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 MP1800A 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

Document No.: M-W2799AE-28.0

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)

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.

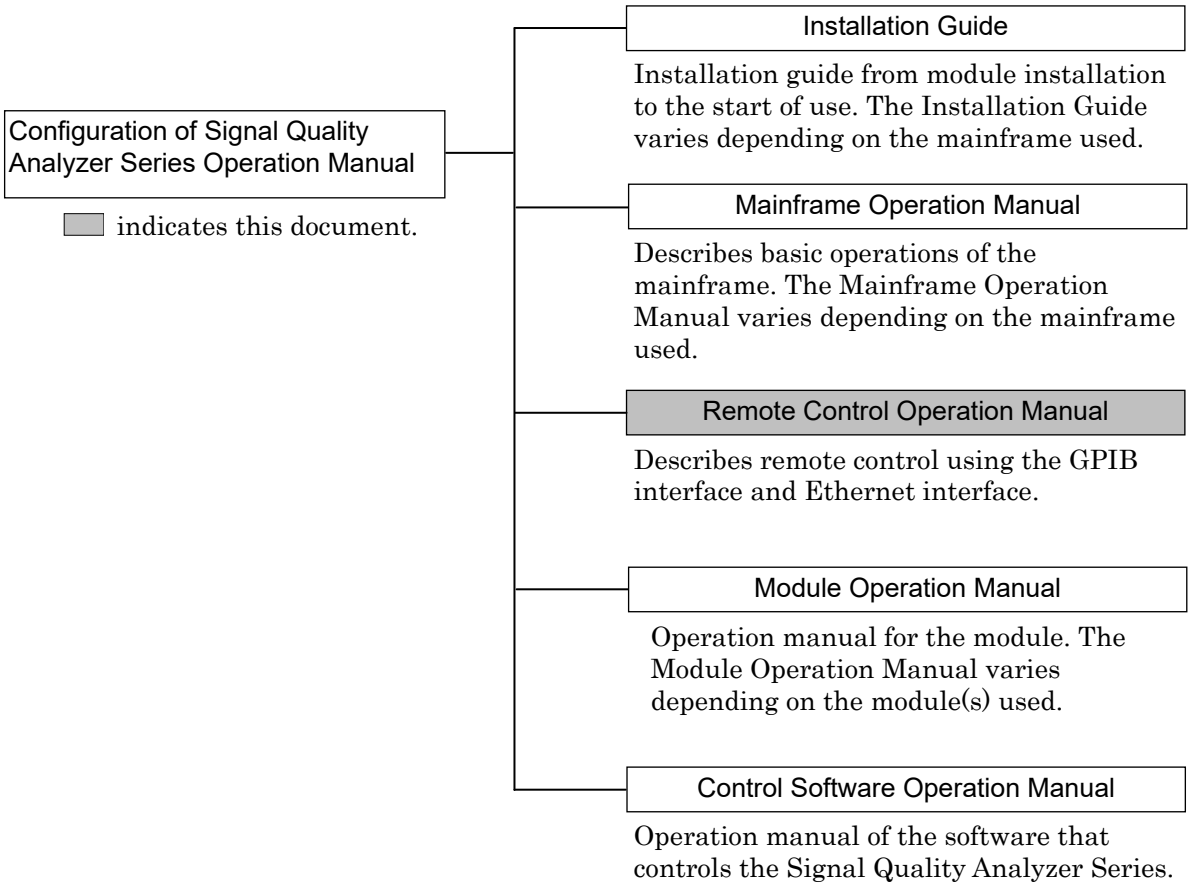


Table of Contents

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

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

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.

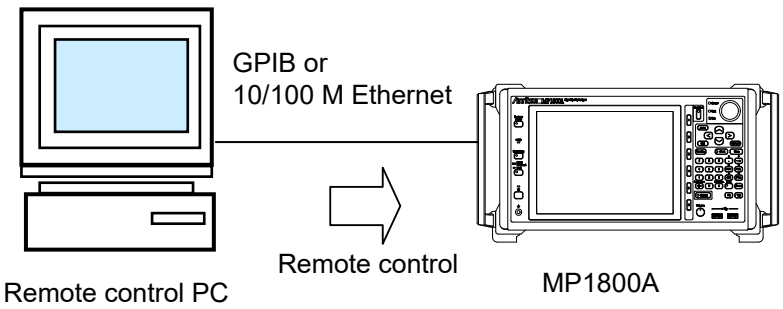


Figure 2.1.1-1 Configuration for remote control of MP1800A from the remote control PC

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.

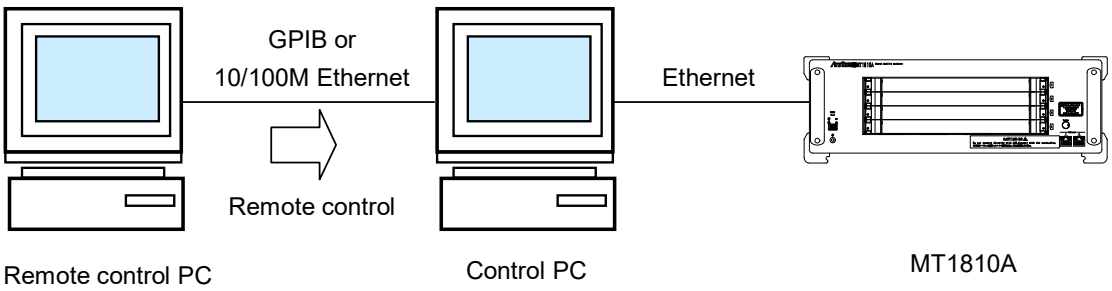


Figure 2.1.2-1 Configuration for remote control of the MX180000A from the remote control PC

[2] 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).

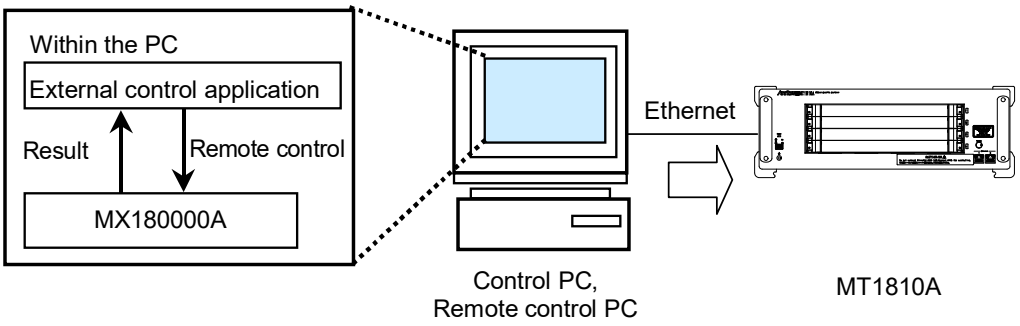


Figure 2.1.2-2 Configuration for remote control of MX180000A within the same PC

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 2.2.1-1 Interface functions

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

*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 2.2.2-1 Device message

Program Message (See Section 3.1.2)	Response Message (See Section 3.2.2)
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

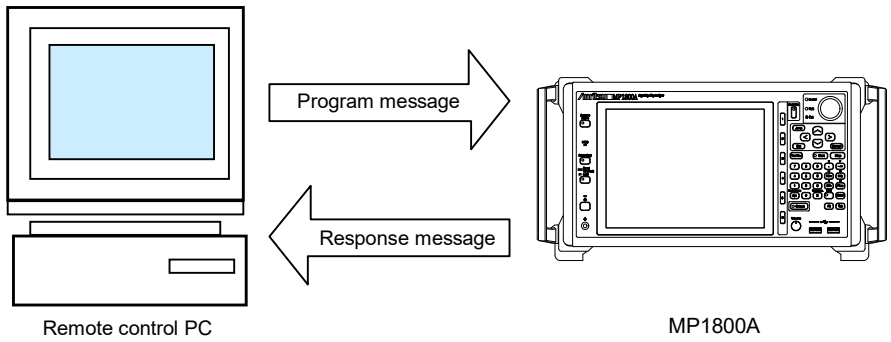


Figure 2.2.2-1 Device messages

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

Table 2.2.2-2 I/O buffer

Input Buffer	Output Queue
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.	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.

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 2.2.3-1 Bus commands

Bus Command	Operation
DCL (Device Clear)	Initializes message exchange of all devices connected to the GPIB bus.
SDC (Selected Device Clear)	Initializes message exchange of the addressed MP1800A. The operation is the same as the DCL.
IFC (Interface Clear)	Initializes the interface.

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	≤ 15 units
Total cable length	≤ 2 m × Number of devices (20 m, max.)

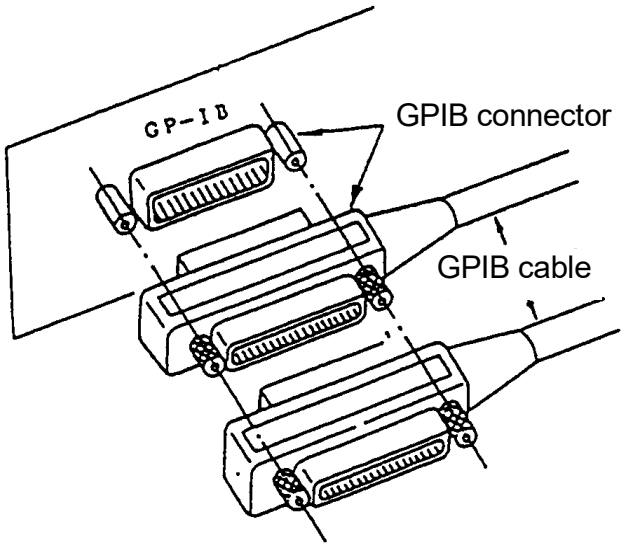


Figure 2.2.4-1 GPIB cable connection

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 2.2.5-1 Setting GPIB

Setting Detail	Setting Item	Setting Value
Selection of the interface used for remote control	Active Interface	GPIB: Uses the GPIB interface for remote control.
Address setting	Address	0 to 30
Command processing mode	Performance	Normal/Enhanced

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.

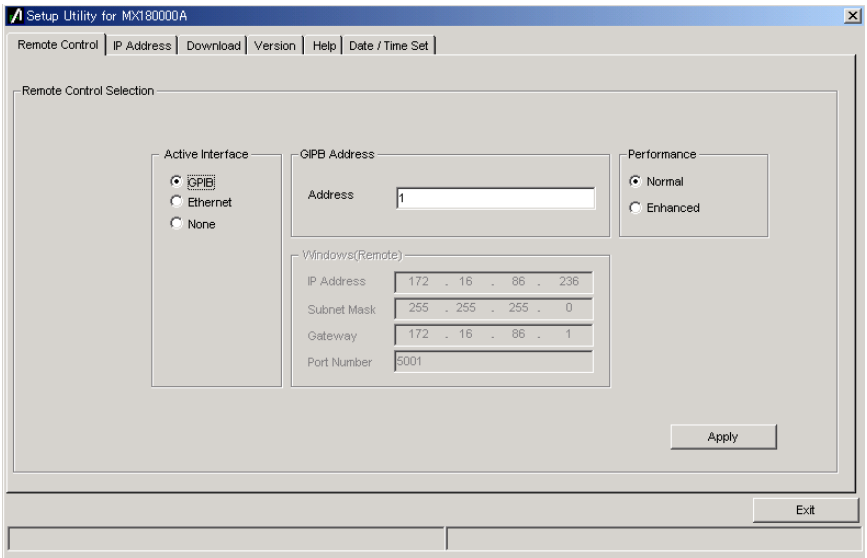


Figure 2.2.5-1 Setup Utility: Remote Control tab – GPIB setting

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

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

Level	Initialization Type	Description
1	Bus initialization	Initializes all interface functions connected to the bus with IFC message from the controller.
2	Message initialization	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.
3	Device initialization	Recovers the known state that is specific to the device with the *RST command regardless of the past use conditions.

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 2.3.1-1 Device message

Program message (See Section 3.1.2)	Response message (See Section 3.2.2)
Program instruction • Device-unique command (See Chapters 7 through 8.) • IEEE488.2 common command (See Chapter 5.)	Program query • Status message (See Chapter 6.) • Response message

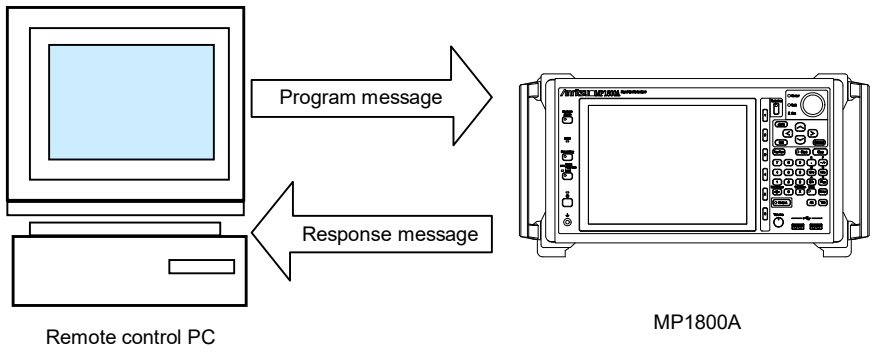


Figure 2.3.1-1 Device messages

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

Table 2.3.1-2 I/O buffer

Input Buffer	Output Queue
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.	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.

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.

2.3.2 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 2.3.2-1 Ethernet board setting

Window	Setting Detail	Setting Item	Setting Value
Remote Control tab	Selection of the interface used for remote control	Active Interface	Ethernet: Uses the Ethernet interface for remote control.
	Port number setting	Port Number	1024 to 5001
Network setting, property setting	IP address	IP Address	0.0.0.0 to 255.255.255.255
	Subnet mask	Sub Net Mask	0.0.0.0 to 255.255.255.255
	Gateway	GateWay	0.0.0.0 to 255.255.255.255

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.

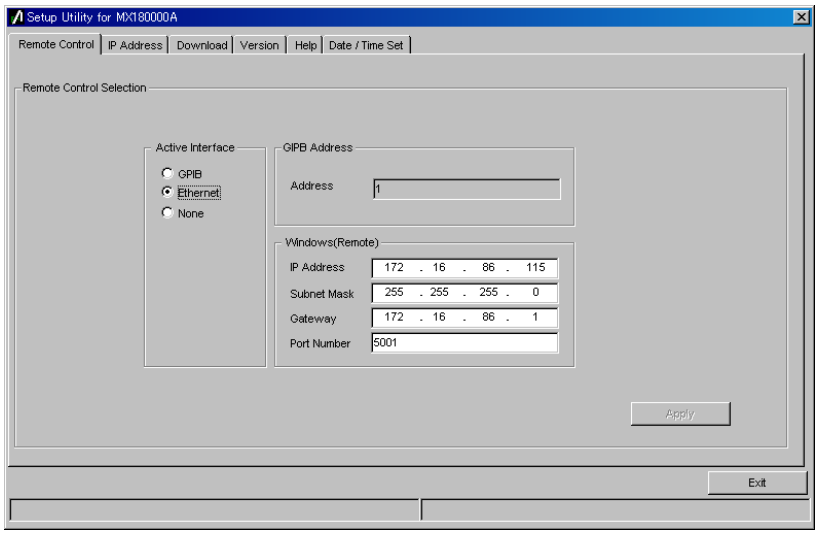


Figure 2.3.2-1 Setup Utility: Remote Control tab – Ethernet setting

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 2.3.2-2 IP address

Class	Network part length	Host part length	Number of hosts that can be assigned
A	8 bits	24 bits	16,777,214 units
B	16 bits	16 bits	65,534 units
C	24 bits	8 bits	254 units

- 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.

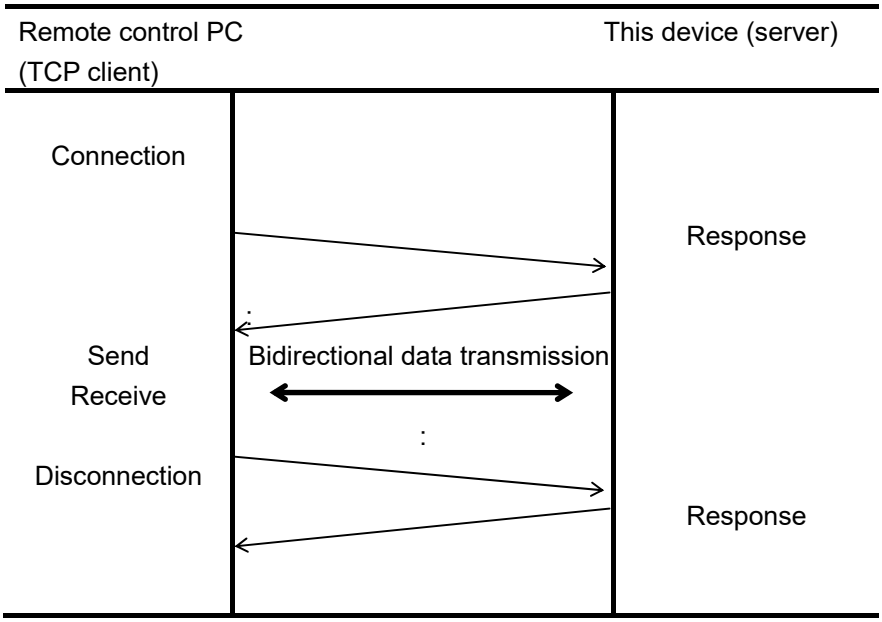


Figure 2.3.3-1 Data flow

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.

Chapter 3 SCPI Format

This chapter describes the SCPI command system.

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.

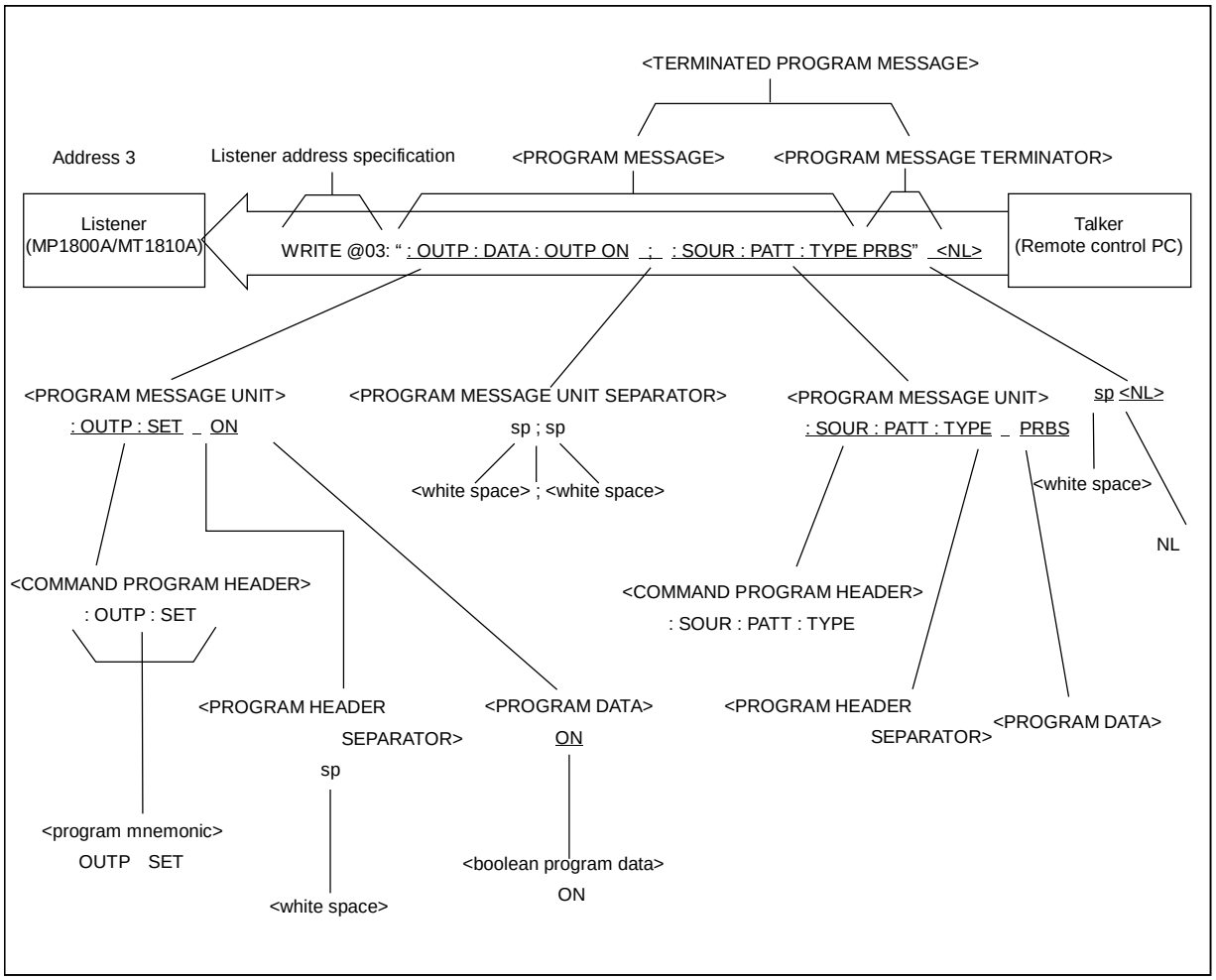


Figure 3.1.1-1 SCPI listener input program message

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

WRITE @Device number: Data

Data → | Mathematical expression
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.

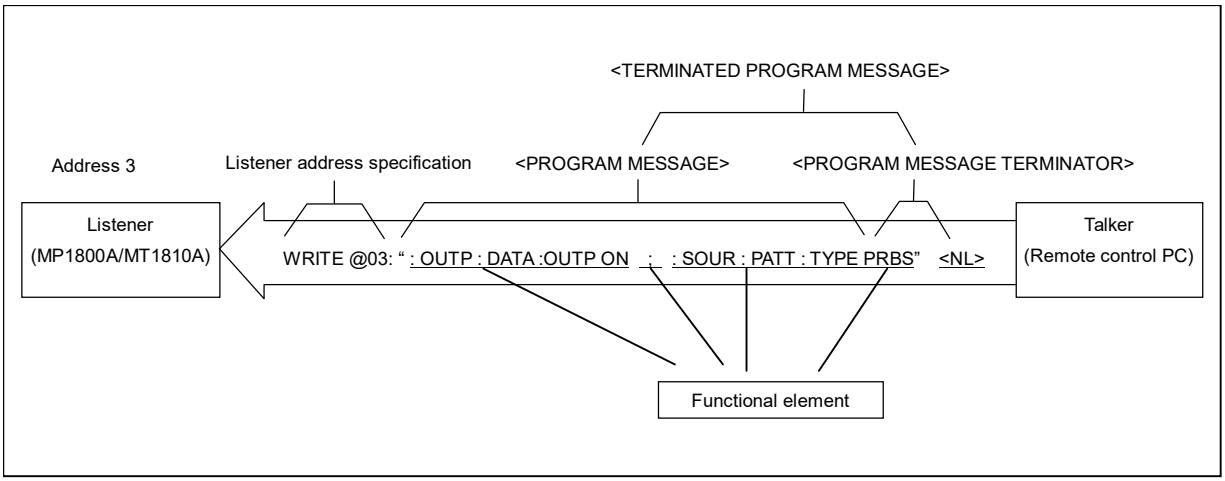
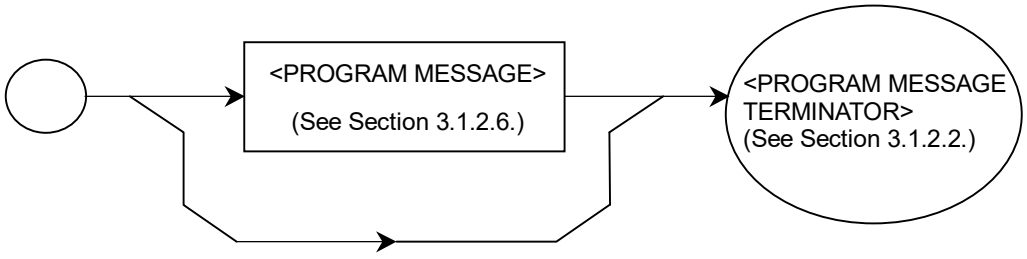


Figure 3.1.2.1-1 <TERMINATED PROGRAM MESSAGE>

Definition
<TERMINATED PROGRAM MESSAGE> is defined as follows:

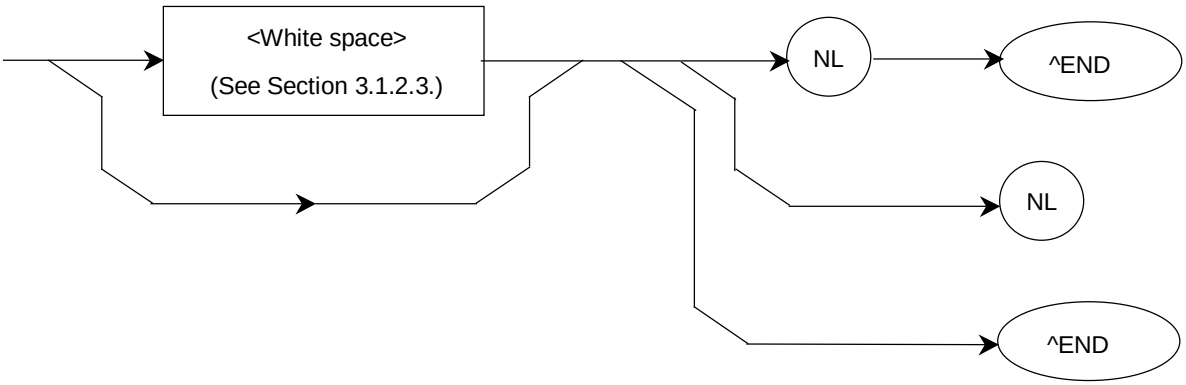


<TERMINATED PROGRAM MESSAGE> is a data message containing all the functional elements required for transmitting the message from the remote control PC to the MP1800A/MT1810A. A <PROGRAM MESSAGE TERMINATOR> is added to the end of a <PROGRAM MESSAGE> to complete transmission of the <PROGRAM MESSAGE>.

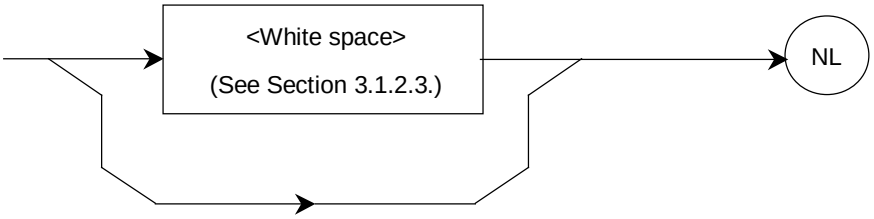
3.1.2.2 <PROGRAM MESSAGE TERMINATOR>

<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:

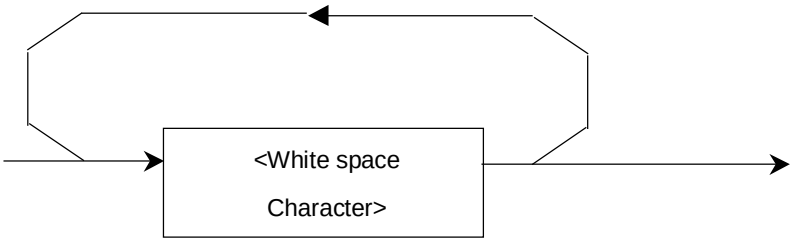


NL Defined as a single ASCII code byte 0A (decimal number 10), i.e. the ASCII control character LF (Line Feed) that performs the carriage return operation bringing the print position to the same character position in the next line. It is also called NL (New Line) because it starts from a new line.
For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

<white space> is defined as follows:



<white space Character> 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.

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.

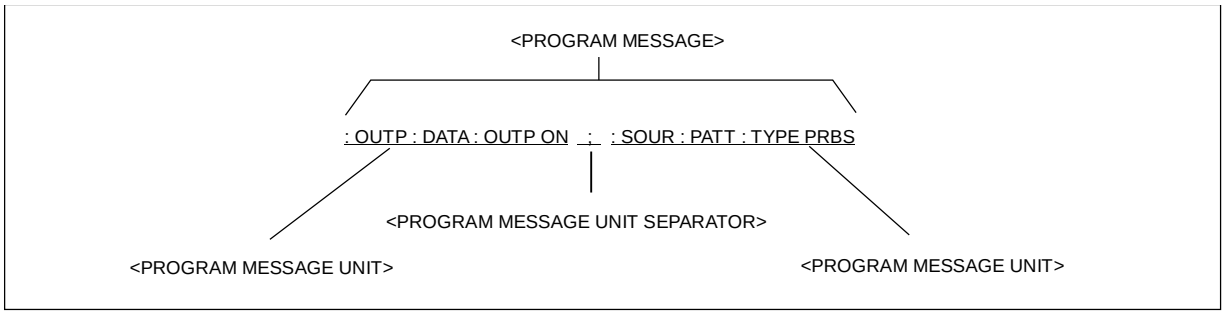
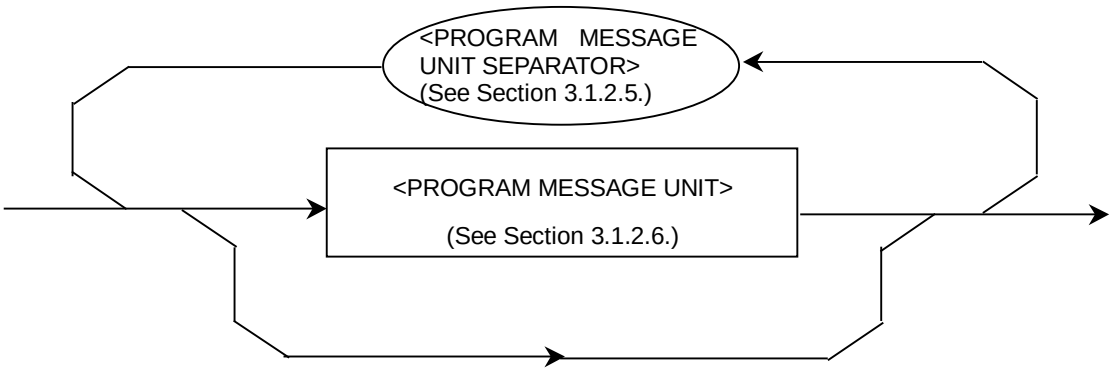


Figure 3.1.2.4-1 <PROGRAM MESSAGE>

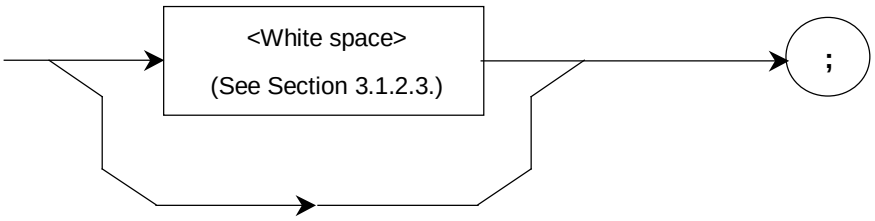
Definition
<PROGRAM MESSAGE> is defined as follows:



<PROGRAM MESSAGE> is a sequence of zero, one or more <PROGRAM MESSAGE UNIT> elements. The <PROGRAM MESSAGE UNIT> element indicates a programming instruction or data to be sent from the remote control PC to the MP1800A/MT1810A. <PROGRAM MESSAGE UNIT SEPARATOR> is used to separate two or more <PROGRAM MESSAGE UNIT> elements.

3.1.2.5 <PROGRAM MESSAGE UNIT SEPARATOR>

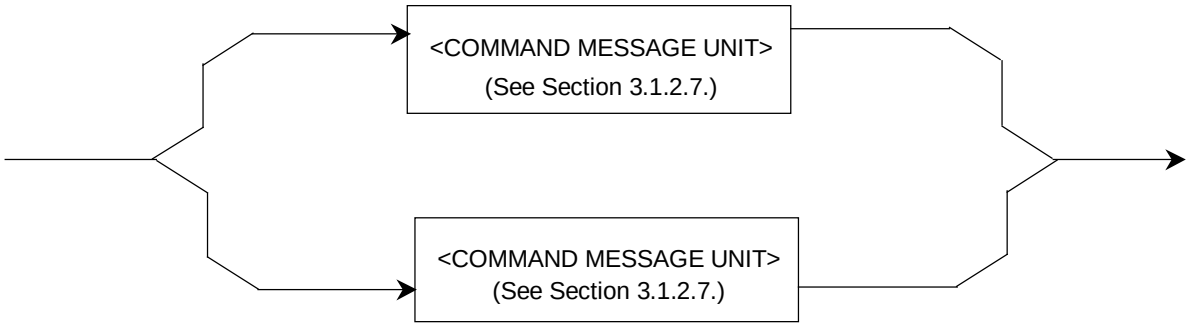
<PROGRAM MESSAGE UNIT SEPARATOR> is defined as follows:



<PROGRAM MESSAGE UNIT SEPARATOR> separates a sequence of two or more <PROGRAMMESSAGE 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:



<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.

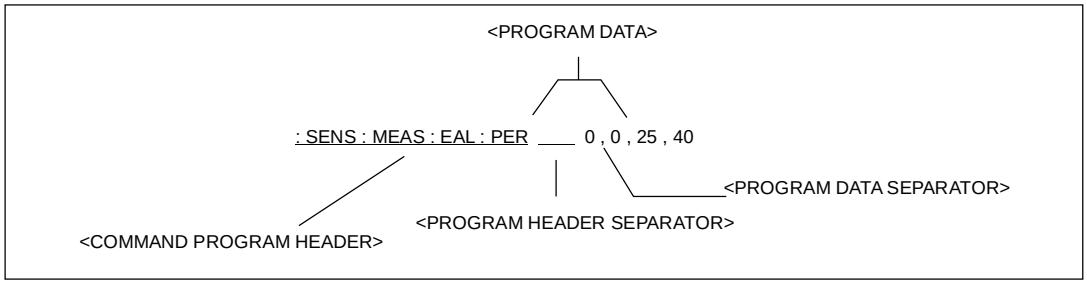
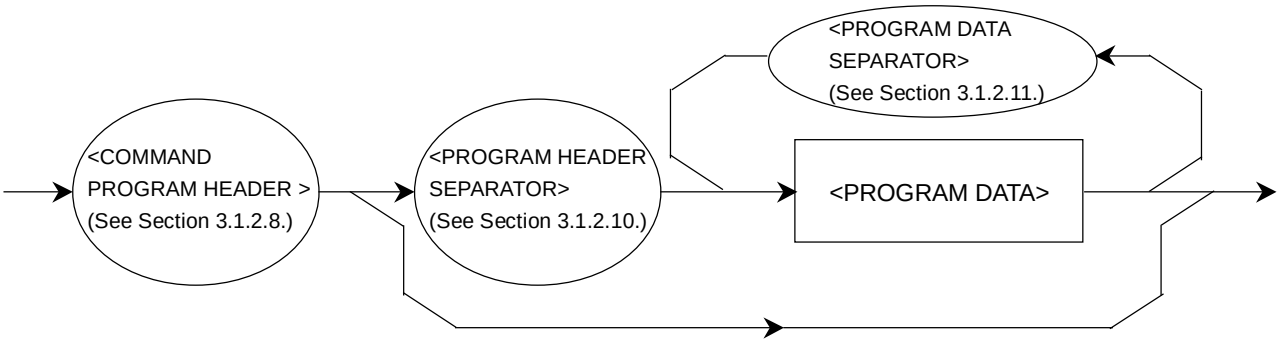


Figure 3.1.2.7-1 <COMMAND MESSAGE UNIT>

<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.

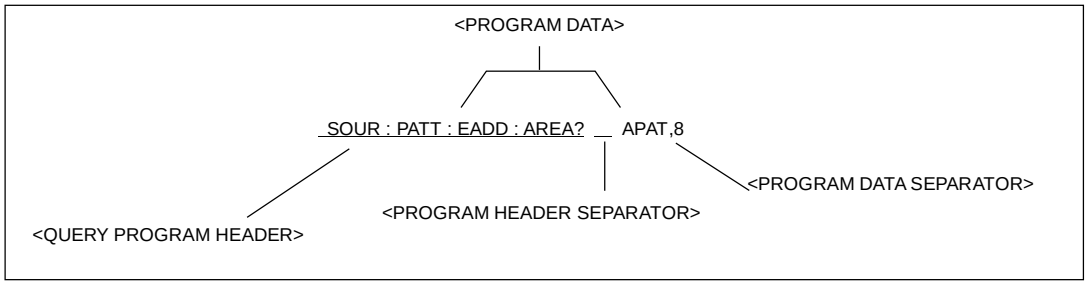
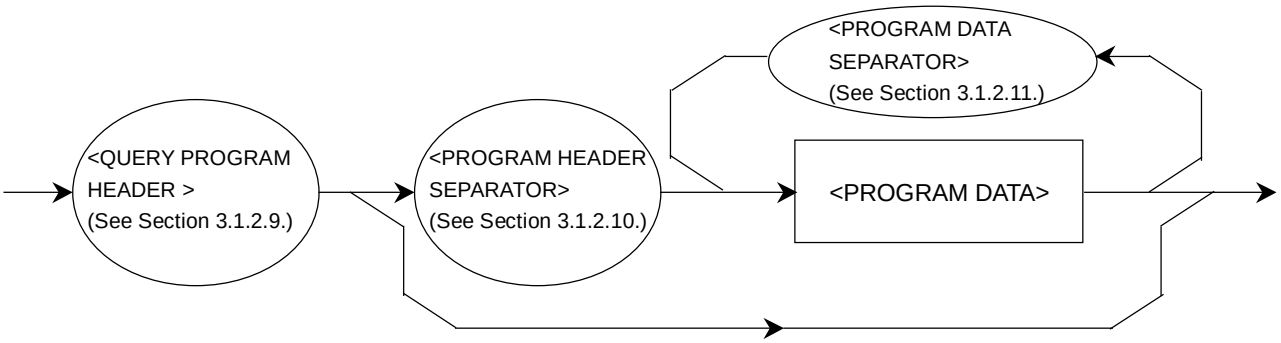


Figure 3.1.2.7-2 <QUERY MESSAGE UNIT>

<QUERY MESSAGE UNIT> is defined as follows:

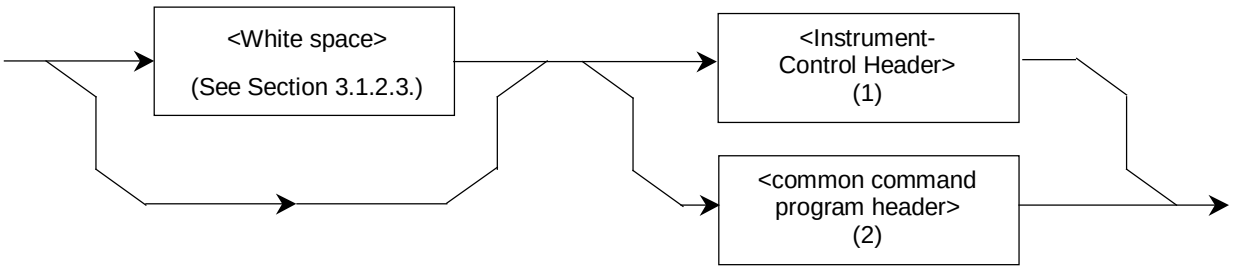


For both <COMMAND MESSAGE UNIT> and <QUERY MESSAGE UNIT>, when program data follows the program header, one space must be inserted as a separator between them. The program header identifies the application, function, and operation of the program. If no program data is added, the header alone indicates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

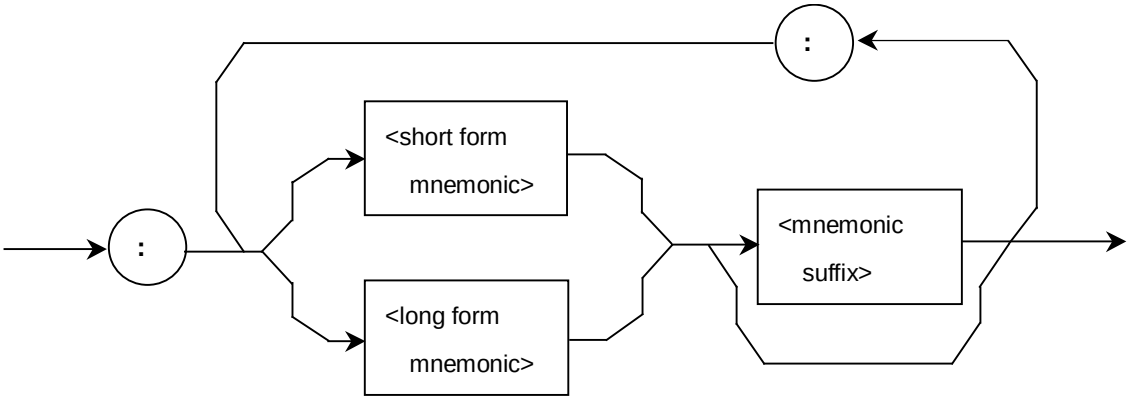
3.1.2.8 <COMMAND PROGRAM HEADER>

<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:

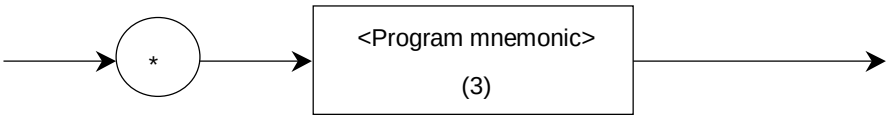


■ <Instrument-Control Headers>
<Instrument-Control Header> is defined in the SCPI. The device-unique commands of the MP1800A/MT1810A comply with the SCPI, so the command format conforms to the SCPI.

- <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.

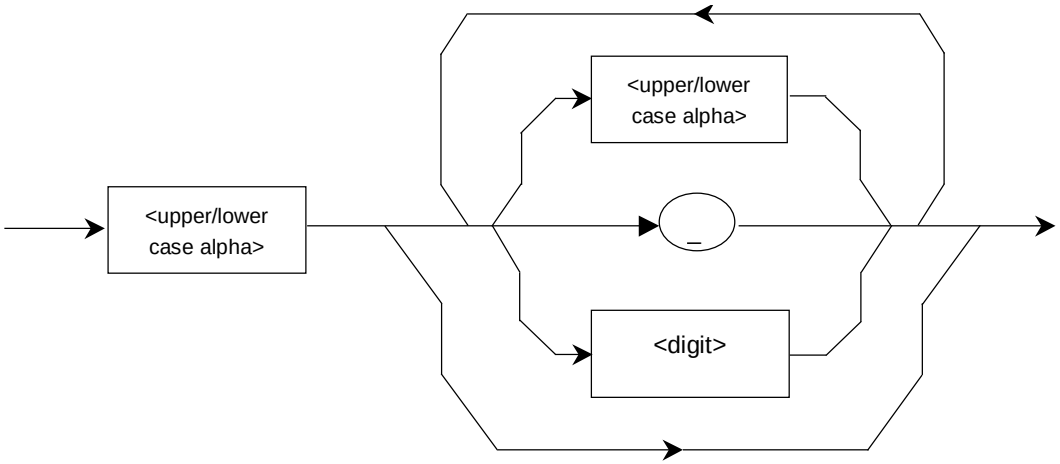
- <numeric suffix>
  Defined as a single ASCII code byte, within the range of the ASCII code bytes 30 to 39 (decimal numbers 48 to 57 = numerical values 0 to 9).

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



■ <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:



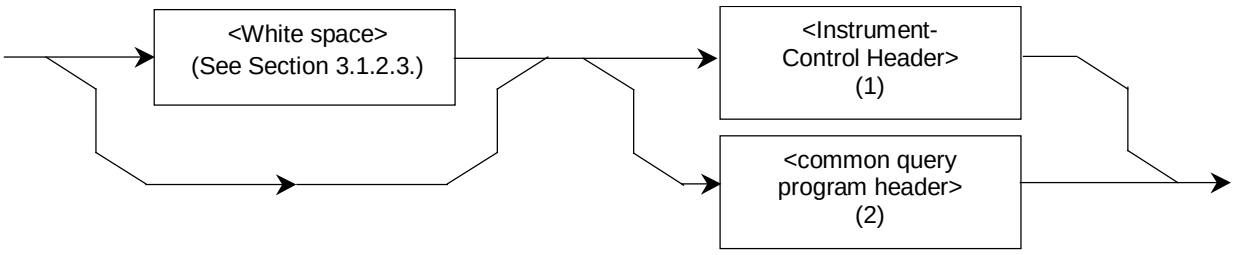
■ <program mnemonic>
A mnemonic must start with an uppercase or lowercase alphabetic character, followed by any combination of uppercase alphabetic characters ("A" to "Z"), lowercase alphabetic characters ("a" to "z"), underbar (_), and numbers ("0" to "9"). The maximum length of a mnemonic is twelve 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).

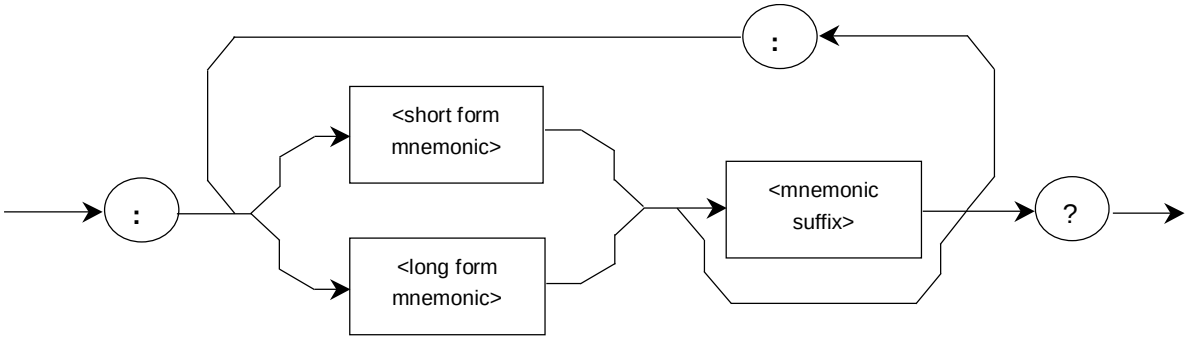
- <digit>
  Defined as a single ASCII code byte, within the range of the ASCII code bytes 30 to 39 (decimal numbers 48 to 57 = numerical values 0 to 9).
- (_)
  Indicates the ASCII code byte 5F (decimal number 95 = underbar). It is defined as a single ASCII code byte.
  Example: :SYSTem (The subsequent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

3.1.2.9 <QUERY PROGRAM HEADER>

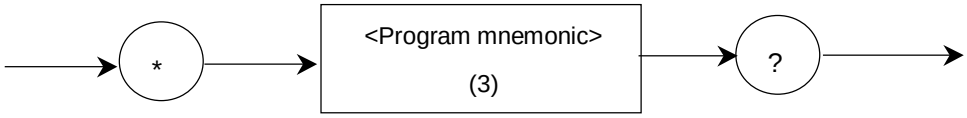
<QUERY PROGRAM HEADER> is defined as follows. A <white space> can be inserted in front of each header.



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



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



■ <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>

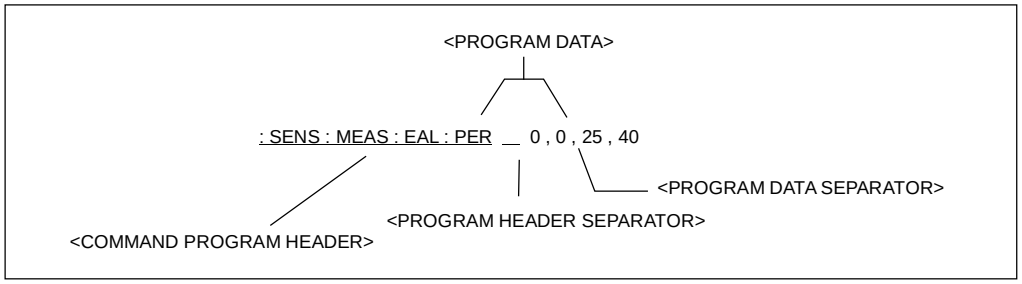
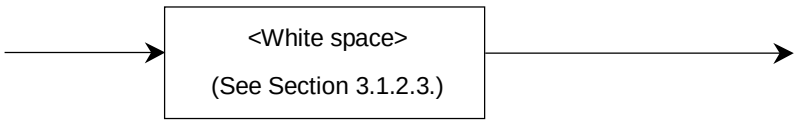


Figure 3.1.2.10-1 <PROGRAM HEADER SEPARATOR>

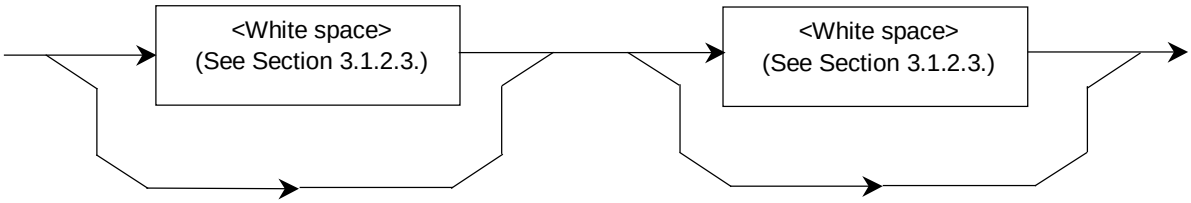
Definition <PROGRAM HEADER SEPARATOR> is defined as follows:



<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.

3.1.3 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.

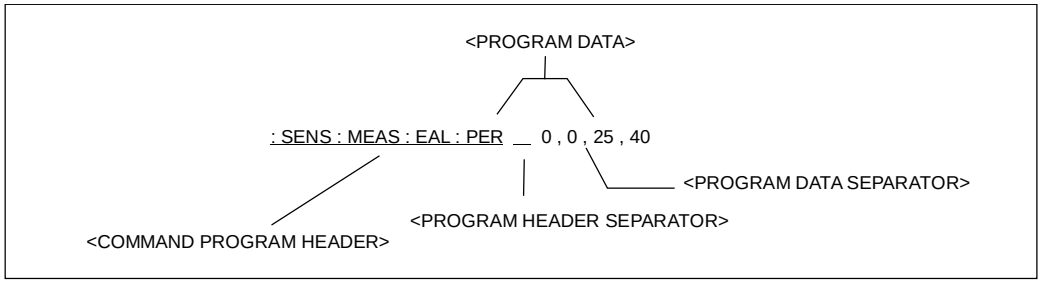


Figure 3.1.3-1 <PROGRAM DATA>

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 3.1.3-1 Program data format

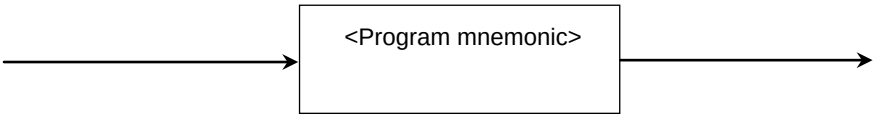
<PROGRAM DATA>	Type
<CHARACTER PROGRAM DATA>	Indicates short alphabetic or alphanumeric data.
<DECIMAL NUMERIC PROGRAM DATA>	Indicates a decimal numerical constant.
<NON-DECIMAL NUMERIC PROGRAM DATA>	Indicates an alphanumeric character other than a decimal.
<BOOLEAN PROGRAM DATA>	Indicates a theoretical value (defined in SCPI).
<STRING PROGRAM DATA>	Indicates a string enclosed within double quotation marks (" ") or single quotation marks (' ').

3.1.3.1 <CHARACTER PROGRAM DATA>

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

Example:	: SENSe : PATTern : TYPE PRBS (Mnemonic data indicating PRBS)
	: SENSe : MEASure : EALarm : UNIT CLOCk
	(Mnemonic data indicating the measurement cycle in Clock Count units.)

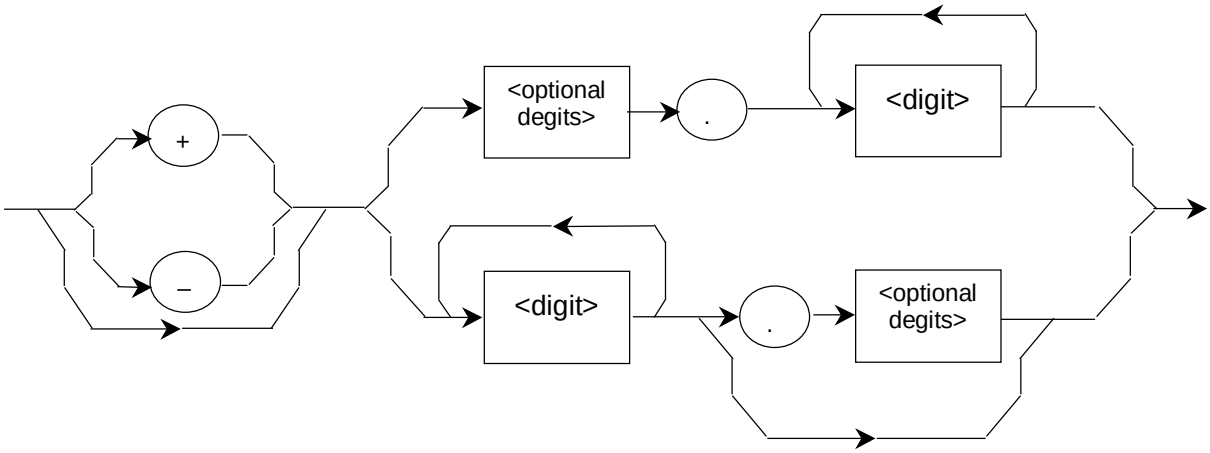
The character data is the same as those described "<program mnemonic>" in Section 3.1.2.8 (3).



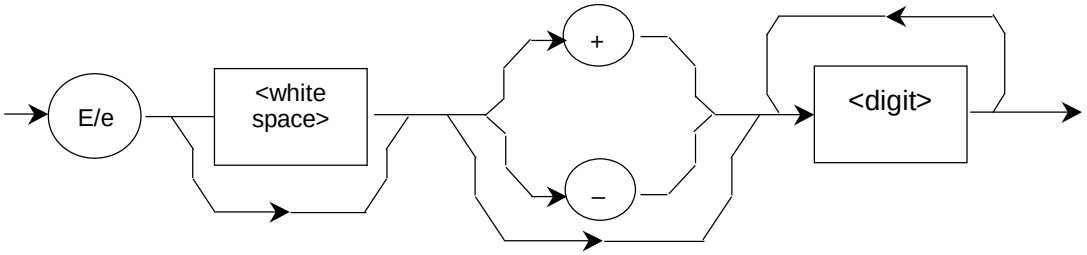
3.1.3.2 <DECIMAL NUMERIC PROGRAM DATA>

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

<mantissa> is defined as follows:



<exponent> is defined as follows:



The MP1800A/MT1810A uses the decimal integer format.

■ Integer format

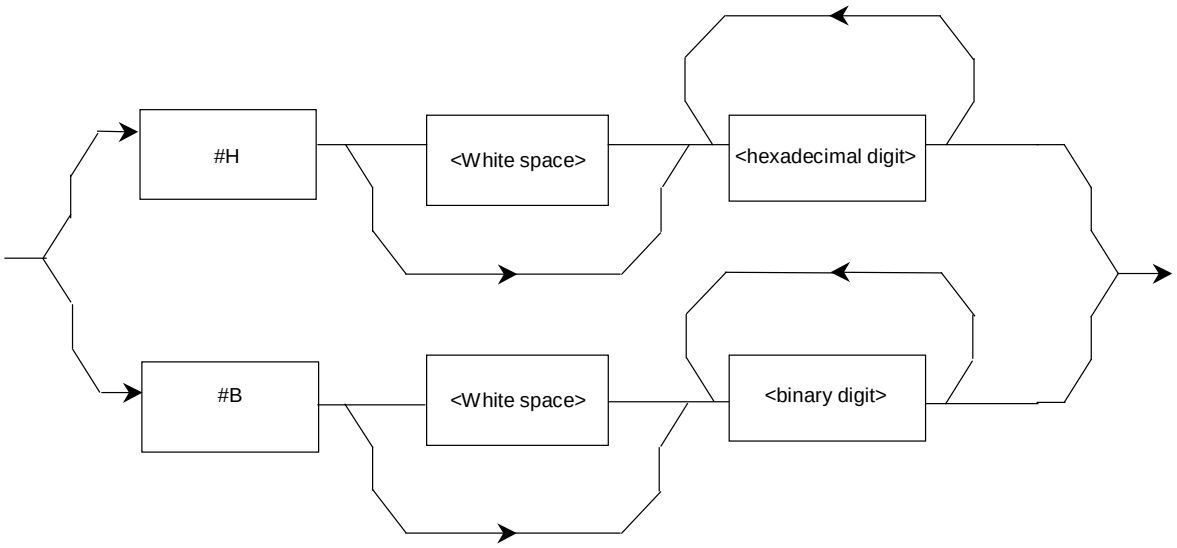
Δ represents a space.

- 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), +Δ5 (not applicable)
- Spaces can be inserted after the numerical value. → +5ΔΔ
- 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 NUERIC 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). Δ represents a space.

- Spaces can be inserted following #H.
  #H1234 → #HΔ1234
  #H00AF → #HΔΔΔ00AF
- Zeros can be omitted.
  #H00FF → #HFF
  #H0000 → #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.
Δ represents a space.

- Spaces can be inserted following #B.
  #B11011011 → #BΔ11011011
  #B00100100 → #BΔΔΔ00100100

3.1.3.4 <BOOLEAN PROGRAM DATA>

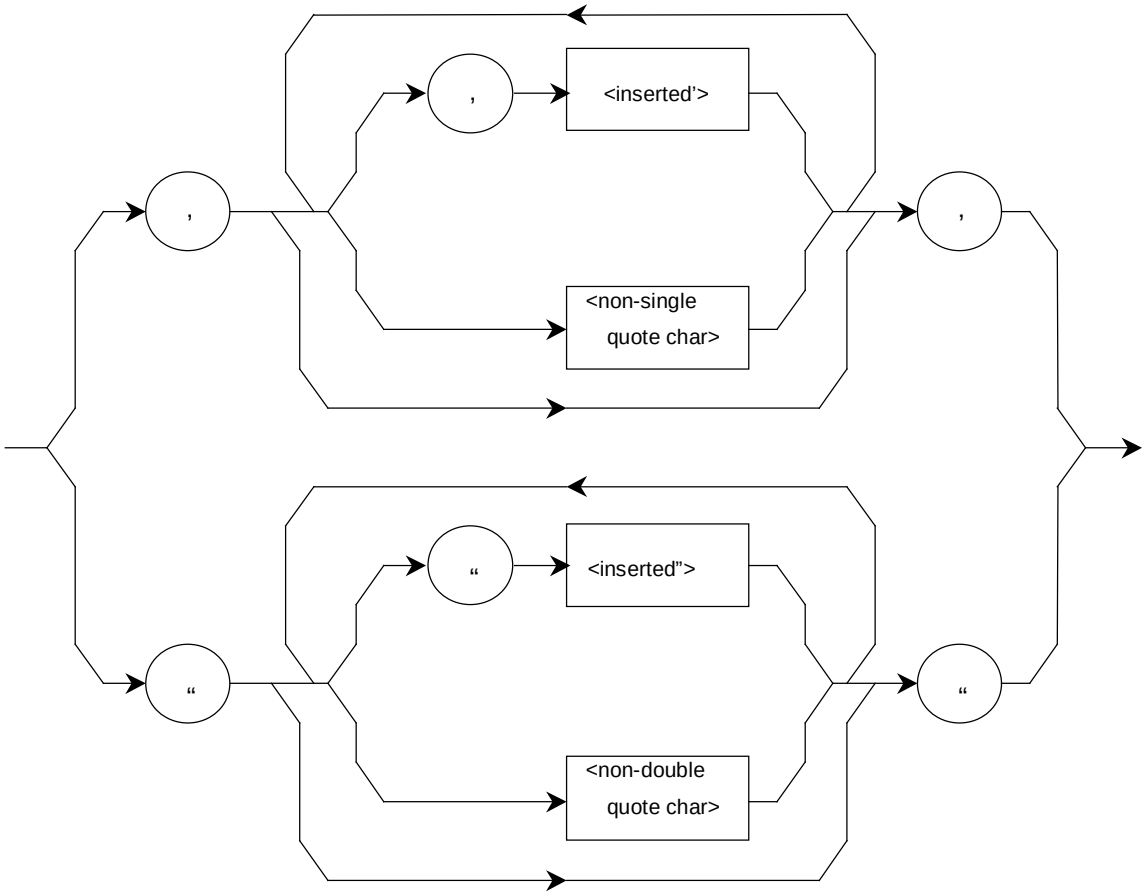
<BOOLEAN PROGRAM DATA> is the program data that is defined in the SCPI, and indicates a theoretical value. As the values corresponding to True and False, ON and OFF of <CHARACTER PROGRAM DATA> and 1 and 0 of <DECIMAL NUMREIC PROGRAM DATA> are defined.

