ASCII text data without separation. The last data byte is terminated by NL^END or NL only.</td>
</tr>
<tr>
<td></td>
<td>&gt; *OPT? &lt; OPT301,OPT302,MU181000A,OPT101,...</td>
</tr>
</tbody>
</table>
Chapter 4  Native Format

This chapter describes the Native command system.

4.1  Native Listener Input Format ................................. 4-2
4.2  Native Talker Output Format ................................. 4-6
4.3  Command Combinations ....................................... 4-10
4.4  Parameters .......................................................... 4-10
4.5  Responses ............................................................ 4-11
4.1 Native Listener Input Format

A program message consists of a sequence of program message units. Each unit is a program instruction (command) or program query.

In the following figure, two program message units "DONΔ1" and "CONΔ1" are connected with the program message unit separator, and transmitted as one program message from the remote control PC to the MP1800A/MT1810A to set data output and clock output to ON.

IEEE488.2 defines program messages as sequences of functional elements. A functional element is the minimum level unit to indicate a function. In the figure above, the uppercase alphabetical characters enclosed within brackets (<> ) indicate some examples.

Functional elements can be roughly classified into four types: separators, terminators, program headers, and program data. This section describes the command formats of the MP1800A/MT1810A for each of the four types (triangles (Δ) indicate spaces).
4.1 Native Listener Input Format

■ Separator
Separators have the three functional elements shown in the table below.

```
<PROGRAM MESSAGE UNIT SEPARATOR>
DON 1 ; ERT 1,3 <LF>
<PROGRAM HEADER SEPARATOR>
<PROGRAM DATA SEPARATOR>
```

<table>
<thead>
<tr>
<th>Functional Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;PROGRAM MESSAGE UNIT SEPARATOR&gt;</td>
<td>Separates two or more &lt;PROGRAM MESSAGE UNIT&gt; elements. It is indicated as 0 or more spaces + one semicolon.</td>
</tr>
<tr>
<td>&lt;PROGRAM DATA SEPARATOR&gt;</td>
<td>Separates two or more &lt;PROGRAM DATA&gt; elements. It is indicated as 0 or more spaces + one comma + 0 or more spaces.</td>
</tr>
<tr>
<td>&lt;PROGRAM HEADER SEPARATOR&gt;</td>
<td>Separates a program header and &lt;PROGRAM DATA&gt; element. It is indicated as one or more spaces.</td>
</tr>
</tbody>
</table>

■ Terminator
Terminators have the functional element shown in the table below.

```
<PROGRAM MESSAGE>
DON 1 ; ERT 1,3 <LF>
<PROGRAM MESSAGE UNIT>
<PROGRAM MESSAGE TERMINATOR>
```

<table>
<thead>
<tr>
<th>Functional Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;PROGRAM MESSAGE TERMINATOR&gt;</td>
<td>Added at the end of &lt;PROGRAM MESSAGE&gt; to terminate a sequence with one or more &lt;PROGRAM MESSAGE UNIT&gt; elements. It is indicated as 0 or more spaces + (CR+LF or LF). For the GPIB, EOI may or may not be asserted at the same time as the transmission of LF.</td>
</tr>
</tbody>
</table>
Program headers have the two functional elements shown in the table below.

![DON Δ1; DGD? <LF>](image)

### Table 4.1-3 Program header functional elements

<table>
<thead>
<tr>
<th>Functional Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;COMMAND PROGRAM HEADER&gt;</td>
<td>This is a command used by the remote control PC to control the MP1800A/MT1810A. The &lt;COMMAND PROGRAM HEADER&gt; of the MP1800A/MT1810A uses &lt;simple command program header&gt; as a coding syntax. The device-unique commands use this syntax. Example: DGDA1</td>
</tr>
<tr>
<td>&lt;QUERY PROGRAM HEADER&gt;</td>
<td>This is a query command to be transmitted in advance to the MP1800A/MT1810A so that the remote control PC receives a response message from the MP1800A/MT1810A. The query indicator “?” is always added to the end of the header. The &lt;QUERY PROGRAM HEADER&gt; of the MP1800A/MT1810A uses &lt;simple query program header&gt; as a coding syntax. The device-unique query commands use this syntax. Example: DGD?</td>
</tr>
</tbody>
</table>
### Program data

The program data of the MP1800A/MT1810A has the three functional elements shown in the table below.

In detailed description of the commands, the types of program data defined in IEEE488.2 corresponding to the device-unique parameter types are described.

<table>
<thead>
<tr>
<th>Functional Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
<td>Indicates a decimal integer/real number. Integers and fixed-point values are used in the MP1800A/MT1810A.</td>
</tr>
<tr>
<td></td>
<td>Example: Integers 005, +5, 5, +5Δ Fixed-point values 12.345, 05, +0.05, 12.</td>
</tr>
<tr>
<td>&lt;NON-DECIMAL NUMERIC PROGRAM DATA&gt;</td>
<td>Indicates a numeric value other than a decimal. Hexadecimal and binary values are used in the MP1800A/MT1810A. Hexadecimal values are indicated as #H + {A to F, a to f, 0 to 9}. Binary values are indicated as #B + 0/1.</td>
</tr>
<tr>
<td></td>
<td>Example: #HABC, #H123 #B101010</td>
</tr>
<tr>
<td>&lt;ARBITRARY BLOCK PROGRAM DATA&gt;</td>
<td>Used to transmit binary data of the specified number of bytes.</td>
</tr>
<tr>
<td></td>
<td>Example: #11A</td>
</tr>
</tbody>
</table>
4.2 Native Talker Output Format

Typical response messages include measurement results, setting states, and status information. Two types of response messages are provided: response messages returned with a header and those returned with no header.

In the following figure, in response to the data output query message unit "DON?" and clock output query message unit "CON?", the corresponding response messages are transmitted from the MP1800A/MT1810A to the remote control PC as ASCII strings with a header.

As in the case of program messages, response messages are defined as sequences of functional elements. Functional elements can be roughly classified into four types: separators, terminators, response headers, and response data. This section describes the command formats of the MP1800A/MT1810A for each of the four types (triangles (Δ) indicate spaces). The major differences in the syntax between the input format of the listener device and the output format of the talker device are as follows:

- **Listener input format**
  A listener input message possesses flexibility so that the MP1800A/MT1810A can easily receive program messages from the remote control PC. Therefore, any differences in a program message description can be compensated to some degree. For example, any <white spaces> can be added as desired to the separator or terminator in this format, enabling to create easy-to-read programs.

- **Talker output format**
  A talker output message is, on the other hand, transmitted strictly according to the syntax so that the remote control PC can easily accept response messages output from the MP1800A/MT1810A. There is only one response message for one function.
Table 4.2-1 summarizes the differences between the listener input format and talker output format.

<table>
<thead>
<tr>
<th>Item</th>
<th>Listener Input Program Message</th>
<th>Talker Output Response Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>Characteristics</td>
<td>Flexible</td>
<td>Strict</td>
</tr>
<tr>
<td>Alphabetic characters</td>
<td>Not case-sensitive except for the header Only uppercase characters for the header</td>
<td>Uppercase characters only</td>
</tr>
<tr>
<td>Before and after the NR3 exponent part E</td>
<td>Uppercase E only</td>
<td>Uppercase E only</td>
</tr>
<tr>
<td>Positive sign (+) of the NR3 exponent part</td>
<td>+ can be omitted</td>
<td>Cannot be omitted.</td>
</tr>
<tr>
<td>Space</td>
<td>Spaces can be added before and after the separator and after the terminator</td>
<td>Not used</td>
</tr>
<tr>
<td>Unit separator</td>
<td>0 or more spaces + semicolon</td>
<td>Semicolon only</td>
</tr>
<tr>
<td>Blank before header</td>
<td>0 or more spaces + header</td>
<td>Header only</td>
</tr>
<tr>
<td>Header separator</td>
<td>Header + one or more spaces</td>
<td>Header + one $20*1</td>
</tr>
<tr>
<td>Data separator</td>
<td>0 or more spaces + comma + 0 or more spaces</td>
<td>Not used</td>
</tr>
<tr>
<td>Terminator</td>
<td>0 or more spaces + any one of the following: [ LF ] [ CR+LF ] [ LF+EOI ] [ CR+LF+EOI ]</td>
<td>Either LF+EOI or CR+LF+EOI*2</td>
</tr>
</tbody>
</table>

*1: ASCII code byte 20 (decimal 32 = ASCII character SP, space)
*2: Can be switched by using the TRM command.
Separator
Separators have the three functional elements shown in the table below.

Table 4.2-2 Separator functional elements

<table>
<thead>
<tr>
<th>Functional element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;RESPONSE MESSAGE UNIT SEPARATOR&gt;</td>
<td>Separates two or more &lt;RESPONSE MESSAGE UNIT&gt; elements. It is indicated as</td>
</tr>
<tr>
<td></td>
<td>a semicolon.</td>
</tr>
<tr>
<td>&lt;RESPONSE HEADER SEPARATOR&gt;</td>
<td>Separates &lt;RESPONSE HEADER&gt; and &lt;RESPONSE DATA&gt; elements. It is indicated as</td>
</tr>
<tr>
<td></td>
<td>one space.</td>
</tr>
<tr>
<td>&lt;RESPONSE DATA SEPARATOR&gt;</td>
<td>Separates two or more &lt;RESPONSE DATA&gt; elements. It is indicated as a comma.</td>
</tr>
</tbody>
</table>

Terminator
Terminators have the functional element shown in the table below.

Table 4.2-3 Terminator functional element

<table>
<thead>
<tr>
<th>Functional Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;RESPONSE MESSAGE TERMINATOR&gt;</td>
<td>Added at the end of &lt;RESPONSE MESSAGE&gt; to terminate message transfer. It is</td>
</tr>
<tr>
<td></td>
<td>indicated as LF or CR+LF (For the GPIB, EOI is asserted at the same time</td>
</tr>
<tr>
<td></td>
<td>as the transmission of LF).</td>
</tr>
</tbody>
</table>
4.2  Native Talker Output Format

- Response header
  Response headers have the functional element shown in the table below.

<table>
<thead>
<tr>
<th>Functional Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;RESPONSE HEADER&gt;</td>
<td>Up to 12 uppercase alphanumeric characters or asterisks that indicate the function of the response data starting with an uppercase character. Example: DONA0</td>
</tr>
</tbody>
</table>

- Response data
  The response data of the MP1800A/MT1810A has the six functional elements shown in the table below.
  In the detailed description of the commands, the types of response data as defined in IEEE488.2 corresponding to the device-unique response types are described.

<table>
<thead>
<tr>
<th>Functional Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;NR1 NUMERIC RESPONSE DATA&gt;</td>
<td>Indicates a decimal integer.</td>
</tr>
<tr>
<td>&lt;NR2 NUMERIC RESPONSE DATA&gt;</td>
<td>Indicates a decimal fixed-point value.</td>
</tr>
<tr>
<td>&lt;NR3 NUMERIC RESPONSE DATA&gt;</td>
<td>Indicates a decimal exponent.</td>
</tr>
<tr>
<td>&lt;HEXADECIMAL NUMERIC RESPONSE DATA&gt;</td>
<td>Indicates a hexadecimal numeric value. Hexadecimal values are indicated as #H + {A to F, a to f, 0 to 9}.</td>
</tr>
<tr>
<td>&lt;STRING RESPONSE DATA&gt;</td>
<td>Indicated as a character string enclosed within double quotation marks.</td>
</tr>
<tr>
<td>&lt;DEFINITE LENGTH ARBITRARY BLOCK RESPONSE DATA&gt;</td>
<td>Indicates 8-bit binary data of a known length.</td>
</tr>
</tbody>
</table>
4.3 Command Combinations

Commands can be combined by using a semicolon (;) as shown in the example below.

Example: > DON 1; CON 1

4.4 Parameters

The following table shows the parameter types used in the MP1800A/MT1810A. In this document, parameter types are indicated by the lowercase alphabetical characters enclosed within brackets (< >). The corresponding <PROGRAM DATA> types as defined in IEEE488.2 are indicated in uppercase characters.

<table>
<thead>
<tr>
<th>Parameter Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;numeric&gt;</td>
<td>Indicates a decimal integer/real number. Integers, fixed-point values, and exponents are used in the MP1800A/MT1810A.</td>
</tr>
<tr>
<td>&lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
<td>Indicates a numeric value other than a decimal. Hexadecimal and binary values are used in the MP1800A/MT1810A. Hexadecimal values are indicated as #H + {A to F, a to f, 0 to 9}. Binary values are indicated as #B + 0/1.</td>
</tr>
<tr>
<td>&lt;numeric&gt;</td>
<td>Used to transmit 8-bit binary data.</td>
</tr>
<tr>
<td>&lt;NON-DECIMAL NUMERIC PROGRAM DATA&gt;</td>
<td></td>
</tr>
<tr>
<td>&lt;bdata&gt;...etc.</td>
<td></td>
</tr>
<tr>
<td>&lt;ARBITRARY BLOCK PROGRAM DATA&gt;</td>
<td></td>
</tr>
</tbody>
</table>
4.5 Responses

The following table shows the response types used in the MP1800A/MT1810A. In this document, response types are indicated by the lowercase alphanumeric characters enclosed within brackets (< >). The corresponding <RESPONSE DATA> types as defined in IEEE488.2 are indicated in uppercase characters.

<table>
<thead>
<tr>
<th>Response Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;numeric&gt; &lt;NR1 NUMERIC RESPONSE DATA&gt;</td>
<td>Indicates a decimal integer.</td>
</tr>
<tr>
<td>&lt;numeric&gt; &lt;NR2 NUMERIC RESPONSE DATA&gt;</td>
<td>Indicates a decimal fixed-point value.</td>
</tr>
<tr>
<td>&lt;numeric&gt; &lt;NR3 NUMERIC RESPONSE DATA&gt;</td>
<td>Indicates a decimal exponent.</td>
</tr>
<tr>
<td>&lt;numeric&gt; &lt;HEXADECIMAL NUMERIC RESPONSE DATA&gt;</td>
<td>Indicates a hexadecimal numeric value.</td>
</tr>
<tr>
<td>&lt;string&gt; &lt;STRING RESPONSE DATA&gt;</td>
<td>Indicated as a character string enclosed within double quotation marks.</td>
</tr>
<tr>
<td>&lt;bdata&gt;...etc. &lt;DEFINITE LENGTH ARBITRARY BLOCK RESPONSE DATA&gt;</td>
<td>Indicates 8-bit binary data of a known length.</td>
</tr>
</tbody>
</table>
Chapter 5  IEEE488.2 Common Commands

This chapter describes IEEE488.2 Common Commands support to MP1800A and MT1810A.

5.1  IEEE488.2 Common Commands.................................. 5-2
Chapter 5  IEEE488.2 Common Commands

5.1 IEEE488.2 Common Commands

Common commands can be used commonly on any of the GPIB interface and Ethernet interface.

All common commands supported by the MP1800A and MT1810A are sequential commands.

Table 5.1-1 below lists the IEEE488.2 common commands supported by the MP1800A and MT1810A.

<table>
<thead>
<tr>
<th>Mnemonic</th>
<th>Command's full spell</th>
</tr>
</thead>
<tbody>
<tr>
<td>*IDN?</td>
<td>Identification Query</td>
</tr>
<tr>
<td>*RST</td>
<td>Reset Command</td>
</tr>
<tr>
<td>*OPC</td>
<td>Operation Complete Command</td>
</tr>
<tr>
<td>*OPC?</td>
<td>Operation Complete Query</td>
</tr>
<tr>
<td>*WAI</td>
<td>Wait Continue Command</td>
</tr>
<tr>
<td>*CLS</td>
<td>Clear Status Command</td>
</tr>
<tr>
<td>*ESE</td>
<td>Standard Event Status Enable Command</td>
</tr>
<tr>
<td>*ESE?</td>
<td>Standard Event Status Enable Query</td>
</tr>
<tr>
<td>*ESR?</td>
<td>Standard Event Status Register Query</td>
</tr>
<tr>
<td>*SRE</td>
<td>Service Request Enable Command</td>
</tr>
<tr>
<td>*SRE?</td>
<td>Service Request Enable Query</td>
</tr>
<tr>
<td>*STB?</td>
<td>Read Status Byte Query</td>
</tr>
<tr>
<td>*TRG</td>
<td>Trigger Command</td>
</tr>
<tr>
<td>*OPT?</td>
<td>Option Identification Query</td>
</tr>
</tbody>
</table>
5.1 IEEE488.2 Common Commands

*IDN? Identification Query

<table>
<thead>
<tr>
<th>Parameter</th>
<th>None</th>
</tr>
</thead>
<tbody>
<tr>
<td>Response</td>
<td>&lt;Manufacturer&gt;, &lt;Model&gt;, &lt;Serial No.&gt;</td>
</tr>
<tr>
<td>Response</td>
<td>&lt;Manufacturer&gt;, &lt;Model&gt; = &lt;CHARACTER RESPONSE DATA&gt;</td>
</tr>
<tr>
<td>Response</td>
<td>MT1800A, MT1810A</td>
</tr>
<tr>
<td>Response</td>
<td>&lt;Serial No.&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt;</td>
</tr>
<tr>
<td>Response</td>
<td>0000000000 to 9999999999</td>
</tr>
<tr>
<td>Function</td>
<td>Main frame Serial number</td>
</tr>
<tr>
<td>Example</td>
<td>&gt; *IDN?</td>
</tr>
<tr>
<td>Example</td>
<td>&lt; ANRITSU, MP1800A, 0123456789</td>
</tr>
</tbody>
</table>

*RST Reset Command

<table>
<thead>
<tr>
<th>Parameter</th>
<th>None</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function</td>
<td>Initializes entire system.</td>
</tr>
<tr>
<td>Function</td>
<td>Cancels the measurement and clear the contents.</td>
</tr>
<tr>
<td>Function</td>
<td>To reset to the factory default, perform SCPI command</td>
</tr>
<tr>
<td>Example</td>
<td>&lt;SR1&gt;</td>
</tr>
</tbody>
</table>

*OPC Operation Complete Command

<table>
<thead>
<tr>
<th>Parameter</th>
<th>None</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function</td>
<td>Sets bit 0 (operation complete bit) of standard event status register and sets</td>
</tr>
<tr>
<td>Function</td>
<td>SRQ to ON, when execution of preceding command is completed.</td>
</tr>
<tr>
<td>Example</td>
<td>&gt; *OPC</td>
</tr>
</tbody>
</table>

*OPC? Operation Complete Query

<table>
<thead>
<tr>
<th>Parameter</th>
<th>None</th>
</tr>
</thead>
<tbody>
<tr>
<td>Response</td>
<td>&lt;NR1 NUMERIC RESPONSE DATA&gt;</td>
</tr>
<tr>
<td>Response</td>
<td>1</td>
</tr>
<tr>
<td>Function</td>
<td>Returns 1 when preceding command is completed.</td>
</tr>
<tr>
<td>Example</td>
<td>&gt; *OPC?</td>
</tr>
<tr>
<td>Example</td>
<td>&lt; OPC 1</td>
</tr>
</tbody>
</table>
**Chapter 5  IEEE488.2 Common Commands**

### *WAI  
**Wait to Continue Command**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>None</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function</td>
<td>Waits to execute the succeeding command until execution of preceding command is completed. (Executes overlapped commands as sequential commands.) This function is enabled only for preceding command.</td>
</tr>
</tbody>
</table>

**Example**

> *WAI

### *CLS  
**Clear Status Command**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>None</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function</td>
<td>Clear all event registers and queues, except output queue and MAV summary message. Reset of enable registers and transition filters for the device unique status registers is executed using the SCPI command :STATus:PRESet. Both output queue and MAV bits are also cleared when an *CLS is sent immediately after &lt;PROGRAM MESSAGE TERMINATOR&gt; and before &lt;QUERY MESSAGE UNIT&gt; element. Execution of succeeding commands is set to wait until execution of the preceding command is completed.</td>
</tr>
</tbody>
</table>

**Example**

> *CLS
5.1 IEEE488.2 Common Commands

*ESE

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Standard Event Status Enable Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
<td>An integer between 0 and 255</td>
</tr>
<tr>
<td>The parameter represents the total of bit digit values when bits to be enabled are selected from bits of standard event enable register. The digit value for bit to be disabled is set to 0. For the MP1800A and MT1810A, register settings are as listed below: Bit 7 (27 = 128) Power On</td>
<td></td>
</tr>
<tr>
<td>Bit 5 (25 = 32) Command error</td>
<td></td>
</tr>
<tr>
<td>Bit 4 (24 = 16) Execution error</td>
<td></td>
</tr>
<tr>
<td>Bit 3 (23 = 8) Errors other than command, query and execution errors</td>
<td></td>
</tr>
<tr>
<td>Bit 0 (20 = 1) Completion of operation</td>
<td></td>
</tr>
<tr>
<td>Function</td>
<td>Sets or clears standard event status enable register.</td>
</tr>
<tr>
<td>Example</td>
<td>To set bits 3 (= 8) and 4 (= 16) of enable register:</td>
</tr>
<tr>
<td>&gt; *ESE 24</td>
<td></td>
</tr>
</tbody>
</table>

*ESE?

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Standard Event Status Enable Query</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td></td>
</tr>
<tr>
<td>&lt;NR1 NUMERIC RESPONSE DATA&gt;</td>
<td>0 to 255: Total of digit values of standard event status enable register bits. For bit settings of standard status register, refer to the *ESE command.</td>
</tr>
<tr>
<td>Function</td>
<td>Queries current value of standard event status enable register.</td>
</tr>
<tr>
<td>Example</td>
<td>&gt; *ESE?</td>
</tr>
<tr>
<td>&lt; ESE 24</td>
<td></td>
</tr>
</tbody>
</table>

*ESR?

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Standard Event Status Register Query</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td></td>
</tr>
<tr>
<td>&lt;NR1 NUMERIC RESPONSE DATA&gt;</td>
<td>0 to 255: Total of digit values of standard event status register bits. Note: For bit settings of standard status register, refer to the *ESE command.</td>
</tr>
<tr>
<td>Function</td>
<td>Queries current value of standard event status register.</td>
</tr>
<tr>
<td>Example</td>
<td>When a command error exists:</td>
</tr>
<tr>
<td>&gt; *ESR?</td>
<td></td>
</tr>
<tr>
<td>&lt; ESR 32</td>
<td></td>
</tr>
</tbody>
</table>
**Chapter 5  IEEE488.2 Common Commands**

---

### *SRE*  
**Service Request Enable Command**

- **Parameter**
  
  `<DECIMAL NUMERIC PROGRAM DATA>`

  An integer between 0 and 255

  Parameter represents the total of bit digit values when bits to be enabled are selected from bits of service request enable register. The digit value for a bit to be disabled is set to 0.

  For the MP1810A and MT1810A, register settings are as listed below:

  - Bit 7 \((2^7 = 128)\) Operation status register summary
  - Bit 5 \((2^5 = 32)\) Event status register summary
  - Bit 4 \((2^4 = 16)\) Indicates that the output queue is not empty.
  - Bit 3 \((2^3 = 8)\) Questionable status register summary
  - Bit 2 \((2^2 = 4)\) Indicates that the error and event queues are not empty.

- **Function**

  Sets bits of service request enable register.

- **Example**

  To set bit 4 (= 16) of enable register:

  ```
  > *SRE 16
  ```

### *SRE?*  
**Service Request Enable Query**

- **Parameter**

  None

- **Response**

  `<NR1 NUMERIC RESPONSE DATA>`

  0 to 255: Total of digit values of service request enable register bits.

  For bit settings of service request enable register bits, refer to the *SRE command.

  Function Queries current value of service request enable register.

- **Example**

  ```
  > *SRE?
  < SRE 16
  ```

### *STB?*  
**Read Status Byte Query**

- **Parameter**

  None

- **Response**

  `<NR1 NUMERIC RESPONSE DATA>`

  - Bit 7 \((2^7 = 128)\) Operation status register summary
  - Bit 6 \((2^6 = 64)\) MSS (Master Summary Status) summary message
  - Bit 5 \((2^5 = 32)\) Event status register summary
  - Bit 4 \((2^4 = 16)\) Indicates that output queue is not empty.
  - Bit 3 \((2^3 = 8)\) Questionable status register summary
  - Bit 2 \((2^2 = 4)\) Indicates that error and event queues are not empty.

  Function Queries current value of status byte including MSS (Master Summary Status) bit.

- **Example**

  When the event status register summary is true:

  ```
  > *STB?
  < STB 32
  ```
5.1 IEEE488.2 Common Commands

*TRG  Trigger Command
Parameter  None
Function  Operates the same as that of IEEE488.1 GET (Group Execute Trigger bus command). It starts or restarts measurement when the MP1810A and MT1810A receive the *TRG command.
This command is valid only for measurements in the Error and Alarm mode.
Example
> *TRG

*OPT?  Option Identification Query
Parameter  None
Response  <ARBITRARY ASCII RESPONSE DATA>
Characters (refer to Table below) corresponding to the name of an option or module installed.
Function  Reports a list of the installed options/modules (see Table 5.1-2).
All installed options and modules are reported, separated by commas (,).
Example
> *OPT?
< OPT 301, OPT 302

<table>
<thead>
<tr>
<th>Main Frame/Module Name</th>
<th>Option No.</th>
<th>Option Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>MP1800A Signal Quality Analyzer</td>
<td>OPTx01</td>
<td>GPIB</td>
</tr>
<tr>
<td></td>
<td>OPTx02</td>
<td>LAN</td>
</tr>
<tr>
<td></td>
<td>OPT014</td>
<td>2-Slot for PPG and/or ED</td>
</tr>
<tr>
<td></td>
<td>OPT015</td>
<td>4-Slot for PPG and/or ED</td>
</tr>
<tr>
<td></td>
<td>OPT016</td>
<td>6-Slot for PPG and/or ED</td>
</tr>
<tr>
<td>MT1810A 4 slot Chassis</td>
<td>OPT014</td>
<td>2-Slot for PPG and/or ED</td>
</tr>
<tr>
<td></td>
<td>OPT015</td>
<td>4-Slot for PPG and/or ED</td>
</tr>
<tr>
<td>MX180000A Signal Quality Analyzer Control Software</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>MU181000A 12.5GHz Synthesizer</td>
<td>OPTx01</td>
<td>Jitter Modulation</td>
</tr>
<tr>
<td>MU181000B 12.5GHz 4port Synthesizer</td>
<td>OPTx01</td>
<td>Jitter Modulation</td>
</tr>
<tr>
<td>MU181020A 12.5Gbit/s PPG</td>
<td>OPT001</td>
<td>9.8 to 12.5 Gbit/s</td>
</tr>
<tr>
<td></td>
<td>OPT002</td>
<td>0.1 to 12.5 Gbit/s</td>
</tr>
<tr>
<td></td>
<td>OPTx10</td>
<td>Variable Data Output (0.05 to 0.8 Vp-p)</td>
</tr>
<tr>
<td></td>
<td>OPTx11</td>
<td>Variable Data Output (0.25 to 2.5 Vp-p)</td>
</tr>
<tr>
<td></td>
<td>OPTx12</td>
<td>High Performance Data Output (0.05 to 2.0 Vp-p)</td>
</tr>
<tr>
<td></td>
<td>OPTx13</td>
<td>Variable Data Output (0.5 to 3.5 Vp-p)</td>
</tr>
</tbody>
</table>
### Table 5.1-2  Option Character List(MP1800A/MT1810A) (Cont'd)

<table>
<thead>
<tr>
<th>Main Frame/Module Name</th>
<th>Option No.</th>
<th>Option Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>MU181020A 12.5Gbit/s PPG (Cont'd)</td>
<td>OPTx21</td>
<td>Differential Clock Output (0.1 to 2.0 Vp-p)</td>
</tr>
<tr>
<td></td>
<td>OPTx30</td>
<td>Variable Data Delay</td>
</tr>
<tr>
<td>MU181040A 12.5Gbit/s ED</td>
<td>OPT001</td>
<td>9.8 to 12.5 Gbit/s</td>
</tr>
<tr>
<td></td>
<td>OPT002</td>
<td>0.1 to 12.5 Gbit/s</td>
</tr>
<tr>
<td></td>
<td>OPTx20</td>
<td>Clock Recovery</td>
</tr>
<tr>
<td></td>
<td>OPTx30</td>
<td>Variable Clock Delay</td>
</tr>
<tr>
<td>MU181600A Optical Transceiver (XFP)</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>MU181601A Optical Transceiver (SFP)</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>MU181620A Stressed Eye Transmitter</td>
<td>OPTx01</td>
<td>1310nm Reference</td>
</tr>
<tr>
<td></td>
<td>OPTx02</td>
<td>1550nm Reference</td>
</tr>
<tr>
<td></td>
<td>OPTx03</td>
<td>1310nm/1550nm Reference</td>
</tr>
<tr>
<td></td>
<td>OPTx11</td>
<td>1310nm Stressed Eye</td>
</tr>
<tr>
<td></td>
<td>OPTx12</td>
<td>1550nm Stressed Eye</td>
</tr>
<tr>
<td></td>
<td>OPTx13</td>
<td>1310nm/1550nm Stressed Eye</td>
</tr>
<tr>
<td>MU181640A Optical Receiver</td>
<td>OPTx04</td>
<td>Band Width 8.5GHz</td>
</tr>
<tr>
<td>MU181800A 12.5GHz Clock Distributor</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>MU181020B 14Gbit/s PPG</td>
<td>OPT002</td>
<td>0.1 to 14Gbit/s</td>
</tr>
<tr>
<td></td>
<td>OPTx11</td>
<td>Variable Data Output (0.25 to 2.5 Vp-p)</td>
</tr>
<tr>
<td></td>
<td>OPTx12</td>
<td>High Performance Data Output (0.05 to 2.0 Vp-p)</td>
</tr>
<tr>
<td></td>
<td>OPTx13</td>
<td>Variable Data Output (0.5 to 3.5 Vp-p)</td>
</tr>
<tr>
<td></td>
<td>OPTx21</td>
<td>Differential Clock Output (0.1 to 2.0 Vp-p)</td>
</tr>
<tr>
<td></td>
<td>OPTx30</td>
<td>Variable Data Delay</td>
</tr>
<tr>
<td>MU181040B 14Gbit/s ED</td>
<td>OPT002</td>
<td>0.1 to 14 Gbit/s</td>
</tr>
<tr>
<td></td>
<td>OPTx20</td>
<td>Clock Recovery</td>
</tr>
<tr>
<td></td>
<td>OPTx30</td>
<td>Variable Clock Delay</td>
</tr>
<tr>
<td>MU182020A 25Gbit/s 1ch MUX</td>
<td>OPTx01</td>
<td>28Gbit/s Extension</td>
</tr>
<tr>
<td>MU182021A 25Gbit/s 2ch MUX</td>
<td>OPTx02</td>
<td>Clock Input Band Switch</td>
</tr>
<tr>
<td></td>
<td>OPTx10</td>
<td>Variable Data Output (0.25 to 1.75Vp-p)</td>
</tr>
<tr>
<td></td>
<td>OPTx11</td>
<td>Variable Data Output (0.5 to 2.5Vp-p)</td>
</tr>
<tr>
<td></td>
<td>OPTx12</td>
<td>Variable Data Output (0.5 to 2.0Vp-p)</td>
</tr>
<tr>
<td></td>
<td>OPTx13</td>
<td>Variable Data Output (0.5 to 3.5Vp-p)</td>
</tr>
<tr>
<td></td>
<td>OPTx21</td>
<td>Differential Clock Output (0.5 to 2.0Vp-p)</td>
</tr>
<tr>
<td></td>
<td>OPTx30</td>
<td>25Gbit/s Variable Data Delay</td>
</tr>
<tr>
<td></td>
<td>OPTx31</td>
<td>28Gbit/s Variable Data Delay</td>
</tr>
<tr>
<td>MU182040A 25Gbit/s 1ch DEMUX</td>
<td>OPTx01</td>
<td>28Gbit/s Extension</td>
</tr>
<tr>
<td>MU182041A 25Gbit/s 2ch DEMUX</td>
<td>OPTx02</td>
<td>Clock Input Band Switch</td>
</tr>
<tr>
<td></td>
<td>OPTx30</td>
<td>25GHz Variable Clock Delay</td>
</tr>
<tr>
<td></td>
<td>OPTx31</td>
<td>28GHz Variable Clock Delay</td>
</tr>
</tbody>
</table>
Chapter 6  Status Report

This chapter describes Status Register configuration and bit definition of Status Register. For the MP1800A and MT1810A, installed Status Register is as below:

IEEE488.2 Regulated Register  ...... Standard Event Register
Status Byte Register
SCPI Regulated Register  ...... OPERational Status Register
Device-unique Status Registers  ...... Device-Unique Status Register
(However, separated from Regulated Register)

6.1  Overview ......................................................6-2
6.2  IEEE488.2 Regulated Register ..............................6-3
6.3  SCPI-Regulated Status Register ...............................6-6
6.4  How to Read and Write Status Register ......................6-8
6.5  How to Clear and Reset Status Register .....................6-9
6.6  Device-Unique Status ........................................6-10
6.7  Status Commands ..............................................6-13
  6.7.1  Status Preset .............................................6-13
  6.7.2  Operation Status Register ...............................6-14
  6.7.3  Synthesizer Status ........................................6-18
  6.7.4  ED Status ................................................6-20
  6.7.5  PPG Status ................................................6-23
  6.7.6  Optical Transceiver (XFP) Status .........................6-25
  6.7.7  Optical Transceiver (SFP) Status .........................6-28
  6.7.8  DEMUX Status ............................................6-30
  6.7.9  MUX Status ...............................................6-33
6.1 Overview

For MP1800A and MT1810A, Status Register Configuration is as below.

Figure 6.1-1 Status Register Configuration
6.2 IEEE488.2 Regulated Register

The bits of the following two IEEE488.2 regulated registers are defined as follows.

Table 6.2-1  IEEE488.2 Regulated Register

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Status byte register</td>
<td>This register can set RQS and 7 summary message bits. It is used as a pair with the service request enable register. When OR of both registers is not 0, SRQ turns ON. RQS is programmed in bit 6. This bit is used to report to the remote control PC that a service request is given.</td>
</tr>
<tr>
<td>Standard event status register</td>
<td>Stores 8 events which the device encounters as the standard events. The logical OR output bit is summarized and displayed in bit 5 of the status byte register as an ESB (Event Status Bit) summary message.</td>
</tr>
</tbody>
</table>
6.2 IEEE488.2 Regulated Register

### Table 6.2-2 Status Byte Register Bit Definition

<table>
<thead>
<tr>
<th>Bit</th>
<th>Mnemonic</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB2</td>
<td>QUE (error/event QUEue)</td>
<td>Indicates that the error and event queues not empty.</td>
</tr>
<tr>
<td>DB3</td>
<td>QUES (QUEStionable status register summary)</td>
<td>QUEStionable status register summary.</td>
</tr>
<tr>
<td>DB4</td>
<td>MAV (Message AVailable)</td>
<td>Indicates that the output queue is not empty. When the device is ready to receive the response message send-out request from the controller, the MAV summary message bit is set to 1 (TRUE). This message can be used to allow the controller to send the queue command to the device and wait until MAV turns TRUE.</td>
</tr>
<tr>
<td>DB5</td>
<td>ESB (Event Summary Bit)</td>
<td>Standard event status register summary</td>
</tr>
<tr>
<td>DB6</td>
<td>RQS (ReQuest Service)</td>
<td>Returns a 7-bit status byte and this RQS to the controller in the serial pole mode.</td>
</tr>
<tr>
<td></td>
<td>MSS (Master Summary Status)</td>
<td>Indicates that the MP1800A/MT1810A has reason to request at least one service. When inquiring the status byte by the *STB? command, this MSS summary message appears in bit 6 in place of the RQS message.</td>
</tr>
<tr>
<td>DB7</td>
<td>OPER (OPERation status register summary)</td>
<td>OPERation status register summary</td>
</tr>
</tbody>
</table>

### Table 6.2-3 Standard Event Status Register Bit Definition

<table>
<thead>
<tr>
<th>Bit</th>
<th>Mnemonic</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB0</td>
<td>OPC (OPeration Complete)</td>
<td>Indicates that all the specified operations are completed.</td>
</tr>
<tr>
<td>DB3</td>
<td>DDE (Device-Dependent Error)</td>
<td>Indicates that an error other than command error, query error or execution error occurs.</td>
</tr>
<tr>
<td>DB4</td>
<td>EXE (EXecution Error)</td>
<td>Indicates that an execution error occurs.</td>
</tr>
<tr>
<td>DB5</td>
<td>CME (CoMmand Error)</td>
<td>Indicates that a command error occurs.</td>
</tr>
<tr>
<td>DB7</td>
<td>PON (Power ON)</td>
<td>Indicates that the power supply turns from OFF to ON.</td>
</tr>
</tbody>
</table>

**Note:**

See Chapter 5 "IEEE488.2 Common Commands" for the setting and query commands for the Status Byte Register and Standard Event Status Registers.
6.3 SCPI-Regulated Status Register

SPCI regulates that the Instrument should contain the following registers in addition to those regulated in IEEE488.2.

**Table 6.3-1 SCPI-regulated Status Registers**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>QUESTIONable Status register</td>
<td>Reports a signal status such as a measurement result. This register is used to send a service request to an external controller when an error occurs. Note that the MP1800A and MT1810A do not use this register.</td>
</tr>
<tr>
<td>OPERation Status register</td>
<td>Reports some MP1800A/MT1810A statuses.</td>
</tr>
</tbody>
</table>

**Figure 6.3-1 OPERation Status Register**

**Table 6.3-2 OPERation Status Register Bit Definition**

<table>
<thead>
<tr>
<th>Bit</th>
<th>Mnemonic</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB3</td>
<td>BAT (BAThtub measuring)</td>
<td>Indicates that the Instrument is measuring Bathtub.</td>
</tr>
<tr>
<td>DB4</td>
<td>MEAS (MEASuring)</td>
<td>Indicates that the Instrument is measuring when any port is in execution.*</td>
</tr>
<tr>
<td>DB5</td>
<td>ISI (ISI)</td>
<td>Indicates that the Instrument is measuring ISI.</td>
</tr>
<tr>
<td>DB6</td>
<td>QAN (Q ANalysis)</td>
<td>Indicates that the Instrument is measuring Q Analysis</td>
</tr>
<tr>
<td>DB7</td>
<td>AAD (Auto ADjust)</td>
<td>Indicates that the Instrument is in Auto Adjust.</td>
</tr>
<tr>
<td>DB8</td>
<td>ASE (Auto SEarching)</td>
<td>Indicates that the Instrument is in Auto Search.</td>
</tr>
<tr>
<td>DB9</td>
<td>EMM (Eye Margin Measuring)</td>
<td>Indicates that the Instrument is measuring eye margin.</td>
</tr>
</tbody>
</table>
### Table 6.3-2 OPERation Status Register Bit Definition (Cont'd)

<table>
<thead>
<tr>
<th>Bit</th>
<th>Mnemonic</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB10</td>
<td>EDI (Eye Diagram)</td>
<td>Indicates that the Instrument is measuring eye diagram.</td>
</tr>
<tr>
<td>DB11</td>
<td>PSET (Pattern SETing)</td>
<td>Indicates that the Instrument is setting pattern.</td>
</tr>
<tr>
<td>DB12</td>
<td>INI (INITialize)</td>
<td>Indicates that the Instrument is in initialization.</td>
</tr>
</tbody>
</table>

*: When remeasurement is performed during measurement, DB4 becomes OFF and then ON again. DB4 bit becomes on at the same time when measurement is started.
6.4 How to Read and Write Status Register

This section describes how to read and write the status register data.

<table>
<thead>
<tr>
<th>Register</th>
<th>Reading</th>
<th>Writing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Status byte register</td>
<td>Read the register bits using the serial pole. A 7-bit status byte and</td>
<td>Disabled</td>
</tr>
<tr>
<td></td>
<td>a RQS message bit are returned. The status byte value does not change.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>*STB? Common query</td>
<td></td>
</tr>
<tr>
<td></td>
<td>A numeric value composed of the status byte register value and the MSS</td>
<td></td>
</tr>
<tr>
<td></td>
<td>summary message is returned.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>The Status register does not change at this time.</td>
<td></td>
</tr>
<tr>
<td>Service request Enable register</td>
<td>*SRE? common query</td>
<td>*SRE common command</td>
</tr>
<tr>
<td></td>
<td>The register bits do not change.</td>
<td></td>
</tr>
<tr>
<td>Standard status register</td>
<td>*ESR? common query</td>
<td>Disabled</td>
</tr>
<tr>
<td></td>
<td>The register bits are cleared after being read.</td>
<td></td>
</tr>
<tr>
<td>Standard event status enable</td>
<td>*ESE? common query</td>
<td>*ESE common command</td>
</tr>
<tr>
<td>register</td>
<td>The register bits do not change.</td>
<td></td>
</tr>
<tr>
<td>SCPI event register</td>
<td>:STATus:...:EVENt?</td>
<td>Disabled</td>
</tr>
<tr>
<td></td>
<td>Device-unique command</td>
<td></td>
</tr>
<tr>
<td></td>
<td>The register bits are cleared.</td>
<td></td>
</tr>
<tr>
<td>SCPI enable register</td>
<td>:STATus:...:ENABle?</td>
<td>:STATus:...:ENABle</td>
</tr>
<tr>
<td></td>
<td>The contents of the register do not change.</td>
<td></td>
</tr>
<tr>
<td>SCPI Transition filter</td>
<td>:STATus:...:PTRansition?</td>
<td>Device-unique command</td>
</tr>
<tr>
<td></td>
<td>:STATus:...:NTRansition?</td>
<td>:STATus:...:PTRansition</td>
</tr>
<tr>
<td></td>
<td>Device-unique command</td>
<td>:STATus:...:NTRansition</td>
</tr>
<tr>
<td></td>
<td>The contents of register bits do not change.</td>
<td></td>
</tr>
<tr>
<td>Error/event queue</td>
<td>:SYSTem:ERRor?</td>
<td>Disabled</td>
</tr>
</tbody>
</table>

Note:

The SCPI event register, SCPI enable register, and SCPI Transition filter listed above indicate the SCPI-regulated status registers and an event or a transition filter in a device-unique status.
6.5 How to Clear and Reset Status Register

This section describes how to clear and reset the status registers.

Table 6.5-1  Clearing and Resetting the Status Registers

<table>
<thead>
<tr>
<th>Register</th>
<th>*RST</th>
<th>*CLS</th>
<th>Power ON</th>
<th>STATus: PRESet</th>
<th>Other method to clear register</th>
</tr>
</thead>
<tbody>
<tr>
<td>Status byte register</td>
<td>No change</td>
<td>Clear</td>
<td>Clear</td>
<td>No change</td>
<td></td>
</tr>
<tr>
<td>Service request enable register</td>
<td>No change</td>
<td>No change</td>
<td>Clear</td>
<td>No change</td>
<td>Executing *SRE 0</td>
</tr>
<tr>
<td>Standard event status register</td>
<td>No change</td>
<td>Clear</td>
<td>Clear</td>
<td>No change</td>
<td>Cleared when an event is read by *ESR?</td>
</tr>
<tr>
<td>Standard event status enable register</td>
<td>No change</td>
<td>No change</td>
<td>Clear*1</td>
<td>No change</td>
<td>Executing *ESE 0</td>
</tr>
<tr>
<td>SCPI event register</td>
<td>No change</td>
<td>Clear</td>
<td>Clear</td>
<td>No change</td>
<td>Cleared when an event is read by :STATus:...:EVENt?</td>
</tr>
<tr>
<td>SCPI enable register</td>
<td>No change</td>
<td>No change</td>
<td>Reset*1</td>
<td>Reset</td>
<td>Executing :STATus:...:ENABle 0</td>
</tr>
<tr>
<td>SCPI Transition filter</td>
<td>No change</td>
<td>Reset</td>
<td>Reset</td>
<td>Reset</td>
<td>Executing :STATus:...:PTRansition 0 and :STATus:...:NTRansition 0</td>
</tr>
<tr>
<td>Error/event queue</td>
<td>No change</td>
<td>Clear</td>
<td>Clear</td>
<td>No change</td>
<td>Reading all events by :SYSTem:ERRor?</td>
</tr>
</tbody>
</table>

*1: When power on as PSC (Power-ON Status Clear) flag is true, it will be cleared (or reset).

*2: To be 128 bits.

Note:
The SCPI event register, SCPI enable register, and SCPI Transition filter listed above indicate the SCPI-regulated status registers.

The following shows the reset values of the registers influenced by the :STATus:PRESet command.

Table 6.5-2  Values Reset by the :STATus:PRESet Command

<table>
<thead>
<tr>
<th>Register</th>
<th>Enable/filter</th>
<th>Reset value</th>
</tr>
</thead>
<tbody>
<tr>
<td>OPERational status register</td>
<td>Enable register</td>
<td>All 0</td>
</tr>
<tr>
<td></td>
<td>PTRansition filter</td>
<td>All 1</td>
</tr>
<tr>
<td></td>
<td>NTRansition filter</td>
<td>All 0</td>
</tr>
</tbody>
</table>
6.6 Device-Unique Status

Signal Quality Analyzer series instruments support each module status as device-unique status. Device-unique Status Configuration is as below.

- **Condition Register** Monitors the device status and changes real time in response to the device status. Thus, this register does not memorize the status.
- **Transition Filter** Sets the Condition Register data in the Event Register. The following three types of transition filters are available depending on which change of the Condition Register is to be evaluated.
  - Positive direction change: The event becomes true only when the corresponding condition changes from false to true.
  - Negative direction change: The event becomes true only when the corresponding condition changes from true to false.
  - Bi-directional change: The event becomes true when a change arise either in the positive or negative direction.
- **Event Register** Memorizes output from Transition Filter.

![Diagram](image.png)

**Figure 6.6-1 Each Status Configuration**
The following show module status bit definition.

### Table 6.6-1  Module Status

<table>
<thead>
<tr>
<th>Status</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>Synthesizer Status</td>
<td>Reports MU181000A 12.5GHz Synthesizer status and MU181000B 12.5GHz 4port Synthesizer status.</td>
</tr>
<tr>
<td>ED Status</td>
<td>Reports MU181040A 12.5Gbit/s ED status and MU181040B 14Gbit/s ED status.</td>
</tr>
<tr>
<td>Optical Transceiver (XFP) Status</td>
<td>Reports MU181600A Optical Transceiver (XFP) status.</td>
</tr>
<tr>
<td>Optical Transceiver (SFP) Status</td>
<td>Reports MU181601A Optical Transceiver (SFP) status.</td>
</tr>
<tr>
<td>MUX Status</td>
<td>MU182020A 25 Gbit/s 1Ch MUX/ MU182021A 25 Gbit/s 2Ch MUX Status Report</td>
</tr>
<tr>
<td>DEMUX Status</td>
<td>MU182040A 25Gbit/s 1Ch DEMUX/ MU182041A 25Gbit/s 2Ch DEMUX Status Report</td>
</tr>
</tbody>
</table>

### Table 6.6-2  Synthesizer Status Bit Definition

<table>
<thead>
<tr>
<th>BIT</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB0</td>
<td>Indicates PLL Unlock occurred.</td>
</tr>
</tbody>
</table>

### Table 6.6-3  ED Status Bit Definition

<table>
<thead>
<tr>
<th>BIT</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB0</td>
<td>Indicates Insertion Error was detected.</td>
</tr>
<tr>
<td>DB1</td>
<td>Indicates Omission Error was detected.</td>
</tr>
<tr>
<td>DB2</td>
<td>Indicates Total Error was detected.</td>
</tr>
<tr>
<td>DB4</td>
<td>Indicates Pattern Sync Loss occurred.</td>
</tr>
<tr>
<td>DB5</td>
<td>Indicates Clock Loss occurred.</td>
</tr>
<tr>
<td>DB6</td>
<td>Indicates Delay Busy occurred.</td>
</tr>
<tr>
<td>DB8</td>
<td>Indicates Transition Error was detected.</td>
</tr>
<tr>
<td>DB9</td>
<td>Indicates Non Transition Error was detected.</td>
</tr>
<tr>
<td>DB10</td>
<td>Indicates CR Unlock occurred.</td>
</tr>
<tr>
<td>DB11</td>
<td>Indicates Delay Calibration Require occurred.</td>
</tr>
<tr>
<td>BIT</td>
<td>Contents</td>
</tr>
<tr>
<td>------</td>
<td>----------------------------------------------</td>
</tr>
<tr>
<td>DB1</td>
<td>Indicates CMU-MUX Unlock occurred.</td>
</tr>
<tr>
<td>DB2</td>
<td>Indicates Delay Busy occurred.</td>
</tr>
<tr>
<td>DB4</td>
<td>Indicates Delay Calibration Require occurred.</td>
</tr>
</tbody>
</table>

Table 6.6-5 Optical Transceiver (XFP) Status Bit Definition

<table>
<thead>
<tr>
<th>BIT</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB0</td>
<td>Indicates Ready status.</td>
</tr>
<tr>
<td>DB1</td>
<td>Indicates Laser Fault occurred.</td>
</tr>
<tr>
<td>DB2</td>
<td>Indicates Unlock occurred.</td>
</tr>
<tr>
<td>DB3</td>
<td>Indicates LOS occurred.</td>
</tr>
<tr>
<td>DB4</td>
<td>Indicates CDR Unlock occurred.</td>
</tr>
</tbody>
</table>

Table 6.6-6 Optical Transceiver (SFP) Status Bit Definition

<table>
<thead>
<tr>
<th>BIT</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB0</td>
<td>Indicates Ready status.</td>
</tr>
<tr>
<td>DB2</td>
<td>Indicates LOS occurred.</td>
</tr>
</tbody>
</table>

Table 6.6-7 DEMUX Status Bit Definition

<table>
<thead>
<tr>
<th>BIT</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB6</td>
<td>Displays Delay Busy occurrences at Data1</td>
</tr>
<tr>
<td>DB7</td>
<td>Displays Delay Busy occurrences at Data2</td>
</tr>
<tr>
<td>DB11</td>
<td>Displays Delay Calibration Require occurrences at Data1</td>
</tr>
<tr>
<td>DB12</td>
<td>Displays Delay Calibration Require occurrences at Data2</td>
</tr>
</tbody>
</table>

Table 6.6-8 MUX Status Bit Definition

<table>
<thead>
<tr>
<th>BIT</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB2</td>
<td>Displays Delay Busy occurrences at Data1</td>
</tr>
<tr>
<td>DB3</td>
<td>Displays Delay Busy occurrences at Data2</td>
</tr>
<tr>
<td>DB4</td>
<td>Displays Delay Calibration Require occurrences at Data1</td>
</tr>
<tr>
<td>DB5</td>
<td>Displays Delay Calibration Require occurrences at Data2</td>
</tr>
</tbody>
</table>
6.7 Status Commands

This section explains about OPERational Status Register and each module-unique status commands.

When reading and writing each module-unique status, the following three commands must be transmitted before transmitting a status command.

1. :UENTry:ID <unit_number>
2. :MODule:ID <module_number>
3. :PORT:ID <port_number>

These three commands identify the operating module, and any commands sent/received after them operate for the identified module. However, :PORT:ID <port_number> can be omitted.

6.7.1 Status Preset

The following command initializes the Enable Register and filter of the OPERational Status Register.

:STATus:PRESet

Function: Initializes the event status register and filter.
Example: > :STATus:PRESet
6.7.2 Operation Status Register

The Operation Status Register is used to indicate an operation status such as "measurement in progress".

**:STATus:OPERation[:EVENT]?**

Response

<table>
<thead>
<tr>
<th>&lt;numeric&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 to 8184</td>
</tr>
<tr>
<td>Sum total of the bit of event register</td>
</tr>
<tr>
<td>(DECIMAL)</td>
</tr>
<tr>
<td>Available bit</td>
</tr>
<tr>
<td>8 (Bit 3)</td>
</tr>
<tr>
<td>Bathtub measurement in progress</td>
</tr>
<tr>
<td>16 (Bit 4)</td>
</tr>
<tr>
<td>Measurement in progress</td>
</tr>
<tr>
<td>32 (Bit 5)</td>
</tr>
<tr>
<td>ISI measurement in progress</td>
</tr>
<tr>
<td>64 (Bit 6)</td>
</tr>
<tr>
<td>Q Analysis measurement in progress</td>
</tr>
<tr>
<td>128 (Bit 7)</td>
</tr>
<tr>
<td>Auto Adjust in progress</td>
</tr>
<tr>
<td>256 (Bit 8)</td>
</tr>
<tr>
<td>Auto Search in progress</td>
</tr>
<tr>
<td>512 (Bit 9)</td>
</tr>
<tr>
<td>Eye Margin measurement in progress</td>
</tr>
<tr>
<td>1024 (Bit 10)</td>
</tr>
<tr>
<td>Eye Diagram measurement in progress</td>
</tr>
<tr>
<td>2048 (Bit 11)</td>
</tr>
<tr>
<td>Pattern loading in progress</td>
</tr>
<tr>
<td>4096 (Bit 12)</td>
</tr>
<tr>
<td>Initialization in progress</td>
</tr>
</tbody>
</table>

Function

Queries event register at OPERation Status Register.

Example

```
> :STATus:OPERation:EVENT?
```

or

```
> :STATus:OPERation?
```

< 16

**:STATus:OPERation:CONDition?**

Response

<table>
<thead>
<tr>
<th>&lt;numeric&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 to 8184</td>
</tr>
<tr>
<td>Sum total of the bit of condition register (DECIMAL)</td>
</tr>
<tr>
<td>Available bit</td>
</tr>
<tr>
<td>8 (Bit 3)</td>
</tr>
<tr>
<td>Bathtub measurement in progress</td>
</tr>
<tr>
<td>16 (Bit 4)</td>
</tr>
<tr>
<td>Measurement in progress</td>
</tr>
<tr>
<td>32 (Bit 5)</td>
</tr>
<tr>
<td>ISI measurement in progress</td>
</tr>
<tr>
<td>64 (Bit 6)</td>
</tr>
<tr>
<td>Q Analysis measurement in progress</td>
</tr>
<tr>
<td>128 (Bit 7)</td>
</tr>
<tr>
<td>Auto Adjust in progress</td>
</tr>
<tr>
<td>256 (Bit 8)</td>
</tr>
<tr>
<td>Auto Search in progress</td>
</tr>
<tr>
<td>512 (Bit 9)</td>
</tr>
<tr>
<td>Eye Margin measurement in progress</td>
</tr>
<tr>
<td>1024 (Bit 10)</td>
</tr>
<tr>
<td>Eye Diagram measurement in progress</td>
</tr>
<tr>
<td>2048 (Bit 11)</td>
</tr>
<tr>
<td>Pattern loading in progress</td>
</tr>
<tr>
<td>4096 (Bit 12)</td>
</tr>
<tr>
<td>Initialization in progress</td>
</tr>
</tbody>
</table>

Function

Queries condition register at OPERation Status Register.

Example

```
> :STATus:OPERation:CONDition?
```

< 16
### Status Commands

#### :STATus:OPERation:ENABle <numeric>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;numeric&gt;</td>
<td>&lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
</tr>
<tr>
<td>0 to 8184</td>
<td>Sum total of the bit of event enable register (DECIMAL)</td>
</tr>
<tr>
<td>Available bit</td>
<td></td>
</tr>
<tr>
<td>8 (Bit 3)</td>
<td>Bathtub measurement in progress</td>
</tr>
<tr>
<td>16 (Bit 4)</td>
<td>Measurement in progress</td>
</tr>
<tr>
<td>32 (Bit 5)</td>
<td>ISI measurement in progress</td>
</tr>
<tr>
<td>64 (Bit 6)</td>
<td>Q Analysis measurement in progress</td>
</tr>
<tr>
<td>128 (Bit 7)</td>
<td>Auto Adjust in progress</td>
</tr>
<tr>
<td>256 (Bit 8)</td>
<td>Auto Search in progress</td>
</tr>
<tr>
<td>512 (Bit 9)</td>
<td>Eye Margin measurement in progress</td>
</tr>
<tr>
<td>1024 (Bit 10)</td>
<td>Eye Diagram measurement in progress</td>
</tr>
<tr>
<td>2048 (Bit 11)</td>
<td>Pattern load in progress</td>
</tr>
<tr>
<td>4096 (Bit 12)</td>
<td>Initialization in progress</td>
</tr>
</tbody>
</table>

Sets parameter to 0, masks of all bits.

**Function**
Sets mask value of event enable register at OPERation status register.

**Example**
To set event enable register to 16 at OPERation status register.

> :STATus:OPERation:ENABle 16

#### :STATus:OPERation:ENABle?

<table>
<thead>
<tr>
<th>Response</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;numeric&gt;</td>
<td>&lt;NR1 NUMERIC RESPONSE DATA&gt;</td>
</tr>
<tr>
<td>0 to 8184</td>
<td>Sum total of the bit of event enable register (DECIMAL)</td>
</tr>
</tbody>
</table>

**Function**
Queries enable register at OPERation status register.

**Example**
> :STATus:OPERation:ENABle?
< 16
:STATus:OPERation:PTRansition <numeric>

Parameter:  
<numeric> = <DECIMAL NUMERIC PROGRAM DATA>
0 to 8184  Sum total of the bit of transition filter (DECIMAL)

Available bit
8 (Bit 3)  Bathtub measurement in progress
16 (Bit 4) Measurement in progress
32 (Bit 5) ISI measurement in progress
64 (Bit 6) Q Analysis measurement in progress
128 (Bit 7) Auto Adjust in progress
256 (Bit 8) Auto Search in progress
512 (Bit 9) Eye Margin measurement in progress
1024 (Bit 10) Eye Diagram measurement in progress
2048 (Bit 11) Pattern load in progress
4096 (Bit 12) Initialization in progress

Function:  
Sets the transition filter (positive direction change) of the OPERation status register.

Example:  
To set the transition filter (positive direction change) of the OPERation status register to 16.
> :STATus:OPERation:PTRansition 16

:STATus:OPERation:PTRansition?

Response:  
<numeric> = <NR1 NUMERIC RESPONSE DATA>
0 to 8184  Sum total of the bit of transition filter (DECIMAL)

Function:  
Queries the transition filter (positive direction change) of the OPERation status register.

Example:  
> :STATus:OPERation:PTRansition?
< 16
6.7 Status Commands

:STATus:OPERation:NTRansition <numeric>

Parameter  
<numeric> = <DECIMAL NUMERIC PROGRAM DATA>  
0 to 8184  Sum total of the bit of transition filter  
           (DECIMAL)  

Available bit  
8 (Bit 3)  Bathtub measurement in progress  
16 (Bit 4)  Measurement in progress  
32 (Bit 5)  ISI measurement in progress  
64 (Bit 6)  Q Analysis measurement in progress  
128 (Bit 7)  Auto Adjust in progress  
256 (Bit 8)  Auto Search in progress  
512 (Bit 9)  Eye Margin measurement in progress  
1024 (Bit 10)  Eye Diagram measurement in progress  
2048 (Bit 11)  Pattern load in progress  
4096 (Bit 12)  Initialization in progress  

Function Sets the transition filter (negative direction change) of the OPERation status register.  

Example  
To set the transition filter (negative direction change) of the OPERation status register to 16.  

> :STATus:OPERation:NTRansition 16

:STATus:OPERation:NTRansition?

Response  
<numeric> = <NR1 NUMERIC RESPONSE DATA>  
0 to 8184  Sum total of the bit of transition filter  
           (DECIMAL)  

Function Queries the transition filter (negative direction change) of the OPERation status register.  

Example  
> :STATus:OPERation:NTRansition?  
< 16
### 6.7.3 Synthesizer Status

Synthesizer Status displays faults at the MU181000A 12.5 GHz Synthesizer and MU181000B 12.5 GHz 4 Port Synthesizer.

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>:INSTrument:SYG125[:EVENt]?</td>
<td>Queries event at 12.5 GHz Synthesizer.</td>
</tr>
</tbody>
</table>
| Example | > :INSTrument:SYG125:EVENt?  
or  
> :INSTrument:SYG125?  
< 1 |

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
</table>
| Example | > :INSTrument:SYG125:CONDition?  
< 1 |

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>:INSTrument:SYG125:PTRansition &lt;numeric&gt;</td>
<td>Sets the transition filter (positive direction change) of the 12.5 GHz Synthesizer Status.</td>
</tr>
</tbody>
</table>
| Example | To set the transition filter (positive direction change) of the 12.5 GHz Synthesizer Status to 1.  
> :INSTrument:SYG125:PTRansition 1 |
6.7 Status Commands

:INSTrument:SYG125:PTRansition?

Response

<nemonic> = <NR1 NUMERIC RESPONSE DATA>
0 to 1   Sum total of the bit of transition filter
         (DECIMAL)

Function

Queries the transition filter (positive direction change) of the 12.5 GHz Synthesizer Status.

Example

> :INSTrument:SYG125:PTRansition?
< 1

:INSTrument:SYG125:NTRansition <numeric>

Parameter

<nemonic> = <DECIMAL NUMERIC PROGRAM DATA>
0 to 1   Sum total of the bit of transition filter
         (DECIMAL)

Available bit
1 (Bit 0)   PLL Unlock occurs

Function

Sets the transition filter (negative direction change) of the 12.5 GHz Synthesizer Status.

Example

To set the transition filter (negative direction change) of the 12.5 GHz Synthesizer Status to 1.
> :INSTrument:SYG125:NTRansition 1

:INSTrument:SYG125:NTRtransition?

Response

<nemonic> = <NR1 NUMERIC RESPONSE DATA>
0 to 1   Sum total of the bit of transition filter
         (DECIMAL)

Function

Queries the transition filter (negative direction change) of the 12.5 GHz Synthesizer Status.

Example

> :INSTrument:SYG125:NTRexercise?
< 1

:INSTrument:SYG125:RESet

Function

Initializes event at 12.5 GHz Synthesizer.

Example

> :INSTrument:SYG125:RESet
Chapter 6  Status Report

6.7.4 ED Status

The ED Status is used to indicate an alarm and error of the MU181040A 12.5 Gbit/s ED and MU181040B 14 Gbit/s ED.

:INSTrument:EDG125[:EVENt]?

<table>
<thead>
<tr>
<th>Response</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;numeric&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt;</td>
<td>Sum total of the event bit (DECIMAL)</td>
</tr>
<tr>
<td>0 to 3959</td>
<td>Sum total of the event bit (DECIMAL)</td>
</tr>
<tr>
<td>Available bit</td>
<td></td>
</tr>
<tr>
<td>1 (Bit 0)</td>
<td>Insertion Error occurs</td>
</tr>
<tr>
<td>2 (Bit 1)</td>
<td>Omission Error occurs</td>
</tr>
<tr>
<td>4 (Bit 2)</td>
<td>Total Error occurs</td>
</tr>
<tr>
<td>16 (Bit 4)</td>
<td>Pattern Sync Loss occurs</td>
</tr>
<tr>
<td>32 (Bit 5)</td>
<td>Clock Loss occurs</td>
</tr>
<tr>
<td>64 (Bit 6)</td>
<td>Delay Busy occurs</td>
</tr>
<tr>
<td>256 (Bit 8)</td>
<td>Transition Error occurs</td>
</tr>
<tr>
<td>512 (Bit 9)</td>
<td>Non Transition Error occurs</td>
</tr>
<tr>
<td>1024 (Bit 10)</td>
<td>CR Unlock occurs</td>
</tr>
<tr>
<td>2048 (Bit 11)</td>
<td>Delay Calibration Require occurs</td>
</tr>
</tbody>
</table>

Function: Queries event at 12.5 Gbit/s ED status.
Example: > :INSTrument:EDG125:EVENt?
          or
          > :INSTrument:EDG125?
          < 1

:INSTrument:EDG125:CONDition?

<table>
<thead>
<tr>
<th>Response</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;numeric&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt;</td>
<td>Sum total of the condition bit (DECIMAL)</td>
</tr>
<tr>
<td>0 to 3959</td>
<td>Sum total of the condition bit (DECIMAL)</td>
</tr>
<tr>
<td>Available bit</td>
<td></td>
</tr>
<tr>
<td>1 (Bit 0)</td>
<td>Insertion Error occurs</td>
</tr>
<tr>
<td>2 (Bit 1)</td>
<td>Omission Error occurs</td>
</tr>
<tr>
<td>4 (Bit 2)</td>
<td>Total Error occurs</td>
</tr>
<tr>
<td>16 (Bit 4)</td>
<td>Pattern Sync Loss occurs</td>
</tr>
<tr>
<td>32 (Bit 5)</td>
<td>Clock Loss occurs</td>
</tr>
<tr>
<td>64 (Bit 6)</td>
<td>Delay Busy occurs</td>
</tr>
<tr>
<td>256 (Bit 8)</td>
<td>Transition Error occurs</td>
</tr>
<tr>
<td>512 (Bit 9)</td>
<td>Non Transition Error occurs</td>
</tr>
<tr>
<td>1024 (Bit 10)</td>
<td>CR Unlock occurs</td>
</tr>
<tr>
<td>2048 (Bit 11)</td>
<td>Delay Calibration Require occurs</td>
</tr>
</tbody>
</table>

Function: Queries condition at 12.5 Gbit/s ED status.
Example: > :INSTrument:EDG125:CONDition?
          < 1
6.7 Status Commands

:INSTrument:EDG125:PTRansition <numeric>

Parameter

<table>
<thead>
<tr>
<th>&lt;numeric&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 to 3959 Sum total of the bit of transition filter (DECIMAL)</td>
</tr>
</tbody>
</table>

Available bit

| 1 (Bit 0) Insertion Error occurs |
| 2 (Bit 1) Omission Error occurs |
| 4 (Bit 2) Total Error occurs |
| 16 (Bit 4) Pattern Sync Loss occurs |
| 32 (Bit 5) Clock Loss occurs |
| 64 (Bit 6) Delay Busy occurs |
| 256 (Bit 8) Transition Error occurs |
| 512 (Bit 9) Non Transition Error occurs |
| 1024 (Bit 10) CR Unlock occurs |
| 2048 (Bit 11) Delay Calibration Require occurs |

Function
Sets the transition filter (positive direction change) of the 12.5 Gbit/s ED Status.

Example
To set the transition filter (positive direction change) of the 12.5 Gbit/s ED Status to 1.

> :INSTrument:EDG125:PTRansition 1

:INSTrument:EDG125:PTRansition?

Response

<table>
<thead>
<tr>
<th>&lt;numeric&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 to 3959 Sum total of the bit of transition filter (DECIMAL)</td>
</tr>
</tbody>
</table>

Function
Queries the transition filter (positive direction change) of the 12.5 Gbit/s ED Status.

Example

> :INSTrument:EDG125:PTRansition?
< 1
**Chapter 6  Status Report**

### :INSTrument:EDG125:NTRansition <numeric>:

**Parameter**

<numeric> = <DECIMAL NUMERIC PROGRAM DATA>

- 0 to 3959: Sum total of the bit of transition filter (DECIMAL)

Available bit

1 (Bit 0): Insertion Error occurs
2 (Bit 1): Omission Error occurs
4 (Bit 2): Total Error occurs
16 (Bit 4): Pattern Sync Loss occurs
32 (Bit 5): Clock Loss occurs
64 (Bit 6): Delay Busy occurs
256 (Bit 8): Transition Error occurs
512 (Bit 9): Non Transition Error occurs
1024 (Bit 10): CR Unlock occurs
2048 (Bit 11): Delay Calibration Require occurs

**Function**

Sets the transition filter (negative direction change) of the 12.5 Gbit/s ED Status.

**Example**

To set the transition filter (negative direction change) of the 12.5 Gbit/s ED Status to 1.

> :INSTrument:EDG125:NTRansition 1

### :INSTrument:EDG125:NTRansition?:

**Response**

<numeric> = <NR1 NUMERIC RESPONSE DATA>

- 0 to 3959: Sum total of the bit transition filter (DECIMAL)

**Function**

Queries the transition filter (negative direction change) of the 12.5 Gbit/s ED Status.

**Example**

> :INSTrument:EDG125:NTRansition?

< 1

### :INSTrument:EDG125:RESet

**Function**

Initializes event at 12.5 Gbit/s ED status.

**Example**

> :INSTrument:EDG125:RESet
6.7 Status Commands

6.7.5 PPG Status

The PPG Status is used to indicate an alarm and error of the MU181020A 12.5 Gbit/s PPG and MU181020B 14 Gbit/s PPG.

**:INSTrument:PPGG125[:EVENt]?**

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;numeric&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 to 22</td>
<td>Sum total of the event bit (DECIMAL)</td>
</tr>
<tr>
<td>Available bit</td>
<td></td>
</tr>
<tr>
<td>2 (Bit 1)</td>
<td>CMU-MUX Unlock occurs</td>
</tr>
<tr>
<td>4 (Bit 2)</td>
<td>Delay Busy occurs</td>
</tr>
<tr>
<td>16 (Bit 4)</td>
<td>Delay Calibration Require occurs</td>
</tr>
</tbody>
</table>

**Function**
Queries event at 12.5 Gbit/s PPG status.

**Example**

```
> :INSTrument:PPGG125:EVENt?
```

or

```
> :INSTrument:PPGG125?
```

< 1

**:INSTrument:PPGG125:CONDition?**

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;numeric&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 to 22</td>
<td>Sum total of the condition bit (DECIMAL)</td>
</tr>
<tr>
<td>Available bit</td>
<td></td>
</tr>
<tr>
<td>2 (Bit 1)</td>
<td>CMU-MUX Unlock occurs</td>
</tr>
<tr>
<td>4 (Bit 2)</td>
<td>Delay Busy occurs</td>
</tr>
<tr>
<td>16 (Bit 4)</td>
<td>Delay Calibration Require occurs</td>
</tr>
</tbody>
</table>

**Function**
Queries condition at 12.5 Gbit/s PPG status.

**Example**

```
> :INSTrument:PPGG125:CONDition?
```

< 1

**:INSTrument:PPGG125:PTRansition <numeric>**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;numeric&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 to 22</td>
<td>Sum total of the bit of transition filter</td>
</tr>
<tr>
<td>Available bit</td>
<td></td>
</tr>
<tr>
<td>2 (Bit 1)</td>
<td>CMU-MUX Unlock occurs</td>
</tr>
<tr>
<td>4 (Bit 2)</td>
<td>Delay Busy occurs</td>
</tr>
<tr>
<td>16 (Bit 4)</td>
<td>Delay Calibration Require occurs</td>
</tr>
</tbody>
</table>

**Function**
Sets the transition filter (positive direction change) of the 12.5 Gbit/s PPG Status.

**Example**

To set the transition filter (positive direction change) of the 12.5 Gbit/s PPG Status to 1.

```
> :INSTrument:PPGG125:PTRansition 1
```
**Chapter 6  Status Report**

**:INSTrument:PPGG125:PTRansition?**

Response  
<numeric> = <NR1 NUMERIC RESPONSE DATA>  
0 to 22  
Sum total of the bit of transition filter (DECIMAL)

Function  
Queries the transition filter (positive direction change) of the 12.5 Gbit/s PPG Status.

Example  
> :INSTrument:PPGG125:PTRansition?  
< 1

**:INSTrument:PPGG125:NTRansition <numeric>**

Parameter  
<numeric> = <DECIMAL NUMERIC PROGRAM DATA>  
0 to 22  
Sum total of the bit of transition (DECIMAL)  
Available bit  
2 (Bit 1)  
CMU-MUX Unlock occurs  
4 (Bit 2)  
Delay Busy occurs  
16 (Bit 4)  
Delay Calibration Require occurs

Function  
Sets the transition filter (negative direction change) of the 12.5 Gbit/s PPG Status.

Example  
To set the transition filter (negative direction change) of the 12.5 Gbit/s PPG Status to 1.  
> :INSTrument:PPGG125:NTRansition 1

**:INSTrument:PPGG125:NTRansition?**

Response  
<numeric> = <NR1 NUMERIC RESPONSE DATA>  
0 to 22  
Sum total of the bit of transition filter (DECIMAL)

Function  
Queries the transition filter (negative direction change) of the 12.5 Gbit/s PPG Status.

Example  
> :INSTrument:PPGG125:NTRansition?  
< 1

**:INSTrument:PPGG125:RESet**

Function  
Initializes event at 12.5 Gbit/s PPG status.

Example  
> :INSTrument:PPGG125:RESet
6.7.6 Optical Transceiver (XFP) Status

The Optical Transceiver (XFP) Status is used to indicate an alarm and error of the MU181600A Optical Transceiver (XFP).

:INSTRument:XFP[:EVENt]?  
**Response**  
\[
\text{<numeric> = <NR1 NUMERIC RESPONSE DATA>}
\]
\[
0 \text{ to } 31 \quad \text{Sum total of the event bit (DECIMAL)}
\]
Available bit
\[
1 \text{ (Bit 0)} \quad \text{Ready}
\]
\[
2 \text{ (Bit 1)} \quad \text{Laser Fault occurs}
\]
\[
4 \text{ (Bit 2)} \quad \text{Unlock occurs}
\]
\[
8 \text{ (Bit 3)} \quad \text{LOS occurs}
\]
\[
16 \text{ (Bit 4)} \quad \text{CDR Unlock occurs}
\]
**Function**  
Queries event at Optical Transceiver (XFP) status.
**Example**

> :INSTRument:XFP:EVENt?

or

> :INSTRument:XFP?

< 1

:INSTRument:XFP:CONDition?  
**Response**  
\[
\text{<numeric> = <NR1 NUMERIC RESPONSE DATA>}
\]
\[
0 \text{ to } 31 \quad \text{Sum total of the condition bit (DECIMAL)}
\]
Available bit
\[
1 \text{ (Bit 0)} \quad \text{Ready}
\]
\[
2 \text{ (Bit 1)} \quad \text{Laser Fault occurs}
\]
\[
4 \text{ (Bit 2)} \quad \text{Unlock occurs}
\]
\[
8 \text{ (Bit 3)} \quad \text{LOS occurs}
\]
\[
16 \text{ (Bit 4)} \quad \text{CDR Unlock occurs}
\]
**Function**  
Queries condition at Optical Transceiver (XFP) status.
**Example**

> :INSTRument:XFP:CONDition?

< 1
Chapter 6  Status Report

:INSTrument:XFP:PTRansition <numeric>
Parameter
<numeric> = <DECIMAL NUMERIC PROGRAM DATA>
0 to 31  Sum total of the bit of transition filter (DECIMAL)

Available bit
1 (Bit 0)  Ready
2 (Bit 1)  Laser Fault occurs
4 (Bit 2)  Unlock occurs
8 (Bit 3)  LOS occurs
16 (Bit 4) CDR Unlock occurs

Function
Sets the transition filter (positive direction change) of the Optical Transceiver (XFP) Status.

Example
To set the transition filter (positive direction change) of the Optical Transceiver (XFP) Status to 1.
> :INSTrument:XFP:PTRansition 1

:INSTrument:XFP:PTRansition?
Response
<numeric> = <NR1 NUMERIC RESPONSE DATA>
0 to 31  Sum total of the bit of transition filter (DECIMAL)

Function
Queries the transition filter (positive direction change) of the Optical Transceiver (XFP) Status.

Example
> :INSTrument:XFP:PTRansition?
< 1

:INSTrument:XFP:NTRansition <numeric>
Parameter
<numeric> = <DECIMAL NUMERIC PROGRAM DATA>
0 to 31  Sum total of the bit transition filter (DECIMAL)

Available bit
1 (Bit 0)  Ready
2 (Bit 1)  Laser Fault occurs
4 (Bit 2)  Unlock occurs
8 (Bit 3)  LOS occurs
16 (Bit 4) CDR Unlock occurs

Function
Sets the transition filter (negative direction change) of the Optical Transceiver (XFP) Status.

Example
To set the transition filter (negative direction change) of the Optical Transceiver (XFP) Status to 1.
> :INSTrument:XFP:NTRansition 1
### :INStrument:XFP:NTRansition?

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;numeric&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0 to 31</td>
</tr>
<tr>
<td></td>
<td>Sum total of the bit of transition filter</td>
</tr>
<tr>
<td></td>
<td>(DECIMAL)</td>
</tr>
</tbody>
</table>

**Function**
Queries the transition filter (negative direction change) of the Optical Transceiver (XFP) Status.

**Example**

```
> :INStrument:XFP:NTRansition?
< 1
```

### :INStrument:XFP:RESet

<table>
<thead>
<tr>
<th>Function</th>
<th>Initializes event at Optical Transceiver (XFP) status.</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Example</th>
<th>&gt; :INStrument:XFP:RESet</th>
</tr>
</thead>
</table>
6.7.7 Optical Transceiver (SFP) Status

The Optical Transceiver (SFP) Status is used to indicate an alarm and error of the MU181601A Optical Transceiver (SFP).

:INSTRument:SFP[:EVENT]?

Response

<table>
<thead>
<tr>
<th>&lt;numeric&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 to 5 Sum total of the event bit (DECIMAL)</td>
</tr>
<tr>
<td>Available bit</td>
</tr>
<tr>
<td>1 (Bit 0) Ready</td>
</tr>
<tr>
<td>4 (Bit 2) LOS occurs</td>
</tr>
</tbody>
</table>

Function
Queries event at Optical Transceiver (SFP) status.

Example
> :INSTRument:SFP:EVENT?

or
> :INSTRument:SFP?

< 1

:INSTRument:SFP:CONDition?

Response

<table>
<thead>
<tr>
<th>&lt;numeric&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 to 5 Sum total of the condition bit (DECIMAL)</td>
</tr>
<tr>
<td>Available bit</td>
</tr>
<tr>
<td>1 (Bit 0) Ready</td>
</tr>
<tr>
<td>4 (Bit 2) LOS occurs</td>
</tr>
</tbody>
</table>

Function
Queries condition at Optical Transceiver (SFP) status.

Example
> :INSTRument:SFP:CONDition?

< 1

:INSTRument:SFP:PTRansition <numeric>

Parameter

<table>
<thead>
<tr>
<th>&lt;numeric&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 to 5 Sum total of the bit of transition filter (DECIMAL)</td>
</tr>
<tr>
<td>Available bit</td>
</tr>
<tr>
<td>1 (Bit 0) Ready</td>
</tr>
<tr>
<td>4 (Bit 2) LOS occurs</td>
</tr>
</tbody>
</table>

Function
Sets the transition filter (positive direction change) of the Optical Transceiver (SFP) Status.

Example
To set the transition filter (positive direction change) of the Optical Transceiver (SFP) Status to 1.
> :INSTRument:SFP:PTRansition 1
### :INSTrument:SFP:PTRansition?

<table>
<thead>
<tr>
<th>Response</th>
<th>(&lt;\text{numeric}&gt; = \langle\text{NR1 NUMERIC RESPONSE DATA}\rangle)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 to 5</td>
<td>Sum total of the bit of transition filter (DECIMAL)</td>
</tr>
</tbody>
</table>

**Function**
Queries the transition filter (positive direction change) of the Optical Transceiver (SFP) Status.

**Example**

```python
> :INSTrument:SFP:PTRansition?
< 1
```

### :INSTrument:SFP:NTRansition \(<\text{numeric}>\)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>(&lt;\text{numeric}&gt; = \langle\text{DECIMAL NUMERIC PROGRAM DATA}\rangle)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 to 5</td>
<td>Sum total of the bit transition filter (DECIMAL)</td>
</tr>
<tr>
<td>Available</td>
<td>bit</td>
</tr>
<tr>
<td>1 (Bit 0)</td>
<td>Ready</td>
</tr>
<tr>
<td>4 (Bit 2)</td>
<td>LOS occurs</td>
</tr>
</tbody>
</table>

**Function**
Sets the transition filter (negative direction change) of the Optical Transceiver (SFP) Status.

**Example**
To set the transition filter (negative direction change) of the Optical Transceiver (SFP) Status to 1.

```python
> :INSTrument:SFP:NTRansition 1
```

### :INSTrument:SFP:NTRansition?

<table>
<thead>
<tr>
<th>Response</th>
<th>(&lt;\text{numeric}&gt; = \langle\text{NR1 NUMERIC RESPONSE DATA}\rangle)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 to 5</td>
<td>Sum total of the bit of transition filter (DECIMAL)</td>
</tr>
</tbody>
</table>

**Function**
Queries the transition filter (negative direction change) of the Optical Transceiver (SFP) Status.

**Example**

```python
> :INSTrument:SFP:NTRansition?
< 1
```

### :INSTrument:SFP:RESet

**Function**
Initializes event at Optical Transceiver (SFP) status.

**Example**

```python
> :INSTrument:SFP:RESet
```
6.7.8 DEMUX Status

DEMUX Status displays errors and alarms occurring at the MU182040A 25 Gbit/s 1Ch DEMUX, and MU182041A 25 Gbit/s 2 Ch DEMUX.

:INSTrument:DMUX[:EVENt]?

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;numeric&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 to 6336</td>
<td>Sum total of the event bit (DECIMAL)</td>
</tr>
</tbody>
</table>

Available bit

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>64 (Bit6)</td>
<td>Delay Busy Occurrence (Data1)</td>
</tr>
<tr>
<td>128 (Bit7)</td>
<td>Delay Busy Occurrence (Data2)</td>
</tr>
<tr>
<td>2048 (Bit11)</td>
<td>Delay Calibration Require Occurrence (Data1)</td>
</tr>
<tr>
<td>4096 (Bit12)</td>
<td>Delay Calibration Require Occurrence (Data2)</td>
</tr>
</tbody>
</table>

Function

Queries event at 25Gbit/s DEMUX status.

Example

> :INSTrument:DMUX:EVENt?

or

> :INSTrument:DMUX?

< 64

:INSTrument:DMUX:CONDition?

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;numeric&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 to 6336</td>
<td>Sum total of the condition bit (DECIMAL)</td>
</tr>
</tbody>
</table>

Available bit

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>64 (Bit6)</td>
<td>Delay Busy Occurrence (Data1)</td>
</tr>
<tr>
<td>128 (Bit7)</td>
<td>Delay Busy Occurrence (Data2)</td>
</tr>
<tr>
<td>2048 (Bit11)</td>
<td>Delay Calibration Require Occurrence (Data1)</td>
</tr>
<tr>
<td>4096 (Bit12)</td>
<td>Delay Calibration Require Occurrence (Data2)</td>
</tr>
</tbody>
</table>

Function

Queries condition at 25Gbit/s DEMUX status.

Example

> :INSTrument:DMUX:CONDition?

< 64
6.7  Status Commands

:INStrument:DMUX:PTRansition <numeric>

Parameter  <numeric> = <DECIMAL NUMERIC PROGRAM DATA>
0 to 6336   Sum total of the bit of transition filter
            (DECIMAL)

Available bit
64 (Bit6)   Delay Busy Occurrence (Data1)
128 (Bit7)  Delay Busy Occurrence (Data2)
2048 (Bit11) Delay Calibration Require Occurrence (Data1)
4096 (Bit12) Delay Calibration Require Occurrence (Data2)

Function Sets the transition filter (positive direction change) of the 25Gbit/s
DEMUX Status.

Example To set the transition filter (positive direction change) of the 25Gbit/s
DEMUX Status to 1.
> :INStrument:DMUX:PTRansition 64

:INStrument:DMUX:PTRansition?

Response  <numeric> = <NR1 NUMERIC RESPONSE DATA>
0 to 6336   Sum total of the bit of transition filter
            (DECIMAL)

Function Queries the transition filter (positive direction change) of the 25Gbit/s
DEMUX Status.

Example
> :INStrument:DMUX:PTRansition?
< 64

:INStrument:DMUX:NTRansition <numeric>

Parameter  <numeric> = <DECIMAL NUMERIC PROGRAM DATA>
0 to 6336   Sum total of the bit of transition filter
            (DECIMAL)

Available bit
64 (Bit6)   Delay Busy Occurrence (Data1)
128 (Bit7)  Delay Busy Occurrence (Data2)
2048 (Bit11) Delay Calibration Require Occurrence (Data1)
4096 (Bit12) Delay Calibration Require Occurrence (Data2)

Function Sets the transition filter (negative direction change) of the 25Gbit/s
DEMUX Status.

Example To set the transition filter (negative direction change) of the 25Gbit/s
DEMUX Status to 1.
> :INStrument:DMUX:NTRansition 64
Chapter 6  Status Report

## :INSTRument:DMUX:NTRansition?

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;numeric&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 to 6336</td>
<td>Sum total of the bit of transition filter</td>
</tr>
<tr>
<td>(DECIMAL)</td>
<td></td>
</tr>
</tbody>
</table>

**Function**

Queries the transition filter (negative direction change) of the 25Gbit/s DEMUX Status.

**Example**

```plaintext
> :INSTRument:DMUX:NTRansition?
< 64
```

## :INSTRument:DMUX:RESet

**Function**

Initializes event at 25Gbit/s DEMUX status.

**Example**

```plaintext
> :INSTRument:DMUX:RESet
```
6.7 Status Commands

6.7.9 MUX Status

MUX Status displays errors and alarms occurring at the MU182020A 25 Gbit/s 1Ch MUX, and MU182021A 25 Gbit/s 2 Ch MUX.

:INSTrument:MUX[:EVENT]?

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;numeric&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 to 60</td>
<td>Sum total of the event bit (DECIMAL)</td>
</tr>
<tr>
<td>Available bit</td>
<td></td>
</tr>
<tr>
<td>4 (Bit2)</td>
<td>Delay Busy Occurrence (Data1)</td>
</tr>
<tr>
<td>8 (Bit3)</td>
<td>Delay Busy Occurrence (Data2)</td>
</tr>
<tr>
<td>16 (Bit4)</td>
<td>Delay Calibration Require Occurrence (Data1)</td>
</tr>
<tr>
<td>32 (Bit5)</td>
<td>Delay Calibration Require Occurrence (Data2)</td>
</tr>
</tbody>
</table>

Function
Queries event at 25Gbit/s MUX status.

Example
> :INSTrument:MUX:EVENT?
or
> :INSTrument:MUX?
< 4

:INSTrument:MUX:CONDition?

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;numeric&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 to 60</td>
<td>Sum total of the condition bit (DECIMAL)</td>
</tr>
<tr>
<td>Available bit</td>
<td></td>
</tr>
<tr>
<td>4 (Bit2)</td>
<td>Delay Busy Occurrence (Data1)</td>
</tr>
<tr>
<td>8 (Bit3)</td>
<td>Delay Busy Occurrence (Data2)</td>
</tr>
<tr>
<td>16 (Bit4)</td>
<td>Delay Calibration Require Occurrence (Data1)</td>
</tr>
<tr>
<td>32 (Bit5)</td>
<td>Delay Calibration Require Occurrence (Data2)</td>
</tr>
</tbody>
</table>

Function
Queries condition at 25Gbit/s MUX status.

Example
> :INSTrument:MUX:CONDition?
< 4
Chapter 6  Status Report

:INSTrument:MUX:PTRansition <numeric>

Parameter  
<numeric> = <DECIMAL NUMERIC PROGRAM DATA>  
0 to 60  
  Sum total of the bit of transition filter  
  (DECIMAL)

Available bit  
4 (Bit2)  
8 (Bit3)  
16 (Bit4)  
32 (Bit5)  
Delay Busy Occurrence (Data1)  
Delay Busy Occurrence (Data2)  
Delay Calibration Require Occurrence (Data1)  
Delay Calibration Require Occurrence (Data2)

Function  
Sets the transition filter (positive direction change) of the 25Gbit/s MUX Status.

Example  
To set the transition filter (positive direction change) of the 25Gbit/s MUX Status to 1.  
> :INSTrument:MUX:PTRansition 4

:INSTrument:MUX:PTRansition?

Response  
<numeric> = <NR1 NUMERIC RESPONSE DATA>  
0 to 60  
  Sum total of the bit of transition filter  
  (DECIMAL)

Function  
Queries the transition filter (positive direction change) of the 25Gbit/s MUX Status.

Example  
> :INSTrument:MUX:PTRansition?
< 4

:INSTrument:MUX:NTRansition <numeric>

Parameter  
<numeric> = <DECIMAL NUMERIC PROGRAM DATA>  
0 to 60  
  Sum total of the bit of transition filter  
  (DECIMAL)

Available bit  
4 (Bit2)  
8 (Bit3)  
16 (Bit4)  
32 (Bit5)  
Delay Busy Occurrence (Data1)  
Delay Busy Occurrence (Data2)  
Delay Calibration Require Occurrence (Data1)  
Delay Calibration Require Occurrence (Data2)

Function  
Sets the transition filter (negative direction change) of the 25Gbit/s MUX Status.

Example  
To set the transition filter (negative direction change) of the 25Gbit/s MUX Status to 1.  
> :INSTrument:MUX:NTRansition 4
### :INSTrument:MUX:NTRansition?

**Response**
- `<numeric> = <NR1 NUMERIC RESPONSE DATA>`
- `0 to 60` Sum total of the bit of transition filter (DECIMAL)

**Function**
Queries the transition filter (negative direction change) of the 25Gbit/s MUX Status.

**Example**
- `> :INSTrument:MUX:NTRansition?`
- `< 4`

### :INSTrument:MUX:RESet

**Function**
Initializes event at 25Gbit/s MUX status.

**Example**
- `> :INSTrument:MUX:RESet`
Chapter 7  SCPI Commands

This chapter describes the SCPI commands.

7.1 Common Commands .............................................. 7-5
  7.1.1 Commands for common settings ....................... 7-5
  7.1.2 Command related to File menu ..................... 7-22
  7.1.3 Common function button .......................... 7-36
  7.1.4 Auto Search ............................................. 7-41
  7.1.5 Auto Adjust ............................................. 7-47
  7.1.6 ED ALL .................................................. 7-51
  7.1.7 Pattern Editor ........................................... 7-59

7.2 Synthesizer Commands ........................................... 7-61
  7.2.1 Operation frequency Setting field .................. 7-63
  7.2.2 Reference Clock Setting field ..................... 7-66
  7.2.3 Spectrum Spread setting commands .............. 7-67
  7.2.4 Jitter setting commands .......................... 7-69

7.3 PPG Commands .................................................. 7-78
  7.3.1 Commands related to Output tab .................. 7-78
  7.3.2 Commands related to Pattern tab ................. 7-111
  7.3.3 Commands related to Error Addition tab .......... 7-183
  7.3.4 Commands related to Misc tab .................... 7-191

7.4 ED Commands .................................................. 7-209
  7.4.1 Commands related to Result tab ................. 7-209
  7.4.2 Commands related to Measurement tab .......... 7-235
  7.4.3 Commands related to Pattern tab ............... 7-256
  7.4.4 Commands related to Input tab .................. 7-319
  7.4.5 Commands related to Capture tab ............... 7-336
  7.4.6 Commands related to Misc tab ................... 7-347

7.5 Optical Module Commands ...................................... 7-359
  7.5.1 XFP/SFP module ...................................... 7-359

7.6 Automatic Measurement Commands ........................... 7-368
  7.6.1 ISI measurement .................................. 7-368
  7.6.2 Eye Margin measurement .......................... 7-383
  7.6.3 Eye Diagram measurement ....................... 7-393
  7.6.4 Q measurement .................................... 7-435
  7.6.5 Bathtub measurement ............................. 7-470

7.7 Stressed Eye Transmitter Commands ....................... 7-490
  7.7.1 Commands related to Optical tab ............... 7-490

7.8 MUX Commands .................................................. 7-504
  7.8.1 Commands related to Data Interface tab ........ 7-504
  7.8.2 Commands related to Clock Interface tab ........ 7-523
  7.8.3 Commands related to Emphasis tab ............. 7-533

7.9 DEMUX Commands ............................................... 7-542
  7.9.1 Commands related to Data Interface tab ....... 7-542
<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.9.2</td>
<td>Commands related to Clock Interface tab</td>
<td>7-549</td>
</tr>
<tr>
<td>7.9.3</td>
<td>Commands related to Result tab</td>
<td>7-558</td>
</tr>
<tr>
<td>7.10</td>
<td>Jitter Commands</td>
<td>7-564</td>
</tr>
<tr>
<td>7.10.1</td>
<td>List of commands</td>
<td>7-564</td>
</tr>
<tr>
<td>7.10.2</td>
<td>Clock Source Settings Commands</td>
<td>7-566</td>
</tr>
<tr>
<td>7.10.3</td>
<td>Jitter Setting Commands</td>
<td>7-570</td>
</tr>
<tr>
<td>7.10.4</td>
<td>Output Setting Commands</td>
<td>7-589</td>
</tr>
<tr>
<td>7.10.5</td>
<td>File Menu Setting Commands</td>
<td>7-595</td>
</tr>
<tr>
<td>7.11</td>
<td>28G/32G bit/s PPG Commands</td>
<td>7-597</td>
</tr>
<tr>
<td>7.11.1</td>
<td>Common Commands</td>
<td>7-597</td>
</tr>
<tr>
<td>7.11.2</td>
<td>Output Tab</td>
<td>7-631</td>
</tr>
<tr>
<td>7.11.3</td>
<td>Pattern Tab</td>
<td>7-652</td>
</tr>
<tr>
<td>7.11.4</td>
<td>Error Addition Tab</td>
<td>7-681</td>
</tr>
<tr>
<td>7.11.5</td>
<td>Pre-Code Tab</td>
<td>7-687</td>
</tr>
<tr>
<td>7.11.6</td>
<td>Misc1 Tab</td>
<td>7-689</td>
</tr>
<tr>
<td>7.11.7</td>
<td>Misc 2 Tab</td>
<td>7-703</td>
</tr>
<tr>
<td>7.12</td>
<td>28G/32G bit/s ED Commands</td>
<td>7-714</td>
</tr>
<tr>
<td>7.12.1</td>
<td>Common Commands</td>
<td>7-714</td>
</tr>
<tr>
<td>7.12.2</td>
<td>Commands Related to Result Tab</td>
<td>7-749</td>
</tr>
<tr>
<td>7.12.3</td>
<td>Commands Related to Measurement Tab</td>
<td>7-764</td>
</tr>
<tr>
<td>7.12.4</td>
<td>Commands Related to Pattern Tab</td>
<td>7-783</td>
</tr>
<tr>
<td>7.12.5</td>
<td>Commands Related to Input Tab</td>
<td>7-820</td>
</tr>
<tr>
<td>7.12.6</td>
<td>Commands Related to Capture Tab</td>
<td>7-838</td>
</tr>
<tr>
<td>7.12.7</td>
<td>Commands Related to Misc1 Tab</td>
<td>7-848</td>
</tr>
<tr>
<td>7.12.8</td>
<td>Commands Related to Misc2 Tab</td>
<td>7-858</td>
</tr>
<tr>
<td>7.12.9</td>
<td>Eye Margin Measurement</td>
<td>7-861</td>
</tr>
<tr>
<td>7.12.10</td>
<td>Eye Diagram Measurement</td>
<td>7-872</td>
</tr>
<tr>
<td>7.12.11</td>
<td>Bathtub measurement</td>
<td>7-905</td>
</tr>
<tr>
<td>7.12.12</td>
<td>Auto Search</td>
<td>7-929</td>
</tr>
<tr>
<td>7.12.13</td>
<td>Auto Adjust</td>
<td>7-936</td>
</tr>
<tr>
<td>7.12.14</td>
<td>Q measurement</td>
<td>7-940</td>
</tr>
<tr>
<td>7.12.15</td>
<td>PAM BER measurement</td>
<td>7-977</td>
</tr>
<tr>
<td>7.12.16</td>
<td>Eye Contour measurement</td>
<td>7-1004</td>
</tr>
</tbody>
</table>
Examples of command expression are shown below.

<Program Command>

<table>
<thead>
<tr>
<th>Program Command</th>
<th>Parameter Type (SCPI, device-unique)</th>
<th>Parameter Type (IEEE488.2,SCPI)</th>
</tr>
</thead>
<tbody>
<tr>
<td>:SOURce:PATTern:TYPE</td>
<td>&lt;type&gt;=&lt;CHARACTER PROGRAM DATA&gt;</td>
<td></td>
</tr>
<tr>
<td>PRBS</td>
<td>PRBS pattern</td>
<td></td>
</tr>
<tr>
<td>ZSUBstitution</td>
<td>Zero-substitution pattern</td>
<td></td>
</tr>
<tr>
<td>DATA</td>
<td>Data pattern</td>
<td></td>
</tr>
<tr>
<td>ALTernate</td>
<td>Alternate pattern</td>
<td></td>
</tr>
<tr>
<td>MIXData</td>
<td>Mixed Data pattern</td>
<td></td>
</tr>
<tr>
<td>MIXalt</td>
<td>Mixed Alternate pattern</td>
<td></td>
</tr>
<tr>
<td>SEQuence</td>
<td>Sequence pattern</td>
<td></td>
</tr>
</tbody>
</table>

Function:
Sets the type of the test pattern.

Example:
To set the test pattern type to the Mixed Data pattern:
> :SOURce:PATTern:TYPE MIXData

Compatibility:
Partially compatible with the MP1632C (ZSUBstitution only).
Chapter 7  SCPI Commands

<Program Query Command>

Meanings of < and > at the beginning are:
> for Program message
< for Response

Notes:

- All the Instrument commands are sequential commands.
- If a command affects other settings, the command may have restriction. For setting parameters subject to be affected and command conditions to be restricted, see the MX180000A Signal Quality Analyzer Control Software Operation Manual and operation manual of each module.
- The parameters of a query command may be omitted when they are the same as those of the corresponding program command.
7.1 Common Commands

This chapter describes the commands related to common settings and functions of the control software.

7.1.1 Commands for common settings

Table 7.1.1-1 Common setting commands

<table>
<thead>
<tr>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of the unit to be operated</td>
<td>:UENTry:ID :UENTry:ID?</td>
</tr>
<tr>
<td>Number of the module to be operated (slot position)</td>
<td>:MODule:ID :MODule:ID?</td>
</tr>
<tr>
<td>Number of the port to be operated (port position)</td>
<td>:PORT:ID :PORT:ID?</td>
</tr>
<tr>
<td>Automatic measurement function to be performed</td>
<td>:SYSTem:CFUNction :SYSTem:CFUNction?</td>
</tr>
<tr>
<td>Query for error message</td>
<td>:SYSTem:ERRor?</td>
</tr>
<tr>
<td>Query for SCPI version</td>
<td>:SYSTem:VERSION?</td>
</tr>
<tr>
<td>Query for software status</td>
<td>:SYSTem:CONDition?</td>
</tr>
<tr>
<td>Query for hardware system configuration</td>
<td>:SYSTem:ORGanization:HARDware?</td>
</tr>
<tr>
<td>Query for system error</td>
<td>:SYSTem:INFormation:ERRor?</td>
</tr>
<tr>
<td>Terminator type</td>
<td>:SYSTem:TERMination :SYSTem:TERMination?</td>
</tr>
<tr>
<td>Query for model name of mainframe and module</td>
<td>:SYSTem:CONDition:UNITs?</td>
</tr>
<tr>
<td>Query for mainframe information</td>
<td>:SYSTem:UNIT?</td>
</tr>
<tr>
<td>Query for module information</td>
<td>:SYSTem:MODule?</td>
</tr>
<tr>
<td>Measured Results Screen Drawing Settings</td>
<td>:SYSTem:DISPlay:RESult :SYSTem:DISPlay:RESult?</td>
</tr>
<tr>
<td>Module screen display</td>
<td>:DISPlay:ACTive</td>
</tr>
</tbody>
</table>
Chapter 7  SCPI Commands

:UENTry:ID <unit_number>

Parameter  
<unit_number> = <DECIMAL NUMERIC PROGRAM DATA>
1 to 4  
Unit number

Function  
Sets the number of the unit to be operated.

Example  
To set the number of the unit to be operated to 2:
> :UENTry:ID 2

Compatibility  
Incompatible with existing models.

:UENTry:ID?

Response  
<unit_number> = <NR1 NUMERIC RESPONSE DATA>
1 to 4

Function  
Queries the number of the unit being operated.

Example  
> :UENTry:ID?
< 2

Compatibility  
Incompatible with existing models.

:MODule:ID <module_number>

Parameter  
<module_number> = <DECIMAL NUMERIC PROGRAM DATA>
1 to 6  
Module number

Function  
Sets the number of the module to be operated (slot position).

Example  
To set the number of the module to be operated (slot position) to 6:
> :MODule:ID 6

Compatibility  
Incompatible with existing models.

:MODule:ID?

Response  
<module_number> = <NR1 NUMERIC RESPONSE DATA>
1 to 6

Function  
Queries the number of the module being operated (slot position).

Example  
> :MODule:ID?
< 6

Compatibility  
Incompatible with existing models.
### 7.1 Common Commands

#### :PORT:ID <port_number>

**Parameter**

<port_number> = <DECIMAL NUMERIC PROGRAM DATA>

Identifier for the port to be operated. Indicates the physical position number of the port assigned to a module.

**Function**

Sets the number of the port to be operated (port position).

**Example**

To set the number of the port to be operated (port position) to 1:

```plaintext
> :PORT:ID 1
```

**Compatibility**

Incompatible with existing models.

#### :PORT:ID?

**Response**

<port_number> = <NR1 NUMERIC RESPONSE DATA>

Identifier for the port to be operated. Indicates the physical position number of the port assigned to a module.

**Function**

Queries the number of the port being operated (port position).

**Example**

```plaintext
> :PORT:ID?
< 1
```

**Compatibility**

Incompatible with existing models.

#### :SYSTem:CFUNction <function>

**Parameter**

<function> = <CHARACTER PROGRAM DATA>

- ASE  Auto Search
- ISI  ISI measurement
- EMAR Eye Margin measurement
- EDI  Eye Diagram measurement
- QAN  Q measurement
- BTUB Bathtub measurement
- AADJ  Auto Adjust
- OFF  Off

**Note:**

When "Off" is set, the operation returns to the port operation previously performed.

**Function**

Sets the automatic measurement function to be performed.

**Example**

To set the common function to be performed to Auto Search:

```plaintext
> :SYSTem:CFUNction ASE
```

**Compatibility**

Incompatible with existing models.
### :SYSTem:CFUNction?

**Response**: 
```
<function> = <CHARACTER RESPONSE DATA>
```

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASE</td>
<td>Auto Search</td>
</tr>
<tr>
<td>ISI</td>
<td>ISI measurement</td>
</tr>
<tr>
<td>EMAR</td>
<td>Eye Margin measurement</td>
</tr>
<tr>
<td>EDI</td>
<td>Eye Diagram measurement</td>
</tr>
<tr>
<td>QAN</td>
<td>Q measurement</td>
</tr>
<tr>
<td>BTUB</td>
<td>Bathtub measurement</td>
</tr>
<tr>
<td>AADJ</td>
<td>Auto Adjust</td>
</tr>
<tr>
<td>OFF</td>
<td>Off</td>
</tr>
</tbody>
</table>

**Function**: Queries the automatic measurement function being performed.

**Example**: 
```
> :SYSTem:CFUNction?
< ASE
```

**Compatibility**: Incompatible with existing models.

### :SYSTem:ERRor?

**Response**: 
```
<error/event_number>,"<error/event_description>"
```

- `<error/event_number>` = `<NR1 NUMERIC RESPONSE DATA>`
  - -32768 to 32767
- "0" indicates that no errors and events have occurred.
- Other values return a general error reserved by SCPI or a device-unique error.
- `<error/event_description>` = `<STRING RESPONSE DATA>`
  - This is an error message corresponding to `<error/event_number>`.
  - The maximum character-string length is 255 characters.

**Function**: Queries the error message in the error/event queue.

**Example**: 
```
> :SYSTem:ERRor?
< 0, "No error"
```

**Compatibility**: Compatible with the MP1632C Digital Data Analyzer (hereinafter, referred to as "MP1632C") and MP1776A Error Detector (hereinafter, referred to as "MP1776A").

### :SYSTem:VERSion?

**Response**: 
```
<version> = <NR2 NUMERIC RESPONSE DATA>
```

- `YYYY.V`  
  - `YYYY`: Year
  - `V`: Revision number

**Function**: Queries the SCPI version to which the MP1800A/MT1810A conforms.

**Example**: 
```
> :SYSTem:VERSion?
< 1999.0
```

**Compatibility**: Compatible with the MP1632C and MP1776A.
7.1 Common Commands

:SYSTem:CONDition?

Response

<mainframe>,<slot1>,...,<slot64>
<mainframe> = <serial>,<mver>,<hver>,<opt1>,<sbver>,<saver>,<opt2>
<serial> = <STRING RESPONSE DATA>
XXXXXXXXXXX 0000000000 to 9999999999
Mainframe serial number

Note:
Alphabetic characters may be included.

<mver> = <STRING RESPONSE DATA>
XXXX.XX.XX 1.00.00 to 9999.99.99
Main application software version

<hver> = <STRING RESPONSE DATA>
XXXX.XX.XX 1.00.00 to 9999.99.99
Mainframe hardware version

<opt1> = <STRING RESPONSE DATA>
OPTXXX Option number (MP1800A/MT1810A)
See Table 7.1.1-2 "Option character correspondence table".

Note:
Outputs the numbers for all installed options.
NONE is output if no option is installed.

<sbver> = <STRING RESPONSE DATA>
XXXX.XX.XX 1.00.00 to 9999.99.99
Sub application software version (Boot part)

<saver> = <STRING RESPONSE DATA>
XXXX.XX.XX 1.00.00 to 9999.99.99
Sub application software version (Application part)

<opt2> = <STRING RESPONSE DATA>
<slot x> = <module>,<serial>,<fpga1>[,<fpga2>],[<boot>],
<Application>,<opt>
x indicates a slot number. The slot number varies depending on the unit number as follows.

Unit 1: 1 to 16 For the MP1800A, numbers from 1 to 6 correspond to actual slots. For the MT1810A, numbers from 1 to 4 correspond to actual slots.

Unit 2: 17 to 32 For the MT1810A, numbers from 17 to 20 correspond to actual slots.

Unit 3: 33 to 48 For the MT1810A, numbers from 33 to 36 correspond to actual slots.

Unit 4: 49 to 64 For the MT1810A, numbers from 49 to 52 correspond to actual slots.
Chapter 7  SCPI Commands

<module> = <STRING RESPONSE DATA>

Module name (e.g.: MU181020A)

See Table 7.1.1-2 "Option character correspondence table".

Note:
NONE is output if no module is installed. For a module that uses two slots, only the slot with the greater number is valid.

<serial> = <STRING RESPONSE DATA>

Serial number

Note:
"-------------" is output if no module is installed. For a module that uses two slots, only the slot with the greater number is valid.

<fpga1>[,<fpga2>,.....] = <STRING RESPONSE DATA>

FPGA version

Note:
"-------------" is output if Logic boot is not installed. For a module that uses two slots, only the slot with the greater number is valid.

<boot> = <STRING RESPONSE DATA>

Logic Boot version

Note:
"-------------" is output if Logic Application is not installed. For a module that uses two slots, only the slot with the greater number is valid.

<application> = <STRING RESPONSE DATA>

Logic Application version

Note:
"-------------" is output if Logic Application is not installed. For a module that uses two slots, only the slot with the greater number is valid.

<opt> = <STRING RESPONSE DATA>

Option number

OPTXXX: For MP1800A/MT1810A

Note:
Outputs the numbers for all installed options. For a module that uses two slots, only the slot with the greater number is valid. The option names of the MU181020B-003 and MU181040B-003 are not output.
7.1 Common Commands

Function
Queries the software status of the MP1800A/MT1810A.

Example
> :SYSTem:CONDition?
<
6201234567,1.00.00,1.00.20,OPT302,1.00.00,1.00.00,OPT12,
OPT14,
MU181000A,6201234568,1.00.00,1.00.00,1.00.00,OPT101,
MU181020A,6201234569,1.00.00,1.00.00,1.00.00,OPT001,OPT2
20,
MU181040A,6201234571,1.00.00,1.00.00,1.00.00,OPT002,OPT2
20

Compatibility
Incompatible with existing models.
### Table 7.1.1-2  Option character correspondence table

<table>
<thead>
<tr>
<th>Model/Name</th>
<th>Option Number</th>
<th>Option Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>MP1800A Signal Quality Analyzer</td>
<td>OPT014</td>
<td>2-Slot for PPG and/or ED</td>
</tr>
<tr>
<td></td>
<td>OPT015</td>
<td>4-Slot for PPG and/or ED</td>
</tr>
<tr>
<td></td>
<td>OPT016</td>
<td>6-Slot for PPG and/or ED</td>
</tr>
<tr>
<td>MT1810A 4 Slot Chassis</td>
<td>OPT014</td>
<td>2-Slot for PPG and/or ED</td>
</tr>
<tr>
<td></td>
<td>OPT015</td>
<td>4-Slot for PPG and/or ED</td>
</tr>
<tr>
<td>MX1800000A Signal Quality Analyzer Control Software</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>MU181000A 12.5GHz Synthesizer</td>
<td>OPTx01</td>
<td>Jitter Modulation</td>
</tr>
<tr>
<td>MU181000B 12.5GHz 4port Synthesizer</td>
<td>OPTx01</td>
<td>Jitter Modulation</td>
</tr>
<tr>
<td>MU181020A 12.5Gbit/s PPG</td>
<td>OPT001</td>
<td>9.8 to 12.5 Gbit/s</td>
</tr>
<tr>
<td></td>
<td>OPT002</td>
<td>0.1 to 12.5 Gbit/s</td>
</tr>
<tr>
<td></td>
<td>OPTx10</td>
<td>Variable Data Output (0.05 to 0.8 Vp-p)</td>
</tr>
<tr>
<td></td>
<td>OPTx11</td>
<td>Variable Data Output (0.25 to 2.5 Vp-p)</td>
</tr>
<tr>
<td></td>
<td>OPTx12</td>
<td>High Performance Data Output (0.05 to 2.0 Vp-p)</td>
</tr>
<tr>
<td></td>
<td>OPTx13</td>
<td>Variable Data Output (0.5 to 3.5 Vp-p)</td>
</tr>
<tr>
<td></td>
<td>OPTx21</td>
<td>Differential Clock Output (0.1 to 2.0 Vp-p)</td>
</tr>
<tr>
<td></td>
<td>OPTx30</td>
<td>Variable Data Delay</td>
</tr>
<tr>
<td>MU181040A 12.5Gbit/s ED</td>
<td>OPT001</td>
<td>9.8 to 12.5 Gbit/s</td>
</tr>
<tr>
<td></td>
<td>OPT002</td>
<td>0.1 to 12.5 Gbit/s</td>
</tr>
<tr>
<td></td>
<td>OPTx20</td>
<td>Clock Recovery</td>
</tr>
<tr>
<td></td>
<td>OPTx30</td>
<td>Variable Clock Delay</td>
</tr>
<tr>
<td>MU181600A Optical Transceiver (XFP)</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>MU181601A Optical Transceiver (SFP)</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>MU181620A Stressed Eye Transmitter</td>
<td>OPTx01</td>
<td>1310nm Reference</td>
</tr>
<tr>
<td></td>
<td>OPTx02</td>
<td>1550nm Reference</td>
</tr>
<tr>
<td></td>
<td>OPTx03</td>
<td>1310nm/1550nm Reference</td>
</tr>
<tr>
<td></td>
<td>OPTx11</td>
<td>1310nm Stressed Eye</td>
</tr>
<tr>
<td></td>
<td>OPTx12</td>
<td>1550nm Stressed Eye</td>
</tr>
<tr>
<td></td>
<td>OPTx13</td>
<td>1310nm/1550nm Stressed Eye</td>
</tr>
<tr>
<td>MU181640A Optical Receiver</td>
<td>OPTx04</td>
<td>Band Width 8.5GHz</td>
</tr>
<tr>
<td>MU181800A 12.5GHz Clock Distributor</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>MU181020B 14Gbit/s PPG</td>
<td>OPT002</td>
<td>0.1 to 14 Gbit/s</td>
</tr>
<tr>
<td></td>
<td>OPTx11</td>
<td>Variable Data Output (0.25 to 2.5 Vp-p)</td>
</tr>
<tr>
<td></td>
<td>OPTx12</td>
<td>High Performance Data Output (0.05 to 2.0 Vp-p)</td>
</tr>
<tr>
<td></td>
<td>OPTx13</td>
<td>Variable Data Output (0.5 to 3.5 Vp-p)</td>
</tr>
<tr>
<td></td>
<td>OPTx21</td>
<td>Differential Clock Output (0.1 to 2.0 Vp-p)</td>
</tr>
<tr>
<td></td>
<td>OPTx30</td>
<td>Variable Data Delay</td>
</tr>
</tbody>
</table>
### Table 7.1.1-2 Option character correspondence table (Cont'd)

<table>
<thead>
<tr>
<th>Model/Name</th>
<th>Option Number</th>
<th>Option Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>MU181040B 14Gbit/s ED</td>
<td>OPT02</td>
<td>0.1 to 14 Gbit/s</td>
</tr>
<tr>
<td></td>
<td>OPTx20</td>
<td>Clock Recovery</td>
</tr>
<tr>
<td></td>
<td>OPTx30</td>
<td>Variable Clock Delay</td>
</tr>
<tr>
<td>MU182020A 25Gbit/s 1ch MUX</td>
<td>OPTx01</td>
<td>28Gbit/s Extension</td>
</tr>
<tr>
<td>MU182021A 25Gbit/s 2ch MUX</td>
<td>OPTx02</td>
<td>Clock Input Band Switch</td>
</tr>
<tr>
<td></td>
<td>OPTx10</td>
<td>Variable Data Output (0.25 to 1.75Vp-p)</td>
</tr>
<tr>
<td></td>
<td>OPTx11</td>
<td>Variable Data Output (0.5 to 2.5Vp-p)</td>
</tr>
<tr>
<td></td>
<td>OPTx12</td>
<td>Variable Data Output (0.5 to 2.0Vp-p)</td>
</tr>
<tr>
<td></td>
<td>OPTx13</td>
<td>Variable Data Output (0.5 to 3.5Vp-p)</td>
</tr>
<tr>
<td></td>
<td>OPTx21</td>
<td>Differential Clock Output (0.5 to 2.0Vp-p)</td>
</tr>
<tr>
<td></td>
<td>OPTx30</td>
<td>25Gbit/s Variable Data Delay</td>
</tr>
<tr>
<td></td>
<td>OPTx31</td>
<td>28Gbit/s Variable Data Delay</td>
</tr>
<tr>
<td>MU182040A 25Gbit/s 1ch DEMUX</td>
<td>OPTx01</td>
<td>28Gbit/s Extension</td>
</tr>
<tr>
<td>MU182041A 25Gbit/s 2ch DEMUX</td>
<td>OPTx02</td>
<td>Clock Input Band Switch</td>
</tr>
<tr>
<td></td>
<td>OPTx30</td>
<td>25GHz Variable Clock Delay</td>
</tr>
<tr>
<td></td>
<td>OPTx31</td>
<td>28GHz Variable Clock Delay</td>
</tr>
</tbody>
</table>

**Note:**

The option names of the MU181020B-003 and MU181040B-003 are not output.
Chapter 7  SCPI Commands

:SYSTem:ORGanization:HARDware?

Response

\(<\text{slot}1>,\ldots,\text{<slot64}>\)

\(<\text{slot}x> = \text{<module>},\text{<serial>},\text{<fpga1>}[\ldots,\text{<fpga2>},\ldots]\text{<boot>},\text{<application>},\text{<opt>}

x indicates a slot number. The slot number varies depending on the unit number as follows.

Unit 1: 1 to 16  For the MP1800A, numbers from 1 to 6 correspond to actual slots.

For the MT1810A, numbers from 1 to 4 correspond to actual slots.

Unit 2: 17 to 32  For the MT1810A, numbers from 17 to 20 correspond to actual slots.

Unit 3: 33 to 48  For the MT1810A, numbers from 33 to 36 correspond to actual slots.

Unit 4: 49 to 64  For the MT1810A, numbers from 49 to 52 correspond to actual slots.

\(<\text{module}> = \text{<STRING RESPONSE DATA>}

XXXXX XXXX  Module name (e.g.,: MU181020A)

See Table 7.1.1-2 "Option character correspondence table".

Note:

NONE is output if no module is installed.

For a module that uses two slots, only the slot with the greater number is valid.

\(<\text{serial}> = \text{<STRING RESPONSE DATA>}

XXXXX XXXX  Serial number

Note:

"---------" is output if no module is installed.

For a module that uses two slots, only the slot with the greater number is valid.

\(<\text{fpga1>}[\ldots,\text{<fpga2>},\ldots]\text{<boot>},\text{<application>},\text{<opt>}

XXXX.XX.XX  FPGA version

Note:

"---------" is output if Logic Boot is not installed.

For a module that uses two slots, only the slot with the greater number is valid.
<application> = <STRING RESPONSE DATA>
XXXX.XX.XX 1.00.00 to 9999.99.99
Logic Application version

**Note:**
"---------" is output if Logic Application is not installed.
For a module that uses two slots, only the slot with the greater number is valid.

<opt> = <STRING RESPONSE DATA>
XXXXXX/XXXXX Option number
OPTXXX: For MP1800A/MT1810A
Outputs the numbers for all installed options.

**Note:**
NONE is output if no module is installed.
For a module that uses two slots, only the slot with the greater number is valid.
The option names of the MU181020B-003 and MU181040B-003 are not output.

### Function
Queries the hardware system configuration of the MP1800A

```plaintext
> :SYStem:ORGanization:HARDware?
< MU181000A,6201234568,1.00.00,1.00.00,1.00.00,OPT101,
MU181020A,6201234569,1.00.00,1.00.00,1.00.00,OPT001,OPT20,
MU181040A,6201234571,1.00.00,1.00.00,1.00.00,OPT002,OPT20
```

### Compatibility
Partially compatible with the MP1632C and MP1776A.
### :SYSTem:INFormation:ERRor? <unit>

**Parameter**
- `<unit>` = `<DECIMAL NUMERIC PROGRAM DATA>`
- 1 to 4

**Response**
- `<numeric>` = `<NR1 NUMERIC RESPONSE DATA>`
  - 0: NONE
  - 1: PLL Unlock
  - 2: Temperature
  - 3: Fan

All the system errors that have currently occurred are displayed, delimited with a comma (,).

**Function**
Queries the System Error contents.

**Example**
To query the System Error contents:

```
> :SYSTem:INFormation:ERRor? 3
< 1,2,3
```

*when a system error has occurred for PLL Unlock, Temperature, or Fan*

```
< 0
```

*when no system error has occurred*

**Compatibility**
Incompatible with existing models.

### :SYSTem:TERMination <numeric>

**Parameter**
- `<numeric>` = `<DECIMAL NUMERIC PROGRAM DATA>`
  - 0: LF+EOI
  - 1: CR+LF+EOI

**Function**
Sets the terminator type of the response data.

**Example**
To set the terminator type to LF+EOI:

```
> :SYSTem:TERMination 0
```

**Compatibility**
Compatible with the MP1632C.

### :SYSTem:TERMination?

**Response**
- `<numeric>` = `<NR1 NUMERIC RESPONSE DATA>`
  - 0: LF+EOI
  - 1: CR+LF+EOI

**Function**
Queries the terminator type of the response data.

**Example**
```
> :SYSTem:TERMination?
< 0
```

**Compatibility**
Compatible with the MP1632C.
7.1 Common Commands

:SYSTem:CONDition:UNITs?

Response  

```
<unit1>,...,<unit4>,<slot1>,...,<slot6>  
= "<mainframe1>,...,<mainframe4>,<module1>,...,<module64>"

<mainframe1> to <mainframe4> = <STRING RESPONSE DATA>
XXXXXXX  Mainframe name (e.g.,: MP1800A)
See Table 7.1.1-2 "Option character correspondence table".

Note:
NONE is output for mainframe2 to mainframe4, if no mainframe is connected.

<module1> to <module64> = <STRING RESPONSE DATA>
XXXXXXX  Module name (e.g.,: MU181020A)
See Table 7.1.1-2 "Option character correspondence table".

Note:
NONE is output if no module is installed.
For a module that uses two slots, only the slot with the greater number is valid.
```

Function
Queries the model name of the mainframe and module.

Example
```
> :SYSTem:CONDition:UNITs?
< MT1810A,MT1810A,MT1810A,NONE,MU181000A,NONE,MU181020A,
  MU181040A,NONE,...,MU181600A,MU181601A,MU181620A,NONE,
  MU181640A,NONE,...,NONE
```

Example
To query the model name of the mainframe and module for the MP1800A (1-box type):
```
> :SYSTem:CONDition:UNITs?
< MP1800A,NONE,NONE,NONE,MU181000A,NONE,MU181020A,
  MU181040A,NONE,MU181600A,NONE,...,NONE
```

Compatibility
Incompatible with existing models.
**Chapter 7  SCPI Commands**

**:SYSTem:UNIT? <numeric>**

Parameter

- `<numeric> = <NR1 NUMERIC PROGRAM DATA>`
  - 1 to 4  Mainframe number
  - "1" for the MP1800A, "1 to 4" for the MT1810A.

Response

- "<mainframe>",
  - `<mainframe> = <unit>,<serial>,<mver>,<hver>,<opt1>,<sbver>,<saver>,<opt2>`
  - `<unit> = <STRING RESPONSE DATA>`
    - XXXXXXXXXX  Mainframe name (e.g., : MP1800A/MT1810A)
    - See Table 7.1.1-2 "Option character correspondence table".
  - `<serial> = <STRING RESPONSE DATA>`
    - XXXXXXXXXX  0000000000 to 9999999999
    - Mainframe serial number
  - `<mver> = <STRING RESPONSE DATA>`
    - XXXX.XX.XX  1.00.00 to 9999.99.99
    - Main application software version
  - `<hver> = <STRING RESPONSE DATA>`
    - XXXX.XX.XX  1.00.00 to 9999.99.99
    - Mainframe hardware version
  - `<opt1> = <STRING RESPONSE DATA>`
    - OPTXXX  Option number (MP1800A/MT1810A)
    - See Table 7.1.1-2 "Option character correspondence table".

*Note:*

- NONE is output if no module is installed.
- For a unit that uses two slots, only the slot with the lower number is valid.

- `<sbver> = <STRING RESPONSE DATA>`
  - XXXX.XX.XX  1.00.00 to 9999.99.99
  - Sub application software version (Boot part)
- `<saver> = <STRING RESPONSE DATA>`
  - XXXX.XX.XX  1.00.00 to 9999.99.99
  - Sub application software version (Application part)
### 7.1 Common Commands

<table>
<thead>
<tr>
<th>Function</th>
<th>Queries the mainframe information including model and serial number.</th>
</tr>
</thead>
</table>
| Example  | > :SYSTem:UNIT?1 <MP1800A,6201234568,1.00.00,1.00.00,OPT301,1.00.00,1.00.00>
| Compatibility | Incompatible with existing models. |

### :SYSTem:MODule? <numeric>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>1 to 6 Slot 1 to 6 when using the MP1800A, 1 to 4 when using the MT1810A</th>
</tr>
</thead>
<tbody>
<tr>
<td>Response</td>
<td>&lt;slot&gt; = &lt;module&gt;,&lt;serial&gt;,&lt;fpga1&gt;,&lt;fpga2&gt;,&lt;boot&gt;,&lt;application&gt;,&lt;opt&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;module&gt; = &lt;STRING RESPONSE DATA&gt; XXXXXXXXXX Module name (e.g., MU181020A) See Table 7.1.1-2 &quot;Option character correspondence table&quot;.</td>
</tr>
<tr>
<td>Note:</td>
<td>NONE is output if no module is installed. For a module that uses two slots, only the slot with the greater number is valid.</td>
</tr>
<tr>
<td></td>
<td>&lt;serial&gt; = &lt;STRING RESPONSE DATA&gt; XXXXXXXXXX 0000000000 to 9999999999 Serial number</td>
</tr>
<tr>
<td>Note:</td>
<td>&quot;--------&quot; is output if no module is installed. For a module that uses two slots, only the slot with the greater number is valid.</td>
</tr>
<tr>
<td></td>
<td>&lt;fpga1&gt;,&lt;fpga2&gt; = &lt;STRING RESPONSE DATA&gt; XXXX.XX.XX 1.00.00 to 9999.99.99 FPGA version</td>
</tr>
<tr>
<td></td>
<td>&lt;boot&gt; = &lt;STRING RESPONSE DATA&gt; XXXX.XX.XX 1.00.00 to 9999.99.99 Logic Boot version</td>
</tr>
<tr>
<td>Note:</td>
<td>&quot;--------&quot; is output if Logic Boot is not installed. For a module that uses two slots, only the slot with the greater number is valid.</td>
</tr>
</tbody>
</table>
Chapter 7  SCPI Commands

<application> = <STRING RESPONSE DATA>
XXXX.XX.XX  1.00.00 to 9999.99.99
Logic Application version

Note:
"---------" is output if Logic Application is not installed.
For a module that uses two slots, only the slot with the greater
number is valid.

<opt> = <STRING RESPONSE DATA>
XXXXXXXX/XXXXX  Option number
OPTXXX: For MP1800A/MT1810A

Note:
Outputs the numbers for all installed options.
NONE is output if no option is installed.
For a module that uses two slots, only the slot with the greater
number is valid.
The option names of the MU181020B-003 and MU181040B-003 are
not output.

Function
Queries the module information on the specified slot.

Example
To query the module information on Slot 3:
> :SYSTem:MODule? 3
<
MU181020A,6201234568,1.00.00,---------,1.00.00,1.00.00,OPT002,OPT211,OPT220,

Compatibility
Incompatible with existing models.

:SYSTem:DISPlay:RESult <boolean>

Parameter  <boolean> = <BOOLEAN PROGRAM DATA>
OFF or 0  Drawing Processing OFF
ON or 1  Drawing Processing ON (default)

Function
Sets measured results drawing processing ON/OFF

Example
Set measured results drawing processing to OFF
> :SYSTem:DISPlay:RESult OFF

Compatibility
Incompatible with existing models.

Remarks
When measured results drawing processing is set to OFF, a dialog
indicating drawing processing is stopped is displayed. Issue the
command to set drawing processing to ON or press the Remote button to
restart measured results drawing processing.
7.1 Common Commands

:SYSTem:DISPlay:RESult?

Response
- `<numeric> = <NR1 NUMERIC RESPONSE DATA>`
  - 0: Drawing Processing OFF
  - 1: Drawing Processing ON (default)

Function
Sets drawing processing ON/OFF

Example
```plaintext
> :SYSTem:DISPlay:RESult?
< 0
```

Compatibility
Incompatible with existing models.

:DISPlay:ACTive <unit>,<slot>[,<tab>]

Parameter
- `<unit> = <DECIMAL NUMERIC PROGRAM DATA>`
  - 1 to 4: Mainframe No.1 to 4
    - When installing the MT1810A, the mainframe No. is specified.
    - When using the MP1800A, No.1 is fixed.
- `<slot> = <DECIMAL NUMERIC PROGRAM DATA>`
  - 1 to 6: Slot No.1 to 6
    - When using the MP1800A, the slot No. is 1 to 6.
    - When using the MT1810A, the slot No. is 1 to 4.
- `<tab> = <DECIMAL NUMERIC PROGRAM DATA>`
  - 1 to X: Tab ID No.1 to X
    - When [<tab>] is omitted, 1 is specified.
    - Tab ID is set to No.1 at the left side dialog, and then the tab ID No. is set to 2, 3, 4…toward the right side.
    - The maximum number (X) of the tab ID varies depending on the module options.

Function
Displays the specified module screen to the front.

**Note:**
When the screen processing for measurement result is Off, this function cannot be used. If using this function, set the screen processing to On using the :SYSTem:DISPlay:RESult.

Example
To display the Pattern tab of the MU181020A module:
(when installing the MU181020A in the unit1 slot1)
```plaintext
> :DISPlay:ACTive 1,1,2
```

Compatibility
Incompatible with existing models.
### 7.1.2 Command related to File menu

#### Table 7.1.2-1  File menu setting commands

<table>
<thead>
<tr>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quick Open</td>
<td>:SYSTem:MMEMory:QRECall</td>
</tr>
<tr>
<td>Quick Save</td>
<td>:SYSTem:MMEMory:QSTorE</td>
</tr>
<tr>
<td>Open</td>
<td>:SYSTem:MMEMory:RECall</td>
</tr>
<tr>
<td>Save</td>
<td>:SYSTem:MMEMory:STORe</td>
</tr>
<tr>
<td>Screen Copy Execute</td>
<td>:SYSTem:PRINt:COPY</td>
</tr>
<tr>
<td>Screen Copy Setup</td>
<td>:SYSTem:PRINt:COPYsetup:SET</td>
</tr>
<tr>
<td></td>
<td>:SYSTem:PRINt:COPYsetup:SET?</td>
</tr>
<tr>
<td>Print</td>
<td>:SYSTem:PRINt:EALarm</td>
</tr>
<tr>
<td>Printer Setup</td>
<td>:SYSTem:PRINt:PRINter:SET</td>
</tr>
<tr>
<td></td>
<td>:SYSTem:PRINt:PRINter:SET?</td>
</tr>
<tr>
<td>Combination Setting</td>
<td>:SYSTem:CONDition:COMBination?</td>
</tr>
<tr>
<td></td>
<td>:COMBination:OPERation:SETTing</td>
</tr>
<tr>
<td></td>
<td>:COMBination:OPERation:SETTing?</td>
</tr>
<tr>
<td></td>
<td>:COMBination:OPERation:ABILity:CHSYnc?</td>
</tr>
<tr>
<td></td>
<td>:COMBination:OPERation:CHSetting</td>
</tr>
<tr>
<td></td>
<td>:COMBination:OPERation:ABILity:MUXChsync?</td>
</tr>
<tr>
<td></td>
<td>:COMBination:OPERation:MUXChsync</td>
</tr>
<tr>
<td></td>
<td>:COMBination:OPERation:MUXChsync?</td>
</tr>
<tr>
<td></td>
<td>:COMBination:OPERation:ABILity:COMBination?</td>
</tr>
<tr>
<td></td>
<td>:COMBination:OPERation:USYNcout</td>
</tr>
<tr>
<td></td>
<td>:COMBination:OPERation:USYNcout?</td>
</tr>
<tr>
<td>Initialize</td>
<td>:SYSTem:MEMory:INITialize</td>
</tr>
</tbody>
</table>
7.1 Common Commands

:SYSTem:MMEMory:QRECall <file_name>

Parameter
- <file_name> = <STRING PROGRAM DATA>
- "<drv>:\<dir>\<file>"
- <drv> = C, D, E, F
- <dir> = <dir1>\<dir2>\... (Omitted for the root directory)
- <file> = File name

Function
Opens all setting data.

Example
To read all setting files from the specified save destination.
> :SYSTem:MMEMory:QRECall "C:\Test\example"

Compatibility
Commands are compatible with the MP1632C.
Parameters are incompatible.

:SYSTem:MMEMory:QSTore <file_name>,<comment>

Parameter
- <file_name> = <STRING PROGRAM DATA>
- "<drv>:\<dir>\<file>"
- <drv> = C, D, E, F
- <dir> = <dir1>\<dir2>\... (Omitted for the root directory)
- <file> = File name
- <comment> = <STRING PROGRAM DATA>
- "XXXXXXX..." Specify a comment of a character string within 60 characters into the file.

Function
Executes "Quick Save".

Note:
The settings will not be read from the saved file if the file name is changed.

Example
To specify save destination for all setting files and save them with a comment and measurement result data:
> :SYSTem:MMEMory:QSTore "C:\Test\example", "setup all"

Compatibility
Commands are compatible with the MP1632C.
Parameters are incompatible.
### :SYSTem:MMEMory:RECall <file_name>,<module>,<data_type>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;file_name&gt;</td>
<td>= &lt;STRING PROGRAM DATA&gt; &quot;&lt;drv&gt;[&lt;dir&gt;]&lt;file&gt;&quot;</td>
</tr>
<tr>
<td>&lt;drv&gt;</td>
<td>= C, D, E, F</td>
</tr>
<tr>
<td>&lt;dir&gt;</td>
<td>= &lt;dir1&gt;&lt;dir2&gt;... (Omitted for the root directory)</td>
</tr>
<tr>
<td>&lt;file&gt;</td>
<td>= File name</td>
</tr>
<tr>
<td>&lt;module&gt;</td>
<td>= &lt;STRING PROGRAM DATA&gt; &quot;&lt;unit&gt;:&lt;slot&gt;:&lt;port&gt;:&lt;module&gt;&quot;</td>
</tr>
<tr>
<td>&lt;unit&gt;</td>
<td>= 1, 2, 3, 4</td>
</tr>
<tr>
<td>&lt;slot&gt;</td>
<td>= 1, 2, 3, 4, ..., 6</td>
</tr>
<tr>
<td>&lt;port&gt;</td>
<td>= 1</td>
</tr>
<tr>
<td>&lt;module&gt;</td>
<td>= Module model name</td>
</tr>
<tr>
<td>&lt;data_type&gt;</td>
<td>= &lt;CHARACTER PROGRAM DATA&gt;</td>
</tr>
<tr>
<td>PPG</td>
<td>PPG Setup</td>
</tr>
<tr>
<td>ED</td>
<td>ED Setup</td>
</tr>
<tr>
<td>SYN</td>
<td>Synthesizer Setup</td>
</tr>
<tr>
<td>XFP</td>
<td>XFP Setup</td>
</tr>
<tr>
<td>SFP</td>
<td>SFP Setup</td>
</tr>
<tr>
<td>CAP</td>
<td>Capture Result</td>
</tr>
<tr>
<td>OTX</td>
<td>Transmitter Setup</td>
</tr>
<tr>
<td>MX1</td>
<td>MUX Setup</td>
</tr>
<tr>
<td>MX2</td>
<td>MUX Setup</td>
</tr>
<tr>
<td>DX1</td>
<td>DEMUX Setup</td>
</tr>
<tr>
<td>DX2</td>
<td>DEMUX Setup</td>
</tr>
</tbody>
</table>

**Function**

Opens module settings and result data.

**Example**

To read a file for the ED module:

> :SYSTem:MMEMory:RECall "C:\Test\example", "1:3:1 MU181040A", ED

**Compatibility**

Compatible with the MP1632C1 and MP1776A commands. Parameters are incompatible.
### 7.1 Common Commands

#### :SYSTem:MMEMory:STORe

<file_name>,<module>,<data_type>,<file_type>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;file_name&gt;</td>
<td>= &lt;STRING PROGRAM DATA&gt; &quot;&lt;drv&gt;:&lt;dir&gt;&lt;file&gt;&quot;</td>
</tr>
<tr>
<td>&lt;drv&gt;</td>
<td>= C, D, E, F</td>
</tr>
<tr>
<td>&lt;dir&gt;</td>
<td>= &lt;dir1&gt;&lt;dir2&gt;... (Omitted for the root directory)</td>
</tr>
<tr>
<td>&lt;file&gt;</td>
<td>= File name</td>
</tr>
<tr>
<td>&lt;module&gt;</td>
<td>= &lt;STRING PROGRAM DATA&gt; &quot;&lt;unit&gt;:&lt;slot&gt;:&lt;port&gt;:&lt;module&gt;&quot;</td>
</tr>
<tr>
<td>&lt;unit&gt;</td>
<td>= 1, 2, 3, 4</td>
</tr>
<tr>
<td>&lt;slot&gt;</td>
<td>= 1, 2, 3, 4, ..., 6</td>
</tr>
<tr>
<td>&lt;port&gt;</td>
<td>= 1</td>
</tr>
<tr>
<td>&lt;module&gt;</td>
<td>= Module model name</td>
</tr>
<tr>
<td>&lt;data_type&gt;</td>
<td>= &lt;CHARACTER PROGRAM DATA&gt;</td>
</tr>
<tr>
<td>PPG</td>
<td>PPG Setup</td>
</tr>
<tr>
<td>ED</td>
<td>ED Setup</td>
</tr>
<tr>
<td>SYN</td>
<td>Synthesizer Setup</td>
</tr>
<tr>
<td>XFP</td>
<td>XFP Setup</td>
</tr>
<tr>
<td>SFP</td>
<td>SFP Setup</td>
</tr>
<tr>
<td>ERA</td>
<td>Error/Alarm Result</td>
</tr>
<tr>
<td>HST</td>
<td>Histogram Result</td>
</tr>
<tr>
<td>CAP</td>
<td>Capture Result</td>
</tr>
<tr>
<td>LOG</td>
<td>Logging Result</td>
</tr>
<tr>
<td>OTX</td>
<td>Transmitter Setup</td>
</tr>
<tr>
<td>MX1</td>
<td>MUX Setup</td>
</tr>
<tr>
<td>MX2</td>
<td>MUX Setup</td>
</tr>
<tr>
<td>DX1</td>
<td>DEMUX Setup</td>
</tr>
<tr>
<td>DX2</td>
<td>DEMUX Setup</td>
</tr>
<tr>
<td>DER</td>
<td>DEMUX Error/Alarm Result</td>
</tr>
<tr>
<td>&lt;file_type&gt;</td>
<td>= &lt;CHARACTER PROGRAM DATA&gt;</td>
</tr>
<tr>
<td>BIN</td>
<td>Binary File</td>
</tr>
<tr>
<td>CSV</td>
<td>CSV File</td>
</tr>
<tr>
<td>TXT</td>
<td>Text File</td>
</tr>
</tbody>
</table>

**Function:**
Saves the setting and measurement result data for a specified module.

**Note:**
The settings will not be read from the saved file if the file name is changed.

**Example**
To save the setting file for the ED module:
> :SYSTem:MMEMory:STORe "C:\Test\example", "1:3:1
MU181040A",ED,BIN

To save the capture result file:
> :SYSTem:MMEMory:STORe "C:\Test\example", "1:3:1
MU181040A",CAP,BIN
Compatibility
Compatible with the MP1632C and MP1776A commands. Parameters are incompatible.

:[SYSTem]:PRINt:COPY

Function
Takes a screen shot.

Example
> :SYSTem:PRINt:COPY

Compatibility
Compatible with the MP1632C and MP1776A.

:[SYSTem]:PRINt:COPYsetup:SET <save_type>,<output>,<save_to>

Parameter
<save_type> = <CHARACTER PROGRAM DATA>
BMP Bitmap format
PNG PNG format
JPG JPG format
<output> = <CHARACTER PROGRAM DATA>
FILE Output to a file.
PRINter Output to the printer.
<save_to> = <STRING PROGRAM DATA>
"<drv>:\<dir>" 
<drv> = C, D, E, F
<dir> = <dir1>\<dir2>...(Omitted for the root directory)

Function
Sets the extension, output destination, and save destination of the screen shot.

Example
To output the screen shot to the specified file in the bitmap format:
> :SYSTem:PRINt:COPYsetup:SET BMP,FILE, "C:\Test\example"

Compatibility
Incompatible with existing models.
### 7.1 Common Commands

**:SYSTem:PRINT:COPYsetup:SET?**

**Response**

<table>
<thead>
<tr>
<th>&lt;save_type&gt; = &lt;CHARACTER RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMP               Bitmap format</td>
</tr>
<tr>
<td>PNG               PNG format</td>
</tr>
<tr>
<td>JPG               JPG format</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>&lt;output&gt; = &lt;CHARACTER RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>FILE             Output to a file.</td>
</tr>
</tbody>
</table>

| PRINTER          Output to the printer. |

<table>
<thead>
<tr>
<th>&lt;save_to&gt; = &lt;STRING RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;&lt;drv&gt;:[&lt;dir&gt;]&quot;</td>
</tr>
<tr>
<td>&lt;drv&gt; = C, D, E, F</td>
</tr>
<tr>
<td>&lt;dir&gt; = &lt;dir1&gt;&lt;dir2&gt;...(Omitted for the root directory)</td>
</tr>
</tbody>
</table>

**Function**

Queries the extension, output destination, and save destination settings of the screen shot.

**Example**

```plaintext
> :SYSTem:PRINT:COPYsetup:SET?
< BMP,FILE, "C:\Test\example"
```

**Compatibility**

Incompatible with existing models.

---

**:SYSTem:PRINT:EALarm <list>,<module>**

**Response**

<table>
<thead>
<tr>
<th>&lt;list&gt; = &lt;CHARACTER PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>EAR               Error/Alarm Result</td>
</tr>
<tr>
<td>HST               Histogram Result</td>
</tr>
<tr>
<td>LOG               Logging Result</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>&lt;module&gt; = &lt;STRING PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;&lt;unit&gt;::&lt;slot&gt;::&lt;port&gt;::&lt;module&gt;&quot;</td>
</tr>
<tr>
<td>&lt;unit&gt; = 1, 2, 3, 4</td>
</tr>
<tr>
<td>&lt;slot&gt; = 1, 2, 3, 4, ..., 6</td>
</tr>
<tr>
<td>&lt;port&gt; = 1, 2, ...</td>
</tr>
<tr>
<td>&lt;module&gt; = Module model name</td>
</tr>
</tbody>
</table>

**Function**

Prints the Error/Alarm, histogram, or logging measurement results. The printing range for the histogram and logging measurement results can be specified by the time.

**Example**

To print the histogram measurement result in the specified range:

```plaintext
> :SYSTem:PRINT:EALarm HST
```

**Compatibility**

Compatible with the MP1632C and MP1776A commands. Parameters are incompatible.
### :SYSTem:PRINt:PRINter:SET <printer>,<direction>

**Parameter**
- `<printer>` = `<STRING PROGRAM DATA>`
  - "XXXXXX..." Printer name
- `<direction>` = `<CHARACTER PROGRAM DATA>`
  - `VER` Vertical
  - `HOR` Horizontal

**Function**
Sets the printer and print direction.

**Example**
```plaintext
> :SYSTem:PRINt:PRINter:SET "printer",VER
```

**Compatibility**
Incompatible with existing models.

### :SYSTem:PRINt:PRINter:SET?

**Response**
- `<printer>` = `<STRING RESPONSE DATA>`
  - "XXXXXX..." Printer name
- `<direction>` = `<CHARACTER RESPONSE DATA>`
  - `VER` Vertical
  - `HOR` Horizontal

**Function**
Queries the default printer and print direction.

**Example**
```plaintext
> :SYSTem:PRINt:PRINter:SET?
< "printer",VER
```

**Compatibility**
Incompatible with existing models.
### 7.1 Common Commands

#### :SYSTem:CONDition:COMBination?

**Parameter**

```
"<slot>,<module>,<status>" = <STRING RESPONSE DATA>

<slot> = <STRING RESPONSE DATA>
1 to 6   Slot number
XX-XX   Slot numbers for which combined modules are inserted

Example: 1-4: 4-CH combination

<module> = <STRING RESPONSE DATA>
XXXXXXXXX   Module model name   Example: MU181020A

<status> = <STRING RESPONSE DATA>
1   Independent
2   2-CH combination
4   4-CH combination
0   Invalid PPG or ED module
```

**Function**

Queries the combination status of the modules inserted into the mainframe slots and the valid/invalid status of a specified PPG or ED module.

**Example**

To query the combination status of the modules inserted into the MP1800A, in the configuration where six PPG modules are inserted and 4-CH combination is configured:

```
> :SYSTem:CONDition:COMBination?
"1-4,MU181020A,4", "5,MU181020A,0", "6,MU181020A,0"
```

To query the combination status of the modules inserted into the MP1800A, in the configuration where three PPG modules are inserted:

```
> :SYSTem:CONDition:COMBination?
"1,MU181020A,1", "5,MU181020A,1", "6,MU181020A,1"
```

**Compatibility**

Incompatible with existing models.
**Chapter 7  SCPI Commands**

**:COMBination:OPERation:SETTing <configuration>[,<unit>]**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;configuration&gt;</td>
<td>&lt;NR1 NUMERIC RESPONSE DATA&gt;</td>
</tr>
<tr>
<td>0</td>
<td>Independent</td>
</tr>
<tr>
<td>21</td>
<td>2 Ch PPG Combination</td>
</tr>
<tr>
<td>22</td>
<td>2 Ch ED Combination</td>
</tr>
<tr>
<td>23</td>
<td>2 Ch PPG/ED Combination</td>
</tr>
<tr>
<td>41</td>
<td>4 Ch PPG Combination</td>
</tr>
<tr>
<td>42</td>
<td>4 Ch ED Combination</td>
</tr>
<tr>
<td>71</td>
<td>25G × 2 Ch PPG Combination</td>
</tr>
<tr>
<td>72</td>
<td>25G × 2 Ch ED Combination</td>
</tr>
<tr>
<td>&lt;unit&gt;</td>
<td>&lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
</tr>
<tr>
<td>1 to 4</td>
<td>Mainframe Nos. 1 to 4</td>
</tr>
</tbody>
</table>

When using two or more MT1810A units in serial connection, specify the mainframe number.

Can be omitted. Mainframe No. 1 is specified when omitted.

**Function**

Sets the combination setting for the specified unit, from Combination or Independent.

**Example**

To set the combination setting of Unit 3 to 4-ch PPG combination:

```plaintext
> :COMBination:OPERation:SETTing 41,3
```

**Compatibility**

Incompatible with existing models.
### 7.1 Common Commands

**:COMBination:OPERation:SETTing? <slot>[,<unit>]**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;slot&gt;</td>
<td>= &lt;CHARACTER PROGRAM DATA&gt;</td>
</tr>
<tr>
<td>SLOT1 to SLOT6</td>
<td>Slot No.1 to 6</td>
</tr>
<tr>
<td>ALL</td>
<td>All modules (units)</td>
</tr>
<tr>
<td>[&lt;unit&gt;]</td>
<td>= &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
</tr>
<tr>
<td>1 to 4</td>
<td>Mainframe Nos. 1 to 4</td>
</tr>
<tr>
<td></td>
<td>When using two or more MT1810A units in serial connection, specify the</td>
</tr>
<tr>
<td></td>
<td>mainframe number.</td>
</tr>
<tr>
<td></td>
<td>Can be omitted. Mainframe No. 1 is specified when omitted.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Response</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;numeric&gt;</td>
<td>= &lt;NR1 NUMERIC RESPONSE DATA&gt;</td>
</tr>
<tr>
<td>0</td>
<td>Independent</td>
</tr>
<tr>
<td>1</td>
<td>Channel Synchronization</td>
</tr>
<tr>
<td>21</td>
<td>2 Ch PPG Combination</td>
</tr>
<tr>
<td>22</td>
<td>2 Ch ED Combination</td>
</tr>
<tr>
<td>23</td>
<td>2 Ch PPG/ED Combination</td>
</tr>
<tr>
<td>41</td>
<td>4 Ch PPG Combination</td>
</tr>
<tr>
<td>42</td>
<td>4 Ch ED Combination</td>
</tr>
<tr>
<td>71</td>
<td>25G × 2 Ch PPG Combination</td>
</tr>
<tr>
<td>72</td>
<td>25G × 2 Ch ED Combination</td>
</tr>
</tbody>
</table>

**Function**
Queries the combination setting of the specified slot.

**Example**
To query the combination setting of Unit 1:

```plaintext
> :COMBination:OPERation:SETTing? ALL
< 0
```

To query the combination setting of Slot 3 in Unit 2:

```plaintext
> :COMBination:OPERation:SETTing? SLOT3,2
< 41
```

To query the combination setting of Unit 3:

```plaintext
> :COMBination:OPERation:SETTing? ALL,3
< 0
```

**Compatibility**
Incompatible with existing models.
Chapter 7  SCPI Commands

:COMBination:OPERation:ABILity:CHSYnc? [<unit>]

Parameter  

[<unit>] = <DECIMAL NUMERIC PROGRAM DATA>

1 to 4  
Mainframe Nos. 1 to 4

When using two or more MT1810A units in serial connection, specify the mainframe number.

Can be omitted. Mainframe No. 1 is specified when omitted.

Response  

<numeric> = <NR1 NUMERIC RESPONSE DATA>

0 to 64  
Total number of PPGs that can configure channel synchronization (decimal)

Available bits:

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>No PPG that can configure channel synchronization</td>
</tr>
<tr>
<td>1</td>
<td>PPG in Slot 1</td>
</tr>
<tr>
<td>2</td>
<td>PPG in Slot 2</td>
</tr>
<tr>
<td>4</td>
<td>PPG in Slot 3</td>
</tr>
<tr>
<td>8</td>
<td>PPG in Slot 4</td>
</tr>
<tr>
<td>16</td>
<td>PPG in Slot 5</td>
</tr>
<tr>
<td>32</td>
<td>PPG in Slot 6</td>
</tr>
</tbody>
</table>

Function  

Queries the slot where PPG that can configure channel synchronization combination is inserted.

Example  

To query the slot in Unit 3 where PPG that can configure channel synchronization is inserted:

> :COMBination:OPERation:ABILity:CHSYnc? 3
< 7

Compatibility  

Incompatible with existing models.
7.1 Common Commands

**:COMBination:OPERation:CHSetting <configuration>[,<unit>]**

Parameter

- `<configuration>` = `<NR1 NUMERIC RESPONSE DATA>`
  - 0 to 64: Total number of PPGs that can configure channel synchronization (decimal)

Available bits:
- 2 (Bit 1): PPG in Slot 2
- 4 (Bit 2): PPG in Slot 3
- 8 (Bit 3): PPG in Slot 4
- 16 (Bit 4): PPG in Slot 5
- 32 (Bit 5): PPG in Slot 6

- `<unit>` = `<DECIMAL NUMERIC PROGRAM DATA>`
  - 1 to 4: Mainframe Nos. 1 to 4

Function

Specify the slot where the PPG for which channel synchronization is to be set is inserted.

Example

To set channel synchronization for the PPGs in Slots 1 through 4 of Unit 3:

> :COMBination:OPERation:CHSetting 14

Compatibility

Incompatible with existing models.

**:COMBination:OPERation:ABILity:MUXChsync?**

Response

- `<numeric>` = `<NR1 NUMERIC RESPONSE DATA>`
  - 0: 25G Channel Synchronization disabled
  - 1: 25G Channel Synchronization enabled

Function

Queries whether 25G Channel Synchronization setting supported

Example

> :COMBination:OPERation:ABILity:MUXChsync?

< 1

Compatibility

Incompatible with existing models.

**:COMBination:OPERation:MUXChsync**

Function

Sets 25G Channel Synchronization

Example

> :COMBination:OPERation:MUXChsync

Compatibility

Incompatible with existing models.
Chapter 7  SCPI Commands

:COMBination:OPERation:MUXChsync?

Response  
<boolean> = <NR1 NUMERIC RESPONSE DATA>
1  25G Channel Synchronization
0  Non-25G Channel Synchronization

Function Queries whether 25G Channel Synchronization configured

Example  
> :COMBination:OPERation:MUXChsync?
< 1

Compatibility  
Incompatible with existing models.

:COMBination:OPERation:ABILITY:COMBination? [<unit>]

Parameter  
[<unit>] = <DECIMAL NUMERIC PROGRAM DATA>
1 to 4  Mainframe Nos. 1 to 4
When using two or more MT1810A units in serial connection, specify the mainframe number.
Can be omitted. Mainframe No. 1 is specified when omitted.

Response  
<numeric> = <NR1 NUMERIC RESPONSE DATA>
0  Independent
21  2 Ch PPG Combination
22  2 Ch ED Combination
23  2 Ch PPG/ED Combination
41  4 Ch PPG Combination
42  4 Ch ED Combination
71  25G × 2 Ch PPG Combination
72  25G × 2 Ch ED Combination

Function Queries the available combination configuration.

Example  
To query the combination configuration available for Unit 1:
> :COMBination:OPERation:ABILITY:COMBination?
< 41

Compatibility  
Incompatible with existing models.

:COMBination:OPERation:USYNcout <boolean>

Parameter  
<boolean> = <BOOLEAN PROGRAM DATA>
0  Unit Sync OFF
1  Unit Sync ON

Function Sets the Unit Sync On/Off

Example  
To set the Unit Sync to On
> :COMBination:OPERation:USYNcout 1

Compatibility  
Incompatible with existing models.
7.1 Common Commands

:COMBination:OPERation:USYNcout?

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;numeric&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>1</td>
</tr>
</tbody>
</table>

**Function**
To query the Unit Sync On/Off

**Example**
> :COMBination:OPERation:USYNcout?
< 1

**Compatibility**
Incompatible with existing models.

:SYSTem:MEMory:INITialize

**Function**
Initializes the internal setting data to the initial settings at factory shipment.

**Example**
> :SYSTem:MEMory:INITialize

**Compatibility**
Compatible with the MP1632C and MP1776A.
7.1.3 Common function button

Table 7.1.3-1 Common function button setting commands

<table>
<thead>
<tr>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>Output ON/OFF</td>
<td>:SOURce:OUTPut:ASET</td>
</tr>
<tr>
<td></td>
<td>:SOURce:OUTPut:ASET?</td>
</tr>
<tr>
<td>Error Addition ON/OFF</td>
<td>:SOURce:PATTern:EADDition:ASET</td>
</tr>
<tr>
<td></td>
<td>:SOURce:PATTern:EADDition:ASET?</td>
</tr>
<tr>
<td>Single Error Addition</td>
<td>:SOURce:PATTern:EADDition:ASINgle</td>
</tr>
<tr>
<td>Meas. Start</td>
<td>:SENSe:MEASure:ASTRt</td>
</tr>
<tr>
<td>Meas. Stop</td>
<td>:SENSe:MEASure:ASTP</td>
</tr>
<tr>
<td>(Query for measurement status)</td>
<td>:SENSe:MEASure:ASTate?</td>
</tr>
<tr>
<td>Open</td>
<td>Refer to Chapter 7.1.2 &quot;Command related to File menu.&quot;</td>
</tr>
<tr>
<td>Save</td>
<td>Refer to Chapter 7.1.2 &quot;Command related to File menu.&quot;</td>
</tr>
<tr>
<td>Error ON/OFF</td>
<td>:SYSTem:BEEPer:ERRor:SET</td>
</tr>
<tr>
<td></td>
<td>:SYSTem:BEEPer:ERRor:SET?</td>
</tr>
<tr>
<td>Alarm ON/OFF</td>
<td>:SYSTem:BEEPer:ALARm:SET</td>
</tr>
<tr>
<td></td>
<td>:SYSTem:BEEPer:ALARm:SET?</td>
</tr>
<tr>
<td>System Error ON/OFF</td>
<td>:SYSTem:BEEPer:SYSTem:SET</td>
</tr>
<tr>
<td></td>
<td>:SYSTem:BEEPer:SYSTem:SET?</td>
</tr>
<tr>
<td>System Error</td>
<td>:SYSTem:BEEPer:SYSTem:TYPE</td>
</tr>
<tr>
<td></td>
<td>:SYSTem:BEEPer:SYSTem:TYPE?</td>
</tr>
<tr>
<td>Combination Setting</td>
<td>Refer to Chapter 7.1.2 &quot;Command related to File menu.&quot;</td>
</tr>
<tr>
<td>Unit Sync Output</td>
<td>:SOURce:PATTern:USYNcout</td>
</tr>
</tbody>
</table>

: SOURce:OUTPut:ASET <boolean>

Parameter: <boolean> = <BOOLEAN PROGRAM DATA>
OFF or 0        Output OFF
ON or 1         Output ON

Function: Sets Data and Clock outputs of optical output and PPG ON or OFF.

Example: To set Data and Clock outputs of PPG to ON:
> :SOURce:OUTPut:ASET ON

Compatibility: Incompatible with existing models.
### 7.1 Common Commands

#### :SOURce:OUTPut:ASET?

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;numeric&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Output OFF</td>
</tr>
<tr>
<td>1</td>
<td>Output ON</td>
</tr>
</tbody>
</table>

**Function**
Queries the ON/OFF state for Data and Clock outputs of optical output and PPG.

**Example**

```
> :SOURce:OUTPut:ASET?
< 1
```

**Compatibility**
Incompatible with existing models.

#### :SOURce:PATTern:EADDition:ASET <boolean>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;boolean&gt; = &lt;BOOLEAN PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>OFF or 0</td>
<td>Error addition OFF</td>
</tr>
<tr>
<td>ON or 1</td>
<td>Error addition ON</td>
</tr>
</tbody>
</table>

**Function**
Sets error addition for all valid modules ON/OFF.

**Example**

```
> :SOURce:PATTern:EADDition:ASET ON
```

**Compatibility**
Incompatible with existing models.

#### :SOURce:PATTern:EADDition:ASET?

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;numeric&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Error addition OFF</td>
</tr>
<tr>
<td>1</td>
<td>Error addition ON</td>
</tr>
</tbody>
</table>

**Function**
Queries the error addition ON/OFF state for all valid modules.

**Example**

```
> :SOURce:PATTern:EADDition:ASET?
< 1
```

**Compatibility**
Incompatible with existing models.

#### :SOURce:PATTern:EADDition:ASINgle

**Function**
Adds a single error for all valid modules.

**Example**

```
> :SOURce:PATTern:EADDition:ASINgle
```

**Compatibility**
Incompatible with existing models.

#### :SENSe:MEASure:ASTRt

**Function**
Starts measurement for all modules.

**Example**

```
> :SENSe:MEASure:ASTRt
```

**Compatibility**
Incompatible with existing models.
**:SENSe:MEASure:ASTP**

**Function**
Stops measurement for all modules.

**Example**
> :SENSe:MEASure:ASTP

**Compatibility**
Incompatible with existing models.

**:SENSe:MEASure:ASTate?**

**Response**

<table>
<thead>
<tr>
<th>numeric</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Measurement stops for all modules.</td>
</tr>
<tr>
<td>1</td>
<td>During measurement</td>
</tr>
</tbody>
</table>

**Note:**
If any module is being measured, "1 (During measurement)" is returned.

**Function**
Queries the measurement state for all modules.

**Example**
> :SENSe:MEASure:ASTate?
< 0

**Compatibility**
Incompatible with existing models.

**:SYSTem:BEEPer:ERRor:SET <boolean>**

**Parameter**

<table>
<thead>
<tr>
<th>boolean</th>
<th>Parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>OFF or 0</td>
<td>Buzzer OFF</td>
</tr>
<tr>
<td>ON or 1</td>
<td>Buzzer ON</td>
</tr>
</tbody>
</table>

**Function**
Sets buzzer at error occurrence ON/OFF.

**Example**
To set buzzer at error occurrence ON:
> :SYSTem:BEEPer:ERRor:SET ON

**Compatibility**
Compatible with the MP1632C and MP1776A.

**:SYSTem:BEEPer:ERRor:SET?**

**Response**

<table>
<thead>
<tr>
<th>numeric</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Buzzer OFF</td>
</tr>
<tr>
<td>1</td>
<td>Buzzer ON</td>
</tr>
</tbody>
</table>

**Function**
Queries the buzzer ON/OFF state at error occurrence.

**Example**
To query the buzzer ON/OFF state at error occurrence:
> :SYSTem:BEEPer:ERRor:SET?
< 1

**Compatibility**
Compatible with the MP1632C and MP1776A.
7.1 Common Commands

:SYSTem:BEEPer:ALARm:SET <boolean>

Parameter

<boolean> = <BOOLEAN PROGRAM DATA>
OFF or 0    Buzzer OFF
ON or 1     Buzzer ON

Function
Sets buzzer at alarm occurrence ON/OFF.

Example
To set buzzer at alarm occurrence OFF:
> :SYSTem:BEEPer:ALARm:SET OFF

Compatibility
Compatible with the MP1632C and MP1776A.

:SYSTem:BEEPer:ALARm:SET?

Response

<numeric> = <NR1 NUMERIC RESPONSE DATA>
0    Buzzer OFF
1    Buzzer ON

Function
Queries the buzzer ON/OFF state at alarm occurrence.

Example
> :SYSTem:BEEPer:ALARm:SET?
< 0

Compatibility
Compatible with the MP1632C and MP1776A.

:SYSTem:BEEPer:SYSTem:SET <boolean>

Parameter

<boolean> = <BOOLEAN PROGRAM DATA>
OFF or 0    Buzzer OFF
ON or 1     Buzzer ON

Function
Sets buzzer at system error occurrence ON/OFF.

Example
To set buzzer at system error occurrence ON:
> :SYSTem:BEEPer:SYSTem:SET ON

Compatibility
Compatible with the MP1632C and MP1776A.

:SYSTem:BEEPer:SYSTem:SET?

Response

<numeric> = <NR1 NUMERIC RESPONSE DATA>
0    Buzzer OFF
1    Buzzer ON

Function
Queries the buzzer ON/OFF state at system error occurrence.

Example
> :SYSTem:BEEPer:SYSTem:SET?
< 1

Compatibility
Compatible with the MP1632C and MP1776A.
Chapter 7  SCPI Commands

:SYSTem:BEEPer:SYSTem:TYPE <type>,<boolean>

Parameter

- <type> = <CHARACTER PROGRAM DATA>
  - PUNLock: PLL unlock
  - FAN: FAN
  - TEMPerature: Temperature
  - ALL: Selects all system errors
- <boolean> = <BOOLEAN PROGRAM DATA>
  - OFF or 0
  - ON or 1

Function
Sets system error buzzer for the target item ON/OFF.

Example
To set system error buzzer for "Temperature" ON:
> :SYSTem:BEEPer:SYSTem:TYPE TEMPerature,ON

Compatibility
Partially compatible with the MP1632C.

:SYSTem:BEEPer:SYSTem:TYPE?

Response

- <type> = <CHARACTER RESPONSE DATA>
  - PUNL, FAN, TEMP, ALL
  - XXX,XXX, Errors for which buzzer is set to ON are delimited with commas (,) and returned.
  - NONE: Buzzer is set to OFF for all items.

Function
Queries the ON/OFF state of system error buzzer for target items.

Example
To query the ON/OFF state of system error buzzer for target items:
> :SYSTem:BEEPer:SYSTem:TYPE?
< PUNL, TEMP

Compatibility
Compatible with the MP1632C.

:SOURce:PATTern:USYNcout

Parameter
None

Function
Outputs the timing signal to synchronize the pattern between multiple devices.

Example
> :SOURce:PATTern:USYNcout

Compatibility
Incompatible with existing models.
7.1 Common Commands

7.1.4 Auto Search

This chapter describes the commands that are used after Auto Search is set by the :SYSTem:CFUNction command.

Figure 7.1.4-1  Auto Search setting screen

Table 7.1.4-1  Auto Search setting commands

<table>
<thead>
<tr>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mode</td>
<td>:SENSe:MEASure:ASEarch:SMODe</td>
</tr>
<tr>
<td></td>
<td>:SENSe:MEASure:ASEarch:SMODe?</td>
</tr>
<tr>
<td>Item</td>
<td>:SENSe:MEASure:ASEarch:MODE</td>
</tr>
<tr>
<td></td>
<td>:SENSe:MEASure:ASEarch:MODE?</td>
</tr>
<tr>
<td>Start</td>
<td>:SENSe:MEASure:ASEarch:STARt</td>
</tr>
<tr>
<td>Stop</td>
<td>:SENSe:MEASure:ASEarch:STOP</td>
</tr>
<tr>
<td>No label (Query for function state)</td>
<td>:SENSe:MEASure:ASEarch:STATe?</td>
</tr>
<tr>
<td>Set All</td>
<td>:SENSe:MEASure:ASEarch:SLASet</td>
</tr>
<tr>
<td>Reset All</td>
<td>:SENSe:MEASure:ASEarch:SLAReset</td>
</tr>
<tr>
<td>Slot Select</td>
<td>:SENSe:MEASure:ASEarch:SELSlot</td>
</tr>
<tr>
<td></td>
<td>:SENSe:MEASure:ASEarch:SELSlot?</td>
</tr>
<tr>
<td>No label (Query for slot state)</td>
<td>:SENSe:MEASure:ASEarch:SLOT?</td>
</tr>
<tr>
<td>No label (Query for Auto Search result)</td>
<td>:CALCulate:DATA:ASEarch?</td>
</tr>
</tbody>
</table>
Chapter 7  SCPI Commands

:SENSe:MEASure:ASEarch:SMODe <mode>
Parameter  "mode" = <CHARACTER PROGRAM DATA>
FINE       Fine mode
COARse     Coarse mode
Function   Sets the Auto Search mode.
Example    To set the Fine mode:
> :SENSe:MEASure:ASEarch:SMODe FINE
Compatibility Incompatible with existing models.

:SENSe:MEASure:ASEarch:SMODe?
Response  "mode" = <CHARACTER RESPONSE DATA>
FINE, COAR
Function   Queries the Auto Search mode.
Example    > :SENSe:MEASure:ASEarch:SMODe?
            < FINE
Compatibility Incompatible with existing models.

:SENSe:MEASure:ASEarch:MODE <mode>
Parameter  "mode" = <CHARACTER PROGRAM DATA>
PTHReshold Phase & Threshold
THReshold  Threshold
PHASe      Phase
Function   Sets the execution item of Auto Search.
Example    To set the execution item of Auto Search to Phase & Threshold:
> :SENSe:MEASure:ASEarch:MODE PTHReshold
Compatibility Compatible with the MP1632C.

:SENSe:MEASure:ASEarch:MODE?
Response  "mode" = <CHARACTER RESPONSE DATA>
PTHR      Phase & Threshold
THR       Threshold
PHAS      Phase
Function   Queries the execution item of Auto Search.
Example    > :SENSe:MEASure:ASEarch:MODE?
            < PTHR
Compatibility Compatible with the MP1632C.
### :SENSe:MEASure:ASEarch:STARt

**Function:** Starts the Auto Search function

**Example:**

```plaintext
> :SENSe:MEASure:ASEarch:STARt
```

**Compatibility:** Compatible with the MP1632C.

### :SENSe:MEASure:ASEarch:STOP

**Function:** Stops the Auto Search function.

**Example:**

```plaintext
> :SENSe:MEASure:ASEarch:STOP
```

**Compatibility:** Compatible with the MP1632C.

### :SENSe:MEASure:ASEarch:STATe?

**Response:**

```
<boolean> = <NR1 NUMERIC RESPONSE DATA>
```

- 1: Started
- 0: Stopped
- -1: Auto Search failure (when any slot has failed)

**Function:** Queries the Auto Search function state.

**Example:**

```plaintext
> :SENSe:MEASure:ASEarch:STATe?
< 1
```

**Compatibility:** Compatible with the MP1632C.

### :SENSe:MEASure:ASEarch:SLASet

**Function:** Sets Auto Search execution for all slots to ON.

**Example:**

```plaintext
> :SENSe:MEASure:ASEarch:SLASet
```

**Compatibility:** Incompatible with existing models.

### :SENSe:MEASure:ASEarch:SLAReset

**Function:** Sets Auto Search execution for all slots to OFF.

**Example:**

```plaintext
> :SENSe:MEASure:ASEarch:SLAReset
```

**Compatibility:** Incompatible with existing models.
Chapter 7  SCPI Commands

:SENSe:MEASure:ASEarch:SELSlot <slot>,<boolean>[,<unit>]

Parameter

- `<slot>` = <CHARACTER PROGRAM DATA>
  - SLOT1 to SLOT6: Slot No.1 to 6
  - When using the MP1800A: 1 to 6, when using the MT1810A: 1 to 4
- `<boolean>` = <BOOLEAN PROGRAM DATA>
  - ON, 1: Auto Search ON
  - OFF, 0: Auto Search OFF
- `<unit>` = <DECIMAL NUMERIC PROGRAM DATA>
  - 1 to 4: Mainframe Nos. 1 to 4
  - When using two or more MT1810A units in serial connection, specify the mainframe number.
  - Can be omitted. Mainframe No. 1 is specified when omitted.

Function
Selects the slot for execution of Auto Search.

Example
To set Auto Search for Slot No.1 to ON:
> :SENSe:MEASure:ASEarch:SELSlot SLOT1,ON

Compatibility
Incompatible with existing models.

:SENSe:MEASure:ASEarch:SELSlot? <slot>[,<unit>]

Parameter

- `<slot>` = <CHARACTER PROGRAM DATA>
  - SLOT1 to SLOT6: Slot No.1 to 6
  - When using the MP1800A: 1 to 6, when using the MT1810A: 1 to 4
- `<unit>` = <DECIMAL NUMERIC PROGRAM DATA>
  - 1 to 4: Mainframe Nos. 1 to 4
  - When using two or more MT1810A units in serial connection, specify the mainframe number.
  - Can be omitted. Mainframe No. 1 is specified when omitted.

Response
 `<boolean>` = <NR1 NUMERIC RESPONSE DATA>
  - 1: ON
  - 0: OFF

Function
Queries the ON/OFF state of Auto Search for each slot.

Example
To query the ON/OFF state of Auto Search for Slot No.1:
> :SENSe:MEASure:ASEarch:SELSlot? SLOT1
< 1

Compatibility
Incompatible with existing models.
### SCPI Commands

#### :SENSe:MEASure:ASEarch:SLOT? [<unit>]

**Parameter**

- `<unit>` = `<DECIMAL NUMERIC PROGRAM DATA>`
- 1 to 4 Mainframe Nos. 1 to 4

When using two or more MT1810A units in serial connection, specify the mainframe number.

Can be omitted. Mainframe No. 1 is specified when omitted.

**Response**

- `<slot>` = `<CHARACTER RESPONSE DATA>`
- SLOT1 to SLOT6 Slot No.1 to 6

When using the MP1800A: 1 to 6, when using the MT1810A: 1 to 4

**Note:**

The above response returns all slots for which Auto Search can be executed by delimiting with commas (,).

**Function**

Queries the slot for which Auto Search can be executed.

**Example**

```plaintext
> :SENSe:MEASure:ASEarch:SLOT?
< SLOT1,SLOT2,SLOT6
```

**Compatibility**

Incompatible with existing models.
### \texttt{CALCulate:DATA:ASEarch? <slot>[,<unit>]} 

**Parameter** 
\(<\text{slot}>\) = \text{<CHARACTER PROGRAM DATA>} 
SLOT1 to SLOT6  
Slot No.1 to 6  
When using the MP1800A: 1 to 6, when using the MT1810A: 1 to 4  
\([\text{<unit>}]\) = \text{<DECIMAL NUMERIC PROGRAM DATA>}  
1 to 4  
Mainframe Nos. 1 to 4  
When using two or more MT1810A units in serial connection, specify the mainframe number.  
Can be omitted. Mainframe No. 1 is specified when omitted.  

**Response** 
\(<\text{data}>,<\text{xdata}>,<\text{delay}>,<\text{numeric1}>,<\text{numeric2}>,<\text{numeric3}>,<\text{numeric4}>\) 
\(<\text{data}>,<\text{xdata}>,<\text{delay}>, = \text{<STRING RESPONSE DATA>}\) 
"Done"  
Normal termination  
"Fail"  
Abnormal termination  
"----"  
Not executed  
\(<\text{numeric1}>,<\text{numeric2}> = \text{<NR2 NUMERIC RESPONSE DATA>}\)  
\(-3.500 \text{ to } 3.300\)  
\(-3.500 \text{ to } 3.300\) V/Step 0.001 V  
"----"  
Not executed  
"----"  
Abnormal termination  
\(<\text{numeric3}> = \text{<NR1 NUMERIC RESPONSE DATA}>\)  
\(-1000 \text{ to } 1000\)  
\(-1000 \text{ to } 1000\) mUI/Step 1 mUI  
"----"  
Not executed  
"----"  
Abnormal termination  
\(<\text{numeric4}> = \text{<NR2 NUMERIC RESPONSE DATA}>\)  
\(-\text{xxx.xx to xxx.xx}\)  
ps (converted from mUI)  
"----"  
Not executed  
"----"  
Abnormal termination  

**Function** 
Queries the Auto Search execution result.  

**Example** 
> \text{:CALCulate:DATA:ASEarch? SLOT1}\n< "Done", "Done", "Done",1.000,1.100, -500,50.00  
< "Done", "Done", "----",1.000,1.100,0,0.00 (When not executed in the Phase direction)  

**Compatibility** 
Incompatible with existing models.
7.1 Common Commands

7.1.5 Auto Adjust

This chapter describes the commands that are used after Auto Adjust is set by the `SYSTem:CFUnction` command.

![Auto Adjust setting screen]

**Figure 7.1.5-1  Auto Adjust setting screen**

**Table 7.1.5-1  Auto Adjust setting commands**

<table>
<thead>
<tr>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>Item</td>
<td>':SENSe:MEASure:AADJust:ITEM'</td>
</tr>
<tr>
<td></td>
<td>':SENSe:MEASure:AADJust:ITEM?'</td>
</tr>
<tr>
<td>Start</td>
<td>':SENSe:MEASure:AADJust:STARt'</td>
</tr>
<tr>
<td>Stop</td>
<td>':SENSe:MEASure:AADJust:STOP'</td>
</tr>
<tr>
<td>No label (Query for function state)</td>
<td>':SENSe:MEASure:AADJust:STATe?'</td>
</tr>
<tr>
<td>Slot Select</td>
<td>':SENSe:MEASure:AADJust:SELSlot'</td>
</tr>
<tr>
<td>No label (Query for slot state)</td>
<td>':SENSe:MEASure:AADJust:SELSlot?'</td>
</tr>
<tr>
<td>Set All</td>
<td>':SENSe:MEASure:AADJust:SLASet'</td>
</tr>
<tr>
<td>Reset All</td>
<td>':SENSe:MEASure:AADJust:SLAReset'</td>
</tr>
</tbody>
</table>
Chapter 7  SCPI Commands

:SENSe:MEASure:AADJust:ITEM <item>

Parameter  

<table>
<thead>
<tr>
<th>&lt;item&gt;</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PTHR</td>
<td>Phase &amp; Threshold: Auto tracking in the voltage and phase directions</td>
</tr>
<tr>
<td>THR</td>
<td>Threshold: Auto tracking in the voltage direction</td>
</tr>
<tr>
<td>PHAS</td>
<td>Phase: Auto tracking in the phase direction</td>
</tr>
</tbody>
</table>

Function  
Sets the execution item of Auto Adjust.

Example
To set the execution item of Auto Adjust to Phase & Threshold:
> :SENSe:MEASure:AADJust:ITEM PTHR

Compatibility  
Incompatible with existing models.

:SENSe:MEASure:AADJust:ITEM?

Response  

<table>
<thead>
<tr>
<th>&lt;item&gt;</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PTHR</td>
<td>Phase &amp; Threshold</td>
</tr>
<tr>
<td>THR</td>
<td>Threshold</td>
</tr>
<tr>
<td>PHAS</td>
<td>Phase</td>
</tr>
</tbody>
</table>

Function  
Queries the execution item of Auto Adjust.

Example
> :SENSe:MEASure:AADJust:ITEM?
< PTHR

Compatibility  
Incompatible with existing models.

:SENSe:MEASure:AADJust:STARt

Function  
Starts the Auto Adjust function.

Example
> :SENSe:MEASure:AADJust:STARt

Compatibility  
Incompatible with existing models.

:SENSe:MEASure:AADJust:STOP

Function  
Stops the Auto Adjust function.

Example
> :SENSe:MEASure:AADJust:STOP

Compatibility  
Incompatible with existing models.
7.1 Common Commands

:SENSe:MEASure:AADJust:STATe?

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;boolean&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Start</td>
</tr>
<tr>
<td>0</td>
<td>Stop</td>
</tr>
</tbody>
</table>

Function: Queries the Auto Adjust function state.

Example:
> :SENSe:MEASure:AADJust:STATe?
< 1

Compatibility: Incompatible with existing models.

:SENSe:MEASure:AADJust:SELSlot <slot>,<boolean>[,<unit>]

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;slot&gt; = &lt;CHARACTER PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SLOT1 to SLOT6 Slot No.1 to 6</td>
</tr>
<tr>
<td></td>
<td>When using the MP1800A: 1 to 6, when using the MT1810A: 1 to 4</td>
</tr>
<tr>
<td></td>
<td>&lt;boolean&gt; = &lt;BOOLEAN PROGRAM DATA&gt;</td>
</tr>
<tr>
<td></td>
<td>ON, 1 Auto Adjust ON</td>
</tr>
<tr>
<td></td>
<td>OFF, 0 Auto Adjust OFF</td>
</tr>
<tr>
<td></td>
<td>[&lt;unit&gt;] = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
</tr>
<tr>
<td></td>
<td>1 to 4 Mainframe Nos. 1 to 4</td>
</tr>
<tr>
<td></td>
<td>When using two or more MT1810A units in serial connection, specify the mainframe number.</td>
</tr>
<tr>
<td></td>
<td>Can be omitted. Mainframe No. 1 is specified when omitted.</td>
</tr>
</tbody>
</table>

Function: Selects the slot for which Auto Adjust is executed.

Example:
To set Slot No.1 to Auto Adjust ON:
> :SENSe:MEASure:AADJust:SELSlot SLOT1,ON

Compatibility: Incompatible with existing models.

:SENSe:MEASure:AADJust:SELSlot? <slot>[,<unit>]

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;slot&gt; = &lt;CHARACTER PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SLOT1 to SLOT6 Slot No.1 to 6</td>
</tr>
<tr>
<td></td>
<td>When using the MP1800A: 1 to 6, when using the MT1810A: 1 to 4</td>
</tr>
<tr>
<td></td>
<td>[&lt;unit&gt;] = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
</tr>
<tr>
<td></td>
<td>1 to 4 Mainframe Nos. 1 to 4</td>
</tr>
<tr>
<td></td>
<td>When using two or more MT1810A units in serial connection, specify the mainframe number.</td>
</tr>
<tr>
<td></td>
<td>Can be omitted. Mainframe No. 1 is specified when omitted.</td>
</tr>
</tbody>
</table>

Response: <boolean> = <NR1 NUMERIC RESPONSE DATA>

<table>
<thead>
<tr>
<th>1</th>
<th>ON</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>OFF</td>
</tr>
</tbody>
</table>

Function: Queries the Auto Adjust ON/OFF state for each slot.

Example:
To query the Auto Adjust ON/OFF state for Slot No.1:
> :SENSe:MEASure:AADJust:SELSlot? SLOT1
Chapter 7  SCPI Commands

Incompatible with existing models.

**:SENSe:MEASure:AADJust:SLOT? [<unit>]**

**Parameter**

- `<unit>` = <DECIMAL NUMERIC PROGRAM DATA>
- 1 to 4  Mainframe Nos. 1 to 4
  - When using two or more MT1810A units in serial connection, specify the mainframe number.
  - Can be omitted. Mainframe No. 1 is specified when omitted.

**Response**

- `<slot>` = <CHARACTER RESPONSE DATA>
- SLOT1 to SLOT6  Slot No.1 to 6
  - When using the MP1800A: 1 to 6, when using the MT1810A: 1 to 4

**Note:**

The above response returns all slots for which Auto Adjust can be executed by delimiting with commas (,).

**Function**

Queries the slot for which Mainframe No.3 Auto Adjust can be executed.

**Example**

To query the slot for which Auto Adjust can be executed:

> :SENSe:MEASure:AADJust:SLOT? 3

< SLOT1,SLOT2, SLOT6

**Compatibility**

Incompatible with existing models.

---

**:SENSe:MEASure:AADJust:SLASet**

**Function**

Select all slots for Auto Adjust execution.

**Example**

> :SENSe:MEASure:AADJust:SLASet

**Compatibility**

Incompatible with existing models.

---

**:SENSe:MEASure:AADJust:SLAReset**

**Function**

Deselect all slots for Auto Adjust execution.

**Example**

> :SENSe:MEASure:AADJust:SLAReset

**Compatibility**

Incompatible with existing models.
7.1 Common Commands

7.1.6 ED ALL

Table 7.1.6-1  ED ALL setting commands

<table>
<thead>
<tr>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>Measurement result query in the case of Combination</td>
<td>:CALCulate:ADATa:EALarm?</td>
</tr>
<tr>
<td>Logging ON/OFF</td>
<td>:DISPlay:ARESult:LOGGing:ONOff</td>
</tr>
<tr>
<td>Clear</td>
<td>:DISPlay:ARESult:LOGGing:CLEar</td>
</tr>
<tr>
<td>Condition</td>
<td>:DISPlay:ARESult:LOGGing:CONDition:ITEM</td>
</tr>
<tr>
<td>Set All/Reset All</td>
<td>:DISPlay:ARESult:LOGGing:CONDition:ALLSet</td>
</tr>
<tr>
<td>1 Second data</td>
<td>:DISPlay:ARESult:LOGGing:CONDition:PERiod</td>
</tr>
<tr>
<td>Error Threshold</td>
<td>:DISPlay:ARESult:LOGGing:CONDition:THReshold</td>
</tr>
<tr>
<td>Squelch</td>
<td>:DISPlay:ARESult:LOGGing:CONDition:SQUelch</td>
</tr>
<tr>
<td>Save</td>
<td>:SYSTem:MMEMory:EDAL:STORe</td>
</tr>
<tr>
<td>Print</td>
<td>:SYSTem:PRINt:EDAL</td>
</tr>
</tbody>
</table>

:CALCulate:ADATa:EALarm?<slot>,<string>

Parameter

| <slot> = <STRING PROGRAM DATA> |
| "X-Y" | X: 1 to 4 Mainframe No. |
| Y: Comb1 to 8 Combination No. |
| <string> = <STRING PROGRAM DATA> |
| "CURRent:<result>" | Current data |
| "LAST:<result>" | Measurement end data |
| "INTermediate:<result>" | Measurement intermediate data |

See Table 7.1.6-3 for details on <result>.

Response

| <string> = <STRING RESPONSE DATA> |
### Table 7.1.6-2  Response formats

<table>
<thead>
<tr>
<th>Type</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Form1 Integer type</td>
<td>\texttt{&quot;XXXXXXX&quot;}</td>
<td>For 0 to 9999999</td>
</tr>
<tr>
<td></td>
<td>\texttt{&quot;X.XXXXXXXX&quot;}</td>
<td>For 1.0000E07 to 9.9999E16</td>
</tr>
<tr>
<td></td>
<td>\texttt{&quot;--------&quot;}</td>
<td>When no data corresponds to the query</td>
</tr>
<tr>
<td>Form2 Fraction type</td>
<td>\texttt{&quot;X.XXXE-XX&quot;}</td>
<td>For 0.0000E16 to 1.0000E–00</td>
</tr>
<tr>
<td></td>
<td>\texttt{&quot;--------&quot;}</td>
<td>When no data corresponds to the query</td>
</tr>
<tr>
<td>Form3 % type</td>
<td>\texttt{&quot;XXX.XXXXX&quot;}</td>
<td>For 0.0000 to 100.0000</td>
</tr>
<tr>
<td></td>
<td>\texttt{&quot;--------&quot;}</td>
<td>When no data corresponds to the query</td>
</tr>
<tr>
<td>Form4 Frequency type</td>
<td>\texttt{&quot;XXXXXXXXX&quot;}</td>
<td>For 0 to MAX (Hz)</td>
</tr>
<tr>
<td></td>
<td>\texttt{&quot;---------&quot;}</td>
<td>When no data corresponds to the query</td>
</tr>
<tr>
<td>Form5 Voltage type</td>
<td>\texttt{&quot;X.XXX&quot;}</td>
<td>For −4.000 to +4.000</td>
</tr>
<tr>
<td></td>
<td>\texttt{&quot;----&quot;}</td>
<td>When no data corresponds to the query</td>
</tr>
</tbody>
</table>

### Function
Queries the measurement data at Combination that corresponds to the parameter.

### Table 7.1.6-3  Parameters

<table>
<thead>
<tr>
<th>Item</th>
<th>&lt;result&gt;</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>Error Rate Total</td>
<td>ER:TOTal</td>
<td>Form2</td>
</tr>
<tr>
<td>INS</td>
<td>ER:INSertion</td>
<td></td>
</tr>
<tr>
<td>OMI</td>
<td>ER:OMISSion</td>
<td></td>
</tr>
<tr>
<td>Error Count Total</td>
<td>EC:TOTal</td>
<td>Form1</td>
</tr>
<tr>
<td>INS</td>
<td>EC:INSertion</td>
<td></td>
</tr>
<tr>
<td>OMI</td>
<td>EC:OMISSion</td>
<td></td>
</tr>
<tr>
<td>Error Rate Transition</td>
<td>ER:TRANSition</td>
<td>Form2</td>
</tr>
<tr>
<td>Non Transition</td>
<td>ER:NONTransition</td>
<td></td>
</tr>
<tr>
<td>Error Count Transition</td>
<td>EC:TRANSition</td>
<td>Form1</td>
</tr>
<tr>
<td>Non Transition</td>
<td>EC:NONTransition</td>
<td></td>
</tr>
<tr>
<td>%EFI Total</td>
<td>EFI:TOTal</td>
<td>Form3</td>
</tr>
<tr>
<td>EI Total</td>
<td>EI:TOTal</td>
<td>Form1</td>
</tr>
<tr>
<td>Frequency (kHz)</td>
<td>FREQuency</td>
<td>Form4</td>
</tr>
<tr>
<td>Clock Count Total</td>
<td>CC:TOTal</td>
<td>Form1</td>
</tr>
<tr>
<td>Clock Loss</td>
<td>AIMterval:CLOSs</td>
<td>Form1</td>
</tr>
<tr>
<td>Sync Loss</td>
<td>AIMterval:PSLoss</td>
<td>Form1</td>
</tr>
<tr>
<td>Data Threshold</td>
<td>THReshold</td>
<td>Form5</td>
</tr>
<tr>
<td>Data Phase</td>
<td>PHASE</td>
<td>Form1</td>
</tr>
<tr>
<td>Threshold EI &gt;1.0E–3</td>
<td>THReshold:EI:TOTal:E_3</td>
<td>Form1</td>
</tr>
<tr>
<td>&gt;1.0E–4</td>
<td>THReshold:EI:TOTal:E_4</td>
<td>Form1</td>
</tr>
<tr>
<td>&gt;1.0E–5</td>
<td>THReshold:EI:TOTal:E_5</td>
<td>Form1</td>
</tr>
<tr>
<td>&gt;1.0E–6</td>
<td>THReshold:EI:TOTal:E_6</td>
<td>Form1</td>
</tr>
<tr>
<td>&gt;1.0E–7</td>
<td>THReshold:EI:TOTal:E_7</td>
<td>Form1</td>
</tr>
<tr>
<td>&gt;1.0E–8</td>
<td>THReshold:EI:TOTal:E_8</td>
<td>Form1</td>
</tr>
<tr>
<td>≤1.0E–8</td>
<td>THReshold:EI:TOTal:UE_8</td>
<td>Form1</td>
</tr>
</tbody>
</table>
### 7.1 Common Commands

#### Table 7.1.6-3 Parameters (Cont’d)

<table>
<thead>
<tr>
<th>Item</th>
<th>&lt;result&gt;</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>Threshold %EFI</td>
<td>&gt;1.0E–3 THReshold:EFI:TOTal:E_3</td>
<td>Form3</td>
</tr>
<tr>
<td></td>
<td>&gt;1.0E–4 THReshold:EFI:TOTal:E_4</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&gt;1.0E–5 THReshold:EFI:TOTal:E_5</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&gt;1.0E–6 THReshold:EFI:TOTal:E_6</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&gt;1.0E–7 THReshold:EFI:TOTal:E_7</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&gt;1.0E–8 THReshold:EFI:TOTal:E_8</td>
<td></td>
</tr>
<tr>
<td></td>
<td>≤1.0E–8 THReshold:EFI:TOTal:UE_8</td>
<td></td>
</tr>
<tr>
<td>G.821</td>
<td>ES G821:ES2:TOTal</td>
<td>Form1</td>
</tr>
<tr>
<td></td>
<td>EFS G821:EFS2:TOTal</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SES G821:SES2:TOTal</td>
<td></td>
</tr>
<tr>
<td></td>
<td>DM G821:DM2:TOTal</td>
<td></td>
</tr>
<tr>
<td></td>
<td>US G821:US2:TOTal</td>
<td></td>
</tr>
<tr>
<td></td>
<td>%ES G821:ES:TOTal</td>
<td>Form3</td>
</tr>
<tr>
<td></td>
<td>%EFS G821:EFS:TOTal</td>
<td></td>
</tr>
<tr>
<td></td>
<td>%SES G821:SES:TOTal</td>
<td></td>
</tr>
<tr>
<td></td>
<td>%DM G821:DM:TOTal</td>
<td></td>
</tr>
<tr>
<td></td>
<td>%US G821:US:TOTal</td>
<td></td>
</tr>
</tbody>
</table>

#### Example

To query the total error rate measurement data for the current data:

```plaintext
> :CALCulate:ADATa:EALarm? "1-1","CURRent:ER:TOTal"
< "0.0000E–16"
```

#### Compatibility

Incompatible with existing models.

---

### :DISPlay:ARESult:LOGGing:ONOFF <boolean>

- **Parameter**
  - `<boolean>` = `<BOOLEAN PROGRAM DATA>`
  - 0 or OFF: Logging OFF
  - 1 or ON: Logging ON

- **Function**
  - Sets logging execution for all ED modules ON/OFF.

- **Example**
  - To set logging execution for all ED modules ON:
    ```plaintext
    > :DISPlay:ARESult:LOGGing:ONOFF ON
    ```

- **Compatibility**
  - Incompatible with existing models.
Chapter 7  SCPI Commands

:DISPLAY:ARESult:LOGGing:ONOFf?

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;numeric&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>OFF</td>
</tr>
<tr>
<td>1</td>
<td>ON</td>
</tr>
</tbody>
</table>

Function Queries the logging execution ON/OFF state for all ED modules.

Example To query the logging execution ON/OFF state for all ED modules:
> :DISPLAY:ARESult:LOGGing:ONOFf?
< 1

Compatibility Incompatible with existing models.

:DISPLAY:ARESult:LOGGing:CLEar

Function Clears the logging results for all ED modules.

Example > :DISPLAY:ARESult:LOGGing:CLEar

Compatibility Incompatible with existing models.

:DISPLAY:ARESult:LOGGing:CONDition:ITEM <item>,<boolean>

Parameter <item> = <CHARACTER PROGRAM DATA>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;item&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>SLOTinfo</td>
<td>Slot information</td>
</tr>
<tr>
<td>TPATtern</td>
<td>Test pattern type</td>
</tr>
<tr>
<td>START</td>
<td>Measurement start time</td>
</tr>
<tr>
<td>END</td>
<td>Measurement end time</td>
</tr>
<tr>
<td>ERATe</td>
<td>Error rate and error count</td>
</tr>
<tr>
<td>EI</td>
<td>Error interval (EI) and error free interval rate (%EFI)</td>
</tr>
<tr>
<td>FREQuency</td>
<td>Frequency</td>
</tr>
<tr>
<td>CCOunt</td>
<td>Clock count</td>
</tr>
<tr>
<td>AOCCur</td>
<td>Alarm occurrence (Alarm Occur) and recovery time (Recover)</td>
</tr>
<tr>
<td>AINTerval</td>
<td>Alarm interval</td>
</tr>
<tr>
<td>DTHReshold</td>
<td>Adds the data input threshold setting value to the logging item.</td>
</tr>
<tr>
<td>CDELay</td>
<td>Adds the clock phase setting value to the logging item.</td>
</tr>
</tbody>
</table>

Disabled when Variable Clock Delay (MU181040A-x30) is not installed.

SDATa Average data for one second

<boolean> = <BOOLEAN PROGRAM DATA>

<table>
<thead>
<tr>
<th>&lt;boolean&gt;</th>
<th>Measurement item</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 or OFF</td>
<td>Measurement item OFF</td>
</tr>
<tr>
<td>1 or ON</td>
<td>Measurement item ON</td>
</tr>
</tbody>
</table>

Function Sets the logging measurement item for all ED modules ON/OFF.

Example To set the logging slot information for all ED modules ON:
> :DISPLAY:ARESult:LOGGing:CONDition:ITEM SLOTinfo,ON

Compatibility Incompatible with existing models.
7.1 Common Commands

**:DISPlay:ARESult:LOGGing:CONDition:ITEM? <item>**

**Parameter**

\[
\text{<item>} = \langle \text{CHARACTER PROGRAM DATA} \rangle
\]

SLOTinfo, TPATtern, STARt, END, ERATe, EI, FREQuency, CCOunt,
AOCCur, AINTerval, DTHReshold, CDELey, SDATa

**Response**

\[
\text{<numeric>} = \langle \text{NR1 NUMERIC RESPONSE DATA} \rangle
\]

0 OFF
1 ON

**Function**
Queries the logging measurement item ON/OFF state for all ED modules.

**Example**

> :DISPlay:ARESult:LOGGing:CONDition:ITEM? SLOTinfo
< 1

**Compatibility**
Incompatible with existing models.

**:DISPlay:ARESult:LOGGing:CONDition:ALLSet <boolean>**

**Parameter**

\[
\text{<boolean>} = \langle \text{BOOLEAN PROGRAM DATA} \rangle
\]

0 or OFF Reset
1 or ON All Set

**Function**
Selects or resets all logging measurement items for all ED modules.

**Example**

To select all logging measurement items for all ED modules:

> :DISPlay:ARESult:LOGGing:CONDition:ALLSet ON

**Compatibility**
Incompatible with existing models.

**:DISPlay:ARESult:LOGGing:CONDition:PERiod <period>**

**Parameter**

\[
\text{<period>} = \langle \text{CHARACTER PROGRAM DATA} \rangle
\]

S10 10 seconds
S30 30 seconds
M1 1 minute
M10 10 minutes
M30 30 minutes
H1 1 hour

**Function**
Sets the logging result capturing period for all ED modules.

**Example**

To set the logging result capturing period for all ED modules to 10 seconds:

> :DISPlay:ARESult:LOGGing:CONDition:PERiod S10

**Compatibility**
Incompatible with existing models.
Chapter 7  SCPI Commands

:DISPlay:ARESult:LOGGing:CONDition:PERiod?

Response  

Response  

:DISPlay:ARESult:LOGGing:CONDition:PERiod?

Response  

Function  
Queries the logging result capturing period for all ED modules.

Example  

Example  

Compatibility  
Incompatible with existing models.

:DISPlay:ARESult:LOGGing:CONDition:THReshold <thre>

Parameter  

Parameter  

Function  
Sets the logging error rate threshold value for all ED modules.

Example  
To set the logging error rate threshold value for all ED modules to 0:

Example  

Compatibility  
Incompatible with existing models.

:DISPlay:ARESult:LOGGing:CONDition:THReshold?

Response  

Response  

Function  
Queries the logging error rate threshold value for all ED modules.

Example  

Example  

Compatibility  
Incompatible with existing models.
### 7.1 Common Commands

#### :DISPlay:ARESult:LOGGing:CONDition:SQUelch <boolean>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;boolean&gt;</code> = <code>&lt;BOOLEAN PROGRAM DATA&gt;</code></td>
<td>Sets the logging memory squelch function for all ED modules ON/OFF.</td>
<td>&gt; :DISPlay:ARESult:LOGGing:CONDition:SQUelch ON</td>
</tr>
<tr>
<td>0 or OFF</td>
<td>Memory squelch function ON</td>
<td>Incompatible with existing models.</td>
</tr>
<tr>
<td>1 or ON</td>
<td>Memory squelch function OFF</td>
<td></td>
</tr>
</tbody>
</table>

#### :DISPlay:ARESult:LOGGing:CONDition:SQUelch?

<table>
<thead>
<tr>
<th>Response</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;numeric&gt;</code> = <code>&lt;NR1 NUMERIC RESPONSE DATA&gt;</code></td>
<td>Queries the logging memory squelch function ON/OFF state for all ED modules.</td>
<td>&gt; :DISPlay:ARESult:LOGGing:CONDition:SQUelch? &lt; 1</td>
</tr>
<tr>
<td>0</td>
<td>Memory squelch function OFF</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Memory squelch function ON</td>
<td></td>
</tr>
</tbody>
</table>

#### :SYSTem:MMEMory:EDAL:STORe <file_name>,<data_type>,<file_type>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;file_name&gt;</code> = <code>&lt;STRING PROGRAM DATA&gt;</code></td>
<td>Saves the ED Result All measurement result data.</td>
<td>&gt; :SYSTem:MMEMory:EDAL:STORe &quot;C:\Test\example&quot;,AERA,CSV</td>
</tr>
<tr>
<td>&quot;&lt;drv&gt;:[&lt;dir&gt;]&lt;file&gt;&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>&lt;drv&gt;</code> = C, D, E, F</td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>&lt;dir&gt;</code> = ...</td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>&lt;file&gt;</code> = File name</td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>&lt;data_type&gt;</code> = <code>&lt;CHARACTER PROGRAM DATA&gt;</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AERA</td>
<td>Error/Alarm Result ALL</td>
<td></td>
</tr>
<tr>
<td>ALOG</td>
<td>Logging Result ALL</td>
<td></td>
</tr>
<tr>
<td><code>&lt;file_type&gt;</code> = <code>&lt;CHARACTER PROGRAM DATA&gt;</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CSV</td>
<td>CSV File</td>
<td></td>
</tr>
<tr>
<td>TXT</td>
<td>Text File</td>
<td></td>
</tr>
</tbody>
</table>

Only TXT can be specified for `<file_type>` when ALOG is set for `<data_type>`.

#### Compatibility

Incompatible with existing models.
### :SYSTem:PRINt:EDAL <list>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;list&gt;</td>
<td>&lt;CHARACTER PROGRAM DATA&gt;</td>
</tr>
<tr>
<td>AERA</td>
<td>Error/Alarm Result All</td>
</tr>
<tr>
<td>ALOG</td>
<td>Logging Result All</td>
</tr>
</tbody>
</table>

**Function**

Prints the ED Result All measurement result data.

**Example**

To print the Error/Alarm ED Result All measurement result:

```
> :SYSTem:PRINt:EDAL AERA
```

**Compatibility**

Incompatible with existing models.
7.1 Common Commands

7.1.7 Pattern Editor

This chapter describes the pattern file save/read commands for the PPG and the ED. Before executing a setting/query command for the MU181020A/MU181040A, specify the slot number of the module to be remotely controlled, by using the :MODule:ID command. Refer to Chapter 7.1 "Common Commands" and 9.1 "Basic Operation of Device Messages" for how to specify a slot number with the :MODule:ID command.

Table 7.1.7-1 Pattern Editor setting commands

<table>
<thead>
<tr>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>Open</td>
<td>:SYSTem:MMEMory:PATTer:n:RECall</td>
</tr>
<tr>
<td>Save</td>
<td>:SYSTem:MMEMory:PATTer:n:STORe</td>
</tr>
</tbody>
</table>

:SYSTem:MMEMory:PATTer:n:RECall <file_name>,<file_type>

Parameter

- `<file_name>` = <STRING PROGRAM DATA>
- "<drv>:\[<dir>]\<file>"
- `<drv>` = C, D, E, F
- `<dir>` = <dir1>\<dir2>\... (Omitted for the root directory)
- `<file>` = File name
- `<file_type>` = <CHARACTER PROGRAM DATA>
  - BIN Binary file
  - TXT Text file

Function

Opens a pattern file.

Example

To open a pattern file in the specified file format from the specified destination:

```
> :SYSTem:MMEMory:PATTer:n:RECall "C:\Test\example",BIN
```

Compatibility

Incompatible with existing models.
Chapter 7  SCPI Commands

:SYSTem:MEMory:PATh:STORe <file_name>,<file_type>,

Parameter

Parameter

Parameter

Parameter

Parameter

Parameter

Parameter

Parameter

Parameter

Parameter

Parameter

Parameter

Parameter

Parameter

Parameter

Parameter

Parameter

Parameter

Parameter

Parameter

Parameter

Parameter

Parameter

Parameter

Parameter

Parameter

Parameter

Parameter

Parameter

Parameter

Function

Saves a pattern file.

Note:

The settings will not be read from the saved file if the file name is changed.

Example

To save a pattern file to the specified destination in the specified file format:

> :SYSTem:MEMory:PATh:STORe "C:\Test\example",TXT

Compatibility

Incompatible with existing models.
7.2 Synthesizer Commands

This section describes MU181000A 12.5 GHz Synthesizer and MU181000B 12.5 GHz 4 port Synthesizer (hereinafter referred to as Synthesizer) commands.

Before executing a setting/query command for the Synthesizer, specify the slot number of the module to be remotely controlled, by using the :MODule:ID command. Refer to Section 7.1 “Common Commands” and 9.1 "Basic Operation of Device Messages" for how to specify a slot number with the :MODule:ID command.

![12.5GHz Synthesizer Window](image)

**Figure 7.2-1  12.5GHz Synthesizer Window**
Figure 7.2-2  12.5 GHz Synthesizer Window (Option x01)
7.2 Synthesizer Commands

7.2.1 Operation frequency Setting field

![Figure 7.2.1-1  Operation Frequency Setting](image)

Table 7.2.1-1  Operation Frequency setting Commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>:OUTPut:CLOCk:OPERation?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:OUTPut:CLOCk:FREQuency?</td>
</tr>
<tr>
<td>[3]</td>
<td>Center Frequency Unit</td>
<td>:OUTPut:CLOCk:FUNit</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:OUTPut:CLOCk:FUNit?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:OUTPut:CLOCk:OFFSet:PPM?</td>
</tr>
</tbody>
</table>

:OUTPut:CLOCk:OPERation <operation>

Parameter

<table>
<thead>
<tr>
<th>&lt;operation&gt; = &lt;CHARACTER PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>VARiable</td>
</tr>
<tr>
<td>PCIexpress1</td>
</tr>
<tr>
<td>PCIexpress2</td>
</tr>
<tr>
<td>PCIexpress345</td>
</tr>
<tr>
<td>SERialata</td>
</tr>
</tbody>
</table>

Function
Sets Operating of Synthesizer.

Example
Sets Operating of Synthesizer to Variable.

> :OUTPut:CLOCk:OPERation VARiable

Compatibility
Incompatible with existing models.
### :OUTPut:CLOCk:OPERation?

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;operation&gt; = &lt;CHARACTER RESPONSE DATA&gt; VAR, PCI1, PCI2, PCI345, SER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function</td>
<td>Queries Operating of Synthesizer.</td>
</tr>
<tr>
<td>Example</td>
<td>&gt; :OUTPut:CLOCk:OPERation?</td>
</tr>
<tr>
<td></td>
<td>&lt;PCI2</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Incompatible with existing models.</td>
</tr>
</tbody>
</table>

**Note:**

The :OUTPut:CLOCk:OPERation and :OUTPut:CLOCk:OPERation? commands can be executed only when MU181000A-x01, MU181000B-x01, or MU181000B-x02 is installed.

### :OUTPut:CLOCk:FREQuency <numeric>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;numeric&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>In case of Operating frequency resolution unit is kHz 100000 to 12500000 in 1 kHz step</td>
</tr>
<tr>
<td></td>
<td>In case of Operating frequency resolution unit is MHz 100 to 12500/Step 1 MHz</td>
</tr>
<tr>
<td>Function</td>
<td>Sets operating frequency of Synthesizer.</td>
</tr>
<tr>
<td>Example</td>
<td>Sets operating frequency of Synthesizer to 12500 MHz.</td>
</tr>
<tr>
<td></td>
<td>&gt; :OUTPut:CLOCk:FREQuency 12500</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Compatible with MP1632C.</td>
</tr>
</tbody>
</table>

### :OUTPut:CLOCk:FREQuency?

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;numeric&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>In case of Operating frequency resolution unit is kHz 100000 to 12500000 in 1 kHz step</td>
</tr>
<tr>
<td></td>
<td>In case of Operating frequency resolution unit is MHz 100 to 12500/Step 1 MHz</td>
</tr>
<tr>
<td>Function</td>
<td>Queries operating frequency of Synthesizer.</td>
</tr>
<tr>
<td>Example</td>
<td>&gt; :OUTPut:CLOCk:FREQuency?</td>
</tr>
<tr>
<td></td>
<td>&lt; 12500</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Compatible with MP1632C.</td>
</tr>
</tbody>
</table>
7.2 Synthesizer Commands

**:OUTPut:CLOCK:FUNit <resolution>**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;resolution&gt; = &lt;CHARACTER PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>KHZ kHz</td>
</tr>
<tr>
<td></td>
<td>MHZ MHz</td>
</tr>
</tbody>
</table>

| Function | Sets operating frequency resolution of Synthesizer. |
| Example  | Sets operating frequency resolution of Synthesizer to MHz. |
|          | > :OUTPut:CLOCK:FUNit MHZ               |

| Compatibility | Incompatible with existing models. |

**:OUTPut:CLOCK:FUNit?**

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;resolution&gt; = &lt;CHARACTER RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>KHZ, MHZ</td>
</tr>
</tbody>
</table>

| Function | Queries operating frequency resolution of Synthesizer. |
| Example  | > :OUTPut:CLOCK:FUNit?                 |
|          | < MHZ                                  |

| Compatibility | Incompatible with existing models. |

**:OUTPut:CLOCK:OFFSet:PPM <numeric>**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;numeric&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>−1000 to 1000 −1,000 to 1,000 in 1 ppm step</td>
</tr>
</tbody>
</table>

| Function | Sets offset value (ppm) from current operating frequency. |
| Example  | Sets offset value (ppm) to 0 ppm. |
|          | > :OUTPut:CLOCK:OFFSet:PPM 0              |

| Compatibility | Incompatible with existing models. |

**:OUTPut:CLOCK:OFFSet:PPM?**

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;numeric&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>−1000 to 1000 −1,000 to 1,000 in 1 ppm step</td>
</tr>
</tbody>
</table>

| Function | Queries offset value (ppm) from current operating frequency. |
| Example  | > :OUTPut:CLOCK:OFFSet:PPM? |
|          | < 0                      |

| Compatibility | Incompatible with existing models. |
7.2.2 Reference Clock Setting field

Table 7.2.2-1 Reference Clock Setting Commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1]</td>
<td>Source</td>
<td>:OUTPut:RCLock:SELect</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:OUTPut:RCLock:SELect?</td>
</tr>
</tbody>
</table>

:OUTPut:RCLock:SELect <clock>

Parameter  

\[<\text{clock}> = \text{<CHARACTER PROGRAM DATA>}\]

\[\text{INT}ernal\quad \text{INTernal 10 MHz reference signal}\]

\[\text{EXT}ernal10\quad \text{EXTernal 10 MHz reference signal}\]

Function  
 Selects Internal of External 10 MHz reference signal.

Example  
 Sets External 10 MHz reference signal.

> :OUTPut:RCLock:SELect EXT10

Compatibility  
Partially compatible with MP1632C.

:OUTPut:RCLock:SELect?

Response  
\[<\text{clock}> = \text{<CHARACTER RESPONSE DATA>}\]

\[\text{INT, EXT}10\]

Function  
Queries the 10 MHz reference signal setting (Internal or External).

Example  
> :OUTPut:RCLock:SELect?

< EXT10

Compatibility  
Partially compatible with MP1632C.
### 7.2.3 Spectrum Spread setting commands

The following commands are enabled only when the Option MU181000A-x01 or MU181000B-x01 is available.

#### Table 7.2.3-1  Spectrum Spread setting commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>:OUTPut:RCLock:SSC:SET?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:OUTPut:RCLock:SSC:METHod?</td>
</tr>
</tbody>
</table>

#### :OUTPut:RCLock:SSC:SET <SSC>

- **Parameter**: 
  - `<SSC>` = `<CHARACTER PROGRAM DATA>`
  - `OFF`  
  - `INTernal`  
  - `EXTernal10`  
  - `EXTernal100`

- **Function**: Sets whether to use the SSC (Spread Spectrum Clocking) function of the synthesizer and the reference signal to be used.

- **Example**: Sets the SSC function to on using external 100 MHz reference signal of the synthesizer.
  
  ```
  > :OUTPut:RCLock:SSC:SET EXTernal 100
  ```

- **Compatibility**: Incompatible with existing models.
Chapter 7  SCPI Commands

**:OUTPut:RCLock:SSC:SET?**

**Response**

< SSC > = < CHARACTER RESPONSE DATA >

*OFF*  
SSC function off

*INT*  
SSC function on (using internal reference signal)

*EXT10*  
SSC function on  
(using external 10 MHz reference signal)

*EXT100*  
SSC function on  
(using external 100 MHz reference signal)

**Function**

Queries setting of SSC (Spread Spectrum Clocking) function of the synthesizer.

**Example**

>` :OUTPut:RCLock:SSC:SET?`

< EXT100

**Compatibility**

Incompatible with existing models.

**:OUTPut:RCLock:SSC:METHOD <spread>**

**Parameter**

<spread> = < CHARACTER PROGRAM DATA >

*DOWN*  
Changes frequency within the range from 0 to −0.5%

*CENTer*  
Changes frequency within the range of ±0.25%

**Function**

Sets spectrum spreading method for SSC (Spread Spectrum Clocking) using 100 MHz reference signal.

**Example**

Sets spectrum spreading method for SSC using 100 MHz reference signal to DOWN.

>` :OUTPut:RCLock:SSC:METHOD DOWN`

**Compatibility**

Incompatible with existing models.

**:OUTPut:RCLock:SSC:METHOD?**

**Response**

<spread> = < CHARACTER RESPONSE DATA >

DOWN, CENT

**Function**

Queries spectrum spreading method for SSC (Spread Spectrum Clocking) using 100 MHz reference signal.

**Example**

>` :OUTPut:RCLock:SSC:METHOD?`

< DOWN

**Compatibility**

Incompatible with existing models.
### 7.2.4 Jitter setting commands

The following commands are enabled only when the Option MU181000A-x01 or MU181000B-x01 is available.

![Jitter setting (Modulation Source: Internal)](image1)

![Jitter setting (Modulation Source: External)](image2)
### Table 7.2.4-1  Jitter setting commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>:SOURce:SYNThesizer:JITTer:OUTPut:SET?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SOURce:SYNThesizer:JITTer:MODulation:SELect?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SOURce:SYNThesizer:JITTer:MODulation:FREQuency?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SOURce:SYNThesizer:JITTer:MODulation:AMPLitude?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SOURce:SYNThesizer:JITTer:MODulation:FREQuency:RANGE?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SOURce:SYNThesizer:JITTer:MODulation:AMPLitude:RANGE?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SOURce:SYNThesizer:JITTer:TRGSource?</td>
</tr>
</tbody>
</table>

**:SOURce:SYNThesizer:JITTer:OUTPut:SET <boolean>**

Parameter  
<boolean> = <BOOLEAN PROGRAM DATA>

- OFF or 0  Jitter modulation off
- ON or 1   Jitter modulation on

Function
Sets on/off of Jitter output.

Example
Sets Jitter output to off.

> :SOURce:SYNThesizer:JITTer:OUTPut:SET OFF

Compatibility
Incompatible with existing models.

**:SOURce:SYNThesizer:JITTer:OUTPut:SET?**

Response  
<numeric> = <NR1 NUMERIC RESPONSE DATA>

- 0  Jitter modulation off
- 1  Jitter modulation on

Function
Queries on/off of Jitter output.

Example

> :SOURce:SYNThesizer:JITTer:OUTPut:SET?
<< 0

Compatibility
Incompatible with existing models.
### :SOURce:SYNThesizer:JITTer:MODulation:SELe<clock>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;clock&gt; = &lt;CHARACTER PROGRAM DATA&gt;</td>
<td>Internal modulation source</td>
</tr>
<tr>
<td>INternal</td>
<td>Internal modulation source</td>
</tr>
<tr>
<td>EXTernal</td>
<td>External modulation source</td>
</tr>
<tr>
<td>IQEXternal</td>
<td>External modulation source (IQ)</td>
</tr>
</tbody>
</table>

**Function**: Sets modulation signal source.

**Example**: Sets modulation signal source to internal modulation source.

```
> :SOURce:SYNThesizer:JITTer:MODulation:SELe INternal
```

**Compatibility**: Incompatible with existing models.

### :SOURce:SYNThesizer:JITTer:MODulation:SELe?

<table>
<thead>
<tr>
<th>Response</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;clock&gt; = &lt;CHARACTER RESPONSE DATA&gt;</td>
<td>INT, EXT, IQEX</td>
</tr>
</tbody>
</table>

**Function**: Queries modulation signal source.

**Example**: Queries modulation signal source.

```
> :SOURce:SYNThesizer:JITTer:MODulation:SELe?
< INT
```

**Compatibility**: Incompatible with existing models.
### :SOURce:SYNThesizer:JITTer:MODulation:FREQuency

<table>
<thead>
<tr>
<th>Parameter</th>
<th>(&lt;\text{numeric}&gt;,&lt;\text{resolution}&gt;)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(&lt;\text{numeric}&gt;) = (&lt;\text{DECIMAL NUMERIC PROGRAM DATA}&gt;)</td>
<td></td>
</tr>
<tr>
<td>When Hz is selected for (&lt;\text{resolution}&gt;)</td>
<td></td>
</tr>
<tr>
<td>9.000 to 1000.0</td>
<td>9.000 to 1000.0 Hz</td>
</tr>
<tr>
<td>Setting steps</td>
<td></td>
</tr>
<tr>
<td>(9.000[\text{Hz}] \leq \text{fmod} \leq 10.000\ [\text{Hz}])</td>
<td>In 0.001-Hz steps</td>
</tr>
<tr>
<td>(10.01[\text{Hz}] \leq \text{fmod} \leq 100.00\ [\text{Hz}])</td>
<td>In 0.01-Hz steps</td>
</tr>
<tr>
<td>(100.1[\text{Hz}] \leq \text{fmod} \leq 1000.0\ [\text{Hz}])</td>
<td>In 0.1-Hz steps</td>
</tr>
<tr>
<td>When kHz is selected for (&lt;\text{resolution}&gt;)</td>
<td></td>
</tr>
<tr>
<td>1.001 to 1000.0</td>
<td>1.00 to 1000.0 kHz</td>
</tr>
<tr>
<td>Setting steps</td>
<td></td>
</tr>
<tr>
<td>(1.001[\text{kHz}] \leq \text{fmod} \leq 10.000\ [\text{kHz}])</td>
<td>In 0.001-kHz steps</td>
</tr>
<tr>
<td>(10.01[\text{kHz}] \leq \text{fmod} \leq 100.00\ [\text{kHz}])</td>
<td>In 0.01-kHz steps</td>
</tr>
<tr>
<td>(100.1[\text{kHz}] \leq \text{fmod} \leq 1000.0\ [\text{kHz}])</td>
<td>In 0.1-kHz steps</td>
</tr>
<tr>
<td>When MHz is selected for (&lt;\text{resolution}&gt;)</td>
<td></td>
</tr>
<tr>
<td>1.001 to 80.00</td>
<td>1.00 to 80 MHz</td>
</tr>
<tr>
<td>Setting steps</td>
<td></td>
</tr>
<tr>
<td>(1.001[\text{MHz}] \leq \text{fmod} \leq 10.000\ [\text{MHz}])</td>
<td>In 0.001-MHz step</td>
</tr>
<tr>
<td>(10.01[\text{MHz}] \leq \text{fmod} \leq 80.00\ [\text{MHz}])</td>
<td>In 0.01-MHz steps</td>
</tr>
<tr>
<td>&lt;\text{resolution}&gt; = &lt;\text{CHARACTER PROGRAM DATA}&gt;</td>
<td></td>
</tr>
<tr>
<td>HZ</td>
<td>Hz units</td>
</tr>
<tr>
<td>KHZ</td>
<td>kHz units</td>
</tr>
<tr>
<td>MHZ</td>
<td>MHz units</td>
</tr>
</tbody>
</table>

**Function**
Sets internal modulation frequency and units.

**Example**
Sets internal modulation frequency to 10 kHz.
```
> :SOURce:SYNThesizer:JITTer:MODulation:FREQuency
10.00,KHZ
```

**Compatibility**
Incompatible with existing models.
### :SOUR:SYNThesizer:JITTer:MODulation:FREQuency?

| Response | <numeric> = <NR2 NUMERIC RESPONSE DATA>
| --- | --- |
| 9.000 to 1000.0 | 9.000 to 1000.0 Hz  
(When Hz is selected for resolution) |
| 1.001 to 1000.0 | 1.00 to 1000.0 kHz  
(When kHz is selected for resolution) |
| 1.001 to 80.00 | 1.00 to 80 MHz  
(When MHz is selected for resolution) |
| <resolution> = <CHARACTER RESPONSE DATA> | HZ, KHZ, MHZ |

**Function**
Queries internal modulation frequency and units.

**Example**

```
> :SOURce:SYNThesizer:JITTer:MODulation:FREQuency?
< 10.000,KHZ
```

**Compatibility**
Incompatible with existing models.

### :SOUR:SYNThesizer:JITTer:MODulation:AMPLitude

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;numeric&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 to 999</td>
<td>0 to 999 mUIp-p, in 1-mUIp-p steps</td>
</tr>
<tr>
<td>1.00 to 32.00</td>
<td>1.00 to 32.00 UIp-p, in 0.01-UIp-p steps</td>
</tr>
<tr>
<td>32.1 to 256.0</td>
<td>32.1 to 256.0 UIp-p, in 0.1-UIp-p steps</td>
</tr>
<tr>
<td>257 to 2049</td>
<td>257 to 2049 UIp-p, in 1-UIp-p steps</td>
</tr>
<tr>
<td>2050 to 4000</td>
<td>2050 to 4000 UIp-p, in 10-UIp-p steps</td>
</tr>
<tr>
<td>&lt;resolution&gt; = &lt;CHARACTER PROGRAM DATA&gt;</td>
<td>MUI mUIp-p units</td>
</tr>
<tr>
<td>UI</td>
<td>UIp-p units</td>
</tr>
</tbody>
</table>

**Function**
Sets internal modulation amplitude and units.

**Example**

```
> :SOURce:SYNThesizer:JITTer:MODulation:AMPLitude 1.10,UI
```

**Compatibility**
Incompatible with existing models.
Chapter 7  SCPI Commands

:SOURce:SYNThesizer:JITTer:MODulation:AMPLitude?

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;numeric&gt; = &lt;NR2 NUMERIC RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 to 999</td>
<td>0 to 999 mUIp-p</td>
</tr>
<tr>
<td>1.00 to 32.00</td>
<td>1.00 to 32.00 UIp-p</td>
</tr>
<tr>
<td>32.1 to 256.0</td>
<td>32.1 to 256.0 UIp-p</td>
</tr>
<tr>
<td>257 to 2049</td>
<td>257 to 2049 UIp-p</td>
</tr>
<tr>
<td>2050 to 4000</td>
<td>2050 to 4000 UIp-p</td>
</tr>
</tbody>
</table>

<resolution> = <CHARACTER RESPONSE DATA>

<table>
<thead>
<tr>
<th>MUI</th>
<th>mUIp-p units</th>
</tr>
</thead>
<tbody>
<tr>
<td>UI</td>
<td>UIp-p units</td>
</tr>
</tbody>
</table>

Function
Queries internal modulation amplitude and units.

Example

> :SOURce:SYNThesizer:JITTer:MODulation:AMPLitude?
< 1.10, UI

Compatibility
Incompatible with existing models.
### :SOURce:SYNThesizer:JITTer:MODulation:FREQuency:RANGe <range>

| Parameter |  
|--------------------|-----------------------------|
| H_275  | 275 Hz  |
| H_550  | 550 Hz  |
| K_1    | 1 kHz   |
| K_1_1  | 1.1 kHz |
| K_2_2  | 2.2 kHz |
| K_2_75 | 2.75 kHz|
| K_4_4  | 4.4 kHz |
| K_5_5  | 5.5 kHz |
| K_11   | 11 kHz  |
| K_22   | 22 kHz  |
| K_27_5 | 27.5 kHz|
| K_44   | 44 kHz  |
| K_55   | 55 kHz  |
| K_100  | 100 kHz |
| K_110  | 110 kHz |
| K_220  | 220 kHz |
| K_250  | 250 kHz |
| K_440  | 440 kHz |
| K_500  | 500 kHz |
| M_1    | 1 MHz   |
| M_2    | 2 MHz   |
| M_4    | 4 MHz   |
| M_80   | 80 MHz  |
| M_500  | 500 MHz |
| G_1    | 1 GHz   |
| FULL   | All range |

**Function**
Sets external modulation frequency range.

**Example**
Sets external modulation frequency range to 100 kHz.

> :SOURce:SYNThesizer:JITTer:MODulation:FREQuency:RANGe K_100

**Compatibility**
Incompatible with existing models.

### :SOURce:SYNThesizer:JITTer:MODulation:FREQuency:RANGe?

| Response  |  
|-----------------------------|-----------------------------|
| <range> = <CHARACTER RESPONSE DATA> |

**Function**
Queries external modulation frequency range.

**Example**
Queries external modulation frequency range.

> :SOURce:SYNThesizer:JITTer:MODulation:FREQuency:RANGe?

< K_100

**Compatibility**
Incompatible with existing models.
Chapter 7  SCPI Commands

:SOURce:SYNThesizer:JITTer:MODulation:AMPLitude:RANGe<range>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;range&gt; = &lt;CHARACTER PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>R_0_22</td>
<td>0.22 UI</td>
</tr>
<tr>
<td>R_2_0</td>
<td>2.00 UI</td>
</tr>
<tr>
<td>R_20</td>
<td>20 UI</td>
</tr>
<tr>
<td>R_200</td>
<td>200 UI</td>
</tr>
<tr>
<td>R_4000</td>
<td>4,000 UI</td>
</tr>
</tbody>
</table>

Function Sets external modulation amplitude.
Example Sets external modulation amplitude to 0.22 UI.
> :SOURce:SYNThesizer:JITTer:MODulation:AMPLitude:RANGe R_0_22
Compatibility Incompatible with existing models.

:SOURce:SYNThesizer:JITTer:MODulation:AMPLitude:RANGe?

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;range&gt; = &lt;CHARACTER RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>R_0_22</td>
<td>0.22 UI</td>
</tr>
<tr>
<td>R_2_0</td>
<td>2.00 UI</td>
</tr>
<tr>
<td>R_20</td>
<td>20 UI</td>
</tr>
<tr>
<td>R_200</td>
<td>200 UI</td>
</tr>
<tr>
<td>R_4000</td>
<td>4,000 UI</td>
</tr>
</tbody>
</table>

Function Queries external modulation amplitude.
Example > :SOURce:SYNThesizer:JITTer:MODulation:AMPLitude:RANGe?
< R_0_22
Compatibility Incompatible with existing models.

:SOURce:SYNThesizer:JITTer:TRGSource<range>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;range&gt; = &lt;CHARACTER PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>F_1</td>
<td>f/1</td>
</tr>
<tr>
<td>F_64</td>
<td>f/64</td>
</tr>
</tbody>
</table>

Function Selects trigger output division ratio.
Example Sets trigger output division ratio to f/1.
> :SOURce:SYNThesizer:JITTer:TRGSource F_1
Compatibility Incompatible with existing models.
## 7.2 Synthesizer Commands

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;range&gt; = &lt;CHARACTER RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>F_1</td>
<td>f/1</td>
</tr>
<tr>
<td>F_64</td>
<td>f/64</td>
</tr>
</tbody>
</table>

**Function**
Queries trigger output division ratio.

**Example**
```plaintext
> :SOURce:SYNThesizer:JITTer:TRGSource?
< F_1
```

**Compatibility**
Incompatible with existing models.
7.3 PPG Commands

This section describes the setting and query commands for the MU181020A 12.5 Gbit/s PPG and MU181020B 14 Gbit/s PPG. The commands related to each tab are described in separate sections below. Before executing a setting/query command for the MU181020A, specify the slot number of the module to be remotely controlled, by using the :MODule:ID command. Refer to Section 7.1 “Common Commands” and 9.1 "Basic Operation of Device Messages" for how to specify a slot number with the :MODule:ID command. However, unless there is a special item, see the MU181020A.

7.3.1 Commands related to Output tab

![Figure 7.3.1-1 Output tab]
7.3 PPG Commands

7.3.1.1 Output setting commands

![Output setting field](image)

**Figure 7.3.1.1-1 Output setting field**

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>:OUTPut:DATA:OUTPut?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:OUTPut:_CLOCK:OUTPut?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:OUTPut:OFFSet?</td>
</tr>
<tr>
<td>[4]</td>
<td>No label (Setting item selection)</td>
<td>:OUTPut:SET</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:OUTPut:SET?</td>
</tr>
</tbody>
</table>

### :OUTPut:DATA:OUTPut <boolean>

**Parameter**

<boolean> = <BOOLEAN PROGRAM DATA>

- OFF or 0: Output OFF
- ON or 1: Output ON

**Function**

Sets data output ON or OFF. The setting commonly applies to Data and XData.

**Example**

To set data output ON:

> :OUTPut:DATA:OUTPut ON

**Compatibility**

Incompatible with the MP1632C.

### :OUTPut:DATA:OUTPut?

**Response**

<numeric> = <NR1 NUMERIC RESPONSE DATA>

- 0: Output OFF
- 1: Output ON

**Function**

Queries the data output ON/OFF.

**Example**

> :OUTPut:DATA:OUTPut?

< 1

**Compatibility**

Incompatible with the MP1632C.
### :OUTPut:CLOCk:OUTPut <boolean>

**Parameter**
- `<boolean>` = `<BOOLEAN PROGRAM DATA>`
  - OFF or 0: Output OFF
  - ON or 1: Output ON

**Function**
Sets clock output ON or OFF. The setting commonly applies to Clock and XClock.

**Example**
To set clock output ON:
> :OUTPut:CLOCk:OUTPut ON

**Compatibility**
Incompatible with existing models.

### :OUTPut:CLOCk:OUTPut?

**Response**
- `<numeric>` = `<NR1 NUMERIC RESPONSE DATA>`
  - 0: Output OFF
  - 1: Output ON

**Function**
Queries the clock output ON/OFF state.

**Example**
> :OUTPut:CLOCk:OUTPut?
< 1

**Compatibility**
Incompatible with existing models.

### :OUTPut:OFFSet <offset>

**Parameter**
- `<offset>` = `<CHARACTER PROGRAM DATA>`
  - VOH: Based on the high level
  - VTH: Based on the center level
  - VOL: Based on the low level

**Function**
Sets the offset reference value for data and clock output.

**Example**
To set the offset reference to Voh:
> :OUTPut:OFFSet VOH

**Compatibility**
Incompatible with existing models.

### :OUTPut:OFFSet?

**Response**
- `<offset>` = `<CHARACTER RESPONSE DATA>`
  - VOH, VTH, VOL

**Function**
Queries the offset reference value for data and clock outputs.

**Example**
> :OUTPut:OFFSet?
< VOH

**Compatibility**
Incompatible with existing models.
### :OUTPut:SET <item>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;item&gt; = &lt;CHARACTER PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATA</td>
<td>Sets Data/XData.</td>
</tr>
<tr>
<td>CLOCk</td>
<td>Sets Clock/XClock.</td>
</tr>
<tr>
<td>CMU</td>
<td>Sets CMU Bitrate.</td>
</tr>
</tbody>
</table>

**Function**
Selects the data output, clock output, or CMU setting screen.

**Example**
To select the data output setting screen:

>` :OUTPut:SET DATA`

**Compatibility**
Incompatible with existing models.

### :OUTPut:SET? 

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;item&gt; = &lt;CHARACTER RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATA, CLOC, CMU</td>
<td></td>
</tr>
</tbody>
</table>

**Function**
Queries the selected setting screen.

**Example**

>` :OUTPut:SET?`

< DATA

**Compatibility**
Incompatible with existing models.
Chapter 7  SCPI Commands

7.3.1.2 Data/XData setting commands

Figure 7.3.1.2-1  Data/XData setting field

Figure 7.3.1.2-2  Level Guard Setting dialog box
### Table 7.3.1.2-1 Data/XData setting commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Tracking</td>
<td>:OUTPut:DATA:TRACking</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:OUTPut:DATA:TRACking?</td>
</tr>
<tr>
<td>2</td>
<td>Level Guard</td>
<td>:OUTPut:DATA:LEVGuard</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:OUTPut:DATA:LEVGuard?</td>
</tr>
<tr>
<td>3</td>
<td>Defined Interface</td>
<td>:OUTPut:DATA:LEVel</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:OUTPut:DATA:LEVel?</td>
</tr>
<tr>
<td>4</td>
<td>Amplitude</td>
<td>:OUTPut:DATA:AMPLitude</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:OUTPut:DATA:AMPLitude?</td>
</tr>
<tr>
<td>5</td>
<td>AC ON/OFF</td>
<td>:OUTPut:DATA:AOFFset</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:OUTPut:DATA:AOFFset?</td>
</tr>
<tr>
<td>6</td>
<td>Offset</td>
<td>:OUTPut:DATA:OFFSet</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:OUTPut:DATA:OFFSet?</td>
</tr>
<tr>
<td>7</td>
<td>External ATT Factor</td>
<td>:OUTPut:DATA:ATTFactor</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:OUTPut:DATA:ATTFactor?</td>
</tr>
<tr>
<td>8</td>
<td>Cross Point</td>
<td>:OUTPut:DATA:CPOint</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:OUTPut:DATA:CPOint?</td>
</tr>
<tr>
<td>9</td>
<td>Delay (mUI setting)</td>
<td>:OUTPut:DATA:UIPadjust</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:OUTPut:DATA:UIPadjust?</td>
</tr>
<tr>
<td>10</td>
<td>Delay (ps setting)</td>
<td>:OUTPut:DATA:PADJust</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:OUTPut:DATA:PADJust?</td>
</tr>
<tr>
<td>11</td>
<td>Delay (ps setting)</td>
<td>:OUTPut:DATA:CLOCk:DELay</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:OUTPut:DATA:CLOCk:DELay?</td>
</tr>
<tr>
<td>12</td>
<td>Relative</td>
<td>:OUTPut:DATA:RELative</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:OUTPut:DATA:RELative?</td>
</tr>
<tr>
<td>13</td>
<td>No label (Relative value setting)</td>
<td>:OUTPut:DATA:RDELay</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:OUTPut:DATA:RDELay?</td>
</tr>
<tr>
<td>14</td>
<td>Calibration</td>
<td>:OUTPut:DATA:PCALibration</td>
</tr>
<tr>
<td>15</td>
<td>Amplitude max</td>
<td>:OUTPut:DATA:LiMitter:AMPLitude</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:OUTPut:DATA:LiMitter:AMPLitude?</td>
</tr>
<tr>
<td>16</td>
<td>Offset max/min</td>
<td>:OUTPut:DATA:LiMitter:OFFSet</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:OUTPut:DATA:LiMitter:OFFSet?</td>
</tr>
<tr>
<td>17</td>
<td>No label (Query for status)</td>
<td>:CALCulate:DATA:MONitor?</td>
</tr>
<tr>
<td>18</td>
<td>No label (Query for data/clock output setting completion status)</td>
<td>:OUTPut:CHANge:CSTate?</td>
</tr>
<tr>
<td>19</td>
<td>Jitter Input</td>
<td>:OUTPut:DATA:JINPut</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:OUTPut:DATA:JINPut?</td>
</tr>
<tr>
<td>20</td>
<td>Unit Offset</td>
<td>:OUTPut:DATA:UDELay:OFFSet</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:OUTPut:DATA:UDELay:OFFSet?</td>
</tr>
</tbody>
</table>
Chapter 7  SCPI Commands

:OUTPut:DATA:TRACking <boolean>

Parameter  
<boolean> = <BOOLEAN PROGRAM DATA>
OFF or 0  Common setting OFF
ON or 1  Common setting ON

Function  
Sets the common setting for the Data and XData output ON or OFF.

Example  
To set the common setting ON:
> :OUTPut:DATA:TRACking ON

Compatibility  
Incompatible with existing models.

:OUTPut:DATA:TRACking?

Response  
<numeric> = <NR1 NUMERIC RESPONSE DATA>
0  Common setting OFF
1  Common setting ON

Function  
Queries the common setting for the Data and XData outputs ON/OFF state.

Example  
> :OUTPut:DATA:TRACking?
< 1

Compatibility  
Incompatible with existing models.

:OUTPut:DATA:LEVGuard <boolean>

Parameter  
<boolean> = <BOOLEAN PROGRAM DATA>
OFF or 0  Output range limitation OFF
ON or 1  Output range limitation ON

Function  
Sets the data output range limitation ON or OFF.

Example  
To set the data output range limitation ON:
> :OUTPut:DATA:LEVGuard ON

Compatibility  
Incompatible with existing models.

:OUTPut:DATA:LEVGuard?

Response  
<numeric> = <NR1 NUMERIC RESPONSE DATA>
0  Output range limitation OFF
1  Output range limitation ON

Function  
Queries the data output range limitation ON/OFF state.

Example  
> :OUTPut:DATA:LEVGuard?
< 1

Compatibility  
Incompatible with existing models.
7.3 PPG Commands

:OUTPut:DATA:LEVel <port>,<level>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;port&gt;</td>
<td>&lt;CHARACTER PROGRAM DATA&gt;</td>
</tr>
<tr>
<td>DATA</td>
<td>Data output</td>
</tr>
<tr>
<td>XDATa</td>
<td>XData output</td>
</tr>
<tr>
<td>&lt;level&gt;</td>
<td>&lt;CHARACTER PROGRAM DATA&gt;</td>
</tr>
<tr>
<td>VARiable</td>
<td>Variable</td>
</tr>
<tr>
<td>NECL</td>
<td>NECL level</td>
</tr>
<tr>
<td>PCML</td>
<td>PCML level</td>
</tr>
<tr>
<td>NCML</td>
<td>NCML level</td>
</tr>
<tr>
<td>SCFL</td>
<td>SCFL level</td>
</tr>
<tr>
<td>LVPecl</td>
<td>LVPECL level</td>
</tr>
<tr>
<td>LVDS200</td>
<td>LVDS200 mV</td>
</tr>
<tr>
<td>LVDS400</td>
<td>LVDS400 mV</td>
</tr>
</tbody>
</table>

Function: Sets the level of the fixed interface for the data output.
Example: To set the level of the XData output to the NECL level:
> :OUTPut:DATA:LEVel XDATa,NECL

Compatibility: Compatible with the MP1632C.

:OUTPut:DATA:LEVel? <port>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;port&gt;</td>
<td>&lt;CHARACTER PROGRAM DATA&gt;</td>
</tr>
<tr>
<td>DATA, XDATa</td>
<td></td>
</tr>
</tbody>
</table>

Response: 
| <level>     | <CHARACTER RESPONSE DATA> |
| VAR, PCML, NCML, SCFL, NECL, LVP, LVDS200, LVDS400 |

Function: Queries the level of the fixed interface for the data output.
Example: To query the level of the XData output:
> :OUTPut:DATA:LEVel? XDATa
< NECL

Compatibility: Compatible with the MP1632C.
### :OUTPut:DATA:AMPLitude <port>,<numeric>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;port&gt;</td>
<td>&lt;CHARACTER PROGRAM DATA&gt;</td>
</tr>
<tr>
<td>DATA</td>
<td>DATA output</td>
</tr>
<tr>
<td>XDATa</td>
<td>XData output</td>
</tr>
<tr>
<td>&lt;numeric&gt;</td>
<td>&lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
</tr>
<tr>
<td>0.050 to 0.800</td>
<td>0.050 to 0.800 V, in 0.002 V steps</td>
</tr>
<tr>
<td>0.250 to 2.500</td>
<td>0.250 to 2.500 V, in 0.002 V steps</td>
</tr>
<tr>
<td>0.050 to 2.000</td>
<td>0.050 to 2.000 V, in 0.002 V steps</td>
</tr>
<tr>
<td>0.500 to 3.500</td>
<td>0.500 to 3.500 V, in 0.002 V steps</td>
</tr>
</tbody>
</table>

**Function**
Sets the amplitude of the data output.

**Example**
To set the amplitude of the Data output to 1.000 V:
>`:OUTPut:DATA:AMPLitude DATA,1.000`

**Compatibility**
Compatible with the MP1632C.

### :OUTPut:DATA:AMPLitude? <port>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;port&gt;</td>
<td>&lt;CHARACTER PROGRAM DATA&gt;</td>
</tr>
<tr>
<td>DATA, XDATa</td>
<td></td>
</tr>
<tr>
<td>&lt;numeric&gt;</td>
<td>&lt;NR2 NUMERIC RESPONSE DATA&gt;</td>
</tr>
<tr>
<td>0.050 to 3.500</td>
<td>0.050 to 3.500 V</td>
</tr>
</tbody>
</table>

**Function**
Queries the amplitude of the data output.

**Example**
To query the amplitude of the Data output:
>`:OUTPut:DATA:AMPLitude? DATA`
< 1.000

**Compatibility**
Compatible with the MP1632C.

### :OUTPut:DATA:AOFFset <boolean>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;boolean&gt;</td>
<td>&lt;BOOLEAN PROGRAM DATA&gt;</td>
</tr>
<tr>
<td>OFF or 0</td>
<td>Offset OFF (DC output)</td>
</tr>
<tr>
<td>ON or 1</td>
<td>Offset ON (AC output)</td>
</tr>
</tbody>
</table>

**Function**
Sets the data output offset ON or OFF.

**Example**
To set the data output offset ON:
>`:OUTPut:DATA:AOFFset ON`

**Compatibility**
Incompatible with existing models.
### 7.3 PPG Commands

**:OUTPut:DATA:AOFFset?**

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;numeric&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Offset OFF</td>
</tr>
<tr>
<td>1</td>
<td>Offset ON</td>
</tr>
</tbody>
</table>

**Function**
Queries the data output offset ON/OFF state.

**Example**
```bash
> :OUTPut:DATA:AOFFset?
< 1
```

**Compatibility**
Incompatible with existing models.

### :OUTPut:DATA:OFFSet <port>, <numeric>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;port&gt; = &lt;CHARACTER PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATA</td>
<td>Data output</td>
</tr>
<tr>
<td>XDATa</td>
<td>XData output</td>
</tr>
<tr>
<td>&lt;numeric&gt;</td>
<td>&lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
</tr>
</tbody>
</table>

-2.000 to 3.300  
−2.000 to 3.300 Voh, in 0.001 V Step  
(MU181020A-x10, x11, x12, x13,  
MU181020B-x11, x12, x13)

-2.500 to 3.225  
−2.500 to 3.225 Vth, in 0.001 V Steps  
(MU181020A-x10)

-3.250 to 3.175  
−3.250 to 3.175 Vth, in 0.001 V Steps  
(MU181020A-x11, MU181020B-x11)

-3.000 to 3.275  
−3.000 to 3.275 Vth, in 0.001 V Steps  
(MU181020A-x12, MU181020B-x12)

-3.000 to 3.050  
−3.000 to 3.050 Vth, in 0.001 V Steps  
(MU181020A-x13, MU181020B-x13)

-3.000 to 3.250  
−3.000 to 3.250 Vol, in 0.001 V Steps  
(MU181020A-x10)

-4.500 to 3.050  
−4.500 to 3.050 Vol, in 0.001 V Steps  
(MU181020A-x11, MU181020B-x11)

-4.000 to 3.250  
−4.000 to 3.250 Vol, in 0.001 V Steps  
(MU181020A-x12, MU181020B-x12)

-4.000 to 2.800  
−4.000 to 2.800 Vol, in 0.001 V Steps  
(MU181020A-x13, MU181020B-x13)

**Function**
Sets the data output offset. The setting range varies depending on the offset reference setting.

**Example**
To set the XData output offset to 1.000 V (Voh):
```bash
> :OUTPut:DATA:OFFSet XDATa, 1.000
```

**Compatibility**
Compatible with the MP1632C.
Chapter 7  SCPI Commands

:OUTPut:DATA:OFFSet? <port>

Parameter  
\(<port> = \langle\text{CHARACTER PROGRAM DATA}\rangle\)
\(= \langle\text{DATA, XDATa}\rangle\)

Response  
\(<\text{numeric}> = \langle\text{NR2 NUMERIC RESPONSE DATA}\rangle\)
-2.000 to 3.300  
\(-2.000\) to \(3.300\) \(\text{Voh}\)
(MU181020A-x10, x11, x12 x13,  
MU181020B-x11, x12 x13)

-2.500 to 3.225  
\(-2.500\) to \(3.225\) \(\text{Vth}\) (MU181020A-x10)

-3.250 to 3.175  
\(-3.250\) to \(3.175\) \(\text{Vth}\)
(MU181020A-x11, MU181020B-x11)

-3.000 to 3.275  
\(-3.000\) to \(3.275\) \(\text{Vth}\)
(MU181020A-x12, MU181020B-x12)

-3.000 to 3.050  
\(-3.000\) to \(3.050\) \(\text{Vth}\)
(MU181020A-x13, MU181020B-x13)

-3.000 to 3.250  
\(-3.000\) to \(3.250\) \(\text{Vol}\) (MU181020A-x10)

-4.500 to 3.050  
\(-4.500\) to \(3.050\) \(\text{Vol}\)
(MU181020A-x11, MU181020B-x11)

-4.000 to 3.250  
\(-4.000\) to \(3.250\) \(\text{Vol}\)
(MU181020A-x12, MU181020B-x12)

-4.000 to 2.800  
\(-4.000\) to \(2.800\) \(\text{Vol}\)
(MU181020A-x13, MU181020B-x13)

Function  
Queries the data output offset.

Example  
To query the XData output offset:
> :OUTPut:DATA:OFFSet? XDATa
< 1.000

Compatibility  
Compatible with the MP1632C.

:OUTPut:DATA:ATTFactor <port>,<numeric>

Parameter  
\(<port> = \langle\text{CHARACTER PROGRAM DATA}\rangle\)
\(= \langle\text{DATA, XDATa}\rangle\)
\(<\text{numeric}> = \langle\text{DECIMAL NUMERIC PROGRAM DATA}\rangle\)

Function  
Sets the External ATT Factor of the data output.

Example  
To set the External ATT Factor of the data output to 20 dB:
> :OUTPut:DATA:ATTFactor DATA,20

Compatibility  
Incompatible with existing models.
### :**OUTPut:DATA:ATTFactor? <port>**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;port&gt; = &lt;CHARACTER PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>DATA, XDATa</td>
</tr>
<tr>
<td>Response</td>
<td>&lt;numeric&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt;</td>
</tr>
<tr>
<td></td>
<td>0 to 40</td>
</tr>
<tr>
<td></td>
<td>0 to 40 dB, in 1 dB steps</td>
</tr>
<tr>
<td>Function</td>
<td>Queries the External ATT Factor of the Data output.</td>
</tr>
<tr>
<td>Example</td>
<td>&gt; :OUTPut:DATA:ATTFactor? DATA</td>
</tr>
<tr>
<td></td>
<td>&lt; 20</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Incompatible with existing models.</td>
</tr>
</tbody>
</table>

### :**OUTPut:DATA:CPOint <port>,<numeric>**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;port&gt; = &lt;CHARACTER PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>DATA Data output</td>
</tr>
<tr>
<td></td>
<td>XDATa XData output</td>
</tr>
<tr>
<td>&lt;numeric&gt;</td>
<td>&lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
</tr>
<tr>
<td></td>
<td>30 to 70</td>
</tr>
<tr>
<td></td>
<td>30 to 70%, in 1% steps</td>
</tr>
<tr>
<td></td>
<td>(MU181020A-x10, x11, MU181020B-x11)</td>
</tr>
<tr>
<td></td>
<td>20 to 80</td>
</tr>
<tr>
<td></td>
<td>20 to 80%, in 1% steps</td>
</tr>
<tr>
<td></td>
<td>(MU181020A-x12, MU181020B-x12)</td>
</tr>
<tr>
<td></td>
<td>20.0 to 90.0</td>
</tr>
<tr>
<td></td>
<td>20.0 to 90.0%, in 0.1% steps</td>
</tr>
<tr>
<td></td>
<td>(MU181020A-x13, MU181020B-x13)</td>
</tr>
<tr>
<td>Function</td>
<td>Sets the data output crosspoint.</td>
</tr>
<tr>
<td>Example</td>
<td>To set the crosspoint of the XData output to 60%:</td>
</tr>
<tr>
<td></td>
<td>&gt; :OUTPut:DATA:CPOint XDATa,60</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Compatible with some parameters of the MP1632C (25 to 75%, in 1% steps).</td>
</tr>
</tbody>
</table>
Chapter 7  SCPI Commands

**:OUTPut:DATA:CPOint? <port>**

**Parameter**

\(<\text{port}> = \langle\text{CHARACTER PROGRAM DATA}\rangle\)

DATA, XDAta

**Response**

\(<\text{numeric}> = \langle\text{NR2 NUMERIC RESPONSE DATA}\rangle\)

20 to 80 20 to 80%, in 1% steps

\((\text{MU181020A-x10, x11, x12, MU181020B-x11, x12})\)

\(<\text{numeric}> = \langle\text{NR2 NUMERIC RESPONSE DATA}\rangle\)

20.0 to 90.0 20.0 to 90.0%, in 0.1% steps

\((\text{MU181020A-x13, MU181020B-x13})\)

**Function**

Queries the data output crosspoint.

**Example**

To query the crosspoint of the XData output:

> :OUTPut:DATA:CPOint? XDAta

< 60

**Compatibility**

Compatible with some parameters of the MP1632C (25 to 75%, in 1% steps).

**:OUTPut:DATA:UIPadjust <numeric>**

**Parameter**

\(<\text{numeric}> = \langle\text{DECIMAL NUMERIC PROGRAM DATA}\rangle\)

\(-1000 \text{ to } 1000 \text{ mUI, in } 1 \text{ mUI steps}\)

\(-64000 \text{ to } 64000 \text{ mUI, in } 1 \text{ mUI steps}\)

(In the case of Combination)

**Function**

Sets the common phase for the Data and XData outputs in mUI units.

**Example**

To set the common phase to 500 mUI:

> :OUTPut:DATA:UIPadjust 500

**Compatibility**

Incompatible with existing models.

**:OUTPut:DATA:UIPadjust?**

**Response**

\(<\text{numeric}> = \langle\text{NR1 NUMERIC RESPONSE DATA}\rangle\)

\(-1000 \text{ to } 1000 \text{ mUI, in } 1 \text{ mUI steps}\)

\(-64000 \text{ to } 64000 \text{ mUI, in } 1 \text{ mUI steps}\)

(In the case of Combination)

**Function**

Queries the common phase for the Data and XData outputs in mUI units.

**Example**

To query the common phase in mUI units:

> :OUTPut:DATA:UIPadjust?

< 500

**Compatibility**

Incompatible with existing models.
7.3 PPG Commands

:OUTPut:DATA:PADJust <numeric>

Parameter  
<numeric> = <DECIMAL NUMERIC PROGRAM DATA>
-10000 to 10000  -10000 to 10000 ps, in 10 ps steps (at 0.1 GHz)
-80 to 80  -80 to 80 ps, in 0.08 ps steps (at 12.5 GHz)
Converted from the calculated frequency value, based on the setting resolution of mUI.

Function  
Sets the common phase for the Data and XData outputs in ps units.
The specified value may not be exactly set as is, due to the limitation of the setting resolution. In this event, the value is adjusted to the most approximate value and set.

Example  
To set the common phase to 100 ps:
> :OUTPut:DATA:PADJust 100

Compatibility  
Incompatible with existing models.

:OUTPut:DATA:PADJust?

Response  
<numeric> = <NR2 NUMERIC RESPONSE DATA>
-10000 to 10000  -10000 to 10000 ps, in 10 ps steps (at 0.1 GHz)
-80 to 80  -80 to 80 ps, in 0.08 ps steps (at 12.5 GHz)
Converted from the calculated frequency value, based on the setting resolution of mUI.

Function  
Queries the common phase for the Data and XData outputs in ps units.

Example  
To query the common phase in ps units:
> :OUTPut:DATA:PADJust?
< 100

Compatibility  
Incompatible with existing models.

:OUTPut:CLOCk:DELay <numeric>

Parameter  
<numeric> = <DECIMAL NUMERIC PROGRAM DATA>
-10000 to 10000  -10000 to 10000 ps, in 10 ps steps (at 0.1 GHz)
-80 to 80  -80 to 80 ps, in 0.08 ps steps (at 12.5 GHz)
Converted from the calculated frequency value, based on the setting resolution of mUI.

Function  
Sets the data output delay in ps units.
The specified value may not be exactly set as is, due to the limitation of the setting resolution. In this event, the value is adjusted to the most approximate value and set.

Example  
To set the data output delay to 100 ps:
> :OUTPut:CLOCk:DELay 100

Compatibility  
Compatible with the MP1632C.
Chapter 7  SCPI Commands

:OUTPut:CLOCk:DELay?

Response  <numeric> = <NR2 NUMERIC RESPONSE DATA>
-10000 to 10000  -10000 to 10000 ps, in 10 ps steps (at 0.1 GHz)
-80 to 80  -80 to 80 ps, in 0.08 ps steps (at 12.5 GHz)
Converted from the calculated frequency value, based on the setting resolution of mUI.

Function  Queries the data output delay in ps units.
Example  To query the data output delay in ps units:
> :OUTPut:CLOCk:DELay?
< 100

Compatibility  Compatible with the MP1632C.

:OUTPut:DATA:RELative <boolean>

Parameter  <boolean> = <BOOLEAN PROGRAM DATA>
0 or OFF  Reference OFF
1 or ON  Reference ON

Function  Sets the data output delay reference ON or OFF.
Example  To set the data output delay reference ON:
> :OUTPut:DATA:RELative ON

Compatibility  Incompatible with existing models.

:OUTPut:DATA:RELative?

Response  <numeric> = <NR1 NUMERIC RESPONSE DATA>
0  Reference OFF
1  Reference ON

Function  Queries the data output delay reference setting (ON/OFF).
Example  To query the data output delay reference setting:
> :OUTPut:DATA:RELative?
< 1

Compatibility  Incompatible with existing models.
### :OUTPut:DATA:RDELay <numeric>[,<unit>]

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;numeric&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
<td>(-2000\text{ to }2000) from (-2000\text{ to }2000) mUI, in 1 mUI steps</td>
</tr>
<tr>
<td>XXXXX.XX</td>
<td>Converted from the calculated frequency value, based on the setting resolution of mUI.</td>
</tr>
<tr>
<td>[&lt;unit&gt;] = &lt;CHARACTER PROGRAM DATA&gt;</td>
<td>UI: mUI units, PS: ps units. The ps unit is selected when [&lt;unit&gt;] is omitted.</td>
</tr>
</tbody>
</table>

**Function**
Sets the value and unit of the data output delay reference. Set a value using a difference from the reference value. The specified value may not be exactly set as is when the ps unit is set, due to the limitation of the setting resolution. The value is adjusted to the most approximate value and set in this event.

**Example**
To set the value for the data output delay reference to \(-1000\) mUI:
```
> :OUTPut:DATA:RDELay -1000,UI
```

**Compatibility**
Incompatible with existing models.

### :OUTPut:DATA:RDELay? [<unit>]

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>[&lt;unit&gt;] = &lt;CHARACTER PROGRAM DATA&gt;</td>
<td>UI: mUI units, PS: ps units. The ps unit is selected when [&lt;unit&gt;] is omitted.</td>
</tr>
</tbody>
</table>

**Response**

| <numeric> = <NR1 NUMERIC RESPONSE DATA> | \(-2000\text{ to }2000\) from \(-2000\text{ to }2000\) mUI, in 1 mUI steps |
| XXXXX.XX | Converted from the calculated frequency value, based on the setting resolution of mUI. |

**Function**
Queries the value and unit of the data output delay reference.

**Example**
To query the value of the data output delay reference in UI units:
```
> :OUTPut:DATA:RDELay? UI
< -1000
```

**Compatibility**
Incompatible with existing models.

### :OUTPut:DATA:PCALibration

**Function**
Executes calibration for Data and XData output phases.

**Example**
```
> :OUTPut:DATA:PCALibration
```

**Compatibility**
Incompatible with existing models.
Chapter 7  SCPI Commands

:OUTPut:DATA:LIMiter:AMPLitude <numeric>

Parameter

<table>
<thead>
<tr>
<th>&lt;numeric&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.050 to 0.800</td>
</tr>
<tr>
<td>(MU181020A-x10)</td>
</tr>
<tr>
<td>0.250 to 2.500</td>
</tr>
<tr>
<td>(MU181020A-x11, MU181020B-x11)</td>
</tr>
<tr>
<td>0.050 to 2.000</td>
</tr>
<tr>
<td>(MU181020A-x12, MU181020B-x12)</td>
</tr>
<tr>
<td>0.500 to 3.500</td>
</tr>
<tr>
<td>(MU181020A-x13, MU181020B-x13)</td>
</tr>
</tbody>
</table>

Function
Sets the limitation value for the data output amplitude.

Example
To set the limitation value for the data output amplitude to 2.000 V:

> :OUTPut:DATA:LIMiter:AMPLitude 2.000

Compatibility
Incompatible with existing models.

:OUTPut:DATA:LIMiter:AMPLitude?

Response

<table>
<thead>
<tr>
<th>&lt;numeric&gt; = &lt;NR2 NUMERIC RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.050 to 0.800</td>
</tr>
<tr>
<td>0.250 to 2.500</td>
</tr>
<tr>
<td>(MU181020A-x11, MU181020B-x11)</td>
</tr>
<tr>
<td>0.050 to 2.000</td>
</tr>
<tr>
<td>(MU181020A-x12, MU181020B-x12)</td>
</tr>
<tr>
<td>0.500 to 3.500</td>
</tr>
<tr>
<td>(MU181020A-x13, MU181020B-x13)</td>
</tr>
<tr>
<td>In 0.002 V steps</td>
</tr>
</tbody>
</table>

Function
Queries the limitation value for the data output amplitude.

Example

> :OUTPut:DATA:LIMiter:AMPLitude?

< 2.000

Compatibility
Incompatible with existing models.
### :OUTPut:DATA:LIMiter:OFFSet <max>,<min>

#### Parameter
- **<max>** = <DECIMAL NUMERIC PROGRAM DATA>
  - -2.000 to 3.300
    - -2.000 to 3.300 Voh, in 0.001 V steps
      - (MU181020A-x10, x11, x12, x13, MU181020B-x11, x12, x13)
- **<min>** = <DECIMAL NUMERIC PROGRAM DATA>
  - -2.800 to 3.250
    - -2.800 to 3.250 Vol, in 0.001 V steps
      - (MU181020A-x10)
  - -4.500 to 3.050
    - -4.500 to 3.050 Vol, in 0.001 V steps
      - (MU181020A-x11, MU181020B-x11)
  - -4.000 to 3.250
    - -4.000 to 3.250 Vol, in 0.001 V steps
      - (MU181020A-x12, MU181020B-x12)
  - -4.000 to 2.800
    - -4.000 to 2.800 Vol, in 0.001 V steps
      - (MU181020A-x13, MU181020B-x13)

#### Function
Sets the limitation value for the data output offset.

#### Example
To set the limitation value for the data output offset to 3.000 Voh max.,
- 2.000 Vol min.:
  > :OUTPut:DATA:LIMiter:OFFSet 3.000, -2.000

#### Compatibility
Incompatible with existing models.

### :OUTPut:DATA:LIMiter:OFFSet?

#### Response
- **<max>** = <NR2 NUMERIC RESPONSE DATA>
  - -2.000 to 3.300
    - -2.000 to 3.300 Voh
      - (MU181020A-x10, x11, x12, x13, MU181020B-x11, x12, x13)
- **<min>** = <NR2 NUMERIC RESPONSE DATA>
  - -2.800 to 3.250
    - -2.800 to 3.250 Vol (MU181020A-x10)
  - -4.500 to 3.050
    - -4.500 to 3.050 Vol (MU181020A-x11, MU181020B-x11)
  - -4.000 to 3.250
    - -4.000 to 3.250 Vol (MU181020A-x12, MU181020B-x12)
  - -4.000 to 2.800
    - -4.000 to 2.800 Vol (MU181020A-x13, MU181020B-x13)

#### Function
Queries the limitation value for the data output offset.

#### Example
> :OUTPut:DATA:LIMiter:OFFSet?
< 3.000,-2.000

#### Compatibility
Incompatible with existing models.
Chapter 7  SCPI Commands

:CALCulate:DATA:MONitor? <item>

Parameter  

| <item> = <STRING RESPONSE DATA> |
| "DBS"   | Delay busy     |
| "CAL"   | Required Calibration |

Response  

| <string> = <STRING RESPONSE DATA> |

<table>
<thead>
<tr>
<th>Form</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Form1</td>
<td>&quot;Occur&quot;</td>
<td>When an alarm occurs</td>
</tr>
<tr>
<td>&quot;Not occur&quot;</td>
<td>When no alarm occurs</td>
<td></td>
</tr>
<tr>
<td>&quot;-------&quot;</td>
<td>When no data corresponds to the query</td>
<td></td>
</tr>
</tbody>
</table>

Function  
Queries the alarm occurrence state for the monitoring item corresponding to the parameter.

Example  
To query the Delay Error occurrence state:  
> :CALCulate:DATA:MONitor? "DBS"  
< "Occur"

Compatibility  
Incompatible with existing models.

:OUTPut:CHANge:CSTate?

Response  

| <state> = <NR1 NUMERIC RESPONSE DATA> |
| 0  | Amplitude and Offset settings of Data/Xdata and Clock/Xclock have been changed completely. |
| 1  | Amplitude or Offset setting of Data/Xdata or Clock/Xclock is being changed. |

Function  
Queries the completion status of changing of data output and clock output settings.

Example  
> :OUTPut:CHANge:CSTate?  
< 1

Compatibility  
Incompatible with existing models.

:OUTPut:DATA:JINPut <boolean>

Parameter  

| <boolean> = <BOOLEAN PROGRAM DATA> |
| 0 or OFF | Jitter input OFF setting |
| 1 or ON  | Jitter input ON setting |

Function  
When a jitter is added to the input signal, the Feedback process for Delay must be set to OFF.  
When a jitter is added, set the Jitter Input button to ON, using this command. When a jitter is not added, set the Jitter Input button to OFF.

Example  
To set the Jitter Input button to ON when a jitter is added:  
> :OUTPut:DATA:JINPut ON

Compatibility  
Incompatible with existing models.
7.3 PPG Commands

:OUTPut:DATA:JINPut?

Response

<numeric> = <NR1 NUMERIC RESPONSE DATA>

0  Jitter input OFF setting
1  Jitter input ON setting

Function
Queries the Jitter Input button setting.

Example
> :OUTPut:DATA:JINPut?
< 1

Compatibility
Incompatible with existing models.

:OUTPut:DATA:UDELay:OFFSet <numeric>

Parameter

<numeric> = <DECIMAL NUMERIC PROGRAM DATA>

–64000 to 64000  –64000 to 64000 mUI/1 mUI Step

Function
Sets the unit of the Delay offset value.

Example
To set the Delay offset to 500 mUI.
> :OUTPut:DATA:UDELay:OFFSet 500

Compatibility
Incompatible with existing models.

:OUTPut:DATA:UDELay:OFFSet?

Response

<numeric> = <NR2 NUMERIC RESPONSE DATA>

–64000 to 64000  –64000 to 64000 mUI/1 mUI Step

Function
Queries the Delay offset value.

Example
> :OUTPut:DATA:UDELay:OFFSet?
< 500

Compatibility
Incompatible with existing models.
7.3.1.3 Clock/XClock setting commands

![Figure 7.3.1.3-1 Clock/XClock setting field](image1)

![Figure 7.3.1.3-2 Level Guard setting dialog box](image2)
<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>:OUTPut:CLOCk:TRACking?</td>
</tr>
<tr>
<td>[2]</td>
<td>Level Guard</td>
<td>:OUTPut:CLOCk:LEVGuard</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:OUTPut:CLOCk:LEVGuard?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:OUTPut:CLOCk:LEVel?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:OUTPut:CLOCk:AMPLitude?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:OUTPut:CLOCk:AOFFset?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:OUTPut:CLOCk:OFFSet?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:OUTPut:CLOCk:ATTFactor?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:OUTPut:CLOCk:DUTY?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:OUTPut:CLOCk:LI Mitter:AMPLitude?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:OUTPut:CLOCk:LI Mitter:OFFSet?</td>
</tr>
<tr>
<td>[11]</td>
<td>No label (Query for data/clock output setting completion status)</td>
<td>:OUTPut:CHANge:CSTate?</td>
</tr>
</tbody>
</table>

Refer to Section 7.3.1.2 “Data/XData setting commands” for details on this command.
### :OUTPut:CLOCk:TRACking <boolean>

**Parameter**

<boolean> = <BOOLEAN PROGRAM DATA>

- OFF or 0  Common setting OFF
- ON or 1  Common setting ON

**Function**

Sets the common setting for the Clock and XClock output ON or OFF.

**Example**

To set the common setting ON:

```plaintext
> :OUTPut:CLOCk:TRACking ON
```

**Compatibility**

Incompatible with existing models.

### :OUTPut:CLOCk:TRACking?

**Response**

<numeric> = <NR1 NUMERIC RESPONSE DATA>

- 0  Common setting OFF
- 1  Common setting ON

**Function**

Queries the common setting for the Clock and XClock outputs ON/OFF state.

**Example**

```plaintext
> :OUTPut:CLOCk:TRACking?
< 1
```

**Compatibility**

Incompatible with existing models.

### :OUTPut:CLOCk:LEVGuard <boolean>

**Parameter**

<boolean> = <BOOLEAN PROGRAM DATA>

- OFF or 0  Output range limitation OFF
- ON or 1  Output range limitation ON

**Function**

Sets the click output range limitation ON or OFF.

**Example**

To set the clock output range limitation ON:

```plaintext
> :OUTPut:CLOCk:LEVGuard ON
```

**Compatibility**

Incompatible with existing models.

### :OUTPut:CLOCk:LEVGuard?

**Response**

<numeric> = <NR1 NUMERIC RESPONSE DATA>

- 0  Output range limitation OFF
- 1  Output range limitation ON

**Function**

Queries the clock output range limitation ON/OFF state.

**Example**

```plaintext
> :OUTPut:CLOCk:LEVGuard?
< 1
```

**Compatibility**

Incompatible with existing models.
### :OUTPut:CLOCk:LEVEL \(<port>,<level>\)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(&lt;port&gt;)</td>
<td>(&lt;\text{CHARACTER PROGRAM DATA}&gt;)</td>
</tr>
<tr>
<td>(\text{CLOCk})</td>
<td>Clock output</td>
</tr>
<tr>
<td>(\text{XCLK})</td>
<td>XClock output</td>
</tr>
<tr>
<td>(&lt;level&gt;)</td>
<td>(&lt;\text{CHARACTER PROGRAM DATA}&gt;)</td>
</tr>
<tr>
<td>(\text{VAR})</td>
<td>Variable</td>
</tr>
<tr>
<td>(\text{NECL})</td>
<td>NECL level</td>
</tr>
<tr>
<td>(\text{PCML})</td>
<td>PCML level</td>
</tr>
<tr>
<td>(\text{NCML})</td>
<td>NCML level</td>
</tr>
<tr>
<td>(\text{SCFL})</td>
<td>SCFL level</td>
</tr>
<tr>
<td>(\text{LVPECL})</td>
<td>LVPECL level</td>
</tr>
<tr>
<td>(\text{LVDS200})</td>
<td>LVDS200 mV</td>
</tr>
<tr>
<td>(\text{LVDS400})</td>
<td>LVDS400 mV</td>
</tr>
</tbody>
</table>

**Function**: Sets the level of the fixed interface for the clock output.

**Example**: To set the level of the Clock output to the NECL level:

```plaintext
> :OUTPut:CLOCk:LEVEL CLOCk,NECL
```

**Compatibility**: Incompatible with existing models.

### :OUTPut:CLOCk:LEVEL? \(<port>\)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(&lt;port&gt;)</td>
<td>(&lt;\text{CHARACTER PROGRAM DATA}&gt;)</td>
</tr>
<tr>
<td>(\text{CLOCk}, \text{XCLK})</td>
<td></td>
</tr>
</tbody>
</table>

**Response**: \(<level>\) = \(<\text{CHARACTER RESPONSE DATA}>\)

VAR,NECL,PCML,NCML,SCFL,LVP,LVDS200,LVDS400

**Function**: Queries the level of the fixed interface for the clock output.

**Example**: To get the level of the Clock output to the NECL level:

```plaintext
> :OUTPut:CLOCk:LEVEL? CLOCk
< NECL
```

**Compatibility**: Incompatible with existing models.

### :OUTPut:CLOCk:AMPLitude \(<port>,<numeric>\)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(&lt;port&gt;)</td>
<td>(&lt;\text{CHARACTER PROGRAM DATA}&gt;)</td>
</tr>
<tr>
<td>(\text{CLOCk})</td>
<td>Clock output</td>
</tr>
<tr>
<td>(\text{XCLK})</td>
<td>XClock output</td>
</tr>
<tr>
<td>(&lt;numeric&gt;)</td>
<td>(&lt;\text{DECIMAL NUMERIC PROGRAM DATA}&gt;)</td>
</tr>
<tr>
<td>0.100 to 2.000</td>
<td>0.100 to 2.000 V, in 0.002 V steps</td>
</tr>
</tbody>
</table>

**Function**: Sets the amplitude of the clock output.

**Example**: To set the amplitude of the Clock output to 1.000 V:

```plaintext
> :OUTPut:CLOCk:AMPLitude CLOCk,1.000
```

**Compatibility**: Compatible with the MP1632C.
Chapter 7  SCPI Commands

:OUTPut:CLOCk:AMPLitude? <port>

| Parameter | <port> = <CHARACTER PROGRAM DATA>  
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>CLOCk, XCLock</td>
<td></td>
</tr>
</tbody>
</table>

| Response | <numeric> = <NR2 NUMERIC RESPONSE DATA>  
|----------|------------------------------------------|
| 0.100 to 2.000 | 0.100 to 2.000 V, in 0.002 V steps  
| (MU181020A-x21, MU181020B-x21) |

| Function | Queries the amplitude of the clock output. |
| Example | To query the amplitude of the Clock output:  
| > :OUTPut:CLOCk:AMPLitude? CLOCk  
| < 1.000 |

| Compatibility | Compatible with the MP1632C. |

:OUTPut:CLOCk:AOFFset <boolean>

| Parameter | <boolean> = <BOOLEAN PROGRAM DATA>  
|-----------|----------------------------------|
| OFF or 0 | Offset OFF (DC output)  
| ON or 1 | Offset ON (AC output) |

| Function | Sets the clock output offset ON or OFF |
| Example | To set the clock output offset ON:  
| > :OUTPut:CLOCk:AOFFset ON |

| Compatibility | Incompatible with existing models. |

:OUTPut:CLOCk:AOFFset?

| Response | <numeric> = <NR1 NUMERIC RESPONSE DATA>  
|----------|------------------------------------------|
| 0 | Offset OFF  
| 1 | Offset ON |

| Function | Queries the clock output offset ON/OFF state. |
| Example | To query the clock output offset ON/OFF state:  
| > :OUTPut:CLOCk:AOFFset?  
| < 1 |

| Compatibility | Incompatible with existing models. |
### 7.3 PPG Commands

**:OUTPut:CLOCk:OFFSet <port>,<numeric>**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;port&gt; = CHARACTER PROGRAM DATA</td>
<td>CLOCk = Clock output, XCLock = XClock output</td>
</tr>
<tr>
<td>&lt;numeric&gt; = DECIMAL NUMERIC PROGRAM DATA</td>
<td>-2.000 to 3.300 = -2.000 to 3.300 Voh, in 0.001 V steps</td>
</tr>
<tr>
<td></td>
<td>-3.000 to 3.250 = -3.000 to 3.250 Vth, in 0.001 V steps</td>
</tr>
<tr>
<td></td>
<td>-4.000 to 3.200 = -4.000 to 3.200 Vol, in 0.001 V steps</td>
</tr>
</tbody>
</table>

**Function**: Sets the Clock output offset (for MU181020A-x21, MU181020B-x21 only).

**Example**: To set the XClock output offset to -1.000 V (Voh):

```plaintext
>:OUTPut:CLOCk:OFFSet XCLock,-1.000
```

**Compatibility**: Partially compatible with the MP1632C.

---

**:OUTPut:CLOCk:OFFSet? <port>**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;port&gt; = CHARACTER PROGRAM DATA</td>
<td>CLOCk, XCLock</td>
</tr>
</tbody>
</table>

**Response**: NR2 NUMERIC RESPONSE DATA |

<table>
<thead>
<tr>
<th>&lt;numeric&gt;</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-2.000 to 3.300</td>
<td>-2.000 to 3.300 Voh</td>
</tr>
<tr>
<td>-3.000 to 3.250</td>
<td>-3.000 to 3.250 Vth</td>
</tr>
<tr>
<td>-4.000 to 3.200</td>
<td>-4.000 to 3.200 Vol</td>
</tr>
</tbody>
</table>

**Function**: Queries the Clock output offset.

**Example**: To query the XClock output offset:

```plaintext
>:OUTPut:CLOCk:OFFSet? XCLock
< -1.000
```

**Compatibility**: Partially compatible with the MP1632C.

---

**:OUTPut:CLOCk:ATTFactor <port>,<numeric>**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;port&gt; = CHARACTER PROGRAM DATA</td>
<td>CLOCk = Clock output, XCLock = XClock output</td>
</tr>
<tr>
<td>&lt;numeric&gt; = DECIMAL NUMERIC PROGRAM DATA</td>
<td>0 to 40 = 0 to 40 dB, in 1 dB steps</td>
</tr>
</tbody>
</table>

**Function**: Sets the External ATT factor of the Clock output.

**Example**: To set the External ATT factor of the Clock output to 20 dB:

```plaintext
>:OUTPut:CLOCk:ATTFactor CLOCk,20
```

**Compatibility**: Incompatible with existing models.
Chapter 7  SCPI Commands

:OUTPut:CLOCk:ATTFactor? <port>
Parameter  <port> = <CHARACTER PROGRAM DATA>
            CLOCk, XCLock
Response   <numeric> = <NR1 NUMERIC RESPONSE DATA>
            0 to 40  0 to 40 dB
Function   Queries the External ATT factor of the Clock output.
Example    To query the External ATT factor of the Clock output:
            > :OUTPut:CLOCk:ATTFactor? CLOCk
            < 20
Compatibility  Incompatible with existing models.

:OUTPut:CLOCk:DUTY <numeric>
Parameter  <numeric> = <DECIMAL NUMERIC PROGRAM DATA>
            –25 to 25  –25 to 25, in single steps
Function   Sets the duty of the clock output.
Example    To set the duty of the clock output to 25:
            > :OUTPut:CLOCk:DUTY 25
Compatibility  Incompatible with existing models.

:OUTPut:CLOCk:DUTY?
Response   <numeric> = <NR1 NUMERIC RESPONSE DATA>
            –25 to 25  –25 to 25
Function   Queries the duty of the clock output.
Example    > :OUTPut:CLOCk:DUTY?
            < 25
Compatibility  Incompatible with existing models.

:OUTPut:CLOCk:LIMitter:AMPLitude <numeric>
Parameter  <numeric> = <DECIMAL NUMERIC PROGRAM DATA>
            0.100 to 2.000  0.100 to 2.000 V, in 0.002 V steps
            (MU181020A-x21, MU181020B-x21)
Function   Sets the limitation value for the clock output amplitude.
Example    To set the limitation value for the clock output amplitude to 2.000 V:
            > :OUTPut:CLOCk:LIMitter:AMPLitude 2.000
Compatibility  Incompatible with existing models.
7.3 PPG Commands

:OUTPut:CLOCK:LIMitter:AMPLitude?

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;numeric&gt; = &lt;NR2 NUMERIC RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.100 to 2.000</td>
</tr>
<tr>
<td></td>
<td>0.100 to 2.000 V</td>
</tr>
<tr>
<td></td>
<td>(MU181020A-x21, MU181020B-x21)</td>
</tr>
</tbody>
</table>

Function
Queries the limitation value for the clock output amplitude.

Example
> :OUTPut:CLOCK:LIMitter:AMPLitude?
< 2.000

Compatibility
Incompatible with existing models.

:OUTPut:CLOCK:LIMitter:OFFSet <max>,<min>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;max&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>−2.000 to 3.300</td>
</tr>
<tr>
<td></td>
<td>−2.000 to 3.300 V, in 0.001 V steps</td>
</tr>
<tr>
<td></td>
<td>&lt;min&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
</tr>
<tr>
<td></td>
<td>−4.000 to 3.200</td>
</tr>
<tr>
<td></td>
<td>−4.000 to 3.200 V, in 0.001 V steps</td>
</tr>
</tbody>
</table>

Function
Sets the limitation value for the clock output offset (for MU181020A-x21, MU181020B-x21 only).

Example
To set the limitation value for the clock output offset to 3.000 V max., −4.000 V min.:
> :OUTPut:CLOCK:LIMitter:OFFSet 3.000, −4.000

Compatibility
Incompatible with existing models.

:OUTPut:CLOCK:LIMitter:OFFSet?

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;max&gt; = &lt;NR2 NUMERIC RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>−2.000 to 3.300</td>
</tr>
<tr>
<td></td>
<td>−2.000 to 3.300 V</td>
</tr>
<tr>
<td></td>
<td>&lt;min&gt; = &lt;NR2 NUMERIC RESPONSE DATA&gt;</td>
</tr>
<tr>
<td></td>
<td>−4.000 to 3.200</td>
</tr>
<tr>
<td></td>
<td>−4.000 to 3.200 V</td>
</tr>
</tbody>
</table>

Function
Queries the limitation value for the clock output offset.

Example
> :OUTPut:CLOCK:LIMitter:OFFSet?
< 3.000,−4.000

Compatibility
Incompatible with existing models.
7.3.1.4 CMU Bit Rate setting commands

Table 7.3.1.4-1 CMU Bit Rate setting commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>:OUTPut:CMU:REFClock?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:OUTPut:CMU:EXTClock?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:OUTPut:CMU:FREQuency?</td>
</tr>
<tr>
<td></td>
<td>(Unit setting)</td>
<td>:OUTPut:CMU:RESolution?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:OUTPut:CMU:OPERation?</td>
</tr>
</tbody>
</table>
7.3 PPG Commands

**:OUTPut:CMU:REFClock <mode>**

Parameter

<mode> = <CHARACTER PROGRAM DATA>
- INTernal: Uses the internal reference signal.
- EXTernal: Uses the external reference signal.

Function

Selects whether to use the internal reference signal or the external reference signal for the clock reference signal.

Example

To select the internal reference signal (Internal) for the clock reference signal:

```
> :OUTPut:CMU:REFClock INTernal
```

Compatibility

Incompatible with existing models.

**:OUTPut:CMU:REFClock?**

Response

<mode> = <CHARACTER RESPONSE DATA>
- INT, EXT

Function

Queries the clock reference signal.

Example

```
> :OUTPut:CMU:REFClock?
< INT
```

Compatibility

Incompatible with existing models.

**:OUTPut:CMU:EXTClock <source>**

Parameter

<source> = <CHARACTER PROGRAM DATA>
- 1_1: 1/1 clock
- 1_64: 1/64 clocks

Function

Sets the division cycle when the external reference signal is used.

Example

To set the division cycle to 1/1 clock:

```
> :OUTPut:CMU:EXTClock 1_1
```

Compatibility

Incompatible with existing models.

**:OUTPut:CMU:EXTClock?**

Response

<source> = <CHARACTER RESPONSE DATA>
- 1_1, 1_64

Function

Queries the division cycle when the external reference signal is used.

Example

```
> :OUTPut:CMU:EXTClock?
< 1_1
```

Compatibility

Incompatible with existing models.
Chapter 7  SCPI Commands

:OUTPut:CMU:FREQuency <numeric>

Parameter  

<numeric> = <DECIMAL NUMERIC PROGRAM DATA>

When the operation frequency setting resolution is MHz:
1225 to 1562  1225 to 1562 MHz, in 1 MHz steps
2450 to 3125  2450 to 3125 MHz, in 1 MHz steps
4900 to 6250  4900 to 6250 MHz, in 1 MHz steps
9800 to 12500  9800 to 12500 MHz, in 1 MHz steps
When the operation frequency setting resolution is kHz:
1225000 to 1562000  1225000 to 1562000 kHz, in 1 kHz steps
2450000 to 3125000  2450000 to 3125000 kHz, in 1 kHz steps
4900000 to 6250000  4900000 to 6250000 kHz, in 1 kHz steps
9800000 to 12500000  9800000 to 12500000 kHz, in 1 kHz steps
The default value is 12500000 kHz.

Function  

Sets the operation frequency of the internal synthesizer.

Example  

To set the operation frequency of the internal synthesizer to 1225 MHz:
> :OUTPut:CMU:FREQuency 1225

Compatibility  

Incompatible with existing models.

:OUTPut:CMU:FREQuency?

Response  

<numeric> = <NR1 NUMERIC RESPONSE DATA>

When the operation frequency setting resolution is MHz:
1225 to 1562  1225 to 1562 MHz
2450 to 3125  2450 to 3125 MHz
4900 to 6250  4900 to 6250 MHz
9800 to 12500  9800 to 12500 MHz
When the operation frequency setting resolution is kHz:
1225000 to 1562000  1225000 to 1562000 kHz
2450000 to 3125000  2450000 to 3125000 kHz
4900000 to 6250000  4900000 to 6250000 kHz
9800000 to 12500000  9800000 to 12500000 kHz

Function  

Queries the operating frequency of the internal synthesizer.

Example  

> :OUTPut:CMU:FREQuency?
< 1225

Compatibility  

Incompatible with existing models.
7.3 PPG Commands

:OUTPut:CMU:RESolution <unit>

Parameter  

<unit> = <CHARACTER PROGRAM DATA>
- KHZ: Sets the kHz unit.
- MHZ: Sets the MHz unit.

Function  
Selects the resolution for setting the operating frequency of the internal synthesizer.

Example  
To set the resolution for setting the operating frequency of the internal synthesizer to kHz:

> :OUTPut:CMU:RESolution KHZ

Compatibility  
Incompatible with existing models.

:OUTPut:CMU:RESolution?

Response  

<unit> = <CHARACTER RESPONSE DATA>
- KHZ, MHZ

Function  
Queries the resolution for setting the operating frequency of the internal synthesizer.

Example  

> :OUTPut:CMU:RESolution?
< KHZ

Compatibility  
Incompatible with existing models.

:OUTPut:CMU:OPERation <numeric>

Parameter  

<numeric> = <DECIMAL NUMERIC PROGRAM DATA>
- 1: Operates with 1/1 of external reference signal (12.5 to 9.8 GHz)
- 2: Operates with 1/2 of external reference signal (6.25 to 4.9 GHz)
- 4: Operates with 1/4 of external reference signal (3.125 to 2.45 GHz)
- 8: Operates with 1/8 of external reference signal (1.5625 to 1.225 GHz)

Function  
Sets the operating frequency rate when the external reference signal is used for the reference clock.

Example  
To set the operating frequency rate of the internal synthesizer to 1/4 (3.125 to 2.45 GHz):

> :OUTPut:CMU:OPERation 4

Compatibility  
Incompatible with existing models.
### :OUTPut:CMU:OPERation?

<table>
<thead>
<tr>
<th>Response</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;numeric&gt;</td>
<td>= &lt;NR1 NUMERIC RESPONSE DATA&gt;</td>
</tr>
<tr>
<td>1</td>
<td>Operates with 1/1 of external reference signal (12.5 to 9.8 GHz)</td>
</tr>
<tr>
<td>2</td>
<td>Operates with 1/2 of external reference signal (6.25 to 4.9 GHz)</td>
</tr>
<tr>
<td>4</td>
<td>Operates with 1/4 of external reference signal (3.125 to 2.45 GHz)</td>
</tr>
<tr>
<td>8</td>
<td>Operates with 1/8 of external reference signal (1.5625 to 1.225 GHz)</td>
</tr>
</tbody>
</table>

### Function
Queries the operating frequency rate setting when the external reference signal is used for the reference clock.

### Example
```
> :OUTPut:CMU:OPERation?
< 4
```

### Compatibility
Incompatible with existing models.
7.3.2 Commands related to Pattern tab

7.3.2.1 Test Pattern setting commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>:SOURce:PATTern:TYPE?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SOURce:PATTern:LOGic?</td>
</tr>
</tbody>
</table>
### :SOURce:PATTern:TYPE \(<\text{type}>\)

**Parameter**  
\(<\text{type}> = \langle\text{CHARACTER PROGRAM DATA}\rangle$

<table>
<thead>
<tr>
<th>(&lt;\text{type}&gt;)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRBS</td>
<td>PRBS pattern</td>
</tr>
<tr>
<td>ZSUbstitution</td>
<td>Zero-substitution pattern</td>
</tr>
<tr>
<td>DATA</td>
<td>Data pattern</td>
</tr>
<tr>
<td>ALTerminate</td>
<td>Alternate pattern</td>
</tr>
<tr>
<td>MIXData</td>
<td>Mixed Data pattern</td>
</tr>
<tr>
<td>MIXalt</td>
<td>Mixed Alternate pattern</td>
</tr>
<tr>
<td>SEQuence</td>
<td>Sequence pattern</td>
</tr>
</tbody>
</table>

**Function**  
Sets the type of the test pattern.

**Example**  
To set the test pattern type to the Mixed Data pattern:

\>` :SOURce:PATTern:TYPE MIXData

**Compatibility**  
Partially compatible with the MP1632C (ZSUbstitution only).

### :SOURce:PATTern:TYPE?

**Response**  
\(<\text{type}> = \langle\text{CHARACTER RESPONSE DATA}\rangle$

<table>
<thead>
<tr>
<th>(&lt;\text{type}&gt;)</th>
<th>(&lt;\text{CHARACTER RESPONSE DATA}&gt;)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRBS, ZSUB, DATA, ALT, MIXD, MIX, SEQ</td>
<td></td>
</tr>
</tbody>
</table>

**Function**  
Queries the type of the test pattern.

**Example**  
\>` :SOURce:PATTern:TYPE?

\< MIXD

**Compatibility**  
Partially compatible with the MP1632C (ZSUbstitution only).

### :SOURce:PATTern:LOGic \(<\text{logic}>\)

**Parameter**  
\(<\text{logic}> = \langle\text{CHARACTER PROGRAM DATA}\rangle$

<table>
<thead>
<tr>
<th>(&lt;\text{logic}&gt;)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>POSitive</td>
<td>Positive logic</td>
</tr>
<tr>
<td>NEGative</td>
<td>Negative logic</td>
</tr>
</tbody>
</table>

**Function**  
Sets the logic (positive or negative) of the test pattern.

**Example**  
To set the logic of the test pattern to the negative logic (NEGative):

\>` :SOURce:PATTern:LOGic NEGative

**Compatibility**  
Incompatible with existing models.

### :SOURce:PATTern:LOGic?

**Response**  
\(<\text{logic}> = \langle\text{CHARACTER RESPONSE DATA}\rangle$

<table>
<thead>
<tr>
<th>(&lt;\text{logic}&gt;)</th>
<th>(&lt;\text{CHARACTER RESPONSE DATA}&gt;)</th>
</tr>
</thead>
<tbody>
<tr>
<td>POS, NEG</td>
<td></td>
</tr>
</tbody>
</table>

**Function**  
Queries the logic of the test pattern.

**Example**  
\>` :SOURce:PATTern:LOGic?

\< NEG

**Compatibility**  
Incompatible with existing models.
7.3 PPG Commands

7.3.2 PRBS pattern setting commands

![Figure 7.3.2.2-1 PRBS pattern setting](image)

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>:SOURce:PATTern:PRBS:LENGth?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SOURce:PATTern:PRBS:MRATio?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SOURce:PATTern:PRBS:BShift?</td>
</tr>
</tbody>
</table>
Chapter 7  SCPI Commands

:SOURce:PATTern:PRBS:LENGth <numeric>

Parameter

<table>
<thead>
<tr>
<th>&lt;numeric&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
</tr>
<tr>
<td>9</td>
</tr>
<tr>
<td>10</td>
</tr>
<tr>
<td>11</td>
</tr>
<tr>
<td>15</td>
</tr>
<tr>
<td>20</td>
</tr>
<tr>
<td>23</td>
</tr>
<tr>
<td>31</td>
</tr>
</tbody>
</table>

Function
Sets the number of stages \(2^n - 1\) during PRBS pattern generation.

Example
To set PRBS15:

```
> :SOURce:PATTern:PRBS:LENGth 15
```

Compatibility
Incompatible with existing models.

:SOURce:PATTern:PRBS:LENGth?

Response

<table>
<thead>
<tr>
<th>&lt;numeric&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>7, 9, 10, 11, 15, 20, 23, 31</td>
</tr>
</tbody>
</table>

Function
Queries the number of stages during PRBS pattern generation.

Example

```
> :SOURce:PATTern:PRBS:LENGth?
< 15
```

Compatibility
Incompatible with existing models.

:SOURce:PATTern:PRBS:MRATio <mratio>

Parameter

<table>
<thead>
<tr>
<th>&lt;mratio&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>M1_2</td>
</tr>
<tr>
<td>M1_4</td>
</tr>
<tr>
<td>M1_8</td>
</tr>
<tr>
<td>M0_8</td>
</tr>
<tr>
<td>I1_2</td>
</tr>
<tr>
<td>M3_4</td>
</tr>
<tr>
<td>M7_8</td>
</tr>
<tr>
<td>M8_8</td>
</tr>
</tbody>
</table>

Function
Sets the mark ratio during PRBS pattern generation.

Example
To set the mark ratio to 1/8:

```
> :SOURce:PATTern:PRBS:MRATio M1_8
```

Compatibility
Compatible with the MP1632C.
### :SOURce:PATTern:PRBS:MRATio?

**Response**

<mratio> = <CHARACTER RESPONSE DATA>
M1_2, M1_4, M1_8, M0_8, I1_2, M3_4, M7_8, M8_8

**Function**

Queries the mark ratio during PRBS pattern generation.

**Example**

> :SOURce:PATTern:PRBS:MRATio?
< M1_8

**Compatibility**

Compatible with the MP1632C.

### :SOURce:PATTern:PRBS:BSHift <numeric>

**Parameter**

<numeric> = <DECIMAL NUMERIC PROGRAM DATA>
1 1 bit
3 3 bits

**Function**

Sets the bit shift during PRBS pattern generation.

**Example**

To set the bit shift to 1 bit:
> :SOURce:PATTern:PRBS:BSHift 1

**Compatibility**

Compatible with the MP1632C.

### :SOURce:PATTern:PRBS:BSHift?

**Response**

<numeric> = <NR1 NUMERIC RESPONSE DATA>
1, 3

**Function**

Queries the bit shift number during PRBS pattern generation.

**Example**

> :SOURce:PATTern:PRBS:BSHift?
< 1

**Compatibility**

Compatible with the MP1632C.
7.3.2.3 Zero-Substitution pattern setting commands

Table 7.3.2.3-1 Zero-Substitution pattern setting commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>:SOURce:PATTern:ZSUBstitute:LENGth?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SOURce:PATTern:ZSUBstitute:ZLENght?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SOURce:PATTern:ZSUBstitute:ADDBit?</td>
</tr>
</tbody>
</table>

:SOURce:PATTern:ZSUBstitute:LENGth <length>

Parameter

\[
\begin{align*}
7 & \quad 2^n \ (n = 7) \\
9 & \quad 2^n \ (n = 9) \\
10 & \quad 2^n \ (n = 10) \\
11 & \quad 2^n \ (n = 11) \\
15 & \quad 2^n \ (n = 15) \\
20 & \quad 2^n \ (n = 20) \\
23 & \quad 2^n \ (n = 23) \\
-7 & \quad 2^{n-1} \ (n = 7) \\
-9 & \quad 2^{n-1} \ (n = 9) \\
-10 & \quad 2^{n-1} \ (n = 10) \\
-11 & \quad 2^{n-1} \ (n = 11) \\
-15 & \quad 2^{n-1} \ (n = 15) \\
-20 & \quad 2^{n-1} \ (n = 20) \\
-23 & \quad 2^{n-1} \ (n = 23)
\end{align*}
\]

Function

Sets the pattern length during Zero-Substitution pattern generation.

Example

To set the pattern length to 2^{15}:

\[ > \text{:SOURce:PATTern:ZSUBstitute:LENGth 15} \]

Compatibility

Compatible with the MP1632C.
7.3 PPG Commands

:SOURce:PATTern:ZSUBstitute:LENGth?

Response  

\[
\text{<length>} = \text{<NR1 NUMERIC RESPONSE DATA>}
\]

7, 9, 10, 11, 15, 20, 23, –7, –9, –10, –11, –15, –20, –23

Function  
Queries the pattern length during Zero-Substitution pattern generation.

Example  

> :SOURce:PATTern:ZSUBstitute:LENGth?
< 15

Compatibility  
Compatible with the MP1632C.

:SOURce:PATTern:ZSUBstitute:ZLENgth <numeric>

Parameter  

\[
\text{<numeric>} = \text{<DECIMAL NUMERIC PROGRAM DATA>}
\]

When ZSUBLength, \(2^n\) is set:

1 to \(2^n−1\) 1 to \(2^n−1\), in 1-bit steps

When ZSUBLength, \(2^n−1\) is set:

1 to \(2^n−2\) 1 to \(2^n−2\), in 1-bit steps

n = 7, 9, 11, 15, 20, 23

Function  
Sets the number of bits to which "0" is inserted (zero-insertion bit count) during Zero-Substitution pattern generation.

Available parameters vary depending on the setting conditions.

Example  

To set the zero-insertion bit count to 10 bits:

> :SOURce:PATTern:ZSUBstitute:ZLENgth 10

Compatibility  
Compatible with the MP1632C.

:SOURce:PATTern:ZSUBstitute:ZLENgth?

Response  

\[
\text{<numeric>} = \text{<NR1 NUMERIC RESPONSE DATA>}
\]

1 to \(2^n−1\), in 1-bit steps

n = 7, 9, 11, 15, 20, 23

Function  
Queries the zero-insertion bit count during Zero-Substitution pattern generation.

Example  

> :SOURce:PATTern:ZSUBstitute:ZLENgth?
< 10

Compatibility  
Compatible with the MP1632C.
### :SOURce:PATTERn:ZSUBstitute:ADDBit <numeric>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;numeric&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
<td>Adds one bit of &quot;0&quot; next to the last of zero-inserted bits.</td>
</tr>
<tr>
<td>0</td>
<td>Adds one bit of &quot;0&quot; next to the last of zero-inserted bits.</td>
</tr>
<tr>
<td>1</td>
<td>Adds one bit of &quot;1&quot; next to the last of zero-inserted bits (same as existing models).</td>
</tr>
</tbody>
</table>

**Function**
Sets whether to add one bit of "0" or "1" next to the last of zero-inserted bits.

**Example**
To set one bit of "1" next to the last of zero-inserted bits:
> :SOURce:PATTERn:ZSUBstitute:ADDBit 1

**Compatibility**
Incompatible with existing models.

### :SOURce:PATTERn:ZSUBstitute:ADDBit?

<table>
<thead>
<tr>
<th>Response</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;numeric&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt;</td>
<td>Adds one bit of &quot;0&quot; next to the last of zero-inserted bits.</td>
</tr>
<tr>
<td>0</td>
<td>Adds one bit of &quot;0&quot; next to the last of zero-inserted bits.</td>
</tr>
<tr>
<td>1</td>
<td>Adds one bit of &quot;1&quot; next to the last of zero-inserted bits (same as existing models).</td>
</tr>
</tbody>
</table>

**Function**
Queries whether to add one bit of "0" or "1" next to the last of zero-inserted bits.

**Example**
> :SOURce:PATTERn:ZSUBstitute:ADDBit?
< 1

**Compatibility**
Incompatible with existing models.
7.3.2.4 Data pattern setting commands

Table 7.3.2.4-1 Data pattern setting commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1]</td>
<td>Data Length</td>
<td>::SOURce::PATTern::DATA::LENGth</td>
</tr>
<tr>
<td></td>
<td></td>
<td>::SOURce::PATTern::DATA::LENGth?</td>
</tr>
<tr>
<td>[2]</td>
<td>No label (Pattern setting)</td>
<td>::SOURce::PATTern::DATA::WHOLe</td>
</tr>
<tr>
<td></td>
<td></td>
<td>::SOURce::PATTern::DATA::WHOLe?</td>
</tr>
<tr>
<td>[3]</td>
<td>No label (Pattern setting)</td>
<td>::SOURce::PATTern::BDATA::WHOLe</td>
</tr>
<tr>
<td></td>
<td></td>
<td>::SOURce::PATTern::BDATA::WHOLe?</td>
</tr>
<tr>
<td>[4]</td>
<td>No label (Data inversion)</td>
<td>::SOURce::PATTern::DREVerse::ADDReSS</td>
</tr>
<tr>
<td>[5]</td>
<td>No label (Data inversion)</td>
<td>::SOURce::PATTern::DREVerse::DELTa</td>
</tr>
<tr>
<td>[6]</td>
<td>No label (All 0 or All 1)</td>
<td>::SOURce::PATTern::DATA::FILL</td>
</tr>
</tbody>
</table>
Chapter 7  SCPI Commands

:SOURce:PATTern:DATA:LENGth <numeric>

Parameter

<numeric> = <DECIMAL NUMERIC PROGRAM DATA>
2 to 134217728  2 to 134,217,728 bits, in 1-bit steps

Note:
At 2 ch Combination, the setting range and Step are doubled.
At 4 ch Combination, the setting range and Step are quadrupled.

Function
Sets the pattern length during Data pattern generation.

Example
To set the pattern length to 100 bits:
> :SOURce:PATTern:DATA:LENGth 100

Compatibility
Incompatible with existing models.

:SOURce:PATTern:DATA:LENGth?

Response
<numeric> = <NR1 NUMERIC RESPONSE DATA>

Function
Queries the pattern length during Data pattern generation.

Example
> :SOURce:PATTern:DATA:LENGth?
< 100

Compatibility
Incompatible with existing models.
:SOURCE:PATTern:DATA:WHOLe <start>,<end>,<data>

Parameter

- **<start>** = <NON-DECIMAL PROGRAM DATA>
- **<end>** = <NON-DECIMAL PROGRAM DATA>
- **<data>** = <STRING PROGRAM DATA>
  - "H***" Specify pattern data in hexadecimal format.
    - 1 to 400 characters (pattern data of 400 \( \times \) 4 bits)
    - Specify a character string consisting of 0 to 9 and A to F.
    - If the last character string of the pattern data is short of a hexadecimal unit (4 bits), add 0(s) up to the last bit to make a hexadecimal string.
  - "B***" Specify pattern data in binary format.
    - 1 to 400 characters (pattern data of 400 bits)
    - Specify a character string consisting of 0 and 1.

Function
Sets the pattern data from the <start> to <end> addresses during Data pattern generation.
The set data overwrites the specified range.

Example
To set the addresses 0 to 1F of the pattern data to 5A:
> :SOURCE:PATTern:DATA:WHOLe #H0,#H1F, "H5A"

Compatibility
Compatible with the MP1632C.

Note:
When the number of data bits specified in <data> is smaller than the range specified by <start> and <end>, set <data> repeatedly up to the specified range.
<Example>
- <start> = #H0, <end> = #H1F, <data> = "HABC"
  Setting data: ABCABCAB
- <start> = #H0, <end> = #H7, <data> = "B011"
  Setting data: 01101101

When the number of data bits specified in <data> is greater than the range specified by <start> and <end>, cut off the part of <data> out of the specified range.
### :SOURce:PATTern:DATA:WHOLe? \(<start>\)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>(&lt;start&gt;) = (&lt;\text{NON-DECIMAL PROGRAM DATA}&gt;)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>#H0 to #H7FFFFFF 0 to 7FFFFFFF bits, in 1-bit steps</td>
</tr>
<tr>
<td></td>
<td>(Specify in hexadecimal)</td>
</tr>
</tbody>
</table>

**Note:**
- The maximum setting is the pattern length.
- At 2 ch Combination, the setting range is doubled.
- At 4 ch Combination, the setting range is quadrupled.

<table>
<thead>
<tr>
<th>Response</th>
<th>(&lt;data&gt;) = (&lt;\text{STRING RESPONSE DATA}&gt;)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>&quot;H***&quot; Returns the pattern data in hexadecimal format.</td>
</tr>
<tr>
<td></td>
<td>Within 400 characters (pattern data of 400 × 4 bits)</td>
</tr>
</tbody>
</table>

| Function | Queries the pattern data of 400 × 4 bits from the \(<start>\) address during Data pattern generation. |

<table>
<thead>
<tr>
<th>Example</th>
<th>To query the pattern data from address 0:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(&gt; :\text{SOURce:PATTern:DATA:WHOLe? #H0} )</td>
</tr>
<tr>
<td></td>
<td>(&lt; &quot;H5A&quot; )</td>
</tr>
</tbody>
</table>

| Compatibility | Compatible with the MP1632C. |

---

Example:
- \(<\text{start}> = \#H0, <\text{end}> = \#HF, <\text{data}> = \#HABCDEF\)
  - Setting data: ABCD
- \(<\text{start}> = \#H0, <\text{end}> = \#H3, <\text{data}> = \#B01100110\)
  - Setting data: 0110
:SOURCE:PATTern:BDATa:WHOLe <start>,<end>,<bdata>

Parameter

- `<start>` = `<NON-DECIMAL PROGRAM DATA>`
- `<end>` = `<NON-DECIMAL PROGRAM DATA>`

<start> = #H0 to #H7FFFFFF 0 to 7FFFFFF bits, in 1-bit steps
(Specify in hexadecimal)

<end> = #H0 to #H7FFFFFF 0 to 7FFFFFF bits, in 1-bit steps
(Specify in hexadecimal)

Note:
The maximum setting is the pattern length.
At 2 ch Combination, the setting range is doubled.
At 4 ch Combination, the setting range is quadrupled.

<bdata> = `<ARBITRARY BLOCK PROGRAM DATA>`

#XYYY<binary> X: Number of YYY digits
YYY: Number of bytes of <binary>
1 to 16,777,216 bytes

<binary>: Binary data up to 16,777,216 bytes

Function
Sets binary data of the pattern data from <start> to <end> addresses
during Data pattern generation.
The set data overwrites the specified range.

Example
To set the addresses 0 to 1F of the pattern data to 41:
> :SOURCE:PATTern:BDATa:WHOLe #H0,#H1F,#11A (A=41)

Compatibility
Compatible with the MP1632C.
Chapter 7  SCPI Commands

:SOURce:PATTern:BDATa:WHOLe? <start>[,<size>]

Parameter

- `<start>` = <NON-DECIMAL PROGRAM DATA>
  - #H0 to #H7FFFFFFF 0 to 7FFFFFFF bits, in 1-bit steps
    (Specify in hexadecimal)

Note:

- The maximum setting is the pattern length.
- At 2 ch Combination, the setting range is doubled.
- At 4 ch Combination, the setting range is quadrupled.

- `<size>` = <NR1 NUMERIC PROGRAM DATA>
  - 1 to 16777216 1 to 1,677,7216 bytes, in 1-byte steps

Response

- `<bdata>` = <DEFINITE LENGTH ARBITRARY BLOCK RESPONSE DATA>
  - #XYYY<binary> X: Number of YYY digits
    YYY: Number of bytes of <binary>
  - 1 to 16,777,216 bytes
  - <binary>: Binary data of up to 16,777,216 bytes

Function

Queries binary data of bytes specified by `<size>` from `<start>` address of the pattern data during Data pattern generation.

Example

To query binary data of the setting pattern from the `<start>` address of the pattern data during Data pattern generation:

```plaintext
> :SOURce:PATTern:BDATa:WHOLe? #H0
< #516000AA
```

To query binary data of 2 bytes from the `<start>` address of the pattern data during Data pattern generation:

```plaintext
> :SOURce:PATTern:BDATa:WHOLe? #H0,2
< #12AA
```

Compatibility

Compatible with the MP1632C.
### 7.3 PPG Commands

#### :SOURce:PATTern:DREVerse:ADDRess <start>,<end>

**Parameter**

- `<start>` = <NON-DECIMAL PROGRAM DATA>
  - #H0 to #H7FFFFFFF 0 to 7FFFFFFF bits, in 1-bit steps
    - (Specify in hexadecimal)
- `<end>` = <NON-DECIMAL PROGRAM DATA>
  - #H0 to #H7FFFFFFF 0 to 7FFFFFFF bits, in 1-bit steps
    - (Specify in hexadecimal)

**Note:**
- The maximum setting is the pattern length.
- At 2 ch Combination, the setting range is doubled.
- At 4 ch Combination, the setting range is quadrupled.

**Function**

Inverts the data in the program during Data pattern generation. Specify the inversion range by the `<start>` and `<end>` addresses.

**Example**

To invert addresses 0 to 4F of the pattern data:

> :SOURce:PATTern:DREVerse:ADDRess #H0,#H4F

**Compatibility**

Compatible with the MP1632C.

#### :SOURce:PATTern:DREVerse:DELTa <start>,<delta>

**Parameter**

- `<start>` = <NON-DECIMAL PROGRAM DATA>
  - #H0 to #H7FFFFFFF 0 to 7FFFFFFF bits, in 1-bit steps
    - (Specify in hexadecimal)
- `<delta>` = <DECIMAL NUMERIC PROGRAM DATA>
  - 1 to 134217728 1 to 134,217,728 bits, in 1-bit steps

**Note:**
- The maximum setting is the pattern length.
- At 2 ch Combination, the setting range is doubled.
- At 4 ch Combination, the setting range is quadrupled.

**Function**

Inverts the data in the program during Data pattern generation. Specify the inversion range by the number of bits from the `<start>` address (relative specification).

**Example**

To invert 128 bits of the pattern data from address 0:

> :SOURce:PATTern:DREVerse:DELTa #H0,128

**Compatibility**

Compatible with the MP1632C.
### Chapter 7  SCPI Commands

**:SOURce:PA TTern:DATA:FILL <range>,<page>,<data>**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;range&gt; = &lt;CHARACTER PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>PAGE</td>
<td>Specifies a page.</td>
</tr>
<tr>
<td></td>
<td>(One page is defined as 128 bits.)</td>
</tr>
<tr>
<td>ALL</td>
<td>Specifies all data.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>&lt;page&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 to 1048576</td>
</tr>
</tbody>
</table>

Max = \[
\frac{\text{Pattern Length}}{128}
\]

(roundering up fractions)

Specify "0" when <range> is set to ALL.

<table>
<thead>
<tr>
<th>&lt;data&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
</tr>
<tr>
<td>1</td>
</tr>
</tbody>
</table>

**Function**

Sets 0s or 1s for the specified page or all data in the program during Data pattern generation.

**Example**

To set 1s for the second page (from 129th to 256th bits) of the pattern data:

> :SOURCE:pattern:DATA:FILL PAGE, 2, 1

**Compatibility**

Incompatible with existing models.
7.3.2.5 Alternate pattern setting commands

![Figure 7.3.2.5-1 Alternate pattern setting](image1)

![Figure 7.3.2.5-2 Alternate pattern setting (Pattern Editor screen)](image2)
Table 7.3.2.5-1  Alternate pattern setting commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Loop Control</td>
<td>:SOURce:PATTern:ALTername:LOOPcont</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SOURce:PATTern:ALTername:LOOPcont?</td>
</tr>
<tr>
<td>3</td>
<td>A/B LoopTime</td>
<td>:SOURce:PATTern:ALTername:LTIMe</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SOURce:PATTern:ALTername:LTIMe?</td>
</tr>
<tr>
<td>4</td>
<td>Data Length</td>
<td>:SOURce:PATTern:ALTername:LENGth</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SOURce:PATTern:ALTername:LENGth?</td>
</tr>
<tr>
<td>5</td>
<td>No label (Pattern setting)</td>
<td>:SOURce:PATTern:ALTername:DATA:WHOLe</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SOURce:PATTern:ALTername:DATA:WHOLe?</td>
</tr>
<tr>
<td>6</td>
<td>No label (Pattern setting)</td>
<td>:SOURce:PATTern:ALTername:BDATa:WHOLe</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SOURce:PATTern:ALTername:BDATa:WHOLe?</td>
</tr>
<tr>
<td>7</td>
<td>No label (Data inversion)</td>
<td>:SOURce:PATTern:ALTername:DREVerse:ADDRess</td>
</tr>
<tr>
<td>8</td>
<td>No label (Data inversion)</td>
<td>:SOURce:PATTern:ALTername:DREVerse:DELTa</td>
</tr>
<tr>
<td>9</td>
<td>No label (All 0 or All 1)</td>
<td>:SOURce:PATTern:ALTername:DATA:FILL</td>
</tr>
</tbody>
</table>

:SOURce:PATTern:ALTername:LOOPcont <loop>

Parameter  
<loop> = <CHARACTER PROGRAM DATA>
INTernal  Switching control using internal signal
EXTernal  Switching control using external signal
MANual    Manual switching control

Function  
Sets the Alternate pattern A/B switching control method.

Example  
To set the switching control using internal signal for the Alternate pattern A/B switching:
> :SOURce:PATTern:ALTername:LOOPcont INTernal

Compatibility  
Incompatible with existing models.

:SOURce:PATTern:ALTername:LOOPcont?

Response  
<loop> = <CHARACTER RESPONSE DATA>
INT, EXT, MAN

Function  
Queries the Alternate pattern A/B switching control method.

Example  
> :SOURce:PATTern:ALTername:LOOPcont?
< INT

Compatibility  
Incompatible with existing models.

:SOURce:PATTern:ALTername:MANual

Function  
Switches the Alternate pattern A/B manually.

Example  
> :SOURce:PATTern:ALTername:MANual

Compatibility  
Incompatible with existing models.
7.3  PPG Commands

:SOURce:PATTern:ALTernate:LTIMe <content>,<numeric>

Parameter

<content> = <CHARACTER PROGRAM DATA>
APAT  Pattern A
BPAT  Pattern B

<numeric> = <DECIMAL NUMERIC PROGRAM DATA>
1 to 511  1 to 511 times, in 1-time steps

Function
Sets the number of generation times of the Alternate pattern A or B.

Example
To set the number of pattern B generation times to 20:
> :SOURce:PATTern:ALTernate:LTIMe BPAT,20

Compatibility
Incompatible with existing models.

:SOURce:PATTern:ALTernate:LTIMe? <content>

Parameter

<content> = <CHARACTER PROGRAM DATA>
APAT, BPAT

Response
<numeric> = <NR1 NUMERIC RESPONSE DATA>
1 to 511  1 to 511 times, in 1-time steps

Function
Queries the number of generation times of the Alternate pattern A or B.

Example
> :SOURce:PATTern:ALTernate:LTIMe? BPAT
< 20

Compatibility
Incompatible with existing models.

:SOURce:PATTern:ALTernate:LENGth <content>,<numeric>

Parameter

<content> = <CHARACTER PROGRAM DATA>
APAT  Pattern A
BPAT  Pattern B

<numeric> = <DECIMAL NUMERIC PROGRAM DATA>
128 to 67108864  128 to 67,108,864 bits, in 128-bit steps

Function
Sets the pattern length of the Alternate pattern A or B.

Example
To set the pattern length of the pattern B to 128 bits:
> :SOURce:PATTern:ALTernate:LENGth BPAT,128

Compatibility
Incompatible with existing models.
**Chapter 7  SCPI Commands**

**:SOURce:PATTern:ALTernate:LENGth? <content>**

**Parameter**

<content> = <CHARACTER PROGRAM DATA>
APAT, BPAT

**Response**

<numeric> = <NR1 NUMERIC RESPONSE DATA>
128 to 67108864
128 to 67,108,864 bits, in 128-bit steps

**Function**
Queries the pattern length of the Alternate pattern A or B.

**Example**
To query the pattern length of the Alternate pattern B:

> :SOURce:PATTern:ALTernate:LENGth? BPAT
< 128

**Compatibility**
Incompatible with existing models.

---

**:SOURce:PATTern:ALTernate:DATA:WHOLe <content>,<start>,<end>,<data>**

**Parameter**

<content> = <CHARACTER PROGRAM DATA>
APAT Pattern A
BPAT Pattern B

<start> = <NON-DECIMAL PROGRAM DATA>
#H0 to #H3FFFFFF 0 to 3FFFFFF bits, in 1-bit steps
(Specify in hexadecimal)

<end> = <NON-DECIMAL PROGRAM DATA>
#H0 to #H3FFFFFF 0 to 3FFFFFF bits, in 1-bit steps
(Specify in hexadecimal)

**Note:**
The maximum setting value is the pattern length.

<data> = <STRING PROGRAM DATA>
"H***" Specify pattern data in hexadecimal format.
1 to 400 characters (pattern data of 400 × 4 bits)
Specify a character string consisting of 0 to 9
and A to F.
If the last character string of the pattern data is
short of a hexadecimal unit (4 bits), add 0(s) up
to the last bit to make a hexadecimal string.

"B***" Specify pattern data in binary format.
1 to 400 characters (pattern data of 400 bits)
Specify a character string consisting of 0 and 1.

**Function**
Sets the pattern data from the <start> to <end> addresses during
Alternate pattern generation.
The set data overwrites the specified range.

**Example**
To set the addresses 0 to 1F of the pattern A to AA:

> :SOURce:PATTern:ALTernate:DATA:WHOLe APAT,#H0,#H1F,
"HAA"

**Compatibility**
Incompatible with existing models.
7.3 PPG Commands

:SOURce:PATTer:n:ALTernate:DATA:WHOLe? <content>,<start>

Parameter

<content> = <CHARACTER PROGRAM DATA>
APAT = Pattern A
BPAT = Pattern B

<start> = <NON-DECIMAL PROGRAM DATA>
#H0 to #H3FFFFFF 0 to 3FFFFFFF bits, in 1-bit steps
(Specify in hexadecimal)

Note:
The maximum setting value is the pattern length.

Response

<data> = <STRING RESPONSE DATA>
"H***" Returns the pattern data in hexadecimal format.
Within 400 characters (pattern data of 400 × 4 bits)

Function
Queries the pattern data of 400 × 4 bits from the <start> address during Alternate pattern generation.

Example
To query the pattern A data from address 0:
> :SOURce: PATTer:n: ALTernate: DATA: WHOLe? APAT, H0
< "HAA"

Compatibility
Incompatible with existing models.
Chapter 7  SCPI Commands

:SOURce:PATTern:ALTernate:BDATa:WHOLe
<content>,<start>,<end>,<bdata>

Parameter
- **<content>** = <CHARACTER PROGRAM DATA>
- **APAT** = Pattern A
- **BPAT** = Pattern B
- **<start>** = <NON-DECIMAL PROGRAM DATA>
  - #H0 to #H3FFFFFF 0 to 3FFFFFF bits, in 1-bit steps
  (Specify in hexadecimal)
- **<end>** = <NON-DECIMAL PROGRAM DATA>
  - #H0 to #H3FFFFFF 0 to 3FFFFFF bits, in 1-bit steps
  (Specify in hexadecimal)

**Note:**
The maximum setting value is the pattern length.
- **<bdata>** = <ARBITRARY BLOCK PROGRAM DATA>
  - #XYYY<binary>
    - **X:** Number of YYY digits
    - **YYY:** Number of bytes of <binary>
    - 1 to 8,388,608 bytes
    - <binary>:Binary data up to 8,388,608 bytes

Function
Sets the binary data from <start> to <end> addresses of the pattern data during Alternate pattern generation.
The set data overwrites the specified range.

Example
To set the addresses 0 to 1F of the pattern A data to 42:
> :SOURce:PATTern:ALTernate:BDATa:WHOLe APAT,#H0,#H1F,#11B
(B=42)

Compatibility
Incompatible with existing models.
### :SOURce:PATTern:ALTernate:BDATa:WHOLe?

<content>,<start>[,<size>]

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;content&gt;</td>
<td>= &lt;CHARACTER PROGRAM DATA&gt;</td>
</tr>
<tr>
<td>APAT</td>
<td>Pattern A</td>
</tr>
<tr>
<td>BPAT</td>
<td>Pattern B</td>
</tr>
<tr>
<td>&lt;start&gt;</td>
<td>= &lt;NON-DECIMAL PROGRAM DATA&gt;</td>
</tr>
<tr>
<td>#H0 to #H3FFFFFF</td>
<td>0 to 3FFFFFF bits, in 1-bit steps</td>
</tr>
<tr>
<td></td>
<td>(Specify in hexadecimal)</td>
</tr>
</tbody>
</table>

**Note:**
The maximum setting value is the pattern length.

| <size> | = <NR1 NUMERIC PROGRAM DATA> |
| 1 to 8388608 | 1 to 8,388,608 bytes, in 1-byte steps |

**Response**

<bdata> = <DEFINITE LENGTH ARBITRARY BLOCK RESPONSE DATA>

#XYYY<binary>

- X: Number of YYY digits
- YYY: Number of bytes of <binary>

1 to 8,388,608 bytes

<binary>: Binary data up to 8,388,608 bytes

Binary data of the setting pattern is queried when [<size>] is omitted.

**Function**

Queries binary data of bytes specified by <size> from the <start> address of the pattern data during Alternate pattern generation.

**Example**

To query binary data of 2 bytes from address 0 of the pattern A data:

> :SOURce:PATTern:ALTernate:BDATa:WHOLe? APAT,#H0,2

< #12AA

To query binary data of the setting pattern from address 0 of the pattern A data:

> :SOURce:PATTern:ALTernate:BDATa:WHOLe? APAT,#H0

< #516000AA

**Compatibility**

Incompatible with existing models.
Chapter 7  SCPI Commands

:SOURce:PATTern:ALTerinate:DREVerse:ADDRess
<content>,<start>,<end>

Parameter
<content> = <CHARACTER PROGRAM DATA>
APAT           Pattern A
BPAT           Pattern B
<start> = <NON-DECIMAL PROGRAM DATA>
#H0 to #H3FFFFFF 0 to 3FFFFFF bits, in 1-bit steps
(Specify in hexadecimal)
<end> = <NON-DECIMAL PROGRAM DATA>
#H0 to #H3FFFFFF 0 to 3FFFFFF bits, in 1-bit steps
(Specify in hexadecimal)

Note:
The maximum setting value is the pattern length.