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.

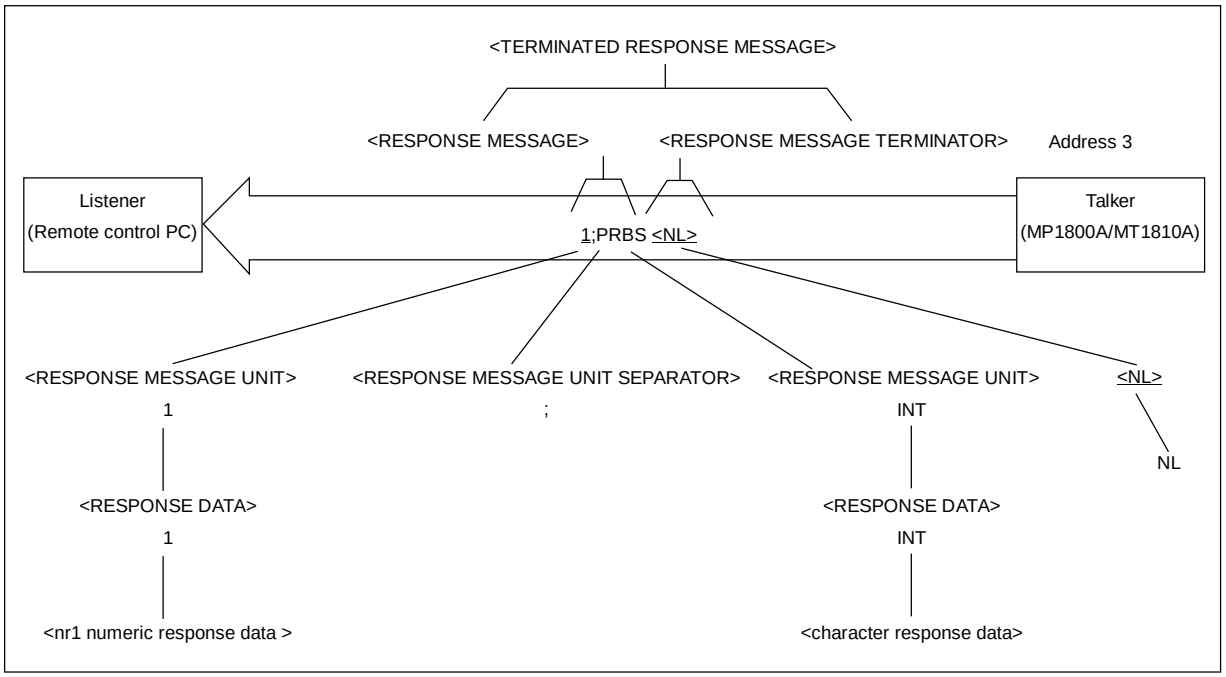


Figure 3.2.1-1 <TERMINATED RESPONS MESSAGE>

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.

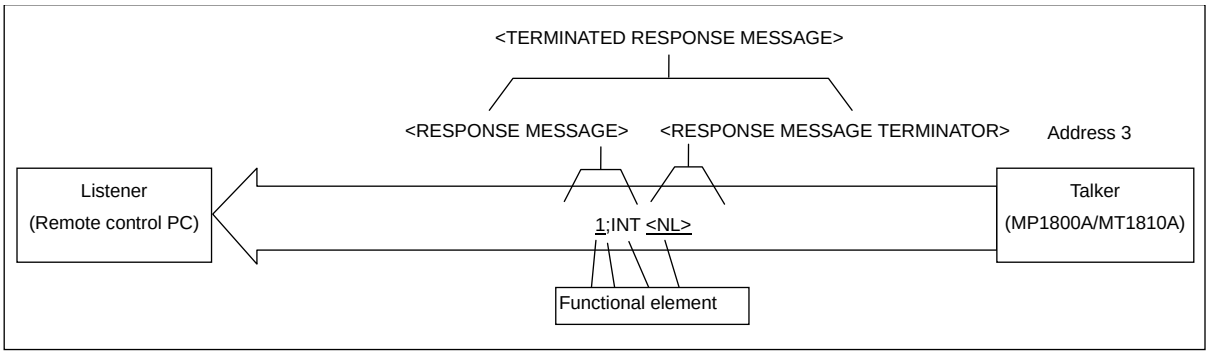
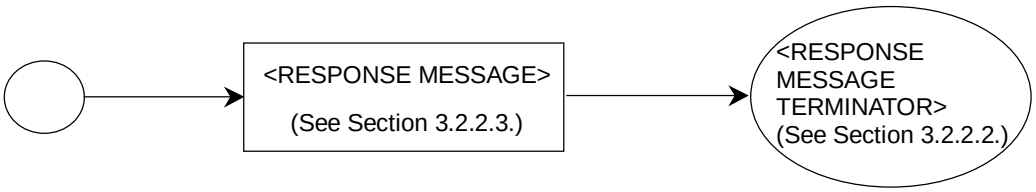


Figure 3.2.2.1-1 <TERMINATED RESPONSE MESSAGE>

Definition <TERMINATED RESPONSE MESSAGE> is defined as follows:

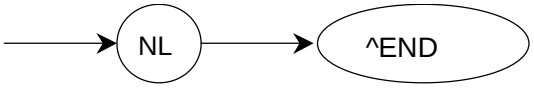


<TERMINATED RESPONSE MESSAGE> is a data message containing all the functional elements required for transmitting the message from the talker (MP1800A/MT1810A) to the remote control PC. A <RESPONSE MESSAGE TERMINATOR> is added to the end of a <RESPONSE MESSAGE> to complete transmission of <RESPONSE MESSAGE>.

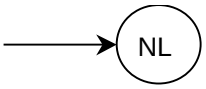
3.2.2.2 <RESPONSE MESSAGE TERMINATOR>

<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:



(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).

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.

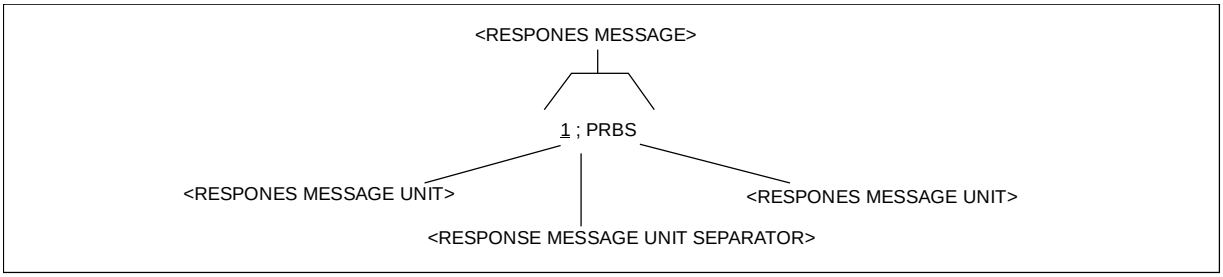
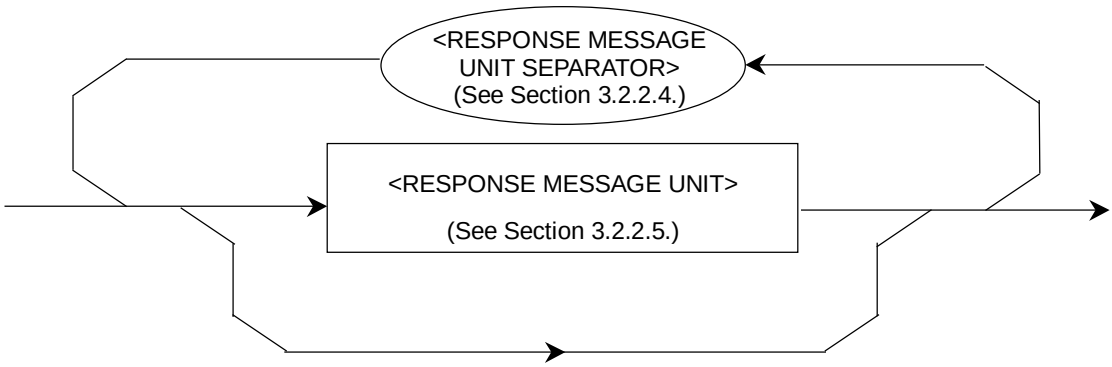


Figure 3.2.2.3-1 <RESPONSE MESSAGE>

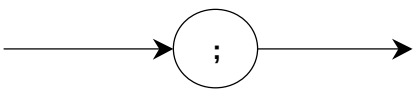
Definition
<RESPONSE MESSAGE> is defined as follows:



<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 <RESONSE MESSAGE UNIT SEPARATOR> element is used to separate two or more <RESPONSE MESSAGE UNIT> elements.

3.2.2.4 <RESPONSE MESSAGE UNIT SEPARATOR>

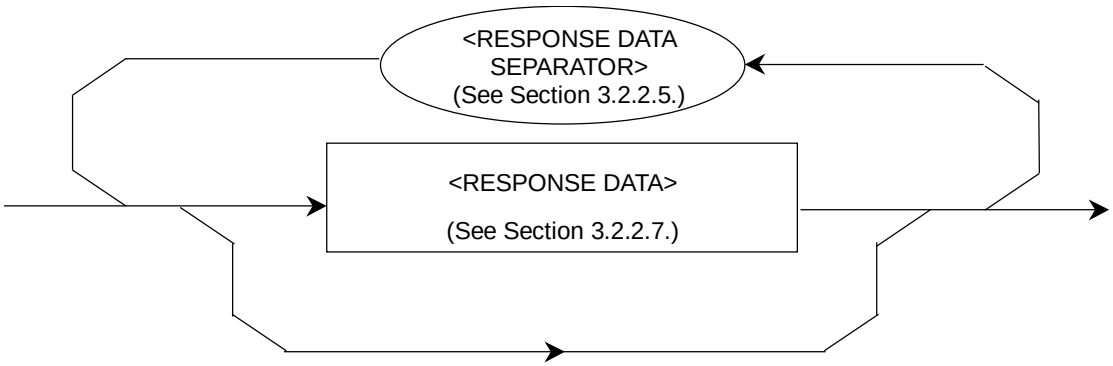
<RESPONSE MESSAGE UNIT SEPARATOR> is defined as follows:



<RESPONSE MESSAGE 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>

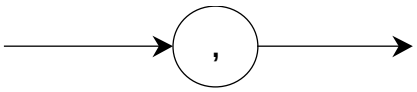
<RESPONSE MESSAGE UNIT> of the MP1800A/MT1810A is a response message unit with no header, and returns only the measurement result data. <RESPONSE MESSAGE UNIT> is defined as follows:



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:



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

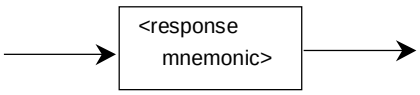
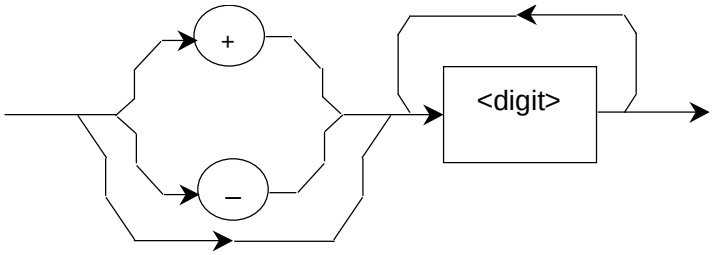
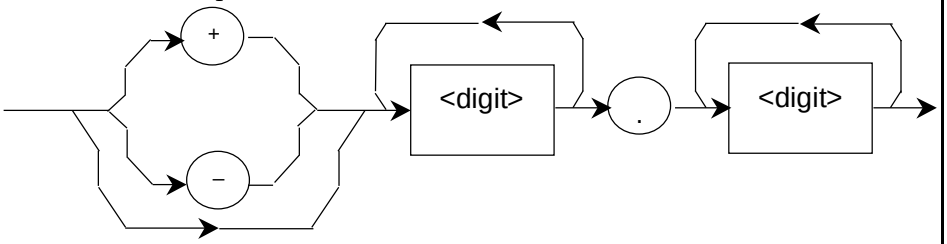
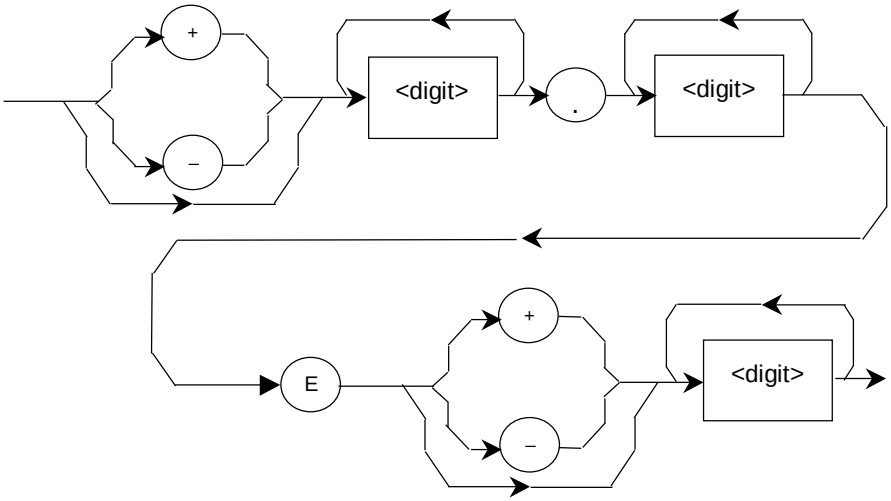
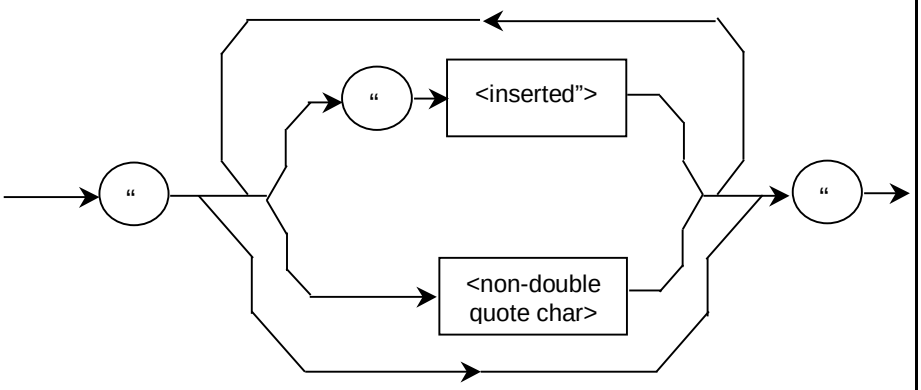
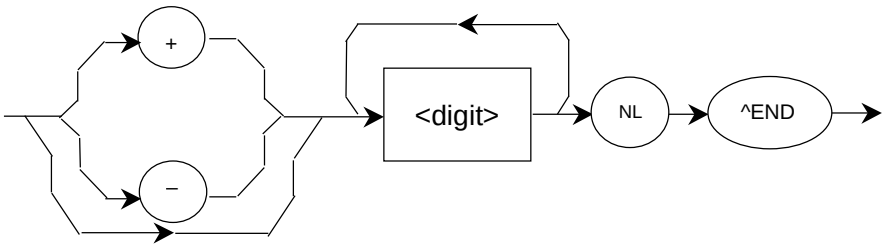
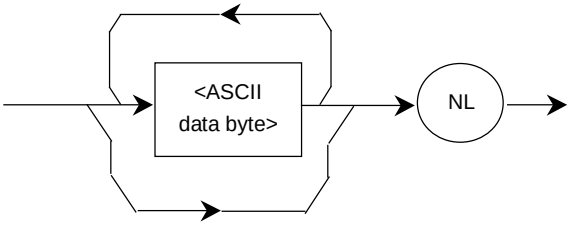
Element	Function
(1) CHARACTER RESPONSE DATA Example: ABC DEFG	Indicates short mnemonic data. 
(2) NR1 NUMERIC RESPONSE DATA Example: 123 +123 –1234	Indicates a decimal integer. 
(3) NR2 NUMERIC RESPONSE DATA Example: 12.3 +12.34 –12.345	Indicates a fixed-point numerical value. 
(4) NR3 NUMERIC RESPONSE DATA Example: 1.23E+45 –12.3E+45	Indicates a real decimal number with an exponent. 

Table 3.2.2.7-1 Response data (Cont'd)

Element	Function
(5) STRING RESPONSE DATA Example: "1234" "ABCD" "1234.5"	Indicates a character string enclosed within double quotation marks(" "). 
(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:  2) For Ethernet interface: 

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 3.2.2.8-1 Syntax differences

Item	Listener Input Program Message	Talker Output Program Message
Characteristics	Flexible	Strict
Alphabetic characters	Uppercase and lowercase characters have the same meanings (non case sensitive).	Uppercase characters only
Before and after the NR3 exponent part E	Δ+E, e+Δ (Δ is 0 or more)	Uppercase E only
Positive sign (+) of the NR3 exponent part	Can be omitted	Cannot be omitted
<white space> (: Δ)*1	Two or more <white space> can be added before and after the separator and before the terminator.	Not used
Unit separator	Δ + semicolon (Δ is 0 or more)	Semicolon only
Blank before header	Δ + header (Δ is 0 or more)	Header only
Header separator	Header + Δ (Δ is 1 or more)	Header + one $20*2
Data separator	Δ + comma + Δ (Δ is 0 or more)	Comma only
Terminator	Δ + {NL, EOI, or NL+EOI} (Δ is 0 or more)	NL+EOI*3

*1: Δ 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

:INPut	:CLOCk	:SELection	<clock>
		:SELection?	
		:RECovery	<string>
		:RECovery?	

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

:INPut:CLOCk:SELection <clock>
:INPut:CLOCk:SELection?
:INPut:CLOCk:RECovery <string>
:INPut:CLOCk:RECovery?

The SCPI command tree shown in Section 3.3 contains the SCPI commands above. The following describes rules on SCPI command description.

<Command format>
A command always starts with a colon (:). A command consists of headers connected by colons (:).

<Abbreviated format for headers>
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.

Example:

Long form	> :INPUT:CLOCK:SELECTION RECOVERED
Short form	> :INP:CLOC:SEL REC
Long + short form	> :Inp:CLOC:SELECTION REC

<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

Parameter Type	Description
<numeric> <DECIMAL NUMERIC PROGRAM DATA>	Indicates a decimal integer.
<numeric> <NON-DECIMAL NUMERIC PROGRAM DATA>	Indicates a numeric value other than a decimal (binary, octal, etc.).
<boolean> <BOOLEAN PROGRAM DATA>	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.
<string>,<display>…etc <STRING PROGRAM DATA>	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 "SES_3:DM_6"
<brate>,<type>…etc <CHARACTER PROGRAM DATA>	Indicates character data. It is expressed as a short string corresponding to the setting details.
<bdata>…etc <ARBITRARY BLOCK PROGRAM DATA>	Used to transmit 8-bit binary data.

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

Response Type	Description
<numeric> <NR1 NUMERIC RESPONSE DATA>	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. > :SYSTem:DATE? < 2006,7,14
<numeric> <NR2 NUMERIC RESPONSE DATA>	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. > :OUTPut:DATA:AMPLitude? < 1.000
<brate>,<type>…etc <CHARACTER RESPONSE DATA>	Returns the short form of the character. > :SOURce:SYNThesizer:JITTer:MODulation:SELect? < INT
<string>,<display>…etc <STRING RESPONSE DATA>	Returns a string enclosed within double quotation marks (" "). If there are short and long forms for the string of the corresponding program command, the short form is returned. > :DISPlay:CUSTomize:BUTTon:NOW? < 1, "EADD" (The short form of "EADDition")
<ARBITRARY ASCII RESPONSE DATA>	Returns 7-bit ASCII text data without separation. The last data byte is terminated by NL^END or NL only. > *OPT? < OPT301,OPT302,MU181000A,OPT101,…

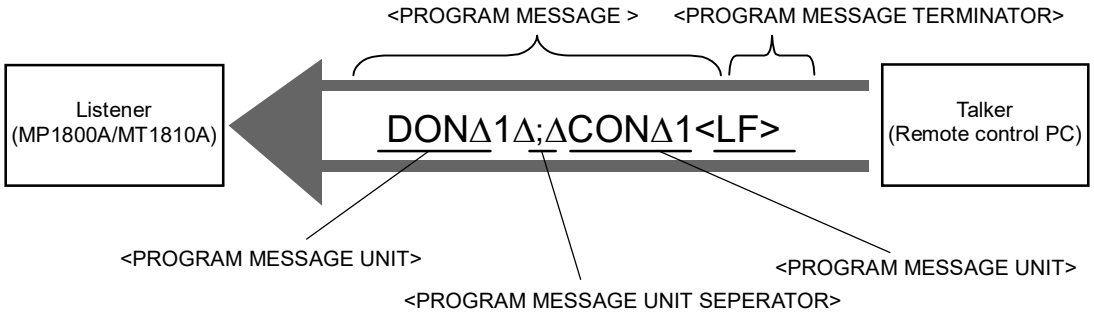
Chapter 4 Native Format

This chapter describes the Native command system.

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).

■ Separator

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

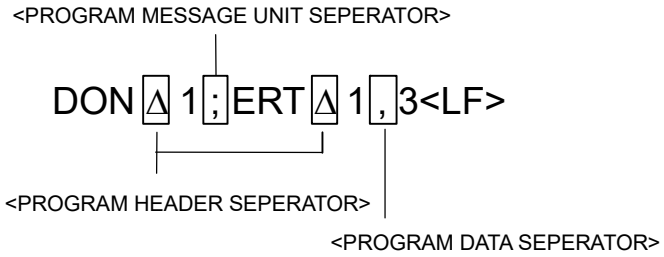


Table 4.1-1 Separator functional elements

Functional Element	Description
<PROGRAM MESSAGE UNIT SEPARATOR>	Separates two or more <PROGRAM MESSAGE UNIT> elements. It is indicated as 0 or more spaces + one semicolon.
<PROGRAM DATA SEPARATOR>	Separates two or more <PROGRAM DATA> elements. It is indicated as 0 or more spaces + one comma + 0 or more spaces.
<PROGRAM HEADER SEPARATOR>	Separates a program header and <PROGRAM DATA> element. It is indicated as one or more spaces.

■ Terminator

Terminators have the functional element shown in the table below.

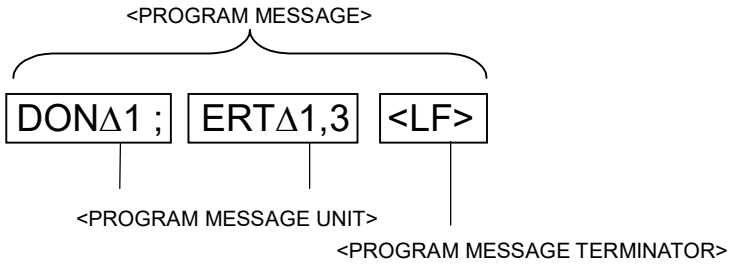


Table 4.1-2 Terminator functional element

Functional Element	Description
<PROGRAM MESSAGE TERMINATOR>	Added at the end of <PROGRAM MESSAGE> to terminate a sequence with one or more <PROGRAM MESSAGE UNIT> 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.

■ Program header

Program headers have the two functional elements shown in the table below.

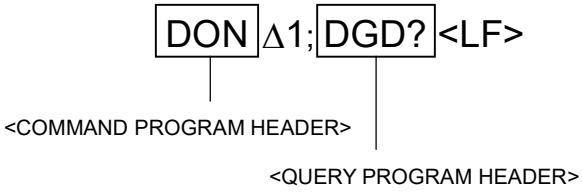


Table 4.1-3 Program header functional elements

Functional Element	Description
<COMMAND PROGRAM HEADER>	This is a command used by the remote control PC to control the MP1800A/MT1810A. The <COMMAND PROGRAM HEADER> of the MP1800A/MT1810A uses <simple command program header> as a coding syntax. <simple command program header> The device-unique commands use this syntax. Example: DGDΔ1
<QUERY PROGRAM HEADER>	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 <QUERY PROGRAM HEADER> of the MP1800A/MT1810A uses <simple query program header> as a coding syntax. <simple query program header> The device-unique query commands use this syntax. Example: DGD?

■ 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.

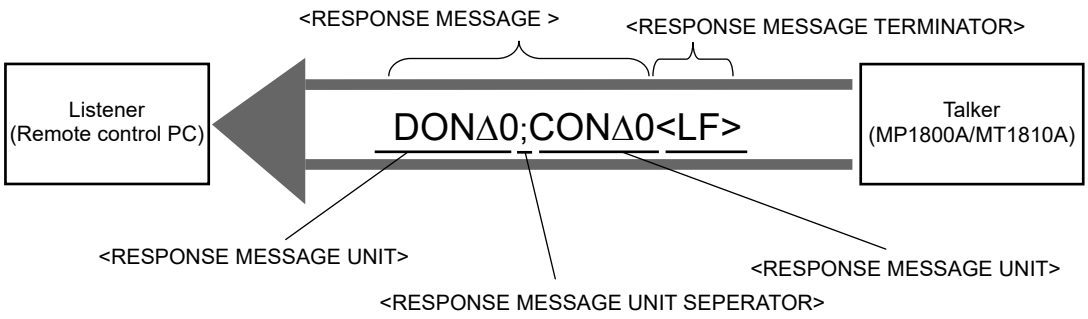
Table 4.1-4 Program data functional elements

Functional Element	Description
<DECIMAL NUMERIC PROGRAM DATA>	Indicates a decimal integer/real number. Integers and fixed-point values are used in the MP1800A/MT1810A. Example: Integers 005, +5, 5, +5Δ Fixed-point values 12.345, 05, +0.05, 12.
<NON-DECIMAL NUMERIC PROGRAM DATA>	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. Example: #HABC, #H123 #B101010
<ARBITRARY BLOCK PROGRAM DATA>	Used to transmit binary data of the specified number of bytes. Example: #11A

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 4.2-1 Syntax differences

Item	Listener Input Program Message	Talker Output Response Message
Characteristics	Flexible	Strict
Alphabetic characters	Not case-sensitive except for the header Only uppercase characters for the header	Uppercase characters only
Before and after the NR3 exponent part E	Uppercase E only	Uppercase E only
Positive sign (+) of the NR3 exponent part	+ can be omitted	Cannot be omitted.
Space	Spaces can be added before and after the separator and after the terminator	Not used
Message unit	[1] Header with program data [2] Header with no program data	[1] Data with a header [2] Data with no header
Unit separator	0 or more spaces + semicolon	Semicolon only
Blank before header	0 or more spaces + header	Header only
Header separator	Header + one or more spaces	Header + one $20*1
Data separator	0 or more spaces + comma + 0 or more spaces	Not used
Terminator	0 or more spaces + any one of the following: { LF / CR+LF / LF+EOI / CR+LF+EOI }	Either LF+EOI or CR+LF+EOI*2

*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.

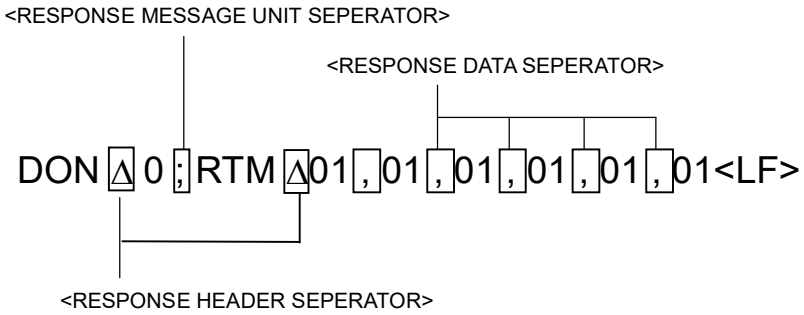


Table 4.2-2 Separator functional elements

Functional element	Description
<RESPONSE MESSAGE UNIT SEPARATOR>	Separates two or more <RESPONSE MESSAGE UNIT> elements. It is indicated as a semicolon.
<RESPONSE HEADER SEPARATOR>	Separates <RESPONSE HEADER> and <RESPONSE DATA> elements. It is indicated as one space.
<RESPONSE DATA SEPARATOR>	Separates two or more <RESPONSE DATA> elements. It is indicated as a comma.

■ Terminator

Terminators have the functional element shown in the table below.

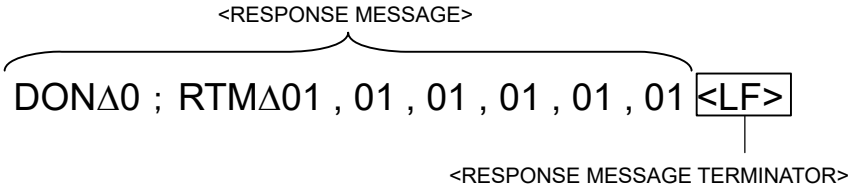


Table 4.2-3 Terminator functional element

Functional Element	Description
<RESPONSE MESSAGE TERMINATOR>	Added at the end of <RESPONSE MESSAGE> to terminate message transfer. It is indicated as LF or CR+LF (For the GPIB, EOI is asserted at the same time as the transmission of LF).

■ Response header

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

Table 4.2-4 Response header functional element

Functional Element	Description
<RESPONSE HEADER>	Up to 12 uppercase alphanumeric characters or asterisks that indicate the function of the response data starting with an uppercase character. Example: DONΔ0

■ 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 4.2-5 Response data functional elements

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

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 4.4-1 Description of parameter types

Parameter Type	Description
<numeric> <DECIMAL NUMERIC PROGRAM DATA>	Indicates a decimal integer/real number. Integers, fixed-point values, and exponents are used in the MP1800A/MT1810A.
<numeric> <NON-DECIMAL NUMERIC PROGRAM DATA>	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 indicates as #B + 0/1.
<bdata>...etc. <ARBITRARY BLOCK PROGRAM DATA>	Used to transmit 8-bit binary data.

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 4.5-1 Description of response types

Response Type	Description
<numeric> <NR1 NUMERIC RESPONSE DATA>	Indicates a decimal integer.
<numeric> <NR2 NUMERIC RESPONSE DATA>	Indicates a decimal fixed-point value.
<numeric> <NR3 NUMERIC RESPONSE DATA>	Indicates a decimal exponent.
<numeric> <HEXADECIMAL NUMERIC RESPONSE DATA>	Indicates a hexadecimal numeric value.
<string> <STRING RESPONSE DATA>	Indicated as a character string enclosed within double quotation marks.
<bdata>...etc. <DEFINITE LENGTH ARBITRARY BLOCK RESPONSE DATA>	Indicates 8-bit binary data of a known length.

Chapter 5 IEEE488.2 Common Commands

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

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 5.1-1 IEEE488.2 Common Commands List

Mnemonic	Command's full spell
*IDN?	Identification Query
*RST	Reset Command
*OPC	Operation Complete Command
*OPC?	Operation Complete Query
*WAI	Wait Continue Command
*CLS	Clear Status Command
*ESE	Standard Event Status Enable Command
*ESE?	Standard Event Status Enable Query
*ESR?	Standard Event Status Register Query
*SRE	Service Request Enable Command
*SRE?	Service Request Enable Query
*STB?	Read Status Byte Query
*TRG	Trigger Command
*OPT?	Option Identification Query

*IDN? Identification Query

Parameter	None
Response	<Manufacturer>, <Model>, <Serial No.> <Manufacturer>, <Model> = <CHARACTER RESPONSE DATA> MT1800A, MT1810A <Serial No.> = <NR1 NUMERIC RESPONSE DATA> 0000000000 to 9999999999 Main frame Serial number
Function	Reports manufacture name, model, etc.
Example	`> *IDN?` `< ANRITSU,MP1800A,0123456789`

*RST Reset Command

Parameter	None
Function	Initializes entire system. Cancels the measurement and clear the contents. To reset to the factory default, perform SCPI command :SYSTem:MEMory:INITialize
Example	`> *RST`

*OPC Operation Complete Command

Parameter	None
Function	Sets bit 0 (operation complete bit) of standard event status register and sets SRQ to ON, when execution of preceding command is completed.
Example	`> *OPC`

*OPC? Operation Complete Query

Parameter	None
Response	<NR1 NUMERIC RESPONSE DATA> 1
Function	Returns 1 when preceding command is completed.
Example	`> *OPC?` `< OPC 1`

*WAI Wait to Continue Command

Parameter	None
Function	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.
Example	`> *WAI`

*CLS Clear Status Command

Parameter	None
Function	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 <PROGRAM MESSAGE TERMINATOR> and before <QUERY MESSAGE UNIT> element. Execution of succeeding commands is set to wait until execution of the preceding command is completed.
Example	`> *CLS`

*ESE Standard Event Status Enable Command

Parameter
<DECIMAL NUMERIC PROGRAM DATA>
An integer between 0 and 255
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 (2^7 = 128) Power On
Bit 5 (2^5 = 32) Command error
Bit 4 (2^4 = 16) Execution error
Bit 3 (2^3 = 8) Errors other than command, query and execution errors
Bit 0 (2^0 = 1) Completion of operation

Function
Sets or clears standard event status enable register.

Example
To set bits 3 (= 8) and 4 (= 16) of enable register:

```
> *ESE 24
```

*ESE? Standard Event Status Enable Query

Parameter
None

Response
<NR1 NUMERIC RESPONSE DATA>
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.

Function
Queries current value of standard event status enable register.

Example

```
> *ESE?
< ESE 24
```

*ESR? Standard Event Status Register Query

Parameter
None

Response
<NR1 NUMERIC RESPONSE DATA>
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.

Function
Queries current value of standard event status register.

Example
When a command error exists:

```
> *ESR?
< ESR 32
```

*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
```

*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 5.1-2 Option Character List (MP1800A/MT1810A)

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

Table 5.1-2 Option Character List(MP1800A/MT1810A) (Cont'd)

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

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

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

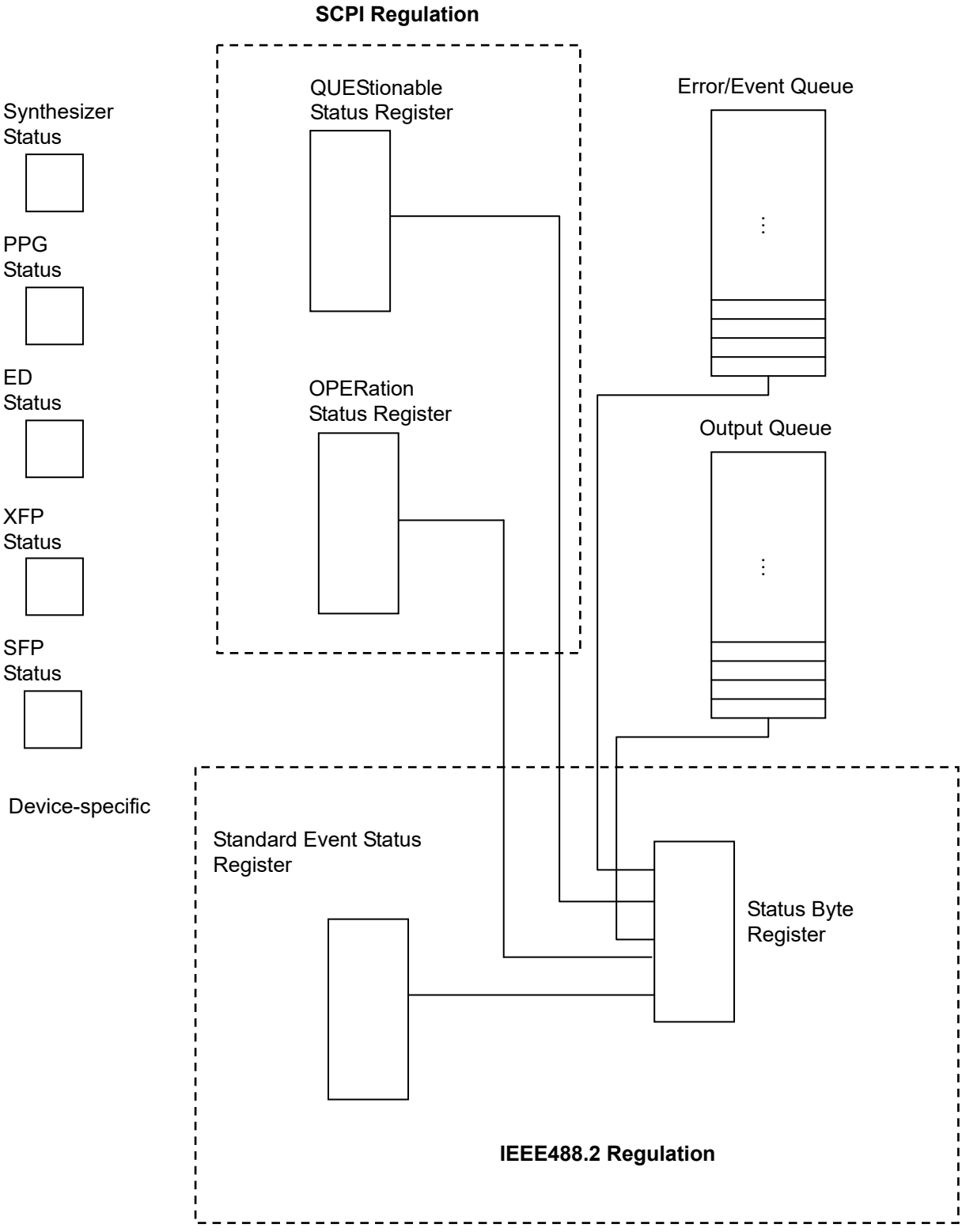


Figure 6.1-1 Status Register Configuration

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

Name	Description
Status byte register	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.
Standard event status register	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.

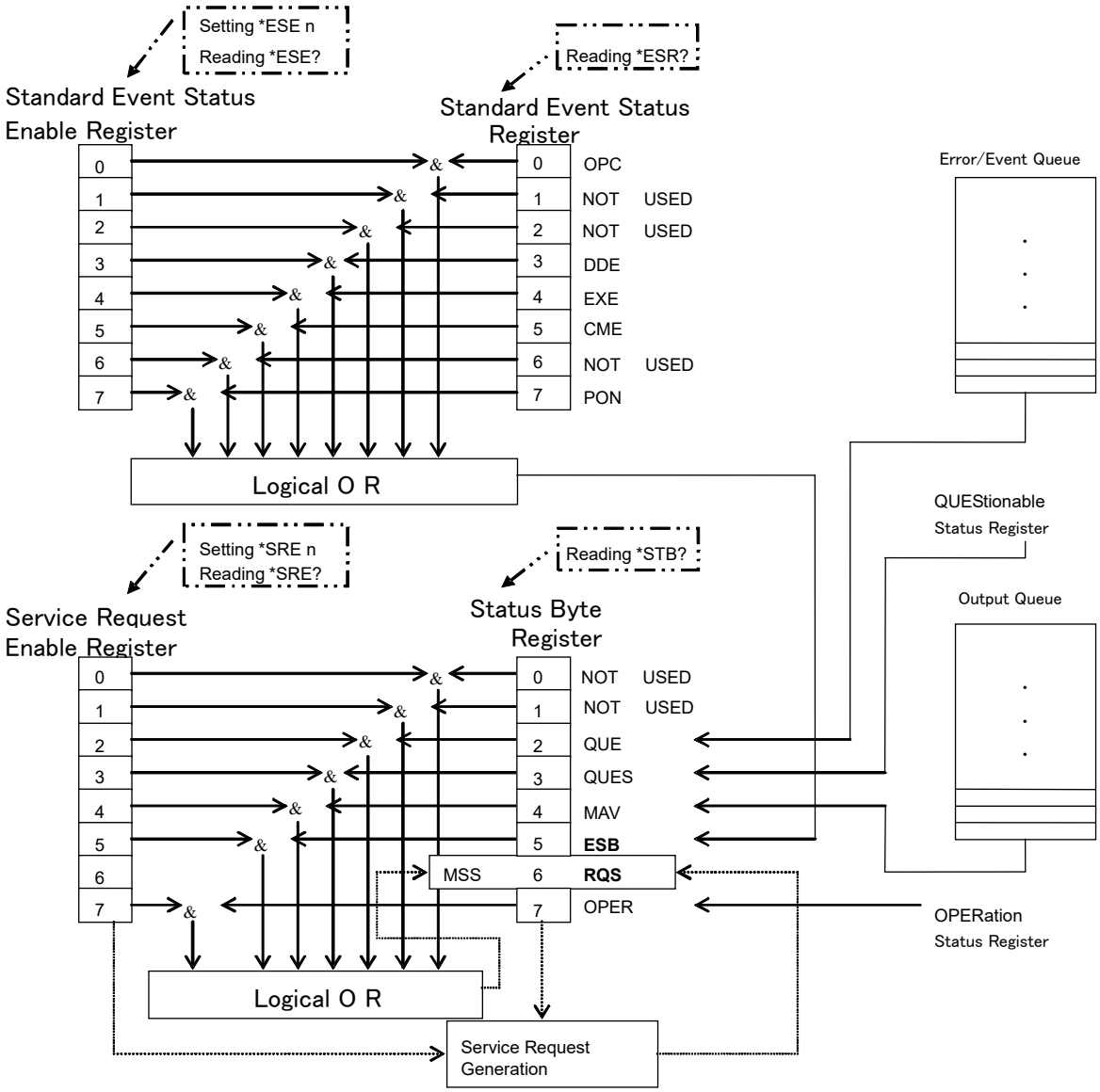


Figure 6.2-1 IEEE488.2 Status Register

Table 6.2-2 Status Byte Register Bit Definition

Bit	Mnemonic	Description
DB2	QUE (error/event QUEue)	Indicates that the error and event queues not empty.
DB3	QUES (QUEStionable status register summary)	QUEStionable status register summary.
DB4	MAV (Message AVailable)	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.
DB5	ESB (Event Summary Bit)	Standard event status register summary
DB6	RQS (ReQuest Service)	Returns a 7-bit status byte and this RQS to the controller in the serial pole mode.
	MSS (Master Summary Status)	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.
DB7	OPER (OPERation status register summary)	OPERation status register summary

Table 6.2-3 Standard Event Status Register Bit Definition

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

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

Name	Description
QUEStionable Status register	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.
OPERation Status register	Reports some MP1800A/MT1810A statuses.

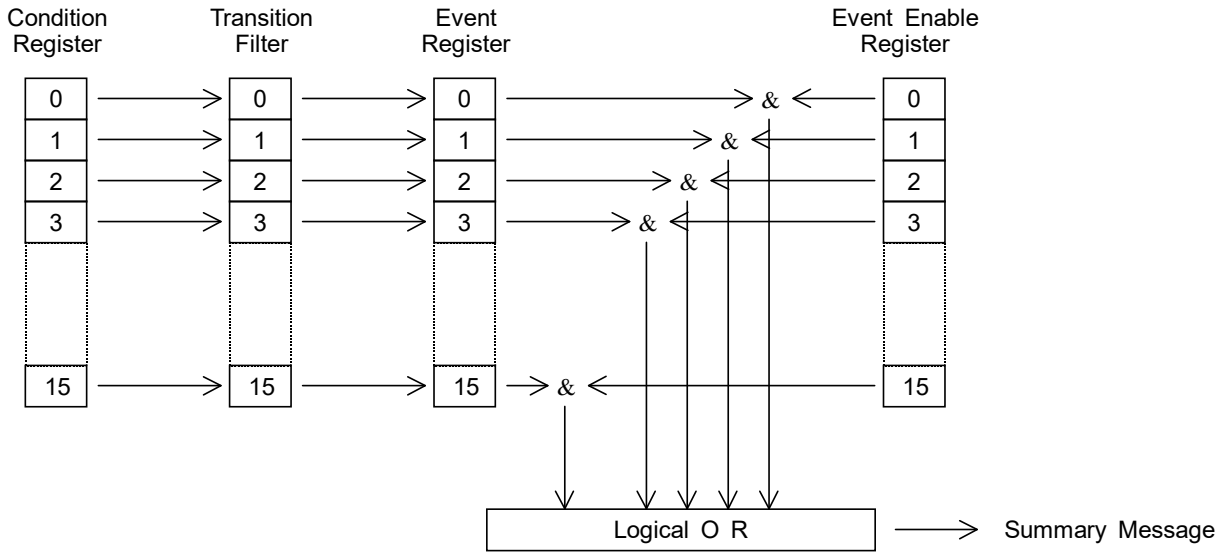


Figure 6.3-1 OPERation Status Register

Table 6.3-2 OPERation Status Register Bit Definition

Bit	Mnemonic	Description
DB3	BAT (BAThtub measuring)	Indicates that the Instrument is measuring Bathtub.
DB4	MEAS (MEASuring)	Indicates that the Instrument is measuring when any port is in execution.*
DB5	ISI (ISI)	Indicates that the Instrument is measuring ISI.
DB6	QAN (Q ANalysis)	Indicates that the Instrument is measuring Q Analysis
DB7	AAD (Auto ADjust)	Indicates that the Instrument is in Auto Adjust.
DB8	ASE (Auto SEarching)	Indicates that the Instrument is in Auto Search.
DB9	EMM (Eye Margin Measuring)	Indicates that the Instrument is measuring eye margin.

Table 6.3-2 OPERation Status Register Bit Definition (Cont'd)

Bit	Mnemonic	Description
DB10	EDI (Eye Diagram)	Indicates that the Instrument is measuring eye diagram.
DB11	PSET (Pattern SETing)	Indicates that the Instrument is setting pattern.
DB12	INI (INItialize)	Indicates that the Instrument is in initialization.

*: 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 6.4-1 Reading and Writing the Status Register

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

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

Register	*RST	*CLS	Power ON	STATus: PRESet	Other method to clear register
Status byte register	No change	Clear	Clear	No change	
Service request enable register	No change	No change	Clear	No change	Executing *SRE 0
Standard event status register	No change	Clear	Clear*2	No change	Cleared when an event is read by *ESR?
Standard event status enable register	No change	No change	Clear*1	No change	Executing *ESE 0
SCPI event register	No change	Clear	Clear*1	No change	Cleared when an event is read by :STATus:...:EVENt?
SCPI enable register	No change	No change	Reset*1	Reset	Executing :STATus:...:ENABle 0
SCPI Transition filter	No change	Reset	Reset*1	Reset	Executing :STATus:...:PTRansition 0 and :STATus:...:NTRansition 0
Error/event queue	No change	Clear	Clear	No change	Reading all events by :SYSTem:ERRor?

*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

Register	Enable/filter	Reset value
OPERational status register	Enable register PTRansition filter NTRansition filter	All 0 All 1 All 0

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.

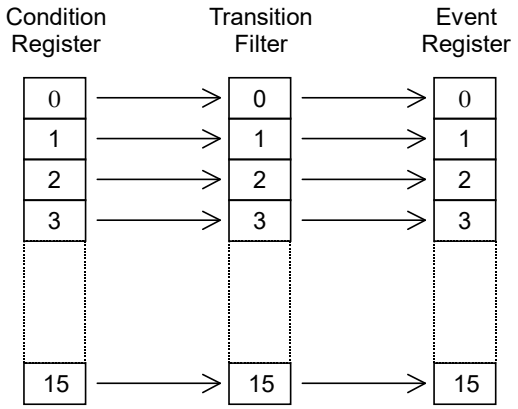


Figure 6.6-1 Each Status Configuration

- 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.

The following show module status bit definition.

Table 6.6-1 Module Status

Status	Contents
Synthesizer Status	Reports MU181000A 12.5GHz Synthesizer status and MU181000B 12.5GHz 4port Synthesizer status.
PPG Status	Reports MU181020A 12.5Gbit/s PPG status and MU181020B 14Gbit/s PPG status.
ED Status	Reports MU181040A 12.5Gbit/s ED status and MU181040B 14Gbit/s ED status.
Optical Transceiver (XFP) Status	Reports MU181600A Optical Transceiver (XFP) status.
Optical Transceiver (SFP) Status	Reports MU181601A Optical Transceiver (SFP) status.
MUX Status	MU182020A 25 Gbit/s 1Ch MUX/ MU182021A 25 Gbit/s 2Ch MUX Status Report
DEMUX Status	MU182040A 25Gbit/s 1Ch DEMUX/ MU182041A 25Gbit/s 2Ch DEMUX Status Report

Table 6.6-2 Synthesizer Status Bit Definition

BIT	Contents
DB0	Indicates PLL Unlock occurred.

Table 6.6-3 ED Status Bit Definition

BIT	Contents
DB0	Indicates Insertion Error was detected.
DB1	Indicates Omission Error was detected.
DB2	Indicates Total Error was detected.
DB4	Indicates Pattern Sync Loss occurred.
DB5	Indicates Clock Loss occurred.
DB6	Indicates Delay Busy occurred.
DB8	Indicates Transition Error was detected.
DB9	Indicates Non Transition Error was detected.
DB10	Indicates CR Unlock occurred.
DB11	Indicates Delay Calibration Require occurred.

Table 6.6-4 PPG Status Bit Definition

BIT	Contents
DB1	Indicates CMU-MUX Unlock occurred.
DB2	Indicates Delay Busy occurred.
DB4	Indicates Delay Calibration Require occurred.

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

BIT	Contents
DB0	Indicates Ready status.
DB1	Indicates Laser Fault occurred.
DB2	Indicates Unlock occurred.
DB3	Indicates LOS occurred.
DB4	Indicates CDR Unlock occurred.

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

BIT	Contents
DB0	Indicates Ready status.
DB2	Indicates LOS occurred.

Table 6.6-7 DEMUX Status Bit Definition

BIT	Contents
DB6	Displays Delay Busy occurrences at Data1
DB7	Displays Delay Busy occurrences at Data2
DB11	Displays Delay Calibration Require occurrences at Data1
DB12	Displays Delay Calibration Require occurrences at Data2

Table 6.6-8 MUX Status Bit Definition

BIT	Contents
DB2	Displays Delay Busy occurrences at Data1
DB3	Displays Delay Busy occurrences at Data2
DB4	Displays Delay Calibration Require occurrences at Data1
DB5	Displays Delay Calibration Require occurrences at Data2

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	<numeric> = <NR1 NUMERIC RESPONSE DATA>	
	0 to 8184	Sum total of the bit of event register (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 loading in progress
	4096 (Bit 12)	Initialization in progress
Function	Queries event register at OPERation Status Register.	
Example	`> :STATus:OPERation:EVENt?`	
	or	
	`> :STATus:OPERation?`	
	`< 16`	

:STATus:OPERation:CONDition?

Response	<numeric> = <NR1 NUMERIC RESPONSE DATA>	
	0 to 8184	Sum total of the bit of condition register (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 loading in progress
	4096 (Bit 12)	Initialization in progress
Function	Queries condition register at OPERation Status Register.	
Example	`> :STATus:OPERation:CONDition?`	
	`< 16`	

:STATus:OPERation:ENABle <numeric>

Parameter	<numeric> = <DECIMAL NUMERIC PROGRAM DATA>	
	0 to 8184	Sum total of the bit of event enable register (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
	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?

Response	<numeric> = <NR1 NUMERIC RESPONSE DATA>	
	0 to 8184	Sum total of the bit of event enable register (DECIMAL)
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`	

: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.

:INSTrument:SYG125[:EVENt]?

Response <numeric> = <NR1 NUMERIC RESPONSE DATA>

0 to 1 Sum total of the event bit (DECIMAL)

Available bit

1 (Bit 0) PLL Unlock occurs

Function Queries event at 12.5 GHz Synthesizer.

Example
```
> :INSTrument:SYG125:EVENt?
```
or
```
> :INSTrument:SYG125?
< 1
```

:INSTrument:SYG125:CONDition?

Response <numeric> = <NR1 NUMERIC RESPONSE DATA>

0 to 1 Sum total of the condition bit (DECIMAL)

Available bit

1 (Bit 0) PLL Unlock occurs

Function Queries condition at 12.5 GHz Synthesizer.

Example
```
> :INSTrument:SYG125:CONDition?
< 1
```

:INSTrument:SYG125:PTRansition <numeric>

Parameter <numeric> = <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 (positive direction change) of the 12.5 GHz Synthesizer Status.

Example To set the transition filter (positive direction change) of the 12.5 GHz Synthesizer Status to 1.
```
> :INSTrument:SYG125:PTRansition 1
```

:INSTrument:SYG125:PTRansition?

Response	<numeric> = <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	<numeric> = <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:NTRansition?

Response	<numeric> = <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:NTRansition?`	
	`< 1`	

:INSTrument:SYG125:RESet

Function	Initializes event at 12.5 GHz Synthesizer.
Example	`> :INSTrument:SYG125:RESet`

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]?

Response

<numeric> = <NR1 NUMERIC RESPONSE DATA>

0 to 3959	Sum total of the event bit (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

Queries event at 12.5 Gbit/s ED status.

Example

```
> :INSTrument:EDG125:EVENt?
```

or

```
> :INSTrument:EDG125?
< 1
```

:INSTrument:EDG125:CONDition?

Response

<numeric> = <NR1 NUMERIC RESPONSE DATA>

0 to 3959	Sum total of the condition bit (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

Queries condition at 12.5 Gbit/s ED status.

Example

```
> :INSTrument:EDG125:CONDition?
< 1
```

:INSTrument:EDG125:PTRansition <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 (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	<numeric> = <NR1 NUMERIC RESPONSE DATA>	
	0 to 3959	Sum total of the bit of transition filter (DECIMAL)
Function	Queries the transition filter (positive direction change) of the 12.5 Gbit/s ED Status.	
Example	`> :INSTrument:EDG125:PTRansition?`	
	`< 1`	

: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`

6.7.5 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]?

Response	<numeric> = <NR1 NUMERIC RESPONSE DATA>	
	0 to 22	Sum total of the event bit (DECIMAL)
	Available bit	
	2 (Bit 1)	CMU-MUX Unlock occurs
	4 (Bit 2)	Delay Busy occurs
	16 (Bit 4)	Delay Calibration Require occurs
Function	Queries event at 12.5 Gbit/s PPG status.	
Example	`> :INSTrument:PPGG125:EVENt?`	
	or	
	`> :INSTrument:PPGG125?`	
	`< 1`	

:INSTrument:PPGG125:CONDition?

Response	<numeric> = <NR1 NUMERIC RESPONSE DATA>	
	0 to 22	Sum total of the condition bit (DECIMAL)
	Available bit	
	2 (Bit 1)	CMU-MUX Unlock occurs
	4 (Bit 2)	Delay Busy occurs
	16 (Bit 4)	Delay Calibration Require occurs
Function	Queries condition at 12.5 Gbit/s PPG status.	
Example	`> :INSTrument:PPGG125:CONDition?`	
	`< 1`	

:INSTrument:PPGG125:PTRansition <numeric>

Parameter	<numeric> = <DECIMAL NUMERIC PROGRAM DATA>	
	0 to 22	Sum total of the bit of transition filter (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 (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`	

: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	<numeric> = <NR1 NUMERIC RESPONSE DATA>	
	0 to 31	Sum total of the event bit (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	Queries event at Optical Transceiver (XFP) status.	
Example	`> :INSTrument:XFP:EVENt?`	
	or	
	`> :INSTrument:XFP?`	
	`< 1`	

:INSTrument:XFP:CONDition?

Response	<numeric> = <NR1 NUMERIC RESPONSE DATA>	
	0 to 31	Sum total of the condition bit (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	Queries condition at Optical Transceiver (XFP) status.	
Example	`> :INSTrument:XFP:CONDition?`	
	`< 1`	

: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?

Response	<numeric> = <NR1 NUMERIC RESPONSE DATA>	
	0 to 31	Sum total of the bit of transition filter (DECIMAL)
Function	Queries the transition filter (negative direction change) of the Optical Transceiver (XFP) Status.	
Example	`> :INSTrument:XFP:NTRansition?`	
	`< 1`	

:INSTrument:XFP:RESet

Function	Initializes event at Optical Transceiver (XFP) status.
Example	`> :INSTrument:XFP:RESet`

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	<numeric> = <NR1 NUMERIC RESPONSE DATA>	
	0 to 5	Sum total of the event bit (DECIMAL)
	Available bit	
	1 (Bit 0)	Ready
	4 (Bit 2)	LOS occurs
Function	Queries event at Optical Transceiver (SFP) status.	
Example	`> :INSTrument:SFP:EVENt?`	
	or	
	`> :INSTrument:SFP?`	
	`< 1`	

:INSTrument:SFP:CONDition?

Response	<numeric> = <NR1 NUMERIC RESPONSE DATA>	
	0 to 5	Sum total of the condition bit (DECIMAL)
	Available bit	
	1 (Bit 0)	Ready
	4 (Bit 2)	LOS occurs
Function	Queries condition at Optical Transceiver (SFP) status.	
Example	`> :INSTrument:SFP:CONDition?`	
	`< 1`	

:INSTrument:SFP:PTRansition <numeric>

Parameter	<numeric> = <DECIMAL NUMERIC PROGRAM DATA>	
	0 to 5	Sum total of the bit of transition filter (DECIMAL)
	Available bit	
	1 (Bit 0)	Ready
	4 (Bit 2)	LOS occurs
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?

Response	<numeric> = <NR1 NUMERIC RESPONSE DATA>	
	0 to 5	Sum total of the bit of transition filter (DECIMAL)
Function	Queries the transition filter (positive direction change) of the Optical Transceiver (SFP) Status.	
Example	`> :INSTrument:SFP:PTRansition?`	
	`< 1`	

:INSTrument:SFP:NTRansition <numeric>

Parameter	<numeric> = <DECIMAL NUMERIC PROGRAM DATA>	
	0 to 5	Sum total of the bit transition filter (DECIMAL)
	Available bit	
	1 (Bit 0)	Ready
	4 (Bit 2)	LOS occurs
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.	
	`> :INSTrument:SFP:NTRansition 1`	

:INSTrument:SFP:NTRansition?

Response	<numeric> = <NR1 NUMERIC RESPONSE DATA>	
	0 to 5	Sum total of the bit of transition filter (DECIMAL)
Function	Queries the transition filter (negative direction change) of the Optical Transceiver (SFP) Status.	
Example	`> :INSTrument:SFP:NTRansition?`	
	`< 1`	

:INSTrument:SFP:RESet

Function	Initializes event at Optical Transceiver (SFP) status.
Example	`> :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]?

Response	<numeric> = <NR1 NUMERIC RESPONSE DATA>	
	0 to 6336	Sum total of the event bit (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	Queries event at 25Gbit/s DEMUX status.	
Example	`> :INSTrument:DMUX:EVENt?`	
	or	
	`> :INSTrument:DMUX?`	
	`< 64`	

:INSTrument:DMUX:CONDition?

Response	<numeric> = <NR1 NUMERIC RESPONSE DATA>	
	0 to 6336	Sum total of the condition bit (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	Queries condition at 25Gbit/s DEMUX status.	
Example	`> :INSTrument:DMUX:CONDition?`	
	`< 64`	

: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`	

:INSTrument:DMUX:NTRansition?

Response	<numeric> = <NR1 NUMERIC RESPONSE DATA>	
	0 to 6336	Sum total of the bit of transition filter (DECIMAL)
Function	Queries the transition filter (negative direction change) of the 25Gbit/s DEMUX Status.	
Example	`> :INSTrument:DMUX:NTRansition?`	
	`< 64`	

:INSTrument:DMUX:RESet

Function	Initializes event at 25Gbit/s DEMUX status.
Example	`> :INSTrument:DMUX:RESet`

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]?

Response	<numeric> = <NR1 NUMERIC RESPONSE DATA>	
	0 to 60	Sum total of the event bit (DECIMAL)
	Available bit	
	4 (Bit2)	Delay Busy Occurrence (Data1)
	8 (Bit3)	Delay Busy Occurrence (Data2)
	16 (Bit4)	Delay Calibration Require Occurrence (Data1)
	32 (Bit5)	Delay Calibration Require Occurrence (Data2)
Function	Queries event at 25Gbit/s MUX status.	
Example	`> :INSTrument:MUX:EVENt?`	
	or	
	`> :INSTrument:MUX?`	
	`< 4`	

:INSTrument:MUX:CONDition?

Response	<numeric> = <NR1 NUMERIC RESPONSE DATA>	
	0 to 60	Sum total of the condition bit (DECIMAL)
	Available bit	
	4 (Bit2)	Delay Busy Occurrence (Data1)
	8 (Bit3)	Delay Busy Occurrence (Data2)
	16 (Bit4)	Delay Calibration Require Occurrence (Data1)
	32 (Bit5)	Delay Calibration Require Occurrence (Data2)
Function	Queries condition at 25Gbit/s MUX status.	
Example	`> :INSTrument:MUX:CONDition?`	
	`< 4`	

: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)	Delay Busy Occurrence (Data1)
	8 (Bit3)	Delay Busy Occurrence (Data2)
	16 (Bit4)	Delay Calibration Require Occurrence (Data1)
	32 (Bit5)	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)	Delay Busy Occurrence (Data1)
	8 (Bit3)	Delay Busy Occurrence (Data2)
	16 (Bit4)	Delay Calibration Require Occurrence (Data1)
	32 (Bit5)	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

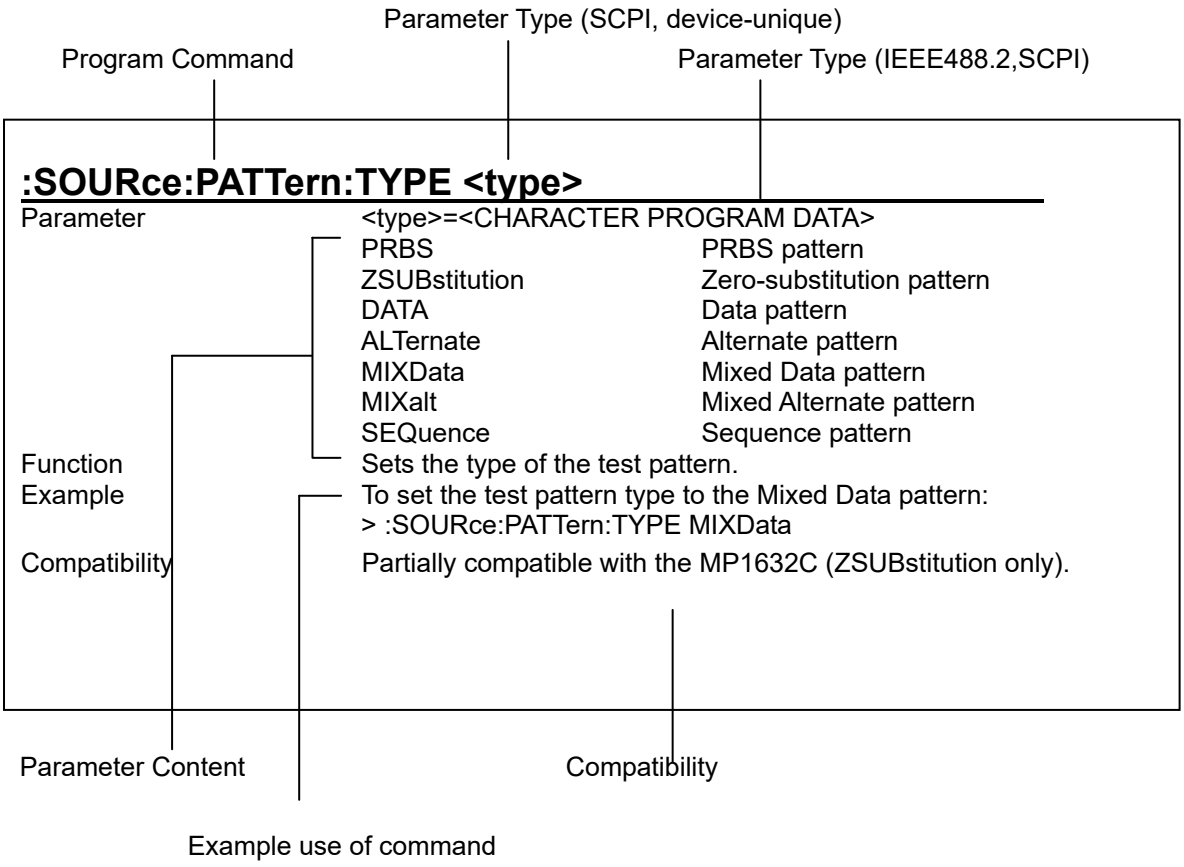
Function	Initializes event at 25Gbit/s MUX status.
Example	`> :INSTrument:MUX:RESet`

Chapter 7 SCPI Commands

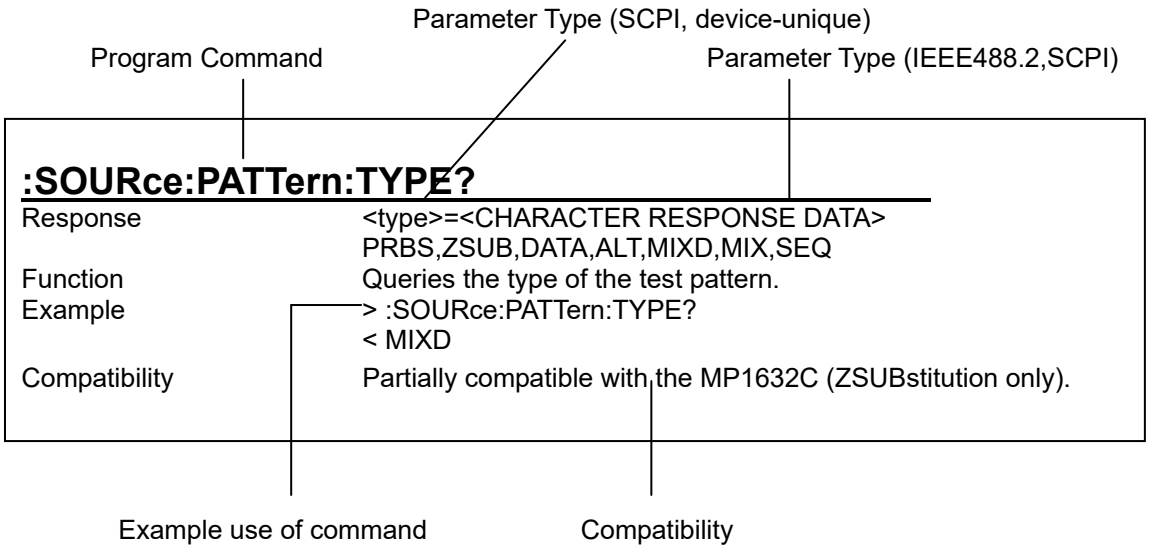
This chapter describes the SCPI commands.

Examples of command expression are shown below.

<Program Command>



<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

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

: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.

: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: `> :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	`> :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: `> :SYSTem:CFUNction ASE`
Compatibility	Incompatible with existing models.

:SYSTem:CFUNction?

Response	<function> = <CHARACTER RESPONSE 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
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.
	Refer to Chapters 8 and 9 for details on the error messages.
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.	

: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 7.1.1-2 "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>

XXXXXXXXX	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>

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

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 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

Model/Name	Option Number	Option Name
MP1800A Signal Quality Analyzer	OPT014	2-Slot for PPG and/or ED
	OPT015	4-Slot for PPG and/or ED
	OPT016	6-Slot for PPG and/or ED
MT1810A 4 Slot Chassis	OPT014	2-Slot for PPG and/or ED
	OPT015	4-Slot for PPG and/or ED
MX180000A Signal Quality Analyzer Control Software	–	–
MU181000A 12.5GHz Synthesizer	OPTx01	Jitter Modulation
MU181000B 12.5GHz 4port Synthesizer	OPTx01	Jitter Modulation
MU181020A 12.5Gbit/s PPG	OPT001	9.8 to 12.5 Gbit/s
	OPT002	0.1 to 12.5 Gbit/s
	OPTx10	Variable Data Output (0.05 to 0.8 Vp-p)
	OPTx11	Variable Data Output (0.25 to 2.5 Vp-p)
	OPTx12	High Performance Data Output (0.05 to 2.0 Vp-p)
	OPTx13	Variable Data Output (0.5 to 3.5 Vp-p)
	OPTx21	Differential Clock Output (0.1 to 2.0 Vp-p)
	OPTx30	Variable Data Delay
MU181040A 12.5Gbit/s ED	OPT001	9.8 to 12.5 Gbit/s
	OPT002	0.1 to 12.5 Gbit/s
	OPTx20	Clock Recovery
	OPTx30	Variable Clock Delay
MU181600A Optical Transceiver (XFP)	–	–
MU181601A Optical Transceiver (SFP)	–	–
MU181620A Stressed Eye Transmitter	OPTx01	1310nm Reference
	OPTx02	1550nm Reference
	OPTx03	1310nm/1550nm Reference
	OPTx11	1310nm Stressed Eye
	OPTx12	1550nm Stressed Eye
	OPTx13	1310nm/1550nm Stressed Eye
MU181640A Optical Receiver	OPTx04	Band Width 8.5GHz
MU181800A 12.5GHz Clock Distributor	–	–
MU181020B 14Gbit/s PPG	OPT002	0.1 to 14 Gbit/s
	OPTx11	Variable Data Output (0.25 to 2.5 Vp-p)
	OPTx12	High Performance Data Output (0.05 to 2.0 Vp-p)
	OPTx13	Variable Data Output (0.5 to 3.5 Vp-p)
	OPTx21	Differential Clock Output (0.1 to 2.0 Vp-p)
	OPTx30	Variable Data Delay

Table 7.1.1-2 Option character correspondence table (Cont'd)

Model/Name	Option Number	Option Name
MU181040B 14Gbit/s ED	OPT002	0.1 to 14 Gbit/s
	OPTx20	Clock Recovery
	OPTx30	Variable Clock Delay
MU182020A 25Gbit/s 1ch MUX MU182021A 25Gbit/s 2ch MUX	OPTx01	28Gbit/s Extension
	OPTx02	Clock Input Band Switch
	OPTx10	Variable Data Output (0.25 to 1.75Vp-p)
	OPTx11	Variable Data Output (0.5 to 2.5Vp-p)
	OPTx12	Variable Data Output (0.5 to 2.0Vp-p)
	OPTx13	Variable Data Output (0.5 to 3.5Vp-p)
	OPTx21	Differential Clock Output (0.5 to 2.0Vp-p)
	OPTx30	25Gbit/s Variable Data Delay
	OPTx31	28Gbit/s Variable Data Delay
MU182040A 25Gbit/s 1ch DEMUX MU182041A 25Gbit/s 2ch DEMUX	OPTx01	28Gbit/s Extension
	OPTx02	Clock Input Band Switch
	OPTx30	25GHz Variable Clock Delay
	OPTx31	28GHz Variable Clock Delay

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>

XXXXXXXXX	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>

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:

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

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,OPT2
20
```

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	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.	

:SYSTem:CONDition:UNITs?

Response

<unit1>,...,<unit4>,<slot1>,...,<slot6>
="<mainframe1>,...,<mainframe4>,<module1>,...,<module64>"
<mainframe1> to <mainframe4> = <STRING RESPONSE DATA>

XXXXXXXXX	Mainframe name (e.g.,: MP1800A) SeeTable 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>

XXXXXXXXX	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.

: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>

XXXXXXXXX Mainframe name (e.g., : 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.

<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 7.1.1-2 "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,OPT301,1.00.00,1.00.0
0
```

Compatibility: Incompatible with existing models.

:SYSTem:MODule? <numeric>

Parameter

<numeric> = <NR1 NUMERIC PROGRAM DATA>

1 to 6 Slot

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>

XXXXXXXXX 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>

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

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,O
PT002,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.

: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
```
> :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)
```
> :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

Setting Items	Commands
Quick Open	:SYSTem:MMEMory:QRECall
Quick Save	:SYSTem:MMEMory:QSTore
Open	:SYSTem:MMEMory:RECall
Save	:SYSTem:MMEMory:STORe
Screen Copy Execute	:SYSTem:PRINt:COPY
Screen Copy Setup	:SYSTem:PRINt:COPYsetup:SET
	:SYSTem:PRINt:COPYsetup:SET?
Print	:SYSTem:PRINt:EALarm
Printer Setup	:SYSTem:PRINt:PRINter:SET
	:SYSTem:PRINt:PRINter:SET?
Combination Setting	:SYSTem:CONDition:COMBination?
	:COMBination:OPERation:SETTing
	:COMBination:OPERation:SETTing?
	:COMBination:OPERation:ABILity:CHSYnc?
	:COMBination:OPERation:CHSetting
	:COMBination:OPERation:ABILity:MUXChsync?
	:COMBination:OPERation:MUXChsync
	:COMBination:OPERation:MUXChsync?
	:COMBination:OPERation:ABILity:COMBination?
	:COMBination:OPERation:USYNcout
	:COMBination:OPERation:USYNcout?
Initialize	:SYSTem:MEMory:INITialize

: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>
"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>

PPG	PPG Setup
ED	ED Setup
SYN	Synthesizer Setup
XFP	XFP Setup
SFP	SFP Setup
CAP	Capture Result
OTX	Transmitter Setup
MX1	MUX Setup
MX2	MUX Setup
DX1	DEMUX Setup
DX2	DEMUX Setup

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.

: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>

PPG	PPG Setup
ED	ED Setup
SYN	Synthesizer Setup
XFP	XFP Setup
SFP	SFP Setup
ERA	Error/Alarm Result
HST	Histogram Result
CAP	Capture Result
LOG	Logging Result
OTX	Transmitter Setup
MX1	MUX Setup
MX2	MUX Setup
DX1	DEMUX Setup
DX2	DEMUX Setup
DER	DEMUX Error/Alarm Result

<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
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.

: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.

:SYSTem:PRINt:EALarm <list>,<module>

Response
<list> = <CHARACTER PROGRAM DATA>
EAR Error/Alarm Result
HST Histogram Result
LOG Logging Result
<module> = <STRING PROGRAM DATA>
"<unit>:<slot>:<port>:<module>"
<unit> = 1, 2, 3, 4
<slot> = 1, 2, 3, 4, …, 6
<port> = 1, 2, …
<module> = Module model name

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:
```
> :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>
	"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: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.

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

Parameter

<configuration> = <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

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

Example

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

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

Compatibility

Incompatible with existing models.

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

Parameter

<slot> = <CHARACTER PROGRAM DATA>

SLOT1 to SLOT6	Slot No.1 to 6
ALL	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 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
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:

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

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

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

To query the combination setting of Unit 3:

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

Compatibility

Incompatible with existing models.

: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:

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:

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

Compatibility

Incompatible with existing models.

: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
	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:	
	`> :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.

: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.

:COMBination:OPERation:USYNcout?

Parameter	<numeric> = <NR1 NUMERIC RESPONSE DATA>
	0 Unit Sync OFF
	1 Unit Sync ON
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

Figure 7.1.3-1 Common function button

Table 7.1.3-1 Common function button setting commands

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

: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.

: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 optical output and PPG.
Example	`> :SOURce:OUTPut:ASET?`
	`< 1`
Compatibility	Incompatible with existing models.

: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.

: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 the MP1632C and MP1776A.

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

Compatibility	Compatible with the MP1632C and MP1776A.

: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.

: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.

7.1.4 Auto Search

This chapter describes the commands that are used after Auto Search is set by the :SYSTem:CFUNction command.

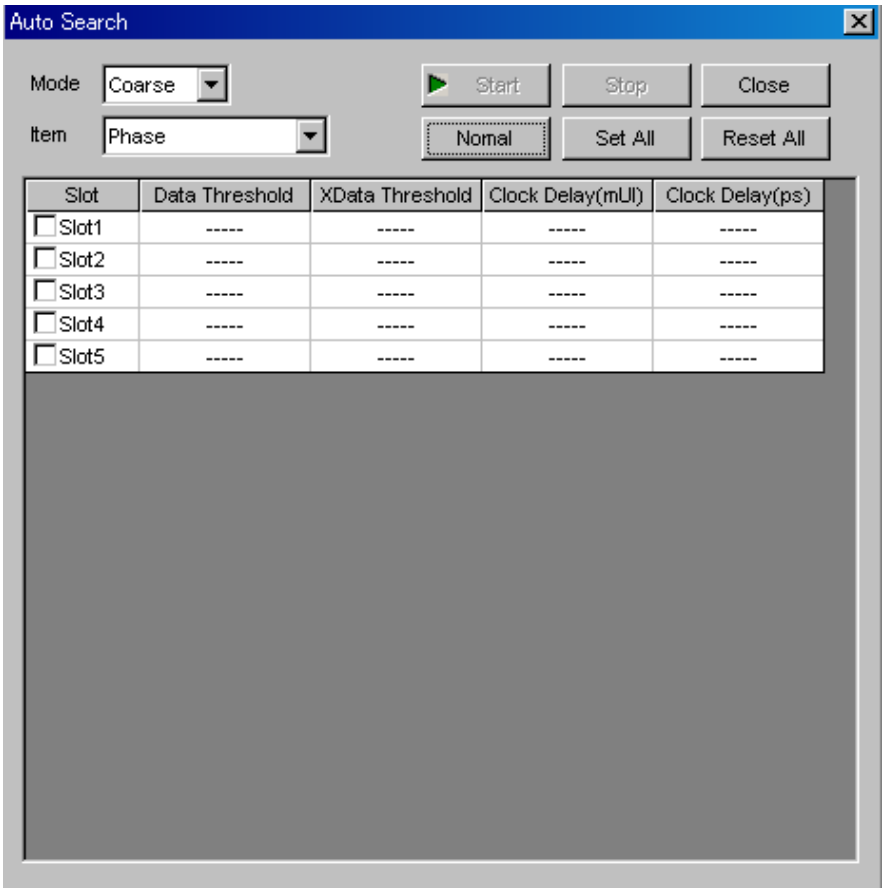


Figure 7.1.4-1 Auto Search setting screen

Table 7.1.4-1 Auto Search setting commands

Setting Items	Commands
Mode	:SENSe:MEASure:ASEarch:SMODe
	:SENSe:MEASure:ASEarch:SMODe?
Item	:SENSe:MEASure:ASEarch:MODE
	:SENSe:MEASure:ASEarch:MODE?
Start	:SENSe:MEASure:ASEarch:STARt
Stop	:SENSe:MEASure:ASEarch:STOP
No label (Query for function state)	:SENSe:MEASure:ASEarch:STATe?
Set All	:SENSe:MEASure:ASEarch:SLASet
Reset All	:SENSe:MEASure:ASEarch:SLAReset
Slot Select	:SENSe:MEASure:ASEarch:SELSlot
	:SENSe:MEASure:ASEarch:SELSlot?
No label (Query for slot state)	:SENSe:MEASure:ASEarch:SLOT?
No label (Query for Auto Search result)	:CALCulate:DATA:ASEarch?

: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	`> :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	<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	`> :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.

: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>,<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.

: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
```
> :SENSe:MEASure:ASEarch:SLOT?
< SLOT1,SLOT2,SLOT6
```

Compatibility
Incompatible with existing models.

:CALCulate:DATA:ASEarch? <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

<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 (converted from mUI)
"----"	Not executed
"----"	Abnormal termination

Function

Queries the Auto Search execution result.

Example

```
> :CALCulate:DATA:ASEarch? SLOT1
< "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.

7.1.5 Auto Adjust

This chapter describes the commands that are used after Auto Adjust is set by the :SYSTem:CFUNction command.

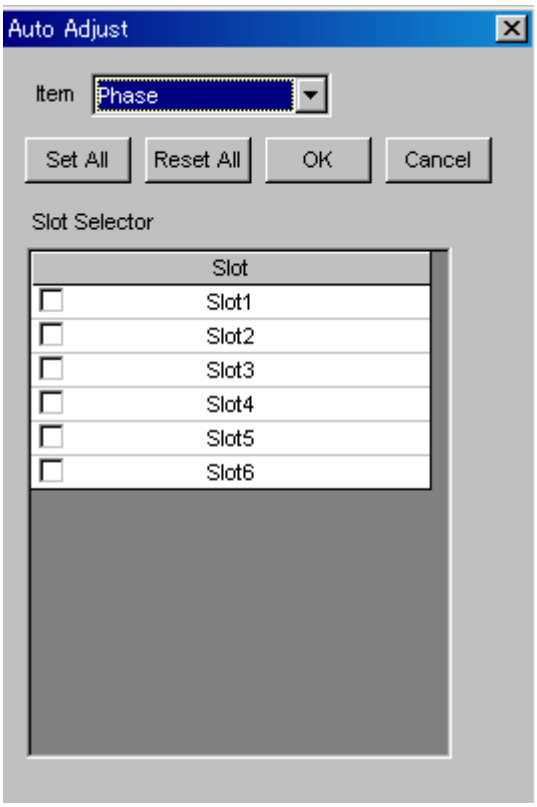


Figure 7.1.5-1 Auto Adjust setting screen

Table 7.1.5-1 Auto Adjust setting commands

Setting Items	Commands
Item	:SENSe:MEASure:AADJust:ITEM
	:SENSe:MEASure:AADJust:ITEM?
Start	:SENSe:MEASure:AADJust:STARt
Stop	:SENSe:MEASure:AADJust:STOP
No label (Query for function state)	:SENSe:MEASure:AADJust:STATe?
Slot Select	:SENSe:MEASure:AADJust:SELSlot
	:SENSe:MEASure:AADJust:SELSlot?
No label (Query for slot state)	:SENSe:MEASure:AADJust:SLOT?
Set All	:SENSe:MEASure:AADJust:SLASet
Reset All	:SENSe:MEASure:AADJust:SLAReset

:SENSe:MEASure:AADJust:ITEM <item>

Parameter	<item> = <CHARACTER PROGRAM DATA>		
	PTHReshold	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	To set the execution item of Auto Adjust to Phase & Threshold:		
	`> :SENSe:MEASure:AADJust:ITEM PTHReshold`		
Compatibility	Incompatible with existing models.		

:SENSe:MEASure:AADJust:ITEM?

Response	<item> = <CHARACTER RESPONSE DATA>	
	PTHR	Phase & Threshold
	THR	Threshold
	PHAS	Phase
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.

:SENSe:MEASure:AADJust:STATe?

Response	<boolean> = <NR1 NUMERIC RESPONSE DATA>
	1 Start
	0 Stop
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>]

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 Adjust ON
	OFF, 0 Auto Adjust 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 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>]

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 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`

```
< 1
```

Compatibility: 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.

7.1.6 ED ALL

Table 7.1.6-1 ED ALL setting commands

Setting Items	Commands
Measurement result query in the case of Combination	:CALCulate:ADATa:EALarm?
Logging ON/OFF	:DISPlay:ARESult:LOGGing:ONOFf
	:DISPlay:ARESult:LOGGing:ONOFf?
Clear	:DISPlay:ARESult:LOGGing:CLEar
Condition	:DISPlay:ARESult:LOGGing:CONDition:ITEM
	:DISPlay:ARESult:LOGGing:CONDition:ITEM?
Set All/Reset All	:DISPlay:ARESult:LOGGing:CONDition:ALLSet
1 Second data	:DISPlay:ARESult:LOGGing:CONDition:PERiod
	:DISPlay:ARESult:LOGGing:CONDition:PERiod?
Error Threshold	:DISPlay:ARESult:LOGGing:CONDition:THReshold
	:DISPlay:ARESult:LOGGing:CONDition:THReshold?
Squelch	:DISPlay:ARESult:LOGGing:CONDition:SQUelch
	:DISPlay:ARESult:LOGGing:CONDition:SQUelch?
Save	:SYSTem:MMEMory:EDAL:STORe
Print	:SYSTem:PRINt:EDAL

: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

Type	Format	Description
Form1 Integer type	"XXXXXXX"	For 0 to 9999999
	"X.XXXXEXX"	For 1.0000E07 to 9.9999E16
	"-------"	When no data corresponds to the query
Form2 Fraction type	"X.XXXXE-XX"	For 0.0000E16 to 1.0000E–00
	"-------"	When no data corresponds to the query
Form3 % type	"XXX.XXXXX"	For 0.0000 to 100.0000
	"-------"	When no data corresponds to the query
Form4 Frequency type	"XXXXXXXXX"	For 0 to MAX (Hz)
	"---------"	When no data corresponds to the query
Form5 Voltage type	"X.XXX"	For –4.000 to +4.000
	"----"	When no data corresponds to the query

Function

Queries the measurement data at Combination that corresponds to the parameter.

Table 7.1.6-3 Parameters

Item		<result>	Format
Error Rate	Total	ER:TOTal	Form2
	INS	ER:INSertion	
	OMI	ER:OMISsion	
Error Count	Total	EC:TOTal	Form1
	INS	EC:INSertion	
	OMI	EC:OMISsion	
Error Rate	Transition	ER:TRANsition	Form2
	Non Transition	ER:NONTransition	
Error Count	Transition	EC:TRANsition	Form1
	Non Transition	EC:NONTransition	
%EFI	Total	EFI:TOTal	Form3
EI	Total	EI:TOTal	Form1
Frequency (kHz)		FREQuency	Form4
Clock Count		CC:TOTal	Form1
Clock Loss		AINTerval:CLOSs	Form1
Sync Loss		AINTerval:PSLoss	Form1
Data Threshold		THReshold	Form5
Data Phase		PHASe	Form1
Threshold EI	>1.0E–3	THReshold:EI:TOTal:E_3	Form1
	>1.0E–4	THReshold:EI:TOTal:E_4	
	>1.0E–5	THReshold:EI:TOTal:E_5	
	>1.0E–6	THReshold:EI:TOTal:E_6	
	>1.0E–7	THReshold:EI:TOTal:E_7	
	>1.0E–8	THReshold:EI:TOTal:E_8	
	≤1.0E–8	THReshold:EI:TOTal:UE_8	

Table 7.1.6-3 Parameters (Cont'd)

Item		<result>	Format
Threshold %EFI	>1.0E–3	THReshold:EFI:TOTal:E_3	Form3
	>1.0E–4	THReshold:EFI:TOTal:E_4	
	>1.0E–5	THReshold:EFI:TOTal:E_5	
	>1.0E–6	THReshold:EFI:TOTal:E_6	
	>1.0E–7	THReshold:EFI:TOTal:E_7	
	>1.0E–8	THReshold:EFI:TOTal:E_8	
	≤1.0E–8	THReshold:EFI:TOTal:UE_8	
G.821	ES	G821:ES2:TOTal	Form1
	EFS	G821:EFS2:TOTal	
	SES	G821:SES2:TOTal	
	DM	G821:DM2:TOTal	
	US	G821:US2:TOTal	
	%ES	G821:ES:TOTal	Form3
	%EFS	G821:EFS:TOTal	
	%SES	G821:SES:TOTal	
	%DM	G821:DM:TOTal	
	%US	G821:US:TOTal	

Example To query the total error rate measurement data for the current data:

```
> :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:

```
> :DISPlay:ARESult:LOGGing:ONOFf ON
```

Compatibility Incompatible with existing models.

:DISPlay:ARESult:LOGGing:ONOFf?

Response	<numeric> = <NR1 NUMERIC RESPONSE DATA>
	0 OFF
	1 ON
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>
	SLOTinfo Slot information
	TPATtern Test pattern type
	STARt Measurement start time
	END Measurement end time
	ERATe Error rate and error count
	EI Error interval (EI) and error free interval rate (%EFI)
	FREQuency Frequency
	CCOunt Clock count
	AOCCur Alarm occurrence (Alarm Occur) and recovery time (Recover)
	AINTerval 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. Disabled when Variable Clock Delay (MU181040A-x30) is not installed.
	SDATa Average data for one second
	<boolean> = <BOOLEAN PROGRAM DATA>
	0 or OFF Measurement item OFF
	1 or ON Measurement item ON
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.

:DISPlay:ARESult:LOGGing:CONDition:ITEM? <item>

Parameter	<item> = <CHARACTER PROGRAM DATA> SLOTinfo, TPATtern, STARt, END, ERATe, EI, FREQuency, CCOunt, AOCCur, AINTerval, DTHReshold, CDELey, SDATa
Response	<numeric> = <NR1 NUMERIC RESPONSE DATA> 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	<boolean> = <BOOLEAN PROGRAM DATA> 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	<period> = <CHARACTER PROGRAM DATA> 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.

:DISPlay:ARESult:LOGGing:CONDition:PERiod?

Response	<period> = <CHARACTER RESPONSE DATA>	
	S10	10 seconds
	S30	30 seconds
	M1	1 minute
	M10	10 minutes
	M30	30 minutes
	H1	1 hour
Function	Queries the logging result capturing period for all ED modules.	
Example	`> :DISPlay:ARESult:LOGGing:CONDition:PERiod?`	
	`< S10`	
Compatibility	Incompatible with existing models.	

:DISPlay:ARESult:LOGGing:CONDition:THReshold <thre>

Parameter	<thre> = <CHARACTER PROGRAM DATA>	
	E_0	<>0
	E_3	>1E–3
	E_4	>1E–4
	E_5	>1E–5
	E_6	>1E–6
	E_7	>1E–7
	E_8	>1E–8
	E_9	>1E–9
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:	
	`> :DISPlay:ARESult:LOGGing:CONDition:THReshold E_0`	
Compatibility	Incompatible with existing models.	

:DISPlay:ARESult:LOGGing:CONDition:THReshold?

Response	<thre> = <CHARACTER RESPONSE DATA>
Function	Queries the logging error rate threshold value for all ED modules.
Example	`> :DISPlay:ARESult:LOGGing:CONDition:THReshold?`
	`< E_0`
Compatibility	Incompatible with existing models.

:DISPlay:ARESult:LOGGing:CONDition:SQUelch <boolean>

Parameter
<boolean> = <BOOLEAN PROGRAM DATA>
0 or OFF — Memory squelch function ON
1 or ON — Memory squelch function OFF

Function
Sets the logging memory squelch function for all ED modules ON/OFF.

Example
To set the logging memory squelch function for all ED modules ON:
```
> :DISPlay:ARESult:LOGGing:CONDition:SQUelch ON
```

Compatibility
Incompatible with existing models.

:DISPlay:ARESult:LOGGing:CONDition:SQUelch?

Response
<numeric> = <NR1 NUMERIC RESPONSE DATA>
0 — Memory squelch function OFF
1 — Memory squelch function ON

Function
Queries the logging memory squelch function ON/OFF state for all ED modules.

Example
```
> :DISPlay:ARESult:LOGGing:CONDition:SQUelch?
< 1
```

Compatibility
Incompatible with existing models.

:SYSTem:MMEMory:EDAL: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 for the root directory)
<file> = File name
<data_type> = <CHARACTER PROGRAM DATA>
AERA — Error/Alarm Result ALL
ALOG — Logging Result ALL
<file_type> = <CHARACTER PROGRAM DATA>
CSV — CSV File
TXT — Text File
Only TXT can be specified for <file_type> when ALOG is set for <data_type>.

Function
Saves the ED Result All measurement result data.

Example
To save the ED Result All measurement result data to the specified destination in the specified file format:
```
> :SYSTem:MMEMory:EDAL:STORe "C:\Test\example",AERA,CSV
```

Compatibility
Incompatible with existing models.

:SYSTem:PRINt:EDAL <list>

Parameter	<list> = <CHARACTER PROGRAM DATA> AERA Error/Alarm Result All ALOG Logging Result All
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.

7.1.7 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

Setting Items	Commands
Open	:SYSTem:MMEMory:PATTern:RECall
Save	:SYSTem:MMEMory:PATTern:STORe

: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 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:PATTern:RECall "C:\Test\example",BIN
```

Compatibility

Incompatible with existing models.

:SYSTem:MMEMory:PATTern:STORe <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

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:MMEMory:PATTern: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.

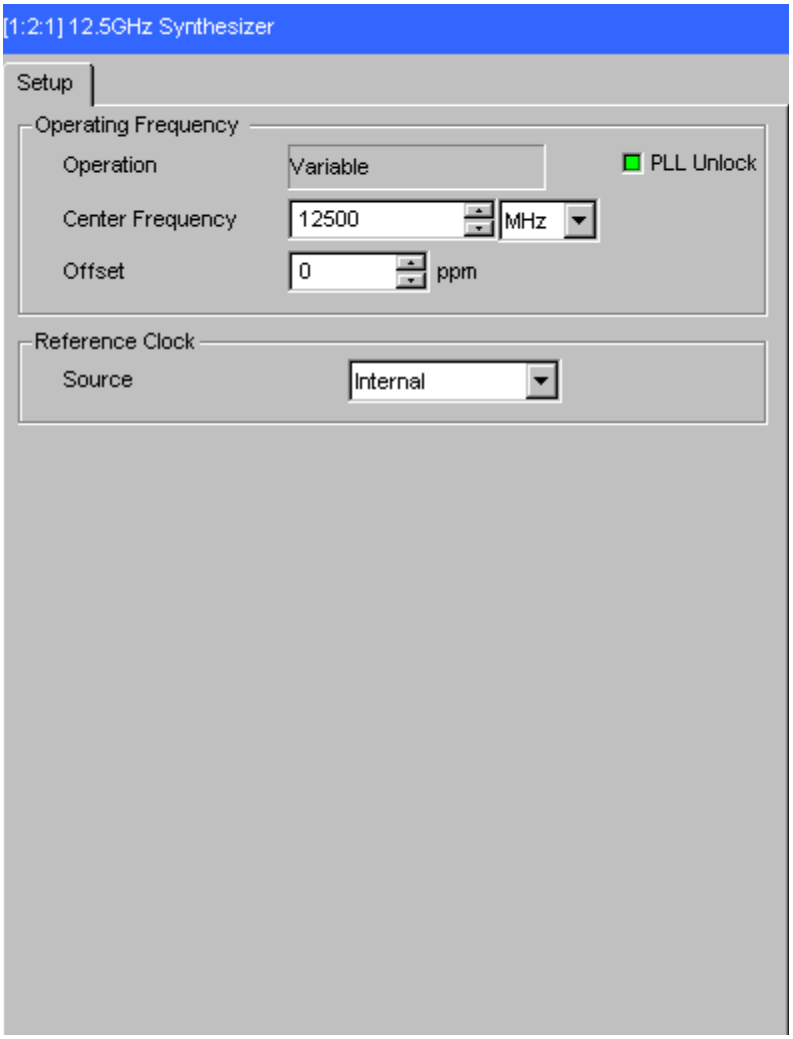


Figure 7.2-1 12.5GHz Synthesizer Window

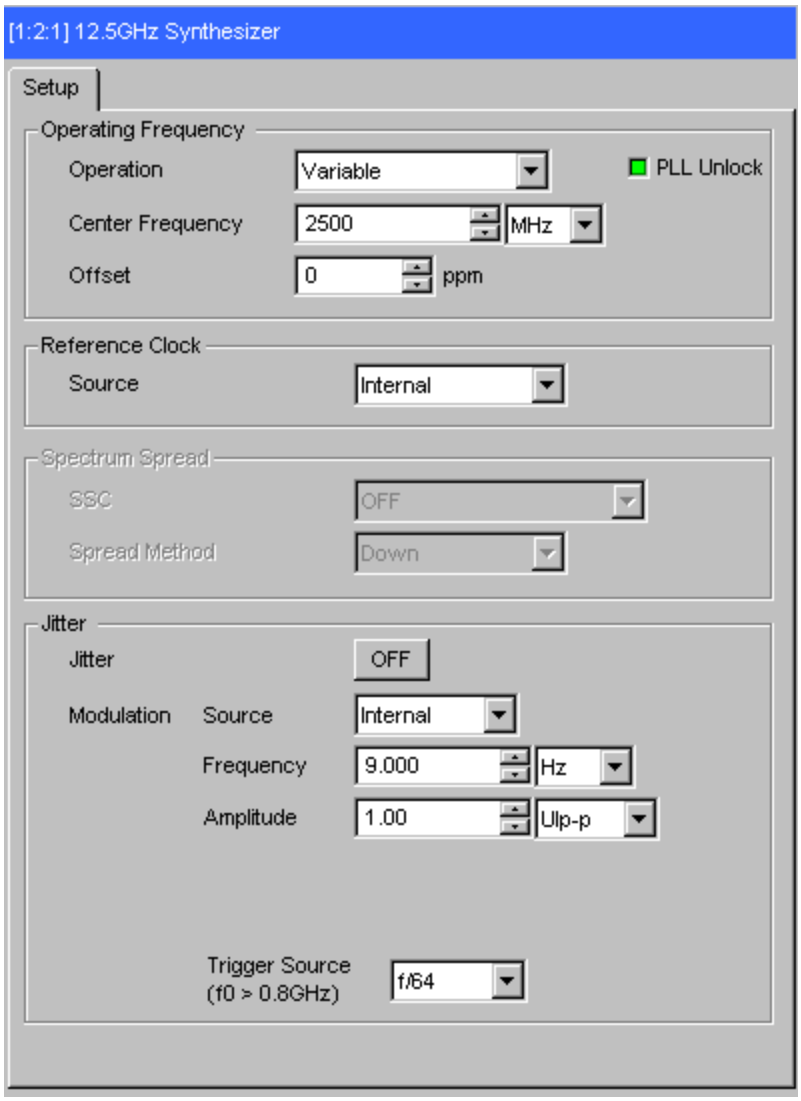


Figure 7.2-2 12.5 GHz Synthesizer Window (Option x01)

7.2.1 Operation frequency Setting field

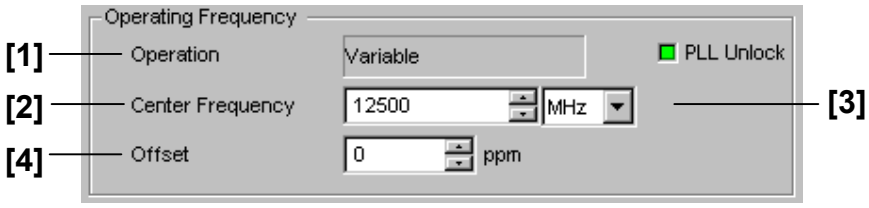


Figure 7.2.1-1 Operation Frequency Setting

Table 7.2.1-1 Operation Frequency setting Commands

No.	Setting Items	Commands
[1]	Operation	:OUTPut:CLOCk:OPERation
		:OUTPut:CLOCk:OPERation?
[2]	Center Frequency	:OUTPut:CLOCk:FREQuency
		:OUTPut:CLOCk:FREQuency?
[3]	Center Frequency Unit	:OUTPut:CLOCk:FUNit
		:OUTPUt:CLOCk:FUNit?
[4]	Offset	:OUTPut:CLOCk:OFFSet:PPM
		:OUTPut:CLOCk:OFFSet:PPM?

:OUTPut:CLOCk:OPERation <operation>

Parameter	<operation> = <CHARACTER PROGRAM DATA>	
	VARiable	Variable (Variable frequency 0.1 to 12.5 GHz)
	PCIexpress1	PCIe-Gen I (2.5 GHz)
	PCIexpress2	PCIe-Gen II (5 GHz)
	PCIexpress345	PCIe-Gen III/IV/V (8 GHz)
	SERialata	Serial-ATA (6 GHz)
Function	Sets Operating of Synthesizer.	
Example	Sets Operating of Synthesizer to Variable.	
	`> :OUTPut:CLOCk:OPERation VARiable`	
Compatibility	Incompatible with existing models.	

:OUTPut:CLOCk:OPERation?

Response	<operation> = <CHARACTER RESPONSE DATA> VAR, PCI1, PCI2, PCI345, SER
Function	Queries Operating of Synthesizer.
Example	`> :OUTPut:CLOCk:OPERation?` `<PCI2`
Compatibility	Incompatible with existing models.

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>

Parameter	<numeric> = <DECIMAL NUMERIC PROGRAM DATA> In case of Operating frequency resolution unit is kHz 100000 to 12500000 in 1 kHz step In case of Operating frequency resolution unit is MHz 100 to 12500/Step 1 MHz
Function	Sets operating frequency of Synthesizer.
Example	Sets operating frequency of Synthesizer to 12500 MHz. `> :OUTPut:CLOCk:FREQuency 12500`
Compatibility	Compatible with MP1632C.

:OUTPut:CLOCk:FREQuency?

Response	<numeric> = <NR1 NUMERIC RESPONSE DATA> In case of Operating frequency resolution unit is kHz 100000 to 12500000 in 1 kHz step In case of Operating frequency resolution unit is MHz 100 to 12500/Step 1 MHz
Function	Queries operating frequency of Synthesizer.
Example	`> :OUTPut:CLOCk:FREQuency?` `< 12500`
Compatibility	Compatible with MP1632C.

:OUTPut:CLOCk:FUNit <resolution>

Parameter	<resolution> = <CHARACTER PROGRAM DATA>
	KHZ kHz
	MHZ MHz
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?

Response	<resolution> = <CHARACTER RESPONSE DATA>
	KHZ, MHZ
Function	Queries operating frequency resolution of Synthesizer.
Example	`> :OUTPut:CLOCk:FUNit?`
	`< MHZ`
Compatibility	Incompatible with existing models.

:OUTPut:CLOCk:OFFSet:PPM <numeric>

Parameter	<numeric> = <DECIMAL NUMERIC PROGRAM DATA>
	–1000 to 1000 –1,000 to 1,000 in 1 ppm step
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?

Response	<numeric> = <NR1 NUMERIC RESPONSE DATA>
	–1000 to 1000 –1,000 to 1,000 in 1 ppm step
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

Figure 7.2.2-1 Reference Clock Setting

Table 7.2.2-1 Reference Clock Setting Commands

No.	Setting Items	Commands
[1]	Source	:OUTPut:RCLock:SELect
		:OUTPut:RCLock:SELect?

:OUTPut:RCLock:SELect <clock>

Parameter	<clock> = <CHARACTER PROGRAM DATA>
	INTernal Internal 10 MHz reference signal
	EXTernal10 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	<clock> = <CHARACTER RESPONSE DATA>
	INT, EXT10
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.

Figure 7.2.3-1 Spectrum Spread setting

Table 7.2.3-1 Spectrum Spread setting commands

No.	Setting Items	Commands
[1]	SSC	:OUTPut:RCLock:SSC:SET
		:OUTPut:RCLock:SSC:SET?
[2]	Spread Method	:OUTPut:RCLock:SSC:METHod
		:OUTPut:RCLock:SSC:METHod?

:OUTPut:RCLock:SSC:SET <SSC>

Parameter

<SSC> = <CHARACTER PROGRAM DATA>

OFF	SSC function off
INTernal	SSC function on (using internal reference signal)
EXTernal10	SSC function on (using external 10 MHz reference signal)
EXTernal100	SSC function on (using external 100 MHz reference signal)

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.

: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.

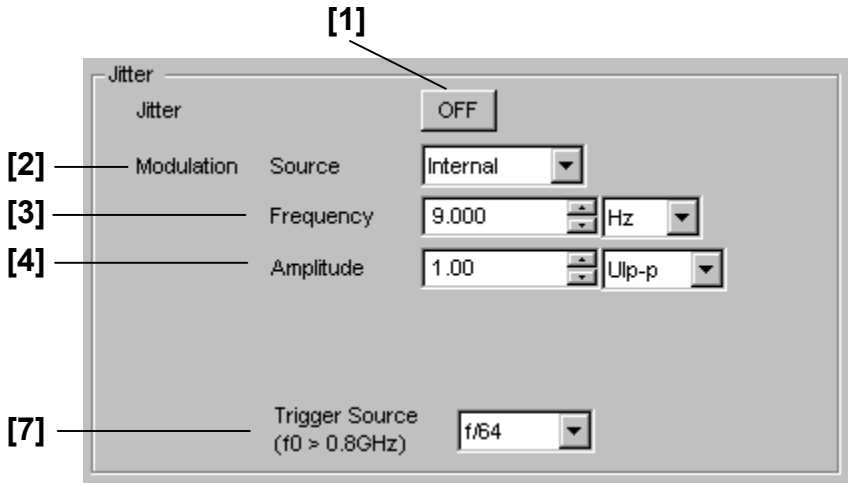


Figure 7.2.4-1 Jitter setting (Modulation Source: Internal)

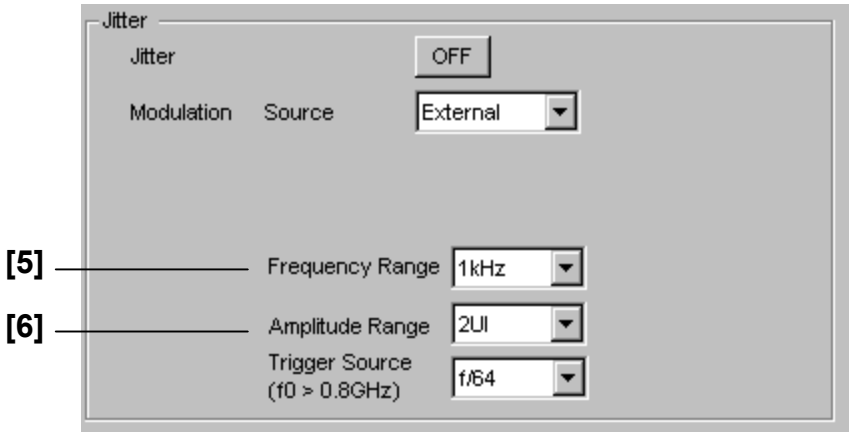


Figure 7.2.4-2 Jitter setting (Modulation Source: External)

Table 7.2.4-1 Jitter setting commands

No.	Setting Items	Commands
[1]	Jitter ON/OFF	:SOURce:SYNThesizer:JITTer:OUTPut:SET
		:SOURce:SYNThesizer:JITTer:OUTPut:SET?
[2]	Modulation Source	:SOURce:SYNThesizer:JITTer:MODulation:SELect
		:SOURce:SYNThesizer:JITTer:MODulation:SELect?
[3]	Frequency	:SOURce:SYNThesizer:JITTer:MODulation:FREQuency
		:SOURce:SYNThesizer:JITTer:MODulation:FREQuency?
[4]	Amplitude	:SOURce:SYNThesizer:JITTer:MODulation:AMPLitude
		:SOURce:SYNThesizer:JITTer:MODulation:AMPLitude?
[5]	Frequency Range	:SOURce:SYNThesizer:JITTer:MODulation:FREQuency:RANGe
		:SOURce:SYNThesizer:JITTer:MODulation:FREQuency:RANGe?
[6]	Amplitude Range	:SOURce:SYNThesizer:JITTer:MODulation:AMPLitude:RANGe
		:SOURce:SYNThesizer:JITTer:MODulation:AMPLitude:RANGe?
[7]	Trigger Source	:SOURce:SYNThesizer:JITTer:TRGSource
		:SOURce:SYNThesizer:JITTer:TRGSource?

: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:SELect <clock>

Parameter	<clock> = <CHARACTER PROGRAM DATA>	
	INTernal	Internal modulation source
	EXTernal	External modulation source
	IQEXternal	External modulation source (IQ)
Function	Sets modulation signal source.	
Example	Sets modulation signal source to internal modulation source.	
	`> :SOURce:SYNThesizer:JITTer:MODulation:SELect INTernal`	
Compatibility	Incompatible with existing models.	

:SOURce:SYNThesizer:JITTer:MODulation:SELect?

Response	<clock> = <CHARACTER RESPONSE DATA>
	INT, EXT, IQEX
Function	Queries modulation signal source.
Example	`> :SOURce:SYNThesizer:JITTer:MODulation:SELect?`
	`< INT`
Compatibility	Incompatible with existing models.

:SOURce:SYNThesizer:JITTer:MODulation:FREQuency <numeric>,<resolution>

Parameter

<numeric> = <DECIMAL NUMERIC PROGRAM DATA>

When Hz is selected for <resolution>

9.000 to 1000.0	9.000 to 1000.0 Hz Setting steps 9.000[Hz] ≤ fmod ≤10.000 [Hz]: In 0.001-Hz steps 10.01[Hz] ≤ fmod ≤100.00 [Hz]: In 0.01-Hz steps 100.1[Hz] ≤ fmod ≤1000.0 [Hz]: In 0.1-Hz steps

When kHz is selected for <resolution>

1.001 to 1000.0	1.00 to 1000.0 kHz Setting steps 1.001[kHz] ≤ fmod ≤10.000 [kHz]: In 0.001-kHz steps 10.01[kHz] ≤ fmod ≤100.00 [kHz]: In 0.01-kHz steps 100.1[kHz] ≤ fmod ≤1000.0 [kHz]: In 0.1-kHz steps

When MHz is selected for <resolution>

1.001 to 80.00	1.00 to 80 MHz Setting steps 1.001[MHz] ≤ fmod ≤10.000 [MHz]: In 0.001-MHz step 10.01[MHz] ≤ fmod ≤80.00 [MHz]: In 0.01-MHz steps

<resolution> = <CHARACTER PROGRAM DATA>

HZ	Hz units
KHZ	kHz units
MHZ	MHz units

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.

:SOURce: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.	

:SOURce:SYNThesizer:JITTer:MODulation:AMPLitude <numeric>,<resolution>

Parameter	<numeric> = <DECIMAL NUMERIC PROGRAM DATA>	
	0 to 999	0 to 999 mUIp-p, in 1-mUp-p steps
	1.00 to 32.00	1.00 to 32.00 UIp-p, in 0.01-UIp-p steps
	32.1 to 256.0	32.1 to 256.0 UIp-p, in 0.1-UIp-p steps
	257 to 2049	257 to 2049 UIp-p, in 1-UIp-p steps
	2050 to 4000	2050 to 4000 UIp-p, in 10-UIp-p steps
	<resolution> = <CHARACTER PROGRAM DATA>	
	MUI	mUIp-p units
	UI	UIp-p units
Function	Sets internal modulation amplitude and units.	
Example	Sets internal modulation amplitude to 1.10 UIp-p.	
	`> :SOURce:SYNThesizer:JITTer:MODulation:AMPLitude 1.10,UI`	
Compatibility	Incompatible with existing models.	

:SOURce:SYNThesizer:JITTer:MODulation:AMPLitude?

Response	<numeric> = <NR2 NUMERIC RESPONSE DATA>	
	0 to 999	0 to 999 mUIp-p
	1.00 to 32.00	1.00 to 32.00 UIp-p
	32.1 to 256.0	32.1 to 256.0 UIp-p
	257 to 2049	257 to 2049 UIp-p
	2050 to 4000	2050 to 4000 UIp-p
	<resolution> = <CHARACTER RESPONSE DATA>	
	MUI	mUIp-p units
	UI	UIp-p units
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

<range> = <CHARACTER PROGRAM DATA>

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

```
> :SOURce:SYNThesizer:JITTer:MODulation:FREQuency:RANGe?
< K_100
```

Compatibility

Incompatible with existing models.

:SOURce:SYNThesizer:JITTer:MODulation:AMPLitude:RANGe<range>

Parameter	<range> = <CHARACTER PROGRAM DATA>	
	R_0_22	0.22 UI
	R_2_0	2.00 UI
	R_20	20 UI
	R_200	200 UI
	R_4000	4,000 UI
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?

Response	<range> = <CHARACTER RESPONSE DATA>	
	R_0_22	0.22 UI
	R_2_0	2.00 UI
	R_20	20 UI
	R_200	200 UI
	R_4000	4,000 UI
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>

Parameter	<range> = <CHARACTER PROGRAM DATA>	
	F_1	f/1
	F_64	f/64
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.	

:SOURce:SYNThesizer:JITTer:TRGSource?

Response	<range> = <CHARACTER RESPONSE DATA>
	F_1 f/1
	F_64 f/64
Function	Queries trigger output division ratio.
Example	`> :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

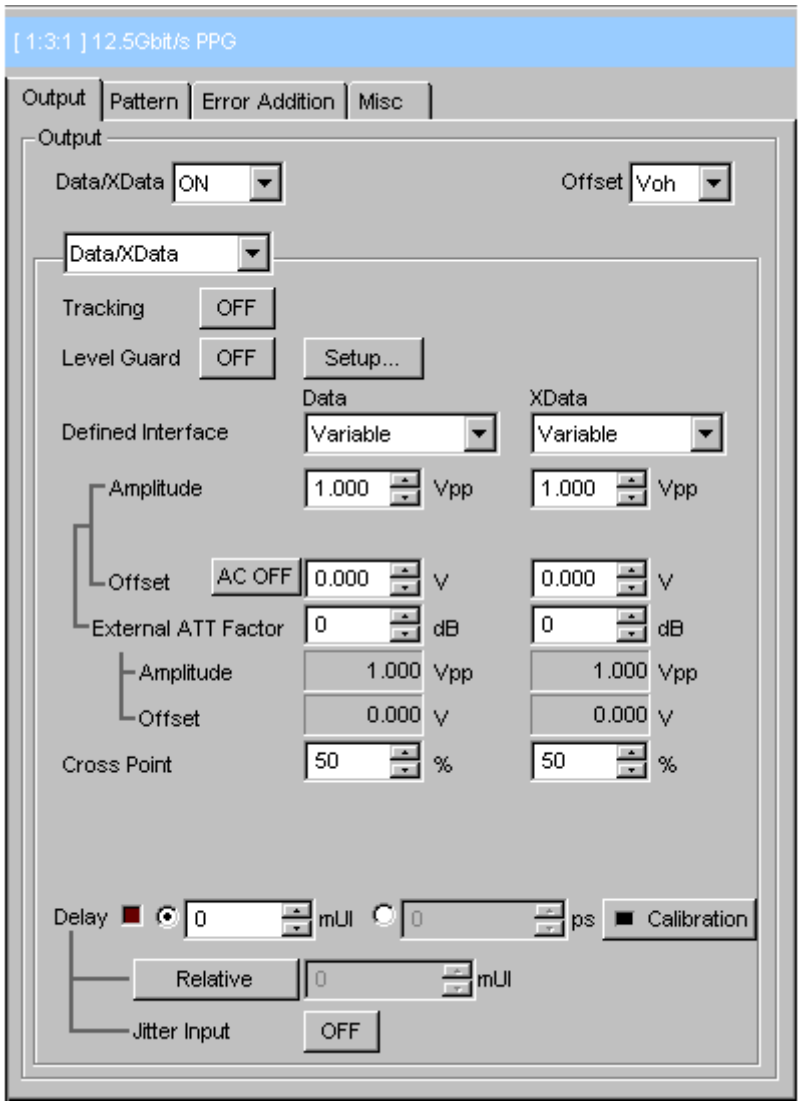


Figure 7.3.1-1 Output tab

7.3.1.1 Output setting commands

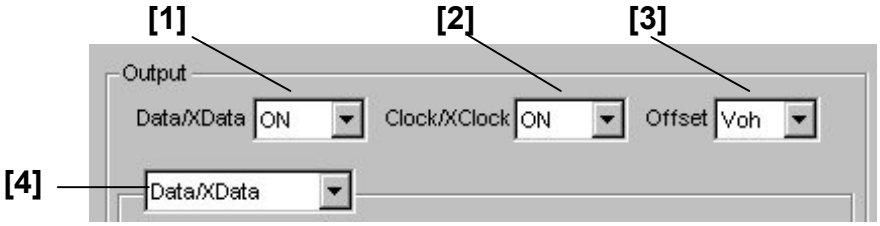


Figure 7.3.1.1-1 Output setting field

Table 7.3.1.1-1 Output setting commands

No.	Setting Items	Commands
[1]	Data/XData	:OUTPut:DATA:OUTPut
		:OUTPut:DATA:OUTPut?
[2]	Clock/XClock	:OUTPut:CLOCk:OUTPut
		:OUTPut:CLOCk:OUTPut?
[3]	Offset	:OUTPut:OFFSet
		:OUTPut:OFFSet?
[4]	No label (Setting item selection)	:OUTPut:SET
		:OUTPut:SET?

: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>

Parameter	<item> = <CHARACTER PROGRAM DATA>	
	DATA	Sets Data/XData.
	CLOCk	Sets Clock/XClock.
	CMU	Sets CMU Bitrate.
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?

Response	<item> = <CHARACTER RESPONSE DATA>
	DATA, CLOC, CMU
Function	Queries the selected setting screen.
Example	`> :OUTPut:SET?`
	`< DATA`
Compatibility	Incompatible with existing models.

7.3.1.2 Data/XData setting commands

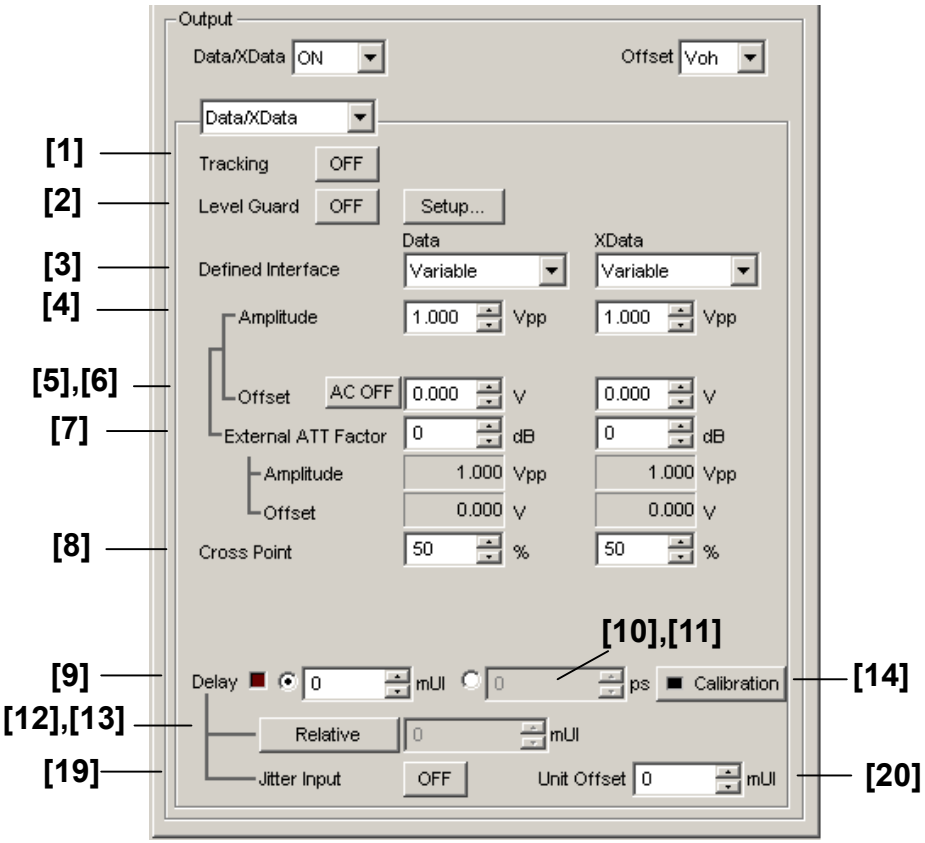


Figure 7.3.1.2-1 Data/XData setting field

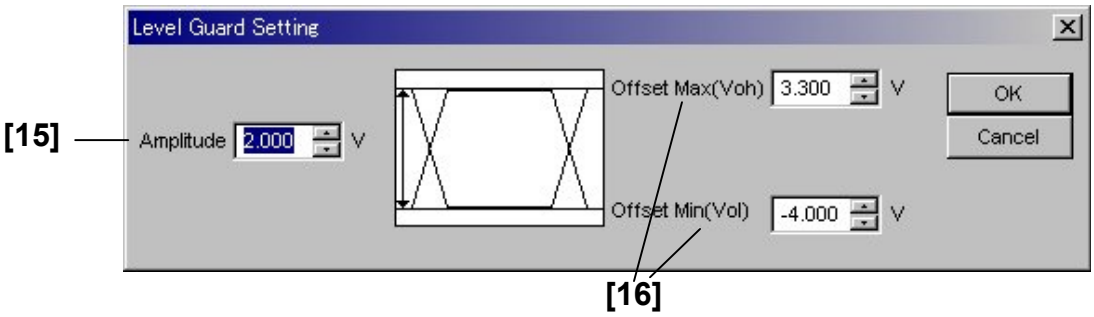


Figure 7.3.1.2-2 Level Guard Setting dialog box

Table 7.3.1.2-1 Data/XData setting commands

No.	Setting Items	Commands
[1]	Tracking	:OUTPut:DATA:TRACking
		:OUTPut:DATA:TRACking?
[2]	Level Guard	:OUTPut:DATA:LEVGuard
		:OUTPut:DATA:LEVGuard?
[3]	Defined Interface	:OUTPut:DATA:LEVel
		:OUTPut:DATA:LEVel?
[4]	Amplitude	:OUTPut:DATA:AMPLitude
		:OUTPut:DATA:AMPLitude?
[5]	AC ON/OFF	:OUTPut:DATA:AOFFset
		:OUTPut:DATA:AOFFset?
[6]	Offset	:OUTPut:DATA:OFFSet
		:OUTPut:DATA:OFFSet?
[7]	External ATT Factor	:OUTPut:DATA:ATTFactor
		:OUTPut:DATA:ATTFactor?
[8]	Cross Point	:OUTPut:DATA:CPOint
		:OUTPut:DATA:CPOint?
[9]	Delay (mUI setting)	:OUTPut:DATA:UIPadjust
		:OUTPut:DATA:UIPadjust?
[10]	Delay (ps setting)	:OUTPut:DATA:PADJust
		:OUTPut:DATA:PADJust?
[11]	Delay (ps setting)	:OUTPut:CLOCk:DELay
		:OUTPut:CLOCk:DELay?
[12]	Relative	:OUTPut:DATA:RELative
		:OUTPut:DATA:RELative?
[13]	No label (Relative value setting)	:OUTPut:DATA:RDELay
		:OUTPut:DATA:RDELay?
[14]	Calibration	:OUTPut:DATA:PCALibration
[15]	Amplitude max	:OUTPut:DATA:LIMitter:AMPLitude
		:OUTPut:DATA:LIMitter:AMPLitude?
[16]	Offset max/min	:OUTPut:DATA:LIMitter:OFFSet
		:OUTPut:DATA:LIMitter:OFFSet?
[17]	No label (Query for status)	:CALCulate:DATA:MONitor?
[18]	No label (Query for data/clock output setting completion status)	:OUTPut:CHANge:CSTate?
[19]	Jitter Input	:OUTPut:DATA:JINPut
		:OUTPut:DATA:JINPut?
[20]	Unit Offset	:OUTPut:DATA:UDELay:OFFSet
		:OUTPut:DATA:UDELay:OFFSet?

: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.	

:OUTPut:DATA:LEVel <port>,<level>

Parameter	<port> = <CHARACTER PROGRAM DATA>	
	DATA	Data output
	XDATa	XData output
	<level> = <CHARACTER PROGRAM DATA>	
	VARiable	Variable
	NECL	NECL level
	PCML	PCML level
	NCML	NCML level
	SCFL	SCFL level
	LVPecl	LVPECL level
	LVDS200	LVDS200 mV
	LVDS400	LVDS400 mV
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>

Parameter	<port> = <CHARACTER PROGRAM DATA>
	DATA, XDATa
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>

Parameter	<port> = <CHARACTER PROGRAM DATA>	
	DATA	DATA output
	XDATa	XData output
	<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 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>

Parameter	<port> = <CHARACTER PROGRAM DATA>	
	DATA, XDATa	
Response	<numeric> = <NR2 NUMERIC RESPONSE DATA>	
	0.050 to 3.500	0.050 to 3.500 V
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>

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	Incompatible with existing models.	

: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	Incompatible with existing models.	

:OUTPut:DATA:OFFSet <port>, <numeric>

Parameter	<port> = <CHARACTER PROGRAM DATA>	
	DATA	Data output
	XDATa	XData output
	<numeric> = <DECIMAL NUMERIC PROGRAM DATA>	
	–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):	
	`> :OUTPut:DATA:OFFSet XDATa,1.000`	
Compatibility	Compatible with the MP1632C.	

:OUTPut:DATA:OFFSet? <port>

Parameter	<port> = <CHARACTER PROGRAM DATA>	
	DATA, XDATa	
Response	<numeric> = <NR2 NUMERIC RESPONSE DATA>	
	–2.000 to 3.300	–2.000 to 3.300 Voh (MU181020A-x10, x11, x12 x13, MU181020B-x11, x12 x13)
	–2.500 to 3.225	–2.500 to 3.225 Vth (MU181020A-x10)
	–3.250 to 3.175	–3.250 to 3.175 Vth (MU181020A-x11, MU181020B-x11)
	–3.000 to 3.275	–3.000 to 3.275 Vth (MU181020A-x12, MU181020B-x12)
	–3.000 to 3.050	–3.000 to 3.050 Vth (MU181020A-x13, MU181020B-x13)
	–3.000 to 3.250	–3.000 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 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> = <CHARACTER PROGRAM DATA>	
	DATA	Data output
	XDATa	XData output
	<numeric> = <DECIMAL NUMERIC PROGRAM DATA>	
	0 to 40	0 to 40 dB, in 1 dB steps
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>

Parameter	<port> = <CHARACTER PROGRAM DATA>
	DATA, XDATa
Response	<numeric> = <NR1 NUMERIC RESPONSE DATA>
	0 to 40 0 to 40 dB, in 1 dB steps
Function	Queries the External ATT Factor of the Data output.
Example	`> :OUTPut:DATA:ATTFactor? DATA`
	`< 20`
Compatibility	Incompatible with existing models.

:OUTPut:DATA:CPOint <port>,<numeric>

Parameter	<port> = <CHARACTER PROGRAM DATA>
	DATA Data output
	XDATa XData output
	<numeric> = <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 output crosspoint.
Example	To set the crosspoint of the XData output to 60%:
	`> :OUTPut:DATA:CPOint XDATa,60`
Compatibility	Compatible with some parameters of the MP1632C (25 to 75%, in 1% steps).

:OUTPut:DATA:CPOint? <port>

Parameter	<port> = <CHARACTER PROGRAM DATA> DATA, XDATa
Response	<numeric> = <NR2 NUMERIC RESPONSE DATA> 20 to 80 — 20 to 80%, in 1% steps (MU181020A-x10, x11, x12, MU181020B-x11, x12) <numeric> = <NR2 NUMERIC RESPONSE DATA> 20.0 to 90.0 — 20.0 to 90.0%, in 0.1% steps (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	<numeric> = <DECIMAL NUMERIC PROGRAM DATA> –1000 to 1000 — –1000 to 1000 mUI, in 1 mUI steps –64000 to 64000 — –64000 to 64000 mUI, in 1 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	<numeric> = <NR1 NUMERIC RESPONSE DATA> –1000 to 1000 — –1000 to 1000 mUI, in 1 mUI steps –64000 to 64000 — –64000 to 64000 mUI, in 1 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.

: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 0.08ps 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.

: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>]

Parameter	<numeric> = <DECIMAL NUMERIC PROGRAM DATA>	
	–2000 to 2000	–2000 to 2000 mUI, in 1 mUI steps
	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. 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>]

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, 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.

:OUTPut:DATA:LIMitter:AMPLitude <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: `> :OUTPut:DATA:LIMitter:AMPLitude 2.000`	
Compatibility	Incompatible with existing models.	

:OUTPut:DATA:LIMitter:AMPLitude?

Response	<numeric> = <NR2 NUMERIC RESPONSE DATA>	
	0.050 to 0.800	0.050 to 0.800 V (MU181020A-x10)
	0.250 to 2.500	0.250 to 2.500 V (MU181020A-x11, MU181020B-x11)
	0.050 to 2.000	0.050 to 2.000 V (MU181020A-x12, MU181020B-x12)
	0.500 to 3.500	0.500 to 3.500 V (MU181020A-x13, MU181020B-x13)
		In 0.002 V steps
Function	Queries the limitation value for the data output amplitude.	
Example	`> :OUTPut:DATA:LIMitter:AMPLitude?` `< 2.000`	
Compatibility	Incompatible with existing models.	

:OUTPut:DATA:LIMitter: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:LIMitter:OFFSet 3.000, -2.000
```

Compatibility

Incompatible with existing models.