Function
Inverts the data in the program during Alternate pattern generation.
Specify the inversion range by the <start> and <end> addresses.

Example
To invert addresses 0 to 1F of the pattern A:
> :SOURce:PATTern:ALTerinate:DREVerse:ADDRess APAT,#H0,#H1F

Compatibility
Incompatible with existing models.

:SOURce:PATTern:ALTerinate:DREVerse:DELTa
<content>,<start>,<delta>

Parameter
<content> = <CHARACTER PROGRAM DATA>
APAT           Pattern A
BPAT           Pattern B
<start> = <NON-DECIMAL PROGRAM DATA>
#H0 to #H3FFFFFF 0 to 3FFFFFF bits, in 1-bit steps
(Specify in hexadecimal)
<delta> = <DECIMAL NUMERIC PROGRAM DATA>
1 to 67108864 1 to 67,108,864 bits, in 1-bit steps

Note:
The maximum setting value is the pattern length.

Function
Inverts the data in the program during Alternate pattern generation.
Specify the inversion range by the number of bits from the <start> address (relative specification).

Example
To invert 64 bits of the pattern A from address 0:
> :SOURce:PATTern:ALTerinate:DREVerse:DELTa APAT,#H0,64

Compatibility
Incompatible with existing models.
### :SOURce:PATTern:ALTernate:DATA:FILL

<content>,<range>,<page>,<data>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;content&gt;</td>
<td>= CHARACTER PROGRAM DATA</td>
</tr>
<tr>
<td>APAT</td>
<td>Pattern A</td>
</tr>
<tr>
<td>BPAT</td>
<td>Pattern B</td>
</tr>
<tr>
<td>&lt;range&gt;</td>
<td>= CHARACTER PROGRAM DATA</td>
</tr>
<tr>
<td>PAGE</td>
<td>Specifies a page.</td>
</tr>
<tr>
<td>ALL</td>
<td>Specifies all data.</td>
</tr>
<tr>
<td>&lt;page&gt;</td>
<td>= DECIMAL NUMERIC PROGRAM DATA</td>
</tr>
<tr>
<td>1 to 524288</td>
<td>1 to 524,288 (Max.) pages, in 1-page steps</td>
</tr>
<tr>
<td>Max</td>
<td>Pattern Length / 128 (rounding up fractions)</td>
</tr>
<tr>
<td></td>
<td>Specify &quot;0&quot; when &lt;range&gt; is set to ALL.</td>
</tr>
<tr>
<td>&lt;data&gt;</td>
<td>= NUMERIC PROGRAM DATA</td>
</tr>
<tr>
<td>0</td>
<td>Fills the specified range with 0s.</td>
</tr>
<tr>
<td>1</td>
<td>Fills the specified range with 1s.</td>
</tr>
</tbody>
</table>

**Function**
Sets 0s or 1s for the specified page or all data in the program during Alternate pattern generation.

**Example**
To set 0s for the second page of the pattern A:

> :SOURce:PATTern:ALTernate:DATA:FILL APAT,PAGE,2,0

**Compatibility**
Incompatible with existing models.
7.3.2.6 Mixed Data pattern setting commands

Figure 7.3.2.6-1  Mixed Data pattern setting

Figure 7.3.2.6-2  Scramble Setup dialog box
### Table 7.3.2.6-1 Mixed Data pattern setting (Pattern Editor screen)

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>:SOURce:PATTern:MIXData:PRBS:BITShift?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SOURce:PATTern:MIXData:PRBS:LENGth?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SOURce:PATTern:MIXData:PRBS:MRATio?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SOURce:PATTern:MIXData:SCRamble?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SOURce:PATTern:MIXData:PRBS:SEQUence?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SOURce:PATTern:MIXData:SRSetting?</td>
</tr>
<tr>
<td>[8]</td>
<td>Number of Block</td>
<td>:SOURce:PATTern:MIXData:BLOCK</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SOURce:PATTern:MIXData:BLOCK?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SOURce:PATTern:MIXData:RLENgth?</td>
</tr>
<tr>
<td>[10]</td>
<td>Number of Row</td>
<td>:SOURce:PATTern:MIXData:ROW</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SOURce:PATTern:MIXData:ROW?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SOURce:PATTern:MIXData:DATA:LENGth?</td>
</tr>
</tbody>
</table>
Table 7.3.2.6-1  Mixed Data pattern setting (Pattern Editor screen) (Cont’d)

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>:SOURce:PATTern:MIXData:DATA:WHOLe?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SOURce:PATTern:MIXData:BDAta:WHOLe?</td>
</tr>
<tr>
<td>[16]</td>
<td>No label (All 0 or All 1)</td>
<td>:SOURce:PATTern:MIXData:DATA:FILL</td>
</tr>
</tbody>
</table>

:SOURce:PATTern:MIXData:PRBS:BITShift <numeric>

Parameter  
<numeric> = <DECIMAL NUMERIC PROGRAM DATA>
1 1 bit
3 3 bits

Function  
Sets the bit shift for the PRBS pattern during Mixed Data pattern generation.

Example  
To set the bit shift to 3-bit shift:
> :SOURce:PATTern:MIXData:PRBS:BITShift 3

Compatibility  
Incompatible with existing models.

:SOURce:PATTern:MIXData:PRBS:BITShift?

Response  
<numeric> = <NR1 NUMERIC RESPONSE DATA>
1 1 bit
3 3 bits

Function  
Queries the bit shift for the PRBS pattern during Mixed Data pattern generation.

Example  
> :SOURce:PATTern:MIXData:PRBS:BITShift?
< 3

Compatibility  
Incompatible with existing models.
7.3 PPG Commands

:SOURce:PATTern:MIXData:PRBS:LENGth <numeric>

Parameter

<numeric> = <DECIMAL NUMERIC PROGRAM DATA>
7 \(2^{n}-1\) (n = 7)
9 \(2^{n}-1\) (n = 9)
10 \(2^{n}-1\) (n = 10)
11 \(2^{n}-1\) (n = 11)
15 \(2^{n}-1\) (n = 15)
20 \(2^{n}-1\) (n = 20)
23 \(2^{n}-1\) (n = 23)
31 \(2^{n}-1\) (n = 31)

Function

Sets the number of stages of the PRBS pattern during Mixed Data pattern generation.

Example

To set the number of stages of the PRBS pattern to 15:

> :SOURce:PATTern:MIXData:PRBS:LENGth 15

Compatibility

Incompatible with existing models.

:SOURce:PATTern:MIXData:PRBS:LENGth?

Response

<numeric> = <NR1 NUMERIC RESPONSE DATA>
7, 9, 10, 11, 15, 20, 23, 31

Function

Queries the number of stages of the PRBS pattern during Mixed Data pattern generation.

Example

> :SOURce:PATTern:MIXData:PRBS:LENGth?

< 15

Compatibility

Incompatible with existing models.

:SOURce:PATTern:MIXData:PRBS:MRATio <mratio>

Parameter

<mratio> = <CHARACTER PROGRAM DATA>
M1_2 1/2
M1_4 1/4
M1_8 1/8
M0_8 0/8
I1_2 1/2INVT
M3_4 3/4
M7_8 7/8
M8_8 8/8

Function

Sets the mark ratio for the PRBS pattern during Mixed Data pattern generation.

Example

To set the mark ratio for the PRBS pattern to 1/4:

> :SOURce:PATTern:MIXData:PRBS:MRATio M1_4

Compatibility

Incompatible with existing models.
Chapter 7  SCPI Commands

:SOURce:PATTern:MIXData:PRBS:MRATio?
Response  \(<mratio> = <CHARACTER RESPONSE DATA>\>
\[M1_2, M1_4, M1_8, M0_8, I1_2, M3_4, M7_8, M8_8\]
Function  Queries the mark ratio for the PRBS pattern during Mixed Data pattern generation.
Example  \(> :SOURce:PATTern:MIXData:PRBS:MRATio?\)
\(< M1_4\)
Compatibility  Incompatible with existing models.

:SOURce:PATTern:MIXData:SCRamble <boolean>
Parameter  \(<boolean> = <BOOLEAN PROGRAM DATA>\>
\[0 \text{ or OFF} \quad \text{Scramble OFF}\]
\[1 \text{ or ON} \quad \text{Scramble ON}\]
Function  Sets scramble ON/OFF of the PRBS7 stage during pattern reception.
Example  To set scramble ON of the PRBS7 stage during pattern reception:
\(> :SOURce:PATTern:MIXData:SCRamble 1\)
Compatibility  Incompatible with existing models.

:SOURce:PATTern:MIXData:SCRamble?
Response  \(<numeric> = <NR1 NUMERIC RESPONSE DATA>\>
\[0 \quad \text{Scramble OFF}\]
\[1 \quad \text{Scramble ON}\]
Function  Queries the scramble ON/OFF state of the PRBS7 stage during pattern reception.
Example  \(> :SOURce:PATTern:MIXData:SCRamble?\)
\(< 1\)
Compatibility  Incompatible with existing models.

:SOURce:PATTern:MIXData:PRBS:SEQuence <sequence>
Parameter  \(<sequence> = <CHARACTER PROGRAM DATA>\>
\[\text{RESTart} \quad \text{PRBS patterns are not continuous.}\]
\[\text{CONSecutive} \quad \text{PRBS patterns are continuous.}\]
Function  Sets the PRBS pattern sequence during Mixed Data pattern generation.
Example  To set the PRBS pattern sequence to Restart:
\(> :SOURce:PATTern:MIXData:PRBS:SEQuence \text{ RESTart}\)
Compatibility  Incompatible with existing models.
7.3 PPG Commands

:SOURce:PATTERN:MIXData:PRBS:SEQUence?

| Response | <sequence> = <CHARACTER RESPONSE DATA> REST, CONS |
| Function | Queries the PRBS pattern sequence during Mixed Data pattern generation. |
| Example  | > :SOURce:PATTern:MIXData:PRBS:SEQuence? |
|          | < REST |
| Compatibility | Incompatible with existing models. |

:SOURce:PATTERN:MIXData:SRSetting <row>,<data>,<prbs>

| Parameter | <row> = <DECIMAL NUMERIC PROGRAM DATA> 1 to 16 1 to 16 rows, in 1-row steps <data> = <BOOLEAN PROGRAM DATA> 0 or OFF Scramble OFF 1 or ON Scramble ON Note, however, that Data of 1 Row Scramble is fixed to Scramble OFF. <prbs> = <BOOLEAN PROGRAM DATA> 0 or OFF Scramble OFF 1 or ON Scramble ON |
| Function  | Sets Scramble ON or OFF for the Data and PRBS of the specified Row. |
| Example   | To set the Scramble settings for 2 Row to Data ON, PRBS OFF: |
|          | > :SOURce:PATTern:MIXData:SRSetting 2,1,0 |
| Compatibility | Incompatible with existing models. |

:SOURce:PATTern:MIXData:SRSetting? <row>

| Parameter | <row> = <DECIMAL NUMERIC PROGRAM DATA> 1 to 16 1 to 16 rows, in 1-row steps |
| Response  | <data>,<prbs> = <NR1 NUMERIC RESPONSE DATA> 0 Scramble OFF 1 Scramble ON |
| Function  | Queries the scramble setting (ON/OFF) for the Data and PRBS of the specified Row. |
| Example   | To query the Scramble setting for 2 Row: |
|          | > :SOURce:PATTern:MIXData:SRSetting? 2 |
|          | < 1,0 |
| Compatibility | Incompatible with existing models. |
Chapter 7  SCPI Commands

:SOURce:PATTern:MIXData:SCRamble:ALLSet <numeric>

Parameter

- `<numeric>` = `<NUMERIC PROGRAM DATA>`
- 0    All Reset
- 1    All Set

Function
Sets or resets all scramble.

Example
To select all scramble:

```
> :SOURce:PATTern:MIXData:SCRamble:ALLSet 1
```

Compatibility
Incompatible with existing models.

:SOURce:PATTern:MIXData:BLOCk <numeric>

Parameter

- `<numeric>` = `<DECIMAL NUMERIC PROGRAM DATA>`
- 1 to 511 1 to 511 blocks, in 1-block steps

Function
Sets the number of blocks during Mixed Data pattern generation.

Example
To set the number of blocks to 30:

```
> :SOURce:PATTern:MIXData:BLOCk 30
```

Compatibility
Incompatible with existing models.

:SOURce:PATTern:MIXData:BLOCk?

Response

- `<numeric>` = `<NR1 NUMERIC RESPONSE DATA>`
- 1 to 511 1 to 511 blocks, in 1-block steps

Function
Queries the number of blocks during Mixed Data pattern generation.

Example
To query the number of blocks:

```
> :SOURce:PATTern:MIXData:BLOCk?
< 30
```

Compatibility
Incompatible with existing models.

:SOURce:PATTern:MIXData:RLENght <numeric>

Parameter

- `<numeric>` = `<DECIMAL NUMERIC PROGRAM DATA>`
- 768 to 2281701376 768 to 2,281,701,376 bits, in 128-bit steps

Note:
- At 2 ch Combination, the setting range and Step are doubled.
- At 4 ch Combination, the setting range and Step are quadrupled.

Function
Sets the pattern length of 1 Row that is edited during Mixed Data pattern generation.

Example
To set the pattern length to 768 bits:

```
> :SOURce:PATTern:MIXData:RLENght 768
```

Compatibility
Incompatible with existing models.
7.3 PPG Commands

:SOURce:PATTern:MIXData:RLENgth?

Response  
<numeric> = <NR1 NUMERIC RESPONSE DATA>
768 to 2281701376  768 to 2,281,701,376 bits, in 128-bit steps

Function  
Queries the pattern length of 1 Row that is edited during Mixed Data pattern generation.

Example  
> :SOURce:PATTern:MIXData:RLENgth?
< 768

Compatibility  
Incompatible with existing models.

:SOURce:PATTern:MIXData:ROW <numeric>

Parameter  
<numeric> = <DECIMAL NUMERIC PROGRAM DATA>
1 to 16  1 to 16 rows, in 1-row steps

Function  
Sets the number of rows within one block that are edited during Mixed Data pattern generation.

Example  
To set the number of rows within one block to 16:
> :SOURce:PATTern:MIXData:ROW 16

Compatibility  
Incompatible with existing models.

:SOURce:PATTern:MIXData:ROW?

Response  
<numeric> = <NR1 NUMERIC RESPONSE DATA>
1 to 16  1 to 16 rows, in 1-row steps

Function  
Queries the number of rows within one block that are edited during Mixed Data pattern generation.

Example  
> :SOURce:PATTern:MIXData:ROW?
< 16

Compatibility  
Incompatible with existing models.
Chapter 7  SCPI Commands

:SOURce:PATTern:MIXData:DATA:LENGth <numeric>

Parameter

<numeric> = <DECIMAL NUMERIC PROGRAM DATA>
512 to 134217728  512 to 134,217,728 bits, in 1-bit steps

Note:

The maximum setting is the pattern length.
At 2 ch Combination, the setting range and Step are doubled.
The maximum setting is restricted to Row Length – 64 bits
At 4 ch Combination, the setting range and Step are quadrupled.
The maximum setting is restricted to Row Length – 128 bits

Function

Sets the pattern length on the pattern data side that is edited during
Mixed Data pattern generation.

Example

To set the pattern length to 512 bits.
> :SOURce:PATTern:MIXData:DATA:LENGth 512

Compatibility

Incompatible with existing models.

:SOURce:PATTern:MIXData:DATA:LENGth?

Response

<numeric> = <NR1 NUMERIC RESPONSE DATA>
512 to 134217728  512 to 134,217,728 bits, in 1-bit steps

Function

Queries the pattern length on the pattern data side that is edited during
Mixed Data pattern generation.

Example

> :SOURce:PATTern:MIXData:DATA:LENGth?
< 512

Compatibility

Incompatible with existing models.
7.3 PPG Commands

:SOURce:PATTern:MIXData:DATA:WHOLe <block>,<start>,<end>,<data>

Parameter

- `<block>` = `<DECIMAL NUMERIC PROGRAM DATA>`
  1 to 511  1 to 511 blocks, in 1-block steps
- `<start>` = `<NON-DECIMAL PROGRAM DATA>`
  #H0 to #H7FFFFFF 0 to 7FFFFFF bits, in 1-bit steps
  (Specify in hexadecimal)
- `<end>` = `<NON-DECIMAL PROGRAM DATA>`
  #H0 to #H7FFFFFF 0 to 7FFFFFF bits, in 1-bit steps
  (Specify in hexadecimal)

>Note:
The maximum setting is the pattern length.
At 2 ch Combination, the setting range is doubled.
At 4 ch Combination, the setting range is quadrupled.

- `<data>` = `<STRING PROGRAM DATA>`

  "H***" Specify pattern data in hexadecimal format.
  1 to 400 characters (pattern data of 400 × 4 bits)
  Specify a character string consisting of 0 to 9
  and A to F.
  If the last character string of the pattern data is
  short of a hexadecimal unit (4 bits), add 0(s) up
  to the last bit to make a hexadecimal string.

  "B***" Specify pattern data in binary format.
  1 to 400 characters (pattern data of 400 bits)
  Specify a character string consisting of 0 and 1.

Function
Sets the pattern data from the `<start>` to `<end>` addresses during Mixed
Data pattern generation.
The set data overwrites the specified range.

Example
To set the addresses 0 to 1F of the block 1 pattern data to 11:
> :SOURce:PATTern:MIXData:DATA:WHOLe 1,#H0,#H1F, "H11"

Compatibility
Incompatible with existing models.
### :SOURce:PATTern:MIXData:DATA:WHOLe? <block>,<start>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;block&gt;</td>
<td>&lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
</tr>
<tr>
<td></td>
<td>1 to 511, 1 to 511 blocks, in 1-block steps</td>
</tr>
<tr>
<td>&lt;start&gt;</td>
<td>&lt;NON-DECIMAL PROGRAM DATA&gt;</td>
</tr>
<tr>
<td></td>
<td>#H0 to #H7FFFFFF, 0 to 7FFFFFF bits, in 1-bit steps</td>
</tr>
<tr>
<td></td>
<td>(Specify in hexadecimal)</td>
</tr>
</tbody>
</table>

**Note:**

- The maximum setting is the pattern length.
- At 2 ch Combination, the setting range is doubled.
- At 4 ch Combination, the setting range is quadrupled.

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;data&gt; = &lt;STRING RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>&quot;H***&quot; Returns the pattern data in hexadecimal format.</td>
</tr>
<tr>
<td></td>
<td>Within 400 characters (pattern data of 400 x 4 bits)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Function</th>
<th>Queries the pattern data of 400 x 4 bits from the &lt;start&gt; address during Mixed Data pattern generation.</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Example</th>
<th>To query the block 1 pattern data from address 0:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>&gt; :SOURce:PATTern:MIXData:DATA:WHOLe? 1,#H0</td>
</tr>
<tr>
<td></td>
<td>&lt; &quot;H11&quot;</td>
</tr>
</tbody>
</table>

| Compatibility      | Incompatible with existing models.                                                                     |
**:SOURce:PATTern:MIXData:BDATa:WHOLe**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| **<block>**      | = <DECIMAL NUMERIC PROGRAM DATA>  
                   | 1 to 511  
                   | 1 to 511 blocks, in 1-block steps                                                                 |
| **<start>**      | = <NON-DECIMAL PROGRAM DATA>  
                   | #H0 to #H7FFFFFF 0 to 7FFFFFF bits, in 1-bit steps  
                   | (Specify in hexadecimal)                                                                          |
| **<end>**        | = <NON-DECIMAL PROGRAM DATA>  
                   | #H0 to #H7FFFFFF 0 to 7FFFFFF bits, in 1-bit steps  
                   | (Specify in hexadecimal)                                                                          |

**Note:**

- The maximum setting is the pattern length.
- At 2 ch Combination, the setting range is doubled.
- At 4 ch Combination, the setting range is quadrupled.

| **<bdata>**      | = <ARBITRARY BLOCK PROGRAM DATA>  
                   | #XYYY<binary>  
                   | X: Number of YYY digits  
                   | YYY: Number of bytes of <binary>  
                   | 1 to 16,777,216 bytes  
                   | <binary>: Binary data up to 16,777,216 bytes  

**Function**

Sets the binary data from **<start>** to **<end>** addresses of the pattern data during Mixed Data pattern generation.

The set data overwrites the specified range.

**Example**

To set the addresses 0 to 1F of the block 2 pattern data to 43:

```
> :SOURce:PATTern:MIXData:BDATa:WHOLe 2,#H0,#H1F,#11C
(C=43)
```

**Compatibility**

Incompatible with existing models.
Chapter 7  SCPI Commands

:SOURce:PATTern:MIXData:BDATa:WHOLe? <block>,<start>[,<size>]

Parameter

- **<block>** = <DECIMAL NUMERIC PROGRAM DATA>
  - 1 to 511
    - 1 to 511 blocks, in 1-block steps
- **<start>** = <NON-DECIMAL PROGRAM DATA>
  - #H0 to #H7FFFFFF
    - 0 to 7FFFFFFF bits, in 1-bit steps
    - (Specify in hexadecimal)

*Note:*

The maximum setting is the pattern length.
At 2 ch Combination, the setting range is doubled.
At 4 ch Combination, the setting range is quadrupled.

- **<size>** = <NR1 NUMERIC PROGRAM DATA>
  - 1 to 16777216
    - 1 to 16,777,216 bytes, in 1-byte steps

Response

- **<bdata>** = <DEFINITE LENGTH ARBITRARY BLOCK RESPONSE DATA>

  - #XYY<binary>
    - X: Number of YYY digits
    - YYY: Number of bytes of <binary>
    - Binary data up to 16,777,216 bytes

Binary data of the setting pattern is queried when [<size>] is omitted.

Function

Queries binary data of bytes specified by <size> from the <start> address of the pattern data during Mixed Data pattern generation.

Example

To query binary data of the block 1 pattern data starting from address 0:

```
> :SOURce:PATTern:MIXData:BDATa:WHOLe? 1,#H0
< #11C
```

Compatibility

Incompatible with existing models.
### :SOURce:PATTern:MIXData:DREVerse:ADDRess \( \langle \text{block} \rangle, \langle \text{start} \rangle, \langle \text{end} \rangle \)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \langle \text{block} \rangle )</td>
<td>(&lt;\text{DECIMAL NUMERIC PROGRAM DATA}&gt;) 1 to 511 1 to 511 blocks, in 1-block steps</td>
</tr>
<tr>
<td>( \langle \text{start} \rangle )</td>
<td>(&lt;\text{NON-DECIMAL PROGRAM DATA}&gt;) #H0 to #H7FFFFFF 0 to 7FFFFFFF bits, in 1-bit steps (Specify in hexadecimal)</td>
</tr>
<tr>
<td>( \langle \text{end} \rangle )</td>
<td>(&lt;\text{NON-DECIMAL PROGRAM DATA}&gt;) #H0 to #H7FFFFFF 0 to 7FFFFFFF bits, in 1-bit steps (Specify in hexadecimal)</td>
</tr>
</tbody>
</table>

**Note:**
The maximum setting is the pattern length. At 2 ch Combination, the setting range is doubled. At 4 ch Combination, the setting range is quadrupled.

**Function:**
Inverts the data in the program during Mixed Data pattern generation. Specify the inversion range by the \( \langle \text{start} \rangle \) and \( \langle \text{end} \rangle \) addresses.

**Example:**
To invert addresses 0 to 4F of the block 3 pattern data:

\[ > :\text{SOURce:PATTern:MIXData:DREVerse:ADDRESS} \ 3, \#H0, \#H4F \]

**Compatibility:**
Incompatible with existing models.

### :SOURce:PATTern:MIXData:DREVerse:DELTa \( \langle \text{block} \rangle, \langle \text{start} \rangle, \langle \text{delta} \rangle \)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \langle \text{block} \rangle )</td>
<td>(&lt;\text{DECIMAL NUMERIC PROGRAM DATA}&gt;) 1 to 511 1 to 511 blocks, in 1-block steps</td>
</tr>
<tr>
<td>( \langle \text{start} \rangle )</td>
<td>(&lt;\text{NON-DECIMAL PROGRAM DATA}&gt;) #H0 to #H7FFFFFF 0 to 7FFFFFFF bits, in 1-bit steps (Specify in hexadecimal)</td>
</tr>
<tr>
<td>( \langle \text{delta} \rangle )</td>
<td>(&lt;\text{DECIMAL NUMERIC PROGRAM DATA}&gt;) 1 to 134217728 1 to 134,217,728 bits, in 1-bit steps</td>
</tr>
</tbody>
</table>

**Note:**
The maximum setting is the pattern length. At 2 ch Combination, the setting range is doubled. At 4 ch Combination, the setting range is quadrupled.

**Function:**
Inverts the data in the program during Mixed Data pattern generation. Specify the inversion range by the number of bits from the \( \langle \text{start} \rangle \) address (relative specification).

**Example:**
To invert 256 bits of the block 3 pattern data from address 0:

\[ > :\text{SOURce:PATTern:MIXData:DREVerse:DELTa} \ 3, \#H0, 256 \]

**Compatibility:**
Incompatible with existing models.
Chapter 7  SCPI Commands

:SOURce:PArAmeter:MIXData:DATA:FILL <block>,<range>,<page>,<data>

Parameter

- **<block>** = <DECIMAL NUMERIC PROGRAM DATA>
  - 1 to 511  1 to 511 blocks, in 1-block steps

- **<range>** = <CHARACTER PROGRAM DATA>
  - PAGE  Specifies a page.
    - (One page is defined as 128 bits.)
  - ALL  Specifies all data.

- **<page>** = <DECIMAL NUMERIC PROGRAM DATA>
  - 1 to 1,048,576  1 to 1,048,576 (Max.) pages, in 1-page steps
  - Max = \( \frac{\text{Pattern Length}}{128} \) (rounding up fractions)
  - Specify "0" when <range> is set to ALL.

- **<data>** = <DECIMAL NUMERIC PROGRAM DATA>
  - 0  Fills the specified range with 0s.
  - 1  Fills the specified range with 1s.

Function

Sets 0s or 1s for the specified page or all data in the program during Mixed Data pattern generation.

Example

To set 0s for the third page of the block 511 pattern data:

> :SOURce:PArAmeter:MIXData:DATA:FILL 511,PAGE,3,0

Compatibility

Incompatible with existing models.
7.3.2.7 Mixed Alternate pattern setting commands

Figure 7.3.2.7-1 Mixed Alternate pattern setting

Figure 7.3.2.7-2 Scramble Setup dialog box
Chapter 7  SCPI Commands

Table 7.3.2.7-1  Mixed Alternate pattern setting commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>:SOURce:PATTern:MIXalt:PRBS:BITShift?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SOURce:PATTern:MIXalt:LOOPcont?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SOURce:PATTern:MIXalt:LTIMe?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SOURce:PATTern:MIXalt:PRBS:LENGth?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SOURce:PATTern:MIXalt:PRBS:MRATio?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SOURce:PATTern:MIXalt:SCRamble?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SOURce:PATTern:MIXalt:PRBS:SEQUence?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SOURce:PATTern:MIXalt:SRSetting?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SOURce:PATTern:MIXalt:BLOCk</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SOURce:PATTern:MIXalt:BLOCk?</td>
</tr>
</tbody>
</table>

Figure 7.3.2.7-3  Mixed Alternate pattern setting (Pattern Editor screen)
## Table 7.3.2.7-1 Mixed Alternate pattern setting commands (Cont’d)

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>:SOURce:PATTern:MIXalt:RLENgth?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SOURce:PATTern:MIXalt:ROW?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SOURce:PATTern:MIXalt:DATA:LENGth?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SOURce:PATTern:MIXalt:DATA:WHOLe?</td>
</tr>
</tbody>
</table>
| [16] | No label (Pattern setting) | :SOURce:PATTern:MIXalt:BDATa:WHOLe  
| | | :SOURce:PATTern:MIXalt:BDATa:WHOLe? |
| [18] | No label (Data inversion) | :SOURce:PATTern:MIXalt:DREVerse:DELTa |
| [19] | No label (All 0 or All 1) | :SOURce:PATTern:MIXalt:DATA:FILL |

### :SOURce:PATTern:MIXalt:PRBS:BITShift <numeric>

- **Parameter**: 
  
  - `<numeric>` = `<DECIMAL NUMERIC PROGRAM DATA>`
  - 1 1 bit
  - 3 3 bits

- **Function**: Sets the bit shift for the PRBS pattern during Mixed Alternate pattern generation.

- **Example**: 
  
  - To set the bit shift to 3-bit shift:
  - `> :SOURce:PATTern:MIXalt:PRBS:BITShift 3`

- **Compatibility**: 
  
  - Incompatible with existing models.

### :SOURce:PATTern:MIXalt:PRBS:BITShift? 

- **Response**: 
  
  - `<numeric>` = `<NR1 NUMERIC RESPONSE DATA>`
  - 1, 3

- **Function**: Queries the bit shift for the PRBS pattern during Mixed Alternate pattern generation.

- **Example**: 
  
  - `> :SOURce:PATTern:MIXalt:PRBS:BITShift?`
  - `< 3`

- **Compatibility**: 
  
  - Incompatible with existing models.
**Chapter 7  SCPI Commands**

### :SOURce:PATTern:MIXalt:LOOPcont <loop>

**Parameter**

- `<loop> = <CHARACTER PROGRAM DATA>`
  - INTernal: Switching control using internal signal
  - EXTernal: Switching control using external signal
  - MANual: Manual switching control

**Function**

Sets the pattern A/B switching control method during Mixed Alternate pattern generation.

**Example**

To set the manual control for the pattern A/B switching:

> :SOURce:PATTern:MIXalt:LOOPcont MANual

**Compatibility**

Incompatible with existing models.

### :SOURce:PATTern:MIXalt:LOOPcont? 

**Response**

- `<loop> = <CHARACTER RESPONSE DATA>`
  - INT, EXT, MAN

**Function**

Queries the pattern A/B switching control method during Mixed Alternate pattern generation.

**Example**

> :SOURce:PATTern:MIXalt:LOOPcont?

< MAN

**Compatibility**

Incompatible with existing models.

### :SOURce:PATTern:MIXalt:MANual

**Function**

Generates the pattern B for the specified number of Loop times during Mixed Alternate pattern generation.

**Example**

> :SOURce:PATTern:MIXalt:MANual

**Compatibility**

Incompatible with existing models.

### :SOURce:PATTern:MIXalt:LTIMe <content>,<numeric>

**Parameter**

- `<content>=<CHARACTER PROGRAM DATA>`
  - APAT: Pattern A
  - BPAT: Pattern B
- `<numeric> = <DECIMAL NUMERIC PROGRAM DATA>`
  - 1 to 511: 1 to 511 times, in 1-time steps

**Function**

Sets the number of generation times of the Alternate pattern A or B.

**Example**

To set the number of pattern B generation times to 20:

> :SOURce:PATTern:MIXalt:LTIMe BPAT,20

**Compatibility**

Incompatible with existing models.
7.3 PPG Commands

:SOURce:PATTern:MIXalt:LTIMe? <content>

Parameter

<content> = <CHARACTER PROGRAM DATA>
APAT, BPAT

Response

<nr1 NUMERIC RESPONSE DATA>
1 to 511 1 to 511 times, in 1-time steps

Function
Queries the number of generation times of the Alternate pattern A or B.

Example
To query the number of pattern B generation times:

> :SOURce:PATTern:MIXalt:LTIMe? BPAT
< 20

Compatibility
Incompatible with existing models.

:SOURce:PATTern:MIXalt:PRBS:LENGth <numeric>

Parameter

<nr1 NUMERIC PROGRAM DATA>
7 \(2^n-1 (n = 7)\)
9 \(2^n-1 (n = 9)\)
10 \(2^n-1 (n = 10)\)
11 \(2^n-1 (n = 11)\)
15 \(2^n-1 (n = 15)\)
20 \(2^n-1 (n = 20)\)
23 \(2^n-1 (n = 23)\)
31 \(2^n-1 (n = 31)\)

Function
Sets the number of stages of the PRBS pattern during Mixed Alternate pattern generation.

Example
To set the number of stages of the PRBS pattern to 7:

> :SOURce:PATTern:MIXalt:PRBS:LENGth 7

Compatibility
Incompatible with existing models.

:SOURce:PATTern:MIXalt:PRBS:LENGth?

Response

<nr1 NUMERIC RESPONSE DATA>
7, 9, 10, 11, 15, 20, 23, 31

Function
Queries the number of stages of the PRBS pattern during Mixed Alternate pattern generation.

Example

> :SOURce:PATTern:MIXalt:PRBS:LENGth?
< 7

Compatibility
Incompatible with existing models.
Chapter 7  SCPI Commands

:SOURce:PATTern:MIXalt:PRBS:MRATio <mratio>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;mratio&gt; = &lt;CHARACTER PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>M1_2</td>
<td>1/2</td>
</tr>
<tr>
<td>M1_4</td>
<td>1/4</td>
</tr>
<tr>
<td>M1_8</td>
<td>1/8</td>
</tr>
<tr>
<td>M0_8</td>
<td>0/8</td>
</tr>
<tr>
<td>I1_2</td>
<td>1/2INVT</td>
</tr>
<tr>
<td>M3_4</td>
<td>3/4</td>
</tr>
<tr>
<td>M7_8</td>
<td>7/8</td>
</tr>
<tr>
<td>M8_8</td>
<td>8/8</td>
</tr>
</tbody>
</table>

Function  
Sets the mark ratio for the PRBS pattern during Mixed Alternate pattern generation.

Example  
To set the mark ratio for the PRBS pattern to 1/8:
>:SOURce:PATTern:MIXalt:PRBS:MRATio M1_8

Compatibility  
Incompatible with existing models.

:SOURce:PATTern:MIXalt:PRBS:MRATio?

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;mratio&gt; = &lt;CHARACTER RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>M1_2, M1_4, M1_8, M0_8, I1_2, M3_4, M7_8, M8_8</td>
<td></td>
</tr>
</tbody>
</table>

Function  
Queries the mark ratio for the PRBS pattern during Mixed Alternate pattern generation.

Example  
>:SOURce:PATTern:MIXalt:PRBS:MRATio?
< M1_8

Compatibility  
Incompatible with existing models.

:SOURce:PATTern:MIXalt:SCRamble <boolean>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;boolean&gt; = &lt;BOOLEAN PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 or OFF</td>
<td>Scramble OFF</td>
</tr>
<tr>
<td>1 or ON</td>
<td>Scramble ON</td>
</tr>
</tbody>
</table>

Function  
Sets scramble ON/OFF of the PRBS7 stage during pattern reception.

Example  
To set scramble ON of the PRBS7 stage during pattern reception:
>:SOURce:PATTern:MIXalt:SCRamble 1

Compatibility  
Incompatible with existing models.
### :SOURce:PATTern:MIXalt:SCRamble?

**Response**

<table>
<thead>
<tr>
<th>&lt;numeric&gt; = NR1 NUMERIC RESPONSE DATA</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
</tr>
<tr>
<td>1</td>
</tr>
</tbody>
</table>

**Function**

Queries the scramble ON/OFF state of the PRBS7 stage during pattern reception.

**Example**

```plaintext
> :SOURce:PATTern:MIXalt:SCRamble?
< 1
```

**Compatibility**

Incompatible with existing models.

---

### :SOURce:PATTern:MIXalt:PRBS:SEQuence <sequence>

**Parameter**

- <sequence> = CHARACTER PROGRAM DATA
- **RESTart**
  - PRBS patterns are not continuous.
- **CONSecutive**
  - PRBS patterns are continuous.

**Function**

Sets the PRBS pattern sequence during Mixed Alternate pattern generation.

**Example**

To set the PRBS pattern sequence to Restart:

```plaintext
> :SOURce:PATTern:MIXalt:PRBS:SEQuence RESTart
```

**Compatibility**

Incompatible with existing models.

---

### :SOURce:PATTern:MIXalt:PRBS:SEQuence?

**Response**

<table>
<thead>
<tr>
<th>&lt;sequence&gt; = CHARACTER RESPONSE DATA</th>
</tr>
</thead>
<tbody>
<tr>
<td>REST, CONS</td>
</tr>
</tbody>
</table>

**Function**

Queries the PRBS pattern sequence during Mixed Alternate pattern generation.

**Example**

```plaintext
> :SOURce:PATTern:MIXalt:PRBS:SEQuence?
< REST
```

**Compatibility**

Incompatible with existing models.
**:SOURce:PATTern:MIXalt:SRSetting <row>,<data>,<prbs>**

Parameter

- `<row>` = <DECIMAL NUMERIC PROGRAM DATA>
  - 1 to 16 rows, in 1-row steps
- `<data>` = <BOOLEAN PROGRAM DATA>
  - 0 or OFF Scramble OFF
  - 1 or ON Scramble ON
- `<prbs>` = <BOOLEAN PROGRAM DATA>
  - 0 or OFF Scramble OFF
  - 1 or ON Scramble ON

Note, however, that Scramble ON/OFF cannot be set for Data of 1 Row.

Function
Sets Scramble ON or OFF for the Data and PRBS of the specified Row.

Example
To set the Scramble settings for 3 Row to Data ON, PRBS OFF:
> :SOURce:PATTern:MIXalt:SRSetting 3,1,0

Compatibility
Incompatible with existing models.

**:SOURce:PATTern:MIXalt:SRSetting? <row>**

Parameter

- `<row>` = <DECIMAL NUMERIC PROGRAM DATA>
  - 1 to 16 rows, in 1-row steps

Response

- `<data>,<prbs>` = <NR1 NUMERIC RESPONSE DATA>
  - 0 Scramble OFF
  - 1 Scramble ON

Function
Queries the scramble setting (ON/OFF) for the Data and PRBS of the specified Row.

Example
To query the Scramble setting for 3 Row:
> :SOURce:PATTern:MIXalt:SRSetting? 3
< 1,0

Compatibility
Incompatible with existing models.

**:SOURce:PATTern:MIXalt:SCRamble:ALLSet <numeric>**

Parameter

- `<numeric>` = <NUMERIC PROGRAM DATA>
  - 0 All Reset
  - 1 All Set

Function
Sets or resets all scramble.

Example
To select all scramble:
> :SOURce:PATTern:MIXalt:SCRamble:ALLSet 1

Compatibility
Incompatible with existing models.
7.3 PPG Commands

**:SOURce:PATTern:MIXalt:BLOCk <numeric>**

Parameter  

<numeric> = <DECIMAL NUMERIC PROGRAM DATA>

1 to 511  

1 to 511 blocks, in 1-block steps

Function  

Sets the number of blocks during Mixed Alternate pattern generation.

Example  

To set the number of blocks to 12:

> :SOURce:PATTern:MIXalt:BLOCk,12

Compatibility  

Incompatible with existing models.

**:SOURce:PATTern:MIXalt:BLOCk?**

Response  

<numeric> = <NR1 NUMERIC RESPONSE DATA>

1 to 511  

1 to 511 blocks, in 1-block steps

Function  

Queries the number of blocks during Mixed Alternate pattern generation.

Example  

> :SOURce:PATTern:MIXalt:BLOCk?

< 12

Compatibility  

Incompatible with existing models.

**:SOURce:PATTern:MIXalt:RLENgth <numeric>**

Parameter  

<numeric> = <DECIMAL NUMERIC PROGRAM DATA>

768 to 2214592512  

768 to 2,214,592,512 bits, in 128-bit steps

Function  

Sets the pattern length of 1 Row that is edited during Mixed Alternate pattern generation.

Example  

To set the pattern length of 1 Row to 768 bits:

> :SOURce:PATTern:MIXalt:RLENgth 768

Compatibility  

Incompatible with existing models.

**:SOURce:PATTern:MIXalt:RLENgth?**

Response  

<numeric> = <NR1 NUMERIC RESPONSE DATA>

768 to 2214592512  

768 to 2,214,592,512 bits, in 128-bit steps

Function  

Queries the pattern length of 1 Row that is edited during Mixed Alternate pattern generation.

Example  

> :SOURce:PATTern:MIXalt:RLENgth?

< 768

Compatibility  

Incompatible with existing models.
Chapter 7  SCPI Commands

:SOURce:PATTern:MIXalt:ROW <numeric>

Parameter  
<numeric> = <DECIMAL NUMERIC PROGRAM DATA>
1 to 16 1 to 16 rows, in 1-row steps

Function  
Sets the number of rows within one block that are edited during Mixed Alternate pattern generation.

Example  
To set the number of rows within one block to 8:
> :SOURce:PATTern:MIXalt:ROW,8

Compatibility  
Incompatible with existing models.

:SOURce:PATTern:MIXalt:ROW?

Response  
<numeric> = <NR1 NUMERIC RESPONSE DATA>
1 to 16 1 to 16 rows, in 1-row steps

Function  
Queries the number of rows within one block that are edited during Mixed Alternate pattern generation.

Example  
> :SOURce:PATTern:MIXalt:ROW?
< 8

Compatibility  
Incompatible with existing models.

:SOURce:PATTern:MIXalt:DATA:LENGth <numeric>

Parameter  
<numeric> = <DECIMAL NUMERIC PROGRAM DATA>
512 to 67108864 512 to 67,108,864 bits, in 128-bit steps

Function  
Sets the pattern length on the pattern data side that is edited during Mixed Alternate pattern generation.

Example  
To set the pattern length on the pattern data side to 12800 bits:
> :SOURce:PATTern:MIXalt:DATA:LENGth 12800

Compatibility  
Incompatible with existing models.

:SOURce:PATTern:MIXalt:DATA:LENGth?

Response  
<numeric> = <NR1 NUMERIC RESPONSE DATA>
512 to 67108864 512 to 67,108,864 bits, in 128-bit steps

Function  
Queries the pattern length on the pattern data side that is edited during Mixed Alternate pattern generation.

Example  
> :SOURce:PATTern:MIXalt:DATA:LENGth?
< 12800

Compatibility  
Incompatible with existing models.

Parameter

- **<channel>** = <CHARACTER PROGRAM DATA>
- **<data>** = <STRING PROGRAM DATA>
  - Specify pattern data in hexadecimal format.
  - 1 to 400 characters (pattern data of 400 \* 4 bits)
  - Specify a character string consisting of 0 to 9 and A to F.
  - If the last character string of the pattern data is short of a hexadecimal unit (4 bits), add 0(s) up to the last bit to make a hexadecimal string.

Function

Sets the pattern data from the <start> to <end> addresses during Mixed Alternate pattern generation.

Example

To set the addresses 0 to 1F of the block 1 pattern A data to AA:

```
```

Compatibility

Incompatible with existing models.
Chapter 7  SCPI Commands

:SOURce:PATTern:MIXalt:DATA:WHOLe? <block>,<content>,<start>

Parameter

- `<block>` = <DECIMAL NUMERIC PROGRAM DATA>
  1 to 511  1 to 511 blocks, in 1-block steps
- `<content>` = <CHARACTER PROGRAM DATA>
  APAT  Pattern A
  BPAT  Pattern B
- `<start>` = <NON-DECIMAL PROGRAM DATA>
  #H0 to #H3FFFFFF 0 to 3FFFFFFF bits, in 1-bit steps
  (Specify in hexadecimal)

Note:

The maximum setting value is the pattern length.

Response

- `<data>` = <STRING RESPONSE DATA>
  "H***" Returns the pattern data in hexadecimal format.
  Within 400 characters (pattern data of 400 × 4 bits)

Function

Queries the pattern data of 400 × 4 bits from the `<start>` address during Mixed Alternate pattern generation.

Example

To query the block 1 pattern A data from address 0:

> :SOURce:PATTern:MIXalt:DATA:WHOLe? 1,APAT,#H0
< "HAA"

Compatibility

Incompatible with existing models.
### 7.3 PPG Commands

#### :SOURce:PATTern:MIXalt:BDATa:WHOLe

`<content>,<start>,<end>,<bdata>`

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;block&gt;</code></td>
<td><code>&lt;DECIMAL NUMERIC PROGRAM DATA&gt;</code>&lt;br&gt;1 to 511 blocks, in 1-block steps</td>
</tr>
<tr>
<td><code>&lt;content&gt;</code></td>
<td><code>&lt;CHARACTER PROGRAM DATA&gt;</code>&lt;br&gt;APAT: Pattern A&lt;br&gt;BPAT: Pattern B</td>
</tr>
<tr>
<td><code>&lt;start&gt;</code></td>
<td><code>&lt;NON-DECIMAL PROGRAM DATA&gt;</code>&lt;br&gt;#H0 to #H3FFFFFF 0 to 3FFFFFF bits, in 1-bit steps (Specify in hexadecimal)</td>
</tr>
<tr>
<td><code>&lt;end&gt;</code></td>
<td><code>&lt;NON-DECIMAL PROGRAM DATA&gt;</code>&lt;br&gt;#H0 to #H3FFFFFF 0 to 3FFFFFF bits, in 1-bit steps (Specify in hexadecimal)</td>
</tr>
</tbody>
</table>
| `<bdata>`     | `X: Number of YYY digits<br>YYY: Number of bytes of <binary>`<br>1 to 8,388,608 bytes<br><binary>: Binary data up to 8,388,608 bytes |`

**Note:**
The maximum setting value is the pattern length.

**Function**
Sets binary data of the pattern data from `<start>` to `<end>` addresses during Mixed Alternate pattern generation.
The set data overwrites the specified range.

**Example**
To set the addresses 0 to 1F of the block 3 pattern A data to 42:
```
> :SOURce:PATTern:MIXalt:BDATa:WHOLe 3,APAT,#H0,#H1F,#11B
```

**Compatibility**
Incompatible with existing models.
Chapter 7  SCPI Commands

:SOURce:PATTern:MIXalt:BDATa:WHOLe?  
<block>,<content>,<start>[,<size>]

Parameter

- `<block>` = `<DECIMAL NUMERIC PROGRAM DATA>`
  
  1 to 511 blocks, in 1-block steps

- `<content>` = `<CHARACTER PROGRAM DATA>`
  
  APAT Pattern A
  
  BPAT Pattern B

- `<start>` = `<NON-DECIMAL PROGRAM DATA>`
  
  #H0 to #H3FFFFFF 0 to 3FFFFFFF bits, in 1-bit steps
  
  (Specify in hexadecimal)

Note:

The maximum setting value is the pattern length.

- `<size>` = `<NR1 NUMERIC PROGRAM DATA>`
  
  1 to 8388608 1 to 8,388,608 bytes, in 1-byte steps

Binary data of the setting pattern is queried when `<size>` is omitted.

Response

- `<bdata>` = `<DEFINITE LENGTH ARBITRARY BLOCK RESPONSE DATA>`
  
  #XYYY<binary>  
  
  X: Number of YYY digits
  
  YYY: Number of bytes of <binary>
  
  1 to 8,388,608 bytes

  <binary>: Binary data up to 8,388,608 bytes

Function

Queries binary data of bytes specified by `<size>` from `<start>` address of the pattern data during Mixed Alternate pattern generation.

Example

To query binary data of the block 3 pattern A data starting from address 0:

```
> :SOURce:PATTern:MIXalt:BDATa:WHOLe? 3,APAT,#H0
< #11B (B=42)
```

Compatibility

Incompatible with existing models.
7.3 PPG Commands

:SOURce:PATTern:MIXalt:DREVerse:ADDRess

\[\text{Parameter} \quad \langle \text{block} \rangle, \langle \text{content} \rangle, \langle \text{start} \rangle, \langle \text{end} \rangle\]

\[\langle \text{block} \rangle = \langle \text{DECIMAL NUMERIC PROGRAM DATA} \rangle\]
1 to 511 1 to 511 blocks, in 1-block steps

\[\langle \text{content} \rangle = \langle \text{CHARACTER PROGRAM DATA} \rangle\]
APAT Pattern A
BPAT Pattern B

\[\langle \text{start} \rangle = \langle \text{NON-DECIMAL PROGRAM DATA} \rangle\]
#H0 to #H3FFFFFF 0 to 3FFFFFF bits, in 1-bit steps
(Specify in hexadecimal)

\[\langle \text{end} \rangle = \langle \text{NON-DECIMAL PROGRAM DATA} \rangle\]
#H0 to #H3FFFFFF 0 to 3FFFFFF bits, in 1-bit steps
(Specify in hexadecimal)

**Note:**
The maximum setting value is the pattern length.

**Function**
Inverts the data in the program during Mixed Alternate pattern generation. Specify the inversion range by the \langle start \rangle and \langle end \rangle addresses.

**Example**
To invert addresses 0 to 1F of the block 3 pattern A data:
\[> :\text{SOURce:PATTern:MIXalt:DREVerse:ADDRess} \ 3, \text{APAT}, \#H0, \#H1F\]

**Compatibility**
Incompatible with existing models.

:SOURce:PATTern:MIXalt:DREVerse:DELTa

\[\langle \text{block} \rangle, \langle \text{content} \rangle, \langle \text{start} \rangle, \langle \text{delta} \rangle\]

\[\langle \text{block} \rangle = \langle \text{DECIMAL NUMERIC PROGRAM DATA} \rangle\]
1 to 511 1 to 511 blocks, in 1-block steps

\[\langle \text{content} \rangle = \langle \text{CHARACTER PROGRAM DATA} \rangle\]
APAT Pattern A
BPAT Pattern B

\[\langle \text{start} \rangle = \langle \text{NON-DECIMAL PROGRAM DATA} \rangle\]
#H0 to #H3FFFFFF 0 to 3FFFFFF bits, in 1-bit steps
(Specify in hexadecimal)

\[\langle \text{delta} \rangle = \langle \text{DECIMAL NUMERIC PROGRAM DATA} \rangle\]
1 to 67108864 1 to 67,108,864 bits, in 1-bit steps

**Note:**
The maximum setting value is the pattern length.

**Function**
Inverts the data in the program during Mixed Alternate pattern generation. Specify the inversion range by the number of bits from the \langle start \rangle address (relative specification).

**Example**
To invert 64 bits of the block 3 pattern A data from address 0:
\[> :\text{SOURce:PATTern:MIXalt:DREVerse:DELTa} \ 3, \text{APAT}, \#H0, 64\]

**Compatibility**
Incompatible with existing models.
<block>,<content>,<range>,<page>,<data>

Parameter

- `<block>` = `<DECIMAL NUMERIC PROGRAM DATA>`
  1 to 511 1 to 511 blocks, in 1-block steps
- `<content>` = `<CHARACTER PROGRAM DATA>`
  APAT Pattern A
  BPAT Pattern B
- `<range>` = `<CHARACTER PROGRAM DATA>`
  PAGE Specifies a page.
  (One page is defined as 128 bits.)
  ALL Specifies all data.
- `<page>` = `<DECIMAL NUMERIC PROGRAM DATA>`
  1 to 524288 1 to 524,288 (Max.) pages, in 1-page steps
  Max = \[
  \frac{\text{Pattern Length}}{128}
  \] (rounding up fractions)
  Specify "0" when `<range>` is set to ALL.
- `<data>` = `<DECIMAL NUMERIC PROGRAM DATA>`
  0 Fills the specified range with 0s.
  1 Fills the specified range with 1s.

Function

Sets 0s or 1s for the specified page or all data in the program during Mixed Alternate pattern generation.

Example

To set 1s for all the data of the block 3 pattern A data:
> :SOURCE:PATTERN:MIXalt:DATA:FILL 3,APAT,ALL,0,1

Compatibility

Incompatible with existing models.
7.3.2.8 Sequence pattern setting commands

Figure 7.3.2.8-1 Sequence pattern setting
Figure 7.3.2.8-2 Sequence Pattern Setting screen
7.3 PPG Commands

Table 7.3.2.8-1  Sequence pattern setting commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>:SOURce:PATTern:SEQuence:PAIR?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SOURce:PATTern:SEQuence:CONDition:ADD?</td>
</tr>
<tr>
<td>[9]</td>
<td>Loop times</td>
<td>:SOURce:PATTern:SEQuence:LOOPtime</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SOURce:PATTern:SEQuence:LOOPtime?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SOURce:PATTern:SEQuence:CONDition:LOPC?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SOURce:PATTern:SEQuence:CONDition:EXTernal?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SOURce:PATTern:SEQuence:CONDition:MANual?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SOURce:PATTern:SEQuence:CONDition:AMATch?</td>
</tr>
</tbody>
</table>

Figure 7.3.2.8-3  Sequence pattern setting (Pattern Editor screen)
### Table 7.3.2.8-1  Sequence pattern setting commands (Cont'd)

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>:SOURce:PATTern:SEQUence:CONDition:BMATch?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SOURce:PATTern:SEQUence:LENGth?</td>
</tr>
<tr>
<td>[16]</td>
<td>No label (Pattern setting)</td>
<td>:SOURce:PATTern:SEQUence:DATA:WHOLe</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SOURce:PATTern:SEQUence:DATA:WHOLe?</td>
</tr>
<tr>
<td>[17]</td>
<td>No label (Pattern setting)</td>
<td>:SOURce:PATTern:SEQUence:BDATa:WHOLe</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SOURce:PATTern:SEQUence:BDATa:WHOLe?</td>
</tr>
<tr>
<td>[18]</td>
<td>No label (Data inversion)</td>
<td>:SOURce:PATTern:SEQUence:DREVerse:ADDRess</td>
</tr>
<tr>
<td>[19]</td>
<td>No label (Data inversion)</td>
<td>:SOURce:PATTern:SEQUence:DREVerse:DELTa</td>
</tr>
</tbody>
</table>

#### :SOURce:PATTern:SEQUence:PAIR <numeric>

- **Parameter**: <numeric> = <DECIMAL NUMERIC PROGRAM DATA>
  
  0 to 6 slots
  
  The maximum number of slots is 6 for the MP1800A, and 4 for the MT1810A.
  
- **Function**: Sets Slot for paired MU181040A and MU181040B at Sequence pattern occurrence.
  
- **Example**
  
  Set slot for paired MU181040A and MU181040B to 2
  
  ```
  > :SOURce:PATTern:SEQUence:PAIR 2
  ```
  
- **Compatibility**: Incompatible with existing models.

#### :SOURce:PATTern:SEQUence:PAIR?

- **Response**: <numeric> = <NR1 NUMERIC RESPONSE DATA>
  
  0 to 6 slots
  
- **Function**: Queries the slot of the MU181040A and MU181040B that is paired during Sequence pattern generation.
  
- **Example**
  
  ```
  > :SOURce:PATTern:SEQUence:PAIR?
  < 2
  ```
  
- **Compatibility**: Incompatible with existing models.

#### :SOURce:PATTern:SEQUence:MANual

- **Function**: Generates a Sequence pattern that is set manually.
  
- **Example**
  
  ```
  > :SOURce:PATTern:SEQUence:MANual
  ```
  
- **Compatibility**: Incompatible with existing models.
### :SOURce:PATTern:SEQuence:TRANsmit

**Function**
Restarts the Sequence pattern generation.

**Example**
>`:SOURce:PATTern:SEQuence:TRANsmit`

**Compatibility**
Incompatible with existing models.

### :SOURce:PATTern:SEQuence:CONDition:ADD <block>

**Parameter**
`<block> = <DECIMAL NUMERIC PROGRAM DATA>`
1 to 128 1 to 128 blocks, in 1-block steps

**Function**
Sets the number of blocks that are edited during Sequence pattern generation.

**Example**
To set the number of blocks that are edited to 10:
>`:SOURce:PATTern:SEQuence:CONDition:ADD 10`

**Compatibility**
Incompatible with existing models.

### :SOURce:PATTern:SEQuence:CONDition:ADD?

**Parameter**
`<block> = <NR1 NUMERIC RESPONSE DATA>`
1 to 128 1 to 128 blocks, in 1-block steps

**Function**
Queries the number of blocks that are edited during Sequence pattern generation.

**Example**
>`:SOURce:PATTern:SEQuence:CONDition:ADD?`
< 10

**Compatibility**
Incompatible with existing models.

### :SOURce:PATTern:SEQuence:COPY <block>

**Parameter**
`<block> = <DECIMAL NUMERIC PROGRAM DATA>`
1 to 128 1 to 128 blocks, in 1-block steps

**Function**
Copies the settings in the specified block during Sequence pattern generation.

**Example**
To copy the settings in block 12:
>`:SOURce:PATTern:SEQuence:COPY 12`

**Compatibility**
Incompatible with existing models.
Chapter 7  SCPI Commands

:SOURce:PATTern:SEQuence:CUT <block>

Parameter  

  <block> = <DECIMAL NUMERIC PROGRAM DATA>
  1 to 128  1 to 128 blocks, in 1-block steps

Function  

  Cuts (copies and removes) the settings in the specified block during Sequence pattern generation.

Example  

  To cut the settings in block 15:
  > :SOURce:PATTern:SEQuence:CUT 15

Compatibility  

  Incompatible with existing models.

:SOURce:PATTern:SEQuence:PASTe <block>

Parameter  

  <block> = <DECIMAL NUMERIC PROGRAM DATA>
  1 to 128  1 to 128 blocks, in 1-block steps

Function  

  Pastes the settings to the specified block during Sequence pattern generation. A block is added to the specified block number.

Example  

  To paste the settings to block 20:
  > :SOURce:PATTern:SEQuence:PASTe 20

Compatibility  

  Incompatible with existing models.

:SOURce:PATTern:SEQuence:CLEar

Function  

  Clears all the setting conditions during Sequence pattern generation.

Example  

  > :SOURce:PATTern:SEQuence:CLEar

Compatibility  

  Incompatible with existing models.

:SOURce:PATTern:SEQuence:LOOPtime <block>, <type>[,<numeric>]

Parameter  

  <block> = <DECIMAL NUMERIC PROGRAM DATA>
  1 to 128  1 to 128 blocks, in 1-block steps
  <type> = <CHARACTER PROGRAM DATA>
  COUNt  Specifies the number of repetition times
  REPeat  Does not specify the number of repetition times.
  [<numeric>] = <DECIMAL NUMERIC PROGRAM DATA>
  1 to 1024  1 to 1024 loops, in 1-loop steps

Note:  

  <numeric> can be omitted when <type> is set to REPeat. Also, the setting in <numeric> is ignored when <type> is set to REPeat.

Function  

  Sets the number of loops for the specified block during Sequence pattern generation.

Example  

  To set the number of loops for block 1 to 150:
  > :SOURce:PATTern:SEQuence:LOOPtime 1COUNt,150

Compatibility  

  Incompatible with existing models.
7.3 PPG Commands

:SOURce:PATTern:SEQuence:LOOPtime? <block>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;block&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1 to 128 1 to 128 blocks, in 1-block steps</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;type&gt; = &lt;CHARACTER RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>COUN, REP</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>&lt;numeric&gt;</th>
<th>= &lt;NR1 NUMERIC RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 to 1024</td>
<td>1 to 1024 loops, in 1-loop steps</td>
</tr>
</tbody>
</table>

**Note:**
When <type> is set to REPeat, <numeric> is omitted.

**Function**
Queries the number of loops for the specified block during Sequence pattern generation.

**Example**
To query the number of loops for block 1:
> :SOURce:PATTern:SEQuence:LOOPtime? 1
< COUN,150

**Compatibility**
Incompatible with existing models.

:SOURce:PATTern:SEQuence:CONDition:LOPC <block>,<loop>[,<destination>]

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;block&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt; (Block No)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1 to 128 1 to 128 blocks, in 1-block steps</td>
</tr>
</tbody>
</table>

| <loop> = <CHARACTER PROGRAM DATA> (Loop Completion) |
|---------|-------------------------------------------------|
| STOP, JUMP, NEXT |

<table>
<thead>
<tr>
<th>&lt;destination&gt;</th>
<th>= &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 to 128</td>
<td>1 to 128 blocks, in 1-block steps</td>
</tr>
</tbody>
</table>

When selecting JUMP, specify the jump destination block number n from 1 to 128. [<destination>] can be omitted when JUMP is not selected.

**Function**
Sets the loop completion condition of each block during Sequence pattern generation.

**Example**
To set NEXT for the loop completion condition of block 3:
> :SOURce:PATTern:SEQuence:CONDition:LOPC 3,NEXT

**Compatibility**
Incompatible with existing models.
Chapter 7  SCPI Commands


Parameter  
<block> = <DECIMAL NUMERIC PROGRAM DATA> (Block No)  
1 to 128  1 to 128 blocks, in 1-block steps

Response  
<loop> = <CHARACTER RESPONSE DATA> (Loop Completion)  
STOP, JUMP, NEXT

[<destination>] = <NR1 NUMERIC RESPONSE DATA>  
1 to 128  1 to 128 blocks, in 1-block steps

When JUMP is selected, the jump destination block number n from 1 to 128 is returned. Omitted when Jump is not specified.

Function  
Queries the loop completion condition of each block during Sequence pattern generation.

Example  
To query the loop completion condition of block 3:  
< NEXT

Compatibility  
Incompatible with existing models.

:SOURce:PATTern:SEQUence:CONDition:EXTERNAL
<brack>,<ext>[,<destination>]

Parameter  
<block> = <DECIMAL NUMERIC PROGRAM DATA> (Block No)  
1 to 128  1 to 128 blocks, in 1-block steps

<ext> = <CHARACTER PROGRAM DATA> (Ext Trigger)  
STOP, JUMP, NEXT, NONE

[<destination>] = <DECIMAL NUMERIC PROGRAM DATA>  
1 to 128  1 to 128 blocks, in 1-block steps

When selecting JUMP, specify the jump destination block number n from 1 to 128. [<destination>] can be omitted when JUMP is not selected.

Function  
Sets the external trigger condition of each block during Sequence pattern generation.

Example  
To set STOP for the external trigger condition of block 5:  
>:SOURce:PATTern:SEQUence:CONDition:EXTERNAL 5,STOP

Compatibility  
Incompatible with existing models.
### :SOURce:PATTern:SEQuence:CONDition:EXTernal? <block>

**Parameter**
- `<block>` = `<DECIMAL NUMERIC PROGRAM DATA>` (Block No)
  - 1 to 128
  - 1 to 128 blocks, in 1-block steps

**Response**
- `<ext>` = `<CHARACTER RESPONSE DATA>` (Ext Trigger)
  - STOP, JUMP, NEXT, NONE
- `<destination>` = `<NR1 NUMERIC RESPONSE DATA>`
  - 1 to 128
  - 1 to 128 blocks, in 1-block steps

When JUMP is selected, the jump destination block number n from 1 to 128 is returned. Omitted when Jump is not specified.

**Function**
Queries the external trigger condition of each block during Sequence pattern generation.

**Example**
To query the external trigger condition of block 5:

```
< STOP
```

**Compatibility**
Incompatible with existing models.

### :SOURce:PATTern:SEQuence:CONDition:MANual <block>,<man>[,<destination>]

**Parameter**
- `<block>` = `<DECIMAL NUMERIC PROGRAM DATA>` (Block No)
  - 1 to 128
  - 1 to 128 blocks, in 1-block steps
- `<man>` = `<CHARACTER PROGRAM DATA>` (Manual Trigger)
  - STOP, JUMP, NEXT, NONE
- `<destination>` = `<DECIMAL NUMERIC PROGRAM DATA>`
  - 1 to 128
  - 1 to 128 blocks, in 1-block steps

When selecting JUMP, specify the jump destination block number n from 1 to 128. `[<destination>]` can be omitted when JUMP is not selected.

**Function**
Sets the manual trigger condition of each block during Sequence pattern generation.

**Example**
To set the manual trigger condition of block 2 to "Jump to block 5".

```
> :SOURce:PATTern:SEQuence:CONDition:MANual 2,JUMP5
```

**Compatibility**
Incompatible with existing models.
### :SOURce:PATTern:SEQuence:CONDition:MANual? <block>

**Parameter**
- `<block>` = `<DECIMAL NUMERIC PROGRAM DATA>` (Block No)
  - 1 to 128  1 to 128 blocks, in 1-block steps

**Response**
- `<man>` = `<CHARACTER RESPONSE DATA>` (Manual Trigger)
  - STOP, JUMP, NEXT, NONE
- `<destination>` = `<NR1 NUMERIC RESPONSE DATA>`
  - 1 to 128  1 to 128 blocks, in 1-block steps

When JUMP is selected, the jump destination block number n from 1 to 128 is returned. Omitted when Jump is not specified.

**Function**
Queries the manual trigger condition of each block during Sequence pattern generation.

**Example**
To query the manual trigger condition of block 2:

```plaintext
< JUMP5
```

**Compatibility**
Incompatible with existing models.

### :SOURce:PATTern:SEQuence:CONDition:AMATch <block>,<a>[,<destination>]

**Parameter**
- `<block>` = `<DECIMAL NUMERIC PROGRAM DATA>` (Block No)
  - 1 to 128  1 to 128 blocks, in 1-block steps
- `<a>` = `<CHARACTER PROGRAM DATA>` (A Pattern Match)
  - STOP, JUMP, NEXT, NONE
- `<destination>` = `<DECIMAL NUMERIC PROGRAM DATA>`
  - 1 to 128  1 to 128 blocks, in 1-block steps

When selecting JUMP, specify the jump destination block number n from 1 to 128. `<destination>` can be omitted when JUMP is not selected.

**Function**
Sets the pattern A match condition of each block during Sequence pattern generation.

**Example**
To set NONE for the pattern A match condition of block 3:

```plaintext
> :SOURce:PATTern:SEQuence:CONDition:AMATch 3,NONE
```

**Compatibility**
Incompatible with existing models.
7.3 PPG Commands


Parameter
<brace> = <DECIMAL NUMERIC PROGRAM DATA> (Block No)
1 to 128 1 to 128 blocks, in 1-block steps

Response
<brace> = <CHARACTER RESPONSE DATA> (A Pattern Match)
STOP, JUMP, NEXT, NONE
[destination] = <DECIMAL NUMERIC RESPONSE DATA>
1 to 128 1 to 128 blocks, in 1-block steps
When JUMP is selected, the jump destination block number n from 1 to 128 is returned. Omitted when Jump is not specified.

Function
Queries the pattern A match condition of each block during Sequence pattern generation.

Example
To query the pattern A match condition of block 3:
< NONE

Compatibility
Incompatible with existing models.

:SOURce:PATTern:SEQuence:CONDition:BMATch <block>,<b>[,destination]

Parameter
<brace> = <DECIMAL NUMERIC PROGRAM DATA> (Block No)
1 to 128 1 to 128 blocks, in 1-block steps
<brace> = <CHARACTER PROGRAM DATA> (B Pattern Match)
STOP, JUMP, NEXT, NONE
[destination] = <DECIMAL NUMERIC PROGRAM DATA>
1 to 128 1 to 128 blocks, in 1-block steps
When selecting JUMP, specify the jump destination block number n from 1 to 128. [destination] can be omitted when JUMP is not selected.

Function
Sets the pattern B match condition of each block during Sequence pattern generation.

Example
To set NEXT for the pattern B match condition of block 7:
> :SOURce:PATTern:SEQuence:CONDition:BMATch 7,NEXT

Compatibility
Incompatible with existing models.
Chapter 7  SCPI Commands


Parameter
<block> = <DECIMAL NUMERIC PROGRAM DATA> (Block No)
1 to 128  1 to 128 blocks, in 1-block steps

Response
\(<b> = <CHARACTER RESPONSE DATA> (B Pattern Match)
STOP, JUMP, NEXT, NONE
[<destination>] = <NR1 NUMERIC RESPONSE DATA>
1 to 128  1 to 128 blocks, in 1-block steps
When JUMP is selected, the jump destination block number n from 1 to
128 is returned. Omitted when Jump is not specified.

Function
Queries the pattern B match condition of each block during Sequence
pattern generation.

Example
To query the pattern B match condition of block 7:
< NEXT

Compatibility
Incompatible with existing models.

:SOURce:PATTern:SEQuence:LENGth <block>,<numeric>

Parameter
<block> = <DECIMAL NUMERIC PROGRAM DATA>
1 to 128  1 to 128 blocks, in 1-block steps
<numeric> = <DECIMAL NUMERIC PROGRAM DATA>
8192 to 1048576 8192 to 1,048,576 bits, in 128-bit steps
Be sure to set a value so that the following condition is satisfied.
Number of blocks × Block length ≤ 128 Mbits

Function
Sets the pattern length of the specified block during Sequence pattern
generation.

Example
To set the pattern length of block 3 to 8192 bits:
> :SOURce:PATTern:SEQuence:LENGth 3,8192

Compatibility
Incompatible with existing models.

:SOURce:PATTern:SEQuence:LENGth? <block>

Parameter
<block> = <DECIMAL NUMERIC PROGRAM DATA>
1 to 128  1 to 128 blocks, in 1-block steps

Response
<numeric> = <NR1 NUMERIC RESPONSE DATA>
8192 to 1048576 8192 to 1,048,576 bits, in 128-bit steps

Function
Queries the pattern length of the specified block during Sequence pattern
generation.

Example
To query the pattern length of block 3:
> :SOURce:PATTern:SEQuence:LENGth? 3
< 8192

Compatibility
Incompatible with existing models.
7.3 PPG Commands

:SOURce:PATTern:SEQuence:DATA:WHOLe
<block>,<start>,<end>,<data>

Parameter

- `<block>` = <DECIMAL NUMERIC PROGRAM DATA>
  1 to 128 blocks, in 1-block steps
- `<start>` = <NON-DECIMAL PROGRAM DATA>
  #H0 to #HFFFFF 0 to FFFFF bits, in 1-bit steps
  (Specify in hexadecimal)
- `<end>` = <NON-DECIMAL PROGRAM DATA>
  #H0 to #HFFFFF 0 to FFFFF bits, in 1-bit steps
  (Specify in hexadecimal)

Note:
The maximum setting value is the pattern length.

- `<data>` = <STRING PROGRAM DATA>

  "H***" Specify pattern data in hexadecimal format.
  1 to 400 characters (pattern data of 400 × 4 bits)
  Specify a character string consisting of 0 to 9
  and A to F.
  If the last character string of the pattern data is
  short of a hexadecimal unit (4 bits), add 0(s) up
  to the last bit to make a hexadecimal string.

  "B***" Specify pattern data in binary format.
  1 to 400 characters (pattern data of 400 bits)
  Specify a character string consisting of 0 and 1.

Function
Sets the pattern data from the `<start>` to `<end>` addresses during
Sequence pattern generation.
The set data overwrites the specified range.

Example
To set the addresses 0 to FF of the block 1 data to 5A:
> :SOURce:PATTern:SEQuence:DATA:WHOLe 1,#H0,#HFF, "H5A"

Compatibility
Incompatible with existing models.
Chapter 7  SCPI Commands

:SOURce:PATTern:SEQuence:DATA:WHOLe? <block>,<start>

Parameter

- `<block>` = <DECIMAL NUMERIC PROGRAM DATA>
  1 to 128 1 to 128 blocks, in 1-block steps
- `<start>` = <NON-DECIMAL PROGRAM DATA>
  #H0 to #HFFFFF 0 to FFFFFF bits, in 1-bit steps
  (Specify in hexadecimal)

**Note:**
The maximum setting value is the pattern length.

Response

- `<data>` = <STRING RESPONSE DATA>
  "H***" Returns the pattern data in hexadecimal format.
  Within 400 characters (pattern data of 400 × 4 bits)

Function

Queries the pattern data of 400 × 4 bits from the `<start>` address during Sequence pattern generation.

Example

To query the block 1 data from address 0:
> :SOURce:PATTern:SEQuence:DATA:WHOLe? 1,#H0,
< "H5A"

Compatibility

Incompatible with existing models.

:SOURce:PATTern:SEQuence:BDATa:WHOLe <block>,<start>,<end>,<bdata>

Parameter

- `<block>` = <DECIMAL NUMERIC PROGRAM DATA>
  1 to 128 1 to 128 blocks, in 1-block steps
- `<start>` = <NON-DECIMAL PROGRAM DATA>
  #H0 to #HFFFFF 0 to FFFFFF bits, in 1-bit steps
  (Specify in hexadecimal)
- `<end>` = <NON-DECIMAL PROGRAM DATA>
  #H0 to #HFFFFF 0 to FFFFFF bits, in 1-bit steps
  (Specify in hexadecimal)

**Note:**
The maximum setting value is the pattern length.

- `<bdata>` = <ARBITRARY BLOCK PROGRAM DATA>
  #XYYY<binary> X: Number of YYY digits
  YYY: Number of bytes of <binary>
  1 to 131,072 bytes
  <binary>: Binary data up to 131,072 bytes

Function

Sets binary data of the pattern data from `<start>` to `<end>` addresses during Sequence pattern generation.
The set data overwrites the specified range.

Example

To set the addresses 0 to FF of the block 1 data to 44:
> :SOURce:PATTern:SEQuence:BDATa:WHOLe 1,#H0,#HFF,#11D

Compatibility

Incompatible with existing models.
7.3 PPG Commands

**:SOURce:PATTern:SEQuence:BDATa:WHOLe? <block>,<start>[,<size>]**

**Parameter**

- `<block>` = <DECIMAL NUMERIC PROGRAM DATA>
  1 to 128 1 to 128 blocks, in 1-block steps
- `<start>` = <NON-DECIMAL PROGRAM DATA>
  #H0 to #HFFFFF 0 to FFFFFF bits, in 1-bit steps
  (Specify in hexadecimal)

**Note:**
The maximum setting value is the pattern length.

- `[<size>]` = <NR1 NUMERIC PROGRAM DATA>
  1 to 131072 1 to 131,072 bytes, in 1-byte steps
  Binary data of the setting pattern is queried when [<size>] is omitted.

**Response**

- `<bdata>` = <DEFINITE LENGTH ARBITRARY BLOCK RESPONSE DATA>
  #XYYY<binary>
  X: Number of YYY digits
  YYY: Number of bytes of <binary>
  1 to 131,072 bytes
  <binary>: Binary data up to 131,072 bytes

**Function**
Queries binary data of bytes specified by `<size>` from `<start>` address of the pattern data during Sequence pattern generation.

**Example**
To query binary data of the block 1 data starting from address 0:
> :SOURce:PATTern:SEQuence:BDATa:WHOLe? 1,#H0
< #11D

**Compatibility**
Incompatible with existing models.

**:SOURce:PATTern:SEQuence:DREVerse:ADDRess <block>,<start>,<end>**

**Parameter**

- `<block>` = <DECIMAL NUMERIC PROGRAM DATA>
  1 to 128 1 to 128 blocks, in 1-block steps
- `<start>` = <NON-DECIMAL PROGRAM DATA>
  #H0 to #HFFFFF 0 to FFFFFF bits, in 1-bit steps
  (Specify in hexadecimal)
- `<end>` = <NON-DECIMAL PROGRAM DATA>
  #H0 to #HFFFFF 0 to FFFFFF bits, in 1-bit steps
  (Specify in hexadecimal)

**Note:**
The maximum setting value is the pattern length.

**Function**
Inverts the data in the program during Sequence pattern generation. Specify the inversion range by the `<start>` and `<end>` addresses.

**Example**
To invert addresses 0 to FF of the block 3 data:
> :SOURce:PATTern:SEQuence:DREVerse:ADDRess 3,#H0,#HFF

**Compatibility**
Incompatible with existing models.
### :SOURce:PATTern:SEQuence:DREVerse:DELTa \(<block>,<start>,<delta>\)

**Parameter**

\(<block> = <DECIMAL NUMERIC PROGRAM DATA> \\
1 to 128 1 to 128 blocks, in 1-block steps \\
\(<start> = <NON-DECIMAL PROGRAM DATA> \\
#H0 to #HFFFFF 0 to FFFFF bits, in 1-bit steps \\
(Specify in hexadecimal) \\
\(<delta> = <DECIMAL NUMERIC PROGRAM DATA> \\
1 to 1048576 1 to 1,048,576 bits, in 1-bit steps \\
**Note:**

The maximum setting value is the pattern length.

**Function**

Inverts the data in the program during Sequence pattern generation.

Specify the inversion range by the number of bits from the \(<start> \) address (relative specification).

**Example**

To invert 128 bits of the block 5 data from address 0:

\[ > :\text{SOURce}:\text{PATTern}:\text{SEQuence}:\text{DREVerse}:\text{DELTa} \ 5,#H0,128 \]

**Compatibility**

Incompatible with existing models.

---

### :SOURce:PATTern:SEQuence:DATA:FILL \(<block>,<range>,<page>,<data>\)

**Parameter**

\(<block> = <DECIMAL NUMERIC PROGRAM DATA> \\
1 to 128 1 to 128 blocks, in 1-block steps \\
\(<range> = <CHARACTER PROGRAM DATA> \\
PAGE Specifies a page. \\
(One page is defined as 128 bits.) \\
ALL Specifies all data. \\
\(<page> = <DECIMAL NUMERIC PROGRAM DATA> \\
1 to 8192 1 to 8,192 pages, in 1-page steps \\
Max = \frac{\text{Pattern Length}}{128} (rounding up fractions) \\
Specify "0" when \(<range> \) is set to ALL. \\
\(<data> = <DECIMAL NUMERIC PROGRAM DATA> \\
0 Fills the specified range with 0s. \\
1 Fills the specified range with 1s. \\
**Function**

Sets 0s or 1s for the specified page or all data in the program during Sequence pattern generation.

**Example**

To set 1s for all the data of the block 8:

\[ > :\text{SOURce}:\text{PATTern}:\text{SEQuence}:\text{DATA}:\text{FILL} \ 8,\text{ALL},0,1 \]

**Compatibility**

Incompatible with existing models.
7.3.3 Commands related to Error Addition tab

Figure 7.3.3-1 Error Addition tab
(For PRBS, Zero-Substitution, Data patterns)

Figure 7.3.3-2 Error Addition tab (For Alternate pattern)
Figure 7.3.3-3  Error Addition tab (For Mixed pattern)
7.3 PPG Commands

Figure 7.3.3-4  Error Addition tab (For Sequence pattern)

Table 7.3.3-1  Error Addition setting commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>:SOURce:PATTern:EADDition:SET?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SOURce:PATTern:EADDition:SOURce?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SOURce:PATTern:EADDition:SELect?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SOURce:PATTern:EADDition:ROUTe?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SOURce:PATTern:EADDition:RATE?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SOURce:PATTern:EADDition:VARiation?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SOURce:PATTern:EADDition:AREA?</td>
</tr>
<tr>
<td>[9]</td>
<td>Set All/Reset All</td>
<td>:SOURce:PATTern:EADDition:ALLSet</td>
</tr>
</tbody>
</table>
## :SOURce:PATTern:EADDition:SET <boolean>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;boolean&gt; = &lt;BOOLEAN PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>OFF or 0 Output OFF</td>
</tr>
<tr>
<td></td>
<td>ON or 1 Output ON</td>
</tr>
</tbody>
</table>

**Function**
Sets whether to add a bit error to the test pattern.

**Example**
To set bit error addition ON:

```
> :SOURce:PATTern:EADDition:SET ON
```

**Compatibility**
Compatible with the MP1632C.

## :SOURce:PATTern:EADDition:SET?

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;numeric&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0 Error addition OFF</td>
</tr>
<tr>
<td></td>
<td>1 Error addition ON</td>
</tr>
</tbody>
</table>

**Function**
Queries the bit error addition ON/OFF state to the test pattern.

**Example**

```
> :SOURce:PATTern:EADDition:SET?
< 1
```

**Compatibility**
Compatible with the MP1632C.

## :SOURce:PATTern:EADDition:SOURce <source>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;source&gt; = &lt;CHARACTER PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>INTernal Internal signal</td>
</tr>
<tr>
<td></td>
<td>EXTTrig Rising of external signal trigger</td>
</tr>
<tr>
<td></td>
<td>EXTDisable Disables external signal</td>
</tr>
</tbody>
</table>

**Function**
Sets the reference signal source for bit error addition to the test pattern.

**Example**
To set the reference signal source for bit error addition to Internal:

```
> :SOURce:PATTern:EADDition:SOURce INTernal
```

**Compatibility**
Incompatible with existing models.

## :SOURce:PATTern:EADDition:SOURce?

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;source&gt; = &lt;CHARACTER RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>INT, EXT, EXTD</td>
</tr>
</tbody>
</table>

**Function**
Queries the reference signal source for bit error addition to the test pattern.

**Example**

```
> :SOURce:PATTern:EADDition:SOURce?
< INT
```

**Compatibility**
Incompatible with existing models.
### :SOURce:PATTern:EADDition:SELect <select>

**Parameter**

<table>
<thead>
<tr>
<th>&lt;select&gt;</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SCAN</td>
<td>Changes the route for which 32 Demux is performed every time when an error is added.</td>
</tr>
<tr>
<td>SELect</td>
<td>Adds an error to the specified route.</td>
</tr>
</tbody>
</table>

**Function**

Sets the route setting method for bit error addition.

**Example**

To set the route setting method to SCAN:

```
> :SOURce:PATTern:EADDition:SELect SCAN
```

**Compatibility**

Incompatible with existing models.

### :SOURce:PATTern:EADDition:SELect?

**Response**

<table>
<thead>
<tr>
<th>&lt;select&gt;</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SCAN</td>
<td></td>
</tr>
<tr>
<td>SEL</td>
<td></td>
</tr>
</tbody>
</table>

**Function**

Queries the route setting method for bit error addition.

**Example**

```
> :SOURce:PATTern:EADDition:SELect?
< SCAN
```

**Compatibility**

Incompatible with existing models.

### :SOURce:PATTern:EADDition:ROUTe <numeric>

**Parameter**

<table>
<thead>
<tr>
<th>&lt;numeric&gt;</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 to 32</td>
<td>1 to 32, in single steps</td>
</tr>
</tbody>
</table>

**Function**

Sets the route to add a bit error.

**Example**

To set Route 2 as the route to add a bit error:

```
> :SOURce:PATTern:EADDition:ROUTe 2
```

**Compatibility**

Compatible with the MP1632C (1 to 8, in single steps).

### :SOURce:PATTern:EADDition:ROUTe?

**Response**

<table>
<thead>
<tr>
<th>&lt;numeric&gt;</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 to 32</td>
<td>1 to 32, in single steps</td>
</tr>
</tbody>
</table>

**Function**

Queries the route to add a bit error.

**Example**

```
> :SOURce:PATTern:EADDition:ROUTe?
< 2
```

**Compatibility**

Compatible with the MP1632C (1 to 8, in single steps).
Chapter 7 SCPI Commands

:SOURce:PATTern:EADDition:RATE <rate>[,<numeric>]

Parameter
- `<rate>` = `<CHARACTER PROGRAM DATA>`
  - E_2 to E_12: E_2 to E_12, in E_1 steps
  - In Combination (for 2 and 4 channels)
  - E_3 to E_12: E_3 to E_12, in E_1 steps
- `<numeric>` = `<DECIMAL NUMERIC PROGRAM DATA>`
  - x: 1 to 9: x: 1 to 9, in single steps
  - Set a value in the format of "xE–n". If x is omitted, 1 is set.

Function
Sets the rate of bit errors to be added.

Example
To set the rate of bit errors to be added to 5E–9:
> :SOURce:PATTern:EADDition:RATE E_9,5

Compatibility
Partially compatible with the MP1632C (only when x is omitted).

:SOURce:PATTern:EADDition:RATE?

Response
- `<rate>` = `<CHARACTER RESPONSE DATA>`
  - E_2 to E_12: E_2 to E_12
  - In Combination (for 2 and 4 channels)
  - E_3 to E_12: E_3 to E_12
- `<numeric>` = `<NR2 NUMERIC RESPONSE DATA>`
  - x: 1 to 9: x: 1 to 9, in single steps

Function
Queries the rate of bit errors to be added.

Example
> :SOURce:PATTern:EADDition:RATE?
< E_9,5

Compatibility
Partially compatible with the MP1632C (only when x is omitted).

:SOURce:PATTern:EADDition:VARiation <var>

Parameter
- `<var>` = `<CHARACTER PROGRAM DATA>`
  - REPeat: Errors are continuously inserted.
  - SINGle: An error is inserted by each click.

Function
Sets the bit error addition method for the test pattern (when Internal is selected).

Example
To set the bit error addition method to Repeat:
> :SOURce:PATTern:EADDition:VARiation REPeat

Compatibility
Incompatible with existing models.
7.3 PPG Commands

:SOURce:PATTern:EADDition:VARiation?

Response

<var> = <CHARACTER RESPONSE DATA>
REP, SING

Function
Queries the bit error addition method (when Internal is selected) to the test pattern.

Example
> :SOURce:PATTern:EADDition:VARiation?
< REP

Compatibility
Incompatible with existing models.

:SOURce:PATTern:EADDition:SINGle

Function
Adds a single error to the test pattern.

Example
> :SOURce:PATTern:EADDition:SINGle

Compatibility
Compatible with the MP1632C.

:SOURce:PATTern:EADDition:AREA <contents>,<row>,<data>[,<prbs>]

Parameter
<contents> = <CHARACTER PROGRAM DATA>
APAT  Pattern A
BPAT  Pattern B
ABPat  Patterns A and B
NONE  (For Mixed Data and Sequence patterns)

<row> = <DECIMAL NUMERIC PROGRAM DATA>
1 to 16  (Specify the row number for Mixed Data patterns)
1 to 128 (Specify the block number for Sequence patterns)
0       (Other than the above patterns)

<data> = <BOOLEAN PROGRAM DATA>
OFF or 0, ON or 1
OFF or 0 for Alternate patterns

[<prbs>] = <BOOLEAN PROGRAM DATA>
OFF or 0, ON or 1 (For Mixed patterns only)
OFF or 0 for Alternate and Sequence patterns

Function
Sets an area to add a bit error for Alternate, Mixed, and Sequence patterns.

Example
To set an area to add a bit error to "Pattern A, 1 Row, Data ON, PRBS ON":
> :SOURce:PATTern:EADDition:AREA APAT,1,1,1

Compatibility
Incompatible with existing models.
Chapter 7  SCPI Commands

:SOURce:PATTern:EADDition:AREA? <contents>,<row>

Parameter

<contents> = <CHARACTER PROGRAM DATA>
APAT  Pattern A
BPAT  Pattern B
ABPat  Patterns A and B
NONE  (For Mixed Data and Sequence patterns)

<row> = <DECIMAL NUMERIC PROGRAM DATA>
1 to 16  (Specify the row number for Mixed Data patterns)
1 to 128 (Specify the block number for Sequence patterns)
0  (For Alternate patterns)

Response

<data>,<prbs> = <NR1 NUMERIC RESPONSE DATA>
0  OFF
1  ON

<prbs> is returned for Mixed patterns only.

Function
Queries an area to add a bit error.

Example
To query an error addition area of Pattern A, 1 Row:
> :SOURce:PATTern:EADDition:AREA? APAT,1
< 1,1

To query an error addition area for Alternate pattern:
> :SOURce:PATTern:EADDition:AREA? APAT,0
< 1

Compatibility
Incompatible with existing models.

:SOURce:PATTern:EADDition:ALLSet <numeric>

Parameter

<numERIC> = <NUMERIC PROGRAM DATA>
0  All Reset
1  All Set

Function
Selects (set) or unselects (reset) all areas to add a bit error for Mixed and Sequence patterns.

Example
To select all areas to add a bit error:
> :SOURce:PATTern:EADDition:ALLSet 1

Compatibility
Incompatible with existing models.
7.3.4 Commands related to Misc tab

Figure 7.3.4-1 Misc tab
7.3.4.1 Pattern Generation setting commands

Figure 7.3.4.1-1 Pattern Generation setting field (for Repeat signal generation)

Figure 7.3.4.1-2 Pattern Generation setting field (for Repeat signal generation)
### Table 7.3.4.1-1 Pattern Generation setting commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1]</td>
<td>Pattern Sequence</td>
<td>:SOURce:PATTern:OMODE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SOURce:PATTern:OMODE?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SOURce:PATTern:BURSt:SEQUence?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SOURce:PATTern:BURSt:MODE?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SOURce:PATTern:REPeat:PULSewidth?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SOURce:PATTern:REPeat:TRIGdelay?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SOURce:PATTern:BURSt:ELENgth?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SOURce:PATTern:BURSt:CYCLE?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SOURce:PATTern:BURSt:TRIGdelay?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SOURce:PATTern:BURSt:PULSewidth?</td>
</tr>
</tbody>
</table>
Chapter 7  SCPI Commands

:SOURce:PATTern:OMODe <mode>
Parameter  
<mode> = <CHARACTER PROGRAM DATA>
REPeat  Repeat signal
BURSt  Burst signal
Function  
Sets the signal generation method from Repeat (consecutive) signal or Burst signal.
Example  
To set the signal generation method to Burst signal:
> :SOURce:PATTern:OMODe BURSt
Compatibility  
Compatible with the MP1632C.

:SOURce:PATTern:OMODe?
Response  
<mode> = <CHARACTER RESPONSE DATA>
REP, BURS
Function  
Queries the signal generation method.
Example  
> :SOURce:PATTern:OMODe?
< BURS
Compatibility  
Compatible with the MP1632C.

:SOURce:PATTern:BURSt:SEQuence <mode>
Parameter  
<mode> = <CHARACTER PROGRAM DATA>
RESTart  Restart from the beginning
CONSecutive  Consecutive output in the Burst interval
CONTinuous  Consecutive output regardless of Burst interval
Function  
Sets the data output sequence for the Burst data signal.
Example  
To set the output sequence to Restart:
> :SOURce:PATTern:BURSt:SEQuence RESTart
Compatibility  
Incompatible with existing models.

:SOURce:PATTern:BURSt:SEQuence?
Response  
<mode> = <CHARACTER RESPONSE DATA>
REST, CONS, CONT
Function  
Queries the data output sequence for the Burst data signal.
Example  
> :SOURce:PATTern:BURSt:SEQuence?
< REST
Compatibility  
Incompatible with existing models.
### :SOURce:PA TTern:BURSt:MODE <mode>

**Parameter**

<mode> = <CHARACTER PROGRAM DATA>

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>INTernal</td>
<td>Generates based on internal reference signal</td>
</tr>
<tr>
<td>EXTTrig</td>
<td>Generates based on external signal trigger edge</td>
</tr>
<tr>
<td>EXTernal</td>
<td>Generates during the high level interval of</td>
</tr>
<tr>
<td></td>
<td>external signal trigger input</td>
</tr>
</tbody>
</table>

**Function**

Sets the Burst signal generation sequence from internal trigger, external trigger, or enable.

**Example**

To set the Burst signal generation sequence to Internal:

> :SOURce:PA TTern:BURSt:MODE INTernal

**Compatibility**

Compatible with the MP1632C.

### :SOURce:PA TTern:BURSt:MODE?

**Response**

<mode> = <CHARACTER RESPONSE DATA>

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>INT</td>
<td></td>
</tr>
<tr>
<td>EXTT</td>
<td></td>
</tr>
<tr>
<td>EXT</td>
<td></td>
</tr>
</tbody>
</table>

**Function**

Queries the Burst signal generation sequence.

**Example**

> :SOURce:PA TTern:BURSt:MODE?

< INT

**Compatibility**

Compatible with the MP1632C.

### :SOURce:PA TTern:REPeat:PULSewidth <numeric>[,<contents>]

**Parameter**

<numeric> = <DECIMAL NUMERIC PROGRAM DATA>

0 to 68719476672 0 to 68,719,476,672 bits, in 16-bit steps

The setting range is from 0 to 64 if the pattern length is 64 or less.

**Note:**

At 2 ch Combination, the setting range and Step are doubled.

At 4 ch Combination, the setting range and Step are quadrupled.

[<contents>] = <CHARACTER PROGRAM DATA>

- APAT  Specify to indicate the pulse width of Alternate pattern A.
- BPAT  Specify to indicate the pulse width of Alternate pattern B.

**Note:**

For patterns other than the Alternate pattern, [<contents>] cannot be set and should be omitted.

**Function**

Sets the pulse width of the timing signal to be output during the timing signal period.

**Example**

To set the pulse width of the timing signal to 128 bits:

> :SOURce:PA TTern:REPeat:PULSewidth 128

To set the pulse width of the timing signal for Alternate pattern A to 256 bits:

> :SOURce:PA TTern:REPeat:PULSewidth 256,APAT

**Compatibility**

Incompatible with existing models.
Chapter 7  SCPI Commands


<table>
<thead>
<tr>
<th>Parameter</th>
<th>[&lt;contents&gt;] = &lt;CHARACTER PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>APAT</td>
<td>Specify to indicate the pulse width of Alternate pattern A.</td>
</tr>
<tr>
<td>BPAT</td>
<td>Specify to indicate the pulse width of Alternate pattern B.</td>
</tr>
</tbody>
</table>

**Note:**
For patterns other than the Alternate pattern, [<contents>] cannot be set and should be omitted.

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;numeric&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function</td>
<td>Queries the pulse width of the timing signal to be output during the timing signal period.</td>
</tr>
<tr>
<td>Example</td>
<td>To query the pulse width of the timing signal:</td>
</tr>
<tr>
<td></td>
<td>&gt; :SOURce:PATTern:REPeat:PULSewidth?</td>
</tr>
<tr>
<td></td>
<td>&lt; 128</td>
</tr>
<tr>
<td></td>
<td>To query the pulse width of the timing signal of Alternate pattern A:</td>
</tr>
<tr>
<td></td>
<td>&gt; :SOURce:PATTern:REPeat:PULSewidth? APAT</td>
</tr>
<tr>
<td></td>
<td>&lt; 256</td>
</tr>
</tbody>
</table>

| Compatibility | Incompatible with existing models. |

:SOURce:PATTern:REPeat:TRIGdelay <numeric>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;numeric&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0 to 68719476672  0 to 68,719,476,672 bits, in 16-bit steps</td>
</tr>
<tr>
<td></td>
<td>The settable value is 0 if the pattern length is 79 bits or less.</td>
</tr>
</tbody>
</table>

**Note:**
At 2 ch Combination, the setting range and Step are doubled.
At 4 ch Combination, the setting range and Step are quadrupled.

<table>
<thead>
<tr>
<th>Function</th>
<th>Sets the delay value for the timing signal pulse to be output during the timing signal period.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example</td>
<td>To set the timing signal pulse delay to 256 bits:</td>
</tr>
<tr>
<td></td>
<td>&gt; :SOURce:PATTern:REPeat:TRIGdelay 256</td>
</tr>
</tbody>
</table>

| Compatibility | Incompatible with existing models. |

:SOURce:PATTern:REPeat:TRIGdelay?

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;numeric&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function</td>
<td>Queries the delay value for the timing signal pulse to be output during the timing signal period.</td>
</tr>
<tr>
<td>Example</td>
<td>&gt; :SOURce:PATTern:REPeat:TRIGdelay?</td>
</tr>
<tr>
<td></td>
<td>&lt; 256</td>
</tr>
</tbody>
</table>

| Compatibility | Incompatible with existing models. |
### :SOURce:PATTern:BURSt:ELENgth <numeric>

**Parameter**

<numeric> = <DECIMAL NUMERIC PROGRAM DATA>

- 640 to 2147483648
- 640 to 2,147,483,648 bits, in 128-bit steps

**Note:**

In the case of Internal, a Disable period of at least 512 bits is required between Burst Cycle and Enable Period.

In the case of Combination, the setting range, resolution, steps, and Disable period are multiplied by 4.

The setting range, Step, and Disable period at Combination are doubled at 2 ch Combination and quadrupled at 4 ch Combination.

**Function**

Sets the data signal generation interval for Burst signal generation.

**Example**

To set the data signal generation interval to 12,800 bits:

```
> :SOURce:PATTern:BURSt:ELENgth 12800
```

**Compatibility**

Incompatible with existing models.

---

### :SOURce:PATTern:BURSt:ELENgth? 

**Response**

<numeric> = <NR1 NUMERIC RESPONSE DATA>

- 640 to 2147483648
- 640 to 2,147,483,648 bits

**Function**

Queries the data signal generation interval for Burst signal generation.

**Example**

```
> :SOURce:PATTern:BURSt:ELENgth?
< 12800
```

**Compatibility**

Incompatible with existing models.

---

### :SOURce:PATTern:BURSt:CYCLe <numeric>

**Parameter**

<numeric> = <DECIMAL NUMERIC PROGRAM DATA>

- 1280 to 2147483648
- 1,280 to 2,147,483,648 bits, in 128-bit steps

**Note:**

At 2 ch Combination, the setting range and Step are doubled.

At 4 ch Combination, the setting range and Step are quadrupled.

**Function**

Sets an addition value of the data signal generation interval and the data stop interval for Burst signal generation (Burst generation cycle).

**Example**

To set the Burst generation cycle to 25,600 bits:

```
> :SOURce:PATTern:BURSt:CYCLe 25600
```

**Compatibility**

Incompatible with existing models.
Chapter 7  SCPI Commands

:SOURce:PATTern:BURSt:CYCle?

Response

Response

1280 to 2147483648 1,280 to 2,147,483,648 bits

Function
Queries an addition value of the data signal generation interval and the data stop interval for Burst signal generation (Burst generation cycle).

Example

Example

>:SOURce:PATTern:BURSt:CYCle?
< 25600

Compatibility
Incompatible with existing models.

:SOURce:PATTern:BURSt:TRIGdelay <port>,<numeric>

Parameter

Parameter

<port> = <CHARACTER PROGRAM DATA>
BURStout1 Burst Output
BURStout2 Burst Output2
Valid only when AUX Output is set to Burst Output2.

<numeric> = <DECIMAL NUMERIC PROGRAM DATA>
0 to 2147483648 0 to 2,147,483,648 bits, in 16-bit steps

Note:
The maximum setting is the pattern length.
At 2 ch Combination, the setting range and Step are doubled.
At 4 ch Combination, the setting range and Step are quadrupled.

Function
Sets the Burst output trigger signal generation timing (delay) to adjust the phase on the DUT side.

Example
To set the Burst output trigger generation timing (delay) to 128 bits:
>:SOURce:PATTern:BURSt:TRIGdelay BURStout1,128

Compatibility
Incompatible with existing models.

:SOURce:PATTern:BURSt:TRIGdelay? <port>

Parameter

Parameter

<port> = <CHARACTER PROGRAM DATA>
BURStout1 Burst Output
BURStout2 Burst Output2
Valid only when AUX Output is set to Burst Output2.

Response

Response

0 to 2147483648 0 to 2,147,483,648 bits

Function
Queries the Burst output trigger signal generation timing (delay).

Example

Example

>:SOURce:PATTern:BURSt:TRIGdelay? BURStout1
< 128

Compatibility
Incompatible with existing models.
7.3 PPG Commands

`:SOURce:PATTern:BURSt:PULSewidth <port>,<numeric>`

**Parameter**

- `<port>` = `<CHARACTER PROGRAM DATA>`
  - BURStout1: Burst Output
  - BURStout2: Burst Output2
  - Valid only when AUX Output is set to Burst Output2.
- `<numeric>` = `<DECIMAL NUMERIC PROGRAM DATA>`
  - 0 to 2147483648: 0 to 2,147,483,648 bits, in 16-bit steps

**Note:**

- The maximum setting is the pattern length.
- At 2 ch Combination, the setting range and Step are doubled.
- At 4 ch Combination, the setting range and Step are quadrupled.

**Function**

Sets the Burst output trigger signal generation interval width to adjust the phase on the DUT side.

**Example**

To set the Burst output trigger signal generation interval width to 1,024 bits:

```
> :SOURce:PATTern:BURSt:PULSewidth BURStout1,1024
```

**Compatibility**

Incompatible with existing models.

`:SOURce:PATTern:BURSt:PULSewidth? <port>`

**Parameter**

- `<port>` = `<CHARACTER PROGRAM DATA>`
  - BURStout1: Burst Output
  - BURStout2: Burst Output2
  - Valid only when AUX Output is set to Burst Output2.

**Response**

- `<numeric>` = `<NR1 NUMERIC RESPONSE DATA>`
  - 0 to 2147483648: 0 to 2,147,483,648 bits

**Function**

Queries the Burst output trigger signal generation interval width.

**Example**

To query the Burst output trigger signal generation interval width:

```
> :SOURce:PATTern:BURSt:PULSewidth? BURS1
< 1024
```

**Compatibility**

Incompatible with existing models.
7.3.4.2 AUX Input setting commands

![Figure 7.3.4.2-1 AUX Input setting field]

**Table 7.3.4.2-1 AUX Input setting commands**

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
</table>

**:INPut:INPut:FUNCtion <function>**

Parameter  

<function> = <CHARACTER PROGRAM DATA>

- BURSt: Burst
- ERRor: Error Injection
- SEQuence: Sequence Control
- ALTernate: Alternate Control

Function  

Sets the function of AUX Input.

Example  

To set the function of AUX Input to Error Injection.

> :INPut:INPut:FUNCtion ERRor

Compatibility  

Incompatible with existing models.

**:INPut:INPut:FUNCtion?**

Response  

<function> = <CHARACTER RESPONSE DATA>

- BURS, ERR, SEQ, ALT

Function  

Queries the function of Auxiliary Input.

Example  

> :INPut:INPut:FUNCtion?

< ERR

Compatibility  

Incompatible with existing models.
7.3.4.3 AUX Output setting commands

Figure 7.3.4.3-1 AUX Output setting field

Figure 7.3.4.3-2 AUX Output setting field (for PRBS, Zero-Substitution, and Data patterns)

Figure 7.3.4.3-3 AUX Output setting field (For Alternate pattern)

Figure 7.3.4.3-4 AUX Output setting field (For Mixed Data pattern)
Chapter 7  SCPI Commands

Figure 7.3.4.3-5  AUX Output setting field (For Mixed Alternate pattern)

Figure 7.3.4.3-6  AUX Output setting field (For Sequence pattern)

Table 7.3.4.3-1  AUX Output setting commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>:OUTPut:SYNC:SOURce?</td>
</tr>
<tr>
<td></td>
<td>(PRBS, Zero-Substitution, Data)</td>
<td>:OUTPut:SYNC:POSiton?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:OUTPut:SYNC:ALTerminate:CONTents?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:OUTPut:SYNC:ALTerminate:POSiton?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:OUTPut:SYNC:MIXData:BLOCk?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:OUTPut:SYNC:MIXData:ROW?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:OUTPut:SYNC:MIXalt:BLOCk?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:OUTPut:SYNC:MIXalt:CONTents?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:OUTPut:SYNC:MIXalt:ROW?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:OUTPut:SYNC:SEQuence:BLOCk?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:OUTPut:SYNC:SEQuence:POSiton?</td>
</tr>
</tbody>
</table>
7.3 PPG Commands

:OUTPut:SYNC:SOURce <source>[,<numeric>]

Parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;source&gt;</td>
<td>&lt;CHARACTER PROGRAM DATA&gt;</td>
</tr>
<tr>
<td>CLOCk8</td>
<td>1/8 clocks</td>
</tr>
<tr>
<td>CLOCk16</td>
<td>1/16 clocks</td>
</tr>
<tr>
<td>CLOCk32</td>
<td>1/32 clocks</td>
</tr>
<tr>
<td>CLOCk64</td>
<td>1/64 clocks</td>
</tr>
<tr>
<td>PATTern</td>
<td>Pattern Sync (Variable)</td>
</tr>
<tr>
<td>BURSt</td>
<td>Burst Output2</td>
</tr>
<tr>
<td>NCLock</td>
<td>1/n clocks</td>
</tr>
</tbody>
</table>

[<numeric>] = <DECIMAL NUMERIC PROGRAM DATA>

2, 4, 8 to 511 2, 4, 8 to 511, in single steps

Omit <numeric> when NCLock (1/n clocks) is not selected.

Valid only for MU181020A-001, and the setting upper limit varies as follows:

1/2 mode: 1, 2, 4, 8, 9 to 255, in single steps
1/4 mode: 1, 2, 4, 8, 9 to 127, in single steps
1/8 mode: 1, 2, 4, 8, 9 to 63, in single steps

Function
Sets the output signal for synchronization output.

Example
To set the output signal for synchronization output to 1/32 clocks:
> :OUTPut:SYNC:SOURce CLOCk32

To set the output signal for synchronization output to 1/511 clocks:
> :OUTPut:SYNC:SOURce NCLock,511

Compatibility
Compatible with the MP1632C.

:OUTPut:SYNC:SOURce?

Response

<table>
<thead>
<tr>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;source&gt; = &lt;CHARACTER RESPONSE DATA&gt;</td>
</tr>
<tr>
<td>CLOC8, CLOC16, CLOC32, CLOC64, PATT, BURS, NCL</td>
</tr>
<tr>
<td>[&lt;numeric&gt;] = &lt;DECIMAL NUMERIC RESPONSE DATA&gt;</td>
</tr>
</tbody>
</table>

Omit <numeric> for other than NCL (1/n clocks).

2, 4, 8 to 511, in single steps

Function
Queries the output signal for synchronization output.

Example
> :OUTPut:SYNC:SOURce?
< CLOC32
> :OUTPut:SYNC:SOURce?
< NCL, 511

Compatibility
Compatible with the MP1632C.
**Chapter 7  SCPI Commands**

**:OUTPut:SYNC:POSition <numeric>**

**Parameter**

<numeric> = <DECIMAL NUMERIC PROGRAM DATA>

1 to 6871947657  1 to 68,719,476,657 bits, in 16-bit steps

In the case of Combination

1 to 274877906625  1 to 274,877,906,625 bits, 64-bit steps

**Function**

Sets the output position for synchronization output for Data, PRBS, and Zero-Substitution patterns.

**Example**

To set the synchronization output position to bit 17:

> :OUTPut:SYNC:POSition 17

**Compatibility**

Partially compatible with the MP1632C.

Note that the resolution (step) is incompatible.

**:OUTPut:SYNC:POSition?**

**Response**

<numeric> = <NR1 NUMERIC RESPONSE DATA>

1 to 274877906625  1 to 274,877,906,625 bits

The maximum setting value is the value set in Pattern Length.

**Function**

Queries the output position for synchronization output for Data, PRBS, and Zero-Substitution patterns.

**Example**

> :OUTPut:SYNC:POSition?

< 17

**Compatibility**

Compatible with the MP1632C.

**:OUTPut:SYNC:ALTernate:CONTents <contents>**

**Parameter**

<contents> = <CHARACTER PROGRAM DATA>

APAT   Outputs synchronization signal for Pattern A.

BPAT   Outputs synchronization signal for Pattern B.

**Function**

Sets the output pattern for synchronization output for Alternate pattern.

**Example**

To set the output pattern for synchronization output to Pattern A:

> :OUTPut:SYNC:ALTernate:CONTents APAT

**Compatibility**

Incompatible with existing models.

**:OUTPut:SYNC:ALTernate:CONTents?**

**Response**

<contents> = <CHARACTER RESPONSE DATA>

APAT   Outputs synchronization signal for Pattern A.

BPAT   Outputs synchronization signal for Pattern B.

**Function**

Queries the output pattern for synchronization output for Alternate pattern.

**Example**

> :OUTPut:SYNC:ALTernate:CONTents?

< APAT

**Compatibility**

Incompatible with existing models.
### :OUTPut:SYNC:ALTERNate:POSition <numeric>

**Parameter**

<numeric> = <DECIMAL NUMERIC PROGRAM DATA>

1 to 67108785  1 to 67,108,785 bits, in 16-bit steps

**Function**

Sets the output position for synchronization output for Alternate pattern.

**Example**

To set the synchronization output position to 33 bits:

> :OUTPut:SYNC:ALTERNate:POSition 33

**Compatibility**

Incompatible with existing models.

### :OUTPut:SYNC:ALTERNate:POSition?

**Response**

<numeric> = <DECIMAL NUMERIC RESPONSE DATA>

1 to 67108785  1 to 67,108,785 bits

The maximum setting value is the value set in Pattern Length.

**Function**

Queries the output position for synchronization output for Alternate pattern.

**Example**

> :OUTPut:SYNC:ALTERNate:POSition?

< 33

**Compatibility**

Incompatible with existing models.

### :OUTPut:SYNC:MIXData:BLOCk <numeric>

**Parameter**

<numeric> = <DECIMAL NUMERIC PROGRAM DATA>

1 to 511  1 to 511 blocks, in 1-block steps

The maximum setting value is the set number of blocks.

**Function**

Sets the output block position for synchronization output for Mixed Data pattern.

**Example**

To set the synchronization output block to block 10:

> :OUTPut:SYNC:MIXData:BLOCk 10

**Compatibility**

Incompatible with existing models.

### :OUTPut:SYNC:MIXData:BLOCk?

**Response**

<numeric> = <NR1 NUMERIC RESPONSE DATA>

1 to 511  1 to 511 blocks

The maximum setting value is the set number of blocks.

**Function**

Queries the output block position for synchronization output for Mixed Data pattern.

**Example**

> :OUTPut:SYNC:MIXData:BLOCk?

< 10

**Compatibility**

Incompatible with existing models.
Chapter 7  SCPI Commands

**:OUTPut:SYNC:MIXData:ROW <numeric>**

Parameter  
<numeric> =  <DECIMAL NUMERIC PROGRAM DATA>
1 to 16  1 to 16 rows, in 1-row steps

Function  
Sets the output Row position for synchronization output for Mixed Data pattern.

Example  
To set the synchronization output Row to 1 Row.
> :OUTPut:SYNC:MIXData:ROW 1

Compatibility  
Incompatible with existing models.

**:OUTPut:SYNC:MIXData:ROW?**

Response  
<numeric> =  <NR1 NUMERIC RESPONSE DATA>
1 to 16  1 to 16 rows, in 1-row steps

Function  
Queries the output Row position for synchronization output for Mixed Data pattern.

Example  
> :OUTPut:SYNC:MIXData:ROW?
< 1

Compatibility  
Incompatible with existing models.

**:OUTPut:SYNC:MIXalt:BLOCk <numeric>**

Parameter  
<numeric> =  <DECIMAL NUMERIC PROGRAM DATA>
1 to 511  1 to 511 blocks, in 1-block steps

Function  
Sets the output block position for synchronization output for Mixed Alternate pattern.

Example  
To set the synchronization output block to block 10:
> :OUTPut:SYNC:MIXalt:BLOCk 10

Compatibility  
Incompatible with existing models.

**:OUTPut:SYNC:MIXalt:BLOCk?**

Response  
<numeric> =  <NR1 NUMERIC RESPONSE DATA>
1 to 511  1 to 511 blocks

Function  
Queries the output block position for synchronization output for Mixed Alternate pattern.

Example  
> :OUTPut:SYNC:MIXalt:BLOCk?
< 10

Compatibility  
Incompatible with existing models.
7.3 PPG Commands

**:OUTPut:SYNC:MIXalt:CONTents <contents>**

**Parameter**
<contents> = <CHARACTER PROGRAM DATA>
APAT  Outputs synchronization signal for Pattern A.
BPAT  Outputs synchronization signal for Pattern B.

**Function**
Sets the output pattern for synchronization output for Mixed Alternate pattern.

**Example**
To set the synchronization output position to Pattern A:
> :OUTPut:SYNC:MIXalt:CONTents APAT

**Compatibility**
Incompatible with existing models.

**:OUTPut:SYNC:MIXalt:CONTents?**

**Response**
<contents> = <CHARACTER RESPONSE DATA>
APAT  Outputs synchronization signal for Pattern A.
BPAT  Outputs synchronization signal for Pattern B.

**Function**
Queries the output pattern for synchronization output for Mixed Alternate pattern.

**Example**
> :OUTPut:SYNC:MIXalt:CONTents?
< APAT

**Compatibility**
Incompatible with existing models.

**:OUTPut:SYNC:MIXalt:ROW <numeric>**

**Parameter**
<numeric> = <DECIMAL NUMERIC PROGRAM DATA>
1 to 16  1 to 16 rows, in 1-row steps
The maximum setting value is the set number of Rows.

**Function**
Sets the output Row position for synchronization output for Mixed Alternate pattern.

**Example**
To set the synchronization output row to 2 Row:
> :OUTPut:SYNC:MIXalt:ROW 2

**Compatibility**
Incompatible with existing models.

**:OUTPut:SYNC:MIXalt:ROW?**

**Response**
<numeric> = <NR1 NUMERIC RESPONSE DATA>
1 to 16  1 to 16 rows
The maximum setting value is the set number of Rows.

**Function**
Queries the output Row position for synchronization output for Mixed Alternate pattern.

**Example**
> :OUTPut:SYNC:MIXalt:ROW?
< 2

**Compatibility**
Incompatible with existing models.
### :OUTPut:SYNC:SEQuence:BLOCk <numeric>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;numeric&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1 to 128</td>
</tr>
<tr>
<td>Function</td>
<td>1 to 128 blocks, in 1-block steps</td>
</tr>
<tr>
<td>Example</td>
<td>The maximum setting value is the set number of blocks.</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Incompatible with existing models.</td>
</tr>
</tbody>
</table>

#### Example
To set the synchronization output block to block 128:

```
> :OUTPut:SYNC:SEQuence:BLOCk 128
```

#### Compatibility
Incompatible with existing models.

### :OUTPut:SYNC:SEQuence:POSition <numeric>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;numeric&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1 to 1048497</td>
</tr>
<tr>
<td>Function</td>
<td>1 to 1,048,497 bits, in 16-bit steps</td>
</tr>
<tr>
<td>Example</td>
<td>Sets the output position for synchronization output for Sequence pattern.</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Incompatible with existing models.</td>
</tr>
</tbody>
</table>

#### Example
To set the synchronization output position to 1,048,497:

```
> :OUTPut:SYNC:SEQuence:POSition 1048497
```

#### Compatibility
Incompatible with existing models.

### :OUTPut:SYNC:SEQuence:POSition?

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;numeric&gt; = &lt;DECIMAL NUMERIC RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1 to 1048497</td>
</tr>
<tr>
<td>Function</td>
<td>1 to 1,048,497 bits</td>
</tr>
<tr>
<td>Example</td>
<td>Queries the output position for synchronization output for Sequence pattern.</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Incompatible with existing models.</td>
</tr>
</tbody>
</table>

#### Example
To set the synchronization output position to 1,048,497:

```
```

#### Compatibility
Incompatible with existing models.
7.4 ED Commands

This section describes the commands for the MU181040A 12.5 Gbit/s ED and the MU181040B 14 Gbit/s ED. Before executing a setting/query command for the MU181040A, specify the slot number of the module to be remotely controlled, by using the `:MODule:ID` command. Refer to Section 7.1 “Common Commands” and 9.1 “Basic Operation of Device Messages” for how to specify a slot number with the `:MODule:ID` command.

However, unless there is a special item, see the MU181040A.

7.4.1 Commands related to Result tab

![Figure 7.4.1-1 Result tab](image_url)
Chapter 7  SCPI Commands

7.4.1.1 Result setting field

![Start/Stop buttons](image1)

![Result setting field](image2)

**Figure 7.4.1.1-1  Start/Stop buttons**

**Figure 7.4.1.1-2  Result setting field**

**Table 7.4.1.1-1  Result setting commands**

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1]</td>
<td>Start</td>
<td><code>:SENSe:MEASure:STARt</code></td>
</tr>
<tr>
<td></td>
<td></td>
<td><code>:DISPlay:RESult:SETTing?</code></td>
</tr>
<tr>
<td></td>
<td></td>
<td><code>:DISPlay:RESult:TIME?</code></td>
</tr>
<tr>
<td>[6]</td>
<td>Query for measurement start time</td>
<td><code>:SENSe:MEASure:EALarm:STARt?</code></td>
</tr>
<tr>
<td>[7]</td>
<td>Query for measurement stop time</td>
<td><code>:SENSe:MEASure:EALarm:STOP?</code></td>
</tr>
<tr>
<td>[8]</td>
<td>Query for measurement elapsed time</td>
<td><code>:SENSe:MEASure:EALarm:ELAPsed?</code></td>
</tr>
<tr>
<td>[10]</td>
<td>Query for intermediate data creation time</td>
<td><code>:SENSe:MEASure:EALarm:ITIME?</code></td>
</tr>
<tr>
<td>[12]</td>
<td>Query for alarm recovery time</td>
<td><code>:SENSe:MEASure:EALarm:ARECover?</code></td>
</tr>
</tbody>
</table>
### :SENSe:MEASure:STARt

**Function**
Starts BER measurement for the specified module. If this command is issued during measurement, the BER measurement is restarted.

**Example**
> :SENSe:MEASure:STARt

**Compatibility**
Compatible with the MP1632C and MP1776A.

### :SENSe:MEASure:STOP

**Function**
Stops the BER measurement of the specified module.

**Example**
> :SENSe:MEASure:STOP

**Compatibility**
Compatible with the MP1632C and MP1776A.

### :SENSe:MEASure:EALarm:STATe?

**Response**

```
<numeric> = <NR1 NUMERIC RESPONSE DATA>
```

```
0   Measurement stops
1   During measurement
```

**Function**
Queries the measurement processing state during bit error alarm measurement.

**Example**
> :SENSe:MEASure:EALarm:STATe?
< 0

**Compatibility**
Compatible with the MP1632C and MP1776A.

### :DISPlay:RESult:SETTing <setting>

**Parameter**

```
<setting> = <CHARACTER PROGRAM DATA>
```

- EALarm: Bit error alarm measurement result
- LOGGing: Logging measurement result
- HISTogram: Histogram measurement result

**Function**
Sets the display item on the BER measurement result screen.

**Example**
To set the display item on the BER measurement result screen to the bit error alarm measurement results:
> :DISPlay:RESult:SETTing EALarm

**Compatibility**
Incompatible with existing models.

### :DISPlay:RESult:SETTing?

**Response**

```
<setting> = <CHARACTER RESPONSE DATA>
```

```
EAL, LOGG, HIST
```

**Function**
Queries the display item on the BER measurement result screen.

**Example**
> :DISPlay:RESult:SETTing?
< EAL

**Compatibility**
Incompatible with existing models.
Chapter 7  SCPI Commands

:DISPLAY:RESULT:TIME <time>

Parameter
<time> = <CHARACTER PROGRAM DATA>
- DTIMe: Displays the current date and time.
- STARt: Displays the measurement start time.
- ELAPsed: Displays the elapsed time based on the measurement period.
- TIMed: Displays the remaining time based on the measurement period.

Function
Sets the measurement time display type.

Example
To set the measurement time display type to DTIMe (current date and time):
> :DISPLAY:RESULT:TIME DTIMe

Compatibility
Compatible with the MP1632C and MP1776A.

:DISPLAY:RESULT:TIME?

Response
<time> = <CHARACTER RESPONSE DATA>
- DTIM, STAR, ELAP, TIM

Function
Queries the measurement time display type.

Example
> :DISPLAY:RESULT:TIME?
< DTIM

Compatibility
Compatible with the MP1632C and MP1776A.

:SENSe:MEASure:EALarm:STARt?

Response
<year>,<month>,<day>,<hour>,<min>,<second> = <NR1 NUMERIC RESPONSE DATA>
- <year> 0, 2000 to 2036 2000 to 2036
- <month> 0, 1 to 12 January to December
- <day> 0, 1 to 31 1st to 31st
- <hour> 0 to 23 0 to 23 hours
- <min> 0 to 59 0 to 59 minutes
- <second> 0 to 59 0 to 59 seconds

"0,0,0,0,0,0" is returned if there is no measurement start time data.

Function
Queries the measurement start time during bit error alarm measurement.

Example
> :SENSe:MEASure:EALarm:STARt?
< 2006,1,1,23,59,59

Compatibility
Compatible with the MP1632C and MP1776A.
7.4 ED Commands

:SENSe:MEASure:EALarm:STOP?

Response

\[
\text{<year>,<month>,<day>,<hour>,<min>,<second> = } \text{<NR1 NUMERIC RESPONSE DATA>}
\]

- \text{<year> 0, 2000 to 2036  2000 to 2036}
- \text{<month> 0, 1 to 12  Jan to Dec}
- \text{<day> 0, 1 to 31  1st to 31st}
- \text{<hour> 0 to 23  0 to 23 hours}
- \text{<min> 0 to 59  0 to 59 minutes}
- \text{<second> 0 to 59  0 to 59 seconds}

"0,0,0,0,0,0" is returned if there is no measurement stop time data.

Function
Queries the measurement end time during bit error alarm measurement.

Example

> :SENSe:MEASure:EALarm:STOP?
< 0,0,0,0,0,0

Compatibility
Compatible with the MP1632C and MP1776A.

:SENSe:MEASure:EALarm:ELAPsed?

Response

\[
\text{<day>,<hour>,<min>,<second> = } \text{<NR1 NUMERIC RESPONSE DATA>}
\]

- \text{<day> 0 to 31  0 to 31 days}
- \text{<hour> 0 to 23  0 to 23 hours}
- \text{<min> 0 to 59  0 to 59 minutes}
- \text{<second> 0 to 59  0 to 59 seconds}

Function
Queries the measurement elapsed time during bit error alarm measurement.

Example

> :SENSe:MEASure:EALarm:ELAPsed?
< 0,1,0,0

Compatibility
Compatible with the MP1632C and MP1776A.

:SENSe:MEASure:EALarm:TIMed?

Response

\[
\text{<day>,<hour>,<min>,<second> = } \text{<NR1 NUMERIC RESPONSE DATA>}
\]

- \text{<day> 0 to 31  0 to 31 days}
- \text{<hour> 0 to 23  0 to 23 hours}
- \text{<min> 0 to 59  0 to 59 minutes}
- \text{<second> 0 to 59  0 to 59 seconds}

"0,0,0,0" is returned when the measurement is finished.

When measurement is not performed, the measurement period set at that time is returned.

Function
Queries the measurement remaining time during bit error alarm measurement.

Example

> :SENSe:MEASure:EALarm:TIMed?
< 0,0,0,1

Compatibility
Compatible with the MP1632C and MP1776A.
Chapter 7  SCPI Commands

:SENSe:MEASure:EALarm:ITIMe?

Response  

Response  

Response  

Response  

Response  

Response  

Response  

Response  

Response  

Response  

Response  

Response  

Response  

Response  

Response  

Response  

Response  

Response  

Response  

Response  

Response  

Response  

Response  

Response  

Function  

Function  

Function  


during bit error alarm measurement.

Function  

Function  

"0,0,0,0,0,0" is returned if there is no measurement intermediate data.

"0,0,0,0,0,0" is returned if there is no measurement intermediate data.

Example  

Example  

Example  

Example  

Example  

Example  

"0,0,0,0,0,0" is returned if no alarm has occurred.

"0,0,0,0,0,0" is returned if no alarm has occurred.

"0,0,0,0,0,0" is returned if no alarm has occurred.

"0,0,0,0,0,0" is returned if no alarm has occurred.

"0,0,0,0,0,0" is returned if no alarm has occurred.

"0,0,0,0,0,0" is returned if no alarm has occurred.

Example  

Example  

Example  

Example  

Example  

Example  

"0,0,0,0,0,0" is returned if no alarm has occurred.

"0,0,0,0,0,0" is returned if no alarm has occurred.

"0,0,0,0,0,0" is returned if no alarm has occurred.

"0,0,0,0,0,0" is returned if no alarm has occurred.

"0,0,0,0,0,0" is returned if no alarm has occurred.

"0,0,0,0,0,0" is returned if no alarm has occurred.

Example  

Example  

Example  

Example  

Example  

Example  

"0,0,0,0,0,0" is returned if no alarm has occurred.

"0,0,0,0,0,0" is returned if no alarm has occurred.

"0,0,0,0,0,0" is returned if no alarm has occurred.

"0,0,0,0,0,0" is returned if no alarm has occurred.

"0,0,0,0,0,0" is returned if no alarm has occurred.

"0,0,0,0,0,0" is returned if no alarm has occurred.

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

Compatibility  

 Compatibility  

Compatibility  

 Compatibility  

Compatibility  

 Compatibility  

Compatibility  

 Compatibility  

Compatibility  

 Compatibility  

Compatibility  

 Compatibility  

Compatibility  

 Compatibility  

Compatibility  

 Compatibility  

Compatibility  

 Compatibility  

Compatibility  

 Compatibility  

Compatibility  

 Compatibility  

Compatibility  

 Compatibility  

Compatibility  

 Compatibility  

Compatibility  

 Compatibility  

Comma...
### :SENSe:MEASure:EALarm:ARECover? <alarm>

**Parameter**

- `<alarm>`: `<CHARACTER PROGRAM DATA>`
- `CLOS`: Clock Loss
- `PSL`: Pattern Sync Loss
- `CRU`: CR Unlock

**Response**

- `<year>,<month>,<day>,<hour>,<min>,<second>`: `<NR1 NUMERIC RESPONSE DATA>`
- `<year>`: `0`, `2000` to `2036`
- `<month>`: `0`, `1` to `12`
- `<day>`: `0`, `1` to `31`
- `<hour>`: `0` to `23`
- `<min>`: `0` to `59`
- `<second>`: `0` to `59`

**Function**

Queries the time when a specified alarm has recovered during bit error alarm measurement.

"0, 0, 0, 0, 0, 0" is returned if no alarm has been recovered (occurred).

**Example**

To query the time when the clock loss has been recovered during bit error alarm measurement:

```
> :SENSe:MEASure:EALarm:ARECover? CLOS
< 0,0,0,1,0,0
```

**Compatibility**

Partially compatible with the MP1632C (only the clock loss and pattern sync loss are compatible).

### :SENSe:DISPlay:SETTIng <setting>

**Parameter**

- `<setting>`: `<CHARACTER PROGRAM DATA>`
- `INPut`: Displays Input items
- `GATing`: Displays Gating items
- `CONDition`: Displays Condition items
- `AUTosync`: Displays AutoSync items
- `SYNCControl`: Displays SyncControl items

**Function**

Sets display items in the Result tab

**Example**

To display the Input items:

```
> :SENSe:DISPlay:SETTIng INPut
```

**Compatibility**

Incompatible with existing models.
7.4.1.2 Error/Alarm setting commands

![Error/Alarm result display screen](image1)

- Error/Alarm
- Date & Time

![Error/Alarm result display screen (when Zoom is selected)](image2)

Figure 7.4.1.2-1 Error/Alarm result display screen

Figure 7.4.1.2-2 Error/Alarm result display screen (when Zoom is selected)
Table 7.4.1.2-1  Error/Alarm setting commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>:DISPlay:RESult:ZOOM?</td>
</tr>
<tr>
<td>[3]</td>
<td>No label (Query for error alarm measurement data)</td>
<td>:CALCulate:DATA:EALarm?</td>
</tr>
<tr>
<td>[4]</td>
<td>No label (Query for monitoring item)</td>
<td>:CALCulate:DATA:MONitor?</td>
</tr>
</tbody>
</table>

**:DISPlay:RESult:ZOOM <boolean>**

Parameter  

<boolean> = <BOOLEAN PROGRAM DATA>

0 or OFF  Display enlargement OFF

1 or ON   Display enlargement ON

Function  

Sets display enlargement for the measurement result display screen ON/OFF.

Example  

To set enlargement for the measurement result display screen ON:

> :DISPlay:RESult:ZOOM ON

Compatibility  

Incompatible with existing models.

**:DISPlay:RESult:ZOOM?**

Response  

<numeric> = <NR1 NUMERIC RESPONSE DATA>

0  Display enlargement OFF

1  Display enlargement ON

Function  

Queries the display enlargement for the measurement result display screen ON/OFF state.

Example  

> :DISPlay:RESult:ZOOM?

< 1

Compatibility  

Incompatible with existing models.

**:DISPlay:RESult:EALarm:HRESet**

Function  

Resets the histories on the measurement result display screen.

Example  

> :DISPlay:RESult:EALarm:HRESet

Compatibility  

Compatible with the MP1632C and MP1776A.
**Chapter 7  SCPI Commands**

:CALCulate:DATA:EALarm? <string>

Parameter

<string> = <STRING PROGRAM DATA>
"CURRent:<result>"  Current data
"LAST:<result>"  Measurement end data
"INTermediate:<result>"  Measurement intermediate data

See Table 7.4.1.2-2 for details on <result>.

Response

<string> = <STRING RESPONSE DATA>

Table 7.4.1.2-2  Response format

<table>
<thead>
<tr>
<th>Form</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Form1</td>
<td>&quot;XXXXXXX&quot;</td>
<td>For 0 to 9999999</td>
</tr>
<tr>
<td>Integer type</td>
<td>&quot;X.XXXXXXX&quot;</td>
<td>For 1.0000E07 to 9.9999E17</td>
</tr>
<tr>
<td></td>
<td>&quot;--------&quot;</td>
<td>When no data corresponds to the query</td>
</tr>
<tr>
<td>Form2</td>
<td>&quot;X.XXXXXE-XX&quot;</td>
<td>For 0.0001E-18 to 1.0000E00</td>
</tr>
<tr>
<td>Fraction type</td>
<td>&quot;--------&quot;</td>
<td>When no data corresponds to the query</td>
</tr>
<tr>
<td>Form3</td>
<td>&quot;XXX.XXXXX&quot;</td>
<td>For 0.0000 to 100.0000</td>
</tr>
<tr>
<td>% type</td>
<td>&quot;--------&quot;</td>
<td>When no data corresponds to the query</td>
</tr>
<tr>
<td>Form4</td>
<td>&quot;XXXXXXXXX&quot;</td>
<td>For 0 to MAX (Hz)</td>
</tr>
<tr>
<td>Frequency type</td>
<td>&quot;--------&quot;</td>
<td>When no data corresponds to the query</td>
</tr>
<tr>
<td>Form5</td>
<td>&quot;X.XXX,XX.XXX&quot;</td>
<td>For −4.000 to +4.000</td>
</tr>
<tr>
<td>Voltage type</td>
<td>&quot;--------&quot;</td>
<td>When no data corresponds to the query</td>
</tr>
<tr>
<td>Form6</td>
<td>&quot;XXXX,XXXX&quot;</td>
<td>For −1000 to +1000, −10000 to +10000</td>
</tr>
<tr>
<td>Phase type</td>
<td>&quot;--------&quot;</td>
<td>When no data corresponds to the query</td>
</tr>
</tbody>
</table>

Function

Queries the measurement data that corresponds to the parameter.

Table 7.4.1.2-3  Parameters

<table>
<thead>
<tr>
<th>Items</th>
<th>&lt;result&gt;</th>
<th>Form</th>
</tr>
</thead>
<tbody>
<tr>
<td>Error Rate</td>
<td>Total</td>
<td>ER:TOTal Form2</td>
</tr>
<tr>
<td>INS</td>
<td>ER:INSertion</td>
<td></td>
</tr>
<tr>
<td>OMI</td>
<td>ER:OMISSion</td>
<td></td>
</tr>
<tr>
<td>Transition</td>
<td>ER:TRANsition</td>
<td></td>
</tr>
<tr>
<td>Non Transition</td>
<td>ER:NONTransion</td>
<td></td>
</tr>
<tr>
<td>Error Count</td>
<td>Total</td>
<td>EC:TOTal Form1</td>
</tr>
<tr>
<td>INS</td>
<td>EC:INSertion</td>
<td></td>
</tr>
<tr>
<td>OMI</td>
<td>EC:OMISSion</td>
<td></td>
</tr>
<tr>
<td>Transition</td>
<td>EC:TRANsition</td>
<td></td>
</tr>
<tr>
<td>Non Transition</td>
<td>EC:NONTransion</td>
<td></td>
</tr>
<tr>
<td>%EFI</td>
<td>Total</td>
<td>EFI:TOTal Form3</td>
</tr>
<tr>
<td>EI</td>
<td>Total</td>
<td>EI:TOTal Form1</td>
</tr>
<tr>
<td>Frequency (kHz)</td>
<td>FREQuency</td>
<td>Form4</td>
</tr>
<tr>
<td>Clock Count</td>
<td>CC:TOTal</td>
<td>Form1</td>
</tr>
<tr>
<td>Clock Loss</td>
<td>AINTerval:CLOSs</td>
<td>Form1</td>
</tr>
</tbody>
</table>
### Table 7.4.1.2-3 Parameters (Cont’d)

<table>
<thead>
<tr>
<th>Items</th>
<th>&lt;result&gt;</th>
<th>Form</th>
</tr>
</thead>
<tbody>
<tr>
<td>CR Unlock</td>
<td>AINTerval:CRUNlock</td>
<td>Form1</td>
</tr>
<tr>
<td>Sync Loss</td>
<td>AINTerval:PSLoss</td>
<td>Form1</td>
</tr>
<tr>
<td>Data Threshold</td>
<td>THReshold</td>
<td>Form5</td>
</tr>
<tr>
<td>Data Phase</td>
<td>PHASe</td>
<td>Form6</td>
</tr>
<tr>
<td>Threshold EI</td>
<td>&gt;1.0E–3 THReshold:EI:TOTal:E_3</td>
<td>Form1</td>
</tr>
<tr>
<td></td>
<td>&gt;1.0E–4 THReshold:EI:TOTal:E_4</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&gt;1.0E–5 THReshold:EI:TOTal:E_5</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&gt;1.0E–6 THReshold:EI:TOTal:E_6</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&gt;1.0E–7 THReshold:EI:TOTal:E_7</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&gt;1.0E–8 THReshold:EI:TOTal:E_8</td>
<td></td>
</tr>
<tr>
<td></td>
<td>≤1.0E–8 THReshold:EI:TOTal:UE_8</td>
<td></td>
</tr>
<tr>
<td>Threshold%EFI</td>
<td>&gt;1.0E–3 THReshold:EFI:TOTal:E_3</td>
<td>Form3</td>
</tr>
<tr>
<td></td>
<td>&gt;1.0E–4 THReshold:EFI:TOTal:E_4</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&gt;1.0E–5 THReshold:EFI:TOTal:E_5</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&gt;1.0E–6 THReshold:EFI:TOTal:E_6</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&gt;1.0E–7 THReshold:EFI:TOTal:E_7</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&gt;1.0E–8 THReshold:EFI:TOTal:E_8</td>
<td></td>
</tr>
<tr>
<td></td>
<td>≤1.0E–8 THReshold:EFI:TOTal:UE_8</td>
<td></td>
</tr>
<tr>
<td>G.821</td>
<td>ES G821:ES2:TOTal</td>
<td>Form1</td>
</tr>
<tr>
<td></td>
<td>EFS G821:EFS2:TOTal</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SES G821:SES2:TOTal</td>
<td></td>
</tr>
<tr>
<td></td>
<td>DM G821:DM2:TOTal</td>
<td></td>
</tr>
<tr>
<td></td>
<td>US G821:US2:TOTal</td>
<td></td>
</tr>
<tr>
<td></td>
<td>%ES G821:ES:TOTal</td>
<td>Form3</td>
</tr>
<tr>
<td></td>
<td>%EFS G821:EFS:TOTal</td>
<td></td>
</tr>
<tr>
<td></td>
<td>%SES G821:SES:TOTal</td>
<td></td>
</tr>
<tr>
<td></td>
<td>%DM G821:DM:TOTal</td>
<td></td>
</tr>
<tr>
<td></td>
<td>%US G821:US:TOTal</td>
<td></td>
</tr>
</tbody>
</table>

**Example**

To query the total error rate measurement data for the current data:

```
> :CALCulate:DATA:EALarm? "CURRENT:ER:TOTal"
< "0.0000E-16"
```

**Compatibility**

Incompatible with existing models.
**Chapter 7  SCPI Commands**

### :CALCulate:DATA:MONitor? <item>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;item&gt; = &lt;STRING PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;BIT:TOTal&quot;</td>
<td>Bit Error (Total Error)</td>
</tr>
<tr>
<td>&quot;CLOSs&quot;</td>
<td>Clock Loss</td>
</tr>
<tr>
<td>&quot;PSLoss&quot;</td>
<td>Pattern Sync Loss</td>
</tr>
<tr>
<td>&quot;CRUNlock&quot;</td>
<td>CR Unlock</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;string&gt; = &lt;STRING RESPONSE DATA&gt;</th>
</tr>
</thead>
</table>

#### Table 7.4.1.2-4  Response

<table>
<thead>
<tr>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;Occur&quot;</td>
<td>When an alarm occurs</td>
</tr>
<tr>
<td>&quot;Not Occur&quot;</td>
<td>When no alarm occurs</td>
</tr>
<tr>
<td>&quot;-------&quot;</td>
<td>When no data corresponds to the query</td>
</tr>
</tbody>
</table>

### Function

Queries the real-time occurrence status of the monitoring item corresponding to the parameter.

### Example

To query the bit error occurrence status.

```
> :CALCulate:DATA:MONitor? "BIT:TOTal"
< "Occur"
```

### Compatibility

Partially compatible with the MP1776A (only the sync loss and pattern sync loss are compatible).
7.4.1.3 Logging setting commands

![Logging setting field](image)

Figure 7.4.1.3-1 Logging setting field
### Figure 7.4.1.3-2  Log Condition dialog box

### Table 7.4.1.3-1  Logging setting commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1]</td>
<td>Logging ON/OFF</td>
<td>:DISPlay:RESult:LOGGing:ONOFF</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:DISPlay:RESult:LOGGing:ONOFF?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:DISPlay:RESult:LOGGing:CONDition:PERiod?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:DISPlay:RESult:LOGGing:CONDition:ITEM?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:DISPlay:RESult:LOGGing:CONDition:THReshold?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:DISPlay:RESult:LOGGing:CONDition:SQUelch?</td>
</tr>
</tbody>
</table>
### :DISPLAY:RESult:LOGGing:ONOFf <boolean>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;boolean&gt; = &lt;BOOLEAN PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 or OFF</td>
<td>OFF</td>
</tr>
<tr>
<td>1 or ON</td>
<td>ON</td>
</tr>
</tbody>
</table>

**Function**: Sets logging execution ON/OFF.

**Example**: To set logging execution ON:

```
> :DISPlay:RESult:LOGGing:ONOFf ON
```

**Compatibility**: Incompatible with existing models.

### :DISPLAY:RESult:LOGGing:ONOFf?

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;numeric&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>OFF</td>
</tr>
<tr>
<td>1</td>
<td>ON</td>
</tr>
</tbody>
</table>

**Function**: Queries the logging execution ON/OFF state.

**Example**: To query the logging execution ON/OFF state:

```
> :DISPlay:RESult:LOGGing:ONOFf?
< 1
```

**Compatibility**: Incompatible with existing models.

### :DISPLAY:RESult:LOGGing:CLEar

**Function**: Clears the logging results.

**Example**: To clear the logging results:

```
> :DISPlay:RESult:LOGGing:CLEar
```

**Compatibility**: Incompatible with existing models.

### :DISPLAY:RESult:LOGGing:CONDition:PERiod <period>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;period&gt; = &lt;CHARACTER PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>S10</td>
<td>10 s</td>
</tr>
<tr>
<td>S30</td>
<td>30 s</td>
</tr>
<tr>
<td>M1</td>
<td>1 min</td>
</tr>
<tr>
<td>M10</td>
<td>10 min</td>
</tr>
<tr>
<td>M30</td>
<td>30 min</td>
</tr>
<tr>
<td>H1</td>
<td>1 hour</td>
</tr>
</tbody>
</table>

**Function**: Sets the logging result capturing period.

**Example**: To set the logging result capturing period to 10 seconds:

```
> :DISPlay:RESult:LOGGing:CONDition:PERiod S10
```

**Compatibility**: Incompatible with existing models.
Chapter 7  SCPI Commands

:DISPlay:RESult:LOGGing:CONDition:PERiod?

Response  
<period> = <CHARACTER RESPONSE DATA>
S10, S30, M1, M10, M30, H1

Function  
Queries the logging result capturing period.

Example  
> :DISPlay:RESult:LOGGing:CONDition:PERiod?
< S10

Compatibility  
Incompatible with existing models.

:DISPlay:RESult:LOGGing:CONDition:ITEM <item>,<boolean>

Parameter  
<item> = <CHARACTER PROGRAM DATA>

SLOTinfo  Slot Information: Slot information
TPATtern  Test Pattern: Test pattern type
START  Start Time: Measurement start time
END  End Time: Measurement end time
ERATe  Error Rate/Count: Error rate and error count
EI  EI/%EFI: Error interval and error free interval rate
FREQuency  Frequency: Frequency
CCOunt  Clock Count: Clock count
AOCCur  Alarm Occur/Recover: Alarm occurrence and recovery time
AINTerval  Alarm Interval: Alarm interval
DTHReshold  Adds the data input threshold setting value to the logging item.
CDELay  Adds the clock phase setting value to the logging item.
SDATa  1 second Data: Average data for 1 second

<boolean> = <BOOLEAN PROGRAM DATA>

0 or OFF  Measurement item OFF
1 or ON  Measurement item ON

Function  
Sets the specified logging measurement item ON/OFF.

Example  
To set the logging slot information ON:
> :DISPlay:RESult:LOGGing:CONDition:ITEM SLOTinfo,ON

Compatibility  
Incompatible with existing models.
### :DISPlay:RESult:LOGGing:CONDition:ITEM? <item>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;item&gt; = &lt;CHARACTER PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SLOTinfo, TPATtern, STARt, END, ERATe, EI, FREQuency, CCOunt, AOCCur, AINTerval, DTHReshold, CDELay, SDATa</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;numeric&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Measurement item OFF</td>
</tr>
<tr>
<td></td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Measurement item ON</td>
</tr>
</tbody>
</table>

| Function | Queries the specified logging measurement item ON/OFF state. |
| Example  | > :DISPlay:RESult:LOGGing:CONDition:ITEM? SLOT |
|          | < 1 |

| Compatibility | Incompatible with existing models. |

### :DISPlay:RESult:LOGGing:CONDition:THReshold <thre>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;thre&gt; = &lt;CHARACTER PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>E_0   =&gt;0</td>
</tr>
<tr>
<td></td>
<td>E_3   &gt;1E-3</td>
</tr>
<tr>
<td></td>
<td>E_4   &gt;1E-4</td>
</tr>
<tr>
<td></td>
<td>E_5   &gt;1E-5</td>
</tr>
<tr>
<td></td>
<td>E_6   &gt;1E-6</td>
</tr>
<tr>
<td></td>
<td>E_7   &gt;1E-7</td>
</tr>
<tr>
<td></td>
<td>E_8   &gt;1E-8</td>
</tr>
<tr>
<td></td>
<td>E_9   &gt;1E-9</td>
</tr>
</tbody>
</table>

| Function | Sets the logging error rate threshold value. |
| Example  | To set the logging error rate threshold value to 0. |
|          | > :DISPlay:RESult:LOGGing:CONDition:THReshold E_0 |

| Compatibility | Incompatible with existing models. |

### :DISPlay:RESult:LOGGing:CONDition:THReshold?

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;thre&gt; = &lt;CHARACTER RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>E_0, E_3, E_4, E_5, E_6, E_7, E_8, E_9</td>
</tr>
</tbody>
</table>

| Function | Queries the logging error rate threshold value. |
| Example  | > :DISPlay:RESult:LOGGing:CONDition:THReshold? |
|          | < E_0 |

| Compatibility | Incompatible with existing models. |
### :DISPlay:RESult:LOGGing:CONDition:SQUelch <boolean>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;boolean&gt; = &lt;BOOLEAN PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 or OFF</td>
<td>Logging memory squelch function OFF</td>
</tr>
<tr>
<td>1 or ON</td>
<td>Logging memory squelch function ON</td>
</tr>
</tbody>
</table>

**Function**
Sets the logging memory squelch function ON/OFF.

**Example**
To set the logging memory squelch function ON:

```
> :DISPlay:RESult:LOGGing:CONDition:SQUelch ON
```

**Compatibility**
Incompatible with existing models.

### :DISPlay:RESult:LOGGing:CONDition:SQUelch?

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;numeric&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Logging memory squelch function OFF</td>
</tr>
<tr>
<td>1</td>
<td>Logging memory squelch function ON</td>
</tr>
</tbody>
</table>

**Function**
Queries the logging memory squelch function ON/OFF state.

**Example**

```
> :DISPlay:RESult:LOGGing:CONDition:SQUelch?
< 1
```

**Compatibility**
Incompatible with existing models.

### :DISPlay:RESult:LOGGing:CONDition:ALLSet <boolean>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;boolean&gt; = &lt;BOOLEAN PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 or OFF</td>
<td>All reset</td>
</tr>
<tr>
<td>1 or ON</td>
<td>All Set</td>
</tr>
</tbody>
</table>

**Function**
Selects or resets all logging measurement items.

**Example**
Selects all logging measurement items.

```
> :DISPlay:RESult:LOGGing:CONDition:ALLSet ON
```

**Compatibility**
Incompatible with existing models.
7.4 ED Commands

7.4.1.4 Histogram setting commands

Table 7.4.1.4-1 Histogram setting commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>:DISPlay:RESult:HISTogram:ONOff?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:DISPlay:RESult:HISTogram:SCALe?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:DISPlay:RESult:HISTogram:TYPE?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:DISPlay:RESult:HISTogram:TIMescale?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:DISPlay:RESult:HISTogram:MARKer?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:DISPlay:RESult:HISTogram:RESolution?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:DISPlay:RESult:HISTogram:DISPtime?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:DISPlay:RESult:HISTogram:EDETect?</td>
</tr>
<tr>
<td>[9]</td>
<td>Query for measurement result</td>
<td>:CALCulate:DATA:HISTogram?</td>
</tr>
</tbody>
</table>
### :DISPlay:RESult:HISTogram:ONOFf <boolean>

**Parameter**
- `<boolean> = <BOOLEAN PROGRAM DATA>`
  - 0 or OFF: Histogram execution OFF
  - 1 or ON: Histogram execution ON

**Function**
Sets histogram execution ON/OFF.

**Example**
To set histogram execution ON:
```
> :DISPlay:RESult:HISTogram:ONOFf ON
```

**Compatibility**
Incompatible with existing models.

### :DISPlay:RESult:HISTogram:ONOFf?

**Response**
- `<numeric> = <NR1 NUMERIC RESPONSE DATA>`
  - 0: Histogram execution OFF
  - 1: Histogram execution ON

**Function**
Queries the histogram execution ON/OFF state.

**Example**
```
> :DISPlay:RESult:HISTogram:ONOFf?
< 1
```

**Compatibility**
Incompatible with existing models.

### :DISPlay:RESult:HISTogram:SCALe <top>,<bottom>

**Parameter**
- `<top> = <CHARACTER PROGRAM DATA>`
  - When the measurement item is Error Rate:
    - E_{-17} to E_{0}: E_{-17} to E_{0}, in single steps
  - When the measurement item is Error Count/EI:
    - E_{1} to E_{18}: E_{1} to E_{18}, in single steps
  - `<bottom> = <CHARACTER PROGRAM DATA>`
    - When the measurement item is Error Rate:
      - E_{18} to E_{1}: E_{18} to E_{1}, in single steps
    - However, `<top>` must be greater than `<bottom>`.
    - When the measurement item is Error Count, EI:
      - E_{0} to E_{17}: E_{0} to E_{17}, in single steps
      - However, `<top>` must be greater than `<bottom>`.

**Function**
Sets the vertical axis scale of the histogram.

**Example**
To set the vertical axis scale of the histogram for Error Rate as top: E_{-3}
and bottom: E_{-4}:
```
> :DISPlay:RESult:HISTogram:SCALe E_{-3},E_{-4}
```

**Compatibility**
Incompatible with existing models.
### :DISPLAY:RESult:HISTogram:SCALe?

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Response</td>
<td><code>&lt;top&gt; = &lt;CHARACTER RESPONSE DATA&gt;</code></td>
</tr>
<tr>
<td></td>
<td><code>&lt;bottom&gt; = &lt;CHARACTER RESPONSE DATA&gt;</code></td>
</tr>
<tr>
<td>Function</td>
<td>Queries the vertical axis scale of the histogram.</td>
</tr>
<tr>
<td>Example</td>
<td><code>&gt; :DISPLAY:RESult:HISTogram:SCALe?</code></td>
</tr>
<tr>
<td></td>
<td><code>&lt; E_3.E_4</code></td>
</tr>
<tr>
<td>Compatibility</td>
<td>Incompatible with existing models.</td>
</tr>
</tbody>
</table>

### :DISPLAY:RESult:HISTogram:TYPE `<type>`

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;type&gt;</code></td>
<td>= &lt;CHARACTER PROGRAM DATA&gt;</td>
</tr>
<tr>
<td>ECOunt</td>
<td>Counts errors.</td>
</tr>
<tr>
<td>ERATe</td>
<td>Counts error rate.</td>
</tr>
<tr>
<td>EI</td>
<td>Counts error interval.</td>
</tr>
<tr>
<td>Function</td>
<td>Sets the measurement item counting type displayed in the histogram.</td>
</tr>
<tr>
<td>Example</td>
<td>To set the measurement item counting type displayed in the histogram to ECOunt (counting errors).</td>
</tr>
<tr>
<td></td>
<td><code>&gt; :DISPLAY:RESult:HISTogram:TYPE ECOunt</code></td>
</tr>
<tr>
<td>Compatibility</td>
<td>Incompatible with existing models.</td>
</tr>
</tbody>
</table>

### :DISPLAY:RESult:HISTogram:TYPE?

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Response</td>
<td><code>&lt;type&gt; = &lt;CHARACTER RESPONSE DATA&gt;</code></td>
</tr>
<tr>
<td>ECO, ERAT, EI</td>
<td></td>
</tr>
<tr>
<td>Function</td>
<td>Queries the measurement item counting type displayed in the histogram.</td>
</tr>
<tr>
<td>Example</td>
<td><code>&gt; :DISPLAY:RESult:HISTogram:TYPE?</code></td>
</tr>
<tr>
<td></td>
<td><code>&lt; ECO</code></td>
</tr>
<tr>
<td>Compatibility</td>
<td>Incompatible with existing models.</td>
</tr>
</tbody>
</table>
Chapter 7  SCPI Commands

**:DISPlay:RESult:HISTogram:TIMescale <day>,<hour>,<min>,<second>*

Parameter

<day> = <DECIMAL NUMERIC PROGRAM DATA>
0 to 99  0 to 99 days, in 1-day steps
<hour> = <DECIMAL NUMERIC PROGRAM DATA>
0 to 23  0 to 23 hours, in 1-hour step
<min> = <DECIMAL NUMERIC PROGRAM DATA>
0 to 59  0 to 59 minutes, in 1-minute step
<second> = <DECIMAL NUMERIC PROGRAM DATA>
0 to 59  0 to 59 seconds, in 1-second step

The minimum setting value is 0, 0, 0, 30.

Function
Sets the histogram display time scale.

Example
To set the histogram display time scale to "0 days, 1 hour, 0 minutes, 0 seconds":
> :DISPlay:RESult:HISTogram:TIMescale 0,1,0,0

Compatibility
Incompatible with existing models.

**:DISPlay:RESult:HISTogram:TIMescale?*

Response

<day> = <NR1 NUMERIC RESPONSE DATA>
0 to 99  0 to 99 days
<hour> = <NR1 NUMERIC RESPONSE DATA>
0 to 23  0 to 23 hours
<min> = <NR1 NUMERIC RESPONSE DATA>
0 to 59  0 to 59 minutes
<second> = <NR1 NUMERIC RESPONSE DATA>
0 to 59  0 to 59 seconds

Function
Queries the histogram display time scale.

Example

> :DISPlay:RESult:HISTogram:TIMescale?
< 0,1,0,0

Compatibility
Incompatible with existing models.

**:DISPlay:RESult:HISTogram:MARKer <position>*

Parameter

<position> = <CHARACTER PROGRAM DATA>
PRE  Searches for error alarms before the current marker position.
POST  Searches for error alarms after the current marker position.

Function
Searches for error alarms before/after the current marker position.

Example
To search for error alarms before the current marker position:
> :DISPlay:RESult:HISTogram:MARKer  PRE

Compatibility
Incompatible with existing models.
Function
Queries the marker setting time, alarm, and measurement results.

Example
> :DISPlay:RESult:HISTogram:MARKer?
< 0,0,1,0,CLOS, "0"

Compatibility
Incompatible with existing models.
**Chapter 7  SCPI Commands**

### :DISPlay:RESult:HISTogram:RESolution <resolution>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;resolution&gt; = &lt;CHARACTER PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
<td>1 s</td>
</tr>
<tr>
<td>S10</td>
<td>10 s</td>
</tr>
<tr>
<td>S30</td>
<td>30 s</td>
</tr>
<tr>
<td>M1</td>
<td>1 min</td>
</tr>
<tr>
<td>M10</td>
<td>10 min</td>
</tr>
<tr>
<td>M30</td>
<td>30 min</td>
</tr>
<tr>
<td>H1</td>
<td>1 hour</td>
</tr>
</tbody>
</table>

**Function**
Sets the histogram calculation resolution.

**Example**
To set the histogram calculation resolution to 1 second:
> :DISPlay:RESult:HISTogram:RESolution S1

**Compatibility**
Incompatible with existing models.

### :DISPlay:RESult:HISTogram:RESolution?  

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;resolution&gt; = &lt;CHARACTER RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1, S10, S30, M1, M10, M30, H1</td>
<td></td>
</tr>
</tbody>
</table>

**Function**
Queries the histogram calculation resolution.

**Example**
> :DISPlay:RESult:HISTogram:RESolution?
< S1

**Compatibility**
Incompatible with existing models.

### :DISPlay:RESult:HISTogram:DISPtime <time>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;time&gt; = &lt;CHARACTER PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
<td>1 s</td>
</tr>
<tr>
<td>S10</td>
<td>10 s</td>
</tr>
<tr>
<td>S30</td>
<td>30 s</td>
</tr>
<tr>
<td>M1</td>
<td>1 min</td>
</tr>
<tr>
<td>M10</td>
<td>10 min</td>
</tr>
<tr>
<td>M30</td>
<td>30 min</td>
</tr>
<tr>
<td>H1</td>
<td>1 hour</td>
</tr>
</tbody>
</table>

**Function**
Sets the histogram display resolution.

**Example**
To set the histogram display resolution to 1 second:
> :DISPlay:RESult:HISTogram:DISPtime S1

**Compatibility**
Incompatible with existing models.
### :DISPlay:RESult:HISTogram:DISPtime?

<table>
<thead>
<tr>
<th><strong>Response</strong></th>
<th>(&lt;\text{time}&gt; = \langle\text{CHARACTER RESPONSE DATA}\rangle)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>S1, S10, S30, M1, M10, M30, H1</td>
</tr>
</tbody>
</table>

**Function**: Queries the histogram display resolution.

**Example**

```plaintext
> :DISPlay:RESult:HISTogram:DISPtime?
< S1
```

**Compatibility**: Incompatible with existing models.

### :DISPlay:RESult:HISTogram:EDETect <mode>

<table>
<thead>
<tr>
<th><strong>Parameter</strong></th>
<th>(&lt;\text{mode}&gt; = \langle\text{CHARACTER PROGRAM DATA}\rangle)</th>
</tr>
</thead>
<tbody>
<tr>
<td>TOT</td>
<td>Total Error</td>
</tr>
<tr>
<td>INS</td>
<td>Insertion Error</td>
</tr>
<tr>
<td>OMI</td>
<td>Omission Error</td>
</tr>
<tr>
<td>TRANSition</td>
<td>Transition Error</td>
</tr>
<tr>
<td>NONTransition</td>
<td>Non Transition Error</td>
</tr>
</tbody>
</table>

**Function**: Sets the bit error alarm measurement method and error detection method for histogram calculation.

**Example**

To set the bit error alarm measurement method and error detection method for histogram calculation to Total Error:

```plaintext
> :DISPlay:RESult:HISTogram:EDETect TOTal
```

**Compatibility**: Incompatible with existing models.

### :DISPlay:RESult:HISTogram:EDETect?

<table>
<thead>
<tr>
<th><strong>Response</strong></th>
<th>(&lt;\text{mode}&gt; = \langle\text{CHARACTER RESPONSE DATA}\rangle)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>TOT, INS, OMI, TRANS, NONT</td>
</tr>
</tbody>
</table>

**Function**: Queries the bit error alarm measurement method and error detection method for histogram calculation.

**Example**

```plaintext
> :DISPlay:RESult:HISTogram:EDETect?
< TOT
```

**Compatibility**: Incompatible with existing models.
Chapter 7  SCPI Commands

:CALCulate:DATA:HISTogram? <day>,<hour>,<min>,<second>

Parameter

- `<day>` = `<DECIMAL NUMERIC PROGRAM DATA>`
  0 to 99 0 to 99 days, in 1-day steps
- `<hour>` = `<DECIMAL NUMERIC PROGRAM DATA>`
  0 to 23 0 to 23 hours, in 1-hour steps
- `<min>` = `<DECIMAL NUMERIC PROGRAM DATA>`
  0 to 59 0 to 59 minutes, in 1-minute steps
- `<second>` = `<DECIMAL NUMERIC PROGRAM DATA>`
  0 to 59 0 to 59 seconds, in 1-second steps

Response

- `<alarm>` = `<CHARACTER RESPONSE DATA>`
  CLOS Clock LOS has occurred.
  SLOS Sync LOS has occurred.
  CRUN CR Unlock has occurred.
  NONE No alarm has occurred.
- `<count>` = `<STRING RESPONSE DATA>`
- `<rate>` = `<STRING RESPONSE DATA>`

<table>
<thead>
<tr>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;XXXXXXX&quot;</td>
<td>For 0 to 9999999</td>
</tr>
<tr>
<td>&quot;X.XXXE0X&quot;</td>
<td>For 1.0000E07 to 9.9999E17</td>
</tr>
<tr>
<td>&quot;-------&quot;</td>
<td>When no data corresponds to the query</td>
</tr>
<tr>
<td>&quot;X.XXXE-XX&quot;</td>
<td>For 0.0001E–18 to 1.0000E00</td>
</tr>
<tr>
<td>&quot;-------&quot;</td>
<td>When no data corresponds to the query</td>
</tr>
</tbody>
</table>

Function

Queries the measurement item on the histogram corresponding to the parameter.

Example

> :CALCulate:DATA:HISTogram? 0,1,0,0
< CLOS, "0", "0.0001E-18"

Compatibility

Incompatible with existing models.
7.4.2 Commands related to Measurement tab

![Measurement tab diagram]

Figure 7.4.2-1 Measurement tab
7.4.2.1 Gating setting commands

Figure 7.4.2.1-1  Gating setting (Time)

Figure 7.4.2.1-2  Gating setting (Clock Count)

Figure 7.4.2.1-3  Gating setting (Error Count)

Figure 7.4.2.1-4  Gating setting (Block Count)
### Table 7.4.2.1-1 Gating setting commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1]</td>
<td>Unit</td>
<td>:SENSe:MEASure:EALarm:UNIT</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:MEASure:EALarm:UNIT?</td>
</tr>
<tr>
<td>[2]</td>
<td>No label (Measurement time setting)</td>
<td>:SENSe:MEASure:EALarm:PERiod</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:MEASure:EALarm:PERiod?</td>
</tr>
<tr>
<td>[3]</td>
<td>No label (Clock count setting)</td>
<td>:SENSe:MEASure:EALarm:CLOCkcnt</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:MEASure:EALarm:CLOCkcnt?</td>
</tr>
<tr>
<td>[4]</td>
<td>No label (Error count setting)</td>
<td>:SENSe:MEASure:EALarm:ERRorcnt</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:MEASure:EALarm:ERRorcnt?</td>
</tr>
<tr>
<td>[5]</td>
<td>No label (Block count setting)</td>
<td>:SENSe:MEASure:EALarm:BLOCkcnt</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:MEASure:EALarm:BLOCkcnt?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:MEASure:EALarm:MODE?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:DISPlay:RESult:EALarm:MODE?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:DISPlay:RESult:EALarm:CALCuration?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:DISPlay:RESult:EALarm:INTerval?</td>
</tr>
</tbody>
</table>

### :SENSe:MEASure:EALarm:UNIT <unit>

**Parameter**  
<unit> = <CHARACTER PROGRAM DATA>
- TIME = Time
- CLOck = Clock Count
- ERRor = Error Count
- BLOck = Block Count

**Function**  
Sets the measurement period unit.

**Example**  
To set the measurement period unit to Time.
> :SENSe:MEASure:EALarm:UNIT TIME

**Compatibility**  
Incompatible with existing models.

### :SENSe:MEASure:EALarm:UNIT? 

**Response**  
<unit> = <CHARACTER RESPONSE DATA>
- TIME, CLOC, ERR, BLOC

**Function**  
Queries the measurement period unit.

**Example**  
> :SENSe:MEASure:EALarm:UNIT?
< TIME

**Compatibility**  
Incompatible with existing models.
### :SENSe:MEASure:EALarm:PERiod \(<day>,<hour>,<min>,<second>\)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(&lt;day&gt;,&lt;hour&gt;,&lt;min&gt;,&lt;second&gt;) = (&lt;\text{DECIMAL NUMERIC PROGRAM DATA}&gt;)</td>
<td></td>
</tr>
<tr>
<td>(&lt;\text{day}&gt;)</td>
<td>0 to 99</td>
</tr>
<tr>
<td>(&lt;\text{hour}&gt;)</td>
<td>0 to 23</td>
</tr>
<tr>
<td>(&lt;\text{min}&gt;)</td>
<td>0 to 59</td>
</tr>
<tr>
<td>(&lt;\text{second}&gt;)</td>
<td>0 to 59</td>
</tr>
</tbody>
</table>

**Function**
Sets the measurement time when the measurement period is set to Time.

**Example**
To set the measurement time when the measurement period is set to Time, to "0 days, 0 hours, 1 minute, 0 seconds":

\>`:SENSe:MEASure:EALarm:PERiod 0,0,1,0

**Compatibility**
Compatible with the MP1632C and MP1776A.

### :SENSe:MEASure:EALarm:PERiod?

<table>
<thead>
<tr>
<th>Response</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(&lt;\text{day},&lt;\text{hour},&lt;\text{min},&lt;\text{second}&gt;) = (&lt;\text{NR1 NUMERIC RESPONSE DATA}&gt;)</td>
<td></td>
</tr>
<tr>
<td>(&lt;\text{day}&gt;)</td>
<td>0 to 99</td>
</tr>
<tr>
<td>(&lt;\text{hour}&gt;)</td>
<td>0 to 23</td>
</tr>
<tr>
<td>(&lt;\text{min}&gt;)</td>
<td>0 to 59</td>
</tr>
<tr>
<td>(&lt;\text{second}&gt;)</td>
<td>0 to 59</td>
</tr>
</tbody>
</table>

**Function**
Queries the measurement time when the measurement period is set to Time.

**Example**

\>`:SENSe:MEASure:EALarm:PERiod?\n
\< 0,0,1,0

**Compatibility**
Compatible with the MP1632C and MP1776A.
### :SENSe:MEASure:EALarm:CLOCkcnt <count>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;count&gt; = &lt;CHARACTER PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>E_4</td>
<td>&gt;1E+4</td>
</tr>
<tr>
<td>E_5</td>
<td>&gt;1E+5</td>
</tr>
<tr>
<td>E_6</td>
<td>&gt;1E+6</td>
</tr>
<tr>
<td>E_7</td>
<td>&gt;1E+7</td>
</tr>
<tr>
<td>E_8</td>
<td>&gt;1E+8</td>
</tr>
<tr>
<td>E_9</td>
<td>&gt;1E+9</td>
</tr>
<tr>
<td>E_10</td>
<td>&gt;1E+10</td>
</tr>
<tr>
<td>E_11</td>
<td>&gt;1E+11</td>
</tr>
<tr>
<td>E_12</td>
<td>&gt;1E+12</td>
</tr>
<tr>
<td>E_13</td>
<td>&gt;1E+13</td>
</tr>
<tr>
<td>E_14</td>
<td>&gt;1E+14</td>
</tr>
<tr>
<td>E_15</td>
<td>&gt;1E+15</td>
</tr>
<tr>
<td>E_16</td>
<td>&gt;1E+16</td>
</tr>
</tbody>
</table>

**Function**
Sets the clock count threshold when the measurement period is set to Clock Count.

**Example**
To set the clock count threshold when the measurement period is set to Clock Count to 1E+4:
> :SENSe:MEASure:EALarm:CLOCkcnt E_4

**Compatibility**
Incompatible with existing models.

### :SENSe:MEASure:EALarm:CLOCkcnt?

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;count&gt; = &lt;CHARACTER RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>E_4, E_5, E_6, E_7, E_8, E_9, E_10, E_11, E_12, E_13, E_14, E_15, E_16</td>
<td></td>
</tr>
</tbody>
</table>

**Function**
Queries the clock count threshold when the measurement period is set to Clock Count.

**Example**
> :SENSe:MEASure:EALarm:CLOCkcnt?
< E_4

**Compatibility**
Incompatible with existing models.
### :SENSe:MEASure:EALarm:ERRorcnt \(<count>\)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>(&lt;count&gt;) = (&lt;CHARACTER PROGRAM DATA&gt;)</th>
</tr>
</thead>
<tbody>
<tr>
<td>E_4</td>
<td>(&gt;1E+4)</td>
</tr>
<tr>
<td>E_5</td>
<td>(&gt;1E+5)</td>
</tr>
<tr>
<td>E_6</td>
<td>(&gt;1E+6)</td>
</tr>
<tr>
<td>E_7</td>
<td>(&gt;1E+7)</td>
</tr>
<tr>
<td>E_8</td>
<td>(&gt;1E+8)</td>
</tr>
<tr>
<td>E_9</td>
<td>(&gt;1E+9)</td>
</tr>
<tr>
<td>E_10</td>
<td>(&gt;1E+10)</td>
</tr>
<tr>
<td>E_11</td>
<td>(&gt;1E+11)</td>
</tr>
<tr>
<td>E_12</td>
<td>(&gt;1E+12)</td>
</tr>
<tr>
<td>E_13</td>
<td>(&gt;1E+13)</td>
</tr>
<tr>
<td>E_14</td>
<td>(&gt;1E+14)</td>
</tr>
<tr>
<td>E_15</td>
<td>(&gt;1E+15)</td>
</tr>
<tr>
<td>E_16</td>
<td>(&gt;1E+16)</td>
</tr>
</tbody>
</table>

**Function**
Sets the error count threshold when the measurement period is set to Error Count.

**Example**
To set the error count threshold when the measurement period is set to Error Count to \(1E+4\):
> :SENSe:MEASure:EALarm:ERRorcnt E_4

**Compatibility**
Incompatible with existing models.

### :SENSe:MEASure:EALarm:ERRorcnt?

<table>
<thead>
<tr>
<th>Response</th>
<th>(&lt;count&gt;) = (&lt;CHARACTER RESPONSE DATA&gt;)</th>
</tr>
</thead>
<tbody>
<tr>
<td>E_4, E_5, E_6, E_7, E_8, E_9, E_10, E_11, E_12, E_13, E_14, E_15, E_16</td>
<td></td>
</tr>
</tbody>
</table>

**Function**
Queries the error count threshold when the measurement period is set to Error Count.

**Example**
> :SENSe:MEASure:EALarm:ERRorcnt?
< E_4

**Compatibility**
Incompatible with existing models.
:SENSe:MEASure:EALarm:BLOCkcnt <count>

Parameter

<table>
<thead>
<tr>
<th>&lt;count&gt;</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>E_2</td>
<td>&gt;1E+2</td>
</tr>
<tr>
<td>E_3</td>
<td>&gt;1E+3</td>
</tr>
<tr>
<td>E_4</td>
<td>&gt;1E+4</td>
</tr>
<tr>
<td>E_5</td>
<td>&gt;1E+5</td>
</tr>
<tr>
<td>E_6</td>
<td>&gt;1E+6</td>
</tr>
<tr>
<td>E_7</td>
<td>&gt;1E+7</td>
</tr>
<tr>
<td>E_8</td>
<td>&gt;1E+8</td>
</tr>
<tr>
<td>E_9</td>
<td>&gt;1E+9</td>
</tr>
<tr>
<td>E_10</td>
<td>&gt;1E+10</td>
</tr>
<tr>
<td>E_11</td>
<td>&gt;1E+11</td>
</tr>
<tr>
<td>E_12</td>
<td>&gt;1E+12</td>
</tr>
<tr>
<td>E_13</td>
<td>&gt;1E+13</td>
</tr>
<tr>
<td>E_14</td>
<td>&gt;1E+14</td>
</tr>
</tbody>
</table>

Function
Sets the block count threshold when the measurement period is set to Block Count.

Example
To set the block count threshold when the measurement period is set to Block Count to 1E+4:
> :SENSe:MEASure:EALarm:BLOCkcnt E_4

Compatibility
Incompatible with existing models.

:SENSe:MEASure:EALarm:BLOCkcnt?

Response

<table>
<thead>
<tr>
<th>&lt;count&gt;</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>E_2, E_3, E_4, E_5, E_6, E_7, E_8, E_9, E_10, E_11, E_12, E_13, E_14</td>
<td></td>
</tr>
</tbody>
</table>

Function
Queries the block count threshold when the measurement period is set to Block Count.

Example
> :SENSe:MEASure:EALarm:BLOCkcnt?
< E_4

Compatibility
Incompatible with existing models.
Chapter 7  SCPI Commands

:SENSe:MEASure:EALarm:MODE <mode>

Parameter  

- `<mode> = <CHARACTER PROGRAM DATA>`
- REPeat: Performs measurement repeatedly.
- SINGle: Performs measurement once.
- UNTimed: Continues measurement until the measurement end is instructed.

Function

Sets the bit error alarm measurement processing mode.

Example

To set the bit error alarm measurement processing mode to REPeat (performing measurement repeatedly):

```
> :SENSe:MEASure:EALarm:MODE REPeat
```

Compatibility

Compatible with the MP1632C and MP1776A.

:SENSe:MEASure:EALarm:MODE?

Response  

- `<mode> = <CHARACTER RESPONSE DATA>`
- REP, SING, UNT

Function

Queries the bit error alarm measurement processing mode.

Example

```
> :SENSe:MEASure:EALarm:MODE?
< REP
```

Compatibility

Compatible with the MP1632C and MP1776A.

:DISPlay:RESult:EALarm:MODE <boolean>

Parameter  

- `<boolean> = <BOOLEAN PROGRAM DATA>`
- 0 or OFF: Measurement result data display immediate update OFF
- 1 or ON: Measurement result data display immediate update ON

Function

Sets measurement result data display immediate update ON/OFF.

Example

To set measurement result data display immediate update ON:

```
> :DISPlay:RESult:EALarm:MODE ON
```

Compatibility

Incompatible with existing models.

:DISPlay:RESult:EALarm:MODE?

Response  

- `<numeric> = <NR1 NUMERIC RESPONSE DATA>`
- 0: OFF
- 1: ON

Function

Queries the measurement result data display immediate update ON/OFF state.

Example

```
> :DISPlay:RESult:EALarm:MODE?
< 1
```

Compatibility

Incompatible with existing models.
### :DISPlay:RESult:EALarm:CALCuration <mode>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;mode&gt; = &lt;CHARACTER PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PROGressive: Progressive accumulation</td>
</tr>
<tr>
<td></td>
<td>IMMediate: Immediate processing</td>
</tr>
</tbody>
</table>

**Function**
Sets the immediate data calculation method.

**Example**
To set the immediate data calculation method to progressive accumulation:

```
> :DISPlay:RESult:EALarm:CALCuration PROGressive
```

**Compatibility**
Incompatible with existing models.

### :DISPlay:RESult:EALarm:CALCuration? 

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;mode&gt; = &lt;CHARACTER RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PROG, IMM</td>
</tr>
</tbody>
</table>

**Function**
Queries the immediate data calculation method.

**Example**

```
> :DISPlay:RESult:EALarm:CALCuration?
< PROG
```

**Compatibility**
Incompatible with existing models.

### :DISPlay:RESult:EALarm:INTerval <interval>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;interval&gt; = &lt;CHARACTER PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>I100: Updates display in 100-ms period.</td>
</tr>
<tr>
<td></td>
<td>I200: Updates display in 200-ms period.</td>
</tr>
</tbody>
</table>

**Function**
Sets the immediate data display update period.

**Example**
To set the immediate data display update period to 100 ms:

```
> :DISPlay:RESult:EALarm:INTerval I100
```

**Compatibility**
Incompatible with existing models.

### :DISPlay:RESult:EALarm:INTerval? 

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;interval&gt; = &lt;CHARACTER RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>I100, I200</td>
</tr>
</tbody>
</table>

**Function**
Queries the immediate data display update period.

**Example**

```
> :DISPlay:RESult:EALarm:INTerval?
< I100
```

**Compatibility**
Incompatible with existing models.
7.4.2.2 Auto Sync setting commands

![Figure 7.4.2.2-1 Auto Sync setting field](image)

![Table 7.4.2.2-1 Auto Sync setting commands](table)

### :SENSe:PATTern:SYNC:ASYNc <boolean>

**Parameter**

\[
<\text{boolean}> = <\text{BOOLEAN PROGRAM DATA}>
\]

0 or OFF Auto Sync OFF
1 or ON Auto Sync ON

**Function**

Sets whether to automatically perform resynchronization (Auto Sync) when the synchronization threshold is exceeded.

**Example**

To enable automatic resynchronization (Auto Sync ON):

\[
> :\text{SENSe}\text{PATTern}:\text{SYNC}:\text{ASYNc} \text{ ON}
\]

**Compatibility**

Compatible with the MP1632C and MP1776A.

### :SENSe:PATTern:SYNC:ASYNc?

**Response**

\[
<\text{numeric}> = <\text{NR1 NUMERIC RESPONSE DATA}>
\]

0 Auto Sync OFF
1 Auto Sync ON

**Function**

Queries whether to automatically perform resynchronization (Auto Sync) when the synchronization threshold is exceeded.

**Example**

\[
> :\text{SENSe}\text{PATTern}:\text{SYNC}:\text{ASYNc}?
< 1
\]

**Compatibility**

Compatible with the MP1632C and MP1776A.
### :SENSe:PATTern:SYNC:THReshold <thre>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;thre&gt; = &lt;CHARACTER PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>INT</td>
<td>Internal</td>
</tr>
<tr>
<td>E_2</td>
<td>E-2</td>
</tr>
<tr>
<td>E_3</td>
<td>E-3</td>
</tr>
<tr>
<td>E_4</td>
<td>E-4</td>
</tr>
<tr>
<td>E_5</td>
<td>E-5</td>
</tr>
<tr>
<td>E_6</td>
<td>E-6</td>
</tr>
<tr>
<td>E_7</td>
<td>E-7</td>
</tr>
<tr>
<td>E_8</td>
<td>E-8</td>
</tr>
</tbody>
</table>

**Function**
Sets the synchronization detection threshold for resynchronization.

**Example**
To set the synchronization detection threshold for resynchronization to Internal:

```plaintext
> :SENSe:PATTern:SYNC:THReshold INT
```

**Compatibility**
Compatible with the MP1776A.

### :SENSe:PATTern:SYNC:THReshold?

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;thre&gt; = &lt;CHARACTER RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>INT, E_2, E_3, E_4, E_5, E_6, E_7, E_8</td>
<td></td>
</tr>
</tbody>
</table>

**Function**
Queries the synchronization detection threshold for resynchronization.

**Example**

```plaintext
> :SENSe:PATTern:SYNC:THReshold?
< INT
```

**Compatibility**
Compatible with the MP1776A.
7.4.2.3 Sync Control setting commands

Figure 7.4.2.3-1  Sync Control setting field

Figure 7.4.2.3-2  Sync Control setting (Pattern Editor screen)
### 7.4 ED Commands

#### Table 7.4.2.3-1  Sync Control setting commands

<table>
<thead>
<tr>
<th>No</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>:SENSe:PATTern:SYNC:PSMode?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:PATTern:SYNC:FLENgth?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:PATTern:SYNC:FPOSition?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:PATTern:SYNC:DATA:WHOLe?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:PATTern:SYNC:PROGram:FILL</td>
</tr>
</tbody>
</table>

**:SENSe:PATTern:SYNC:PSMode <mode>**

Parameter  

<mode> = <CHARACTER PROGRAM DATA>  
FRAMe  
NORMal  
QUICk  
FAST  

Function  
Sets the synchronization mode for the test pattern.  
Example  
To set the synchronization mode for the test pattern to frame detection ON:  
> :SENSe:PATTern:SYNC:PSMode FRAMe  
Compatibility  
Compatible with the MP1632C and MP1776A.

**:SENSe:PATTern:SYNC:PSMode?**

Response  
<mode> = <CHARACTER RESPONSE DATA>  
FRAM, NORM, QUIC, FAST  

Function  
Queries the synchronization mode for the test pattern.  
Example  
> :SENSe:PATTern:SYNC:PSMode?  
< FRAM  
Compatibility  
Compatible with the MP1632C and MP1776A.
Chapter 7  SCPI Commands

:SENSe:PATTern:SYNC:FLENgth <length>

Parameter  

<length> = <DECIMAL NUMERIC PROGRAM DATA>

4 to 64  4 to 64 bits, in 4-bit steps

Function  

Sets the unique pattern length when the synchronization mode is set to FRAME (frame detection ON).

Example  

To set the unique pattern length when the synchronization mode is set to FRAME (frame detection ON) to 4 bits:

> :SENSe:PATTern:SYNC:FLENgth 4

Compatibility  

Compatible with the MP1632C and the MP1776A.

:SENSe:PATTern:SYNC:FLENgth?

Response  

<length> = <NR1 NUMERIC RESPONSE DATA>

Function  

Queries the unique pattern length when the synchronization mode is set to FRAME (frame detection ON).

Example  

> :SENSe:PATTern:SYNC:FLENgth?

< 4

Compatibility  

Compatible with the MP1632C and the MP1776A.

:SENSe:PATTern:SYNC:FPOSition <length>

Parameter  

<length> = <DECIMAL NUMERIC PROGRAM DATA>

1 to 134217725  1 to 134,217,725 bits, in 1-bit steps

The maximum setting value is "Pattern length – Frame length + 1" bits.

In the case of Combination, the setting range and resolution (steps) are multiplied by 4.

Function  

Sets the frame synchronization start position when the synchronization mode is set to FRAME (frame detection ON).

Example  

To set the frame synchronization start position when the synchronization mode is set to FRAME (frame detection ON) to bit 65:

> :SENSe:PATTern:SYNC:FPOSition 65

Compatibility  

Incompatible with existing models.

:SENSe:PATTern:SYNC:FPOSition?

Response  

<length> = <NR1 NUMERIC RESPONSE DATA>

Function  

Queries the frame synchronization start position when the synchronization mode is set to FRAME (frame detection ON).

Example  

> :SENSe:PATTern:SYNC:FPOSition?

< 17

Compatibility  

Incompatible with existing models.
### :SENSe:PATTern:SYNC:DATA:WHOLe <data>

**Parameter**

<data> = <STRING PROGRAM DATA>

"H***...*" 1 to 16 characters

(Specify pattern data in hexadecimal format)

"B***...*" 1 to 64 characters

(Specify pattern data in binary format)

**Function**

Edits the mask pattern when the synchronization mode is set to FRAME (frame detection ON). Up to 64-bit data can be edited in hexadecimal or binary format.

**Example**

To set the mask pattern when the synchronization mode is set to FRAME (frame detection ON) to one hexadecimal character:

```
> :SENSe:PATTern:SYNC:DATA:WHOLe "H001"
```

**Compatibility**

Incompatible with existing models.

### :SENSe:PATTern:SYNC:DATA:WHOLe?

**Response**

<data> = <STRING RESPONSE DATA>

"H***...*" 1 to 16 characters

(Specify pattern data in hexadecimal format)

Hexadecimal data of up to 64 bits is returned.

**Function**

Queries the mask pattern when the synchronization mode is set to FRAME (frame detection ON).

**Example**

```
> :SENSe:PATTern:SYNC:DATA:WHOLe?
< "H001"
```

**Compatibility**

Incompatible with existing models.

### :SENSe:PATTern:SYNC:DREVerse:ADDRes<start>,<end>

**Parameter**

<start> = <NON-DECIMAL PROGRAM DATA>

#H0 to #H3F 0 to 3F bits, in 1-bit steps

(Specify in hexadecimal)

<end> = <NON-DECIMAL PROGRAM DATA>

#H0 to #H3F 0 to 3F bits, in 1-bit steps

(Specify in hexadecimal)

**Function**

Inverts the mask pattern data when the synchronization mode is set to FRAME (frame detection ON). Specify the inversion range by the <start> and <end> addresses.

**Example**

To invert addresses from bit 1 to bit 17 of the mask pattern data when the synchronization mode is set to FRAME (frame detection ON):

```
> :SENSe:PATTern:SYNC:DREVerse:ADDRes #H001,#H011
```

**Compatibility**

Incompatible with existing models.
### :SENSe:PATTern:SYNC:PROGram:FILL <start>,<end>,<data>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;start&gt;</td>
<td>&lt;NON-DECIMAL PROGRAM DATA&gt;</td>
</tr>
<tr>
<td></td>
<td>#H0 to #H3F 0 to 3F bits, in 1-bit steps</td>
</tr>
<tr>
<td></td>
<td>(Specify in hexadecimal)</td>
</tr>
<tr>
<td>&lt;end&gt;</td>
<td>&lt;NON-DECIMAL PROGRAM DATA&gt;</td>
</tr>
<tr>
<td></td>
<td>#H0 to #H3F 0 to 3F bits, in 1-bit steps</td>
</tr>
<tr>
<td></td>
<td>(Specify in hexadecimal)</td>
</tr>
<tr>
<td>&lt;data&gt;</td>
<td>&lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
</tr>
<tr>
<td></td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Fills the specified range with 0s.</td>
</tr>
<tr>
<td></td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Fills the specified range with 1s.</td>
</tr>
</tbody>
</table>

**Function**
Sets 0s or 1s for the specified range of the mask pattern data when the synchronization mode is set to FRAMe (frame detection ON). Specify the range with the <start> and <end> addresses.

**Example**
To set 0s for bit 1 to bit 17 of the mask pattern when the synchronization mode is set to FRAMe (frame detection ON):

```
> :SENSe:PATTern:SYNC:PROGram:FILL #H001, #H011, 0
```
7.4.2.4 Error/Alarm Condition setting commands

Table 7.4.2.4-1 Error/Alarm Condition setting commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>:CALCulate:EALarm:ERRor:EDETect?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:CALCulate:EALarm:ERRor:INTerval?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:CALCulate:EALarm:PERFormance:THReshold?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:CALCulate:EALarm:CLEValuation?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:CALCulate:EALarm:CRUNlock?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:CALCulate:EALarm:SLEValuation?</td>
</tr>
</tbody>
</table>

Figure 7.4.2.4-1 Error/Alarm Condition setting field

Figure 7.4.2.4-2 Error/Alarm Condition setting field (For Recovered Clock)
Chapter 7  SCPI Commands

:CALCulate:EALarm:ERRor:EDETect <mode>

Parameter  
<mode> = <CHARACTER PROGRAM DATA>
  INSomi  Detects INS/OMI error.
  TRANsition  Detects Transition/Non Transition error.

Function  
Sets the error detection method during bit error alarm measurement.

Example  
To set the error detection method during bit error alarm measurement to INS/OMI error detection:
  > :CALCulate:EALarm:ERRor:EDETect INSomi

Compatibility  
Incompatible with existing models.

:CALCulate:EALarm:ERRor:EDETect?

Response  
<mode> = <CHARACTER RESPONSE DATA>
  INS, TRAN

Function  
Queries the error detection method during bit error alarm measurement.

Example  
> :CALCulate:EALarm:ERRor:EDETect?
  < INS

Compatibility  
Incompatible with existing models.

:CALCulate:EALarm:ERRor:INTerval <numeric>,<suffix>

Parameter  
<numeric> = <DECIMAL NUMERIC PROGRAM DATA>
  1  Sets the measurement time to 1.
  10  Sets the measurement time to 10.
  100  Sets the measurement time to 100.

<suffix> = <CHARACTER PROGRAM DATA>
  ms  Sets the measurement time unit to millisecond.
  s  Sets the measurement time unit to second.

Function  
Sets the interval for EI and %EFI measurement.
  1 ms, 10 ms, 100 ms, or 1s can be set.

Example  
To set the interval for EI and %EFI measurement to 1 ms:
  > :CALCulate:EALarm:ERRor:INTerval 1,ms

Compatibility  
Compatible with the MP1632C and MP1776A.
### 7.4 ED Commands

**:CALCulate:EALarm:ERRor:INTerval?**

| Response | <numeric> = <NR1 NUMERIC RESPONSE DATA>  
| 1, 10, 100  
| <suffix> = <CHARACTER RESPONSE DATA>  
| ms, s |
| Function | Queries the interval for EI and %EFI measurement. |
| Example | > :CALCulate:EALarm:ERRor:INTerval?  
| < 1, ms |
| Compatibility | Compatible with the MP1632C and MP1776A. |

**:CALCulate:EALarm:PERFormance:THReshold <thre>**

| Parameter | <thre> = <STRING PROGRAM DATA>  
| "SES_3:DM_6" SES:1E–3, DM:1E–6  
| "SES_4:DM_8" SES:1E–4, DM:1E–8 |
| Function | Sets the thresholds for SES and DM calculations during performance measurement. |
| Example | To set the thresholds for SES and DM calculations during performance measurement to SES: 1E–3, DM: 1E–6:  
| > :CALCulate:EALarm:PERFormance:THReshold "SES_3:DM_6" |
| Compatibility | Compatible with the MP1632C and MP1776A. |

**:CALCulate:EALarm:PERFormance:THReshold?**

| Response | <thre> = <STRING RESPONSE DATA>  
| "SES_3:DM_6", "SES_4:DM_8" |
| Function | Queries the thresholds for SES and DM calculations during performance measurement. |
| Example | > :CALCulate:EALarm:PERFormance:THReshold?  
| < "SES_3:DM_6" |
| Compatibility | Compatible with the MP1632C and MP1776A. |

**:CALCulate:EALarm:CLEValuation <boolean>**

| Parameter | <boolean> = <BOOLEAN PROGRAM DATA>  
| 0 or OFF  
| 1 or ON  
| Does not set to the evaluation target.  
| Sets to the evaluation target. |
| Function | Sets whether to include the Clock Loss occurrence period to the measurement evaluation target. |
| Example | To include the Clock Loss occurrence period to the measurement evaluation target:  
| > :CALCulate:EALarm:CLEValuation ON |
| Compatibility | Compatible with the MP1632C and MP1776A. |
Chapter 7  SCPI Commands

**:CALCulate:EALarm:CLEValuation?**

**Response**  
<numeric> = <NR1 NUMERIC RESPONSE DATA>
0  Out of evaluation target
1  Evaluation target

**Function**  
Queries whether the Clock Loss occurrence period is included to the measurement evaluation target.

**Example**  
> :CALCulate:EALarm:CLEValuation?
< 1

**Compatibility**  
Compatible with the MP1632C and MP1776A.

**:CALCulate:EALarm:CRUNlock <boolean>**

**Parameter**  
<boolean> = <BOOLEAN PROGRAM DATA>
0 or OFF  Does not set to the evaluation target.
1 or ON  Sets to the evaluation target.

**Function**  
Sets whether to include the CR Unlock occurrence period to the measurement evaluation target.

**Example**  
To include the CR Unlock occurrence period to the measurement evaluation target:
> :CALCulate:EALarm:CRUNlock ON

**Compatibility**  
Incompatible with existing models.

**:CALCulate:EALarm:CRUNlock?**

**Response**  
<numeric> = <NR1 NUMERIC RESPONSE DATA>
0  Out of evaluation target
1  Evaluation target

**Function**  
Queries whether the CR Unlock occurrence period is included to the measurement evaluation target.

**Example**  
> :CALCulate:EALarm:CRUNlock?
< 1

**Compatibility**  
Incompatible with existing models.
### :CALCulate:EALarm:SLEValuation <boolean>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>(&lt;\text{boolean}&gt; = \langle\text{BOOLEAN PROGRAM DATA}\rangle)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0 or OFF (\rightarrow) Does not set to the evaluation target.</td>
</tr>
<tr>
<td></td>
<td>1 or ON (\rightarrow) Sets to the evaluation target.</td>
</tr>
</tbody>
</table>

**Function**
Sets whether to include the Sync Loss occurrence period to the measurement evaluation target.

**Example**
To include the Sync Loss occurrence period to the measurement evaluation target:

\[ > \text{:CALCulate:EALarm:SLEValuation ON} \]

**Compatibility**
Compatible with the MP1632C and MP1776A.

### :CALCulate:EALarm:SLEValuation? 

<table>
<thead>
<tr>
<th>Response</th>
<th>(&lt;\text{numeric}&gt; = \langle\text{NR1 NUMERIC RESPONSE DATA}\rangle)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0 (\rightarrow) Out of evaluation target</td>
</tr>
<tr>
<td></td>
<td>1 (\rightarrow) Evaluation target</td>
</tr>
</tbody>
</table>

**Function**
Queries whether the Sync Loss occurrence period is included to the measurement evaluation target.

**Example**

\[ > \text{:CALCulate:EALarm:SLEValuation?} \]
\[ < 1 \]

**Compatibility**
Compatible with the MP1632C and MP1776A.
7.4.3 Commands related to Pattern tab

7.4.3.1 Test Pattern setting commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>:SENSe:PATTern:TYPE?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:PATTern:LOGic?</td>
</tr>
</tbody>
</table>
### :SENSe:PATTern:TYPE <type>

**Parameter**

- `<type>` = `<CHARACTER PROGRAM DATA>`
  - PRBS: PRBS pattern
  - ZSUBstitution: Zero Substitution pattern
  - DATA: Data pattern
  - MIXData: Mixed Data pattern
  - SEQuence: Sequence pattern

**Function**
Sets the type of the test pattern.

**Example**
To set the test pattern type to the PRBS pattern:

```plaintext
> :SENSe:PATTern:TYPE PRBS
```

**Compatibility**
Partially compatible with the MP1632C and MP1776A (for ZSUBstitution only).

### :SENSe:PATTern:TYPE? 

**Response**

- `<type>` = `<CHARACTER RESPONSE DATA>`
  - PRBS, ZSUB, DATA, MIXD, SEQ

**Function**
Queries the type of the test pattern.

**Example**

```plaintext
> :SENSe:PATTern:TYPE?
< PRBS
```

**Compatibility**
Partially compatible with the MP1632C and MP1776A (for ZSUB only).

### :SENSe:PATTern:LOGic <logic>

**Parameter**

- `<logic>` = `<CHARACTER PROGRAM DATA>`
  - POSitive: Positive logic
  - NEGative: Negative logic

**Function**
Sets the logic (positive or negative) of the test pattern.

**Example**
To set the logic of the test pattern to the positive logic (POSitive):

```plaintext
> :SENSe:PATTern:LOGic POSitive
```

**Compatibility**
Incompatible with existing models.

### :SENSe:PATTern:LOGic? 

**Response**

- `<logic>` = `<CHARACTER RESPONSE DATA>`
  - POS, NEG

**Function**
Queries the logic of the test pattern.

**Example**

```plaintext
> :SENSe:PATTern:LOGic?
< POS
```

**Compatibility**
Incompatible with existing models.
7.4.3.2  PRBS pattern setting commands

Table 7.4.3.2-1  PRBS pattern setting commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>:SENSe:PATTern:PRBS:LENGth?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:PATTern:PRBS:MRATio?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:PATTern:PRBS:BSHift?</td>
</tr>
</tbody>
</table>

:SENSe:PATTern:PRBS:LENGth <numeric>

Parameter  
<numeric> = <DECIMAL NUMERIC PROGRAM DATA>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;numeric&gt;</th>
<th>Function</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>2^(n-1)</td>
<td></td>
<td>&gt; :SENSe:PATTern:PRBS:LENGth 7</td>
</tr>
<tr>
<td>9</td>
<td>2^(n-1)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>2^(n-1)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>2^(n-1)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>2^(n-1)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>2^(n-1)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>2^(n-1)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>31</td>
<td>2^(n-1)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Function  
Sets the number of stages \((2^n-1)\) during PRBS pattern reception.

Example  
To set the number of stages during PRBS pattern reception to \(2^7-1\):  
> :SENSe:PATTern:PRBS:LENGth 7

Compatibility  
Incompatible with existing models.
7.4 ED Commands

:SENSe:PATTern:PRBS:LENGth?

Response: \(<\text{numeric}> = <\text{NR1 NUMERIC RESPONSE DATA}>\)  
7, 9, 10, 11, 15, 20, 23, 31

Function: Queries the number of stages during PRBS pattern reception.

Example:  
> :SENSe:PATTern:PRBS:LENGth?  
< 7

Compatibility: Incompatible with existing models.

:SENSe:PATTern:PRBS:MRATio <mratio>

Parameter: \(<\text{mratio}> = <\text{CHARACTER PROGRAM DATA}>\)  
M1_2 1/2  
M1_4 1/4  
M1_8 1/8  
M0_8 0/8  
I1_2 1/2INVT  
M3_4 3/4  
M7_8 7/8  
M8_8 8/8

Function: Sets the mark ratio during PRBS pattern reception.

Example: To set the mark ratio during PRBS pattern reception to 1/2:  
> :SENSe:PATTern:PRBS:MRATio M1_2

Compatibility: Compatible with the MP1632C.

:SENSe:PATTern:PRBS:MRATio?

Response: \(<\text{mratio}> = <\text{CHARACTER RESPONSE DATA}>\)  
M1_2, M1_4, M1_8, M0_8, I1_2, M3_4, M7_8, M8_8

Function: Queries the mark ratio during PRBS pattern reception.

Example:  
> :SENSe:PATTern:PRBS:MRATio?  
< M1_2

Compatibility: Compatible with the MP1632C.

:SENSe:PATTern:PRBS:BSHift <numeric>

Parameter: \(<\text{numeric}> = <\text{DECIMAL NUMERIC PROGRAM DATA}>\)  
1 1 bit  
3 3 bits

Function: Sets the bit shift during PRBS pattern reception.

Example: To set the bit shift during PRBS pattern reception to 1 bit:  
> :SENSe:PATTern:PRBS:BSHift 1

Compatibility: Compatible with the MP1632C.
### :SENSe:PATTern:PRBS:BSHift?

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;numeric&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>1, 3</td>
<td></td>
</tr>
</tbody>
</table>

**Function**
Queries the bit shift during PRBS pattern reception.

**Example**
```
> :SENSe:PATTern:PRBS:BSHift?
< 1
```

**Compatibility**
Compatible with the MP1632C.
7.4.3.3 Zero Substitution pattern setting commands

![Figure 7.4.3.3-1 Zero Substitution pattern setting](image)

### Table 7.4.3.3-1 Zero Substitution pattern setting commands

<table>
<thead>
<tr>
<th>No</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>:SENSe:PATTern:ZSUBstitute:LENGth?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:PATTern:ZSUBstitute:ZLENgth?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:PATTern:ZSUBstitute:ADDBit?</td>
</tr>
</tbody>
</table>

### :SENSe:PATTern:ZSUBstitute:LENGth <length>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;length&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>$2^n (n = 7)$</td>
</tr>
<tr>
<td>9</td>
<td>$2^n (n = 9)$</td>
</tr>
<tr>
<td>10</td>
<td>$2^n (n = 10)$</td>
</tr>
<tr>
<td>11</td>
<td>$2^n (n = 11)$</td>
</tr>
<tr>
<td>15</td>
<td>$2^n (n = 15)$</td>
</tr>
<tr>
<td>20</td>
<td>$2^n (n = 20)$</td>
</tr>
<tr>
<td>23</td>
<td>$2^n (n = 23)$</td>
</tr>
<tr>
<td>–7</td>
<td>$2^{n-1} (n = 7)$</td>
</tr>
<tr>
<td>–9</td>
<td>$2^{n-1} (n = 9)$</td>
</tr>
<tr>
<td>–10</td>
<td>$2^{n-1} (n = 10)$</td>
</tr>
<tr>
<td>–11</td>
<td>$2^{n-1} (n = 11)$</td>
</tr>
<tr>
<td>–15</td>
<td>$2^{n-1} (n = 15)$</td>
</tr>
<tr>
<td>–20</td>
<td>$2^{n-1} (n = 20)$</td>
</tr>
<tr>
<td>–23</td>
<td>$2^{n-1} (n = 23)$</td>
</tr>
</tbody>
</table>

### Function
Sets the pattern length during Zero Substitution pattern reception.

### Example
To set the pattern length during Zero Substitution pattern reception to $2^7$:

> :SENSe:PATTern:ZSUBstitute:LENGth 7

### Compatibility
Compatible with the MP1632C and MP1776A.
Chapter 7  SCPI Commands

:SENSe:PATTern:ZSUBstitute:LENGth?

Response

<length> = <NR1 NUMERIC RESPONSE DATA>
7, 9, 10, 11, 15, 20, 23, −7, −9, −10, −11, −15, −20, −23

Function
Queries the pattern length during Zero Substitution pattern reception.

Example
> :SENSe:PATTern:ZSUBstitute:LENGth?
< 7

Compatibility
Compatible with the MP1632C and MP1776A.

:SENSe:PATTern:ZSUBstitute:ZLENgth <numeric>

Parameter

<numeric> = <DECIMAL NUMERIC PROGRAM DATA>

When ZSUBlength, $2^n$ is set:
1 to $2^{n−1}$

When ZSUBlength, $2^{n−1}$ is set
1 to $2^{n−2}$

$n = 7, 9, 11, 15, 20, 23$

Function
Sets the number of bits to which "0" is inserted (zero-insertion bit count) during Zero Substitution pattern reception.
Available parameters vary depending on the setting conditions.

Example
To set the zero-insertion bit count during Zero Substitution pattern reception to 10 bits:
> :SENSe:PATTern:ZSUBstitute:ZLENgth 10

Compatibility
Partially compatible with the MP1632C and MP1776A.

:SENSe:PATTern:ZSUBstitute:ZLENgth?

Response

<numeric> = <NR1 NUMERIC RESPONSE DATA>
1 to $2^{n−1}$

$n = 7, 9, 11, 15, 20, 23$

Function
Queries the zero-insertion bit count during Zero Substitution pattern reception.

Example
> :SENSe:PATTern:ZSUBstitute:ZLENgth?
< 10

Compatibility
Partially compatible with the MP1632C and MP1776A.
7.4 ED Commands

**:SENSe:PATTern:ZSUBstitute:ADDBit <numeric>**

**Parameter**

<table>
<thead>
<tr>
<th>&lt;numeric&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
</tr>
<tr>
<td>1</td>
</tr>
</tbody>
</table>

**Function**
Sets whether to add one bit of "0" or "1" next to the last of zero-inserted bits during Zero Substitution pattern reception.

**Example**
To add one bit of "0" next to the last of zero-inserted bits during Zero Substitution pattern reception:
> :SENSe:PATTern:ZSUBstitute:ADDBit 0

**Compatibility**
Incompatible with existing models.

**:SENSe:PATTern:ZSUBstitute:ADDBit?**

**Response**

<table>
<thead>
<tr>
<th>&lt;numeric&gt; = &lt;DECIMAL NUMERIC RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
</tr>
<tr>
<td>1</td>
</tr>
</tbody>
</table>

**Function**
Queries whether one bit of "0" or "1" is added next to the last of zero-inserted bits during Zero Substitution pattern reception.

**Example**
> :SENSe:PATTern:ZSUBstitute:ADDBit?
< 0

**Compatibility**
Incompatible with existing models.
7.4.3.4 Data pattern setting commands

![Data pattern setting](image)

**Figure 7.4.3.4-1** Data pattern setting

![Data pattern setting (Pattern Editor screen)](image)

**Figure 7.4.3.4-2** Data pattern setting (Pattern Editor screen)

**Table 7.4.3.4-1** Data pattern setting commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>:SENSe:PATTern:DATA:LENGth?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:PATTern:DATA:WHOLE?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:PATTern:BDATa:WHOLE?</td>
</tr>
<tr>
<td>[6]</td>
<td>No label (All 0 or 1)</td>
<td>:SENSe:PATTern:DATA:FILL</td>
</tr>
</tbody>
</table>
**SENSe:PATTern:DATA:LENGth <numeric>**

**Parameter**

<numeric> = <DECIMAL NUMERIC PROGRAM DATA>

2 to 134217728  2 to 134,217,728 bits, in 1-bit steps

**Note:**

At 2 ch Combination, the setting range and Step are doubled.
At 4 ch Combination, the setting range and Step are quadrupled.

**Function**

Sets the pattern length during Data pattern reception.

**Example**

To set the pattern length during Data pattern reception to 2 bits:

> :SENSe:PATTern:DATA:LENGth 2

**Compatibility**

Incompatible with existing models.

**SENSe:PATTern:DATA:LENGth?**

**Response**

<numeric> = <NR1 NUMERIC RESPONSE DATA>

**Function**

Queries the pattern length during Data pattern reception.

**Example**

> :SENSe:PATTern:DATA:LENGth?

< 2

**Compatibility**

Incompatible with existing models.

**SENSe:PATTern:DATA:WHALe <start>,<end>,<data>**

**Parameter**

<start> = <NON-DECIMAL PROGRAM DATA>

#H0 to #H7FFFFFF 0 to 7FFFFFF bits, in 1-bit steps

(Specify in hexadecimal)

<end> = <NON-DECIMAL PROGRAM DATA>

#H0 to #H7FFFFFF 0 to 7FFFFFF bits, in 1-bit steps

(Specify in hexadecimal)

**Note:**

The maximum setting is the pattern length.
At 2 ch Combination, the setting range is doubled.
At 4 ch Combination, the setting range is quadrupled.

<data> = <STRING PROGRAM DATA>

"H***...*" Specify pattern data in hexadecimal format.
1 to 400 characters (pattern data of 400 × 4 bits)
Specify a character string consisting of 0 to 9
and A to F.

If the last character string of the pattern data is
short of a hexadecimal unit (4 bits), add 0(s) up
to the last bit to make a hexadecimal string.

"B***...*" Specify pattern data in binary format.
1 to 400 characters (pattern data of 400 bits)
Specify a character string consisting of 0 and 1.
pattern reception.
The set data overwrites the specified range.

**Example**
To set the addresses from bit 0 to bit 7FFFFFF of the pattern data in hexadecimal format:
> :SENSe:PATTern:DATA:WHOLE #H0,#H7FFFFFF,"H001"

**Compatibility**
Compatible with the MP1632C and MP1776A.

**Note:**
When the number of data bits specified in `<data>` is smaller than the range specified by `<start>` and `<end>`, set `<data>` repeatedly up to the specified range.

*Example*
- `<start>` = #H0, `<end>` = #H1F, `<data>` = "HABC"
  Setting data: ABCABCAB
- `<start>` = #H0, `<end>` = #H7, `<data>` = "B011"
  Setting data: 01101101

When the number of data bits specified in `<data>` is greater than the range specified by `<start>` and `<end>`, cut off the part of `<data>` out of the specified range.

*Example*
- `<start>` = #H0, `<end>` = #HF, `<data>` = #HABCDEF
  Setting data: ABCD
- `<start>` = #H0, `<end>` = #H3, `<data>` = #B01100110
  Setting data: 0110
7.4 ED Commands

:SENSe:PATTern:DATA:WHOLe? <start>

Parameter

<start> = <NON-DECIMAL PROGRAM DATA>
#H0 to #H7FFFFFFF 0 to 7FFFFFFF bits, in 1-bit steps
(Specify in hexadecimal)

Note:

The maximum setting is the pattern length.
At 2 ch Combination, the setting range is doubled.
At 4 ch Combination, the setting range is quadrupled.

Response

<data> = <STRING RESPONSE DATA>
"H***...*" Returns the pattern data in hexadecimal format.
Within 400 characters (pattern data of 400 × 4 bits)

Function

Queries the pattern data of 400 × 4 bits from the <start> address during Data pattern reception.

Example

To query the pattern data of 400 × 4 bits from the bit 0 address during Data pattern reception.

> :SENSe:PATTern:DATA:WHOLe? #H0
< "H001"

Compatibility

Compatible with the MP1632C and MP1776A.

:SENSe:PATTern:BDATa:WHOLe <start>,<end>,<bdata>

Parameter

<start> = <NON-DECIMAL PROGRAM DATA>
#H0 to #H7FFFFFFF 0 to 7FFFFFFF bits, in 1-bit steps
(Specify in hexadecimal)

<end> = <NON-DECIMAL PROGRAM DATA>
#H0 to #H7FFFFFFF 0 to 7FFFFFFF bits, in 1-bit steps
(Specify in hexadecimal)

<bdata> = <ARBITRARY BLOCK PROGRAM DATA>
#XYYY<binary> X: Number of YYY digits
YYY: Number of bytes of <binary> 1 to 16777216 bytes
<binary>:Binary data up to 16777216 bytes

Note:

The maximum setting is the pattern length.
At 2 ch Combination, the setting range is doubled.
At 4 ch Combination, the setting range is quadrupled.

Function

Sets binary data of the pattern data from <start> to <end> addresses during Data pattern reception.
The set data overwrites the specified range.

Example

To set binary data of the pattern data from bit 0 to bit 7FFFFFFF addresses during Data pattern reception:

> :SENSe:PATTern:BDATa:WHOLe #H0,#H7FFFFFFF,#10011

Compatibility

Compatible with the MP1632C and MP1776A.
Chapter 7  SCPI Commands

**:SENSe:PATTern:BDATa:WHOLe? <start>[,size]**

**Parameter**

- `<start>` = `<NON-DECIMAL PROGRAM DATA>`
  - #H0 to #H7FFFFFF 0 to 7FFFFFFF bits, in 1-bit steps (Specify in hexadecimal)

- `[size]` = `<NR1 NUMERIC PROGRAM DATA>`
  - 1 to 16777216 1 to 16,777,216 bytes, in 1-byte steps

Binary data of the setting pattern is queried when `[size]` is omitted.

**Note:**

- The maximum setting is the pattern length.
- At 2 ch Combination, the setting range is doubled.
- At 4 ch Combination, the setting range is quadrupled.

**Response**

- `<bdata>` = `<DEFINITE LENGTH ARBITRARY BLOCK RESPONSE DATA>`
  - `#XYYY<binary>`
    - X: Number of YYY digits
    - YYY: Number of bytes of `<binary>`
  - 1 to 16,777,216 bytes
    - `<binary>`: Binary data of up to 16,777,216 bytes

**Function**

Queries binary data of bytes specified by `<size>` from `<start>` address of the pattern data during Data pattern reception.

**Example**

To query binary data of 16,777,216 bytes from bit 0 address of the pattern data during Data pattern reception:

```plaintext
> :SENSe:PATTern:BDATa:WHOLe? #H0
< #10011
```

**Compatibility**

Compatible with the MP1632C and MP1776A.

**:SENSe:PATTern:DREVerse:ADDRess <start>,<end>**

**Parameter**

- `<start>` = `<NON-DECIMAL PROGRAM DATA>`
  - #H0 to #H7FFFFFF 0 to 7FFFFFFF bits, in 1-bit steps (Specify in hexadecimal)

- `<end>` = `<NON-DECIMAL PROGRAM DATA>`
  - #H0 to #H7FFFFFF 0 to 7FFFFFFF bits, in 1-bit steps (Specify in hexadecimal)

**Note:**

- The maximum setting is the pattern length.
- At 2 ch Combination, the setting range is doubled.
- At 4 ch Combination, the setting range is quadrupled.

**Function**

Inverts the data in the program during Data pattern reception. Specify the inversion range by the `<start>` and `<end>` addresses.

**Example**

To invert addresses from bit 0 to bit 7FFFFFF in the program during Data pattern reception:

```plaintext
> :SENSe:PATTern:DREVerse:ADDRess #H0,#H7FFFFFF
```

**Compatibility**

Compatible with the MP1632C.
SENSe:PATTern:DREVerse:DELTa <start>,<delta>

Parameter

- **<start>** = **<NON-DECIMAL PROGRAM DATA>**
  - #H0 to #H7FFFFFFF 0 to 7FFFFFFF bits, in 1-bit steps (Specify in hexadecimal)
- **<delta>** = **<DECIMAL NUMERIC PROGRAM DATA>**
  - 1 to 134217728 1 to 134,217,728 bits, in 1-bit steps

**Note:**
- The maximum setting is the pattern length.
- At 2 ch Combination, the setting range is doubled.
- At 4 ch Combination, the setting range is quadrupled.

**Function**
Inverts the data in the program during Data pattern reception. Specify the inversion range by the number of bits from the <start> address (relative specification).

**Example**
To invert 1 bit of the pattern data from address 0 in the program during Data pattern reception:
> :SENSe:PATTern:DREVerse:DELTa #H0,1

**Compatibility**
Compatible with the MP1632C.

SENSe:PATTern:DATA:FILL <range>,<page>,<data>

Parameter

- **<range>** = **<CHARACTER PROGRAM DATA>**
  - PAGE Specifies a page.
    - (One page is defined as 128 bits.)
  - ALL Specifies all data.
- **<page>** = **<DECIMAL NUMERIC PROGRAM DATA>**
  - 1 to 1048576 1 to 1,048,576 (Max.) pages, in 1-page steps
  - Max = \( \frac{\text{Pattern Length}}{128} \) (rounding up fractions)
- **<data>** = **<DECIMAL NUMERIC PROGRAM DATA>**
  - 0 Fills the specified range with 0s.
  - 1 Fills the specified range with 1s.

**Function**
Sets 0s or 1s for the specified page or all data in the program during Data pattern reception.

**Example**
To set 0s for the first page of the program during Data pattern reception:
> :SENSe:PATTern:DATA:FILL PAGE,1,0

**Compatibility**
Incompatible with existing models.
7.4.3.5 Mixed Data pattern setting commands

Figure 7.4.3.5-1 Mixed Data pattern setting

Figure 7.4.3.5-2 Descramble Setup dialog box
Table 7.4.3.5-1  Mixed Data pattern setting commands

<table>
<thead>
<tr>
<th>No</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1]</td>
<td>Bit Shift</td>
<td>::SENSe::PATTern::MIXData::PRBS::BITShift</td>
</tr>
<tr>
<td></td>
<td></td>
<td>::SENSe::PATTern::MIXData::PRBS::BITShift?</td>
</tr>
<tr>
<td>[2]</td>
<td>PRBS Pattern</td>
<td>::SENSe::PATTern::MIXData::PRBS::LENGth</td>
</tr>
<tr>
<td></td>
<td></td>
<td>::SENSe::PATTern::MIXData::PRBS::LENGth?</td>
</tr>
<tr>
<td>[3]</td>
<td>PRBS Mark Ratio</td>
<td>::SENSe::PATTern::MIXData::PRBS::MRATio</td>
</tr>
<tr>
<td></td>
<td></td>
<td>::SENSe::PATTern::MIXData::PRBS::MRATio?</td>
</tr>
<tr>
<td>[4]</td>
<td>Descramble</td>
<td>::SENSe::PATTern::MIXData::DESCramble</td>
</tr>
<tr>
<td></td>
<td></td>
<td>::SENSe::PATTern::MIXData::DESCramble?</td>
</tr>
<tr>
<td>[5]</td>
<td>PRBS Sequence</td>
<td>::SENSe::PATTern::MIXData::PRBS::SEQUence</td>
</tr>
<tr>
<td></td>
<td></td>
<td>::SENSe::PATTern::MIXData::PRBS::SEQUence?</td>
</tr>
<tr>
<td>[6]</td>
<td>Setup</td>
<td>::SENSe::PATTern::MIXData::SRSetting</td>
</tr>
<tr>
<td></td>
<td></td>
<td>::SENSe::PATTern::MIXData::SRSetting?</td>
</tr>
<tr>
<td>[7]</td>
<td>Set All/Reset All</td>
<td>::SENSe::PATTern::MIXData::DSrCramble::ALLSet</td>
</tr>
<tr>
<td>[8]</td>
<td>Number of Block</td>
<td>::SENSe::PATTern::MIXData::BLOCk</td>
</tr>
<tr>
<td></td>
<td></td>
<td>::SENSe::PATTern::MIXData::BLOCk?</td>
</tr>
<tr>
<td>[9]</td>
<td>Row Length</td>
<td>::SENSe::PATTern::MIXData::RLENgth</td>
</tr>
<tr>
<td></td>
<td></td>
<td>::SENSe::PATTern::MIXData::RLENgth?</td>
</tr>
<tr>
<td>[10]</td>
<td>Data Length</td>
<td>::SENSe::PATTern::MIXData::DATA::LENGth</td>
</tr>
<tr>
<td></td>
<td></td>
<td>::SENSe::PATTern::MIXData::DATA::LENGth?</td>
</tr>
<tr>
<td>[11]</td>
<td>Number of Row</td>
<td>::SENSe::PATTern::MIXData::ROW</td>
</tr>
<tr>
<td></td>
<td></td>
<td>::SENSe::PATTern::MIXData::ROW?</td>
</tr>
</tbody>
</table>

Figure 7.4.3.5-3  Mixed Data pattern setting (Pattern Editor screen)
Table 7.4.3.5-1  Mixed Data pattern setting commands (Cont’d)

<table>
<thead>
<tr>
<th>No</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>:SENSe:PATTern:MIXData:DATA:WHOLe?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:PATTern:MIXData:BDATa:WHOLe?</td>
</tr>
<tr>
<td>[16]</td>
<td>No label (Data inversion)</td>
<td>:SENSe:PATTern:MIXData:DATA:FILL</td>
</tr>
</tbody>
</table>

**:SENSe:PATTern:MIXData:PRBS:BITShift <numeric>**

Parameter  

<numeric> = <DECIMAL NUMERIC PROGRAM DATA>
1 1 bit
3 3 bits

Function  
Sets the bit shift for the PRBS pattern during Mixed Data pattern reception.

Example  
To set the bit shift for the PRBS pattern during Mixed Data pattern reception to 1-bit shit:

> :SENSe:PATTern:MIXData:PRBS:BITShift 1

Compatibility  
Incompatible with existing models.

**:SENSe:PATTern:MIXData:PRBS:BITShift?**

Response  
<numeric> = <NR1 NUMERIC RESPONSE DATA>
1 1 bit
3 3 bits

Function  
Queries the bit shift for the PRBS pattern during Mixed Data pattern reception.

Example  
> :SENSe:PATTern:MIXData:PRBS:BITShift?
< 1

Compatibility  
Incompatible with existing models.
:SENSe:PATTern:MIXData:PRBS:LENGth <numeric>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;numeric&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>$2^{n-1}$ (n = 7)</td>
</tr>
<tr>
<td>9</td>
<td>$2^{n-1}$ (n = 9)</td>
</tr>
<tr>
<td>10</td>
<td>$2^{n-1}$ (n = 10)</td>
</tr>
<tr>
<td>11</td>
<td>$2^{n-1}$ (n = 11)</td>
</tr>
<tr>
<td>15</td>
<td>$2^{n-1}$ (n = 15)</td>
</tr>
<tr>
<td>20</td>
<td>$2^{n-1}$ (n = 20)</td>
</tr>
<tr>
<td>23</td>
<td>$2^{n-1}$ (n = 23)</td>
</tr>
<tr>
<td>31</td>
<td>$2^{n-1}$ (n = 31)</td>
</tr>
</tbody>
</table>

Function
Sets the number of stages of the PRBS pattern during Mixed Data pattern reception.

Example
To set the number of stages of the PRBS pattern during Mixed Data pattern reception to $2^{7-1}$:

```
> :SENSe:PATTern:MIXData:PRBS:LENGth 7
```

Compatibility
Incompatible with existing models.

:SENSe:PATTern:MIXData:PRBS:LENGth?

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;numeric&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>7, 9, 10, 11, 15, 20, 23, 31</td>
<td></td>
</tr>
</tbody>
</table>

Function
Queries the number of stages of the PRBS pattern during Mixed Data pattern reception.

Example
To query the number of stages of the PRBS pattern during Mixed Data pattern reception:

```
> :SENSe:PATTern:MIXData:PRBS:LENGth?
< 7
```

Compatibility
Incompatible with existing models.
Chapter 7  SCPI Commands

:SENSe:PATTern:MIXData:PRBS:MRATio <mratio>

Parameter  

<mratio> = <CHARACTER PROGRAM DATA>

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>M1_2</td>
<td>1/2</td>
</tr>
<tr>
<td>M1_4</td>
<td>1/4</td>
</tr>
<tr>
<td>M1_8</td>
<td>1/8</td>
</tr>
<tr>
<td>M0_8</td>
<td>0/8</td>
</tr>
<tr>
<td>I1_2</td>
<td>1/2INVT</td>
</tr>
<tr>
<td>M3_4</td>
<td>3/4</td>
</tr>
<tr>
<td>M7_8</td>
<td>7/8</td>
</tr>
<tr>
<td>M8_8</td>
<td>8/8</td>
</tr>
</tbody>
</table>

Function  
Sets the mark ratio for the PRBS pattern during Mixed Data pattern reception.

Example  
To set the mark ratio for the PRBS pattern during Mixed Data pattern reception to 1/2:

> :SENSe:PATTern:MIXData:PRBS:MRATio M1_2

Compatibility  
Incompatible with existing models.

:SENSe:PATTern:MIXData:PRBS:MRATio?

Response  

<mratio>=<CHARACTER RESPONSE DATA>

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>M1_2, M1_4, M1_8, M0_8, I1_2, M3_4, M7_8, M8_8</td>
<td></td>
</tr>
</tbody>
</table>

Function  
Queries the mark ratio for the PRBS pattern during Mixed Data pattern reception.

Example  
To query the mark ratio for the PRBS pattern during Mixed Data pattern reception:

> :SENSe:PATTern:MIXData:PRBS:MRATio?

< M1_2

Compatibility  
Incompatible with existing models.

:SENSe:PATTern:MIXData:DESCramble <boolean>

Parameter  

<boolean> = <BOOLEAN PROGRAM DATA>

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0 or OFF</td>
<td>Descramble OFF</td>
</tr>
<tr>
<td>1 or ON</td>
<td>Descramble ON</td>
</tr>
</tbody>
</table>

Function  
Sets Descramble ON/OFF of the PRBS7 stage during pattern reception.

Example  
To set Descramble ON of the PRBS7 stage during pattern reception:

> :SENSe:PATTern:MIXData:DESCramble ON

Compatibility  
Incompatible with existing models.
### :SENSe:PATTern:MIXData:DESCramble?

**Response**

<numeric> = <NR1 NUMERIC RESPONSE DATA>

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Descramble OFF</td>
</tr>
<tr>
<td>1</td>
<td>Descramble ON</td>
</tr>
</tbody>
</table>

**Function**

Queries the Descramble ON/OFF state of the PRBS7 stage during pattern reception.

**Example**

```plaintext
> :SENSe:PATTern:MIXData:DESCramble?
< 1
```

**Compatibility**

Incompatible with existing models.

---

### :SENSe:PATTern:MIXData:PRBS:SEQuence <sequence>

**Parameter**

<sequence> = <CHARACTER PROGRAM DATA>

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>REST</td>
<td>Discontinuous PRBS pattern</td>
</tr>
<tr>
<td>CON</td>
<td>Continuous PRBS pattern</td>
</tr>
</tbody>
</table>

**Function**

Sets the PRBS pattern sequence during Mixed Data pattern reception.

**Example**

To set the PRBS pattern sequence during Mixed Data pattern reception to Restart:

```plaintext
> :SENSe:PATTern:MIXData:PRBS:SEQuence RESTart
```

**Compatibility**

Incompatible with existing models.

---

### :SENSe:PATTern:MIXData:PRBS:SEQuence?

**Response**

<sequence> = <CHARACTER RESPONSE DATA>

REST, CONS

**Function**

Queries the PRBS pattern sequence during Mixed Data pattern reception.

**Example**

```plaintext
> :SENSe:PATTern:MIXData:PRBS:SEQuence?
< REST
```

**Compatibility**

Incompatible with existing models.
**SENSe:PATTern:MIXData:SRSetting**

Parameter:
- `<row>` = `<DECIMAL NUMERIC PROGRAM DATA>`
  - 1 to 16: 1 to 16 rows, in 1-row steps
- `<data>` = `<BOOLEAN PROGRAM DATA>`
  - 0 or OFF: Descramble OFF
  - 1 or ON: Descramble ON
- `<prbs>` = `<BOOLEAN PROGRAM DATA>`
  - 0 or OFF: Descramble OFF
  - 1 or ON: Descramble ON

Note, however, that Descramble OFF is fixed for Data of 1 Row.

Function:
Sets Descramble ON or OFF for the Data and PRBS of the specified Row.

Example:
To set the descrambling range as “1 Row, Data ON, PRBS ON”:

```
> :SENSe:PATTern:MIXData:SRSetting 1,ON,ON
```

Compatibility:
Incompatible with existing models.

**SENSe:PATTern:MIXData:SRSetting?**

Parameter:
- `<row>` = `<DECIMAL NUMERIC PROGRAM DATA>`
  - 1 to 16: 1 to 16 rows, in 1-row steps

Response:
- `<data>,<prbs>` = `<NR1 NUMERIC RESPONSE DATA>`
  - 0: Descramble OFF
  - 1: Descramble ON

Function:
Queries the Descramble setting (ON/OFF) for the Data and PRBS of the specified Row.

Example:
```
> :SENSe:PATTern:MIXData:SRSetting? 1
< 1,1
```

Compatibility:
Incompatible with existing models.

**SENSe:PATTern:MIXData:DESCramble:ALLSet**

Parameter:
- `<numeric>` = `<NUMERIC PROGRAM DATA>`
  - 0: All Reset
  - 1: All Set

Function:
Selects or deselects all descramble.

Example:
To select all descramble:

```
> :SENSe:PATTern:MIXData:DESCramble:ALLSet 1
```

Compatibility:
Incompatible with existing models.
### 7.4 ED Commands

#### :SENSe:PATTern:MIXData:BLOCk <numeric>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;numeric&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function</td>
<td>Sets the number of blocks during Mixed Data pattern reception.</td>
</tr>
<tr>
<td>Example</td>
<td>To set the number of blocks during Mixed Data pattern reception to 1:</td>
</tr>
<tr>
<td></td>
<td>&gt; :SENSe:PATTern:MIXData:BLOCk 1</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Incompatible with existing models.</td>
</tr>
</tbody>
</table>

#### :SENSe:PATTern:MIXData:BLOCk?

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;numeric&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function</td>
<td>Queries the number of blocks during Mixed Data pattern reception.</td>
</tr>
<tr>
<td>Example</td>
<td>&gt; :SENSe:PATTern:MIXData:BLOCk?</td>
</tr>
<tr>
<td></td>
<td>&lt; 1</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Incompatible with existing models.</td>
</tr>
</tbody>
</table>

#### :SENSe:PATTern:MIXData:RLENgth <numeric>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;numeric&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Note: At 2 ch Combination, the setting range and Step are doubled.</td>
<td></td>
</tr>
<tr>
<td>At 4 ch Combination, the setting range and Step are quadrupled.</td>
<td></td>
</tr>
<tr>
<td>Function</td>
<td>Sets the pattern length of 1 Row that is edited during Mixed Data pattern reception.</td>
</tr>
<tr>
<td>Example</td>
<td>To set the pattern length of 1 Row that is edited during Mixed Data pattern reception to 768 bits:</td>
</tr>
<tr>
<td></td>
<td>&gt; :SENSe:PATTern:MIXData:RLENgth 768</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Incompatible with existing models.</td>
</tr>
</tbody>
</table>

#### :SENSe:PATTern:MIXData:RLENgth?

| Response                                       | <numeric> = <NR1 NUMERIC RESPONSE DATA> |
| Function                                       | Queries the pattern length of 1 Row that is edited during Mixed Data pattern reception. |
| Example                                        | > :SENSe:PATTern:MIXData:RLENgth?        |
|                                                 | < 768                                   |
| Compatibility                                  | Incompatible with existing models.      |
Chapter 7  SCPI Commands

:SENSe:PATTern:MIXData:DATA:LENGth <numeric>

Parameter  
<numeric> = <DECIMAL NUMERIC PROGRAM DATA>
512 to 134217728  512 to 134,217,728 bits, in 1-bit steps

Note:
At 2 ch Combination, the setting range and Step are doubled.
At 4 ch Combination, the setting range and Step are quadrupled.

Function  
Sets the pattern length on the pattern data side that is edited during Mixed Data pattern reception.

Example  
To set the pattern length on the pattern data side that is edited during Mixed Data pattern reception to 16 bits:
> :SENSe:PATTern:MIXData:DATA:LENGth 16

Compatibility  
Incompatible with existing models.

:SENSe:PATTern:MIXData:DATA:LENGth?

Response  
<numeric> = <NR1 NUMERIC RESPONSE DATA>

Function  
Queries the pattern length on the pattern data side that is edited during Mixed Data pattern reception.

Example  
> :SENSe:PATTern:MIXData:DATA:LENGth?
< 16

Compatibility  
Incompatible with existing models.

:SENSe:PATTern:MIXData:ROW <numeric>

Parameter  
<numeric> = <DECIMAL NUMERIC PROGRAM DATA>
1 to 16  1 to 16 rows, in 1-row steps

Function  
Sets the number of rows within one block that are edited during Mixed Data pattern reception.

Example  
To set the number of rows within one block that are edited during Mixed Data pattern reception to 1:
> :SENSe:PATTern:MIXData:ROW 1

Compatibility  
Incompatible with existing models.

:SENSe:PATTern:MIXData:ROW?

Response  
<numeric> = <NR1 NUMERIC RESPONSE DATA>
1 to 16  1 to 16 Rows

Function  
Queries the number of rows within one block that are edited during Mixed Data pattern reception.

Example  
> :SENSe:PATTern:MIXData:ROW?
< 1

Compatibility  
Incompatible with existing models.
:SENSe:PATTern:MIXData:DATA:WHOLe <block>,<start>,<end>,<data>

Parameter

- `<block>` = <DECIMAL NUMERIC PROGRAM DATA>
  1 to 511 blocks, in 1-block steps
- `<start>` = <NON-DECIMAL PROGRAM DATA>
  #H0 to #H7FFFFFF 0 to 7FFFFFF bits, in 1-bit steps (Specify in hexadecimal)
- `<end>` = <NON-DECIMAL PROGRAM DATA>
  #H0 to #H7FFFFFF 0 to 7FFFFFF bits, in 1-bit steps (Specify in hexadecimal)

Note:

- The maximum setting is the pattern length.
- At 2 ch Combination, the setting range is doubled.
- At 4 ch Combination, the setting range is quadrupled.

- `<data>` = <STRING PROGRAM DATA>
  "H***...*" Specify pattern data in hexadecimal format.
  1 to 400 characters (pattern data of 400 x 4 bits)
  Specify a character string consisting of 0 to 9 and A to F.
  If the last character string of the pattern data is short of a hexadecimal unit (4 bits), add 0(s) up to the last bit to make a hexadecimal string.

  "B***...*" Specify pattern data in binary format.
  1 to 400 characters (pattern data of 400 bits)
  Specify a character string consisting of 0 and 1.

Function
Sets the pattern data in a block from the `<start>` to `<end>` addresses during Mixed Data pattern reception.
The set data overwrites the specified range.

Example
To set the addresses from bit 0 to bit 7FFFFFF of the pattern data in block 3 during Mixed Data pattern reception in hexadecimal:
> :SENSe:PATTern:MIXData:DATA:WHOLe 3,#H0,#7FFFFFFF, "H001"

Compatibility
Incompatible with existing models.
### :SENSe:PATrern:MIXData:DATA:WHLel? <block>,<start>

**Parameter**

- `<block>` = `<DECIMAL NUMERIC PROGRAM DATA>`
  - 1 to 511
  - 1 to 511 blocks, in 1-block steps
- `<start>` = `<NON-DECIMAL PROGRAM DATA>`
  - #H0 to #H7FFFFFF
  - 0 to 7FFFFFFF bits, in 1-bit steps (Specify in hexadecimal)

**Note:**

- The maximum setting is the pattern length.
- At 2 ch Combination, the setting range is doubled.
- At 4 ch Combination, the setting range is quadrupled.

**Response**

- `<data>` = `<STRING RESPONSE DATA>`
  - "H***...*" Returns the pattern data in hexadecimal format.
  - Within 400 characters (pattern data of 400 × 4 bits)

**Function**

Queries the pattern data of 400 × 4 bits in a block from the `<start>` address during Mixed Data pattern reception.

**Example**

To query the pattern data of 400 × 4 bits in block 3 from the bit 0 address during Mixed Data pattern reception.

```
> :SENSe:PATrern:MIXData:DATA:WHLel? 3,#H0
< "H001"
```

**Compatibility**

Incompatible with existing models.
**:SENSe:PATTern:MIXData:BDATa:WHOLe**  
*<block>,<start>,<end>,<bdata>*

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;block&gt;</td>
<td>1 to 511 blocks, in 1-block steps</td>
</tr>
<tr>
<td>&lt;start&gt;</td>
<td>0 to 7FFFFFF bits, in 1-bit steps (Specify in hexadecimal)</td>
</tr>
<tr>
<td>&lt;end&gt;</td>
<td>0 to 7FFFFFF bits, in 1-bit steps (Specify in hexadecimal)</td>
</tr>
<tr>
<td>&lt;bdata&gt;</td>
<td>1 to 16,777,216 bytes</td>
</tr>
</tbody>
</table>

**Note:**  
The maximum setting is the pattern length.  
At 2 ch Combination, the setting range is doubled.  
At 4 ch Combination, the setting range is quadrupled.

**Function:**  
Sets binary data of the pattern data in each block from <start> to <end> addresses during Mixed Data pattern reception.  
The set data overwrites the specified range.

**Example:**  
To set binary data of the pattern data in block 3 from bit 0 to bit 7FFFFFF addresses during Mixed Data pattern reception.  
> :SENSe:PATTern:MIXData:BDATa:WHOLe  
3,#H0,#H7FFFFFF,#10011

**Compatibility:**  
Incompatible with existing models.
:SENSe:PATTern:MIXData:BDATa:WHOLe? <block>,<start>[,<size>]

Parameter
- <block> = <DECIMAL NUMERIC PROGRAM DATA>
  1 to 511 1 to 511 blocks, in 1-block steps
- <start> = <NON-DECIMAL PROGRAM DATA>
  #H0 to #H7FFFFFF 0 to 7FFFFFF bits, in 1-bit steps (Specify in hexadecimal)

Note:
- The maximum setting is the pattern length.
- At 2 ch Combination, the setting range is doubled.
- At 4 ch Combination, the setting range is quadrupled.

- [<size>] = <NR1 NUMERIC PROGRAM DATA>
  1 to 16777216 1 to 16777216 bytes, in 1-byte steps

Response
- <bdata> = <DEFINITE LENGTH ARBITRARY BLOCK RESPONSE DATA>
  #XYYY<binary> X: Number of YYY digits
  YYY: Number of bytes of <binary>
  1 to 16,777,216 bytes
  <binary>:Binary data up to 16,777,216 bytes

Binary data of the setting pattern bytes is queried when [<size>] is omitted.

Function
Queries binary data of bytes specified by <size> from the <start> address of the pattern data in each block during Mixed Data pattern reception.

Example
To query binary data of up to the setting pattern from the <Start> address of the pattern data in block 3 during Mixed Data pattern reception:

> :SENSe:PATTern:MIXData:BDATa:WHOLe? 3,#H0
< #10011

Compatibility
Incompatible with existing models.
**7.4 ED Commands**

**:SENSe:PATTern:MIXData:DREVerse:ADDRess <block>,<start>,<end>**

**Parameter**
- **<block>** = <DECIMAL NUMERIC PROGRAM DATA>
  1 to 511
- **<start>** = <NON-DECIMAL PROGRAM DATA>
  #H0 to #H7FFFFFF 0 to 7FFFFFFF bits, in 1-bit steps (Specify in hexadecimal)
- **<end>** = <NON-DECIMAL PROGRAM DATA>
  #H0 to #H7FFFFFF 0 to 7FFFFFFF bits, in 1-bit steps (Specify in hexadecimal)

**Note:**
- The maximum setting is the pattern length.
  - At 2 ch Combination, the setting range is doubled.
  - At 4 ch Combination, the setting range is quadrupled.

**Function**
Inverts the data in the program part in a block during Mixed Data pattern reception. Specify the inversion range by the **<start>** and **<end>** addresses.

**Example**
To invert the data in the program part in block 3 from bit 0 to bit 7FFFFFFF addresses during Mixed Data pattern reception:
> :SENSe:PATTern:MIXData:DREVerse:ADDRess 3,#H0,#H7FFFFFFF

**Compatibility**
Incompatible with existing models.

**:SENSe:PATTern:MIXData:DREVerse:DELTa <block>,<start>,<delta>**

**Parameter**
- **<block>** = <DECIMAL NUMERIC PROGRAM DATA>
  1 to 511
- **<start>** = <NON-DECIMAL PROGRAM DATA>
  #H0 to #H7FFFFFF 0 to 7FFFFFFF bits, in 1-bit steps (Specify in hexadecimal)
- **<delta>** = <DECIMAL NUMERIC PROGRAM DATA>
  1 to 134217728 1 to 134,217,728 bits, in 1-bit steps

**Note:**
- The maximum setting is the pattern length.
  - At 2 ch Combination, the setting range is doubled.
  - At 4 ch Combination, the setting range is quadrupled.

**Function**
Inverts the data in the program part in a block during Mixed Data pattern reception. Specify the inversion range by the number of bits from the **<start>** address (relative specification).

**Example**
To invert 1 bit of the pattern data from address 0 in the program part in block 3 during Mixed Data pattern reception:
> :SENSe:PATTern:MIXData:DREVerse:DELTa 3,#H0,1

**Compatibility**
Incompatible with existing models.
:SENSe:PATTern:MIXData:DATA:FILL <block>,<range>,<page>,<data>

Parameter
- **<block>** = <DECIMAL NUMERIC PROGRAM DATA>
  1 to 511 1 to 511 blocks, in 1-block steps
- **<range>** = <CHARACTER PROGRAM DATA>
  - PAGE Specifies a page.
    - (One page is defined as 128 bits.)
  - ALL Specifies all data.
- **<page>** = <DECIMAL NUMERIC PROGRAM DATA>
  1 to 1048576 1 to 1,048,576 (Max.) pages, in 1-page steps
  - Max = \( \frac{\text{Pattern Length}}{128} \) (rounding up fractions)
  - Specify "0" when <range> is set to ALL.
- **<data>** = <DECIMAL NUMERIC PROGRAM DATA>
  0 Fills the specified range with 0s.
  1 Fills the specified range with 1s.

Function
Sets 0s or 1s for the specified page or all data in the program part in a block during Mixed Data pattern reception.

Example
To set 0s for the first page of the program part in block 3 during Mixed Data pattern reception:
> :SENSe:PA\text{T}tern:MIX\text{Data}:DATA:FILL 3,PAGE,1,0

Compatibility
Incompatible with existing models.
7.4.3.6 Sequence pattern setting commands

Figure 7.4.3.6-1 Sequence pattern setting

Figure 7.4.3.6-2 Match Pattern setting
Figure 7.4.3.6-3  Sequence Pattern setting

Figure 7.4.3.6-4  Sequence pattern setting (Pattern Editor screen)
### Table 7.4.3.6-1  Sequence pattern setting commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>:SENSe:PATTern:SEQuence:MATCh:LENGth?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:PATTern:SEQuence:MATCh:EDIT?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:PATTern:SEQuence:MATCh:MASK?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:PATTern:SEQuence:CONDition:ADD?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:PATTern:SEQuence:LENGth?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:PATTern:SEQuence:DATA:WHOLe?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:PATTern:SEQuence:BDATa:WHOLe?</td>
</tr>
<tr>
<td>[14]</td>
<td>No label (All 0 or All 1)</td>
<td>:SENSe:PATTern:SEQuence:DATA:FILL</td>
</tr>
</tbody>
</table>

`:SENSe:PATTern:SEQuence:MATCh:LENGth`<block>,<content>,<length>

- **Parameter**
  - `<block>` = `<DECIMAL NUMERIC PROGRAM DATA>`
  - 1 to 128 1 to 128 blocks, in 1·block steps
  - `<content>` = `<CHARACTER PROGRAM DATA>`
  - APAT Pattern A
  - BPAT Pattern B
  - `<length>` = `<DECIMAL NUMERIC PROGRAM DATA>`
  - 4 to 64 4 to 64 bits, in 4-bit steps

- **Function**
  - Sets the pattern length of the Match pattern in a block when Test Pattern is Sequence.

- **Example**
  - To set the pattern length of the Match pattern A in block 3 to 4 bits when Test Pattern is Sequence:
    - `> :SENSe:PATTern:SEQuence:MATCh:LENGth 3,APAT,4`

- **Compatibility**
  - Incompatible with existing models.
### :SENSe:PATTern:SEQuence:MATCh:LENGth? <block>,<content>

**Parameter**
- `<block>` = `<DECIMAL NUMERIC PROGRAM DATA>`
  - 1 to 128 1 to 128 blocks, 1-block steps
- `<content>` = `<CHARACTER PROGRAM DATA>`
  - APAT Pattern A
  - BPAT Pattern B

**Response**
- `<length>` = `<NR1 NUMERIC RESPONSE DATA>`
  - 4 to 64

**Function**
Queries the pattern length of the Match pattern in a block when Test Pattern is Sequence.

**Example**
To query the pattern length of the Match pattern A in Block No. 3 when Test Pattern is Sequence:

```
> :SENSe:PATTern:SEQuence:MATCh:LENGth? 3,APAT
< "4"
```

**Compatibility**
Incompatible with existing models.

### :SENSe:PATTern:SEQuence:MATCh:EDIT <block>,<content>,<form>,<data>

**Parameter**
- `<block>` = `<DECIMAL NUMERIC PROGRAM DATA>`
  - 1 to 128 1 to 128 blocks, 1-block steps
- `<content>` = `<CHARACTER PROGRAM DATA>`
  - APAT Pattern A
  - BPAT Pattern B
- `<form>` = `<CHARACTER PROGRAM DATA>`
  - BIN Binary
  - HEX Hexadecimal
- `<data>` = `<STRING PROGRAM DATA>`
  - When `form` = BIN: 1 to 64 characters (64-bit data)
  - When `form` = HEX: 1 to 16 characters (64-bit data)

**Function**
Edits the Match pattern in a block when Test Pattern is Sequence.

**Example**
To set binary data "1010" for the Match Pattern A in Block No. 3 when Test Pattern is Sequence:

```
> :SENSe:PATTern:SEQuence:MATCh:EDIT 3,APAT,BIN, "1010"
```

**Compatibility**
Incompatible with existing models.
### :SENSe:PATTern:SEQuence:MATCh:EDIT? <block>,<content>,<form>

**Parameter**
- `<block>` = `<DECIMAL NUMERIC PROGRAM DATA>`
  - 1 to 128 1 to 128 blocks, 1-block steps
- `<content>` = `<CHARACTER PROGRAM DATA>`
  - APAT, BPAT
- `<form>` = `<CHARACTER PROGRAM DATA>`
  - BIN, HEX

**Response**
- `<data>` = `<STRING RESPONSE DATA>`
  - When form = BIN: Returns the pattern data in binary format.
  - When form = HEX: Returns the pattern data in hexadecimal format.

**Function**
Queries the Match pattern in a block that is set when **Test Pattern** is **Sequence**.

**Example**
To query the Match pattern A in block 3 that is set when **Test Pattern** is **Sequence**:

```
< "1010"
```

**Compatibility**
Incompatible with existing models.

### :SENSe:PATTern:SEQuence:MATCh:MASK <block>,<content>,<form>,<data>

**Parameter**
- `<block>` = `<DECIMAL NUMERIC PROGRAM DATA>`
  - 1 to 128 1 to 128 blocks, 1-block steps
- `<content>` = `<CHARACTER PROGRAM DATA>`
  - APAT Pattern A
  - BPAT Pattern B
- `<form>` = `<CHARACTER PROGRAM DATA>`
  - BIN Binary
  - HEX Hexadecimal
- `<data>` = `<STRING PROGRAM DATA>`
  - When form = BIN: 1 to 64 characters (64-bit data)
  - When form = HEX: 1 to 16 characters (64-bit data)
  - The editable data length depends on the pattern length of Match Pattern.

**Function**
Edits the Mask pattern in a block when **Test Pattern** is **Sequence**.

**Example**
To set two bytes of the Mask Pattern A in block 3 in binary to "1010" when **Test Pattern** is **Sequence**:

```
> :SENSe:PATTern:SEQuence:MATCh:MASK 3,APAT,BIN, "1010"
```

**Compatibility**
Incompatible with existing models.
Chapter 7  SCPI Commands


Parameter
- `<block>` = `<DECIMAL NUMERIC PROGRAM DATA>`
  1 to 128 1 to 128 blocks, 1-block steps
- `<content>` = `<CHARACTER PROGRAM DATA>`
  APAT  Pattern A
  BPAT  Pattern B
- `<form>` = `<CHARACTER PROGRAM DATA>`
  BIN,HEX

Response
- `<data>` = `<STRING RESPONSE DATA>`
  When form = BIN: Returns the pattern data in binary format.
  When form = HEX: Returns the pattern data in hexadecimal format.

Function
Queries the Mask pattern in a block that is set when Test Pattern is Sequence.

Example
To query the Mask pattern A in block 3 that is set when Test Pattern is Sequence:
< "1010"

Compatibility
Incompatible with existing models.

:SENSe:PATTern:SEQuence:CONDition:ADD <block>

Parameter
- `<block>` = `<DECIMAL NUMERIC PROGRAM DATA>` (Block No)
  1 to 128 1 to 128 blocks, 1-block steps

Function
Sets the number of blocks that are edited during Sequence pattern reception.

Example
To set the number of blocks that are edited to 10:
> :SENSe:PATTern:SEQuence:CONDition:ADD 10

Compatibility
Incompatible with existing models.

:SENSe:PATTern:SEQuence:CONDition:ADD?

Response
- `<block>` = `<NR1 NUMERIC RESPONSE DATA>` (Block No)
  1 to 128 1 to 128 block

Function
Queries the number of blocks that are edited during Sequence pattern reception.

Example
> :SENSe:PATTern:SEQuence:CONDition:ADD?
< "10"

Compatibility
Incompatible with existing models.
### :SENSe:PATTern:SEQuence:COPY <block>

**Parameter**  
\(<block> = \langle\text{DECIMAL NUMERIC PROGRAM DATA}\rangle\)  
1 to 128  
1 to 128 blocks, 1-block steps

**Function**  
Copies the setting conditions in the specified block during Sequence pattern reception.

**Example**  
To copy the setting conditions in block 12:  
> :SENSe:PATTern:SEQuence:COPY 12

**Compatibility**  
Incompatible with existing models.

### :SENSe:PATTern:SEQuence:CUT <block>

**Parameter**  
\(<block> = \langle\text{DECIMAL NUMERIC PROGRAM DATA}\rangle\)  
1 to 128  
1 to 128 blocks, 1-block steps

**Function**  
Cuts (copies and removes) the setting conditions in the specified block during Sequence pattern reception.

**Example**  
To cut the setting conditions in block 15:  
> :SENSe:PATTern:SEQuence:CUT 15

**Compatibility**  
Incompatible with existing models.

### :SENSe:PATTern:SEQuence:PASTe <block>

**Parameter**  
\(<block> = \langle\text{DECIMAL NUMERIC PROGRAM DATA}\rangle\)  
1 to 128  
1 to 128 blocks, 1-block steps

**Function**  
Pastes the setting conditions to the specified block during Sequence pattern reception.

**Example**  
To paste the setting conditions to block 20:  
> :SENSe:PATTern:SEQuence:PASTe 20

**Compatibility**  
Incompatible with existing models.

### :SENSe:PATTern:SEQuence:CLEar

**Function**  
Clears all the setting conditions during Sequence pattern reception.

**Example**  
To clear all the setting conditions during Sequence pattern reception:  
> :SENSe:PATTern:SEQuence:CLEar

**Compatibility**  
Incompatible with existing models.
### :SENSe:PAT tern:SEQuence:LENGth <block>,<numeric>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;block&gt; =</td>
<td>&lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
</tr>
<tr>
<td>1 to 128</td>
<td>1 to 128 blocks, 1-block steps</td>
</tr>
<tr>
<td>&lt;numeric&gt; =</td>
<td>&lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
</tr>
<tr>
<td>8192 to 1048576</td>
<td>8192 to 1,048,576 bits, in 128-bit steps</td>
</tr>
</tbody>
</table>

Be sure to set a value so that the following condition is satisfied:

Number of blocks \(\times\) Block length \(\leq\) 128 Mbits

**Function**

Sets the length of the specified block during Sequence pattern reception.

**Example**

To set the length of block 1 to 1 bit during Sequence pattern reception:

\[> \text{:SENSe:PAT tern:SEQuence:LENGth 1, 1} \]

**Compatibility**

Incompatible with existing models.

### :SENSe:PAT tern:SEQuence:LENGth? <block>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;block&gt; =</td>
<td>&lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
</tr>
<tr>
<td>1 to 128</td>
<td>1 to 128 blocks, 1-block steps</td>
</tr>
</tbody>
</table>

**Response**

\(<\text{NR1 NUMERIC RESPONSE DATA}>\)

| 8192 to 1048576    | 8192 to 1,048,576 bits, in 128-bit steps                                   |

**Function**

Queries the length of the specified block during Sequence pattern reception.

**Example**

To query the length of block 1 during Sequence pattern reception:

\[> \text{:SENSe:PAT tern:SEQuence:LENGth? 1} \]
\[< 1 \]

**Compatibility**

Incompatible with existing models.
:SENSe:PATrern:SEQuence:DATA:WHOLe
<block>,<start>,<end>,<data>

Parameter
- `<block>` = `<DECIMAL NUMERIC PROGRAM DATA>`
  1 to 128  1 to 128 blocks, 1-block steps
- `<start>` = `<NON-DECIMAL PROGRAM DATA>`
  #H0 to #HFFFFF  0 to FFFFFF bits, in 1-bit steps
    (Specify in hexadecimal)
- `<end>` = `<NON-DECIMAL PROGRAM DATA>`
  #H0 to #HFFFFF  0 to FFFFFF bits, in 1-bit steps
    (Specify in hexadecimal)

Note:
The maximum setting value is the pattern length.
- `<data>` = `<STRING PROGRAM DATA>`
  "H***...*" Specify pattern data in hexadecimal format.
    1 to 400 characters (pattern data of 400 × 4 bits)
    Specify a character string consisting of 0 to 9 and A to F.
    If the last character string of the pattern data is
    short of a hexadecimal unit (4 bits), add 0(s) up
    to the last bit to make a hexadecimal string.
  "B***...*" Specify pattern data in binary format.
    1 to 400 characters (pattern data of 400 bits)
    Specify a character string consisting of 0 and 1.

Function
Sets the pattern data from the `<start>` to `<end>` addresses during
Sequence pattern reception.
The set data overwrites the specified range.

Example
To set the addresses from bit 0 to bit FFFFF of the block 1 pattern data
in hexadecimal format during Sequence pattern reception:
> :SENSe:PATrern:SEQuence:DATA:WHOLe 1,#H0,#HFFFFF, "H001"

Compatibility
Incompatible with existing models.
### :SENSe:PATTern:SEQuence:DATA:WHOLe? <block>,<start>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;block&gt;</td>
<td>&lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
</tr>
<tr>
<td></td>
<td>1 to 128 blocks, 1-block steps</td>
</tr>
<tr>
<td>&lt;start&gt;</td>
<td>&lt;NON-DECIMAL PROGRAM DATA&gt;</td>
</tr>
<tr>
<td></td>
<td>#H0 to #HFFFFF 0 to FFFFFF bits, in 1-bit steps</td>
</tr>
<tr>
<td></td>
<td>(Specify in hexadecimal)</td>
</tr>
</tbody>
</table>

**Note:**
The maximum setting value is the pattern length.

<table>
<thead>
<tr>
<th>Response</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;data&gt;</td>
<td>&lt;STRING RESPONSE DATA&gt;</td>
</tr>
<tr>
<td></td>
<td>&quot;H***...*&quot; Returns the pattern data in hexadecimal format.</td>
</tr>
<tr>
<td></td>
<td>Within 400 characters (pattern data of 400 x 4 bits)</td>
</tr>
</tbody>
</table>

| Function           | Queries the pattern data of 400 x 4 bits from the <start> address during Sequence pattern reception. |

<table>
<thead>
<tr>
<th>Example</th>
<th>To query the block 1 pattern data of 400 x 4 bits from bit 0 address during Sequence pattern reception:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>&gt; :SENSe:PATTern:SEQuence:DATA:WHOLe? 1,#H0</td>
</tr>
<tr>
<td></td>
<td>&lt; &quot;H001001001001001001001...&quot;</td>
</tr>
</tbody>
</table>

| Compatibility      | Incompatible with existing models.              |
:SENSe:PATTern:SEQuence:BDATa:WHOLe
<block>,<start>,<end>,<bdata>

Parameter

- `<block>` = <DECIMAL NUMERIC PROGRAM DATA>
  1 to 128 1 to 128 blocks, 1-block steps
- `<start>` = <NON-DECIMAL PROGRAM DATA>
  #H0 to #HFFFFF 0 to FFFFF bits, in 1-bit steps
  (Specify in hexadecimal)
- `<end>` = <NON-DECIMAL PROGRAM DATA>
  #H0 to #HFFFFF 0 to FFFFF bits, in 1-bit steps
  (Specify in hexadecimal)

Note:
The maximum setting value is the pattern length.

- `<bdata>` = <ARBITRARY BLOCK PROGRAM DATA>
  #XYYY<binary>
  X: Number of YYY digits
  YYY: Number of bytes of `<binary>`
  1 to 131,072 bytes
  <binary>: Binary data up to 131,072 bytes

Function
Sets the binary data from `<start>` to `<end>` addresses of the pattern data during Sequence pattern reception.
The set data overwrites the specified range.

Example
To set the binary data from bit 0 to bit FFFFF addresses of the block 1 pattern data during Sequence pattern reception:
>`:SENSe:PATTern:SEQuence:BDATa:WHOLe1,#H0,#HFFFFF,#10011`

Compatibility
Incompatible with existing models.
Chapter 7  SCPI Commands

:SENSe:PATTern:SEQuence:BDATa:WHOLe? <block>,<start>[,<size>]

Parameter

- <block> = <DECIMAL NUMERIC PROGRAM DATA>
  1 to 128  1 to 128 blocks, 1-block steps
- <start> = <NON-DECIMAL PROGRAM DATA>
  #H0 to #HFFFFF  0 to FFFFF bits, in 1-bit steps
  (Specify in hexadecimal)

Note:

The maximum setting value is the pattern length.

- [<size>] = <NR1 NUMERIC PROGRAM DATA>
  1 to 131072  1 to 131,072 bytes, in 1-byte steps

Response

- <bdata> = <DEFINITE LENGTH ARBITRARY BLOCK RESPONSE DATA>
  #XYYY<binary>  X: Number of YYY digits
  YYY: Number of bytes of <binary>
  1 to 131,072 bytes
  <binary>: Binary data up to 131,072 bytes

Function

Queries binary data of bytes specified by <size> from the <start> address of the pattern data during Sequence pattern reception.

Example

To query binary data of 16000 bytes of the block 1 pattern data starting from address bit 0 during Sequence pattern reception:

> :SENSe:PATTern:SEQuence:BDATa:WHOLe? 1,#H0,16000
< #10011

Compatibility

Incompatible with existing models.

:SENSe:PATTern:SEQuence:DREVerse:ADDRess <block>,<start>,<end>

Parameter

- <block> = <DECIMAL NUMERIC PROGRAM DATA>
  1 to 128  1 to 128 blocks, 1-block steps
- <start> = <NON-DECIMAL PROGRAM DATA>
  #H0 to #HFFFFF  0 to FFFFF bits, in 1-bit steps
  (Specify in hexadecimal)
- <end> = <NON-DECIMAL PROGRAM DATA>
  #H0 to #HFFFFF  0 to FFFFF bits, in 1-bit steps
  (Specify in hexadecimal)

Note:

The maximum setting value is the pattern length.

Function

Inverts the data in the program during Sequence pattern reception.

Specify the inversion range by the <start> and <end> addresses.

Example

To invert the data in the program of block 1 from bit 0 to bit FFFFF addresses during Sequence pattern reception:

> :SENSe:PATTern:SEQuence:DREVerse:ADDRess 1,#H0,#HFFFFF
< #10011

Compatibility

Incompatible with existing models.
:SENSe:PATTern:SEQuence:DREVerse:DELTa <block>,<start>,<delta>

Parameter

- **<block>** = <DECIMAL NUMERIC PROGRAM DATA>
  1 to 128 1 to 128 blocks, 1-block steps
- **<start>** = <NON-DECIMAL PROGRAM DATA>
  #H0 to #FFFFF 0 to FFFFFF bits, in 1-bit steps
  (Specify in hexadecimal)
- **<delta>** = <DECIMAL NUMERIC PROGRAM DATA>
  1 to 1048576 1 to 1,048,576 bits, in 1-bit steps

**Note:**

The maximum setting value is the pattern length.

Function

Inverts the data in the program during Sequence pattern reception.

Specify the inversion range by the number of bits from the **<start>** address (relative specification).

Example

To invert 1 bit of block 1 data from address bit 0 in the program during Sequence pattern reception:

> :SENSe:PATTern:SEQuence:DREVerse:DELTa 1,#H0,1

Compatibility

Incompatible with existing models.

:SENSe:PATTern:SEQuence:DATA:FILL <block>,<range>,<page>,<data>

Parameter

- **<block>** = <DECIMAL NUMERIC PROGRAM DATA>
  1 to 128 1 to 128 blocks, 1-block steps
- **<range>** = <CHARACTER PROGRAM DATA>
  PAGE Specifies a page.
  (One page is defined as 128 bits.)
  ALL Specifies all data.
- **<page>** = <DECIMAL NUMERIC PROGRAM DATA>
  1 to 8,192 1 to 8,192 (Max.) pages, in 1-page steps
  Max = \[
  \frac{\text{Pattern Length}}{128}
  \] (rounding up fractions)
  Specify "0" when **<range>** is set to ALL.
- **<data>** = <DECIMAL NUMERIC PROGRAM DATA>
  0 Fills the specified range with 0s.
  1 Fills the specified range with 1s.

Function

Sets 0s or 1s for the specified page or all data in the program during Sequence pattern reception.

Example

To set 0s for the first page of the program during Sequence pattern reception:

> :SENSe:PATTern:SEQuence:DATA:FILL 1,PAGE,1,0

Compatibility

Incompatible with existing models.
Chapter 7  SCPI Commands

7.4.3.7  Commands related to Mask

Figure 7.4.3.7-1  Mask ON/OFF setting

Figure 7.4.3.7-2  Bit Window Setup dialog box (for PRBS pattern)

Figure 7.4.3.7-3  Bit Window and Block Window settings in Pattern Editor screen
(for Zero Substitution, Data, Mixed Data, and Sequence patterns)
## 7.4 ED Commands

### Table 7.4.3.7-1 Mask setting commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ON/OFF</td>
<td>:SENSe:PATTern:MASK:BLOCk?</td>
</tr>
<tr>
<td></td>
<td>ON/OFF</td>
<td>:SENSe:PATTern:MASK:BIT?</td>
</tr>
<tr>
<td></td>
<td>ON/OFF</td>
<td>:SENSe:PATTern:MASK:EXTernal?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:PATTern:MASK:BLKWindow?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:PATTern:MASK:BTWindow?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:PATTern:MASK:BDATa:WHOLe?</td>
</tr>
<tr>
<td>[8]</td>
<td>Block Window (Data inversion)</td>
<td>:SENSe:PATTern:MASK:DREVerse:DELTa</td>
</tr>
<tr>
<td>[9]</td>
<td>Block Window (All 0 or All 1)</td>
<td>:SENSe:PATTern:MASK:DATA:FILL</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:PATTern:MASK:MIXData:BLKWindow?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:PATTern:MASK:MIXData:BDATa:WHOLe?</td>
</tr>
<tr>
<td>[14]</td>
<td>Block Window (All 0 or All 1)</td>
<td>:SENSe:PATTern:MASK:MIXData:DATA:FILL</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:PATTern:MASK:SEQUence:BLKWindow?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:PATTern:MASK:SEQUence:BDATa:WHOLe?</td>
</tr>
<tr>
<td>[18]</td>
<td>Block Window (Data inversion)</td>
<td>:SENSe:PATTern:MASK:SEQUence:DREVerse:DELTa</td>
</tr>
<tr>
<td>[19]</td>
<td>Block Window (All 0 or All 1)</td>
<td>:SENSe:PATTern:MASK:SEQUence:DATA:FILL</td>
</tr>
</tbody>
</table>

**:SENSe:PATTern:MASK:BLOCk <boolean>**

**Parameter**

- `<boolean> = <BOOLEAN PROGRAM DATA>`
- 0 or OFF: Block Window OFF
- 1 or ON: Block Window ON

**Function**

Sets Block Window ON/OFF to the received pattern.

**Example**

To set Block Window ON to the received pattern:

```plaintext
> :SENSe:PATTern:MASK:BLOCk ON
```

**Compatibility**

Incompatible with existing models.
### :SENSe:PATTern:MASK:BLOCk?

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;numeric&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Block Window OFF</td>
</tr>
<tr>
<td>1</td>
<td>Block Window ON</td>
</tr>
</tbody>
</table>

**Function:** Queries the Block Window ON/OFF state of the received pattern.

**Example:**
```
> :SENSe:PATTern:MASK:BLOCk?
< 1
```

**Compatibility:** Incompatible with existing models.

### :SENSe:PATTern:MASK:BIT <boolean>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;boolean&gt; = &lt;BOOLEAN PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 or OFF</td>
<td>Bit Window OFF</td>
</tr>
<tr>
<td>1 or ON</td>
<td>Bit Window ON</td>
</tr>
</tbody>
</table>

**Function:** Sets Bit Window ON/OFF to the received pattern.

**Example:**
```
To set Bit Window ON to the received pattern:
> :SENSe:PATTern:MASK:BIT ON
```

**Compatibility:** Incompatible with existing models.

### :SENSe:PATTern:MASK:BIT?

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;numeric&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Bit Window OFF</td>
</tr>
<tr>
<td>1</td>
<td>Bit Window ON</td>
</tr>
</tbody>
</table>

**Function:** Queries the Bit Window ON/OFF state of the received pattern.

**Example:**
```
To query the Bit Window ON/OFF state of the received pattern.
> :SENSe:PATTern:MASK:BIT?
< 1
```

**Compatibility:** Incompatible with existing models.

### :SENSe:PATTern:MASK:EXTernal <boolean>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;boolean&gt; = &lt;BOOLEAN PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 or OFF</td>
<td>External Mask OFF</td>
</tr>
<tr>
<td>1 or ON</td>
<td>External Mask ON</td>
</tr>
</tbody>
</table>

**Function:** Sets External Mask ON/OFF to the received pattern.

**Example:**
```
To set External Mask ON to the received pattern:
> :SENSe:PATTern:MASK:EXTernal ON
```

**Compatibility:** Incompatible with existing models.
### :SENSe:PATTern:MASK:EXTernal?

**Response**

\[
\text{<numeric> = <NR1 NUMERIC RESPONSE DATA>}
\]

- 0: External Mask OFF
- 1: External Mask ON

**Function**

Queries the External Mask ON/OFF state of the received pattern.

**Example**

> :SENSe:PATTern:MASK:EXTernal?

< 1

**Compatibility**

Incompatible with existing models.

### :SENSe:PATTern:MASK:BLKWindow <start>,<end>,<data>

**Parameter**

- `<start>` = `<NON-DECIMAL PROGRAM DATA>`
- #H0 to #H7FFFFFF: 0 to 7FFFFFF bits, in 1-bit steps
  (Specify in hexadecimal)
- `<end>` = `<NON-DECIMAL PROGRAM DATA>`
- #H0 to #H7FFFFFF: 0 to 7FFFFFF bits, in 1-bit steps
  (Specify in hexadecimal)

**Note:**

- The maximum setting is the pattern length.
- At 2 ch Combination, the setting range is doubled.
- At 4 ch Combination, the setting range is quadrupled.

- `<data>` = `<STRING PROGRAM DATA>`
  - "H***...*": Specify pattern data in hexadecimal format.
    1 to 400 characters (pattern data of 400 \( \times \) 4 bits)
    Specify a character string consisting of 0 to 9 and A to F.
  - "B***...*": Specify pattern data in binary format.
    1 to 400 characters (pattern data of 400 bits)
    Specify a character string consisting of 0 and 1.

**Function**

Sets the mask pattern data from the `<Start>` to `<end>` addresses during Block Window setting.

The set data overwrites the specified range.

**Example**

To set the mask pattern data from bit 0 to bit 7FFFFFF addresses in hexadecimal format during Block Window setting:

> :SENSe:PATTern:MASK:BLKWindow #H0,#H7FFFFFF, "H001"

**Compatibility**

Incompatible with existing models.
Chapter 7  SCPI Commands

:SENSe:PATTern:MASK:BLKWindow? <start>

Parameter  

<start>=<NON-DECIMAL PROGRAM DATA>

#H0 to #H7FFFFFF  0 to 7FFFFFFF bits, in 1-bit steps (Specify in hexadecimal)

Note:
The maximum setting is the pattern length.
At 2 ch Combination, the setting range is doubled.
At 4 ch Combination, the setting range is quadrupled.

Response

<data> = <STRING RESPONSE DATA>

"H***...*" Returns the pattern data in hexadecimal format.
Within 400 characters (pattern data of 400 \times 4 bits)

Function

Queries the mask pattern data of 400 \times 4 bits from the <Start> address during Block Window setting.

Example

To query the mask pattern data of 400 \times 4 bits from bit 0 address during Block Window setting:

> :SENSe:PATTern:MASK:BLKWindow? #H0

< "H001"

Compatibility

Incompatible with existing models.

:SENSe:PATTern:MASK:BTWindow <data>

Parameter

<data> = <STRING PROGRAM DATA>

"H***...*" Specify pattern data in hexadecimal format.
1 to 8 characters (pattern data of 32 bits)
Specify a character string consisting of 0 to 9 and A to F.

"B***...*" Specify pattern data in binary format.
1 to 32 characters (pattern data of 32 bits)
Specify a character string consisting of 0 and 1.

Function

Sets the mask pattern data during Bit Window setting.

Example

To set the mask pattern data in hexadecimal format during Bit Window setting.

> :SENSe:PATTern:MASK:BTWindow "H001"

Compatibility

Incompatible with existing models.
7.4 ED Commands

:SENSe:PATTern:MASK:BTWindow?

Response

- `<data> = <STRING RESPONSE DATA>`
- "H***...*"
  - Returns the pattern data in hexadecimal format.
  - Within 8 characters (pattern data of 32 bits)

Function

Queries the mask pattern data during Bit Window setting.

Example

> :SENSe:PATTern:MASK:BTWindow?
< "#00100100"

Compatibility

Incompatible with existing models.

:SENSe:PATTern:MASK:BDATa:WHOLe <start>,<end>,<bdata>

Parameter

- `<start> = <NON-DECIMAL PROGRAM DATA >`
  - #H0 to #H7FFFFFF 0 to 7FFFFFFF bits, in 1-bit steps
    (Specify in hexadecimal)
- `<end> = <NON-DECIMAL PROGRAM DATA >`
  - #H0 to #H7FFFFFF 0 to 7FFFFFFF bits, in 1-bit steps
    (Specify in hexadecimal)
- `<bdata> = <ARBITRARY BLOCK PROGRAM DATA >`
  - #XYYY<binary> X: Number of YYY digits
    YYY: Number of bytes of <binary>
    1 to 16,777,216 bytes
    <binary>: Binary data up to 16,777,216 bytes

Note:

- The maximum setting is the pattern length.
- At 2 ch Combination, the setting range is doubled.
- At 4 ch Combination, the setting range is quadrupled.

Function

Sets binary data of the mask pattern data from `<start>` to `<end>` addresses during Block Window setting.
The set data overwrites the specified range.

Example

To set binary data of the mask pattern data from bit 0 to bit 7FFFFFFF addresses during Block Window setting:

> :SENSe:PATTern:MASK:BDATa:WHOLe #H0,#H7FFFFFF,#11A

Compatibility

Incompatible with existing models.
Chapter 7  SCPI Commands

:SENSe:PATTern:MASK:BDATa:WHOLe? <start>[,<size>]

Parameter

<start> = <NON-DECIMAL PROGRAM DATA>
#H0 to #H7FFFFFF 0 to 7FFFFFFF bits, in 1-bit steps
(Specify in hexadecimal)

<size> = <NR1 NUMERIC PROGRAM DATA>
1 to 1677216 1 to 1677216 bytes, in 1-byte steps

Note:
The maximum setting is the pattern length.
At 2 ch Combination, the setting range is doubled.
At 4 ch Combination, the setting range is quadrupled.

Response

<bdata> = <DEFINITE LENGTH ARBITRARY BLOCK RESPONSE DATA>
#XYYY<binary> X: Number of YYY digits
YYY: Number of bytes of <binary>
1 to 16,777,216 bytes
<binary>: Binary data of up to 16,777,216 bytes

Binary data of the setting pattern is queried when [size] is omitted.

Function

Queries binary data of bytes specified by <size> from the <start> address
of the mask pattern data during Block Window setting.

Example

To query binary data of 2 bytes from bit 0 address of the pattern data
during Block Window setting:
> :SENSe:PATTern:MASK:BDATa:WHOLe? #H0,2
<#12AA

To query binary data of the setting pattern from bit 0 address of the
pattern data during Block Window setting (in the case of [size] is
omitted):
> :SENSe:PATTern:MASK:BDATa:WHOLe? #H0
< #516000AAAAAA… (binary data of 16,000 bytes)

Compatibility

Incompatible with existing models.
### :SENSe:PATTern:MASK:DREVerse:ADDRess \(<start>,<end>\)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(&lt;start&gt;)</td>
<td>(&lt;\text{NON-DECIMAL PROGRAM DATA}&gt;)</td>
</tr>
<tr>
<td>(#H0) to (#H7FFFFFF)</td>
<td>0 to 7FFFFFFF bits, in 1-bit steps (Specify in hexadecimal)</td>
</tr>
<tr>
<td>(&lt;end&gt;)</td>
<td>(&lt;\text{NON-DECIMAL PROGRAM DATA}&gt;)</td>
</tr>
<tr>
<td>(#H0) to (#H7FFFFFF)</td>
<td>0 to 7FFFFFFF bits, in 1-bit steps (Specify in hexadecimal)</td>
</tr>
</tbody>
</table>

**Note:**
- The maximum setting is the pattern length.
- At 2 ch Combination, the setting range is doubled.
- At 4 ch Combination, the setting range is quadrupled.

**Function:**
Inverts the mask pattern data during Block Window setting. Specify the inversion range by the \(<\text{start}>\) and \(<\text{end}>\) addresses.

**Example:**
To invert addresses from bit 0 to bit 7FFFFFFF during Block Window setting:

> :SENSe:PATTern:MASK:DREVerse:ADDRess \(#H0,#H7FFFFFF\)

**Compatibility:**
Incompatible with existing models.

### :SENSe:PATTern:MASK:DREVerse:DELTa \(<start>,<\text{delta}>\)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(&lt;\text{start}&gt;)</td>
<td>(&lt;\text{NON-DECIMAL PROGRAM DATA}&gt;)</td>
</tr>
<tr>
<td>(#H0) to (#H7FFFFFF)</td>
<td>0 to 7FFFFFFF bits, in 1-bit steps (Specify in hexadecimal)</td>
</tr>
<tr>
<td>(&lt;\text{delta}&gt;)</td>
<td>(&lt;\text{DECIMAL NUMERIC PROGRAM DATA}&gt;)</td>
</tr>
<tr>
<td>1 to 134217728</td>
<td>1 to 134,217,728 bits, in 1-bit steps</td>
</tr>
</tbody>
</table>

**Note:**
- The maximum setting is the pattern length.
- At 2 ch Combination, the setting range is doubled.
- At 4 ch Combination, the setting range is quadrupled.

**Function:**
Inverts the mask pattern data during Block Window setting. Specify the inversion range by the number of bits from the \(<\text{start}>\) address (relative specification).

**Example:**
To invert 1 bit of the mask pattern data from address 0 during Block Window setting:

> :SENSe:PATTern:MASK:DREVerse:DELTa \(#H0,1\)

**Compatibility:**
Incompatible with existing models.
**:SENSe:PATTERN:MASK:DATA:FILL <range>,<page>,<data>**

**Parameter**

- **<range>** = <CHARACTER PROGRAM DATA>
  - PAGE: Specifies a page.
    (One page is defined as 128 bits.)
  - ALL: Specifies all data.
- **<page>** = <DECIMAL NUMERIC PROGRAM DATA>
  - 1 to 1048576: 1 to 1,048,576 (Max.) pages, in 1-page steps
  - Max = \(\frac{\text{Pattern Length}}{128}\) (rounding up fractions)
- **<data>** = <DECIMAL NUMERIC PROGRAM DATA>
  - 0: Fills the specified range with 0s.
  - 1: Fills the specified range with 1s.

**Function**

Sets 0s or 1s for the specified page or all data in mask pattern data during Block Window setting.

**Example**

To set 0s for the first page of the mask pattern data during Block Window

`> :SENSe:PATTERn:MASK:DATA:FILL PAGE,1,0`

**Compatibility**

Incompatible with existing models.
7.4 ED Commands

**SENSe:PATTern:MASK:MIXData:BLKWindow**

### Parameter

- `<block>` = `<DECIMAL NUMERIC PROGRAM DATA>`
  - 1 to 511
  - 1 to 511 blocks, in 1-block steps
- `<start>` = `<NON-DECIMAL PROGRAM DATA>`
  - #H0 to #H7FFFFFF
  - 0 to 7FFFFFF bits, in 1-bit steps
  - (Specify in hexadecimal)
- `<end>` = `<NON-DECIMAL PROGRAM DATA>`
  - #H0 to #H7FFFFFF
  - 0 to 7FFFFFF bits, in 1-bit steps
  - (Specify in hexadecimal)

**Note:**

- The maximum setting is the pattern length.
- At 2 ch Combination, the setting range is doubled.
- At 4 ch Combination, the setting range is quadrupled.

- `<data>` = `<STRING PROGRAM DATA>`
  - "H***...*"
  - Specify pattern data in hexadecimal format.
  - 1 to 400 characters (pattern data of 400 × 4 bits)
  - Specify a character string consisting of 0 to 9 and A to F.
  - If the last character string of the pattern data is short of a hexadecimal unit (4 bits), add 0(s) up to the last bit to make a hexadecimal string.
  - "B***...*"
  - Specify pattern data in binary format.
  - 1 to 400 characters (pattern data of 400 bits)
  - Specify a character string consisting of 0 and 1.

### Function

Sets the mask pattern data in a block from the `<start>` to `<end>` addresses when **Test Sequence** is **Mixed** and **Block Window** is set to **OFF**. The set data overwrites the specified range.

### Example

To set the addresses from bit 0 to bit 7FFFFFF of the mask pattern data in block 3, in hexadecimal:

> :SENSe:PATTern:MASK:MIXData:BLKWindow3,#H0,#H7FFFFFF,"H001"

### Compatibility

Incompatible with existing models.
:SENSe:PATTern:MASK:MIXData:BLKWindow? <block>,<start>

Parameter

- `<block>` = <DECIMAL NUMERIC PROGRAM DATA>
  1 to 511 blocks, in 1-block steps
- `<start>` = <NON-DECIMAL PROGRAM DATA >
  #H0 to #H7FFFFFF 0 to 7FFFFFFF bits, in 1-bit steps
  (Specify in hexadecimal)

**Note:**

The maximum setting is the pattern length.
At 2 ch Combination, the setting range is doubled.
At 4 ch Combination, the setting range is quadrupled.

Response

- `<data>` = <STRING RESPONSE DATA>
  "H***…*" Returns the pattern data in hexadecimal format.
  Within 400 characters (pattern data of 400 × 4 bits)

Function

Queries the mask pattern data of 400 × 4 bits in a block from the `<start>` address when Test Sequence is Mixed and Block Window is set to OFF.

Example

To query the mask pattern data of 400 × 4 bits in block 3 from the bit 0 address:

> :SENSe:PATTern:MASK:MIXData:BLKWindow? 3,#H0

"H001001001001001001…"

Compatibility

Incompatible with existing models.
### :SENSe:PATTern:MASK:MIXData:BDATa:WHOLe

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;block&gt;</td>
<td>&lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
</tr>
<tr>
<td></td>
<td>1 to 511 blocks, in 1-block steps</td>
</tr>
<tr>
<td>&lt;start&gt;</td>
<td>&lt;NON-DECIMAL PROGRAM DATA&gt;</td>
</tr>
<tr>
<td></td>
<td>#H0 to #H7FFFFFF 0 to 7FFFFFFF bits, in 1-bit steps (Specify in hexadecimal)</td>
</tr>
<tr>
<td>&lt;end&gt;</td>
<td>&lt;NON-DECIMAL PROGRAM DATA&gt;</td>
</tr>
<tr>
<td></td>
<td>#H0 to #H7FFFFFFF 0 to 7FFFFFFF bits, in 1-bit steps (Specify in hexadecimal)</td>
</tr>
</tbody>
</table>

**Note:**
- The maximum setting is the pattern length.
- At 2 ch Combination, the setting range is doubled.
- At 4 ch Combination, the setting range is quadrupled.

<table>
<thead>
<tr>
<th>&lt;bdata&gt;</th>
<th>&lt;ARBITRARY BLOCK PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>#XYYY&lt;binary&gt;</td>
<td>X: Number of YYY digits</td>
</tr>
<tr>
<td></td>
<td>YYY: Number of bytes of &lt;binary&gt;</td>
</tr>
<tr>
<td></td>
<td>1 to 16,777,216 bytes</td>
</tr>
</tbody>
</table>

**Function:**
Sets binary data of the Block Window mask pattern data <start> to <end> addresses when Test Sequence is **Mixed**.
The set data overwrites the specified range.

**Example:**
To set binary data of the Block Window mask pattern data from bit 0 to bit 7FFFFFFF addresses when Test Sequence is **Mixed**:

```
>:SENSe:PATTern:MASK:MIXData:BDATa:WHOLe 1,#H0,#H7FFFFFFF,#11A
```

**Compatibility:**
Incompatible with existing models.
Chapter 7  SCPI Commands

:SENSe:PATTern:MASK:MIXData:BDATa:WHOLe? <block>,<start>[,<size>]

Parameter

- `<block>` = `<DECIMAL NUMERIC PROGRAM DATA>`
  - 1 to 511 1 to 511 blocks, in 1-block steps
- `<start>` = `<NON-DECIMAL PROGRAM DATA>`
  - `#H0` to `#H7FFFFFF` 0 to 7FFFFFF bits, in 1-bit steps
  (Specify in hexadecimal)

Note:

- The maximum setting is the pattern length.
- At 2 ch Combination, the setting range is doubled.
- At 4 ch Combination, the setting range is quadrupled.

- `[<size>]` = `<NR1 NUMERIC PROGRAM DATA>`
  - 1 to 1677216 1 to 1,677,216 bytes/1 byte Step

Response

- `<bdata>` = `<DEFINITE LENGTH ARBITRARY BLOCK RESPONSE DATA>`
  - `#XYYY<binary>`
    - `X`: Number of `YYY` digits
    - `YYY`: Number of bytes of `<binary>`
    - 1 to 16,777,216 bytes

  - `<binary>`: Binary data up to 16,777,216 bytes

Function

Queries binary data of bytes specified by `<size>` from the `<start>` address of the Block Window mask pattern data when Test Sequence is Mixed.

Example

To query binary data of up to 2 bytes from bit 0 address of the Block Window mask pattern data when Test Sequence is Mixed:

```
> :SENSe:PATTern:MASK:MIXData:BDATa:WHOLe? 1,#H0,2
< #12AA
```

To query binary data of the setting pattern from bit 0 address of the Block Window mask pattern data when Test Sequence is Mixed (in the case of `[<size>]` is omitted):

```
> :SENSe:PATTern:MASK:BDATa:WHOLe? 1,#H0
< #516000AAAAAA...
```

Compatibility

Incompatible with existing models.
### 7.4 ED Commands

**SENSe:PATTern:MASK:MIXData:DREVerse:ADDRess**<br>`<block>,<start>,<end>`

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;block&gt;</code></td>
<td><code>&lt;DECIMAL NUMERIC PROGRAM DATA&gt;</code> 1 to 511 blocks, in 1-block steps</td>
</tr>
<tr>
<td><code>&lt;start&gt;</code></td>
<td><code>&lt;NON-DECIMAL PROGRAM DATA&gt;</code> #H0 to #H7FFFFFF 0 to 7FFFFFFF bits, in 1-bit steps (Specify in hexadecimal)</td>
</tr>
<tr>
<td><code>&lt;end&gt;</code></td>
<td><code>&lt;NON-DECIMAL PROGRAM DATA&gt;</code> #H0 to #H7FFFFFFF 0 to 7FFFFFFF bits, in 1-bit steps (Specify in hexadecimal)</td>
</tr>
</tbody>
</table>

**Note:**

The maximum setting is the pattern length. At 2 ch Combination, the setting range is doubled. At 4 ch Combination, the setting range is quadrupled.

**Function**

Inverts the Block Window mask pattern data when Test Sequence is Mixed. Specify the inversion range by the `<start>` and `<end>` addresses.

**Example**

To invert the Block Window mask pattern data from bit 0 to bit 7FFFFFF addresses when Test Sequence is Mixed:

```plaintext
>:SENSe:PATTern:MASK:MIXData:DREVerse:ADDRess 1,#H0,#H7FFFFFF
```

**Compatibility**

Incompatible with existing models.

**SENSe:PATTern:MASK:MIXData:DREVerse:DELTa**<br>`<block>,<start>,<delta>`

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;block&gt;</code></td>
<td><code>&lt;DECIMAL NUMERIC PROGRAM DATA&gt;</code> 1 to 511 blocks, in 1-block steps</td>
</tr>
<tr>
<td><code>&lt;start&gt;</code></td>
<td><code>&lt;NON-DECIMAL PROGRAM DATA&gt;</code> #H0 to #H7FFFFFF 0 to 7FFFFFFF bits, in 1-bit steps (Specify in hexadecimal)</td>
</tr>
<tr>
<td><code>&lt;delta&gt;</code></td>
<td><code>&lt;DECIMAL NUMERIC PROGRAM DATA&gt;</code> 1 to 134217728 1 to 134,217,728 bits, in 1-bit steps</td>
</tr>
</tbody>
</table>

**Note:**

The maximum setting is the pattern length. At 2 ch Combination, the setting range is doubled. At 4 ch Combination, the setting range is quadrupled.

**Function**

Inverts the Block Window mask pattern data when Test Sequence is Mixed. Specify the inversion range by the number of bits from the `<start>` address (relative specification).

**Example**

To invert 1 bit of the Block Window mask pattern data from address 0 when Test Sequence is Mixed:

```plaintext
>:SENSe:PATTern:MASK:MIXData:DREVerse:DELTa 1,#H0,1
```

**Compatibility**

Incompatible with existing models.
### :SENSe:PATTern:MASK:MIXData:DATA:FILL

**<block>,<range>,<page>,<data>**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;block&gt;&lt;DECIMAL NUMERIC PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1 to 511 1 to 511 blocks, in 1-block steps</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;range&gt;=&lt;CHARACTER PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>PAGE</td>
<td>Specifies a page. (One page is defined as 128 bits.)</td>
</tr>
<tr>
<td>ALL</td>
<td>Specifies all data.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;page&gt;&lt;DECIMAL NUMERIC PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1 to 1,048,576 (Max.) pages, in 1-page steps</td>
</tr>
</tbody>
</table>

\[
\text{Max} = \left\lfloor \frac{\text{Pattern Length}}{128} \right\rfloor \text{ (rounding up fractions)}
\]

Specify "0" when <range> is set to ALL.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;data&gt;&lt;DECIMAL NUMERIC PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Fills the specified range with 0s.</td>
</tr>
<tr>
<td>1</td>
<td>Fills the specified range with 1s.</td>
</tr>
</tbody>
</table>

**Function**
Sets 0s or 1s for the specified page or all data in the Block Window mask pattern data when **Test Sequence** is **Mixed**.

**Example**
To set 0s for the first page of the Block Window mask pattern data during

```
> :SENSe:PATTern:MASK:MIXData:DATA:FILL 1, PAGE, 1, 0
```

**Compatibility**
Incompatible with existing models.
:SENSe:PATTern:MASK:SEQuence:BLKWindow
[block],<start>,<end>,<data>

Parameter

- `<block>` = `<CHARACTER PROGRAM DATA>`
  1 to 128 1 to 128 blocks, 1-block steps
- `<start>` = `<NON-DECIMAL PROGRAM DATA>`
  #H0 to #HFFFFF 0 to FFFFFF bits, in 1-bit steps
  (Specify in hexadecimal)
- `<end>` = `<NON-DECIMAL PROGRAM DATA>`
  #H0 to #HFFFFF 0 to FFFFFF bits, in 1-bit steps
  (Specify in hexadecimal)

Note:
The maximum setting value is the pattern length.

- `<data>` = `<STRING PROGRAM DATA>`
  "H***...*" Specify pattern data in hexadecimal format.
  1 to 400 characters (pattern data of 400 × 4 bits)
  Specify a character string consisting of 0 to 9 and A to F.
  "B***...*" Specify pattern data in binary format.
  1 to 400 characters (pattern data of 400 bits)
  Specify a character string consisting of 0 and 1.

Function
Sets the mask pattern data in a Block from the `<start>` to `<end>`
addresses when Test Pattern is Sequence or Block Window setting.
The set data overwrites the specified range.

Example
To set the addresses from bit 0 to bit FFFFFF of the mask pattern data in
block 3 during Block Window setting in hexadecimal:
> :SENSe:PATTern:MASK:SEQuence:BLKWindow 3,#H0,#HFFFFF,"H001"

Compatibility
Incompatible with existing models.
Chapter 7  SCPI Commands

:SENSe:PATTern:MASK:SEQuence:BLKWindow? <block>,<start>

Parameter

- `<block>` = <CHARACTER PROGRAM DATA>
  1 to 128  1 to 128 blocks, 1-block steps
- `<start>` = <NON-DECIMAL PROGRAM DATA>
  #H0 to #HFFFFF  0 to FFFFFF bits, in 1-bit steps
  (Specify in hexadecimal)

Note:
The maximum setting value is the pattern length.

Response

- `<data>` = <STRING RESPONSE DATA>
  "H***…*"  Returns the pattern data in hexadecimal format.
  Within 400 characters (pattern data of 400 × 4 bits)

Function

Queries the mask pattern data of 400 × 4 bits in a block from the `<start>` address when Test Pattern is Sequence or Block Window setting.

Example

To query the mask pattern data of 400 × 4 bits in block 3 from the bit 0 address during Block Window setting.

```
> :SENSe:PATTern:MASK:SEQuence:BLKWindow? 3,#H0
< "H001..."
```

Compatibility

Incompatible with existing models.
:SENSe:PATTern:MASK:SEQUence:BDATa:WHOLe
[block],<start>,<end>,<bdata>

Parameter
<brick> = <DECIMAL NUMERIC PROGRAM DATA>
1 to 128 1 to 128 blocks, 1-block steps
<start> = <NON-DECIMAL PROGRAM DATA>
#H0 to #HFFFFF 0 to FFFFFF bits, in 1-bit steps
(Specify in hexadecimal)
<end> = <NON-DECIMAL PROGRAM DATA>
#H0 to #HFFFFF 0 to FFFFFF bits, in 1-bit steps
(Specify in hexadecimal)

Note:
The maximum setting value is the pattern length.

<bdata> = <ARBITRARY BLOCK PROGRAM DATA>
#XYYY<binary> X: Number of YYY digits
YYY: Number of bytes of <binary>
1 to 131,072 bytes
<binary>: Binary data up to 131,072 bytes

Function
Sets binary data of the Block Window mask pattern data from <start> to
<end> addresses when Test Pattern is Sequence.
The set data overwrites the specified range.

Example
To set binary data of the Block Window mask pattern data from bit 0 to
bit 7FFFFFFFF addresses when Test Pattern is Sequence.
>:SENSe:PATTern:MASK:SEQUence:BDATa:WHOLe
1,#H0,#7FFFFFFF,#11A

Compatibility
Incompatible with existing models.
Chapter 7  SCPI Commands

:SENSE:_PATTERN:MASK:SEQUence:BDATa:WHOLe?
<block>,<start>[,<size>]

Parameter

<block> = <DECIMAL NUMERIC PROGRAM DATA>
1 to 128 1 to 128 blocks, 1-block steps
<brstart> = <NON-DECIMAL PROGRAM DATA >
#H0 to #HFFFFF 0 to FFFFFF bits, in 1-bit steps
(Specify in hexadecimal)

Note:
The maximum setting value is the pattern length.

[size] = <NR1 NUMERIC PROGRAM DATA>
1 to 131072 1 to 131,072 bytes/1 byte Step

Response
<brdata> = <DEFINITE LENGTH ARBITRARY BLOCK RESPONSE DATA>
#XYYY<binary> X: Number of YYY digits
YYY: Number of bytes of <binary>
1 to 131,072 bytes
<binary>: Binary data up to 131,072 bytes
Binary data of the setting pattern is queried when [size] is omitted.

Function
Queries binary data of bytes specified by <size> from the <start> address
of the Block Window mask pattern data when Test Pattern is Sequence.

Example
To query binary data of up to 2 bytes from bit 0 the <Start> address
of the Block Window mask pattern data when Test Pattern is Sequence.
> :SENSe:PATTern:MASK:SEQUence:BDATa:WHOLe? 1,#H0,2
<#12AA
To query binary data of the setting pattern from bit 0 address of the
Block Window mask pattern data when Test Pattern is Sequence (in the
case of [size] is omitted):
> :SENSe:PARAMeter:MASK:SEQUence:BDATa:WHOLe? 1,#H0
< #516000000000... (binary data of 16,000 bytes)

Compatibility
Incompatible with existing models.
### 7.4 ED Commands


**Parameter**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;block&gt;</td>
<td>= &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
</tr>
<tr>
<td></td>
<td>1 to 128</td>
</tr>
<tr>
<td>&lt;start&gt;</td>
<td>= &lt;NON-DECIMAL PROGRAM DATA&gt;</td>
</tr>
<tr>
<td></td>
<td>#H0 to #HFFFFF</td>
</tr>
<tr>
<td>&lt;end&gt;</td>
<td>= &lt;NON-DECIMAL PROGRAM DATA&gt;</td>
</tr>
<tr>
<td></td>
<td>#H0 to #HFFFFF</td>
</tr>
</tbody>
</table>

**Function**

Inverts the Block Window mask pattern data when Test Pattern is Sequence. Specify the inversion range by the <start> and <end> addresses.

**Example**

To invert the Block Window mask pattern from bit 0 to bit FFFFFF addresses when Test Pattern is Sequence:

```plaintext
> :SENSe:PAReM:MASK:SEQuence:DREVerse:ADDRess 1,#H0,#HFFFFF
```

**Compatibility**

Incompatible with existing models.

#### :SENSe:PATTern:MASK:SEQuence:DREVerse:DELTa

**Parameter**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;block&gt;</td>
<td>= &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
</tr>
<tr>
<td></td>
<td>1 to 128</td>
</tr>
<tr>
<td>&lt;start&gt;</td>
<td>= &lt;NON-DECIMAL PROGRAM DATA&gt;</td>
</tr>
<tr>
<td></td>
<td>#H0 to #HFFFFF</td>
</tr>
<tr>
<td>&lt;delta&gt;</td>
<td>= &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
</tr>
<tr>
<td></td>
<td>1 to 1048576</td>
</tr>
</tbody>
</table>

**Function**

Inverts the Block Window mask pattern data when Test Sequence is Mixed. Specify the inversion range by the number of bits from the <start> address (relative specification).

**Example**

To invert 1 bit of the Block Window mask pattern data from address 0 when Test Pattern is Sequence:

```plaintext
> :SENSe:PAReM:MASK:SEQuence:DREVerse:DELTa 1,#H0,1
```

**Compatibility**

Incompatible with existing models.
<block>,<range>,<page>,<data>

Parameter

- **<block>** = <DECIMAL NUMERIC PROGRAM DATA>
  1 to 128 blocks, 1-block steps

- **<range>** = <CHARACTER PROGRAM DATA>
  - **PAGE** Specifies a page.
  - **ALL** Specifies all data.

- **<page>** = <DECIMAL NUMERIC PROGRAM DATA>
  1 to 8192 (Max.) pages, in 1-page steps
  Max= \[
  \text{Pattern Length} \over 128
  \] (rounding up fractions)

- Specify "0" when **<range>** is set to ALL.

- **<data>** = <DECIMAL NUMERIC PROGRAM DATA>
  0 Fills the specified range with 0s.
  1 Fills the specified range with 1s.

Function
Sets 0s or 1s for the specified page or all data in Block Window mask pattern data when **Test Pattern** is **Sequence**.

Example
To set 0s for the first page of the Block Window mask pattern data when **Test Pattern** is **Sequence**.

```
> :SENSe:PATTern:MASK:SEQuence:DATA:FILL 1,PAGE,1,0
```

Compatibility
Incompatible with existing models.
7.4.4 Commands related to Input tab

Figure 7.4.4-1 Input tab
7.4.4.1 Data Input setting commands

![Figure 7.4.4.1-1 Single-End Data setting](image1)

![Figure 7.4.4.1-2 Differential 50 Ohm setting](image2)
### Table 7.4.4.1-1  Data Input setting commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
</table>
                        :INPut:DATA:INTerface? |
| [2] | No label  
(Set the data input port.) | :INPut:DATA:SINGle  
                        :INPut:DATA:SINGle? |
| [3] | No label  
(Set the differential type.) | :INPut:DATA:DIFFerential  
                        :INPut:DATA:DIFFerential? |
| [4] | Data Threshold  
XData Threshold | :INPut:DATA:THReshold  
                        :INPut:DATA:THReshold? |
| [5] | No label  
(Set the differential reference signal.) | :INPut:DATA:DIFFerential:BASis  
                        :INPut:DATA:DIFFerential:BASis? |
| [6] | No label  
(Set the threshold for differential data input.) | :INPut:DATA:DIFFerential:THReshold  
                        :INPut:DATA:DIFFerential:THReshold? |
                        :INPut:DATA:TERMination? |
| [8] | No label  
(Termination voltage setting) | :INPut:DATA:TLEVel  
                        :INPut:DATA:TLEVel? |
Chapter 7  SCPI Commands

:INPut:DATA:INTerface <interface>

Parameter  

<table>
<thead>
<tr>
<th>&lt;interface&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>SINGle</td>
</tr>
<tr>
<td>DIF50ohm</td>
</tr>
<tr>
<td>DIF100ohm</td>
</tr>
</tbody>
</table>

Function  
Sets the data input interface.

Example  
To set the data input interface to Single Ended:

> :INPut:DATA:INTerface SINGle

Compatibility  
Incompatible with existing models.

:INPut:DATA:INTerface?

Response  

<table>
<thead>
<tr>
<th>&lt;interface&gt; = &lt;RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>SING, DIF50, DIF100</td>
</tr>
</tbody>
</table>

Function  
Queries the data input interface.

Example  

> :INPut:DATA:INTerface?

< SING

Compatibility  
Incompatible with existing models.

:INPut:DATA:SINGle <item>

Parameter  

<table>
<thead>
<tr>
<th>&lt;item&gt; = &lt;PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATA</td>
</tr>
<tr>
<td>XDATa</td>
</tr>
</tbody>
</table>

Function  
Sets the input port when the data input interface is set to Single.

Example  
To set the input port to Data:

> :INPut:DATA:SINGle DATA

Compatibility  
Incompatible with existing models.

:INPut:DATA:SINGle?

Response  

<table>
<thead>
<tr>
<th>&lt;item&gt; = &lt;RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATA, XDAT</td>
</tr>
</tbody>
</table>

Function  
Queries the input port when the data input interface is set to Single.

Example  

> :INPut:DATA:SINGle?

< DATA

Compatibility  
Incompatible with existing models.
### :INPut:DATA:DIFFerential <item>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;item&gt;</td>
<td>&lt;CHARACTER PROGRAM DATA&gt;</td>
</tr>
<tr>
<td>INDependent</td>
<td>Data/Xdata independent setting</td>
</tr>
<tr>
<td>TRACking</td>
<td>Data/Xdata common setting</td>
</tr>
<tr>
<td>ALTernate</td>
<td>Data/Xdata differential setting</td>
</tr>
</tbody>
</table>

**Function**
Sets the differential type when the data input interface is differential.

**Example**
To set the differential type to Data/Xdata independent setting:
```
> :INPut:DATA:DIFFerential INDependent
```

**Compatibility**
Incompatible with existing models.

### :INPut:DATA:DIFFerential?

<table>
<thead>
<tr>
<th>Response</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;item&gt;</td>
<td>&lt;CHARACTER RESPONSE DATA&gt;</td>
</tr>
<tr>
<td>IND, TRAC, ALT</td>
<td></td>
</tr>
</tbody>
</table>

**Function**
Queries the differential type when the data input interface is set to Differential.

**Example**
```
> :INPut:DATA:DIFFerential?
< IND
```

**Compatibility**
Incompatible with existing models.

### :INPut:DATA:THReshold <port>, <numeric>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;port&gt;</td>
<td>&lt;CHARACTER PROGRAM DATA&gt;</td>
</tr>
<tr>
<td>DATA</td>
<td>Selects Data.</td>
</tr>
<tr>
<td>XDATa</td>
<td>Selects Xdata.</td>
</tr>
<tr>
<td>&lt;numeric&gt;</td>
<td>&lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
</tr>
<tr>
<td>-3.500 to 3.300</td>
<td>-3.500 to 3.300 V, in 0.001 V steps (When MU181040A-002 and MU181040B-002 are installed)</td>
</tr>
<tr>
<td>-0.350 to 0.350</td>
<td>-0.350 to 0.350 V, in 0.001 V steps (when MU181040A-001 is installed)</td>
</tr>
</tbody>
</table>

**Function**
Sets the data input threshold value for the specified input port.

**Example**
To set the data input threshold value for the Data port to -3.000 V:
```
> :INPut:DATA:THReshold DATA, -3.000
```

**Compatibility**
Partially compatible with the MP1632C and MP1776A (parameters are incompatible). (The <port> setting is not provided.)
### :INPut:DATA:THReshold? <port>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;port&gt; = &lt;CHARACTER PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATA</td>
<td>Selects Data.</td>
</tr>
<tr>
<td>XDATa</td>
<td>Selects Xdata.</td>
</tr>
</tbody>
</table>

**Response**

<table>
<thead>
<tr>
<th>&lt;numeric&gt; = &lt;NR2 NUMERIC RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>-3.500 to 3.300 V (When MU181040A-002 and MU181040B-002 are installed)</td>
</tr>
<tr>
<td>-0.350 to 0.350 V (when MU181040A-001 is installed)</td>
</tr>
</tbody>
</table>

**Function**
Queries the data input threshold value for the specified input port.

**Example**
To query the data input threshold value for the Data port:

```
> :INPut:DATA:THReshold? DATA
< -3.000
```

**Compatibility**
Partially compatible with the MP1632C and MP1776A (parameters are incompatible). (The <port> setting is not provided.)

### :INPut:DATA:DIFFerential:BASis <basis>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;basis&gt; = &lt;CHARACTER PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATA</td>
<td>Sets the differential reference signal to Data.</td>
</tr>
<tr>
<td>XDATa</td>
<td>Sets the differential reference signal to XData.</td>
</tr>
</tbody>
</table>

**Function**
Sets the differential reference signal for the data input threshold value.

**Example**
To set the differential reference signal for the data input threshold value to DATA:

```
> :INPut:DATA:DIFFerential:BASis DATA
```

**Compatibility**
Incompatible with existing models.
7.4 ED Commands

:INPut:DATA:DIFFerential:BASis?

Response  

<\text{basis}> = \langle\text{CHARACTER RESPONSE DATA}\rangle

DATA, XDAT

Function  

Queries the differential reference signal for the data input threshold value.

Example  

> :INPut:DATA:DIFFerential:BASis?

< DATA

Compatibility  

Incompatible with existing models.

:INPut:DATA:DIFFerential:THReshold <\text{numeric}>

Parameter  

<\text{numeric}> = \langle\text{DECIMAL NUMERIC PROGRAM DATA}\rangle

\(-3.000\) to \(3.000\) \(-3.000\) to \(3.000\) V, in \(0.001\) V steps (when MU181040A-002 and MU181040B-002 are installed)

\(-0.700\) to \(0.700\) \(-0.700\) to \(0.700\) V, in \(0.001\) V steps (when MU181040A-001 is installed)

Function  

Sets the data input threshold value for differential input.

Example  

To set the data input threshold value to \(-3.000\) V:

> :INPut:DATA:DIFFerential:THReshold -3.000

Compatibility  

Incompatible with existing models.

:INPut:DATA:DIFFerential:THReshold?

Response  

<\text{numeric}> = \langle\text{NR2 NUMERIC RESPONSE DATA}\rangle

Function  

Queries the data input threshold value for differential input.

Example  

> :INPut:DATA:DIFFerential:THReshold?

< -3.000

Compatibility  

Incompatible with existing models.

:INPut:DATA:TERMination <\text{term}>

Parameter  

<\text{term}> = \langle\text{CHARACTER PROGRAM DATA}\rangle

GND \(50\ \Omega\) to GND

NECL \(50\ \Omega\) to \(-2\) V

LVPecl \(50\ \Omega\) to \(1.3\) V

PCML \(50\ \Omega\) to \(3.3\) V

VARiable \(50\ \Omega\) to Variable Voltage

Function  

Sets the data input termination condition.

Example  

To set the data input termination condition to GND:

> :INPut:DATA:TERMination GND

Compatibility  

Incompatible with existing models.
Chapter 7  SCPI Commands

:INPut:DATA:TERMination?

Response  
<term> = <CHARACTER RESPONSE DATA>

Function  Queries the data input termination condition.

Example  
> :INPut:DATA:TERMination?
< GND

Compatibility  Incompatible with existing models.

:INPut:DATA:TLEVel <numeric>

Parameter  
<numeric> = <DICIMAL NUMERIC PROGRAM DATA>
-2.50 to 3.50 -2.50 to 3.50 V, in 0.01 V steps

Function  Sets the termination voltage when the input termination condition is set to Variable.
The setting is invalid when the data input interface is Differential 100 Ω.

Example  
To set the termination voltage when the input termination condition is set to Variable to -2.00 V:
> :INPut:DATA:TLEVel -2.00

Compatibility  Incompatible with existing models.

:INPut:DATA:TLEVel?

Response  
<numeric> = <NR2 NUMERIC RESPONSE DATA>
-2.50 to 3.50 -2.50 to 3.50 V

Function  Queries the termination voltage when the input termination condition is set to Variable.

Example  
> :INPut:DATA:TLEVel?
< -2.00

Compatibility  Incompatible with existing models.
7.4 ED Commands

7.4.4.2 Clock Input setting commands

Figure 7.4.4.2-1 Clock Input setting (For External Clock)

Figure 7.4.4.2-2 Clock Input setting (For Recovered Clock)

Figure 7.4.4.2-3 Clock Termination Setting dialog box

Figure 7.4.4.2-4 Clock Input setting (For no delay option)
Table 7.4.2-1  Clock Input setting commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
</table>

:INPut:CLOCK:SELection <sel>

Parameter  
<sel> = <CHARACTER PROGRAM DATA>
RECovered  Recovered Clock
EXTernal   External Clock

Function  
Sets the clock input type.

Example  
To set the clock input type to the Recovered Clock:
> :INPut:CLOCK:SELection RECovered

Compatibility  
Incompatible with existing models.

:INPut:CLOCK:SELection?

Response  
<sel> = <CHARACTER RESPONSE DATA>
REC  Recovered Clock
EXT  External Clock

Function  
Queries the clock input type.

Example  
> :INPut:CLOCK:SELection?
< REC

Compatibility  
Incompatible with existing models.
7.4 ED Commands

:INPut:CLOCk:RECovery <freq>

Parameter

Parameter | Description |
-----------|-------------|
"OC_3" | OC3/STM1 0.155520 Gbit/s |
"OC_12" | OC12/STM4 0.622080 Gbit/s |
"FC" | 1GFC 1.062500 Gbit/s |
"GbE" | GbE 1.250000 Gbit/s |
"SATA1_5" | SATA1.5Gb/s 1.500000 Gbit/s |
"2G_FC" | 2GFC 2.125000 Gbit/s |
"OC_48" | OC48/STM16 2.488320 Gbit/s |
"PCI_EX1" | PCI Express I 2.500000 Gbit/s |
"OC_48_G709" | OTU1 2.666060 Gbit/s |
"SATA3" | SATA 3Gb/s 3.000000 Gbit/s |
"XAUI" | XAUI 3.125000 Gbit/s |
"4G_FC" | 4GFC 4.250000 Gbit/s |
"PCI_EX2" | PCI Express II 5.000000 Gbit/s |
"SATA6" | SATA 6Gb/s 6.000000 Gbit/s |
"OC_192" | OC192/STM64 9.953280 Gbit/s |
"10GbE" | 10GbE 10.312500 Gbit/s |
"10G_FC" | 10GFC 10.518750 Gbit/s |
"OC_192_G975" | G975 FEC 10.664228 Gbit/s |
"OC_192_G709" | OTU2 10.709225 Gbit/s |
"10GbE_G709" | 10GbE over FEC 11.095700 Gbit/s |
"10G_FC_G709" | 10GFC over FEC 11.316800 Gbit/s |
"Variable" | Variable :Variable |

Function
Sets the rated frequency when the clock input type is the recovery clock.
Example
To set the rated frequency to "OC3":
> :INPut:CLOCk:RECovery "OC_3"
Compatibility
Incompatible with existing models.

:INPut:CLOCk:RECovery?

Response

Response | Description |
----------|-------------|

Function
Queries the rated frequency when the clock input type is the recovery clock.
Example
> :INPut:CLOCk:RECovery?
< "OC_3"
Compatibility
Incompatible with existing models.
### :INPut:CLOCk:CRFReq <numeric>

**Parameter**  
\(<\text{numeric}> = \langle\text{DECIMAL NUMERIC PROGRAM DATA}\rangle\)  
0.100000, 0.125000, 0.140600, 0.155520, 0.156300, 0.171900,  
0.187500, 0.200000, 0.250000, 0.281300, 0.312500, 0.343800,  
0.375000, 0.400000, 0.500000, 0.562500, 0.622080, 0.625000,  
0.687500, 0.750000, 0.800000, 1.000000, 1.062500, 1.125000,  
1.250000, 1.375000, 1.500000, 1.600000, 2.000000, 2.125000,  
2.488320, 2.250000, 2.500000, 2.666060, 2.750000, 3.000000,  
3.125000, 3.200000, 4.250000  
Unit is Gbit/s

<table>
<thead>
<tr>
<th>4.900000 to 6.250000</th>
<th>4.900000 to 6.250000 Gbit/s /1 kbit/s Step</th>
</tr>
</thead>
<tbody>
<tr>
<td>9.800000 to 12.500000</td>
<td>9.800000 to 12.500000 Gbit/s /1 kbit/s Step</td>
</tr>
</tbody>
</table>

Note, however:  
When MU181040A-001 is installed:  9.8 to 12.5 Gbit/s  
When MU181040A/B-002 or MU181040A/B-x20 is installed: 0.1 to 12.5 Gbit/s

**Function**  
Sets the rated frequency for the clock recovery.

**Example**  
To set the rated frequency for the clock recovery to 0.100000 Gbit/s (when MU181040A-x20, MU181040B-x20 is installed):

```plaintext
> :INPut:CLOCk:CRFReq 0.100000
```

To set the rated frequency for the clock recovery to 9.953280 Gbit/s (when MU181040A-001 is installed):

```plaintext
> :INPut:CLOCk:CRFReq 9.953280
```

**Compatibility**  
Incompatible with existing models.

### :INPut:CLOCk:CRFReq?

**Response**  
\(<\text{numeric}> = \langle\text{NR2 NUMERIC RESPONSE DATA}\rangle\)  
0.100000, 0.125000, 0.140600, 0.155520, 0.156300, 0.171900,  
0.187500, 0.200000, 0.250000, 0.281300, 0.312500, 0.343800,  
0.375000, 0.400000, 0.500000, 0.562500, 0.622080, 0.625000,  
0.687500, 0.750000, 0.800000, 1.000000, 1.062500, 1.125000,  
1.250000, 1.375000, 1.500000, 1.600000, 2.000000, 2.125000,  
2.488320, 2.250000, 2.500000, 2.666060, 2.750000, 3.000000,  
3.125000, 3.200000, 4.250000  
Unit is Gbit/s.

<table>
<thead>
<tr>
<th>4.900000 to 6.250000</th>
<th>4.900000 to 6.250000 Gbit/s</th>
</tr>
</thead>
<tbody>
<tr>
<td>9.800000 to 12.500000</td>
<td>9.800000 to 12.500000 Gbit/s</td>
</tr>
</tbody>
</table>

**Function**  
Queries the rated frequency for the clock recovery.

**Example**  
```plaintext
> :INPut:CLOCk:CRFReq?
< 0.100000
```

**Compatibility**  
Incompatible with existing models.
:INPut:CLOCk:TERMination <term>

Parameter  
<term> = <CHARACTER PROGRAM DATA>
- GND  50 Ω to GND
- NECL  50 Ω to −2 V
- LVPecl  50 Ω to 1.3 V
- PCML  50 Ω to 3.3 V
- VARiable  50 Ω to Variable Voltage

Function  
Sets the clock input termination condition.

Example  
To set the clock input termination condition to GND:
> :INPut:CLOCk:TERMination GND

Compatibility  
Incompatible with existing models.

:INPut:CLOCk:TERMination?  

Response  
<term> = <CHARACTER RESPONSE DATA>
- GND, NECL, LVP, PCML, VAR

Function  
Queries the clock input termination condition.

Example  
> :INPut:CLOCk:TERMination?
< GND

Compatibility  
Incompatible with existing models.

:INPut:CLOCk:TLEVel <numeric>

Parameter  
<numeric> = <NR2 NUMERIC PROGRAM DATA>
- −2.50 to 3.50  −2.50 to 3.50 V, in 0.01 V steps

Function  
Sets the termination voltage when the input termination condition is set to Variable.

Example  
To set the termination voltage when the input termination condition is set to Variable to −2 V:
> :INPut:CLOCk:TLEVel -2.00

Compatibility  
Incompatible with existing models.

:INPut:CLOCk:TLEVel?

Response  
<numeric> = <NR2 NUMERIC RESPONSE DATA>
- −2.50 to 3.50  −2.50 to 3.50 V

Function  
Queries the termination voltage when the input termination condition is set to Variable.

Example  
> :INPut:CLOCk:TLEVel?
< -2.00

Compatibility  
Incompatible with existing models.
Chapter 7  SCPI Commands

**:INPut:CLOCk:RECLOck <logic>**

**Parameter**

- `<logic>` = `<CHARACTER PROGRAM DATA>`
  - POSitive: Positive logic
  - NEGative: Negative logic

**Function**
Selects the phase unit for the clock recovery clock.

**Example**
To set the phase unit for the clock recovery clock to POSitive:

```
> :INPut:CLOCk:RECLOCK POSitive
```

**Compatibility**
Incompatible with existing models.

**:INPut:CLOCk:RECLOck?**

**Response**

- `<logic>` = `<CHARACTER RESPONSE DATA>`
  - POS, NEG

**Function**
Queries the phase unit for the clock recovery clock.

**Example**

```
> :INPut:CLOCk:RECLOCK?
< POS
```

**Compatibility**
Incompatible with existing models.

**:INPut:CLOCk:DELay <numeric>[,<unit>]**

**Parameter**

- `<numeric>` = `<DECIMAL NUMERIC PROGRAM DATA>`
  - In mUI units: -1000 to 1000, -1000 to 1000 mUI, in 1 mUI steps
  - In ps units: Converted from the calculated frequency value, based on the setting resolution of mUI.

- `<unit>` = `<CHARACTER PROGRAM DATA>`
  - UI: mUI unit
  - PS: ps unit

  (The ps unit is selected when `<unit>` is omitted.)

**Function**
Sets the value and unit of the clock input phase variable.
The specified value may not be exactly set as is when the ps unit is set, due to the limitation of the setting resolution.
The value is adjusted to the most approximate value and set in this event.

**Example**
To set the clock input phase variable to -1000 mUI:

```
> :INPut:CLOCk:DELay -1000,UI
```

**Compatibility**
Compatible with the MP1632C and MP1776A.
7.4 ED Commands

:INPut:CLOCk:DELay? [<unit>]

Parameter

<table>
<thead>
<tr>
<th>&lt;unit&gt; = &lt;CHARACTER PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>UI mUI unit</td>
</tr>
<tr>
<td>PS ps unit</td>
</tr>
<tr>
<td>(The ps unit is selected when &lt;unit&gt; is omitted.)</td>
</tr>
</tbody>
</table>

Response

<table>
<thead>
<tr>
<th>&lt;numeric&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>In mUI units: –1000 to 1000 –1000 to 1000 mUI</td>
</tr>
<tr>
<td>In ps units: Converted from the calculated frequency value, based on the setting resolution of mUI.</td>
</tr>
</tbody>
</table>

Function
Queries the value and unit of the clock input phase variable.

Example
To query the value of the clock input phase variable in UI unit:

> :INPut:CLOCk:DELay? UI
< -1000

Compatibility
Compatible with the MP1632C and MP1776A.

:INPut:CLOCk:RELative <boolean>

Parameter

<table>
<thead>
<tr>
<th>&lt;boolean&gt; = &lt;BOOLEAN PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 or OFF Reference OFF</td>
</tr>
<tr>
<td>1 or ON Reference ON</td>
</tr>
</tbody>
</table>

Function
Sets the clock input phase variable reference ON/OFF.

Example
To set the clock input phase variable reference ON:

> :INPut:CLOCk:RELative ON

Compatibility
Incompatible with existing models.

:INPut:CLOCk:RELative?

Response

<table>
<thead>
<tr>
<th>&lt;boolean&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 Reference OFF</td>
</tr>
<tr>
<td>1 Reference ON</td>
</tr>
</tbody>
</table>

Function
Queries the clock input phase variable reference setting.

Example

> :INPut:CLOCk:RELative?
< 1

Compatibility
Incompatible with existing models.
:INPut:CLOCK:RDELay <numeric>[,<unit>]

Parameter

Parameter

<nemonic> = <DECIMAL NUMERIC PROGRAM DATA>

In mUI units: 
-1000 to 1000 
-1000 to 1000 mUI, in 1 mUI steps

In ps units: 
Converted from the calculated frequency value, 
based on the setting resolution of mUI.

[unit] = <CHARACTER PROGRAM DATA>

UI 
mUI unit

PS 
ps unit

(The ps unit is selected when <unit> is omitted.)

Function

Sets the value of the clock input phase variable reference.
Set a value using a difference from the reference value.
The specified value may not be exactly set as is when the ps unit is set, 
due to the limitation of the setting resolution. The value is adjusted to 
the most approximate value and set in this event.

Example

To set the clock phase variable reference to -1000 mUI:
> :INPut:CLOCK:RDELay -1000,UI

Compatibility

Incompatible with existing models.

:INPut:CLOCK:RDELay? [<unit>]

Parameter

[unit] = <CHARACTER PROGRAM DATA>

UI 
mUI units

PS 
ps units

(The ps unit is selected when <unit> is omitted.)

Response

<nemonic> = <NR1 NUMERIC RESPONSE DATA>

In mUI units: 
-1000 to 1000 
-1000 to 1000 mUI

In ps units: 
Converted from the calculated frequency value, 
based on the setting resolution of mUI.

Function

Queries the value of the clock input phase variable reference.

Example

To query the value of the clock input phase variable reference in mUI units.
:INPut:CLOCK:RDELay? UI
< -1000

Compatibility

Incompatible with existing models.

:INPut:CLOCK:CALibration

Function

Calibrates the input clock phase setting value.

Example

> :INPut:CLOCK:CALibration

Compatibility

Incompatible with existing models.
.:INPut:CLOCk:JINPut <boolean>

Parameter

<boolean> = <BOOLEAN PROGRAM DATA>
0 or OFF Jitter input OFF setting
1 or ON Jitter input ON setting

Function

When a jitter is added to the input signal, the Feedback process for Delay must be set to OFF.
When a jitter is added, set the Jitter Input button to ON, using this command. When a jitter is not added, set the Jitter Input button to OFF.

Example

To set the Jitter Input button to ON when a jitter is added:

> :INPut:CLOCk:JINPut ON

Compatibility

Incompatible with existing models.

.:INPut:CLOCk:JINPut?

Response

<numeric> = <NR1 NUMERIC RESPONSE DATA>
0 Jitter input OFF setting
1 Jitter input ON setting

Function

Queries the Jitter Input button setting.

Example

> :INPut:CLOCk:JINPut?
< 1

Compatibility

Incompatible with existing models.
Chapter 7  SCPI Commands

7.4.5 Commands related to Capture tab

Figure 7.4.5-1  Capture tab (initial state)

Figure 7.4.5-2  Condition Setting

Figure 7.4.5-3  Capture Acquisition dialog box
Table 7.4.5-1 Capture tab setting commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1]</td>
<td>Capture Start</td>
<td>:SENSe:CAPTure:STARt</td>
</tr>
<tr>
<td>[4]</td>
<td>Number of Block</td>
<td>:SENSe:CAPTure:CONDition:BLOCk</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:CAPTure:CONDition:BLOCk?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:CAPTure:CONDition:TRIGger?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:CAPTure:CONDition:POSition?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:CAPTure:CONDition:MPLength?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:CAPTure:CONDition:MPEdit?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:CAPTure:CONDition:MASKedit?</td>
</tr>
<tr>
<td>[12]</td>
<td>Query for capture data acquisition state</td>
<td>:SENSe:CAPTure:ACQuisition:STATEt?</td>
</tr>
<tr>
<td>[14]</td>
<td>Query for capture data length per block</td>
<td>:SENSe:CAPTure:BPATtern:LENGth?</td>
</tr>
<tr>
<td>[16]</td>
<td>Query for capture result data</td>
<td>:SENSe:CAPTure:BPATtern:BDATa:WHOLe?</td>
</tr>
<tr>
<td>[17]</td>
<td>Query for capture result error position</td>
<td>:SENSe:CAPTure:BPATtern:ERRor:WHOLe?</td>
</tr>
<tr>
<td>[18]</td>
<td>Query for capture result error position</td>
<td>:SENSe:CAPTure:BPATtern:BERRor:WHOLe?</td>
</tr>
</tbody>
</table>

**:SENSe:CAPTure:STARt**

Function: Starts capture result acquisition.
Example: > :SENSe:CAPTure:STARt
Compatibility: Incompatible with existing models.

**:SENSe:CAPTure:STOP**

Function: Stops capture result acquisition.
Example: > :SENSe:CAPTure:STOP
Compatibility: Incompatible with existing models.

**:SENSe:CAPTure:TRIGger**

Function: Generates a capture manual trigger.
Example: > :SENSe:CAPTure:TRIGger
Compatibility: Incompatible with existing models.
### :SENSe:CAPTure:CONDition:BLOCk <block>

**Parameter**

- `<block>` = `<CHARACTER PROGRAM DATA>`
- B1 1 block
- B2 2 blocks
- B4 4 blocks
- B8 8 blocks
- B16 16 blocks
- B32 32 blocks
- B64 64 blocks
- B128 128 blocks

**Function**
Sets the capture block division number.

**Example**
To set the capture block division number to 8 blocks:
> :SENSe:CAPTure:CONDition:BLOCk B8

**Compatibility**
Incompatible with existing models.

### :SENSe:CAPTure:CONDition:BLOCk?

**Response**

- `<block>` = `<CHARACTER RESPONSE DATA>`
- B1 1 block
- B2 2 blocks
- B4 4 blocks
- B8 8 blocks
- B16 16 blocks
- B32 32 blocks
- B64 64 blocks
- B128 128 blocks

**Function**
Queries the capture block division number.

**Example**
> :SENSe:CAPTure:CONDition:BLOCk?
  < B8

**Compatibility**
Incompatible with existing models.

### :SENSe:CAPTure:CONDition:TRIGger <trigger>

**Parameter**

- `<trigger>` = `<CHARACTER PROGRAM DATA>`
- EDETect Error Detect: When an error is detected
- PATTern When matches the match pattern
- MANual When a manual trigger is generated
- EXTernal When external trigger is selected

**Function**
Sets the trigger to store the data to memory for the capture function.

**Example**
To set the manual trigger:
> :SENSe:CAPTure:CONDition:TRIGger MANual

**Compatibility**
Incompatible with existing models.
### :SENSe:CAPTure:CONDition:TRIGger?

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;trigger&gt; = &lt;CHARACTER RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function</td>
<td>Queries the trigger to store the data to memory for the capture function.</td>
</tr>
<tr>
<td>Example</td>
<td>&gt; :SENSe:CAPTure:CONDition:TRIGger?</td>
</tr>
<tr>
<td></td>
<td>&lt; MAN</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Incompatible with existing models.</td>
</tr>
</tbody>
</table>

### :SENSe:CAPTure:CONDition:POSition <pos>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;pos&gt; = &lt;CHARACTER PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>TOP Store the data to the top of the memory.</td>
</tr>
<tr>
<td></td>
<td>MIDDle Store the data in the middle of the memory.</td>
</tr>
<tr>
<td></td>
<td>BOTTom Store the data to the last of the memory.</td>
</tr>
<tr>
<td>Function</td>
<td>Sets the data storage memory position for the capture function.</td>
</tr>
<tr>
<td>Example</td>
<td>To set the data storage memory position to TOP:</td>
</tr>
<tr>
<td></td>
<td>&gt; :SENSe:CAPTure:CONDition:POSition TOP</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Incompatible with existing models.</td>
</tr>
</tbody>
</table>

### :SENSe:CAPTure:CONDition:POSition?

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;pos&gt; = &lt;CHARACTER RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>TOP, MIDD, BOTT</td>
</tr>
<tr>
<td>Function</td>
<td>Queries the data storage memory position for the capture function.</td>
</tr>
<tr>
<td>Example</td>
<td>&gt; :SENSe:CAPTure:CONDition:POSition?</td>
</tr>
<tr>
<td></td>
<td>&lt; TOP</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Incompatible with existing models.</td>
</tr>
</tbody>
</table>

### :SENSe:CAPTure:CONDition:MPLength <length>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;length&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>4 to 64 4 to 64 bits, in 4-bit steps</td>
</tr>
<tr>
<td>Function</td>
<td>Sets the capture trigger match pattern length.</td>
</tr>
<tr>
<td>Example</td>
<td>To set the capture trigger match pattern length to 12 bits:</td>
</tr>
<tr>
<td></td>
<td>&gt; :SENSe:CAPTure:CONDition:MPLength 12</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Incompatible with existing models.</td>
</tr>
</tbody>
</table>
Chapter 7  SCPI Commands

:SENSe:CAPTure:CONDition:MPLength?
Response  \(<\text{length}> = \langle\text{NR1 NUMERIC RESPONSE DATA}\rangle\)
4 to 64  4 to 64 bits
Function  Queries the capture trigger match pattern length.
Example  > :SENSe:CAPTure:CONDition:MPLength?
< 12
Compatibility  Incompatible with existing models.

:SENSe:CAPTure:CONDition:MPEDit \(<\text{data}>\)
Parameter  \(<\text{data}> = \langle\text{STRING PROGRAM DATA} \rangle\)
"**...**" Edits the pattern data in hexadecimal format.
Specify a 1- to 16-character string, consisting of 0 to 9 and A to F.
Function  Edits the capture trigger match pattern.
Example  To set the capture trigger match pattern to 1010:
> :SENSe:CAPTure:CONDition:MPEDit "1010"
Compatibility  Incompatible with existing models.

:SENSe:CAPTure:CONDition:MPEDit?
Response  \(<\text{data}> = \langle\text{STRING RESPONSE DATA} \rangle\)
"**...**" Returns the pattern data in hexadecimal format.
Specify a 1- to 16-character string, consisting of 0 to 9 and A to F.
Function  Queries the capture trigger match pattern.
Example  > :SENSe:CAPTure:CONDition:MPEDit?
< "1010"
Compatibility  Incompatible with existing models.

:SENSe:CAPTure:CONDition:MASKedit \(<\text{data}>\)
Parameter  \(<\text{data}> = \langle\text{STRING PROGRAM DATA} \rangle\)
"**...**" Edits the pattern data in hexadecimal format.
Specify a 1- to 16-character string, consisting of 0 to 9 and A to F.
Function  Edits the capture trigger mask pattern.
Example  To set the capture trigger mask pattern to FFFF:
> :SENSe:CAPTure:CONDition:MASKedit "FFFF"
Compatibility  Incompatible with existing models.
7.4 ED Commands

:SENSe:CAPTure:CONDition:MASKedit?

Response

```
<data> = <STRING RESPONSE DATA>

"**…**" Returns the pattern data in hexadecimal format.
Specify a 1- to 16-character string, consisting of
0 to 9 and A to F.
```

Function
Queries the capture trigger mask pattern.

Example
```
> :SENSe:CAPTure:CONDition:MASKedit?
< "FFFD"
```

Compatibility
Incompatible with existing models.

:SENSe:CAPTure:ACQuisition:BNUMber?

Response

```
<block> = <NR1 NUMERIC RESPONSE DATA>
0 to 128 0 to 128 blocks, in 1-block steps, number of valid
capture data blocks
```

Function
Queries the number of valid capture data blocks.

Example
```
> :SENSe:CAPTure:ACQuisition:BNUMber?
< 128
```

Compatibility
Incompatible with existing models.

:SENSe:CAPTure:ACQuisition:STARt <range>[,<start>,<number>]

Parameter

```
<range> = <CHARACTER PROGRAM DATA>
ALL Acquires all capture data.
SELect Acquires capture data of the specified blocks.
[start>] = <DECIMAL NUMERIC PROGRAM DATA>
1 to 128 Block No. 1 to 128 (Resolution: 1) , acquisition
start block
[number>] = <DECIMAL NUMERIC PROGRAM DATA>
1 to 128 Block No. 1 to 128 (Resolution: 1) , number of
blocks to be acquired
```

Note:
The maximum setting value of <start> and <number> is the
maximum value of valid data. <start> and <number> are omitted
when <range> = All.

Function
Acquires capture data.

Example
```
To acquire all capture data:
> :SENSe:CAPTure:ACQuisition:STARt ALL

To acquire capture data of Blocks 1 to 64:
> :SENSe:CAPTure:ACQuisition:STARt SELect,1,64
```

Compatibility
Incompatible with existing models.
Chapter 7  SCPI Commands

:SENSe:CAPTure:ACQuisition:STATe?

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;state&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Capture data acquisition is not performed.</td>
</tr>
<tr>
<td>1</td>
<td>Capture data acquisition is being performed.</td>
</tr>
</tbody>
</table>

Function
Queries capture data acquisition status.

Example
> :SENSe:CAPTure:ACQuisition:STATe?
< 1

Compatibility
Incompatible with existing models.

:SENSe:CAPTure:ACQuisition:CANCel

Function
Cancels capture data acquisition.

Example
> :SENSe:CAPTure:ACQuisition:CANCel

Compatibility
Incompatible with existing models.

:SENSe:CAPTure:BPATtern:LENGth?

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;length&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Maximum capture data / 1 to Maximum capture data / n [bits]</td>
</tr>
<tr>
<td></td>
<td>(n = 1, 2, 4, 8, 16, 32, 64, 128)</td>
</tr>
<tr>
<td></td>
<td>Capture data length per block</td>
</tr>
</tbody>
</table>

Note:
At 2 ch Combination, the length of capture data is doubled.
At 4 ch Combination, the length of capture data is quadrupled.

Function
Queries the capture data length per block.

Example
> :SENSe:CAPTure:BPATtern:LENGth?
< 4194304

Compatibility
Incompatible with existing models.
### :SENSe:CAPTure:BPATtern:DATA:WHOLe? <block>,<start>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;block&gt;</td>
<td>= &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
</tr>
<tr>
<td></td>
<td>1 to 128 1 to 128 blocks, 1-block steps</td>
</tr>
<tr>
<td>&lt;start&gt;</td>
<td>= &lt;NON-DECIMAL PROGRAM DATA&gt;</td>
</tr>
<tr>
<td></td>
<td>#H0 to #H7FFFFFF 0 to 7FFFFFF bits, in 1-bit steps</td>
</tr>
<tr>
<td></td>
<td>(Specify in hexadecimal)</td>
</tr>
</tbody>
</table>

**Note:**
- The maximum setting is the pattern length.
- At 2 ch Combination, the setting range is doubled.
- At 4 ch Combination, the setting range is quadrupled.

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;data&gt; = &lt;STRING PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>&quot;H***...*&quot; Returns the pattern data in hexadecimal format.</td>
</tr>
<tr>
<td></td>
<td>Within 400 characters (pattern data of 400 × 4 bits)</td>
</tr>
</tbody>
</table>

| Function       | Queries the capture data in the specified block, in 400 × 4 bits starting |
|----------------| from the <start> address. Data whose error bit positions are inverted      |
|                | from those in the ED reference data can be obtained.                       |

| Example        | To query the capture data in block 3, in 400 × 4 bits from the bit 0       |
|----------------| address:                                                                    |
|                | > :SENSe:CAPTure:BPATtern:DATA:WHOLe? 3,#H0                                |
|                | < "H001"                                                                    |

| Compatibility  | Incompatible with existing models.                                          |
**Chapter 7  SCPI Commands**

:SENSe:CAPTure:BPATtern:BDATa:WHOLe? <block>,<start>[,<size>]

**Parameter**

- `<block>` = `<DECIMAL NUMERIC PROGRAM DATA>`
  1 to 128 1 to 128 blocks, 1-block steps
- `<start>` = `<NON-DECIMAL PROGRAM DATA>`
  #H0 to #H7FFFFFF 0 to 7FFFFFF bits, in 1-bit steps
  (Specify in hexadecimal)
- `[<size>]` = `<NR1 NUMERIC PROGRAM DATA>`
  1 to 16777216 1 to 16,777,216 bytes, in 1-byte steps

**Note:**

- The maximum setting is the pattern length.
- At 2 ch Combination, the setting range is doubled.
- At 4 ch Combination, the setting range is quadrupled.

**Response**

- `<bdata>` = `<DEFINITE LENGTH ARBITRARY BLOCK RESPONSE DATA>`
  #XYYY<binary>
  X: Number of YYY digits
  YYY: Number of bytes of `<binary>`
  1 to 16,777,216 bytes
  <binary>: Binary data of up to 16,777,216 bytes

**Function**

Queries the capture data in the specified block, in 16,777,216-byte binary data starting from the `<start>` address. Data whose error bit positions are inverted from those in the ED reference data can be obtained.

**Example**

To query the capture data in block 3, in 16,777,216-byte binary data from the bit 0 address:

```
> :SENSe:CAPTure:BPATtern:BDATA:WHOLe? 3,#H0
< #10011
```

**Compatibility**

Incompatible with existing models.
:SENSe:CAPTure:BPATtern:ERRor:WHOLE? <block>,<start>

Parameter

- `<block>` = `<DECIMAL NUMERIC PROGRAM DATA>`
  1 to 128  1 to 128 blocks, 1-block steps
- `<start>` = `<NON-DECIMAL PROGRAM DATA>`
  #H0 to #H7FFFFFF  0 to 7FFFFFF bits, in 1-bit steps
  (Specify in hexadecimal)

Note:
The maximum setting is the pattern length.
At 2 ch Combination, the setting range is doubled.
At 4 ch Combination, the setting range is quadrupled.

Response

- `<data>` = `<STRING PROGRAM DATA>`
  "H***…*"  Returns the pattern data in hexadecimal format.
  Within 400 characters (pattern data of 400 × 4 bits)

Function

Queries the capture error positions in the specified block, in 400 × 4 bits starting from the `<start>` address.

Example

To query the capture error positions in block 3, in 400 × 4 bits from the bit 0 address:

> :SENSe:CAPTure:BPATtern:ERRor:WHOLE? 3,#H0

< "#001"

Compatibility

Incompatible with existing models.
Chapter 7  SCPI Commands

:SENSe:CAPTure:BPATtern:BERRor:WHOLe? <block>,<start>[,<size>]

Parameter

- `<block>` = `<DECIMAL NUMERIC PROGRAM DATA>`
  1 to 128  
  1 to 128 blocks, 1-block steps
- `<start>` = `<NON-DECIMAL PROGRAM DATA>`
  #H0 to #H7FFFFFF  
  0 to 7FFFFFF bits, in 1-bit steps
  (Specify in hexadecimal)
- `[<size>]` = `<NR1 NUMERIC PROGRAM DATA>`
  1 to 16777216  
  1 to 16,777,216 bytes, in 1-byte steps

**Note:**

The maximum setting is the pattern length.
At 2 ch Combination, the setting range is doubled.
At 4 ch Combination, the setting range is quadrupled.

Response

- `<bdata>` = `<DEFINITE LENGTH ARBITRARY BLOCK RESPONSE DATA>`
  
  #XYYY<binary>  
  X: Number of YYY digits
  YYY: Number of bytes of `<binary>`
  
  1 to 16,777,216 bytes

  <binary>: Binary data of up to 16,777,216 bytes

Function

Queries the capture error positions in the specified block, in
16,777,216-byte binary data starting from the `<start>` address.

Example

To query the capture error positions in block 3, in 16,777,216-byte binary
data from the bit 0 address:

> :SENSe:CAPTure:BPATtern:BERRor:WHOLe? 3,#H0
< #10011

Compatibility

Incompatible with existing models.
7.4 ED Commands

7.4.6 Commands related to Misc tab

Figure 7.4.6-1  Misc tab
7.4.6.1 Pattern Sequence setting commands

Figure 7.4.6.1-1 Pattern Sequence setting field

Table 7.4.6.1-2 Pattern Sequence setting commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1]</td>
<td>Sequence</td>
<td>:SENSe::PATTern::IMoDe</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe::PATTern::IMoDe?</td>
</tr>
<tr>
<td>[2]</td>
<td>Source</td>
<td>:SENSe::PATTern::BURSt::MODE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe::PATTern::BURSt::MODE?</td>
</tr>
<tr>
<td>[3]</td>
<td>Delay</td>
<td>:SENSe::PATTern::BURSt::DELaY</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe::PATTern::BURSt::DELaY?</td>
</tr>
<tr>
<td>[4]</td>
<td>Delay</td>
<td>:SENSe::PATTern::BURSt::ADJust</td>
</tr>
<tr>
<td></td>
<td>Manual/Auto</td>
<td></td>
</tr>
<tr>
<td>[5]</td>
<td>Enable Period</td>
<td>:SENSe::PATTern::BURSt::ELENght</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe::PATTern::BURSt::ELENght?</td>
</tr>
<tr>
<td>[6]</td>
<td>Burst Cycle</td>
<td>:SENSe::PATTern::BURSt::CYCle</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe::PATTern::BURSt::CYCle?</td>
</tr>
</tbody>
</table>
### :SENSe:PATTern:IMODe <mode>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;mode&gt; = &lt;CHARACTER PROGRAM DATA&gt;</td>
<td></td>
</tr>
<tr>
<td>REPeat</td>
<td>Repeat signal</td>
</tr>
<tr>
<td>BURSt</td>
<td>Burst signal</td>
</tr>
</tbody>
</table>

**Function**
Sets the signal reception method from Repeat (consecutive) signal or Burst signal.

**Example**
To set the signal reception method to Repeat signal:

```plaintext
> :SENSe:PATTern:IMODe REPeat
```

**Compatibility**
Compatible with the MP1632C and MP1776A.

### :SENSe:PATTern:IMODe?

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;mode&gt; = &lt;CHARACTER RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>REP, BURS</td>
<td></td>
</tr>
</tbody>
</table>

**Function**
Queries the signal reception method.

**Example**

```plaintext
> :SENSe:PATTern:IMODe?
< REP
```

**Compatibility**
Compatible with the MP1632C and MP1776A.

### :SENSe:PATTern:BURSt:MODE <mode>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;mode&gt; = &lt;CHARACTER PROGRAM DATA&gt;</td>
<td></td>
</tr>
<tr>
<td>INTerner</td>
<td>Internal</td>
</tr>
<tr>
<td>EXTTrig</td>
<td>External Trigger</td>
</tr>
<tr>
<td>EXTeenable</td>
<td>External Enable</td>
</tr>
</tbody>
</table>

**Function**
Sets the Burst signal reception sequence.

**Example**
To set the Burst signal reception sequence to Internal:

```plaintext
> :SENSe:PATTern:BURSt:MODE INTerner
```

**Compatibility**
Incompatible with existing models.

### :SENSe:PATTern:BURSt:MODE?

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;mode&gt; = &lt;CHARACTER RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>INT, EXTT, EXT</td>
<td></td>
</tr>
</tbody>
</table>

**Function**
Queries the Burst signal reception sequence.

**Example**

```plaintext
> :SENSe:PATTern:BURSt:MODE?
< INT
```

**Compatibility**
Incompatible with existing models.
### :SENSe:PATTern:BURSt:DELay \(<\text{numeric}>\)

**Parameter**  
\(<\text{numeric}> = <\text{DECIMAL NUMERIC PROGRAM DATA}>\)  
0 to 2147483648  
0 to 2,147,483,648 bits, in 16-bit steps  
The maximum setting value is the Burst Cycle set value.  
Up to 2,147,483,584 bits can be set for \(<\text{numeric}>\) when the Burst Cycle is invalid.

**Function**  
Sets the delay value for the Burst signal.

**Example**  
To set the delay value for the Burst signal to 0 bits:  
> :SENSe:PATTern:BURSt:DELay 0

**Compatibility**  
Incompatible with existing models.

**Note:**  
At 2 ch Combination, the setting range and Step are doubled.  
At 4 ch Combination, the setting range and Step are quadrupled.

### :SENSe:PATTern:BURSt:DELay?  

**Response**  
\(<\text{numeric}> = <\text{NR1 NUMERIC RESPONSE DATA}>\)

**Function**  
Queries the delay value for the Burst signal.

**Example**  
> :SENSe:PATTern:BURSt:DELay?

< 0

**Compatibility**  
Incompatible with existing models.

### :SENSe:PATTern:BURSt:ADJust \(<\text{adjust}>\)

**Parameter**  
\(<\text{adjust}> = <\text{CHARACTER PROGRAM DATA}>\)  
AUTO  
MANual  
Executes automatic adjustment.  
Does not execute automatic adjustment

**Function**  
Sets whether to execute automatic adjustment of Burst Trigger Delay.

**Example**  
> :SENSe:PATTern:BURSt:ADJust AUTO

**Compatibility**  
Incompatible with existing models.

### :SENSe:PATTern:BURSt:ADJust?  

**Response**  
\(<\text{adjust}> = <\text{CHARACTER PROGRAM DATA}>\)

**Function**  
Queries the Burst Trigger Delay adjustment method.

**Example**  
> :SENSe:PATTern:BURSt:ADJust?

< AUTO

**Compatibility**  
Incompatible with existing models.
:SENSe:PATTern:BURSt:ELENgth <numeric>

Parameter
<numeric> = <DECIMAL NUMERIC PROGRAM DATA>
12800 to 2147483520 12,800 to 2,147,483,520 bits, in 128-bit steps

Note:
At 2 ch Combination, the setting range and Step are doubled.
At 4 ch Combination, the setting range and Step are quadrupled.

Function
Sets the data signal reception interval for Burst signal reception.

Example
To set the Burst signal reception interval to 12,800 bits:
> :SENSe:PATTern:BURSt:ELENgth 12800

Compatibility
Incompatible with existing models.

:SENSe:PATTern:BURSt:ELENgth?

Response
<numeric> = <NR1 NUMERIC RESPONSE DATA>

Function
Queries the data signal reception interval for Burst signal reception.

Example
> :SENSe:PATTern:BURSt:ELENgth?
< 12800

Compatibility
Incompatible with existing models.

:SENSe:PATTern:BURSt:CYCle <numeric>

Parameter
<numeric> = <DECIMAL NUMERIC PROGRAM DATA>
25600 to 2147483648 25,600 to 2,147,483,648 bits, in 128-bit steps

Note:
At 2 ch Combination, the setting range and Step are doubled.
At 4 ch Combination, the setting range and Step are quadrupled.

Function
Sets the Burst signal generation cycle.

Example
To set the Burst signal generation cycle to 25,600 bits:
> :SENSe:PATTern:BURSt:CYCle 25600

Compatibility
Incompatible with existing models.

:SENSe:PATTern:BURSt:CYCle?

Response
<numeric> = <NR1 NUMERIC RESPONSE DATA>

Function
Queries the Burst signal generation cycle.

Example
> :SENSe:PATTern:BURSt:CYCle?
< 25600

Compatibility
Incompatible with existing models.
Chapter 7  SCPI Commands

7.4.6.2  AUX Output setting commands

Figure 7.4.6.2-1  Pattern Sync setting  (For PRBS, Zero Substitution, and Data patterns)

Figure 7.4.6.2-2  Pattern Sync setting (For Mixed pattern)

Figure 7.4.6.2-3  Pattern Sync setting (For Sequence pattern)
### 7.4 ED Commands

#### Table 7.4.6.2-1 AUX Output setting commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>:OUTPut:SYNC:SOURce?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:OUTPut:SYNC:POSition?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:OUTPut:SYNC:MIXData:BLOCk?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:OUTPut:SYNC:MIXData:ROW?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:OUTPut:SYNC:SEQuence:BLOCk?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:OUTPut:SYNC:SEQuence:POSition?</td>
</tr>
</tbody>
</table>

**:OUTPut:SYNC:SOURce <source>[,<numeric>]**

- **Parameter**
  - `<source>` = `<CHARACTER PROGRAM DATA>`
  - CLOCk8: 1/8 clocks
  - CLOCk16: 1/16 clocks
  - CLOCk32: 1/32 clocks
  - CLOCk64: 1/64 clocks
  - PATTern: Pattern Sync (Variable)
  - SGAin: Sync Gain
  - ERRorout: Error Output
  - NCLock: 1/n clocks
  - `[<numeric>]` = `<DECIMAL NUMERIC PROGRAM DATA>`
  - Omits `<numeric>` when NCLock (1/n clocks) is not selected.
  - 16, 32, 64: 16, 32, 64 (for MU181040A-001)
  - 8 to 511: 8 to 511, in single step (for MU181040A-002, MU181040B-002)

**Function**
Sets the output signal for synchronization output.

**Example**
To set the output signal for synchronization output to 1/8 clocks:

```
> :OUTPut:SYNC:SOURce CLOCk8
```

To set the output signal for synchronization output to 1/511 clocks:

```
> :OUTPut:SYNC:SOURce NCLockn,511
```

**Compatibility**
Compatible with the MP1632C.
Partially compatible with the MP1776A.
Chapter 7  SCPI Commands

:OUTPut:SYNC:SOURce?

Response  
<source> = <CHARACTER RESPONSE DATA>
CLOC8, CLOC16, CLOC32, CLOC64, PATT, SGA, ERR, NCL
[numeric] = <DECIMAL NUMERIC RESPONSE DATA>
Omits <numeric> for other than NCL (1/n clocks).
8 to 511

Function  
Queries the output signal for synchronization output.

Example  
> :OUTPut:SYNC:SOURce?
< CLOC4
> :OUTPut:SYNC:SOURce?
< NCL,511

Compatibility  
Compatible with the MP1632C.
Partially compatible with the MP1776A.

:OUTPut:SYNC:POSition <numeric>

Parameter  
<numeric> = <DECIMAL NUMERIC PROGRAM DATA>
1 to 68719476657 1 to 68,719,476,657 bits, in 16-bit steps

Note:  
At 2 ch Combination, the setting range and Step are doubled.
At 4 ch Combination, the setting range and Step are quadrupled.

Function  
Sets the output position for synchronization output for Data, PRBS, and Zero Substitution patterns.

Example  
To set the synchronization output position to bit 1:
> :OUTPut:SYNC:POSition 1

Compatibility  
Incompatible with existing models.

:OUTPut:SYNC:POSition?

Response  
<numeric> = <NR1 NUMERIC RESPONSE DATA>
1 to 274877906625 1 to 274,877,906,625 bits

Function  
Queries the output position for synchronization output for Data, PRBS, and Zero Substitution patterns.

Example  
> :OUTPut:SYNC:POSition?
< 1

Compatibility  
Incompatible with existing models.
7.4  ED Commands

:OUTPut:SYNC:MIXData:BLOCk <numeric>

Parameter  
<numeric> = <DECIMAL NUMERIC PROGRAM DATA>
1 to 511  1 to 511 blocks, 1-block steps
The maximum setting value is the value set in Block No.

Function  
Sets the output block for synchronization output for Mixed Data pattern.

Example  
To set the output block for synchronization output to block 1:
> :OUTPut:SYNC:MIXData:BLOCk 1

Compatibility  
Incompatible with existing models.

:OUTPut:SYNC:MIXData:BLOCk?

Response  
<numeric> = <NR1 NUMERIC RESPONSE DATA>

Function  
Queries the output block for synchronization output for Mixed Data pattern.

Example  
> :OUTPut:SYNC:MIXData:BLOCk?
< 1

Compatibility  
Incompatible with existing models.

:OUTPut:SYNC:MIXData:ROW <numeric>

Parameter  
<numeric> = <DECIMAL NUMERIC PROGRAM DATA>
1 to 16  1 to 16 rows, in 1-row steps
The maximum setting value is the value set in Row No.

Function  
Sets the output Row position for synchronization output for Mixed Data pattern.

Example  
To set the synchronization output Row to 1 Row.
> :OUTPut:SYNC:MIXData:ROW 1

Compatibility  
Incompatible with existing models.

:OUTPut:SYNC:MIXData:ROW?

Response  
<numeric> = <NR1 NUMERIC RESPONSE DATA>
1 to 16  1 to 16 Rows

Function  
Queries the output Row position for synchronization output for Mixed Data pattern.

Example  
> :OUTPut:SYNC:MIXData:ROW?
< 1

Compatibility  
Incompatible with existing models.
### :OUTPut:SYNC:SEQUence:BLOCk <numeric>

**Parameter**

<numeric> = <DECIMAL NUMERIC PROGRAM DATA>

1 to 128  
1 to 128 blocks, 1-block steps

The maximum setting value is the value set in Block No.

**Function**

Sets the output block for synchronization output for Sequence pattern.

**Example**

To set the output block for synchronization output to block 128:

```plaintext
> :OUTPut:SYNC:SEQUence:BLOCk 128
```

**Compatibility**

Incompatible with existing models.

---

### :OUTPut:SYNC:SEQUence:BLOCk?

**Response**

<numeric> = <NR1 NUMERIC RESPONSE DATA>

**Function**

Queries the output block for synchronization output for Sequence pattern.

**Example**

```plaintext
> :OUTPut:SYNC:SEQUence:BLOCk?
< 128
```

**Compatibility**

Incompatible with existing models.

---

### :OUTPut:SYNC:SEQUence:POSition <numeric>

**Parameter**

<numeric> = <DECIMAL NUMERIC PROGRAM DATA>

1 to 1048497  
1 to 1,048,497 bits, in 16-bit steps

**Function**

Sets the output position for synchronization output for Sequence pattern.

**Example**

To set the synchronization output position to bit 1:

```plaintext
> :OUTPut:SYNC:SEQUence:POSition 1
```

**Compatibility**

Incompatible with existing models.

---

### :OUTPut:SYNC:SEQUence:POSition?

**Response**

<numeric> = <NR1 NUMERIC RESPONSE DATA>

1 to 1048497  
1 to 1,048,497 bits

**Function**

Queries the output position for synchronization output for Sequence pattern.

**Example**

```plaintext
> :OUTPut:SYNC:SEQUence:POSition?
< 1
```

**Compatibility**

Incompatible with existing models.
7.4.6.3  AUX Input setting commands

![Figure 7.4.6.3-1  Auto Input setting field](image)

Table 7.4.6.3-1  Auto Input setting commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Item</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>:INPut:AUX:USAGe?</td>
</tr>
</tbody>
</table>

:INPut:AUX:USAGe <usage>

**Parameter**

*<usage> = <CHARACTER PROGRAM DATA>*

BURSt  Burst Source

MASK  External Mask

CAPTure  Capture External Trigger

**Function**

Sets the usage of the common connector input.

**Example**

To set the usage of the common connector input to Burst Source:

> :INPut:AUX:USAGe BURSt

**Compatibility**

Incompatible with existing models.

:INPut:AUX:USAGe?

**Response**

*<usage> = <CHARACTER RESPONSE DATA>*

BURS, MASK, CAPT

**Function**

Queries the usage of the common connector input.

**Example**

> :INPut:AUX:USAGe?

< BURS

**Compatibility**

Incompatible with existing models.
7.4.6.4 Measurement Restart setting commands

![Measurement Restart setting field](image)

Figure 7.4.6.4-1 Measurement Restart setting field

Table 7.4.6.4-1 Measurement Restart setting commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Item</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>:SENSe:MEASure:MREStart?</td>
</tr>
</tbody>
</table>

:SENSe:MEASure:MREStart <data>,<clock>

Parameter

- `<data> = <BOOLEAN PROGRAM DATA>`
  - 0 or OFF: Does not restart measurement when the data threshold is changed (Measurement restart at data threshold change OFF).
  - 1 or ON: Restarts measurement when the data threshold is changed (Measurement restart at data threshold change ON).

- `<clock> = <BOOLEAN PROGRAM DATA>`
  - 0 or OFF: Does not restart measurement when the clock delay is changed (Measurement restart at clock delay change OFF).
  - 1 or ON: Restarts measurement when the clock delay is changed (Measurement restart at clock delay change ON).

Function

Sets the measurement restart condition.

Example

To set the measurement restart condition to "Measurement restart at data threshold change ON, Measurement restart at clock delay change ON".

> :SENSe:MEASure:MREStart 1,1

Compatibility

Incompatible with existing models.

:SENSe:MEASure:MREStart?

Response

- `<data>,<clock> = <NR1 NUMERIC RESPONSE DATA>`
  - 0: Measurement restart OFF
  - 1: Measurement restart ON

Function

Queries the measurement restart condition.

Example

> :SENSe:MEASure:MREStart?

< 1,1

Compatibility

Incompatible with existing models.
7.5 Optical Module Commands

This section describes the optical module commands. The target modules are as follows:

Before executing a setting/query command for the MU181600A or the MU181601A, specify the slot number of the module to be remotely controlled, by using the :MODule:ID command. Refer to Section 7.1 "Common Commands" and 9.1 "Basic Operation of Device Messages" for how to specify a slot number with the :MODule:ID command.

MU181600A Optical Transceiver (XFP)
MU181601A Optical Transceiver (SFP)

7.5.1 XFP/SFP module

This section describes the setting commands for the MU181600A Optical Transceiver (XFP) and the MU181601A Optical transceiver (SFP).

---

Figure 7.5.1-1  XFP screen

Figure 7.5.1-2  Memory Map setting (XFP)
### Table 7.5.1-1 XFP/SFP module

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>:SOURce:OPTical:SIGNal:OUTPut?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SOURce:OPTical:SIGNal:DRATe?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SOURce:OPTical:SIGNal:WLENght?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:OPTical:SIGNal:MEMory:ASCii?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:OPTical:SIGNal:MEMory:HEX?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:OPTical:POWerdown?</td>
</tr>
<tr>
<td>[6]</td>
<td>RX Rate Select</td>
<td>:SENSe:OPTical:RATeselect</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:OPTical:RATeselect?</td>
</tr>
</tbody>
</table>

**Figure 7.5.1-3 SFP screen**

**Figure 7.5.1-4 Memory Map setting (SFP)**

**Chapter 7 SCPI Commands**

[1:1.1] Optical Transceiver (SFP)
7.5 Optical Module Commands

:SOURce:OPTical:SIGNal:OUTPut <boolean>

Parameter
<boolean> = <BOOLEAN PROGRAM DATA>
OFF or 0  Optical output OFF
ON or 1  Optical output ON

Function
Sets optical output ON/OFF.

Example
To set optical output ON:
> :SOURce:OPTical:SIGNal:OUTPut ON

Compatibility
Incompatible with existing models.

:SOURce:OPTical:SIGNal:OUTPut?

Response
<boolean> = <NR1 NUMRIC RESPONSE DATA>
0  Optical output OFF
1  Optical output ON

Note:
"----" is returned when no XFP/SFP module is inserted.

Function
Queries the ON/OFF state of optical output.

Example
To query the ON/OFF state of optical output:
> :SOURce:OPTical:SIGNal:OUTPut?
< 1
< ---- (when no XFP/SFP module is inserted)

Compatibility
Incompatible with existing models.

:SOURce:OPTical:SIGNal:DRATe <numeric>

Parameter
<numeric> = <DECIMAL NUMERIC PROGRAM DATA>
For the recommended 850 nm XFP module:
9.9 to 11.1  9.9 to 11.1 Gbit/s, in 0.2-Gbit/s steps
Invalid for the recommended 1310 nm and 1550 nm XFP modules.
For XFP modules other than recommended ones:
9.5 to 12.5  9.5 to 12.5 Gbit/s, in 0.2-Gbit/s steps

Function
Sets the optical I/O interface data rate of the XFP module.
Enable/disable state and operating range vary depending on the recommended module.

Example
To set the I/O interface data rate of the XFP module to 9.9 Gbit/s:
> :SOURce:OPTical:SIGNal:DRATe 9.9

Compatibility
Incompatible with existing models.
Chapter 7  SCPI Commands

:SOURce:OPTical:SIGNal:DRATe?

Response  

<numERIC> = <NR1 NUMRIC RESPONSE DATA>
9.5 to 12.5  9.5 to 12.5 Gbit/s

"----" is returned for an invalid value.

Note:

"----" is returned when no XFP/SFP module is inserted.

Function  

Queries the optical I/O interface data rate of the XFP module.

Example  

To query the optical I/O interface data rate of the XFP module:

> :SOURce:OPTical:SIGNal:DRATe?
< 9.9
< "----" (when no XFP/SFP module is inserted)

Compatibility  

Incompatible with existing models.

:SOURce:OPTical:SIGNal:WLENgth?

Response  

<string> = <STRING RESPONSE DATA>
"XXXX"   Display   XXXX (Up to four digits, right-aligned)
"-----"   Disabled   ---- (Four characters)

Function  

Queries the optical wavelength used at the optical I/O interface.

Example  

To query the optical wavelength used at the optical I/O interface:

> :SOURce:OPTical:SIGNal:WLENgth?
< "850"
< "----- " (when no XFP/SFP module is inserted)

Compatibility  

Incompatible with existing models.
### :SENSe:OPTical:SIGNal:MEMory:DATA <page>,<address>,<data>

**Parameter**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;page&gt;</td>
<td>= &lt;NON-DECIMAL NUMERIC PROGRAM DATA&gt;</td>
</tr>
<tr>
<td></td>
<td>For the XFP module:</td>
</tr>
<tr>
<td></td>
<td>P00 to PFF 00 to FF (Hexadecimal)</td>
</tr>
<tr>
<td></td>
<td>For SFP module:</td>
</tr>
<tr>
<td></td>
<td>A0/A2</td>
</tr>
<tr>
<td>&lt;address&gt;</td>
<td>= &lt;CHARACTER PROGRAM DATA&gt;</td>
</tr>
<tr>
<td></td>
<td>A00 to AFF 00 to FF</td>
</tr>
<tr>
<td>&lt;data&gt;</td>
<td>= &lt;STRING PROGRAM DATA&gt;</td>
</tr>
<tr>
<td></td>
<td>&quot;00&quot; to &quot;FF&quot; 00 to FF</td>
</tr>
</tbody>
</table>

Each byte is delimited with a comma (,) (up to 32 bytes).

**Note:**

Data written outside the address range is invalid.

**Function**

Sets the module register.

**Example**

To set data "FF,FF,FF,FF" from Address 00 on Page 00 in the register of the XFP module:

```plaintext
> :SENSe:OPTical:SIGNal:MEMory:DATA P00, A00, "FF,FF,FF,FF"
```

**Compatibility**

Incompatible with existing models.
<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;page&gt; = &lt;CHARACTER PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>For the XFP module:</td>
</tr>
<tr>
<td></td>
<td>P00 to PFF 00 to FF (Hexadecimal)</td>
</tr>
<tr>
<td></td>
<td>For the SFP module:</td>
</tr>
<tr>
<td></td>
<td>A0/A2</td>
</tr>
<tr>
<td>&lt;address&gt;</td>
<td>= &lt;CHARACTER PROGRAM DATA&gt;</td>
</tr>
<tr>
<td></td>
<td>A00 to AFF 00 to FF</td>
</tr>
<tr>
<td>&lt;size&gt;</td>
<td>= &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
</tr>
<tr>
<td></td>
<td>0 to 32 0 to 32 byte in 1 step</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;data&gt; = &lt;STRING RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>A character string other than 0x20-0x7e is output replaced by '?.'</td>
</tr>
<tr>
<td></td>
<td>Each byte is delimited with a comma (,).</td>
</tr>
</tbody>
</table>

**Note:**

When the size exceeding the address range is specified, data within the address range is returned.

"----" is returned when no XFP/SFP module is inserted.

**Function**

Queries the data in the register on the module in ASCII characters.

**Example**

To query the 4-byte data from Address 00 on Page 00 in the register of the XFP module in ASCII character:

```
> :SENSe:OPTical:SIGNal:MEMory:ASCii? P00,A00,4
< "?,a,b,?"
< "-,-,-,-" (non-display area)
```

**Compatibility**

Incompatible with existing models.
### :SENSe:OPTical:SIGNal:MEMory:HEX? <page>,<address>,<size>

**Parameter**

- `<page>` = `<CHARACTER PROGRAM DATA>`
  - For the XFP module
    - P00 to PFF  00 to FF (Hexadecimal)
  - For the SFP module
    - A0/A2
- `<address>` = `<CHARACTER PROGRAM DATA>`
  - A00 to AFF  00 to FF
- `<size>` = `<DECIMAL NUMERIC PROGRAM DATA>`
  - 0 to 32  0 to 32 byte In 1 step

**Response**

- `<data>` = `<STRING RESPONSE DATA>`
  - "00" to "FF"  00 to FF
  - Each byte is delimited with a comma (,).

**Note:**

- When the size exceeding the address range is specified, data within the address range is returned.
- "----" is returned when no XFP/SFP module is inserted.

**Function**

Queries the data in the register on the module in hexadecimal format.

**Example**

To query the 4-byte data from Address 00 on Page 00 in the register of the XFP module in hexadecimal format:

```plaintext
> :SENSe:OPTical:SIGNal:MEMory:HEX? P00,A00,4
< "FF,FF,FF,FF"
```

**Compatibility**

Incompatible with existing models.
Chapter 7  SCPI Commands

:CALCulate:OPTical:STATus? <string>

Parameter

<table>
<thead>
<tr>
<th>&lt;string&gt;</th>
<th>STRING PROGRAM DATA</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;READY&quot;</td>
<td>Ready</td>
</tr>
<tr>
<td>&quot;TX:LASer&quot;</td>
<td>Tx: Laser</td>
</tr>
<tr>
<td>&quot;TX:UNLock&quot;</td>
<td>Tx: Unlock (Cannot be set for the SFP module (MU181601A).)</td>
</tr>
<tr>
<td>&quot;RX:LOS&quot;</td>
<td>Rx: LOS</td>
</tr>
<tr>
<td>&quot;RX:CDR&quot;</td>
<td>Rx: CDR unlock (Cannot be set for the SFP module (MU181601A).)</td>
</tr>
</tbody>
</table>

Response

| <string> | STRING RESPONSE DATA |

Table 7.5.1-2  Response format

<table>
<thead>
<tr>
<th>Type</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Form1</td>
<td>&quot;Occur&quot;</td>
<td>When the specified status occurs.</td>
</tr>
<tr>
<td>&quot;None&quot;</td>
<td></td>
<td>When the specified status does not occur.</td>
</tr>
<tr>
<td>&quot;-----&quot;</td>
<td></td>
<td>When no data corresponds to the query</td>
</tr>
</tbody>
</table>

Function
Queries the status.

Example
To query whether optical output of the module is normal:

```
> :CALCulate:OPTical:STATus? "TX:LASer"
< "None"
```

Compatibility
Incompatible with existing models.

:SOURce:OPTical:POWerdown <power>

Parameter

<table>
<thead>
<tr>
<th>&lt;power&gt;</th>
<th>CHARACTER PROGRAM DATA</th>
</tr>
</thead>
<tbody>
<tr>
<td>STANdbay</td>
<td>Standby mode</td>
</tr>
<tr>
<td>OPERation</td>
<td>Normal operation mode</td>
</tr>
</tbody>
</table>

Function
Sets the XFP module to the standby mode.

Example
To set the XFP module to the standby mode:

```
> :SOURce:OPTical:POWerdown STANdbay
```

Compatibility
Incompatible with existing models.
### 7.5 Optical Module Commands

#### :SOURce:OPTical:POWerdown?

| Response | \(<\text{power}\> = \langle\text{CHARACTER RESPONSE DATA}\rangle\)  
|-----------|------------------------------------------------|
|           | STAN, OPER  

**Note:**

"----" is returned when no XFP module is inserted.

**Function**

Queries the standby mode setting state of the XFP module.

**Example**

> :SOURce:OPTical:POWerdown?

< STANby  

< "----" (when no XFP module is inserted)

**Compatibility**

Incompatible with existing models.

#### :SOURce:OPTical:RATeselect \(<\text{ratesel}\>\)

| Parameter | \(<\text{ratesel}\> = \langle\text{CHARACTER PROGRAM DATA}\rangle\)  
|-----------|------------------------------------------------|
|           | REDuced Enables band restriction  
|           | FULL Enables band restriction  

**Function**

Enables/disables band restriction for the SFP module.

**Example**

To enable band restriction for the SFP module:

> :SOURce:OPTical:RATeselect REDuced

**Compatibility**

Incompatible with existing models.

#### :SOURce:OPTical:RATeselect?

| Response | \(<\text{ratesel}\> = \langle\text{CHARACTER RESPONSE DATA}\rangle\)  
|-----------|------------------------------------------------|
|           | RED, FULL  

**Note:**

"----" is returned when no SFP module is inserted.

**Function**

Queries the band restriction setting state for the SFP module.

**Example**

> :SOURce:OPTical:RATeselect?

< "----" (when no SFP module is inserted)  

< FULL

**Compatibility**

Incompatible with existing models.
7.6 Automatic Measurement Commands

This section describes the commands for the measurement application. Before starting to use the automatic measurement commands, specify the automatic measurement function to be operated by using the "SYSTem:CFUNction" command. Refer to Section 7.1 "Common Commands" and 9.1 "Basic Operation of Device Messages" for how to specify a slot number with the :SYSTem:CFUNction.

7.6.1 ISI measurement

This section describes the commands for ISI measurement.

7.6.1.1 Measurement setting commands

![ISI measurement window]

Figure 7.6.1.1-1  ISI measurement window

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items/Read Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Setting item selection</td>
<td>:DISPlay:RESult:ISI:ITEM</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:DISPlay:RESult:ISI:ITEM?</td>
</tr>
<tr>
<td>2</td>
<td>Slot selection</td>
<td>:SENSe:MEASure:ISI:SELSlot</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:MEASure:ISI:SELSlot?</td>
</tr>
<tr>
<td>3</td>
<td>Start</td>
<td>:SENSe:MEASure:STARt</td>
</tr>
<tr>
<td>4</td>
<td>Stop</td>
<td>:SENSe:MEASure:ISI:STOP</td>
</tr>
<tr>
<td>5</td>
<td>No label (Query for measurement state)</td>
<td>:SENSe:MEASure:ISI:STATe?</td>
</tr>
</tbody>
</table>
### 7.6 Automatic Measurement Commands

#### Table 7.6.1-1  ISI measurement setting commands (Cont’d)

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items/Read Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>:DISPlay:RESult:ISI:TIME?</td>
</tr>
<tr>
<td>[7]</td>
<td>Query for Date&amp;Time</td>
<td>:SENSe:MEASure:ISI:DTIMe?</td>
</tr>
<tr>
<td>[8]</td>
<td>Query for measurement start time</td>
<td>:SENSe:MEASure:ISI:STARt?</td>
</tr>
<tr>
<td>[9]</td>
<td>Query for measurement elapsed time</td>
<td>:SENSe:MEASure:ISI:ELAPsed?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:DISPlay:RESult:ISI:SCALe?</td>
</tr>
<tr>
<td>[18]</td>
<td>Number of Block</td>
<td>:DISPlay:RESult:ISI:BLOCk?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:MEASure:ISI:MODE?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:MEASure:ISI:PERiod?</td>
</tr>
<tr>
<td>[22]</td>
<td>No label (Query for measurement result)</td>
<td>:CALCulate:DATA:ISI?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:DISPlay:RESult:ISI:FVIeW?</td>
</tr>
</tbody>
</table>

**:DISPlay:RESult:ISI:ITEM <mode>**

- **Parameter**
  - `<mode>` = `<CHARACTER PROGRAM DATA>`
  - `COUNt` = Error Count
  - `RATE` = Error Rate

- **Function**
  Selects the measurement items displayed during ISI measurement.

- **Example**
  To set the measurement items displayed during ISI measurement to Error Count:
  ```plaintext```
  ```
  > :DISPlay:RESult:ISI:ITEM COUNt
  ```

- **Compatibility**
  Incompatible with existing models.
Chapter 7  SCPI Commands

:DISPlay:RESult:ISI:ITEM?

Response  
<mode> = <CHARACTER RESPONSE DATA>
COUN, RATE

Function  
Queries the measurement items displayed during ISI measurement.

Example  
> :DISPlay:RESult:ISI:ITEM?
< COUN

Compatibility  
Incompatible with existing models.

:SENSe:MEASure:ISI:SELSlot <slot>[,<unit>]

Parameter  
<slot> = <CHARACTER PROGRAM DATA>
SLOT1 to SLOT6     Slots No. 1 to No. 6
1 to 6 when using the MP1800A, 1 to 4 when using the MT1810A
Any of 1 through 4 can be selected when the target slot constitutes a 1- to
4-CH combination.

[unit] = <DECIMAL NUMERIC PROGRAM DATA>
1 to 4     Mainframe Nos. 1 to 4
When using two or more MT1810A units in serial connection, specify the
mainframe number.
Can be omitted. Mainframe No. 1 is specified when omitted.

Function  
Selects the slot and combination group for which to perform ISI
measurement.

Example  
To set Slot1 to perform ISI measurement:
> :SENSe:MEASure:ISI:SELSlot SLOT1

Compatibility  
Incompatible with existing models.

:SENSe:MEASure:ISI:SELSlot? [<unit>]

Parameter  
[unit] = <DECIMAL NUMERIC PROGRAM DATA>
1 to 4     Mainframe Nos. 1 to 4
When using two or more MT1810A units in serial connection, specify the
mainframe number.
Can be omitted. Mainframe No. 1 is specified when omitted.

Response  
<slot> = <CHARACTER RESPONSE DATA>
SLOT1 to SLOT6     Slots No. 1 to No. 6
The first slot number is returned if the target slot constitutes a
combination.
1 to 6 when using the MP1800A, 1 to 4 when using the MT1810A

Function  
Queries the slot and combination group for which to perform ISI
measurement.

Example  
> :SENSe:MEASure:ISI:SELSlot?  
< SLOT1

Compatibility  
Incompatible with existing models.
7.6 Automatic Measurement Commands

:SENSe:MEASure:STARt

Function
Stats ISI measurement.

Example
> :SENSe:MEASure:STARt

Compatibility
Incompatible with existing models.

:SENSe:MEASure:ISI:STOP

Function
Stops ISI measurement

Example
> :SENSe:MEASure:ISI:STOP

Compatibility
Incompatible with existing models.

:SENSe:MEASure:ISI:STATe?

Response
<boolean> = <NR1 NUMERIC RESPONSE DATA>
1   ISI measurement has been started.
0   ISI measurement has been stopped.

Function
Queries the ISI measurement processing state.

Example
> :SENSe:MEASure:ISI:STATe?
< 1

Compatibility
Incompatible with existing models.

:DISPlay:RESult:ISI:TIME <type>

Parameter
<type> = <CHARACTER PROGRAM DATA>
DTIMe  Displays the current date and time.
STARt  Displays the measurement start time.
ELAPsed Displays the elapsed time based on the measurement period.

Function
Selects the ISI measurement time display type.

Example
To set the ISI measurement time display type to measurement start time (Start Time):
> :DISPlay:RESult:ISI:TIME STARt

Compatibility
Incompatible with existing models.

:DISPlay:RESult:ISI:TIME?

Response
<type> = <CHARACTER RESPONSE DATA>
DTIM, STAR, ELAP

Function
Queries the ISI measurement time display type.

Example
> :DESPlay:RESult:ISI:TIME?
< STAR

Compatibility
Incompatible with existing models.
Chapter 7  SCPI Commands

:SENSe:MEASure:ISI:DTIMe?

Response

- `<year>` = `<NR1 NUMERIC RESPONSE DATA>`
  - 2000 to 2036  2000 to 2036 year
- `<month>` = `<NR1 NUMERIC RESPONSE DATA>`
  - 1 to 12  January to December
- `<day>` = `<NR1 NUMERIC RESPONSE DATA>`
  - 1 to 31  1st to 31st
- `<hour>` = `<NR1 NUMERIC RESPONSE DATA>`
  - 0 to 23  0 to 23 hours
- `<minute>` = `<NR1 NUMERIC RESPONSE DATA>`
  - 0 to 59  0 to 59 minutes
- `<second>` = `<NR1 NUMERIC RESPONSE DATA>`
  - 0 to 59  0 to 59 seconds

Function
Queries the current date and time during ISI measurement.

Example
> :SENSe:MEASure:ISI:DTIMe?
< 2006,4,1,23,59,59

Compatibility
Incompatible with existing models.

:SENSe:MEASure:ISI:STARt?

Response

- `<year>` = `<NR1 NUMERIC RESPONSE DATA>`
  - 0, 2000 to 2036  2000 to 2036 year
- `<month>` = `<NR1 NUMERIC RESPONSE DATA>`
  - 0, 1 to 12  January to December
- `<day>` = `<NR1 NUMERIC RESPONSE DATA>`
  - 0, 1 to 31  1st to 31st
- `<hour>` = `<NR1 NUMERIC RESPONSE DATA>`
  - 0 to 23  0 to 23 hours
- `<minute>` = `<NR1 NUMERIC RESPONSE DATA>`
  - 0 to 59  0 to 59 minutes
- `<second>` = `<NR1 NUMERIC RESPONSE DATA>`
  - 0 to 59  0 to 59 seconds

Note:
The following is output when no measurement is performed:
0,0,0,0,0,0

Function
Queries the ISI measurement start time.

Example
> :SENSe:MEASure:ISI:STARt?
< 2006,4,1,23,59,59

Compatibility
Incompatible with existing models.
7.6 Automatic Measurement Commands

:SENSe:MEASure:ISI:ELAPsed?

Response

Response: <day> = <NR1 NUMERIC RESPONSE DATA>
0 to 99  0 to 99 days
<hour> = <NR1 NUMERIC RESPONSE DATA>
0 to 23  0 to 23 hours
<minute> = <NR1 NUMERIC RESPONSE DATA>
0 to 59  0 to 59 minutes
<second> = <NR1 NUMERIC RESPONSE DATA>
0 to 59  0 to 59 seconds

Note:

The following is output when no measurement is performed:

0,0,0,0

Function

Queries the ISI measurement elapsed time.

Example

> :SENSe:MEASure:ISI:ELAPsed?
< 99,23,59,59

Compatibility

Incompatible with existing models.

:DISPlay:RESult:ISI:SCALe <top>,<bottom>

Parameter

Parameter: <top> = <CHARACTER PROGRAM DATA>
E_3 to E_14  E–3 to E–14, in single steps (When the measurement item is Rate)
E_2 to E_18  E+2 to E+18, in single steps (When the measurement item is Count)

Parameter: <bottom> = <CHARACTER PROGRAM DATA>
E_6 to E_18  E–6 to E–18, in single steps (When the measurement item is Rate)
0, E_0 to E_14  0, E+0 to E+14, in single steps (When the measurement item is Count)

The <bottom> value must be greater than the <top> value when set to Rate, and the <top> value must be greater than the <bottom> value when set to Count).

Function

Sets the ISI measurement vertical scale.

Example

To set the ISI measurement vertical scale as top: E–3 and bottom: E–5 when the measurement item is Rate:

> :DISPlay:RESult:ISI:SCALe E_3,E_5

Compatibility

Incompatible with existing models.
### :DISPlay:RESult:ISI:SCALe?

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;top&gt;,&lt;bottom&gt; = &lt;CHARACTER RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>E_18 to E_0      E–18 to E+0, in single steps (When the measurement item is Rate)</td>
</tr>
<tr>
<td></td>
<td>0, E_0 to E_18  0, E+0 to E+18, in single steps (When the measurement item is Count)</td>
</tr>
</tbody>
</table>

**Function**
Queries the ISI measurement vertical scale when the measurement item is Rate.

**Example**
```
> :DISPlay:RESult:ISI:SCALe?
< E_3,E_5
```

**Compatibility**
Incompatible with existing models.

---

### :DISPlay:RESult:ISI:MMOVe <move>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;move&gt; = &lt;CHARACTER PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>LEFT</td>
<td>Moves to the left</td>
</tr>
<tr>
<td>RIGHT</td>
<td>Moves to the right</td>
</tr>
</tbody>
</table>

**Function**
Sets the ISI measurement marker moving direction.

**Example**
To move the ISI measurement marker to the left:
```
> :DISPlay:RESult:ISI:MMOVe LEFT
```

**Compatibility**
Incompatible with existing models.
### :DISPlay:RESult:ISI:MARKer?

**Response**

- `<pos>` = `<NR1 NUMERIC RESPONSE DATA>`
  - 1 to 2275483648 (1 to 2,275,483,648 bits)
- Maximum value: Maximum length of Mixed Data pattern Multiplied by 4 in the case of Combination.
- `<block>` = `<NR1 NUMERIC RESPONSE DATA>`
  - 1 to 256 (1 to 256 blocks)
- `<count>` = `<STRING RESPONSE DATA>`
- `<rate>` = `<STRING RESPONSE DATA>`

**Table 7.6.1.1-2 Response Format**

<table>
<thead>
<tr>
<th>Form Type</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Form 1 Integer</td>
<td>&quot;XXXXXXX&quot;</td>
<td>When 0 to 9999999</td>
</tr>
<tr>
<td>Form 1 Fraction</td>
<td>&quot;X.XXXXXXX&quot;</td>
<td>When 1.0000E07 to 9.9999E17</td>
</tr>
<tr>
<td></td>
<td>&quot;--------&quot;</td>
<td>When no data corresponds to the query.</td>
</tr>
<tr>
<td>Form 2 Fraction</td>
<td>&quot;X.XXXE-XX&quot;</td>
<td>When 0.0001E–18 to 1.0000E00</td>
</tr>
<tr>
<td></td>
<td>&quot;--------&quot;</td>
<td>When no data corresponds to the query.</td>
</tr>
</tbody>
</table>

**Function**

Queries the error count and error rate at the marker position.

**Example**

`> :DISPlay:RESult:ISI:MARKer?`

`< 134217728, 64, "      0", "0.0001E-18"

**Compatibility**

Incompatible with existing models.

### :DISPlay:RESult:ISI:ZIN

**Function**

Specifies the Bit Width of the Pattern Position indicated by the ISI measurement marker to be the Measurement Length of the measurement to be performed next.

**Example**

`> :DISPlay:RESult:ISI:ZIN

**Compatibility**

Incompatible with existing models.

### :DISPlay:RESult:ISI:ZOUT

**Function**

Changes the result display from the current ISI measurement result to the result of a higher layer.

**Example**

`> :DISPlay:RESult:ISI:ZOUT

**Compatibility**

Incompatible with existing models.
Chapter 7  SCPI Commands

:DISPlay:RESult:ISI:DEPTh?

Response  
<depth1> = <NR1 NUMERIC RESPONSE DATA>
1 to 6   The layer displaying the measurement result
<depth2> = <NR2 NUMERIC RESPONSE DATA>
1 to 6   All layers

*Note:*  
<depth1> and <depth2> correspond to the displayed layer and all
layers, respectively.

Function  Queries the ISI measurement layer.

Example  
> :DISPlay:RESult:ISI:DEPTh?
< 2,5

Compatibility  Incompatible with existing models.

:DISPlay:RESult:ISI:PLENgth?

Response  
<length> = <NR1 NUMERIC RESPONSE DATA>
2 to 2281701376  2 to 2,281,701,376 bits
Maximum value: Maximum length of Mixed Data pattern
Multiplied by 4 in the case of Combination.

Function  Queries the ISI measurement Pattern Length.

Example  
> :DISPlay:RESult:ISI:PLENgth?
< 134217728

Compatibility  Incompatible with existing models.

:DISPlay:RESult:ISI:MLENgth?

Response  
<length> = <NR1 NUMERIC RESPONSE DATA>
2 to 2281701376  2 to 2,281,701,376 bits
Maximum value: Maximum length of Mixed Data pattern
Multiplied by 4 in the case of Combination.

Function  Queries the ISI Measurement Length.

Example  
> :DISPlay:RESult:ISI:MLENgth?
< 134217728

Compatibility  Incompatible with existing models.
### Automatic Measurement Commands

#### :DISPlay:RESult:ISI:BLOCk?

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;block&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2 to 256</td>
</tr>
<tr>
<td>Function</td>
<td>Queries the ISI measurement block division number.</td>
</tr>
<tr>
<td>Example</td>
<td>&gt; :DISPlay:RESult:ISI:BLOCk?</td>
</tr>
<tr>
<td></td>
<td>&lt; 64</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Incompatible with existing models.</td>
</tr>
</tbody>
</table>

#### :DISPlay:RESult:ISI:BWIDth?

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;width&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1 to 285212672</td>
</tr>
<tr>
<td></td>
<td>1 to 285,212,672 bits</td>
</tr>
<tr>
<td>Function</td>
<td>Queries the number of bits within one block in ISI measurement.</td>
</tr>
<tr>
<td>Example</td>
<td>&gt; :DISPlay:RESult:ISI:BWIDth?</td>
</tr>
<tr>
<td></td>
<td>&lt; 2097152</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Incompatible with existing models.</td>
</tr>
</tbody>
</table>

#### :SENSe:MEASure:ISI:MODE <mode>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;mode&gt; = &lt;CHARACTER PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>SINGle</td>
<td>Single measurement</td>
</tr>
<tr>
<td>UNTimed</td>
<td>Untimed measurement</td>
</tr>
<tr>
<td>Function</td>
<td>Selects the measurement processing mode.</td>
</tr>
<tr>
<td>Example</td>
<td>To set the measurement processing mode to Single measurement:</td>
</tr>
<tr>
<td></td>
<td>&gt; :SENSe:MEASure:ISI:MODE SINGle</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Incompatible with existing models.</td>
</tr>
</tbody>
</table>

#### :SENSe:MEASure:ISI:MODE?

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;mode&gt; = &lt;CHARACTER RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>SING, UNT</td>
<td>Queries the synchronization output type.</td>
</tr>
<tr>
<td>Example</td>
<td>&gt; :SENSe:MEASure:ISI:MODE?</td>
</tr>
<tr>
<td></td>
<td>&lt; SING</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Incompatible with existing models.</td>
</tr>
</tbody>
</table>
Chapter 7  SCPI Commands

:SENSe:MEASure:ISI:PERiod <day>,<hour>,<min>,<second>

Parameter

<day>,<hour>,<min>,<second> = <DECIMAL NUMERIC PROGRAM DATA>
<day>  00 to 99  0 to 99 days, in 1-day steps
<hour>  00 to 23  0 to 23 hours, in 1-hour steps
<min>   00 to 59  0 to 59 minutes, in 1-minute steps
<second> 00 to 59  0 to 59 seconds, in 1-second steps

Function
Sets the ISI measurement time.

Example
To set the ISI measurement time to 99 days, 23 hours, 59 minutes, 59 seconds:

Compatibility
Incompatible with existing models.

:SENSe:MEASure:ISI:PERiod?

Response

<day>,<hour>,<min>,<second> = <NR1 NUMERIC RESPONSE DATA>
<day>  0 to 99  0 to 99 days
<hour>  00 to 23  0 to 23 hours
<min>   00 to 59  0 to 59 minutes
<second> 00 to 59  0 to 59 seconds

Note:
Each response consists of two digits (for example, 01,01,01,01).

Function
Queries the ISI measurement time.

Example
> :SENSe:MEASure:ISI:PERiod?
< 99,23,59,59

Compatibility
Incompatible with existing models.
### :CALCulate:DATA:ISI? <pos>

**Parameter**  
\(<\text{pos}\> = <\text{DECIMAL NUMERIC PROGRAM DATA}>\)  
2 to 2281701376  
2 to 2,281,701,376 bits  
Maximum value: Maximum length of Mixed Data pattern  
Multiplied by 4 in the case of Combination.

**Response**  
\(<\text{count}\> = <\text{STRING RESPONSE DATA}>\)  
Form1  
\(<\text{rate}\> = <\text{STRING RESPONSE DATA}>\)  
Form2

<table>
<thead>
<tr>
<th>Form</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Form1</td>
<td>&quot;XXXXXXX&quot;</td>
<td>When 0 to 9999999</td>
</tr>
<tr>
<td>Integer type</td>
<td>&quot;X.XXXEEXX&quot;</td>
<td>When 1.0000E07 to 9.9999E17</td>
</tr>
<tr>
<td>Form2</td>
<td>&quot;X.XXXE-XX&quot;</td>
<td>When 0.0001E–18 to 1.0000E00</td>
</tr>
<tr>
<td>Fraction type</td>
<td>&quot;------&quot;</td>
<td>When no data corresponds to the query.</td>
</tr>
</tbody>
</table>

**Function**  
Queries the error count and error rate of the block including the specified position.

**Example**   
> :CALCulate:DATA:ISI? 134217728  
< " 0", "0.0001E-18"

**Compatibility**  
Incompatible with existing models.

### :DISPlay:RESult:ISI:MTIMes?

**Response**  
\(<\text{numeric}\> = <\text{NR1 NUMERIC RESPONSE DATA}>\)  
1 to 26367187500000  
1 to 26,367,187,500,000 times

**Function**  
Queries the number of pattern repetition times for ISI measurement.

**Example**   
> :DISPlay:RESult:ISI:MTIMes?  
< 1

**Compatibility**  
Incompatible with existing models.

### :DISPlay:RESult:ISI:FVIew <boolean>

**Parameter**  
\(<\text{boolean}\> = <\text{BOOLEAN PROGRAM DATA}>\)  
ON or 1  
Full View mode  
OFF or 0  
Normal view mode

**Function**  
This command is valid for the lower-most layer during ISI measurement, when Combination is configured. Used to set the graph display mode.

**Example**   
To set Full View mode for the graph display mode during ISI measurement when Combination is configured:  
> :DISPlay:RESult:ISI:FVIew ON

**Compatibility**  
Incompatible with existing models.
### :DISPlay:RESult:ISI:FVlew?

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;numeric&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Full View mode.</td>
</tr>
<tr>
<td>0</td>
<td>Normal view mode.</td>
</tr>
</tbody>
</table>

**Function**: Queries the graph display mode during ISI measurement.

**Example**

```
> :DISPlay:RESult:ISI:FVlew?
< 1
```

**Compatibility**: Incompatible with existing models.
### 7.6.1.2 File menu setting commands

Table 7.6.2-1  File menu setting commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
</table>

:SYSTem:MMEMory:ISI:RECall <file_name>

**Parameter**

- `<file_name>` = <STRING PROGRAM DATA>
- "<drv>:\[<dir>]\<file>"
- `<drv>` = C, D, E, F
- `<dir>` = <dir1>\<dir2>\... (Omitted when root directory)
- `<file>` = File name

**Function**

Opens the ISI measurement result data.

**Example**

> :SYSTem:MMEMory:ISI:RECall "C:\Test\example"

**Compatibility**

Incompatible with existing models.

:SYSTem:MMEMory:ISI:STORe <file_name>,<file_type>

**Parameter**

- `<file_name>` = <STRING PROGRAM DATA>
- "<drv>:\[<dir>]\<file>"
- `<drv>` = C, D, E, F
- `<dir>` = <dir1>\<dir2>\... (Omitted when root directory)
- `<file>` = File name
- `<file_type>` = <CHARACTER PROGRAM DATA>

**Function**

Saves the ISI measurement result data.

**Note:**

The settings will not be read from the saved file if the file name is changed.

**Example**

To save the ISI measurement result data by specifying the save destination, file name, and file format:

> :SYSTem:MMEMory:ISI:STORe "C:\Test\example",TXT

**Compatibility**

Incompatible with existing models.
### :SYSTem:PRINT:ISI <list>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;list&gt;</td>
<td>= &lt;CHARACTER PROGRAM DATA&gt;</td>
</tr>
<tr>
<td>ISI</td>
<td>ISI Result</td>
</tr>
</tbody>
</table>

**Function**
Prints the ISI measurement result data.

**Example**
>` :SYSTem:PRINT:ISI ISI`

**Compatibility**
Incompatible with existing models.
7.6 Automatic Measurement Commands

7.6.2 Eye Margin measurement

This section describes the commands for Eye Margin measurement.

7.6.2.1 Eye Margin measurement setting commands

![Eye Margin measurement screen]

**Figure 7.6.2.1-1  Eye Margin measurement screen**

**Table 7.6.2.1-1  Eye Margin measurement setting commands**

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>:SENSe:MEASure:EMARgin:MARGin:SLOT?</td>
</tr>
<tr>
<td>[6]</td>
<td>No label (Query for measurement state)</td>
<td>:SENSe:MEASure:EMARgin:STATe?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:MEASure:EMARgin:MARGin:THReshold?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:MEASure:EMARgin:MARGin:RESolution?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:MEASure:EMARgin:MARGin:ASEarch?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:MEASure:EMARgin:TIME?</td>
</tr>
</tbody>
</table>
Table 7.6.2.1-1  Eye Margin measurement setting commands (Cont’d)

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td>Query for measurement start time</td>
<td>:SENSe:MEASure:EMARgin:STARt?</td>
</tr>
<tr>
<td>13</td>
<td>Query for measurement elapsed time</td>
<td>:SENSe:MEASure:EMARgin:ELAPsed?</td>
</tr>
<tr>
<td>14</td>
<td>Query for measurement result</td>
<td>:CALCulate:DATA:EMARgin?</td>
</tr>
<tr>
<td>15</td>
<td>No label (Query for measurement status)</td>
<td>:CALCulate:DATA:EMARgin:STATus?</td>
</tr>
</tbody>
</table>

:SENSe:MEASure:EMARgin:MARGin:ASLot

Function
Selects all Eye Margin measurement slots to be set to ON.

Example
> :SENSe:MEASure:EMARgin:MARGin:ASLot

Compatibility
Incompatible with existing models.

:SENSe:MEASure:EMARgin:MARGin:RSLot

Function
Selects all Eye Margin measurement slots to be set to OFF.

Example
> :SENSe:MEASure:EMARgin:MARGin:RSLot

Compatibility
Incompatible with existing models.

:SENSe:MEASure:EMARgin:MARGin:SLOT <slot>,<boolean>[,<unit>]

Parameter
- `<slot>` = <CHARACTER PROGRAM DATA>
  - SLOT1 to SLOT6  Slots No. 1 to No. 6
  - 1 to 6 when using the MP1800A, 1 to 4 when using the MT1810A
- `<boolean>` = <BOOLEAN PROGRAM DATA>
  - ON or 1  ON
  - OFF or 0  OFF
- `<unit>` = <DECIMAL NUMERIC PROGRAM DATA>
  - 1 to 4  Mainframe Nos. 1 to 4

When using two or more MT1810A units in serial connection, specify the mainframe number.

Can be omitted. Mainframe No. 1 is specified when omitted.

Function
Selects the Eye Margin measurement slot.

Example
To select Eye Margin measurement Slot 1 to be ON:
> :SENSe:MEASure:EMARgin:MARGin:SLOT SLOT1,ON

Compatibility
Incompatible with existing models.
7.6 Automatic Measurement Commands

:**SENSe:MEASure:EMARgin:MARGin:SLOT? <slot>[,<unit>]**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;slot&gt; = CHARACTER PROGRAM DATA</td>
<td>SLOT1 to SLOT6 Slots No. 1 to No. 6</td>
</tr>
<tr>
<td>1 to 6 when using the MP1800A, 1 to 4 when using the MT1810A</td>
<td>[&lt;unit&gt;] = DECIMAL NUMERIC PROGRAM DATA</td>
</tr>
<tr>
<td>1 to 4 Mainframe Nos. 1 to 4</td>
<td>When using two or more MT1810A units in serial connection, specify the mainframe number.</td>
</tr>
<tr>
<td>Can be omitted. Mainframe No. 1 is specified when omitted.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Response</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;boolean&gt; = NR1 NUMERIC RESPONSE DATA</td>
<td>1 ON</td>
</tr>
<tr>
<td>0 OFF</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Queries whether the specified slot is targeted for the Eye Margin measurement.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Example</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>To query whether the slot 1 is targeted for the Eye Margin measurement:</td>
<td>&gt; :SENSe:MEASure:EMARgin:MARGin:SLOT? SLOT1 &lt; 1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Compatibility</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Incompatible with existing models.</td>
<td></td>
</tr>
</tbody>
</table>

:**SENSe:MEASure:STARt**

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Starts the Eye Margin measurement.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Example</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&gt; :SENSe:MEASure:STARt</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Compatibility</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Incompatible with existing models.</td>
<td></td>
</tr>
</tbody>
</table>

:**SENSe:MEASure:STOP**

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stops the Eye Margin measurement.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Example</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&gt; :SENSe:MEASure:STOP</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Compatibility</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Incompatible with existing models.</td>
<td></td>
</tr>
</tbody>
</table>

:**SENSe:MEASure:EMARgin:STATe?**

<table>
<thead>
<tr>
<th>Response</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;boolean&gt; = NR1 NUMERIC RESPONSE DATA</td>
<td>1 Starts measurement</td>
</tr>
<tr>
<td>0 Stops measurement</td>
<td>–1 Measurement failure</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Queries the Eye Margin measurement state.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Example</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&gt; :SENSe:MEASure:EMARgin:STATe?</td>
<td>&lt; 1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Compatibility</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compatible with the MP1632C.</td>
<td></td>
</tr>
</tbody>
</table>
Chapter 7  SCPI Commands

:SENSe:MEASure:EMARgin:MARGin:THReshold <thre>
Parameter  
<thre> = <CHARACTER PROGRAM DATA>
E_3 to E_12  1E–3 to 1E–12, in single steps
Function  
Sets the threshold value that is to be the boundary of the Eye Margin measurement margin.
Example  
To set the threshold value that is to be the Eye Margin measurement margin to 1E–4:
> :SENSe:MEASure:EMARgin:MARGin:THReshold E_4
Compatibility  
Compatible with the MP1632C.

:SENSe:MEASure:EMARgin:MARGin:THReshold?
Response  
<thre> = <CHARACTER RESPONSE DATA>
E_3 to E_12  1E–3 to 1E–12
Function  
Queries the threshold value that is to be the boundary of the Eye Margin measurement margin.
Example  
> :SENSe:MEASure:EMARgin:MARGin:THReshold?
< E_4
Compatibility  
Compatible with the MP1632C.

:SENSe:MEASure:EMARgin:MARGin:RESolution <type>
Parameter  
$type> = <CHARACTER PROGRAM DATA>
FINE   Fine mode
COARse  Coarse mode
Function  
Sets the error rate resolution of the Eye Margin measurement.
Example  
To set the error rate resolution of the Eye Margin measurement to Fine mode:
> :SENSe:MEASure:EMARgin:MARGin:RESolution FINE
Compatibility  
Compatible with the MP1632C.

:SENSe:MEASure:EMARgin:MARGin:RESolution?
Response  
$type> = <CHARACTER RESPONSE DATA>
FINE, COAR
Function  
Queries the error rate resolution of the Eye Margin measurement.
Example  
> :SENSe:MEASure:EMARgin:MARGin:RESolution?
< FINE
Compatibility  
Compatible with the MP1632C.
7.6 Automatic Measurement Commands

**:SENSe:MEASure:EMARgin:MARGin:ASEarch <boolean>[,<mode>]**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;boolean&gt;</td>
<td>= &lt;BOOLEAN PROGRAM DATA&gt;</td>
</tr>
<tr>
<td>ON or 1</td>
<td>Auto Search ON</td>
</tr>
<tr>
<td>OFF or 0</td>
<td>Auto Search OFF</td>
</tr>
<tr>
<td>&lt;mode&gt;</td>
<td>= &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
</tr>
<tr>
<td>1</td>
<td>Coarse</td>
</tr>
<tr>
<td>2</td>
<td>Fine</td>
</tr>
</tbody>
</table>

**Note:**

<mode> can be omitted. Coarse is specified when omitted.

**Function**
Selects whether to perform Auto Search upon start of the Eye Margin measurement.

**Example**
To set Auto Search upon start of the Eye Margin measurement to ON:
> :SENSe:MEASure:EMARgin:MARGin:ASEarch ON

**Compatibility**
Incompatible with existing models.

**:SENSe:MEASure:EMARgin:MARGin:ASEarch?**

<table>
<thead>
<tr>
<th>Response</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;boolean&gt;</td>
<td>= &lt;NR1 NUMERIC RESPONSE DATA&gt;</td>
</tr>
<tr>
<td>1</td>
<td>ON (Coarse)</td>
</tr>
<tr>
<td>0</td>
<td>OFF</td>
</tr>
<tr>
<td>2</td>
<td>ON (Fine)</td>
</tr>
</tbody>
</table>

**Function**
Queries whether to perform Auto Search upon start of the Eye Margin measurement.

**Example**
> :SENSe:MEASure:EMARgin:MARGin:ASEarch?
< 1

** Compatibility**
Incompatible with existing models.

**:SENSe:MEASure:EMARgin:TIME <type>**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;type&gt;</td>
<td>= &lt;CHARACTER PROGRAM DATA&gt;</td>
</tr>
<tr>
<td>DTIMe</td>
<td>Displays the current date and time.</td>
</tr>
<tr>
<td>STARt</td>
<td>Displays the measurement start time.</td>
</tr>
<tr>
<td>ELAPsed</td>
<td>Displays the elapsed time based on the measurement period.</td>
</tr>
</tbody>
</table>

**Function**
Selects the Eye Margin measurement time display type.

**Example**
To set the Eye Margin measurement time display type to measurement start time (Start Time):
> :SENSe:MEASure:EMARgin:TIME STARt

**Compatibility**
Compatible with the MP1632C.
**Chapter 7  SCPI Commands**

:SENSe:MEASure:EMARgin:TIME?

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;type&gt; = &lt;CHARACTER RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>DTIM Displays the current date and time.</td>
</tr>
<tr>
<td></td>
<td>STAR Displays the measurement start time.</td>
</tr>
<tr>
<td></td>
<td>ELAP Displays the elapsed time based on the measurement period.</td>
</tr>
</tbody>
</table>

**Function**
Queries the Eye Margin measurement time display type.

**Example**
> :SENSe:MEASure:EMARgin:TIME?
< STAR

**Compatibility**
Compatible with the MP1632C.

:SENSe:MEASure:EMARgin:DTIMe?

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;year&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2000 to 2036 2000 to 2036 year</td>
</tr>
<tr>
<td></td>
<td>&lt;month&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt;</td>
</tr>
<tr>
<td></td>
<td>1 to 12 January to December</td>
</tr>
<tr>
<td></td>
<td>&lt;day&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt;</td>
</tr>
<tr>
<td></td>
<td>1 to 31 1st to 31st</td>
</tr>
<tr>
<td></td>
<td>&lt;hour&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt;</td>
</tr>
<tr>
<td></td>
<td>0 to 23 0 to 23 hours</td>
</tr>
<tr>
<td></td>
<td>&lt;minute&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt;</td>
</tr>
<tr>
<td></td>
<td>0 to 59 0 to 59 minutes</td>
</tr>
<tr>
<td></td>
<td>&lt;second&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt;</td>
</tr>
<tr>
<td></td>
<td>0 to 59 0 to 59 seconds</td>
</tr>
</tbody>
</table>

**Function**
Queries the current date and time during Eye Margin measurement.

**Example**
> :SENSe:MEASure:EMARgin:DTIMe?
< 2006,4,1,23,59,59

**Compatibility**
Incompatible with existing models.
7.6 Automatic Measurement Commands

**:SENSe:MEASure:EMARgin:STARt?**

<table>
<thead>
<tr>
<th>Response</th>
<th>(&lt;\text{year}&gt;) = \langle \text{NR1 NUMERIC RESPONSE DATA} \rangle</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0, 2000 to 2036  (2000) to (2036) year</td>
</tr>
<tr>
<td></td>
<td>(&lt;\text{month}&gt;) = \langle \text{NR1 NUMERIC RESPONSE DATA} \rangle</td>
</tr>
<tr>
<td></td>
<td>0 to 12  (0) to December</td>
</tr>
<tr>
<td></td>
<td>(&lt;\text{day}&gt;) = \langle \text{NR1 NUMERIC RESPONSE DATA} \rangle</td>
</tr>
<tr>
<td></td>
<td>0 to 31  (0) to 31st</td>
</tr>
<tr>
<td></td>
<td>(&lt;\text{hour}&gt;) = \langle \text{NR1 NUMERIC RESPONSE DATA} \rangle</td>
</tr>
<tr>
<td></td>
<td>0 to 23  (0) to 23 hours</td>
</tr>
<tr>
<td></td>
<td>(&lt;\text{minute}&gt;) = \langle \text{NR1 NUMERIC RESPONSE DATA} \rangle</td>
</tr>
<tr>
<td></td>
<td>0 to 59  (0) to 59 minutes</td>
</tr>
<tr>
<td></td>
<td>(&lt;\text{second}&gt;) = \langle \text{NR1 NUMERIC RESPONSE DATA} \rangle</td>
</tr>
<tr>
<td></td>
<td>0 to 59  (0) to 59 seconds</td>
</tr>
</tbody>
</table>

**Note:**

The following is output when no measurement is performed:

\(0,0,0,0,0,0\)

**Function**

Queries the Eye Margin measurement start time.

**Example**

\(> :\text{SENSe:MEASure:EMARgin:STARt}? \)

\(< 2006,4,1,23,59,59 \)

**Compatibility**

Compatible with the MP1632C.

**:SENSe:MEASure:EMARgin:ELAPsed?**

<table>
<thead>
<tr>
<th>Response</th>
<th>(&lt;\text{day}&gt;) = \langle \text{NR1 NUMERIC RESPONSE DATA} \rangle</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0 to 99  (0) to 99 days</td>
</tr>
<tr>
<td></td>
<td>(&lt;\text{hour}&gt;) = \langle \text{NR1 NUMERIC RESPONSE DATA} \rangle</td>
</tr>
<tr>
<td></td>
<td>0 to 23  (0) to 23 hours</td>
</tr>
<tr>
<td></td>
<td>(&lt;\text{minute}&gt;) = \langle \text{NR1 NUMERIC RESPONSE DATA} \rangle</td>
</tr>
<tr>
<td></td>
<td>0 to 59  (0) to 59 minutes</td>
</tr>
<tr>
<td></td>
<td>(&lt;\text{second}&gt;) = \langle \text{NR1 NUMERIC RESPONSE DATA} \rangle</td>
</tr>
<tr>
<td></td>
<td>0 to 59  (0) to 59 seconds</td>
</tr>
</tbody>
</table>

**Note:**

The following is output when no measurement is performed:

\(0,0,0,0\)

**Function**

Queries the Eye Margin measurement elapsed time.

**Example**

\(> :\text{SENSe:MEASure:EMARgin:ELAPsed}? \)

\(< 99,23,59,59 \)

**Compatibility**

Compatible with the MP1632C.
**Chapter 7  SCPI Commands**

**:CALCulate:DATA:EMARgin? <slot>,<string>[,<unit>]**

Parameter

- `<slot>` = `<CHARACTER PROGRAM DATA>`
  - SLOT1 to SLOT6  Slots No. 1 to No. 6
- `<string>` = `<STRING PROGRAM DATA>`
  - See Table 7.6.2.1-2 for the `<string>` contents.
- `[<unit>]` = `<DECIMAL NUMERIC PROGRAM DATA>`
  - 1 to 4  Mainframe Nos. 1 to 4

When using two or more MT1810A units in serial connection, specify the mainframe number.
Can be omitted. Mainframe No. 1 is specified when omitted.

<table>
<thead>
<tr>
<th>Items</th>
<th><code>&lt;string&gt;</code></th>
<th>Response form</th>
</tr>
</thead>
<tbody>
<tr>
<td>Phase margin</td>
<td>&quot;PHASe&quot;</td>
<td>Form1</td>
</tr>
<tr>
<td>Threshold margin</td>
<td>&quot;THReshold&quot;</td>
<td>Form2</td>
</tr>
<tr>
<td>Period</td>
<td>&quot;PERiod&quot;</td>
<td>Form1</td>
</tr>
</tbody>
</table>

Response

- `<result>` = `<STRING RESPONSE DATA>`
  - For the contents of the `<result>` measurement data see Table 7.6.2.1-3.

<table>
<thead>
<tr>
<th>Form</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Form1</td>
<td>&quot; XXXX&quot;</td>
<td>For 0 to MAX (mUI/ps) displays in the selected unit.</td>
</tr>
<tr>
<td>Phase type</td>
<td>&quot;Failed&quot;</td>
<td>For measurement failure</td>
</tr>
<tr>
<td></td>
<td>&quot;------&quot;</td>
<td>When no data corresponds to the query.</td>
</tr>
<tr>
<td>Form2</td>
<td>&quot; XXXX&quot;</td>
<td>0 to MAX (mVpp)</td>
</tr>
<tr>
<td>Threshold type</td>
<td>&quot;Failed&quot;</td>
<td>For measurement failure</td>
</tr>
<tr>
<td></td>
<td>&quot;------&quot;</td>
<td>When no data corresponds to the query.</td>
</tr>
</tbody>
</table>

Function

Queries the Eye Margin measurement result.

Example

> :CALCulate:DATA:EMARgin? SLOT1, "THReshold"
< "0"

Compatibility

Incompatible with existing models.
### 7.6 Automatic Measurement Commands

**:CALCulate:DATA:EMARgin:STATus?**

<table>
<thead>
<tr>
<th>Response</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;string&gt; = &lt;STRING RESPONSE DATA&gt;</td>
<td>When no alarm exists</td>
</tr>
<tr>
<td>&quot;&quot;</td>
<td>Sync Loss</td>
</tr>
<tr>
<td>&quot;Clock Loss&quot;</td>
<td>Clock Loss</td>
</tr>
<tr>
<td>&quot;CR Unlock&quot;</td>
<td>CR Unlock</td>
</tr>
<tr>
<td>&quot;Out of range&quot;</td>
<td>Out of range</td>
</tr>
<tr>
<td>&quot;Frequency NG&quot;</td>
<td>Frequency NG</td>
</tr>
<tr>
<td>&quot;Illegal Error&quot;</td>
<td>Illegal Error</td>
</tr>
</tbody>
</table>

**Note:**

When two or more alarms have occurred, they are displayed in the following order:
- Illegal Error > Frequency NG > Clock Loss (CR Unlock) > Sync Loss > Out of range

**Function**
Queries the Eye Margin measurement status.

**Example**

```
> :CALCulate:DATA:EMARgin:STATus?
"Sync Loss"
```

**Compatibility**
Incompatible with existing models.
7.6.2.2  File menu setting commands

### Table 7.6.2.2-1  File menu setting commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>[3]</td>
<td>Print</td>
<td>:SYSTem:PRINt:MARGin</td>
</tr>
</tbody>
</table>

#### :SYSTem:MMEMory:MARGin:RECall <file_name>

**Parameter**

- `<file_name>` = `<STRING PROGRAM DATA>`
- `"<drv>:\[<dir>]\<file>"`
- `<drv>` = C, D, E, F
- `<dir>` = `<dir1>\<dir2>\...` (Omitted when root directory)
- `<file>` = `File name`

**Function**
Opens the Eye Margin measurement result data.

**Example**

```plaintext
> :SYSTem:MMEMory:MARGin:RECall "C:\Test\example"
```

**Compatibility**
Incompatible with existing models.

#### :SYSTem:MMEMory:MARGin:STORe <file_name>,<file_type>

**Parameter**

- `<file_name>` = `<STRING PROGRAM DATA>`
- `"<drv>:\[<dir>]\<file>"`
- `<drv>` = C, D, E, F
- `<dir>` = `<dir1>\<dir2>\...` (Omitted when root directory)
- `<file>` = `File name`
- `<file_type>` = `<CHARACTER PROGRAM DATA>`
  - BIN  Binary file
  - CSV  CSV file
  - TXT  Text file

**Function**
Saves the Eye Margin measurement result data.

**Note:**
The settings will not be read from the saved file if the file name is changed.

**Example**

To save the Eye Margin measurement result data by specifying the save destination, file name, and file format:

```plaintext
> :SYSTem:MMEMory:MARGin:STORe "C:\Test\example",TXT
```

**Compatibility**
Incompatible with existing models.

#### :SYSTem:PRINt:MARGin <list>

**Parameter**

- `<list>` = `<CHARACTER PROGRAM DATA>`
  - EMG  Eye Margin Result

**Function**
Prints the Eye Margin measurement result data.

**Example**

```plaintext
> :SYSTem:PRINt:MARGin EMG
```

**Compatibility**
Incompatible with existing models.
7.6.3 Eye Diagram measurement

This section describes the commands for Eye Diagram measurement. The commands related to each tab are described in separate sections below.

Figure 7.6.3-1  Eye Diagram measurement screen
7.6.3.1 Condition tab

![Figure 7.6.3.1-1 Condition tab]

Table 7.6.3.1-1 Condition tab setting commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>:DISPlay:RESult:EDIagram:TABSelect?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:MEASure:EDIagram:DIAGram:SLOT?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:MEASure:EDIagram:MTESt:SLOT?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:MEASure:EDIagram:DIAGram:RESolution?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:MEASure:EDIagram:DIAGram:TRANsition?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:MEASure:EDIagram:DIAGram:TYPE?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:MEASure:EDIagram:DIAGram:ASEarch?</td>
</tr>
</tbody>
</table>
7.6 Automatic Measurement Commands

**:DISPlay:RESult:EDiagram:TABSelect <tab>**

**Parameter**
- `<tab>` = `<CHARACTER PROGRAM DATA>`
  - CONDITION = Condition tab
  - DIAGram = Diagram tab
  - MASKedit = Mask Edit tab

**Function**
Switches the tab display in the Eye Diagram measurement screen.

**Example**
To switch to the Condition tab in the Eye Diagram measurement screen:

> :DISPlay:RESult:EDiagram:TABSelect CONDITION

**Compatibility**
Incompatible with existing models.

**:DISPlay:RESult:EDiagram:TABSelect?**

**Response**
- `<tab>` = `<CHARACTER RESPONSE DATA>`
  - COND, DIAG, MASK

**Function**
Queries the tab currently active in the Eye Diagram measurement screen.

**Example**

> :DISPlay:RESult:EDiagram:TABSelect?

  < COND

**Compatibility**
Incompatible with existing models.

**:SENSe:MEASure:EDiagram:DIAGram:SLOT <slot>,<boolean>[,<unit>]**

**Parameter**
- `<slot>` = `<CHARACTER PROGRAM DATA>`
  - SLOT1 to SLOT6 = Slots No. 1 to No. 6
  - 1 to 6 when using the MP1800A, 1 to 4 when using the MT1810A
- `<boolean>` = `<BOOLEAN PROGRAM DATA>`
  - ON or 1 = Eye Diagram measurement ON
  - OFF or 0 = Eye Diagram measurement OFF
- `<unit>` = `<DECIMAL NUMERIC PROGRAM DATA>`
  - 1 to 4 = Mainframe Nos. 1 to 4

When using two or more MT1810A units in serial connection, specify the mainframe number.

Can be omitted. Mainframe No. 1 is specified when omitted.

**Function**
Selects the Eye Diagram measurement slot.

**Example**
To set Eye Diagram measurement Slot 1 to ON:

> :SENSe:MEASure:EDiagram:DIAGram:SLOT SLOT1,ON

**Compatibility**
Incompatible with existing models.
Chapter 7  SCPI Commands

:SENSe:MEASure:EDIagram:DIAGram:SLOT? <slot>[,<unit>]

Parameter

[slot] = <CHARACTER PROGRAM DATA>
SLOT1 to SLOT6  Slots No. 1 to No. 6
1 to 6 when using the MP1800A, 1 to 4 when using the MT1810A
[unit] = <DECIMAL NUMERIC PROGRAM DATA>
1 to 4  Mainframe Nos. 1 to 4
When using two or more MT1810A units in serial connection, specify the mainframe number.
Can be omitted. Mainframe No. 1 is specified when omitted.

Response

<boolean> = <NR1 NUMERIC RESPONSE DATA>
1  Eye Diagram measurement ON
0  Eye Diagram measurement OFF

Function

Queries whether the specified slot is targeted for the Eye Diagram measurement.

Example

To query whether the slot 1 is targeted for the Eye Diagram measurement:
> :SENSe:MEASure:EDIagram:DIAGram:SLOT? SLOT1
< 1

Compatibility

Incompatible with existing models.

:SENSe:MEASure:EDIagram:MTESt:SLOT <slot>,<boolean>[,<unit>]

Parameter

[slot] = <CHARACTER PROGRAM DATA>
SLOT1 to SLOT6  Slots No. 1 to No. 6
1 to 6 when using the MP1800A, 1 to 4 when using the MT1810A
<boolean> = <BOOLEAN PROGRAM DATA>
ON or 1  Mask Test measurement ON
OFF or 0  Mask Test measurement OFF
[unit] = <DECIMAL NUMERIC PROGRAM DATA>
1 to 4  Mainframe Nos. 1 to 4
When using two or more MT1810A units in serial connection, specify the mainframe number.
Can be omitted. Mainframe No. 1 is specified when omitted.

Function

Selects the Mask Test measurement slot.

Example

To set Mask Test measurement Slot 1 to be ON:
> :SENSe:MEASure:EDIagram:MTESt:SLOT SLOT1,ON

Compatibility

Incompatible with existing models.
7.6 Automatic Measurement Commands

:SENSe:MEASure:EDIagram:MTESt:SLOT? <slot>[,<unit>]

Parameter

- `<slot>` = `<CHARACTER PROGRAM DATA>`
  - SLOT1 to SLOT6 Slots No. 1 to No. 6
  - 1 to 6 when using the MP1800A, 1 to 4 when using the MT1810A
- `<unit>` = `<DECIMAL NUMERIC PROGRAM DATA>`
  - 1 to 4 Mainframe Nos. 1 to 4
    - When using two or more MT1810A units in serial connection, specify the mainframe number.
    - Can be omitted. Mainframe No. 1 is specified when omitted.

Response

- `<boolean>` = `<NR1 NUMERIC RESPONSE DATA>`
  - 1 ON
  - 0 OFF

Function

Queries whether the specified slot is targeted for the Mask Test measurement.

Example

To query whether the slot 1 is targeted for the Mask Test measurement:

```
> :SENSe:MEASure:EDIagram:MTESt:SLOT? SLOT1
< 1
```

Compatibility

Incompatible with existing models.

:SENSe:MEASure:EDIagram:DIAGram:RESolution <type>

Parameter

- `<type>` = `<CHARACTER PROGRAM DATA>`
  - FINE Fine mode
  - COARse Coarse mode

Function

Sets the error boundary point search resolution.

Example

To set the error boundary point search resolution to Fine mode:

```
> :SENSe:MEASure:EDIagram:DIAGram:RESolution FINE
```

Compatibility

Incompatible with existing models.

:SENSe:MEASure:EDIagram:DIAGram:RESolution?

Response

- `<type>` = `<CHARACTER RESPONSE DATA>`
  - FINE, COAR

Function

Queries the error boundary point search resolution.

Example

```
> :SENSe:MEASure:EDIagram:DIAGram:RESolution?
< FINE
```

Compatibility

Incompatible with existing models.
## Chapter 7  SCPI Commands

### :SENSe:MEASure:EDIagram:DIAGram:TRANsition <type>

**Parameter**

<type> = <CHARACTER PROGRAM DATA>

- OFF
- TRANsition
- NONTran

**Function**

Sets Transition/Non Transition error detection.

**Example**

To set to Transition error detection:

```plaintext
> :SENSe:MEASure:EDIagram:DIAGram:TRANsition TRANsition
```

**Compatibility**

Incompatible with existing models.

### :SENSe:MEASure:EDIagram:DIAGram:TRANsition?

**Response**

<type> = <CHARACTER RESPONSE DATA>

- OFF
- TRAN
- NONT

**Function**

Queries the Transition/Non Transition error detection setting state.

**Example**

```plaintext
> :SENSe:MEASure:EDIagram:DIAGram:TRANsition?
< TRAN
```

**Compatibility**

Incompatible with existing models.

### :SENSe:MEASure:EDIagram:DIAGram:TYPE <type>

**Parameter**

<type> = <CHARACTER PROGRAM DATA>

- POINt8  8 points
- POINt16 16 points
- POINt32 32 points
- POINt64 64 points
- POINt132 132 points

**Function**

Sets the number of measurement points during Eye Diagram measurement.

**Example**

To set the number of measurement points during Eye Diagram measurement to 8 points:

```plaintext
> :SENSe:MEASure:EDIagram:DIAGram:TYPE POINt8
```

**Compatibility**

Incompatible with existing models.

### :SENSe:MEASure:EDIagram:DIAGram:TYPE?

**Response**

<type> = <CHARACTER RESPONSE DATA>

- POIN8, POIN16, POIN32, POIN64, POIN132

**Function**

Queries the number of measurement points during Eye Diagram measurement.

**Example**

```plaintext
> :SENSe:MEASure:EDIagram:DIAGram:TYPE?
< POIN8
```

**Compatibility**

Incompatible with existing models.
### 7.6 Automatic Measurement Commands

**:SENSe:MEASure:EDiagram:DIAGram:ASEarch <setting>[,<mode>]**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;setting&gt; = &lt;BOOLEAN PROGRAM DATA&gt;</td>
<td></td>
</tr>
<tr>
<td>ON or 1</td>
<td>Auto Search ON</td>
</tr>
<tr>
<td>OFF or 0</td>
<td>Auto Search OFF</td>
</tr>
<tr>
<td>&lt;mode&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Coarse</td>
</tr>
<tr>
<td>2</td>
<td>Fine</td>
</tr>
</tbody>
</table>

**Note:**

<mode> can be omitted. Coarse is specified when omitted.

**Function**

Selects whether to execute Auto Search when starting the Eye Diagram Measurement.

**Example**

To set Auto Search to On when starting the Eye Diagram Measurement.

> :SENSe:MEASure:EDiagram:DIAGram:ASEarch ON,2

**Compatibility**

Incompatible with existing models.

---

**:SENSe:MEASure:EDiagram:DIAGram:ASEarch?**

<table>
<thead>
<tr>
<th>Response</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;setting&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt;</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>ON (Fine)</td>
</tr>
<tr>
<td>1</td>
<td>ON (Coarse)</td>
</tr>
<tr>
<td>0</td>
<td>OFF</td>
</tr>
</tbody>
</table>

**Function**

Queries whether to execute Auto Search when starting the Eye Diagram Measurement.

**Example**

> :SENSe:MEASure:EDiagram:DIAGram:ASEarch?

< 2

**Compatibility**

Incompatible with existing models.
7.6.3.2 Diagram tab

7.6.3.2.1 Graph display setting commands

Figure 7.6.3.2-1 Diagram tab

Figure 7.6.3.2.1-1 Graph display field
### Table 7.6.3.2.1-1  Graph display setting commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1]</td>
<td>No label (Result display slot setting)</td>
<td>:DISPlay:RESult:EDIagram:SLOT</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:DISPlay:RESult:EDIagram:SLOT?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:DISPlay:RESult:EDIagram:TEMPLate:SELect?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:DISPlay:RESult:EDIagram:TEMPLate:THReshold?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:DISPlay:RESult:EDIagram:SCALe:VOLTage?</td>
</tr>
</tbody>
</table>

:DISPlay:RESult:EDIagram:SLOT <slot>[,<unit>]

**Parameter**

\(<slot> = <CHARACTER PROGRAM DATA>\>

SLOT1 to SLOT6  Slots No. 1 to No. 6
1 to 6 when using the MP1800A, 1 to 4 when using the MT1810A

\(<unit> = <DECIMAL NUMERIC PROGRAM DATA>\>

1 to 4  Mainframe Nos. 1 to 4
When using two or more MT1810A units in serial connection, specify the mainframe number.
Can be omitted. Mainframe No. 1 is specified when omitted.

**Function**

Selects the Eye Diagram result display target slot.

**Example**

To set the Eye Diagram result display target slot to slot No. 1:

\(> :\text{DISPlay}:\text{RESult}:\text{EDIagram}:\text{SLOT \ SLOT1}\)

**Compatibility**

Incompatible with existing models.

:DISPlay:RESult:EDIagram:SLOT? [<unit>]

**Parameter**

\(<\text{unit}> = <\text{DECIMAL NUMERIC PROGRAM DATA}>\>

1 to 4  Mainframe Nos. 1 to 4
When using two or more MT1810A units in serial connection, specify the mainframe number.
Can be omitted. Mainframe No. 1 is specified when omitted.

**Response**

\(<\text{slot> = <CHARACTER RESPONSE DATA}>\>

SLOT1 to SLOT6  Slots No. 1 to No. 6

**Function**

Queries the Eye Diagram result display target slot.

**Example**

\(> :\text{DISPlay}:\text{RESult}:\text{EDIagram}:\text{SLOT?}\)

< SLOT1

**Compatibility**

Incompatible with existing models.
Chapter 7  SCPI Commands

:DISPlay:RESult:EDIagram:TEMPPlate:SESelect <type>

Parameter  
	<type> = <CHARACTER PROGRAM DATA>
	E1  Edit1
	E2  Edit2
	E3  Edit3
	E4  Edit4

Function Sets the mask pattern for performing the Eye Diagram mask test.

Example To set the mask pattern for performing the Eye Diagram mask test to Edit2:
	> :DISPlay:RESult:EDIagram:TEMPPlate:SESelect E2

Compatibility Incompatible with existing models.

:DISPlay:RESult:EDIagram:TEMPPlate:SESelect?

Response <type> = <CHARACTER RESPONSE DATA>
	E1, E2, E3, E4

Function Queries the mask pattern for performing the Eye Diagram mask test.

Example > :DISPlay:RESult:EDIagram:TEMPPlate:SESelect?
	< E2

Compatibility Incompatible with existing models.

:DISPlay:RESult:EDIagram:TEMPPlate:THReshold <thre>

Parameter  
	<thre> = <CHARACTER PROGRAM DATA>
	E_3 to E_12  1E–3 to 1E–12, in single steps

Function Sets the Error Threshold for performing the Eye Diagram mask test.

Example To set the Error Threshold for performing the Eye Diagram mask test to E–3:
	> :DISPlay:RESult:EDIagram:TEMPPlate:THReshold E_3

Compatibility Incompatible with existing models.

:DISPlay:RESult:EDIagram:TEMPPlate:THReshold?

Response <thre> = <CHARACTER RESPONSE DATA>

Function Queries the Error Threshold for performing the Eye Diagram mask test.

Example > :DISPlay:RESult:EDIagram:TEMPPlate:THReshold?
	< E_3

Compatibility Incompatible with existing models.
### 7.6 Automatic Measurement Commands

**:DISPlay:RESult:EDiagram:SCALe:ASCale**

**Function**
Automatically adjusts the Eye Diagram measurement graph scale.

**Example**
> :DISPlay:RESult:EDiagram:SCALe:ASCale

**Compatibility**
Incompatible with existing models.

**:DISPlay:RESult:EDiagram:SCALe:VOLTage <max>,<step>**

**Parameter**

<max> = <DECIMAL NUMERIC PROGRAM DATA>
-3.990 to 4.000 V, in 0.001-V steps

<step> = <DECIMAL NUMERIC PROGRAM DATA>
0.001 to 0.800 V, in 0.001-V steps

The range varies depending on the option status and input interface setting.

**Function**
Sets the maximum value in the direction of the Threshold of the Eye Diagram graph, and the grid resolution.

**Example**
To set the maximum value in the direction of the Threshold of the Eye Diagram graph to 4.000 V, and the grid resolution to 0.800 V:
> :DISPlay:RESult:EDiagram:SCALe:VOLTage 4.000,0.800

**Compatibility**
Incompatible with existing models.

**:DISPlay:RESult:EDiagram:SCALe:VOLTage?**

**Response**

<max> = <NR2 NUMERIC RESPONSE DATA>
-3.990 to 4.000 V

<step> = <NR2 NUMERIC RESPONSE DATA>
0.001 to 0.800 V

**Function**
Queries the maximum value in the direction of the Threshold of the Eye Diagram graph, and the grid resolution.

**Example**
> :DISPlay:RESult:EDiagram:SCALe:VOLTage?
< 4.000,0.800

**Compatibility**
Incompatible with existing models.
Chapter 7  SCPI Commands

:DISPlay:RESult:EDIagram:SCALe:PHASe <max>,<step>

Parameter
- `<max>` = `<DECIMAL NUMERIC PROGRAM DATA>`
  - -990 to 1000 mUI, in 1-mUI steps
- `<step>` = `<DECIMAL NUMERIC PROGRAM DATA>`
  - 1 to 200 mUI, in 1-mUI steps

Function
Sets the maximum value in the direction of the Phase of the Eye Diagram graph, and the grid resolution.

Example
To set the maximum value in the direction of the Phase of the Eye Diagram graph to 1000, and the grid resolution to 10 mUI:
> :DISPlay:RESult:EDIagram:SCALe:PHASe 1000,10

Compatibility
Incompatible with existing models.

:DISPlay:RESult:EDIagram:SCALe:PHASe?

Response
- `<max>` = `<NR2 NUMERIC RESPONSE DATA>`
- `<step>` = `<NR2 NUMERIC RESPONSE DATA>`

Function
Queries the maximum value in the direction of the Phase of the Eye Diagram graph, and the grid resolution.

Example
> :DISPlay:RESult:EDIagram:SCALe:PHASe?
< 1000,10

Compatibility
Incompatible with existing models.
7.6 Automatic Measurement Commands

7.6.3.2.2 Condition setting commands

Figure 7.6.3.2.2-1 Condition setting screen (Actual)

Figure 7.6.3.2.2-2 Condition setting screen (Estimate)
Table 7.6.3.2.2-1  Graph display setting commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>:DISPlay:RESult:EDIagram:DIAGram:TABSelect?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:MEASure:EDiagram:DIAGram:THReshold?</td>
</tr>
<tr>
<td></td>
<td>Meas reset All</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>:DISPlay:RESult:EDiagram:ERATe?</td>
</tr>
<tr>
<td></td>
<td>Display reset All</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>:DISPlay:RESult:EDiagram:EERate?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:DISPlay:RESult:EDiagram:DISPlay:MASK?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:DISPlay:RESult:EDiagram:ESTimate?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:MEASure:EDiagram:DIAGram:BTHReshold?</td>
</tr>
</tbody>
</table>

Table 7.6.3.2.2-2  Graph display Query commands

<table>
<thead>
<tr>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>Query for measurement point</td>
<td>:SENSe:MEASure:EDiagram:DIAGram:POINt?</td>
</tr>
<tr>
<td>Query for measurement result</td>
<td>:CALCulate:DATA:DIAGram?</td>
</tr>
</tbody>
</table>

:DISPlay:RESult:EDiagram:DIAGram:TABSelect <tab>

Parameter

- \(<\text{tab}>\) = <\text{CHARACTER PROGRAM DATA}>
  - \(\text{CONDition}\)  Condition tab
  - \(\text{DETail}\)  Detail tab
  - \(\text{RESult}\)  Result tab

Function

Switches the child tab display on the Diagram tab.

Example

To switch to the Result tab on the Diagram tab:

\(>\) :DISPlay:RESult:EDiagram:DIAGram:TABSelect RESult

Compatibility

Incompatible with existing models.

:DISPlay:RESult:EDiagram:DIAGram:TABSelect?

Response

- \(<\text{tab}>\) = <\text{CHARACTER RESPONSE DATA}>
  - \(\text{COND, DET, RES}\)

Function

Queries the child tab currently active on the Diagram tab.

Example

To query the child tab currently active on the Diagram tab:

\(>\) :DISPlay:RESult:EDiagram:DIAGram:TABSelect?
7.6 Automatic Measurement Commands

---

Compatibility

Incompatible with existing models.

**:SENSe:MEASure:EDIagram:DIAGram:THReshold** `<thre>,<boolean>`

**Parameter**

<table>
<thead>
<tr>
<th>Type</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;type&gt;</code></td>
<td><code>&lt;CHARACTER PROGRAM DATA&gt;</code></td>
</tr>
<tr>
<td><code>&lt;boolean&gt;</code></td>
<td><code>&lt;BOOLEAN PROGRAM DATA&gt;</code></td>
</tr>
<tr>
<td><code>E_3</code> to <code>E_12</code></td>
<td><code>1E–3</code> to <code>1E–12</code>, in single steps</td>
</tr>
<tr>
<td><code>ON</code> or <code>1</code></td>
<td>Selects ON</td>
</tr>
<tr>
<td><code>OFF</code> or <code>0</code></td>
<td>Selects OFF</td>
</tr>
</tbody>
</table>

**Function**

Sets the Eye Diagram measurement target error rate.

**Example**

To set the Eye Diagram measurement target error rate to `1E–12`:

```
> :SENSe:MEASure:EDIagram:DIAGram:THReshold E_12,ON
```

Compatibility

Incompatible with existing models.

---

**:SENSe:MEASure:EDIagram:DIAGram:THReshold?**

**Response**

<table>
<thead>
<tr>
<th>Type</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;type&gt;</code></td>
<td><code>&lt;CHARACTER RESPONSE DATA&gt;</code></td>
</tr>
<tr>
<td><code>E_3</code> to <code>E_12</code></td>
<td><code>1E–3</code> to <code>1E–12</code></td>
</tr>
<tr>
<td><code>NONE</code></td>
<td>When all OFF</td>
</tr>
</tbody>
</table>

The selected threshold is delimited with a comma (,) and returned.

**Function**

Queries the Eye Diagram measurement target error rate.

**Example**

```
> :SENSe:MEASure:EDIagram:DIAGram:THReshold?
< E_3,E_10,E_12
```

Compatibility

Incompatible with existing models.

---

**:DISPlay:RESult:EDIagram:ERATe** `<thre>,<boolean>`

**Parameter**

<table>
<thead>
<tr>
<th>Type</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;type&gt;</code></td>
<td><code>&lt;CHARACTER PROGRAM DATA&gt;</code></td>
</tr>
<tr>
<td><code>&lt;boolean&gt;</code></td>
<td><code>&lt;BOOLEAN PROGRAM DATA&gt;</code></td>
</tr>
<tr>
<td><code>E_3</code> to <code>E_17</code></td>
<td><code>1E–3</code> to <code>1E–17</code>, in single steps</td>
</tr>
<tr>
<td><code>ON</code> or <code>1</code></td>
<td>Display select ON</td>
</tr>
<tr>
<td><code>OFF</code> or <code>0</code></td>
<td>Display select OFF</td>
</tr>
</tbody>
</table>

**Function**

Sets the target error rate of the Eye Diagram measurement result display.

**Example**

To set the target error rate of the Eye Diagram measurement result display to `1E–12`:

```
> :DISPlay:RESult:EDIagram:ERATe E_12,1
```

Compatibility

Incompatible with existing models.
Chapter 7  SCPI Commands

:DISPlay:RESult:EDIagram:ERATe?

Response  
<type> = <CHARACTER RESPONSE DATA>
E_3 to E_17  1E–3 to 1E–17
NONE  When all OFF
The selected threshold is delimited with a comma (,) and returned.

Function  Queries the target error rate of the Eye Diagram measurement result display

Example  
> :DISPlay:RESult:EDIagram:ERATe?
< NONE

Compatibility  Incompatible with existing models.

:SENSe:MEASure:EDIagram:DIAGram:ATHReshold <boolean>

Parameter  
<boolean> = <BOOLEAN PROGRAM DATA>
ON or 1  Sets all ON
OFF or 0  Sets all OFF

Function  Sets all Eye Diagram measurement target error rates to be ON or OFF.

Example  
To set all Eye Diagram measurement target error rates to be ON:
> :SENSe:MEASure:EDIagram:DIAGram:ATHReshold ON

Compatibility  Incompatible with existing models.

:DISPlay:RESult:EDIagram:AERate <boolean>

Parameter  
<boolean> = <BOOLEAN PROGRAM DATA>
ON or 1  Sets all ON
OFF or 0  Sets all OFF

Function  Sets all target error rates of the Eye Diagram measurement result display to be ON or OFF.

Example  
To set all target error rates of the Eye Diagram measurement result display to be ON:
> :DISPlay:RESult:EDIagram:AERate 1

Compatibility  Incompatible with existing models.
### :DISPlay:RESult:EDIagram:EERate <numeric>

**Parameter**

<numeric> = <DECIMAL PROGRAM DATA>
- 13 to 199
- 1E–13 to 1E–199, in single steps

Up to five parameters can be set, delimited with a comma (,).

**Function**

Sets the result display error rate in Estimate measurement from E–13 to E–199.

**Example**

To display the result display error rate in Estimate measurement for 1E–15, 1E–100, 1E–150, 1E–199, and 1E–180:

```
>:DISPlay:RESult:EDIagram:EERate 15,100,150,199,180
```

**Compatibility**

Incompatible with existing models.

### :DISPlay:RESult:EDIagram:EERate?

**Response**

<numeric> = <NR1 NUMERIC RESPONSE DATA>
- 13 to 199
- 1E–13 to 1E–199

Returns five responses, delimited with a comma (,).

**Function**

Queries the result display error rate setting in Estimate measurement from E–13 to E–199.

**Example**

> :DISPlay:RESult:EDIagram:EERate?

```
<15,100,150,199,180
```

**Compatibility**

Incompatible with existing models.

### :DISPlay:RESult:EDIagram:DISPlay:MASK <boolean>

**Parameter**

<boolean> = <BOOLEAN PROGRAM DATA>
- ON or 1  Display
- OFF or 0  Clear

**Function**

Selects between displaying and clearing the mask test result during Eye Diagram measurement.

**Example**

To display the mask test result during Eye Diagram measurement:

```
> :DISPlay:RESult:EDIagram:DISPlay:MASK ON
```

**Compatibility**

Incompatible with existing models.

### :DISPlay:RESult:EDIagram:DISPlay:MASK?

**Response**

<boolean> = <NR1 NUMERIC RESPONSE DATA>
- 1  Display
- 0  Clear

Queries the setting whether to display or clear the mask test result during Eye Diagram measurement.

**Example**

> :DISPlay:RESult:EDIagram:DISPlay:MASK?

```
< 1
```

**Compatibility**

Incompatible with existing models.
Chapter 7  SCPI Commands

:DISPlay:RESult:EDIagram:ADJust

<table>
<thead>
<tr>
<th>Function</th>
<th>Automatically adjusts the mask pattern during Eye Diagram measurement.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example</td>
<td>&gt; :DISPlay:RESult:EDIagram:ADJust</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Incompatible with existing models.</td>
</tr>
</tbody>
</table>

:DISPlay:RESult:EDIagram:ESTimate <type>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;type&gt; = &lt;CHARACTER PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACTual</td>
<td>Actual measurement</td>
</tr>
<tr>
<td>ESTimate</td>
<td>Estimate measurement</td>
</tr>
</tbody>
</table>

**Note:**
The Estimate measurement cannot be selected when the number of measurement points is set to 8.

<table>
<thead>
<tr>
<th>Function</th>
<th>Selects between Actual and Estimate measurement during Eye Diagram measurement.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example</td>
<td>To set Actual measurement during Eye Diagram measurement:</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Incompatible with existing models.</td>
</tr>
</tbody>
</table>

:DISPlay:RESult:EDIagram:ESTimate?

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;type&gt; = &lt;CHARACTER RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACT</td>
<td>Actual</td>
</tr>
<tr>
<td>EST</td>
<td>Estimate</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Function</th>
<th>Queries the Actual/Estimate measurement state during Eye Diagram measurement.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example</td>
<td>&gt; :DISPlay:RESult:EDIagram:ESTimate?</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Incompatible with existing models.</td>
</tr>
</tbody>
</table>
7.6 Automatic Measurement Commands

**:SENSe:MEASure:EDIagram:DIAGram:BTHReshold <upper>,<lower>**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;upper&gt;</td>
<td>&lt;CHARACTER PROGRAM DATA&gt;</td>
</tr>
<tr>
<td>E_3 to E_7</td>
<td>1E–3 to 1E–7, in single step</td>
</tr>
<tr>
<td>&lt;lower&gt;</td>
<td>&lt;CHARACTER PROGRAM DATA&gt;</td>
</tr>
<tr>
<td>E_7 to E_12</td>
<td>1E–7 to E–12, in single step</td>
</tr>
</tbody>
</table>

**Note:**

The difference between the <upper> and <lower> value must be at least 3.

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sets the reference error rate during Eye Diagram Estimate.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Example</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>To set the reference error rate during Eye Diagram Estimate to E–3 and E–12:</td>
<td></td>
</tr>
<tr>
<td>&gt; :SENSe:MEASure:EDIagram:DIAGram:BTHReshold E_3, E_12</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Compatibility</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Incompatible with existing models.</td>
<td></td>
</tr>
</tbody>
</table>

**:SENSe:MEASure:EDIagram:DIAGram:BTHReshold?**

<table>
<thead>
<tr>
<th>Response</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;upper&gt;</td>
<td>&lt;CHARACTER RESPONSE DATA&gt;</td>
</tr>
<tr>
<td>E_3 to E_7</td>
<td>1E–3 to 1E–7</td>
</tr>
<tr>
<td>&lt;lower&gt;</td>
<td>&lt;CHARACTER RESPONSE DATA&gt;</td>
</tr>
<tr>
<td>E_7 to E_12</td>
<td>1E–7 to E–12</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Queries the reference error rate during Eye Diagram Estimate.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Example</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&gt; :SENSe:MEASure:EDIagram:DIAGram:BTHReshold?</td>
<td></td>
</tr>
<tr>
<td>&lt; E_3, E_12</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Compatibility</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Incompatible with existing models.</td>
<td></td>
</tr>
</tbody>
</table>
### Query Valid Points in each Threshold during Eye Diagram Measurement for a Specific Slot

**Command:**

```plaintext
:SENSe:MEASure:EDiagram:DIAGram:POINt? <slot>,<thre>[,<unit>]
```

**Parameter**

- `<slot>` = <CHARACTER PROGRAM DATA>
  
  SLOT1 to SLOT6  
  Slots No. 1 to No. 6

- `<thre>` = <CHARACTER PROGRAM DATA>
  
  E_3 to E_12  
  1E–3 to 1E–12

- `<unit>` = <DECIMAL NUMERIC PROGRAM DATA>
  
  1 to 4  
  Mainframe Nos. 1 to 4

When using two or more MT1810A units in serial connection, specify the mainframe number.

Can be omitted. Mainframe No. 1 is specified when omitted.

**Response**

- `<numeric>` = <NR1 NUMERIC RESPONSE DATA>
  
  1 to 132  
  Returns a valid point.

  0  
  Returns 0 when all points are invalid.

Multiple valid points are delimited with a comma (,) and returned.

**Function**

Queries the valid points in each threshold during the Eye Diagram measurement for a specific slot.

**Example**

To query the valid points in slot No. 1 during measurement with threshold E–3:

```plaintext
> :SENSe:MEASure:EDiagram:DIAGram:POINt? SLOT1,E_3
< 1,2,3,4,132
```

**Compatibility**

Incompatible with existing models.
7.6 Automatic Measurement Commands

**:CALCulate:DATA:DIAGram? <slot>,<string>[,<unit>]**

**Parameter**

- `<slot>` = *<CHARACTER PROGRAM DATA>*
  - SLOT1 to SLOT6 Slots No. 1 to No. 6
- `<string>` = *<STRING PROGRAM DATA>*
- `<result>` Measurement data
  - See Table 7.6.3.2.2-4 for the `<result>` contents.
- `[<unit>]` = *<DECIMAL NUMERIC PROGRAM DATA>*
  - 1 to 4 Mainframe Nos. 1 to 4
  - When using two or more MT1810A units in serial connection, specify the mainframe number.
  - Can be omitted. Mainframe No. 1 is specified when omitted.

**Response**

- `<string>` = *<STRING RESPONSE DATA>*

<table>
<thead>
<tr>
<th>Form</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Form1</td>
<td>&quot;XXXXX,YYYYY&quot;</td>
<td>XXXXX: Phase (mUI)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>XXXXX: Phase (ps) (Displays in the selected unit.)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>YYYYY: Threshold (mV)</td>
</tr>
<tr>
<td></td>
<td>&quot;-----,--.---&quot;</td>
<td>When no data corresponds to the query.</td>
</tr>
</tbody>
</table>

**Function**

Queries the Eye Diagram measurement result of a specific slot and point.

**Table 7.6.3.2.2-4** Eye Diagram measurement item results

<table>
<thead>
<tr>
<th>Items</th>
<th>&lt;Result&gt;</th>
<th>Response Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>Eye Map type</td>
<td>&quot;MAP:E_3:POINt1&quot;,...,&quot;MAP:E_3:POINt132&quot;</td>
<td>Form1</td>
</tr>
<tr>
<td>1E-3</td>
<td>&quot;MAP:E_4:POINt1&quot;,...,&quot;MAP:E_4:POINt132&quot;</td>
<td></td>
</tr>
<tr>
<td>1E-4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1E-11</td>
<td>&quot;MAP:E_11:POINt1&quot;,...,&quot;MAP:E_11:POINt132&quot;</td>
<td></td>
</tr>
<tr>
<td>1E-12</td>
<td>&quot;MAP:E_12:POINt1&quot;,...,&quot;MAP:E_12:POINt132&quot;</td>
<td></td>
</tr>
</tbody>
</table>

**Example**

To query the Eye Diagram measurement result at slot No. 1, E-3, point 1:

> :CALCulate:DATA:DIAGram? SLOT1, "MAP:E_3:POINt1"

" 100, 1.000"

**Compatibility**

Incompatible with existing models.
7.6.3.2.3 Detail setting commands

Figure 7.6.3.2.3-1 Detail Mask Point Test screen

Figure 7.6.3.2.3-2 Detail Mask Line Test screen
Table 7.6.3.2.3-1  Detail setting commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>:SENSe:MEASure:EDIagram:DETail:SLOT?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:MEASure:EDIagram:DETail:MODE?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:MEASure:EDIagram:DETail:STOP</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:MEASure:EDIagram:DETail:STATE?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:MEASure:EDIagram:DETail:LINE:RANGe?</td>
</tr>
<tr>
<td>Marker2</td>
<td></td>
<td>:DISPlay:RESult:EDIagram:DETail:MARKer?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:MEASure:EDIagram:DETail:LINE:ROTation?</td>
</tr>
<tr>
<td>[9]</td>
<td>Query for Point test result</td>
<td>:CALCulate:DATA:EDIagram:DETail:POINt?</td>
</tr>
</tbody>
</table>

**:SENSe:MEASure:EDIagram:DETail:MASK**

Function
Perform automatic mask adjustment during Eye Diagram Detail measurement.

Example
> :SENSe:MEASure:EDIagram:DETail:MASK

Compatibility
Incompatible with existing models.

**:SENSe:MEASure:EDIagram:DETail:SLOT <slot>[,<unit>]**

Parameter

- `<slot>` = `<CHARACTER PROGRAM DATA>`
  - SLOTS1 to SLOT6: Slots No. 1 to No. 6
  - 1 to 6 when using the MP1800A, 1 to 4 when using the MT1810A
- `<unit>` = `<DECIMAL NUMERIC PROGRAM DATA>`
  - 1 to 4: Mainframe Nos. 1 to 4

When using two or more MT1810A units in serial connection, specify the mainframe number.

Can be omitted. Mainframe No. 1 is specified when omitted.

Function
Selects the Mask pass/fail measurement target slot during Eye Diagram measurement.

Example
To set the Mask pass/fail measurement target slot during Eye Diagram measurement to Slot1:
> :SENSe:MEASure:EDIagram:DETail:SLOT SLOT1

Compatibility
Incompatible with existing models.
Chapter 7  SCPI Commands

:SENSe:MEASure:EDIagram:DETail:SLOT? [<unit>]
Response  
<slot> = <CHARACTER RESPONSE DATA>
SLOT1 to SLOT6  Slots No. 1 to No. 6
1 to 6 when using the MP1800A, 1 to 4 when using the MT1810A
[unit] = <DECIMAL NUMERIC PROGRAM DATA>
1 to 4  Mainframe Nos. 1 to 4
When using two or more MT1810A units in serial connection, specify the mainframe number.
Can be omitted. Mainframe No. 1 is specified when omitted.
Function  Queries the Mask pass/fail measurement target slot during Eye Diagram measurement.
Example  
> :SENSe:MEASure:EDIagram:DETail:SLOT?
< SLOT1
Compatibility  
Incompatible with existing models.

:SENSe:MEASure:EDIagram:DETail:MODE <mode>
Parameter  
<mode> = <CHARACTER PROGRAM DATA>
POINt  Point Test
MASK  Mask Line Test
Function  Selects the measurement mode during Eye Diagram Detail measurement.
Example  
To set the measurement mode during Eye Diagram Detail measurement to Point Test mode:
> :SENSe:MEASure:EDIagram:DETail:MODE POINt
Compatibility  
Incompatible with existing models.

:SENSe:MEASure:EDIagram:DETail:MODE?
Response  
<mode> = <CHARACTER RESPONSE DATA>
POIN, MASK
Function  Queries the measurement mode during Eye Diagram Detail measurement.
Example  
> :SENSe:MEASure:EDIagram:DETail:MODE?
< POIN
Compatibility  
Incompatible with existing models.

:SENSe:MEASure:EDIagram:DETail:STARt
Function  Starts the Mask pass/fail judgment during Eye Diagram Detail measurement.
Example  
> :SENSe:MEASure:EDIagram:DETail:STARt
Compatibility  
Incompatible with existing models.
### :SENSe:MEASure:EDIagram:DETail:STOP

<table>
<thead>
<tr>
<th>Function</th>
<th>Stops the Mask pass/fail judgment during Eye Diagram Detail measurement.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example</td>
<td>:SENSe:MEASure:EDIagram:DETail:STOP</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Incompatible with existing models.</td>
</tr>
</tbody>
</table>

#### Example

```
> :SENSe:MEASure:EDIagram:DETail:STOP
```

### :SENSe:MEASure:EDIagram:DETail:STATe?

<table>
<thead>
<tr>
<th>Response</th>
<th>(&lt;\text{numeric}&gt; = \langle\text{NR1 NUMERIC RESPONSE DATA}\rangle)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Start</td>
</tr>
<tr>
<td>0</td>
<td>Stop</td>
</tr>
<tr>
<td>Function</td>
<td>Queries the Mask pass/fail judgment status during Eye Diagram Detail measurement.</td>
</tr>
<tr>
<td>Example</td>
<td>:SENSe:MEASure:EDIagram:DETail:STATe?</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Incompatible with existing models.</td>
</tr>
</tbody>
</table>

```
> :SENSe:MEASure:EDIagram:DETail:STATe?
< 1
```

### :SENSe:MEASure:EDIagram:DETail:LINE:RANGe \(<\text{range}\>\)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>(&lt;\text{range}&gt; = \langle\text{CHARACTER PROGRAM DATA}\rangle)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALL</td>
<td>Whole range</td>
</tr>
<tr>
<td>MARKer</td>
<td>Marked range only</td>
</tr>
<tr>
<td>Function</td>
<td>Selects the measurement range during Eye Diagram Detail Line measurement.</td>
</tr>
<tr>
<td>Example</td>
<td>To set the measurement range during Eye Diagram Detail Line measurement to the marked range:</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Incompatible with existing models.</td>
</tr>
</tbody>
</table>

```
> :SENSe:MEASure:EDIagram:DETail:LINE:RANGe MARKer
```

### :SENSe:MEASure:EDIagram:DETail:LINE:RANGe?

<table>
<thead>
<tr>
<th>Response</th>
<th>(&lt;\text{range}&gt; = \langle\text{CHARACTER RESPONSE DATA}\rangle)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALL, MARK</td>
<td></td>
</tr>
<tr>
<td>Function</td>
<td>Queries the measurement range during Eye Diagram Detail Line measurement.</td>
</tr>
<tr>
<td>Example</td>
<td>:SENSe:MEASure:EDIagram:DETail:LINE:RANGe?</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Incompatible with existing models.</td>
</tr>
</tbody>
</table>

```
> :SENSe:MEASure:EDIagram:DETail:LINE:RANGe?
< MARK
```
### :DISPlay:RESult:EDiagram:DETail:MARKer <marker>,<phase>,<voltage>

**Parameter**
- `<marker>` = `<DECIMAL NUMERIC PROGRAM DATA>`
  - 1: Marker 1
  - 2: Marker 2
- `<phase>` = `<DECIMAL NUMERIC PROGRAM DATA>`
  - 
    -1000 to 1000: –1000 to 1000 mUI, in 1-mUI steps
    (horizontal direction)
- `<voltage>` = `<DECIMAL NUMERIC PROGRAM DATA>`
  - 
    -4000 to 4000: –4000 to 4000 mV, in 1-mV step
    (vertical direction)

**Note:**
- The value of Marker 2 must be equal to or greater than that of marker 1 in the phase direction.
- The marker is moved onto the mask line closest to the set point.

**Function**
Sets the positions for measurement range selection markers 1 and 2 during Eye Diagram Detail Line measurement.

**Example**
To set the position for marker 1 (100 mUI, 2000 mV):

```
>:DISPlay:RESult:EDiagram:DETail:MARKer 1,100,2000
```

**Compatibility**
Incompatible with existing models.

### :DISPlay:RESult:EDiagram:DETail:MARKer? <marker>

**Parameter**
- `<marker>` = `<DECIMAL NUMERIC PROGRAM DATA>`
  - 1: Marker 1
  - 2: Marker 2

**Response**
- `<phase>,<voltage>`
  - `<phase>` = `<NR2 NUMERIC RESPONSE DATA>`
    - 
      -1000 to 1000: –1000 to 1000 mUI (horizontal direction)
      XXXXX: XXXXX ps
  - `<voltage>` = `<NR2 NUMERIC RESPONSE DATA>`
    - 
      -4000 to 4000: –4000 to 4000 mV (vertical direction)

**Note:**
- The converted value from mUI

**Function**
Queries the positions for the measurement range selection markers 1 and 2 during Eye Diagram Detail Line measurement.

**Example**
To query the set value for marker 1 (1,000 V, 100 mUI):

```
>:DISPlay:RESult:EDiagram:DETail:MARKer? 1
< 100,1000
```

**Compatibility**
Incompatible with existing models.
### 7.6 Automatic Measurement Commands

#### :SENSe:MEASure:EDIagram:DETail:LINE:ROTation <type>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;type&gt; = &lt;CHARACTER PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>LEFT</td>
<td>Counterclockwise</td>
</tr>
<tr>
<td>RIGHT</td>
<td>Clockwise</td>
</tr>
</tbody>
</table>

Selects the Mask pass/fail judgment direction during Eye Diagram Detail Line measurement.

**Example**
To set the Mask pass/fail judgment direction during Eye Diagram Detail Line measurement to be clockwise.

> :SENSe:MEASure:EDIagram:DETail:LINE:ROTation RIGHT

**Compatibility**
Incompatible with existing models.

#### :SENSe:MEASure:EDIagram:DETail:LINE:ROTation?

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;type&gt; = &lt;CHARACTER RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>LEFT, RIGHT</td>
<td></td>
</tr>
</tbody>
</table>

Queries the Mask pass/fail judgment direction during Eye Diagram Detail Line measurement.

**Example**

> :SENSe:MEASure:EDIagram:DETail:LINE:ROTation?

< RIGHT

**Compatibility**
Incompatible with existing models.

#### :CALCulate:DATA:EDIagram:DETail:PNUMber?

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;numeric&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 to 20000</td>
<td>Number of Detail measurement point</td>
</tr>
</tbody>
</table>

Queries the number of measurement points during Detail measurement.

**Example**

> :CALCulate:DATA:EDIagram:DETail:PNUMber?

< 1000

**Compatibility**
Incompatible with existing models.
Chapter 7  SCPI Commands

:CALCulate:DATA:EDIagram:DETail:POINt? <string>

Parameter

<string> = <STRING PROGRAM DATA>

<Result> = Measurement data

See Table 7.6.3.2.3-3 for the <result> contents.

Response

<string> = <STRING RESPONSE DATA>

Table 7.6.3.2.3-2  Response format

<table>
<thead>
<tr>
<th>Form</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Form1</td>
<td>&quot;XXXXX,YYYYY,ZZ&quot;</td>
<td>XXXXX: Phase (mUI) XXXXX: Phase (ps) Note: Displays in the specified units. YY.YYY: Threshold (mV) ZZ: OK, NG</td>
</tr>
<tr>
<td>Eye Map type</td>
<td>&quot;-----,--.---,--&quot;</td>
<td>When no data corresponds to the query.</td>
</tr>
</tbody>
</table>

Function

Queries the measurement results of the Eye Diagram Detail Point Test.

Table 7.6.3.2.3-3  Eye Diagram Detail Point Test setting items <result>

<table>
<thead>
<tr>
<th>Items</th>
<th>&lt;result&gt;</th>
<th>Response Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>Judgment type</td>
<td>&quot;JDG:POINt1&quot;</td>
<td>Form1</td>
</tr>
<tr>
<td>Point1</td>
<td>&quot;JDG:POINt2&quot;</td>
<td></td>
</tr>
<tr>
<td>Point2</td>
<td>&quot;JDG:POINt31&quot;</td>
<td></td>
</tr>
<tr>
<td>Point31</td>
<td>&quot;JDG:POINt32&quot;</td>
<td></td>
</tr>
<tr>
<td>Point32</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Queries the measurement result of the Eye Diagram Detail Point Test at the measurement point.

Example

To query the measurement result of the Eye Diagram Detail Point Test at measurement point 1:

> :CALCulate:DATA:EDIagram:DETail:POINt? "JDG:POINt1"
< " 0, 4000,OK"

Compatibility

Incompatible with existing models.
7.6 Automatic Measurement Commands

:CALCulate:DATA:EDIagram:DETail:LINE? <string>

Parameter

<string> = <STRING PROGRAM DATA>

<string> = <STRING RESPONSE DATA>

Measurement data

See Table 7.6.3.2.3-5 for the <result> contents.

Response

Table 7.6.3.2.3-4 Response format

<table>
<thead>
<tr>
<th>Form</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Form1</td>
<td>&quot;XXXXX,YYYYY,ZZ&quot;</td>
<td>XXXXX: Phase (mUI) XXXXX: Phase (ps)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Note: Displays in the specified units.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>YYYYY: Threshold (mV)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ZZ: OK, NG</td>
</tr>
<tr>
<td></td>
<td>&quot;-----,--.---,--&quot;</td>
<td>When no data corresponds to the query.</td>
</tr>
</tbody>
</table>

Function

Queries the measurement result of the Eye Diagram Detail Mask Line Test.

Table 7.6.3.2.3-5 Eye Diagram Detail Mask Line Test measurement items <result>

<table>
<thead>
<tr>
<th>Items</th>
<th>&lt;Result&gt;</th>
<th>Response Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>Judgment type</td>
<td></td>
<td>Form1</td>
</tr>
<tr>
<td>Point1</td>
<td>&quot;JDG:POINt1&quot;</td>
<td></td>
</tr>
<tr>
<td>Point2</td>
<td>&quot;JDG:POINt2&quot;</td>
<td></td>
</tr>
<tr>
<td>Point19999</td>
<td>&quot;JDG:POINt19999&quot;</td>
<td></td>
</tr>
<tr>
<td>Point20000</td>
<td>&quot;JDG:POINt20000&quot;</td>
<td></td>
</tr>
</tbody>
</table>

Example

To query the measurement result of the Eye Diagram Detail Mask Line Test at measurement point 1:

>:CALCulate:DATA:EDIagram:DETail:LINE? "JDG:POINt1"

<" 0, 4000,OK"

Compatibility

Incompatible with existing models.
7.6.3.2.4 Result query command

Table 7.6.3.2.4-1 Result query command

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
</table>

Figure 7.6.3.2.4-1 Result screen

No. Setting Items Commands
:CALCulate:DATA:EDIagram:MASK? <string>[,<unit>]

Parameter  
<string> = <STRING PROGRAM DATA>  
Measurement data  
See Table 7.6.3.2.4-3: for the <result> contents.  
[unit] = <DECIMAL NUMERIC PROGRAM DATA>  
1 to 4  
Mainframe Nos. 1 to 4  
When using two or more MT1810A units in serial connection, specify the mainframe number.  
Can be omitted. Mainframe No. 1 is specified when omitted.

Response  
<string> = <STRING RESPONSE DATA>

Table 7.6.3.2.4-2  Response format

<table>
<thead>
<tr>
<th>Form</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
</table>
| Form1  | “XX,Y,Y,Y…Y”     | XX: OK, NG  
|        |                   | Y: 1 to 32 (NG mask point)                       |
|        |                   | Note: Multiple NGs are delimited with a comma (,) and returned for the corresponding number of NGs. |
|        | “--.--”           | When no data corresponds to the query.           |

Function  
Queries the measurement result of the Eye Diagram Mask Test.

Table 7.6.3.2.4-3  Eye Diagram Result setting items and results

<table>
<thead>
<tr>
<th>Items</th>
<th>&lt;Result&gt;</th>
<th>Response Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>Judgment type</td>
<td>Slot1</td>
<td>&quot;SLT1&quot;</td>
</tr>
<tr>
<td></td>
<td>Slot2</td>
<td>&quot;SLT2&quot;</td>
</tr>
<tr>
<td></td>
<td>Slot6</td>
<td>&quot;SLT6&quot;</td>
</tr>
</tbody>
</table>

Example  
To query the Mask Test result of Mainframe No.3 Slot1:  
>:CALCulate:DATA:EDIagram:MASK? "SLT1"3  
"NG,1,2,3,4,32"  
"OK"

Compatibility  
Incompatible with existing models.
Chapter 7  SCPI Commands

7.6.3.2.5  Diagram tab (Status) setting commands

![Diagram tab Start/Stop buttons and Status field]

**Figure 7.6.3.2.5-1**  Diagram tab Start/Stop buttons and Status field

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>[4]</td>
<td>No label (Setting of time display)</td>
<td>:SENSe:MEASure:EDiagram:TIME</td>
</tr>
</tbody>
</table>

**:SENSe:MEASure:EDiagram:DIAGram:START**

**Function**  Starts the Eye Diagram measurement.

**Example**  
> :SENSe:MEASure:EDiagram:DIAGram:START

**Compatibility**  Incompatible with existing models.
7.6 Automatic Measurement Commands

**:SENSe:MEASure:EDiagram:DIAGram:STOP**

**Function**
Stops the Eye Diagram measurement.

**Example**
> :SENSe:MEASure:EDiagram:DIAGram:STOP

**Compatibility**
Incompatible with existing models.

**:SENSe:MEASure:EDiagram:STATe?**

**Response**

<table>
<thead>
<tr>
<th>&lt;numeric&gt;</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Measurement has been started.</td>
</tr>
<tr>
<td>0</td>
<td>Measurement has been stopped</td>
</tr>
<tr>
<td>-1</td>
<td>Measurement has failed.</td>
</tr>
</tbody>
</table>

**Function**
Queries the Eye Diagram measurement status.

**Example**
> :SENSe:MEASure:EDiagram:STATe?
< 1

**Compatibility**
Incompatible with existing models.

**:SENSe:MEASure:EDiagram:TIME <type>**

**Parameter**

<table>
<thead>
<tr>
<th>&lt;type&gt;</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DTIMe</td>
<td>Displays the current date and time.</td>
</tr>
<tr>
<td>STARt</td>
<td>Displays the measurement start time.</td>
</tr>
<tr>
<td>ELAPsed</td>
<td>Displays the elapsed time based on the measurement period.</td>
</tr>
</tbody>
</table>

**Function**
Selects the Eye Diagram measurement time display type.

**Example**
To set the Eye Diagram measurement time display type to measurement start time (Start Time):
> :SENSe:MEASure:EDiagram:TIME STARt

**Compatibility**
Incompatible with existing models.

**:SENSe:MEASure:EDiagram:TIME?**

**Response**

<table>
<thead>
<tr>
<th>&lt;type&gt;</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DTIM</td>
<td>Displays the current date and time.</td>
</tr>
<tr>
<td>STAR</td>
<td>Displays the measurement start time.</td>
</tr>
<tr>
<td>ELAP</td>
<td>Displays the elapsed time based on the measurement period.</td>
</tr>
</tbody>
</table>

**Function**
Queries the Eye Diagram measurement time display type.

**Example**
> :SENSe:MEASure:EDiagram:TIME?
< STAR

**Compatibility**
Incompatible with existing models.
Chapter 7  SCPI Commands

:SENSe:MEASure:EDIagram:DTIMe?

Response  

<year> = <NR1 NUMERIC RESPONSE DATA>  
2000 to 2036  2000 to 2036 year  
2000 to 2036 year  

Reference  

<month> = <NR1 NUMERIC RESPONSE DATA>  
1 to 12  January to December  
1 st to 12 th  

Reference  

<day> = <NR1 NUMERIC RESPONSE DATA>  
1 to 31  1 st to 31 st  
1 to 31  

Reference  

<hour> = <NR1 NUMERIC RESPONSE DATA>  
0 to 23  0 to 23 hours  
0 to 23  

Reference  

<minute> = <NR1 NUMERIC RESPONSE DATA>  
0 to 59  0 to 59 minutes  
0 to 59  

Reference  

<second> = <NR1 NUMERIC RESPONSE DATA>  
0 to 59  0 to 59 seconds  
0 to 59  

Function  

Queries the current date and time during Eye Diagram measurement.

Example  

> :SENSe:MEASure:EDIagram:DTIMe?  
< 2006,4,1,23,59,59

Compatibility  

Incompatible with existing models.

:SENSe:MEASure:EDIagram:STARt?

Response  

<year> = <NR1 NUMERIC RESPONSE DATA>  
0, 2000 to 2036  2000 to 2036 year  

Reference  

<month> = <NR1 NUMERIC RESPONSE DATA>  
0, 1 to 12  January to December  
0, 1 st to 12 th  

Reference  

<day> = <NR1 NUMERIC RESPONSE DATA>  
0, 1 to 31  1 st to 31 st  
0, 1 to 31  

Reference  

<hour> = <NR1 NUMERIC RESPONSE DATA>  
0 to 23  0 to 23 hours  
0 to 23  

Reference  

<minute> = <NR1 NUMERIC RESPONSE DATA>  
0 to 59  0 to 59 minutes  
0 to 59  

Reference  

<second> = <NR1 NUMERIC RESPONSE DATA>  
0 to 59  0 to 59 seconds  
0 to 59  

Note:  

The following is output when no measurement is performed:  
0,0,0,0,0,0

Function  

Queries the Eye Diagram measurement start time (Start Time).

Example  

> :SENSe:MEASure:EDIagram:STARt?  
< 2006,4,1,23,59,59

Compatibility  

Incompatible with existing models.
### :SENSe:MEASure:EDiagram:ELAPsed?

**Response**

<table>
<thead>
<tr>
<th>&lt;day&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>0, 1 to 99</td>
</tr>
<tr>
<td>&lt;hour&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt;</td>
</tr>
<tr>
<td>0 to 23</td>
</tr>
<tr>
<td>&lt;minute&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt;</td>
</tr>
<tr>
<td>0 to 59</td>
</tr>
<tr>
<td>&lt;second&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt;</td>
</tr>
<tr>
<td>0 to 59</td>
</tr>
</tbody>
</table>

**Note:**

The following is output when no measurement is performed:

0,0,0,0

**Function**

Queries the Eye Diagram measurement elapsed time (Elapsed Time).

**Example**

> :SENSe:MEASure:EDiagram:ELAPsed?

< 99,23,59,59

**Compatibility**

Incompatible with existing models.

### :CALCulate:DATA:EDiagram:STATus?

**Response**

<table>
<thead>
<tr>
<th>&lt;string&gt; = &lt;STRING RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;&quot;</td>
</tr>
<tr>
<td>&quot;Sync Loss&quot;</td>
</tr>
<tr>
<td>&quot;Clock Loss&quot;</td>
</tr>
<tr>
<td>&quot;CR Unlock&quot;</td>
</tr>
<tr>
<td>&quot;Out of range&quot;</td>
</tr>
<tr>
<td>&quot;Frequency NG&quot;</td>
</tr>
<tr>
<td>&quot;Illegal Error&quot;</td>
</tr>
</tbody>
</table>

**Note:**

When two or more alarms have occurred, they are displayed in the following order:

Illegal Error > Frequency NG > Clock Loss (CR Unlock) > Sync Loss > Out of range

**Function**

Queries the status condition during Eye Diagram measurement.

**Example**

> :CALCulate:DATA:EDiagram:STATus?

< "Sync Loss"

**Compatibility**

Incompatible with existing models.
7.6.3.2.6 Mask Edit tab

![Figure 7.6.3.2.6-1 Mask Edit tab](image)

Table 7.6.3.2.6-1 Mask Edit tab setting commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>:DISPlay:RESult:EDIagram:TEMPlate:PATTern?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:DISPlay:RESult:EDIagram:TEMPlate:EDIT?</td>
</tr>
</tbody>
</table>
### :DISPlay:RESult:EDiagram:TEMPlate:PATTern <type>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;type&gt; = &lt;CHARACTER PROGRAM DATA&gt;</td>
<td>Sets the pattern to be edited during Eye Diagram measurement.</td>
</tr>
<tr>
<td>E1</td>
<td>Edit1</td>
</tr>
<tr>
<td>E2</td>
<td>Edit2</td>
</tr>
<tr>
<td>E3</td>
<td>Edit3</td>
</tr>
<tr>
<td>E4</td>
<td>Edit4</td>
</tr>
</tbody>
</table>

**Function**

Sets the pattern to be edited during Eye Diagram measurement.

**Example**

To set the pattern to be edited during Eye Diagram measurement to Edit4:

```
> :DISPlay:RESult:EDIagram:TEMPlate:PATTern E4
```

**Compatibility**

Incompatible with existing models.

### :DISPlay:RESult:EDiagram:TEMPlate:PATTern?

<table>
<thead>
<tr>
<th>Response</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;type&gt; = &lt;CHARACTER RESPONSE DATA&gt;</td>
<td>Queries the pattern that was edited during Eye Diagram measurement.</td>
</tr>
<tr>
<td>E1, E2, E3, E4</td>
<td></td>
</tr>
</tbody>
</table>

**Function**

Queries the pattern that was edited during Eye Diagram measurement.

**Example**

````
> :DISPlay:RESult:EDIagram:TEMPlate:PATTern?
< E4
```

**Compatibility**

Incompatible with existing models.

### :DISPlay:RESult:EDiagram:TEMPlate:ASCale

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Executes automatic adjustment of the Mask Edit measurement graph scale.</td>
<td></td>
</tr>
</tbody>
</table>

**Example**

````
> :DISPlay:RESult:EDIagram:TEMPlate:ASCale
```

**Compatibility**

Incompatible with existing models.

### :DISPlay:RESult:EDiagram:TEMPlate:SCALe<max>,<step>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;max&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
<td>Sets the maximum value and the grid resolution in the direction of the Threshold of the Mask Edit graph.</td>
</tr>
<tr>
<td>–3.990 to 4.000</td>
<td>–3.990 to 4.000 V, in 0.001 V steps</td>
</tr>
<tr>
<td>&lt;step&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
<td>1 to 800</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sets the maximum value and the grid resolution in the direction of the Threshold of the Mask Edit graph.</td>
<td></td>
</tr>
</tbody>
</table>

**Example**

To set the maximum value and the grid resolution in the direction of the Threshold of the Mask Edit graph to 4.000 V and 800 mV, respectively:

````
> :DISPlay:RESult:EDIagram:TEMPlate:SCALe 4.000,800
```

**Compatibility**

Incompatible with existing models.
### :DISPlay:RESult:EDIagram:TEMPLate:SCALe?

**Response**

\[
<\text{max}> = <\text{NR2 NUMERIC RESPONSE DATA}>
\]

\[
<\text{step}> = <\text{NR2 NUMERIC RESPONSE DATA}>
\]

**Function**

Queries the maximum value and the grid resolution in the direction of the Threshold of the Mask Edit graph.

**Example**

> :DISPlay:RESult:EDIagram:TEMPLate:SCALe?

\(< \text{max}, \text{step} >
\]

**Compatibility**

Incompatible with existing models.

### :DISPlay:RESult:EDIagram:TEMPLate:PSCcale <max>,<step>

**Parameter**

\[
<\text{max}> = <\text{DECIMAL NUMERIC PROGRAM DATA}>
\]

\(-990 \text{ to } 1000\)

\[
<\text{step}> = <\text{DECIMAL NUMERIC PROGRAM DATA}>
\]

\(1 \text{ to } 200\)

**Function**

Sets the maximum value and the grid resolution in the direction of the Phase of the Mask Edit graph.

**Example**

To set the maximum value and the grid resolution in the direction of the Phase of the Mask Edit graph to 1000 and 10 mUI, respectively:

> :DISPlay:RESult:EDIagram:TEMPLate:PSCcale 1000,10

**Compatibility**

Incompatible with existing models.

### :DISPlay:RESult:EDIagram:TEMPLate:SAMPle <string>

**Parameter**

\[
<\text{string}> = <\text{STRING PROGRAM DATA}>
\]

\("<\text{drv}>:\text{[<dir>]}<\text{file}>"
\]

<drv> = C, D, E, F

<dir> = <dir1>\text{[<dir2>]}\text{...} (Omitted when root directory)

<file> = File name

**Function**

Expands the selected sample pattern to the Edit Pattern.

**Example**

> :DISPlay:RESult:EDIagram:TEMPLate:SAMPle

\("C:\text{...}\text{sample.MSK}"

**Compatibility**

Incompatible with existing models.
7.6 Automatic Measurement Commands

:DISPlay:RESult:EDiagram:TEMPLate:PNUMber?

Response

<numeric> = <NR1 NUMERIC RESPONSE DATA>
0 to 32  0 to 32 points, in 1-point steps

Function
Queries the number of Mask points.

Example
> :DISPlay:RESult:EDiagram:TEMPLate:PNUMber?
< 0

Compatibility
Incompatible with existing models.

:DISPlay:RESult:EDiagram:TEMPLate:EDIT <point>, <phase>, <voltage>

Parameter

<point> = <DECIMAL NUMERIC PROGRAM DATA>
1 to 32  1 to 32 points, in 1-point steps

<phase> = <DECIMAL NUMERIC PROGRAM DATA>
–1000 to 1000  –1000 to 1000 mUI, in 1-mUI steps

<voltage> = <DECIMAL NUMERIC PROGRAM DATA>
–4.000 to 4.000  –4.000 to 4.000 V, in 0.001-V steps

Function
Sets a Mask point position.

Example
To set a Mask point position (1 point, 1000 mUI, 4.000 V):
> :DISPlay:RESult:EDiagram:TEMPLate:EDIT 1,1000,4.000

Compatibility
Incompatible with existing models.

:DISPlay:RESult:EDiagram:TEMPLate:EDIT? <point>

Parameter

<point> = <DECIMAL NUMERIC PROGRAM DATA>
1 to 32  1 to 32 points

Response

<phase> = <NR1 NUMERIC RESPONSE DATA>
<voltage> = <NR1 NUMERIC RESPONSE DATA>

Function
Queries a Mask point position.

Example
To query the position of point 1:
> :DISPlay:RESult:EDiagram:TEMPLate:EDIT? 1
< 1000,4.000

Compatibility
Incompatible with existing models.

:DISPlay:RESult:EDiagram:TEMPLate:PLOT <phase>, <voltage>

Parameter

<phase> = <DECIMAL NUMERIC PROGRAM DATA>
–1000 to 1000  –1000 to 1000 mUI, in 1-mUI steps

<voltage> = <DECIMAL NUMERIC PROGRAM DATA>
–4000 to 4000  –4000 to 4000 mV, in 1-mV steps

Function
Sets a Mask point position.

Example
To set a Mask point position to 1000 mUI (phase), 4000 mV (voltage):
> :DISPlay:RESult:EDiagram:TEMPLate:PLOT 1000,4000

Compatibility
Incompatible with existing models.
### :DISPlay:RESult:EDIagram:TEMPLate:DELeTe <pos>

**Parameter**

\(<\text{pos}> = \langle\text{DECIMAL NUMERIC PROGRAM DATA}\rangle\)

1 to 32  
1 to 32 points, in 1-point steps

**Function**

Deletes the selected Mask point.

**Example**

To delete Mask point 3:

> :DISPlay:RESult:EDIagram:TEMPLate:DELeTe 3

**Compatibility**

Incompatible with existing models.

### :DISPlay:RESult:EDIagram:TEMPLate:CLEar

**Function**

Clears the selected Mask pattern.

**Example**

> :DISPlay:RESult:EDIagram:TEMPLate:CLEar

**Compatibility**

Incompatible with existing models.

### :DISPlay:RESult:EDIagram:TEMPLate:VERTical

**Function**

Plots a Mask pattern onto the vertically-symmetric position.

**Example**

> :DISPlay:RESult:EDIagram:TEMPLate:VERTical

**Compatibility**

Incompatible with existing models.

### :DISPlay:RESult:EDIagram:TEMPLate:HORizontal

**Function**

Plots a Mask pattern onto the horizontally-symmetric position.

**Example**

> :DISPlay:RESult:EDIagram:TEMPLate:HORizontal

**Compatibility**

Incompatible with existing models.
7.6

7.6.3.2.7

Automatic Measurement Commands

File menu setting commands
Table 7.6.3.2.7-1 File menu setting commands
No.
[1]
[2]
[3]

Setting Items
Open
Save
Print

Commands
:SYSTem:MMEMory:DIAGram:RECall
:SYSTem:MMEMory:DIAGram:STORe
:SYSTem:PRINt:DIAGram

:SYSTem:MMEMory:DIAGram:RECall <file_name>,<file_type>
Parameter

Function
Example
Compatibility

<file_name> = <STRING PROGRAM DATA>
"<drv>:\[<dir>]<file>"
<drv> = C, D, E, F
<dir> = <dir1>\<dir2>\… (Omitted when root directory)
<file> = File name
<file_type> = <CHARACTER PROGRAM DATA>
BIN
Binary file
TXT
Text (UMP) file
Opens the Eye Diagram measurement result data.
> :SYSTem:MMEMory:DIAGram:RECall "C:\Test\example",TXT
Incompatible with existing models.

7
SCPI Commands

7-433


**Chapter 7  SCPI Commands**

**:SYSTem:MMEMory:DIAGram:STORe <file_name>,<data_type>,<file_type>**

Parameter

- `<file_name>` = `<STRING PROGRAM DATA>`
  - "<drv>:\[<dir>]<file>"
  - `<drv>` = C, D, E, F
  - `<dir>` = `<dir1>\<dir2>\...` (Omitted when root directory)
  - `<file>` = File name
- `<data_type>` = `<CHARACTER PROGRAM DATA>`
  - EDG = Eye Diagram Result
  - EMP = Eye Mask Point Result
  - EMD = Eye Mask Detail Result
  - UMP = Eye Mask Template File
- `<file_type>` = `<CHARACTER PROGRAM DATA>`
  - BIN = Binary file
  - CSV = CSV file
  - TXT = Text file

When BIN is selected for `<file_type>`, the Eye Diagram measurement settings and results are saved, regardless of the setting of `<data_type>`. When UMP is selected for `<data_type>`, select TXT for `<file_type>`.

Function

Saves the Eye Diagram measurement result data.

**Note:**

The settings will not be read from the saved file if the file name is changed.

Example

To save the Eye Mask Detail measurement results of the Eye Diagram measurement, by specifying the file format, file name, and save destination:

```
> :SYSTem:MMEMory:DIAGram:STORe "C:\Test\example.txt",EMD,TXT
```

Compatibility

Incompatible with existing models.

---

**:SYSTem:PRINT:DIAGram <list>**

Parameter

- `<list>` = `<CHARACTER PROGRAM DATA>`
  - EDG = Eye Diagram Result
  - EMP = Eye Mask Point Result
  - EMD = Eye Mask Detail Result

Function

Prints the Eye Diagram measurement result data.

Example

Prints the Eye Diagram measurement result data of the Eye Diagram measurement.

```
> :SYSTem:PRINT:DIAGram EDG
```

Compatibility

Incompatible with existing models.
7.6 Automatic Measurement Commands

7.6.4 Q measurement

This section describes the commands for Q measurement.

Figure 7.6.4.1-1 Q measurement screen

7.6.4.1 Graph display setting commands

Figure 7.6.4.1-1 Graph display screen (Threshold vs Q tab)
### Figure 7.6.4.1-2  Graph display screen (Phase vs Q tab)

#### Table 7.6.4.1-1  Graph display setting commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>:DISPlay:RESult:QANalysis:MODE?</td>
</tr>
<tr>
<td></td>
<td>Min</td>
<td>:DISPlay:RESult:QANalysis:SCALE:VERTical?</td>
</tr>
<tr>
<td></td>
<td>Step</td>
<td>:DISPlay:RESult:QANalysis:SCALE:HORizontal?</td>
</tr>
<tr>
<td></td>
<td>Display graph selection</td>
<td>:DISPlay:RESult:QANalysis:THReshold:ITEM?</td>
</tr>
<tr>
<td></td>
<td>Current No.</td>
<td>:DISPlay:RESult:QANalysis:THReshold:MNUMber?</td>
</tr>
<tr>
<td></td>
<td>Display graph selection</td>
<td>:DISPlay:RESult:QANalysis:PHASe:ITEM?</td>
</tr>
<tr>
<td></td>
<td>Current No.</td>
<td>:DISPlay:RESult:QANalysis:PHASe:MNUMber?</td>
</tr>
</tbody>
</table>
7.6 Automatic Measurement Commands

**:DISPlay:RESult:QANalysis:MODE <mode>**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;mode&gt; = &lt;CHARACTER PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>THREshold</td>
<td>Vth vs. Q measurement</td>
</tr>
<tr>
<td>PHASe</td>
<td>Phase vs. Q measurement</td>
</tr>
</tbody>
</table>

**Function**
Sets the graph display item from Vth vs. Q measurement or Phase vs. Q measurement.
The settings for the display scale and the marker correspond to this setting.

**Example**
To set to Vth vs. Q measurement:
> :DISPlay:RESult:QANalysis:MODE THREshold

**Compatibility**
Incompatible with existing models.

**:DISPlay:RESult:QANalysis:MODE?**

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;mode&gt; = &lt;CHARACTER RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>THRE, PHAS</td>
<td></td>
</tr>
</tbody>
</table>

**Function**
Queries the item to be displayed in the graph during Vth vs. Q measurement.

**Example**
> :DISPlay:RESult:QANalysis:MODE?
< THRE

**Compatibility**
Incompatible with existing models.

**:DISPlay:RESult:QANalysis:SCALe:ASCale**

**Function**
Automatically adjusts the graph display in Q measurement.

**Example**
> :DISPlay:RESult:QANalysis:SCALe:ASCale

**Compatibility**
Incompatible with existing models.
### Chapter 7  SCPI Commands

*DISPlay:RESult:QAnalys:SCALe:VERTical* `<max>,[<step>]`

Parameter `<max>,[<step>]` = `<DECIMAL NUMERIC PROGRAM DATA>`

**Table 7.6.4.1-2 Parameters for Threshold vs. Q tab**

<table>
<thead>
<tr>
<th>Format</th>
<th><code>&lt;max&gt;</code></th>
<th><code>&lt;step&gt;</code></th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vth vs BER (Y = Log)</td>
<td>2 to 7 (E–2 to E–7), in single steps</td>
<td>Inputs the minimum value.</td>
<td>A difference between <code>&lt;max&gt;</code> and <code>&lt;step&gt;</code> must be 7 or greater even number.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>8 to 14 (E–8 to E–14), in single steps</td>
<td></td>
</tr>
<tr>
<td>Vth vs BER (Y = Log (−Ln))</td>
<td>2 to 7 (E–2 to E–7), in single steps</td>
<td>Inputs the minimum value.</td>
<td>A difference between <code>&lt;max&gt;</code> and <code>&lt;step&gt;</code> must be 7 or greater even number.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>8 to 14 (E–8 to E–14), in single steps</td>
<td></td>
</tr>
<tr>
<td>Vth vs Q</td>
<td>10 to 1000, in 10 steps –40 to 60, in 10 steps</td>
<td>1 to 100, in single steps</td>
<td>When Linear</td>
</tr>
<tr>
<td>Times vs Q</td>
<td>10 to 1000, in 10 steps –40 to 60, in 10 steps</td>
<td>1 to 11, in single steps</td>
<td>When Log</td>
</tr>
<tr>
<td>Histogram</td>
<td>50 to 1000, in 50 steps</td>
<td>Omitted</td>
<td></td>
</tr>
</tbody>
</table>

**Table 7.6.4.1-3 Parameters for Phase vs. Q tab**

<table>
<thead>
<tr>
<th>Format</th>
<th><code>&lt;max&gt;</code></th>
<th><code>&lt;step&gt;</code></th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vth vs BER (Y = Log)</td>
<td>2 to 7 (E–2 to E–7), in single steps</td>
<td>Inputs the minimum value.</td>
<td>A difference between <code>&lt;max&gt;</code> and <code>&lt;step&gt;</code> must be 7 or greater even number.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>8 to 14 (E–8 to E–14), in single steps</td>
<td></td>
</tr>
<tr>
<td>Vth vs BER (Y = Log (−Ln))</td>
<td>2 to 7 (E–2 to E–7), in single steps</td>
<td>Inputs the minimum value.</td>
<td>A difference between <code>&lt;max&gt;</code> and <code>&lt;step&gt;</code> must be 7 or greater even number.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>8 to 14 (E–8 to E–14), in single steps</td>
<td></td>
</tr>
<tr>
<td>Vth vs Q</td>
<td>10 to 1000, in 10 steps –40 to 60, in 10 steps</td>
<td>1 to 100, in single steps</td>
<td>When Linear</td>
</tr>
<tr>
<td>Phase vs Q</td>
<td>10 to 1000, in 10 steps –40 to 60, in 10 steps</td>
<td>1 to 11, in single steps</td>
<td>When Log</td>
</tr>
<tr>
<td>Phase vs σ</td>
<td>0.0010 to 1.0000, in 0.0010 steps</td>
<td>0.001 to 0.1000, 0.0001 steps</td>
<td></td>
</tr>
<tr>
<td>Phase vs μ</td>
<td>–3.990 to 4.000 (V), in 0.010 V steps</td>
<td>0.001 to 0.800 (V), in 0.001 V steps</td>
<td></td>
</tr>
<tr>
<td>Phase vs Opt BER</td>
<td>10 to 1 (E–10 to E–1), in single steps</td>
<td>Inputs the minimum value.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>199 to 20 (E–199 to E–20), in single steps</td>
<td></td>
</tr>
<tr>
<td>Phase vs Opt Threshold</td>
<td>–3.990 to 4.000 (V), in 0.010 V steps</td>
<td>0.001 to 0.800 (V), in 0.001 V steps</td>
<td></td>
</tr>
<tr>
<td>Phase vs Correlation</td>
<td>10 to 100 (%), in 10 % steps</td>
<td>0 to 90 (%), in 10 % steps</td>
<td>A difference between <code>&lt;max&gt;</code> and <code>&lt;step&gt;</code> must be 10 or greater</td>
</tr>
</tbody>
</table>
Note:

Settings vary, depending on the status of the installed options for each slot, and the data input interface settings.

Function
Sets the vertical scale of the graph display during Q measurement.
Inputs the minimum value instead of the step width, only when the vertical axis displays the error rate.

Example
To set the vertical scale during QLog display to maximum 10 dB, in 1-dB steps:
> :DISPlay:RESult:QANalysis:SCALe:VERTical 10,1

Compatibility
Incompatible with existing models.

:DISPlay:RESult:QANalysis:SCALe:VERTical?

Response
<max>,<step> = <NR1 NUMERIC RESPONSE DATA>

Function
Queries the vertical scale of the graph display during Q measurement.

Example
To query the vertical graph scale during QLog display:
> :DISPlay:RESult:QANalysis:SCALe:VERTical?
< 10,1

Compatibility
Incompatible with existing models.
:DISPlay:RESult:QAnalysis:SCALe:HORizontal <max>,<step>

Parameter

<max>,<step> = <DECIMAL NUMERIC PROGRAM DATA>

Table 7.6.4.1-4 Parameters for Threshold vs. Q tab

<table>
<thead>
<tr>
<th>Format</th>
<th>&lt;max&gt;</th>
<th>&lt;step&gt;</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vth vs BER (Y = \log)</td>
<td>(-3.990) to (4.000) (V), in (0.010) V steps</td>
<td>(0.001) to (0.800) (V), in (0.001) V steps</td>
<td>(\text{When Linear}&lt;\max&gt;) must be greater than &lt;step&gt;.</td>
</tr>
<tr>
<td>Vth vs BER (Y = \log (-\ln))</td>
<td>(-3.990) to (4.000) (V), in (0.010) V steps</td>
<td>(0.001) to (0.800) (V), in (0.001) V steps</td>
<td>(\text{When Log})</td>
</tr>
<tr>
<td>Vth vs Q</td>
<td>(-3.990) to (4.000) (V), in (0.010) V steps</td>
<td>(0.001) to (0.800) (V), in (0.001) V steps</td>
<td></td>
</tr>
<tr>
<td>Times vs Q</td>
<td>100 to 1000, in 50 steps</td>
<td>10 (fixed)</td>
<td></td>
</tr>
<tr>
<td>Histogram</td>
<td>0.16 to 1000.00, in 0.01 steps</td>
<td>16/32/64/128/256</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(-49.84) to (60.00) (dB), in 0.01 steps</td>
<td>16/32/64</td>
<td></td>
</tr>
</tbody>
</table>

Table 7.6.4.1-5 Parameters for Phase vs. Q tab

<table>
<thead>
<tr>
<th>Format</th>
<th>&lt;max&gt;</th>
<th>&lt;step&gt;</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vth vs BER (Y = \log)</td>
<td>(-3.990) to (4.000) (V), in (0.010) V steps</td>
<td>(0.001) to (0.800) (V), in (0.001) V steps</td>
<td></td>
</tr>
<tr>
<td>Vth vs BER (Y = \log (-\ln))</td>
<td>(-3.990) to (4.000) (V), in (0.010) V steps</td>
<td>(0.001) to (0.800) (V), in (0.001) V steps</td>
<td></td>
</tr>
<tr>
<td>Vth vs Q</td>
<td>(-3.990) to (4.000) (V), in (0.010) V steps</td>
<td>(0.001) to (0.800) (V), in (0.001) V steps</td>
<td></td>
</tr>
<tr>
<td>Phase vs Q</td>
<td>(-900) to (1000) (mUI), in 100 mUI steps</td>
<td>10 to 200 (mUI), in 10 mUI steps</td>
<td></td>
</tr>
<tr>
<td>Phase vs (\sigma)</td>
<td>(-900) to (1000) (mUI), in 100 mUI steps</td>
<td>10 to 200 (mUI), in 10 mUI steps</td>
<td></td>
</tr>
<tr>
<td>Phase vs (\mu)</td>
<td>(-900) to (1000) (mUI), in 100 mUI steps</td>
<td>10 to 200 (mUI), in 10 mUI steps</td>
<td></td>
</tr>
<tr>
<td>Phase vs Opt BER</td>
<td>(-900) to (1000) (mUI), in 100 mUI steps</td>
<td>10 to 200 (mUI), in 10 mUI steps</td>
<td></td>
</tr>
<tr>
<td>Phase vs Opt Threshold</td>
<td>(-900) to (1000) (mUI), in 100 mUI steps</td>
<td>10 to 200 (mUI), in 10 mUI steps</td>
<td></td>
</tr>
<tr>
<td>Phase vs Correlation</td>
<td>(-900) to (1000) (mUI), in 100 mUI steps</td>
<td>10 to 200 (mUI), in 10 mUI steps</td>
<td></td>
</tr>
</tbody>
</table>

Note:
Settings vary, depending on the status of the installed options for each slot, and the data input interface settings.

Function
Sets the horizontal scale of the graph display during Q measurements. The format varies depending on the graph type.

Example
To set the horizontal scale during Vth display to maximum \(-1.000\) V, in 0.100 steps:
7.6 Automatic Measurement Commands

> :DISPlay:RESult:QANalysis:SCALe:HORizontal -1.000,0.100

Compatibility
Incompatible with existing models.

:DISPlay:RESult:QANalysis:SCALe:HORizontal?

Response
<max>,<step> = <NR1 NUMERIC RESPONSE DATA>

Function
Queries the horizontal scale of the graph display during Q measurement.

Example
To query the horizontal graph scale during Vth display:
> :DISPlay:RESult:QANalysis:SCALe:HORizontal?
< -1.000,0.100

Compatibility
Incompatible with existing models.

:DISPlay:RESult:QANalysis:THReshold:ITEM <mode>

Parameter
<mode> = <CHARACTER PROGRAM DATA>
LBER   Vth vs. BER (Y axis Log)
LNber  Vth vs. BER (Y axis Log (–Ln))
VTHQ   Vth vs. Q
TIMQ   Times vs. Q
HISTogram  Histogram

Function
Sets the item to be displayed in the graph during Vth vs. Q measurement.

Example
To set the item to be displayed to Vth vs. Q:
> :DISPlay:RESult:QANalysis:THReshold:ITEM VTHQ

Compatibility
Incompatible with existing models.

:DISPlay:RESult:QANalysis:THReshold:ITEM?

Response
<mode> = <CHARACTER RESPONSE DATA>
LBER, LNB, VTHQ, TIMQ, HIST

Function
Queries the item to be displayed in the graph during Vth vs. Q measurement.

Example
> :DISPlay:RESult:QANalysis:THReshold:ITEM?
< VTHQ

Compatibility
Incompatible with existing models.
Chapter 7  SCPI Commands

### :DISPlay:RESult:QAnalysis:THReshold:MNUMber <numeric>

**Parameter**  
<numeric> = <DECIMAL PROGRAM DATA>  
1 to 1000 Measurement number (number of measurements during repeat measurement)

**Function**  
Sets the measurement number for which the Vth vs. Q measurement result is displayed in a graph.

**Example**  
To set the measurement number for graph display to 100:

```plaintext
> :DISPlay:RESult:QAnalyze:THReshold:MNUMber 100
```

**Compatibility**  
Incompatible with existing models.

### :DISPlay:RESult:QAnalysis:THReshold:MNUMber?

**Response**  
<numeric> = <NR1 NUMERIC RESPONSE DATA>

**Function**  
Queries the measurement number for which the Vth vs. Q measurement result is displayed in a graph.

**Example**  
To query the measurement number for graph display:

```plaintext
> :DISPlay:RESult:QAnalyze:THReshold:MNUMber?
< 100
```

**Compatibility**  
Incompatible with existing models.

### :DISPlay:RESult:QAnalyze:PHASe:ITEM <mode>

**Parameter**  
<mode> = <CHARACTER PROGRAM DATA>  
- LBER   Vth vs. BER (Y axis Log)
- LNBER  Vth vs. BER (Y axis Log (-Ln))
- VTHQ   Vth vs. Q
- PHAQ   Phase vs. Q
- PHASigma Phase vs. $\sigma$
- PHAMu Phase vs. $\mu$
- PHABer Phase vs. Optimum BER
- PHAVth Phase vs. Optimum Vth
- PHACorre Phase vs. Correlation coefficient

**Function**  
Sets the item to be displayed in the graph during Phase vs. Q measurement.

**Example**  
To set the item to be displayed in the graph to Phase vs. Q:

```plaintext
> :DISPlay:RESult:QAnalyze:PHASe:ITEM PHAQ
```

**Compatibility**  
Incompatible with existing models.
### 7.6 Automatic Measurement Commands

#### :DISPlay:RESult:QANalysis:PHASe:ITEM?

Response

\[
<\text{mode}> = \langle \text{CHARACTER RESPONSE DATA} \rangle
\]

LBER, LNB, VTHQ, PHAQ, PHAS, PHAM, PHAB, PHAV, PHAC

Function

Queries the item to be displayed in the graph during Phase vs. Q measurement.

Example

> :DISPlay:RESult:QANalysis:PHASe:ITEM?

< PHAQ

Compatibility

Incompatible with existing models.

#### :DISPlay:RESult:QANalysis:PHASe:MNUMber <numeric>

Parameter

\[
<\text{numeric}> = \langle \text{DECIMAL PROGRAM DATA} \rangle
\]

1 to 1000 Measurement number 1 to 1000 (number of measurements during repeat measurement)

Function

Sets the measurement number for which the Phase vs. Q measurement result is displayed in a graph.

Example

To set the measurement number graph display to 100:

> :DISPlay:RESult:QANalysis:PHASe:MNUMber 100

Compatibility

Incompatible with existing models.

#### :DISPlay:RESult:QANalysis:PHASe:MNUMber?

Response

\[
<\text{numeric}> = \langle \text{NR1 NUMERIC RESPONSE DATA} \rangle
\]

Function

Queries the measurement number for which the Phase vs. Q measurement result is displayed in a graph.

Example

> :DISPlay:RESult:QANalysis:PHASe:MNUMber?

< 100

Compatibility

Incompatible with existing models.
7.6.4.2 Vth vs. Q measurement commands

Table 7.6.4.2-1 Vth vs. Q measurement setting commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>[10]</td>
<td>Error Threshold</td>
<td></td>
</tr>
<tr>
<td>[12]</td>
<td>Auto Search</td>
<td></td>
</tr>
<tr>
<td>[13]</td>
<td>Correlation Filter</td>
<td></td>
</tr>
<tr>
<td>[14]</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## 7.6 Automatic Measurement Commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>Error Threshold</td>
<td><code>SENSe:MEASure:QANalysis:THReshold:ERANge</code></td>
</tr>
<tr>
<td></td>
<td></td>
<td><code>SENSe:MEASure:QANalysis:THReshold:ERANge?</code></td>
</tr>
<tr>
<td>11</td>
<td>Calculation Threshold</td>
<td><code>SENSe:MEASure:QANalysis:THReshold:CALRange</code></td>
</tr>
<tr>
<td></td>
<td></td>
<td><code>SENSe:MEASure:QANalysis:THReshold:CALRange?</code></td>
</tr>
<tr>
<td>12</td>
<td>Auto Search</td>
<td><code>SENSe:MEASure:QANalysis:THReshold:ASEarch</code></td>
</tr>
<tr>
<td></td>
<td></td>
<td><code>SENSe:MEASure:QANalysis:THReshold:ASEarch?</code></td>
</tr>
<tr>
<td>13</td>
<td>Correlation Filter</td>
<td><code>SENSe:MEASure:QANalysis:THReshold:CFILter</code></td>
</tr>
<tr>
<td></td>
<td></td>
<td><code>SENSe:MEASure:QANalysis:THReshold:CFILter?</code></td>
</tr>
<tr>
<td>14</td>
<td>Current</td>
<td><code>SENSe:MEASure:QANalysis:THReshold:CURRent</code></td>
</tr>
<tr>
<td></td>
<td></td>
<td><code>SENSe:MEASure:QANalysis:THReshold:CURRent?</code></td>
</tr>
</tbody>
</table>

### :SENSe:MEASure:QANalysis:THReshold:START

**Function**
Starts Vth vs. Q measurement.

**Example**
`> :SENSe:MEASure:QANalysis:THReshold:START`

**Compatibility**
Incompatible with existing models.

### :SENSe:MEASure:QANalysis:THReshold:STOP

**Function**
Stops Vth vs. Q measurement.

**Example**
`> :SENSe:MEASure:QANalysis:THReshold:STOP`

**Compatibility**
Incompatible with existing models.

### :SENSe:MEASure:QANalysis:STATe?

**Response**
`<numeric> = <NR1 NUMERIC RESPONSE DATA>`

- 0: Measurement has been stopped.
- 1: During measurement

**Function**
Queries the Q measurement processing state.

**Example**
`> :SENSe:MEASure:QANalysis:STATe?`

**Compatibility**
Incompatible with existing models.
Chapter 7  SCPI Commands

:SENSe:MEASure:QANalysis:THReshold:SLOT <numeric>[,<unit>]

Parameter

<numeric> = <DECIMAL NUMERIC PROGRAM DATA>
1 to 6  Slots No. 1 to No. 6
1 to 6 when using the MP1800A, 1 to 4 when using the MT1810A
[unit] = <DECIMAL NUMERIC PROGRAM DATA>
1 to 4  Mainframe Nos. 1 to 4
When using two or more MT1810A units in serial connection, specify the mainframe number.
Can be omitted. Mainframe No. 1 is specified when omitted.

Function
Sets the Vth vs. Q measurement target slot.

Example
To set the target slot to Slot 2:
> :SENSe:MEASure:QANalysis:THReshold:SLOT 2

Compatibility
Incompatible with existing models.

:SENSe:MEASure:QANalysis:THReshold:SLOT? [<unit>]

Parameter

[unit] = <DECIMAL NUMERIC PROGRAM DATA>
1 to 4  Mainframe Nos. 1 to 4
When using two or more MT1810A units in serial connection, specify the mainframe number.
Can be omitted. Mainframe No. 1 is specified when omitted.

Response

<numeric> = <NR1 NUMERIC RESPONSE DATA>
1 to 6  Slots No. 1 to No. 6

Function
Queries the Vth vs. Q measurement target slot.

Example
> :SENSe:MEASure:QANalysis:THReshold:SLOT?
< 2

Compatibility
Incompatible with existing models.

:SENSe:MEASure:QANalysis:THReshold:MODE <mode>

Parameter

<mode> = <CHARACTER PROGRAM DATA>
SINGle  Executes measurement once.
REPeat  Repeats measurement for the specified number of times.
UNTImed  Repeats measurement until it is directed to stop.

Function
Sets the measurement processing mode during Vth vs. Q measurement.

Example
To set the measurement processing mode to Single mode:
> :SENSe:MEASure:QANalysis:THReshold:MODE SINGle

Compatibility
Incompatible with existing models.
### :SENSe:MEASure:QANalysis:THReshold:MODE?

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Response</td>
<td>&lt;mode&gt; = &lt;CHARACTER RESPONSE DATA&gt; SING, REP, UNT</td>
</tr>
<tr>
<td>Function</td>
<td>Queries the measurement processing mode during Vth vs. Q measurement.</td>
</tr>
<tr>
<td>Example</td>
<td>&gt; :SENSe:MEASure:QANalysis:THReshold:MODE?</td>
</tr>
<tr>
<td></td>
<td>&lt; SING</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Incompatible with existing models.</td>
</tr>
</tbody>
</table>

### :SENSe:MEASure:QANalysis:THReshold:TIMes <numeric>

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter</td>
<td>&lt;numeric&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt; 2 to 1000 2 to 1000 times, in 1-time steps</td>
</tr>
<tr>
<td>Function</td>
<td>Sets the number of measurements during Vth vs. Q measurement.</td>
</tr>
<tr>
<td>Example</td>
<td>To set the number of measurements to 100 times:</td>
</tr>
<tr>
<td></td>
<td>&gt; :SENSe:MEASure:QANalysis:THReshold:TIMes 100</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Incompatible with existing models.</td>
</tr>
</tbody>
</table>

### :SENSe:MEASure:QANalysis:THReshold:TIMes?

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Response</td>
<td>&lt;numeric&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt;</td>
</tr>
<tr>
<td>Function</td>
<td>Queries the number of measurements during Vth vs. Q measurement.</td>
</tr>
<tr>
<td>Example</td>
<td>&gt; :SENSe:MEASure:QANalysis:THReshold:TIMes?</td>
</tr>
<tr>
<td></td>
<td>&lt; 100</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Incompatible with existing models.</td>
</tr>
</tbody>
</table>

### :SENSe:MEASure:QANalysis:THReshold:INTerval <numeric>

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter</td>
<td>&lt;numeric&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt; 0 to 9999 0 to 9999 s, in 1-s steps</td>
</tr>
<tr>
<td>Function</td>
<td>Sets the measurement interval time during Vth vs. Q measurement.</td>
</tr>
<tr>
<td>Example</td>
<td>To set the measurement interval time to 50 sec.:</td>
</tr>
<tr>
<td></td>
<td>&gt; :SENSe:MEASure:QANalysis:THReshold:INTerval 50</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Incompatible with existing models.</td>
</tr>
</tbody>
</table>

### :SENSe:MEASure:QANalysis:THReshold:INTerval?

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Response</td>
<td>&lt;numeric&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt;</td>
</tr>
<tr>
<td>Function</td>
<td>Queries the measurement interval time during Vth vs. Q measurement.</td>
</tr>
<tr>
<td>Example</td>
<td>&gt; :SENSe:MEASure:QANalysis:THReshold:INTerval?</td>
</tr>
<tr>
<td></td>
<td>&lt; 50</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Incompatible with existing models.</td>
</tr>
</tbody>
</table>
Chapter 7  SCPI Commands

:SENSe:MEASure:QANalysis:THReshold:DELay <numeric>
Parameter  <numeric> = <DECIMAL NUMERIC PROGRAM DATA>
           −1000 to 1000  −1000 to 1000 mUI, in 1-mUI steps
Function    Sets the measurement start delay position during Vth vs. Q
            measurement.
Example     To set the measurement start delay position to 100 mUI:
            > :SENSe:MEASure:QANalysis:THReshold:DELay 100
Compatibility Incompatible with existing models.

:SENSe:MEASure:QANalysis:THReshold:DELay?
Response    <numeric> = <NR1 NUMERIC RESPONSE DATA>
Function    Queries the measurement start delay position during Vth vs. Q
            measurement.
Example     > :SENSe:MEASure:QANalysis:THReshold:DELay?
            < 100
Compatibility Incompatible with existing models.

:SENSe:MEASure:QANalysis:THReshold:RESolution <mode>
Parameter  <mode> = <CHARACTER PROGRAM DATA>
           FINE   Fine mode
           COARse Coarse mode
Function    Sets the measurement resolution (Fine/Coarse mode) during Vth vs. Q
            measurement.
Example     To set the measurement resolution to Fine mode:
            > :SENSe:MEASure:QANalysis:THReshold:RESolution FINE
Compatibility Incompatible with existing models.

:SENSe:MEASure:QANalysis:THReshold:RESolution?
Response    <mode> = <CHARACTER RESPONSE DATA>
            FINE, COAR
Function    Queries the measurement resolution (Fine/Coarse mode) during Vth vs. Q
            measurement.
Example     > :SENSe:MEASure:QANalysis:THReshold:RESolution?
            < FINE
Compatibility Incompatible with existing models.
7.6 Automatic Measurement Commands

:SENSe:MEASure:QANalysis:THReshold:ERANge <upper>,<lower>

| Parameter               | <upper> = "<CHARACTER PROGRAM DATA>" |
|                        | E_3 to E_5, E–3 to E–5, in single steps |
|                        | <lower> = "<CHARACTER PROGRAM DATA>" |
|                        | E_7 to E_12, E–7 to E–12, in single steps |
| Function               | Sets the measurement error threshold range during Vth vs. Q measurement. |
| Example                | To set the measurement error threshold range to E–3 to E–12: |
|                        | > :SENSe:MEASure:QANalysis:THReshold:ERANge E_3,E_12 |
| Compatibility          | Incompatible with existing models. |

:SENSe:MEASure:QANalysis:THReshold:ERANge?

| Response               | <upper> = "<CHARACTER RESPONSE DATA>" |
|                        | <lower> = "<CHARACTER PROGRAM DATA>" |
| Function               | Queries the measurement error threshold range during Vth vs. Q measurement. |
| Example                | > :SENSe:MEASure:QANalysis:THReshold:ERANge? |
|                        | < E_3,E_12 |
| Compatibility          | Incompatible with existing models. |

:SENSe:MEASure:QANalysis:THReshold:CALRange <upper>,<lower>

| Parameter               | <upper> = "<CHARACTER PROGRAM DATA>" |
|                        | E_3 to E_5, E–3 to E–5, in single steps |
|                        | <lower> = "<CHARACTER PROGRAM DATA>" |
|                        | E_7 to E_12, E–7 to E–12, in single steps |
| Function               | Sets the error threshold range during Vth vs. Q measurement recalculation. |
| Example                | To set the error threshold range during recalculation to E–5 to E–10: |
|                        | > :SENSe:MEASure:QANalysis:THReshold:CALRange E_5,E_10 |
| Compatibility          | Incompatible with existing models. |

:SENSe:MEASure:QANalysis:THReshold:CALRange?

| Response               | <upper> = "<CHARACTER RESPONSE DATA>" |
|                        | <lower> = "<CHARACTER RESPONSE DATA>" |
| Function               | Queries the error threshold range during Vth vs. Q measurement recalculation. |
| Example                | > :SENSe:MEASure:QANalysis:THReshold:CALRange? |
|                        | < E_5,E_10 |
| Compatibility          | Incompatible with existing models. |
Chapter 7  SCPI Commands

:SENSe:MEASure:QANalysys:THReshold:ASEarch <item>

Parameter  
<item> = <CHARACTER PROGRAM DATA>
OFF   Auto search OFF
THReshold    Threshold auto search
PTHReshold    Phase & threshold auto search

PTHReshold  Phase & Threshold auto search (Fine) Function Sets whether to execute auto search during Vth vs. Q measurement.

Example  
To set the threshold auto search ON:
> :SENSe:MEASure:QANalysys:THReshold:ASEarch THReshold

Compatibility  
Incompatible with existing models.

:SENSe:MEASure:QANalysys:THReshold:ASEarch?

Response  
<item> = <CHARACTER RESPONSE DATA>
OFF, THR, PTHR

Function  
Queries whether to execute auto search during Vth vs. Q measurement.

Example  
> :SENSe:MEASure:QANalysys:THReshold:ASEarch?
< THR

Compatibility  
Incompatible with existing models.

:SENSe:MEASure:QANalysys:THReshold:CFILter <boolean>,[<numeric>]

Parameter  
<boolean> = <BOOLEAN PROGRAM DATA>
0 or OFF  Filter ON
1 or ON  Filter OFF

[<numeric>] = <DECIMAL NUMERIC PROGRAM DATA>
80 to 100  80 to 100%, in 1% steps

When <boolean> is set to "0" or "OFF", <numeric> can be omitted.

Function  
Sets the minimum correlation coefficient with which the Q value is recognized to be valid during Vth vs. Q measurement.

Example  
To set the minimum correlation coefficient to 90%:
> :SENSe:MEASure:QANalysys:THReshold:CFILter 1,90

Compatibility  
Incompatible with existing models.
7.6 Automatic Measurement Commands

:SENSe:MEASure:QANalysis:THReshold:CFILter?

Response

<boolean> = <NR1 NUMERIC RESPONSE DATA>
0  OFF
1  ON

<numeric> = <NR1 NUMERIC RESPONSE DATA>
80 to 100

When <boolean> is 0, <numeric> is omitted.

Function
Queries the minimum correlation coefficient with which the Q value is recognized to be valid during Vth vs. Q measurement.

Example
> :SENSe:MEASure:QANalysis:THReshold:CFILter?
< 1,90

Compatibility
Incompatible with existing models.

:SENSe:MEASure:QANalysis:THReshold:CURRent <boolean>

Parameter

<boolean> = <BOOLEAN PROGRAM DATA>
0 or OFF  OFF: Updates at measurement termination
1 or ON  ON: Updates every second

Function
Sets the graph display update period during Vth vs. Q measurement.

Example
To set the graph display update timing to every second:
> :SENSe:MEASure:QANalysis:THReshold:CURRent 1

Compatibility
Incompatible with existing models.

:SENSe:MEASure:QANalysis:THReshold:CURRent?

Response

<boolean> = <NR1 NUMERIC RESPONSE DATA>
0  OFF
1  ON

Function
Queries the graph display update period during Vth vs. Q measurement.

Example
> :SENSe:MEASure:QANalysis:THReshold:CURRent?
< 1

Compatibility
Incompatible with existing models.
7.6.4.3 Phase vs. Q measurement setting commands

![Figure 7.6.4.3-1 Phase vs. Q measurement setting field]

Table 7.6.4.3-1 Phase vs. Q measurement setting commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>:SENSe:MEAsure:QAnalysis:PHAsc:SLOT?</td>
</tr>
<tr>
<td></td>
<td>Start</td>
<td>:SENSe:MEAsure:QAnalysis:PHAsc:RANGE?</td>
</tr>
<tr>
<td></td>
<td>End</td>
<td>:SENSe:MEAsure:QAnalysis:PHAsc:RANGE?</td>
</tr>
<tr>
<td></td>
<td>Center</td>
<td>:SENSe:MEAsure:QAnalysis:PHAsc:WIDTh</td>
</tr>
<tr>
<td></td>
<td>Span</td>
<td>:SENSe:MEAsure:QAnalysis:PHAsc:WIDTh?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:MEAsure:QAnalysis:PHAsc:RESolution?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:MEAsure:QAnalysis:PHAsc:ERANge?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:MEAsure:QAnalysis:PHAsc:CALRange?</td>
</tr>
</tbody>
</table>
Table 7.6.4.3-1  Phase vs. Q measurement setting commands (Cont’d)

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>:SENSe:MEASure:QANalysis:PHASe:ASEarch?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:MEASure:QANalysis:PHASe:CFILter?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:MEASure:QANalysis:PHASe:CURRent?</td>
</tr>
</tbody>
</table>

**:SENSe:MEASure:QANalysis:PHASe:STARt**

Function
Starts Phase vs. Q measurement.

Example
> :SENSe:MEASure:QANalysis:PHASe:STARt

Compatibility
Incompatible with existing models.

**:SENSe:MEASure:QANalysis:PHASe:STOP**

Function
Stops Phase vs. Q measurement.

Example
> :SENSe:MEASure:QANalysis:PHASe:STOP

Compatibility
Incompatible with existing models.

**:SENSe:MEASure:QANalysis:STATe?**

Response
numeric> = <NR1 NUMERIC RESPONSE DATA>
0   During measurement stop
1   During measurement

Function
Queries the measurement processing state during Q measurement.

Example
> :SENSe:MEASure:QANalysis:STATe?
< 1

Compatibility
Incompatible with existing models.
**:SENSe:MEASure:QANalySis:PHASe:SLOT <numeric>[,<unit>]**

**Parameter**

- `<numeric>` = `<DECIMAL NUMERIC PROGRAM DATA>`
  - 1 to 6   Slots No. 1 to No. 6
  - 1 to 6 when using the MP1800A, 1 to 4 when using the MT1810A
- `[<unit>]` = `<DECIMAL NUMERIC PROGRAM DATA>`
  - 1 to 4   Mainframe Nos. 1 to 4
  - When using two or more MT1810A units in serial connection, specify the mainframe number.
  - Can be omitted. Mainframe No. 1 is specified when omitted.

**Function**

Selects the Phase vs. Q measurement target slot.

**Example**

To set the target slot to Slot2:

```
> :SENSe:MEASure:QANalySis:PHASe:SLOT 2
```

**Compatibility**

Incompatible with existing models.

**:SENSe:MEASure:QANalySis:PHASe:SLOT? [<unit>]**

**Parameter**

- `[<unit>]` = `<DECIMAL NUMERIC PROGRAM DATA>`
  - 1 to 4   Mainframe Nos. 1 to 4
  - When using two or more MT1810A units in serial connection, specify the mainframe number.
  - Can be omitted. Mainframe No. 1 is specified when omitted.

**Response**

- `<numeric>` = `<NR1 NUMERIC RESPONSE DATA>`
  - 1 to 6   Slots No. 1 to No. 6

**Function**

Queries the Phase vs. Q measurement target slot.

**Example**

```
> :SENSe:MEASure:QANalySis:PHASe:SLOT?
< 2
```

**Compatibility**

Incompatible with existing models.

**:SENSe:MEASure:QANalySis:PHASe:RANGe <start>,<end>,<step>**

**Parameter**

- `<start>` = `<DECIMAL NUMERIC PROGRAM DATA>`
  - –1000 to 1000   –1000 to 1000 mUI, in 1-mUI steps
- `<end>` = `<DECIMAL NUMERIC PROGRAM DATA>`
  - –1000 to 1000   –1000 to 1000 mUI, in 1-mUI steps
- `<step>` = `<DECIMAL NUMERIC PROGRAM DATA>`
  - 1 to 200   1 to 200 mUI, in 1-mUI steps

**Function**

Sets the measurement range and measurement step during Phase vs. Q measurement.

**Example**

To set the measurement range to –200 to 300 mUI and measurement step to 10 mUI:

```
> :SENSe:MEASure:QANalySis:PHASe:RANGe -200,300,10
```

**Compatibility**

Incompatible with existing models.
7.6 Automatic Measurement Commands

:SENSe:MEASure:QANalyses:PHAse:RANGe?

Response

<start> = <NR1 NUMERIC RESPONSE DATA>
<end> = <NR1 NUMERIC RESPONSE DATA>
<step> = <NR1 NUMERIC RESPONSE DATA>

Function
Queries the measurement range during Phase vs. Q measurement.

Example

> :SENSe:MEASure:QANalyses:PHAse:RANGe?
< -200,300,10

Compatibility
Incompatible with existing models.

:SENSe:MEASure:QANalyses:PHAse:WIDTh <center>,<span>,<step>

Parameter

<center> = <DECIMAL NUMERIC PROGRAM DATA>
–1000 to 1000 –1000 to 1000 mUI, in 1-mUI steps
<span> = <DECIMAL NUMERIC PROGRAM DATA>
0 to 2000 0 to 2000 mUI, in 1-mUI steps
<step> = <DECIMAL NUMERIC PROGRAM DATA>
1 to 200 1 to 200 mUI, in 1-mUI steps

Function
Sets the measurement range and measurement step during Phase vs. Q measurement.

Example

To set the measurement range to 100 mUI, centered at 200 mUI, in 10-mUI steps:

> :SENSe:MEASure:QANalyses:PHAse:WIDTh 100,200,10

Compatibility
Incompatible with existing models.

:SENSe:MEASure:QANalyses:PHAse:WIDTh?

Response

<center> = <NR1 NUMERIC RESPONSE DATA>
<span> = <NR1 NUMERIC RESPONSE DATA>
<step> = <NR1 NUMERIC RESPONSE DATA>

Function
Queries the measurement range during Phase vs. Q measurement.

Example

> :SENSe:MEASure:QANalyses:PHAse:WIDTh?
< 100,200,10

Compatibility
Incompatible with existing models.
Chapter 7  SCPI Commands

:SENSe:MEASure:QANalysis:PHASe:RESolution <mode>
Parameter  <mode> = <CHARACTER PROGRAM DATA>
FINE       Fine mode
COARse     Coarse mode
Function   Sets the measurement resolution (Fine/Coarse mode) during Phase vs. Q measurement.
Example    To set the measurement resolution to Fine mode:
            > :SENSe:MEASure:QANalysis:PHASe:RESolution FINE
Compatibility Incompatible with existing models.

:SENSe:MEASure:QANalysis:PHASe:RESolution?
Response   <mode> = <CHARACTER RESPONSE DATA>
FINE, COAR
Function   Queries the measurement resolution (Fine/Coarse mode) during Phase vs. Q measurement.
Example    > :SENSe:MEASure:QANalysis:PHASe:RESolution?
            < FINE
Compatibility Incompatible with existing models.

:SENSe:MEASure:QANalysis:PHASe:ERANge <upper>,<lower>
Parameter  <upper> = <CHARACTER PROGRAM DATA>
E_3 to E_5  E–3 to E–5, in single steps
<lower> = <CHARACTER PROGRAM DATA>
E_7 to E_12  E–7 to E–12, in single steps
Function   Sets the measurement error threshold range during Phase vs. Q measurement.
Example    To set the measurement error threshold range to E–3 to E–12:
            > :SENSe:MEASure:QANalysis:PHASe:ERANge E_3,E_12
Compatibility Incompatible with existing models.

:SENSe:MEASure:QANalysis:PHASe:ERANge?
Response   <upper> = <CHARACTER RESPONSE DATA>
<lower> = <CHARACTER RESPONSE DATA>
Function   Queries the measurement error threshold range during Phase vs. Q measurement.
Example    > :SENSe:MEASure:QANalysis:PHASe:ERANge?
            < E_3,E_12
Compatibility Incompatible with existing models.
7.6 Automatic Measurement Commands

:SENSe:MEASure:QANalysis:PHASe:CALRange <upper>,<lower>

Parameter

- <upper> = <CHARACTER PROGRAM DATA>
  - E_3 to E_5  E–3 to E–5, in single steps
- <lower> = <CHARACTER PROGRAM DATA>
  - E_7 to E_12  E–7 to E–12, in single steps

Function
Sets the error threshold range during Phase vs. Q measurement recalculation.

Example
To set the error threshold range during recalculation to E–5 to E–10:
  > :SENSe:MEASure:QANalysis:PHASe:CALRange E_5,E_10

Compatibility
Incompatible with existing models.

:SENSe:MEASure:QANalysis:PHASe:CALRange?

Response

- <upper> = <CHARACTER RESPONSE DATA>
- <lower> = <CHARACTER RESPONSE DATA>

Function
Queries the error threshold range during Phase vs. Q measurement recalculation.

Example
  > :SENSe:MEASure:QANalysis:PHASe:CALRange?
  < E_5,E_10

Compatibility
Incompatible with existing models.

:SENSe:MEASure:QANalysis:PHASe:ASEarch <item>

Parameter

- <item> = <CHARACTER PROGRAM DATA>
  - OFF   Auto search OFF
  - THReshold  Threshold auto search

Function
Sets execution of auto search during Phase vs. Q measurement.

Example
To set the threshold auto search ON:
  > :SENSe:MEASure:QANalysis:PHASe:ASEarch THReshold

Compatibility
Incompatible with existing models.

:SENSe:MEASure:QANalysis:PHASe:ASEarch?

Response

- <item> = <CHARACTER RESPONSE DATA>
  - OFF, THR

Function
Queries the execution of auto search during Phase vs. Q measurement.

Example
  > :SENSe:MEASure:QANalysis:PHASe:ASEarch?
  < THR

Compatibility
Incompatible with existing models.
:SENSe:MEASure:QANalysis:PHASe:CFILter <boolean>[,<numeric>]  

Parameter:
- `<boolean>` = `<BOOLEAN PROGRAM DATA>`
- 0 or OFF: Filter OFF
- 1 or ON: Filter ON
- `<numeric>` = `<DECIMAL NUMERIC PROGRAM DATA>`
- 80 to 100: 80 to 100%, in 1% steps

Function:
Sets the minimum correlation coefficient with which the Q value is recognized to be valid during Phase vs. Q measurement.

Example:
To set the minimum correlation coefficient to 90%:
```plaintext
> :SENSe:MEASure:QANalysis:PHASe:CFILter 1,90
```

Compatibility:
Incompatible with existing models.

:SENSe:MEASure:QANalysis:PHASe:CFILter?

Response:
- `<numeric>` = `<NR1 NUMERIC RESPONSE DATA>`
  - 0: OFF
  - 1: ON
- `<numeric>` = `<NR1 NUMERIC RESPONSE DATA>`
  - 80 to 100: 80 to 100%

Function:
Queries the minimum correlation coefficient with which the Q value is recognized to be valid during Phase vs. Q measurement.

Example:
```plaintext
> :SENSe:MEASure:QANalysis:PHASe:CFILter?
< 1,90
```

Compatibility:
Incompatible with existing models.

:SENSe:MEASure:QANalysis:PHASe:CURRent <boolean>

Parameter:
- `<boolean>` = `<BOOLEAN PROGRAM DATA>`
- 0 or OFF: OFF: Updates at measurement termination
- 1 or ON: ON: Updates every second

Function:
Sets the graph display update period during Phase vs. Q measurement.

Example:
To set the graph display update timing to every second:
```plaintext
> :SENSe:MEASure:QANalysis:PHASe:CURRent 1
```

Compatibility:
Incompatible with existing models.
### :SENSe:MEASure:QANalysis:PHASe:CURRent?

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;boolean&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>OFF</td>
</tr>
<tr>
<td>1</td>
<td>ON</td>
</tr>
</tbody>
</table>

**Function**
Queries the graph display update period during Phase vs. Q measurement.

**Example**

```
> :SENSe:MEASure:QANalysis:PHASe:CURRent?
< 1
```

**Compatibility**
Incompatible with existing models.
7.6.4.4 Measurement results and status query commands

Table 7.6.4.4-1 Measurement results and status query commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1]</td>
<td>No label (Setting of time display)</td>
<td>:SENSe:MEASure:QANalysis:TIME</td>
</tr>
<tr>
<td>[3]</td>
<td>Query for measurement start time</td>
<td>:SENSe:MEASure:QANalysis:STARt?</td>
</tr>
<tr>
<td>[8]</td>
<td>Query for Phase vs. Q measurement status</td>
<td>:CALCulate:DATA:QANalysis:PSTatus?</td>
</tr>
</tbody>
</table>

:SENSe:MEASure:QANalysis:TIME <type>

Parameter  
<type> = <CHARACTER PROGRAM DATA>

Parameter  
DTIME  
Displays the current date and time.

START  
Displays the measurement start time.

ELAPsed  
Displays the elapsed time based on the measurement period.

Function  
Selects the Q measurement time display type.

Example  
To set the Q measurement time display type to measurement start time (Start Time):

> :SENSe:MEASure:QANalysis:TIME START

Compatibility  
Incompatible with existing models.
7.6 Automatic Measurement Commands

:SENSe:MEASure:QANalysis:TIME?

Response

<type> = <CHARACTER RESPONSE DATA>
DTIM, STAR, ELAP

Function
Queries the Q measurement time display type.

Example
> :SENSe:MEASure:QANalysis:TIME?
< STAR

Compatibility
Incompatible with existing models.

:SENSe:MEASure:QANalysis:DTIMe?

Response

<year>,<month>,<day>,<hour>,<min>,<sec>

Function
Queries the current date and time during Q measurement.

Example
> :SENSe:MEASure:QANalysis:DTIMe?
< 2005,12,31,23,59,59

Compatibility
Incompatible with existing models.
**Chapter 7  SCPI Commands**

**:SENSe:MEASure:QANalysis:STARt?**

**Response**

- `<year>` = `<NR1 NUMERIC RESPONSE DATA>`
  - 0, 2000 to 2036  2000 to 2036 year
- `<month>` = `<NR1 NUMERIC RESPONSE DATA>`
  - 0, 1 to 12  January to December
- `<day>` = `<NR1 NUMERIC RESPONSE DATA>`
  - 0, 1 to 31  1st to 31st
- `<hour>` = `<NR1 NUMERIC RESPONSE DATA>`
  - 0 to 23  0 to 23 hours
- `<minute>` = `<NR1 NUMERIC RESPONSE DATA>`
  - 0 to 59  0 to 59 minutes
- `<second>` = `<NR1 NUMERIC RESPONSE DATA>`
  - 0 to 59  0 to 59 seconds

"0,0,0,0,0,0" is returned if there is no measurement start time data.

**Function**

Queries the measurement start time (Start Time) during Q measurement.

**Example**

```plaintext
> :SENSe:MEASure:QANalysis:STARt?
< 2005,12,31,23,59,59
```

**Compatibility**

Incompatible with existing models.

**:SENSe:MEASure:QANalysis:ELAPsed?**

**Response**

- `<day>` = `<NR1 NUMERIC RESPONSE DATA>`
  - 0 to 99  0 to 99 days
- `<hour>` = `<NR1 NUMERIC RESPONSE DATA>`
  - 0 to 23  0 to 23 hours
- `<minute>` = `<NR1 NUMERIC RESPONSE DATA>`
  - 0 to 59  0 to 59 minutes
- `<second>` = `<NR1 NUMERIC RESPONSE DATA>`
  - 0 to 59  0 to 59 seconds

"0,0,0,0" is returned if there is no measurement elapsed time data.

**Function**

Queries the measurement elapsed time (Elapsed time) during Q measurement.

**Example**

```plaintext
> :SENSe:MEASure:QANalysis:ELAPsed?
< 31,23,59,59
```

**Compatibility**

Incompatible with existing models.
7.6 Automatic Measurement Commands

:CALCulate:DATA:QANalysis:THReshold? <string>

Parameter <string> = <STRING PROGRAM DATA>

=result1> For the contents of measurement data <result1> see Table 7.6.4.4-2.

=result2> For the contents of immediate data <result2> see Table 7.6.4.4-3.

=result3> For the contents of statistic data <result3> see Table 7.6.4.4-4.

<table>
<thead>
<tr>
<th>Table 7.6.4.4-2</th>
<th>Vth vs. Q measurement result &lt;result1&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Items</td>
<td>&lt;result1&gt;</td>
</tr>
<tr>
<td>Top side Vth/Error Rate</td>
<td>Number of measurements 1 to 1000</td>
</tr>
<tr>
<td>Bottom side Vth/Error Rate</td>
<td></td>
</tr>
</tbody>
</table>

The measurement result for each Vth is delimited with a comma (,) and returned.

<table>
<thead>
<tr>
<th>Table 7.6.4.4-3</th>
<th>Vth vs. Q immediate data &lt;result2&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Items</td>
<td>&lt;result2&gt;</td>
</tr>
<tr>
<td>Q value</td>
<td>Number of measurements 1 to 1000</td>
</tr>
<tr>
<td>Optimum BER</td>
<td></td>
</tr>
<tr>
<td>Optimum Vth</td>
<td></td>
</tr>
<tr>
<td>Correlation bottom</td>
<td></td>
</tr>
<tr>
<td>Correlation top</td>
<td></td>
</tr>
<tr>
<td>σ bottom</td>
<td></td>
</tr>
<tr>
<td>σ top</td>
<td></td>
</tr>
<tr>
<td>μ bottom</td>
<td></td>
</tr>
<tr>
<td>μ top</td>
<td></td>
</tr>
<tr>
<td>Q Equation</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 7.6.4.4-4</th>
<th>Vth vs. Q measurement statistic data &lt;result3&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Items</td>
<td>&lt;Result3&gt;</td>
</tr>
<tr>
<td>Total Data</td>
<td>&quot;TOTal&quot;</td>
</tr>
<tr>
<td>Valid Data</td>
<td>&quot;VALid&quot;</td>
</tr>
<tr>
<td>Q Max</td>
<td>&quot;QMAX&quot;</td>
</tr>
<tr>
<td>Q Min</td>
<td>&quot;QMIN&quot;</td>
</tr>
<tr>
<td>Q Mean</td>
<td>&quot;QMEan&quot;</td>
</tr>
<tr>
<td>Q σ</td>
<td>&quot;QSGM&quot;</td>
</tr>
<tr>
<td>Q-5 σ</td>
<td>&quot;QSGM5&quot;</td>
</tr>
</tbody>
</table>

Response <string> = <STRING RESPONSE DATA>
Chapter 7  SCPI Commands

Table 7.6.4-5  Response format

<table>
<thead>
<tr>
<th>Form</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Form1</td>
<td>&quot;XX.XXX.Y.YYYYE-YYYY&quot;</td>
<td>XX.XXX: Threshold (V)</td>
</tr>
<tr>
<td></td>
<td>&quot;---------&quot;</td>
<td>Y.YYYYE–YYY: Error Rate</td>
</tr>
<tr>
<td></td>
<td>When no data corresponds to the query.</td>
<td></td>
</tr>
<tr>
<td>Form2</td>
<td>&quot;XXX.XX&quot;</td>
<td>When 0.00 to 100.00</td>
</tr>
<tr>
<td></td>
<td>&quot;---------&quot;</td>
<td>When no data corresponds to the query.</td>
</tr>
<tr>
<td>Form3</td>
<td>&quot;X.XXXE-XXX&quot;</td>
<td>When 0.0000E–016 to 1.0000E000</td>
</tr>
<tr>
<td></td>
<td>&quot;---------&quot;</td>
<td>When no data corresponds to the query.</td>
</tr>
<tr>
<td></td>
<td>&quot;&lt; 1.0E-199&quot;</td>
<td>When E–199 or less</td>
</tr>
<tr>
<td>Form4</td>
<td>&quot;XX.XXXX&quot;</td>
<td>When –4.0000 to 4.0000</td>
</tr>
<tr>
<td></td>
<td>&quot;---------&quot;</td>
<td>When no data corresponds to the query.</td>
</tr>
<tr>
<td>Form5</td>
<td>&quot;XXXX&quot;</td>
<td>When 0 to 9999</td>
</tr>
<tr>
<td></td>
<td>&quot;---------&quot;</td>
<td>When no data corresponds to the query.</td>
</tr>
</tbody>
</table>

Function
Queries the Vth vs. Q measurement result.

Example
To query the measurement result of the 10th time, on the top side:

> :CALCulate:DATA:QANalysis:THReshold? "VTHQ:TOP,10"
< "1.000,1.2345E-003", "1.100,1.2345E-004",
"1.200,1.2345E-005",
"1.300,1.2345E-006"

Compatibility
Incompatible with existing models.

:CALCulate:DATA:QANalysis:TSTatus?

Response

<STRING RESPONSE DATA>
""
"Sync Loss"  Sync. Loss
"Clock Loss"  Clock Loss
"CR Unlock"  CR Unlock
"Out of range"  Out of range
"Frequency NG"  Frequency NG
"Illegal Error"  Illegal Error

Note:
Displays any one of the alarms.
When two or more alarms have occurred, they are displayed in the following order:
Illegal Error > Frequency NG > Clock Loss (CR Unlock) > Sync Loss > Out of range

Function
Queries the Vth vs. Q measurement status.

Example
> :CALCulate:DATA:QANalysis:TSTatus?
< "Sync Loss"

Compatibility
Incompatible with existing models.
7.6 Automatic Measurement Commands

:CALCulate:DATA:QANalysis:PHASe? <string>

Parameter

<string> = <STRING PROGRAM DATA>

<result1>
For the contents of measurement data <result1>, see Table 7.6.4.4-6.

<result2>
For the contents of immediate data <result2>, see Table 7.6.4.4-7.

Table 7.6.4.4-6 Phase vs. Q measurement result <result1>

<table>
<thead>
<tr>
<th>Items</th>
<th>&lt;result1&gt;</th>
<th>Response Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>Top side</td>
<td>Vth-Error Rate</td>
<td></td>
</tr>
<tr>
<td>Vth-Error Rate</td>
<td>Phase</td>
<td>&quot;VTHQ:TOP,XXXXX&quot;</td>
</tr>
<tr>
<td></td>
<td>–1000 to 1000 mUI</td>
<td></td>
</tr>
<tr>
<td>Bottom side</td>
<td>Vth-Error Rate</td>
<td></td>
</tr>
<tr>
<td>Vth-Error Rate</td>
<td></td>
<td>&quot;VTHQ:BTM,XXXXX&quot;</td>
</tr>
</tbody>
</table>

Table 7.6.4.4-7 Phase vs. Q immediate data <result2>

<table>
<thead>
<tr>
<th>Items</th>
<th>&lt;result2&gt;</th>
<th>Response Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q value</td>
<td>Phase</td>
<td></td>
</tr>
<tr>
<td></td>
<td>–1000 to 1000</td>
<td>&quot;QVALue:–1000&quot;... &quot;QVALue:1000&quot;</td>
</tr>
<tr>
<td>Optimum BER</td>
<td></td>
<td>&quot;OPTBer:–1000&quot;... &quot;OPTBer:1000&quot;</td>
</tr>
<tr>
<td>Optimum Vth</td>
<td></td>
<td>&quot;OPTVth:–1000&quot;... &quot;OPTVth:1000&quot;</td>
</tr>
<tr>
<td>Correlation bottom</td>
<td></td>
<td>&quot;CORBtm:–1000&quot;... &quot;CORBtm:1000&quot;</td>
</tr>
<tr>
<td>Correlation top</td>
<td></td>
<td>&quot;CORTop:–1000&quot;... &quot;CORTop:1000&quot;</td>
</tr>
<tr>
<td>Σ bottom</td>
<td></td>
<td>&quot;SGMBtm:–1000&quot;... &quot;SGMBtm:1000&quot;</td>
</tr>
<tr>
<td>Σ top</td>
<td></td>
<td>&quot;SGMTop:–1000&quot;... &quot;SGMTop:1000&quot;</td>
</tr>
<tr>
<td>M bottom</td>
<td></td>
<td>&quot;MUBtm:–1000&quot;... &quot;MUBtm:1000&quot;</td>
</tr>
<tr>
<td>M top</td>
<td></td>
<td>&quot;MUTop:–1000&quot;... &quot;MUTop:1000&quot;</td>
</tr>
<tr>
<td>Q equation</td>
<td></td>
<td>&quot;QEQ:–1000&quot;... &quot;QEQ:1000&quot;</td>
</tr>
</tbody>
</table>

Response

<string> = <STRING RESPONSE DATA>

Table 7.6.4.4-8 Response format

<table>
<thead>
<tr>
<th>Form</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Form1</td>
<td>&quot;XX.XXX,Y.YYYYE–YYY&quot;</td>
<td>XX.XXX: Threshold (V) Y.YYYYE–YYY: Error Rate</td>
</tr>
<tr>
<td></td>
<td>&quot;.......&quot;,.......&quot;</td>
<td>When no data corresponds to the query.</td>
</tr>
<tr>
<td>Form2</td>
<td>&quot;XXX.XX&quot;</td>
<td>When 0.00 to 100.00</td>
</tr>
<tr>
<td></td>
<td>&quot;.........&quot;</td>
<td>When no data corresponds to the query.</td>
</tr>
<tr>
<td>Form3</td>
<td>&quot;X.XXXXE-XXX&quot;</td>
<td>When 0.0000E-016 to 1.0000E000</td>
</tr>
<tr>
<td></td>
<td>&quot;.........&quot;</td>
<td>When no data corresponds to the query.</td>
</tr>
<tr>
<td></td>
<td>&quot;&lt; 1.0E-199&quot;</td>
<td>When E-199 or less</td>
</tr>
<tr>
<td>Form4</td>
<td>&quot;XX.XXX&quot;</td>
<td>When –4.0000 to 4.0000</td>
</tr>
<tr>
<td></td>
<td>&quot;.........&quot;</td>
<td>When no data corresponds to the query.</td>
</tr>
</tbody>
</table>

Function

Queries the Phase vs. Q measurement result.

Example

To query the TOP side measurement result:
Chapter 7  SCPI Commands

>:CALCulate:DATA:QANalysis:PHASe? "VTHQ:TOP,100"
< "10,1.000,1.2345E-003", "10,1.100,1.2345E-004",
"10,1.200,1.2345E-005"

Compatibility
Incompatible with existing models.

>:CALCulate:DATA:QANalysis:PSTatus?

Response
<string> = <STRING RESPONSE DATA>

"" When no alarm exists
"Sync Loss" Sync. Loss
"Clock Loss" Clock Loss
"CR Unlock" CR Unlock
"Out of range" Out of range
"Frequency NG" Frequency NG
"Illegal Error" Illegal Error

Note:
Displays any one of the alarms.
When two or more alarms have occurred, they are displayed in the following order:
Illegal Error > Frequency NG > Clock Loss (CR Unlock) > Sync
Loss > Out of range

Function
Queries the Phase vs. Q measurement status.

Example
>:CALCulate:DATA:QANalysis:PSTatus?
< "Sync Loss"
7.6 Automatic Measurement Commands

7.6.4.5 File menu setting commands

Table 7.6.4.5-1 File menu setting commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>:SENSe:MEASure:QANalysis:THReshold:LSCale?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:MEASure:QANalysis:PHASe:LSCale</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:MEASure:QANalysis:PHASe:LSCale?</td>
</tr>
</tbody>
</table>

:SYSTem:MMEMory:QANalysis:RECall <file_name>

Parameter

- `<file_name>` = `<STRING PROGRAM DATA>`
- 
  "<drv>:\[<dir>]<file>"
- 
  `<drv>` = C, D, E, F
- 
  `<dir>` = `<dir1>\<dir2>\...` (Omitted when root directory)
- 
  `<file>` = File name

Function

Opens the Q measurement result data.

Example

> :SYSTem:MMEMory:QANalysis:RECall "C:\Test\example"

Compatibility

Incompatible with existing models.
### :SYSTem:MMEMory:QAnalysis:STORe

**Parameter**

- `<file_name>` = `<STRING PROGRAM DATA>`
  - "<drv>:<dir>:<file>"
  - `<drv>` = C, D, E, F
  - `<dir>` = `<dir1>:<dir2>:...` (Omitted when root directory)
  - `<file>` = File name

- `<data_type>` = `<CHARACTER PROGRAM DATA>`
  - VTQ = Vth vs. Q Result
  - PSQ = Phase vs. Q Result

- `<file_type>` = `<CHARACTER PROGRAM DATA>`
  - BIN = Binary file
  - CSV = CSV file
  - TXT = Text file

**Function**

Saves the Q measurement result data.

**Note:**

The settings will not be read from the saved file if the file name is changed.

**Example**

Saves the Vth vs. Q measurement result data of Q measurement by specifying the save destination, file name, and file format.

```plaintext
> :SYSTem:MMEMory:QAnalysis:STORe "C:\Test\example",VTQ,CSV
```

**Compatibility**

Incompatible with existing models.

---

### :SYSTem:PRINT:QAnalysis <list>

**Parameter**

- `<list>` = `<CHARACTER PROGRAM DATA>`
  - VTQ = Prints Vth vs. Q measurement results.
  - PSQ = Prints Phase vs. Q measurement results.

**Function**

Prints the Q measurement result data.

**Example**

To print the Vth vs. Q measurement results:

```plaintext
:SYSTem:PRINT:QAnalysis VTQ
```

**Compatibility**

Incompatible with existing models.
### :SENSe:MEASure:QANalysis:THReshold:LScale <scale>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;scale&gt;</td>
<td>&lt;CHARACTER PROGRAM DATA&gt;</td>
</tr>
<tr>
<td>LI Near</td>
<td>Linear display</td>
</tr>
<tr>
<td>LOG</td>
<td>Log display</td>
</tr>
</tbody>
</table>

**Function**
Sets the Q value display scale during Vth vs. Q measurement to Linear or Log display.

**Example**
To set the display scale to Log display:

```
> :SENSe:MEASure:QANalysis:THReshold:LScale LOG
```

**Compatibility**
Incompatible with existing models.

### :SENSe:MEASure:QANalysis:THReshold:LScale? 

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;scale&gt; = &lt;CHARACTER RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>LI, LOG</td>
<td></td>
</tr>
</tbody>
</table>

**Function**
Queries whether the Q value display scale during Vth vs. Q measurement is set to Linear or Log display.

**Example**

```
> :SENSe:MEASure:QANalysis:THReshold:LScale?
< LOG
```

**Compatibility**
Incompatible with existing models.

### :SENSe:MEASure:QANalysis:PHASe:LScale <scale>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;scale&gt;</td>
<td>&lt;CHARACTER PROGRAM DATA&gt;</td>
</tr>
<tr>
<td>LI Near</td>
<td>Linear display</td>
</tr>
<tr>
<td>LOG</td>
<td>Log display</td>
</tr>
</tbody>
</table>

**Function**
Sets the Q value display scale during Phase vs. Q measurement to Linear or Log display.

**Example**
To set the display scale to Log display:

```
> :SENSe:MEASure:QANalysis:PHASe:LScale LOG
```

**Compatibility**
Incompatible with existing models.

### :SENSe:MEASure:QANalysis:PHASe:LScale? 

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;scale&gt; = &lt;CHARACTER RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>LI, LOG</td>
<td></td>
</tr>
</tbody>
</table>

**Function**
Queries whether the Q value display scale during Phase vs. Q measurement is set to Linear or Log display.

**Example**

```
> :SENSe:MEASure:QANalysis:PHASe:LScale?
< LOG
```

**Compatibility**
Incompatible with existing models.
7.6.5 Bathtub measurement

This section describes the commands for Bathtub measurement.

Figure 7.6.5-1  Bathtub measurement screen

7.6.5.1 Measurement setting commands

Figure 7.6.5.1-1  Measurement setting field
# Automatic Measurement Commands

## Table 7.6.5.1-1  Measurement setting commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1]</td>
<td>Start</td>
<td>:SENSe:MEASure:BATHtub:STARt</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:MEASure:BATHtub:SLOT?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:MEASure:BATHtub:MODE?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:MEASure:BATHtub:TIME?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:MEASure:BATHtub:INTerval?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:MEASure:BATHtub:DATA?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:MEASure:BATHtub:RESolution:MUI?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:MEASure:BATHtub:ECount?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:MEASure:BATHtub:RANGe?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:MEASure:BATHtub:CALRange?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:MEASure:BATHtub:ASEarch?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:MEASure:BATHtub:CURRent?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:MEASure:BATHtub:JCALculation?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:MEASure:BATHtub:J2Meas?</td>
</tr>
</tbody>
</table>

### :SENSe:MEASure:BATHtub:START

**Function**
Starts Bathtub measurement.

**Example**
> :SENSe:MEASure:BATHtub:STARt

**Compatibility**
Incompatible with existing models.

### :SENSe:MEASure:BATHtub:STOP

**Function**
Stops Bathtub measurement.

**Example**
> :SENSe:MEASure:BATHtub:STOP

**Compatibility**
Incompatible with existing models.
### :SENSe:MEASure:BATHtub:STATe?

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;numeric&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>1</td>
</tr>
</tbody>
</table>

**Function**

Queries the measurement processing state during Bathtub measurement.

**Example**

```plaintext
> :SENSe:MEASure:BATHtub:STATe?
< 1
```

**Compatibility**

Incompatible with existing models.

### :SENSe:MEASure:BATHtub:SLOT <numeric>[,<unit>]

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;numeric&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1 to 6</td>
</tr>
<tr>
<td></td>
<td>1 to 6 when using the MP1800A, 1 to 4 when using the MT1810A</td>
</tr>
<tr>
<td>[&lt;unit&gt;]</td>
<td>&lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
</tr>
<tr>
<td></td>
<td>1 to 4</td>
</tr>
<tr>
<td></td>
<td>When using two or more MT1810A units in serial connection, specify the mainframe number.</td>
</tr>
<tr>
<td></td>
<td>Can be omitted. Mainframe No. 1 is specified when omitted.</td>
</tr>
</tbody>
</table>

**Function**

Sets the Bathtub measurement target slot.

**Example**

To set the target slot to slot 3.

```plaintext
> :SENSe:MEASure:BATHtub:SLOT 3
```

**Compatibility**

Incompatible with existing models.

### :SENSe:MEASure:BATHtub:SLOT? [<unit>]

<table>
<thead>
<tr>
<th>Parameter</th>
<th>[&lt;unit&gt;] = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1 to 4</td>
</tr>
<tr>
<td></td>
<td>When using two or more MT1810A units in serial connection, specify the mainframe number.</td>
</tr>
<tr>
<td></td>
<td>Can be omitted. Mainframe No. 1 is specified when omitted.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;numeric&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt;</th>
</tr>
</thead>
</table>

**Function**

Queries the Bathtub measurement target slot.

**Example**

```plaintext
> :SENSe:MEASure:BATHtub:SLOT?
< 3
```

**Compatibility**

Incompatible with existing models.
### :SENSe:MEASure:BATHtub:MODE <mode>

**Parameter**

<table>
<thead>
<tr>
<th>&lt;mode&gt; = &lt;CHARACTER PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>SINGLE Execute measurement once.</td>
</tr>
<tr>
<td>REPeat Repeats measurement for the specified number of times.</td>
</tr>
<tr>
<td>UNTImed Repeats measurement until measurement is stopped.</td>
</tr>
</tbody>
</table>

**Function**

Sets the measurement processing mode during Bathtub measurement.

**Example**

To set the measurement processing mode to Single mode.

\[ > :SENSe:MEASure:BATHtub:MODE \text{ SINGle} \]

**Compatibility**

Incompatible with existing models.

### :SENSe:MEASure:BATHtub:MODE?

**Response**

| <mode> = <CHARACTER RESPONSE DATA> |
| SING, REP, UNT |

**Function**

Queries the measurement processing mode during Bathtub measurement.

**Example**

> :SENSe:MEASure:BATHtub:MODE?

< SING

**Compatibility**

Incompatible with existing models.

### :SENSe:MEASure:BATHtub:TIMes <numeric>

**Parameter**

<table>
<thead>
<tr>
<th>&lt;numeric&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 to 1000 2 to 1000 times, in 1-time steps</td>
</tr>
</tbody>
</table>

**Function**

Sets the number of measurements during Bathtub measurement.

**Example**

To set the number of measurements to 100 times:

\[ > :SENSe:MEASure:BATHtub:TIMes 100 \]

**Compatibility**

Incompatible with existing models.

### :SENSe:MEASure:BATHtub:TIMes?

**Response**

| <numeric> = <NR1 NUMERIC RESPONSE DATA> |

**Function**

Queries the number of measurements during Bathtub measurement.

**Example**

> :SENSe:MEASure:BATHtub:TIMes?

< 100

**Compatibility**

Incompatible with existing models.
Chapter 7  SCPI Commands

:SENSe:MEASure:BATHtub:INTerval <numeric>

Parameter  
<numeric> = <DECIMAL NUMERIC PROGRAM DATA>
0 to 9999    0 to 9999 s, in 1-s steps

Function  
Sets the measurement interval time during Bathtub measurement.

Example  
To set the measurement interval time to 50 s:
> :SENSe:MEASure:BATHtub:INTerval 50

Compatibility  
Incompatible with existing models.

:SENSe:MEASure:BATHtub:INTerval?

Response  
<numeric> = <NR1 NUMERIC RESPONSE DATA>
0 to 9999    0 to 9999 s

Function  
Queries the measurement interval time during Bathtub measurement.

Example  
> :SENSe:MEASure:BATHtub:INTerval?
< 50

Compatibility  
Incompatible with existing models.

:SENSe:MEASure:BATHtub:DATA <numeric>

Parameter  
<numeric> = <DECIMAL NUMERIC PROGRAM DATA>
–3.500 to 3.300    –3.500 to 3.300 V, in 0.001-V steps
The range varies according to the option and input interface settings.

Function  
Sets the measurement voltage threshold position during Bathtub measurement.

Example  
To set the measurement voltage threshold position to 0.5 V:
> :SENSe:MEASure:BATHtub:DATA 0.5

Compatibility  
Incompatible with existing models.

:SENSe:MEASure:BATHtub:DATA?

Response  
<numeric> = <NR1 NUMERIC RESPONSE DATA>

Function  
Queries the measurement voltage threshold position during Bathtub measurement.

Example  
> :SENSe:MEASure:BATHtub:DATA?
< 0.505

Compatibility  
Incompatible with existing models.
### 7.6 Automatic Measurement Commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>:SENSe:MEASure:BATHtub:RESolution:MUI &lt;numeric&gt;</strong></td>
<td>Sets the Bathtub measurement phase resolution (mUI).</td>
<td>&gt; :SENSe:MEASure:BATHtub:RESolution:MUI 10</td>
</tr>
<tr>
<td><strong>:SENSe:MEASure:BATHtub:ECOunt &lt;mode&gt;</strong></td>
<td>Sets the minimum number of error counts during Bathtub measurement.</td>
<td>&gt; :SENSe:MEASure:BATHtub:ECOunt FINE</td>
</tr>
<tr>
<td><strong>:SENSe:MEASure:BATHtub:ECOunt?</strong></td>
<td>Queries the minimum number of error counts during Bathtub measurement.</td>
<td>&gt; :SENSe:MEASure:BATHtub:ECOunt? &lt; FINE</td>
</tr>
</tbody>
</table>
Chapter 7  SCPI Commands

:SENSe:MEASure:BATHtub:RANGe <range>

Parameter  

- <range> = <CHARACTER PROGRAM DATA>
- E_7 to E_14  
- E–7 to E–14

Function  
Sets the lower limit value of measurement error threshold during Bathtub measurement.

Example  
To set the lower limit value of measurement error threshold to E–12:

```
> :SENSe:MEASure:BATHtub:RANGe E_12
```

Compatibility  
Incompatible with existing models.

:SENSe:MEASure:BATHtub:RANGe?

Response  

- <range> = <CHARACTER RESPONSE DATA>

Function  
Queries the lower limit value of measurement error threshold during Bathtub measurement.

Example  
To query the lower limit value of measurement error threshold:

```
> :SENSe:MEASure:BATHtub:RANGe?
< E_12
```

Compatibility  
Incompatible with existing models.

:SENSe:MEASure:BATHtub:CALRange <upper>,<lower>

Parameter  

- <upper> = <CHARACTER PROGRAM DATA>
- E_3 to E_5  
- E–3 to E–5
- <lower> = <CHARACTER PROGRAM DATA>
- E_7 to E_14  
- E–7 to E–14

Function  
Sets the error threshold range during Bathtub measurement recalculation.

Example  
To set the error threshold range during recalculation to E–5 to E–10:

```
> :SENSe:MEASure:BATHtub:CALRange E_5,E_10
```

Compatibility  
Incompatible with existing models.

:SENSe:MEASure:BATHtub:CALRange?

Response  

- <upper>,<lower> = <CHARACTER RESPONSE DATA>

Function  
Queries the error threshold range during Bathtub measurement recalculation.

Example  
To query the error threshold range during recalculation:

```
> :SENSe:MEASure:BATHtub:CALRange?
< E_5,E_10
```

Compatibility  
Incompatible with existing models.
7.6 Automatic Measurement Commands

:SENSe:MEASure:BATHtub:ASEarch

Parameter

- `<item> = <CHARACTER PROGRAM DATA>`
  - OFF: Auto search OFF
  - PHAS: Phase auto search (Coarse)
  - PTHReshold: Phase & threshold auto search (Coarse)
  - FPHase: Phase auto search (Fine)
  - FPTHreshold: Phase & Threshold auto search (Fine)

Function
Sets execution of auto search during Bathtub measurement.

Example
To set the phase auto search ON:

```
> :SENSe:MEASure:BATHtub:ASEarch PHASe
```

Compatibility
Incompatible with existing models.

:SENSe:MEASure:BATHtub:ASEarch?

Response

- `<item> = <CHARACTER RESPONSE DATA>`
  - OFF, PHAS, PTHR, FPH, FPTH

Function
Queries the execution of auto search during Bathtub measurement.

Example

```
> :SENSe:MEASure:BATHtub:ASEarch?
< PHAS
```

Compatibility
Incompatible with existing models.

:SENSe:MEASure:BATHtub:CURRent <boolean>

Parameter

- `<boolean> = <BOOLEAN PROGRAM DATA>`
  - 0 or OFF: OFF: Updates at measurement termination.
  - 1 or ON: ON: Updates every second.

Function
Sets the graph display update period during Bathtub measurement.

Example
To set the graph display update timing to every second:

```
> :SENSe:MEASure:BATHtub:CURRent 1
```

Compatibility
Incompatible with existing models.

:SENSe:MEASure:BATHtub:CURRent?

Response

- `<boolean> = <NR1 NUMERIC RESPONSE DATA>`
  - 0, 1

Function
Queries the graph display update period during Bathtub measurement.

Example

```
> :SENSe:MEASure:BATHtub:CURRent?
< 1
```

Compatibility
Incompatible with existing models.
Chapter 7  SCPI Commands

:*SENSe:MEASure:BATHtub:JCALculation <setting>
Parameter  <setting> = <DECIMAL NUMERIC PROGRAM DATA>
  7 to 20  E–7 to E–20
Specified as E–n where n: 7 to 20/step
Function  Sets the error rate used to calculate jitter at Bathtub measurement
Example  To set jitter calculation error rate to E–14
  > :SENSe:MEASure:BATHtub:JCALculation 14
Compatibility  Incompatible with existing models.

:*SENSe:MEASure:BATHtub:JCALculation?
Response  <setting> = <NR1 NUMERIC RESPONSE DATA>
  7 to 20
Function  Queries the error rate used to calculate jitter at Bathtub measurement
Example  > :SENSe:MEASure:BATHtub:JCALculation?
  < 14
Compatibility  Incompatible with existing models.

:*SENSe:MEASure:BATHtub:J2Meas
Parameter  <mode> = <DECIMAL NUMERIC PROGRAM DATA>
  ACTual  Calculates J2 from BER measurement point.
  ESTimate  Calculates J2 from Best Fit Line.
Function  Selects J2 calculation method.
Example  To set J2 calculation method to Actual
  > :SENSe:MEASure:BATHtub:J2Meas ACTual
Compatibility  Incompatible with existing models.

:*SENSe:MEASure:BATHtub:J2Meas?
Response  <mode> = <NR1 NUMERIC RESPONSE DATA>
  ACT, EST
Function  Queries J2 calculation method.
Example  > :SENSe:MEASure:BATHtub:J2Meas?
  < ACT
Compatibility  Incompatible with existing models.
7.6 Automatic Measurement Commands

7.6.5.2 Graph display setting commands

Figure 7.6.5.2-1  Graph display screen

Table 7.6.5.2-1  Graph display setting commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>:DISPlay:RESult:BATHtub:ITEM?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:DISPlay:RESult:BATHtub:MNUMber?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:DISPlay:RESult:BATHtub:SCALe:HORizontal?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:DISPlay:RESult:BATHtub:SCALe:VERTical?</td>
</tr>
</tbody>
</table>

:DISPlay:RESult:BATHtub:ITEM <mode>

Parameter
- <mode> = <CHARACTER PROGRAM DATA>
- PBER  Phase vs. BER (Y axis Log)
- PLNBer  Phase vs. BER (Y axis Log (~Ln))
- HISTogram  Histogram

Function
Sets the item to be displayed in the graph during Bathtub measurement.

Example
To set the display item to Histogram:
> :DISPlay:RESult:BATHtub:ITEM HISTogram

Compatibility
Incompatible with existing models.
Chapter 7  SCPI Commands

:DISPlay:RESult:BATHtub:ITEM?

Response  
<mode> = <CHARACTER RESPONSE DATA>
PBER, PLNB, HIST

Function  
Queries the item to be displayed in the graph during Bathtub measurement.

Example  
> :DISPlay:RESult:BATHtub:ITEM?
< HIST

Compatibility  
Incompatible with existing models.

:DISPlay:RESult:BATHtub:MNUMber <numeric>

Parameter  
<numeric> = <DECIMAL PROGRAM DATA>
1 to 1000 Measurement number 1 to 1000 (number of measurements during repeat measurement)

Function  
Selects the measurement number for which the Bathtub measurement result is displayed in a graph.

Example  
To select the measurement number 100 to display the graph:
> :DISPlay:RESult:BATHtub:MNUMber 100

Compatibility  
Incompatible with existing models.

:DISPlay:RESult:BATHtub:MNUMber?

Response  
<numeric> = <NR1 NUMERIC RESPONSE DATA>

Function  
Queries the measurement number for which the Bathtub measurement result is displayed in a graph.

Example  
> :DISPlay:RESult:BATHtub:MNUMber?
< 100

Compatibility  
Incompatible with existing models.

:DISPlay:RESult:BATHtub:SCALe:ASCale

Function  
Automatically adjusts the graph display during Bathtub measurement

Example  
> :DISPlay:RESult:BATHtub:SCALe:ASCale

Compatibility  
Incompatible with existing models.
### 7.6 Automatic Measurement Commands

#### :DISPlay:RESult:BATHtub:SCALe:HORizontal <max>[,<step>]

**Parameter**

<max>,<step> = <DECIMAL NUMERIC PROGRAM DATA>

**Table 7.6.5.2-2 Parameter**

<table>
<thead>
<tr>
<th>Form</th>
<th>&lt;max&gt;</th>
<th>&lt;step&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Phase type</td>
<td>-900 to 1000 (mUI), in 100-mUI steps</td>
<td>10 to 200 (mUI), in 10-mUI steps</td>
</tr>
<tr>
<td>Histogram type</td>
<td>16,32,64,128,256</td>
<td>Omitted</td>
</tr>
</tbody>
</table>

**Function**

Sets the horizontal scale of the graph display during Bathtub measurement.

In case of a histogram, <step> is omitted.

**Example**

To set the scale to maximum 800 mUI, in 100-mUI steps:

> :DISPlay:RESult:BATHtub:SCALe:HORizontal 800,100

**Compatibility**

Incompatible with existing models.

#### :DISPlay:RESult:BATHtub:SCALe:HORizontal?

**Response**

<max>,<step> = <NR2 NUMERIC RESPONSE DATA>

Same as the :DISPlay:RESult:BATHtub:SCALe:HORizontal command parameter.

**Function**

Queries the horizontal scale of the graph display during Bathtub measurement.

**Example**

> :DISPlay:RESult:BATHtub:SCALe:HORizontal?

< 800,100

**Compatibility**

Incompatible with existing models.

#### :DISPlay:RESult:BATHtub:SCALe:VERTical <max>,<min>

**Parameter**

<max>,<min> = <DECIMAL NUMERIC PROGRAM DATA>

**Table 7.6.5.2-3 Parameter**

<table>
<thead>
<tr>
<th>Form</th>
<th>&lt;max&gt;</th>
<th>&lt;min&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Histogram type</td>
<td>6 to 14 (E+6 to E+14)</td>
<td>Input 0</td>
</tr>
<tr>
<td>Error Rate type</td>
<td>2 to 7 (E–2 to E–7)</td>
<td>8 to 15 (E–8 to E–15)</td>
</tr>
</tbody>
</table>

**Note:**

In case of the Error Rate type, the difference between the maximum and minimum value must be at least 6, and the value must be an even number.

**Function**

Sets the vertical scale of the graph display during Bathtub measurement.

**Example**

To set the scale during Error Rate type display to maximum E–4 and minimum E–10:

> :DISPlay:RESult:BATHtub:SCALe:VERTical 4,10

**Compatibility**

Incompatible with existing models.
Chapter 7  SCPI Commands

:DISPlay:RESult:BATHtub:SCALe:VERTical?

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;min&gt;,&lt;max&gt; = &lt;NR2 NUMERIC RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Same as the :DISPlay:RESult:BATHtub:SCALe:VERTical command parameter.</td>
</tr>
<tr>
<td>Function</td>
<td>Queries the vertical scale of the graph display during Bathtub measurement.</td>
</tr>
<tr>
<td>Example</td>
<td>&gt; :DISPlay:RESult:BATHtub:SCALe:VERTical?</td>
</tr>
<tr>
<td></td>
<td>&lt; 4,10</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Incompatible with existing models.</td>
</tr>
</tbody>
</table>
7.6 Automatic Measurement Commands

7.6.5.3 Measurement result and status query commands

Figure 7.6.5.3-1 Measurement result and status display

Table 7.6.5.3-1 Measurement result and status query commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>:SENSe:MEASure:BATHtub:DTIME?</td>
</tr>
</tbody>
</table>

:SENSe:MEASure:BATHtub:DTIME <type>

Parameter

- <type> = <CHARACTER PROGRAM DATA>
  - DTIMe: Displays the current date and time.
  - STARt: Displays the measurement start time.
  - ELAPsed: Displays the elapsed time based on the measurement period.

Function

Selects the Bathtub measurement time display type.

Example

To set the Bathtub measurement time display type to measurement start time (Start Time):

> :SENSe:MEASure:BATHtub:DTIME START

Compatibility

Incompatible with existing models.

:SENSe:MEASure:BATHtub:DTIME?

Response

- <type> = <CHARACTER RESPONSE DATA>
  - DTIM, STAR, ELAP

Function

Queries the Bathtub measurement time display type.

Example

> :SENSe:MEASure:BATHtub:DTIME?
  < STAR

Compatibility

Incompatible with existing models.
### :SENSe:MEASure:BATHtub:TIME:DTIMe?

**Response**

- `<year>` = `<NR1 NUMERIC RESPONSE DATA>`
- 2000 to 2036 2000 to 2036 year
- `<month>` = `<NR1 NUMERIC RESPONSE DATA>`
- 1 to 12 January to December
- `<day>` = `<NR1 NUMERIC RESPONSE DATA>`
- 1 to 31 1st to 31st
- `<hour>` = `<NR1 NUMERIC RESPONSE DATA>`
- 0 to 23 0 to 23 hours
- `<minute>` = `<NR1 NUMERIC RESPONSE DATA>`
- 0 to 59 0 to 59 minutes
- `<second>` = `<NR1 NUMERIC RESPONSE DATA>`
- 0 to 59 0 to 59 seconds

**Function**

Queries the current date and time during the Bathtub measurement.

**Example**

```plaintext
> :SENSe:MEASure:BATHtub:TIME:DTIMe?
< 2005,12,31,23,59,59
```

**Compatibility**

Incompatible with existing models.

### :SENSe:MEASure:BATHtub:TIME:STARt?

**Response**

- `<year>` = `<NR1 NUMERIC RESPONSE DATA>`
- 0, 2000 to 2036 2000 to 2036 year
- `<month>` = `<NR1 NUMERIC RESPONSE DATA>`
- 0, 1 to 12 January to December
- `<day>` = `<NR1 NUMERIC RESPONSE DATA>`
- 0, 1 to 31 1st to 31st
- `<hour>` = `<NR1 NUMERIC RESPONSE DATA>`
- 0 to 23 0 to 23 hours
- `<minute>` = `<NR1 NUMERIC RESPONSE DATA>`
- 0 to 59 0 to 59 minutes
- `<second>` = `<NR1 NUMERIC RESPONSE DATA>`
- 0 to 59 0 to 59 seconds

"0,0,0,0,0,0" is returned if there is no measurement start time data.

**Function**

Queries the measurement start time during Bathtub measurement.

**Example**

```plaintext
> :SENSe:MEASure:BATHtub:TIME:STARt?
< 2005,12,31,23,59,59
```

**Compatibility**

Incompatible with existing models.
### 7.6 Automatic Measurement Commands

<table>
<thead>
<tr>
<th>SCPI Commands</th>
<th>Response</th>
<th>Function</th>
<th>Example</th>
<th>Compatibility</th>
</tr>
</thead>
<tbody>
<tr>
<td>:SENSe:MEASure:BATHtub:TIME:ELAPsed?</td>
<td>&lt;day&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt; 0 to 99 0 to 99 days &lt;hour&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt; 0 to 23 0 to 23 hours &lt;minute&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt; 0 to 59 0 to 59 minutes &lt;second&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt; 0 to 59 0 to 59 seconds</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Queries the measurement elapsed time during Bathtub measurement.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>&gt; :SENSe:MEASure:BATHtub:TIME:ELAPsed?</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt; 31,23,59,59</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Incompatible with existing models.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

"0,0,0,0" is returned if there is no measurement elapsed time data.
Chapter 7  SCPI Commands

**:CALCulate:DATA:BATHtub? <string>**

Parameter  
<string> = <STRING PROGRAM DATA>

$result1$  For the contents of measurement data $result1$, see Table 7.6.5.3-2.

$result2$  For the contents of immediate data $result2$, see Table 7.6.5.3-3.

$result3$  For the contents of statistic data $result3$, see Table 7.6.5.3-4.

| Table 7.6.5.3-2  Phase vs. BER measurement result $result1$ |
|-----------------|-----------------|-----------------|
| **Items**       | **$result1$**   | **Response Format** |
| Phase vs. BER   | Number of measurements 1 to 1000 | "BATH:1"..."BATH:1000" | Form1 |
| Histogram       | "HIST:1"..."HIST:1000" | Form2 |

The measurement result for each Phase is delimited by a comma (,) and returned.

| Table 7.6.5.3-3  Phase vs. BER immediate data $result2$ |
|-----------------|-----------------|-----------------|
| **Items**       | **$result2$**   | **Response Format** |
| Optimum BER     | Number of measurements 1 to 1000 | "OPTBer:1"..."OPTBer:1000" | Form3 |
| Optimum Phase   | "OPTPh:1"..."OPTPh:1000" | Form4 |
| Total Jitter    | "TJ:1"..."TJ:1000" | Form4 |
| Deterministic Jitter | "DJ:1"..."DJ:1000" | Form4 |
| Random Jitter   | "RJ:1"..."RJ:1000" | Form4 |
| J2              | "J2:1"..."J2:1000" | Form4 |
| J9              | "J9:1"..."J9:1000" | Form4 |

| Table 7.6.5.3-4  Phase vs. BER measurement statistic data $result3$ |
|-----------------|-----------------|-----------------|
| **Items**       | **$result3$**   | **Response Format** |
| Total Data      | "TOTal" | Form5 |
| Valid Data      | "VALid" | Form5 |
| Total Jitter Mean | "TJMean" | Form4 |
| Deterministic Jitter Mean | "DJMean" | Form4 |
| Random Jitter Mean | "RJMean" | Form4 |
| J2 Mean         | "J2Mean" | Form4 |
| J9 Mean         | "J9Mean" | Form4 |
### 7.6 Automatic Measurement Commands

**Response**

\[<\text{string}> = <\text{STRING RESPONSE DATA}>\]

**Table 7.6.5.3-5 Response format**

<table>
<thead>
<tr>
<th>Form</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
</table>
| Form 1 Phase-BER   | "XXXX.XX, Y.YYYYE-YYY"     | XXXX.XX: Phase (mUI or PS)  
                       | "--------, --------"       | Y.YYYYE-YYY: Error Rate  
                       | When no data corresponds to the query. |
| Form 2 Histogram   | "XXXX.XX, Y.YYYYE-YYY"     | XXXX.XX: Phase (mUI or PS)  
                       | "--------, --------"       | Y.YYYYE-YYY: Error Count  
                       | When no data corresponds to the query. |
| Form 3 Fraction 1  | "X.XXXE-XXX"               | When 0.0000E–016 to 1.0000E000  
                       | "--------"                | When no data corresponds to the query. |
| Form 4 Fraction 2  | "XXXX.XX"                  | When –9999.99 to 9999.99  
                       | "--------"                | When no data corresponds to the query. |
| Form 5 Integer     | "XXXX"                     | When 0 to 9999  
                       | "----"                    | When no data corresponds to the query. |

**Function**

Queries the Phase vs. BER measurement result.

**Example**

To query the Phase vs. BER the measurement result for the 10th time:

```plaintext
> :CALCulate:DATA:BATHtub? "BATH:10"
< "0,1.2345E-003","20,1.2345E-004","40,1.2345E-005",
"60,1.2345E-006"
```

**Compatibility**

Incompatible with existing models.

---

### :CALCulate:DATA:BATHtub:STATus?

**Response**

\[<\text{string}> = <\text{STRING RESPONSE DATA}>\]

- "" When no alarm exists
- "Sync Loss" Sync. Loss
- "Clock Loss" Clock Loss
- "CR Unlock" CR Unlock
- "Out of range" Out of range
- "Frequency NG" Frequency NG
- "Illegal Error" Illegal Error

**Note:**

Displays any one of the alarms.

Illegal Error > Frequency NG > Clock Loss (CR Unlock) > Sync Loss > Out of range

**Function**

Queries the Q measurement status

**Example**

```plaintext
> :CALCulate:DATA:BATHtub:STATus?
< "Sync Loss"
```
7.6.5.4  File menu setting commands

Table 7.6.5.4-1  File menu setting commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
</table>

:SYSTem:MMEMory:BATHtub:RECall <file_name>

Parameter
- `<file_name>` = <STRING PROGRAM DATA>
- "<drv>:\[<dir>]<file>"
- `<drv>` = C, D, E, F
- `<dir>` = `<dir1>\<dir2>\...` (Omitted when root directory)
- `<file>` = File name

Function
Opens the Bathtub measurement result data.

Example
> :SYSTem:MMEMory:BATHtub:RECall "C:\Test\example"

Compatibility
Incompatible with existing models.

:SYSTem:MMEMory:BATHtub:STORe <file_name>,<data_type>,<file_type>

Parameter
- `<file_name>` = <STRING PROGRAM DATA>
- "<drv>:\[<dir>]<file>"
- `<drv>` = C, D, E, F
- `<dir>` = `<dir1>\<dir2>\...` (Omitted when root directory)
- `<file>` = File name
- `<data_type>` = <CHARACTER RESPONSE DATA>
- PSB  Phase vs. BER Result
- `<file_type>` = <CHARACTER RESPONSE DATA>
- BIN  Binary file
- CSV  CSV file
- TXT  Text file

Function
Saves the Bathtub measurement result data.

**Note:**

The settings will not be read from the saved file if the file name is changed.

Example
To save the Bathtub measurement result data by specifying the save destination (C:\Test), file name (example), and file format (CSV):
> :SYSTem:MMEMory:BATHtub:STORe "C:\Test\example",PSB,CSV

Compatibility
Incompatible with existing models.
### :SYSTem:PRINt:BATHtub <list>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;list&gt; = &lt;CHARACTER PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PSB</td>
</tr>
<tr>
<td></td>
<td>Phase vs. BER Result</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Function</th>
<th>Prints the Bathtub measurement result data.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example</td>
<td>&gt; :SYSTem:PRINt:BATHtub</td>
</tr>
</tbody>
</table>

| Compatibility   | Incompatible with existing models.        |
7.7 Stressed Eye Transmitter Commands

This section describes the MU181620A Stressed Eye Transmitter (hereinafter referred to as Transmitter) commands.

Before executing a setting/query command for the Transmitter, specify the slot number of the module to be remotely controlled, by using the :MODule:ID command. Refer to Section 7.1 “Common Commands” and 9.1 “Basic Operation of Device Messages” for how to specify a slot number with the :MODule:ID command.

7.7.1 Commands related to Optical tab

7.7.1.1 Optical Output setting commands

![Optical Output setting](image)

Figure 7.7.1.1-1 Optical Output setting (when Power Control is set to Power)
### Table 7.7.1.1-1 Optical Output setting commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>:TRANsmitter:OPTical:OUTPut?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:TRANsmitter:OPTical:OUTPut:BITRate?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:TRANsmitter:OPTical:OUTPut:RETiming?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:TRANsmitter:OPTical:OUTPut:TESTtype?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:TRANsmitter:OPTical:OUTPut:WAVelength?</td>
</tr>
</tbody>
</table>

**Figure 7.7.1.1-2 Optical Output setting**

*(when Power Control is set to Attenuation)*
### Table 7.7.1.1-1  Optical Output setting commands (Cont’d)

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>LED display</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>-TRANsmitter:OPTical:OUTPut:EXTinction:RATio?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-TRANsmitter:OPTical:OUTPut:EXTinction:OCALibration?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-TRANsmitter:OPTical:OUTPut:EXTinction:CALibration?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-TRANsmitter:OPTical:OUTPut:CONTrol?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-TRANsmitter:OPTical:OUTPut:CONTrol:UNIT?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-TRANsmitter:OPTical:OUTPut:POWer?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-TRANsmitter:OPTical:OUTPut:ATTenuator?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-TRANsmitter:OPTical:OUTPut:RELative?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-TRANsmitter:OPTical:OUTPut:RATTenuator?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-TRANsmitter:OPTical:OUTPut:ATTFactor?</td>
</tr>
<tr>
<td></td>
<td>LED display</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>-TRANsmitter:OPTical:OUTPut:AVERage?</td>
</tr>
</tbody>
</table>

**:TRANsmitter:OPTical:OUTPut <boolean>**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;boolean&gt; = &lt;BOOLEAN PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>OFF or 0: Optical output OFF</td>
</tr>
<tr>
<td></td>
<td>ON or 1: Optical output ON</td>
</tr>
</tbody>
</table>

**Function**

Sets optical output ON/OFF. Even when set to ON, the optical output is still Off, however, when the common mainframe output (:SOURce:OUTPut:ASET) is set to OFF.

**Example**

To set optical output ON/OFF:

> :TRANsmitter:OPTical:OUTPut ON

**Compatibility**

Incompatible with existing models.
### :TRANsmitter:OPTical:OUTPut?

**Response**

<boolean> = <NR1 NUMERIC RESPONSE DATA>

0 Optical output OFF
1 Optical output ON

**Function**

Queries the ON/OFF state of optical output.

**Example**

> :TRANsmitter:OPTical:OUTPut?
< 1

**Compatibility**

Incompatible with existing models.

### :TRANsmitter:OPTical:OUTPut:BITRate <numeric>

**Parameter**

<numeric> = <DECIMAL NUMERIC PROGRAM DATA>

0.100 to 12.500 0.100 to 12.500 Gbit/s / 0.001 Gbit/s Step

**Function**

Sets the input bit rate at Data Input.

**Example**

To set the bit rate to 10.205 Gbit/s:

> :TRANsmitter:OPTical:OUTPut:BITRate 10.205

**Compatibility**

Incompatible with existing models.

### :TRANsmitter:OPTical:OUTPut:BITRate?

**Response**

<numeric> = <NR2 NUMERIC RESPONSE DATA>

0.100 to 12.500 0.100 to 12.500 Gbit/s

**Function**

Queries the input bit rate at Data Input.

**Example**

> :TRANsmitter:OPTical:OUTPut:BITRate?
< 10.205

**Compatibility**

Incompatible with existing models.

### :TRANsmitter:OPTical:OUTPut:RETiming <boolean>

**Parameter**

<boolean> = <BOOLEAN PROGRAM DATA>

OFF or 0 Use of internal D-FF OFF

(when used with bit rate = 0.1 to 1 Gbit/s)

ON or 1 Use of internal D-FF ON

(when used with bit rate = 1 to 12.5 Gbit/s)

**Function**

Sets use of internal D-FF ON/OFF.

**Example**

To set use of internal D-FF ON:

> :TRANsmitter:OPTical:OUTPut:RETiming ON

**Compatibility**

Incompatible with existing models.
### :TRANsmitter:OPTical:OUTPut:RETiming?

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;boolean&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Use of internal D-FF is OFF</td>
</tr>
<tr>
<td>1</td>
<td>Use of internal D-FF is ON</td>
</tr>
</tbody>
</table>

**Function**
Queries ON/OFF state of use of internal D-FF.

**Example**
```
> :TRANsmitter:OPTical:OUTPut:RETiming?
< 1
```

**Compatibility**
Incompatible with existing models.

### :TRANsmitter:OPTical:OUTPut:TESTtype <route>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;route&gt; = &lt;CHARACTER PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>REF</td>
<td>Reference</td>
</tr>
<tr>
<td>STR</td>
<td>StressedEye (when any of MU181620A-x11, x12, and x13 is installed.)</td>
</tr>
</tbody>
</table>

**Function**
Sets the optical output route.

**Example**
To set the optical output route to Reference:
```
> :TRANsmitter:OPTical:OUTPut:TESTtype REFerence
```

**Compatibility**
Incompatible with existing models.

### :TRANsmitter:OPTical:OUTPut:TESTtype?

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;route&gt; = &lt;CHARACTER RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>REF, STR</td>
<td></td>
</tr>
</tbody>
</table>

**Function**
Queries the optical output route.

**Example**
```
> :TRANsmitter:OPTical:OUTPut:TESTtype?
< REF
```

**Compatibility**
Incompatible with existing models.

### :TRANsmitter:OPTical:OUTPut:WAVelength <length>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;length&gt; = &lt;CHARACTER PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>W_1310</td>
<td>1310 nm (when MU181620A-x01, x03, x11, or x13 is installed.)</td>
</tr>
<tr>
<td>W_1550</td>
<td>1550 nm (when MU181620A-x02, x03, x11, or x13 is installed.)</td>
</tr>
</tbody>
</table>

**Function**
Sets the optical output wavelength.

**Example**
To set the optical output wavelength to 1310 nm:
```
> :TRANsmitter:OPTical:OUTPut:WAVelength W_1310
```

**Compatibility**
Incompatible with existing models.
7.7 Stressed Eye Transmitter Commands

:TRANsmitter:OPTical:OUTPut:WAVelength?

Response: 

\[ <\text{length}> = <\text{CHARACTER RESPONSE DATA}> \]

W_1310, W_1550

Function: Queries the optical output wavelength.

Example: 

> :TRANsmitter:OPTical:OUTPut:WAVelength?

< W_1310

Compatibility: Incompatible with existing models.

:TRANsmitter:OPTical:OUTPut:EXTinction:OVERrange?

Response: 

\[ <\text{numeric}> = <\text{NR1 NUMERIC RESPONSE DATA}> \]

0: Within specifications (LED: green)
1: Beyond specifications (LED: red)

Table 7.7.1.1-2 Specification Conformance

<table>
<thead>
<tr>
<th>Wavelength[nm]</th>
<th>Test Type</th>
<th>Extinction Ratio[db]</th>
<th>Specification Conformance</th>
</tr>
</thead>
<tbody>
<tr>
<td>1310 Reference</td>
<td>4.0 to 10.0</td>
<td>0: Within specifications</td>
<td></td>
</tr>
<tr>
<td>1310 Stressed Eye</td>
<td>2.0 to 3.9, 10.1 to 11.0</td>
<td>1: Beyond specifications</td>
<td></td>
</tr>
<tr>
<td>1310 Stressed Eye</td>
<td>1.0 to 1.9, 6.1 to 7.0</td>
<td>1: Beyond specifications</td>
<td></td>
</tr>
<tr>
<td>1550 Reference</td>
<td>6.0 to 10.0</td>
<td>0: Within specifications</td>
<td></td>
</tr>
<tr>
<td>1550 Stressed Eye</td>
<td>2.0 to 5.9, 10.1 to 11.0</td>
<td>1: Beyond specifications</td>
<td></td>
</tr>
<tr>
<td>1550 Stressed Eye</td>
<td>1.0 to 1.9, 5.1 to 7.0</td>
<td>1: Beyond specifications</td>
<td></td>
</tr>
</tbody>
</table>

Function: Queries whether the current extinction ratio of the optical output is within or beyond specifications.

Example: 

> :TRANsmitter:OPTical:OUTPut:EXTinction:OVERrange?

< 1

Compatibility: Incompatible with existing models.

:TRANsmitter:OPTical:OUTPut:EXTinction:RATio <numeric>

Parameter: 

\[ <\text{numeric}> = <\text{DECIMAL NUMERIC PROGRAM DATA}> \]

When Test Type is set to Reference: 

2.0 to 11.0 2.0 to 11.0 dB / 0.1 dB Step

When Test Type is set to Stressed Eye: 

1.0 to 7.0 1.0 to 7.0 dB / 0.1 dB Step

Function: Sets the extinction ratio of optical output.

Example: To set the extinction ratio of optical output to 5.5 dB: 

> :TRANsmitter:OPTical:OUTPut:EXTinction:RATio 5.5

Compatibility: Incompatible with existing models.
### Chapter 7  SCPI Commands

#### :TRANsmitter:OPTical:OUTPut:EXTinction:RATio?

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;numeric&gt; = &lt;NR2 NUMERIC RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1.0 to 11.0 1.0 to 11.0 dB</td>
</tr>
</tbody>
</table>

**Function**
Queries the extinction ratio of optical output.

**Example**
> :TRANsmitter:OPTical:OUTPut:EXTinction:RATio?
< 5.5

**Compatibility**
Incompatible with existing models.

#### :TRANsmitter:OPTical:OUTPut:EXTinction:OCALibration <boolean>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;boolean&gt; = &lt;BOOLEAN PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>OFF or 0</td>
<td>Offset OFF</td>
</tr>
<tr>
<td>ON or 1</td>
<td>Offset ON</td>
</tr>
</tbody>
</table>

**Function**
Sets whether to enable the offset value such that the monitored value display of the optical output extinction ratio matches the display value on the sampling oscilloscope.

**Example**
To set the monitored value offset of optical output extinction ratio to ON:
> :TRANsmitter:OPTical:OUTPut:EXTinction:OCALibration ON

**Compatibility**
Incompatible with existing models.

#### :TRANsmitter:OPTical:OUTPut:EXTinction:OCALibration?

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;boolean&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0 Offset OFF</td>
</tr>
<tr>
<td></td>
<td>1 Offset ON</td>
</tr>
</tbody>
</table>

**Function**
Queries whether the monitored value offset of optical output extinction ratio is enabled.

**Example**
> :TRANsmitter:OPTical:OUTPut:EXTinction:OCALibration?
< 1

**Compatibility**
Incompatible with existing models.
7.7 Stressed Eye Transmitter Commands

**:TRANsmitter:OPTical:OUTPut:EXTinction:CALibration** <numeric>

Parameter

- <numeric> = <DECIMAL NUMERIC PROGRAM DATA>
- −3.0 to 3.0
- −3.0 to 3.0 dB / 0.1 dB Step

wherein, Extinction Ratio + Calibration (offset) ≥ 0.1

Valid only when the monitored value offset of extinction ratio is set to ON.

Function

Sets the monitored value offset of the optical output extinction ratio.

Cannot be set when offset is set to OFF.

Example

To set the monitored value offset of optical output extinction ratio to −0.5 dB:

```
> :TRANsmitter:OPTical:OUTPut:EXTinction:CALibration -0.5
```

Compatibility

Incompatible with existing models.

**:TRANsmitter:OPTical:OUTPut:EXTinction:CALibration?**

Response

- <numeric> = <NR2 NUMERIC RESPONSE DATA>
- −3.0 to 3.0
- −3.0 to 3.0 dB

Function

Queries the monitored value offset of the optical output extinction ratio.

Valid only when the monitored value offset of extinction ratio is set to ON.

Example

```
> :TRANsmitter:OPTical:OUTPut:EXTinction:CALibration?
< -0.5
```

Compatibility

Incompatible with existing models.

**:TRANsmitter:OPTical:OUTPut:EXTinction:MONitor?**

Response

- <string> = <STRING RESPONSE DATA>

Table 7.7.1.1-3 Response

<table>
<thead>
<tr>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;XX.X&quot;</td>
<td>0.1 to 14.0 dB / 0.1 dB Step</td>
</tr>
<tr>
<td>&quot;-----&quot;</td>
<td>When no data corresponds to the query</td>
</tr>
</tbody>
</table>

Function

Queries the extinction ratio monitor value obtained by adding the offset value to the optical output extinction ratio.

Example

```
> :TRANsmitter:OPTical:OUTPut:EXTinction:MONitor?
< "5.0"
```

Compatibility

Incompatible with existing models.
Chapter 7  SCPI Commands

:TRANsmitter:OPTical:OUTPut:CONTrol <power>

Parameter  
<power> = <CHARACTER PROGRAM DATA>
    OFF OFF
    POW Power
    ATTenuation Attenuation

Function  
Sets the control method for optical output power.

Example  
To set the control method for optical output power to Power:
    > :TRANsmitter:OPTical:OUTPut:CONTrol POWer

Compatibility  
Incompatible with existing models.

:TRANsmitter:OPTical:OUTPut:CONTrol?

Response  
<power> = <CHARACTER RESPONSE DATA>
    OFF, POW, ATT

Function  
Queries the control method for optical output power.

Example  
> :TRANsmitter:OPTical:OUTPut:CONTrol?
    < POW

Compatibility  
Incompatible with existing models.

:TRANsmitter:OPTical:OUTPut:CONTrol:UNIT <unit>

Parameter  
<unit> = <CHARACTER PROGRAM DATA>
    DBM Unit: dBm
    UW Unit: μW

Function  
Sets display unit for optical output power.

Example  
To set display unit for optical output power to dBm:
    > :TRANsmitter:OPTical:OUTPut:CONTrol:UNIT DBM

Compatibility  
Incompatible with existing models.

:TRANsmitter:OPTical:OUTPut:CONTrol:UNIT?

Response  
<unit> = <CHARACTER RESPONSE DATA>
    DBM, UW

Function  
Queries display unit for optical output power.

Example  
> :TRANsmitter:OPTical:OUTPut:CONTrol:UNIT?
    < DBM

Compatibility  
Incompatible with existing models.
### 7.7 Stressed Eye Transmitter Commands

#### :TRANsmitter:OPTical:OUTPut:POWer <numeric>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;numeric&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>When Wavelength is set to 1310 nm while Unit is set to dBm:</td>
</tr>
<tr>
<td></td>
<td>-20.00 to -4.00 -20.00 to -4.00 dBm / 0.01 dBm Step</td>
</tr>
<tr>
<td></td>
<td>When Wavelength is set to 1310 nm while Unit is set to µW:</td>
</tr>
<tr>
<td></td>
<td>10 to 398 10 to 398 µW / 1 µW Step</td>
</tr>
<tr>
<td></td>
<td>When Wavelength is set to 1550 nm while Unit is set to dBm:</td>
</tr>
<tr>
<td></td>
<td>-20.00 to -2.00 -20.00 to -2.00 dBm / 0.01 dBm Step</td>
</tr>
<tr>
<td></td>
<td>When Wavelength is set to 1550 nm while Unit is set to µW:</td>
</tr>
<tr>
<td></td>
<td>10 to 630 10 to 630 µW / 1 µW Step</td>
</tr>
</tbody>
</table>

**Function**

Sets the optical output power.

**Example**

To set the optical output power to -3.27 dBm:

> :TRANsmitter:OPTical:OUTPut:POWer -3.27

**Compatibility**

Incompatible with existing models.

#### :TRANsmitter:OPTical:OUTPut:POWer?

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;numeric&gt; = &lt;NR2 NUMERIC RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>When Unit is set to dBm:</td>
</tr>
<tr>
<td></td>
<td>-20.00 to -2.00 -20.00 to -2.00 dBm</td>
</tr>
<tr>
<td></td>
<td>When Unit is set to µW:</td>
</tr>
<tr>
<td></td>
<td>10 to 630 10 to 630 µW</td>
</tr>
</tbody>
</table>

**Function**

Queries the optical output power.

Valid only when the control method of optical output power is set to Power.

**Example**

> :TRANsmitter:OPTical:OUTPut:POWer?

< -3.27

**Compatibility**

Incompatible with existing models.
Chapter 7  SCPI Commands

:TRANsmitter:OPTical:OUTPut:ATTenuator <numeric>

Parameter  

<numeric> = <DECIMAL NUMERIC PROGRAM DATA>

When Wavelength is set to 1310 nm:

0.00 to 16.00  0.00 to 16.00 dB / 0.01 dB Step

When Wavelength is set to 1550 nm:

0.00 to 18.00  0.00 to 18.00 dB / 0.01 dB Step

Function  
Sets the optical output attenuation.

Valid only when the control method of optical output power is set to Attenuation.

Example  
To set the optical output attenuation to 12.34 dB:

> :TRANsmitter:OPTical:OUTPut:ATTenuator 12.34

Compatibility  
Incompatible with existing models.

:TRANsmitter:OPTical:OUTPut:ATTenuator?

Response  

<numeric> = <NR2 NUMERIC RESPONSE DATA>

0.00 to 18.00  0.00 to 18.00 dB

Function  
Queries the optical output attenuation.

Valid only when the control method of optical output power is set to Attenuation.

Example  
> :TRANsmitter:OPTical:OUTPut:ATTenuator?

< 12.34

Compatibility  
Incompatible with existing models.

:TRANsmitter:OPTical:OUTPut:RELative <boolean>

Parameter  

<boolean> = <BOOLEAN PROGRAM DATA>

OFF or 0  Attenuation reference OFF
ON or 1  Attenuation reference ON

Function  
Sets optical output attenuation reference ON/OFF.

Setting to ON enables attenuation reference settings.

Valid only when the control method of optical output power is set to Attenuation.

Example  
To set optical output attenuation reference to ON:

> :TRANsmitter:OPTical:OUTPut:RELative ON

Compatibility  
Incompatible with existing models.
### :TRANsmitter:OPTical:OUTPut:RELative?

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;boolean&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Attenuation reference OFF</td>
</tr>
<tr>
<td>1</td>
<td>Attenuation reference ON</td>
</tr>
</tbody>
</table>

**Function**
Queries ON/OFF state of optical output attenuation reference.
Valid only when the control method of optical output power is set to Attenuation.

**Example**

```
> :TRANsmitter:OPTical:OUTPut:RELative?
< 1
```

**Compatibility**
Incompatible with existing models.

### :TRANsmitter:OPTical:OUTPut:RATTenuator <numeric>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;numeric&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>When Wavelength is set to 1310 nm:</td>
</tr>
<tr>
<td></td>
<td>−16.00 to 16.00 −16.00 to 16.00 dB / 0.01 dB Step</td>
</tr>
<tr>
<td></td>
<td>When Wavelength is set to 1550 nm:</td>
</tr>
<tr>
<td></td>
<td>−18.00 to 18.00 −18.00 to 18.00 dB / 0.01 dB Step</td>
</tr>
</tbody>
</table>

**Function**
Sets the optical output attenuation reference value.
Valid only when the control method of optical output power is set to Attenuation.

**Example**

```
To set the optical output attenuation reference to 7.66 dB:
> :TRANsmitter:OPTical:OUTPut:RATTenuator 7.66
```

**Compatibility**
Incompatible with existing models.

### :TRANsmitter:OPTical:OUTPut:RATTenuator?

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;numeric&gt; = &lt;NR2 NUMERIC RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>−18.00 to 18.00 −18.00 to 18.00 dB</td>
</tr>
</tbody>
</table>

**Function**
Queries the optical output attenuation reference value.
Valid only when the control method of optical output power is set to Attenuation.

**Example**

```
> :TRANsmitter:OPTical:OUTPut:RATTenuator?
< 7.66
```

**Compatibility**
Incompatible with existing models.
### :TRANsmitter:OPTical:OUTPut:ATTFactor <numeric>

| Parameter | <numeric> = <DECIMAL NUMERIC PROGRAM DATA>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Function</td>
<td>Sets the attenuation when an optical output signal passes through an external attenuator.</td>
</tr>
<tr>
<td>Example</td>
<td>&gt; :TRANsmitter:OPTical:OUTPut:ATTFactor 21.16</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Incompatible with existing models.</td>
</tr>
</tbody>
</table>

### :TRANsmitter:OPTical:OUTPut:ATTFactor? 

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;numeric&gt; = &lt;NR2 NUMERIC RESPONSE DATA&gt; 0.00 to 40.00 0.00 to 40.00 dB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function</td>
<td>Queries the attenuation of the external attenuator.</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Incompatible with existing models.</td>
</tr>
</tbody>
</table>

### :TRANsmitter:OPTical:OUTPut:PMONitor:CALCulation? 

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;numeric&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt; 0 Average value calculation is not being performed (LED off) 1 Average value is being calculated (LED on)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function</td>
<td>Queries whether the Power Monitor value is being averaged.</td>
</tr>
<tr>
<td>Example</td>
<td>&gt; :TRANsmitter:OPTical:OUTPut:PMONitor:CALCulation? &lt; 1</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Incompatible with existing models.</td>
</tr>
</tbody>
</table>
### 7.7 Stressed Eye Transmitter Commands

#### :TRANsmitter:OPTical:OUTPut:PMONitor?

Response <string> = <STRING RESPONSE DATA>

<table>
<thead>
<tr>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;XXX.XX&quot;</td>
<td>XXX.XX dBm</td>
</tr>
<tr>
<td>&quot;XXX&quot;</td>
<td>XXX μW</td>
</tr>
<tr>
<td>&quot;0&gt;&quot;</td>
<td>When Unit is set to dBm while the value is larger than 0.0 dBm</td>
</tr>
<tr>
<td>&quot;&lt;-18&quot;</td>
<td>When Unit is set to dBm while the value is less than –18.0 dBm</td>
</tr>
<tr>
<td>&quot;&lt;15&quot;</td>
<td>When Unit is set to μW while the value is less than 15 μW</td>
</tr>
<tr>
<td>&quot;1000&gt;&quot;</td>
<td>When Unit is set to μW while the value is larger than 1000 μW</td>
</tr>
<tr>
<td>&quot;.....&quot;</td>
<td>When no data corresponds to the query</td>
</tr>
</tbody>
</table>

Response data unit is set by :TRANsmitter:OPTical:OUTPut:CONTrol:UNIT. An optical power value including the attenuation of the external attenuator that is set by TRANsmitter:OPTical:OUTPut:ATTFactor is returned.

**Function**
Queries the optical output power value at the optical output connector.

**Example**
> :TRANsmitter:OPTical:OUTPut:PMONitor?

< "-11.11"

**Compatibility**
Incompatible with existing models.

---

#### :TRANsmitter:OPTical:OUTPut:AVERage <average>

**Parameter**
<average> = <CHARACTER PROGRAM DATA>

- OFF: No averaging
- A_2: Averages data of 2 times of output.
- A_5: Averages data of 5 times of output.
- A_10: Averages data of 10 times of output.
- A_20: Averages data of 20 times of output.
- A_50: Averages data of 50 times of output.
- A_100: Averages data of 100 times of output.

**Function**
Sets the number of averaging times for optical output power.

**Example**
To set the number of averaging times for optical output power to 100:
> :TRANsmitter:OPTical:OUTPut:AVERage A_100

**Compatibility**
Incompatible with existing models.

---

#### :TRANsmitter:OPTical:OUTPut:AVERage?

Response <average> = <CHARACTER RESPONSE DATA>

OFF, A_2, A_5, A_10, A_20, A_50, A_100

**Function**
Queries the number of averaging times for optical output power.

**Example**
> :TRANsmitter:OPTical:OUTPut:AVERage?

< A_100

**Compatibility**
Incompatible with existing models.
7.8 MUX Commands

This section describes the setting and query commands for the MU182020A 25 Gbit/s 1ch MUX, and MU182021A 25 Gbit/s 2ch MUX. The commands related to each tab are described in separate sections below. Before executing a setting/query command for the MU181040A, specify the slot number of the module to be remotely controlled, by using the :MODule:ID command. Refer to Section 7.1 “Common Commands” and 9.1 "Basic Operation of Device Messages" for how to specify a slot number with the :MODule:ID command. However, unless there is a special item, see the MU181021A.

7.8.1 Commands related to Data Interface tab

7.8.1.1 Data Interface setting field

![Figure 7.8.1.1-1 Data Interface setting field](image)
### 7.8 MUX Commands

#### Level Guard Setting

![Level Guard Setting](image)

**Figure 7.8.1.1-2 Level Guard setting field**

#### Table 7.8.1.1-1 Data Interface setting commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>:MUX:DATA:OUTPut?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:MUX:OUTPut:OFFSet?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:MUX:DATA:INTERface:TRACking?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:MUX:DATA:TRACking?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:MUX:DATA:LEVGuard?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:MUX:DATA:LEVel?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:MUX:DATA:AMPLitude?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:MUX:DATA:AOFFset?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:MUX:DATA:OFFSet?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:MUX:DATA:ATTFactor?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:MUX:DATA:CPOint?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:MUX:DATA:HPJitter?</td>
</tr>
<tr>
<td>[14]</td>
<td>Delay (mUI settings)</td>
<td>:MUX:DATA:UIPadjust</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:MUX:DATA:UIPadjust?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:MUX:DATA:PADJust?</td>
</tr>
</tbody>
</table>
Table 7.8.1.1-1 Data Interface setting commands (Cont’d)

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>:MUX:DATA:RELative?</td>
</tr>
<tr>
<td>[18]</td>
<td>No label (Settings of Relative values)</td>
<td>:MUX:DATA:RDElay</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:MUX:DATA:RDElay?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:MUX:DATA:JINPut?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:MUX:DATA:LIMitter:AMPLitude?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:MUX:DATA:LIMitter:OFFSet?</td>
</tr>
<tr>
<td>[22]</td>
<td>Unit Offset</td>
<td>:MUX:DATA:UDELay:OFFSet</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MUX:DATA:UDELay:OFFSet?</td>
</tr>
</tbody>
</table>

:MUX:DATA:OUTPut <boolean>[,<interface>]

Parameter
- `<boolean>` = `<BOOLEAN PROGRAM DATA>`
  OFF or 0, ON or 1
- `<interface>` = `<DECIMAL NUMERIC PROGRAM DATA>`
  1 to 2 Data Interface for Data1, and Data2
  If `<Interface>` is omitted, the default is 1.

Function
Sets Data/XData outputs ON/OFF.

Example
To set Data/Xdata output of Data1 Interface to ON:
> :MUX:DATA:OUTPut ON,1

Compatibility
Not backwards compatible


Parameter
- `<interface>` = `<DECIMAL NUMERIC PROGRAM DATA>`
  1 to 2 Data Interface for Data1, and Data2
  If `<Interface>` is omitted, the default is 1.

Response
- `<numeric>` = `<NR1 NUMERIC RESPONSE DATA>`
  0 Data/XData output is OFF
  1 Data/XData output is ON

Function
Queries whether Data/XData output is ON or OFF.

Example
To query whether Data/XData output of Data1 Interface is ON or OFF:
> :MUX:DATA:OUTPut? 1
< 1

Compatibility
Not backwards compatible
### :MUX:OUTPut:OFFSet <offset>

**Parameter**  
<offset> = <CHARACTER PROGRAM DATA>  
- VOH: High level  
- VTH: High level and Low level center value  
- VOL: Low level  

**Function**  
Sets offset reference value for Data/XData and Clock/XClock outputs.

**Example**  
To set offset reference value for Data/XData and Clock/XClock outputs to VOH:

```
> :MUX:OUTPut:OFFSet VOH
```

**Compatibility**  
Not backwards compatible

### :MUX:OUTPut:OFFSet?

**Response**  
<offset> = <CHARACTER RESPONSE DATA>  
- VOH, VTH, VOL

**Function**  
Queries offset reference value for Data/XData.

**Example**  
To query offset reference value for Data/XData for Data1Interface:

```
> :MUX:OUTPut:OFFSet?
< VOH
```

**Compatibility**  
Not backwards compatible

### :MUX:OUTPut:BMONitor? [<interface>]

**Parameter**  
<interface> = <DECIMAL NUMERIC PROGRAM DATA>  
- 1 to 2: Data Interface for Data1, and Data2  
  - If [<Interface>] is omitted, the default is 1.

**Response**  
<string> = <STRING RESPONSE DATA>

**Table 7.8.1-2  Response format**

<table>
<thead>
<tr>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;XX.XXXX&quot;</td>
<td>XX.XXXX Gbit/s</td>
</tr>
<tr>
<td>&quot;......&quot;</td>
<td>When no data corresponds to the query</td>
</tr>
</tbody>
</table>

**Function**  
Queries BitRate.

**Example**  
To query BitRate of Data1Interface:

```
> :MUX:OUTPut:BMONitor? 1
< "28.00000"
```

**Compatibility**  
Not backwards compatible
<table>
<thead>
<tr>
<th>:<strong>MUX:DATA:INTerface:TRACking</strong> &lt;boolean&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Parameter</strong></td>
</tr>
<tr>
<td>&lt;boolean&gt; = &lt;BOOLEAN PROGRAM DATA&gt;</td>
</tr>
<tr>
<td>OFF or 0, ON or 1</td>
</tr>
<tr>
<td><strong>Function</strong></td>
</tr>
<tr>
<td>Sets shared settings for Data Interface ON/OFF.</td>
</tr>
<tr>
<td><strong>Example</strong></td>
</tr>
<tr>
<td>To set shared settings for Data Interface to ON:</td>
</tr>
<tr>
<td>&gt; :MUX:DATA:INTerface:TRACking ON</td>
</tr>
<tr>
<td><strong>Compatibility</strong></td>
</tr>
<tr>
<td>Not backwards compatible</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>:<strong>MUX:DATA:INTerface:TRACking?</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Response</strong></td>
</tr>
<tr>
<td>&lt;boolean&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt;</td>
</tr>
<tr>
<td>0  Shared settings for Data Interface are OFF</td>
</tr>
<tr>
<td>1  Shared settings for Data Interface are ON</td>
</tr>
<tr>
<td><strong>Function</strong></td>
</tr>
<tr>
<td>Queries shared ON/OFF settings for Data Interface.</td>
</tr>
<tr>
<td><strong>Example</strong></td>
</tr>
<tr>
<td>To query shared ON/OFF settings for Data Interface:</td>
</tr>
<tr>
<td>&gt; :MUX:DATA:INTerface:TRACking?</td>
</tr>
<tr>
<td>&lt; 1</td>
</tr>
<tr>
<td><strong>Compatibility</strong></td>
</tr>
<tr>
<td>Not backwards compatible</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>:<strong>MUX:DATA:TRACking</strong> &lt;boolean&gt;[,&lt;interface&gt;]</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Parameter</strong></td>
</tr>
<tr>
<td>&lt;boolean&gt; = &lt;BOOLEAN PROGRAM DATA&gt;</td>
</tr>
<tr>
<td>OFF or 0, ON or 1</td>
</tr>
<tr>
<td>&lt;interface&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
</tr>
<tr>
<td>1 to 2</td>
</tr>
<tr>
<td>Data Interface for Data1, and Data2</td>
</tr>
<tr>
<td>If [&lt;Interface&gt;] is omitted, the default is 1.</td>
</tr>
<tr>
<td><strong>Function</strong></td>
</tr>
<tr>
<td>Sets shared settings for Data/XData output ON or OFF.</td>
</tr>
<tr>
<td><strong>Example</strong></td>
</tr>
<tr>
<td>To set shared settings for Data/XData of Data1Interface to ON:</td>
</tr>
<tr>
<td>&gt; :MUX:DATA:TRACking ON,1</td>
</tr>
<tr>
<td><strong>Compatibility</strong></td>
</tr>
<tr>
<td>Not backwards compatible</td>
</tr>
</tbody>
</table>
### :MUX:DATA:TRACking? [<interface>]

**Parameter**
- `<interface>` = `<DECIMAL NUMERIC PROGRAM DATA>`
  - `1` to `2` Data Interface for Data1, and Data2
  - If `<interface>` is omitted, the default is `1`.

**Response**
- `<numeric>` = `<NR1 NUMERIC RESPONSE DATA>`
  - `0` Shared settings for Data/XData output are OFF
  - `1` Shared settings for Data/XData output are ON

**Function**
Queries shared settings for Data/XData output.

**Example**
To query shared settings for Data/XData output for Data1Interface:

```
> :MUX:DATA:TRACking? 1
< 1
```

**Compatibility**
Not backwards compatible

### :MUX:DATA:LEVGuard <boolean>[,<interface>]

**Parameter**
- `<boolean>` = `<BOOLEAN PROGRAM DATA>`
  - OFF or `0`, ON or `1`
- `<interface>` = `<DECIMAL NUMERIC PROGRAM DATA>`
  - `1` to `2` Data Interface for Data1, and Data2
  - If `<interface>` is omitted, the default is `1`.

**Function**
Sets Data/XData output range limit ON/OFF.

**Example**
To set range limit for Data/XData output of Data1Interface to ON:

```
> :MUX:DATA:LEVGuard ON,1
```

**Compatibility**
Not backwards compatible

### :MUX:DATA:LEVGuard? [interface]

**Parameter**
- `<interface>` = `<DECIMAL NUMERIC PROGRAM DATA>`
  - `1` to `2` Data Interface for Data1, and Data2
  - If `<interface>` is omitted, the default is `1`.

**Response**
- `<numeric>` = `<NR1 NUMERIC RESPONSE DATA>`
  - `0` Data/XData output range limit is OFF
  - `1` Data/XData output range limit is ON

**Function**
Queries Data/XData output range limit.

**Example**
To query Data/XData output range limit for Data1Interface:

```
> :MUX:DATA:LEVGuard? 1
< 1
```

**Compatibility**
Not backwards compatible
**:MUX:DATA:LEVel <port>,<level>[,<interface>]**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;port&gt; = &lt;CHARACTER PROGRAM DATA&gt;</td>
<td>DATA, XDAta</td>
</tr>
<tr>
<td>&lt;level&gt; = &lt;CHARACTER PROGRAM DATA&gt;</td>
<td>VARiable Variable</td>
</tr>
<tr>
<td>&lt;interface&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
<td>1 to 2 Data Interface for Data1, and Data2</td>
</tr>
</tbody>
</table>

**Function**
Sets data output level for specific port.

**Example**
To set data output level for XData of Data1Interface to NECL level:
> :MUX:DATA:LEVel XDAta,NECL,1

**Compatibility**
Not backwards compatible
7.8  MUX Commands

:MUX:DATA:LEVel? <port>[,<interface>]

Parameter

- `<port> = <CHARACTER PROGRAM DATA>`
  - DATA, XDATa
- `<interface> = <DECIMAL NUMERIC PROGRAM DATA>`
  - 1 to 2  Data Interface for Data1, and Data2
  - If [<Interface>] is omitted, the default is 1.

Response

- `<level> = <CHARACTER RESPONSE DATA>`
  - VAR, PCML, NCML, SCFL, NECL, LVP, LVDS400

Function

Queries data output level for specific port.

Example

To query data output level for XData of Data1Interface:

```
> :MUX:DATA:LEVel? XDATa,1
< NECL
```

Compatibility

Not backwards compatible

:MUX:DATA:AMPLitude <port>,<numeric>[,<interface>]

Parameter

- `<port> = <CHARACTER PROGRAM DATA>`
  - DATA, XDATa
- `<numeric> = <DECIMAL NUMERIC PROGRAM DATA>`
  - 0.250 to 1.750 0.250 to 1.750 V/0.002 V step
    - (MU182020A-x10, MU182021A-x10)
  - 0.500 to 2.500 0.500 to 2.500 V/0.002 V step
    - (MU182020A-x11, MU182021A-x11)
  - 0.500 to 2.000 0.500 to 2.000 V/0.002 V step
    - (MU182020A-x12, MU182021A-x12)
  - 0.500 to 3.500 0.500 to 3.500 V/0.002 V step
    - (MU182020A-x13, MU182021A-x13)
- `<interface> = <DECIMAL NUMERIC PROGRAM DATA>`
  - 1 to 2  Data Interface for Data1, and Data2
  - If [<Interface>] is omitted, the default is 1.

Function

Sets data output amplitude for specific port.

Example

To set output amplitude for Data of Data1Interface to 1.000 V:

```
> :MUX:DATA:AMPLitude DATA,1.000,1
```

Compatibility

Not backwards compatible
Chapter 7  SCPI Commands

:MUX:DATA:AMPLitude? <port>[,<interface>]

Parameter
<port> = <CHARACTER PROGRAM DATA>
DATA, XDATa
<interface> = <DECIMAL NUMERIC PROGRAM DATA>
1 to 2 Data Interface for Data1, and Data2
If [<interface>] is omitted, the default is 1.

Response
<numerical> = <NR2 NUMERIC RESPONSE DATA>

Function
Queries data output amplitude for specific port.

Example
To query data output amplitude for Data of Data1Interface:
> :MUX:DATA:AMPLitude? DATA,1
< 1.000

Compatibility
Not backwards compatible

:MUX:DATA:AOFFSET <boolean>[,<interface>]

Parameter
<boolean> = <BOOLEAN PROGRAM DATA>
OFF or 0 Offset OFF (DC output)
ON or 1 Offset ON (AC output)
<interface> = <DECIMAL NUMERIC PROGRAM DATA>
1 to 2 Data Interface for Data1, and Data2
If [<Interface>] is omitted, the default is 1.

Function
Sets Data output offset ON/OFF.

Example
To set Data output offset for Data1Interface to ON:
> :MUX:DATA:AOFFSET ON,1

Compatibility
Not backwards compatible


Parameter
<interface> = <DECIMAL NUMERIC PROGRAM DATA>
1 to 2 Data Interface for Data1, and Data2
If [<Interface>] is omitted, the default is 1.

Response
<numerical> = <NR1 NUMERIC RESPONSE DATA>
0 Offset OFF
1 Offset ON

Function
Queries whether Data output offset ON or OFF.

Example
To query Data output offset for Data1Interface:
> :MUX:DATA:AOFFSET? 1
< 1

Compatibility
Not backwards compatible
### 7.8 MUX Commands

:[MUX:DATA:OFFSet] <port>,<numeric>[,<interface>]

**Parameter**
- `<port>` = <CHARACTER PROGRAM DATA>
  - DATA, XDAta
- `<numeric>` = <DECIMAL NUMERIC PROGRAM DATA>
  - –2.000 to 3.300 Voh/0.001V step
    - (MU182020A-x10, x11, x12, x13,
      MU182021A-x10, x11, x12, x13)
  - –2.875 to 3.175 Vth/0.001V step
    - (MU182020A-x10, MU182021A-x10)
  - –3.000 to 3.050 Vth/0.001V step
    - (MU182020A-x11, MU182021A-x11)
  - –3.000 to 3.050 Vth/0.001V step
    - (MU182020A-x12, MU182021A-x12)
  - –3.000 to 3.050 Vth/0.001V step
    - (MU182020A-x13, MU182021A-x13)
  - –3.750 to 3.050 Vol/0.001V step
    - (MU182020A-x10, MU182021A-x10)
  - –4.000 to 2.800 Vol/0.001V step
    - (MU182020A-x11, x12, x13,
      MU182021A-x11, x12, x13)
- `<interface>` = <DECIMAL NUMERIC PROGRAM DATA>
  - 1 to 2 Data Interface for Data1, and Data2
  - If `<interface>` is omitted, the default is 1.

**Function**
Sets data output offset for specific port.

**Example**
To set data output offset for Data of Data1Interface to 1.000 Voh:
> :MUX:DATA:OFFSet DATA,1.000,1

**Compatibility**
Not backwards compatible

:[MUX:DATA:OFFSet?] <port>[,<interface>]

**Parameter**
- `<port>` = <CHARACTER PROGRAM DATA>
  - DATA, XDAta
- `<interface>` = <DECIMAL NUMERIC PROGRAM DATA>
  - 1 to 2 Data Interface for Data1, and Data2
  - If `<interface>` is omitted, the default is 1.

**Response**
- `<numeric>` = <NR2 NUMERIC RESPONSE DATA>

**Function**
Queries data output offset for specific port.

**Example**
To query data output offset for Data of Data1Interface:
> :MUX:DATA:OFFSet? DATA,1
< 1.000

**Compatibility**
Not backwards compatible
Chapter 7  SCPI Commands

:MUX:DATA:ATTFactor <port>,<numeric>[,<interface>]

Parameter
- <port> = <CHARACTER PROGRAM DATA>
  DATA, XDATa
- <numeric> = <DECIMAL NUMERIC PROGRAM DATA>
  0 to 40 0 to 40 dB/1dB Step
- <interface> = <DECIMAL NUMERIC PROGRAM DATA>
  1 to 2 Data Interface for Data1, and Data2
  If [<Interface>] is omitted, the default is 1.

Function
Sets data output ATT Factor for specific port.

Example
To set data output ATT Factor for Data of Data1Interface to 20 dB:
> :MUX:DATA:ATTFactor DATA,20,1

Compatibility
Not backwards compatible

:MUX:DATA:ATTFactor? <port>[,<interface>]

Parameter
- <port> = <CHARACTER PROGRAM DATA>
  DATA, XDATa
- <interface> = <DECIMAL NUMERIC PROGRAM DATA>
  1 to 2 Data Interface for Data1, and Data2
  If [<Interface>] is omitted, the default is 1.

Response
- <numeric> = <NR2 NUMERIC RESPONSE DATA>

Function
Queries data output ATT Factor for specific port.

Example
To query data output ATT Factor for Data of Data1Interface:
> :MUX:DATA:ATTFactor? DATA,1
< 20

Compatibility
Not backwards compatible

:MUX:DATA:CPOint <port>,<numeric>[,<interface>]

Parameter
- <port> = <CHARACTER PROGRAM DATA>
  DATA, XDATa
- <numeric> = <DECIMAL NUMERIC PROGRAM DATA>
  20 to 80 20 to 80%/0.1% Step
- <interface> = <DECIMAL NUMERIC PROGRAM DATA>
  1 to 2 Data Interface for Data1, and Data2
  If [<Interface>] is omitted, the default is 1.

Function
Sets data output crosspoint for specific port.

Example
To set data output crosspoint for XData of Data1Interface to 60.0%:
> :MUX:DATA:CPOint XDATa,60.0,1

Compatibility
Not backwards compatible
7.8 MUX Commands

:MUX:DATA:CPOint? <port>[,<interface>]

Parameter
- <port> = <CHARACTER PROGRAM DATA>
  - DATA, XDAta
- <interface> = <DECIMAL NUMERIC PROGRAM DATA>
  - 1 to 2 Data Interface for Data1, and Data2
    - If [<Interface>] is omitted, the default is 1.

Response
- <numeric> = <NR2 NUMERIC RESPONSE DATA>

Function
Queries data output crosspoint for specific port.

Example
To query data output crosspoint for XDAta of Data1Interface:
> :MUX:DATA:CPOint? XDAta,1
< 60.0

Compatibility
Not backwards compatible

:MUX:DATA:HPJitter <numeric>[,<interface>]

Parameter
- <numeric> = <DECIMAL NUMERIC PROGRAM DATA>
  - –20 to 20 –20 to 20/1 Step
- <interface> = <DECIMAL NUMERIC PROGRAM DATA>
  - 1 to 2 Data Interface for Data1, and Data2
    - If [<Interface>] is omitted, the default is 1.

Function
Sets Half Period Jitter for specific port.

Example
To set Half Period Jitter for Data1Interface to 10:
> :MUX:DATA:HPJitter 10,1

Compatibility
Not backwards compatible


Parameter
- <interface> = <DECIMAL NUMERIC PROGRAM DATA>
  - 1 to 2 Data Interface for Data1, and Data2
    - If [<Interface>] is omitted, the default is 1.

Response
- <numeric> = <NR2 NUMERIC RESPONSE DATA>

Function
Queries data output crosspoint for specific port.

Example
To query Half Period Jitter of Data1Interface:
> :MUX:DATA:HPJitter? 1
< 10

Compatibility
Not backwards compatible
Chapter 7  SCPI Commands

MUX:DATA:UIPadjust <numeric>[,<interface>]

Parameter
<numeric> = <DECIMAL NUMERIC PROGRAM DATA>
-64000 to 64000
-64000 to 64000 mUI/2 mUI Step

<interface> = <DECIMAL NUMERIC PROGRAM DATA>
1 to 2
Data Interface for Data1, and Data2
If [<interface>] is omitted, the default is 1.

Function
Sets shared settings for Data and Xdata output in mUI units.

Example
To set Data output shared settings for Data1Interface to 500 mUI:
> :MUX:DATA:UIPadjust 500,1

Compatibility
Not backwards compatible

MUX:DATA:UIPadjust? [<interface>]

Parameter
<interface> = <DECIMAL NUMERIC PROGRAM DATA>
1 to 2
Data Interface for Data1, and Data2
If [<interface>] is omitted, the default is 1.

Response
<numeric> = < NR1 NUMERIC RESPONSE DATA>

Function
Queries shared settings for Data and Xdata output in mUI units.

Example
To query shared settings for Data output of Data1Interface:
> :MUX:DATA:UIPadjust? 1
< 500

Compatibility
Not backwards compatible

MUX:DATA:PADjust <numeric>[,<interface>]

Parameter
<numeric> = <DECIMAL NUMERIC PROGRAM DATA>
-2560 to 2560
-2560 to 2560 ps/0.08 ps Steps (25 GHz)
Converted from the calculated frequency value, based on the setting resolution of mUI.

<interface> = <DECIMAL NUMERIC PROGRAM DATA>
1 to 2
Data Interface for Data1, and Data2
If [<interface>] is omitted, the default is 1.

Function
Sets shared settings for Data and Xdata output in ps units.
Sometimes, the value cannot be set, depending on the setting resolution.
In this case, set the nearest approximation.

Example
To set Data output shared settings for Data1Interface to 1000 ps:
> :MUX:DATA:PADjust 1000,1

Compatibility
Not backwards compatible

7-516
7.8 MUX Commands

:MUX:DATA:PADJ ust? [<interface>]

Parameter

\(<\text{interface}> = \langle \text{DECIMAL NUMERIC PROGRAM DATA} \rangle\)

1 to 2  Data Interface for Data1, and Data2

If \([\text{<Interface>}]) is omitted, the default is 1.

Response

\(<\text{numeric}> = \langle \text{NR2 NUMERIC RESPONSE DATA} \rangle\)

Function

Queries shared settings for Data and Xdata output in ps units.

Example

To query shared settings for Data output of DataInterface:

> :MUX:DATA:UPadjust? 1
< 1000

Compatibility

Not backwards compatible

:MUX:DATA:PCALibration [<interface>]

Parameter

\(<\text{interface}> = \langle \text{DECIMAL NUMERIC PROGRAM DATA} \rangle\)

1 to 2  Data Interface for Data1, and Data2

If \([\text{<Interface>}]) is omitted, the default is 1.

Function

Calibrates Data and XData output phase.

Example

To calibrate Data and XData output phase for DataInterface:

> :MUX:DATA:PCALibration 1

Compatibility

Not backwards compatible

:MUX:DATA:RELative <boolean>[,<interface>]

Parameter

\(<\text{boolean}> = \langle \text{BOOLEAN PROGRAM DATA} \rangle\)

OFF or 0  Reference OFF (AC output)

ON or 1  Reference ON (DC output)

\(<\text{interface}> = \langle \text{DECIMAL NUMERIC PROGRAM DATA} \rangle\)

1 to 2  Data Interface for Data1, and Data2

If \([\text{<Interface>}]) is omitted, the default is 1.

Function

Sets Data output delay reference ON or OFF.

Example

To set Data output delay reference for DataInterface to ON:

> :MUX:DATA:RELative ON,1

Compatibility

Not backwards compatible
### :MUX:DATA:RELative? [<interface>]

**Parameter**

- `<interface>` = <DECIMAL NUMERIC PROGRAM DATA>
- 1 to 2 Data Interface for Data1, and Data2
  
  If `<Interface>` is omitted, the default is 1.

**Response**

- `<numeric>` = <NR1 NUMERIC RESPONSE DATA>
- 0 Reference OFF
- 1 Reference ON

**Function**

Queries whether Data output delay reference ON or OFF.

**Example**

To query Data output delay reference for Data1Interface:

```
> :MUX:DATA:RELative? 1
< 1
```

**Compatibility**

Not backwards compatible

### :MUX:DATA:RDELay <numeric>,[<unit>][,<interface>]

**Parameter**

- `<numeric>` = <DECIMAL NUMERIC PROGRAM DATA>
  
  (mUI units) –128000 to 128000 –128000 to 128000 mUI/2 mUI Step
  
  (ps units) Converted from the calculated frequency value, based on the setting resolution of mUI.
- `<unit>` = <CHARACTER PROGRAM DATA>
  
  UI mUI units
  
  PS ps units (If `<unit>` is not specified, ps is set by default.)
- `<interface>` = <DECIMAL NUMERIC PROGRAM DATA>
  
  1 to 2 Data Interface for Data1, and Data2
  
  If `<Interface>` is omitted, the default is 1.

**Function**

Sets value and units for data output delay reference.

The setting is performed as an increment from the reference value.

Sometimes at PS setting, some values cannot be set depending on the setting resolution. In this case, set the closest approximation.

**Example**

To set Data output delay reference for Data1Interface to –1000 mUI:

```
> :MUX:DATA:RDELay -1000,UI,1
```

**Compatibility**

Not backwards compatible
### 7.8 MUX Commands

#### :MUX:DATA:RDELay? [<unit>],[<interface>]

<table>
<thead>
<tr>
<th>Parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;unit&gt; = &lt;CHARACTER PROGRAM DATA&gt;</td>
</tr>
<tr>
<td>UI</td>
</tr>
<tr>
<td>PS</td>
</tr>
<tr>
<td>(If &lt;unit&gt; is not specified, ps is set by default.)</td>
</tr>
</tbody>
</table>

| <interface> = <DECIMAL NUMERIC PROGRAM DATA> |
| 1 to 2  | Data Interface for Data1, and Data2 |
| If [<Interface>] is omitted, the default is 1. |

<table>
<thead>
<tr>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;numeric&gt; = &lt;NR2 NUMERIC RESPONSE DATA&gt;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Queries data output delay reference value.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>To query data output delay reference value for Data1Interface in mUI units:</td>
</tr>
<tr>
<td>&gt; :MUX:DATA:RDELay? UI,1</td>
</tr>
<tr>
<td>&lt; -1000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Compatibility</th>
</tr>
</thead>
<tbody>
<tr>
<td>Not backwards compatible</td>
</tr>
</tbody>
</table>

#### :MUX:DATA:JINPut <boolean>[,<interface>]

<table>
<thead>
<tr>
<th>Parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;boolean&gt; = &lt;BOOLEAN PROGRAM DATA&gt;</td>
</tr>
<tr>
<td>OFF or 0</td>
</tr>
<tr>
<td>ON or 1</td>
</tr>
</tbody>
</table>

| <interface> = <DECIMAL NUMERIC PROGRAM DATA> |
| 1 to 2  | Data Interface for Data1, and Data2 |
| If [<Interface>] is omitted, the default is 1. |

<table>
<thead>
<tr>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sets Jitter Input addition status.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>To set Jitter Input addition status for Data1Interface to ON:</td>
</tr>
<tr>
<td>&gt; :MUX:DATA:JINPut ON,1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Compatibility</th>
</tr>
</thead>
<tbody>
<tr>
<td>Not backwards compatible</td>
</tr>
</tbody>
</table>

#### :MUX:DATA:JINPut? [<interface>]

<table>
<thead>
<tr>
<th>Parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;interface&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
</tr>
<tr>
<td>1 to 2</td>
</tr>
<tr>
<td>If [&lt;Interface&gt;] is omitted, the default is 1.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;numeric&gt; = &lt;NR2 NUMERIC RESPONSE DATA&gt;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Queries Jitter Input addition status.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>To query Jitter Input addition status of Data1Interface:</td>
</tr>
<tr>
<td>&gt; :MUX:DATA:JINPut? 1</td>
</tr>
<tr>
<td>&lt; ON</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Compatibility</th>
</tr>
</thead>
<tbody>
<tr>
<td>Not backwards compatible</td>
</tr>
</tbody>
</table>
Chapter 7  SCPI Commands

:MUX:DATA:LIMitter:AMPLitude <numeric>[,<interface>]

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;numeric&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.250 to 1.750</td>
</tr>
<tr>
<td></td>
<td>0.250 to 1.750 V/0.002 V step</td>
</tr>
<tr>
<td></td>
<td>(MU182020A-x10, MU182021A-x10)</td>
</tr>
<tr>
<td></td>
<td>0.500 to 2.500</td>
</tr>
<tr>
<td></td>
<td>0.500 to 2.500 V/0.002 V step</td>
</tr>
<tr>
<td></td>
<td>(MU182020A-x11, MU182021A-x11)</td>
</tr>
<tr>
<td></td>
<td>0.500 to 2.000</td>
</tr>
<tr>
<td></td>
<td>0.500 to 2.000 V/0.002 V step</td>
</tr>
<tr>
<td></td>
<td>(MU182020A-x12, MU182021A-x12)</td>
</tr>
<tr>
<td></td>
<td>0.500 to 3.500</td>
</tr>
<tr>
<td></td>
<td>0.500 to 3.500 V/0.002 V step</td>
</tr>
<tr>
<td></td>
<td>(MU182020A-x13, MU182021A-x13)</td>
</tr>
<tr>
<td>&lt;interface&gt;</td>
<td>= &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
</tr>
<tr>
<td></td>
<td>1 to 2</td>
</tr>
<tr>
<td></td>
<td>Data Interface for Data1, and Data2</td>
</tr>
<tr>
<td></td>
<td>If [&lt;Interface&gt;] is omitted, the default is 1.</td>
</tr>
</tbody>
</table>

Function
Sets Data/XData output amplitude limit.

Example
To set Data/XData output amplitude limit for Data1Interface to 2.000:
> :MUX:DATA:LIMitter:AMPLitude 2.000,1

Compatibility
Not backwards compatible


<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;interface&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1 to 2</td>
</tr>
<tr>
<td></td>
<td>Data Interface for Data1, and Data2</td>
</tr>
<tr>
<td></td>
<td>If [&lt;Interface&gt;] is omitted, the default is 1.</td>
</tr>
</tbody>
</table>

Response
<numeric> = <NR2 NUMERIC RESPONSE DATA>

Function
Queries Data/XData output amplitude limit.

Example
To query Data/XData output amplitude limit of Data1Interface:
> :MUX:DATA:LIMitter:AMPLitude? 1
< 2.000

Compatibility
Not backwards compatible
7.8 MUX Commands

:MUX:DATA:Lim:OFFSet <max>,<min>[,<interface>]

Parameter

<table>
<thead>
<tr>
<th>&lt;max&gt;,&lt;min&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>−2.000 to 3.300</td>
</tr>
<tr>
<td>MU182020A-x10, x11, x12, x13, MU182021A-x10, x11, x12, x13</td>
</tr>
<tr>
<td>−3.750 to 3.050</td>
</tr>
<tr>
<td>MU182020A-x10, MU182021A-x10</td>
</tr>
<tr>
<td>−4.000 to 2.800</td>
</tr>
<tr>
<td>MU182020A-x11, x12, x13, MU182021A-x11, x12, x13</td>
</tr>
<tr>
<td>&lt;interface&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
</tr>
<tr>
<td>1 to 2</td>
</tr>
<tr>
<td>If [&lt;Interface&gt;] is omitted, the default is 1.</td>
</tr>
</tbody>
</table>

Function
Sets Data/XData output offset limit.

Example
To set Data/XData output offset limit for Data1Interface to max 3.000 Voh, min −2.000 Voh:
> :MUX:DATA:Lim:OFFSet 3.000,−2.000,1

Compatibility
Not backwards compatible


Parameter

<table>
<thead>
<tr>
<th>&lt;interface&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 to 2</td>
</tr>
<tr>
<td>If [&lt;Interface&gt;] is omitted, the default is 1.</td>
</tr>
</tbody>
</table>

Response

<table>
<thead>
<tr>
<th>&lt;max&gt; = &lt;NR2 NUMERIC RESPONSE DATA&gt;</th>
</tr>
</thead>
</table>

| <min> = <NR2 NUMERIC RESPONSE DATA> |

Function
Queries Data/XData output offset limit.

Example
To query Data/XData output offset limit of Data1Interface:
> :MUX:DATA:Lim:OFFSet? 1
< 3.000, −2.000

Compatibility
Not backwards compatible
### :MUX:DATA:UDELay:OFFSet <numeric>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;numeric&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function</td>
<td>Sets the unit of the Delay offset value.</td>
</tr>
<tr>
<td>Example</td>
<td>To set the Delay offset to 500 mUI:</td>
</tr>
<tr>
<td></td>
<td>&gt; :MUX:DATA:UDELay:OFFSet 500</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Not backwards compatible</td>
</tr>
</tbody>
</table>

---

### :MUX:DATA:UDELay:OFFSet?

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;numeric&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function</td>
<td>Queries the Delay offset value.</td>
</tr>
<tr>
<td>Example</td>
<td>&gt; :MUX:DATA:UDELay:OFFSet?</td>
</tr>
<tr>
<td></td>
<td>&lt; 500</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Not backwards compatible</td>
</tr>
</tbody>
</table>
7.8 MUX Commands

7.8.2 Commands related to Clock Interface tab

7.8.2.1 Clock Output setting field

![Diagram of Clock Interface setting field]

Figure 7.8.2.1-1 Clock Interface setting field

![Diagram of Level Guard setting field]

Figure 7.8.2.1-2 Level Guard setting field
### Table 7.8.2.1-1  Clock Interface setting commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>:MUX:CLOCK:OUTPut?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:MUX:CLOCK:TRACking?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:MUX:CLOCK:LEVGuard?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:MUX:CLOCK:LEVEL?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:MUX:CLOCK:AMPLitude?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:MUX:CLOCK:AOFFset?</td>
</tr>
<tr>
<td>[7]</td>
<td>Offset (Specific Port)</td>
<td>:MUX:CLOCK:OFFSet</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:MUX:CLOCK:OFFSet?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:MUX:CLOCK:ATTFactor?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:MUX:CLOCK:DUTY?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:MUX:CLOCK:LIMitter:AMPLitude?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:MUX:CLOCK:LIMitter:OFFSet?</td>
</tr>
</tbody>
</table>

**:MUX:CLOCK:OUTPut <boolean>**

**Parameter**  
<boolean> = <BOOLEAN PROGRAM DATA>  
OFF or 0, ON or 1

**Function**  
Sets Clock/XClock output (XClock only MU182021A) ON/OFF.

**Example**  
To set Clock/XClock output to ON:

> :MUX:CLOCK:OUTPut ON

**Compatibility**  
Not backwards compatible
7.8 MUX Commands

:MUX:CLOCKS:OUTPut?

Response  
<numeric> = <NR1 NUMERIC RESPONSE DATA>
0  Clock/XClock output is OFF
1  Clock/XClock output is ON

Function  
Queries whether Clock/XClock (XClock only MU182021A) output ON or OFF.

Example  
To query whether Clock/XClock output ON or OFF:
> :MUX:CLOCKS:OUTPut?
< 1

Compatibility  
Not backwards compatible

:MUX:CLOCKS:TRACKing <boolean>

Parameter  
<boolean> = <BOOLEAN PROGRAM DATA>
OFF or 0, ON or 1

Function  
Sets shared settings for Clock/XClock ON/OFF.

Example  
To set shared settings for Clock/XClock of Clock Interface to ON:
> :MUX:CLOCKS:TRACKing ON

Compatibility  
Not backwards compatible

:MUX:CLOCKS:TRACKing?

Response  
<numeric> = <NR1 NUMERIC RESPONSE DATA>
0  Shared settings for Clock/XClock output OFF
1  Shared settings for Clock/XClock output ON

Function  
Queries shared settings for Clock/XClock output.

Example  
To query shared settings for Clock/XClock output of Clock Interface:
> :MUX:CLOCKS:TRACKing?
< 1

Compatibility  
Not backwards compatible

:MUX:CLOCKS:LEVGuard <boolean>

Parameter  
<boolean> = <BOOLEAN PROGRAM DATA>
OFF or 0, ON or 1

Function  
Sets Clock/XClock output range limit ON/OFF.

Example  
To set Clock/XClock output range limit for Clock Interface to ON:
> :MUX:CLOCKS:LEVGuard ON

Compatibility  
Not backwards compatible
Chapter 7  SCPI Commands

:MUX:CLOCk:LEVGuard?

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;numeric&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Clock/XClock output range limit OFF</td>
</tr>
<tr>
<td>1</td>
<td>Clock/XClock output range limit ON</td>
</tr>
</tbody>
</table>

Function
Queries Clock/XClock output range limit.

Example
To query Clock/XClock output range limit of Clock Interface:

\[
> :MUX:CLOCk:LEVGuard?
\]

< 1

Compatibility
Not backwards compatible

:MUX:CLOCk:LEVel <port>,<level>

Parameter
<br> <port> = <CHARACTER PROGRAM DATA>
  CLOCk, XCLock
<br> <level> = <CHARACTER PROGRAM DATA>
  VARible Variable
  PCML PCML level
  NCML NCML level
  SCFL SCFL level
  NECL NECL level
  LVP ecl LVPECL level

Function
Sets clock output level for specific port.

Example
To set clock output level for XClock of Clock Interface to NECL level:

\[
> :MUX:CLOCk:LEVel XCLock,NECL
\]

Compatibility
Not backwards compatible

:MUX:CLOCk:LEVel? <port>

Parameter
<br> <port> = <CHARACTER PROGRAM DATA>
  CLOCk, XCLock

Response
<br> <level> = <CHARACTER RESPONSE DATA>
  VAR, PCML, NCML, SCFL, NECL, LVP

Function
Queries clock output level for specific port.

Example
To query clock output level for XClock of Clock Interface:

\[
> :MUX:CLOCk:LEVel? XCLock
\]

< NECL

Compatibility
Not backwards compatible
### :MUX:CLOCk:AMPLitude <port>,<numeric>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;port&gt;</td>
<td>&lt;CHARACTER PROGRAM DATA&gt;</td>
</tr>
<tr>
<td>CLOCk, XCLOCk</td>
<td></td>
</tr>
<tr>
<td>&lt;numeric&gt;</td>
<td>&lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
</tr>
<tr>
<td>0.500 to 2.000</td>
<td>0.500 to 2.000 V/0.002 V step</td>
</tr>
</tbody>
</table>

**Function**
Sets clock output amplitude for specific port.

**Example**
To set clock output amplitude for Clock of Clock Interface to 1.000 V:

```
> :MUX:CLOCk:AMPLitude CLOCk,1.000
```

**Compatibility**
Not backwards compatible

### :MUX:CLOCk:AMPLitude? <port>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;port&gt;</td>
<td>&lt;CHARACTER PROGRAM DATA&gt;</td>
</tr>
<tr>
<td>CLOCk, XCLOCk</td>
<td></td>
</tr>
</tbody>
</table>

**Response**

| <numeric>          | <NR2 NUMERIC RESPONSE DATA>                     |

**Function**
Queries clock output amplitude for specific port.

**Example**
To query clock output amplitude for Clock of Clock Interface:

```
> :MUX:CLOCk:AMPLitude? CLOCk
< 1.000
```

**Compatibility**
Not backwards compatible

### :MUX:CLOCk:AOFFset <boolean>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;boolean&gt;</td>
<td>&lt;BOOLEAN PROGRAM DATA&gt;</td>
</tr>
<tr>
<td>OFF or 0</td>
<td>Offset OFF (DC output)</td>
</tr>
<tr>
<td>ON or 1</td>
<td>Offset ON (AC output)</td>
</tr>
</tbody>
</table>

**Function**
Sets clock output offset ON/OFF.

**Example**
To set output offset for Clock of Clock Interface to ON:

```
> :MUX:CLOCk:AOFFset ON
```

**Compatibility**
Not backwards compatible

### :MUX:CLOCk:AOFFset?

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;NR1 NUMERIC RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Offset OFF</td>
</tr>
<tr>
<td>1</td>
<td>Offset ON</td>
</tr>
</tbody>
</table>

**Function**
Queries clock output offset ON/OFF.

**Example**
To query clock output offset of Clock Interface:

```
> :MUX:CLOCk:AOFFset?
< 1
```

**Compatibility**
Not backwards compatible
Chapter 7  SCPI Commands

:MUX:CLOCK:OFFSET <port>,<numeric>

Parameter  
<port> = <CHARACTER PROGRAM DATA>  
CLOCK, XCLock  
<numeric> = <DECIMAL NUMERIC PROGRAM DATA>  
-2.000 to 3.300  
-3.000 to 3.050  
-4.000 to 2.800  
Function  
Sets clock output offset for specific port.  
Example  
To set output offset for Clock of Clock Interface to –1.000 Voh:  
> :MUX:CLOCK:OFFSET CLOCK,-1.000  
Compatibility  
Not backwards compatible

:MUX:CLOCK:OFFSET? <port>

Parameter  
<port> = <CHARACTER PROGRAM DATA>  
CLOCK, XCLock  
Response  
<numeric> = <NR2 NUMERIC RESPONSE DATA>  
Function  
Queries clock output offset for specific port.  
Example  
To query clock output offset for Clock of Clock Interface:  
> :MUX:CLOCK:OFFSET? CLOCK  
< -1.000  
Compatibility  
Not backwards compatible

:MUX:CLOCK:ATTFactor <port>,<numeric>

Parameter  
<port> = <CHARACTER PROGRAM DATA>  
CLOCK, XCLock  
<numeric> = <DECIMAL NUMERIC PROGRAM DATA>  
0 to 40  
Function  
Sets clock output ATT factor for specific port.  
Example  
To set clock output ATT Factor for Clock of Clock Interface to 20 dB:  
> :MUX:CLOCK:ATTFactor CLOCK,20  
Compatibility  
Not backwards compatible
7.8 MUX Commands

:MUX:CLOCk:ATTFactor? <port>
Parameter <port> = <CHARACTER PROGRAM DATA>
CLOCk, XCLOCk
Response <numeric> = <NR2 NUMERIC RESPONSE DATA>
Function Queries clock output ATT factor for specific port.
Example To query clock output ATT Factor for Clock of Clock Interface:
> :MUX:CLOCk:ATTFactor? CLOCk
< 20
Compatibility Not backwards compatible

:MUX:CLOCk:DUTY <numeric>
Parameter <numeric> = <DECIMAL NUMERIC PROGRAM DATA>
–25 to 25 –25 to 25/1 step
Function Sets Duty.
Example To set Duty for Clock Interface to 25:
> :MUX:CLOCk:DUTY 25
Compatibility Not backwards compatible

:MUX:CLOCk:DUTY?
Response <numeric> = <NR2 NUMERIC RESPONSE DATA>
Function Queries Duty.
Example To query Duty of Clock Interface:
> :MUX:CLOCk:DUTY?
< 25
Compatibility Not backwards compatible

:MUX:CLOCk:LIMitter:AMPLitude <numeric>
Parameter <numeric> = <DECIMAL NUMERIC PROGRAM DATA>
0.500 to 2.000 0.500 to 2.000 V/0.002 V step
Function Sets limit for Clock/XClock output amplitude.
Example To set limit for Clock/XClock output amplitude of Clock Interface to 2.000:
> :MUX:CLOCk:LIMitter:AMPLitude 2.000
Compatibility Not backwards compatible
Chapter 7  SCPI Commands

:MUX:CLOCK:LIMITER:AMPLitude?
Response  <numeric> = <NR2 NUMERIC RESPONSE DATA>
Function  Queries limit for Clock/XClock output amplitude.
Example  To query limit for Clock/XClock output amplitude of Clock Interface:
> :MUX:CLOCK:LIMITER:AMPLitude?
< 2.000
Compatibility  Not backwards compatible

:MUX:CLOCK:LIMITER:OFFSET <max>,<min>
Parameter  <max>,<min> = <DECIMAL NUMERIC PROGRAM DATA>
–2.000 to 3.300  –2.000 to 3.300 Voh/0.001 V step
–4.000 to 2.800  –4.000 to 2.800 Vol/0.001 V step
Function  Sets limit for Clock/XClock output offset.
Example  To set limit for Clock/XClock output offset of Clock Interface to max
3.000 Voh and min –2.000 Voh:
> :MUX:CLOCK:LIMITER:OFFSET 3.000,-2.000
Compatibility  Not backwards compatible

:MUX:CLOCK:LIMITER:OFFSET?
Response  <max> = <NR2 NUMERIC RESPONSE DATA>
          <min> = <NR2 NUMERIC RESPONSE DATA>
Function  Queries limit for Clock/XClock output offset.
Example  To query limit for Clock/XClock output offset of Clock Interface:
> :MUX:CLOCK:LIMITER:OFFSET?
< 3.000,-2.000
Compatibility  Not backwards compatible
7.8.2.2 Clock Input setting field

Table 7.8.2.2-1 Clock Input setting commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1]</td>
<td>Clock Input Band Switch</td>
<td>:MUX:CLOk:BAND</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:MUX:CLOk:BAND?</td>
</tr>
</tbody>
</table>
### :MUX:CLOCK:_BAND <sw>

**Parameter**  
<sw> = <CHARACTER PROGRAM DATA>  
HALFrate, FULLrate

**Function**  
Sets input rate switching for external clock source.

**Example**  
To set input rate switching for external clock source to Half Rate:

> :MUX:CLOCK:_BAND HALF

**Compatibility**  
Not backwards compatible

---

### :MUX:CLOCK:_BAND?  

**Response**  
<sw> = <CHARACTER RESPONSE DATA>  
HALF, FULL

**Function**  
Queries input rate switching for external clock source.

**Example**  
To query the input rate switching:

> :MUX:CLOCK:_BAND?
  
< HALF

**Compatibility**  
Not backwards compatible
7.8 MUX Commands

7.8.3 Commands related to Emphasis tab

7.8.3.1 Emphasis setting field

![Emphasis setting diagram]

**Figure 7.8.3.1-1 Emphasis setting**

**Table 7.8.3.1-1 Emphasis setting commands**

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>:MUX:EMPHasis:SET?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:MUX:EMPHasis:TRACking?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:MUX:EMPHasis:LEVel?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:MUX:EMPHasis:AMPLitude?</td>
</tr>
</tbody>
</table>
### Table 7.8.3.1-1  Emphasis setting commands (Cont’d)

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>:MUX:EMPHasis:RATio:SIGN?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:MUX:EMPHasis:AMPLitude:RATio?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:MUX:EMPHasis:ATTFactor?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:MUX:EMPHasis:AOFFset?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:MUX:EMPHasis:OFFSet?</td>
</tr>
<tr>
<td>[10]</td>
<td>Pulse Width (mUI setting)</td>
<td>:MUX:EMPHasis:PULSewidth:UIPadjust</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:MUX:EMPHasis:PULSewidth:UIPadjust?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:MUX:EMPHasis:PULSewidth:PADJust?</td>
</tr>
<tr>
<td>[12]</td>
<td>Delay (mUI setting)</td>
<td>:MUX:EMPHasis:UIPadjust</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:MUX:EMPHasis:UIPadjust?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:MUX:EMPHasis:PADJust?</td>
</tr>
</tbody>
</table>

---

**MUX:EMPHasis:SET <boolean>**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;boolean&gt; = &lt;BOOLEAN PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>OFF or 0</td>
<td>Emphasis signal OFF</td>
</tr>
<tr>
<td>ON or 1</td>
<td>Emphasis signal ON</td>
</tr>
</tbody>
</table>

**Function**
Sets Emphasis signal ON/OFF.

**Example**
To set Emphasis signal to ON:
> :MUX:EMPHasis:SET ON

**Compatibility**
Not backwards compatible

**MUX:EMPHasis:SET?**

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;numeric&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Emphasis signal OFF</td>
</tr>
<tr>
<td>1</td>
<td>Emphasis signal ON</td>
</tr>
</tbody>
</table>

**Function**
Queries Emphasis signal ON/OFF.

**Example**
To query Emphasis signal ON/OFF:
> :MUX:EMPHasis:SET?
> < 1

**Compatibility**
Not backwards compatible
### :MUX:EMPHasis:TRACking <boolean>

**Parameter**

<boolean> = <BOOLEAN PROGRAM DATA>
- OFF or 0: Emphasis Tracking OFF
- ON or 1: Emphasis Tracking ON

**Function**
Sets Emphasis Tracking ON/OFF.

**Example**
To set Emphasis Tracking to ON:

```
> :MUX:EMPHasis:TRACking ON
```

**Compatibility**
Not backwards compatible

### :MUX:EMPHasis:TRACking? 

**Response**

<numeric> = <NR1 NUMERIC RESPONSE DATA>
- 0: Emphasis Tracking OFF
- 1: Emphasis Tracking ON

**Function**
Queries Emphasis Tracking ON/OFF.

**Example**
To query Emphasis Tracking ON/OFF:

```
> :MUX:EMPHasis:TRACking?
< 1
```

**Compatibility**
Not backwards compatible

### :MUX:EMPHasis:LEVel <port>,<level>

**Parameter**

- <port> = <CHARACTER PROGRAM DATA>
  - DATA, XDATa
- <level> = <CHARACTER PROGRAM DATA>
  - Variable

**Function**
Sets output level for specific port.

**Example**
To set Emphasis XData output level to Variable:

```
> :MUX:EMPHasis:LEVel XDATa,VARiable
```

**Compatibility**
Not backwards compatible

### :MUX:EMPHasis:LEVel? <port>

**Parameter**

- <port> = <CHARACTER PROGRAM DATA>
  - DATA, XDATa

**Response**

- <level> = <CHARACTER RESPONSE DATA>
  - VAR

**Function**
Queries output level for specific port.

**Example**
To query Emphasis XData output level:

```
> :MUX:EMPHasis:LEVel? XDATa
< VAR
```

**Compatibility**
Not backwards compatible
Chapter 7  SCPI Commands

:MUX:EMPHasis:AMPLitude <port>,<numeric>
Parameter
<port> = <CHARACTER PROGRAM DATA>
DATA, XDATa
<nNumeric> = <DECIMAL NUMERIC PROGRAM DATA>
0.002 to 2.582 0.002 to 2.582 V/0.002 V step
Function
Sets output amplitude for specific port.
Example
To set Emphasis Data output amplitude to 1.000 V:
> :MUX:EMPHasis:AMPLitude DATA,1.000
Compatibility
Not backwards compatible

:MUX:EMPHasis:AMPLitude? <port>
Parameter
<port> = <CHARACTER PROGRAM DATA>
DATA, XDATa
Response
<nNumeric> = <NR2 NUMERIC RESPONSE DATA>
Function
Queries output amplitude for specific port.
Example
To query Emphasis Data output amplitude:
> :MUX:EMPHasis:AMPLitude? DATA
< 1.000
Compatibility
Not backwards compatible

:MUX:EMPHasis:RATio:SIGN <sign>
Parameter
<sign> = <CHARACTER PROGRAM DATA>
PLUS Set Amplitude Ratio to PLUS (+)
MINus Set Amplitude Ratio to MINUS (–)
Function
Switches Amplitude Ratio sign.
Example
To set Emphasis Amplitude Ratio to PLUS (+):
> :MUX:EMPHasis:RATio:SIGN PLUS
Compatibility
Not backwards compatible

:MUX:EMPHasis:RATio:SIGN?
Response
<sign> = <CHARACTER RESPONSE DATA>
PLUS, MIN
Function
Queries Amplitude Ratio sign.
Example
> :MUX:EMPHasis:RATio:SIGN?
< PLUS
Compatibility
Not backwards compatible
### 7.8 MUX Commands

#### :MUX:EMPHasis:AMPLitude:RATio <port>,<numeric>

**Parameter**

- `<port>` = `<CHARACTER PROGRAM DATA>
  DATA, XDATa`
- `<numeric>` = `<DECIMAL NUMERIC PROGRAM DATA>
  –9.0 to 9.0 –9.0 to 9.0 dB/0.1 dB step`

**Function**

Sets Emphasis value for specific port.

**Example**

To set Data Emphasis to 2.5 dB:

```
> :MUX:EMPHasis:AMPLitude:RATio DATA,2.5
```

**Compatibility**

Not backwards compatible

#### :MUX:EMPHasis:AMPLitude:RATio? <port>

**Parameter**

- `<port>` = `<CHARACTER PROGRAM DATA>
  DATA, XDATa`

**Response**

- `<numeric>` = `<NR2 NUMERIC RESPONSE DATA>`

**Function**

Queries Emphasis value for specific port.

**Example**

To query Emphasis value for Emphasis Data:

```
< 2.5
```

**Compatibility**

Not backwards compatible

#### :MUX:EMPHasis:ATTFactor <port>,<numeric>[,<interface>]

**Parameter**

- `<port>` = `<CHARACTER PROGRAM DATA>
  DATA, XDATa`
- `<numeric>` = `<DECIMAL NUMERIC PROGRAM DATA>
  0 to 40 0 to 40 dB/1 dB step`
- `<interface>` = `<DECIMAL NUMERIC PROGRAM DATA>
  1 to 2 Data Interface for Data1, and Data2`
  If `[<Interface>]` is omitted, the default is 1.

**Function**

Sets data output ATT factor for specific port.

**Example**

To set data output ATT Factor for Data of Data1Interface to 20 dB:

```
> :MUX:EMPHasis:ATTFactor DATA,20,1
```

**Compatibility**

Not backwards compatible
### :MUX:EMPHasis:ATTFactor? <port>[,<interface>]

**Parameter**
- `<port>` = `<CHARACTER PROGRAM DATA>`
  - DATA, XDATa
- `<interface>` = `<DECIMAL NUMERIC PROGRAM DATA>`
  - 1 to 2 Data Interface for Data1, and Data2
  - If `<interface>` is omitted, the default is 1.

**Response**
- `<numeric>` = `<NR2 NUMERIC RESPONSE DATA>`

**Function**
Queries data output ATT factor for specific port.

**Example**
To query data output ATT factor for Data of Data1 Interface:

```
> :MUX:EMPHasis:ATTFactor? DATA, 1
< 20
```

**Compatibility**
Not backwards compatible

### :MUX:EMPHasis:AOFFset <boolean>

**Parameter**
- `<boolean>` = `<BOOLEAN PROGRAM DATA>`
  - OFF or 0 Offset OFF (DC output)
  - ON or 1 Offset ON (AC output)

**Function**
Sets Emphasis Data output offset ON/OFF.

**Example**
To set Emphasis Data output offset to ON:

```
> :MUX:EMPHasis:AOFFset ON
```

**Compatibility**
Not backwards compatible

### :MUX:EMPHasis:AOFFset?

**Response**
- `<numeric>` = `<NR1 NUMERIC RESPONSE DATA>`
  - 0 Offset OFF
  - 1 Offset ON

**Function**
Queries Emphasis Data output offset ON/OFF.

**Example**
To query Emphasis Data output offset:

```
> :MUX:EMPHasis:AOFFset?
< 1
```

**Compatibility**
Not backwards compatible
:MUX:EMPHasis:OFFSet <port>,<numeric>

Parameter

- `<port>` = `<CHARACTER PROGRAM DATA>`
  
  - DATA, XDATA

- `<numeric>` = `<DECIMAL NUMERIC PROGRAM DATA>`
  
  - -3.000 to 3.175
  
  - -3.000 to 3.175 Vth/0.001 V step

Function

Sets output offset for specific port.

Example

To set Emphasis Data output offset to 1.000 Vth:

```
> :MUX:EMPHasis:OFFSet DATA,1.000
```

Compatibility

Not backwards compatible

---

:MUX:EMPHasis:OFFSet? <port>

Parameter

- `<port>` = `<CHARACTER PROGRAM DATA>`
  
  - DATA, XDATA

Response

- `<numeric>` = `<NR2 NUMERIC RESPONSE DATA>`

Function

Queries output offset for specific port.

Example

To query output offset for Emphasis Data:

```
> :MUX:EMPHasis:OFFSet? DATA
< 1.000
```

Compatibility

Not backwards compatible

---

:MUX:EMPHasis:PULSewidth:UIPadjust <numeric>

Parameter

- `<numeric>` = `<DECIMAL NUMERIC PROGRAM DATA>`
  
  - -500 to 1500
  
  - -500 to 1500 mUI/2 mUI Step

Function

Sets Emphasis phase in mUI

Sets the Emphasis Pulse Width value in mUI units.

Example

To set Emphasis phase to 500 mUI:

To set the Emphasis Pulse Width value to 500 mUI:

```
> :MUX:EMPHasis:PULSewidth:UIPadjust 500
```

Compatibility

Not backwards compatible

---

:MUX:EMPHasis:PULSewidth:UIPadjust?

Response

- `<numeric>` = `<NR2 NUMERIC RESPONSE DATA>`

Function

Queries Emphasis phase in mUI units.

Example

```
> :MUX:EMPHasis:PULSewidth:UIPadjust?
< 500
```

Compatibility

Not backwards compatible
Chapter 7  SCPI Commands

:MUX:EMPHasis:PULSewidth:PADJust <numeric>
Parameter  
<numeric> = <DECIMAL NUMERIC PROGRAM DATA>
-20 to 60  
-20 to 60 ps/0.08 ps Steps (at 25 GHz)
Converted from the calculated frequency value, based on the setting resolution of mUI.
Function  
Sets the Emphasis Pulse Width in ps units.
The specified value may not be exactly set as is, due to the limitation of the setting resolution. In this event, the value is adjusted to the most approximate value and set.
Example  
To set Emphasis phase to 30 ps:
To set the Emphasis Pulse Width value to 30 ps:
> :MUX:EMPHasis:PULSewidth:PADJust 30
Compatibility  
Not backwards compatible

:MUX:EMPHasis:PULSewidth:PADJust?
Response  
<numeric> = <NR2 NUMERIC RESPONSE DATA>
Function  
Queries the Emphasis Pulse Width value in ps units.
Example  
> :MUX:EMPHasis:PULSewidth:PADJust?
< 30
Compatibility  
Not backwards compatible

:MUX:EMPHasis:UIPadjust <numeric>
Parameter  
<numeric> = <DECIMAL NUMERIC PROGRAM DATA>
-2000 to 2000  
-2000 to 2000 mUI/2 mUI Step
Function  
Sets Emphasis output phase in mUI units.
Example  
To set Emphasis output phase to 500 mUI:
> :MUX:EMPHasis:UIPadjust 500
Compatibility  
Not backwards compatible

:MUX:EMPHasis:UIPadjust?
Response  
<numeric> = <NR1 NUMERIC RESPONSE DATA>
Function  
Queries Emphasis output phase.
Example  
> :MUX:EMPHasis:UIPadjust?
< 500
Compatibility  
Not backwards compatible
### :MUX:EMPHasis:PADJust <numeric>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;numeric&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function</td>
<td>Sets Emphasis output phase in ps units.</td>
</tr>
<tr>
<td>Example</td>
<td>To set Emphasis output phase to 80 ps:</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Not backwards compatible</td>
</tr>
</tbody>
</table>

-80 to 80
-80 to 80 ps/0.08 ps Step (at 25 GHz)
Converted from the calculated frequency value, based on the setting resolution of mUI.

### :MUX:EMPHasis:PADJust?

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;numeric&gt; = &lt;NR2 NUMERIC RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function</td>
<td>Queries Emphasis output phase in ps units.</td>
</tr>
<tr>
<td>Example</td>
<td>&gt; :MUX:EMPHasis:UIPadjust?</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Not backwards compatible</td>
</tr>
</tbody>
</table>

< 80

### :MUX:EMPHasis:PCALibration

<table>
<thead>
<tr>
<th>Parameter</th>
<th>None</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function</td>
<td>Calibrates Emphasis Delay.</td>
</tr>
<tr>
<td>Example</td>
<td>&gt; :MUX:EMPHasis:PCALibration</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Not backwards compatible</td>
</tr>
</tbody>
</table>
7.9 DEMUX Commands

This section describes the setting and query commands for the MU182040A 25 Gbit/s 1ch DEMUX, and MU182041A 25 Gbit/s 2ch DEMUX. The commands related to each tab are described in separate sections below. Before executing a setting/query command for the MU182040A, specify the slot number of the module to be remotely controlled, by using the :MODule:ID command. Refer to Chapter 7.1 "Common Commands" and 9.1 "Basic Operation of Device Messages" for how to specify a slot number with the :MODule:ID command. However, unless there is a special item, see the MU182041A.

7.9.1 Commands related to Data Interface tab

7.9.1.1 Data Interface setting field

![Figure 7.9.1.1-1 Data Interface setting field](image)

<table>
<thead>
<tr>
<th>Data</th>
<th>Data1 Interface</th>
<th>Data2 Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input Condition</td>
<td>Single-Ended</td>
<td></td>
</tr>
<tr>
<td>Data Threshold</td>
<td>0.000 V</td>
<td></td>
</tr>
<tr>
<td>Termination</td>
<td>GND</td>
<td>0.00 V</td>
</tr>
<tr>
<td>Data-XData</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Figure 7.9.1.1-1 Data Interface setting field**
### 7.9 DEMUX Commands

#### Table 7.9.1.1-1 Data Interface setting commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>:DEMux:DATA:INTerface?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:DEMux:DATA:DIFFerential?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:DEMux:DATA:SINGle?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:DEMux:DATA:THReshold?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:DEMux:DATA:DIFFerential:BASis?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:DEMux:DATA:DIFFerential:THReshold?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:DEMux:DATA:TERMination?</td>
</tr>
<tr>
<td>[8]</td>
<td>No label (Termination voltage setting)</td>
<td>:DEMux:DATA:TLEVel</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:DEMux:DATA:TLEVel?</td>
</tr>
</tbody>
</table>

**:DEMux:DATA:INTerface <input>[,<interface>]**

- **Parameter**
  - `<input>` = <CHARACTER PROGRAM DATA>
    - SINGle: Single Ended
    - DIF50ohm: Differential 50 Ω
    - DIF100ohm: Differential 100 Ω
  - `<interface>` = <DECIMAL NUMERIC PROGRAM DATA>
    - 1 to 2: Data Interface for Data1, and Data2
      - If `<interface>` is omitted, the default is 1.

- **Function**
  - Sets data input interface.

- **Example**
  - To set Data1Interface interface to single connection:
    ```plaintext
    > :DEMux:DATA:INTerface SINGle,1
    ```

- **Compatibility**
  - Not backwards compatible
Chapter 7  SCPI Commands

:DEMux:DATA:INTerface? [<interface>]

Parameter

- `<interface>` = `<DECIMAL NUMERIC PROGRAM DATA>
  1 to 2     Data Interface for Data1, and Data2
  If [<Interface>] is omitted, the default is 1.

Response

- `<interface>` = `<CHARACTER RESPONSE DATA>
  SING, DIF50, DIF100

Function

Queries data input interface.

Example

To query Data1Interface interface:

```
> :DEMux:DATA:INTerface? 1
< SING
```

Compatibility

Not backwards compatible

:DEMux:DATA:DIFFerential <item>[,<interface>]

Parameter

- `<item>` = `<CHARACTER PROGRAM DATA>
  INDependent     Data/XData independent settings
  TRACking       Data/XData shared settings
  ALTernate      Data/XData alternate settings
- `<interface>` = `<DECIMAL NUMERIC PROGRAM DATA>
  1 to 2     Data Interface for Data1, and Data2
  If [<Interface>] is omitted, the default is 1.

Function

Sets differential type for Differential input data.

Example

To set Differential Independent for Data1Interface:

```
> :DEMux:DATA:DIFFerential INDependent,1
```

Compatibility

Not backwards compatible

:DEMux:DATA:DIFFerential? [<interface>]

Parameter

- `<interface>` = `<DECIMAL NUMERIC PROGRAM DATA>
  1 to 2     Data Interface for Data1, and Data2
  If [<Interface>] is omitted, the default is 1.

Response

- `<item>` = `<CHARACTER RESPONSE DATA>
  IND, TRAC, ALT

Function

Queries differential type for Differential input data.

Example

To query differential type for Data1Interface:

```
> :DEMux:DATA:DIFFerential? 1
< IND
```

Compatibility

Not backwards compatible
### :DEMUX:DATA:SINGle <item>[,<interface>]

**Parameter**

- `<item>` = `<CHARACTER PROGRAM DATA>`
  - DATA: Data is selected
  - XDATa: XData is selected
- `<interface>` = `<DECIMAL NUMERIC PROGRAM DATA>`
  - 1 to 2: Data Interface for Data1, and Data2
    - If `<Interface>` is omitted, the default is 1.

**Function**

Sets input port for Single data input.

**Example**

To set Data1Interface input port to data:

```
> :DEMUX:DATA:SINGle DATA,1
```

**Compatibility**

Not backwards compatible

### :DEMUX:DATA:SINGle? [<interface>]

**Parameter**

- `<interface>` = `<DECIMAL NUMERIC PROGRAM DATA>`
  - 1 to 2: Data Interface for Data1, and Data2
    - If `<Interface>` is omitted, the default is 1.

**Response**

- `<item>` = `<CHARACTER RESPONSE DATA>`
  - DATA: Data is selected
  - XDAT: XData is selected

**Function**

Queries input port for Single data input.

**Example**

To query Data1Interface input port:

```
> :DEMUX:DATA:SINGle? 1
< DATA
```

**Compatibility**

Not backwards compatible

### :DEMUX:DATA:THReshold <port>,<numeric>[,<interface>]

**Parameter**

- `<port>` = `<CHARACTER PROGRAM DATA>`
  - DATA, XDATa (At Single, input the selected port.)
- `<numeric>` = `<DECIMAL NUMERIC PROGRAM DATA>`
  - −3.500 to 3.300: −3.500 to 3.300 V/0.001 V Step
- `<interface>` = `<DECIMAL NUMERIC PROGRAM DATA>`
  - 1 to 2: Data Interface for Data1, and Data2
    - If `<Interface>` is omitted, the default is 1.

**Function**

Sets date input threshold value for specific port.

**Example**

To set threshold value for Data1Interface input port to −3 V:

```
> :DEMUX:DATA:THReshold DATA, -3,1
```

**Compatibility**

Not backwards compatible
Chapter 7  SCPI Commands

:DEMux:DATA:THReshold? <port>[,<interface>]

Parameter

\(<\text{port}> = \langle\text{CHARACTER PROGRAM DATA}\rangle\)
- DATA, XDATa (At Single, input the selected port.)
\(<\text{interface}> = \langle\text{DECIMAL NUMERIC PROGRAM DATA}\rangle\)
- 1 to 2 Data Interface for Data1, and Data2
  - If [<Interface>] is omitted, the default is 1.

Response

\(<\text{numeric}> = \langle\text{NR2 NUMERIC RESPONSE DATA}\rangle\)

Function
Queries date input threshold value for specific port.

Example
To query threshold value for Data1Interface input port:
\(> :\text{DEMux:DATA:THReshold? DATA,1}\)
\(< -3.000\)

Compatibility
Not backwards compatible

:DEMux:DATA:DIFFerential:BASis <basis>[,<interface>]

Parameter

\(<\text{basis}> = \langle\text{CHARACTER PROGRAM DATA}\rangle\)
- DATA Data set as reference
- XDATa XData set as reference
\(<\text{interface}> = \langle\text{DECIMAL NUMERIC PROGRAM DATA}\rangle\)
- 1 to 2 Data Interface for Data1, and Data2
  - If [<Interface>] is omitted, the default is 1.

Function
Sets differential signal for data input threshold value.

Example
To set differential reference signal for Data1Interface threshold value to Data:
\(> :\text{DEMux:DATA:DIFFerential:BASis DATA,1}\)

Compatibility
Not backwards compatible

:DEMux:DATA:DIFFerential:BASis? [<interface>]

Parameter

\(<\text{interface}> = \langle\text{DECIMAL NUMERIC PROGRAM DATA}\rangle\)
- 1 to 2 Data Interface for Data1, and Data2
  - If [<Interface>] is omitted, the default is 1.

Response

\(<\text{basis}> = \langle\text{CHARACTER RESPONSE DATA}\rangle\)
- DATA Data set as reference
- XDATa XData set as reference

Function
Queries differential signal for data input threshold value.

Example
To query differential reference signal for Data1Interface threshold value:
\(> :\text{DEMux:DATA:DIFFerential:BASis? 1}\)
\(< \text{DATA}\)

Compatibility
Not backwards compatible
7.9 DEMUX Commands

:DEMUX:DATA:DIFFerential:THReshold <numeric>[,<interface>]

Parameter

- `<numeric>` = <DECIMAL NUMERIC PROGRAM DATA>
- `-1.500 to 1.500` V
- `<interface>` = <DECIMAL NUMERIC PROGRAM DATA>
- `1 to 2` Data Interface for Data1, and Data2
  
  If `[<Interface>]` is omitted, the default is 1.

Function
Sets data input threshold value at differential input.

Example
To set Data1Interface differential data input value to -1 V:

```
> :DEMUX:DATA:DIFFerential:THReshold -1,1
```

Compatibility
Not backwards compatible


Parameter

- `<interface>` = <DECIMAL NUMERIC PROGRAM DATA>
- `1 to 2` Data Interface for Data1, and Data2
  
  If `[<Interface>]` is omitted, the default is 1.

Response

- `<numeric>` = <NR2 NUMERIC RESPONSE DATA>

Function
Queries data input threshold value at differential input.

Example
To query Data1Interface differential data input value:

```
> :DEMUX:DATA:DIFFerential:THReshold? 1
< -2.000
```

Compatibility
Not backwards compatible

:DEMUX:DATA:TERmination <term>[,<interface>]

Parameter

- `<term>` = <CHARACTER PROGRAM DATA>
- `GND` 50 Ω to GND
- `VARiable` 50 Ω to Variable Voltage

- `<interface>` = <DECIMAL NUMERIC PROGRAM DATA>
- `1 to 2` Data Interface for Data1, and Data2
  
  If `[<Interface>]` is omitted, the default is 1.

Function
Sets date input termination conditions.

Example
To set Data1Interface termination conditions to GND:

```
> :DEMUX:DATA:TERmination GND,1
```

Compatibility
Not backwards compatible
Chapter 7  SCPI Commands

:DEMux:DATA:TERMination? [<interface>]

Parameter  
<interface> = <DECIMAL NUMERIC PROGRAM DATA>
1 to 2  
Data Interface for Data1, and Data2
If [<Interface>] is omitted, the default is 1.

Response  
<term> = <CHARACTER RESPONSE DATA>
GND, VAR

Function  
Queries date input termination conditions.

Example  
To query Data1Interface termination conditions:
> :DEMux:DATA:TERMination? 1
< GND

Compatibility  
Not backwards compatible

:DEMux:DATA:TLEVel <numeric>[,<interface>]

Parameter  
<numeric> = <DICIMAL NUMERIC PROGRAM DATA>
–2.50 to 3.50  
–2.50 to 3.50 V/0.01 V Step
<interface> = <DECIMAL NUMERIC PROGRAM DATA>
1 to 2  
Data Interface for Data1, and Data2
If [<Interface>] is omitted, the default is 1.

Function  
Sets termination voltage at Variable input termination condition.

Example  
To set Data1Interface variable termination voltage to –2.5:
> :DEMux:DATA:TLEVel -2.5,1

Compatibility  
Not backwards compatible

:DEMux:DATA:TLEVel? [<interface>]

Parameter  
<interface> = <DECIMAL NUMERIC PROGRAM DATA>
1 to 2  
Data Interface for Data1, and Data2
If [<Interface>] is omitted, the default is 1.

Response  
<numeric> = <NR2 NUMERIC RESPONSE DATA>

Function  
Queries termination voltage at Variable input termination condition.

Example  
To query Data1Interface variable termination voltage:
> :DEMux:DATA:TLEVel? 1
< -2.50

Compatibility  
Not backwards compatible
7.9 DEMUX Commands

7.9.2 Commands related to Clock Interface tab

7.9.2.1 Clock Interface setting field

Table 7.9.2.1-1 Clock Interface setting commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1]</td>
<td>Selection</td>
<td>:DEMux:CLOCk:SELection</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:DEMux:CLOCk:SELection?</td>
</tr>
<tr>
<td>[2]</td>
<td>Delay (mUI units)</td>
<td>:DEMux:CLOCk:UIDelay</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:DEMux:CLOCk:UIDelay?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:DEMux:CLOCk:PDELay?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:DEMux:CLOCk:RELative?</td>
</tr>
</tbody>
</table>
Table 7.9.2.1-1  Clock Interface setting commands (Cont’d)

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>[6]</td>
<td>No label (Settings of relative values)</td>
<td>:DEMux:CLOCk:RDELay</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:DEMux:CLOCk:RDELay?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:DEMux:CLOCk:JINPut?</td>
</tr>
<tr>
<td>[8]</td>
<td>Clock Input Band Switch</td>
<td>:DEMux:CLOCk:BAND</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:DEMux:CLOCk:BAND?</td>
</tr>
</tbody>
</table>

:DEMux:CLOCk:SELection <sel>

Parameter <sel> = <CHARACTER PROGRAM DATA>

Function Sets clock input type.

Example To set Data1Interface clock input type to external clock:

> :DEMux:CLOCk:SELection EXTernal

Compatibility Not backwards compatible

:DEMux:CLOCk:SELection?

Response <sel> = <CHARACTER RESPONSE DATA>

Example To query Data1Interface clock input type:

> :DEMux:CLOCk:SELection?

< EXT

Compatibility Not backwards compatible

:DEMux:CLOCk:UIDelay <numeric>[,<interface>]

Parameter

- <numeric> = <DECIMAL NUMERIC PROGRAM DATA>
  - –2000 to 2000
  - –2000 to 2000mUI/2mUI Step
- <interface> = <DECIMAL NUMERIC PROGRAM DATA>
  - 1 to 2 Data Interface for Data1, and Data2
  - If [<Interface>] is omitted, the default is 1.

Function Sets clock input phase variation in mUI units.

Example To set Data1Interface clock phase variation to –1000 mUI:

> :DEMux:CLOCk:UIDelay -1000,1

Compatibility Not backwards compatible
### :DEMUX:CLOCK:UIDelay? [<interface>]

**Parameter**

<interface> = <DECIMAL NUMERIC PROGRAM DATA>

<table>
<thead>
<tr>
<th>1 to 2</th>
<th>Data Interface for Data1, and Data2</th>
</tr>
</thead>
</table>

If [<Interface>] is omitted, the default is 1.

**Response**

<numeric> = <NR1 NUMERIC RESPONSE DATA>

**Function**

Queries clock input phase variation in mUI units.

**Example**

To query Data1Interface clock phase variation in mUI units:

```
> :DEMUX:CLOCK:UIDelay? 1
< -1000
```

**Compatibility**

Not backwards compatible

### :DEMUX:CLOCK:PDELAY<numeric>[,<interface>]

**Parameter**

<numeric> = <DECIMAL NUMERIC PROGRAM DATA>

| –80 to 80 | –80 to 80 ps/0.8ps Steps (at 25 GHz) |

Converted from the calculated frequency value, based on the setting resolution of mUI.

<interface> = <DECIMAL NUMERIC PROGRAM DATA>

<table>
<thead>
<tr>
<th>1 to 2</th>
<th>Data Interface for Data1, and Data2</th>
</tr>
</thead>
</table>

If [<Interface>] is omitted, the default is 1.

**Function**

Sets clock input phase variation in ps units.

The specified value may not be exactly set as is when the ps unit is set, due to the limitation of the setting resolution. The value is adjusted to the most approximate value and set in this event.

**Example**

To set Data1Interface clock face variation to –40 ps:

```
> :DEMUX:CLOCK:PDELAY -40,1
```

**Compatibility**

Not backwards compatible

### :DEMUX:CLOCK:PDELAY? [<interface>]

**Parameter**

<interface> = <DECIMAL NUMERIC PROGRAM DATA>

<table>
<thead>
<tr>
<th>1 to 2</th>
<th>Data Interface for Data1, and Data2</th>
</tr>
</thead>
</table>

If [<Interface>] is omitted, the default is 1.

**Response**

<numeric> = <NR1 NUMERIC RESPONSE DATA>

**Function**

Queries clock input phase variation and units.

**Example**

To query Data1Interface clock phase variation in ps units:

```
> :DEMUX:CLOCK:PDELAY? 1
< -40
```

**Compatibility**

Not backwards compatible
Chapter 7  SCPI Commands

:DEMux:CLOCk:CALibration [<interface>]

Parameter  
<interface> = <DECIMAL NUMERIC PROGRAM DATA>
1 to 2  Data Interface for Data1, and Data2
If [<Interface>] is omitted, the default is 1.

Function  
Calibrates input a clock phase setting.

Example  
To calibrate Data1Interface input clock phase setting:
> :DEMux:CLOCk:CALibration 1

Compatibility  
Not backwards compatible

:DEMux:CLOCk:RELative <boolean>[,<interface>]

Parameter  
<boolean> = <BOOLEAN PROGRAM DATA>
0 or OFF  Reference OFF
1 or ON  Reference ON
<interface> = <DECIMAL NUMERIC PROGRAM DATA>
1 to 2  Data Interface for Data1, and Data2
If [<Interface>] is omitted, the default is 1.

Function  
Sets clock input phase variation reference ON/OFF.

Example  
To set Data1Interface clock input phase variation reference to ON:
> :DEMux:CLOCk:RELative ON,1

Compatibility  
Not backwards compatible

:DEMux:CLOCk:RELative? [<interface>]

Parameter  
<interface> = <DECIMAL NUMERIC PROGRAM DATA>
1 to 2  Data Interface for Data1, and Data2
If [<Interface>] is omitted, the default is 1.

Response  
<boolean> = <NR1 NUMERIC RESPONSE DATA>
0  Reference OFF
1  Reference ON

Function  
Queries clock input phase variation reference setting.

Example  
To query Data1Interface clock input phase variation reference:
> :DEMux:CLOCk:RELative? 1
< 1

Compatibility  
Not backwards compatible
:DEMUX:CLOCk:RDELay<numeric>[,<unit>][,<interface>]

Parameter

<numeric> = <DECIMAL NUMERIC PROGRAM DATA>
(mUI units) –4000 to 4000 –4000 to 4000 mUI/2 mUI Step
(ps units) Converted from the calculated frequency value, based on the
setting resolution of mUI.

<unit> = <CHARACTER PROGRAM DATA>
UI mUI units
PS ps units
(The ps unit is selected when <unit> is omitted.)

<interface> = <DECIMAL NUMERIC PROGRAM DATA>
1 to 2 Data Interface for Data1, and Data2
If [<interface>] is omitted, the default is 1.

Function

Sets clock input phase variation reference value and units.
Setting is performed as an increment from the reference value.
The specified value may not be exactly set as is when the ps unit is set,
due to the limitation of the setting resolution. The value is adjusted to
the most approximate value and set in this event.

Example

To set Data1Interface clock phase variation reference to –1000 mUI:
> :DEMUX:CLOCk:RDELay -1000,UI,1

Compatibility

Not backwards compatible

:DEMUX:CLOCk:RDELay? [<unit>][,<interface>]

Parameter

<unit> = <CHARACTER PROGRAM DATA>
UI mUI units
PS ps units
(The ps unit is selected when <unit> is omitted.)

<interface> = <DECIMAL NUMERIC PROGRAM DATA>
1 to 2 Data Interface for Data1, and Data2
If [<interface>] is omitted, the default is 1.

Response

<numeric> = <NR1 NUMERIC RESPONSE DATA>

Function

Queries clock input phase variation reference value and units.

Example

To query Data1Interface clock phase variation reference UI units:
> :DEMUX:CLOCk:RDELay? UI 1
< -1000

Compatibility

Not backwards compatible
### Chapter 7  SCPI Commands

#### :DEMux:CLOCk:JINPut <boolean>[,<interface>]

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;boolean&gt; = &lt;BOOLEAN PROGRAM DATA&gt;</td>
<td>Jitter Input ON/OFF</td>
</tr>
<tr>
<td>0 or OFF</td>
<td>Jitter Input OFF</td>
</tr>
<tr>
<td>1 or ON</td>
<td>Jitter Input ON</td>
</tr>
<tr>
<td>&lt;interface&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
<td>Data Interface for Data1, and Data2</td>
</tr>
<tr>
<td>1 to 2</td>
<td>If [&lt;Interface&gt;] is omitted, the default is 1.</td>
</tr>
</tbody>
</table>

**Function**
Sets Jitter Input ON/OFF.

**Example**
To set Data1Interface JitterInput to ON:
> :DEMux:Clock:JINPut ON,1

**Compatibility**
Not backwards compatible

#### :DEMux:CLOCk:JINPut? [<interface>]

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;interface&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
<td>Data Interface for Data1, and Data2</td>
</tr>
<tr>
<td>1 to 2</td>
<td>If [&lt;Interface&gt;] is omitted, the default is 1.</td>
</tr>
</tbody>
</table>

**Response**
<numeric> = <NR1 NUMERIC RESPONSE DATA>
| 0 | Jitter Input OFF |
| 1 | Jitter Input ON |

**Function**
Queries Jitter Input ON/OFF.

**Example**
To query Data1Interface Jitter Input ON/OFF:
> :DEMux:Clock:JINPut? 1
< 1

**Compatibility**
Not backwards compatible

#### :DEMux:CLOCk:BAND <sw>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;sw&gt; = &lt;CHARACTER PROGRAM DATA&gt;</td>
<td></td>
</tr>
<tr>
<td>HALFrate</td>
<td>Selects Half Clock Rate</td>
</tr>
<tr>
<td>FULLrate</td>
<td>Selects Full Clock Rate</td>
</tr>
</tbody>
</table>

**Function**
Switches Clock Input Band Switch.

**Example**
To switch Clock Input Band Switch to Half:
> :DEMux:Clock:BAND HALF

**Compatibility**
Not backwards compatible
### :DEMuX:CLOCk:BAND?

<table>
<thead>
<tr>
<th>Response</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>(&lt;\text{sw}&gt; = \langle\text{CHARACTER RESPONSE DATA}\rangle)</td>
<td>(\langle\text{SW}\rangle = \langle\text{CHARACTER RESPONSE DATA}\rangle)</td>
</tr>
<tr>
<td>HALFrate</td>
<td>Selects Half Clock Rate</td>
</tr>
<tr>
<td>FULLrate</td>
<td>Selects Full Clock Rate</td>
</tr>
</tbody>
</table>

**Function**
Queries Clock Input Band Switch.

**Example**
To query Clock Input Band Switch:

```plaintext
>:DEMuX:CLOCk:BAND?
< HALF
```

**Compatibility**
Not backwards compatible
7.9.2.2 Measurement Restart setting field

![Diagram of Measurement Restart setting field]

Figure 7.9.2.2-1 Measurement Restart setting field

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>:DEMux:MEASure:MREStart?</td>
</tr>
</tbody>
</table>
### DEMUX:MEAS: MREStart <data>,<clock> [,<interface>] 

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;data&gt; = &lt;BOOLEAN PROGRAM DATA&gt;</td>
<td>0 or OFF: Set measurement restart at Data Threshold change to OFF. 1 or ON: Set measurement restart at Data Threshold change to ON.</td>
</tr>
<tr>
<td>&lt;clock&gt; = &lt;BOOLEAN PROGRAM DATA&gt;</td>
<td>0 or OFF: Set measurement restart at Clock Delay change to OFF. 1 or ON: Set measurement restart at Clock Delay change to ON.</td>
</tr>
<tr>
<td>&lt;interface&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
<td>1 to 2: Data Interface for Data1, and Data2. If [&lt;Interface&gt;] is omitted, the default is 1.</td>
</tr>
</tbody>
</table>

**Function**: Sets measurement restart conditions.

**Example**: To set Data1Interface measurement restart conditions:

```
> :DEMux:MEAS: MREStart 0,0,1
```

**Compatibility**: Not backwards compatible.

### DEMUX:MEAS: MREStart? [<interface>] 

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;interface&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
<td>1 to 2: Data Interface for Data1, and Data2. If [&lt;Interface&gt;] is omitted, the default is 1.</td>
</tr>
</tbody>
</table>

**Response**

- <data> = <NR1 NUMERIC RESPONSE DATA>  
- <clock> = <NR1 NUMERIC RESPONSE DATA>
  - 0: Measurement restart OFF  
  - 1: Measurement restart ON

**Function**: Queries measurement restart conditions.

**Example**: To query Data1Interface measurement restart conditions:

```
> :DEMux:MEAS: MREStart?
< 0,0
```

**Compatibility**: Not backwards compatible.
7.9.3 Commands related to Result tab

7.9.3.1 Result setting field

Figure 7.9.3.1-1 Result tab
## 7.9 DEMUX Commands

### Table 7.9.3.1-1  Result setting commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1]</td>
<td>Start</td>
<td>:DEMux:MEASure:STARt</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:DEMux:RESult:ZOOM??</td>
</tr>
<tr>
<td></td>
<td>(Query measurement data of error alarm)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(Query window items)</td>
<td></td>
</tr>
</tbody>
</table>

### :DEMux:MEASure:STARt

**Function**
- Starts MU181040A BER measurement in same unit and restarts during measurement.

**Example**
- To start MU181040A BER measurement in same unit:
  ```plaintext
  > :DEMux:MEASure:STARt
  ```

**Compatibility**
- Not backwards compatible

### :DEMux:MEASure:STOP

**Function**
- Stops MU181040A BER measurement in same unit.

**Example**
- To stop MU181040A BER measurement in same unit:
  ```plaintext
  > :DEMux:MEASure:STOP
  ```

**Compatibility**
- Not backwards compatible

### :DEMux:MEASure:EALarm:STATe?

**Response**
- `<numeric> = <NR1 NUMERIC RESPONSE DATA>`
  - 0: Stopped
  - 1: Measuring

**Function**
- Queries measurement progress at MU181040A bit error or alarm in same unit.

**Example**
- To query measurement progress at MU181040A bit error or alarm in same unit:
  ```plaintext
  > :DEMux:MEASure:EALarm:STATe?
  < 0
  ```

**Compatibility**
- Not backwards compatible
Chapter 7  SCPI Commands

:DEMux:RESult:ZOOM <boolean>

Parameter  
<boolean> = <BOOLEAN PROGRAM DATA>
0 or OFF  Zoom out display
1 or ON  Zoom in display

Function  
Sets zooming for measured results display ON/OFF.

Example  
To set zooming for measured results display to ON:
> :DEMux:RESult:ZOOM ON

Compatibility  
Not backwards compatible

:DEMux:RESult:ZOOM?

Response  
<numeric> = <NR1 NUMERIC RESPONSE DATA>
0  Zoom out display
1  Zoom in display

Function  
Queries zooming for measured results display ON/OFF.

Example  
To query zooming for measured results display to ON/OFF:
> :DEMux:RESult:ZOOM?
< 1

Compatibility  
Not backwards compatible

:DEMux:RESult:EALarm:HRESet

Function  
Resets measured results display history.

Example  
To reset measured results display history:
> :DEMux:RESult:EALarm:HRESet

Compatibility  
Not backwards compatible
7.9 DEMUX Commands

:DEMUX:CALCulate:DATA:EALarm? <string>[,<interface>]

Parameter

<string> = <STRING PROGRAM DATA>
“CURRent:<result>” Current data
“LAST:<result>” Measurement end data
Refer to the parameter table for the <result> contents.
<intface> = <DECIMAL NUMERIC PROGRAM DATA>
1 Measured results for Data1
2 Measured results for Data2
3 Measured results for 25G × 2chTotal
If [<interface>] is omitted, 1 is set by default.
3 is disabled at settings other than 25G × 2 ch
Combination.
If there is no Result screen due to restrictions,
the command itself is disabled.

Response

<string> = <STRING RESPONSE DATA>

Table 7.9.3.1-2 Response format

<table>
<thead>
<tr>
<th>Form</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Form1</td>
<td>&quot;XXXXXXX&quot;</td>
<td>For 0 to 9999999</td>
</tr>
<tr>
<td>Integer</td>
<td>&quot;X.XXXXXXX&quot;</td>
<td>For 1.0000E07 to 9.9999E16</td>
</tr>
<tr>
<td>type</td>
<td>&quot;.......&quot;</td>
<td>When no data corresponds to the query</td>
</tr>
<tr>
<td>Form2</td>
<td>&quot;X.XXXE-XX&quot;</td>
<td>For 0.0000E–16 to 1.0000E00</td>
</tr>
<tr>
<td>Fraction</td>
<td>&quot;.......&quot;</td>
<td>When no data corresponds to the query</td>
</tr>
<tr>
<td>type</td>
<td>&quot;XXX.XXXXX&quot;</td>
<td>For 0.0000 to 100.0000</td>
</tr>
<tr>
<td>Form3</td>
<td>&quot;.......&quot;</td>
<td>When no data corresponds to the query</td>
</tr>
<tr>
<td>% type</td>
<td>&quot;XXXXXXXXX&quot;</td>
<td>For 0 to MAX(Hz)</td>
</tr>
<tr>
<td>Form4</td>
<td>&quot;.......&quot;</td>
<td>When no data corresponds to the query</td>
</tr>
<tr>
<td>Frequency</td>
<td>&quot;XX.XXX,XX.XXX&quot;</td>
<td>For –4.000 to +4.000(V)</td>
</tr>
<tr>
<td>Voltage</td>
<td>&quot;.......&quot;</td>
<td>When no data corresponds to the query</td>
</tr>
<tr>
<td>type</td>
<td>&quot;xxxxx.xxx.xx&quot;</td>
<td>mUI units, ps units</td>
</tr>
<tr>
<td>Phase</td>
<td>&quot;.......&quot;</td>
<td>When no data corresponds to the query</td>
</tr>
<tr>
<td>type</td>
<td>&quot;-------&quot;</td>
<td></td>
</tr>
</tbody>
</table>

Function
Queries measured data for parameter.

Example
To query Data Interface1 current measured data total error rate:
< : "0.0000E-16"

To query 25G × 2 chTotal current data Threshold measured data
< "-4.000,------ " (in case of Single-end)

Compatibility
Not backwards compatible
### Table 7.9.3.1-3 Parameter

<table>
<thead>
<tr>
<th>Form</th>
<th>&lt;result&gt;</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>Error Rate</td>
<td>Total INS OMI</td>
<td>ER:TOTal</td>
</tr>
<tr>
<td>Error Count</td>
<td>Total INS OMI</td>
<td>EC:TOTal</td>
</tr>
<tr>
<td>Error Rate</td>
<td>Transition Non Transition</td>
<td>ER:TRANSition</td>
</tr>
<tr>
<td>Error Count</td>
<td>Transition Non Transition</td>
<td>EC:TRANSition</td>
</tr>
<tr>
<td>%EFI</td>
<td>Total</td>
<td>EFI:TOTal</td>
</tr>
<tr>
<td>EI Total</td>
<td>E1:TOTal</td>
<td>Form1</td>
</tr>
<tr>
<td>Frequency(kHz)</td>
<td></td>
<td>FREQuency</td>
</tr>
<tr>
<td>Clock Count</td>
<td></td>
<td>CC:TOTal</td>
</tr>
<tr>
<td>Clock Alarm</td>
<td></td>
<td>AINTerval:CALarm</td>
</tr>
<tr>
<td>Sync Loss</td>
<td></td>
<td>AINTerval:PSLoss</td>
</tr>
<tr>
<td>Data Threshold</td>
<td>THReshold</td>
<td>Form5</td>
</tr>
<tr>
<td>Data Phase</td>
<td>PHASe</td>
<td>Form6</td>
</tr>
<tr>
<td>Threshold EI</td>
<td>&gt;1.0E-3</td>
<td>THReshold:EI:TOTal:E_3</td>
</tr>
<tr>
<td></td>
<td>&gt;1.0E-4</td>
<td>THReshold:EI:TOTal:E_4</td>
</tr>
<tr>
<td></td>
<td>&gt;1.0E-5</td>
<td>THReshold:EI:TOTal:E_5</td>
</tr>
<tr>
<td></td>
<td>&gt;1.0E-6</td>
<td>THReshold:EI:TOTal:E_6</td>
</tr>
<tr>
<td></td>
<td>&gt;1.0E-7</td>
<td>THReshold:EI:TOTal:E_7</td>
</tr>
<tr>
<td></td>
<td>&gt;1.0E-8</td>
<td>THReshold:EI:TOTal:E_8</td>
</tr>
<tr>
<td></td>
<td>≤1.0E-8</td>
<td>THReshold:EI:TOTal:UE_8</td>
</tr>
<tr>
<td>Threshold %EFI</td>
<td>&gt;1.0E-3</td>
<td>THReshold:EFI:TOTal:E_3</td>
</tr>
<tr>
<td></td>
<td>&gt;1.0E-4</td>
<td>THReshold:EFI:TOTal:E_4</td>
</tr>
<tr>
<td></td>
<td>&gt;1.0E-5</td>
<td>THReshold:EFI:TOTal:E_5</td>
</tr>
<tr>
<td></td>
<td>&gt;1.0E-6</td>
<td>THReshold:EFI:TOTal:E_6</td>
</tr>
<tr>
<td></td>
<td>&gt;1.0E-7</td>
<td>THReshold:EFI:TOTal:E_7</td>
</tr>
<tr>
<td></td>
<td>&gt;1.0E-8</td>
<td>THReshold:EFI:TOTal:E_8</td>
</tr>
<tr>
<td></td>
<td>≤1.0E-8</td>
<td>THReshold:EFI:TOTal:UE_8</td>
</tr>
<tr>
<td>G.821</td>
<td>ES</td>
<td>G821:ES2:TOTal</td>
</tr>
<tr>
<td></td>
<td>EFS</td>
<td>G821:EFS2:TOTal</td>
</tr>
<tr>
<td></td>
<td>SES</td>
<td>G821:SES2:TOTal</td>
</tr>
<tr>
<td></td>
<td>DM</td>
<td>G821:DM2:TOTal</td>
</tr>
<tr>
<td></td>
<td>US</td>
<td>G821:US2:TOTal</td>
</tr>
<tr>
<td></td>
<td>%ES</td>
<td>G821:ES:TOTal</td>
</tr>
<tr>
<td></td>
<td>%EFS</td>
<td>G821:EFS:TOTal</td>
</tr>
<tr>
<td></td>
<td>%SES</td>
<td>G821:SES:TOTal</td>
</tr>
<tr>
<td></td>
<td>%DM</td>
<td>G821:DM:TOTal</td>
</tr>
<tr>
<td></td>
<td>%US</td>
<td>G821:US:TOTal</td>
</tr>
</tbody>
</table>
7.9 DEMUX Commands

**:DEMux:DISPLAY:SETTING <setting>**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;setting&gt; = &lt;CHARACTER PROGRAM DATA&gt;</td>
<td>Sets displayed items on the Result tab:</td>
</tr>
<tr>
<td>DATA1</td>
<td>Display Data1 Interface items</td>
</tr>
<tr>
<td>DATA2</td>
<td>Display Data2 Interface items</td>
</tr>
<tr>
<td>AUTOadjust</td>
<td>Display AutoAdjust items</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Function</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Displays the Data1 Interface Items.</td>
<td></td>
</tr>
<tr>
<td>&gt; :DEMux:DISPLAY:SETTING DATA1</td>
<td></td>
</tr>
</tbody>
</table>

| Compatibility | Not backwards compatible |

**:DEMux:CALCulate:DATA:MONitor? <item>**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;item&gt; = &lt;STRING PROGRAM DATA&gt;</td>
<td>Queries real-time occurrence status of monitor items specified by parameters.</td>
</tr>
<tr>
<td>&quot;BIT:TOTal&quot;</td>
<td>Bit Error (Total Error)</td>
</tr>
<tr>
<td>&quot;PSLoss&quot;</td>
<td>Pattern Sync Loss</td>
</tr>
<tr>
<td>&quot;CALarm&quot;</td>
<td>Clock Alarm</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Response</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;string&gt; = &lt;STRING RESPONSE DATA&gt;</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 7.9.3.1-4 Response format</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Form</strong></td>
</tr>
<tr>
<td>Form1</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Function</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Queries real-time occurrence status of monitor items specified by parameters.</td>
<td></td>
</tr>
<tr>
<td>To query Bit Error occurrence status:</td>
<td></td>
</tr>
<tr>
<td>&gt; :DEMux:CALCulate:DATA:MONitor? &quot;BIT TOTal&quot;</td>
<td></td>
</tr>
<tr>
<td>&lt; &quot;Occur&quot;</td>
<td></td>
</tr>
</tbody>
</table>

| Compatibility | Not backwards compatible |
7.10 Jitter Commands

This section describes the setting and query commands for the MU181500B Jitter Modulation Source.

Before executing a setting/query command for the MU181500B, specify the slot number of the module to be remotely controlled, by using the :MODule:ID command. Refer to Section 7.1 "Common Commands" and 9.1 "Basic Operation of Device Messages" for how to specify a slot number with the :MODule:ID command.

7.10.1 List of commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Command Header 1</th>
<th>Command Header 2</th>
<th>Command Header 3</th>
<th>Command Header 4</th>
<th>Command/Query</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>:OUTPut</td>
<td>:AUX</td>
<td>:JCondition</td>
<td>Q</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td>:REFClock</td>
<td>C/Q</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td>:SELect</td>
<td>Q</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td>:SUBRateclock</td>
<td>C/Q</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td>:AMPLitude</td>
<td>C/Q</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td>:MONitor</td>
<td>Q</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
<td>:CLOck</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
<td>:FREQuency</td>
<td>C/Q</td>
<td></td>
<td>Synthesizer compatible</td>
</tr>
<tr>
<td>9</td>
<td></td>
<td></td>
<td>:OFFset</td>
<td>PPM</td>
<td>C/Q</td>
<td>Synthesizer compatible</td>
</tr>
<tr>
<td>10</td>
<td></td>
<td></td>
<td>:RCLock</td>
<td></td>
<td>C/Q</td>
<td>Synthesizer compatible</td>
</tr>
<tr>
<td>11</td>
<td>:SOURce</td>
<td>:JITTer</td>
<td>:BUJ</td>
<td>:AMPLitude</td>
<td>C/Q</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td></td>
<td></td>
<td></td>
<td>:BITRate</td>
<td>C/Q</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td></td>
<td></td>
<td></td>
<td>:ENABle</td>
<td>C/Q</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td></td>
<td></td>
<td></td>
<td>:LPFilter</td>
<td>C/Q</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td></td>
<td></td>
<td></td>
<td>:PRBS</td>
<td>C/Q</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td></td>
<td></td>
<td></td>
<td>:EXTJitter</td>
<td>C/Q</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td></td>
<td></td>
<td></td>
<td>:AMPLitude</td>
<td>C/Q</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td></td>
<td></td>
<td></td>
<td>:DEFault</td>
<td>C</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td></td>
<td></td>
<td></td>
<td>:ENABle</td>
<td>C/Q</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td></td>
<td></td>
<td></td>
<td>:FILTer</td>
<td>C/Q</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td></td>
<td></td>
<td></td>
<td>:HFAMplitude</td>
<td>C/Q</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td></td>
<td></td>
<td></td>
<td>:HPFFilter</td>
<td>C/Q</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td></td>
<td></td>
<td></td>
<td>:LFAMplitude</td>
<td>C/Q</td>
<td></td>
</tr>
<tr>
<td>24</td>
<td></td>
<td></td>
<td></td>
<td>:LPFilter</td>
<td>C/Q</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td></td>
<td></td>
<td></td>
<td>:MONitor</td>
<td>Q</td>
<td></td>
</tr>
</tbody>
</table>
### Table 7.10.1-1 MU181500B Command List (Cont'd)

<table>
<thead>
<tr>
<th>No.</th>
<th>Command Header 1</th>
<th>Command Header 2</th>
<th>Command Header 3</th>
<th>Command Header 4</th>
<th>Command/Query</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>27</td>
<td></td>
<td></td>
<td></td>
<td>:ENABLE</td>
<td>C/Q</td>
<td></td>
</tr>
<tr>
<td>28</td>
<td></td>
<td></td>
<td></td>
<td>:FREQuency</td>
<td>C/Q</td>
<td></td>
</tr>
<tr>
<td>29</td>
<td></td>
<td></td>
<td>:SSC</td>
<td>:DEViation</td>
<td>C/Q</td>
<td></td>
</tr>
<tr>
<td>30</td>
<td></td>
<td></td>
<td></td>
<td>:ENABLE</td>
<td>C/Q</td>
<td></td>
</tr>
<tr>
<td>31</td>
<td></td>
<td></td>
<td></td>
<td>:FREQuency</td>
<td>C/Q</td>
<td></td>
</tr>
<tr>
<td>32</td>
<td></td>
<td></td>
<td></td>
<td>:TYPE</td>
<td>C/Q</td>
<td></td>
</tr>
<tr>
<td>33</td>
<td></td>
<td>:OUTPut</td>
<td>:PATA</td>
<td>:JOVerload</td>
<td>Q</td>
<td></td>
</tr>
<tr>
<td>34</td>
<td></td>
<td></td>
<td></td>
<td>:MONitor</td>
<td>Q</td>
<td></td>
</tr>
<tr>
<td>35</td>
<td></td>
<td></td>
<td></td>
<td>:SELect</td>
<td>C/Q</td>
<td></td>
</tr>
<tr>
<td>36</td>
<td>:SYSTem</td>
<td>:INPut</td>
<td>:CSELect</td>
<td></td>
<td>C/Q</td>
<td></td>
</tr>
<tr>
<td>37</td>
<td></td>
<td></td>
<td></td>
<td>:MODule</td>
<td>Q</td>
<td></td>
</tr>
<tr>
<td>38</td>
<td>:MMEMemory</td>
<td>:RECall</td>
<td></td>
<td></td>
<td>C</td>
<td></td>
</tr>
<tr>
<td>39</td>
<td></td>
<td>:STORe</td>
<td></td>
<td></td>
<td>C</td>
<td></td>
</tr>
</tbody>
</table>
7.10.2 Clock Source Settings Commands

Figure 7.10.2-1  Clock Source Settings

Table 7.10.2-1  Clock Source Settings Commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting item</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>:SYSTem:INPut:CSELeCt?</td>
</tr>
<tr>
<td>[3]</td>
<td>Center Frequency/Input Clock Frequency</td>
<td>:OUTPut:CLOck:FREQuency</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:OUTPut:CLOck:FREQuency?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:OUTPut:CLOck:OFFSet:PPM?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:OUTPut:RCLock:SELeCt?</td>
</tr>
</tbody>
</table>
### :SYSTem:INPut:CSELect <input>

**Parameter**

- `<input>` = <CHARACTER PROGRAM DATA>
- `EXTernal` External synthesizer
- `INTernal [1 to 7]` Internal synthesizer

**Note:**

`INTernal [1 to 7]` specifies the unit and slot numbers from the smallest as 1 to 7 when there is more than one built-in synthesizer. If [1 to 7] is omitted, 1 is assumed.

**Function**

Selects synthesizer for clock input source.

**Example**

To choose #2 built-in synthesizer as clock input source.

```
> :SYSTem:INPut:CSELect INT2
```

**Compatibility**

Incompatible with existing models.

### :SYSTem:INPut:CSELect?

**Response**

- `<numeric>` = <CHARACTER RESPONSE DATA>
- `EXT, INT1 to 7`
- `<info>` = <STRING RESPONSE DATA>
- "X:Y Z"
- X:Unit No. 1 to 4
- Y:Slot No. 1 to 6
- Z:Module name MU181000A or MU181000B

**Function**

Queries synthesizer for clock input source.

**Example**

```
> :SYSTem:INPut:CSELect?
< INT2,"1:4 MU181000A"
```

**Compatibility**

Incompatible with existing models.

### :SYSTem:INPut:CSELect:MODule?

**Response**

- `<info>,<info1>,<info2>,...,<info7>` = <STRING RESPONSE DATA>
- `<info>` = "X:Y Z"
- X:Unit No. 1 to 4
- Y:Slot No. 1 to 6
- Z:Module name MU181000A or MU181000B

**Function**

Queries system built-in synthesizer.

**Example**

```
> :SYSTem:INPut:CSELect:MODule?
< "1:2 MU181000A", "1:4 MU181000A"
```

**Compatibility**

Incompatible with existing models.
Chapter 7  SCPI Commands

:OUTPut:CLOCK:FREQuency <numeric>

Parameter  
<numeric> = <DECIMAL NUMERIC PROGRAM DATA>
12800002 to 15000000   12,800,002 to 15,000,000 kHz/2 kHz step
6400001 to 12500000   6,400,001 to 12,500,000 kHz/1 kHz step
3200001 to 6250000   3,200,001 to 6,250,000 kHz/1 kHz step
1600001 to 3125000   1,600,001 to 3,125,000 kHz/1 kHz step
800001 to 1562500   800,000 to 1,562,500 kHz/1 kHz step

Note:  
The settings steps are in accordance with the screen control.

Function  
Sets operation frequency of tracking built-in synthesizer (MU181000A/B).

Example  
To set operation frequency of tracking built-in synthesizer (MU181000A/B) to 12,500,000 kHz:
> :OUTPut:CLOCK:FREQuency 12500000

Compatibility  
Compatible with MU181000A/B

:OUTPut:CLOCK:FREQuency?

Response  
When ClockSource is not External.  
<numeric> = <NR1 NUMERIC RESPONSE DATA>
800001 to 15000000   800,001 to 15,000,000 kHz

When ClockSource is External.  
<string> = <STRING RESPONSE DATA>

Format | Explanation
------ | ---------------
"XX.XXXXX"  | 100000 to 15000000  | 100,001 to 15,000,000 kHz
"-----"  | When no data corresponds to the query (Out of range)

Function  
Queries operation frequency of tracking built-in synthesizer.

Example  
> :OUTPut:CLOCK:FREQuency?
< 12500000

Compatibility  
Compatible with MU181000A/B

:OUTPut:CLOCK:OFFSet:PPM <numeric>

Parameter  
<numeric> = <DECIMAL NUMERIC PROGRAM DATA>
-1000 to 1000   -1,000 to +1,000 ppm/Step 1 ppm

Function  
Sets offset of frequency (ppm) currently output from tracking built-in synthesizer.

Example  
To set offset of currently output frequency to 0 ppm:
> :OUTPut:CLOCK:OFFSet:PPM 0

Compatibility  
Compatible with MU181000A/B
7.10 Jitter Commands

:OUTPut:CLOCk:OFFSet:PPM?

Response  
<numeric> = <NR1 NUMERIC RESPONSE DATA>
−1000 to 1000  −1,000 to +1,000 ppm

Function  
Queries offset (ppm) of frequency currently output from tracking built-in synthesizer.

Example  
> :OUTPut:CLOCk:OFFSet:PPM?
< 0

Compatibility  
Compatible with MU181000A/B

:OUTPut:RCLock:SELect <clock>

Parameter  
<clock> = <CHARACTER PROGRAM DATA>
INTernal  Internal reference signal 10 MHz
EXTernal10  External reference signal 10 MHz

Function  
Selects either built-in or external input for 10 MHz reference signal of tracking built-in synthesizer.

Example  
To set 10 MHz reference signal to external input:
> :OUTPut:RCLock:SELect EXT10

Compatibility  
Compatible with MU181000A/B

:OUTPut:RCLock:SELect?

Response  
<clock> = <CHARACTER RESPONSE DATA>
INT, EXT10

Function  
Queries setting for 10 MHz reference signal of tracking built-in synthesizer.

Example  
> :OUTPut:RCLock:SELect?
< EXT10

Compatibility  
Compatible with MU181000A/B
7.10.3 Jitter Setting Commands

7.10.3.1 SJ/SJ2 Setting

Figure 7.10.3.1-1 SJ/SJ2 Settings

Table 7.10.3.1-1 SJ/SJ2 Setting Commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting item</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>:SOURce:JITTer:SJ[2]:ENABle?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SOURce:JITTer:SJ[2]:FREQuency?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SOURce:JITTer:SJ[2]:AMPLitude?</td>
</tr>
</tbody>
</table>
7.10 Jitter Commands

:SOURce:JITTer: SJ[2]: ENABle <boolean>

Parameter

<boolean> = <BOOLEAN PROGRAM DATA>
OFF or 0  SJ/SJ2 OFF
ON or 1   SJ/SJ2 ON

Function

Sets SJ or SJ2 ON/OFF.

Note:


Example

To set SJ setting to ON:

> :SOURce:JITTer: SJ: ENABle ON

Compatibility

Incompatible with existing models.

:SOURce:JITTer: SJ[2]: FREQuency <numeric>

Parameter

<numeric> = <DECIMAL NUMERIC PROGRAM DATA>
10 to 250000000  10 to 250,000,000 Hz
Setting Step

10[Hz] ≤ fmod ≤ 10[kHz]  1 Hz Step
10[kHz] ≤ fmod ≤ 100[kHz]  10 Hz Step
100[kHz] ≤ fmod ≤ 1[MHz]  100 Hz Step
1[MHz] ≤ fmod ≤ 10[MHz]  1 kHz Step
10[MHz] ≤ fmod ≤ 100[MHz]  10 kHz Step
100[MHz] ≤ fmod ≤ 250[MHz]  100 kHz Step

Function

Sets SJ or SJ2 modulation frequency and units.

Example

To set SJ2 modulation frequency to 10 kHz:

> :SOURce:JITTer: SJ2: FREQuency 10000

Compatibility

Incompatible with existing models.
Chapter 7  SCPI Commands

:SOURce:JITTer:SJ[2]:FREQuency?

Response  
<numeric> = <NR2 NUMERIC RESPONSE DATA>
10 to 250000000 10 to 250,000,000 Hz

Function  
Queries SJ or SJ2 modulation frequency.

Example  
To query SJ2 modulation frequency:
> :SOURce:JITTer:SJ2:FREQuency?
< 10000

Compatibility  
Incompatible with existing models.

:SOURce:JITTer:SJ[2]:AMPLitude <numeric>

Parameter  
<numeric> = <DECIMAL NUMERIC PROGRAM DATA>

<table>
<thead>
<tr>
<th>Data Generation</th>
<th>Half-rate(MUX)</th>
<th>Full-rate(PPG), Full-rate(MUX)</th>
<th>Quarter-rate (MUX)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Setting Range (UIp-p)</td>
<td>Step (UI)</td>
<td>Setting Range (UIp-p)</td>
</tr>
<tr>
<td>10 Hz to 1 MHz</td>
<td>0 to 50.000</td>
<td>0.002</td>
<td>0 to 40.000</td>
</tr>
<tr>
<td>1.001 MHz to 10 MHz</td>
<td>0 to 10.000</td>
<td>0.002</td>
<td>0 to 8.000</td>
</tr>
<tr>
<td>10.01 MHz to 250 MHz</td>
<td>0 to 0.550</td>
<td>0.002</td>
<td>0 to 0.500</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Data Generation</th>
<th>32G PPG</th>
<th>64G MUX</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Setting Range (UIp-p)</td>
<td>Setting Range (UIp-p)</td>
</tr>
<tr>
<td>10 Hz to 100 kHz</td>
<td>0 to 2000.000</td>
<td>0.002</td>
</tr>
<tr>
<td>100.1 kHz to 1 MHz</td>
<td>0 to 200.000</td>
<td>0.002</td>
</tr>
<tr>
<td>1.001 MHz to 10 MHz</td>
<td>0 to 16.000</td>
<td>0.002</td>
</tr>
<tr>
<td>10.01 MHz to 250 MHz</td>
<td>0 to 1.0</td>
<td>0.002</td>
</tr>
</tbody>
</table>

Note:

The setting step varies according to the Data Generator setting.
For details on setting ranges and steps, refer to 1.3.2 “Jitter Modulation Performance” in MU181500B Jitter Modulation Source Operation Manual.

Function  
Sets SJ or SJ2 modulation amount.

Example  
To set SJ modulation amount to 1.10 UIp-p:
> :SOURCE:JITTer:SJ:AMPLitude 1.10

Compatibility  
Incompatible with existing models.
7.10 Jitter Commands

:SOURce:JITTer: SJ[2]: AMPLitude?

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;numeric&gt; = &lt;NR2 NUMERIC RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0 to 2000.000</td>
</tr>
<tr>
<td></td>
<td>0 to 2000.000 UI</td>
</tr>
</tbody>
</table>

Function Queries SJ or SJ2 modulation amount.

Example To query SJ modulation amount:
> :SOURce:JITTer: SJ: AMPLitude?
< 1.10

Compatibility Incompatible with existing models.

:SOURce:JITTer: SJ[2]: PSAMplitude?

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;numeric&gt; = &lt;NR2 NUMERIC RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0 to xx.xxx</td>
</tr>
<tr>
<td></td>
<td>0 to xx.xxx ps p-p</td>
</tr>
</tbody>
</table>

Function Queries modulation amount (ps p-p) of SJ or SJ2.

Example To query SJ's modulation amount (ps p-p):
> :SOURce:JITTer: SJ: PSAMplitude?
< 1.10

Compatibility Incompatible with existing models.
### 7.10.3.2 SSC Settings

![ SSC Settings Diagram ](image)

#### Figure 7.10.3.2-1  SSC Settings

#### Table 7.10.3.2-1  SSC Setting Commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting item</th>
<th>Command</th>
</tr>
</thead>
</table>
| [1] | SSC ON/OFF   | :SOURce·JITTer·SSC·ENABle  
|     |              | :SOURce·JITTer·SSC·ENABle? |
| [2] | SSC Type     | :SOURce·JITTer·SSC·TYPE    
|     |              | :SOURce·JITTer·SSC·TYPE?   |
| [3] | SSC Frequency| :SOURce·JITTer·SSC·FREQuency |
|     |              | :SOURce·JITTer·SSC·FREQuency? |
| [4] | SSC Deviation| :SOURce·JITTer·SSC·DEViation |
|     |              | :SOURce·JITTer·SSC·DEViation? |
7.10 Jitter Commands

:SOURce:JITTer:SSC:ENABle <boolean>

Parameter

<boolean> = <BOOLEAN PROGRAM DATA>
OFF or 0      SSC OFF
ON or 1       SSC ON

Function
Sets SSC ON/OFF.

Example
To set SSC to ON:
> :SOURce:JITTer:SSC:ENABle ON

Compatibility
Incompatible with existing models.

:SOURce:JITTer:SSC:ENABle?

Response

<numERIC> = <NR1 NUMERIC RESPONSE DATA>
0   SSC OFF
1   SSC ON

Function
Queries SSC ON/OFF status.

Example
> :SOURce:JITTer:SSC:ENABle?
< 1

Compatibility
Incompatible with existing models.

:SOURce:JITTer:SSC:TYPE <type>

Parameter

<type> = <CHARACTER PROGRAM DATA>
DOWN
CENTer
UP

Function
Selects SSC impression type.

Example
To set SSC impression type to DOWN:
> :SOURce:JITTer:SSC:TYPE DOWN

Compatibility
Incompatible with existing models.

:SOURce:JITTer:SSC:TYPE?

Response

?type> = <CHARACTER RESPONSE DATA>
DOWN, CENT, UP

Function
Queries SSC impression type.

Example
> :SOURce:JITTer:SSC:TYPE?
< DOWN

Compatibility
Incompatible with existing models.
Chapter 7  SCPI Commands

:SOURce:JITTer:SSC:FREQuency <numeric>

Parameter  
<numeric> = <DECIMAL NUMERIC PROGRAM DATA>
28000 to 37000  28000 to 37000 Hz/1 Hz step

Function  
Sets SSC modulation frequency.

Example  
To set SSC modulation frequency to 30 kHz:

> :SOURce:JITTer:SSC:FREQuency 30000

Compatibility  
Incompatible with existing models.

:SOURce:JITTer:SSC:FREQuency?

Response  
<numeric> = <NR2 NUMERIC RESPONSE DATA>
28000 to 37000  28000 to 37000 Hz

Function  
Queries SSC modulation frequency.

Example  

> :SOURce:JITTer:SSC:FREQuency?
< 30000

Compatibility  
Incompatible with existing models.

:SOURce:JITTer:SSC:DEViation <numeric>

Parameter  
<numeric> = <DECIMAL NUMERIC PROGRAM DATA>
0 to 7000  0 to 7000 ppm/1 ppm Step

Function  
Sets SSC frequency deviation.

Example  
To set SSC frequency deviation to 2000 ppm:

> :SOURce:JITTer:SSC:DEViation 2000

Compatibility  
Incompatible with existing models.

:SOURce:JITTer:SSC:DEViation?

Response  
<numeric> = <NR2 NUMERIC RESPONSE DATA>
0 to 7000  0 to 7000 ppm

Function  
Queries SSC frequency deviation.

Example  

> :SOURce:JITTer:SSC:DEViation?
< 2000

Compatibility  
Incompatible with existing models.
7.10.3.3 RJ Settings

Figure 7.10.3.3-1 RJ Settings

Figure 7.10.3.3-2 RJ Settings (2)
Table 7.10.3.3-1  RJ Setting Commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Item</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1]</td>
<td>RJ ON/OFF</td>
<td>:SOURce::JITTer::RJ::ENABle</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SOURce::JITTer::RJ::ENABle?</td>
</tr>
<tr>
<td>[2]</td>
<td>Jitter Amplitude (UIp-p)</td>
<td>:SOURce::JITTer::RJ::AMPLitude</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SOURce::JITTer::RJ::AMPLitude?</td>
</tr>
<tr>
<td>[3]</td>
<td>Jitter Amplitude (UIrms)</td>
<td>:SOURce::JITTer::RJ::MONitor?</td>
</tr>
<tr>
<td>[4]</td>
<td>Filter</td>
<td>:SOURce::JITTer::RJ::FILTer</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SOURce::JITTer::RJ::FILTer?</td>
</tr>
<tr>
<td>[5]</td>
<td>High Pass Filter</td>
<td>:SOURce::JITTer::RJ::HPFilter</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SOURce::JITTer::RJ::HPFilter?</td>
</tr>
<tr>
<td>[6]</td>
<td>Low Pass Filter</td>
<td>:SOURce::JITTer::RJ::LPFilter</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SOURce::JITTer::RJ::LPFilter?</td>
</tr>
<tr>
<td>[7]</td>
<td>Jitter Amplitude LF (ps rms)</td>
<td>:SOURce::JITTer::RJ::LFAMplitude</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SOURce::JITTer::RJ::LFAMplitude?</td>
</tr>
<tr>
<td>[8]</td>
<td>Jitter Amplitude HF (ps rms)</td>
<td>:SOURce::JITTer::RJ::HFAMplitude</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SOURce::JITTer::RJ::HFAMplitude?</td>
</tr>
<tr>
<td>[9]</td>
<td>Amplitude LF/HF Default</td>
<td>:SOURce::JITTer::RJ::DEFault</td>
</tr>
<tr>
<td>[10]</td>
<td>Jitter Amplitude Query</td>
<td>:SOURce::JITTer::RJ::PSAMplitude?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SOURce::JITTer::RJ::RMSConvert?</td>
</tr>
</tbody>
</table>

:SOURce::JITTer::RJ::ENABle <boolean>

Parameter:  
<boolean> = <BOOLEAN PROGRAM DATA>
OFF or 0  RJ OFF
ON or 1   RJ ON

Function:  
Sets RJ ON/OFF.

Example:   
To set RJ setting to ON:
> :SOURce::JITTer::RJ::ENABle ON

Compatibility:  
Incompatible with existing models.

:SOURce::JITTer::RJ::ENABle?

Response:  
<numeric> = <NR1 NUMERIC RESPONSE DATA>
0   RJ OFF
1   RJ ON

Function:  
Queries RJ ON/OFF status.

Example:   
> :SOURce::JITTer::RJ::ENABle?
< 1

Compatibility:  
Incompatible with existing models.
### :SOURce:JITTer:RJ:AMPLitude <numeric>

**Parameter**

- **Data Generator**
- **Clock Frequency**
- **Half-rate(MUX)**
- **Full-rate(PPG), Full-rate(MUX)**
- **Quarter-rate(MUX)**

<table>
<thead>
<tr>
<th>Clock Frequency</th>
<th>Half-rate(MUX)</th>
<th>Full-rate(PPG), Full-rate(MUX)</th>
<th>Quarter-rate(MUX)</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt; 2.5 GHz</td>
<td>Setting Range (UIp-p)</td>
<td>0 to 0.2f 0.004</td>
<td>Setting Range (UIp-p)</td>
</tr>
<tr>
<td>≥ 2.5 GHz</td>
<td>0 to 0.500 0.004</td>
<td>0 to 0.500 0.002</td>
<td>0 to 0.496 0.008</td>
</tr>
</tbody>
</table>

- **f**: Clock Frequency
- When F = 1 GHz, the upper limit is 0.2 UIp-p.
- When F = 100 MHz, the upper limit is 0.02 UIp-p.

**Note:**

The setting step varies according to the Data Generator setting.
For details on setting ranges and steps, refer to 1.3.2 “Jitter Modulation Performance” in *MU181500B Jitter Modulation Source Operation Manual*.

**Function**

Sets RJ amplitude amount.

**Example**

To set RJ amplitude amount to 0.5 UI:

```
> :SOURce:JITTer:RJ:AMPLitude 0.5
```

**Compatibility**

Incompatible with existing models.

### :SOURce:JITTer:RJ:AMPLitude?

**Response**

- <numeric> = <NR2 NUMERIC RESPONSE DATA>
  - 0 to 0.500
  - 0 to 0.500 UIp-p

**Function**

Queries RJ amplitude amount.

**Example**

```
> :SOURce:JITTer:RJ:AMPLitude?
< 0.500
```

**Compatibility**

Incompatible with existing models.

### :SOURce:JITTer:RJ:MONitor?

**Response**

- <numeric> = <NR2 NUMERIC RESPONSE DATA>
  - 0 to X.XXXX
  - 0 to 0.500 UIp-p RMS conversion value

**Function**

Queries RJ amplitude amount as RMS value.

**Example**

```
> :SOURce:JITTer:RJ:MONitor?
< 0.0300
```

**Compatibility**

Incompatible with existing models.
**Chapter 7  SCPI Commands**

### :SOURce:JITTer:RJ:FILTer <type>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;type&gt; = &lt;CHARACTER PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>USER</td>
<td>User settings</td>
</tr>
<tr>
<td>PCIexpress1</td>
<td>PCI-express Data clocked settings</td>
</tr>
<tr>
<td>PCIexpress2</td>
<td>PCI-express Common Ref.clock settings</td>
</tr>
</tbody>
</table>

**Function**
Selects RJ filter standard.

**Example**
To set RJ filter standard user setting:
> :SOURce:JITTer:RJ:FILTer USER

**Compatibility**
Incompatible with existing models.

### :SOURce:JITTer:RJ:FILTer?

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;type&gt; = &lt;CHARACTER RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>USER, PCI1, PCI2</td>
<td></td>
</tr>
</tbody>
</table>

**Function**
Queries RJ filter standard.

**Example**
> :SOURce:JITTer:RJ:FILTer?
< USER

**Compatibility**
Incompatible with existing models.

### :SOURce:JITTer:RJ:HPFilter <setting>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;setting&gt; = &lt;CHARACTER PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>OFF</td>
<td>OFF</td>
</tr>
<tr>
<td>M_10</td>
<td>10 MHz High-pass filter</td>
</tr>
<tr>
<td>M_20</td>
<td>20 MHz High-pass filter</td>
</tr>
</tbody>
</table>

**Function**
Selects high-pass filter for use at RJ.

**Example**
To select 10 MHz high-pass filter for RJ:
> :SOURce:JITTer:RJ:HPFilter M_10

**Compatibility**
Incompatible with existing models.

### :SOURce:JITTer:RJ:HPFilter?

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;setting&gt; = &lt;CHARACTER RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>OFF, M_10, M_20</td>
<td></td>
</tr>
</tbody>
</table>

**Function**
Queries high-pass filter for use at RJ.

**Example**
> :SOURce:JITTer:RJ:HPFilter?
< M_10

**Compatibility**
Incompatible with existing models.
7.10 Jitter Commands

:SOURce:JITTer:RJ:LPFilter <setting>

Parameter  
<setting> = <CHARACTER PROGRAM DATA>
- OFF   OFF
- M_100  100 MHz Low-pass filter

Function   Selects low-pass filter for use at RJ.
Example    To select 100 MHz low-pass filter for use at RJ:
> :SOURce:JITTer:RJ:LPFilter M_100

Compatibility Incompatible with existing models.

:SOURce:JITTer:RJ:LPFilter?

Response   <setting> = <CHARACTER RESPONSE DATA>
- OFF, M_100

Function   Queries low-pass filter for use at RJ.
Example    > :SOURce:JITTer:RJ:LPFilter?
< M_100

Compatibility Incompatible with existing models.

:SOURce:JITTer:RJ:LFAMplitude <numeric>

Parameter  <numeric> = <DECIMAL NUMERIC PROGRAM DATA>
- 0.0 to 8.8  0.0 to 8.8 ps rms/0.1 ps Step

Note: When the Data Generator setting is Half-rate Clock (MUX), the setting step becomes 0.2 ps. For details on setting ranges and steps, refer to 1.3.2 “Jitter Modulation Performance” in MU181500B Jitter Modulation Source Operation Manual.

Function   Sets LF modulation amount.
Example    To set LF modulation amount to 0.5 ps:
> :SOURce:JITTer:RJ:LFAMplitude 0.5

Compatibility Incompatible with existing models.

:SOURce:JITTer:RJ:LFAMplitude?

Response   <numeric> = <NR2 NUMERIC RESPONSE DATA>
- 0.0 to 8.8  0.0 to 8.8 rms p-p

Function   Queries LF modulation amount of RJ.
Example    > :SOURce:JITTer:RJ:LFAMplitude?
< 0.5

Compatibility Incompatible with existing models.
Chapter 7  SCPI Commands

:SOURce:JITTer:RJ:HFAMplitude <numeric>

Response  

<numeric> = <DECIMAL NUMERIC PROGRAM DATA>  
0.0 to 8.8  0.0 to 8.8 ps rms/0.1 ps Step  

Note:  

When the Data Generator setting is Half-rate Clock (MUX), the setting step becomes 0.2 ps. For details on setting ranges and steps, refer to 1.3.2 “Jitter Modulation Performance” in MU181500B Jitter Modulation Source Operation Manual.

Function  

Sets HF modulation amount of RJ.  

Example  

To set HF modulation amount of RJ to 0.5 ps rms:  

> :SOURce:JITTer:RJ:HFAMplitude 0.5  

Compatibility  

Incompatible with existing models.

:SOURce:JITTer:RJ:HFAMplitude?

Response  

<numeric> = <NR2 NUMERIC RESPONSE DATA>  
0.0 to 8.8  0.0 to 8.8 ps rms  

Function  

Queries HF modulation amount of RJ.  

Example  

>SOURce:JITTer:RJ:HFAMplitude?  

< 0.5  

Compatibility  

Incompatible with existing models.

:SOURce:JITTer:RJ:DEFault

Parameter  

None  

Function  

Sets RJ LF and HF modulation settings to default values.  

Example  

>SOURce:JITTer:RJ:DEFault  

Compatibility  

Incompatible with existing models.

:SOURce:JITTer:RJ:PSAMplitude? <unit>

Parameter  

<unit> = <CHARACTER PROGRAM DATA>  
PP  ps p-p  
RMS  ps rms  

Response  

<numeric> = <NR2 NUMERIC RESPONSE DATA>  
0 to xx.xx  0 to xx.xx ps p-p  
0 to xx.xx  0 to xx.xx ps rms  

Function  

Queries RJ’s amplitude amount (ps p-p or ps rms).  

Example  

To query RJ’s amplitude amount (ps p-p):  

>SOURce:JITTer:RJ:PSAMplitude? PP  

< 0.500  

Compatibility  

Incompatible with existing models.
### 7.10 Jitter Commands

**:SOURce:JITTer:RJ:RMSConvert <numeric>**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;numeric&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>10 to 16 E–10 to E–16 BER of p·p/rms Conversion Coefficient</td>
</tr>
</tbody>
</table>

**Function**
Specifies a BER value for p·p/rms conversion of RJ modulation.

**Example**
To specify a BER value for p·p/rms conversion of RJ modulation to E-15:

> :SOURce:JITTer:RJ:RMSConvert 15

**Compatibility**
Incompatible with existing models.

**:SOURce:JITTer:RJ:RMSConvert?**

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;numeric&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>10 to 16 E–10 to E–16 BER of p·p/rms Conversion Coefficient</td>
</tr>
</tbody>
</table>

**Function**
Queries the BER value specified for p·p/rms conversion of RJ modulation.

**Example**

> :SOURce:JITTer:RJ:RMSConvert?

< 15

**Compatibility**
Incompatible with existing models.
7.10.3.4 BUJ Settings

Figure 7.10.3.4-1  BUJ Settings

Table 7.10.3.4-1  BUJ Setting Commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting item</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1]</td>
<td>BUJ ON/OFF</td>
<td>:SOURce-JITTer:BUJ:ENABle</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SOURce-JITTer:BUJ:ENABle?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SOURce-JITTer:BUJ:PRBS?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SOURce-JITTer:BUJ:AMPLitude?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SOURce-JITTer:BUJ:BITRate?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SOURce-JITTer:BUJ:LPFilter?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SOURce-JITTer:BUJ:PSAMplitude?</td>
</tr>
</tbody>
</table>
### :SOURce:JITTer:BUJ:ENABle <boolean>

**Parameter**

- `<boolean>` = `<BOOLEAN PROGRAM DATA>`
  - OFF or 0 = BUJ OFF
  - ON or 1 = BUJ ON

**Function**
Sets BUJ ON/OFF.

**Example**
To set BUJ setting to ON:

```
> :SOURce:JITTer:BUJ:ENABle ON
```

**Compatibility**
Incompatible with existing models.

### :SOURce:JITTer:BUJ:ENABle?

**Response**

- `<numeric>` = `<NR1 NUMERIC RESPONSE DATA>`
  - 0 = BUJ OFF
  - 1 = BUJ ON

**Function**
Queries BUJ ON/OFF status.

**Example**
```
> :SOURce:JITTer:BUJ:ENABle?
< 1
```

**Compatibility**
Incompatible with existing models.

### :SOURce:JITTer:BUJ:PRBS <numeric>

**Parameter**

- `<numeric>` = `<DECIMAL NUMERIC PROGRAM DATA>`
  - 7 = PRBS2^7–1 (n = 7)
  - 9 = PRBS2^9–1 (n = 9)
  - 11 = PRBS2^11–1 (n = 11)
  - 15 = PRBS2^15–1 (n = 15)
  - 23 = PRBS2^23–1 (n = 23)
  - 31 = PRBS2^31–1 (n = 31)

**Function**
Sets BUJ PRBS pattern.

**Example**
To set BUJ PRBS pattern to PRBS7:
```
> :SOURce:JITTer:BUJ:PRBS 7
```

**Compatibility**
Incompatible with existing models.

### :SOURce:JITTer:BUJ:PRBS?

**Response**

- `<numeric>` = `<NR1 NUMERIC RESPONSE DATA>`
  - 7, 9, 11, 15, 23, 31

**Function**
Queries BUJ PRBS pattern setting.

**Example**
```
> :SOURce:JITTer:BUJ:PRBS?
< 7
```

**Compatibility**
Incompatible with existing models.
Chapter 7  SCPI Commands

:SOURce:JITTer:BUJ:AMPLitude <numeric>

Parameter

\[ <\text{numeric}> = <\text{DECIMAL NUMERIC PROGRAM DATA}> \]

<table>
<thead>
<tr>
<th>Data Generator</th>
<th>Half-rate(MUX)</th>
<th>Full-rate(PPG), Full-rate(MUX)</th>
<th>Quarter-rate(MUX)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Setting Range (UIp-p)</td>
<td>Step (UI)</td>
<td>Setting Range (UIp-p)</td>
</tr>
<tr>
<td>&lt; 2.5 GHz</td>
<td>0 to 0.2f</td>
<td>0.004</td>
<td>0 to 0.2f</td>
</tr>
<tr>
<td>≥ 2.5 GHz</td>
<td>0 to 0.500</td>
<td>0.004</td>
<td>0 to 0.500</td>
</tr>
</tbody>
</table>

f: Clock Frequency
When F = 1 GHz, the upper limit is 0.2 UIp-p.
When F = 100 MHz, the upper limit is 0.02 UIp-p.

Note:
- The setting step varies according to the Data Generator setting.
- For details on setting ranges and steps, refer to 1.3.2 “Jitter Modulation Performance” in MU181500B Jitter Modulation Source Operation Manual.

Function
Sets BUJ modulation amount.

Example
To set BUJ modulation amount to 0.5 UI:

\[ > :\text{SOURce:JITTer:BUJ:AMPLitude 0.5} \]

Compatibility
Incompatible with existing models.

:SOURce:JITTer:BUJ:AMPLitude?

Response

\[ <\text{numeric}> = <\text{NR2 NUMERIC RESPONSE DATA}> \]
0 to 0.500 0 to 0.500 UIp-p

Function
Queries BUJ modulation amount.

Example

\[ > :\text{SOURce:JITTer:BUJ:AMPLitude?} \]
\[ < 0.500 \]

Compatibility
Incompatible with existing models.

:SOURce:JITTer:BUJ:BITRate <numeric>

Parameter

\[ <\text{numeric}> = <\text{DECIMAL NUMERIC PROGRAM DATA}> \]
9.800000 to 12.500000 9.800000 to 12.500000 Gbit/s/0.000001 Step
4.900000 to 6.250000  4.900000 to 6.250000 Gbit/s/0.000001 Step
0.100000 to 3.200000  0.100000 to 3.200000 Gbit/s/0.000001 Step

Function
Sets BUJ bit rate.

Example
To set BUJ bit rate to 1 Gbit/s:

\[ > :\text{SOURce:JITTer:BUJ:BITRate 1.0} \]

Compatibility
Incompatible with existing models.
7.10 Jitter Commands

:SOURce:JITTer:BUJ:BITRate?

Response
<numeric> = <NR2 NUMERIC RESPONSE DATA>
0.100000 to 12.500000

Function
Queries BUJ bit rate.

Example
> :SOURce:JITTer:BUJ:BITRate?
< 1.0000

Compatibility
Incompatible with existing models.

:SOURce:JITTer:BUJ:LPFilter <setting>

Parameter
<setting> = <CHARACTER PROGRAM DATA>
OFF   OFF
M_500  500 MHz low pass filter
M_300  300 MHz low pass filter
M_200  200 MHz low pass filter
M_100  100 MHz low pass filter
M_50   50 MHz low pass filter

Function
Selects low-pass filter used at BUJ.

Example
To select 100 MHz low-pass filter used at BUJ:
> :SOURce:JITTer:BUJ:LPFilter M_100

Compatibility
Incompatible with existing models.

:SOURce:JITTer:BUJ:LPFilter?

Response
<setting> = <CHARACTER RESPONSE DATA>
OFF, M_500, M_300, M_200, M_100, M_50

Function
Queries low-pass filter used BUJ.

Example
> :SOURce:JITTer:BUJ:LPFilter?
< M_100

Compatibility
Incompatible with existing models.

:SOURce:JITTer:BUJ:PSAMplitude?

Response
<numerical> = <NR2 NUMERIC RESPONSE DATA>
0 to x.xxx  0 to x.xxx ps p-p

Function
Queries BUJ’s modulation amount (ps p-p).

Example
> :SOURce:JITTer:BUJ:PSAMplitude?
< 0.500

Compatibility
Incompatible with existing models.
7.10.3.5 Ext Jitter Input Setting

Table 7.10.3.5-1 Ext. Jitter Setting Command

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting item</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ON/OFF</td>
<td>:SOURce:JITTer:EXTJitter:ENABle?</td>
</tr>
</tbody>
</table>

:SOURce:JITTer:EXTJitter:ENABle <boolean>

Parameter: 
<boolean> = <BOOLEAN PROGRAM DATA>
OFF or 0  Ext. Jitter Input OFF
ON or 1   Ext. Jitter Input ON

Function: Sets external jitter input ON/OFF.
Example: To set external data input to ON:
> :SOURce:JITTer:EXTJitter:ENABle ON

Compatibility: Incompatible with existing models.

:SOURce:JITTer:EXTJitter:ENABle?

Response: 
<numeric> = <NR1 NUMERIC RESPONSE DATA>
0  Ext. Jitter Input OFF
1  Ext. Jitter Input ON

Function: Queries external jitter input ON/OFF status.
Example: 
> :SOURce:JITTer:EXTJitter:ENABle?
< 1

Compatibility: Incompatible with existing models.
7.10.4 Output Setting Commands

Figure 7.10.4-1  Output Settings

Figure 7.10.4-2  Auxiliary Output Settings
### Table 7.10.4-1  Auxiliary Input/Output Setting Commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting item</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>:SOURce:OUTPut:DATA:SELect?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:OUTPut:AUX:SELect?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:OUTPut:AUX:REFClock?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:OUTPut:AUX:SUBRateclock?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:OUTPut:AUX:SUBRateclock:AMPLitude?</td>
</tr>
</tbody>
</table>

**:SOURce:OUTPut:DATA:SELect <select>**

Parameter  
<select> = <CHARACTER PROGRAM DATA>
- **FULLrate**: PPG module used when clock at Full rate.
- **HALFrate**: MUX Module use when clock at Half rate.
- **MUXFullrate**: MUX module using Full rate clock
- **QUARterrate**: MUX module using Quarter rate clock

Function  
Selects data generation module.

Example  
To use MUX as data generation module at Half rate:

> :SOURce:OUTPut:DATA:SELect HALFrate

Compatibility  
Incompatible with existing models.

**:SOURce:OUTPut:DATA:SELect?**

Response  
<numeric> = <CHARACTER RESPONSE DATA>
- **FULL**, **HALF**, **MUXF**, **QUAR**, **PPG32**, **MUX64**

Function  
Queries data generation module status.

Example  
> :SOURce:OUTPut:DATA:SELect?

< HALF

Compatibility  
Incompatible with existing models.
### 7.10 Jitter Commands

#### :SOURce:OUTPut:DATA:MONitor?

**Response** 

\(<\text{string}> = \text{<STRING RESPONSE DATA>}\>

<table>
<thead>
<tr>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;XX.XXXXX&quot;</td>
<td>0.0500000 to 66.0000000 0.0500000 to 66.0000000 Gbit/s</td>
</tr>
<tr>
<td>&quot;-----&quot;</td>
<td>When no data corresponds to the query (Out of range)</td>
</tr>
</tbody>
</table>

**Function** 
Queries BitRate of data output.

**Example**

\(>:\text{SOURce:OUTPut:DATA:MONitor}\?)

\(<"12.5000000"\>

**Compatibility** 
Incompatible with existing models.

#### :SOURce:OUTPut:DATA:JOVerload?

**Response** 

\(<\text{numeric}> = \text{<NR1 NUMERIC RESPONSE DATA>}\>

0  Not overloaded  
1  Overloaded

**Function** 
Queries whether jitter impressed on data output is Overload condition.

**Example**

\(>:\text{SOURce:OUTPut:DATA:JOVerload}\?)

\(<1\>

**Compatibility** 
Incompatible with existing models.

#### :OUTPut:AUX:SELect <select>

**Parameter** 

\(<\text{select}> = \text{<CHARACTER PROGRAM DATA>}\>

INTernal  Uses internal signal  
EXTernal  Uses external signal

**Function** 
Selects signal source used at auxiliary output.

**Example**

To use internal signal as auxiliary output:

\(>:\text{OUTPut:AUX:SELect INTernal}\)

**Compatibility** 
Incompatible with existing models.

#### :OUTPut:AUX:SELect?

**Response** 

\(<\text{numeric}> = \text{<CHARACTER RESPONSE DATA>}\>

INT, EXT

**Function** 
Queries signal source used as auxiliary output.

**Example**

\(>:\text{OUTPut:AUX:SELect}\?)

\(<\text{INT}\>

**Compatibility** 
Incompatible with existing models.


Chapter 7  SCPI Commands

:OUTPut:AUX:REFClock <numeric>

Parameter  

<nemonic> = <DECIMAL NUMERIC PROGRAM DATA>

1 1/1 Clock output
2 1/2 Clock output
4 1/4 Clock output

Function  Sets reference clock output.
Example  To set reference clock output to 1/2:
> :OUTPut:AUX:REFClock 2

Compatibility  Incompatible with existing models.

:OUTPut:AUX:REFClock?

Response  

<nemonic> = <NR1 NUMERIC RESPONSE DATA>

1, 2, 4,

Function  Queries reference clock output status.
Example  > :OUTPut:AUX:REFClock?
< 2

Compatibility  Incompatible with existing models.

:OUTPut:AUX:REFClock:MONitor?

Response  

<string> = <STRING RESPONSE DATA>

<table>
<thead>
<tr>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;XXXXXX&quot;</td>
<td>12500 to 16250000 12,500 to 16,250,000 kHz</td>
</tr>
<tr>
<td>&quot;-----&quot;</td>
<td>When no data corresponds to the query</td>
</tr>
</tbody>
</table>

Function  Queries reference clock output frequency.
Example  > :OUTPut:AUX:REFClock:MONitor?
< "100000"

Compatibility  Incompatible with existing models.

:OUTPut:AUX:SUBRateclock <numeric>

Parameter  

<nemonic> = <DECIMAL NUMERIC PROGRAM DATA>

8 to 256 1/N (N = 8 to 256)

Function  Sets sub-rate clock division rate.
Example  To set sub-rate clock division rate to 1/8:
> :OUTPut:AUX:SUBRateclock 8

Compatibility  Incompatible with existing models.
7.10 Jitter Commands

:OUTPut:AUX:SUBRateclock?

Response  
<numeric> = <NR1 NUMERIC RESPONSE DATA>
8 to 256  
1/N (N = 8 to 256)

Function  Queries sub-rate clock division rate.

Example  
> :OUTPut:AUX:SUBRateclock?
< 8

Compatibility  Incompatible with existing models.

:OUTPut:AUX:SUBRateclock:AMPLitude <numeric>

Parameter  
<numeric> = <DECIMAL NUMERIC PROGRAM DATA>
0.10 to 0.70  
0.10 to 0.70 V/0.01 V Step

Function  Sets sub-rate clock output amplitude.

Example  
To set sub-rate clock output amplitude to 0.5 V:
> :OUTPut:AUX:SUBRateclock:AMPLitude 0.5

Compatibility  Incompatible with existing models.

:OUTPut:AUX:SUBRateclock:AMPLitude?

Response  
<numeric> = <NR2 NUMERIC RESPONSE DATA>
0.10 to 0.70

Function  Queries sub-rate clock amplitude status.

Example  
> :OUTPut:AUX:SUBRateclock:AMPLitude?
< 0.50

Compatibility  Incompatible with existing models.

:OUTPut:AUX:SUBRateclock:MONitor?

Response  
<string> = <STRING RESPONSE DATA>

<table>
<thead>
<tr>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;XXXXXXX&quot;</td>
<td>195 to 4062500  195 to 4,062,500 kHz</td>
</tr>
<tr>
<td>&quot;-----&quot;</td>
<td>When no data corresponds to the query</td>
</tr>
</tbody>
</table>

Function  Queries sub-rate clock output frequency.

Example  
> :OUTPut:AUX:SUBRateclock:MONitor?
< "100000"

Compatibility  Incompatible with existing models.
## :OUTPut:AUX:JCONdition?

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;numeric&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>No jitter</td>
</tr>
<tr>
<td>1</td>
<td>Jitter added</td>
</tr>
</tbody>
</table>

**Function**
Queries status of jitter addition at AUX output (Reference Clock and Sub-rate Clock).

**Example**

`> :OUTPut:AUX:JCONdition?`

`< 1`

**Compatibility**
Incompatible with existing models.
7.10.5 File Menu Setting Commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting item</th>
<th>Command</th>
</tr>
</thead>
</table>

**:SYSTem:MMEMory:RECall <file_name>,<module>,<data_type>**

Parameter

- `<file_name>` = `<STRING PROGRAM DATA>
  
  "<drv>\[<dir>]<file>"`

- `<drv>` = C, D, E, F
- `<dir>` = `<dir1><dir2>...` (Omitted when root directory)
- `<file>` = File name
- `<module>` = `<STRING PROGRAM DATA>`
  
  "<unit>:<slot>:<port> <module>"

- `<unit>` = 1 to 4
- `<slot>` = 1 to 6
- `<port>` = 1
- `<module>` = MU181500B

**<data_type>** = `<CHARACTER PROGRAM DATA>`

**Function**

Loads MP181500B setting data.

**Example**

> :SYSTem:MMEMory:RECall "C:\Test\example","1:1:1 MU181500B",JMS

**Compatibility**

Incompatible with existing models.
Chapter 7  SCPI Commands

:SYStem:MMEMory:STORe
<file_name>,<module>,<data_type>,<file_type>

Parameter

<file_name> = <STRING PROGRAM DATA>
"<drv>:\[<dir>]<file>"
<drv> = C, D, E, F
<dir>=<dir1><dir2>...(Omitted when root directory)
<file> = File name
<module> = <STRING PROGRAM DATA>
"<unit>:<slot>:<port> <module>"
<unit> = 1 to 4
<slot> = 1 to 6
<port> = 1
<module> = MU181500B
<data_type> = <CHARACTER RESPONSE DATA>
JMS  Jitter Modulation Source Setup
<file_type> = <CHARACTER PROGRAM DATA>
BIN  Binary file

Function
Saves MP181500B setting data.

Note:
Note that if the name of the saved file is changed, it is no longer possible to read the setting data.

Example
To save MU181500B setting data, specifying save destination, file name and file format:
> :SYStem:MMEMory:STORe "C:\Test\example","1:1:1
MU181500B",JMS,BIN

Compatibility
Incompatible with existing models.
7.11 28G/32G bit/s PPG Commands

This section describes the setting and query commands for the MU183020A 28G/32G bit/s PPG, MU183021A 28G/32G bit/s 4ch PPG (hereafter, MU183020A/21A).

Before executing a setting/query command for the MU183020A/21A, specify the slot number of the module to be remotely controlled, by using the :MODule:ID command. Refer to Section 7.1 “Common Commands” and 9.1 “Basic Operation of Device Messages” for how to specify a slot number with the :MODule:ID command.

7.11.1 Common Commands

This section describes the commands related to common settings and functions of the control software.

7.11.1.1 Commands for common settings

<table>
<thead>
<tr>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of the unit to be operated</td>
<td>:UENTry:ID</td>
</tr>
<tr>
<td></td>
<td>:UENTry:ID?</td>
</tr>
<tr>
<td>Number of the module to be operated (slot position)</td>
<td>:MODule:ID</td>
</tr>
<tr>
<td></td>
<td>:MODule:ID?</td>
</tr>
<tr>
<td>Number of the port to be operated</td>
<td>:PORT:ID</td>
</tr>
<tr>
<td></td>
<td>:PORT:ID?</td>
</tr>
<tr>
<td>Number of the data interface to be operated</td>
<td>:INTerface:ID</td>
</tr>
<tr>
<td></td>
<td>:INTerface:ID?</td>
</tr>
<tr>
<td>Query for error message</td>
<td>:SYSTem:ERRor?</td>
</tr>
<tr>
<td>Query for SCPI version</td>
<td>:SYSTem:VERSion?</td>
</tr>
<tr>
<td>Query for software status</td>
<td>:SYSTem:CONDition?</td>
</tr>
<tr>
<td>Query for hardware system configuration</td>
<td>:SYSTem:ORGanization:HARDware?</td>
</tr>
<tr>
<td>Query for system error</td>
<td>:SYSTem:INFormation:ERRor?</td>
</tr>
<tr>
<td>Terminator type</td>
<td>:SYSTem:TERMination</td>
</tr>
<tr>
<td>Query for model name of mainframe and module</td>
<td>:SYSTem:CONDition:UNITs?</td>
</tr>
<tr>
<td>Query for mainframe information</td>
<td>:SYSTem:UNIT?</td>
</tr>
<tr>
<td>Query for module information</td>
<td>:SYSTem:MODule?</td>
</tr>
<tr>
<td>Module screen display</td>
<td>:DISPlay:ACTive</td>
</tr>
</tbody>
</table>
Chapter 7  SCPI Commands

:UENTry:ID <unit_number>
Parameter  <unit_number> = <DECIMAL NUMERIC PROGRAM DATA>
            1 to 4  Unit number
Function  Sets the number of the unit to be operated.
Example  To set the number of the unit to be operated to 2:
            > :UENTry:ID 2
Compatibility  Incompatible with existing models

:UENTry:ID?
Response  <unit_number> = <NR1 NUMERIC RESPONSE DATA>
            1 to 4
Function  Queries the number of the unit being operated.
Example  > :UENTry:ID?
            < 2
Compatibility  Incompatible with existing models

:MODule:ID <module_number>
Parameter  <module_number> = <DECIMAL NUMERIC PROGRAM DATA>
            1 to 6  Module number
Function  Sets the number of the module to be operated (slot position).
Example  To set the number of the module to be operated (slot position) to 6:
            > :MODule:ID 6
Compatibility  Incompatible with existing models

:MODule:ID?
Response  <module_number> = <NR1 NUMERIC RESPONSE DATA>
            1 to 6
Function  Queries the number of the module being operated (slot position).
Example  > :MODule:ID?
            < 6
Compatibility  Incompatible with existing models
7.11 28G/32G bit/s PPG Commands

:**PORT:ID <port_number>**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;port_number&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identifier for the port to be operated.</td>
<td></td>
</tr>
<tr>
<td>Indicates the physical position number of the port assigned to a module.</td>
<td></td>
</tr>
<tr>
<td>Function</td>
<td>Sets the number of the port to be operated (port position).</td>
</tr>
<tr>
<td>Example</td>
<td>To set the number of the port to be operated (port position) to 1:</td>
</tr>
<tr>
<td></td>
<td>&gt; :PORT:ID 1</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Incompatible with existing models</td>
</tr>
</tbody>
</table>

:**PORT:ID?**

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;port_number&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identifier for the port to be operated.</td>
<td></td>
</tr>
<tr>
<td>Indicates the physical position number of the port assigned to a module.</td>
<td></td>
</tr>
<tr>
<td>Function</td>
<td>Queries the number of the port being operated (port position).</td>
</tr>
<tr>
<td>Example</td>
<td>&gt; :PORT:ID?</td>
</tr>
<tr>
<td></td>
<td>&lt; 1</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Incompatible with existing models</td>
</tr>
</tbody>
</table>

:**INTerface:ID <interface_number>**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;interface_number&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 to 4 Data interface number</td>
<td></td>
</tr>
<tr>
<td>Function</td>
<td>Sets the number of data interface to be operated (Data 1 to 4).</td>
</tr>
<tr>
<td>Example</td>
<td>To set the number of data interface to be operated to 4:</td>
</tr>
<tr>
<td></td>
<td>&gt; :INTerface:ID 4</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Incompatible with existing models</td>
</tr>
</tbody>
</table>

:**INTerface:ID?**

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;interface_number&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 to 4</td>
<td>Queries the number of the data interface being operated (Data 1 to 4).</td>
</tr>
<tr>
<td>Function</td>
<td></td>
</tr>
<tr>
<td>Example</td>
<td>&gt; :INTerface:ID?</td>
</tr>
<tr>
<td></td>
<td>&lt; 4</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Incompatible with existing models</td>
</tr>
</tbody>
</table>
**Chapter 7  SCPI Commands**

### :SYSTem:ERRor?

**Response**

<error/event_number>,"<error/event_description>"

<error/event_number> = <NR1 NUMERIC RESPONSE DATA>

-32768 to 32767

"0" indicates that no errors and events have occurred. Other values return a general error reserved by SCPI or a device-unique error.

<error/event_description> = <STRING RESPONSE DATA>

This is an error message corresponding to <error/event_number>. The maximum character-string length is 255 characters.

**Function**

Queries the error message in the error/event queue.

For the details about the error message, refer to Chapter 8 and 9 in the MX180000A Remote Control Operation Manual.

**Example**

> :SYSTem:ERRor?

< 0, "No error"

**Compatibility**

Compatible with the MP1632C and MP1776A.

### :SYSTem:VERSion?

**Response**

<version> = <NR2 NUMERIC RESPONSE DATA>

YYYY.V

YYYY :year

V :revision number

**Function**

Queries the SCPI version to which the MP1800A/MT1810A conforms.

**Example**

> :SYSTem:VERSion?

< 1999.0

**Compatibility**

Compatible with the MP1632C and MP1776A.
:SYSTem:CONDition?

Response

"<mainframe>,<slot1>,...,<slot64>"

<mainframe> = <serial>,<mver>,<hver>,<opt1>,<sbver>,<saver>,<opt2>
<serial> = <STRING RESPONSE DATA>
XXXXXXXXXXXX 0000000000 to 9999999999
Mainframe serial number

Note:
Alphabetic characters may be included.

<mver> = <STRING RESPONSE DATA>
XXXX.XX.XX 1.00.00 to 9999.99.99
Main application software version

<hver> = <STRING RESPONSE DATA>
XXXX.XX.XX 1.00.00 to 9999.99.99
Mainframe hardware version

<opt1> = <STRING RESPONSE DATA>
OPTXXX Option number (MP1800A/MT1810A)
See Table 7.11.1.1-2 “Option Character Correspondence Table”.

Note:
Outputs the numbers for all installed options.
NONE is output if no option is installed.

<sbver> = <STRING RESPONSE DATA>
XXXX.XX.XX 1.00.00 to 9999.99.99
Sub application software version (Boot part)

<saver> = <STRING RESPONSE DATA>
XXXX.XX.XX 1.00.00 to 9999.99.99
Sub application software version (Application part)

<opt2> = <STRING RESPONSE DATA>

<slot x> = <module>,<serial>,<fpga1>[,<fpga2>],<boot>,
<application>,<opt>
x indicates a slot number. The slot number varies depending on the unit number as follows.
Unit 1: 1 to 16 For the MP1800A, numbers from 1 to 6 correspond to actual slots. For the MT1810A, numbers from 1 to 4 correspond to actual slots.
Unit 2: 17 to 32 For the MT1810A, numbers from 17 to 20 correspond to actual slots.
Unit 3: 33 to 48 For the MT1810A, numbers from 33 to 36 correspond to actual slots.
Unit 4: 49 to 64 For the MT1810A, numbers from 49 to 52 correspond to actual slots.

<module> = <STRING RESPONSE DATA>
Chapter 7  SCPI Commands

Module name (e.g.: MU183020A)
See Table 7.11.1.1-2 “Option Character Correspondence Table”.

Note:
NONE is output if no module is installed. For a module that uses two slots, only the slot with the greater number is valid.

<serial> = <STRING RESPONSE DATA>
XXXXXXXXXX 0000000000 to 9999999999
Serial number

Note:
"-------------" is output if no module is installed. For a module that uses two slots, only the slot with the greater number is valid.

<fpga1>[,<fpga2>,.....] = <STRING RESPONSE DATA>
XXXX.XX.XX 1.00.00 to 9999.99.99
FPGA version

<boot> = <STRING RESPONSE DATA>
XXXX.XX.XX 1.00.00 to 9999.99.99
Logic Boot version

Note:
"-------------" is output if Logic Boot is not installed. For a module that uses two slots, only the slot with the greater number is valid.

<application> = <STRING RESPONSE DATA>
XXXX.XX.XX 1.00.00 to 9999.99.99
Logic Application version

Note:
"-------------" is output if Logic Application is not installed. For a module that uses two slots, only the slot with the greater number is valid.

<opt> = <STRING RESPONSE DATA>
XXXXXXX/XXXXX Option number
OPTXXX: For MP1800A/MT1810A

Note:
Outputs the numbers for all installed options. NONE is output if no option is installed. For a module that uses two slots, only the slot with the greater number is valid. The option names of the MU181020B-003 and MU181040B-003 are not output.

Function
Queries the software status of the MP1800A/MT1810A.
7.11 28G/32G bit/s PPG Commands

Example

```
> :SYSTem:CONDition?

6201234567,1.00.00,1.00.20,OPT302,1.00.00,1.00.00,OPT12,
OPT14,MU181000A,6201234568,1.00.00,1.00.00,1.00.00,OPT10
1,MU181020A,6201234569,1.00.00,1.00.00,1.00.00,OPT001,OP
T220,MU181040A,6201234571,1.00.00,1.00.00,1.00.00,OPT002
,OPT220
```

Compatibility

Incompatible with existing models
Table 7.11.1-2  Option Character Correspondence Table

<table>
<thead>
<tr>
<th>Model/Name</th>
<th>Option Number</th>
<th>Option Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>MP1800A Signal Quality Analyzer</td>
<td>OPT014</td>
<td>2-Slot PPG/ED</td>
</tr>
<tr>
<td></td>
<td>OPT015</td>
<td>4-Slot PPG/ED</td>
</tr>
<tr>
<td></td>
<td>OPT016</td>
<td>6-Slot PPG/ED</td>
</tr>
<tr>
<td></td>
<td>OPT032</td>
<td>32Gbit/s PPG/ED</td>
</tr>
<tr>
<td>MT1810A 4 Slot Chassis</td>
<td>OPT014</td>
<td>2-Slot PPG/ED</td>
</tr>
<tr>
<td></td>
<td>OPT015</td>
<td>4-Slot PPG/ED</td>
</tr>
<tr>
<td></td>
<td>OPT032</td>
<td>32Gbit/s PPG/ED</td>
</tr>
<tr>
<td>MX180000A Signal Quality Analyzer Control Software</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>MU181000A 12.5GHz Synthesizer</td>
<td>OPTx01</td>
<td>Jitter Modulation</td>
</tr>
<tr>
<td>MU181000B 12.5GHz 4port Synthesizer</td>
<td>OPTx01</td>
<td>Jitter Modulation</td>
</tr>
<tr>
<td>MU181020A 12.5Gbit/s PPG</td>
<td>OPT001</td>
<td>9.8 to 12.5Gbit/s</td>
</tr>
<tr>
<td></td>
<td>OPT002</td>
<td>0.1 to 12.5Gbit/s</td>
</tr>
<tr>
<td></td>
<td>OPTx10</td>
<td>Variable Data Output (0.05 to 0.8 Vp-p)</td>
</tr>
<tr>
<td></td>
<td>OPTx11</td>
<td>Variable Data Output (0.25 to 2.5 Vp-p)</td>
</tr>
<tr>
<td></td>
<td>OPTx12</td>
<td>High Performance Data Output (0.05 to 2.0 Vp-p)</td>
</tr>
<tr>
<td></td>
<td>OPTx13</td>
<td>Variable Data Output (0.5 to 3.5 Vp-p)</td>
</tr>
<tr>
<td></td>
<td>OPTx21</td>
<td>Differential Clock Output (0.1 to 2.0 Vp-p)</td>
</tr>
<tr>
<td></td>
<td>OPTx30</td>
<td>Variable Data Delay</td>
</tr>
<tr>
<td>MU181040A 12.5Gbit/s ED</td>
<td>OPT001</td>
<td>9.8 to 12.5Gbit/s</td>
</tr>
<tr>
<td></td>
<td>OPT002</td>
<td>0.1 to 12.5Gbit/s</td>
</tr>
<tr>
<td></td>
<td>OPTx20</td>
<td>Clock Recovery</td>
</tr>
<tr>
<td></td>
<td>OPTx30</td>
<td>Variable Clock Delay</td>
</tr>
<tr>
<td>MU181600A Optical Transceiver (XFP)</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>MU181601A Optical Transceiver (SFP)</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>MU181620A Stressed Eye Transmitter</td>
<td>OPTx01</td>
<td>1310nm Reference</td>
</tr>
<tr>
<td></td>
<td>OPTx02</td>
<td>1550nm Reference</td>
</tr>
<tr>
<td></td>
<td>OPTx03</td>
<td>1310nm/1550nm Reference</td>
</tr>
<tr>
<td></td>
<td>OPTx11</td>
<td>1310nm Stressed Eye</td>
</tr>
<tr>
<td></td>
<td>OPTx12</td>
<td>1550nm Stressed Eye</td>
</tr>
<tr>
<td></td>
<td>OPTx13</td>
<td>1310nm/1550nm Stressed Eye</td>
</tr>
<tr>
<td>MU181640A Optical Receiver</td>
<td>OPTx04</td>
<td>Band Width 8.5GHz</td>
</tr>
<tr>
<td>MU181800A 12.5GHz Clock Distributor</td>
<td>—</td>
<td>—</td>
</tr>
</tbody>
</table>
### Table 7.11.1-2 Option Character Correspondence Table (Cont’d)

<table>
<thead>
<tr>
<th>Model/Name</th>
<th>Option Number</th>
<th>Option Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>MU181020B 14Gbit/s PPG</td>
<td>OPT002</td>
<td>0.1 to 14.Gbit/s</td>
</tr>
<tr>
<td></td>
<td>OPTx11</td>
<td>Variable Data Output (0.25 to 2.5 Vp-p)</td>
</tr>
<tr>
<td></td>
<td>OPTx12</td>
<td>High Performance Data Output (0.05 to 2.0 Vp-p)</td>
</tr>
<tr>
<td></td>
<td>OPTx13</td>
<td>Variable Data Output (0.5 to 3.5 Vp-p)</td>
</tr>
<tr>
<td></td>
<td>OPTx21</td>
<td>Differential Clock Output (0.1 to 2.0 Vp-p)</td>
</tr>
<tr>
<td></td>
<td>OPTx30</td>
<td>Variable Data Delay</td>
</tr>
<tr>
<td>MU181040B 14Gbit/s ED</td>
<td>OPT002</td>
<td>0.1 to 14.Gbit/s</td>
</tr>
<tr>
<td></td>
<td>OPTx20</td>
<td>Clock Recovery</td>
</tr>
<tr>
<td></td>
<td>OPTx30</td>
<td>Variable Clock Delay</td>
</tr>
<tr>
<td>MU182020A 25Gbit/s 1ch MUX MU182021A 25Gbit/s 2ch MUX</td>
<td>OPTx01</td>
<td>28Gbit/s Extension</td>
</tr>
<tr>
<td></td>
<td>OPTx02</td>
<td>Clock Input Band Switch</td>
</tr>
<tr>
<td></td>
<td>OPTx10</td>
<td>Variable Data Output (0.25 to 1.75 Vp-p)</td>
</tr>
<tr>
<td></td>
<td>OPTx11</td>
<td>Variable Data Output (0.5 to 2.5 Vp-p)</td>
</tr>
<tr>
<td></td>
<td>OPTx12</td>
<td>Variable Data Output (0.5 to 2.0 Vp-p)</td>
</tr>
<tr>
<td></td>
<td>OPTx13</td>
<td>Variable Data Output (0.5 to 3.5 Vp-p)</td>
</tr>
<tr>
<td></td>
<td>OPTx21</td>
<td>Clock output (0.5 to 2.0 Vp-p)</td>
</tr>
<tr>
<td></td>
<td>OPTx30</td>
<td>25Gbit/s Variable Data Delay</td>
</tr>
<tr>
<td></td>
<td>OPTx31</td>
<td>28Gbit/s Variable Data Delay</td>
</tr>
<tr>
<td>MU182040A 25Gbit/s 1ch DEMUX MU182041A 25Gbit/s 2ch DEMUX</td>
<td>OPTx01</td>
<td>28Gbit/s Extension</td>
</tr>
<tr>
<td></td>
<td>OPTx02</td>
<td>Clock Input Band Switch</td>
</tr>
<tr>
<td></td>
<td>OPTx30</td>
<td>25GHz Variable Clock Delay</td>
</tr>
<tr>
<td></td>
<td>OPTx31</td>
<td>28GHz Variable Clock Delay</td>
</tr>
<tr>
<td>MU183020A 28G/32G bit/s PPG</td>
<td>OPTx01</td>
<td>32G bit/s Extension</td>
</tr>
<tr>
<td></td>
<td>OPTx12</td>
<td>1ch 2V Data Output</td>
</tr>
<tr>
<td></td>
<td>OPTx13</td>
<td>1ch 3.5V Data Output</td>
</tr>
<tr>
<td></td>
<td>OPTx30</td>
<td>1ch Data Delay</td>
</tr>
<tr>
<td></td>
<td>OPTx22</td>
<td>2ch 2V Data Output</td>
</tr>
<tr>
<td></td>
<td>OPTx23</td>
<td>2ch 3.5V Data Output</td>
</tr>
<tr>
<td></td>
<td>OPTx31</td>
<td>2ch Data Delay</td>
</tr>
<tr>
<td>MU183021A 28G/32G bit/s 4ch PPG</td>
<td>OPTx01</td>
<td>32G bit/s Extension</td>
</tr>
<tr>
<td></td>
<td>OPTx12</td>
<td>4ch 2V Data Output</td>
</tr>
<tr>
<td></td>
<td>OPTx13</td>
<td>4ch 3.5V Data Output</td>
</tr>
<tr>
<td></td>
<td>OPTx30</td>
<td>4ch Data Delay</td>
</tr>
<tr>
<td>MU183040A 28G/32G bit/s ED</td>
<td>OPTx01</td>
<td>32G bit/s Extension</td>
</tr>
<tr>
<td></td>
<td>OPTx10</td>
<td>1ch ED</td>
</tr>
<tr>
<td></td>
<td>OPTx20</td>
<td>2ch ED</td>
</tr>
<tr>
<td>MU183041A 28G/32G bit/s 4ch ED</td>
<td>OPTx01</td>
<td>32G bit/s Extension</td>
</tr>
</tbody>
</table>

**Note:**
The option names of the MU181020B-003 and MU181040B-003 are not output.
:SYSTem:ORGanization:HARDware?

Response

<slot1>,...,<slot64>

<slotx> =

<module>,<serial>,<fpga1>[,<fpga2>],<boot>,<application>,<opt>

x indicates a slot number.

The slot number varies depending on the unit number as follows.

Unit 1: 1 to 16 For the MP1800A, numbers from 1 to 6 correspond to actual slots. For the MT1810A, numbers from 1 to 4 correspond to actual slots.

Unit 2: 17 to 32 For the MT1810A, numbers from 17 to 20 correspond to actual slots.

Unit 3: 33 to 48 For the MT1810A, numbers from 33 to 36 correspond to actual slots.

Unit 4: 49 to 64 For the MT1810A, numbers from 49 to 52 correspond to actual slots.

<module> = <STRING RESPONSE DATA>

XXXXXXXXXX Module name (e.g.: MU181020A)

See Table 7.11.1.1-2 “Option Character Correspondence Table”.

Note:

NONE is output if no module is installed. For a module that uses two slots, only the slot with the greater number is valid.

<serial> = <STRING RESPONSE DATA>

XXXXXXXXXX 0000000000 to 9999999999 Serial number

Note:

"-------------" is output if no module is installed. For a module that uses two slots, only the slot with the greater number is valid.

<fpga1>[,<fpga2>,.....] = <STRING RESPONSE DATA>

XXXX.XX.XX 1.00.00 to 9999.99.99 FPGA version

<boot> = <STRING RESPONSE DATA>

XXXX.XX.XX 1.00.00 to 9999.99.99 Logic Boot version

Note:

"-------------" is output if Logic Boot is not installed. For a module that uses two slots, only the slot with the greater number is valid.

<application> = <STRING RESPONSE DATA>

XXXX.XX.XX 1.00.00 to 9999.99.99 Logic Application version
Note:

"-------------" is output if Logic Application is not installed.
For a module that uses two slots, only the slot with the greater number is valid.

<opt> = <STRING RESPONSE DATA>

XXXXXXX/XXXXX Option number OPTXXX:
For MP1800A/MT1810A, outputs the numbers for all installed options.

Note:

NONE is output if no option is installed. For a module that uses two slots, only the slot with the greater number is valid.
The option names of the MU181020B-003 and MU181040B-003 are not output.

Function
Queries the hardware system configuration of the MP1800A.

Example
> :SYSTem:ORGanization:HARDware?
< MU181000A,6201234568,1.00.00,1.00.00,1.00.00,OPT101,
MU181020A,6201234569,1.00.00,1.00.00,1.00.00,OPT001,OPT2
20,MU181040A,6201234571,1.00.00,1.00.00,1.00.00,OPT002,O
PT220

Compatibility
Partially compatible with the MP1632C and MP1776A.

:SYSTem:INFormation:ERRor? <unit>

Parameter
<unit> = <DECIMAL NUMERIC PROGRAM DATA>
1 to 4 1 to 4, 1 step

Response
<numeric> = <NR1 NUMERIC RESPONSE DATA>
0 NONE
1 PLL Unlock
2 Temperature
3 Fan

All the system errors that have currently occurred are displayed, delimited with a comma (,).

Function
Queries the System Error contents.

Example
> :SYSTem:INFormation:ERRor? 3
< 1,2,3 (when a system error has occurred for PLL Unlock, Temperature, or Fan)
< 0 (when no system error has occurred)

Compatibility
Incompatible with existing models
Chapter 7  SCPI Commands

:SYSTem:TERMination <numeric>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;numeric&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>LF + EOI</td>
</tr>
<tr>
<td>1</td>
<td>CR + LF + EOI</td>
</tr>
</tbody>
</table>

**Function**
Sets the terminator type of the response data.

**Example**
To set the terminator type to LF + EOI:

```
> :SYSTem:TERMination 0
```

**Compatibility**
Compatible with the MP1632C.

:SYSTem:TERMination?

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;numeric&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>LF + EOI</td>
</tr>
<tr>
<td>1</td>
<td>CR + LF + EOI</td>
</tr>
</tbody>
</table>

**Function**
Queries the terminator type of the response data.

**Example**

```
> :SYSTem:TERMination?
< 0
```

**Compatibility**
Compatible with the MP1632C.
**:SYSTem:CONDition:UNITs?**

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;unit1&gt;,...,&lt;unit4&gt;,&lt;slot1&gt;,...,&lt;slot6&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>=&quot;&lt;mainframe1&gt;,...,&lt;mainframe4&gt;,&lt;module1&gt;,...,&lt;module64&gt;&quot;&lt;mainframe1&gt; to &lt;mainframe4&gt; = &lt;STRING RESPONSE DATA&gt;</td>
</tr>
<tr>
<td></td>
<td>XXXXXXXXXX Mainframe name (e.g.: MP1800A)</td>
</tr>
<tr>
<td></td>
<td>See Table 7.11.1-2 “Option Character Correspondence Table”.</td>
</tr>
</tbody>
</table>

**Note:**

NONE is output for mainframe2 to 4, if no mainframe is connected.

<module1> to <module64> = <STRING RESPONSE DATA>

XXXxxxxxxx Module name (e.g.: MU181020A)

See Table 7.11.1-2 “Option Character Correspondence Table”.

**Note:**

NONE is output if no module is installed. For a module that uses two slots, only the slot with the greater number is valid.

**Function**

Queries the model name of the mainframe and module.

**Example**

> :SYSTem:CONDition:UNITs?  
< MT1810A,MT1810A,MT1810A,NONE,MU181000A,NONE,MU181020A, 
MU181040A,NONE,...,MU181600A,MU181601A,MU181620A,NONE, 
MU181640A,NONE,...,NONE

**Example**

To query the model name of the mainframe and module for the MP1800A (1-box type):

> :SYSTem:CONDition:UNITs?  
< "MP1800A,NONE,NONE,NONE,MU181000A,NONE,MU181020A, 
MU181040A,NONE,MU181600A,NONE,...,NONE"

**Compatibility**

Incompatible with existing models
## Chapter 7  SCPI Commands

### :SYStem:UNIT? <numeric>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;numeric&gt;</td>
<td>Mainframe number</td>
</tr>
<tr>
<td>1 to 4</td>
<td>“1” for the MP1800A, “1 to 4” for the MT1810A.</td>
</tr>
</tbody>
</table>

### Response

<table>
<thead>
<tr>
<th>&lt;mainframe&gt;</th>
<th>Mainframe name (e.g., : MP1800A/MT1810A)</th>
</tr>
</thead>
<tbody>
<tr>
<td>=</td>
<td>See Table 7.11.1.1-2 “Option Character Correspondence Table”.</td>
</tr>
</tbody>
</table>

#### Note:

NONE is output if no module is installed. For a unit that uses two slots, only the slot with the lower number is valid.

<table>
<thead>
<tr>
<th>&lt;serial&gt;</th>
<th>Mainframe serial number</th>
</tr>
</thead>
<tbody>
<tr>
<td>=</td>
<td>0000000000 to 9999999999</td>
</tr>
</tbody>
</table>

#### Note:

Alphabetic characters may be included.

<table>
<thead>
<tr>
<th>&lt;mver&gt;</th>
<th>Main application software version</th>
</tr>
</thead>
<tbody>
<tr>
<td>=</td>
<td>1.00.00 to 9999.99.99</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>&lt;hver&gt;</th>
<th>Mainframe hardware version</th>
</tr>
</thead>
<tbody>
<tr>
<td>=</td>
<td>1.00.00 to 9999.99.99</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>&lt;opt1&gt;</th>
<th>Option number (MP1800A/MT1810A)</th>
</tr>
</thead>
<tbody>
<tr>
<td>=</td>
<td>See Table 7.11.1.1-2 “Option Character Correspondence Table”.</td>
</tr>
</tbody>
</table>

#### Note:

Outputs the numbers for all installed options. NONE is output if no option is installed.

<table>
<thead>
<tr>
<th>&lt;sbver&gt;</th>
<th>Sub application software version (Boot part)</th>
</tr>
</thead>
<tbody>
<tr>
<td>=</td>
<td>1.00.00 to 9999.99.99</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>&lt;saver&gt;</th>
<th>Sub application software version (Application part)</th>
</tr>
</thead>
<tbody>
<tr>
<td>=</td>
<td>1.00.00 to 9999.99.99</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>&lt;opt2&gt;</th>
<th>Sub application software version (Application part)</th>
</tr>
</thead>
<tbody>
<tr>
<td>=</td>
<td>1.00.00 to 9999.99.99</td>
</tr>
</tbody>
</table>

### Function

Queries the mainframe information including model and serial number.

### Example

```
> :SYSTem:UNIT? 1
<
"MP1800A,6201234568,1.00.00,1.00.00,OPT301,1.00.00,1.00.
```
### 7.11 28G/32G bit/s PPG Commands

<table>
<thead>
<tr>
<th>Compatibility</th>
<th>Incompatible with existing models</th>
</tr>
</thead>
<tbody>
<tr>
<td>00&quot;</td>
<td></td>
</tr>
</tbody>
</table>
Chapter 7  SCPI Commands

**:SYSTem:MODule? <numeric>**

Parameter

<numeric> = <NR1 NUMERIC PROGRAM DATA>

1 to 6

1 to 6 when using the MP1800A, 1 to 4 when using the MT1810A

Response

<slot> = <module>,<serial>,<fpga1>,<fpga2>,<boot>,<application>,<opt>

<module> = <STRING RESPONSE DATA>

XXXXXXXXXX Module name (e.g.,: MU181020A)

See Table 7.11.1-2 “Option Character Correspondence Table”.

Note:

NONE is output if no module is installed. For a module that uses two slots, only the slot with the greater number is valid.

<serial> = <STRING RESPONSE DATA>

XXXXXXXXXX 0000000000 to 9999999999

Serial number

Note:

"---------" is output if no module is installed. For a module that uses two slots, only the slot with the greater number is valid.

<fpga1>,<fpga2> = <STRING RESPONSE DATA>

XXXX.XX.XX 1.00.00 to 9999.99.99

FPGA version

<boot> = <STRING RESPONSE DATA>

XXXX.XX.XX 1.00.00 to 9999.99.99

Logic Boot version

Note:

"-----------" is output if Logic Boot is not installed. For a module that uses two slots, only the slot with the greater number is valid.

<application> = <STRING RESPONSE DATA>

XXXX.XX.XX 1.00.00 to 9999.99.99

Logic Application version

Note:

"-----------" is output if Logic Application is not installed. For a module that uses two slots, only the slot with the greater number is valid.

<opt> = <STRING RESPONSE DATA>

XXXXXXXXXXXX Option number

OPTXXX: For MP1800A/MT1810A

Note:

Outputs the numbers for all installed options. NONE is output if no option is installed. For a module that uses two slots, only the slot with the greater number is valid. The option names of the MU181020B-003 and MU181040B-003 are not output.

Function

Queries the module information on the specified slot.

Example

To query the module information on Slot 3:
7.11 28G/32G bit/s PPG Commands

> :SYSTem:MODuLe? 3
< "MU181020A, 6201234568, 1.00.00, --------, 1.00.00, 1.00.00, OPT002, OPT211, PT220,"

Compatibility
Incompatible with existing models

`DISPLAY:ACTive <unit>,<slot>[,<tab>]`

Parameter
- `<unit>` = `<DECIMAL NUMERIC PROGRAM DATA>`
  - 1 to 4 Mainframe No. 1 to 4
    - When installing the MT1810A, the mainframe No. is specified.
    - When using the MP1800A, No. 1 is fixed.
- `<slot>` = `<DECIMAL NUMERIC PROGRAM DATA>`
  - 1 to 6 Slot No. 1 to 6
    - When using the MP1800A, the slot No. is 1 to 6.
    - When using the MT1810A, the slot No. is 1 to 4.
- `[<tab>]` = `<DECIMAL NUMERIC PROGRAM DATA>`
  - 1 to X Tab ID No. 1 to X
    - When `[<tab>]` is omitted, 1 is specified.
    - Tab ID is set to No. 1 at the left side dialog, and then the tab ID No. is set to 2, 3, 4...toward the right side.
    - The maximum number (X) of the tab ID varies depending on the module options.

<table>
<thead>
<tr>
<th>Tab ID</th>
<th>MU183020A-x12/x13</th>
<th>MU183020A-x22/x23, MU183021A</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Output</td>
<td>Output</td>
</tr>
<tr>
<td>2</td>
<td>Pattern</td>
<td>Pattern</td>
</tr>
<tr>
<td>3</td>
<td>Error Addition</td>
<td>Error Addition</td>
</tr>
<tr>
<td>4</td>
<td>Misc 1</td>
<td>Precode</td>
</tr>
<tr>
<td>5</td>
<td>Misc 2</td>
<td>Misc 1</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>Misc 2</td>
</tr>
</tbody>
</table>

Function
Displays the specified module screen to the front.

Example
To display the Pattern tab of the MU183020A module:
(when installing the MU183020A in the unit1 slot1)
> :DISPLAY:ACTive 1,1,2

Compatibility
Incompatible with existing models.
### 7.11.1.2 File menu commands

#### Table 7.11.1.2-1  File Menu Setting Commands

<table>
<thead>
<tr>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quick Open</td>
<td>:SYSTem:MMEMory:QRECall</td>
</tr>
<tr>
<td>Quick Save</td>
<td>:SYSTem:MMEMory:QSTore</td>
</tr>
<tr>
<td>Open</td>
<td>:SYSTem:MMEMory:RECall</td>
</tr>
<tr>
<td>Save</td>
<td>:SYSTem:MMEMory:STORe</td>
</tr>
<tr>
<td>Screen Copy Execute</td>
<td>:SYSTem:PRINT:COPY</td>
</tr>
<tr>
<td>Printer Setup</td>
<td>:SYSTem:PRINT:PRINter:SET</td>
</tr>
<tr>
<td></td>
<td>:SYSTem:PRINT:PRINter:SET?</td>
</tr>
<tr>
<td>Combination Setting</td>
<td>:COMBination:OPERation:ABILITY:CHSync?</td>
</tr>
<tr>
<td></td>
<td>:COMBination:OPERation:CHSetting</td>
</tr>
<tr>
<td></td>
<td>:COMBination:OPERation:PPGChsync</td>
</tr>
<tr>
<td></td>
<td>:COMBination:OPERation:USYNcout</td>
</tr>
<tr>
<td></td>
<td>:COMBination:OPERation:USYNcout?</td>
</tr>
<tr>
<td>Initialize</td>
<td>:SYSTem:MEMory:INITialize</td>
</tr>
</tbody>
</table>

#### :SYSTem:MMEMory:QRECall <file_name>

**Parameter**

- `<file_name> = <STRING PROGRAM DATA>`
- "<drv>:\[<dir>]<file>"
- `<drv> = C, D, E, F`
- `<dir> = <dir1>\<dir2>\...(Omitted when root directory)`
- `<file> = File name`

**Function**

Opens all setting data.

**Example**

To read all setting files from the specified save destination:

```
> :SYSTem:MMEMory:QRECall "C:\Test\example"
```

**Compatibility**

Commands are compatible with those of the MP1632C.

Parameters are not compatible with the existing parameters.
7.11 28G/32G bit/s PPG Commands

:SYSTem:MMEMory:QSTore <file_name>,<comment>

Parameter

<file_name> = <STRING PROGRAM DATA>
"<drv>\[<dir>]<file>"
<drv> = C, D, E, F
<dir> = <dir1>\<dir2>...(Omitted when root directory)
<file> = File name
<comment> = <STRING PROGRAM DATA>
"XXXXXX..." Specify a comment of a character string within 60 characters into the file.

Function

Executes "Quick Save".

Note:

Note that if the name of the saved file is changed, it is no longer possible to read the setting data.

Example

To save all setting files with a comment and measurement result data, specifying save destination:

> :SYSTem:MMEMory:QSTore "C:\Test\example","setup all"

Compatibility

Commands are compatible with those of the MP1632C.
Parameters are not compatible with the existing parameters.
### :SYStem:MMEMory:RECall <file_name>,<module>,<data_type>

#### Parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;file_name&gt;</td>
<td>= &lt;STRING PROGRAM DATA&gt;</td>
</tr>
<tr>
<td>&quot;&lt;drv&gt;[&lt;dir&gt;]&lt;file&gt;&quot;</td>
<td></td>
</tr>
<tr>
<td>&lt;drv&gt;</td>
<td>= C, D, E, F</td>
</tr>
<tr>
<td>&lt;dir&gt;</td>
<td>= &lt;dir1&gt;&lt;dir2&gt;... (Omitted when root directory)</td>
</tr>
<tr>
<td>&lt;file&gt;</td>
<td>= File name</td>
</tr>
<tr>
<td>&lt;module&gt;</td>
<td>= &lt;STRING PROGRAM DATA&gt;</td>
</tr>
<tr>
<td>&quot;&lt;unit&gt;:&lt;slot&gt;:&lt;port&gt; &lt;module&gt;&quot;</td>
<td></td>
</tr>
<tr>
<td>&lt;unit&gt;</td>
<td>= 1, 2, 3, 4</td>
</tr>
<tr>
<td>&lt;slot&gt;</td>
<td>= 1, 2, 3, 4..., 6</td>
</tr>
<tr>
<td>&lt;port&gt;</td>
<td>= 1</td>
</tr>
<tr>
<td>&lt;module&gt;</td>
<td>= Module model name</td>
</tr>
<tr>
<td>&lt;data_type&gt;</td>
<td>= &lt;CHARACTER PROGRAM DATA&gt;</td>
</tr>
<tr>
<td>P32</td>
<td>28G/32G PPG Setup</td>
</tr>
<tr>
<td>P32X</td>
<td>28G/32G 4ch PPG Setup</td>
</tr>
<tr>
<td>E32</td>
<td>28G/32G ED Setup</td>
</tr>
<tr>
<td>E32X</td>
<td>28G/32G 4ch ED Setup</td>
</tr>
<tr>
<td>ERA_D1</td>
<td>32G ED Error/Alarm Result (Data1)</td>
</tr>
<tr>
<td>ERA_D2</td>
<td>32G ED Error/Alarm Result (Data2)</td>
</tr>
<tr>
<td>ERA_D3</td>
<td>32G ED Error/Alarm Result (Data3)</td>
</tr>
<tr>
<td>ERA_D4</td>
<td>32G ED Error/Alarm Result (Data4)</td>
</tr>
<tr>
<td>ERA_C1</td>
<td>32G ED Error/Alarm Result (2ch Combi1-2)</td>
</tr>
<tr>
<td>ERA_C2</td>
<td>32G ED Error/Alarm Result (2ch Combi3-4)</td>
</tr>
<tr>
<td>ERA_C4</td>
<td>32G ED Error/Alarm Result (4ch Combi1-4)</td>
</tr>
</tbody>
</table>

#### Function

Opens module settings and result data.

#### Example

To read a file for the PPG module:

```plaintext
> :SYStem:MMEMory:RECall "C:\Test\example","1:3:1 MU183020A",P32
```

#### Compatibility

Commands are compatible with those of the MP1632C1 and MP1776A.
Parameters are not compatible with the existing parameters.
### :SYSTem:MMEMory:STORe

<file_name>,<module>,<data_type>,<file_type>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;file_name&gt;</td>
<td>= &lt;STRING PROGRAM DATA&gt; &quot;&lt;drv&gt;:&lt;dir1&gt;:&lt;file&gt;&quot;</td>
</tr>
<tr>
<td>&lt;drv&gt;</td>
<td>= C, D, E, F</td>
</tr>
<tr>
<td>&lt;dir&gt;</td>
<td>= &lt;dir1&gt;:&lt;dir2&gt;... (Omitted when root directory)</td>
</tr>
<tr>
<td>&lt;file&gt;</td>
<td>= File name</td>
</tr>
<tr>
<td>&lt;module&gt;</td>
<td>= &lt;STRING PROGRAM DATA&gt; &quot;&lt;unit&gt;:&lt;slot&gt;:&lt;port&gt; &lt;module&gt;&quot;</td>
</tr>
<tr>
<td>&lt;unit&gt;</td>
<td>= 1, 2, 3, 4</td>
</tr>
<tr>
<td>&lt;slot&gt;</td>
<td>= 1, 2, 3, 4..., 6</td>
</tr>
<tr>
<td>&lt;port&gt;</td>
<td>= 1</td>
</tr>
<tr>
<td>&lt;module&gt;</td>
<td>= Module model name</td>
</tr>
<tr>
<td>&lt;data_type&gt;</td>
<td>= &lt;CHARACTER PROGRAM DATA&gt;</td>
</tr>
<tr>
<td>PPG</td>
<td>PPG Setup</td>
</tr>
<tr>
<td>ED</td>
<td>ED Setup</td>
</tr>
<tr>
<td>SYN</td>
<td>Synthesizer Setup</td>
</tr>
<tr>
<td>XFP</td>
<td>XFP Setup</td>
</tr>
<tr>
<td>SFP</td>
<td>SFP Setup</td>
</tr>
<tr>
<td>ERA</td>
<td>Error/Alarm Result</td>
</tr>
<tr>
<td>OTX</td>
<td>Transmitter Setup</td>
</tr>
<tr>
<td>MX1</td>
<td>MUX Setup</td>
</tr>
<tr>
<td>MX2</td>
<td>MUX Setup</td>
</tr>
<tr>
<td>DX1</td>
<td>DEMUX Setup</td>
</tr>
<tr>
<td>DX2</td>
<td>DEMUX Setup</td>
</tr>
<tr>
<td>DER</td>
<td>DEMUX Error/Alarm Result</td>
</tr>
<tr>
<td>P32</td>
<td>28G/32G PPG Setup</td>
</tr>
<tr>
<td>P32X</td>
<td>28G/32G 4ch PPG Setup</td>
</tr>
<tr>
<td>E32</td>
<td>28G/32G ED Setup</td>
</tr>
<tr>
<td>E32X</td>
<td>28G/32G 4ch ED Setup</td>
</tr>
<tr>
<td>&lt;file_type&gt;</td>
<td>= &lt;CHARACTER PROGRAM DATA&gt;</td>
</tr>
<tr>
<td>BIN</td>
<td>Binary File</td>
</tr>
<tr>
<td>CSV</td>
<td>CSV File</td>
</tr>
<tr>
<td>TXT</td>
<td>Text File</td>
</tr>
</tbody>
</table>

**Function**: Saves the setting and measurement result data for a specified module.

**Note**: Note that if the name of the saved file is changed, it is no longer possible to read the setting data.

**Example**: To save the setting file for the ED module:

```
> :SYSTem:MMEMory:STORe "C:\Test\example","1:3:1 MU183020A",P32,BIN
```

**Compatibility**: Commands are compatible with those of the MP1632C and MP1776A. Parameters are not compatible with the existing parameters.
### :SYStem:PRINt:COPY

**Function**
Executes Screen Copy.

**Example**
>`:SYStem:PRINt:COPY`

**Compatibility**
Compatible with the MP1632C and MP1776A.

### :SYStem:PRINt:PRINter:SET <printer>,<direction>

**Parameter**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;printer&gt;</td>
<td>&lt;STRING PROGRAM DATA&gt;</td>
</tr>
<tr>
<td></td>
<td>&quot;XXXXXXX...&quot;  Printer name</td>
</tr>
<tr>
<td>&lt;direction&gt;</td>
<td>&lt;CHARACTER PROGRAM DATA&gt;</td>
</tr>
<tr>
<td>VER</td>
<td>Vertical</td>
</tr>
<tr>
<td>HOR</td>
<td>Horizontal</td>
</tr>
</tbody>
</table>

**Function**
Sets the printer and print direction.

**Example**
>`:SYStem:PRINt:PRINter:SET "printer",VER`

**Compatibility**
Incompatible with existing models.

### :SYStem:PRINt:PRINter:SET?

**Response**

<table>
<thead>
<tr>
<th>Response</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;printer&gt;</td>
<td>&lt;STRING RESPONSE DATA&gt;</td>
</tr>
<tr>
<td></td>
<td>&quot;XXXXXXX...&quot;  Printer name</td>
</tr>
<tr>
<td>&lt;direction&gt;</td>
<td>&lt;CHARACTER RESPONSE DATA&gt;</td>
</tr>
<tr>
<td>VER</td>
<td>Vertical</td>
</tr>
<tr>
<td>HOR</td>
<td>Horizontal</td>
</tr>
</tbody>
</table>

**Function**
Queries the default printer and print direction.

**Example**
>`:SYStem:PRINt:PRINter:SET?`

< "printer",VER

**Compatibility**
Incompatible with existing models.
:COMBination:OPERation:ABILity:CHSYnc? [<unit>]

Parameter

[<unit>] = <DECIMAL NUMERIC PROGRAM DATA>
1 to 4 Mainframe No.1 to 4
When installing the MT1810A, the mainframe No. is specified.
Can be omitted. Mainframe No. 1 is specified when omitted.

Response

<numeric> = <NR1 NUMERIC RESPONSE DATA>
0 to 63 Total number of PPGs that can configure channel synchronization (decimal)

Available bits
0 No Channel Synchronization
1 (Bit 0) PPG in Slot 1
2 (Bit1) PPG in Slot 2
4 (Bit2) PPG in Slot 3
8 (Bit3) PPG in Slot 4
16 (Bit4) PPG in Slot 5
32 (Bit5) PPG in Slot 6

Function
Queries the slot where PPG that can configure channel synchronization combination is inserted.

Example
To query the slot in Unit 3 where PPG that can configure channel synchronization is inserted:
> :COMBination:OPERation:ABILity:CHSYnc? 3
< 7

Compatibility
Incompatible with existing models.
Chapter 7  SCPI Commands

**:COMBination:OPERation:CHSetting <configuration>[,<unit>]**

Parameter

- `<configuration>` = `<DECIMAL NUMERIC PROGRAM DATA>`
- 0 to 62  Total number of PPGs that can configure channel synchronization (decimal)

Available bits

- 0  Channel Synchronization OFF
- 2 (Bit 1)  PPG in Slot 2
- 4 (Bit 2)  PPG in Slot 3
- 8 (Bit 3)  PPG in Slot 4
- 16 (Bit 4)  PPG in Slot 5
- 32 (Bit 5)  PPG in Slot 6

- `[<unit>]` = `<DECIMAL NUMERIC PROGRAM DATA>`
- 1 to 4  Mainframe Nos. 1 to 4

When installing the MT1810A, the mainframe No. is specified. Can be omitted. Mainframe No. 1 is specified when omitted.

Function  Specify the slot where the PPG for which channel synchronization is to be set is inserted.

Example  To set channel synchronization for the PPGs in Slots 1 through 4 of Unit 3:

> :COMBination:OPERation:CHSetting 14,3

Compatibility  Incompatible with existing models.

**:COMBination:OPERation:PPGChsync <configuration>[,<unit>]**

Parameter

- `<configuration>` = `<DECIMAL NUMERIC PROGRAM DATA>`
- 0  2ch Combination
- 1  4ch Combination
- 3  Channel Synchronization
- 4  64G × 2ch Combination

- `[<unit>]` = `<DECIMAL NUMERIC PROGRAM DATA>`
- 1 to 4  Mainframe Nos. 1 to 4

When installing the MT1810A, the mainframe No. is specified. Can be omitted. Mainframe No. 1 is specified when omitted.

Function  Sets the combination type for inter-module sync. You need to set the Channel Synchronization in advance with the :COMBination:OPERation:CHSetting command.

Example  To set the PPG (Unit3) to 2ch Combination:

> :COMBination:OPERation:PPGChsync 0,3

Compatibility  Incompatible with existing models.
**:COMBination:OPERation:USYNcout <boolean>**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;boolean&gt; = &lt;BOOLEAN PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Unit Sync OFF</td>
</tr>
<tr>
<td>1</td>
<td>Unit Sync ON</td>
</tr>
</tbody>
</table>

**Function**
Sets the Unit Sync to On/Off.

**Example**
To set the Unit Sync to On:
> :COMBination:OPERation:USYNcout 1

**Compatibility**
Incompatible with existing models.

**:COMBination:OPERation:USYNcout?**

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;numeric&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Unit Sync OFF</td>
</tr>
<tr>
<td>1</td>
<td>Unit Sync ON</td>
</tr>
</tbody>
</table>

**Function**
Queries the Unit Sync On/Off.

**Example**
> :COMBination:OPERation:USYNcout?  
< 1

**Compatibility**
Incompatible with existing models.

**:SYSTem:MEMory:INITialize**

**Function**
Initializes the internal setting data to the initial settings at factory shipment.

**Example**
> :SYSTem:MEMory:INITialize

**Compatibility**
Compatible with the MP1632C and MP1776A.
Chapter 7  SCPI Commands

7.11.1.3 Common function buttons

Figure 7.11.1.3-1  Common Function Buttons

Table 7.11.1.3-1  Common Function Button Setting Commands

<table>
<thead>
<tr>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>Output ON/OFF</td>
<td>:SOURce:OUTPut:ASET</td>
</tr>
<tr>
<td></td>
<td>:SOURce:OUTPut:ASET?</td>
</tr>
<tr>
<td>Error Addition ON/OFF</td>
<td>:SOURce:PATTern:EADDition:ASET</td>
</tr>
<tr>
<td></td>
<td>:SOURce:PATTern:EADDition:ASET?</td>
</tr>
<tr>
<td>Single Error Addition</td>
<td>:SOURce:PATTern:EADDition:ASINgle</td>
</tr>
<tr>
<td>Open</td>
<td>Refer to 7.11.1.2 “File menu commands”.</td>
</tr>
<tr>
<td>Save</td>
<td>Refer to 7.11.1.2 “File menu commands”.</td>
</tr>
<tr>
<td>Print</td>
<td>Refer to 7.11.1.2 “File menu commands”.</td>
</tr>
<tr>
<td>System Error ON/OFF</td>
<td>:SYSTem:BEEPer:SYSTem:SET</td>
</tr>
<tr>
<td></td>
<td>:SYSTem:BEEPer:SYSTem:SET?</td>
</tr>
<tr>
<td>System Error</td>
<td>:SYSTem:BEEPer:SYSTem:TYPE</td>
</tr>
<tr>
<td></td>
<td>:SYSTem:BEEPer:SYSTem:TYPE?</td>
</tr>
<tr>
<td>Unit Sync Output</td>
<td>:SOURce:PATTern:USYNcout</td>
</tr>
</tbody>
</table>

:SOURce:OUTPut:ASET <boolean>

Parameter:  
<boolean> = <BOOLEAN PROGRAM DATA>

- OFF or 0  Output OFF
- ON or 1   Output ON

Function:  
Sets Data and Clock outputs of PPG ON or OFF.

Example:  
To set Data and Clock outputs of PPG to ON:

> :SOURce:OUTPut:ASET ON

Compatibility:  
Incompatible with existing models.

:SOURce:OUTPut:ASET?

Response:  
<numeric> = <NR1 NUMERIC RESPONSE DATA>

- 0  Output OFF
- 1  Output ON

Function:  
Queries the ON/OFF state for Data and Clock outputs of PPG.

Example:  
> :SOURce:OUTPut:ASET?
> < 1

Compatibility:  
Incompatible with existing models.
7.11 28G/32G bit/s PPG Commands

:SOURce:PATTern:EADDition:ASET <boolean>

Parameter
<boolean> = <BOOLEAN PROGRAM DATA>
OFF or 0    Error addition OFF
ON or 1     Error addition ON

Function
Sets error addition for all valid modules ON/OFF.

Example
To set error addition for all valid modules to ON:
> :SOURce:PATTern:EADDition:ASET ON

Compatibility
Incompatible with existing models.

:SOURce:PATTern:EADDition:ASET?

Response
<numeric> = <NR1 NUMERIC RESPONSE DATA>
0    Error addition OFF
1    Error addition ON

Function
Queries the error addition ON/OFF state for all valid modules.

Example
> :SOURce:PATTern:EADDition:ASET?
< 1

Compatibility
Incompatible with existing models.

:SOURce:PATTern:EADDition:ASINgle

Function
Adds a single error for all valid modules.

Example
> :SOURce:PATTern:EADDition:ASINgle

Compatibility
Incompatible with existing models.

:SYStem:BEEPer:SYSTem:SET <boolean>

Parameter
<boolean> = <BOOLEAN PROGRAM DATA>
OFF or 0    Buzzer OFF
ON or 1     Buzzer ON

Function
Sets buzzer at error occurrence ON/OFF.

Example
To set buzzer at error occurrence ON:
> :SYStem:BEEPer:SYSTem:SET ON

Compatibility
Compatible with the MP1632C and MP1776A.
Chapter 7  SCPI Commands

:SYSTem:BEEPer:SYSTem:SET?
Response  
<numERIC> = <NR1 NUMERIC RESPONSE DATA>
0  Buzzer OFF
1  Buzzer ON
Function  Queries the buzzer ON/OFF state at error occurrence.
Example  
> :SYSTem:BEEPer:SYSTem:SET?
< 1
Compatibility  Compatible with the MP1632C and MP1776A.

:SYSTem:BEEPer:SYSTem:TYPE <type>,<boolean>
Parameter  
<type> = <CHARACTER PROGRAM DATA>
PUNLock  PLL Unlock
FAN  FAN
TEMPerature  Temperature
ALL  Selects all system errors
<boolean> = <BOOLEAN PROGRAM DATA>
OFF or 0
ON or 1
Function  Sets system error buzzer for the target item ON/OFF.
Example  To set system error buzzer for "Temperature" ON:
> :SYSTem:BEEPer:SYSTem:TYPE TEMPerature,ON
Compatibility  Partially compatible with the MP1632C.

:SYSTem:BEEPer:SYSTem:TYPE?
Response  
<type> = <CHARACTER RESPONSE DATA>
PUNL, FAN, TEMP, ALL
XXX, XXX,  Errors for which buzzer is set to ON are
delimited with commas (,) and returned.
NONE  Buzzer is set to OFF for all items.
Function  Queries the ON/OFF state of system error buzzer for target items.
Example  
> :SYSTem:BEEPer:SYSTem:TYPE?
< PUNL,TEMP
Compatibility  Compatible with the MP1632C.

:SOURce:PATTern:USYNcout
Parameter  None
Function  Outputs the timing signal to synchronize the pattern between multiple main frames.
Example  
> :SOURce:PATTern:USYNcout
Compatibility  Incompatible with existing models.
7.11.4 PPG status commands

The PPG status commands query the status (such as an alarm) of the MU183020A 28G/32G bit/s Pulse Pattern Generator and MU183021A 28G/32G bit/s 4ch Pulse Pattern Generator.

Note that all the commands described below are incompatible with existing PPGs.

:INStrument:PPGG32[:EVENt]?

Response

<numeric> = <NR1 NUMERIC RESPONSE DATA>

0 to 15  Sum total of the bit of event register (DECIMAL)

Available bits
1 (Bit0)  Delay Calibration Require (Data1)
2 (Bit1)  Delay Calibration Require (Data2)
4 (Bit2)  Delay Calibration Require (Data3)
8 (Bit3)  Delay Calibration Require (Data4)

Function  Queries event at 28G/32G bit/s PPG status.

Example

> :INStrument:PPGG32:EVENt?
or
> :INStrument:PPGG32?
< 4

:INStrument:PPGG32:CONDition?

Response

<numeric> = <NR1 NUMERIC RESPONSE DATA>

0 to 15  Sum total of the bit of condition register (DECIMAL)

Available bits
1 (Bit0)  Delay Calibration Require (Data1)
2 (Bit1)  Delay Calibration Require (Data2)
4 (Bit2)  Delay Calibration Require (Data3)
8 (Bit3)  Delay Calibration Require (Data4)

Function  Queries condition at 28G/32G bit/s PPG status.

Example

> :INStrument:PPGG32:CONDition?
< 4
### :INSTrument:PPGG32:PTRansition <numeric>

**Parameter**

<table>
<thead>
<tr>
<th>&lt;numeric&gt;</th>
<th>&lt;DECIMAL NUMERIC PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 to 15</td>
<td>Sum total of the bit of transition filter (DECIMAL)</td>
</tr>
</tbody>
</table>

**Available bits**

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Delay Calibration Require (Data1)</td>
</tr>
<tr>
<td>1</td>
<td>Delay Calibration Require (Data2)</td>
</tr>
<tr>
<td>2</td>
<td>Delay Calibration Require (Data3)</td>
</tr>
<tr>
<td>4</td>
<td>Delay Calibration Require (Data4)</td>
</tr>
</tbody>
</table>

**Function**

Sets transition filter (positive direction transition) at 28G/32G bit/s PPG status.

**Example**

To set the transition filter (positive direction transition) at 28G/32G bit/s PPG status to 1:

```plaintext
> :INSTrument:PPGG32:PTRansition 4
```

### :INSTrument:PPGG32:PTRansition? 

**Response**

<table>
<thead>
<tr>
<th>&lt;numeric&gt;</th>
<th>&lt;NR1 NUMERIC RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 to 15</td>
<td>Sum total of the bit of transition filter (DECIMAL)</td>
</tr>
</tbody>
</table>

**Function**

Queries contents of transition filter (positive direction transition) at 28G/32G bit/s PPG status.

**Example**

```plaintext
> :INSTrument:PPGG32:PTRansition?
< 4
```

### :INSTrument:PPGG32:NTRansition <numeric>

**Parameter**

<table>
<thead>
<tr>
<th>&lt;numeric&gt;</th>
<th>&lt;DECIMAL NUMERIC PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 to 15</td>
<td>Sum total of the bit of transition filter (DECIMAL)</td>
</tr>
</tbody>
</table>

**Available bits**

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Delay Calibration Require (Data1)</td>
</tr>
<tr>
<td>1</td>
<td>Delay Calibration Require (Data2)</td>
</tr>
<tr>
<td>2</td>
<td>Delay Calibration Require (Data3)</td>
</tr>
<tr>
<td>4</td>
<td>Delay Calibration Require (Data4)</td>
</tr>
</tbody>
</table>

**Function**

Sets transition filter (negative direction transition) at 28G/32G bit/s PPG status.

**Example**

To set the transition filter (negative direction transition) at 28G/32G bit/s PPG status to 1:

```plaintext
> :INSTrument:PPGG32:NTRansition 4
```
### :INSTrument:PPGG32:NTRansition?

**Response**


\( \text{<numeric>} = \text{<NR1 NUMERIC RESPONSE DATA>} \)

0 to 15  
Sum total of the bit of transition filter  
(DECIMAL)

**Function**

Queries contents of transition filter (negative direction transition) at 28G/32G bit/s PPG status.

**Example**

> :INSTrument:PPGG32:NTRansition?

< 4

---

### :INSTrument:PPGG32:RESet

**Function**

Initializes event at 28G/32G bit/s PPG status.

**Example**

> :INSTrument:PPGG32:RESet
7.11.1.5 Data Interface Setting Commands

![Image](Image.png)

**Figure 7.11.1.5-1 Data Interface Select**

<table>
<thead>
<tr>
<th>Setting Item</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Interface Select</td>
<td>::INTerface::ID, ::INTerface::ID?</td>
</tr>
</tbody>
</table>

### ::INTerface::ID < interface_number >

**Parameter**

- `<interface_number> = <DECIMAL NUMERIC PROGRAM DATA>`
- `1 to 4` Data channel

**Function**

Sets the control channel number.

The setting range varies by the model name.

- MU183020A-x12/x13: 1
- MU183020A-x22/x23: 1 to 2
- MU183021A: 1 to 4

**Example**

To specify Channel 3 of MU183021A:

```
> ::INTerface::ID 3
```

### ::INTerface::ID?

**Response**

- `<interface_number> = <NR1 NUMERIC RESPONSE DATA>`
- `1 to 4`

**Function**

Queries the channel of MU183020A and MU183021A specified as control targets.

**Example**

```
> ::INTerface::ID?
< 3
```
7.11.6 Pattern Editor

This section describes the commands for saving and loading pattern files to MU183020A/21A. Before executing the commands, use the :MODule:ID and :INTerface:ID commands to specify both slot number and data interface number of the module you want to remote-control. For how to specify the slot number with the :MODule:ID command, refer to 7.1, “Common Commands” and 9.1, “Basic Operation of Device Messages”.

<table>
<thead>
<tr>
<th>Setting Item</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Open</td>
<td>:SYSTem:MMEMory:PATTern:RECall</td>
</tr>
<tr>
<td>Save</td>
<td>:SYSTem:MMEMory:PATTern:STORe</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;file_name&gt; = &lt;STRING PROGRAM DATA&gt;</td>
<td></td>
</tr>
<tr>
<td>&quot;&lt;drv&gt;:[&lt;dir&gt;]&lt;file&gt;&quot;</td>
<td></td>
</tr>
<tr>
<td>&lt;drv&gt; = C, D, E, F</td>
<td></td>
</tr>
<tr>
<td>&lt;dir&gt; = &lt;dir1&gt;&lt;dir2&gt;...(Omitted when root directory)</td>
<td></td>
</tr>
<tr>
<td>&lt;file&gt; = File name</td>
<td></td>
</tr>
<tr>
<td>&lt;file_type&gt; = &lt;CHARACTER PROGRAM DATA&gt;</td>
<td></td>
</tr>
<tr>
<td>BIN = Binary File</td>
<td></td>
</tr>
<tr>
<td>TXT = Text File</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Opens the pattern file.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Example</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>To load the pattern file, specifying the save location (C:\Test), file name (example) and file format (Binary):</td>
<td></td>
</tr>
<tr>
<td>&gt; :SYSTem:MMEMory:PATTern:RECall &quot;C:\Test\example&quot;,BIN</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Compatibility</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Incompatible with existing models.</td>
<td></td>
</tr>
</tbody>
</table>
Chapter 7  SCPI Commands

:SYSTem:MEMORY:PATTern:STORe <file_name>,<file_type>

Parameter
<file_name> = <STRING PROGRAM DATA>
"<drv>\[<dir>]<file>"
<drv> = C, D, E, F
<dir> = <dir1>\<dir2>\…(Omitted when root directory)
<file> = File name
<file_type> = <CHARACTER PROGRAM DATA>
BIN Binary File
TXT Text File

Function
Saves the pattern file.

Note:
The settings will not be read from the saved file if the file name is changed.

Example
To save the pattern file, specifying the save location (C:\Test), file name (example) and file format (Text):
> :SYSTem:MEMORY:PATTern:STORe "C:\Test\example",TXT

Compatibility
Incompatible with existing models.
7.11.2 Output Tab

Figure 7.11.2-1  Output Tab
7.11.2.1 Output Setting Commands

![Output Settings]

Figure 7.11.2.1-1  Output Settings

Table 7.11.2.1-1  Output Setting Commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1]</td>
<td>Bit Rate Setting</td>
<td>:OUTPut:DATA:STANdard</td>
</tr>
<tr>
<td></td>
<td>Preset Standard Bit Rate</td>
<td>:OUTPut:DATA:STANdard?</td>
</tr>
<tr>
<td></td>
<td>Bit Rate Monitor</td>
<td>:OUTPut:DATA:BITRate?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:OUTPut:CLOSEk:OUTPut?</td>
</tr>
</tbody>
</table>
### :OUTPut:DATA:STANdard <freq>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Variable</th>
<th>Variable*</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;Variable&quot;</td>
<td>&quot;PCI_EX1&quot; PCI Express Gen1</td>
<td>2.500000 Gbit/s</td>
</tr>
<tr>
<td></td>
<td>&quot;PCI_EX2&quot; PCI Express Gen2</td>
<td>5.000000 Gbit/s</td>
</tr>
<tr>
<td></td>
<td>&quot;USB3&quot; USB3.0</td>
<td>5.000000 Gbit/s</td>
</tr>
<tr>
<td></td>
<td>&quot;PCI_EX3&quot; PCI Express Gen3</td>
<td>8.000000 Gbit/s</td>
</tr>
<tr>
<td>&quot;8G_FC&quot;</td>
<td>8GFC</td>
<td>8.500000 Gbit/s</td>
</tr>
<tr>
<td>&quot;OC_192&quot;</td>
<td>OC192/STM64</td>
<td>9.953280 Gbit/s</td>
</tr>
<tr>
<td>&quot;IB_QDR&quot;</td>
<td>Infiniband QDR</td>
<td>10.000000 Gbit/s</td>
</tr>
<tr>
<td>&quot;USB3_1&quot;</td>
<td>USB3.1</td>
<td>10.000000 Gbit/s</td>
</tr>
<tr>
<td>&quot;10G_FC&quot;</td>
<td>10GFC</td>
<td>10.518750 Gbit/s</td>
</tr>
<tr>
<td>&quot;OC_192_G975&quot;</td>
<td>G975 FEC</td>
<td>10.664228 Gbit/s</td>
</tr>
<tr>
<td>&quot;OC_192_G709&quot;</td>
<td>OTU2</td>
<td>10.709225 Gbit/s</td>
</tr>
<tr>
<td>&quot;10GbE_G709&quot;</td>
<td>10GbE over FEC</td>
<td>11.095700 Gbit/s</td>
</tr>
<tr>
<td>&quot;10G_FC_G709&quot;</td>
<td>10GFC over FEC</td>
<td>11.316800 Gbit/s</td>
</tr>
<tr>
<td>&quot;16G_FC&quot;</td>
<td>16GFC</td>
<td>14.025000 Gbit/s</td>
</tr>
<tr>
<td>&quot;IB_FDR&quot;</td>
<td>Infiniband FDR</td>
<td>14.062500 Gbit/s</td>
</tr>
<tr>
<td>&quot;PCI_EX4&quot;</td>
<td>PCI Express Gen4</td>
<td>16.000000 Gbit/s</td>
</tr>
<tr>
<td>&quot;SAS&quot;</td>
<td>SAS</td>
<td>24.000000 Gbit/s</td>
</tr>
<tr>
<td>&quot;IB_EDR&quot;</td>
<td>Infiniband EDR</td>
<td>25.781250 Gbit/s</td>
</tr>
<tr>
<td>&quot;100GbE&quot;</td>
<td>100GbE(25.78x4)</td>
<td>25.781250 Gbit/s</td>
</tr>
<tr>
<td>&quot;100G_OTU4&quot;</td>
<td>100G OTU4</td>
<td>27.952496 Gbit/s</td>
</tr>
<tr>
<td>&quot;32G_FC&quot;</td>
<td>32GFC</td>
<td>28.050000 Gbit/s</td>
</tr>
<tr>
<td>&quot;100G_ULH&quot;</td>
<td>100G ULH</td>
<td>32.100000 Gbit/s</td>
</tr>
</tbody>
</table>

*: The bit rate can be set by :OUTPut:DATA:BITRate.

**Note:**
Available preset frequencies vary depending on the presence of Option x01.

**Function:**
Specifies the preset bit rate that the 32G PPG module outputs.

**Example:**
To set the preset frequency to "USB3_1":

```plaintext
> :OUTPut:DATA:STANdard "USB3_1"
```

**Compatibility:**
Incompatible with the existing modules.
### :OUTPut:DATA:STANdard?

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;freq&gt; = &lt;STRING RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>&quot;Variable&quot;,&quot;PCI_EX1&quot;,&quot;PCI_EX2&quot;,&quot;USB3&quot;,&quot;PCI_EX3&quot;,</td>
</tr>
<tr>
<td></td>
<td>&quot;8G_FC&quot;,&quot;OC_192&quot;,&quot;IB_QDR&quot;,&quot;USB3_1&quot;,&quot;10G_FC&quot;,&quot;OC_192_G975&quot;,</td>
</tr>
<tr>
<td></td>
<td>&quot;OC_192_G709&quot;,&quot;10GbE_G709&quot;,&quot;10G_FC_G709&quot;,</td>
</tr>
<tr>
<td></td>
<td>&quot;16G_FC&quot;,&quot;IB_FDR&quot;,&quot;PCI_EX4&quot;,&quot;SAS&quot;,&quot;IB_EDR&quot;,&quot;100GbE&quot;,</td>
</tr>
<tr>
<td></td>
<td>&quot;100G_OTU&quot;,&quot;32G_FC&quot;,&quot;100G_ULH&quot;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Function</th>
<th>Queries the preset bit rate that the 32G PPG module outputs.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example</td>
<td>&gt; :OUTPut:DATA:STANdard?</td>
</tr>
<tr>
<td></td>
<td>&lt; &quot;USB3_1&quot;</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Incompatible with the existing modules.</td>
</tr>
</tbody>
</table>
7.11 28G/32G bit/s PPG Commands

:OUTPut:DATA:BITRate <numeric>

Parameter <numeric> = <DECIMAL NUMERIC PROGRAM DATA>

When synchronized with MU181000A/B and set to Full clock output:
2.400000 to 12.500000 Gbit/s / 0.000001 Gbit/s step
12.500002 to 20.000000 Gbit/s / 0.000002 Gbit/s step
20.000002 to 25.000000 Gbit/s / 0.000002 Gbit/s step
25.000004 to 28.100000 Gbit/s / 0.000004 Gbit/s step*1
25.000004 to 32.100000 Gbit/s / 0.000004 Gbit/s step*2

When synchronized with MU181000A/B and set to Half clock output:
2.400000 to 25.000000 Gbit/s / 0.000002 Gbit/s step
25.000004 to 28.100000 Gbit/s / 0.000004 Gbit/s step*1
25.000004 to 32.100000 Gbit/s / 0.000004 Gbit/s step*2

When synchronized with MU181000A/B + MU181500B and set to Full clock output:
2.400000 to 3.125000 Gbit/s / 0.000001 Gbit/s step
3.200001 to 6.250000 Gbit/s / 0.000001 Gbit/s step
6.400001 to 12.500000 Gbit/s / 0.000001 Gbit/s step
12.800002 to 15.000000 Gbit/s / 0.000002 Gbit/s step
15.000002 to 20.000000 Gbit/s / 0.000002 Gbit/s step
20.000002 to 25.000000 Gbit/s / 0.000002 Gbit/s step
25.600004 to 28.100000 Gbit/s / 0.000004 Gbit/s step*1
25.600004 to 32.100000 Gbit/s / 0.000004 Gbit/s step*2

When synchronized with MU181000A/B + MU181500B and set to Half clock output:
2.400000 to 3.125000 Gbit/s / 0.000002 Gbit/s step
3.200002 to 6.250000 Gbit/s / 0.000002 Gbit/s step
6.400002 to 12.500000 Gbit/s / 0.000002 Gbit/s step
12.800002 to 25.000000 Gbit/s / 0.000002 Gbit/s step
25.600004 to 28.100000 Gbit/s / 0.000004 Gbit/s step*1
25.600004 to 32.100000 Gbit/s / 0.000004 Gbit/s step*2

*1: Not available option x01
*2: Available option x01

Function
Sets the bit rate outputting from the mainframe.
When the preset bit rate is set to Variable by :OUTPut:DATA:STANdard, the bit rate can be set by this command.

Example
To set the bit rate to 32.1 Gbit/s:
> :OUTPut:DATA:BITRate 32.100000

Compatibility
Incompatible with the existing modules.
Chapter 7  SCPI Commands

:OUTPut:DATA:BITRate?
Response
When Clock Source is not External.

<numeric> = <NR1 NUMERIC RESPONSE DATA>
2.400000 to 32.100000       2.400000 to 32.100000 Gbit/s
When Clock Source is External.

<string> = <STRING RESPONSE DATA>
"XX.XXXXX"
2.400000 to 28.100000       2.400 000 to 28.100 000 Gbit/s
(Not available option x01)
2.400000 to 32.100000       2.400 000 to 32.100 000 Gbit/s
(Available option x01)

Function
Queries operation frequency of tracking built-in synthesizer.

Example
To set clock output to ON:
>:OUTPut:CLOCk:OUTPut ON

Compatibility
Incompatible with the existing modules.

:OUTPut:CLOCk:OUTPut <boolean>
Response
<boolean> = <BOOLEAN PROGRAM DATA>
OFF or 0       Output OFF
ON or 1        Output ON

Function
Sets clock output ON/OFF.

Example
To set clock output to ON:
>:OUTPut:CLOCk:OUTPut ON

Compatibility
Compatible with MU181020A/B.

:OUTPut:CLOCk:OUTPut?
Response
<numeric> = <NR1 NUMERIC RESPONSE DATA>
0       Output OFF
1       Output ON

Function
Queries clock output ON/OFF status.

Example
>:OUTPut:CLOCk:OUTPut?
< 1

Compatibility
Compatible with MU181020A/B.
7.11.2.2 Data/XData Setting Commands

Figure 7.11.2.2-1  Data/XData Settings

Figure 7.11.2.2-2  Level Guard Settings Dialog Box

Table 7.11.2.2-1  Data/XData Setting Commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>:OUTPut:DATA:OUTPut?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:OUTPut:OFFset?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:OUTPut:DATA:TRACking?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:OUTPut:DATA:LEVGuard?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:OUTPut:DATA:LEVel?</td>
</tr>
</tbody>
</table>
Table 7.11.2.2-1  Data/XData Setting Commands (Cont’d)

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>:OUTPut:DATA:AMPLitude?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:OUTPut:DATA:AOFFset?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:OUTPut:DATA:OFFSet?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:OUTPut:DATA:ATTFactor?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:OUTPut:DATA:CPOint?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:OUTPut:DATA:HPJitter?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:OUTPut:DATA:LI Mitter:AMPLitude?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:OUTPut:DATA:LI Mitter:OFFSet?</td>
</tr>
<tr>
<td>[14]</td>
<td>Without label (Queries the status of data output setting completion)</td>
<td>:OUTPut:CHANge:CSTATE?</td>
</tr>
</tbody>
</table>

:OUTPut:DATA:OUTPut <boolean>

Parameter  
<boolean> = <BOOLEAN PROGRAM DATA>
OFF or 0    Output OFF
ON or 1     Output ON

Function  
Sets data output ON or OFF. The setting commonly applies to Data and XData.

Example  
To set data output ON:
> :OUTPut:DATA:OUTPut ON

Compatibility  
Compatible with MU181020A/B.

:OUTPut:DATA:OUTPut?

Response  
<numeric> = <NR1 NUMERIC RESPONSE DATA>

0         Output OFF
1         Output ON

Function  
Queries data output ON/OFF.

Example  
> :OUTPut:DATA:OUTPut?
< 1

Compatibility  
Compatible with MU181020A/B.
:OUTPut:OFFSet <offset>
Parameter
<offset> = <CHARACTER PROGRAM DATA>
VOH Based on the high level
VTH Based on the center level
VOL Based on the low level
Function
Sets the offset reference value for data output.
Example
To set the offset reference to Voh:
> :OUTPut:OFFSet VOH
Compatibility
Compatible with MU181020A/B.

:OUTPut:OFFSet?
Response
<offset> = <CHARACTER RESPONSE DATA>
VOH, VTH, VOL
Function
Queries the offset reference value for data outputs.
Example
> :OUTPut:OFFSet?
< VOH
Compatibility
Compatible with MU181020A/B.

:OUTPut:DATA:TRACking <boolean>
Parameter
<boolean> = <BOOLEAN PROGRAM DATA>
OFF or 0 Common setting OFF
ON or 1 Common setting ON
Function
Sets the common setting for the Data and XData output ON or OFF.
Example
To set the common setting ON:
> :OUTPut:DATA:TRACking ON
Compatibility
Compatible with MU181020A/B.

:OUTPut:DATA:TRACking?
Response
<nologic> = <NR1 NUMERIC RESPONSE DATA>
0 Common setting OFF
1 Common setting ON
Function
Queries the common setting for the Data and XData outputs ON/OFF state.
Example
> :OUTPut:DATA:TRACking?
< 1
Compatibility
Compatible with MU181020A/B.
Chapter 7  SCPI Commands

:OUTPut:DATA:LEVGuard <boolean>
Parameter
<boolean> = <BOOLEAN PROGRAM DATA>
OFF or 0       Output range limitation OFF
ON or 1        Output range limitation ON
Function
Sets the data output range limitation ON or OFF.
Example
To set the data output range limitation ON:
> :OUTPut:DATA:LEVGuard ON
Compatibility
Compatible with MU181020A/B.

:OUTPut:DATA:LEVGuard?
Response
<numERIC> = <NR1 NUMERIC RESPONSE DATA>
0       Output range limitation OFF
1       Output range limitation ON
Function
Queries the data output range limitation ON/OFF state.
Example
> :OUTPut:DATA:LEVGuard?
< 1
Compatibility
Compatible with MU181020A/B.

:OUTPut:DATA:LEVel <port>,<level>
Parameter
,port> = <CHARACTER PROGRAM DATA>
DATA       Data output
XDATa      XData output
<level> = <CHARACTER PROGRAM DATA>
VARiable   Variable
PCML       PCML level
NCML       NCML level
SCFL       SCFL level
NECL       NECL level
LVPecl     LVPECL level
Function
Sets the level of the fixed interface for the data output.
Example
To set the level of the XData output to the NECL level:
> :OUTPut:DATA:LEVel XDATa,NECL
Compatibility
Compatible with MU181020A/B.
7.11 28G/32G bit/s PPG Commands

**:OUTPut:DATA:LEVel? <port>**

Parameter: 
\(<\text{port}\> = \langle\text{CHARACTER PROGRAM DATA}\rangle\)
DATA, XDAta

Response: 
\(<\text{level}\> = \langle\text{CHARACTER RESPONSE DATA}\rangle\)
VAR, PCML, NCML, SCFL, NECL, LVP

Function: Queries the level of the fixed interface for the data output.

Example: 
To query the level of the XData output:
> :OUTPut:DATA:LEVel? XDAta
< NECL

Compatibility: Compatible with MU181020A/B.

**:OUTPut:DATA:AMPLitude <port>,<numeric>**

Parameter: 
\(<\text{port}\> = \langle\text{CHARACTER PROGRAM DATA}\rangle\)
DATA DATA output
XDAta XData output
\(<\text{numeric}\> = \langle\text{DECIMAL NUMERIC PROGRAM DATA}\rangle\)
0.500 to 2.000 0.500 to 2.000 V/0.002 V Step
(MU183020A-x12/x22, MU183021A-x12)
0.500 to 3.500 0.500 to 3.500 V/0.002 V Step
(MU183020A-x13/x23, MU183021A-x13)

Function: Sets the amplitude of the data output.

Example: 
To set the amplitude of the Data output to 1.000 V:
> :OUTPut:DATA:AMPLitude DATA, 1.000

Compatibility: Partially compatible with MU181020A/B.

**:OUTPut:DATA:AMPLitude? <port>**

Parameter: 
\(<\text{port}\> = \langle\text{CHARACTER PROGRAM DATA}\rangle\)
DATA, XDAta

Response: 
\(<\text{numeric}\> = \langle\text{NR2 NUMERIC RESPONSE DATA}\rangle\)
0.500 to 2.000 0.500 to 2.000 V
(MU183020A-x12/x22, MU183021A-x12)
0.500 to 3.500 0.500 to 2.000 V
(MU183020A-x13/x23, MU183021A-x13)

Function: Queries the amplitude of the data output.

Example: 
To query the amplitude of the Data output:
> :OUTPut:DATA:AMPLitude? DATA
< 1.000

Compatibility: Compatible with MU181020A/B.
Chapter 7  SCPI Commands

:OUTPut:DATA:AOFFset <boolean>
Parameter  
<boolean> = <BOOLEAN PROGRAM DATA>
  OFF or 0 Offset OFF (DC output)
  ON or 1 Offset ON (AC output)
Function  
Sets the data output offset ON or OFF.
Example  
To set the data output offset ON:
  > :OUTPut:DATA:AOFFset ON
Compatibility  
Compatible with MU181020A/B.

:OUTPut:DATA:AOFFset?
Response  
<numeric> = <NR1 NUMERIC RESPONSE DATA>
  0 Offset OFF
  1 Offset ON
Function  
Queries the data output offset ON/OFF state.
Example  
> :OUTPut:DATA:AOFFset?
< 1
Compatibility  
Compatible with MU181020A/B.

:OUTPut:DATA:OFFSet <port>,<numeric>
Parameter  
<port> = <CHARACTER PROGRAM DATA>
  DATA Data output
  XDAta XData output
<numeric> = <DECIMAL NUMERIC PROGRAM DATA>
  Voh –2.000 to 3.300 –2.000 to 3.300 Voh/0.001 V Step
  Vth –3.000 to 3.050 –3.000 to 3.050 Vth/0.001 V Step
  Vol –4.000 to 2.800 –4.000 to 2.800 Vol/0.001 V Step
Function  
Sets the data output offset. The setting range varies depending on the offset reference setting.
Example  
To set the XData output offset to 1.000 V (Voh):
  > :OUTPut:DATA:OFFSet XDAta,1.000
Compatibility  
Compatible with MU181020A/B.
:OUTPut:DATA:OFFSet? <port>

Parameter   <port> = <CHARACTER PROGRAM DATA>
            DATA, XDAta
Response    <numeric> = <NR2 NUMERIC RESPONSE DATA>
            Voh    –2.000 to 3.300 –2.000 to 3.300 Voh/0.001 V Step
            Vth    –3.000 to 3.050 –3.000 to 3.050 Vth/0.001 V Step
            Vol    –4.000 to 2.800 –4.000 to 2.800 Vol/0.001 V Step
Function    Queries the data output offset.
Example     To query the XData output offset:
            > :OUTPut:DATA:OFFSet? XDAta
            < 1.000
Compatibility Compatible with MU181020A/B.

:OUTPut:DATA:ATTFactor <port>,<numeric>

Parameter   <port> = <CHARACTER PROGRAM DATA>
            DATA   Data output
            XDAta  XData output
            <numeric> = <DECIMAL NUMERIC PROGRAM DATA>
            0 to 40  0 to 40 dB/1 dB Step
Function    Sets the External ATT Factor of the data output.
Example     To set the External ATT Factor of the Data output to 20 dB:
            > :OUTPut:DATA:ATTFactor DATA,20
Compatibility Compatible with MU181020A/B.

:OUTPut:DATA:ATTFactor? <port>

Parameter   <port> = <CHARACTER PROGRAM DATA>
            DATA, XDAta
Response    <numeric> = <NR1 NUMERIC RESPONSE DATA>
            0 to 40  0 to 40 dB/1 dB Step
Function    Queries the External ATT Factor of the Data output.
Example     > :OUTPut:DATA:ATTFactor? DATA
            < 20
Compatibility Compatible with MU181020A/B.
Chapter 7  SCPI Commands

:OUTPut:DATA:CPOint <port>,<numeric>

Parameter
- `<port>` = `<CHARACTER PROGRAM DATA>`
  - DATA: Data output
  - XDAta: XData output
- `<numeric>` = `<DECIMAL NUMERIC PROGRAM DATA>`
  - 20.0 to 80.0: 20 to 80%/0.1% Step

Function
Sets the data output crosspoint.

Example
To set the crosspoint of the XData output to 60%:
> :OUTPut:DATA:CPOint XDAta,60.0

Compatibility
Partially compatible with MU181020A/B.

:OUTPut:DATA:CPOint? <port>

Parameter
- `<port>` = `<CHARACTER PROGRAM DATA>`
  - DATA, XDAta

Response
- `<numeric>` = `<NR2 NUMERIC RESPONSE DATA>`
  - 20.0 to 80.0: 20 to 80%/0.1% Step

Function
Queries the data output crosspoint.

Example
To query the crosspoint of the XData output:
> :OUTPut:DATA:CPOint? XDAta
< 60.0

Compatibility
Compatible with MU181020A/B.

:OUTPut:DATA:HPJitter <numeric>

Parameter
- `<numeric>` = `<DECIMAL NUMERIC PROGRAM DATA>`
  - –20 to 20: –20 to 20/1 Step

Function
Sets the half period jitter.

Example
To set the half period jitter to 10:
> :OUTPut:DATA:HPJitter 10

Compatibility
Incompatible with existing models.

:OUTPut:DATA:HPJitter?

Response
- `<numeric>` = `<NR2 NUMERIC RESPONSE DATA>`
  - –20 to 20

Function
Queries the half period jitter.

Example
> :OUTPut:DATA:HPJitter?
< 10

Compatibility
Incompatible with existing models.
7.11  28G/32G bit/s PPG Commands

:OUTPut:DATA:LIMitter:AMPLitude <numeric>
Parameter  <numeric> = <DECIMAL NUMERIC PROGRAM DATA>
            0.500 to 2.000  0.500 to 2.000 V/0.002 V Step
            (MU183020A-x12/x22, MU183021A-x12)
            0.500 to 3.500  0.500 to 3.500 V/0.002 V Step
            (MU183020A-x13/x23, MU183021A-x13)
Function    Sets the limitation value for the data output amplitude.
Example     To set the limitation value for the data output amplitude to 2.000 V:
            > :OUTPut:DATA:LIMitter:AMPLitude 2.000
Compatibility Partially compatible with MU181020A/B.

:OUTPut:DATA:LIMitter:AMPLitude?
Response   <numeric> = <NR2 NUMERIC RESPONSE DATA>
            0.500 to 2.000  0.500 to 2.000 V
            (MU183020A-x12/x22, MU183021A-x12)
            0.500 to 3.500  0.500 to 3.500 V
            (MU183020A-x13/x23, MU183021A-x13)
Function    Queries the limitation value for the data output amplitude.
Example     > :OUTPut:DATA:LIMitter:AMPLitude?
            < 2.000
Compatibility Compatible with MU181020A/B.

:OUTPut:DATA:LIMitter:OFFSet <max>,<min>
Parameter  <max> = <DECIMAL NUMERIC PROGRAM DATA>
            −2.000 to 3.300 −2.000 to 3.300 Voh/0.001 V Step
            (MU183020A-x12/x13/x22/x23, 
            MU183021A-x/12x13)
            <min> = <DECIMAL NUMERIC PROGRAM DATA>
            −4.000 to 2.800 −4.000 to 2.800 Vol/0.001 V Step
            (MU183020A-x12/x13/x22/x23, 
            MU183021A-x/12x13)
Function    Sets the limitation value for the data output offset.
Example     To set the limitation value for the data output offset to 3.000 Voh max.,
            −2.000 Vol min.:
            > :OUTPut:DATA:LIMitter:OFFSet 3.000,-2.000
Compatibility Compatible with MU181020A/B.
Chapter 7  SCPI Commands

:OUTPut:DATA:LIMitter:OFFSet?

Response  <max> = <NR2 NUMERIC RESPONSE DATA>
-2.000 to 3.300  -2.000 to 3.300 Voh/0.001 V Step
(MU183020A-x12/x13/x22/x23,
MU183021A-x12/x13)

<min> = <NR2 NUMERIC PROGRAM DATA>
-4.000 to 2.800  -4.000 to 2.800 Vol/0.001 V Step
(MU183020A-x12/x13/x22/x23,
MU183021A-x12/x13)

Function  Queries the limitation value for the data output offset.
Example
> :OUTPut:DATA:LIMitter:OFFSet?
< 3.000,-2.000

Compatibility  Compatible with MU181020A/B.

:OUTPut:CHANge:CSTate?

Response  <max> = <NR1 NUMERIC RESPONSE DATA>
0  Completed changing Amplitude and Offset setting of Data/XData.
1  Now changing Amplitude and Offset setting of Data/XData.

Function  Queries the status of data output setting change completion.
Example
> :OUTPut:CHANge:CSTate?
< 1

Compatibility  Compatible with MU181020A/B.
7.11.2.3 Delay Setting Commands

![Diagram](image)

**Table 7.11.2.3-1  Delay Setting Commands**

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1]</td>
<td>Delay (mUI setting)</td>
<td>:OUTPut:DATA:UIPadjust</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:OUTPut:DATA:UIPadjust?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:OUTPut:DATA:PADJust?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:OUTPut:DATA:RELative?</td>
</tr>
<tr>
<td></td>
<td>(Relative value setting)</td>
<td>:OUTPut:DATA:RDELay?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:OUTPut:DATA:JINput?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:OUTPut:DATA:UDELay:OFFSet?</td>
</tr>
</tbody>
</table>

**:OUTPut:DATA:UIPadjust <numeric>**

- **Parameter**: 
  
  
  &lt;numeric&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;  
  
  -1000 to 1000  
  
  -1000 to 1000 mUI/2 mUI Step  
  
  -64000 to 64000  
  
  -64000 to 64000 mUI/2 mUI Step  
  
  (In the case of Combination)

- **Function**: Sets the common phase for the Data and XData outputs in mUI units.

- **Example**: To set the common phase to 500 mUI:
  
  > :OUTPut:DATA:UIPadjust 500

- **Compatibility**: Compatible with MU181020A/B.
### :OUTPut:DATA:UIPadjust?

**Response**

<numeric> = <NR1 NUMERIC RESPONSE DATA>
-1000 to 1000  
-64000 to 64000  

(In the case of Combination)

**Function**

Queries the common phase for the Data and XData outputs in mUI units.

**Example**

To query the common phase in mUI units:

```plaintext
> :OUTPut:DATA:UIPadjust?
< 500
```

**Compatibility**

Compatible with MU181020A/B.

---

### :OUTPut:DATA:PADJust <numeric>

**Parameter**

<numeric> = <DECIMAL NUMERIC PROGRAM DATA>
-416 to 416  
-80 to 80  
-31.14 to 31.14

Converted from the calculated frequency value, based on the setting resolution of mUI.

**Function**

Sets the common phase for the Data and XData outputs in ps units. The specified value may not be exactly set as is, due to the limitation of the setting resolution. In this event, the value is adjusted to the most approximate value and set.

**Example**

To set the common phase to 100 ps:

```plaintext
> :OUTPut:DATA:PADJust 100
```

**Compatibility**

Compatible with MU181020A/B.

---

### :OUTPut:DATA:PADJust?

**Response**

<numeric> = <NR2 NUMERIC RESPONSE DATA>
-416 to 416  
-80 to 80  
-31.14 to 31.14

Converted from the calculated frequency value, based on the setting resolution of mUI.

**Function**

Queries the common phase for the Data and XData outputs in ps units.

**Example**

To query the common phase in ps units:

```plaintext
> :OUTPut:DATA:PADJust?
< 100
```

**Compatibility**

Compatible with MU181020A/B.
**:OUTPut:DATA:RELative <boolean>**

**Parameter**

<boolean> = <BOOLEAN PROGRAM DATA>

- 0 or OFF: Reference OFF
- 1 or ON: Reference ON

**Function**

Sets the data output delay reference ON or OFF.

**Example**

To set the data output delay reference ON:

```
> :OUTPut:DATA:RELative ON
```

**Compatibility**

Compatible with MU181020A/B.

---

**:OUTPut:DATA:RELative?**

**Response**

<numeric> = <NR1 NUMERIC RESPONSE DATA>

- 0: Reference OFF
- 1: Reference ON

**Function**

Queries the data output delay reference setting (ON/OFF).

**Example**

```
> :OUTPut:DATA:RELative?
< 1
```

**Compatibility**

Compatible with MU181020A/B.

---

**:OUTPut:DATA:RDELay <numeric>[,<unit>]**

**Parameter**

<numeric> = <DECIMAL NUMERIC PROGRAM DATA>

- –2000 to 2000: –2000 to 2000 mUI/2 mUI Step
- XXXXX.XX: Converted from the calculated frequency value, based on the setting resolution of mUI.

[<unit>] = <CHARACTER PROGRAM DATA>

- UI: mUI units
- PS: ps units

The ps unit is selected when [<unit>] is omitted.

**Function**

Sets the value and unit of the data output delay reference. Set a value using a difference from the reference value. The specified value may not be exactly set as is when the ps unit is set, due to the limitation of the setting resolution. In this event, the value is adjusted to the most approximate value and set.

**Example**

To set the value for the data output delay reference to –1000 mUI:

```
> :OUTPut:DATA:RDELay –1000,UI
```

**Compatibility**

Compatible with MU181020A/B.
**Chapter 7  SCPI Commands**

**:OUTPut:DATA:RDELay? [<unit>]**

Parameter

- `<unit>` = `<CHARACTER PROGRAM DATA>`
  - UI  mUI units
  - PS  ps units
  - The ps unit is selected when `<unit>` is omitted.

Response

- `<numeric>` = `<NR1 NUMERIC RESPONSE DATA>`
  - –2000 to 2000  –2000 to 2000 mUI/2 mUI Step
  - XXXXX.XX  Converted from the calculated frequency value, based on the setting resolution of mUI.

Function

Queries the value and unit of the data output delay reference.

Example

```
> :OUTPut:DATA:RDELay? UI
< –1000
```

Compatibility

Compatible with MU181020A/B.

**:OUTPut:DATA:PCALibration**

Function

Executes calibration for Data and XData output phases.

Example

```
> :OUTPut:DATA:PCALibration
```

Compatibility

Compatible with MU181020A/B.

**:OUTPut:DATA:JINPut <boolean>**

Parameter

- `<boolean>` = `<BOOLEAN PROGRAM DATA>`
  - 0 or OFF  Jitter input OFF setting
  - 1 or ON  Jitter input ON setting

Function

When a jitter is added to the input signal, the Feedback process for Delay must be set to OFF.

When a jitter is added, set the Jitter Input button to ON, using this command. When a jitter is not added, set the Jitter Input button to OFF.

Example

```
To set the Jitter Input button to ON when a jitter is added:
> :OUTPut:DATA:JINPut ON
```

Compatibility

Compatible with MU181020A/B.

**:OUTPut:DATA:JINPut?**

Response

- `<numeric>` = `<NR1 NUMERIC RESPONSE DATA>`
  - 0  Jitter input OFF setting
  - 1  Jitter input ON setting

Function

Queries the Jitter Input button setting.

Example

```
> :OUTPut:DATA:JINPut?
< 1
```

Compatibility

Compatible with MU181020A/B.
:OUTPut:DATA:UDELay:OFFSet <numeric>

Parameter  

<numeric> = <DECIMAL NUMERIC PROGRAM DATA>

-128000 to 128000  -128 000 to 128 000 mUI/2 mUI Step

Function  
Sets the unit of the Delay offset value.

Example  
To set the Delay offset to 500 mUI:

> :OUTPut:DATA:UDELay:OFFSet 500

Compatibility  
Incompatible with existing models.

:OUTPut:DATA:UDELay:OFFSet?

Response  

<numeric> = <NR2 NUMERIC RESPONSE DATA>

-128000 to 128000  -128 000 to 128 000 mUI/2 mUI Step

Function  
Queries the Delay offset value.

Example  

> :OUTPut:DATA:UDELay:OFFSet?

< 500

Compatibility  
Incompatible with existing models.
Chapter 7  SCPI Commands

7.11.3 Pattern Tab

7.11.3.1 Test Pattern Setting Commands

Table 7.11.3.1-1  Test Pattern Setting Commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
</table>
<pre><code> |               | `:SOURce:PATTern:TYPE?` |
 |               | `:SOURce:PATTern:LOGic?` |
</code></pre>

`:SOURce:PATTern:TYPE <type>`

Parameter  

| <type> = <CHARACTER PROGRAM DATA>  
PRBS          | PRBS pattern  
ZSUBstitution | Zero-substitution pattern  
DATA          | Data pattern  
MIXData       | Mixed Data pattern  

Function  
Sets the type of the test pattern.

Example  
To set the test pattern type to the Mixed Data pattern:

```
> :SOURce:PATTern:TYPE MIXData
```
### :SOURce:PATTern:TYPE?

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;type&gt; = &lt;CHARACTER RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRBS, ZSUB, DATA, MIXD</td>
<td></td>
</tr>
</tbody>
</table>

**Function**
Queries the type of the test pattern.

**Example**

```plaintext
> :SOURce:PATTern:TYPE?
< MIXD
```

**Compatibility**
Compatible with MU181020A/B.

### :SOURce:PATTern:LOGic <logic>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;logic&gt; = &lt;CHARACTER PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>POSitive</td>
<td>Positive logic</td>
</tr>
<tr>
<td>NEGative</td>
<td>Negative logic</td>
</tr>
</tbody>
</table>

**Function**
Sets the logic (positive or negative) of the test pattern.

**Example**

To set the logic of the test pattern to the negative logic (NEGative):

```plaintext
> :SOURce:PATTern:LOGic NEGative
```

**Compatibility**
Compatible with MU181020A/B.

### :SOURce:PATTern:LOGic?

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;logic&gt; = &lt;CHARACTER RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>POS, NEG</td>
<td></td>
</tr>
</tbody>
</table>

**Function**
Queries the logic of the test pattern.

**Example**

```plaintext
> :SOURce:PATTern:LOGic?
< NEG
```

**Compatibility**
Compatible with MU181020A/B.
7.11.3.2 PRBS pattern setting commands

![Figure 7.11.3.2-1 Test Pattern Setting (When PRBS Is Selected)]

Table 7.11.3.2-1 PRBS Pattern Setting Commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>:SOURce:PATTern:PRBS:LENGth?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SOURce:PATTern:PRBS:MRATio?</td>
</tr>
</tbody>
</table>

:SOURce:PATTern:PRBS:LENGth <numeric>

Parameter

- \(<\text{numeric}> = \langle\text{DECIMAL NUMERIC PROGRAM DATA}\rangle\>
- 7: \(2^{7} - 1 (n = 7)\)
- 9: \(2^{9} - 1 (n = 9)\)
- 10: \(2^{10} - 1 (n = 10)\)
- 11: \(2^{11} - 1 (n = 11)\)
- 15: \(2^{15} - 1 (n = 15)\)
- 20: \(2^{20} - 1 (n = 20)\)
- 23: \(2^{23} - 1 (n = 23)\)
- 31: \(2^{31} - 1 (n = 31)\)

Function

Sets the number of stages \((2^n - 1 (n = 7, 9, 10, 11, 15, 20, 23, \text{ or } 31))\) during PRBS pattern generation.

Example

To set PRBS15:

\(> :\text{SOURce}:\text{PATTern}:\text{PRBS}:\text{LENGth} \text{ 15}\)

Compatibility

Compatible with MU181020A/B.
7.11 28G/32G bit/s PPG Commands

:SOURce:PATTern:PRBS:LENGth?

Response  
<numeric> = <NR1 NUMERIC RESPONSE DATA>
7, 9, 10, 11, 15, 20, 23, 31

Function  
Queries the number of stages during PRBS pattern generation.

Example  
> :SOURce:PATTern:PRBS:LENGth?
< 15

Compatibility  
Compatible with MU181020A/B.

:SOURce:PATTern:PRBS:MRATio <mratio>

Parameter  
<mratio> = <CHARACTER PROGRAM DATA>
M1_2 1/2
I1_2 1/2INVT

Function  
Sets the mark ratio during PRBS pattern generation.

Example  
To set the mark ratio to 1/2INVT:
> :SOURce:PATTern:PRBS:MRATio I1_2

Compatibility  
Compatible with MU181020A/B.

:SOURce:PATTern:PRBS:MRATio?

Response  
<mratio> = <CHARACTER RESPONSE DATA>
M1_2, I1_2

Function  
Queries the mark ratio during PRBS pattern generation.

Example  
> :SOURce:PATTern:PRBS:MRATio?
< I1_2

Compatibility  
Compatible with MU181020A/B.

:SOURce:PATTern:PRBS:BSHift?

Response  
<numeric> = <NR1 NUMERIC RESPONSE DATA>
1

Function  
Queries the bit shift number during PRBS pattern generation.

Example  
> :SOURce:PATTern:PRBS:BSHift?
< 1

Compatibility  
Compatible with MU181020A/B.
7.11.3.3 Zero-Substitution pattern setting commands

Table 7.11.3.3-1 Zero-Substitution Pattern Setting Commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>:SOURce:PATTern:ZSUBstitute:LENGth?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SOURce:PATTern:ZSUBstitute:ZLENgh?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SOURce:PATTern:ZSUBstitute:ADDBit?</td>
</tr>
</tbody>
</table>

:SOURce:PATTern:ZSUBstitute:LENGth <length>

Parameter  

- <length> = <DECIMAL NUMERIC PROGRAM DATA>
  - 7  \(2^n (n = 7)\)
  - 9  \(2^n (n = 9)\)
  - 10 \(2^n (n = 10)\)
  - 11 \(2^n (n = 11)\)
  - 15 \(2^n (n = 15)\)
  - 20 \(2^n (n = 20)\)
  - 23 \(2^n (n = 23)\)
  - –7 \(2^n-1 (n = 7)\)
  - –9 \(2^n-1 (n = 9)\)
  - –10 \(2^n-1 (n = 10)\)
  - –11 \(2^n-1 (n = 11)\)
  - –15 \(2^n-1 (n = 15)\)
  - –20 \(2^n-1 (n = 20)\)
  - –23 \(2^n-1 (n = 23)\)

Function
Sets the pattern length during Zero-Substitution pattern generation.

Example
To set the pattern length to 2^{15}:

> :SOURce:PATTern:ZSUBstitute:LENGth 15

Compatibility
Compatible with MU181020A/B.
7.11  28G/32G bit/s PPG Commands

:SOURce:PATTern:ZSUBstitute:LENGth?

Response

\(<\text{length}> = <\text{NR1 NUMERIC RESPONSE DATA}>\)

7, 9, 10, 11, 15, 20, 23, –7, –9, –10, –11, –15, –20, –23

Function
Queries the pattern length during Zero-Substitution pattern generation.

Example

> :SOURce:PATTern:ZSUBstitute:LENGth?
< 15

Compatibility
Compatible with MU181020A/B.

:SOURce:PATTern:ZSUBstitute:ZLENgth <numeric>

Parameter

\(<\text{numeric}> = <\text{DECIMAL NUMERIC PROGRAM DATA}>\)

When ZSUBLength, 2^n is set:
1 to 2^n – 1  1 to 2^n – 1/1 bit Step

When ZSUBLength, 2^n – 1 is set:
1 to 2^n – 2  1 to 2^n – 2/1 bit Step

n = 7, 9, 11, 15, 20, 23

Function
Sets the number of bits to which "0" is inserted (zero-insertion bit count) during Zero-Substitution pattern generation. Available parameters vary depending on the setting conditions.

Example

To set the zero-insertion bit count to 10 bits:
> :SOURce:PATTern:ZSUBstitute:ZLENgth 10

Compatibility
Compatible with MU181020A/B.

:SOURce:PATTern:ZSUBstitute:ZLENgth?

Response

\(<\text{numeric}> = <\text{NR1 NUMERIC RESPONSE DATA}>\)

1 to 2^n – 1

n = 7, 9, 11, 15, 20, 23

Function
Queries the zero-insertion bit count during Zero-Substitution pattern generation.

Example

> :SOURce:PATTern:ZSUBstitute:ZLENgth?
< 10

Compatibility
Compatible with MU181020A/B.
### :SOURce:PATTern:ZSUBstitute:ADDBit <numeric>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;numeric&gt;</td>
<td>= &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
</tr>
<tr>
<td>0</td>
<td>The 2ⁿ-th bit is set to “0”.</td>
</tr>
<tr>
<td>1</td>
<td>The 2ⁿ-th bit is set to “1”. (same as existing models).</td>
</tr>
</tbody>
</table>

**Function**
Sets whether to add one bit of “0” or “1” at the end of zeroes inserted successively when Zero-Substitution pattern is generated.

**Example**
To set one bit of “1” next to the last of zero-inserted bits:
> :SOURce:PATTern:ZSUBstitute:ADDBit 1

**Compatibility**
Compatible with MU181020A/B.

### :SOURce:PA**TTern:ZSUBstitute:ADDBit?**

<table>
<thead>
<tr>
<th>Response</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;numeric&gt;</td>
<td>= &lt;NR1 NUMERIC RESPONSE DATA&gt;</td>
</tr>
<tr>
<td>0</td>
<td>The 2ⁿ-th bit is set to “0”.</td>
</tr>
<tr>
<td>1</td>
<td>The 2ⁿ-th bit is set to “1”. (same as existing models).</td>
</tr>
</tbody>
</table>

**Function**
Queries whether to add one bit of “0” or “1” next to the last of zero-inserted bits.

**Example**
> :SOURce:PATTern:ZSUBstitute:ADDBit?
< 1

**Compatibility**
Compatible with MU181020A/B.
7.11.3.4 Data pattern setting commands

![Figure 7.11.3.4-1 Test Pattern Setting (When Data Is Selected)](image)

![Figure 7.11.3.4-2 Data Pattern Setting (Pattern Editor)](image)

### Table 7.11.3.4-1 Data Pattern Setting Commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Data Length</td>
<td>:SOURce:PATTer:n:DATA:LENGth</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SOURce:PATTer:n:DATA:LENGth?</td>
</tr>
<tr>
<td>2</td>
<td>No label (Pattern setting)</td>
<td>:SOURce:PATTer:n:DATA:WHOlE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SOURce:PATTer:n:DATA:WHOlE?</td>
</tr>
<tr>
<td>3</td>
<td>No label (Pattern setting)</td>
<td>:SOURce:PATTer:n:BDATA:WHOlE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SOURce:PATTer:n:BDATA:WHOlE?</td>
</tr>
<tr>
<td>4</td>
<td>No label (Data inversion)</td>
<td>:SOURce:PATTer:n:DREVerse:ADDRess</td>
</tr>
<tr>
<td>5</td>
<td>No label (Data inversion)</td>
<td>SOURce:PATTer:n:DREVerse:DELTa</td>
</tr>
<tr>
<td>6</td>
<td>No label (All 0 or All 1)</td>
<td>SOURce:PATTer:n:DATA:FILL</td>
</tr>
</tbody>
</table>
**Chapter 7  SCPI Commands**

**:SOURce:PATTern:DATA:LENGth <numeric>**

**Parameter**

<table>
<thead>
<tr>
<th></th>
<th>&lt;numeric&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Independent</td>
<td>2 to 268435456</td>
</tr>
<tr>
<td>2 ch</td>
<td>4 to 536870912</td>
</tr>
<tr>
<td>Combination</td>
<td>8 to 1073741824</td>
</tr>
<tr>
<td>4 ch</td>
<td>8 to 1073741824</td>
</tr>
<tr>
<td>Combination</td>
<td>8 to 1073741824</td>
</tr>
</tbody>
</table>

**Function**
Sets the pattern length during Data pattern generation.

**Example**
To set the pattern length to 100 bits:

> :SOURce:PATTern:DATA:LENGth 100

**Compatibility**
Partially compatible with MU181020A/B (the parameter range varies depending on the settings).

**:SOURce:PATTern:DATA:LENGth?**

**Response**

<numeric> = <NR1 NUMERIC RESPONSE DATA>

**Function**
Queries the pattern length during Data pattern generation.

**Example**

> :SOURce:PATTern:DATA:LENGth?  
< 100

**Compatibility**
Compatible with MU181020A/B.
### :SOURce:PATTern:DATA:WHOLe <start>,<end>,<data>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;start&gt; = &lt;NON-DECIMAL PROGRAM DATA&gt;</td>
<td>#H0 #HFFFFFFF 0 to FFFFFFFF bits/1 bit Step (Specify in hexadecimal)</td>
</tr>
<tr>
<td>&lt;end&gt; = &lt;NON-DECIMAL PROGRAM DATA&gt;</td>
<td>#H0 to #HFFFFFFF 0 to FFFFFFFF bits/1 bit Step (Specify in hexadecimal)</td>
</tr>
</tbody>
</table>

**Note:**

The maximum setting is the pattern length. At 2 ch Combination, the setting range and Step are doubled. At 4 ch Combination, the setting range and Step are quadrupled.

**<data> = <STRING PROGRAM DATA>**

- "H****" Specify pattern data in hexadecimal format. 1 to 400 characters (pattern data of 400 × 4 bits) Specify a character string consisting of 0 to 9 and A to F. If the last character string of the pattern data is short of a hexadecimal unit (4 bits), add 0 (s) up to the last bit to make a hexadecimal string.

- "B****" Specify pattern data in binary format. 1 to 400 characters (pattern data of 400 bits) Specify a character string consisting of 0 and 1.

**Function**

Sets the pattern data from the <start> to <end> addresses during Data pattern generation. The set data overwrites the specified range.

**Example**

To set the addresses 0 to 1F of the pattern data to 5A:

> :SOURce:PATTern:DATA:WHOLe #H0,#H1F, "H5A"

**Compatibility**

Partially compatible with MU181020A/B (the parameter range varies depending on the settings).

**Note:**

When the number of data bits specified in <data> is smaller than the range specified by <start> and <end>, set <data> repeatedly up to the specified range.

**Example**

- <start> = #H0, <end> = #H1F, <data> = "HABC"  
  Setting data: ABCABCAB
- <start> = #H0, <end> = #H7, <data> = "B011"  
  Setting data: 01101101

When the number of data bits specified in <data> is greater than the range specified by <start> and <end>, cut off the part of <data> out of the specified range.

**Example**

- <start> = #H0, <end> = #HF, <data> = #HABCDEF
### Setting data: ABCD

- `<start> = #H0, <end> = #H3, <data> = #B01100110`
- Setting data: 0110

### :SOURce:PATTern:DATA:WHOLe? `<start>`

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;start&gt;</code> = <code>&lt;NON-DECIMAL PROGRAM DATA&gt;</code></td>
<td>#H0 to #HFFFFFFF 0 to FFFFFFFF bits/1 bit Step (Specify in hexadecimal)</td>
</tr>
</tbody>
</table>

**Note:**
- The maximum setting is the pattern length.
- At 2 ch Combination, the setting range is doubled.
- At 4 ch Combination, the setting range is quadrupled.

<table>
<thead>
<tr>
<th>Response</th>
<th><code>&lt;data&gt; = </code>&lt;STRING RESPONSE DATA&gt;`</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;H***&quot;</td>
<td>Returns the pattern data in hexadecimal format. Within 400 characters (pattern data of 400 × 4 bits)</td>
</tr>
</tbody>
</table>

**Function:** Queries the pattern data of 400 × 4 bits from the `<start>` address during Data pattern generation.

**Example:**

To query the pattern data from address 0:

```plaintext
> :SOURce:PATTern:DATA:WHOLe? #H0
< "H5A5A5A5A5A0000000"
```

**Compatibility:** Partially compatible with MU181020A/B.
:SOURCE:PATTERN:BDATA:WHOLE <start>,<end>,<bdata>

**Parameter**

<start> = <NON-DECIMAL PROGRAM DATA>

#H0 to #HFFFFFFF 0 to FFFFFFFF bits/1 bit Step

(Specify in hexadecimal)

<end> = <NON-DECIMAL PROGRAM DATA>

#H0 to #HFFFFFFF 0 to FFFFFFFF bits/1 bit Step

(Specify in hexadecimal)

**Note:**

The maximum setting is the pattern length.

At 2 ch Combination, the setting range is doubled.

At 4 ch Combination, the setting range is quadrupled.

<bdata> = <ARBITRARY BLOCK PROGRAM DATA>

#XYYY<binary>

X: Number of YYY digits

YYY: Number of bytes of <binary>

1 to 16,777,216 bytes

<binary>: Binary data up to 16,777,216 bytes

**Function**

Sets binary data of the pattern data from <start> to <end> addresses
during Data pattern generation. The set data overwrites the specified
range.

**Example**

To set the addresses 0 to 1F of the pattern data to 41:

> :SOURCE:PATTERN:BDATA:WHOLE #H0,#H1F,#11A (A=41)

**Compatibility**

Partially compatible with MU181020A/B (the parameter range varies
depending on the settings).

**Note:**

When the number of data bits specified in <bdata> is smaller
than the range specified by <start> and <end>, set <data>
repeatedly up to the specified range.

**<Example>**

- <start> = #H0, <end> = #H1F, <bdata> = #1201 (0 = 30, 1 = 31)
  Setting data: 30 31 30 31...

When the number of data bits specified in <data> is greater than
the range specified by <start> and <end>, cut off the part of <data>
out of the specified range.

**<Example>**

- <start> = #H0, <end> = #HF, <data> = #2100123456789
  Setting data: 30 (Values between 31 and 39 are rounded down to
  30.)
**Chapter 7  SCPI Commands**

### :SOURce:PATTern:BDATa:WHOLe? <start>[,<size>]

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;start&gt; = &lt;NON-DECIMAL PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>#H0 to #HFFFFFFF  0 to FFFFFFFF bits/1 bit Step</td>
</tr>
<tr>
<td><strong>Note:</strong></td>
<td>The maximum setting is the pattern length.</td>
</tr>
<tr>
<td></td>
<td>At 2 ch Combination, the setting range is doubled.</td>
</tr>
<tr>
<td></td>
<td>At 4 ch Combination, the setting range is quadrupled.</td>
</tr>
<tr>
<td></td>
<td>[&lt;size&gt;] = &lt;NR1 NUMERIC PROGRAM DATA&gt;</td>
</tr>
<tr>
<td></td>
<td>1 to 16777216  1 to 16,777,216 bytes/1 byte Step</td>
</tr>
<tr>
<td>Response</td>
<td>Binary data of the setting pattern is queried when [&lt;size&gt;] is omitted.</td>
</tr>
<tr>
<td>&lt;bdata&gt;</td>
<td>= &lt;DEFINITE LENGTH ARBITRARY BLOCK RESPONSE DATA&gt;</td>
</tr>
<tr>
<td></td>
<td>#XYYY&lt;binary&gt;  X: Number of YYY digits</td>
</tr>
<tr>
<td></td>
<td>YYY: Number of bytes of &lt;binary&gt;</td>
</tr>
<tr>
<td></td>
<td>1 to 16,777,216 bytes</td>
</tr>
<tr>
<td></td>
<td>&lt;binary&gt;:Binary data up to 16,777,216 bytes</td>
</tr>
<tr>
<td>Function</td>
<td>Queries binary data of bytes specified by &lt;size&gt; from &lt;start&gt; address of the pattern data during Data pattern generation.</td>
</tr>
<tr>
<td>Example</td>
<td>To query binary data of the setting pattern from the &lt;start&gt; address of the pattern data during Data pattern generation:</td>
</tr>
<tr>
<td></td>
<td>&gt; :SOURce:PATTern:BDATa:WHOLe? #H0</td>
</tr>
<tr>
<td></td>
<td>&lt; #213AA...</td>
</tr>
<tr>
<td></td>
<td>To query binary data of 2 bytes from the &lt;start&gt; address of the pattern data during Data pattern generation:</td>
</tr>
<tr>
<td></td>
<td>&gt; :SOURce:PATTern:BDATa:WHOLe? #H0,2</td>
</tr>
<tr>
<td></td>
<td>&lt; #12AA</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Partially compatible with MU181020A/B.</td>
</tr>
</tbody>
</table>
7.11 28G/32G bit/s PPG Commands

:SOURce:PATTern:DREVerse:ADDRess <start>,<end>

Parameter

<table>
<thead>
<tr>
<th>Parameter Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;start&gt;</td>
<td>= &lt;NON-DECIMAL PROGRAM DATA&gt;</td>
</tr>
<tr>
<td>#H0 to #HFFFFFFF</td>
<td>0 to FFFFFFFF bits/1 bit Step</td>
</tr>
<tr>
<td></td>
<td>(Specify in hexadecimal)</td>
</tr>
<tr>
<td>&lt;end&gt;</td>
<td>= &lt;NON-DECIMAL PROGRAM DATA&gt;</td>
</tr>
<tr>
<td>#H0 to #HFFFFFFF</td>
<td>0 to FFFFFFFF bits/1 bit Step</td>
</tr>
<tr>
<td></td>
<td>(Specify in hexadecimal)</td>
</tr>
</tbody>
</table>

Note:
- The maximum setting is the pattern length.
- At 2 ch Combination, the setting range is doubled.
- At 4 ch Combination, the setting range is quadrupled.

Function
Inverts the data in the program during Data pattern generation. Specify the inversion range by the <start> and <end> addresses.

Example
To invert addresses 0 to 4F of the pattern data:
> :SOURce:PATTern:DREVerse:ADDRess #H0,#H4F

Compatibility
Partially compatible with MU181020A/B (the parameter range varies depending on the settings).

:SOURce:PATTern:DREVerse:DELTa <start>,<delta>

Parameter

<table>
<thead>
<tr>
<th>Parameter Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;start&gt;</td>
<td>= &lt;NON-DECIMAL PROGRAM DATA&gt;</td>
</tr>
<tr>
<td>#H0 to #HFFFFFFF</td>
<td>0 to FFFFFFFF bits/1 bit Step</td>
</tr>
<tr>
<td></td>
<td>(Specify in hexadecimal)</td>
</tr>
<tr>
<td>&lt;delta&gt;</td>
<td>= &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
</tr>
<tr>
<td>1 to 268435456</td>
<td>1 to 268,435,456 bits/1 bit Step</td>
</tr>
</tbody>
</table>

Note:
- The maximum setting is the pattern length.
- At 2 ch Combination, the setting range is doubled.
- At 4 ch Combination, the setting range is quadrupled.

Function
Inverts the data in the program during Data pattern generation. Specify the inversion range by the relative bit number from the <start> address.

Example
To invert addresses 0 to 128 bits of the pattern data:
> :SOURce:PATTern:DREVerse:DELTa #H0,128

Compatibility
Partially compatible with MU181020A/B (the parameter range varies depending on the settings).
Chapter 7  SCPI Commands

:SOURce:PATTern:DATA:FILL <range>,<page>,<data>

Parameter

<range> = <CHARACTER PROGRAM DATA>

PAGE Specifies a page.
(One page is defined as 128 bits.)

ALL Specifies all data.

<page> = <DECIMAL NUMERIC PROGRAM DATA>

1 to 1048576 1 to 1,048,576 (Max.) pages, in 1-page steps

Max = \( \frac{\text{Pattern Length}}{128} \) (rounding up fractions)

Specify “0” when <range> is set to ALL.

<data> = <DECIMAL NUMERIC PROGRAM DATA>

0 Fills the specified range with 0s.

1 Fills the specified range with 1s.

Function

Sets 0s or 1s for the specified page or all data in the program during Data pattern generation.

Example

To set 1s for the second page (from 129th to 256th bits) of the pattern data:

> :SOURce:PATTern:DATA:FILL PAGE, 2, 1

Compatibility

Partially compatible with MU181020A/B (the parameter range varies depending on the settings).
7.11.3.5 Mixed Data pattern setting commands

Figure 7.11.3.5-1 Test Pattern Setting (When Mixed Is Selected)

Figure 7.11.3.5-2 Scramble Setup Dialog Box
Table 7.11.3.5-1  Mixed Data Mixed Data Pattern Setting Commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>[8]</td>
<td>Number of Block</td>
<td>:SOURce:PATTern:MIXData:BLOCk</td>
</tr>
<tr>
<td>[10]</td>
<td>Number of Row</td>
<td>:SOURce:PATTern:MIXData:ROW</td>
</tr>
</tbody>
</table>
Table 7.11.3.5-1 Mixed Data Mixed Data Pattern Setting Commands (Cont’d)

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>[16]</td>
<td>No label (All 0 or All 1)</td>
<td>:SOURce:PATTern:MIXData:DATA:FILL</td>
</tr>
</tbody>
</table>

:SOURce:PATTern:MIXData:PRBS:BITShift?

Response  
<numeric> = <NR1 NUMERIC RESPONSE DATA>
1
1 bit
Function
Sets the bit shift for the PRBS pattern during Mixed Data pattern generation.
Example
> :SOURce:PATTern:MIXData:PRBS:BITShift?
< 1
Compatibility
Partially compatible with MU181020A/B.

:SOURce:PATTern:MIXData:PRBS:LENGth <numeric>

Parameter  
<numeric> = <DECIMAL NUMERIC PROGRAM DATA>
7  $2^n-1$ (n = 7)
9  $2^n-1$ (n = 9)
10  $2^n-1$ (n = 10)
11  $2^n-1$ (n = 11)
15  $2^n-1$ (n = 15)
20  $2^n-1$ (n = 20)
23  $2^n-1$ (n = 23)
31  $2^n-1$ (n = 31)
Function
Sets the number of stages of the PRBS pattern during Mixed Data pattern generation.
Example
To set the number of stages of the PRBS pattern to 15:
> :SOURce:PATTern:MIXData:PRBS:LENGth 15
Compatibility
Partially compatible with MU181020A/B.

:SOURce:PATTern:MIXData:PRBS:LENGth?

Response  
<numeric> = <NR1 NUMERIC RESPONSE DATA>
7, 9, 10, 11, 15, 20, 23, 31
Function
Queries the number of stages of the PRBS pattern during Mixed Data pattern generation.
Example
> :SOURce:PATTern:MIXData:PRBS:LENGth?
< 15
Compatibility
Partially compatible with MU181020A/B.
Chapter 7  SCPI Commands

:SOURce:PATTern:MIXData:PRBS:MRATio <mratio>

Parameter  
<mratio> = <CHARACTER PROGRAM DATA>
M1_2  1/2
I1_2  1/2INVT

Function  
Sets the mark ratio for the PRBS pattern during Mixed Data pattern generation.

Example  
To set the mark ratio for the PRBS pattern to 1/2:
> :SOURce:PATTern:MIXData:PRBS:MRATio M1_2

Compatibility  
Partially compatible with MU181020A/B.

:SOURce:PATTern:MIXData:PRBS:MRATio?

Response  
<mratio> = <CHARACTER RESPONSE DATA>
M1_2, I1_2

Function  
Queries the mark ratio for the PRBS pattern during Mixed Data pattern generation.

Example  
> :SOURce:PATTern:MIXData:PRBS:MRATio?
< M1_2

Compatibility  
Partially compatible with MU181020A/B.

:SOURce:PATTern:MIXData:SCRamble <boolean>

Parameter  
<boolean> = <BOOLEAN PROGRAM DATA>
0 or OFF  Scramble OFF
1 or ON  Scramble ON

Function  
Sets scramble ON/OFF of the PRBS7 stage during pattern reception.

Example  
To set scramble ON of the PRBS7 stage during pattern reception:
> :SOURce:PATTern:MIXData:SCRamble 1

Compatibility  
Partially compatible with MU181020A/B.

:SOURce:PATTern:MIXData:SCRamble?

Response  
<numerical> = <NR1 NUMERIC RESPONSE DATA>
0  Scramble OFF
1  Scramble ON

Function  
Queries the scramble ON/OFF state of the PRBS7 stage during pattern reception.

Example  
> :SOURce:PATTern:MIXData:SCRamble?
< 1

Compatibility  
Partially compatible with MU181020A/B.
7.11 28G/32G bit/s PPG Commands

:SOURce:PATTern:MIXData:PRBS:SEQuence <sequence>

Parameter

<sequence> = <CHARACTER PROGRAM DATA>
RESTart  PRBS patterns are not continuous.
CONSecutive  PRBS patterns are continuous.

Function
Sets the PRBS pattern sequence during Mixed Data pattern generation.

Example
To set the PRBS pattern sequence to Restart:
> :SOURce:PATTern:MIXData:PRBS:SEQuence RESTart

Compatibility
Partially compatible with MU181020A/B.

:SOURce:PATTern:MIXData:PRBS:SEQuence?

Response
<sequence> = <CHARACTER RESPONSE DATA>
REST, CONS

Function
Queries the PRBS pattern sequence during Mixed Data pattern generation.

Example
> :SOURce:PATTern:MIXData:PRBS:SEQuence?
< REST

Compatibility
Partially compatible with MU181020A/B.

:SOURce:PATTern:MIXData:SRSetting <row>,<data>,<prbs>

Parameter

<row> = <DECIMAL NUMERIC PROGRAM DATA>
1 to 16  1 to 16 Row/1 Row Step
<data> = <BOOLEAN PROGRAM DATA>
0 or OFF  Scramble OFF
1 or ON  Scramble ON

Note, however, that Data of 1 Row Scramble is fixed to Scramble OFF.

<prbs> = <BOOLEAN PROGRAM DATA>
0 or OFF  Scramble OFF
1 or ON  Scramble ON

Function
Sets Scramble ON or OFF for the Data and PRBS of the specified Row.

Example
To set Scramble ON for the Data and OFF for the PRBS.
> :SOURce:PATTern:MIXData:SRSetting 2,1,0

Compatibility
Partially compatible with MU181020A/B.
**Chapter 7  SCPI Commands**

:SOURce:PATTern:MIXData:SRSetting? <row>

Parameter  
<row> = <DECIMAL NUMERIC PROGRAM DATA>
1 to 16  1 to 16 Row/1 Row Step

Response  
<data>,<prbs> = <NR1 NUMERIC RESPONSE DATA>
0  Scramble OFF
1  Scramble ON

Function  
Queries the scramble setting (ON/OFF) for the Data and PRBS of the specified Row.

Example  
To query the Scramble setting for 2 Row:
> :SOURce:PATTern:MIXData:SRSetting? 2
< 1,0

Compatibility  
Partially compatible with MU181020A/B.

:SOURce:PATTern:MIXData:SCRamble:ALLSet <numeric>

Parameter  
<numeric> = <NUMERIC PROGRAM DATA>
0  All Reset
1  All Set

Function  
Sets or resets all scramble.

Example  
To select all scramble:
> :SOURce:PATTern:MIXData:SCRamble:ALLSet 1

Compatibility  
Partially compatible with MU181020A/B.

:SOURce:PATTern:MIXData:BLOCk <numeric>

Parameter  
<numeric> = <DECIMAL NUMERIC PROGRAM DATA>
1 to 511  1 to 511 Block/1 Step

Function  
Sets the number of blocks during Mixed Data pattern generation.

Example  
To set the number of blocks to 30:
> :SOURce:PATTern:MIXData:BLOCk 30

Compatibility  
Partially compatible with MU181020A/B.

:SOURce:PATTern:MIXData:BLOCk?

Response  
<numeric> = <NR1 NUMERIC RESPONSE DATA>
1 to 511  1 to 511 Block/1 Step

Function  
Queries the number of blocks during Mixed Data pattern generation.

Example  
> :SOURce:PATTern:MIXData:BLOCk?
< 30

Compatibility  
Partially compatible with MU181020A/B.
### :SOURce:PATTern:MIXData:RLENgth <numeric>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;numeric&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Independent</td>
<td>1536 to 2415919104 1,536 to 2,415,919,104 bits/256 bit Step</td>
</tr>
<tr>
<td>2 ch Combination</td>
<td>3072 to 4831838208 3,072 to 4,831,838,208 bits/512 bit Step</td>
</tr>
<tr>
<td>4 ch Combination</td>
<td>6144 to 9663676416 6,144 to 9,663,676,416 bits/1,024 bit Step</td>
</tr>
</tbody>
</table>

**Function**
Sets the pattern length of 1 Row that is edited during Mixed Data pattern generation.

**Example**
To set the pattern length to 1536 bits:
> :SOURce:PATTern:MIXData:RLENgth 1536

**Compatibility**
Partially compatible with MU181020A/B (the parameter range varies depending on the settings).

### :SOURce:PATTern:MIXData:RLENgth?

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;numeric&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Independent</td>
<td>1536 to 2415919104 1,536 to 2,415,919,104 bits/256 bit Step</td>
</tr>
<tr>
<td>2 ch Combination</td>
<td>3072 to 4831838208 3,072 to 4,831,838,208 bits/512 bit Step</td>
</tr>
<tr>
<td>4 ch Combination</td>
<td>6144 to 9663676416 6,144 to 9,663,676,416 bits/1,024 bit Step</td>
</tr>
</tbody>
</table>

**Function**
Queries the pattern length of 1 Row that is edited during Mixed Data pattern generation.

**Example**
> :SOURce:PATTern:MIXData:RLENgth?
< 1536

**Compatibility**
Partially compatible with MU181020A/B.

### :SOURce:PATTern:MIXData:ROW <numeric>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;numeric&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 to 16</td>
<td>1 to 16 Rows, in 1-row steps</td>
</tr>
</tbody>
</table>

**Function**
Sets the number of rows within one block that are edited during Mixed Data pattern generation.

**Example**
To set the number of rows within one block to 16:
> :SOURce:PATTern:MIXData:ROW 16

**Compatibility**
Partially compatible with MU181020A/B.
Chapter 7 SCPI Commands

:SOURce:PATTERn:MIXData:ROW?

Response  

\(<\text{numeric}> = \text{<NR1 NUMERIC RESPONSE DATA>}\)

Function  

Queries the number of rows within one block that are edited during Mixed Data pattern generation.

Example  

\(> :\text{SOURce}:\text{PATTERn}:\text{MIXData}:\text{ROW}?\)

\(<16\)

Compatibility  

Partially compatible with MU181020A/B.

:SOURce:PATTERn:MIXData:DATA:LENGth <numeric>

Parameter  

\(<\text{numeric}> = \text{<DECIMAL NUMERIC PROGRAM DATA>}\)

Independent  

1024 to 268435456  
1,024 to 268,435,456 bits/
1 bit Step

2 ch  
2048 to 536870912  
2,048 to 536,870,912 bits/
2 bit Step

4 ch  
4096 to 1073741824  
4,096 to 1,073,741,824 bits/
4 bit Step

Combination  

2048 to 536870912  
2,048 to 536,870,912 bits/
2 bit Step

4096 to 1073741824  
4,096 to 1,073,741,824 bits/
4 bit Step

Function  

Sets the pattern length on the pattern data side that is edited during Mixed Data pattern generation.

Example  

To set the pattern length to 1024 bits

\(> :\text{SOURce}:\text{PATTERn}:\text{MIXData}:\text{DATA}:\text{LENGth} 1024\)

Compatibility  

Partially compatible with MU181020A/B (the parameter range varies depending on the settings).

:SOURce:PATTERn:MIXData:DATA:LENGth?

Response  

\(<\text{numeric}> = \text{<NR1 NUMERIC RESPONSE DATA>}\)

Function  

Queries the pattern length on the pattern data side that is edited during Mixed Data pattern generation.

Example  

\(> :\text{SOURce}:\text{PATTERn}:\text{MIXData}:\text{DATA}:\text{LENGth}?\)

\(<1024\)

Compatibility  

Partially compatible with MU181020A/B.
### :SOURce:PATTern:MIXData:DATA:WHOLe \(<block>,<start>,<end>,<data>\)

**Parameter**

- \(<block>\) = \(<\text{DECIMAL NUMERIC PROGRAM DATA}\)  
  1 to 511  
  1 to 511 Block/1 Step

- \(<\text{start}>\) = \(<\text{NON-DECIMAL PROGRAM DATA}\)  
  \#H0 to \#HFFFFFFF  
  0 to FFFFFFFF bits/1 bit Step  
  (Specify in hexadecimal)

- \(<\text{end}>\) = \(<\text{NON-DECIMAL PROGRAM DATA}\)  
  \#H0 to \#HFFFFFFF  
  0 to FFFFFFFF bits/1 bit Step  
  (Specify in hexadecimal)

**Note:**

- The maximum setting is the pattern length.
- At 2 ch Combination, the setting range is doubled.
- At 4 ch Combination, the setting range is quadrupled.

- \(<\text{data}>\) = \(<\text{STRING PROGRAM DATA}\)  
  "H***"  
  Specify pattern data in hexadecimal format.  
  1 to 400 characters (pattern data of 400 (4 bits))  
  Specify a character string consisting of 0 to 9 and A to F.  
  If the last character string of the pattern data is short of a hexadecimal unit (4 bits), add 0(s) up to the last bit to make a hexadecimal string.

  "B***"  
  Specify pattern data in binary format.  
  1 to 400 characters (pattern data of 400 bits)  
  Specify a character string consisting of 0 and 1.

**Function**

Sets the pattern data from the \(<\text{start}>\) to \(<\text{end}>\) addresses during Mixed Data pattern generation. The set data overwrites the specified range.

**Example**

To set the addresses 0 to 1F of the block 1 pattern data to 11:

> :SOURce:PATTern:MIXData:DATA:WHOLe 1,\#H0,\#H1F,"H11"

**Compatibility**

Partially compatible with MU181020A/B (the parameter range varies depending on the settings).
### :SOURce:PATTern:MIXData:DATA:WHOLe? <block>,<start>

**Parameter**
- `<block>` = `<DECIMAL NUMERIC PROGRAM DATA>`
  - 1 to 511 Block/1 Step
- `<start>` = `<NON-DECIMAL PROGRAM DATA>`
  - `#H0` to `#HFFFFFFF` 0 to FFFFFFFF bits/1 bit Step
  - (Specify in hexadecimal)

**Note:**
- The maximum setting is the pattern length.
- At 2 ch Combination, the setting range is doubled.
- At 4 ch Combination, the setting range is quadrupled.

**Response**
- `<data>` = `<STRING RESPONSE DATA>`
  - "H***" Returns the pattern data in hexadecimal format.
  - Within 400 characters (pattern data of 400 \( \times 4 \) bits)

**Function**
Queries the pattern data of 400 \( \times 4 \) bits from the `<start>` address during Mixed Data pattern generation.

**Example**
To query the block 1 pattern data from address 0:
> :SOURce:PATTern:MIXData:DATA:WHOLe? 1,#H0
< "H1111"

**Compatibility**
Partially compatible with MU181020A/B.
**7.11 28G/32G bit/s PPG Commands**

### :SOURce:PATTern:MIXData:BDATa:WHOLe

#### Parameter

- `<block>` = <DECIMAL NUMERIC PROGRAM DATA>
  
  1 to 511  
  1 to 511 Block/1 Step

- `<start>` = <NON-DECIMAL PROGRAM DATA>
  
  #H0 to #HFFFFFFF  
  0 to FFFFFFFF bits/1 bit Step  
  (Specify in hexadecimal)

- `<end>` = <NON-DECIMAL PROGRAM DATA>
  
  #H0 to #HFFFFFFF  
  0 to FFFFFFFF bits/1 bit Step  
  (Specify in hexadecimal)

**Note:**

- The maximum setting is the pattern length.
- At 2 ch Combination, the setting range is doubled.
- At 4 ch Combination, the setting range is quadrupled.

#### <bdata>

- `<bdata>` = <ARBITRARY BLOCK PROGRAM DATA>
  
  #XYY<binary>  
  X: Number of YYY digits  
  YYY: Number of bytes of <binary>

  1 to 16,777,216 (bytes)

  <binary>: Binary data up to 16,777,216 bytes

**Function**

Sets the binary data from `<start>` to `<end>` addresses of the pattern data during Mixed Data pattern generation. The set data overwrites the specified range.

**Example**

To set the addresses 0 to 1F of the block 2 pattern data to 43:

```
> :SOURce:PATTern:MIXData:BDATa:WHOLe 2,#H0,#H1F,#11C  
(C=43)
```

**Compatibility**

Partially compatible with MU181020A/B (the parameter range varies depending on the settings).
**:SOURce:PATTern:MIXData:BDATa:WHOLe? <block>,<start>[,<size>]**

**Parameter**

<table>
<thead>
<tr>
<th>&lt;block&gt;</th>
<th>&lt;DECIMAL NUMERIC PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 to 511</td>
<td>1 to 511 Block/1 Step</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>&lt;start&gt;</th>
<th>&lt;NON-DECIMAL PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>#H0 to #HFFFFFFF</td>
<td>0 to FFFFFFF bits/1 bit Step</td>
</tr>
<tr>
<td></td>
<td>(Specify in hexadecimal)</td>
</tr>
</tbody>
</table>

**Note:**
- The maximum setting is the pattern length.
- At 2 ch Combination, the setting range is doubled.
- At 4 ch Combination, the setting range is quadrupled.

<table>
<thead>
<tr>
<th>&lt;size&gt;</th>
<th>&lt;NR1 NUMERIC PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 to 16777216</td>
<td>1 to 16,777,216 bytes/1 byte Step</td>
</tr>
</tbody>
</table>

**Response**

<table>
<thead>
<tr>
<th>&lt;bdata&gt;</th>
<th>&lt;DEFINITE LENGTH ARBITRARY BLOCK RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>#XYYY&lt;binary&gt;</td>
<td>X: Number of YYY digits</td>
</tr>
<tr>
<td></td>
<td>YYY: Number of bytes of &lt;binary&gt;</td>
</tr>
<tr>
<td></td>
<td>1 to 16,777,216 (bytes)</td>
</tr>
<tr>
<td></td>
<td>&lt;binary&gt;: Binary data up to 16,777,216 bytes</td>
</tr>
</tbody>
</table>

Binary data of the setting pattern is queried when `<size>` is omitted.

**Function**

Queries binary data of bytes specified by `<size>` from the `<start>` address of the pattern data during Mixed Data pattern generation.

**Example**

To query binary data of the block 1 pattern data starting from address 0:

```
> :SOURce:PATTern:MIXData:BDATa:WHOLe? 1,#H0
< #11C
```

**Compatibility**

Partially compatible with MU181020A/B.
### :SOURce:PATTern:MIXData:DREVerse:ADDRess <block>,<start>,<end>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;block&gt;</td>
<td>&lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
</tr>
<tr>
<td></td>
<td>1 to 511 Block/1 Step</td>
</tr>
<tr>
<td>&lt;start&gt;</td>
<td>&lt;NON-DECIMAL PROGRAM DATA&gt;</td>
</tr>
<tr>
<td></td>
<td>#H0 to #HFFFFFFF 0 to FFFFFFF bits/1 bit Step</td>
</tr>
<tr>
<td></td>
<td>(Specify in hexadecimal)</td>
</tr>
<tr>
<td>&lt;end&gt;</td>
<td>&lt;NON-DECIMAL PROGRAM DATA&gt;</td>
</tr>
<tr>
<td></td>
<td>#H0 to #HFFFFFFF 0 to FFFFFFF bits/1 bit Step</td>
</tr>
<tr>
<td></td>
<td>(Specify in hexadecimal)</td>
</tr>
</tbody>
</table>

**Note:**
- The maximum setting is the pattern length.
- At 2 ch Combination, the setting range is doubled.
- At 4 ch Combination, the setting range is quadrupled.

**Function:**
Inverts the data in the program during Mixed Data pattern generation. Specify the inversion range by the <start> and <end> addresses.

**Example:**
To invert addresses 0 to 4F of the block 3 pattern data:

```
> :SOURce:PATTern:MIXData:DREVerse:ADDRess 3,#H0,#H4F
```

**Compatibility:**
Partially compatible with MU181020A/B (the parameter range varies depending on the settings).

### :SOURce:PATTern:MIXData:DREVerse:DELTa <block>,<start>,<delta>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;block&gt;</td>
<td>&lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
</tr>
<tr>
<td></td>
<td>1 to 511 Block/1 Step</td>
</tr>
<tr>
<td>&lt;start&gt;</td>
<td>&lt;NON-DECIMAL PROGRAM DATA&gt;</td>
</tr>
<tr>
<td></td>
<td>#H0 to #HFFFFFFF 0 to FFFFFFF bits/1 bit Step</td>
</tr>
<tr>
<td></td>
<td>(Specify in hexadecimal)</td>
</tr>
<tr>
<td>&lt;delta&gt;</td>
<td>&lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
</tr>
<tr>
<td></td>
<td>1 to 134217728 1 to 134,217,728 bits/1 bit Step</td>
</tr>
</tbody>
</table>

**Note:**
- The maximum setting is the pattern length.
- At 2 ch Combination, the setting range is doubled.
- At 4 ch Combination, the setting range is quadrupled.

**Function:**
Inverts the data in the program during Mixed Data pattern generation. Specify the inversion range by the number of bits from the <start> address (relative specification).

**Example:**
To invert 256 bits of the block 3 pattern data from address 0:

```
> :SOURce:PATTern:MIXData:DREVerse:DELTa 3,#H0,256
```

**Compatibility:**
Partially compatible with MU181020A/B (the parameter range varies depending on the settings).
Chapter 7  SCPI Commands

:SOURce:PATTern:MIXData:DATA:FILL <block>,<range>,<page>,<data>

Parameter

- **<block>** = <DECIMAL NUMERIC PROGRAM DATA>
  1 to 511
  1 to 511 Block/1 Step

- **<range>** = <CHARACTER PROGRAM DATA>
  PAGE
  Specifies a page.
  (One page is defined as 128 bits.)
  ALL
  Specifies all data.

- **<page>** = <DECIMAL NUMERIC PROGRAM DATA>
  1 to 1048576
  1 to 1,048,576 (Max) pages, in 1-page steps

  Max = \[
  \frac{\text{Pattern Length}}{128}
  \] (rounding up fractions)

  Specify “0” when <range> is set to ALL.

- **<data>** = <DECIMAL NUMERIC PROGRAM DATA>
  0
  Fills the specified range with 0s.
  1
  Fills the specified range with 1s.

Function

Sets 0s or 1s for the specified page or all data in the program during Mixed Data pattern generation.

Example

To set 0s for the third page of the block 511 pattern data:

> :SOURce:PATTern:MIXData:DATA:FILL 511,PAGE,3,0

Compatibility

Partially compatible with MU181020A/B (the parameter range varies depending on the settings).
7.11.4 Error Addition Tab

![Error Addition Tab Diagram](image)

Figure 7.11.4-1  Error Addition Settings
(PRBS, Zero-Substitution, and Data Patterns)

![Error Addition Table](image)

Figure 7.11.4-2  Error Addition Settings (Mixed Pattern)
## Table 7.11.4-1  Error Addition Setting Commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>:SOURce-PATTern:EADDition:SET?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SOURce-PATTern:EADDition:SOURce?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SOURce-PATTern:EADDition:SELECT?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SOURce-PATTern:EADDition:ROUTE?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SOURce-PATTern:EADDition:RATE?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SOURce-PATTern:EADDition:VARIation?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SOURce-PATTern:EADDition:AREA?</td>
</tr>
<tr>
<td>[9]</td>
<td>Set All/Reset All</td>
<td>:SOURce-PATTern:EADDition:ALLSet</td>
</tr>
</tbody>
</table>

### :SOURce:PATTern:EADDition:SET <boolean>

**Parameter**  
<boolean> = <BOOLEAN PROGRAM DATA>
- OFF or 0: Output OFF
- ON or 1: Output ON

**Function**  
Sets whether to add a bit error to the test pattern.

**Example**  
To set bit error addition ON:
> :SOURce:PATTern:EADDition:SET ON

**Compatibility**  
Partially compatible with MU181020A/B.

### :SOURce:PATTern:EADDition:SET?

**Response**  
<numeric> = <NR1 NUMERIC RESPONSE DATA>
- 0: Error addition OFF
- 1: Error addition ON

**Function**  
Queries the bit error addition ON/OFF state to the test pattern.

**Example**  
> :SOURce:PATTern:EADDition:SET?
< 1

**Compatibility**  
Partially compatible with MU181020A/B.
**:SOURce:PATTern:EADDition:SOURce <source>**

**Parameter**
- `<source>` = `<CHARACTER PROGRAM DATA>`
- INTernal Internal signal
- EXTTrig Rising of external signal trigger
- EXTDisable Enables external signal

**Function**
Sets the reference signal source for bit error addition to the test pattern.

**Example**
To set the reference signal source for bit error addition to Internal:

```
> :SOURce:PATTern:EADDition:SOURce INTernal
```

**Compatibility**
Partially compatible with MU181020A/B.

**:SOURce:PATTern:EADDition:SOURce?**

**Response**
- `<source>` = `<CHARACTER RESPONSE DATA>`
- INT, EXTT, EXTD

**Function**
Queries the reference signal source for bit error addition to the test pattern.

**Example**
```
> :SOURce:PATTern:EADDition:SOURce?
< INT
```

**Compatibility**
Partially compatible with MU181020A/B.

**:SOURce:PATTern:EADDition:SELect <select>**

**Parameter**
- `<select>` = `<CHARACTER PROGRAM DATA>`
- SCAN Changes the route for which 32 Demux is performed every time when an error is added.
- SELect Adds an error to the specified route.

**Function**
Sets the route setting method for bit error addition.

**Example**
To set the route setting method to SCAN:
```
> :SOURce:PATTern:EADDition:SELect SCAN
```

**Compatibility**
Partially compatible with MU181020A/B.

**:SOURce:PATTern:EADDition:SELect?**

**Response**
- `<select>` = `<CHARACTER RESPONSE DATA>`
- SCAN, SEL

**Function**
Queries the route setting method for bit error addition.

**Example**
```
> :SOURce:PATTern:EADDition:SELect?
< SCAN
```

**Compatibility**
Partially compatible with MU181020A/B.
Chapter 7  SCPI Commands

:SOURce:PATTern:EADDition:ROUTe <numeric>
Parameter  <numeric> = <DECIMAL NUMERIC PROGRAM DATA>
      1 to 32 1 to 32/1 Step
Function Sets the route to add a bit error.
Example  To set Route 2 as the route to add a bit error:
       > :SOURce:PATTern:EADDition:ROUTe 2
Compatibility Partially compatible with MU181020A/B.

:SOURce:PATTern:EADDition:ROUTe?
Response <numeric> = <NR1 NUMERIC RESPONSE DATA>
      1 to 32 1 to 32/1 Step
Function Queries the route to add a bit error.
Example  > :SOURce:PATTern:EADDition:ROUTe?
       < 2
Compatibility Partially compatible with MU181020A/B.

:SOURce:PATTern:EADDition:RATE <rate>[,<numeric>]
Parameter  <rate> = <CHARACTER PROGRAM DATA>
            E_3 to E_12 E–3 to E–12/E–1 Step
            <numeric> = <DECIMAL NUMERIC PROGRAM DATA>
            x:1 to 9 x:1 to 9/1 Step
Set a value in the format of “xE–n”. If x is omitted, 1 is set.
Function Sets the rate of bit errors to be added.
Example  To set the rate of bit errors to be added to 5E–9:
       > :SOURce:PATTern:EADDition:RATE E_9,5
Compatibility Partially compatible with MU181020A/B.

:SOURce:PATTern:EADDition:RATE?
Response <rate> = <CHARACTER RESPONSE DATA>
            E_3 to E_12 E–3 to E–12
            <numeric> = <NR1 NUMERIC RESPONSE DATA>
            x:1 to 9 x:1 to 9/1 Step
Function Queries the rate of bit errors to be added.
Example  > :SOURce:PATTern:EADDition:RATE?
       < E_9,5
Compatibility Partially compatible with MU181020A/B.
7.11 28G/32G bit/s PPG Commands

:SOURce:PATTern:EADDition:VARiation <var>

Parameter
- <var> = <CHARACTER PROGRAM DATA>
- REP (Repeat) Errors are continuously inserted.
- SING (Single) An error is inserted by each click.

Function
Sets the bit error addition method for the test pattern (when Internal is selected).

Example
To set the bit error addition method to Repeat:
> :SOURce:PATTern:EADDition:VARiation REPeat

Compatibility
Partially compatible with MU181020A/B.

:SOURce:PATTern:EADDition:VARiation?

Response
- <var> = <CHARACTER RESPONSE DATA>
- REP, SING

Function
Queries the bit error addition method (when Internal is selected) to the test pattern.

Example
> :SOURce:PATTern:EADDition:VARiation?
< REP

Compatibility
Partially compatible with MU181020A/B.

:SOURce:PATTern:EADDition:SINGle

Function
Adds a single error to the test pattern.

Example
> :SOURce:PATTern:EADDition:SINGle

Compatibility
Partially compatible with MU181020A/B.

:SOURce:PATTern:EADDition:AREA <contents>,<row>,<data>[,<prbs>]

Parameter
- <contents> = <CHARACTER PROGRAM DATA>
- NONE (For Mixed Data and Sequence patterns)
- <row> = <DECIMAL NUMERIC PROGRAM DATA>
  1 to 16 (Specify the row number for Mixed Data patterns)
- <data> = <BOOLEAN PROGRAM DATA>
  OFF or 0, ON or 1
- [<prbs>] = <BOOLEAN PROGRAM DATA>
  OFF or 0, ON or 1 (For Mixed patterns only)

Function
Sets an area to add a bit error for Alternate, Mixed, and Sequence patterns.

Example
To set an area to add a bit error to "Pattern A, 1 Row, Data ON, PRBS ON":
> :SOURce:PATTern:EADDition:AREA NONE,1,1,1

Compatibility
Partially compatible with MU181020A/B.
Chapter 7  SCPI Commands

:SOURce:PATTern:EADDition:AREA? <contents>,<row>

Parameter
<contents> = <CHARACTER PROGRAM DATA>
NONE (For Mixed Data and Sequence patterns)
<row> = <DECIMAL NUMERIC PROGRAM DATA>
1 to 16 (Specify the row number for Mixed Data patterns)

Response
<data>,<prbs> = <NR1 NUMERIC RESPONSE DATA>
0 OFF
1 ON
<brbs> is returned for Mixed patterns only.

Function
Queries an area to add a bit error.

Example
To query an error addition area of Pattern A, 1 Row:
> :SOURce:PATTern:EADDition:AREA? NONE,1
< 1,1

Compatibility
Partially compatible with MU181020A/B.

:SOURce:PATTern:EADDition:ALLSet <numeric>

Parameter
<numeric> = <NUMERIC PROGRAM DATA>
0 All Reset
1 All Set

Function
Selects (set) or unselects (reset) all areas to add a bit error for Mixed and Sequence patterns.

Example
To select all areas to add a bit error:
> :SOURce:PATTern:EADDition:ALLSet 1

Compatibility
Partially compatible with MU181020A/B.
7.11.5 Pre-Code Tab

Figure 7.11.5-1 shows the Pre-Code tab that is added to the MU183020A-x22/x23 and MU183021A PPGs.

![Pre-Code Tab](image)

**Figure 7.11.5-1 Pre-Code Tab**

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

`:SOURce:PRECode:SET <boolean>`

Parameter  
<boolean> = <BOOLEAN PROGRAM DATA>
OFF to 0  
ON or 1  

Function  
Sets the Pre-Code Setting ON or OFF.

Example  
To set the Pre-Code Setting ON:
> :SOURce:PRECode:SET ON

Compatibility  
Compatible with MU181020A/B.

`:SOURce:PRECode:SET?`

Response  
<numeric> = <NR1 NUMERIC RESPONSE DATA>
0  
1  

Function  
Queries the Pre-Code Setting ON or OFF.

Example  
> :SOURce:PRECode:SET?
< 1

Compatibility  
Compatible with MU181020A/B.
Chapter 7  SCPI Commands

:SOURce:PRECode:TYPE <type>

Parameter  
- <type> = <CHARACTER PROGRAM DATA>
  - DQPSk  DQPSK
  - DPQPsK  DPQPSK

Function  
Set the Pre-Code Setting type.

Example  
To set the Pre-Code Setting type to DQPSK:
  > :SOURce:PRECode:TYPE DQPSk

Compatibility  
Partially compatible with MU181020A/B.

:SOURce:PRECode:TYPE?

Response  
- <type> = <CHARACTER RESPONSE DATA>
  - DQPS, DPQP

Function  
Queries the Pre-Code Setting type.

Example  
> :SOURce:PRECode:TYPE?
  < DQPS

Compatibility  
Partially compatible with MU181020A/B.

:SOURce:PRECode:INITialize <numeric>

Parameter  
- <numeric> = <DECIMAL NUMERIC PROGRAM DATA>
  - 0, 1

Function  
Sets Initialize Data of Pre-Code Setting.

Example  
To set Initialize Data of Pre-Code Setting to 1:
  > :SOURce:PRECode:INITialize 1

Compatibility  
Compatible with MU181020A/B.

:SOURce:PRECode:INITialize?

Response  
- <numeric> = <NR1 NUMERIC RESPONSE DATA>
  - 0, 1

Function  
Queries Initialize Data of Pre-Code Setting.

Example  
> :SOURce:PRECode:INITialize?
  < 1

Compatibility  
Compatible with MU181020A/B.
7.11.6 Misc1 Tab

![Misc1 Tab Diagram]

7.11.6.1 Pattern Generation Setting Commands

![Pattern Sequence Setting Diagram]

(When Generating Repeat Signal)
Table 7.11.6.1-1  Pattern Generation Setting Commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Pattern Sequence</td>
<td>::SOURce:PATTern:OMODE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>::SOURce:PATTern:OMODE?</td>
</tr>
<tr>
<td>2</td>
<td>Data Sequence</td>
<td>::SOURce:PATTern:BURSt:SEQUence</td>
</tr>
<tr>
<td></td>
<td></td>
<td>::SOURce:PATTern:BURSt:SEQUence?</td>
</tr>
<tr>
<td>3</td>
<td>Source</td>
<td>::SOURce:PATTern:BURSt:MODE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>::SOURce:PATTern:BURSt:MODE?</td>
</tr>
<tr>
<td>4</td>
<td>Pulse Width</td>
<td>::SOURce:PATTern:REPeat:PULSewidth</td>
</tr>
<tr>
<td></td>
<td></td>
<td>::SOURce:PATTern:REPeat:PULSewidth?</td>
</tr>
<tr>
<td>5</td>
<td>Delay (For Repeat)</td>
<td>::SOURce:PATTern:REPeat:TRIGdelay</td>
</tr>
<tr>
<td></td>
<td></td>
<td>::SOURce:PATTern:REPeat:TRIGdelay?</td>
</tr>
<tr>
<td>6</td>
<td>Enable Period</td>
<td>::SOURce:PATTern:BURSt:ELENgth</td>
</tr>
<tr>
<td></td>
<td></td>
<td>::SOURce:PATTern:BURSt:ELENgth?</td>
</tr>
<tr>
<td>7</td>
<td>Burst Cycle</td>
<td>::SOURce:PATTern:BURSt:CYCle</td>
</tr>
<tr>
<td></td>
<td></td>
<td>::SOURce:PATTern:BURSt:CYCle?</td>
</tr>
<tr>
<td>8</td>
<td>Delay (For Burst)</td>
<td>::SOURce:PATTern:BURSt:TRIGdelay</td>
</tr>
<tr>
<td></td>
<td></td>
<td>::SOURce:PATTern:BURSt:TRIGdelay?</td>
</tr>
<tr>
<td>9</td>
<td>Pulse Width</td>
<td>::SOURce:PATTern:BURSt:PULSewidth</td>
</tr>
<tr>
<td></td>
<td></td>
<td>::SOURce:PATTern:BURSt:PULSewidth?</td>
</tr>
</tbody>
</table>
**:SOURce:PATTern:OMODE <mode>**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;mode&gt;</td>
<td>= &lt;CHARACTER PROGRAM DATA&gt;</td>
</tr>
<tr>
<td>REPeat</td>
<td>Repeat signal</td>
</tr>
<tr>
<td>BURSt</td>
<td>Burst signal</td>
</tr>
</tbody>
</table>

**Function**
Sets the signal generation method from Repeat (consecutive) signal or Burst signal.

**Example**
To set the signal generation method to Burst signal:
```
> :SOURce:PATTern:OMODE BURSt
```

**Compatibility**
Partially compatible with MU181020A/B.

**:SOURce:PATTern:OMODE?**

<table>
<thead>
<tr>
<th>Response</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;mode&gt;</td>
<td>= &lt;CHARACTER RESPONSE DATA&gt;</td>
</tr>
<tr>
<td>REP, BURS</td>
<td></td>
</tr>
</tbody>
</table>

**Function**
Queries the signal generation method.

**Example**
```
> :SOURce:PATTern:OMODE?
< BURS
```

**Compatibility**
Partially compatible with MU181020A/B.

**:SOURce:PATTern:BURSt:SEQuence <mode>**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;mode&gt;</td>
<td>= &lt;CHARACTER PROGRAM DATA&gt;</td>
</tr>
<tr>
<td>RESTart</td>
<td>Restart from the beginning</td>
</tr>
<tr>
<td>CONSecutive</td>
<td>Consecutive output in the Burst interval</td>
</tr>
<tr>
<td>CONTinuous</td>
<td>Consecutive output regardless of Burst interval</td>
</tr>
</tbody>
</table>

**Function**
Sets the data output sequence for the Burst data signal.

**Example**
To set the output sequence to Restart:
```
> :SOURce:PATTern:BURSt:SEQuence RESTart
```

**Compatibility**
Partially compatible with MU181020A/B.

**:SOURce:PATTern:BURSt:SEQuence?**

<table>
<thead>
<tr>
<th>Response</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;mode&gt;</td>
<td>= &lt;CHARACTER RESPONSE DATA&gt;</td>
</tr>
<tr>
<td>REST, CONS, CONT</td>
<td></td>
</tr>
</tbody>
</table>

**Function**
Queries the data output sequence for the Burst data signal.

**Example**
```
> :SOURce:PATTern:BURSt:SEQuence?
< REST
```

**Compatibility**
Partially compatible with MU181020A/B.
### :SOURce:PATTern:BURSt:MODE <mode>

**Parameter**

<table>
<thead>
<tr>
<th>&lt;mode&gt;</th>
<th>Program Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>INTernal</td>
<td>Generates based on internal reference signal</td>
</tr>
<tr>
<td>EXTTrig</td>
<td>Generates based on external signal trigger edge</td>
</tr>
<tr>
<td>EXTernal</td>
<td>Generates during the high level interval of external signal trigger input</td>
</tr>
</tbody>
</table>

**Function**
Sets the Burst signal generation sequence from internal trigger, external trigger, or enable.

**Example**
To set the Burst signal generation sequence to Internal:

```>
:SOURce:PATTern:BURSt:MODE INTernal
```

**Compatibility**
Partially compatible with MU181020A/B.

### :SOURce:PATTern:BURSt:MODE? 

**Response**

```
<mode> = <CHARACTER RESPONSE DATA>
```

| INT, EXTT, EXT |

**Function**
Queries the Burst signal generation sequence.

**Example**

```>
:SOURce:PATTern:BURSt:MODE?
```

```
< INT
```

**Compatibility**
Partially compatible with MU181020A/B.

### :SOURce:PATTern:REPeat:PULSewidth <numeric>

**Parameter**

<table>
<thead>
<tr>
<th>&lt;numeric&gt;</th>
<th>Program Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRBS/ZeroSub/Data</td>
<td>Mixed</td>
</tr>
<tr>
<td>Independent</td>
<td>0 to 34359738240 bits/8 bits Step 0 to 2415918976/8 bits Step</td>
</tr>
<tr>
<td>2 ch</td>
<td>0 to 68719476480/16 bits Step 0 to 4831837952/16 bits Step</td>
</tr>
<tr>
<td>Combination</td>
<td>0 to 137438952960/32 bits Step 0 to 9663675904/32 bits Step</td>
</tr>
</tbody>
</table>

**Function**
Sets the pulse width of the timing signal to be output during the timing signal period.

**Example**
To set the pulse width of the timing signal to 128 bits:

```>
:SOURce:PATTern:REPeat:PULSewidth 128
```

**Compatibility**
Partially compatible with MU181020A/B (the parameter range varies depending on the settings).
**:SOURCE:PATTern:REPeat:PULSewidth?**

**Response**

\(<\text{numeric}> = \langle\text{NR1 NUMERIC RESPONSE DATA}\rangle\)

- 0 to 137438952960 0 to 137,438,952,960 bits

**Function**

Queries the pulse width of the timing signal to be output during the timing signal period.

**Example**

To query the pulse width of the timing signal:

\(> :\text{SOURCE:PATTern:REPeat:PULSewidth?}\)

\(< 128\)

**Compatibility**

Partially compatible with MU181020A/B (the parameter range varies depending on the settings).

**:SOURCE:PATTern:REPeat:TRIGdelay <numeric>**

**Parameter**

\(<\text{numeric}> = \langle\text{DECIMAL NUMERIC PROGRAM DATA}\rangle\)

<table>
<thead>
<tr>
<th>PRBS/ZeroSub/Data</th>
<th>0 to 34359738240 bits/8 bits Step</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRBS/ZeroSub/Data</td>
<td>0 to 2415918976/8 bits Step</td>
</tr>
<tr>
<td>Independent</td>
<td>0 to 68719476480/16 bits Step</td>
</tr>
<tr>
<td>Combination</td>
<td>0 to 137438952960/32 bits Step</td>
</tr>
<tr>
<td>PRBS/ZeroSub/Data</td>
<td>0 to 4831837952/32 bits Step</td>
</tr>
<tr>
<td>Independent</td>
<td>0 to 9663675904/32 bits Step</td>
</tr>
<tr>
<td>Combination</td>
<td>0 to 68719476480/16 bits Step</td>
</tr>
<tr>
<td>PRBS/ZeroSub/Data</td>
<td>0 to 34359738240 bits/8 bits Step</td>
</tr>
<tr>
<td>Independent</td>
<td>0 to 2415918976/8 bits Step</td>
</tr>
<tr>
<td>Combination</td>
<td>0 to 137438952960/32 bits Step</td>
</tr>
<tr>
<td>PRBS/ZeroSub/Data</td>
<td>0 to 4831837952/32 bits Step</td>
</tr>
<tr>
<td>Independent</td>
<td>0 to 9663675904/32 bits Step</td>
</tr>
<tr>
<td>Combination</td>
<td>0 to 68719476480/16 bits Step</td>
</tr>
<tr>
<td>PRBS/ZeroSub/Data</td>
<td>0 to 34359738240 bits/8 bits Step</td>
</tr>
<tr>
<td>Independent</td>
<td>0 to 2415918976/8 bits Step</td>
</tr>
<tr>
<td>Combination</td>
<td>0 to 137438952960/32 bits Step</td>
</tr>
<tr>
<td>PRBS/ZeroSub/Data</td>
<td>0 to 4831837952/32 bits Step</td>
</tr>
<tr>
<td>Independent</td>
<td>0 to 9663675904/32 bits Step</td>
</tr>
<tr>
<td>Combination</td>
<td>0 to 68719476480/16 bits Step</td>
</tr>
<tr>
<td>PRBS/ZeroSub/Data</td>
<td>0 to 34359738240 bits/8 bits Step</td>
</tr>
<tr>
<td>Independent</td>
<td>0 to 2415918976/8 bits Step</td>
</tr>
<tr>
<td>Combination</td>
<td>0 to 137438952960/32 bits Step</td>
</tr>
<tr>
<td>PRBS/ZeroSub/Data</td>
<td>0 to 4831837952/32 bits Step</td>
</tr>
<tr>
<td>Independent</td>
<td>0 to 9663675904/32 bits Step</td>
</tr>
<tr>
<td>Combination</td>
<td>0 to 68719476480/16 bits Step</td>
</tr>
<tr>
<td>PRBS/ZeroSub/Data</td>
<td>0 to 34359738240 bits/8 bits Step</td>
</tr>
<tr>
<td>Independent</td>
<td>0 to 2415918976/8 bits Step</td>
</tr>
<tr>
<td>Combination</td>
<td>0 to 137438952960/32 bits Step</td>
</tr>
<tr>
<td>PRBS/ZeroSub/Data</td>
<td>0 to 4831837952/32 bits Step</td>
</tr>
<tr>
<td>Independent</td>
<td>0 to 9663675904/32 bits Step</td>
</tr>
<tr>
<td>Combination</td>
<td>0 to 68719476480/16 bits Step</td>
</tr>
<tr>
<td>PRBS/ZeroSub/Data</td>
<td>0 to 34359738240 bits/8 bits Step</td>
</tr>
<tr>
<td>Independent</td>
<td>0 to 2415918976/8 bits Step</td>
</tr>
<tr>
<td>Combination</td>
<td>0 to 137438952960/32 bits Step</td>
</tr>
<tr>
<td>PRBS/ZeroSub/Data</td>
<td>0 to 4831837952/32 bits Step</td>
</tr>
<tr>
<td>Independent</td>
<td>0 to 9663675904/32 bits Step</td>
</tr>
<tr>
<td>Combination</td>
<td>0 to 68719476480/16 bits Step</td>
</tr>
<tr>
<td>PRBS/ZeroSub/Data</td>
<td>0 to 34359738240 bits/8 bits Step</td>
</tr>
<tr>
<td>Independent</td>
<td>0 to 2415918976/8 bits Step</td>
</tr>
<tr>
<td>Combination</td>
<td>0 to 137438952960/32 bits Step</td>
</tr>
<tr>
<td>PRBS/ZeroSub/Data</td>
<td>0 to 4831837952/32 bits Step</td>
</tr>
<tr>
<td>Independent</td>
<td>0 to 9663675904/32 bits Step</td>
</tr>
<tr>
<td>Combination</td>
<td>0 to 68719476480/16 bits Step</td>
</tr>
</tbody>
</table>

**Function**

Sets the delay value for the timing signal pulse to be output during the timing signal period.

**Example**

To set the timing signal pulse delay to 256 bits:

\(> :\text{SOURCE:PATTern:REPeat:TRIGdelay} 256\)

**Compatibility**

Partially compatible with MU181020A/B (the parameter range varies depending on the settings).

**:SOURCE:PATTern:REPeat:TRIGdelay?**

**Response**

\(<\text{numeric}> = \langle\text{NR1 NUMERIC RESPONSE DATA}\rangle\)

- 0 to 137438952960 0 to 137,438,952,960 bits

**Function**

Queries the delay value for the timing signal pulse to be output during the timing signal period.

**Example**

To query the delay value for the timing signal:

\(> :\text{SOURCE:PATTern:REPeat:TRIGdelay}?\)

\(< 256\)

**Compatibility**

Partially compatible with MU181020A/B (the parameter range varies depending on the settings).
**Chapter 7  SCPI Commands**

`:SOURce:PATTern:BURSt:ELENgth <numeric>`

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;numeric&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Internal</td>
<td>1024 to 2,147,483,392 bits 1,024 to 2,147,483,648 bits</td>
</tr>
<tr>
<td>External</td>
<td>256 bits Step 256 bits Step</td>
</tr>
<tr>
<td>Independent</td>
<td>2048 to 4,294,966,784 bits 2,048 to 4,294,967,296 bits</td>
</tr>
<tr>
<td>2 ch</td>
<td>512 bits Step 512 bits Step</td>
</tr>
<tr>
<td>Combination</td>
<td>4,096 to 8,589,933,568 bits 4,096 to 8,589,934,592 bits</td>
</tr>
<tr>
<td>4 ch</td>
<td>1024 bits Step 1024 bits Step</td>
</tr>
<tr>
<td>Combination</td>
<td>8,192 to 17,179,869,184 bits 8,192 to 17,179,869,720 bits</td>
</tr>
</tbody>
</table>

Function  
Sets the data signal generation interval for Burst signal generation.

Example  
To set the data signal generation interval to 12,800 bits:
> :SOURce:PATTern:BURSt:ELENgth 12800

Compatibility  
Partially compatible with MU181020A/B (the parameter range varies depending on the settings).

`:SOURce:PATTern:BURSt:ELENgth?`

Response  
<nemonic> = <NR1 NUMERIC RESPONSE DATA>  
1024 to 8,589,933,568 1,024 to 8,589,934,592 bits

Function  
Queries the data signal generation interval for Burst signal generation.

Example  
> :SOURce:PATTern:BURSt:ELENgth?
< 12800

Compatibility  
Partially compatible with MU181020A/B (the parameter range varies depending on the settings).

`:SOURce:PATTern:BURSt:CYClE <numeric>`

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;numeric&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Independent</td>
<td>1,536 to 2,147,483,648 bits/256 bits Step</td>
</tr>
<tr>
<td>2 ch Combination</td>
<td>3,072 to 4,294,967,296 bits/512 bits Step</td>
</tr>
<tr>
<td>4 ch Combination</td>
<td>6,144 to 8,589,934,592 bits/1024 bits Step</td>
</tr>
</tbody>
</table>

Function  
Sets an addition value of the data signal generation interval and the data stop interval for Burst signal generation (Burst generation cycle).

Example  
To set the Burst generation cycle to 25,600 bits:
> :SOURce:PATTern:BURSt:CYclE 25600

Compatibility  
Partially compatible with MU181020A/B (the parameter range varies depending on the settings).
7.11 28G/32G bit/s PPG Commands

:SOURce:PATTern:BURSt:CYCLE?

Response
<numeric> = <NR1 NUMERIC RESPONSE DATA>
1536 to 8589934592  1,536 to 8,589,934,592 bits

Function
Queries an addition value of the data signal generation interval and the data stop interval for Burst signal generation (Burst generation cycle).

Example
> :SOURce:PATTern:BURSt:CYCLE?
< 25600

Compatibility
Partially compatible with MU181020A/B (the parameter range varies depending on the settings).

:SOURce:PATTern:BURSt:TRIGdelay <port>,<numeric>

Parameter
<port> = <CHARACTER PROGRAM DATA>
BURStout1 Burst Output
BURStout2 Burst Output2
Valid only when AUX Output is set to Burst Output2
<numeric> = <DECIMAL NUMERIC PROGRAM DATA>
Independent 0 to 2147483520/ 8 bits Step
0 to 2,147,483,520 bits
2 ch Combination 0 to 4294967040/16 bits Step
0 to 4,294,967,040 bits
4 ch Combination 0 to 8589934080/32 bits Step
0 to 8,589,934,080 bits

Note:
The maximum setting is the pattern length.

Function
Sets the Burst output trigger signal generation timing (delay) to adjust the phase on the DUT side.

Example
To set the Burst output trigger generation timing (delay) to 128 bits:
> :SOURce:PATTern:BURSt:TRIGdelay BURStout1,128

Compatibility
Partially compatible with MU181020A/B (the parameter range varies depending on the settings).

:SOURce:PATTern:BURSt:TRIGdelay? <port>

Parameter
<port> = <CHARACTER PROGRAM DATA>
BURStout1 Burst Output
BURStout2 Burst Output2
Valid only when AUX Output is set to Burst Output2.

Response
<numeric> = <NR1 NUMERIC RESPONSE DATA>
0 to 8589934080 0 to 8,589,934,080 bits

Function
Queries the Burst output trigger signal generation timing (delay).

Example
> :SOURce:PATTern:BURSt:TRIGdelay? BURStout1
< 128

Compatibility
Partially compatible with MU181020A/B (the parameter range varies depending on the settings).
Chapter 7  SCPI Commands

:SOURce:PATTern:BURSt:PULSewidth <port>,<numeric>

Parameter  

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>&lt;port&gt;</th>
<th>&lt;numeric&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;port&gt;</td>
<td>&lt;CHARACTER PROGRAM DATA&gt;</td>
<td>BURStout1 Burst Output</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>BURStout2 Burst Output2</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Valid only when AUX Output is set to Burst Output2.</td>
<td></td>
</tr>
<tr>
<td>&lt;numeric&gt;</td>
<td>&lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
<td>Independent 0 to 2147483520/ 0 to 2,147,483,520 bits</td>
<td>8 bits Step</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2 ch Combination 0 to 4294967040/ 0 to 4,294,967,040 bits</td>
<td>16 bits Step</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4 ch Combination 0 to 8589934080/ 0 to 8,589,934,080 bits</td>
<td>32 bits Step</td>
</tr>
</tbody>
</table>

Note:  
The maximum setting is the pattern length.

Function  
Sets the Burst output trigger signal generation interval width to adjust the phase on the DUT side.

Example  
To set the Burst output trigger signal generation interval width to 1,024 bits:  
> :SOURce:PATTern:BURSt:PULSewidth BURStout1,1024

Compatibility  
Partially compatible with MU181020A/B (the parameter range varies depending on the settings).

:SOURce:PATTern:BURSt:PULSewidth? <port>

Parameter  

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>&lt;port&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;port&gt;</td>
<td>&lt;CHARACTER PROGRAM DATA&gt;</td>
<td>BURStout1 Burst Output</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BURStout2 Burst Output2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Valid only when AUX Output is set to Burst Output2.</td>
</tr>
</tbody>
</table>

Response  

| <numeric>    | <NR1 NUMERIC RESPONSE DATA> | 0 to 8589934080 | 0 to 8,589,934,080 bits |

Function  
Queries the Burst output trigger signal generation interval width.

Example  
To query the Burst output trigger signal generation interval width:  
> :SOURce:PATTern:BURSt:PULSewidth? BURStout1  
< 1024

Compatibility  
Partially compatible with MU181020A/B (the parameter range varies depending on the settings).
7.11.2 AUX Input setting commands

![Figure 7.11.6.2-1 AUX Input Setting](image)

**Table 7.11.6.2-1 AUX Input Setting Commands**

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>:INPut:INPut:FUNCtion?</td>
</tr>
</tbody>
</table>

**:INPut:INPut:FUNCtion <function>**

Parameter

- `<function> = <CHARACTER PROGRAM DATA>`
- `BURSt` Burst
- `ERRor` Error Injection
  
However, Error Injection is set at Repeat.

Function

Sets the function of AUX Input.

Example

To set the function of AUX Input to Error Injection.

> :INPut:INPut:FUNCtion ERRor

Compatibility

Compatible with MU181020A/B.

**:INPut:INPut:FUNCtion?**

Response

- `<function> = <CHARACTER RESPONSE DATA>`
- `BURS, ERR`

Function

Queries the function of Auxiliary Input.

Example

> :INPut:INPut:FUNCtion?

< ERR

Compatibility

Compatible with MU181020A/B.
### 7.11.6.3 AUX Output setting commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>:OUTPut:SYNC:SOURce?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:OUTPut:SYNC:POSition?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:OUTPut:SYNC:MIXData:BLOCk?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:OUTPut:SYNC:MIXData:ROW?</td>
</tr>
</tbody>
</table>

**Figure 7.11.6.3-1** AUX Output Setting  
(When 1/N Clock Is Selected)

**Figure 7.11.6.3-2** AUX Output Setting  
(When Pattern Sync Is Selected: PRBS, Zero-Substitution, Data Pattern)

**Figure 7.11.6.3-3** AUX Output Setting  
(When Pattern Sync Is Selected: Mixed Data Pattern)
7.11 28G/32G bit/s PPG Commands

:OUTPut:SYNC:SOURce <source>[,<numeric>]

Parameter
- <source> = <CHARACTER PROGRAM DATA>
- PATTern: Pattern Sync (Variable)
- BURSt: Burst Output2
- NCLock: 1/N clock
- OFF: OFF
- [<numeric>] = <DECIMAL NUMERIC PROGRAM DATA>
- 4 to 512: 4 to 512 / 2 Step

Function
- Sets the output signal for synchronization output.

Example
- To set the output signal to Pattern Sync:
  > :OUTPut:SYNC:SOURce PATT
- To set the output signal for synchronization output to 1/512 clocks:
  > :OUTPut:SYNC:SOURce NCLock, 512

Compatibility
- Partially compatible with MU181020A/B (the parameter range varies depending on the settings).

:OUTPut:SYNC:SOURce?

Response
- <source> = <CHARACTER RESPONSE DATA>
- PATT, BURS, NCL, OFF
- [<numeric>] = <NR1 NUMERIC RESPONSE DATA>
- 4 to 512

Function
- Queries the output signal for synchronization output.

Example
- To query the output signal for synchronization output:
  > :OUTPut:SYNC:SOURce?
  < PATT
  > :OUTPut:SYNC:SOURce?
  < NCL, 512

Compatibility
- Partially compatible with MU181020A/B (the parameter range varies depending on the settings).
Chapter 7  SCPI Commands

:OUTPut:SYNC:POSition <numeric>

Parameter  <numeric> = <DECIMAL NUMERIC PROGRAM DATA>

<table>
<thead>
<tr>
<th>Independent</th>
<th>1 to 34,359,738,105/8 bits Step</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 ch Combination</td>
<td>1 to 68,719,476,209/16 bits Step</td>
</tr>
<tr>
<td>4 ch Combination</td>
<td>1 to 137,438,952,417/32 bits Step</td>
</tr>
</tbody>
</table>

Function  Sets the output position for synchronization output for Data, PRBS, and Zero-Substitution patterns.

Example  To set the synchronization output position to bit 17:
> :OUTPut:SYNC:POSition 17

Compatibility  Partially compatible with MU181020A/B (the parameter range varies depending on the settings).

:OUTPut:SYNC:POSition?

Response  <numeric> = <NR1 NUMERIC RESPONSE DATA>

| 1 to 137438952417 | 1 to 137,438,952,417 bits |

Function  Queries the output position for synchronization output for Data, PRBS, and Zero-Substitution patterns.

Example  > :OUTPut:SYNC:POSition?
< 17

Compatibility  Partially compatible with MU181020A/B (the parameter range varies depending on the settings).

:OUTPut:SYNC:MIXData:BLOCk <numeric>

Parameter  <numeric> = <DECIMAL NUMERIC PROGRAM DATA>

| 1 to 511 | 1 to 511 Block/1 Block Step |

Function  Sets the output block position for synchronization output for Mixed Data pattern.

Example  To set the synchronization output block to block 10:
> :OUTPut:SYNC:MIXData:BLOCk 10

Compatibility  Compatible with MU181020A/B.
7.11 28G/32G bit/s PPG Commands

:OUTPut:SYNC:MIXData:BLOCk?

Response
<numeric> = <NR1 NUMERIC RESPONSE DATA>
1 to 511
1 to 511 Block
The maximum setting value is the set number of blocks.

Function
Queries the output block position for synchronization output for Mixed Data pattern.

Example
> :OUTPut:SYNC:MIXData:BLOCk?
10

Compatibility
Compatible with MU181020A/B.

:OUTPut:SYNC:MIXData:ROW <numeric>

Parameter
<numeric> = <DECIMAL NUMERIC PROGRAM DATA>
1 to 16
1 to 16 Row/1 Row Step
The maximum setting value is the set number of Rows.

Function
Sets the output Row position for synchronization output for Mixed Data pattern.

Example
To set the synchronization output Row to 1 Row.
> :OUTPut:SYNC:MIXData:ROW 1

Compatibility
Compatible with MU181020A/B.

:OUTPut:SYNC:MIXData:ROW?

Response
<numeric> = <NR1 NUMERIC RESPONSE DATA>
1 to 16
1 to 16 Row/1 Row Step
The maximum setting value is the set number of Rows.

Function
Queries the output Row position for synchronization output for Mixed Data pattern.

Example
> :OUTPut:SYNC:MIXData:ROW?
1

Compatibility
Compatible with MU181020A/B.
7.11.6.4 Gating Output setting commands

Table 7.11.6.4-1  Gating Output Setting Commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>:OUTPut:SYNC:GATing:SOURce?</td>
</tr>
</tbody>
</table>

:OUTPut:SYNC:GATing:SOURce <output>

Parameter
- <output> = <CHARACTER PROGRAM DATA>
  - OFF: Gating Output OFF
  - ON: Gating Output ON

Function
Sets Gating Output to ON or OFF.

Example
To set Gating Output to OFF.
> :OUTPut:SYNC:GATing:SOURce OFF

:OUTPut:SYNC:GATing:SOURce?

Response
- <output> = <CHARACTER RESPONSE DATA>
  - OFF, ON

Function
Queries if Gating Output is ON or OFF.

Example
> :OUTPut:SYNC:GATing:SOURce?
< OFF
7.11 28G/32G bit/s PPG Commands

7.11.7 Misc 2 Tab

7.11.7.1 Clock Setting commands

![Image of Clock Setting](image)

**Figure 7.11.7.1-1 Clock Setting (When MU181000B Is Selected)**

![Image of Clock Setting](image)

**Figure 7.11.7.1-2 Clock Setting (When External Is Selected)**

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>:SYSTem:INPut:CSELect?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SYSTem:OUTPut:BITRate?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:OUTPut:CLOCK:OFFSet:PPM?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:OUTPut:RCLock:SELect?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SYSTem:OUTPut:CRATe?</td>
</tr>
</tbody>
</table>
Bit Rate in Figure 7.11.7.1-1 “Clock Setting (When MU181000B Is Selected)” is set by the command :OUTPut:DATA:BITRate described in 7.11.2.1 “Output Setting Commands.”

:SYSTem:INPut:CSELect <input>

Parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;input&gt; = &lt;CHARACTER PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>EXTernal External synthesizer</td>
</tr>
<tr>
<td></td>
<td>INTernal[1 to 7] Built-in synthesizer or jitter generation source</td>
</tr>
</tbody>
</table>

Note:

INTernal [1 to 7] specifies the unit and slot numbers from the smallest as 1 to 7 when there is more than one built-in synthesizer. If [1 to 7] is omitted, 1 is assumed.

Function

Selects synthesizer for clock input source.

Example

To choose #2 built-in synthesizer as clock input source.

> :SYSTem:INPut:CSELect INT2

Compatibility

Compatible with MU181500B.

:SYSTem:INPut:CSELect?

Response

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;numeric&gt; = &lt;CHARACTER RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>EXT, INT1 to 7</td>
</tr>
<tr>
<td></td>
<td>&lt;info&gt; = &lt;STRING RESPONSE DATA&gt;</td>
</tr>
<tr>
<td></td>
<td>“X:Y Z”</td>
</tr>
<tr>
<td></td>
<td>X: unit No. 1 to 4</td>
</tr>
<tr>
<td></td>
<td>Y: slot No. 1 to 6</td>
</tr>
<tr>
<td></td>
<td>Z: module model name MU181000A/B or MU181500B</td>
</tr>
</tbody>
</table>

Function

Queries the synthesizer of the selecting clock input source.

Example

> :SYSTem:INPut:CSELect?

< INT2,"1:4 MU181500B"

Compatibility

Compatible with MU181500B.
7.11 28G/32G bit/s PPG Commands

:SYSTem:INPUT:CSELeCt:MODule?

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;info&gt;,&lt;info1&gt;,&lt;info2&gt;,...,&lt;info7&gt; = &lt;STRING RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>&lt;info&gt; = &quot;X:Y Z&quot;</td>
</tr>
<tr>
<td></td>
<td>X: unit No. 1 to 4</td>
</tr>
<tr>
<td></td>
<td>Y: slot No. 1 to 6</td>
</tr>
<tr>
<td></td>
<td>Z: module model name MU181000A, MU181000B, or MU181500B</td>
</tr>
<tr>
<td></td>
<td>NONE is returned when built-in synthesizer and jitter signal source are not present.</td>
</tr>
</tbody>
</table>

Function
Queries the built-in synthesizer in the system.

Example
> :SYSTem:INPUT:CSELeCt:MODule?
< "1:2 MU181000A","1:4 MU181000A"
< "NONE"

Compatibility
Compatible with MU181500B.

:SYSTem:OUTPut:BITRate <input>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;input&gt; = &lt;CHARACTER PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>When using the external clock:</td>
</tr>
<tr>
<td></td>
<td>HIGH 20 to 28.1 G (32.1G) bit/s (when setting Fullrate clock)</td>
</tr>
<tr>
<td></td>
<td>MIDDlE 16 to 20.4 Gbit/s (when setting Fullrate clock)</td>
</tr>
<tr>
<td></td>
<td>LOW 2.4 to 16 Gbit/s (when setting Fullrate clock)</td>
</tr>
<tr>
<td></td>
<td>HALF 2.4 to 28.1 (32.1) G bit/s (when setting Halfrate clock)</td>
</tr>
<tr>
<td></td>
<td>HIGH1 25 to 28.1 (32.1) G bit/s (when setting Fullrate/Halfrate clock)</td>
</tr>
</tbody>
</table>

When using the eternal clock and MU181500B Jitter Modulation Source:

|           | HIGH 25 to 32.1 Gbit/s (when setting Fullrate clock) |
|           | MIDDlE1 20 to 28.1 G (32.1G) bit/s (when setting Fullrate clock) |
|           | MIDDlE2 12.5 to 20 Gbit/s (when setting Fullrate clock) |
|           | LOW 2.4 to 15 Gbit/s (when setting Fullrate clock) |
|           | HALF1 30 to 32.1 Gbit/s (when setting Halfrate clock) |
|           | HALF2 2.4 to 28.1 (30.0) G bit/s (when setting Halfrate clock) |

Function
Selects operation bit rate band when inputting external clock.

Example
To set operation bit rate band to Halfrate clock 2.4 to 28.1 (32.1) Gbit/s when inputting external clock:
> :SYSTem:OUTPut:BITRate HALF

Compatibility
Incompatible with existing models.
Chapter 7  SCPI Commands

:SYSTem:OUTPut:BITRate?

Response  <input> = <CHARACTER RESPONSE DATA>
HIGH, MIDD, LOW, HALF, MIDD1, MIDD2, HALF1, HALF2, HIGH1

Function  Queries operation bit rate band when inputting external clock.

Example  > :SYSTem:OUTPut:BITRate?
< MIDDLE

Compatibility  Incompatible with existing models.

:OUTPut:CLOCk:OFFSet:PPM <numeric>

Parameter  <numeric> = <DECIMAL NUMERIC PROGRAM DATA>
–1000 to 1000 –1,000 to +1,000 ppm/Step 1 ppm

Function  Sets offset of frequency (ppm) currently output from tracking built-in synthesizer.

Example  To set offset of currently output frequency to 0 ppm:
> :OUTPut:CLOCk:OFFSet:PPM 0

Compatibility  Compatible with MU181500B and MU181000A/B.

:OUTPut:CLOCk:OFFSet:PPM?

Response  <numeric> = <NR1 NUMERIC RESPONSE DATA>
–1000 to 1000 –1,000 to +1,000 ppm

Function  Queries offset (ppm) of frequency currently output from tracking built-in synthesizer.

Example  > :OUTPut:CLOCk:OFFSet:PPM?
< 0

Compatibility  Compatible with MU181500B and MU181000A/B.

:OUTPut:RCLock:SELect <clock>

Parameter  <clock> = <CHARACTER PROGRAM DATA>
INTernal  Internal reference signal 10 MHz
EXTernal10  External reference signal 10 MHz

Function  Selects either built-in or external input for 10 MHz reference signal of tracking built-in synthesizer.

Example  To set 10 MHz reference signal to external input:
> :OUTPut:RCLock:SELect EXT10

Compatibility  Compatible with MU181500B and MU181000A/B.
### :OUTPut:RCLock:SELeCt?

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;clock&gt; = &lt;CHARACTER RESPONSE DATA&gt; INT, EXT10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function</td>
<td>Queries setting for 10 MHz reference signal of tracking built-in synthesizer.</td>
</tr>
<tr>
<td>Example</td>
<td>&gt; :OUTPut:RCLock:SELeCt?</td>
</tr>
<tr>
<td></td>
<td>&lt; EXT10</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Compatible with MU181500B and MU181000A/B.</td>
</tr>
</tbody>
</table>

### :SYStem:OUTPut:CRATe <rate>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;rate&gt; = &lt;CHARACTER PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>HALFrate 1/2 clock output to data</td>
</tr>
<tr>
<td></td>
<td>FULLrate 1/1 clock output to data</td>
</tr>
<tr>
<td>Function</td>
<td>Selects output data and clock correlation.</td>
</tr>
<tr>
<td>Example</td>
<td>To set output data/clock to Half Rate:</td>
</tr>
<tr>
<td></td>
<td>&gt; :SYStem:OUTPut:CRATe HALF</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Incompatible with existing models.</td>
</tr>
</tbody>
</table>

### :SYStem:OUTPut:CRATe?

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;rate&gt; = &lt;CHARACTER RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>HALF, FULL</td>
</tr>
<tr>
<td>Function</td>
<td>Queries the relationship between output data and clock.</td>
</tr>
<tr>
<td>Example</td>
<td>&gt; :SYStem:OUTPut:CRATe?</td>
</tr>
<tr>
<td></td>
<td>&lt; HALF</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Incompatible with existing models.</td>
</tr>
</tbody>
</table>
### :SYSTem:INPut:CFReq?

<table>
<thead>
<tr>
<th>Response</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;string&gt; = &lt;STRING RESPONSE DATA&gt;</td>
<td>When using the external clock:</td>
</tr>
<tr>
<td>“10 to 16.05 GHz”</td>
<td>(when setting Fullrate clock)</td>
</tr>
<tr>
<td>“10 to 14.05 GHz”</td>
<td>Without Opt-x01</td>
</tr>
<tr>
<td>“8 to 10.02 GHz”</td>
<td>(when setting Fullrate clock)</td>
</tr>
<tr>
<td>“2.4 to 16 GHz”</td>
<td>(when setting Fullrate clock)</td>
</tr>
<tr>
<td>“1.2 to 16.05 GHz”</td>
<td>(when setting Halfrate clock)</td>
</tr>
<tr>
<td>“1.2 to 14.05 GHz”</td>
<td>Without Opt-x01</td>
</tr>
<tr>
<td>“6.25 to 8.025 GHz”</td>
<td>(when setting Fullrate/Halfrate clock)</td>
</tr>
<tr>
<td>“6.25 to 7.025 GHz”</td>
<td>Without Opt-x01</td>
</tr>
</tbody>
</table>

When using the external clock and MU181500B Jitter Modulation Source:

| “6.25 to 8.025 GHz” | (when setting Fullrate clock) |
| “10.0 to 15.0 GHz” | (when setting Fullrate clock) |
| “10.0 to 14.05 GHz” | Without Opt-x01 |
| “6.25 to 10.0 GHz” | (when setting Fullrate clock) |
| “2.4 to 15.0 GHz” | (when setting Fullrate clock) |
| “7.5 to 8.025 GHz” | (when setting Fullrate clock) |
| “1.2 to 15.0 GHz” | (when setting Halfrate clock) |
| “1.2 to 14.05 GHz” | Without Opt-x01 |

**Function**

Queries frequency band of clock input to MU183020A/MU183021A when inputting external clock.

**Example**

```plaintext
> :SYSTem:INPut:CFReq?
< "1.2 to 16.05 GHz"
```

**Compatibility**

Incompatible with the existing modules.
7.11.7.2 Combination setting commands

Figure 7.11.7.2-1 Combination Setting Commands (Combination)

Table 7.11.7.2-1 Combination Setting Commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>:MCOMbination:OPERation:SETTing?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:MCOMbination:OPERation:CHSetting?</td>
</tr>
</tbody>
</table>
Chapter 7  SCPI Commands

:MCOMbination:OPERation:SETTing <numeric>

Parameter  

<table>
<thead>
<tr>
<th>&lt;numeric&gt; =</th>
<th>DESCIMAL NUMERIC PROGRAM DATA</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Independent</td>
</tr>
<tr>
<td>1</td>
<td>Channel Synchronization</td>
</tr>
<tr>
<td>2</td>
<td>2ch Combination</td>
</tr>
<tr>
<td>3</td>
<td>2ch Combination CH Sync</td>
</tr>
<tr>
<td>4</td>
<td>4ch Combination</td>
</tr>
<tr>
<td>5</td>
<td>64G x 2ch Combination</td>
</tr>
</tbody>
</table>

Function  

Sets Combination and Independent at the 32G PPG/ED module.

Example  

To set to 4ch Combination:

```
> :MCOMbination:OPERation:SETTing 4
```

Compatibility  

Incompatible with existing models.

:MCOMbination:OPERation:SETTing?

Response  

<table>
<thead>
<tr>
<th>&lt;numeric&gt; =</th>
<th>NR1 NUMERIC RESPONSE DATA</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Independent</td>
</tr>
<tr>
<td>1</td>
<td>Channel Synchronization</td>
</tr>
<tr>
<td>2</td>
<td>2ch Combination</td>
</tr>
<tr>
<td>3</td>
<td>2ch Combination CH Sync</td>
</tr>
<tr>
<td>4</td>
<td>4ch Combination</td>
</tr>
<tr>
<td>5</td>
<td>64G x 2ch Combination</td>
</tr>
</tbody>
</table>

Function  

Queries the combination status at the 32G PPG/ED module.

Example  

```
> :MCOMbination:OPERation:SETTing?
< 0
```

Compatibility  

Incompatible with existing models.

:MCOMbination:OPERation:CHSetting <interface>

Parameter  

<table>
<thead>
<tr>
<th>&lt;interface&gt; =</th>
<th>DECIMAL NUMERIC PROGRAM DATA</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 to 14</td>
<td>Total number of Data Interface that can configure channel synchronization (decimal)</td>
</tr>
</tbody>
</table>

Available bits  

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 (Bit1)</td>
<td>Selects Data2</td>
</tr>
<tr>
<td>4 (Bit2)</td>
<td>Selects Data3</td>
</tr>
<tr>
<td>8 (Bit3)</td>
<td>Selects Data4</td>
</tr>
</tbody>
</table>

Function  

Specifies the Data Interface for setting channel synchronization.

Example  

To set Data 1 to 4 to channel synchronization.