:OUTPut:DATA:LIMitter: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:LIMitter:OFFSet?
< 3.000,-2.000
```

Compatibility

Incompatible with existing models.

:CALCulate:DATA:MONitor? <item>

Parameter
<item> = <STRING RESPONSE DATA>
"DBS" Delay busy
"CAL" Required Calibration

Response
<string> = <STRING RESPONSE DATA>

Table 7.3.1.2-2 Response

Form	Format	Description
Form1	"Occur"	When an alarm occurs
	"Not occur"	When no alarm occurs
	"-------"	When no data corresponds to the query

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.

: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

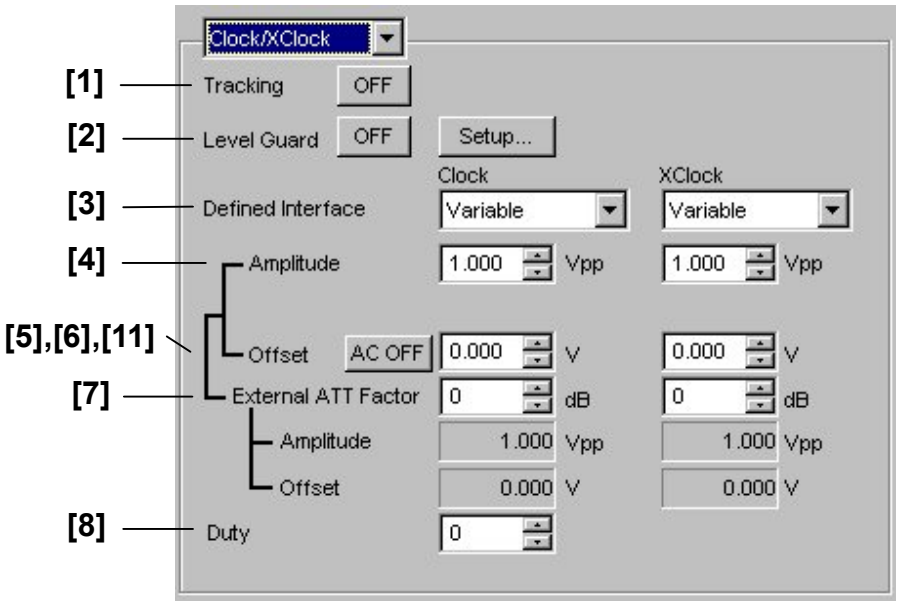


Figure 7.3.1.3-1 Clock/XClock setting field

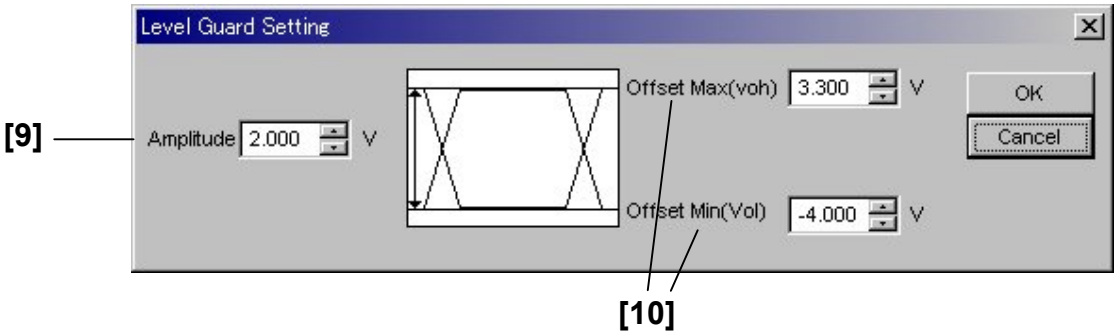


Figure 7.3.1.3-2 Level Guard setting dialog box

Table 7.3.1.3-1 Clock/XClock setting commands

No.	Setting Items	Commands
[1]	Tracking	:OUTPut:CLOCk:TRACking
		:OUTPut:CLOCk:TRACking?
[2]	Level Guard	:OUTPut:CLOCk:LEVGuard
		:OUTPut:CLOCk:LEVGuard?
[3]	Defined Interface	:OUTPut:CLOCk:LEVel
		:OUTPut:CLOCk:LEVel?
[4]	Amplitude	:OUTPut:CLOCk:AMPLitude
		:OUTPut:CLOCk:AMPLitude?
[5]	AC ON/OFF	:OUTPut:CLOCk:AOFFset
		:OUTPut:CLOCk:AOFFset?
[6]	Offset	:OUTPut:CLOCk:OFFSet
		:OUTPut:CLOCk:OFFSet?
[7]	External ATT Factor	:OUTPut:CLOCk:ATTFactor
		:OUTPut:CLOCk:ATTFactor?
[8]	Duty	:OUTPut:CLOCk:DUTY
		:OUTPut:CLOCk:DUTY?
[9]	Amplitude	:OUTPut:CLOCk:LIMitter:AMPLitude
		:OUTPut:CLOCk:LIMitter:AMPLitude?
[10]	Offset Max/Min	:OUTPut:CLOCk:LIMitter:OFFSet
		:OUTPut:CLOCk:LIMitter:OFFSet?
[11]	No label (Query for data/clock output setting completion status)	:OUTPut:CHANge:CSTate? 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:	
	`> :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	`> :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:	
	`> :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	`> :OUTPut:CLOCk:LEVGuard?`	
	`< 1`	
Compatibility	Incompatible with existing models.	

:OUTPut:CLOCk:LEVel <port>,<level>

Parameter	<port> = <CHARACTER PROGRAM DATA>	
	CLOCk	Clock output
	XCLock	XClock output
	<level> = <CHARACTER PROGRAM DATA>	
	VARiable	Variable
	NECL	NECL level
	PCML	PCML level
	NCML	NCML level
	SCFL	SCFL level
	LVPecl	LVPECL level
	LVDS200	LVDS200 mV
	LVDS400	LVDS400 mV
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: `> :OUTPut:CLOCk:LEVel CLOCk,NECL`	
Compatibility	Incompatible with existing models.	

:OUTPut:CLOCk:LEVel? <port>

Parameter	<port> = <CHARACTER PROGRAM DATA>
	CLOCk, XCLock
Response	<level> = <CHARACTER RESPONSE DATA>
	VAR,NECL,PCML,NCML,SCFL,LVP,LVDS200,LVDS400
Function	Queries the level of the fixed interface for the clock output.
Example	`> :OUTPut:CLOCk:LEVel? CLOCk` `< NECL`
Compatibility	Incompatible with existing models.

:OUTPut:CLOCk:AMPLitude <port>,<numeric>

Parameter	<port> = <CHARACTER PROGRAM DATA>	
	CLOCk	Clock output
	XCLock	XClock output
	<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 amplitude of the clock output.	
Example	To set the amplitude of the Clock output to 1.000 V: `> :OUTPut:CLOCk:AMPLitude CLOCk,1.000`	
Compatibility	Compatible with the MP1632C.	

:OUTPut:CLOCk:AMPLitude? <port>

Parameter	<port> = <CHARACTER PROGRAM DATA>	
	CLOCk, XCLock	
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	`> :OUTPut:CLOCk:AOFFset?`	
	`< 1`	
Compatibility	Incompatible with existing models.	

:OUTPut:CLOCk:OFFSet <port>,<numeric>

Parameter	<port> = <CHARACTER PROGRAM DATA>	
	CLOCk	Clock output
	XCLock	XClock output
	<numeric> = <DECIMAL NUMERIC PROGRAM DATA>	
	–2.000 to 3.300	–2.000 to 3.300 Voh, in 0.001 V steps
	–3.000 to 3.250	–3.000 to 3.250 Vth, in 0.001 V steps
	–4.000 to 3.200	–4.000 to 3.200 Vol, in 0.001 V steps
Function	Sets the Clock output offset (for MU181020A-x21, MU181020B-x21only).	
Example	To set the XClock output offset to –1.000 V (Voh):	
	`>:OUTPut:CLOCk:OFFSet XCLock,-1.000`	
Compatibility	Partially compatible with the MP1632C.	

:OUTPut:CLOCk:OFFSet? <port>

Parameter	<port> = <CHARACTER PROGRAM DATA>	
	CLOCk, XCLock	
Response	<numeric> = <NR2 NUMERIC RESPONSE DATA>	
	–2.000 to 3.300	–2.000 to 3.300 Voh
	–3.000 to 3.250	–3.000 to 3.250 Vth
	–4.000 to 3.200	–4.000 to 3.200 Vol
Function	Queries the Clock output offset.	
Example	To query the XClock output offset:	
	`> :OUTPut:CLOCk:OFFSet? XCLock`	
	`< -1.000`	
Compatibility	Partially compatible with the MP1632C.	

:OUTPut:CLOCk:ATTFactor <port>,<numeric>

Parameter	<port> = <CHARACTER PROGRAM DATA>	
	CLOCk	Clock output
	XCLock	XClock output
	<numeric> = <DECIMAL NUMERIC PROGRAM DATA>	
	0 to 40	0 to 40 dB, in 1 dB steps
Function	Sets the External ATT factor of the Clock output.	
Example	To set the External ATT factor of the Clock output to 20 dB:	
	`> :OUTPut:CLOCk:ATTFactor CLOCk,20`	
Compatibility	Incompatible with existing models.	

: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.

:OUTPut:CLOCk:LIMitter:AMPLitude?

Response	<numeric> = <NR2 NUMERIC RESPONSE DATA>
	0.100 to 2.000 0.100 to 2.000 V (MU181020A-x21, MU181020B-x21)
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>

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 (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?

Response	<max> = <NR2 NUMERIC RESPONSE DATA>
	–2.000 to 3.300 –2.000 to 3.300 V
	<min> = <NR2 NUMERIC RESPONSE DATA>
	–4.000 to 3.200 –4.000 to 3.200 V
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

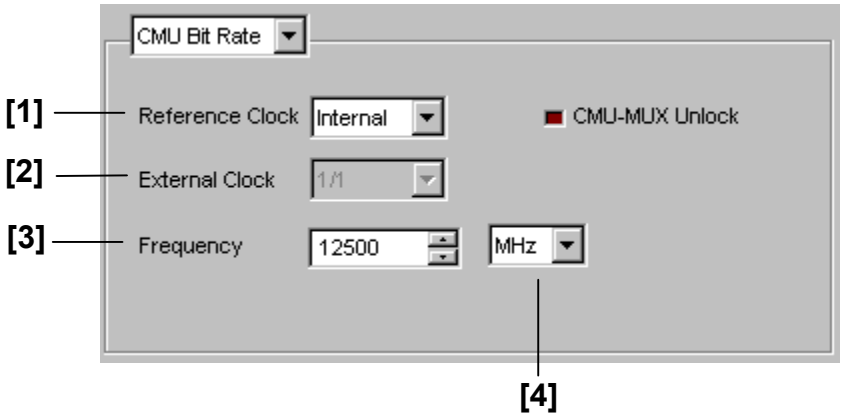


Figure 7.3.1.4-1 CMU Bit Rate setting field (when Reference Clock is set to Internal)

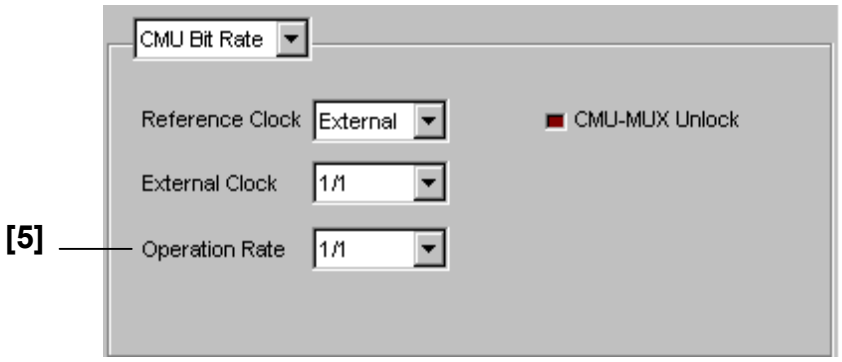


Figure 7.3.1.4-2 CMU Bit Rate setting field (when Reference Clock is set to External)

Table 7.3.1.4-1 CMU Bit Rate setting commands

No.	Setting Items	Commands
[1]	Reference Clock	:OUTPut:CMU:REFClock
		:OUTPut:CMU:REFClock?
[2]	External Clock	:OUTPut:CMU:EXTClock
		:OUTPut:CMU:EXTClock?
[3]	Frequency	:OUTPut:CMU:FREQuency
		:OUTPut:CMU:FREQuency?
[4]	No label (Unit setting)	:OUTPut:CMU:RESolution
		:OUTPut:CMU:RESolution?
[5]	Operation Rate	:OUTPut:CMU:OPERation
		:OUTPut:CMU:OPERation?

: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.

: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.

: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?

Response	<numeric> = <NR1 NUMERIC RESPONSE 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	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

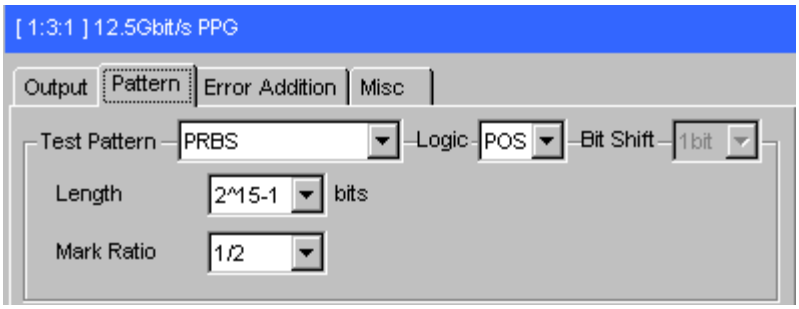


Figure 7.3.2-1 Pattern tab

7.3.2.1 Test Pattern setting commands

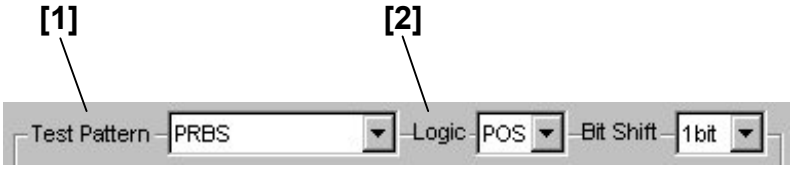


Figure 7.3.2.1-1 Test Pattern setting field

Table 7.3.2.1-1 Test Pattern setting commands

No.	Setting Items	Commands
[1]	Test Pattern	:SOURce:PATTern:TYPE
		:SOURce:PATTern:TYPE?
[2]	Logic	:SOURce:PATTern:LOGic
		:SOURce:PATTern:LOGic?

:SOURce:PATTern:TYPE <type>

Parameter	<type> = <CHARACTER PROGRAM DATA>	
	PRBS	PRBS pattern
	ZSUBstitution	Zero-substitution pattern
	DATA	Data pattern
	ALTernate	Alternate pattern
	MIXData	Mixed Data pattern
	MIXalt	Mixed Alternate pattern
	SEQuence	Sequence 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`	
Compatibility	Partially compatible with the MP1632C (ZSUBstitution only).	

:SOURce:PATTern:TYPE?

Response	<type> = <CHARACTER RESPONSE DATA> PRBS, ZSUB, DATA, ALT, MIXD, MIX, SEQ
Function	Queries the type of the test pattern.
Example	`> :SOURce:PATTern:TYPE?` `< MIXD`
Compatibility	Partially compatible with the MP1632C (ZSUBstitution only).

:SOURce: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 negative logic (NEGative): `> :SOURce:PATTern:LOGic NEGative`	
Compatibility	Incompatible with existing models.	

:SOURce:PATTern:LOGic?

Response	<logic> = <CHARACTER RESPONSE DATA> POS, NEG
Function	Queries the logic of the test pattern.
Example	`> :SOURce:PATTern:LOGic?` `< NEG`
Compatibility	Incompatible with existing models.

7.3.2.2 PRBS pattern setting commands

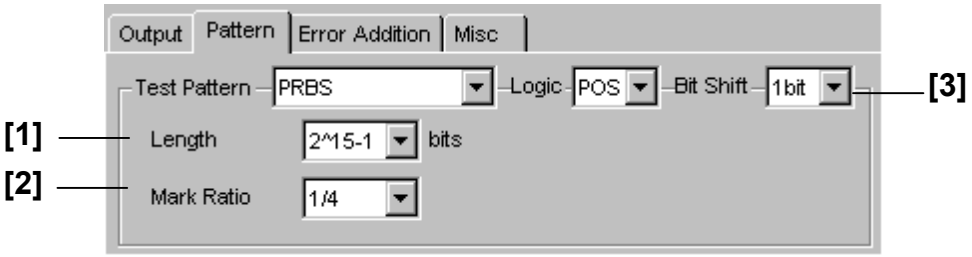


Figure 7.3.2.2-1 PRBS pattern setting

Table 7.3.2.2-1 PRBS pattern setting commands

No.	Setting Items	Commands
[1]	Length	:SOURce:PATTern:PRBS:LENGth
		:SOURce:PATTern:PRBS:LENGth?
[2]	Mark Ratio	:SOURce:PATTern:PRBS:MRATio
		:SOURce:PATTern:PRBS:MRATio?
[3]	Bit Shift	:SOURce:PATTern:PRBS:BSHift
		:SOURce:PATTern:PRBS:BSHift?

:SOURce:PATTern: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 (2^n-1 (n = 7, 9, 10, 11, 15, 20, 23, or 31)) during PRBS pattern generation.	
Example	To set PRBS15:	
	`> :SOURce:PATTern:PRBS:LENGth 15`	
Compatibility	Incompatible with existing models.	

: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	Incompatible with existing models.

:SOURce:PATTern: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 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

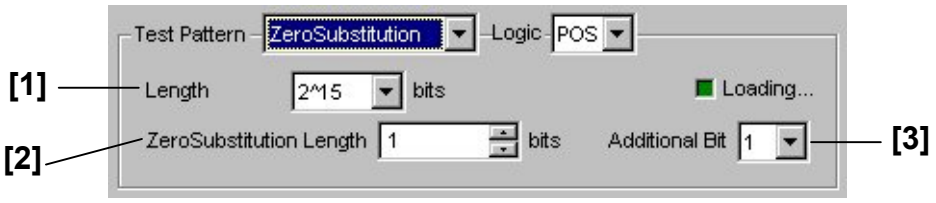


Figure 7.3.2.3-1 Zero-Substitution pattern setting

Table 7.3.2.3-1 Zero-Substitution pattern setting commands

No.	Setting Items	Commands
[1]	Length	:SOURce:PATTern:ZSUBstitute:LENGth
		:SOURce:PATTern:ZSUBstitute:LENGth?
[2]	Zero Substitution Length	:SOURce:PATTern:ZSUBstitute:ZLENgth
		:SOURce:PATTern:ZSUBstitute:ZLENgth?
[3]	Additional Bit	:SOURce:PATTern:ZSUBstitute:ADDBit
		:SOURce:PATTern:ZSUBstitute:ADDBit?

: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 the MP1632C.

:SOURce: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 generation.
Example	`> :SOURce:PATTern:ZSUBstitute:LENGth?` `< 15`
Compatibility	Compatible with the MP1632C.

:SOURce: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, 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	<numeric> = <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>

Parameter	<numeric> = <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.	
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?

Response	<numeric> = <NR1 NUMERIC RESPONSE 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	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

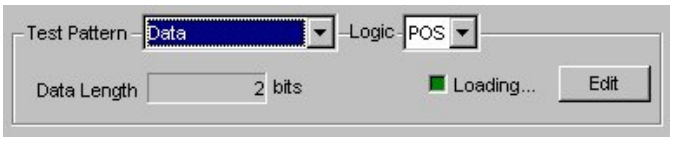


Figure 7.3.2.4-1 Data pattern setting

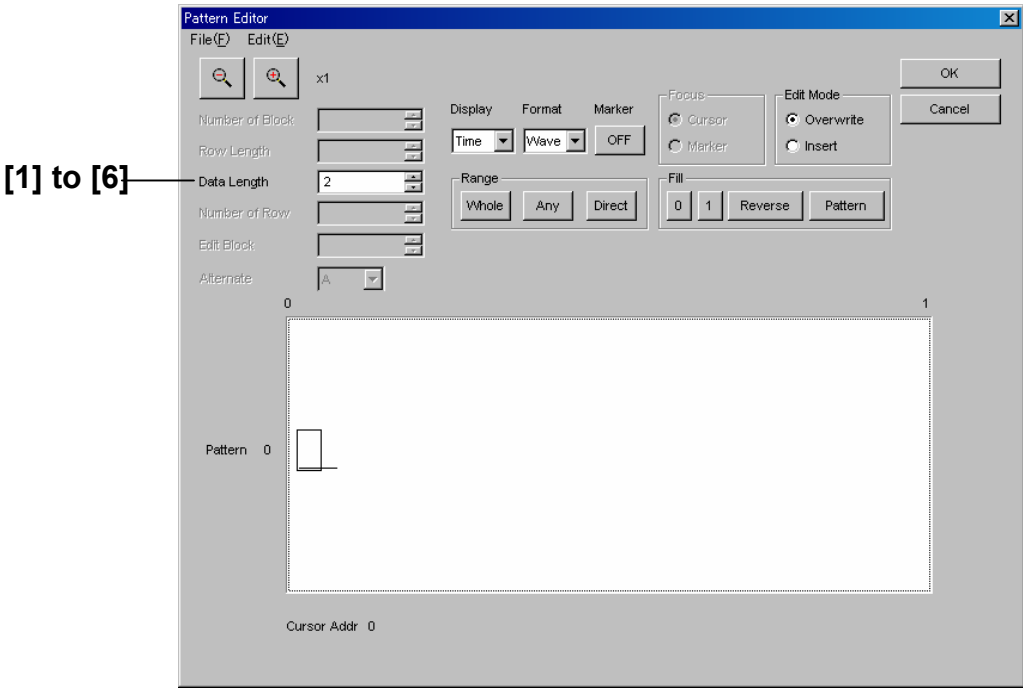


Figure 7.3.2.4-2 Data pattern setting (Pattern Editor screen)

Table 7.3.2.4-1 Data pattern setting commands

No.	Setting Items	Commands
[1]	Data Length	:SOURce:PATTern:DATA:LENGth
		:SOURce:PATTern:DATA:LENGth?
[2]	No label (Pattern setting)	:SOURce:PATTern:DATA:WHOLe
		:SOURce:PATTern:DATA:WHOLe?
[3]	No label (Pattern setting)	:SOURce:PATTern:BDATa:WHOLe
		:SOURce:PATTern:BDATa:WHOLe?
[4]	No label (Data inversion)	:SOURce:PATTern:DREVerse:ADDRess
[5]	No label (Data inversion)	:SOURce:PATTern:DREVerse:DELTa
[6]	No label (All 0 or All 1)	:SOURce:PATTern:DATA:FILL

: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>

#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 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

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.

<Example>

- <start> = #H0, <end> = #HF, <data> = #HABCDEF
  Setting data: ABCD
- <start> = #H0, <end> = #H3, <data> = #B01100110
  Setting data: 0110

:SOURce:PATTern:DATA:WHOLe? <start>

Parameter

<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 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 generation.

Example

To query the pattern data from address 0:

```
> :SOURce:PATTern:DATA:WHOLe? #H0
< "H5A"
```

Compatibility

Compatible with the MP1632C.

:SOURce:PATTern:BDATa:WHOLe <start>,<end>,<bdata>

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.

<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.

:SOURce:PATTern:BDATa:WHOLe? <start>[,<size>]

Parameter

<start> = <NON-DECIMAL PROGRAM DATA>

#H0 to #H7FFFFFF	0 to 7FFFFF 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 1,6777,216 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 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:

```
> :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:

```
> :SOURce:PATTern:BDATa:WHOLe? #H0,2
< #12AA
```

Compatibility

Compatible with the MP1632C.

:SOURce:PATTern:DREVerse:ADDRess <start>,<end>

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.

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 #H7FFFFFF	0 to 7FFFFFF 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.

: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

$$\text{Max} = \frac{\text{Pattern Length}}{128} \text{ (rounding up fractions)}$$

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

Incompatible with existing models.

7.3.2.5 Alternate pattern setting commands

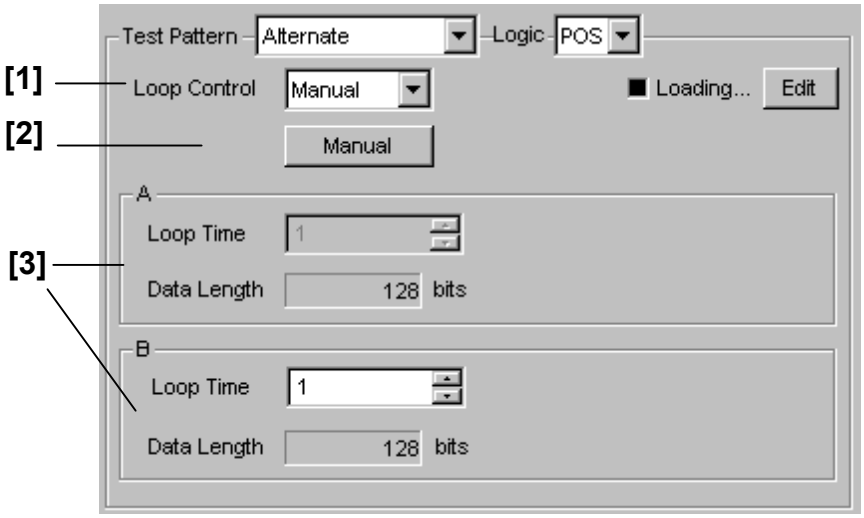


Figure 7.3.2.5-1 Alternate pattern setting

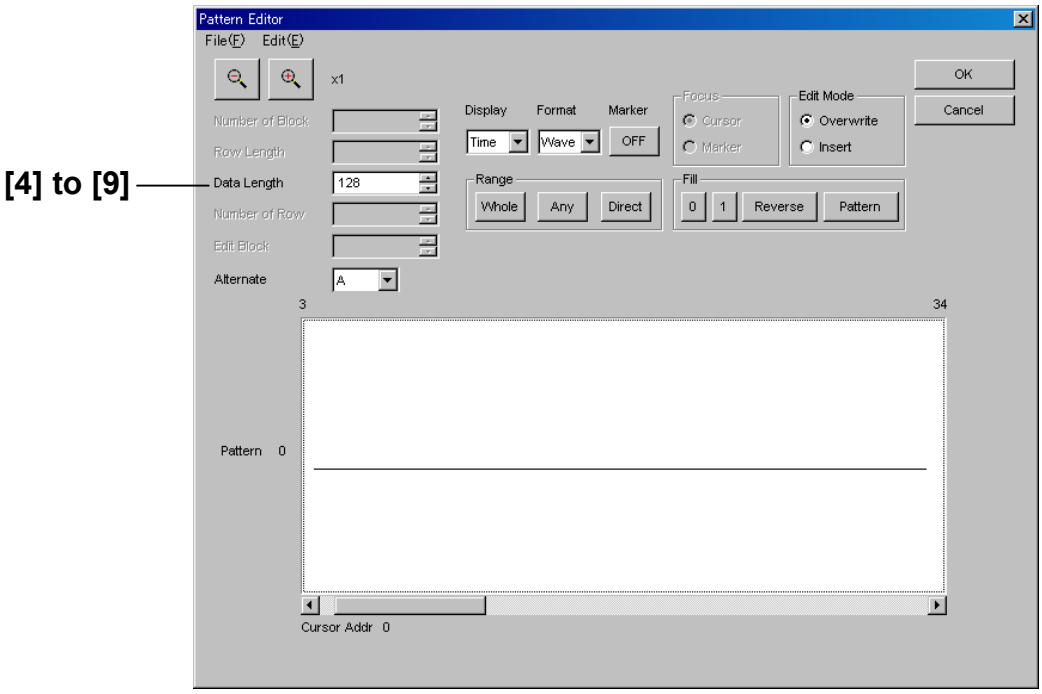


Figure 7.3.2.5-2 Alternate pattern setting (Pattern Editor screen)

Table 7.3.2.5-1 Alternate pattern setting commands

No.	Setting Items	Commands
[1]	Loop Control	:SOURce:PATTern:ALTernate:LOOPcont
		:SOURce:PATTern:ALTernate:LOOPcont?
[2]	Manual	:SOURce:PATTern:ALTernate:MANual
[3]	A/B LoopTime	:SOURce:PATTern:ALTernate:LTIMe
		:SOURce:PATTern:ALTernate:LTIMe?
[4]	Data Length	:SOURce:PATTern:ALTernate:LENGth
		:SOURce:PATTern:ALTernate:LENGth?
[5]	No label (Pattern setting)	:SOURce:PATTern:ALTernate:DATA:WHOLe
		:SOURce:PATTern:ALTernate:DATA:WHOLe?
[6]	No label (Pattern setting)	:SOURce:PATTern:ALTernate:BDATa:WHOLe
		:SOURce:PATTern:ALTernate:BDATa:WHOLe?
[7]	No label (Data inversion)	:SOURce:PATTern:ALTernate:DREVerse:ADDRess
[8]	No label (Data inversion)	:SOURce:PATTern:ALTernate:DREVerse:DELTa
[9]	No label (All 0 or All 1)	:SOURce:PATTern:ALTernate:DATA:FILL

:SOURce:PATTern:ALTernate: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:ALTernate:LOOPcont INTernal
```

Compatibility Incompatible with existing models.

:SOURce:PATTern:ALTernate:LOOPcont?

Response <loop> = <CHARACTER RESPONSE DATA>

INT, EXT, MAN

Function Queries the Alternate pattern A/B switching control method.

Example

```
> :SOURce:PATTern:ALTernate:LOOPcont?
< INT
```

Compatibility Incompatible with existing models.

:SOURce:PATTern:ALTernate:MANual

Function Switches the Alternate pattern A/B manually.

Example

```
> :SOURce:PATTern:ALTernate:MANual
```

Compatibility Incompatible with existing models.

: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.

: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.

:SOURce:PATTern: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 3FFFFFF 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:PATTern:ALTernate:DATA:WHOLe? APAT,H0
< "HAA"
```

Compatibility

Incompatible with existing models.

: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>]

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)

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.

:SOURce:PATTern:ALTernate: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:ALTernate:DREVerse:ADDRess APAT,#H0,#H1F
```

Compatibility

Incompatible with existing models.

:SOURce:PATTern:ALTernate: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:ALTernate:DREVerse:DELTa APAT,#H0,64
```

Compatibility

Incompatible with existing models.

:SOURce:PATTern:ALTernate:DATA:FILL <content>,<range>,<page>,<data>

Parameter

<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

$$\text{Max} = \frac{\text{Pattern Length}}{128} \text{ (rounding up fractions)}$$

Specify "0" when <range> is set to ALL.

<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 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

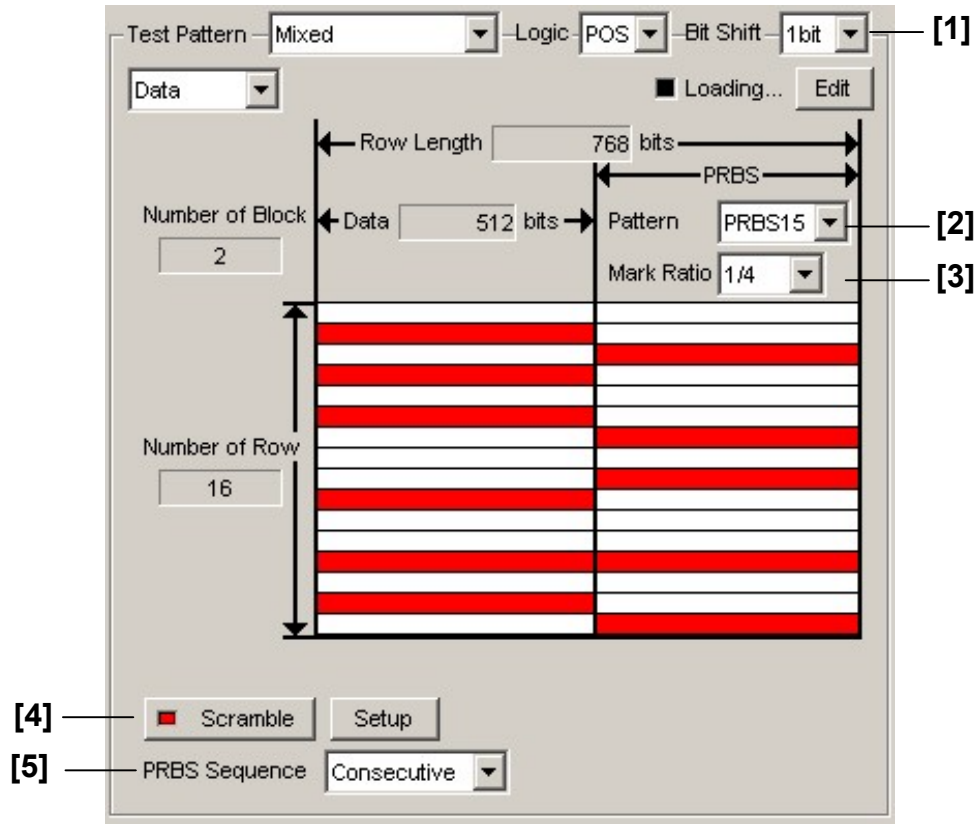


Figure 7.3.2.6-1 Mixed Data pattern setting

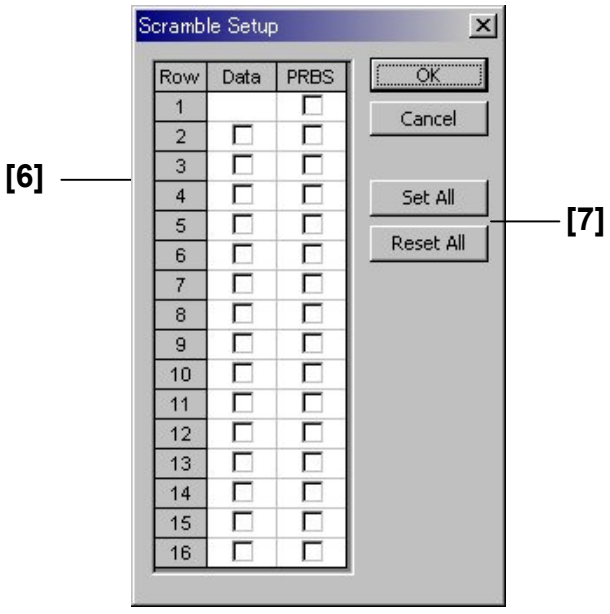


Figure 7.3.2.6-2 Scramble Setup dialog box

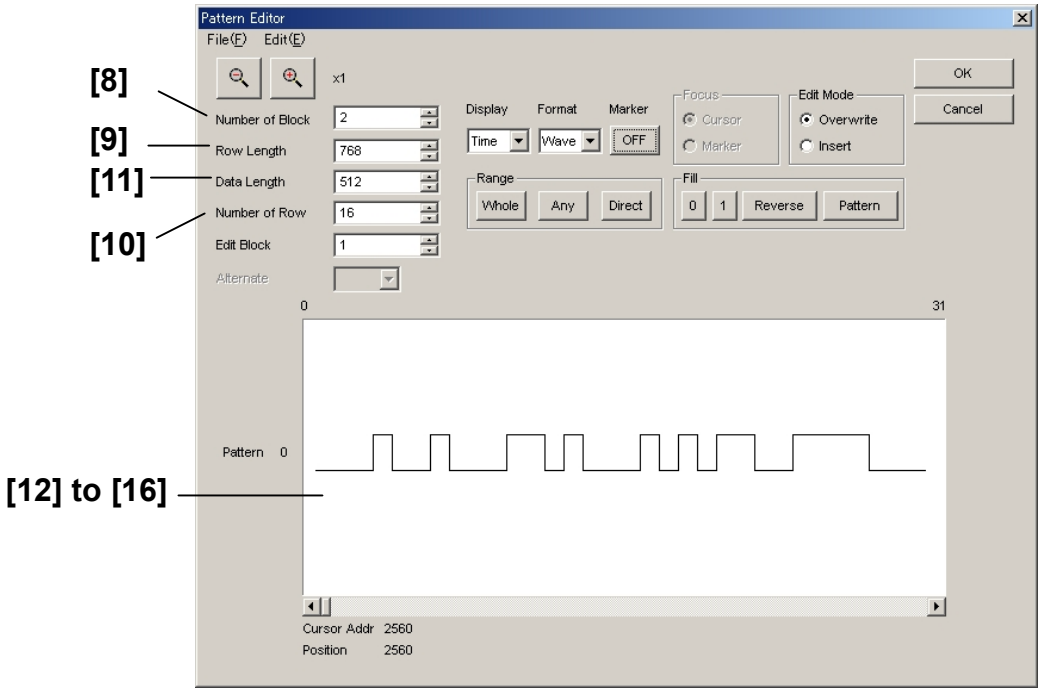


Figure 7.3.2.6-3 Mixed Data pattern setting (Pattern Editor screen)

Table 7.3.2.6-1 Mixed Data pattern setting (Pattern Editor screen)

No.	Setting Items	Commands
[1]	Bit Shift	:SOURce:PATTern:MIXData:PRBS:BITShift
		:SOURce:PATTern:MIXData:PRBS:BITShift?
[2]	Pattern	:SOURce:PATTern:MIXData:PRBS:LENGth
		:SOURce:PATTern:MIXData:PRBS:LENGth?
[3]	Mark Ratio	:SOURce:PATTern:MIXData:PRBS:MRATio
		:SOURce:PATTern:MIXData:PRBS:MRATio?
[4]	Scramble	:SOURce:PATTern:MIXData:SCRamble
		:SOURce:PATTern:MIXData:SCRamble?
[5]	PRBS Sequence	:SOURce:PATTern:MIXData:PRBS:SEQuence
		:SOURce:PATTern:MIXData:PRBS:SEQuence?
[6]	Setup	:SOURce:PATTern:MIXData:SRSetting
		:SOURce:PATTern:MIXData:SRSetting?
[7]	Set All/Reset All	:SOURce:PATTern:MIXData:SCRamble:ALLSet
[8]	Number of Block	:SOURce:PATTern:MIXData:BLOCk
		:SOURce:PATTern:MIXData:BLOCk?
[9]	Row Length	:SOURce:PATTern:MIXData:RLENgth
		:SOURce:PATTern:MIXData:RLENgth?
[10]	Number of Row	:SOURce:PATTern:MIXData:ROW
		:SOURce:PATTern:MIXData:ROW?
[11]	Data Length	:SOURce:PATTern:MIXData:DATA:LENGth
		:SOURce:PATTern:MIXData:DATA:LENGth?

Table 7.3.2.6-1 Mixed Data pattern setting (Pattern Editor screen) (Cont'd)

No.	Setting Items	Commands
[12]	No label (Pattern setting)	:SOURce:PATTern:MIXData:DATA:WHOLe
		:SOURce:PATTern:MIXData:DATA:WHOLe?
[13]	No label (Pattern setting)	:SOURce:PATTern:MIXData:BDATa:WHOLe
		:SOURce:PATTern:MIXData:BDATa:WHOLe?
[14]	No label (Data inversion)	:SOURce:PATTern:MIXData:DREVerse:ADDRess
[15]	No label (Data inversion)	:SOURce:PATTern:MIXData:DREVerse:DELTa
[16]	No label (All 0 or All 1)	:SOURce:PATTern:MIXData:DATA:FILL

: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.

: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.	

: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 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	Incompatible with existing models.

:SOURce:PATTern:MIXData:SCRamble?

Response	<numeric> = <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	Incompatible with existing models.

: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	Incompatible with existing models.

: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.

: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	`> :SOURce:PATTern:MIXData:BLOCk?` `< 30`
Compatibility	Incompatible with existing models.

:SOURce:PATTern:MIXData:RLENgth <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:RLENgth 768`
Compatibility	Incompatible with existing models.

: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.

: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.

: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>

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 7FFFFF 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 Mixed Data pattern generation.

Example

To query the block 1 pattern data from address 0:

```
> :SOURce:PATTern:MIXData:DATA:WHOLe? 1,#H0
< "H11"
```

Compatibility

Incompatible with existing models.

:SOURce:PATTern:MIXData:BDATa:WHOLe <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 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.

: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 7FFFFF 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>

#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 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 <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 #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.

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

Incompatible with existing models.

:SOURce:PATTern:MIXData:DREVerse:DELTa <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 #H7FFFFFF 0 to 7FFFFFF 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 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

Incompatible with existing models.

:SOURce: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

$$\text{Max} = \frac{\text{Pattern Length}}{128} \text{ (rounding up fractions)}$$

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

Incompatible with existing models.

7.3.2.7 Mixed Alternate pattern setting commands

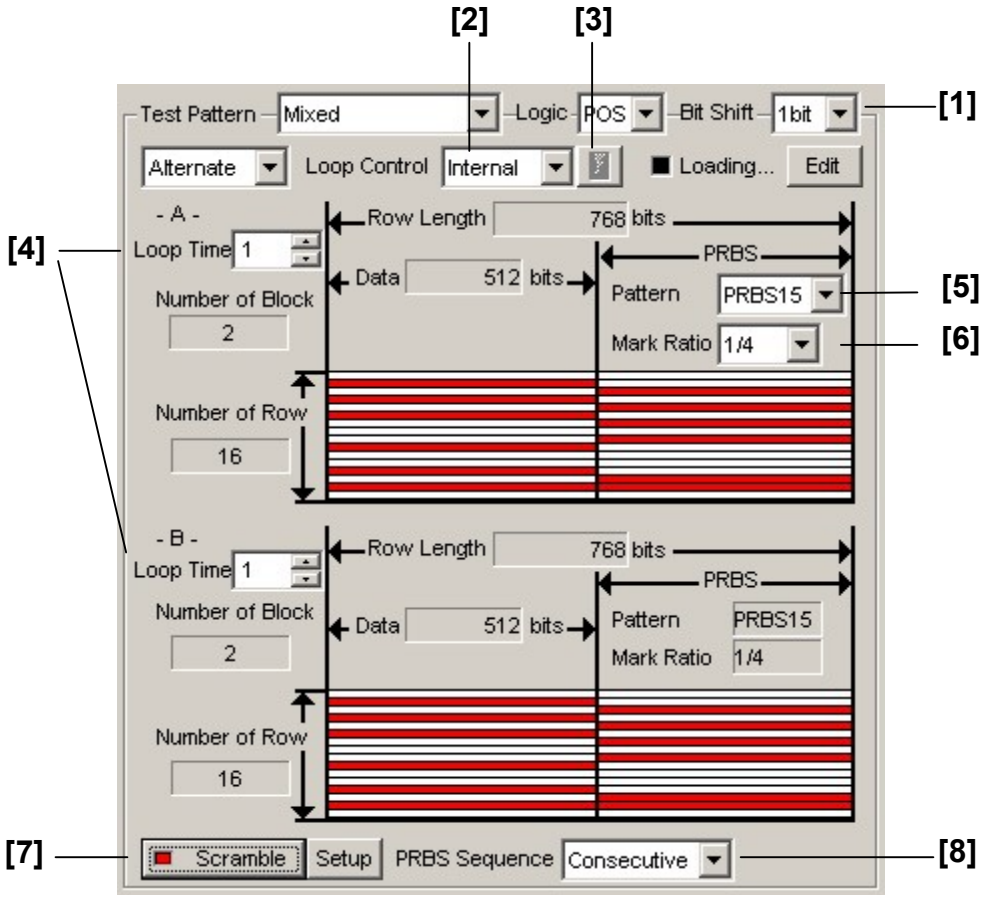


Figure 7.3.2.7-1 Mixed Alternate pattern setting

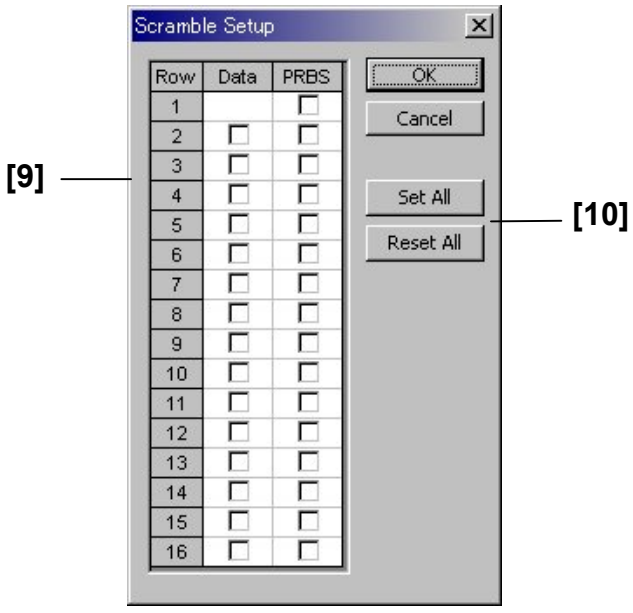


Figure 7.3.2.7-2 Scramble Setup dialog box

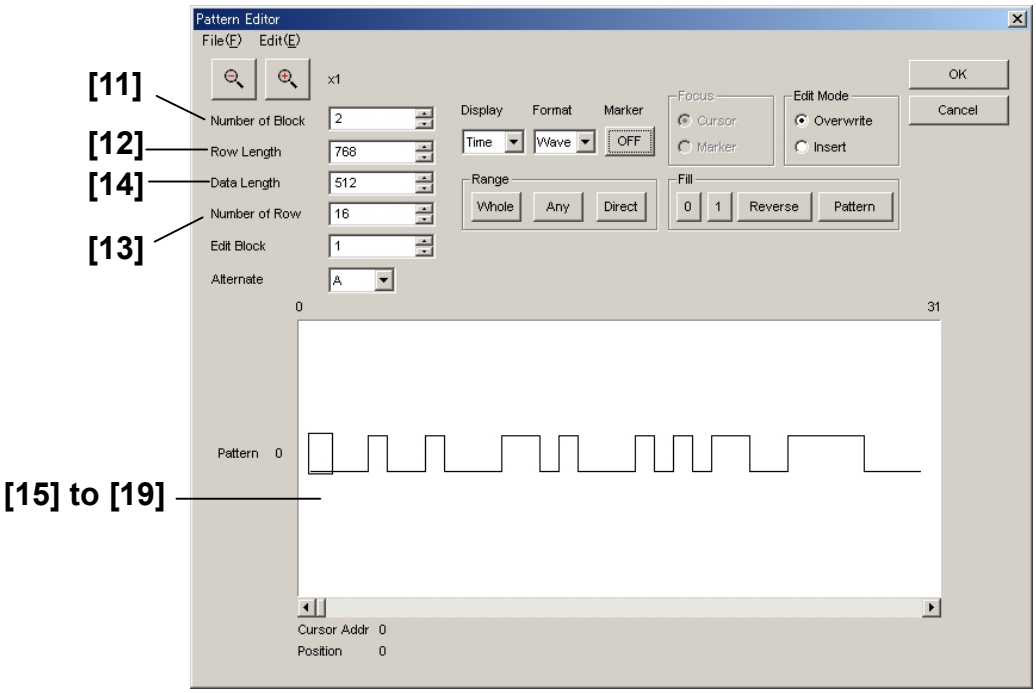


Figure 7.3.2.7-3 Mixed Alternate pattern setting (Pattern Editor screen)

Table 7.3.2.7-1 Mixed Alternate pattern setting commands

No.	Setting Items	Commands
[1]	Bit Shift	:SOURce:PATTern:MIXalt:PRBS:BITShift
		:SOURce:PATTern:MIXalt:PRBS:BITShift?
[2]	Loop Control	:SOURce:PATTern:MIXalt:LOOPcont
		:SOURce:PATTern:MIXalt:LOOPcont?
[3]	Manual button	:SOURce:PATTern:MIXalt:MANual
[4]	Loop Time	:SOURce:PATTern:MIXalt:LTIMe?
		:SOURce:PATTern:MIXalt:LTIMe?
[5]	Pattern	:SOURce:PATTern:MIXalt:PRBS:LENGth
		:SOURce:PATTern:MIXalt:PRBS:LENGth?
[6]	Mark Ratio	:SOURce:PATTern:MIXalt:PRBS:MRATio
		:SOURce:PATTern:MIXalt:PRBS:MRATio?
[7]	Scramble	:SOURce:PATTern:MIXalt:SCRamble
		:SOURce:PATTern:MIXalt:SCRamble?
[8]	PRBS Sequence	:SOURce:PATTern:MIXalt:PRBS:SEQuence
		:SOURce:PATTern:MIXalt:PRBS:SEQuence?
[9]	Scramble Setup	:SOURce:PATTern:MIXalt:SRSetting
		:SOURce:PATTern:MIXalt:SRSetting?
[10]	Set All/Reset All	:SOURce:PATTern:MIXalt:SCRamble:ALLSet
[11]	Number of Block	:SOURce:PATTern:MIXalt:BLOCk
		:SOURce:PATTern:MIXalt:BLOCk?

Table 7.3.2.7-1 Mixed Alternate pattern setting commands (Cont'd)

No.	Setting Items	Commands
[12]	Row Length	:SOURce:PATTern:MIXalt:RLENgth
		:SOURce:PATTern:MIXalt:RLENgth?
[13]	Number of Row	:SOURce:PATTern:MIXalt:ROW
		:SOURce:PATTern:MIXalt:ROW?
[14]	Data Length	:SOURce:PATTern:MIXalt:DATA:LENGth
		:SOURce:PATTern:MIXalt:DATA:LENGth?
[15]	No label (Pattern setting)	:SOURce:PATTern:MIXalt:DATA:WHOLe
		:SOURce:PATTern:MIXalt:DATA:WHOLe?
[16]	No label (Pattern setting)	:SOURce:PATTern:MIXalt:BDATa:WHOLe
		:SOURce:PATTern:MIXalt:BDATa:WHOLe?
[17]	No label (Data inversion)	:SOURce:PATTern:MIXalt:DREVerse:ADDRess
[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.

: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.	

:SOURce:PATTern:MIXalt: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	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	<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 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	<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 Alternate pattern generation.
Example	`> :SOURce:PATTern:MIXalt:PRBS:LENGth?` `< 7`
Compatibility	Incompatible with existing models.

:SOURce:PATTern:MIXalt: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 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?

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 Alternate pattern generation.
Example	`> :SOURce:PATTern:MIXalt:PRBS:MRATio?` `< M1_8`
Compatibility	Incompatible with existing models.

:SOURce:PATTern:MIXalt: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:MIXalt:SCRamble 1`	
Compatibility	Incompatible with existing models.	

:SOURce:PATTern:MIXalt:SCRamble?

Response	<numeric> = <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: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: `> :SOURce:PATTern:MIXalt:PRBS:SEQuence RESTart`
Compatibility	Incompatible with existing models.

:SOURce:PATTern:MIXalt:PRBS:SEQuence?

Response	<sequence> = <CHARACTER RESPONSE DATA> REST, CONS
Function	Queries the PRBS pattern sequence during Mixed Alternate pattern generation.
Example	`> :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 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 Scramble ON/OFF cannot be set for Data of 1 Row.
	<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 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 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.

: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.

: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.

:SOURce:PATTern:MIXalt:DATA:WHOLe <block>,<content>,<start>,<end>,<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

<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 Mixed Alternate pattern generation.
The set data overwrites the specified range.

Example

To set the addresses 0 to 1F of the block 1 pattern A data to AA:

```
> :SOURce:PATTern:MIXalt:DATA:WHOLe 1,APAT,#H0,#H1F, "HAA"
```

Compatibility

Incompatible with existing models.

: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 3FFFFFF 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.