```
> :MCOMbination:OPERation:CHSetting 14
```

Compatibility  

Incompatible with existing models.
### :MCOMbination:OPERation:CHSetting?

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;interface&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 to 14</td>
<td>Total number of Data Interface that can configure channel synchronization (decimal)</td>
</tr>
<tr>
<td>Available bits</td>
<td></td>
</tr>
<tr>
<td>2 (Bit1)</td>
<td>Data2</td>
</tr>
<tr>
<td>4 (Bit2)</td>
<td>Data3</td>
</tr>
<tr>
<td>8 (Bit3)</td>
<td>Data4</td>
</tr>
</tbody>
</table>

**Function**

Queries Data Interface for setting Channel Synchronization.

**Example**

```
> :MCOMbination:OPERation:CHSetting?
< 2  (at Data1 and 2)
< 6  (at Data1 through 3)
< 14 (at Data1 through 4)
```

**Compatibility**

Incompatible with existing models.
7.11.7.3 Grouping setting commands

Figure 7.11.7.3-1 Grouping Setting Dialog Box

Table 7.11.7.3-1 Grouping Setting Commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1]</td>
<td>Selects tabs to be grouped together.</td>
<td>:SOURce:GROup:SETTing</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SOURce:GROup:SETTing?</td>
</tr>
<tr>
<td>[2]</td>
<td>Selects Data Ifs to be grouped together.</td>
<td>:SOURce:GROup:SELeCT</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SOURce:GROup:SELeCT?</td>
</tr>
</tbody>
</table>

:SOURce:GROup:SETTing <tab>,<interface>

Parameter

- `<tab>` = <CHARACTER PROGRAM DATA>
  - PATTern = Pattern tab
  - OUTPut = Output tab
- `<interface>` = <NR1 NUMERIC PROGRAM DATA>
  - 0 to 15
  - Available bits
    - 1 (Bit0) = Data1
    - 2 (Bit1) = Data2
    - 4 (Bit2) = Data3
    - 8 (Bit3) = Data4

Function
Sets the Grouping function.

Example
To group the Pattern tabs of Data1 and Data2 together:
> :SOURce:GROup:SETTing PATTern,3

To group the Pattern tabs of Data1 to Data4 together:
> :SOURce:GROup:SETTing OUTPut,15

Compatibility
Incompatible with existing models.
7.11 28G/32G bit/s PPG Commands

:SOURce:GROup:SETTing? <tab>

Parameter
<tab> = <CHARACTER PROGRAM DATA>
PATTern      Pattern tab
OUTPut       Output tab

Response
<interface> = <NR1 NUMERIC RESPONSE DATA>
0 to 15
Available bits
1 (Bit0)      Data1
2 (Bit1)      Data2
4 (Bit2)      Data3
8 (Bit3)      Data4

Function
Queries the Grouping setting.
To query the Data Interfaces whose Output tabs are grouped together:

Example
> :SOURce:GROup:SETTing? OUTPut
< 12 (Data3 and Data4 are grouped together.)

Compatibility
Incompatible with existing models.

:SOURce:GROup:SELect <group>

Parameter
<group> = <DECIMAL NUMERIC PROGRAM DATA>
1 One group (Data1-2 or Data1-4)
2 Two groups (Data1-2 and Data3-4)

Function
Sets the number of groups.

Example
To set the number of groups to 2:
> :SOURce:GROup:SELect 2

Compatibility
Incompatible with existing models.

:SOURce:GROup:SELect?

Response
<interface> = <NR1 NUMERIC RESPONSE DATA>
1 One group (Data1-2 or Data1-4)
2 Two groups (Data1-2 and Data3-4)

Function
Queries the number of groups.

Example
> :SOURce:GROup:SELect?
< 2

Compatibility
Incompatible with existing models.
7.12 28G/32G bit/s ED Commands


Before executing a setting/query command for the MU183040A/40B/41A/41B, specify the slot number of the module to be remotely controlled, by using the :MODule:ID command. Refer to Section 7.1 “Common Commands” and 9.1 “Basic Operation of Device Messages” for how to specify a slot number with the :MODule:ID command.

7.12.1 Common Commands

7.12.1.1 Commands for common settings

<table>
<thead>
<tr>
<th>Table 7.12.1.1-1 Common Setting Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Setting Items</strong></td>
</tr>
<tr>
<td>Number of the unit to be operated</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Number of the module to be operated (slot position)</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Number of the data interface to be operated</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Setting of automatic measurement to be operated</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Query for error message</td>
</tr>
<tr>
<td>Query for SCPI version</td>
</tr>
<tr>
<td>Query for software status</td>
</tr>
<tr>
<td>Query for hardware system configuration</td>
</tr>
<tr>
<td>Query for system error</td>
</tr>
<tr>
<td>Terminator type</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Query for system error</td>
</tr>
<tr>
<td>Query for mainframe information</td>
</tr>
<tr>
<td>Query for module information</td>
</tr>
<tr>
<td>Setting of measurement result drawing</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Module screen display</td>
</tr>
</tbody>
</table>
### :UENTry:ID <unit_number>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;unit_number&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function</td>
<td>Sets the number of the unit to be operated</td>
</tr>
<tr>
<td>Example</td>
<td>To set the number of the unit to be operated to 2:</td>
</tr>
<tr>
<td></td>
<td>&gt; :UENTry:ID 2</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Incompatible with existing models.</td>
</tr>
</tbody>
</table>

### :UENTry:ID?

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;unit_number&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function</td>
<td>Queries the number of the unit being operated.</td>
</tr>
<tr>
<td>Example</td>
<td>&gt; :UENTry:ID?</td>
</tr>
<tr>
<td></td>
<td>&lt; 2</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Incompatible with existing models.</td>
</tr>
</tbody>
</table>

### :MODule:ID <module_number>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;module_number&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function</td>
<td>Sets the number of the module to be operated (slot position).</td>
</tr>
<tr>
<td>Example</td>
<td>To set the number of the module to be operated (slot position) to 6:</td>
</tr>
<tr>
<td></td>
<td>&gt; :MODule:ID 6</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Incompatible with existing models.</td>
</tr>
</tbody>
</table>

### :MODule:ID?

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;module_number&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function</td>
<td>Queries the number of the module being operated (slot position).</td>
</tr>
<tr>
<td>Example</td>
<td>&gt; :MODule:ID?</td>
</tr>
<tr>
<td></td>
<td>&lt; 6</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Incompatible with existing models.</td>
</tr>
</tbody>
</table>
**Chapter 7  SCPI Commands**

### :INTerface:ID <interface_number>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;interface_number&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1 to 4  Data interface number</td>
</tr>
</tbody>
</table>

The setting range varies according to the module model.

- MU183040A/B-x10: 1
- MU183040A/B-x20: 1 to 2
- MU183041A/B: 1 to 4

**Function**

Sets the number of the data interface (Data 1 to 4) to be operated.

**Example**

To set the number of the data interface to be operated to 4:

```
> :INTerface:ID 4
```

**Compatibility**

Incompatible with existing models.

### :INTerface:ID?

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;interface_number&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1 to 4</td>
</tr>
</tbody>
</table>

**Function**

Queries the number of the data interface (Data 1 to 4) being operated.

**Example**

```
> :INTerface:ID?
< 4
```

**Compatibility**

Incompatible with existing models.

### :SYSTem:CFUNction <function>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;function&gt; = &lt;CHARACTER PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASE</td>
<td>Auto Search</td>
</tr>
<tr>
<td>ISI</td>
<td>ISI measurement</td>
</tr>
<tr>
<td>EMAR</td>
<td>Eye Margin measurement</td>
</tr>
<tr>
<td>EDI</td>
<td>Eye Diagram measurement</td>
</tr>
<tr>
<td>QAN</td>
<td>Q measurement</td>
</tr>
<tr>
<td>BTUB</td>
<td>Bathtub measurement</td>
</tr>
<tr>
<td>AADJ</td>
<td>Auto Adjust</td>
</tr>
<tr>
<td>OFF</td>
<td>Off</td>
</tr>
<tr>
<td>ASE32</td>
<td>Auto Search (MU183040A/41A/40B/41B)</td>
</tr>
<tr>
<td>EMAR32</td>
<td>Eye Margin measurement (MU183040A/41A/40B/41B)</td>
</tr>
<tr>
<td>EDI32</td>
<td>Eye Diagram measurement (MU183040A/41A/40B/41B)</td>
</tr>
<tr>
<td>BTUB32</td>
<td>Bathtub measurement (MU183040A/41A/40B/41B)</td>
</tr>
<tr>
<td>AADJ32</td>
<td>Auto Adjust (MU183040A/41A/40B/41B)</td>
</tr>
<tr>
<td>QAN32</td>
<td>Q measurement (MU183040A/41A/40B/41B)</td>
</tr>
<tr>
<td>PAMB</td>
<td>PAM BER measurement (MU183040B/41B)</td>
</tr>
<tr>
<td>ECT</td>
<td>Eye Contour measurement (MU183040A/41A/40B/41B)</td>
</tr>
</tbody>
</table>

**Note:**

When “OFF” is set, the operation returns to the port operation previously performed.
### Function
Sets the automatic measurement function to be performed.

### Example
To set the common function to be performed to Auto Search (32G):

```
> :SYSTem:CFUNction ASE32
```

### Compatibility
Incompatible with existing models.

### :SYSTem:CFUNction?

<table>
<thead>
<tr>
<th>Response</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASE</td>
<td>Auto Search</td>
</tr>
<tr>
<td>ISI</td>
<td>ISI measurement</td>
</tr>
<tr>
<td>EMAR</td>
<td>Eye Margin measurement</td>
</tr>
<tr>
<td>EDI</td>
<td>Eye Diagram measurement</td>
</tr>
<tr>
<td>QAN</td>
<td>Q measurement</td>
</tr>
<tr>
<td>BTUB</td>
<td>Bathtub measurement</td>
</tr>
<tr>
<td>AADJ</td>
<td>Auto Adjust</td>
</tr>
<tr>
<td>OFF</td>
<td>Off</td>
</tr>
<tr>
<td>ASE32</td>
<td>Auto Search (MU183040A/41A/40B/41B)</td>
</tr>
<tr>
<td>EMAR32</td>
<td>Eye Margin measurement (MU183040A/41A/40B/41B)</td>
</tr>
<tr>
<td>EDI32</td>
<td>Eye Diagram measurement (MU183040A/41A/40B/41B)</td>
</tr>
<tr>
<td>BTUB32</td>
<td>Bathtub measurement (MU183040A/41A/40B/41B)</td>
</tr>
<tr>
<td>AADJ32</td>
<td>Auto Adjust (MU183040A/41A/40B/41B)</td>
</tr>
<tr>
<td>QAN32</td>
<td>Q measurement (MU183040A/41A/40B/41B)</td>
</tr>
<tr>
<td>PAMB</td>
<td>PAM BER measurement (MU183040B/41B)</td>
</tr>
<tr>
<td>ECT</td>
<td>Eye Contour measurement (MU183040A/41A/40B/41B)</td>
</tr>
</tbody>
</table>

### Function
Queries the automatic measurement function being performed.

### Example
```
> :SYSTem:CFUNction?
< ASE32
```

### Compatibility
Incompatible with existing models
Chapter 7  SCPI Commands

:SYSTem:ERRor?

Response  
<error/event_number>,"<error/event_description>"

<error/event_number> = <NR1 NUMERIC RESPONSE DATA>
–32768 to 32767
“0” indicates that no errors and events have occurred.
Other values return a general error reserved by SCPI or a device-unique error.

<error/event_description> = <STRING RESPONSE DATA>
This is an error message corresponding to <error/event_number>.
The maximum character-string length is 255 characters.

Function  
Queries the error message in the error/event queue.
Refer to Chapters 8 and 9 for details on the error messages.

Example  
> :SYSTem:ERRor?
< 0,"No error"

Compatibility  
Compatible with the MP1632C and MP1776A.

:SYSTem:VERSion?

Response  
<version> = <NR2 NUMERIC RESPONSE DATA>
YYYY.V  YYY: Year
V: Revision number

Function  
Queries the SCPI version to which the MU183040A/41A/40B/41B conforms.

Example  
> :SYSTem:VERSion?
< 1999.0

Compatibility  
Compatible with MU181040A/B.
:SYSTem:CONDition?

Response

"<mainframe>,<slot1>,...,<slot64>"

<mainframe> = <serial>,<mver>,<hver>,<opt1>,<sbver>,<saver>,<opt2>

<serial> = <STRING RESPONSE DATA>

XXXXXXXXXX 0000000000 to 9999999999
Mainframe serial number

Note:

Alphabetic characters may be included.

<mver> = <STRING RESPONSE DATA>

XXXX.XX.XX 1.00.00 to 9999.99.99
Main application software version

<hver> = <STRING RESPONSE DATA>

XXXX.XX.XX 1.00.00 to 9999.99.99
Mainframe hardware version

<opt1> = <STRING RESPONSE DATA>

OPTXXX Option number (MP1800A/MT1810A)

See Table 7.12.1.1-2 "Option Character Correspondence Table".

Note:

Outputs the numbers for all installed options.
NONE is output if no option is installed.

<sbver> = <STRING RESPONSE DATA>

XXXX.XX.XX 1.00.00 to 9999.99.99
Sub application software version (Boot part)

<saver> = <STRING RESPONSE DATA>

XXXX.XX.XX 1.00.00 to 9999.99.99
Sub application software version (Application part)

<opt2> = <STRING RESPONSE DATA>

<slot x> = <module>,<serial>,<fpga1>,<fpga2>,<boot>,<application>,<opt>

x indicates a slot number. The slot number varies depending on the unit number as follows.

Unit 1: 1 to 16 For the MP1800A, numbers from 1 to 6 correspond to actual slots. For the MT1810A, numbers from 1 to 4 correspond to actual slots.

Unit 2: 17 to 32 For the MT1810A, numbers from 17 to 20 correspond to actual slots.

Unit 3: 33 to 48 For the MT1810A, numbers from 33 to 36 correspond to actual slots.

Unit 4: 49 to 64 For the MT1810A, numbers from 49 to 52 correspond to actual slots.
<module> = <STRING RESPONSE DATA>
XXXXXX Module name (e.g.: MU181020A)
See Table 7.12.1.1-2 “Option Character Correspondence Table”.

Note:
NONE is output if no module is installed.
For a module that uses two slots, only the slot with the greater number is valid.

<serial> = <STRING RESPONSE DATA>
XXXXX Serial number
0000000000 to 9999999999

Note:
“-------------” is output if no module is installed.
For a module that uses two slots, only the slot with the greater number is valid.

<fpga1>[,<fpga2>,.....] = <STRING RESPONSE DATA>
XXXX.XX.XX FPGA version
1.00.00 to 9999.99.99

<boot> = <STRING RESPONSE DATA>
XXXX.XX.XX Logic Boot version
1.00.00 to 9999.99.99

Note:
“-------------” is output if Logic boot is not installed.
For a module that uses two slots, only the slot with the greater number is valid.

<application> = <STRING RESPONSE DATA>
XXXX.XX.XX Logic Application version
1.00.00 to 9999.99.99

Note:
“-------------” is output if Logic Application is not installed.
For a module that uses two slots, only the slot with the greater number is valid.

<opt> = <STRING RESPONSE DATA>
XXXXXX/XXXXX Option number
OPTXXX: For MP1800A/MT1810A

Note:
Outputs the numbers for all installed options.
For a module that uses two slots, only the slot with the greater number is valid.
The option names of the MU181020B-003 and MU181040B-003 are not output.
Function
Queries the software status of MU183040A/41A/40B/41B.

Example
> :SYSTem:CONDition?
<
6201234567,1.00.00,1.00.20,OPT302,1.00.00,1.00.00,OPT12,OPT14,MU181000A,6201234568,1.00.00,1.00.00,1.00.00,OPT10,MU181020A,6201234569,1.00.00,1.00.00,1.00.00,OPT001,OPT010,MU181040A,6201234571,1.00.00,1.00.00,1.00.00,OPT002,OPT220

Compatibility
Incompatible with existing models.

<table>
<thead>
<tr>
<th>Model/Name</th>
<th>Option Number</th>
<th>Option Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>MP1800A</td>
<td>OPT014</td>
<td>2-Slot PPG/ED</td>
</tr>
<tr>
<td></td>
<td>OPT015</td>
<td>4-Slot PPG/ED</td>
</tr>
<tr>
<td></td>
<td>OPT016</td>
<td>6-Slot PPG/ED</td>
</tr>
<tr>
<td></td>
<td>OPT032</td>
<td>32Gbit/s PPG/ED support</td>
</tr>
<tr>
<td>MT1810A</td>
<td>OPT014</td>
<td>2-Slot PPG/ED</td>
</tr>
<tr>
<td>4 Slot Chassis</td>
<td>OPT015</td>
<td>4-Slot PPG/ED</td>
</tr>
<tr>
<td></td>
<td>OPT032</td>
<td>32Gbit/s PPG/ED support</td>
</tr>
<tr>
<td>MX180000A</td>
<td>OPTx01</td>
<td>Jitter Modulation</td>
</tr>
<tr>
<td>Signal Quality Analyzer Control Software</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>MU181000A</td>
<td>OPTx01</td>
<td>Jitter Modulation</td>
</tr>
<tr>
<td>12.5GHz Synthesizer</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>MU181000B</td>
<td>OPTx01</td>
<td>Jitter Modulation</td>
</tr>
<tr>
<td>12.5GHz 4port Synthesizer</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>MU181020A</td>
<td>OPT001</td>
<td>9.8 to 12.5 Gbit/s</td>
</tr>
<tr>
<td>12.5Gbit/s PPG</td>
<td>OPT002</td>
<td>0.1 to 12.5 Gbit/s</td>
</tr>
<tr>
<td></td>
<td>OPTx10</td>
<td>Variable Data Output (0.05 to 0.8 Vp-p)</td>
</tr>
<tr>
<td></td>
<td>OPTx11</td>
<td>Variable Data Output (0.25 to 2.0 Vp-p)</td>
</tr>
<tr>
<td></td>
<td>OPTx12</td>
<td>High Performance Data Output (0.05 to 2.0 Vp-p)</td>
</tr>
<tr>
<td></td>
<td>OPTx13</td>
<td>Variable Data Output (0.5 to 3.5 Vp-p)</td>
</tr>
<tr>
<td></td>
<td>OPTx21</td>
<td>Differential Clock Output (0.1 to 2.0 Vp-p)</td>
</tr>
<tr>
<td></td>
<td>OPTx30</td>
<td>Variable Data Delay</td>
</tr>
<tr>
<td>MU181040A</td>
<td>OPT001</td>
<td>9.8 to 12.5 Gbit/s</td>
</tr>
<tr>
<td>12.5Gbit/s ED</td>
<td>OPT002</td>
<td>0.1 to 12.5 Gbit/s</td>
</tr>
<tr>
<td></td>
<td>OPTx20</td>
<td>Clock Recovery</td>
</tr>
<tr>
<td></td>
<td>OPTx30</td>
<td>Variable Clock Delay</td>
</tr>
<tr>
<td>MU181600A</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>Optical Transceiver (XFP)</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>MU181601A</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>Optical Transceiver (SFP)</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>Model/Name</td>
<td>Option Number</td>
<td>Option Name</td>
</tr>
<tr>
<td>-------------------------------------</td>
<td>---------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>MU181620A Stressed Eye Transmitter</td>
<td>OPTx01</td>
<td>1310nm Reference</td>
</tr>
<tr>
<td></td>
<td>OPTx02</td>
<td>1550nm Reference</td>
</tr>
<tr>
<td></td>
<td>OPTx03</td>
<td>1310nm/1550nm Reference</td>
</tr>
<tr>
<td></td>
<td>OPTx11</td>
<td>1310nm Stressed Eye</td>
</tr>
<tr>
<td></td>
<td>OPTx12</td>
<td>1550nm Stressed Eye</td>
</tr>
<tr>
<td></td>
<td>OPTx13</td>
<td>1310nm/1550nm Stressed Eye</td>
</tr>
<tr>
<td>MU181640A Optical Receiver</td>
<td>OPTx04</td>
<td>Band Width 8.5 GHz</td>
</tr>
<tr>
<td>MU181800A 12.5GHz Clock Distributor</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MU181020B 14Gbit/s PPG</td>
<td>OPTx02</td>
<td>0.1 to 14 Gbit/s</td>
</tr>
<tr>
<td></td>
<td>OPTx11</td>
<td>Variable Data Output (0.25 to 2.5 Vp-p)</td>
</tr>
<tr>
<td></td>
<td>OPTx12</td>
<td>High Performance Data Output (0.05 to 2.0 Vp-p)</td>
</tr>
<tr>
<td></td>
<td>OPTx13</td>
<td>Variable Data Output (0.5 to 3.5 Vp-p)</td>
</tr>
<tr>
<td></td>
<td>OPTx21</td>
<td>Differential Clock Output (0.1 to 2.0 Vp-p)</td>
</tr>
<tr>
<td></td>
<td>OPTx30</td>
<td>Variable Data Delay</td>
</tr>
<tr>
<td>MU181040B 14Gbit/s ED</td>
<td>OPTx02</td>
<td>0.1 to 14 Gbit/s</td>
</tr>
<tr>
<td></td>
<td>OPTx20</td>
<td>Clock Recovery</td>
</tr>
<tr>
<td></td>
<td>OPTx30</td>
<td>Variable Clock Delay</td>
</tr>
<tr>
<td>MU182020A 25Gbit/s 1ch MUX</td>
<td>OPTx01</td>
<td>28Gbit/s Extension</td>
</tr>
<tr>
<td></td>
<td>OPTx10</td>
<td>Variable Data Output (0.25 to 1.75 Vp-p)</td>
</tr>
<tr>
<td></td>
<td>OPTx11</td>
<td>Variable Data Output (0.5 to 2.5 Vp-p)</td>
</tr>
<tr>
<td></td>
<td>OPTx12</td>
<td>Variable Data Output (0.5 to 2.0 Vp-p)</td>
</tr>
<tr>
<td></td>
<td>OPTx13</td>
<td>Variable Data Output (0.5 to 3.5 Vp-p)</td>
</tr>
<tr>
<td></td>
<td>OPTx21</td>
<td>Differential Clock Output (0.5 to 2.0 Vp-p)</td>
</tr>
<tr>
<td></td>
<td>OPTx30</td>
<td>25Gbit/s Variable Data Delay</td>
</tr>
<tr>
<td></td>
<td>OPTx31</td>
<td>28Gbit/s Variable Data Delay</td>
</tr>
<tr>
<td>MU182040A 25Gbit/s 1ch DEMUX</td>
<td>OPTx01</td>
<td>28Gbit/s Extension</td>
</tr>
<tr>
<td></td>
<td>OPTx02</td>
<td>Clock Input Band Switch</td>
</tr>
<tr>
<td></td>
<td>OPTx30</td>
<td>25GHz Variable Clock Delay</td>
</tr>
<tr>
<td></td>
<td>OPTx31</td>
<td>28GHz Variable Clock Delay</td>
</tr>
<tr>
<td>MU183020A 28G/32G bit/s PPG</td>
<td>OPTx01</td>
<td>32G bit/s Extension</td>
</tr>
<tr>
<td></td>
<td>OPTx10</td>
<td>1ch 1V Data Output</td>
</tr>
<tr>
<td></td>
<td>OPTx12</td>
<td>1ch 2V Data Output</td>
</tr>
<tr>
<td></td>
<td>OPTx13</td>
<td>1ch 3.5V Data Output</td>
</tr>
<tr>
<td></td>
<td>OPTx30</td>
<td>1ch Data Delay</td>
</tr>
<tr>
<td></td>
<td>OPTx20</td>
<td>2ch 1V Data Output</td>
</tr>
<tr>
<td></td>
<td>OPTx22</td>
<td>2ch 2V Data Output</td>
</tr>
<tr>
<td></td>
<td>OPTx23</td>
<td>2ch 3.5V Data Output</td>
</tr>
<tr>
<td></td>
<td>OPTx31</td>
<td>2ch Data Delay</td>
</tr>
</tbody>
</table>
## Table 7.12.1.1-2 Option Character Correspondence Table (Cont'd)

<table>
<thead>
<tr>
<th>Model/Name</th>
<th>Option Number</th>
<th>Option Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>MU183021A 28G/32G bit/s 4ch PPG</td>
<td>OPTx01</td>
<td>32G bit/s Extension</td>
</tr>
<tr>
<td></td>
<td>OPTx10</td>
<td>4ch 1V Data Output</td>
</tr>
<tr>
<td></td>
<td>OPTx12</td>
<td>4ch 2V Data Output</td>
</tr>
<tr>
<td></td>
<td>OPTx13</td>
<td>4ch 3.5V Data Output</td>
</tr>
<tr>
<td></td>
<td>OPTx30</td>
<td>4ch Data Delay</td>
</tr>
<tr>
<td>MU183040A 28G/32G bit/s ED</td>
<td>OPTx01</td>
<td>32 Gbit/s Extension</td>
</tr>
<tr>
<td></td>
<td>OPTx10</td>
<td>1ch ED</td>
</tr>
<tr>
<td></td>
<td>OPTx20</td>
<td>2ch ED</td>
</tr>
<tr>
<td>MU183041A 28G/32G bit/s 4ch ED</td>
<td>OPTx01</td>
<td>32 Gbit/s Extension</td>
</tr>
<tr>
<td>MU183040B 28G/32G bit/s High Sensitivity ED</td>
<td>OPTx01</td>
<td>32 Gbit/s Extension</td>
</tr>
<tr>
<td></td>
<td>OPTx10</td>
<td>1ch ED</td>
</tr>
<tr>
<td></td>
<td>OPTx20</td>
<td>2ch ED</td>
</tr>
<tr>
<td>MU183041B 28G/32G bit/s 4ch High Sensitivity ED</td>
<td>OPTx01</td>
<td>32 Gbit/s Extension</td>
</tr>
</tbody>
</table>

**Note:**

The option names of the MU181020B-003 and MU181040B-003 are not output.
Chapter 7  SCPI Commands

:SYSTem:ORGanization:HARDware?

Response  

<slot1>,...<slot64>
<slotx> =

<module>,<serial>,<fpga1>,<fpga2>,<boot>,<application>,<opt>

x indicates a slot number. The slot number varies depending on the unit number as follows.

Unit 1: 1 to 16  
For the MP1800A, numbers from 1 to 6 correspond to actual slots.
For the MT1810A, numbers from 1 to 4 correspond to actual slots.

Unit 2: 17 to 32  
For the MT1810A, numbers from 17 to 20 correspond to actual slots.

Unit 3: 33 to 48  
For the MT1810A, numbers from 33 to 36 correspond to actual slots.

Unit 4: 49 to 64  
For the MT1810A, numbers from 49 to 52 correspond to actual slots.

<module> = <STRING RESPONSE DATA>

XXXXXXXXXX Module name (e.g., MU181020A)
See Table 7.12.1.1-2 “Option Character Correspondence Table”.

Note:

NONE is output if no module is installed.
For a module that uses two slots, only the slot with the greater number is valid.

<serial> = <STRING RESPONSE DATA>

XXXXXXXXXX 0000000000 to 9999999999 Serial number

Note:

“---------” is output if no module is installed.
For a module that uses two slots, only the slot with the greater number is valid.

<fpga1>,<fpga2>,... = <STRING RESPONSE DATA>

XXXX.XX.XX 1.00.00 to 9999.99.99 FPGA version

<boot> = <STRING RESPONSE DATA>

XXXX.XX.XX 1.00.00 to 9999.99.99 Logic Boot version

Note:

“----------” is output if Logic boot is not installed.
For a module that uses two slots, only the slot with the greater number is valid.
<application> = <STRING RESPONSE DATA>
XXXX.XX.XX 1.00.00 to 9999.99.99
Logic Application version

Note:
“--------” is output if Logic Application is not installed.
For a module that uses two slots, only the slot with the greater
number is valid.

<opt> = <STRING RESPONSE DATA>
XXXXXX/XXXXX Option number OPTXXX:
For MP1800A/MT1810A, outputs the numbers
for all installed options.

Note:
Outputs the numbers for all installed options.
For a module that uses two slots, only the slot with the greater
number is valid.
The option names of the MU181020B-003 and MU181040B-003 are
not output.

Function
Queries the hardware system configuration of the MP1800A:
Example
> :SYSTem:ORGanization:HARDware?
< MU181000A,6201234568,1.00.00,1.00.00,1.00.00,OPT101,
MU181020A,6201234569,1.00.00,1.00.00,1.00.00,OPT001,OPT20,
MU181040A,6201234571,1.00.00,1.00.00,1.00.00,OPT002,OPT220

Compatibility
Partially compatible with MU181040A/B.

:SYSTem:INFormation:ERRor? <unit>
Parameter <unit> = <DECIMAL NUMERIC PROGRAM DATA>
1 to 4: 1 to 4, 1step
Response <numeric> = <NR1 NUMERIC RESPONSE DATA>
0 NONE
1 PLL Unlock
2 Temperature
3 Fan
All the system errors that have currently occurred are displayed,
delimited with a comma (,).
Function Queries the System Error contents.
Example
> :SYSTem:INFormation:ERRor? 3
<1,2,3 (when a system error has occurred for PLL Unlock, Temperature,
or Fan)
< 0 (when no system error has occurred)
Compatibility Incompatible with existing models.
Chapter 7  SCPI Commands

:SYSTem:TERMination <numeric>

Parameter

- `<numeric>` = `<DECIMAL NUMERIC PROGRAM DATA>`
  - 0: LF + EOI
  - 1: CR + LF + EOI

Function
Sets the terminator type of the response data.

Example
To set the terminator type to LF + EOI:

```
> :SYSTem:TERMination 0
```

Compatibility
Compatible with the MP1632C.

:SYSTem:TERMination?

Response

- `<numeric>` = `<NR1 NUMERIC RESPONSE DATA>`
  - 0: LF + EOI
  - 1: CR + LF + EOI

Function
Queries the terminator type of the response data.

Example

```
> :SYSTem:TERMination?
< 0
```

Compatibility
Compatible with the MP1632C.
### :SYSTem:CONDition:UNITs?

**Response**

- `<unit1>,..,<unit4>,<slot1>,..,<slot6> =
  "<mainframe1>,..,<mainframe4>,<module1>,..,<module64>"
- `<mainframe1> to <mainframe4> = <STRING RESPONSE DATA>
- XXXXXXXXXX Mainframe name (e.g.: MP1800A)
  See Table 7.12.1.1-2 “Option Character Correspondence Table”.
- `<module1> to <module64> = <STRING RESPONSE DATA>
  XXXXXXXXXX Module name (e.g.: MU181020A)
  See Table 7.12.1.1-2 “Option Character Correspondence Table”.

**Note:**

- NONE is output for mainframe2 to mainframe4, if no mainframe is connected.

- NONE is output if no module is installed.
  For a module that uses two slots, only the slot with the greater number is valid.

**Function**

Queries the model name of the mainframe and module.

**Example**

```bash
> :SYSTem:CONDition:UNITs?
< MT1810A,MT1810A,MT1810A,NONE,MU181000A,NONE,MU181020A,
  MU181040A,NONE,...,MU181600A,MU181601A,MU181620A,NONE,
  MU181640A,NONE,...,NONE
```

**Compatibility**

Incompatible with existing models.
Chapter 7  SCPI Commands

:SYSTem:UNIT? <numeric>

Parameter

<nemonic> = <NR1 NUMERIC PROGRAM DATA>
1 to 4  Mainframe number
“1” for the MP1800A, “1 to 4” for the MT1810A.

Response

<mainframe> =
<unit>,<serial>,<mver>,<hver>,<opt1>,<sbver>,<saver>,<opt2>
<unit> = <STRING RESPONSE DATA>
XXXXXXXXX  Mainframe name (e.g.,: MP1800A)
See Table 7.12.1.1-2 “Option Character Correspondence Table”.

Note:

NONE is output if no module is installed.
For a module that uses two slots, only the slot with the greater number is valid.

<serial> = <STRING RESPONSE DATA>
XXXXXXXXX  0000000000 to 9999999999
Mainframe serial number

Note:

Alphabetic characters may be included.

<mver> = <STRING RESPONSE DATA>
XXXX.XX.XX  1.00.00 to 9999.99.99
Main application software version

<hver> = <STRING RESPONSE DATA>
XXXX.XX.XX  1.00.00 to 9999.99.99
Mainframe hardware version

<opt1> = <STRING RESPONSE DATA>
OPTXXX  Option number (MP1800A/MT1810A)
See Table 7.12.1.1-2 “Option Character Correspondence Table”.

Note:

Outputs the numbers for all installed options.
NONE is output if no option is installed.

<sbver> = <STRING RESPONSE DATA>
XXXX.XX.XX  1.00.00 to 9999.99.99
Sub application software version (Boot part)

<saver> = <STRING RESPONSE DATA>
XXXX.XX.XX  1.00.00 to 9999.99.99
Sub application software version (Application part)

<opt2> = <STRING RESPONSE DATA>

Function

Queries the mainframe information including model and serial number.
Example

```
> :SYSTem:UNIT? 1
<
"MP1800A,6201234568,1.00.00,1.00.00,OPT301,1.00.00,1.00.00"
```

Compatibility

Incompatible with existing models.

:SYSTem:MODule? <numeric>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;numeric&gt;</td>
<td>= &lt;NR1 NUMERIC PROGRAM DATA&gt; Slot 1 to 6 1 to 6 when using the MP1800A, 1 to 4 when using the MT1810A</td>
</tr>
<tr>
<td>Response</td>
<td>&lt;slot&gt; = &lt;module&gt;,&lt;serial&gt;,&lt;fpga1&gt;,&lt;fpga2&gt;,&lt;boot&gt;,&lt;application&gt;,&lt;opt&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;module&gt; = &lt;STRING RESPONSE DATA&gt; XXXXXXXXXX Module name (e.g.,: MU181020A) See Table 7.12.1.1-2 “Option Character Correspondence Table”.</td>
</tr>
<tr>
<td>Note:</td>
<td>NONE is output if no module is installed. For a module that uses two slots, only the slot with the greater number is valid.</td>
</tr>
</tbody>
</table>

| <serial> | = <STRING RESPONSE DATA> XXXXXXXXXX 0000000000 to 9999999999 Serial number |
| Note:    | “---------” is output if no module is installed. For a module that uses two slots, only the slot with the greater number is valid. |

| <fpga1>[,<fpga2>,.....] | = <STRING RESPONSE DATA> XXXX.XX.XX 1.00.00 to 9999.99.99 FPGA version |
| Note:    | “---------” is output if Logic boot is not installed. For a module that uses two slots, only the slot with the greater number is valid. |

| <boot> | = <STRING RESPONSE DATA> XXXX.XX.XX 1.00.00 to 9999.99.99 Logic Boot version |
| Note:    | “---------” is output if Logic boot is not installed. For a module that uses two slots, only the slot with the greater number is valid. |

| <application> | = <STRING RESPONSE DATA> XXXX.XX.XX 1.00.00 to 9999.99.99 Logic Application version |
| Note:    | “---------” is output if Logic Application is not installed. For a module that uses two slots, only the slot with the greater number is valid. |
Chapter 7  SCPI Commands

<opt> = <STRING RESPONSE DATA>
XXXXXXX/XXXXX  Option number
OPTXXX: For MP1800A/MT1810A

Note:
Outputs the numbers for all installed options.
For a module that uses two slots, only the slot with the greater
number is valid.
The option names of the MU181020B-003 and MU181040B-003 are
not output.

Function  Queries the module information on the specified slot.
Example  To query the module information on Slot 3:
> :SYSTem:MODule? 3
<
"MU181020A,6201234568,1.00.00,--------,1.00.00,1.00.00,
OPT002,OPT211,
PT220,"

Compatibility  Incompatible with existing models.

:SYSTem:DISPlay:RESult <boolean>

Parameter  <boolean> = <BOOLEAN PROGRAM DATA>
OFF or 0  Drawing Processing OFF
ON or 1  Drawing Processing ON (default)

Function  Sets measured results drawing processing ON/OFF
Example  To set measured results drawing processing to OFF:
> :SYSTem:DISPlay:RESult OFF

Compatibility  Incompatible with existing models
Remarks  When measured results drawing processing is set to OFF, a dialog
indicating drawing processing is stopped is displayed. Issue the
command to set drawing processing to ON or press the Remote button to
restart measured results drawing processing.

:SYSTem:DISPlay:RESult?

Response  <numeric> = <NR1 NUMERIC RESPONSE DATA>
0  Drawing Processing OFF
1  Drawing Processing ON (default)

Function  Queries drawing processing ON/OFF.
Example  > :SYSTem:DISPlay:RESult?
< 0

Compatibility  Incompatible with existing models.
:DISPlay:ACTive <unit>,<slot>[,<tab>]

Parameter

- `<unit>` = <DECIMAL NUMERIC PROGRAM DATA>
  - 1 to 4: Mainframe No. 1 to 4
    - When using two or more MT1810A units in serial connection, specify the mainframe number.
    - When using the MP1800A, No. 1 is fixed.

- `<slot>` = <DECIMAL NUMERIC PROGRAM DATA>
  - 1 to 6: Slot No. 1 to 6
    - 1 to 6 when using the MP1800A, 1 to 4 when using the MT1810A

- `[<tab>]` = <DECIMAL NUMERIC PROGRAM DATA>
  - 1 to X: Tab ID No. 1 to X
    - When [<,<tab>] is omitted, 1 is specified.
    - Tab ID is set to No. 1 at the left side dialog, and then the tab ID No. is set to 2, 3, 4...toward the right side.
    - The maximum number (X) of the tab ID varies depending on the module options.

<table>
<thead>
<tr>
<th>Tab ID</th>
<th>Tab Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Result</td>
</tr>
<tr>
<td>2</td>
<td>Measurement</td>
</tr>
<tr>
<td>3</td>
<td>Pattern</td>
</tr>
<tr>
<td>4</td>
<td>Input</td>
</tr>
<tr>
<td>5</td>
<td>Capture</td>
</tr>
<tr>
<td>6</td>
<td>Misc 1</td>
</tr>
<tr>
<td>7</td>
<td>Misc 2</td>
</tr>
</tbody>
</table>

Function

- Displays the specified module screen to the front.

**Note:**

- When the screen processing for measurement result is Off, this function cannot be used. If using this function, set the screen processing to On using the :SYSTem:DISPlay:RESult.

Example

To display the Input tab of the MU183040A/41A/40B/41B module:
- (when installing the MU183040A/41A/40B/41B in the unit 1 slot 4)

```
> :DISPlay:ACTive 1,4,4
```

Compatibility

- Incompatible with existing models.
### 7.12.1.2 Commands for File menu

#### Table 7.12.1.2-1 Commands Related to File Menu

<table>
<thead>
<tr>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quick Open</td>
<td>:SYSTem:MMEMory:QRECall</td>
</tr>
<tr>
<td>Quick Save</td>
<td>:SYSTem:MMEMory:QSTore</td>
</tr>
<tr>
<td>Open</td>
<td>:SYSTem:MMEMory:RECall</td>
</tr>
<tr>
<td>Save</td>
<td>:SYSTem:MMEMory:STOREe</td>
</tr>
<tr>
<td>Screen Copy Execute</td>
<td>:SYSTem:PRINt:COPY</td>
</tr>
<tr>
<td>Screen Copy Setup</td>
<td>:SYSTem:PRINt:COPYsetup:SET</td>
</tr>
<tr>
<td></td>
<td>:SYSTem:PRINt:COPYsetup:SET?</td>
</tr>
<tr>
<td>Print</td>
<td>:SYSTem:PRINt:EALarm</td>
</tr>
<tr>
<td>Printer Setup</td>
<td>:SYSTem:PRINt:PRINter:SET</td>
</tr>
<tr>
<td></td>
<td>:SYSTem:PRINt:PRINter:SET?</td>
</tr>
<tr>
<td>Initialize</td>
<td>:SYSTem:MEMory:INITialize</td>
</tr>
</tbody>
</table>

**:SYSTem:MMEMory:QRECall <file_name>**

**Parameter**

\(<file\_name> = \langle STRING PROGRAM DATA \rangle \)

\(<drv>:\backslash[<dir>]\<file>\>

\(<drv> = C, D, E, F\)

\(<dir> = <dir1>\<dir2>\... \text{(Omitted for the root directory)}\)

\(<file> = \text{File name}\)

**Function**

Opens all setting data.

**Example**

To read all setting files from the specified save destination:

\(> :SYSTem:MMEMory:QRECall "C:\Test\example"\)

**Compatibility**

Commands are compatible with the MP1632C.

Parameters are incompatible.
### :SYSTem:MMEMory:QSTore <file_name>,<comment>

**Parameter**
- `<file_name>` = `<STRING PROGRAM DATA>
  
  "<drv>:\[<dir>]<file>"
- `<drv>` = C, D, E, F
- `<dir>` = `<dir1>:<dir2>\... (Omitted for the root directory)
- `<file>` = File name
- `<comment>` = `<STRING PROGRAM DATA>
  "XXXXXX..." Specify a comment of a character string within 60 characters into the file.

**Function**
Executes "Quick Save".

**Note:**
The settings will not be read from the saved file if the file name is changed.

**Example**
To specify save destination for all setting files and save them with a comment and measurement result data:

> :SYSTem:MMEMory:QSTore "C:\Test\example", "setup all"

**Compatibility**
Commands are compatible with the MP1632C.
Parameters are incompatible.

### :SYSTem:MMEMory:RECall <file_name>,<module>,<data_type>

**Parameter**
- `<file_name>` = `<STRING PROGRAM DATA>
  "<drv>:\[<dir>]<file>"
- `<drv>` = C, D, E, F
- `<dir>` = `<dir1>:<dir2>\... (Omitted for the root directory)
- `<file>` = File name
- `<module>` = `<STRING PROGRAM DATA>
  "<unit>:<slot>:<port> <module>"
- `<unit>` = 1, 2, 3, 4
- `<slot>` = 1, 2, 3, 4,...6
- `<port>` = 1
- `<module>` = Module model name
- `<data_type>` = `<CHARACTER PROGRAM DATA>
  - P32 28G/32G PPG Setup
  - P32X 28G/32G 4ch PPG Setup
  - E32 28G/32G ED Setup
  - E32X 28G/32G 4ch ED Setup
  - CP32 28G/32G Capture Result

**Function**
Opens module settings and result data.

**Example**
To read a file for the ED module:

> :SYSTem:MMEMory:RECall "C:\Test\example", "1:3:1 MU183040A", E32

**Compatibility**
Compatible with the MP1632C1 and MP1776A commands.
Parameters are incompatible.
Chapter 7 SCPI Commands

:SYSTem:MMEMory:STORe
<file_name>,<module>,<data_type>,<file_type>

Parameter
<file_name> = <STRING PROGRAM DATA>
"<drv>:\[<dir>]<file>"
<drv> = C, D, E, F
<dir> = <dir1>\<dir2>\... (Omitted for the root directory)
<file> = File name
/module> = <STRING PROGRAM DATA>
"<unit>:<slot>:<port> <module>"
<unit> = 1, 2, 3, 4
[slot] = 1, 2, 3, 4...,6
<port> = 1
<module> = Module model name
<data_type> = <CHARACTER PROGRAM DATA>
P32 28G/32G PPG Setup
P32X 28G/32G 4ch PPG Setup
E32 28G/32G ED Setup
E32X 28G/32G 4ch ED Setup
CP32 28G/32G Capture Result
ERA_D1 32G ED Error/Alarm Result (Data1)
ERA_D2 32G ED Error/Alarm Result (Data2)
ERA_D3 32G ED Error/Alarm Result (Data3)
ERA_D4 32G ED Error/Alarm Result (Data4)
ERA_C1 32G ED Error/Alarm Result (2ch Combi1-2)
ERA_C2 32G ED Error/Alarm Result (2ch Combi3-4)
ERA_C4 32G ED Error/Alarm Result (4ch Combi1-4)
<file_type> = <CHARACTER PROGRAM DATA>
BIN Binary File
CSV CSV File
TXT Text File

Function
Saves the setting and measurement result data for a specified module.

Note:
The settings will not be read from the saved file if the file name is changed.

Example
To save the setting file for the ED module:
> :SYSTem:MMEMory:STORe "C:\Test\example", "1:3:1 MU183040A",E32,BIN

Compatibility
Compatible with the MU181040A/B commands.
Parameters are incompatible.
7.12 28G/32G bit/s ED Commands

:SYSTem:PRINT:COPY

Function  Takes a screen shot.
Example   > :SYSTem:PRINT:COPY
Compatibility Compatible with MU181040A/B.

:SYSTem:PRINT:COPYsetup:SET <save_type>,<output>,<save_to>

Parameter  
<save_type> = <CHARACTER PROGRAM DATA>
BMP         Bitmap format
PNG         PNG format
JPG         JPG format
<output> = <CHARACTER PROGRAM DATA>
FILE        Output to a file.
PRINter     Output to the printer.
<save_to> = <STRING PROGRAM DATA>
"<drv>:\[<dir>]"
<drv> = C, D, E, F
<dir> = <dir1>\<dir2>\... (Omitted for the root directory)

Function  Sets the extension, output destination, and save destination of the screen shot.
Example   To output the screen shot to the specified file in the bitmap format:
> :SYSTem:PRINT:COPYsetup:SET BMP,FILE,"C:\Test\example"
Compatibility Incompatible with existing models

:SYSTem:PRINT:COPYsetup:SET?

Response  <save_type> = <CHARACTER RESPONSE DATA>
BMP        Bitmap format
PNG        PNG format
JPG        JPG format
<output> = <CHARACTER RESPONSE DATA>
FILE       Output to a file.
PRINter    Output to the printer.
<save_to> = <STRING RESPONSE DATA>
"<drv>:\[<dir>]"
<drv> = C, D, E, F
<dir> = <dir1>\<dir2>\... (Omitted for the root directory)

Function  Queries the extension, output destination, and save destination settings of the screen shot.
Example   > :SYSTem:PRINT:COPYsetup:SET?
< BMP,FILE,"C:\Test\example"
Compatibility Incompatible with existing models
Chapter 7  SCPI Commands

:SYSTem:PRINT:EALarm <list>,<module>

Parameter

- <list> = <CHARACTER PROGRAM DATA>
- ERA_D1 32G ED Error/Alarm Result (Data1)
- ERA_D2 32G ED Error/Alarm Result (Data2)
- ERA_D3 32G ED Error/Alarm Result (Data3)
- ERA_D4 32G ED Error/Alarm Result (Data4)
- ERA_C1 32G ED Error/Alarm Result (2ch Combi1-2)
- ERA_C2 32G ED Error/Alarm Result (2ch Combi3-4)
- ERA_C4 32G ED Error/Alarm Result (4ch Combi1-4)

- <module> = <STRING PROGRAM DATA>
  "<unit>:<slot>:<port>:<module>"
  - <unit> = 1, 2, 3, 4
  - <slot> = 1, 2, 3, 4,...,6
  - <port> = 1,2...
  - <module> = Module name

Function
Prints the Error/Alarm results.

Example
To print the 2ch Combi3-4 measurement result:

Compatibility
Compatible with the MU181040A/B commands.
Parameters are incompatible.

:SYSTem:PRINT:PRINTER:SET <printer>,<direction>

Parameter

- <printer> = <STRING PROGRAM DATA>
  "XXXXXXX..." Printer name
- <direction> = <CHARACTER PROGRAM DATA>
  VER Vertical
  HOR Horizontal

Function
Sets the printer and print direction.

Example
> :SYSTem:PRINT:PRINTER:SET "printer",VER

Compatibility
Incompatible with existing models

:SYSTem:PRINT:PRINTER:SET?

Response

- <printer> = <STRING RESPONSE DATA>
  "XXXXXXX..." Printer name
- <direction> = <CHARACTER RESPONSE DATA>
  VER Vertical
  HOR Horizontal

Function
Queries the default printer and print direction.

Example
> :SYSTem:PRINT:PRINTER:SET?
< "printer",VER

Compatibility
Incompatible with existing models
:SYSTem:MEMory:INITialize

**Function**
Initializes the internal setting data to the initial settings at factory shipment.

**Example**
> :SYSTem:MEMory:INITialize

**Compatibility**
Compatible with the MP1632C and MP1776A.
7.12.1.3 Common function buttons

Table 7.12.1.3-1 Common Function Button Setting Commands

<table>
<thead>
<tr>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>Meas. Start</td>
<td>:SENSe:MEASure:ASTRt</td>
</tr>
<tr>
<td>Meas. Stop</td>
<td>:SENSe:MEASure:ASTP</td>
</tr>
<tr>
<td>(Query for measurement status)</td>
<td>:SENSe:MEASure:ASTate?</td>
</tr>
<tr>
<td>Open</td>
<td>Refer to 7.12.1.2 “Commands for File menu”.</td>
</tr>
<tr>
<td>Save</td>
<td>Refer to 7.12.1.2 “Commands for File menu”.</td>
</tr>
<tr>
<td>Print</td>
<td>Refer to 7.12.1.2 “Commands for File menu”.</td>
</tr>
<tr>
<td>Error ON/OFF</td>
<td>:SYSTem:BEEPer:ERRor:SET</td>
</tr>
<tr>
<td></td>
<td>:SYSTem:BEEPer:ERRor:SET?</td>
</tr>
<tr>
<td>Alarm ON/OFF</td>
<td>:SYSTem:BEEPer:ALARm:SET</td>
</tr>
<tr>
<td></td>
<td>:SYSTem:BEEPer:ALARm:SET?</td>
</tr>
<tr>
<td>System Error ON/OFF</td>
<td>:SYSTem:BEEPer:SYSTem:SET</td>
</tr>
<tr>
<td></td>
<td>:SYSTem:BEEPer:SYSTem:SET?</td>
</tr>
<tr>
<td>System Error</td>
<td>:SYSTem:BEEPer:SYSTem:TYPE</td>
</tr>
<tr>
<td></td>
<td>:SYSTem:BEEPer:SYSTem:TYPE?</td>
</tr>
</tbody>
</table>

:SENSe:MEASure:ASTRt

Function: Starts measurement for all modules.
Example: > :SENSe:MEASure:ASTRt
Compatibility: Incompatible with existing models

:SENSe:MEASure:ASTP

Function: Stops measurement for all modules.
Example: > :SENSe:MEASure:ASTP
Compatibility: Incompatible with existing models
**:SENSe:MEASure:ASTate?**

**Response**
- `<numeric> = <NR1 NUMERIC RESPONSE DATA>`
- 0: Measurement stops for all modules.
- 1: During measurement

**Note:**
- If any module is being measured, "1 (During measurement)" is returned.

**Function**
- Queries the measurement state for all modules.

**Example**
- `> :SENSe:MEASure:ASTate?`
- `< 0`

**Compatibility**
- Incompatible with existing models

---

**:SYSTem:BEEPer:ERRor:SET <boolean>**

**Parameter**
- `<boolean> = <BOOLEAN PROGRAM DATA>`
- OFF or 0: Buzzer OFF
- ON or 1: Buzzer ON

**Function**
- Sets buzzer at error occurrence ON/OFF.

**Example**
- To set buzzer at error occurrence ON:
  - `> :SYSTem:BEEPer:ERRor:SET ON`

**Compatibility**
- Compatible with MU181040A/B.

---

**:SYSTem:BEEPer:ERRor:SET?**

**Response**
- `<numeric> = <NR1 NUMERIC RESPONSE DATA>`
- 0: Buzzer OFF
- 1: Buzzer ON

**Function**
- Queries the buzzer ON/OFF state at error occurrence.

**Example**
- `> :SYSTem:BEEPer:ERRor:SET?`
- `< 1`

**Compatibility**
- Compatible with MU181040A/B.

---

**:SYSTem:BEEPer:ALARm:SET <boolean>**

**Parameter**
- `<boolean> = <BOOLEAN PROGRAM DATA>`
- OFF or 0: Buzzer OFF
- ON or 1: Buzzer ON

**Function**
- Sets buzzer at alarm occurrence ON/OFF.

**Example**
- To set buzzer at alarm occurrence OFF:
  - `> :SYSTem:BEEPer:ALARm:SET OFF`

**Compatibility**
- Compatible with MU181040A/B.
Chapter 7  SCPI Commands

:SYSTem:BEEPer:ALARm:SET?

Response  
<numeric> = <NR1 NUMERIC RESPONSE DATA>
0     Buzzer OFF
1     Buzzer ON

Function  Queries the buzzer ON/OFF state at alarm occurrence.
Example  > :SYSTem:BEEPer:ALARm:SET?
< 0

Compatibility  Compatible with MU181040A/B.

:SYSTem:BEEPer:SYSTem:SET <boolean>

Parameter  
<boolean> = <BOOLEAN PROGRAM DATA>
OFF or 0  Buzzer OFF
ON or 1    Buzzer ON

Function  Sets buzzer at system error occurrence ON/OFF.
Example  To set buzzer at system error occurrence ON:
> :SYSTem:BEEPer:SYSTem:SET ON

Compatibility  Compatible with MU181040A/B.

:SYSTem:BEEPer:SYSTem:SET?

Response  
<numeric> = <NR1 NUMERIC RESPONSE DATA>
0     Buzzer OFF
1     Buzzer ON

Function  Queries the buzzer ON/OFF state at system error occurrence.
Example  > :SYSTem:BEEPer:SYSTem:SET?
< 1

Compatibility  Compatible with MU181040A/B.

:SYSTem:BEEPer:SYSTem:TYPE <type>,<boolean>

Parameter  
<type> = <CHARACTER PROGRAM DATA>
PUNLock     PLL Unlock
FAN         FAN
TEMPerature Temperature
ALL         Selects all system errors
<boolean> = <BOOLEAN PROGRAM DATA>
OFF or 0  Buzzer OFF
ON or 1    Buzzer ON

Function  Sets system error buzzer for the target item ON/OFF.
Example  To set system error buzzer for "Temperature" ON:
> :SYSTem:BEEPer:SYSTem:TYPE TEMPerature,ON

Compatibility  Partially compatible with the MP1632C.
### :SYSTem:BEEPer:SYSTem:TYPE?

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;type&gt; = &lt;CHARACTER RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PUNL, FAN, TEMP, ALL</td>
</tr>
<tr>
<td></td>
<td>XXX, XXX</td>
</tr>
<tr>
<td></td>
<td>Errors for which buzzer is set to ON are delimited with commas (,) and returned.</td>
</tr>
<tr>
<td></td>
<td>NONE</td>
</tr>
<tr>
<td></td>
<td>Buzzer is set to OFF for all items.</td>
</tr>
</tbody>
</table>

#### Response

- **XXX, XXX**: Errors for which buzzer is set to ON are delimited with commas (,) and returned.
- **NONE**: Buzzer is set to OFF for all items.

#### Function

Queries the ON/OFF state of system error buzzer for target items.

#### Example

```
> :SYSTem:BEEPer:SYSTem:TYPE?
< PUNL, TEMP
```

#### Compatibility

Compatible with the MP1632C.
7.12.1.4 ED status

The ED status shows alarms and errors in MU183040A/41A/40B/41B. There is no compatibility of commands with existing Anritsu EDs.

**:INStrument:EDG32[:EVENt]?**

<table>
<thead>
<tr>
<th>Available bit</th>
<th>Description</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 (Bit0)</td>
<td>Error (Data1)</td>
<td>&lt;numeric&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt;</td>
</tr>
<tr>
<td>2 (Bit1)</td>
<td>Error (Data2)</td>
<td>0 to 16383 Sum total of the bit of event register (DECIMAL)</td>
</tr>
<tr>
<td>4 (Bit2)</td>
<td>Error (Data3)</td>
<td>1</td>
</tr>
<tr>
<td>8 (Bit3)</td>
<td>Error (Data4)</td>
<td>2</td>
</tr>
<tr>
<td>16 (Bit4)</td>
<td>Pattern Sync Loss (Data1)</td>
<td>4</td>
</tr>
<tr>
<td>32 (Bit5)</td>
<td>Pattern Sync Loss (Data2)</td>
<td>8</td>
</tr>
<tr>
<td>64 (Bit6)</td>
<td>Pattern Sync Loss (Data3)</td>
<td>16</td>
</tr>
<tr>
<td>128 (Bit7)</td>
<td>Pattern Sync Loss (Data4)</td>
<td>32</td>
</tr>
<tr>
<td>256 (Bit8)</td>
<td>Clock Loss</td>
<td>64</td>
</tr>
<tr>
<td>1024 (Bit10)</td>
<td>Delay Calibration Require (Data1)</td>
<td>128</td>
</tr>
<tr>
<td>2048 (Bit11)</td>
<td>Delay Calibration Require (Data2)</td>
<td>256</td>
</tr>
<tr>
<td>4096 (Bit12)</td>
<td>Delay Calibration Require (Data3)</td>
<td>512</td>
</tr>
<tr>
<td>8192 (Bit13)</td>
<td>Delay Calibration Require (Data4)</td>
<td>1024</td>
</tr>
</tbody>
</table>

Function

Queries event at 28G/32G bit/s ED status.

Example

```
> :INSTRument:EDG32:EVENt?

or

> :INSTRument:EDG32?

< 1
```
7.12 28G/32G bit/s ED Commands

### :INSTrument:EDG32:CONDition?

<table>
<thead>
<tr>
<th>Response</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;numeric&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt;</td>
<td>Sum total of the bit of condition register (DECIMAL)</td>
</tr>
<tr>
<td>0 to 16383</td>
<td>Available bit</td>
</tr>
<tr>
<td>1 (Bit0)</td>
<td>Error (Data1)</td>
</tr>
<tr>
<td>2 (Bit1)</td>
<td>Error (Data2)</td>
</tr>
<tr>
<td>4 (Bit2)</td>
<td>Error (Data3)</td>
</tr>
<tr>
<td>8 (Bit3)</td>
<td>Error (Data4)</td>
</tr>
<tr>
<td>16 (Bit4)</td>
<td>Pattern Sync Loss (Data1)</td>
</tr>
<tr>
<td>32 (Bit5)</td>
<td>Pattern Sync Loss (Data2)</td>
</tr>
<tr>
<td>64 (Bit6)</td>
<td>Pattern Sync Loss (Data3)</td>
</tr>
<tr>
<td>128 (Bit7)</td>
<td>Pattern Sync Loss (Data4)</td>
</tr>
<tr>
<td>256 (Bit8)</td>
<td>Clock Loss</td>
</tr>
<tr>
<td>1024 (Bit10)</td>
<td>Delay Calibration Require (Data1)</td>
</tr>
<tr>
<td>2048 (Bit11)</td>
<td>Delay Calibration Require (Data2)</td>
</tr>
<tr>
<td>4096 (Bit12)</td>
<td>Delay Calibration Require (Data3)</td>
</tr>
<tr>
<td>8192 (Bit13)</td>
<td>Delay Calibration Require (Data4)</td>
</tr>
</tbody>
</table>

**Function**: Queries condition at 28G/32G bit/s ED status.

**Example**

```
> :INSTrument:EDG32:CONDition?
< 1
```
### :INSTrument:EDG32:PTRansition <numeric>

**Parameter**

<table>
<thead>
<tr>
<th>&lt;numeric&gt;</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 to 16383</td>
<td>Sum total of the bit of transition filter (DECIMAL)</td>
</tr>
</tbody>
</table>

**Available bit**

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 (Bit0)</td>
<td>Error (Data1)</td>
</tr>
<tr>
<td>2 (Bit1)</td>
<td>Error (Data2)</td>
</tr>
<tr>
<td>4 (Bit2)</td>
<td>Error (Data3)</td>
</tr>
<tr>
<td>8 (Bit3)</td>
<td>Error (Data4)</td>
</tr>
<tr>
<td>16 (Bit4)</td>
<td>Pattern Sync Loss (Data1)</td>
</tr>
<tr>
<td>32 (Bit5)</td>
<td>Pattern Sync Loss (Data2)</td>
</tr>
<tr>
<td>64 (Bit6)</td>
<td>Pattern Sync Loss (Data3)</td>
</tr>
<tr>
<td>128 (Bit7)</td>
<td>Pattern Sync Loss (Data4)</td>
</tr>
<tr>
<td>256 (Bit8)</td>
<td>Clock Loss</td>
</tr>
<tr>
<td>1024 (Bit10)</td>
<td>Delay Calibration Require (Data1)</td>
</tr>
<tr>
<td>2048 (Bit11)</td>
<td>Delay Calibration Require (Data2)</td>
</tr>
<tr>
<td>4096 (Bit12)</td>
<td>Delay Calibration Require (Data3)</td>
</tr>
<tr>
<td>8192 (Bit13)</td>
<td>Delay Calibration Require (Data4)</td>
</tr>
</tbody>
</table>

**Function**

Sets transition filter (positive direction transition) at 28G/32G bit/s ED status.

**Example**

To set the transition filter (positive direction transition) at 28G/32G bit/s ED status to 1:

> :INSTrument:EDG32:PTRansition 1

### :INSTrument:EDG32:PTRansition?

**Response**

<table>
<thead>
<tr>
<th>&lt;numeric&gt;</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 to 16383</td>
<td>Sum total of the bit of transition filter (DECIMAL)</td>
</tr>
</tbody>
</table>

**Function**

Queries contents of transition filter (positive direction transition) at 28G/32G bit/s ED status.

**Example**

> :INSTrument:EDG32:PTRansition?

< 1
:INSTRument:EDG32:NTRansition <numeric>

Parameter

- <numeric> = <DECIMAL NUMERIC PROGRAM DATA>
- 0 to 16383 Sum total of the bit of transition filter (DECIMAL)

Available bit

- 1 (Bit0) Error (Data1)
- 2 (Bit1) Error (Data2)
- 4 (Bit2) Error (Data3)
- 8 (Bit3) Error (Data4)
- 16 (Bit4) Pattern Sync Loss (Data1)
- 32 (Bit5) Pattern Sync Loss (Data2)
- 64 (Bit6) Pattern Sync Loss (Data3)
- 128 (Bit7) Pattern Sync Loss (Data4)
- 256 (Bit8) Clock Loss
- 1024 (Bit10) Delay Calibration Require (Data1)
- 2048 (Bit11) Delay Calibration Require (Data2)
- 4096 (Bit12) Delay Calibration Require (Data3)
- 8192 (Bit13) Delay Calibration Require (Data4)

Function

Sets transition filter (negative direction transition) at 28G/32G bit/s ED status.

Example

To set the transition filter (negative direction transition) at 28G/32G bit/s ED status to 1:

> :INSTRument:EDG32:NTRansition 1

:INSTRument:EDG32:NTRansition?

Response

- <numeric> = <NR1 NUMERIC RESPONSE DATA>
- 0 to 16383 Sum total of the bit of transition filter (DECIMAL)

Function

Queries contents of transition filter (negative direction transition) at 28G/32G bit/s ED status.

Example

> :INSTRument:EDG32:NTRansition?

< 1

:INSTRument:EDG32:RESSet

Function

Initializes event at 28G/32G bit/s ED status.

Example

> :INSTRument:EDG32:RESSet
7.12.1.5 Data Interface setting commands

Table 7.12.1.5-1 Data Interface Setting Commands

<table>
<thead>
<tr>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Interface Select</td>
<td>:INTerface:ID</td>
</tr>
<tr>
<td></td>
<td>:INTerface:ID?</td>
</tr>
</tbody>
</table>

**:INTerface:ID <interface_number>**

Parameter: 

<interface_number> = <DECIMAL NUMERIC PROGRAM DATA>
1 to 4 
Data interface number

Function: 

Sets the number of data interface to be operated (Data 1 to 4). The setting range varies according to the module model.

- MU183040A/B-x10: 1
- MU183040A/B-x20: 1 to 2
- MU183041A/B: 1 to 4

Example: 

To set the number of data interface to be operated to 4:

```
> :INTerface:ID 4
```

Compatibility: 

Incompatible with existing models

**:INTerface:ID?**

Parameter: 

<interface_number> = <NR1 NUMERIC RESPONSE DATA>
1 to 4

Function: 

Queries the number of the data interface being operated (Data 1 to 4).

Example: 

```
> :INTerface:ID?
< 4
```

Compatibility: 

Incompatible with existing models
7.12.1.6 Data Interface setting commands

This section describes the commands for saving and loading pattern files to MU183040A/41A/40B/41B. Before executing the commands, use the :MODule:ID and :INTerface:ID commands to specify both slot number and data interface number of the module you want to remote-control. For how to specify the slot number with the :MODule:ID command, refer to 7.1, “Common Commands” and 9.1, “Basic Operation of Device Messages”.

Table 7.12.1.6-1 Pattern Editor Setting Commands

<table>
<thead>
<tr>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>Open</td>
<td>:SYSTem:MMEMory:PATTern:RECall</td>
</tr>
<tr>
<td>Save</td>
<td>:SYSTem:MMEMory:PATTern:STORe</td>
</tr>
</tbody>
</table>

:SYSTem:MMEMory:PATTern:RECall <file_name>,<file_type>

Parameter

- `<file_name>` = <STRING PROGRAM DATA>
- `<drv>`:\<dir>\<file>`
- `<drv>` = C, D, E, F
- `<dir>` = <dir1>\<dir2>\...(Omitted when root directory)
- `<file>` = File name
- `<file_type>` = <CHARACTER PROGRAM DATA>
  - BIN Binary File
  - TXT Text File

Function

Opens the pattern file.

Example

To load the pattern file, specifying the save location (C:\Test), file name (example) and file format (Binary):

```
> :SYSTem:MMEMory:PATTern:RECall "C:\Test\example",BIN
```

Compatibility

Incompatible with existing models.
### :SYSTem:MMEMory:PATTern:STORe "<file_name>"","<file_type>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;file_name&gt;</code></td>
<td>&lt;STRING PROGRAM DATA&gt;</td>
</tr>
<tr>
<td><code>&lt;drv&gt;</code></td>
<td>C, D, E, F</td>
</tr>
<tr>
<td><code>&lt;dir&gt;</code></td>
<td><code>&lt;dir1&gt;\&lt;dir2&gt;\...(Omitted when root directory)</code></td>
</tr>
<tr>
<td><code>&lt;file&gt;</code></td>
<td>File name</td>
</tr>
<tr>
<td><code>&lt;file_type&gt;</code></td>
<td>&lt;CHARACTER PROGRAM DATA&gt;</td>
</tr>
<tr>
<td>BIN</td>
<td>Binary File</td>
</tr>
<tr>
<td>TXT</td>
<td>Text File</td>
</tr>
</tbody>
</table>

**Function**

Saves the pattern file.

**Note:**

The settings will not be read from the saved file if the file name is changed.

**Example**

To save the pattern file, specifying the save location (C:\Test), file name (example) and file format (Text):

```
> :SYSTem:MMEMory:PATTern:STORe "C:\Test\example",TXT
```

**Compatibility**

Incompatible with existing models.
7.12.2 Commands Related to Result Tab

![Result Tab]

Figure 7.12.2-1  Result Tab
7.12.2.1 Result setting commands

![Start/Stop Settings](image1)

![Result Settings](image2)

Table 7.12.2.1-1 Result Setting Commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1]</td>
<td>Start</td>
<td>:SENSe:MEASure:STARt</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:DISPlay:RESult:COMBination?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:DISPlay:RESult:TIME?</td>
</tr>
<tr>
<td>[6]</td>
<td>Query for measurement start time</td>
<td>:SENSe:MEASure:EALarm:STARt?</td>
</tr>
<tr>
<td>[7]</td>
<td>Query for measurement stop time</td>
<td>:SENSe:MEASure:EALarm:STOP?</td>
</tr>
<tr>
<td>[8]</td>
<td>Query for measurement elapsed time</td>
<td>:SENSe:MEASure:EALarm:ELAPsed?</td>
</tr>
</tbody>
</table>
7.12 28G/32G bit/s ED Commands

:SENSe:MEASure:STARt
Function Starts BER measurement for the specified module. If this command is issued during measurement, the BER measurement is restarted.
Example > :SENSe:MEASure:STARt
Compatibility Compatible with MU181040A/B.

:SENSe:MEASure:STOP
Function Stops the BER measurement of the specified module.
Example > :SENSe:MEASure:STOP
Compatibility Compatible with MU181040A/B.

:SENSe:MEASure:EALarm:STATe?
Response <numeric> = <NR1 NUMERIC RESPONSE DATA>
0 Measurement stops
1 During measurement
Function Queries the measurement processing state during bit error alarm measurement.
Example > :SENSe:MEASure:EALarm:STATe?
< 0
Compatibility Compatible with MU181040A/B.

:DISPlay:RESult:COMBination <item>
Parameter <item> = <DECIMAL NUMERIC PROGRAM DATA>
1 Independent
21 Displays the result of 2ch Combination1-2.
22 Displays the result of 2ch Combination3-4.
23 Displays the result of 2ch Combination Data1/2 and Data3/4 simultaneously.
4 4ch Combination
Function Sets the combination to be displayed.
Example To set Independent as the combination to be displayed:
> :DISPlay:RESult:COMBination 1
Compatibility Incompatible with existing models
Chapter 7  SCPI Commands

:DISPlay:RESult:COMBination?

Response

<table>
<thead>
<tr>
<th>&lt;item&gt;</th>
<th>Numerical Response Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Independent</td>
</tr>
<tr>
<td>21</td>
<td>2ch Combination 1-2</td>
</tr>
<tr>
<td>22</td>
<td>2ch Combination 3-4</td>
</tr>
<tr>
<td>23</td>
<td>Simultaneous display of 2ch Combination Data 1/2 and Data 3/4.</td>
</tr>
<tr>
<td>4</td>
<td>4ch Combination</td>
</tr>
</tbody>
</table>

Function
Queries the status of the displayed combination.

Example

> :DISPlay:RESult:COMBination?
< 1

Compatibility
Incompatible with existing models.

:DISPlay:RESult:TIME <time>

Parameter

<table>
<thead>
<tr>
<th>&lt;time&gt;</th>
<th>Character Program Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>DTIMe</td>
<td>Displays the current date and time.</td>
</tr>
<tr>
<td>STARt</td>
<td>Displays the measurement start time.</td>
</tr>
<tr>
<td>ELAPsed</td>
<td>Displays the elapsed time based on the measurement period.</td>
</tr>
<tr>
<td>TIMed</td>
<td>Displays the remaining time based on the measurement period.</td>
</tr>
</tbody>
</table>

Function
Sets the measurement time display type.

Example

To set the measurement time display type to DTIMe (current date and time):

> :DISPlay:RESult:TIME DTIMe

Compatibility
Compatible with MU181040A/B.

:DISPlay:RESult:TIME?

Response

<time> = Character Response Data
DTIM, STAR, ELAP, TIM

Function
Queries the measurement time display type.

Example

> :DISPlay:RESult:TIME?
< DTIM

Compatibility
Compatible with MU181040A/B.
### :SENSe:MEASure:EALarm:STARt?

**Response**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;year&gt;</td>
<td>0, 2000 to 2036</td>
</tr>
<tr>
<td>&lt;month&gt;</td>
<td>0, 1 to 12</td>
</tr>
<tr>
<td>&lt;day&gt;</td>
<td>0, 1 to 31</td>
</tr>
<tr>
<td>&lt;hour&gt;</td>
<td>0 to 23</td>
</tr>
<tr>
<td>&lt;min&gt;</td>
<td>0 to 59</td>
</tr>
<tr>
<td>&lt;second&gt;</td>
<td>0 to 59</td>
</tr>
</tbody>
</table>

- **<year>** 0, 2000 to 2036  Year 2000 to 2036
- **<month>** 0, 1 to 12  January to December
- **<day>** 0, 1 to 31  1st to 31st
- **<hour>** 0 to 23  0 to 23 hours
- **<min>** 0 to 59  0 to 59 minutes
- **<second>** 0 to 59  0 to 59 seconds

“0,0,0,0,0,0” is returned if there is no measurement start time data.

**Function**

Queries the measurement start time during bit error alarm measurement.

**Example**

```
> :SENSe:MEASure:EALarm:STARt?
< 2012,1,1,23,59,59
```

**Compatibility**

Compatible with MU181040A/B.

---

### :SENSe:MEASure:EALarm:STOP?

**Response**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;year&gt;</td>
<td>0, 2000 to 2036</td>
</tr>
<tr>
<td>&lt;month&gt;</td>
<td>0, 1 to 12</td>
</tr>
<tr>
<td>&lt;day&gt;</td>
<td>0, 1 to 31</td>
</tr>
<tr>
<td>&lt;hour&gt;</td>
<td>0 to 23</td>
</tr>
<tr>
<td>&lt;min&gt;</td>
<td>0 to 59</td>
</tr>
<tr>
<td>&lt;second&gt;</td>
<td>0 to 59</td>
</tr>
</tbody>
</table>

- **<year>** 0, 2000 to 2036  Year 2000 to 2036
- **<month>** 0, 1 to 12  January to December
- **<day>** 0, 1 to 31  1st to 31st
- **<hour>** 0 to 23  0 to 23 hours
- **<min>** 0 to 59  0 to 59 minutes
- **<second>** 0 to 59  0 to 59 seconds

“0,0,0,0,0,0” is returned if there is no measurement start time data.

**Function**

Queries the measurement end time during bit error alarm measurement.

**Example**

```
> :SENSe:MEASure:EALarm:STOP?
< 0,0,0,0,0,0
```

**Compatibility**

Compatible with MU181040A/B.
Chapter 7  SCPI Commands

:SENSe:MEASure:EALarm:ELAPsed?
Response  
<day>,<hour>,<min>,<second> = <NR1 NUMERIC RESPONSE DATA>
<day>  0 to 31  0 to 31 days
<hour>  0 to 23  0 to 23 hours
<min>  0 to 59  0 to 59 minutes
<second>  0 to 59  0 to 59 seconds

Function  Queries the measurement elapsed time during bit error alarm measurement.

Example  > :SENSe:MEASure:EALarm:ELAPsed?
< 0,1,0,0

Compatibility  Compatible with MU181040A/B.

:SENSe:MEASure:EALarm:TIMed?
Response  
<day>,<hour>,<min>,<second> = <NR1 NUMERIC RESPONSE DATA>
<day>  0 to 31  0 to 31 days
<hour>  0 to 23  0 to 23 hours
<min>  0 to 59  0 to 59 minutes
<second>  0 to 59  0 to 59 seconds

“0,0,0,0” is returned when the measurement is finished.

When measurement is not performed, the measurement period set at that time is returned.

Function  Queries the measurement remaining time during bit error alarm measurement.

Example  > :SENSe:MEASure:EALarm:TIMed?
< 0,0,0,1

Compatibility  Compatible with MU181040A/B.

:SENSe:MEASure:EALarm:ITIMe?
Response  
<year>,<month>,<day>,<hour>,<min>,<second> = <NR1 NUMERIC RESPONSE DATA>
<year>  0, 2000 to 2036  Year 2000 to 2036
<month>  0, 1 to 12  January to December
<day>  0, 1 to 31  1st to 31st
<hour>  0 to 23  0 to 23 hours
<min>  0 to 59  0 to 59 minutes
<second>  0 to 59  0 to 59 seconds

“0,0,0,0,0,0” is returned if there is no measurement intermediate data.

Function  Queries the measurement intermediate data creation time during bit error alarm measurement.

Example  > :SENSe:MEASure:EALarm:ITIMe?
< 0,0,0,0,0,0
Compatibility

Compatible with MU181040A/B.

:SENSe:MEASure:EALarm:AOCCur? <alarm>

Parameter

<alarm> = <CHARACTER PROGRAM DATA>
CLOS  Clock Loss
PSL   Pattern Sync Loss

Response

<y,mo,da,hr,mi,se> = <NR1 NUMERIC RESPONSE DATA>
<y> 0, 2000 to 2036  Year 2000 to 2036
<mo> 0, 1 to 12    January to December
<da> 0, 1 to 31    1st to 31st
<hr> 0 to 23        0 to 23 hours
<mi> 0 to 59        0 to 59 minutes
<se> 0 to 59        0 to 59 seconds

“0,0,0,0,0,0” is returned if no alarm has occurred.

Function

Queries the last time when a specified alarm has occurred during bit error alarm measurement.

Example

To query the time when the last clock loss has occurred during bit error alarm measurement:

> :SENSe:MEASure:EALarm:AOCCur? CLOS
< 0,0,0,1,0,0

Compatibility

Partially compatible with the MP1632C (only the clock loss and pattern sync loss are compatible).

:SENSe:DISPlay:SETTing <setting>

Parameter

<setting> = <CHARACTER PROGRAM DATA>
INPut Displays Input items
GATing Displays Gating items
CONDition Displays Condition items
AUTosync Displays AutoSync items
SYNCcontrol Displays SyncControl items

Function

Sets display items in the Result tab

Example

To display the Input items:

> :SENSe:DISPlay:SETTing INPut

Compatibility

Compatible with MU181040A/B.
7.12.2.2 Error/Alarm setting commands

Figure 7.12.2.2-1 Error/Alarm Result Display Area

Figure 7.12.2.2-2 Error/Alarm Result Display Area (When Zoom Is Clicked)
Table 7.12.2-1  Error/Alarm Setting Commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>:DISPlay:RESult:ZOOM?</td>
</tr>
<tr>
<td>[3]</td>
<td>No label (Query for error alarm measurement data)</td>
<td>:CALCulate:DATA:EALarm?</td>
</tr>
<tr>
<td>[4]</td>
<td>No label (Query for monitoring item)</td>
<td>:CALCulate:DATA:MONitor?</td>
</tr>
<tr>
<td>[5]</td>
<td>No label (Query for Combination measurement data)</td>
<td>:CALCulate:CDATa:EALarm?</td>
</tr>
<tr>
<td>[6]</td>
<td>No label (Query for Combination monitoring item)</td>
<td>:CALCulate:CDATa:MONitor?</td>
</tr>
</tbody>
</table>

**:DISPlay:RESult:ZOOM <boolean>**

Parameter  
<boolean> = <BOOLEAN PROGRAM DATA>

0 or OFF  
Display enlargement OFF

1 or ON  
Display enlargement ON

Function  
Sets display enlargement for the measurement result display screen ON/OFF.

Example  
To set enlargement for the measurement result display screen ON:

> :DISPlay:RESult:ZOOM ON

Compatibility  
Compatible with MU181040A/B.

**:DISPlay:RESult:ZOOM?**

Response  
<numERIC> = <NR1 NUMERIC RESPONSE DATA>

0  
Display enlargement OFF

1  
Display enlargement ON

Function  
Queries the display enlargement for the measurement result display screen ON/OFF state.

Example  
> :DISPlay:RESult:ZOOM?

< 1

Compatibility  
Compatible with MU181040A/B.

**:DISPlay:RESult:EALarm:HRESet**

Function  
Resets the histories on the measurement result display screen.

Example  
> :DISPlay:RESult:EALarm:HRESet

Compatibility  
Compatible with MU181040A/B.
Chapter 7  SCPI Commands

:CALCulate:DATA:EALarm? <string>

Parameter

<string> = <STRING PROGRAM DATA>
"CURRent:<result>"  Current data
"LAST:<result>"  Measurement end data
See Table 7.12.2.2-3 for details on <result>.

Response

<string> = <STRING RESPONSE DATA>

Table 7.12.2.2-2  Response Format

<table>
<thead>
<tr>
<th>Form</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Form1</td>
<td>&quot;XXXXXXX&quot;</td>
<td>For 0 to 9999999</td>
</tr>
<tr>
<td>Integer type</td>
<td>&quot;X.XXXXXXX&quot;</td>
<td>For 1.0000E07 to 9.9999E17</td>
</tr>
<tr>
<td></td>
<td>&quot;........&quot;</td>
<td>When no data corresponds to the query</td>
</tr>
<tr>
<td>Form2</td>
<td>&quot;X.XXXE-XX&quot;</td>
<td>For 0.0001E-18 to 1.0000E00</td>
</tr>
<tr>
<td>Fraction type</td>
<td>&quot;........&quot;</td>
<td>When no data corresponds to the query</td>
</tr>
<tr>
<td>Form3</td>
<td>&quot;XXX.XXXX&quot;</td>
<td>For 0.0000 to 100.0000</td>
</tr>
<tr>
<td>% type</td>
<td>&quot;........&quot;</td>
<td>When no data corresponds to the query</td>
</tr>
<tr>
<td>Form4</td>
<td>&quot;XXXXXXXXXX&quot;</td>
<td>For 0 to MAX (Hz)</td>
</tr>
<tr>
<td>Frequency type</td>
<td>&quot;........&quot;</td>
<td>When no data corresponds to the query</td>
</tr>
<tr>
<td>Form5</td>
<td>&quot;X.XX, XX.XX&quot;</td>
<td>For -4.000 to + 4.000</td>
</tr>
<tr>
<td>Voltage type</td>
<td>&quot;........&quot;, &quot;........&quot;</td>
<td>When no data corresponds to the query</td>
</tr>
<tr>
<td>Form6</td>
<td>&quot;XXXX, XXXX&quot;</td>
<td>For -1000 to +1000, -10000 to +10000</td>
</tr>
<tr>
<td>Phase type</td>
<td>&quot;........&quot;, &quot;........&quot;</td>
<td>When no data corresponds to the query</td>
</tr>
</tbody>
</table>

Function

Queries the measurement data that corresponds to the parameter.

Example

To query the total error rate measurement data for the current data:

> :CALCulate:DATA:EALarm? "CURRent:ER:TOTal"

< "0.0000E-16"

Compatibility

Compatible with MU181040A/B.
### Table 7.12.2.2-3 Parameters

<table>
<thead>
<tr>
<th>Items</th>
<th>&lt;result&gt;</th>
<th>Form</th>
</tr>
</thead>
<tbody>
<tr>
<td>Error Rate</td>
<td>Total ER:TOTal</td>
<td>Form2</td>
</tr>
<tr>
<td>INS</td>
<td>ER:INSertion</td>
<td></td>
</tr>
<tr>
<td>OMI</td>
<td>ER:OMISsion</td>
<td></td>
</tr>
<tr>
<td>Transition</td>
<td>ER:TRANsition</td>
<td></td>
</tr>
<tr>
<td>Non Transition</td>
<td>ER:NONTransition</td>
<td></td>
</tr>
<tr>
<td>Error Count</td>
<td>Total EC:TOTal</td>
<td>Form1</td>
</tr>
<tr>
<td>INS</td>
<td>EC:INSertion</td>
<td></td>
</tr>
<tr>
<td>OMI</td>
<td>EC:OMISsion</td>
<td></td>
</tr>
<tr>
<td>Transition</td>
<td>EC:TRANsition</td>
<td></td>
</tr>
<tr>
<td>Non Transition</td>
<td>EC:NONTransition</td>
<td></td>
</tr>
<tr>
<td>%EFI</td>
<td>Total EFI:TOTal</td>
<td>Form3</td>
</tr>
<tr>
<td>EI</td>
<td>EI:TOTal</td>
<td>Form1</td>
</tr>
<tr>
<td>Frequency(kHz)</td>
<td>FREQuency</td>
<td>Form4</td>
</tr>
<tr>
<td>Clock Count</td>
<td>CC:TOTal</td>
<td>Form1</td>
</tr>
<tr>
<td>Clock Loss</td>
<td>AINTerval:CLOSs</td>
<td>Form1</td>
</tr>
<tr>
<td>Sync Loss</td>
<td>AINTerval:PSLoss</td>
<td>Form1</td>
</tr>
<tr>
<td>Data Threshold</td>
<td>THReshold</td>
<td>Form5</td>
</tr>
<tr>
<td>Data Phase</td>
<td>PHASe</td>
<td>Form6</td>
</tr>
</tbody>
</table>
Chapter 7  SCPI Commands

:CALCulate:DATA:MONitor? <item>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;item&gt; = &lt;STRING PROGRAM DATA&gt;</td>
<td></td>
</tr>
<tr>
<td>&quot;BIT:TOTal&quot;</td>
<td>Bit Error (Total Error)</td>
</tr>
<tr>
<td>&quot;CLOSSs&quot;</td>
<td>Clock Loss</td>
</tr>
<tr>
<td>&quot;PSLoss&quot;</td>
<td>Pattern Sync Loss</td>
</tr>
</tbody>
</table>

Response

<STRING RESPONSE DATA>

Table 7.12.2.2-4  Response

<table>
<thead>
<tr>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;Occur&quot;</td>
<td>When an alarm occurs</td>
</tr>
<tr>
<td>&quot;Not Occur&quot;</td>
<td>When no alarm occurs</td>
</tr>
<tr>
<td>&quot;------&quot;</td>
<td>When no data corresponds to the query</td>
</tr>
</tbody>
</table>

Function

Queries the real-time occurrence status of the monitoring item corresponding to the parameter.

Example

To query the bit error occurrence status:

> :CALCulate:DATA:MONitor? "BIT:TOTal"
< "Occur"

Compatibility

Compatible with MU181040A/B.
### :CALCulate:CDATa:EALarm? <string>[,<item>]

**Parameter**

- `<string>` = `<STRING PROGRAM DATA>`
- "CURRent:<result>" Current data
- "LAST:<result>" Measurement end data

See Table 7.12.2.2-6 for details on `<result>`.

- `<item>` = `<DECIMAL NUMERIC PROGRAM DATA>`
  - 21 Result of 2ch Combination Data 1/2.
  - 22 Result of 2ch Combination Data 3/4.
  - 4 Result of 4ch Combination.

**Note:** This can be omitted. When `<item>` is omitted at 2ch Combination, the result of Data1/2 is queried.

**Response**

- `<string>` = `<STRING RESPONSE DATA>`

<table>
<thead>
<tr>
<th>Form</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Form1</td>
<td>&quot;XXXXXXX&quot;</td>
<td>For 0 to 9999999</td>
</tr>
<tr>
<td>Integer type</td>
<td>&quot;X.XXXEEXX&quot;</td>
<td>For 1.0000E07 to 9.9999E17</td>
</tr>
<tr>
<td></td>
<td>&quot;-------&quot;</td>
<td>When no data corresponds to the query</td>
</tr>
<tr>
<td>Form2</td>
<td>&quot;X.XXXXXE-XX&quot;</td>
<td>For 0.0001E-18 to 1.0000E00</td>
</tr>
<tr>
<td>Fraction type</td>
<td>&quot;-------&quot;</td>
<td>When no data corresponds to the query</td>
</tr>
<tr>
<td>Form3</td>
<td>&quot;XXX.XXXXX&quot;</td>
<td>For 0.0000 to 100.0000</td>
</tr>
<tr>
<td>% type</td>
<td>&quot;-------&quot;</td>
<td>When no data corresponds to the query</td>
</tr>
<tr>
<td>Form4</td>
<td>&quot;XXXXXXXXXX&quot;</td>
<td>For 0 to MAX (Hz)</td>
</tr>
<tr>
<td>Frequency type</td>
<td>&quot;-------&quot;</td>
<td>When no data corresponds to the query</td>
</tr>
<tr>
<td>Form5</td>
<td>&quot;X.XXX, XX.XXX&quot;</td>
<td>For −4.000 to +4.000</td>
</tr>
<tr>
<td>Voltage type</td>
<td>&quot;-------, -------&quot;</td>
<td>When no data corresponds to the query</td>
</tr>
<tr>
<td>Form6</td>
<td>&quot;XXXX, XXXX&quot;</td>
<td>For −1000 to +1000, −10000 to +10000</td>
</tr>
<tr>
<td>Phase type</td>
<td>&quot;-------, -------&quot;</td>
<td>When no data corresponds to the query</td>
</tr>
</tbody>
</table>

**Function**: Queries the combination measurement data corresponding to Parameter.

**Example**: To query the measurement data which totaled the error rate of current data:

```
> :CALCulate:CDATa:EALarm? "CURRent:ER:TOTal"
< "0.0000E-16"
```

**Compatibility**: Incompatible with existing models
### Table 7.12.2.2-6 Parameters

<table>
<thead>
<tr>
<th>Items</th>
<th>&lt;result&gt;</th>
<th>Form</th>
</tr>
</thead>
<tbody>
<tr>
<td>Error Rate</td>
<td>Total</td>
<td>ER:TOTal</td>
</tr>
<tr>
<td></td>
<td>INS</td>
<td>ER:INSertion</td>
</tr>
<tr>
<td></td>
<td>OMI</td>
<td>ER:OMISsion</td>
</tr>
<tr>
<td>Error Count</td>
<td>Total</td>
<td>EC:TOTal</td>
</tr>
<tr>
<td></td>
<td>INS</td>
<td>EC:INSertion</td>
</tr>
<tr>
<td></td>
<td>OMI</td>
<td>EC:OMISsion</td>
</tr>
<tr>
<td>%EFI</td>
<td>Total</td>
<td>EFI:TOTal</td>
</tr>
<tr>
<td>EI</td>
<td>Total</td>
<td>EI:TOTal</td>
</tr>
<tr>
<td>Frequency(kHz)</td>
<td>FREQuency</td>
<td>Form4</td>
</tr>
<tr>
<td>Clock Count</td>
<td>CC:TOTal</td>
<td>Form1</td>
</tr>
<tr>
<td>Clock Loss</td>
<td>AINTerval:CLOSs</td>
<td>Form1</td>
</tr>
<tr>
<td>Sync Loss</td>
<td>AINTerval:PSLoss</td>
<td>Form1</td>
</tr>
<tr>
<td>Data Threshold</td>
<td>THReshold</td>
<td>Form5</td>
</tr>
<tr>
<td>Data Phase</td>
<td>PHASe</td>
<td>Form6</td>
</tr>
</tbody>
</table>
:CALCulate:CDATa:MONitor? <string>[,<item>]

Parameter

<string> = <STRING PROGRAM DATA>
"BIT:TOTal"  Bit Error(Total Error)
"CLOSs"  Clock Loss
"PSLoss"  Pattern Sync Loss

=item> = <DECIMAL NUMERIC PROGRAM DATA>
21  Result of 2ch Combination Data 1/2.
22  Result of 2ch Combination Data 3/4.
4  Result of 4ch Combination.

Note:
This can be omitted. Result of Data 1/2 is queried when [<item>] is omitted at 2ch Combination.

Response

<string> = <STRING RESPONSE DATA>

Table 7.12.2.2-7  Response

<table>
<thead>
<tr>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;Occur&quot;</td>
<td>When an alarm occurs</td>
</tr>
<tr>
<td>&quot;Not Occur&quot;</td>
<td>When no alarm occurs</td>
</tr>
<tr>
<td>&quot;------&quot;</td>
<td>When no data corresponds to the query</td>
</tr>
</tbody>
</table>

Function
Queries the real-time occurring state of the monitor item of the combination corresponding to Parameter

Example
To query the occurring state of Bit Error:
> :CALCulate:CDATa:MONitor? "BIT:TOTal"
< "Occur"

Compatibility
Incompatible with existing models
7.12.3 Commands Related to Measurement Tab

![Figure 7.12.3-1  Measurement Tab](image)

Figure 7.12.3-1  Measurement Tab
7.12.3.1 Gating setting commands

Figure 7.12.3.1-1  Gating Setting (Time)

Figure 7.12.3.1-2  Gating Setting (Clock Count)

Figure 7.12.3.1-3  Gating Setting (Error Count)

Figure 7.12.3.1-4  Gating Setting (Block Count)
### Gating Setting Commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1]</td>
<td>Unit</td>
<td>:SENSe:MEASure:EALarm:UNIT</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:MEASure:EALarm:UNIT?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:MEASure:EALarm:PERiod?</td>
</tr>
<tr>
<td>[3]</td>
<td>No label (Clock count)</td>
<td>:SENSe:MEASure:EALarm:CLOCkcnt</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:MEASure:EALarm:CLOCkcnt?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:MEASure:EALarm:ERRorcnt?</td>
</tr>
<tr>
<td>[5]</td>
<td>No label (Block count)</td>
<td>:SENSe:MEASure:EALarm:BLOCkcnt</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:MEASure:EALarm:BLOCkcnt?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:MEASure:EALarm:MODE?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:DISPlay:RESult:EALarm:MODE?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:DISPlay:RESult:EALarm:CALCuration?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:DISPlay:RESult:EALarm:INTerval?</td>
</tr>
</tbody>
</table>

### :SENSe:MEASure:EALarm:UNIT <unit>

**Parameter**

<unit> = <CHARACTER PROGRAM DATA>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TIME</td>
<td>Time</td>
</tr>
<tr>
<td>CLOCk</td>
<td>Clock Count</td>
</tr>
<tr>
<td>ERRor</td>
<td>Error Count</td>
</tr>
<tr>
<td>BLOCk</td>
<td>Block Count</td>
</tr>
</tbody>
</table>

**Function**

Sets the unit of measurement cycle.

**Example**

To set the unit of measurement cycle to Time:

> :SENSe:MEASure:EALarm:UNIT TIME

**Compatibility**

Compatible with MU181040A/B.

### :SENSe:MEASure:EALarm:UNIT? (Response)

Response

<unit> = <CHARACTER RESPONSE DATA>

<table>
<thead>
<tr>
<th>Response</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>TIME</td>
<td>CLOC, ERR, BLOC</td>
</tr>
</tbody>
</table>

**Function**

Queries the unit of measurement cycle.

**Example**

To query the unit of measurement cycle:

> :SENSe:MEASure:EALarm:UNIT?

< TIME

**Compatibility**

Compatible with MU181040A/B.
:SENSe:MEASure:EALarm:PERiod <day>,<hour>,<min>,<second>

**Parameter**

- `<day>`, `<hour>`, `<min>`, `<second>` = `<DECIMAL NUMERIC PROGRAM DATA>`
  - `<day>`: 0 to 99 0 to 99 days/day Step
  - `<hour>`: 0 to 23 0 to 23 hours/hour Step
  - `<min>`: 0 to 59 0 to 59 minutes/minute Step
  - `<second>`: 0 to 59 0 to 59 seconds/second Step

**Function**
Sets the measurement time when the measurement cycle setting is Time.

**Example**
To set the measurement time to “0 day 0 hour 1 min 0 second” when the measurement cycle setting is Time:

> :SENSe:MEASure:EALarm:PERiod 0,0,1,0

**Compatibility**
Compatible with MU181040A/B.

:SENSe:MEASure:EALarm:PERiod?

**Response**

- `<day>`, `<hour>`, `<min>`, `<second>` = `<NR1 NUMERIC RESPONSE DATA>`
  - `<day>`: 0 to 99 0 to 99 days
  - `<hour>`: 0 to 23 0 to 23 hours
  - `<min>`: 0 to 59 0 to 59 minutes
  - `<second>`: 0 to 59 0 to 59 seconds

**Function**
Queries the measurement time when the measurement cycle setting is Time.

**Example**

> :SENSe:MEASure:EALarm:PERiod?

< 0,0,1,0

**Compatibility**
Compatible with MU181040A/B.
### :SENSe:MEASure:EALarm:CLOCkcnt <count>

**Parameter**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;count&gt;</td>
<td>= &lt;CHARACTER PROGRAM DATA&gt;</td>
</tr>
<tr>
<td>E_4</td>
<td>&gt;1 E + 4</td>
</tr>
<tr>
<td>E_5</td>
<td>&gt;1 E + 5</td>
</tr>
<tr>
<td>E_6</td>
<td>&gt;1 E + 6</td>
</tr>
<tr>
<td>E_7</td>
<td>&gt;1 E + 7</td>
</tr>
<tr>
<td>E_8</td>
<td>&gt;1 E + 8</td>
</tr>
<tr>
<td>E_9</td>
<td>&gt;1 E + 9</td>
</tr>
<tr>
<td>E_10</td>
<td>&gt;1 E + 10</td>
</tr>
<tr>
<td>E_11</td>
<td>&gt;1 E + 11</td>
</tr>
<tr>
<td>E_12</td>
<td>&gt;1 E + 12</td>
</tr>
<tr>
<td>E_13</td>
<td>&gt;1 E + 13</td>
</tr>
<tr>
<td>E_14</td>
<td>&gt;1 E + 14</td>
</tr>
<tr>
<td>E_15</td>
<td>&gt;1 E + 15</td>
</tr>
<tr>
<td>E_16</td>
<td>&gt;1 E + 16</td>
</tr>
</tbody>
</table>

**Function**

Sets the threshold for the number of clocks when the measurement cycle setting is Clock Count.

**Example**

To set the threshold for the number of clocks to 1 E + 4 when the measurement cycle setting is Clock Count:

> :SENSe:MEASure:EALarm:CLOCkcnt E_4

**Compatibility**

Compatible with MU181040A/B.

### :SENSe:MEASure:EALarm:CLOCkcnt?

**Response**

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;count&gt; = &lt;CHARACTER RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>E_4, E_5, E_6, E_7, E_8, E_9, E_10, E_11, E_12, E_13, E_14, E_15, E_16</td>
<td></td>
</tr>
</tbody>
</table>

**Function**

Queries the threshold for the number of clocks when the measurement cycle setting is Clock Count.

**Example**

> :SENSe:MEASure:EALarm:CLOCkcnt?

< E_4

**Compatibility**

Compatible with MU181040A/B.
:SENSe:MEASure:EALarm:ERRorcnt <count>

Parameter

<table>
<thead>
<tr>
<th>&lt;count&gt;</th>
<th>= &lt;CHARACTER PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>E_4</td>
<td>&gt;1 E + 4</td>
</tr>
<tr>
<td>E_5</td>
<td>&gt;1 E + 5</td>
</tr>
<tr>
<td>E_6</td>
<td>&gt;1 E + 6</td>
</tr>
<tr>
<td>E_7</td>
<td>&gt;1 E + 7</td>
</tr>
<tr>
<td>E_8</td>
<td>&gt;1 E + 8</td>
</tr>
<tr>
<td>E_9</td>
<td>&gt;1 E + 9</td>
</tr>
<tr>
<td>E_10</td>
<td>&gt;1 E + 10</td>
</tr>
<tr>
<td>E_11</td>
<td>&gt;1 E + 11</td>
</tr>
<tr>
<td>E_12</td>
<td>&gt;1 E + 12</td>
</tr>
<tr>
<td>E_13</td>
<td>&gt;1 E + 13</td>
</tr>
<tr>
<td>E_14</td>
<td>&gt;1 E + 14</td>
</tr>
<tr>
<td>E_15</td>
<td>&gt;1 E + 15</td>
</tr>
<tr>
<td>E_16</td>
<td>&gt;1 E + 16</td>
</tr>
</tbody>
</table>

Function
Sets the threshold for the number of errors when the measurement cycle setting is Error Count.

Example
To set the threshold for the number of errors to 1 E + 4 when the measurement cycle setting is Error Count:

> :SENSe:MEASure:EALarm:ERRorcnt E_4

Compatibility
Compatible with MU181040A/B.

:SENSe:MEASure:EALarm:ERRorcnt?

Response

<count> = <CHARACTER RESPONSE DATA>

| E_4, E_5, E_6, E_7, E_8, E_9, E_10, E_11, E_12, E_13, E_14, E_15, E_16 |

Function
Queries the threshold for the number of errors when the measurement cycle setting is Error Count.

Example
To query the threshold for the number of errors when the measurement cycle setting is Error Count:

> :SENSe:MEASure:EALarm:ERRorcnt?

< E_4

Compatibility
Compatible with MU181040A/B.
**:SENSe:MEASure:EALarm:BLOCkcnt <count>**

**Parameter**  
\(<\text{count}> = \langle\text{CHARACTER PROGRAM DATA}\rangle\)  
- E_2 >1 E + 2  
- E_3 >1 E + 3  
- E_4 >1 E + 4  
- E_5 >1 E + 5  
- E_6 >1 E + 6  
- E_7 >1 E + 7  
- E_8 >1 E + 8  
- E_9 >1 E + 9  
- E_10 >1 E + 10  
- E_11 >1 E + 11  
- E_12 >1 E + 12  
- E_13 >1 E + 13  
- E_14 >1 E + 14

**Function**  
Sets the threshold for the number of blocks when the measurement cycle setting is Block Count.

**Example**  
To set the threshold for the number of blocks to 1 E + 4 when the measurement cycle setting is Block Count:

> :SENSe:MEASure:EALarm:BLOCkcnt E_4

**Compatibility**  
Compatible with MU181040A/B.

**:SENSe:MEASure:EALarm:BLOCkcnt?**

**Response**  
\(<\text{count}> = \langle\text{CHARACTER RESPONSE DATA}\rangle\)  
E_2, E_3, E_4, E_5, E_6, E_7, E_8, E_9, E_10, E_11, E_12, E_13, E_14

**Function**  
Queries the threshold for the number of blocks when the measurement cycle setting is Block Count.

**Example**  
> :SENSe:MEASure:EALarm:BLOCkcnt?

< E_4

**Compatibility**  
Compatible with MU181040A/B.

**:SENSe:MEASure:EALarm:MODE <mode>**

**Parameter**  
\(<\text{mode}> = \langle\text{CHARACTER PROGRAM DATA}\rangle\)  
- REPeat  
  Executes the measurement repeatedly.  
- SINGle  
  Executes a single measurement.  
- UNTImed  
  Executes the measurement until the measurement end is instructed.

**Function**  
Sets the measurement processing mode for the bit-error/alarm measurement.

**Example**  
To set the measurement processing mode for the bit-error/alarm measurement to Repeat:
7.12 28G/32G bit/s ED Commands

>:SENSe:MEASure:EALarm:MODE REPeat

Compatibility

Compatible with MU181040A/B.

:SENSe:MEASure:EALarm:MODE?

Response

<mode> = <CHARACTER RESPONSE DATA>
REp, SING, UNT

Function
Queries the measurement processing mode for the bit-error/alarm measurement.

Example

>:SENSe:MEASure:EALarm:MODE?
< REp

Compatibility

Compatible with MU181040A/B.

:DISPlay:RESult:EALarm:MODE <boolean>

Parameter

<boolean> = <BOOLEAN PROGRAM DATA>
0 or OFF
1 or ON

Function
Sets whether to turn on or off the actual value update of the measurement result data display.

Example

To turn on the actual value update of the measurement result data display:
>:DISPlay:RESult:EALarm:MODE ON

Compatibility

Compatible with MU181040A/B.

:DISPlay:RESult:EALarm:MODE?

Response

<numeric> = <NR1 NUMERIC RESPONSE DATA>
0 OFF
1 ON

Function
Queries whether the actual value update of the measurement result data display is on.

Example

>:DISPlay:RESult:EALarm:MODE?
< 1

Compatibility

Compatible with MU181040A/B.
### :DISPlay:RESult:EALarm:CALCuration <mode>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;mode&gt; = CHARACTER PROGRAM DATA</td>
<td></td>
</tr>
<tr>
<td>PROGressive</td>
<td>Cumulative processing.</td>
</tr>
<tr>
<td>IMMEDIATE</td>
<td>Immediate processing.</td>
</tr>
</tbody>
</table>

**Function**
Sets the method of calculating the immediate data.

**Example**
To set the method of calculating the immediate data to Cumulative processing:

```
> :DISPlay:RESult:EALarm:CALCuration PROGressive
```

**Compatibility**
Compatible with MU181040A/B.

### :DISPlay:RESult:EALarm:CALCuration? 

| Response     | CHARACTER RESPONSE DATA: PROG, IMM |

**Function**
Queries the method of calculating the immediate data.

**Example**

```
> :DISPlay:RESult:EALarm:CALCuration?
< PROG
```

**Compatibility**
Compatible with MU181040A/B.

### :DISPlay:RESult:EALarm:INTerval <interval>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>CHARACTER PROGRAM DATA: I100, I200, I500</th>
</tr>
</thead>
<tbody>
<tr>
<td>I100</td>
<td>Updates display in 100-ms period.</td>
</tr>
<tr>
<td>I200</td>
<td>Updates display in 200-ms period.</td>
</tr>
<tr>
<td>I500</td>
<td>Updates display in 500-ms period.</td>
</tr>
</tbody>
</table>

**Function**
Sets the update cycle of the immediate data display.

**Example**
To set the update cycle of the immediate data display to 100 ms:

```
> :DISPlay:RESult:EALarm:INTerval I100
```

**Compatibility**
Compatible with MU181040A/B.

### :DISPlay:RESult:EALarm:INTerval? 

| Response     | CHARACTER RESPONSE DATA: I100, I200, I500 |

**Function**
Queries the update cycle of the immediate data display.

**Example**

```
> :DISPlay:RESult:EALarm:INTerval?
< I100
```

**Compatibility**
Compatible with MU181040A/B.
7.12.3.2 Auto Sync setting commands

![Auto Sync Setting](image)

Figure 7.12.3.2-1 Auto Sync Setting

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>:SENSe:PATTern:SYNC:ASYNc?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:PATTern:SYNC:THReshold?</td>
</tr>
</tbody>
</table>

:SENSe:PATTern:SYNC:ASYNc <boolean>

Parameter

- `<boolean>` = `<BOOLEAN PROGRAM DATA>`
- 0 or OFF: Auto Sync OFF
- 1 or ON: Auto Sync ON

Function

Sets whether or not to automatically retry the synchronization when the synchronization threshold is exceeded.

Example

To set to automatically retry the synchronization:

> :SENSe:PATTern:SYNC:ASYNc ON

Compatibility

Compatible with MU181040A/B.

:SENSe:PATTern:SYNC:ASYNc?

Response

- `<numeric>` = `<NR1 NUMERIC RESPONSE DATA>`
- 0: Auto Sync OFF
- 1: Auto Sync ON

Function

Queries whether the synchronization is automatically retried or not when the synchronization threshold is exceeded.

Example

> :SENSe:PATTern:SYNC:ASYNc?

< 1

Compatibility

Compatible with MU181040A/B.
Chapter 7  SCPI Commands

:SENSe:PATTern:SYNC:THReshold <thre>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>(&lt;thre&gt; = &lt;CHARACTER PROGRAM DATA&gt;)</th>
</tr>
</thead>
<tbody>
<tr>
<td>INT</td>
<td>Internal</td>
</tr>
<tr>
<td>E_2</td>
<td>E–2</td>
</tr>
<tr>
<td>E_3</td>
<td>E–3</td>
</tr>
<tr>
<td>E_4</td>
<td>E–4</td>
</tr>
<tr>
<td>E_5</td>
<td>E–5</td>
</tr>
<tr>
<td>E_6</td>
<td>E–6</td>
</tr>
<tr>
<td>E_7</td>
<td>E–7</td>
</tr>
<tr>
<td>E_8</td>
<td>E–8</td>
</tr>
</tbody>
</table>

Function: Sets the synchronization threshold for determining whether or not to retry the synchronization.

Example: To set the synchronization threshold for determining whether or not to retry the synchronization to Internal:

\[> :SENSe:PATTern:SYNC:THReshold INT\]

Compatibility: Compatible with MU181040A/B.

:SENSe:PATTern:SYNC:THReshold?

<table>
<thead>
<tr>
<th>Response</th>
<th>(&lt;thre&gt; = &lt;CHARACTER RESPONSE DATA&gt;)</th>
</tr>
</thead>
<tbody>
<tr>
<td>INT, E_2, E_3, E_4, E_5, E_6, E_7, E_8</td>
<td></td>
</tr>
</tbody>
</table>

Function: Queries the synchronization threshold for determining whether or not to retry the synchronization.

Example:

\[> :SENSe:PATTern:SYNC:THReshold?\]
\[< INT\]

Compatibility: Compatible with MU181040A/B.
7.12.3.3  Sync Control setting commands

Figure 7.12.3.3-1  Sync Control Setting

Figure 7.12.3.3-2  Pattern Editor Dialog Box
### Table 7.12.3.3-1  Sync Control Setting Commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>:SENSe:PATTern:SYNC:PSMode?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:PATTern:SYNC:FLENgth?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:PATTern:SYNC:FPOSition?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:PATTern:SYNC:DATA:WHOLe?</td>
</tr>
<tr>
<td>[6]</td>
<td>No label (All 0 or 1)</td>
<td>:SENSe:PATTern:SYNC:PROGram:FILL</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:PATTern:SYNC:ACORder?</td>
</tr>
</tbody>
</table>

**:SENSe:PATTern:SYNC:PSMode <mode>**

Parameter: 

- `<mode> = <CHARACTER PROGRAM DATA>`
- **FRAMe**: Frame detection ON
- **QUICk**: Quick synchronization control

Function: Sets the synchronization control of the test pattern.

Example: To set the synchronization control of the test pattern to Frame detection ON:

> :SENSe:PATTern:SYNC:PSMode FRAMe

Compatibility: Partially compatible with MU181040A/B.

**:SENSe:PATTern:SYNC:PSMode?**

Response: 

- `<mode> = <CHARACTER RESPONSE DATA>`
- **FRAM, QUIC**

Function: Queries the synchronization control of the test pattern.

Example: 

> :SENSe:PATTern:SYNC:PSMode?

< FRAM

Compatibility: Compatible with MU181040A/B.
:SENSe:PATtern:SYNC:FLENgth <length>

Parameter

\(<\text{length}\> = \text{<DECIMAL NUMERIC PROGRAM DATA>}

4 to 64

4 to 64 bits/4 bits Step

At 2ch Combination, the maximum value and step are doubled.

At 4ch Combination, the maximum value and step are quadrupled.

Function

Sets the unique pattern length for the synchronization control with Frame detection ON.

Example

To set the unique pattern length for the synchronization control with Frame detection ON:

\(> :\text{SENSe}\text{;}\text{PATtern}\text{;}\text{SYNC}\text{;}\text{FLENgth} \ 4\)

Compatibility

Partially compatible with MU181040A/B.

:SENSe:PATtern:SYNC:FLENgth?

Response

\(<\text{length}\> = \text{<NR1 NUMERIC RESPONSE DATA>}

Function

Queries the unique pattern length for the synchronization control with Frame detection ON.

Example

\(> :\text{SENSe}\text{;}\text{PATtern}\text{;}\text{SYNC}\text{;}\text{FLENgth}?\)

\(< \ 4\)

Compatibility

Partially compatible with MU181040A/B.

:SENSe:PATtern:SYNC:FPOSition <length>

Parameter

\(<\text{length}\> = \text{<DECIMAL NUMERIC PROGRAM DATA>}

1 to 268,435,453

1 to 268,435,453 bits/1 bit Step

Allowable parameter setting: Pattern length - Frame length + 1 (bits)

At 2ch Combination Setting,

1 to 536,870,905

1 to 536,870,905 bits/2 bit Step

At 4ch Combination Setting,

1 to 1,073,741,809

1 to 1,073,741,809 bits/4 bit Step

Function

Sets the frame synchronization start bit for the synchronization control with Frame detection ON.

Example

To set the frame synchronization start bit for the synchronization control with Frame detection ON to 65 bits:

\(> :\text{SENSe}\text{;}\text{PATtern}\text{;}\text{SYNC}\text{;}\text{FPOSition} \ 65\)

Compatibility

Partially compatible with MU181040A/B.
Chapter 7  SCPI Commands

:SENSe:PATTern:SYNC:FPOSition?

Response  
<length> = <NR1 NUMERIC RESPONSE DATA>

Function  
Queries the frame synchronization start bit for the synchronization control with Frame detection ON.

Example  
> :SENSe:PATTern:SYNC:FPOSition?
< 17

Compatibility  
Partially compatible with MU181040A/B.

:SENSe:PATTern:SYNC:DATA:WHOLe <data>

Parameter  
<data> = <STRING PROGRAM DATA>
"H*...*" Specifies the pattern data consisting of 1 to 16 hexadecimal characters.
"B*...*" Specifies the pattern data consisting of 1 to 64 binary characters.

Function  
Edits the mask pattern for the synchronization control with Frame detection ON.
Edits the data of up to 64 bits in HEX or BIN.
At 2ch Combination, the maximum value and step are doubled.
At 4ch Combination, the maximum value and step are quadrupled.

Example  
To set the mask pattern for the synchronization control with Frame detection ON to a hexadecimal character:
> :SENSe:PATTern:SYNC:DATA:WHOLe "H001"

Compatibility  
Compatible with MU181040A/B.

:SENSe:PATTern:SYNC:DATA:WHOLe?

Response  
<data> = <STRING RESPONSE DATA>
"H*...*" Specifies the pattern data consisting of 1 to 16 hexadecimal characters.

Function  
Queries the mask pattern for the synchronization control with Frame detection ON.

Example  
> :SENSe:PATTern:SYNC:DATA:WHOLe?
< "H001"

Compatibility  
Compatible with MU181040A/B.
**:SENSe:PATTern:SYNC:DREVerse:ADDRess <start>,<end>**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;start&gt;</td>
<td>&lt;NON-DECIMAL PROGRAM DATA&gt; #H0 to #H3F 0 to 3F bits/1 bit Step (Specify in hexadecimal.)</td>
</tr>
<tr>
<td>&lt;end&gt;</td>
<td>&lt;NON-DECIMAL PROGRAM DATA&gt; #H0 to #H3F 0 to 3F bits/1 bit Step (Specify in hexadecimal.)</td>
</tr>
</tbody>
</table>

**Function**

Inverts the mask pattern data for the synchronization control with Frame detection ON. The range to invert is specified by <start> and <end> addresses.

**Note:**

At 2ch Combination, the setting range is doubled.

At 4ch Combination, the setting range is quadrupled.

**Example**

To invert 1 to 17 bits of the mask pattern data for the synchronization control with Frame detection ON:

> :SENSe:PATTern:SYNC:DREVerse:ADDRess #H001,#H011

**Compatibility**

Compatible with MU181040A/B.

---

**:SENSe:PATTern:SYNC:PROGram:FILL <start>,<end>,<data>**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;start&gt;</td>
<td>&lt;NON-DECIMAL PROGRAM DATA&gt; #H0 to #H3F 0 to 3F bits/1 bit Step (Specify in hexadecimal.)</td>
</tr>
<tr>
<td>&lt;end&gt;</td>
<td>&lt;NON-DECIMAL PROGRAM DATA&gt; #H0 to #H3F 0 to 3F bits/1 bit Step (Specify in hexadecimal.)</td>
</tr>
<tr>
<td>&lt;data&gt;</td>
<td>&lt;DECIMAL NUMERIC PROGRAM DATA&gt; 0 Fills the specified range with 0s. 1 Fills the specified range with 1s.</td>
</tr>
</tbody>
</table>

**Function**

Sets to fill the specified range of the mask pattern data for the synchronization control with Frame detection ON with 0s or 1s. The setting range is specified by <start> and <end> addresses.

**Note:**

At 2ch Combination, the setting range is doubled.

At 4ch Combination, the setting range is quadrupled.

**Example**

To fill 1 to 17 bits of the mask pattern for the synchronization control with Frame detection ON with 0s:

> :SENSe:PATTern:SYNC:PRoGram:FILL #H001,#H011,0

**Compatibility**

Compatible with MU181040A/B.
Chapter 7  SCPI Commands

:SENSe:PATTern:SYNC:ACORder <boolean>

Parameter  
<boolean>=<BOOLEAN PROGRAM DATA>
0 or OFF    Auto Channel Order OFF
1 or ON     Auto Channel Order ON

Function  
Sets whether to change the channel order of synchronization at 2ch Combination. Set Off when PAM4 Decoder is used.

Example  
To set so that the channel order of synchronization is changed.
> :SENSe:PATTern:SYNC:ACORder ON

Compatibility  
Incompatible with existing models

:SENSe:PATTern:SYNC:ACORder?

Response  
<num=NR1 NUMERIC RESPONSE DATA>
0           Auto Channel Order OFF
1           Auto Channel Order ON

Function  
Queries whether to change the channel order of synchronization at 2ch Combination.

Example  
> :SENSe:PATTern:SYNC:ACORder?
< 1

Compatibility  
Incompatible with existing models
### 7.12.3.4 Error/Alarm Condition setting commands

![Error/Alarm Condition Setting](image)

#### Table 7.12.3.4-1 Error/Alarm Condition Setting Commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>:CALCulate:EALarm:ERRor:EDETect?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:CALCulate:EALarm:ERRor:INTerval?</td>
</tr>
</tbody>
</table>

**:CALCulate:EALarm:ERRor:EDETect <mode>**

**Parameter**

<mode> = <CHARACTER PROGRAM DATA>

- INSomi: Detects INS/OMI errors.
- TRANSition: Detects Transition/Non Transition errors.

**Function**

Sets the method of detecting errors during the bit-error/alarm measurement.

**Example**

To set the method of detecting errors during the bit-error/alarm measurement to the INS/OMI error detection:

```
> :CALCulate:EALarm:ERRor:EDETect INSomi
```

**Compatibility**

Compatible with MU181040A/B.

**:CALCulate:EALarm:ERRor:EDETect?**

**Response**

<mode> = <CHARACTER RESPONSE DATA>

- INS, TRAN

**Function**

Queries the method of detecting errors during the bit-error/alarm measurement.

**Example**

```
> :CALCulate:EALarm:ERRor:EDETect?
< INS
```

**Compatibility**

Compatible with MU181040A/B.
Chapter 7  SCPI Commands

:CALCulate:EALarm:ERRor:INTerval <numeric>,<suffix>

Parameter

<numeric> = <DECIMAL NUMERIC PROGRAM DATA>

1 Sets the measurement time value to 1.
10 Sets the measurement time value to 10.
100 Sets the measurement time value to 100.

<suffix> = <CHARACTER PROGRAM DATA>

ms Sets the measurement time unit to ms.
s Sets the measurement time unit to second.

Function

Sets the interval for EI and %EFI measurement.
Only 1 ms, 10 ms, 100 ms, and 1 s are available.

Example

To set the interval for EI and EFI measurement to 1 ms:

> :CALCulate:EALarm:ERRor:INTerval 1,ms

Compatibility

Compatible with MU181040A/B.

:CALCulate:EALarm:ERRor:INTerval?

Response

<numeric> = <NR1 NUMERIC RESPONSE DATA>

1, 10, 100

<suffix> = <CHARACTER RESPONSE DATA>

ms, s

Function

Queries the interval for EI and %EFI measurement.

Example

> :CALCulate:EALarm:ERRor:INTerval?

< 1ms

Compatibility

Compatible with MU181040A/B.
7.12.4 Commands Related to Pattern Tab

7.12.4.1 Test Pattern setting commands

![Pattern Tab](image)

![Test Pattern Area](image)

**Table 7.12.4.1-1 Test Pattern Setting Commands**

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>:SENSe:PATTern:TYPE?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:PATTern:LOGic?</td>
</tr>
</tbody>
</table>
Chapter 7  SCPI Commands

:SENSe:PATTern:TYPE <type>
Parameter  <type> = <CHARACTER PROGRAM DATA>
PRBS         PRBS pattern
ZSUBstitution Zero-Substitution pattern
DATA         Data pattern
MIXData      Mixed Data pattern
Function     Sets the type of the test pattern.
Example      To set the test pattern type to the PRBS pattern:
             > :SENSe:PATTern:TYPE PRBS
Compatibility Partially compatible with MU181040A/B. (Only ZSUBstitution)

:SENSe:PATTern:TYPE?
Response  <type> = <CHARACTER RESPONSE DATA>
PRBS, ZSUB, DATA, MIXD
Function  Queries the type of the test pattern.
Example   > :SENSe:PATTern:TYPE?
           < PRBS
Compatibility Compatible with MU181040A/B.

:SENSe:PATTern:LOGic <logic>
Parameter  <logic> = <CHARACTER PROGRAM DATA>
POSitive   Positive logic
NEGative   Negative logic
Function   Sets the logic (positive or negative) of the test pattern.
Example    To set the logic of the test pattern to the positive:
           > :SENSe:PATTern:LOGic POSitive
Compatibility Compatible with MU181040A/B.

:SENSe:PATTern:LOGic?
Response  <logic> = <CHARACTER RESPONSE DATA>
POS, NEG
Function  Queries the logic of the test pattern.
Example   > :SENSe:PATTern:LOGic?
           < POS
Compatibility Compatible with MU181040A/B.
7.12.4.2 PRBS pattern setting commands

Table 7.12.4.2-1 PRBS Pattern Setting Commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>:SENSe:PATTern:PRBS:LENGth?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:PATTern:PRBS:MRATio?</td>
</tr>
</tbody>
</table>

Example

To set the number of stages for PRBS pattern reception to \(2^7 - 1\):

\[ > \text{:SENSe:PATTern:PRBS:LENGth 7} \]

Compatibility

Compatible with MU181040A/B.
Chapter 7  SCPI Commands

:SENSe:PATTern:PRBS:LENGth?

Response   <numeric> = <NR1 NUMERIC RESPONSE DATA>
            7, 9, 10, 11, 15, 20, 23, 31
Function   Queries the number of stages for PRBS pattern reception.
Example    > :SENSe:PATTern:PRBS:LENGth?
            < 7
Compatibility Compatible with MU181040A/B.

:SENSe:PATTern:PRBS:MRATio <mratio>

Parameter  <mratio> = <CHARACTER PROGRAM DATA>
            M1_2         1/2
            I1_2         1/2INVt
Function   Sets the mark ratio for PRBS pattern reception.
Example    To set the mark ratio for PRBS pattern reception to 1/2:
            > :SENSe:PATTern:PRBS:MRATio M1_2
Compatibility Compatible with MU181040A/B.

:SENSe:PATTern:PRBS:MRATio?

Response   <mratio> = <CHARACTER RESPONSE DATA>
            M1_2, I1_2
Function   Queries the mark ratio for PRBS pattern reception.
Example    > :SENSe:PATTern:PRBS:MRATio?
            < M1_2
Compatibility Compatible with MU181040A/B.

:SENSe:PATTern:PRBS:BSHift?

Response   <numeric> = <NR1 NUMERIC RESPONSE DATA>
            1
Function   Queries the bit shift number for PRBS pattern reception.
Example    > :SENSe:PATTern:PRBS:BSHift?
            < 1
Compatibility Partially compatible with MU181040A/B.
7.12.4.3 Zero Substitution pattern setting commands

![Test Pattern Setting (When Zero Substitution Is Selected)](image)

Table 7.12.4.3-1 Zero Substitution Pattern Setting Commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>:SENSe:PATTern:ZSUBstitute:LENGth?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:PATTern:ZSUBstitute:ZLENgth?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:PATTern:ZSUBstitute:ADDBit?</td>
</tr>
</tbody>
</table>

:SENSe:PATTern:ZSUBstitute:LENGth <length>

Parameter

<table>
<thead>
<tr>
<th>&lt;length&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
</tr>
<tr>
<td>9</td>
</tr>
<tr>
<td>10</td>
</tr>
<tr>
<td>11</td>
</tr>
<tr>
<td>15</td>
</tr>
<tr>
<td>20</td>
</tr>
<tr>
<td>23</td>
</tr>
<tr>
<td>−7</td>
</tr>
<tr>
<td>−9</td>
</tr>
<tr>
<td>−10</td>
</tr>
<tr>
<td>−11</td>
</tr>
<tr>
<td>−15</td>
</tr>
<tr>
<td>−20</td>
</tr>
<tr>
<td>−23</td>
</tr>
</tbody>
</table>

Function

Sets the pattern length for Zero-Substitution pattern reception.

Example

To set the pattern length for Zero-Substitution pattern reception to $2^7$:

> :SENSe:PATTern:ZSUBstitute:LENGth 7

Compatibility

Compatible with MU181040A/B.
Chapter 7  SCPI Commands

:SENSe:PATTern:ZSUBstitute:LENGth?
Response  <length> = <NR1 NUMERIC RESPONSE DATA>
7, 9, 10, 11, 15, 20, 23, −7, −9, −10, −11, −15, −20, −23
Function  Queries the pattern length for Zero-Substitution pattern reception.
Example  > :SENSe:PATTern:ZSUBstitute:LENGth?
< 7
Compatibility  Compatible with MU181040A/B.

:SENSe:PATTern:ZSUBstitute:ZLENgth <numeric>
Parameter  <numeric> = <DECIMAL NUMERIC PROGRAM DATA>
When ZSUBlength, 2^n is set:
1 to 2^n-1  1 to 2^n-1/1 bit Step
When ZSUBlength, 2^{n-1} is set:
1 to 2^{n-2}  1 to 2^{n-2}/1 bit Step
n = 7, 9, 11, 15, 20, 23
Function  Sets the number of bits to which “0” is inserted (zero-insertion bit count) for Zero-Substitution pattern reception.
Example  To set 10 bits to the number of bits to which “0” is inserted for Zero-Substitution pattern reception:
> :SENSe:PATTern:ZSUBstitute:ZLENgth 10
Compatibility  Compatible with MU181040A/B.

:SENSe:PATTern:ZSUBstitute:ZLENgth?
Response  <numeric> = <NR1 NUMERIC RESPONSE DATA>
1 to 2^n-1
n = 7, 9, 11, 15, 20, 23
Function  Queries the number of bits to which “0” is inserted (zero-insertion bit count) for Zero-Substitution pattern reception.
Example  >:SENSe:PATTern:ZSUBstitute:ZLENgth?
< 10
Compatibility  Compatible with MU181040A/B.
:SENSe:PATTern:ZSUBstitute:ADDBit <numeric>

Parameter
<numeric> = <DECIMAL NUMERIC PROGRAM DATA>
0  The 2\textsuperscript{n}th bit is set to “0”.
1  The 2\textsuperscript{n}th bit is set to “1”. (Similar to the existing models)

Function
Sets to add 0 or 1 after the last bit of the consecutively-inserted 0s for Zero-Substitution pattern reception.

Example
To set to add 0 after the last bit of the consecutively-inserted 0s for Zero-Substitution pattern reception:
> :SENSe:PATTern:ZSUBstitute:ADDBit 0

Compatibility
Compatible with MU181040A/B.

:SENSe:PATTern:ZSUBstitute:ADDBit?

Response
<numeric> = <NR1 NUMERIC RESPONSE DATA>
0  The 2\textsuperscript{n}th bit is set to “0”.
1  The 2\textsuperscript{n}th bit is set to “1”. (Similar to the existing models)

Function
Queries whether one bit of “0” or “1” is added next to the last of zero-inserted.

Example
> :SENSe:PATTern:ZSUBstitute:ADDBit?
< 0

Compatibility
Compatible with MU181040A/B.
7.12.4.4 Data pattern setting commands

![Figure 7.12.4.4-1 Test Pattern Setting (When Data Is Selected)](image1)

![Figure 7.12.4.4-2 Data Pattern Setting (Pattern Editor Dialog Box)](image2)

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>:SENSe:PATTern:DATA:LENGth?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:PATTern:DATA:WHOLe?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:PATTern:BDATa:WHOLe?</td>
</tr>
<tr>
<td>[5]</td>
<td>No label (Data inversion)</td>
<td>:SENSe:PATTern:DREVerse:DELTa</td>
</tr>
<tr>
<td>[6]</td>
<td>No label (All 0 or 1)</td>
<td>:SENSe:PATTern:DATA:FILL</td>
</tr>
</tbody>
</table>
### :SENSe:PATTern:DATA:LENGth \(<\text{numeric}>\)

**Parameter**

\(<\text{numeric}> = \langle\text{DECIMAL NUMERIC PROGRAM DATA}\rangle \)

2 to 268435456  \(\text{2 to 268,435,456 bits/1 bit Step}\)

**Note:**

At 2ch Combination, the setting range and Step are doubled.
At 4ch Combination, the setting range and Step are quadrupled.

**Function**

Sets the pattern length during Data pattern reception.

**Example**

To set the pattern length during Data pattern reception to 2 bits:

```plaintext
> :SENSe:PATTern:DATA:LENGth 2
```

**Compatibility**

Partially compatible with MU181040A/B.

### :SENSe:PATTern:DATA:LENGth?  

**Response**

\(<\text{numeric}> = \langle\text{NR1 NUMERIC RESPONSE DATA}\rangle \)

**Function**

Queries the pattern length during Data pattern reception.

**Example**

```plaintext
> :SENSe:PATTern:DATA:LENGth?
< 2
```

**Compatibility**

Partially compatible with MU181040A/B.

### :SENSe:PATTern:DATA:WHOLe \(<\text{start}>, <\text{end}>, <\text{data}>\)

**Parameter**

\(<\text{start}> = \langle\text{NON-DECIMAL PROGRAM DATA}\rangle \)

\(\#H0\) to \(\#H7FFFFFF\)  \(0\) to \(7FFFFFF\) bits/1 bit Step  
(Specify in hexadecimal.)

\(<\text{end}> = \langle\text{NON-DECIMAL PROGRAM DATA}\rangle \)

\(\#H0\) to \(\#H7FFFFFF\)  \(0\) to \(7FFFFFF\) bits/1 bit Step  
(Specify in hexadecimal.)

**Note:**

The maximum setting is the pattern length.
At 2ch Combination, the setting range is doubled.
At 4ch Combination, the setting range is quadrupled.

\(<\text{data}> = \langle\text{STRING PROGRAM DATA}\rangle \)

"H***...*"

Specify pattern data in hexadecimal format.
1 to 400 characters (pattern data of \(400 \times 4\) bits)
Specify a character string consisting of 0 to 9 and A to F.
If the last character string of the pattern data is short of a hexadecimal unit (4 bits), add 0(s) after the last bit until the last character string becomes hexadecimal.
"B***...*" Specify pattern data in binary format.
1 to 400 characters (pattern data of 400 bits)
Specify a character string consisting of 0 and 1.

Function
Sets the pattern data from the <start> to <end> addresses during Data pattern reception. The set data overwrites the specified range.

Example
To set the addresses from bit 0 to bit 7FFFFFF of the pattern data in hexadecimal format:
> :SENSe:PAAtt:DATa:WHOLe #H0,#H7FFFFFF,"H001"

Compatibility
Partially compatible with MU181040A/B.

Note:
When the number of data bits specified in <data> is smaller than the range specified by <start> and <end>, set <data> repeatedly up to the amount of the specified range.

<Example>
•<start> = #H0, <end> = #H1F, <data> = "HABC"
  Setting data: ABCABCAB
•<start> = #H0, <end> = #H7, <data> = "B011"
  Setting data: 01101101

When the number of data bits specified in <data> is greater than the range specified by <start> and <end>, cut off the part of <data> out of the specified range.

<Example>
•<start> = #H0, <end> = #HF, <data> = #HABCDEF
  Setting data: ABCD
•<start> = #H0, <end> = #H3, <data> = #B01100110
  Setting data: 0110
:SENSe:PATTern:DATA:WHOLe? <start>

Parameter

<start> = <NON-DECIMAL PROGRAM DATA>
#H0 to #HFFFFFFFF 0 to FFFFFFFF bits/1 bit Step
(Specify in hexadecimal.)

Note:

- The maximum setting is the pattern length.
- At 2ch Combination, the setting range is doubled.
- At 4ch Combination, the setting range is quadrupled.

Response

<data> = <STRING RESPONSE DATA>
"H***...*" Returns the pattern data in hexadecimal format.
Up to 400 characters (pattern data of 400 × 4 bits)

Function

Queries the pattern data of 400 × 4 bits from the <start> address during Data pattern reception.

Example

To query the pattern data of 400 × 4 bits from 0-bit address during Data pattern reception:

> :SENSe:PATTern:DATA:WHOLe? #H0
< "H001"

Compatibility

Partially compatible with MU181040A/B.

:SENSe:PATTern:BDATa:WHOLe <start>,<end>,<bdata>

Parameter

<start> = <NON-DECIMAL PROGRAM DATA>
#H0 to #HFFFFFFFF 0 to FFFFFFFF bits/1 bit Step
(Specify in hexadecimal.)

<end> = <NON-DECIMAL PROGRAM DATA>
#H0 to #HFFFFFFFF 0 to FFFFFFFF bits/1 bit Step
(Specify in hexadecimal.)

Note:

- The maximum setting is the pattern length.
- At 2ch Combination, the setting range is doubled.
- At 4ch Combination, the setting range is quadrupled.

<bdata> = <ARBITRARY BLOCK PROGRAM DATA>
#XYYY<binary> X: Number of YYY digits
YYY: Number of bytes of <binary>,
1 to 16,777,216 (bytes)
<binary>: Binary data up to 16,777,216 bytes

Function

Sets binary data of the pattern data from <start> to <end> addresses during Data pattern reception. The set data overwrites the specified range.

Example

To set the binary data of the pattern data from 0 to FFFFFFFF bits addresses during Data pattern reception:
Chapter 7  SCPI Commands

>:SENSe:PATTern:BDATa:WHOLe #H0,#FFFFFFF,#10011

Compatibility
Partially compatible with MU181040A/B.

### :SENSe:PATTern:BDATa:WHOLe? <start> [,<size>] 

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| **<start>** | = <NON-DECIMAL PROGRAM DATA>  
  #H0 to #FFFFFFF  
  0 to FFFFFFFF bits/1 bit Step  
  (Specify in hexadecimal.) |
| **<size>** | = <NR1 NUMERIC PROGRAM DATA>  
  1 to 16777216  
  1 to 16,777,216 bytes/1 byte Step |

Binary data of the setting pattern is queried when [<size>] is omitted.

**Note:**
- The maximum setting is the pattern length.
- At 2ch Combination, the setting range is doubled.
- At 4ch Combination, the setting range is quadrupled.

**Response**
<bdata> = <DEFINITE LENGTH ARBITRARY BLOCK RESPONSE DATA>

#XYYY<binary>  
X: Number of YYY digits  
YYY: Number of bytes of <binary>,  
1 to 16,777,216 (bytes)  
<binary>: Binary data up to 16,777,216 bytes

**Function**
Queries binary data of bytes specified by <size> from <start> address of the pattern data during Data pattern reception.

**Example**
To query the pattern data of 16,777,216 bytes from 0-bit address during Data pattern reception:
>:SENSe:PATTern:BDATa:WHOLe? #H0  
< #10011

**Compatibility**
Partially compatible with MU181040A/B.
:SENSe:PATTern:DREVerse:ADDRes  <start>,<end>

Parameter

<start> = <NON-DECIMAL PROGRAM DATA>
#H0 to #HFFFFFFF  0 to FFFFFFFF bits/1 bit Step
(Specify in hexadecimal.)

<end> = <NON-DECIMAL PROGRAM DATA>
#H0 to #HFFFFFFF  0 to FFFFFFFF bits/1 bit Step
(Specify in hexadecimal.)

Note:
The maximum setting is the pattern length.
At 2ch Combination, the setting range is doubled.
At 4ch Combination, the setting range is quadrupled.

Function

Inverts the program data during Data pattern reception.
The range to invert is specified by <start> and <end> addresses.

Example

To invert 0 to 7FFFFFFF bit address of the program at Data pattern reception:
> :SENSe:PA  #H0,#H7FFFFFFF

Compatibility

Partially compatible with MU181040A/B.

:SENSe:PATTern:DREVerse:DELTa <start>,<delta>

Parameter

<start> = <NON-DECIMAL PROGRAM DATA>
#H0 to #HFFFFFFF  0 to FFFFFFFF bits/1 bit Step
(Specify in hexadecimal.)

<delta> = <DECIMAL NUMERIC PROGRAM DATA>
1 to 268435456  1 to 268,435,456 bits/1 bit Step

Note:
The maximum setting is the pattern length.
At 2ch Combination, the setting range is doubled.
At 4ch Combination, the setting range is quadrupled.

Function

Inverts data of program part at Data pattern reception. Specify the range to invert with the number of relative bits from the <start> address.

Example

To invert 1 bit that is a relative bit of 0 bit address in the program at the time of Data pattern reception:
> :SENSe:PA  #H0,1

Compatibility

Partially compatible with MU181040A/B.
Chapter 7  SCPI Commands

:SENSe:PATTern:DATA:FILL <range>,<page>,<data>

Parameter

- `<range> = <CHARACTER PROGRAM DATA>`
  - PAGE: Specifies a page. (One page is defined as 128 bits.)
  - ALL: Specifies all data.

- `<page> = <DECIMAL NUMERIC PROGRAM DATA>`
  - 1 to 2097152: 1 to 2,097,152 (Max.) pages, in 1-page steps
  - Max = \( \frac{\text{Pattern Length}}{128} \) (rounding up fractions)
  - Specify “0” when `<range>` is set to ALL.

- `<data> = <DECIMAL NUMERIC PROGRAM DATA>`
  - 0: Fills the specified range with 0s.
  - 1: Fills the specified range with 1s.

Function

Sets 0s or 1s for the specified page or all data in the program during Data pattern reception.

Example

To set 0s for the page 1 in the program during Data pattern reception:

>| :SENSe:PATTern:DATA:FILL PAGE,1,0 |

Compatibility

Partially compatible with MU181040A/B.
7.12.4.5 Mixed Data pattern setting commands

Figure 7.12.4.5-1 Test Pattern Setting (When Mixed Is Selected)

Figure 7.12.4.5-2 Descramble Setup Dialog Box
### Table 7.12.4.5-1 Mixed Data pattern Setting Commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
</table>
| 1   | Bit Shift              | :SENSe:PATTern:MIXData:PRBS:BITShift  
|     |                        | :SENSe:PATTern:MIXData:PRBS:BITShift?                                  |
| 2   | PRBS Pattern           | :SENSe:PATTern:MIXData:PRBS:LENGth                                      |
|     |                        | :SENSe:PATTern:MIXData:PRBS:LENGth?                                    |
| 3   | PRBS Mark Ratio        | :SENSe:PATTern:MIXData:PRBS:MRATio                                      |
|     |                        | :SENSe:PATTern:MIXData:PRBS:MRATio?                                    |
| 4   | Descramble             | :SENSe:PATTern:MIXData:DEScramble                                       |
|     |                        | :SENSe:PATTern:MIXData:DEScramble?                                     |
| 5   | PRBS Sequence          | :SENSe:PATTern:MIXData:PRBS:SEQuence                                    |
|     |                        | :SENSe:PATTern:MIXData:PRBS:SEQuence?                                  |
| 6   | Setup                  | :SENSe:PATTern:MIXData:SRSetting                                       |
|     |                        | :SENSe:PATTern:MIXData:SRSetting?                                      |
| 7   | Set All/Reset All      | :SENSe:PATTern:MIXData:DESCramble:ALLSet                               |
| 8   | Number of Block        | :SENSe:PATTern:MIXData:BLOCk                                            |
|     |                        | :SENSe:PATTern:MIXData:BLOCk?                                          |
| 9   | Row Length             | :SENSe:PATTern:MIXData:RLENgth                                         |
|     |                        | :SENSe:PATTern:MIXData:RLENgth?                                        |
| 10  | Data Length            | :SENSe:PATTern:MIXData:DATA:LENGth                                     |
|     |                        | :SENSe:PATTern:MIXData:DATA:LENGth?                                    |
| 11  | Number of Row          | :SENSe:PATTern:MIXData:ROW                                              |
|     |                        | :SENSe:PATTern:MIXData:ROW?                                            |
| 12  | No label (Pattern setting) | :SENSe:PATTern:MIXData:DATA:WHOLe                                 |
|     |                        | :SENSe:PATTern:MIXData:DATA:WHOLe?                                     |

![Pattern Editor Dialog Box](image)
Table 7.12.4-5-1  Mixed Data pattern Setting Commands (Cont’d)

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>:SENSe:PATTern:MIXData:BDATa:WHOLEe?</td>
</tr>
<tr>
<td>[16]</td>
<td>No label (Data inversion)</td>
<td>:SENSe:PATTern:MIXData:DATA:FILL</td>
</tr>
</tbody>
</table>

:SENSe:PATTern:MIXData:PRBS:BITShift?

**Response**

<numeric> = <NR1 NUMERIC RESPONSE DATA>

1 1 bit

**Function**

Queries the bit shift for the PRBS pattern during Mixed Data pattern reception

**Example**

> :SENSe:PATTern:MIXData:PRBS:BITShift?
< 1

**Compatibility**

Partially compatible with MU181040A/B.

:SENSe:PATTern:MIXData:PRBS:LENGth <numeric>

**Parameter**

<numeric> = <DECIMAL NUMERIC PROGRAM DATA>

7 2<sup>n</sup>–1 (n = 7)
9 2<sup>n</sup>–1 (n = 9)
10 2<sup>n</sup>–1 (n = 10)
11 2<sup>n</sup>–1 (n = 11)
15 2<sup>n</sup>–1 (n = 15)
20 2<sup>n</sup>–1 (n = 20)
23 2<sup>n</sup>–1 (n = 23)
31 2<sup>n</sup>–1 (n = 31)

**Function**

Sets the number of stages of the PRBS pattern during Mixed Data pattern reception.

**Example**

To set the number of stages of the PRBS pattern during Mixed Data pattern reception to 2<sup>7</sup>–1:

> :SENSe:PATTern:MIXData:PRBS:LENGth 7

**Compatibility**

Compatible with MU181040A/B.
**Chapter 7  SCPI Commands**

:SENSe:PATTern:MIXData:PRBS:LENGth?

**Response**  
<numeric> = <NR1 NUMERIC RESPONSE DATA>
7, 9, 10, 11, 15, 20, 23, 31

**Function**  
Queries the number of stages of the PRBS pattern during Mixed Data pattern reception.

**Example**  
> :SENSe:PATTern:MIXData:PRBS:LENGth?
< 7

**Compatibility**  
Compatible with MU181040A/B.

:SENSe:PATTern:MIXData:PRBS:MRATio <mratio>

**Parameter**  
<mratio> = <CHARACTER PROGRAM DATA>
M1_2 1/2
I1_2 1/2INV

**Function**  
Sets the mark ratio for the PRBS pattern during Mixed Data pattern reception.

**Example**  
To set the mark ratio for the PRBS pattern during Mixed Data pattern reception to 1/2:
> :SENSe:PATTern:MIXData:PRBS:MRATio M1_2

**Compatibility**  
Partially compatible with MU181040A/B.

:SENSe:PATTern:MIXData:PRBS:MRATio?

**Response**  
<mratio> = <CHARACTER RESPONSE DATA>
M1_2, I1_2

**Function**  
Queries the mark ratio for the PRBS pattern during Mixed Data pattern reception.

**Example**  
> :SENSe:PATTern:MIXData:PRBS:MRATio?
< M1_2

**Compatibility**  
Compatible with MU181040A/B.

:SENSe:PATTern:MIXData:DEScramble <boolean>

**Parameter**  
<boolean> = <BOOLEAN PROGRAM DATA>
0 or OFF Descramble OFF
1 or ON Descramble ON

**Function**  
Sets descramble ON/OFF of the PRBS7 stage during pattern reception.

**Example**  
To set descramble ON of the PRBS7 stage during pattern reception:
> :SENSe:PATTern:MIXData:DEScramble ON

**Compatibility**  
Compatible with MU181040A/B.
:SENSe:PATTern:MIXData:DESCramble?

Response

<table>
<thead>
<tr>
<th>&lt;numeric&gt;</th>
<th>=</th>
<th>&lt;NR1 NUMERIC RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td>Descramble OFF</td>
</tr>
<tr>
<td>1</td>
<td></td>
<td>Descramble ON</td>
</tr>
</tbody>
</table>

Function
Queries the descramble ON/OFF state of the PRBS7 stage during pattern reception.

Example

> :SENSe:PATTern:MIXData:DESCramble?
< 1

Compatibility
Compatible with MU181040A/B.

:SENSe:PATTern:MIXData:PRBS:SEQuence <sequence>

Parameter

<table>
<thead>
<tr>
<th>&lt;sequence&gt;</th>
<th>=</th>
<th>&lt;CHARACTER PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESTart</td>
<td></td>
<td>Inconsecutive PRBS pattern</td>
</tr>
<tr>
<td>CONSecutive</td>
<td></td>
<td>Consecutive PRBS pattern</td>
</tr>
</tbody>
</table>

Function
Sets the PRBS pattern sequence during Mixed Data pattern reception.

Example

To set the PRBS pattern sequence to Restart during Mixed Data pattern reception:

> :SENSe:PATTern:MIXData:PRBS:SEQuence RESTart

Compatibility
Compatible with MU181040A/B.

:SENSe:PATTern:MIXData:PRBS:SEQuence?

Response

<table>
<thead>
<tr>
<th>&lt;sequence&gt;</th>
<th>=</th>
<th>&lt;CHARACTER RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>REST, CONS</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Function
Queries the PRBS pattern sequence during Mixed Data pattern reception.

Example

> :SENSe:PATTern:MIXData:PRBS:SEQuence?
< REST

Compatibility
Compatible with MU181040A/B.
**Chapter 7  SCPI Commands**

**:SENSe:PATTern:MIXData:SRSetting <row>,<data>,<prbs>**

Parameter
- `<row>` = <DECIMAL NUMERIC PROGRAM DATA>
  - 1 to 16 1 to 16 Row/1 Row Step
- `<data>` = <BOOLEAN PROGRAM DATA>
  - 0 or OFF Descramble OFF
  - 1 or ON Descramble ON
- `<prbs>` = <BOOLEAN PROGRAM DATA>
  - 0 or OFF Descramble OFF
  - 1 or ON Descramble ON

Note, however, that Data of 1 Row is always Descramble OFF.

Function
Sets Descramble ON or OFF for the Data and PRBS of the specified Row.

Example
To set Descramble setting of 1 Row to Data ON and PRBS ON:
> :SENSe:PATTern:MIXData:SRSetting 1,ON,ON

Compatibility
Compatible with MU181040A/B.

**:SENSe:PATTern:MIXData:SRSetting? <row>**

Parameter
- `<row>` = <DECIMAL NUMERIC PROGRAM DATA>
  - 1 to 16 1 to 16 Row/1 Row Step

Response
- `<data>,<prbs>` = <NR1 NUMERIC RESPONSE DATA>
  - 0 Descramble OFF
  - 1 Descramble ON

Function
Queries the descramble setting (ON/OFF) for the Data and PRBS of the specified Row.

Example
> :SENSe:PATTern:MIXData:SRSetting? 1
< 1,1

Compatibility
Compatible with MU181040A/B.

**:SENSe:PATTern:MIXData:DESCramble:ALLSet <numeric>**

Parameter
- `<numeric>` = <NUMERIC PROGRAM DATA>
  - 0 Resets all.
  - 1 Selects all.

Function
Selects all or resets all of Descramble setting.

Example
To select all of Descramble setting:
> :SENSe:PATTern:MIXData:DESCramble:ALLSet 1

Compatibility
Compatible with MU181040A/B.
**:SENSe:PATTern:MIXData:BLOCk <numeric>**

**Parameter**

<numeric> = <DECIMAL NUMERIC PROGRAM DATA>

1 to 511

Function

Sets the number of blocks during Mixed Data pattern reception.

Example

To set the number of blocks during Mixed Data pattern reception to 1:

> :SENSe:PATTern:MIXData:BLOCk 1

Compatibility

Compatible with MU181040A/B.

**:SENSe:PATTern:MIXData:BLOCk?**

**Response**

<numeric> = <NR1 NUMERIC RESPONSE DATA>

1 to 511

Function

Queries the number of blocks during Mixed Data pattern reception.

Example

> :SENSe:PATTern:MIXData:BLOCk?

< 1

Compatibility

Compatible with MU181040A/B.

**:SENSe:PATTern:MIXData:RLENgth <numeric>**

**Parameter**

<numeric> = <DECIMAL NUMERIC PROGRAM DATA>

1536 to 2415919104

1,536 to 2,415,919,104 bits/256 bits Step

Note:

At 2ch Combination, the setting range varies and Step is doubled.

3072 to 4831838208

3,072 to 4,831,838,208 bits/512 bits Step

At 4ch Combination, the setting range varies and Step is quadrupled.

6144 to 9663676416

6,144 to 9,663,676,416 bits/1,024 bits Step

Function

Sets the pattern length of 1 Row that is edited during Mixed Data pattern reception.

Example

To set Pattern length of 1 Row edited at Mixed Data pattern reception to 1,536 bits:

> :SENSe:PATTern:MIXData:RLENgth 1536

Compatibility

Incompatible with existing models
Chapter 7  SCPI Commands

:SENSe:PATTern:MIXData:RLENgth?

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;numeric&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function</td>
<td>Queries the pattern length of 1 Row that is edited during Mixed Data pattern reception.</td>
</tr>
<tr>
<td>Example</td>
<td>:SENSe:PATTern:MIXData:RLENgth?</td>
</tr>
<tr>
<td></td>
<td>&lt; 1536</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Partially compatible with MU181040A/B.</td>
</tr>
</tbody>
</table>

:SENSe:PATTern:MIXData:DATA:LENGth <numeric>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;numeric&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>1024 to 268435456</td>
<td>1,024 to 268,435,456 bits/1 bit Step</td>
</tr>
<tr>
<td>Note:</td>
<td></td>
</tr>
<tr>
<td>At 2ch Combination, the setting range and Step are doubled.</td>
<td></td>
</tr>
<tr>
<td>2048 to 536870912</td>
<td>2,048 to 536,870,912 bits/2 bit Step</td>
</tr>
<tr>
<td>At 4ch Combination, the setting range and Step are quadrupled.</td>
<td></td>
</tr>
<tr>
<td>4096 to 1073741824</td>
<td>4,096 to 1,073,741,824 bits/4 bit Step</td>
</tr>
</tbody>
</table>

| Function | Sets the pattern length on the pattern data side that is edited during Mixed Data pattern reception. |
| Example  | To set the pattern length on the pattern data side that is edited during Mixed Data pattern reception: |
|          | > :SENSe:PATTern:MIXData:DATA:LENGth 1024 |
| Compatibility | Partially compatible with MU181040A/B. |

:SENSe:PATTern:MIXData:DATA:LENGth?

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;numeric&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function</td>
<td>Queries the pattern length on the pattern data side that is edited during Mixed Data pattern reception.</td>
</tr>
<tr>
<td>Example</td>
<td>&gt; :SENSe:PATTern:MIXData:DATA:LENGth?</td>
</tr>
<tr>
<td></td>
<td>&lt; 2048</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Partially compatible with MU181040A/B.</td>
</tr>
</tbody>
</table>
**:SENSe:PATTern:MIXData:ROW <numeric>**

**Parameter**

<numeric> = <DECIMAL NUMERIC PROGRAM DATA>

1 to 16

1 to 16 Row/1 Row Step

**Function**

Sets the number of rows within one block that is edited during Mixed Data pattern reception.

**Example**

To set the number of rows within one block that is edited during Mixed Data pattern reception to 1:

> :SENSe:PATTern:MIXData:ROW 1

**Compatibility**

Compatible with MU181040A/B.

**:SENSe:PATTern:MIXData:ROW?**

**Response**

<numeric> = <NR1 NUMERIC RESPONSE DATA>

1 to 16

1 to 16 Row

**Function**

Queries the number of rows within one block that is edited during Mixed Data pattern reception.

**Example**

> :SENSe:PATTern:MIXData:ROW?

< 1

**Compatibility**

Compatible with MU181040A/B.
:SENSe:PATTern:MIXData:DATA:WHOLe <block>,<start>,<end>,<data>

Parameter

- `<block>` = <DECIMAL NUMERIC PROGRAM DATA>
  - 1 to 511
  - 1 to 511 Block/1 Step

- `<start>` = <NON-DECIMAL PROGRAM DATA>
  - #H0 to #HFFFFFFF
  - 0 to FFFFFFF bits/1 bit Step
  - (Specify in hexadecimal.)

- `<end>` = <NON-DECIMAL PROGRAM DATA>
  - #H0 to #HFFFFFFF
  - 0 to FFFFFFF bits/1 bit Step
  - (Specify in hexadecimal.)

Note:
The maximum setting is the pattern length.
At 2ch Combination, the setting range is doubled.
At 4ch Combination, the setting range is quadrupled.

- `<data>` = <STRING PROGRAM DATA>
  - "H***...*"
    - Specify pattern data in hexadecimal format.
    - 1 to 400 characters (pattern data of 400 × 4 bits)
    - Specify a character string consisting of 0 to 9
      and A to F.
    - If the last character string of the pattern data is
      short of a hexadecimal unit(4 bits), add 0(s) after
      the last bit until the last character string
      becomes hexadecimal.
  
  - "B***...*"
    - Specify pattern data in binary format.
    - 1 to 400 characters (pattern data of 400 bits)
    - Specify a character string consisting of 0 and 1.

Function
Sets the pattern data from `<start>` to `<end>` addresses in each block
during Mixed Data pattern reception. The set data overwrites the
specified range.

Example
To set the pattern data of addresses from 0 to 7FFFFFFF bits of 3 blocks
in hexadecimal during Mixed Data pattern reception:
> :SENSe:PATTern:MIXData:DATA:WHOLe 3,#H0,#H7FFFFFFF,
  "H001"

Compatibility
Partially compatible with MU181040A/B.
:SENSe:PA**Ttern:MIXData:DATA:WHOLe? <block>,<start>

Parameter

- `<block>` = <DECIMAL NUMERIC PROGRAM DATA>
  1 to 511  1 to 511 Block/1 Step
- `<start>` = <NON-DECIMAL PROGRAM DATA>
  #H0 to #HFFFFFFF  0 to FFFFFFF bits/1 bit Step
  (Specify in hexadecimal.)

**Note:**

The maximum setting is the pattern length.
At 2ch Combination, the setting range is doubled.
At 4ch Combination, the setting range is quadrupled.

Response

- `<data>` = <STRING RESPONSE DATA>
  "H***...*": Returns the pattern data in hexadecimal format.
  Up to 400 characters (pattern data of 400 × 4 bits)

Function

Queries the pattern data of 400 × 4 bits from the `<start>` address in each block during Data pattern reception.

Example

To query the pattern data of 400 × 4 bits from 0-bit address in 3 blocks during Data pattern reception:

```
> :SENSe:PA**Ttern:MIXData:DATA:WHOLe? 3,#H0
< "#001"
```

Compatibility

Partially compatible with MU181040A/B.
### :SENSe:PATTern:MIXData:BDATa:WHOLe

#### Parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;block&gt;</code></td>
<td><code>&lt;DECIMAL NUMERIC PROGRAM DATA&gt;</code></td>
</tr>
<tr>
<td><code>1 to 511</code></td>
<td><code>1 to 511 Block/1 Step</code></td>
</tr>
<tr>
<td><code>&lt;start&gt;</code></td>
<td><code>&lt;NON-DECIMAL PROGRAM DATA&gt;</code></td>
</tr>
<tr>
<td><code>#H0 to #HFFFFFFF</code></td>
<td><code>0 to FFFFFFFF bits/1 bit Step</code></td>
</tr>
<tr>
<td>(Specify in hexadecimal.)</td>
<td></td>
</tr>
<tr>
<td><code>&lt;end&gt;</code></td>
<td><code>&lt;NON-DECIMAL PROGRAM DATA&gt;</code></td>
</tr>
<tr>
<td><code>#H0 to #HFFFFFFF</code></td>
<td><code>0 to FFFFFFFF bits/1 bit Step</code></td>
</tr>
<tr>
<td>(Specify in hexadecimal.)</td>
<td></td>
</tr>
<tr>
<td><code>&lt;bdata&gt;</code></td>
<td><code>&lt;ARBITRARY BLOCK PROGRAM DATA&gt;</code></td>
</tr>
<tr>
<td><code>#XYYY&lt;binary&gt;</code></td>
<td><code>X: Number of YYY digits</code></td>
</tr>
<tr>
<td><code>YYY: Number of bytes of &lt;binary&gt;</code>,</td>
<td><code>1 to 16,777,216 (bytes)</code></td>
</tr>
<tr>
<td><code>&lt;binary&gt;</code></td>
<td>Binary data up to 16,777,216 bytes</td>
</tr>
</tbody>
</table>

#### Note:

- The maximum setting is the pattern length.
- At 2ch Combination, the setting range is doubled.
- At 4ch Combination, the setting range is quadrupled.

#### Function

Sets binary data of the pattern data from `<start>` to `<end>` addresses in each block during Data pattern reception.

The set data overwrites the specified range.

#### Example

To set binary data from 0 to 7FFFFF address in three block pattern data during Data pattern reception:

```
>:SENSe:PATTern:MIXData:BDATa:WHOLe
3,#H0,#H7FFFFFF,#10011
```

#### Compatibility

Partially compatible with MU181040A/B.
### 7.12 28G/32G bit/s ED Commands

**:SENSe:PATTern:MIXData:BDATa:WHOLe? <block>,<start>[,<size>]**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;block&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
<td>1 to 511 Block/1 Step</td>
</tr>
<tr>
<td>&lt;start&gt; = &lt;NON-DECIMAL PROGRAM DATA&gt;</td>
<td>#H0 to #HFFFFFFF 0 to FFFFFFFF bits/1 bit Step</td>
</tr>
<tr>
<td>[&lt;size&gt;] = &lt;NR1 NUMERIC PROGRAM DATA&gt;</td>
<td>1 to 16777216 1 to 16,777,216 bytes/1 byte Step</td>
</tr>
</tbody>
</table>

**Note:**

The maximum setting is the pattern length.
At 2ch Combination, the setting range is doubled.
At 4ch Combination, the setting range is quadrupled.

**Response**

<bdata> = <DEFINITE LENGTH ARBITRARY BLOCK RESPONSE DATA>

#XYYY<binary>

- X: Number of YYY digits
- YYY: Number of bytes of <binary>,
  - 1 to 16,777,216 (bytes)

<binary>:Binary data up to 16,777,216 bytes

Binary data of the setting pattern is queried when [<size>] is omitted.

**Function**

Queries binary data of bytes specified by <size> from <start> address of the pattern data in each block during Data pattern reception.

**Example**

To query binary data of up to the setting pattern from the <Start> address of the pattern data in block 3 during Mixed Data pattern reception:

```plaintext
> :SENSe:PATTern:MIXData:BDATa:WHOLe? 3,#H0
< #10011
```

**Compatibility**

Partially compatible with MU181040A/B.
## Chapter 7 SCPI Commands

### :SENSe:PATTern:MIXData:DREVerse:ADDRess <block>,<start>,<end>

**Parameter**

- `<block>` = <DECIMAL NUMERIC PROGRAM DATA>
  
  1 to 511 1 to 511 Block/1 Step

- `<start>` = <NON-DECIMAL PROGRAM DATA>
  
  #H0 to #HFFFFFFF 0 to FFFFFFFF bits/1 bit Step

- `<end>` = <NON-DECIMAL PROGRAM DATA>
  
  #H0 to #HFFFFFFF 0 to FFFFFFFF bits/1 bit Step

**Note:**

- The maximum setting is the pattern length.
- At 2ch Combination, the setting range is doubled.
- At 4ch Combination, the setting range is quadrupled.

**Function**

Inverts the data in the program part in a block during Mixed Data pattern reception. Specify the inversion range by the `<start>` and `<end>` addresses.

**Example**

To invert the data in the program part in block 3 from bit 0 to bit 7FFFFFF addresses during Mixed Data pattern reception:

> :SENSe:PATTern:MIXData:DREVerse:ADDRess 3,#H0,#H7FFFFFFF

**Compatibility**

Partially compatible with MU181040A/B.

### :SENSe:PATTern:MIXData:DREVerse:DELTa <block>,<start>,<delta>

**Parameter**

- `<block>` = <DECIMAL NUMERIC PROGRAM DATA>
  
  1 to 511 1 to 511 Block/1 Step

- `<start>` = <NON-DECIMAL PROGRAM DATA>
  
  #H0 to #HFFFFFFF 0 to FFFFFFFF bits/1 bit Step

- `<delta>` = <DECIMAL NUMERIC PROGRAM DATA>
  
  1 to 268435456 1 to 268,435,456 bits/1 bit Step

**Note:**

- The maximum setting is the pattern length.
- At 2ch Combination, the setting range is doubled.
- At 4ch Combination, the setting range is quadrupled.

**Function**

Inverts data of program part in each block at Data pattern reception. Specify the range to invert with the number of relative bits from the `<start>` address.

**Example**

To invert data of program part in three blocks by one bit relative bit count from 0 bit at Data pattern reception:

> :SENSe:PATTern:MIXData:DREVerse:DELTa 3,#H0,1

**Compatibility**

Partially compatible with MU181040A/B.
:SENSe:PATTern:MIXData:DATA:FILL <block>,<range>,<page>,<data>

Parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;block&gt;</td>
<td>&lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
</tr>
<tr>
<td></td>
<td>1 to 511 Block/1 Step</td>
</tr>
<tr>
<td>&lt;range&gt;</td>
<td>&lt;CHARACTER PROGRAM DATA&gt;</td>
</tr>
<tr>
<td></td>
<td>PAGE Specifies pages (One page is defined as 128 bits.)</td>
</tr>
<tr>
<td></td>
<td>ALL Specifies all data.</td>
</tr>
<tr>
<td></td>
<td>&lt;page&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
</tr>
<tr>
<td></td>
<td>1 to 2097152 1 to 2,097,152 (Max) page/1 page Step</td>
</tr>
<tr>
<td>Max</td>
<td>Pattern Length = \frac{128}{128} (rounding up fractions)</td>
</tr>
<tr>
<td></td>
<td>Specify “0” when &lt;range&gt; is set to ALL.</td>
</tr>
<tr>
<td>&lt;data&gt;</td>
<td>&lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
</tr>
<tr>
<td></td>
<td>0 Fills “0” in specified range.</td>
</tr>
<tr>
<td></td>
<td>1 Fills “1” in specified range.</td>
</tr>
</tbody>
</table>

Function

Sets 0s or 1s for the specified page or all data in the program of block during Data pattern reception.

Example

To set 0s to the one page in the program of three blocks during Data pattern reception:

> :SENSe:PATTern:MIXData:DATA:FILL 3,PAGE,1,0

Compatibility

Partially compatible with MU181040A/B.
7.12.4.6 Mask setting commands

Figure 7.12.4.6-1 Mask ON/OFF Setting

Figure 7.12.4.6-2 Bit Window Setup Dialog Box (For PRBS Pattern)

Figure 7.12.4.6-3 Bit Window and Block Window Settings in the Pattern Editor Dialog Box (For Zero Substitution, Data Patterns)
Table 7.12.4.6-1  Mask Setting Commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>:SENSe:PAThern:MASK:BLOCK?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:PAThern:MASK:BIT?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:PAThern:MASK:EXTERNAL?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:PAThern:MASK:BLKWindow?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:PAThern:MASK:BTWindow?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:PAThern:MASK:BDATA:WHOLE?</td>
</tr>
<tr>
<td>[8]</td>
<td>Block Window (Inverting data)</td>
<td>:SENSe:PAThern:MASK:REVerse:DELTa</td>
</tr>
<tr>
<td>[9]</td>
<td>Block Window (All 0 or 1)</td>
<td>:SENSe:PAThern:MASK:DATA:FILL</td>
</tr>
</tbody>
</table>

**:SENSe:PAThern:MASK:BLOCK <boolean>**

Parameter : `<boolean> = <BOOLEAN PROGRAM DATA>`
- 0 or OFF: Block Window OFF
- 1 or ON: Block Window ON

Function: Sets On/OFF of Block Window to received pattern.

Example: To set Block Window ON to received pattern:
> :SENSe:PAThern:MASK:BLOCK ON

Compatibility: Compatible with MU181040A/B.

**:SENSe:PAThern:MASK:BLOCK?**

Response : `<numeric> = <NR1 NUMERIC RESPONSE DATA>`
- 0: Block Window OFF
- 1: Block Window ON

Function: Queries On/OFF of Block Window to received pattern.

Example: To query Block Window status:
> :SENSe:PAThern:MASK:BLOCK?
< 1

Compatibility: Compatible with MU181040A/B.
Chapter 7  SCPI Commands

:SENSe:PATTern:MASK:BIT <boolean>
Parameter  
<boolean> = <BOOLEAN PROGRAM DATA>
0 or OFF  
Bit Window OFF
1 or ON  
Bit Window ON
Function  
Sets On/Off of Bit Window to received pattern.
Example  
To set Bit Window ON to received pattern:
> :SENSe:PATTern:MASK:BIT ON
Compatibility  
Compatible with MU181040A/B.

:SENSe:PATTern:MASK:BIT?
Response  
<numeric> = <NR1 NUMERIC RESPONSE DATA>
0  
Bit Window OFF
1  
Bit Window ON
Function  
Queries On/Off of Bit Window to received pattern.
Example  
> :SENSe:PATTern:MASK:BIT?
< 1
Compatibility  
Compatible with MU181040A/B.

:SENSe:PATTern:MASK:EXTernal <boolean>
Parameter  
<boolean> = <BOOLEAN PROGRAM DATA>
0 or OFF  
External Mask OFF
1 or ON  
External Mask ON
Function  
Sets On/Off of External Mask to received pattern
Example  
To set External Mask ON to received pattern:
> :SENSe:PATTern:MASK:EXTernal ON
Compatibility  
Compatible with MU181040A/B.

:SENSe:PATTern:MASK:EXTernal?
Response  
<numeric> = <NR1 NUMERIC RESPONSE DATA>
0  
External Mask OFF
1  
External Mask ON
Function  
Queries On/Off of External Mask to received pattern.
Example  
> :SENSe:PATTern:MASK:EXTernal?
< 1
Compatibility  
Compatible with MU181040A/B.
:SENSe:PATTern:MASK:BLKWindow <start>,<end>,<data>

Parameter

- **<start>** = <NON-DECIMAL PROGRAM DATA>
  - #H0 to #H1FFFFF 0 to 1FFFFF bits/1 bit Step
    (Specify in hexadecimal.)

- **<end>** = <NON-DECIMAL PROGRAM DATA>
  - #H0 to #H1FFFFF 0 to 1FFFFF bits/1 bit Step
    (Specify in hexadecimal.)

- **<data>** = <STRING PROGRAM DATA>
  - "H***...*" Specify pattern data in hexadecimal format.
    1 to 400 characters (pattern data of 400 × 4 bits)
    Specify a character string consisting of 0 to 9
    and A to F.
  - "B***...*" Specify pattern data in binary format.
    1 to 400 characters (pattern data of 400 bits)
    Specify a character string consisting of 0 and 1.

Function

Sets the mask pattern data from the <Start> to <end> addresses while setting Block Window.
The set data overwrites the specified range.

Example

To set the mask pattern data by bit address from 0 to 1FFFFF for Block Window setting in hexadecimal:

```
> :SENSe:PATTern:MASK:BLKWindow #H0,#H1FFFFF, "H001"
```

Compatibility

Incompatible with existing models. Operation is different from MU181040A/B.

Notes:

The maximum allowable value for Block Window is 2 Mbits (2,097,152bits).

The setting step varies depending on the Pattern Length setting.

<table>
<thead>
<tr>
<th>Pattern Length setting</th>
<th>Block Window step</th>
</tr>
</thead>
<tbody>
<tr>
<td>2^N</td>
<td>2,097,152*N bits</td>
</tr>
<tr>
<td>2,097,153^N</td>
<td>4,194,304*N bits</td>
</tr>
<tr>
<td>4,194,305^N</td>
<td>8,388,608*N bits</td>
</tr>
<tr>
<td>8,388,609^N</td>
<td>16,777,216*N bits</td>
</tr>
<tr>
<td>16,777,217^N</td>
<td>33,554,432*N bits</td>
</tr>
<tr>
<td>33,554,433^N</td>
<td>67,108,864*N bits</td>
</tr>
<tr>
<td>67,108,864^N</td>
<td>134,217,728*N bits</td>
</tr>
<tr>
<td>134,217,729^N</td>
<td>268,435,456*N bits</td>
</tr>
</tbody>
</table>

In addition, relation between pattern length and step is increased by N times according to the Combination setting.

- Independent: N = 1
- 2ch Combination: N = 2
- 4ch Combination: N = 4
Chapter 7  SCPI Commands

:SENSe:PATTern:MASK:BLKWindow? <start>

Parameter  
\(<start> = \langle\text{NON-DECIMAL PROGRAM DATA}\rangle \)
\#H0 to \#H1FFFFFF 0 to 1FFFFF bits/1 bit Step  
(Specify in hexadecimal.)

Response  
\(<\text{data}> = \langle\text{STRING RESPONSE DATA}\rangle\)
"H***...*" Returns the pattern data in hexadecimal format.  
4 to 400 characters (pattern data of 400 \times 4 bits)

Function  
Queries the mask pattern data at the time of a Block Window setup by 400 \times 4 bits from the \(<\text{start}>\)address.

Example  
To query the mask pattern data at the time of a Block Window setup by 400 \times 4 bits from the 0 bit address:
\[ > :\text{SENSe}:\text{PATTern}:\text{MASK}:\text{BLKWindow}? \#H0 \]
\[ < \"H001\" \]

Compatibility  
Incompatible with existing models MU181040A/B.

:SENSe:PATTern:MASK:BTWindow <data>

Parameter  
\(<\text{data}> = \langle\text{STRING PROGRAM DATA}\rangle \)
"H***...*" Specify pattern data in hexadecimal format.  
1 to 8 characters (pattern data of 32 bits)
Specify a character string consisting of 0 to 9 and A to F.
"B***...*" Specify pattern data in binary format.  
1 to 32 characters (pattern data of 32 bits)
Specify a character string consisting of 0 and 1.

Function  
Sets Mask pattern data of Bit Window setting.

Example  
To set the mask pattern data of Bit Window setting in hexadecimal:
\[ > :\text{SENSe}:\text{PATTern}:\text{MASK}:\text{BTWindow} \"H001\" \]

Compatibility  
Incompatible with existing models MU181040A/B.

:SENSe:PATTern:MASK:BTWindow?

Response  
\(<\text{data}> = \langle\text{STRING RESPONSE DATA}\rangle\)
"H***...*" Returns the pattern data in hexadecimal format.  
Max. 8 characters (pattern data of 32 bits)

Function  
Queries Mask pattern data of Bit Window setting.

Example  
\[ > :\text{SENSe}:\text{PATTern}:\text{MASK}:\text{BTWindow}? \]
\[ < \"H00100100\" \]

Compatibility  
Incompatible with existing models MU181040A/B.
### :SENSe:PATTern:MASK:BDATa:WHOLe <start>,<end>,<bdata>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;start&gt;</td>
<td><code>&lt;NON-DECIMAL PROGRAM DATA&gt;</code>&lt;br&gt;<code>#H0 to #H1FFFFF</code> 0 to 1FFFFF bits / 1 bit Step&lt;br&gt;(Specify in hexadecimal.)</td>
</tr>
<tr>
<td>&lt;end&gt;</td>
<td><code>&lt;NON-DECIMAL PROGRAM DATA&gt;</code>&lt;br&gt;<code>#H0 to #H1FFFFF</code> 0 to 1FFFFF bits / 1 bit Step&lt;br&gt;(Specify in hexadecimal.)</td>
</tr>
<tr>
<td>&lt;bdata&gt;</td>
<td><code>&lt;ARBITRARY BLOCK PROGRAM DATA&gt;</code>&lt;br&gt;<code>#XYYY&lt;binary&gt;</code>&lt;br&gt;<code>X:</code> Number of YYY digits&lt;br&gt;<code>YYY:</code> Number of bytes of <code>&lt;binary&gt;</code>&lt;br&gt;<code>1 to 16,777,216 (bytes)</code>&lt;br&gt;<code>&lt;binary&gt;</code>: Binary data up to 16,777,216 byte max.</td>
</tr>
</tbody>
</table>

**Function**

Sets binary mask pattern data from the `<start>` address to the `<end>` address when **Block Window is ON**. The set data overwrites the specified range.

**Example**

To set binary mask pattern data of the bit addresses from 0 to 7FFFFF when **Block Window is ON**:

```
> :SENSe:PATTern:MASK:BDATa:WHOLe #H0,#H7FFFF,#11A
```

**Compatibility**

Incompatible with existing models MU181040A/B.
**Chapter 7  SCPI Commands**

**:SENSe:PATTern:MASK:BDATa:WHOLe? <start>[,<size>]**

**Parameter**
- `<start>` = `<NON-DECIMAL PROGRAM DATA>`
  - #H0 to #H1FFFFF 0 to 1FFFFF bits / 1 bit Step
  - (Specify in hexadecimal.)

- `[<size>]` = `<NR1 NUMERIC PROGRAM DATA>`
  - 1 to 16777216 1 to 16,777,216 bytes / 1 byte Step

**Response**
- `<bdata>` = `<DEFINITE LENGTH ARBITRARY BLOCK RESPONSE DATA>`
  - #XYYY<binary>
    - X: digits of YYY
    - YYY: Number of bytes of `<binary>`:
      - 1 to 16,777,216 (bytes)
  - `<binary>`: Binary data up to 16,777,216 bytes

Binary data of the setting pattern is queried when `[<size>]` is omitted.

**Function**
Queries binary data of bytes specified by `<size>` from the `<start>` address of the mask pattern data while setting Block Window.

**Example**
To query binary data of 2 bytes from bit 0 address of the pattern data while setting Block Window:
- `> :SENSe:PATTern:MASK:BDATa:WHOLe? #H0,2`
  - `< #12AA`

To query binary data of the setting pattern from bit 0 address of the pattern data while setting Block Window (in the case of `[<size>]` is omitted):
- `> :SENSe:PATTern:MASK:BDATa:WHOLe? #H0`
  - `< #516000AAAAAA…… (binary data of 16,777,216 bytes)`

**Compatibility**
Incompatible with existing models MU181040A/B.

**:SENSe:PATTern:MASK:DREVerse:ADDRess <start>,<end>**

**Parameter**
- `<start>` = `<NON-DECIMAL PROGRAM DATA>`
  - #H0 to #H1FFFFF 0 to 1FFFFF bits / 1 bit Step
  - (Specify in hexadecimal.)

- `<end>` = `<NON-DECIMAL PROGRAM DATA>`
  - #H0 to #H1FFFFF 0 to 1FFFFF bits / 1 bit Step
  - (Specify in hexadecimal.)

**Function**
Inverts the mask pattern data while setting Block Window. Specify the inversion range by the `<start>` and `<end>` addresses.

**Example**
To invert addresses from bit 0 to bit 7FFFF while setting Block Window:
- `> :SENSe:PATTern:MASK:DREVerse:ADDRess #H0,#H7FFFF`

**Compatibility**
Incompatible with existing models MU181040A/B.
### :SENSe:PATTern:MASK:DREVerse:DELTa <start>,<delta>

**Parameter**
- `<start>` = <NON-DECIMAL PROGRAM DATA>
  - `#H0` to `#H1FFFFF` 0 to `1FFFFF` bits / 1 bit Step
    (Specify in hexadecimal.)
- `<delta>` = <DECIMAL NUMERIC PROGRAM DATA>
  - 1 to 2097152 1 to 2,097,152 bits / 1 bit Step

**Function**
Inverts the mask pattern data while setting Block Window. Specify the inversion range by the number of bits from the `<start>` address (relative specification).

**Example**
To invert 1 bit of the mask pattern data from address 0 while setting Block Window:
> :SENSe:PATTern:MASK:DREVerse:DELTa #H0,1

**Compatibility**
Incompatible with existing models MU181040A/B.

### :SENSe:PATTern:MASK:DATA:FILL <range>,<page>,<data>

<table>
<thead>
<tr>
<th>Parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;range&gt;</code> = &lt;CHARACTER PROGRAM DATA&gt;</td>
</tr>
<tr>
<td>PAGE</td>
</tr>
<tr>
<td>ALL</td>
</tr>
<tr>
<td><code>&lt;page&gt;</code> = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
</tr>
<tr>
<td>1 to 2097152</td>
</tr>
<tr>
<td>Max = ( \frac{\text{Pattern Length}}{128} ) (round up)</td>
</tr>
<tr>
<td>Specify “0” when <code>&lt;range&gt;</code> is ALL.</td>
</tr>
<tr>
<td><code>&lt;data&gt;</code> = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
</tr>
<tr>
<td>0</td>
</tr>
<tr>
<td>1</td>
</tr>
</tbody>
</table>

**Function**
Sets the specified page or all data of mask pattern data to 0 or 1 while setting Block Window.

**Example**
To set page 1 of mask pattern data to 0 while setting Block Window:
> :SENSe:PATTern:MASK:DATA:FILL PAGE,1,0

**Compatibility**
Incompatible with existing models MU181040A/B.
7.12.5 Commands Related to Input Tab

Figure 7.12.5-1  Input Tab
7.12.5.1 Data Input setting commands

Figure 7.12.5.1-1 Data Setting (When Single-Ended Is Selected)

Figure 7.12.5.1-2 Data Setting (When Differential 50 Ohm Is Selected)
### Table 7.12.5.1-1  Data Input Setting Commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>:INPut:DATA:INTerface?</td>
</tr>
</tbody>
</table>
| [2] | No label
(Differential type setting)                         | :INPut:DATA:DIFFerential          |
|     |                                                           | :INPut:DATA:DIFFerential?         |
| [3] | No label
(Data input port setting)                             | :INPut:DATA:SINGle                |
|     |                                                           | :INPut:DATA:SINGle?               |
|     | XData Threshold                                          | :INPut:DATA:THReshold?            |
| [5] | No label
(Reference signal setting of differential input)       | :INPut:DATA:DIFFerential:BASis    |
|     |                                                           | :INPut:DATA:DIFFerential:BASis?   |
| [6] | No label
(Threshold setting of differential input)               | :INPut:DATA:DIFFerential:THReshold|
|     |                                                           | :INPut:DATA:DIFFerential:THReshold? |
|     |                                                           | :INPut:DATA:TERMination?          |
| [8] | No label
(Termination level setting)                             | :INPut:DATA:TLEVel                |
|     |                                                           | :INPut:DATA:TLEVel?               |
### :INPut:DATA:INTerface <interface>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;interface&gt; = &lt;CHARACTER PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>SINGle</td>
<td>Single Ended</td>
</tr>
<tr>
<td>DIF50ohm</td>
<td>Differential 50 Ω</td>
</tr>
<tr>
<td>DIF100ohm</td>
<td>Differential 100 Ω</td>
</tr>
</tbody>
</table>

**Function**
Sets the data input interface.

**Example**
To set the data input interface to Single Ended:
> :INPut:DATA:INTerface SINGle

**Compatibility**
Compatible with MU181040A/B.

### :INPut:DATA:INTerface?

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;interface&gt; = &lt;CHARACTER RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>SING, DIF50, DIF100</td>
<td></td>
</tr>
</tbody>
</table>

**Function**
Queries the data input interface.

**Example**
> :INPut:DATA:INTerface?
< SING

**Compatibility**
Compatible with MU181040A/B.

### :INPut:DATA:DIFFerential <item>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;item&gt; = &lt;CHARACTER PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>INDependent</td>
<td>Data/Xdata independent setting</td>
</tr>
<tr>
<td>TRACking</td>
<td>Data/Xdata common setting</td>
</tr>
<tr>
<td>ALTernate</td>
<td>Data/Xdata differential setting</td>
</tr>
</tbody>
</table>

**Function**
Sets the differential type when the data input interface is differential.

**Example**
To set the differential type to Data/Xdata independent setting:
> :INPut:DATA:DIFFerential INDependent

**Compatibility**
Compatible with MU181040A/B.

### :INPut:DATA:DIFFerential?

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;item&gt; = &lt;CHARACTER RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>IND, TRAC, ALT</td>
<td></td>
</tr>
</tbody>
</table>

**Function**
Queries the differential type when the data input interface is set to Differential.

**Example**
> :INPut:DATA:DIFFerential?
< IND

**Compatibility**
Compatible with MU181040A/B.
Chapter 7  SCPI Commands

:INPut:DATA:SINGle <item>
Parameter <item> = <CHARACTER PROGRAM DATA>
Data          Selects Data.
XDATa         Selects XData.
Function      Sets the input port when the data input interface is set to Single.
Example       To set the input port to Data:
               > :INPut:DATA:SINGle DATA
Compatibility  Compatible with MU181040A/B.

:INPut:DATA:SINGle?
Response <item> = <CHARACTER RESPONSE DATA>
Data, XDAT
Function     Queries the input port when the data input interface is set to Single.
Example      > :INPut:DATA:SINGle?
< DATA
Compatibility  Compatible with MU181040A/B.

:INPut:DATA:THReshold <port>,<numeric>
Parameter <port> = <CHARACTER PROGRAM DATA>
Data          Selects Data.
XDATa         Selects XData.
<numeric> = <DECIMAL NUMERIC PROGRAM DATA>
–3.500 to 3.300  –3.500 to 3.300 V/0.001 V Step
Function      Sets the data input threshold value for the specified input port.
Example       To set the data input threshold value for the Data port to –3.000 V:
               > :INPut:DATA:THReshold DATA,-3.000
Compatibility  Compatible with MU181040A/B.

:INPut:DATA:THReshold? <port>
Parameter <port> = <CHARACTER PROGRAM DATA>
Data          Selects Data.
XDATa         Selects XData.
Response <numeric> = <NR2 NUMERIC RESPONSE DATA>
–3.500 to 3.300  –3.500 to 3.300 V
Function      Queries the data input threshold value for the specified input port.
Example       To query the data input threshold value for the Data port:
               > :INPut:DATA:THReshold? DATA
< -3.000
Compatibility  Compatible with MU181040A/B.
7.12 28G/32G bit/s ED Commands

:INPut:DATA:DIFFerential:BASis <basis>

Parameter  
<basis> = <CHARACTER PROGRAM DATA>

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATA</td>
<td>Sets the differential reference signal to Data.</td>
</tr>
<tr>
<td>XDATa</td>
<td>Sets the differential reference signal to XData.</td>
</tr>
</tbody>
</table>

Function
Sets the differential reference signal for the data input threshold value.

Example
To set the differential reference signal for the data input threshold value to DATA:

> :INPut:DATA:DIFFerential:BASis DATA

Compatibility
Compatible with MU181040A/B.

:INPut:DATA:DIFFerential:BASis?

Response  
<basis> = <CHARACTER RESPONSE DATA>

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATA</td>
<td>XDAT</td>
</tr>
</tbody>
</table>

Function
Queries the differential reference signal for the data input threshold value.

Example

> :INPut:DATA:DIFFerential:BASis?
< DATA

Compatibility
Compatible with MU181040A/B.

:INPut:DATA:DIFFerential:THReshold <numeric>

Parameter  
<numeric> = <DECIMAL NUMERIC PROGRAM DATA>

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-3.000  to 3.000 V/0.001 V Step</td>
<td></td>
</tr>
</tbody>
</table>

Function
Sets the data input threshold value for differential input.

Example
To set the data input threshold value to -3.000 V:

> :INPut:DATA:DIFFerential:THReshold -3.000

Compatibility
Compatible with MU181040A/B.

:INPut:DATA:DIFFerential:THReshold?

Response  
<numeric> = <NR2 NUMERIC RESPONSE DATA>

Function
Queries the data input threshold value for differential input.

Example

> :INPut:DATA:DIFFerential:THReshold?
< -3.000

Compatibility
Compatible with MU181040A/B.
### :INPut:DATA:TERMination <term>

**Parameter**
- `<term> = <CHARACTER PROGRAM DATA>`
- GND: 50 Ω to GND
- VARiable: 50 Ω to Variable Voltage

**Function**
Sets the data input termination condition.

**Example**
To set the data input termination condition to GND:
```
> :INPut:DATA:TERMination GND
```

**Compatibility**
Partially compatible with MU181040A/B.

### :INPut:DATA:TERMination?

**Response**
- `<term> = <CHARACTER RESPONSE DATA>`
- GND, VAR

**Function**
Queries the data input termination condition.

**Example**
```
> :INPut:DATA:TERMination?
< GND
```

**Compatibility**
Partially compatible with MU181040A/B.

### :INPut:DATA:TLEVel <numeric>

**Parameter**
- `<numeric> = <DICIMAL NUMERIC PROGRAM DATA>`
- –2.50 to 3.50 –2.50 to 3.50 V/0.01 V Step

**Function**
Sets the termination voltage when the input termination condition is set to Variable.
The setting is invalid when the data input interface is Differential 100 Ω.

**Example**
To set the termination voltage when the input termination condition is set to Variable to –2.00 V:
```
> :INPut:DATA:TLEVel -2.00
```

**Compatibility**
Compatible with MU181040A/B.

### :INPut:DATA:TLEVel?

**Response**
- `<numeric> = <NR2 NUMERIC RESPONSE DATA>`
- –2.50 to 3.50 –2.50 to 3.50 V

**Function**
Queries the termination voltage when the input termination condition is set to Variable.

**Example**
```
> :INPut:DATA:TLEVel?
< -2.00
```

**Compatibility**
Compatible with MU181040A/B.
7.12.5.2 Clock Input setting commands

Figure 7.12.5.2-1 Clock Setting (When External Clock Is Selected)

Figure 7.12.5.2-2 Clock Setting (Option 22 When Recovered Clock Is Selected)

Figure 7.12.5.2-3 Clock Setting (Option 23 When Recovered Clock Is Selected)
Table 7.12.5.2-1  Clock Input Setting Commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>:INPut:CLOCK:SELection?</td>
</tr>
<tr>
<td></td>
<td>Recovery Preset</td>
<td>:INPut:CLOCK:RECovery?</td>
</tr>
<tr>
<td></td>
<td>Operation Bitrate</td>
<td>:INPut:CLOCK:CRFReq?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:INPut:CLOCK:LBANdwidth?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:INPut:CLOCK:LBW?</td>
</tr>
<tr>
<td></td>
<td>Tracking</td>
<td>:INPut:CLOCK:PTRacking?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:INPut:CLOCK:DELay?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:INPut:CLOCK:RELative?</td>
</tr>
<tr>
<td></td>
<td>(Relative value setting)</td>
<td>:INPut:CLOCK:RDELay?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:INPut:CLOCK:JINPut</td>
</tr>
</tbody>
</table>

:INPut:CLOCK:SELection <sel>

Parameter  
<sel> = <CHARACTER RESPONSE DATA>
RECovered   Recovered Clock
EXTernal    External Clock

Function  
Sets the type of the input clock.
To set the type of the input clock to Recovered Clock:
Example  
> :INPut:CLOCK:SELection RECovered
Compatibility  
Partially compatible with MU181040A/B.

:INPut:CLOCK:SELection?

Response  
<sel> = <CHARACTER RESPONSE DATA>
REC   Recovered Clock
EXT   External Clock

Function  
Queries the clock input type.
Example  
> :INPut:CLOCK:SELection?
< EXT
Compatibility  
Partially compatible with MU181040A/B.
### :INPut:CLOCk:RECovery <freq>

The preset frequencies vary depending on your option.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;Variable&quot;</td>
<td>Variable</td>
</tr>
<tr>
<td>&quot;OC_48&quot;</td>
<td>OC48/STM16: 2.488320 Gbit/s</td>
</tr>
<tr>
<td>&quot;PCI_EX1&quot;</td>
<td>PCI Express Gen1: 2.500000 Gbit/s</td>
</tr>
<tr>
<td>&quot;IB_SDR&quot;</td>
<td>Infiniband SDR: 2.500000 Gbit/s</td>
</tr>
<tr>
<td>&quot;OC_48_G709&quot;</td>
<td>OTU1: 2.666060 Gbit/s</td>
</tr>
<tr>
<td>&quot;SATA3&quot;</td>
<td>SATA 3Gb/s: 3.000000 Gbit/s</td>
</tr>
<tr>
<td>&quot;XAUI&quot;</td>
<td>XAUI: 3.125000 Gbit/s</td>
</tr>
<tr>
<td>&quot;4G_FC&quot;</td>
<td>4GFC: 4.250000 Gbit/s</td>
</tr>
<tr>
<td>&quot;PCI_EX2&quot;</td>
<td>PCI Express Gen2: 5.000000 Gbit/s</td>
</tr>
<tr>
<td>&quot;USB3&quot;</td>
<td>USB3.0: 5.000000 Gbit/s</td>
</tr>
<tr>
<td>&quot;IB_DDR&quot;</td>
<td>Infiniband DDR: 5.000000 Gbit/s</td>
</tr>
<tr>
<td>&quot;SATA6&quot;</td>
<td>SATA 6Gb/s: 6.000000 Gbit/s</td>
</tr>
<tr>
<td>&quot;HSBI&quot;</td>
<td>HSBI: 6.250000 Gbit/s</td>
</tr>
<tr>
<td>&quot;PCI_EX3&quot;</td>
<td>PCI Express Gen3: 8.000000 Gbit/s</td>
</tr>
<tr>
<td>&quot;8G_FC&quot;</td>
<td>8GFC: 8.500000 Gbit/s</td>
</tr>
<tr>
<td>&quot;OC_192&quot;</td>
<td>OC192/STM64: 9.953280 Gbit/s</td>
</tr>
<tr>
<td>&quot;IB_QDR&quot;</td>
<td>Infiniband QDR: 10.000000 Gbit/s</td>
</tr>
<tr>
<td>&quot;10GbE&quot;</td>
<td>10GbE: 10.312500 Gbit/s</td>
</tr>
<tr>
<td>&quot;10G_FC&quot;</td>
<td>10GFC: 10.518750 Gbit/s</td>
</tr>
<tr>
<td>&quot;OC_192_G975&quot;</td>
<td>G975 FEC: 10.664228 Gbit/s</td>
</tr>
<tr>
<td>&quot;OC_192_G709&quot;</td>
<td>OTU2: 10.709225 Gbit/s</td>
</tr>
<tr>
<td>&quot;10GbE_G709&quot;</td>
<td>10GbE over FEC: 11.095700 Gbit/s</td>
</tr>
<tr>
<td>&quot;10G_FC_G709&quot;</td>
<td>10GFC over FEC: 11.316800 Gbit/s</td>
</tr>
<tr>
<td>&quot;16G_FC&quot;</td>
<td>16GFC: 14.025000 Gbit/s</td>
</tr>
<tr>
<td>&quot;IB_FDR&quot;</td>
<td>Infiniband_FDR: 14.062500 Gbit/s</td>
</tr>
<tr>
<td>&quot;PCI_EX4&quot;</td>
<td>PCI Express Gen4: 16.000000 Gbit/s</td>
</tr>
<tr>
<td>&quot;SAS&quot;</td>
<td>SAS: 24.000000 Gbit/s</td>
</tr>
<tr>
<td>&quot;IB_EDR&quot;</td>
<td>Infiniband_EDR: 25.781250 Gbit/s</td>
</tr>
<tr>
<td>&quot;100GbE&quot;</td>
<td>100GbE (25.78x4): 25.781250 Gbit/s</td>
</tr>
<tr>
<td>&quot;100G_OTU4&quot;</td>
<td>100G OTU4: 27.952496 Gbit/s</td>
</tr>
<tr>
<td>&quot;32G_FC&quot;</td>
<td>32GFC: 28.050000 Gbit/s</td>
</tr>
</tbody>
</table>

When the Option-23 is installed:

| "Variable" | Variable |
| "IB_EDR" | Infiniband_EDR: 25.781250 Gbit/s |
| "100GbE" | 100GbE (25.78x4): 25.781250 Gbit/s |
| "100G_OTU4" | 100G OTU4: 27.952496 Gbit/s |
| "32G_FC" | 32GFC: 28.050000 Gbit/s |
| "100G_ULH" | 100G ULH: 32.100000 Gbit/s |

**Note:**

The preset frequencies vary depending on your option.
Chapter 7  SCPI Commands

Function
Sets the preset frequency of the clock input that is set to Recovered Clock.

Example
To set the preset frequency to “16G_FC”:
> :INPut:CLOCk:RECovery "16G_FC"

Compatibility
Partially compatible with MU181040A/B.

:INPut:CLOCk:RECovery?

Response
<freq> = <STRING RESPONSE DATA>
"OC_48", "PCI_EX1", "IB_SDR", "OC_48_G709", "SATA3", "XAUI",
"4G_FC", "PCI_EX2", "USB3", "IB_DDR", "SATA6", "HSBI",
"PCI_EX3", "8G_FC", "OC_192", "10GbE", "10G_FC",
"16G_FC", "IB_FDR", "PCI_EX4", "SAS", "IB_EDR", "100GbE",
"100G_OTU", "32G_FC", "100G_ULH", "Variable"

Function
Queries the preset frequency of the clock input that is set to Recovered Clock.

Example
> :INPut:CLOCk:RECovery?
< "16G_FC"

Compatibility
Partially compatible with MU181040A/B.

:INPut:CLOCk:CRFReq <numeric>

Parameter
<numeric> = <DECIMAL NUMERIC PROGRAM DATA>
When the Option-22 is installed:
2.400000 to 28.100000  2.40000 to 28.10000 Gbit/s /1 kbit/s Step
When the Option-23 is installed:
25.500000 to 32.100000  25.50000 to 32.10000 Gbit/s /1 kbit/s Step

Function
Set the frequency for Clock Recovery.

Example
To set the frequency for Clock Recovery to 28.100000 Gbit/s:
> :INPut:CLOCk:CRFReq 28.100000

Compatibility
Partially compatible with MU181040A/B.
7.12 28G/32G bit/s ED Commands

:INPut:CLOCk:CRFReq?

Response

<numeric> = <NR2 NUMERIC RESPONSE DATA>
When the Option-22 is installed:
2.400000 to 28.100000
When the Option-23 is installed:
25.500000 to 32.100000

Function
Queries the frequency for Clock Recovery.

Example
> :INPut:CLOCk:CRFReq?
< 28.100000

Compatibility
Partially compatible with MU181040A/B.

:INPut:CLOCk:LBANdwidth <numeric>

Parameter

<numeric> = <DECIMAL NUMERIC PROGRAM DATA>
When the Option-22 is installed:

<table>
<thead>
<tr>
<th>Operation Bitrate [Gbit/s]</th>
<th>Range [MHz] (Step: 1 MHz)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.400 000 to 5.500 000</td>
<td>Fixed to 3 MHz</td>
</tr>
<tr>
<td>5.500 001 to 7.500 000</td>
<td>3 to 4 MHz</td>
</tr>
<tr>
<td>7.500 001 to 9.500 000</td>
<td>3 to 5 MHz</td>
</tr>
<tr>
<td>9.500 001 to 10.500 000</td>
<td>3 to 6 MHz</td>
</tr>
<tr>
<td>10.500 001 to 12.500 000</td>
<td>3 to 7 MHz</td>
</tr>
<tr>
<td>12.500 001 to 14.500 000</td>
<td>3 to 8 MHz</td>
</tr>
<tr>
<td>14.500 001 to 15.500 000</td>
<td>3 to 9 MHz</td>
</tr>
<tr>
<td>15.500 001 to 17.500 000</td>
<td>3 to 10 MHz</td>
</tr>
<tr>
<td>17.500 001 to 19.500 000</td>
<td>3 to 11 MHz</td>
</tr>
<tr>
<td>19.500 001 to 20.500 000</td>
<td>3 to 12 MHz</td>
</tr>
<tr>
<td>20.500 001 to 22.500 000</td>
<td>3 to 13 MHz</td>
</tr>
<tr>
<td>22.500 001 to 24.500 000</td>
<td>3 to 14 MHz</td>
</tr>
<tr>
<td>24.500 001 to 25.500 000</td>
<td>3 to 15 MHz</td>
</tr>
<tr>
<td>25.500 001 to 27.500 000</td>
<td>3 to 16 MHz</td>
</tr>
<tr>
<td>27.500 001 to 28.100 000</td>
<td>3 to 17 MHz</td>
</tr>
</tbody>
</table>

When the Option-23 is installed, this command is not required.

Function
Sets the loop band for Clock Recovery.

Example
To set the loop band for Clock Recovery to 8 MHz:
> :INPut:CLOCk:LBANdwidth 8

Compatibility
Incompatible with existing models.
### Chapter 7  SCPI Commands

#### :INPut:CLOCk:LBANdwidth?

**Response**

<numeric> = <NR2 NUMERIC RESPONSE DATA>

When the Option-22 is installed:
- 3 to 17 MHz

When the Option-23 is installed:
- XX.XX MHz (Displayed to two decimal places)
  - Operation Bitrate/1667 MHz
  - or Operation Bitrate/2578 MHz

**Function**
Queries the loop band for Clock Recovery.

**Example**

> :INPut:CLOCk:LBANdwidth?
< 8 (When the Option-22 is installed)
16.85 (When the Option-23 is installed)

**Compatibility**
Incompatible with existing models.

#### :INPut:CLOCk:LBW <sel>

**Parameter**

<sel> = <CHARACTER PROGRAM DATA>

- B_1667: Sets the LBW[MHz] to Bitrate [GBit/s] / 1667.
- B_2578: Sets the LBW[MHz] to Bitrate [GBit/s] / 2578.
- JTOLerance: Sets the LBW to the maximum value for the Jitter Tolerance measurement.
- VARiable: Sets an arbitrary value to the LBW[MHz].

**Note:**

VARiable is available only when the Option-22 is installed.

**Function**
Selects a jitter transmission loop band (LBW[MHz]) setting.

**Example**

To set the jitter transmission loop band (LBW[MHz]) to Bitrate/1667:

> :INPut:CLOCk:LBW 1667

**Compatibility**
Incompatible with existing models.

#### :INPut:CLOCk:LBW?

**Response**

<sel> = <CHARACTER RESPONSE DATA>

- B_1667, B_2578, JTOL, VAR

**Function**
Queries the jitter transmission loop band (LBW[MHz]) setting.

**Example**

> :INPut:CLOCk:LBW?
< 1667

**Compatibility**
Incompatible with existing models.
### :INPut: CLOCk: PTRacking <input>

**Parameter**
- `<input>` = `<DECIMAL NUMERIC PROGRAM DATA>`
- OFF
- INTeral[1 to 8] = MU183020A/21A

**Note:**
INTernal[1 to 8] specifies the unit and slot numbers from the smallest as 1 to 8 when there are multiple 32G PPGs installed to MP1800A/MT1810A. If [1 to 8] is omitted, 1 is assumed. Tracking is available only when the tracking target MU183020A/21A is interworking with either internal synthesizer or jitter generation source.

**Function**
Sets the Operation Bitrate of 32G PPG to the Operation Bitrate of MP1800A/MT1810A.

**Example**
To set the Operation Bitrate of 32G PPG installed to Slot2:
```
> :INPut:CLOCk:PTRacking INT2
```

**Compatibility**
Incompatible with existing models.

### :INPut: CLOCk: PTRacking?

**Response**
- `<numeric>` = `<NR2 NUMERIC RESPONSE DATA>`
- OFF, INT1 to 8
- `<info>` = `<STRING RESPONSE DATA>`
- "X:Y Z"
  - X: Unit No.1 to 4
  - Y: Slot No.1 to 6
  - Z: Module name MU183020A/21A

**Function**
Queries the MU183020A/21A that is being tracked.

**Example**
```
> :INPut:CLOCk:PTRacking?
< INT2,"1:2 MU183020A"
```

**Compatibility**
Incompatible with existing models.
### :INPut:CLOCk:DELay <numeric>[,<unit>]

**Parameter**

- **<numeric>** = `<DECIMAL NUMERIC PROGRAM DATA>`
  - In mUI units: \(-1000\) to \(1000\)
  - In ps units: Converted from the calculated frequency value, based on the setting resolution of mUI.

- **<unit>** = `<CHARACTER PROGRAM DATA>`
  - UI mUI unit
  - PS ps unit
  - (The ps unit is selected when <unit> is omitted.)

**Function**

Sets the value and unit of the clock input phase variable.

The specified value may not be exactly set as is when the ps unit is set, due to the limitation of the setting resolution. The value is adjusted to the most approximate value and set in this event.

**Example**

To set the clock input phase variable to \(-1000\) mUI:

```
> :INPut:CLOCk:DELay -1000,UI
```

**Compatibility**

Compatible with MU181040A/B.

### :INPut:CLOCk:DELay? [<unit>]

**Parameter**

- **<unit>** = `<CHARACTER PROGRAM DATA>`
  - UI mUI unit
  - PS ps unit
  - (The ps unit is selected when <unit> is omitted.)

**Response**

- **<numeric>** = `<NR1 NUMERIC RESPONSE DATA>`
  - In mUI units: \(-1000\) to \(1000\)
  - In ps units: Converted from the calculated frequency value, based on the setting resolution of mUI.

**Function**

Queries the value and unit of the clock input phase variable.

**Example**

To query the value of the clock input phase variable in UI unit:

```
> :INPut:CLOCk:DELay? UI
< -1000
```

**Compatibility**

Compatible with MU181040A/B.

### :INPut:CLOCk:RELative <boolean>

**Parameter**

- **<boolean>** = `<BOOLEAN PROGRAM DATA>`
  - 0 or OFF Reference OFF
  - 1 or ON Reference ON

**Function**

Sets the clock input phase variable reference ON/OFF.

**Example**

To set the clock input phase variable reference ON:

```
> :INPut:CLOCk:RELative ON
```

**Compatibility**

Compatible with MU181040A/B.
### :INPut:CLOck:RELative?

**Response**
- `<boolean> = NR1 NUMERIC RESPONSE DATA`
  - 0: Reference OFF
  - 1: Reference ON

**Function**
Queries the clock input phase variable reference setting.

**Example**
```
> :INPut:CLOck:RELative?
< 1
```

**Compatibility**
Compatible with MU181040A/B.

### :INPut:CLOck:RDELay <numeric>[,<unit>]

**Parameter**
- `<numeric> = DECIMAL NUMERIC PROGRAM DATA`
  - In mUI units: -2000 to 2000, -2000 to 2000 mUI, in 2 mUI steps
  - In ps units: Converted from the calculated frequency value, based on the setting resolution of mUI.

- `<unit> = CHARACTER PROGRAM DATA`
  - UI: mUI unit
  - PS: ps unit
  - (The ps unit is selected when `<unit>` is omitted.)

**Function**
Sets the value of the clock input phase variable reference.
Set a value using a difference from the reference value.
The specified value may not be exactly set as is when the ps unit is set, due to the limitation of the setting resolution. The value is adjusted to the most approximate value and set in this event.

**Example**
```
To set the clock phase variable reference to –1000 mUI:
> :INPut:CLOck:RDELay –1000,UI
```

**Compatibility**
Compatible with MU181040A/B.


**Chapter 7  SCPI Commands**

### :INPut:CLOCk:RDELay? [<unit>]

**Parameter**

- `<unit> = <CHARACTER PROGRAM DATA>`
- **UI** mUI unit
- **PS** ps unit
  (The ps unit is selected when `<unit>` is omitted.)

**Response**

- `<numeric> = <NR1 NUMERIC RESPONSE DATA>`
- **In mUI units:** –2000 to 2000 –2000 to 2000 mUI
- **In ps units:** Converted from the calculated frequency value, based on the setting resolution of mUI.

**Function**

Queries the value of the clock input phase variable reference.

**Example**

To query the value of the clock input phase variable reference in mUI units.

```
> :INPut:CLOCk:RDELay? UI
< -1000
```

**Compatibility**

Compatible with MU181040A/B.

### :INPut:CLOCk:CALibration

**Function**

Calibrates the input clock phase setting value.

**Example**

```
> :INPut:CLOCk:CALibration
```

**Compatibility**

Compatible with MU181040A/B.

### :INPut:CLOCk:JINPut <boolean>

**Parameter**

- `<boolean> = <BOOLEAN PROGRAM DATA>`
  - 0 or OFF Jitter input OFF setting
  - 1 or ON Jitter input ON setting

**Function**

Sets the Jitter Input button ON/OFF.

**Example**

To set the Jitter Input button to ON when a jitter is added:

```
> :INPut:CLOCk:JINPut ON
```

**Compatibility**

Compatible with MU181040A/B.

### :INPut:CLOCk:JINPut?

**Response**

- `<numeric> = <NR1 NUMERIC RESPONSE DATA>`
  - 0 Jitter input OFF setting
  - 1 Jitter input ON setting

**Function**

Queries the Jitter Input button setting.

**Example**

```
> :INPut:CLOCk:JINPut?
< 1
```

**Compatibility**

Compatible with MU181040A/B.
7.12.5.3 Measurement Restart setting commands

![Measurement Restart Setting](image)

**Table 7.12.5.3-1 Measurement Restart Setting Commands**

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Item</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>:SENSe:MEASure:MREStart?</td>
</tr>
</tbody>
</table>

**:SENSe:MEASure:MREStart <data>,<clock>**

- **Parameter**
  - \(<data> = <BOOLEAN PROGRAM DATA>\)
  - 0 or OFF: Does not restart measurement when the data threshold is changed.
  - 1 or ON: Restarts measurement when the data threshold is changed.
  - \(<clock> = <BOOLEAN PROGRAM DATA>\)
  - 0 or OFF: Does not restart measurement when the clock delay is changed.
  - 1 or ON: Restarts measurement when the clock delay is changed.

- **Function**: Sets the measurement restart condition.

- **Example**
  - To set the measurement condition so that the measurement is restarted when the Data Threshold or Clock Delay value is changed:
    - `> :SENSe:MEASure:MREStart 1,1`

- **Compatibility**: Compatible with MU181040A/B.

**:SENSe:MEASure:MREStart?**

- **Response**
  - \(<data>,<clock> = <NR1 NUMERIC RESPONSE DATA>\)
  - 0: Measurement restart OFF
  - 1: Measurement restart ON

- **Function**: Queries the measurement restart condition.

- **Example**
  - To check the measurement restart condition:
    - `> :SENSe:MEASure:MREStart?`
    - `< 1,1`

- **Compatibility**: Compatible with MU181040A/B.
7.12.6 Commands Related to Capture Tab

Figure 7.12.6-1  Capture Tab

Figure 7.12.6-2  Condition Setting Dialog Box
### 7.12.28G/32G bit/s ED Commands

**Figure 7.12.6-3  Capture Acquisition Dialog Box**

#### Table 7.12.6-1  Capture Tab Setting Commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1]</td>
<td>Capture Start</td>
<td>(\text{SENSe:CAPTure:STARt})</td>
</tr>
<tr>
<td>[2]</td>
<td>Capture Stop</td>
<td>(\text{SENSe:CAPTure:STOP})</td>
</tr>
<tr>
<td>[3]</td>
<td>Trigger</td>
<td>(\text{SENSe:CAPTure:TRIGger})</td>
</tr>
<tr>
<td>[4]</td>
<td>Number of Block</td>
<td>(\text{SENSe:CAPTure:CONDition:BLOCk})</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(\text{SENSe:CAPTure:CONDition:BLOCk?})</td>
</tr>
<tr>
<td>[5]</td>
<td>Trigger</td>
<td>(\text{SENSe:CAPTure:CONDition:TRIGger})</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(\text{SENSe:CAPTure:CONDition:TRIGger?})</td>
</tr>
<tr>
<td>[6]</td>
<td>Position</td>
<td>(\text{SENSe:CAPTure:CONDition:POSition})</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(\text{SENSe:CAPTure:CONDition:POSition?})</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(\text{SENSe:CAPTure:CONDition:MPLength?})</td>
</tr>
<tr>
<td>[8]</td>
<td>Match Pattern</td>
<td>(\text{SENSe:CAPTure:CONDition:MPEDit})</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(\text{SENSe:CAPTure:CONDition:MPEDit?})</td>
</tr>
<tr>
<td>[9]</td>
<td>Mask Pattern</td>
<td>(\text{SENSe:CAPTure:CONDition:MASKedit})</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(\text{SENSe:CAPTure:CONDition:MASKedit?})</td>
</tr>
<tr>
<td>[10]</td>
<td>Query for capture data block count</td>
<td>(\text{SENSe:CAPTure:ACQuisition:BNUMber?})</td>
</tr>
<tr>
<td>[11]</td>
<td>Start of capture data acquisition</td>
<td>(\text{SENSe:CAPTure:ACQuisition:STARt})</td>
</tr>
<tr>
<td>[12]</td>
<td>Query for capture data acquisition state</td>
<td>(\text{SENSe:CAPTure:ACQuisition:STATe?})</td>
</tr>
<tr>
<td>[13]</td>
<td>Cancel of capture data acquisition</td>
<td>(\text{SENSe:CAPTure:ACQuisition:CANCel})</td>
</tr>
<tr>
<td>[14]</td>
<td>Query for capture data length per block</td>
<td>(\text{SENSe:CAPTure:BPATtern:LENGth})</td>
</tr>
<tr>
<td>[15]</td>
<td>Query for capture result data</td>
<td>(\text{SENSe:CAPTure:BPATtern:DATA:WHOLe})</td>
</tr>
<tr>
<td>[16]</td>
<td>Query for capture result data</td>
<td>(\text{SENSe:CAPTure:BPATtern:BDATa:WHOLe})</td>
</tr>
<tr>
<td>[17]</td>
<td>Query for capture result error position</td>
<td>(\text{SENSe:CAPTure:BPATtern:ERRor:WHOLe})</td>
</tr>
<tr>
<td>[18]</td>
<td>Query for capture result error position</td>
<td>(\text{SENSe:CAPTure:BPATtern:BERRor:WHOLe})</td>
</tr>
</tbody>
</table>
### Chapter 7 SCPI Commands

#### :SENSe:CAPTure:STARt

<table>
<thead>
<tr>
<th>Function</th>
<th>Starts capture result acquisition.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example</td>
<td>&gt; :SENSe:CAPTure:STARt</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Compatible with MU181040A/B.</td>
</tr>
</tbody>
</table>

#### :SENSe:CAPTure:STOP

<table>
<thead>
<tr>
<th>Function</th>
<th>Stops capture result acquisition.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example</td>
<td>&gt; :SENSe:CAPTure:STOP</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Compatible with MU181040A/B.</td>
</tr>
</tbody>
</table>

#### :SENSe:CAPTure:TRIGger

<table>
<thead>
<tr>
<th>Function</th>
<th>Generates a capture manual trigger.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example</td>
<td>&gt; :SENSe:CAPTure:TRIGger</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Compatible with MU181040A/B.</td>
</tr>
</tbody>
</table>

#### :SENSe:CAPTure:CONDition:BLOCk <block>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;block&gt;</td>
<td>&lt;CHARACTER PROGRAM DATA&gt;</td>
</tr>
<tr>
<td>B1</td>
<td>1 Block</td>
</tr>
<tr>
<td>B2</td>
<td>2 Block</td>
</tr>
<tr>
<td>B4</td>
<td>4 Block</td>
</tr>
<tr>
<td>B8</td>
<td>8 Block</td>
</tr>
<tr>
<td>B16</td>
<td>16 Block</td>
</tr>
<tr>
<td>B32</td>
<td>32 Block</td>
</tr>
<tr>
<td>B64</td>
<td>64 Block</td>
</tr>
<tr>
<td>B128</td>
<td>128 Block</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Function</th>
<th>Sets the capture block division number.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example</td>
<td>To set the capture block division number to 8 blocks:</td>
</tr>
<tr>
<td></td>
<td>&gt; :SENSe:CAPTure:CONDition:BLOCk B8</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Compatible with MU181040A/B.</td>
</tr>
</tbody>
</table>
### :SENSe:CAPTure:CONDition:BLOCk?

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;block&gt; = &lt;CHARACTER RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>B1</td>
<td>1 Block</td>
</tr>
<tr>
<td>B2</td>
<td>2 Block</td>
</tr>
<tr>
<td>B4</td>
<td>4 Block</td>
</tr>
<tr>
<td>B8</td>
<td>8 Block</td>
</tr>
<tr>
<td>B16</td>
<td>16 Block</td>
</tr>
<tr>
<td>B32</td>
<td>32 Block</td>
</tr>
<tr>
<td>B64</td>
<td>64 Block</td>
</tr>
<tr>
<td>B128</td>
<td>128 Block</td>
</tr>
</tbody>
</table>

**Function**
Queries the capture block division number.

**Example**
Queries the capture block division number.
> :SENSe:CAPTure:CONDition:BLOCk?
< B8

**Compatibility**
Compatible with MU181040A/B.

### :SENSe:CAPTure:CONDition:TRIGger <trigger>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;trigger&gt; = &lt;CHARACTER PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>EDET</td>
<td>Error Detect: When an error is detected</td>
</tr>
<tr>
<td>PATT</td>
<td>When matches the match pattern</td>
</tr>
<tr>
<td>MAN</td>
<td>When a manual trigger is generated</td>
</tr>
<tr>
<td>EXT</td>
<td>When external trigger is selected</td>
</tr>
</tbody>
</table>

**Function**
Sets the trigger to store the data to memory for the capture function.

**Example**
To set the manual trigger:
> :SENSe:CAPTure:CONDition:TRIGger MANual

**Compatibility**
Compatible with MU181040A/B.

### :SENSe:CAPTure:CONDition:TRIGger?

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;trigger&gt; = &lt;CHARACTER RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>EDET, PATT, MAN, EXT</td>
<td></td>
</tr>
</tbody>
</table>

**Function**
Queries the trigger to store the data to memory for the capture function.

**Example**
> :SENSe:CAPTure:CONDition:TRIGger?
< MAN

**Compatibility**
Compatible with MU181040A/B.
Chapter 7  SCPI Commands

:SENSe:CAPTure:CONDition:POSition <pos>

Parameter  
<pos> = <CHARACTER PROGRAM DATA>
  TOP  Store the data to the top of the memory.
  MIDDle Store the data in the middle of the memory.
  BOTTom Store the data to the last of the memory.

Function  Sets the data storage memory position for the capture function.

Example  To set the data storage memory position to TOP:
  > :SENSe:CAPTure:CONDition:POSition TOP

Compatibility  Compatible with MU181040A/B.

:SENSe:CAPTure:CONDition:POSition?

Response  
<pos> = <CHARACTER RESPONSE DATA>
  TOP, MIDD, BOTT

Function  Queries the data storage memory position for the capture function.

Example  > :SENSe:CAPTure:CONDition:POSition?
  < TOP

Compatibility  Compatible with MU181040A/B

:SENSe:CAPTure:CONDition:MPLength <length>

Parameter  
<length> = <DECIMAL NUMERIC PROGRAM DATA>
  4 to 64  4 to 64 bits, in 4-bit steps

Function  Sets the capture trigger match pattern length.

Example  To set the capture trigger match pattern length to 12 bits:
  > :SENSe:CAPTure:CONDition:MPLength 12

Compatibility  Compatible with MU181040A/B

:SENSe:CAPTure:CONDition:MPLength?

Response  
<length> = <NR1 NUMERIC RESPONSE DATA>
  4 to 64  4 to 64 bits

Function  Queries the capture trigger match pattern length.

Example  > :SENSe:CAPTure:CONDition:MPLength?
  < 12

Compatibility  Compatible with MU181040A/B
### :SENSe:CAPTure:CONDition:MPEDit <data>

**Parameter**

- `<data>` = `<STRING PROGRAM DATA>``
  ```
  "***...***
  Edits the pattern data in hexadecimal format.
  Specify a 1- to 16-character string, consisting of
  0 to 9 and A to F.
  ```

**Function**
Edits the capture trigger match pattern.

**Example**
To set the capture trigger match pattern to 1010:

```
> :SENSe:CAPTure:CONDition:MPEDit "1010"
```

**Compatibility**
Compatible with MU181040A/B

---

### :SENSe:CAPTure:CONDition:MPEDit?

**Response**

- `<data>` = `<STRING RESPONSE DATA>``
  ```
  "***...***
  Returns the pattern data in hexadecimal format.
  Specify a 1- to 16-character string, consisting of
  0 to 9 and A to F.
  ```

**Function**
Queries the capture trigger match pattern.

**Example**

```
> :SENSe:CAPTure:CONDition:MPEDit?
< "1010"
```

**Compatibility**
Compatible with MU181040A/B

---

### :SENSe:CAPTure:CONDition:MASKedit <data>

**Parameter**

- `<data>` = `<STRING PROGRAM DATA>``
  ```
  "***...***
  Edits the pattern data in hexadecimal format.
  Specify a 1- to 16-character string, consisting of
  0 to 9 and A to F.
  ```

**Function**
Edits the capture trigger mask pattern.

**Example**
To set the capture trigger mask pattern to FFFF:

```
> :SENSe:CAPTure:CONDition:MASKedit "FFFF"
```

**Compatibility**
Compatible with MU181040A/B.

---

### :SENSe:CAPTure:CONDition:MASKedit?

**Response**

- `<data>` = `<STRING RESPONSE DATA>``
  ```
  "***...***
  Returns the pattern data in hexadecimal format.
  Specify a 1- to 16-character string, consisting of
  0 to 9 and A to F.
  ```

**Function**
Queries the capture trigger mask pattern.

**Example**

```
> :SENSe:CAPTure:CONDition:MASKedit?
< "FFFF"
```

**Compatibility**
Compatible with MU181040A/B.
Chapter 7  SCPI Commands

:SENSe:CAPTure:ACQuisition:BNUMber?

Parameter  
\(<\text{block}> = <\text{NR1 NUMERIC RESPONSE DATA}>\)  
- 0 to 128 0 to 128 blocks, in 1-block steps, number of valid capture data blocks

Function  
Queries the number of valid capture data blocks.

Example  
> :SENSe:CAPTure:ACQuisition:BNUMber?  
< 128

Compatibility  
Compatible with MU181040A/B.

:SENSe:CAPTure:ACQuisition:STARt <range>[,<start>,<number>]

Parameter  
\(<\text{range}> = <\text{CHARACTER PROGRAM DATA}>\)  
- All Acquires all capture data.  
- SELect Acquires capture data of the specified blocks.  
\([<\text{start}>] = <\text{DECIMAL NUMERIC PROGRAM DATA}>\)  
- 1 to 128 1 to 128 blocks, in 1-block steps, acquisition start block  
\([<\text{number}>] = <\text{DECIMAL NUMERIC PROGRAM DATA}>\)  
- 1 to 128 1 to 128 blocks, in 1-block steps, number of blocks to be acquired

Note:  
The maximum setting value of \(<\text{start}>\) and \(<\text{number}>\) is the maximum value of valid data. \(<\text{start}>\) and \(<\text{number}>\) are omitted when \(<\text{range}> = \text{All}\.\)

Function  
Acquires capture data.

Example  
To acquire all capture data:  
> :SENSe:CAPTure:ACQuisition:STARt ALL  
To acquire capture data of Blocks 1 to 64:  
> :SENSe:CAPTure:ACQuisition:STARt SELect,1,64

Compatibility  
Compatible with MU181040A/B.

:SENSe:CAPTure:ACQuisition:STATe?

Response  
\(<\text{state}> = <\text{NR1 NUMERIC RESPONSE DATA}>\)  
- 0 Capture data acquisition is not performed.  
- 1 Capture data acquisition is being performed

Function  
Queries capture data acquisition status.

Example  
> :SENSe:CAPTure:ACQuisition:STATe?  
< 1

Compatibility  
Compatible with MU181040A/B.
### :SENSe:CAPTure:ACQuisition:CANCel

**Function**
Cancels capture data acquisition.

**Example**
> :SENSe:CAPTure:ACQuisition:CANCel

**Compatibility**
Compatible with MU181040A/B.

### :SENSe:CAPTure:BPATtern:LENGth?

**Response**
\(<\text{length}> = \langle \text{NR1 NUMERIC RESPONSE DATA}\rangle\)
- Maximum capture data / 1 to Maximum capture data / n [bits]
  
\((n = 1, 2, 4, 8, 16, 32, 64, 128)\)
- Capture data length per block

**Function**
Queries the capture data length per block.

**Example**
> :SENSe:CAPTure:BPATtern:LENGth?

< 4194304

**Compatibility**
Compatible with MU181040A/B.

### :SENSe:CAPTure:BPATtern:DATA:WHOLe? <block>,<start>

**Parameter**
- \(<\text{block}> = \langle \text{DECIMAL NUMERIC PROGRAM DATA}\rangle\)
  
1 to 128 1 to 128 blocks, in 1-block steps
- \(<\text{start}> = \langle \text{NON-DECIMAL PROGRAM DATA}\rangle\)
  
#H0 to #H7FFFFFF 0 to 7FFFFFF bits, in 1-bit steps
  
(Specify in hexadecimal)

**Response**
\(<\text{data}> = \langle \text{STRING RESPONSE DATA}\rangle\)

"H***...*"

Returns the pattern data in hexadecimal format.

Within 400 characters (pattern data of 400 \(\times\) 4 bits)

**Function**
Queries the capture data in the specified block, in 400 \(\times\) 4 bits starting from the \(<\text{start}>\) address. Data whose error bit positions are inverted from those in the ED reference data can be obtained.

**Example**
To query the capture data in block 3, in 400 \(\times\) 4 bits from the bit 0 address:
> :SENSe:CAPTure:BPATtern:DATA:WHOLe? 3,#H0

< "H001"

**Compatibility**
Partially compatible with MU181040A/B (the parameter range varies depending on the settings).
Chapter 7  SCPI Commands

:SENSe:CAPTure:BPATtern:BDATa:WHOLe? <block>,<start>[,<size>]

Parameter

- <block> = <DECIMAL NUMERIC PROGRAM DATA>
  1 to 128  1 to 128 Blocks, in 1 Step
- <start> = <NON-DECIMAL PROGRAM DATA>
  #H0 to #H7FFFFFF  0 to 7FFFFFFF bits, in 1-bit steps
  (Specify in hexadecimal)
- [<size>] = <NR1 NUMERIC PROGRAM DATA>
  1 to 1048576  1 to 1,048,576 bytes, in 1 byte Step

Response

- <bdata> = <DEFINITE LENGTH ARBITRARY BLOCK RESPONSE DATA>
  #XYYY<binary>
  X: Number of YYY digits
  YYY: Number of bytes of <binary>
  1 to 1,048,576 bytes
  <binary>: Binary data of up to 1,048,576 bytes

Function

Queries the capture data in the specified block, in 1,048,576 byte binary data starting from the <start> address. Data whose error bit positions are inverted from those in the ED reference data can be obtained.

Example

To query the capture data in block 3, in 1,048,576 byte binary data from the bit 0 address:

> :SENSe:CAPTure:BPATtern:BDATa:WHOLe? 3,#H0
< #10011

Compatibility

Partially compatible with MU181040A/B (the parameter range varies depending on the settings).

:SENSe:CAPTure:BPATtern:ERRor:WHOLe? <block>,<start>

Parameter

- <block> = <DECIMAL NUMERIC PROGRAM DATA>
  1 to 128  1 to 128 Blocks, in 1 Step
- <start> = <NON-DECIMAL PROGRAM DATA>
  #H0 to #H7FFFFFF  0 to 7FFFFFFF bits , in 1 bit steps
  (Specify in hexadecimal)

Response

- <data> = <STRING RESPONSE DATA>
  "H***…*"
  Returns the pattern data in hexadecimal format.
  Within 400 characters (pattern data of 400 × 4 bits)

Function

Queries the capture error positions in the specified block, in 400 × 4 bits starting from the <start> address.

Example

To query the capture error positions in block 3, in 400 × 4 bits from the bit 0 address:

> :SENSe:CAPTure:BPATtern:ERRor:WHOLe? 3,#H0
< "H001"

Compatibility

Partially compatible with MU181040A/B (the parameter range varies depending on the settings).
:SENSe:CAPTure:BPATtern:BERRor:WHOLe? <block>,<start>[,<size>]

Parameter

- **<block>** = <DECIMAL NUMERIC PROGRAM DATA>
  - 1 to 128 blocks, in 1 steps
- **<start>** = <NON-DECIMAL PROGRAM DATA>
  - #H0 to #H7FFFFF bits, in 1 bit steps
  - (Specify in hexadecimal)
- **[<size>]** = <NR1 NUMERIC PROGRAM DATA>
  - 1 to 1048576 bytes, in 1 byte Steps

Response

- **<bdata>** = <DEFINITE LENGTH ARBITRARY BLOCK RESPONSE DATA>
  - ##XYYY<binary>
    - X: Number of YYY digits
    - YYY: Number of bytes of <binary>
  - 1 to 1,048,576 bytes
  - <binary>: Binary data of up to 1,048,576 bytes

Function

Queries the capture error positions in the specified block, in 1,048,576 bytes binary data starting from the <start> address.

Example

To query the capture error positions in block 3, in 1,048,576 bytes binary data from the bit 0 address:

> :SENSe:CAPTure:BPATtern:BERRor:WHOLe? 3,#H0
< #10011

Compatibility

Partially compatible with MU181040A/B. (the parameter range varies depending on the settings).
7.12.7 Commands Related to Misc1 Tab

Figure 7.12.7-1  Misc1 Tab
7.12.7.1 Pattern Sequence setting commands

Table 7.12.7.1-1 Pattern Sequence Setting Commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1]</td>
<td>Sequence</td>
<td>:SENSe:PATTern:IMODe</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:PATTern:IMODe?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:PATTern:BURSt:MODE?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:PATTern:BURSt:DELay?</td>
</tr>
<tr>
<td></td>
<td>Manual/Auto</td>
<td>:SENSe:PATTern:BURSt:ADJust?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:PATTern:BURSt:ELENgth?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:PATTern:BURSt:CYCLE?</td>
</tr>
</tbody>
</table>

:SENSe:PATTern:IMODe <mode>

Parameter

<mode> = <CHARACTER PROGRAM DATA>
- REPeat: Repeat signal
- BURSt: Burst signal

Function
Sets the signal reception method from Repeat (consecutive) signal or Burst signal.

Example
To set the signal reception method to Repeat signal:

> :SENSe:PATTern:IMODe REPeat

Compatibility
Compatible with MU181040A/B.
Chapter 7 SCPI Commands

:SENSe:PATTern:IMODe?
Response  <mode> = <CHARACTER RESPONSE DATA>
    REP, BURS
Function  Queries the signal reception method.
Example  > :SENSe:PATTern:IMODe?
    < REP
Compatibility  Compatible with MU181040A/B.

:SENSe:PATTern:BURSt:MODE <mode>
Parameter  <mode> = <CHARACTER PROGRAM DATA>
    INTernal  Internal
    EXITTrig  External Trigger
    EXTenable  External Enable
Function  Sets the Burst signal reception sequence.
Example  To set the Burst signal reception sequence to Internal:
    > :SENSe:PATTern:BURSt:MODE  INTernal
Compatibility  Compatible with MU181040A/B.

:SENSe:PATTern:BURSt:MODE?
Response  <mode> = <CHARACTER RESPONSE DATA>
    INT, EXTT, EXT
Function  Queries the Burst signal reception sequence.
Example  > :SENSe:PATTern:BURSt:MODE?
    < INT
Compatibility  Compatible with MU181040A/B.

:SENSe:PATTern:BURSt:DELay <numeric>
Parameter  <numeric> = <DECIMAL NUMERIC PROGRAM DATA>
    0 to 2147483640  0 to 2,147,483,640 bits/8 bits Step
    Note:
    At 2ch Combination, the setting range and Step are doubled.
    At 4ch Combination, the setting range and Step are quadrupled.
Function  Sets the delay value for the Burst signal.
Example  To set the delay value for the Burst signal to 0 bits:
    > :SENSe:PATTern:BURSt:DELay  0
Compatibility  Partially compatible with MU181040A/B.
:SENSe:PATTern:BURSt:DELay?
Response  
<numeric> = <NR1 NUMERIC RESPONSE DATA>
Function  Queries the delay value for the Burst signal.
Example  
> :SENSe:PATTern:BURSt:DELay?
< 0
Compatibility  Partially compatible with MU181040A/B.

:SENSe:PATTern:BURSt:ADJust <adjust >
Parameter  
<adjust> = <CHARACTER PROGRAM DATA>
AUTO  Executes automatic adjustment.
MANual  Does not execute automatic adjustment
Function  Sets whether to execute automatic adjustment of Burst Trigger Delay.
Example  
> :SENSe:PATTern:BURSt:ADJust AUTO
Compatibility  Compatible with MU181040A/B.

:SENSe:PATTern:BURSt:ADJust?
Response  
<adjust> = <CHARACTER RESPONSE DATA>
Function  Queries the Burst Trigger Delay adjustment method.
Example  
> :SENSe:PATTern:BURSt:ADJust?
< AUTO
Compatibility  Compatible with MU181040A/B.

:SENSe:PATTern:BURSt:ELENgth <numeric>
Parameter  
<numeric> = <DECIMAL NUMERIC PROGRAM DATA>
12800 to 2147483392  12,800 to 2,147,483,392 bits/128 bits Step
Note:  
At 2ch Combination, the setting range and Step are doubled.
At 4ch Combination, the setting range and Step are quadrupled.
Function  Sets the data signal reception interval for Burst signal reception.
Example  
To set the Burst signal reception interval to 12,800 bits:
> :SENSe:PATTern:BURSt:ELENgth 12800
Compatibility  Partially compatible with MU181040A/B.
Chapter 7  SCPI Commands

:SENSe:PATTern:BURSt:ELENgth?

| Response | <numeric> = <NR1 NUMERIC RESPONSE DATA> |
| Function | Queries the data signal reception interval for Burst signal reception. |
| Example  | > :SENSe:PATTern:BURSt:ELENgth?
          | < 12800 |
| Compatibility | Partially compatible with MU181040A/B. |

:SENSe:PATTern:BURSt:CYCLe <numeric>

| Parameter | <numeric> = <DECIMAL NUMERIC PROGRAM DATA> |
|          | 25600 to 2147483648  25,600 to 2,147,483,648 bits/128 bits Step |
| Note:     | At 2ch Combination, the setting range and Step are doubled. |
|           | At 4ch Combination, the setting range and Step are quadrupled. |
| Function  | Sets the Burst signal generation cycle. |
| Example   | To set the Burst signal generation cycle to 25,600 bits: |
|           | > :SENSe:PATTern:BURSt:CYCLe 25600 |
| Compatibility | Partially compatible with MU181040A/B. |

:SENSe:PATTern:BURSt:CYCLe?

| Response | <numeric> = <NR1 NUMERIC RESPONSE DATA> |
| Function | Queries the Burst signal generation cycle. |
| Example  | > :SENSe:PATTern:BURSt:CYCLe?  |
|          | < 25600 |
| Compatibility | Partially compatible with MU181040A/B. |
7.12.7.2 AUX Input setting commands

![AUX Input Setting](image)

Figure 7.12.7.2-1 AUX Input Setting

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>:INPut:AUX:USAGe?</td>
</tr>
</tbody>
</table>

:INPut:AUX:USAGe <usage>

Parameter

<usage> = <CHARACTER PROGRAM DATA>
- BURSt: Burst Source
- MASK: External Mask
- CAPTure: Capture External Trigger

Function
Sets the usage of the common connector input.

Example
To set the usage of the common connector input to Burst Source:

> :INPut:AUX:USAGe BURSt

Compatibility
Compatible with MU181040A/B.

:INPut:AUX:USAGe?

Response
<usage> = <CHARACTER RESPONSE DATA>
- BURS, MASK, CAPT

Function
Queries the usage of the common connector input.

Example
To query the usage of the common connector input:

> :INPut:AUX:USAGe?

< BURS

Compatibility
Compatible with MU181040A/B.
Chapter 7  SCPI Commands

7.12.7.3 AUX Output setting commands

Table 7.12.7.3-1  AUX Output Setting Commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>:OUTPut:SYNC:SOURce?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:OUTPut:SYNC:POsition?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:OUTPut:SYNC:MIXData:BLOCk?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:OUTPut:SYNC:MIXData:ROW?</td>
</tr>
</tbody>
</table>
### :OUTPut:SYNC:SOURce <source>[,<numeric>]

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;source&gt;</td>
<td>CHARACTER PROGRAM DATA &gt; CLO CK8 1/8 clock CLO CK16 1/16 clock CLO CK32 1/32 clock CLO CK64 1/64 clock PATT ern Pattern Sync (Variable) SGA in Sync Gain ER Rorut Error Output NCLock 1/n clock</td>
</tr>
<tr>
<td>[&lt;numeric&gt;]</td>
<td>DECIMAL NUMERIC PROGRAM DATA &gt; 4 to 512 4,6,8,10,…,510,512/2 Step</td>
</tr>
</tbody>
</table>

#### Function
Sets the output signal for synchronization output.

#### Example
To set the output signal for synchronization output to 1/8 clocks:
```
> :OUTPut:SYNC:SOURce CLOck8
```
To set the output signal for synchronization output to 1/511 clocks:
```
> :OUTPut:SYNC:SOURce NCLock,510
```

#### Compatibility
Partially compatible with MU181040A/B.

### :OUTPut:SYNC:SOURce?

<table>
<thead>
<tr>
<th>Response</th>
<th>CHARACTER RESPONSE DATA CLOC8, CLOC16, CLOC32, CLOC64, PATT, SGA, ERR, NCL</th>
</tr>
</thead>
<tbody>
<tr>
<td>[&lt;numeric&gt;]</td>
<td>DECIMAL NUMERIC RESPONSE DATA Omits &lt;numeric&gt; for other than NCL (1/n clocks).</td>
</tr>
</tbody>
</table>

#### Function
Queries the output signal for synchronization output.

#### Example
```
> :OUTPut:SYNC:SOURce?
< CLOC4
> :OUTPut:SYNC:SOURce?
< NCL,510
```

#### Compatibility
Partially compatible with MU181040A/B.
**Chapter 7  SCPI Commands**

**:OUTPut:SYNC:POSition <numeric>**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;numeric&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1 to 34359738105 1 to 34,359,738,105 bits/8bits Step</td>
</tr>
</tbody>
</table>

*Note:*

At 2ch Combination, the setting range varies and Step is doubled.

1 to 68719476209 1 to 68,719,476,209 bits/16 bits Step

At 4ch Combination, the setting range varies and Step is quadrupled.

1 to 137438952417 1 to 1,374,389,524,171 bits/32 bits Step

**Function**

Sets the output position for synchronization output for Data, PRBS, and Zero Substitution patterns.

**Example**

To set the synchronization output position to bit 1:

```
> :OUTPut:SYNC:POSition 1
```

**Compatibility**

Partially compatible with MU181040A/B.

**:OUTPut:SYNC:POSition?**

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;numeric&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function</td>
<td>Queries the output position for synchronization output for Data, PRBS, and Zero Substitution patterns.</td>
</tr>
<tr>
<td>Example</td>
<td>&gt; :OUTPut:SYNC:POSition?</td>
</tr>
<tr>
<td></td>
<td>&lt; 1</td>
</tr>
</tbody>
</table>

**Compatibility**

Partially compatible with MU181040A/B (the parameter range varies depending on the settings).

**:OUTPut:SYNC:MIXData:BLOCk <numeric>**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;numeric&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1 to 511 1 to 511 Block/1 Block Step</td>
</tr>
</tbody>
</table>

The maximum setting value is the value set in Block No

**Function**

Sets the output block for synchronization output for Mixed Data pattern

**Example**

To set the output block for synchronization output to block 1:

```
> :OUTPut:SYNC:MIXData:BLOCk 1
```

**Compatibility**

Compatible with MU181040A/B.
:OUTPut:SYNC:MIXData:BLOCk?

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;numeric&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function</td>
<td>Queries the output block for synchronization output for Mixed Data pattern.</td>
</tr>
<tr>
<td>Example</td>
<td>&gt; :OUTPut:SYNC:MIXData:BLOCk?</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Compatible with MU181040A/B.</td>
</tr>
</tbody>
</table>

:OUTPut:SYNC:MIXData:ROW <numeric>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;numeric&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 to 16</td>
<td>1 to 16 Row/1 Row Step</td>
</tr>
<tr>
<td></td>
<td>The maximum setting value is the value set in Row No.</td>
</tr>
<tr>
<td>Function</td>
<td>Sets the output Row position for synchronization output for Mixed Data pattern.</td>
</tr>
<tr>
<td>Example</td>
<td>To set the synchronization output Row to 1 Row.</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Compatible with MU181040A/B.</td>
</tr>
</tbody>
</table>

:OUTPut:SYNC:MIXData:ROW?

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;numeric&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 to 16</td>
<td>1 to 16 Row</td>
</tr>
<tr>
<td>Function</td>
<td>Queries the output Row position for synchronization output for Mixed Data pattern.</td>
</tr>
<tr>
<td>Example</td>
<td>&gt; :OUTPut:SYNC:MIXData:ROW?</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Compatible with MU181040A/B.</td>
</tr>
</tbody>
</table>
Chapter 7  SCPI Commands

7.12.8 Commands Related to Misc2 Tab

7.12.8.1 Combination setting commands

Table 7.12.8.1-1 Combination Setting Commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Item</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1]</td>
<td>Combination Setting</td>
<td>:MCOMBination:OPERation:SETTING &lt;configuration&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:MCOMBination:OPERation:SETTING?</td>
</tr>
</tbody>
</table>

**:MCOMBination:OPERation:SETTING <configuration>**

Parameter  
<configuration> = <DECIMAL NUMERIC PROGRAM DATA>

0  Independent
2  2ch Combination
4  4ch Combination

Function  
Sets the combination setting for 32G ED Module, from Combination or Independent.

Example  
To set the combination setting to 4ch Combination.
> :MCOMBination:OPERation:SETTING 4

Compatibility  
Incompatible with existing models

:**MCOMBination:OPERation:SETTING?**

Response  
<numeric> = <NR1 NUMERIC RESPONSE DATA>

0  Independent
2  2ch Combination
4  4ch Combination

Function  
Queries the combination setting of 32G ED Module.

Example  
> :MCOMBination:OPERation:SETTING?
< 4

Compatibility  
Incompatible with existing models
7.12.8.2 Grouping setting commands

**Figure 7.12.8.2-1** Grouping Setting Dialog Box for Misc 2 Tab

**Table 7.12.8.2-1** Combination Setting Commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1]</td>
<td>Selects tabs to be grouped together.</td>
<td>:SENSe:GROup:SETTing</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:GROup:SETTing?</td>
</tr>
<tr>
<td>[2]</td>
<td>Selects Data IFs to be grouped together.</td>
<td>:SENSe:GROup:SELECT</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:GROup:SELECT?</td>
</tr>
</tbody>
</table>

:SENSe:GROup:SETTing <tab>,<interface>

**Parameter**
- `<tab>` = `<CHARACTER PROGRAM DATA>`
- PATTern Pattern tab
- INPut Input tab
- `<interface>` = `<NR1 NUMERIC PROGRAM DATA>`
- 0 to 15
- Available bits
  - 1 (Bit0) Data1
  - 2 (Bit1) Data2
  - 4 (Bit2) Data3
  - 8 (Bit3) Data4

**Function**
Sets the Grouping function.

**Example**
To group the Pattern tabs of Data1 and Data 2 together:
> :SENSe:GROup:SETTing PATTern,3

To group the Input tabs of Data1 to Data 4 together:
> :SENSe:GROup:SETTing INPut,15

**Compatibility**
Incompatible with existing models.
Chapter 7  SCPI Commands

:SENSe:GROup:SETTing? <tab>

Parameter  
				<tab> = <CHARACTER PROGRAM DATA>
			PATTern  Pattern tab
			INPut  Input tab

Response  
			<interface> = <NR1 NUMERIC RESPONSE DATA>
			Available bits
			1 (Bit0)  Data1
			2 (Bit1)  Data2
			4 (Bit2)  Data3
			8 (Bit3)  Data4

Function  
Queries the Grouping setting.

Example  
To query the Data Interfaces whose Input tabs are grouped together:

> :SENSe:GROup:SETTing? INPut
< 12 (Data3 and Data4 are grouped together.)

Compatibility  
Incompatible with existing models.

:SENSe:GROup:SELeCT <group>

Parameter  
				<group> = <DECIMAL NUMERIC PROGRAM DATA>
			1  One group (Data1-2 or Data1-4)
			2  Two groups (Data1-2 and Data3-4)

Function  
Sets the number of groups.

Example  
To set the number of groups to 2:

> :SENSe:GROup:SELeCT 2

Compatibility  
Incompatible with existing models.

:SENSe:GROup:SELeCT?

Response  
			<group> = <NR1 NUMERIC RESPONSE DATA>
			1  One group (Data1-2 or Data1-4)
			2  Two groups (Data1-2 and Data3-4)

Function  
Queries the number of groups.

Example  
> :SENSe:GROup:SELeCT?
< 2

Compatibility  
Incompatible with existing models.
7.12.9 Eye Margin Measurement

This section describes the commands for Eye Margin measurement.

7.12.9.1 Eye Margin measurement setting commands

Figure 7.12.9.1-1  Eye Margin Window

Figure 7.12.9.1-2  Eye Margin Window (PAM mode)
Table 7.12.9.1-1  Eye Margin measurement Setting Commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Set All</td>
<td>$SENSe:MEASure:EMARgin:MARGin:ASLot</td>
</tr>
<tr>
<td>2</td>
<td>Reset All</td>
<td>$SENSe:MEASure:EMARgin:MARGin:RSLot</td>
</tr>
<tr>
<td>3</td>
<td>Slot</td>
<td>$SENSe:MEASure:EMARgin:MARGin:SLOT</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$SENSe:MEASure:EMARgin:MARGin:SLOT?</td>
</tr>
<tr>
<td>4</td>
<td>Start</td>
<td>$SENSe:MEASure:STARt</td>
</tr>
<tr>
<td>5</td>
<td>Stop</td>
<td>$SENSe:MEASure:STOP</td>
</tr>
<tr>
<td>6</td>
<td>No label (Query for measurement state)</td>
<td>$SENSe:MEASure:EMARgin:STATe?</td>
</tr>
<tr>
<td>7</td>
<td>Error Threshold</td>
<td>$SENSe:MEASure:EMARgin:MARGin:THReshold</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$SENSe:MEASure:EMARgin:MARGin:THReshold?</td>
</tr>
<tr>
<td>8</td>
<td>Fine/Coarse</td>
<td>$SENSe:MEASure:EMARgin:MARGin:RESolution</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$SENSe:MEASure:EMARgin:MARGin:RESolution?</td>
</tr>
<tr>
<td>9</td>
<td>Auto Search</td>
<td>$SENSe:MEASure:EMARgin:MARGin:ASEarch</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$SENSe:MEASure:EMARgin:MARGin:ASEarch?</td>
</tr>
<tr>
<td>10</td>
<td>Selection of measurement time display</td>
<td>$SENSe:MEASure:EMARgin:TIME</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$SENSe:MEASure:EMARgin:TIME?</td>
</tr>
<tr>
<td>11</td>
<td>Query for date and time</td>
<td>$SENSe:MEASure:EMARgin:DTIME?</td>
</tr>
<tr>
<td>12</td>
<td>Query for measurement start time</td>
<td>$SENSe:MEASure:EMARgin:STARt?</td>
</tr>
<tr>
<td>13</td>
<td>Query for measurement elapsed time</td>
<td>$SENSe:MEASure:EMARgin:ELAPsed?</td>
</tr>
<tr>
<td>14</td>
<td>Query for measurement result</td>
<td>$CALCulate:DATA:EMARgin?</td>
</tr>
<tr>
<td>15</td>
<td>No label (Query for measurement status)</td>
<td>$CALCulate:DATA:EMARgin:STATUs?</td>
</tr>
<tr>
<td>16</td>
<td>Selection of search target threshold (In PAM mode only)</td>
<td>$SENSe:MEASure:EMARgin:MARGin:ASEarch:PAMSet</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$SENSe:MEASure:EMARgin:MARGin:ASEarch:PAMSet?</td>
</tr>
</tbody>
</table>

\$SENSe:MEASure:EMARgin:MARGin:ASLot

Function          Selects all Eye Margin measurement slots to be set to ON.
Example           > \$SENSe:MEASure:EMARgin:MARGin:ASLot
Compatibility     Incompatible with existing models

\$SENSe:MEASure:EMARgin:MARGin:RSLot

Function          Selects all Eye Margin measurement slots to be set to OFF.
Example           > \$SENSe:MEASure:EMARgin:MARGin:RSLot
Compatibility     Incompatible with existing models
:SENSe:MEASure:EMARgin:MARGin:SLOT
[slot],<interface>[,<unit>]

Parameter

- `<slot>` = <CHARACTER PROGRAM DATA>
  - SLOT1 to SLOT6 Slots No. 1 to 6
  - 1 to 6 when using the MP1800A, 1 to 4 when using the MT1810A
- `<interface>` = <DECIMAL NUMERIC PROGRAM DATA>
  - 1 to 4 Data1 to 4
- `<boolean>` = <BOOLEAN PROGRAM DATA>
  - ON or 1 ON
  - OFF or 0 OFF
- `[<unit>]` = <DECIMAL NUMERIC PROGRAM DATA>
  - 1 to 4 Mainframe No. 1 to 4
  - When using two or more MT1810A units in serial connection, specify the mainframe number.
  - Can be omitted. Mainframe No. 1 is specified when omitted.

Function
- Selects the Eye Margin measurement slot.

Example
- To select Eye Margin measurement Slot 1, Data 3 to be ON:
  > :SENSe:MEASure:EMARgin:MARGin:SLOT SLOT1,3,ON

Compatibility
- Incompatible with existing models

:SENSe:MEASure:EMARgin:MARGin:SLOT? <slot>,<interface>[,<unit>]

Parameter

- `<slot>` = <CHARACTER PROGRAM DATA>
  - SLOT1 to SLOT6 Slots No. 1 to 6
  - 1 to 6 when using the MP1800A, 1 to 4 when using the MT1810A
- `<interface>` = <DECIMAL NUMERIC PROGRAM DATA>
  - 1 to 4 Data1 to 4
- `[<unit>]` = <DECIMAL NUMERIC PROGRAM DATA>
  - 1 to 4 Mainframe No. 1 to 4
  - When using two or more MT1810A units in serial connection, specify the mainframe number.
  - Can be omitted. Mainframe No. 1 is specified when omitted.

Response
- `<boolean>` = <NR1 NUMERIC RESPONSE DATA>
  - 1 ON
  - 0 OFF

Function
- Queries whether the specified slot is targeted for the Eye Margin measurement.

Example
- To query whether the slot 1, Data 3 is targeted for the Eye Margin measurement:
  > :SENSe:MEASure:EMARgin:MARGin:SLOT? SLOT1,3
  < 1

Compatibility
- Incompatible with existing models
Chapter 7  SCPI Commands

**:SENSe:MEASure:STARt**

**Function**
Starts the Eye Margin measurement.

**Example**
> :SENSe:MEASure:STARt

**Compatibility**
Incompatible with existing models

**:SENSe:MEASure:STOP**

**Function**
Stops the Eye Margin measurement.

**Example**
> :SENSe:MEASure:STOP

**Compatibility**
Incompatible with existing models

**:SENSe:MEASure:EMARgin:STATe?**

**Response**
<boolean> = <NR1 NUMERIC RESPONSE DATA>
1  Starts measurement
0  Stops measurement
–1  Measurement failure

**Function**
Queries the Eye Margin measurement state.

**Example**
> :SENSe:MEASure:EMARgin:STATe?
< 1

**Compatibility**
Compatible with the MP1632C.

**:SENSe:MEASure:EMARgin:MARGin:THReshold <thre>**

**Parameter**
<thre> = <CHARACTER PROGRAM DATA>
E_3 to E_12  1E–3 to 1E–12, in single steps

**Function**
Sets the threshold value that is to be the boundary of the Eye Margin measurement margin.

**Example**
To set the threshold value that is to be the Eye Margin measurement margin to 1E–4:
> :SENSe:MEASure:EMARgin:MARGin:THReshold E_4

**Compatibility**
Compatible with the MP1632C.

**:SENSe:MEASure:EMARgin:MARGin:THReshold?**

**Response**
<thre> = <CHARACTER RESPONSE DATA>
E_3 to E_12  1E–3 to 1E–12

**Function**
Queries the threshold value that is to be the boundary of the Eye Margin measurement margin.

**Example**
> :SENSe:MEASure:EMARgin:MARGin:THReshold?
< E_4

**Compatibility**
Compatible with the MP1632C.
### :SENSe:MEASure:EMARgin:MARGin:RESolution <type>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;type&gt; = &lt;CHARACTER PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>FINE</td>
</tr>
<tr>
<td></td>
<td>FINE mode</td>
</tr>
<tr>
<td></td>
<td>COARse</td>
</tr>
<tr>
<td></td>
<td>COARse mode</td>
</tr>
</tbody>
</table>

**Function**
Sets the accuracy of the error ratio for the Eye Margin measurement.

**Example**
To set the accuracy of the error ratio for the Eye Margin measurement to the Fine mode:
> :SENSe:MEASure:EMARgin:MARGin:RESolution FINE

**Compatibility**
Compatible with the MP1632C.

---

### :SENSe:MEASure:EMARgin:MARGin:RESolution? 

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;type&gt; = &lt;CHARACTER RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>FINE, COAR</td>
</tr>
</tbody>
</table>

**Function**
Queries the accuracy of the error ratio for the Eye Margin measurement.

**Example**
> :SENSe:MEASure:EMARgin:MARGin:RESolution?
< FINE

**Compatibility**
Compatible with the MP1632C.

---

### :SENSe:MEASure:EMARgin:MARGin:ASEarch <boolean>[,<mode>]

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;boolean&gt; = &lt;BOOLEAN PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ON or 1</td>
</tr>
<tr>
<td></td>
<td>Auto Search ON</td>
</tr>
<tr>
<td></td>
<td>OFF or 0</td>
</tr>
<tr>
<td></td>
<td>Auto Search OFF</td>
</tr>
<tr>
<td></td>
<td>&lt;mode&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
</tr>
<tr>
<td></td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Coarse</td>
</tr>
<tr>
<td></td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>Fine</td>
</tr>
<tr>
<td></td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>PAM Coarse</td>
</tr>
<tr>
<td></td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>PAM Fine</td>
</tr>
</tbody>
</table>

**Note:**

<mode> can be omitted. Coarse is specified when omitted.

**Function**
Selects whether to perform Auto Search upon start of the Eye Margin measurement.

**Example**
To set Auto Search upon start of the Eye Margin measurement to ON:
> :SENSe:MEASure:EMARgin:MARGin:ASEarch ON,1

**Compatibility**
Incompatible with existing models.
Chapter 7  SCPI Commands

:SENSe:MEASure:EMARgin:MARGin:ASEarch?

Response  

| <boolean> = <NR1 NUMERIC RESPONSE DATA> |
|-----------|--------------------------------------------------|
| 0         | OFF                                               |
| 1         | ON (Coarse)                                       |
| 2         | ON (Fine)                                          |
| 3         | ON (PAM Coarse)                                   |
| 4         | ON (PAM Fine)                                     |

Function  
Queries whether to perform Auto Search upon start of the Eye Margin measurement.

Example  

> :SENSe:MEASure:EMARgin:MARGin:ASEarch?  
< 1

Compatibility  
Incompatible with existing models

:SENSe:MEASure:EMARgin:TIME <type>

Parameter  

| <type> = <CHARACTER PROGRAM DATA> |
|-----------|--------------------------------------------------|
| DTIME     | Displays the current date and time.               |
| STARt     | Displays the measurement start time.              |
| ELAPsed   | Displays the elapsed time based on the measurement period. |

Function  
Selects the Eye Margin measurement time display type.

Example  

To set the Eye Margin measurement time display type to measurement start time (Start Time):

> :SENSe:MEASure:EMARgin:TIME STARTt

Compatibility  
Compatible with the MP1632C.

:SENSe:MEASure:EMARgin:TIME?

Response  

| <type> = <CHARACTER RESPONSE DATA> |
|-----------|--------------------------------------------------|
| DTIME     | Displays the current date and time.               |
| STAR     | Displays the measurement start time.              |
| ELAP     | Displays the elapsed time based on the measurement period. |

Function  
Queries the Eye Margin measurement time display type.

Example  

> :SENSe:MEASure:EMARgin:TIME?  
< STAR

Compatibility  
Compatible with the MP1632C.
:SENSe:MEASure:EMARgin:DTIMe?

Response

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;year&gt;</td>
<td>&lt;NR1 NUMERIC RESPONSE DATA&gt;</td>
</tr>
<tr>
<td>2000 to 2036</td>
<td>Year 2000 to 2036</td>
</tr>
<tr>
<td>&lt;month&gt;</td>
<td>&lt;NR1 NUMERIC RESPONSE DATA&gt;</td>
</tr>
<tr>
<td>1 to 12</td>
<td>January to December</td>
</tr>
<tr>
<td>&lt;day&gt;</td>
<td>&lt;NR1 NUMERIC RESPONSE DATA&gt;</td>
</tr>
<tr>
<td>1 to 31</td>
<td>1st to 31st</td>
</tr>
<tr>
<td>&lt;hour&gt;</td>
<td>&lt;NR1 NUMERIC RESPONSE DATA&gt;</td>
</tr>
<tr>
<td>0 to 23</td>
<td>0 to 23 hours</td>
</tr>
<tr>
<td>&lt;minute&gt;</td>
<td>&lt;NR1 NUMERIC RESPONSE DATA&gt;</td>
</tr>
<tr>
<td>0 to 59</td>
<td>0 to 59 minutes</td>
</tr>
<tr>
<td>&lt;second&gt;</td>
<td>&lt;NR1 NUMERIC RESPONSE DATA&gt;</td>
</tr>
<tr>
<td>0 to 59</td>
<td>0 to 59 seconds</td>
</tr>
</tbody>
</table>

Function
Queries the current date and time during Eye Margin measurement.

Example
> :SENSe:MEASure:EMARgin:DTIMe?
< 2012,4,1,23,59,59

Compatibility
Incompatible with existing models

:SENSe:MEASure:EMARgin:STARt?

Response

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;year&gt;</td>
<td>&lt;NR1 NUMERIC RESPONSE DATA&gt;</td>
</tr>
<tr>
<td>0, 2000 to 2036</td>
<td>Year 2000 to 2036</td>
</tr>
<tr>
<td>&lt;month&gt;</td>
<td>&lt;NR1 NUMERIC RESPONSE DATA&gt;</td>
</tr>
<tr>
<td>0 to 12</td>
<td>0 to December</td>
</tr>
<tr>
<td>&lt;day&gt;</td>
<td>&lt;NR1 NUMERIC RESPONSE DATA&gt;</td>
</tr>
<tr>
<td>0 to 31</td>
<td>0 to 31st</td>
</tr>
<tr>
<td>&lt;hour&gt;</td>
<td>&lt;NR1 NUMERIC RESPONSE DATA&gt;</td>
</tr>
<tr>
<td>0 to 23</td>
<td>0 to 23 hours</td>
</tr>
<tr>
<td>&lt;minute&gt;</td>
<td>&lt;NR1 NUMERIC RESPONSE DATA&gt;</td>
</tr>
<tr>
<td>0 to 59</td>
<td>0 to 59 minutes</td>
</tr>
<tr>
<td>&lt;second&gt;</td>
<td>&lt;NR1 NUMERIC RESPONSE DATA&gt;</td>
</tr>
<tr>
<td>0 to 59</td>
<td>0 to 59 seconds</td>
</tr>
</tbody>
</table>

Note:
The following is output when no measurement is performed:

0,0,0,0,0,0

Function
Queries the Eye Margin measurement start time.

Example
> :SENSe:MEASure:EMARgin:STARt?
< 2012,4,1,23,59,59

Compatibility
Compatible with the MP1632C.
Chapter 7  SCPI Commands

:SENSe:MEASure:EMARgin:ELAPsed?

Response  
<day> = <NR1 NUMERIC RESPONSE DATA>
0 to 99  0 to 99 days
<hour> = <NR1 NUMERIC RESPONSE DATA>
0 to 23  0 to 23 hours
<minute> = <NR1 NUMERIC RESPONSE DATA>
0 to 59  0 to 59 minutes
<second> = <NR1 NUMERIC RESPONSE DATA>
0 to 59  0 to 59 seconds

Note:
The following is output when no measurement is performed:
0,0,0,0

Function
Queries the Eye Margin measurement elapsed time.

Example
> :SENSe:MEASure:EMARgin:ELAPsed?
< 99,23,59,59

Compatibility
Compatible with the MP1632C.

:CALCulate:DATA:EMARgin? <slot>,<interface>,<string>[,<unit>]

Parameter
<slot> = <CHARACTER PROGRAM DATA>
SLOT1 to SLOT6  Slots No. 1 to 6
<interface> = <DECIMAL NUMERIC PROGRAM DATA>
1 to 4  Data1 to 4
<string> = <STRING PROGRAM DATA>
See Table 7.12.9.1-2 for the <string> contents.
[unit] = <DECIMAL NUMERIC PROGRAM DATA>
1 to 4  Mainframe No. 1 to 4
When using two or more MT1810A units in serial connection, specify the mainframe number.
Can be omitted. Mainframe No. 1 is specified when omitted.

Table 7.12.9.1-2  Eye Margin Measurement Items <string>

<table>
<thead>
<tr>
<th>Items</th>
<th>&lt;string&gt;</th>
<th>Response Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>Phase margin</td>
<td>&quot;PHASe&quot;</td>
<td>Form1</td>
</tr>
<tr>
<td>Threshold margin</td>
<td>&quot;THReshold&quot;</td>
<td>Form2</td>
</tr>
<tr>
<td>Period</td>
<td>&quot;PERiod&quot;</td>
<td>Form1</td>
</tr>
</tbody>
</table>

Response  
<string> = <STRING RESPONSE DATA>
For the contents of the <result> measurement data, see Table 7.12.9.1-3.
Table 7.12.9.1-3  Eye Margin Measurement Results

<table>
<thead>
<tr>
<th>Form</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Form1</td>
<td>&quot; XXXX&quot;</td>
<td>For 0 to MAX (mUI/ps) Displays in the selected unit.</td>
</tr>
<tr>
<td>Phase type</td>
<td>&quot;Failed&quot;</td>
<td>For measurement failure</td>
</tr>
<tr>
<td></td>
<td>&quot;......&quot;</td>
<td>When no data corresponds to the query.</td>
</tr>
<tr>
<td>Form2</td>
<td>&quot; XXXX&quot;</td>
<td>0 to MAX (mVpp)</td>
</tr>
<tr>
<td>Threshold type</td>
<td>&quot;Failed&quot;</td>
<td>For measurement failure</td>
</tr>
<tr>
<td></td>
<td>&quot;......&quot;</td>
<td>When no data corresponds to the query.</td>
</tr>
</tbody>
</table>

Function
Queries the Eye Margin measurement result of Slot 1, Data 2.
Example
> :CALCulate:DATA:EMARgin? SLOT1,2, "THReshold"
< "     0"
Compatibility
Incompatible with existing models

:CALCulate:DATA:EMARgin:STATus?

Response
<string> = <STRING RESPONSE DATA>
"" When no alarm exists
"Sync Loss" Sync. Loss
"Clock Loss" Clock Loss
"Out of range" Out of range
" Illegal Error" Illegal Error
"Meas. Threshold exceeded" Number of errors exceeds the target rate

Note:
When two or more alarms have occurred, they are displayed in the following order:
Illegal Error > Clock Loss > Sync Loss > Out of range

Function
Queries the Eye Margin measurement status.
Example
> :CALCulate:DATA:EMARgin:STATus?
< "Sync Loss"
Compatibility
Incompatible with existing models
### :SENSe:MEASure:EMARgin:MARGin:ASEarch:PAMSet

**Parameter**

- `<slot>` = `<CHARACTER PROGRAM DATA>`
  - SLOT1 to SLOT6 Slots No. 1 to 6
  - 1 to 6 when using the MP1800A, 1 to 4 when using the MT1810A
- `<interface>` = `<DECIMAL NUMERIC PROGRAM DATA>`
  - 1 to 4 Data1 to 4
- `<threshold>` = `<CHARACTER PROGRAM DATA>`
  - TOP Searches the Top threshold
  - MIDDle Searches the Middle threshold
  - BOTTom Searches the Bottom threshold
- `[<unit>]` = `<DECIMAL NUMERIC PROGRAM DATA>`
  - 1 to 4 Mainframe No. 1 to 4

When using two or more MT1810A units in serial connection, specify the mainframe number. Can be omitted. Mainframe No. 1 is specified when omitted.

**Function**

Selects search target threshold for running PAM Auto Search in the Eye Margin measurement.

**Example**

To set Data 3 of Slot No.1 for Top threshold search.

```
> :SENSe:MEASure:EMARgin:MARGin:ASEarch:PAMSet SLOT1,3,TOP
```

**Compatibility**

Incompatible with existing models

---

### :SENSe:MEASure:EMARgin:MARGin:ASEarch:PAMSet? `<slot>,<interface>',[<unit>]`

**Parameter**

- `<slot>` = `<CHARACTER PROGRAM DATA>`
  - SLOT1 to SLOT6 Slots No. 1 to 6
  - 1 to 6 when using the MP1800A, 1 to 4 when using the MT1810A
- `<interface>` = `<DECIMAL NUMERIC PROGRAM DATA>`
  - 1 to 4 Data1 to 4
- `[<unit>]` = `<DECIMAL NUMERIC PROGRAM DATA>`
  - 1 to 4 Mainframe No. 1 to 4

When using two or more MT1810A units in serial connection, specify the mainframe number. Can be omitted. Mainframe No. 1 is specified when omitted.

**Response**

- `<threshold>` = `<CHARACTER RESPONSE DATA>`
  - TOP, MIDD, BOTT

**Function**

Queries search target threshold for running PAM Auto Search in the Eye Margin measurement.

**Example**

To query the search target threshold of Data 3 of Slot No.1.

```
> :SENSe:MEASure:EMARgin:MARGin:ASEarch:PAMSet? SLOT1,3
< 1
```

**Compatibility**

Incompatible with existing models
7.12.9.2 File menu setting commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
</table>

:SYSTem:MMEMory:MARGin:RECall <file_name>

Parameter

- `<file_name> = <STRING PROGRAM DATA>`
- "<drv>:\[<dir>]\<file>"
- `<drv> = C, D, E, F`
- `<dir> = <dir1>\<dir2>\... (Omitted when root directory)`
- `<file> = File name`

Function
Opens the Eye Margin measurement result data.

Example

> :SYSTem:MMEMory:MARGin:RECall "C:\Test\example"

Compatibility
Incompatible with existing models

:SYSTem:MMEMory:MARGin:STORe <file_name>,<file_type>

Parameter

- `<file_name> = <STRING PROGRAM DATA>`
- "<drv>:\[<dir>]\<file>"
- `<drv> = C, D, E, F`
- `<dir> = <dir1>\<dir2>\... (Omitted when root directory)`
- `<file> = File name`
- `<file_type> = <CHARACTER PROGRAM DATA>`
  - BIN Binary file
  - CSV CSV file
  - TXT Text file

Function
Saves the Eye Margin measurement result data.

*Note:*

The settings will not be read from the saved file if the file name is changed.

Example

To save the Eye Margin measurement result data by specifying the save destination, file name, and file format:

> :SYSTem:MMEMory:MARGin:STORe "C:\Test\example",TXT

Compatibility
Incompatible with existing models
Chapter 7  SCPI Commands

7.12.10  Eye Diagram Measurement

This section describes the commands for Eye Diagram measurement. Commands are described for each tab.

7.12.10.1  Condition tab

Figure 7.12.10-1  Eye Diagram Window

Figure 7.12.10.1-1  Condition Tab

Figure 7.12.10.1-2  Condition Tab (PAM mode)
### 7.12 28G/32G bit/s ED Commands

#### Table 7.12.10.1-1  Condition Tab Setting Commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>:DISPlay:RESult:EDIagram:TABSelect?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:MEASure:EDIagram:DIAGram:SLOT?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:MEASure:EDIagram:MTESt:SLOT?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:MEASure:EDIagram:DIAGram:TRANsition?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:MEASure:EDIagram:DIAGram:TYPE?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:MEASure:EDIagram:DIAGram:ASEarch?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:MEASure:EDIagram:DIAGram:ASEarch:PAMSet?</td>
</tr>
</tbody>
</table>

---

**:DISPlay:RESult:EDIagram:TABSelect <tab>**

**Parameter**

<tab> = <CHARACTER PROGRAM DATA>

**CONDition**  
Condition tab

**DIAGram**  
Diagram tab

**MASKedit**  
Mask Edit tab

**Function**

Switches the Eye Diagram measurement tab display.

**Example**

To display the Eye Diagram measurement Condition tab:

> :DISPlay:RESult:EDIagram:TABSelect CONDITION

**Compatibility**

Incompatible with existing models

---

**:DISPlay:RESult:EDIagram:TABSelect?**

**Response**

<tab> = <CHARACTER RESPONSE DATA>

**COND, DIAG, MASK**

**Function**

Queries the Eye Diagram measurement tab display.

**Example**

> :DISPlay:RESult:EDIagram:TABSelect?

< COND

**Compatibility**

Incompatible with existing models
Chapter 7  SCPI Commands

:SENSe:MEASure:EDIagram:DIAGram:SLOT
[slot],<interface>,<boolean>[,<unit>]

Parameter

- `<slot>` = <CHARACTER PROGRAM DATA>
  - SLOT1 to SLOT6  Slots No. 1 to 6
  - 1 to 6 when using the MP1800A, 1 to 4 when using the MT1810A
- `<interface>` = <DECIMAL NUMERIC PROGRAM DATA>
  - 1 to 4  Data1 to 4
- `<boolean>` = <BOOLEAN PROGRAM DATA>
  - ON or 1  Eye Diagram measurement ON
  - OFF or 0  Eye Diagram measurement OFF
- `<unit>` = <DECIMAL NUMERIC PROGRAM DATA>
  - 1 to 4  Mainframe No. 1 to 4

When using two or more MT1810A units in serial connection, specify the mainframe number.
Can be omitted. Mainframe No. 1 is specified when omitted.

Function

Selects the Eye Diagram measurement slot.

Example

To set Eye Diagram measurement Slot 1 to ON:

> :SENSe:MEASure:EDIagram:DIAGram:SLOT SLOT1,2,ON

Compatibility

Incompatible with existing models

:SENSe:MEASure:EDIagram:DIAGram:SLOT?
[slot],<interface>[,<unit>]

Parameter

- `<slot>` = <CHARACTER PROGRAM DATA>
  - SLOT1 to SLOT6  Slots No. 1 to 6
  - 1 to 6 when using the MP1800A, 1 to 4 when using the MT1810A
- `<interface>` = <DECIMAL NUMERIC PROGRAM DATA>
  - 1 to 4  Data1 to 4
- `<unit>` = <DECIMAL NUMERIC PROGRAM DATA>
  - 1 to 4  Mainframe No. 1 to 4

When using two or more MT1810A units in serial connection, specify the mainframe number.
Can be omitted. Mainframe No. 1 is specified when omitted.

Response

- `<boolean>` = <NR1 NUMERIC RESPONSE DATA>
  - 1  Eye Diagram measurement ON
  - 0  Eye Diagram measurement OFF

Function

Queries whether the specified slot is targeted for the Eye Diagram measurement.

Example

To query whether the Slot 1, Data 2 is targeted for the Eye Diagram measurement:

> :SENSe:MEASure:EDIagram:DIAGram:SLOT? SLOT1,2

Compatibility

Incompatible with existing models
:SENSe:MEASure:EDIagram:MTESt:SLOT
[slot],<interface>,<boolean>[,<unit>]

Parameter
(slot) = <CHARACTER PROGRAM DATA>
SLOT1 to SLOT6  Slots No. 1 to 6
1 to 6 when using the MP1800A, 1 to 4 when using the MT1810A

(interface) = <DECIMAL NUMERIC PROGRAM DATA>
1 to 4  Data1 to 4

(boolean) = <BOOLEAN PROGRAM DATA>
ON or 1  Mask Test measurement ON
OFF or 0  Mask Test measurement OFF

[unit] = <DECIMAL NUMERIC PROGRAM DATA>
1 to 4  Mainframe No. 1 to 4
When using two or more MT1810A units in serial connection, specify the
mainframe number.
Can be omitted. Mainframe No. 1 is specified when omitted.

Function
Selects the Mask Test measurement slot.

Example
To set Mask Test measurement Slot 1, Data 1 to be ON:
> :SENSe:MEASure:EDlagram:MTESt:SLOT SLOT1,1,ON

Compatibility
Incompatible with existing models

:SENSe:MEASure:EDlagram:MTESt:SLOT? <slot>,<interface>[,<unit>]

Parameter
(slot) = <CHARACTER PROGRAM DATA>
SLOT1 to SLOT6  Slots No. 1 to 6
1 to 6 when using the MP1800A, 1 to 4 when using the MT1810A

(interface) = <DECIMAL NUMERIC PROGRAM DATA>
1 to 4  Data1 to 4

[unit] = <DECIMAL NUMERIC PROGRAM DATA>
1 to 4  Mainframe No. 1 to 4
When using two or more MT1810A units in serial connection, specify the
mainframe number.
Can be omitted. Mainframe No. 1 is specified when omitted.

Response
(boolean) = <NR1 NUMERIC RESPONSE DATA>
1  ON
0  OFF

Function
Queries whether the specified slot is targeted for the Mask Test measurement.

Example
To query whether the slot 1, Data 1 is targeted for the Mask Test measurement:
> :SENSe:MEASure:EDlagram:MTESt:SLOT? SLOT1,1
< 1

Compatibility
Incompatible with existing models
Chapter 7  SCPI Commands

:SENSe:MEASure:EDiagram:DIAGram:TRANsition <type>
Parameter

\[ \text{<type> = <CHARACTER PROGRAM DATA>} \]
OFF         OFF
TRANsition  Transition
NONTran     Non Transition

Function
Sets Transition/Non Transition error detection.
Example
To set to Transition error detection:
\[ > :\text{SENSe:MEASure:EDiagram:DIAGram:TRANsition} \text{ TRANsition} \]
Compatibility
Incompatible with existing models

:SENSe:MEASure:EDiagram:DIAGram:TRANsition?
Response

\[ \text{<type> = <CHARACTER RESPONSE DATA>} \]
OFF, TRAN, NONT

Function
Queries the Transition/Non Transition error detection setting state.
Example
\[ > :\text{SENSe:MEASure:EDiagram:DIAGram:TRANsition} ? \]
\[ < \text{TRAN} \]

Compatibility
Incompatible with existing models

:SENSe:MEASure:EDiagram:DIAGram:TYPE <type>
Parameter

\[ \text{<type> = <CHARACTER PROGRAM DATA>} \]
POINt8      8 Point
POINt16     16 Point
POINt32     32 Point
POINt64     64 Point
POINt128    128 Point

Function
Sets the number of measurement points during Eye Diagram measurement.
Example
To set the number of measurement points during Eye Diagram measurement to 8 points:
\[ > :\text{SENSe:MEASure:EDiagram:DIAGram:TYPE} \text{ POINt8} \]
Compatibility
Incompatible with existing models

:SENSe:MEASure:EDiagram:DIAGram:TYPE?
Response

\[ \text{POIN8, POIN16, POIN32, POIN64, POIN128} \]

Function
Queries the number of measurement points during Eye Diagram measurement.
Example
\[ > :\text{SENSe:MEASure:EDiagram:DIAGram:TYPE} ? \]
\[ < \text{POIN8} \]

Compatibility
Incompatible with existing models
:SENSe:MEASure:EDIagram:DIAGram:ASEarch <setting>[,<mode>]

Parameter  
<setting> = <BOOLEAN PROGRAM DATA>
ON or 1 Auto Search ON
OFF or 0 Auto Search OFF

<mode> = <DECIMAL NUMERIC PROGRAM DATA>
1 Coarse
2 Fine
3 PAM Coarse
4 PAM Fine

Note:  
<mode> can be omitted. Coarse is specified when omitted.

Function  
Selects whether to execute Auto Search when starting the Eye Diagram Measurement.

Example  
To set Auto Search to On when starting the Eye Diagram Measurement:
> :SENSe:MEASure:EDIagram:DIAGram:ASEarch ON,2

Compatibility  
Incompatible with existing models

:SENSe:MEASure:EDIagram:DIAGram:ASEarch?

Response  
<setting> = <NR1 NUMERIC RESPONSE DATA>
0 OFF
1 ON (Coarse)
2 ON (Fine)
3 ON (PAM Coarse)
4 ON (PAM Fine)

Function  
Queries whether to execute Auto Search when starting the Eye Diagram Measurement.

Example  
> :SENSe:MEASure:EDIagram:DIAGram:ASEarch?
< 2

Compatibility  
Incompatible with existing models
Chapter 7  SCPI Commands

:SENSe:MEASure:EDiagram:DIAGram:ASEarch:PAMSet
[slot],<interface>,<threshold>[,<unit>]

Parameter

[slot] = <CHARACTER PROGRAM DATA>
SLOT1 to SLOT6    Slots No. 1 to 6
1 to 6 when using the MP1800A, 1 to 4 when using the MT1810A

<interface> = <DECIMAL NUMERIC PROGRAM DATA>
1 to 4    Data1 to 4

<threshold> = <CHARACTER PROGRAM DATA>
TOP    Searches the Top threshold
MIDDle    Searches the Middle threshold
BOTTom    Searches the Bottom threshold

[unit]=<DECIMAL NUMERIC PROGRAM DATA>
1 to 4    Mainframe No. 1 to 4
When using two or more MT1810A units in serial connection, specify the mainframe number.
Can be omitted. Mainframe No. 1 is specified when omitted.

Function
Selects search target threshold for running PAM Auto Search in the Eye Diagram measurement.

Example
To set Data 3 of Slot No.1 for Top threshold search:
> :SENSe:MEASure:EDiagram:DIAGram:ASEarch:PAMSet
SLOT1,3,TOP

Compatibility
Incompatible with existing models

:SENSe:MEASure:EDiagram:DIAGram:ASEarch:PAMSet?
[slot],<interface>[[,]<unit>]

Parameter

[slot] = <CHARACTER PROGRAM DATA>
SLOT1 to SLOT6    Slots No. 1 to 6
1 to 6 when using the MP1800A, 1 to 4 when using the MT1810A

<interface> = <DECIMAL NUMERIC PROGRAM DATA>
1 to 4    Data1 to 4

[unit]=<DECIMAL NUMERIC PROGRAM DATA>
1 to 4    Mainframe No. 1 to 4
When using two or more MT1810A units in serial connection, specify the mainframe number.
Can be omitted. Mainframe No. 1 is specified when omitted.

Response
<threshold> = < CHARACTER RESPONSE DATA>
TOP, MIDD, BOTT

Function
Queries search target threshold for running PAM Auto Search in the Eye Diagram measurement.

Example
To query the search target threshold of Data 3 of Slot No.1:
> :SENSe:MEASure:EDiagram:DIAGram:ASEarch:PAMSet? SLOT1,3
< 1

Compatibility
Incompatible with existing models

7-878
7.12.10.2 Diagram tab

Figure 7.12.10.2-1 Diagram Tab

7.12.10.2.1 Graph display setting commands

Figure 7.12.10.2.1-1 Graph Display Field
Table 7.12.10.2.1-1  Graph Display Setting Commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(Result display slot setting)</td>
<td>:DISPlay:RESult:EDiagram:SLOT?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:DISPlay:RESult:EDiagram:TEMPLate:SELect?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:DISPlay:RESult:EDiagram:TEMPLate:THReshold?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:DISPlay:RESult:EDiagram:SCALE:VOLTage?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:DISPlay:RESult:EDiagram:SCALE:PHASe?</td>
</tr>
</tbody>
</table>

:DISPlay:RESult:EDiagram:SLOT <slot>,<interface>[,<unit>]

Parameter

- `<slot>` = `<CHARACTER PROGRAM DATA>`
- SLOT1 to SLOT6, Slots No. 1 to 6
- 1 to 6 when using the MP1800A, 1 to 4 when using the MT1810A

- `<interface>` = `<DECIMAL NUMERIC PROGRAM DATA>`
- 1 to 4, Data 1 to 4

- `[<unit>]` = `<DECIMAL NUMERIC PROGRAM DATA>`
- 1 to 4, Mainframe No. 1 to 4

When using two or more MT1810A units in serial connection, specify the mainframe number.

Can be omitted. Mainframe No. 1 is specified when omitted.

Function
Selects the Eye Diagram result display target slot.

Example
To set the Eye Diagram result display target slot to slot No. 1, Data 1:
> :DISPlay:RESult:EDiagram:SLOT SLOT1,1

Compatibility
Incompatible with existing models
**:DISPlay:RESult:EDiagram:SLOT? [<unit>]**

**Parameter**

- `<unit>` = `<DECIMAL NUMERIC PROGRAM DATA>`
- 1 to 4: Mainframe No. 1 to 4
- When using two or more MT1810A units in serial connection, specify the mainframe number.
- Can be omitted. Mainframe No. 1 is specified when omitted.

**Response**

- `<slot>` = `<CHARACTER RESPONSE DATA>`
- SLOT1 to SLOT6: Slots No. 1 to 6
- `<interface>` = `<NR1 NUMERIC PROGRAM DATA>`
- 1 to 4: Data1 to 4

**Function**

Queries the Eye Diagram result display target slot.

**Example**

```plaintext
> :DISPlay:RESult:EDiagram:SLOT?
< SLOT1,1
```

**Compatibility**

Incompatible with existing models

---

**:DISPlay:RESult:EDiagram:TEMPlate:SELeクト <type>**

**Parameter**

- `<type>` = `<CHARACTER PROGRAM DATA>`
- E1: Edit1
- E2: Edit2
- E3: Edit3
- E4: Edit4

**Function**

Sets the mask pattern for performing the Eye Diagram mask test.

**Example**

To set the mask pattern for performing the Eye Diagram mask test to Edit2:

```plaintext
> :DISPlay:RESult:EDiagram:TEMPlate:SELeクト E2
```

**Compatibility**

Incompatible with existing models

---

**:DISPlay:RESult:EDiagram:TEMPlate:SELeクト?**

**Response**

- `<type>` = `<CHARACTER RESPONSE DATA>`
- E1, E2, E3, E4

**Function**

Queries the mask pattern for performing the Eye Diagram mask test.

**Example**

```plaintext
> :DISPlay:RESult:EDiagram:TEMPlate:SELeクト?
< E2
```

**Compatibility**

Incompatible with existing models
### Chapter 7  SCPI Commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>:DISPlay:RESult:EDiagram:TEMPlate:THReshold &lt;thre&gt;</strong></td>
<td>Sets the Error Threshold for performing the Eye Diagram mask test.</td>
</tr>
<tr>
<td><strong>Example</strong></td>
<td>To set the Error Threshold for performing the Eye Diagram mask test to E–3:</td>
</tr>
<tr>
<td></td>
<td>&gt; :DISPlay:RESult:EDiagram:TEMPlate:THReshold E_3</td>
</tr>
<tr>
<td><strong>Compatibility</strong></td>
<td>Incompatible with existing models</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>:DISPlay:RESult:EDiagram:TEMPlate:THReshold?</strong></td>
<td>Queries the Error Threshold for performing the Eye Diagram mask test.</td>
</tr>
<tr>
<td><strong>Example</strong></td>
<td>&gt; :DISPlay:RESult:EDiagram:TEMPlate:THReshold?</td>
</tr>
<tr>
<td></td>
<td>&lt; E_3</td>
</tr>
<tr>
<td><strong>Compatibility</strong></td>
<td>Incompatible with existing models</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>:DISPlay:RESult:EDiagram:SCALe:ASCale</strong></td>
<td>Automatically adjusts the Eye Diagram measurement graph scale.</td>
</tr>
<tr>
<td><strong>Example</strong></td>
<td>&gt; :DISPlay:RESult:EDiagram:SCALe:ASCale</td>
</tr>
<tr>
<td><strong>Compatibility</strong></td>
<td>Incompatible with existing models</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>:DISPlay:RESult:EDiagram:SCALe:VOLTage &lt;max&gt;,&lt;step&gt;</strong></td>
<td>Sets the maximum value in the direction of the Threshold of the Eye Diagram graph, and the grid resolution.</td>
</tr>
<tr>
<td><strong>Parameter</strong></td>
<td></td>
</tr>
</tbody>
</table>
| | <max> = <DECIMAL NUMERIC PROGRAM DATA> 
| | –3.990 to 4.000 –3.990 to 4.000 V, in 0.001 V steps 
| | <step> = <DECIMAL NUMERIC PROGRAM DATA> 
| | 0.001 to 0.800 0.001 to 0.800 V, in 0.001 V steps 
| **Function** | The range varies depending on the option status and input interface setting. 
| **Example** | To set the maximum value in the direction of the Threshold of the Eye Diagram graph to 4.000 V, and the grid resolution to 0.800 V: |
| | > :DISPlay:RESult:EDiagram:SCALe:VOLTage 4.000,0.800 |
| **Compatibility** | Incompatible with existing models |
### :DISPlay:RESult:EDiagram:SCALE:VOLTage?

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Response</td>
<td>&lt;max&gt; = &lt;NR2 NUMERIC RESPONSE DATA&gt;</td>
</tr>
<tr>
<td></td>
<td>–3.990 to 4.000 –3.990 to 4.000 V</td>
</tr>
<tr>
<td>&lt;step&gt; = &lt;NR2 NUMERIC RESPONSE DATA&gt;</td>
<td>0.001 to 0.800 0.001 to 0.800 V</td>
</tr>
<tr>
<td>Function</td>
<td>Queries the maximum value in the direction of the Threshold of the Eye Diagram graph, and the grid resolution.</td>
</tr>
<tr>
<td>Example</td>
<td>&gt; :DISPlay:RESult:EDiagram:SCALE:VOLTage?</td>
</tr>
<tr>
<td></td>
<td>&lt; 4.000, 0.800</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Incompatible with existing models</td>
</tr>
</tbody>
</table>

#### Example

To set the maximum value in the direction of the Phase of the Eye Diagram graph to 1000, and the grid resolution to 10 mUI:

```plaintext
> :DISPlay:RESult:EDiagram:SCALE:PHASe 1000,10
```

#### Compatibility

Incompatible with existing models

### :DISPlay:RESult:EDiagram:SCALE:PHASe <max>,<step>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Response</td>
<td>&lt;max&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
</tr>
<tr>
<td></td>
<td>–990 to 1000 –990 to 1000 mUI, in 1-mUI steps</td>
</tr>
<tr>
<td>&lt;step&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
<td>1 to 200 1 to 200 mUI, in 1-mUI steps</td>
</tr>
<tr>
<td>Function</td>
<td>Sets the maximum value in the direction of the Phase of the Eye Diagram graph, and the grid resolution.</td>
</tr>
<tr>
<td>Example</td>
<td>To set the maximum value in the direction of the Phase of the Eye Diagram graph to 1000, and the grid resolution to 10 mUI:</td>
</tr>
<tr>
<td></td>
<td>&gt; :DISPlay:RESult:EDiagram:SCALE:PHASe 1000,10</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Incompatible with existing models</td>
</tr>
</tbody>
</table>

### :DISPlay:RESult:EDiagram:SCALE:PHASe?

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Response</td>
<td>&lt;max&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt;</td>
</tr>
<tr>
<td>&lt;step&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt;</td>
<td></td>
</tr>
<tr>
<td>Function</td>
<td>Queries the maximum value in the direction of the Phase of the Eye Diagram graph, and the grid resolution.</td>
</tr>
<tr>
<td>Example</td>
<td>&gt; :DISPlay:RESult:EDiagram:SCALE:PHASe?</td>
</tr>
<tr>
<td></td>
<td>&lt; 1000,10</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Incompatible with existing models</td>
</tr>
</tbody>
</table>
7.12.10.2.2 Condition setting commands

Figure 7.12.10.2.2-1  Condition Tab (Actual)

Figure 7.12.10.2.2-2  Condition Tab (Estimate)
### Table 7.12.10.2.2-1  Condition Tab Setting Commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>:DISPLAY:RESULT:EDiagram:TABSelect?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:MEASure:DIAGram:THReshold?</td>
</tr>
<tr>
<td></td>
<td>Meas reset All</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>:DISPLAY:RESULT:EDiagram:ERATe?</td>
</tr>
<tr>
<td></td>
<td>Display reset All</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Estimate measurement</td>
<td>DISPLAY:RESULT:EDiagram:EERate?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:DISPLAY:RESULT:EDiagram:DISPLAY:MASK?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:DISPLAY:RESULT:EDiagram:ESTimate?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:MEASure:DIAGram:BTHReshold?</td>
</tr>
<tr>
<td>[12]</td>
<td>Query for measurement result</td>
<td>:CALCulate:DATA:DIAGram?</td>
</tr>
</tbody>
</table>

**:DISPLAY:RESULT:EDiagram:DIAGram:TABSelect <tab>**

**Parameter**

- `<tab>` = `<CHARACTER PROGRAM DATA>`
- `CONDition` Condition tab
- `RESult` Result tab

**Function**

Switches the tab display within the Diagram tab.

**Example**

To display the Result tab within the Diagram tab:

```
> :DISPLAY:RESULT:EDiagram:DIAGram:TABSelect RESult
```

**Compatibility**

Incompatible with existing models

**:DISPLAY:RESULT:EDiagram:DIAGram:TABSelect?**

**Response**

- `<tab>` = `<CHARACTER RESPONSE DATA>`
- `COND, RES`

**Function**

Queries the tab selected in the Diagram tab.

**Example**

```
> :DISPLAY:RESULT:EDiagram:DIAGram:TABSelect?
< COND
```

**Compatibility**

Incompatible with existing models
Chapter 7  SCPI Commands

:SENSe:MEASure:EDIagram:DIAGram:THReshold <thre>,<boolean>

Parameter  
- `<type>` = <CHARACTER PROGRAM DATA>
  - E_3 to E_12  1E–3 to 1E–12/1 Step
- `<boolean>` = <BOOLEAN PROGRAM DATA>
  - ON or 1  Selects ON
  - OFF or 0  Selects OFF

Function  
Sets the Eye Diagram measurement target error rate.

Example  
To set the Eye Diagram measurement target error rate to 1E–12:
> :SENSe:MEASure:EDIagram:DIAGram:THReshold E_12,ON

Compatibility  
Incompatible with existing models

:SENSe:MEASure:EDIagram:DIAGram:THReshold?

Response  
- `<type>` = <CHARACTER RESPONSE DATA>
  - E_3 to E_12  1E–3 to 1E–12
  - NONE  When all OFF

Function  
Queries the Eye Diagram measurement target error rate.

Example  
> :SENSe:MEASure:EDIagram:DIAGram:THReshold?
< E_3,E_10,E_12

Compatibility  
Incompatible with existing models

:SENSe:MEASure:EDIagram:DIAGram:ATHReshold <boolean>

Parameter  
- `<boolean>` = <BOOLEAN PROGRAM DATA>
  - ON or 1  Sets all ON
  - OFF or 0  Sets all OFF

Function  
Sets all Eye Diagram measurement target error rates to be ON or OFF.

Example  
To set all Eye Diagram measurement target error rates to be ON:
> :SENSe:MEASure:EDIagram:DIAGram:ATHReshold ON

Compatibility  
Incompatible with existing models
### :DISPlay:RESult:EDiagram:ERATe <thre>,<boolean>

**Parameter**
- `<type>` = `<CHARACTER PROGRAM DATA>`
  - E_3 to E_17: 1E–3 to 1E–17/1 Step
- `<boolean>` = `<BOOLEAN PROGRAM DATA>`
  - ON or 1: Display select ON
  - OFF or 0: Display select OFF

**Function**
Sets the target error rate of the Eye Diagram measurement result display.

**Example**
To set the target error rate of the Eye Diagram measurement result display to 1E–12:
```
> :DISPlay:RESult:EDiagram:ERATe E_12,1
```

**Compatibility**
Incompatible with existing models

### :DISPlay:RESult:EDiagram:ERATe?

**Response**
- `<type>` = `<CHARACTER RESPONSE DATA>`
  - E_3 to E_17: 1E–3 to 1E–17
  - NONE: When all OFF
  - The selected threshold is delimited with a comma (,) and returned.

**Function**
Queries the target error rate of the Eye Diagram measurement result display.

**Example**
```
> :DISPlay:RESult:EDiagram:ERATe?
< NONE
```

**Compatibility**
Incompatible with existing models

### :DISPlay:RESult:EDiagram:AERate <boolean>

**Parameter**
- `<boolean>` = `<BOOLEAN PROGRAM DATA>`
  - ON or 1: Sets all ON
  - OFF or 0: Sets all OFF

**Function**
Sets all target error rates of the Eye Diagram measurement result display to be ON or OFF.

**Example**
To set all target error rates of the Eye Diagram measurement result display to be ON:
```
> :DISPlay:RESult:EDiagram:AERate 1
```

**Compatibility**
Incompatible with existing models
### :DISPlay:RESult:EDiagram:EERate <numeric>

**Parameter**
- `<numeric>` = `<DECIMAL PROGRAM DATA>`
  - 13 to 199
  - 1E–13 to 1E–199/1 Step
- Up to five parameters can be set, delimited with a comma (,).

**Function**
Sets the result display error rate in Estimate measurement from E–13 to E–199.

**Example**
To display the result display error rate in Estimate measurement for 1E–15, 1E–100, 1E–150, 1E–199, and 1E–180:
```
> :DISPlay:RESult:EDiagram:EERate 15,100,150,199,180
```

**Compatibility**
Incompatible with existing models

### :DISPlay:RESult:EDiagram:EERate?

**Response**
- `<numeric>` = `<NR1 NUMERIC RESPONSE DATA>`
  - 13 to 199
  - 1E–13 to 1E–199
- Returns five responses, delimited with a comma (,).

**Function**
Queries the result display error rate setting in Estimate measurement from E–13 to E–199.

**Example**
```
> :DISPlay:RESult:EDiagram:EERate?
< 15,100,150,199,180
```

**Compatibility**
Incompatible with existing models

### :DISPlay:RESult:EDiagram:DISPlay:MASK <boolean>

**Parameter**
- `<boolean>` = `<BOOLEAN PROGRAM DATA>`
  - ON or 1  Display
  - OFF or 0  Clear

**Function**
Selects between displaying and clearing the mask test result during Eye Diagram measurement.

**Example**
To display the mask test result during Eye Diagram measurement:
```
> :DISPlay:RESult:EDiagram:DISPlay:MASK ON
```

**Compatibility**
Incompatible with existing models

### :DISPlay:RESult:EDiagram:DISPlay:MASK?

**Response**
- `<boolean>` = `<NR1 NUMERIC RESPONSE DATA>`
  - 1  Display
  - 0  Clear

**Function**
Queries the setting whether to display or clear the mask test result during Eye.

**Example**
```
> :DISPlay:RESult:EDiagram:DISPlay:MASK?
< 1
```

**Compatibility**
Incompatible with existing models
:DISPlay:RESult:EDiagram:ESTimate <type>

Parameter
<type> = <CHARACTER PROGRAM DATA>
  ACTual        Actual measurement
  ESTimate     Estimate measurement

Note:
The Estimate measurement cannot be selected when the number of measurement points is set to 8.

Function
Selects between Actual and Estimate measurement during Eye Diagram measurement.

Example
To set Actual measurement during Eye Diagram measurement:
> :DISPlay:RESult:EDiagram:ESTimate ACTual

Compatibility
Incompatible with existing models

:DISPlay:RESult:EDiagram:ESTimate?

Response
<type> = <CHARACTER RESPONSE DATA>
  ACT  Actual measurement
  EST  Estimate measurement

Function
Queries the Actual/Estimate measurement state during Eye Diagram measurement.

Example
> :DISPlay:RESult:EDiagram:ESTimate?
< ACT

Compatibility
Incompatible with existing models

:SENSe:MEASure:EDiagram:DIAGram:BTHReshold <upper>,<lower>

Parameter
<upper> = <CHARACTER PROGRAM DATA>
  E_3 to E_7  1E–3 to 1E–7/1 Step
  E_7 to E_12 1E–7 to E–12/1 Step

<lower> = <CHARACTER PROGRAM DATA>
  E_7 to E_12 1E–7 to E–12/1 Step

Note:
The difference between the <upper> and <lower> value must be at least 3.

Function
Sets the reference error rate during Eye Diagram Estimate.

Example
To set the reference error rate during Eye Diagram Estimate to E–3 and E–12:
> :SENSe:MEASure:EDiagram:DIAGram:BTHReshold E_3,E_12

Compatibility
Incompatible with existing models
Chapter 7  SCPI Commands

:SENSe:MEASure:EDIagram:DIAGram:BTHReshold?

Response

<upper> = <CHARACTER RESPONSE DATA>
E_3 to E_7  1E−3 to 1E−7
<lower> = <CHARACTER RESPONSE DATA>
E_7 to E_12  1E−7 to E−12

Function
Queries the reference error rate during Eye Diagram Estimate.

Example
> :SENSe:MEASure:EDiagram:DIAGram:BTHReshold?
< E_3,E_12

Compatibility
Incompatible with existing models

:DISPlay:RESult:EDIagram:ADJust

Function
Automatically adjusts the mask pattern at the Eye Diagram measurement.

Example
> :DISPlay:RESult:EDIagram:ADJust

Compatibility
Incompatible with existing models

:SENSe:MEASure:EDiagram:DIAGram:POINt?
<slot>,<interface>,<thre>[,<unit>]

Parameter

<slot> = <CHARACTER PROGRAM DATA>
SLOT1 to SLOT6  Slots No. 1 to 6
<interface> = <DECIMAL NUMERIC PROGRAM DATA>
1 to 4  Data1 to 4
<thre> = <CHARACTER PROGRAM DATA>
E_3 to E_12  1E−3 to 1E−12
[unit] = <DECIMAL NUMERIC PROGRAM DATA>
1 to 4  Mainframe No. 1 to 4

When using two or more MT1810A units in serial connection, specify the mainframe number. Can be omitted. Mainframe No. 1 is specified when omitted.

Response

<numeric> = <NR1 NUMERIC RESPONSE DATA>
1 to 128  Returns a valid point.
0  Returns 0 when all points are invalid.
Multiple valid points are delimited with a comma (,) and returned

Function
Queries the valid points in each threshold during the Eye Diagram measurement for a specific slot.

Example
To query the valid points in slot No. 1 during measurement with threshold E–3:
> :SENSe:MEASure:EDiagram:DIAGram:POINt? SLOT1,1,E_3
< 1,2,3,4,128

Compatibility
Incompatible with existing models
### 7.12 28G/32G bit/s ED Commands

**:CALCulate:DATA:DIAGram? <slot>,<interface>,<string>[,<unit>]**

**Parameter**

- **<slot> = <CHARACTER PROGRAM DATA>**
  - SLOT1 to SLOT6 Slots No. 1 to 6
- **<interface> = <DECIMAL NUMERIC PROGRAM DATA>**
  - 1 to 4 Data1 to 4
- **<string> = <STRING PROGRAM DATA>**
  - Measurement data
  - See Table 7.12.10.2.2-2 for the <string> contents.

**Table 7.12.10.2.2-2  Eye Diagram Measurement Item<string>**

<table>
<thead>
<tr>
<th>Item</th>
<th>&lt;string&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Eye Map type</td>
<td>1E–3, &quot;MAP:E_3:POINt1&quot;,...,&quot;MAP:E_3:POINt128&quot;</td>
</tr>
<tr>
<td></td>
<td>1E–4, &quot;MAP:E_4:POINt1&quot;,...,&quot;MAP:E_4:POINt128&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1E–11, &quot;MAP:E_11:POINt1&quot;,...,&quot;MAP:E_11:POINt128&quot;</td>
</tr>
<tr>
<td></td>
<td>1E–12, &quot;MAP:E_12:POINt1&quot;,...,&quot;MAP:E_12:POINt128&quot;</td>
</tr>
</tbody>
</table>

- **<unit> = <DECIMAL NUMERIC PROGRAM DATA>**
  - 1 to 4 Mainframe No. 1 to 4

When using two or more MT1810A units in serial connection, specify the mainframe number.

Can be omitted. Mainframe No. 1 is specified when omitted.

**Response**

- **<result> = <STRING RESPONSE DATA>**

**Table 7.12.10.2.2-3  Response Format**

<table>
<thead>
<tr>
<th>Type</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Form1</td>
<td>&quot;XXXXX,YYYYY&quot;</td>
<td>XXXXX: Phase (mUI)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>XXXXX: Phase (ps) (Displays in the selected unit.)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>YYYYY: Threshold (mV)</td>
</tr>
<tr>
<td></td>
<td>&quot;-----,-----&quot;</td>
<td>When no data corresponds to the query.</td>
</tr>
</tbody>
</table>

**Function**

Queries the Eye Diagram measurement result of a specific slot and point.

**Example**

To query the Eye Diagram measurement result at slot No. 1, Data 2, E–3, point 1:

```plaintext
>:CALCulate:DATA:DIAGram? SLOT1,2,"MAP:E_3:POINt1"
<" 100, 1000"
```

**Compatibility**

Incompatible with existing models.
7.12.10.2.3 Result query command

![Figure 7.12.10.2.3-1 Result Tab](image)

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Item</th>
<th>Command</th>
</tr>
</thead>
</table>
### 7.12 28G/32G bit/s ED Commands

**:CALCulate:DATA:EDiagram:MASK? <slot>,<interface>[,<unit>]**

**Parameter**

- `<slot>` = `<CHARACTER PROGRAM DATA>`
  - SLOT1 to SLOT6 Slots No. 1 to 6
- `<interface>` = `<DECIMAL NUMERIC PROGRAM DATA>`
  - 1 to 4 Data1 to 4
- `<unit>` = `<DECIMAL NUMERIC PROGRAM DATA>`
  - 1 to 4 Mainframe No. 1 to 4

When using two or more MT1810A units in serial connection, specify the mainframe number. Can be omitted. Mainframe No. 1 is specified when omitted.

**Response**

- `<string>` = `<STRING RESPONSE DATA>`

**Table 7.12.10.2.3-2  Response Format**

<table>
<thead>
<tr>
<th>Type</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Form1</td>
<td>&quot;XX,Y,Y,...Y&quot;</td>
<td>XX: Pass, Failure</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Y: 1 to 32 (Failure mask point)</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Note:</strong> Multiple NGs are delimited with a comma (,) and returned for the corresponding number of NGs.</td>
</tr>
<tr>
<td></td>
<td>&quot;...&quot;</td>
<td>When no data corresponds to the query.</td>
</tr>
</tbody>
</table>

**Function**

Queries the measurement result of the Eye Diagram Mask Test.

**Example**

To query the Mask Test result of Mainframe No.3 Slot1 Data 1:

```plaintext
>:CALCulate:DATA:EDiagram:MASK? SLOT1,1,3
"Failure,1,2,3,4,32"
"Pass"
```

**Compatibility**

Incompatible with existing models (Parameter varies by model.)
7.12.10.2.4 Diagram tab (Status) setting commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>[4]</td>
<td>No label (Setting of time display)</td>
<td>:SENSe:MEASure:EDIagram:TIME</td>
</tr>
<tr>
<td>[6]</td>
<td>Query for measurement start time</td>
<td>:SENSe:MEASure:EDIagram:STARt?</td>
</tr>
<tr>
<td>[7]</td>
<td>Query for measurement elapsed time</td>
<td>:SENSe:MEASure:EDIagram:ELAPsed?</td>
</tr>
<tr>
<td>[8]</td>
<td>Query for measurement status</td>
<td>:CALCulate:DATA:EDIagram:STATus?</td>
</tr>
</tbody>
</table>

**:SENSe:MEASure:EDIagram:DIAGram:STARt**

- **Function**: Starts the Eye Diagram measurement.
- **Example**: `>:SENSe:MEASure:EDIagram:DIAGram:STARt`
- **Compatibility**: Incompatible with existing models
7.12 28G/32G bit/s ED Commands

:SENSe:MEASure:EDIagram:DIAGram:STOP

Function
Stops the Eye Diagram measurement.

Example
> :SENSe:MEASure:EDIagram:DIAGram:STOP

Compatibility
Incompatible with existing models

:SENSe:MEASure:EDIagram:STATe?

Response

<table>
<thead>
<tr>
<th>&lt;numeric&gt;</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Measurement has been started.</td>
</tr>
<tr>
<td>0</td>
<td>Measurement has been stopped.</td>
</tr>
<tr>
<td>-1</td>
<td>Measurement has failed.</td>
</tr>
</tbody>
</table>

Function
Queries the Eye Diagram measurement status.

Example

> :SENSe:MEASure:EDIagram:STATe?

< 1

Compatibility
Incompatible with existing models

:SENSe:MEASure:EDIagram:TIME <type>

Parameter

<table>
<thead>
<tr>
<th>&lt;type&gt;</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DTIMe</td>
<td>Displays the current date and time.</td>
</tr>
<tr>
<td>STARt</td>
<td>Displays the measurement start time.</td>
</tr>
<tr>
<td>ELAPsed</td>
<td>Displays the elapsed time based on the measurement period.</td>
</tr>
</tbody>
</table>

Function
Selects the Eye Diagram measurement time display type.

Example

To set the Eye Diagram measurement time display type to measurement start time (Start Time):

> :SENSe:MEASure:EDIagram:TIME STARt

Compatibility
Incompatible with existing models

:SENSe:MEASure:EDIagram:TIME?

Response

<table>
<thead>
<tr>
<th>&lt;type&gt;</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DTIM</td>
<td>Displays the current date and time.</td>
</tr>
<tr>
<td>STAR</td>
<td>Displays the measurement start time.</td>
</tr>
<tr>
<td>ELAP</td>
<td>Displays the elapsed time based on the measurement period.</td>
</tr>
</tbody>
</table>

Function
Queries the Eye Diagram measurement time display type.

Example

> :SENSe:MEASure:EDIagram:TIME?

< STAR

Compatibility
Incompatible with existing models
Chapter 7  SCPI Commands

:SENSe:MEASure:EDIoagram:DTIMe?

Response

<table>
<thead>
<tr>
<th>Response</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;year&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt;</td>
<td>2000 to 2036 Year 2000 to 2036</td>
</tr>
<tr>
<td>&lt;month&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt;</td>
<td>1 to 12 January to December</td>
</tr>
<tr>
<td>&lt;day&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt;</td>
<td>1 to 31 1st to 31st</td>
</tr>
<tr>
<td>&lt;hour&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt;</td>
<td>0 to 23 0 to 23 hours</td>
</tr>
<tr>
<td>&lt;minute&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt;</td>
<td>0 to 59 0 to 59 minutes</td>
</tr>
<tr>
<td>&lt;second&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt;</td>
<td>0 to 59 0 to 59 seconds</td>
</tr>
</tbody>
</table>

Function
Queries the current date and time during Eye Diagram measurement.

Example
> :SENSe:MEASure:EDIoagram:DTIMe?
< 2012,4,1,23,59,59

Compatibility
Incompatible with existing models

:SENSe:MEASure:EDIoagram:STARt?

Response

<table>
<thead>
<tr>
<th>Response</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;year&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt;</td>
<td>0, 2000 to 2036 Year 2000 to 2036</td>
</tr>
<tr>
<td>&lt;month&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt;</td>
<td>0, 1 to 12 January to December</td>
</tr>
<tr>
<td>&lt;day&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt;</td>
<td>0, 1 to 31 1st to 31st</td>
</tr>
<tr>
<td>&lt;hour&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt;</td>
<td>0 to 23 0 to 23 hours</td>
</tr>
<tr>
<td>&lt;minute&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt;</td>
<td>0 to 59 0 to 59 minutes</td>
</tr>
<tr>
<td>&lt;second&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt;</td>
<td>0 to 59 0 to 59 seconds</td>
</tr>
</tbody>
</table>

Note:
The following is output when no measurement is performed:
0,0,0,0,0,0

Function
Queries the Eye Diagram measurement start time (Start Time).

Example
> :SENSe:MEASure:EDIoagram:STARt?
< 2012,4,1,23,59,59

Compatibility
Incompatible with existing models
:SENSe:MEASure:EDIagram:ELAPsed?

Response

<table>
<thead>
<tr>
<th>&lt;day&gt;</th>
<th>&lt;NR1 NUMERIC RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>0, 1 to 99</td>
<td>1 to 99 days</td>
</tr>
<tr>
<td>&lt;hour&gt;</td>
<td>&lt;NR1 NUMERIC RESPONSE DATA&gt;</td>
</tr>
<tr>
<td>0 to 23</td>
<td>0 to 23 hours</td>
</tr>
<tr>
<td>&lt;minute&gt;</td>
<td>&lt;NR1 NUMERIC RESPONSE DATA&gt;</td>
</tr>
<tr>
<td>0 to 59</td>
<td>0 to 59 minutes</td>
</tr>
<tr>
<td>&lt;second&gt;</td>
<td>&lt;NR1 NUMERIC RESPONSE DATA&gt;</td>
</tr>
<tr>
<td>0 to 59</td>
<td>0 to 59 seconds</td>
</tr>
</tbody>
</table>

Note:
The following is output when no measurement is performed:
0,0,0,0

Function
Queries the Eye Diagram measurement elapsed time (Elapsed Time).

Example
> :SENSe:MEASure:EDIagram:ELAPsed?
< 99,23,59,59

Compatibility
Incompatible with existing models

:CALCulate:DATA:EDIagram:STATus?

Response

<table>
<thead>
<tr>
<th>&lt;string&gt;</th>
<th>&lt;STRING RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;&quot;</td>
<td>When no alarm exists</td>
</tr>
<tr>
<td>&quot;Sync Loss&quot;</td>
<td>Sync Loss</td>
</tr>
<tr>
<td>&quot;Clock Loss&quot;</td>
<td>Clock Loss</td>
</tr>
<tr>
<td>&quot;Out of range&quot;</td>
<td>Out of range</td>
</tr>
<tr>
<td>&quot;Illegal Error&quot;</td>
<td>Illegal Error</td>
</tr>
<tr>
<td>&quot;Meas. Threshold exceeded&quot;</td>
<td>Number of errors exceeds the target rate</td>
</tr>
</tbody>
</table>

Note:
When two or more alarms have occurred, they are displayed in the following order:
Illegal Error > Clock Loss > Sync Loss > Out or range

Function
Queries the status condition during Eye Diagram measurement.

Example
> :CALCulate:DATA:EDIagram:STATus?
< "Sync Loss"

Compatibility
Incompatible with existing models
Chapter 7  SCPI Commands

7.12.10.2.5 Mask Edit tab setting commands

Table 7.12.10.2.5-1  Mask Edit Tab Setting Commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>:DISPLAY:RES ult:EDIagram:TEMplate:PATTERN?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:DISPLAY:RES ult:EDIagram:TEMplate:SCALE?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:DISPLAY:RES ult:EDIagram:TEMplate:PScale?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:DISPLAY:RES ult:EDIagram:TEMplate:EDIT?</td>
</tr>
</tbody>
</table>
:DISPlay:RESult:EDiagram:TEMPlate:PATTern <type>

Parameter
- <type> = <CHARACTER PROGRAM DATA>
  - E1: Edit1
  - E2: Edit2
  - E3: Edit3
  - E4: Edit4

Function
Sets the pattern to be edited during Eye Diagram measurement.

Example
To set the pattern to be edited during Eye Diagram measurement to Edit4:
> :DISPlay:RESult:EDiagram:TEMPlate:PATTern E4

Compatibility
Incompatible with existing models

:DISPlay:RESult:EDiagram:TEMPlate:PA TTern?

Response
- <type> = <CHARACTER RESPONSE DATA>
  - E1, E2, E3, E4

Function
Queries the pattern that was edited during Eye Diagram measurement.

Example
> :DISPlay:RESult:EDiagram:TEMPlate:PA TTern?
< E4

Compatibility
Incompatible with existing models

:DISPlay:RESult:EDiagram:TEMPlate:ASCale

Function
Executes automatic adjustment of the Mask Edit measurement graph scale.

Example
> :DISPlay:RESult:EDiagram:TEMPlate:ASCale

Compatibility
Incompatible with existing models

:DISPlay:RESult:EDiagram:TEMPlate:SCAl e <max>,<step>

Parameter
- <max> = <DECIMAL NUMERIC PROGRAM DATA>
  - –3.990 to 4.000
  - 1 to 800
- <step> = <DECIMAL NUMERIC PROGRAM DATA>
  - –3.990 to 4.000 V/0.001 V Step
  - 1 to 800 mV/1 mV Step

Function
Sets the maximum value and the grid resolution in the direction of the Threshold of the Mask Edit graph.

Example
To set the maximum value and the grid resolution in the direction of the Threshold of the Mask Edit graph to 4.000 V and 800 mV, respectively:
> :DISPlay:RESult:EDiagram:TEMPlate:SCAl e 4.000,800

Compatibility
Incompatible with existing models
Chapter 7  SCPI Commands

:DISPlay:RESult:EDiagram:TEMPLATE:SCALE?

Response  
$max = <NR2\ \text{NUMERIC\ RESPONSE\ DATA}>$

$step = <NR2\ \text{NUMERIC\ RESPONSE\ DATA}>$

Function  
Queries the maximum value and the grid resolution in the direction of the Threshold of the Mask Edit graph.

Example  
$>:\text{DISPlay:RESult:EDiagram:TEMPLATE:SCALE}?$

$4000,800$

Compatibility  
Incompatible with existing models

:DISPlay:RESult:EDiagram:TEMPLATE:PScale $<max>,<step>$

Parameter  
$max = <\text{DECIMAL\ NUMERIC\ PROGRAM\ DATA}>$

$-990$ to $1000$  $-990$ to $1000$ mUI/1 mUI Step

$step = <\text{DECIMAL\ NUMERIC\ PROGRAM\ DATA}>$

$1$ to $200$  $1$ to $200$ mUI/1 mUI Step

Function  
Sets the maximum value and the grid resolution in the direction of the Phase of the Mask Edit graph.

Example  
To set the maximum value and the grid resolution in the direction of the Phase of the Mask Edit graph to $1000$ and $10$ mUI, respectively:

$>:\text{DISPlay:RESult:EDiagram:TEMPLATE:PScale} 1000,10$

Compatibility  
Incompatible with existing models

:DISPlay:RESult:EDiagram:TEMPLATE:PScale?

Response  
$max = <\text{NR2\ \text{NUMERIC\ RESPONSE\ DATA}>}$

$step = <\text{NR2\ \text{NUMERIC\ RESPONSE\ DATA}>}$

Function  
Queries the maximum value and the grid resolution in the direction of the Phase of the Mask Edit graph.

Example  
$>:\text{DISPlay:RESult:EDiagram:TEMPLATE:PScale}?$

$1000,10$

Compatibility  
Incompatible with existing models

:DISPlay:RESult:EDiagram:TEMPLATE:SAMPle $<\text{string}>$

Parameter  
$<\text{string}> = <\text{STRING\ PROGRAM\ DATA}>$

"<drv>:\<dir>\<file>"$

$<\text{drv}> = C, D, E, F$

$<\text{dir}> = <\text{dir1>\<dir2>\...}(\text{Omitted\ when\ root\ directory})$

$<\text{file}> = \text{File\ name}$

Function  
Expands the selected sample pattern to the Edit Pattern.

Example  
$>:\text{DISPlay:RESult:EDiagram:TEMPLATE:SAMPle}$

"C:\Test\sample.MSK"

Compatibility  
Incompatible with existing models
:DISPlay:RESult:EDiagram:TEMPLate:PNUMber?

Response  
<numeric> = <NR1 NUMERIC RESPONSE DATA>  
0 to 32 0 to 32 points

Function  
Queries the number of Mask points.

Example  
> :DISPlay:RESult:EDiagram:TEMPLate:PNUMber?  
< 0

Compatibility  
Incompatible with existing models

:DISPlay:RESult:EDiagram:TEMPLate:EDIT <point>,<phase>,<voltage>

Parameter  
<point> = <DECIMAL NUMERIC PROGRAM DATA>  
1 to 32 1 to 32 points, in 1-point steps

<phase> = <DECIMAL NUMERIC PROGRAM DATA>  
–1000 to 1000 –1000 to 1000 mUI/1 mUI Step

<voltage> = <DECIMAL NUMERIC PROGRAM DATA>  
–4.000 to 4.000 –4.000 to 4.000 V/0.001 V Step

Function  
Sets a Mask point position.

Example  
To set a Mask point position (1 point, 1000 mUI, 4.000 V):  
> :DISPlay:RESult:EDiagram:TEMPLate:EDIT 1,1000,4.000

Compatibility  
Incompatible with existing models

:DISPlay:RESult:EDiagram:TEMPLate:EDIT? <point>

Parameter  
<point> = <DECIMAL NUMERIC PROGRAM DATA>  
1 to 32 1 to 32 points

Response  
<phase> = <NR1 NUMERIC RESPONSE DATA>  
<voltage> = <NR1 NUMERIC RESPONSE DATA>

Function  
Queries a Mask point position.

Example  
To query the position of point 1:  
> :DISPlay:RESult:EDiagram:TEMPLate:EDIT? 1  
< 1000,4.000

Compatibility  
Incompatible with existing models

:DISPlay:RESult:EDiagram:TEMPLate:PLOT <phase>,<voltage>

Parameter  
<phase> = <DECIMAL NUMERIC PROGRAM DATA>  
–1000 to 1000 –1000 to 1000 mUI/1 mUI Step

<voltage> = <DECIMAL NUMERIC PROGRAM DATA>  
–4000 to 4000 –4000 to 4000 mV/1 mV Step

Function  
Sets a Mask point position.

Example  
To set a Mask point position to 1000 mUI (phase), 4000 mV (voltage):  
> :DISPlay:RESult:EDiagram:TEMPLate:PLOT 1000,4000

Compatibility  
Incompatible with existing models
### Chapter 7  SCPI Commands

#### :DISPlay:RESult:EDIagram:TEMPLATE:DELete <pos>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;pos&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1 to 32</td>
</tr>
<tr>
<td></td>
<td>1 to 32 points, in 1-point steps</td>
</tr>
</tbody>
</table>

**Function**
Deletes the selected Mask point.

**Example**
To delete Mask point 3:

```
> :DISPlay:RESult:EDIagram:TEMPLATE:DELete 3
```

**Compatibility**
Incompatible with existing models.

#### :DISPlay:RESult:EDIagram:TEMPLATE:CLEar

**Function**
Clears the selected Mask pattern.

**Example**

```
> :DISPlay:RESult:EDIagram:TEMPLATE:CLEar
```

**Compatibility**
Incompatible with existing models.

#### :DISPlay:RESult:EDIagram:TEMPLATE:VERTical

**Function**
Plots a Mask pattern onto the vertically-symmetric position.

**Example**

```
> :DISPlay:RESult:EDIagram:TEMPLATE:VERTical
```

**Compatibility**
Incompatible with existing models.

#### :DISPlay:RESult:EDIagram:TEMPLATE:HORizontal

**Function**
Plots a Mask pattern onto the horizontally-symmetric position.

**Example**

```
> :DISPlay:RESult:EDIagram:TEMPLATE:HORizontal
```

**Compatibility**
Incompatible with existing models.
7.12.10.2.6 File menu setting commands

Table 7.12.10.2.6-1  File Menu Setting Commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
</table>

:SYSTem:MMEMory:DIAGram:RECall <file_name>,<file_type>

Parameter

- `<file_name>` = <STRING PROGRAM DATA>
- "<drv>:\<dir>\<file>"
- `<drv>` = C, D, E, F
- `<dir>` = <dir1><dir2>...(Omitted when root directory)
- `<file>` = File name
- `<file_type>` = <CHARACTER PROGRAM DATA>
  - BIN Binary file
  - TXT Text (UMP) file

Function

Opens the Eye Diagram measurement result data.

Example

> :SYSTem:MMEMory:DIAGram:RECall "C:\\Test\\example",TXT

Compatibility

Incompatible with existing models
Chapter 7  SCPI Commands

:SYSTem:MMEMory:DIAGram:STORe  
<file_name>,<data_type>,<file_type>

Parameter

<file_name> = <STRING PROGRAM DATA>
"<drv>:\[<dir>]<file>"
<drv> = C, D, E, F
<dir> = <dir1>\<dir2>\...(Omitted when root directory)
<file> = File name
<data_type> = <CHARACTER PROGRAM DATA>
EDG          Eye Diagram Result
UMP          Eye Mask Template File
<file_type>  = <CHARACTER PROGRAM DATA>
BIN          Binary file
CSV          CSV file
TXT          Text file

When BIN is selected for <file_type>, the Eye Diagram measurement settings and results are saved, regardless of the setting of <data_type>. When UMP is selected for <data_type>, select TXT for <file_type>.

Function
Saves the Eye Diagram measurement result data.

Note:  
The settings will not be read from the saved file if the file name is changed.

Example
To save the Eye Mask Detail measurement results of the Eye Diagram measurement, by specifying the file format, file name, and save destination:
> :SYSTem:MMEMory:DIAGram:STORe "C:\Test\example",EDG,TXT

Compatibility
Incompatible with existing models
7.12.11 Bathtub measurement

This section describes the commands for Bathtub measurement.

Figure 7.12.11-1 Bathtub Window
7.12.11.1 Measurement setting commands

- [1], [3]
- [2], [3]
- [4]
- [8]
- [9]
- [10]
- [11]
- [12]
- [13]
- [14]
- [15]
- [16]
- [17]
- [18]
- [19]
- [20]
- [21]
### Table 7.12.11.1-1  Measurement Setting Commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1]</td>
<td>Start</td>
<td>:SENSe:MEASure:BATHtub:STARt</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:MEASure:BATHtub:SLOT?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:MEASure:BATHtub:MODE?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:MEASure:BATHtub:TIMes?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:MEASure:BATHtub:INTerval?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:MEASure:BATHtub:DATA?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:MEASure:BATHtub:RESolution:MUI?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:MEASure:BATHtub:ECOunt?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:MEASure:BATHtub:RANGE?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:MEASure:BATHtub:CALRange?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:MEASure:BATHtub:ASEarch?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:MEASure:BATHtub:CURRent?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:MEASure:BATHtub:JCALculation?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:MEASure:BATHtub:J2Meas?</td>
</tr>
<tr>
<td>[17]</td>
<td>Selection of Slot(s) to be measured</td>
<td>:SENSe:MEASure:BATHtub:SELSlot</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:MEASure:BATHtub:SELSlot?</td>
</tr>
<tr>
<td>[18]</td>
<td>Selection of Slot(s) you want to view measurement results</td>
<td>:SENSe:MEASure:BATHtub:DISPlay</td>
</tr>
<tr>
<td></td>
<td>(In PAM mode only)</td>
<td>:SENSe:MEASure:BATHtub:ASEarch:PAMSet?</td>
</tr>
</tbody>
</table>
### Chapter 7  SCPI Commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
<th>Example</th>
<th>Compatibility</th>
</tr>
</thead>
<tbody>
<tr>
<td>:SENSe:MEASure:BATHtub:STARt</td>
<td>Starts Bathtub measurement.</td>
<td>&gt; :SENSe:MEASure:BATHtub:STARt</td>
<td>Incompatible with existing models</td>
</tr>
<tr>
<td>:SENSe:MEASure:BATHtub:STOP</td>
<td>Stops Bathtub measurement.</td>
<td>&gt; :SENSe:MEASure:BATHtub:STOP</td>
<td>Incompatible with existing models</td>
</tr>
<tr>
<td>:SENSe:MEASure:BATHtub:STATe?</td>
<td>Queries the measurement processing state during Bathtub measurement.</td>
<td>&gt; :SENSe:MEASure:BATHtub:STATe? &lt; 1</td>
<td>Incompatible with existing models</td>
</tr>
</tbody>
</table>

**:SENSe:MEASure:BATHtub:SLOT <numeric>,<interface>[,<unit>]**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Example</th>
<th>Compatibility</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;numeric&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
<td>1 to 6 Slots No. 1 to 6</td>
<td>&gt; :SENSe:MEASure:BATHtub:SLOT 3,1</td>
<td>Incompatible with existing models</td>
</tr>
<tr>
<td>&lt;interface&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
<td>1 to 4 Data1 to 4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>[&lt;unit&gt;] = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
<td>1 to 4 Mainframe No. 1 to 4</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Sets the Bathtub measurement target slot.
:SENSe:MEASure:BATHtub:SLOT? [<unit>]

Parameter

[<unit>] = <DECIMAL NUMERIC PROGRAM DATA>
1 to 4 Mainframe No. 1 to 4
When using two or more MT1810A units in serial connection, specify the mainframe number.
Can be omitted. Mainframe No. 1 is specified when omitted.

Response

<numeric> = <NR1 NUMERIC RESPONSE DATA>
1 to 6 Slot1 to 6
<numeric> = <NR1 NUMERIC RESPONSE DATA>
1 to 4 Data1 to 4

Function
Queries the Bathtub measurement target slot and Data Interface.

Example
> :SENSe:MEASure:BATHtub:SLOT?
< 3,1

Compatibility
Incompatible with existing models

:SENSe:MEASure:BATHtub:MODE <mode>

Parameter

(mode) = <CHARACTER PROGRAM DATA>
SINGle Executes measurement once.
REPeat Repeats measurement for the specified number of times.
UNTImed Repeats measurement until measurement is stopped.

Function
Sets the measurement processing mode during Bathtub measurement.

Example
To set the measurement processing mode to Single mode:
> :SENSe:MEASure:BATHtub:MODE SINGle

Compatibility
Incompatible with existing models

:SENSe:MEASure:BATHtub:MODE?

Response

(mode) = <CHARACTER RESPONSE DATA>
SING, REP, UNT

Function
Queries the measurement processing mode during Bathtub measurement.

Example
> :SENSe:MEASure:BATHtub:MODE?
< SING

Compatibility
Incompatible with existing models
Chapter 7  SCPI Commands

:SENSe:MEASure:BATHtub:TIMes <numeric>
Parameter  <numeric> = <DECIMAL NUMERIC PROGRAM DATA>
            2 to 1000  2 to 1000 times, in 1-time steps
Function   Sets the number of measurements during Bathtub measurement.
Example    To set the number of measurements to 100 times:
           > :SENSe:MEASure:BATHtub:TIMes 100
Compatibility Incompatible with existing models

:SENSe:MEASure:BATHtub:TIMes?
Response   <numeric> = <NR1 NUMERIC RESPONSE DATA>
Function   Queries the number of measurements during Bathtub measurement.
Example    > :SENSe:MEASure:BATHtub:TIMes?
           < 100
Compatibility Incompatible with existing models

:SENSe:MEASure:BATHtub:INTerval <numeric>
Parameter  <numeric> = <DECIMAL NUMERIC PROGRAM DATA>
            0 to 9999  0 to 9999 s, in 1-s steps
Function   Sets the measurement interval time during Bathtub measurement.
Example    To set the measurement interval time to 50 s:
           > :SENSe:MEASure:BATHtub:INTerval 50
Compatibility Incompatible with existing models

:SENSe:MEASure:BATHtub:INTerval?
Response   <numeric> = <NR1 NUMERIC RESPONSE DATA>
            0 to 9999  0 to 9999 s
Function   Queries the measurement interval time during Bathtub measurement.
Example    > :SENSe:MEASure:BATHtub:INTerval?
           < 50
 Compatibility Incompatible with existing models
**:SENSe:MEASure:BATHtub:DATA <numeric>**

**Parameter**

<numeric> = <DECIMAL NUMERIC PROGRAM DATA>

-3.500 to 3.300

-3.500 to 3.300 V, in 0.001-V steps

The range varies according to the option and input interface settings.

**Function**

Sets the measurement voltage threshold position during Bathtub measurement.

**Example**

To set the measurement voltage threshold position to 0.5 V:

> :SENSe:MEASure:BATHtub:DATA 0.5

**Compatibility**

Incompatible with existing models

**:SENSe:MEASure:BATHtub:DATA?**

**Response**

<numeric> = <NR2 NUMERIC RESPONSE DATA>

**Function**

Queries the measurement voltage threshold position during Bathtub measurement.

**Example**

> :SENSe:MEASure:BATHtub:DATA? 0.505

**Compatibility**

Incompatible with existing models

**:SENSe:MEASure:BATHtub:RESolution:MUI <numeric>**

**Parameter**

<numeric> = <DECIMAL NUMERIC PROGRAM DATA>

2 to 100

2 to 100 mUI, in 2-mUI steps

**Function**

Sets the Bathtub measurement phase resolution (mUI).

**Example**

To set the phase resolution to 10 mUI:

> :SENSe:MEASure:BATHtub:RESolution:MUI 10

**Compatibility**

Incompatible with existing models

**:SENSe:MEASure:BATHtub:RESolution:MUI?**

**Response**

<numeric> = <NR1 NUMERIC RESPONSE DATA>

**Function**

Queries the Bathtub measurement phase resolution (mUI).

**Example**

> :SENSe:MEASure:BATHtub:RESolution:MUI? < 10

**Compatibility**

Incompatible with existing models
### :SENSe:MEASure:BATHtub:ECOunt <mode>

**Parameter**

<mode> = <CHARACTER PROGRAM DATA>

- **FINE**  
  Fine (Number of error counts: 100)
- **COARse**  
  Coarse (Number of error counts: 1)

**Function**

Sets the minimum number of error counts during Bathtub measurement.

**Example**

To set the number of error counts to 100:

```
> :SENSe:MEASure:BATHtub:ECOunt FINE
```

**Compatibility**

Incompatible with existing models

### :SENSe:MEASure:BATHtub:ECOunt?

**Response**

<numeric> = <CHARACTER RESPONSE DATA>

FINE, COAR

**Function**

Queries the minimum number of error counts during Bathtub measurement.

**Example**

```
> :SENSe:MEASure:BATHtub:ECOunt?
< FINE
```

**Compatibility**

Incompatible with existing models

### :SENSe:MEASure:BATHtub:RANGe <range>

**Parameter**

<range> = <CHARACTER PROGRAM DATA>

- E_7 to E_14  
  E–7 to E–14

**Function**

Sets the lower limit value of measurement error threshold during Bathtub measurement.

**Example**

To set the lower limit value of measurement error threshold to E–12:

```
> :SENSe:MEASure:BATHtub:RANGe E_12
```

**Compatibility**

Incompatible with existing models

### :SENSe:MEASure:BATHtub:RANGe?

**Response**

<range> = <CHARACTER RESPONSE DATA>

**Function**

Queries the lower limit value of measurement error threshold during Bathtub measurement.

**Example**

```
> :SENSe:MEASure:BATHtub:RANGe?
< E_12
```

**Compatibility**

Incompatible with existing models
### :SENSe:MEASure:BATHtub:CALRange <upper>,<lower>

**Parameter**
- `<upper>` = `<CHARACTER PROGRAM DATA>`
  - E_3 to E_6
  - E–3 to E–6
- `<lower>` = `<CHARACTER PROGRAM DATA>`
  - E_7 to E_14
  - E–7 to E–14

**Function**
Sets the error threshold range during Bathtub measurement recalculation.

**Example**
To set the error threshold range during recalculation to E–5 to E–10:
> :SENSe:MEASure:BATHtub:CALRange E_5,E_10

**Compatibility**
Incompatible with existing models

### :SENSe:MEASure:BATHtub:CALRange?

**Response**
- `<upper>,<lower>` = `<CHARACTER RESPONSE DATA>`

**Function**
Queries the error threshold range during Bathtub measurement recalculation.

**Example**
> :SENSe:MEASure:BATHtub:CALRange?

`< E_5,E_10`

**Compatibility**
Incompatible with existing models

### :SENSe:MEASure:BATHtub:ASEarch

**Parameter**
- `<item>` = `<CHARACTER PROGRAM DATA>`
  - OFF
  - PHAS
  - PTHReshold
  - FPHase
  - FPTHreshold
  - PPTHreshold
  - PPFPThreshold

**Function**
Sets execution of auto search during Bathtub measurement.

**Example**
To set the phase auto search ON:
> :SENSe:MEASure:BATHtub:ASEarch PHAS

**Compatibility**
Incompatible with existing models

### :SENSe:MEASure:BATHtub:ASEarch?

**Response**
- `<item>` = `<CHARACTER RESPONSE DATA>`
  - OFF, PHAS, PTHR, FPH, FPTH, PPTH, PPFP

**Function**
Queries the execution of auto search during Bathtub measurement.

**Example**
> :SENSe:MEASure:BATHtub:ASEarch?

`< PHAS`

**Compatibility**
Incompatible with existing models
Chapter 7  SCPI Commands

:SENSe:MEASure:BATHtub:CURRent <boolean>

Parameter  
<boolean> = <BOOLEAN PROGRAM DATA>
0 or OFF  OFF: Updates at measurement termination.
1 or ON  ON: Updates every second.

Function  
Sets the graph display update period during Bathtub measurement.

Example  
To set the graph display update timing to every second:
> :SENSe:MEASure:BATHtub:CURRent 1

Compatibility  
Incompatible with existing models

:SENSe:MEASure:BATHtub:CURRent?

Response  
<boolean> = <NR1 NUMERIC RESPONSE DATA>
0, 1

Function  
Queries the graph display update period during Bathtub measurement.

Example  
> :SENSe:MEASure:BATHtub:CURRent?
< 1

Compatibility  
Incompatible with existing models

:SENSe:MEASure:BATHtub:JCALculation <setting>

Parameter  
<setting> = <DECIMAL NUMERIC PROGRAM DATA>
7 to 20  E–7 to E–20
Specified as E–n where n: 7 to 20/1 step

Function  
Sets the error rate used to calculate jitter at Bathtub measurement.

Example  
To set jitter calculation error rate to E–14:
> :SENSe:MEASure:BATHtub:JCALculation 14

Compatibility  
Incompatible with existing models

:SENSe:MEASure:BATHtub:JCALculation?

Response  
<setting> = <NR1 NUMERIC RESPONSE DATA>
7 to 20

Function  
Queries the error rate used to calculate jitter at Bathtub measurement.

Example  
> :SENSe:MEASure:BATHtub:JCALculation?
< 14

Compatibility  
Incompatible with existing models
:SENSe:MEASure:BATHtub:J2Meas <mode>

Parameter

<mode> = <CHARACTER PROGRAM DATA>

ACTual      Calculates J2 from BER measurement point.
ESTimate    Calculates J2 from Best Fit Line.

Function

Selects J2 calculation method.

Example

To set J2 calculation method to Actual:
> :SENSe:MEASure:BATHtub:J2Meas ACTual

Compatibility

Incompatible with existing models

:SENSe:MEASure:BATHtub:J2Meas?

Response

<mode> = <CHARACTER RESPONSE DATA>

ACT, EST

Function

Queries J2 calculation method.

Example

> :SENSe:MEASure:BATHtub:J2Meas?
< ACT

Compatibility

Incompatible with existing models

:SENSe:MEASure:BATHtub:SELSlot

<slot>,<interface>,<boolean>[,<unit>]

Parameter

<Slot> = <CHARACTER PROGRAM DATA>

SLOT1 to SLOT6    Slot No. 1 to 6
1 to 6 when using the MP1800A, 1 to 4 when using the MT1810A

<int>erface> = <DECIMAL NUMERIC PROGRAM DATA>

1 to 4    Data1 to 4

<boolean> = <BOOLEAN PROGRAM DATA>

ON or 1    ON
OFF or 0    OFF

[unit>] = <DECIMAL NUMERIC PROGRAM DATA>

1 to 4    Mainframe No. 1 to 4
When multiple MT1810As are connected, specify the mainframe No. to
each of them.
Can be omitted. Mainframe No. 1 is specified when omitted.

Function

Selects a Slot and Data Interface combination as the Bathtub
measurement target.

Example

To select Slot 1 and Data3 as the Bathtub measurement target:
> :SENSe:MEASure:BATHtub:SELSlot SLOT1,3,ON

Compatibility

Incompatible with existing models
Chapter 7  SCPI Commands

:SENSe:MEASure:BATHtub:SELSlot? <slot>,<interface> [,<unit>]

Parameter
- <Slot> = <CHARACTER PROGRAM DATA>
  SLOT1 to SLOT6  Slot No. 1 to 6
  1 to 6 when using the MP1800A, 1 to 4 when using the MT1810A
- <interface> = <DECIMAL NUMERIC PROGRAM DATA>
  1 to 4  Data1 to 4
- <boolean> = <BOOLEAN PROGRAM DATA>
  ON or 1  ON
  OFF or 0  OFF
- [<unit>] = <DECIMAL NUMERIC PROGRAM DATA>
  1 to 4  Mainframe No. 1 to 4

When multiple MT1810As are connected, specify the mainframe No. to each of them.
Can be omitted. Mainframe No. 1 is specified when omitted.

Response
- <boolean> = <NR1 NUMERIC RESPONSE DATA>
  1 ON
  0 OFF

Function
Queries whether the specified Slot and Data Interface combination is subject to the Bathtub measurement.

Example
To query whether Slot 1 and Data3 are subject to measurement:
> :SENSe:MEASure:BATHtub:SELSlot? SLOT1,3
< 1

Compatibility
Incompatible with existing models

:SENSe:MEASure:BATHtub:DISPlay <slot>,<interface> [,<unit>]

Parameter
- <slot> = <CHARACTER PROGRAM DATA>
  SLOT1 to SLOT6  Slot No. 1 to 6
  1 to 6 when using the MP1800A, 1 to 4 when using the MT1810A
- <interface> = <DECIMAL NUMERIC PROGRAM DATA>
  1 to 4  Data1 to 4
- [<unit>] = <DECIMAL NUMERIC PROGRAM DATA>
  1 to 4  Mainframe No. 1 to 4

When multiple MT1810As are connected, specify the mainframe No. to each of them.
Can be omitted. Mainframe No. 1 is specified when omitted.

Function
Selects a Slot and Data Interface combination you want to view Bathtub measurement results.

Example
To view Bathtub measurement results of Slot3 and Data1:
> :SENSe:MEASure:BATHtub:DISPlay Slot3,1

Compatibility
Incompatible with existing models
### :SENSe:MEASure:BATHtub:SLASet

<table>
<thead>
<tr>
<th>Parameter</th>
<th>None</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function</td>
<td>Sets all Slot and Data Interface combinations as the Bathtub measurement objects.</td>
</tr>
<tr>
<td>Example</td>
<td>&gt; :SENSe:MEASure:BATHtub:SLASet</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Incompatible with existing models</td>
</tr>
</tbody>
</table>

### :SENSe:MEASure:BATHtub:SLAReset

<table>
<thead>
<tr>
<th>Parameter</th>
<th>None</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function</td>
<td>Resets all Slot and Data Interface combinations set as the Bathtub measurement objects.</td>
</tr>
<tr>
<td>Example</td>
<td>&gt; :SENSe:MEASure:BATHtub:SLAReset</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Incompatible with existing models</td>
</tr>
</tbody>
</table>

### :SENSe:MEASure:BATHtub:ASEarch:PAMSet

<table>
<thead>
<tr>
<th>&lt;slot&gt;,&lt;interface&gt;,&lt;threshold&gt;[,&lt;unit&gt;]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter:</td>
</tr>
<tr>
<td>&lt;slot&gt; = &lt;CHARACTER PROGRAM DATA&gt;</td>
</tr>
<tr>
<td>1 to 6</td>
</tr>
<tr>
<td>Slots No. 1 to 6</td>
</tr>
<tr>
<td>1 to 6 when using the MP1800A, 1 to 4 when using the MT1810A</td>
</tr>
<tr>
<td>&lt;interface&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
</tr>
<tr>
<td>1 to 4</td>
</tr>
<tr>
<td>Data1 to 4</td>
</tr>
<tr>
<td>&lt;threshold&gt; = &lt;CHARACTER PROGRAM DATA&gt;</td>
</tr>
<tr>
<td>TOP</td>
</tr>
<tr>
<td>Searches the Top threshold</td>
</tr>
<tr>
<td>MIDDle</td>
</tr>
<tr>
<td>Searches the Middle threshold</td>
</tr>
<tr>
<td>BOTTtom</td>
</tr>
<tr>
<td>Searches the Bottom threshold</td>
</tr>
<tr>
<td>[&lt;unit&gt;] = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
</tr>
<tr>
<td>1 to 4</td>
</tr>
<tr>
<td>Mainframe No. 1 to 4</td>
</tr>
<tr>
<td>When using two or more MT1810A units in serial connection, specify the mainframe number.</td>
</tr>
<tr>
<td>Can be omitted. Mainframe No. 1 is specified when omitted.</td>
</tr>
<tr>
<td>Function:</td>
</tr>
<tr>
<td>Selects search target threshold for running PAM Auto Search in the Bathtub measurement.</td>
</tr>
<tr>
<td>Example:</td>
</tr>
<tr>
<td>To set Data 3 of Slot No.1 for Top threshold search:</td>
</tr>
<tr>
<td>&gt; :SENSe:MEASure:BATHtub:ASEarch:PAMSet SLOT1,3,TOP</td>
</tr>
<tr>
<td>Compatibility:</td>
</tr>
<tr>
<td>Incompatible with existing models</td>
</tr>
</tbody>
</table>
Chapter 7  SCPI Commands

:SENSe:MEASure:BATHtub:ASEarch:PAMSet? <slot>,<interface>[,<unit>]

Parameter

- `<slot>` = `<CHARACTER PROGRAM DATA>`
  1 to 6  Slots No. 1 to 6
  1 to 6 when using the MP1800A, 1 to 4 when using the MT1810A
- `<interface>` = `<DECIMAL NUMERIC PROGRAM DATA>`
  1 to 4  Data1 to 4
- `[<unit>]` = `<DECIMAL NUMERIC PROGRAM DATA>`
  1 to 4  Mainframe No. 1 to 4
When using two or more MT1810A units in serial connection, specify the mainframe number.
Can be omitted. Mainframe No. 1 is specified when omitted.

Response

- `<threshold>` = `<CHARACTER RESPONSE DATA>`
  TOP, MIDD, BOTT

Function

Queries search target threshold for running PAM Auto Search in the Bathtub measurement.

Example

To query the search target threshold of Data 3 of Slot No.1:
```
> :SENSe:MEASure:BATHtub:ASEarch:PAMSet? SLOT1,3
< 1
```

Compatibility

Incompatible with existing models
### 7.12.11.2 Graph display setting commands

![Graph Display Field](image)

#### Figure 7.12.11.2-1  Graph Display Field

#### Table 7.12.11.2-1  Graph Display Setting Commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>:DIPlay:RESult:BATHtub:ITEM?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:DIPlay:RESult:BATHtub:MNUMber?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:DIPlay:RESult:BATHtub:SCALe:HORizontal?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:DIPlay:RESult:BATHtub:SCALe:VERTical?</td>
</tr>
</tbody>
</table>
### :DISPlay:RESULT:BATHtub:ITEM <mode>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;mode&gt; = &lt;CHARACTER PROGRAM DATA&gt;</td>
<td>Sets the item to be displayed in the graph during Bathtub measurement.</td>
</tr>
<tr>
<td>PBER</td>
<td>Phase vs BER (Y axis Log)</td>
</tr>
<tr>
<td>PLNBER</td>
<td>Phase vs BER (Y axis Log (−Ln))</td>
</tr>
<tr>
<td>HISTogram</td>
<td>Histogram</td>
</tr>
</tbody>
</table>

**Example**

To set the display item to Histogram:

```
> :DISPlay:RESULT:BATHtub:ITEM HISTogram
```

**Compatibility**

Incompatible with existing models

### :DISPlay:RESULT:BATHtub:ITEM?

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;mode&gt; = &lt;CHARACTER RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>PBER, PLNBER, HIST</td>
<td>Queries the item to be displayed in the graph during Bathtub measurement.</td>
</tr>
</tbody>
</table>

**Example**

```
> :DISPlay:RESULT:BATHtub:ITEM?
HIST
```

**Compatibility**

Incompatible with existing models

### :DISPlay:RESULT:BATHtub:MNUMber <numeric>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;numeric&gt; = &lt;DECIMAL PROGRAM DATA&gt;</td>
<td>Selects the measurement number for which the Bathtub measurement result is displayed in a graph.</td>
</tr>
<tr>
<td>1 to 1000</td>
<td>Measurement number 1 to 1000 (number of measurements during repeat measurement)</td>
</tr>
</tbody>
</table>

**Example**

To select the measurement number 100 to display the graph:

```
> :DISPlay:RESULT:BATHtub:MNUMber 100
```

**Compatibility**

Incompatible with existing models

### :DISPlay:RESULT:BATHtub:MNUMber?

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;numeric&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;numeric&gt;</td>
<td>Queries the measurement number for which the Bathtub measurement result is displayed in a graph.</td>
</tr>
</tbody>
</table>

**Example**

```
> :DISPlay:RESULT:BATHtub:MNUMber?
100
```

**Compatibility**

Incompatible with existing models
:DISPlay:RESult:BATHtub:SCALe:ASCale

Function
Automatically adjusts the graph display during Bathtub measurement.

Example
> :DISPlay:RESult:BATHtub:SCALe:ASCale

Compatibility
Incompatible with existing models

:DISPlay:RESult:BATHtub:SCALe:HORizontal <max>[,<step>]

Parameter
<max>,<step> = <DECIMAL NUMERIC PROGRAM DATA>

<table>
<thead>
<tr>
<th>Form</th>
<th>&lt;max&gt;</th>
<th>&lt;step&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Phase type</td>
<td>–900 to 1000 (mUI), in 100-mUI steps</td>
<td>10 to 200 (mUI), in 10-mUI steps</td>
</tr>
<tr>
<td>Histogram type</td>
<td>16,32,64,128,256</td>
<td>Omitted</td>
</tr>
</tbody>
</table>

Function
Sets the horizontal scale of the graph display during Bathtub measurement. In case of a histogram, <step> is omitted.

Example
To set the scale to maximum 800 mUI, in 100-mUI steps:
> :DISPlay:RESult:BATHtub:SCALe:HORizontal 800,100

Compatibility
Incompatible with existing models

:DISPlay:RESult:BATHtub:SCALe:HORizontal?

Response
<max>,<step> = <NR2 NUMERIC RESPONSE DATA>
Same as the :DISPlay:RESult:BATHtub:SCALe:HORizontal command parameter.

Function
Queries the horizontal scale of the graph display during Bathtub measurement.

Example
> :DISPlay:RESult:BATHtub:SCALe:HORizontal?
< 800,100

Compatibility
Incompatible with existing models
**Chapter 7  SCPI Commands**

:DISPlay:RESult:BATHtub:SCALe:VERTical <max>,<min>

Parameter  

<max>,<min> = <DECIMAL NUMERIC PROGRAM DATA>

<table>
<thead>
<tr>
<th>Form</th>
<th>&lt;max&gt;</th>
<th>&lt;min&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Histogram Type</td>
<td>6 to 15 (E + 6 to E + 15)</td>
<td>Input 0</td>
</tr>
<tr>
<td>Error Rate Type</td>
<td>2 to 7 (E–2 to E–7)</td>
<td>8 to 15 (E–8 to E–15)</td>
</tr>
</tbody>
</table>

**Note:**

In case of the Error Rate type, the difference between the maximum and minimum value must be at least 6, and the value must be an even number.

**Function**

Sets the vertical scale of the graph display during Bathtub measurement.

**Example**

To set the scale during Error Rate type display to maximum E–4 and minimum E–10:

> :DISPlay:RESult:BATHtub:SCALe:VERTical 4,10

**Compatibility**

Incompatible with existing models

:DISPlay:RESult:BATHtub:SCALe:VERTical?

Response  

<min>,<max> = <NR2 NUMERIC RESPONSE DATA>

Same as the :DISPlay:RESult:BATHtub:SCALe:VERTical command parameter.

**Function**

Queries the vertical scale of the graph display during Bathtub measurement.

**Example**

> :DISPlay:RESult:BATHtub:SCALe:VERTical?

< 4,10

**Compatibility**

Incompatible with existing models
7.12.11.3 Measurement result and status query commands

Table 7.12.11.3-1 Measurement Result and Status Query Commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>:SENSe:MEASure:BATHtub:DTIME?</td>
</tr>
</tbody>
</table>

**:SENSe:MEASure:BATHtub:DTIME <type>**

Parameter

- `<type> = <CHARACTER PROGRAM DATA>`
  - DTIMe: Displays the current date and time.
  - STARt: Displays the measurement start time.
  - ELAPsed: Displays the elapsed time based on the measurement period.

Function

Selects the Bathtub measurement time display type.

Example

To set the Bathtub measurement time display type to measurement start time (Start Time):

```
> :SENSe:MEASure:BATHtub:DTIME STARt
```

Compatibility

Incompatible with existing models

**:SENSe:MEASure:BATHtub:DTIME?**

Response

- `<type> = <CHARACTER RESPONSE DATA>`
  - DTIM, STAR, ELAP

Function

Queries the Bathtub measurement time display type.

Example

```
> :SENSe:MEASure:BATHtub:DTIME?
< STAR
```

Compatibility

Incompatible with existing models
Chapter 7  SCPI Commands

:SENSe:MEASure:BATHtub:TIME:DTIMe?

Response

Response

<year> = <NR1 NUMERIC RESPONSE DATA>
2000 to 2036  Year 2000 to 2036

Response

<month> = <NR1 NUMERIC RESPONSE DATA>
1 to 12  January to December

Response

<day> = <NR1 NUMERIC RESPONSE DATA>
1 to 31  1st to 31st

Response

<hour> = <NR1 NUMERIC RESPONSE DATA>
0 to 23  0 to 23 hours

Response

<minute> = <NR1 NUMERIC RESPONSE DATA>
0 to 59  0 to 59 minutes

Response

<second> = <NR1 NUMERIC RESPONSE DATA>
0 to 59  0 to 59 seconds

Function

Queries the current date and time during the Bathtub measurement.

Example

> :SENSe:MEASure:BATHtub:TIME:DTIMe?
< 2012,12,31,23,59,59

Compatibility

Incompatible with existing models

:SENSe:MEASure:BATHtub:TIME:STARt?

Response

Response

<year> = <NR1 NUMERIC RESPONSE DATA>
0, 2000 to 2036  Year 2000 to 2036

Response

<month> = <NR1 NUMERIC RESPONSE DATA>
0, 1 to 12  January to December

Response

<day> = <NR1 NUMERIC RESPONSE DATA>
0, 1 to 31  1st to 31st

Response

<hour> = <NR1 NUMERIC RESPONSE DATA>
0 to 23  0 to 23 hours

Response

<minute> = <NR1 NUMERIC RESPONSE DATA>
0 to 59  0 to 59 minutes

Response

<second> = <NR1 NUMERIC RESPONSE DATA>
0 to 59  0 to 59 seconds

“0,0,0,0,0,0” is returned if there is no measurement start time data.

Function

Queries the current date and time during the Bathtub measurement.

Example

> :SENSe:MEASure:BATHtub:TIME:STARt?
< 2012,12,31,23,59,59

Compatibility

Incompatible with existing models
:SENSe:MEASure:BATHtub:TIME:ELAPsed?

Response  
<day> = <NR1 NUMERIC RESPONSE DATA>
0 to 99   0 to 99 days
<hour> = <NR1 NUMERIC RESPONSE DATA>
0 to 23   0 to 23 hours
<minute> = <NR1 NUMERIC RESPONSE DATA>
0 to 59   0 to 59 minutes
<second> = <NR1 NUMERIC RESPONSE DATA>
0 to 59   0 to 59 seconds

“0,0,0,0” is returned if there is no measurement start time data.

Function  
Queries the measurement elapsed time during Bathtub measurement.

Example  
> :SENSe:MEASure:BATHtub:TIME:ELAPsed?
< 31,23,59,59

Compatibility  
Incompatible with existing models

:CALCulate:DATA:BATHtub? <string>

Parameter  
<string> = <STRING PROGRAM DATA>

For the contents of measurement data <result1>, see Table 7.12.11.3-2.
For the contents of measurement data <result2>, see Table 7.12.11.3-3.
For the contents of measurement data <result3>, see Table 7.12.11.3-4.

Table 7.12.11.3-2  Phase vs. BER Measurement Result <result1>

<table>
<thead>
<tr>
<th>Items</th>
<th>&lt;Result1&gt;</th>
<th>Response Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>Phase vs BER (Phase/Error Rate)</td>
<td>Number of measurements 1 to 1000</td>
<td>&quot;BATH:1&quot;...&quot;BATH:1000&quot;</td>
</tr>
<tr>
<td>Histogram (Phase/Error Count)</td>
<td>&quot;HIST:1&quot;...&quot;HIST:1000&quot;</td>
<td>Form2</td>
</tr>
</tbody>
</table>

The measurement result for each Phase is delimited by a comma (,) and returned.

Table 7.12.11.3-3  Phase vs. BER Immediate Data <result2>

<table>
<thead>
<tr>
<th>Items</th>
<th>&lt;Result2&gt;</th>
<th>Response Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>Optimum BER</td>
<td>Number of measurements 1 to 1000</td>
<td>&quot;OPTBer:1&quot;...&quot;OPTBer:1000&quot;</td>
</tr>
<tr>
<td>Optimum Phase</td>
<td>&quot;OPTPh:1&quot;...&quot;OPTPh:1000&quot;</td>
<td>Form4</td>
</tr>
<tr>
<td>Total Jitter</td>
<td>&quot;TJ:1&quot;...&quot;TJ:1000&quot;</td>
<td>Form4</td>
</tr>
<tr>
<td>Deterministic Jitter</td>
<td>&quot;DJ:1&quot;...&quot;DJ:1000&quot;</td>
<td>Form4</td>
</tr>
<tr>
<td>Random Jitter</td>
<td>&quot;RJ:1&quot;...&quot;RJ:1000&quot;</td>
<td>Form4</td>
</tr>
<tr>
<td>J2</td>
<td>&quot;J2:1&quot;...&quot;J2:1000&quot;</td>
<td>Form4</td>
</tr>
<tr>
<td>J9</td>
<td>&quot;J9:1&quot;...&quot;J9:1000&quot;</td>
<td>Form4</td>
</tr>
</tbody>
</table>
Table 7.12.11.3-4  Phase vs. BER Measurement Statistic Data <result3>

<table>
<thead>
<tr>
<th>Items</th>
<th>&lt;Result3&gt;</th>
<th>Response Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Data</td>
<td>&quot;TOTal&quot;</td>
<td>Form5</td>
</tr>
<tr>
<td>Valid Data</td>
<td>&quot;VALid&quot;</td>
<td>Form5</td>
</tr>
<tr>
<td>Total Jitter Mean</td>
<td>&quot;TJMean&quot;</td>
<td>Form4</td>
</tr>
<tr>
<td>Deterministic Jitter Mean</td>
<td>&quot;DJMean&quot;</td>
<td>Form4</td>
</tr>
<tr>
<td>Random Jitter Mean</td>
<td>&quot;RJMean&quot;</td>
<td>Form4</td>
</tr>
<tr>
<td>J2 Mean</td>
<td>&quot;J2Mean&quot;</td>
<td>Form4</td>
</tr>
<tr>
<td>J9 Mean</td>
<td>&quot;J9Mean&quot;</td>
<td>Form4</td>
</tr>
</tbody>
</table>

Response  
\(<\text{string}> = \langle\text{STRING RESPONSE DATA}\rangle\)

Table 7.12.11.3-5  Response Format

<table>
<thead>
<tr>
<th>Form</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Form1</td>
<td>&quot;XXXX.XX, \n\nY.YYYYE-YYY&quot;</td>
<td>XXXX.XX: Phase (mUI or PS) \nY.YYYYE-YYY: Error Rate \n&quot;------,------&quot; When no data corresponds to the query.</td>
</tr>
<tr>
<td>Form2</td>
<td>&quot;XXXX.XX, \n\nY.YYYYE-YYY&quot;</td>
<td>XXXX.XX: Phase (mUI or PS) \nY.YYYYE-YYY: Error Count \n&quot;------,------&quot; When no data corresponds to the query.</td>
</tr>
<tr>
<td>Form3</td>
<td>&quot;0.XXXXXXE-XXX&quot;</td>
<td>When 0.0000E–016 to 1.0000E000 \n&quot;--------&quot; When no data corresponds to the query.</td>
</tr>
<tr>
<td>Form4</td>
<td>&quot;XXXX.XX&quot;</td>
<td>When –9999.99 to 9999.99 \n&quot;--------&quot; When no data corresponds to the query.</td>
</tr>
<tr>
<td>Form5</td>
<td>&quot;XXXX&quot;</td>
<td>When 0 to 9999 \n&quot;----&quot; When no data corresponds to the query.</td>
</tr>
</tbody>
</table>

Function  
Queries the Phase vs. BER measurement result.

Example  
To query the Phase vs. BER the measurement result for the 10th time:

```
> :CALCulate:DATA:BATHtub? "BATH:10"
< "0,1.2345E-003","20,1.2345E-004","40,1.2345E-005","60,1.2345E-006"
```

Compatibility  
Incompatible with existing models
### :CALCulate:DATA:BATHtub:STATus?

<table>
<thead>
<tr>
<th>Response</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;string&gt; = &lt;STRING RESPONSE DATA&gt;</td>
<td></td>
</tr>
<tr>
<td>&quot;&quot;</td>
<td>When no alarm exists.</td>
</tr>
<tr>
<td>&quot;Sync Loss&quot;</td>
<td>Sync Loss</td>
</tr>
<tr>
<td>&quot;Clock Loss&quot;</td>
<td>Clock Loss</td>
</tr>
<tr>
<td>&quot;Out of range&quot;</td>
<td>Out of range</td>
</tr>
<tr>
<td>&quot;Illegal Error&quot;</td>
<td>Illegal Error</td>
</tr>
<tr>
<td>&quot;Meas. Threshold exceeded&quot;</td>
<td>Number of errors exceeds the target rate</td>
</tr>
</tbody>
</table>

**Note:**
- Displays any one of the alarms.
- Illegal Error > Clock Loss > Sync Loss > Out or range

**Function**
Queries the Q measurement status.

**Example**
```
> :CALCulate:DATA:BATHtub:STATus?
< "Sync Loss"
```
Chapter 7  SCPI Commands

7.12.11.4 File menu setting commands

Table 7.12.11.4-1  File Menu Setting Commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
</table>

**:SYStem:MMEMory:BATHtub:RECall <file_name>**

Parameter

- `<file_name>` = <STRING PROGRAM DATA>
- `"<drv>:\[<dir>]\<file>"`
- `<drv>` = C, D, E, F
- `<dir>` = `<dir1>/<dir2>...(Omitted when root directory)`
- `<file>` = File name

Function

Opens the Bathtub measurement result data.

Example

> :SYStem:MMEMory:BATHtub:RECall "C:\Test\example"

Compatibility

Incompatible with existing models

**:SYStem:MMEMory:BATHtub:STORe**

Parameter

- `<file_name>` = <STRING PROGRAM DATA>
- `"<drv>:\[<dir>]\<file>"`
- `<drv>` = C, D, E, F
- `<dir>` = `<dir1>/<dir2>...(Omitted when root directory)`
- `<file>` = File name
- `<data_type>` = <CHARACTER RESPONSE DATA>
- PSB Phase vs BER Result
- `<file_type>` = <CHARACTER RESPONSE DATA>
- BIN Binary File
- CSV CSV File
- TXT Text File

Function

Saves the Bathtub measurement result data.

*Note:*

The settings will not be read from the saved file if the file name is changed.

Example

To save the Bathtub measurement result data by specifying the save destination (C:\Test), file name (example), and file format (CSV):

> :SYStem:MMEMory:BATHtub:STORe "C:\Test\example",PSB,CSV

Compatibility

Incompatible with existing models
7.12.12 Auto Search

This chapter describes the commands that are used after Auto Search is set by the `SYSTem:CFUNction` command.

**Figure 7.12.12-1  Auto Search Dialog Box**

**Figure 7.12.12-2  Auto Search Dialog Box (PAM Mode)**

**Table 7.12.12-1  Auto Search Setting Commands**

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td><code>:SENSe:MEASure:ASEarch:SMoDe?</code></td>
</tr>
<tr>
<td></td>
<td></td>
<td><code>:SENSe:MEASure:ASEarch:MODE?</code></td>
</tr>
<tr>
<td>[8]</td>
<td>Slot Select</td>
<td><code>:SENSe:MEASure:ASEarch:SELSlot</code></td>
</tr>
<tr>
<td></td>
<td></td>
<td><code>:SENSe:MEASure:ASEarch:SELSlot?</code></td>
</tr>
</tbody>
</table>
Table 7.12.12-1  Auto Search Setting Commands (Cont’d)

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>[9]</td>
<td>No label (Query for slot state)</td>
<td>:SENSe:MEASure:ASEarch:SLOT?</td>
</tr>
<tr>
<td>[10]</td>
<td>No label (Query for auto search results)</td>
<td>:CALCulate:DATA:ASEarch?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:MEASure:ASEarch:PAMSet?</td>
</tr>
</tbody>
</table>

:**SENSe:MEASure:ASEarch:SMODe <mode>**

Parameter  

\(<mode> = \langle CHARACTER PROGRAM DATA >\>

FINE  Fine mode
COARse Coarse mode
PAMFine PAM4 Fine mode
PAMCoarse PAM4 Coarse mode

Function  

Sets the Auto Search mode.

Example  

To set the Fine mode:

> :SENSe:MEASure:ASEarch:SMODe FINE

Compatibility  

Incompatible with existing models

:**SENSe:MEASure:ASEarch:SMODe?**

Response  

\(<mode> = \langle CHARACTER RESPONSE DATA >\>

FINE, COAR, PAMF, PAMC

Function  

Queries the Auto Search mode.

Example  

> :SENSe:MEASure:ASEarch:SMODe?

< FINE

Compatibility  

Incompatible with existing models

:**SENSe:MEASure:ASEarch:MODE <mode>**

Parameter  

\(<mode> = \langle CHARACTER PROGRAM DATA >\>

PTHReshold Phase & Threshold
THRReshold Threshold
PHASE Phase

Function  

Sets the execution item of Auto Search.

Example  

To set the execution item of Auto Search to Phase & Threshold:

> :SENSe:MEASure:ASEarch:MODE PTHReshold

Compatibility  

Compatible with the MP1632C.
**:SENSe:MEASure:ASEarch:MODE?**

**Response**

\[<\text{mode}> = \text{<CHARACTER RESPONSE DATA>}\]

- **PTHR**: Phase & Threshold
- **THR**: Threshold
- **PHAS**: Phase

**Function**

Queries the execution item of Auto Search.

**Example**

```
> :SENSe:MEASure:ASEarch:MODE?
< PTHR
```

**Compatibility**

Compatible with the MP1632C.

---

**:SENSe:MEASure:ASEarch:STARt**

**Function**

Starts the Auto Search function.

**Example**

```
> :SENSe:MEASure:ASEarch:STARt
```

**Compatibility**

Compatible with the MP1632C.

---

**:SENSe:MEASure:ASEarch:STOP**

**Function**

Stops the Auto Search function.

**Example**

```
> :SENSe:MEASure:ASEarch:STOP
```

**Compatibility**

Compatible with the MP1632C.

---

**:SENSe:MEASure:ASEarch:STATe?**

**Response**

\[<\text{boolean}> = \text{<NR1 NUMERIC RESPONSE DATA>}\]

- 1: Started
- 0: Stopped
- -1: Auto Search failure (when any slot has failed)

**Function**

Queries the Auto Search function state.

**Example**

```
> :SENSe:MEASure:ASEarch:STATe?
< 1
```

**Compatibility**

Compatible with the MP1632C.

---

**:SENSe:MEASure:ASEarch:SLASet**

**Function**

Sets Auto Search execution for all slots to ON.

**Example**

```
> :SENSe:MEASure:ASEarch:SLASet
```

**Compatibility**

Incompatible with existing models
**Chapter 7  SCPI Commands**

:SENSe:MEASure:ASEarch:SLAReset

**Function**
Sets Auto Search execution for all slots to OFF.

**Example**
> :SENSe:MEASure:ASEarch:SLAReset

**Compatibility**
Incompatible with existing models

:SENSe:MEASure:ASEarch:SELSlot
<slot>,<interface>,<boolean>[,<unit>]

**Parameter**
- `<slot>` = `<CHARACTER PROGRAM DATA>`
  - SLOT1 to SLOT6  Slots No. 1 to 6
  - 1 to 6 when using the MP1800A, 1 to 4 when using the MT1810A
- `<interface>` = `<DECIMAL NUMERIC PROGRAM DATA>`
  - 1 to 4  Data1 to 4
- `<boolean>` = `<BOOLEAN PROGRAM DATA>`
  - ON, 1  Auto Search ON
  - OFF, 0  Auto Search OFF
- `[<unit>]` = `<DECIMAL NUMERIC PROGRAM DATA>`
  - 1 to 4  Mainframe No. 1 to 4
  - When using two or more MT1810A units in serial connection, specify the mainframe number.
  - Can be omitted. Mainframe No. 1 is specified when omitted.

**Function**
Selects the slot for execution of Auto Search.

**Example**
To set Auto Search for Slot No.1, Data 3 to ON:
> :SENSe:MEASure:ASEarch:SELSlot SLOT1,3,ON

**Compatibility**
Incompatible with existing models (Parameter varies by model.)
:SENSe:MEASure:ASEarch:SELSlot? <slot>,<interface>[[,<unit>]]

Parameter

- `<slot>` = <CHARACTER PROGRAM DATA>
  - SLOT1 to SLOT6: Slots No. 1 to 6
  - 1 to 6 when using the MP1800A, 1 to 4 when using the MT1810A
- `<interface>` = <DECIMAL NUMERIC PROGRAM DATA>
  - 1 to 4: Data1 to 4
- `[<unit>]` = <DECIMAL NUMERIC PROGRAM DATA>
  - 1 to 4: Mainframe No. 1 to 4

When using two or more MT1810A units in serial connection, specify the mainframe number.
Can be omitted. Mainframe No. 1 is specified when omitted.

Response

- `<boolean>` = <NR1 NUMERIC RESPONSE DATA>
  - 1: ON
  - 0: OFF

Function

Queries the ON/OFF state of Auto Search for each slot.

Example

To query the ON/OFF state of Auto Search for Slot No.1, Data 3:

```
> :SENSe:MEASure:ASEarch:SELSlot? SLOT1,3
< 1
```

Compatibility

Incompatible with existing models (Parameter varies by model.)

---

:SENSe:MEASure:ASEarch:SLOT? <slot>[,[<unit>]]

Parameter

- `<slot>` = <CHARACTER PROGRAM DATA>
  - SLOT1 to SLOT6: Slots No. 1 to 6
  - 1 to 6 when using the MP1800A, 1 to 4 when using the MT1810A
- `[<unit>]` = <DECIMAL NUMERIC PROGRAM DATA>
  - 1 to 4: Mainframe No. 1 to 4

When using two or more MT1810A units in serial connection, specify the mainframe number.
Can be omitted. Mainframe No. 1 is specified when omitted.

Response

- `<interface>` = <NR1 NUMERIC RESPONSE DATA>
  - 1 to 4: Data1 to 4

Note:

The above response returns all slots for which Auto Search can be executed by delimiting with commas (,).

Function

Queries the data interface of slot 1 for which Auto Search can be executed.

Example

```
> :SENSe:MEASure:ASEarch:SLOT? SLOT1
< 1,2,3,4
```

Compatibility

Incompatible with existing models
Chapter 7  SCPI Commands

:CALCulate:DATA:ASEarch? <slot>,<interface>[,<unit>]

Parameter

- `<slot>` = `<CHARACTER PROGRAM DATA>`
  - SLOT1 to SLOT6 Slots No. 1 to 6
  - 1 to 6 when using the MP1800A, 1 to 4 when using the MT1810A
- `<interface>` = `<DECIMAL NUMERIC PROGRAM DATA>`
  - 1 to 4 Data1 to 4
- `[<unit>]` = `<DECIMAL NUMERIC PROGRAM DATA>`
  - 1 to 4 Mainframe No. 1 to 4
  - When using two or more MT1810A units in serial connection, specify the mainframe number.
  - Can be omitted. Mainframe No. 1 is specified when omitted.

Response

- `<data>,<xdata>,<delay>,<numeric1>,<numeric2>,<numeric3>,<numeric4>`
  - `<data>,<xdata>,<delay>` = `<STRING RESPONSE DATA>`
    - "Done" Normal termination
    - "Fail" Abnormal termination
    - "----" Not executed
  - `<numeric1>,<numeric2>` = `<NR2 NUMERIC RESPONSE DATA>`
    - –3.500 to 3.300 –3.500 to 3.300 V/Step 0.001 V
    - "----" Not executed
    - "----" Abnormal termination
  - `<numeric3>` = `<NR1 NUMERIC RESPONSE DATA>`
    - –1000 to 1000 –1000 to 1000 mUI/Step 1 mUI
    - "----" Not executed
    - "----" Abnormal termination
  - `<numeric4>` = `<NR2 NUMERIC RESPONSE DATA>`
    - –xxx.xx to xxx.xx ps (This value is converted from mUI units.)
    - "----" Not executed
    - "----" Abnormal termination

Function

Queries the Auto Search execution result.

Example

To query the Auto Search execution result of slot 1, Data 1:

```plaintext
> :CALCulate:DATA:ASEarch? SLOT1,1
< "Done", "Done", "Done", 1.000, 1.100, -500, 50.00
< "Done", "Done", "----", 1.000, 1.100, 0, 0.00 (When not executed in the Phase direction)
```

Compatibility

Incompatible with existing models. (Parameter varies by model.)
### :SENSe:MEASure:ASEarch:PAMSet

**Parameter**

- `<slot>` = <CHARACTER PROGRAM DATA>
- `SLOT1` to `SLOT6` Slots No. 1 to 6
- 1 to 6 when using the MP1800A, 1 to 4 when using the MT1810A
- `<interface>` = <DECIMAL NUMERIC PROGRAM DATA>
- 1 to 4 Data1 to 4
- `<threshold>` = <CHARACTER PROGRAM DATA>
- `TOP` Top threshold
- `MIDDle` Middle threshold
- `BOTTom` Bottom threshold
- `[<unit>]` = <DECIMAL NUMERIC PROGRAM DATA>
- 1 to 4 Mainframe No. 1 to 4

When using two or more MT1810A units in serial connection, specify the mainframe number.

Can be omitted. Mainframe No. 1 is specified when omitted.

**Function**
Sets search target threshold for running PAM Auto Search.

**Example**
To set Data 3 of Slot No. 1 as Bottom threshold search target:

```plaintext
> :SENSe:MEASure:ASEarch:PAMSet SLOT1,3,BOTTom
```

**Compatibility**
Incompatible with existing models.

### :SENSe:MEASure:ASEarch:PAMSet?

**Parameter**

- `<slot>` = <CHARACTER PROGRAM DATA>
- `SLOT1` to `SLOT6` Slots No. 1 to 6
- 1 to 6 when using the MP1800A, 1 to 4 when using the MT1810A
- `<interface>` = <DECIMAL NUMERIC PROGRAM DATA>
- 1 to 4 Data1 to 4
- `[<unit>]` = <DECIMAL NUMERIC PROGRAM DATA>
- 1 to 4 Mainframe No. 1 to 4

When using two or more MT1810A units in serial connection, specify the mainframe number.

Can be omitted. Mainframe No. 1 is specified when omitted.

**Response**

- `<threshold>` = <CHARACTER RESPONSE DATA>
- `TOP`, `MIDD`, `BOTT`

**Function**
Queries search target threshold of the slot for running PAM Auto Search.

**Example**
To query search target threshold of Data 3 of Slot No. 1:

```plaintext
> :SENSe:MEASure:ASEarch:PAMSet? SLOT1,3
< BOTT
```

**Compatibility**
Incompatible with existing models.
7.12.13 Auto Adjust

This chapter describes the commands that are used after Auto Adjust is set by the `:SYSTEM:CFUNCTION` command.

![Auto Adjust Dialog Box](image)

Figure 7.12.13-1 Auto Adjust Dialog Box

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1]</td>
<td>Item</td>
<td>:SENSe:MEASure:AADJust32:ITEM</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:MEASure:AADJust32:ITEM?</td>
</tr>
<tr>
<td>[4]</td>
<td>No label (Query for function state)</td>
<td>:SENSe:MEASure:AADJust32:STATe?</td>
</tr>
<tr>
<td>[5]</td>
<td>Slot Select</td>
<td>:SENSe:MEASure:AADJust32:SELSlot</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:MEASure:AADJust32:SELSlot?</td>
</tr>
<tr>
<td>[6]</td>
<td>No label (Query for slot state)</td>
<td>:SENSe:MEASure:AADJust32:SLOT?</td>
</tr>
<tr>
<td>[8]</td>
<td>Reset All</td>
<td>:SENSe:MEASure:AADJust32:SLAReset</td>
</tr>
</tbody>
</table>
:SENSe:MEASure:AADJust32:ITEM <item>

Parameter
<item> = <CHARACTER PROGRAM DATA>
PThReshold Phase & Threshold: Phase & Threshold: Auto tracking in the voltage and phase directions
THReshold Threshold: Auto tracking in the voltage direction
PHASe Phase: Auto tracking in the phase direction

Function
Sets the execution item of Auto Adjust.

Example
> :SENSe:MEASure:AADJust32:ITEM PTHReshold

Compatibility
Incompatible with existing models

:SENSe:MEASure:AADJust32:ITEM?

Response
<item> = <CHARACTER RESPONSE DATA>
PTH Phase & Threshold
THR Threshold
PHAS Phase

Function
Queries the execution item of Auto Adjust.

Example
> :SENSe:MEASure:AADJust32:ITEM?
< PTHR

Compatibility
Incompatible with existing models

:SENSe:MEASure:AADJust32:STARt

Function
Starts the Auto Adjust function.

Example
> :SENSe:MEASure:AADJust32:STARt

Compatibility
Incompatible with existing models

:SENSe:MEASure:AADJust32:STOP

Function
Stops the Auto Adjust function.

Example
> :SENSe:MEASure:AADJust32:STOP

Compatibility
Incompatible with existing models

:SENSe:MEASure:AADJust32:STATe?

Response
<boolean> = <NR1 NUMERIC RESPONSE DATA>
1 Start
0 Stop

Function
Queries the Auto Adjust function state.

Example
> :SENSe:MEASure:AADJust32:STATe?
< 1

Compatibility
Incompatible with existing models
Chapter 7  SCPI Commands

:SENSe:MEASure:AADJust32:SELSlot <slot>,<interface>,<boolean>[,<unit>]

Parameter

- <slot> = <CHARACTER PROGRAM DATA>
  - SLOT1 to SLOT6  Slots No. 1 to 6
  - 1 to 6 when using the MP1800A, 1 to 4 when using the MT1810A
- <interface> = <DECIMAL NUMERIC PROGRAM DATA>
  - 1 to 4  Data1 to 4
- <boolean> = <BOOLEAN PROGRAM DATA>
  - ON, 1  Auto Adjust ON
  - OFF, 0  Auto Adjust OFF
- [<unit>] = <DECIMAL NUMERIC PROGRAM DATA>
  - 1 to 4  Mainframe No. 1 to 4
  - When using two or more MT1810A units in serial connection, specify the mainframe number.
  - Can be omitted. Mainframe No. 1 is specified when omitted.

Function

Selects the slot and data interface for which Auto Adjust is executed.

Example

To set Data 2 of Slot No.1 to Auto Adjust ON:

> :SENSe:MEASure:AADJust32:SELSlot SLOT1,2,ON

Compatibility

Incompatible with existing models

:SENSe:MEASure:AADJust32:SELSlot? <slot>,<interface>[,<unit>]

Parameter

- <slot> = <CHARACTER PROGRAM DATA>
  - SLOT1 to SLOT6  Slots No. 1 to 6
  - 1 to 6 when using the MP1800A, 1 to 4 when using the MT1810A
- <interface> = <DECIMAL NUMERIC PROGRAM DATA>
  - 1 to 4  Data1 to 4
- [<unit>] = <DECIMAL NUMERIC PROGRAM DATA>
  - 1 to 4  Mainframe No. 1 to 4
  - When using two or more MT1810A units in serial connection, specify the mainframe number.
  - Can be omitted. Mainframe No. 1 is specified when omitted.

Response

- <boolean> = <NR1 NUMERIC RESPONSE DATA>
  - 0  OFF
  - 1  ON

Function

Queries the Auto Adjust ON/OFF state for each data interface.

Example

To query the Auto Adjust ON/OFF state for Data 2 of Slot No.1:

> :SENSe:MEASure:AADJust32:SELSlot? SLOT1,2

< 1

Compatibility

Incompatible with existing models
### :SENSe:MEASure:AADJust32:SLOT? [<unit>]

<table>
<thead>
<tr>
<th><strong>Parameter</strong></th>
<th>&lt;unit&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1 to 4</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Response</strong></td>
<td>&lt;slot&gt; = &lt;CHARACTER RESPONSE DATA&gt;</td>
</tr>
<tr>
<td></td>
<td>SLOT1 to SLOT6</td>
</tr>
<tr>
<td></td>
<td>1 to 6 when using the MP1800A, 1 to 4 when using the MT1810A</td>
</tr>
</tbody>
</table>

**Note:**

The above response returns all slots for which Auto Adjust can be executed by delimiting with commas (,).

<table>
<thead>
<tr>
<th><strong>&lt;interface&gt;</strong></th>
<th>&lt;NR1 NUMERIC RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 to 4</td>
<td>Data interface No.1 to 4</td>
</tr>
</tbody>
</table>

**Function:** Queries the data interface for which Auto Adjust can be executed.

**Example:** To query the data interface in Mainframe No.3 for which Auto Adjust can be executed:

```plaintext
> :SENSe:MEASure:AADJust32:SLOT? 3
< SLOT1,1,SLOT1,2, SLOT1,3
```

**Compatibility:** Incompatible with existing models

### :SENSe:MEASure:AADJust32:SLASet

**Function:** Select all slots for Auto Adjust execution.

**Example:**

```plaintext
> :SENSe:MEASure:AADJust32:SLASet
```

**Compatibility:** Incompatible with existing models

### :SENSe:MEASure:AADJust32:SLAReset

**Function:** Deselect all slots for Auto Adjust execution.

**Example:**

```plaintext
> :SENSe:MEASure:AADJust32:SLAReset
```

**Compatibility:** Incompatible with existing models
7.12.14 Q measurement

This section describes the commands for Q measurement.

Figure 7.12.14-1  Q measurement screen
7.12.14.1 Vth vs. Q measurement commands

![Diagram of Vth vs. Q measurement setting field]

Table 7.12.14.1-1  Vth vs. Q measurement setting commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
</table>
### Table 7.12.14.1-1  Vth vs. Q measurement setting commands (Cont’d)

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>:SENSe:MEASure:QANalysis:THReshold:RESolution?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:MEASure:QANalysis:THReshold:ERANge?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:MEASure:QANalysis:THReshold:CALRange?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:MEASure:QANalysis:THReshold:ASEarch?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:MEASure:QANalysis:THReshold:CFILter?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:MEASure:QANalysis:THReshold:CURRent?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:MEASure:QANalysis:THReshold:ASEarch:PA MSet?</td>
</tr>
</tbody>
</table>

**:SENSe:MEASure:QANalysis:THReshold:STARt**

<table>
<thead>
<tr>
<th>Function</th>
<th>Starts Vth vs. Q measurement.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example</td>
<td>&gt; :SENSe:MEASure:QANalysis:THReshold:STARt</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Incompatible with existing models.</td>
</tr>
</tbody>
</table>

**:SENSe:MEASure:QANalysis:THReshold:STOP**

<table>
<thead>
<tr>
<th>Function</th>
<th>Stops Vth vs. Q measurement.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example</td>
<td>&gt; :SENSe:MEASure:QANalysis:THReshold:STOP</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Incompatible with existing models.</td>
</tr>
</tbody>
</table>

**:SENSe:MEASure:QANalysis:STATe?**

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;numeric&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Measurement has been stopped.</td>
</tr>
<tr>
<td>1</td>
<td>During measurement</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Function</th>
<th>Queries the Q measurement processing state.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example</td>
<td>&gt; :SENSe:MEASure:QANalysis:STATe?</td>
</tr>
<tr>
<td></td>
<td>&lt; 1</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Incompatible with existing models.</td>
</tr>
</tbody>
</table>
### :SENSe:MEASure:QANalysis:THReshold:SLOT

#### Parameter

- `<numeric>` = `<DECIMAL NUMERIC PROGRAM DATA>`
  - 1 to 6  Slots No. 1 to 6
  - 1 to 6 when using the MP1800A, 1 to 4 when using the MT1810A
- `<interface>` = `<DECIMAL NUMERIC PROGRAM DATA>`
  - 1 to 4  Data1 to 4
- `[<unit>]` = `<DECIMAL NUMERIC PROGRAM DATA>`
  - 1 to 4  Mainframe No. 1 to 4

When using two or more MT1810A units in serial connection, specify the mainframe number.

Can be omitted. Mainframe No. 1 is specified when omitted.

#### Function

Sets the Vth vs. Q measurement target slot.

#### Example

To set the target slot to Slot 2, Data1:

```plaintext
> :SENSe:MEASure:QANalysis:THReshold:SLOT 2,1
```

#### Compatibility

Incompatible with existing models.

### :SENSe:MEASure:QANalysis:THReshold:SLOT?

#### Parameter

- `[<unit>]` = `<DECIMAL NUMERIC PROGRAM DATA>`
  - 1 to 4  Mainframe No. 1 to 4

When using two or more MT1810A units in serial connection, specify the mainframe number.

Can be omitted. Mainframe No. 1 is specified when omitted.

#### Response

- `<numeric>` = `<NR1 NUMERIC RESPONSE DATA>`
  - 1 to 6  Slots No. 1 to 6
- `<interface>` = `<NR1 NUMERIC RESPONSE DATA>`
  - 1 to 4  Data1 to 4

#### Function

Queries the Vth vs. Q measurement target slot.

#### Example

> :SENSe:MEASure:QANalysis:THReshold:SLOT?

< 2,1

#### Compatibility

Incompatible with existing models.
### :SENSe:MEASure:QANalysis:THReshold:MODE <mode>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;mode&gt; = &lt;CHARACTER PROGRAM DATA&gt;</td>
<td>Parameter</td>
</tr>
<tr>
<td>SINGLE</td>
<td>Executes measurement once.</td>
</tr>
<tr>
<td>REPeat</td>
<td>Repeats measurement for the specified number of times.</td>
</tr>
<tr>
<td>UNTImed</td>
<td>Repeats measurement until it is directed to stop.</td>
</tr>
</tbody>
</table>

**Function**
Sets the measurement processing mode during Vth vs. Q measurement.

**Example**
To set the measurement processing mode to Single mode:
> :SENSe:MEASure:QANalysis:THReshold:MODE SINGle

**Compatibility**
Incompatible with existing models.

### :SENSe:MEASure:QANalysis:THReshold:MODE? 

<table>
<thead>
<tr>
<th>Response</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;mode&gt; = &lt;CHARACTER RESPONSE DATA&gt;</td>
<td>Response</td>
</tr>
<tr>
<td>SING, REP, UNT</td>
<td></td>
</tr>
</tbody>
</table>

**Function**
Queries the measurement processing mode during Vth vs. Q measurement.

**Example**
> :SENSe:MEASure:QANalysis:THReshold:MODE? 
< SING

**Compatibility**
Incompatible with existing models.

### :SENSe:MEASure:QANalysis:THReshold:TIMes <numeric>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;numeric&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
<td>Parameter</td>
</tr>
<tr>
<td>2 to 1000</td>
<td>2 to 1000 times, in 1-time steps</td>
</tr>
</tbody>
</table>

**Function**
Sets the number of measurements during Vth vs. Q measurement.

**Example**
To set the number of measurements to 100 times:
> :SENSe:MEASure:QANalysis:THReshold:TIMes 100

**Compatibility**
Incompatible with existing models.

### :SENSe:MEASure:QANalysis:THReshold:TIMes? 

<table>
<thead>
<tr>
<th>Response</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;numeric&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt;</td>
<td>Response</td>
</tr>
</tbody>
</table>

**Function**
Queries the number of measurements during Vth vs. Q measurement.

**Example**
> :SENSe:MEASure:QANalysis:THReshold:TIMes?
< 100

**Compatibility**
Incompatible with existing models.
**:SENSe:MEASure:QANalysis:THReshold:INTerval <numeric>**

**Parameter**

<numeric> = <DECIMAL NUMERIC PROGRAM DATA>

0 to 9999

0 to 9999 s, in 1-s steps

**Function**

Sets the measurement interval time during Vth vs. Q measurement.

**Example**

To set the measurement interval time to 50 sec.:

> :SENSe:MEASure:QANalysis:THReshold:INTerval 50

**Compatibility**

Incompatible with existing models.

**:SENSe:MEASure:QANalysis:THReshold:INTerval?**

**Response**

<numeric> = <NR1 NUMERIC RESPONSE DATA>

**Function**

Queries the measurement interval time during Vth vs. Q measurement.

**Example**

> :SENSe:MEASure:QANalysis:THReshold:INTerval?

< 50

**Compatibility**

Incompatible with existing models.

**:SENSe:MEASure:QANalysis:THReshold:DELay <numeric>**

**Parameter**

<numeric> = <DECIMAL NUMERIC PROGRAM DATA>

–1000 to 1000

–1000 to 1000 mUI, in 2-mUI steps

**Function**

Sets the measurement start delay position during Vth vs. Q measurement.

**Example**

To set the measurement start delay position to 100 mUI:

> :SENSe:MEASure:QANalysis:THReshold:DELay 100

**Compatibility**

Incompatible with existing models.

**:SENSe:MEASure:QANalysis:THReshold:DELay?**

**Response**

<numeric> = <NR1 NUMERIC RESPONSE DATA>

**Function**

Queries the measurement start delay position during Vth vs. Q measurement.

**Example**

> :SENSe:MEASure:QANalysis:THReshold:DELay?

< 100

**Compatibility**

Incompatible with existing models.
Chapter 7  SCPI Commands

:SENSe:MEASure:QANalysiss:THReshold:RESolution <mode>

Parameter  
<mode> = <CHARACTER PROGRAM DATA>
FINE          Fine mode
COARse       Coarse mode

Function  
Sets the measurement resolution (Fine/Coarse mode) during Vth vs. Q measurement.

Example  
To set the measurement resolution to Fine mode:
> :SENSe:MEASure:QANalysiss:THReshold:RESolution FINE

Compatibility  
Incompatible with existing models.

:SENSe:MEASure:QANalysiss:THReshold:RESolution?

Response  
<mode> = <CHARACTER RESPONSE DATA>
FINE, COAR

Function  
Queries the measurement resolution (Fine/Coarse mode) during Vth vs. Q measurement.

Example  
> :SENSe:MEASure:QANalysiss:THReshold:RESolution?
< FINE

Compatibility  
Incompatible with existing models.

:SENSe:MEASure:QANalysiss:THReshold:ERANge <upper>,<lower>

Parameter  
<upper> = <CHARACTER PROGRAM DATA>
E_3 to E_5   E–3 to E–5, in single steps
<lower> = <CHARACTER PROGRAM DATA>
E_7 to E_12  E–7 to E–12, in single steps

Function  
Sets the measurement error threshold range during Vth vs. Q measurement.

Example  
To set the measurement error threshold range to E–3 to E–12:
> :SENSe:MEASure:QANalysiss:THReshold:ERANge E_3,E_12

Compatibility  
Incompatible with existing models.

:SENSe:MEASure:QANalysiss:THReshold:ERANge?

Response  
<upper> = <CHARACTER RESPONSE DATA>
<lower> = <CHARACTER PROGRAM DATA>

Function  
Queries the measurement error threshold range during Vth vs. Q measurement.

Example  
> :SENSe:MEASure:QANalysiss:THReshold:ERANge?
< E_3,E_12

Compatibility  
Incompatible with existing models.
:SENSe:MEASure:QANalysiS:THReshoid:CALRange <upper>,<lower>

Parameter
- <upper> = <CHARACTER PROGRAM DATA>
  - E_3 to E_5  E–3 to E–5, in single steps
- <lower> = <CHARACTER PROGRAM DATA>
  - E_7 to E_12  E–7 to E–12, in single steps

Function
Sets the error threshold range during Vth vs. Q measurement recalculation.

Example
To set the error threshold range during recalculation to E–5 to E–10:
> :SENSe:MEASure:QANalysiS:THReshoid:CALRange E_5,E_10

Compatibility
Incompatible with existing models.

:SENSe:MEASure:QANalysiS:THReshoid:CALRange?

Response
- <upper> = <CHARACTER RESPONSE DATA>
- <lower> = <CHARACTER RESPONSE DATA>

Function
Queries the error threshold range during Vth vs. Q measurement recalculation.

Example
> :SENSe:MEASure:QANalysiS:THReshoid:CALRange?
< E_5,E_10

Compatibility
Incompatible with existing models.

:SENSe:MEASure:QANalysiS:THReshoid:ASEarch <item>

Parameter
- <item> = <CHARACTER PROGRAM DATA>
  - OFF   Auto search OFF
  - THReshoid  Threshold auto search
  - PTHReshoid  Phase & threshold auto search

Function
Sets whether to execute auto search during Vth vs. Q measurement.

Example
To set the threshold auto search ON:
> :SENSe:MEASure:QANalysiS:THReshoid:ASEarch THReshoid

Compatibility
Incompatible with existing models.

:SENSe:MEASure:QANalysiS:THReshoid:ASEarch?

Response
- <item> = <CHARACTER RESPONSE DATA>
  - OFF, THR, PTHR

Function
Queries whether to execute auto search during Vth vs. Q measurement.

Example
> :SENSe:MEASure:QANalysiS:THReshoid:ASEarch?
< THR

Compatibility
Incompatible with existing models.
Chapter 7  SCPI Commands

:SENSe:MEASure:QANalysis:THReshold:CFILter<boolean>,[<numeric>]

Parameter

- `<boolean>` = `<BOOLEAN PROGRAM DATA>`
  - 0 or OFF: Filter ON
  - 1 or ON: Filter OFF
- `<numeric>` = `<DECIMAL NUMERIC PROGRAM DATA>`
  - 80 to 100: 80 to 100%, in 1% steps

When `<boolean>` is set to "0" or "OFF", `<numeric>` can be omitted.

Function

Sets the minimum correlation coefficient with which the Q value is recognized to be valid during Vth vs. Q measurement.

Example

To set the minimum correlation coefficient to 90%:

```
> :SENSe:MEASure:QANalysis:THReshold:CFILter 1,90
```

Compatibility

Incompatible with existing models.

:SENSe:MEASure:QANalysis:THReshold:CFILter?

Response

- `<boolean>` = `<NR1 NUMERIC RESPONSE DATA>`
  - 0: OFF
  - 1: ON
- `<numeric>` = `<NR1 NUMERIC RESPONSE DATA>`
  - 80 to 100

When `<boolean>` is 0, `<numeric>` is omitted.

Function

Queries the minimum correlation coefficient with which the Q value is recognized to be valid during Vth vs. Q measurement.

Example

```
> :SENSe:MEASure:QANalysis:THReshold:CFILter?
< 1,90
```

Compatibility

Incompatible with existing models.

:SENSe:MEASure:QANalysis:THReshold:CURRent <boolean>

Parameter

- `<boolean>` = `<BOOLEAN PROGRAM DATA>`
  - 0 or OFF: OFF: Updates at measurement termination
  - 1 or ON: ON: Updates every second

Function

Sets the graph display update period during Vth vs. Q measurement.

Example

To set the graph display update timing to every second:

```
> :SENSe:MEASure:QANalysis:THReshold:CURRent 1
```

Compatibility

Incompatible with existing models.
### :SENSe:MEASure:QANalysis:THReshold:CURRent?

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;boolean&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>OFF</td>
</tr>
<tr>
<td>1</td>
<td>ON</td>
</tr>
</tbody>
</table>

**Function**
Queries the graph display update period during Vth vs. Q measurement.

**Example**
```plaintext
> :SENSe:MEASure:QANalysis:THReshold:CURRent?
< 1
```

**Compatibility**
Incompatible with existing models.

### :SENSe:MEASure:QANalysis:THReshold:ASEarch:PAMSet <threshold>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;threshold&gt; = &lt;CHARACTER PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>OFF</td>
<td>PAM Auto Search OFF</td>
</tr>
<tr>
<td>TOP</td>
<td>Searches the Top threshold</td>
</tr>
<tr>
<td>MIDDLE</td>
<td>Searches the Middle threshold</td>
</tr>
<tr>
<td>BOTTOM</td>
<td>Searches the Bottom threshold</td>
</tr>
</tbody>
</table>

**Function**
Sets the search target threshold for PAM Auto Search when measuring Vth vs Q.

**Example**
To set to Top threshold search:
```plaintext
> :SENSe:MEASure:QANalysis:THReshold:ASEarch:PAMSet TOP
```

**Compatibility**
Incompatible with existing models.

### :SENSe:MEASure:QANalysis:THReshold:ASEarch:PAMSet?

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;item&gt; = &lt;CHARACTER RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>OFF, TOP, MIDDLE, BOTTOM</td>
<td></td>
</tr>
</tbody>
</table>

**Function**
Queries the search target threshold for PAM Auto Search when measuring Vth vs Q.

**Example**
```plaintext
> :SENSe:MEASure:QANalysis:THReshold:ASEarch:PAMSet?
< TOP
```

**Compatibility**
Incompatible with existing models.
7.12.14.2 Phase vs. Q measurement setting commands

Table 7.12.14.2-1 Phase vs. Q measurement setting commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1]</td>
<td>Start</td>
<td><code>:SENSe:MEASure:QANalysis:PHASe:STARt</code></td>
</tr>
<tr>
<td></td>
<td>Start End</td>
<td><code>:SENSe:MEASure:QANalysis:PHASe:RANGe?</code></td>
</tr>
<tr>
<td></td>
<td>Center Span</td>
<td><code>:SENSe:MEASure:QANalysis:PHASe:WIDTh</code></td>
</tr>
<tr>
<td></td>
<td></td>
<td><code>:SENSe:MEASure:QANalysis:PHASe:WIDTh?</code></td>
</tr>
<tr>
<td></td>
<td></td>
<td><code>:SENSe:MEASure:QANalysis:PHASe:RESolution?</code></td>
</tr>
<tr>
<td></td>
<td></td>
<td><code>:SENSe:MEASure:QANalysis:PHASe:ERANge?</code></td>
</tr>
</tbody>
</table>
### 7.12 28G/32G bit/s ED Commands

#### Table 7.12.14.2-1 Phase vs. Q measurement setting commands (Cont’d)

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>:SENSe:MEASure:QANalysis:PHASE:CALRange?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:MEASure:QANalysis:PHASE:ASEarch?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:MEASure:QANalysis:PHASE:CFILTter?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:MEASure:QANalysis:PHASE:CURRENT?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:MEASure:QANalysis:PHASE:ASEarch:PAMSet?</td>
</tr>
</tbody>
</table>

### :SENSe:MEASure:QANalysis:PHASE:STARt

**Function**

Starts Phase vs. Q measurement.

**Example**

> :SENSe:MEASure:QANalysis:PHASE:STARt

**Compatibility**

Incompatible with existing models.

### :SENSe:MEASure:QANalysis:PHASE:STOP

**Function**

Stops Phase vs. Q measurement.

**Example**

> :SENSe:MEASure:QANalysis:PHASE:STOP

**Compatibility**

Incompatible with existing models.

### :SENSe:MEASure:QANalysis:STATe?

**Response**

```
numeric = <NR1 NUMERIC RESPONSE DATA>
```

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>During measurement stop</td>
</tr>
<tr>
<td>1</td>
<td>During measurement</td>
</tr>
</tbody>
</table>

**Function**

Queries the measurement processing state during Q measurement.

**Example**

> :SENSe:MEASure:QANalysis:STATe?

< 1

**Compatibility**

Incompatible with existing models.
### :SENSe:MEASure:QANalysIs:PHASe:SLOT

<numeric>,<interface>[,<unit>]

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;numeric&gt;</td>
<td>= &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
</tr>
<tr>
<td>1 to 6</td>
<td>Slots No. 1 to 6</td>
</tr>
<tr>
<td>1 to 6 when using the MP1800A, 1 to 4 when using the MT1810A</td>
<td></td>
</tr>
<tr>
<td>&lt;interface&gt;</td>
<td>= &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
</tr>
<tr>
<td>1 to 4</td>
<td>Data1 to 4</td>
</tr>
<tr>
<td>[&lt;unit&gt;]</td>
<td>= &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
</tr>
<tr>
<td>1 to 4</td>
<td>Mainframe No. 1 to 4</td>
</tr>
</tbody>
</table>

When using two or more MT1810A units in serial connection, specify the mainframe number.

Can be omitted. Mainframe No. 1 is specified when omitted.

**Function**
Selects the Phase vs. Q measurement target slot.

**Example**
To set the target slot to Slot2, Data1:

```
> :SENSe:MEASure:QANalysIs:PHASe:SLOT 2,1
```

**Compatibility**
Incompatible with existing models.

### :SENSe:MEASure:QANalysIs:PHASe:SLOT? [<unit>]

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>[&lt;unit&gt;]</td>
<td>= &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
</tr>
<tr>
<td>1 to 4</td>
<td>Mainframe No. 1 to 4</td>
</tr>
</tbody>
</table>

When using two or more MT1810A units in serial connection, specify the mainframe number.

Can be omitted. Mainframe No. 1 is specified when omitted.

**Response**

<table>
<thead>
<tr>
<th>&lt;numeric&gt;</th>
<th>= &lt;NR1 NUMERIC RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 to 6</td>
<td>Slots No. 1 to 6</td>
</tr>
<tr>
<td>&lt;interface&gt;</td>
<td>= &lt;NR1 NUMERIC RESPONSE DATA&gt;</td>
</tr>
<tr>
<td>1 to 4</td>
<td>Data1 to 4</td>
</tr>
</tbody>
</table>

**Function**
Queries the Phase vs. Q measurement target slot.

**Example**
```
```

**Compatibility**
Incompatible with existing models.
\texttt{:SENSe:MEASure:QANalys\textis{is}:PH\textis{is}:RANGe \textit{<start>},<end>,<step>}

**Parameter**
- \textit{\texttt{	exttt{<start>}}} = \texttt{<DECIMAL NUMERIC PROGRAM DATA>}
  - \texttt{--1000 to 998} \quad \texttt{--1000 to 998 mUI/2 mUI Step}
- \textit{\texttt{<end>}} = \texttt{<DECIMAL NUMERIC PROGRAM DATA>}
  - \texttt{--998 to 1000} \quad \texttt{--998 to 1000 mUI/2 mUI Step}
- \textit{\texttt{<step>}} = \texttt{<DECIMAL NUMERIC PROGRAM DATA>}
  - \texttt{2 to 200} \quad \texttt{2 to 200 mUI/2 mUI Step}

**Function**
Sets the measurement range and measurement step during Phase vs. Q measurement.

**Example**
To set the measurement range to \texttt{--200 to 300 mUI} and measurement step to \texttt{10 mUI}:
\[> \texttt{:SENSe:MEASure:QANalys\textis{is}:PH\textis{is}:RANGe --200,300,10}\]

**Compatibility**
Incompatible with existing models.

\texttt{:SENSe:MEASure:QANalys\textis{is}:PH\textis{is}:RANGe?}

**Response**
- \textit{\texttt{\texttt{<start>}}} = \texttt{<NR1 NUMERIC RESPONSE DATA>}
- \textit{\texttt{<end>}} = \texttt{<NR1 NUMERIC RESPONSE DATA>}
- \textit{\texttt{<step>}} = \texttt{<NR1 NUMERIC RESPONSE DATA>}

**Function**
Queries the measurement range during Phase vs. Q measurement.

**Example**
\[> \texttt{:SENSe:MEASure:QANalys\textis{is}:PH\textis{is}:RANGe?}\]
\[< --200,300,10\]

**Compatibility**
Incompatible with existing models.

\texttt{:SENSe:MEASure:QANalys\textis{is}:PH\textis{is}:WIDTh \textit{<center>},<span>,<step>}

**Parameter**
- \textit{\texttt{\texttt{<center>}}} = \texttt{<DECIMAL NUMERIC PROGRAM DATA>}
  - \texttt{--998 to 998} \quad \texttt{--998 to 998 mUI/2 mUI Step}
- \textit{\texttt{<span>}} = \texttt{<DECIMAL NUMERIC PROGRAM DATA>}
  - \texttt{4 to 2000} \quad \texttt{4 to 2000 mUI/4 mUI Step}
- \textit{\texttt{<step>}} = \texttt{<DECIMAL NUMERIC PROGRAM DATA>}
  - \texttt{2 to 200} \quad \texttt{2 to 200 mUI/2 mUI Step}

**Function**
Sets the measurement range and measurement step during Phase vs. Q measurement.

**Example**
To set the measurement range to \texttt{100 mUI}, centered at \texttt{200 mUI}, in \texttt{10-mUI steps}:
\[> \texttt{:SENSe:MEASure:QANalys\textis{is}:PH\textis{is}:WIDTh 100,200,10}\]

**Compatibility**
Incompatible with existing models.
**Chapter 7  SCPI Commands**

**:SENSe:MEASure:QANalysis:PHASe:WIDTh?**

**Response**
- `<center>` = `<NR1 NUMERIC RESPONSE DATA>`
- `<span>` = `<NR1 NUMERIC RESPONSE DATA>`
- `<step>` = `<NR1 NUMERIC RESPONSE DATA>`

**Function**
Queries the measurement range during Phase vs. Q measurement.

**Example**
> :SENSe:MEASure:QANalysis:PHASe:WIDTh?
  > <100,200,10>

**Compatibility**
Incompatible with existing models.

**:SENSe:MEASure:QANalysis:PHASe:RESolution <mode>**

**Parameter**
- `<mode>` = `<CHARACTER PROGRAM DATA>`
  - FINE  Fine mode
  - COARse  Coarse mode

**Function**
Sets the measurement resolution (Fine/Coarse mode) during Phase vs. Q measurement.

**Example**
To set the measurement resolution to Fine mode:
> :SENSe:MEASure:QANalysis:PHASe:RESolution FINE

**Compatibility**
Incompatible with existing models.

**:SENSe:MEASure:QANalysis:PHASe:RESolution?**

**Response**
- `<mode>` = `<CHARACTER RESPONSE DATA>`
  - FINE, COAR

**Function**
Queries the measurement resolution (Fine/Coarse mode) during Phase vs. Q measurement.

**Example**
> :SENSe:MEASure:QANalysis:PHASe:RESolution?
  > FINE

**Compatibility**
Incompatible with existing models.

**:SENSe:MEASure:QANalysis:PHASe:ERANge <upper>,<lower>**

**Parameter**
- `<upper>` = `<CHARACTER PROGRAM DATA>`
  - E_3 to E_5  E–3 to E–5, in single steps
  - E_7 to E_12  E–7 to E–12, in single steps
- `<lower>` = `<CHARACTER PROGRAM DATA>`
  - E_3 to E_5  E–3 to E–5, in single steps
  - E_7 to E_12  E–7 to E–12, in single steps

**Function**
Sets the measurement error threshold range during Phase vs. Q measurement.

**Example**
To set the measurement error threshold range to E–3 to E–12:
> :SENSe:MEASure:QANalysis:PHASe:ERANge E_3,E_12

**Compatibility**
Incompatible with existing models.
7.12 28G/32G bit/s ED Commands

:SENSE:MEASURE:QANalysis:PHASE:ERANGE?

Response
<upper> = <CHARACTER RESPONSE DATA>
<lower> = <CHARACTER RESPONSE DATA>

Function
Queries the measurement error threshold range during Phase vs. Q measurement.

Example
> :SENSe:MEASURE:QANalysis:PHASE:ERANGE?
< E_3,E_12

Compatibility
Incompatible with existing models.

:SENSE:MEASURE:QANalysis:PHASE:CALRange <upper>,<lower>

Parameter
<upper> = <CHARACTER PROGRAM DATA>
E_3 to E_5  E–3 to E–5, in single steps
<lower> = <CHARACTER PROGRAM DATA>
E_7 to E_12  E–7 to E–12, in single steps

Function
Sets the error threshold range during Phase vs. Q measurement recalculation.

Example
To set the error threshold range during recalculation to E–5 to E–10:
> :SENSe:MEASURE:QANalysis:PHASE:CALRange E_5,E_10

Compatibility
Incompatible with existing models.

:SENSE:MEASURE:QANalysis:PHASE:CALRange?

Response
<upper> = <CHARACTER RESPONSE DATA>
<lower> = <CHARACTER RESPONSE DATA>

Function
Queries the error threshold range during Phase vs. Q measurement recalculation.

Example
> :SENSe:MEASURE:QANalysis:PHASE:CALRange?
< E_5,E_10

Compatibility
Incompatible with existing models.

:SSENSe:MEASURE:QANalysis:PHASE:ASEarch <item>

Parameter
<item> = <CHARACTER PROGRAM DATA>
OFF   Auto search OFF
THReshold  Threshold auto search

Function
Sets execution of auto search during Phase vs. Q measurement.

Example
To set the threshold auto search ON:
> :SENSe:MEASURE:QANalysis:PHASE:ASEarch THReshold

Compatibility
Incompatible with existing models.
### Chapter 7  SCPI Commands

#### :SENSe:MEASure:QANalysis:PHASe:ASEarch?

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;item&gt; = &lt;CHARACTER RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>OFF, THR</td>
</tr>
</tbody>
</table>

**Function**
Queries the execution of auto search during Phase vs. Q measurement.

**Example**

> :SENSe:MEASure:QANalysis:PHASe:ASEarch?

< THR

**Compatibility**
Incompatible with existing models.

#### :SENSe:MEASure:QANalysis:PHASe:CFILter <boolean>[,<numeric>]

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;boolean&gt; = &lt;BOOLEAN PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0 or OFF Filter OFF</td>
</tr>
<tr>
<td></td>
<td>1 or ON Filter ON</td>
</tr>
<tr>
<td></td>
<td>&lt;numeric&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
</tr>
<tr>
<td></td>
<td>80 to 100 80 to 100%, in 1% steps</td>
</tr>
</tbody>
</table>

**Function**
Sets the minimum correlation coefficient with which the Q value is recognized to be valid during Phase vs. Q measurement.

**Example**
To set the minimum correlation coefficient to 90%:

> :SENSe:MEASure:QANalysis:PHASe:CFILter 1,90

**Compatibility**
Incompatible with existing models.

#### :SENSe:MEASure:QANalysis:PHASe:CFILter?

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;numeric&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0 OFF</td>
</tr>
<tr>
<td></td>
<td>1 ON</td>
</tr>
<tr>
<td></td>
<td>&lt;numeric&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt;</td>
</tr>
<tr>
<td></td>
<td>80 to 100 80 to 100%</td>
</tr>
</tbody>
</table>

**Function**
Queries the minimum correlation coefficient with which the Q value is recognized to be valid during Phase vs. Q measurement.

**Example**

> :SENSe:MEASure:QANalysis:PHASe:CFILter?

< 1,90

**Compatibility**
Incompatible with existing models.
**SENSe:MEASure:QANalysS:PHASe:CURRent <boolean>**

**Parameter**
- `<boolean> = <BOOLEAN PROGRAM DATA>`
  - 0 or OFF: OFF: Updates at measurement termination
  - 1 or ON: ON: Updates every second

**Function**
Sets the graph display update period during Phase vs. Q measurement.

**Example**
To set the graph display update timing to every second:
> :SENSe:MEASure:QANalysS:PHASe:CURRent 1

**Compatibility**
Incompatible with existing models.

**SENSe:MEASure:QANalysS:PHASe:CURRent?**

**Response**
- `<boolean> = <NR1 NUMERIC RESPONSE DATA>`
  - 0: OFF
  - 1: ON

**Function**
Queries the graph display update period during Phase vs. Q measurement.

**Example**
> :SENSe:MEASure:QANalysS:PHASe:CURRent?
< 1

**Compatibility**
Incompatible with existing models.

**SENSe:MEASure:QANalysS:PHASe:ASEarch:PAMSet <threshold>**

**Parameter**
- `<threshold> = <CHARACTER PROGRAM DATA>`
  - OFF: PAM Auto Search OFF
  - TOP: Searches the Top threshold
  - MIDDle: Searches the Middle threshold
  - BOTTom: Searches the Bottom threshold

**Function**
Sets the search target threshold for PAM Auto Search when measuring Phase vs Q.

**Example**
To set to Top threshold search.
> :SENSe:MEASure:QANalysS:PHASe:ASEarch:PAMSet TOP

**Compatibility**
Incompatible with existing models.

**SENSe:MEASure:QANalysS:PHASe:ASEarch:PAMSet?**

**Response**
- `<item> = <CHARACTER RESPONSE DATA>`
  - OFF, TOP, MIDD, BOTT

**Function**
Queries the search target threshold for PAM Auto Search when measuring Phase vs Q.

**Example**
> :SENSe:MEASure:QANalysS:PHASe:ASEarch:PAMSet?
< TOP

**Compatibility**
Incompatible with existing models.
7.12.14.3 Graph display setting commands

Figure 7.12.14.3-1  Graph display screen (Threshold vs Q tab)

Figure 7.12.14.3-2  Graph display screen (Phase vs Q tab)
### Table 7.12.14.3-1  Graph display setting commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>:DISPlay:RESult:QANalysis:MODE?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:DISPlay:RESult:QANalysis:SCALE:VERTical?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:DISPlay:RESult:QANalysis:SCALE:HORizontal?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:DISPlay:RESult:QANalysis:THReshold:MNUMber</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:DISPlay:RESult:QANalysis:PHASe:MNUMber</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:DISPlay:RESult:QANalysis:THReshold:MNUMber</td>
</tr>
</tbody>
</table>

#### :DISPlay:RESult:QANalysis:MODE <mode>

**Parameter**

- `<mode>` = `<CHARACTER PROGRAM DATA>`
  - THREshold: Vth vs. Q measurement
  - PHASe: Phase vs. Q measurement

**Function**

Sets the graph display item from Vth vs. Q measurement or Phase vs. Q measurement.

The settings for the display scale and the marker correspond to this setting.

**Example**

To set to Vth vs. Q measurement:

> :DISPlay:RESult:QANalysis:MODE THREshold

**Compatibility**

Incompatible with existing models.

#### :DISPlay:RESult:QANalysis:MODE?

**Response**

- `<mode>` = `<CHARACTER RESPONSE DATA>`
  - THRE, PHAS

**Function**

Queries the item to be displayed in the graph during Vth vs. Q measurement.

**Example**

> :DISPlay:RESult:QANalysis:MODE?

< THRE

**Compatibility**

Incompatible with existing models.
### :DISPlay:RESult:QANalysis:SCALe:ASCale

**Function**
Automatically adjusts the graph display in Q measurement.

**Example**
> :DISPlay:RESult:QANalysis:SCALe:ASCale

**Compatibility**
Incompatible with existing models.

### :DISPlay:RESult:QANalysis:SCALe:VERTical <max>,[<step>]

**Parameter**
<max>,<step> = <DECIMAL NUMERIC PROGRAM DATA>

#### Table 7.12.14.3-2 Parameters for Threshold vs. Q tab

<table>
<thead>
<tr>
<th>Format</th>
<th>&lt;max&gt;</th>
<th>&lt;step&gt;</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vth vs BER(Y = Log)</td>
<td>2 to 7 (E−2 to E−7), in single steps</td>
<td>Inputs the minimum value. 8 to 14 (E−8 to E−14), in single steps</td>
<td>A difference between &lt;max&gt; and &lt;step&gt; must be 7 or greater even number.</td>
</tr>
<tr>
<td>Vth vs BER(Y = Log(−Ln))</td>
<td>2 to 7 (E−2 to E−7), in single steps</td>
<td>Inputs the minimum value. 8 to 14 (E−8 to E−14), in single steps</td>
<td>A difference between &lt;max&gt; and &lt;step&gt; must be 7 or greater even number.</td>
</tr>
<tr>
<td>Vth vs Q</td>
<td>10 to 1000, in 10 steps</td>
<td>1 to 100, in single steps</td>
<td>When Linear</td>
</tr>
<tr>
<td></td>
<td>−40 to 60, in 10 steps</td>
<td>1 to 11, in single steps</td>
<td>When Log</td>
</tr>
<tr>
<td>Times vs Q</td>
<td>10 to 1000, in 10 steps</td>
<td>1 to 100, in single steps</td>
<td>When Linear</td>
</tr>
<tr>
<td></td>
<td>−40 to 60, in 10 steps</td>
<td>1 to 11, in single steps</td>
<td>When Log</td>
</tr>
<tr>
<td>Histogram</td>
<td>50 to 1000, in 50 steps</td>
<td>Omitted</td>
<td></td>
</tr>
</tbody>
</table>
### Table 7.12.14.3-3 Parameters for Phase vs. Q tab

<table>
<thead>
<tr>
<th>Format</th>
<th>&lt;max&gt;</th>
<th>&lt;step&gt;</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vth BER (Y = Log)</td>
<td>2 to 7 (E−2 to E−7), in</td>
<td>Inputs the minimum value.</td>
<td>A difference between &lt;max&gt; and &lt;step&gt; must be 7 or greater even number.</td>
</tr>
<tr>
<td></td>
<td>single steps</td>
<td>8 to 14 (E−8 to E−14), in</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>single steps</td>
<td></td>
</tr>
<tr>
<td>Vth BER (Y = Log(−Ln))</td>
<td>2 to 7 (E−2 to E−7), in</td>
<td>Inputs the minimum value.</td>
<td>A difference between &lt;max&gt; and &lt;step&gt; must be 7 or greater even number.</td>
</tr>
<tr>
<td></td>
<td>single steps</td>
<td>8 to 14 (E−8 to E−14), in</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>single steps</td>
<td></td>
</tr>
<tr>
<td>Vth BER</td>
<td>10 to 1000, in 10 steps</td>
<td>1 to 100, in single steps</td>
<td>When Linear</td>
</tr>
<tr>
<td></td>
<td>−40 to 60, in 10 steps</td>
<td>1 to 11, in single steps</td>
<td>When Log</td>
</tr>
<tr>
<td>Phase vs Q</td>
<td>10 to 1000, in 10 steps</td>
<td>1 to 100, in single steps</td>
<td>When Linear</td>
</tr>
<tr>
<td></td>
<td>−40 to 60, in 10 steps</td>
<td>1 to 11, in single steps</td>
<td>When Log</td>
</tr>
<tr>
<td>Phase vs σ</td>
<td>0.0010 to 1.0000, in</td>
<td>0.001 to 0.1000, 0.0001 steps</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.0010 steps</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Phase vs μ</td>
<td>−3.990 to 4.000 (V), in</td>
<td>0.001 to 0.800 (V), in 0.001</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.010 V steps</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Phase vs Opt BER</td>
<td>10 to 1 (E−10 to E−1), in</td>
<td>Inputs the minimum value.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>single steps</td>
<td>199 to 20 (E−199 to E−20),</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>in single steps</td>
<td></td>
</tr>
<tr>
<td>Phase vs Opt Threshold</td>
<td>−3.990 to 4.000 (V), in</td>
<td>0.001 to 0.800 (V), in 0.001</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.010 V steps</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Phase vs Correlation</td>
<td>10 to 100 (%), in 10 %</td>
<td>0 to 90 (%), in 10 % steps</td>
<td>A difference between &lt;max&gt; and &lt;step&gt; must be 10 or greater</td>
</tr>
<tr>
<td></td>
<td>steps</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Note:**

Settings vary, depending on the status of the installed options for each slot, and the data input interface settings.

**Function**

Sets the vertical scale of the graph display during Q measurement. Inputs the minimum value instead of the step width, only when the vertical axis displays the error rate.

**Example**

To set the vertical scale during QLog display to maximum 10 dB, in 1-dB steps:

```plaintext
> :DISPlay:RESult:QANalysis:SCALe:VERTical 10,1
```

**Compatibility**

Incompatible with existing models.

---

**:DISPlay:RESult:QANalysis:SCALe:VERTical?**

**Response**

```
<max>,<step> = <NR1 NUMERIC RESPONSE DATA>
```

**Function**

Queries the vertical scale of the graph display during Q measurement.

**Example**

To query the vertical graph scale during QLog display:

```plaintext
> :DISPlay:RESult:QANalysis:SCALe:VERTical?
< 10,1
```

**Compatibility**

Incompatible with existing models.
**:DISPlay:RESult:QANalysis:SCALe:HORizontal <max>,<step>**

Parameter: $<\text{max}>$, $<\text{step}> = <\text{DECIMAL NUMERIC PROGRAM DATA}>$

### Table 7.12.14.3-4  Parameters for Threshold vs. Q tab

<table>
<thead>
<tr>
<th>Format</th>
<th>$&lt;\text{max}&gt;$</th>
<th>$&lt;\text{step}&gt;$</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vth vs BER ($Y = \text{Log}$)</td>
<td>$-3.990$ to $4.000$ (V), in $0.010$ V steps</td>
<td>$0.001$ to $0.800$ (V), in $0.001$ V steps</td>
<td></td>
</tr>
<tr>
<td>Vth vs BER ($Y = \text{Log}(-\text{Ln})$)</td>
<td>$-3.990$ to $4.000$ (V), in $0.010$ V steps</td>
<td>$0.001$ to $0.800$ (V), in $0.001$ V steps</td>
<td></td>
</tr>
<tr>
<td>Vth vs Q</td>
<td>$-3.990$ to $4.000$ (V), in $0.010$ V steps</td>
<td>$0.001$ to $0.800$ (V), in $0.001$ V steps</td>
<td></td>
</tr>
<tr>
<td>Times vs Q</td>
<td>$100$ to $1000$, in $50$ steps</td>
<td>$10$ (fixed)</td>
<td></td>
</tr>
<tr>
<td>Histogram</td>
<td>$0.16$ to $1000.00$, in $0.01$ steps</td>
<td>$16/32/64/128/256$</td>
<td>When Linear $&lt;\text{max}&gt;$ must be greater than $&lt;\text{step}&gt;$.</td>
</tr>
<tr>
<td></td>
<td>$-49.84$ to $60.00$ (dB), in $0.01$ steps</td>
<td>$16/32/64$</td>
<td>When Log</td>
</tr>
</tbody>
</table>

### Table 7.12.14.3-5  Parameters for Phase vs. Q tab

<table>
<thead>
<tr>
<th>Format</th>
<th>$&lt;\text{max}&gt;$</th>
<th>$&lt;\text{step}&gt;$</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vth vs BER ($Y = \text{Log}$)</td>
<td>$-3.990$ to $4.000$ (V), in $0.010$ V steps</td>
<td>$0.001$ to $0.800$ (V), in $0.001$ V steps</td>
<td></td>
</tr>
<tr>
<td>Vth vs BER ($Y = \text{Log}(-\text{Ln})$)</td>
<td>$-3.990$ to $4.000$ (V), in $0.010$ V steps</td>
<td>$0.001$ to $0.800$ (V), in $0.001$ V steps</td>
<td></td>
</tr>
<tr>
<td>Vth vs Q</td>
<td>$-3.990$ to $4.000$ (V), in $0.010$ V steps</td>
<td>$0.001$ to $0.800$ (V), in $0.001$ V steps</td>
<td></td>
</tr>
<tr>
<td>Phase vs Q</td>
<td>$-900$ to $1000$ (mUI), in $100$ mUI steps</td>
<td>$10$ to $200$ (mUI), in $10$ mUI steps</td>
<td></td>
</tr>
<tr>
<td>Phase vs $\sigma$</td>
<td>$-900$ to $1000$ (mUI), in $100$ mUI steps</td>
<td>$10$ to $200$ (mUI), in $10$ mUI steps</td>
<td></td>
</tr>
<tr>
<td>Phase vs $\mu$</td>
<td>$-900$ to $1000$ (mUI), in $100$ mUI steps</td>
<td>$10$ to $200$ (mUI), in $10$ mUI steps</td>
<td></td>
</tr>
<tr>
<td>Phase vs Opt BER</td>
<td>$-900$ to $1000$ (mUI), in $100$ mUI steps</td>
<td>$10$ to $200$ (mUI), in $10$ mUI steps</td>
<td></td>
</tr>
<tr>
<td>Phase vs Opt Threshold</td>
<td>$-900$ to $1000$ (mUI), in $100$ mUI steps</td>
<td>$10$ to $200$ (mUI), in $10$ mUI steps</td>
<td></td>
</tr>
<tr>
<td>Phase vs Correlation</td>
<td>$-900$ to $1000$ (mUI), in $100$ mUI steps</td>
<td>$10$ to $200$ (mUI), in $10$ mUI steps</td>
<td></td>
</tr>
</tbody>
</table>

**Note:**

Settings vary, depending on the status of the installed options for each slot, and the data input interface settings.
Function  
Sets the horizontal scale of the graph display during Q measurements. The format varies depending on the graph type.

Example  
To set the horizontal scale during Vth display to maximum –1.000 V, in 0.100 steps:
> :DISPlay:RESult:QANalysis:SCALe:HORizontal -1.000,0.100

Compatibility  
Incompatible with existing models.

:DISPlay:RESult:QANalysis:SCALe:HORizontal?

Response  
<max>,<step> = <NR1 NUMERIC RESPONSE DATA>

Function  
Queries the horizontal scale of the graph display during Q measurement.

Example  
To query the horizontal graph scale during Vth display:
> :DISPlay:RESult:QANalysis:SCALe:HORizontal?
< -1.000,0.100

Compatibility  
Incompatible with existing models.

:DISPlay:RESult:QANalysis:THReshold:ITEM <mode>

Parameter  
<mode> = <CHARACTER PROGRAM DATA>
LBER Vth vs. BER (Y axis Log)
LNBer Vth vs. BER (Y axis Log (–Ln))
VTHQ Vth vs. Q
TIMQ Times vs. Q
HISTogram Histogram

Function  
Sets the item to be displayed in the graph during Vth vs. Q measurement.

Example  
To set the item to be displayed to Vth vs. Q:
> :DISPlay:RESult:QANalysis:THReshold:ITEM VTHQ

Compatibility  
Incompatible with existing models.

:DISPlay:RESult:QANalysis:THReshold:ITEM?

Response  
<mode> = <CHARACTER RESPONSE DATA>
LBER, LNB, VTHQ, TIMQ, HIST

Function  
Queries the item to be displayed in the graph during Vth vs. Q measurement.

Example  
> :DISPlay:RESult:QANalysis:THReshold:ITEM?  
< VTHQ

Incompatible with existing models.
### :DISPlay:RESult:QAnalysis:THReshold:MNUMber <numeric>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;numeric&gt; = &lt;DECIMAL PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 to 1000</td>
<td>Measurement number (number of measurements during repeat measurement)</td>
</tr>
</tbody>
</table>

**Function**
Sets the measurement number for which the Vth vs. Q measurement result is displayed in a graph.

**Example**
To set the measurement number for graph display to 100:

```
> :DISPlay:RESult:QAnalysis:THReshold:MNUMber 100
```

**Compatibility**
Incompatible with existing models.

### :DISPlay:RESult:QAnalysis:THReshold:MNUMber? 

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;numeric&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt;</th>
</tr>
</thead>
</table>

**Function**
Queries the measurement number for which the Vth vs. Q measurement result is displayed in a graph.

**Example**
To query the measurement number for graph display:

```
> :DISPlay:RESult:QAnalysis:THReshold:MNUMber?
< 100
```

**Compatibility**
Incompatible with existing models.

### :DISPlay:RESult:QAnalysis:PHASe:ITEM <mode>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;mode&gt; = &lt;CHARACTER PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>LBER</td>
<td>Vth vs. BER (Y axis Log)</td>
</tr>
<tr>
<td>LNBER</td>
<td>Vth vs. BER (Y axis Log (–Ln))</td>
</tr>
<tr>
<td>VTHQ</td>
<td>Vth vs. Q</td>
</tr>
<tr>
<td>PHAQ</td>
<td>Phase vs. Q</td>
</tr>
<tr>
<td>PHASigma</td>
<td>Phase vs. σ</td>
</tr>
<tr>
<td>PHAMu</td>
<td>Phase vs. μ</td>
</tr>
<tr>
<td>PHABer</td>
<td>Phase vs. Optimum BER</td>
</tr>
<tr>
<td>PHAVth</td>
<td>Phase vs. Optimum Vth</td>
</tr>
<tr>
<td>PHACorre</td>
<td>Phase vs. Correlation coefficient</td>
</tr>
</tbody>
</table>

**Function**
Sets the item to be displayed in the graph during Phase vs. Q measurement.

**Example**
To set the item to be displayed in the graph to Phase vs. Q:

```
> :DISPlay:RESult:QAnalysis:PHASe:ITEM PHAQ
```

**Compatibility**
Incompatible with existing models.
:DISPlay:RESUlt:QANalysis:PHASe:ITEM?

Response       <mode> = <CHARACTER RESPONSE DATA>
                LBER, LNB, VTHQ, PHAQ, PHAS, PHAM, PHAB, PHAV, PHAC

Function       Queries the item to be displayed in the graph during Phase vs. Q measurement.

Example        > :DISPlay:RESult:QANalysis:PHASe:ITEM?
                < PHAQ

Compatibility   Incompatible with existing models.

:DISPlay:RESult:QANalysis:PHASe:MNUMber <numeric>

Parameter      <numeric> = <DECIMAL PROGRAM DATA>
                1 to 1000               Measurement number 1 to 1000 (number of measurements during repeat measurement)

Function       Sets the measurement number for which the Phase vs. Q measurement result is displayed in a graph.

Example        To set the measurement number graph display to 100:
                > :DISPlay:RESult:QANalysis:PHASe:MNUMber 100

Compatibility   Incompatible with existing models.

:DISPlay:RESult:QANalysis:PHASe:MNUMber?

Response       <numeric> = <NR1 NUMERIC RESPONSE DATA>

Function       Queries the measurement number for which the Phase vs. Q measurement result is displayed in a graph.

Example        > :DISPlay:RESult:QANalysis:PHASe:MNUMber?
                < 100

Compatibility   Incompatible with existing models.
7.12.14.4 Measurement results and status query commands

Figure 7.12.14.4-1 Measurement results and status display field

Table 7.12.14.4-1 Measurement results and status query commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1]</td>
<td>No label (Setting of time display)</td>
<td>:SENSe:MEASure:QANalysis:TIME</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:MEASure:QANalysis:TIME?</td>
</tr>
<tr>
<td>[3]</td>
<td>Query for measurement start time</td>
<td>:SENSe:MEASure:QANalysis:STARt?</td>
</tr>
<tr>
<td>[7]</td>
<td>Query for Phase vs. Q measurement result</td>
<td>:CALCulate:DATA:QANalysis:PHAse?</td>
</tr>
<tr>
<td>[8]</td>
<td>Query for Phase vs. Q measurement status</td>
<td>:CALCulate:DATA:QANalysis:PSTatus?</td>
</tr>
</tbody>
</table>

:SENSe:MEASure:QANalysis:TIME <type>

Parameter

- <type> = <CHARACTER PROGRAM DATA>
- DTIMe: Displays the current date and time.
- STARt: Displays the measurement start time.
- ELAPsed: Displays the elapsed time based on the measurement period.

Function

Selects the Q measurement time display type.

Example

To set the Q measurement time display type to measurement start time (Start Time):

```
> :SENSe:MEASure:QANalysis:TIME STARt
```

Compatibility

Incompatible with existing models.
7.12 28G/32G bit/s ED Commands

:SENSe:MEASure:QANalysis:TIME?

Response  
<type> = <CHARACTER RESPONSE DATA>  
DTIM, STAR, ELAP

Function  
Queries the Q measurement time display type.

Example  
> :SENSe:MEASure:QANalysis:TIME?  
< STAR

Compatibility  
Incompatible with existing models.

:SENSe:MEASure:QANalysis:DTIME?

Response  
<year>,<month>,<day>,<hour>,<min>,<sec>  
<year> = <NR1 NUMERIC RESPONSE DATA>  
2000 to 2036  Year 2000 to 2036  
(month) = <NR1 NUMERIC RESPONSE DATA>  
1 to 12  January to December  
<day> = <NR1 NUMERIC RESPONSE DATA>  
1 to 31  1st to 31st  
(hour) = <NR1 NUMERIC RESPONSE DATA>  
0 to 23  0 to 23 hours  
<minute> = <NR1 NUMERIC RESPONSE DATA>  
0 to 59  0 to 59 minutes  
<second> = <NR1 NUMERIC RESPONSE DATA>  
0 to 59  0 to 59 seconds

Function  
Queries the current date and time during Q measurement.

Example  
> :SENSe:MEASure:QANalysis:DTIME?  
< 2005,12,31,23,59,59

Compatibility  
Incompatible with existing models.
Chapter 7  SCPI Commands

:SENSe:MEASure:QANalysis:STARt?

Response

<table>
<thead>
<tr>
<th>&lt;year&gt;</th>
<th>&lt;NR1 NUMERIC RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>0, 2000 to 2036</td>
<td>Year 2000 to 2036</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>&lt;month&gt;</th>
<th>&lt;NR1 NUMERIC RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>0, 1 to 12</td>
<td>January to December</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>&lt;day&gt;</th>
<th>&lt;NR1 NUMERIC RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>0, 1 to 31</td>
<td>1st to 31st</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>&lt;hour&gt;</th>
<th>&lt;NR1 NUMERIC RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 to 23</td>
<td>0 to 23 hours</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>&lt;minute&gt;</th>
<th>&lt;NR1 NUMERIC RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 to 59</td>
<td>0 to 59 minutes</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>&lt;second&gt;</th>
<th>&lt;NR1 NUMERIC RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 to 59</td>
<td>0 to 59 seconds</td>
</tr>
</tbody>
</table>

"0,0,0,0,0,0" is returned if there is no measurement start time data.

Function
Queries the measurement start time (Start Time) during Q measurement.

Example
> :SENSe:MEASure:QANalysis:STARt?
< 2005,12,31,23,59,59

Compatibility
Incompatible with existing models.

:SENSe:MEASure:QANalysis:ELAPsed?

Response

<table>
<thead>
<tr>
<th>&lt;day&gt;</th>
<th>&lt;NR1 NUMERIC RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 to 99</td>
<td>0 to 99 days</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>&lt;hour&gt;</th>
<th>&lt;NR1 NUMERIC RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 to 23</td>
<td>0 to 23 hours</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>&lt;minute&gt;</th>
<th>&lt;NR1 NUMERIC RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 to 59</td>
<td>0 to 59 minutes</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>&lt;second&gt;</th>
<th>&lt;NR1 NUMERIC RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 to 59</td>
<td>0 to 59 seconds</td>
</tr>
</tbody>
</table>

"0,0,0,0" is returned if there is no measurement elapsed time data.

Function
Queries the measurement elapsed time (Elapsed time) during Q measurement.

Example
> :SENSe:MEASure:QANalysis:ELAPsed?
< 31,23,59,59

Compatibility
Incompatible with existing models.
7.12  28G/32G bit/s ED Commands

:CALCulate:DATA:QANalysis:THReshold? <string>

Parameter

<string> = <STRING PROGRAM DATA>

<string1> For the contents of measurement data <string1> see Table 7.12.14.4-2.
<string2> For the contents of immediate data <string2> see Table 7.12.14.4-3.
<string3> For the contents of statistic data <string3> see Table 7.12.14.4-4.

<table>
<thead>
<tr>
<th>Items</th>
<th>Response</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>Top side Vth/Error Rate</td>
<td>&quot;VTHQ:TOP,1&quot;... &quot;VTHQ:TOP,1000&quot;</td>
<td>Form1</td>
</tr>
<tr>
<td>Number of measurements</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 to 1000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bottom side Vth/Error Rate</td>
<td>&quot;VTHQ:BTM,1&quot;... &quot;VTHQ:BTM,1000&quot;</td>
<td>Form1</td>
</tr>
</tbody>
</table>

The measurement result for each Vth is delimited with a comma (,) and returned.

<table>
<thead>
<tr>
<th>Items</th>
<th>Response</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q value</td>
<td>&quot;QVALue:1&quot;... &quot;QVALue:1000&quot;</td>
<td>Form2</td>
</tr>
<tr>
<td>Number of measurements</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 to 1000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Optimum BER</td>
<td>&quot;OPTBer:1&quot;... &quot;OPTBer:1000&quot;</td>
<td>Form3</td>
</tr>
<tr>
<td>Optimum Vth</td>
<td>&quot;OPTVth:1&quot;... &quot;OPTVth:1000&quot;</td>
<td>Form4</td>
</tr>
<tr>
<td>Correlation bottom</td>
<td>&quot;CORBtm:1&quot;... &quot;CORBtm:1000&quot;</td>
<td>Form2</td>
</tr>
<tr>
<td>Correlation top</td>
<td>&quot;CORTop:1&quot;... &quot;CORTop:1000&quot;</td>
<td>Form4</td>
</tr>
<tr>
<td>( \sigma ) bottom</td>
<td>&quot;SGMBtm:1&quot;... &quot;SGMBtm:1000&quot;</td>
<td>Form2</td>
</tr>
<tr>
<td>( \mu ) bottom</td>
<td>&quot;SGMTop:1&quot;... &quot;SGMTop:1000&quot;</td>
<td>Form2</td>
</tr>
<tr>
<td>( \sigma ) top</td>
<td>&quot;MUBtm:1&quot;... &quot;MUBtm:1000&quot;</td>
<td>Form4</td>
</tr>
<tr>
<td>( \mu ) top</td>
<td>&quot;MUTop:1&quot;... &quot;MUTop:1000&quot;</td>
<td>Form4</td>
</tr>
<tr>
<td>Q Equation</td>
<td>&quot;QEQL:1&quot;... &quot;QEQL:1000&quot;</td>
<td>Form2</td>
</tr>
</tbody>
</table>
## Chapter 7  SCPI Commands

### Table 7.12.14.4-4  Vth vs. Q measurement statistic data <result3>

<table>
<thead>
<tr>
<th>Items</th>
<th>&lt;Result3&gt;</th>
<th>Response Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Data</td>
<td>&quot;TOTal&quot;</td>
<td>Form5</td>
</tr>
<tr>
<td>Valid Data</td>
<td>&quot;VALid&quot;</td>
<td>Form5</td>
</tr>
<tr>
<td>Q Max</td>
<td>&quot;QMAX&quot;</td>
<td>Form2</td>
</tr>
<tr>
<td>Q Min</td>
<td>&quot;QMIN&quot;</td>
<td>Form2</td>
</tr>
<tr>
<td>Q Mean</td>
<td>&quot;QMEan&quot;</td>
<td>Form2</td>
</tr>
<tr>
<td>Q σ</td>
<td>&quot;QSGM&quot;</td>
<td>Form2</td>
</tr>
<tr>
<td>Q-5 σ</td>
<td>&quot;QSGM5&quot;</td>
<td>Form2</td>
</tr>
</tbody>
</table>

**Response**

<string> = <STRING RESPONSE DATA>

### Table 7.12.14.4-5  Response format

<table>
<thead>
<tr>
<th>Form</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vth-Q type</td>
<td>&quot;XX.XXX,Y.YYYYE-YYY&quot;</td>
<td>XX.XXX: Threshold (V)</td>
</tr>
<tr>
<td></td>
<td>&quot;-----,-------&quot;</td>
<td>When no data corresponds to the query.</td>
</tr>
<tr>
<td>%type (dB type)</td>
<td>&quot;XXX.XX&quot;</td>
<td>When 0.00 to 100.00</td>
</tr>
<tr>
<td></td>
<td>&quot;--------&quot;</td>
<td>When no data corresponds to the query.</td>
</tr>
<tr>
<td>Fraction type</td>
<td>&quot;X.XXXXXE-XXX&quot;</td>
<td>When 0.0000E–016 to 1.0000E000</td>
</tr>
<tr>
<td></td>
<td>&quot;--------&quot;</td>
<td>When no data corresponds to the query.</td>
</tr>
<tr>
<td></td>
<td>&quot;&lt; 1.0E-199&quot;</td>
<td>When E–199 or less</td>
</tr>
<tr>
<td>Voltage type</td>
<td>&quot;XX.XXXXX&quot;</td>
<td>When –4.0000 to 4.0000</td>
</tr>
<tr>
<td></td>
<td>&quot;--------&quot;</td>
<td>When no data corresponds to the query.</td>
</tr>
<tr>
<td>Integer type</td>
<td>&quot;XXXX&quot;</td>
<td>When 0 to 9999</td>
</tr>
<tr>
<td></td>
<td>&quot;----&quot;</td>
<td>When no data corresponds to the query.</td>
</tr>
</tbody>
</table>

### Function

Queries the Vth vs. Q measurement result.

### Example

To query the measurement result of the 10th time, on the top side:

```plaintext
> :CALCulate:DATA:QANalysis:THReshold? "VTHQ:TOP,10"
< "1.000,1.2345E-003", "1.100,1.2345E-004",
 "1.200,1.2345E-005",
 "1.300,1.2345E-006"
```

### Compatibility

Incompatible with existing models.
### :CALCulate:DATA:QANalysis:TSTatus?

<table>
<thead>
<tr>
<th>Response</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;&quot;</td>
<td>When no alarm exists</td>
</tr>
<tr>
<td>&quot;Sync Loss&quot;</td>
<td>Sync. Loss</td>
</tr>
<tr>
<td>&quot;Clock Loss&quot;</td>
<td>Clock Loss</td>
</tr>
<tr>
<td>&quot;CR Unlock&quot;</td>
<td>CR Unlock</td>
</tr>
<tr>
<td>&quot;Out of range&quot;</td>
<td>Out of range</td>
</tr>
<tr>
<td>&quot;Frequency NG&quot;</td>
<td>Frequency NG</td>
</tr>
<tr>
<td>&quot;Illegal Error&quot;</td>
<td>Illegal Error</td>
</tr>
<tr>
<td>&quot;Meas. Threshold exceeded&quot;</td>
<td>Number of errors exceeds the target rate</td>
</tr>
</tbody>
</table>

**Note:**
- Displays any one of the alarms.
- When two or more alarms have occurred, they are displayed in the following order:
  - Illegal Error > Frequency NG > Clock Loss (CR Unlock) > Sync Loss > Out of range

**Function:** Queries the Vth vs. Q measurement status.

**Example**

```
> :CALCulate:DATA:QANalysis:TSTatus?
< "Sync Loss"
```

**Compatibility**
- Incompatible with existing models.
Chapter 7  SCPI Commands

:CALCulate:DATA:QANalysis:PHASe? <string>

Parameter

<string> = <STRING PROGRAM DATA>

<Result1> For the contents of measurement data <result1>, see Table 7.12.14.4-6.

<Result2> For the contents of immediate data <result2>, see Table 7.12.14.4-7.

Table 7.12.14.4-6  Phase vs. Q measurement result <result1>

<table>
<thead>
<tr>
<th>Items</th>
<th>&lt;result1&gt;</th>
<th>Response Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>Top side</td>
<td>Phase −1000 to 1000 mUI</td>
<td>&quot;VTHQ:TOP,XXXXX&quot; Form1</td>
</tr>
<tr>
<td>Vth·Error Rate</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bottom side</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Vth·Error Rate</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 7.12.14.4-7  Phase vs. Q immediate data <result2>

<table>
<thead>
<tr>
<th>Items</th>
<th>&lt;result2&gt;</th>
<th>Response Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q value</td>
<td>Phase −1000 to 1000</td>
<td>&quot;QVALue:–1000&quot;... &quot;QVALue:1000&quot; Form2</td>
</tr>
<tr>
<td>Optimum BER</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Optimum Vth</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Correlation bottom</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Correlation top</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Σ bottom</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Σ top</td>
<td></td>
<td></td>
</tr>
<tr>
<td>M bottom</td>
<td></td>
<td></td>
</tr>
<tr>
<td>M top</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Q equation</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Response

<string> = <STRING RESPONSE DATA>

Table 7.12.14.4-8  Response format

<table>
<thead>
<tr>
<th>Form</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Form1 Vth·ER type</td>
<td>&quot;XX.XXX,Y.YYYYE-YYY&quot;</td>
<td>XX.XXX: Threshold (V)</td>
</tr>
<tr>
<td></td>
<td>&quot;--------,--------,--------&quot;</td>
<td>When no data corresponds to the query.</td>
</tr>
<tr>
<td>Form2 %type (dB type)</td>
<td>&quot;XXX.XX&quot;</td>
<td>When 0.00 to 100.00</td>
</tr>
<tr>
<td>Form3 Fraction type</td>
<td>&quot;X.XXXE-XXX&quot;</td>
<td>When 0.0000E–016 to 1.0000E000</td>
</tr>
<tr>
<td></td>
<td>&quot;--------&quot;</td>
<td>When no data corresponds to the query.</td>
</tr>
<tr>
<td>Form4 Voltage type</td>
<td>&quot;XX.XXX&quot;</td>
<td>When –4.0000 to 4.0000</td>
</tr>
<tr>
<td></td>
<td>&quot;--------&quot;</td>
<td>When no data corresponds to the query.</td>
</tr>
</tbody>
</table>
### Function
Queries the Phase vs. Q measurement result.

### Example
To query the TOP side measurement result:

```plaintext
> :CALCulate:DATA:QANalysis:PHASE? "VTHQ:TOP,100"
< "10,1.000,1.2345E-003","10,1.100,1.2345E-004","10,1.200,1.2345E-005"
```

### Compatibility
Incompatible with existing models.

---

### :CALCulate:DATA:QANalysis:PSTatus?

#### Response

- `<string>` = `<STRING RESPONSE DATA>`
  - `""` When no alarm exists
  - `"Sync Loss"` Sync. Loss
  - `"Clock Loss"` Clock Loss
  - `"CR Unlock"` CR Unlock
  - `"Out of range"` Out of range
  - `"Frequency NG"` Frequency NG
  - `"Illegal Error"` Illegal Error
  - `"Meas. Threshold exceeded"` Number of errors exceeds the target rate

#### Note:
- Displays any one of the alarms.
- When two or more alarms have occurred, they are displayed in the following order:
  - Illegal Error > Frequency NG > Clock Loss (CR Unlock) > Sync Loss > Out of range

#### Function
Queries the Phase vs. Q measurement status.

#### Example

```plaintext
> :CALCulate:DATA:QANalysis:PSTatus?
< "Sync Loss"
```
### 7.12.14.5 File menu setting commands

#### Table 7.12.14.5-1  File menu setting commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>:SENSe:MEASure:QANalysis:THReshold:LScale?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:MEASure:QANalysis:PHASe:LScale</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:MEASure:QANalysis:PHASe:LScale?</td>
</tr>
</tbody>
</table>

**:SYSTem:MMEMory:QANalysis:RECall <file_name>**

<table>
<thead>
<tr>
<th>Parameter</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;file_name&gt;</td>
<td>= &lt;STRING PROGRAM DATA&gt;</td>
</tr>
<tr>
<td></td>
<td>&quot;&lt;drv&gt;[&lt;dir&gt;]&lt;file&gt;&quot;</td>
</tr>
<tr>
<td></td>
<td>&lt;drv&gt; = C, D, E, F</td>
</tr>
</tbody>
</table>
|           | <dir> = <dir1>\<dir2>\...
|           | (Omitted when root directory)    |
|           | <file> = File name               |

**Function**

Opens the Q measurement result data.

**Example**

> :SYSTem:MMEMory:QANalysis:RECall "C:\Test\example"

**Compatibility**

Incompatible with existing models.
:SYSTem:MMEMory:QANalysiss:STORe
 <file_name>,<data_type>,<file_type>

Parameter
 <file_name> = <STRING PROGRAM DATA>
 "<drv>:\[<dir>]\<file>"
 <drv> = C, D, E, F
 <dir> = <dir1>\<dir2>\... (Omitted when root directory)
 <file> = File name
 <data_type> = <CHARACTER PROGRAM DATA>
 VTQ Vth vs. Q Result
 PSQ Phase vs. Q Result
 <file_type> = <CHARACTER PROGRAM DATA>
 BIN Binary file
 CSV CSV file
 TXT Text file

Function
 Saves the Q measurement result data.

Note:
 The settings will not be read from the saved file if the file name is changed.

Example
 Saves the Vth vs. Q measurement result data of Q measurement,
 specifying the save destination, file name, and file format:
 > :SYSTem:MMEMory:QANalysiss:STORe
 "C:\Test\example",VTQ,CSV

Compatibility
 Incompatible with existing models.

:SYSTem:PRINT:QANalysiss <list>

Parameter
 <list> = <CHARACTER PROGRAM DATA>
 VTQ Prints Vth vs. Q measurement results.
 PSQ Prints Phase vs. Q measurement results.

Function
 Prints the Q measurement result data.

Example
 To print the Vth vs. Q measurement results:
 > :SYSTem:PRINT:QANalysiss VTQ

Compatibility
 Incompatible with existing models.
Chapter 7  SCPI Commands

:SENSe:MEASure:QANalysISS:THReshold:LSCale <scale>

Parameter  
<scale> = <CHARACTER PROGRAM DATA>
  LIINear Linear display
  LOG Log display

Function  
Sets the Q value display scale during Vth vs. Q measurement to Linear or Log display.

Example  
To set the display scale to Log display:
> :SENSe:MEASure:QANalysISS:THReshold:LSCale LOG

Compatibility  
Incompatible with existing models.

:SENSe:MEASure:QANalysISS:THReshold:LSCale?

Response  
<scale> = <CHARACTER RESPONSE DATA>
  LIIN, LOG

Function  
Queries whether the Q value display scale during Vth vs. Q measurement is set to Linear or Log display.

Example  
> :SENSe:MEASure:QANalysISS:THReshold:LSCale?
  < LOG

Compatibility  
Incompatible with existing models.

:SENSe:MEASure:QANalysISS:PHASe:LSCale <scale>

Parameter  
<scale> = <CHARACTER PROGRAM DATA>
  LIINear Linear display
  LOG Log display

Function  
Sets the Q value display scale during Phase vs. Q measurement to Linear or Log display.

Example  
To set the display scale to Log display:
> :SENSe:MEASure:QANalysISS:PHASe:LSCale LOG

Compatibility  
Incompatible with existing models.

:SENSe:MEASure:QANalysISS:PHASe:LSCale?

Response  
<scale> = <CHARACTER RESPONSE DATA>
  LIIN, LOG

Function  
Queries whether the Q value display scale during Phase vs. Q measurement is set to Linear or Log display.

Example  
> :SENSe:MEASure:QANalysISS:PHASe:LSCale?
  < LOG

Compatibility  
Incompatible with existing models.
7.12 28G/32G bit/s ED Commands

7.12.15 PAM BER measurement

This section describes PAM BER measurement commands.

Figure 7.12.15-1  PAM BER Measurement Screen

Table 7.12.15-1  PAM BER Measurement Commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>:DISPlay:RESult:PAMBer:TABSelect?</td>
</tr>
</tbody>
</table>
### :DISPlay:RESult:PAMBer:TABSelect \(<\text{tab}>\)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>(&lt;\text{tab}&gt;) = (&lt;\text{CHARACTER PROGRAM DATA}&gt;)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PBSerial</td>
<td>Selects the 3 Eye Serial tab.</td>
</tr>
<tr>
<td>PBParallel</td>
<td>Selects the 3 Eye Parallel tab.</td>
</tr>
</tbody>
</table>

| Function       | Selects the PAM BER measurement type. Selects serial measurement using 3 Eye Serial or the 3 Eye Parallel measurement tab using 3ch. |

<table>
<thead>
<tr>
<th>Example</th>
<th>To select the 3Lvl Serial tab:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>&gt; :DISPlay:RESult:PAMBerTABSelect PBSerial</td>
</tr>
</tbody>
</table>

| Compatibility  | Incompatible with existing models. |

### :DISPlay:RESult:PAMBer:TABSelect? 

<table>
<thead>
<tr>
<th>Response</th>
<th>(&lt;\text{tab}&gt;) = (&lt;\text{CHARACTER RESPONSE DATA}&gt;)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PBS, PBP</td>
<td></td>
</tr>
</tbody>
</table>

| Function       | Queries the tab displayed for PAM BER measurement. |

<table>
<thead>
<tr>
<th>Example</th>
<th>&gt; :DISPlay:RESult:PAMBerTABSelect?</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>&lt; PBS</td>
</tr>
</tbody>
</table>

| Compatibility  | Incompatible with existing models. |

7.12.15.1 3 Eye Serial Measurement Commands

![Figure 7.12.15.1-1 3 Eye Serial Measurement Screen](image)

Table 7.12.15.1-1 3 Eye Serial Measurement Commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>::SENSe:MEASure:PBSerial:MODE?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>::SENSe:MEASure:PBSerial:PERiod?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>::SENSe:PATTern:PBSerial:TYPE?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>::SENSe:MEASure:PBSerial:ASEarch?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>::SENSe:MEASure:PBSerial:SLOT?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>::SENSe:MEASure:PBSerial:PAMSet?</td>
</tr>
<tr>
<td>[7]</td>
<td>Middle Eye Phase Tracking</td>
<td>::SENSe:MEASure:PBSerial:MEPHase</td>
</tr>
<tr>
<td></td>
<td></td>
<td>::SENSe:MEASure:PBSerial:MEPHase?</td>
</tr>
</tbody>
</table>
## Table 7.12.15.1-1  3 Eye Serial Measurement Commands (Cont’d)

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>:SENSe:MEASure:PBSerial:DATA:THReshold?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:MEASure:PBSerial:CLOCk:DELay?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:MEASure:PBSerial:TIME?</td>
</tr>
<tr>
<td></td>
<td>Query for measurement start time</td>
<td>:SENSe:MEASure:PBSerial:TIME:STARt?</td>
</tr>
<tr>
<td></td>
<td>Query for measurement elapsed time</td>
<td>:SENSe:MEASure:PBSerial:TIME:ELAPsed?</td>
</tr>
</tbody>
</table>

### :SENSe:MEASure:PBSerial:MODE <mode >

**Parameter**

<mode> = <CHARACTER PROGRAM DATA>

- REPeat: Executes the measurement repeatedly.
- SINGle: Executes the measurement once.

**Function**

Selects the repeated measurement setting for PAM BER measurement using 3 Eye Serial.

**Example**

To select the repeated measurement:

```plaintext
> :SENSe:MEASure:PBSerial:MODE REPeat
```

**Compatibility**

Incompatible with existing models.

### :SENSe:MEASure:PBSerial:MODE?

**Parameter**

<mode> = <CHARACTER RESPONSE DATA>

- REP, SING

**Function**

Queries the measurement repeat setting.

**Example**

```plaintext
> :SENSe:MEASure:PBSerial:MODE?
< REP
```

**Compatibility**

Incompatible with existing models.
:SENSe:MEASure:PBSerial:PERiod <hour>,<min>,<second>

Parameter

<hour>,<min>,<second> = <DECIMAL NUMERIC PROGRAMDATA>

- <hour> 0 to 23 0 to 23 hours/hour Step
- <min> 0 to 59 0 to 59 minutes/minute Step
- <second> 0 to 59 0 to 59 seconds/second Step

Function
Sets the time for measuring PAM BER in 3 Eye Serial mode.

Example
To set the measurement time to one minute:
> :SENSe:MEASure:PBSerial:PERiod 0,1,0

Compatibility
Incompatible with existing models.

:SENSe:MEASure:PBSerial:PERiod?

Response

<hour>,<min>,<second> = <NR1 NUMERIC RESPONSE DATA>

- <hour> 0 to 23 0 to 23 hours/hour Step
- <min> 0 to 59 0 to 59 minutes/minute Step
- <second> 0 to 59 0 to 59 seconds/second Step

Function
Queries the time for measuring PAM BER in 3 Eye Serial mode.

Example
> :SENSe:MEASure:PBSerial:PERiod?
< 0,1,0

Compatibility
Incompatible with existing models.

:SENSe:PATTern:PBSerial:TYPE <type>

Parameter

<type> = <STRING PROGRAM DATA>
"file name"

Specify the file name according to “Appendix F” in the MU183040B Operation Manual.

The following are example settings:
"PRBS7"  PRBS7
"PRBS9"  PRBS9
"PRBS10" PRBS10
"PRBS11" PRBS11
"PRBS15" PRBS15
"PRBS20" PRBS20
"PRBS13Q" PRBS13Q
"PRQS10"  PRQS10

Function
Selects the test pattern.

Example
To set the test pattern to GrayPRBS15:
> :SENSe:PATTern:PBSerial:TYPE "GrayPRBS15"

Compatibility
Incompatible with existing models.
Chapter 7  SCPI Commands

:SENSe:PATTern:PBSerial:TYPE?

Response  
	<type> = <STRING RESPONSE DATA>

"file name"

For the file names, refer to “Appendix F” in the MU183040B Operation Manual.

Function  
Queries the test pattern.

Example  
> :SENSe:PATTern:PBSerial:TYPE?
< "GrayPRBS15"

Compatibility  
Incompatible with existing models.

:SENSe:MEASure:PBSerial:ASEarch <item>

Parameter  
	<slot> = <CHARACTER PROGRAM DATA>

OFF       Auto Search OFF
PAMFine   PAM4 Fine mode
PAMCoarse PAM4 Coarse mode

Function  
Sets Auto Search execution for 3 Eye Serial PAM BER measurement.

Example  
To set Auto Search to PAM4 Fine mode:
> :SENSe:MEASure:PBSerial:ASEarch PAMFine

Compatibility  
Incompatible with existing models.

:SENSe:MEASure:PBSerial:ASEarch?

Response  
	<item> = <CHARACTER RESPONSE DATA>

OFF, PAMF, PAMC

Function  
Queries the Auto Search execution mode for 3 Eye Serial PAM BER measurement.

Example  
> :SENSe:MEASure:PBSerial:ASEarch?
< PAMF

Compatibility  
Incompatible with existing models.
**:SENSe:MEASure:PBSerial:SLOT <slot>,<interface>[,<unit>]**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;slot&gt;</td>
<td>= &lt;DECIMAL PROGRAM DATA&gt; 1 to 6: Slots No.1 to 6; 1 to 6 when using the MP1800A, 1 to 4 when using the MT1810A</td>
</tr>
<tr>
<td>&lt;interface&gt;</td>
<td>= &lt;DECIMAL NUMERIC PROGRAM DATA&gt; 1 to 4: Data1 to 4</td>
</tr>
<tr>
<td>[&lt;unit&gt;]</td>
<td>= &lt;DECIMAL NUMERIC PROGRAM DATA&gt; 1 to 4: Mainframe No.1 to 4</td>
</tr>
</tbody>
</table>

When using two or more MT1810A units in serial connection, specify the mainframe number. Can be omitted. Mainframe No. 1 is specified when omitted.

**Function**
Selects the target slot for 3 Eye Serial PAM BER measurement.

**Example**
To set the measurement target to slot No. 4 and Data1:
> :SENSe:MEASure:PBSerial:SLOT 4,1

**Compatibility**
Incompatible with existing models.

:**SENSe:MEASure:PBSerial:SLOT? [<unit>]**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>[&lt;unit&gt;]</td>
<td>= &lt;DECIMAL NUMERIC PROGRAM DATA&gt; 1 to 4: Mainframe No.1 to 4</td>
</tr>
</tbody>
</table>

When using two or more MT1810A units in serial connection, specify the mainframe number. Can be omitted. Mainframe No. 1 is specified when omitted.

**Response**

| <slot>      | = <NR1 NUMERIC RESPONSE DATA> 1 to 6: Slots No. 1 to 6                                         |
| <interface> | = <NR1 NUMERIC RESPONSE DATA> 1 to 4: Data1 to 4                                              |

**Function**
Queries the target slot and data interface for 3 Eye Serial PAM BER measurement.

**Example**
> :SENSe:MEASure:PBSerial:SLOT? 4,1

**Compatibility**
Incompatible with existing models.
Chapter 7  SCPI Commands

:SENSe:MEASure:PBSerial:PAMSet <threshold>,<boolean>

Parameter
<threshold> = <CHARACTER PROGRAM DATA>
  TOP     Top threshold
  MIDDle  Middle threshold
  BOTTom  Bottom threshold
<boolean> = <BOOLEAN PROGRAM DATA>
  ON, 1   ON
  OFF, 0  OFF

Function
Selects the threshold to be measured in 3 Eye Serial PAM BER measurement.

Example
To set the Top threshold as the measurement target:
> :SENSe:MEASure:PBSerial:PAMSet TOP,ON

Compatibility
Incompatible with existing models.

:SENSe:MEASure:PBSerial:PAMSet? <threshold>

Parameter
<threshold> = <CHARACTER PROGRAM DATA>
  TOP     Top threshold
  MIDDle  Middle threshold
  BOTTom  Bottom threshold

Response
<boolean> = <NR1 NUMERIC RESPONSE DATA>
  1   ON
  0   OFF

Function
Queries the measurement target threshold.

Example
To query whether the Top threshold is the target of measurement:
> :SENSe:MEASure:PBSerial:PAMSet? TOP
< 1

Compatibility
Incompatible with existing models.

:SENSe:MEASure:PBSerial:MEPHase <boolean>

Parameter
<boolean> = <BOOLEAN PROGRAM DATA>
  ON, 1   ON
  OFF, 0  OFF

Function
Sets whether to use the Middle Eye phase value as a reference for BER measurement in 3 Eye Serial PAM BER measurement.

Example
To use the Middle Eye phase value in PAM4 BER measurement:
> :SENSe:MEASure:PBSerial:MEPHase ON

Compatibility
Incompatible with existing models.
:SENSe:MEASure:PBSerial:MEPHase? <threshold>

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;boolean&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ON</td>
</tr>
<tr>
<td>0</td>
<td>OFF</td>
</tr>
</tbody>
</table>

Function: Queries whether the Middle Eye phase value is used as a reference in 3 Eye Serial PAM BER measurement.

Example:
```
> :SENSe:MEASure:PBSerial:MEPHase?
< 1
```

Compatibility: Incompatible with existing models.

:SENSe:MEASure:PBSerial:STARt

Function: Starts 3 Eye Serial PAM BER measurement.

Example:
```
> :SENSe:MEASure:PBSerial:STARt
```

Compatibility: Incompatible with existing models.

:SENSe:MEASure:PBSerial:STOP

Function: Stops 3 Eye Serial PAM BER measurement.

Example:
```
> :SENSe:MEASure:PBSerial:STOP
```

Compatibility: Incompatible with existing models.

:SENSe:MEASure:PBSerial:STATe?

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;numeric&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Measurement stopped.</td>
</tr>
<tr>
<td>1</td>
<td>Measurement in progress</td>
</tr>
</tbody>
</table>

Function: Queries the measurement progress status for 3 Eye Serial PAM BER measurement.

Example:
```
> :SENSe:MEASure:PBSerial:STATe?
< 1
```

Compatibility: Incompatible with existing models.
## :CALCulate:DATA:PBSerial? <string>

**Parameter**

\[
\text{<string>} = \text{<STRING PROGRAM DATA>}
\]

"TOP:<result>"
"MIDDle:<result>"
"BOTTom:<result>"
"TOTal:<result>"

### Table 7.12.15.1-2  3 Eye Serial Measurement Results <result>

<table>
<thead>
<tr>
<th>Item</th>
<th>&lt;result&gt;</th>
<th>Response Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>Error Count</td>
<td>EC</td>
<td>Form1</td>
</tr>
<tr>
<td>Error Rate</td>
<td>ER</td>
<td>Form2</td>
</tr>
<tr>
<td>Frequency(Hz)</td>
<td>FREQuency</td>
<td>Form3</td>
</tr>
<tr>
<td>Data Threshold</td>
<td>THReshold</td>
<td>Form4</td>
</tr>
<tr>
<td>Data Phase</td>
<td>PHASe</td>
<td>Form5</td>
</tr>
</tbody>
</table>

### Response

\[
\text{<string>} = \text{<STRING RESPONSE DATA>}
\]

### Table 7.12.15.1-3  Response Format

<table>
<thead>
<tr>
<th>Type</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Form1</td>
<td>&quot;XXXXXXX&quot;</td>
<td>0 to 9999999</td>
</tr>
<tr>
<td>Integer type</td>
<td>&quot;X.XXXE+XX&quot;</td>
<td>1.0000E07 to 9.9999E17</td>
</tr>
<tr>
<td></td>
<td>&quot;-----&quot;</td>
<td>When no data corresponds to the query</td>
</tr>
<tr>
<td>Form2</td>
<td>&quot;X.XXXE-XX&quot;</td>
<td>0.0001E–18 to 1.0000E00</td>
</tr>
<tr>
<td>Fraction type</td>
<td>&quot;-----&quot;</td>
<td>When no data corresponds to the query</td>
</tr>
<tr>
<td>Form3</td>
<td>&quot;XXXXXXXXXX&quot;</td>
<td>For 0 to MAX (Hz)</td>
</tr>
<tr>
<td>Frequency type</td>
<td>&quot;-----&quot;</td>
<td>When no data corresponds to the query</td>
</tr>
<tr>
<td>Form4</td>
<td>&quot;X.XXX.XXX.XXX&quot;</td>
<td>For –4.000 to +4.000</td>
</tr>
<tr>
<td>Voltage type</td>
<td>&quot;-----&quot;</td>
<td>When no data corresponds to the query</td>
</tr>
<tr>
<td>Form5</td>
<td>&quot;XXXX&quot;</td>
<td>For –1000 to +1000, –10000 to +10000</td>
</tr>
<tr>
<td>Phase type</td>
<td>&quot;-----&quot;</td>
<td>When no data corresponds to the query</td>
</tr>
</tbody>
</table>

**Function**
Queries the ED 1ch results for PAM BER measurement.

**Example**

To query the total error rate result for 3 Eye Serial measurement:

> :CALCulate:DATA:PBSerial? "TOTal:ER"

< "1.2345E-03"

**Compatibility**
Incompatible with existing models.
:SENSe:MEASure:PBSerial:DATA:THReshold
<threshold>,<port>,<numeric>

Parameter

<threshold> = <CHARACTER PROGRAM DATA>
  TOP       Top level
  MIDDle   Middle level
  BOTTom   Bottom level

<port> = <CHARACTER PROGRAM DATA>
  DATA     Selects Data.
  XDATa    Selects XData.

<numeric> = <DECIMAL NUMERIC PROGRAM DATA>
  –3.500 to 3.300  –3.500 to 3.300 V/0.001 V Step

Function
Sets input port and measurement threshold for Top, Middle, and Bottom levels respectively in the 3 Eye Serial PAM BER measurement.

Example
To select Data and specify 0.100 V for the measurement threshold of Top level:
> :SENSe:MEASure:PBSerial:DATA:THReshold TOP,DATA,0.100

Compatibility
Incompatible with existing models.

:SENSe:MEASure:PBSerial:DATA:THReshold? <threshold>,<port>

Parameter

<threshold> = <CHARACTER PROGRAM DATA>
  TOP       Top level
  MIDDle   Middle level
  BOTTom   Bottom level

<port> = <CHARACTER PROGRAM DATA>
  DATA     Selects Data.
  XDATa    Selects XData.

Response
<numeric> = <NR1 NUMERIC RESPONSE DATA>
  –3.500 to 3.300  –3.500 to 3.300 V/0.001 V Step

Function
Queries measurement threshold for Top, Middle, and Bottom levels respectively in the 3 Eye Serial PAM BER measurement.

Example
To query threshold of Top level, data input:
< 0.100

Compatibility
Incompatible with existing models.
**Chapter 7  SCPI Commands**

### :SENSe:MEASure:PBSerial:CLOCk:DELay <threshold>,<numeric>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;threshold&gt;</td>
<td>&lt;CHARACTER PROGRAM DATA&gt;</td>
</tr>
<tr>
<td>TOP</td>
<td>Top level</td>
</tr>
<tr>
<td>MIDDle</td>
<td>Middle level</td>
</tr>
<tr>
<td>BOTTtom</td>
<td>Bottom level</td>
</tr>
<tr>
<td>&lt;numeric&gt;</td>
<td>&lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
</tr>
<tr>
<td>–1000 to 1000</td>
<td>–1000 to 1000 mUI/2 mUI Step</td>
</tr>
</tbody>
</table>

**Function**
Sets Delay for Top, Middle, and Bottom levels respectively in the 3 Eye Serial PAM BER measurement.

**Example**
To set Delay of Top level to 300 mUI: 
```
> :SENSe:MEASure:PBSerial:CLOCk:DELay TOP,300
```

**Compatibility**
Incompatible with existing models.

### :SENSe:MEASure:PBSerial:CLOCk:DELay? <threshold>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;threshold&gt;</td>
<td>&lt;CHARACTER PROGRAM DATA&gt;</td>
</tr>
<tr>
<td>TOP</td>
<td>Top level</td>
</tr>
<tr>
<td>MIDDle</td>
<td>Middle level</td>
</tr>
<tr>
<td>BOTTtom</td>
<td>Bottom level</td>
</tr>
</tbody>
</table>

**Response**
<numeric> = <NR1 NUMERIC RESPONSE DATA> 

|  –1000 to 1000 | –1000 to 1000 mUI/2 mUI Step |

**Function**
Queries Delay for Top, Middle, and Bottom levels respectively in the 3 Eye Serial PAM BER measurement.

**Example**
To query the measurement threshold of Top level, data input:
```
< 300
```

**Compatibility**
Incompatible with existing models.

### :SENSe:MEASure:PBSerial:TIME <type>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;type&gt;</td>
<td>&lt;CHARACTER PROGRAM DATA&gt;</td>
</tr>
<tr>
<td>DTIMe</td>
<td>Displays the current date and time.</td>
</tr>
<tr>
<td>STARt</td>
<td>Displays the measurement start time.</td>
</tr>
<tr>
<td>ELAPsed</td>
<td>Displays the elapsed time based on the measurement period.</td>
</tr>
</tbody>
</table>

**Function**
Selects the measurement time display for 3 Eye Serial PAM BER measurement.

**Example**
To display elapsed measurement time:
```
> :SENSe:MEASure:PBSerial:TIME ELAPsed
```

**Compatibility**
Incompatible with existing models.
7.12 28G/32G bit/s ED Commands

:SENSe:MEASure:PBSerial:TIME?

| Response | <type> = <CHARACTER RESPONSE DATA>  
DTIM, STAR, ELAP |
| Function | Queries the measurement time display for 3 Eye Serial PAM BER measurement. |
| Example | > :SENSe:MEASure:PBSerial:TIME?  
< ELAP |
| Compatibility | Incompatible with existing models. |

:SENSe:MEASure:PBSerial:TIME:DTIMe?

| Response | <year>,<month>,<day>,<hour>,<min>,<second>  
<year> = <NR1 NUMERIC RESPONSE DATA>  
2000 to 2036 Year 2000 to 2036  
<month> = <NR1 NUMERIC RESPONSE DATA>  
1 to 12 January to December  
<day> = <NR1 NUMERIC RESPONSE DATA>  
1 to 31 1st to 31st  
<hour> = <NR1 NUMERIC RESPONSE DATA>  
0 to 23 0 to 23 hours  
<minute> = <NR1 NUMERIC RESPONSE DATA>  
0 to 59 0 to 59 minutes  
<second> = <NR1 NUMERIC RESPONSE DATA>  
0 to 59 0 to 59 seconds |
| Function | Queries the current time for 3 Eye Serial PAM BER measurement. |
| Example | > :SENSe:MEASure:PBSerial:TIME:DTIMe?  
< 2016,12,31,23,59,59 |
| Compatibility | Incompatible with existing models. |
Chapter 7  SCPI Commands

:SENSe:MEASure:PBSerial:TIME:STARt?

Response  

<y>ear>,<m>onth>,<d>ay>,<h>our>,<m>in>,<s>econd>

<y>ear> = <NR1 NUMERIC RESPONSE DATA>
2000 to 2036 Year 2000 to 2036
1 to 12 January to December
</d>ay> = <NR1 NUMERIC RESPONSE DATA>
1 to 31 1st to 31st
<h>our> = <NR1 NUMERIC RESPONSE DATA>
0 to 23 0 to 23 hours
<m>in>ute> = <NR1 NUMERIC RESPONSE DATA>
0 to 59 0 to 59 minutes
<s>econd> = <NR1 NUMERIC RESPONSE DATA>
0 to 59 0 to 59 seconds

Function  Queries the start time for 3 Eye Serial PAM BER measurement.

Example  > :SENSe:MEASure:PBSerial:TIME:STARt?
< 2016,12,31,23,59,59

Compatibility  Incompatible with existing models.

:SENSe:MEASure:PBSerial:TIME:ELAPsed?

Response  

<h>our>,<m>in>,<s>econd>

<h>our> = <NR1 NUMERIC RESPONSE DATA>
0 to 23 0 to 23 hours
<m>in>ute> = <NR1 NUMERIC RESPONSE DATA>
0 to 59 0 to 59 minutes
<s>econd> = <NR1 NUMERIC RESPONSE DATA>
0 to 59 0 to 59 seconds

Function  Queries the elapsed time for 3 Eye Serial PAM BER measurement.

Example  > :SENSe:MEASure:PBSerial:TIME:ELAPsed?
< 23,59,59

Compatibility  Incompatible with existing models.
### 7.12 28G/32G bit/s ED Commands

#### 7.12.15.2 3 Eye Parallel Measurement Commands

![3 Eye Parallel Measurement Screen](image-url)

**Figure 7.12.15.2-1 3 Eye Parallel Measurement Screen**

**Table 7.12.15.2-1 3 Eye Parallel Measurement Commands**

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>:SENSe:MEASure:PBParallel:MODE?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:MEASure:PBParallel:PERiod?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:PATTern:PBParallel:TYPE?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:MEASure:PBParallel:ASEarch?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:MEASure:PBParallel:SLOT?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:MEASure:PBParallel:PAMSet?</td>
</tr>
</tbody>
</table>
### Table 7.12.15.2-1  3 Eye Parallel Measurement Commands (Cont’d)

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>:SENSe:MEASure:PBParallel:DATA:THReshold?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:MEASure:PBParallel:CLOCk:DELay?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:MEASure:PBParallel:TIME?</td>
</tr>
<tr>
<td></td>
<td>Query for measurement start time</td>
<td>:SENSe:MEASure:PBParallel:TIME:STARt?</td>
</tr>
<tr>
<td></td>
<td>Query for measurement elapsed time</td>
<td>:SENSe:MEASure:PBParallel:TIME:ELAPsed?</td>
</tr>
</tbody>
</table>

### :SENSe:MEASure:PBParallel:MODE <mode>

- **Parameter**
  - <mode> = <CHARACTER PROGRAM DATA>
  - REPeat: Executes the measurement repeatedly.
  - SINGle: Executes the measurement once.
- **Function**
  - Selects the repeated measurement setting for PAM BER measurement using 3 Eye Parallel.
- **Example**
  - To select the repeated measurement:
    ```
    > :SENSe:MEASure:PBParallel:MODE REPeat
    ```
- **Compatibility**
  - Incompatible with existing models.

### :SENSe:MEASure:PBParallel:MODE?

- **Response**
  - <mode> = <CHARACTER RESPONSE DATA>
  - REP, SING
- **Function**
  - Queries the measurement repeat setting.
- **Example**
  - To query the measurement repeat setting:
    ```
    > :SENSe:MEASure:PBParallel:MODE?
    < REP
    ```
- **Compatibility**
  - Incompatible with existing models.

### :SENSe:MEASure:PBParallel:PERiod <hour>,<min>,<second>

- **Parameter**
  - <hour>,<min>,<second> = <DECIMAL NUMERIC PROGRAMDATA>
  - <hour> 0 to 23 0 to 23 hours/hour Step
  - <min> 0 to 59 0 to 59 minutes/minute Step
  - <second> 0 to 59 0 to 59 seconds/second Step
- **Function**
  - Sets the measurement time using 3 Eye Parallel.
- **Example**
  - To set the measurement time to one minute:
    ```
    > :SENSe:MEASure:PBParallel:PERiod 0,1,0
    ```
- **Compatibility**
  - Incompatible with existing models.
:SENSe:MEASure:PBParallel:PERiod?

Response  

(hour),<min>,<second> = <NR1 NUMERIC RESPONSE DATA>

(hour)  0 to 23  0 to 23 hours/hour Step

(min)  0 to 59  0 to 59 minutes/minute Step

(second)  0 to 59  0 to 59 seconds/second Step

Function Queries the measurement time using 3 Eye Parallel.

Example  

> :SENSe:MEASure:PBParallel:PERiod?

< 0,1,0

Compatibility Incompatible with existing models.

:SENSe:PATTern:PBParallel:TYPE <type>

Parameter  

(type) = <STRING PROGRAM DATA>

"PRBS7"  PRBS7

"PRBS9"  PRBS9

"PRBS10"  PRBS10

"PRBS11"  PRBS11

"PRBS15"  PRBS15

"PRBS20"  PRBS20

"PRBS13Q"  PRBS13Q

"PRQS10"  PRQS10

Function Selects the test pattern.

Example  

To set the test pattern to PRBS13Q:

> :SENSe:PATTern:PBParallel:TYPE "PRBS13Q"

Compatibility Incompatible with existing models.

:SENSe:PATTern:PBParallel:TYPE?

Response  

(type) = <STRING RESPONSE DATA>

"PRBS7", "PRBS9", "PRBS10", "PRBS11", "PRBS15", "PRBS20",

"PRBS13Q", "PRQS10"

Function Queries the test pattern.

Example  

> :SENSe:PATTern:PBParallel:TYPE?

< "PRBS13Q"

Compatibility Incompatible with existing models.
Chapter 7  SCPI Commands

:SENSe:MEASure:PBParallel:ASEarch <item>

Parameter

<table>
<thead>
<tr>
<th>&lt;slot&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>= &lt;CHARACTER PROGRAM DATA&gt;</td>
</tr>
<tr>
<td>OFF</td>
</tr>
<tr>
<td>Auto Search OFF</td>
</tr>
<tr>
<td>PAMFine</td>
</tr>
<tr>
<td>PAM4 Fine mode</td>
</tr>
<tr>
<td>PAMCoarse</td>
</tr>
<tr>
<td>PAM4 Coarse mode</td>
</tr>
</tbody>
</table>

Function  Sets Auto Search execution for 3 Eye Parallel PAM BER measurement.

Example

To set Auto Search to PAM4 Fine mode:

> :SENSe:MEASure:PBParallel:ASEarch PAMFine

Compatibility  Incompatible with existing models.

:SENSe:MEASure:PBParallel:ASEarch?

Response

<table>
<thead>
<tr>
<th>&lt;item&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>= &lt;CHARACTER RESPONSE DATA&gt;</td>
</tr>
<tr>
<td>OFF, PAMF, PAMC</td>
</tr>
</tbody>
</table>

Function  Queries the Auto Search execution mode for 3 Eye Parallel PAM BER measurement.

Example

> :SENSe:MEASure:PBParallel:ASEarch?

< PAMF

Compatibility  Incompatible with existing models.

:SENSe:MEASure:PBParallel:SLOT

<threshold>,<slot>,<interface>[,.<unit>]

Parameter

<table>
<thead>
<tr>
<th>&lt;threshold&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>= &lt;CHARACTER PROGRAM DATA&gt;</td>
</tr>
<tr>
<td>TOP</td>
</tr>
<tr>
<td>Top threshold</td>
</tr>
<tr>
<td>MIDDle</td>
</tr>
<tr>
<td>Middle threshold</td>
</tr>
<tr>
<td>BOTTtom</td>
</tr>
<tr>
<td>Bottom threshold</td>
</tr>
<tr>
<td>&lt;slot&gt;</td>
</tr>
<tr>
<td>= &lt;DECIMAL PROGRAM DATA&gt;</td>
</tr>
<tr>
<td>1 to 6</td>
</tr>
<tr>
<td>Slots No. 1 to 6</td>
</tr>
<tr>
<td>1 to 6 when using the MP1800A, 1 to 4 when using the MT1810A</td>
</tr>
<tr>
<td>&lt;interface&gt;</td>
</tr>
<tr>
<td>= &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
</tr>
<tr>
<td>1 to 4</td>
</tr>
<tr>
<td>Data1 to 4</td>
</tr>
<tr>
<td>[&lt;unit&gt;]</td>
</tr>
<tr>
<td>= &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
</tr>
<tr>
<td>1 to 4</td>
</tr>
<tr>
<td>Mainframe No. 1 to 4</td>
</tr>
</tbody>
</table>

When using two or more MT1810A units in serial connection, specify the mainframe number.

Can be omitted. Mainframe No. 1 is specified when omitted.

Function  Selects the ED (slot and data interface) for measuring individual thresholds in 3 Eye Parallel PAM BER measurement.

Example

To measure the Top threshold using slot No. 4 and Data1:

> :SENSe:MEASure:PBParallel:SLOT TOP,4,1

Compatibility  Incompatible with existing models.
:SENSe:MEASure:PBParallel:SLOT? <threshold>[,<unit>]

Parameter

<threshold> = <CHARACTER PROGRAM DATA>
TOP   Top threshold
MIDDle  Middle threshold
BOTTom  Bottom threshold
[unit>] = <DECIMAL NUMERIC PROGRAM DATA>
1 to 4   Mainframe No. 1 to 4

When using two or more MT1810A units in serial connection, specify the mainframe number.
Can be omitted. Mainframe No. 1 is specified when omitted.

Response

[slot] = <NR1 NUMERIC RESPONSE DATA>
1 to 6   Slots No. 1 to 6
1 to 6 when using the MP1800A, 1 to 4 when using the MT1810A

<interface> = <NR1 NUMERIC RESPONSE DATA>
1 to 4   Data1 to 4

Function

Queries the ED slot and data interface forming the measurement target for each threshold in 3 Eye Parallel PAM BER measurement.

Example

To query the ED for which the Top threshold is the target of measurement:
> :SENSe:MEASure:PBParallel:SLOT? TOP
< 4,1

Compatibility

Incompatible with existing models.

:SENSe:MEASure:PBParallel:PAMSet <threshold>,<boolean>

Parameter

<threshold> = <CHARACTER PROGRAM DATA>
TOP   Top threshold
MIDDle  Middle threshold
BOTTom  Bottom threshold
<boolean> = <BOOLEAN PROGRAM DATA>
ON, 1   ON
OFF, 0  OFF

Function

Selects the threshold to be measured in 3 Eye Parallel PAM BER measurement.

Example

To set the Top threshold as the measurement target:
> :SENSe:MEASure:PBParallel:PAMSet TOP,ON

Compatibility

Incompatible with existing models.
### Chapter 7  SCPI Commands

**:SENSe:MEASure:PBParallel:PAMSet? <threshold>**

**Parameter**

- `<threshold>` = `<CHARACTER PROGRAM DATA>`
  - TOP        Top threshold
  - MIDDle    Middle threshold
  - BOTTom    Bottom threshold

**Response**

- `<boolean>` = `<NR1 NUMERIC RESPONSE DATA>`
  - 1   ON
  - 0   OFF

**Function**

Queries the measurement target threshold.

**Example**

To query whether the Top threshold is the target of measurement:

```
> :SENSe:MEASure:PBParallel:PAMSet? TOP
< 1
```

**Compatibility**

Incompatible with existing models.

---

**:SENSe:MEASure:PBParallel:STARt**

**Function**

Starts 3 Eye Parallel PAM BER measurement.

**Example**

```
> :SENSe:MEASure:PBParallel:STARt
```

**Compatibility**

Incompatible with existing models.

---

**:SENSe:MEASure:PBParallel:STOP**

**Function**

Stops 3 Eye Parallel PAM BER measurement.

**Example**

```
> :SENSe:MEASure:PBParallel:STOP
```

**Compatibility**

Incompatible with existing models.

---

**:SENSe:MEASure:PBParallel:STATE?**

**Response**

- `<numeric>` = `<NR1 NUMERIC RESPONSE DATA>`
  - 0   Measurement stopped.
  - 1   Measurement in progress

**Function**

Queries the measurement progress status for 3 Eye Parallel PAM BER measurement.

**Example**

```
> :SENSe:MEASure:PBParallel:STATE?
< 1
```

**Compatibility**

Incompatible with existing models.
:CALCulate:DATA:PBParallel? <string>

Parameter

<string> = <STRING PROGRAM DATA>
"TOP:<result>"
"MIDDle:<result>"
"BOTTom:<result>"
"TOTal:<result>"

Table 7.12.15.2-2 3 Eye Parallel Measurement Results <result>

<table>
<thead>
<tr>
<th>Items</th>
<th>&lt;result&gt;</th>
<th>Response Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>Error Count</td>
<td>EC</td>
<td>Form1</td>
</tr>
<tr>
<td>Error Rate</td>
<td>ER</td>
<td>Form2</td>
</tr>
<tr>
<td>Frequency (Hz)</td>
<td>FREQuency</td>
<td>Form3</td>
</tr>
<tr>
<td>Data Threshold</td>
<td>THReshold</td>
<td>Form4</td>
</tr>
<tr>
<td>Data Phase</td>
<td>PHASe</td>
<td>Form5</td>
</tr>
</tbody>
</table>

Response

<string> = <STRING RESPONSE DATA>

Table 7.12.15.2-3 Response Format

<table>
<thead>
<tr>
<th>Form</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Form1</td>
<td>&quot;XXXXXXXX&quot;</td>
<td>0 to 9999999</td>
</tr>
<tr>
<td>Integer type</td>
<td>&quot;X.XXXExx&quot;</td>
<td>1.0000E07 to 9.9999E17</td>
</tr>
<tr>
<td>&quot;-----&quot;</td>
<td>When no data corresponds to the query</td>
<td></td>
</tr>
<tr>
<td>Form2</td>
<td>&quot;X.XXXE-XX&quot;</td>
<td>0.0001E–18 to 1.0000E00</td>
</tr>
<tr>
<td>Fraction type</td>
<td>&quot;-----&quot;</td>
<td>When no data corresponds to the query</td>
</tr>
<tr>
<td>Form3</td>
<td>&quot;XXXXXXXXXXX&quot;</td>
<td>For 0 to MAX (Hz)</td>
</tr>
<tr>
<td>Frequency type</td>
<td>&quot;-----&quot;</td>
<td>When no data corresponds to the query</td>
</tr>
<tr>
<td>Form4</td>
<td>&quot;X.XXX.XXX&quot;</td>
<td>For −4.000 to +4.000</td>
</tr>
<tr>
<td>Voltage type</td>
<td>&quot;-----&quot;</td>
<td>When no data corresponds to the query</td>
</tr>
<tr>
<td>Form5</td>
<td>&quot;XXXX,XXXX&quot;</td>
<td>For −1000 to +1000, −10000 to +10000</td>
</tr>
<tr>
<td>Phase type</td>
<td>&quot;-----&quot;</td>
<td>When no data corresponds to the query</td>
</tr>
</tbody>
</table>

Function

Queries the 3 Eye Parallel results for PAM BER measurement.

Example

To query the total error rate result for 3 Eye Parallel measurement:
> :CALCulate:DATA:PBParallel? "TOTal:ER"
< "1.2345E–03"

Compatibility

Incompatible with existing models.
### :SENSe:MEASure:PBParallel:DATA:THReshold <threshold>,<port>,<numeric>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;threshold&gt;</td>
<td>= &lt;CHARACTER PROGRAM DATA&gt;</td>
</tr>
<tr>
<td>TOP</td>
<td>Top level</td>
</tr>
<tr>
<td>MIDDle</td>
<td>Middle level</td>
</tr>
<tr>
<td>BOTTtom</td>
<td>Bottom level</td>
</tr>
<tr>
<td>&lt;port&gt;</td>
<td>= &lt;CHARACTER PROGRAM DATA&gt;</td>
</tr>
<tr>
<td>DATA</td>
<td>Selects Data.</td>
</tr>
<tr>
<td>XDATa</td>
<td>Selects XData.</td>
</tr>
<tr>
<td>&lt;numeric&gt;</td>
<td>= &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
</tr>
<tr>
<td>–3.500 to 3.300</td>
<td>–3.500 to 3.300 V/0.001 V Step</td>
</tr>
</tbody>
</table>

### Function
Sets input port and measurement threshold for Top, Middle, and Bottom levels respectively in the 3 Eye Parallel PAM BER measurement.

### Example
To set the measurement threshold to Data input, 0.100 V:

```plaintext
> :SENSe:MEASure:PBParallel:DATA:THReshold TOP,DATA,0.100
```

### Compatibility
Incompatible with existing models.

### :SENSe:MEASure:PBParallel:DATA:THReshold? <threshold>,<port>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;threshold&gt;</td>
<td>= &lt;CHARACTER PROGRAM DATA&gt;</td>
</tr>
<tr>
<td>TOP</td>
<td>Top level</td>
</tr>
<tr>
<td>MIDDle</td>
<td>Middle level</td>
</tr>
<tr>
<td>BOTTtom</td>
<td>Bottom level</td>
</tr>
<tr>
<td>&lt;port&gt;</td>
<td>= &lt;CHARACTER PROGRAM DATA&gt;</td>
</tr>
<tr>
<td>DATA</td>
<td>Selects Data.</td>
</tr>
<tr>
<td>XDATa</td>
<td>Selects XData.</td>
</tr>
<tr>
<td>&lt;numeric&gt;</td>
<td>= &lt;NR1 NUMERIC RESPONSE DATA&gt;</td>
</tr>
<tr>
<td>–3.500 to 3.300</td>
<td>–3.500 to 3.300 V/0.001 V Step</td>
</tr>
</tbody>
</table>

### Response

### Function
Queries input port and measurement threshold for Top, Middle, and Bottom levels respectively in the 3 Eye Parallel PAM BER measurement.

### Example
To query threshold of Top level, data input:

```plaintext
< 0.100
```

### Compatibility
Incompatible with existing models.
7.12 28G/32G bit/s ED Commands

:SENSe:MEASure:PBParallel:CLOCk:DELay <threshold>,<numeric>

Parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;threshold&gt;</td>
<td>= &lt;CHARACTER PROGRAM DATA&gt;</td>
</tr>
<tr>
<td>TOP</td>
<td>Top level</td>
</tr>
<tr>
<td>MIDDle</td>
<td>Middle level</td>
</tr>
<tr>
<td>BOTTom</td>
<td>Bottom level</td>
</tr>
<tr>
<td>&lt;numeric&gt;</td>
<td>= &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
</tr>
<tr>
<td>–1000 to 1000</td>
<td>–1000 to 1000 mUI/2 mUI Step</td>
</tr>
</tbody>
</table>

Function

Sets Delay for Top, Middle, and Bottom levels respectively for the 3 Eye Parallel PAM BER measurement.

Example

To set Delay of Top level to 300 mUI:

> :SENSe:MEASure:PBParallel:CLOCk:DELay TOP,300

Compatibility

Incompatible with existing models.


Parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;threshold&gt;</td>
<td>= &lt;CHARACTER PROGRAM DATA&gt;</td>
</tr>
<tr>
<td>TOP</td>
<td>Top level</td>
</tr>
<tr>
<td>MIDDle</td>
<td>Middle level</td>
</tr>
<tr>
<td>BOTTom</td>
<td>Bottom level</td>
</tr>
</tbody>
</table>

Response

<table>
<thead>
<tr>
<th>Numeric</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>–1000 to 1000</td>
<td>–1000 to 1000 mUI/2 mUI Step</td>
</tr>
</tbody>
</table>

Function

Queries Delay for Top, Middle, and Bottom levels respectively in the 3 Eye Parallel PAM BER measurement.

Example

To query the measurement threshold of Top level, data input:


< 300

Compatibility

Incompatible with existing models.

:SENSe:MEASure:PBParallel:TIME <type>

Parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;type&gt;</td>
<td>= &lt;CHARACTER PROGRAM DATA&gt;</td>
</tr>
<tr>
<td>DTIMe</td>
<td>Displays the current date and time.</td>
</tr>
<tr>
<td>STARt</td>
<td>Displays the measurement start time.</td>
</tr>
<tr>
<td>ELAPsed</td>
<td>Displays the elapsed time based on the measurement period.</td>
</tr>
</tbody>
</table>

Function

Selects the measurement time display for 3 Eye Parallel PAM BER measurement.

Example

To display elapsed measurement time:

> :SENSe:MEASure:PBParallel:TIME ELAPsed

Compatibility

Incompatible with existing models.
Chapter 7  SCPI Commands

:SENSe:MEASure:PBParallel:TIME?

Response  
<type> = <CHARACTER RESPONSE DATA>
DTIM, STAR, ELAP

Function  
Queries the measurement time display for 3 Eye Parallel PAM BER measurement.

Example  
> :SENSe:MEASure:PBParallel:TIME?
< ELAP

Compatibility  
Incompatible with existing models.

:SENSe:MEASure:PBParallel:TIME:DTIMe?

Response  
<year>,<month>,<day>,<hour>,<min>,<second>
<year> = <NR1 NUMERIC RESPONSE DATA>  
2000 to 2036  Year 2000 to 2036
<month> = <NR1 NUMERIC RESPONSE DATA>  
1 to 12  January to December
<day> = <NR1 NUMERIC RESPONSE DATA>  
1 to 31  1st to 31st
<hour> = <NR1 NUMERIC RESPONSE DATA>  
0 to 23  0 to 23 hours
<minute> = <NR1 NUMERIC RESPONSE DATA>  
0 to 59  0 to 59 minutes
<second> = <NR1 NUMERIC RESPONSE DATA>  
0 to 59  0 to 59 seconds

Function  
Queries the current time for 3 Eye Parallel PAM BER measurement.

Example  
> :SENSe:MEASure:PBParallel:TIME:DTIMe?
< 2016,12,31,23,59,59

Compatibility  
Incompatible with existing models.
:SENSe:MEASure:PBParallel:TIME:STARt?

Response

Response

(year>,<month>,<day>,<hour>,<min>,<second>

(year> = <NR1 NUMERIC RESPONSE DATA>
2000 to 2036 Year 2000 to 2036

(month> = <NR1 NUMERIC RESPONSE DATA>
1 to 12 January to December

(day> = <NR1 NUMERIC RESPONSE DATA>
1 to 31 1st to 31st

(hour> = <NR1 NUMERIC RESPONSE DATA>
0 to 23 0 to 23 hours

(minute> = <NR1 NUMERIC RESPONSE DATA>
0 to 59 0 to 59 minutes

(second> = <NR1 NUMERIC RESPONSE DATA>
0 to 59 0 to 59 seconds

Function Queries the start time for 3 Eye Parallel PAM BER measurement.

Example

> :SENSe:MEASure:PBParallel:TIME:STARt?
< 2016,12,31,23,59,59

Compatibility Incompatible with existing models.

:SENSe:MEASure:PBParallel:TIME:ELAPsed?

Response

Response

(hour>,<min>,<second>

(hour> = <NR1 NUMERIC RESPONSE DATA>
0 to 23 0 to 23 hours

(minute> = <NR1 NUMERIC RESPONSE DATA>
0 to 59 0 to 59 minutes

(second> = <NR1 NUMERIC RESPONSE DATA>
0 to 59 0 to 59 seconds

Function Queries the elapsed time for 3 Eye Parallel PAM BER measurement.

Example

> :SENSe:MEASure:PBParallel:TIME:ELAPsed?
< 23,59,59

Compatibility Incompatible with existing models.
7.12.15.3 File Menu Setting Commands

Table 7.12.15.3-1  File Menu Setting Commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
</table>

:SYSTem:MMEMory:PBSerial:RECall <file_name>

Parameter

- `<file_name> = <STRING PROGRAM DATA>`
- `"<drv>\[<dir>]\<file>"`
- `<drv> = C, D, E, F`
- `<dir> = <dir1>\<dir2>\...(Omitted when root directory)`
- `<file> = File name`

Function

Opens the measurement result data for 3 Eye Serial PAM BER measurement.

Example

> :SYSTem:MMEMory:PBSerial:RECall "C:\Test\example"

Compatibility

Incompatible with existing models.

:SYSTem:MMEMory:PBSerial:STORe <file_name>,<file_type>

Parameter

- `<file_name> = <STRING PROGRAM DATA>`
- `"<drv>\[<dir>]\<file>"`
- `<drv> = C, D, E, F`
- `<dir> = <dir1>\<dir2>\...(Omitted when root directory)`
- `<file> = File name`
- `<file_type> = <CHARACTER PROGRAM DATA>`
  - BIN  Binary file
  - CSV  CSV file
  - TXT  Text file

Function

Saves the measurement result data for 3 Eye Serial PAM BER measurement.

**Note:**

The settings will not be read from the saved file if the file name is changed.

Example

To save the 3 Eye Serial PAM BER measurement result data, specifying the save destination, file name and file format:

> :SYSTem:MMEMory:PBSerial:STORe "C:\Test\example",TXT

Compatibility

Incompatible with existing models.
7.12 28G/32G bit/s ED Commands

:SYSTem:MEMory:PBParallel:RECall <file_name>

Parameter

- `<file_name>` = `<STRING PROGRAM DATA>`
- `"<drv>\\[<dir>]\\<file>"`
- `<drv>` = C, D, E, F
- `<dir>` = `<dir1>\\<dir2>\\...(Omitted when root directory)`
- `<file>` = File name

Function
Opens the measurement result data for 3 Eye Parallel PAM BER measurement.

Example

```
> :SYSTem:MEMory:PBParallel:RECall "C:\Test\example"
```

Compatibility
Incompatible with existing models.

:SYSTem:MEMory:PBParallel:STORe <file_name>,<file_type>

Parameter

- `<file_name>` = `<STRING PROGRAM DATA>`
- `"<drv>\\[<dir>]\\<file>"`
- `<drv>` = C, D, E, F
- `<dir>` = `<dir1>\\<dir2>\\...(Omitted when root directory)`
- `<file>` = File name
- `<file_type>` = `<CHARACTER PROGRAM DATA>`
  - BIN: Binary file
  - CSV: CSV file
  - TXT: Text file

Function
Saves the measurement result data for 3 Eye Parallel PAM BER measurement.

**Note:**
The settings will not be read from the saved file if the file name is changed.

Example

To save the 3 Eye Parallel PAM BER measurement result data, specifying the save destination, file name and file format:

```
> :SYSTem:MEMory:PBParallel:STORe "C:\Test\example",TXT
```

Compatibility
Incompatible with existing models.
7.12.16 Eye Contour measurement

This section describes Eye Contour measurement commands. The commands on each tab are described in separate sections below.

For details of commands specific to Eye Contour measurement, refer to 7.12.16.1 “Eye Contour Measurement Commands”.

Commands specific to PAM4 Eye Contour measurement are described in 7.12.16.2 “PAM4 Eye Contour Measurement Commands”.

Commands shared with Eye Diagram measurement are listed in 7.12.16.3 “Compatible Commands with Eye Diagram Measurement” and described in detail in 7.12.10 “Eye Diagram Measurement”.

7.12.16.1 Eye Contour Measurement Commands

![Eye Contour Measurement Screen]

Figure 7.12.16.1-1 Eye Contour Measurement Screen
Table 7.12.16.1-1  Eye Contour Tab Setting Commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>:DISPLAY:RESULT:EDDiagram:TABLESelect?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:MEASure:EDDiagram:DIAGram:ASEarch?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:MEASure:EDDiagram:DIAGram:TYPE?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:MEASure:EDDiagram:DIAGram:LMRatio?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:DISPLAY:RESULT:EDDiagram:HETHreshold?</td>
</tr>
</tbody>
</table>

Figure 7.12.16.1-2  Eye Contour Mask Judge Screen
**Chapter 7  SCPI Commands**

### :DISPlay:RESult:EDiagram:TABSelect <tab>

**Parameter**
- `<tab>` = `<CHARACTER PROGRAM DATA>`
  - CONTour: Contour tab
  - MASKedit: Mask Edit tab

**Function**
Selects the Eye Contour measurement tab display.

**Example**
To display the Eye Contour measurement tab:

```
> :DISPlay:RESult:EDiagram:TABSelect CONTour
```

**Compatibility**
Some parameters are compatible with Eye Diagram measurement (32G).

### :DISPlay:RESult:EDiagram:TABSelect?

**Response**
- `<tab>` = `<CHARACTER RESPONSE DATA>`
  - CONT, MASK

**Function**
Queries the Eye Contour measurement tab display.

**Example**
```
> :DISPlay:RESult:EDiagram:TABSelect?
< CONT
```

**Compatibility**
Some parameters are compatible with Eye Diagram measurement (32G).

### :SENSe:MEASure:EDiagram:DIAGram:ASEarch <setting> [,<mode>]

**Parameter**
- `<setting>` = `<BOOLEAN PROGRAM DATA>`
  - ON or 1: Auto Search ON
  - OFF or 0: Auto Search OFF
- `<mode>` = `<DECIMAL NUMERIC PROGRAM DATA>`
  - 1: Coarse
  - 3: PAM4 Coarse
  - 4: PAM4 Fine

**Note:**
- `<mode>` can be omitted. Coarse is specified when omitted.

**Function**
Selects the Auto Search mode when starting Eye Contour measurement.

**Example**
To execute Auto Search Coarse when starting Eye Contour measurement:
```
> :SENSe:MEASure:EDiagram:DIAGram:ASEarch ON,1
```

**Compatibility**
Some parameters are compatible with Eye Diagram measurement (32G).
:SENSe:MEASure:EDIagram:DIAGram:ASEarch?

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;setting&gt; = &lt;DECIMAL NUMERIC RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0   OFF</td>
</tr>
<tr>
<td></td>
<td>1   ON (Coarse)</td>
</tr>
<tr>
<td></td>
<td>3   ON (PAM Coarse)</td>
</tr>
<tr>
<td></td>
<td>4   ON (PAM4 Fine)</td>
</tr>
</tbody>
</table>

Function: Queries the Auto Search type to be executed when starting Eye Contour measurement.

Example:
> :SENSe:MEASure:EDIagram:DIAGram:ASEarch?
< 1

Compatibility: Some parameters are compatible with Eye Diagram measurement (32G).

:SENSe:MEASure:EDIagram:DIAGram:TYPE <type>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;type&gt; = &lt;CHARACTER PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>POINt4  4 Point</td>
</tr>
<tr>
<td></td>
<td>POINt8  8 Point</td>
</tr>
<tr>
<td></td>
<td>POINt16 16 Point</td>
</tr>
</tbody>
</table>

Function: Sets the number of measurement points for Eye Contour measurement.

Example:
To change the number of measurement points to 8 for Eye Contour measurement:
> :SENSe:MEASure:EDIagram:DIAGram:TYPE POINt8

Compatibility: Some parameters are compatible with Eye Diagram measurement (32G).

:SENSe:MEASure:EDIagram:DIAGram:LMRatio <range>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;range&gt; = &lt;CHARACTER PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>E_7 to E_12</td>
</tr>
<tr>
<td></td>
<td>E–7 to E–12</td>
</tr>
</tbody>
</table>

Function: Sets the measurement error threshold lower limit for Eye Contour measurement.

Example:
To set the measurement error threshold lower limit to E–12:
> :SENSe:MEASure:EDIagram:DIAGram:LMRatio E_12

Compatibility: Incompatible with existing models.
### Chapter 7  SCPI Commands

#### :SENSe:MEASure:EDIagram:DIAGram:LMRatio?

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;range&gt; = &lt;CHARACTER RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>E_7 to E_12</td>
<td>E–7 to E–12</td>
</tr>
</tbody>
</table>

**Function**
Queries the measurement error threshold lower limit for Eye Contour measurement.

**Example**
```
> :SENSe:MEASure:EDIagram:DIAGram:LMRatio?
< E_12
```

**Compatibility**
Incompatible with existing models.

#### :DISPlay:RESult:EDIagram:HETHreshold <range>,<numeric1>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;range&gt; = &lt;CHARACTER PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>E_6 to E_20  E–6 to E–20</td>
</tr>
<tr>
<td></td>
<td>&lt;numeric1&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
</tr>
<tr>
<td></td>
<td>1.0 to 9.9  Mantissa</td>
</tr>
</tbody>
</table>

**Function**
Sets the error ratio displayed for Eye Contour.

**Example**
```
To display the Eye Contour for 5.5E–10:
> :DISPlay:RESult:EDIagram:HETHreshold E_10,5.5
```

**Compatibility**
Incompatible with existing models.

#### :DISPlay:RESult:EDIagram:HETHreshold?

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;range&gt; = &lt;CHARACTER RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>E_6 to E_20</td>
<td>E–6 to E–20</td>
</tr>
<tr>
<td>&lt;numeric1&gt;</td>
<td>&lt;DECIMAL NUMERIC RESPONSE DATA&gt;</td>
</tr>
<tr>
<td>1.0 to 9.9</td>
<td>Mantissa</td>
</tr>
</tbody>
</table>

**Function**
Queries the setting of error ratio displayed for Eye Contour.

**Example**
```
> :DISPlay:RESult:EDIagram:HETHreshold?
< E_10,5.5
```

**Compatibility**
Incompatible with existing models.
:CALCulate:DATA:DIAGram? <slot>,<interface>,<string>[,<unit>]

Parameter

- `<slot>` = <CHARACTER PROGRAM DATA>
  1 to 6   Slots No. 1 to 6
- `<interface>` = <DECIMAL NUMERIC PROGRAM DATA>
  1 to 4   Data 1 to 4
- `<string>` = <STRING PROGRAM DATA>
  1 to 16  Measurement point

Specify the measurement point and the error rate of the Eye contour.

Examples of `<string>` when specifying the measurement point 1 are shown in the table below.

<table>
<thead>
<tr>
<th>Error rate</th>
<th><code>&lt;string&gt;</code></th>
</tr>
</thead>
<tbody>
<tr>
<td>1E–6</td>
<td>&quot;MAP:E_6:POINt1&quot;</td>
</tr>
<tr>
<td>1E–7</td>
<td>&quot;MAP:E_7:POINt1&quot;</td>
</tr>
<tr>
<td>:</td>
<td>:</td>
</tr>
<tr>
<td>1E–19</td>
<td>&quot;MAP:E_19:POINt1&quot;</td>
</tr>
<tr>
<td>1E–20</td>
<td>&quot;MAP:E_20:POINt1&quot;</td>
</tr>
<tr>
<td>Highlight Error Threshold</td>
<td>&quot;MAP:HET:POINt1&quot;</td>
</tr>
</tbody>
</table>

- `<unit>` = <DECIMAL NUMERIC PROGRAM DATA>
  1 to 4   Mainframe No. 1 to 4

When using two or more MT1810A units in serial connection, specify the mainframe number.

Can be omitted. Mainframe No. 1 is specified when omitted.

Response

- `<result>` = <STRING RESPONSE DATA>
  Measured data

For the format of `<result>`, refer to Table 7.12.16.1-3.

<table>
<thead>
<tr>
<th>Form</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Form1</td>
<td>&quot;XXXXX,YYYYY&quot;</td>
<td>XXXXX: Phase (mUI)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>XXXXX: Phase (ps) (Displays in the selected unit.)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>YYYYY: Threshold (mV)</td>
</tr>
<tr>
<td>&quot;-----,-----&quot;</td>
<td>When no data corresponds to the query</td>
<td></td>
</tr>
</tbody>
</table>

Function

Queries the Eye Contour measurement result of a specific slot and point.

Example

To query the Eye Contour measurement result at slot No. 1, Data 2, Highlight Error Threshold, point 1:

```
>:CALCulate:DATA:DIAGram? SLOT1,2,"MAP:HET:POINt1"
" 100,1000"
```

Compatibility

Some parameters are compatible with Eye Diagram measurement (32G).
Chapter 7  SCPI Commands

:DISPlay:RESult:EDIagram:DIAGram:TABSelect <tab>

Parameter  
<tab> = <CHARACTER PROGRAM DATA>
CONDition    Condition tab
MJUDge       Mask Judge tab
RESult       Result tab

Function    Switches tab display on the Diagram tab.
Example     To display the Result tab on the Diagram tab:
> :DISPlay:RESult:EDIagram:DIAGram:TABSelect RESult

Compatibility  Some parameters of this command are not the same as those for the Eye Diagram measurement (32G).

:DISPlay:RESult:EDIagram:DIAGram:TABSelect?

Response  <tab> = <CHARACTER RESPONSE DATA>
COND, MJUD, RES

Function    Queries tab display on the Diagram tab.
Example     > :DISPlay:RESult:EDIagram:TABSelect?
< COND

Compatibility  Some parameters of this command are not the same as those for the Eye Diagram measurement (32G).

:CALCulate:DATA:EDIagram:DETail:PNUMber?

Response  <numeric> = <NR1 NUMERIC RESPONSE DATA>
0 to 32    Result point number

Function    Queries result point number of Mask Judge.
Example     > :CALCulate:DATA:EDIagram:DETail:PNUMber?
< 6

Compatibility  Some parameters of the Eye Contour measurement and the Eye Diagram measurement of the MU181040A/B are interchangeable.
::CALCulate::DATA::EDiagram::DETail::POINt? <string>

Parameter

<string> = <STRING PROGRAM DATA>

<Result>
Measurement data

Refer to Table 7.12.16.1-5 for the details of <result>.

Response

<string>=<STRING RESPONSE DATA>

Table 7.12.16.1-4  Response Format

<table>
<thead>
<tr>
<th>Form</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Form1</td>
<td>&quot;XXXXX,YYYYY,ZZ&quot;</td>
<td>XXXXX: Phase (mUI) XXXXX: Phase (ps)</td>
</tr>
<tr>
<td></td>
<td>&quot;-----,--.---,--&quot;</td>
<td>When no data corresponds to the query</td>
</tr>
</tbody>
</table>

Note:
Displays in the specified unit.
YY.YYY: Threshold (mV)
ZZ: OK, NG

Function

Queries the measurement results of the Eye Contour Mask Judge measurement.

Table 7.12.16.1-5  Eye Contour Mask Judge Measurement Item <result>

<table>
<thead>
<tr>
<th>Items</th>
<th>&lt;result&gt;</th>
<th>Response Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>Judgment type</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Point1</td>
<td>&quot;JDG:POINt1&quot;</td>
<td>Form1</td>
</tr>
<tr>
<td>Point2</td>
<td>&quot;JDG:POINt2&quot;</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Point31</td>
<td>&quot;JDG:POINt31&quot;</td>
<td></td>
</tr>
<tr>
<td>Point32</td>
<td>&quot;JDG:POINt32&quot;</td>
<td></td>
</tr>
</tbody>
</table>

Example

To query the measurement result of Point1 of the Eye Contour Mask Judge:

> :CALCulate::DATA::EDiagram::DETail::POINt? "JDG:POINt1"
< "0, 4000,OK"

Compatibility

Parameters of this command are the same as those for the existing MU181040A/B command for the Eye Diagram measurement.
7.12.16.2 PAM4 Eye Contour Measurement Commands

Figure 7.12.16.2-1  PAM4 Eye Contour Measurement Screen

Figure 7.12.16.2-2  PAM4 Eye Contour Mask Judge Screen
### Table 7.12.16.2-1  PAM4 Eye Contour Tab Setting Commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Input Signal</td>
<td>:SENSe:MEASure:EDIagram:DIAGram:ISIGnal</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:MEASure:EDIagram:DIAGram:ISIGnal?</td>
</tr>
<tr>
<td>2</td>
<td>Selection of slot(s) and channel(s) to be measured</td>
<td>:SENSe:MEASure:EDIagram:DIAGram:PAM:SLOT</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:MEASure:EDIagram:DIAGram:PAM:SLOT?</td>
</tr>
<tr>
<td>3</td>
<td>Selection of threshold(s) to be measured (Upper/Middle/Lower)</td>
<td>:SENSe:MEASure:EDIagram:DIAGram:PAM:PAMSet</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:MEASure:EDIagram:DIAGram:PAM:PAMSet?</td>
</tr>
<tr>
<td>4</td>
<td>Threshold</td>
<td>:SENSe:MEASure:EDIagram:DIAGram:PAM:THReshold</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:MEASure:EDIagram:DIAGram:PAM:THReshold?</td>
</tr>
<tr>
<td>5</td>
<td>Phase</td>
<td>:SENSe:MEASure:EDIagram:DIAGram:PAM:PHASe</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:MEASure:EDIagram:DIAGram:PAM:PHASe?</td>
</tr>
<tr>
<td>6</td>
<td>Selection of pattern</td>
<td>:SENSe:MEASure:EDIagram:DIAGram:PAM(TYPE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:MEASure:EDIagram:DIAGram:PAM:TYPE</td>
</tr>
<tr>
<td>7</td>
<td>Selection of mask(s)</td>
<td>:DISPlay:RESult:EDIagram:TEMPlate:PAM:SELect</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:DISPlay:RESult:EDIagram:TEMPlate:PAM:SELect?</td>
</tr>
<tr>
<td>8</td>
<td>Mask Adjust</td>
<td>:DISPlay:RESult:EDIagram:PAM:ADJust</td>
</tr>
<tr>
<td>9</td>
<td>Selection of threshold(s) to be displayed on PAM4 measurement result graph</td>
<td>:DISPlay:RESult:EDIagram:PAM:THReshold</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:DISPlay:RESult:EDIagram:PAM:THReshold?</td>
</tr>
<tr>
<td>10</td>
<td>Query for measurement results</td>
<td>:CALCulate:DATA:DIAGram:PAM4?</td>
</tr>
<tr>
<td>11</td>
<td>Query Mask point number</td>
<td>:CALCulate:DATA:DIagram:DETail:PNUMber:PAM4?</td>
</tr>
<tr>
<td>12</td>
<td>Query Mask Judge results</td>
<td>:CALCulate:DATA:DIagram:DETail:POINt:PAM4?</td>
</tr>
<tr>
<td>13</td>
<td>Query for Mask Test result</td>
<td>:CALCulate:DATA:DIagram:MASK:PAM4?</td>
</tr>
</tbody>
</table>
Chapter 7  SCPI Commands

:SENSe:MEASure:EDIagram:DIAGram:ISIGnal <mode>

Parameter  
<mode> = <CHARACTER PROGRAM DATA>
 NRZ  NRZ signal
 PAM4  PAM4 signal

Function  
Selects the input signal for Eye Contour measurement.

Example  
To select the PAM4 signal:
> :SENSe:MEASure:EDIagram:DIAGram:ISIGnal PAM4

Compatibility  
Incompatible with existing models.

:SENSe:MEASure:EDIagram:DIAGram:ISIGnal?

Response  
<mode> = <CHARACTER RESPONSE DATA>
 NRZ, PAM4

Function  
Queries the input signal for Eye Contour measurement.

Example  
> :SENSe:MEASure:EDIagram:DIAGram:ISIGnal?
< PAM4

Compatibility  
Incompatible with existing models.

:SENSe:MEASure:EDIagram:DIAGram:PAM:SLOT <slot>,<interface>[,<unit>]

Parameter  
<slot> = <CHARACTER PROGRAM DATA>
 SLOT1 to SLOT6  Slot No.1 to 6
 1 to 6 when using the MP1800A, 1 to 4 when using the MT1810A
 <interface> = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 4  Data1 to 4
 [<unit>] = <DECIMAL NUMERIC PROGRAM DATA>
 1 to 4  Mainframe No.1 to 4

When using two or more MT1810A units in serial connection, specify the mainframe number.
Can be omitted. Mainframe No. 1 is specified when omitted.

Function  
Sets the measurement slot and Data IF at the PAM4 Eye Contour measurement.

Example  
To set “slot 4” and “Data1” to ED used at the Eye Contour measurement:
> :SENSe:MEASure:EDIagram:DIAGram:PAM:SLOT SLOT4,1

Compatibility  
Incompatible with existing models.

Parameter

- [<unit>] = <DECIMAL NUMERIC PROGRAM DATA>
- 1 to 4 Mainframe No.1 to 4
  When using two or more MT1810A units in serial connection, specify the mainframe number.
  Can be omitted. Mainframe No. 1 is specified when omitted.

Response

- <slot> = <CHARACTER RESPONSE DATA>
- SLOT1 to SLOT6 Slots No. 1 to 6
- 1 to 6 when using the MP1800A, 1 to 4 when using the MT1810A
- <interface> = <NR1 NUMERIC RESPONSE DATA>  
- 1 to 4 Data1 to 4

Function

Queries the measurement slot and Data IF at the PAM4 Eye Contour measurement.

Example

To query the ED slot and Data IF used at the PAM4 Eye Contour measurement:

> :SENSe:MEASure:EDiagram:DIAGram:PAM:SLOT?
< SLOT4,1

Compatibility

Incompatible with existing models.

:SENSe:MEASure:EDiagram:DIAGram:PAM:PAMSet <thre>,<boolean>

Parameter

- <thre> = <CHARACTER PROGRAM DATA>
  - UPPER Upper threshold
  - MIDDle Middle threshold
  - LOWer Lower threshold
- <boolean> = <BOOLEAN PROGRAM DATA>
  - ON or 1 Selects as a measuring object.
  - OFF or 0 Does not select as a measuring object.

Function

Selects the threshold(s) to be measured in PAM4 Eye Contour measurement.

Example

To select the Middle thresholds as measuring objects:

> :SENSe:MEASure:EDiagram:DIAGram:PAM:PAMSet MIDDle,1

Compatibility

Incompatible with existing models.
### :SENSe:MEASure:EDiagram:DIAGram:PAM:PAMSet? <thre>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;thre&gt; = &lt;CHARACTER PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Upper</td>
<td>Upper threshold</td>
</tr>
<tr>
<td>Middle</td>
<td>Middle threshold</td>
</tr>
<tr>
<td>Lower</td>
<td>Lower threshold</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;boolean&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>OFF</td>
</tr>
<tr>
<td>1</td>
<td>ON</td>
</tr>
</tbody>
</table>

**Function**
Queries the threshold(s) to be measured in PAM4 Eye Contour measurement.

**Example**
> :SENSe:MEASure:EDiagram:DIAGram:PAM:PAMSet? MIDDle < 1

**Compatibility**
Incompatible with existing models.

### :SENSe:MEASure:EDiagram:DIAGram:PAM:THReshold <thre>,<port>,<numeric>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;thre&gt; = &lt;CHARACTER PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Upper</td>
<td>Upper threshold</td>
</tr>
<tr>
<td>Middle</td>
<td>Middle threshold</td>
</tr>
<tr>
<td>Lower</td>
<td>Lower threshold</td>
</tr>
<tr>
<td>&lt;port&gt;</td>
<td>&lt;CHARACTER PROGRAM DATA&gt;</td>
</tr>
<tr>
<td>Data</td>
<td>Data</td>
</tr>
<tr>
<td>XDATa</td>
<td>XData</td>
</tr>
<tr>
<td>&lt;numeric&gt;</td>
<td>&lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
</tr>
<tr>
<td>–3.500 to 3.300</td>
<td>–3.500 to 3.300 V/0.001 V Step</td>
</tr>
</tbody>
</table>

**Function**
Sets the threshold at which PAM4 Eye Contour measurement starts.

**Example**
To set Data Input of 0.100 V for the Upper measurement threshold at which measurement starts:
> :SENSe:MEASure:EDiagram:DIAGram:PAM:THReshold UPPer,Data,0.100

**Compatibility**
Incompatible with existing models.

**Parameter**
- `<thre>` = `<CHARACTER PROGRAM DATA>`
  - UPPer: Upper threshold
  - MIDDle: Middle threshold
  - LOWer: Lower threshold
- `<port>` = `<CHARACTER PROGRAM DATA>`
  - DATA: Data
  - XDATa: XData

**Response**
- `<numeric>` = `<NR1 NUMERIC RESPONSE DATA>`
  - –3.500 to 3.300 V/0.001 V Step

**Function**
Queries the threshold at which PAM4 Eye Contour measurement starts.

**Example**
To query the Upper threshold at which PAM4 Eye Contour measurement starts:

```
> :SENSe:MEASure:EDiagram:DIAGram:PAM:THReshold?
UPPer,Data
< 0.100
```

**Compatibility**
Incompatible with existing models.

### :SENSe:MEASure:EDiagram:DIAGram:PAM:PHASe <thre>,<numeric>

**Parameter**
- `<thre>` = `<CHARACTER PROGRAM DATA>`
  - UPPer: Upper threshold
  - MIDDle: Middle threshold
  - LOWer: Lower threshold
- `<numeric>` = `<DECIMAL NUMERIC PROGRAM DATA>`
  - –1000 to 1000 mUI/2 mUI Step

**Function**
Sets the phase at which PAM4 Eye Contour measurement starts.

**Example**
To set 300 mUI as the phase of the Upper threshold at which measurement starts:

```
> :SENSe:MEASure:EDiagram:DIAGram:PAM:PHASe UPPer,300
```

**Compatibility**
Incompatible with existing models.
**Chapter 7  SCPI Commands**

### :SENSe:MEASure:EDIagram:DIAGram:PAM:PHASe? <thre>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;thre&gt; = &lt;CHARACTER PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>UPPer</td>
</tr>
<tr>
<td></td>
<td>MIDDle</td>
</tr>
<tr>
<td></td>
<td>LOWer</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;numeric&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>–1000 to 1000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Function</th>
<th>Queries the phase at which PAM4 Eye Contour measurement starts.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example</td>
<td>To query the phase set for the Upper threshold at which PAM4 Eye Contour measurement starts.</td>
</tr>
<tr>
<td></td>
<td>&gt; :SENSe:MEASure:EDIagram:DIAGram:PAM:PHASe? UPPer &lt; 300</td>
</tr>
</tbody>
</table>

| Compatibility | Incompatible with existing models. |

### :SENSe:MEASure:EDIagram:DIAGram:PAM:TYPE <type>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;type&gt; = &lt;STRING PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>&quot;file name&quot;</td>
</tr>
</tbody>
</table>

Specify the file name according to “Appendix F” in the *MU183040B Operation Manual*. The following are example settings:

- "PRBS7" PRBS7
- "PRBS9" PRBS9
- "PRBS10" PRBS10
- "PRBS11" PRBS11
- "PRBS15" PRBS15
- "PRBS20" PRBS20
- "PRQS10" PRQS10
- "SSPR" SSPR
- "QPRBS13-CEI" QPRBS13-CEI

<table>
<thead>
<tr>
<th>Function</th>
<th>Sets the test pattern for PAM4 Eye Contour measurement.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example</td>
<td>To set &quot;QPRBS13-CEI&quot; for the test pattern:</td>
</tr>
<tr>
<td></td>
<td>&gt; :SENSe:MEASure:EDIagram:DIAGram:PAM:TYPE &quot;QPRBS13-CEI&quot;</td>
</tr>
</tbody>
</table>

| Compatibility | Incompatible with existing models. |
### :SENSe:MEASure:EDIagram:DIAGram:PAM:TYPE?

**Response**

```
$type = <STRING RESPONSE DATA>
"file name"
```
For the file names, refer to “Appendix F” in the MU183040B Operation Manual.

**Function**
Queries the test pattern for PAM4 Eye Contour measurement.

**Example**
```
> :SENSe:MEASure:EDIagram:DIAGram:PAM:TYPE?
< "QPRBS13-CEI"
```

**Compatibility**
Incompatible with existing models.

### :DISPlay:RESult:EDIagram:TEMPlate:PAM:SELect <thre>,<type>

**Parameter**

```
$thre = <CHARACTER PROGRAM DATA>
UPPer Upper Eye
MIDDle Middle Eye
LOWer Lower Eye
$type = <CHARACTER PROGRAM DATA>
OFF Mask OFF
E1 Uses Mask Edit1.
E2 Uses Mask Edit2.
E3 Uses Mask Edit3.
E4 Uses Mask Edit4.
```

**Function**
Selects the mask to use for the specified eye (Upper/Middle/Lower).

**Example**
To use Mask Edit3 for Lower Eye:
```
> :DISPlay:RESult:EDIagram:TEMPlate:PAM:SELect LOWer,E3
```

**Compatibility**
Incompatible with existing models.


**Parameter**

```
$thre = <CHARACTER PROGRAM DATA>
UPPer Upper Eye
MIDDle Middle Eye
LOWer Lower Eye
```

**Response**

```
$type = <CHARACTER RESPONSE DATA>
OFF, E1, E2, E3, E4
```

**Function**
Queries the mask to use for the specified eye (Upper/Middle/Lower).

**Example**
To query the mask used for Lower Eye:
```
< E3
```

**Compatibility**
Incompatible with existing models.
Chapter 7  SCPI Commands

:DISPlay:RESult:EDiagram:PAM:ADJust <thre>

Parameter          <thre> = <CHARACTER PROGRAM DATA>
                    UPPer        Upper Eye
                    MIDDle      Middle Eye
                    LOWer       Lower Eye
Function            Adjusts the mask for the specified eye (Upper/Middle/Lower).
Example             To adjust the mask for the Upper Eye:
                    > :DISPlay:RESult:EDiagram:PAM:ADJust UPPer
Compatibility       Incompatible with existing models.

:DISPlay:RESult:EDiagram:PAM:THReshold <thre>

Parameter          <thre> = <CHARACTER PROGRAM DATA>
                    UPPer        Upper Eye
                    MIDDle      Middle Eye
                    LOWer       Lower Eye
                    ALL         All Eyes
Function            Selects the eye (Upper/Middle/Lower) for displaying a result graph of
                    PAM4 Eye Contour measurement.
Example             To display the result for Lower Eye:
                    > :DISPlay:RESult:EDiagram:PAM:THReshold LOWer
Compatibility       Incompatible with existing models.

:DISPlay:RESult:EDiagram:PAM:THReshold?

Response           <type> = <CHARACTER RESPONSE DATA>
                    UPP, MIDD, LOW
Function            Queries the eye (Upper/Middle/Lower) that displays the result graph of
                    PAM4 Eye Contour measurement.
Example             > :DISPlay:RESult:EDiagram:PAM:THReshold?
                    < LOW
Compatibility       Incompatible with existing models.
:CALCulate:DATA:DIAGram:PAM4? <threshold>,<string>[,<unit>]

Parameter

- <threshold> = <CHARACTER PROGRAM DATA>
  - UPPer Queries the result of Upper Eye.
  - MIDDle Queries the result of Middle Eye.
  - LOWer Queries the result of Lower Eye.
- <string> = <STRING PROGRAM DATA>
  - 1 to 16 Measurement point
  - Specify the measurement point and the error rate of the Eye contour.
  - Examples of <string> when specifying the measurement point 1 are shown in the table below.

<table>
<thead>
<tr>
<th>Error rate</th>
<th>&lt;string&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>1E–6</td>
<td>&quot;MAP:E_6:POINt1&quot;</td>
</tr>
<tr>
<td>1E–7</td>
<td>&quot;MAP:E_7:POINt1&quot;</td>
</tr>
<tr>
<td></td>
<td>&quot;MAP:E_19:POINt1&quot;</td>
</tr>
<tr>
<td>1E–20</td>
<td>&quot;MAP:HET:POINt1&quot;</td>
</tr>
</tbody>
</table>

- [<unit>] = <DECIMAL NUMERIC PROGRAM DATA>
  - 1 to 4 Mainframe No. 1 to 4
  - When using two or more MT1810A units in serial connection, specify the mainframe number.
  - Can be omitted. Mainframe No. 1 is specified when omitted.

Response

- <result> = <STRING RESPONSE DATA>
  - <result> Measured data
  - For the format of <result>, refer to Table 7.12.16.2-3.

<table>
<thead>
<tr>
<th>Form</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Form1</td>
<td>&quot;XXXXX,YYYYY&quot;</td>
<td>XXXXX: Phase (mUI)</td>
</tr>
<tr>
<td></td>
<td>XXXXX: Phase (ps) Displays in the selected unit.)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>YYYYY: Threshold (mV)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&quot;-----,-----&quot;</td>
<td>When no data corresponds to the query</td>
</tr>
</tbody>
</table>

Function

Queries the Eye Contour measurement result of a specific slot and point.

Example

- To query the Eye Contour measurement result of the Highlight Error Threshold Point 1 for Upper Eye:
  \[ :\text{CALCulate:DATA:DIAGram? UPPer,"MAP:HET:POINt1"} \]
  \<" 100,1000" \]

Compatibility

Incompatible with existing models.
**Chapter 7  SCPI Commands**


<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;threshold&gt; = &lt;CHARACTER PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>UPPer</td>
<td>Queries the result of Upper Eye.</td>
</tr>
<tr>
<td>MIDDle</td>
<td>Queries the result of Middle Eye.</td>
</tr>
<tr>
<td>LOWer</td>
<td>Queries the result of Lower Eye.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;numeric&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 to 32</td>
<td>Result point number</td>
</tr>
</tbody>
</table>

| Function     | Queries the result point number of Mask Judge for Middle Eye. |


| Compatibility| Incompatible with existing models. |
:CALCulate:DATA:EDiagram:DETail:POINt:PAM4? <threshold>,<string>

**Parameter**

- **<threshold>** = <CHARACTER PROGRAM DATA>
- **UPPer** Queries the result of Upper Eye.
- **MIDDle** Queries the result of Middle Eye.
- **LOWer** Queries the result of Lower Eye.
- **<string>** = <STRING PROGRAM DATA>

**Response**

- **<string>** = <STRING RESPONSE DATA>

For the format of **<result>**, refer to Table 7.12.16.2-5.

<table>
<thead>
<tr>
<th>Form</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Form1</td>
<td>&quot;XXXXX,YYYYY,ZZ&quot;</td>
<td>XXXXX: Phase (mUI)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>XXXXX: Phase (ps)</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Note:</strong> Displays in the specified unit.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>YY.YYY: Threshold (mV)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ZZ: Pass, Failure</td>
</tr>
<tr>
<td></td>
<td>&quot;-----,--.---,--&quot;</td>
<td>When no data corresponds to the query</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Queries the measurement results of the Eye Contour Mask Judge measurement.</td>
</tr>
</tbody>
</table>

**Example**

To query the Point 1 measurement result of Mask Judge for Middle Eye Contour:

```plaintext
"    0, 4000,Pass"
```

**Compatibility**

Incompatible with existing models.
Chapter 7  SCPI Commands


Parameter

<threshold> = <CHARACTER PROGRAM DATA>
UPPer  Queries the result of Upper Eye.
MIDDle  Queries the result of Middle Eye.
LOWer  Queries the result of Lower Eye.

Response

<string> = <STRING RESPONSE DATA>

### Table 7.12.2-6  Response Format

<table>
<thead>
<tr>
<th>Form</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
</table>
| Form1 | "XX,Y,Y,Y ... Y" | XX: Pass,Failure  
Y: 1 to 32 (Failure mask point)  
**Note:**  
Multiple NGs are delimited with a comma (,) and returned for the corresponding number of NGs.  
"--,--" | When no data corresponds to the query. |

### Function

Queries the measurement result of the Eye Diagram Mask Test.

### Example

To query the Mask Test result of Upper Eye:

```
"Failure,1,2,3,4,32"
"Pass"
```

### Compatibility

Incompatible with existing models.
7.12.16.3 Compatible Commands with Eye Diagram Measurement

For commands description, refer to 7.12.10.1 “Condition tab”. The following command listed in Table 7.12.10.1-1 cannot be used in the software version 8.04.00 or later.

:SENSe:MEASure:EDIagram:DIAGram:ASEarch:PAMSet

Use the following command listed in Table 7.12.16.2-1 instead of the above command.

:SENSe:MEASure:EDIagram:DIAGram:PAM:PAMSet

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>:SENSe:MEASure:EDIagram:DIAGram:SLOT?</td>
</tr>
</tbody>
</table>
### Table 7.12.16.3-2  Graph Display Setting Commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1]</td>
<td>No label (Result display slot setting)</td>
<td>:DISPlay:RESult:EDIagram:SLOT</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:DISPlay:RESult:EDIagram:SLOT?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:DISPlay:RESult:EDIagram:TEMPplate:SELect?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:DISPlay:RESult:EDIagram:SCALe:VOLTage?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:DISPlay:RESult:EDIagram:SCALe:PHASEe?</td>
</tr>
</tbody>
</table>

For commands description, refer to 7.12.10.2.1 “Graph display setting commands”.

![Figure 7.12.16.3-2  Graph Display Field](image-url)
7.12.16.3.1 Condition setting commands

Table 7.12.16.3.1-1 Condition Tab Setting Commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>:DISPlay:RESuIt:EDIagram:DISPlay:MASK?</td>
</tr>
</tbody>
</table>

For commands description, refer to 7.12.10.2.2 “Condition setting commands”.
7.12.16.3.2 Eye Contour tab (Status) setting commands

Table 7.12.16.3.2-1 Diagram Tab (Status) Setting Commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>[4]</td>
<td>No label (Setting of time display)</td>
<td>:SENSe:MEASure:EDIagram:TIME</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSe:MEASure:EDIagram:TIME?</td>
</tr>
<tr>
<td>[6]</td>
<td>Query for measurement start time</td>
<td>:SENSe:MEASure:EDIagram:STARt?</td>
</tr>
<tr>
<td>[7]</td>
<td>Query for measurement elapsed time</td>
<td>:SENSe:MEASure:EDIagram:ELAPsed?</td>
</tr>
<tr>
<td>[8]</td>
<td>Query for measurement status</td>
<td>:CALCulate:DATA:EDIagram:STATus?</td>
</tr>
</tbody>
</table>

For commands description, refer to 7.12.10.2.4 “Diagram tab (Status) setting commands”.
7.12.16.3.3 Mask Edit tab setting commands

Table 7.12.16.3.3-1 Mask Edit Tab Setting Commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>:DISPlay:RESult:EDIagram:TEMPLate:PATTern?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:DISPlay:RESult:EDIagram:TEMPLate:SCALe?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:DISPlay:RESult:EDIagram:TEMPLate:PSCale?</td>
</tr>
<tr>
<td>[7]</td>
<td>Phase (mUI) Threshold</td>
<td>:DISPlay:RESult:EDIagram:TEMPLate:EDIT</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:DISPlay:RESult:EDIagram:TEMPLate:EDIT?</td>
</tr>
</tbody>
</table>

For commands description, refer to 7.12.10.2.5 “Mask Edit tab setting commands”.
Chapter 7  SCPI Commands

7.12.16.3.4  File menu setting commands

Table 7.12.16.3.4-1  File Menu Setting Commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
</table>

For commands description, refer to 7.12.10.2.6 “File menu setting commands”.

7.12.16.3.5  Result Query Command

For details of command, refer to 7.12.10.2.3 “Result query command”.

Table 7.12.16.3.5-1  Result Query Command

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Item</th>
<th>Command</th>
</tr>
</thead>
</table>

For details of command, refer to 7.12.10.2.3 “Result query command”.
Chapter 8  Native Commands

This chapter describes the Native commands specific to the MP1800A.

8.1  Common Commands ................................................. 8-2
  8.1.1  Commands for common settings ............................ 8-2
  8.1.2  Commands related to File menu ............................. 8-16
  8.1.3  Common function button ................................... 8-20
  8.1.4  Auto Search .................................................. 8-25

8.2  Synthesizer Commands........................................... 8-31
  8.2.1  Operation Frequency setting commands ................. 8-33
  8.2.2  Reference Clock setting commands ..................... 8-37
  8.2.3  Spectrum Spread setting commands ..................... 8-38
  8.2.4  Jitter setting commands .................................. 8-40

8.3  PPG Commands ....................................................... 8-49
  8.3.1  Commands related to Output tab ......................... 8-49
  8.3.2  Commands related to Pattern tab ....................... 8-78
  8.3.3  Commands related to Error Addition tab ............. 8-147
  8.3.4  Commands related to Misc tab ............................ 8-155

8.4  ED Commands .......................................................... 8-174
  8.4.1  Commands related to Result tab ......................... 8-174
  8.4.2  Commands related to Measurement tab ................. 8-211
  8.4.3  Commands related to Pattern tab ....................... 8-231
  8.4.4  Commands related to Input tab ........................... 8-298
  8.4.5  Commands related to Capture tab ....................... 8-314
  8.4.6  Commands related to Misc tab ............................ 8-325

8.5  Automatic Measurement Commands ............................ 8-338
  8.5.1  Eye Margin measurement setting
         commands .................................................. 8-338
  8.5.2  Eye Diagram measurement commands .................. 8-347
  8.5.3  Q measurement ............................................... 8-389
8.1 Common Commands

This section describes the commands related to common settings and functions of the control software. Before executing a setting/query command, specify the slot number of the module to be remotely controlled, by using the :MODule:ID command. Refer to Chapter 7.1 "Common Commands" and 9.1 "Basic Operation of Device Messages" for how to specify a slot number with the :MODule:ID command.

8.1.1 Commands for common settings

<table>
<thead>
<tr>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>Query for software status</td>
<td>CND?</td>
</tr>
<tr>
<td>Query for hardware system configuration</td>
<td>HDR?</td>
</tr>
<tr>
<td>Query for system error</td>
<td>INF?</td>
</tr>
<tr>
<td>Terminator type</td>
<td>TRM?</td>
</tr>
<tr>
<td>Query for model name of mainframe and module</td>
<td>CUN?</td>
</tr>
<tr>
<td>Query for mainframe information</td>
<td>UNT?</td>
</tr>
<tr>
<td>Query for module information</td>
<td>MDN?</td>
</tr>
</tbody>
</table>
8.1 Common Commands

CND?

Response

<mainframe>,<slot1>,...,<slot6>
<mainframe> = <serial>,<mver>,<hver>,<opt1>,<sbver>,<saver>,<opt2>
<serial> = <STRING RESPONSE DATA>
XXXXXXXXXX (FIX10) 0000000000 to 9999999999
Main frame serial number

Note:
Alphabetic characters may be included.

<mver> = <STRING RESPONSE DATA>
XXXX.XX.XX (FIX10) 1.00.00 to 9999.99.99
Main application software version

<hver> = <STRING RESPONSE DATA>
XXXX.XX.XX (FIX10) 1.00.00 to 9999.99.99
Mainframe hardware version

<opt1> = <STRING RESPONSE DATA>
OPTXXX (FIX6) Option number (MP1800A/MT1810A)
See Table 8.1.1-2 "Option character correspondence table"

Note:
Outputs the numbers for all installed options.
NONE is output if no option is installed.

<sbver> = <STRING RESPONSE DATA>
XXXX.XX.XX (FIX10) 1.00.00 to 9999.99.99
Sub application software version (Boot part)

<saver> = <STRING RESPONSE DATA>
XXXX.XX.XX (FIX10) 1.00.00 to 9999.99.99
Sub application software version
(Application part)

<opt2> = <STRING RESPONSE DATA>
[slotx] = <module>,<serial>,<fpga1>[,<fpga2>],<boot>,<application>,<opt>
x indicates a slot number. The slot number varies depending on the unit number as follows.

Unit 1: 1 to 16
For the MP1800A, numbers from 1 to 6 correspond to actual slots. For the MT1810A, numbers from 1 to 4 correspond to actual slots.

Unit 2: 17 to 32
For the MT1810A, numbers from 17 to 20 correspond to actual slots.

Unit 3: 33 to 48
For the MT1810A, numbers from 33 to 36 correspond to actual slots.

Unit 4: 49 to 64
For the MT1810A, numbers from 49 to 52 correspond to actual slots.
Chapter 8  Native Commands

<module> = <STRING RESPONSE DATA>
XXXXX (FIX9)  See Table 8.1.1-2 "Option character correspondence table".

Note:
NONE is output if no module is installed.
For a unit that uses two slots, only the slot with the greater number is valid.

<serial> = <STRING RESPONSE DATA>
XXXXX (FIX10)  0000000000 to 9999999999
Serial number

Note:
"-------------" is output if no module is installed.
For a module that uses two slots, only the slot with the greater number is valid.

<fpga1>[,<fpga2>,.....] = <STRING RESPONSE DATA>
XXXX.XX.XX (FIX10)  1.00.00 to 9999.99.99
FPGA version

<boot> = <STRING RESPONSE DATA>
XXXX.XX.XX (FIX10)  1.00.00 to 9999.99.99
Logic Boot version

Note:
"-------------" is output if Logic Boot is not installed.
For a module that uses two slots, only the slot with the greater number is valid.

<application> = <STRING RESPONSE DATA>
XXXX.XX.XX (FIX10)  1.00.00 to 9999.99.99
Logic Application version

Note:
"-------------" is output if Logic Boot is not installed.
For a module that uses two slots, only the slot with the greater number is valid.

<opt> = <STRING RESPONSE DATA>
XXXXX (FIX6)  Option number
OPTXXX: For MP1800A/MT1810A

Note:
Outputs the numbers for all installed options.
NONE is output if no option is installed.
For a module that uses two slots, only the slot with the greater number is valid.
The option names of the MU181020B-003 and MU181040B-003 are not output.
### 8.1 Common Commands

<table>
<thead>
<tr>
<th>Function</th>
<th>Queries the software status of the MP1800A/MT1810A.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example</td>
<td>&gt; CND?</td>
</tr>
<tr>
<td></td>
<td>&lt; CND 6201234567, 1.00.00, 1.00.20,OPT301,OPT302,</td>
</tr>
<tr>
<td></td>
<td>1.00.00, 1.00.00,OPT 12,</td>
</tr>
<tr>
<td></td>
<td>MU181000A, 6201234568, 0001.00.00,0001.00.00,0001.00.00,OPT</td>
</tr>
<tr>
<td></td>
<td>T101,</td>
</tr>
<tr>
<td></td>
<td>MU181020A, 6201234569, 0001.00.00,0001.00.00,0001.00.00,OPT</td>
</tr>
<tr>
<td></td>
<td>T001,</td>
</tr>
<tr>
<td></td>
<td>OPT220,</td>
</tr>
<tr>
<td></td>
<td>MU181040A, 6201234571, 0001.00.00,0001.00.00,0001.00.00,OPT</td>
</tr>
<tr>
<td></td>
<td>T002,</td>
</tr>
<tr>
<td></td>
<td>OPT220</td>
</tr>
</tbody>
</table>

| Compatibility     | Incompatible with existing models.                  |
### Table 8.1.1-2 Option character correspondence table

<table>
<thead>
<tr>
<th>Model/Name</th>
<th>Option Number</th>
<th>Option Name</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Model/Name</strong></td>
<td><strong>Option Number</strong></td>
<td><strong>Option Name</strong></td>
</tr>
<tr>
<td>MP1800A Signal Quality Analyzer</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MT1810A 4 Slot Chassis</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MX180000A Signal Quality Analyzer Control</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MU181000A 12.5 GHz Synthesizer</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MU181000B 12.5 GHz 4 port Synthesizer</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MU181020A 12.5 Gbit/s PPG</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MU181040A 12.5 Gbit/s ED</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MU181600A Optical Transceiver (XFP)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MU181601A Optical Transceiver (SFP)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MU181620A Stressed Eye Transmitter</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MU181640A Optical Receiver</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MU181800A 12.5GHz Clock Distributor</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Table 8.1.1-2 Option character correspondence table (Cont’d)

<table>
<thead>
<tr>
<th>Model/Name</th>
<th>Option Number</th>
<th>Option Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>MU181020B 14 Gbit/s PPG</td>
<td>OPT002</td>
<td>0.1 to 14 Gbit/s</td>
</tr>
<tr>
<td></td>
<td>OPTx11</td>
<td>Variable Data Output (0.25 to 2.5 Vp-p)</td>
</tr>
<tr>
<td></td>
<td>OPTx12</td>
<td>High Performance Data Output (0.05 to 2.0 Vp-p)</td>
</tr>
<tr>
<td></td>
<td>OPTx13</td>
<td>Variable Data Output (0.5 to 3.5 Vp-p)</td>
</tr>
<tr>
<td></td>
<td>OPTx21</td>
<td>Differential Clock Output (0.1 to 2.0 Vp-p)</td>
</tr>
<tr>
<td></td>
<td>OPTx30</td>
<td>Variable Data Delay</td>
</tr>
<tr>
<td>MU181040B 14 Gbit/s ED</td>
<td>OPT002</td>
<td>0.1 to 14 Gbit/s</td>
</tr>
<tr>
<td></td>
<td>OPTx20</td>
<td>Clock Recovery</td>
</tr>
<tr>
<td></td>
<td>OPTx30</td>
<td>Variable Clock Delay</td>
</tr>
<tr>
<td>MU182020A 25 Gbit/s 1ch MUX MU182021A 25 Gbit/s 2ch MUX</td>
<td>OPTx01</td>
<td>28 Gbit/s Extension</td>
</tr>
<tr>
<td></td>
<td>OPTx02</td>
<td>Clock Input Band Switch</td>
</tr>
<tr>
<td></td>
<td>OPTx10</td>
<td>Variable Data Output (0.25 to 1.75Vp-p)</td>
</tr>
<tr>
<td></td>
<td>OPTx11</td>
<td>Variable Data Output (0.5 to 2.5Vp-p)</td>
</tr>
<tr>
<td></td>
<td>OPTx12</td>
<td>Variable Data Output (0.5 to 2.0Vp-p)</td>
</tr>
<tr>
<td></td>
<td>OPTx13</td>
<td>Variable Data Output (0.5 to 3.5Vp-p)</td>
</tr>
<tr>
<td></td>
<td>OPTx21</td>
<td>Differential Clock Output (0.5 to 2.0Vp-p)</td>
</tr>
<tr>
<td></td>
<td>OPTx30</td>
<td>25Gbit/s Variable Data Delay</td>
</tr>
<tr>
<td></td>
<td>OPTx31</td>
<td>28Gbit/s Variable Data Delay</td>
</tr>
<tr>
<td>MU182040A 25 Gbit/s 1ch DEMUX MU182041A 25 Gbit/s 2ch DEMUX</td>
<td>OPTx01</td>
<td>28Gbit/s Extension</td>
</tr>
<tr>
<td></td>
<td>OPTx02</td>
<td>Clock Input Band Switch</td>
</tr>
<tr>
<td></td>
<td>OPTx30</td>
<td>25GHz Variable Clock Delay</td>
</tr>
<tr>
<td></td>
<td>OPTx31</td>
<td>28GHz Variable Clock Delay</td>
</tr>
</tbody>
</table>

**Note:**
The option names of the MU181020B-003 and MU181040B-003 are not output.
Chapter 8  Native Commands

HDR?

Response  

\(<\text{slot}_1>,...,<\text{slot}_6>\)

\(<\text{slot}_x> = \langle\text{module}\rangle,\langle\text{serial}\rangle,\langle\text{fpga}_1\rangle[,\langle\text{fpga}_2\rangle],\langle\text{boot}\rangle,\langle\text{application}\rangle,\langle\text{opt}\rangle\)

\(x\) indicates a slot number. The slot number varies depending on the unit number as follows.

**Unit 1: 1 to 16**

For the MP1800A, numbers from 1 to 6 correspond to actual slots.

For the MT1810A, numbers from 1 to 4 correspond to actual slots.

**Unit 2: 17 to 32**

For the MT1810A, numbers from 17 to 20 correspond to actual slots.

**Unit 3: 33 to 48**

For the MT1810A, numbers from 33 to 36 correspond to actual slots.

**Unit 4: 49 to 64**

For the MT1810A, numbers from 49 to 52 correspond to actual slots.

\(<\text{module}> = \langle\text{STRING RESPONSE DATA}\rangle\)

XXXXXXXX (FIX9) Module name

See Table 8.1.1-2 "Option character correspondence table".

**Note:**

NONE is output if no module is installed.

For a unit that uses two slots, only the slot with the lower number is valid.

\(<\text{serial}> = \langle\text{STRING RESPONSE DATA}\rangle\)

XXXXXXXXXXX (FIX10) 0000000000 to 9999999999

Serial number

**Note:**

"-------------" is output if no module is installed.

For a unit that uses two slots, only the slot with the lower number is valid.

\(<\text{fpga}_1\rangle[,<\text{fpga}_2\rangle,.....] = \langle\text{STRING RESPONSE DATA}\rangle\)

XXXX.XX.XX (FIX10) 1.00.00 to 9999.99.99

FPGA version

\(<\text{boot}> = \langle\text{STRING RESPONSE DATA}\rangle\)

XXXX.XX.XX (FIX10) 1.00.00 to 9999.99.99

Logic Boot version

**Note:**

"-------------" is output if Logic Boot is not installed.

For a module that uses two slots, only the slot with the lower number is valid.
### 8.1 Common Commands

<application> = <STRING RESPONSE DATA>

XXXX.XX.XX (FIX10) 1.00.00 to 9999.99.99
Logic Application version

**Note:**

"-------------" is output if Logic Application is not installed.
For a module that uses two slots, only the slot with the lower number is valid.

<opt> = <STRING RESPONSE DATA>

XXXXXX (FIX6) Option number

OPTXXX: For MP1800A/MT1810A

**Note:**

Outputs the numbers for all installed options.
NONE is output if no option is installed.
For a module that uses two slots, only the slot with the lower number is valid.
The option names of the MU181020B-003 and MU181040B-003 are not output.

<table>
<thead>
<tr>
<th>Function</th>
<th>Example</th>
</tr>
</thead>
</table>
| Queries the hardware system configuration of the MP1800A/MT1810A. To query the software status of the MP1800A: | > HDR?  
< HDR  
"MU181000A, 6201234568, 0001.00.00, 0001.00.00, 0001.00.00, OPT101,  
MU181020A, 6201234569, 0001.00.00, 0001.00.00, 0001.00.00, OPT001,  
OPT220,  
MU181040A, 6201234571, 0001.00.00, 0001.00.00, 0001.00.00, OPT002,  
OPT220"

| Compatibility | Partially compatible with the MP1632C and MP1776A. |

**Compatibility**

Partially compatible with the MP1632C and MP1776A.
### INF? <unit>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;unit&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1 to 4 Mainframe number</td>
</tr>
<tr>
<td>Response</td>
<td>&lt;numeric&gt; = &lt;DEFINITE LENGTH ARBITRARY BLOCK RESPONSE DATA&gt;</td>
</tr>
<tr>
<td></td>
<td>#B0000 NONE</td>
</tr>
<tr>
<td></td>
<td>#B100 PLL Unlock</td>
</tr>
<tr>
<td></td>
<td>#B0100 Temperature</td>
</tr>
<tr>
<td></td>
<td>#B0010 Fan</td>
</tr>
<tr>
<td>Function</td>
<td>Queries the content of the system error that has currently occurred.</td>
</tr>
<tr>
<td>Example</td>
<td>To query the content of the system error that has occurred in Mainframe 1:</td>
</tr>
<tr>
<td></td>
<td>&gt; INF? 1</td>
</tr>
<tr>
<td></td>
<td>&lt; INF #B1000</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Incompatible with existing models.</td>
</tr>
</tbody>
</table>

### TRM <numeric>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;numeric&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Switches the terminator type to LF+EOI.</td>
</tr>
<tr>
<td>1</td>
<td>Switches the terminator type to CR+LF+EOI.</td>
</tr>
<tr>
<td>Function</td>
<td>Sets the terminator type of the response data.</td>
</tr>
<tr>
<td>Example</td>
<td>To set the terminator type to LF+EOI:</td>
</tr>
<tr>
<td></td>
<td>&gt; TRM 0</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Compatible with the MP1632C.</td>
</tr>
</tbody>
</table>

### TRM? 

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;numeric&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt; (FIX1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Terminator type LF+EOI</td>
</tr>
<tr>
<td>1</td>
<td>Terminator type CR+LF+EOI</td>
</tr>
<tr>
<td>Function</td>
<td>Queries the terminator type of the response data.</td>
</tr>
<tr>
<td>Example</td>
<td>&gt; TRM?</td>
</tr>
<tr>
<td></td>
<td>&lt; TRM 0</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Compatible with the MP1632C.</td>
</tr>
</tbody>
</table>
8.1 Common Commands

CUN?

<table>
<thead>
<tr>
<th>Function</th>
<th>Queries the model name of the MP1800A/MT1810A and module.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example</td>
<td>&gt; CUN?</td>
</tr>
<tr>
<td></td>
<td>&lt; NONE,MU181000A,NONE,MU181020A,</td>
</tr>
<tr>
<td></td>
<td>MU181040A,NONE,...,MU181600A,MU181601A,MU181620A,NONE,</td>
</tr>
<tr>
<td></td>
<td>MU181640A,NONE,...,NONE</td>
</tr>
<tr>
<td>Example</td>
<td>To query the model name of the MP1800A/MT1810A and module for the MP1800A (1-box type):</td>
</tr>
<tr>
<td></td>
<td>&gt; CUN?</td>
</tr>
<tr>
<td></td>
<td>&lt; MP1800A,NONE,NONE,NONE,MU181000A,NONE,MU181020A,</td>
</tr>
<tr>
<td></td>
<td>MU181040A,NONE,...,NONE,MU181600A,NONE,...,NONE</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Incompatible with existing models.</td>
</tr>
</tbody>
</table>

Response

<unit1>,...,<unit4>,<slot1>,...,<slot6>

=","<mainframe1>,...,<mainframe4>,<module1>,...,<module6>

<mainframe1> to <mainframe4> = <STRING RESPONSE DATA>

XXXXXXXXXX Main frame name (e.g. MT1800A)

See Table 8.1.1-2 "Option character correspondence table".

Note:

NONE is output for mainframe2 to mainframe4, if no mainframe is connected.

<module1> to <module6> = <STRING RESPONSE DATA>

XXXXXXXXXX Module name (e.g. MU181020A)

See Table 8.1.1-2 "Option character correspondence table".

Note:

NONE is output if no module is installed.

For a unit that uses two slots, only the slot with the lower number is valid.
Chapter 8  Native Commands

UNT? <numeric>

Parameter

\(<\text{numeric}> = \langle\text{NR1 NUMERIC PROGRAM DATA}\rangle
1 \text{ to } 4 \quad \text{Mainframe number}
\)

For the MP1800A set to "1" and for the MT1810A set to 1 to 4.

Response

"\langle\text{mainframe}\rangle."

\(\langle\text{mainframe}\rangle = \langle\text{module}\rangle,\langle\text{serial}\rangle,\langle\text{mver}\rangle,\langle\text{hver}\rangle,\langle\text{opt1}\rangle,\langle\text{sbver}\rangle,\langle\text{saver}\rangle,\langle\text{opt2}\rangle\)

\(\langle\text{module}\rangle = \langle\text{STRING RESPONSE DATA}\rangle\)

\(XXXXXXX\quad \text{Mainframe name (e.g. MP1800A, MT1810A)}\)

See Table 8.1.1-2 "Option character correspondence table".

Note:

NONE is output if no module is installed.

For a unit that uses two slots, only the slot with the lower number is valid.

\(\langle\text{serial}\rangle = \langle\text{STRING RESPONSE DATA}\rangle\)

\(XXXXXXXXXXX\quad 0000000000 \text{ to } 9999999999\)

Main frame serial number

Note:

Alphabetic characters may be included.

\(\langle\text{mver}\rangle = \langle\text{STRING RESPONSE DATA}\rangle\)

\(XXXX.XX.XX\quad 1.00.00 \text{ to } 9999.99.99\)

Main application software version

\(\langle\text{hver}\rangle = \langle\text{STRING RESPONSE DATA}\rangle\)

\(XXXX.XX.XX\quad 1.00.00 \text{ to } 9999.99.99\)

Mainframe hardware version

\(\langle\text{opt1}\rangle = \langle\text{STRING RESPONSE DATA}\rangle\)

OPTXXX Option number (MP1800A/MT1810A)

See Table 8.1.1-2 "Option character correspondence table".

Note:

Outputs the numbers for all installed options.

NONE is output if no option is installed.

\(\langle\text{sbver}\rangle = \langle\text{STRING RESPONSE DATA}\rangle\)

\(XXXX.XX.XX\quad 1.00.00 \text{ to } 9999.99.99\)

Sub application software version (Boot part)

\(\langle\text{saver}\rangle = \langle\text{STRING RESPONSE DATA}\rangle\)

\(XXXX.XX.XX\quad 1.00.00 \text{ to } 9999.99.99\)

Sub application software version (Application part)

Function

Queries the mainframe information including model and serial number.
8.1 Common Commands

Example

To query the Mainframe 1 information:

> UNT? 1

<

"MP1800A,6201234568,1.00.00,1.00.00,OPT01,1.00.00,1.00.00"

Compatibility

Incompatible with existing models.
### Chapter 8  Native Commands

#### MDN? <numeric>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| <numeric> | = NR1 NUMERIC PROGRAM DATA >  
1 to 6 | Slots 1 to 6  
1 to 6 when using the MP1800A, 1 to 4 when using the MT1810A  

<table>
<thead>
<tr>
<th>Response</th>
<th>Description</th>
</tr>
</thead>
</table>
| <slot>   | =  
"<module>,<serial>,<fpga1>,<fpga2>,<boot>,<application>,<opt>"  
| <module>  | = STRING RESPONSE DATA >  
| XXXXXXXXX | Module name (e.g. MU181020A)  
See Table 8.1.1-2 "Option character correspondence table".  

**Note:**  
NONE is output if no module is installed.  
For a unit that uses two slots, only the slot with the lower number is valid.  

| <serial> | STRING RESPONSE DATA >  
| XXXXXXXXX | 0000000000 to 9999999999  
Serial number  

**Note:**  
"-------------" is output if no module is installed.  
For a unit that uses two slots, only the slot with the lower number is valid.  

| <fpga1>,<fpga2>,...... | STRING RESPONSE DATA >  
| XXXX.XX.XX | 1.00.00 to 9999.99.99  
FPGA version  
| <boot> | STRING RESPONSE DATA >  
| XXXX.XX.XX | 1.00.00 to 9999.99.99  
Logic Boot version  

**Note:**  
"-------------" is output if Logic Boot is not installed.  
For a unit that uses two slots, only the slot with the lower number is valid.  

| <application> | STRING RESPONSE DATA >  
| XXXX.XX.XX | 1.00.00 to 9999.99.99  
Logic Application version  

**Note:**  
"-------------" is output if Logic Boot is not installed.  
For a unit that uses two slots, only the slot with the lower number is valid.  

| <opt> | STRING RESPONSE DATA >  
| XXXXXXX/XXXXX | Option number  
OPTXXX: For MP1800A/MT1810A  

8.1 Common Commands

**Note:**
- Outputs the numbers for all installed options.
- NONE is output if no option is installed.
- For a unit that uses two slots, only the slot with the lower number is valid.
- The option names of the MU181020B-003 and MU181040B-003 are not output.

<table>
<thead>
<tr>
<th>Function</th>
<th>Queries the module information on the specified slot.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example</td>
<td>To query the module information on Slot 3:</td>
</tr>
<tr>
<td></td>
<td>&gt; MDN? 3</td>
</tr>
<tr>
<td></td>
<td>&lt; MDN</td>
</tr>
<tr>
<td></td>
<td>&quot;MU181020A,6201234568,1.00.00,-----,1.00.00,1.00.00,OPT002,OPT211,OPT220, &quot;</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Incompatible with existing models.</td>
</tr>
</tbody>
</table>
8.1.2 Commands related to File menu

Table 8.1.2-1  File Menu Setting Command

<table>
<thead>
<tr>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>Screen Copy</td>
<td>HCP</td>
</tr>
<tr>
<td>Combination Setting</td>
<td>CST?</td>
</tr>
<tr>
<td></td>
<td>CMA?</td>
</tr>
<tr>
<td></td>
<td>CST</td>
</tr>
<tr>
<td></td>
<td>CHA?</td>
</tr>
<tr>
<td></td>
<td>CHS</td>
</tr>
<tr>
<td>Initialize</td>
<td>INI</td>
</tr>
</tbody>
</table>

HCP

Function

Takes a screen shot.

Example

> HCP

Compatibility

Compatible with the MP1632C.

CST? <slot>[,<unit>]

Parameter

<slot> = <DECIMAL NUMERIC PROGRAM DATA>
1 to 6  Slot No.1 to 6
0      All modules (units)

When using the MP1800A: 1 to 6, when using the MT1810A: 1 to 4

[unit] = <DECIMAL NUMERIC PROGRAM DATA>
1 to 4  Mainframe No. 1 to 4

When using two or more MT1810A units in serial connection, specify the mainframe number.

Can be omitted. Mainframe No. 1 is specified when omitted.

Response

<numeric> = <NR1 NUMERIC RESPONSE DATA> <FIX4>
0      Independent
1      Channel Synchronization
21     2 Ch PPG Combination
22     2 Ch ED Combination
23     2 Ch PPG/ED Combination
41     4 Ch PPG Combination
42     4 Ch ED Combination
71     25G × 2 Ch PPG Combination
72     25G × 2 Ch ED Combination

Function

Queries the combination setting of the specified slot.

Example

To query the combination setting of Unit 1:

> CST?  0
< CST    0

To query the combination setting of Slot 3 in Unit 2:
8.1 Common Commands

> CST? 3,2
< CST 41

Compatibility
Incompatible with existing models.

CMA? [<unit>]

Parameter

- `<unit>` = <DECIMAL NUMERIC PROGRAM DATA>
- 1 to 4 Mainframe No. 1 to 4
When using two or more MT1810A units in serial connection, specify the mainframe number.
Can be omitted. Mainframe No. 1 is specified when omitted.

Response

- `<numeric>` = <NR1 NUMERIC RESPONSE DATA> <FIX4>
- 0 Independent
- 21 2 Ch PPG Combination
- 22 2 Ch ED Combination
- 23 2 Ch PPG/ED Combination
- 41 4 Ch PPG Combination
- 42 4 Ch ED Combination
- 71 25G × 2 Ch PPG Combination
- 72 25G × 2 Ch ED Combination

Function
Queries the available combination configuration.

Example
To query the combination configuration available for Unit 1:
> CMA?
< CMA 41

Compatibility
Incompatible with existing models.
**CST <configuration>,<unit>**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;configuration&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt;</td>
<td>0: Independent&lt;br&gt;21: 2 Ch PPG Combination&lt;br&gt;22: 2 Ch ED Combination&lt;br&gt;23: 2 Ch PPG/ED Combination&lt;br&gt;41: 4 Ch PPG Combination&lt;br&gt;42: 4 Ch ED Combination&lt;br&gt;71: 25G × 2 Ch PPG Combination&lt;br&gt;72: 25G × 2 Ch ED Combination</td>
</tr>
</tbody>
</table>

[<unit>] = <DECIMAL NUMERIC PROGRAM DATA>

1 to 4: Mainframe No. 1 to 4

When using two or more MT1810A units in serial connection, specify the mainframe number.

Can be omitted. Mainframe No. 1 is specified when omitted.

**Function**

Sets the combination setting for the specified unit, from Combination or Independent.

**Example**

To set the combination setting of Unit 3 to 4-ch PPG combination:

```
> CST 41,3
```

**Compatibility**

Incompatible with existing models.

---

**CHA? [<unit>]**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| [<unit>] = <DECIMAL NUMERIC PROGRAM DATA> | 1 to 4: Mainframe No. 1 to 4

When using two or more MT1810A units in serial connection, specify the mainframe number.

Can be omitted. Mainframe No. 1 is specified when omitted.

Response

<numeric> = <NR1 NUMERIC RESPONSE DATA> <FIX4>

0 to 64: Total number of PPGs that can configure channel synchronization (decimal)

Available bits:

0: No PPG that can configure channel synchronization
1 (Bit 0): PPG in Slot 1
2 (Bit 1): PPG in Slot 2
4 (Bit 2): PPG in Slot 3
8 (Bit 3): PPG in Slot 4
16 (Bit 4): PPG in Slot 5
32 (Bit 5): PPG in Slot 6

**Function**

Queries the slot where PPG that can configure channel synchronization combination is inserted.

**Example**

To query the slot in Unit 3 where PPG that can configure channel synchronization is inserted:
8.1 Common Commands

> CHA? 3
< CHA 7

Compatibility
Incompatible with existing models.

### CHS <configuration>[,<unit>]

**Parameter**

- `<configuration>` = `<NR1 NUMERIC PROGRAM DATA>`
  - 0 to 64: Total number of PPGs that can configure channel synchronization (decimal)
  - Available bits:
    - 2 (Bit 1): PPG in Slot 2
    - 4 (Bit 2): PPG in Slot 3
    - 8 (Bit 3): PPG in Slot 4
    - 16 (Bit 4): PPG in Slot 5
    - 32 (Bit 5): PPG in Slot 6

- `<unit>` = `<DECIMAL NUMERIC PROGRAM DATA>`
  - 1 to 4: Mainframe No. 1 to 4

When using two or more MT1810A units in serial connection, specify the mainframe number.
Can be omitted. Mainframe No. 1 is specified when omitted.

**Function**
Specify the slot where the PPG for which channel synchronization is to be set is inserted.

**Example**
To set channel synchronization for the PPGs in Slots 1 through 4 of Unit 3:
> CHS 14,3

**Compatibility**
Incompatible with existing models.

### INI

**Function**
Initializes the internal setting data to the initial settings at factory shipment.

**Example**
> INI

**Compatibility**
Compatible with the MP1632C and MP1776A.
8.1.3 Common function button

Table 8.1.3-1 Common Function Button Setting Commands

<table>
<thead>
<tr>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data/Clock Output ON/OFF</td>
<td>OON</td>
</tr>
<tr>
<td></td>
<td>OON?</td>
</tr>
<tr>
<td>Error Addition ON/OFF</td>
<td>EAS</td>
</tr>
<tr>
<td></td>
<td>EAS?</td>
</tr>
<tr>
<td>Single Error Addition</td>
<td>EAI</td>
</tr>
<tr>
<td>Meas. Start</td>
<td>SAT</td>
</tr>
<tr>
<td>Meas. Stop</td>
<td>SOT</td>
</tr>
<tr>
<td>(Query for measurement status)</td>
<td>STT?</td>
</tr>
<tr>
<td>Buzzer Setting Error ON/OFF</td>
<td>MON</td>
</tr>
<tr>
<td></td>
<td>MON?</td>
</tr>
<tr>
<td>Buzzer Setting Alarm ON/OFF</td>
<td>ALM</td>
</tr>
<tr>
<td></td>
<td>ALM?</td>
</tr>
<tr>
<td>Buzzer Setting System Error ON/OFF</td>
<td>SYS</td>
</tr>
<tr>
<td></td>
<td>SYS?</td>
</tr>
<tr>
<td>Buzzer Setting System Error</td>
<td>SYT</td>
</tr>
<tr>
<td></td>
<td>SYT?</td>
</tr>
</tbody>
</table>
8.1 Common Commands

**OON <numeric>**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;numeric&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Data and Clock output OFF</td>
</tr>
<tr>
<td>1</td>
<td>Data and Clock output ON</td>
</tr>
</tbody>
</table>

**Function**
Sets ON or OFF of Data and Clock outputs of optical PPG.

**Example**
To set Data and Clock outputs of optical PPG to ON:
> OON 1

**Compatibility**
Compatible with the MP1632C and MP1761C Pulse Pattern Generator (hereinafter referred to as MP1761C).

**OON?**

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;numeric&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt; (FIX)</th>
</tr>
</thead>
</table>

**Function**
Queries the ON/OFF state for Data and Clock outputs of optical PPG.

**Example**
> OON?
< OON 1

**Compatibility**
Compatible with the MP1632C and MP1761C.

**EAS <numeric>**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;numeric&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Error addition OFF</td>
</tr>
<tr>
<td>1</td>
<td>Error addition ON</td>
</tr>
</tbody>
</table>

**Function**
Sets error addition for all valid modules ON/OFF.

**Example**
To set error addition for all valid modules ON:
> EAS 1

**Compatibility**
Incompatible with existing models.

**EAS?**

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;numeric&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt; (FIX)</th>
</tr>
</thead>
</table>

**Function**
Queries the error addition ON/OFF state for all valid modules.

**Example**
> EAS?
< EAS 1

**Compatibility**
Incompatible with existing models.

**EAI**

**Function**
Adds an error for all valid modules.

**Example**
> EAI

**Compatibility**
Incompatible with existing models.
## Chapter 8  Native Commands

### SAT

**Function**
Starts measurement for all modules.

**Example**
> SAT

**Compatibility**
Incompatible with existing models.

### SOT

**Function**
Stops measurement for all modules.

**Example**
> SOT

**Compatibility**
Incompatible with existing models.

### STT?

**Response**
\(<\text{numeric}> = \text{<NR1 NUMERIC RESPONSE DATA>} \) (FIX1)

- 0: Measurement stops for all modules
- 1: During measurement

**Note:**
If any module is being measured, the response is deemed as being "during measurement".

**Function**
Queries the measurement state for all modules.

**Example**
> STT?
< STT 0

**Compatibility**
Incompatible with existing models.

### MON <numeric>

**Parameter**
\(<\text{numeric}> = \text{<DECIMAL NUMERIC PROGRAM DATA>} \)

- 0: Buzzer OFF
- 1: Buzzer ON

**Function**
Sets buzzer at error occurrence ON/OFF.

**Example**
To set buzzer at error occurrence ON:
> MON 1

**Compatibility**
Compatible with the MP1632C and MP1762D Error Detector (hereinafter referred to as MP1762D).

### MON?

**Response**
\(<\text{numeric}> = \text{<NR1 NUMERIC RESPONSE DATA>} \) (FIX1)

**Function**
Queries the buzzer ON/OFF state at error occurrence.

**Example**
> MON?
< MON 1

**Compatibility**
Compatible with the MP1632C and MP1762D.
### ALM <numeric>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;numeric&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Buzzer OFF</td>
</tr>
<tr>
<td>1</td>
<td>Buzzer ON</td>
</tr>
</tbody>
</table>

**Function**
Sets buzzer at alarm occurrence ON/OFF.

**Example**
To set buzzer at alarm occurrence OFF:
> ALM 0

**Compatibility**
Compatible with the MP1632C and MP1762D.

### ALM?

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;numeric&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt; (FIX1)</th>
</tr>
</thead>
</table>

**Function**
Queries the buzzer ON/OFF state at alarm occurrence.

**Example**
> ALM?
< ALM 0

**Compatibility**
Compatible with the MP1632C and MP1762D.

### SYS <numeric>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;numeric&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Buzzer OFF</td>
</tr>
<tr>
<td>1</td>
<td>Buzzer ON</td>
</tr>
</tbody>
</table>

**Function**
Sets buzzer at system error occurrence ON/OFF.

**Example**
To set buzzer at system error occurrence ON:
> SYS 1

**Compatibility**
Compatible with the MP1632C.

### SYS?

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;numeric&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt; (FIX1)</th>
</tr>
</thead>
</table>

**Function**
Queries buzzer at system error occurrence ON/OFF.

**Example**
> SYS?
< SYS 1

**Compatibility**
Compatible with the MP1632C.
**SYT <numeric>**

**Parameter**

<numeric> = <NON-DECIMAL NUMERIC PROGRAM DATA>

- #B0000: Does not activate system error alarming buzzer for all items.
- #B1000: Activates buzzer for "PLL Unlock" only.
- #B0100: Activates buzzer for "FAN" only.
- #B0010: Activates buzzer for "Temperature" only.
- #B1111: Activates system error alarming buzzer for all items.

For items for which the buzzer is activated, set the corresponding bits to 1.
Set corresponding bits to 0 for items for which the buzzer is not activated.

**Function**
Sets system error alarming buzzer for the target item ON/OFF.

**Example**
To set system error alarming buzzer for PLL Unlock ON only:

```
> SYT #B1000
```

**Compatibility**
Partially compatible with the MP1632C.

---

**SYT?**

**Response**

<numeric> = <DEFINITE LENGTH ARBITORARY BLOCK RESPONSE DATA>

**Function**
Queries the ON/OFF state of system error alarming buzzer for target items.

**Example**

```
> SYT?
< SYT #B1000
```

**Compatibility**
Partially compatible with the MP1632C.
8.1.4 Auto Search

Figure 8.1.4-1 Auto Search dialog box
### Table 8.1.4-1  Auto Search Setting Commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1]</td>
<td>Start</td>
<td>AST</td>
</tr>
<tr>
<td>[2]</td>
<td>Stop</td>
<td>ASO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AMS?</td>
</tr>
<tr>
<td>[4]</td>
<td>Item</td>
<td>ASM</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ASM?</td>
</tr>
<tr>
<td>[5]</td>
<td>Start/Stop</td>
<td>SRH</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SRH?</td>
</tr>
<tr>
<td>[6]</td>
<td>No label (Query for slot state)</td>
<td>ASL?</td>
</tr>
<tr>
<td>[7]</td>
<td>Slot Select</td>
<td>ASE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ASE?</td>
</tr>
<tr>
<td>[8]</td>
<td>Set All</td>
<td>ASA</td>
</tr>
<tr>
<td>[9]</td>
<td>Reset All</td>
<td>ASR</td>
</tr>
<tr>
<td>[10]</td>
<td>No label (Query for results)</td>
<td>ARS?</td>
</tr>
</tbody>
</table>

#### AST

**Function**

Starts Auto Search function.

**Example**

```
> AST
```

**Compatibility**

Incompatible with existing models.

#### ASO

**Function**

Stops Auto Search function.

**Example**

```
> ASO
```

**Compatibility**

Incompatible with existing models.

#### AMS <mode>

**Parameter**

\(<\text{mode}> = \langle\text{DECIMAL NUMERIC PROGRAM DATA}\rangle\)

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Fine mode</td>
</tr>
<tr>
<td>1</td>
<td>Coarse mode</td>
</tr>
</tbody>
</table>

**Function**

Sets the Auto Search mode.

**Example**

To set the Auto Search mode to Fine mode:

```
> AMS 0
```

**Compatibility**

Incompatible with existing models.
8.1 Common Commands

**AMS?**
Response  
\(<mode> = \langle NR1 \; NUMERIC \; RESPONSE \; DATA \rangle \; (FIX1)\)
Function  
Queries the Auto Search mode.
Example  
\(> \; AMS?\)
\(< \; AMS\; 0\)
Compatibility  
Incompatible with existing models.

**ASM <mode>**
Parameter  
\(<mode> = \langle DECIMAL \; NUMERIC \; PROGRAM \; DATA \rangle\)
0 Phase & Threshold
1 Threshold
2 Phase
Function  
Sets the execution item of Auto Search.
Example  
To set the execution item of Auto Search to Phase & Threshold:
\(> \; ASM\; 0\)
Compatibility  
Compatible with the MP1632C.

**ASM?**
Response  
\(<mode> = \langle NR1 \; NUMERIC \; RESPONSE \; DATA \rangle \; (FIX1)\)
Function  
Queries the execution item of Auto Search.
Example  
\(> \; ASM?\)
\(< \; ASM\; 0\)
Compatibility  
Compatible with the MP1632C.

**SRH <numeric>**
Parameter  
\(<numeric> = \langle NR1 \; NUMERIC \; PROGRAM \; DATA \rangle\)
1 Start
0 Stop
Function  
Starts or stops the Auto Search function.
Example  
To start the Auto Search function:
\(> \; SRH\; 1\)
Compatibility  
Compatible with the MP1632C and MP1762D.
Chapter 8  Native Commands

SRH?

Response  
<numeric> = <NR1 NUMERIC RESPONSE DATA>
1  
Start
0  
Stop
−1  
Auto Search failure

Function  
Queries the Auto Search function state.

Example  
> SRH?
< SRH 1

Compatibility  
Partially compatible with the MP1632C and MP1762D.

ASL? [<unit>]

Parameter  
[<unit>] = <DECIMAL NUMERIC PROGRAM DATA>
1 to 4  
Mainframe No. 1 to 4
When using two or more MT1810A units in serial connection, specify the mainframe number.
Can be omitted. Mainframe No. 1 is specified when omitted.

Response  
<slot> = <NR1 NUMERIC RESPONSE DATA>
1 to 6  
SLOT1 to SLOT6

Note:
The above response returns all slots for which Auto Search can be executed by delimiting with commas (,).

When using the MP1800A: 1 to 6, when using the MT1810A: 1 to 4

Function  
Queries the slot for which Auto Search can be executed.

Example  
> ASL?
< ASL 1, 2, 6

Compatibility  
Incompatible with existing models.

ASE <slot>,<numeric>[,<unit>]

Parameter  
<slot> = <DECIMAL NUMERIC PROGRAM DATA>
1 to 6  
SLOT1 to SLOT6
When using the MP1800A: 1 to 6, when using the MT1810A: 1 to 4
<numeric> = <DECIMAL NUMERIC PROGRAM DATA>
1  
ON
0  
OFF
[<unit>] = <DECIMAL NUMERIC PROGRAM DATA>
1 to 4  
Mainframe No. 1 to 4
When using two or more MT1810A units in serial connection, specify the mainframe number.
Can be omitted. Mainframe No. 1 is specified when omitted.

Function  
Selects the slot for which Auto Search is to be executed.

Example  
To set Auto Search for Slot 1 to ON:
8.1 Common Commands

> ASE 1, 1

Compatibility
Incompatible with existing models.

**ASE? <slot>[,<unit>]**

| Parameter | <slot> = <DECIMAL NUMERIC PROGRAM DATA>
|           | 1 to 6 SLOT1 to SLOT6
|           | When using the MP1800A: 1 to 6, when using the MT1810A: 1 to 4
|           | [<unit>] = <DECIMAL NUMERIC PROGRAM DATA>
|           | 1 to 4 Mainframe No. 1 to 4
|           | When using two or more MT1810A units in serial connection, specify the mainframe number.
|           | Can be omitted. Mainframe No. 1 is specified when omitted.

| Response  | <numeric> = <NR1 NUMERIC RESPONSE DATA> (FIX1)
| Function  | Queries the slot for which Auto Search is to be executed.
| Example   | > ASE? 1
|           | < ASE 1

Compatibility
Incompatible with existing models.

**ASA**

Function
Sets all slots for which Auto Search is to be executed to ON.

Example
> ASA

Compatibility
Incompatible with existing models.

**ASR**

Function
Sets all slots for which Auto Search is to be executed to OFF.

Example
> ASR

Compatibility
Incompatible with existing models.
**Chapter 8  Native Commands**

**ARS? <slot>[,<unit>]**

**Parameter**

<slot> = <DECIMAL NUMERIC PROGRAM DATA>

1 to 6  SLOT 1 to SLOT 6

When using the MP1800A: 1 to 6, when using the MT1810A: 1 to 4

[unit] = <DECIMAL NUMERIC PROGRAM DATA>

1 to 4  Mainframe No. 1 to 4

When using two or more MT1810A units in serial connection, specify the mainframe number.

Can be omitted. Mainframe No. 1 is specified when omitted.

**Response**

<data>,<xdata>,<delay>,<numeric1>,<numeric2>,<numeric3>

<data>,<xdata>,<delay> = <NR1 NUMERIC RESPONSE DATA> (FIX1)

0  Normal termination
1  Abnormal termination
2  Not executed

<numeric1>,<numeric2> = <NR2 NUMERIC RESPONSE DATA> (FIX6)

–3.500 to 3.300  V

**Note:**

"-----" when not executed

<numeric3> = <NR1 NUMERIC RESPONSE DATA> (FIX5)

–1000 to 1000  mUI

**Note:**

"-----" when no executed

<numeric4> = <NR2 NUMERIC RESPONSE DATA> (FIX7)

–xxxx.xx to xxx.xx  ps

**Note:**

"-----" when no executed

**Function**

Queries the Auto Search execution result.

**Example**

> ARS? 1

< ARS 0,0,0, 3.000, 3.000, 700, 50.00

< ARS 0,0,2, 3.000, 3.000,-----,----- (Delay auto search is not executed)

**Compatibility**

Incompatible with existing models.
8.2 Synthesizer Commands

This section describes MU181000A 12.5 GHz Synthesizer and MU181000B 12.5 GHz 4 port Synthesizer (hereinafter referred to as Synthesizer) commands.

Before executing a setting/query command for the Synthesizer, specify the slot number of the module to be remotely controlled, by using the :MODule:ID command. Refer to Section 7.1 “Common Commands” and 9.1 "Basic Operation of Device Messages" for how to specify a slot number with the :MODule:ID command.

![Synthesizer Window]

**Figure 8.2-1  Synthesizer Window**
Figure 8.2-2  12.5 GHz Synthesizer Window (Option x01)
8.2 Synthesizer Commands

8.2.1 Operation Frequency setting commands

![Figure 8.2.1-1  Operation Frequency setting field]

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1]</td>
<td>Operation</td>
<td>OPE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>OPE?</td>
</tr>
<tr>
<td>[2]</td>
<td>Center Frequency</td>
<td>FRQ</td>
</tr>
<tr>
<td></td>
<td></td>
<td>FRQ?</td>
</tr>
<tr>
<td>[3]</td>
<td>Center Frequency Unit</td>
<td>RES</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RES?</td>
</tr>
<tr>
<td>[4]</td>
<td>Offset</td>
<td>COP</td>
</tr>
<tr>
<td></td>
<td></td>
<td>COP?</td>
</tr>
</tbody>
</table>

Table 8.2.1-1  Operation Frequency setting commands
Chapter 8  Native Commands

OPE <numeric>

Parameter

<table>
<thead>
<tr>
<th>&lt;numeric&gt;</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Variable</td>
</tr>
<tr>
<td>1</td>
<td>PCIe-Gen I (2.5 GHz)</td>
</tr>
<tr>
<td>2</td>
<td>PCIe-Gen II (5 GHz)</td>
</tr>
<tr>
<td>3</td>
<td>Serial-ATA (6 GHz)</td>
</tr>
</tbody>
</table>

Function
Sets the operation of the synthesizer.

Example
To set the operation of the synthesizer to Variable:

```
> OPE 0
```

Compatibility
Incompatible with existing models.

OPE?

Response

<table>
<thead>
<tr>
<th>&lt;numeric&gt;</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Variable</td>
</tr>
<tr>
<td>1</td>
<td>PCIe-Gen I (2.5 GHz)</td>
</tr>
<tr>
<td>2</td>
<td>PCIe-Gen II (5 GHz)</td>
</tr>
<tr>
<td>3</td>
<td>Serial-ATA (6 GHz)</td>
</tr>
</tbody>
</table>

Function
Queries the operating of the synthesizer.

Example

```
> OPE?
< OPE 2
```

Compatibility
Incompatible with existing models.

**Note:**
The OPE and OPE? commands can be executed only when MU181000A-x01 or MU181000B-x01 is installed.
### 8.2 Synthesizer Commands

#### FRQ <numeric>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>(&lt;\text{numeric}&gt; = \langle\text{DECIMAL NUMERIC PROGRAM DATA}\rangle)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>When the operation frequency setting resolution unit is kHz:</td>
</tr>
<tr>
<td></td>
<td>100000 to 12500000 100,000 to 12,500,000 kHz, in 1 kHz steps</td>
</tr>
<tr>
<td></td>
<td>When the operation frequency setting resolution unit is MHz:</td>
</tr>
<tr>
<td></td>
<td>100 to 12500 100 to 12,500 MHz, in 1 MHz steps</td>
</tr>
<tr>
<td>Function</td>
<td>Sets the operation frequency of the synthesizer.</td>
</tr>
<tr>
<td>Example</td>
<td>To set the operation frequency of the synthesizer to 12500000 kHz:</td>
</tr>
<tr>
<td></td>
<td>&gt; FRQ 12500000</td>
</tr>
<tr>
<td></td>
<td>To set the operation frequency of the synthesizer to 100 MHz:</td>
</tr>
<tr>
<td></td>
<td>&gt; FRQ 100</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Partially compatible with the MP1632C and MP1761C.</td>
</tr>
<tr>
<td></td>
<td>Compatible with the MP1775A Pulse Pattern Generator (hereinafter, referred to as &quot;MP1775A&quot;).</td>
</tr>
</tbody>
</table>

#### FRQ? 

<table>
<thead>
<tr>
<th>Response</th>
<th>(&lt;\text{numeric}&gt; = \langle\text{NR1 NUMERIC RESPONSE DATA}\rangle)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>When the operation frequency setting resolution unit is kHz (FIX8):</td>
</tr>
<tr>
<td></td>
<td>100000 to 12500000 100,000 to 12,500,000 kHz</td>
</tr>
<tr>
<td></td>
<td>When the operation frequency setting resolution unit is MHz (FIX5):</td>
</tr>
<tr>
<td></td>
<td>100 to 12500 100 to 12,500 MHz</td>
</tr>
<tr>
<td>Function</td>
<td>Queries the operating frequency of the synthesizer.</td>
</tr>
<tr>
<td>Example</td>
<td>To query the operating frequency of the synthesizer in kHz units:</td>
</tr>
<tr>
<td></td>
<td>&gt; FRQ?</td>
</tr>
<tr>
<td></td>
<td>&lt; FRQ 12500000</td>
</tr>
<tr>
<td></td>
<td>&lt; FRQ 100000</td>
</tr>
<tr>
<td></td>
<td>To query the operating frequency of the synthesizer in MHz units:</td>
</tr>
<tr>
<td></td>
<td>&gt; FRQ?</td>
</tr>
<tr>
<td></td>
<td>&lt; FRQ 100</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Partially compatible with the MP1632C and MP1761C.</td>
</tr>
<tr>
<td></td>
<td>Compatible with the MP1775A.</td>
</tr>
</tbody>
</table>
Chapter 8  Native Commands

RES <type>

Parameter  
<type> = <DECIMAL NUMERIC PROGRAM DATA>
0      kHz
1      MHz

Function  
Sets the resolution for setting the operating frequency of the synthesizer.

Example  
To set the resolution for setting the operating frequency of the synthesizer to kHz:
> RES 0

Compatibility  
Compatible with the MP1761C and MP1775A.

RES?

Response  
<type> = <NR1 NUMERIC RESPONSE DATA> (FIX1)

Function  
Queries the resolution for setting the operating frequency of the synthesizer.

Example  
> RES?
< RES 0

Compatibility  
Compatible with the MP1761C and MP1775A.

COP <numeric>

Parameter  
<numeric> = <DECIMAL NUMERIC PROGRAM DATA>
–1000 to 1000 –1,000 to +1,000 ppm, in 1 ppm steps

Function  
Sets the offset value (ppm) for the frequency currently being output.

Example  
To set the offset value for the frequency currently being output to 0 ppm:
> COP -1000

Compatibility  
Incompatible with existing models.

COP?

Response  
<numeric> = <NR1 NUMERIC RESPONSE DATA> (FIX5)
–1000 to 1000 –1,000 to +1,000 ppm

Function  
Queries the offset value (ppm) for the frequency currently being output.

Example  
> COP?
< COP −10
< COP −1000

Compatibility  
Incompatible with existing models.
8.2 Synthesizer Commands

8.2.2 Reference Clock setting commands

![Reference Clock setting field](image)

Figure 8.2.2-1 Reference Clock setting field

Table 8.2.2-1 Reference Clock setting commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Item</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1]</td>
<td>Source</td>
<td>RFC</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RFC?</td>
</tr>
</tbody>
</table>

**RFC <numeric>**

Parameter  
<numeric> = <DECIMAL NUMERIC PROGRAM DATA>
0  
10-MHz internal reference signal
1  
10-MHz external reference signal

Function  
Selects an internal signal or an external input signal for the 10-MHz reference signal.

Example  
To set 10-MHz reference signal for external input:
>`RFC 1`

Compatibility  
Partially compatible with the MP1632C.

**RFC?**

Response  
<numeric> = <NR1 NUMERIC RESPONSE DATA> (FIX1)

Function  
Queries the setting of the 10-MHz reference signal currently selected.

Example  
>`RFC?`
>`RFC 1`

Compatibility  
Partially compatible with the MP1632C.
8.2.3 Spectrum Spread setting commands

The following commands are enabled only when the Option MU181000A-x01 or MU181000B-x01 is available.

Figure 8.2.3-1  Spectrum Spread setting

Table 8.2.3-1  Spectrum Spread setting commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1]</td>
<td>SCC</td>
<td>RSU</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RSU?</td>
</tr>
<tr>
<td>[2]</td>
<td>Spread Method</td>
<td>RSM</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RSM?</td>
</tr>
</tbody>
</table>

RSU <numeric>

Parameter  
\(<\text{numeric}> = \langle\text{DECIMAL NUMERIC PROGRAM DATA}\rangle\>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>SSC function off</td>
</tr>
<tr>
<td>1</td>
<td>SSC function on (using internal reference signal)</td>
</tr>
<tr>
<td>2</td>
<td>SSC function on (using external 10 MHz reference signal)</td>
</tr>
<tr>
<td>3</td>
<td>SSC function on (using external 100 MHz reference signal)</td>
</tr>
</tbody>
</table>

Function  
Sets whether to use the SSC (Spread Spectrum Clocking) function of the synthesizer and the reference signal to be used.

Example  
To set the SSC function to on using internal reference signal of the synthesizer:

\>$\text{RSU 1}$

Compatibility  
Incompatible with existing models.
### 8.2 Synthesizer Commands

#### RSU?

<table>
<thead>
<tr>
<th>Response</th>
<th>Function</th>
<th>Example</th>
<th>Compatibility</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;numeric&gt; = &lt;NR1 NUMERIC RESPONSE DATA (FIX1)&gt;</td>
<td>Queries setting of SSC (Spread Spectrum Clocking) function of the synthesizer.</td>
<td>&gt; RSU? &lt; RSU 1</td>
<td>Incompatible with existing models.</td>
</tr>
</tbody>
</table>

| 0 | SSC function off |
| 1 | SSC function on (using internal reference signal) |
| 2 | SSC function on (using external 10 MHz reference signal) |
| 3 | SSC function on (using external 100 MHz reference signal) |

#### RSM <numeric>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Function</th>
<th>Example</th>
<th>Compatibility</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;numeric&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
<td>Sets spectrum spreading method for SSC (Spread Spectrum Clocking) using 100 MHz reference signal.</td>
<td>&gt; RSM 0</td>
<td>Incompatible with existing models.</td>
</tr>
</tbody>
</table>

| 0 | Changes frequency within the range from 0 to −0.5% |
| 1 | Changes frequency within the range of ±0.25% |

#### RSM?

<table>
<thead>
<tr>
<th>Response</th>
<th>Function</th>
<th>Example</th>
<th>Compatibility</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;numeric&gt; = &lt;NR1 NUMERIC RESPONSE DATA (FIX1)&gt;</td>
<td>Queries spectrum spreading method for SSC (Spread Spectrum Clocking) using 100 MHz reference signal.</td>
<td>&gt; RSM? &lt; RSM 0</td>
<td>Incompatible with existing models.</td>
</tr>
</tbody>
</table>

| 0 | Changes frequency within the range from 0 to −0.5% |
| 1 | Changes frequency within the range of ±0.25% |
8.2.4 Jitter setting commands

The following commands are enabled only when the Option MU181000A-x01 or MU181000B-x01 is available.

Figure 8.2.4-1 Jitter setting (Modulation Source: Internal)

Figure 8.2.4-2 Jitter setting (Modulation Source: External)
8.2 Synthesizer Commands

### Table 8.2.4-1 Jitter setting commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1]</td>
<td>Jitter ON/OFF</td>
<td>JOU</td>
</tr>
<tr>
<td></td>
<td></td>
<td>JOU?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>JMS?</td>
</tr>
<tr>
<td>[3]</td>
<td>Frequency</td>
<td>JMF</td>
</tr>
<tr>
<td></td>
<td></td>
<td>JMF?</td>
</tr>
<tr>
<td>[4]</td>
<td>Amplitude</td>
<td>JMP</td>
</tr>
<tr>
<td></td>
<td></td>
<td>JMP</td>
</tr>
<tr>
<td>[5]</td>
<td>Frequency Range</td>
<td>JMG</td>
</tr>
<tr>
<td></td>
<td></td>
<td>JMG?</td>
</tr>
<tr>
<td>[6]</td>
<td>Amplitude Range</td>
<td>JMA</td>
</tr>
<tr>
<td></td>
<td></td>
<td>JMA?</td>
</tr>
<tr>
<td>[7]</td>
<td>Trigger Source</td>
<td>JTS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>JTS?</td>
</tr>
</tbody>
</table>

**JOU <numeric>**

Parameter

- `<numeric> = <DECIMAL NUMERIC PROGRAM DATA>`
  - 0: Jitter modulation off
  - 1: Jitter modulation on

Function

Sets on/off of Jitter output.

Example

To set Jitter output to off:

> JOU 0

Compatibility

Incompatible with existing models.

**JOU?**

Response

- `<numeric> = <NR1 NUMERIC RESPONSE DATA> (FIX1)`
  - 0: Jitter modulation off
  - 1: Jitter modulation on

Function

Queries on/off of Jitter output.

Example

> JOU?
< JOU 0

Compatibility

Incompatible with existing models.
### JMS `<numeric>`

<table>
<thead>
<tr>
<th>Parameter</th>
<th><code>&lt;numeric&gt;</code> = <code>&lt;DECIMAL NUMERIC PROGRAM DATA&gt;</code></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Internal modulation source</td>
</tr>
<tr>
<td>1</td>
<td>External modulation source</td>
</tr>
<tr>
<td>2</td>
<td>External modulation source (IQ)</td>
</tr>
</tbody>
</table>

- **Function**: Sets modulation signal source.
- **Example**:
  
  To set modulation signal source to internal modulation source:
  
  ```
  > JMS 0
  ```

- **Compatibility**:
  
  Incompatible with existing models.

### JMS?  

<table>
<thead>
<tr>
<th>Response</th>
<th><code>&lt;numeric&gt;</code> = <code>&lt;NR1 NUMERIC RESPONSE DATA&gt;</code> (FIX1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Internal modulation source</td>
</tr>
<tr>
<td>1</td>
<td>External modulation source</td>
</tr>
<tr>
<td>2</td>
<td>External modulation source (IQ)</td>
</tr>
</tbody>
</table>

- **Function**: Queries modulation signal source.
- **Example**:
  
  ```
  > JMS?
  < JMS 0
  ```

- **Compatibility**:
  
  Incompatible with existing models.
## 8.2 Synthesizer Commands

### JMF <numeric1>,<numeric2>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;numeric1&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
<td>When Hz is selected for &lt;numeric2&gt;</td>
</tr>
<tr>
<td>9.000 to 1000.0</td>
<td>9.000 to 1000.0 Hz</td>
</tr>
<tr>
<td>Setting steps</td>
<td>9.000[Hz] ≤ fmod ≤ 10.000 [Hz]: In 0.001-Hz steps</td>
</tr>
<tr>
<td></td>
<td>10.01[Hz] ≤ fmod ≤ 100.00 [Hz]: In 0.01-Hz steps</td>
</tr>
<tr>
<td></td>
<td>100.1[Hz] ≤ fmod ≤ 1000.0 [Hz]: In 0.1-Hz steps</td>
</tr>
<tr>
<td>When kHz is selected for &lt;numeric2&gt;</td>
<td>1.001 to 2000.0 kHz</td>
</tr>
<tr>
<td>Setting steps</td>
<td>1.001[kHz] ≤ fmod ≤ 10.000 [kHz]: In 0.001-kHz steps</td>
</tr>
<tr>
<td></td>
<td>10.01[kHz] ≤ fmod ≤ 100.00 [kHz]: In 0.01-kHz steps</td>
</tr>
<tr>
<td></td>
<td>100.1[kHz] ≤ fmod ≤ 1000.0 [kHz]: In 0.1-kHz steps</td>
</tr>
<tr>
<td>When MHz is selected for &lt;numeric2&gt;</td>
<td>1.001 to 80.00 MHz</td>
</tr>
<tr>
<td>Setting steps</td>
<td>1.001[MHz] ≤ fmod ≤ 10.000 [MHz]: In 0.001-MHz step</td>
</tr>
<tr>
<td></td>
<td>10.01[MHz] ≤ fmod ≤ 80.00 [MHz]: In 0.01-MHz step</td>
</tr>
<tr>
<td>&lt;numeric2&gt; = &lt;CHARACTER PROGRAM DATA&gt;</td>
<td>Hz units</td>
</tr>
<tr>
<td>0</td>
<td>kHz units</td>
</tr>
<tr>
<td>1</td>
<td>MHz units</td>
</tr>
<tr>
<td>Function</td>
<td>Sets internal modulation frequency and units.</td>
</tr>
<tr>
<td>Example</td>
<td>To set internal modulation frequency to 10 kHz:</td>
</tr>
<tr>
<td></td>
<td>&gt; JMF 10.00,1</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Incompatible with existing models.</td>
</tr>
</tbody>
</table>
Chapter 8  Native Commands

**JMF?**

<table>
<thead>
<tr>
<th>Response</th>
<th>(&lt;\text{numeric1}&gt; = \langle\text{NR2 NUMERIC RESPONSE DATA}\rangle \text{ (FIX6)})</th>
</tr>
</thead>
<tbody>
<tr>
<td>9.000 to 1000.0</td>
<td>9.000 to 1000.0 Hz (When Hz is selected for resolution)</td>
</tr>
<tr>
<td>1.001 to 1000.0</td>
<td>1.00 to 1000.0 kHz (When kHz is selected for resolution)</td>
</tr>
<tr>
<td>1.001 to 80.00</td>
<td>1.00 to 80 MHz (When MHz is selected for resolution)</td>
</tr>
</tbody>
</table>

\(<\text{numeric2}> = \langle\text{NR1 NUMERIC RESPONSE DATA}\rangle \text{ (FIX1)}\)

<table>
<thead>
<tr>
<th></th>
<th>Hz units</th>
<th>kHz units</th>
<th>MHz units</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Function**
Queries internal modulation frequency and units.

**Example**

\(\text{> JMF?}\)
\(<\text{JMF 10.000,1}\)

**Compatibility**
Incompatible with existing models.

---

**JMP <numeric1>,<numeric2>**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>(&lt;\text{numeric1}&gt; = \langle\text{DECIMAL NUMERIC PROGRAM DATA}\rangle)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 to 999</td>
<td>0 to 999 mUIp-p, in 1-mUIp-p steps</td>
</tr>
<tr>
<td>1.00 to 32.00</td>
<td>1.00 to 32.00 UIp-p, in 0.01-UIp-p steps</td>
</tr>
<tr>
<td>32.1 to 256.0</td>
<td>32.1 to 256.0 UIp-p, in 0.1-UIp-p steps</td>
</tr>
<tr>
<td>257 to 2049</td>
<td>257 to 2049 UIp-p, in 1-UIp-p steps</td>
</tr>
<tr>
<td>2050 to 4000</td>
<td>2050 to 4000 UIp-p, in 10-UIp-p steps</td>
</tr>
<tr>
<td></td>
<td>(&lt;\text{numeric2}&gt; = \langle\text{DECIMAL NUMERIC PROGRAM DATA}\rangle)</td>
</tr>
<tr>
<td>0</td>
<td>mUIp-p units</td>
</tr>
<tr>
<td>1</td>
<td>UIp-p units</td>
</tr>
</tbody>
</table>

**Function**
Sets internal modulation amplitude and units.

**Example**
To set internal modulation amplitude to 1.100 UIp-p:
\(\text{> JMP 1.100,1}\)

**Compatibility**
Incompatible with existing models.
### 8.2 Synthesizer Commands

#### JMP?

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;numeric1&gt; = &lt;NR2 NUMERIC RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 to 999</td>
<td>0 to 999 mUIp·p</td>
</tr>
<tr>
<td>1.00 to 32.00</td>
<td>1.00 to 32.00 UIp·p</td>
</tr>
<tr>
<td>32.1 to 256.0</td>
<td>32.1 to 256.0 UIp·p</td>
</tr>
<tr>
<td>257 to 2049</td>
<td>257 to 2049 UIp·p</td>
</tr>
<tr>
<td>2050 to 4000</td>
<td>2050 to 4000 UIp·p</td>
</tr>
</tbody>
</table>

<numeric2> = <DECIMAL NUMERIC PROGRAM DATA> (FIX1)

<table>
<thead>
<tr>
<th>Function</th>
<th>Queries internal modulation amplitude and units.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example</td>
<td>&gt; JMP?</td>
</tr>
<tr>
<td></td>
<td>&lt; JMP 1.100,1</td>
</tr>
</tbody>
</table>

Compatibility: Incompatible with existing models.
### JMG <numeric>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;numeric&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>275 Hz</td>
</tr>
<tr>
<td>1</td>
<td>550 Hz</td>
</tr>
<tr>
<td>2</td>
<td>1 kHz</td>
</tr>
<tr>
<td>3</td>
<td>1.1 kHz</td>
</tr>
<tr>
<td>4</td>
<td>2.2 kHz</td>
</tr>
<tr>
<td>5</td>
<td>2.75 kHz</td>
</tr>
<tr>
<td>6</td>
<td>4.4 kHz</td>
</tr>
<tr>
<td>7</td>
<td>5.5 kHz</td>
</tr>
<tr>
<td>8</td>
<td>11 kHz</td>
</tr>
<tr>
<td>9</td>
<td>22 kHz</td>
</tr>
<tr>
<td>10</td>
<td>27.5 kHz</td>
</tr>
<tr>
<td>11</td>
<td>44 kHz</td>
</tr>
<tr>
<td>12</td>
<td>55 kHz</td>
</tr>
<tr>
<td>13</td>
<td>100 kHz</td>
</tr>
<tr>
<td>14</td>
<td>110 kHz</td>
</tr>
<tr>
<td>15</td>
<td>220 kHz</td>
</tr>
<tr>
<td>16</td>
<td>250 kHz</td>
</tr>
<tr>
<td>17</td>
<td>440 kHz</td>
</tr>
<tr>
<td>18</td>
<td>500 kHz</td>
</tr>
<tr>
<td>19</td>
<td>1 MHz</td>
</tr>
<tr>
<td>20</td>
<td>2 MHz</td>
</tr>
<tr>
<td>21</td>
<td>4 MHz</td>
</tr>
<tr>
<td>22</td>
<td>80 MHz</td>
</tr>
<tr>
<td>23</td>
<td>500 MHz</td>
</tr>
<tr>
<td>24</td>
<td>1 GHz</td>
</tr>
<tr>
<td>27</td>
<td>All range</td>
</tr>
</tbody>
</table>

**Function**: Sets external modulation frequency range.

**Example**: To set external modulation frequency range to 275 kHz:

```
> JMG 0
```

**Compatibility**: Incompatible with existing models.

### JMG?

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;numeric&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt; (FIX2)</th>
</tr>
</thead>
</table>

**Function**: Queries external modulation frequency range.

**Example**:

```
> JMG?
< JMG 0
```

**Compatibility**: Incompatible with existing models.
### JMA<numeric>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Definition</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;numeric&gt;</td>
<td>= &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0.22 UI</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>2.00 UI</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>20 UI</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>200 UI</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>4,000 UI</td>
<td></td>
</tr>
</tbody>
</table>

**Function**: Sets external modulation amplitude.

**Example**: To set external modulation amplitude to 0.22 UI:
```plaintext
> JMA 0
```

**Compatibility**: Incompatible with existing models.

### JMA?

<table>
<thead>
<tr>
<th>Response</th>
<th>Definition</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;numeric&gt;</td>
<td>= &lt;NR1 NUMERIC RESPONSE DATA&gt; (FIX1)</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0.22 UI</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>2.00 UI</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>20 UI</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>200 UI</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>4,000 UI</td>
<td></td>
</tr>
</tbody>
</table>

**Function**: Queries external modulation amplitude.

**Example**: 
```plaintext
> JMA?
< JMA 0
```

**Compatibility**: Incompatible with existing models.

### JTS<numeric>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Definition</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;numeric&gt;</td>
<td>= &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>f/1</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>f/64</td>
<td></td>
</tr>
</tbody>
</table>

**Function**: Selects trigger output division ratio.

**Example**: To set trigger output division ratio to f/1:
```plaintext
> JTR 0
```

**Compatibility**: Incompatible with existing models.
## JTS?

<table>
<thead>
<tr>
<th>Response</th>
<th>[&lt;\text{numeric}&gt; = &lt;\text{NR1 NUMERIC RESPONSE DATA}&gt;(\text{FIX1})]</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>f/1</td>
</tr>
<tr>
<td>1</td>
<td>f/64</td>
</tr>
</tbody>
</table>

**Function**
Queries trigger output division ratio.

**Example**

```plaintext
> JTR?
< JTR 0
```

**Compatibility**
Incompatible with existing models.
8.3 PPG Commands

This section describes the setting/query commands for the MU181020A 12.5 Gbit/s PPG and the MU181020B 14 Gbit/s PPG. The commands related to each tab are described in separate sections below. Before executing a setting/query command for the MU181020A/B, specify the slot number of the module to be remotely controlled, by using the :MODule:ID command. Refer to Section 7.1 “Common Commands” and 9.1 “Basic Operation of Device Messages” for how to specify a slot number with the :MODule:ID command.

However, unless there is a special item, see the MU181020A.

8.3.1 Commands related to Output tab

![Output tab diagram](image)

Figure 8.3.1-1 Output tab
8.3.1.1  Output setting commands

![Output setting field](image)

Figure 8.3.1.1-1  Output setting field

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1]</td>
<td>Data/XData</td>
<td>DON</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DON?</td>
</tr>
<tr>
<td>[2]</td>
<td>Clock/XClock</td>
<td>CON</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CON?</td>
</tr>
<tr>
<td>[3]</td>
<td>Offset</td>
<td>OFS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>OFS?</td>
</tr>
<tr>
<td>[4]</td>
<td>No label</td>
<td>OOF</td>
</tr>
<tr>
<td></td>
<td>(Setting item selection)</td>
<td>OOF?</td>
</tr>
</tbody>
</table>

**DON <numeric>**

Parameter  
\(<\text{numeric}> = \text{<DECIMAL NUMERIC PROGRAM DATA>}\>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Output OFF</td>
</tr>
<tr>
<td>1</td>
<td>Output ON</td>
</tr>
</tbody>
</table>

Function  
Sets data output ON or OFF.

Example  
To set data output ON:

```
> DON 1
```

Compatibility  
Compatible with the MP1632C.

**DON?**

Response  
\(<\text{numeric}> = \text{<NR1 NUMERIC RESPONSE DATA> (FIX1)}\>

Function  
Queries the data output ON or OFF state.

Example  
To set data output ON:

```
> DON?
< DON 1
```

Compatibility  
Compatible with the MP1632C.
CON <numeric>

Parameter

<table>
<thead>
<tr>
<th>&lt;numeric&gt;</th>
<th>&lt;DECIMAL NUMERIC PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Output OFF</td>
</tr>
<tr>
<td>1</td>
<td>Output ON</td>
</tr>
</tbody>
</table>

Function
Sets clock output ON or OFF.

Example
To set clock output ON:
> CON 1

Compatibility
Compatible with the MP1632C.

CON?

Response

<numeric> = <NR1 NUMERIC RESPONSE DATA> (FIX1)

Function
Queries the clock output ON or OFF state.

Example
> CON?
< CON 1

Compatibility
Compatible with the MP1632C.

OFS <offset>

Parameter

<table>
<thead>
<tr>
<th>&lt;offset&gt;</th>
<th>&lt;DECIMAL NUMERIC PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>VOH High level</td>
</tr>
<tr>
<td>1</td>
<td>VTH Center value between High and Low levels</td>
</tr>
<tr>
<td>2</td>
<td>VOL Low level</td>
</tr>
</tbody>
</table>

Function
Sets the offset reference value for data and clock output.

Example
To set the offset reference value to Voh:
> OFS 0

Compatibility
Compatible with the MP1632C, MP1761C and MP1775A.

OFS?

Response

<offset> = <NR1 NUMERIC RESPONSE DATA> (FIX1)

Function
Queries the offset reference value for data and clock outputs.

Example
> OFS?
< OFS 1

Compatibility
Compatible with the MP1632C, MP1761C and MP1775A.
**Chapter 8  Native Commands**

**OOF <numeric>**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;numeric&gt;</td>
<td>= &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td></td>
<td>Data/XData setting screen</td>
</tr>
<tr>
<td>1</td>
<td></td>
<td>Clock/XClock setting screen</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>CMU Bitrate setting screen</td>
</tr>
</tbody>
</table>

**Function**
Selects the data output, clock output, or CMU setting screen.

**Example**
To select the data output setting screen:
> OOF 0

**Compatibility**
Incompatible with existing models.

**OOF?**

<table>
<thead>
<tr>
<th>Response</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;numeric&gt;</td>
<td>= &lt;NR1 NUMERIC RESPONSE DATA&gt; (FIX1)</td>
<td></td>
</tr>
</tbody>
</table>

**Function**
Queries the selected setting screen.

**Example**
> OOF?
< OOF 0

**Compatibility**
Incompatible with existing models.
8.3.1.2 Data/XData setting commands

Figure 8.3.1.2-1  Data/XData setting field

Figure 8.3.1.2-2  Level Guard Setting dialog box
### Table 8.3.1.2-1  Data/XData setting commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1]</td>
<td>Tracking</td>
<td>TRK</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TRK?</td>
</tr>
<tr>
<td>[2]</td>
<td>Level Guard</td>
<td>DGD</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DGD?</td>
</tr>
<tr>
<td>[3]</td>
<td>Defined Interface (Data)</td>
<td>DAL</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DAL?</td>
</tr>
<tr>
<td>[4]</td>
<td>Defined Interface (XData)</td>
<td>XDL</td>
</tr>
<tr>
<td></td>
<td></td>
<td>XDL?</td>
</tr>
<tr>
<td>[5]</td>
<td>Amplitude (Data)</td>
<td>DAP</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DAP?</td>
</tr>
<tr>
<td>[6]</td>
<td>Amplitude (XData)</td>
<td>NAP</td>
</tr>
<tr>
<td></td>
<td></td>
<td>NAP?</td>
</tr>
<tr>
<td>[7]</td>
<td>AC ON/OFF</td>
<td>DAD</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DAD?</td>
</tr>
<tr>
<td>[8]</td>
<td>Offset (Data)</td>
<td>DOS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DOS?</td>
</tr>
<tr>
<td>[9]</td>
<td>Offset (XData)</td>
<td>NOS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>NOS?</td>
</tr>
<tr>
<td>[10]</td>
<td>External ATT Factor (Data)</td>
<td>DAT</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DAT?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>XAT?</td>
</tr>
<tr>
<td>[12]</td>
<td>Cross Point (Data)</td>
<td>DCR</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DCR?</td>
</tr>
<tr>
<td>[13]</td>
<td>Cross Point (XData)</td>
<td>XDC</td>
</tr>
<tr>
<td></td>
<td></td>
<td>XDC?</td>
</tr>
<tr>
<td>[14]</td>
<td>Phase Adjust</td>
<td>PAU</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PAU?</td>
</tr>
<tr>
<td>[15]</td>
<td>Phase Adjust</td>
<td>PAD</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PAD?</td>
</tr>
<tr>
<td>[16]</td>
<td>Phase Adjust</td>
<td>CDL</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CDL?</td>
</tr>
<tr>
<td>[17]</td>
<td>Relative 0</td>
<td>PRO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PRO?</td>
</tr>
<tr>
<td>[18]</td>
<td>No label</td>
<td>PRE</td>
</tr>
<tr>
<td></td>
<td>(Relative value setting)</td>
<td>PRE?</td>
</tr>
<tr>
<td>[19]</td>
<td>Calibration</td>
<td>PCA</td>
</tr>
</tbody>
</table>
Table 8.3.1.2-1  Data/XData setting commands (Cont’d)

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>[20]</td>
<td>Amplitude max</td>
<td>DLM</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DLM?</td>
</tr>
<tr>
<td>[21]</td>
<td>Offset max/min</td>
<td>DOF</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DOF?</td>
</tr>
<tr>
<td>[22]</td>
<td>No label (Delay status query)</td>
<td>DLY?</td>
</tr>
<tr>
<td>[23]</td>
<td>No label (Query for data/clock output setting completion status)</td>
<td>OCS?</td>
</tr>
<tr>
<td>[24]</td>
<td>Jitter Input</td>
<td>PJO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PJO?</td>
</tr>
</tbody>
</table>

**TRK <numeric>**

**Parameter**  
<numeric> = <DECIMAL NUMERIC PROGRAM DATA>  
0  OFF  
1  ON  

**Function**  
Sets the common setting for the Data and XData output ON or OFF.  

**Example**  
To set the common setting ON:  
> TRK 1  

**Compatibility**  
Compatible with the MP1632C and MP1761C.

**TRK?**

**Response**  
<numeric> = <NR1 NUMERIC RESPONSE DATA> (FIX1)

**Function**  
Queries the common setting for the Data and XData outputs ON or OFF state.

**Example**  
> TRK?  
< TRK 1  

**Compatibility**  
Compatible with the MP1632C and MP1761C.
### DGD <numeric>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;numeric&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>OFF</td>
</tr>
<tr>
<td>1</td>
<td>ON</td>
</tr>
</tbody>
</table>

**Function**
Sets the data output range limitation ON or OFF.

**Example**
To set the data output range limitation ON:
> DGD 1

**Compatibility**
Incompatible with existing models.

### DGD?

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;numeric&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt; (FIX1)</th>
</tr>
</thead>
</table>

**Function**
Queries the data output range limitation ON or OFF state.

**Example**
> DGD?
< DGD 1

**Compatibility**
Incompatible with existing models.

### DAL(XDL) <level>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;level&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Variable</td>
</tr>
<tr>
<td>1</td>
<td>NECL</td>
</tr>
<tr>
<td>2</td>
<td>PCML</td>
</tr>
<tr>
<td>3</td>
<td>NCML</td>
</tr>
<tr>
<td>4</td>
<td>SCFL</td>
</tr>
<tr>
<td>5</td>
<td>LVPECL</td>
</tr>
<tr>
<td>6</td>
<td>LVDS200 mV</td>
</tr>
<tr>
<td>7</td>
<td>LVDS400 mV</td>
</tr>
</tbody>
</table>

**Function**
Sets the level of the fixed interface for the Data (XData) output.

**Example**
To set the level of the XData output to the NECL level:
> XDL 1

**Compatibility**
Compatible with the MP1632C.

### DAL(XDL)?

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;level&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt; (FIX1)</th>
</tr>
</thead>
</table>

**Function**
Queries the level of the fixed interface for the Data (XData) output.

**Example**
To query the level of the XData output:
> XDL?
< XDL 1

**Compatibility**
Compatible with the MP1632C.
### DAP(NAP) <numeric>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;numeric&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.050 to 0.800</td>
<td>0.050 to 0.800 V, in 0.002-V steps (MU181020A-x10)</td>
</tr>
<tr>
<td>0.250 to 2.500</td>
<td>0.250 to 2.500 V, in 0.002-V steps (MU181020A-x11, MU181020B-x11)</td>
</tr>
<tr>
<td>0.050 to 2.000</td>
<td>0.050 to 2.000 V, in 0.002-V steps (MU181020A-x12, MU181020B-x12)</td>
</tr>
<tr>
<td>0.500 to 3.500</td>
<td>0.500 to 3.500 V, in 0.002 V steps (MU181020A-x13, MU181020B-x13)</td>
</tr>
</tbody>
</table>

**Function**: Sets the amplitude of the Data (XData) output.

**Example**: To set the amplitude of the Data output to 1.000 V:

```bash
> DAP 1.000
```

**Compatibility**: Compatible with the MP1632C, MP1761C and MP1775A (0.50 to 2.00 V).

---

### DAP(NAP)?

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;numeric&gt; = &lt;NR2 NUMERIC RESPONSE DATA&gt; (FIX5)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function</td>
<td>Queries the amplitude of the Data (XData) output.</td>
</tr>
<tr>
<td>Example</td>
<td>To query the amplitude of the Data output:</td>
</tr>
<tr>
<td></td>
<td>&gt; DAP?</td>
</tr>
<tr>
<td></td>
<td>&lt; DAP 1.000</td>
</tr>
</tbody>
</table>

**Compatibility**: Compatible with the MP1632C, MP1761C and MP1775A.

---

### DAD <numeric>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;numeric&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Offset OFF (DC output)</td>
</tr>
<tr>
<td>1</td>
<td>Offset ON (AC output)</td>
</tr>
</tbody>
</table>

**Function**: Sets the data output offset ON or OFF

**Example**: To set the data output offset ON:

```bash
> DAD 1
```

**Compatibility**: Incompatible with existing models.

---

### DAD?

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;numeric&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt; (FIX1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function</td>
<td>Queries the data output offset ON or OFF state.</td>
</tr>
<tr>
<td>Example</td>
<td>&gt; DAD?</td>
</tr>
<tr>
<td></td>
<td>&lt; DAD 1</td>
</tr>
</tbody>
</table>

**Compatibility**: Incompatible with existing models.
Chapter 8  Native Commands

DOS(NOS) <numeric>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;numeric&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>−2.000 to 3.300</td>
<td>−2.000 to 3.300 Voh, in 0.001-V steps (MU181020A-x10, x11, x12, x13, MU181020B-x11, x12, x13, x14)</td>
</tr>
<tr>
<td>−2.500 to 3.225</td>
<td>−2.500 to 3.225 Vth, in 0.001-V steps (MU181020A-x10)</td>
</tr>
<tr>
<td>−3.250 to 3.175</td>
<td>−3.250 to 3.175 Vth, in 0.001-V steps (MU181020A-x11, MU181020B-x11)</td>
</tr>
<tr>
<td>−3.000 to 3.275</td>
<td>−3.000 to 3.275 Vth, in 0.001-V steps (MU181020A-x12, MU181020B-x12)</td>
</tr>
<tr>
<td>−3.000 to 3.050</td>
<td>−3.000 to 3.050 Vth, in 0.001-V steps (MU181020A-x13, MU181020B-x13)</td>
</tr>
<tr>
<td>−3.000 to 3.250</td>
<td>−3.000 to 3.250 Vol, in 0.001-V steps (MU181020A-x10)</td>
</tr>
<tr>
<td>−4.500 to 3.050</td>
<td>−4.500 to 3.050 Vol, in 0.001-V steps (MU181020A-x11, MU181020B-x11)</td>
</tr>
<tr>
<td>−4.000 to 3.250</td>
<td>−4.000 to 3.250 Vol, in 0.001-V steps (MU181020A-x12, MU181020B-x12)</td>
</tr>
<tr>
<td>−4.000 to 2.800</td>
<td>−4.000 to 2.800 Vol, in 0.001-V steps (MU181020A-x13, MU181020B-x13)</td>
</tr>
</tbody>
</table>

Function
Sets the Data (XData) output offset.

Example
To set the XData output offset to 1.000 Voh:
> NOS 1.000

Compatibility
Compatible with the MP1632C, MP1761C and MP1775A.

DOS(NOS)?

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;numeric&gt; = &lt;NR2 NUMERIC RESPONSE DATA&gt; (FIX6)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function</td>
<td>Queries the Data (XData) output offset.</td>
</tr>
<tr>
<td>Example</td>
<td>To query the XData output offset:</td>
</tr>
<tr>
<td></td>
<td>&gt; NOS?</td>
</tr>
<tr>
<td></td>
<td>&lt; NOS 1.000</td>
</tr>
<tr>
<td></td>
<td>&lt; NOS -1.000</td>
</tr>
</tbody>
</table>

Compatibility
Compatible with the MP1632C, MP1761C and MP1775A.
**DAT(XAT) <numeric>**

Parameter

\[ \text{<numeric>} = \text{<DECIMAL NUMERIC PROGRAM DATA>} \]

0 to 40

0 to 40 dB, in 1-dB steps

Function

Sets the External ATT Factor of the Data (XData) output.

Example

To set the External ATT Factor of the Data output to 20 dB:

\[ \text{> DAT 20} \]

Compatibility

Incompatible with existing models.

---

**DAT(XAT)?**

Response

\[ \text{<numeric>} = \text{<NR1 NUMERIC RESPONSE DATA>} \text{ (FIX2)} \]

Function

Queries the External ATT Factor of the Data (XData) output.

Example

To query the External ATT Factor of Data output:

\[ \text{> DAT?} \]

\[ \text{< DAT 20} \]

\[ \text{< DAT 0} \]

Compatibility

Incompatible with existing models.

---

**DCR(XDC) <numeric>**

Parameter

\[ \text{<numeric>} = \text{<DECIMAL NUMERIC PROGRAM DATA>} \]

30 to 70

30 to 70%, in 1% steps (MU181020A-x10, -x11, MU181020B-x11)

20 to 80

20 to 80%, in 1% steps (MU181020A-x12, MU181020B-x12)

20.0 to 90.0

20.0 to 90.0%, in 0.1% steps (MU181020A-x13, MU181020B-x13)

Function

Sets the Data (XData) output crosspoint.

Example

To set the crosspoint of the XData output to 60%:

\[ \text{> XDC 60} \]

Compatibility

Partially compatible with the MP1632C parameters (25 to 75%, in 1% steps).

---

**DCR(XDC)?**

Response

\[ \text{<numeric>} = \text{<NR1 NUMERIC RESPONSE DATA>} \text{ (FIX2)} \]

\[ \text{<numeric>} = \text{<NR2 NUMERIC RESPONSE DATA>} \text{ (FIX4)} \]

Function

Queries the Data (XData) output crosspoint.

Example

To query the crosspoint of the XData output:

\[ \text{> XDC?} \]

\[ \text{< XDC 60} \]

Compatibility

Partially compatible with the MP1632C parameters (25 to 75%, in 1% steps).
**Chapter 8  Native Commands**

**PAU <numeric>**

Parameter

<numeric> = <DECIMAL NUMERIC PROGRAM DATA>

-1000 to 1000  
-10,000 mUI, in 1-mUI steps

In the case of Combination

-64000 to 64000  
64,000 mUI, in 1-mUI steps

Function

Sets the data output delay in mUI units.

Example

To set the data output delay to 500 mUI:

> PAU 500

Compatibility

Incompatible with existing models.

**PAU?**

Response

<numeric> = <NR1 NUMERIC RESPONSE DATA> (FIX6)

Function

Queries the data output delay in mUI units.

Example

> PAU?

< PAU   -500
< PAU  32000
< PAU -32000

Compatibility

Incompatible with existing models.

**PAD <numeric>**

Parameter

<numeric> = <DECIMAL NUMERIC PROGRAM DATA>

-10000 to 10000  
-10,000 to 10,000 ps, in 10-ps steps (at 0.1 GHz)

-1000.0 to 1000.0  
-1,000 to 1,000 ps, in 1-ps steps (at 1 GHz)

-80.00 to 80.00  
-80 to 80 ps, in 0.08-ps steps (at 12.5 GHz)

In the case of Combination

-640000 to 640000  
640,000 ps, in 10-ps steps (at 0.1 GHz)

-64000.0 to 64000.0  
64,000 ps, in 1-ps steps (at 1 GHz)

-5120.00 to 5120.00  
5,120 ps, in 0.08-ps steps (at 12.5 GHz)

Converted from the calculated frequency value, based on the setting resolution of mUI.

Function

Sets the data output delay in ps units.

The specified value may not be exactly set as is, due to the limitation of the setting resolution. In this event, the value is adjusted to the most approximate value and set.

Example

To set the data output delay to 100 ps:

> PAD 100

Compatibility

Incompatible with existing models.
8.3 PPG Commands

### PAD?

**Response**

\[ \text{<numeric> = <NR1 NUMERIC RESPONSE DATA> \<NR2 NUMERIC RESPONSE DATA> (FIX8)} \]

**Function**

 Queries the data output delay in ps units.

**Example**

\[
\text{> PAD?} \\
\text{< PAD \ -10000 (at 0.1 GHz)} \\
\text{< PAD \ -64000.0 (at 1 GHz)}
\]

**Compatibility**

Incompatible with existing models.

### CDL <numeric>

**Parameter**

\[ \text{<numeric> = <DECIMAL NUMERIC PROGRAM DATA> \-10000 to 10000 \-10,000 to 10,000 ps, in 10-ps steps (at 0.1 GHz)} \\
\text{\-1000.0 to 1000.0 \-1,000 to 1,000 ps, in 1-ps steps (at 1 GHz)} \\
\text{\-80.00 to 80.00 \-80 to 80 ps, in 0.08-ps steps (at 12.5 GHz)} \\
\text{In the case of Combination} \\
\text{\-640000 to 640000 \-640,000 to 640,000 ps, in 10-ps steps (at 0.1 GHz)} \\
\text{\-64000.0 to 64000.0 \-64,000 to 64,000 ps, in 1-ps steps (at 1 GHz)} \\
\text{\-5120.00 to 5120.00 \-5,120 to 5,120 ps, in 0.08-ps steps (at 12.5 GHz)} \]

Converted from the calculated frequency value, based on the setting resolution of mUI.

**Function**

Sets the data output delay in ps units.

The specified value may not be exactly set as is, due to the limitation of the setting resolution. In this event, the value is adjusted to the most approximate value and set.

**Example**

To set the data output delay to 100 ps:

\[
\text{> CDL 100}
\]

**Compatibility**

Compatible with the MP1632C, MP1761C and MP1775A.

### CDL?

**Response**

\[ \text{<numeric> = <NR1 NUMERIC RESPONSE DATA> \<NR2 NUMERIC RESPONSE DATA> (FIX8)} \]

Converted from the calculated frequency value, based on the setting resolution of mUI.

**Function**

Queries the data output delay in ps units.

**Example**

\[
\text{> CDL?} \\
\text{< CDL \ -10000 (at 0.1 GHz)} \\
\text{< CDL \ -5120.00 (at 12.5 GHz)}
\]

**Compatibility**

Compatible with the MP1632C, MP1761C and MP1775A.
Chapter 8  Native Commands

PRO <numeric>

Parameter  
<numeric> = <DECIMAL NUMERIC PROGRAM DATA>

0       Reference OFF
1       Reference ON

Function  
Sets the data output delay reference ON or OFF.

Example  
To set the data output delay reference ON:

> PRO 1

Compatibility  
Incompatible with existing models.

PRO?

Response  
<numeric> = <NR1 NUMERIC RESPONSE DATA> (FIX1)

Function  
Queries the data output delay reference setting.

Example  
> PRO?
< PRO 1

Compatibility  
Incompatible with existing models.
### PRE <numeric>[,<unit>]

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;numeric&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(mUI units)</td>
<td>–2000 to 2000</td>
<td>–2,000 to 2,000 mUI, in 1-mUI steps</td>
</tr>
<tr>
<td>(ps units)</td>
<td>Converted from the calculated frequency value, based on the setting resolution of mUI.</td>
<td></td>
</tr>
<tr>
<td>–20000 to 20000</td>
<td>–20,000 to 20,000 ps, in 10-ps steps</td>
<td>(at 0.1 GHz)</td>
</tr>
<tr>
<td>–2000.0 to 2000.0</td>
<td>–2,000 to 2,000 ps, in 1-ps steps</td>
<td>(at 1 GHz)</td>
</tr>
<tr>
<td>–160.00 to 160.00</td>
<td>–160 to 160 ps, in 0.08-ps steps</td>
<td>(at 12.5 GHz)</td>
</tr>
<tr>
<td>In the case of Combination</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(mUI units)</td>
<td>–64000 to 64000</td>
<td>–64,000 to 64,000 mUI, in 1-mUI steps</td>
</tr>
<tr>
<td>(ps units)</td>
<td>Converted from the calculated frequency value, based on the setting resolution of mUI.</td>
<td></td>
</tr>
<tr>
<td>–640000 to 640000</td>
<td>–640,000 to 640,000 ps, in 10-ps steps</td>
<td>(at 0.1 GHz)</td>
</tr>
<tr>
<td>–64000.0 to 64000.0</td>
<td>–64,000 to 64,000 ps, in 1-ps steps</td>
<td>(at 1 GHz)</td>
</tr>
<tr>
<td>–5120.00 to 5120.00</td>
<td>–5120 to 5120 ps, in 0.08-ps steps</td>
<td>(at 12.5 GHz)</td>
</tr>
<tr>
<td>[&lt;unit&gt;] = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
<td>mUI units</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>ps units</td>
<td>(The mUI unit is selected when &lt;unit&gt; is omitted.)</td>
</tr>
</tbody>
</table>

**Function**

Sets the value and unit of the data output delay reference.
Set a value using a difference from the reference value.
The specified value may not be exactly set as is when the ps unit is set, due to the limitation of the setting resolution. The value is adjusted to the most approximate value and set in this event.

**Example**

To set the value for the data output delay reference to –1000 mUI:

```
> PRE -1000,0
```

**Compatibility**

Incompatible with existing models.
Chapter 8  Native Commands

PRE? [<unit>]
Parameter   [<unit>] = <DECIMAL NUMERIC PROGRAM DATA>
  0  mUI units
  1  ps units
   (The ps unit is selected when <unit> is omitted.)
Response    <numeric> = <NR1 NUMERIC RESPONSE DATA>
             <NR2 NUMERIC RESPONSE DATA> (FIX8)
Function    Queries the value and units for the data output delay reference.
Example     > PRE? 0
            < PRE    -1000
            < PRE     0
Compatibility Incompatible with existing models.

PCA
Function    Executes calibration for Data and XData output phases.
Example     > PCA
Compatibility Incompatible with existing models.

DLM <numeric>
Parameter   <numeric> = <DECIMAL NUMERIC PROGRAM DATA>
  0.050 to 0.800  0.050 to 0.800 V, in 0.002-V steps
                  (MU181020A-x10)
  0.250 to 2.500  0.250 to 2.500 V, in 0.002-V steps
                  (MU181020A-x11, MU181020B-x11)
  0.050 to 2.000  0.050 to 2.000 V, in 0.002-V steps
                  (MU181020A-x12, MU181020B-x12)
  0.500 to 3.500  0.500 to 3.500 V, in 0.002-V steps
                  (MU181020A-x13, MU181020B-x13)
Function    Sets the limitation value for the data output amplitude.
Example     To set the limitation value for the data output amplitude to 2.000 V:
            > DLM 2.000
Compatibility Incompatible with existing models.

DLM?
Response    <numeric> = <NR2 NUMERIC RESPONSE DATA> (FIX5)
Function    Queries the limitation value for the data output amplitude.
Example     > DLM?
            < DLM 2.000
Compatibility Incompatible with existing models.
**8.3 PPG Commands**

**DOF <max>,<min>**

Parameter

<max> = <DECIMAL NUMERIC PROGRAM DATA>

-2.000 to 3.300

–2.000 to 3.300 Voh, in 0.001-Voh steps

(MU181020A-x10, x11, x12, x13,
MU181020B-x11, x12, x13)

<min> = <DECIMAL NUMERIC PROGRAM DATA>

-2.800 to 3.250

–2.800 to 3.250 Vol, in 0.001-Vol steps

(MU181020A-x10)

-4.500 to 3.050

–4.500 to 3.050 Vol, in 0.001-Vol steps

(MU181020A-x11, MU181020B-x11)

-4.000 to 3.250

–4.000 to 3.250 Vol, in 0.001-Vol steps

(MU181020A-x12, MU181020B-x12)

-4.000 to 2.800

–4.000 to 2.800 Vol, in 0.001-Vol steps

(MU181020A-x13, MU181020B-x13)

Function

Sets the limitation value for the data output offset.

Example

To set the limitation value for the data output offset to 3.000 Voh max.,

-2.000 Vol min.:

> DOF 3.000,-2.000

Compatibility

Incompatible with existing models.

**DOF?**

Response

<max> = <NR2 NUMERIC RESPONSE DATA> (FIX6)

<min> = <NR2 NUMERIC RESPONSE DATA> (FIX6)

Function

Queries the limitation value for the data output offset.

Example

> DOF?

< DOF 3.000,1.000

< DOF -1.000,-2.000

Compatibility

Incompatible with existing models.

**DLY?**

Response

<numerie> = <NR1 NUMERIC RESPONSE DATA> (FIX1)

0 Ready

1 Delay Busy

Function

Queries the PPG Delay setting state.

Example

> DLY?

< DLY 1

Compatibility

Compatible with the MP1632C, MP1761C and MP1775A.
## OCS?

**Response**

<state> = <NR1 NUMERIC RESPONSE DATA> (FIX 1)

- 0: Amplitude and Offset settings of Data/XData and Clock/XClock have been changed completely.
- 1: Amplitude or Offset setting of Data/XData or Clock/XClock is being changed.

**Function**

Queries the completion status of changing of data output and clock output settings.

**Example**

```plaintext
> OCS?
< OCS 1
```

**Compatibility**

Incompatible with existing models.

---

## PJO <numeric>

**Parameter**

<numeric> = <DECIMAL NUMERIC PROGRAM DATA>

- 0: Jitter input OFF setting
- 1: Jitter input ON setting

**Function**

When a jitter is added to the input signal, the Feedback process for Delay must be set to OFF.

When a jitter is added, set the Jitter Input button to ON, using this command. When a jitter is not added, set the Jitter Input button to OFF.

**Example**

To set the Jitter Input button to ON when a jitter is added:

```plaintext
> PJO 1
```

**Compatibility**

Incompatible with existing models.

---

## PJO?

**Response**

<numeric> = <NR1 NUMERIC RESPONSE DATA> (FIX 1)

- 0: Jitter input OFF setting
- 1: Jitter input ON setting

**Function**

Queries the Jitter Input button setting.

**Example**

```plaintext
> PJO?
< PJO 1
```

**Compatibility**

Incompatible with existing models.
8.3.1.3 Clock/XClock setting commands

**Figure 8.3.1.3-1 Clock/XClock setting field**

**Figure 8.3.1.3-2 Level Guard Setting dialog box**
### Table 8.3.1.3-1 Clock/XClock setting commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1]</td>
<td>Tracking</td>
<td>CGR</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CGR?</td>
</tr>
<tr>
<td>[2]</td>
<td>Level Guard</td>
<td>CLG</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CLG?</td>
</tr>
<tr>
<td>[3]</td>
<td>Defined Interface (Clock)</td>
<td>CLV</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CLV?</td>
</tr>
<tr>
<td>[4]</td>
<td>Defined Interface (XClock)</td>
<td>XLV</td>
</tr>
<tr>
<td></td>
<td></td>
<td>XLV?</td>
</tr>
<tr>
<td>[5]</td>
<td>Amplitude (Clock)</td>
<td>CAP</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CAP?</td>
</tr>
<tr>
<td>[6]</td>
<td>Amplitude (XClock)</td>
<td>XCA</td>
</tr>
<tr>
<td></td>
<td></td>
<td>XCA?</td>
</tr>
<tr>
<td>[7]</td>
<td>AC ON/OFF</td>
<td>CLA</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CLA?</td>
</tr>
<tr>
<td>[8]</td>
<td>Offset (Clock)</td>
<td>COS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>COS?</td>
</tr>
<tr>
<td>[9]</td>
<td>Offset (XClock)</td>
<td>XCF</td>
</tr>
<tr>
<td></td>
<td></td>
<td>XCF?</td>
</tr>
<tr>
<td>[10]</td>
<td>External ATT Factor (Clock)</td>
<td>CAF</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CAF?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>XAF?</td>
</tr>
<tr>
<td>[12]</td>
<td>Duty (Clock)</td>
<td>CDT</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CDT?</td>
</tr>
<tr>
<td>[13]</td>
<td>Amplitude</td>
<td>CLM</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CLM?</td>
</tr>
<tr>
<td>[14]</td>
<td>Offset Max/Min</td>
<td>CLO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CLO?</td>
</tr>
<tr>
<td>[15]</td>
<td>No label (Query for data/clock output setting completion status)</td>
<td>OCS? Refer to 8.3.1.2 &quot;Data/XData setting commands&quot; for details on this command.</td>
</tr>
</tbody>
</table>
### CGR <numeric>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;numeric&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>OFF</td>
</tr>
<tr>
<td>1</td>
<td>ON</td>
</tr>
</tbody>
</table>

**Function**
Sets the common setting for the Clock and XClock output ON or OFF.

**Example**
To set the common setting ON:
> CGR 1

**Compatibility**
Compatible with the MP1632C.

### CGR?

**Response**
<numeric> = <NR1 NUMERIC RESPONSE DATA> (FIX1)

**Function**
Queries the common setting for the Clock and XClock outputs ON or OFF state.

**Example**
> CGR?
< CGR 1

**Compatibility**
Compatible with the MP1632C.

### CLG <numeric>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;numeric&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>OFF</td>
</tr>
<tr>
<td>1</td>
<td>ON</td>
</tr>
</tbody>
</table>

**Function**
Sets the clock output range limitation ON or OFF.

**Example**
To set the clock output range limitation ON:
> CLG 1

**Compatibility**
Incompatible with existing models.

### CLG?

**Response**
<numeric> = <NR1 NUMERIC RESPONSE DATA> (FIX1)

**Function**
Queries the clock output range limitation ON or OFF state.

**Example**
> CLG?
< CLG 1

**Compatibility**
Incompatible with existing models.
Chapter 8  Native Commands

CLV(XLV)<level>
Parameter
0  Variable
1  NECL
2  PCML
3  NCML
4  SCFL
5  LVPECL
6  LVDS200 mV
7  LVDS400 mV

Function  Sets the level of the fixed interface for the Clock (XClock) output.
Example  To set the level of the Clock output to the NECL level:
> CLV 1
Compatibility  Incompatible with existing models.

CLV?(XLV?)
Response  <level> = <NR1 NUMERIC RESPONSE DATA> (FIX1)
Function  Queries the level of the fixed interface for the Clock (XClock) output.
Example  To query the level of the Clock output:
> CLV?
  < CLV 1
Compatibility  Incompatible with existing models.

CAP(XCA) <numeric>
Parameter  <numeric> = <DECIMAL NUMERIC PROGRAM DATA>
0.100 to 2.000  0.100 to 2.000 V, in 0.002-V steps

Function  Sets the amplitude of the Clock (XClock) output.
Example  To set the amplitude of the Clock output to 1.000 V:
> CAP 1.000
Compatibility  Compatible with the MP1632C, MP1761C and MP1775A.

CAP(XCA)?
Response  <numeric> = <NR2 NUMERIC RESPONSE DATA> (FIX5)
Function  Queries the amplitude of the Clock (XClock) output.
Example  To query the amplitude of the Clock output:
> CAP?
  < CAP 1.000
Compatibility  Compatible with the MP1632C, MP1761C and MP1775A.
8.3 PPG Commands

**CLA <numeric>**

| Parameter | <numeric> = <DECIMAL NUMERIC PROGRAM DATA>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Offset OFF (DC output)</td>
</tr>
<tr>
<td>1</td>
<td>Offset ON (AC output)</td>
</tr>
</tbody>
</table>

**Function**
Sets the clock output offset ON or OFF.

**Example**
To set the clock output offset ON:
> CLA 1

**Compatibility**
Incompatible with existing models.

**CLA?**

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;numeric&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt; (FIX1)</th>
</tr>
</thead>
</table>

**Function**
Queries the clock output offset ON or OFF state.

**Example**
> CLA?
< CLA 1

**Compatibility**
Incompatible with existing models.

**COS(XCF) <numeric>**

| Parameter | <numeric> = <DECIMAL NUMERIC PROGRAM DATA>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>−2.000 to 3.300</td>
<td>−2.000 to 3.300 Voh, in 0.001-V steps</td>
</tr>
<tr>
<td>−3.000 to 3.250</td>
<td>−3.000 to 3.250 Vth, in 0.001-V steps</td>
</tr>
<tr>
<td>−4.000 to 3.200</td>
<td>−4.000 to 3.200 Vol, in 0.001-V steps</td>
</tr>
</tbody>
</table>

**Function**
Sets the Clock (XClock) output offset.

**Example**
To set the XClock output offset to −1.000 Voh:
> XCF -1.000

**Compatibility**
Partially compatible with the MP1632C, MP1761C and MP1775A (only when <numeric> is set to a value within the range from −4.0 to 2.0).

**COS(XCF)?**

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;numeric&gt; = &lt;NR2 NUMERIC RESPONSE DATA&gt; (FIX6)</th>
</tr>
</thead>
</table>

**Function**
Queries the Clock (XClock) output offset.

**Example**
To query the XClock output offset:
> XCF?
< XCF -1.000
< XCF 2.000

**Compatibility**
Partially compatible with the MP1632C, MP1761C and MP1775A.
**Chapter 8  Native Commands**

### CAF(XAF)<numeric>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;numeric&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function</td>
<td>Sets the External ATT Factor of the Clock (XClock) output.</td>
</tr>
<tr>
<td>Example</td>
<td>To set the External ATT Factor of the Clock output to 20 dB:</td>
</tr>
<tr>
<td></td>
<td>&gt; CAF 20</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Incompatible with existing models.</td>
</tr>
</tbody>
</table>

### CAF(XAF)?

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;numeric&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt; (FIX2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function</td>
<td>Queries the External ATT Factor of the Clock (XClock) output.</td>
</tr>
<tr>
<td>Example</td>
<td>To query the External ATT Factor of the Clock output:</td>
</tr>
<tr>
<td></td>
<td>&gt; CAF?</td>
</tr>
<tr>
<td></td>
<td>&lt; CAF 20</td>
</tr>
<tr>
<td></td>
<td>&lt; CAF 0</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Incompatible with existing models.</td>
</tr>
</tbody>
</table>

### CDT <numeric>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;numeric&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function</td>
<td>Sets the Duty of the Clock output.</td>
</tr>
<tr>
<td>Example</td>
<td>To set the Duty of the Clock output to 25:</td>
</tr>
<tr>
<td></td>
<td>&gt; CDT -25</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Compatible with the MP1632C.</td>
</tr>
</tbody>
</table>

### CDT?

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;numeric&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt; (FIX3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function</td>
<td>Queries the Duty of the Clock output.</td>
</tr>
<tr>
<td>Example</td>
<td>&gt; CDT?</td>
</tr>
<tr>
<td></td>
<td>&lt; CDT -25</td>
</tr>
<tr>
<td></td>
<td>&lt; CDT -5</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Compatible with the MP1632C.</td>
</tr>
</tbody>
</table>
### CLM <numeric>

**Parameter**  
<numeric> = <DECIMAL NUMERIC PROGRAM DATA>
0.100 to 2.000 0.100 to 2.000 V, in 0.002-V steps

**Function**  
Sets the limitation value for the clock output amplitude.

**Example**  
To set the limitation value for the clock output amplitude to 2.000 V:
> CLM 2.000

**Compatibility**  
Incompatible with existing models.

### CLM?

**Response**  
<numeric> = <NR2 NUMERIC RESPONSE DATA> (FIX5)

**Function**  
Queries the limitation value for the clock output amplitude.

**Example**  
> CLM?
< CLM 2.000

**Compatibility**  
Incompatible with existing models.

### CLO <max>,<min>

**Parameter**  
<max> = <DECIMAL NUMERIC PROGRAM DATA>
–2.000 to 3.300 –2.000 to 3.300 V, in 0.001-V steps

<min> = <DECIMAL NUMERIC PROGRAM DATA>
–4.000 to 3.200 –4.000 to 3.200 V, in 0.001-V steps

**Function**  
Sets the limitation value for the clock output offset.

**Example**  
To set the limitation value for the clock output offset to 3.000 V max., –4.000 V min.:
> CLO 3.000,–4.000
> CLO 3.300,0.000

**Compatibility**  
Incompatible with existing models.

### CLO?

**Response**  
<max> = <NR2 NUMERIC RESPONSE DATA> (FIX6)

<min> = <NR2 NUMERIC RESPONSE DATA> (FIX6)

**Function**  
Queries the limitation value for the clock output offset.

**Example**  
> CLO?
< CLO 2.000,–4.000
< CLO –2.000,–4.000
< CLO 3.300,0.000

**Compatibility**  
Incompatible with existing models.
8.3.1.4 CMU Bit Rate setting commands

- **Reference Clock**: CRE, CRE?
- **External Clock**: CEC, CEC?
- **Frequency**: CRF, CRF?
- **Operation Rate**: COR, COR? (Frequency unit setting)

---

Table 8.3.1.4-1 CMU Bit Rate setting commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1]</td>
<td>Reference Clock</td>
<td>CRE</td>
</tr>
<tr>
<td>[2]</td>
<td>External Clock</td>
<td>CEC</td>
</tr>
<tr>
<td>[3]</td>
<td>Frequency</td>
<td>CRF</td>
</tr>
<tr>
<td>[4]</td>
<td>No label (Frequency unit setting)</td>
<td>CRS</td>
</tr>
<tr>
<td>[5]</td>
<td>Operation Rate</td>
<td>COR</td>
</tr>
</tbody>
</table>
8.3 PPG Commands

CRE <mode>

Parameter  
<mode> = <DECIMAL NUMERIC PROGRAM DATA>
0  Uses the internal reference signal.
1  Uses the external reference signal.

Function  
Selects whether to use the internal reference signal or the external reference signal for the clock reference signal.

Example  
To select the internal reference signal for the clock reference signal:
> CRE 0

Compatibility  
Incompatible with existing models.

CRE?

Response  
<mode> = <NR1 NUMERIC RESPONSE DATA> (FIX1)

Function  
Queries the clock reference signal.

Example  
> CRE?
< CRE 0

Compatibility  
Incompatible with existing models.

CEC <source>

Parameter  
<source> = <DECIMAL NUMERIC PROGRAM DATA>
0  1/1 Clock
1  1/64 Clock

Function  
Sets the division cycle when the external reference signal is used.

Example  
To set the division cycle to 1/1 Clock:
> CEC 0

Compatibility  
Incompatible with existing models.

CEC?

Response  
<source> = <NR1 NUMERIC RESPONSE DATA> (FIX1)

Function  
Queries the division cycle when the external reference signal is used.

Example  
> CEC?
< CEC 0

Compatibility  
Incompatible with existing models.
CRF <numeric>

Parameter

<numeric> = <DECIMAL NUMERIC PROGRAM DATA>

When the operation frequency setting resolution is MHz:

- 1225 to 1562
- 2450 to 3125
- 4900 to 6250
- 9800 to 12500

When the operation frequency setting resolution is kHz:

- 1225000 to 1562500
- 2450000 to 3125000
- 4900000 to 6250000
- 9800000 to 12500000

Default value is 12500000 kHz.

Function

Sets the operation frequency of the internal synthesizer.

Example

To set the operation frequency of the internal synthesizer to 1225 MHz:

> CRF 1225

Compatibility

Incompatible with existing models.

CRF?

Response

<numeric> = <NR1 NUMERIC RESPONSE DATA> (FIX5/8)

Function

Queries the operating frequency of the internal synthesizer.

Example

> CRF?

When the operation frequency setting resolution is MHz (FIX5):

< CRF 1225
< CRF 12500

When the operation frequency setting resolution is kHz (FIX8):

< CRF 12500000
< CRF 1225000

Compatibility

Incompatible with existing models.

CRS <numeric>

Parameter

<numeric> = <DECIMAL NUMERIC PROGRAM DATA>

- 0: Sets the kHz unit.
- 1: Sets the MHz unit.

Function

Selects the resolution for setting the operating frequency of the internal synthesizer.

Example

To set the resolution for setting the operating frequency of the internal synthesizer to kHz:

> CRS 0

Compatibility

Incompatible with existing models.
8.3 PPG Commands

CRS?

Response

<numeric> = <NR1 NUMERIC RESPONSE DATA> (FIX1)

Function

Queries the resolution for setting the operating frequency of the internal synthesizer.

Example

> CRS?
< CRS 0

Compatibility

Incompatible with existing models.

COR <numeric>

Parameter

<numeric> = <DECIMAL NUMERIC PROGRAM DATA>

1 Operates with 1/1 of external reference signal (12.5 to 9.8 GHz)
2 Operates with 1/2 of external reference signal (6.25 to 4.9 GHz)
4 Operates with 1/4 of external reference signal (3.125 to 2.45 GHz)
8 Operates with 1/8 of external reference signal (1.5625 to 1.225 GHz)

Function

Sets the operating frequency rate when the external reference signal is used for the reference clock.

Example

To set the operating frequency rate of the internal synthesizer to 1/4 (3.125 to 2.45 GHz):
> COR 4

Compatibility

Incompatible with existing models.

COR?

Response

<numeric> = <NR1 NUMERIC RESPONSE DATA> (FIX1)

Function

Queries the operating frequency rate setting state when the external reference signal is used for the reference clock.

Example

> COR?
< COR 4

Compatibility

Incompatible with existing models.
Chapter 8  Native Commands

8.3.2 Commands related to Pattern tab

8.3.2.1 Test Pattern setting commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1]</td>
<td>Test Pattern</td>
<td>PTS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PTS?</td>
</tr>
<tr>
<td>[2]</td>
<td>Logic</td>
<td>LGC</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LGC?</td>
</tr>
</tbody>
</table>

Figure 8.3.2-1 Pattern tab

Figure 8.3.2.1-1 Test Pattern setting field

Table 8.3.2.1-1 Test Pattern setting commands
8.3 PPG Commands

### PTS <type>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;type&gt;</td>
<td>&lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td></td>
<td>Alternate pattern</td>
</tr>
<tr>
<td>1</td>
<td></td>
<td>Data pattern</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>Zero-Substitution pattern</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>PRBS pattern</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>Mixed Data pattern</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>Mixed Alternate pattern</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>Sequence pattern</td>
</tr>
</tbody>
</table>

**Function**: Sets the type of the test pattern.

**Example**: To set the test pattern type to the Mixed Data pattern:

```plaintext
> PTS 4
```

**Compatibility**: Compatible with the MP1761C.

### PTS?

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;type&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt; (FIX1)</th>
</tr>
</thead>
</table>

**Function**: Queries the type of the test pattern.

**Example**: To query the type of the test pattern:

```plaintext
> PTS?
< PTS 4
```

**Compatibility**: Compatible with the MP1761C.

### LGC <logic>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;logic&gt;</td>
<td>&lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td></td>
<td>Positive (positive logic)</td>
</tr>
<tr>
<td>1</td>
<td></td>
<td>Negative (negative logic)</td>
</tr>
</tbody>
</table>

**Function**: Sets the logic (positive or negative) at test pattern generation.

**Example**: To set the logic of the test pattern to the negative logic (Negative):

```plaintext
> LGC 1
```

**Compatibility**: Compatible with the MP1632C, MP1761C and MP1775A.

### LGC?

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;logic&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt; (FIX1)</th>
</tr>
</thead>
</table>

**Function**: Queries the logic at test pattern generation.

**Example**: To query the logic at test pattern generation:

```plaintext
> LGC?
< LGC 1
```

**Compatibility**: Compatible with the MP1632C, MP1761C and MP1775A.
8.3.2.2 PRBS pattern setting commands

Table 8.3.2.2-1  PRBS pattern setting commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1]</td>
<td>Length</td>
<td>PTN</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PTN?</td>
</tr>
<tr>
<td>[2]</td>
<td>Mark Ratio</td>
<td>MRK</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MRK?</td>
</tr>
<tr>
<td>[3]</td>
<td>Bit Shift</td>
<td>SFT</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SFT?</td>
</tr>
</tbody>
</table>
### 8.3 PPG Commands

#### PTN <numeric>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;numeric&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>PRBS</strong></td>
</tr>
<tr>
<td>2</td>
<td>$2^{7-1}$</td>
</tr>
<tr>
<td>3</td>
<td>$2^{9-1}$</td>
</tr>
<tr>
<td>4</td>
<td>$2^{10-1}$</td>
</tr>
<tr>
<td>5</td>
<td>$2^{11-1}$</td>
</tr>
<tr>
<td>6</td>
<td>$2^{15-1}$</td>
</tr>
<tr>
<td>7</td>
<td>$2^{20-1}$</td>
</tr>
<tr>
<td>8</td>
<td>$2^{23-1}$</td>
</tr>
<tr>
<td>9</td>
<td>$2^{31-1}$</td>
</tr>
<tr>
<td></td>
<td>12 $2^{7-1}$</td>
</tr>
<tr>
<td></td>
<td>13 $2^{9-1}$</td>
</tr>
<tr>
<td></td>
<td>14 $2^{10-1}$</td>
</tr>
<tr>
<td></td>
<td>15 $2^{11-1}$</td>
</tr>
<tr>
<td></td>
<td>16 $2^{15-1}$</td>
</tr>
<tr>
<td></td>
<td>17 $2^{20-1}$</td>
</tr>
<tr>
<td></td>
<td>18 $2^{23-1}$</td>
</tr>
</tbody>
</table>

**Function:** Sets the number of stages ($2^n - 1$ ($n = 7, 9, 10, 11, 15, 20, 23,$ or 31)) during PRBS pattern or Zero-Substitution pattern generation.

**Example**

To set PRBS15:

```
> PTN 6
```

**Compatibility**

Compatible with the MP1632C, MP1761C and MP1775A.

#### PTN?

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;numeric&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt; (FIX2)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Function</strong></td>
<td>Queries the number of stages during PRBS pattern or Zero-Substitution pattern generation.</td>
</tr>
</tbody>
</table>

**Example**