:SOURce:PATTern:MIXalt:BDATa:WHOLe <content>,<start>,<end>,<bdata>

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 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 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
```

(B=42)

Compatibility

Incompatible with existing models.

:SOURce:PATTern:MIXalt:BDATa:WHOLe? <block>,<content>,<start>[,<size>]

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 3FFFFFF 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.

:SOURce:PATTern:MIXalt:DREVerse:ADDRess <block>,<content>,<start>,<end>

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 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 Mixed Alternate pattern generation. Specify the inversion range by the <start> and <end> addresses.

Example

To invert addresses 0 to 1F of the block 3 pattern A data:

```
> :SOURce:PATTern:MIXalt:DREVerse:ADDRess 3,APAT,#H0,#H1F
```

Compatibility

Incompatible with existing models.

:SOURce:PATTern:MIXalt:DREVerse:DELTa <block>,<content>,<start>,<delta>

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 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 Mixed 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 block 3 pattern A data from address 0:

```
> :SOURce:PATTern:MIXalt:DREVerse:DELTa 3,APAT,#H0,64
```

Compatibility

Incompatible with existing models.

:SOURce:PATTern:MIXalt:DATA:FILL <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

$$\text{Max} = \frac{\text{Pattern Length}}{128} \text{ (rounding up fractions)}$$

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

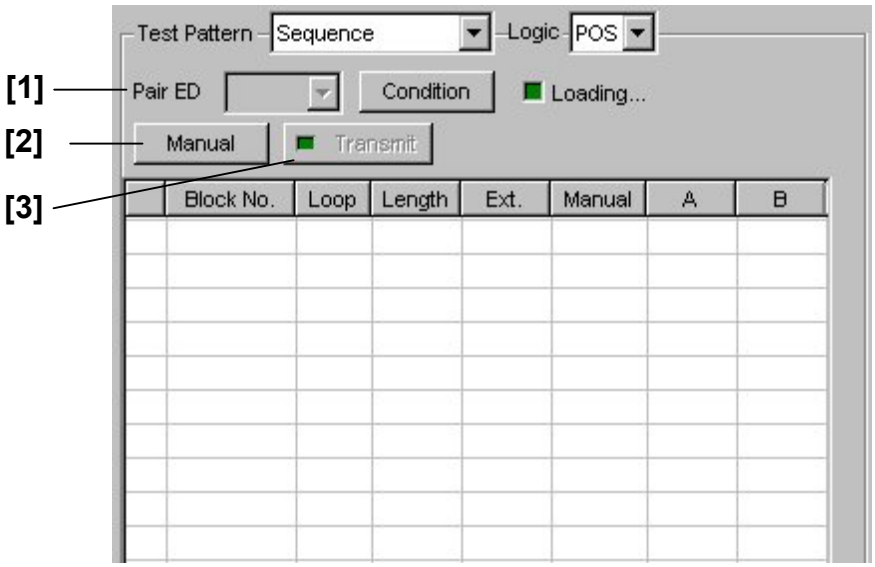


Figure 7.3.2.8-1 Sequence pattern setting

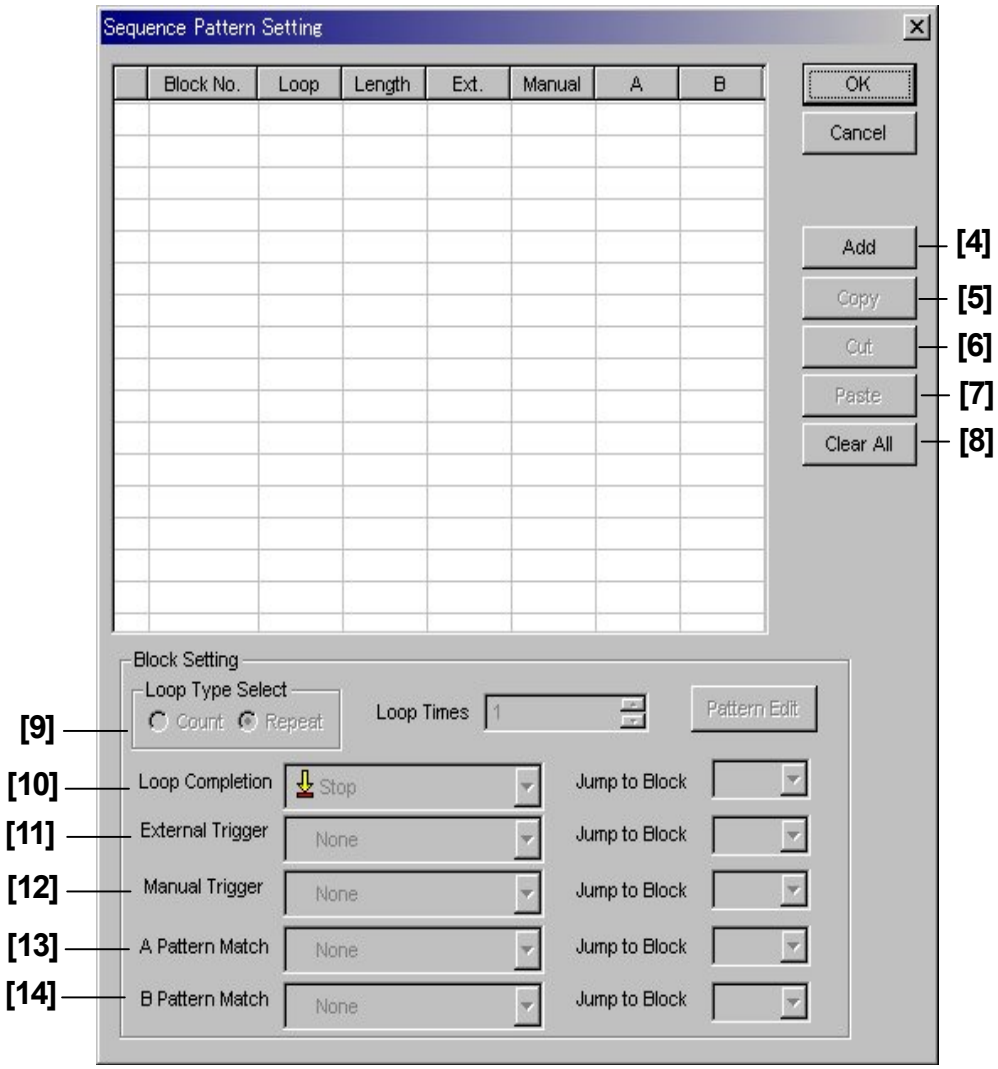


Figure 7.3.2.8-2 Sequence Pattern Setting screen

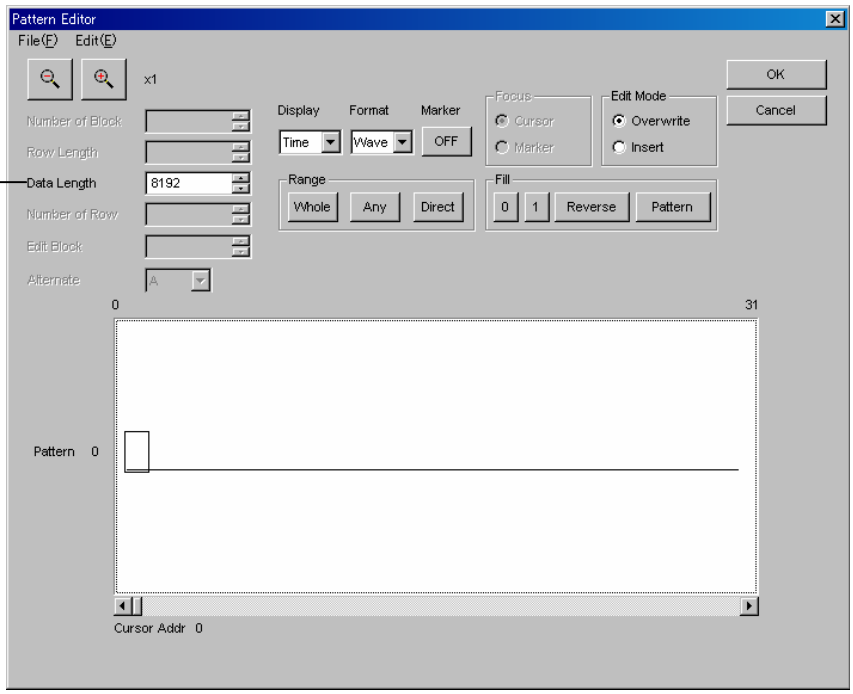


Figure 7.3.2.8-3 Sequence pattern setting (Pattern Editor screen)

Table 7.3.2.8-1 Sequence pattern setting commands

No.	Setting Items	Commands
[1]	Pair ED	:SOURce:PATTern:SEQuence:PAIR
		:SOURce:PATTern:SEQuence:PAIR?
[2]	Manual	:SOURce:PATTern:SEQuence:MANual
[3]	Restarting	:SOURce:PATTern:SEQuence:TRANsmit
[4]	Add	:SOURce:PATTern:SEQuence:CONDition:ADD
		:SOURce:PATTern:SEQuence:CONDition:ADD?
[5]	Copy	:SOURce:PATTern:SEQuence:COPY
[6]	Cut	:SOURce:PATTern:SEQuence:CUT
[7]	Paste	:SOURce:PATTern:SEQuence:PASTe
[8]	Clear All	:SOURce:PATTern:SEQuence:CLEar
[9]	Loop times	:SOURce:PATTern:SEQuence:LOOPtime
		:SOURce:PATTern:SEQuence:LOOPtime?
[10]	Loop Completion	:SOURce:PATTern:SEQuence:CONDition:LOPC
		:SOURce:PATTern:SEQuence:CONDition:LOPC?
[11]	External Trigger	:SOURce:PATTern:SEQuence:CONDition:EXTernal
		:SOURce:PATTern:SEQuence:CONDition:EXTernal?
[12]	Manual Trigger	:SOURce:PATTern:SEQuence:CONDition:MANual
		:SOURce:PATTern:SEQuence:CONDition:MANual?
[13]	A Pattern Match	:SOURce:PATTern:SEQuence:CONDition:AMATch
		:SOURce:PATTern:SEQuence:CONDition:AMATch?

Table 7.3.2.8-1 Sequence pattern setting commands (Cont'd)

No.	Setting Items	Commands
[14]	B Pattern Match	:SOURce:PATTern:SEQuence:CONDition:BMATch
		:SOURce:PATTern:SEQuence:CONDition:BMATch?
[15]	Data Length	:SOURce:PATTern:SEQuence:LENGth
		:SOURce:PATTern:SEQuence:LENGth?
[16]	No label (Pattern setting)	:SOURce:PATTern:SEQuence:DATA:WHOLe
		:SOURce:PATTern:SEQuence:DATA:WHOLe?
[17]	No label (Pattern setting)	:SOURce:PATTern:SEQuence:BDATa:WHOLe
		:SOURce:PATTern:SEQuence:BDATa:WHOLe?
[18]	No label (Data inversion)	:SOURce:PATTern:SEQuence:DREVerse:ADDRess
[19]	No label (Data inversion)	:SOURce:PATTern:SEQuence:DREVerse:DELTa
[20]	No label (Data inversion)	:SOURce:PATTern:SEQuence:DATA:FILL

:SOURce:PATTern:SEQuence:PAIR <numeric>

Parameter <numeric> = <DECIMAL NUMERIC PROGRAM DATA>
0 to 6 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 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.

: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.

:SOURce:PATTern:SEQuence:LOOPtime? <block>

Parameter	<block> = <DECIMAL NUMERIC PROGRAM DATA> 1 to 128 1 to 128 blocks, in 1-block steps
Response	<type> = <CHARACTER RESPONSE DATA> COUN, REP [<numeric>] = <NR1 NUMERIC RESPONSE DATA> 1 to 1024 1 to 1024 loops, in 1-loop steps ***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>]

Parameter	<block> = <DECIMAL NUMERIC PROGRAM DATA> (Block No) 1 to 128 1 to 128 blocks, in 1-block steps <loop> = <CHARACTER PROGRAM DATA> (Loop Completion) STOP, JUMP, NEXT [<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 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.

:SOURce:PATTern:SEQuence:CONDition:LOPC? <block>

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:
	`> :SOURce:PATTern:SEQuence:CONDition:LOPC? 3`
	`< NEXT`

Compatibility	Incompatible with existing models.

:SOURce:PATTern:SEQuence:CONDition:EXTernal <block>,<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:
	`> :SOURce:PATTern:SEQuence:CONDition:EXTernal? 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:
	`> :SOURce:PATTern:SEQuence:CONDition:MANual? 2`
	`< 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: `> :SOURce:PATTern:SEQuence:CONDition:AMATch 3,NONE`
Compatibility	Incompatible with existing models.

:SOURce:PATTern:SEQuence:CONDition:AMATch? <block>

Parameter	<block> = <DECIMAL NUMERIC PROGRAM DATA> (Block No) 1 to 128 1 to 128 blocks, in 1-block steps
Response	<a> = <CHARACTER RESPONSE DATA> (A 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 A match condition of each block during Sequence pattern generation.

Example	To query the pattern A match condition of block 3:
	`> :SOURce:PATTern:SEQuence:CONDition:AMATch? 3`
	`< NONE`

Compatibility	Incompatible with existing models.

:SOURce:PATTern:SEQuence:CONDition:BMATch <block>,<b>[,<destination>]

Parameter	<block> = <DECIMAL NUMERIC PROGRAM DATA> (Block No) 1 to 128 1 to 128 blocks, in 1-block steps <b> = <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.

:SOURce:PATTern:SEQuence:CONDition:BMATch? <block>

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:
	`> :SOURce:PATTern:SEQuence:CONDition:BMATch? 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.

:SOURce:PATTern:SEQuence:DATA:WHOLe <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 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.

: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 FFFFF 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 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 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.

: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 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

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 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 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:

```
> :SOURce:PATTern:SEQuence:DREVerse: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

$$\text{Max} = \frac{\text{Pattern Length}}{128} \text{ (rounding up fractions)}$$

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:

```
> :SOURce:PATTern:SEQuence:DATA:FILL 8,ALL,0,1
```

Compatibility

Incompatible with existing models.

7.3.3 Commands related to Error Addition tab

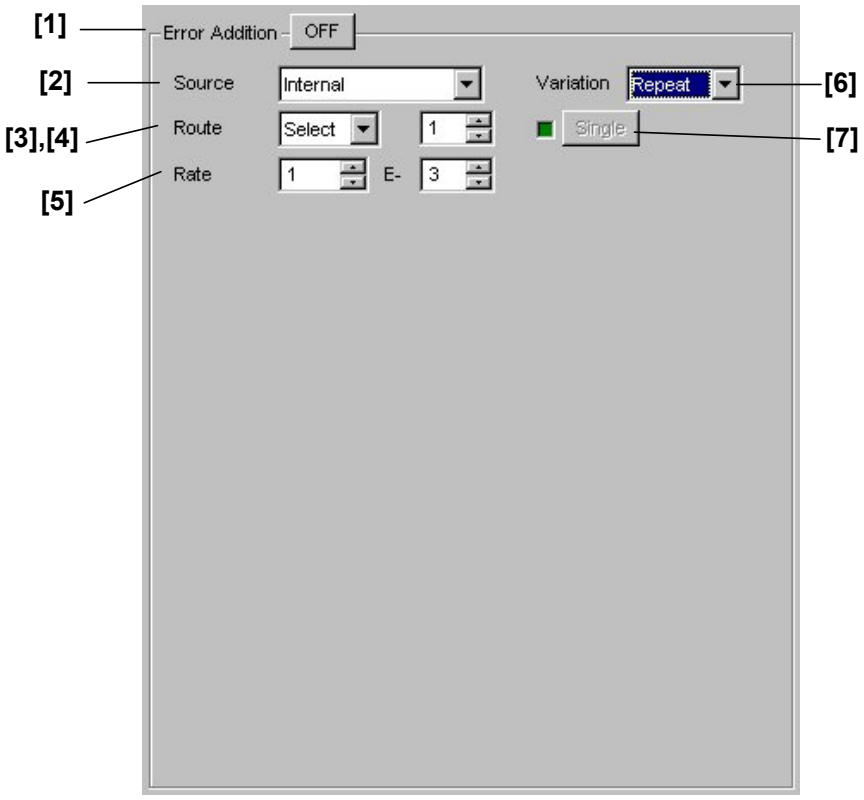


Figure 7.3.3-1 Error Addition tab
(For PRBS, Zero-Substitution, Data patterns)

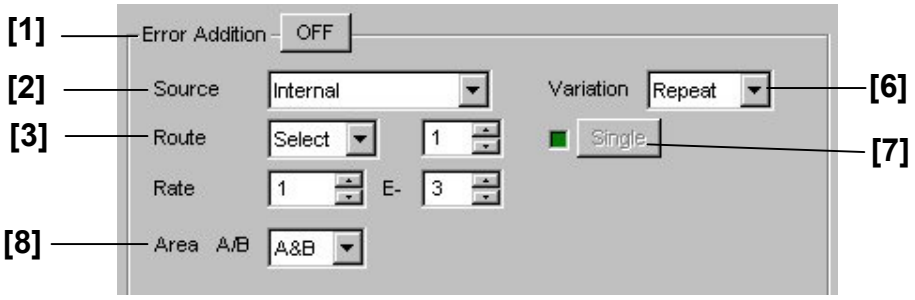


Figure 7.3.3-2 Error Addition tab (For Alternate pattern)

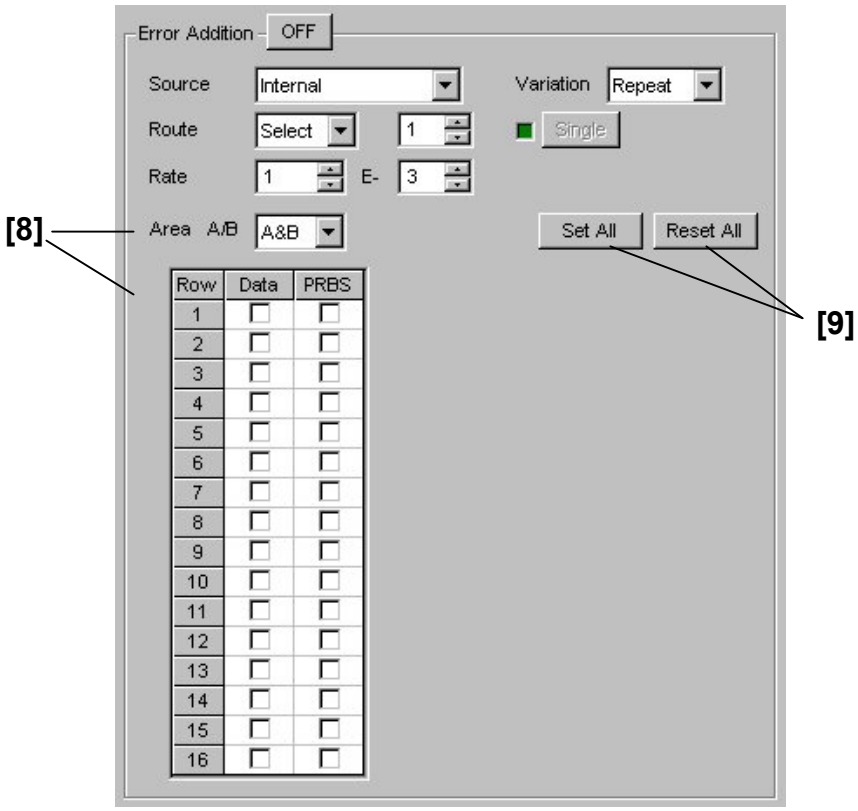


Figure 7.3.3-3 Error Addition tab (For Mixed pattern)

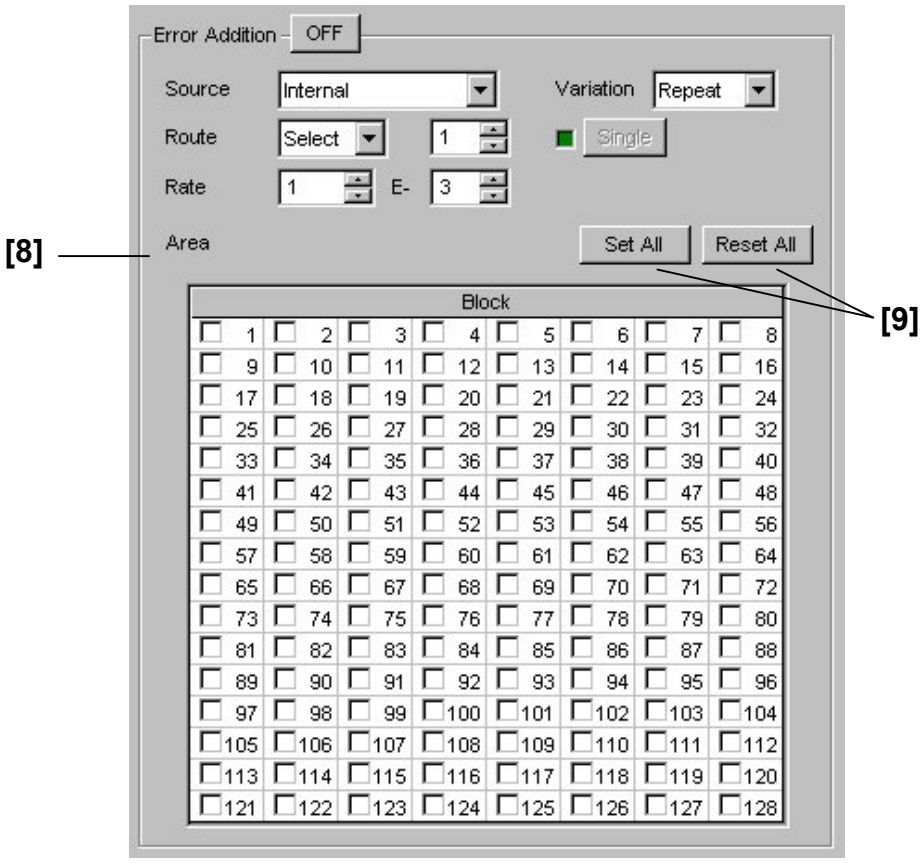


Figure 7.3.3-4 Error Addition tab (For Sequence pattern)

Table 7.3.3-1 Error Addition setting commands

No.	Setting Items	Commands
[1]	Error Addition ON/OFF	:SOURce:PATTern:EADDition:SET
		:SOURce:PATTern:EADDition:SET?
[2]	Source	:SOURce:PATTern:EADDition:SOURce
		:SOURce:PATTern:EADDition:SOURce?
[3]	Route	:SOURce:PATTern:EADDition:SELect
		:SOURce:PATTern:EADDition:SELect?
[4]	Error Addition Route setting	:SOURce:PATTern:EADDition:ROUTe
		:SOURce:PATTern:EADDition:ROUTe?
[5]	Rate	:SOURce:PATTern:EADDition:RATE
		:SOURce:PATTern:EADDition:RATE?
[6]	Variation	:SOURce:PATTern:EADDition:VARiation
		:SOURce:PATTern:EADDition:VARiation?
[7]	Single	:SOURce:PATTern:EADDition:SINGle
[8]	Area A/B, Row, Block	:SOURce:PATTern:EADDition:AREA
		:SOURce:PATTern:EADDition:AREA?
[9]	Set All/Reset All	:SOURce:PATTern:EADDition:ALLSet

: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	Compatible with the MP1632C.	

: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	Compatible with the MP1632C.	

:SOURce:PATTern:EADDition:SOURce <source>

Parameter	<source> = <CHARACTER PROGRAM DATA>	
	INTernal	Internal signal
	EXTTrig	Rising of external signal trigger
	EXTDisable	Disables 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	Incompatible with existing models.	

: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	Incompatible with existing models.

: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	Incompatible with existing models.	

: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	Incompatible with existing models.

:SOURce:PATTern:EADDition:ROUTe <numeric>

Parameter	<numeric> = <DECIMAL NUMERIC PROGRAM DATA>	
	1 to 32	1 to 32, in single steps
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	<numeric> = <NR1 NUMERIC RESPONSE DATA>	
	1 to 32	1 to 32, in single steps
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).	

: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.

: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.	

: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

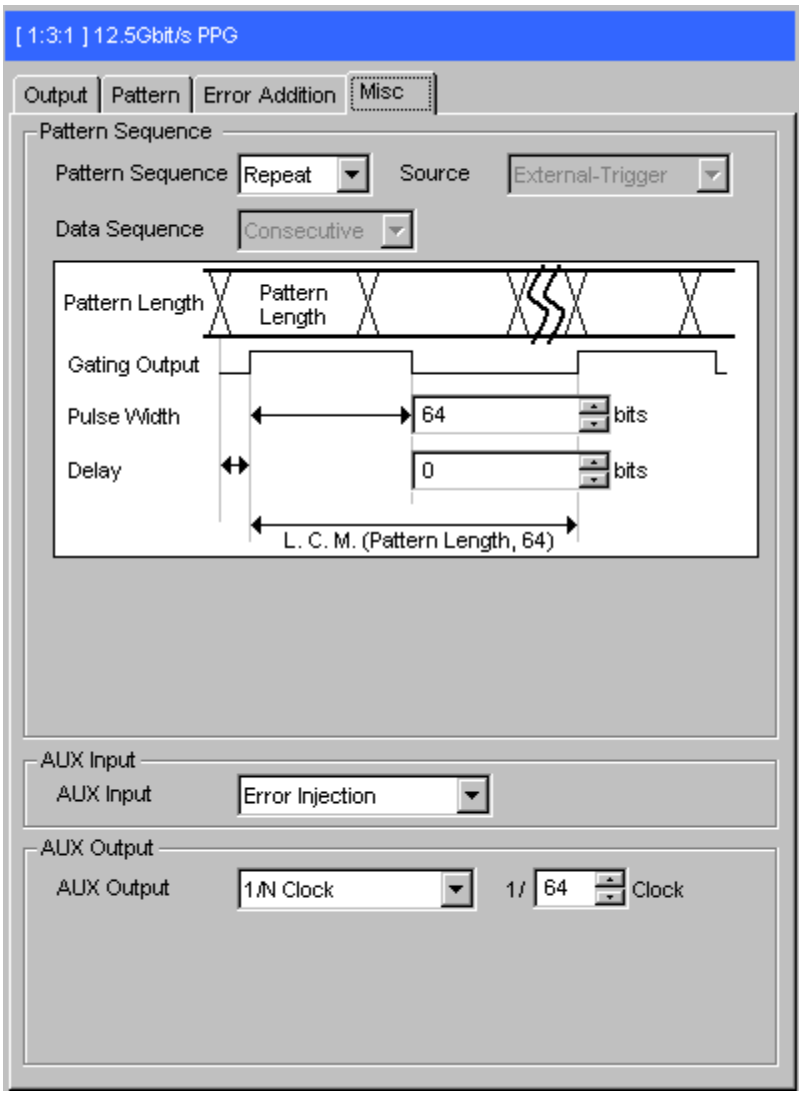


Figure 7.3.4-1 Misc tab

7.3.4.1 Pattern Generation setting commands

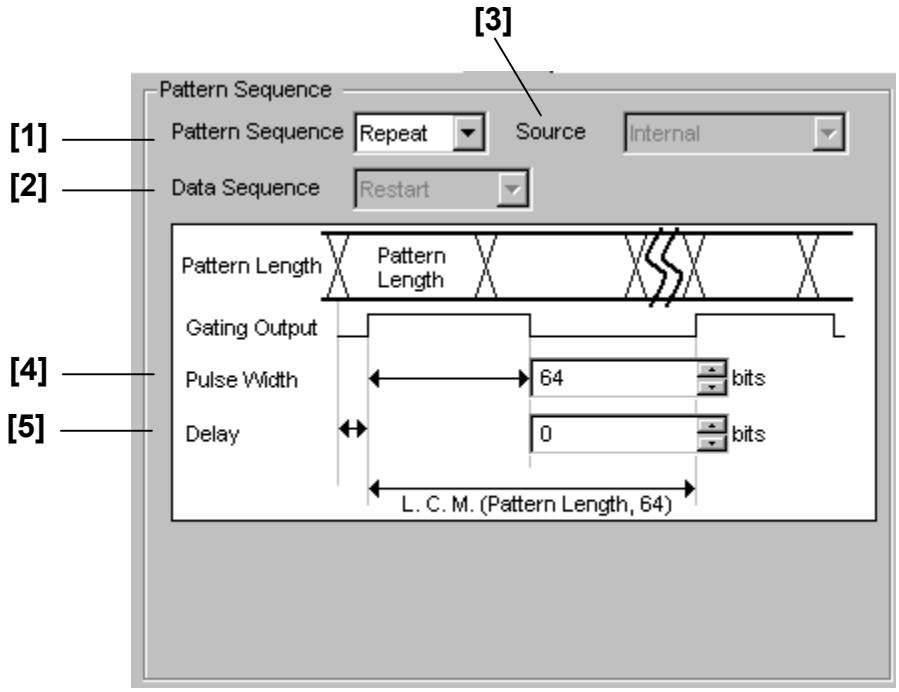


Figure 7.3.4.1-1 Pattern Generation setting field (for Repeat signal generation)

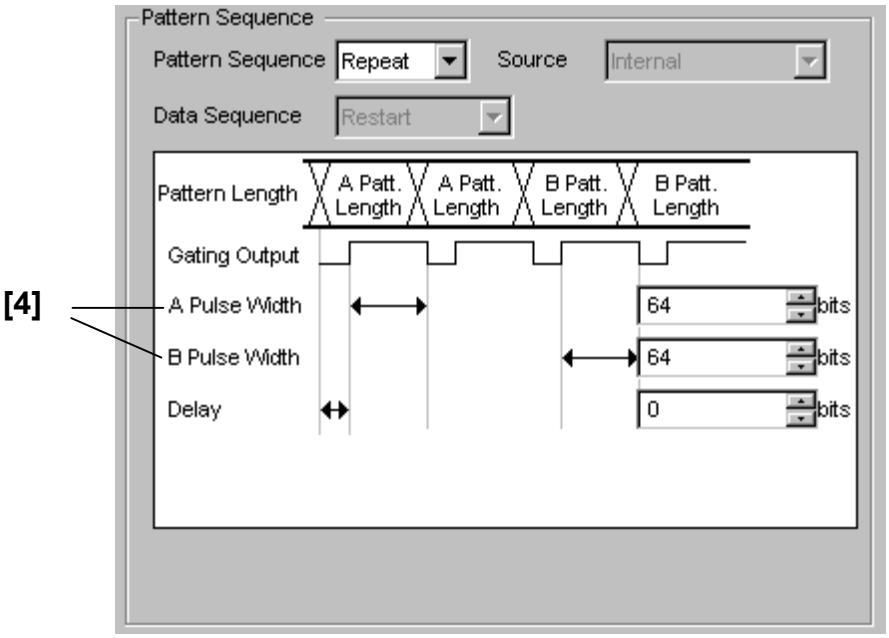


Figure 7.3.4.1-2 Pattern Generation setting field (for Repeat signal generation)

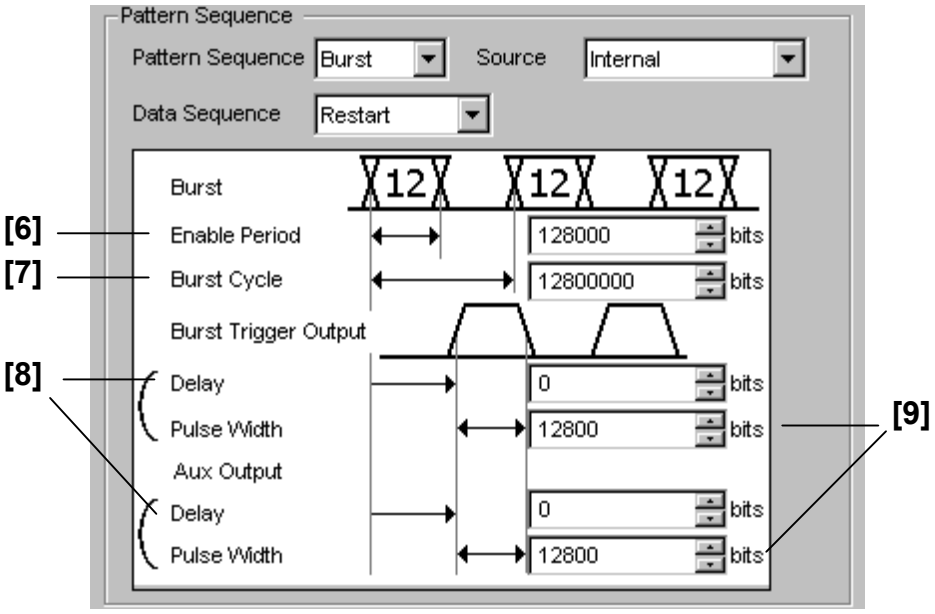


Figure 7.3.4.1-3 Pattern Generation setting field (for Burst signal generation)

Table 7.3.4.1-1 Pattern Generation setting commands

No.	Setting Items	Commands
[1]	Pattern Sequence	:SOURce:PATTern:OMODe
		:SOURce:PATTern:OMODe?
[2]	Data Sequence	:SOURce:PATTern:BURSt:SEQuence
		:SOURce:PATTern:BURSt:SEQuence?
[3]	Source	:SOURce:PATTern:BURSt:MODE
		:SOURce:PATTern:BURSt:MODE?
[4]	Pulse Width	:SOURce:PATTern:REPeat:PULSewidth
		:SOURce:PATTern:REPeat:PULSewidth?
[5]	Delay (for Repeat)	:SOURce:PATTern:REPeat:TRIGdelay
		:SOURce:PATTern:REPeat:TRIGdelay?
[6]	Enable Period	:SOURce:PATTern:BURSt:ELENgth
		:SOURce:PATTern:BURSt:ELENgth?
[7]	Burst Cycle	:SOURce:PATTern:BURSt:CYCLe
		:SOURce:PATTern:BURSt:CYCLe?
[8]	Delay (for Burst)	:SOURce:PATTern:BURSt:TRIGdelay
		:SOURce:PATTern:BURSt:TRIGdelay?
[9]	Pulse Width	:SOURce:PATTern:BURSt:PULSewidth
		:SOURce:PATTern:BURSt:PULSewidth?

: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:PATTern:BURSt:MODE <mode>

Parameter
<mode> = <CHARACTER PROGRAM DATA>

INTernal	Generates based on internal reference signal
EXTTrig	Generates based on external signal trigger edge
EXTernal	Generates during the high level interval of external signal trigger input

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
Compatible with the MP1632C.

: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
Compatible with the MP1632C.

:SOURce:PATTern: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:PATTern:REPeat:PULSewidth 128
```

To set the pulse width of the timing signal for Alternate pattern A to 256 bits:

```
> :SOURce:PATTern:REPeat:PULSewidth 256,APAT
```

Compatibility
Incompatible with existing models.

:SOURce:PATTern:REPeat:PULSewidth? [<contents>]

Parameter
[<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.

Response
<numeric> = <NR1 NUMERIC RESPONSE DATA>

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:
```
> :SOURce:PATTern:REPeat:PULSewidth?
< 128
```
To query the pulse width of the timing signal of Alternate pattern A:
```
>:SOURce:PATTern:REPeat:PULSewidth? APAT
< 256
```

Compatibility
Incompatible with existing models.

:SOURce:PATTern:REPeat:TRIGdelay <numeric>

Parameter
<numeric> = <DECIMAL NUMERIC PROGRAM DATA>
0 to 68719476672 0 to 68,719,476,672 bits, in 16-bit steps
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.

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:
```
> :SOURce:PATTern:REPeat:TRIGdelay 256
```

Compatibility
Incompatible with existing models.

:SOURce:PATTern:REPeat:TRIGdelay?

Response
<numeric> = <NR1 NUMERIC RESPONSE DATA>

Function
Queries the delay value for the timing signal pulse to be output during the timing signal period.

Example
```
> :SOURce:PATTern:REPeat:TRIGdelay?
< 256
```

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.

:SOURce:PATTern:BURSt:CYCLe?

Response	<numeric> = <NR1 NUMERIC RESPONSE DATA>
	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	`> :SOURce:PATTern:BURSt:CYCLe?`
	`< 25600`
Compatibility	Incompatible with existing models.

: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>
	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	<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 timing (delay).
Example	`> :SOURce:PATTern:BURSt:TRIGdelay? BURStout1`
	`< 128`
Compatibility	Incompatible with existing models.

: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

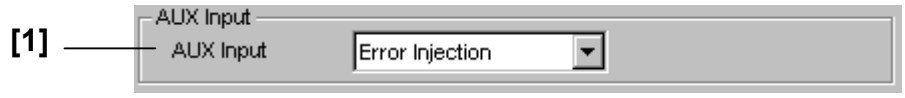


Figure 7.3.4.2-1 AUX Input setting field

Table 7.3.4.2-1 AUX Input setting commands

No.	Setting Items	Commands
[1]	AUX Input	:INPut:INPut:FUNCtion
		:INPut:INPut:FUNCtion?

:INPut:INPut:FUNCtion <function>

Parameter	<function> = <CHARACTER PROGRAM DATA>
	BURSt Burst
	ERRor Error Injection
	SEQuence Sequence Control
	ALTernate Alternate Control
	Set ERRor (error injection) in the case of Repeat.
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

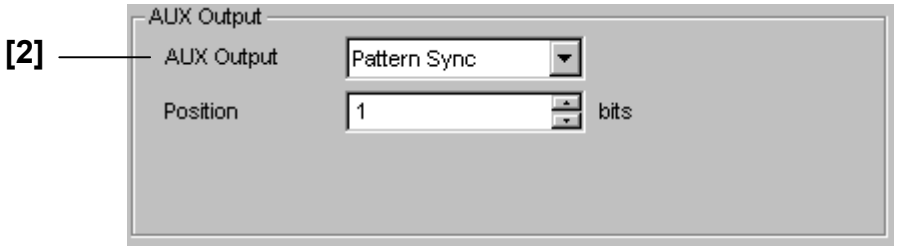


Figure 7.3.4.3-2 AUX Output setting field
(for PRBS, Zero-Substitution, and Data patterns)

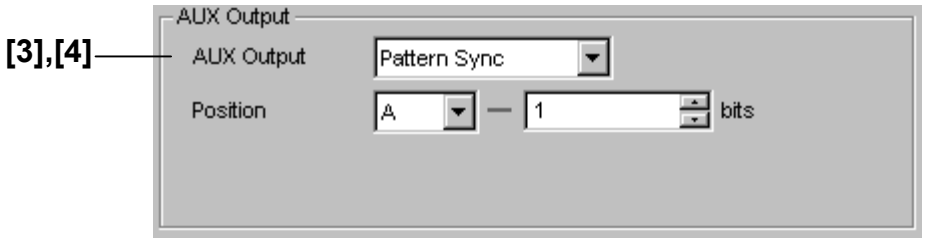


Figure 7.3.4.3-3 AUX Output setting field (For Alternate pattern)

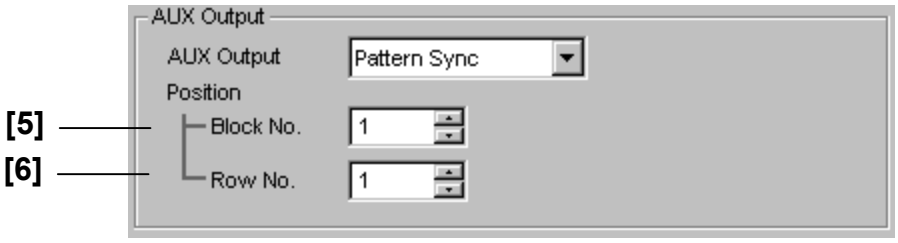


Figure 7.3.4.3-4 AUX Output setting field (For Mixed Data pattern)

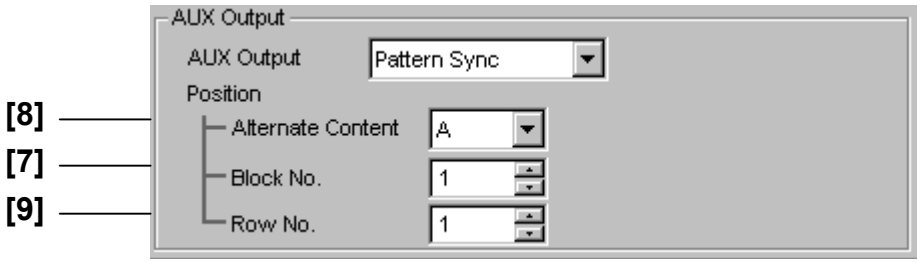


Figure 7.3.4.3-5 AUX Output setting field (For Mixed Alternate pattern)

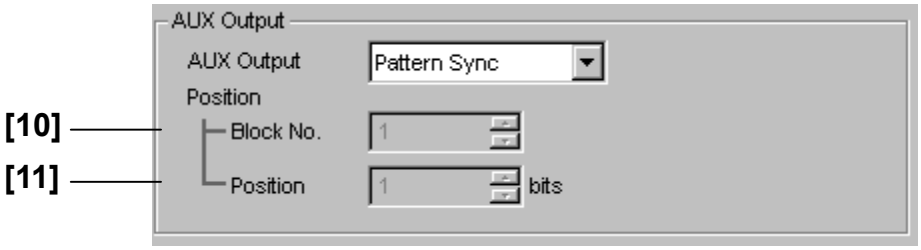


Figure 7.3.4.3-6 AUX Output setting field (For Sequence pattern)

Table 7.3.4.3-1 AUX Output setting commands

No.	Setting Items	Commands
[1]	AUX Output	:OUTPut:SYNC:SOURce
		:OUTPut:SYNC:SOURce?
[2]	Position (PRBS, Zero-Substitution, Data)	:OUTPut:SYNC:POSition
		:OUTPut:SYNC:POSition?
[3]	A/B switching (Alternate)	:OUTPut:SYNC:ALTernate:CONTents
		:OUTPut:SYNC:ALTernate:CONTents?
[4]	Position (Alternate)	:OUTPut:SYNC:ALTernate:POSition
		:OUTPut:SYNC:ALTernate:POSition?
[5]	Block No. (Mixed-Data)	:OUTPut:SYNC:MIXData:BLOCk
		:OUTPut:SYNC:MIXData:BLOCk?
[6]	Row No. (Mixed-Data)	:OUTPut:SYNC:MIXData:ROW
		:OUTPut:SYNC:MIXData:ROW?
[7]	Block No. (Mixed-Alternate)	:OUTPut:SYNC:MIXalt:BLOCk
		:OUTPut:SYNC:MIXalt:BLOCk?
[8]	Alternate Contents (Mixed-Alternate)	:OUTPut:SYNC:MIXalt:CONTents
		:OUTPut:SYNC:MIXalt:CONTents?
[9]	Row No. (Mixed-Alternate)	:OUTPut:SYNC:MIXalt:ROW
		:OUTPut:SYNC:MIXalt:ROW?
[10]	Block No. (Sequence)	:OUTPut:SYNC:SEQuence:BLOCk
		:OUTPut:SYNC:SEQuence:BLOCk?
[11]	Position (Sequence)	:OUTPut:SYNC:SEQuence:POSition
		:OUTPut:SYNC:SEQuence:POSition?

: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)
	BURSt Burst Output2
	NCLock 1/n clocks
	[<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	<source> = <CHARACTER RESPONSE DATA>
	CLOC8, CLOC16, CLOC32, CLOC64, PATT,BURS, NCL
	[<numeric>] = <DECIMAL NUMERIC RESPONSE DATA>
	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.

:OUTPut:SYNC:POSition <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, 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 274,877,906,625bits
	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.

: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 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	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 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	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 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: `> :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 The maximum setting value is the set number of 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.

: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>

Parameter	<numeric> = <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: `> :OUTPut:SYNC:SEQuence:BLOCk 128`
Compatibility	Incompatible with existing models.

:OUTPut:SYNC:SEQuence:BLOCk?

Response	<numeric> = <NR1 NUMERIC RESPONSE DATA> 1 to 128 1 to 128 blocks The maximum setting value is the set number of blocks.
Function	Queries the output block position for synchronization output for Sequence pattern.
Example	`> :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 1,048,497: `> :OUTPut:SYNC:SEQuence:POSition 1048497`
Compatibility	Incompatible with existing models.

:OUTPut:SYNC:SEQuence:POSition?

Response	<numeric> = <DECIMAL NUMERIC RESPONSE DATA> 1 to 1048497 1 to 1,048,497 bits
Function	Queries the output position for synchronization output for Sequence pattern.
Example	`> :OUTPut:SYNC:SEQuence:POSition?` `< 1048497`
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

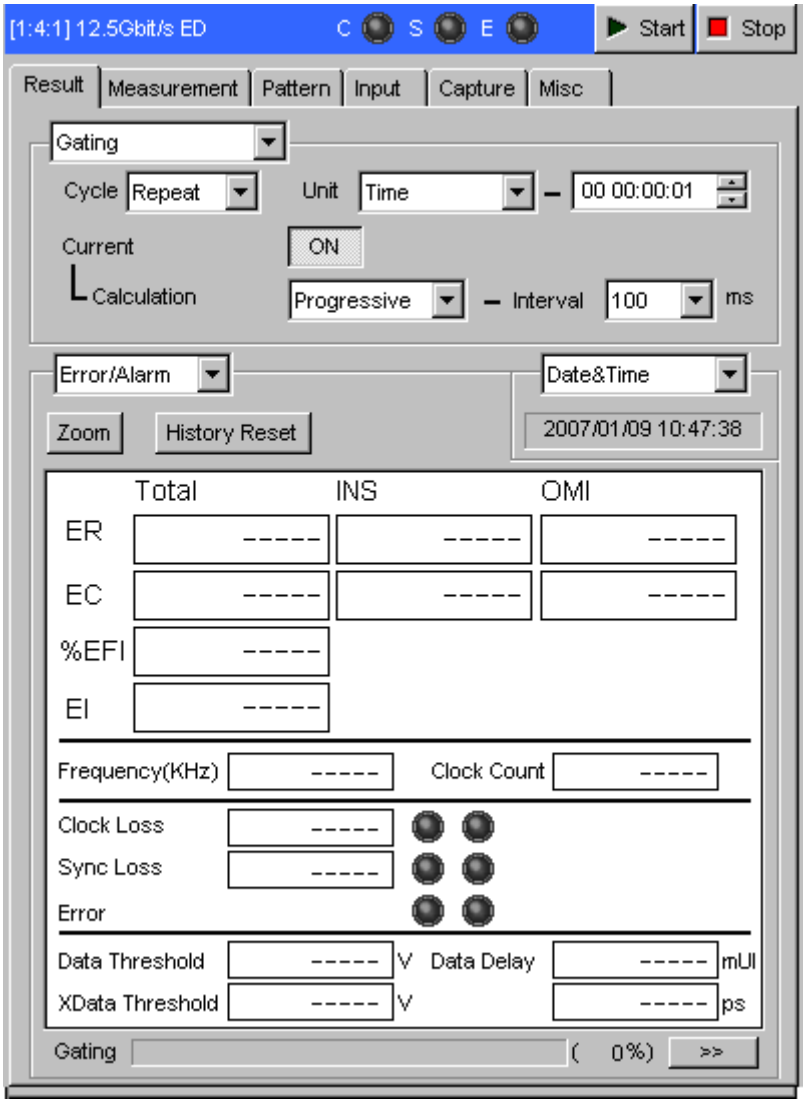


Figure 7.4.1-1 Result tab

7.4.1.1 Result setting field

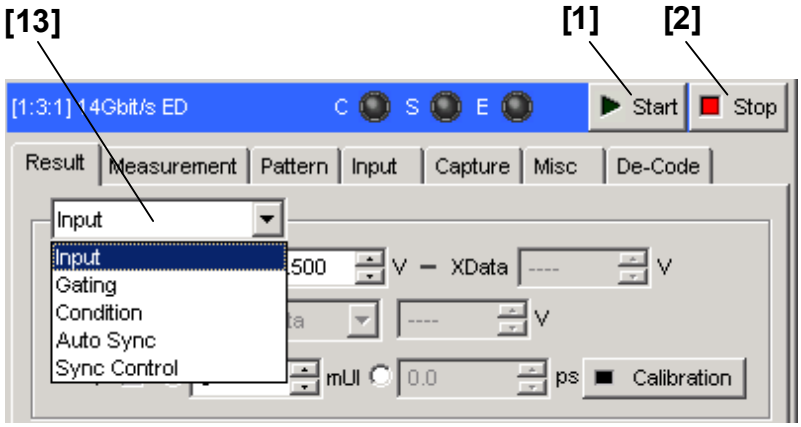


Figure 7.4.1.1-1 Start/Stop buttons

Figure 7.4.1.1-2 Result setting field

Table 7.4.1.1-1 Result setting commands

No.	Setting Items	Commands
[1]	Start	:SENSe:MEASure:STARt
[2]	Stop	:SENSe:MEASure:STOP
[3]	Measurement state query	:SENSe:MEASure:EALarm:STATe?
[4]	Display result setting	:DISPlay:RESult:SETTing
		:DISPlay:RESult:SETTing?
[5]	Display time setting	:DISPlay:RESult:TIME
		:DISPlay:RESult:TIME?
[6]	Query for measurement start time	:SENSe:MEASure:EALarm:STARt?
[7]	Query for measurement stop time	:SENSe:MEASure:EALarm:STOP?
[8]	Query for measurement elapsed time	:SENSe:MEASure:EALarm:ELAPsed?
[9]	Query for measurement remaining time	:SENSe:MEASure:EALarm:TIMed?
[10]	Query for intermediate data creation time	:SENSe:MEASure:EALarm:ITIMe?
[11]	Query for alarm occurrence time	:SENSe:MEASure:EALarm:AOCCur?
[12]	Query for alarm recovery time	:SENSe:MEASure:EALarm:ARECover?
[13]	Display measurement items	:SENSe:DISPlay:SETTing

: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.

: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.		

:SENSe:MEASure:EALarm:STOP?

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 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 <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 the MP1632C and MP1776A.

: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 the MP1632C and MP1776A.

:SENSe:MEASure:EALarm:ITIMe?

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 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 the MP1632C and MP1776A.

:SENSe:MEASure:EALarm:AOCCur? <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	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

Function
Queries the last time when a specified alarm has occurred during bit error alarm measurement.
"0,0,0,0,0,0" is returned if no alarm has occurred.

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: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	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 ours
	<min>	0 to 59	0 to 59 minutes
	<second>	0 to 59	0 to 59 seconds
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

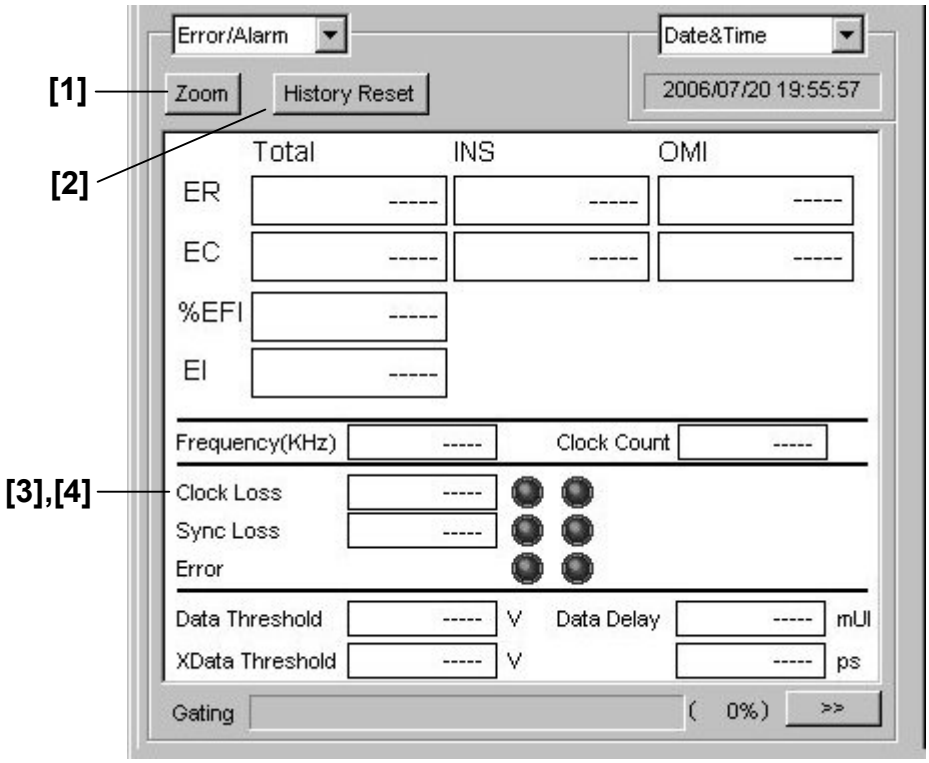


Figure 7.4.1.2-1 Error/Alarm result display screen

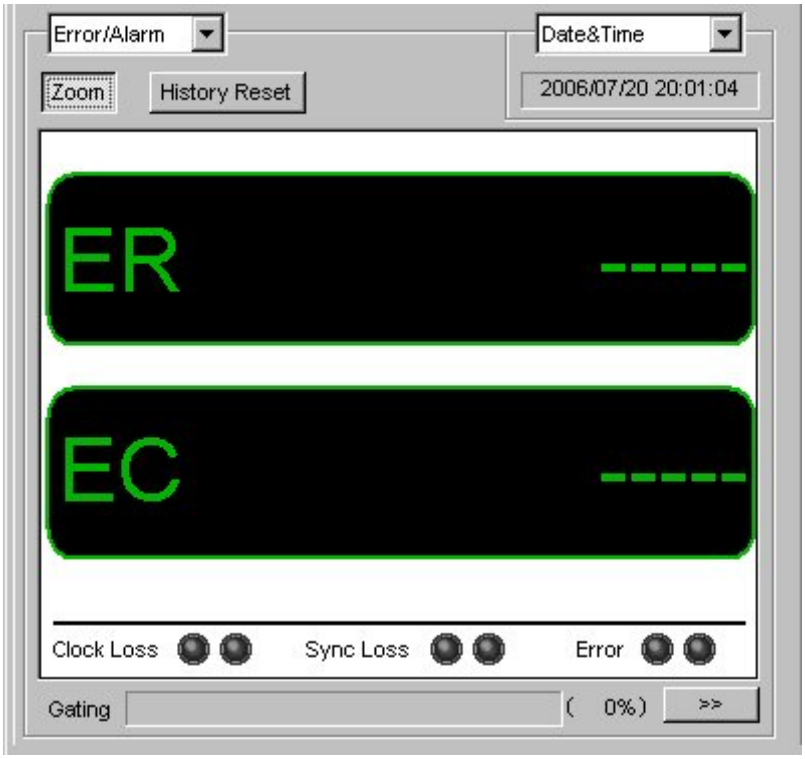


Figure 7.4.1.2-2 Error/Alarm result display screen (when Zoom is selected)

Table 7.4.1.2-1 Error/Alarm setting commands

No.	Setting Items	Commands
[1]	Zoom	:DISPlay:RESult:ZOOM
		:DISPlay:RESult:ZOOM?
[2]	History Reset	:DISPlay:RESult:EALarm:HRESet
[3]	No label (Query for error alarm measurement data)	:CALCulate:DATA:EALarm?
[4]	No label (Query for monitoring item)	:CALCulate:DATA:MONitor?

: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.

: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

Form	Format	Description
Form1 Integer type	"XXXXXXX"	For 0 to 9999999
	"X.XXXXEXX"	For 1.0000E07 to 9.9999E17
	"-------"	When no data corresponds to the query
Form2 Fraction type	"X.XXXXE-XX"	For 0.0001E–18 to 1.0000E00
	"-------"	When no data corresponds to the query
Form3 % type	"XXX.XXXXX"	For 0.0000 to 100.0000
	"-------"	When no data corresponds to the query
Form4 Frequency type	"XXXXXXXXX"	For 0 to MAX (Hz)
	"-------"	When no data corresponds to the query
Form5 Voltage type	"X.XXX,XX.XXX"	For –4.000 to +4.000
	"-------"	When no data corresponds to the query
Form6 Phase type	"XXXX,XXXX"	For –1000 to +1000, –10000 to +10000
	"-------"	When no data corresponds to the query

Function

Queries the measurement data that corresponds to the parameter.

Table 7.4.1.2-3 Parameters

Items		<result>	Form
Error Rate	Total	ER:TOTal	Form2
	INS	ER:INSertion	
	OMI	ER:OMISsion	
	Transition	ER:TRANsition	
	Non Transition	ER:NONTransition	
Error Count	Total	EC:TOTal	Form1
	INS	EC:INSertion	
	OMI	EC:OMISsion	
	Transition	EC:TRANsition	
	Non Transition	EC:NONTransition	
%EFI	Total	EFI:TOTal	Form3
EI	Total	EI:TOTal	Form1
Frequency (kHz)		FREQuency	Form4
Clock Count		CC:TOTal	Form1
Clock Loss		AINTerval:CLOSs	Form1

Table 7.4.1.2-3 Parameters (Cont'd)

Items		<result>	Form
CR Unlock		AINTerval:CRUNlock	Form1
Sync Loss		AINTerval:PSLoss	Form1
Data Threshold		THReshold	Form5
Data Phase		PHASe	Form6
Threshold EI	>1.0E–3	THReshold:EI:TOTal:E_3	Form1
	>1.0E–4	THReshold:EI:TOTal:E_4	
	>1.0E–5	THReshold:EI:TOTal:E_5	
	>1.0E–6	THReshold:EI:TOTal:E_6	
	>1.0E–7	THReshold:EI:TOTal:E_7	
	>1.0E–8	THReshold:EI:TOTal:E_8	
	≤1.0E–8	THReshold:EI:TOTal:UE_8	
Threshold%EFI	>1.0E–3	THReshold:EFI:TOTal:E_3	Form3
	>1.0E–4	THReshold:EFI:TOTal:E_4	
	>1.0E–5	THReshold:EFI:TOTal:E_5	
	>1.0E–6	THReshold:EFI:TOTal:E_6	
	>1.0E–7	THReshold:EFI:TOTal:E_7	
	>1.0E–8	THReshold:EFI:TOTal:E_8	
	≤1.0E–8	THReshold:EFI:TOTal:UE_8	
G.821	ES	G821:ES2:TOTal	Form1
	EFS	G821:EFS2:TOTal	
	SES	G821:SES2:TOTal	
	DM	G821:DM2:TOTal	
	US	G821:US2:TOTal	
	%ES	G821:ES:TOTal	Form3
	%EFS	G821:EFS:TOTal	
	%SES	G821:SES:TOTal	
	%DM	G821:DM:TOTal	
	%US	G821:US:TOTal	

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.

:CALCulate:DATA:MONitor? <item>

Parameter

<item> = <STRING PROGRAM DATA>

"BIT:TOTal"	Bit Error (Total Error)
"CLOSs"	Clock Loss
"PSLoss"	Pattern Sync Loss
"CRUNlock"	CR Unlock

Response

<string> = <STRING RESPONSE DATA>

Table 7.4.1.2-4 Response

Format	Description
"Occur"	When an alarm occurs
"Not Occur"	When no alarm occurs
"-------"	When no data corresponds to the query

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

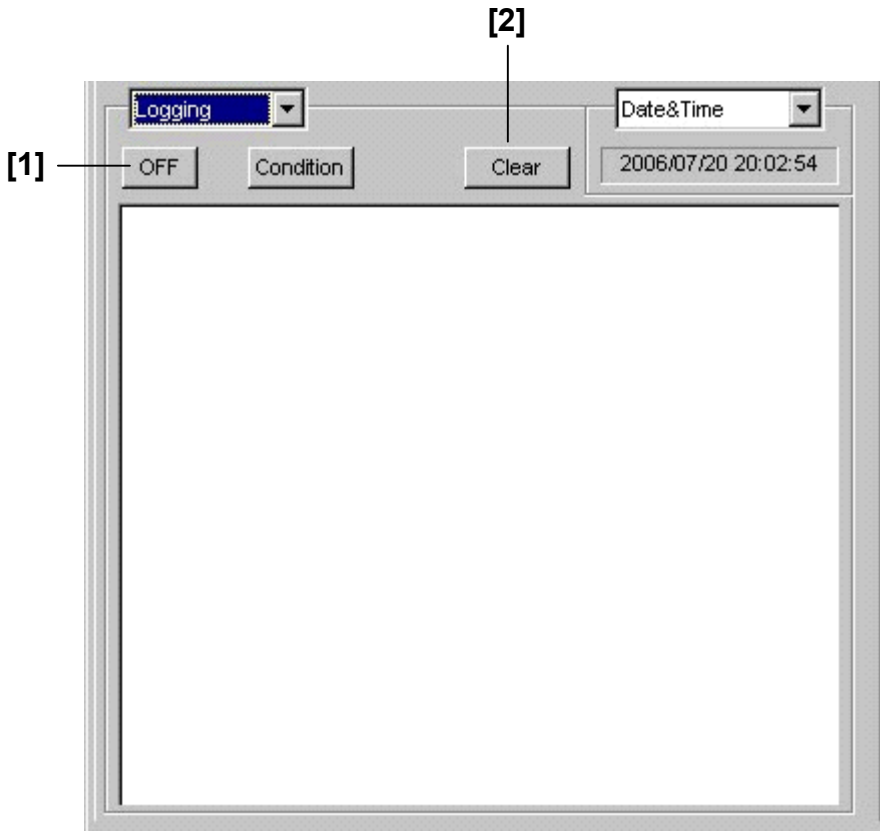


Figure 7.4.1.3-1 Logging setting field

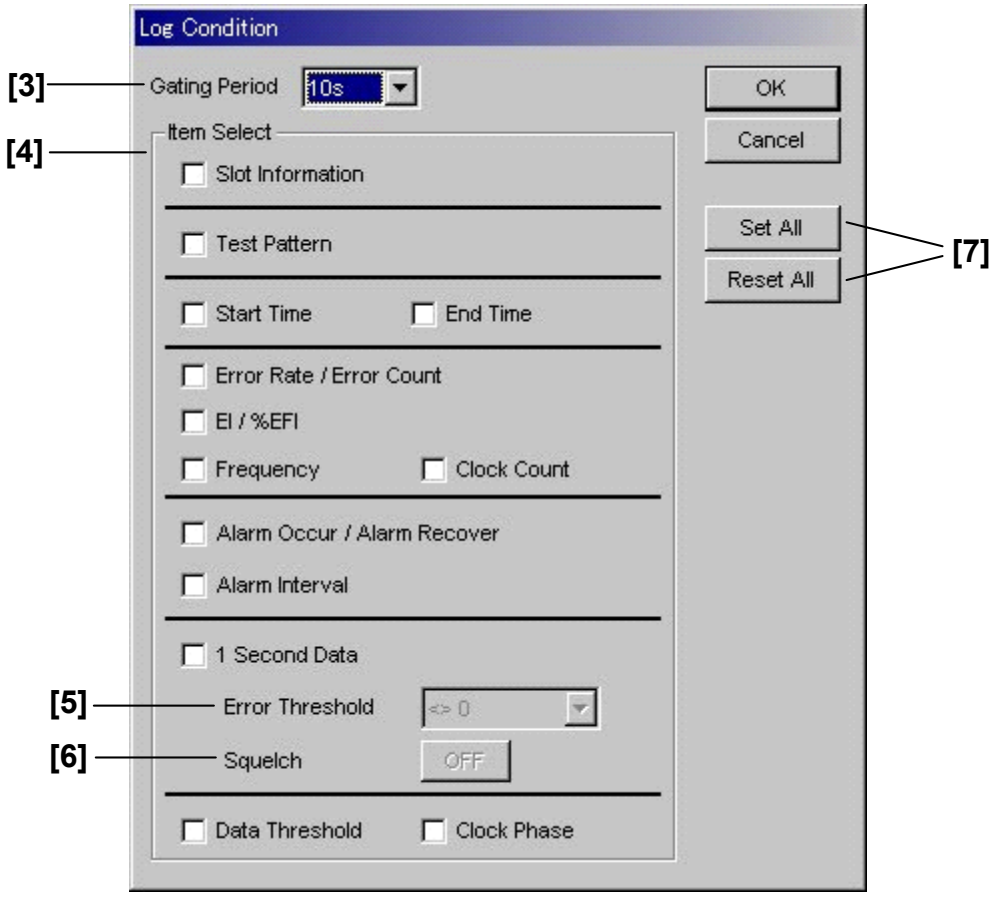


Figure 7.4.1.3-2 Log Condition dialog box

Table 7.4.1.3-1 Logging setting commands

No.	Setting Items	Commands
[1]	Logging ON/OFF	:DISPlay:RESult:LOGGing:ONOFf
		:DISPlay:RESult:LOGGing:ONOFf?
[2]	Clear	:DISPlay:RESult:LOGGing:CLEar
[3]	Gating Period	:DISPlay:RESult:LOGGing:CONDition:PERiod
		:DISPlay:RESult:LOGGing:CONDition:PERiod?
[4]	Item Select	:DISPlay:RESult:LOGGing:CONDition:ITEM
		:DISPlay:RESult:LOGGing:CONDition:ITEM?
[5]	Error Threshold	:DISPlay:RESult:LOGGing:CONDition:THReshold
		:DISPlay:RESult:LOGGing:CONDition:THReshold?
[6]	Squelch	:DISPlay:RESult:LOGGing:CONDition:SQUelch
		:DISPlay:RESult:LOGGing:CONDition:SQUelch?
[7]	Set All/Reset All	:DISPlay:RESult:LOGGing:CONDition:ALLSet

:DISPlay:RESult:LOGGing:ONOFf <boolean>

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

Response	<numeric> = <NR1 NUMERIC RESPONSE DATA>	
	0	OFF
	1	ON
Function	Queries the logging execution ON/OFF state.	
Example	`> :DISPlay:RESult:LOGGing:ONOFf?`	
	`< 1`	
Compatibility	Incompatible with existing models.	

:DISPlay:RESult:LOGGing:CLEar

Function	Clears the logging results.
Example	`> :DISPlay:RESult:LOGGing:CLEar`
Compatibility	Incompatible with existing models.

:DISPlay:RESult:LOGGing:CONDition:PERiod <period>

Parameter	<period> = <CHARACTER PROGRAM DATA>	
	S10	10 s
	S30	30 s
	M1	1 min
	M10	10 min
	M30	30 min
	H1	1 hour
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.	

: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>

Parameter	<item> = <CHARACTER PROGRAM DATA> SLOTinfo, TPATtern, STARt, END, ERATe, EI, FREQuency, CCOunt, AOCCur, AINTerval, DTHReshold, CDELay, SDATa
Response	<numeric> = <NR1 NUMERIC RESPONSE DATA> 0 Measurement item OFF 1 Measurement item ON
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>

Parameter	<thre> = <CHARACTER PROGRAM DATA> E_0 <>0 E_3 >1E–3 E_4 >1E–4 E_5 >1E–5 E_6 >1E–6 E_7 >1E–7 E_8 >1E–8 E_9 >1E–9
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?

Response	<thre> = <CHARACTER RESPONSE DATA> E_0, E_3, E_4, E_5, E_6, E_7,E_8,E_9
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>

Parameter	<boolean> = <BOOLEAN PROGRAM DATA>	
	0 or OFF	Logging memory squelch function OFF
	1 or ON	Logging memory squelch function ON
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?

Response	<numeric> = <NR1 NUMERIC RESPONSE DATA>	
	0	Logging memory squelch function OFF
	1	Logging memory squelch function ON
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>

Parameter	<boolean> = <BOOLEAN PROGRAM DATA>	
	0 or OFF	All reset
	1 or ON	All Set
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.	

7.4.1.4 Histogram setting commands

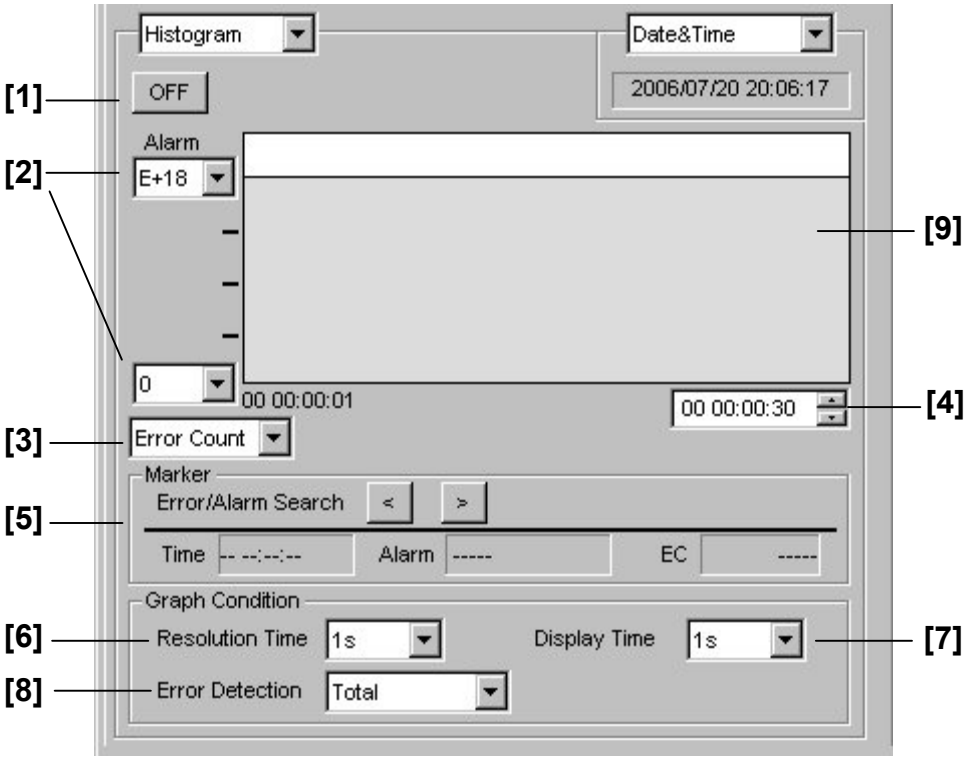


Figure 7.4.1.4-1 Histogram setting field

Table 7.4.1.4-1 Histogram setting commands

No.	Setting Items	Commands
[1]	Histogram ON/OFF	:DISPlay:RESult:HISTogram:ONOFf
		:DISPlay:RESult:HISTogram:ONOFf?
[2]	Scale setting	:DISPlay:RESult:HISTogram:SCALe
		:DISPlay:RESult:HISTogram:SCALe?
[3]	Measurement item setting	:DISPlay:RESult:HISTogram:TYPE
		:DISPlay:RESult:HISTogram:TYPE?
[4]	Display time setting	:DISPlay:RESult:HISTogram:TIMescale
		:DISPlay:RESult:HISTogram:TIMescale?
[5]	Marker	:DISPlay:RESult:HISTogram:MARKer
		:DISPlay:RESult:HISTogram:MARKer?
[6]	Resolution Time	:DISPlay:RESult:HISTogram:RESolution
		:DISPlay:RESult:HISTogram:RESolution?
[7]	Display Time	:DISPlay:RESult:HISTogram:DISPtime
		:DISPlay:RESult:HISTogram:DISPtime?
[8]	Error Detection	:DISPlay:RESult:HISTogram:EDETect
		:DISPlay:RESult:HISTogram:EDETect?
[9]	Query for measurement result	:CALCulate:DATA:HISTogram?

: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?

Response	<top> = <CHARACTER RESPONSE DATA> <bottom> = <CHARACTER RESPONSE DATA>
Function	Queries the vertical axis scale of the histogram.
Example	> :DISPlay:RESult:HISTogram:SCALe? < E_3.E_4
Compatibility	Incompatible with existing models.

:DISPlay:RESult:HISTogram:TYPE <type>

Parameter	<type> = <CHARACTER PROGRAM DATA>	
	ECOunt	Counts errors.
	ERATe	Counts error rate.
	EI	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 ECOunt (counting errors). > :DISPlay:RESult:HISTogram:TYPE ECOunt	
Compatibility	Incompatible with existing models.	

:DISPlay:RESult:HISTogram:TYPE?

Response	<type> = <CHARACTER RESPONSE DATA> ECO, ERAT, EI
Function	Queries the measurement item counting type displayed in the histogram.
Example	> :DISPlay:RESult:HISTogram:TYPE? < ECO
Compatibility	Incompatible with existing models.

: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.	

:DISPlay:RESult:HISTogram:MARKer?

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
"-" is displayed if there is no alarm and error.
<alarm> = <CHARACTER RESPONSE DATA>
CLOS Clock LOS has occurred.
SLOS Sync LOS has occurred.
CRUN CR Unlock has occurred.
NONE No alarm has occurred.
<result> = <STRING RESPONSE DATA>
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 7.4.1.4-2 Response Format

Form	Format	Description
Form1 Integer type	"XXXXXXX"	For 0 to 9999999
	"X.XXXXEXX"	For 0.0000E07 to 9.9999E17
	"-------"	When no data corresponds to the query
Form2 Fraction type	"X.XXXXE-XX"	For 0.0001E–18 to 1.0000E00
	"-------"	When no data corresponds to the query

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.

:DISPlay:RESult:HISTogram:RESolution <resolution>

Parameter	<resolution> = <CHARACTER PROGRAM DATA>	
	S1	1 s
	S10	10 s
	S30	30 s
	M1	1 min
	M10	10 min
	M30	30 min
	H1	1 hour
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?

Response	<resolution> = <CHARACTER RESPONSE DATA>
	S1, S10, S30, M1, M10, M30, H1
Function	Queries the histogram calculation resolution.
Example	`> :DISPlay:RESult:HISTogram:RESolution?`
	`< S1`
Compatibility	Incompatible with existing models.

:DISPlay:RESult:HISTogram:DISPtime <time>

Parameter	<time> = <CHARACTER PROGRAM DATA>	
Parameter	<resolution> = <CHARACTER PROGRAM DATA>	
	S1	1 s
	S10	10 s
	S30	30 s
	M1	1 min
	M10	10 min
	M30	30 min
	H1	1 hour
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?

Response	<time> = <CHARACTER RESPONSE DATA> S1, S10, S30, M1, M10, M30, H1
Function	Queries the histogram display resolution.
Example	`> :DISPlay:RESult:HISTogram:DISPtime?` `< S1`
Compatibility	Incompatible with existing models.

:DISPlay:RESult:HISTogram:EDETect <mode>

Parameter	<mode> = <CHARACTER PROGRAM DATA>	
	TOTal	Total Error
	INS	Insertion Error
	OMI	Omission Error
	TRANsition	Transition Error
	NONTransition	Non Transition Error
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: `> :DISPlay:RESult:HISTogram:EDETect TOTal`	
Compatibility	Incompatible with existing models.	

:DISPlay:RESult:HISTogram:EDETect?

Response	<mode> = <CHARACTER RESPONSE DATA> TOT, INS, OMI, TRAN, NONT
Function	Queries the bit error alarm measurement method and error detection method for histogram calculation.
Example	`> :DISPlay:RESult:HISTogram:EDETect?` `< TOT`
Compatibility	Incompatible with existing models.

: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 7.4.1.4-3 Response Format

	Format	Description
<count>	"XXXXXXX"	For 0 to 9999999
	"X.XXXXEXX"	For 1.0000E07 to 9.9999E17
	"-------"	When no data corresponds to the query
<rate>	"X.XXXXE-XX"	For 0.0001E–18 to 1.0000E00
	"-------"	When no data corresponds to the query

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

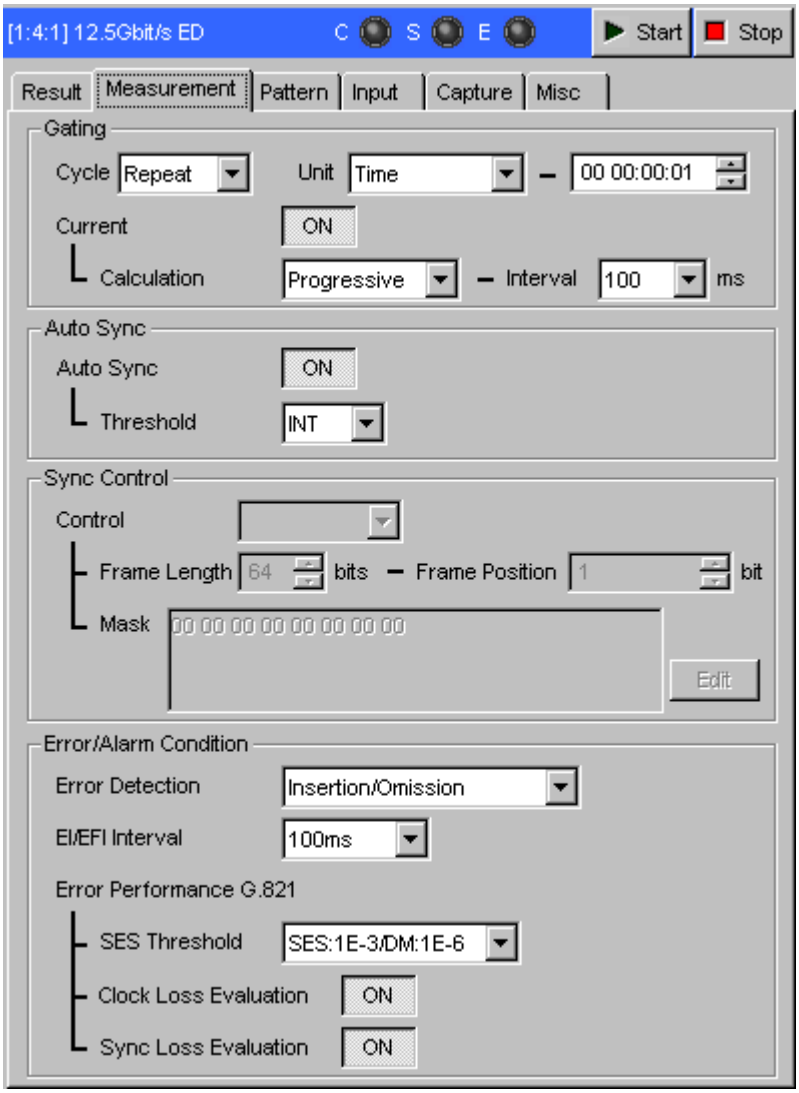


Figure 7.4.2-1 Measurement tab

7.4.2.1 Gating setting commands

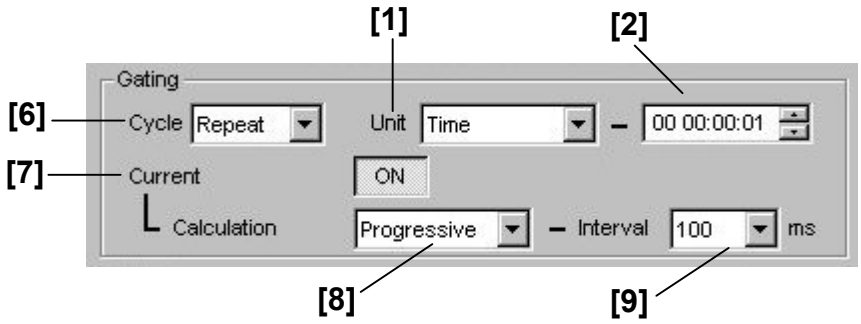


Figure 7.4.2.1-1 Gating setting (Time)

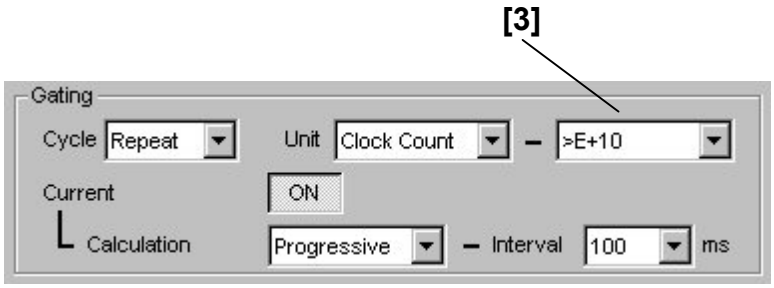


Figure 7.4.2.1-2 Gating setting (Clock Count)

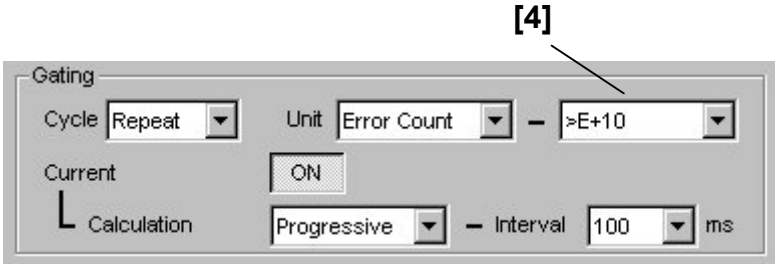


Figure 7.4.2.1-3 Gating setting (Error Count)

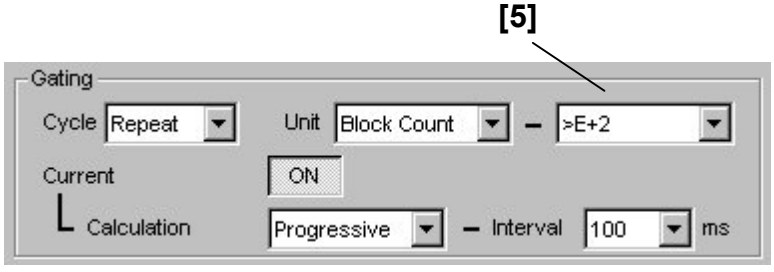


Figure 7.4.2.1-4 Gating setting (Block Count)

Table 7.4.2.1-1 Gating setting commands

No.	Setting Items	Commands
[1]	Unit	:SENSe:MEASure:EALarm:UNIT
		:SENSe:MEASure:EALarm:UNIT?
[2]	No label (Measurement time setting)	:SENSe:MEASure:EALarm:PERiod
		:SENSe:MEASure:EALarm:PERiod?
[3]	No label (Clock count setting)	:SENSe:MEASure:EALarm:CLOCkcnt
		:SENSe:MEASure:EALarm:CLOCkcnt?
[4]	No label (Error count setting)	:SENSe:MEASure:EALarm:ERRorcnt
		:SENSe:MEASure:EALarm:ERRorcnt?
[5]	No label (Block count setting)	:SENSe:MEASure:EALarm:BLOCkcnt
		:SENSe:MEASure:EALarm:BLOCkcnt?
[6]	Cycle	:SENSe:MEASure:EALarm:MODE
		:SENSe:MEASure:EALarm:MODE?
[7]	Current	:DISPlay:RESult:EALarm:MODE
		:DISPlay:RESult:EALarm:MODE?
[8]	Calculation	:DISPlay:RESult:EALarm:CALCuration
		:DISPlay:RESult:EALarm:CALCuration?
[9]	Interval	:DISPlay:RESult:EALarm:INTerval
		:DISPlay:RESult:EALarm:INTerval?

: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>

Parameter	<day>,<hour>,<min>,<second> = <DECIMAL NUMERIC PROGRAM DATA>		
	<day>	0 to 99	0 to 99 days, in 1-day steps
	<hour>	0 to 23	0 to 23 hours, in 1-hour steps
	<min>	0 to 59	0 to 59 minutes, in 1-minute steps
	<second>	0 to 59	0 to 59 seconds, in 1-second steps
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?

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 period is set to Time.		
Example	`> :SENSe:MEASure:EALarm:PERiod?` `< 0,0,1,0`		
Compatibility	Compatible with the MP1632C and MP1776A.		

:SENSe:MEASure:EALarm:CLOCkcnt <count>

Parameter	<count> = <CHARACTER PROGRAM DATA>	
	E_4	>1E+4
	E_5	>1E+5
	E_6	>1E+6
	E_7	>1E+7
	E_8	>1E+8
	E_9	>1E+9
	E_10	>1E+10
	E_11	>1E+11
	E_12	>1E+12
	E_13	>1E+13
	E_14	>1E+14
	E_15	>1E+15
	E_16	>1E+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 1E+4:	
	`> :SENSe:MEASure:EALarm:CLOCkcnt E_4`	
Compatibility	Incompatible with existing models.	

:SENSe:MEASure:EALarm:CLOCkcnt?

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 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>

Parameter	<count> = <CHARACTER PROGRAM DATA>	
	E_4	>1E+4
	E_5	>1E+5
	E_6	>1E+6
	E_7	>1E+7
	E_8	>1E+8
	E_9	>1E+9
	E_10	>1E+10
	E_11	>1E+11
	E_12	>1E+12
	E_13	>1E+13
	E_14	>1E+14
	E_15	>1E+15
	E_16	>1E+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 1E+4:	
	`> :SENSe:MEASure:EALarm:ERRorcnt E_4`	
Compatibility	Incompatible with existing models.	

: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 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	<count> = <CHARACTER PROGRAM DATA>	
	E_2	>1E+2
	E_3	>1E+3
	E_4	>1E+4
	E_5	>1E+5
	E_6	>1E+6
	E_7	>1E+7
	E_8	>1E+8
	E_9	>1E+9
	E_10	>1E+10
	E_11	>1E+11
	E_12	>1E+12
	E_13	>1E+13
	E_14	>1E+14
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	<count> = <CHARACTER RESPONSE DATA>
	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 block count threshold when the measurement period is set to Block Count.
Example	`> :SENSe:MEASure:EALarm:BLOCkcnt?`
	`< E_4`
Compatibility	Incompatible with existing models.

: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>

Parameter	<mode> = <CHARACTER PROGRAM DATA> PROGressive Progressive accumulation IMMediate Immediate processing
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?

Response	<mode> = <CHARACTER RESPONSE DATA> PROG, IMM
Function	Queries the immediate data calculation method.
Example	`> :DISPlay:RESult:EALarm:CALCuration?` `< PROG`
Compatibility	Incompatible with existing models.

:DISPlay:RESult:EALarm:INTerval <interval>

Parameter	<interval> = <CHARACTER PROGRAM DATA> I100 Updates display in 100-ms period. I200 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: `> :DISPlay:RESult:EALarm:INTerval I100`
Compatibility	Incompatible with existing models.

:DISPlay:RESult:EALarm:INTerval?

Response	<interval> = <CHARACTER RESPONSE DATA> I100, I200
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

Table 7.4.2.2-1 Auto Sync setting commands

No.	Setting Items	Commands
[1]	Auto Sync	:SENSe:PATTern:SYNC:ASYNc
		:SENSe:PATTern:SYNC:ASYNc?
[2]	Threshold	:SENSe:PATTern:SYNC:THReshold
		:SENSe:PATTern:SYNC:THReshold?

: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 to automatically perform resynchronization (Auto Sync) when the synchronization threshold is exceeded.
Example	To enable automatic resynchronization (Auto Sync ON): `> :SENSe:PATTern:SYNC:ASYNc ON`
Compatibility	Compatible with the MP1632C and MP1776A.

:SENSe:PATTern:SYNC:ASYNc?

Response	<numeric> = <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	`> :SENSe:PATTern:SYNC:ASYNc?` `< 1`
Compatibility	Compatible with the MP1632C and MP1776A.

:SENSe:PATTern:SYNC:THReshold <thre>

Parameter	<thre> = <CHARACTER PROGRAM DATA>	
	INT	Internal
	E_2	E–2
	E_3	E–3
	E_4	E–4
	E_5	E–5
	E_6	E–6
	E_7	E–7
	E_8	E–8
Function	Sets the synchronization detection threshold for resynchronization.	
Example	To set the synchronization detection threshold for resynchronization to Internal: `> :SENSe:PATTern:SYNC:THReshold INT`	
Compatibility	Compatible with the MP1776A.	

:SENSe:PATTern:SYNC:THReshold?

Response	<thre> = <CHARACTER RESPONSE DATA> INT, E_2, E_3, E_4, E_5, E_6, E_7, E_8
Function	Queries the synchronization detection threshold for resynchronization.
Example	`> :SENSe:PATTern:SYNC:THReshold?` `< INT`
Compatibility	Compatible with the MP1776A.

7.4.2.3 Sync Control setting commands

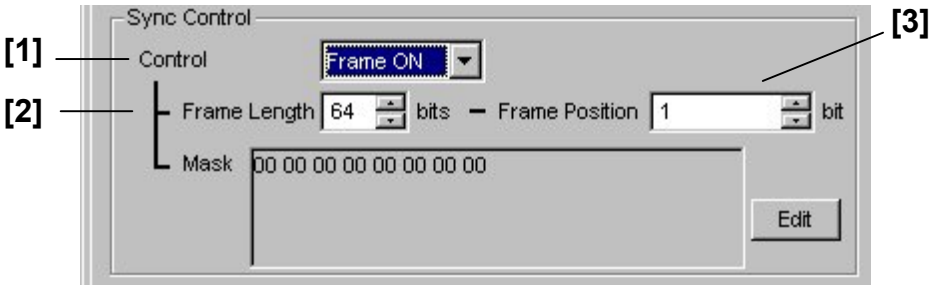


Figure 7.4.2.3-1 Sync Control setting field

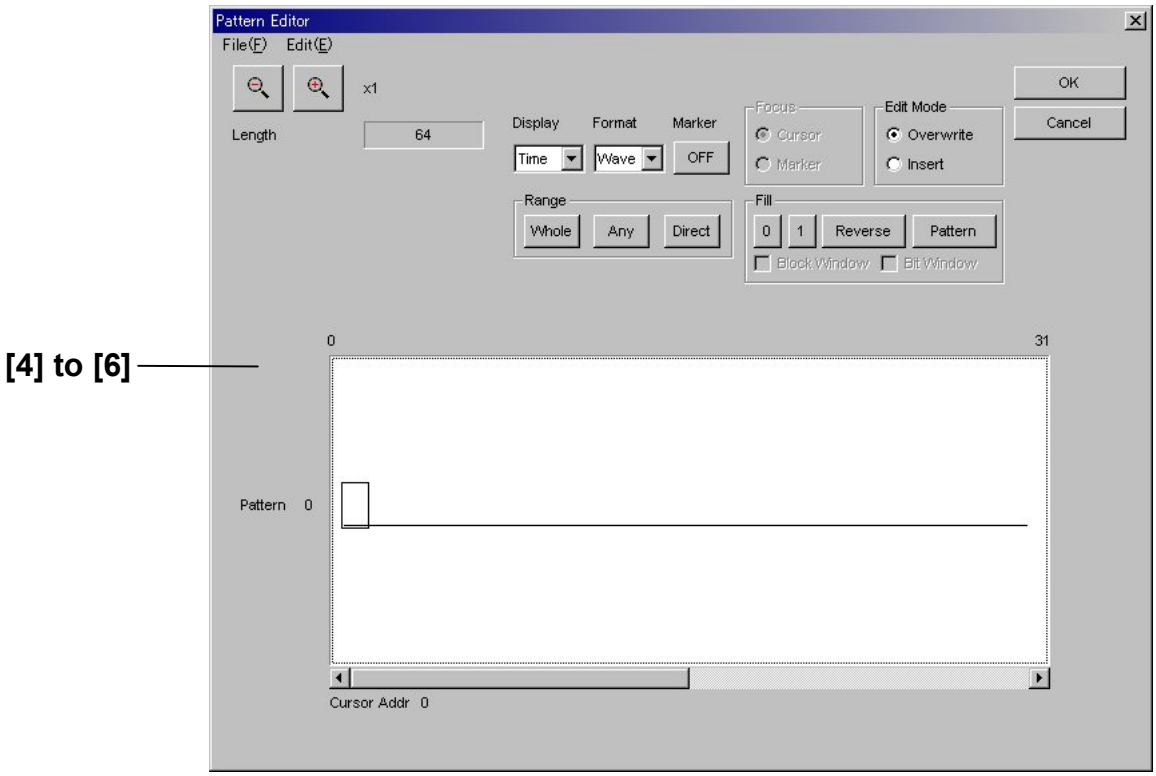


Figure 7.4.2.3-2 Sync Control setting (Pattern Editor screen)

Table 7.4.2.3-1 Sync Control setting commands

No	Setting Items	Commands
[1]	Control	:SENSe:PATTern:SYNC:PSMode
		:SENSe:PATTern:SYNC:PSMode?
[2]	Frame Length	:SENSe:PATTern:SYNC:FLENgth
		:SENSe:PATTern:SYNC:FLENgth?
[3]	Frame Position	:SENSe:PATTern:SYNC:FPOSition
		:SENSe:PATTern:SYNC:FPOSition?
[4]	Mask (Setting of mask pattern)	:SENSe:PATTern:SYNC:DATA:WHOLe
		:SENSe:PATTern:SYNC:DATA:WHOLe?
[5]	No label (Data inversion)	:SENSe:PATTern:SYNC:DREVerse:ADDRess
[6]	No label (All 0 or 1)	:SENSe:PATTern:SYNC:PROGram:FILL

:SENSe:PATTern:SYNC:PSMode <mode>

Parameter <mode> = <CHARACTER PROGRAM DATA>

FRAMe	Frame detection ON
NORMal	Frame detection OFF
QUICk	Quick synchronizing mode
FAST	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 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.

: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:ADDRess <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:ADDRess #H001,#H011
```

Compatibility
Incompatible with existing models.

:SENSe:PATTern:SYNC:PROGram:FILL <start>,<end>,<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 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
```

Compatibility

Incompatible with existing models.

7.4.2.4 Error/Alarm Condition setting commands

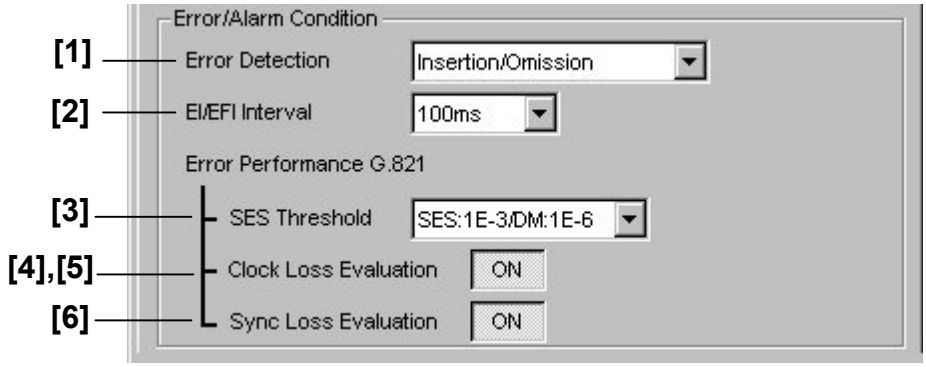


Figure 7.4.2.4-1 Error/Alarm Condition setting field

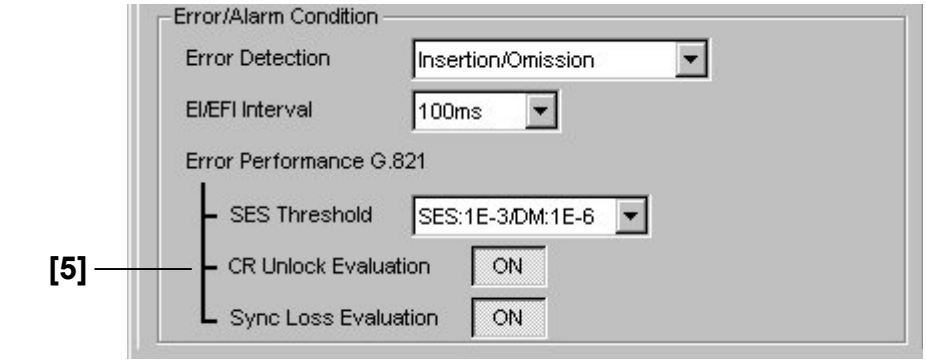


Figure 7.4.2.4-2 Error/Alarm Condition setting field (For Recovered Clock)

Table 7.4.2.4-1 Error/Alarm Condition setting commands

No.	Setting Items	Commands
[1]	Error Detection	:CALCulate:EALarm:ERRor:EDETect
		:CALCulate:EALarm:ERRor:EDETect?
[2]	EI/EFI Interval	:CALCulate:EALarm:ERRor:INTerval
		:CALCulate:EALarm:ERRor:INTerval?
[3]	SES Threshold	:CALCulate:EALarm:PERFormance:THReshold
		:CALCulate:EALarm:PERFormance:THReshold?
[4]	Clock Loss Evaluation	:CALCulate:EALarm:CLEValuation
		:CALCulate:EALarm:CLEValuation?
[5]	CR Unlock Evaluation	:CALCulate:EALarm:CRUNlock
		:CALCulate:EALarm:CRUNlock?
[6]	Sync Loss Evaluation	:CALCulate:EALarm:SLEValuation
		:CALCulate:EALarm:SLEValuation?

: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.	

: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 Does not set to the evaluation target. 1 or ON 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.

: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>

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 Sync Loss occurrence period to the measurement evaluation target.	
Example	To include the Sync Loss occurrence period to the measurement evaluation target: `> :CALCulate:EALarm:SLEValuation ON`	
Compatibility	Compatible with the MP1632C and MP1776A.	

:CALCulate:EALarm:SLEValuation?

Response	<numeric> = <NR1 NUMERIC RESPONSE DATA>	
	0	Out of evaluation target
	1	Evaluation target
Function	Queries whether the Sync Loss occurrence period is included to the measurement evaluation target.	
Example	`> :CALCulate:EALarm:SLEValuation?` `< 1`	
Compatibility	Compatible with the MP1632C and MP1776A.	

7.4.3 Commands related to Pattern tab

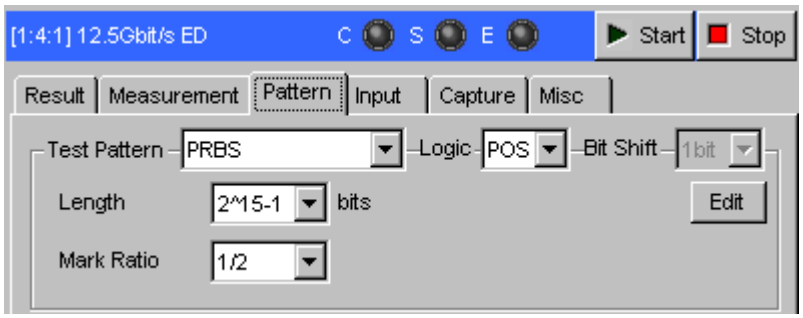


Figure 7.4.3-1 Pattern tab

7.4.3.1 Test Pattern setting commands

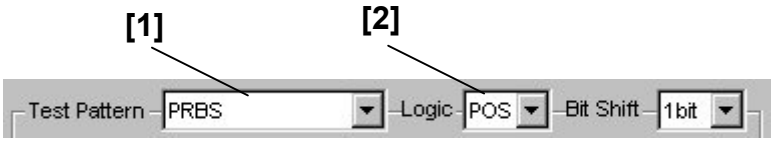


Figure 7.4.3.1-1 Test Pattern setting field

Table 7.4.3.1-1 Test Pattern setting commands

No.	Setting Items	Commands
[1]	Test Pattern	:SENSe:PATTern:TYPE
		:SENSe:PATTern:TYPE?
[2]	Logic	:SENSe:PATTern:LOGic
		:SENSe:PATTern:LOGic?

: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: `> :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	`> :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): `> :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	`> :SENSe:PATTern:LOGic?` `< POS`
Compatibility	Incompatible with existing models.

7.4.3.2 PRBS pattern setting commands

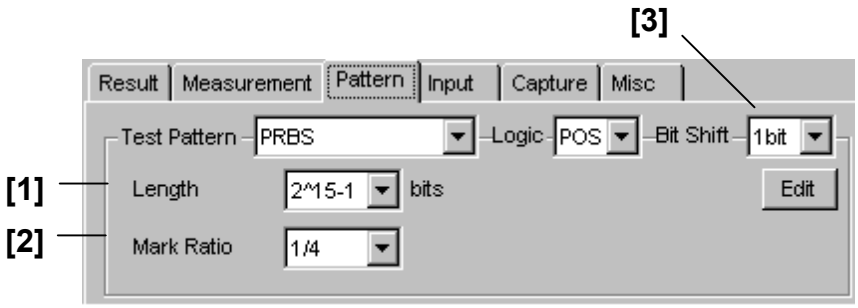


Figure 7.4.3.2-1 PRBS setting

Table 7.4.3.2-1 PRBS pattern setting commands

No.	Setting Items	Commands
[1]	Length	:SENSe:PATTern:PRBS:LENGth
		:SENSe:PATTern:PRBS:LENGth?
[2]	Mark Ratio	:SENSe:PATTern:PRBS:MRATio
		:SENSe:PATTern:PRBS:MRATio?
[3]	Bit Shift	:SENSe:PATTern:PRBS:BSHift
		:SENSe:PATTern:PRBS:BSHift?

:SENSe:PATTern: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 (2^n-1 (n = 7, 9, 10, 11, 15, 20, 23, or 31)) 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.

:SENSe: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 reception.
Example	> :SENSe:PATTern:PRBS:LENGth? < 7
Compatibility	Incompatible with existing models.

:SENSe:PATTern: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 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	<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 reception.
Example	> :SENSe:PATTern:PRBS:MRATio? < M1_2
Compatibility	Compatible with the MP1632C.

:SENSe:PATTern:PRBS:BSHift <numeric>

Parameter	<numeric> = <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?

Response	<numeric> = <NR1 NUMERIC RESPONSE DATA> 1, 3
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

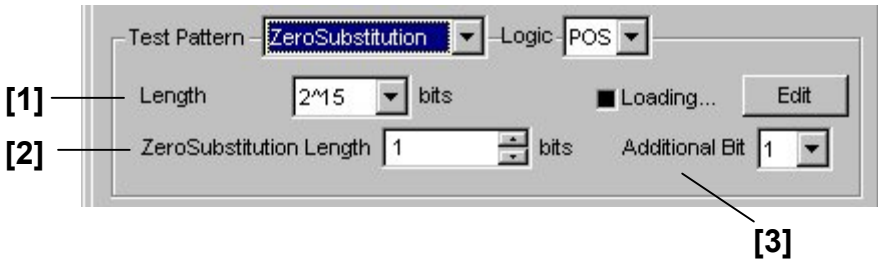


Figure 7.4.3.3-1 Zero Substitution pattern setting

Table 7.4.3.3-1 Zero Substitution pattern setting commands

No	Setting Items	Commands
[1]	Length	:SENSe:PATTern:ZSUBstitute:LENGth
		:SENSe:PATTern:ZSUBstitute:LENGth?
[2]	Zero Substitution Length	:SENSe:PATTern:ZSUBstitute:ZLENgth
		:SENSe:PATTern:ZSUBstitute:ZLENgth?
[3]	Additional Bit	:SENSe:PATTern:ZSUBstitute:ADDBit
		:SENSe:PATTern:ZSUBstitute:ADDBit?

:SENSe: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 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.

: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 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 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.

:SENSe:PATTern:ZSUBstitute:ADDBit <numeric>

Parameter	<numeric> = <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: `> :SENSe:PATTern:ZSUBstitute:ADDBit 0`	
Compatibility	Incompatible with existing models.	

:SENSe:PATTern:ZSUBstitute:ADDBit?

Response	<numeric> = <DECIMAL NUMERIC RESPONSE DATA>	
	0	One bit of "0" is added next to the last of zero-inserted bits.
	1	One bit of "1" is added next to the last of zero-inserted bits (same as existing models).
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

Figure 7.4.3.4-1 Data pattern setting

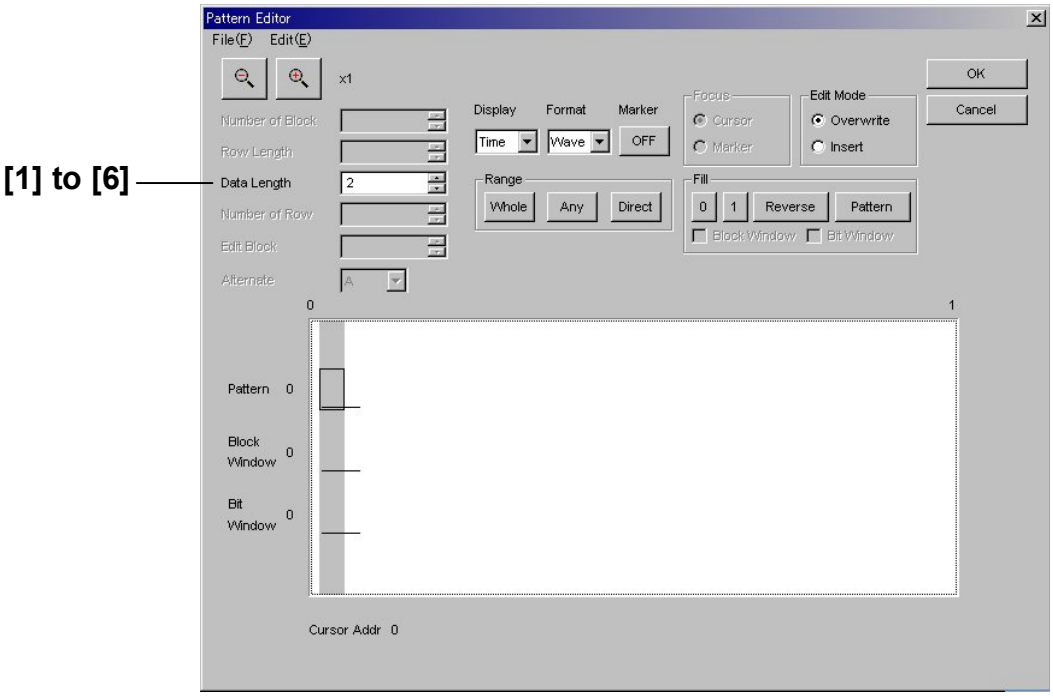


Figure 7.4.3.4-2 Data pattern setting (Pattern Editor screen)

Table 7.4.3.4-1 Data pattern setting commands

No.	Setting Items	Commands
[1]	Data Length	:SENSe:PATTern:DATA:LENGth
		:SENSe:PATTern:DATA:LENGth?
[2]	No label (Pattern setting)	:SENSe:PATTern:DATA:WHOLe
		:SENSe:PATTern:DATA:WHOLe?
[3]	No label (Pattern setting)	:SENSe:PATTern:BDATa:WHOLe
		:SENSe:PATTern:BDATa:WHOLe?
[4]	No label (Data inversion)	:SENSe:PATTern:DREVerse:ADDRess
[5]	No label (Data inversion)	:SENSe:PATTern:DREVerse:DELTa
[6]	No label (All 0 or 1)	:SENSe:PATTern:DATA:FILL

: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:WHOLe <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

:SENSe:PATTern:DATA:WHOLe? <start>

Parameter
<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 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 #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 16777216 bytes
<binary>:Binary data up to 16777216 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 binary data of the pattern data from bit 0 to bit 7FFFFFF addresses during Data pattern reception:

```
> :SENSe:PATTern:BDATa:WHOLe #H0,#H7FFFFFF,#10011
```

Compatibility
Compatible with the MP1632C and MP1776A.

:SENSe:PATTern:BDATa:WHOLe? <start>[,size]

Parameter

<start> = <NON-DECIMAL PROGRAM DATA>

#H0 to #H7FFFFFF 0 to 7FFFFF 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:

```
> :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 7FFFFF bits, in 1-bit steps (Specify in hexadecimal)

<end> = <NON-DECIMAL PROGRAM DATA>

#H0 to #H7FFFFFF 0 to 7FFFFF 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:

```
> :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 #H7FFFFFF	0 to 7FFFFFF 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

$$\text{Max} = \frac{\text{Pattern Length}}{128} \quad \text{(rounding up fractions)}$$

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 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

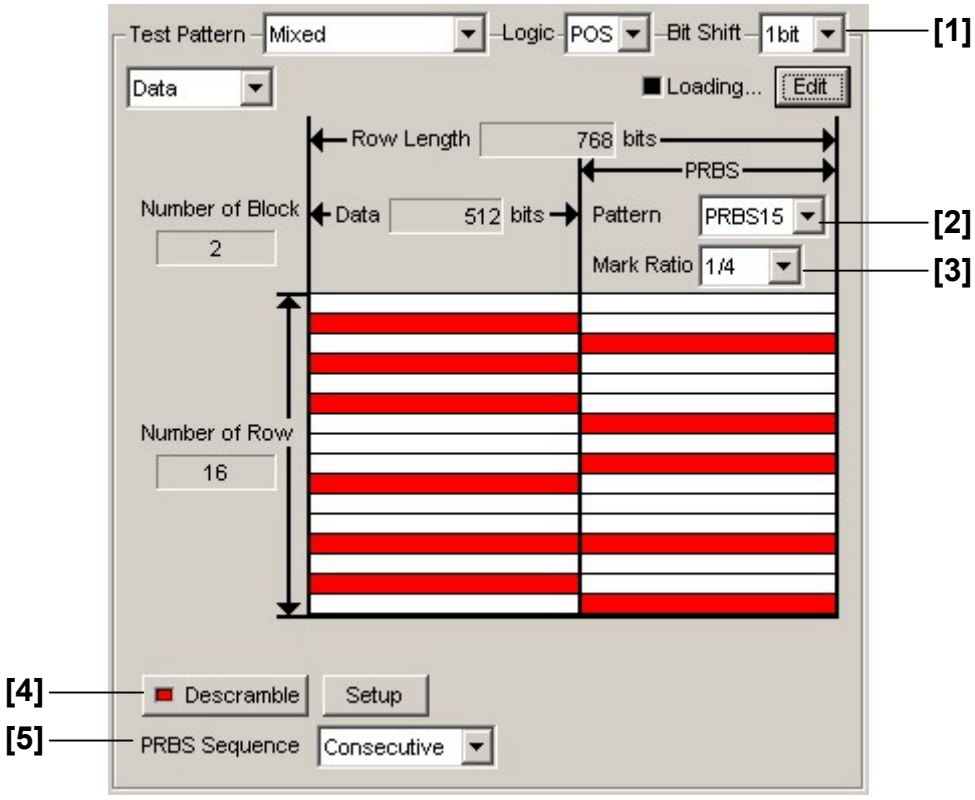


Figure 7.4.3.5-1 Mixed Data pattern setting

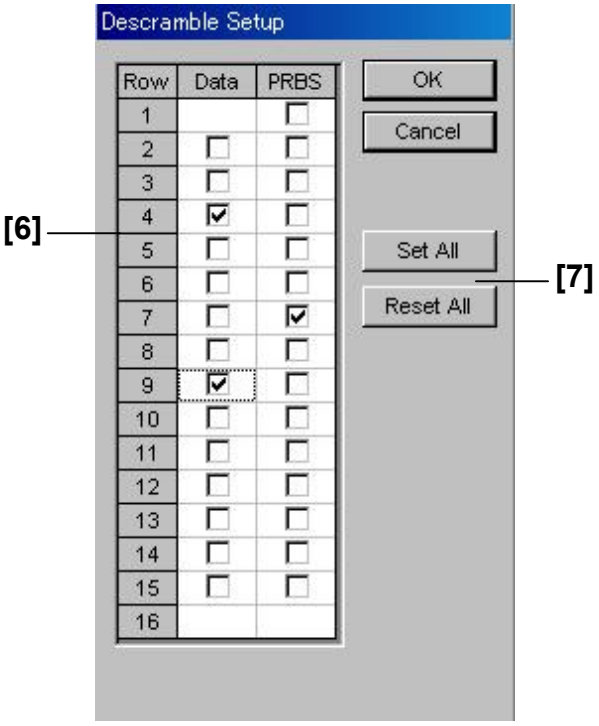


Figure 7.4.3.5-2 Descramble Setup dialog box

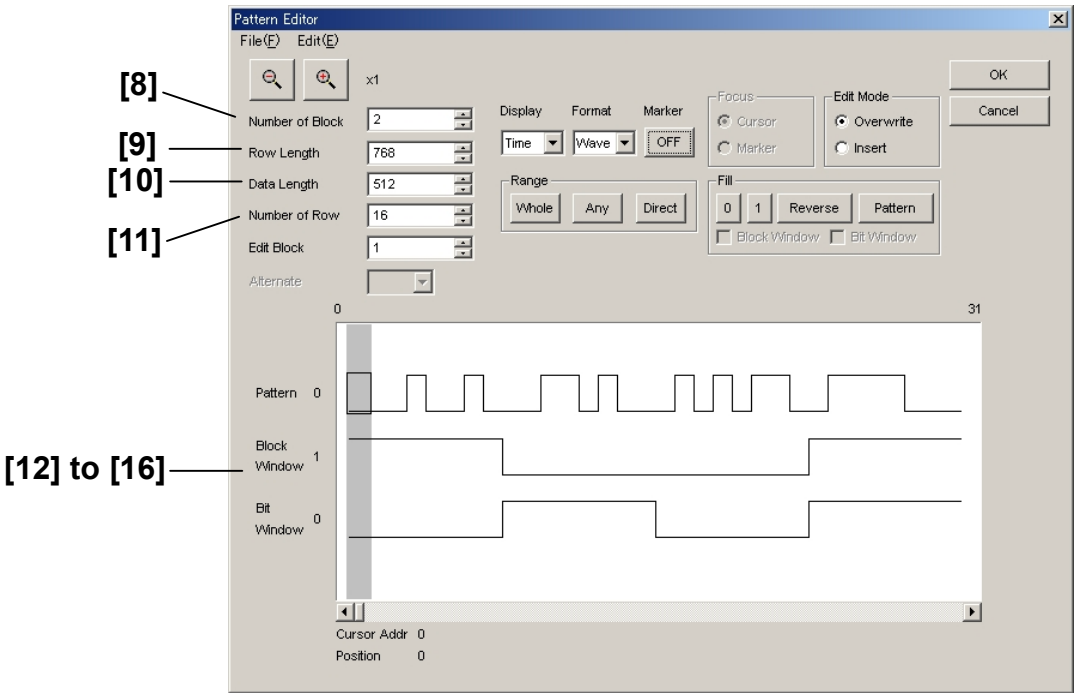


Figure 7.4.3.5-3 Mixed Data pattern setting (Pattern Editor screen)

Table 7.4.3.5-1 Mixed Data pattern setting commands

No	Setting Items	Commands
[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?

Table 7.4.3.5-1 Mixed Data pattern setting commands (Cont'd)

No	Setting Items	Commands
[12]	No label (Pattern setting)	:SENSe:PATTern:MIXData:DATA:WHOLe
		:SENSe:PATTern:MIXData:DATA:WHOLe?
[13]	No label (Pattern setting)	:SENSe:PATTern:MIXData:BDATa:WHOLe
		:SENSe:PATTern:MIXData:BDATa:WHOLe?
[14]	No label (Data inversion)	:SENSe:PATTern:MIXData:DREVerse:ADDRess
[15]	No label (Data inversion)	:SENSe:PATTern:MIXData:DREVerse:DELTa
[16]	No label (Data inversion)	:SENSe:PATTern:MIXData:DATA:FILL

: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>

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 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?

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	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.

:SENSe: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 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> 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 reception.
Example	`> :SENSe:PATTern:MIXData:PRBS:MRATio?` `< M1_2`
Compatibility	Incompatible with existing models.

: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	Incompatible with existing models.	

:SENSe:PATTern:MIXData:DESCramble?

Response	<numeric> = <NR1 NUMERIC RESPONSE DATA> 0 Descramble OFF 1 Descramble ON
Function	Queries the Descramble ON/OFF state of the PRBS7 stage during pattern reception.
Example	`> :SENSe:PATTern:MIXData:DESCramble?` `< 1`
Compatibility	Incompatible with existing models.

:SENSe:PATTern:MIXData:PRBS:SEQuence <sequence>

Parameter	<sequence> = <CHARACTER PROGRAM DATA> RESTart Discontinuous PRBS pattern CONSecutive Continuous PRBS pattern
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: `> :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	`> :SENSe:PATTern:MIXData:PRBS:SEQuence?` `< REST`
Compatibility	Incompatible with existing models.

:SENSe: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 Descramble OFF
	1 or ON Descramble ON
	Note, however, that Descramble OFF is fixed for Data of 1 Row.
	<prbs> = <BOOLEAN PROGRAM DATA>
	0 or OFF Descramble OFF
	1 or ON Descramble ON
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? <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 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 <numeric>

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.

:SENSe: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 reception.
Example	To set the number of blocks during Mixed Data pattern reception to 1: `> :SENSe:PATTern:MIXData:BLOCk 1`
Compatibility	Incompatible with existing models.

:SENSe:PATTern:MIXData:BLOCk?

Response	<numeric> = <NR1 NUMERIC RESPONSE DATA> 1 to 511 1 to 511 blocks
Function	Queries the number of blocks during Mixed Data pattern reception.
Example	`> :SENSe:PATTern:MIXData:BLOCk?` `< 1`
Compatibility	Incompatible with existing models.

:SENSe:PATTern:MIXData:RLENgth <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 reception.
Example	To set the pattern length of 1 Row that is edited during Mixed Data pattern reception to 768 bits: `> :SENSe:PATTern:MIXData:RLENgth 768`
Compatibility	Incompatible with existing models.

: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.

: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 1 to 511 blocks, in 1-block steps

<start> = <NON-DECIMAL PROGRAM DATA>

#H0 to #H7FFFFFF 0 to 7FFFFF bits, in 1-bit steps (Specify in hexadecimal)

<end> = <NON-DECIMAL PROGRAM DATA>

#H0 to #H7FFFFFF 0 to 7FFFFF 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 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,#H7FFFFFF,
"H001"
```

Compatibility

Incompatible with existing models.

:SENSe:PATTern:MIXData:DATA:WHOLe? <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 7FFFFF 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:PATTern:MIXData:DATA:WHOLe? 3,#H0
< "H001"
```

Compatibility

Incompatible with existing models.

:SENSe:PATTern:MIXData:BDATa:WHOLe <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 7FFFFF bits, in 1-bit steps (Specify in hexadecimal)

<end> = <NON-DECIMAL PROGRAM DATA>

#H0 to #H7FFFFFF 0 to 7FFFFF 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 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 7FFFFF 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.

:SENSe:PATTern:MIXData:DREVerse:ADDRess <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 #H7FFFFFF 0 to 7FFFFF bits, in 1-bit steps (Specify in hexadecimal)

<end> = <NON-DECIMAL PROGRAM DATA>

#H0 to #H7FFFFFF 0 to 7FFFFF 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 7FFFFFF addresses during Mixed Data pattern reception:

```
> :SENSe:PATTern:MIXData:DREVerse:ADDRess 3,#H0,#H7FFFFFF
```

Compatibility

Incompatible with existing models.

:SENSe:PATTern:MIXData:DREVerse:DELTa <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 #H7FFFFFF 0 to 7FFFFF 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

$$\text{Max} = \frac{\text{Pattern Length}}{128} \text{ (rounding up fractions)}$$

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:PATTern:MIXData:DATA:FILL 3,PAGE,1,0
```

Compatibility

Incompatible with existing models.

7.4.3.6 Sequence pattern setting commands

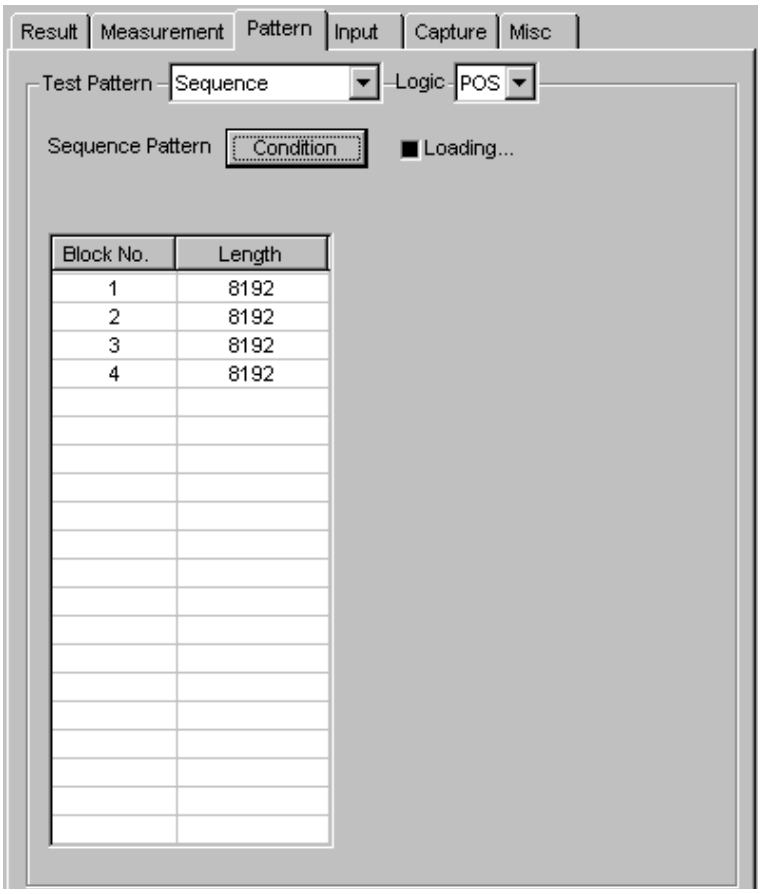


Figure 7.4.3.6-1 Sequence pattern setting

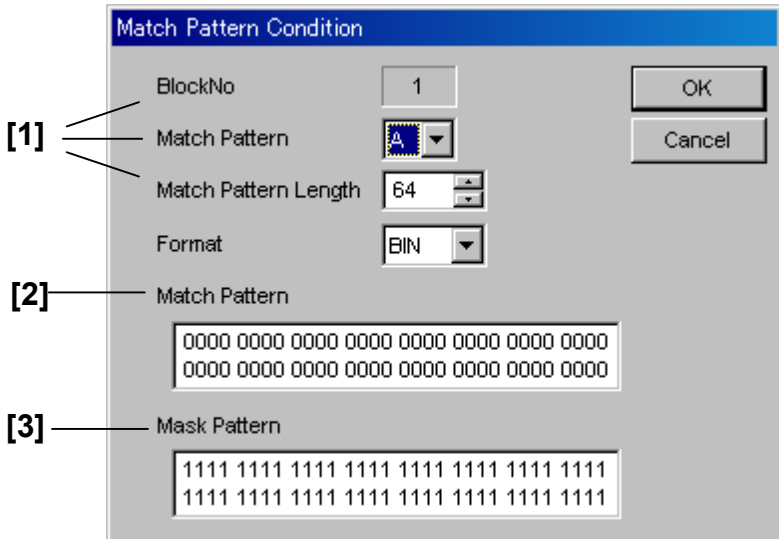


Figure 7.4.3.6-2 Match Pattern setting

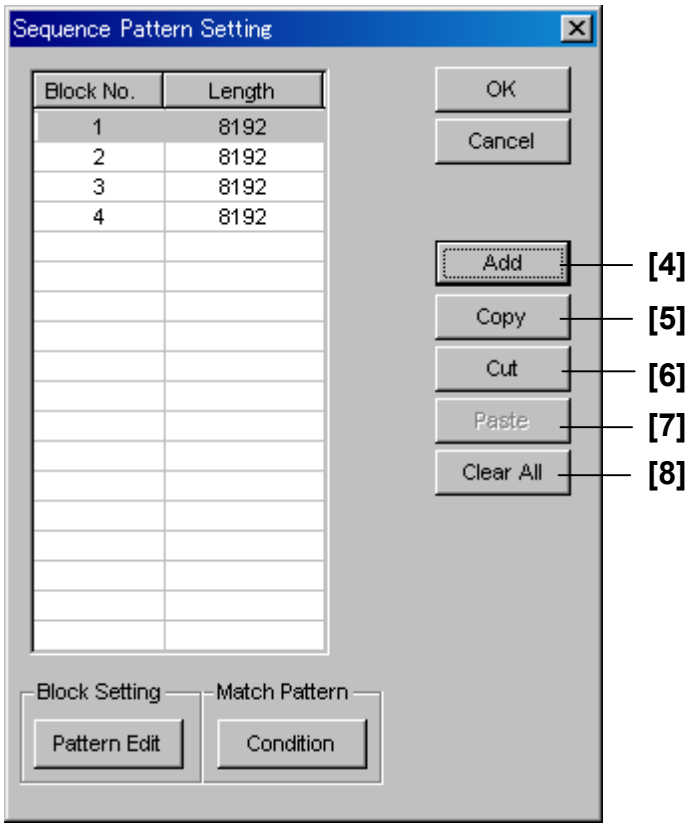


Figure 7.4.3.6-3 Sequence Pattern setting

[9] to [14]

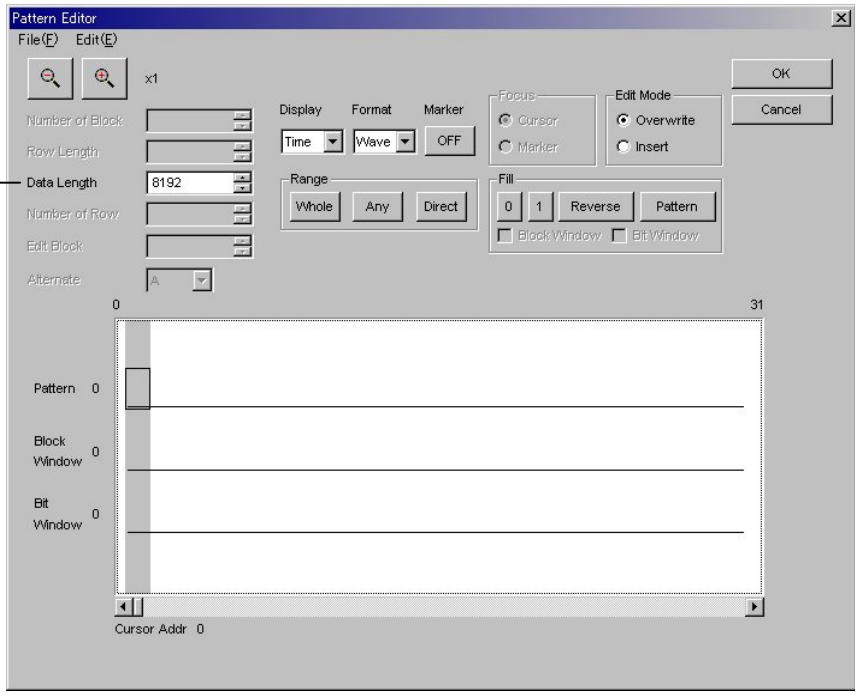


Figure 7.4.3.6-4 Sequence pattern setting (Pattern Editor screen)

Table 7.4.3.6-1 Sequence pattern setting commands

No.	Setting Items	Commands
[1]	Match Pattern Length	:SENSe:PATTern:SEQuence:MATCh:LENGth
		:SENSe:PATTern:SEQuence:MATCh:LENGth?
[2]	Match Pattern	:SENSe:PATTern:SEQuence:MATCh:EDIT
		:SENSe:PATTern:SEQuence:MATCh:EDIT?
[3]	Mask Pattern	:SENSe:PATTern:SEQuence:MATCh:MASK
		:SENSe:PATTern:SEQuence:MATCh:MASK?
[4]	Add	:SENSe:PATTern:SEQuence:CONDition:ADD
		:SENSe:PATTern:SEQuence:CONDition:ADD?
[5]	Copy	:SENSe:PATTern:SEQuence:COPY
[6]	Cut	:SENSe:PATTern:SEQuence:CUT
[7]	Paste	:SENSe:PATTern:SEQuence:PASTe
[8]	Clear All	:SENSe:PATTern:SEQuence:CLEar
[9]	Data Length	:SENSe:PATTern:SEQuence:LENGth
		:SENSe:PATTern:SEQuence:LENGth?
[10]	No label (Pattern setting)	:SENSe:PATTern:SEQuence:DATA:WHOLe
		:SENSe:PATTern:SEQuence:DATA:WHOLe?
[11]	No label (Pattern setting)	:SENSe:PATTern:SEQuence:BDATa:WHOLe
		:SENSe:PATTern:SEQuence:BDATa:WHOLe?
[12]	No label (Data inversion)	:SENSe:PATTern:SEQuence:DREVerse:ADDRess
[13]	No label (Data inversion)	:SENSe:PATTern:SEQuence:DREVerse:DELTa
[14]	No label (All 0 or All 1)	:SENSe:PATTern:SEQuence:DATA:FILL

: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)
	Range of string length: Depends on the length of the Match Pattern
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**:
	`> :SENSe:PATTern:SEQuence:MATCh:EDIT? 3,APAT,BIN`
	`< "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.