```
> PTN?
< PTN 6
< PTN 18
```

**Compatibility**

Compatible with the MP1632C, MP1761C and MP1775A.
**Chapter 8  Native Commands**

**MRK <mratio>**

Parameter  

<mratio> = <DECIMAL NUMERIC PROGRAM DATA>  

<table>
<thead>
<tr>
<th></th>
<th>POS</th>
<th>NEG</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0/8</td>
<td>8/8</td>
</tr>
<tr>
<td>1</td>
<td>1/8</td>
<td>7/8</td>
</tr>
<tr>
<td>2</td>
<td>1/4</td>
<td>3/4</td>
</tr>
<tr>
<td>3</td>
<td>1/2</td>
<td>1/2INVT</td>
</tr>
</tbody>
</table>

Function Sets the mark ratio during PRBS pattern generation.

Example To set the mark ratio to 1/8:

> MRK 1

Compatibility Compatible with the MP1632C and MP1761C.

**MRK?**

Response  

<mratio> = <NR1 NUMERIC RESPONSE DATA> (FIX1)

Function Queries the mark ratio during PRBS pattern generation.

Example

> MRK?

< MRK 1

Compatibility Compatible with the MP1632C and MP1761C.

**SFT <numeric>**

Parameter  

<numeric> = <DECIMAL NUMERIC PROGRAM DATA>  

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1 bit</td>
</tr>
<tr>
<td>1</td>
<td>3 bits</td>
</tr>
</tbody>
</table>

Function Sets the bit shift during PRBS pattern generation.

Example To set the bit shift to 1 bit:

> SFT 0

Compatibility Compatible with the MP1632C and MP1761C.

**SFT?**

Response  

<numeric> = <NR1 NUMERIC RESPONSE DATA> (FIX1)

Function Queries the bit shift number during PRBS pattern generation.

Example

> SFT?

< SFT 0

Compatibility Compatible with the MP1632C and MP1761C.
8.3.2.3 Zero-Substitution pattern setting commands

![Zero-Substitution pattern setting field](image)

Table 8.3.2.3-1 Zero-Substitution pattern setting commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1]</td>
<td>Length</td>
<td>PTN</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Refer to 8.3.2.2 &quot;PRBS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>pattern setting commands&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>for detail on this command.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PTN?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Refer to 8.3.2.2 &quot;PRBS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>pattern setting commands&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>for detail on this command.</td>
</tr>
<tr>
<td>[2]</td>
<td>Zero-Substitution Length</td>
<td>ZLN</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ZLN?</td>
</tr>
<tr>
<td>[3]</td>
<td>Additional Bit</td>
<td>ZAD</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ZAD?</td>
</tr>
</tbody>
</table>

**ZLN <numeric>**

Parameter

<numeric> = <DECIMAL NUMERIC PROGRAM DATA>

When ZSUBLength, $2^n$ is set:

1 to $2^{n-1}$

1 to $2^{n-1}$, in 1-bit steps

When ZSUBLength, $2^{n-1}$ is set:

1 to $2^{n-2}$

1 to $2^{n-2}$, in 1-bit steps

$n = 7, 9, 11, 15, 20, 23$

Function

Sets the number of bits to which "0" is inserted (zero-insertion bit count) during Zero-Substitution pattern generation.

Available parameters vary depending on the setting conditions.

Example

To set the zero-insertion bit count to 10 bits:

> ZLN 10

Compatibility

Partially compatible with the MP1761C.
Chapter 8  Native Commands

ZLN?

Response  <numeric> = <NR1 NUMERIC RESPONSE DATA> (FIX7)
Function   Queries the zero-insertion bit count during Zero-Substitution pattern
generation.
Example    > ZLN?
           < ZLN       1
           < ZLN 8388607
Compatibility Partially compatible with the MP1632C, MP1761C and MP1775A.

ZAD <numeric>

Parameter  <numeric> = <DECIMAL NUMERIC PROGRAM DATA>
Function   0  Adds one bit of "0" next to the last of
            zero-inserted bits.
           1  Adds one bit of "1" next to the last of
            zero-inserted bits (same as existing models).
Example    To set one bit of "1" next to the last of zero-inserted bits:
           > ZAD 1
Compatibility Incompatible with existing models.

ZAD?

Response  <numeric> = <NR1 NUMERIC RESPONSE DATA> (FIX1)
Function   Queries whether to add one bit of "0" or "1" next to the last of
           zero-inserted bits.
Example    > ZAD?
           < ZAD 1
Compatibility Incompatible with existing models.
8.3.2.4 Data pattern setting commands

![Data pattern setting field](image1)

**Figure 8.3.2.4-1** Data pattern setting field

![Data pattern setting (Pattern Editor screen)](image2)

**Figure 8.3.2.4-2** Data pattern setting (Pattern Editor screen)

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1]</td>
<td>Data Length</td>
<td>DLN</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DLN?</td>
</tr>
<tr>
<td>[2]</td>
<td>No label (Pattern setting)</td>
<td>PDT</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PDT?</td>
</tr>
<tr>
<td>[3]</td>
<td>No label (Pattern setting)</td>
<td>BDT</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BDT?</td>
</tr>
<tr>
<td>[4]</td>
<td>No label (Data inversion)</td>
<td>DRA</td>
</tr>
<tr>
<td>[5]</td>
<td>No label (Data inversion)</td>
<td>DRD</td>
</tr>
<tr>
<td>[6]</td>
<td>No label (All 0 or All 1)</td>
<td>ALL</td>
</tr>
<tr>
<td>[7]</td>
<td>No label (Pattern setting)</td>
<td>WRT</td>
</tr>
<tr>
<td>[8]</td>
<td>No label (Pattern reading)</td>
<td>RED?</td>
</tr>
</tbody>
</table>
## Chapter 8  Native Commands

### DLN <numeric>

**Parameter**
- `<numeric>` = `<DECIMAL NUMERIC PROGRAM DATA>`
- 2 to 134217728 2 to 134,217,728 bit, in 1-bit steps

**Note:**
- At 2 ch Combination, the setting range and Step are doubled.
- At 4 ch Combination, the setting range and Step are quadrupled.

**Function**
Sets the pattern length during Data pattern generation.

**Example**
To set the pattern length to 100 bits:
> DLN 100

**Compatibility**
Compatible with the MP1632C, MP1761C and MP1775A.

### DLN?  

**Response**
- `<numeric>` = `<NR1 NUMERIC RESPONSE DATA>` (FIX10)

**Function**
Queries the pattern length during Data pattern generation.

**Example**
- > DLN?
- < DLN  2
- < DLN 134217728

**Compatibility**
Compatible with the MP1632C, MP1761C and MP1775A.

**Note:**
The number of query digits differs from existing models.
### 8.3 PPG Commands

#### PDT <start>,<end>,<data>

**Parameter**

- `<start>` = *<NON-DECIMAL NUMERIC PROGRAM DATA>*
  
  - #H0 to #H7FFFFFFF  0 to 7FFFFFFF bits, in 1-bit steps
  - (Specify in hexadecimal)

- `<end>` = *<NON-DECIMAL NUMERIC PROGRAM DATA>*
  
  - #H0 to #H7FFFFFFF  0 to 7FFFFFFF bits, in 1-bit steps
  - (Specify in hexadecimal)

**Note:**

- The maximum setting is the pattern length.
- At 2 ch Combination, the setting range is doubled.
- At 4 ch Combination, the setting range is quadrupled.

- `<data>` = *<NON-DECIMAL NUMERIC PROGRAM DATA>*
  
  - #H**...* Specify pattern data in hexadecimal format.
  - 1 to 400 characters (pattern data of 400 × 4 bits).
  - Specify a character string consisting of 0 to 9 and A to F.
  - If the last character string of the pattern data is short of a hexadecimal unit (4 bits), add 0(s) up to the last bit to make a hexadecimal string.

  - #B**...* Specify pattern data in binary format.
  - 1 to 400 characters (pattern data of 400 bits).
  - Specify a character string consisting of 0 and 1.

**Function**

Sets the pattern data from the `<start>` to `<end>` addresses during Data pattern generation.

The set data overwrites the specified range.

**Example**

- To set the addresses 0 to 1F of the pattern data to 5A:
  
  > PDT #H0, #H1F, #H5A

**Compatibility**

Compatible with the MP1632C.

**Note:**

- When the number of data bits specified in `<data>` is less than the range specified by `<start>` and `<end>`, set `<data>` repeatedly up to the specified range.
  
  **Example**

  - `<start>` = #H0, `<end>` = #H1F, `<data>` = #HABC
    
    Setting data: ABCABCABC
  
  - `<start>` = #H0, `<end>` = #H7, `<data>` = #B011
    
    Setting data: 01101101

- When the number of data bits specified in `<data>` is more than the range specified by `<start>` and `<end>`, cutoff the part of `<data>` out of the specified range.

  **Example**

  - `<start>` = #H0, `<end>` = #HF, `<data>` = #HABCDEF
**Setting data:** ABCD
- \(<\text{start}> = \#H0, \text{<end>} = \#H3, \text{<data>} = \#B01100110\)
  
  Setting data: 0110

---

**PDT? <start>**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>(&lt;\text{start}&gt; = \langle\text{NON-DECIMAL NUMERIC PROGRAM DATA}\rangle)</th>
</tr>
</thead>
<tbody>
<tr>
<td>#H0 to #H7FFFFFFF</td>
<td>0 to #H7FFFFFFF bits</td>
</tr>
<tr>
<td>(\text{Specify in hexadecimal})</td>
<td></td>
</tr>
</tbody>
</table>

**Note:**

- The maximum setting is the pattern length.
- At 2 ch Combination, the setting range is doubled.
- At 4 ch Combination, the setting range is quadrupled.

<table>
<thead>
<tr>
<th>Response</th>
<th>(&lt;\text{data}&gt; = \langle\text{HEXADECIMAL NUMERIC RESPONSE DATA}\rangle)</th>
</tr>
</thead>
<tbody>
<tr>
<td>#H**...*</td>
<td>Returns the pattern data in hexadecimal format.</td>
</tr>
<tr>
<td>(\text{Within 400 characters (pattern data of } 400 \times 4 \text{ bits)})</td>
<td></td>
</tr>
</tbody>
</table>

**Function:**

Queries the pattern data of \(400 \times 4\) bits from the \(<\text{start}>\) address during Data pattern generation.

**Example:**

To query the pattern data from address 0:

\[> \text{PDT? #H0}\]
\[< \text{PDT #H0123456789ABCDEF... (1 to 400 characters in hexadecimal)}\]
\[< \text{PDT #H0}\]

**Compatibility:**

Compatible with the MP1632C.
8.3 PPG Commands

**BDT <start>,<end>,<bdata>**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;start&gt;</td>
<td>&lt;NON-DECIMAL NUMERIC PROGRAM DATA&gt;</td>
</tr>
<tr>
<td></td>
<td>#H0 to #H7FFFFFFF 0 to 7FFFFFFF bits, in 1-bit steps (Specify in hexadecimal)</td>
</tr>
<tr>
<td>&lt;end&gt;</td>
<td>&lt;NON-DECIMAL NUMERIC PROGRAM DATA&gt;</td>
</tr>
<tr>
<td></td>
<td>#H0 to #H7FFFFFFF 0 to 7FFFFFFF bits, in 1-bit steps (Specify in hexadecimal)</td>
</tr>
<tr>
<td>&lt;bdata&gt;</td>
<td>&lt;ARBITRARY BLOCK PROGRAM DATA&gt;</td>
</tr>
<tr>
<td></td>
<td>#XYYY&lt;binary&gt;</td>
</tr>
<tr>
<td></td>
<td>X: Number of YYY digits</td>
</tr>
<tr>
<td></td>
<td>YYY: Number of bytes of &lt;binary&gt;</td>
</tr>
<tr>
<td></td>
<td>1 to 16,777,216 bytes</td>
</tr>
<tr>
<td></td>
<td>&lt;binary&gt;: Binary data up to 16,777,216 bytes</td>
</tr>
</tbody>
</table>

**Note:**
- The maximum setting is the pattern length.
- At 2 ch Combination, the setting range is doubled.
- At 4 ch Combination, the setting range is quadrupled.

**Function**
Sets binary data of the pattern data from <start> to <end> addresses during Data pattern generation. The set data overwrites the specified range.

**Example**
To set the addresses 0 to 1F of the pattern data to 41:
> BDT #H0,#H1F,#11A (A=41)

**Compatibility**
Compatible with the MP1632C.
Chapter 8  Native Commands

BDT? <start>[,<size>]

Parameter

- <start> = <NON-DECIMAL NUMERIC PROGRAM DATA>
  #H0 to #H7FFFFFFF  0 to 7FFFFFFF bits
  (Specify in hexadecimal)

Note:

- The maximum setting is the pattern length.
- At 2 ch Combination, the setting range is doubled.
- At 4 ch Combination, the setting range is quadrupled.

- [<size>] = <DECIMAL NUMERIC PROGRAM DATA>
  1 to 16777216  1 to 16,777,216, in 1-byte steps
  Binary data of the setting pattern is queried when [<size>] is omitted.

Response

- <bdata> = <DEFINITE LENGTH ARBITRARY BLOCK RESPONSE DATA>
  #XYYY<binary>
  X: Number of YYY digits
  YYY: Number of bytes of <binary>
  1 to 16,777,216 bytes
  <binary>: Binary data up to 16,777,216 bytes

Function

Queries binary data of bytes specified by <size> from <start> address of the pattern data during Data pattern generation.

Example

To query the pattern data during Data pattern generation for 2 bytes of binary data from address 0:

> BDT? #H0,2
< BDT #12AA

To query the pattern data during Data pattern generation for the setting pattern of binary data from address 0:

> BDT? #H0
< BDT #516000AA... (Binary data of 16,000 bytes)

Compatibility

Compatible with the MP1632C.

DRA <start>,<end>

Parameter

- <start> = <NON-DECIMAL NUMERIC PROGRAM DATA>
  #H0 to #H7FFFFFFF  0 to 7FFFFFFF bits, in 1-bit steps
  (Specify in hexadecimal)

- <end> = <NON-DECIMAL NUMERIC PROGRAM DATA>
  #H0 to #H7FFFFFFF  0 to 7FFFFFFF bits, in 1-bit steps
  (Specify in hexadecimal)

Note:

- The maximum setting is the pattern length.
- At 2 ch Combination, the setting range is doubled.
- At 4 ch Combination, the setting range is quadrupled.

Function

Inverts the data in the program during Data pattern generation. Specify
### 8.3 PPG Commands

#### DRD <start>,<delta>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;start&gt;</td>
<td>&lt;NON-DECIMAL NUMERIC PROGRAM DATA&gt;</td>
</tr>
<tr>
<td></td>
<td>#H0 to #H7FFFFFFF 0 to 7FFFFFFF bits</td>
</tr>
<tr>
<td></td>
<td>(Specify in hexadecimal)</td>
</tr>
<tr>
<td>&lt;delta&gt;</td>
<td>&lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
</tr>
<tr>
<td></td>
<td>1 to 134217728 1 to 134,217,728 bit</td>
</tr>
</tbody>
</table>

**Note:**
- The maximum setting is the pattern length.
- At 2 ch Combination, the setting range is doubled.
- At 4 ch Combination, the setting range is quadrupled.

**Function:**
Inverts the data in the program during Data pattern generation. Specify the inversion range by the number of bits from the <start> address (relative specification).

**Example**
To invert 128 bits of the pattern data from address 0:
> DRD #H0,128

**Compatibility**
Compatible with the MP1632C.

#### ALL <data>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;data&gt;</td>
<td>&lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
</tr>
<tr>
<td>0</td>
<td>Fills all bits with 0s.</td>
</tr>
<tr>
<td>1</td>
<td>Fills all bits with 1s.</td>
</tr>
</tbody>
</table>

**Function**
Sets 0 or 1 for all bits in the program during Data pattern generation.

**Example**
To set 1 for all bits in the pattern data:
> ALL 1

**Compatibility**
Compatible with the MP1632C, MP1761C and MP1775A.
Chapter 8  Native Commands

WRT <byte>,<address>

Parameter

- `<byte>` = <DECIMAL NUMERIC PROGRAM DATA>
  - Number of pattern transfer bytes
  - 1 to 16777216
  - 1 to 16,777,216 bytes, in 1-byte steps

  **Note:**
  - At 2 ch Combination, the setting range is doubled.
  - At 4 ch Combination, the setting range is quadrupled.

- `<address>` = <DECIMAL NUMERIC PROGRAM DATA>
  - Start address of pattern input
  - 0 to 134217720
  - 0 to 134,217,720, in single steps

  **Note:**
  - At 2 ch Combination, the setting range is doubled.
  - At 4 ch Combination, the setting range is quadrupled.

Function

Sets the number of bytes and the pattern input start address for binary transfer during Data pattern generation.

Example

To transfer 16-byte data from address 0:

```
> WRT 16,0
< ABCDEFGHIJKLMNOP
```

Compatibility

Compatible with the MP1632C, MP1761C and MP1775A.

RED? <byte>,<address>

Parameter

- `<byte>` = <DECIMAL NUMERIC PROGRAM DATA>
  - Number of pattern transfer bytes
  - 1 to 16777216
  - 1 to 16,777,216 bytes, in 1-byte steps

  **Note:**
  - At 2 ch Combination, the setting range is doubled.
  - At 4 ch Combination, the setting range is quadrupled.

- `<address>` = <DECIMAL NUMERIC PROGRAM DATA>
  - Start address of pattern input
  - 0 to 134217720
  - 0 to 134,217,720, in single steps

  **Note:**
  - At 2 ch Combination, the setting range is doubled.
  - At 4 ch Combination, the setting range is quadrupled.

Response

Data pattern row (binary data)

Function

Sets the number of bytes to be read and the pattern input start address using binary transfer during Data pattern generation.

Example

To read 16-byte data from address 0:

```
> RED? 16,0
< ABCDEFGHIJKLMNOP (Binary data of 16 bytes)
```

Compatibility

Compatible with the MP1632C, MP1761C and MP1775A.
8.3.2.5 Alternate pattern setting commands

Figure 8.3.2.5-1 Alternate pattern setting field

Figure 8.3.2.5-2 Alternate pattern setting (Pattern Editor screen)
### Table 8.3.2.5-1 Alternate pattern setting commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Loop Control</td>
<td>APS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>APS?</td>
</tr>
<tr>
<td>2</td>
<td>Manual</td>
<td>AMN</td>
</tr>
<tr>
<td>3</td>
<td>A/B LoopTime</td>
<td>LPT</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LPT?</td>
</tr>
<tr>
<td>4</td>
<td>Data Length/Alternate</td>
<td>ALE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ALE?</td>
</tr>
<tr>
<td>5</td>
<td>No label (Pattern setting)</td>
<td>ADW</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ADW?</td>
</tr>
<tr>
<td>6</td>
<td>No label (Pattern setting)</td>
<td>ABW</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ABW?</td>
</tr>
<tr>
<td>7</td>
<td>No label (Data inversion)</td>
<td>ARA</td>
</tr>
<tr>
<td>8</td>
<td>No label (Data inversion)</td>
<td>ARD</td>
</tr>
<tr>
<td>9</td>
<td>No label (All 0 or All 1)</td>
<td>ADF</td>
</tr>
</tbody>
</table>

**APS <loop>**

**Parameter**

<loop> = <DECIMAL NUMERIC PROGRAM DATA>

- 0 Internal: Switching control using internal signal
- 1 External: Switching control using external signal
- 2 Manual: Manual switching control

**Function**

Sets the Alternate pattern A/B switching control method.

**Example**

To set the external control for the Alternate pattern A/B switching:

```plaintext
> APS 1
```

**Compatibility**

Compatible with the MP1761C.

**APS?**

**Response**

<loop> = <NR1 NUMERIC RESPONSE DATA> (FIX1)

**Function**

Queries the Alternate pattern A/B switching control method.

**Example**

```plaintext
> APS?
< APS 1
```

**Compatibility**

Compatible with the MP1761C.

**AMN**

**Function**

Switches the Alternate pattern A/B manually.

**Example**

```plaintext
> AMN
```

**Compatibility**

Incompatible with existing models.
### LPT <content>,<numeric>

**Parameter**
- `<content>` = `<DECIMAL NUMERIC PROGRAM DATA>`
- 0 Pattern A
- 1 Pattern B
- `<numeric>` = `<DECIMAL NUMERIC PROGRAM DATA>`
- 1 to 511 1 to 511 times, in 1-time steps

**Function**
Sets the number of generation times of the Alternate pattern A or B.

**Example**
To set the number of pattern B generation times to 20:
> LPT 1,20

**Compatibility**
Incompatible with existing models.

### LPT? <content>

**Parameter**
- `<content>` = `<DECIMAL NUMERIC PROGRAM DATA>`
- 0 Pattern A
- 1 Pattern B

**Response**
- `<numeric>` = `<NR1 NUMERIC RESPONSE DATA>` (FIX3)

**Function**
Queries the number of generation times of the Alternate pattern A or B.

**Example**
To query the number of generation times of the Alternate pattern B:
> LPT? 1
< LPT 511
< LPT 1

**Compatibility**
Incompatible with existing models.

### ALE <content>,<numeric>

**Parameter**
- `<content>` = `<DECIMAL NUMERIC PROGRAM DATA>`
- 0 Pattern A
- 1 Pattern B
- `<numeric>` = `<DECIMAL NUMERIC PROGRAM DATA>`
- 128 to 67108864 128 to 67,108,864 bit, in 128-bit steps

**Function**
Sets the Alternate pattern length.

**Example**
To set the pattern length of the pattern B to 128 bits:
> ALE 1,128

**Compatibility**
Incompatible with existing models.
### Chapter 8  Native Commands

#### ALE? <content>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;content&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Pattern A</td>
</tr>
<tr>
<td>1</td>
<td>Pattern B</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;numeric&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt; (FIX8)</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Function</th>
<th>Queries the Alternate pattern length.</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Example</th>
<th>To query the pattern length of the Alternate pattern B:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>&gt; ALE? 1</td>
</tr>
<tr>
<td></td>
<td>&lt; ALE 128</td>
</tr>
<tr>
<td></td>
<td>&lt; ALE 67108864</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Compatibility</th>
<th>Incompatible with existing models.</th>
</tr>
</thead>
</table>

#### ADW <content>,<start>,<end>,<data>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;content&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Pattern A</td>
</tr>
<tr>
<td>1</td>
<td>Pattern B</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>&lt;start&gt;</th>
<th>&lt;NON-DECIMAL NUMERIC PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>#H0 to #H3FFFFFF</td>
<td>0 to 3FFFFFF bits, in 1-bit steps</td>
</tr>
<tr>
<td></td>
<td>(Specify in hexadecimal)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>&lt;end&gt;</th>
<th>&lt;NON-DECIMAL NUMERIC PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>#H0 to #H3FFFFFF</td>
<td>0 to 3FFFFFF bits, in 1-bit steps</td>
</tr>
<tr>
<td></td>
<td>(Specify in hexadecimal)</td>
</tr>
</tbody>
</table>

**Note:**
The maximum setting value is the pattern length.

<table>
<thead>
<tr>
<th>&lt;data&gt;</th>
<th>&lt;NON-DECIMAL NUMERIC PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>#H**...*</td>
<td>Specify pattern data in hexadecimal format.</td>
</tr>
<tr>
<td></td>
<td>1 to 400 characters (pattern data of 400 × 4 bits)</td>
</tr>
<tr>
<td></td>
<td>Specify a character string consisting of 0 to 9 and A to F.</td>
</tr>
<tr>
<td></td>
<td>If the last character string of the pattern data is short of a hexadecimal unit (4 bits), add 0(s) up to the last bit to make a hexadecimal string.</td>
</tr>
<tr>
<td>#B**...*</td>
<td>Specify pattern data in binary format.</td>
</tr>
<tr>
<td></td>
<td>1 to 400 characters (pattern data of 400 bits)</td>
</tr>
<tr>
<td></td>
<td>Specify a character string consisting of 0 and 1.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Function</th>
<th>Sets the pattern data from the &lt;start&gt; to &lt;end&gt; addresses during Alternate pattern generation.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>The set data overwrites the specified range.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Example</th>
<th>To set the addresses 0 to 1F of the pattern A to AA:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>&gt; ADW 0,#H0,#H1F,#HAA</td>
</tr>
</tbody>
</table>

| Compatibility | Incompatible with existing models. |
### ADW? <content>,<start>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;content&gt;</td>
<td>= &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
</tr>
<tr>
<td>0</td>
<td>Pattern A</td>
</tr>
<tr>
<td>1</td>
<td>Pattern B</td>
</tr>
<tr>
<td>&lt;start&gt;</td>
<td>= &lt;NON-DECIMAL NUMERIC PROGRAM DATA&gt;</td>
</tr>
<tr>
<td>#H0 to #H3FFFFFF</td>
<td>0 to 3FFFFFF bits</td>
</tr>
<tr>
<td></td>
<td>(Specify in hexadecimal)</td>
</tr>
</tbody>
</table>

**Note:**
The maximum setting value is the pattern length.

<table>
<thead>
<tr>
<th>Response</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;data&gt;</td>
<td>= &lt;HEXADECIMAL NUMERIC RESPONSE DATA&gt;</td>
</tr>
<tr>
<td>#H**...*</td>
<td>Returns the pattern data in hexadecimal format.</td>
</tr>
<tr>
<td></td>
<td>Within 400 characters (pattern data of 400 (\times) 4 bits)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Queries the pattern data of 400 (\times) 4 bits from the &lt;start&gt; address during Alternate pattern generation.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Example</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>To query the pattern A data from address 0:</td>
<td></td>
</tr>
<tr>
<td>&gt; ADW? 0,H0</td>
<td></td>
</tr>
<tr>
<td>&lt; ADW #H0123456789ABCDEF... (1 to 400 characters in Hex.)</td>
<td></td>
</tr>
<tr>
<td>&lt; ADW #H0</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Compatibility</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Incompatible with existing models.</td>
<td></td>
</tr>
</tbody>
</table>
### ABW <content>,<start>,<end>,<bdata>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;content&gt;</td>
<td>= &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
</tr>
<tr>
<td>0</td>
<td>Pattern A</td>
</tr>
<tr>
<td>1</td>
<td>Pattern B</td>
</tr>
<tr>
<td>&lt;start&gt;</td>
<td>= &lt;NON-DECIMAL NUMERIC PROGRAM DATA&gt;</td>
</tr>
<tr>
<td>#H0 to #H3FFFFFF</td>
<td>0 to 3FFFFFF bits, in 1-bit steps</td>
</tr>
<tr>
<td></td>
<td>(Specify in hexadecimal)</td>
</tr>
<tr>
<td>&lt;end&gt;</td>
<td>= &lt;NON-DECIMAL NUMERIC PROGRAM DATA&gt;</td>
</tr>
<tr>
<td>#H0 to #H3FFFFFF</td>
<td>0 to 3FFFFFF bits, in 1-bit steps</td>
</tr>
<tr>
<td></td>
<td>(Specify in hexadecimal)</td>
</tr>
</tbody>
</table>

**Note:**
The maximum setting value is the pattern length.

<table>
<thead>
<tr>
<th>&lt;bdata&gt;</th>
<th>= &lt;ARBITRARY BLOCK PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>#XYYY&lt;binary&gt;</td>
<td>X: Number of YYY digits</td>
</tr>
<tr>
<td></td>
<td>YYY: Number of bytes of &lt;binary&gt;</td>
</tr>
<tr>
<td></td>
<td>1 to 8,388,608 bytes</td>
</tr>
<tr>
<td></td>
<td>&lt;binary&gt;: Binary data up to 8,388,608 bytes</td>
</tr>
</tbody>
</table>

**Function**
Sets the binary data from <start> to <end> addresses of the pattern data during Alternate pattern generation.
The set data overwrites the specified range.

**Example**
To set the addresses 0 to 1F of the pattern A data to 42:

```plaintext
> ABW 0,#H0,#H1F,#11B (B=42)
```

**Compatibility**
Incompatible with existing models.
### ABW? <content>,<start>[,<size>]

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;content&gt;</td>
<td>= &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
</tr>
<tr>
<td>0</td>
<td>Pattern A</td>
</tr>
<tr>
<td>1</td>
<td>Pattern B</td>
</tr>
<tr>
<td>&lt;start&gt;</td>
<td>= &lt;NON-DECIMAL NUMERIC PROGRAM DATA&gt;</td>
</tr>
<tr>
<td>#H0 to #H3FFFFFF</td>
<td>0 to 3FFFFFF bits, in 1-bit steps</td>
</tr>
</tbody>
</table>

(Specify in hexadecimal)

**Note:**
The maximum setting value is the pattern length.

[size] = <DECIMAL NUMERIC PROGRAM DATA >
1 to 8388608 1 to 8,388,608, in 1-byte steps

<table>
<thead>
<tr>
<th>Response</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;bdata&gt;</td>
<td>= &lt;DEFINITE LENGTH ARBITRARY BLOCK RESPONSE DATA&gt;</td>
</tr>
<tr>
<td>#XYYY&lt;binary&gt;</td>
<td>X: Number of YYY digits</td>
</tr>
<tr>
<td></td>
<td>YYY: Number of bytes of &lt;binary&gt;</td>
</tr>
<tr>
<td></td>
<td>1 to 8,388,608 bytes</td>
</tr>
</tbody>
</table>

<binary>: Binary data up to 8,388,608 bytes

Binary data of the setting pattern is queried when [size] is omitted.

**Function**
Queries binary data of bytes specified by <size> from the <start> address of the pattern data during Alternate pattern generation.

**Example**
To query binary data of 2 bytes from address 0 of the pattern A data:

> ABW? 0,#H0,2
< ABW #12AA

To query binary data of the setting pattern from address 0 of the pattern A data:

> ABW? 0,#H0
< ABW #516000AA... (Binary data of 8,388,608 bytes)

**Compatibility**
Incompatible with existing models.
ARA <content>,<start>,<end>

Parameter

<content> = <DECIMAL NUMERIC PROGRAM DATA>
0  Pattern A
1  Pattern B
<start> = <NON-DECIMAL NUMERIC PROGRAM DATA>
#H0 to #H3FFFFFF 0 to 3FFFFFF bits, in 1-bit steps
(Specify in hexadecimal)
<end> = <NON-DECIMAL NUMERIC PROGRAM DATA>
#H0 to #H3FFFFFF 0 to 3FFFFFF bits, in 1-bit steps
(Specify in hexadecimal)

Note:
The maximum setting value is the pattern length.

Function
Inverts the data in the program during Alternate pattern generation.
Specify the inversion range by the <start> and <end> addresses.

Example
To invert addresses 0 to 1F of the pattern A:
> ARA 0,#H0,#H1F

Compatibility
Incompatible with existing models.

ARD <content>,<start>,<delta>

Parameter

<content> = <DECIMAL NUMERIC PROGRAM DATA>
0  Pattern A
1  Pattern B
<start> = <NON-DECIMAL NUMERIC PROGRAM DATA>
#H0 to #H3FFFFFF 0 to 3FFFFFF bits, in 1-bit steps
(Specify in hexadecimal)
<delta> = <DECIMAL NUMERIC PROGRAM DATA>
1 to 67108864 1 to 67,108,864 bit, in 1-bit steps

Note:
The maximum setting value is the pattern length.

Function
Inverts the data in the program during Alternate pattern generation.
Specify the inversion range by the number of bits from the <start>
address (relative specification).

Example
To invert 64 bits of the pattern A from address 0:
> ARD 0,#H0,64

Compatibility
Incompatible with existing models.
### ADF <content>,<range>,<page>,<data>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;content&gt;</td>
<td>= DECIMAL NUMERIC PROGRAM DATA</td>
</tr>
<tr>
<td>0</td>
<td>Pattern A</td>
</tr>
<tr>
<td>1</td>
<td>Pattern B</td>
</tr>
<tr>
<td>&lt;range&gt;</td>
<td>= DECIMAL NUMERIC PROGRAM DATA</td>
</tr>
<tr>
<td>0</td>
<td>Specifies a page.</td>
</tr>
<tr>
<td></td>
<td>(One page is defined as 128 bits.)</td>
</tr>
<tr>
<td>1</td>
<td>Specifies all data.</td>
</tr>
<tr>
<td>&lt;page&gt;</td>
<td>= DECIMAL NUMERIC PROGRAM DATA</td>
</tr>
<tr>
<td>1 to 524288</td>
<td>1 to 524,288 (Max.) pages, in 1-page steps</td>
</tr>
<tr>
<td>Max</td>
<td>= Pattern Length / 128 (rounding up fractions)</td>
</tr>
<tr>
<td></td>
<td>Specify &quot;0&quot; when &lt;range&gt; is set to ALL.</td>
</tr>
<tr>
<td>&lt;data&gt;</td>
<td>= NUMERIC PROGRAM DATA</td>
</tr>
<tr>
<td>0</td>
<td>Fills the specified range with 0s.</td>
</tr>
<tr>
<td>1</td>
<td>Fills the specified range with 1s.</td>
</tr>
</tbody>
</table>

**Function**
Sets 0s or 1s for the specified page or all data in the program during Alternate pattern generation.

**Example**
To set 0s for the second page of the pattern A:

```
> ADF 0,0,2,0
```

**Compatibility**
Incompatible with existing models.
8.3.2.6 Mixed Data pattern setting commands

Figure 8.3.2.6-1 Mixed Data pattern setting field

Figure 8.3.2.6-2 Descramble Setup dialog box
8.3 PPG Commands

Native Commands

Figure 8.3.2.6-3  Mixed Data pattern setting (Pattern Editor screen)

Table 8.3.2.6-1  Mixed Data pattern setting commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1]</td>
<td>Bit Shift</td>
<td>MPB</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MPP</td>
</tr>
<tr>
<td>[2]</td>
<td>PRBS Pattern</td>
<td>MPP</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MPP?</td>
</tr>
<tr>
<td>[3]</td>
<td>PRBS Mark Ratio</td>
<td>MPM</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MPM?</td>
</tr>
<tr>
<td>[4]</td>
<td>Scramble ON/OFF</td>
<td>MSC</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MSC?</td>
</tr>
<tr>
<td>[5]</td>
<td>PRBS Sequence</td>
<td>MPS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MPS?</td>
</tr>
<tr>
<td>[6]</td>
<td>Scramble Setup</td>
<td>MSS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MSS?</td>
</tr>
<tr>
<td>[7]</td>
<td>Set All/Reset All</td>
<td>MSL</td>
</tr>
<tr>
<td>[8]</td>
<td>Number of Block</td>
<td>MBL</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MBL?</td>
</tr>
<tr>
<td>[9]</td>
<td>Row Length</td>
<td>MRL</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MRL?</td>
</tr>
<tr>
<td>[10]</td>
<td>Data Length</td>
<td>MRW</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MRW?</td>
</tr>
<tr>
<td>[11]</td>
<td>Number of Row</td>
<td>MDL</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MDL?</td>
</tr>
</tbody>
</table>
### Table 8.3.2.6-1  Mixed Data pattern setting commands (Cont’d)

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>[12]</td>
<td>No label (Pattern setting)</td>
<td>MDW</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MDW?</td>
</tr>
<tr>
<td>[13]</td>
<td>No label (Pattern setting)</td>
<td>MBW</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MBW?</td>
</tr>
<tr>
<td>[14]</td>
<td>No label (Data inversion)</td>
<td>MDA</td>
</tr>
<tr>
<td>[15]</td>
<td>No label (Data inversion)</td>
<td>MDD</td>
</tr>
<tr>
<td>[16]</td>
<td>No label (All 0 or All 1)</td>
<td>MDF</td>
</tr>
</tbody>
</table>

#### MPB <numeric>

**Parameter**

<numeric> = <DECIMAL NUMERIC PROGRAM DATA>

0  1 bit
1  3 bits

**Function**

Sets the bit shift for the PRBS pattern during Mixed Data pattern generation.

**Example**

To set the bit shift to 3 bits:

> MPB 1

**Compatibility**

Incompatible with existing models.

#### MPB?

**Response**

<numeric> = <NR1 NUMERIC RESPONSE DATA> (FIX1)

**Function**

Queries the bit shift for the PRBS pattern during Mixed Data pattern generation.

**Example**

> MPB?

< MPB 1

**Compatibility**

Incompatible with existing models.
### MPP <numeric>

**Parameter**

<numeric> = <DECIMAL NUMERIC PROGRAM DATA>

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>(2^{7-1})</td>
</tr>
<tr>
<td>3</td>
<td>(2^{9-1})</td>
</tr>
<tr>
<td>4</td>
<td>(2^{10-1})</td>
</tr>
<tr>
<td>5</td>
<td>(2^{11-1})</td>
</tr>
<tr>
<td>6</td>
<td>(2^{15-1})</td>
</tr>
<tr>
<td>7</td>
<td>(2^{20-1})</td>
</tr>
<tr>
<td>8</td>
<td>(2^{23-1})</td>
</tr>
<tr>
<td>9</td>
<td>(2^{31-1})</td>
</tr>
</tbody>
</table>

**Function**

Sets the number of stages of the PRBS pattern during Mixed Data pattern generation.

**Example**

To set the number of stages of the PRBS pattern to 15:

> MPP 6

**Compatibility**

Incompatible with existing models.

### MPP?

**Response**

<numeric> = <NR1 NUMERIC RESPONSE DATA> (FIX1)

**Function**

Queries the number of stages of the PRBS pattern during Mixed Data pattern generation.

**Example**

> MPP?

< MPP 6

**Compatibility**

Incompatible with existing models.

### MPM <mratio>

**Parameter**

<mratio> = <DECIMAL NUMERIC PROGRAM DATA>

<table>
<thead>
<tr>
<th>Value</th>
<th>POS</th>
<th>NEG</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0/8</td>
<td>8/8</td>
</tr>
<tr>
<td>1</td>
<td>1/8</td>
<td>7/8</td>
</tr>
<tr>
<td>2</td>
<td>1/4</td>
<td>3/4</td>
</tr>
<tr>
<td>3</td>
<td>1/2</td>
<td>1/2INVT</td>
</tr>
</tbody>
</table>

**Function**

Sets the mark ratio for the PRBS pattern during Mixed Data pattern generation.

**Example**

To set the mark ratio for the PRBS pattern to 1/4:

> MPM 2

**Compatibility**

Incompatible with existing models.
Chapter 8  Native Commands

MPM?
Response  <mratio> = <NR1 NUMERIC RESPONSE DATA> (FIX1)
Function  Queries the mark ratio for the PRBS pattern during Mixed Data pattern generation.
Example  > MPM?
< MPM 2
Compatibility  Incompatible with existing models.

MSC <numeric>
Parameter  <numeric> = <DECIMAL NUMERIC PROGRAM DATA>
  0  Scramble OFF
  1  Scramble ON
Function  Sets whether to scramble the PRBS7 stage before pattern generation.
Example  To set Scramble ON:
> MSC 1
Compatibility  Incompatible with existing models.

MSC?
Response  <numeric> = <NR1 NUMERIC RESPONSE DATA> (FIX1)
Function  Queries whether to scramble the PRBS7 stage before pattern generation.
Example  > MSC?
< MSC 1
Compatibility  Incompatible with existing models.

MPS <sequence>
Parameter  <sequence> = <DECIMAL NUMERIC PROGRAM DATA>
  0  Restart: PRBS patterns are not continuous.
  1  Consecutive: PRBS patterns are continuous.
Function  Sets the PRBS pattern sequence during Mixed Data pattern generation.
Example  To set the PRBS pattern sequence to Restart:
> MPS 0
Compatibility  Incompatible with existing models.
8.3  PPG Commands

**MPS?**

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;sequence&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt; (FIX1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function</td>
<td>Queries the PRBS pattern sequence during Mixed Data pattern generation.</td>
</tr>
<tr>
<td>Example</td>
<td>&gt; MPS?</td>
</tr>
<tr>
<td></td>
<td>&lt; MPS 0</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Incompatible with existing models.</td>
</tr>
</tbody>
</table>

**MSS <row>,<data>,<prbs>**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;row&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1 to 16</td>
</tr>
<tr>
<td></td>
<td>1 to 16 rows, in 1-row steps</td>
</tr>
<tr>
<td>&lt;data&gt;</td>
<td>&lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
</tr>
<tr>
<td></td>
<td>0 Scramble OFF</td>
</tr>
<tr>
<td></td>
<td>1 Scramble ON</td>
</tr>
<tr>
<td>&lt;prbs&gt;</td>
<td>&lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
</tr>
<tr>
<td></td>
<td>0 Scramble OFF</td>
</tr>
<tr>
<td></td>
<td>1 Scramble ON</td>
</tr>
<tr>
<td>Function</td>
<td>Sets Scramble ON or OFF for the Data and PRBS of the specified Row.</td>
</tr>
<tr>
<td>Example</td>
<td>To set the Scramble settings for 2 Row to Data ON, PRBS OFF:</td>
</tr>
<tr>
<td></td>
<td>&gt; MSS 2,1,0</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Incompatible with existing models.</td>
</tr>
</tbody>
</table>

**MSS? <row>**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;row&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1 to 16</td>
</tr>
<tr>
<td></td>
<td>1 to 16 rows, in 1-row steps</td>
</tr>
<tr>
<td>Response</td>
<td>&lt;data&gt;,&lt;prbs&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt; (FIX1)</td>
</tr>
<tr>
<td>Function</td>
<td>Queries the scramble setting (ON/OFF) for the Data and PRBS of the specified Row.</td>
</tr>
<tr>
<td>Example</td>
<td>To query the Scramble setting for 2 Row:</td>
</tr>
<tr>
<td></td>
<td>&gt; MSS? 2</td>
</tr>
<tr>
<td></td>
<td>&lt; MSS 1,0</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Incompatible with existing models.</td>
</tr>
</tbody>
</table>
Chapter 8  Native Commands

MSL <numeric>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;numeric&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>All Reset</td>
</tr>
<tr>
<td>1</td>
<td>All Set</td>
</tr>
</tbody>
</table>

Function
Sets or resets all Scramble settings.

Example
To select all Scramble settings:
> MSL 1

Compatibility
Incompatible with existing models.

MBL <numeric>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;numeric&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 to 511</td>
<td>1 to 511 blocks, in 1-block steps</td>
</tr>
</tbody>
</table>

Function
Sets the number of blocks during Mixed Data pattern generation.

Example
To set the number of blocks to 30:
> MBL 30

Compatibility
Incompatible with existing models.

MBL?

Response
<numeric> = <NR1 NUMERIC RESPONSE DATA> (FIX3)

Function
Queries the number of blocks during Mixed Data pattern generation.

Example
> MBL?
< MBL   1
< MBL 511

Compatibility
Incompatible with existing models.

MRL <numeric>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;numeric&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>768 to 2281701376</td>
<td>768 to 2,281,701,376 bit, in 128-bit steps</td>
</tr>
</tbody>
</table>

Note:
At 2 ch Combination, the setting range and Step are doubled.
At 4 ch Combination, the setting range and Step are quadrupled.

Function
Sets the pattern length of 1 Row that is edited during Mixed Data pattern generation.

Example
To set the pattern length to 768 bits:
> MRL 768

Compatibility
Incompatible with existing models.
### MRL?

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;numeric&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt; (FIX11)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function</td>
<td>Queries the pattern length of 1 Row that is edited during Mixed Data pattern generation.</td>
</tr>
<tr>
<td>Example</td>
<td>&gt; MRL?</td>
</tr>
<tr>
<td></td>
<td>&lt; MRL 768</td>
</tr>
<tr>
<td></td>
<td>&lt; MRL 2281701376</td>
</tr>
<tr>
<td></td>
<td>&lt; MRL 9126805504</td>
</tr>
</tbody>
</table>

Compatibility: Incompatible with existing models.

### MRW <numeric>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;numeric&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1 to 16 1 to 16 rows, in 1-row steps</td>
</tr>
<tr>
<td>Function</td>
<td>Sets the number of rows within one block that are edited during Mixed Data pattern generation.</td>
</tr>
<tr>
<td>Example</td>
<td>To set the number of rows within one block to 16:</td>
</tr>
<tr>
<td></td>
<td>&gt; MRW 16</td>
</tr>
</tbody>
</table>

Compatibility: Incompatible with existing models.

### MRW?

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;numeric&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt; (FIX2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function</td>
<td>Queries the number of rows within one block that are edited during Mixed Data pattern generation.</td>
</tr>
<tr>
<td>Example</td>
<td>&gt; MRW?</td>
</tr>
<tr>
<td></td>
<td>&lt; MRW 16</td>
</tr>
<tr>
<td></td>
<td>&lt; MRW 1</td>
</tr>
</tbody>
</table>

Compatibility: Incompatible with existing models.

### MDL <numeric>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;numeric&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>512 to 134217728 512 to 134,217,728 bit, in 1-bit steps</td>
</tr>
</tbody>
</table>

**Note:**

- The maximum setting is the pattern length.
- At 2 ch Combination, the setting range and Step are doubled.
- At 4 ch Combination, the setting range and Step are quadrupled.

| Function | Sets the Data pattern length that is edited during Mixed Data pattern generation. |
| Example  | To set the pattern length to 512 bits: |
|          | > MDL 512 |

Compatibility: Incompatible with existing models.
Chapter 8  Native Commands

MDL?

Response  

<numeric> = <NR1 NUMERIC RESPONSE DATA> (FIX10)

Function  

Queries the Data pattern length that is edited during Mixed Data pattern generation.

Example  

> MDL?
< MDL 512
< MDL 134217728

Compatibility  

Incompatible with existing models.

MDW <block>,<start>,<end>,<data>

Parameter  

<block> = <DECIMAL NUMERIC PROGRAM DATA>
1 to 511  1 to 511 blocks, in 1-block steps

<start> = <NON-DECIMAL NUMERIC PROGRAM DATA>
#H0 to #H7FFFFFFF 0 to 7FFFFFFF bits, in 1-bit steps
(Specify in hexadecimal)

<end> = <NON-DECIMAL NUMERIC PROGRAM DATA>
#H0 to #H7FFFFFFF 0 to 7FFFFFFF bits, in 1-bit steps
(Specify in hexadecimal)

Note:

The maximum setting is the pattern length.
At 2 ch Combination, the setting range is doubled.
At 4 ch Combination, the setting range is quadrupled.

<data> = <NON-DECIMAL NUMERIC PROGRAM DATA>

#H**...* Specify pattern data in hexadecimal format.
1 to 400 characters (pattern data of 400 \times 4 bits)
Specify a character string consisting of 0 to 9 and A to F.
If the last character string of the pattern data is short of a hexadecimal unit (4 bits), add 0(s) up to the last bit to make a hexadecimal string.

#B**...* Specify pattern data in binary format.
1 to 400 characters (pattern data of 400 bits)
Specify a character string consisting of 0 and 1.

Function  

Sets the pattern data from the <start> to <end> addresses during Mixed Data pattern generation.
The set data overwrites the specified range.

Example  

To set the addresses 0 to 1F of the block 1 pattern data to 11:
> MDW 1,#H0,#H1F,#H11

Compatibility  

Incompatible with existing models.
### MDW? <block>,<start>

**Parameter**

- `<block>` = `<DECIMAL NUMERIC PROGRAM DATA>`
  - 1 to 511 blocks, in 1-block steps
- `<start>` = `<NON-DECIMAL NUMERIC PROGRAM DATA>`
  - #H0 to #H7FFFFFFF (0 to 7FFFFFFF bits, in 1-bit steps)

**Note:**

- The maximum setting is the pattern length.
- At 2 ch Combination, the setting range is doubled.
- At 4 ch Combination, the setting range is quadrupled.

**Response**

- `<data>` = `<HEXADECIMAL NUMERIC RESPONSE DATA>`
  - "H***" Returns the pattern data in hexadecimal format.
  - Within 400 characters (pattern data of 400 × 4 bits)

**Function**

Queries the pattern data of 400 × 4 bits from the `<start>` address during Mixed Data pattern generation.

**Example**

To query the block 1 pattern data from address 0:

```
> MDW? 1,#H0
< MDW #H0123456789ABCDEF... (1 to 400 characters in Hex.)
< MDW #H0
```

**Compatibility**

Incompatible with existing models.
**MBW,<block>,<start>,<end>,<bdata>**

**Parameter**
- `<block>` = `<DECIMAL NUMERIC PROGRAM DATA>`
  - 1 to 511 blocks, in 1-block steps
- `<start>` = `<NON-DECIMAL NUMERIC PROGRAM DATA>`
  - #H0 to #H7FFFFFFF 0 to 7FFFFFFF bits, in 1-bit steps
  - (Specify in hexadecimal)
- `<end>` = `<NON-DECIMAL NUMERIC PROGRAM DATA>`
  - #H0 to #H7FFFFFFF 0 to 7FFFFFFF bits, in 1-bit steps
  - (Specify in hexadecimal)

**Note:**
- The maximum setting is the pattern length.
- At 2 ch Combination, the setting range is doubled.
- At 4 ch Combination, the setting range is quadrupled.

- `<bdata>` = `<ARBITRARY BLOCK PROGRAM DATA>`
  - #XYY<binary>
    - X: Number of YYY digits
    - YYY: Number of bytes of <binary>, 1 to 16,777,216 bytes
  - <binary>: Binary data up to 16,777,216 bytes

**Function**
Sets the binary data from `<start>` to `<end>` addresses of the pattern data during Mixed Data pattern generation.
The set data overwrites the specified range.

**Example**
To set the addresses 0 to 1F of the block 1 pattern data to 43:
> MBW 1,#H0,#H1F,#11C (C=43)

**Compatibility**
Incompatible with existing models.
## MBW? <block>,<start>[,<size>]

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;block&gt;</td>
<td>&lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
</tr>
<tr>
<td>1 to 511</td>
<td>1 to 511 blocks, in 1-block steps</td>
</tr>
<tr>
<td>&lt;start&gt;</td>
<td>&lt;NON-DECIMAL NUMERIC PROGRAM DATA&gt;</td>
</tr>
<tr>
<td>#H0 to #H1FFFFFFF</td>
<td>0 to 1FFFFFFF bits, in 1-bit steps</td>
</tr>
<tr>
<td>(Specify in hexadecimal)</td>
<td></td>
</tr>
</tbody>
</table>

**Note:**
- The maximum setting is the pattern length.
- At 2 ch Combination, the setting range is doubled.
- At 4 ch Combination, the setting range is quadrupled.

<table>
<thead>
<tr>
<th>&lt;size&gt;</th>
<th>&lt;DECIMAL NUMERIC PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 to 16777216</td>
<td>1 to 16,777,216, in 1-byte steps</td>
</tr>
</tbody>
</table>

**Response**
- <bdata> = <DEFINITE LENGTH ARBITRARY BLOCK RESPONSE DATA>
- Binary data of the setting pattern is queried when [<size>] is omitted.

**Function**
Queries binary data of bytes specified by <size> from the <start> address of the pattern data during Mixed Data pattern generation.

**Example**
- To query binary data of 2 bytes in the block 1 starting from address 0:
  
  ```
  > MBW? 1,#H0,2
  < MBW #12AA
  ```

- To query binary data of the setting pattern in the block 1 starting from address 0:
  
  ```
  > MBW? 1,#H0
  < MBW #516000AA... (Binary data of 16,777,216 bytes)
  ```

**Compatibility**
- Incompatible with existing models.
Chapter 8  Native Commands

MDA <block>,<start>,<end>

Parameter
- `<block>` = `<DECIMAL NUMERIC PROGRAM DATA>`
  - 1 to 511 blocks, in 1-block steps
- `<start>` = `<NON-DECIMAL NUMERIC PROGRAM DATA>`
  - #H0 to #H7FFFFFFF bits, in 1-bit steps
    (Specify in hexadecimal)
- `<end>` = `<NON-DECIMAL NUMERIC PROGRAM DATA>`
  - #H0 to #H7FFFFFFF bits, in 1-bit steps
    (Specify in hexadecimal)

*Note:* The maximum setting is the pattern length.

At 2 ch Combination, the setting range is doubled.

At 4 ch Combination, the setting range is quadrupled.

Function
Inverts the data in the program during Mixed Data pattern generation.
Specify the inversion range by the `<start>` and `<end>` addresses.

Example
To invert addresses 0 to 4F of the block 1 pattern data:
> MDA 1,#H0,#H4F

Compatibility
Incompatible with existing models.

MDD <block>,<start>,<delta>

Parameter
- `<block>` = `<DECIMAL NUMERIC PROGRAM DATA>`
  - 1 to 511 blocks, in 1-block steps
- `<start>` = `<NON-DECIMAL NUMERIC PROGRAM DATA>`
  - #H0 to #H7FFFFFFF bits, in 1-bit steps
    (Specify in hexadecimal)
- `<delta>` = `<DECIMAL NUMERIC PROGRAM DATA>`
  - 1 to 134,217,728 bit, in 1-bit steps

*Note:* The maximum setting is the pattern length.

At 2 ch Combination, the setting range is doubled.

At 4 ch Combination, the setting range is quadrupled.

Function
Inverts the data in the program during Mixed Data pattern generation.
Specify the inversion range by the number of bits from the `<start>` address (relative specification).

Example
To invert 256 bits of the block 1 pattern data from address 0:
> MDD 1,#H0,256

Compatibility
Incompatible with existing models.
MDF <block>,<range>,<page>,<data>

Parameter

| <block> = <DECIMAL NUMERIC PROGRAM DATA> |
| 1 to 511 | 1 to 511 blocks, in 1-block steps |
| <range> = <DECIMAL NUMERIC PROGRAM DATA> |
| 0 | Specifies a page. |
| 1 | Specifies all data. |
| <page> = <DECIMAL NUMERIC PROGRAM DATA> |
| 1 to 1048576 | 1 to 1,048,576 (Max.) pages, in 1-page steps |
| Max = Pattern Length / 128 (rounding up fractions) |
| Specify "0" when <range> is set to ALL. |
| <data> = <DECIMAL NUMERIC PROGRAM DATA> |
| 0 | Fills the specified range with 0s. |
| 1 | Fills the specified range with 1s. |

Function
Sets 0s or 1s for the specified page or all data in the program during Mixed Data pattern generation.

Example
To set 0s for the third page:
> MDF 1,0,3,0

Compatibility
Incompatible with existing models.
8.3.2.7 Mixed Alternate pattern setting commands

Figure 8.3.2.7-1 Mixed Alternate pattern setting field
### 8.3 PPG Commands

#### Native Commands

![Figure 8.3.2.7-2 Descramble Setup dialog box](image)

![Figure 8.3.2.7-3 Mixed Alternate pattern setting (Pattern Editor screen)](image)
Table 8.3.2.7-1  Mixed Alternate pattern setting commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1]</td>
<td>Bit Shift</td>
<td>XPB</td>
</tr>
<tr>
<td></td>
<td></td>
<td>XPB?</td>
</tr>
<tr>
<td>[2]</td>
<td>Loop Control</td>
<td>XLC</td>
</tr>
<tr>
<td></td>
<td></td>
<td>XLC?</td>
</tr>
<tr>
<td>[4]</td>
<td>Loop Time</td>
<td>XPT</td>
</tr>
<tr>
<td></td>
<td></td>
<td>XPT?</td>
</tr>
<tr>
<td>[5]</td>
<td>Pattern</td>
<td>XPP</td>
</tr>
<tr>
<td></td>
<td></td>
<td>XPP?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>XPM?</td>
</tr>
<tr>
<td>[7]</td>
<td>Scramble ON/OFF</td>
<td>XSC</td>
</tr>
<tr>
<td></td>
<td></td>
<td>XSC?</td>
</tr>
<tr>
<td>[8]</td>
<td>PRBS Sequence</td>
<td>XPS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>XPS?</td>
</tr>
<tr>
<td>[9]</td>
<td>Scramble Setup</td>
<td>XSR</td>
</tr>
<tr>
<td></td>
<td></td>
<td>XSR?</td>
</tr>
<tr>
<td>[10]</td>
<td>Set All/Reset All</td>
<td>XSA</td>
</tr>
<tr>
<td>[11]</td>
<td>Number of Block</td>
<td>XBL</td>
</tr>
<tr>
<td></td>
<td></td>
<td>XBL?</td>
</tr>
<tr>
<td>[12]</td>
<td>Row Length</td>
<td>XRL</td>
</tr>
<tr>
<td></td>
<td></td>
<td>XRL?</td>
</tr>
<tr>
<td>[13]</td>
<td>Data Length</td>
<td>XML</td>
</tr>
<tr>
<td></td>
<td></td>
<td>XML?</td>
</tr>
<tr>
<td>[14]</td>
<td>Number of Row</td>
<td>XRW</td>
</tr>
<tr>
<td></td>
<td></td>
<td>XRW?</td>
</tr>
<tr>
<td>[15]</td>
<td>No label (Pattern setting)</td>
<td>XDW</td>
</tr>
<tr>
<td></td>
<td></td>
<td>XDW?</td>
</tr>
<tr>
<td>[16]</td>
<td>No label (Pattern setting)</td>
<td>XBW</td>
</tr>
<tr>
<td></td>
<td></td>
<td>XBW?</td>
</tr>
<tr>
<td>[17]</td>
<td>No label (Data inversion)</td>
<td>XDA</td>
</tr>
<tr>
<td>[18]</td>
<td>No label (Data inversion)</td>
<td>XDD</td>
</tr>
<tr>
<td>[19]</td>
<td>No label (All 0 or All 1)</td>
<td>XDF</td>
</tr>
</tbody>
</table>
8.3 PPG Commands

XPB <numeric>
Parameter  
<numeric> = <DECIMAL NUMERIC PROGRAM DATA>
1  1 bit
3  3 bits
Function  
Sets the bit shift for the PRBS pattern during Mixed Alternate pattern generation.
Example  
To set the bit shift to 3-bit shift:
> XPB 1
Compatibility  
Incompatible with existing models.

XPB?
Response  
<numeric> = <NR1 NUMERIC RESPONSE DATA> (FIX1)
Function  
Queries the bit shift for the PRBS pattern during Mixed Alternate pattern generation.
Example  
> XPB?
< XPB 1
Compatibility  
Incompatible with existing models.

XLC <loop>
Parameter  
<loop> = <DECIMAL NUMERIC PROGRAM DATA>
0  Internal:  Switching control using internal signal
1  External:  Switching control using external signal
2  Manual:  Manual switching control
Function  
Sets the pattern A/B switching control method during Mixed Alternate pattern generation.
Example  
To set the manual control for the pattern A/B switching:
> XLC 2
Compatibility  
Incompatible with existing models.

XLC?
Response  
<loop> = <NR1 NUMERIC RESPONSE DATA> (FIX1)
Function  
Queries the pattern A/B switching control method during Mixed Alternate pattern generation.
Example  
> XLC?
< XLC 2
Compatibility  
Incompatible with existing models.
## XMN

<table>
<thead>
<tr>
<th>Function</th>
<th>Generates the pattern B for the specified number of Loop times during Mixed Alternate pattern generation.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example</td>
<td>&gt; XMN</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Incompatible with existing models.</td>
</tr>
</tbody>
</table>

## XPT \(<content>,<numeric>\)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>(&lt;content&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Pattern A</td>
</tr>
<tr>
<td>1</td>
<td>Pattern B</td>
</tr>
<tr>
<td>(&lt;numeric&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;)</td>
<td></td>
</tr>
<tr>
<td>1 to 511</td>
<td>1 to 511 times, in 1-time steps</td>
</tr>
<tr>
<td>Function</td>
<td>Sets the number of generation times of the Mixed Alternate pattern A or B.</td>
</tr>
<tr>
<td>Example</td>
<td>To set the number of pattern B generation times to 20:</td>
</tr>
<tr>
<td></td>
<td>&gt; XPT 1,20</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Incompatible with existing models.</td>
</tr>
</tbody>
</table>

## XPT? \(<content>\)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>(&lt;content&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt; (FIX3))</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Pattern A</td>
</tr>
<tr>
<td>1</td>
<td>Pattern B</td>
</tr>
<tr>
<td>Response</td>
<td>(&lt;numeric&gt; = NR1 NUMERIC RESPONSE DATA&gt;)</td>
</tr>
<tr>
<td>Function</td>
<td>Queries the number of generation times of the Mixed Alternate pattern A or B.</td>
</tr>
<tr>
<td>Example</td>
<td>To set the number of pattern B generation times:</td>
</tr>
<tr>
<td></td>
<td>&gt; XPT? 1</td>
</tr>
<tr>
<td></td>
<td>&lt; XPT 1</td>
</tr>
<tr>
<td></td>
<td>&lt; XPT 511</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Incompatible with existing models.</td>
</tr>
</tbody>
</table>
### PPG Commands

#### XPP \(<\text{numeric}>\) Parameter

- **Function**: Sets the number of stages of the PRBS pattern during Mixed Alternate pattern generation.
- **Example**: To set the number of stages of the PRBS pattern to 7:
  ```plaintext
  > XPP 2
  ```
- **Compatibility**: Incompatible with existing models.

#### XPP? Parameter

- **Function**: Queries the number of stages of the PRBS pattern during Mixed Alternate pattern generation.
- **Example**: To set the number of stages of the PRBS pattern to 7:
  ```plaintext
  > XPP?
  < XPP 2
  ```
- **Compatibility**: Incompatible with existing models.

#### XPM \(<\text{mratio}>\) Parameter

- **Function**: Sets the mark ratio for the PRBS pattern during Mixed Alternate pattern generation.
- **Example**: To set the mark ratio for the PRBS pattern to 1/8:
  ```plaintext
  > XPM 1
  ```
- **Compatibility**: Incompatible with existing models.

---

<table>
<thead>
<tr>
<th>(\text{mratio})</th>
<th>POS</th>
<th>NEG</th>
</tr>
</thead>
<tbody>
<tr>
<td>(0)</td>
<td>0/8</td>
<td>8/8</td>
</tr>
<tr>
<td>(1)</td>
<td>1/8</td>
<td>7/8</td>
</tr>
<tr>
<td>(2)</td>
<td>1/4</td>
<td>3/4</td>
</tr>
<tr>
<td>(3)</td>
<td>1/2</td>
<td>1/2</td>
</tr>
<tr>
<td>(\text{INVT})</td>
<td>0/8</td>
<td>8/8</td>
</tr>
</tbody>
</table>

---

<table>
<thead>
<tr>
<th>(\text{numeric})</th>
<th>(\text{NR1 NUMERIC RESPONSE DATA})</th>
</tr>
</thead>
<tbody>
<tr>
<td>(0)</td>
<td>0</td>
</tr>
<tr>
<td>(1)</td>
<td>1</td>
</tr>
<tr>
<td>(2)</td>
<td>2</td>
</tr>
<tr>
<td>(3)</td>
<td>3</td>
</tr>
<tr>
<td>(4)</td>
<td>4</td>
</tr>
<tr>
<td>(5)</td>
<td>5</td>
</tr>
<tr>
<td>(6)</td>
<td>6</td>
</tr>
<tr>
<td>(7)</td>
<td>7</td>
</tr>
<tr>
<td>(8)</td>
<td>8</td>
</tr>
<tr>
<td>(9)</td>
<td>9</td>
</tr>
<tr>
<td>(10)</td>
<td>10</td>
</tr>
<tr>
<td>(11)</td>
<td>11</td>
</tr>
<tr>
<td>(12)</td>
<td>12</td>
</tr>
<tr>
<td>(13)</td>
<td>13</td>
</tr>
<tr>
<td>(14)</td>
<td>14</td>
</tr>
<tr>
<td>(15)</td>
<td>15</td>
</tr>
<tr>
<td>(16)</td>
<td>16</td>
</tr>
<tr>
<td>(17)</td>
<td>17</td>
</tr>
<tr>
<td>(18)</td>
<td>18</td>
</tr>
<tr>
<td>(19)</td>
<td>19</td>
</tr>
<tr>
<td>(20)</td>
<td>20</td>
</tr>
<tr>
<td>(21)</td>
<td>21</td>
</tr>
<tr>
<td>(22)</td>
<td>22</td>
</tr>
<tr>
<td>(23)</td>
<td>23</td>
</tr>
<tr>
<td>(24)</td>
<td>24</td>
</tr>
<tr>
<td>(25)</td>
<td>25</td>
</tr>
<tr>
<td>(26)</td>
<td>26</td>
</tr>
<tr>
<td>(27)</td>
<td>27</td>
</tr>
<tr>
<td>(28)</td>
<td>28</td>
</tr>
<tr>
<td>(29)</td>
<td>29</td>
</tr>
<tr>
<td>(30)</td>
<td>30</td>
</tr>
<tr>
<td>(31)</td>
<td>31</td>
</tr>
<tr>
<td>(32)</td>
<td>32</td>
</tr>
<tr>
<td>(33)</td>
<td>33</td>
</tr>
<tr>
<td>(34)</td>
<td>34</td>
</tr>
<tr>
<td>(35)</td>
<td>35</td>
</tr>
<tr>
<td>(36)</td>
<td>36</td>
</tr>
<tr>
<td>(37)</td>
<td>37</td>
</tr>
<tr>
<td>(38)</td>
<td>38</td>
</tr>
<tr>
<td>(39)</td>
<td>39</td>
</tr>
<tr>
<td>(40)</td>
<td>40</td>
</tr>
<tr>
<td>(41)</td>
<td>41</td>
</tr>
<tr>
<td>(42)</td>
<td>42</td>
</tr>
<tr>
<td>(43)</td>
<td>43</td>
</tr>
<tr>
<td>(44)</td>
<td>44</td>
</tr>
<tr>
<td>(45)</td>
<td>45</td>
</tr>
<tr>
<td>(46)</td>
<td>46</td>
</tr>
<tr>
<td>(47)</td>
<td>47</td>
</tr>
<tr>
<td>(48)</td>
<td>48</td>
</tr>
<tr>
<td>(49)</td>
<td>49</td>
</tr>
<tr>
<td>(50)</td>
<td>50</td>
</tr>
<tr>
<td>(51)</td>
<td>51</td>
</tr>
<tr>
<td>(52)</td>
<td>52</td>
</tr>
<tr>
<td>(53)</td>
<td>53</td>
</tr>
<tr>
<td>(54)</td>
<td>54</td>
</tr>
<tr>
<td>(55)</td>
<td>55</td>
</tr>
<tr>
<td>(56)</td>
<td>56</td>
</tr>
<tr>
<td>(57)</td>
<td>57</td>
</tr>
<tr>
<td>(58)</td>
<td>58</td>
</tr>
<tr>
<td>(59)</td>
<td>59</td>
</tr>
<tr>
<td>(60)</td>
<td>60</td>
</tr>
<tr>
<td>(61)</td>
<td>61</td>
</tr>
<tr>
<td>(62)</td>
<td>62</td>
</tr>
<tr>
<td>(63)</td>
<td>63</td>
</tr>
<tr>
<td>(64)</td>
<td>64</td>
</tr>
<tr>
<td>(65)</td>
<td>65</td>
</tr>
<tr>
<td>(66)</td>
<td>66</td>
</tr>
<tr>
<td>(67)</td>
<td>67</td>
</tr>
<tr>
<td>(68)</td>
<td>68</td>
</tr>
<tr>
<td>(69)</td>
<td>69</td>
</tr>
<tr>
<td>(70)</td>
<td>70</td>
</tr>
<tr>
<td>(71)</td>
<td>71</td>
</tr>
<tr>
<td>(72)</td>
<td>72</td>
</tr>
<tr>
<td>(73)</td>
<td>73</td>
</tr>
<tr>
<td>(74)</td>
<td>74</td>
</tr>
<tr>
<td>(75)</td>
<td>75</td>
</tr>
<tr>
<td>(76)</td>
<td>76</td>
</tr>
<tr>
<td>(77)</td>
<td>77</td>
</tr>
<tr>
<td>(78)</td>
<td>78</td>
</tr>
<tr>
<td>(79)</td>
<td>79</td>
</tr>
<tr>
<td>(80)</td>
<td>80</td>
</tr>
<tr>
<td>(81)</td>
<td>81</td>
</tr>
<tr>
<td>(82)</td>
<td>82</td>
</tr>
<tr>
<td>(83)</td>
<td>83</td>
</tr>
<tr>
<td>(84)</td>
<td>84</td>
</tr>
<tr>
<td>(85)</td>
<td>85</td>
</tr>
<tr>
<td>(86)</td>
<td>86</td>
</tr>
<tr>
<td>(87)</td>
<td>87</td>
</tr>
<tr>
<td>(88)</td>
<td>88</td>
</tr>
<tr>
<td>(89)</td>
<td>89</td>
</tr>
<tr>
<td>(90)</td>
<td>90</td>
</tr>
<tr>
<td>(91)</td>
<td>91</td>
</tr>
<tr>
<td>(92)</td>
<td>92</td>
</tr>
<tr>
<td>(93)</td>
<td>93</td>
</tr>
<tr>
<td>(94)</td>
<td>94</td>
</tr>
<tr>
<td>(95)</td>
<td>95</td>
</tr>
<tr>
<td>(96)</td>
<td>96</td>
</tr>
<tr>
<td>(97)</td>
<td>97</td>
</tr>
<tr>
<td>(98)</td>
<td>98</td>
</tr>
<tr>
<td>(99)</td>
<td>99</td>
</tr>
<tr>
<td>(100)</td>
<td>100</td>
</tr>
</tbody>
</table>
Chapter 8 Native Commands

XPM?
Response
<mratio> = <NR1 NUMERIC RESPONSE DATA> (FIX1)
Function Queries the mark ratio for the PRBS pattern during Mixed Alternate pattern generation.
Example
> XPM?
< XPM 1
Compatibility Incompatible with existing models.

XSC <numeric>
Parameter <numeric> = <DECIMAL NUMERIC PROGRAM DATA>
0 Scramble OFF
1 Scramble ON
Function Sets whether to scramble the PRBS7 stage before pattern generation.
Example To set Scramble ON:
> XSC 1
Compatibility Incompatible with existing models.

XSC?
Response <numeric> = <NR1 NUMERIC RESPONSE DATA> (FIX1)
Function Queries whether to scramble the PRBS7 stage before pattern generation.
Example
> XSC?
< XSC 1
Compatibility Incompatible with existing models.

XPS <sequence>
Parameter <sequence> = <DECIMAL NUMERIC PROGRAM DATA>
0 Restart: PRBS patterns are not continuous.
1 Consecutive: PRBS patterns are continuous.
Function Sets the PRBS pattern sequence during Mixed Alternate pattern generation.
Example To set the PRBS pattern sequence to Restart:
> XPS 0
Compatibility Incompatible with existing models.
### 8.3 PPG Commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
<th>Example</th>
<th>Compatibility</th>
</tr>
</thead>
</table>
| **XPS?** | Queries the PRBS pattern sequence during Mixed Alternate pattern generation. | > XPS?  
< XPS 0 | Incompatible with existing models. |

#### XSR <row>,<data>,<prbs>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Example</th>
<th>Compatibility</th>
</tr>
</thead>
</table>
| <row> = <DECIMAL NUMERIC PROGRAM DATA> | 1 to 16  
1 to 16 rows, in 1-row steps | > XSR 3,1,0 | Incompatible with existing models. |
| <data> = <DECIMAL NUMERIC PROGRAM DATA> | 0 Scramble OFF  
1 Scramble ON | | |
| <prbs> = <DECIMAL NUMERIC PROGRAM DATA> | 0 Scramble OFF  
1 Scramble ON | | |

Sets Scramble ON or OFF for the Data and PRBS of the specified Row.

#### XSR? <row>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Example</th>
<th>Compatibility</th>
</tr>
</thead>
</table>
| <row> = <DECIMAL NUMERIC PROGRAM DATA> | 1 to 16  
1 to 16 rows, in 1-row steps | > XSR? 3  
< XSR 1,0 | Incompatible with existing models. |
Chapter 8  Native Commands

XSA <numeric>
Parameter  
<numeric> = <DECIMAL NUMERIC PROGRAM DATA>
  0  All Reset
  1  All Set
Function  
Sets or resets all Scramble settings.
Example  
To select all Scramble settings:
  > XSA 1
Compatibility  
Incompatible with existing models.

XBL <numeric>
Parameter  
<numeric> = <DECIMAL NUMERIC PROGRAM DATA>
  1 to 511  1 to 511 blocks, in 1-block steps
Function  
Sets the number of blocks during Mixed Alternate pattern generation.
Example  
To set the number of blocks to 12:
  > XBL 12
Compatibility  
Incompatible with existing models.

XBL?
Response  
<numeric> = <NR1 NUMERIC RESPONSE DATA> (FIX3)
Function  
Queries the number of blocks during Mixed Alternate pattern generation.
Example  
> XBL?
  < XBL 1
  < XBL 511
Compatibility  
Incompatible with existing models.

XRL<numeric>
Parameter  
<numeric> = <DECIMAL NUMERIC PROGRAM DATA>
  768 to 2214592512  768 to 2,214,592,512 bits, in 128-bit steps
Function  
Sets the pattern length of 1 Row that is edited during Mixed Alternate pattern generation.
Example  
To set the pattern length of 1 Row to 768 bits:
  > XRL 768
Compatibility  
Incompatible with existing models.
**XRL?**

Response: 

<nemonic> = <NR1 NUMERIC RESPONSE DATA> (FIX10)

Function:
Queries the pattern length of 1 Row that is edited during Mixed Alternate pattern generation.

Example:

> XRL?
< XRL 768
< XRL 2214592512

Compatibility:
Incompatible with existing models.

**XML <numeric>**

Parameter:

<nemonic> = <DECIMAL NUMERIC PROGRAM DATA>
512 to 67108864 512 to 67,108,864 bits, in 128-bit steps

Function:
Sets the Data pattern length that is edited during Mixed Alternate pattern generation.

Example:
To set the Data pattern length to 12,800 bits:

> XML 12800

Compatibility:
Incompatible with existing models.

**XML?**

Response:

<nemonic> = <NR1 NUMERIC RESPONSE DATA> (FIX8)

Function:
Queries the Data pattern length that is edited during Mixed Alternate pattern generation.

Example:

> XML?
< XML 512
< XML 67108864

Compatibility:
Incompatible with existing models.

**XRW <numeric>**

Parameter:

<nemonic> = <DECIMAL NUMERIC PROGRAM DATA>
1 to 16 1 to 16 rows, in 1-row steps

Function:
Sets the number of rows within one block that are edited during Mixed Alternate pattern generation.

Example:
To set the number of rows within one block to 8:

> XRW 8

Compatibility:
Incompatible with existing models.
Chapter 8  Native Commands

XRW?

Response  

<numeric> = <NR1 NUMERIC RESPONSE DATA> (FIX2)

Function  
Queries the number of rows within one block that are edited during Mixed Alternate pattern generation.

Example  

> XRW?
< XRW  8
< XRW  16

Compatibility  
Incompatible with existing models.

XDW <block>,<content>,<start>,<end>,<data>

Parameter  

<block> = <DECIMAL NUMERIC PROGRAM DATA>
1 to 511 1 to 511 blocks, in 1-block steps

<content> = <DECIMAL NUMERIC PROGRAM DATA>
0 Pattern A
1 Pattern B

<start> = <NON-DECIMAL NUMERIC PROGRAM DATA>
#H0 to #H3FFFFFF 0 to 3FFFFFF bits, in 1-bit steps
(Specify in hexadecimal)

<end> = <NON-DECIMAL NUMERIC PROGRAM DATA>
#H0 to #H3FFFFFF 0 to 3FFFFFF bits, in 1-bit steps
(Specify in hexadecimal)

Note:  
The maximum setting value is the pattern length.

<data> = <NON-DECIMAL NUMERIC PROGRAM DATA>
#H**...* Specify pattern data in hexadecimal format.
1 to 400 characters (pattern data of 400 \times 4 bits)
Specify a character string consisting of 0 to 9 and A to F.
If the last character string of the pattern data is short of a hexadecimal unit (4 bits), add 0(s) up to the last bit to make a hexadecimal string.

#B**...* Specify pattern data in binary format.
1 to 400 characters (pattern data of 400 bits)
Specify a character string consisting of 0 and 1.

Function  
Sets the pattern data from the <start> to <end> addresses during Mixed Alternate pattern generation.
The set data overwrites the specified range.

Example  
To set the addresses 0 to 1F of the block 1 pattern B data to AA:

> XDW  1,1,#H0,#H1F,#HAA

Compatibility  
Incompatible with existing models.
### XDW? <block>,<content>,<start>

**Parameter**
- `<block>` = `<DECIMAL NUMERIC PROGRAM DATA>`
  - 1 to 511: 1 to 511 blocks, in 1-block steps
- `<content>` = `<DECIMAL NUMERIC PROGRAM DATA>`
  - 0: Pattern A
  - 1: Pattern B
- `<start>` = `<NON-DECIMAL NUMERIC PROGRAM DATA>`
  - #H0 to #H3FFFFFF: 0 to 3FFFFFF bit, in 1-bit steps
    (Specify in hexadecimal)

**Note:**
The maximum setting value is the pattern length.

**Response**
- `<data>` = `<HEXADECIMAL NUMERIC RESPONSE DATA>`
  - #H**...**: Returns the pattern data in hexadecimal format.
    Within 400 characters (pattern data of 400 \times 4 bits)

**Function**
Queries the pattern data of 400 \times 4 bits from the `<start>` address during Mixed Alternate pattern generation.

**Example**
To query the block 1 pattern B data from address 0:
```plaintext
> XDW? 1,1,H0
< XDW #H0123456789ABCDEF... (1 to 400 characters in Hex.)
< XDW #H0
```

**Compatibility**
Incompatible with existing models.
XBW <block>,<content>,<start>,<end>,<bdata>

Parameter

- `<block>` = `<DECIMAL NUMERIC PROGRAM DATA>`
  - 1 to 511: 1 to 511 blocks, in 1-block steps
- `<content>` = `<DECIMAL NUMERIC PROGRAM DATA>`
  - 0: Pattern A
  - 1: Pattern B
- `<start>` = `<NON-DECIMAL NUMERIC PROGRAM DATA>`
  - #H0 to #H3FFFFFF: 0 to 3FFFFFF bits, in 1-bit steps
    - (Specify in hexadecimal)
- `<end>` = `<NON-DECIMAL NUMERIC PROGRAM DATA>`
  - #H0 to #H3FFFFFF: 0 to 3FFFFFF bits, in 1-bit steps
    - (Specify in hexadecimal)

Note:
The maximum setting value is the pattern length.

- `<bdata>` = `<ARBITRARY BLOCK PROGRAM DATA>`
  - #XYYY<binary>
    - X: Number of YYY digits
    - YYY: Number of bytes of `<binary>`
    - 1 to 8,388,608 bytes
  - `<binary>`: Binary data up to 8,388,608 bytes

Function
Sets binary data of the pattern data from `<start>` to `<end>` addresses during Mixed Alternate pattern generation.
The set data overwrites the specified range.

Example
To set the addresses 0 to 1F of the block 1 pattern B data to 42:
> XBW 1,1,#H0,#H1F,#11B (B=42)

Compatibility
Incompatible with existing models.
### XBW? <block>,<content>,<start>[,<size>]

**Parameter**
- `<block>` = `<DECIMAL NUMERIC PROGRAM DATA>`
  - 1 to 511: 1 to 511 blocks, in 1-block steps
- `<content>` = `<DECIMAL NUMERIC PROGRAM DATA>`
  - 0: Pattern A
  - 1: Pattern B
- `<start>` = `<NON-DECIMAL NUMERIC PROGRAM DATA>`
  - #H0 to #H3FFFFFF: 0 to 3FFFFFF bit, in 1-bit steps
    - (Specify in hexadecimal)

**Note:**
The maximum setting value is the pattern length.

- `[<size>]` = `<DECIMAL NUMERIC PROGRAM DATA>`
  - 1 to 8388608: 1 to 8,388,608, in 1-byte steps

**Response**
- `<bdata>` = `<DEFINITE LENGTH ARBITRARY BLOCK RESPONSE DATA>`
  - Binary data of the setting pattern is queried when `[<size>]` is omitted.

**Function**
Queries binary data of bytes specified by `<size>` from `<start>` address of the pattern data during Mixed Alternate pattern generation.

**Example**
To query binary data of 2 bytes in the block 1 pattern B starting from address 0:

```
> XBW? 1,1#H0,2
< XBW #12AA
```

To query binary data of the setting pattern in the block 1 pattern A starting from address 0:

```
> XBW? 1,0,#H0
< XBW #516000AA... (Binary data of 16,000 bytes)
```

**Compatibility**
Incompatible with existing models.
### XDA \( \langle \text{block} \rangle, \langle \text{content} \rangle, \langle \text{start} \rangle, \langle \text{end} \rangle \)

**Parameter**

- \( \langle \text{block} \rangle = \langle \text{DECIMAL NUMERIC PROGRAM DATA} \rangle \)
  - 1 to 511: 1 to 511 blocks, in 1-block steps
- \( \langle \text{content} \rangle = \langle \text{DECIMAL NUMERIC PROGRAM DATA} \rangle \)
  - 0: Pattern A
  - 1: Pattern B
- \( \langle \text{start} \rangle = \langle \text{NON-DECIMAL NUMERIC PROGRAM DATA} \rangle \)
  - \#H0 to \#H3FFFFFF: 0 to 3FFFFFF bit, in 1-bit steps
    (Specify in hexadecimal)
- \( \langle \text{end} \rangle = \langle \text{NON-DECIMAL NUMERIC PROGRAM DATA} \rangle \)
  - \#H0 to \#H3FFFFFF: 0 to 3FFFFFF bit, in 1-bit steps
    (Specify in hexadecimal)

**Note:**
The maximum setting value is the pattern length.

**Function**

Inverts the data in the program during Mixed Alternate pattern generation. Specify the inversion range by the \( \langle \text{start} \rangle \) and \( \langle \text{end} \rangle \) addresses.

**Example**

To invert addresses 0 to 1F of the block 1 pattern B:

\[
> \text{XDA} \ 1, 1, \#H0, \#H1F
\]

**Compatibility**

Incompatible with existing models.

### XDD \( \langle \text{block} \rangle, \langle \text{content} \rangle, \langle \text{start} \rangle, \langle \text{delta} \rangle \)

**Parameter**

- \( \langle \text{block} \rangle = \langle \text{DECIMAL NUMERIC PROGRAM DATA} \rangle \)
  - 1 to 511: 1 to 511 blocks, in 1-block steps
- \( \langle \text{content} \rangle = \langle \text{DECIMAL NUMERIC PROGRAM DATA} \rangle \)
  - 0: Pattern A
  - 1: Pattern B
- \( \langle \text{start} \rangle = \langle \text{NON-DECIMAL NUMERIC PROGRAM DATA} \rangle \)
  - \#H0 to \#H3FFFFFF: 0 to 3FFFFFF bit, in 1-bit steps
    (Specify in hexadecimal)
- \( \langle \text{delta} \rangle = \langle \text{DECIMAL NUMERIC PROGRAM DATA} \rangle \)
  - 1 to 67108864: 1 to 67,108,864 bits, in 1-bit steps

**Note:**
The maximum setting value is the pattern length.

**Function**

Inverts the data in the program during Mixed Alternate pattern generation. Specify the inversion range by the number of bits from the \( \langle \text{start} \rangle \) address (relative specification).

**Example**

To invert 64 bits of the block 1 pattern B from address 0:

\[
> \text{XDD} \ 1, 1, \#H0, 64
\]

**Compatibility**

Incompatible with existing models.
8.3 PPG Commands

### XDF <block>,<content>,<range>,<page>,<data>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| <block>   | <DECIMAL NUMERIC PROGRAM DATA>  
1 to 511  | 1 to 511 blocks, in 1-block steps |
| <content> | <DECIMAL NUMERIC PROGRAM DATA>  
0         | Pattern A  
1         | Pattern B  
| <range>   | <DECIMAL NUMERIC PROGRAM DATA>  
0         | Specifies a page.  
1         | Specifies all data.  
(One page is defined as 128 bits.)  
| <page>    | <DECIMAL NUMERIC PROGRAM DATA>  
1 to 524288 | 1 to 524,288 (Max.) pages, in 1-page steps  
Max = Pattern Length / 128 (rounding up fractions)  
Specify "0" when <range> is set to 1.  
| <data>    | <NUMERIC PROGRAM DATA>  
0         | Fills the specified range with 0s.  
1         | Fills the specified range with 1s.  

### Function
Sets 0s or 1s for the specified page or all data in the program during Mixed Alternate pattern generation.

### Example
To set 1s for all the data of the block 1 pattern B:

> XDF 1,1,1,0,1

### Compatibility
Incompatible with existing models.
8.3.2.8 Sequence pattern setting commands

Figure 8.3.2.8-1 Sequence pattern setting field

Figure 8.3.2.8-2 Sequence Pattern Setting screen
### Table 8.3.2.8-1 Sequence pattern setting commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1]</td>
<td>Pair ED</td>
<td>SPR</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SPR?</td>
</tr>
<tr>
<td>[3]</td>
<td>Transmit</td>
<td>SRE</td>
</tr>
<tr>
<td>[4]</td>
<td>Add</td>
<td>SAD</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SAD?</td>
</tr>
<tr>
<td>[5]</td>
<td>Copy</td>
<td>SCO</td>
</tr>
<tr>
<td>[6]</td>
<td>Cut</td>
<td>SCU</td>
</tr>
<tr>
<td>[7]</td>
<td>Paste</td>
<td>SPA</td>
</tr>
<tr>
<td>[8]</td>
<td>Clear All</td>
<td>SCD</td>
</tr>
<tr>
<td>[9]</td>
<td>Loop times</td>
<td>SLP</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SLP?</td>
</tr>
<tr>
<td>[10]</td>
<td>Loop Completion</td>
<td>SCL</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SCL?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SCE?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SCM?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SCA?</td>
</tr>
<tr>
<td>[14]</td>
<td>B Pattern Match</td>
<td>SCB</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SCB?</td>
</tr>
</tbody>
</table>

**Figure 8.3.2.8-3** Sequence pattern setting (Pattern Editor screen)
Table 8.3.2.8-1  Sequence pattern setting commands (Cont’d)

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>[15]</td>
<td>Data Length</td>
<td>SLG</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SLG?</td>
</tr>
<tr>
<td>[16]</td>
<td>No label (Pattern setting)</td>
<td>SDW</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SDW?</td>
</tr>
<tr>
<td>[17]</td>
<td>No label (Pattern setting)</td>
<td>SBW</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SBW?</td>
</tr>
<tr>
<td>[18]</td>
<td>No label (Data inversion)</td>
<td>SDA</td>
</tr>
<tr>
<td>[19]</td>
<td>No label (Data inversion)</td>
<td>SDD</td>
</tr>
<tr>
<td>[20]</td>
<td>No label (All 0 or All 1)</td>
<td>SDF</td>
</tr>
</tbody>
</table>

**SPR <numeric>**

Parameter: <numeric> = <DECIMAL NUMERIC PROGRAM DATA>

0 to 6 slots

*The number of slots is 0 to 6 for the MP1800A, and 0 to 4 for the MT1810A.*

Function: Sets the slot of ED that is paired during Sequence pattern generation.

Example: To set the slot of ED to be paired to Slot 2:

```plaintext
> SPR 2
```

Compatibility: Incompatible with existing models.

**SPR?**

Response: <numeric> = <NR1 NUMERIC RESPONSE DATA> (FIX2)

Function: Queries the slot of ED that is paired during Sequence pattern generation.

Example: To query the slot of ED paired to Slot 2:

```plaintext
> SPR?
< SPR  2
```

Compatibility: Incompatible with existing models.

**SMA**

Function: Generates a Sequence pattern that is set manually.

Example: To generate a Sequence pattern manually:

```plaintext
> SMA
```

Compatibility: Incompatible with existing models.

**SRE**

Function: Restarts the Sequence pattern generation.

Example: To restart the Sequence pattern generation:

```plaintext
> SRE
```

Compatibility: Incompatible with existing models.
### SAD <block>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;block&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 to 128</td>
<td>1 to 128 blocks, in 1-block steps</td>
</tr>
</tbody>
</table>

**Function**
Set the number of blocks that are edited during Sequence pattern generation.

**Example**
To set the number of blocks that are edited to 10:
> SAD 10

**Compatibility**
Incompatible with existing models.

### SAD?

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;block&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt; (FIX3)</th>
</tr>
</thead>
</table>

**Function**
Queries the number of blocks that are edited during Sequence pattern generation.

**Example**
To query the number of blocks that are edited:
> SAD?
< SAD 1
< SAD 128

**Compatibility**
Incompatible with existing models.

### SCO <block>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;block&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 to 128</td>
<td>1 to 128 blocks, in 1-block steps</td>
</tr>
</tbody>
</table>

**Function**
Copies the setting conditions in the specified block during Sequence pattern generation.

**Example**
To copy the setting conditions in block 12:
> SCO 12

**Compatibility**
Incompatible with existing models.

### SCU <block>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;block&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 to 128</td>
<td>1 to 128 blocks, in 1-block steps</td>
</tr>
</tbody>
</table>

**Function**
Cuts (copies and removes) the settings in the specified block during Sequence pattern generation.

**Example**
To cut the settings in block 15:
> SCU 15

**Compatibility**
Incompatible with existing models.
### SPA <block>

**Parameter**

\[ \text{<block> = DECIMAL NUMERIC PROGRAM DATA} \]

1 to 128 blocks, in 1-block steps

**Function**

Pastes the settings to the specified block during Sequence pattern generation.

A block is added to the specified block number.

**Example**

To paste the setting conditions to block 20:

\>` SPA 20

**Compatibility**

Incompatible with existing models.

### SCD

**Function**

Clears all the setting conditions during Sequence pattern generation.

**Example**

\>` SCD

**Compatibility**

Incompatible with existing models.

### SLP <block>, <type> [,<numeric>]}

**Parameter**

\[ \text{<block> = DECIMAL NUMERIC PROGRAM DATA} \]

1 to 128 blocks, in 1-block steps

\[ \text{<type> = DECIMAL NUMERIC PROGRAM DATA} \]

0 Specifies the number of repetition times.

1 Does not specify the number of repetition times.

\[ \text{<numeric> = DECIMAL NUMERIC PROGRAM DATA} \]

1 to 1024 loops, in 1-loop steps

**Note:**

<numeric> can be omitted when <type> is set to 1. Also, the setting in <numeric> is ignored when <type> is set to 1.

**Function**

Sets the number of loops for the specified block during Sequence pattern generation.

**Example**

To set the number of loops for block 1 to 150:

\>` SLP 1,0,150

**Compatibility**

Incompatible with existing models.
8.3 PPG Commands

SLP? <block>

Parameter

- `<block>` = <DECIMAL NUMERIC PROGRAM DATA>
  1 to 128 1 to 128 blocks, in 1-block steps

Response

- `<type>` = <NR1 NUMERIC RESPONSE DATA> (FIX1)
  `[<numeric>]` = <NR1 NUMERIC RESPONSE DATA> (FIX4)

  **Note:**
  When `<type>` is set to 1, `<numeric>` is omitted.

Function

Queries the number of loops for the specified block during Sequence pattern generation.

Example

To query the number of loops for block 1:

- > SLP? 1
- < SLP 1
- < SLP 0, 1
- < SLP 0, 1024

Compatibility

Incompatible with existing models.

SCL <block>,<loop>,[<jump>]

Parameter

- `<block>` = <DECIMAL NUMERIC PROGRAM DATA>
  1 to 128 1 to 128 blocks, in 1-block steps
- `<loop>` = <DECIMAL NUMERIC PROGRAM DATA>
  0  STOP
  1  JUMP
  2  NEXT
- `[<jump>]` = <DECIMAL NUMERIC PROGRAM DATA>
  1 to 128 1 to 128 blocks, in 1-block steps

Function

Sets the loop completion condition of each block during Sequence pattern generation.

Example

To set NEXT for the loop completion condition of block 3:

- > SCL 3,2

Compatibility

Incompatible with existing models.
Chapter 8  Native Commands

SCL? <block>

Parameter

<block> = <DECIMAL NUMERIC PROGRAM DATA>
1 to 128  1 to 128 blocks, in 1-block steps

Response

<loop> = <NR1 NUMERIC RESPONSE DATA> (FIX1)
[jump>] = <NR1 NUMERIC RESPONSE DATA> (FIX3)
1 to 128  1 to 128 blocks

When selecting JUMP, specify the jump destination block number n from
1 to 128. Can be omitted when JUMP is not selected.

Function

Queries the loop completion condition of each block during Sequence
pattern generation.

Example

To query the loop completion condition of block 3:

> SCL? 3
< SCL 2
< SCL 1, 128
< SCL 1, 1

Compatibility

Incompatible with existing models.

SCE <block>, <ext>, [jump]

Parameter

<block> = <DECIMAL NUMERIC PROGRAM DATA>
1 to 128  1 to 128 blocks, in 1-block steps
<ext> = <CHARACTER PROGRAM DATA>
0 STOP
1 JUMP
2 NEXT
3 NONE

[jump>] = <DECIMAL NUMERIC PROGRAM DATA>
1 to 128  1 to 128 blocks, in 1-block steps

When selecting JUMP, specify the jump destination block number n from
1 to 128. Can be omitted when JUMP is not selected.

Function

Sets the external trigger condition of each block during Sequence pattern
generation.

Example

To set STOP for the external trigger condition of block 5:

> SCE 5, 0

Compatibility

Incompatible with existing models.
8.3 PPG Commands

SCE? <block>

Parameter

<block> = <DECIMAL NUMERIC PROGRAM DATA>
1 to 128 1 to 128 blocks, in 1-block steps

Response

<ext> = <NR1 NUMERIC RESPONSE DATA> (FIX1)
0, 1, 2, 3
[jump>] = <NR1 NUMERIC RESPONSE DATA> (FIX3)
1 to 128 1 to 128 blocks
When selecting JUMP, specify the jump destination block number n from 1 to 128. Can be omitted when JUMP is not selected.

Function
Queries the external trigger condition of each block during Sequence pattern generation.

Example
To query the external trigger condition of block 5:
> SCE? 5
< SCE 0
< SCE 1, 1
< SCE 1,128

Compatibility
Incompatible with existing models.

SCM <block>,<man>[,<jump>]

Parameter

<block> = <DECIMAL NUMERIC PROGRAM DATA>
1 to 128 1 to 128 blocks, in 1-block steps
<man> = <DECIMAL NUMERIC PROGRAM DATA>
0 STOP
1 JUMP
2 NEXT
3 NONE
[jump>] = <DECIMAL NUMERIC PROGRAM DATA>
1 to 128 1 to 128 blocks, in 1-block steps
When selecting JUMP, specify the jump destination block number n from 1 to 128. Can be omitted when JUMP is not selected.

Function
Sets the manual trigger condition of each block during Sequence pattern generation.

Example
To set the manual trigger condition of block 2 to "Jump to block 5".
> SCM 2,1,5

Compatibility
Incompatible with existing models.
Chapter 8  Native Commands

SCM? <block>

Parameter

<block> = <DECIMAL NUMERIC PROGRAM DATA>
1 to 128  1 to 128 blocks, in 1-block steps

Response

<man> = <NR1 NUMERIC RESPONSE DATA> (FIX1)
0, 1, 2, 3
<jump> = <NR1 NUMERIC RESPONSE DATA> (FIX3)
1 to 128  1 to 128 blocks

When selecting JUMP, specify the jump destination block number n from
1 to 128. Can be omitted when JUMP is not selected.

Function
Queries the manual trigger condition of each block during Sequence
pattern generation.

Example
To query the manual trigger condition of block 2:
> SCM? 2
< SCM 2
< SCM 1, 5
< SCM 1,128

Compatibility
Incompatible with existing models.

SCA <block>,<a>[,<jump>]

Parameter

<block> = <DECIMAL NUMERIC PROGRAM DATA>
1 to 128  1 to 128 blocks, in 1-block steps
<a> = <DECIMAL NUMERIC PROGRAM DATA>
0 STOP
1 JUMP
2 NEXT
3 NONE
<jump> = <DECIMAL NUMERIC PROGRAM DATA>
1 to 128  1 to 128 blocks, in 1-block steps

When selecting JUMP, specify the jump destination block number n from
1 to 128. Can be omitted when JUMP is not selected.

Function
Sets the pattern A match condition of each block during Sequence
pattern generation.

Example
To set NONE for the pattern A match condition of block 3:
> SCA 3,3

Compatibility
Incompatible with existing models.
### SCA? <block>

**Parameter**

- `<block>` = `<DECIMAL NUMERIC PROGRAM DATA>`
- 1 to 128 blocks, in 1-block steps

**Response**

- `<a>` = `<NR1 NUMERIC RESPONSE DATA>` (FIX1)
- 0, 1, 2, 3
- `<jump>` = `<NR1 NUMERIC RESPONSE DATA>` (FIX3)
- 1 to 128 blocks, in 1-block steps

When selecting JUMP, specify the jump destination block number n from 1 to 128. Can be omitted when JUMP is not selected.

**Function**

Queries the pattern A match condition of each block during Sequence pattern generation.

**Example**

To query the pattern A match condition of block 3:

```
> SCA? 3
< SCA 3
< SCA 1, 1
< SCA 1, 128
```

**Compatibility**

Incompatible with existing models.

### SCB <block>,<b>, [,<jump>]

**Parameter**

- `<block>` = `<DECIMAL NUMERIC PROGRAM DATA>`
- 1 to 128 blocks, in 1-block steps
- `<b>` = `<DECIMAL NUMERIC PROGRAM DATA>`
- 0 STOP
- 1 JUMP
- 2 NEXT
- 3 NONE
- `<jump>` = `<DECIMAL NUMERIC PROGRAM DATA>`
- 1 to 128 blocks, in 1-block steps

When selecting JUMP, specify the jump destination block number n from 1 to 128. Can be omitted when JUMP is not selected.

**Function**

Sets the pattern B match condition of each block during Sequence pattern generation.

**Example**

To set NEXT for the pattern B match condition of block 7:

```
> SCB 7,2
```

**Compatibility**

Incompatible with existing models.
### SCB? <block>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;block&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt; 1 to 128 1 to 128 blocks, in 1-block steps</th>
</tr>
</thead>
<tbody>
<tr>
<td>Response</td>
<td>&lt;b&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt; (FIX1) 0, 1, 2, 3 1 to 128 1 to 128 blocks</td>
</tr>
<tr>
<td></td>
<td>[&lt;jump&gt;] = &lt;NR1 NUMERIC RESPONSE DATA&gt; (FIX3) 1 to 128 1 to 128 blocks</td>
</tr>
<tr>
<td></td>
<td>When selecting JUMP, specify the jump destination block number n from 1 to 128. Can be omitted when JUMP is not selected.</td>
</tr>
<tr>
<td>Function</td>
<td>Queries the pattern B match condition of each block during Sequence pattern generation.</td>
</tr>
<tr>
<td>Example</td>
<td>To query the pattern B match condition of block 7:</td>
</tr>
<tr>
<td></td>
<td>&gt; SCB? 7</td>
</tr>
<tr>
<td></td>
<td>&lt; SCB 2</td>
</tr>
<tr>
<td></td>
<td>&lt; SCB 1, 10</td>
</tr>
<tr>
<td></td>
<td>&lt; SCB 1,128</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Incompatible with existing models.</td>
</tr>
</tbody>
</table>

### SLG <block>,<numeric>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;block&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt; 1 to 128 1 to 128 blocks, in 1-block steps &lt;numeric&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt; 8192 to 1048576 8,192 to 1,048,576 bits, in 128-bit steps</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function</td>
<td>Sets the pattern length of the specified block during Sequence pattern generation.</td>
</tr>
<tr>
<td>Example</td>
<td>To set the pattern length of block 3 to 8192 bits:</td>
</tr>
<tr>
<td></td>
<td>&gt; SLG 3,8192</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Incompatible with existing models.</td>
</tr>
</tbody>
</table>

### SLG? <block>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;block&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt; 1 to 128 1 to 128 blocks, in 1-block steps</th>
</tr>
</thead>
<tbody>
<tr>
<td>Response</td>
<td>&lt;numeric&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt; (FIX7) 8192 to 1048576 8,192 to 1,048,576 bits, in 128-bit steps</td>
</tr>
<tr>
<td>Function</td>
<td>Queries the pattern length of the specified block during Sequence pattern generation.</td>
</tr>
<tr>
<td>Example</td>
<td>To query the pattern length of block 3:</td>
</tr>
<tr>
<td></td>
<td>&gt; SLG? 3</td>
</tr>
<tr>
<td></td>
<td>&lt; SLG 8192</td>
</tr>
<tr>
<td></td>
<td>&lt; SLG 1048576</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Incompatible with existing models.</td>
</tr>
</tbody>
</table>
8.3 PPG Commands

**SDW <block>,<start>,<end>,<data>**

**Parameter**

<block> = <DECIMAL NUMERIC PROGRAM DATA>
1 to 128  
1 to 128 blocks, in 1-block steps

<start> = <NON-DECIMAL NUMERIC PROGRAM DATA>
#H0 to #HFFFFF  
0 to FFFFF bits, in 1-bit steps
(Specify in hexadecimal)

<end> = <NON-DECIMAL NUMERIC PROGRAM DATA>
#H0 to #HFFFFF  
0 to FFFFF bits, in 1-bit steps
(Specify in hexadecimal)

**Note:**
The maximum setting value is the pattern length.

<data> = <NON-DECIMAL NUMERIC PROGRAM DATA>
#H**...*  
Specify pattern data in hexadecimal format.
1 to 400 characters (pattern data of 400 \( \times \) 4 bits)
Specify a character string consisting of 0 to 9 and A to F.
If the last character string of the pattern data is short of a hexadecimal unit (4 bits), add 0(s) up to the last bit to make a hexadecimal string.

#B**...*  
Specify pattern data in binary format.
1 to 400 characters (pattern data of 400 bits)
Specify a character string consisting of 0 and 1.

**Function**

Sets the pattern data from the <start> to <end> addresses during Sequence pattern generation.
The set data overwrites the specified range.

**Example**

To set the addresses 0 to FFFF of the block 1 data to 5A:

> SDW 1,#H0,#HFFFFF,#H5A

**Compatibility**

Incompatible with existing models.
SDW <block>,<start>

Parameter

<block> = <DECIMAL NUMERIC PROGRAM DATA>
1 to 128 1 to 128 blocks, in 1-block steps

<start> = <NON-DECIMAL NUMERIC PROGRAM DATA>
#H0 to #HFFFFFF 0 to FFFFF Fbits, in 1-bit steps
(Specify in hexadecimal)

Note:
The maximum setting value is the pattern length.

Response

<data> = <HEXADECIMAL NUMERIC RESPONSE DATA>
#H**...* Returns the pattern data in hexadecimal format.
Within 400 characters (pattern data of 400 × 4 bits)

Function

Queries the pattern data of 400 × 4 bits from the <start> address during Sequence pattern generation.

Example

To query the pattern data of 400 × 4 bits from address 0:

> SDW? 1,#H0
< SDW #H0123456789ABCDEF… (1 to 400 characters in Hex.)
< SDW #H0

Compatibility

Incompatible with existing models.

SBW <block>,<start>,<end>,<bdata>

Parameter

<block> = <DECIMAL NUMERIC PROGRAM DATA>
1 to 128 1 to 128 blocks, in 1-block steps

<start> = <NON-DECIMAL NUMERIC PROGRAM DATA>
#H0 to #HFFFFFF 0 to FFFFF Fbits, in 1-bit steps
(Specify in hexadecimal)

<end> = <NON-DECIMAL NUMERIC PROGRAM DATA>
#H0 to #HFFFFFF 0 to FFFFF Fbits, in 1-bit steps
(Specify in hexadecimal)

Note:
The maximum setting value is the pattern length.

<bdata> = <ARBITRARY BLOCK PROGRAM DATA>
#XYYY<binary> X: Number of YYY digits
YYY: Number of bytes of <binary>
1 to 131,072 bytes
<binary>: Binary data up to 131,072 bytes

Function

Sets binary data of the pattern data from <start> to <end> addresses during Sequence pattern generation.
The set data overwrites the specified range.

Example

To set the addresses 0 to FFFF of the block 1 data to 44:

> SBW 1,#H0,#HFFFFFF,#11D

Compatibility

Incompatible with existing models.
8.3 PPG Commands

SBW? <block>,<start>[,<size>]

Parameter

- `<block>` = `<DECIMAL NUMERIC PROGRAM DATA>`
  1 to 128 1 to 128 blocks, in 1-block steps
- `<start>` = `<NON-DECIMAL NUMERIC PROGRAM DATA>`
  #H0 to #HFFFFF 0 to FFFFFF bits, in 1-bit steps
  (Specify in hexadecimal)
- `[<size>]` = `<DECIMAL NUMERIC PROGRAM DATA>`
  1 to 131072 1 to 131,072, in 1-byte steps

Note:

The maximum setting value is the pattern length.

Response

- `<bdata>` = `<DEFINITE LENGTH ARBITRARY BLOCK RESPONSE DATA>`
  Binary data of the setting pattern is queried when `[<size>]` is omitted.

Function

Queries binary data of bytes specified by `<size>` from `<start>` address of the pattern data during Sequence pattern generation.

Example

To query the pattern data of 2 bytes in block 1 from address 0:

> SBW? 1,#H0,2
< SBW #12DD

To query the pattern data of the setting pattern in block 1 from address 0:

> SBW? 1,#H0
< SBW #516000AA… (Binary data of 16,000 bytes)

Compatibility

Incompatible with existing models.

SDA <block>,<start>,<end>

Parameter

- `<block>` = `<DECIMAL NUMERIC PROGRAM DATA>`
  1 to 128 1 to 128 blocks, in 1-block steps
- `<start>` = `<NON-DECIMAL NUMERIC PROGRAM DATA>`
  #H0 to #HFFFFF 0 to FFFFFF bits, in 1-bit steps
  (Specify in hexadecimal)
- `<end>` = `<NON-DECIMAL NUMERIC PROGRAM DATA>`
  #H0 to #HFFFFF 0 to FFFFFF bits, in 1-bit steps
  (Specify in hexadecimal)

Note:

The maximum setting value is the pattern length.

Function

Inverts the data in the program during Sequence pattern generation.

Specify the inversion range by the `<start>` and `<end>` addresses.

Example

To invert addresses 0 to FFF of the block 3 data:

> SDA 3,#H0,#HFFF

Compatibility

Incompatible with existing models.
Chapter 8  Native Commands

SDD <block>,<start>,<delta>

Parameter

- **<block>** = <DECIMAL NUMERIC PROGRAM DATA>
  1 to 128 1 to 128 blocks, in 1-bit steps
- **<start>** = <NON-DECIMAL NUMERIC PROGRAM DATA>
  #H0 to #HFFFFF 0 to FFFFF bits, in 1-bit steps
  (Specify in hexadecimal)
- **<delta>** = <DECIMAL NUMERIC PROGRAM DATA>
  1 to 1048576 1 to 1,048,576, in 1-bit steps

**Note:**
The maximum setting value is the pattern length.

Function

Inverts the data in the program during Sequence pattern generation.
Specify the inversion range by the number of bits from the <start> address (relative specification).

Example

To invert 128 bits of the block 5 data from address 0:
> SDD 5,#H0,128

Compatibility

Incompatible with existing models.

SDF <block>,<range>,<page>,<data>

Parameter

- **<block>** = <DECIMAL NUMERIC PROGRAM DATA>
  1 to 128 1 to 128 blocks, in 1-bit steps
- **<range>** = <DECIMAL NUMERIC PROGRAM DATA>
  0 Specifies a page.
  (One page is defined as 128 bits.)
  1 Specifies all data.
- **<page>** = <DECIMAL NUMERIC PROGRAM DATA>
  1 to 8192 1 to 8,192 (Max.) pages, in 1-page steps
  Max = Pattern Length / 128 (rounding up fractions)
  Specify "0" when <range> is set to 1.
- **<data>** = <DECIMAL NUMERIC PROGRAM DATA>
  0 Fills the specified range with 0s.
  1 Fills the specified range with 1s.

Function

Sets 0s or 1s for the specified page or all data in the program during Sequence pattern generation.

Example

To set 1s for all the data of the block 8:
> SDF 8,1,0,1

Compatibility

Incompatible with existing models.
8.3.3 Commands related to Error Addition tab

Figure 8.3.3-1 Error Addition setting field (For PRBS, Zero-Substitution, Data patterns)

Figure 8.3.3-2 Error Addition setting field (For Alternate pattern)
Figure 8.3.3-3  Error Addition setting field (For Mixed pattern)
8.3 PPG Commands

Figure 8.3.3-4  Error Addition setting field (For Sequence pattern)

Table 8.3.3-1  Error Addition tab setting commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1]</td>
<td>Error Addition ON, OFF</td>
<td>EAD</td>
</tr>
<tr>
<td></td>
<td></td>
<td>EAD?</td>
</tr>
<tr>
<td>[2]</td>
<td>Source</td>
<td>EEI</td>
</tr>
<tr>
<td></td>
<td></td>
<td>EEI?</td>
</tr>
<tr>
<td>[3]</td>
<td>Route</td>
<td>ESL</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ESL?</td>
</tr>
<tr>
<td>[4]</td>
<td>Error Addition Route setting</td>
<td>ECH</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ECH?</td>
</tr>
<tr>
<td>[5]</td>
<td>Rate</td>
<td>ERT</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ERT?</td>
</tr>
<tr>
<td>[6]</td>
<td>Variation</td>
<td>EAV</td>
</tr>
<tr>
<td></td>
<td></td>
<td>EAV?</td>
</tr>
<tr>
<td>[7]</td>
<td>Single</td>
<td>ESI</td>
</tr>
<tr>
<td>[8]</td>
<td>Area A/B (Error addition area setting)</td>
<td>EAR</td>
</tr>
<tr>
<td></td>
<td></td>
<td>EAR?</td>
</tr>
<tr>
<td>[9]</td>
<td>Set All/Reset All</td>
<td>EAA</td>
</tr>
</tbody>
</table>
**Chapter 8  Native Commands**

**EAD <numeric>**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;numeric&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Bit error addition OFF</td>
</tr>
<tr>
<td>1</td>
<td>Bit error addition ON</td>
</tr>
<tr>
<td>7</td>
<td>Single error addition</td>
</tr>
</tbody>
</table>

**Function**
Sets whether to add a bit error to the test pattern.

**Example**
To set bit error addition ON:
> EAD 1

**Compatibility**
Partially compatible with the MP1632C, MP1761C and MP1775A (Only 0: OFF, 7: Single).

**EAD?**

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;numeric&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt; (FIX1)</th>
</tr>
</thead>
</table>

**Function**
Queries the bit error addition ON or OFF state to the test pattern.

**Example**
> EAD?
< EAD 1

**Compatibility**
Partially compatible with the MP1632C, MP1761C and MP1775A.

**EEI <source>**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;source&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Internal: Internal signal</td>
</tr>
<tr>
<td>1</td>
<td>External Trigger: Rising of external signal trigger</td>
</tr>
<tr>
<td>2</td>
<td>External Disable: Disables external signal</td>
</tr>
</tbody>
</table>

**Function**
Sets the reference signal source for bit error addition to the test pattern.

**Example**
To set the reference signal source for bit error addition to Internal:
> EEI 0

**Compatibility**
Compatible with the MP1761C.

**EEI?**

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;source&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt; (FIX1)</th>
</tr>
</thead>
</table>

**Function**
Queries the reference signal source for bit error addition to the test pattern.

**Example**
> EEI?
< EEI 0

**Compatibility**
Compatible with the MP1761C.
### ESL <select>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Function</th>
<th>Example</th>
<th>Compatibility</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;select&gt;</td>
<td>&lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
<td>0</td>
<td>Scan: Changes the route for which 32 Demux of 1/1 signal is performed every time when an error is added.</td>
<td>Incompatible with existing models.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1</td>
<td>Select: Adds an error to the specified route.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Response</td>
<td>&lt;select&gt;</td>
<td></td>
<td>&lt;NR1 NUMERIC RESPONSE DATA&gt; (FIX1)</td>
<td></td>
</tr>
<tr>
<td>Function</td>
<td>Queries the route addition method for bit error addition.</td>
<td></td>
<td>&lt; ESL 0</td>
<td></td>
</tr>
<tr>
<td>Example</td>
<td>To set the route addition method to Scan:</td>
<td>&gt; ESL?</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt; ESL 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Compatibility</td>
<td>Incompatible with existing models.</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### ECH <numeric>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Function</th>
<th>Example</th>
<th>Compatibility</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;numeric&gt;</td>
<td>&lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
<td>1 to 32 1 to 32, in single steps</td>
<td>Sets the route to add a bit error.</td>
<td>Compatible with the MP1632C (Routes 1 to 8 only). Compatible with the MP1761C and MP1775A.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>To set Route 2 as the route to add a bit error:</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>&gt; ECH 2</td>
<td></td>
</tr>
<tr>
<td>Compatibility</td>
<td>Compatible with the MP1632C (Routes 1 to 8 only). Compatible with the MP1761C and MP1775A.</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### ECH?

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;numeric&gt;</th>
<th>Function</th>
<th>Example</th>
<th>Compatibility</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;numeric&gt;</td>
<td>&lt;NR1 NUMERIC RESPONSE DATA&gt; (FIX2)</td>
<td>Queries the route to add a bit error.</td>
<td>&gt; ECH?</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>&lt; ECH 1</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>&lt; ECH 32</td>
<td></td>
</tr>
<tr>
<td>Compatibility</td>
<td>Compatible with the MP1632C (Routes 1 to 8 only). Compatible with the MP1761C and MP1775A.</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Chapter 8  Native Commands

ERT <numeric>,<rate>

**Parameter**
Set a value in the format of "xE−n".

x: <numeric>, n: <rate>

- **<numeric>** =  <DECIMAL NUMERIC PROGRAM DATA>
  1 to 9  1 to 9, in single step
- **<rate>** =  <DECIMAL NUMERIC PROGRAM DATA>
  2 to 12  2 to 12, in single step

**Note:**
3 to 12 in the case of Combination.

**Function**
Sets the rate of bit errors to be added.

**Example**
To set the rate of bit errors to be added to 5E−9:

> ERT 5,9

**Compatibility**
Incompatible with existing models.

ERT?

**Response**

- **<numeric>** =  <NR1 NUMERIC RESPONSE DATA> (FIX1)
- **<rate>** =  <NR1 NUMERIC RESPONSE DATA> (FIX2)

**Function**
Queries the rate of bit errors to be added.

**Example**
To set the rate of bit errors to be added:

> ERT?

< ERT 5,9
< ERT 9,12

**Compatibility**
Incompatible with existing models.

EAV <var>

**Parameter**

- **<var>** =  <DECIMAL NUMERIC PROGRAM DATA>
  0  Inserts Repeat errors continuously.
  1  Inserts Single error by button operation (one-shot).

**Function**
Sets the bit error addition method for the test pattern (when Internal is selected).

**Example**
To set the bit error addition method to Repeat:

> EAV 0

**Compatibility**
Incompatible with existing models.
8.3 PPG Commands

**EAV?**

**Response**  
<var> = <NR1 NUMERIC RESPONSE DATRA> (FIX1)

**Function**  
Queries the bit error addition method (when Internal is selected) to the test pattern.

**Example**  
< EAV ?  
> EAV 0

**Compatibility**  
Incompatible with existing models.

**ESI**

**Function**  
Adds a single error to the test pattern.

**Example**  
> ESI

**Compatibility**  
Incompatible with existing models.

**EAR <content>,<numeric>,<data>,<prbs>**

**Parameter**  
<content> = <DECIMAL NUMERIC PROGRAM DATA>

- 0: Pattern A
- 1: Pattern B
- 2: Patterns A and B
- 3: None (Always set for Mixed Data and Sequence patterns)

<numeric> = <DECIMAL NUMERIC PROGRAM DATA>

- 1 to 16: 1 to 16 rows, in 1-row steps (For Mixed Data and Mixed Alternate patterns)
- 1 to 128: 1 to 128 blocks, in 1-block steps (For Sequence pattern)
- 0: None (For Alternate pattern)

<data> = <DECIMAL NUMERIC PROGRAM DATA>

- 0: Bit error addition OFF (Always set for Alternate pattern)
- 1: Bit error addition ON

<prbs> = <DECIMAL NUMERIC PROGRAM DATA>

- 0: Bit error addition OFF (Always set for Alternate pattern)
- 1: Bit error addition ON

**Function**  
Sets an area to add a bit error for Alternate, Mixed Data, Mixed Alternate and Sequence patterns.

**Example**  
To set an area to add a bit error to "Pattern A, 1 Row, Data ON, PRBS ON" for Mixed Alternate pattern:
> EAR 0,1,1,1

**Compatibility**  
Incompatible with existing models.
### EAR? <content>,<numeric>

**Parameter**
- `<content>` = `<DECIMAL NUMERIC PROGRAM DATA>`
- `<numeric>` = `<DECIMAL NUMERIC PROGRAM DATA>`

**Response**
- `<data>,<prbs>` = `<NR1 NUMERIC RESPONSE DATA>` (FIX1)
  
  **Note:**
  - `<prbs>` is returned only for Mixed Data and Mixed Alternate patterns.

**Function**
Queries an area to add a bit error.

**Example**
To query the error addition ON/OFF state of Pattern A, 1 Row for Mixed Alternate pattern:

```plaintext
> EAR? 0,1
< EAR 1,1
```

To query the error addition ON/OFF state of Pattern A, for Alternate pattern:

```plaintext
> EAR? 1,0
< EAR 1
```

**Compatibility**
Incompatible with existing models.

---

### EAA <numeric>

**Parameter**
- `<numeric>` = `<NUMERIC PROGRAM DATA>`
  - 0 All Reset
  - 1 All Set

**Function**
Selects (set) or unselects (reset) all areas to add a bit error for Mixed and Sequence patterns.

**Example**
To select all areas to add a bit error:

```plaintext
> EAA 1
```

**Compatibility**
Incompatible with existing models.
8.3.4 Commands related to Misc tab

![Figure 8.3.4-1 Misc tab](image)

Figure 8.3.4-1 Misc tab
8.3.4.1 Pattern Generation setting commands

Figure 8.3.4.1-1 Pattern Generation setting field (for Repeat signal generation)

Figure 8.3.4.1-2 Pattern Generation setting field (for Repeat signal generation, Alternate pattern)
8.3 PPG Commands

Native Commands

Figure 8.3.4.1-3 Pattern Generation setting field (for Burst signal generation)

Table 8.3.4.1-1 Pattern Generation setting commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1]</td>
<td>Pattern Sequence</td>
<td>PTO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PTO?</td>
</tr>
<tr>
<td>[2]</td>
<td>Data Sequence</td>
<td>BSQ</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BSQ?</td>
</tr>
<tr>
<td>[3]</td>
<td>Source</td>
<td>BRM</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BRM?</td>
</tr>
<tr>
<td>[4]</td>
<td>Pulse Width</td>
<td>RPW</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RPW?</td>
</tr>
<tr>
<td>[5]</td>
<td>Delay (For Repeat signal generation)</td>
<td>RTD</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RTD?</td>
</tr>
<tr>
<td>[6]</td>
<td>Enable Period</td>
<td>BRE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BRE?</td>
</tr>
<tr>
<td>[7]</td>
<td>Burst Cycle</td>
<td>BRC</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BRC?</td>
</tr>
<tr>
<td>[8]</td>
<td>Delay (For Burst signal generation)</td>
<td>TGD</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TGD?</td>
</tr>
<tr>
<td>[9]</td>
<td>Enable Pulse Width</td>
<td>PWD</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWD?</td>
</tr>
</tbody>
</table>
### PTO <mode>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Repeat signal</td>
<td>Burst signal</td>
<td></td>
</tr>
</tbody>
</table>

**Function**
Sets the signal generation method from Repeat (consecutive) signal or Burst signal.

**Example**
To set the signal generation method to Burst signal:
> PTO 1

**Compatibility**
Compatible with the MP1632C.

### PTO? 

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;NR1 NUMERIC RESPONSE DATA&gt; (FIX1)</th>
</tr>
</thead>
</table>

**Function**
Queries the signal generation method.

**Example**
> PTO?
< PTO 1

**Compatibility**
Compatible with the MP1632C.

### BSQ <mode>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>RESTart</th>
<th>CONSecutive</th>
<th>CONTinuous</th>
</tr>
</thead>
<tbody>
<tr>
<td>Restart from the beginning</td>
<td>Consecutive output in the Burst interval</td>
<td>Consecutive output regardless of Burst interval</td>
<td></td>
</tr>
</tbody>
</table>

**Function**
Sets the data output sequence for the Burst signal.

**Example**
To set the output sequence to Restart:
> BSQ 0

**Compatibility**
Incompatible with existing models.

### BSQ? 

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;NR1 NUMERIC RESPONSE DATA&gt; (FIX1)</th>
</tr>
</thead>
</table>

**Function**
Queries the data output sequence for the Burst signal.

**Example**
> BSQ?
< BSQ 0

**Compatibility**
Incompatible with existing models.
8.3 PPG Commands

BRM <mode>

Parameter  
<mode> = <DECIMAL NUMERIC PROGRAM DATA>
0  Internal: Generates based on internal reference signal.
1  Ext. Enable: Generates during the high level interval of external signal trigger input
2  Ext Trigger: Generates based on external signal trigger edge.

Function  
Sets the Burst signal generation sequence from internal trigger (0), external trigger (2), or external enable (1).

Example  
To set the Burst signal generation sequence to Internal:
> BRM 0

Compatibility  
Compatible with the MP1632C.

BRM?

Response  
<mode> = <NR1 NUMERIC RESPONSE DATA> (FIX1)

Function  
Queries the Burst signal generation sequence.

Example  
> BRM?
< BRM 0

Compatibility  
Compatible with the MP1632C.

RPW <numeric> [,<content>]

Parameter  
<numeric> = <DECIMAL NUMERIC PROGRAM DATA>
0 to 68719476672 0 to 68,719,476,672 bits, in 16-bit steps
The setting range is from 0 to 64 if the pattern length is 64 or less.

Note:
At 2 ch Combination, the setting range and Step are doubled.
At 4 ch Combination, the setting range and Step are quadrupled.

[<content>] = <DECIMAL NUMERIC PROGRAM DATA>
0  Pattern A
1  Pattern B

Note:
For patterns other than the Alternate pattern, <content> cannot be set and should be omitted.

Function  
Sets the pulse width of the timing signal to be output during the timing signal period.

Example  
To set the pulse width of the timing signal to 128 bits:
> RPW 128

Compatibility  
Incompatible with existing models.
### RPW? [content]

<table>
<thead>
<tr>
<th>Parameter</th>
<th>[content] = &lt;DECIMAL NUMERIC PROGRAM DATA&gt; (FIX12)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Pattern A</td>
</tr>
<tr>
<td>1</td>
<td>Pattern B</td>
</tr>
</tbody>
</table>

**Note:**
For patterns other than the Alternate pattern, `<content>` cannot be set and should be omitted.

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;numeric&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function</td>
<td>Queries the pulse width of the timing signal to be output during the timing signal period.</td>
</tr>
<tr>
<td>Example</td>
<td>&gt; RPW?</td>
</tr>
<tr>
<td></td>
<td>&lt; RPW 64</td>
</tr>
<tr>
<td></td>
<td>&lt; RPW 2281701312</td>
</tr>
</tbody>
</table>

**Compatibility:**
Incompatible with existing models.

### RTD <numeric>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;numeric&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 to 6871947672</td>
<td>0 to 68,719,476,672 bits, in 16-bit steps</td>
</tr>
</tbody>
</table>

The settable value is 0 if the pattern length is 79 bits or less.

**Note:**
At 2 ch Combination, the setting range and Step are doubled.
At 4 ch Combination, the setting range and Step are quadrupled.

<table>
<thead>
<tr>
<th>Function</th>
<th>Sets the delay value for the timing signal pulse to be output during the timing signal period.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example</td>
<td>To set the timing signal pulse delay to 256 bits:</td>
</tr>
<tr>
<td></td>
<td>&gt; RTD 256</td>
</tr>
</tbody>
</table>

**Compatibility:**
Incompatible with existing models.

### RTD?

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;numeric&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt; (FIX12)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function</td>
<td>Queries the delay value for the timing signal pulse to be output during the timing signal period.</td>
</tr>
<tr>
<td>Example</td>
<td>&gt; RTD?</td>
</tr>
<tr>
<td></td>
<td>&lt; RTD 16</td>
</tr>
<tr>
<td></td>
<td>&lt; RTD 2281701312</td>
</tr>
</tbody>
</table>

**Compatibility:**
Incompatible with existing models.
**BRE <numeric>**

**Parameter**

<numeric> = <DECIMAL NUMERIC PROGRAM DATA>

640 to 2147483648  640 to 2,147,483,648 bits, in 128-bits steps

**Note:**

In the case of Internal, a Disable period of at least 512 bits is required between Burst Cycle and Enable Period.

The setting range and Step at Combination are doubled at 2 ch Combination and quadrupled at 4 ch Combination.

**Function**

Sets the data signal generation interval for Burst signal generation.

**Example**

To set the data signal generation interval to 12,800 bits:

> BRE 12800

**Compatibility**

Incompatible with the MP1632C parameters.

---

**BRE?**

**Response**

<numeric> = <NR1 NUMERIC RESPONSE DATA> (FIX11)

**Function**

Queries the data signal generation interval for Burst signal generation.

**Example**

> BRE?

< BRE 12800

< BRE 8589934080

**Compatibility**

Incompatible with the MP1632C parameters.

---

**BRC <numeric>**

**Parameter**

<numeric> = <DECIMAL NUMERIC PROGRAM DATA>

1280 to 2147483648  1,280 to 2,147,483,648 bits, in 128-bit steps

**Note:**

At 2 ch Combination, the setting range and Step are doubled.

At 4 ch Combination, the setting range and Step are quadrupled.

**Function**

Sets an addition value of the data signal generation interval and the data stop interval for Burst signal generation (Burst generation cycle).

**Example**

To set the Burst generation cycle to 25,600 bits:

> BRC 25600

**Compatibility**

Incompatible with the MP1632C parameters.
Chapter 8  Native Commands

BRC?

Response  
<numeric> = <NR1 NUMERIC RESPONSE DATA> (FIX11)

Function  
Queries the addition value of the data signal generation interval and the data stop interval for Burst signal generation (Burst generation cycle).

Example  
> BRC?
< BRC 1280
< BRC 858993592

Compatibility  
Incompatible with the MP1632C parameters.

TGD <port>,<numeric>

Parameter  
<port> = <CHARACTER PROGRAM DATA>
0 Burst Output
1 Burst Output2
Valid only when AUX Output is set to Burst Output2.
<numeric> = <DECIMAL NUMERIC PROGRAM DATA>
0 to 2147483648 0 to 2,147,483,648 bits, in 16-bit steps

Note:
The maximum setting is the pattern length.
At 2 ch Combination, the setting range is doubled.
At 4 ch Combination, the setting range is quadrupled.

Function  
Sets the Burst output trigger signal generation timing (delay) to adjust the phase on the DUT side.

Example  
To set the Burst output trigger generation timing (delay) to 128 bits:
> TGD 0,128

Compatibility  
Incompatible with existing models.

TGD? <port>

Parameter  
<port> = <CHARACTER PROGRAM DATA>
0 Burst Output
1 Burst Output2
Valid only when AUX Output is set to Burst Output2.

Response  
<numeric> = <NR1 NUMERIC RESPONSE DATA> (FIX11)

Function  
Queries the Burst output trigger signal generation timing (delay).

Example  
To query the Burst output trigger signal generation timing:
> TGD? 0
< TGD 1
< TGD 8589934336

Compatibility  
Incompatible with existing models.
8.3 PPG Commands

**PWD <port>,<numeric>**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;port&gt; = &lt;CHARACTER PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Burst Output</td>
</tr>
<tr>
<td>1</td>
<td>Burst Output2</td>
</tr>
</tbody>
</table>

Valid only when AUX Output is set to Burst Output2.

<numeric> = <DECIMAL NUMERIC PROGRAM DATA>

| 0 to 2147483648 | 0 to 2,147,483,648 bits, in 16-bit steps |

**Note:**

The maximum setting is the pattern length.

At 2 ch Combination, the setting range and Step are doubled.

At 4 ch Combination, the setting range and Step are quadrupled.

The maximum setting value is "Burst cycle – 64 bits".

**Function**

Sets the Burst output trigger signal generation interval width to adjust the phase on the DUT side.

**Example**

To set the Burst output trigger signal generation interval width to 1,024 bits:

> PWD 0,1024

**Compatibility**

Incompatible with existing models.

**PWD?<port>**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;port&gt; = &lt;CHARACTER PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Burst Output</td>
</tr>
<tr>
<td>1</td>
<td>Burst Output2</td>
</tr>
</tbody>
</table>

Valid only when AUX Output is set to Burst Output2.

| 0 to 21474835848 | 0 to 2,147,483,584 bit |

**Response**

<numeric> = <NR1 NUMERIC RESPONSE DATA> (FIX11)

**Function**

Queries the Burst output trigger signal generation interval width.

**Example**

To query the Burst output trigger signal generation interval width:

> PWD? 0

< PWD 1

< PWD 8589934336

**Compatibility**

Incompatible with existing models.
8.3.4.2 AUX Input setting commands

Table 8.3.4.2-1 AUX Input setting commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Item</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1]</td>
<td>AUX Input</td>
<td>FNC</td>
</tr>
</tbody>
</table>

FNC <function>

Parameter

- \(<function> = \langle\text{DECIMAL NUMERIC PROGRAM DATA}\rangle\>

- 0  Burst Source
- 1  Error Injection
- 2  Sequence Control
- 3  Alternate Control

Function

Sets the function of AUX Input.

Example

To set the function of AUX Input to Error Injection:

> FNC 1

Compatibility

Incompatible with existing models.

FNC?

Response

\(<function> = \langle\text{NR1 NUMERIC RESPONSE DATA}\rangle\> \ (\text{FIX1})

Function

Queries the function of AUX Input.

Example

> FNC?

< FNC 1

Compatibility

Incompatible with existing models.
### AUX Output setting commands

8.3.4.3 AUX Output setting commands

**Figure 8.3.4.3-1 AUX Output setting field**

**Figure 8.3.4.3-2 AUX Output setting field**

(For Alternate pattern)

**Figure 8.3.4.3-3 AUX Output setting field**

(For Alternate pattern)

**Figure 8.3.4.3-4 AUX Output setting field**

(For Mixed Data pattern)
Figure 8.3.4.3-5  AUX Output setting field (For Mixed Alternate pattern)

Figure 8.3.4.3-6  AUX Output setting field (For Sequence pattern)

Table 8.3.4.3-1  AUX Output setting commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1]</td>
<td>AUX Output</td>
<td>SOP</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP?</td>
</tr>
<tr>
<td>[2]</td>
<td>Position</td>
<td>PSP</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PSP?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SCN?</td>
</tr>
<tr>
<td>[4]</td>
<td>Position</td>
<td>APO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>APO?</td>
</tr>
<tr>
<td>[5]</td>
<td>Block No. (Mixed Data)</td>
<td>SBL</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SBL?</td>
</tr>
<tr>
<td>[6]</td>
<td>Row No. (Mixed Data)</td>
<td>SRW</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SRW?</td>
</tr>
<tr>
<td>[7]</td>
<td>Block No. (Mixed Alternate)</td>
<td>SML</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SML?</td>
</tr>
<tr>
<td>[8]</td>
<td>Alternate Content (Mixed Alternate)</td>
<td>SXO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SXO?</td>
</tr>
<tr>
<td>[9]</td>
<td>Row No. (Mixed Alternate)</td>
<td>SXW</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SXW?</td>
</tr>
<tr>
<td>[10]</td>
<td>Block No. (Sequence)</td>
<td>SQB</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SQB?</td>
</tr>
<tr>
<td>[11]</td>
<td>Position (Sequence)</td>
<td>SQP</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SQP?</td>
</tr>
</tbody>
</table>
### SOP <source>[,<numeric>]

**Parameter**

- `<source>` = `<DECIMAL NUMERIC PROGRAM DATA>`
- `0` = 1/64 clock
- `1` = Pattern Sync (Variable)
- `2` = Pattern Sync (Variable)
- `4` = 1/8 clock
- `6` = 1/16 clock
- `8` = 1/32 clock
- `10` = Burst Output2
- `11` = 1/n clock

- `[<numeric>]` = `<DECIMAL NUMERIC PROGRAM DATA>`

When `11` (1/n clock) is not selected, `<numeric>` cannot be set and should be omitted.

- `2, 4, 8` to `511`:
  - `2, 4, 8` to `511`, in single steps

Valid only for MU181020A-001, and the setting upper limit varies as follows:

- In 1/2 mode: `1, 2, 4, 8, 9` to `255`, in single steps
- In 1/4 mode: `1, 2, 4, 8, 9` to `127`, in single steps
- In 1/8 mode: `1, 2, 4, 8, 9` to `63`, in single steps

**Function**

Sets the output signal for synchronization output.

**Example**

To set the output signal for synchronization output to 1/32 clock:

```
> SOP 8
```

To set the output signal for synchronization output to 1/511 clock:

```
> SOP 11,511
```

**Compatibility**

Partially compatible with the MP1632C and MP1775A (1: Pattern Sync only).

Compatible with the MP1761C.

### SOP?

**Response**

- `<source>` = `<NR1 NUMERIC RESPONSE DATA>` (FIX2)
- `[<numeric>]` = `<DECIMAL NUMERIC PROGRAM DATA>` (FIX3)

Omit `<numeric>` for other than 1/n clock.

**Function**

Queries the output signal for synchronization output.

**Example**

```
> SOP?
< SOP 11,511
< SOP 11, 8
< SOP  2
```

**Compatibility**

Partially compatible with the MP1632C and MP1775A.

Compatible with the MP1761C.
Chapter 8  Native Commands

**PSP <numeric>**

Parameter  
<numeric> = <DECIMAL NUMERIC PROGRAM DATA>  
1 to 68719476657    1 to 68,719,476,657 bits, in 16-bit steps  
In the case of Combination  
1 to 274877906625    1 to 274,877,906,625 bits, in 64-bit steps  

Function  
Sets the output position for synchronization output for Data, PRBS, and Zero-Substitution patterns.  

Example  
To set the synchronization output position to 17 bits:  
> PSP 17

Compatibility  
Compatible with the MP1632C.  
(Note that the resolution (step) is incompatible.)  
Compatible with the MP1761C.  
(Note that the resolution (step) should be 1-bit steps)

**PSP?**

Response  
<numeric> = <NR1 NUMERIC RESPONSE DATA> (FIX12)  

Function  
Queries the output position for synchronization output for Data, PRBS, and Zero-Substitution patterns.  

Example  
> PSP?  
< PSP 1  
< PSP 2147483569

Compatibility  
Compatible with the MP1632C and MP1761C.

**SCN <content>**

Parameter  
<content> = <DECIMAL NUMERIC PROGRAM DATA>  
0          Outputs synchronization signal at the top of Pattern A.  
1          Outputs synchronization signal at the top of Pattern B.  

Function  
Sets the output position for synchronization output for Alternate pattern.  

Example  
To set the output position for synchronization output at the top of Pattern A:  
> SCN 0

Compatibility  
Incompatible with existing models.
### SCN?

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;content&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt; (FIX1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function</td>
<td>Queries the output position for synchronization output for Alternate pattern.</td>
</tr>
<tr>
<td>Example</td>
<td>&gt; SCN?</td>
</tr>
<tr>
<td></td>
<td>&lt; SCN 0</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Incompatible with existing models.</td>
</tr>
</tbody>
</table>

### APO <numeric>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;numeric&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1 to 67108785 1 to 67,108,785 bits, in 16-bit steps</td>
</tr>
<tr>
<td>Function</td>
<td>Sets the output position for synchronization output for Alternate pattern.</td>
</tr>
<tr>
<td>Example</td>
<td>To set the synchronization output position to 33 bits:</td>
</tr>
<tr>
<td></td>
<td>&gt; APO 33</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Incompatible with existing models.</td>
</tr>
</tbody>
</table>

### APO?

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;numeric&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt; (FIX8)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function</td>
<td>Queries the output position for synchronization output for Alternate pattern.</td>
</tr>
<tr>
<td>Example</td>
<td>&gt; APO?</td>
</tr>
<tr>
<td></td>
<td>&lt; APO 1</td>
</tr>
<tr>
<td></td>
<td>&lt; APO 67108785</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Incompatible with existing models.</td>
</tr>
</tbody>
</table>

### SBL <numeric>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;numeric&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1 to 511 1 to 511 blocks, in 1-block steps</td>
</tr>
<tr>
<td>Function</td>
<td>The maximum setting value is the set number of blocks.</td>
</tr>
<tr>
<td>Example</td>
<td>Sets the output block position for synchronization output for Mixed Data pattern.</td>
</tr>
<tr>
<td></td>
<td>To set the synchronization output block to block 10:</td>
</tr>
<tr>
<td></td>
<td>&gt; SBL 10</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Incompatible with existing models.</td>
</tr>
</tbody>
</table>
### SBL?

**Response**

\(<\text{numeric}> = \text{<NR1 NUMERIC RESPONSE DATA>}\) (FIX3)

**Function**

Queries the output block position for synchronization output for Mixed Data pattern.

**Example**

> SBL?

< SBL 1

< SBL 511

**Compatibility**

Incompatible with existing models.

---

### SRW <numeric>

**Parameter**

\(<\text{numeric}> = \text{<DECIMAL NUMERIC PROGRAM DATA}>\)

1 to 16 1 to 16 rows, in 1-row steps

The maximum setting value is the set number of Rows.

**Function**

Sets the output Row position for synchronization output for Mixed Data pattern.

**Example**

To set the synchronization output Row to 1 Row:

> SRW 1

**Compatibility**

Incompatible with existing models.

---

### SRW?

**Response**

\(<\text{numeric}> = \text{<NR1 NUMERIC RESPONSE DATA>}\) (FIX2)

**Function**

Queries the output Row position for synchronization output for Mixed Data pattern.

**Example**

> SRW?

< SRW 1

< SRW 16

**Compatibility**

Incompatible with existing models.

---

### SML <numeric>

**Parameter**

\(<\text{numeric}> = \text{<DECIMAL NUMERIC PROGRAM DATA}>\)

1 to 511 1 to 511 blocks, in 1-block steps

The maximum setting value is the set number of blocks.

**Function**

Sets the output block position for synchronization output for Mixed Alternate pattern.

**Example**

To set the synchronization output block to block 10:

> SML 10

**Compatibility**

Incompatible with existing models.
### SML?

**Response**

\(<\text{numeric}> = \langle\text{NR1 NUMERIC RESPONSE DATA}\rangle \text{ (FIX3)}\)

**Function**

Queries the output block position for synchronization output for Mixed Alternate pattern.

**Example**

```plaintext
> SML?
< SML  1
< SML  511
```

**Compatibility**

Incompatible with existing models.

### SXO <content>

**Parameter**

\(<\text{content}> = \langle\text{DECIMAL NUMERIC PROGRAM DATA}\rangle\)

- 0  Outputs synchronization signal at the top of Pattern A.
- 1  Outputs synchronization signal at the top of Pattern B.

**Function**

Sets the output position for synchronization output for Mixed Alternate pattern.

**Example**

To set the synchronization output position to the top of Pattern A:

```plaintext
> SXO  0
```

**Compatibility**

Incompatible with existing models.

### SXO?

**Response**

\(<\text{content}> = \langle\text{NR1 NUMERIC RESPONSE DATA}\rangle \text{ (FIX1)}\)

**Function**

Queries the output position for synchronization output for Mixed Alternate pattern.

**Example**

```plaintext
> SXO?
< SXO  0
```

**Compatibility**

Incompatible with existing models.

### SXW <numeric>

**Parameter**

\(<\text{numeric}> = \langle\text{DECIMAL NUMERIC PROGRAM DATA}\rangle\)

- 1 to 16  1 to 16 rows, in 1-row steps
  
  The maximum setting value is the set number of Rows.

**Function**

Sets the output Row position for synchronization output for Mixed Alternate pattern.

**Example**

To set the synchronization output row to 2 Row:

```plaintext
> SXW  2
```

**Compatibility**

Incompatible with existing models.
### SXW?

**Response**

\[<\text{numeric}> = <\text{NR1 NUMERIC RESPONSE DATA}> \text{ (FIX2)}\]

**Function**

Queries the output Row position for synchronization output for Mixed Alternate pattern.

**Example**

```
> SXW?
< SXW  2
< SXW 16
```

**Compatibility**

Incompatible with existing models.

---

### SQB <numeric>

**Parameter**

\[<\text{numeric}> = <\text{DECIMAL NUMERIC PROGRAM DATA}>\]

- 1 to 128
- 1 to 128 blocks, in 1-block steps

The maximum setting value is the set number of blocks.

**Function**

Sets the output block position for synchronization output for Sequence pattern.

**Example**

To set the synchronization output block to block 128:

```
> SQB 128
```

**Compatibility**

Incompatible with existing models.

---

### SQB?

**Response**

\[<\text{numeric}> = <\text{NR1 NUMERIC RESPONSE DATA}> \text{ (FIX3)}\]

**Function**

Queries the output block position for synchronization output for Sequence pattern.

**Example**

```
> SQB?
< SQB   1
< SQB 128
```

**Compatibility**

Incompatible with existing models.

---

### SQP <numeric>

**Parameter**

\[<\text{numeric}> = <\text{DECIMAL NUMERIC PROGRAM DATA}>\]

- 1 to 1048497
- 1 to 1,048,497 bits, in 16-bit steps

**Function**

Sets the output position for synchronization output for Sequence pattern.

**Example**

To set the synchronization output position to 1,048,497:

```
> SQP 1048497
```

**Compatibility**

Incompatible with existing models.
## SQP?

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Response</strong></td>
<td>&lt;numeric&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt; (FIX7)</td>
</tr>
<tr>
<td><strong>Function</strong></td>
<td>Queries the output position for synchronization output for Sequence pattern.</td>
</tr>
<tr>
<td><strong>Example</strong></td>
<td>&gt; SQP?</td>
</tr>
<tr>
<td></td>
<td>&lt; SQP 1</td>
</tr>
<tr>
<td></td>
<td>&lt; SQP 104849?</td>
</tr>
<tr>
<td><strong>Compatibility</strong></td>
<td>Incompatible with existing models.</td>
</tr>
</tbody>
</table>
8.4 ED Commands

This section describes the commands for the MU181040A 12.5 Gbit/s ED and the MU181040B 14 Gbit/s ED. The commands related to each tab are described in separate sections below. Before executing a setting/query command for the MU181040A, specify the slot number of the module to be remotely controlled, by using the \MODule:ID command. Refer to Section 7.1 “Common Commands” and 9.1 “Basic Operation of Device Messages” for how to specify a slot number with the \MODule:ID command.

However, unless there is a special item, see the MU181040A.

8.4.1 Commands related to Result tab

![Figure 8.4.1-1 Result tab](image-url)
8.4 ED Commands

8.4.1.1 Result setting commands

Table 8.4.1.1-1 Result setting commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1]</td>
<td>Start</td>
<td>STA</td>
</tr>
<tr>
<td>[2]</td>
<td>Stop</td>
<td>STO</td>
</tr>
<tr>
<td>[3]</td>
<td>Query for measurement status</td>
<td>MSR?</td>
</tr>
<tr>
<td>[4]</td>
<td>Switching of measurement result display</td>
<td>SET</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SET?</td>
</tr>
<tr>
<td>[5]</td>
<td>Switching of time display</td>
<td>TIM</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TIM?</td>
</tr>
<tr>
<td>[6]</td>
<td>Query for measurement start time</td>
<td>MSA?</td>
</tr>
<tr>
<td>[7]</td>
<td>Query for measurement end time</td>
<td>MSO?</td>
</tr>
<tr>
<td>[8]</td>
<td>Query for measurement elapsed time</td>
<td>MLP?</td>
</tr>
<tr>
<td>[9]</td>
<td>Query for measurement remaining time</td>
<td>ETT?</td>
</tr>
<tr>
<td>[10]</td>
<td>Query for intermediate data creation time</td>
<td>INT?</td>
</tr>
<tr>
<td>[11]</td>
<td>Query for alarm occurrence time</td>
<td>AMD?</td>
</tr>
</tbody>
</table>
**STA**

**Function**
Starts BER measurement for the specified module. If this command is issued during measurement, the BER measurement is restarted.

**Example**

```
> STA
```

**Compatibility**
Compatible with the MP1632C and MP1762D.

---

**STO**

**Function**
Stops the BER measurement of the specified module.

**Example**

```
> STO
```

**Compatibility**
Compatible with the MP1632C and MP1762D.

---

**MSR?**

**Response**

```
<numeric> = <NR1 NUMERIC RESPONSE DATA> (FIX1)
```

- 0: Measurement stops
- 1: During measurement

**Function**
Queries the measurement processing state during bit error alarm measurement.

**Example**

```
> MSR?
< MSR 0
```

**Compatibility**
Partially compatible with the MP1632C (0: measurement stops).

---

**SET <setting>**

**Parameter**

```
<setting> = <DECIMAL NUMERIC PROGRAM DATA>
```

- 0: Bit error alarm measurement result
- 1: Logging measurement result
- 2: Histogram measurement result

**Function**
Sets the display item on the BER measurement result screen.

**Example**

```
To set the display item on the BER measurement result screen to the bit error alarm measurement results:
> SET 0
```

**Compatibility**
Incompatible with existing models.

---

**SET?**

**Response**

```
<setting> = <NR1 NUMERIC RESPONSE DATA> (FIX1)
```

**Function**
Queries the display item on the BER measurement result screen.

**Example**

```
> SET?
< SET 0
```

**Compatibility**
Incompatible with existing models.
TIM <time>

Parameter  

<table>
<thead>
<tr>
<th>&lt;time&gt;</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Displays the current date and time (Date&amp;Time).</td>
</tr>
<tr>
<td>1</td>
<td>Displays the current date and time (Date&amp;Time).</td>
</tr>
<tr>
<td>3</td>
<td>Displays the remaining time based on the measurement period (Remain Time).</td>
</tr>
<tr>
<td>4</td>
<td>Displays the elapsed time based on the measurement period (Elapsed Time).</td>
</tr>
<tr>
<td>5</td>
<td>Displays the measurement start time (Start Time).</td>
</tr>
</tbody>
</table>

Function  
Sets the measurement time display type.

Example  
To set the measurement time display type to current time:

```
> TIM 0
```

Compatibility  
Compatible with the MP1632C.
Partially compatible with the MP1762D.
(0: Date&Time, 1: Date&Time, 3: Remain Time, 4: Elapsed Time)

TIM?

Response  

<time> = <NR1 NUMERIC RESPONSE DATA> (FIX1)

<table>
<thead>
<tr>
<th>&lt;time&gt;</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0, 3, 4, 5</td>
<td></td>
</tr>
</tbody>
</table>

Function  
Queries the measurement time display type.

Example  

```
> TIM?
< TIM 0
```

Compatibility  
Compatible with the MP1632C.
Partially compatible with the MP1762D.
MSA?

Response

<year>,<month>,<day>,<hour>,<min>,<second> = <NR1 NUMERIC RESPONSE DATA>

<year> 0000, 2000 to 2036 Year 2000 to 2036 (FIX4)
<month> 00, 01 to 12 January to December (FIX2)
<day> 00, 01 to 31 1st to 31st (FIX2)
<hour> 00 to 23 0 to 23 hours (FIX2)
<min> 00 to 59 0 to 59 minutes (FIX2)
<second> 00 to 59 0 to 59 seconds (FIX2)

Function

Queries the measurement start time during bit error alarm measurement.
"0000,00,00,00,00,00" is returned if there is no measurement start time data.

Example

> MSA?
< MSA 2036,12,31,23,59,59

Compatibility

Incompatible with existing models.

MSO?

Response

<year>,<month>,<day>,<hour>,<min>,<second> = <NR1 NUMERIC RESPONSE DATA>

<year> 0000, 2000 to 2036 Year 2000 to 2036 (FIX4)
<month> 00, 01 to 12 January to December (FIX2)
<day> 00, 01 to 31 1st to 31st (FIX2)
<hour> 00 to 23 0 to 23 hours (FIX2)
<min> 00 to 59 0 to 59 minutes (FIX2)
<second> 00 to 59 0 to 59 seconds (FIX2)

Function

Queries the measurement end time during bit error alarm measurement.
"0000,00,00,00,00,00" is returned if there is no measurement end time data.

Example

> MSO?
< MSO 2036,12,31,23,59,59

Compatibility

Compatible with the MP1632C.
### MLP?

**Response**

<day>,<hour>,<min>,<second> = <NR1 NUMERIC RESPONSE DATA>

- **<day>**
  - 00 to 31
  - 0 to 31st (FIX2)

- **<hour>**
  - 00 to 23
  - 0 to 23 hours (FIX2)

- **<min>**
  - 00 to 59
  - 0 to 59 minutes (FIX2)

- **<second>**
  - 00 to 59
  - 0 to 59 seconds (FIX2)

**Function**

Queries the measurement elapsed time during bit error alarm measurement.

"00,00,00,00" is returned if there is no measurement elapsed time data.

**Example**

```plaintext
> MLP?
< MLP 00,00,00,00
```

**Compatibility**

Incompatible with existing models.

### ETI?

**Response**

<day>,<hour>,<min>,<second> = <NR1 NUMERIC RESPONSE DATA>

- **<day>**
  - 00 to 31
  - 0 to 31st (FIX2)

- **<hour>**
  - 00 to 23
  - 0 to 23 hours (FIX2)

- **<min>**
  - 00 to 59
  - 0 to 59 minutes (FIX2)

- **<second>**
  - 00 to 59
  - 0 to 59 seconds (FIX2)

"0,0,0,0" is returned when the measurement is finished.

When measurement is not performed, the measurement period set at that time is returned.

**Function**

Queries the measurement remaining time during bit error alarm measurement.

**Example**

```plaintext
> ETI?
< ETI 00,00,00,00
```

**Compatibility**

Incompatible with existing models.
Chapter 8  Native Commands

INT?

Response

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;year&gt;,&lt;month&gt;,&lt;day&gt;,&lt;hour&gt;,&lt;min&gt;,&lt;second&gt;</td>
<td>= &lt;NR1 NUMERIC RESPONSE DATA&gt;</td>
</tr>
<tr>
<td>&lt;year&gt;</td>
<td>0000, 2000 to 2036 Year 2000 to 2036 (FIX4)</td>
</tr>
<tr>
<td>&lt;month&gt;</td>
<td>00, 01 to 12 January to December (FIX2)</td>
</tr>
<tr>
<td>&lt;day&gt;</td>
<td>00, 01 to 31 1st to 31st (FIX2)</td>
</tr>
<tr>
<td>&lt;hour&gt;</td>
<td>00 to 23 0 to 23 hours (FIX2)</td>
</tr>
<tr>
<td>&lt;min&gt;</td>
<td>00 to 59 0 to 59 minutes (FIX2)</td>
</tr>
<tr>
<td>&lt;second&gt;</td>
<td>00 to 59 0 to 59 seconds (FIX2)</td>
</tr>
</tbody>
</table>

Function
Queries the measurement intermediate data creation time during bit error alarm measurement. "0000,00,00,00,00,00" is returned if there is no measurement intermediate data.

Example

> INT?
< INT 2036,12,31,23,59,59

Compatibility
Incompatible with existing models.

AMD? <alarm>

Parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;alarm&gt;</td>
<td>= &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
</tr>
<tr>
<td>0</td>
<td>Outputs all item data</td>
</tr>
<tr>
<td>3</td>
<td>Clock Loss occurrence time</td>
</tr>
<tr>
<td>4</td>
<td>Clock Loss recovery time</td>
</tr>
<tr>
<td>5</td>
<td>Pattern Sync Loss occurrence time</td>
</tr>
<tr>
<td>6</td>
<td>Pattern Sync Loss recovery time</td>
</tr>
<tr>
<td>7</td>
<td>CR Unlock occurrence time</td>
</tr>
<tr>
<td>8</td>
<td>CR Unlock recovery time</td>
</tr>
</tbody>
</table>

Response

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;year&gt;,&lt;month&gt;,&lt;day&gt;,&lt;hour&gt;,&lt;min&gt;,&lt;second&gt;</td>
<td>= &lt;NR1 NUMERIC RESPONSE DATA&gt;</td>
</tr>
<tr>
<td>&lt;year&gt;</td>
<td>00 to 36 Year 2000 to 2036 (FIX2)</td>
</tr>
<tr>
<td>&lt;month&gt;</td>
<td>01 to 12 January to December (FIX2)</td>
</tr>
<tr>
<td>&lt;day&gt;</td>
<td>01 to 31 1st to 31st (FIX2)</td>
</tr>
<tr>
<td>&lt;hour&gt;</td>
<td>00 to 23 0 to 23 hours (FIX2)</td>
</tr>
<tr>
<td>&lt;min&gt;</td>
<td>00 to 59 0 to 59 minutes (FIX2)</td>
</tr>
<tr>
<td>&lt;second&gt;</td>
<td>00 to 59 0 to 59 seconds (FIX2)</td>
</tr>
</tbody>
</table>

Function
Queries the time when a specified alarm has occurred or recovered during bit error alarm measurement. "99-99-99 99:99:99" is returned if no alarm has been occurred.

Example

To query the time when the clock loss has occurred during bit error alarm measurement:

> AMD? 3
< 06-01-01 01:01:01

To query the time for occurrences and recoveries of all alarm items during bit error alarm measurement:

> AMD? 0
8.4 ED Commands

< 06-01-01 01:01:01, 06-01-01 01:02:01, 06-01-01 01:01:01, 06-01-01 02:01:01, 06-01-01 02:01:01, 06-01-01 01:01:01, 06-01-01 02:01:01

Compatibility

Compatible with the MP1762D.
8.4.1.2 Error/Alarm setting commands

Figure 8.4.1.2-1 Error/Alarm result display screen

Figure 8.4.1.2-2 Error/Alarm result display screen (when Zoom is selected)
Figure 8.4.1.2-3  Error/Alarm result display screen (Result Sub Display)
### Table 8.4.1.2-1  Error/Alarm setting commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Zoom</td>
<td>ZOM</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ZOM?</td>
</tr>
<tr>
<td>2</td>
<td>History Reset</td>
<td>HRE</td>
</tr>
<tr>
<td>3</td>
<td>No label (Query for measurement data)</td>
<td>END?</td>
</tr>
<tr>
<td>4</td>
<td>No label (Query for monitoring item)</td>
<td>MTR?</td>
</tr>
<tr>
<td>5</td>
<td>ER (Total)</td>
<td>ER?</td>
</tr>
<tr>
<td>6</td>
<td>EC (Total)</td>
<td>EC?</td>
</tr>
<tr>
<td>7</td>
<td>%EFI</td>
<td>EFI?</td>
</tr>
<tr>
<td>8</td>
<td>EI</td>
<td>EI?</td>
</tr>
<tr>
<td>9</td>
<td>Frequency (kHz)</td>
<td>FRQ?</td>
</tr>
<tr>
<td>10</td>
<td>Clock Count</td>
<td>CC?</td>
</tr>
<tr>
<td>11</td>
<td>Clock Loss</td>
<td>CLI?</td>
</tr>
<tr>
<td>12</td>
<td>Sync Loss</td>
<td>SLI?</td>
</tr>
<tr>
<td>13</td>
<td>Error</td>
<td>ERS?</td>
</tr>
<tr>
<td>14</td>
<td>Error Performance</td>
<td>PFP?</td>
</tr>
<tr>
<td>15</td>
<td>Error Performance %</td>
<td>PFC?</td>
</tr>
<tr>
<td>16</td>
<td>Threshold EI</td>
<td>THE?</td>
</tr>
<tr>
<td>17</td>
<td>Threshold %EFI</td>
<td>THF?</td>
</tr>
<tr>
<td>18</td>
<td>No label (Query for alarm interval measurement data)</td>
<td>AIN?</td>
</tr>
<tr>
<td>19</td>
<td>No label (Query for 1 second data)</td>
<td>OSD?</td>
</tr>
<tr>
<td>20</td>
<td>No label (Query for intermediate data)</td>
<td>IMD?</td>
</tr>
<tr>
<td>21</td>
<td>No label (Query for error rate detection status)</td>
<td>BES?</td>
</tr>
</tbody>
</table>
## 8.4 ED Commands

### ZOM <numeric>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;numeric&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Zoom out</td>
</tr>
<tr>
<td>1</td>
<td>Zoom in</td>
</tr>
</tbody>
</table>

**Function**
Sets display enlargement for the measurement result display screen IN and OUT.

**Example**
To set enlargement for the measurement result display screen IN:

```
> ZOM 1
```

**Compatibility**
Incompatible with existing models.

### ZOM?  

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;numeric&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt; (FIX1)</th>
</tr>
</thead>
</table>

**Function**
Queries the display enlargement for the measurement result display screen IN and OUT states.

**Example**

```
> ZOM?
< ZOM 1
```

**Compatibility**
Incompatible with existing models.

### HRE

**Function**
Resets the histories on the measurement result display screen.

**Example**

```
> HRE
```

**Compatibility**
Compatible with the MP1632C.
Chapter 8  Native Commands

**END? <result>,<item>**

Parameter

\(<\text{result}> = \text{DECIMAL NUMERIC PROGRAM DATA}\>

<table>
<thead>
<tr>
<th>0</th>
<th>Time data</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Alarm measurement data</td>
</tr>
<tr>
<td>2</td>
<td>Error measurement data</td>
</tr>
<tr>
<td>3</td>
<td>Threshold EI,EFI data</td>
</tr>
<tr>
<td>4</td>
<td>Error performance</td>
</tr>
</tbody>
</table>

\(<\text{item}> = \text{DECIMAL NUMERIC PROGRAM DATA}\>

<table>
<thead>
<tr>
<th>0</th>
<th>Outputs all items</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 to 13</td>
<td>Outputs data of 1 item</td>
</tr>
</tbody>
</table>

For the display result items corresponding to each <result> see Table 8.4.1.2-2.

Response

\(<\text{string}> = \text{STRING RESPONSE DATA}\>

Table 8.4.1.2-2  Response Format

<table>
<thead>
<tr>
<th>Form</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Form1</td>
<td>XXXXXXX</td>
<td>For 0 to 9999999</td>
</tr>
<tr>
<td>Integer type</td>
<td>X.XXXXEXXX</td>
<td>For 1.0000E07 to 9.9999E17</td>
</tr>
<tr>
<td></td>
<td>ERR</td>
<td>When no data corresponds to the query</td>
</tr>
<tr>
<td>Form2</td>
<td>X.XXXXE-XX</td>
<td>For 0.0000E–18 to 1.0000E00</td>
</tr>
<tr>
<td>Fraction type</td>
<td>ERR</td>
<td>When no data corresponds to the query</td>
</tr>
<tr>
<td>Form3</td>
<td>XXX.XXXXX</td>
<td>For 0.0000 to 100.0000</td>
</tr>
<tr>
<td>% type</td>
<td>ERR</td>
<td>When no data corresponds to the query</td>
</tr>
<tr>
<td>Form4</td>
<td>XXXXXXXXXXX</td>
<td>For 0 to MAX (Hz)</td>
</tr>
<tr>
<td>Frequency type</td>
<td>ERR</td>
<td>When no data corresponds to the query</td>
</tr>
<tr>
<td>Form5</td>
<td>XX-XX-XX XX:XX:XX</td>
<td>For year-month-day hour:minute:second</td>
</tr>
<tr>
<td>Time type</td>
<td>X XX-XX-XX</td>
<td>For day hour:minute:second</td>
</tr>
<tr>
<td></td>
<td>ERR</td>
<td>When no data corresponds to the query</td>
</tr>
<tr>
<td>Form6</td>
<td>XXXXXXXXXXX,XXX.XXX</td>
<td>For less than 1E+8</td>
</tr>
<tr>
<td>Mixed type</td>
<td>X.XXXXEXXX,XXX.XXX</td>
<td>For 1E+8 and above</td>
</tr>
<tr>
<td></td>
<td>ERR</td>
<td>When no data corresponds to the query</td>
</tr>
</tbody>
</table>

Function

Queries the measurement data corresponding to the parameter.
### Table 8.4.1.2-3 Parameter

<table>
<thead>
<tr>
<th>&lt;result&gt;</th>
<th>&lt;item&gt;</th>
<th>Items</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>Measurement start time</td>
<td>Form5</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>Measurement end time</td>
<td>Form5</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>Measurement elapsed time</td>
<td>Form5</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>Measurement remaining time</td>
<td>Form5</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>Clock Loss interval counts</td>
<td>Form1</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>Sync Loss interval counts</td>
<td>Form1</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>CR Unlock interval counts</td>
<td>Form1</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>Error Rate (total)</td>
<td>Form2</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>Error Count (total)</td>
<td>Form1</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>Clock Count</td>
<td>Form1</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>EI</td>
<td>Form1</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>%EFI</td>
<td>Form3</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>Error Rate (INS)</td>
<td>Form2</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>Error Rate (OMI)</td>
<td>Form2</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>Error Count (INS)</td>
<td>Form1</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>Error Count (OMI)</td>
<td>Form1</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>Error Rate (Transition)</td>
<td>Form2</td>
</tr>
<tr>
<td></td>
<td>11</td>
<td>Error Rate (Non Transition)</td>
<td>Form2</td>
</tr>
<tr>
<td></td>
<td>12</td>
<td>Error Count (Transition)</td>
<td>Form1</td>
</tr>
<tr>
<td></td>
<td>13</td>
<td>Error Count (Non Transition)</td>
<td>Form1</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>&gt;1.0E–3</td>
<td>Form6</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>&gt;1.0E–4</td>
<td>Form6</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>&gt;1.0E–5</td>
<td>Form6</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>&gt;1.0E–6</td>
<td>Form6</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>&gt;1.0E–7</td>
<td>Form6</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>&gt;1.0E–8</td>
<td>Form6</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>≤1.0E–8</td>
<td>Form6</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>ES</td>
<td>Form1</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>EFS</td>
<td>Form1</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>SES</td>
<td>Form1</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>DM</td>
<td>Form1</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>US</td>
<td>Form1</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>%ES</td>
<td>Form3</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>%EFS</td>
<td>Form3</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>%SES</td>
<td>Form3</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>%DM</td>
<td>Form3</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>%US</td>
<td>Form3</td>
</tr>
</tbody>
</table>

**Example**

> END? 0,1

< 05-11-30 18:00:00

**Compatibility**

Partially compatible with the MP1762D.
### MTR? <item>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Bit Error (Total Error)</td>
</tr>
<tr>
<td>3</td>
<td>Clock Loss or CR Unlock</td>
</tr>
<tr>
<td>4</td>
<td>Pattern Sync Loss</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Response</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Numeric</td>
<td>&lt;NR1 NUMERIC RESPONSE DATA&gt; (FIX1)</td>
</tr>
<tr>
<td>0</td>
<td>When an alarm occurs</td>
</tr>
<tr>
<td>1</td>
<td>When no alarm occurs</td>
</tr>
<tr>
<td>2</td>
<td>When no data corresponds to the query</td>
</tr>
</tbody>
</table>

**Function**: Queries the real-time occurrence status of the monitoring item corresponding to the parameter.

**Example**: To query the real-time occurrence status of the Total Error:

```
> MTR? 0
< MTR 0
```

**Compatibility**: Incompatible with existing models.

### ER?

<table>
<thead>
<tr>
<th>Response</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Numeric</td>
<td>&lt;NR3 NUMERIC RESPONSE DATA&gt;</td>
</tr>
<tr>
<td>0.0000E–18 to 1.0000E–00</td>
<td>Error rate</td>
</tr>
</tbody>
</table>

**Function**: Outputs the error rate measurement result according to the output format.

"0.0000E–00" is returned if no data exists.

**Example**: To query the error rate:

```
> ER?
< ER 0.0000E-00
< ER 1.9999E-1
```

**Compatibility**: Compatible with the MP1632C and MP1762D.

### EC?

<table>
<thead>
<tr>
<th>Response</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Numeric</td>
<td>&lt;NR1 or NR3 NUMERIC RESPONSE DATA&gt;</td>
</tr>
<tr>
<td>1.0000E07 to 9.9999E17</td>
<td>Error count</td>
</tr>
<tr>
<td>0 to 9999999</td>
<td></td>
</tr>
</tbody>
</table>

**Function**: Outputs the error count measurement results according to the output format.

"1.0000E–99" is returned if no data exists.

**Example**: To query the error count:

```
> EC?
< EC 1.0000E-99
< EC 1.0000E09
< EC 9999999
```

**Compatibility**: Compatible with the MP1632C and MP1762D.
### EFI?

<table>
<thead>
<tr>
<th>Response</th>
<th>(&lt;\text{numeric}&gt; = &lt;\text{NR2 NUMERIC RESPONSE DATA}&gt;)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.0000 to 100.0000 Error free interval rate</td>
</tr>
<tr>
<td>Function</td>
<td>Outputs the measurement result of the error free interval rate according to the output format.</td>
</tr>
<tr>
<td></td>
<td>&quot;999.9999&quot; is returned if no data exists.</td>
</tr>
<tr>
<td>Example</td>
<td>&gt; EFI?</td>
</tr>
<tr>
<td></td>
<td>&lt; EFI 999.9999</td>
</tr>
<tr>
<td></td>
<td>&lt; EFI 20.0088</td>
</tr>
<tr>
<td></td>
<td>&lt; EFI 100.0000</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Compatible with the MP1632C and MP1762D.</td>
</tr>
</tbody>
</table>

### EI?

<table>
<thead>
<tr>
<th>Response</th>
<th>(&lt;\text{numeric}&gt; = &lt;\text{NR1 or NR3 NUMERIC RESPONSE DATA}&gt;)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1.0000E07 to 9.9999E17 Error interval 0 to 9999999</td>
</tr>
<tr>
<td>Function</td>
<td>Outputs the measurement result of the error interval count according to the output format.</td>
</tr>
<tr>
<td></td>
<td>&quot;1.0000E–99&quot; is returned if no data exists.</td>
</tr>
<tr>
<td>Example</td>
<td>&gt; EI?</td>
</tr>
<tr>
<td></td>
<td>&lt; EI 1.0000E–99</td>
</tr>
<tr>
<td></td>
<td>&lt; EI 1.0000E09</td>
</tr>
<tr>
<td></td>
<td>&lt; EI 1999984</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Compatible with the MP1632C and MP1762D.</td>
</tr>
</tbody>
</table>
Chapter 8 Native Commands

FRQ?
Response \(<\text{numeric}> = <\text{NR2 NUMERIC RESPONSE DATA}>\)

\(XXXXX.XXX\) Clock frequency

Function Outputs the measurement results of the clock frequency according to the output format.

"FRQ 0.000" is returned for clock loss.

Example

\(> \text{FRQ?}\)

\(<\text{FRQ 0.000}\)

\(<\text{FRQ 12500.000}\)

Compatibility Compatible with the MP1632C and MP1762D.

CC?
Response \(<\text{numeric}> = <\text{NR1 or NR3 NUMERIC RESPONSE DATA}>\)

1.0000E07 to 9.9999E17 Clock count

0 to 99999999

Function Outputs the measurement results of the clock count according to the output format.

"1.0000E–99" is returned if no data exists.

Example

\(> \text{CC?}\)

\(<\text{CC 1.0000E–99}\)

\(<\text{CC 1.0000E–09}\)

\(<\text{CC 5000256}\)

Compatibility Compatible with the MP1632C and MP1762D.

CLI?
Response \(<\text{numeric}> = <\text{NR1 NUMERIC RESPONSE DATA}> (\text{FIX1})\)

0 Non Clock Loss status

1 Clock Loss status

Function Displays the clock input status.

Example

\(> \text{CLI?}\)

\(<\text{CLI 0}\)

Compatibility Compatible with the MP1632C and MP1762D.
### SLI?

**Response**  
<numeric> = <NR1 NUMERIC RESPONSE DATA> (FIX1)  
0  
Non Sync Loss status  
1  
Sync Loss status  

**Function**  
Displays the pattern synchronization status.  

**Example**  
> SLI?  
< SLI 0  

**Compatibility**  
Compatible with the MP1632C and MP1762D.

### ERS?

**Response**  
<numeric> = <NR1 NUMERIC RESPONSE DATA> (FIX1)  
0  
Status when no error detected  
1  
Status when error detected  

**Function**  
Displays the error detection status.  

**Example**  
> ERS?  
< ERS 0  

**Compatibility**  
Compatible with the MP1632C and MP1762D.

### PFP?

**Response**  
<numeric> = <NR2 or NR3 NUMERIC RESPONSE DATA> (FIX10)  
XXX.XXXX  
%ES data  
<numeric> = <NR2 or NR3 NUMERIC RESPONSE DATA> (FIX10)  
XXX.XXXX  
%EFS data  
<numeric> = <NR2 or NR3 NUMERIC RESPONSE DATA> (FIX10)  
XXX.XXXX  
%SFS data  
<numeric> = <NR2 or NR3 NUMERIC RESPONSE DATA> (FIX10)  
XXX.XXXX  
%US data  
<numeric> = <NR2 or NR3 NUMERIC RESPONSE DATA> (FIX10)  
XXX.XXXX  
%DM data  

Displays the above items delimited with a comma (,) (PFP XX,XX,XX,XX).  
"1.0000E–99" is returned if no measurement data exists.  

**Function**  
Outputs the % items of the performance measurement results.  

**Example**  
> PFP?  
< PFP  
1.0000E-99,1.0000E-99,1.0000E-99,1.0000E-99,1.0000E-99  
< PFP  
51.4285, 48.5714, 25.7142, 100.0000, 2.7777  

**Compatibility**  
Compatible with the MP1632C.
Chapter 8  Native Commands

PFC?

Response

\(<\text{numeric}> = \langle\text{NR1 or NR3 NUMERIC RESPONSE DATA}\rangle \text{ (FIX10)}\)

XXXXXXXXXX  ES data

XXXXXXXXXX  EFS data

XXXXXXXXXX  SFS data

XXXXXXXXXX  US data

XXXXXXXXXX  DM data

XXXXXXXXXX  EC data

Displays the above items delimited with a comma (,) \(\text{(PFC XX,XX,XX,XX,XX,XX)}\).

"1.0000E–99" is returned if no measurement data exists.

Function

Outputs the Count items of the performance measurement results.

Example

\(> \text{PFC?}\)

\(< \text{PFC}\)

\(1.0000E-99,1.0000E-99,1.0000E-99,1.0000E-99,1.0000E-99,\)

\(1.0000E-99\)

\(< \text{PFC}\)

\(1800,\ 1800,\ 900,\ 59,\)

\(100,1.0000E-99\)

Compatibility

Compatible with the MP1632C.
8.4 ED Commands

THE?

Response

\(<\text{numeric}> = <\text{NR1 or NR3 NUMERIC RESPONSE DATA}> (\text{FIX10})\>

\(\text{XXXXXXXX} \quad \>1\text{E–03 data}\)
\(<\text{numeric}> = <\text{NR1 or NR3 NUMERIC RESPONSE DATA}> (\text{FIX10})\>

\(\text{XXXXXXXX} \quad \>1\text{E–04 data}\)
\(<\text{numeric}> = <\text{NR1 or NR3 NUMERIC RESPONSE DATA}> (\text{FIX10})\>

\(\text{XXXXXXXX} \quad \>1\text{E–05 data}\)
\(<\text{numeric}> = <\text{NR1 or NR3 NUMERIC RESPONSE DATA}> (\text{FIX10})\>

\(\text{XXXXXXXX} \quad \>1\text{E–06 data}\)
\(<\text{numeric}> = <\text{NR1 or NR3 NUMERIC RESPONSE DATA}> (\text{FIX10})\>

\(\text{XXXXXXXX} \quad \>1\text{E–07 data}\)
\(<\text{numeric}> = <\text{NR1 or NR3 NUMERIC RESPONSE DATA}> (\text{FIX10})\>

\(\text{XXXXXXXX} \quad \leq1\text{E–08 data}\)

Displays the above items delimited with a comma (,) (THE XX,XX,XX,XX,XX,XX).

"1.0000E–99" is returned if no measurement data exists.

Function

Outputs the threshold EI measurement results.

Example

> THE?

< THE

1.0000E–99, 1.0000E–99, 1.0000E–99, 1.0000E–99, 1.0000E–99,
1.0000E–99, 1.0000E–99

< THE

3, 4, 5, 6, 7, 8, 9

Compatibility

Compatible with the MP1632C.
**THF?**

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;numeric&gt; = &lt;NR2 or NR3 NUMERIC RESPONSE DATA&gt; (FIX10)</th>
</tr>
</thead>
<tbody>
<tr>
<td>XXX.XXXX</td>
<td>&gt;1E–03 data</td>
</tr>
<tr>
<td>XXX.XXXX</td>
<td>&gt;1E–04 data</td>
</tr>
<tr>
<td>XXX.XXXX</td>
<td>&gt;1E–05 data</td>
</tr>
<tr>
<td>XXX.XXXX</td>
<td>&gt;1E–06 data</td>
</tr>
<tr>
<td>XXX.XXXX</td>
<td>&gt;1E–07 data</td>
</tr>
<tr>
<td>XXX.XXXX</td>
<td>&gt;1E–08 data</td>
</tr>
<tr>
<td>Displays the above items delimited with a comma (,) (THF XX,XX,XX,XX,XX).</td>
<td></td>
</tr>
</tbody>
</table>

"1.0000E–99" is returned if no measurement data exists.

**Function**

Outputs the threshold %EFI measurement results.

**Example**

> THF?

< THF

1.0000E-99,1.0000E-99,1.0000E-99,1.0000E-99,1.0000E-99
1.0000E-99,1.0000E-99

< THF 97.0000, 96.0000, 95.0000, 94.0000,
93.0000, 92.0000, 91.0000

**Compatibility**

Compatible with the MP1632C.

---

**AIN?**

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;numeric&gt; = &lt;NR1 or NR3 NUMERIC RESPONSE DATA&gt; (FIX10)</th>
</tr>
</thead>
<tbody>
<tr>
<td>XXXXXXXXXXX</td>
<td>Pattern Sync Loss data</td>
</tr>
<tr>
<td>XXXXXXXXXXX</td>
<td>Clock Loss data</td>
</tr>
<tr>
<td>XXXXXXXXXXX</td>
<td>CR Unlock data</td>
</tr>
</tbody>
</table>

Displays the above items delimited with a comma (,) (AIN XX,XX,XX).

"1.0000E–99" is returned if no measurement data exists.

**Function**

Outputs the alarm interval measurement results.

**Example**

> AIN?

< AIN 1.0000E-99,1.0000E-99,1.0000E-99
< AIN 0, 0,1.0000E-99

**Compatibility**

Partially compatible with the MP1632C.
### OSD?

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;numeric&gt; = &lt;NR1 or NR3 NUMERIC RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Average error rate for 1 second, average error count for 1 second</td>
</tr>
<tr>
<td></td>
<td>X.XXXXXE–XX, XXXXXXXX (average error count for 1 second is less than 1E+8)</td>
</tr>
<tr>
<td></td>
<td>X.XXXXXE–XX.X.XXXXXXX (average error count for 1 second is 1E+8 and above)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Function</th>
<th>Outputs the measurement results for 1 second according to the output format.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>&quot;0.0000E–00,1.0000E–99&quot; is returned if no data exists.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Example</th>
<th>To output the measurement results for 1 second according to the output format:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>&gt; OSD?</td>
</tr>
<tr>
<td></td>
<td>&lt; OSD 0.0000E–00,1.0000E–99</td>
</tr>
<tr>
<td></td>
<td>&lt; OSD 2.0000E–01,1.0000E09</td>
</tr>
</tbody>
</table>

| Compatibility | Compatible with the MP1762D. |

### IMD? <result>,<item>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;result&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Same as END?</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;string&gt; = &lt;STRING RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Same as END?</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Function</th>
<th>Displays the measurement intermediate data.</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Example</th>
<th>&gt; IMD? 0,1</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>&lt; 05-11-30 18:00:00</td>
</tr>
</tbody>
</table>

| Compatibility | Compatible with the MP1762D. |
# BES? <error>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;error&gt; = &lt; DECIMAL NUMERIC PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Total Error</td>
</tr>
<tr>
<td>1</td>
<td>Insertion Error</td>
</tr>
<tr>
<td>2</td>
<td>Omission Error</td>
</tr>
<tr>
<td>3</td>
<td>Transition Error</td>
</tr>
<tr>
<td>4</td>
<td>Non-Transition Error</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;numeric&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt; (FIX1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Error detection</td>
</tr>
<tr>
<td>1</td>
<td>No error detection</td>
</tr>
</tbody>
</table>

**Function**: Displays the error detection status.

**Example**

```
> BES? 0
<BES 0
```

**Compatibility**: Compatible with the MP1632C.

(The response 0 is returned for No error detection, and 1 is returned for Error detection.)
8.4.1.3 Logging setting commands

Figure 8.4.1.3-1 Logging setting
Table 8.4.1.3-1 Logging setting commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1]</td>
<td>Logging ON/OFF</td>
<td>LON</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LON?</td>
</tr>
<tr>
<td>[2]</td>
<td>Clear</td>
<td>LCR</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LCP?</td>
</tr>
<tr>
<td>[4]</td>
<td>Item Select</td>
<td>LCD</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LCD?</td>
</tr>
<tr>
<td>[5]</td>
<td>Error Threshold</td>
<td>LCT</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LCT?</td>
</tr>
<tr>
<td>[6]</td>
<td>Squelch</td>
<td>LCS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LCS?</td>
</tr>
<tr>
<td>[7]</td>
<td>Set All/Reset All</td>
<td>LAS</td>
</tr>
</tbody>
</table>
8.4 ED Commands

LON <numeric>

Parameter  
<numeric> = <DECIMAL NUMERIC PROGRAM DATA>
0 OFF
1 ON

Function  
Sets logging execution ON/OFF.

Example  
To set logging execution ON:
> LON 1

Compatibility  
Incompatible with existing models.

LON?

Response  
<numeric> = <NR1 NUMERIC RESPONSE DATA> (FIX1)

Function  
Queries the logging execution ON/OFF state.

Example  
> LON?
< LON 1

Compatibility  
Incompatible with existing models.

LCR

Function  
Clears the logging results.

Example  
> LCR

Compatibility  
Incompatible with existing models.

LCP <numeric>

Parameter  
<numeric> = <DECIMAL NUMERIC PROGRAM DATA>
0 10 s
1 30 s
2 1 min
3 10 min
4 30 min
5 1 hour

Function  
Sets the logging result capturing period.

Example  
To set the logging result capturing period to 10 seconds:
> LCP 0

Compatibility  
Incompatible with existing models.
### LCP?

**Response**

<numeric> = <NR1 NUMERIC RESPONSE DATA> (FIX1)

**Function**

Queries the logging result capturing period.

**Example**

> LCP?

< LCP 0

**Compatibility**

Incompatible with existing models.

### LCD <item>,<numeric>

**Parameter**

<item> = <DECIMAL NUMERIC PROGRAM DATA>

0 Slot Information: Slot information
1 Test Pattern: Test pattern type
2 Start Time: Measurement start time
3 End Time: Measurement end time
4 Error Rate / Count: Error rate and error count
5 EI·%EFI: Error interval and error free interval rate
6 Frequency: Frequency
7 Clock Count: Clock count
8 Alarm Occur / Recover: Alarm occurrence and recovery time
9 Alarm Interval: Alarm interval
10 1 second Data: Average data for 1 second
11 Adds the data input threshold setting value to the logging item.
12 Adds the clock phase setting value to the logging item.

<numeric> = <DECIMAL NUMERIC PROGRAM DATA>

0 OFF
1 ON

**Function**

Sets the specified logging measurement item ON/OFF.

**Example**

To set the logging slot information ON:

> LCD 0,1

**Compatibility**

Incompatible with existing models.
### LCD? <item>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;item&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Response</td>
<td>&lt;numeric&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt; (FIX1)</td>
</tr>
<tr>
<td>Response Value</td>
<td>0 OFF</td>
</tr>
<tr>
<td>Response Value</td>
<td>1 ON</td>
</tr>
<tr>
<td>Function</td>
<td>Sets the specified logging measurement item ON/OFF.</td>
</tr>
<tr>
<td>Example</td>
<td>&gt; LCD? 0</td>
</tr>
<tr>
<td>Example</td>
<td>&lt; LCD 1</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Incompatible with existing models.</td>
</tr>
</tbody>
</table>

### LCT <thre>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;thre&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Response</td>
<td>&lt;thre&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt; (FIX1)</td>
</tr>
<tr>
<td>Response Value</td>
<td>0 &lt;&gt;0</td>
</tr>
<tr>
<td>Response Value</td>
<td>1 &gt;1E–3</td>
</tr>
<tr>
<td>Response Value</td>
<td>2 &gt;1E–4</td>
</tr>
<tr>
<td>Response Value</td>
<td>3 &gt;1E–5</td>
</tr>
<tr>
<td>Response Value</td>
<td>4 &gt;1E–6</td>
</tr>
<tr>
<td>Response Value</td>
<td>5 &gt;1E–7</td>
</tr>
<tr>
<td>Response Value</td>
<td>6 &gt;1E–8</td>
</tr>
<tr>
<td>Response Value</td>
<td>7 &gt;1E–9</td>
</tr>
<tr>
<td>Function</td>
<td>Sets the logging error rate threshold value.</td>
</tr>
<tr>
<td>Example</td>
<td>To set the logging error rate threshold value to 0:</td>
</tr>
<tr>
<td>Example</td>
<td>&gt; LCT 0</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Incompatible with existing models.</td>
</tr>
</tbody>
</table>

### LCT?

| Response        | <thre> = <NR1 NUMERIC RESPONSE DATA> (FIX1) |
| Function        | Queries the logging error rate threshold value. |
| Example         | > LCT?                                  |
| Example         | < LCT 0                                 |
| Compatibility   | Incompatible with existing models.      |
**Chapter 8  Native Commands**

### LCS <numeric>

**Parameter**

<numeric> = <DECIMAL NUMERIC PROGRAM DATA>

- 0  Memory squelch function ON
- 1  Memory squelch function OFF

**Function**

Sets the logging memory squelch function ON/OFF.

**Example**

To set the logging memory squelch function ON:

> LCS 0

**Compatibility**

Incompatible with existing models.

### LCS?

**Response**

<numeric> = <NR1 NUMERIC RESPONSE DATA> (FIX1)

**Function**

Queries the logging memory squelch function ON/OFF state.

**Example**

> LCS?

< LCS 0

**Compatibility**

Incompatible with existing models.

### LAS <all>

**Parameter**

<all> = <DECIMAL NUMERIC PROGRAM DATA>

- 0  Reset
- 1  All Set

**Function**

Selects or reset all logging measurement items.

**Example**

To reset all logging measurement items:

> LAS 0

**Compatibility**

Incompatible with existing models.
8.4.1.4 Histogram setting commands

**Figure 8.4.1.4-1  Histogram setting**

**Table 8.4.1.4-1  Histogram setting commands**

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1]</td>
<td>Histogram ON/OFF</td>
<td>HON, HON?</td>
</tr>
<tr>
<td>[2]</td>
<td>Scale setting</td>
<td>HAL, HAL?</td>
</tr>
<tr>
<td>[3]</td>
<td>Measurement item setting</td>
<td>HTP, HTP?</td>
</tr>
<tr>
<td>[4]</td>
<td>Display time setting</td>
<td>HTS, HTS?</td>
</tr>
<tr>
<td>[5]</td>
<td>Marker</td>
<td>HMK, HMK?</td>
</tr>
<tr>
<td>[6]</td>
<td>Resolution Time</td>
<td>HRS, HRS?</td>
</tr>
<tr>
<td>[7]</td>
<td>Display Time</td>
<td>HDT, HDT?</td>
</tr>
<tr>
<td>[8]</td>
<td>Error Detect</td>
<td>HED, HED?</td>
</tr>
<tr>
<td>[9]</td>
<td>Query for measurement result</td>
<td>HIS, HIS?</td>
</tr>
</tbody>
</table>
### HON <numeric>

**Parameter**

| 0 | OFF |
| 1 | ON  |

**Function**

Sets histogram execution ON/OFF.

**Example**

To set histogram execution ON:

> HON 1

**Compatibility**

Incompatible with existing models.

### HON?

**Response**

<numeric> = <NR1 NUMERIC RESPONSE DATA> (FIX1)

**Function**

Queries the histogram execution ON/OFF state.

**Example**

> HON?

< HON 1

**Compatibility**

Incompatible with existing models.

### HAL <top>,<bottom>

**Parameter**

- **<top> = <DECIMAL NUMERIC PROGRAM DATA>**
  - When the measurement item is Error Rate:
    - 18 to 0: E–18 to E+0 (Step: 1)
  - When the measurement item is Error Count • EI:
    - 1 to 18: E+1, E+2, … E+18 (Step: 1)
- **<bottom> = <DECIMAL NUMERIC PROGRAM DATA>**
  - When the measurement item is Error Rate:
    - 18 to 0: E+0 to E–18 (Step: 1)
  - (However <top> must be greater than <bottom>)
  - When the measurement item is Error Rate • EI:
    - 0 to 17: E+0, E+1, E+2, … to E+17 (Step: 1)
  - (However <top> must be greater than <bottom>)

**Function**

Sets the vertical axis scale of the histogram.

**Example**

To set the vertical axis scale of the histogram as top: E–3 and bottom: E–7:

> HAL 3, 7

**Compatibility**

Incompatible with existing models.
**8.4 ED Commands**

### HAL?

| Response | \(<\text{top}\> = \langle\text{NR1 NUMERIC RESPONSE DATA}\rangle \ (\text{FIX2})\)  
|          | \(<\text{bottom}\> = \langle\text{NR1 NUMERIC RESPONSE DATA}\rangle \ (\text{FIX2})\) |
| Function | Queries the vertical axis scale of the histogram. |
| Example  | \(> \text{HAL}\)?  
|          | \(< \text{HAL 5, 0}\)  
|          | \(< \text{HAL 19, 17}\) |
| Compatibility | Incompatible with existing models. |

### HTP <type>

| Parameter | \(<\text{type}\> = \langle\text{DECIMAL NUMERIC PROGRAM DATA}\rangle\)  
|           | 0 Counts errors.  
|           | 1 Counts error rate.  
|           | 2 Counts error interval. |
| Function  | Sets the measurement item counting type displayed in the histogram. |
| Example   | To set the measurement item counting type displayed in the histogram to  
|           | 0 (counting errors):  
|           | \(> \text{HTP 0}\) |
| Compatibility | Incompatible with existing models. |

### HTP?

| Response | \(<\text{type}\> = \langle\text{NR1 NUMERIC RESPONSE DATA}\rangle \ (\text{FIX1})\) |
| Function | Queries the measurement item counting type displayed in the histogram. |
| Example  | \(> \text{HTP}\)?  
|          | \(< \text{HTP 0}\) |
| Compatibility | Incompatible with existing models. |
HTS <day>,<hour>,<min>,<second>

**Parameter**
- `<day>` = <DECIMAL NUMERIC PROGRAM DATA>
  
  0 to 99  0 to 99 days, in 1-day steps

- `<hour>` = <DECIMAL NUMERIC PROGRAM DATA>
  
  0 to 23  0 to 23 hours, in 1-hour steps

- `<min>` = <DECIMAL NUMERIC PROGRAM DATA>
  
  0 to 59  0 to 59 minutes, in 1-minute steps

- `<second>` = <DECIMAL NUMERIC PROGRAM DATA>
  
  0 to 59  0 to 59 seconds, in 1-second steps

**Function**
Sets the histogram display time.

**Example**
To set the histogram display time to "0 days, 3 hours, 59 minutes, 59 seconds":

```
> HTS 0,3,59,59
```

**Compatibility**
Incompatible with existing models.

---

HTS?

**Response**
- `<day>` = <NR1 NUMERIC RESPONSE DATA> (FIX2)
  
  00 to 99  0 to 99 days

- `<hour>` = < NR1 NUMERIC RESPONSE DATA> (FIX2)
  
  00 to 23  0 to 23 hours

- `<min>` = < NR1 NUMERIC RESPONSE DATA> (FIX2)
  
  00 to 59  0 to 59 minutes

- `<second>` = <NR1 NUMERIC RESPONSE DATA> (FIX2)
  
  00 to 59  0 to 59 seconds

**Note:**
The minimum setting value is 00, 00, 00, 30.

**Function**
Queries the histogram display time.

**Example**
```
> HTS?
< HTS 00,03,59,59
```

**Compatibility**
Incompatible with existing models.
8.4  ED Commands

**HMK <position>**

Parameter  

\[ <\text{position}> = \langle\text{DECIMAL NUMERIC PROGRAM DATA}\rangle \]

0  Searches for error alarms before the current marker position.

1  Searches for error alarms after the current marker position.

Function  

Searches for error alarms before/after the current marker position.

Example  

To search for error alarms after the current marker position:

\[ > \text{HMK 1} \]

Compatibility  

Incompatible with existing models.

**HMK?**

Response  

\[ <\text{day}> = \langle\text{NR1 NUMERIC RESPONSE DATA}\rangle \text{ (FIX2)} \]

00 to 99  0 to 99 days

\[ <\text{hour}> = \langle\text{NR1 NUMERIC RESPONSE DATA}\rangle \text{ (FIX2)} \]

00 to 23  0 to 23 hours

\[ <\text{min}> = \langle\text{NR1 NUMERIC RESPONSE DATA}\rangle \text{ (FIX2)} \]

00 to 59  0 to 59 minutes

\[ <\text{second}> = \langle\text{NR1 NUMERIC RESPONSE DATA}\rangle \text{ (FIX2)} \]

00 to 59  0 to 59 seconds

"--" is displayed if there is no alarm and error.

\[ <\text{alarm}> = \langle\text{NR1 NUMERIC RESPONSE DATA}\rangle \text{ (FIX1)} \]

0  Clock Loss

1  Sync Loss

2  CR Unlock

3  No alarm

\[ <\text{result}> = \langle\text{STRING RESPONSE DATA}\rangle \]

When the measurement item is Error Count or Error Interval, the response is returned in Form1.

When the measurement item is Error Rate, the response is returned in Form2.

**Table 8.4.1.4-2  Response format**

<table>
<thead>
<tr>
<th>Form</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Form1</td>
<td>X.XXXXXX</td>
<td>For 0 to 9999999</td>
</tr>
<tr>
<td>Integer type</td>
<td>X.XXXE.XXX</td>
<td>For 1.0000E07 to 9.9999E17</td>
</tr>
<tr>
<td></td>
<td>1.0000E-99</td>
<td>When no data corresponds to the query</td>
</tr>
<tr>
<td>Form2</td>
<td>X.XXXE-XX</td>
<td>For 0.0001E-18 to 1.0000E00</td>
</tr>
<tr>
<td>Fraction type</td>
<td>1.0000E-99</td>
<td>When no data corresponds to the query</td>
</tr>
</tbody>
</table>

Function  

Queries the marker setting time, alarm, and error rate.

Example  

\[ > \text{HMK?} \]

\[ < \text{HMK 00, 01, 30, 30, 3, 1.0000E07} \]

Compatibility  

Incompatible with existing models.
**Chapter 8  Native Commands**

### HRS <numeric>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;numeric&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1 s</td>
</tr>
<tr>
<td>1</td>
<td>10 s</td>
</tr>
<tr>
<td>2</td>
<td>30 s</td>
</tr>
<tr>
<td>3</td>
<td>1 min</td>
</tr>
<tr>
<td>4</td>
<td>10 min</td>
</tr>
<tr>
<td>5</td>
<td>30 min</td>
</tr>
<tr>
<td>6</td>
<td>1 hour</td>
</tr>
</tbody>
</table>

**Function**
Sets the histogram calculation resolution.

**Example**
To set the histogram calculation resolution to 1 second:
> HRS 0

**Compatibility**
Incompatible with existing models.

### HRS?

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;numeric&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt; (FIX1)</th>
</tr>
</thead>
</table>

**Function**
Queries the histogram calculation resolution.

**Example**
> HRS?
< HRS 0

**Compatibility**
Incompatible with existing models.

### HDT <numeric>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;numeric&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1 s</td>
</tr>
<tr>
<td>1</td>
<td>10 s</td>
</tr>
<tr>
<td>2</td>
<td>30 s</td>
</tr>
<tr>
<td>3</td>
<td>1 min</td>
</tr>
<tr>
<td>4</td>
<td>10 min</td>
</tr>
<tr>
<td>5</td>
<td>30 min</td>
</tr>
<tr>
<td>6</td>
<td>1 hour</td>
</tr>
</tbody>
</table>

**Function**
Sets the histogram display resolution.

**Example**
To set the histogram display resolution to 1 second:
> HDT 0

**Compatibility**
Incompatible with existing models.
### HDT?

**Response**

\[ \text{<numeric>} = \text{<NR1 NUMERIC RESPONSE DATA>} \text{ (FIX1)} \]

**Function**

Queries the histogram display resolution.

**Example**

\[
\begin{align*}
> \text{HDT} \? \\
< \text{HDT} \ 0
\end{align*}
\]

**Compatibility**

Incompatible with existing models.

---

### HED <mode>

**Parameter**

\[ \text{<mode>} = \text{<DECIMAL NUMERIC PROGRAM DATA>} \]

- 0: Total
- 1: INS
- 2: OMI
- 3: Transition
- 4: Non Transition

**Function**

Sets the bit error alarm measurement method and error detection method for histogram calculation.

**Example**

To set the bit error alarm measurement method and error detection method for histogram calculation to Total:

\[
\begin{align*}
> \text{HED} \ 0
\end{align*}
\]

**Compatibility**

Incompatible with existing models.

---

### HED?

**Response**

\[ \text{<mode>} = \text{<NR1 NUMERIC RESPONSE DATA>} \text{ (FIX1)} \]

**Function**

Queries the bit error alarm measurement method and error detection method for histogram calculation.

**Example**

\[
\begin{align*}
> \text{HED} \? \\
< \text{HED} \ 0
\end{align*}
\]

**Compatibility**

Incompatible with existing models.
Chapter 8  Native Commands

HIS? <day>,<hour>,<min>,<second>

Parameter
<day> = <DECIMAL NUMERIC PROGRAM DATA>
0 to 99 0 to 99 days, in 1-day steps
<hour> = <DECIMAL NUMERIC PROGRAM DATA>
0 to 23 0 to 23 hours, in 1-hour steps
<min> = <DECIMAL NUMERIC PROGRAM DATA>
0 to 59 0 to 59 minutes, in 1-minute steps
<second> = <DECIMAL NUMERIC PROGRAM DATA>
0 to 59 0 to 59 seconds, in 1-second steps

Response
<alarm> = <NR1 NUMERIC RESPONSE DATA> (FIX1)
0 Clock Loss
1 Sync Loss
2 CR Unlock
3 No alarm
<count> = <STRING RESPONSE DATA>
When the measurement item is Error Count, the response is returned in Form1.
<brate> = <STRING RESPONSE DATA>
When the measurement item is Error Rate, the response is returned in Form2.

<table>
<thead>
<tr>
<th>Form</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Form1</td>
<td>XXXXXXXX</td>
<td>For 0 to 9999999</td>
</tr>
<tr>
<td>Integer type</td>
<td>X.XXXXXXX</td>
<td>For 1.0000E07 to 9.9999E17</td>
</tr>
<tr>
<td></td>
<td>1.0000E-99</td>
<td>When no data corresponds to the query</td>
</tr>
<tr>
<td>Form2</td>
<td>X.XXXXXE-XX</td>
<td>For 0.0001E–18 to 1.0000E00</td>
</tr>
<tr>
<td>Fraction type</td>
<td>1.0000E-99</td>
<td>When no data corresponds to the query</td>
</tr>
</tbody>
</table>

Function
Queries the measurement result item on the histogram corresponding to the parameter.

Example
To query the histogram measurement result for "0 days, 0 hours, 0 minutes, 0 seconds":
> HIS? 0,0,0,1
< HIS 3,0,0.0001E-18

Compatibility
Incompatible with existing models.
8.4.2 Commands related to Measurement tab

Figure 8.4.2-1 Measurement tab
8.4.2.1 Gating setting commands

Figure 8.4.2.1-1 Gating setting (Time)

Figure 8.4.2.1-2 Gating setting (Clock Count)

Figure 8.4.2.1-3 Gating setting (Error Count)

Figure 8.4.2.1-4 Gating setting (Block Count)
<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>[1]</td>
<td>Unit</td>
<td>UNT</td>
<td>UNT?</td>
</tr>
<tr>
<td>[2]</td>
<td>No label (Measurement time setting)</td>
<td>PRD</td>
<td>PRD?</td>
</tr>
<tr>
<td>[3]</td>
<td>No label (Clock count setting)</td>
<td>CLC</td>
<td>CLC?</td>
</tr>
<tr>
<td>[4]</td>
<td>No label (Error count setting)</td>
<td>ERC</td>
<td>ERC?</td>
</tr>
<tr>
<td>[5]</td>
<td>No label (Block count setting)</td>
<td>BLC</td>
<td>BLC?</td>
</tr>
<tr>
<td>[6]</td>
<td>Cycle</td>
<td>MOD</td>
<td>MOD?</td>
</tr>
<tr>
<td>[7]</td>
<td>Current</td>
<td>CUR</td>
<td>CUR?</td>
</tr>
<tr>
<td>[8]</td>
<td>Calculation</td>
<td>CAL</td>
<td>CAL?</td>
</tr>
<tr>
<td>[9]</td>
<td>Interval</td>
<td>ITV</td>
<td>ITV?</td>
</tr>
</tbody>
</table>
### Chapter 8  Native Commands

#### UNI <unit>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;unit&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Time</td>
</tr>
<tr>
<td>1</td>
<td>Clock Count</td>
</tr>
<tr>
<td>2</td>
<td>Error Count</td>
</tr>
<tr>
<td>3</td>
<td>Block Count</td>
</tr>
</tbody>
</table>

**Function**
Sets the measurement period unit.

**Example**
To set the measurement period unit to Time:
```
> UNI 0
```

**Compatibility**
Incompatible with existing models.

#### UNI?

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;unit&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt; (FIX1)</th>
</tr>
</thead>
</table>

**Function**
Queries the measurement period unit.

**Example**
```
> UNI?
< UNI 0
```

**Compatibility**
Incompatible with existing models.

#### PRD <day>,<hour>,<min>,<second>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;day&gt;,&lt;hour&gt;,&lt;min&gt;,&lt;second&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;day&gt;</td>
<td>0 to 99  0 to 99 days, in 1-day steps</td>
</tr>
<tr>
<td>&lt;hour&gt;</td>
<td>0 to 23  0 to 23 hours, in 1-hour steps</td>
</tr>
<tr>
<td>&lt;min&gt;</td>
<td>0 to 59  0 to 59 minutes, in 1-minute steps</td>
</tr>
<tr>
<td>&lt;second&gt;</td>
<td>0 to 59  0 to 59 seconds, in 1-second steps</td>
</tr>
</tbody>
</table>

(Not all values can be set to 00 simultaneously.)

**Function**
Sets the measurement time when the measurement period is set to Time.

**Example**
To set the measurement time when the measurement period is set to Time, to "0 days, 1 hour, 0 minutes, 0 seconds":
```
> PRD 0,1,0,0
```

**Compatibility**
Compatible with the MP1632C and MP1762D.
**PRD?**

**Response**

\[<\text{day}>,<\text{hour}>,<\text{min}>,<\text{second}> = <\text{NR1 NUMERIC RESPONSE DATA}>\]  
(FIX2)

- **<day>**: 00 to 99  (0 to 99 days)
- **<hour>**: 00 to 23  (0 to 23 hours)
- **<min>**: 00 to 59  (0 to 59 minutes)
- **<second>**: 00 to 59  (0 to 59 seconds)

**Function**

Queries the measurement time when the measurement period is set to Time.

**Example**

To query the measurement time:

```plaintext
> PRD?
< PRD 00,01,00,00
```

**Compatibility**

Compatible with the MP1632C and MP1762D.

---

**CLC<count>**

**Parameter**

\[<\text{count}> = <\text{DECIMAL NUMERIC PROGRAM DATA}>\]

- **0**: \(>1\times10^4\)
- **1**: \(>1\times10^5\)
- **2**: \(>1\times10^6\)
- **3**: \(>1\times10^7\)
- **4**: \(>1\times10^8\)
- **5**: \(>1\times10^9\)
- **6**: \(>1\times10^{10}\)
- **7**: \(>1\times10^{11}\)
- **8**: \(>1\times10^{12}\)
- **9**: \(>1\times10^{13}\)
- **10**: \(>1\times10^{14}\)
- **11**: \(>1\times10^{15}\)
- **12**: \(>1\times10^{16}\)

**Function**

Sets the clock count threshold when the measurement period is set to Clock Count.

**Example**

To set the clock count threshold when the measurement period is set to Clock Count to \(1\times10^4\):

```plaintext
> CLC 0
```

**Compatibility**

Incompatible with existing models.
Chapter 8  Native Commands

CLC?
Response  \(<\text{count}> = \text{NR1 NUMERIC RESPONSE DATA} \) (FIX2)
Function  Queries the clock count threshold when the measurement period is set to Clock Count.
Example
> CLC?
< CLC  0
< CLC  12
Compatibility  Incompatible with existing models.

ERC <count>
Parameter  \(<\text{count}> = \text{DECIMAL NUMERIC PROGRAM DATA} \)
0  \(>1\text{E}+4\)
1  \(>1\text{E}+5\)
2  \(>1\text{E}+6\)
3  \(>1\text{E}+7\)
4  \(>1\text{E}+8\)
5  \(>1\text{E}+9\)
6  \(>1\text{E}+10\)
7  \(>1\text{E}+11\)
8  \(>1\text{E}+12\)
9  \(>1\text{E}+13\)
10  \(>1\text{E}+14\)
11  \(>1\text{E}+15\)
12  \(>1\text{E}+16\)
Function  Sets the error count threshold when the measurement period is set to Error Count.
Example
To set the error count threshold when the measurement period is set to Error Count to \(1\text{E}+4\):
> ERC 0
Compatibility  Incompatible with existing models.

ERC?
Response  \(<\text{count}> = \text{NR1 NUMERIC RESPONSE DATA} \) (FIX2)
Function  Queries the error count threshold when the measurement period is set to Error Count.
Example
> ERC?
< ERC  0
< ERC  12
Compatibility  Incompatible with existing models.
**BLC <count>**

Parameter

<table>
<thead>
<tr>
<th>&lt;count&gt;</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>&gt;1E+2</td>
</tr>
<tr>
<td>1</td>
<td>&gt;1E+3</td>
</tr>
<tr>
<td>2</td>
<td>&gt;1E+4</td>
</tr>
<tr>
<td>3</td>
<td>&gt;1E+5</td>
</tr>
<tr>
<td>4</td>
<td>&gt;1E+6</td>
</tr>
<tr>
<td>5</td>
<td>&gt;1E+7</td>
</tr>
<tr>
<td>6</td>
<td>&gt;1E+8</td>
</tr>
<tr>
<td>7</td>
<td>&gt;1E+9</td>
</tr>
<tr>
<td>8</td>
<td>&gt;1E+10</td>
</tr>
<tr>
<td>9</td>
<td>&gt;1E+11</td>
</tr>
<tr>
<td>10</td>
<td>&gt;1E+12</td>
</tr>
<tr>
<td>11</td>
<td>&gt;1E+13</td>
</tr>
<tr>
<td>12</td>
<td>&gt;1E+14</td>
</tr>
</tbody>
</table>

Function

Sets the block count threshold when the measurement period is set to Block Count.

Example

To set the block count threshold when the measurement period is set to Block Count to 1E+2:

```plaintext
> BLC 0
```

Compatibility

Incompatible with existing models.

**BLC?**

Response

```plaintext
<count> = <NR1 NUMERIC RESPONSE DATA> (FIX2)
```

Function

Queries the block count threshold when the measurement period is set to Block Count.

Example

```plaintext
> BLC?
< BLC 0
< BLC 12
```

Compatibility

Incompatible with existing models.

**MOD <mode>**

Parameter

<table>
<thead>
<tr>
<th>&lt;mode&gt;</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Repeat</td>
</tr>
<tr>
<td>1</td>
<td>Single</td>
</tr>
<tr>
<td>2</td>
<td>Untimed</td>
</tr>
</tbody>
</table>

Function

Sets the bit error alarm measurement processing mode.

Example

To set the bit error alarm measurement processing mode to Repeat (performing measurement repeatedly):

```plaintext
> MOD 0
```

Compatibility

Compatible with the MP1632C and MP1762D.
Chapter 8  Native Commands

MOD?
Response  <mode> = <NR1 NUMERIC RESPONSE DATA> (FIX1)
Function  Queries the bit error alarm measurement processing mode.
Example  > MOD?
< MOD 0
Compatibility  Compatible with the MP1632C and MP1762D.

CUR <numeric>
Parameter  <numeric> = <DECIMAL NUMERIC PROGRAM DATA>
0  Current OFF
1  Current ON
Function  Sets measurement result data display immediate update ON/OFF.
Example  To set measurement result data display immediate update ON.
> CUR 1
Compatibility  Compatible with the MP1632C and MP1762D.

CUR?
Response  <numeric> = <NR1 NUMERIC RESPONSE DATA> (FIX1)
Function  Queries the measurement result data display immediate update ON/OFF state.
Example  > CUR?
< CUR 1
Compatibility  Compatible with the MP1632C and MP1762D.

CAL<mode>
Parameter  <mode> = <DECIMAL NUMERIC PROGRAM DATA>
0  Progressive accumulation
1  Immediate processing
Function  Sets the immediate data calculation method.
Example  To set the immediate data calculation method to progressive accumulation:
> CAL 0
Compatibility  Compatible with the MP1762D.
**CAL?**

**Response**

<mode> = <NR1 NUMERIC RESPONSE DATA> (FIX1)

**Function**

Queries the immediate data calculation method.

**Example**

> CAL?
< CAL 0

**Compatibility**

Compatible with the MP1762D.

---

**ITV <numeric>**

**Parameter**

<numeric> = <DECIMAL NUMERIC PROGRAM DATA>

0 Updates display in 100 ms period.
1 Updates display in 200 ms period.

**Function**

Sets the immediate data display update period.

**Example**

To set the immediate data display update period to 100 ms:

> ITV 0

**Compatibility**

Partially compatible with the MP1632C (100 ms setting supported, 200 ms setting not supported).

Compatible with the MP1762D.

---

**ITV?**

**Response**

<numeric> = <NR1 NUMERIC RESPONSE DATA> (FIX1)

**Function**

Queries the immediate data display update period.

**Example**

> ITV?
< ITV 0

**Compatibility**

Partially compatible with the MP1632C (100 ms setting supported, 200 ms setting not supported).

Compatible with the MP1762D.
8.4.2.2 Auto Sync setting commands

![Figure 8.4.2.2-1 Auto Sync setting](image)

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1]</td>
<td>Auto Sync</td>
<td>SYN</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SYN?</td>
</tr>
<tr>
<td>[2]</td>
<td>Threshold Gain</td>
<td>SYE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SYE?</td>
</tr>
</tbody>
</table>
## SYN <numeric>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Example</th>
<th>Compatibility</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;numeric&gt;</td>
<td>= &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>Auto Sync OFF</td>
<td>&gt; SYN 1</td>
<td>Compatible with MP1632C and MP1762D.</td>
</tr>
<tr>
<td>1</td>
<td>Auto Sync ON</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Function**
Sets whether to automatically perform resynchronization (Auto Sync) when the synchronization threshold is exceeded.

**Example**
To enable automatic resynchronization (Auto Sync ON):
> SYN 1

**Compatibility**
Compatible with the MP1632C and MP1762D.

## SYN?

<table>
<thead>
<tr>
<th>Response</th>
<th>= &lt;NR1 NUMERIC RESPONSE DATA&gt; (FIX1)</th>
<th>Example</th>
<th>Compatibility</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;numeric&gt;</td>
<td></td>
<td>&gt; SYN?</td>
<td></td>
</tr>
</tbody>
</table>

**Function**
Queries the Auto Sync ON/OFF state.

**Example**
> SYN?
< SYN 1

**Compatibility**
Compatible with the MP1632C and MP1762D.

## SYE <thre>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>= &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</th>
<th>Example</th>
<th>Compatibility</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;thre&gt;</td>
<td>= &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>E–2</td>
<td>&gt; SYE 0</td>
<td>Compatible with MP1762D.</td>
</tr>
<tr>
<td>1</td>
<td>E–3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>E–4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>E–5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>E–6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>E–7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>E–8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Internal</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Function**
Sets the synchronization detection threshold for resynchronization.

**Example**
To set the synchronization detection threshold for resynchronization E–2:
> SYE 0

**Compatibility**
Compatible with the MP1762D.

## SYE?

<table>
<thead>
<tr>
<th>Response</th>
<th>= &lt;NR1 NUMERIC RESPONSE DATA&gt; (FIX1)</th>
<th>Example</th>
<th>Compatibility</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;thre&gt;</td>
<td>= &lt;NR1 NUMERIC RESPONSE DATA&gt; (FIX1)</td>
<td>&gt; SYE?</td>
<td></td>
</tr>
</tbody>
</table>

**Function**
Queries the synchronization detection threshold for resynchronization.

**Example**
> SYE?
< SYE 0

**Compatibility**
Compatible with the MP1762D.
Chapter 8  Native Commands

8.4.2.3  Sync Control setting commands

Figure 8.4.2.3-1  Sync Control setting commands

Figure 8.4.2.3-2  Sync Control setting (Pattern Editor screen)
### Table 8.4.2.3-1  Sync Control setting commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1]</td>
<td>Control</td>
<td>SYM</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SYM?</td>
</tr>
<tr>
<td>[2]</td>
<td>Frame Length</td>
<td>FLN</td>
</tr>
<tr>
<td></td>
<td></td>
<td>FLN?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>FPS?</td>
</tr>
<tr>
<td>[4]</td>
<td>Mask (Setting of mask pattern)</td>
<td>PDW</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PDW?</td>
</tr>
<tr>
<td>[5]</td>
<td>No label (Data inversion)</td>
<td>PSA</td>
</tr>
<tr>
<td>[6]</td>
<td>No label (All 0s or 1s)</td>
<td>PFL</td>
</tr>
</tbody>
</table>

#### SYM <mode>

- **Parameter**: <mode> = <DECIMAL NUMERIC PROGRAM DATA>
  - 0: Frame detection OFF
  - 1: Frame detection ON
  - 2: Quick synchronizing mode
  - 3: Fast synchronizing mode

- **Function**: Sets the synchronization mode for the test pattern.

- **Example**: To set the synchronization mode for the test pattern to frame detection off:
  ```
  > SYM 0
  ```

- **Compatibility**: Compatible with the MP1762D.

#### SYM?

- **Response**: <mode> = <NR1 NUMERIC RESPONSE DATA> (FIX1)

- **Function**: Queries the synchronization mode for the test pattern.

- **Example**: To query the synchronization mode for the test pattern:
  ```
  > SYM?
  < SYM 0
  ```

- **Compatibility**: Compatible with the MP1762D.
**Chapter 8  Native Commands**

### FLN <length>

**Parameter**

\(<\text{length}> = \langle\text{DECIMAL NUMERIC PROGRAM DATA}\rangle\)

- 4 to 64 bits, in 4-bit steps

**Function**

Sets the unique pattern length when the synchronization mode is set to frame detection ON.

**Example**

To set the unique pattern length to 4 bits when the synchronization mode is set to frame detection ON:

\(> \text{FLN 4}\)

**Compatibility**

Compatible with the MP1632C and MP1762D.

### FLN? 

**Response**

\(<\text{length}> = \langle\text{NR1 NUMERIC RESPONSE DATA}\rangle\) (FIX3)

**Function**

Queries the unique pattern length when the synchronization mode is set to frame detection ON.

**Example**

\(> \text{FLN}\?

\(< \text{FLN 4}\)

\(< \text{FLN 64}\)

**Compatibility**

Compatible with the MP1632C and MP1762D.

### FPS <length>

**Parameter**

\(<\text{length}> = \langle\text{DECIMAL NUMERIC PROGRAM DATA}\rangle\)

- 1 to 134217725 bits, in 64-bit steps

The maximum setting value is "Pattern length – Frame length + 1 bit".

In the case of Combination, the setting range and resolution (steps) are multiplied by 4.

**Function**

Sets the frame synchronization start position when the synchronization mode is set to 1 (frame detection ON).

**Example**

To set the frame synchronization start position when the synchronization mode is set to 1 (frame detection ON) to bit 1:

\(> \text{FPS 1}\)

**Compatibility**

Incompatible with existing models.

### FPS? 

**Response**

\(<\text{length}> = \langle\text{NR1 NUMERIC RESPONSE DATA}\rangle\)

**Function**

Queries the frame synchronization start position when the synchronization mode is set to 1 (frame detection ON).

**Example**

\(> \text{FPS}\?

\(< \text{FPS 1}\)

\(< \text{FPS 134217665}\)

**Compatibility**

Incompatible with existing models.
### PDW <data>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;data&gt;</td>
<td>= &lt;NON-DECIMAL PROGRAM DATA&gt;</td>
</tr>
<tr>
<td>#H***...*</td>
<td>1 to 16 characters</td>
</tr>
<tr>
<td></td>
<td>(Specify pattern data in hexadecimal format)</td>
</tr>
<tr>
<td>#B***...*</td>
<td>1 to 64 characters</td>
</tr>
<tr>
<td></td>
<td>(Specify pattern data in binary format)</td>
</tr>
</tbody>
</table>

**Note:**
The maximum setting value is the pattern length.

**Function**
Edits the mask pattern when the synchronization mode is set to 1 (frame detection ON).

Up to 512-bit data can be edited in hexadecimal or binary format.

**Example**
To set the mask pattern when the synchronization mode is set to 1 (frame detection ON) in hexadecimal format:

> PDW #H001

**Compatibility**
Incompatible with existing models.

### PDW?

<table>
<thead>
<tr>
<th>Response</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;data&gt;</td>
<td>= &lt;HEXADECIMAL NUMERIC RESPONSE DATA&gt;</td>
</tr>
<tr>
<td>#H***...*</td>
<td>1 to 16 characters</td>
</tr>
<tr>
<td></td>
<td>(Specify pattern data in hexadecimal format)</td>
</tr>
</tbody>
</table>

Hexadecimal data maximum of 64 bits is returned.

**Function**
Queries the mask pattern when the synchronization mode is set to frame detection ON.

**Example**
> PDW?

< PDW #H001

**Compatibility**
Incompatible with existing models.
**PSA <start>,<end>**

**Parameter**

- `<start> = <NON-DECIMAL PROGRAM DATA>`
- `#H0 to #H3F` 0 to 3F bits, in 1-bit steps
  (Specify in hexadecimal)
- `<end> = <NON-DECIMAL PROGRAM DATA>`
- `#H0 to #H3F` 0 to 3F bits, in 1-bit steps
  (Specify in hexadecimal)

**Function**

Inverts the mask pattern data when the synchronization mode is set to frame detection ON.

Specify the inversion range by the start and end addresses.

**Example**

To invert the mask pattern data from bit 0 to bit 3F addresses when the synchronization mode is set to 1 (frame detection ON):

```
> PSA #H0,#H3F
```

**Compatibility**

Incompatible with existing models.

---

**PFL <start>,<page>,<data>**

**Parameter**

- `<start> = <NON-DECIMAL PROGRAM DATA>`
- `#H0 to #H3F` 0 to 3F bits, in 1-bit steps
  (Specify in hexadecimal)
- `<end> = <NON-DECIMAL PROGRAM DATA>`
- `#H0 to #H3F` 0 to 3F bits, in 1-bit steps
  (Specify in hexadecimal)
- `<data> = <DECIMAL NUMERIC PROGRAM DATA>`
- `0` Fills the specified range with 0s.
- `1` Fills the specified range with 1s.

**Function**

Sets 0s or 1s for the specified range of the mask pattern data when the synchronization mode is set to 1 (frame detection ON). Specify the range with the start and end addresses.

**Example**

To set 0s for bit 0 to bit 3F of the mask pattern when the synchronization mode is set to 1 (frame detection ON):

```
> PFL #H0,#H3F,0
```

**Compatibility**

Incompatible with existing models.
8.4.2.4 Error/Alarm Condition setting commands

Table 8.4.2.4-1 Error/Alarm Condition setting commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1]</td>
<td>Error Detect</td>
<td>ERD</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ERD?</td>
</tr>
<tr>
<td>[2]</td>
<td>EI/EFI Interval</td>
<td>EIT</td>
</tr>
<tr>
<td></td>
<td></td>
<td>EIT?</td>
</tr>
<tr>
<td>[3]</td>
<td>SES Threshold</td>
<td>ETH</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ETH?</td>
</tr>
<tr>
<td>[4]</td>
<td>Clock Loss</td>
<td>CLS</td>
</tr>
<tr>
<td></td>
<td>Evaluation</td>
<td>CLS?</td>
</tr>
<tr>
<td>[5]</td>
<td>CR Unlock</td>
<td>CRU</td>
</tr>
<tr>
<td></td>
<td>Evaluation</td>
<td>CRU?</td>
</tr>
<tr>
<td>[6]</td>
<td>Sync Loss</td>
<td>SLS</td>
</tr>
<tr>
<td></td>
<td>Evaluation</td>
<td>SLS?</td>
</tr>
</tbody>
</table>
## ERD <mode>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;mode&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>INS/OMI</td>
</tr>
<tr>
<td>1</td>
<td>Transition/Non Transition</td>
</tr>
</tbody>
</table>

**Function**

Sets the error detection method during bit error alarm measurement.

**Example**

To set the error detection method during bit error alarm measurement to INS/OMI:

```plaintext
> ERD 0
```

**Compatibility**

Incompatible with existing models.

## ERD?

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;mode&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt; (FIX1)</th>
</tr>
</thead>
</table>

**Function**

Queries the error detection method during bit error alarm measurement.

**Example**

```plaintext
> ERD?
< ERD 0
```

**Compatibility**

Incompatible with existing models.

## EIT <numeric>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;numeric&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1 ms</td>
</tr>
<tr>
<td>1</td>
<td>10 ms</td>
</tr>
<tr>
<td>2</td>
<td>100 ms</td>
</tr>
<tr>
<td>3</td>
<td>1 s</td>
</tr>
</tbody>
</table>

**Function**

Sets the interval for EI and %EFI measurement.

**Example**

To set the interval for EI and %EFI measurement to 1 ms:

```plaintext
> EIT 0
```

**Compatibility**

Compatible with the MP1762D.

## EIT?

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;numeric&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt; (FIX1)</th>
</tr>
</thead>
</table>

**Function**

Queries the interval for EI and %EFI measurement.

**Example**

```plaintext
> EIT?
< EIT 0
```

**Compatibility**

Compatible with the MP1762D.
### ETH <thre>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;thre&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>SES: 1E–3, DM: 1E–6</td>
</tr>
<tr>
<td>1</td>
<td>SES: 1E–4, DM: 1E–8</td>
</tr>
</tbody>
</table>

**Function**
Sets the thresholds for SES and DM calculations during performance measurement.

**Example**
To set the thresholds for SES and DM calculations during performance measurement to SES: 1E–3, DM: 1E–6:
> ETH 0

**Compatibility**
Compatible with the MP1632C and MP1762D.

### ETH?

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;thre&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt; (FIX1)</th>
</tr>
</thead>
</table>

**Function**
Queries the thresholds for SES and DM calculations during performance measurement.

**Example**
> ETH?
< ETH 0

**Compatibility**
Compatible with the MP1632C and MP1762D.

### CLS <numeric>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;numeric&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Out of evaluation target</td>
</tr>
<tr>
<td>1</td>
<td>Evaluation target</td>
</tr>
</tbody>
</table>

**Function**
Sets whether to include the Clock Loss occurrence period to the measurement evaluation target.

**Example**
To include the Clock Loss occurrence period to the measurement evaluation target:
> CLS 1

**Compatibility**
Compatible with the MP1632C and MP1762D.

### CLS?

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;numeric&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt; (FIX1)</th>
</tr>
</thead>
</table>

**Function**
Queries whether the Clock Loss occurrence period is included to the measurement evaluation target.

**Example**
> CLS?
< CLS 1

**Compatibility**
Compatible with the MP1632C and MP1762D.
Chapter 8  Native Commands

CRU <numeric>

Parameter  

<numeric> = <DECIMAL NUMERIC PROGRAM DATA>

0  Out of evaluation target
1  Evaluation target

Function  

Sets whether to include the CR Unlock occurrence period to the measurement evaluation target.

Example  

To include the CR Unlock occurrence period to the measurement evaluation target:

> CRU 1

Compatibility  

Incompatible with existing models.

CRU?

Response  

<numeric> = <NR1 NUMERIC RESPONSE DATA> (FIX1)

Function  

Queries whether the CR Unlock occurrence period is included to the measurement evaluation target.

Example  

> CRU?
< CRU 1

Compatibility  

Incompatible with existing models.

SLS <numeric>

Parameter  

<numeric> = <DECIMAL NUMERIC PROGRAM DATA>

0  Out of evaluation target
1  Evaluation target

Function  

Sets whether to include the Sync Loss occurrence period to the measurement evaluation target.

Example  

To include the Sync Loss occurrence period to the measurement evaluation target:

> SLS 1

Compatibility  

Compatible with the MP1632C and MP1762D.

SLS?

Response  

<numeric> = <NR1 NUMERIC RESPONSE DATA> (FIX1)

Function  

Queries whether the Sync Loss occurrence period is included to the measurement evaluation target.

Example  

> SLS?
< SLS 1

Compatibility  

Compatible with the MP1632C and MP1762D.
8.4.3 Commands related to Pattern tab

8.4.3.1 Test Pattern setting commands

Table 8.4.3.1-1 Test Pattern setting commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1]</td>
<td>Test Pattern</td>
<td>PTS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PTS?</td>
</tr>
<tr>
<td>[2]</td>
<td>Logic</td>
<td>LGC</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LGC?</td>
</tr>
</tbody>
</table>
PTS <type>
Parameter
- <type> = <DECIMAL NUMERIC PROGRAM DATA>
- 1: Data pattern
- 2: Zero-Substitution pattern
- 3: PRBS pattern
- 4: Mixed Data pattern
- 6: Sequence pattern

Function
Sets the type of the test pattern.

Example
To set the test pattern type to the Data pattern:

```plaintext
> PTS 1
```

Compatibility
Compatible with the MP1762D.

PTS?
Response
- <type> = <NR1 NUMERIC RESPONSE DATA> (FIX1)

Function
Queries the type of the test pattern.

Example

```plaintext
> PTS?
< PTS 1
```

Compatibility
Compatible with the MP1762D.

LGC <logic>
Parameter
- <logic> = <DECIMAL NUMERIC PROGRAM DATA>
- 0: Positive
- 1: Negative

Function
Sets the logic (positive or negative) of the pattern during pattern reception.

Example
To set the logic of the pattern during pattern reception to the positive logic:

```plaintext
> LGC 0
```

Compatibility
Compatible with the MP1632C and MP1762D.

LGC?
Response
- <logic> = <NR1 NUMERIC RESPONSE DATA> (FIX1)

Function
Queries the logic of the pattern during pattern reception.

Example

```plaintext
> LGC?
< LGC 0
```

Compatibility
Compatible with the MP1632C and MP1762D.
8.4.3.2 PRBS pattern setting commands

![Figure 8.4.3.2-1 PRBS pattern setting](image)

Table 8.4.3.2-1 PRBS pattern setting commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1]</td>
<td>Length</td>
<td>PTN</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PTN?</td>
</tr>
<tr>
<td>[2]</td>
<td>Mark Ratio</td>
<td>MRK</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MRK?</td>
</tr>
<tr>
<td>[3]</td>
<td>Bit Shift</td>
<td>SFT</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SFT?</td>
</tr>
</tbody>
</table>
### PTN <numeric>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;numeric&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PRBS</td>
</tr>
<tr>
<td>2</td>
<td>$2^7-1$</td>
</tr>
<tr>
<td>3</td>
<td>$2^9-1$</td>
</tr>
<tr>
<td>4</td>
<td>$2^{10}-1$</td>
</tr>
<tr>
<td>5</td>
<td>$2^{11}-1$</td>
</tr>
<tr>
<td>6</td>
<td>$2^{15}-1$</td>
</tr>
<tr>
<td>7</td>
<td>$2^{20}-1$</td>
</tr>
<tr>
<td>8</td>
<td>$2^{23}-1$</td>
</tr>
<tr>
<td>9</td>
<td>$2^{31}-1$</td>
</tr>
<tr>
<td></td>
<td>12</td>
</tr>
<tr>
<td></td>
<td>13</td>
</tr>
<tr>
<td></td>
<td>14</td>
</tr>
<tr>
<td></td>
<td>15</td>
</tr>
<tr>
<td></td>
<td>16</td>
</tr>
<tr>
<td></td>
<td>17</td>
</tr>
<tr>
<td></td>
<td>18</td>
</tr>
</tbody>
</table>

**Function**
Sets the number of stages ($2^n-1$ (n = 7, 9, 10, 11, 15, 20, 23, or 31)) during PRBS and Zero Substitution pattern reception.

**Example**
To set the number of stages during PRBS pattern reception to $2^7-1$:

```plaintext
> PTN 2
```

**Compatibility**
Partially compatible with the MP1632C (applies to PRBS only).
Compatible with the MP1762D.

### PTN?

**Response**

<numeric> = <NR1 NUMERIC RESPONSE DATA> (FIX2)

**Function**
Queries the number of stages ($2^n-1$ (n = 7, 9, 10, 11, 15, 20, 23, or 31)) during PRBS and Zero Substitution pattern reception.

**Example**

```plaintext
> PTN?
< PTN 2
< PTN 18
```

**Compatibility**
Partially compatible with the MP1632C (applies to PRBS only).
Compatible with the MP1762D.
### MRK <mratio>

**Parameter**

<mratio> = <DECIMAL NUMERIC PROGRAM DATA>

<table>
<thead>
<tr>
<th>POS</th>
<th>NEG</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0/8</td>
</tr>
<tr>
<td>1</td>
<td>1/8</td>
</tr>
<tr>
<td>2</td>
<td>1/4</td>
</tr>
<tr>
<td>3</td>
<td>1/2</td>
</tr>
</tbody>
</table>

**Function**

Sets the mark ratio during PRBS pattern reception.

**Example**

To set the mark ratio during PRBS pattern reception to 0/8:

> MRK 0

**Compatibility**

Compatible with the MP1632C and MP1762D.

### MRK?

**Response**

<mratio> = <NR1 NUMERIC RESPONSE DATA> (FIX1)

**Function**

Queries the mark ratio during PRBS pattern reception.

**Example**

> MRK?

< MRK 0

**Compatibility**

Compatible with the MP1632C and MP1762D.

### SFT <numeric>

**Parameter**

<numeric> = <DECIMAL NUMERIC PROGRAM DATA>

- 0: 1-bit
- 1: 3-bit

**Function**

Sets the bit shift during PRBS pattern reception.

**Example**

To set the bit shift during PRBS pattern reception to 1-bit shift:

> SFT 0

**Compatibility**

Compatible with the MP1632C and MP1762D.

### SFT?

**Response**

<numeric> = <NR1 NUMERIC RESPONSE DATA> (FIX1)

**Function**

Queries the bit shift during PRBS pattern reception.

**Example**

> SFT?

< SFT 0

**Compatibility**

Compatible with the MP1632C and MP1762D.
8.4.3.3 Zero Substitution pattern setting commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1]</td>
<td>Length</td>
<td>PTN</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Refer to Section 8.4.3.2 &quot;PRBS pattern setting commands&quot; for detail on this command.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PTN?</td>
</tr>
<tr>
<td>[2]</td>
<td>Zero Substitution length</td>
<td>ZLN</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ZLN?</td>
</tr>
<tr>
<td>[3]</td>
<td>Additional Bit</td>
<td>ZAD</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ZAD?</td>
</tr>
</tbody>
</table>

**ZLN <numeric>**

**Parameter**

\(<\text{numeric}> = <\text{DECIMAL NUMERIC PROGRAM DATA}>\)

1 to \(2^n-1\) 1 to \(2^n-1\), in 1-bit steps  
\((\text{when ZSUBlength, } 2n \text{ is set})\)

1 to \(2^n-2\) 1 to \(2^n-2\), in 1-bit steps  
\((\text{when ZSUBlength, } 2^{n-1} \text{ is set})\)

\(n = 7, 9, 10, 11, 15, 20, 23\)

**Function**

Sets the number of bits to which "0" is inserted (zero-insertion bit count) during Zero-Substitution pattern reception.

Available parameters vary depending on the setting conditions.

**Example**

To set the zero-insertion bit count to 1 bit:

\(> \ ZLN \ 1\)

**Compatibility**

Partially compatible with the MP1762D.
8.4 ED Commands

**ZLN?**

**Response**  
\(<\text{numeric}\> = <\text{NR1 NUMERIC RESPONSE DATA}> \ (\text{FIX7})\)  
1 to \(2^{n-1}\)  
\(n = 7, 9, 10, 11, 15, 20, 23\)

**Function**  
Queries the zero-insertion bit count during Zero-Substitution pattern reception.

**Example**  
> ZLN?  
< ZLN 1  
< ZLN 8388607

**Compatibility**  
Partially compatible with the MP1632C and the MP1762D.

**ZAD <numeric>**

**Parameter**  
\(<\text{numeric}\> = <\text{DECIMAL NUMERIC PROGRAM DATA}>\)  
0  
Adds one bit of "0" next to the last of zero-inserted bits.  
1  
Adds one bit of "1" next to the last of zero-inserted bits (same as existing models).

**Function**  
Sets whether to add one bit of "0" or "1" next to the last of zero-inserted bits during Zero-Substitution pattern reception.

**Example**  
To add one bit of "0" next to the last of zero-inserted bits during Zero-Substitution pattern reception:  
> ZAD 0

**Compatibility**  
Incompatible with existing models.

**ZAD?**

**Response**  
\(<\text{numeric}\> = <\text{NR1 NUMERIC RESPONSE DATA}> \ (\text{FIX1})\)

**Function**  
Queries whether one bit of "0" or "1" is added next to the last of zero-inserted bits during Zero-Substitution pattern reception.

**Example**  
> ZAD?  
< ZAD 0

**Compatibility**  
Incompatible with existing models.
8.4.3.4 Data pattern setting commands

![Data pattern setting interface](image1)

Figure 8.4.3.4-1  Data pattern setting

![Data pattern setting (Pattern Editor)](image2)

Figure 8.4.3.4-2  Data pattern setting (Pattern Editor)

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1]</td>
<td>Data Length</td>
<td>DLN</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DLN?</td>
</tr>
<tr>
<td>[2]</td>
<td>No label (Pattern setting)</td>
<td>PDT</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PDT?</td>
</tr>
<tr>
<td>[3]</td>
<td>No label (Pattern setting)</td>
<td>BDT</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BDT?</td>
</tr>
<tr>
<td>[4]</td>
<td>No label (Data inversion)</td>
<td>DRA</td>
</tr>
<tr>
<td>[5]</td>
<td>No label (Data inversion)</td>
<td>DRD</td>
</tr>
<tr>
<td>[6]</td>
<td>No label (All 0s or 1s)</td>
<td>ALL</td>
</tr>
<tr>
<td>[7]</td>
<td>No label (Pattern setting)</td>
<td>WRT</td>
</tr>
<tr>
<td>[8]</td>
<td>No label (Pattern reading)</td>
<td>RED?</td>
</tr>
</tbody>
</table>
### DLN <numeric>

**Parameter**
<numeric> = <DECIMAL NUMERIC PROGRAM DATA>

2 to 134217728

2 to 134,217,728 bits, in 1-bit steps

**Note:**
At 2 ch Combination, the setting range and Step are doubled.
At 4 ch Combination, the setting range and Step are quadrupled.

**Function**
Sets the pattern length during Data pattern reception.

**Example**
To set the pattern length during Data pattern reception to 100 bits:

> DLN 100

**Compatibility**
Compatible with the MP1632C and MP1762D.

### DLN?

**Response**
<numeric> = <NR1 NUMERIC RESPONSE DATA> (FIX10)

**Function**
Queries the pattern length during Data pattern reception.

**Example**

> DLN?

< DLN 2

< DLN 134217728

**Compatibility**
Compatible with the MP1632C and MP1762D.
**Chapter 8  Native Commands**

**PDT <start>,<end>,<data>**  

**Parameter**

- `<start>` = `<NON-DECIMAL PROGRAM DATA>`
  - #H0 to #H7FFFFFFF 0 to 7FFFFFFF bits, in 1-bit steps
  - (Specify in hexadecimal)

- `<end>` = `<NON-DECIMAL PROGRAM DATA>`
  - #H0 to #H7FFFFFFF 0 to 7FFFFFFF bits, in 1-bit steps
  - (Specify in hexadecimal)

**Note:**
- The maximum setting is the pattern length.
- At 2 ch Combination, the setting range is doubled.
- At 4 ch Combination, the setting range is quadrupled.

- `<data>` = `<NON-DECIMAL PROGRAM DATA>`
  - #H***...* Specify pattern data in hexadecimal format.
    - 1 to 400 characters (pattern data of 400 \(\times\) 4 bits)
    - Specify a character string consisting of 0 to 9 and A to F.
    - If the last character string of the pattern data is short of a hexadecimal unit (4 bits), add 0(s) up to the last bit to make a hexadecimal string.

  - #B***...* Specify pattern data in binary format.
    - 1 to 400 characters (pattern data of 400 bits)
    - Specify a character string consisting of 0 and 1.

**Function**

Sets the pattern data from the `<start>` to `<end>` addresses during Data pattern reception. The set data overwrites the specified range.

**Example**

To set 0 to 1F addresses of the pattern data to 5A:

> PDT #H0,#H1F,#H5A

**Compatibility**

Compatible with the MP1632C.

**Note:**

- When the number of bits specified in `<data>` is smaller than that of the range specified by `<start>` and `<end>`, `<data>` is set repeatedly for the specified range.

  `<Example>`

  - `<start>` = #H0, `<end>` = #H1F, `<data>` = #HABC
    
    Setting data: ABCABCAB

  - `<start>` = #H0, `<end>` = #H7, `<data>` = #B011
    
    Setting data: 01101101

- When the number of bits specified in `<data>` is larger than that of the range specified by `<start>` and `<end>`, `<data>` outside the specified range is cut off.

  `<Example>`

  - `<start>` = #H0, `<end>` = #HF, `<data>` = #HABCDEF
    
    Setting data: ABCD

  - `<start>` = #H0, `<end>` = #H3, `<data>` = #B01100110
    
    Setting data: 0110
### PDT? <start>

**Parameter**

- `<start>` = `<NON-DECIMAL PROGRAM DATA>`
- `#H0` to `#H7FFFFFFF` = 0 to `7FFFFFFF` bits
  
  (Specify in hexadecimal)

**Note:**

- The maximum setting is the pattern length.
- At 2 ch Combination, the setting range is doubled.
- At 4 ch Combination, the setting range is quadrupled.

**Response**

- `<data>` = `<HEXADECIMAL NUMERIC RESPONSE DATA>`
- `#H***...*` = Returns the pattern data in hexadecimal format.
  
  Within 400 characters (pattern data of `400 \times 4` bits)

**Function**

Queries `400 \times 4` bits of the pattern data from the `<start>` address during Data pattern reception.

**Example**

To query the pattern data from 0 address:

- `> PDT? #H0`
- `< PDT #H0123456789ABCDEF...` (minimum: 1, maximum: 400 in hexadecimal)
- `< PDT #H0`

**Compatibility**

Compatible with the MP1632C.
## BDT <start>,<end>,<bdata>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;start&gt; =  &lt;NON-DECIMAL PROGRAM DATA&gt;</td>
<td>#H0 to #H7FFFFFFF 0 to 7FFFFFFF bits, in 1-bit steps (Specify in hexadecimal)</td>
</tr>
<tr>
<td>&lt;end&gt; =  &lt;NON-DECIMAL PROGRAM DATA&gt;</td>
<td>#H0 to #H7FFFFFFF 0 to 7FFFFFFF bits, in 1-bit steps (Specify in hexadecimal)</td>
</tr>
</tbody>
</table>

**Note:**
- The maximum setting is the pattern length.
- At 2 ch Combination, the setting range is doubled.
- At 4 ch Combination, the setting range is quadrupled.

| <bdata> =  <ARBITRARY BLOCK PROGRAM DATA> | #XYYY<binary> | X: Number of YYY digits |
| | | YYY: Number of bytes of <binary>, 1 to 16,777,216 bytes |

**Note:**
- At 2 ch Combination, the setting range is doubled.
- At 4 ch Combination, the setting range is quadrupled.

**Function**
Sets binary data of the pattern data from <start> to <end> addresses during Data pattern reception.

The set data overwrites the specified range.

**Example**
To set 0 to 1F addresses of the pattern data to 41:

```
> BDT #H0,#H1F,#11A (A=41)
```

**Compatibility**
Compatible with the MP1632C.
### BDT? <start>[,<size>]

**Parameter**

\[
<\text{start}> = \text{NON-DECIMAL PROGRAM DATA}\]

\[
\#H0 \text{ to } \#H7FFFFFFF \quad 0 \text{ to } 7FFFFFFF \text{ bits, in 1-bit steps}
\]

(Specify in hexadecimal)

\[
<\text{size}> = \text{NR1 NUMERIC PROGRAM DATA}\]

\[
1 \text{ to } 16777216 \quad 1 \text{ to } 16,777,216 \text{ bytes, in 1-byte steps}
\]

**Note:**

- The maximum setting is the pattern length.
- At 2 ch Combination, the setting range is doubled.
- At 4 ch Combination, the setting range is quadrupled.

**Response**

\[
<\text{bdata}> = \text{DEFINITE LENGTH ARBITRARY BLOCK RESPONSE DATA}\]

Queries binary data of up to the setting pattern when \(<\text{size}>\) is omitted.

**Function**

Queries binary data of bytes specified by \(<\text{size}>\) from \(<\text{start}>\) address of the pattern data during Data pattern reception.

**Example**

To query 2 bytes of binary data from 0 address of the pattern data during Data pattern reception:

\[
> \text{BDT? } \#H0, 2 \\
< \text{BDT } #12AA
\]

To query the setting pattern of binary data from 0 address of the pattern data during Data pattern generation:

\[
> \text{BDT? } \#H0 \\
< \text{BDT } #516000AA... \quad \text{(Binary data of 16,777,216 bytes)}
\]

**Compatibility**

Compatible with the MP1632C.

### DRA <start>\,<end>

**Parameter**

\[
<\text{start}> = \text{NON-DECIMAL PROGRAM DATA}\]

\[
\#H0 \text{ to } \#H7FFFFFFF \quad 0 \text{ to } 7FFFFFFF \text{ bits, in 1-bit steps}
\]

(Specify in hexadecimal)

\[
<\text{end}> = \text{NON-DECIMAL PROGRAM DATA}\]

\[
\#H0 \text{ to } \#H7FFFFFFF \quad 0 \text{ to } 7FFFFFFF \text{ bits, in 1-bit steps}
\]

(Specify in hexadecimal)

**Note:**

- The maximum setting is the pattern length.
- At 2 ch Combination, the setting range is doubled.
- At 4 ch Combination, the setting range is quadrupled.

**Function**

Inverts the data in the program during Data pattern reception. Specify the inversion range by the \(<\text{start}>\) and \(<\text{end}>\) addresses.

**Example**

To invert 0 to 4F addresses of the pattern data:

\[
> \text{DRA } \#H0, \#H4F
\]

**Compatibility**

Compatible with the MP1632C.
**Chapter 8  Native Commands**

**DRD <start>,<delta>**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;start&gt; = &lt;NON-DECIMAL PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>#H0 to #H7FFFFFFF 0 to 7FFFFFFF bits, in 1-bit steps</td>
</tr>
<tr>
<td></td>
<td>(Specify in hexadecimal)</td>
</tr>
<tr>
<td>&lt;delta&gt;</td>
<td>= &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
</tr>
<tr>
<td></td>
<td>1 to 134217728 1 to 134,217,728 bits, in 1-bit steps</td>
</tr>
</tbody>
</table>

**Note:**

The maximum setting is the pattern length.
At 2 ch Combination, the setting range is doubled.
At 4 ch Combination, the setting range is quadrupled.

**Function**

Inverts the data in the program during Data pattern reception.
Specify the inversion range by the number of bits from the <start> address (relative specification).

**Example**

To invert 128 bits of data from 0 address of the pattern data:
```
> DRD #H0,128
```

**Compatibility**

Compatible with the MP1632C.

---

**ALL <data>**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;data&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Fills all bits with 0s.</td>
</tr>
<tr>
<td>1</td>
<td>Fills all bits with 1s.</td>
</tr>
</tbody>
</table>

**Function**

Sets 0s or 1s for all data in the program during Data pattern reception.

**Example**

To set 1s to all bits of the pattern data:
```
> ALL 1
```

**Compatibility**

Compatible with the MP1632C and MP1762D.
8.4 ED Commands

**WRT<byte>,<address>**

**Parameter**
- `<byte>` = `<DECIMAL NUMERIC PROGRAM DATA>`
  Pattern transmission byte number
  1 to 16777216 1 to 16,777,216, in 1-byte steps

*Note:*
- At 2 ch Combination, the setting range is doubled.
- At 4 ch Combination, the setting range is quadrupled.

- `<address>` = `<DECIMAL NUMERIC PROGRAM DATA>`
  Pattern input start address
  0 to 134217720 0 to 134,217,720, in single step

*Note:*
- At 2 ch Combination, the setting range is doubled.
- At 4 ch Combination, the setting range is quadrupled.

**Function**
Sets the number of bytes and the start address for binary transmission during Data pattern reception.

**Example**
To transmit 16 bytes of data from 0 address:
> WRT 16,0
< ABCDEFGHIJKLMNOP
(16 bytes of binary data)

**Compatibility**
Compatible with the MP1632C and MP1762D.

**RED?<byte>,<address>**

**Parameter**
- `<byte>` = `<DECIMAL NUMERIC PROGRAM DATA>`
  Pattern transmission byte number
  1 to 16777216 1 to 16,777,216, in 1-byte steps

*Note:*
- At 2 ch Combination, the setting range is doubled.
- At 4 ch Combination, the setting range is quadrupled.

- `<address>` = `<DECIMAL NUMERIC PROGRAM DATA>`
  Pattern output start address
  0 to 134217720 0 to 134,217,720, in single step

*Note:*
- At 2 ch Combination, the setting range is doubled.
- At 4 ch Combination, the setting range is quadrupled.

**Response**
Data pattern string (binary data)

**Function**
Sets the number of bytes and the start address for read using binary transmission during Data pattern reception.

**Example**
To read 16 bytes of data from 0 address:
> RED? 16,0
< ABCDEFGHIJKLMNOP
(16 bytes of binary data)

**Compatibility**
Compatible with the MP1632C and MP1762D.
8.4.3.5 Mixed-Data pattern setting commands

Figure 8.4.3.5-1 Mixed-Data pattern setting

Figure 8.4.3.5-2 Descramble Setup
Figure 8.4.3.5-3 Mixed-Data pattern setting (Pattern Editor)
### Table 8.4.3.5-1  Mixed-Data pattern setting commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1]</td>
<td>Bit Shift</td>
<td>MPB</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MPB?</td>
</tr>
<tr>
<td>[2]</td>
<td>PRBS Pattern</td>
<td>MPP</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MPP?</td>
</tr>
<tr>
<td>[3]</td>
<td>PRBS Marl Ratio</td>
<td>MPM</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MPM?</td>
</tr>
<tr>
<td>[4]</td>
<td>Descramble ON/OFF</td>
<td>MSC</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MSC?</td>
</tr>
<tr>
<td>[5]</td>
<td>PRBS Sequence</td>
<td>MPS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MPS?</td>
</tr>
<tr>
<td>[6]</td>
<td>Setup</td>
<td>MSS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MSS?</td>
</tr>
<tr>
<td>[7]</td>
<td>Set All/Reset All</td>
<td>MSL</td>
</tr>
<tr>
<td>[8]</td>
<td>Number of Block</td>
<td>MBL</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MBL?</td>
</tr>
<tr>
<td>[9]</td>
<td>Row Length</td>
<td>MRL</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MRL?</td>
</tr>
<tr>
<td>[10]</td>
<td>Data Length</td>
<td>MDL</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MDL?</td>
</tr>
<tr>
<td>[11]</td>
<td>Number of Row</td>
<td>MRW</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MRW?</td>
</tr>
<tr>
<td>[12]</td>
<td>No label (Pattern setting)</td>
<td>MDW</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MDW?</td>
</tr>
<tr>
<td>[13]</td>
<td>No label (Pattern setting)</td>
<td>MBW</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MBW?</td>
</tr>
<tr>
<td>[14]</td>
<td>No label (Data inversion)</td>
<td>MDA</td>
</tr>
<tr>
<td>[15]</td>
<td>No label (Data inversion)</td>
<td>MDD</td>
</tr>
<tr>
<td>[16]</td>
<td>No label (All 1s or 0s)</td>
<td>MDF</td>
</tr>
</tbody>
</table>
### MPB <numeric>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;numeric&gt;</td>
<td>= &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
</tr>
<tr>
<td>0</td>
<td>1-bit</td>
</tr>
<tr>
<td>1</td>
<td>3-bit</td>
</tr>
</tbody>
</table>

**Function**
Sets the bit shift for the PRBS pattern during Mixed Data pattern reception.

**Example**
To set the bit shift for the PRBS pattern during Mixed Data pattern reception to 1-bit shift:
> MPB 0

**Compatibility**
Incompatible with existing models.

### MPB?

<table>
<thead>
<tr>
<th>Response</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;numeric&gt;</td>
<td>= &lt;NR1 NUMERIC RESPONSE DATA&gt; (FIX1)</td>
</tr>
</tbody>
</table>

**Function**
Queries the bit shift for the PRBS pattern during Mixed Data pattern reception.

**Example**
> MPB?
< MPB 0

**Compatibility**
Incompatible with existing models.

### MPP <numeric>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;numeric&gt;</td>
<td>= &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
</tr>
<tr>
<td>2</td>
<td>$2^7-1$</td>
</tr>
<tr>
<td>3</td>
<td>$2^6-1$</td>
</tr>
<tr>
<td>4</td>
<td>$2^5-1$</td>
</tr>
<tr>
<td>5</td>
<td>$2^4-1$</td>
</tr>
<tr>
<td>6</td>
<td>$2^3-1$</td>
</tr>
<tr>
<td>7</td>
<td>$2^2-1$</td>
</tr>
<tr>
<td>8</td>
<td>$2^1-1$</td>
</tr>
<tr>
<td>9</td>
<td>$2^0-1$</td>
</tr>
</tbody>
</table>

**Function**
Sets the number of stages of the PRBS pattern during Mixed Data pattern reception.

**Example**
To set the number of stages of the PRBS pattern during Mixed Data pattern reception to $2^7-1$:
> MPP 2

**Compatibility**
Incompatible with existing models.
Chapter 8  Native Commands

MPP?

Response  <numeric> = <NR1 NUMERIC RESPONSE DATA> (FIX1)
Function  Queries the number of stages of the PRBS pattern during Mixed Data pattern reception.
Example  > MPP?
          < MPP 2
Compatibility  Incompatible with existing models.

MPM <mratio>

Parameter  <mratio> = <DECIMAL NUMERIC PROGRAM DATA>
            |
            | POS   | NEG |
            |-------|-----|
            | 0     | 0/8 | 8/8 |
            | 1     | 1/8 | 7/8 |
            | 2     | 1/4 | 3/4 |
            | 3     | 1/2 | 1/2INVT |
Function  Sets the mark ratio for the PRBS pattern during Mixed Data pattern reception.
Example  To set the PRBS pattern mark (Positive) mark ratio to 0/8 during Mixed Data pattern reception:
          > MPM 0
Compatibility  Incompatible with existing models.

MPM?

Response  <mratio> = <NR1 NUMERIC RESPONSE DATA> (FIX1)
Function  Queries the mark ratio for the PRBS pattern during Mixed Data pattern reception.
Example  > MPM?
          < MPM 0
Compatibility  Incompatible with existing models.
8.4  ED Commands

**MSC <numeric>**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;numeric&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Descramble OFF</td>
</tr>
<tr>
<td>1</td>
<td>Descramble ON</td>
</tr>
</tbody>
</table>

**Function**
Sets Descramble ON/OFF of the PRBS7 stage during pattern reception.

**Example**
To set Descramble ON of the PRBS7 stage during pattern reception:
> MSC 1

**Compatibility**
Incompatible with existing models.

**MSC?**

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;numeric&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt; (FIX1)</th>
</tr>
</thead>
</table>

**Function**
Queries the Descramble ON/OFF state of the PRBS7 stage during pattern reception.

**Example**
> MSC?
< MSC 1

**Compatibility**
Incompatible with existing models.

**MPS <sequence>**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;sequence&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Restart: Inconsecutive PRBS pattern</td>
</tr>
<tr>
<td>1</td>
<td>Consecutive: Consecutive PRBS pattern</td>
</tr>
</tbody>
</table>

**Function**
Sets the PRBS pattern sequence during Mixed Data pattern reception.

**Example**
To set the PRBS pattern sequence during Mixed Data pattern reception to Restart:
> MPS 0

**Compatibility**
Incompatible with existing models.

**MPS?**

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;sequence&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt; (FIX1)</th>
</tr>
</thead>
</table>

**Function**
Queries the PRBS pattern sequence during Mixed Data pattern reception.

**Example**
> MPS?
< MPS 0

**Compatibility**
Incompatible with existing models.
**Chapter 8  Native Commands**

### MSS <row>,<data>,<prbs>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;row&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1 to 16</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>&lt;DECIMAL NUMERIC PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;data&gt;</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Descramble OFF</td>
</tr>
<tr>
<td></td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Descramble ON</td>
</tr>
</tbody>
</table>

Note, however, that Descramble OFF is fixed for data of 1 row.

<table>
<thead>
<tr>
<th>&lt;prbs&gt;</th>
<th>&lt;DECIMAL NUMERIC PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Descramble OFF</td>
</tr>
<tr>
<td></td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Descramble ON</td>
</tr>
</tbody>
</table>

**Function**
Sets Descramble ON or OFF for the Data and PRBS of the specified Row.

**Example**
To set the Descramble settings for 1 Row to Data OFF, PRBS ON:

```
> MSS 1,0,1
```

**Compatibility**
Incompatible with existing models.

### MSS? <row>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;row&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1 to 16</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>&lt;DECIMAL NUMERIC PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;data&gt;</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;prbs&gt;</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;NR1 NUMERIC RESPONSE DATA&gt; (FIX1)</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Function</th>
<th>Queries the descramble setting (ON/OFF) for the Data and PRBS of the specified Row.</th>
</tr>
</thead>
</table>

**Example**

```
> MSS? 1
< MSS 0,1
```

** Compatibility**
Incompatible with existing models.
### MSL <numeric>

**Parameter**

<numeric> = <DECIMAL NUMERIC PROGRAM DATA>

- 0: All Reset
- 1: All Set

**Function**

Selects or cancels all Descramble settings.

**Example**

To select all Descramble settings:

```
> MSL 1
```

**Compatibility**

Incompatible with existing models.

### MBL <numeric>

**Parameter**

<numeric> = <DECIMAL NUMERIC PROGRAM DATA>

- 1 to 511: 1 to 511 blocks, in 1-block steps

**Function**

Sets the number of blocks during Mixed Data pattern reception.

**Example**

To set the number of blocks during Mixed Data pattern reception to 1:

```
> MBL 1
```

**Compatibility**

Incompatible with existing models.

### MBL?

**Response**

<numeric> = <NR1 NUMERIC RESPONSE DATA> (FIX3)

**Function**

Queries the number of blocks during Mixed Data pattern reception.

**Example**

```
> MBL?
< MBL 1
< MBL 511
```

**Compatibility**

Incompatible with existing models.

### MRW <numeric>

**Parameter**

<numeric> = <DECIMAL NUMERIC PROGRAM DATA>

- 1 to 16: 1 to 16 rows, in 1-row steps

**Function**

Sets the number of rows within one block that are edited during Mixed Data pattern reception.

**Example**

To set the number of rows within one block that are edited during Mixed Data pattern reception to 1:

```
> MRW 1
```

**Compatibility**

Incompatible with existing models.
Chapter 8  Native Commands

**MRW?**

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;numeric&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt; (FIX2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function</td>
<td>Queries the number of rows within one block that are edited during Mixed Data pattern reception.</td>
</tr>
<tr>
<td>Example</td>
<td>&gt; MRW?</td>
</tr>
<tr>
<td></td>
<td>&lt; MRW 1</td>
</tr>
<tr>
<td></td>
<td>&lt; MRW 16</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Incompatible with existing models.</td>
</tr>
</tbody>
</table>

**MRL <numeric>**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;numeric&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>768 to 2281701376 768 to 2,281,701,376 bits, in 128-bit steps</td>
</tr>
<tr>
<td><em>Note:</em></td>
<td>At 2 ch Combination, the setting range and Step are doubled.</td>
</tr>
<tr>
<td></td>
<td>At 4 ch Combination, the setting range and Step are quadrupled.</td>
</tr>
<tr>
<td>Function</td>
<td>Sets the pattern length of 1 row that is edited during Mixed Data pattern reception.</td>
</tr>
<tr>
<td>Example</td>
<td>To set pattern length to 768 bits:</td>
</tr>
<tr>
<td></td>
<td>&gt; MRL 768</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Incompatible with existing models.</td>
</tr>
</tbody>
</table>

**MRL?**

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;numeric&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt; (FIX11)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function</td>
<td>Queries the pattern length of 1 row that is edited during Mixed Data pattern reception.</td>
</tr>
<tr>
<td>Example</td>
<td>&gt; MRL?</td>
</tr>
<tr>
<td></td>
<td>&lt; MRL 768</td>
</tr>
<tr>
<td></td>
<td>&lt; MRL 2281701376</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Incompatible with existing models.</td>
</tr>
</tbody>
</table>
### MDL <numeric>

**Parameter**

<numeric> = <DECIMAL NUMERIC PROGRAM DATA>

512 to 134217728  
512 to 134,217,728 bits, in 1-bit steps

**Note:**

At 2 ch Combination, the setting range and Step are doubled.

At 4 ch Combination, the setting range and Step are quadrupled.

**Function**

Sets the Data pattern length that is edited during Mixed Data pattern reception.

**Example**

To set the pattern length to 512 bits:

> MDL 512

**Compatibility**

Incompatible with existing models.

### MDL? 

**Response**

<numeric> = <NR1 NUMERIC RESPONSE DATA> (FIX10)

**Function**

Queries the Data pattern length that is edited during Mixed Data pattern reception.

**Example**

> MDL?

< MDL 512

< MDL 134217728

**Compatibility**

Incompatible with existing models.
MDW <block>,<start>,<end>,<data>

**Parameter**

- `<block>` = `<DECIMAL NUMERIC PROGRAM DATA>`
  1 to 511 1 to 511 blocks, in 1-block steps
- `<start>` = `<NON-DECIMAL PROGRAM DATA>`
  #H0 to #H7FFFFFFF 0 to 7FFFFFFF bits, in 1-bit steps
  (Specify in hexadecimal)
- `<end>` = `<NON-DECIMAL PROGRAM DATA>`
  #H0 to #H7FFFFFFF 0 to 7FFFFFFF bits, in 1-bit steps
  (Specify in hexadecimal)

**Note:**
- The maximum setting is the pattern length.
- At 2 ch Combination, the setting range is doubled.
- At 4 ch Combination, the setting range is quadrupled.

- `<data>` = `<NON-DECIMAL PROGRAM DATA>`
  #H***...* Specify pattern data in hexadecimal format.
  1 to 400 characters (pattern data of 400 × 4 bits)
  Specify a character string consisting of 0 to 9 and A to F.
  If the last character string of the pattern data is short of a hexadecimal unit (4 bits), add 0(s) up to the last bit to make a hexadecimal string.
  #B***...* Specify pattern data in binary format.
  1 to 400 characters (pattern data of 400 bits)
  Specify a character string consisting of 0 and 1.

**Function**
Sets the pattern data from the `<start>` to `<end>` addresses during Mixed Data pattern reception. The set data overwrites the specified range.

**Example**
To set block 1 pattern data from 0 to 1F addresses to 11:
> MDW 1,#H0,#H1F,#H11

**Compatibility**
Incompatible with existing models.
MDW? <block>,<start>

### Parameter

- `<block>` = `<DECIMAL NUMERIC PROGRAM DATA>`
  - 1 to 511 blocks, in 1-block steps
- `<start>` = `<NON-DECIMAL PROGRAM DATA>`
  - #H0 to #H7FFFFFFF
  - 0 to 7FFFFFFF bits, in 1-bit steps
  - (Specify in hexadecimal)

**Note:**

- The maximum setting is the pattern length.
- At 2 ch Combination, the setting range is doubled.
- At 4 ch Combination, the setting range is quadrupled.

### Response

- `<data>` = `<HEXADECIMAL NUMERIC RESPONSE DATA>`
  - #H***...* Returns the pattern data in hexadecimal format.
  - Within 400 characters (pattern data of 400 \( \times 4 \) bits)

### Function

Queries 400 \( \times 4 \) bits of the pattern data from the `<start>` address during Mixed Data pattern reception.

### Example

To query block 1 pattern data from 0 address:

\[
\begin{align*}
&> \text{MDW? 1,#H0} \\
&< \text{MDW #H0123456789ABCDEF... (minimum: 1, maximum: 400 in hexadecimal)} \\
&< \text{MDW #H0}
\end{align*}
\]

### Compatibility

Incompatible with existing models.
**MBW <block>,<start>,<end>,<bdata>**

**Parameter**

- `<block>` = `<DECIMAL NUMERIC PROGRAM DATA>`
  - 1 to 511 blocks, in 1-block steps
- `<start>` = `<NON-DECIMAL PROGRAM DATA>`
  - #H0 to #H7FFFFFFF (0 to 7FFFFFFF bits, in 1-bit steps)
    - (Specify in hexadecimal)
- `<end>` = `<NON-DECIMAL PROGRAM DATA>`
  - #H0 to #H7FFFFFFF (0 to 7FFFFFFF bits, in 1-bit steps)
    - (Specify in hexadecimal)

**Note:**

The maximum setting is the pattern length.
- At 2 ch Combination, the setting range is doubled.
- At 4 ch Combination, the setting range is quadrupled.

- `<bdata>` = `<ARBITRARY BLOCK PROGRAM DATA>`
  - #XYYY<binary>:
    - X: Number of YYY digits
    - YYY: Number of bytes of <binary>, 1 to 16,777,216 bytes
    - <binary>: Binary data up to maximum bytes

**Function**

Sets binary data of the pattern data in each block from `<start>` to `<end>` addresses during Mixed Data pattern reception.
The set data overwrites the specified range.

**Example**

To set the pattern data from 0 to 1F addresses to 43 during Mixed Data pattern reception:

> MBW 1,#H0,#H1F,#11C (C=43)

**Compatibility**

Incompatible with existing models.
**MBW? <block>,<start>[,<size>]**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| <block>  | = <DECIMAL NUMERIC PROGRAM DATA>  
1 to 511 1 to 511 blocks, in 1-block steps |
| <start>  | = <NON-DECIMAL PROGRAM DATA>  
#H0 to #H7FFFFFFF 0 to 7FFFFFFF bits, in 1-bit steps  
(Specify in hexadecimal) |

**Note:**
- The maximum setting is the pattern length.
- At 2 ch Combination, the setting range is doubled.
- At 4 ch Combination, the setting range is quadrupled.

| <size> | = <NR1 NUMERIC PROGRAM DATA>  
1 to 16777216 1 to 16,777,216 bytes, in 1-byte steps |

**Response**
- <bdata> = <DEFINITE LENGTH ARBITRARY BLOCK RESPONSE DATA>
  - Queries up to the setting pattern of binary data when [<size>] is omitted.

**Function**
- Queries binary data of bytes specified by <size> from the <start> address of the pattern data in each block during Mixed Data pattern reception.

**Example**
- To query 2 bytes of block 1 pattern data from 0 address:
  ```
  > MBW? 1,#H0,2  
  < MBW #12AA
  ```
- To query the setting pattern of block 1 binary data from 0 address:
  ```
  > MBW? 1,#H0  
  < MBW #516000AA... (Binary data of 16,777,216 bytes)
  ```

**Compatibility**
- Incompatible with existing models.
### MDA <block>,<start>,<end>

**Parameter**
- `<block>` = `<DECIMAL NUMERIC PROGRAM DATA>`
  1 to 511  1 to 511 blocks, in 1-block steps
- `<start>` = `<NON-DECIMAL PROGRAM DATA>`
  
  #H0 to #H7FFFFFFF  0 to 7FFFFFFF bits, in 1-bit steps
  *(Specify in hexadecimal)*
- `<end>` = `<NON-DECIMAL PROGRAM DATA>`
  
  #H0 to #H7FFFFFFF  0 to 7FFFFFFF bits, in 1-bit steps
  *(Specify in hexadecimal)*

**Note:**
- The maximum setting is the pattern length.
- At 2 ch Combination, the setting range is doubled.
- At 4 ch Combination, the setting range is quadrupled.

**Function**
Inverts the data in the program during Mixed Data pattern reception.
Specify the inversion range by the start and end addresses.

**Example**
To invert block 1 pattern data from 0 to 4F addresses:
> MDA 1,#H0,#H4F

**Compatibility**
Incompatible with existing models.

### MDD <block>,<start>,<delta>

**Parameter**
- `<block>` = `<DECIMAL NUMERIC PROGRAM DATA>`
  1 to 511  1 to 511 blocks, in 1-block steps
- `<start>` = `<NON-DECIMAL PROGRAM DATA>`
  
  #H0 to #H7FFFFFFF  0 to 7FFFFFFF bits, in 1-bit steps
  *(Specify in hexadecimal)*
- `<delta>` = `<DECIMAL NUMERIC PROGRAM DATA>`
  1 to 134217728  1 to 134,217,728 bits, in 1-bit steps

**Note:**
- The maximum setting is the pattern length.
- At 2 ch Combination, the setting range is doubled.
- At 4 ch Combination, the setting range is quadrupled.

**Function**
Inverts the data in the program during Mixed Data pattern reception.
Specify the inversion range by the number of bits from the start address (relative specification).

**Example**
To invert 256 bits of block 1 pattern data from 0 address.
> MDD 1,#H0,256

**Compatibility**
Incompatible with existing models.
MDF <block>,<range>,<page>,<data>

Parameter

- `<block>` = `<DECIMAL NUMERIC PROGRAM DATA>`
  - 1 to 511: 1 to 511 blocks, in 1-block steps
- `<range>` = `<DECIMAL NUMERIC PROGRAM DATA>`
  - 0: Specifies a page (one page is defined as 128 bits).
  - 1: Specifies all data.
- `<page>` = `<DECIMAL NUMERIC PROGRAM DATA>`
  - 1 to 1048576: 1 to 1,048,576 (Max.) pages, in 1-page steps
  - Max = PatternLength/128 (rounding up fractions)
  - Specify "0" when `<range>` is set to ALL.
- `<data>` = `<DECIMAL NUMERIC PROGRAM DATA>`
  - 0: Fills the specified range with 0s.
  - 1: Fills the specified range with 1s.

Note:
The maximum setting value is the pattern length.

Function
Sets 0s or 1s for the specified page or all data in the program during Mixed Data pattern reception.

Example
To set page 3 of block 1 to 0:
> MDF 1,0,3,0

Compatibility
Incompatible with existing models.
Chapter 8  Native Commands

8.4.3.6  Sequence pattern setting commands

Figure 8.4.3.6-1  Sequence pattern setting

Figure 8.4.3.6-2  Match Pattern Condition
8.4 ED Commands

Figure 8.4.3.6-3  Sequence Pattern Setting

Figure 8.4.3.6-4  Sequence pattern setting (Pattern Editor screen)
### Table 8.4.3.6-1  Sequence pattern setting commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1]</td>
<td>BlockNo</td>
<td>SQL</td>
</tr>
<tr>
<td></td>
<td>Match Pattern</td>
<td>SQL?</td>
</tr>
<tr>
<td></td>
<td>Match Pattern Length</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>SQM?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SQE?</td>
</tr>
<tr>
<td>[4]</td>
<td>Add</td>
<td>SAD</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SAD?</td>
</tr>
<tr>
<td>[5]</td>
<td>Copy</td>
<td>SCO</td>
</tr>
<tr>
<td>[6]</td>
<td>Cut</td>
<td>SCU</td>
</tr>
<tr>
<td>[7]</td>
<td>Paste</td>
<td>SPA</td>
</tr>
<tr>
<td>[8]</td>
<td>Clear All</td>
<td>SCD</td>
</tr>
<tr>
<td>[9]</td>
<td>Number of Block</td>
<td>SLG</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SLG?</td>
</tr>
<tr>
<td>[10]</td>
<td>No label (Pattern setting)</td>
<td>SDW</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SDW?</td>
</tr>
<tr>
<td>[11]</td>
<td>No label (Pattern setting)</td>
<td>SBW</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SBW?</td>
</tr>
<tr>
<td>[12]</td>
<td>No label (Data inversion)</td>
<td>SDA</td>
</tr>
<tr>
<td>[13]</td>
<td>No label (Data inversion)</td>
<td>SDD</td>
</tr>
<tr>
<td>[14]</td>
<td>No label (All 0s or 1s)</td>
<td>SDF</td>
</tr>
</tbody>
</table>
SQL <block>,<content>,<length>

Parameter
- <block> = <DECIMAL NUMERIC PROGRAM DATA>
  1 to 128  1 to 128 blocks, in 1-block steps
- <content> = <DECIMAL NUMERIC PROGRAM DATA>
  0  Pattern A
  1  Pattern B
- <length> = <DECIMAL NUMERIC PROGRAM DATA>
  4 to 64  4 to 64 bits, in 4-bit steps

Function
Sets the pattern length of the Match pattern during Sequence pattern setting.

Example
To set the pattern length of the Match pattern A of block 1 to 4 bits during Sequence pattern setting:
> SQL 1,0,4

Compatibility
Incompatible with existing models.

SQL? <block>,<content>

Parameter
- <block> = <DECIMAL NUMERIC PROGRAM DATA>
  1 to 128  1 to 128 blocks, in 1-block steps
- <content> = <DECIMAL NUMERIC PROGRAM DATA>

Response
- <length> = <NR1 NUMERIC RESPONSE DATA> (FIX2)
  4 to 64  4 to 64 bits

Function
Queries the pattern length of the Match pattern during Sequence pattern setting.

Example
To query the pattern length of the Match pattern A of block 1 during Sequence pattern setting:
> SQL? 1,0
< SQL 4
< SQL 64

Compatibility
Incompatible with existing models.
**SQM <block>,<content>,<data>**

**Parameter**
- <block> = <DECIMAL NUMERIC PROGRAM DATA>
  - 1 to 128: 1 to 128 blocks, in 1-block steps
- <content> = <DECIMAL NUMERIC PROGRAM DATA>
  - 0: Pattern A
  - 1: Pattern B
- <data> = <NON-DECIMAL NUMERIC PROGRAM DATA>
  - #H**...*: 1 to 16 characters
  - (Specify pattern data in hexadecimal format)
  - #B**...*: 1 to 64 characters
  - (Specify pattern data in binary format)

The editable data length depends on the Length setting.

**Function**
Edits the Match pattern during Sequence pattern setting.

**Example**
To edit one binary character of the Match pattern A of block 1 during Sequence pattern setting:
> SQM 1,0,#H1

**Compatibility**
Incompatible with existing models.

---

**SQM? <block>,<content>,<form>**

**Parameter**
- <block> = <DECIMAL NUMERIC PROGRAM DATA>
  - 1 to 128: 1 to 128 blocks, in 1-block steps
- <content> = <DECIMAL NUMERIC PROGRAM DATA>
  - 0: Pattern A
  - 1: Pattern B
- <form> = <DECIMAL NUMERIC PROGRAM DATA>
  - 0: BIN
  - 1: HEX

**Response**
- <data> = <HEXADECIMAL NUMERIC RESPONSE DATA>
  - #H**...*: Pattern data is returned in hexadecimal format.
  - #B**...*: Pattern data is returned in binary format.

**Function**
Queries the Match pattern that is set during Sequence pattern setting.

**Example**
To query the Match pattern A of block 1 that is set during Sequence pattern setting in hexadecimal format:
> SQM? 1,0,1

< SQM #H1

**Compatibility**
Incompatible with existing models.
**SQE <block>,<content>,<data>**

**Parameter**
- `<block>` = `<DECIMAL NUMERIC PROGRAM DATA>`
- 1 to 128 blocks, in 1-block steps
- `<content>` = `<DECIMAL NUMERIC PROGRAM DATA>`
- 0 Pattern A
- 1 Pattern B
- `<data>` = `<NON-DECIMAL PROGRAM DATA>`
- #H**...* 1 to 16 characters
  (Specify pattern data in hexadecimal format)
- #B**...* 1 to 64 characters
  (Specify pattern data in binary format)

The editable data length depends on the Length setting.

**Function**
Edits the Mask pattern during Sequence pattern setting.

**Example**
To edit one binary character of the Mask pattern A of block 1 during Sequence pattern setting in hexadecimal format:

```
> SQE 1,0,#H1
```

**Compatibility**
Incompatible with existing models.

---

**SQE? <block>,<content>,<form>**

**Parameter**
- `<block>` = `<DECIMAL NUMERIC PROGRAM DATA>`
- 1 to 128 blocks, in 1-block steps
- `<content>` = `<DECIMAL NUMERIC PROGRAM DATA>`
- `<form>` = `<DECIMAL NUMERIC PROGRAM DATA>`
- 0 BIN
- 1 HEX

**Response**
- `<data>` = `<HEXADECIMAL NUMERIC RESPONSE DATA>`
- #H**...* Pattern data is returned in hexadecimal format.
- #B**...* Pattern data is returned in binary format.

**Function**
Queries the Mask pattern that is set during Sequence pattern setting.

**Example**
To query the Mask pattern A of block 1 that is set during Sequence pattern setting in hexadecimal format:

```
> SQE? 1,0,1
< SQE #H1
```

**Compatibility**
Incompatible with existing models.
### SAD <block>

**Parameter**  
\(<\text{block}> = \text{DECIMAL NUMERIC PROGRAM DATA}\>  
1 to 128  
1 to 128 blocks, in 1-block steps

**Function**  
Sets the number of blocks that are edited during Sequence pattern reception.

**Example**  
To set the number of blocks that are edited to 10:  
\(> \text{SAD 10}\)

**Compatibility**  
Incompatible with existing models

### SAD?

**Response**  
\(<\text{block}> = \text{NR1 NUMERIC RESPONSE DATA} \quad (\text{FIX3})\>

**Function**  
Queries the number of blocks that are edited during Sequence pattern reception.

**Example**  
\(> \text{SAD?}\)  
\(<\text{SAD 10}\)  
\(<\text{SAD 128}\)

**Compatibility**  
Incompatible with existing models

### SCO <block>

**Parameter**  
\(<\text{block}> = \text{DECIMAL NUMERIC PROGRAM DATA}\>  
1 to 128  
1 to 128 blocks, in 1-block steps

**Function**  
Copies the settings in the specified block during Sequence pattern reception.

**Example**  
To copy the settings in block 1 during Sequence pattern reception:  
\(> \text{SCO 1}\)

**Compatibility**  
Incompatible with existing models.

### SCU <block>

**Parameter**  
\(<\text{block}> = \text{DECIMAL NUMERIC PROGRAM DATA}\>  
1 to 128  
1 to 128 blocks, in 1-block steps

**Function**  
Cuts (copies and removes) the settings in the specified block during Sequence pattern reception.

**Example**  
To cut the settings in block 1 during Sequence pattern reception:  
\(> \text{SCU 1}\)

**Compatibility**  
Incompatible with existing models.
### SPA<block>

**Parameter**
\[
<block> = \text{DECIMAL NUMERIC PROGRAM DATA} \\
1 \text{ to } 1 \text{ to } 128 \text{ blocks, in } 1 \text{-block steps}
\]

**Function**
Paste the settings in the specified block during Sequence pattern reception.

**Example**
To paste the settings in block 1 during Sequence pattern reception:
\[> \text{SPA } 1\]

**Compatibility**
Incompatible with existing models.

### SCD

**Function**
Clears all the settings during Sequence pattern reception.

**Example**
\[> \text{SCD}\]

**Compatibility**
Incompatible with existing models.

### SLG <block>,<numeric>

**Parameter**
\[
<block> = \text{DECIMAL NUMERIC PROGRAM DATA} \\
1 \text{ to } 128 \text{ blocks, in } 1 \text{-block steps} \\
<numeric> = \text{DECIMAL NUMERIC PROGRAM DATA} \\
8192 \text{ to } 1048576 \text{ bits, in } 128 \text{-bit steps}
\]

Be sure to set a value so that the following condition is satisfied.
\[
\text{Number of blocks} \times \text{Block length} \leq 128 \text{ Mbits}
\]

**Function**
Sets the length of the specified block during Sequence pattern reception.

**Example**
To set the length of block 1 to 1 bit during Sequence pattern reception:
\[> \text{SLG } 1,8192\]

**Compatibility**
Incompatible with existing models.

### SLG? <block>

**Parameter**
\[
<block> = \text{DECIMAL NUMERIC PROGRAM DATA} \\
1 \text{ to } 128 \text{ blocks, in } 1 \text{-block steps}
\]

**Response**
\[
<numeric> = \text{NR1 NUMERIC RESPONSE DATA} (\text{FIX7}) \\
8192 \text{ to } 1048576 \text{ bits}
\]

**Function**
Queries the length of the specified block during Sequence pattern reception.

**Example**
To query the length of block 1 during Sequence pattern reception:
\[> \text{SLG? } 1\]
\[< \text{SLG } 8192\]
\[< \text{SLG } 1048576\]

**Compatibility**
Incompatible with existing models.
SDW <block>,<start>,<end>,<data>

Parameter

<block> = <DECIMAL NUMERIC PROGRAM DATA>
1 to 128    1 to 128 blocks, in 1-block steps

<start> = <NON-DECIMAL PROGRAM DATA>
#H0 to #HFFFFF    0 to FFFFFF bits, in 1-bit steps
(Specify in hexadecimal)

<end> = <NON-DECIMAL PROGRAM DATA>
#H0 to #HFFFFF    0 to FFFFFF bits, in 1-bit steps
(Specify in hexadecimal)

**Note:**
The maximum setting value is the pattern length.

<data> = <NON-DECIMAL PROGRAM DATA>

#H***...*
Specify pattern data in hexadecimal format.
1 to 400 characters (pattern data of 400 \( \times \) 4 bits)
Specify a character string consisting of 0 to 9 and A to F.
If the last character string of the pattern data is short of a hexadecimal unit (4 bits), add 0(s) up to
the last bit to make a hexadecimal string.

#B***...*
Specify pattern data in binary format.
1 to 400 characters (pattern data of 400 bits)
Specify a character string consisting of 0 and 1.

**Function**
Sets the binary data from <start> to <end> addresses of the pattern data during Sequence pattern reception.
The set data overwrites the specified range.

**Example**
To set the pattern data of block 1 data from 0 to FFFF addresses to 5A:
> SDW 1,#H0,#HFFFFF,#H5A

**Compatibility**
Incompatible with existing models.
8.4 ED Commands

**SDW? <block>,<start>**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;block&gt;</td>
<td>&lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
</tr>
<tr>
<td>1 to 128</td>
<td>1 to 128 blocks, in 1-block steps</td>
</tr>
<tr>
<td>&lt;start&gt;</td>
<td>&lt;NON-DECIMAL PROGRAM DATA&gt;</td>
</tr>
<tr>
<td>#H0 to #HFFFFF</td>
<td>0 to FFFFF bits, in 1-bit steps (Specify in hexadecimal)</td>
</tr>
</tbody>
</table>

**Note:**
The maximum setting value is the pattern length.

<table>
<thead>
<tr>
<th>Response</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;data&gt;</td>
<td>HEXADECIMAL NUMERIC RESPONSE DATA</td>
</tr>
<tr>
<td>#H***...*</td>
<td>Specify pattern data in hexadecimal format.</td>
</tr>
<tr>
<td>Within 400 characters (pattern data of 400 × 4 bits)</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Queries the pattern data of 400 × 4 bits from the &lt;start&gt; address during Sequence pattern reception.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Example</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>To query 400 × 4 bits of pattern data from 0 address:</td>
<td></td>
</tr>
<tr>
<td>&gt; SDW? 1,#H0</td>
<td></td>
</tr>
<tr>
<td>&lt; SDW #H0123456789ABCDEF... (minimum: 1, maximum: 400 in hexadecimal)</td>
<td></td>
</tr>
<tr>
<td>&lt; SDW #H0</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Compatibility</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Incompatible with existing models.</td>
<td></td>
</tr>
</tbody>
</table>
SBW <block>,<start>,<end>,<bdata>

Parameter

- `<block>` = `<DECIMAL NUMERIC PROGRAM DATA>`
  - 1 to 128: 1 to 128 blocks, in 1-block steps

- `<start>` = `<NON-DECIMAL PROGRAM DATA>`
  - `#H0` to `#HFFFFF`: 0 to FFFFF bits, in 1-bit steps
    - (Specify in hexadecimal)

- `<end>` = `<NON-DECIMAL PROGRAM DATA>`
  - `#H0` to `#HFFFFF`: 0 to FFFFF bits, in 1-bit steps
    - (Specify in hexadecimal)

**Note:**
The maximum setting value is the pattern length.

- `<bdata>` = `<ARBITRARY BLOCK PROGRAM DATA>`
  - `#XYYY<binary>`: Number of YYY digits
    - X: Number of YYY digits
    - YYY: Number of bytes of `<binary>`, 1 to 131,072 bytes
    - `<binary>`: Binary data up to 131,072 bytes

Function

Sets the binary data from `<start>` to `<end>` addresses of the pattern data during Sequence pattern reception.

The set data overwrites the specified range.

Example

To set block 1 data from 0 to FFFFF addresses to 44:

```
> SBW 1,#H0,#HFFFFF,#11D
```

Compatibility

Incompatible with existing models.
### SBW? <block>,<start>[,<size>]

**Parameter**
- `<block>` = <DECIMAL NUMERIC PROGRAM DATA>
  - 1 to 128 1 to 128 blocks, in 1-block steps
- `<start>` = <NON-DECIMAL PROGRAM DATA>
  - #H0 to #HFFFFF 0 to FFFFF bits, in 1-bit steps
  - (Specify in hexadecimal)
- `[<size>]` = <NR1 NUMERIC PROGRAM DATA>
  - 1 to 131072 1 to 131,072 bytes, in 1-byte steps

**Response**
- `<bdata>` = <DEFINITE LENGTH ARBITRARY BLOCK RESPONSE DATA>
  - `#XYYY<binary>`
    - X: Number of YYY digits
    - YYY: Number of bytes of `<binary>`, 1 to 131,072 bytes
  - `<binary>`: Binary data up to 131,072 bytes

**Function**
Queries binary data of bytes specified by `<size>` from the `<start>` address of the pattern data during Sequence pattern reception.

**Example**
To query 2 bytes of block 1 data from 0 address:
```shell
> SBW? 1,#H0,2
< SBW #12DD
```
To query the setting pattern of block 1 data from 0 address:
```shell
> SBW? 1,#H0
< SBW #516000AA... (16,000 bytes of binary data)
```

**Compatibility**
Incompatible with existing models.

---

**Note:**
The maximum setting value is the pattern length.
Chapter 8  Native Commands

SDA <block>,<start>,<end>

Parameter
- <block> = <DECIMAL NUMERIC PROGRAM DATA>
  1 to 128 1 to 128 blocks, in 1-block steps
- <start> = <NON-DECIMAL PROGRAM DATA>
  #H0 to #HFFFFF 0 to FFFFF bits, in 1-bit steps
  (Specify in hexadecimal)
- <end> = <NON-DECIMAL PROGRAM DATA>
  #H0 to #HFFFFF 0 to FFFFF bits, in 1-bit steps
  (Specify in hexadecimal)

Note:
The maximum setting value is the pattern length.

Function
Inverts the data in the program during Sequence pattern reception.
Specify the inversion range by the <start> and <end> addresses.

Example
To invert block 3 data from 0 to FFF addresses:
> SDA 3,#H0,#HFFF

Compatibility
Incompatible with existing models.

SDD <block>,<start>,<delta>

Parameter
- <block> = <DECIMAL NUMERIC PROGRAM DATA>
  1 to 128 1 to 128 blocks, in 1-block steps
- <start> = <NON-DECIMAL PROGRAM DATA>
  #H0 to #HFFFFF 0 to FFFFF bits, in 1-bit steps
  (Specify in hexadecimal)
- <delta> = <DECIMAL NUMERIC PROGRAM DATA>
  1 to 1048576 1 to 1,048,576 bits, in 1-bit steps
  (Specify in hexadecimal)

Note:
The maximum setting value is the pattern length.

Function
Inverts the data in the program during Sequence pattern reception.
Specify the inversion range by the number of bits from the start address (relative specification).

Example
To invert block 5 data from bit 0 to bit 128 addresses:
> SDD 5,#H0,128

Compatibility
Incompatible with existing models.
### SDF <block>,<range>,<page>,<data>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;block&gt;</td>
<td>&lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
</tr>
<tr>
<td>1 to 128</td>
<td>1 to 128 blocks, in 1-block steps</td>
</tr>
<tr>
<td>&lt;range&gt;</td>
<td>&lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
</tr>
<tr>
<td>0</td>
<td>Specifies a page (one page is defined as 128 bits)</td>
</tr>
<tr>
<td>1</td>
<td>Specifies all data.</td>
</tr>
<tr>
<td>&lt;page&gt;</td>
<td>&lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
</tr>
<tr>
<td>1 to 8192</td>
<td>1 to 8,192 (Max.) pages, in 1-page steps</td>
</tr>
<tr>
<td>Max</td>
<td>Pattern Length/128 (rounding up fractions)</td>
</tr>
<tr>
<td>Specify</td>
<td>&quot;0&quot; when &lt;range&gt; is set to ALL.</td>
</tr>
<tr>
<td>&lt;data&gt;</td>
<td>&lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
</tr>
<tr>
<td>0</td>
<td>Fills the specified range with 0s.</td>
</tr>
<tr>
<td>1</td>
<td>Fills the specified range with 1s.</td>
</tr>
</tbody>
</table>

**Function**
Sets 0s or 1s for the specified page or all data in the program during Sequence pattern reception.

**Example**
To set 0s for the first page in block 1:
> SDF 1,0,1,0

**Compatibility**
Incompatible with existing models.
8.4.3.7 Mask setting commands

Figure 8.4.3.7-1 Mask ON/OFF setting

Figure 8.4.3.7-2 Bit Window setting (PRBS pattern)

Figure 8.4.3.7-3 Bit Window, Block Window setting (Zero-Substitution, Data, Mixed-Data, Sequence pattern)
### 8.4 ED Commands

#### Table 8.4.3.7-1  Mask setting commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1]</td>
<td>Block Window ON/OFF</td>
<td>MGE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MGE?</td>
</tr>
<tr>
<td>[2]</td>
<td>Bit Window ON/OFF</td>
<td>MSE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MSE?</td>
</tr>
<tr>
<td>[3]</td>
<td>External Mask ON/OFF</td>
<td>MEX</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MEX?</td>
</tr>
<tr>
<td>[4]</td>
<td>Block Window (Zero-Substitution, Data)</td>
<td>MGB</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MGB?</td>
</tr>
<tr>
<td>[5]</td>
<td>Bit Window</td>
<td>CHM</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CHM?</td>
</tr>
<tr>
<td>[6]</td>
<td>Block Window (Pattern setting)</td>
<td>BDM</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BDM?</td>
</tr>
<tr>
<td>[7]</td>
<td>Block Window (Data inversion)</td>
<td>DRM</td>
</tr>
<tr>
<td>[8]</td>
<td>Block Window (Data inversion)</td>
<td>DBM</td>
</tr>
<tr>
<td>[9]</td>
<td>Block Window (All 0s or 1s)</td>
<td>ALB</td>
</tr>
<tr>
<td>[10]</td>
<td>Block Window (Mixed Data)</td>
<td>MDB</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MDB?</td>
</tr>
<tr>
<td>[11]</td>
<td>Block Window (Pattern setting)</td>
<td>MBM</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MBM?</td>
</tr>
<tr>
<td>[12]</td>
<td>Block Window (Data inversion)</td>
<td>MDM</td>
</tr>
<tr>
<td>[13]</td>
<td>Block Window (Data inversion)</td>
<td>MMB</td>
</tr>
<tr>
<td>[14]</td>
<td>Block Window (All 0s or 1s)</td>
<td>MFB</td>
</tr>
<tr>
<td>[15]</td>
<td>Block Window (Sequence)</td>
<td>MSB</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MSB?</td>
</tr>
<tr>
<td>[16]</td>
<td>Block Window (Pattern setting)</td>
<td>SBM</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SBM?</td>
</tr>
<tr>
<td>[17]</td>
<td>Block Window (Data inversion)</td>
<td>SDM</td>
</tr>
<tr>
<td>[18]</td>
<td>Block Window (Data inversion)</td>
<td>SDB</td>
</tr>
<tr>
<td>[19]</td>
<td>Block Window (All 0s or 1s)</td>
<td>SMF</td>
</tr>
</tbody>
</table>
Chapter 8  Native Commands

MGE <numeric>
 Parameter  
<numeric> = <DECIMAL NUMERIC PROGRAM DATA>
  0  Block Window OFF
  1  Block Window ON
 Function  Sets Block Window ON/OFF to the received pattern.
 Example  
To set Block Window ON to the received pattern:
  > MGE 1
 Compatibility  
Compatible with the MP1762D.

MGE?
 Response  <numeric> = <NR1 NUMERIC RESPONSE DATA> (FIX1)
 Function  Queries the Block Window ON/OFF state of the received pattern.
 Example  
> MGE?
 < MGE 1
 Compatibility  
Compatible with the MP1762D.

MSE <numeric>
 Parameter  
<numeric> = <DECIMAL NUMERIC PROGRAM DATA>
  0  Bit Window OFF
  1  Bit Window ON
 Function  Sets Bit Window ON/OFF to the received pattern.
 Example  
To set Bit Window ON to the received pattern:
  > MSE 1
 Compatibility  
Compatible with the MP1762D.

MSE?
 Response  <numeric> = <NR1 NUMERIC RESPONSE DATA> (FIX1)
 Function  Queries the Bit Window ON/OFF state of the received pattern.
 Example  
> MSE?
 < MSE 1
 Compatibility  
Compatible with the MP1762D.
### MEX <numeric>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;numeric&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>External Mask OFF</td>
</tr>
<tr>
<td>1</td>
<td>External Mask ON</td>
</tr>
</tbody>
</table>

**Function**
Sets External Mask ON/OFF to the received pattern.

**Example**
To set External Mask ON to the received pattern:

```plaintext
> MEX 1
```

**Compatibility**
Incompatible with existing models.

### MEX?

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;numeric&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt; (FIX1)</th>
</tr>
</thead>
</table>

**Function**
Queries the External Mask ON/OFF state of the received pattern.

**Example**
To query the External Mask ON/OFF state of the received pattern:

```plaintext
> MEX?
< MEX 1
```

**Compatibility**
Incompatible with existing models.
MGB <start>,<end>,<data>

Parameter

- `<start>` = `<NON-DECIMAL PROGRAM DATA>`
  
  #H0 to #H7FFFFFFF 0 to 7FFFFFFF bits, in 1-bit steps
  (Specify in hexadecimal)

- `<end>` = `<NON-DECIMAL PROGRAM DATA>`
  
  #H0 to #H7FFFFFFF 0 to 7FFFFFFF bits, in 1-bit steps
  (Specify in hexadecimal)

**Note:**

- The maximum setting is the pattern length.
- At 2 ch Combination, the setting range is doubled.
- At 4 ch Combination, the setting range is quadrupled.

- `<data>` = `<NON-DECIMAL PROGRAM DATA>`
  
  #H***...* Specify pattern data in hexadecimal format.
  1 to 400 characters (pattern data of 400 × 4 bits)
  Specify a character string consisting of 0 to 9 and A to F.
  If the last character string of the pattern data is short of a hexadecimal unit (4 bits), add 0(s) up to the last bit to make a hexadecimal string.

  #B***...* Specify pattern data in binary format.
  1 to 400 characters (pattern data of 400 bits)
  Specify a character string consisting of 0 and 1.

**Function**

Sets the mask pattern data from the `<start>` to `<end>` addresses during Block Window setting. The set data overwrites the specified range.

**Example**

To set one character of the mask pattern data from bit 0 to bit 7FFFFFFF addresses in hexadecimal format during Block Window setting:

> MGB #H0,#H7FFFFFFF,#H1

**Compatibility**

Incompatible with existing models.
### MGB? <start>

**Parameter**

- `<start>` = `<NON-DECIMAL PROGRAM DATA>`
- `#H0` to `#HF7FFFFFFF`  
  0 to `7FFFFFFF` bits, in 1-bit steps  
  (Specify in hexadecimal)

**Note:**

- The maximum setting value is the pattern length.
- In the case of Combination, the setting range is multiplied by 4.

**Response**

- `<data>` = `<HEXADECIMAL NUMERIC RESPONSE DATA>`
- `#H***...*`  
  Returns the pattern data in hexadecimal format.
  Within 400 characters (pattern data of `400 x 4` bits)

**Function**

Queries `400 x 4` bits of mask pattern data from the `<start>` address during Block Window setting.

**Example**

To query `400 x 4` bits of the mask pattern data from bit 0 address during Block Window setting:

```
> MGB? #H0  
< MGB #H1
```

**Compatibility**

Incompatible with existing models.

### CHM <data>

**Parameter**

- `<data>` = `<NON-DECIMAL PROGRAM DATA>`
- `#H***...*`  
  Specify pattern data in hexadecimal format.
  1 to 8 characters (pattern data of 32 bits)
  Specify a character string consisting of 0 to 9 and A to F.

- `#B***...*`  
  Specify pattern data in binary format.
  1 to 32 characters (pattern data of 32 bits)
  Specify a character string consisting of 0 and 1.

**Note:**

- The number of settable characters doubled at 2 ch Combination.
- The number of settable characters are quadrupled at 4 ch Combination.

**Function**

Sets the mask pattern data during Bit Window setting.

**Example**

To set one character of the mask pattern data in hexadecimal format during Bit Window setting:

```
> CHM #H1
```

**Compatibility**

Incompatible with existing models.
**Chapter 8  Native Commands**

**CHM?**

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;data&gt; = &lt;HEXADECIMAL NUMERIC RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function</td>
<td>Queries the mask pattern data during Bit Window setting</td>
</tr>
<tr>
<td>Example</td>
<td>&gt; CHM?</td>
</tr>
<tr>
<td></td>
<td>&lt; CHM #H01234567</td>
</tr>
<tr>
<td></td>
<td>&lt; CHM #H00000000</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Incompatible with existing models.</td>
</tr>
</tbody>
</table>

**BDM <start>,<end>,<bdata>**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;start&gt; = &lt;NON-DECIMAL PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>#H0 to #H7FFFFFF 0 to 7FFFFFFF bits, in 1-bit steps</td>
</tr>
<tr>
<td></td>
<td>(Specify in hexadecimal)</td>
</tr>
<tr>
<td>&lt;end&gt;</td>
<td>= &lt;NON-DECIMAL PROGRAM DATA&gt;</td>
</tr>
<tr>
<td></td>
<td>#H0 to #H7FFFFFF 0 to 7FFFFFFF bits, in 1-bit steps</td>
</tr>
<tr>
<td></td>
<td>(Specify in hexadecimal)</td>
</tr>
</tbody>
</table>

**Note:**

The maximum setting is the pattern length.
At 2 ch Combination, the setting range is doubled.
At 4 ch Combination, the setting range is quadrupled.

<table>
<thead>
<tr>
<th>&lt;bdata&gt; = &lt;ARBITRARY BLOCK PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>#XYYY&lt;binary&gt;</td>
</tr>
<tr>
<td>X: Number of YYY digits</td>
</tr>
<tr>
<td>YYY: Number of bytes of &lt;binary&gt;</td>
</tr>
<tr>
<td>1 to 16,777,216 bytes</td>
</tr>
<tr>
<td>&lt;binary&gt;: Binarydata up to 16,777,216 bytes</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Function</th>
<th>Sets binary data of the mask pattern data from &lt;start&gt; to &lt;end&gt; addresses during Block Window setting.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example</td>
<td>The set data overwrites the specified range.</td>
</tr>
<tr>
<td></td>
<td>To set binary data of the mask pattern data from bit 0 to bit 7FFFFFFF addresses during Block Window setting:</td>
</tr>
<tr>
<td></td>
<td>&gt; BDM #H0,#H7FFFFFF,#11A</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Incompatible with existing models.</td>
</tr>
</tbody>
</table>
8.4 ED Commands

BDM? <start>[,<size>]

Parameter

<start> = <NON-DECIMAL PROGRAM DATA>
#H0 to #H7FFFFFFF 0 to 7FFFFFFF bits, in 1-bit steps
(Specify in hexadecimal)

<size> = <NR1 NUMERIC PROGRAM DATA>
1 to 1677216 1 to 1677216 bytes, in 1-byte steps

Note:
The maximum setting is the pattern length.
At 2 ch Combination, the setting range is doubled.
At 4 ch Combination, the setting range is quadrupled.

Response

<bdata> = <DEFINITE LENGTH ARBITRARY BLOCK RESPONSE DATA>
#XYYY<binary>
X: Number of YYY digits
YYY: Number of bytes of <binary>
1 to 16,777,216 bytes

<binary>: Binary data of up to 16,777,216 bytes

Binary data of the setting pattern is queried when [size] is omitted.

Function

Queries binary data of bytes specified by <size> from the <start> address
of the mask pattern data during Block Window setting.

Example

To query binary data of 2 bytes from bit 0 address of the pattern data
during Block Window setting:
> BDM? #H0,2
< BDM #12AA

To query binary data of the setting pattern from bit 0 address of the
pattern data during Block Window setting (in the case of [size] is
omitted):
> BDM? #H0
< BDM #516000000000 (binary data of 16,000 bytes)

Compatibility

Incompatible with existing models.
Chapter 8  Native Commands

**DRM <start>,<end>**

**Parameter**

\[\text{<start>} = \text{<NON-DECIMAL PROGRAM DATA>}\]

#H0 to #H7FFFFFF 0 to 7FFFFFFF bits, in 1-bit steps

(Specify in hexadecimal)

\[\text{<end>} = \text{<NON-DECIMAL PROGRAM DATA>}\]

**Note:**

The maximum setting is the pattern length.

At 2 ch Combination, the setting range is doubled.

At 4 ch Combination, the setting range is quadrupled.

**Function**

Inverts the mask pattern data during Block Window setting. Specify the inversion range by the <start> and <end> addresses.

**Example**

To invert addresses from bit 0 to bit 7FFFFFF during Block Window setting:

\[> \text{DRM} \ #H0,\ #H7FFFFFF\]

**Compatibility**

Incompatible with existing models.

---

**DBM <start>,<delta>**

**Parameter**

\[\text{<start>} = \text{<NON-DECIMAL PROGRAM DATA>}\]

#H0 to #H7FFFFFF 0 to 7FFFFFFF bits, in 1-bit steps

(Specify in hexadecimal)

\[\text{<delta>} = \text{<DECIMAL NUMERIC PROGRAM DATA>}\]

1 to 134217728 1 to 134,217,728 bits, in 1-bit steps

**Note:**

The maximum setting is the pattern length.

At 2 ch Combination, the setting range is doubled.

At 4 ch Combination, the setting range is quadrupled.

**Function**

Inverts the mask pattern data during Block Window setting. Specify the inversion range by the number of bits from the <start> address (relative specification).

**Example**

To invert 1 bit of the mask pattern data from address 0 during Block Window setting:

\[> \text{DBM} \ #H0,\ 1\]

**Compatibility**

Incompatible with existing models.
### ALB <range>,<page>,<data>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;range&gt; = CHARACTER PROGRAM DATA</td>
<td>Specifies a page. (One page is defined as 128 bits.)</td>
</tr>
<tr>
<td>0</td>
<td>Specifies a page. (One page is defined as 128 bits.)</td>
</tr>
<tr>
<td>1</td>
<td>Specifies all data.</td>
</tr>
<tr>
<td>&lt;page&gt; = DECIMAL NUMERIC PROGRAM DATA</td>
<td>1 to 1,048,576 (Max.) pages, in 1-page steps Max = Pattern Length/128 (rounding up fractions) Specify &quot;0&quot; when &lt;range&gt; is set to ALL.</td>
</tr>
<tr>
<td>1</td>
<td>Fills the specified range with 1s.</td>
</tr>
</tbody>
</table>

| Function | Sets 0s or 1s for the specified page or all data in mask pattern data during Block Window setting. |

<table>
<thead>
<tr>
<th>Example</th>
<th>To set 0s for the first page of the mask pattern data during Block Window setting:</th>
</tr>
</thead>
<tbody>
<tr>
<td>&gt; ALB 0,1,0</td>
<td></td>
</tr>
</tbody>
</table>

| Compatibility | Incompatible with existing models. |
MDB <block>,<start>,<end>,<data>

Parameter

- `<block>` = <DECIMAL NUMERIC PROGRAM DATA>
  1 to 511 blocks, in 1-block steps
- `<start>` = <NON-DECIMAL PROGRAM DATA>
  #H0 to #H7FFFFFFF 0 to 7FFFFFFF bits, in 1-bit steps
  (Specify in hexadecimal)
- `<end>` = <NON-DECIMAL PROGRAM DATA>
  #H0 to #H7FFFFFFF 0 to 7FFFFFFF bits, in 1-bit steps
  (Specify in hexadecimal)

Note:
The maximum setting is the pattern length.
At 2 ch Combination, the setting range is doubled.
At 4 ch Combination, the setting range is quadrupled.

- `<data>` = <NON-DECIMAL PROGRAM DATA>
  #H***...* Specify pattern data in hexadecimal format.
  1 to 400 characters (pattern data of 400 \( \times \) 4 bits)
  Specify a character string consisting of 0 to 9 and A to F.
  If the last character string of the pattern data is short of a hexadecimal unit (4 bits), add 0(s) up to the last bit to make a hexadecimal string.
  
  #B***...* Specify pattern data in binary format.
  1 to 400 characters (pattern data of 400 bits)
  Specify a character string consisting of 0 and 1.

Function
Sets binary data of the Block Window mask pattern data `<start>` to `<end>` addresses during Mixed Data pattern setting.
The set data overwrites the specified range.

Example
To set one character of the mask pattern data from bit 0 to bit 7FFFFFFF addresses in hexadecimal format during Block Window setting:

> MDB 1,#H0,#H7FFFFFFF,#H1

Compatibility
Incompatible with existing models.
8.4 ED Commands

**MDB? <block>,<start>**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;block&gt;</td>
<td>&lt;DECIMAL NUMERIC PROGRAM DATA&gt; 1 to 511 blocks, in 1-block steps</td>
</tr>
<tr>
<td>&lt;start&gt;</td>
<td>&lt;NON-DECIMAL PROGRAM DATA&gt; #H0 to #H7FFFFFFF 0 to 7FFFFFFF bits, in 1-bit steps (Specify in hexadecimal)</td>
</tr>
</tbody>
</table>

**Note:**
- The maximum setting is the pattern length.
- At 2 ch Combination, the setting range is doubled.
- At 4 ch Combination, the setting range is quadrupled.

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;data&gt; = &lt;HEXADECIMAL NUMERIC RESPONSE DATA&gt;</th>
</tr>
</thead>
</table>

| Function | Queries 400 × 4 bits of Block Window mask pattern data from the <start> to <end> addresses during Mixed Data pattern setting. |

<table>
<thead>
<tr>
<th>Example</th>
<th>To query 400 × 4 bits of mask pattern data from bit 0 address during Block Window setting:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>&gt; MDB? 1,#H0</td>
</tr>
<tr>
<td></td>
<td>&lt; MDB #H1</td>
</tr>
</tbody>
</table>

| Compatibility | Incompatible with existing models. |

<table>
<thead>
<tr>
<th>Note:</th>
<th>The maximum setting is the pattern length.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>At 2 ch Combination, the setting range is doubled.</td>
</tr>
<tr>
<td></td>
<td>At 4 ch Combination, the setting range is quadrupled.</td>
</tr>
</tbody>
</table>
MBM <block>,<start>,<end>,<bdata>

Parameter

- <block> = <DECIMAL NUMERIC PROGRAM DATA>
  1 to 511  1 to 511 blocks, in 1-block steps
- <start> = <NON-DECIMAL PROGRAM DATA>
  #H0 to #H7FFFFFF  0 to 7FFFFFFF bits, in 1-bit steps
  (Specify in hexadecimal)
- <end> = <NON-DECIMAL PROGRAM DATA>
  #H0 to #H7FFFFFFF  0 to 7FFFFFFF bits, in 1-bit steps
  (Specify in hexadecimal)

Note:
The maximum setting is the pattern length.
At 2 ch Combination, the setting range is doubled.
At 4 ch Combination, the setting range is quadrupled.

- <bdata> = <ARBITRARY BLOCK PROGRAM DATA>
  #XYYY<binary>
  X: Number of YYY digits
  YYY: Number of bytes of <binary>
  1 to 16,777,216 bytes
  <binary>: Binary data up to 16,777,216 bytes

Function

Sets binary data of the Block Window mask pattern <start> to <end> addresses during Mixed Data pattern setting.
The set data overwrites the specified range.

Example

To set binary data of the Block Window mask pattern from bit 0 to bit 7FFFFFFF addresses during Mixed Data pattern setting or Block Window setting:

> MBM 1,#H0,#H7FFFFFFF,#11A

Compatibility

Incompatible with existing models.
**MBM? <block>,<start>[,<size>]**

**Parameter**

- `<block>` = `<DECIMAL NUMERIC PROGRAM DATA>`
  - 1 to 511 blocks, in 1-block steps
- `<start>` = `<NON-DECIMAL PROGRAM DATA>`
  - #H0 to #H7FFFFFF 0 to 7FFFFFFF bits, in 1-bit steps
  - (Specify in hexadecimal)

**Note:**

- The maximum setting is the pattern length.
- At 2 ch Combination, the setting range is doubled.
- At 4 ch Combination, the setting range is quadrupled.

- `[<size>]` = `<NR1 NUMERIC PROGRAM DATA>`
  - 1 to 1677216 1 to 1,677,216 bytes/1 byte Step

**Response**

- `<bdata>` = `<DEFINITE LENGTH ARBITRARY BLOCK RESPONSE DATA>`
  - #XYYY<binary>
    - X: Number of YYY digits
    - YYY: Number of bytes of <binary>
    - 1 to 16,777,216bytes
  - <binary>: Binary data up to 16,777,216bytes

Binary data of the setting pattern is queried when `[<size>]` is omitted.

**Function**

Queries binary data of bytes specified by `<size>` from the `<start>` address of the Block Window mask pattern during Mixed Data pattern setting.

**Example**

To query binary data of up to 2 bytes from bit 0 the `<start>` address of the Block Window mask pattern during Mixed Data pattern setting or Block Window setting:

```plaintext
> MBM? 1,#H0,2
< MBM #12AA
```

To query binary data of the setting pattern from bit 0 address of the Block Window mask pattern during Mix Data pattern setting or Block Window setting (in the case of `[<size>]` is omitted):

```plaintext
> MBM? 1,#H0
< MBM #516000AAAAAAAA... (binary data of 16,000 bytes)
```

**Compatibility**

Incompatible with existing models.
MDM <block>,<start>,<end>

Parameter

- `<block>` = `<DECIMAL NUMERIC PROGRAM DATA>`
- 1 to 511 blocks, in 1-block steps
- `<start>` = `<NON-DECIMAL PROGRAM DATA>`
- #H0 to #H7FFFFFF
- 0 to 7FFFFFFF bits, in 1-bit steps
- (Specify in hexadecimal)
- `<end>` = `<NON-DECIMAL PROGRAM DATA>`
- #H0 to #H7FFFFFF
- 0 to 7FFFFFFF bits, in 1-bit steps
- (Specify in hexadecimal)

**Note:**

- The maximum setting is the pattern length.
- At 2 ch Combination, the setting range is doubled.
- At 4 ch Combination, the setting range is quadrupled.

**Function**

Inverts the Block Window mask pattern in during Mixed Data pattern setting. Specify the inversion range by the `<start>` and `<end>` addresses.

**Example**

To invert the Block Window mask pattern from bit 0 to bit 7FFFFFFF addresses during Mixed Data pattern setting:

> MDM 1,#H0,#H7FFFFFF

**Compatibility**

Incompatible with existing models.

---

MMB <block>,<start>,<delta>

Parameter

- `<block>` = `<DECIMAL NUMERIC PROGRAM DATA>`
- 1 to 511 blocks, in 1-block steps
- `<start>` = `<NON-DECIMAL PROGRAM DATA>`
- #H0 to #H7FFFFFFF
- 0 to 7FFFFFFF bits, in 1-bit steps
- (Specify in hexadecimal)
- `<delta>` = `<DECIMAL NUMERIC PROGRAM DATA>`
- 1 to 134217728 bits, in 1-bit steps
- (Specify in hexadecimal)

**Note:**

- The maximum setting is the pattern length.
- At 2 ch Combination, the setting range is doubled.
- At 4 ch Combination, the setting range is quadrupled.

**Function**

Inverts the Block Window mask pattern during Mixed Data pattern setting. Specify the inversion range by the number of bits from the `<start>` address (relative specification).

**Example**

To invert 1 bit of the Block Window mask pattern from address 0 during Mixed Data pattern setting:

> MMB 1,#H0,1

**Compatibility**

Incompatible with existing models.
**MFB <block>,<range>,<page>,<data>**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;block&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
<td>1 to 511 1 to 511 blocks, in 1-block steps</td>
</tr>
<tr>
<td>&lt;range&gt; = &lt;CHARACTER PROGRAM DATA&gt;</td>
<td>Specifies a page. (One page is defined as 128 bits.)</td>
</tr>
<tr>
<td>0</td>
<td>Specifies a page. (One page is defined as 128 bits.)</td>
</tr>
<tr>
<td>1</td>
<td>Specifies all data.</td>
</tr>
<tr>
<td>&lt;page&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
<td>1 to 1048576 1 to 1,048,576 (Max.) pages, in 1-page steps</td>
</tr>
<tr>
<td>Max</td>
<td>Pattern Length/128 (rounding up fractions)</td>
</tr>
<tr>
<td>Specify &quot;0&quot; when &lt;range&gt; is set to ALL.</td>
<td></td>
</tr>
<tr>
<td>&lt;data&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
<td>0 Fills the specified range with 0s.</td>
</tr>
<tr>
<td>1</td>
<td>Fills the specified range with 1s.</td>
</tr>
<tr>
<td>Function</td>
<td>Sets 0s or 1s for the specified page or all data in the Block Window mask pattern during Mixed Data pattern setting.</td>
</tr>
<tr>
<td>Example</td>
<td>To set 0s for the first page of the Block Window mask pattern during Mixed Data pattern setting:</td>
</tr>
<tr>
<td></td>
<td>&gt; MFB 1,0,1,0</td>
</tr>
</tbody>
</table>

**Compatibility**

Incompatible with existing models.
MSB <block>,<start>,<end>,<data>

Parameter

- **<block>** = <DECIMAL NUMERIC PROGRAM DATA>
  1 to 128 1 to 128 blocks, in 1-block steps
- **<start>** = <NON-DECIMAL PROGRAM DATA>
  #H0 to #HFFFFF 0 to FFFFF bits, in 1-bit steps
  (Specify in hexadecimal)
- **<end>** = <NON-DECIMAL PROGRAM DATA>
  #H0 to #HFFFFF 0 to FFFFF bit, in 1-bit steps
  (Specify in hexadecimal)

**Note:**
The maximum setting value is the pattern length.

- **<data>** = <NON-DECIMAL PROGRAM DATA>
  #H***...* Specify pattern data in hexadecimal format.
  1 to 400 characters (pattern data of 400 × 4 bits)
  Specify a character string consisting of 0 to 9 and A to F.
  If the last character string of the pattern data is short of a hexadecimal unit (4 bits), add 0(s) up to the last bit to make a hexadecimal string.
  #B***...* Specify pattern data in binary format.
  1 to 400 characters (pattern data of 400 bits)
  Specify a character string consisting of 0 and 1.

**Function**
Sets the mask pattern data from the <start> to <end> addresses during Sequence pattern setting and Block Window setting. The set data overwrites the specified range.

**Example**
To set one character of the mask pattern data from bit 0 to bit FFFFFF addresses in hexadecimal format during Block Window setting:
> MSB 1,#H0,#HFFFF,#H1

**Compatibility**
Incompatible with existing models.
### MSB? <block>,<start>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;block&gt;</td>
<td>= &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
</tr>
<tr>
<td></td>
<td>1 to 128 blocks, in 1-block steps</td>
</tr>
<tr>
<td>&lt;start&gt;</td>
<td>= &lt;NON-DECIMAL PROGRAM DATA&gt;</td>
</tr>
<tr>
<td></td>
<td>#H0 to #HFFFFF 0 to FFFFFF bits, in 1-bit steps</td>
</tr>
</tbody>
</table>

**Note:**
The maximum setting value is the pattern length.

| Response | <data> = <HEXADECIMAL NUMERIC RESPONSE DATA> |

| Function | Queries 400 × 4 bits of masked pattern data from the <start> address during Sequence pattern setting and Block Window setting. |

| Example | To query 400 × 4 bits of masked pattern data from bit 0 address during Block Window setting: |
|         | > MSB? 1,#H0 |
|         | < MSB #1 |

| Compatibility | Incompatible with existing models. |

**Example**

To query 400 × 4 bits of masked pattern data from bit 0 address during Block Window setting:

```
> MSB? 1,#H0
< MSB #1
```
**SBM <block>,<start>,<end>,<bdata>**

**Parameter**
- `<block>` = `< DECIMAL NUMERIC PROGRAM DATA>`
  - 1 to 128
  - 1 to 128 blocks, in 1-block steps
- `<start>` = `<NON-DECIMAL PROGRAM DATA>`
  - `#H0` to `#HFFFFF`
  - 0 to `FFFFF` bits, in 1-bit steps
  - (Specify in hexadecimal)
- `<end>` = `<NON-DECIMAL PROGRAM DATA>`
  - `#H0` to `#HFFFFF`
  - 0 to `FFFFF` bits, in 1-bit steps
  - (Specify in hexadecimal)

**Note:**
The maximum setting value is the pattern length.

- `<bdata>` = `<ARBITRARY BLOCK PROGRAM DATA>`
  - `#XYY<binary>`
    - X: Number of `YYY` digits
    - YYY: Number of bytes of `<binary>`
    - 1 to 131,072 bytes
    - `<binary>`: Binary data up to 131,072bytes

**Function**
Sets binary data of the Block Window mask pattern from `<start>` to `<end>` addresses during Sequence pattern setting.
The set data overwrites the specified range.

**Example**
To set binary data of the Block Window mask pattern 3 from bit 0 to bit 7`FFFFF` addresses during Sequence pattern setting:
```
> SBM 1,#H0,#H7FFFFF,#11A
```

**Compatibility**
Incompatible with existing models.
**SBM? <block>,<start>[,<size>]**

**Parameter**

- `<block>` = `<DECIMAL NUMERIC PROGRAM DATA>`
  1 to 128 1 to 128 Blocks, in 1-block steps
- `<start>` = `<NON-DECIMAL PROGRAM DATA>`
  #H0 to #HFFFFF 0 to FFFFF bits, in 1-bit steps
  (Specify in hexadecimal)

**Note:**
The maximum setting value is the pattern length.

- `[<size>]` = `<NR1 NUMERIC PROGRAM DATA>`
  1 to 131072 1 to 131,072 bytes/1 byte Step

**Response**

- `<bdata>` = `<DEFINITE LENGTH ARBITRARY BLOCK RESPONSE DATA>`
  #XYYY<binary>
  X: Number of YYY digits
  YYY: Number of bytes of `<binary>`
  1 to 131,072 bytes
  `<binary>`: Binary data up to 131,072 bytes

Binary data of the setting pattern is queried when `[<size>]` is omitted.

**Function**
Queries binary data of bytes specified by `<size>` from the `<start>` address of the Block Window mask pattern during Sequence pattern setting.

**Example**
To query binary data of up to 2 bytes from bit 0 the `<start>` address of the Block Window mask pattern during Sequence pattern setting:

```shell
> SBM? 1,#H0,2
< SBM #12AA
```

To query binary data of the setting pattern from bit 0 address of the Block Window mask pattern during Sequence pattern setting (in the case of `[<size>]` is omitted):

```shell
> SBM? 1,#H0
< SBM #516000AAAAAA... (binary data of 16,000 bytes)
```

**Compatibility**
Incompatible with existing models.
**SDM <block>,<start>,<end>**

**Parameter**
- `<block>` = `<DECIMAL NUMERIC PROGRAM DATA>`
  1 to 128 1 to 128 blocks, in 1-block steps
- `<start>` = `<NON-DECIMAL PROGRAM DATA>`
  #H0 to #HFFFFF 0 to FFFFF bits, in 1-bit steps
  (Specify in hexadecimal)
- `<end>` = `<NON-DECIMAL PROGRAM DATA>`
  #H0 to #HFFFFF 0 to FFFFF bits, in 1-bit steps
  (Specify in hexadecimal)

**Note:**
The maximum setting value is the pattern length.

**Function**
Inverts the Block Window mask pattern during Sequence pattern setting.
Specify the inversion range by the `<start>` and `<end>` addresses.

**Example**
To invert the Block Window mask pattern from bit 0 to bit FFFFF addresses during Sequence pattern setting:
> SDM 1,#H0,#HFFFFF

**Compatibility**
Incompatible with existing models.

---

**SDB <block>,<start>,<delta>**

**Parameter**
- `<block>` = `<DECIMAL NUMERIC PROGRAM DATA>`
  1 to 128 1 to 128 blocks, in 1-block steps
- `<start>` = `<NON-DECIMAL PROGRAM DATA>`
  #H0 to #HFFFFF 0 to FFFFF bits, in 1-bit steps
  (Specify in hexadecimal)
- `<delta>` = `<DECIMAL NUMERIC PROGRAM DATA>`
  1 to 1048576 1 to 1,048,576 bits, in 1-bit steps

**Note:**
The maximum setting value is the pattern length.

**Function**
Inverts the Block Window mask pattern during Sequence pattern setting.
Specify the inversion range by the number of bits from the `<start>` address (relative specification).

**Example**
To invert 1 bit of the Block Window mask pattern from address 0 during Sequence pattern setting:
> SDB 1,#H0,1

**Compatibility**
Incompatible with existing models.
**SMF <block>,<range>,<page>,<data>**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;block&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
<td>1 to 128 Blocks, in 1-block steps</td>
</tr>
<tr>
<td>&lt;range&gt; = &lt;CHARACTER PROGRAM DATA&gt;</td>
<td>Specifies a page. (One page is defined as 128 bits.) Specifies all data.</td>
</tr>
<tr>
<td>&lt;page&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
<td>1 to 8192 (Max.) pages, in 1-page steps</td>
</tr>
<tr>
<td>&lt;data&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
<td>Fills the specified range with 0s. Fills the specified range with 1s.</td>
</tr>
</tbody>
</table>

**Function**
Sets 0s or 1s for the specified page or all data in Block Window mask pattern during Sequence pattern setting.

**Example**
To set 0s for the first page of the Block Window mask pattern during Sequence pattern setting:

```
> SMF 1,0,1,0
```

**Compatibility**
Incompatible with existing models.
8.4.4 Commands related to Input tab

Figure 8.4.4-1 Input tab setting
8.4.4.1 Data Input setting commands

Figure 8.4.4.1-1 Single-End Data

Figure 8.4.4.1-2 Differential 50 Ohm
Chapter 8 Native Commands

Table 8.4.4.1-1 Data Input setting commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1]</td>
<td>Input Condition</td>
<td>DSD</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DCD?</td>
</tr>
<tr>
<td>[2]</td>
<td>Differential type setting</td>
<td>DAC</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DCD?</td>
</tr>
<tr>
<td>[3]</td>
<td>Data input port setting</td>
<td>DSL</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DCD?</td>
</tr>
<tr>
<td>[4]</td>
<td>Data Threshold</td>
<td>DTH</td>
</tr>
<tr>
<td></td>
<td>XData Threshold</td>
<td>DTH?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DDB?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DDT?</td>
</tr>
<tr>
<td>[7]</td>
<td>Termination</td>
<td>DTM</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DTM?</td>
</tr>
<tr>
<td>[8]</td>
<td>No label</td>
<td>DTL</td>
</tr>
<tr>
<td></td>
<td>(Termination voltage setting)</td>
<td>DTL?</td>
</tr>
</tbody>
</table>
### DSD <interface>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;interface&gt;</td>
<td>= &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
</tr>
<tr>
<td>0</td>
<td>Single Ended</td>
</tr>
<tr>
<td>1</td>
<td>Differential 50 Ω</td>
</tr>
<tr>
<td>2</td>
<td>Differential 100 Ω</td>
</tr>
</tbody>
</table>

**Function**: Sets the data input interface.

**Example**: To set the data input interface to Single Ended:

```
> DSD 0
```

**Compatibility**: Partially compatible with the MP1762D (0: Single Ended is compatible).

### DAC <tracking>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;tracking&gt;</td>
<td>= &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
</tr>
<tr>
<td>0</td>
<td>Independent</td>
</tr>
<tr>
<td>1</td>
<td>Tracking</td>
</tr>
<tr>
<td>2</td>
<td>Alternate</td>
</tr>
</tbody>
</table>

**Function**: Sets the differential type when the data input interface is Differential.

**Example**: To set the differential type to independent setting:

```
> DAC 0
```

**Compatibility**: Partially compatible with the MP1762D (0: Independent and 1: Tracking are compatible).

### DSL <port>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;port&gt;</td>
<td>= &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
</tr>
<tr>
<td>0</td>
<td>Data</td>
</tr>
<tr>
<td>1</td>
<td>XData</td>
</tr>
</tbody>
</table>

**Function**: Switches between types of data input interfaces (Data, XData). Switches the data input interface when set to Single Ended and selects the input port when set to Differential

**Example**: To switch the data input interface to Data:

```
> DSL 0
```

**Compatibility**: Compatible with the MP1762D.
**Chapter 8  Native Commands**

### DCD?

**Response**

- `<port>` = `<NR1 NUMERIC RESPONSE DATA> (FIX1)
  - 0  Data
  - 1  XData
- `<tracking>` = `<NR1 NUMERIC RESPONSE DATA> (FIX1)
  - 0  Independent (Data/XData independent setting)
  - 1  Tracking (Data/XData common setting)
  - 2  Alternate (Data/XData differential setting)
- `<interface>` = `<NR1 NUMERIC RESPONSE DATA> (FIX1)
  - 0  Single Ended
  - 1  Differential 50 Ω
  - 2  Differential 100 Ω

**Function**

Queries the data input interface.

**Example**

```plaintext
> DCD?
< DCD 0,0,0
```

**Compatibility**

Incompatible with existing models.

### DTH <numeric>

**Parameter**

- `<numeric>` = `<DECIMAL NUMERIC PROGRAM DATA> –3.500 to 3.300 V, in 0.001-V steps (when MU181040A-002 and MU181040B-002 are installed)
- –0.350 to 0.350 V, in 0.001-V steps (when MU181040A-001 is installed)

**Function**

Sets the data input threshold value for the displayed and set port.

**Example**

To set the data input threshold value for the displayed and set port to –3.000 V:

```plaintext
> DTH -3.000
```

**Compatibility**

Partially compatible with the MP1632C (within the range of –3.000 to 3.300 V, in 0.001-V steps).

Compatible with the MP1762D.

### DTH?

**Response**

- `<numeric>` = `<NR2 NUMERIC RESPONSE DATA> (FIX6)

**Function**

Queries the data input threshold value for the displayed and set port.

**Example**

```plaintext
> DTH?
< DTH -3.000
< DTH 3.300
```

**Compatibility**

Partially compatible with the MP1632C (within the range of –3.000 to 3.300 V, in 0.001-V steps).

Compatible with the MP1762D.
### DDB <basis>

**Parameter**

\[ <\text{basis}> = \text{<DECIMAL NUMERIC PROGRAM DATA>} \]

- 0: Based on Data
- 1: Based on XData

**Function**

Sets the differential reference signal for the data input threshold value.

**Example**

To set the differential reference signal for the data input threshold value to be based on Data:

\[ > \text{DDB 0} \]

**Compatibility**

Incompatible with existing models.

### DDB?

**Response**

\[ <\text{basis}> = \text{<NR1 NUMERIC RESPONSE DATA>} \text{ (FIX1)} \]

**Function**

Queries the differential reference signal for the data input threshold value.

**Example**

\[ > \text{DDB?} \]
\[ <\text{DDB 0} \]

**Compatibility**

Incompatible with existing models.

### DDT <numeric>

**Parameter**

\[ <\text{numeric}> = \text{<DECIMAL NUMERIC PROGRAM DATA>} \]

- –3.000 to 3.000: –3.000 to 3.300 V, in 0.001-V steps (when MU181040A–002 and MU181040B-002 are installed)
- –0.700 to 0.700: –0.700 to 0.700 V, in 0.001-V steps (when MU181040A-001 is installed)

**Function**

Sets the data input threshold value for differential input.

**Example**

To set the differential data input threshold value to –3.000 V:

\[ > \text{DDT -3.000} \]

**Compatibility**

Incompatible with existing models.

### DDT?

**Response**

\[ <\text{numeric}> = \text{<NR2 NUMERIC RESPONSE DATA>} \text{ (FIX6)} \]

**Function**

Queries the data input threshold value for differential input.

**Example**

\[ > \text{DDT?} \]
\[ <\text{DDT -3.000} \]
\[ <\text{DDT 3.000} \]

**Compatibility**

Incompatible with existing models.
**Chapter 8  Native Commands**

**DTM <term>**

Parameter: 
<term> = <DECIMAL NUMERIC PROGRAM DATA>

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>GND (50 Ω to GND)</td>
</tr>
<tr>
<td>1</td>
<td>NECL (50 Ω to –2 V)</td>
</tr>
<tr>
<td>3</td>
<td>LVPECL (50 Ω to 1.3 V)</td>
</tr>
<tr>
<td>4</td>
<td>VARIABLE (50 Ω to Variable Voltage)</td>
</tr>
<tr>
<td>5</td>
<td>PCML (50 Ω to 3.3 V)</td>
</tr>
</tbody>
</table>

Function: Sets the data input termination condition.

Example: To set the data input termination condition to GND:

> DTM 0

Compatibility: Partially compatible with the MP1762D (0: GND).

**DTM?**

Response: 
<term> = <NR1 NUMERIC RESPONSE DATA> (FIX1)

Function: Queries the data input termination condition.

Example: To query the data input termination condition:

> DTM?

< DTM 0

Compatibility: Partially compatible with the MP1762D (0: GND).

**DTL <numeric>**

Parameter: 
<numeric> = <DECIMAL NUMERIC PROGRAM DATA>

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>–2.50</td>
<td>to 3.50</td>
</tr>
<tr>
<td>–2.50</td>
<td>to 3.50 V, in 0.01-V steps</td>
</tr>
</tbody>
</table>

Function: Sets the termination voltage when the input termination condition is set to Variable. The setting is impossible when set to "Differential 100 Ω".

Example: To set the termination voltage when the input termination condition is set to Variable to –2.00 V:

> DTL -2.00

Compatibility: Incompatible with existing models.

**DTL?**

Response: 
<numeric> = <NR2 NUMERIC RESPONSE DATA> (FIX5)

Function: Queries the termination voltage when the input termination condition is set to Variable.

Example: To query the termination voltage:

> DTL?

< DTL -2.00

< DTL 3.50

Compatibility: Incompatible with existing models.
8.4.4.2 Clock Input setting commands

Figure 8.4.4.2-1 External Clock setting

Figure 8.4.4.2-2 Recovered Clock setting

Figure 8.4.4.2-3 Clock Termination setting dialog box
### Table 8.4.4.2-1  Clock Input setting commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1]</td>
<td>Selection</td>
<td>RRC</td>
</tr>
<tr>
<td>[2]</td>
<td>Setting for specific frequency of clock recovery</td>
<td>RFQ</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DRC?</td>
</tr>
<tr>
<td>[3]</td>
<td>Setting for clock recovery frequency</td>
<td>CFQ</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CFQ?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CTM?</td>
</tr>
<tr>
<td>[5]</td>
<td>No label (Termination voltage setting)</td>
<td>CTL</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CTL?</td>
</tr>
<tr>
<td>[6]</td>
<td>Polarity</td>
<td>CRC</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CRC?</td>
</tr>
<tr>
<td>[7]</td>
<td>Delay (mUI setting)</td>
<td>CPA</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CPA?</td>
</tr>
<tr>
<td></td>
<td>Delay (ps setting)</td>
<td>CPU</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CPU?</td>
</tr>
<tr>
<td>[8]</td>
<td>Relative</td>
<td>CRO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CRO?</td>
</tr>
<tr>
<td>[9]</td>
<td>No label (Relative value setting)</td>
<td>CRD</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CRD?</td>
</tr>
<tr>
<td>[10]</td>
<td>Calibration</td>
<td>CCA</td>
</tr>
<tr>
<td>[11]</td>
<td>No label (LED)</td>
<td>DLY?</td>
</tr>
<tr>
<td>[12]</td>
<td>Jitter Input</td>
<td>CJO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CJO?</td>
</tr>
</tbody>
</table>
### RRC <sel>

**Parameter**
- `<sel>` = `<DECIMAL NUMERIC PROGRAM DATA>`
- 0: External Clock
- 1: Recovered Clock

**Function**
Sets the clock input type.

**Example**
To set the clock input type to the External clock:

```
> RRC 0
```

**Compatibility**
Compatible with the MP1762D.

### RFQ <freq>

**Parameter**
- `<freq>` = `<DECIMAL NUMERIC PROGRAM DATA>`
- 0: OC3/STM1 0.155520 Gbit/s
- 1: OC12/STM4 0.622080 Gbit/s
- 2: 1GFC 1.062500 Gbit/s
- 3: GbE 1.250000 Gbit/s
- 4: SATA 1.5Gb/s 1.500000 Gbit/s
- 5: 2GFC 2.125000 Gbit/s
- 6: OC48/STM16 2.488320 Gbit/s
- 7: PCI Express I 2.500000 Gbit/s
- 8: OTU1 2.666060 Gbit/s
- 9: SATA 3Gb/s 3.000000 Gbit/s
- 10: XAUI 3.125000 Gbit/s
- 11: 4GFC 4.250000 Gbit/s
- 12: PCI Express II 5.000000 Gbit/s
- 14: OC192/STM64 9.953280 Gbit/s
- 15: 10GbE 10.312500 Gbit/s
- 16: 10GFC 10.518750 Gbit/s
- 17: G975 FEC 10.664228 Gbit/s
- 18: OTU2 10.709225 Gbit/s
- 19: 10GbE over FEC 11.095700 Gbit/s
- 20: 10GFC over FEC 11.316800 Gbit/s
- 21: Variable Variable

**Function**
Sets the rated frequency when the clock input type is the recovery clock.

**Example**
To set the rated frequency when the clock input type is the recovery clock to 0.155520 Gbit/s:

```
> RFQ 0
```

**Compatibility**
Partially compatible with the MP1762D.
Chapter 8  Native Commands

DRC?
Response  
<sel> = <NR1 NUMERIC RESPONSE DATA> (FIX1)  
0   External Clock  
1   Recovered Clock  

<freq> = <NR1 NUMERIC RESPONSE DATA> (FIX2)  
0 to 21  
Omitted when <sel> is 0.  

Function  
Queries the input type and the rated frequency of clock input.  

Example  
> DRC?  
< DRC 0  
< DRC 1, 1  
< DRC 1,21  

Compatibility  
Compatible with the MP1762D.  

CFQ <numeric>
Parameter  
<numeric> = <NR2 NUMERIC PROGRAM DATA>  
0.100000, 0.125000, 0.140600, 0.155520, 0.156300, 0.171900,  
0.187500, 0.200000, 0.250000, 0.281300, 0.312500, 0.343800,  
0.375000, 0.400000, 0.500000, 0.562500, 0.622080, 0.625000,  
0.687500, 0.750000, 0.800000, 1.000000, 1.062500, 1.125000,  
1.250000, 1.375000, 1.500000, 1.600000, 2.000000, 2.125000,  
2.488320, 2.250000, 2.500000, 2.666060, 2.750000, 3.000000,  
3.125000, 3.200000, 4.250000  
Unit is Gbit/s.  
4.900000 to 6.250000  
9.800000 to 12.500000  
4.900000 to 6.250000 Gbit/s /1 kbit/s Step  
9.800000 to 12.500000 Gbit/s /1 kbit/s Step  
Note, however:  
When MU181040A-001 is installed: 9.8 to 12.5 Gbit/s  
When MU181040A-002 or MU181040A-x22 is installed: 0.1 to 12.5 Gbit/s  
When MU181040B-002 or MU181040B-x22 is installed: 0.1 to 14 Gbit/s  

Function  
Sets the frequency for the clock data recovery (for MU181040A-x20,  
MU181040B-x20).  

Example  
To set the frequency for the clock data recovery to 0.100000 Gbit/s:  
> CFQ 0.100000  

Compatibility  
Incompatible with existing models.
8.4 ED Commands

**CFQ?**

<table>
<thead>
<tr>
<th>Response</th>
<th>(&lt;\text{numeric}&gt; = \langle\text{NR2 NUMERIC RESPONSE DATA}\rangle) (\text{FIX9})</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function</td>
<td>Queries the frequency for the clock data recovery.</td>
</tr>
<tr>
<td>Example</td>
<td>(&gt;\text{CFQ})</td>
</tr>
<tr>
<td></td>
<td>(&lt;\text{CFQ} \ 0.100000)</td>
</tr>
<tr>
<td></td>
<td>(&lt;\text{CFQ} \ 12.500000)</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Incompatible with existing models.</td>
</tr>
</tbody>
</table>

**CTM <term>**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>(&lt;\text{term}&gt; = \langle\text{NR1 NUMERIC PROGRAM DATA}\rangle)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>GND (50 (\Omega) to GND)</td>
</tr>
<tr>
<td>1</td>
<td>NECL (50 (\Omega) to (-2) (\text{V}))</td>
</tr>
<tr>
<td>3</td>
<td>LVPECL (50 (\Omega) to 1.3 (\text{V}))</td>
</tr>
<tr>
<td>4</td>
<td>VARIABLE (50 (\Omega) to Variable Voltage)</td>
</tr>
<tr>
<td>5</td>
<td>PCML (50 (\Omega) to 3.3 (\text{V}))</td>
</tr>
<tr>
<td>Function</td>
<td>Sets the clock input termination condition.</td>
</tr>
<tr>
<td>Example</td>
<td>To set the clock input termination condition to GND:</td>
</tr>
<tr>
<td></td>
<td>(&gt;\text{CTM} \ 0)</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Partially compatible with the MP1632C and MP1762D (0: GND, 1: ECL).</td>
</tr>
</tbody>
</table>

**CTM?**

<table>
<thead>
<tr>
<th>Response</th>
<th>(&lt;\text{term}&gt; = \langle\text{NR1 NUMERIC RESPONSE DATA}\rangle) (\text{FIX1})</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function</td>
<td>Queries the clock input termination condition.</td>
</tr>
<tr>
<td>Example</td>
<td>(&gt;\text{CTM})</td>
</tr>
<tr>
<td></td>
<td>(&lt;\text{CTM} \ 0)</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Partially compatible with the MP1632C and MP1762D (0: GND, 1: ECL).</td>
</tr>
</tbody>
</table>

**CTL <numeric>**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>(&lt;\text{numeric}&gt; = \langle\text{DICIMAL NUMERIC PROGRAM DATA}\rangle)</th>
</tr>
</thead>
<tbody>
<tr>
<td>–2.50 to 3.50</td>
<td>–2.50 to 3.50 (\text{V}), in 0.01 steps</td>
</tr>
<tr>
<td>Function</td>
<td>Sets the termination voltage when the input termination condition is set to Variable.</td>
</tr>
<tr>
<td>Example</td>
<td>To set the termination voltage when the input termination condition is set to Variable to (-2.00) (\text{V}):</td>
</tr>
<tr>
<td></td>
<td>(&gt;\text{CTL} \ -2.00)</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Incompatible with existing models.</td>
</tr>
</tbody>
</table>
Chapter 8  Native Commands

CTL?

Response  \( <\text{numeric}> = <\text{NR2 NUMERIC RESPONSE DATA}> \) (FIX5)

Function  Queries the termination voltage when the input termination condition is set to Variable.

Example  
\[
\begin{align*}
> & \text{CTL} \\
< & \text{CTL} -2.00 \\
< & \text{CTL} 3.00
\end{align*}
\]

Compatibility  Incompatible with existing models.

CRC \(<\text{logic}>\)

Parameter  \( <\text{logic}> = <\text{NR1 NUMERIC PROGRAM DATA}> \)

\[
\begin{align*}
0 & \quad \text{Positive} \\
1 & \quad \text{Negative}
\end{align*}
\]

Function  Selects the phase unit for the clock recovery clock.

Example  To set the phase unit for the clock recovery clock to Positive:
\[
\begin{align*}
> & \text{CRC} 0
\end{align*}
\]

Compatibility  Incompatible with existing models.

CRC?

Response  \( <\text{logic}> = <\text{NR1 NUMERIC RESPONSE DATA}> \) (FIX1)

Function  Queries the phase unit for the clock recovery clock.

Example  
\[
\begin{align*}
> & \text{CRC} \\
< & \text{CRC} 0
\end{align*}
\]

Compatibility  Incompatible with existing models.

CPA \(<\text{numeric}>\)

Parameter  \( <\text{numeric}> = <\text{DECIMAL NUMERIC PROGRAM DATA}> \)

\[
\begin{align*}
-10000 & \text{ to } 10000 \quad -10,000 \text{ to } 10,000 \text{ ps, in } 10\text{-ps steps (at 0.1 GHz)} \\
-1000.0 & \text{ to } 1000.0 \quad -1,000 \text{ to } 1,000 \text{ ps, in } 1\text{-ps steps (at 1 GHz)} \\
-80.00 & \text{ to } 80.00 \quad -80 \text{ to } 80 \text{ ps, in } 0.08\text{-ps steps (at 12.5 GHz)}
\end{align*}
\]

The setting range and resolution (steps) are converted from the calculated frequency value, based on the setting resolution of mUI.

Function  Sets the value of the clock input phase in ps units (only when MU181040A-x30 and MU181040B-x30 are installed). Values which cannot be set due to the limitation of the setting resolution are rounded off to the nearest settable value.

Example  To set the clock input phase to \(-10,000\) ps in ps units:
\[
\begin{align*}
> & \text{CPA} -10000
\end{align*}
\]

Compatibility  Compatible with the MP1632C and MP1762D.
### CPA?

<table>
<thead>
<tr>
<th>Response</th>
<th>(&lt;\text{numeric}&gt; = \langle\text{NR1 NUMERIC RESPONSE DATA}\rangle) (&lt;\langle\text{NR2 NUMERIC RESPONSE DATA}\rangle \ (\text{FIX7}))</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function</td>
<td>Queries value of the clock input phase in ps units.</td>
</tr>
</tbody>
</table>
| Example  | > CPA?  
|          | < CPA \(-1000.0\)  
|          | > CPA \(10000\)  |
| Compatibility | Compatible with the MP1632C and MP1762D. |

### CPU <numeric>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>(&lt;\text{numeric}&gt; = \langle\text{DECIMAL NUMERIC PROGRAM DATA}\rangle) (-1000 \text{ to } 1000) (-1,000 \text{ to } 1,000 \text{ mUI, in } 1\text{-mUI steps})</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function</td>
<td>Sets the value of the clock input phase in mUI units (only when MU181040A-x30 and MU181040B-x30 are installed).</td>
</tr>
</tbody>
</table>
| Example   | To set the clock input phase to \(-1,000 \text{ mUI}\) in mUI units:  
|           | > CPU \(-1000\)  |
| Compatibility | Incompatible with existing models. |

### CPU?

<table>
<thead>
<tr>
<th>Response</th>
<th>(&lt;\text{numeric}&gt; = \langle\text{NR1 NUMERIC RESPONSE DATA}\rangle \ (\text{FIX5}))</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function</td>
<td>Queries the value of the clock input phase in mUI units.</td>
</tr>
</tbody>
</table>
| Example  | > CPU?  
|          | < CPU \(-1000\)  
|          | < CPU \(1000\)  |
| Compatibility | Incompatible with existing models. |

### CRO <numeric>

| Parameter | \(<\text{numeric}> = \langle\text{DECIMAL NUMERIC PROGRAM DATA}\rangle\)  
| 0 | Reference OFF  
| 1 | Reference ON  |
| Function | Sets the clock input phase variable reference ON/OFF. |
| Example | To set the clock input phase variable reference ON:  
|         | > CRO \(1\)  |
| Compatibility | Incompatible with existing models. |
Chapter 8  Native Commands

CRO?

Response  
\(<\text{numeric}> = \langle\text{NR1 NUMERIC RESPONSE DATA}\rangle\) (\text{FIX1})

Function  
Queries the clock input phase variable reference setting.

Example  
\[\text{CRO?}\]
\[\text{CRO 1}\]

Compatibility  
Incompatible with existing models.

CRD <numeric>[,<unit>]

Parameter  
\(<\text{numeric}> = \langle\text{DECIMAL NUMERIC PROGRAM DATA}\rangle\)

- In mUI units: \(-2000\) to \(2000\), \(-2,000\) to \(2,000\) mUI, in \(1\)-mUI steps
- In ps units: Converted from the calculated frequency value, based on the setting resolution of mUI.
- \(-20000\) to \(20000\), \(-20,000\) to \(20,000\) ps, in \(10\)-ps steps (at \(0.1\) GHz)
- \(-2000.0\) to \(2000.0\), \(-2,000\) to \(2,000\) ps, in \(1\)-ps steps (at \(1\) GHz)
- \(-160.00\) to \(160.00\), \(-160\) to \(160\) ps, in \(0.08\)-ps steps (at \(12.5\) GHz)

\[<\text{unit}> = \langle\text{NR1 NUMERIC PROGRAM DATA}\rangle\]

- 0 mUI unit
- 1 ps unit (The ps unit is selected when \(<\text{unit}>\) is omitted.)

Function  
Sets the value and unit of the clock input phase variable reference. Set a value using a difference from the reference value. The specified value may not be exactly set as is when the ps unit is set, due to the limitation of the setting resolution. The value is adjusted to the most approximate value and set in this event.

Example  
To set the clock input phase variable reference to \(-1,000\) mUI:
\[\text{CRD} -1000, 0\]

Compatibility  
Incompatible with existing models.

CRD? [<unit>]

Parameter  
\[<\text{unit}> = \langle\text{NR1 NUMERIC PROGRAM DATA}\rangle\]

- 0 mUI unit
- 1 ps unit (The ps unit is selected when \(<\text{unit}>\) is omitted.)

Response  
\(<\text{numeric}> = \langle\text{NR1 NUMERIC RESPONSE DATA}\rangle\)
\\(<\text{NR2 NUMERIC RESPONSE DATA}\rangle\) (\text{FIX7})

Function  
Queries the value of the clock input phase variable reference.

Example  
\[\text{CRD? 0}\]
\[\text{CRD -1000}\]
\[\text{CRD -160.00}\]

Compatibility  
Incompatible with existing models.
### CCA

**Function**
Calibrates the input clock phase setting value.

**Example**
> CCA

**Compatibility**
Incompatible with existing models.

### DLY?

**Response**

<table>
<thead>
<tr>
<th>numeric</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Ready status</td>
</tr>
<tr>
<td>1</td>
<td>Busy status</td>
</tr>
</tbody>
</table>

**Function**
Displays the operation status of the clock input phase setting delay.

**Example**
> DLY?
< DLY 0

**Compatibility**
Compatible with the MP1632C and MP1762D.

### CJO <numeric>

**Parameter**

<table>
<thead>
<tr>
<th>numeric</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Jitter input OFF setting</td>
</tr>
<tr>
<td>1</td>
<td>Jitter input ON setting</td>
</tr>
</tbody>
</table>

**Function**
When a jitter is added to the input signal, the Feedback process for Delay must be set to OFF.

When a jitter is added, set the Jitter Input button to ON, using this command. When a jitter is not added, set the Jitter Input button to OFF.

**Example**
To set the Jitter Input button to ON when a jitter is added:
> CJO 1

**Compatibility**
Incompatible with existing models.

### CJO?

**Response**

<table>
<thead>
<tr>
<th>numeric</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Jitter input OFF setting</td>
</tr>
<tr>
<td>1</td>
<td>Jitter input ON setting</td>
</tr>
</tbody>
</table>

**Function**
Queries the Jitter Input button setting.

**Example**
> CJO?
< CJO 1

**Compatibility**
Incompatible with existing models.
8.4.5 Commands related to Capture tab

Figure 8.4.5-1 Capture tab (initial state)

Figure 8.4.5-2 Condition Setting
### 8.4 ED Commands

**Figure 8.4.5-3** Setting for Capture Data reading

**Table 8.4.5-1** Capture tab setting commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1]</td>
<td>Capture</td>
<td>CSA</td>
</tr>
<tr>
<td>[2]</td>
<td>Capture</td>
<td>CSP</td>
</tr>
<tr>
<td>[3]</td>
<td>Trigger</td>
<td>CTG</td>
</tr>
<tr>
<td>[4]</td>
<td>Number of Block</td>
<td>CBC</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CBC?</td>
</tr>
<tr>
<td>[5]</td>
<td>Trigger</td>
<td>BTG</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BTG?</td>
</tr>
<tr>
<td>[6]</td>
<td>Position</td>
<td>CPS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CPS?</td>
</tr>
<tr>
<td></td>
<td>Length</td>
<td>CMP?</td>
</tr>
<tr>
<td>[8]</td>
<td>Match Pattern</td>
<td>MPE</td>
</tr>
<tr>
<td></td>
<td>Length</td>
<td>MPE?</td>
</tr>
<tr>
<td>[9]</td>
<td>Mask Pattern</td>
<td>CMK</td>
</tr>
<tr>
<td></td>
<td>Length</td>
<td>CMK?</td>
</tr>
<tr>
<td>[10]</td>
<td>Query for capture data block count</td>
<td>CAB?</td>
</tr>
<tr>
<td>[12]</td>
<td>Query for capture data acquisition state</td>
<td>CAT?</td>
</tr>
<tr>
<td>[13]</td>
<td>Cancel of capture data acquisition</td>
<td>CAC</td>
</tr>
<tr>
<td>[14]</td>
<td>Query for capture data length per block</td>
<td>CBL</td>
</tr>
<tr>
<td>[15]</td>
<td>Query for capture result data</td>
<td>CDW?</td>
</tr>
<tr>
<td>[16]</td>
<td>Query for capture result data</td>
<td>CBW?</td>
</tr>
<tr>
<td>[17]</td>
<td>Query for error position in capture result</td>
<td>CED?</td>
</tr>
<tr>
<td>[18]</td>
<td>Query for error position in capture result</td>
<td>CEB?</td>
</tr>
</tbody>
</table>
### Chapter 8  Native Commands

#### CSA

<table>
<thead>
<tr>
<th>Function</th>
<th>Starts capture result acquisition.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example</td>
<td>&gt; CSA</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Incompatible with existing models.</td>
</tr>
</tbody>
</table>

#### CSP

<table>
<thead>
<tr>
<th>Function</th>
<th>Stops capture result acquisition.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example</td>
<td>&gt; CSP</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Incompatible with existing models.</td>
</tr>
</tbody>
</table>

#### CTG

<table>
<thead>
<tr>
<th>Function</th>
<th>Generates a capture manual trigger.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example</td>
<td>&gt; CTG</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Incompatible with existing models.</td>
</tr>
</tbody>
</table>

#### CBC <numeric>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>(&lt;\text{numeric}) = &lt;\text{DECIMAL NUMERIC PROGRAM DATA}&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1, 2, 4, 8, 16, 32, 64, 128 Capture block division number</td>
</tr>
<tr>
<td>Function</td>
<td>Sets the capture block division number.</td>
</tr>
<tr>
<td>Example</td>
<td>To set the capture block division number to 1:</td>
</tr>
<tr>
<td></td>
<td>&gt; CBC 1</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Incompatible with existing models.</td>
</tr>
</tbody>
</table>

#### CBC?

<table>
<thead>
<tr>
<th>Response</th>
<th>(&lt;\text{numeric}) = &lt;\text{NR1 NUMERIC RESPONSE DATA}&gt; (FIX3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function</td>
<td>Queries the capture block division number.</td>
</tr>
<tr>
<td>Example</td>
<td>&gt; CBC?</td>
</tr>
<tr>
<td></td>
<td>&lt; CBC 1</td>
</tr>
<tr>
<td></td>
<td>&lt; CBC 128</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Incompatible with existing models.</td>
</tr>
</tbody>
</table>
### BTG <trigger>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;trigger&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Error Detect: When an error is detected</td>
</tr>
<tr>
<td>1</td>
<td>When matches the match pattern</td>
</tr>
<tr>
<td>2</td>
<td>When a manual trigger is generated</td>
</tr>
<tr>
<td>4</td>
<td>When external trigger is selected</td>
</tr>
</tbody>
</table>

**Function**
Sets the trigger to store the data to memory for the capture function.

**Example**
To set the trigger to store the data to memory for the capture function, to Error Detect:

```plaintext
> BTG 0
```

**Compatibility**
Incompatible with existing models.

### BTG? (query)

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;trigger&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt; (FIX1)</th>
</tr>
</thead>
</table>

**Function**
Queries the trigger to store the data to memory for the capture function.

**Example**

```plaintext
> BTG?
< BTG 0
```

**Compatibility**
Incompatible with existing models.

### CPS <pos>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;pos&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Store the data to the top of the memory.</td>
</tr>
<tr>
<td>1</td>
<td>Store the data in the middle of the memory.</td>
</tr>
<tr>
<td>2</td>
<td>Store the data to the last of the memory.</td>
</tr>
</tbody>
</table>

**Function**
Sets the data storage memory position for the capture function.

**Example**
To set the data storage memory position for the capture function to the top of the memory:

```plaintext
> CPS 0
```

**Compatibility**
Incompatible with existing models.

### CPS? (query)

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;pos&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt; (FIX1)</th>
</tr>
</thead>
</table>

**Function**
Queries the data storage memory position for the capture function.

**Example**

```plaintext
> CPS?
< CPS 0
```

**Compatibility**
Incompatible with existing models.
**Chapter 8  Native Commands**

### CMP <numeric>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;numeric&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function</td>
<td>Sets the capture trigger match pattern length.</td>
</tr>
<tr>
<td>Example</td>
<td>To set the match pattern length to 4 bits:</td>
</tr>
<tr>
<td></td>
<td>&gt; CMP 4</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Incompatible with existing models.</td>
</tr>
</tbody>
</table>

**Function**

Sets the capture trigger match pattern length.

**Example**

To set the match pattern length to 4 bits:

```
> CMP 4
```

**Compatibility**

Incompatible with existing models.

### CMP?

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;numeric&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt; (FIX 2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function</td>
<td>Queries the capture trigger match pattern length.</td>
</tr>
<tr>
<td>Example</td>
<td>&gt; CMP?</td>
</tr>
<tr>
<td></td>
<td>&lt; CMP  4</td>
</tr>
<tr>
<td></td>
<td>&lt; CMP  64</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Incompatible with existing models.</td>
</tr>
</tbody>
</table>

**Response**

<numeric> = <NR1 NUMERIC RESPONSE DATA> (FIX 2)

**Function**

Queries the capture trigger match pattern length.

**Example**

```
> CMP?
< CMP  4
< CMP  64
```

**Compatibility**

Incompatible with existing models.

### MPE <data>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;data&gt; = &lt;NON-DECIMAL PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function</td>
<td>Edits the capture trigger match pattern.</td>
</tr>
<tr>
<td>Example</td>
<td>To edit one character of the capture trigger match pattern to &quot;A&quot;:</td>
</tr>
<tr>
<td></td>
<td>&gt; MPE #HA</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Incompatible with existing models.</td>
</tr>
</tbody>
</table>

**Function**

Edits the capture trigger match pattern.

**Example**

To edit one character of the capture trigger match pattern to "A":

```
> MPE #HA
```

**Compatibility**

Incompatible with existing models.

### MPE?

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;data&gt; = &lt;HEXADECIMAL NUMERIC RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function</td>
<td>Queries the capture trigger match pattern.</td>
</tr>
<tr>
<td>Example</td>
<td>&gt; MPE?</td>
</tr>
<tr>
<td></td>
<td>&lt; MPE #H01234...ABCDEF (1 to 16 characters in hexadecimal)</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Incompatible with existing models.</td>
</tr>
</tbody>
</table>

**Response**

<data> = <HEXADECIMAL NUMERIC RESPONSE DATA>

**Function**

Queries the capture trigger match pattern.

**Example**

```
> MPE?
< MPE #H01234...ABCDEF (1 to 16 characters in hexadecimal)
```

**Compatibility**

Incompatible with existing models.
8.4  ED Commands

CMK <data>

Parameter  
<data> = <NON-DECIMAL PROGRAM DATA>
#H**...*  Edits the pattern data in hexadecimal format.
Specify a 1- to 16-character string, consisting of 0 to 9 and A to F.

Function  
Edits the capture trigger mask pattern.

Example  
To edit one character of the capture trigger mask pattern to "A"
> CMK #HA

Compatibility  
Incompatible with existing models.

CMK?

Response  
<data> = <HEXADECIMAL NUMERIC RESPONSE DATA>

Function  
Queries the capture trigger mask pattern.

Example  
> CMK?
< CMK #H01234...ABCDEF (1 to 16 characters in hexadecimal)

Compatibility  
Incompatible with existing models.

CAB?

Response  
<block> = <NR1 NUMERIC RESPONSE DATA> (FIX3)
0 to 128 0 to 128 blocks, in 1-block steps, number of valid capture data blocks

Function  
Queries the number of valid capture data blocks.

Example  
> CAB?
< CAB 128

Compatibility  
Incompatible with existing models.
## CAS <range>[,<start>,<number>]

### Parameter
- **<range>** = <DECIMAL NUMERIC PROGRAM DATA>
  - 0: Acquires all capture data.
  - 1: Acquires capture data of the specified blocks.
- **<start>** = <DECIMAL NUMERIC PROGRAM DATA>
  - 1 to 128: 1 to 128, in 1-block steps, acquisition start block
- **<number>** = <DECIMAL NUMERIC PROGRAM DATA>
  - 1 to 128: 1 to 128, in 1-block steps, number of blocks to be acquired

### Note:
The maximum setting value of <start> and <number> is the maximum value of valid data. <start> and <number> are omitted when <range> = 0.

### Function
Acquires capture data.

### Example
- To acquire all capture data:
  > CAS 0
- To acquire capture data of Blocks 1 to 64:
  > CAS 1, 1, 64

### Compatibility
Incompatible with existing models.

## CAT?

### Response
- **<state>** = <NR1 NUMERIC RESPONSE DATA> (FIX1)
  - 0: Capture data acquisition is not performed.
  - 1: Capture data acquisition is being performed.

### Function
Queries capture data acquisition status.

### Example
- > CAT?
- < CAT 1

### Compatibility
Incompatible with existing models.

## CAC

### Function
Cancels capture data acquisition.

### Example
- > CAC

### Compatibility
Incompatible with existing models.
### CBL?

**Response**

\[
\text{<length>} = \langle NR1 \text{ NUMERIC RESPONSE DATA} \rangle \langle \text{FIX10} \rangle
\]

Maximum capture data/1 to maximum capture data/n bits
\((n = 1, 2, 4, 8, 16, 32, 64, 128)\)

Capture data length per block

**Note:**

- The maximum capture data length is doubled at 2 ch Combination.
- The maximum capture data length is quadrupled at 4 ch Combination.

**Function**

Queries capture data length per block.

**Example**

To query capture data length when the number of blocks is 32:

\[
> \text{CBL}?
\]

\[
< \text{CBL} \quad 4194304
\]

**Compatibility**

Incompatible with existing models.

### CDW? <block>,<start>

**Parameter**

\[
\langle \text{block} \rangle = \langle \text{DECIMAL NUMERIC PROGRAM DATA} \rangle
\]

1 to 128 1 to 128 blocks, in 1-block steps

\[
\langle \text{start} \rangle = \langle \text{NON-DECIMAL PROGRAM DATA} \rangle
\]

#H0 to #H7FFFFFF 0 to 7FFFFFF bits, in 1-bit steps

(Specify in hexadecimal.)

**Note:**

- The maximum setting is the pattern length.
- At 2 ch Combination, the setting range and Step are doubled.
- At 4 ch Combination, the setting range and Step are quadrupled.

**Response**

\[
\langle \text{data} \rangle = \langle \text{HEXADECIMAL NUMERIC RESPONSE DATA} \rangle
\]

#H***...*

Returns pattern data in hexadecimal format. 400 characters max. (pattern data of 400 \(\times\) 4 bits)

**Function**

Queries data of 400 \(\times\) 4 bits from \(<\text{start}\>\) address in the specified block acquired by capture function. In the acquired data, the error bit positions in ED reference data are reversed.

**Example**

To query data of 400 \(\times\) 4 bits from address 0 in block 3 acquired by capture function:

\[
> \text{CDW} \ 3, \#H0
\]

\[
< \text{CDW} \ #H001
\]

**Compatibility**

Incompatible with existing models.
### CBW? \(<\text{block}>\),<start>\[,<\text{size}>\]

#### Parameter

- \(<\text{block}>\) = <DECIMAL NUMERIC PROGRAM DATA>
  - 1 to 128 blocks, in 1-block steps
- \(<\text{start}>\) = <NON-DECIMAL PROGRAM DATA>
  - \(#H0\) to \(#H7FFFFFF\) 0 to 7FFFFFFF bits, in 1-bit steps
  - (Specify in hexadecimal.)
- \(<\text{size}>\) = <NR1 NUMERIC PROGRAM DATA>
  - 1 to 16777216 1 to 16,777,216 bytes, in 1-byte steps

**Note:**

The maximum setting is the pattern length.
- At 2 ch Combination, the setting range and Step are doubled.
- At 4 ch Combination, the setting range and Step are quadrupled.

#### Response

\(<\text{bdata}>\) = <DEFINITE LENGTH ARBITRARY BLOCK RESPONSE DATA>

\(#XYYY<\text{binary}>\)

- \(X:\) Number of \(YYY\) digits
- \(YYY:\) Number of \(<\text{binary}>\) bytes: 1 to 16,777,216 (bytes)

\(<\text{binary}>\): Binary data up to 16,777,216 bytes

#### Function

Queries binary data of 16,777,216 bytes from \(<\text{start}>\) address in the specified block acquired by capture function. In the acquired data, the error bit positions in ED reference data are reversed.

#### Example

To query binary data of 16,777,216 bytes from address 0 in block 3 acquired by capture function:

\$ > CBW? 3,#H0
\$< CBW #10011

#### Compatibility

Incompatible with existing models.
CED? <block>,<start>

Parameter

- `<block> = <DECIMAL NUMERIC PROGRAM DATA>`
- 1 to 128 blocks, in 1-block steps
- `<start> = <NON-DECIMAL PROGRAM DATA>`
- #H0 to #H7FFFFFF 0 to 7FFFFFF bits, in 1-bit steps
- (Specify in hexadecimal.)

**Note:**

- The maximum setting is the pattern length.
- At 2 ch Combination, the setting range and Step are doubled.
- At 4 ch Combination, the setting range and Step are quadrupled.

Response

- `<data> = <HEXADECIMAL NUMERIC RESPONSE DATA>`
- #H***…* Returns pattern data in hexadecimal format.
- 400 characters max. (pattern data of 400 × 4 bits)

Function

Queries error position of 400 × 4 bits from `<start>` address in the specified block acquired by capture function.

Example

To query error position of 400 × 4 bits from address 0 in block 3 acquired by capture function:

```
> CED? 3,#H0
< CED #H001
```

Compatibility

Incompatible with existing models.
Chapter 8  Native Commands

CEB? <block>,<start>[,<size>]

Parameter

<block> = <DECIMAL NUMERIC PROGRAM DATA>
1 to 128  1 to 128 blocks, in 1-block steps

<start> = <NON-DECIMAL PROGRAM DATA>
#H0 to #H7FFFFFFF  0 to 7FFFFFFF bits, in 1-bit steps
(Specify in hexadecimal.)

[size] = <NR1 NUMERIC PROGRAM DATA>
1 to 16777216  1 to 16,777,216 bytes, in 1-byte steps

Note:
The maximum setting is the pattern length.
At 2 ch Combination, the setting range and Step are doubled.
At 4 ch Combination, the setting range and Step are quadrupled.

Response

<bdata> = <DEFINITE LENGTH ARBITRARY BLOCK RESPONSE DATA>
#XYYY<binary>
X: Number of YYY digits
YYY: Number of <binary> bytes: 1 to 16,777,216 (bytes)

<binary>: Binary data up to 16,777,216 bytes

Function
 Queries error position of 16,777,216 bytes from <start> address in the
 specified block acquired by capture function.

Example
To query error position of 16,777,216 bytes from address 0 in block 3
acquired by capture function:
> CEB? 3,#H0
< CEB #10011

Compatibility
Incompatible with existing models.
8.4.6 Commands related to Misc tab

Figure 8.4.6-1 Misc tab
8.4.6.1 Pattern Sequence setting commands

![Diagram of Pattern Sequence setting]

Table 8.4.6.1-1 Pattern Sequence setting commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1]</td>
<td>Sequence</td>
<td>BST(PTI)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BST?(PTI?)</td>
</tr>
<tr>
<td>[2]</td>
<td>Source</td>
<td>BSM</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BSM?</td>
</tr>
<tr>
<td>[3]</td>
<td>Burst Trigger Delay</td>
<td>ABD</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ABD?</td>
</tr>
<tr>
<td>[4]</td>
<td>No label (Automatic Adjustment)</td>
<td>BAD</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BAD?</td>
</tr>
<tr>
<td>[5]</td>
<td>Enable Period</td>
<td>BEL</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BEL?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BCY?</td>
</tr>
</tbody>
</table>
8.4 ED Commands

### BST (PTI) <mode>

**Parameter**

- `<mode>` = <DECIMAL NUMERIC PROGRAM DATA>
- 0  Repeat: Repeat signal
- 1  Burst: Burst signal

**Function**
Sets the signal reception mode to Repeat (consecutive) or Burst.

**Example**
To set the signal reception mode to Repeat:

> BST(PTI) 0

**Compatibility**
Compatible with the MP1632C (PTI).

### BST? (PTI?)

**Response**
- `<mode>` = <NR1 NUMERIC RESPONSE DATA> (FIX1)

**Function**
Queries the signal reception mode.

**Example**

> BST(PTI)?

> BST(PTI) 0

**Compatibility**
Compatible with the MP1632C (PTI?).

### BSM <mode>

**Parameter**

- `<mode>` = <DECIMAL NUMERIC PROGRAM DATA>
- 0  Internal
- 1  External Trigger
- 2  External Enable

**Function**
Sets the Burst signal reception sequence.

**Example**
To set the Burst signal reception sequence to Internal:

> BSM 0

**Compatibility**
Incompatible with existing models.

### BSM?

**Response**
- `<mode>` = <NR1 NUMERIC RESPONSE DATA> (FIX1)

**Function**
Queries the Burst signal reception sequence.

**Example**

> BSM?

< BSM 0

**Compatibility**
Incompatible with existing models.
**Chapter 8  Native Commands**

### ABD <numeric>

**Parameter**

<numeric> = <DECIMAL NUMERIC PROGRAM DATA>

0 to 2147483648  0 to 2,147,483,648 bits, in 16-bit steps

The maximum value for this parameter is the value set for Burst Cycle. If Burst Cycle is unavailable, the maximum value is 2,147,483,584 bits.

**Note:**

At 2 ch Combination, the setting range and Step are doubled.

At 4 ch Combination, the setting range and Step are quadrupled.

**Function**

Sets the delay value for the Burst signal when Burst External Trigger is selected.

**Example**

To set the delay value for the Burst signal when Burst External Trigger is selected to 0 bit:

> ABD 0

**Compatibility**

Incompatible with existing models.

### ABD?

**Response**

<numeric> = <NR1 NUMERIC RESPONSE DATA> (FIX11)

**Function**

Queries the delay value for the Burst signal when Burst External Trigger is selected.

**Example**

> ABD?

< ABD           0

< ABD  2147483648

**Compatibility**

Incompatible with existing models.

### BAD <adjust>

**Parameter**

<adjust> = <DECIMAL NUMERIC PROGRAM DATA>

0 Adjusts automatically

1 Does not adjust automatically

**Function**

Sets whether to execute automatic adjustment of Burst Trigger Delay.

**Example**

> BAD 0

**Compatibility**

Incompatible with existing models.

### BAD?

**Response**

<adjust> = <NR1 NUMERIC RESPONSE DATA> (FIX1)

**Function**

Queries the Burst Trigger Delay adjustment method.

**Example**

> BAD?

< BAD 0

**Compatibility**

Incompatible with existing models.
**BEL <numeric>**

Parameter: \(<\text{numeric}> = \langle\text{DECIMAL NUMERIC PROGRAM DATA}\rangle\)
- 12800 to 2147483520 12,800 to 2,147,483,520 bits, in 128-bit steps

*Note:*
- At 2 ch Combination, the setting range and Step are doubled.
- At 4 ch Combination, the setting range and Step are quadrupled.

Function: Sets the data signal reception interval for Burst signal reception.

Example: To set the Burst signal reception interval for Burst signal reception to 12,800 bits:
- `> BEL 12800`

Compatibility: Incompatible with existing models.

**BEL?**

Response: \(<\text{numeric}> = \langle\text{NR1 NUMERIC RESPONSE DATA}\rangle\) (FIX10)

Function: Queries the data signal reception interval for Burst signal reception.

Example:
- `> BEL?`
- `< BEL 12800`
- `< BEL 134217728`

Compatibility: Incompatible with existing models.

**BCY <numeric>**

Parameter: \(<\text{numeric}> = \langle\text{DECIMAL NUMERIC PROGRAM DATA}\rangle\)
- 25600 to 2147483648 25,600 to 2,147,483,648 bits, in 128-bit steps

*Note:*
- At 2 ch Combination, the setting range and Step are doubled.
- At 4 ch Combination, the setting range and Step are quadrupled.

Function: Sets the Burst signal generation cycle.

Example: To set the Burst signal generation cycle to 25,600 bits:
- `> BCY 25600`

Compatibility: Incompatible with existing models.

**BCY?**

Response: \(<\text{numeric}> = \langle\text{NR1 NUMERIC RESPONSE DATA}\rangle\) (FIX11)

Function: Queries the Burst signal generation cycle.

Example:
- `> BCY?`
- `< BCY 25600`
- `< BCY 2147483648`

Compatibility: Incompatible with existing models.
8.4.6.2 AUX Output setting commands

[Image of AUX Output setting]

- AUX Output
- Pattern Sync
- Position
- 1
- 1

Figure 8.4.6.2-1 Pattern Sync setting
(For PRBS, Zero Substitution, and Data patterns)

[Image of AUX Output setting]

- AUX Output
- Pattern Sync
- Position
- Block No.
- 1
- Row No.
- 1

Figure 8.4.6.2-2 Pattern Sync setting (For Mixed pattern)

[Image of AUX Output setting]

- AUX Output
- Pattern Sync
- Position
- Block No.
- 1
- Row No.
- 1
- bits

Figure 8.4.6.2-3 Pattern Sync setting (For Sequence pattern)
### 8.4 ED Commands

#### Table 8.4.6.2-1  AUX Output setting commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1]</td>
<td>AUX Output</td>
<td>SOP</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOP?</td>
</tr>
<tr>
<td>[2]</td>
<td>Position (PRBS, Zero Substitution, Data)</td>
<td>PSP</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PSP?</td>
</tr>
<tr>
<td>[3]</td>
<td>Block No. (Mixed Data)</td>
<td>SMB</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SMB?</td>
</tr>
<tr>
<td>[4]</td>
<td>Row No. (Mixed Data)</td>
<td>SMR</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SMR?</td>
</tr>
<tr>
<td>[5]</td>
<td>Block No. (Sequence)</td>
<td>SSB</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SSB?</td>
</tr>
<tr>
<td>[6]</td>
<td>Position (Sequence)</td>
<td>SSP</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SSP?</td>
</tr>
</tbody>
</table>

**SOP <source>[,<numeric>]**

**Parameter**
- `<source>` = `<DECIMAL NUMERIC PROGRAM DATA>`
  - 0 1/32 clock
  - 1 Pattern Sync (Variable)
  - 2 Pattern Sync (Variable)
  - 4 1/8 clock (When MU181040A-002 and MU181040B-002 are installed)
  - 6 1/16 clock
  - 9 1/64 clock
  - 10 Sync Gain
  - 11 Error Output
  - 13 1/n clock
- `<numeric>` = `<DECIMAL NUMERIC PROGRAM DATA>`
  Omits `<numeric>` when NCLock (1/n clocks) is not selected.
  - 16, 32, 64 16, 32, 64 (for MU181040A-001)
  - 8 to 511 8 to 511, in single step (for MU181040A-002, MU181040B-002)

**Function**
Sets the output signal for synchronization output.

**Example**
To set the output signal for synchronization output to 1/32 clock:

```
> SOP 0
```

To set the output signal for synchronization output to 1/511 clock:

```
> SOP 13,511
```

**Compatibility**
- Partially compatible with the MP1632C (Compatible with 1 only).
- Partially compatible with the MP1762D (Compatible with 0 and 2 only).
Chapter 8  Native Commands

**SOP?**

### Response

- `<source>` = `<NR1 NUMERIC RESPONSE DATA>` (FIX2)
- `<numeric>` = `<NR1 NUMERIC RESPONSE DATA>` (FIX3)

  8 to 511

  Omit `<numeric>` when 1/n clock is not selected.

### Function

Queries the output signal for synchronization output.

### Example

```plaintext
> SOP?
< SOP 0
< SOP 13, 8
< SOP 13,511
```

### Compatibility

Partially compatible with the MP1632C (Compatible with 1 only).

Partially compatible with the MP1762D (Compatible with 0 and 2 only).

---

**PSP <numeric>**

### Parameter

- `<numeric>` = `<DECIMAL NUMERIC PROGRAM DATA>`

  1 to 68719476657  1 to 68,719,476,657 bits, in 16-bit steps

  In the case of combination,

  1 to 274877906625  1 to 274,877,906,625 bits, in 64-bit steps

### Function

Sets the output position for synchronization output for Data, PRBS, and Zero Substitution patterns.

### Example

To set the output position for synchronization output for Data, PRBS, and Zero Substitution patterns to 1 bit:

```plaintext
> PSP 1
```

### Compatibility

Partially compatible with the MP1632C and MP1762D (16-bit resolution (steps) compatible).

---

**PSP?**

### Response

- `<numeric>` = `<NR1 NUMERIC RESPONSE DATA>` (FIX12)

### Function

Queries the output position for synchronization output for Data, PRBS, and Zero Substitution patterns.

### Example

```plaintext
> PSP?
< PSP 1
< PSP 2147483633
```

### Compatibility

Partially compatible with the MP1632C and MP1762D (16-bit resolution (steps) compatible).
8.4 ED Commands

SMB <numeric>

Parameter

<numeric> = <DECIMAL NUMERIC PROGRAM DATA>
1 to 511 1 to 511 blocks, in 1-block steps
The maximum setting value is the value set in Block No.

Function

Sets the output block position for synchronization output for Mixed Data pattern.

Example

To set the output block position for synchronization output for Mixed Data pattern to block 1:
> SMB 1

Compatibility

Incompatible with existing models.

SMB?

Response

<numeric> = <NR1 NUMERIC RESPONSE DATA> (FIX3)

Function

Queries the output block position for synchronization output for Mixed Data pattern.

Example

> SMB?
< SMB 1
< SMB 511

Compatibility

Incompatible with existing models.

SMR <numeric>

Parameter

<numeric> = <DECIMAL NUMERIC PROGRAM DATA>
1 to 16 1 to 16 rows, in 1-row steps
The maximum setting value is the value set in Row No.

Function

Sets the output row position for synchronization output for Mixed Data pattern.

Example

To set output row position for synchronization output for Mixed Data pattern to row 1:
> SMR 1

Compatibility

Incompatible with existing models.

SMR?

Response

<numeric> = <NR1 NUMERIC RESPONSE DATA> (FIX2)

Function

Queries the output row position for synchronization output for Mixed Data pattern.

Example

> SMR?
< SMR 1
< SMR 16

Compatibility

Incompatible with existing models.
SSB <numeric>

Parameter
<numeric> = <DECIMAL NUMERIC PROGRAM DATA>
1 to 128 1 to 128 blocks, in 1-block steps
The maximum setting value is the value set in Block No.

Function
Sets the output block position for synchronization output for Sequence pattern.

Example
To set the output block position for synchronization output for Sequence pattern to block 128:
> SSB 128

Compatibility
Incompatible with existing models.

SSB?

Response
<numeric> = <NR1 NUMERIC RESPONSE DATA> (FIX3)

Function
Queries the output block position for synchronization output for Sequence pattern.

Example
> SSB?
< SSB 1
< SSB 128

Compatibility
Incompatible with existing models.

SSP <numeric>

Parameter
<numeric> = <DECIMAL NUMERIC PROGRAM DATA>
1 to 1048497 1 to 1,048,497 bits, in 16-bit steps

Function
Sets the output position for synchronization output for Sequence pattern.

Example
To set the synchronization output position for Sequence pattern to bit 1:
> SSP 1

Compatibility
Incompatible with existing models.

SSP?

Response
<numeric> = <NR1 NUMERIC RESPONSE DATA> (FIX7)

Function
Queries the output position for synchronization output for Sequence pattern.

Example
> SSP?
< SSP 1
< SSP 1048497

Compatibility
Incompatible with existing models.
8.4.6.3 AUX Input setting commands

Figure 8.4.6.3-1  AUX Input setting

Table 8.4.6.3-1  AUX Input setting commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1]</td>
<td>AUX Input</td>
<td>AUS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AUS?</td>
</tr>
</tbody>
</table>

AUS <usage>

Parameter  <usage> = <DECIMAL NUMERIC PROGRAM DATA>
0    Burst
1    External Mask
2    Capture External Trigger

Function  Sets the usage of the common connector input.
Example   To set the usage of the common connector input to Burst:
          > AUS 0
Compatibility  Incompatible with existing models.

AUS?

Response  <usage> = <NR1 NUMERIC RESPONSE DATA> (FIX1)
Function   Queries the usage of the common connector input.
Example    > AUS?
           < AUS 0
Compatibility  Incompatible with existing models.
8.4.6.4 Measurement Restart setting commands

Table 8.4.6.4-1 Measurement Restart setting commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1]</td>
<td>Measurement Restart</td>
<td>MRS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MRS?</td>
</tr>
</tbody>
</table>

MRS <data>,<clock>

Parameter

<data> = <DECIMAL NUMERIC PROGRAM DATA>
0  Does not restart measurement when the data threshold is changed (Measurement restart at data threshold change OFF).
1  Restarts measurement when the data threshold is changed (Measurement restart at data threshold change ON).

<clock> = <DECIMAL NUMERIC PROGRAM DATA>
0  Does not restart measurement when the clock delay is changed (Measurement restart at clock delay change OFF).
1  Restarts measurement when the clock delay is changed (Measurement restart at clock delay change ON).

Function
Sets the measurement restart condition.

Example
To set the measurement restart condition to "Measurement restart at data threshold change ON and clock delay change ON":
> MRS 1,1

Compatibility
Incompatible with existing models.
8.4 ED Commands

### MRS?

**Response**

<table>
<thead>
<tr>
<th>Response</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;data&gt;,&lt;clock&gt;</td>
<td>&lt;NR1 NUMERIC RESPONSE DATA&gt; (FIX1)</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>Does not restart measurement when the data threshold is changed (Measurement restart at data threshold change OFF).</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Restarts measurement when the data threshold is changed (Measurement restart at data threshold change ON).</td>
<td></td>
</tr>
<tr>
<td>&lt;clock&gt;</td>
<td>&lt;NR1 NUMERIC PROGRAM DATA&gt; (FIX1)</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>Does not restart measurement when the clock delay is changed (Measurement restart at clock delay change OFF).</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Restarts measurement when the clock delay is changed (Measurement restart at clock delay change ON).</td>
<td></td>
</tr>
</tbody>
</table>

**Function**

Queries the measurement restart condition.

**Example**

> MRS?

< MRS 1,1

**Compatibility**

Incompatible with existing models.
8.5 Automatic Measurement Commands

This section describes the commands for the measurement application. Refer to Section 7.1 "Common Commands" and 9.1 "Basic Operation of Device Messages" for how to specify a slot number with the :SYSTem:CFUNction command.

8.5.1 Eye Margin measurement setting commands

Figure 8.5.1-1  Eye Margin measurement screen
### 8.5 Automatic Measurement Commands

#### Table 8.5.1-1  Eye Margin measurement setting commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1]</td>
<td>Set All</td>
<td>MEA</td>
</tr>
<tr>
<td>[2]</td>
<td>Reset All</td>
<td>MER</td>
</tr>
<tr>
<td>[3]</td>
<td>Slot</td>
<td>MES</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MES?</td>
</tr>
<tr>
<td>[4]</td>
<td>Start</td>
<td>EST</td>
</tr>
<tr>
<td></td>
<td></td>
<td>EST?</td>
</tr>
<tr>
<td>[5]</td>
<td>Error Threshold</td>
<td>EYT</td>
</tr>
<tr>
<td></td>
<td></td>
<td>EYT?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MMR?</td>
</tr>
<tr>
<td>[7]</td>
<td>Auto Search ON/OFF</td>
<td>EMA</td>
</tr>
<tr>
<td></td>
<td></td>
<td>EMA?</td>
</tr>
<tr>
<td>[8]</td>
<td>Selection of measurement time display</td>
<td>MTI</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MTI?</td>
</tr>
<tr>
<td>[9]</td>
<td>Setting of time display</td>
<td>MDT?</td>
</tr>
<tr>
<td>[10]</td>
<td>Query for measurement start time</td>
<td>ESA?</td>
</tr>
<tr>
<td>[11]</td>
<td>Query for measurement elapsed time</td>
<td>MLP?</td>
</tr>
<tr>
<td>[12]</td>
<td>Query for measurement result</td>
<td>EMM?</td>
</tr>
<tr>
<td>[13]</td>
<td>Query for measurement status</td>
<td>EMW?</td>
</tr>
</tbody>
</table>

#### MEA

- **Function**: Selects all slots to ON.
- **Example**: `> MEA`
- **Compatibility**: Incompatible with existing models.

#### MER

- **Function**: Selects all slots to OFF.
- **Example**: `> MER`
- **Compatibility**: Incompatible with existing models.
Chapter 8  Native Commands

**MES <slot>,<numeric>[,<unit>]**

**Parameter**
- `<slot>` = `<DECIMAL NUMERIC PROGRAM DATA>`
  - 1 to 6  Slot No. 1 to 6
  - 1 to 6 when using the MP1800A, 1 to 4 when using the MT1810A
- `<numeric>` = `<DECIMAL NUMERIC PROGRAM>`
  - 1  ON
  - 0  OFF
- `[<unit>]` = `<DECIMAL NUMERIC PROGRAM DATA>`
  - 1 to 4  Mainframe No. 1 to 4
  - Specify the mainframe No. when connected to the MT1810A.
  - `<unit>` is optional. When omitted, 1 is specified.

**Function**
Selects the Eye Margin measurement slot.

**Example**
Eye Margin measurement  To set Slot 1 to ON:
> MES 1,1

**Compatibility**
Incompatible with existing models.

---

**MES? <slot>[,<unit>]**

**Parameter**
- `<slot>` = `<DECIMAL NUMERIC PROGRAM DATA>`
  - 1 to 6  Slot No. 1 to 6
  - 1 to 6 when using the MP1800A, 1 to 4 when using the MT1810A
- `[<unit>]` = `<DECIMAL NUMERIC PROGRAM DATA>`
  - 1 to 4  Mainframe No. 1 to 4
  - Specify the mainframe No. when connected to the MT1810A.
  - `<unit>` is optional. When omitted, 1 is specified.

**Response**
- `<numeric>` = `<NR1 NUMERIC RESPONSE DATA> (FIX1)`

**Function**
Eye Margin measurement Queries the slot.

**Example**
> MES? 1
< MES 1

**Compatibility**
Incompatible with existing models.

---

**EST <type>**

**Parameter**
- `<type>` = `<NR1 NUMERIC PROGRAM DATA>`
  - 1  Starts measurement
  - 0  Stops measurement

**Function**
Starts and stops Eye Margin measurement.

**Example**
To start Eye Margin measurement:
> EST 1

**Compatibility**
Compatible with the MP1632C.
### 8.5 Automatic Measurement Commands

#### EST?

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>numeric</td>
<td>The numeric response value.</td>
</tr>
</tbody>
</table>

- **Response**
  - `<numeric>` = `<NR1 NUMERIC RESPONSE DATA>` (FIX1)
  - 1: Starts measurement
  - 0: Stops measurement
  - 2: Measurement failure

- **Function**
  - Queries the Eye Margin measurement status.

- **Example**
  - `> EST?`
  - `< EST 2`

- **Compatibility**
  - Compatible with the MP1632C.

#### EYT `<thre>`

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;thre&gt;</code></td>
<td>The decimal numerical program data.</td>
</tr>
</tbody>
</table>

- **Parameter**
  - `<thre>` = `<DECIMAL NUMERIC PROGRAM DATA>`
  - 1: 1E–3
  - 2: 1E–4
  - 3: 1E–5
  - 4: 1E–6
  - 5: 1E–7
  - 6: 1E–8
  - 7: 1E–9
  - 8: 1E–10
  - 9: 1E–11
  - 10: 1E–12

- **Function**
  - Sets the threshold value that is to be the boundary of the Eye Margin measurement margin.

- **Example**
  - To set the threshold value that is to be the Eye Margin measurement margin to 1E–3:
    - `> EYT 1`

- **Compatibility**
  - Incompatible with existing models.

#### EYT?

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;thre&gt;</code></td>
<td>The decimal numerical program data.</td>
</tr>
</tbody>
</table>

- **Response**
  - `<thre>` = `<NR1 NUMERIC RESPONSE DATA>` (FIX2)

- **Function**
  - Queries the threshold value that is to be the boundary of the Eye Margin measurement margin.

- **Example**
  - `> EYT?`
  - `< EYT 1`
  - `< EYT 10`

- **Compatibility**
  - Incompatible with existing models.
Chapter 8  Native Commands

MMR <type>
Parameter  
<type> = <DECIMAL NUMERIC PROGRAM DATA>
0  COARse
1  FINE

Function  
Sets the accuracy of the error ratio for the Eye Margin measurement.

Example  
To set the accuracy of the error ratio for the Eye Margin measurement to FINE:
> MMR 1

Compatibility  
Compatible with the MP1632C.

MMR?
Response  
<thre> = <NR1 NUMERIC RESPONSE DATA> (FIX1)

Function  
Queries the accuracy of the error ratio for the Eye Margin measurement.

Example  
> MMR?
< MMR 1

Compatibility  
Compatible with the MP1632C.

EMA <numeric>
Parameter  
<numERIC> = <DECIMAL NUMERIC PROGRAM DATA>
2  Auto Search ON (Fine)
1  Auto Search ON (Coarse)
0  Auto Search OFF

Function  
Selects between Auto Search ON/OFF upon start of Eye Margin measurement.

Example  
To set Auto Search upon start of Eye Margin measurement to ON:
> EMA 1

Compatibility  
Incompatible with existing models.

EMA?
Response  
<numERIC> = <NR1 NUMERIC RESPONSE DATA> (FIX1)

Function  
Queries the Auto Search ON/OFF state upon start of Eye Margin measurement.

Example  
> EMA?
< EMA 1

Compatibility  
Incompatible with existing models.
8.5 Automatic Measurement Commands

**MTI <type>**

**Parameter**

- `<type>` = <DECIMAL NUMERIC PROGRAM DATA>
  - 0 Displays the current date and time.
  - 1 Displays the measurement start time.
  - 2 Displays the elapsed time based on the measurement period.

**Function**

Selects the Eye Margin measurement time display type.

**Example**

To set the Eye Margin measurement time display type to measurement start time (Start Time):

> MTI 1

**Compatibility**

Compatible with the MP1632C.

**MTI?**

**Response**

- `<type>` = <NR1 NUMERIC RESPONSE DATA> (FIX1)

**Function**

Queries the Eye Margin measurement time display type.

**Example**

> MTI?

< MTI 1

**Compatibility**

Compatible with the MP1632C.

**MDT?**

**Response**

- `<year>` = <NR1 NUMERIC RESPONSE DATA> (FIX4)
  - 2000 to 2036 Year 2000 to 2036
- `<month>` = <NR1 NUMERIC RESPONSE DATA> (FIX2)
  - 01 to 12 January to December
- `<day>` = <NR1 NUMERIC RESPONSE DATA> (FIX2)
  - 01 to 31 1st to 31st
- `<hour>` = <NR1 NUMERIC RESPONSE DATA> (FIX2)
  - 00 to 23 0 to 23 hours
- `<minute>` = <NR1 NUMERIC RESPONSE DATA> (FIX2)
  - 00 to 59 0 to 59 minutes
- `<second>` = <NR1 NUMERIC RESPONSE DATA> (FIX2)
  - 00 to 59 0 to 59 seconds

**Function**

Queries the current date and time during Eye Margin measurement.

**Example**

> MDT?

< MDT 2006,04,01,23,59,59

**Compatibility**

Incompatible with existing models.
Chapter 8 Native Commands

ESA?

Response

<year> = <NR1 NUMERIC RESPONSE DATA> (FIX4)
0000, 2000 to 2036 (year) Year 2000 to 2036

<month> = <NR1 NUMERIC RESPONSE DATA> (FIX2)
00, 1 to 12 (month) January to December

<day> = <NR1 NUMERIC RESPONSE DATA> (FIX2)
00, 1 to 31 (day) 1st to 31st

<hour> = <NR1 NUMERIC RESPONSE DATA> (FIX2)
00 to 23 (hour) 0 to 23 hours

<minute> = <NR1 NUMERIC RESPONSE DATA> (FIX2)
00 to 59 (minute) 0 to 59 minutes

<second> = <NR1 NUMERIC RESPONSE DATA> (FIX2)
00 to 59 (second) 0 to 59 seconds

Note:
The following is output when no measurement is performed:
ESA 0000,00,00,00,00,00

Function
Queries the Eye Margin measurement start time (Start Time).

Example
> ESA?
< ESA 2006,04,01,23,59,59

Compatibility
Compatible with the MP1632C.

MLP?

Response

<day> = <NR1 NUMERIC RESPONSE DATA> (FIX2)
00 to 99 0 to 99 days

<hour> = <NR1 NUMERIC RESPONSE DATA> (FIX2)
00 to 23 0 to 23 hours

<minute> = <NR1 NUMERIC RESPONSE DATA> (FIX2)
00 to 59 0 to 59 minutes

<second> = <NR1 NUMERIC RESPONSE DATA> (FIX2)
00 to 59 0 to 59 seconds

Note:
The following is output when no measurement is performed:
MLP 00,00,00,00

Function
Queries the Eye Margin measurement elapsed time (Elapsed Time).

Example
> MLP?
< MLP 99,23,59,59

Compatibility
Compatible with the MP1632C.
8.5 Automatic Measurement Commands

EMM? <slot>,<numeric>[,<unit>]

Parameter
- <slot> = <DECIMAL PROGRAM DATA>
  1 to 6       Slot No. 1 to 6
- <numeric> = <DECIMAL PROGRAM DATA>
  "<result>"   Measurement data
  For the contents of <result> see Table 8.5.1-3.
- [<unit>] = <DECIMAL NUMERIC PROGRAM DATA>
  1 to 4       Mainframe No. 1 to 4
  Specify the mainframe No. when connected to the MT1810A.
  <unit> is optional. When omitted, 1 is specified.

Response
- <string> = <STRING RESPONSE DATA> (FIX7)

Table 8.5.1-2 Response format

<table>
<thead>
<tr>
<th>Form</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Form1</td>
<td>XXXXXXXX</td>
<td>When 0 to MAX (μs/ps)</td>
</tr>
<tr>
<td></td>
<td>Failed</td>
<td>For measurement failure</td>
</tr>
<tr>
<td></td>
<td>......</td>
<td>When no data corresponds to the query</td>
</tr>
<tr>
<td>Form2</td>
<td>XXXXX</td>
<td>For 0 to MAX (mVpp)</td>
</tr>
<tr>
<td></td>
<td>Failed</td>
<td>For measurement failure</td>
</tr>
<tr>
<td></td>
<td>......</td>
<td>When no data corresponds to the query</td>
</tr>
</tbody>
</table>

Function
Queries the Eye Margin measurement result.

Table 8.5.1-3 Parameters

<table>
<thead>
<tr>
<th>Items</th>
<th>&lt;result&gt;</th>
<th>Response Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>Phase margin</td>
<td>0</td>
<td>Form1</td>
</tr>
<tr>
<td>Threshold margin</td>
<td>1</td>
<td>Form2</td>
</tr>
<tr>
<td>Period</td>
<td>2</td>
<td>Form1</td>
</tr>
</tbody>
</table>

Example
Queries the phase margin result of the Eye Margin measurement for Slot1.
> EMM? 1,0
< EMM 1000

Compatibility
Incompatible with existing models.
### EMW?

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;numeric&gt; = &lt;DECIMAL NUMERIC RESPONSE DATA&gt; (FIX1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>When no alarm exists.</td>
</tr>
<tr>
<td>1</td>
<td>Sync. Loss</td>
</tr>
<tr>
<td>2</td>
<td>Clock Loss</td>
</tr>
<tr>
<td>3</td>
<td>CR Unlock</td>
</tr>
<tr>
<td>4</td>
<td>Out of range</td>
</tr>
<tr>
<td>5</td>
<td>Frequency NG</td>
</tr>
<tr>
<td>6</td>
<td>Illegal Error</td>
</tr>
</tbody>
</table>

**Note:**
- Displays any one of the alarms.
- When two or more alarms have occurred, they are displayed in the following order:
  - Illegal Error > Frequency NG > Clock Loss (CR Unlock) > Sync Loss > Out of range

**Function**: Queries the measurement status during Eye Margin measurement.

**Example**

> EMW?
< EMW 1

**Compatibility**: Incompatible with existing models.
8.5.2 Eye Diagram measurement commands

This section describes the commands for Eye Diagram measurement. The commands related to each tab are described in separate sections below.

Figure 8.5.2-1  Eye Diagram measurement screen
8.5.2.1 Condition tab setting commands

![Figure 8.5.2.1-1 Condition tab](image)

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1]</td>
<td>Tab selection</td>
<td>ETB</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ETB?</td>
</tr>
<tr>
<td>[2]</td>
<td>Slot Eye Diagram</td>
<td>EDS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>EDS?</td>
</tr>
<tr>
<td>[3]</td>
<td>Slot Mask Test</td>
<td>EMS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>EMS?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>EMR?</td>
</tr>
<tr>
<td>[5]</td>
<td>Transition Bit Measurement</td>
<td>EMT</td>
</tr>
<tr>
<td></td>
<td></td>
<td>EMT?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>EYP?</td>
</tr>
<tr>
<td>[7]</td>
<td>Auto Search</td>
<td>EDA</td>
</tr>
<tr>
<td></td>
<td></td>
<td>EDA?</td>
</tr>
</tbody>
</table>
### ETB <tab>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;tab&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>Condition tab</td>
</tr>
<tr>
<td>1</td>
<td>Diagram tab</td>
</tr>
<tr>
<td>2</td>
<td>Mask Edit tab</td>
</tr>
</tbody>
</table>

**Function**: Switches the tab display in the Eye Diagram measurement screen.

**Example**: To display the Condition tab in the Eye Diagram measurement screen:
```
> ETB 0
```

**Compatibility**: Incompatible with existing models.

### ETB?

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;tab&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt; (FIX1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 to 2</td>
<td></td>
</tr>
</tbody>
</table>

**Function**: Queries the tab currently active in the Eye Diagram measurement screen.

**Example**: To query the active tab:
```
> ETB?
< ETB 0
```

**Compatibility**: Incompatible with existing models.

### EDS <slot>,<numeric> [,<unit>]

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;slot&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
<td></td>
</tr>
<tr>
<td>1 to 6</td>
<td>Slot No. 1 to 6</td>
</tr>
<tr>
<td>&lt;numeric&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>ON</td>
</tr>
<tr>
<td>0</td>
<td>OFF</td>
</tr>
<tr>
<td>[&lt;unit&gt;] = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
<td></td>
</tr>
<tr>
<td>1 to 4</td>
<td>Mainframe No. 1 to 4</td>
</tr>
</tbody>
</table>

Specify the mainframe No. when connected to the MT1810A. `<unit>` is optional. When omitted, 1 is specified.

**Function**: Selects the Eye Diagram measurement slot.

**Example**: To set Slot1 to ON:
```
> EDS 1,1
```

**Compatibility**: Incompatible with existing models.
Chapter 8   Native Commands

**EDS? <slot>[,<unit>]**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;slot&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
<td>Slot No. 1 to 6</td>
</tr>
<tr>
<td>1 to 6</td>
<td></td>
</tr>
<tr>
<td>1 to 6 when using the MP1800A, 1 to 4 when using the MT1810A.</td>
<td></td>
</tr>
<tr>
<td>[&lt;unit&gt;] = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
<td>Mainframe No. 1 to 4</td>
</tr>
<tr>
<td>1 to 4</td>
<td></td>
</tr>
<tr>
<td>Specify the mainframe No. when connected to the MT1810A.</td>
<td></td>
</tr>
<tr>
<td>&lt;unit&gt; is optional. When omitted, 1 is specified.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;numeric&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt; (FIX1)</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Function</th>
<th>To query whether the specified slot is targeted for the Eye Diagram measurement:</th>
</tr>
</thead>
</table>

| Example                    | > EDS? 1                                                                         |
|----------------------------| < EDS 1                                                                          |

| Compatibility              | Incompatible with existing models.                                                |

**EMS <slot>,<numeric>[,<unit>]**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;slot&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
<td>Slot No. 1 to 6</td>
</tr>
<tr>
<td>1 to 6</td>
<td></td>
</tr>
<tr>
<td>1 to 6 when using the MP1800A, 1 to 4 when using the MT1810A.</td>
<td></td>
</tr>
<tr>
<td>&lt;numeric&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>ON</td>
</tr>
<tr>
<td>0</td>
<td>OFF</td>
</tr>
<tr>
<td>[&lt;unit&gt;] = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
<td>Mainframe No. 1 to 4</td>
</tr>
<tr>
<td>1 to 4</td>
<td></td>
</tr>
<tr>
<td>Specify the mainframe No. when connected to the MT1810A.</td>
<td></td>
</tr>
<tr>
<td>&lt;unit&gt; is optional. When omitted, 1 is specified.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Function</th>
<th>Selects the slot for Mask Test measurement.</th>
</tr>
</thead>
</table>

| Example                    | > EMS 1, 1                                    |

| Compatibility              | Incompatible with existing models.            |
### Automatic Measurement Commands

#### EMS? <slot>[,<unit>]

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;slot&gt;</td>
<td>&lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
</tr>
<tr>
<td>1 to 6</td>
<td>Slot No. 1 to 6</td>
</tr>
<tr>
<td>1 to 6</td>
<td>when using the MP1800A, 1 to 4 when using the MT1810A.</td>
</tr>
<tr>
<td>[&lt;unit&gt;]</td>
<td>&lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
</tr>
<tr>
<td>1 to 4</td>
<td>Mainframe No. 1 to 4</td>
</tr>
<tr>
<td>&lt;unit&gt;</td>
<td>Optional. When omitted, 1 is specified.</td>
</tr>
</tbody>
</table>

| Response        | <numeric> = <NR1 NUMERIC RESPONSE DATA> (FIX1)                              |

| Function        | Queries whether the specified slot is an object of Mask Test measurement.  |

| Example         | To query whether the specified slot is an object of Mask Test measurement:  |
|-----------------| > EMS? 1                                                                   |
|                 | < EMS 1                                                                   |

| Compatibility   | Incompatible with existing models.                                        |

#### EMR <type>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;type&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Coarse mode</td>
</tr>
<tr>
<td>1</td>
<td>Fine mode</td>
</tr>
</tbody>
</table>

| Function        | Sets the resolution of the measurement error rate.                       |

| Example         | To set the resolution of the measurement error rate to Fine:             |
|-----------------| > EMR 1                                                                   |

| Compatibility   | Incompatible with existing models.                                       |

#### EMR?

| Response        | <type> = <NR1 NUMERIC RESPONSE DATA> (FIX1)                              |

| Function        | Queries the resolution of the measurement error rate.                    |

| Example         | > EMR?                                                                    |
|-----------------| < EMR 1                                                                  |

| Compatibility   | Incompatible with existing models.                                       |
### EMT <type>

**Parameter**  
<type> = <DECIMAL NUMERIC PROGRAM DATA>

- 0: OFF
- 1: Transition
- 2: Non Transition

**Function**  
Sets Transition/Non Transition error detection.

**Example**  
To set Transition/Non Transition error detection to Transition:

```
> EMT 1
```

**Compatibility**  
Incompatible with existing models.

### EMT?

**Response**  
<type> = <NR1 NUMERIC RESPONSE DATA> (FIX1)

**Function**  
Queries the Transition/Non Transition error detection setting state.

**Example**  
```
> EMT?
< EMT 1
```

**Compatibility**  
Incompatible with existing models.

### EYP <type>

**Parameter**  
<type> = <DECIMAL NUMERIC PROGRAM DATA>

- 3: 8 points
- 4: 16 points
- 5: 32 points
- 6: 64 points
- 7: 132 points

**Function**  
Sets the number of measurement points during Eye Diagram measurement.

**Example**  
To set the number of measurement points during Eye Diagram measurement to 8 points:

```
> EYP 3
```

**Compatibility**  
Incompatible with existing models.

### EYP?

**Response**  
<type> = <NR1 NUMERIC RESPONSE DATA> (FIX1)

**Function**  
Queries the number of measurement points during Eye Diagram measurement.

**Example**  
```
> EYP?
< EYP 3
```

**Compatibility**  
Incompatible with existing models.
### EDA <numeric>

**Parameter**

<numeric> = <DECIMAL NUMERIC PROGRAM DATA>
- 2: Auto Search ON (Fine)
- 1: Auto Search ON (Coarse)
- 0: Auto Search OFF

**Function**
Sets the Auto Search ON/OFF when starting Eye Diagram measurement.

**Example**
To set the Auto Search to On when starting Eye Diagram measurement:
> EDA 1

**Compatibility**
Incompatible with existing models.

### EDA?

**Response**
<numeric> = <NR1 NUMERIC RESPONSE DATA> (FIX1)

**Function**
Queries the Auto Search ON/OFF when starting Eye Diagram measurement.

**Example**
> EDA?
< EDA 1

**Compatibility**
Incompatible with existing models.
8.5.2.2 Diagram tab setting commands

Figure 8.5.2.2-1 Diagram tab
8.5.2.2.1 Graph display setting commands

Figure 8.5.2.2.1-1  Eye Diagram field on Diagram tab

Table 8.5.2.2.1-1  Diagram tab setting commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1]</td>
<td>No label (Setting of result display slot)</td>
<td>DPS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DPS?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ETS?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ETT?</td>
</tr>
<tr>
<td>[4]</td>
<td>Auto Scale</td>
<td>ASC</td>
</tr>
<tr>
<td>[5]</td>
<td>Max Step (Threshold)</td>
<td>VSC</td>
</tr>
<tr>
<td></td>
<td></td>
<td>VSC?</td>
</tr>
<tr>
<td>[6]</td>
<td>Max Step (Phase)</td>
<td>PSC</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PSC?</td>
</tr>
</tbody>
</table>
### DPS <slot>[,<unit>]

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;slot&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
<td>Slot No. 1 to 6</td>
</tr>
<tr>
<td>1 to 6 when using the MP1800A, 1 to 4 when using the MT1810A.</td>
<td></td>
</tr>
<tr>
<td>&lt;unit&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
<td>Mainframe No. 1 to 4</td>
</tr>
<tr>
<td>1 to 4 when using the MP1800A, 1 to 4 when using the MT1810A.</td>
<td></td>
</tr>
</tbody>
</table>

**Function**
Selects the Eye Diagram result display target slot.

**Example**
To set the Eye Diagram result display target slot to Slot1:

```plaintext
> DPS 1
```

**Compatibility**
Incompatible with existing models.

### DPS? [<unit>]

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>[&lt;unit&gt;] = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
<td>Mainframe No. 1 to 4</td>
</tr>
<tr>
<td>1 to 4 when using the MP1800A, 1 to 4 when using the MT1810A.</td>
<td></td>
</tr>
<tr>
<td>&lt;unit&gt; is optional. When omitted, 1 is specified.</td>
<td></td>
</tr>
</tbody>
</table>

**Response**
<br>

| <slot> = <NR1 NUMERIC RESPONSE DATA> (FIX2) |

**Function**
Queries the Eye Diagram result display target slot.

**Example**
To set the Eye Diagram result display target slot:

```plaintext
> DPS?
< DPS 1
< DPS 64
```

**Compatibility**
Incompatible with existing models.

### ETS <type>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;type&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
<td>Edit1</td>
</tr>
<tr>
<td>0</td>
<td>Edit2</td>
</tr>
<tr>
<td>1</td>
<td>Edit3</td>
</tr>
<tr>
<td>2</td>
<td>Edit4</td>
</tr>
<tr>
<td>3</td>
<td></td>
</tr>
</tbody>
</table>

**Function**
Sets the mask pattern for performing the Eye Diagram mask test.

**Example**
To set the mask pattern for performing the Eye Diagram mask test to Edit4:

```plaintext
> ETS 3
```

**Compatibility**
Incompatible with existing models.
8.5 Automatic Measurement Commands

**ETS?**

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;type&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt; (FIX1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function</td>
<td>Queries the mask pattern for performing the Eye Diagram mask test.</td>
</tr>
<tr>
<td>Example</td>
<td>&gt; ETS?</td>
</tr>
<tr>
<td></td>
<td>&lt; ETS 3</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Incompatible with existing models.</td>
</tr>
</tbody>
</table>

**ETT <thre>**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;thre&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>3 to 12</td>
</tr>
<tr>
<td></td>
<td>1E–3 to 1E–12, in single steps</td>
</tr>
<tr>
<td>Function</td>
<td>Sets the Error Threshold for performing the Eye Diagram mask test.</td>
</tr>
<tr>
<td>Example</td>
<td>To set the Error Threshold for performing the Eye Diagram mask test to E–3:</td>
</tr>
<tr>
<td></td>
<td>&gt; ETT 3</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Incompatible with existing models.</td>
</tr>
</tbody>
</table>

**ETT?**

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;thre&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt; (FIX2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function</td>
<td>Queries the Error Threshold for performing the Eye Diagram mask test.</td>
</tr>
<tr>
<td>Example</td>
<td>&gt; ETT?</td>
</tr>
<tr>
<td></td>
<td>&lt; ETT 3</td>
</tr>
<tr>
<td></td>
<td>&lt; ETT 12</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Incompatible with existing models.</td>
</tr>
</tbody>
</table>

**ASC**

<table>
<thead>
<tr>
<th>Function</th>
<th>Automatically adjusts the Eye Diagram measurement graph scale.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example</td>
<td>&gt; ASC</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Incompatible with existing models.</td>
</tr>
</tbody>
</table>
Chapter 8  Native Commands

VSC <max>,<step>

Parameter

- `<max> = <DECIMAL NUMERIC PROGRAM DATA>`
  - `-3.990 to 4.000 V, in 0.001-V steps`
- `<step> = <DECIMAL NUMERIC PROGRAM DATA>`
  - `0.001 to 0.800 V, in 0.001-V steps`

The range varies depending on the option status and input interface setting.

Function

Sets the maximum value in the direction of the Threshold of the Eye Diagram graph, and the grid resolution.

Example

To set the maximum value in the direction of the Threshold of the Eye Diagram graph to 4.000 V, and the grid resolution to 0.800 V:

```
> VSC 4.000,0.800
```

Compatibility

Incompatible with existing models.

VSC?

Response

- `<max> = <NR2 NUMERIC RESPONSE DATA>(FIX6)`
  - `-3.990 to 4.000 V`
- `<step> = <NR2 NUMERIC RESPONSE DATA>(FIX5)`
  - `0.001 to 0.800 V`

Function

Queries the maximum value in the direction of the Threshold of the Eye Diagram graph, and the grid resolution.

Example

```
> VSC?
<VSC 4.000,0.800
<VSC -3.000,0.001
```

Compatibility

Incompatible with existing models.

PSC <max>,<step>

Parameter

- `<max> = <DECIMAL NUMERIC PROGRAM DATA>`
  - `-990 to 1000 mUI, in 1-mUI steps`
- `<step> = <DECIMAL NUMERIC PROGRAM DATA>`
  - `1 to 200 mUI, in 1-mUI steps`

Function

Sets the maximum value in the direction of the Phase of the Eye Diagram graph, and the grid resolution.

Example

To set the maximum value in the direction of the Phase of the Eye Diagram graph to 0, and the grid resolution to 10 mUI:

```
> PSC 0,10
```
## PSC?

<table>
<thead>
<tr>
<th>Response</th>
<th>(&lt;\text{min}&gt; = \text{&lt;NR1 NUMERIC RESPONSE DATA&gt; (FIX4)})</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(-990) to (1000) (-990) to (1000) mUI</td>
</tr>
<tr>
<td></td>
<td>(&lt;\text{step}&gt; = \text{&lt;NR1 NUMERIC RESPONSE DATA&gt; (FIX3)})</td>
</tr>
<tr>
<td></td>
<td>(1) to (200) (1) to (200) mUI</td>
</tr>
</tbody>
</table>

| Function | Queries the maximum value in the direction of the Phase of the Eye Diagram graph, and the grid resolution. |
| Example  | > PSC?                                                     |
|          | < PSC \(-990\), \(1\)                                  |
|          | < PSC \(0\), \(1\)                                    |
|          | < PSC \(1000\), \(200\)                               |

| Compatibility | Incompatible with existing models. |
8.5.2.2.2 Condition setting commands

Figure 8.5.2.2.2-1  Condition, Detail and Result tab selection on Diagram tab
8.5 Automatic Measurement Commands

Figure 8.5.2.2.2-2 Condition setting screen (Estimate)
### Table 8.5.2.2-1  Diagram tab setting commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1]</td>
<td>Select tab</td>
<td>DTB</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DTB?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ETR?</td>
</tr>
<tr>
<td>[3]</td>
<td>Meas set All</td>
<td>DAS</td>
</tr>
<tr>
<td></td>
<td>Meas reset All</td>
<td></td>
</tr>
<tr>
<td>[4]</td>
<td>Display</td>
<td>DER</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DER?</td>
</tr>
<tr>
<td>[5]</td>
<td>Display set All</td>
<td>DAR</td>
</tr>
<tr>
<td></td>
<td>Display reset All</td>
<td></td>
</tr>
<tr>
<td>[6]</td>
<td>Setting of error rate for</td>
<td>DES</td>
</tr>
<tr>
<td></td>
<td>Estimate measurement</td>
<td>DES?</td>
</tr>
<tr>
<td>[7]</td>
<td>Mask ON/OFF</td>
<td>DPM</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DPM?</td>
</tr>
<tr>
<td>[8]</td>
<td>Mask Adjust</td>
<td>DPA</td>
</tr>
<tr>
<td>[9]</td>
<td>Actual/Estimate</td>
<td>DPE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DPE?</td>
</tr>
<tr>
<td>[10]</td>
<td>Meas.ratio</td>
<td>DBT</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DBT?</td>
</tr>
<tr>
<td>[11]</td>
<td>Query for measurement point</td>
<td>EDP?</td>
</tr>
<tr>
<td>[12]</td>
<td>Query for measurement result</td>
<td>EDM?</td>
</tr>
</tbody>
</table>

#### DTB <tab>

Parameter

<tab> = <DECIMAL NUMERIC PROGRAM DATA>

- 0  Condition tab
- 1  Detail tab
- 2  Result tab

Function

Switches the child tab display on the Diagram tab.

Example

To switch to the Result tab on the Diagram tab:

```javascript
> DTB 2
```

Compatibility

Incompatible with existing models.

#### DTB?

Response

<tab> = <NR1 NUMERIC RESPONSE DATA> (FIX1)

- 0 to 2

Function

Queries the child tab currently active on the Diagram tab.

Example

```javascript
> DTB?
< DTB 2
```

Compatibility

Incompatible with existing models.
8.5 Automatic Measurement Commands

**ETR <thre>**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;thre&gt;</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>B1000000000</td>
<td>1E–3</td>
<td></td>
</tr>
<tr>
<td>B0100000000</td>
<td>1E–4</td>
<td></td>
</tr>
<tr>
<td>B0010000000</td>
<td>1E–5</td>
<td></td>
</tr>
<tr>
<td>B0001000000</td>
<td>1E–6</td>
<td></td>
</tr>
<tr>
<td>B0000100000</td>
<td>1E–7</td>
<td></td>
</tr>
<tr>
<td>B0000010000</td>
<td>1E–8</td>
<td></td>
</tr>
<tr>
<td>B0000001000</td>
<td>1E–9</td>
<td></td>
</tr>
<tr>
<td>B0000000100</td>
<td>1E–10</td>
<td></td>
</tr>
<tr>
<td>B0000000010</td>
<td>1E–11</td>
<td></td>
</tr>
<tr>
<td>B0000000001</td>
<td>1E–12</td>
<td></td>
</tr>
</tbody>
</table>

**Note:**
Set 1 only for bits that are set to ON.

**Function**
Selects the Eye Diagram measurement target error rate.

**Example**
To set the Eye Diagram measurement target error rates 1E–3, 1E–7, and 1E–12 to set ON:

> ETR B1000100001

**Compatibility**
Incompatible with existing models.

**ETR?**

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;thre&gt;</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>&lt;thre&gt;</td>
<td>= &lt;NUMERIC RESPONSE DATA&gt; (FIX11)</td>
</tr>
</tbody>
</table>

**Function**
Queries the Eye Diagram measurement target error rate.

**Example**

> ETR?

< ETR B1000100001

**Compatibility**
Incompatible with existing models.

**DAS <numeric>**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;numeric&gt;</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>OFF</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>ON</td>
<td></td>
</tr>
</tbody>
</table>

**Function**
Sets all Eye Diagram measurement target error rates to ON/OFF.

**Example**
To set all Eye Diagram measurement target error rates to ON:

> DAS 1

**Compatibility**
Incompatible with existing models.
DER <thre>

Parameter

\(<\text{thre}> = \langle\text{BINARY NUMERIC PROGRAM DATA}\rangle\\nB100000000000000 = 1E–3\\nB010000000000000 = 1E–4\\nB001000000000000 = 1E–5\\nB000100000000000 = 1E–6\\nB000010000000000 = 1E–7\\nB000001000000000 = 1E–8\\nB000000100000000 = 1E–9\\nB000000010000000 = 1E–10\\nB000000001000000 = 1E–11\\nB000000000100000 = 1E–12\\nB000000000010000 = 1E–13\\nB000000000001000 = 1E–14\\nB000000000000100 = 1E–15\\nB000000000000010 = 1E–16\\nB000000000000001 = 1E–17

Note:
Set 1 only for bits that are set to ON.

Function
Selects the Eye Diagram measurement result display target.

Example
To change the Eye Diagram measurement result display targets 1E–12 and 1E–17 to ON:
> DER B000000000100001

Compatibility
Incompatible with existing models.

DER?

Response
\(<\text{thre}> = \langle\text{NUMERIC RESPONSE DATA}\rangle (\text{FIX16})

Function
Queries the Eye Diagram measurement result display target.

Example
> DER?
< DER B0000000000100001

Compatibility
Incompatible with existing models.

DAR <numeric>

Parameter
\(<\text{numeric}> = \langle\text{DECIMAL NUMERIC PROGRAM DATA}\rangle\\n0 = \text{OFF}\\n1 = \text{ON}

Function
Sets all Eye Diagram measurement result display targets to ON/OFF.

Example
To change all Eye Diagram measurement result display targets to ON:
> DAR 1

Compatibility
Incompatible with existing models.
### DES <numeric>

| Parameter | <numeric> = <DECIMAL PROGRAM DATA>  
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>13 to 199</td>
</tr>
<tr>
<td></td>
<td>1E–13 to 1E–199, in single steps</td>
</tr>
<tr>
<td></td>
<td>Up to five parameters can be set, delimited with a comma (,).</td>
</tr>
<tr>
<td>Function</td>
<td>Sets the result display error rate in Estimate measurement from E–13 to E–199.</td>
</tr>
</tbody>
</table>
| Example   | To display the result display error rate in Estimate measurement for  
|           | 1E–15, 1E–100, 1E–150, 1E–199, and 1E–180: |  
|           | >DES 15, 100, 150, 199, 180 |  
| Compatibility | Incompatible with existing models. |  

### DES?

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;numeric&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt; (FIX3) (FIX3) (FIX3) (FIX3) (FIX3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function</td>
<td>Queries the result display error rate in Estimate measurement from E–13 to E–199.</td>
</tr>
<tr>
<td>Example</td>
<td>&gt; DES?</td>
</tr>
<tr>
<td></td>
<td>&lt; DES 15, 100, 150, 199, 180</td>
</tr>
<tr>
<td></td>
<td>&lt; DES 15, 30, 70, 80, 99</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Incompatible with existing models.</td>
</tr>
</tbody>
</table>

### DPM <numeric>

| Parameter | <numeric> = <DECIMAL NUMERIC PROGRAM DATA>  
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1                  Display</td>
</tr>
<tr>
<td></td>
<td>0                  Clear</td>
</tr>
<tr>
<td>Function</td>
<td>Selects between displaying and clearing the mask during Eye Diagram measurement.</td>
</tr>
<tr>
<td>Example</td>
<td>To display the mask during Eye Diagram measurement:</td>
</tr>
<tr>
<td></td>
<td>&gt; DPM 1</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Incompatible with existing models.</td>
</tr>
</tbody>
</table>

### DPM?

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;numeric&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt; (FIX1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function</td>
<td>Queries the setting whether to display or clear the mask during Eye Diagram measurement.</td>
</tr>
<tr>
<td>Example</td>
<td>&gt; DPM?</td>
</tr>
<tr>
<td></td>
<td>&lt; DPM 1</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Incompatible with existing models.</td>
</tr>
</tbody>
</table>

### Chapter 8  Native Commands

#### DPA

<table>
<thead>
<tr>
<th>Function</th>
<th>Automatically adjusts the mask during Eye Diagram measurement.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example</td>
<td>&gt; DPA</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Incompatible with existing models.</td>
</tr>
</tbody>
</table>

#### DPE <type>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;type&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0   Actual measurement</td>
</tr>
<tr>
<td></td>
<td>1   Estimate measurement</td>
</tr>
</tbody>
</table>

**Note:**
The Estimate measurement cannot be selected when the number of measurement points is set to 8.

<table>
<thead>
<tr>
<th>Function</th>
<th>Selects between Actual and Estimate measurement during Eye Diagram measurement.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example</td>
<td>To set Actual measurement during Eye Diagram measurement:</td>
</tr>
<tr>
<td></td>
<td>&gt; DPE 0</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Incompatible with existing models.</td>
</tr>
</tbody>
</table>

#### DPE?

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;type&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt; (FIX1)</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Function</th>
<th>Queries the Actual/Estimate measurement state during Eye Diagram measurement.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example</td>
<td>&gt; DPE?</td>
</tr>
<tr>
<td></td>
<td>&lt; DPE 0</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Incompatible with existing models.</td>
</tr>
</tbody>
</table>

#### DBT <upper>,<lower>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;upper&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>3 to 7  1E–3 to 1E–7, in single steps</td>
</tr>
<tr>
<td></td>
<td>&lt;lower&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
</tr>
<tr>
<td></td>
<td>7 to 12 1E–7 to E–12, in single steps</td>
</tr>
</tbody>
</table>

**Note:**
The difference between the <upper> and <lower> value must be 3 or greater.

<table>
<thead>
<tr>
<th>Function</th>
<th>Sets the reference error rate during Eye Diagram Estimate.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example</td>
<td>To set the reference error rate during Eye Diagram Estimate to E–3 or E–12:</td>
</tr>
<tr>
<td></td>
<td>&gt; DBT 3, 12</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Incompatible with existing models.</td>
</tr>
</tbody>
</table>
8.5  Automatic Measurement Commands

**DBT?**

Response  
<upper> = <NR1 NUMERIC RESPONSE DATA> (FIX1)  
<lower> = <NR1 NUMERIC RESPONSE DATA> (FIX2)

Function  
Queries the reference error rate during Eye Diagram Estimate.

Example  
> DBT?
< DBT 3,12
< DBT 3,7

Compatibility  
Incompatible with existing models.

**EDP? <numeric>,<thre>[,<unit>]**

Parameter  
<numeric> = <DECIMAL NUMERIC PROGRAM DATA>  
1 to 6  
1 to 6 when using the MP1800A, 1 to 4 when using the MT1810A.  
<thre> = <DECIMAL NUMERIC PROGRAM DATA>  
3 to 12  
1E–3 to 1E–12  
[unit] = <DECIMAL NUMERIC PROGRAM DATA>  
1 to 4  
Mainframe No. 1 to 4  
Specify the mainframe No. when connected to the MT1810A.  
[unit] is optional. When omitted, 1 is specified.

Response  
<numeric> = <NR1 NUMERIC RESPONSE DATA> (FIX3)  
1 to 132  
Returns a valid point.  
0  
Returns 0 when all points are invalid.  
Multiple points are delimited with a comma (,) and returned.

Function  
Queries the valid points in each threshold during the Eye Diagram measurement for a specific slot.

Example  
To query the valid points in slot No. 1 during measurement with threshold $1\times 10^{-3}$:  
> EDP? 1,3
< EDP 1, 2, 3, 50,132
< EDP 0

Compatibility  
Incompatible with existing models.
EDM? <numeric1>,<numeric2>,<numeric3>[,<unit>]

**Parameter**

- `<numeric1>` = `<DECIMAL NUMERIC PROGRAM DATA>`
  - 1 to 6 Slot No. 1 to 6
- `<numeric2>` = `<DECIMAL NUMERIC PROGRAM DATA>`
  - 3 to 12 1E–3 to 1E–12, for error rate specification
- `<numeric3>` = `<DECIMAL NUMERIC PROGRAM DATA>`
  - 1 to 132 Measurement points
- `[<unit>]` = `<DECIMAL NUMERIC PROGRAM DATA>`
  - 1 to 4 Mainframe No. 1 to 4
  - Specify the mainframe No. when connected to the MT1810A.
  - `<unit>` is optional. When omitted, 1 is specified.

**Response**

- `<phase>` = `<NR1 NUMERIC RESPONSE DATA>`(FIX5)
  - –1000 to 1000 Phase data (mUI)
  - XXXXX Phase data (ps)

**Note:**

The converted value from mUI

When no measurement result exists, "-----" is returned.

- `<threshold>` = `<NR1 NUMERIC RESPONSE DATA>` (FIX5)
  - YYYYY Threshold data (mV)

When no measurement result exists, "-----" is returned.

**Function**

Queries the Eye Diagram measurement results.

**Example**

To query the Eye Diagram measurement results at Slot 3, error rate 1E–7, measurement point 1 (measurement point: –300 mUI, 500 mV):

```
> EDM? 3,7,1
< EDM -300, 500
< EDM -----,-----
```

**Compatibility**

Incompatible with existing models.
8.5.2.2.3 Detail setting commands

Figure 8.5.2.2.3-1 Detail Mask Point Test screen

Figure 8.5.2.2.3-2 Detail Mask Line Test screen
### Table 8.5.2.2.3-1  Detail setting commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1]</td>
<td>Mask Adjust</td>
<td>DMK</td>
</tr>
<tr>
<td>[2]</td>
<td>Slot selection</td>
<td>DTS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DTS?</td>
</tr>
<tr>
<td>[3]</td>
<td>Test Mode</td>
<td>DMO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DMO?</td>
</tr>
<tr>
<td>[4]</td>
<td>Start/Stop</td>
<td>DST</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DSP</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DSA?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DLR?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DMR?</td>
</tr>
<tr>
<td>[7]</td>
<td>Rotation Direction</td>
<td>DLT</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DLT?</td>
</tr>
<tr>
<td>[8]</td>
<td>Query for measurement point</td>
<td>DPN?</td>
</tr>
<tr>
<td>[9]</td>
<td>Query for Point test result</td>
<td>DDP?</td>
</tr>
<tr>
<td>[10]</td>
<td>Query for Line test result</td>
<td>DDL?</td>
</tr>
</tbody>
</table>

### DMK

**Function**

Performs automatic mask adjustment during Eye Diagram Detail measurement.

**Example**

> DMK

**Compatibility**

Incompatible with existing models.

### DTS <slot>[,<unit>]

**Parameter**

- `<slot>` = `<DECIMAL NUMERIC PROGRAM DATA>`
- 1 to 6  
  Slot No. 1 to 6
- 1 to 6 when using the MP1800A, 1 to 4 when using the MT1810A.
- `<unit>` = `<DECIMAL NUMERIC PROGRAM DATA>`
- 1 to 4  
  Mainframe No. 1 to 4
- Specify the mainframe No. when connected to the MT1810A.
- `<unit>` is optional. When omitted, 1 is specified.

**Function**

Selects the Mask pass/fail measurement target slot during Eye Diagram measurement.

**Example**

To set the Mask pass/fail measurement target slot during Eye Diagram measurement to Slot1:

> DTS 1

**Compatibility**

Incompatible with existing models.
### DTS? [<unit>]

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;unit&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
<td>Specify the mainframe No. when connected to the MT1810A. &lt;unit&gt; is optional. When omitted, 1 is specified.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Response</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;slot&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt; (FIX2)</td>
<td>Queries the Mask pass/fail measurement target slot during Eye Diagram measurement.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Example</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>&gt; DTS?</td>
<td>&lt; DTS 1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Compatibility</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Incompatible with existing models.</td>
<td></td>
</tr>
</tbody>
</table>

### DMO <mode>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;mode&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>Point Test</td>
</tr>
<tr>
<td>1</td>
<td>Mask Line Test</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Selects the measurement mode during Eye Diagram Detail measurement.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Example</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>To set the measurement mode during Eye Diagram Detail measurement to Point Test mode:</td>
<td></td>
</tr>
<tr>
<td>&gt; DMO 0</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Compatibility</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Incompatible with existing models.</td>
<td></td>
</tr>
</tbody>
</table>

### DMO? |

<table>
<thead>
<tr>
<th>Response</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;mode&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt; (FIX1)</td>
<td>Queries the measurement mode during Eye Diagram Detail measurement.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Example</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&gt; DMO?</td>
<td>&lt; DMO 0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Compatibility</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Incompatible with existing models.</td>
<td></td>
</tr>
</tbody>
</table>

### DST

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Starts the Mask pass/fail judgment during Eye Diagram Detail measurement.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Example</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&gt; DST</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Compatibility</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Incompatible with existing models.</td>
<td></td>
</tr>
</tbody>
</table>
### DSP

<table>
<thead>
<tr>
<th>Function</th>
<th>Stops the Mask pass/fail judgment during Eye Diagram Detail measurement.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example</td>
<td>&gt; DSP</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Incompatible with existing models.</td>
</tr>
</tbody>
</table>

### DSA?

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;numeric&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt; (FIX1)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1 Start</td>
</tr>
<tr>
<td></td>
<td>0 Stop</td>
</tr>
<tr>
<td>Function</td>
<td>Queries the Mask pass/fail judgment status during Eye Diagram Detail measurement.</td>
</tr>
<tr>
<td>Example</td>
<td>&gt; DSA?</td>
</tr>
<tr>
<td></td>
<td>&lt; DSA 1</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Incompatible with existing models.</td>
</tr>
</tbody>
</table>

### DLR <range>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;range&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0 ALL</td>
</tr>
<tr>
<td></td>
<td>1 Marker</td>
</tr>
<tr>
<td>Function</td>
<td>Selects the measurement range during Eye Diagram Detail Line measurement.</td>
</tr>
<tr>
<td>Example</td>
<td>To set the measurement range during Eye Diagram Detail Line measurement to Marker:</td>
</tr>
<tr>
<td></td>
<td>&gt; DLR 1</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Incompatible with existing models.</td>
</tr>
</tbody>
</table>

### DLR?

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;range&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt; (FIX1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function</td>
<td>Queries the measurement range during Eye Diagram Detail Line measurement.</td>
</tr>
<tr>
<td>Example</td>
<td>&gt; DLR?</td>
</tr>
<tr>
<td></td>
<td>&lt; DLR 1</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Incompatible with existing models.</td>
</tr>
</tbody>
</table>
8.5 Automatic Measurement Commands

DMR <marker>,<phase>,<voltage>

Parameter

<marker> = <DECIMAL NUMERIC PROGRAM DATA>
1 Marker 1
2 Marker 2

<phase> = <DECIMAL NUMERIC PROGRAM DATA>
–1000 to 1000 –1000 to 1000 mUI, in 1-mUI steps
(horizontal direction)

<voltage> = <DECIMAL NUMERIC PROGRAM DATA>
–4000 to 4000 –4000 to 4000 mV, in 1-mV steps
(vertical direction)

Note:
Sets the value of Marker 2 must be equal to or greater than that of
marker 1 in the phase direction.
The marker is moved onto the mask line closest to the set point.

Function
Sets the positions for the measurement range selection markers 1 and 2
during Eye Diagram Detail Line measurement.

Example
To set the position for marker 1 (100 mUI, 2000 mV):
> DMR 1,100,2000

Compatibility
Incompatible with existing models.

DMR? <marker>

Parameter

<marker> = <DECIMAL NUMERIC PROGRAM DATA>
1 Marker 1
2 Marker 2

<phase>,<voltage> (FIX5)

Response

<phase> = <NR2 NUMERIC RESPONSE DATA>
–1000 to 1000 –1000 to 1000 mUI (horizontal direction)
XXXXX XXXXX ps

Note:
The value converted from mUI

<voltage> = <NR2 NUMERIC RESPONSE DATA> (FIX5)
–4000 to 4000 –4000 to 4000 mV (vertical direction)

Function
Queries the positions for the measurement range selection markers 1
and 2 during Eye Diagram Detail Line measurement.

Example
To query the set value for marker 1 (1,000 V, 100 mUI):
> DMR? 1
< DMR -1000,-4000
< DMR 0, 0
< DMR 1000, 4000

Compatibility
Incompatible with existing models.
Chapter 8  Native Commands

**DLT <type>**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;type&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Counterclockwise</td>
</tr>
<tr>
<td>1</td>
<td>Clockwise</td>
</tr>
</tbody>
</table>

**Function**
Selects the Mask pass/fail judgment direction during Eye Diagram Detail Line measurement.

**Example**
To set the Mask pass/fail judgment direction during Eye Diagram Detail Line measurement to clockwise:
> DLT 1

**Compatibility**
Incompatible with existing models.

**DLT?**

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;type&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt; (FIX1)</th>
</tr>
</thead>
</table>

**Function**
Queries the Mask pass/fail judgment direction during Eye Diagram Detail Line measurement.

**Example**
> DLT?
< DLT 1

**Compatibility**
Incompatible with existing models.

**DPN?**

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;numeric&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt; (FIX5)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 to 20000</td>
<td>Detail measurement point</td>
</tr>
</tbody>
</table>

**Function**
Queries the number of points during Eye Diagram Detail measurement.

**Example**
> DPN?
< DPN 1000
< DPN 20000

**Compatibility**
Incompatible with existing models.
8.5 Automatic Measurement Commands

**DDP? <point>**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;point&gt;</td>
<td>= &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
<td>1 to 32 Measurement point</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;numeric1&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt; (FIX5)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>–1000 to 1000 Phase data (mUI)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>XXXXX Phase data (ps)</td>
</tr>
<tr>
<td>Note:</td>
<td></td>
<td>Displays in the selected unit.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;numeric2&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt; (FIX5)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>–4000 to 4000 Threshold data (mV)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;numeric3&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt; (FIX1)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 OK</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 NG</td>
</tr>
<tr>
<td>Function</td>
<td>Queries the measurement result of the Eye Diagram Detail Point Test.</td>
<td></td>
</tr>
<tr>
<td>Example</td>
<td>To query the measurement result of the Eye Diagram Detail Point Test at measurement point 1:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&gt; DDP? 1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&lt; DDP -1000,-4000,0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&lt; DDP 0, 0,0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&lt; DDP 1000, 4000,1</td>
<td></td>
</tr>
<tr>
<td>Compatibility</td>
<td>Incompatible with existing models.</td>
<td></td>
</tr>
</tbody>
</table>
### DDL? <point>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Response</th>
<th>Note:</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;point&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
<td>&lt;numeric1&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt; (FIX5)</td>
<td>Displays in the selected unit.</td>
</tr>
<tr>
<td>1 to 20000 Measurement point</td>
<td>–1000 to 1000 Phase data (mUI)</td>
<td>When no data exists, &quot;-----&quot; is displayed.</td>
</tr>
<tr>
<td>10000</td>
<td>XXXXX Phase data (ps)</td>
<td>When no data exists, &quot;-----&quot; is displayed.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Note: Displays in the selected unit.</td>
</tr>
<tr>
<td>Function</td>
<td>Example</td>
<td>Compatibility</td>
</tr>
<tr>
<td>Queries the measurement result of the Eye Diagram Detail Mask Line Test.</td>
<td>To query the measurement result of the Eye Diagram Detail Mask Line Test at measurement point 1:</td>
<td></td>
</tr>
<tr>
<td>Compatibility</td>
<td></td>
<td>Incompatible with existing models.</td>
</tr>
</tbody>
</table>

To query the measurement result of the Eye Diagram Detail Mask Line Test at measurement point 1:

```
> DDL? 1
< DDL -1000,-4000,0
< DDL 0, 0,0
< DDL 1000, 4000,1
```
8.5.2.2.4 Result query commands

![Figure 8.5.2.2.4-1 Result tab](image)

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1]</td>
<td>Query for Mask Test result</td>
<td>MSK?</td>
</tr>
</tbody>
</table>

Table 8.5.2.2.4-1 Result query command
### MSK? <numeric>[,<unit>]

**Parameter**  
<numeric> = <DECIMAL NUMERIC PROGRAM DATA>  
1 to 6 \[ Slot No. 1 to 6 \]  
1 to 6 when using the MP1800A, 1 to 4 when using the MT1810A.  
[unit] = <DECIMAL NUMERIC PROGRAM DATA>  
1 to 4 \[ Mainframe No. 1 to 4 \]  
Specify the mainframe No. when connected to the MT1810A.  
[unit] is optional. When omitted, 1 is specified.

**Response**  
<numeric1>,<numeric2>  
<numeric1> = <NR1 NUMERIC RESPONSE DATA>(FIX1)  
0 \[ OK \]  
1 \[ NG \]  
<numeric2> = <NR1 NUMERIC RESPONSE DATA>(FIX2)  
0 to 32 \[ NG mask points \]  

**Note:**  
0 is returned when all items are OK  
Multiple NGs are delimited with a comma (,) and returned for the corresponding number of NGs.

**Function**  
Queries the measurement result of the Eye Diagram Mask Test.

**Example**  
To query the Mask Test result for Slot 1:  
> MSK? 1  
< MSK 1,2,4,32  
< MSK 0,0  

**Compatibility**  
Incompatible with existing models.
8.5.2.2.5 Diagram tab (Status) setting commands

[Figure 8.5.2.2.5-1] Diagram tab Start/Stop buttons and Status field

[Table 8.5.2.2.5-1] Diagram tab (Status) setting commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1]</td>
<td>Start</td>
<td>DTA</td>
</tr>
<tr>
<td>[2]</td>
<td>Stop</td>
<td>DTO</td>
</tr>
<tr>
<td>[3]</td>
<td>Query for measurement state</td>
<td>MSR?</td>
</tr>
<tr>
<td>[4]</td>
<td>No label (Setting of time display)</td>
<td>ETM</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ETM?</td>
</tr>
<tr>
<td>[5]</td>
<td>Query for current time</td>
<td>EDT?</td>
</tr>
<tr>
<td>[6]</td>
<td>Query for measurement start time</td>
<td>MSA?</td>
</tr>
<tr>
<td>[7]</td>
<td>Query for measurement elapsed time</td>
<td>ELP?</td>
</tr>
<tr>
<td>[8]</td>
<td>Query for measurement status</td>
<td>EDW?</td>
</tr>
</tbody>
</table>
Chapter 8  Native Commands

**DTA**

<table>
<thead>
<tr>
<th>Function</th>
<th>Starts the Eye Diagram measurement.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example</td>
<td>&gt; DTA</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Incompatible with existing models.</td>
</tr>
</tbody>
</table>

**DTO**

<table>
<thead>
<tr>
<th>Function</th>
<th>Stops the Eye Diagram measurement.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example</td>
<td>&gt; DTO</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Incompatible with existing models.</td>
</tr>
</tbody>
</table>

**MSR?**

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;numeric&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt; (FIX1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>During measurement</td>
</tr>
<tr>
<td>0</td>
<td>Measurement has been stopped.</td>
</tr>
<tr>
<td>2</td>
<td>Measurement has failed.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Function</th>
<th>Queries the Eye Diagram measurement status.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example</td>
<td>&gt; MSR?</td>
</tr>
<tr>
<td></td>
<td>&lt; MSR 1</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Incompatible with existing models.</td>
</tr>
</tbody>
</table>

**ETM <type>**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;type&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Displays the current date and time.</td>
</tr>
<tr>
<td>1</td>
<td>Displays the measurement start time.</td>
</tr>
<tr>
<td>2</td>
<td>Displays the elapsed time based on the measurement period.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Function</th>
<th>Selects the Eye Diagram measurement time display type.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example</td>
<td>To set the Eye Diagram measurement time display type to measurement start time (Start Time):</td>
</tr>
<tr>
<td></td>
<td>&gt; ETM 1</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Incompatible with existing models.</td>
</tr>
</tbody>
</table>

**ETM?**

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;type&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt; (FIX1)</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Function</th>
<th>Queries the Eye Diagram measurement time display type.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example</td>
<td>&gt; ETM?</td>
</tr>
<tr>
<td></td>
<td>&lt; ETM 1</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Incompatible with existing models.</td>
</tr>
</tbody>
</table>
8.5 Automatic Measurement Commands

**EDT?**

**Response**

| <year> = <NR1 NUMERIC RESPONSE DATA> (FIX4) |
| 2000 to 2036 Year 2000 to 2036 |
| <month> = <NR1 NUMERIC RESPONSE DATA> (FIX2) |
| 01 to 12 January to December |
| <day> = <NR1 NUMERIC RESPONSE DATA> (FIX2) |
| 01 to 31 1st to 31st |
| <hour> = <NR1 NUMERIC RESPONSE DATA> (FIX2) |
| 00 to 23 0 to 23 hours |
| <minute> = <NR1 NUMERIC RESPONSE DATA> (FIX2) |
| 00 to 59 0 to 59 minutes |
| <second> = <NR1 NUMERIC RESPONSE DATA> (FIX2) |
| 00 to 59 0 to 59 seconds |

**Function**
Queries the current date and time during Eye Diagram measurement.

**Example**

> EDT?

< EDT 2006,04,01,23,59,59

**Compatibility**
Incompatible with existing models.

---

**MSA?**

**Response**

| <year> = <NR1 NUMERIC RESPONSE DATA> (FIX4) |
| 0000, 2000 to 2036 Year 2000 to 2036 |
| <month> = <NR1 NUMERIC RESPONSE DATA> (FIX2) |
| 00, 01 to 12 January to December |
| <day> = <NR1 NUMERIC RESPONSE DATA> (FIX2) |
| 00, 01 to 31 1st to 31st |
| <hour> = <NR1 NUMERIC RESPONSE DATA> (FIX2) |
| 00 to 23 0 to 23 hours |
| <minute> = <NR1 NUMERIC RESPONSE DATA> (FIX2) |
| 00 to 59 0 to 59 minutes |
| <second> = <NR1 NUMERIC RESPONSE DATA> (FIX2) |
| 00 to 59 0 to 59 seconds |

**Note:**

The following is output when no measurement is performed:

MSA 0000,00,00,00,00,00

**Function**
Queries the Eye Diagram measurement start time.

**Example**

> MSA?

< MSA 2006,04,01,23,59,59

**Compatibility**
Incompatible with existing models.
8-382

Chapter 8  Native Commands

ELP?
Response  
<day> = <NR1 NUMERIC RESPONSE DATA> (FIX2)  
00, 01 to 99  1 to 99 days  
<hour> = <NR1 NUMERIC RESPONSE DATA> (FIX2)  
00 to 23  0 to 23 hours  
<minute> = <NR1 NUMERIC RESPONSE DATA> (FIX2)  
00 to 59  0 to 59 minutes  
<second> = <NR1 NUMERIC RESPONSE DATA> (FIX2)  
00 to 59  0 to 59 seconds  

Note:  
The following is output when no measurement is performed:  
ELP 00,00,00,00

Function  
Queries the Eye Diagram measurement elapsed time (Elapsed Time).

Example  
> ELP?
< ELP 99,23,59,59

Compatibility  
Incompatible with existing models.

EDW?
Response  
<numeric> = <DECIMAL NUMERIC RESPONSE DATA> (FIX1)  
0  When no alarm exists.  
1  Sync Loss  
2  Clock Loss (CR Unlock)  
3  CR Unlock  
4  Out of range  
5  Frequency NG  
6  Illegal Error  

Note:  
Displays any one of the alarms.  
When two or more alarms have occurred, they are displayed in the  
following order:  
Illegal Error > Frequency NG > Clock Loss (CR Unlock) > Sync  
Loss > Out of range

Function  
Queries the status condition during Eye Diagram measurement.

Example  
> EDW?
< EDW 1

Compatibility  
Incompatible with existing models.
8.5 Automatic Measurement Commands

8.5.2.2.6 Mask Edit tab setting commands

![Diagram of Mask Edit tab](image)

Table 8.5.2.2.6-1 Mask Edit tab setting commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1]</td>
<td>Mask</td>
<td>TTP</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TTP?</td>
</tr>
<tr>
<td>[2]</td>
<td>Auto Scale</td>
<td>MAS</td>
</tr>
<tr>
<td>[3]</td>
<td>Max Step (Threshold)</td>
<td>MVS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MVS?</td>
</tr>
<tr>
<td>[4]</td>
<td>Max Step (Phase)</td>
<td>MPH</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MPH?</td>
</tr>
<tr>
<td>[5]</td>
<td>Sample</td>
<td>SAM</td>
</tr>
<tr>
<td>[6]</td>
<td>Query for mask point count</td>
<td>TPN?</td>
</tr>
<tr>
<td>[7]</td>
<td>Phase (mUI) Threshold</td>
<td>TPV</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TPV?</td>
</tr>
<tr>
<td>[8]</td>
<td>Plot</td>
<td>TPL</td>
</tr>
<tr>
<td>[9]</td>
<td>Delete</td>
<td>TDL</td>
</tr>
<tr>
<td>[10]</td>
<td>Clear</td>
<td>TCL</td>
</tr>
<tr>
<td>[12]</td>
<td>Horizontal</td>
<td>THO</td>
</tr>
</tbody>
</table>
Chapter 8  Native Commands

TTP <type>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;type&gt;</td>
<td>&lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
</tr>
<tr>
<td>0</td>
<td>Edit1</td>
</tr>
<tr>
<td>1</td>
<td>Edit2</td>
</tr>
<tr>
<td>2</td>
<td>Edit3</td>
</tr>
<tr>
<td>3</td>
<td>Edit4</td>
</tr>
</tbody>
</table>

Function
Sets the pattern to be edited during Eye Diagram measurement.

Example
To change the pattern to be edited during Eye Diagram measurement to Edit4:

> TTP 3

Compatibility
Incompatible with existing models.

TTP?

<table>
<thead>
<tr>
<th>Response</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;type&gt;</td>
<td>&lt;NR1 NUMERIC RESPONSE DATA&gt; (FIX1)</td>
</tr>
</tbody>
</table>

Function
Queries the pattern that was edited during Eye Diagram measurement.

Example

> TTP?
< TTP 3

Compatibility
Incompatible with existing models.

MAS

Function
Executes automatic adjustment of the Mask Edit measurement graph scale.

Example

> MAS

Compatibility
Incompatible with existing models.
8.5 Automatic Measurement Commands

**MVS <max>,<step>**

**Parameter**
- `<max>` = <DECIMAL NUMERIC PROGRAM DATA>
  - -3.990 to 4.000 –3.990 to 4.000 V, in 0.001 V steps
- `<step>` = <DECIMAL NUMERIC PROGRAM DATA>
  - 1 to 800 1 to 800 mV, in 1 mV steps

**Function**
Sets the maximum value and the grid resolution in the direction of the Threshold of the Mask Edit graph.

**Example**
To set the maximum value and the grid resolution in the direction of the Threshold of the Mask Edit graph to 4.000 V and 800 mV respectively:

```
> MVS 4.000,800
```

**Compatibility**
Incompatible with existing models.

**MVS?**

**Response**
- `<max>,<step>`
- `<max>` = <NR2 NUMERIC RESPONSE DATA> (FIX6)
- `<step>` = <NR2 NUMERIC RESPONSE DATA> (FIX3)

**Function**
Queries the maximum value and the grid resolution in the direction of the Threshold of the Mask Edit graph.

**Example**
```
> MVS?
< MVS -3.990, 1
< MVS 0.000, 1
< MVS 4.000,800
```

**Compatibility**
Incompatible with existing models.

**MPH <max>,<step>**

**Parameter**
- `<max>` = <DECIMAL NUMERIC PROGRAM DATA>
  - -990 to 1000 –990 to 1000 mUI, in 1-mUI steps
- `<step>` = <DECIMAL NUMERIC PROGRAM DATA>
  - 1 to 200 1 to 200 mUI, in 1- mUI steps

**Function**
Sets the maximum value and the grid resolution in the direction of the Phase of the Mask Edit graph.

**Example**
To set the maximum value in the direction of the Phase of the Mask Edit graph to 1000 and the grid resolution value to 10 mUI respectively:

```
> MPH 1000,10
```

**Compatibility**
Incompatible with existing models.
Chapter 8  Native Commands

MPH?

Response  
<max>,<step>  
<max> = <NR2 NUMERIC RESPONSE DATA> (FIX4)  
<step> = <NR2 NUMERIC RESPONSE DATA> (FIX3)

Function  
Queries the maximum value and the grid resolution in the direction of  
the Phase of the Mask Edit graph.

Example  
> MPH?  
< MPH -990, 1  
< MPH 1000,200

Compatibility  
Incompatible with existing models.

SAM <string>

Response  
<string> = <STRING PROGRAM DATA>  
"<drv>:\<dir>\<file>"  
<drv> = C, D, E, F  
<dir> = <dir1>\<dir2>\... (Omitted when root directory)  
<file> = File name

Function  
Expands the selected sample pattern to the Edit Pattern.

Example  
> SAM "C:\...\sample.MSK"

Compatibility  
Incompatible with existing models.

TPN?

Response  
<numeric> = <NR1 NUMERIC RESPONSE DATA> (FIX2)  
0 to 32 0 to 32 points

Function  
Queries the number of Mask points.

Example  
> TPN?  
< TPN 0  
< TPN 32

Compatibility  
Incompatible with existing models.
8.5 Automatic Measurement Commands

TPV <point>,<phase>,<voltage>

Parameter
- <point> = <DECIMAL NUMERIC PROGRAM DATA>
  - 1 to 32 points, in 1-point steps
- <phase> = <DECIMAL NUMERIC PROGRAM DATA>
  - –1000 to 1000 mUI, in 1-mUI steps
- <voltage> = <DECIMAL NUMERIC PROGRAM DATA>
  - –4.000 to 4.000 V, in 0.001-V steps

Function
Sets a Mask point position.

Example
To set a Mask point position (point: 1, phase: 1000, voltage: 4.000):
> TPV 1,1000,4.000

Compatibility
Incompatible with existing models.

TPV? <point>

Parameter
- <point> = <DECIMAL NUMERIC PROGRAM DATA>
  - 1 to 32 points

Response
- <phase> = <NR1 NUMERIC RESPONSE DATA> (FIX5)
- <voltage> = <NR2 NUMERIC RESPONSE DATA> (FIX6)

Function
Queries a Mask point position.

Example
To query the position of point 1:
> TPV? 1
< TPV -1000,-4.000
< TPV 1000,4.000

Compatibility
Incompatible with existing models.

TPL <phase>,<voltage>

Parameter
- <phase> = <DECIMAL NUMERIC PROGRAM DATA>
  - –1000 to 1000 mUI, in 1-mUI steps
- <voltage> = <DECIMAL NUMERIC PROGRAM DATA>
  - –4000 to 4000 mV, in 1-mV steps

Function
Sets a Mask point position.

Example
To set a Mask point position to 1000 mUI (phase), 4000 mV (voltage):
> TPL 1000,4000

Compatibility
Incompatible with existing models.
### TDL <pos>

**Parameter**

<pos> = <DECIMAL NUMERIC PROGRAM DATA>

1 to 32

1 to 32 points, in 1-point steps

**Function**

Deletes the selected Mask point.

**Example**

To delete Mask point 32:

> TDL 32

**Compatibility**

Incompatible with existing models.

### TCL

**Function**

Clears the selected Mask point.

**Example**

> TCL

**Compatibility**

Incompatible with existing models.

### TVE

**Function**

Plots a Mask pattern onto the vertically-symmetric position.

**Example**

> TVE

**Compatibility**

Incompatible with existing models.

### THO

**Function**

Plots a Mask pattern onto the horizontally-symmetric position.

**Example**

> THO

**Compatibility**

Incompatible with existing models.
8.5.3 Q measurement

This section describes the commands for Q measurement.

Figure 8.5.3-1  Q measurement screen
8.5.3.1 Graph display setting commands

Figure 8.5.3.1-1  Graph display screen (Threshold vs Q tab)

Figure 8.5.3.1-2  Graph display screen (Phase vs Q tab)
### 8.5 Automatic Measurement Commands

#### Table 8.5.3.1-1  Graph display setting commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1]</td>
<td>Measurement mode switching</td>
<td>QMO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>QMO?</td>
</tr>
<tr>
<td>[2]</td>
<td>Auto Scale</td>
<td>QSC</td>
</tr>
<tr>
<td></td>
<td>Min</td>
<td>QSV?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>QSH?</td>
</tr>
<tr>
<td>[5]</td>
<td>Threshold vs. Q Display graph selection</td>
<td>QTI</td>
</tr>
<tr>
<td></td>
<td></td>
<td>QTI?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>QMN?</td>
</tr>
<tr>
<td>[7]</td>
<td>Phase vs. Q Display graph selection</td>
<td>QPI</td>
</tr>
<tr>
<td></td>
<td></td>
<td>QPI?</td>
</tr>
<tr>
<td>[8]</td>
<td>Phase vs. Q Current No.</td>
<td>QPM</td>
</tr>
<tr>
<td></td>
<td></td>
<td>QPM?</td>
</tr>
</tbody>
</table>

**QMO <mode>**

Parameter: `<mode> = <DECIMAL NUMERIC PROGRAM DATA>`

- **Vth vs. Q measurement**
- **Phase vs. Q measurement**

Function: Sets the graph display item from Vth vs. Q measurement or Phase vs. Q measurement.

Example: To set to Vth vs. Q measurement:

> QMO 1

Compatibility: Incompatible with existing models.

**QMO?**

Response: `<mode> = <NR1 NUMERIC RESPONSE DATA> (FIX1)`

Function: Queries the graph display item.

Example: To query the graph display item:

> QMO?

< QMO 1

Compatibility: Incompatible with existing models.

**QSC**

Function: Automatically adjusts the graph display during Q measurement.

Example: To adjust the graph display:

> QSC

Compatibility: Incompatible with existing models.
## Chapter 8  Native Commands

### QSV <max>[,<step>]

Parameter  

\[ \text{<max>[,<step>] = <DECIMAL NUMERIC PROGRAM DATA>} \]

<table>
<thead>
<tr>
<th>Table 8.5.3.1-2  Parameters for Threshold vs. Q tab</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Format</strong></td>
</tr>
<tr>
<td>Vth vs BER(Y = Log)</td>
</tr>
<tr>
<td>Vth vs BER(Y = Log(−Ln))</td>
</tr>
<tr>
<td>Vth vs Q</td>
</tr>
<tr>
<td>Times vs Q</td>
</tr>
<tr>
<td>Histogram</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 8.5.3.1-3  Parameters for Phase vs. Q tab</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Format</strong></td>
</tr>
<tr>
<td>Vth vs BER(Y = Log)</td>
</tr>
<tr>
<td>Vth vs BER(Y = Log(−Ln))</td>
</tr>
<tr>
<td>Vth vs Q</td>
</tr>
<tr>
<td>Phase vs Q</td>
</tr>
<tr>
<td>Phase vs σ</td>
</tr>
<tr>
<td>Phase vs μ</td>
</tr>
<tr>
<td>Phase vs Opt BER</td>
</tr>
<tr>
<td>Phase vs Opt Threshold</td>
</tr>
<tr>
<td>Phase vs Correlation</td>
</tr>
</tbody>
</table>
8.5 Automatic Measurement Commands

**Note:**
Settings vary, depending on the status of the installed options for each slot, and the data input interface settings.

<table>
<thead>
<tr>
<th>Function</th>
<th>Sets the vertical scale of the graph display during Q measurement. Input the minimum value instead of the step width, only when the vertical axis displays the error rate.</th>
</tr>
</thead>
</table>
| Example  | To set the vertical scale during Q Log display to maximum 10 dB, in 1-dB steps:  
> QSV 10,1  
Compatibility | Incompatible with existing models. |

**QSV?**

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;max&gt;,&lt;step&gt; = &lt;NR2 NUMERIC RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function</td>
<td>Queries the vertical scale of the graph display during Q measurement.</td>
</tr>
</tbody>
</table>
| Example  | To query the vertical scale during Q Log display:  
> QSV?  
< QSV 100,10  
Compatibility | Incompatible with existing models. |
QSH <max>,<step>

Parameter: <max>,<step> = <DECIMAL NUMERIC PROGRAM DATA>

Table 8.5.3.1-4  Parameters for Threshold vs. Q tab

<table>
<thead>
<tr>
<th>Format</th>
<th>&lt;max&gt;</th>
<th>&lt;step&gt;</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vth vs BER(Y = Log)</td>
<td>–3.990 to 4.000 (V), in 0.010 V steps</td>
<td>0.001 to 0.800 (V), in 0.001 V steps</td>
<td></td>
</tr>
<tr>
<td>Vth vs BER(Y = Log(-Ln))</td>
<td>–3.990 to 4.000 (V), in 0.010 V steps</td>
<td>0.001 to 0.800 (V), in 0.001 V steps</td>
<td></td>
</tr>
<tr>
<td>Vth vs Q</td>
<td>–3.990 to 4.000 (V), in 0.010 V steps</td>
<td>0.001 to 0.800 (V), in 0.001 V steps</td>
<td></td>
</tr>
<tr>
<td>Times vs Q</td>
<td>100 to 1000, in 50 steps</td>
<td>10 (fixed)</td>
<td></td>
</tr>
<tr>
<td>Histogram</td>
<td>0.16 to 1000.00, in 0.01 steps</td>
<td>16/32/64/128/256</td>
<td>When Linear &lt;max&gt; must be greater than &lt;step&gt;.</td>
</tr>
<tr>
<td></td>
<td>–49.84 to 60.00 (dB), in 0.01 steps</td>
<td>16/32/64</td>
<td>When Log</td>
</tr>
</tbody>
</table>

Table 8.5.3.1-5  Parameters for Phase vs. Q tab

<table>
<thead>
<tr>
<th>Format</th>
<th>&lt;max&gt;</th>
<th>&lt;step&gt;</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vth vs BER(Y = Log)</td>
<td>–3.990 to 4.000 (V), in 0.010 V steps</td>
<td>0.001 to 0.800 (V), in 0.001 V steps</td>
<td></td>
</tr>
<tr>
<td>Vth vs BER(Y = Log(-Ln))</td>
<td>–3.990 to 4.000 (V), in 0.010 V steps</td>
<td>0.001 to 0.800 (V), in 0.001 V steps</td>
<td></td>
</tr>
<tr>
<td>Vth vs Q</td>
<td>–3.990 to 4.000 (V), in 0.010 V steps</td>
<td>0.001 to 0.800 (V), in 0.001 V steps</td>
<td></td>
</tr>
<tr>
<td>Phase vs Q</td>
<td>–900 to 1000 (mUI), in 100 mUI steps</td>
<td>10 to 200 (mUI), in 10 mUI steps</td>
<td></td>
</tr>
<tr>
<td>Phase vs σ</td>
<td>–900 to 1000 (mUI), in 100 mUI steps</td>
<td>10 to 200 (mUI), in 10 mUI steps</td>
<td></td>
</tr>
<tr>
<td>Phase vs μ</td>
<td>–900 to 1000 (mUI), in 100 mUI steps</td>
<td>10 to 200 (mUI), in 10 mUI steps</td>
<td></td>
</tr>
<tr>
<td>Phase vs Opt BER</td>
<td>–900 to 1000 (mUI), in 100 mUI steps</td>
<td>10 to 200 (mUI), in 10 mUI steps</td>
<td></td>
</tr>
<tr>
<td>Phase vs Opt Threshold</td>
<td>–900 to 1000 (mUI), in 100 mUI steps</td>
<td>10 to 200 (mUI), in 10 mUI steps</td>
<td></td>
</tr>
<tr>
<td>Phase vs Opt Correlation</td>
<td>–900 to 1000 (mUI), in 100 mUI steps</td>
<td>10 to 200 (mUI), in 10 mUI steps</td>
<td></td>
</tr>
</tbody>
</table>

Note:
Settings vary, depending on the status of the installed options for each slot, and the data input interface settings.

Function: Sets the horizontal scale of the graph display during Q measurements.

Example: To set the horizontal scale during Vth display to maximum –1.000 V, in...
8.5 Automatic Measurement Commands

0.100-steps:
> QSH -1.000, 0.100

Compatibility
Incompatible with existing models.

QSH?

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;max&gt;,&lt;step&gt; = &lt;NR2 NUMERIC RESPONSE DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function</td>
<td>Queries the horizontal scale of the graph display during Q measurement.</td>
</tr>
<tr>
<td>Example</td>
<td>&gt; QSH?</td>
</tr>
<tr>
<td></td>
<td>&lt; QSH -1.000, 0.100</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Incompatible with existing models.</td>
</tr>
</tbody>
</table>

QTI <mode>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;mode&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Vth vs. BER (Y axis Log)</td>
</tr>
<tr>
<td>1</td>
<td>Vth vs. BER (Y axis Log (-Ln))</td>
</tr>
<tr>
<td>2</td>
<td>Vth vs. Q</td>
</tr>
<tr>
<td>3</td>
<td>Times vs. Q</td>
</tr>
<tr>
<td>4</td>
<td>Histogram</td>
</tr>
<tr>
<td>Function</td>
<td>Sets the item to be displayed in the graph in Vth vs. Q measurement.</td>
</tr>
<tr>
<td>Example</td>
<td>&gt; QTI 2</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Incompatible with existing models.</td>
</tr>
</tbody>
</table>

QTI?

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;mode&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt; (FIX1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function</td>
<td>Queries the item to be displayed in the graph in Vth vs. Q measurement.</td>
</tr>
<tr>
<td>Example</td>
<td>&gt; QTI?</td>
</tr>
<tr>
<td></td>
<td>&lt; QTI 2</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Incompatible with existing models.</td>
</tr>
</tbody>
</table>
### QMN <numeric>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;numeric&gt;</td>
<td>= &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
</tr>
<tr>
<td>1 to 1000</td>
<td>Measurement number (number of measurements during repeat measurement)</td>
</tr>
</tbody>
</table>

| Function | Sets the measurement number for which the Vth vs. Q measurement result is displayed in a graph. |

<table>
<thead>
<tr>
<th>Example</th>
<th>To set the measurement number for graph display to 2:</th>
</tr>
</thead>
<tbody>
<tr>
<td>&gt; QMN 2</td>
<td></td>
</tr>
</tbody>
</table>

| Compatibility | Incompatible with existing models. |

### QMN?

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;numeric&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt; (FIX4)</th>
</tr>
</thead>
</table>

| Function | Queries the measurement number for which the Vth vs. Q measurement result is displayed in a graph. |

<table>
<thead>
<tr>
<th>Example</th>
<th>&gt; QMN?</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt; QMN 2</td>
<td></td>
</tr>
<tr>
<td>&lt; QMN 1000</td>
<td></td>
</tr>
</tbody>
</table>

| Compatibility | Incompatible with existing models. |

### QPI <mode>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;mode&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Vth vs. BER (Y axis Log)</td>
</tr>
<tr>
<td>1</td>
<td>Vth vs. BER (Y axis Log (–Ln))</td>
</tr>
<tr>
<td>2</td>
<td>Vth vs. Q</td>
</tr>
<tr>
<td>3</td>
<td>Phase vs. Q</td>
</tr>
<tr>
<td>4</td>
<td>Phase vs. σ</td>
</tr>
<tr>
<td>5</td>
<td>Phase vs. μ</td>
</tr>
<tr>
<td>6</td>
<td>Phase vs. Optimum BER</td>
</tr>
<tr>
<td>7</td>
<td>Phase vs. Optimum Vth</td>
</tr>
<tr>
<td>8</td>
<td>Phase vs. Correlation coefficient</td>
</tr>
</tbody>
</table>

| Function | Sets the item to be displayed in the graph during Phase vs. Q measurement. |

<table>
<thead>
<tr>
<th>Example</th>
<th>To set the item to be displayed in the graph to Phase vs. Q:</th>
</tr>
</thead>
<tbody>
<tr>
<td>&gt; QPI 3</td>
<td></td>
</tr>
</tbody>
</table>

| Compatibility | Incompatible with existing models. |
8.5 Automatic Measurement Commands

**QPI?**

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;mode&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt; (FIX1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example</td>
<td>Queries the item to be displayed in the graph during Phase vs. Q measurement.</td>
</tr>
<tr>
<td></td>
<td>&gt; QPI?</td>
</tr>
<tr>
<td></td>
<td>&lt; QPI 3</td>
</tr>
</tbody>
</table>

**Compatibility**

Incompatible with existing models.

**QPM <numeric>**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;numeric&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 to 1000</td>
<td>Measurement number (number of measurements during repeat measurement)</td>
</tr>
</tbody>
</table>

| Function | Sets the measurement number for which the Phase vs. Q measurement result is displayed in a graph. |
| Example  | To set the measurement number graph display to 2: |
|          | > QPM 2 |

**Compatibility**

Incompatible with existing models.

**QPM?**

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;numeric&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt; (FIX4)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function</td>
<td>Queries the measurement number for which the Phase vs. Q measurement result is displayed in a graph.</td>
</tr>
<tr>
<td>Example</td>
<td>&gt; QPM?</td>
</tr>
<tr>
<td></td>
<td>&lt; QPM 2</td>
</tr>
<tr>
<td></td>
<td>&lt; QPM 1000</td>
</tr>
</tbody>
</table>

**Compatibility**

Incompatible with existing models.
8.5.3.2  Vth vs. Q measurement commands

![Vth vs. Q measurement setting field]

Figure 8.5.3.2-1  Vth vs. Q measurement setting field
### 8.5 Automatic Measurement Commands

#### Table 8.5.3.2-1 Vth vs. Q measurement setting commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1]</td>
<td>Start</td>
<td>QTT</td>
</tr>
<tr>
<td>[2]</td>
<td>Stop</td>
<td>QTP</td>
</tr>
<tr>
<td>[3]</td>
<td>Query for measurement progress state</td>
<td>QAS?</td>
</tr>
<tr>
<td>[4]</td>
<td>Slot</td>
<td>QSL, QSL?</td>
</tr>
<tr>
<td>[6]</td>
<td>Time</td>
<td>QTM, QTM?</td>
</tr>
<tr>
<td>[7]</td>
<td>Interval</td>
<td>QTN, QTN?</td>
</tr>
<tr>
<td>[8]</td>
<td>Delay</td>
<td>QTD, QTD?</td>
</tr>
<tr>
<td>[9]</td>
<td>Fine/Coarse</td>
<td>QTR, QTR?</td>
</tr>
<tr>
<td>[10]</td>
<td>Error Threshold</td>
<td>QTE, QTE?</td>
</tr>
<tr>
<td>[12]</td>
<td>Auto Search</td>
<td>QTA, QTA?</td>
</tr>
<tr>
<td>[13]</td>
<td>Correlation Filter</td>
<td>QTF, QTF?</td>
</tr>
<tr>
<td>[14]</td>
<td>Current</td>
<td>QTU, QTU?</td>
</tr>
</tbody>
</table>

**QTT**

- **Function**: Starts Vth vs. Q measurement.
- **Example**: `> QTT`
- **Compatibility**: Incompatible with existing models.

**QTP**

- **Function**: Stops Vth vs. Q measurement.
- **Example**:
  - To stop measurement: `> QTP`
- **Compatibility**: Incompatible with existing models.
Chapter 8  Native Commands

QAS?
Response  \(<\text{numeric}> = \langle\text{NR1 NUMERIC RESPONSE DATA}\rangle\) (FIX1)
0   Measurement has been stopped.
1   During measurement
Function Queries the Vth vs Q measurement processing state.
Example
> QAS?
< QAS 1
Compatibility Incompatible with existing models.

QSL <numeric>[,<unit>]
Parameter  \(<\text{numeric}> = \langle\text{DECIMAL NUMERIC PROGRAM DATA}\rangle\)
          1 to 6   Slot No. 1 to 6
          1 to 6 when using the MP1800A, 1 to 4 when using the MT1810A
          [<unit>] = \langle\text{DECIMAL NUMERIC PROGRAM DATA}\rangle
          1 to 4   Mainframe No. 1 to 4
          Specify the mainframe No. when connected to the MT1810A.
          [<unit>] is optional. When omitted, 1 is specified.
Function Sets the Vth vs. Q measurement target slot.
Example
To set the target slot to Slot 2:
> QSL 2
Compatibility Incompatible with existing models.

QSL? [<unit>]
Parameter  [<unit>] = \langle\text{DECIMAL NUMERIC PROGRAM DATA}\rangle
          1 to 4   Mainframe No. 1 to 4
          Specify the mainframe No. when connected to the MT1810A.
          [<unit>] is optional. When omitted, 1 is specified.
Response  \(<\text{numeric}> = \langle\text{NR1 NUMERIC RESPONSE DATA}\rangle\) (FIX2)
Function Queries the Vth vs. Q measurement target slot.
Example
> QSL?
< QSL 2
Compatibility Incompatible with existing models.
8.5 Automatic Measurement Commands

QTO <mode>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;mode&gt; = DECIMAL NUMERIC PROGRAM DATA</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Single</td>
</tr>
<tr>
<td>1</td>
<td>Repeat</td>
</tr>
<tr>
<td>2</td>
<td>Untimed</td>
</tr>
</tbody>
</table>

Function: Sets the measurement processing mode during Vth vs. Q measurement.

Example: To set the measurement processing mode to Single mode:
> QTO 0

Compatibility: Incompatible with existing models.

QTO?

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;mode&gt; = NR1 NUMERIC RESPONSE DATA (FIX1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function</td>
<td>Queries the measurement processing mode during Vth vs. Q measurement.</td>
</tr>
</tbody>
</table>

Example: > QTO?
< QTO 0

Compatibility: Incompatible with existing models.

QTM <numeric>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;numeric&gt; = DECIMAL NUMERIC PROGRAM DATA</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2 to 1000 2 to 1000 times, in 1-time steps</td>
</tr>
</tbody>
</table>

Function: Sets the number of measurements during Vth vs. Q measurement.

Example: To set the number of measurements to 100 times:
> QTM 100

Compatibility: Incompatible with existing models.

QTM?

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;NUMERIC&gt; = NR1 NUMERIC RESPONSE DATA (FIX4)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function</td>
<td>Queries the number of measurements during Vth vs. Q measurement.</td>
</tr>
</tbody>
</table>

Example: > QTM?
< QTM 2
< QTM 1000

Compatibility: Incompatible with existing models.
Chapter 8  Native Commands

**QTN <numeric>**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;numeric&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0 to 9999</td>
</tr>
<tr>
<td>Function</td>
<td>Sets the measurement interval time during Vth vs. Q measurement.</td>
</tr>
<tr>
<td>Example</td>
<td>To set the measurement interval time to 50 s:</td>
</tr>
<tr>
<td></td>
<td>&gt; QTN 50</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Incompatible with existing models.</td>
</tr>
</tbody>
</table>

**QTN?**

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;NUMERIC&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt; (FIX4)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function</td>
<td>Queries the measurement interval time during Vth vs. Q measurement.</td>
</tr>
<tr>
<td>Example</td>
<td>&gt; QTN?</td>
</tr>
<tr>
<td></td>
<td>&lt; QTN 50</td>
</tr>
<tr>
<td></td>
<td>&lt; QTN 9999</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Incompatible with existing models.</td>
</tr>
</tbody>
</table>

**QTD <numeric>**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;numeric&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>–1000 to 1000</td>
</tr>
<tr>
<td>Function</td>
<td>Sets the measurement start delay position during Vth vs. Q measurement.</td>
</tr>
<tr>
<td>Example</td>
<td>To set the measurement start delay position to –100 mUI:</td>
</tr>
<tr>
<td></td>
<td>&gt; QTD –100</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Incompatible with existing models.</td>
</tr>
</tbody>
</table>

**QTD?**

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;numeric&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt; (FIX5)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function</td>
<td>Queries the measurement start delay position during Vth vs. Q measurement.</td>
</tr>
<tr>
<td>Example</td>
<td>&gt; QTD?</td>
</tr>
<tr>
<td></td>
<td>&lt; QTD –1000</td>
</tr>
<tr>
<td></td>
<td>&lt; QTD –100</td>
</tr>
<tr>
<td></td>
<td>&lt; QTD 1000</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Incompatible with existing models.</td>
</tr>
</tbody>
</table>
### 8.5 Automatic Measurement Commands

#### QTR <mode>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;mode&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>Fine mode</td>
</tr>
<tr>
<td>1</td>
<td>Coarse mode</td>
</tr>
</tbody>
</table>

**Function**: Sets the measurement resolution (Fine/Coarse mode) during Vth vs. Q measurement.

**Example**: To set the measurement resolution to Fine mode:

```plaintext
> QTR 0
```

**Compatibility**: Incompatible with existing models.

#### QTR? [Response: <mode> = <NR1 NUMERIC RESPONSE DATA> (FIX1)]

**Function**: Queries the measurement resolution (Fine/Coarse mode) during Vth vs. Q measurement.

**Example**: To query the measurement resolution:

```plaintext
> QTR?
< QTR 0
```

**Compatibility**: Incompatible with existing models.

#### QTE <upper>,<lower>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;upper&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
<td></td>
</tr>
<tr>
<td>3 to 5</td>
<td>E–3 to E–5, in single steps</td>
</tr>
<tr>
<td>&lt;lower&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
<td></td>
</tr>
<tr>
<td>7 to 12</td>
<td>E–7 to E–12, in single steps</td>
</tr>
</tbody>
</table>

**Function**: Sets the measurement error threshold range during Vth vs. Q measurement.

**Example**: To set the measurement error threshold range to E–3 to E–12:

```plaintext
> QTE 3, 12
```

**Compatibility**: Incompatible with existing models.

#### QTE? [Response: <upper> = <DECIMAL NUMERIC RESPONSE DATA> (FIX1), <lower> = <DECIMAL NUMERIC RESPONSE DATA> (FIX2)]

**Function**: Queries the measurement error threshold range during Vth vs. Q measurement.

**Example**: To query the measurement error threshold range:

```plaintext
> QTE?
< QTE 3, 7
< QTE 5, 12
```

**Compatibility**: Incompatible with existing models.
**QTC <upper>,<lower>**

**Parameter**
- `<upper>` = `<DECIMAL NUMERIC PROGRAM DATA>`
  - 3 to 5: E–3 to E–5, in single steps
- `<lower>` = `<DECIMAL NUMERIC PROGRAM DATA>`
  - 7 to 12: E–7 to E–12, in single steps

**Function**
Sets the error threshold range during Vth vs. Q measurement recalculation.

**Example**
To set the error threshold range during recalculation from E–5 to E–10:
> QTC 5,10

**Compatibility**
Incompatible with existing models.

**QTC?**

**Response**
- `<upper>` = `<DECIMAL NUMERIC RESPONSE DATA>` (FIX1)
- `<lower>` = `<DECIMAL NUMERIC RESPONSE DATA>` (FIX2)

**Function**
Queries the error threshold range during Vth vs. Q measurement recalculation.

**Example**
- > QTC?
- < QTC 3, 7
- < QTC 5,12

**Compatibility**
Incompatible with existing models.

**QTA <numeric>**

**Parameter**
- `<numeric>` = `<DECIMAL NUMERIC PROGRAM DATA>`
  - 0: Auto search OFF
  - 1: Threshold auto search (Coarse)
  - 2: Phase & Threshold auto search (Coarse)
  - 3: Threshold auto search (Fine)
  - 4: Phase & Threshold auto search (Fine)

**Function**
Sets whether to execute auto search during Vth vs. Q measurement.

**Example**
To set to the threshold auto search to ON:
> QTA 1

**Compatibility**
Incompatible with existing models.

**QTA?**

**Response**
- `<numeric>` = `<NR1 NUMERIC RESPONSE DATA>` (FIX1)

**Function**
Queries whether to execute auto search during Vth vs. Q measurement.

**Example**
- > QTA?
- < QTA 1

**Compatibility**
Incompatible with existing models.
8.5 Automatic Measurement Commands

**QTF [<numeric1>], [<numeric2>]**

**Parameter**

- `<numeric1>` = `<DECIMAL NUMERIC PROGRAM DATA>`
  - 0: OFF
  - 1: ON
- `[<numeric2>]` = `<DECIMAL NUMERIC PROGRAM DATA>`
  - 80 to 100: 80 to 100%, in 1% steps
  - Omitted when `<numeric1>` is 0.

**Function**

Sets the minimum correlation coefficient with which the Q value is recognized to be valid during Vth vs. Q measurement.

**Example**

To set the minimum correlation coefficient to 80%:

> QTF 1, 80
> QTF 0

**Compatibility**

Incompatible with existing models.

**QTF?**

**Response**

- `<numeric1>` = `<NR2 NUMERIC RESPONSE DATA>` (FIX1)
- `<numeric2>` = `<NR2 NUMERIC RESPONSE DATA>` (FIX3)

**Function**

Queries the minimum correlation coefficient with which the Q value is recognized to be valid during Vth vs. Q measurement.

**Example**

> QTF?
  < QTF 1, 80
  < QTF 1, 100

**Compatibility**

Incompatible with existing models.

**QTU <numeric>**

**Parameter**

- `<numeric>` = `<DECIMAL NUMERIC PROGRAM DATA>`
  - 0: OFF: Updates at measurement termination
  - 1: ON: Updates every second

**Function**

Sets the graph display update period during Vth vs. Q measurement.

**Example**

To set the graph display update timing to every second:

> QTU 1

**Compatibility**

Incompatible with existing models.

**QTU?**

**Response**

- `<numeric>` = `<NR1 NUMERIC RESPONSE DATA>` (FIX1)

**Function**

Queries the graph display update period during Vth vs. Q measurement.

**Example**

> QTU?
  < QTU 1

**Compatibility**

Incompatible with existing models.
8.5.3.3 Phase vs. Q measurement commands

Figure 8.5.3.3-1 Phase vs. Q measurement setting field
### 8.5 Automatic Measurement Commands

#### Table 8.5.3.3-1 Phase vs. Q measurement setting commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1]</td>
<td>Start QPT</td>
<td>QPT</td>
</tr>
<tr>
<td>[2]</td>
<td>Stop QPP</td>
<td>QPP</td>
</tr>
<tr>
<td>[3]</td>
<td>Query for measurement progress state QAS?</td>
<td>QAS?</td>
</tr>
<tr>
<td>[4]</td>
<td>Slot QSP</td>
<td>QSP</td>
</tr>
<tr>
<td></td>
<td>QSP?</td>
<td></td>
</tr>
<tr>
<td></td>
<td>QMR?</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Start QPO</td>
<td>QPO</td>
</tr>
<tr>
<td></td>
<td>QPO?</td>
<td></td>
</tr>
<tr>
<td></td>
<td>End QPO</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Center Span QPW</td>
<td>QPW</td>
</tr>
<tr>
<td></td>
<td>QPW?</td>
<td></td>
</tr>
<tr>
<td></td>
<td>QPR?</td>
<td></td>
</tr>
<tr>
<td>[7]</td>
<td>Error Threshold QPE</td>
<td>QPE</td>
</tr>
<tr>
<td></td>
<td>QPE?</td>
<td></td>
</tr>
<tr>
<td>[8]</td>
<td>Calculation Threshold QPC</td>
<td>QPC</td>
</tr>
<tr>
<td></td>
<td>QPC?</td>
<td></td>
</tr>
<tr>
<td>[9]</td>
<td>Auto Search QPA</td>
<td>QPA</td>
</tr>
<tr>
<td></td>
<td>QPA?</td>
<td></td>
</tr>
<tr>
<td>[10]</td>
<td>Correlation Filter QPF</td>
<td>QPF</td>
</tr>
<tr>
<td></td>
<td>QPF?</td>
<td></td>
</tr>
<tr>
<td></td>
<td>QPU?</td>
<td></td>
</tr>
</tbody>
</table>

**QPT**

Function: Starts Phase vs. Q measurement.
Example: `> QPT`
Compatibility: Incompatible with existing models.

**QPP**

Function: Stops Phase vs. Q measurement.
Example: `> QPP`
Compatibility: Incompatible with existing models.
Chapter 8  Native Commands

QAS?

Response

<numeric> = <NR1 NUMERIC RESPONSE DATA> (FIX1)

0  Measurement has been stopped.
1  During measurement

Function

Queries the Phase vs Q measurement processing state.

Example

> QAS?
< QAS 1

Compatibility

Incompatible with existing models.

QSP <numeric>[,<unit>]

Parameter

<numeric> = <DECIMAL NUMERIC PROGRAM DATA>
1 to 6  Slot No. 1 to 6
1 to 6 when using the MP1800A, 1 to 4 when using the MT1810A
[<unit>] = <DECIMAL NUMERIC PROGRAM DATA>
1 to 4  Mainframe No. 1 to 4
Specify the mainframe No. when connected to the MT1810A.
<br/>&lt;unit&gt; is optional. When omitted, 1 is specified.

Function

Sets the Phase vs. Q measurement target slot.

Example

To set the target slot to Slot 2:

> QSP 2

Compatibility

Incompatible with existing models.

QSP? [<unit>]

Parameter

[<unit>] = <DECIMAL NUMERIC PROGRAM DATA>
1 to 4  Mainframe No. 1 to 4
Specify the unit No. when connected to the MT1810A.
<br/>&lt;unit&gt; is optional. When omitted, 1 is specified.

Response

<numeric> = <NR1 NUMERIC RESPONSE DATA> (FIX2)

Function

Queries the Phase vs. Q measurement target slot.

Example

> QSP?
< QSP 2
< QSP 64

Compatibility

Incompatible with existing models.
8.5 Automatic Measurement Commands

**QMR <mode>**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;mode&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>Range: Specifies the start position and end positions of the measurement range.</td>
</tr>
<tr>
<td>1</td>
<td>Width: Specifies the center position and width of the measurement range.</td>
</tr>
</tbody>
</table>

**Function**

Selects the method to specify the Phase vs. Q measurement range.

**Example**

To set the method to specify the measurement range to Width:

```plaintext
> QMR 1
```

**Compatibility**

Incompatible with existing models.

**QMR?**

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;mode&gt; = &lt; NR1 NUMERIC RESPONSE DATA&gt; (FIX1)</th>
</tr>
</thead>
</table>

**Function**

Queries the method to specify the Phase vs. Q measurement range.

**Example**

To set the method to specify the Phase vs. Q measurement range:

```plaintext
> QMR?
```

```plaintext
< QMR 1
```

**Compatibility**

Incompatible with existing models.

**QPO <start>,<end>,<step>**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;start&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
<td></td>
</tr>
<tr>
<td>–1000 to 1000 –1000 to 1000 mUI, in 1-mUI steps</td>
<td></td>
</tr>
<tr>
<td>&lt;end&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
<td></td>
</tr>
<tr>
<td>–1000 to 1000 –1000 to 1000 mUI, in 1-mUI steps</td>
<td></td>
</tr>
<tr>
<td>&lt;step&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
<td></td>
</tr>
<tr>
<td>1 to 200 1 to 200 mUI, in 1-mUI steps</td>
<td></td>
</tr>
</tbody>
</table>

**Function**

Sets the measurement range and measurement step during Phase vs. Q measurement.

**Example**

To set the measurement range from –200 to 300 mUI, in 10-mUI steps:

```plaintext
> QPO -200,300,10
```

**Compatibility**

Incompatible with existing models.
QPO?

Response
<start> = <NR1 NUMERIC RESPONSE DATA> (FIX5)
<end> = <NR1 NUMERIC RESPONSE DATA> (FIX5)
<step> = <NR1 NUMERIC RESPONSE DATA> (FIX3)

Function
Queries the measurement range and measurement step during Phase vs. Q measurement.

Example
> QPO?
< QPO -1000, -999, 1
< QPO 999, 1000, 1

Compatibility
Incompatible with existing models.

QPW <center>,<span>,<step>

Parameter
<center> = <DECIMAL NUMERIC PROGRAM DATA>
−1000 to 1000 −1000 to 1000 mUI, in 1-mUI steps
<span> = <DECIMAL NUMERIC PROGRAM DATA>
0 to 2000 0 to 2000 mUI, in 1-mUI steps
<step> = <DECIMAL NUMERIC PROGRAM DATA>
1 to 200 1 to 200 mUI, in 1-mUI steps

Function
Sets the center of the Phase vs Q measurement, measurement range, and measurement steps.

Example
To set the measurement range to 200 mUI, centered at 100 mUI, in 10-mUI steps:
> QPW 100,200,10

Compatibility
Incompatible with existing models.

QPW?

Response
<center> = <NR1 NUMERIC RESPONSE DATA> (FIX5)
<span> = <NR1 NUMERIC RESPONSE DATA> (FIX4)
<step> = <NR1 NUMERIC RESPONSE DATA> (FIX3)

Function
Queries the center measurement range and measurement step during Phase vs. Q measurement.

Example
> QPW?
< QPW -1000, 0, 1
< QPW 999, 1000, 1

Compatibility
Incompatible with existing models.
8.5 Automatic Measurement Commands

### QPR <mode>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;mode&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Fine mode</td>
</tr>
<tr>
<td>1</td>
<td>Coarse mode</td>
</tr>
</tbody>
</table>

**Function**
Sets the measurement resolution (Fine/Coarse mode) during Phase vs. Q measurement.

**Example**
To set the measurement resolution to Fine mode:
> QPR 0

**Compatibility**
Incompatible with existing models.

### QPR? 

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;mode&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt; (FIX1)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Function</strong></td>
<td>Queries the measurement resolution (Fine/Coarse mode) during Phase vs. Q measurement.</td>
</tr>
<tr>
<td><strong>Example</strong></td>
<td>&gt; QPR?</td>
</tr>
<tr>
<td></td>
<td>&lt; QPR 0</td>
</tr>
</tbody>
</table>

**Compatibility**
Incompatible with existing models.

### QPE <upper>,<lower>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>&lt;upper&gt; = &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>3 to 5</td>
<td>E–3 to E–5, in single steps</td>
</tr>
<tr>
<td>&lt;lower&gt;</td>
<td>&lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
</tr>
<tr>
<td>7 to 12</td>
<td>E–7 to E–12, in single steps</td>
</tr>
</tbody>
</table>

**Function**
Sets the measurement error threshold range during Phase vs. Q measurement.

**Example**
To set the measurement error threshold range to E–3 to E–12:
> QPE 3,12

**Compatibility**
Incompatible with existing models.

### QPE? 

<table>
<thead>
<tr>
<th>Response</th>
<th>&lt;upper&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt; (FIX1)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>&lt;lower&gt; = &lt;NR1 NUMERIC RESPONSE DATA&gt; (FIX2)</td>
</tr>
<tr>
<td><strong>Function</strong></td>
<td>Queries the measurement error threshold range during Phase vs. Q measurement.</td>
</tr>
<tr>
<td><strong>Example</strong></td>
<td>&gt; QPE?</td>
</tr>
<tr>
<td></td>
<td>&lt; QPE 3, 7</td>
</tr>
<tr>
<td></td>
<td>&lt; QPE 5,12</td>
</tr>
</tbody>
</table>

**Compatibility**
Incompatible with existing models.
**QPC \(<upper>,<lower>\)**

**Parameter**  
- \(<upper>\) = \(<\text{DECIMAL NUMERIC PROGRAM DATA}>\)  
  3 to 5 \(\text{E}^{-3}\) to \(\text{E}^{-5}\), in single steps
- \(<lower>\) = \(<\text{DECIMAL NUMERIC PROGRAM DATA}>\)  
  7 to 12 \(\text{E}^{-7}\) to \(\text{E}^{-12}\), in single steps

**Function**  
Sets the error threshold range during Phase vs. Q measurement recalculation.

**Example**  
To set the error threshold range during recalculation to \(\text{E}^{-5}\) to \(\text{E}^{-10}\):

\[> \text{QPC } 5,10\]

**Compatibility**  
Incompatible with existing models.

---

**QPC?**

**Response**  
- \(<upper>\) = \(<\text{NR1 NUMERIC RESPONSE DATA}>\) (FIX1)
- \(<lower>\) = \(<\text{NR1 NUMERIC RESPONSE DATA}>\) (FIX2)

**Function**  
Queries the error threshold range during Phase vs. Q measurement recalculation.

**Example**  
\[> \text{QPC}\]
\[< \text{QPC } 3,7\]
\[< \text{QPC } 5,10\]

**Compatibility**  
Incompatible with existing models.

---

**QPA \(<\text{numeric}>\)**

**Parameter**  
- \(<\text{numeric}>\) = \(<\text{DECIMAL NUMERIC PROGRAM DATA}>\)  
  0  \(\text{Auto Search OFF}\)
  1  \(\text{Threshold auto search (Coarse)}\)
  2  \(\text{Threshold auto search (Fine)}\)

**Function**  
Sets whether to execute auto search during Phase vs. Q measurement.

**Example**  
To set the threshold auto search to ON:

\[> \text{QPA } 1\]

**Compatibility**  
Incompatible with existing models.

---

**QPA?**

**Response**  
- \(<\text{numeric}>\) = \(<\text{NR1 NUMERIC RESPONSE DATA}>\) (FIX1)

**Function**  
Queries whether to execute auto search during Phase vs. Q measurement.

**Example**  
\[> \text{QPA}\]
\[< \text{QPA } 1\]

**Compatibility**  
Incompatible with existing models.
8.5 Automatic Measurement Commands

**QPF <numeric1>[,<numeric2>]**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;numeric1&gt;</td>
<td>= &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
</tr>
<tr>
<td>0</td>
<td>OFF</td>
</tr>
<tr>
<td>1</td>
<td>ON</td>
</tr>
<tr>
<td>&lt;numeric2&gt;</td>
<td>= &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
</tr>
<tr>
<td>80 to 100</td>
<td>80 to 100%, in 1% steps</td>
</tr>
</tbody>
</table>

Omitted when <numeric1> is 0.

- **Function**: Sets the minimum correlation coefficient with which the Q value is recognized to be valid during Phase vs. Q measurement.

- **Example**: To set the minimum correlation coefficient to 90%:
  
  > QPF 1,90

  To set the minimum correlation coefficient to OFF:

  > QPF 0

- **Compatibility**: Incompatible with existing models.

**QPF?**

<table>
<thead>
<tr>
<th>Response</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;numeric1&gt;</td>
<td>= &lt;NR2 NUMERIC RESPONSE DATA&gt; (FIX1)</td>
</tr>
<tr>
<td>&lt;numeric2&gt;</td>
<td>= &lt;NR2 NUMERIC RESPONSE DATA&gt; (FIX3)</td>
</tr>
</tbody>
</table>

- **Function**: Queries the minimum correlation coefficient with which the Q value is recognized to be valid during Phase vs. Q measurement.

- **Example**:  
  
  > QPF?  
  < QPF 1, 80  
  < QPF 1,100

- **Compatibility**: Incompatible with existing models.

**QPU <numeric>**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;numeric&gt;</td>
<td>= &lt;DECIMAL NUMERIC PROGRAM DATA&gt;</td>
</tr>
<tr>
<td>0</td>
<td>OFF: Updates at measurement termination</td>
</tr>
<tr>
<td>1</td>
<td>ON: Updates every second</td>
</tr>
</tbody>
</table>

- **Function**: Sets the graph display during Phase vs. Q measurement.

- **Example**: To set the graph display update timing to every second:

  > QPU 1

- **Compatibility**: Incompatible with existing models.
## QPU?

<table>
<thead>
<tr>
<th>Response</th>
<th>(&lt;\text{numeric}&gt; = \langle\text{NR1 NUMERIC RESPONSE DATA}\rangle ) (FIX1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function</td>
<td>Queries the graph display update period during Phase vs. Q measurement.</td>
</tr>
<tr>
<td>Example</td>
<td>(&gt; \text{QPU}?)</td>
</tr>
<tr>
<td></td>
<td>(&lt; \text{QPU 1})</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Incompatible with existing models.</td>
</tr>
</tbody>
</table>
8.5 Automatic Measurement Commands

8.5.3.4 Measurement results and status query commands

![Image 8.5.3.4-1 Measurement results and status display field]

Table 8.5.3.4-1 Measurement results and status query commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1]</td>
<td>No label (Setting of time display)</td>
<td>QAT</td>
</tr>
<tr>
<td></td>
<td></td>
<td>QAT?</td>
</tr>
<tr>
<td>[2]</td>
<td>Query for current time</td>
<td>QSO?</td>
</tr>
<tr>
<td>[3]</td>
<td>Query for measurement start time</td>
<td>QST?</td>
</tr>
<tr>
<td>[4]</td>
<td>Query for measurement elapsed time</td>
<td>QEL?</td>
</tr>
<tr>
<td>[5]</td>
<td>Query for Vth vs. Q measurement result</td>
<td>QTH?</td>
</tr>
<tr>
<td>[6]</td>
<td>Query for Vth vs. Q measurement status</td>
<td>QTW?</td>
</tr>
<tr>
<td>[7]</td>
<td>Query for Phase vs. Q measurement result</td>
<td>QPH?</td>
</tr>
<tr>
<td>[8]</td>
<td>Query for Phase vs. Q measurement status</td>
<td>QSW?</td>
</tr>
</tbody>
</table>

**QAT <type>**

Parameter

\(<type> = \text{DECIMAL NUMERIC PROGRAM DATA}\>

0       Displays the current date and time.
1       Displays the measurement start time.
2       Displays the elapsed time based on the measurement period.

Function

Selects the Q measurement time display type.

Example

To set the Q measurement time display type to measurement start time (Start Time):

> QAT 1

Compatibility

Incompatible with existing models.
Chapter 8  Native Commands

QAT?
Response  
<type> = <NR1 NUMERIC RESPONSE DATA> (FIX1)

Function  
Queries the Q measurement time display type.

Example  
> QAT?
< QAT 1

Compatibility  
Incompatible with existing models.

QSO?
Response  
<year> = <NR1 NUMERIC RESPONSE DATA> (FIX4)  
2000 to 2036  Year 2000 to 2036
<month> = <NR1 NUMERIC RESPONSE DATA> (FIX2)  
01 to 12  January to December
<day> = <NR1 NUMERIC RESPONSE DATA> (FIX2)  
01 to 31  1st to 31st
<hour> = <NR1 NUMERIC RESPONSE DATA> (FIX2)  
00 to 23  0 to 23 hours
<minute> = <NR1 NUMERIC RESPONSE DATA> (FIX2)  
00 to 59  0 to 59 minutes
<second> = <NR1 NUMERIC RESPONSE DATA> (FIX2)  
00 to 59  0 to 59 seconds

Function  
Queries the current date and time during Q measurement.

Example  
> QSO?
< QSO 2005,12,31,23,59,59

Compatibility  
Incompatible with existing models.
8.5 Automatic Measurement Commands

QST?

Response

\[
\text{<year>} = \text{<NR1 NUMERIC RESPONSE DATA>} \text{ (FIX4)}
\]
0000, 2000 to 2036 Year 2000 to 2036
\[
\text{<month>} = \text{<NR1 NUMERIC RESPONSE DATA>} \text{ (FIX2)}
\]
00, 01 to 12 January to December
\[
\text{<day>} = \text{<NR1 NUMERIC RESPONSE DATA>} \text{ (FIX2)}
\]
00, 01 to 31 1st to 31st
\[
\text{<hour>} = \text{<NR1 NUMERIC RESPONSE DATA>} \text{ (FIX2)}
\]
00 to 23 0 to 23 hours
\[
\text{<minute>} = \text{<NR1 NUMERIC RESPONSE DATA>} \text{ (FIX2)}
\]
00 to 59 0 to 59 minutes
\[
\text{<second>} = \text{<NR1 NUMERIC RESPONSE DATA>} \text{ (FIX2)}
\]
00 to 59 0 to 59 seconds

Note:
The following is output when no measurement is performed:
QST 0000,00,00,00,00

Function
Queries the measurement start time (Start Time) during Q measurement.

Example

\[
> \text{QST?}
\]
\[
< \text{QST 2005,12,31,23,59,59}
\]

Compatibility
Incompatible with existing models.

QEL?

Response

\[
\text{<day>} = \text{<NR1 NUMERIC RESPONSE DATA>} \text{ (FIX2)}
\]
00 to 99 0 to 99 days
\[
\text{<hour>} = \text{<NR1 NUMERIC RESPONSE DATA>} \text{ (FIX2)}
\]
00 to 23 0 to 23 hours
\[
\text{<minute>} = \text{<NR1 NUMERIC RESPONSE DATA>} \text{ (FIX2)}
\]
00 to 59 0 to 59 minutes
\[
\text{<second>} = \text{<NR1 NUMERIC RESPONSE DATA>} \text{ (FIX2)}
\]
00 to 59 0 to 59 seconds

Note:
The following is output when no measurement is performed:
QEL 00,00,00

Function
Queries the measurement elapsed time (Elapsed time) during Q measurement.

Example

\[
> \text{QEL?}
\]
\[
< \text{QEL 31,23,59,59}
\]

Compatibility
Incompatible with existing models.
**QTH? <numeric1>[,<numeric2>]**

Parameter:  
<numeric1>,<numeric2> = <DECIMAL NUMERIC PROGRAM DATA>  
<numeric1>,<numeric2>For the contents of measurement data <result1>, see Table 8.5.3.4-2.  
<numeric1>,<numeric2>For the contents of immediate data <result2>, see Table 8.5.3.4-3.  
<numeric1>For the contents of statistic data <result3>, see Table 8.5.3.4-4.

<table>
<thead>
<tr>
<th>Items</th>
<th>&lt;numeric1&gt;,&lt;numeric2&gt;</th>
<th>Response Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>Top side</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Vth/Error Rate</td>
<td>Number of measurements</td>
<td>0,1,...0,1000</td>
</tr>
<tr>
<td></td>
<td>1 to 1000</td>
<td>Form1</td>
</tr>
<tr>
<td>Bottom side</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Vth/Error Rate</td>
<td></td>
<td>1,1,...1,1000</td>
</tr>
<tr>
<td></td>
<td>1 to 1000</td>
<td>Form1</td>
</tr>
</tbody>
</table>

The measurement result for each Vth is delimited with a comma (,) and returned.

<table>
<thead>
<tr>
<th>Items</th>
<th>&lt;numeric1&gt;,&lt;numeric2&gt;</th>
<th>Response Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q value</td>
<td>2,1,...2,1000</td>
<td>Form2</td>
</tr>
<tr>
<td>Optimum BER</td>
<td>3,1,...3,1000</td>
<td>Form3</td>
</tr>
<tr>
<td>Optimum Vth</td>
<td>4,1,...4,1000</td>
<td>Form4</td>
</tr>
<tr>
<td>Correlation bottom</td>
<td>5,1,...5,1000</td>
<td>Form2</td>
</tr>
<tr>
<td>Correlation top</td>
<td>6,1,...6,1000</td>
<td>Form2</td>
</tr>
<tr>
<td>σ bottom</td>
<td>7,1,...7,1000</td>
<td>Form2</td>
</tr>
<tr>
<td>σ top</td>
<td>8,1,...8,1000</td>
<td>Form2</td>
</tr>
<tr>
<td>μ bottom</td>
<td>9,1,...9,1000</td>
<td>Form4</td>
</tr>
<tr>
<td>μ top</td>
<td>10,1,...10,1000</td>
<td>Form4</td>
</tr>
<tr>
<td>Q Equation</td>
<td>11,1,...11,1000</td>
<td>Form2</td>
</tr>
</tbody>
</table>
### 8.5 Automatic Measurement Commands

#### Table 8.5.3.4-4  Vth vs. Q measurement statistic data

<table>
<thead>
<tr>
<th>Items</th>
<th>&lt;numeric1&gt;</th>
<th>Response Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Data</td>
<td>12</td>
<td>Form5</td>
</tr>
<tr>
<td>Valid Data</td>
<td>13</td>
<td>Form5</td>
</tr>
<tr>
<td>Q Max</td>
<td>14</td>
<td>Form2</td>
</tr>
<tr>
<td>Q Min</td>
<td>15</td>
<td>Form2</td>
</tr>
<tr>
<td>Q Mean</td>
<td>16</td>
<td>Form2</td>
</tr>
<tr>
<td>Q σ</td>
<td>17</td>
<td>Form2</td>
</tr>
<tr>
<td>Q-5 σ</td>
<td>18</td>
<td>Form2</td>
</tr>
</tbody>
</table>

<numeric2> is omitted for statistic data.

#### Response

Response format:

<table>
<thead>
<tr>
<th>Form</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
</table>
| Form1 Vth-Q type | XX.XXX,Y.YYYYE-YYY | XX.XXX: Threshold (V)  
                      Y.YYYYE-YYY: Error Rate  
                      -----,-------: When no data corresponds to the query. |
| Form2 % type (dB type) | XXX.XX | When 0.00 to 100.00  
                      -----: When no data corresponds to the query. |
| Form3 Fraction type | X.XXXXE-XXX | When 0.0000E–016 to 1.0000E–000  
                      -----: When no data corresponds to the query.  
                      < 1.0E-199: When E–199 or less |
| Form4 Voltage type | XX.XXXX | When –4.0000 to 4.0000  
                      -----: When no data corresponds to the query. |
| Form5 Integer type | XXXX | When 0 to 9999  
                      ----: When no data corresponds to the query. |

**Function**

Queries the Vth vs. Q measurement result.

**Example**

To query the measurement result of the 10th time, on the top side:

> QTH? 0,10

< QTH

1.000,1.2345E–003,1.000,1.2345E–003,1.000,1.2345E–003,…

**Compatibility**

Incompatible with existing models.
### Chapter 8  Native Commands

#### QTW?

<table>
<thead>
<tr>
<th>Response</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>When no alarm exists.</td>
</tr>
<tr>
<td>1</td>
<td>Sync. Loss</td>
</tr>
<tr>
<td>2</td>
<td>Clock Loss</td>
</tr>
<tr>
<td>3</td>
<td>CR Unlock</td>
</tr>
<tr>
<td>4</td>
<td>Out of range</td>
</tr>
<tr>
<td>5</td>
<td>Frequency NG</td>
</tr>
<tr>
<td>6</td>
<td>Illegal Error</td>
</tr>
</tbody>
</table>

**Note:**
Displays any one of the alarms.
When two or more alarms have occurred, they are displayed in the following order:
Illegal Error > Frequency NG > Clock Loss (CR Unlock) > Sync Loss > Out of range

**Function**
Queries the Vth vs. Q measurement status.

**Example**
```
> QTW?
< QTW 1
```

**Compatibility**
Incompatible with existing models.
QPH? <numeric1>,<numeric2>

Parameter <numeric1>,<numeric2> = <DECIMAL NUMERIC PROGRAM DATA>
<numeric1>,<numeric2>For the contents of measurement data <result1>
see Table 8.5.3.4-6.
<numeric1>,<numeric2>For the contents of immediate data <result2>
see Table 8.5.3.4-7.

Table 8.5.3.4-6  Vth vs. Q measurement result <result1>

<table>
<thead>
<tr>
<th>Items</th>
<th>&lt;numeric1&gt;,&lt;numeric2&gt;</th>
<th>Response Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>Top side Vth-Error Rate</td>
<td>Phase −1000 to 1000 mUI</td>
<td>1,−1000,…0,1000 Form1</td>
</tr>
<tr>
<td>Bottom side Vth-Error Rate</td>
<td>1,−1000,…1,1000</td>
<td>Form1</td>
</tr>
</tbody>
</table>

Table 8.5.3.4-7  Vth vs. Q immediate data <result2>

<table>
<thead>
<tr>
<th>Items</th>
<th>&lt;numeric1&gt;,&lt;numeric2&gt;</th>
<th>Response Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q value</td>
<td>Phase −1000 to 1000 mUI</td>
<td>2,−1000,…2,1000 Form2</td>
</tr>
<tr>
<td>Optimum BER</td>
<td>3,−1000,…3,1000</td>
<td>Form3</td>
</tr>
<tr>
<td>Optimum Vth</td>
<td>4,−1000,…4,1000</td>
<td>Form4</td>
</tr>
<tr>
<td>Correlation bottom</td>
<td>5,−1000,…5,1000</td>
<td>Form2</td>
</tr>
<tr>
<td>Correlation top</td>
<td>6,−1000,…6,1000</td>
<td>Form2</td>
</tr>
<tr>
<td>σ bottom</td>
<td>7,−1000,…7,1000</td>
<td>Form2</td>
</tr>
<tr>
<td>σ top</td>
<td>8,−1000,…8,1000</td>
<td>Form2</td>
</tr>
<tr>
<td>μ bottom</td>
<td>9,−1000,…9,1000</td>
<td>Form4</td>
</tr>
<tr>
<td>μ top</td>
<td>10,−1000,…10,1000</td>
<td>Form4</td>
</tr>
<tr>
<td>Q Equation</td>
<td>11,−1000,…11,1000</td>
<td>Form2</td>
</tr>
</tbody>
</table>

Response <string> = <STRING RESPONSE DATA>
Table 8.5.3.4-8  Response format

<table>
<thead>
<tr>
<th>Form</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Form1</td>
<td>XX.XXX,Y.YYYYE-YYY</td>
<td>XX.XXX: Threshold (V)</td>
</tr>
<tr>
<td>Vth-ER type</td>
<td></td>
<td>Y.YYYYE-YYY: Error Rate</td>
</tr>
<tr>
<td></td>
<td>----,-------</td>
<td>When no data corresponds to the query.</td>
</tr>
<tr>
<td>Form2</td>
<td>XXX.XX</td>
<td>When 0.00 to 100.00</td>
</tr>
<tr>
<td>% type</td>
<td></td>
<td>When no data corresponds to the query.</td>
</tr>
<tr>
<td>(dB type)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Form3</td>
<td>X.XXXXXE-XXX</td>
<td>When 0.0000E–016 to 1.0000E–000</td>
</tr>
<tr>
<td>Fraction type</td>
<td></td>
<td>When no data corresponds to the query.</td>
</tr>
<tr>
<td></td>
<td>--------</td>
<td>When E–199 or less</td>
</tr>
<tr>
<td>Form4</td>
<td>XX.XXXX</td>
<td>When −4.0000 to 4.0000</td>
</tr>
<tr>
<td>Voltage type</td>
<td>------</td>
<td>When no data corresponds to the query.</td>
</tr>
</tbody>
</table>

Function
Queries the Phase vs. Q measurement result.

Example
To query the measurement result on the TOP side, at a phase of 100 mUI:
> QPH? 0,100
< QPH
1.000,1.2345E-003,1.000,1.2345E-003,1.000,1.2345E-003,…

Compatibility
Incompatible with existing models.

QSW?

Response

<numeric> = <DECIMAL NUMERIC RESPONSE DATA> (FIX1)
0     When no alarm exists.
1     Sync. Loss
2     Clock Loss
3     CR Unlock
4     Out of range
5     Frequency NG
6     Illegal Error

Note:
Displays any one of the alarms.
When two or more alarms have occurred, they are displayed in the following order:
Illegal Error > Frequency NG > Clock Loss (CR Unlock) > Sync Loss > Out of range

Function
Queries the Phase vs. Q measurement status condition.

Example
> QSW?
< QSW 1

Compatibility
Incompatible with existing models.
8.5 Automatic Measurement Commands

8.5.3.5 File menu setting commands

Table 8.5.3.5-1  File menu setting commands

<table>
<thead>
<tr>
<th>No.</th>
<th>Setting Items</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1]</td>
<td>Q Unit</td>
<td>QTS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>QTS?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>QPS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>QPS?</td>
</tr>
</tbody>
</table>

QTS <numeric>

Parameter  \(<\text{numeric}> = \langle\text{DECIMAL NUMERIC PROGRAM DATA}\rangle\>

0 Linear display
1 Log display

Function Sets the Q value display scale during Vth vs. Q measurement to Linear or Log display.

Example To set the display scale to Log display:
\(> \text{QTS 1}\)

Compatibility Incompatible with existing models.

QTS?

Response \(<\text{numeric}> = \langle\text{NR1 NUMERIC RESPONSE DATA}\rangle\> (\text{FIX1})\)

Function Queries whether the Q value display scale during Vth vs. Q measurement is set to Linear or Log display.

Example

\(> \text{QTS}\)

\(<\text{QTS 1}\)

Compatibility Incompatible with existing models.

QPS <scale>

Parameter \(<\text{scale}> = \langle\text{DECIMAL NUMERIC PROGRAM DATA}\rangle\>

0 Linear display
1 Log display

Function Sets the Q value display scale during Phase vs. Q measurement to Linear or Log display.

Example To set the display scale to Log display:
\(> \text{QPS 1}\)

Compatibility Incompatible with existing models.
Chapter 8  Native Commands

QPS?

<table>
<thead>
<tr>
<th>Response</th>
<th>(&lt;\text{scale}\rangle = \langle\text{NR1 NUMERIC RESPONSE DATA}\rangle ) (FIX1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function</td>
<td>Queries whether the Q value display scale during Phase vs. Q measurement is set to Linear or Log display.</td>
</tr>
</tbody>
</table>
| Example  | \(> \text{QPS}\)  
         | \(< \text{QPS 1}\) |
| Compatibility | Incompatible with existing models. |
Chapter 9  Basic Operation of Device Messages

This chapter explains about basic operation of Device Message and Error Message of the MP1800A/MT1810A.

9.1 Basic Operation of Device Messages ........................... 9-2
9.2 Error Messages............................................................. 9-4
  9.2.1 Command Error ................................................ 9-4
  9.2.2 Execution Error.................................................. 9-5
  9.2.3 Device Specific Error ........................................ 9-6
  9.2.4 Supplement / Error No. and
       the event status register ........................................ 9-6
9.1 Basic Operation of Device Messages

- When performing MP1800A/MT1810A port operation by using device messages, the following 3 commands must be transmitted before transmitting port operation commands:

  1. :UENTry:ID <unit_number>
  2. :MODule:ID <module_number>
  3. :PORT:ID <port_number>

The above 3 commands identify the port to be operated. The device messages transmitted after them are regarded to be for the port.

- The target unit for unit operation commands is that specified by 1. :UENTry:ID command.

- The target module for module operation commands is that specified by the :UENTry:ID command and 2. :MODule:ID command.

- In case of operation for PPG module that is setting unit 1, module 2, Port 1, the following three commands must be transmitted before transmitting operation commands:

  Sets to unit 1
  :UENTry:ID 1

  Sets to module 2
  :MODule:ID 2

  Sets to port 1
  :PORT:ID 1
• In case of performing MP1800A/MT1810A operation for function of automatic measurement by using device messages, the following command must be transmitted before operating function:

`:SYSTem:CFUNction <function>`

The above command specifies the location of the automatic measurement function to be operated. The device messages transmitted after them are regarded to be for the specified function. Commands except for selected function will not be enabled.

• In case of performing operation for function of Eye Margin Measurement, the following command must be transmitted before transmitting operation commands.

Sets Eye Margin Measurement.
`:SYSTem:CFUNction EMAR

Operation command must be transmitted or later.

It returns to port operation before starting to operate automatic measurement function.
`:SYSTem:CFUNction OFF

Information of specified location by these commands memorize while the MP1800A/MT1810A is turn on(It is not necessary to transmit each time).
Chapter 9  Basic Operation of Device Messages

9.2 Error Messages

The error code and message is returned as the response of ‘SYSTem:ERRor?’ SCPI command. The following is shown the detail of error message supported in MP1800A/MT1810A.

- Command error
- Execution error
- Device Specific Error

9.2.1 Command Error

When the following error occurred, the event status register "bit 5" is set.

The Error is occurred when the following event is happened.

1. When the MP1800A/MT1810A receives a message against the IEEE488.2 standard.
2. When the device receives a header that does not conform to the specifications of device specific commands and common commands.
3. When a GET (Group Execute Trigger) is sent into a program message.

Table 9.2.1-1  Error messages for command errors

<table>
<thead>
<tr>
<th>Code</th>
<th>Message</th>
<th>Error detection condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>−101</td>
<td>Invalid character</td>
<td>An invalid character is included in a header or parameter.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Notes:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• The top of mnemonic should be stated from alphabet. After this, alphabet character &quot;A&quot; to &quot;Z&quot;, &quot;a&quot; to &quot;z&quot;, underline &quot;_&quot;, number &quot;0&quot; to &quot;9&quot; can be followed.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• A rule of character data is similar to program mnemonic.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Ex. : Case where a header includes #:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:OUTPut:CLOk:#REQuency 200</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Case where a parameter include *:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:OUTPut:CLOk:FREQuency:UNIT MHZ*</td>
</tr>
<tr>
<td>−102</td>
<td>Syntax error</td>
<td>Case that the command excepted a &lt;white space&gt; is only &quot;*&quot;.</td>
</tr>
<tr>
<td>−104</td>
<td>Data type error</td>
<td>The parameter type differs from the specified type.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Notes:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• The character data is send as parameter although it should be sent a numeric parameter.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• The numeric data is send as parameter although it should be a character parameter.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Ex. : Case where a parameter sends data of character type to a command that accepts only numeric data:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:SENSSe:MEASure:EALarm:PERiod 1A,D,M,S</td>
</tr>
</tbody>
</table>
### 9.2 Error Messages

#### Table 9.2.1-1 Error messages for command errors (Cont’d)

<table>
<thead>
<tr>
<th>Code</th>
<th>Message</th>
<th>Error detection condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>–108</td>
<td>Parameter not allowed</td>
<td>The number of parameters does not agree with the defined (required) number.</td>
</tr>
</tbody>
</table>
|      |                          | Ex.: Case where a command of four parameter required send five parameters:  
|      |                          | ‘SENSe:MEASure:EALarm:PERiod 1,1,10,20,30                                                                                                                   |
|      |                          | Case where a command of one parameter required sent no parameter:  
|      |                          | ‘SENSe:MEASure:EALarm:MODE                                                                                                                                     |
| –110 | Command header error     | The top of character except a <white space> is a command separator or not a alphabet (including case that it is only command separator).                  |
| –112 | Program mnemonic too long| The number of program mnemonics is 12 or more.                                                                                                               |
| –113 | Undefined header         | The header syntax is correct but is not defined in the MP1800A/MT1810A.                                                                                   |
|      |                          | Ex.: Case where a wrong header is included:  
|      |                          | ‘SENSe:PA Tern:PYPE PRBS                                                                                                                                       |

#### 9.2.2 Execution Error

When the following error occurred, the event status register "bit 4" is set.

1. When <PROGRAM DATA> followed by the header is against the device specification.
2. When a program message cannot be executed because of the MP1800A/MT1810A condition.

#### Table 9.2.2-1 Error messages for execution errors

<table>
<thead>
<tr>
<th>Code</th>
<th>Message</th>
<th>Error detection condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>–220</td>
<td>Parameter error</td>
<td>An error is found in the parameter.</td>
</tr>
<tr>
<td>–221</td>
<td>Setting conflict</td>
<td>The parameter is correct but cannot be set because of the MP1800A/MT1810A condition.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Ex.: Although the Pattern generation setting is not &quot;Burst&quot;, then the command to change the data output sequence for a Burst data is sent</td>
</tr>
</tbody>
</table>
9.2.3 Device Specific Error

Device specific error indicates that an error other than command error, query error or execution error occurred. When a device specific error occurred, the event status register "bit 3" is set.

<table>
<thead>
<tr>
<th>Code</th>
<th>Message</th>
<th>Error detection condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>–310</td>
<td>System error</td>
<td>A system error occurred.</td>
</tr>
</tbody>
</table>

9.2.4 Supplement / Error No. and the event status register

<table>
<thead>
<tr>
<th>Error No.</th>
<th>Error type</th>
<th>Event Status Register to Be Set</th>
<th>Bit No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>–100 to –199</td>
<td>Command error</td>
<td>CME_BIT</td>
<td>5</td>
</tr>
<tr>
<td>–200 to –299</td>
<td>Execution error</td>
<td>EXE_BIT</td>
<td>4</td>
</tr>
<tr>
<td>–300 to –399</td>
<td>Device specific error</td>
<td>DDE_BIT</td>
<td>3</td>
</tr>
</tbody>
</table>
Appendix A  Command Compatibility List

A.1 Remote Command Compatibility ........................................... A-2
  A.1.1 Compatibility with MP1632C ................................... A-4
  A.1.2 Compatibility with MP1761A/B/C.......................... A-23
  A.1.3 Compatibility with MP1762A/C/D ....................... A-25
  A.1.4 Compatibility with MP1775A................................. A-29
  A.1.5 Compatibility with MP1776A................................. A-31
A.1 Remote Command Compatibility

When executing in the MP1800A or the MT1810A a program used for an existing Anritsu device, the program must be reconfigured with respect to the items described below.

This section describes the compatibility between the MP1800A or the MT1810A commands and the commands for the following existing devices:

- MP1632C Digital Data Analyzer
- MP1761A/B/C Pulse Pattern Generator
- MP1762A/C/D Error Detector
- MP1775A Pulse Pattern Generator
- MP1776A Error Detector

The compatibility level for a given command is classified as “Compatible”, “Partially compatible”, and “Incompatible”.

Compatible: The command for the existing device can be used as is. (A)

Partially compatible: The command is the same, but some parameters differ from those for the corresponding MP1800A or the MT1810A command. These parameters must be limited or used within the restrictions for the MP1800A or the MT1810A. (B)

Incompatible: The command is not compatible with the MP1800A or the MT1810A (the command has changed). Even if a command with the same name may exist, the function is different, or the function itself is not supported in the MP1800A or the MT1810A. (C)

Use the corresponding the MP1800A or the MT1810A command.

Commands newly provided for the MP1800A or the MT1810A are also incompatible with the existing devices. (–)

Compatibility with MP1632C and MP1776A:
When using a command for the MP1632C and MP1776A that is compatible or partially compatible, delete the channel designation number.
Example: To set the data input threshold voltage:
:SENSe3:PATTern:TYPE ZSUBstitution (for MP1632C)
→ :SENSe:PATTern:TYPE ZSUBstitution (for MP1800A/MT1810A)

Example: To set the data input threshold voltage:
:INPut1:DATA:THReshold (for MP1776A)
→ :INPut:DATA:THReshold (for MP1800A/MT1810A)
Appendix A  Command Compatibility List

A.1.1 Compatibility with MP1632C

This section describes the compatibility between the MP1800A or the MT1810A commands and the MP1632C Digital Data Analyzer.

Table A.1.1-1  MP1632C PPG commands

<table>
<thead>
<tr>
<th>Function</th>
<th>Control Message</th>
<th>Data Request Message</th>
<th>Compatibility</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pattern Output Mode</td>
<td>PTO</td>
<td>PTO?</td>
<td>A</td>
</tr>
<tr>
<td>Transmission Pattern</td>
<td>PTN</td>
<td>PTN?</td>
<td>C</td>
</tr>
<tr>
<td>Transmission Pattern Mark Ratio</td>
<td>MRK</td>
<td>MRK?</td>
<td>A</td>
</tr>
<tr>
<td>Transmission Pattern Logic</td>
<td>LGC</td>
<td>LGC?</td>
<td>A</td>
</tr>
<tr>
<td>Bit Shift</td>
<td>SFT</td>
<td>SFT?</td>
<td>A</td>
</tr>
<tr>
<td>Zero-Subst Pattern Length</td>
<td>ZPL</td>
<td>ZPL?</td>
<td>A</td>
</tr>
<tr>
<td>Length of Consecutive Zeros of Zero Subst Pattern</td>
<td>ZLN</td>
<td>ZLN?</td>
<td>A</td>
</tr>
<tr>
<td>PRGM Pattern Length</td>
<td>DLN</td>
<td>DLN?</td>
<td>A</td>
</tr>
<tr>
<td>Burst Mode</td>
<td>BRM</td>
<td>BRM?</td>
<td>A</td>
</tr>
<tr>
<td>Burst Cycle</td>
<td>BRC</td>
<td>BRC?</td>
<td>C</td>
</tr>
<tr>
<td>Burst Enable Length</td>
<td>BRE</td>
<td>BRE?</td>
<td>C</td>
</tr>
<tr>
<td>Number of Pattern Pages</td>
<td>PAG</td>
<td>PAG?</td>
<td>–</td>
</tr>
<tr>
<td>Pattern Data</td>
<td>BIT</td>
<td>BIT?</td>
<td>–</td>
</tr>
<tr>
<td>Pattern Binary Data</td>
<td>PDT</td>
<td>PDT?</td>
<td>A</td>
</tr>
<tr>
<td>Reversion of Transmission Pattern Specification Address Data (address specification)</td>
<td>DRA</td>
<td>–</td>
<td>A</td>
</tr>
<tr>
<td>Reversion of Transmission Pattern Specification Address Data (delta specification)</td>
<td>DRD</td>
<td>–</td>
<td>A</td>
</tr>
<tr>
<td>Pattern Data Input Byte Number</td>
<td>WRT</td>
<td>–</td>
<td>A</td>
</tr>
<tr>
<td>Pattern Data Output Byte Number</td>
<td>–</td>
<td>RED</td>
<td>A</td>
</tr>
<tr>
<td>ALL Preset (All Pages)</td>
<td>ALL</td>
<td>–</td>
<td>A</td>
</tr>
<tr>
<td>ALL Preset (One Pages)</td>
<td>PST</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>Error Insertion</td>
<td>EAD</td>
<td>EAD?</td>
<td>B</td>
</tr>
<tr>
<td>Error Insertion Route</td>
<td>ECH</td>
<td>ECH?</td>
<td>A</td>
</tr>
<tr>
<td>Level of PRBS 1</td>
<td>PML</td>
<td>PML?</td>
<td>–</td>
</tr>
<tr>
<td>Level of PRGM 1</td>
<td>GML</td>
<td>GML?</td>
<td>–</td>
</tr>
</tbody>
</table>
## A.1 Remote Command Compatibility

### Table A.1.1-1  MP1632C PPG commands (Cont’d)

<table>
<thead>
<tr>
<th>Function</th>
<th>Control Message</th>
<th>Data Request Message</th>
<th>Compatibility</th>
</tr>
</thead>
<tbody>
<tr>
<td>On/Off of Output</td>
<td>OON</td>
<td>OON?</td>
<td>C</td>
</tr>
<tr>
<td>On/Off of Clock Output</td>
<td>CON</td>
<td>CON?</td>
<td>A</td>
</tr>
<tr>
<td>On/Off of XClock Output</td>
<td>XCO</td>
<td>XCO?</td>
<td>A</td>
</tr>
<tr>
<td>Setting of Clock Delay</td>
<td>CDL</td>
<td>CDL?</td>
<td>A</td>
</tr>
<tr>
<td>Clock Output Level</td>
<td>CLL</td>
<td>CLL?</td>
<td>–</td>
</tr>
<tr>
<td>XClock Output Level</td>
<td>XCL</td>
<td>XCL?</td>
<td>–</td>
</tr>
<tr>
<td>Clock Output Amplitude</td>
<td>CAP</td>
<td>CAP?</td>
<td>A</td>
</tr>
<tr>
<td>XClock Output Amplitude</td>
<td>XCA</td>
<td>XCA?</td>
<td>A</td>
</tr>
<tr>
<td>Clock Output Offset</td>
<td>COS</td>
<td>COS?</td>
<td>B</td>
</tr>
<tr>
<td>XClock Output Offset</td>
<td>XCF</td>
<td>XCF?</td>
<td>A</td>
</tr>
<tr>
<td>Offset Reference Port Selection</td>
<td>OPS</td>
<td>OPS?</td>
<td>A</td>
</tr>
<tr>
<td>Offset Reference Value</td>
<td>OFS</td>
<td>OFS?</td>
<td>A</td>
</tr>
<tr>
<td>Setting of Clock Duty</td>
<td>CDT</td>
<td>CDT?</td>
<td>A</td>
</tr>
<tr>
<td>Setting of XClock Duty</td>
<td>XDT</td>
<td>XDT?</td>
<td>A</td>
</tr>
<tr>
<td>Setting of Clock DC Impedance</td>
<td>CIM</td>
<td>CIM?</td>
<td>–</td>
</tr>
<tr>
<td>ON/OFF of Data Output</td>
<td>DON</td>
<td>DON?</td>
<td>A</td>
</tr>
<tr>
<td>ON/OFF of XData Output</td>
<td>XDO</td>
<td>XDO?</td>
<td>C</td>
</tr>
<tr>
<td>Setting of Data DC Impedance</td>
<td>DIM</td>
<td>DIM?</td>
<td>–</td>
</tr>
<tr>
<td>Data Cross Point</td>
<td>DCR</td>
<td>DCR?</td>
<td>B</td>
</tr>
<tr>
<td>XData Cross Point</td>
<td>XDC</td>
<td>XDC?</td>
<td>B</td>
</tr>
<tr>
<td>Data Output Level</td>
<td>DAL</td>
<td>DAL?</td>
<td>A</td>
</tr>
<tr>
<td>XData Output Level</td>
<td>XDL</td>
<td>XDL?</td>
<td>A</td>
</tr>
<tr>
<td>Data Output Amplitude</td>
<td>DAP</td>
<td>DAP?</td>
<td>A</td>
</tr>
<tr>
<td>XData Output Amplitude</td>
<td>NAP</td>
<td>NAP?</td>
<td>A</td>
</tr>
<tr>
<td>Data Output Offset</td>
<td>DOS</td>
<td>DOS?</td>
<td>A</td>
</tr>
<tr>
<td>XData Output Offset</td>
<td>NOS</td>
<td>NOS?</td>
<td>A</td>
</tr>
<tr>
<td>Selection of Sync Output Signal</td>
<td>SOP</td>
<td>SOP?</td>
<td>B</td>
</tr>
<tr>
<td>Selection of Pattern Sync Output Position</td>
<td>PSP</td>
<td>PSP?</td>
<td>B</td>
</tr>
<tr>
<td>PPG Clock Loss</td>
<td>–</td>
<td>PCL?</td>
<td>A</td>
</tr>
<tr>
<td>Delay Trouble 3</td>
<td>–</td>
<td>DTR?</td>
<td>A</td>
</tr>
<tr>
<td>Delay Setting State</td>
<td>–</td>
<td>DLY?</td>
<td>A</td>
</tr>
<tr>
<td>Setting of Clock/XClock Grouping</td>
<td>CGR</td>
<td>CGR?</td>
<td>A</td>
</tr>
<tr>
<td>Setting of Data/XData Grouping</td>
<td>TRK</td>
<td>TRK?</td>
<td>A</td>
</tr>
<tr>
<td>PPG/ED Pattern Common Setting</td>
<td>PCO</td>
<td>PCO?</td>
<td>–</td>
</tr>
</tbody>
</table>
### Appendix A  Command Compatibility List

#### Table A.1.1-2  MP1632C Internal Synthesizer commands

<table>
<thead>
<tr>
<th>Function</th>
<th>Control Message</th>
<th>Data Request Message</th>
<th>Compatibility</th>
</tr>
</thead>
<tbody>
<tr>
<td>Setting of Clock Frequency</td>
<td>FRQ</td>
<td>FRQ?</td>
<td>B</td>
</tr>
<tr>
<td>Clock Reference Signal</td>
<td>RFC</td>
<td>RFC?</td>
<td>A</td>
</tr>
<tr>
<td>PLL Lock State</td>
<td>–</td>
<td>PLL?</td>
<td>–</td>
</tr>
</tbody>
</table>

#### Table A.1.1-3  MP1632C ED commands

<table>
<thead>
<tr>
<th>Function</th>
<th>Control Message</th>
<th>Data Request Message</th>
<th>Compatibility</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pattern Reception Mode (Pattern input mode)</td>
<td>PTI</td>
<td>PTI?</td>
<td>A</td>
</tr>
<tr>
<td>Reception Pattern</td>
<td>PTN</td>
<td>PTN?</td>
<td>B</td>
</tr>
<tr>
<td>Reception Pattern Mark Ratio</td>
<td>MRK</td>
<td>MRK?</td>
<td>A</td>
</tr>
<tr>
<td>Reception Pattern Logic</td>
<td>LGC</td>
<td>LGC?</td>
<td>A</td>
</tr>
<tr>
<td>Bit Shift</td>
<td>SFT</td>
<td>SFT?</td>
<td>A</td>
</tr>
<tr>
<td>Zero-Subst Pattern Length</td>
<td>ZPL</td>
<td>ZPL?</td>
<td>–</td>
</tr>
<tr>
<td>Length of Consecutive Zeros of Zero Subst Pattern</td>
<td>ZLN</td>
<td>ZLN?</td>
<td>B</td>
</tr>
<tr>
<td>PRGM Pattern Length</td>
<td>DLN</td>
<td>DLN?</td>
<td>A</td>
</tr>
<tr>
<td>Automatic Synchronization Function</td>
<td>SYN</td>
<td>SYN?</td>
<td>A</td>
</tr>
<tr>
<td>Synchronization Loss Threshold</td>
<td>LTH</td>
<td>LTH?</td>
<td>–</td>
</tr>
<tr>
<td>Synchronization Gain Threshold</td>
<td>GTH</td>
<td>GTH?</td>
<td>–</td>
</tr>
<tr>
<td>Internal Synchronization Threshold</td>
<td>ITH</td>
<td>ITH?</td>
<td>–</td>
</tr>
<tr>
<td>Frame Synchronization Function</td>
<td>FSY</td>
<td>FSY?</td>
<td>–</td>
</tr>
<tr>
<td>Frame Length</td>
<td>FLN</td>
<td>FLN?</td>
<td>A</td>
</tr>
<tr>
<td>Number of Pattern Pages</td>
<td>PAG</td>
<td>PAG?</td>
<td>–</td>
</tr>
<tr>
<td>Number of Pattern Pages (address specification)</td>
<td>ADR</td>
<td>ADR?</td>
<td>–</td>
</tr>
<tr>
<td>Pattern Bit</td>
<td>BIT</td>
<td>BIT?</td>
<td>–</td>
</tr>
<tr>
<td>Pattern Data</td>
<td>PDT</td>
<td>PDT?</td>
<td>A</td>
</tr>
<tr>
<td>Pattern Binary Data</td>
<td>BDT</td>
<td>BDT?</td>
<td>A</td>
</tr>
<tr>
<td>Reversion of Transmission Pattern Specification Address Data (address specification)</td>
<td>DRA</td>
<td>–</td>
<td>A</td>
</tr>
<tr>
<td>Reversion of Transmission Pattern Specification Address Data (delta specification)</td>
<td>DRD</td>
<td>–</td>
<td>A</td>
</tr>
<tr>
<td>Pattern Data Input Byte Number</td>
<td>WRT</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>Pattern Data Output Byte Number</td>
<td>–</td>
<td>RED</td>
<td>–</td>
</tr>
<tr>
<td>ALL Preset (All Pages)</td>
<td>ALL</td>
<td>–</td>
<td>A</td>
</tr>
<tr>
<td>ALL Preset (One Pages)</td>
<td>PST</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>Level of PRBS 1</td>
<td>PML</td>
<td>PML?</td>
<td>–</td>
</tr>
<tr>
<td>Level of PRGM 1</td>
<td>GML</td>
<td>GML?</td>
<td>–</td>
</tr>
<tr>
<td>Setting of Measurement Item</td>
<td>TIT</td>
<td>TIT?</td>
<td>–</td>
</tr>
</tbody>
</table>
### Table A.1.1-3  MP1632C ED commands (Cont’d)

<table>
<thead>
<tr>
<th>Function</th>
<th>Control Message</th>
<th>Data Request Message</th>
<th>Compatibility</th>
</tr>
</thead>
<tbody>
<tr>
<td>Measurement Start/Restart</td>
<td>STA</td>
<td>–</td>
<td>A</td>
</tr>
<tr>
<td>Measurement Stop</td>
<td>STO</td>
<td>–</td>
<td>A</td>
</tr>
<tr>
<td>Measurement restart setting</td>
<td>MRS</td>
<td>MRS?</td>
<td>C</td>
</tr>
<tr>
<td>Setting of Measurement Mode</td>
<td>MOD</td>
<td>MOD?</td>
<td>A</td>
</tr>
<tr>
<td>Measurement Period Setting</td>
<td>PRD</td>
<td>PRD?</td>
<td>A</td>
</tr>
<tr>
<td>Timed Start Setting</td>
<td>TSS</td>
<td>TSS?</td>
<td>–</td>
</tr>
<tr>
<td>Timed Start Time Setting</td>
<td>STI</td>
<td>STI?</td>
<td>–</td>
</tr>
<tr>
<td>Error Detection Mode Selection</td>
<td>ETY</td>
<td>ETY?</td>
<td>–</td>
</tr>
<tr>
<td>Measurement Mask Route Selection</td>
<td>SCH</td>
<td>SCH?</td>
<td>–</td>
</tr>
<tr>
<td>Measurement Start Time</td>
<td>–</td>
<td>MSA?</td>
<td>–</td>
</tr>
<tr>
<td>Measurement Stop Time</td>
<td>–</td>
<td>MSO?</td>
<td>A</td>
</tr>
<tr>
<td>Measurement State</td>
<td>–</td>
<td>MSR?</td>
<td>A</td>
</tr>
<tr>
<td>Measurement Elapsed Time</td>
<td>–</td>
<td>MLP?</td>
<td>B</td>
</tr>
<tr>
<td>Measurement Residual Time</td>
<td>–</td>
<td>ETT?</td>
<td>A</td>
</tr>
<tr>
<td>Intermediate Data Creation Time</td>
<td>–</td>
<td>INT?</td>
<td>–</td>
</tr>
<tr>
<td>Alarm Occurrence Time</td>
<td>–</td>
<td>AOT?</td>
<td>–</td>
</tr>
<tr>
<td>Alarm Recovery Time</td>
<td>–</td>
<td>ART?</td>
<td>–</td>
</tr>
<tr>
<td>Setting of Eye Margin Measurement Mode</td>
<td>EMD</td>
<td>EMD?</td>
<td>–</td>
</tr>
<tr>
<td>Eye Margin Measurement Item Setting</td>
<td>EYT</td>
<td>EYT?</td>
<td>C</td>
</tr>
<tr>
<td>Eye Margin Measurement Threshold Setting</td>
<td>EMT</td>
<td>EMT?</td>
<td>C</td>
</tr>
<tr>
<td>Eye Margin Measurement Resolution Setting</td>
<td>EMR</td>
<td>EMR?</td>
<td>C</td>
</tr>
<tr>
<td>Eye Diagram Measurement Threshold Setting</td>
<td>EDT</td>
<td>EDT?</td>
<td>C</td>
</tr>
<tr>
<td>Eye Diagram Measurement Effective Points</td>
<td>–</td>
<td>EDP?</td>
<td>A</td>
</tr>
<tr>
<td>Setting of Auto search Mode</td>
<td>ASM</td>
<td>ASM?</td>
<td>B</td>
</tr>
<tr>
<td>Auto search Function</td>
<td>SRH</td>
<td>SRH?</td>
<td>B</td>
</tr>
<tr>
<td>PRBS pattern Search Function</td>
<td>PSH</td>
<td>PSH?</td>
<td>–</td>
</tr>
<tr>
<td>Clock Input Polarity</td>
<td>CPL</td>
<td>CPL?</td>
<td>–</td>
</tr>
<tr>
<td>Clock Input Phase Setting</td>
<td>CPA</td>
<td>CPA?</td>
<td>B</td>
</tr>
<tr>
<td>Clock Input Terminal Voltage</td>
<td>CTM</td>
<td>CTM?</td>
<td>–</td>
</tr>
<tr>
<td>Data Input Level</td>
<td>DAL</td>
<td>DAL?</td>
<td>B</td>
</tr>
<tr>
<td>Data Input Threshold Setting</td>
<td>DTH</td>
<td>DTH?</td>
<td>A</td>
</tr>
<tr>
<td>Setting of Data Input Terminal Voltage</td>
<td>DTM</td>
<td>DTM?</td>
<td>B</td>
</tr>
<tr>
<td>Clock Loss Process Option</td>
<td>CLS</td>
<td>CLS?</td>
<td>A</td>
</tr>
<tr>
<td>Sync Loss Process Option</td>
<td>SLS</td>
<td>SLS?</td>
<td>A</td>
</tr>
</tbody>
</table>
### Table A.1.1-3 MP1632C ED commands (Cont’d)

<table>
<thead>
<tr>
<th>Function</th>
<th>Control Message</th>
<th>Data Request Message</th>
<th>Compatibility</th>
</tr>
</thead>
<tbody>
<tr>
<td>Measurement Interval Time Selection</td>
<td>ITV</td>
<td>ITV?</td>
<td>B</td>
</tr>
<tr>
<td>Error Performance Data Measurement ON/OFF</td>
<td>PRF</td>
<td>PRF?</td>
<td>–</td>
</tr>
<tr>
<td>Error Performance Threshold Setting</td>
<td>ETH</td>
<td>ETH?</td>
<td>A</td>
</tr>
<tr>
<td>Threshold EI/%EFI Measurement ON/OFF</td>
<td>TEI</td>
<td>TEI?</td>
<td>–</td>
</tr>
<tr>
<td>Power Fail Measurement ON/OFF</td>
<td>PFM</td>
<td>PFM?</td>
<td>–</td>
</tr>
<tr>
<td>Error Ratio Measurement Result</td>
<td>–</td>
<td>ER?</td>
<td>A</td>
</tr>
<tr>
<td>Error Count Measurement Result</td>
<td>–</td>
<td>EC?</td>
<td>A</td>
</tr>
<tr>
<td>Clock Count Measurement Result</td>
<td>–</td>
<td>CC?</td>
<td>A</td>
</tr>
<tr>
<td>Error Interval Count Measurement Result</td>
<td>–</td>
<td>EI?</td>
<td>A</td>
</tr>
<tr>
<td>Error Free Interval Rate Measurement Result</td>
<td>–</td>
<td>EFT?</td>
<td>A</td>
</tr>
<tr>
<td>Clock Frequency Measurement Result</td>
<td>–</td>
<td>FRQ?</td>
<td>A</td>
</tr>
<tr>
<td>Performance % Measurement Result</td>
<td>–</td>
<td>PFP?</td>
<td>A</td>
</tr>
<tr>
<td>Performance Count Measurement Result</td>
<td>–</td>
<td>PFC?</td>
<td>A</td>
</tr>
<tr>
<td>Threshold EI Measurement Result</td>
<td>–</td>
<td>THE?</td>
<td>A</td>
</tr>
<tr>
<td>Threshold %EFI Measurement Result</td>
<td>–</td>
<td>THF?</td>
<td>A</td>
</tr>
<tr>
<td>Alarm Interval Measurement Result</td>
<td>–</td>
<td>AIN?</td>
<td>A</td>
</tr>
<tr>
<td>1 Second Average Error Ratio Measurement Result</td>
<td>–</td>
<td>OER?</td>
<td>–</td>
</tr>
<tr>
<td>1 Second Average Error Count Measurement Result</td>
<td>–</td>
<td>OEC?</td>
<td>–</td>
</tr>
<tr>
<td>Data Output Format</td>
<td>FMT</td>
<td>FMT?</td>
<td>–</td>
</tr>
<tr>
<td>Intermediate Measurement Result Output Function</td>
<td>–</td>
<td>IMD?</td>
<td>A</td>
</tr>
<tr>
<td>Eye Margin Measurement Result</td>
<td>–</td>
<td>EMM?</td>
<td>–</td>
</tr>
<tr>
<td>Eye Diagram Measurement Result</td>
<td>–</td>
<td>EDM?</td>
<td>C</td>
</tr>
<tr>
<td>Measurement Result Store Function</td>
<td>BST</td>
<td>–</td>
<td>A</td>
</tr>
<tr>
<td>Measurement Result Store Buffer Clear Function</td>
<td>BCL</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>Clock Loss Status</td>
<td>–</td>
<td>CLI?</td>
<td>A</td>
</tr>
<tr>
<td>Sync Loss Status</td>
<td>–</td>
<td>SLI?</td>
<td>A</td>
</tr>
<tr>
<td>Error Detection Status</td>
<td>–</td>
<td>ERS?</td>
<td>A</td>
</tr>
<tr>
<td>Error Detection Status</td>
<td>–</td>
<td>BES?</td>
<td>A</td>
</tr>
<tr>
<td>Power Fail Status</td>
<td>–</td>
<td>POF?</td>
<td>–</td>
</tr>
<tr>
<td>Delay Trouble</td>
<td>–</td>
<td>DTR?</td>
<td>–</td>
</tr>
<tr>
<td>Delay Setting Status</td>
<td>–</td>
<td>DLY?</td>
<td>A</td>
</tr>
<tr>
<td>Synchronous Signal Output Selection</td>
<td>SOP</td>
<td>SOP?</td>
<td>B</td>
</tr>
<tr>
<td>PPG/ED Pattern Common Setting</td>
<td>PCO</td>
<td>PCO?</td>
<td>–</td>
</tr>
<tr>
<td>Function</td>
<td>Control Message</td>
<td>Data Request Message</td>
<td>Compatibility</td>
</tr>
<tr>
<td>--------------------------------------------------------------------------</td>
<td>-----------------</td>
<td>----------------------</td>
<td>---------------</td>
</tr>
<tr>
<td>Child window Open/Close</td>
<td>WOC</td>
<td>WOC?</td>
<td>–</td>
</tr>
<tr>
<td>System Child window panel selection</td>
<td>SYW</td>
<td>SYW?</td>
<td>–</td>
</tr>
<tr>
<td>System Child window panel selection</td>
<td>SEW</td>
<td>SEW?</td>
<td>–</td>
</tr>
<tr>
<td>Test Menu System Child window panel selection</td>
<td>TEW</td>
<td>TEW?</td>
<td>–</td>
</tr>
<tr>
<td>Real Time/Measurement Time display selection</td>
<td>TIM</td>
<td>TIM?</td>
<td>A</td>
</tr>
<tr>
<td>Result Child window Sub window selection</td>
<td>REW</td>
<td>REW?</td>
<td>–</td>
</tr>
<tr>
<td>Intermediate Result display Function</td>
<td>CUR</td>
<td>CUR?</td>
<td>A</td>
</tr>
<tr>
<td>Performance Result Display Function</td>
<td>PFD</td>
<td>PFD?</td>
<td>–</td>
</tr>
<tr>
<td>Zoom window display ON/OFF setting</td>
<td>ZOM</td>
<td>ZOM?</td>
<td>A</td>
</tr>
<tr>
<td>Zoom window display Item setting</td>
<td>ZOI</td>
<td>ZOI?</td>
<td>–</td>
</tr>
<tr>
<td>Zoom window Performance item setting</td>
<td>ZOP</td>
<td>ZOP?</td>
<td>–</td>
</tr>
<tr>
<td>Zoom window Threshold EI/%EFI setting</td>
<td>ZOT</td>
<td>ZOT?</td>
<td>–</td>
</tr>
<tr>
<td>History Lamp Reset</td>
<td>HRE</td>
<td>–</td>
<td>A</td>
</tr>
<tr>
<td>Eye Margin Measurement Display Error Rate</td>
<td>DER</td>
<td>DER?</td>
<td>C</td>
</tr>
<tr>
<td>Eye Diagram Display window selection</td>
<td>EDD</td>
<td>EDD?</td>
<td>–</td>
</tr>
<tr>
<td>Eye diagram display Auto Scale</td>
<td>ASC</td>
<td>–</td>
<td>C</td>
</tr>
<tr>
<td>Eye diagram Voltage Scale Setting</td>
<td>VSC</td>
<td>VSC?</td>
<td>C</td>
</tr>
<tr>
<td>Eye diagram Phase Scale Setting</td>
<td>PSC</td>
<td>PSC?</td>
<td>C</td>
</tr>
<tr>
<td>Eye diagram Marker Display ON/OFF</td>
<td>EMA</td>
<td>EMA?</td>
<td>–</td>
</tr>
<tr>
<td>Marker Operation Mode</td>
<td>MKM</td>
<td>MKM?</td>
<td>–</td>
</tr>
<tr>
<td>Marker Specification Threshold</td>
<td>MKT</td>
<td>MKT?</td>
<td>–</td>
</tr>
<tr>
<td>Marker Move Specification</td>
<td>MMV</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>Marker Position</td>
<td>MKP</td>
<td>MKP?</td>
<td>–</td>
</tr>
<tr>
<td>Customize window display Slot setting</td>
<td>CUS</td>
<td>CUS?</td>
<td>–</td>
</tr>
<tr>
<td>Customize window display Item setting</td>
<td>CUI</td>
<td>CUI?</td>
<td>–</td>
</tr>
<tr>
<td>Customize window display Item Port setting</td>
<td>CUP</td>
<td>CUP?</td>
<td>–</td>
</tr>
<tr>
<td>Customize window Pattern Setting Unit</td>
<td>CPT</td>
<td>CPT?</td>
<td>A</td>
</tr>
<tr>
<td>Customize Edit Pattern Position</td>
<td>CPF</td>
<td>CPF?</td>
<td>–</td>
</tr>
<tr>
<td>Customize window Measurement display Item</td>
<td>CRI</td>
<td>CRI?</td>
<td>–</td>
</tr>
<tr>
<td>System Error Monitor Buzzer ON/OFF</td>
<td>SYS</td>
<td>SYS?</td>
<td>A</td>
</tr>
<tr>
<td>Error Monitor Buzzer ON/OFF</td>
<td>MON</td>
<td>MON?</td>
<td>A</td>
</tr>
<tr>
<td>Alarm Monitor Buzzer ON/OFF</td>
<td>ALM</td>
<td>ALM?</td>
<td>A</td>
</tr>
<tr>
<td>System Error Type</td>
<td>SYT</td>
<td>SYT?</td>
<td>B</td>
</tr>
<tr>
<td>Internal Timer Setting</td>
<td>RTM</td>
<td>RTM?</td>
<td>–</td>
</tr>
</tbody>
</table>
## Appendix A  Command Compatibility List

### Table A.1.1-4  MP1632C Main Frame commands (Cont’d)

<table>
<thead>
<tr>
<th>Function</th>
<th>Control Message</th>
<th>Data Request Message</th>
<th>Compatibility</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input/Output Buffer Size</td>
<td>BUF</td>
<td>BUF?</td>
<td>–</td>
</tr>
<tr>
<td>Hardware Organization</td>
<td>–</td>
<td>HDR?</td>
<td>C</td>
</tr>
<tr>
<td>Software Organization</td>
<td>–</td>
<td>SFR?</td>
<td>–</td>
</tr>
<tr>
<td>Data Recall</td>
<td>RCL</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>Data Save</td>
<td>SAV</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>Quick Recall</td>
<td>QRC</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>Quick Save</td>
<td>QSA</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>File No./Directory Mode Selection</td>
<td>FIL</td>
<td>FIL?</td>
<td>–</td>
</tr>
<tr>
<td>File Catalog</td>
<td>–</td>
<td>CAT?</td>
<td>–</td>
</tr>
<tr>
<td>Floppy Access Status</td>
<td>–</td>
<td>MAC?</td>
<td>–</td>
</tr>
<tr>
<td>Initialize</td>
<td>INI</td>
<td>–</td>
<td>A</td>
</tr>
<tr>
<td>Hard Copy</td>
<td>HCP</td>
<td>–</td>
<td>A</td>
</tr>
<tr>
<td>Prints Text</td>
<td>PTX</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>Threshold EI/%EFI print selection</td>
<td>THR</td>
<td>THR?</td>
<td>–</td>
</tr>
<tr>
<td>Performance Data print selection</td>
<td>EPF</td>
<td>EPF?</td>
<td>–</td>
</tr>
<tr>
<td>Error Measurement print selection</td>
<td>ERP</td>
<td>ERP?</td>
<td>–</td>
</tr>
<tr>
<td>Alarm Interval print selection</td>
<td>ALP</td>
<td>ALP?</td>
<td>–</td>
</tr>
<tr>
<td>Frequency Data print selection</td>
<td>FRP</td>
<td>FRP?</td>
<td>–</td>
</tr>
<tr>
<td>Manual Print Start</td>
<td>PSA</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>Addressed Pattern Print</td>
<td>PAD</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>All Pattern Print</td>
<td>PAL</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>Eye Margin Measurement Result Print</td>
<td>PEM</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>Recall Eye Diagram Print</td>
<td>PER</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>LSB/MSB Swap Command</td>
<td>SWP</td>
<td>SWP?</td>
<td>–</td>
</tr>
<tr>
<td>Termination Select</td>
<td>TRM</td>
<td>TRM?</td>
<td>A</td>
</tr>
<tr>
<td>Service Request Enable Register (ED)</td>
<td>SRQ</td>
<td>SRQ?</td>
<td>C</td>
</tr>
<tr>
<td>Status Byte Register (ED)</td>
<td>STB</td>
<td>STB?</td>
<td>C</td>
</tr>
<tr>
<td>Standard Event Status Enable Register (ED)</td>
<td>ESE</td>
<td>ESE?</td>
<td>C</td>
</tr>
<tr>
<td>Event Status Register (ED)</td>
<td>ESR</td>
<td>ESR?</td>
<td>C</td>
</tr>
<tr>
<td>Extended Event Status Enable Register (ED)(ED)</td>
<td>EES</td>
<td>EES?</td>
<td>C</td>
</tr>
<tr>
<td>Event Status Register (ED)</td>
<td>EER</td>
<td>EER?</td>
<td>C</td>
</tr>
<tr>
<td>Service Request Enable Register (PPG/Synthe)</td>
<td>SRQ</td>
<td>SRQ?</td>
<td>C</td>
</tr>
<tr>
<td>Status Byte Register (PPG/Synthe)</td>
<td>STB</td>
<td>STB?</td>
<td>C</td>
</tr>
<tr>
<td>Standard Event Status Enable Register (PPG/Synthe)</td>
<td>ESE</td>
<td>ESE?</td>
<td>C</td>
</tr>
</tbody>
</table>
### Table A.1.1-4  MP1632C Main Frame commands (Cont’d)

<table>
<thead>
<tr>
<th>Function</th>
<th>Control Message</th>
<th>Data Request Message</th>
<th>Compatibility</th>
</tr>
</thead>
<tbody>
<tr>
<td>Event Status Register (PPG/Synthe)</td>
<td>ESR</td>
<td>ESR?</td>
<td>C</td>
</tr>
<tr>
<td>Extended Event Status Enable Register (PPG/Synthe)</td>
<td>EES</td>
<td>EES?</td>
<td>C</td>
</tr>
<tr>
<td>Event Status Register (PPG/Synthe)</td>
<td>EER</td>
<td>EER?</td>
<td>C</td>
</tr>
</tbody>
</table>

### Table A.1.1-5  MP1632C SOURce Subsystem

<table>
<thead>
<tr>
<th>Commands</th>
<th>Compatibility</th>
</tr>
</thead>
<tbody>
<tr>
<td>:SOURce:PATTern:OMODE &lt;mode&gt;</td>
<td>A</td>
</tr>
<tr>
<td>:SOURce:PATTern:OMODE?</td>
<td>A</td>
</tr>
<tr>
<td>:SOURce:PATTern:TYPE &lt;type&gt;</td>
<td>B</td>
</tr>
<tr>
<td>:SOURce:PATTern:TYPE?</td>
<td>B</td>
</tr>
<tr>
<td>:SOURce:PATTern:PRBS:MRATio &lt;mratio&gt;</td>
<td>A</td>
</tr>
<tr>
<td>:SOURce:PATTern:PRBS:MRATio?</td>
<td>A</td>
</tr>
<tr>
<td>:SOURce:PATTern:PRBS:BShift &lt;numeric&gt;</td>
<td>A</td>
</tr>
<tr>
<td>:SOURce:PATTern:PRBS:BShift?</td>
<td>A</td>
</tr>
<tr>
<td>:SOURce:PATTern:ZSUBstitute:LENGth &lt;numeric&gt;</td>
<td>A</td>
</tr>
<tr>
<td>:SOURce:PATTern:ZSUBstitute:LENGth?</td>
<td>A</td>
</tr>
<tr>
<td>:SOURce:PATTern:ZSUBstitute:ZLENgth &lt;numeric&gt;</td>
<td>A</td>
</tr>
<tr>
<td>:SOURce:PATTern:ZSUBstitute:ZLENgth?</td>
<td>A</td>
</tr>
<tr>
<td>:SOURce:PATTern:ZSUBstitute:LOGic &lt;logic&gt;</td>
<td>–</td>
</tr>
<tr>
<td>:SOURce:PATTern:PROgram:LENGth &lt;numeric&gt;</td>
<td>–</td>
</tr>
<tr>
<td>:SOURce:PATTern:PROgram:LENGth?</td>
<td>–</td>
</tr>
<tr>
<td>:SOURce:PATTern:PROgram:LOGic &lt;logic&gt;</td>
<td>–</td>
</tr>
<tr>
<td>:SOURce:PATTern:PROgram:LOGic?</td>
<td>–</td>
</tr>
<tr>
<td>:SOURce:PATTern:BURSt:MODE &lt;mode&gt;</td>
<td>A</td>
</tr>
<tr>
<td>:SOURce:PATTern:BURSt:MODE?</td>
<td>A</td>
</tr>
<tr>
<td>:SOURce:PATTern:BURSt:CYCLe &lt;numeric&gt;</td>
<td>C</td>
</tr>
<tr>
<td>:SOURce:PATTern:BURSt:CYCLe?</td>
<td>C</td>
</tr>
<tr>
<td>:SOURce:PATTern:BURSt:ELENgth &lt;numeric&gt;</td>
<td>C</td>
</tr>
<tr>
<td>:SOURce:PATTern:BURSt:ELENgth?</td>
<td>C</td>
</tr>
<tr>
<td>:SOURce:PATTern:DATA:WHOLe &lt;start&gt;,&lt;end&gt;,&lt;data&gt;</td>
<td>A</td>
</tr>
<tr>
<td>:SOURce:PATTern:DATA:WHOLe? &lt;start&gt;</td>
<td>A</td>
</tr>
</tbody>
</table>
### Appendix A Command Compatibility List

#### Table A.1.1-5 MP1632C SOURce Subsystem (Cont’d)

<table>
<thead>
<tr>
<th>Commands</th>
<th>Compatibility</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>:SOURce:PATTern:BDATa:WHOLe &lt;start&gt;,&lt;end&gt;,&lt;data&gt;</code></td>
<td>A</td>
</tr>
<tr>
<td><code>:SOURce:PATTern:BDATa:WHOLe? &lt;start&gt;</code></td>
<td>A</td>
</tr>
<tr>
<td><code>:SOURce:PATTern:DREVerse:ADDRess &lt;start&gt;,&lt;end&gt;</code></td>
<td>A</td>
</tr>
<tr>
<td><code>:SOURce:PATTern:DREVerse:DELTa &lt;start&gt;,&lt;delta&gt;</code></td>
<td>A</td>
</tr>
<tr>
<td><code>:SOURce:PATTern:EADDition:SET &lt;boolean&gt;</code></td>
<td>C</td>
</tr>
<tr>
<td><code>:SOURce:PATTern:EADDition:SINGle</code></td>
<td>A</td>
</tr>
<tr>
<td><code>:SOURce:PATTern:EADDition:RATE &lt;mode&gt;</code></td>
<td>B</td>
</tr>
<tr>
<td><code>:SOURce:PATTern:EADDition:ROUTe &lt;route&gt;</code></td>
<td>A</td>
</tr>
<tr>
<td><code>:SOURce:PATTern:EADDition:ROUTe?</code></td>
<td>A</td>
</tr>
<tr>
<td><code>:SOURce:PATTern:LOGic:PRBS &lt;mark&gt;</code></td>
<td>–</td>
</tr>
<tr>
<td><code>:SOURce:PATTern:LOGic:PRGM &lt;mark&gt;</code></td>
<td>–</td>
</tr>
</tbody>
</table>

#### Table A.1.1-6 MP1632C OUTPut Subsystem

<table>
<thead>
<tr>
<th>Commands</th>
<th>Compatibility</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>:OUTPut:SET</code></td>
<td>C</td>
</tr>
<tr>
<td><code>:OUTPut:SET?</code></td>
<td>C</td>
</tr>
<tr>
<td><code>:OUTPut:CLOCk:OUTPut &lt;port&gt;,&lt;boolean&gt;</code></td>
<td>C</td>
</tr>
<tr>
<td><code>:OUTPut:CLOCk:OUTPut? &lt;port&gt;</code></td>
<td>C</td>
</tr>
<tr>
<td><code>:OUTPut:CLOCk:DELa y &lt;numeric&gt;</code></td>
<td>–</td>
</tr>
<tr>
<td><code>:OUTPut:CLOCk:DELa y?</code></td>
<td>–</td>
</tr>
<tr>
<td><code>:OUTPut:CLOCk:LEV el &lt;port&gt;,&lt;level&gt;</code></td>
<td>–</td>
</tr>
<tr>
<td><code>:OUTPut:CLOCk:LEV el? &lt;port&gt;</code></td>
<td>–</td>
</tr>
<tr>
<td><code>:OUTPut:CLOCk:AMPLitude &lt;port&gt;,&lt;numeric&gt;</code></td>
<td>A</td>
</tr>
<tr>
<td><code>:OUTPut:CLOCk:AMPLitude? &lt;port&gt;</code></td>
<td>A</td>
</tr>
<tr>
<td><code>:OUTPut:CLOCk:OFFSet &lt;port&gt;,&lt;numeric&gt;</code></td>
<td>B</td>
</tr>
<tr>
<td><code>:OUTPut:CLOCk:OFFSet? &lt;port&gt;</code></td>
<td>B</td>
</tr>
<tr>
<td><code>:OUTPut:CLOCk:OREFerence &lt;port&gt;,&lt;offset&gt;</code></td>
<td>–</td>
</tr>
<tr>
<td><code>:OUTPut:CLOCk:OREFerence? &lt;port&gt;</code></td>
<td>–</td>
</tr>
<tr>
<td><code>:OUTPut:CLOCk:Duty &lt;port&gt;,&lt;numeric&gt;</code></td>
<td>A</td>
</tr>
<tr>
<td><code>:OUTPut:CLOCk:Duty? &lt;port&gt;</code></td>
<td>A</td>
</tr>
</tbody>
</table>
### Remote Command Compatibility

#### Table A.1.1-6 MP1632C OUTPut Subsystem (Cont’d)

<table>
<thead>
<tr>
<th>Commands</th>
<th>Compatibility</th>
</tr>
</thead>
<tbody>
<tr>
<td>:OUTPut:CLOCk:IMPedance &lt;impedance&gt;</td>
<td>–</td>
</tr>
<tr>
<td>:OUTPut:CLOCk:IMPedance?</td>
<td>–</td>
</tr>
<tr>
<td>:OUTPut:DATA:OUTPut &lt;port&gt;,&lt;boolean&gt;</td>
<td>C</td>
</tr>
<tr>
<td>:OUTPut:DATA:OUTPut? &lt;port&gt;</td>
<td>C</td>
</tr>
<tr>
<td>:OUTPut:DATA:IMPedance &lt;impedance&gt;</td>
<td>–</td>
</tr>
<tr>
<td>:OUTPut:DATA:CPoint &lt;port&gt;,&lt;numeric&gt;</td>
<td>B</td>
</tr>
<tr>
<td>:OUTPut:DATA:CPPoint? &lt;port&gt;</td>
<td>B</td>
</tr>
<tr>
<td>:OUTPut:DATA:LEVel &lt;port&gt;,&lt;level&gt;</td>
<td>A</td>
</tr>
<tr>
<td>:OUTPut:DATA:LEVel? &lt;port&gt;</td>
<td>A</td>
</tr>
<tr>
<td>:OUTPut:DATA:AMPLitude &lt;port&gt;,&lt;numeric&gt;</td>
<td>A</td>
</tr>
<tr>
<td>:OUTPut:DATA:AMPLitude? &lt;port&gt;</td>
<td>A</td>
</tr>
<tr>
<td>:OUTPut:DATA:OFFSet &lt;port&gt;,&lt;numeric&gt;</td>
<td>A</td>
</tr>
<tr>
<td>:OUTPut:DATA:OFFSet? &lt;port&gt;</td>
<td>A</td>
</tr>
<tr>
<td>:OUTPut:DATA:OREFerence &lt;port&gt;,&lt;offset&gt;</td>
<td>–</td>
</tr>
<tr>
<td>:OUTPut:DATA:OREFerence? &lt;port&gt;</td>
<td>–</td>
</tr>
<tr>
<td>:OUTPut:SYNC:SOURce &lt;source&gt;</td>
<td>A</td>
</tr>
<tr>
<td>:OUTPut:SYNC:SOURce?</td>
<td>A</td>
</tr>
<tr>
<td>:OUTPut:SYNC:POSition &lt;address&gt;</td>
<td>B</td>
</tr>
<tr>
<td>:OUTPut:SYNC:SOURce &lt;source&gt;</td>
<td>A</td>
</tr>
<tr>
<td>:OUTPut:SYNC:SOURce?</td>
<td>A</td>
</tr>
<tr>
<td>:OUTPut:CLOCk:FREQuency &lt;numeric&gt;</td>
<td>B</td>
</tr>
<tr>
<td>:OUTPut:CLOCk:FREQuency?</td>
<td>B</td>
</tr>
<tr>
<td>:OUTPut:RCLock:SELect &lt;clock&gt;</td>
<td>B</td>
</tr>
<tr>
<td>:OUTPut:RCLock:SELect?</td>
<td>B</td>
</tr>
</tbody>
</table>

#### Table A.1.1-7 MP1632C CALCulate Subsystem

<table>
<thead>
<tr>
<th>Commands</th>
<th>Compatibility</th>
</tr>
</thead>
<tbody>
<tr>
<td>:CALCulate:DATA:MONitor?</td>
<td>C</td>
</tr>
</tbody>
</table>
## Appendix A  Command Compatibility List

### Table A.1.1-8  MP1632C INStrument Subsystem

<table>
<thead>
<tr>
<th>Commands</th>
<th>Compatibility</th>
</tr>
</thead>
<tbody>
<tr>
<td>:INSTrument:COUPle:COLCk:TRACking &lt;boolean&gt;</td>
<td></td>
</tr>
<tr>
<td>:INSTrument:COUPle:COLCk:TRACking?</td>
<td></td>
</tr>
<tr>
<td>:INSTrument:COUPle:DATA:TRACking &lt;boolean&gt;</td>
<td></td>
</tr>
<tr>
<td>:INSTrument:COUPle:DATA:TRACking?</td>
<td></td>
</tr>
<tr>
<td>:INSTrument:COUPle:PATTern:SET &lt;boolean&gt;</td>
<td></td>
</tr>
<tr>
<td>:INSTrument:COUPle:PATTern:SET?</td>
<td></td>
</tr>
<tr>
<td>:INSTrument:COUPle:PATTern:SET &lt;boolean&gt;</td>
<td></td>
</tr>
<tr>
<td>:INSTrument:COUPle:PATTern:SET?</td>
<td></td>
</tr>
</tbody>
</table>

### Table A.1.1-9  MP1632C SENSe Subsystem

<table>
<thead>
<tr>
<th>Commands</th>
<th>Compatibility</th>
</tr>
</thead>
<tbody>
<tr>
<td>:SENSe:PATTern:IMODe &lt;mode&gt;</td>
<td>A</td>
</tr>
<tr>
<td>:SENSe:PATTern:IMODe?</td>
<td>A</td>
</tr>
<tr>
<td>:SENSe:PATTern:TYPE &lt;type&gt;</td>
<td>B</td>
</tr>
<tr>
<td>:SENSe:PATTern:TYPE?</td>
<td>B</td>
</tr>
<tr>
<td>:SENSe:PATTern:PRBS:MRATio &lt;mratio&gt;</td>
<td>A</td>
</tr>
<tr>
<td>:SENSe:PATTern:PRBS:MRATio?</td>
<td>A</td>
</tr>
<tr>
<td>:SENSe:PATTern:PRBS:BShift &lt;numeric&gt;</td>
<td>A</td>
</tr>
<tr>
<td>:SENSe:PATTern:PRBS:BShift?</td>
<td>A</td>
</tr>
<tr>
<td>:SENSe:PATTern:ZSUBstitute:LENGth &lt;numeric&gt;</td>
<td>A</td>
</tr>
<tr>
<td>:SENSe:PATTern:ZSUBstitute:LENGth?</td>
<td>A</td>
</tr>
<tr>
<td>:SENSe:PATTern:ZSUBstitute:ZLENgth &lt;numeric&gt;</td>
<td>B</td>
</tr>
<tr>
<td>:SENSe:PATTern:ZSUBstitute:ZLENgth?</td>
<td>B</td>
</tr>
<tr>
<td>:SENSe:PATTern:ZSUBstitute:LOGic &lt;logic&gt;</td>
<td></td>
</tr>
<tr>
<td>:SENSe:PATTern:ZSUBstitute:LOGic?</td>
<td></td>
</tr>
<tr>
<td>:SENSe:PATTern:PROGram:LENGth &lt;numeric&gt;</td>
<td></td>
</tr>
<tr>
<td>:SENSe:PATTern:PROGram:LENGth?</td>
<td></td>
</tr>
<tr>
<td>:SENSe:PATTern:PROGram:LOGic &lt;logic&gt;</td>
<td></td>
</tr>
<tr>
<td>:SENSe:PATTern:PROGram:LOGic?</td>
<td></td>
</tr>
<tr>
<td>:SENSe:PATTern:SYNC:ASYNc &lt;boolean&gt;</td>
<td>A</td>
</tr>
<tr>
<td>:SENSe:PATTern:SYNC:ASYNc?</td>
<td>A</td>
</tr>
<tr>
<td>:SENSe:PATTern:SYNC:THReshold:LOSS &lt;thre&gt;</td>
<td></td>
</tr>
<tr>
<td>:SENSe:PATTern:SYNC:THReshold:LOSS?</td>
<td></td>
</tr>
</tbody>
</table>
## A.1 Remote Command Compatibility

### Table A.1.1-9 MP1632C SENSE Subsystem (Cont’d)

<table>
<thead>
<tr>
<th>Commands</th>
<th>Compatibility</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>:SENSe:PATTern:SYNC:THReshold:GAIN &lt;thre&gt;</code></td>
<td>C</td>
</tr>
<tr>
<td><code>:SENSe:PATTern:SYNC:PSMode &lt;mode&gt;</code></td>
<td>A</td>
</tr>
<tr>
<td><code>:SENSe:PATTern:SYNC:PSMode?</code></td>
<td>A</td>
</tr>
<tr>
<td><code>:SENSe:PATTern:SYNC:FLENght &lt;numeric&gt;</code></td>
<td>A</td>
</tr>
<tr>
<td><code>:SENSe:PATTern:SYNC:FLENght?</code></td>
<td>A</td>
</tr>
<tr>
<td><code>:SENSe:PATTern:DATA:WHOLOe &lt;start&gt;,&lt;end&gt;,&lt;data&gt;</code></td>
<td>A</td>
</tr>
<tr>
<td><code>:SENSe:PATTern:DATA:WHOLOe? &lt;start&gt;</code></td>
<td>A</td>
</tr>
<tr>
<td><code>:SENSe:PATTern:BDATa:WHOLOe &lt;start&gt;,&lt;end&gt;,&lt;bdata&gt;</code></td>
<td>A</td>
</tr>
<tr>
<td><code>:SENSe:PATTern:BDATa:WHOLOe? &lt;start&gt;</code></td>
<td>A</td>
</tr>
<tr>
<td><code>:SENSe:PATTern:DREVerse:ADDRes &lt;start&gt;,&lt;end&gt;</code></td>
<td>A</td>
</tr>
<tr>
<td><code>:SENSe:PATTern:DREVerse:DELTa &lt;start&gt;,&lt;delta&gt;</code></td>
<td>A</td>
</tr>
<tr>
<td><code>:SENSe:PATTern:LOGic:PRBS &lt;mark&gt;</code></td>
<td>–</td>
</tr>
<tr>
<td><code>:SENSe:PATTern:LOGic:PRGM &lt;mark&gt;</code></td>
<td>–</td>
</tr>
<tr>
<td><code>:SENSe:MEASure:TEST?</code></td>
<td>–</td>
</tr>
<tr>
<td><code>:SENSe:MEASure:START</code></td>
<td>A</td>
</tr>
<tr>
<td><code>:SENSe:MEASure:STOP</code></td>
<td>A</td>
</tr>
<tr>
<td><code>:SENSe:MEASure:MREStart &lt;boolean&gt;</code></td>
<td>A</td>
</tr>
<tr>
<td><code>:SENSe:MEASure:MREStart?</code></td>
<td>A</td>
</tr>
<tr>
<td><code>:SENSe:MEASure:EAAlarm:MODE &lt;mode&gt;</code></td>
<td>A</td>
</tr>
<tr>
<td><code>:SENSe:MEASure:EAAlarm:MODE?</code></td>
<td>A</td>
</tr>
<tr>
<td><code>:SENSe:MEASure:EAAlarm:PERiod &lt;d&gt;,&lt;h&gt;,&lt;m&gt;,&lt;s&gt;</code></td>
<td>A</td>
</tr>
<tr>
<td><code>:SENSe:MEASure:EAAlarm:PERiod?</code></td>
<td>A</td>
</tr>
<tr>
<td><code>:SENSe:MEASure:EAAlarm:BTIMe:SET &lt;boolean&gt;</code></td>
<td>–</td>
</tr>
<tr>
<td><code>:SENSe:MEASure:EAAlarm:BTIMe:STARt &lt;y&gt;,&lt;m&gt;,&lt;d&gt;,&lt;h&gt;,&lt;m&gt;,&lt;s&gt;</code></td>
<td>–</td>
</tr>
<tr>
<td><code>:SENSe:MEASure:EAAlarm:BTIMe:STARt?</code></td>
<td>–</td>
</tr>
<tr>
<td><code>:SENSe:MEASure:EAAlarm:ERRor:TYPE?</code></td>
<td>–</td>
</tr>
<tr>
<td><code>:SENSe:MEASure:EAAlarm:MASK:ROUTE &lt;route&gt;,&lt;boolean&gt;</code></td>
<td>–</td>
</tr>
<tr>
<td><code>:SENSe:MEASure:EAAlarm:STARt?</code></td>
<td>A</td>
</tr>
<tr>
<td><code>:SENSe:MEASure:EAAlarm:STOP?</code></td>
<td>A</td>
</tr>
<tr>
<td><code>:SENSe:MEASure:EAAlarm:STATE?</code></td>
<td>A</td>
</tr>
<tr>
<td><code>:SENSe:MEASure:EAAlarm:ELAPsed?</code></td>
<td>A</td>
</tr>
<tr>
<td><code>:SENSe:MEASure:EAAlarm:TIMEd?</code></td>
<td>A</td>
</tr>
<tr>
<td><code>:SENSe:MEASure:EAAlarm:ITIME?</code></td>
<td>A</td>
</tr>
<tr>
<td><code>:SENSe:MEASure:EAAlarm:AOCCur? &lt;alarm&gt;</code></td>
<td>A</td>
</tr>
<tr>
<td><code>:SENSe:MEASure:EAAlarm:ARECver? &lt;alarm&gt;</code></td>
<td>A</td>
</tr>
</tbody>
</table>
### Table A.1.1-9 MP1632C SENSe Subsystem (Cont’d)

<table>
<thead>
<tr>
<th>Commands</th>
<th>Compatibility</th>
</tr>
</thead>
<tbody>
<tr>
<td>:SENSe:MEASure:EMARgin:MODE &lt;mode&gt;</td>
<td>—</td>
</tr>
<tr>
<td>:SENSe:MEASure:EMARgin:MODE?</td>
<td>—</td>
</tr>
<tr>
<td>:SENSe:MEASure:EMARgin:TYPE?</td>
<td>—</td>
</tr>
<tr>
<td>:SENSe:MEASure:EMARgin:TYPE &lt;type&gt;</td>
<td>—</td>
</tr>
<tr>
<td>:SENSe:MEASure:EMARgin:MARGin:THReshold &lt;thre&gt;</td>
<td>A</td>
</tr>
<tr>
<td>:SENSe:MEASure:EMARgin:MARGin:THReshold?</td>
<td>A</td>
</tr>
<tr>
<td>:SENSe:MEASure:EMARgin:MARGin:RESolution &lt;type&gt;</td>
<td>—</td>
</tr>
<tr>
<td>:SENSe:MEASure:EMARgin:MARGin:POINt? &lt;thre&gt;</td>
<td>—</td>
</tr>
<tr>
<td>:SENSe:MEASure:EMARgin:STATe?</td>
<td>—</td>
</tr>
<tr>
<td>:SENSe:MEASure:EMARgin:STATe?</td>
<td>—</td>
</tr>
<tr>
<td>:SENSe:MEASure:EMARgin:STOP?</td>
<td>—</td>
</tr>
<tr>
<td>:SENSe:MEASure:EMARgin:ELAPsed?</td>
<td>—</td>
</tr>
<tr>
<td>:SENSe:MEASure:ASEarch:MODE &lt;mode&gt;</td>
<td>B</td>
</tr>
<tr>
<td>:SENSe:MEASure:ASEarch:MODE?</td>
<td>B</td>
</tr>
<tr>
<td>:SENSe:MEASure:ASEarch:STATe?</td>
<td>A</td>
</tr>
<tr>
<td>:SENSe:MEASure:ASEarch:STATe?</td>
<td>A</td>
</tr>
<tr>
<td>:SENSe:MEASure:ASEarch:PATTern &lt;boolean&gt;</td>
<td>—</td>
</tr>
<tr>
<td>:SENSe:MEASure:ASEarch:PATTern?</td>
<td>—</td>
</tr>
<tr>
<td>:SENSe:MEASure:ASEarch:PATTern?</td>
<td>A</td>
</tr>
</tbody>
</table>

### Table A.1.1-10 MP1632C INPut Subsystem

<table>
<thead>
<tr>
<th>Commands</th>
<th>Compatibility</th>
</tr>
</thead>
<tbody>
<tr>
<td>:INPut:CLOCk:POLarity &lt;pol&gt;</td>
<td>—</td>
</tr>
<tr>
<td>:INPut:CLOCk:POLarity?</td>
<td>—</td>
</tr>
<tr>
<td>:INPut:CLOCk:DELay &lt;numeric&gt;</td>
<td>A</td>
</tr>
<tr>
<td>:INPut:CLOCk:DELay?</td>
<td>A</td>
</tr>
<tr>
<td>:INPut:CLOCk:TERMination &lt;term&gt;</td>
<td>A</td>
</tr>
<tr>
<td>:INPut:CLOCk:TERMination?</td>
<td>A</td>
</tr>
<tr>
<td>:INPut:DATA:LEVel &lt;level&gt;</td>
<td>C</td>
</tr>
<tr>
<td>:INPut:DATA:LEVel?</td>
<td>C</td>
</tr>
<tr>
<td>:INPut:DATA:THReshold &lt;numeric&gt;</td>
<td>B</td>
</tr>
<tr>
<td>:INPut:DATA:THReshold?</td>
<td>B</td>
</tr>
<tr>
<td>:INPut:DATA:TERMination &lt;term&gt;</td>
<td>C</td>
</tr>
<tr>
<td>:INPut:DATA:TERMination?</td>
<td>C</td>
</tr>
</tbody>
</table>
## Remote Command Compatibility

### Table A.1.1-11  MP1632C CALCulate Subsystem

<table>
<thead>
<tr>
<th>Commands</th>
<th>Compatibility</th>
</tr>
</thead>
<tbody>
<tr>
<td>:CALCulate:EAAlarm:CLEVELuation &lt;boolean&gt;</td>
<td>A</td>
</tr>
<tr>
<td>:CALCulate:EAAlarm:CLEVELuation?</td>
<td>A</td>
</tr>
<tr>
<td>:CALCulate:EAAlarm:SLEVELuation &lt;boolean&gt;</td>
<td>A</td>
</tr>
<tr>
<td>:CALCulate:EAAlarm:SLEVELuation?</td>
<td>A</td>
</tr>
<tr>
<td>:CALCulate:EAAlarm:ERROR:INTerval &lt;numeric&gt;,&lt;suffix&gt;</td>
<td>A</td>
</tr>
<tr>
<td>:CALCulate:EAAlarm:ERROR:INTerval?</td>
<td>A</td>
</tr>
<tr>
<td>:CALCulate:EAAlarm:PERformance:MEASurement &lt;boolean&gt;</td>
<td>–</td>
</tr>
<tr>
<td>:CALCulate:EAAlarm:PERformance:THReshold &lt;thre&gt;</td>
<td>A</td>
</tr>
<tr>
<td>:CALCulate:EAAlarm:PERformance:THReshold?</td>
<td>A</td>
</tr>
<tr>
<td>:CALCulate:EAAlarm:TEINterval:MEASurement &lt;boolean&gt;</td>
<td>–</td>
</tr>
<tr>
<td>:CALCulate:EAAlarm:PFAil &lt;boolean&gt;</td>
<td>–</td>
</tr>
<tr>
<td>:CALCulate:EAAlarm:PFAil?</td>
<td>–</td>
</tr>
<tr>
<td>:CALCulate:DATA:EAAlarm? &lt;string&gt;</td>
<td>B</td>
</tr>
<tr>
<td>:CALCulate:DATA:EMARgin? &lt;string&gt;</td>
<td>–</td>
</tr>
<tr>
<td>:CALCulate:DATA:STORe &lt;string&gt;</td>
<td>–</td>
</tr>
<tr>
<td>:CALCulate:DATA:CLEAR &lt;string&gt;</td>
<td>–</td>
</tr>
<tr>
<td>:CALCulate:DATA:MONitor? &lt;item&gt;</td>
<td>B</td>
</tr>
<tr>
<td>:CALCulate:DATA:MONitor? &lt;string&gt;</td>
<td>A</td>
</tr>
</tbody>
</table>
## Appendix A  Command Compatibility List

### Table A.1.1-12  MP1632C DISPlay Subsystem

<table>
<thead>
<tr>
<th>Commands</th>
<th>Compatibility</th>
</tr>
</thead>
<tbody>
<tr>
<td>:DISPlay:WINDow:OPEN &lt;disp&gt;</td>
<td>–</td>
</tr>
<tr>
<td>:DISPlay:WINDow:CLOSE &lt;disp&gt;</td>
<td>–</td>
</tr>
<tr>
<td>:DISPlay:SYSTem[:NAME] &lt;name&gt;</td>
<td>–</td>
</tr>
<tr>
<td>:DISPlay:SYSTem[:NAME]?</td>
<td>–</td>
</tr>
<tr>
<td>:DISPlay:SETup[:NAME] &lt;name&gt;</td>
<td>–</td>
</tr>
<tr>
<td>:DISPlay:SETup[:NAME]?</td>
<td>–</td>
</tr>
<tr>
<td>:DISPlay:TEST[:NAME] &lt;name&gt;</td>
<td>–</td>
</tr>
<tr>
<td>:DISPlay:TEST[:NAME]?</td>
<td>–</td>
</tr>
<tr>
<td>:DISPlay:RESult:TIME &lt;time&gt;</td>
<td>A</td>
</tr>
<tr>
<td>:DISPlay:RESult:TIME?</td>
<td>A</td>
</tr>
<tr>
<td>:DISPlay:RESult:EALarm[:NAME] &lt;name&gt;</td>
<td>–</td>
</tr>
<tr>
<td>:DISPlay:RESult:EALarm[:NAME]?</td>
<td>–</td>
</tr>
<tr>
<td>:DISPlay:RESult:EALarm:MODE &lt;mode&gt;</td>
<td>C</td>
</tr>
<tr>
<td>:DISPlay:RESult:EALarm:MODE?</td>
<td>C</td>
</tr>
<tr>
<td>:DISPlay:RESult:EALarm:ALL:PTYPE &lt;ptype&gt;</td>
<td>–</td>
</tr>
</tbody>
</table>
Table A.1.1-12  MP1632C DISPlay Subsystem (Cont’d)

<table>
<thead>
<tr>
<th>Commands</th>
<th>Compatibility</th>
</tr>
</thead>
<tbody>
<tr>
<td>:DISPlay:RESult:EALarm:ZOOM1</td>
<td>–</td>
</tr>
<tr>
<td>:DISPlay:RESult:EALarm:ZOOM2</td>
<td>–</td>
</tr>
<tr>
<td>:DISPlay:RESult:EALarm:ZOOM3</td>
<td>–</td>
</tr>
<tr>
<td>:DISPlay:RESult:EALarm:ZOOM4</td>
<td>–</td>
</tr>
<tr>
<td>:DISPlay:RESult:EALarm:ZOOM1</td>
<td>–</td>
</tr>
<tr>
<td>:DISPlay:RESult:EALarm:ZOOM2</td>
<td>–</td>
</tr>
<tr>
<td>:DISPlay:RESult:EALarm:ZOOM3</td>
<td>–</td>
</tr>
<tr>
<td>:DISPlay:RESult:EALarm:ZOOM4</td>
<td>–</td>
</tr>
<tr>
<td>:DISPlay:RESult:EALarm:HRESet</td>
<td>A</td>
</tr>
<tr>
<td>:DISPlay:RESult:EMARgin:ERATe &lt;erate&gt;,&lt;boolean&gt;</td>
<td>–</td>
</tr>
<tr>
<td>:DISPlay:RESult:EMARgin:SCALE:VOLTage &lt;min&gt;,&lt;step&gt;</td>
<td>–</td>
</tr>
<tr>
<td>:DISPlay:RESult:EMARgin:SCALE:PHASe &lt;min&gt;,&lt;step&gt;</td>
<td>–</td>
</tr>
<tr>
<td>:DISPlay:RESult:EMARgin:MARKer:MODE &lt;mode&gt;</td>
<td>–</td>
</tr>
<tr>
<td>:DISPlay:RESult:EMARgin:MARKer:ERATe &lt;erate&gt;</td>
<td>–</td>
</tr>
<tr>
<td>:DISPlay:RESult:EMARgin:MARKer:MOVE &lt;marker&gt;,&lt;mode&gt;</td>
<td>–</td>
</tr>
<tr>
<td>:DISPlay:RESult:EMARgin:TEMPlate:SELect &lt;select&gt;,&lt;boolean&gt;</td>
<td>–</td>
</tr>
<tr>
<td>:DISPlay:RESult:EMARgin:TEMPlate:MOVE &lt;template&gt;,&lt;mode&gt;</td>
<td>–</td>
</tr>
<tr>
<td>:DISPlay:RESult:EMARgin:TEMPlate:PNUMber &lt;template&gt;,&lt;point&gt;</td>
<td>–</td>
</tr>
<tr>
<td>:DISPlay:RESult:EMARgin:TEMPlate:POINt:VOLTage &lt;template&gt;,&lt;point&gt;,&lt;voltage&gt;</td>
<td>–</td>
</tr>
<tr>
<td>:DISPlay:RESult:EMARgin:TEMPlate:POINt:PHASe &lt;template&gt;,&lt;point&gt;,&lt;phase&gt;</td>
<td>–</td>
</tr>
<tr>
<td>:DISPlay:RESult:EMARgin:TEMPlate:CONNect &lt;template&gt;,&lt;boolean&gt;</td>
<td>–</td>
</tr>
<tr>
<td>:DISPlay:RESult:CUSTomize:SETup1</td>
<td>SETup2</td>
</tr>
<tr>
<td>:DISPlay:RESult:CUSTomize:SETup1</td>
<td>SETup2</td>
</tr>
<tr>
<td>:DISPlay:RESult:CUSTomize:SETup1</td>
<td>SETup2</td>
</tr>
<tr>
<td>:DISPlay:RESult:CUSTomize:SETup1</td>
<td>SETup2</td>
</tr>
<tr>
<td>:DISPlay:RESult:CUSTomize:SETup1</td>
<td>SETup2</td>
</tr>
</tbody>
</table>
### Appendix A  Command Compatibility List

#### Table A.1.1-12 MP1632C DISPlay Subsystem (Cont’d)

<table>
<thead>
<tr>
<th>Commands</th>
<th>Compatibility</th>
</tr>
</thead>
<tbody>
<tr>
<td>:DISPlay:RESult:CUSTomize:SETup1</td>
<td>–</td>
</tr>
<tr>
<td>:DISPlay:RESult:CUSTomize:SETup2</td>
<td>–</td>
</tr>
<tr>
<td>:DISPlay:RESult:CUSTomize:SETup3</td>
<td>–</td>
</tr>
<tr>
<td>:DISPlay:RESult:CUSTomize:SETup4</td>
<td>–</td>
</tr>
<tr>
<td>:DISPlay:RESult:CUSTomize:SETup5</td>
<td>–</td>
</tr>
<tr>
<td>:DISPlay:RESult:CUSTomize:PATTern:UNIT &lt;unit&gt;</td>
<td>–</td>
</tr>
<tr>
<td>:DISPlay:RESult:CUSTomize:PATTern:OFFSet &lt;numeric&gt;</td>
<td>–</td>
</tr>
<tr>
<td>:DISPlay:RESult:CUSTomize:RESult:ITEM &lt;item&gt;</td>
<td>–</td>
</tr>
</tbody>
</table>

#### Table A.1.1-13 MP1632C SYSTem Subsystem

<table>
<thead>
<tr>
<th>Commands</th>
<th>Compatibility</th>
</tr>
</thead>
<tbody>
<tr>
<td>:SYSTem:BEEPer:ERRor:SET &lt;boolean&gt;</td>
<td>A</td>
</tr>
<tr>
<td>:SYSTem:BEEPer:ERRor:SET?</td>
<td>A</td>
</tr>
<tr>
<td>:SYSTem:BEEPer:ALARm:SET &lt;boolean&gt;</td>
<td>A</td>
</tr>
<tr>
<td>:SYSTem:BEEPer:ALARm:SET?</td>
<td>A</td>
</tr>
<tr>
<td>:SYSTem:BEEPer:SYSTem:SET &lt;boolean&gt;</td>
<td>A</td>
</tr>
<tr>
<td>:SYSTem:BEEPer:SYSTem:SET?</td>
<td>A</td>
</tr>
<tr>
<td>:SYSTem:BEEPer:SYSTem:TYPE &lt;type&gt;,&lt;boolean&gt;</td>
<td>B</td>
</tr>
<tr>
<td>:SYSTem:BEEPer:SYSTem:TYPE?</td>
<td>A</td>
</tr>
<tr>
<td>:SYSTem:MODE?</td>
<td>–</td>
</tr>
<tr>
<td>:SYSTem:DATE &lt;year&gt;,&lt;month&gt;,&lt;day&gt;</td>
<td>–</td>
</tr>
<tr>
<td>:SYSTem:DATE?</td>
<td>–</td>
</tr>
<tr>
<td>:SYSTem:TIME &lt;hour&gt;,&lt;min&gt;,&lt;sec&gt;</td>
<td>–</td>
</tr>
<tr>
<td>:SYSTem:TIME?</td>
<td>–</td>
</tr>
<tr>
<td>:SYSTem:BSIZe? &lt;mode&gt;</td>
<td>–</td>
</tr>
<tr>
<td>:SYSTem:ERRor?</td>
<td>A</td>
</tr>
<tr>
<td>:SYSTem:VERSion?</td>
<td>A</td>
</tr>
<tr>
<td>:SYSTem:ORGanization:HARDware?</td>
<td>C</td>
</tr>
<tr>
<td>:SYSTem:ORGanization:SOFTware?</td>
<td>–</td>
</tr>
<tr>
<td>:SYSTem:MMEMemory:RECall &lt;type&gt;,&lt;file_name&gt;,&lt;unit&gt;</td>
<td>C</td>
</tr>
<tr>
<td>:SYSTem:MMEMemory:RECall &quot;EYE:TEMPlate1&quot;, &lt;file_name&gt;,0</td>
<td>–</td>
</tr>
<tr>
<td>:SYSTem:MMEMemory:STORE &lt;type&gt;,&lt;file_name&gt;,&lt;unit&gt;</td>
<td>C</td>
</tr>
<tr>
<td>:SYSTem:MMEMemory:STORE &quot;EYE:TEMPlate1&quot;, &lt;file_name&gt;,0</td>
<td>–</td>
</tr>
<tr>
<td>:SYSTem:MMEMemory:QRECall &lt;file_name&gt;</td>
<td>C</td>
</tr>
<tr>
<td>:SYSTem:MMEMemory:QSTore &lt;file_name&gt;,&lt;comment&gt;</td>
<td>C</td>
</tr>
<tr>
<td>:SYSTem:MMEMemory:CATalog? &lt;drv_dir&gt;,&lt;type&gt;</td>
<td>–</td>
</tr>
<tr>
<td>:SYSTem:MMEMemory:INITialize</td>
<td>A</td>
</tr>
</tbody>
</table>
### A.1 Remote Command Compatibility

#### Table A.1.1-13 MP1632C SYStem Subsystem (Cont’d)

<table>
<thead>
<tr>
<th>Commands</th>
<th>Compatibility</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>:SYSTem:PRINt:COPY</code></td>
<td>A</td>
</tr>
<tr>
<td><code>:SYSTem:PRINt:TEXT &lt;string&gt;</code></td>
<td>–</td>
</tr>
<tr>
<td><code>:SYSTem:PRINt:EALarm &lt;item&gt;</code></td>
<td>C</td>
</tr>
<tr>
<td><code>:SYSTem:PRINt:PATTern:TABLE:ADDRess &lt;unit&gt;,&lt;form&gt;,&lt;start&gt;,&lt;end&gt;</code></td>
<td>–</td>
</tr>
<tr>
<td><code>:SYSTem:PRINt:PATTern:TABLE:ALL &lt;unit&gt;,&lt;form&gt;</code></td>
<td>–</td>
</tr>
<tr>
<td><code>:SYSTem:PRINt:CANCel</code></td>
<td>–</td>
</tr>
<tr>
<td><code>:SYSTem:PRINt:EMARgin &lt;boolean&gt;</code></td>
<td>–</td>
</tr>
<tr>
<td><code>:SYSTem:PRINt:REMARgin &lt;boolean&gt;</code></td>
<td>–</td>
</tr>
<tr>
<td><code>:SYSTem:TERMination</code></td>
<td>A</td>
</tr>
<tr>
<td><code>:SYSTem:TERMination?</code></td>
<td>A</td>
</tr>
</tbody>
</table>

#### Table A.1.1-14 MP1632C STATus Subsystem

<table>
<thead>
<tr>
<th>Commands</th>
<th>Compatibility</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>:STATus:PRESet</code></td>
<td>A</td>
</tr>
<tr>
<td><code>:STATus:OPERation[:,EVENt]?</code></td>
<td>C</td>
</tr>
<tr>
<td><code>:STATus:OPERation:CONDition?</code></td>
<td>C</td>
</tr>
<tr>
<td><code>:STATus:OPERation:ENABle &lt;numeric&gt;</code></td>
<td>C</td>
</tr>
<tr>
<td><code>:STATus:OPERation:ENABle?</code></td>
<td>C</td>
</tr>
<tr>
<td><code>:STATus:OPERation:PTRansition &lt;numeric&gt;</code></td>
<td>C</td>
</tr>
<tr>
<td><code>:STATus:OPERation:PTRansition?</code></td>
<td>C</td>
</tr>
<tr>
<td><code>:STATus:OPERation:NTRansition &lt;numeric&gt;</code></td>
<td>C</td>
</tr>
<tr>
<td><code>:STATus:OPERation:NTRansition?</code></td>
<td>C</td>
</tr>
<tr>
<td><code>:STATus:OPERation:INSTrument[:,EVENt]?</code></td>
<td>–</td>
</tr>
<tr>
<td><code>:STATus:OPERation:INSTrument:ENABle &lt;numeric&gt;</code></td>
<td>–</td>
</tr>
<tr>
<td><code>:STATus:OPERation:INSTrument:PTRansition &lt;numeric&gt;</code></td>
<td>–</td>
</tr>
<tr>
<td><code>:STATus:OPERation:INSTrument:NTRansition &lt;numeric&gt;</code></td>
<td>–</td>
</tr>
<tr>
<td><code>:STATus:QUEStionable[:,EVENt]?</code></td>
<td>–</td>
</tr>
<tr>
<td><code>:STATus:QUEStionable:CONDition?</code></td>
<td>–</td>
</tr>
<tr>
<td><code>:STATus:QUEStionable:ENABle &lt;numeric&gt;</code></td>
<td>–</td>
</tr>
<tr>
<td><code>:STATus:QUEStionable:ENABle?</code></td>
<td>–</td>
</tr>
</tbody>
</table>
### Appendix A  Command Compatibility List

#### Table A.1.1-14  MP1632C STATus Subsystem (Cont’d)

<table>
<thead>
<tr>
<th>Commands</th>
<th>Compatibility</th>
</tr>
</thead>
<tbody>
<tr>
<td>:STATus:QUEStionable:PTRansition &lt;numeric&gt;</td>
<td>–</td>
</tr>
<tr>
<td>:STATus:QUEStionable:PTRansition?</td>
<td>–</td>
</tr>
<tr>
<td>:STATus:QUEStionable:NTRansition &lt;numeric&gt;</td>
<td>–</td>
</tr>
<tr>
<td>:STATus:QUEStionable:NTRansition?</td>
<td>–</td>
</tr>
<tr>
<td>:STATus:QUEStionable:MONitor[:EVENt]?</td>
<td>–</td>
</tr>
<tr>
<td>:STATus:QUEStionable:MONitor:ENABle &lt;numeric&gt;</td>
<td>–</td>
</tr>
<tr>
<td>:STATus:QUEStionable:MONitor:ENABle?</td>
<td>–</td>
</tr>
<tr>
<td>:STATus:QUEStionable:MONitor:PTRansition &lt;numeric&gt;</td>
<td>–</td>
</tr>
<tr>
<td>:STATus:QUEStionable:MONitor:NTRansition &lt;numeric&gt;</td>
<td>–</td>
</tr>
</tbody>
</table>
A.1.2 Compatibility with MP1761A/B/C

This section describes the compatibility between the MP1800A or the MT1810A commands and the MP1761A/B/C Pulse Pattern Generator.

<table>
<thead>
<tr>
<th>Section</th>
<th>Function</th>
<th>Control Message</th>
<th>Data Request Message</th>
<th>Compatibility</th>
</tr>
</thead>
<tbody>
<tr>
<td>INTERNAL CLOCK</td>
<td>Internal clock frequency</td>
<td>FRQ</td>
<td>FRQ?</td>
<td>B</td>
</tr>
<tr>
<td></td>
<td>Internal clock resolution switching</td>
<td>RES</td>
<td>RES?</td>
<td>A</td>
</tr>
<tr>
<td>MEMORY</td>
<td>File No./Directory mode switching</td>
<td>FIL</td>
<td>FIL?</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>FD data recall</td>
<td>RCL</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>FD data delete</td>
<td>DEL</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>FD data save</td>
<td>SAV</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>FD data resave</td>
<td>RSV</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>Memory mode switch</td>
<td>MEM</td>
<td>MEM?</td>
<td>C</td>
</tr>
<tr>
<td></td>
<td>FD format</td>
<td>FDF</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>File contents search</td>
<td>–</td>
<td>FSH?</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>Memory FD mode</td>
<td>–</td>
<td>FMD?</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>FD access status</td>
<td>–</td>
<td>MAC?</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>FD error message</td>
<td>–</td>
<td>FDE?</td>
<td>–</td>
</tr>
<tr>
<td>PATTERN</td>
<td>Pattern logic</td>
<td>LGC</td>
<td>LGC?</td>
<td>C</td>
</tr>
<tr>
<td></td>
<td>Generation pattern switch</td>
<td>PTS</td>
<td>PTS?</td>
<td>B</td>
</tr>
<tr>
<td></td>
<td>Zero Subst/PRBS stage</td>
<td>PTN</td>
<td>PTN?</td>
<td>C</td>
</tr>
<tr>
<td></td>
<td>PRBS mark ratio</td>
<td>MRK</td>
<td>MRK?</td>
<td>C</td>
</tr>
<tr>
<td></td>
<td>Alternate A/B display switch</td>
<td>ALT</td>
<td>ALT?</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>Error insertion</td>
<td>EAD</td>
<td>EAD?</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>Alternate A/B loop times</td>
<td>LPT</td>
<td>LPT?</td>
<td>C</td>
</tr>
<tr>
<td></td>
<td>Data length</td>
<td>DLN</td>
<td>DLN?</td>
<td>C</td>
</tr>
<tr>
<td></td>
<td>Zero Subst length</td>
<td>ZLN</td>
<td>ZLN?</td>
<td>C</td>
</tr>
<tr>
<td></td>
<td>Page number</td>
<td>ADR</td>
<td>ADR?</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>Pattern bit</td>
<td>BIT</td>
<td>BIT?</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>Pattern data preset (All pages ,All bits)</td>
<td>ALL</td>
<td>–</td>
<td>C</td>
</tr>
<tr>
<td></td>
<td>Pattern data preset (1 page ,All bits)</td>
<td>PST</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>Pattern Sync trigger position</td>
<td>PSP</td>
<td>PSP?</td>
<td>C</td>
</tr>
</tbody>
</table>
### Table A.1.2-1 MP1761A/B/C device message (Cont’d)

<table>
<thead>
<tr>
<th>Section</th>
<th>Function</th>
<th>Control Message</th>
<th>Data Request Message</th>
<th>Compatibility</th>
</tr>
</thead>
<tbody>
<tr>
<td>PATTERN (Cont’d)</td>
<td>Page No./Pattern sync trigger position display switch</td>
<td>PPD</td>
<td>PPD?</td>
<td>–</td>
</tr>
<tr>
<td>OUTPUT</td>
<td>Data output termination</td>
<td>DTM</td>
<td>DTM?</td>
<td>A</td>
</tr>
<tr>
<td></td>
<td>Clock1 output termination</td>
<td>CTM</td>
<td>CTM?</td>
<td>A</td>
</tr>
<tr>
<td></td>
<td>Offset reference value</td>
<td>OFS</td>
<td>OFS?</td>
<td>C</td>
</tr>
<tr>
<td></td>
<td>Data output amplitude</td>
<td>DAP</td>
<td>DAP?</td>
<td>C</td>
</tr>
<tr>
<td></td>
<td>XData output amplitude</td>
<td>NAP</td>
<td>NAP?</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>Data output offset</td>
<td>DOS</td>
<td>DOS?</td>
<td>C</td>
</tr>
<tr>
<td></td>
<td>XData output offset</td>
<td>NOS</td>
<td>NOS?</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>Clock1 output amplitude</td>
<td>CAP</td>
<td>CAP?</td>
<td>C</td>
</tr>
<tr>
<td></td>
<td>Clock1 output delay time</td>
<td>CDL</td>
<td>CDL?</td>
<td>A</td>
</tr>
<tr>
<td></td>
<td>Clock1 output offset</td>
<td>COS</td>
<td>COS?</td>
<td>B</td>
</tr>
<tr>
<td></td>
<td>Output ON/OFF</td>
<td>OON</td>
<td>OON?</td>
<td>C</td>
</tr>
<tr>
<td></td>
<td>Data/XData display switch</td>
<td>DDS</td>
<td>DDS?</td>
<td>A</td>
</tr>
<tr>
<td></td>
<td>Data/XData tracking</td>
<td>TRK</td>
<td>TRK?</td>
<td>C</td>
</tr>
<tr>
<td></td>
<td>1/1 SPEED, 1/4 SPEED switch</td>
<td>SPD</td>
<td>SPD?</td>
<td>–</td>
</tr>
<tr>
<td>Front panel</td>
<td>Sync signal output selection</td>
<td>SOP</td>
<td>SOP?</td>
<td>C</td>
</tr>
<tr>
<td>Rear panel</td>
<td>Error insertion channel</td>
<td>ECH</td>
<td>ECH?</td>
<td>C</td>
</tr>
<tr>
<td>Function switch</td>
<td>Mark ratio and bit shift number</td>
<td>SFT</td>
<td>SFT?</td>
<td>C</td>
</tr>
<tr>
<td></td>
<td>External error insertion</td>
<td>EEI</td>
<td>EEI?</td>
<td>C</td>
</tr>
<tr>
<td></td>
<td>Alternate pattern A/B switch signal selection</td>
<td>APS</td>
<td>APS?</td>
<td>C</td>
</tr>
<tr>
<td>Other</td>
<td>Initialize</td>
<td>INI</td>
<td>–</td>
<td>A</td>
</tr>
<tr>
<td></td>
<td>Pattern data input byte number</td>
<td>WRT</td>
<td>–</td>
<td>A</td>
</tr>
<tr>
<td></td>
<td>Pattern data output byte number</td>
<td>–</td>
<td>RED?</td>
<td>A</td>
</tr>
<tr>
<td></td>
<td>Internal timer setting</td>
<td>RTM</td>
<td>RTM?</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>Internal synthesizer PLL</td>
<td>–</td>
<td>PLL?</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>Power cut, recovery status</td>
<td>–</td>
<td>PWI?</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>Delay status</td>
<td>–</td>
<td>DLY?</td>
<td>A</td>
</tr>
<tr>
<td></td>
<td>Termination code selection</td>
<td>TRM</td>
<td>TRM?</td>
<td>A</td>
</tr>
</tbody>
</table>
A.1 Remote Command Compatibility

A.1.3 Compatibility with MP1762A/C/D

This section describes the compatibility between the MP1800A or the MT1810A commands and the MP1762A/C/D Error Detector.

<table>
<thead>
<tr>
<th>Section</th>
<th>Function</th>
<th>Control Message</th>
<th>Data Request Message</th>
<th>Compatibility</th>
</tr>
</thead>
<tbody>
<tr>
<td>INPUT</td>
<td>Data input threshold voltage</td>
<td>DTH</td>
<td>DTH?</td>
<td>A</td>
</tr>
<tr>
<td></td>
<td>Eye margin measurement result (Threshold)</td>
<td>–</td>
<td>THM?</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>Eye margin measurement result (Error ratio)</td>
<td>EYT</td>
<td>EYT?</td>
<td>C</td>
</tr>
<tr>
<td></td>
<td>Eye margin measurement start</td>
<td>EST</td>
<td>EST?</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>Eye margin measurement display</td>
<td>EME</td>
<td>EME?</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>Clock input phase</td>
<td>CPA</td>
<td>CPA?</td>
<td>B</td>
</tr>
<tr>
<td></td>
<td>Clock input polarity</td>
<td>CPL</td>
<td>CPL?</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>Clock input termination voltage</td>
<td>CTM</td>
<td>CTM?</td>
<td>A</td>
</tr>
<tr>
<td></td>
<td>Eye margin measurement result (Phase)</td>
<td>–</td>
<td>PHM?</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>Data/XData tracking</td>
<td>DAC</td>
<td>DCD?</td>
<td>B</td>
</tr>
<tr>
<td></td>
<td>Data input termination condition</td>
<td>DTC</td>
<td>DTC?</td>
<td>B</td>
</tr>
<tr>
<td></td>
<td>Recovery frequency</td>
<td>RFQ</td>
<td>DRC?</td>
<td>C</td>
</tr>
<tr>
<td></td>
<td>PLL reset</td>
<td>RPL</td>
<td>RPL?</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>Input data select</td>
<td>DSL</td>
<td>DCD?</td>
<td>A</td>
</tr>
<tr>
<td></td>
<td>Clock select</td>
<td>RRC</td>
<td>DRC?</td>
<td>A</td>
</tr>
<tr>
<td></td>
<td>Single-ended/Differential select</td>
<td>DSD</td>
<td>DCD?</td>
<td>B</td>
</tr>
<tr>
<td></td>
<td>Delay status</td>
<td>–</td>
<td>DLY?</td>
<td>A</td>
</tr>
<tr>
<td></td>
<td>Data input termination voltage</td>
<td>DTM</td>
<td>DTM?</td>
<td>B</td>
</tr>
<tr>
<td></td>
<td>Automatic phase threshold search</td>
<td>SRH</td>
<td>SRH?</td>
<td>–</td>
</tr>
<tr>
<td>MEMORY</td>
<td>File contents search</td>
<td>–</td>
<td>FSH?</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>FD data delete</td>
<td>DEL</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>FD error message</td>
<td>–</td>
<td>FDE?</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>FD format</td>
<td>FDF</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>FD data recall</td>
<td>RCL</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>FD access status</td>
<td>–</td>
<td>MAC?</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>File No./Directory mode switching</td>
<td>FIL</td>
<td>FIL?</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>FD data save</td>
<td>SAV</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>Memory FD mode</td>
<td>–</td>
<td>FMD?</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>FD data resave</td>
<td>RSV</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>Memory mode switch</td>
<td>MEM</td>
<td>MEM?</td>
<td>–</td>
</tr>
</tbody>
</table>
### Table A.1.3-1 MP1762A/C/D device message (Cont’d)

<table>
<thead>
<tr>
<th>Section</th>
<th>Function</th>
<th>Control Message</th>
<th>Data Request Message</th>
<th>Compatibility</th>
</tr>
</thead>
<tbody>
<tr>
<td>PATTERN</td>
<td>Pattern logic</td>
<td>LGC</td>
<td>LGC?</td>
<td>A</td>
</tr>
<tr>
<td></td>
<td>Generation pattern switch</td>
<td>PTS</td>
<td>PTS?</td>
<td>A</td>
</tr>
<tr>
<td></td>
<td>Zero Subst/PRBS stage</td>
<td>PTN</td>
<td>PTN?</td>
<td>A</td>
</tr>
<tr>
<td></td>
<td>PRBS mark ratio</td>
<td>MRK</td>
<td>MRK?</td>
<td>A</td>
</tr>
<tr>
<td></td>
<td>Synchronous method</td>
<td>SYM</td>
<td>SYM?</td>
<td>A</td>
</tr>
<tr>
<td></td>
<td>Display selection</td>
<td>DSP</td>
<td>DSP?</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>Page number</td>
<td>PAG</td>
<td>PAG?</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>Pattern bit</td>
<td>BIT</td>
<td>BIT?</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>Frame length</td>
<td>FLN</td>
<td>FLN?</td>
<td>A</td>
</tr>
<tr>
<td></td>
<td>Bit window pattern</td>
<td>CHM</td>
<td>CHM?</td>
<td>C</td>
</tr>
<tr>
<td></td>
<td>Alternate A/B display switching</td>
<td>ALT</td>
<td>ALT?</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>Alternate pattern A/B switch</td>
<td>APS</td>
<td>APS?</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>Error analysis data</td>
<td>–</td>
<td>EAB?</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>Error analysis page</td>
<td>EAP</td>
<td>EAP?</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>Error analysis trigger</td>
<td>EAT</td>
<td>EAT?</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>Number of pattern data input bytes</td>
<td>WRT</td>
<td>–</td>
<td>A</td>
</tr>
<tr>
<td></td>
<td>Block window ON/OFF</td>
<td>MGE</td>
<td>MGE?</td>
<td>A</td>
</tr>
<tr>
<td></td>
<td>Block window pattern</td>
<td>MGB</td>
<td>MGB?</td>
<td>C</td>
</tr>
<tr>
<td></td>
<td>Bit window ON/OFF</td>
<td>MSE</td>
<td>MSE?</td>
<td>A</td>
</tr>
<tr>
<td></td>
<td>Bit window page</td>
<td>MSK</td>
<td>MSK?</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>Bit window preset (all pages, all bits)</td>
<td>HAL</td>
<td>–</td>
<td>C</td>
</tr>
<tr>
<td></td>
<td>Bit window preset (1 page, all bits)</td>
<td>HPS</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>Pattern data preset (1 pages, all bits)</td>
<td>PST</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>Pattern data preset (all pages, all bits)</td>
<td>ALL</td>
<td>–</td>
<td>A</td>
</tr>
<tr>
<td></td>
<td>Block window preset (1 page, all bits)</td>
<td>MPS</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>Number of bytes of block window data output</td>
<td>–</td>
<td>MRD?</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>Block window preset (all pages, all bits)</td>
<td>MAL</td>
<td>MAL?</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>Number of bytes of block window data input</td>
<td>–</td>
<td>MWT</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>Page/pattern sync trigger position display switch</td>
<td>PPD</td>
<td>PPD?</td>
<td>A</td>
</tr>
<tr>
<td></td>
<td>Pattern sync position</td>
<td>PSP</td>
<td>PSP?</td>
<td>B</td>
</tr>
<tr>
<td></td>
<td>Zero Subst length</td>
<td>ZLN</td>
<td>ZLN?</td>
<td>B</td>
</tr>
<tr>
<td></td>
<td>Number of pattern data output bytes</td>
<td>–</td>
<td>RED?</td>
<td>–</td>
</tr>
</tbody>
</table>
### Table A.1.3-1 MP1762A/C/D device message (Cont’d)

<table>
<thead>
<tr>
<th>Section</th>
<th>Function</th>
<th>Control Message</th>
<th>Data Request Message</th>
<th>Compatibility</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MEASURE-MENT</strong></td>
<td>Measurement result for clock count</td>
<td>–</td>
<td>CC?</td>
<td>A</td>
</tr>
<tr>
<td></td>
<td>Clock Loss status</td>
<td>–</td>
<td>CLI?</td>
<td>A</td>
</tr>
<tr>
<td></td>
<td>Intermediate result display</td>
<td>CUR</td>
<td>CUR?</td>
<td>A</td>
</tr>
<tr>
<td></td>
<td>Data length</td>
<td>DLN</td>
<td>DLN?</td>
<td>A</td>
</tr>
<tr>
<td></td>
<td>Measurement result display mode</td>
<td>DMS</td>
<td>DMS?</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>Measurement start/restart</td>
<td>STA</td>
<td>–</td>
<td>A</td>
</tr>
<tr>
<td></td>
<td>Measurement stop</td>
<td>STO</td>
<td>–</td>
<td>A</td>
</tr>
<tr>
<td></td>
<td>Automatic synchronous threshold</td>
<td>SYE</td>
<td>SYE?</td>
<td>A</td>
</tr>
<tr>
<td></td>
<td>Automatic synchronization</td>
<td>SYN</td>
<td>SYN?</td>
<td>A</td>
</tr>
<tr>
<td></td>
<td>Clear data from buffer</td>
<td>EDC</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>Store data in buffer</td>
<td>EDS</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>EI measurement result</td>
<td>EIT</td>
<td>EIT?</td>
<td>A</td>
</tr>
<tr>
<td></td>
<td>EI/EF interval time</td>
<td>–</td>
<td>OSD?</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>1s measurement result</td>
<td>–</td>
<td>OS?</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>Clear intermediate data</td>
<td>IMC</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>Output intermediate data</td>
<td>–</td>
<td>IMD?</td>
<td>A</td>
</tr>
<tr>
<td></td>
<td>Store intermediate data in buffer</td>
<td>IMS</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>Internal timer setting</td>
<td>RTM</td>
<td>RTM?</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>Measurement mode</td>
<td>MOD</td>
<td>MOD?</td>
<td>A</td>
</tr>
<tr>
<td></td>
<td>Measurement period setting</td>
<td>PRD</td>
<td>PRD?</td>
<td>A</td>
</tr>
<tr>
<td></td>
<td>Measurement condition</td>
<td>–</td>
<td>MSR?</td>
<td>A</td>
</tr>
<tr>
<td></td>
<td>Error detection status</td>
<td>–</td>
<td>ERS?</td>
<td>A</td>
</tr>
<tr>
<td></td>
<td>Alarm measurement result</td>
<td>–</td>
<td>AMD?</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>Error count result</td>
<td>–</td>
<td>EC?</td>
<td>A</td>
</tr>
<tr>
<td></td>
<td>Clock frequency data</td>
<td>–</td>
<td>FRQ?</td>
<td>A</td>
</tr>
<tr>
<td></td>
<td>Synchronous loss status</td>
<td>–</td>
<td>SLI?</td>
<td>A</td>
</tr>
<tr>
<td></td>
<td>Synchronous signal output</td>
<td>SOP</td>
<td>SOP?</td>
<td>A</td>
</tr>
<tr>
<td><strong>PATTERN</strong></td>
<td>Real time /measurement time display switching</td>
<td>TIM</td>
<td>TIM?</td>
<td>A</td>
</tr>
<tr>
<td></td>
<td>Measurement termination data</td>
<td>–</td>
<td>END?</td>
<td>B</td>
</tr>
<tr>
<td></td>
<td>Error rate measurement result</td>
<td>–</td>
<td>ER?</td>
<td>A</td>
</tr>
<tr>
<td><strong>Front panel</strong></td>
<td>Printer function</td>
<td>PRN</td>
<td>PRN?</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>Manual print</td>
<td>PSA</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>Alarm monitor</td>
<td>ALM</td>
<td>ALM?</td>
<td>A</td>
</tr>
<tr>
<td></td>
<td>Error monitor</td>
<td>MON</td>
<td>MON?</td>
<td>A</td>
</tr>
<tr>
<td></td>
<td>Sync output selection</td>
<td>SOP</td>
<td>SOP?</td>
<td>A</td>
</tr>
<tr>
<td><strong>Rear panel</strong></td>
<td>GPIB2 address</td>
<td>GPA</td>
<td>GPA?</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>Mark ratio and bit shift</td>
<td>SFT</td>
<td>SFT?</td>
<td>A</td>
</tr>
<tr>
<td></td>
<td>Clock loss process</td>
<td>CLS</td>
<td>CLS?</td>
<td>A</td>
</tr>
<tr>
<td></td>
<td>Sync loss process</td>
<td>SLS</td>
<td>SLS?</td>
<td>A</td>
</tr>
<tr>
<td></td>
<td>Burst measurement mode</td>
<td>BST</td>
<td>BST?</td>
<td>A</td>
</tr>
<tr>
<td></td>
<td>Error performance data threshold selection</td>
<td>ETH</td>
<td>ETH?</td>
<td>A</td>
</tr>
</tbody>
</table>
### Table A.1.3-1 MP1762A/C/D device message (Cont’d)

<table>
<thead>
<tr>
<th>Section</th>
<th>Function</th>
<th>Control Message</th>
<th>Data Request Message</th>
<th>Compatibility</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rear panel function switch (Cont’d)</td>
<td>Intermediate data calculation</td>
<td>CAL</td>
<td>CAL?</td>
<td>A</td>
</tr>
<tr>
<td></td>
<td>Error detection mode</td>
<td>ETY</td>
<td>ETY?</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>1s data print threshold</td>
<td>DOT</td>
<td>DOT?</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>1s data print</td>
<td>OSC</td>
<td>OSC?</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>Intermediate data print</td>
<td>ITM</td>
<td>ITM?</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>Measurement interval time</td>
<td>ITV</td>
<td>ITV?</td>
<td>A</td>
</tr>
<tr>
<td></td>
<td>%EFI measurement result</td>
<td>–</td>
<td>EFI?</td>
<td>A</td>
</tr>
<tr>
<td></td>
<td>Error performance print</td>
<td>EPF</td>
<td>EPF?</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>Data print format</td>
<td>FMT</td>
<td>FMT?</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>Paper saving function</td>
<td>PSV</td>
<td>PSV?</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>Threshold EI,EFI data print</td>
<td>THR</td>
<td>THR?</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>Termination code selection</td>
<td>TRM</td>
<td>TRM?</td>
<td>A</td>
</tr>
</tbody>
</table>
A.1.4 Compatibility with MP1775A

This section describes the compatibility between the MP1800A or the MT1810A commands and the MP1775A Pulse Pattern Generator.

<table>
<thead>
<tr>
<th>Section</th>
<th>Function</th>
<th>Control Message</th>
<th>Data Request Message</th>
<th>Compatibility</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clock</td>
<td>Operating clock</td>
<td>CLK</td>
<td>CLK?</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>Internal clock frequency resolution</td>
<td>RES</td>
<td>RES?</td>
<td>A</td>
</tr>
<tr>
<td></td>
<td>Internal clock frequency</td>
<td>FRQ</td>
<td>FRQ?</td>
<td>B</td>
</tr>
<tr>
<td>Pattern</td>
<td>Output pattern logic</td>
<td>LGC</td>
<td>LGC?</td>
<td>A</td>
</tr>
<tr>
<td></td>
<td>Output pattern</td>
<td>PTS</td>
<td>PTS?</td>
<td>C</td>
</tr>
<tr>
<td></td>
<td>Zero Subst, PRBS pattern stages</td>
<td>PTN</td>
<td>PTN?</td>
<td>A</td>
</tr>
<tr>
<td></td>
<td>PRGM data channel setting</td>
<td>PCH</td>
<td>PCH?</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>Data length</td>
<td>DLN</td>
<td>DLN?</td>
<td>A</td>
</tr>
<tr>
<td></td>
<td>Number of page</td>
<td>PAG</td>
<td>PAG?</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>Number of page</td>
<td>ADR</td>
<td>ADR?</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>Pattern bits</td>
<td>BIT</td>
<td>BIT?</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>Pattern data input bytes</td>
<td>WRT</td>
<td>–</td>
<td>A</td>
</tr>
<tr>
<td></td>
<td>Pattern data output bytes</td>
<td>–</td>
<td>RED?</td>
<td>A</td>
</tr>
<tr>
<td></td>
<td>Preset (all bits/channel)</td>
<td>ALL</td>
<td>–</td>
<td>A</td>
</tr>
<tr>
<td></td>
<td>Preset (all bits/page)</td>
<td>PST</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>0 continuous bit length</td>
<td>ZLN</td>
<td>ZLN?</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>Error addition</td>
<td>EAD</td>
<td>EAD?</td>
<td>C</td>
</tr>
<tr>
<td></td>
<td>Error addition channel</td>
<td>ECH</td>
<td>ECH?</td>
<td>A</td>
</tr>
<tr>
<td>Output</td>
<td>Output offset reference value</td>
<td>OFS</td>
<td>OFS?</td>
<td>A</td>
</tr>
<tr>
<td></td>
<td>Data output termination selection</td>
<td>DTM</td>
<td>DTM?</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>Data output display setting</td>
<td>SCD</td>
<td>SCD?</td>
<td>C</td>
</tr>
<tr>
<td></td>
<td>Data output common setting</td>
<td>DCM</td>
<td>DCM?</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>Data output amplitude</td>
<td>DAP</td>
<td>DAP?</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>Data output offset</td>
<td>DOS</td>
<td>DOS?</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>Clock output termination selection</td>
<td>CTM</td>
<td>CTM?</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>Clock output display selection</td>
<td>SCC</td>
<td>SCC?</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>Clock output common setting</td>
<td>CCM</td>
<td>CCM?</td>
<td>C</td>
</tr>
<tr>
<td></td>
<td>Clock output amplitude</td>
<td>CAP</td>
<td>CAP?</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>Clock output offset</td>
<td>COS</td>
<td>COS?</td>
<td>B</td>
</tr>
<tr>
<td></td>
<td>Clock output delay</td>
<td>CDL</td>
<td>CDL?</td>
<td>A</td>
</tr>
<tr>
<td></td>
<td>Data/Clock output ON/OFF</td>
<td>OON</td>
<td>OON?</td>
<td>C</td>
</tr>
<tr>
<td></td>
<td>Data Cross point</td>
<td>DCP</td>
<td>DCP?</td>
<td>C</td>
</tr>
<tr>
<td></td>
<td>Clock Duty</td>
<td>CDT</td>
<td>CDT?</td>
<td>A</td>
</tr>
<tr>
<td></td>
<td>Synchronous output selection</td>
<td>SOP</td>
<td>SOP?</td>
<td>C</td>
</tr>
</tbody>
</table>
### Table A.1.4-1  MP1775A device message (Cont’d)

<table>
<thead>
<tr>
<th>Section</th>
<th>Function</th>
<th>Control Message</th>
<th>Data Request Message</th>
<th>Compatibility</th>
</tr>
</thead>
<tbody>
<tr>
<td>Memory</td>
<td>Memory function selection</td>
<td>MEM</td>
<td>MEM?</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>File No./ Directory mode selection</td>
<td>FIL</td>
<td>FIL?</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>Data recall</td>
<td>RCL</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>Data save</td>
<td>SAV</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>Data resave</td>
<td>RSV</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>Data delete</td>
<td>DEL</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>FD format</td>
<td>FDF</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>File content search</td>
<td>–</td>
<td>FSH?</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>FD access status</td>
<td>–</td>
<td>MAC?</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>FD error message</td>
<td>–</td>
<td>FDE?</td>
<td>–</td>
</tr>
<tr>
<td>Other</td>
<td>Initialize</td>
<td>INI</td>
<td>–</td>
<td>A</td>
</tr>
<tr>
<td></td>
<td>Internal timer setting</td>
<td>RTM</td>
<td>RTM?</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>PLL lock state</td>
<td>–</td>
<td>PLL?</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>Delay state</td>
<td>–</td>
<td>DLY?</td>
<td>A</td>
</tr>
<tr>
<td></td>
<td>Response data termination code</td>
<td>TRM</td>
<td>TRM?</td>
<td>A</td>
</tr>
<tr>
<td></td>
<td>END event status enable register</td>
<td>ESE1</td>
<td>ESE1?</td>
<td>C</td>
</tr>
<tr>
<td></td>
<td>END event status register</td>
<td>–</td>
<td>ESR1?</td>
<td>C</td>
</tr>
<tr>
<td></td>
<td>ERROR event status enable register</td>
<td>ESE2</td>
<td>ESE2?</td>
<td>C</td>
</tr>
<tr>
<td></td>
<td>ERROR event status register</td>
<td>–</td>
<td>ESR2?</td>
<td>C</td>
</tr>
</tbody>
</table>
### A.1.5 Compatibility with MP1776A

This section describes the compatibility between the MP1800A or the MT1810A commands and the MP1776A Error Detector.

<table>
<thead>
<tr>
<th>Commands</th>
<th>Compatibility</th>
</tr>
</thead>
<tbody>
<tr>
<td>:SENSe1</td>
<td>2</td>
</tr>
<tr>
<td>:SENSe1</td>
<td>2</td>
</tr>
<tr>
<td>:SENSe1</td>
<td>2</td>
</tr>
<tr>
<td>:SENSe1</td>
<td>2</td>
</tr>
<tr>
<td>:SENSe1</td>
<td>2</td>
</tr>
<tr>
<td>:SENSe1</td>
<td>2</td>
</tr>
<tr>
<td>:SENSe1</td>
<td>2</td>
</tr>
<tr>
<td>:SENSe1</td>
<td>2</td>
</tr>
<tr>
<td>:SENSe1</td>
<td>2</td>
</tr>
<tr>
<td>:SENSe1</td>
<td>2</td>
</tr>
<tr>
<td>:SENSe1</td>
<td>2</td>
</tr>
<tr>
<td>:SENSe1</td>
<td>2</td>
</tr>
<tr>
<td>:SENSe1</td>
<td>2</td>
</tr>
<tr>
<td>:SENSe1</td>
<td>2</td>
</tr>
<tr>
<td>:SENSe1</td>
<td>2</td>
</tr>
<tr>
<td>:SENSe1</td>
<td>2</td>
</tr>
<tr>
<td>:SENSe1</td>
<td>2</td>
</tr>
<tr>
<td>:SENSe1</td>
<td>2</td>
</tr>
<tr>
<td>:SENSe1</td>
<td>2</td>
</tr>
<tr>
<td>:SENSe1</td>
<td>2</td>
</tr>
<tr>
<td>:SENSe1</td>
<td>2</td>
</tr>
<tr>
<td>:SENSe1</td>
<td>2</td>
</tr>
<tr>
<td>:SENSe1</td>
<td>2</td>
</tr>
<tr>
<td>:SENSe1</td>
<td>2</td>
</tr>
<tr>
<td>:SENSe1</td>
<td>2</td>
</tr>
<tr>
<td>:SENSe1</td>
<td>2</td>
</tr>
<tr>
<td>:SENSe1</td>
<td>2</td>
</tr>
<tr>
<td>:SENSe1</td>
<td>2</td>
</tr>
<tr>
<td>:SENSe1</td>
<td>2</td>
</tr>
<tr>
<td>:SENSe1</td>
<td>2</td>
</tr>
<tr>
<td>:SENSe1</td>
<td>2</td>
</tr>
<tr>
<td>:SENSe1</td>
<td>2</td>
</tr>
<tr>
<td>:SENSe1</td>
<td>2</td>
</tr>
<tr>
<td>:SENSe1</td>
<td>2</td>
</tr>
<tr>
<td>:SENSe1</td>
<td>2</td>
</tr>
<tr>
<td>:SENSe1</td>
<td>2</td>
</tr>
<tr>
<td>:SENSe1</td>
<td>2</td>
</tr>
<tr>
<td>:SENSe1</td>
<td>2</td>
</tr>
<tr>
<td>:SENSe1</td>
<td>2</td>
</tr>
<tr>
<td>:SENSe1</td>
<td>2</td>
</tr>
</tbody>
</table>
### Table A.1.5-1 MP1776A SCPI SENSE Subsystem (Cont’d)

<table>
<thead>
<tr>
<th>Commands</th>
<th>Compatibility</th>
</tr>
</thead>
<tbody>
<tr>
<td>:SENSe1</td>
<td>2</td>
</tr>
<tr>
<td>:SENSe1</td>
<td>2</td>
</tr>
<tr>
<td>:SENSe1</td>
<td>2</td>
</tr>
<tr>
<td>:SENSe1</td>
<td>2</td>
</tr>
<tr>
<td>:SENSe1</td>
<td>2</td>
</tr>
<tr>
<td>:SENSe1</td>
<td>2</td>
</tr>
<tr>
<td>:SENSe1</td>
<td>2</td>
</tr>
<tr>
<td>:SENSe1</td>
<td>2</td>
</tr>
<tr>
<td>:SENSe1</td>
<td>2</td>
</tr>
<tr>
<td>:SENSe1</td>
<td>2</td>
</tr>
<tr>
<td>:SENSe1</td>
<td>2</td>
</tr>
<tr>
<td>:SENSe1</td>
<td>2</td>
</tr>
<tr>
<td>:SENSe1</td>
<td>2</td>
</tr>
</tbody>
</table>

### Table A.1.5-2 MP1776A INPut Subsystem

<table>
<thead>
<tr>
<th>Commands</th>
<th>Compatibility</th>
</tr>
</thead>
<tbody>
<tr>
<td>:INPut1</td>
<td>2</td>
</tr>
<tr>
<td>:INPut1</td>
<td>2</td>
</tr>
<tr>
<td>:INPut1</td>
<td>2</td>
</tr>
<tr>
<td>:INPut1</td>
<td>2</td>
</tr>
<tr>
<td>:INPut1</td>
<td>2</td>
</tr>
<tr>
<td>:INPut1</td>
<td>2</td>
</tr>
<tr>
<td>:INPut1</td>
<td>2</td>
</tr>
<tr>
<td>:INPut1</td>
<td>2</td>
</tr>
<tr>
<td>:INPut1</td>
<td>2</td>
</tr>
<tr>
<td>:INPut1</td>
<td>2</td>
</tr>
<tr>
<td>:INPut1</td>
<td>2</td>
</tr>
<tr>
<td>:INPut1</td>
<td>2</td>
</tr>
</tbody>
</table>
### Table A.1.5-3 MP1776A CALCulate Subsystem

<table>
<thead>
<tr>
<th>Commands</th>
<th>Compatibility</th>
</tr>
</thead>
<tbody>
<tr>
<td>:CALCulate1</td>
<td>2</td>
</tr>
<tr>
<td>:CALCulate1</td>
<td>2</td>
</tr>
<tr>
<td>:CALCulate1</td>
<td>2</td>
</tr>
<tr>
<td>:CALCulate1</td>
<td>2</td>
</tr>
<tr>
<td>:CALCulate1</td>
<td>2</td>
</tr>
<tr>
<td>:CALCulate1</td>
<td>2</td>
</tr>
<tr>
<td>:CALCulate1</td>
<td>2</td>
</tr>
<tr>
<td>:CALCulate1</td>
<td>2</td>
</tr>
<tr>
<td>:CALCulate1</td>
<td>2</td>
</tr>
<tr>
<td>:CALCulate1</td>
<td>2</td>
</tr>
<tr>
<td>:CALCulate1</td>
<td>2</td>
</tr>
<tr>
<td>:CALCulate1</td>
<td>2</td>
</tr>
<tr>
<td>:CALCulate1</td>
<td>2</td>
</tr>
<tr>
<td>:CALCulate1</td>
<td>2</td>
</tr>
</tbody>
</table>

### Table A.1.5-4 MP1776A OUTPut Subsystem

<table>
<thead>
<tr>
<th>Commands</th>
<th>Compatibility</th>
</tr>
</thead>
<tbody>
<tr>
<td>:OUTPut1</td>
<td>2</td>
</tr>
<tr>
<td>:OUTPut1</td>
<td>2</td>
</tr>
</tbody>
</table>

### Table A.1.5-5 MP1776A INSTrument Subsystem

<table>
<thead>
<tr>
<th>Commands</th>
<th>Compatibility</th>
</tr>
</thead>
<tbody>
<tr>
<td>:INSTrument1</td>
<td>2</td>
</tr>
<tr>
<td>:INSTrument1</td>
<td>2</td>
</tr>
</tbody>
</table>
### Table A.1.5-6  MP1776A DISPlay Subsystem

<table>
<thead>
<tr>
<th>Commands</th>
<th>Compatibility</th>
</tr>
</thead>
<tbody>
<tr>
<td>:DISPlay:WINDow:TILE</td>
<td>–</td>
</tr>
<tr>
<td>:DISPlay:WINDow:CASCade</td>
<td>–</td>
</tr>
<tr>
<td>:DISPlay:WINDow:OPEN &lt;disp&gt;</td>
<td>–</td>
</tr>
<tr>
<td>:DISPlay:WINDow:CLOSE &lt;disp&gt;</td>
<td>–</td>
</tr>
<tr>
<td>:DISPlay:MFRame[:NAME] &lt;name&gt;</td>
<td>–</td>
</tr>
<tr>
<td>:DISPlay:MFRame[:NAME]?</td>
<td>–</td>
</tr>
<tr>
<td>:DISPlay:RESult:TIME &lt;type&gt;</td>
<td>A</td>
</tr>
<tr>
<td>:DISPlay:RESult:TIME?</td>
<td>A</td>
</tr>
<tr>
<td>:DISPlay:RESult:EALarm:MODE &lt;mode&gt;</td>
<td>B</td>
</tr>
<tr>
<td>:DISPlay:RESult:EALarm:HRESet</td>
<td>A</td>
</tr>
</tbody>
</table>
### Table A.1.5-7  MP1776A SYSTem Subsystem

<table>
<thead>
<tr>
<th>Commands</th>
<th>Compatibility</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYSTem:BEEPer:ERRor:SET &lt;boolean&gt;</td>
<td>A</td>
</tr>
<tr>
<td>SYSTem:BEEPer:ERRor:SET?</td>
<td>A</td>
</tr>
<tr>
<td>SYSTem:BEEPer:ERRor:CHANnel &lt;ch&gt;,&lt;boolean&gt;</td>
<td>–</td>
</tr>
<tr>
<td>SYSTem:BEEPer:ERRor:CHANnel? &lt;ch&gt;</td>
<td>–</td>
</tr>
<tr>
<td>SYSTem:BEEPer:ALARm:SET &lt;boolean&gt;</td>
<td>A</td>
</tr>
<tr>
<td>SYSTem:BEEPer:ALARm:SET?</td>
<td>A</td>
</tr>
<tr>
<td>SYSTem:BEEPer:ALARm:CHANnel &lt;ch&gt;,&lt;boolean&gt;</td>
<td>–</td>
</tr>
<tr>
<td>SYSTem:BEEPer:ALARm:CHANnel? &lt;ch&gt;</td>
<td>–</td>
</tr>
<tr>
<td>SYSTem:BEEPer:SYSTem:SET &lt;boolean&gt;</td>
<td>A</td>
</tr>
<tr>
<td>SYSTem:BEEPer:SYSTem:SET?</td>
<td>A</td>
</tr>
<tr>
<td>SYSTem:BEEPer:SYSTem:TYPE &lt;type&gt;,&lt;boolean&gt;</td>
<td>–</td>
</tr>
<tr>
<td>SYSTem:BEEPer:SYSTem:TYPE?</td>
<td>–</td>
</tr>
<tr>
<td>SYSTem:DATE &lt;year&gt;,&lt;month&gt;,&lt;day&gt;</td>
<td>–</td>
</tr>
<tr>
<td>SYSTem:DATE?</td>
<td>–</td>
</tr>
<tr>
<td>SYSTem:TIME &lt;hour&gt;,&lt;min&gt;,&lt;sec&gt;</td>
<td>–</td>
</tr>
<tr>
<td>SYSTem:TIME?</td>
<td>–</td>
</tr>
<tr>
<td>SYSTem:ERRor?</td>
<td>A</td>
</tr>
<tr>
<td>SYSTem:VERSion?</td>
<td>A</td>
</tr>
<tr>
<td>SYSTem:ORGanization:HARDware?</td>
<td>C</td>
</tr>
<tr>
<td>SYSTem:ORGanization:SOFTware?</td>
<td>–</td>
</tr>
<tr>
<td>SYSTem:MMEMory:RECall &lt;type&gt;,&lt;file_name&gt;,&lt;ch&gt;</td>
<td>C</td>
</tr>
<tr>
<td>SYSTem:MMEMory:STORe &lt;type&gt;,&lt;file_name&gt;,&lt;ch&gt;</td>
<td>C</td>
</tr>
<tr>
<td>SYSTem:MMEMory:STORe:CATalog? &lt;drv_dir&gt;,&lt;type&gt;</td>
<td>–</td>
</tr>
<tr>
<td>SYSTem:MEMory:INITialize</td>
<td>A</td>
</tr>
<tr>
<td>SYSTem:PRINt:COPY</td>
<td>A</td>
</tr>
<tr>
<td>SYSTem:PRINt:EALarm &lt;item&gt;</td>
<td>C</td>
</tr>
<tr>
<td>SYSTem:PRINt:PATTern &lt;ch&gt;,&lt;form&gt;,&lt;start&gt;,&lt;end&gt;</td>
<td>–</td>
</tr>
<tr>
<td>SYSTem:PRINt:CANCel</td>
<td>–</td>
</tr>
</tbody>
</table>
### Table A.1.5-8  MP1776A STATus Subsystem

<table>
<thead>
<tr>
<th>Commands</th>
<th>Compatibility</th>
</tr>
</thead>
<tbody>
<tr>
<td>STATus:PRESet</td>
<td>A</td>
</tr>
<tr>
<td>STATus:OPERation[:EVENt]?</td>
<td>C</td>
</tr>
<tr>
<td>STATus:OPERation:CONDtion?</td>
<td>C</td>
</tr>
<tr>
<td>STATus:OPERation:ENABle &lt;numeric&gt;</td>
<td>C</td>
</tr>
<tr>
<td>STATus:OPERation:ENABle?</td>
<td>C</td>
</tr>
<tr>
<td>STATus:OPERation:PTRansition &lt;numeric&gt;</td>
<td>C</td>
</tr>
<tr>
<td>STATus:OPERation:PTRansition?</td>
<td>C</td>
</tr>
<tr>
<td>STATus:OPERation:NTRansition &lt;numeric&gt;</td>
<td>C</td>
</tr>
<tr>
<td>STATus:OPERation:NTRansition?</td>
<td>C</td>
</tr>
<tr>
<td>STATus:OPERation:INStrument[:EVENt]?</td>
<td>–</td>
</tr>
<tr>
<td>STATus:OPERation:INStrument:CONDtion?</td>
<td>–</td>
</tr>
<tr>
<td>STATus:OPERation:INStrument:ENABle &lt;numeric&gt;</td>
<td>–</td>
</tr>
<tr>
<td>STATus:OPERation:INStrument:ENABle?</td>
<td>–</td>
</tr>
<tr>
<td>STATus:OPERation:INStrument:PTRansition &lt;numeric&gt;</td>
<td>–</td>
</tr>
<tr>
<td>STATus:OPERation:INStrument:PTRansition?</td>
<td>–</td>
</tr>
<tr>
<td>STATus:OPERation:INStrument:NTRansition &lt;numeric&gt;</td>
<td>–</td>
</tr>
<tr>
<td>STATus:OPERation:INStrument:NTRansition?</td>
<td>–</td>
</tr>
<tr>
<td>STATus:QUEStionable[:EVENt]?</td>
<td>–</td>
</tr>
<tr>
<td>STATus:QUEStionable:ENABle &lt;numeric&gt;</td>
<td>–</td>
</tr>
<tr>
<td>STATus:QUEStionable:ENABle?</td>
<td>–</td>
</tr>
<tr>
<td>STATus:QUEStionable:PTRansition &lt;numeric&gt;</td>
<td>–</td>
</tr>
<tr>
<td>STATus:QUEStionable:PTRansition?</td>
<td>–</td>
</tr>
<tr>
<td>STATus:QUEStionable:NTRansition &lt;numeric&gt;</td>
<td>–</td>
</tr>
<tr>
<td>STATus:QUEStionable:NTRansition?</td>
<td>–</td>
</tr>
<tr>
<td>STATus:QUEStionable:MONitor[:EVENt]?</td>
<td>–</td>
</tr>
<tr>
<td>STATus:QUEStionable:MONitor:CONDtion?</td>
<td>–</td>
</tr>
</tbody>
</table>
### Table A.1.5-8 MP1776A STATus Subsystem (Cont’d)

<table>
<thead>
<tr>
<th>Commands</th>
<th>Compatibility</th>
</tr>
</thead>
<tbody>
<tr>
<td>STATus:QUESTIONable:MONitor:ENABle &lt;numeric&gt;</td>
<td>–</td>
</tr>
<tr>
<td>STATus:QUESTIONable:MONitor:ENABle?</td>
<td>–</td>
</tr>
<tr>
<td>STATus:QUESTIONable:MONitor:PTRansition &lt;numeric&gt;</td>
<td>–</td>
</tr>
<tr>
<td>STATus:QUESTIONable:MONitor:PTRansition?</td>
<td>–</td>
</tr>
<tr>
<td>STATus:QUESTIONable:MONitor:NTRansition &lt;numeric&gt;</td>
<td>–</td>
</tr>
<tr>
<td>STATus:QUESTIONable:MONitor:NTRansition?</td>
<td>–</td>
</tr>
<tr>
<td>STATus:QUESTIONable:MONitor:CH1</td>
<td>:CH2</td>
</tr>
</tbody>
</table>