:SENSe:PATTern:SEQuence:MATCh:MASK? <block>,<content>,<form>

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**:
	`> :SENSe:PATTern:SEQuence:MATCh:MASK? 3,APAT,BIN`
	`< "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> = <DECIMAL NUMERIC PROGRAM DATA> 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> = <DECIMAL NUMERIC PROGRAM DATA> 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> = <DECIMAL NUMERIC PROGRAM DATA> 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:PATTern:SEQuence:LENGth <block>,<numeric>

Parameter	<block> = <DECIMAL NUMERIC PROGRAM DATA>
	1 to 128 1 to 128 blocks, 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 length of the specified block during Sequence pattern reception.
Example	To set the length of block 1 to 1 bit during Sequence pattern reception:
	`> :SENSe:PATTern:SEQuence:LENGth 1, 1`
Compatibility	Incompatible with existing models.

:SENSe:PATTern:SEQuence:LENGth? <block>

Parameter	<block> = <DECIMAL NUMERIC PROGRAM DATA>
	1 to 128 1 to 128 blocks, 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 length of the specified block during Sequence pattern reception.
Example	To query the length of block 1 during Sequence pattern reception:
	`> :SENSe:PATTern:SEQuence:LENGth? 1`
	`< 1`
Compatibility	Incompatible with existing models.

:SENSe:PATTern: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 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 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:PATTern:SEQuence:DATA:WHOLe 1,#H0,#HFFFFF, "H001"
```

Compatibility

Incompatible with existing models.

:SENSe:PATTern:SEQuence:DATA: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 #HFFFFF	0 to FFFFF 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 reception.

Example

To query the block 1 pattern data of 400 × 4 bits from bit 0 address during Sequence pattern reception:

```
> :SENSe:PATTern:SEQuence:DATA:WHOLe? 1,#H0
< "H00100100100100100100…"
```

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:WHOLe 1,#H0,#HFFFFF,#10011
```

Compatibility

Incompatible with existing models.

: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

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 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
```

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 FFFFFbits, 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

$$\text{Max} = \frac{\text{Pattern Length}}{128} \text{ (rounding up fractions)}$$

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.

7.4.3.7 Commands related to Mask

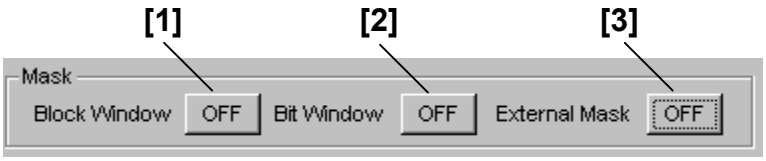


Figure 7.4.3.7-1 Mask ON/OFF setting

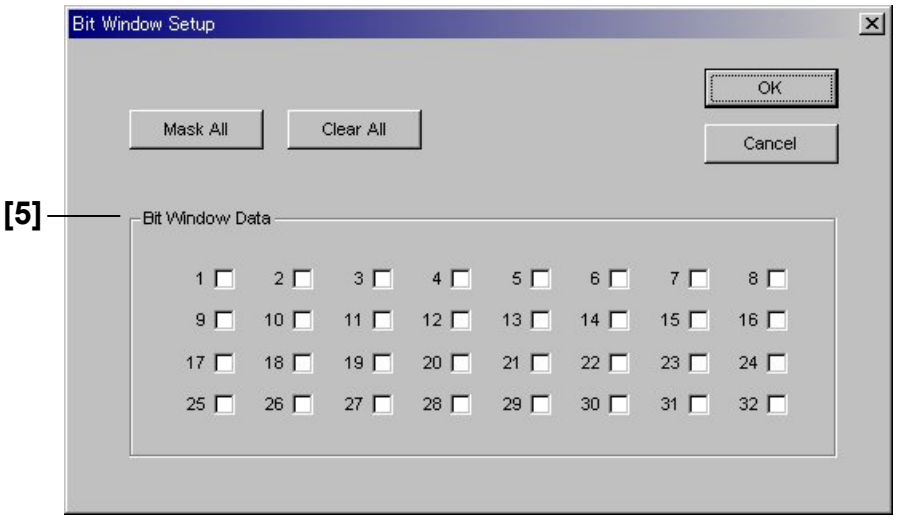


Figure 7.4.3.7-2 Bit Window Setup dialog box (for PRBS pattern)

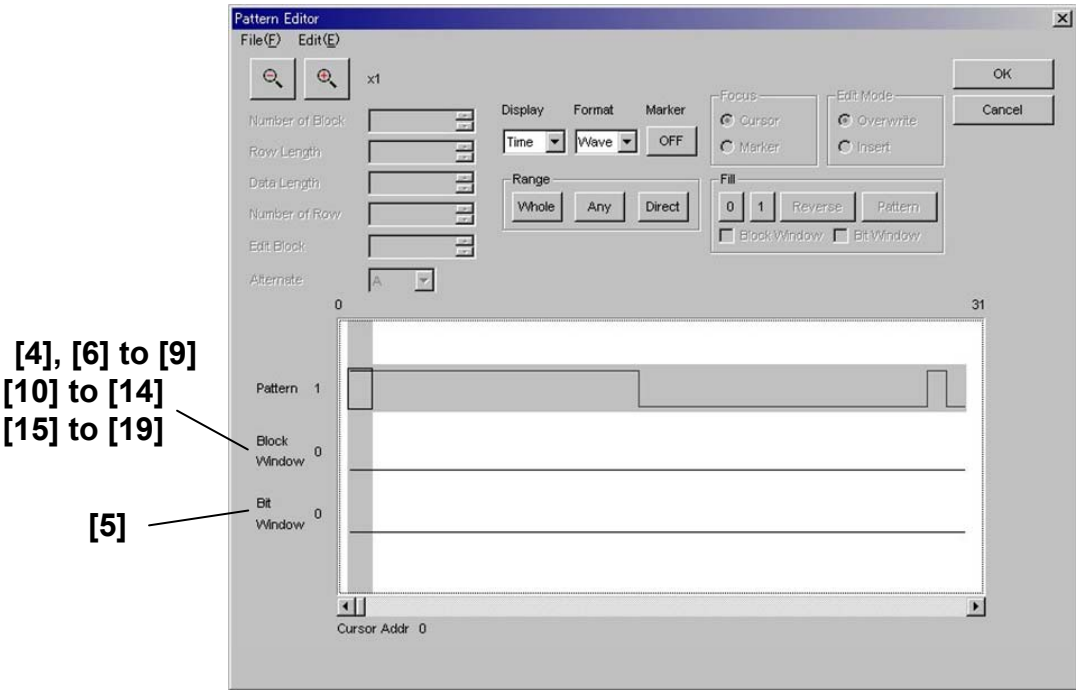


Figure 7.4.3.7-3 Bit Window and Block Window settings in Pattern Editor screen (for Zero Substitution, Data, Mixed Data, and Sequence patterns)

Table 7.4.3.7-1 Mask setting commands

No.	Setting Items	Commands
[1]	Block Window ON/OFF	:SENSe:PATTern:MASK:BLOCk
		:SENSe:PATTern:MASK:BLOCk?
[2]	Bit Window ON/OFF	:SENSe:PATTern:MASK:BIT
		:SENSe:PATTern:MASK:BIT?
[3]	External Mask ON/OFF	:SENSe:PATTern:MASK:EXTernal
		:SENSe:PATTern:MASK:EXTernal?
[4]	Block Window (Zero-Substitution, Data)	:SENSe:PATTern:MASK:BLKWindow
		:SENSe:PATTern:MASK:BLKWindow?
[5]	Block Window	:SENSe:PATTern:MASK:BTWindow
		:SENSe:PATTern:MASK:BTWindow?
[6]	Block Window (Pattern setting)	:SENSe:PATTern:MASK:BDATa:WHOLe
		:SENSe:PATTern:MASK:BDATa:WHOLe?
[7]	Block Window (Data inversion)	:SENSe:PATTern:MASK:DREVerse:ADDRess
[8]	Block Window (Data inversion)	:SENSe:PATTern:MASK:DREVerse:DELTa
[9]	Block Window (All 0 or All 1)	:SENSe:PATTern:MASK:DATA:FILL
[10]	Block Window (Mixed Data)	:SENSe:PATTern:MASK:MIXData:BLKWindow
		:SENSe:PATTern:MASK:MIXData:BLKWindow?
[11]	Block Window (Pattern setting)	:SENSe:PATTern:MASK:MIXData:BDATa:WHOLe
		:SENSe:PATTern:MASK:MIXData:BDATa:WHOLe?
[12]	Block Window (Data inversion)	:SENSe:PATTern:MASK:MIXData:DREVerse:ADDRess
[13]	Block Window (Data inversion)	:SENSe:PATTern:MASK:MIXData:DREVerse:DELTa
[14]	Block Window (All 0 or All 1)	:SENSe:PATTern:MASK:MIXData:DATA:FILL
[15]	Block Window (Sequence)	:SENSe:PATTern:MASK:SEQuence:BLKWindow
		:SENSe:PATTern:MASK:SEQuence:BLKWindow?
[16]	Block Window (Pattern setting)	:SENSe:PATTern:MASK:SEQuence:BDATa:WHOLe
		:SENSe:PATTern:MASK:SEQuence:BDATa:WHOLe?
[17]	Block Window (Data inversion)	:SENSe:PATTern:MASK:SEQuence:DREVerse:ADDRess
[18]	Block Window (Data inversion)	:SENSe:PATTern:MASK:SEQuence:DREVerse:DELTa
[19]	Block Window (All 0 or All 1)	:SENSe:PATTern:MASK:SEQuence:DATA:FILL

: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:

```
> :SENSe:PATTern:MASK:BLOCk ON
```

Compatibility Incompatible with existing models.

:SENSe:PATTern:MASK:BLOCk?

Response	<numeric> = <NR1 NUMERIC RESPONSE DATA>	
	0	Block Window OFF
	1	Block Window ON
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>

Parameter	<boolean> = <BOOLEAN PROGRAM DATA>	
	0 or OFF	Bit Window OFF
	1 or ON	Bit Window ON
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?

Response	<numeric> = <NR1 NUMERIC RESPONSE DATA>	
	0	Bit Window OFF
	1	Bit Window ON
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>

Parameter	<boolean> = <BOOLEAN PROGRAM DATA>	
	0 or OFF	External Mask OFF
	1 or ON	External Mask ON
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
<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 × 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.

:SENSe:PATTern:MASK:BLKWindow? <start>

Parameter
<start>=<NON-DECIMAL PROGRAM DATA>
#H0 to #H7FFFFFF 0 to 7FFFFF 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 from the <Start> address during Block Window setting.

Example
To query the mask pattern data of 400 × 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.

: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?
< "H00100100"
```

Compatibility
Incompatible with existing models.

:SENSe:PATTern:MASK:BDATa:WHOLe <start>,<end>,<bdata>

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.

<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 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 7FFFFFF addresses during Block Window setting:
```
> :SENSe:PATTern:MASK:BDATa:WHOLe #H0,#H7FFFFFF,#11A
```

Compatibility
Incompatible with existing models.

:SENSe:PATTern:MASK:BDATa:WHOLe? <start>[,<size>]

Parameter

<start> = <NON-DECIMAL PROGRAM DATA>

#H0 to #H7FFFFFF 0 to 7FFFFF 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>

Parameter

<start> = <NON-DECIMAL PROGRAM DATA>

#H0 to #H7FFFFFF 0 to 7FFFFF bits, in 1-bit steps (Specify in hexadecimal)

<end> = <NON-DECIMAL PROGRAM DATA >

#H0 to #H7FFFFFF 0 to 7FFFFF 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 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:

```
> :SENSe:PATTern:MASK:DREVerse:ADDRess #H0,#H7FFFFF
```

Compatibility

Incompatible with existing models.

:SENSe:PATTern:MASK:DREVerse:DELTa <start>,<delta>

Parameter

<start> = <NON-DECIMAL PROGRAM DATA>

#H0 to #H7FFFFFF 0 to 7FFFFF 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 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:

```
> :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

$$\text{Max} = \frac{\text{Pattern Length}}{128} \text{ (rounding up fractions)}$$

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 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.

:SENSe:PATTern:MASK:MIXData:BLKWindow <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 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 1 to 511 blocks, in 1-block steps

<start> = <NON-DECIMAL PROGRAM DATA >

#H0 to #H7FFFFFF 0 to 7FFFFF 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 <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 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 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 7FFFFFF addresses when **Test Sequence** is **Mixed**:

```
>:SENSe:PATTern:MASK:MIXData:BDATa:WHOLe
1,#H0,#H7FFFFFF,#11A
```

Compatibility

Incompatible with existing models.

: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 16,777,216bytes

<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 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... (binary data of 16,000 bytes)
```

Compatibility

Incompatible with existing models.

:SENSe:PATTern:MASK:MIXData:DREVerse:ADDRess <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 #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.

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**:

```
>:SENSe:PATTern:MASK:MIXData:DREVerse:ADDRess
1,#H0,#H7FFFFFF
```

Compatibility

Incompatible with existing models.

:SENSe:PATTern:MASK:MIXData:DREVerse:DELTa <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 #H7FFFFFF 0 to 7FFFFFF 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 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**:

```
> :SENSe:PATTern:MASK:MIXData:DREVerse:DELTa 1,#H0,1
```

Compatibility

Incompatible with existing models.

:SENSe:PATTern:MASK: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
	$\text{Max} = \frac{\text{Pattern Length}}{128}$ (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 Block Window mask pattern date 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 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.
"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 FFFFF 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.

: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 FFFFF 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

<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 131,072bytes
- <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 7FFFFFF addresses when **Test Pattern** is **Sequence**.

```
>:SENSe:PATTern:MASK:SEQuence:BDATa:WHOLe
1,#H0,#H7FFFFFF,#11A
```

Compatibility

Incompatible with existing models.

: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

<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 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:PATTern:MASK:SEQuence:BDATa:WHOLe? 1,#H0
< #516000AAAAAA... (binary data of 16,000 bytes)
```

Compatibility

Incompatible with existing models.

:SENSe:PATTern:MASK: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 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 FFFFF addresses when **Test Pattern** is **Sequence**:

```
> :SENSe:PATTern:MASK:SEQuence:DREVerse:ADDRess
1,#H0,#HFFFFF
```

Compatibility

Incompatible with existing models.

:SENSe:PATTern:MASK: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 #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 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**:

```
> :SENSe:PATTern:MASK:SEQuence:DREVerse:DELTa 1,#H0,1
```

Compatibility

Incompatible with existing models.

:SENSe:PATTern:MASK: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 8192 — 1 to 8192 (Max.) pages, in 1-page steps

$\text{Max} = \frac{\text{Pattern Length}}{128}$ (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 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

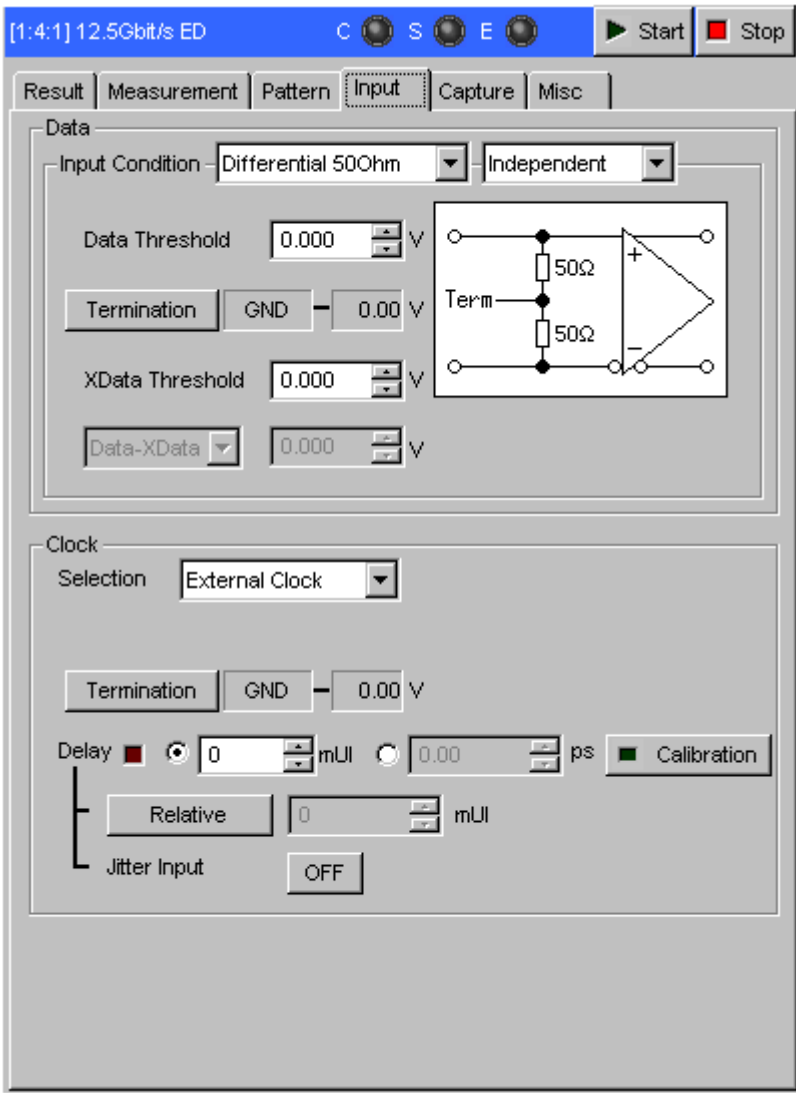


Figure 7.4.4-1 Input tab

7.4.4.1 Data Input setting commands

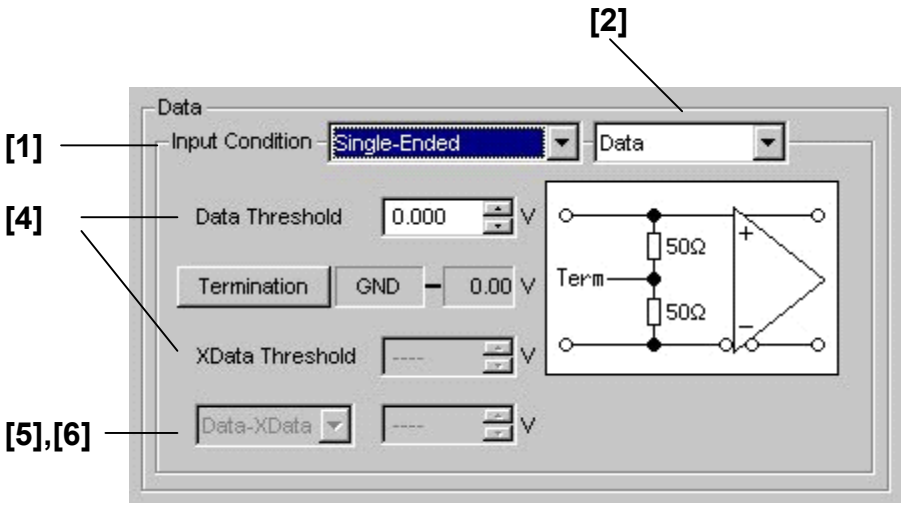


Figure 7.4.4.1-1 Single-End Data setting

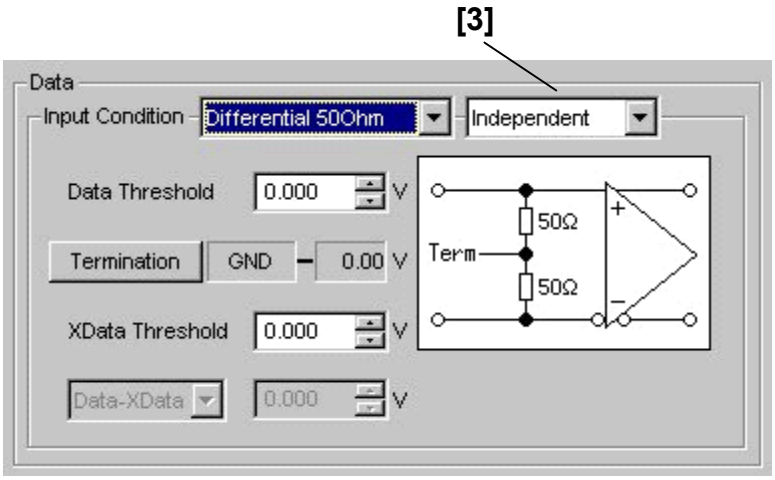


Figure 7.4.4.1-2 Differential 50 Ohm setting

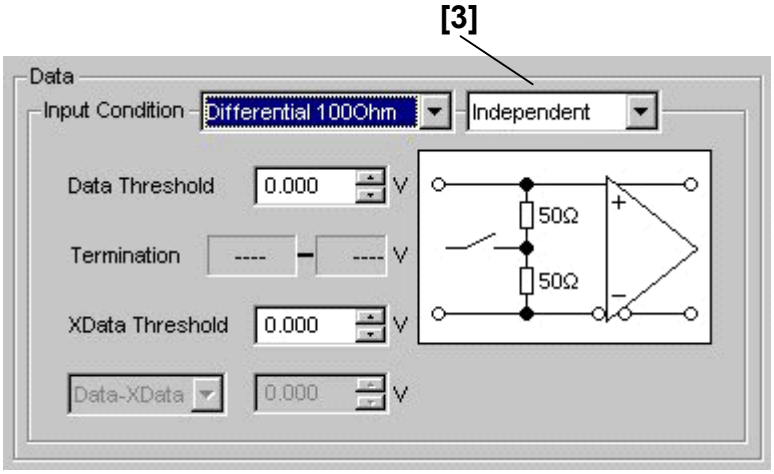


Figure 7.4.4.1-3 Differential 100 Ohm setting

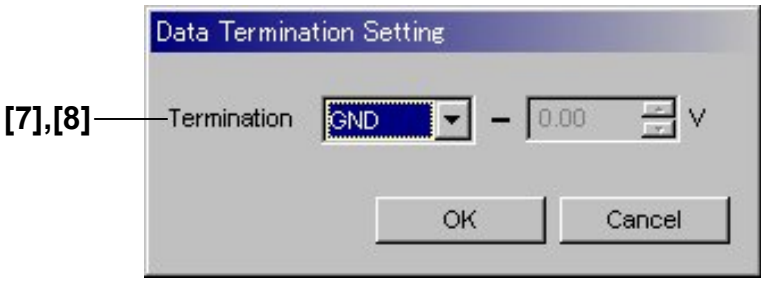


Figure 7.4.4.1-4 Termination Setting dialog box

Table 7.4.4.1-1 Data Input setting commands

No.	Setting Items	Commands
[1]	Input Condition	:INPut:DATA:INTerface
		: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?
[7]	Termination	:INPut:DATA:TERMination
		:INPut:DATA:TERMination?
[8]	No label (Termination voltage setting)	:INPut:DATA:TLEVel
		:INPut:DATA:TLEVel?

:INPut:DATA:INTerface <interface>

Parameter	<interface> = <CHARACTER PROGRAM DATA>
	SINGle Single Ended
	DIF50ohm Differential 50 Ω
	DIF100ohm Differential 100 Ω
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	<interface> = <CHARACTER RESPONSE DATA> SING, DIF50, DIF100
Function	Queries the data input interface.
Example	`> :INPut:DATA:INTerface?` `< SING`
Compatibility	Incompatible with existing models.

: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	Incompatible with existing models.

: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	Incompatible with existing models.

:INPut:DATA:DIFFerential <item>

Parameter	<item> = <CHARACTER PROGRAM DATA>	
	INDependent	Data/Xdata independent setting
	TRACking	Data/Xdata common setting
	ALTernate	Data/Xdata differential setting
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?

Response	<item> = <CHARACTER RESPONSE DATA> IND, TRAC, ALT
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>

Parameter	<port> = <CHARACTER PROGRAM DATA>	
	DATA	Selects Data.
	XDATa	Selects Xdata.
	Be sure to input the selected port when Single is selected.	
	<numeric> = <DECIMAL NUMERIC PROGRAM DATA>	
	–3.500 to 3.300	–3.500 to 3.300 V, in 0.001 V steps (When MU181040A-002 and MU181040B-002 are installed)
	–0.350 to 0.350	–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 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>

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 (When MU181040A-002 and MU181040B-002 are installed)
	–0.350 to 0.350	–0.350 to 0.350 V (when MU181040A-001 is installed)
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>

Parameter	<basis> = <CHARACTER PROGRAM DATA>	
	DATA	Sets the differential reference signal to Data.
	XDATa	Sets the differential reference signal to 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 DATA: `> :INPut:DATA:DIFFerential:BASis DATA`	
Compatibility	Incompatible with existing models.	

:INPut:DATA:DIFFerential:BASis?

Response	<basis> = <CHARACTER RESPONSE DATA> 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 <numeric>

Parameter	<numeric> = <DECIMAL NUMERIC PROGRAM DATA>	
	–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	<numeric> = <NR2 NUMERIC RESPONSE DATA>
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 <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 data input termination condition.	
Example	To set the data input termination condition to GND: > :INPut:DATA:TERMination GND	
Compatibility	Incompatible with existing models.	

: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.

7.4.4.2 Clock Input setting commands

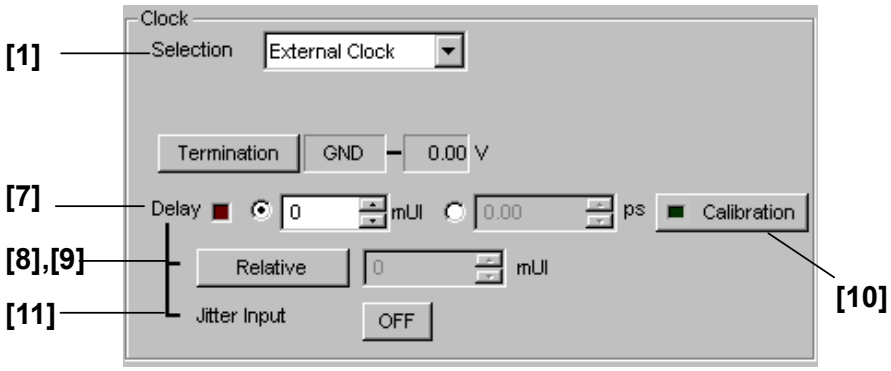


Figure 7.4.4.2-1 Clock Input setting (For External Clock)

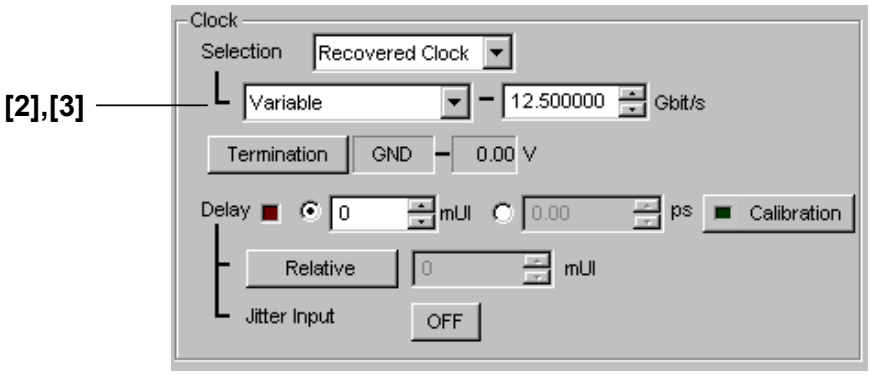


Figure 7.4.4.2-2 Clock Input setting (For Recovered Clock)

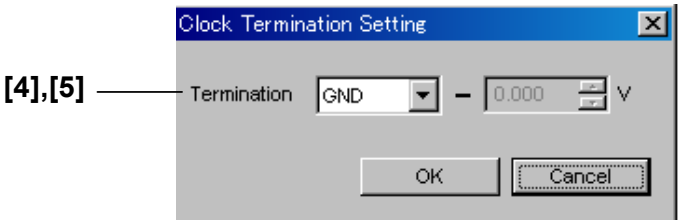


Figure 7.4.4.2-3 Clock Termination Setting dialog box

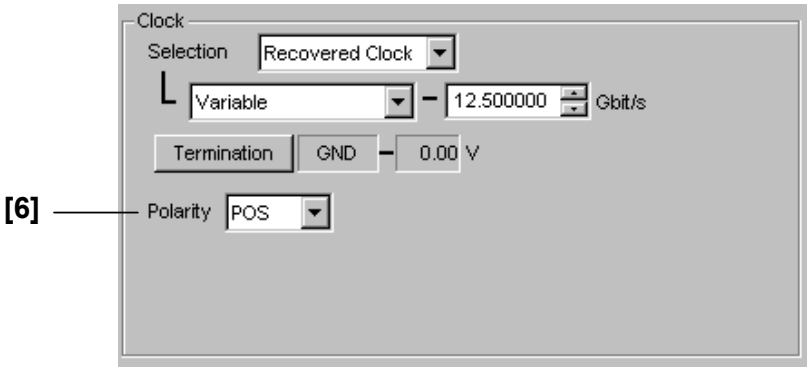


Figure 7.4.4.2-4 Clock Input setting (For no delay option)

Table 7.4.4.2-1 Clock Input setting commands

No.	Setting Items	Commands
[1]	Selection	:INPut:CLOCk:SELection
		:INPut:CLOCk:SELection?
[2]	Setting for specified frequency of Clock Recovery	:INPut:CLOCk:RECovery
		:INPut:CLOCk:RECovery?
[3]	Setting for frequency of Clock Recovery	:INPut:CLOCk:CRFReq
		:INPut:CLOCk:CRFReq?
[4]	Termination	:INPut:CLOCk:TERMination
		:INPut:CLOCk:TERMination?
[5]	No label (Termination voltage setting)	:INPut:CLOCk:TLEVel
		:INPut:CLOCk:TLEVel?
[6]	Polarity	:INPut:CLOCk:RECLOCk
		:INPut:CLOCk:RECLOCk?
[7]	Delay	:INPut:CLOCk:DELay
		:INPut:CLOCk:DELay?
[8]	Relative	:INPut:CLOCk:RELative
		:INPut:CLOCk:RELative?
[9]	No label (Setting of the Relative value)	:INPut:CLOCk:RDELay
		:INPut:CLOCk:RDELay?
[10]	Calibration	:INPut:CLOCk:CALibration
[11]	Jitter Input	:INPut:CLOCk:JINPut
		:INPut:CLOCk:JINPut?

: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.

:INPut:CLOCk:RECovery <freq>

Parameter	<freq> = <STRING PROGRAM DATA>		
	"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: <freq> = <STRING RESPONSE DATA>

"OC_3", "OC_12", "FC", "GbE", "SATA1_5", "2G_FC", "OC_48", "PCI_EX1", "OC_48_G709", "SATA3", "XAUI", "4G_FC", "PCI_EX2", "SATA6", "OC_192", "10GbE", "10G_FC", "OC_192_G975", "OC_192_G709", "10GbE_G709", "10G_FC_G709", "Variable"

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	<numeric> = <DECIMAL 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 4.900000 to 6.250000 Gbit/s /1 kbit/s Step 9.800000 to 12.500000 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/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):
	`> :INPut:CLOCk:CRFReq 0.100000`
	To set the rated frequency for the clock recovery to 9.953280 Gbit/s (when MU181040A-001 is installed):
	`> :INPut:CLOCk:CRFReq 9.953280`

Compatibility	Incompatible with existing models.

:INPut:CLOCk:CRFReq?

Response	<numeric> = <NR2 NUMERIC RESPONSE 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 4.900000 to 6.250000 Gbit/s 9.800000 to 12.500000 9.800000 to 12.500000 Gbit/s
Function	Queries the rated frequency for the clock recovery.
Example	`> :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.

: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.	

: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 –1000 to 1000 mUI
	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 the MP1632C and MP1776A.	

: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	Incompatible with existing models.	

: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	Incompatible with existing models.	

:INPut:CLOCk:RDELay <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 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	<numeric> = <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.	

7.4.5 Commands related to Capture tab

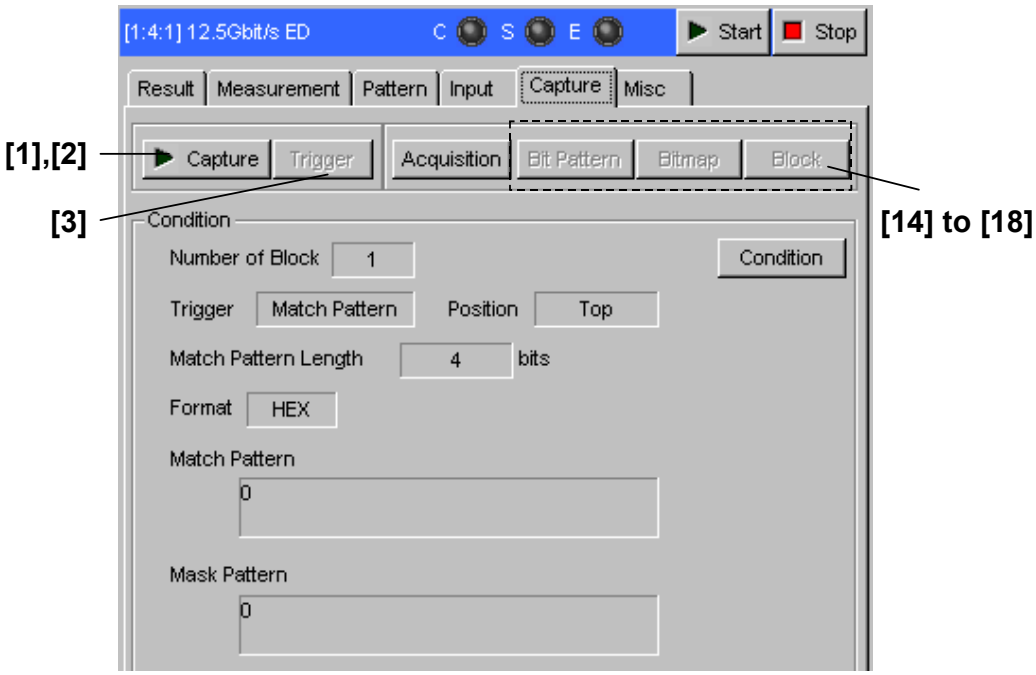


Figure 7.4.5-1 Capture tab (initial state)

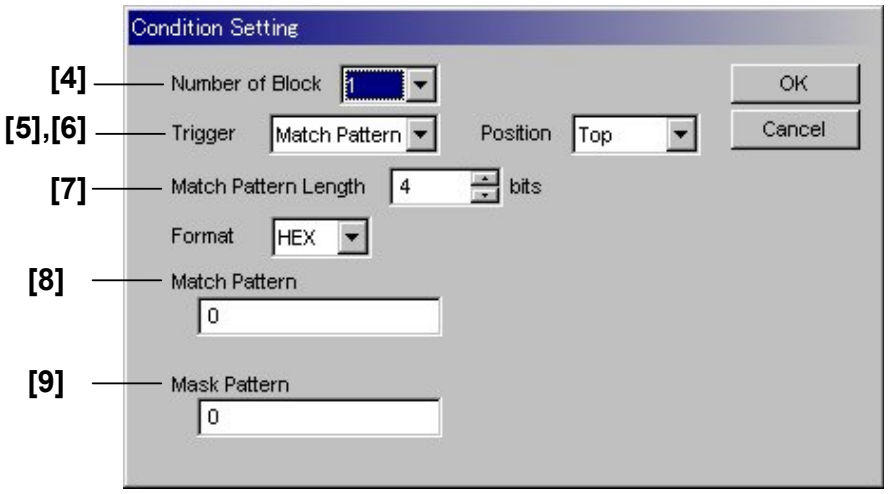


Figure 7.4.5-2 Condition Setting

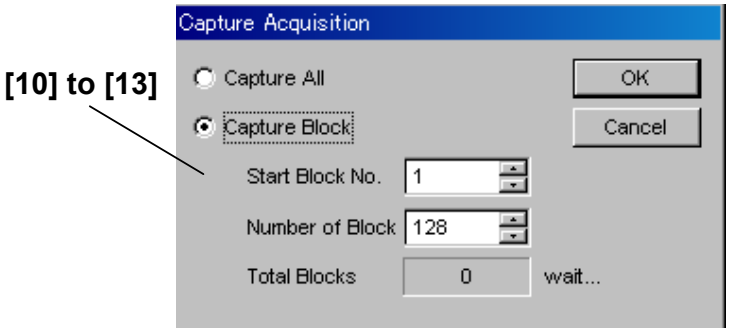


Figure 7.4.5-3 Capture Acquisition dialog box

Table 7.4.5-1 Capture tab setting commands

No.	Setting Items	Commands
[1]	Capture Start	:SENSe:CAPTure:STARt
[2]	Capture Stop	:SENSe:CAPTure:STOP
[3]	Trigger	:SENSe:CAPTure:TRIGger
[4]	Number of Block	:SENSe:CAPTure:CONDition:BLOCk
		:SENSe:CAPTure:CONDition:BLOCk?
[5]	Trigger	:SENSe:CAPTure:CONDition:TRIGger
		:SENSe:CAPTure:CONDition:TRIGger?
[6]	Position	:SENSe:CAPTure:CONDition:POSition
		:SENSe:CAPTure:CONDition:POSition?
[7]	Match Pattern Length	:SENSe:CAPTure:CONDition:MPLength
		:SENSe:CAPTure:CONDition:MPLength?
[8]	Match Pattern	:SENSe:CAPTure:CONDition:MPEDit
		:SENSe:CAPTure:CONDition:MPEDit?
[9]	Mask Pattern	:SENSe:CAPTure:CONDition:MASKedit
		:SENSe:CAPTure:CONDition:MASKedit?
[10]	Query for capture data block count	:SENSe:CAPTure:ACQuisition:BNUMber?
[11]	Start of capture data acquisition	:SENSe:CAPTure:ACQuisition:STARt
[12]	Query for capture data acquisition state	:SENSe:CAPTure:ACQuisition:STATe?
[13]	Cancel of capture data acquisition	:SENSe:CAPTure:ACQuisition:CANCel
[14]	Query for capture data length per block	:SENSe:CAPTure:BPATtern:LENGth?
[15]	Query for capture result data	:SENSe:CAPTure:BPATtern:DATA:WHOLe?
[16]	Query for capture result data	:SENSe:CAPTure:BPATtern:BDATa:WHOLe?
[17]	Query for capture result error position	:SENSe:CAPTure:BPATtern:ERRor:WHOLe?
[18]	Query for capture result error position	:SENSe:CAPTure:BPATtern:BERRor:WHOLe?

: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?

Response	<trigger> = <CHARACTER RESPONSE DATA> EDET, PATT, MAN, EXT
Function	Queries the trigger to store the data to memory for the capture function.
Example	> :SENSe:CAPTure:CONDition:TRIGger? < MAN
Compatibility	Incompatible with existing models.

: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	Incompatible with existing models.	

: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	Incompatible with existing models.

: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	Incompatible with existing models.	

: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	Incompatible with existing models.

: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	Incompatible with existing models.

: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	Incompatible with existing models.

: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	Incompatible with existing models.

: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	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.

:SENSe:CAPTure:ACQuisition:STATe?

Response	<state> = <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	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?

Response	<length> = <NR1 NUMERIC RESPONSE DATA> 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:*** At 2 ch Combination, the length of capture data is doubled. At 4 ch Combination, the length of capture date 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>

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 7FFFFF 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 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.

: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 7FFFFF 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 7FFFFF 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
< "H001"
```

Compatibility

Incompatible with existing models.

: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.

7.4.6 Commands related to Misc tab

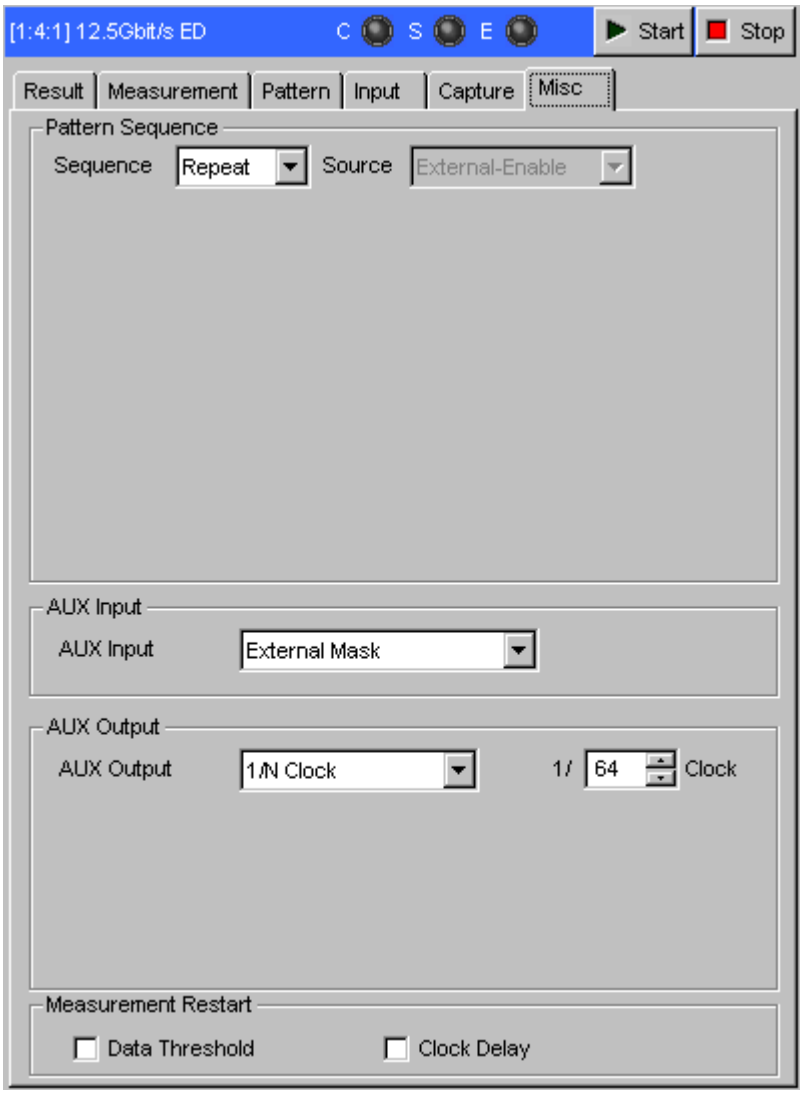


Figure 7.4.6-1 Misc tab

7.4.6.1 Pattern Sequence setting commands

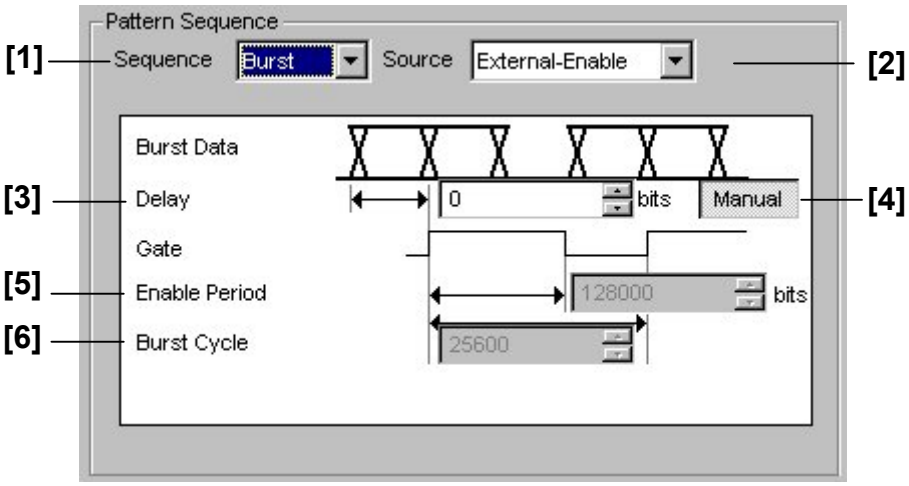


Figure 7.4.6.1-1 Pattern Sequence setting field

Table 7.4.6.1-2 Pattern Sequence setting commands

No.	Setting Items	Commands
[1]	Sequence	:SENSe:PATTern:IMODe
		:SENSe:PATTern:IMODe?
[2]	Source	:SENSe:PATTern:BURSt:MODE
		:SENSe:PATTern:BURSt:MODE?
[3]	Delay	:SENSe:PATTern:BURSt:DELay
		:SENSe:PATTern:BURSt:DELay?
[4]	Delay Manual/Auto	:SENSe:PATTern:BURSt:ADJust
[5]	Enable Period	:SENSe:PATTern:BURSt:ELENgth
		:SENSe:PATTern:BURSt:ELENgth?
[6]	Burst Cycle	:SENSe:PATTern:BURSt:CYCLe
		:SENSe:PATTern:BURSt:CYCLe?

: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 the MP1632C and MP1776A.

:SENSe:PATTern:IMODe?

Response	<mode> = <CHARACTER RESPONSE DATA> REP, BURS
Function	Queries the signal reception method.
Example	`> :SENSe:PATTern:IMODe?` `< REP`
Compatibility	Compatible with the MP1632C and MP1776A.

:SENSe:PATTern:BURSt:MODE <mode>

Parameter	<mode> = <CHARACTER PROGRAM DATA> INTernal Internal EXTTrig 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	Incompatible with existing models.

: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	Incompatible with existing models.

:SENSe:PATTern:BURSt:DELay <numeric>

Parameter	<numeric> = <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 <numeric> when the Burst Cycle is invalid. ***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.
Example	To set the delay value for the Burst signal to 0 bits: `> :SENSe:PATTern:BURSt:DELay 0`
Compatibility	Incompatible with existing models.

: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	Incompatible with existing models.

: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	Incompatible with existing models.

:SENSe:PATTern:BURSt:ADJust?

Response	<adjust> = <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.

7.4.6.2 AUX Output setting commands

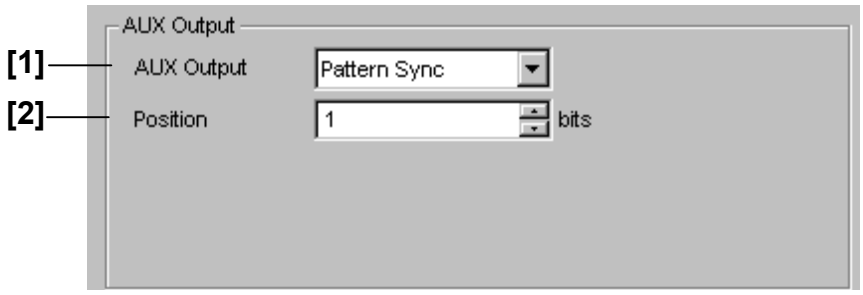


Figure 7.4.6.2-1 Pattern Sync setting (For PRBS, Zero Substitution, and Data patterns)

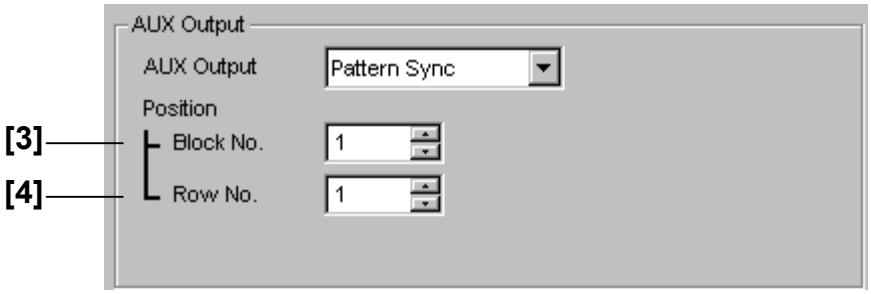


Figure 7.4.6.2-2 Pattern Sync setting (For Mixed pattern)

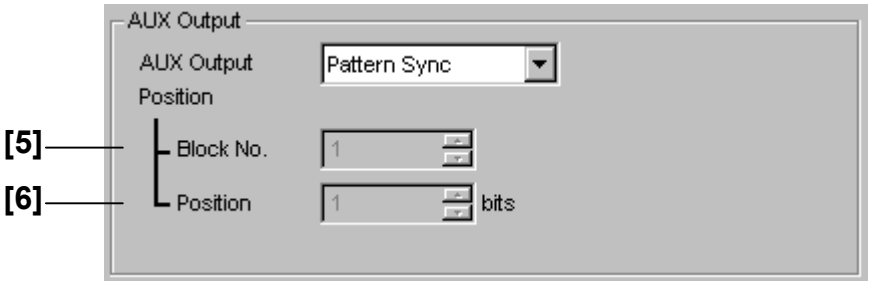


Figure 7.4.6.2-3 Pattern Sync setting (For Sequence pattern)

Table 7.4.6.2-1 AUX Output setting commands

No.	Setting Items	Commands
[1]	AUX Output	:OUTPut:SYNC:SOURce
		:OUTPut:SYNC:SOURce?
[2]	Position (PRBS, Zero Substitution, Data)	:OUTPut:SYNC:POSition
		:OUTPut:SYNC:POSition?
[3]	Block No. (Mixed Data)	:OUTPut:SYNC:MIXData:BLOCk
		:OUTPut:SYNC:MIXData:BLOCk?
[4]	Row No. (Mixed Data)	:OUTPut:SYNC:MIXData:ROW
		:OUTPut:SYNC:MIXData:ROW?
[5]	Block No. (Sequence)	:OUTPut:SYNC:SEQuence:BLOCk
		:OUTPut:SYNC:SEQuence:BLOCk?
[6]	Position (Sequence)	:OUTPut:SYNC:SEQuence:POSition
		:OUTPut:SYNC:SEQuence:POSition?

:OUTPut:SYNC:SOURce <source>[,<numeric>]

Parameter

<source> = <CHARACTER PROGRAM DATA>

CLOCk8	1/8 clocks (for MU181040A-002, MU181040B-002)
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.

: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.

: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: `> :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	`> :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: `> :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	`> :OUTPut:SYNC:SEQuence:POSition?` `< 1`
Compatibility	Incompatible with existing models.

7.4.6.3 AUX Input setting commands

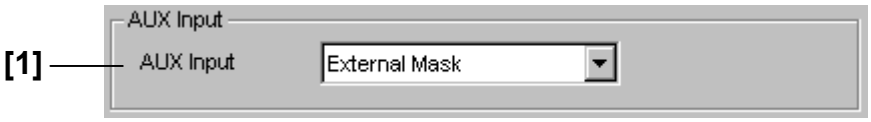


Figure 7.4.6.3-1 Auto Input setting field

Table 7.4.6.3-1 Auto Input setting commands

No.	Setting Item	Commands
[1]	AUX Input	:INPut:AUX:USAGe
		:INPut:AUX:USAGe?

: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

Figure 7.4.6.4-1 Measurement Restart setting field

Table 7.4.6.4-1 Measurement Restart setting commands

No.	Setting Item	Commands
[1]	Measurement Restart	:SENSe:MEASure:MREStart
		:SENSe:MEASure:MREStart?

: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).

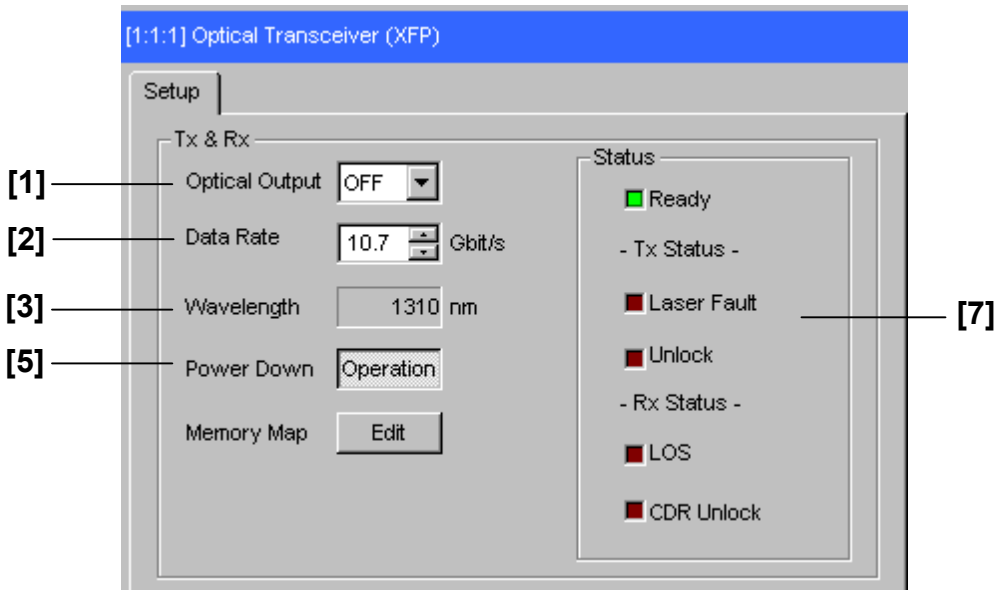


Figure 7.5.1-1 XFP screen

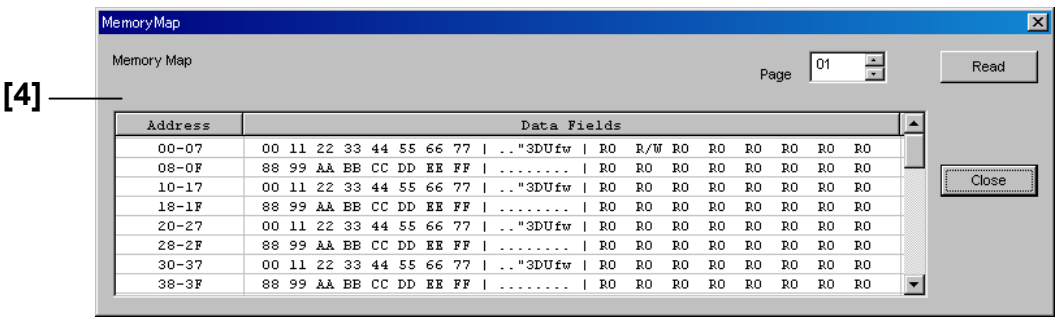


Figure 7.5.1-2 Memory Map setting (XFP)

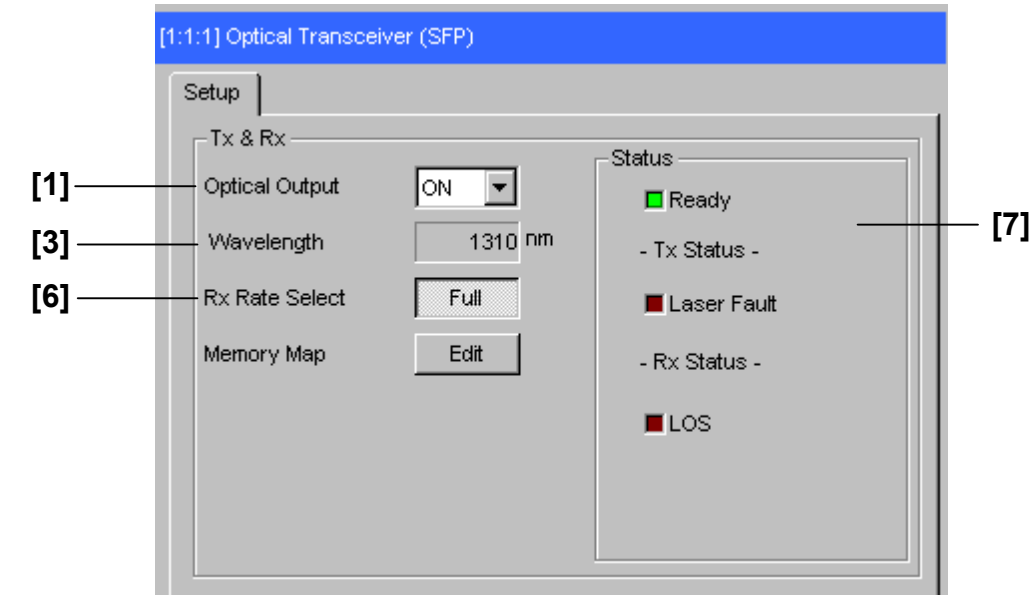


Figure 7.5.1-3 SFP screen

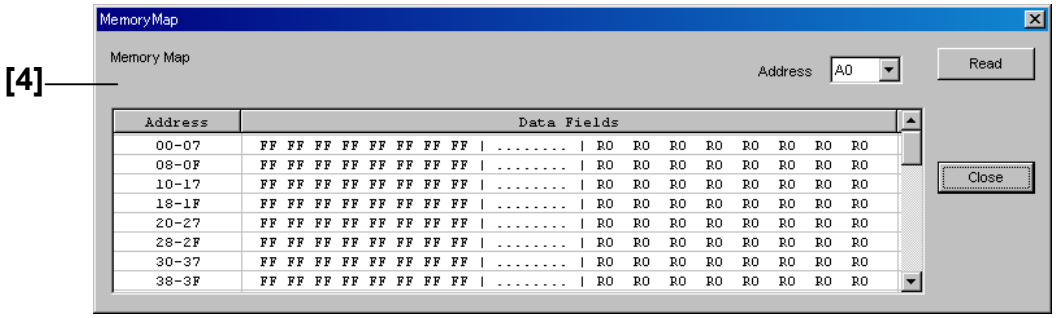


Figure 7.5.1-4 Memory Map setting (SFP)

Table 7.5.1-1 XFP/SFP module

No.	Setting Items	Commands
[1]	Optical Output	:SOURce:OPTical:SIGNal:OUTPut
		:SOURce:OPTical:SIGNal:OUTPut?
[2]	Data Rate	:SOURce:OPTical:SIGNal:DRATe
		:SOURce:OPTical:SIGNal:DRATe?
[3]	Wavelength	:SOURce:OPTical:SIGNal:WLENgth?
[4]	Memory Map	:SENSe:OPTical:SIGNal:MEMory:DATA
		:SENSe:OPTical:SIGNal:MEMory:ASCii?
		:SENSe:OPTical:SIGNal:MEMory:HEX?
[5]	Power Down	:SENSe:OPTical:POWerdown
		:SENSe:OPTical:POWerdown?
[6]	RX Rate Select	:SENSe:OPTical:RATeselect
		:SENSe:OPTical:RATeselect?
[7]	Status	:CALCulate:OPTical:STATus?

: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.

: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

<page> = <NON-DECIMAL NUMERIC PROGRAM DATA>
For the XFP module:
P00 to PFF 00 to FF (Hexadecimal)
For SFP module:
A0/A2
<address> = <CHARACTER PROGRAM DATA>
A00 to AFF 00 to FF
<data> = <STRING PROGRAM DATA>
"00" to "FF" 00 to FF
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:

```
> :SENSe:OPTical:SIGNal:MEMory:DATA P00,A00,"FF,FF,FF,FF"
```

Compatibility

Incompatible with existing models.

:SENSe:OPTical:SIGNal:MEMory:ASCii? <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>
A character string other than 0x20-0x7e is output replaced by '?.'
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 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:

```
> :SENSe:OPTical:SIGNal:MEMory:HEX? P00,A00,4
< "FF,FF,FF,FF"
```

Compatibility

Incompatible with existing models.

:CALCulate:OPTical:STATus? <string>

Parameter

<string> = <STRING PROGRAM DATA>

"READy"	Ready
"TX:LASer"	Tx: Laser
"TX:UNLock"	Tx: Unlock (Cannot be set for the SFP module (MU181601A).)
"RX:LOS"	Rx: LOS
"RX:CDR"	Rx: CDR unlock (Cannot be set for the SFP module (MU181601A).)

Response

<string> = <STRING RESPONSE DATA>

Table 7.5.1-2 Response format

Type	Format	Description
Form1	"Occur"	When the specified status occurs.
	"None"	When the specified status does not occur.
	"-----"	When no data corresponds to the query

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

<power> = <CHARACTER PROGRAM DATA>

STANdby	Standby mode
OPERation	Normal operation mode

Function

Sets the XFP module to the standby mode.

Example

To set the XFP module to the standby mode:

```
> :SOURce:OPTical:POWerdown STANdby
```

Compatibility

Incompatible with existing models.

:SOURce:OPTical:POWerdown?

Response	<power> = <CHARACTER RESPONSE DATA> 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?`
	`< STANdby`
	`< "----"` (when no XFP module is inserted)

Compatibility	Incompatible with existing models.

:SOURce:OPTical:RATeselect <ratesel>

Parameter	<ratesel> = <CHARACTER PROGRAM DATA> REDuced Enables band restriction FULL Disables 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	<ratesel> = <CHARACTER RESPONSE DATA> 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

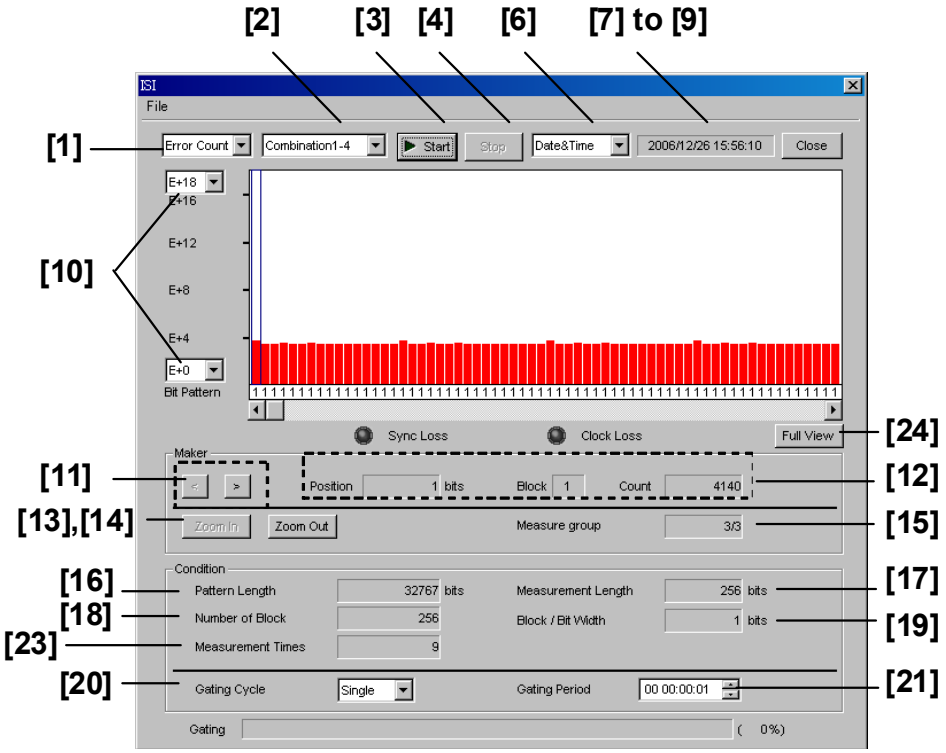


Figure 7.6.1.1-1 ISI measurement window

Table 7.6.1.1-1 ISI measurement setting commands

No.	Setting Items/Read Items	Commands
[1]	Setting item selection	:DISPlay:RESult:ISI:ITEM
		:DISPlay:RESult:ISI:ITEM?
[2]	Slot selection	:SENSe:MEASure:ISI:SELSlot
		:SENSe:MEASure:ISI:SELSlot?
[3]	Start	:SENSe:MEASure:STARt
[4]	Stop	:SENSe:MEASure:ISI:STOP
[5]	No label (Query for measurement state)	:SENSe:MEASure:ISI:STATe?

Table 7.6.1.1-1 ISI measurement setting commands (Cont'd)

No.	Setting Items/Read Items	Commands
[6]	Measurement time display selection	:DISPlay:RESult:ISI:TIME
		:DISPlay:RESult:ISI:TIME?
[7]	Query for Date&Time	:SENSe:MEASure:ISI:DTIMe?
[8]	Query for measurement start time	:SENSe:MEASure:ISI:STARt?
[9]	Query for measurement elapsed time	:SENSe:MEASure:ISI:ELAPsed?
[10]	Scale setting	:DISPlay:RESult:ISI:SCALe
		:DISPlay:RESult:ISI:SCALe?
[11]	Marker movement	:DISPlay:RESult:ISI:MMOVe
[12]	Query for marker position	:DISPlay:RESult:ISI:MARKer?
[13]	Zoom In	:DISPlay:RESult:ISI:ZIN
[14]	Zoom Out	:DISPlay:RESult:ISI:ZOUT
[15]	Query for measurement layer	:DISPlay:RESult:ISI:DEPTh?
[16]	Pattern Length	:DISPlay:RESult:ISI:PLENgth?
[17]	Measurement Length	:DISPlay:RESult:ISI:MLENgth?
[18]	Number of Block	:DISPlay:RESult:ISI:BLOCk?
[19]	Block/Bit Width	:DISPlay:RESult:ISI:BWIDth?
[20]	Gating Cycle	:SENSe:MEASure:ISI:MODE
		:SENSe:MEASure:ISI:MODE?
[21]	Gating Period	:SENSe:MEASure:ISI:PERiod
		:SENSe:MEASure:ISI:PERiod?
[22]	No label (Query for measurement result)	:CALCulate:DATA:ISI?
[23]	Measurement Times	:DISPlay:RESult:ISI:MTIMes?
[24]	Full View	:DISPlay:RESult:ISI:FVIew
		:DISPlay:RESult:ISI:FVIew?

: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:

```
> :DISPlay:RESult:ISI:ITEM COUNt
```

Compatibility
Incompatible with existing models.

: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.

: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.

: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.

:SENSe:MEASure:ISI: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 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

<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)
<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?

Response	<top>,<bottom> = <CHARACTER RESPONSE DATA>	
	E_18 to E_0	E–18 to E+0, in single steps (When the measurement item is Rate)
	0, E_0 to E_18	0, E+0 to E+18, in single steps (When the measurement item is Count)
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>

Parameter	<move> = <CHARACTER PROGRAM DATA>	
	LEFT	Moves to the left
	RIGHt	Moves to the right
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>
Form1
<rate> = <STRING RESPONSE DATA>
Form2

Table 7.6.1.1-2 Response Format

Form	Format	Description
Form1 Integer type	"XXXXXXX"	When 0 to 9999999
	"X.XXXXEXX"	When 1.0000E07 to 9.9999E17
	"-------"	When no data corresponds to the query.
Form2 Fraction type	"X.XXXXE-XX"	When 0.0001E–18 to 1.0000E00
	"-------"	When no data corresponds to the query.

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.

: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.

:DISPlay:RESult:ISI:BLOCk?

Response	<block> = <NR1 NUMERIC RESPONSE DATA>
	2 to 256 2 to 256 blocks
Function	Queries the ISI measurement block division number.
Example	> :DISPlay:RESult:ISI:BLOCk?
	< 64
Compatibility	Incompatible with existing models.

:DISPlay:RESult:ISI:BWIDth?

Response	<width> = <NR1 NUMERIC RESPONSE DATA>
	1 to 285212672 1 to 285,212,672 bits
	Maximum value: Maximum length of Mixed Data pattern/64
Function	Queries the number of bits within one block in ISI measurement.
Example	> :DISPlay:RESult:ISI:BWIDth?
	< 2097152
Compatibility	Incompatible with existing models.

:SENSe:MEASure:ISI:MODE <mode>

Parameter	<mode> = <CHARACTER PROGRAM DATA>
	SINGle Single measurement
	UNTimed Untimed measurement
Function	Selects the measurement processing mode.
Example	To set the measurement processing mode to Single measurement:
	> :SENSe:MEASure:ISI:MODE SINGle
Compatibility	Incompatible with existing models.

:SENSe:MEASure:ISI:MODE?

Response	<mode> = <CHARACTER RESPONSE DATA>
	SING, UNT
Function	Queries the synchronization output type.
Example	> :SENSe:MEASure:ISI:MODE?
	< SING
Compatibility	Incompatible with existing models.

: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:
	`> :SENSe:MEASure:ISI:PERiod 99,23,59,59`
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
<pos> = <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
<count> = <STRING RESPONSE DATA>
Form1
<rate> = <STRING RESPONSE DATA>
Form2

Table 7.6.1.1-3 Response Format

Form	Format	Description
Form1 Integer type	"XXXXXXX"	When 0 to 9999999
	"X.XXXXEXX"	When 1.0000E07 to 9.9999E17
	"-------"	When no data corresponds to the query.
Form2 Fraction type	"X.XXXXE-XX"	When 0.0001E–18 to 1.0000E00
	"-------"	When no data corresponds to the query.

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
<numeric> = <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
<boolean> = <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:FVIew?

Response	<numeric> = <NR1 NUMERIC RESPONSE DATA>
	1 Full View mode.
	0 Normal view mode.
Function	Queries the graph display mode during ISI measurement.
Example	`> :DISPlay:RESult:ISI:FVIew?`
	`< 1`
Compatibility	Incompatible with existing models.

7.6.1.2 File menu setting commands

Table 7.6.1.2-1 File menu setting commands

No.	Setting Items	Commands
[1]	Open	:SYSTem:MMEMory:ISI:RECall
[2]	Save	:SYSTem:MMEMory:ISI:STORe
[3]	Print	:SYSTem:PRINt:ISI

: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>

BIN	Binary file
CSV	CSV file
TXT	Text file

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>

Parameter	<list> = <CHARACTER PROGRAM DATA> ISI ISI Result
Function	Prints the ISI measurement result data.
Example	`> :SYSTem:PRINt:ISI ISI`
Compatibility	Incompatible with existing models.

7.6.2 Eye Margin measurement

This section describes the commands for Eye Margin measurement.

7.6.2.1 Eye Margin measurement setting commands

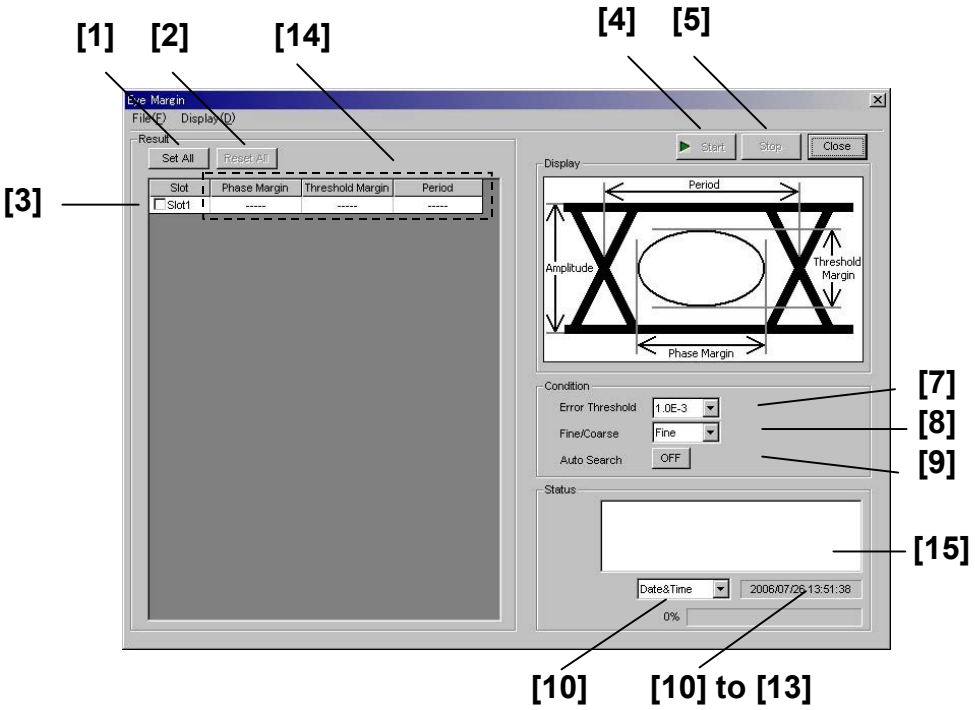


Figure 7.6.2.1-1 Eye Margin measurement screen

Table 7.6.2.1-1 Eye Margin measurement setting commands

No.	Setting Items	Commands
[1]	Set All	:SENSe:MEASure:EMARgin:MARGin:ASLot
[2]	Reset All	:SENSe:MEASure:EMARgin:MARGin:RSLot
[3]	Slot	:SENSe:MEASure:EMARgin:MARGin:SLOT
		:SENSe:MEASure:EMARgin:MARGin:SLOT?
[4]	Start	:SENSe:MEASure:STARt
[5]	Stop	:SENSe:MEASure:STOP
[6]	No label (Query for measurement state)	:SENSe:MEASure:EMARgin:STATe?
[7]	Error Threshold	:SENSe:MEASure:EMARgin:MARGin:THReshold
		:SENSe:MEASure:EMARgin:MARGin:THReshold?
[8]	Fine/Coarse	:SENSe:MEASure:EMARgin:MARGin:RESolution
		:SENSe:MEASure:EMARgin:MARGin:RESolution?
[9]	Auto Search	:SENSe:MEASure:EMARgin:MARGin:ASEarch
		:SENSe:MEASure:EMARgin:MARGin:ASEarch?
[10]	Selection of measurement time display	:SENSe:MEASure:EMARgin:TIME
		:SENSe:MEASure:EMARgin:TIME?
[11]	Query for date and time	:SENSe:MEASure:EMARgin:DTIMe?

Table 7.6.2.1-1 Eye Margin measurement setting commands (Cont'd)

No.	Setting Items	Commands
[12]	Query for measurement start time	:SENSe:MEASure:EMARgin:STARt?
[13]	Query for measurement elapsed time	:SENSe:MEASure:EMARgin:ELAPsed?
[14]	Query for measurement result	:CALCulate:DATA:EMARgin?
[15]	No label (Query for measurement status)	:CALCulate:DATA:EMARgin:STATus?

: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.

:SENSe:MEASure:EMARgin:MARGin: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 Eye Margin measurement.

Example
To query whether the slot 1 is targeted for the Eye Margin measurement:

```
> :SENSe:MEASure:EMARgin:MARGin:SLOT? SLOT1
< 1
```

Compatibility
Incompatible with existing models.

: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>

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.

:SENSe:MEASure:EMARgin:MARGin:ASEarch <boolean>[,<mode>]

Parameter

<boolean> = <BOOLEAN PROGRAM DATA>

ON or 1	Auto Search ON
OFF or 0	Auto Search OFF

<mode> = <DECIMAL NUMERIC PROGRAM DATA>

1	Coarse
2	Fine

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?

Response

<boolean> = <NR1 NUMERIC RESPONSE DATA>

1	ON (Coarse)
0	OFF
2	ON (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 STARt
```

Compatibility

Compatible with the MP1632C.

:SENSe:MEASure:EMARgin:TIME?

Response	<type> = <CHARACTER RESPONSE DATA>	
	DTIM	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	<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 Eye Margin measurement.	
Example	`> :SENSe:MEASure:EMARgin:DTIMe?`	
	`< 2006,4,1,23,59,59`	
Compatibility	Incompatible with existing models.	

:SENSe:MEASure:EMARgin:STARt?

Response

<year> = <NR1 NUMERIC RESPONSE DATA>
0, 2000 to 2036 2000 to 2036 year
<month> = <NR1 NUMERIC RESPONSE DATA>
0 to 12 0 to December
<day> = <NR1 NUMERIC RESPONSE DATA>
0 to 31 0 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 Eye Margin measurement start time.

Example

```
> :SENSe:MEASure:EMARgin:STARt?
< 2006,4,1,23,59,59
```

Compatibility

Compatible with the MP1632C.

: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>,<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 7.6.2.1-2 Eye Margin measurement items <string>

Items	<string>	Response form
Phase margin	"PHASe"	Form1
Threshold margin	"THReshold"	Form2
Period	"PERiod"	Form1

Response

<result> = <STRING RESPONSE DATA>
For the contents of the <result> measurement data see Table 7.6.2.1-3.

Table 7.6.2.1-3 Eye Margin measurement results <result>

Form	Format	Description
Form1 Phase type	" XXXX"	For 0 to MAX (mUI/ps) Displays in the selected unit.
	"Failed"	For measurement failure
	"------"	When no data corresponds to the query.
Form2 Threshold type	" XXXX"	0 to MAX (mVpp)
	"Failed"	For measurement failure
	"------"	When no data corresponds to the query.

Function

Queries the Eye Margin measurement result.

Example

```
> :CALCulate:DATA:EMARgin? SLOT1, "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
"CR Unlock"	CR Unlock
"Out of range"	Out of range
"Frequency NG"	Frequency NG
"Illegal Error"	Illegal Error

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 or 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

No.	Setting Items	Commands
[1]	Open	:SYSTem:MMEMory:MARGin:RECall
[2]	Save	:SYSTem:MMEMory:MARGin:STORe
[3]	Print	:SYSTem:PRINt:MARGin

: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.

:SYSTem:PRINt:MARGin <list>

Parameter
<list> = <CHARACTER PROGRAM DATA>

EMG	Eye Margin Result

Function
Prints the Eye Margin measurement result data.

Example
```
> :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.

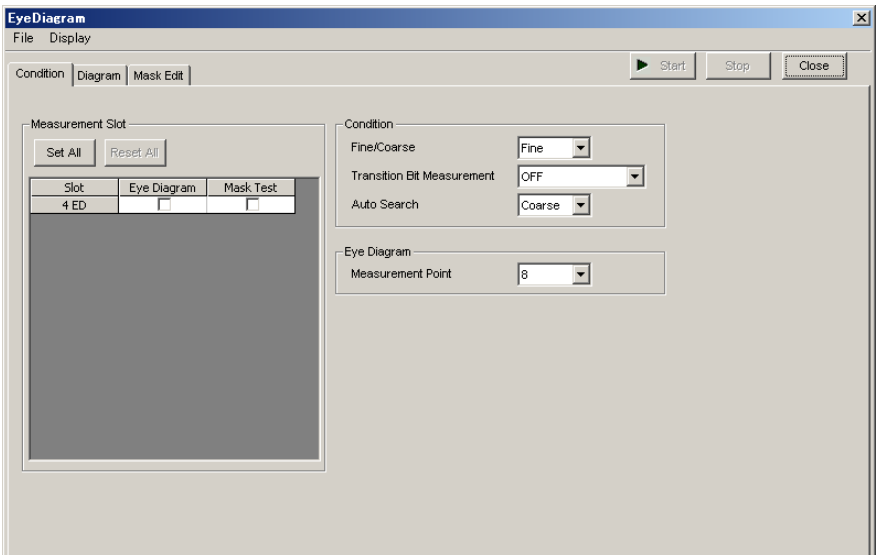


Figure 7.6.3-1 Eye Diagram measurement screen

7.6.3.1 Condition tab

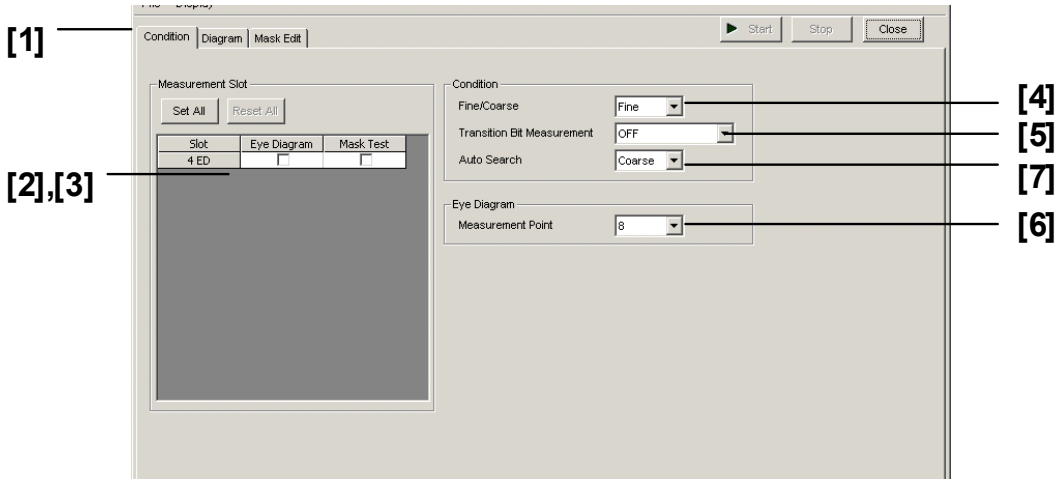


Figure 7.6.3.1-1 Condition tab

Table 7.6.3.1-1 Condition tab setting commands

No.	Setting items	Commands
[1]	Tab Selection	:DISPlay:RESult:EDIagram:TABSelect
		:DISPlay:RESult:EDIagram:TABSelect?
[2]	Slot Eye Diagram	:SENSe:MEASure:EDIagram:DIAGram:SLOT
		:SENSe:MEASure:EDIagram:DIAGram:SLOT?
[3]	Slot Mask Test	:SENSe:MEASure:EDIagram:MTESt:SLOT
		:SENSe:MEASure:EDIagram:MTESt:SLOT?
[4]	Fine/Coarse	:SENSe:MEASure:EDIagram:DIAGram:RESolution
		:SENSe:MEASure:EDIagram:DIAGram:RESolution?
[5]	Transition Bit Measurement	:SENSe:MEASure:EDIagram:DIAGram:TRANsition
		:SENSe:MEASure:EDIagram:DIAGram:TRANsition?
[6]	Measurement Point	:SENSe:MEASure:EDIagram:DIAGram:TYPE
		:SENSe:MEASure:EDIagram:DIAGram:TYPE?
[7]	Auto Search	:SENSe:MEASure:EDIagram:DIAGram:ASEarch
		:SENSe:MEASure:EDIagram:DIAGram:ASEarch?

: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.

: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.

: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.

:SENSe:MEASure:EDIagram:DIAGram:TRANsition <type>

Parameter	<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:	
	`> :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	`> :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:	
	`> :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	`> :SENSe:MEASure:EDIagram:DIAGram:TYPE?`
	`< 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
	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>
	2 ON (Fine)
	1 ON (Coarse)
	0 OFF
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

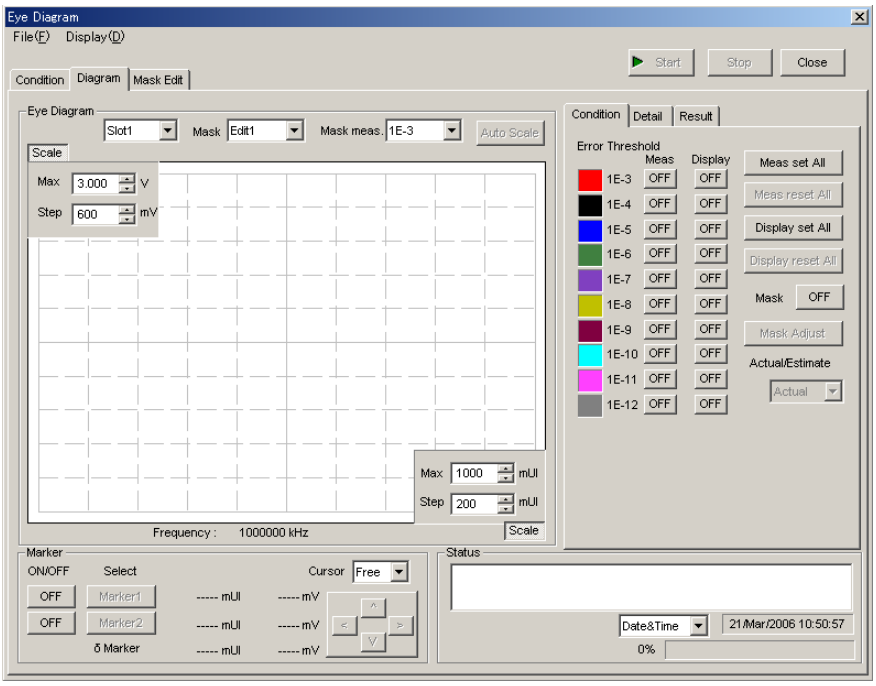


Figure 7.6.3.2-1 Diagram tab

7.6.3.2.1 Graph display setting commands

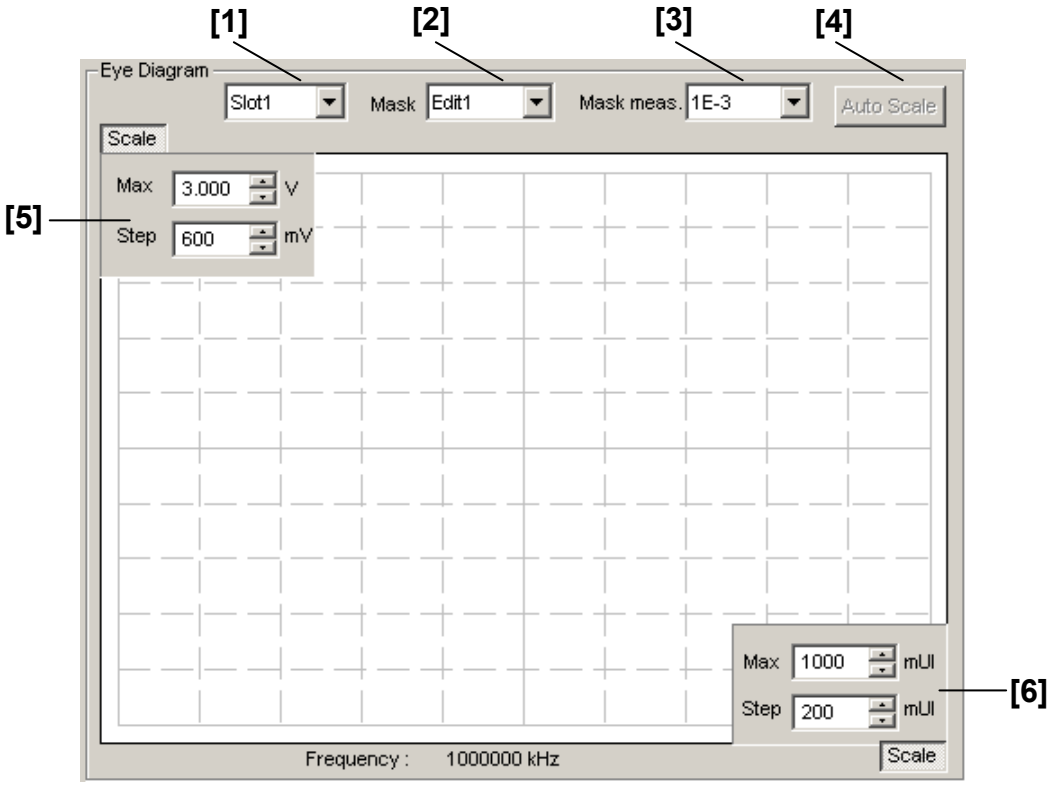


Figure 7.6.3.2.1-1 Graph display field

Table 7.6.3.2.1-1 Graph display setting commands

No.	Setting Items	Commands
[1]	No label (Result display slot setting)	:DISPlay:RESult:EDIagram:SLOT
		:DISPlay:RESult:EDIagram:SLOT?
[2]	Mask	:DISPlay:RESult:EDIagram:TEMPlate:SELect
		:DISPlay:RESult:EDIagram:TEMPlate:SELect?
[3]	Mask meas.	:DISPlay:RESult:EDIagram:TEMPlate:THReshold
		:DISPlay:RESult:EDIagram:TEMPlate:THReshold?
[4]	Auto Search	:DISPlay:RESult:EDIagram:SCALe:ASCale
[5]	Max/Step	:DISPlay:RESult:EDIagram:SCALe:VOLTage
		:DISPlay:RESult:EDIagram:SCALe:VOLTage?
[6]	Max/Step	:DISPlay:RESult:EDIagram:SCALe:PHASe
		:DISPlay:RESult:EDIagram:SCALe:PHASe?

: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:

```
> :DISPlay:RESult:EDIagram:SLOT SLOT1
```

Compatibility
Incompatible with existing models.

:DISPlay:RESult:EDIagram: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 Slots No. 1 to No. 6

Function
Queries the Eye Diagram result display target slot.

Example

```
> :DISPlay:RESult:EDIagram:SLOT?
< SLOT1
```

Compatibility
Incompatible with existing models.

:DISPlay:RESult:EDIagram:TEMPlate:SELect <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:TEMPlate:SELect E2`
Compatibility	Incompatible with existing models.

:DISPlay:RESult:EDIagram:TEMPlate:SELect?

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:TEMPlate:SELect?` `< E2`
Compatibility	Incompatible with existing models.

:DISPlay:RESult:EDIagram:TEMPlate: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:TEMPlate:THReshold E_3`
Compatibility	Incompatible with existing models.

:DISPlay:RESult:EDIagram:TEMPlate:THReshold?

Response	<thre> = <CHARACTER RESPONSE DATA>
Function	Queries the Error Threshold for performing the Eye Diagram mask test.
Example	`> :DISPlay:RESult:EDIagram:TEMPlate:THReshold?` `< E_3`
Compatibility	Incompatible with existing models.

: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 –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 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 –3.990 to 4.000 V <step> = <NR2 NUMERIC RESPONSE DATA> 0.001 to 0.800 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.

:DISPlay:RESult:EDIagram:SCALe:PHASe <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 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.

7.6.3.2.2 Condition setting commands

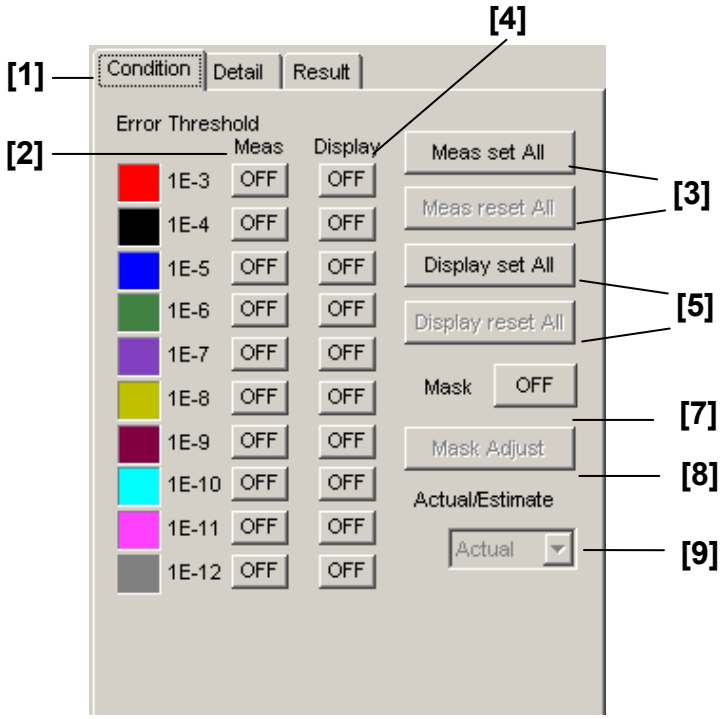


Figure 7.6.3.2.2-1 Condition setting screen (Actual)

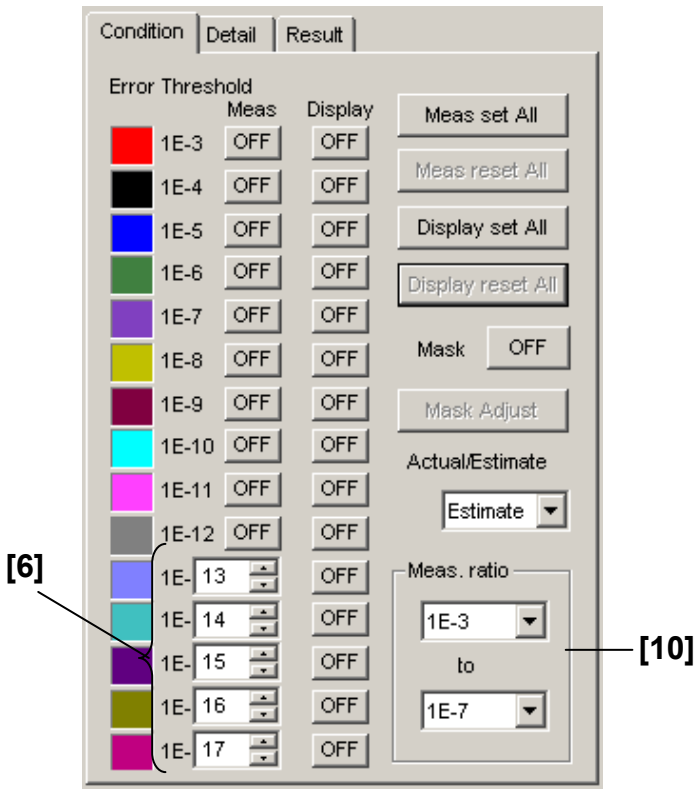


Figure 7.6.3.2.2-2 Condition setting screen (Estimate)

Table 7.6.3.2.2-1 Graph display setting commands

No.	Setting Items	Commands
[1]	Condition Detail Result	:DISPlay:RESult:EDIagram:DIAGram:TABSelect
		:DISPlay:RESult:EDIagram:DIAGram:TABSelect?
[2]	Meas	:SENSe:MEASure:EDIagram:DIAGram:THReshold
		:SENSe:MEASure:EDIagram:DIAGram:THReshold?
[3]	Meas set All Meas reset All	:SENSe:MEASure:EDIagram:DIAGram:ATHReshold
[4]	Display	:DISPlay:RESult:EDIagram:ERATe
		:DISPlay:RESult:EDIagram:ERATe?
[5]	Display set All Display reset All	:DISPlay:RESult:EDIagram:AERate
[6]	Setting of error rate for Estimate measurement	DISPlay:RESult:EDIagram:EERate
		DISPlay:RESult:EDIagram:EERate?
[7]	Mask	:DISPlay:RESult:EDIagram:DISPlay:MASK
		:DISPlay:RESult:EDIagram:DISPlay:MASK?
[8]	Mask Adjust	:DISPlay:RESult:EDIagram:ADJust
[9]	Actual/Estimate	:DISPlay:RESult:EDIagram:ESTimate
		:DISPlay:RESult:EDIagram:ESTimate?
[10]	Meas.ratio	:SENSe:MEASure:EDIagram:DIAGram:BTHReshold
		:SENSe:MEASure:EDIagram:DIAGram:BTHReshold?

Table 7.6.3.2.2-2 Graph display Query commands

Setting Items	Commands
Query for measurement point	:SENSe:MEASure:EDIagram:DIAGram:POINt?
Query for measurement result	:CALCulate:DATA:DIAGram?

:DISPlay:RESult:EDIagram:DIAGram:TABSelect <tab>

Parameter <tab> = <CHARACTER PROGRAM DATA>

CONDition	Condition tab
DETail	Detail tab
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 <tab> = <CHARACTER RESPONSE DATA>

COND, DET, RES

Function Queries the child tab currently active on the Diagram tab.

Example

```
> :DISPlay:RESult:EDIagram:TABSelect?
```

```
< COND
```

Compatibility: Incompatible with existing models.

:SENSe:MEASure:EDIagram:DIAGram:THReshold <thre>,<boolean>

Parameter:

<type> = <CHARACTER PROGRAM DATA>

E_3 to E_12	1E–3 to 1E–12, in single steps

<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

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:

<type> = <CHARACTER PROGRAM DATA>

E_3 to E_17	1E–3 to 1E–17, in single steps

<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.

: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
Function	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.

:DISPlay:RESult:EDIagram:ADJust

Function	Automatically adjusts the mask pattern during Eye Diagram measurement.
Example	`> :DISPlay:RESult:EDIagram:ADJust`
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 EST Estimate
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, in single step
<lower> = <CHARACTER PROGRAM DATA>
E_7 to E_12 1E–7 to E–12, in single 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.

: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.

: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:
	`> :SENSe:MEASure:EDIagram:DIAGram:POINt? SLOT1,E_3`
	`< 1,2,3,4,132`
Compatibility	Incompatible with existing models.

: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 7.6.3.2.2-3 Response format

Form	Format	Description
Form1 Eye Map type	"XXXXX,YYYYY"	XXXXX: Phase (mUI) XXXXX: Phase (ps) (Displays in the selected unit.) YYYYY: Threshold (mV)
	"-----,--.---"	When no data corresponds to the query.

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

Items		<Result>	Response Format
Eye Map type	1E-3 1E-4 | 1E-11 1E-12	"MAP:E_3:POINt1",...,"MAP:E_3:POINt132" "MAP:E_4:POINt1",...,"MAP:E_4:POINt132" | "MAP:E_11:POINt1",...,"MAP:E_11:POINt132" "MAP:E_12:POINt1",...,"MAP:E_12:POINt132"	Form1

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

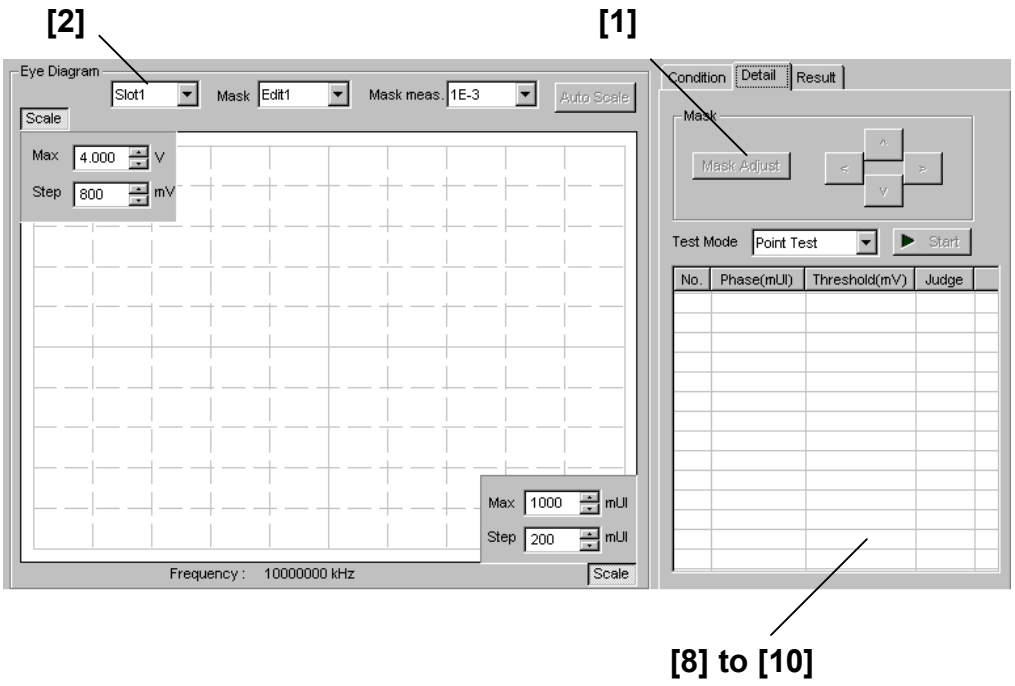


Figure 7.6.3.2.3-1 Detail Mask Point Test screen

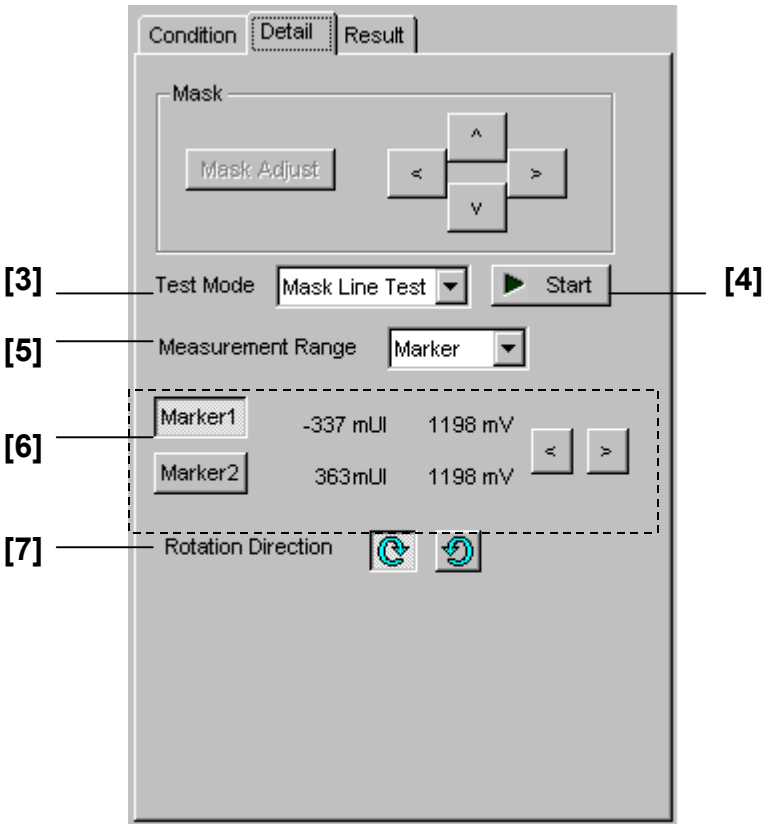


Figure 7.6.3.2.3-2 Detail Mask Line Test screen

Table 7.6.3.2.3-1 Detail setting commands

No.	Setting Items	Commands
[1]	Mask Adjust	:SENSe:MEASure:EDIagram:DETail:MASK
[2]	Slot selection	:SENSe:MEASure:EDIagram:DETail:SLOT
		:SENSe:MEASure:EDIagram:DETail:SLOT?
[3]	Test Mode	:SENSe:MEASure:EDIagram:DETail:MODE
		:SENSe:MEASure:EDIagram:DETail:MODE?
[4]	Start/Stop	:SENSe:MEASure:EDIagram:DETail:STARt
		:SENSe:MEASure:EDIagram:DETail:STOP
		:SENSe:MEASure:EDIagram:DETail:STATe?
[5]	Measurement Range	:SENSe:MEASure:EDIagram:DETail:LINE:RANGe
		:SENSe:MEASure:EDIagram:DETail:LINE:RANGe?
[6]	Marker1 Marker2	:DISPlay:RESult:EDIagram:DETail:MARKer
		:DISPlay:RESult:EDIagram:DETail:MARKer?
[7]	Rotation Direction	:SENSe:MEASure:EDIagram:DETail:LINE:ROTation
		:SENSe:MEASure:EDIagram:DETail:LINE:ROTation?
[8]	Query for measurement point	:CALCulate:DATA:EDIagram:DETail:PNUMber?
[9]	Query for Point test result	:CALCulate:DATA:EDIagram:DETail:POINt?
[10]	Query for Line test result	:CALCulate:DATA:EDIagram:DETail:LINE?

:SENSe:MEASure:EDIagram:DETail:MASK

Function — Performs 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>
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 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.

: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

Function: Stops the Mask pass/fail judgment during Eye Diagram Detail measurement.

Example:
```
> :SENSe:MEASure:EDIagram:DETail:STOP
```

Compatibility: Incompatible with existing models.

:SENSe:MEASure:EDIagram:DETail:STATe?

Response: <numeric> = <NR1 NUMERIC RESPONSE DATA>

1	Start
0	Stop

Function: Queries the Mask pass/fail judgment status during Eye Diagram Detail measurement.

Example:
```
> :SENSe:MEASure:EDIagram:DETail:STATe?
< 1
```

Compatibility: Incompatible with existing models.

:SENSe:MEASure:EDIagram:DETail:LINE:RANGe <range>

Parameter: <range> = <CHARACTER PROGRAM DATA>

ALL	Whole range
MARKer	Marked range only

Function: Selects the measurement range during Eye Diagram Detail Line measurement.

Example: To set the measurement range during Eye Diagram Detail Line measurement to the marked range:
```
> :SENSe:MEASure:EDIagram:DETail:LINE:RANGe MARKer
```

Compatibility: Incompatible with existing models.

:SENSe:MEASure:EDIagram:DETail:LINE:RANGe?

Response: <range> = <CHARACTER RESPONSE DATA>
ALL, MARK

Function: Queries the measurement range during Eye Diagram Detail Line measurement.

Example:
```
> :SENSe:MEASure:EDIagram:DETail:LINE:RANGe?
< MARK
```

Compatibility: Incompatible with existing models.

: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 4000mV, 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

Note:

The converted value from mUI

<voltage> = <NR2 NUMERIC RESPONSE DATA>

–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):

```
> :DISPlay:RESult:EDIagram:DETail:MARKer? 1
< 100,1000
```

Compatibility

Incompatible with existing models.

:SENSe:MEASure:EDIagram:DETail:LINE:ROTation <type>

Parameter	<type> = <CHARACTER PROGRAM DATA> LEFT Counterclockwise RIGHt Clockwise 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?

Response	<type> = <CHARACTER RESPONSE DATA> LEFT, RIGH
Function	Queries the Mask pass/fail judgment direction during Eye Diagram Detail Line measurement.
Example	`> :SENSe:MEASure:EDIagram:DETail:LINE:ROTation?` `< RIGH`
Compatibility	Incompatible with existing models.

:CALCulate:DATA:EDIagram:DETail:PNUMber?

Response	<numeric> = <NR1 NUMERIC RESPONSE DATA> 1 to 20000 Number of Detail measurement point
Function	Queries the number of measurement points during Detail measurement.
Example	`> :CALCulate:DATA:EDIagram:DETail:PNUMber?` `< 1000`
Compatibility	Incompatible with existing models.

: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

Form	Format	Description
Form1 Eye Map type	"XXXXX,YYYYY,ZZ"	XXXXX: Phase (mUI) XXXXX: Phase (ps) ***Note:*** Displays in the specified units. YY.YYY: Threshold (mV) ZZ: OK, NG
	"-----,--.---,--"	When no data corresponds to the query.

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>

Items		<result>	Response Format
Judgment type	Point1 Point2 | Point31 Point32	"JDG:POINt1" "JDG:POINt2" | "JDG:POINt31" "JDG:POINt32"	Form1

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.

:CALCulate:DATA:EDIagram:DETail:LINE? <string>

Parameter

<string> = <STRING PROGRAM DATA>
<result> Measurement data
See Table 7.6.3.2.3-5 for the <result> contents.

Response

<string> = <STRING RESPONSE DATA>

Table 7.6.3.2.3-4 Response format

Form	Format	Description
Form1 Eye Map type	"XXXXX,YYYYY,ZZ"	XXXXX: Phase (mUI) XXXXX: Phase (ps) ***Note:*** Displays in the specified units. YYYYY: Threshold (mV) ZZ: OK, NG
	"-----,--.---,--"	When no data corresponds to the query.

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>

Items		<Result>	Response Format
Judgment type	Point1 Point2 | Point19999 Point20000	"JDG:POINt1" "JDG:POINt2" | "JDG:POINt19999" "JDG:POINt20000"	Form1

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

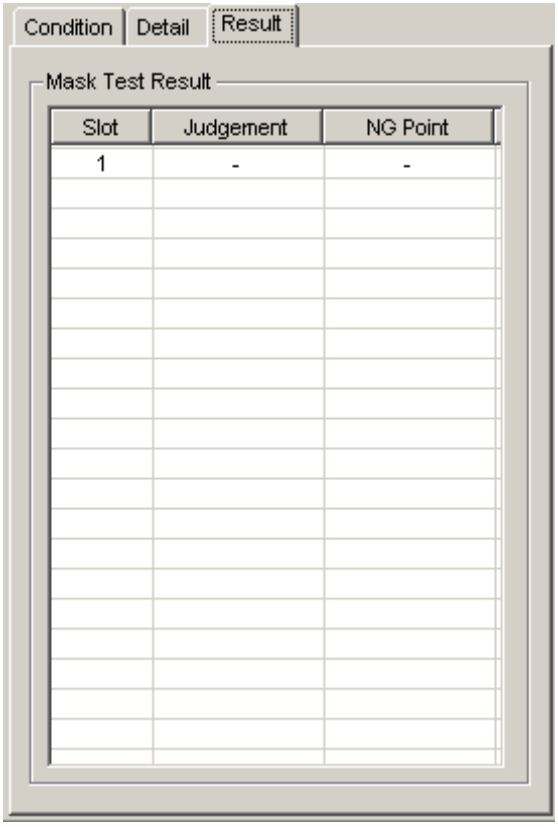


Figure 7.6.3.2.4-1 Result screen

Table 7.6.3.2.4-1 Result query command

No.	Setting Items	Commands
[1]	Query for Mask Test result	:CALCulate:DATA:EDIagram:MASK?

:CALCulate:DATA:EDIagram:MASK? <string>[,<unit>]

Parameter

<string> = <STRING PROGRAM DATA>
<result> 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

Form	Format	Description
Form1 EyeResult type	"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

Items		<Result>	Response Format
Judgment type	Slot1 Slot2 | Slot6	"SLT1" "SLT2" | "SLT6"	Form1

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.

7.6.3.2.5 Diagram tab (Status) setting commands

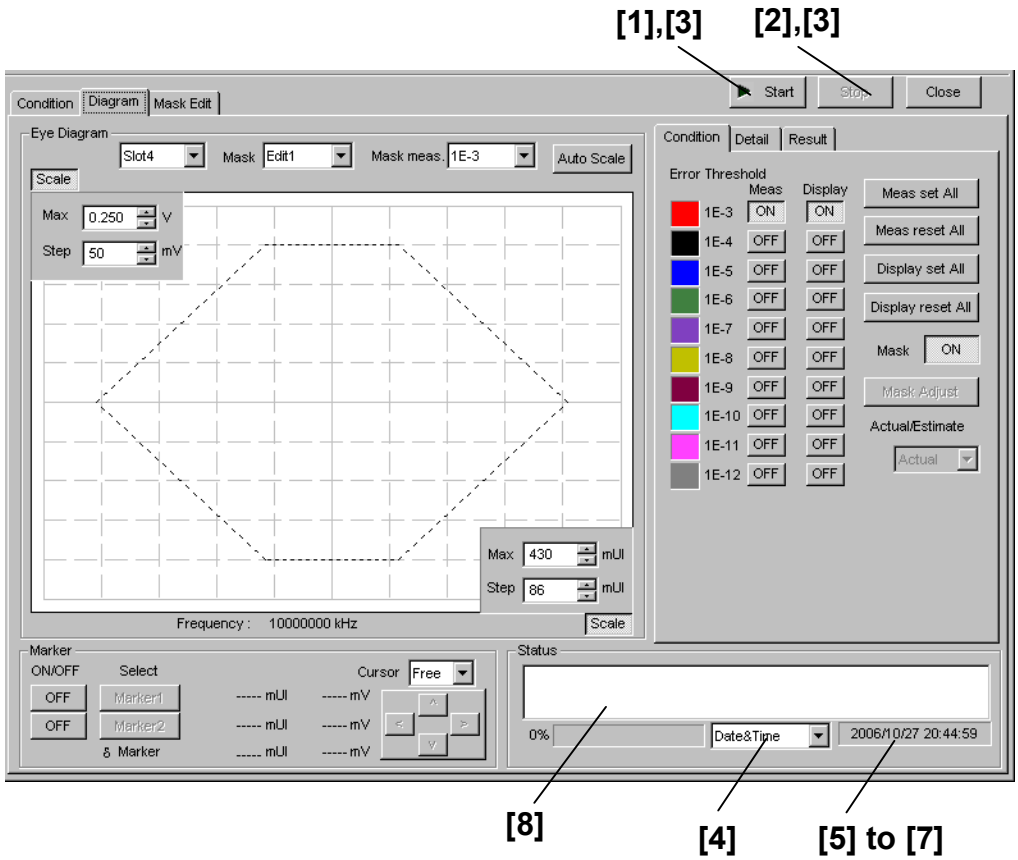


Figure 7.6.3.2.5-1 Diagram tab Start/Stop buttons and Status field

Table 7.6.3.2.5-1 Diagram tab (Status) setting commands

No.	Setting Items	Commands
[1]	Start	:SENSe:MEASure:EDIagram:DIAGram:STARt
[2]	Stop	:SENSe:MEASure:EDIagram:DIAGram:STOP
[3]	Query for measurement state	:SENSe:MEASure:EDIagram:STATe?
[4]	No label (Setting of time display)	:SENSe:MEASure:EDIagram:TIME
		:SENSe:MEASure:EDIagram:TIME?
[5]	Query for current time	:SENSe:MEASure:EDIagram:DTIMe?
[6]	Query for measurement start time	:SENSe:MEASure:EDIagram:STARt?
[7]	Query for measurement elapsed time	:SENSe:MEASure:EDIagram:ELAPsed?
[8]	Query for measurement status	:CALCulate:DATA:EDIagram:STATus?

:SENSe:MEASure:EDIagram:DIAGram:STARt

Function	Starts the Eye Diagram measurement.
Example	`> :SENSe:MEASure:EDIagram:DIAGram:STARt`
Compatibility	Incompatible with existing models.

: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	<numeric> = <NR1 NUMERIC RESPONSE DATA>	
	1	Measurement has been started.
	0	Measurement has been stopped
	–1	Measurement has failed.
Function	Queries the Eye Diagram measurement status.	
Example	`> :SENSe:MEASure:EDIagram:STATe?`	
	`< 1`	
Compatibility	Incompatible with existing models.	

:SENSe:MEASure:EDIagram: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 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	<type> = <CHARACTER RESPONSE DATA>	
	DTIM	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 Diagram measurement time display type.	
Example	`> :SENSe:MEASure:EDIagram:TIME?`	
	`< STAR`	
Compatibility	Incompatible with existing models.	

:SENSe:MEASure:EDIagram: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 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
<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 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
<day> = <NR1 NUMERIC RESPONSE DATA>
0, 1 to 99 1 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 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
<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:

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 or 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

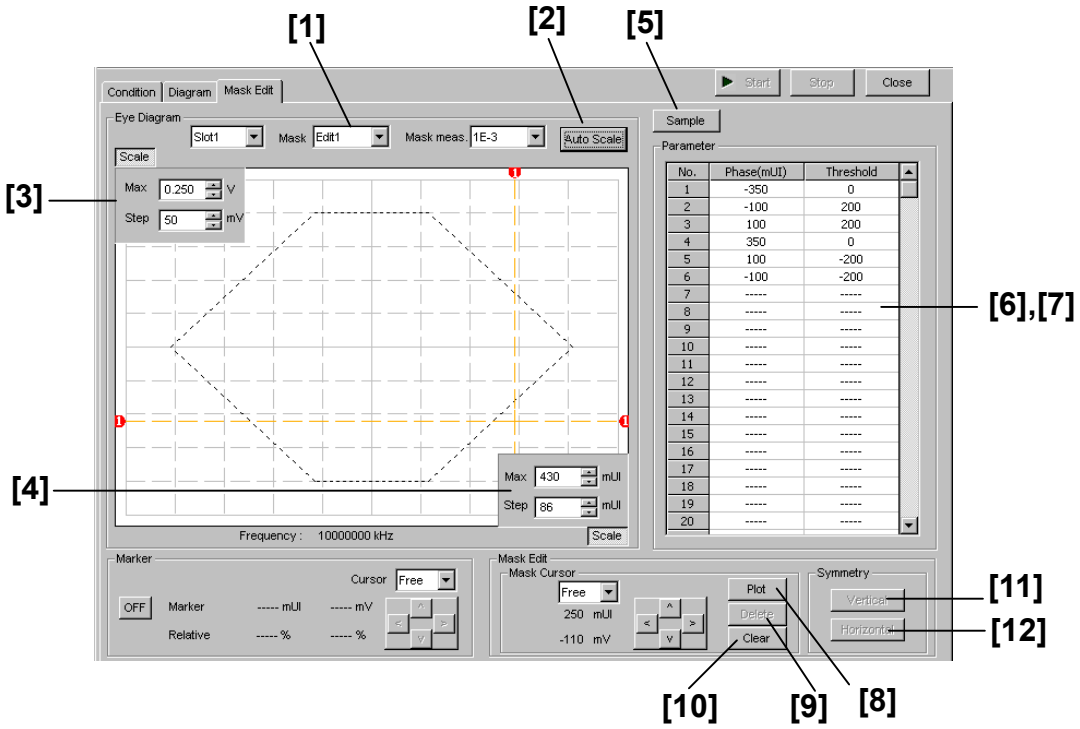


Figure 7.6.3.2.6-1 Mask Edit tab

Table 7.6.3.2.6-1 Mask Edit tab setting commands

No.	Setting Items	Commands
[1]	Mask	:DISPlay:RESult:EDIagram:TEMPlate:PATTern
		:DISPlay:RESult:EDIagram:TEMPlate:PATTern?
[2]	Auto Scale	:DISPlay:RESult:EDIagram:TEMPlate:ASCale
[3]	Max Step	:DISPlay:RESult:EDIagram:TEMPlate:SCALe
		:DISPlay:RESult:EDIagram:TEMPlate:SCALe?
[4]	Max Step	:DISPlay:RESult:EDIagram:TEMPlate:PSCale
		:DISPlay:RESult:EDIagram:TEMPlate:PSCale?
[5]	Sample	:DISPlay:RESult:EDIagram:TEMPlate:SAMPle
[6]	Query for mask point count	:DISPlay:RESult:EDIagram:TEMPlate:PNUMber?
[7]	Phase (mUI) Threshold	:DISPlay:RESult:EDIagram:TEMPlate:EDIT
		:DISPlay:RESult:EDIagram:TEMPlate:EDIT?
[8]	Plot	:DISPlay:RESult:EDIagram:TEMPlate:PLOT
[9]	Delete	:DISPlay:RESult:EDIagram:TEMPlate:DELete
[10]	Clear	:DISPlay:RESult:EDIagram:TEMPlate:CLEar
[11]	Vertical	:DISPlay:RESult:EDIagram:TEMPlate:VERTical
[12]	Horizontal	:DISPlay:RESult:EDIagram:TEMPlate:HORizontal

: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:PATTern?

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:PATTern?` `< 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:SCALe<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: `>:DISPlay:RESult:EDIagram:TEMPlate:SCALe 4.000,800`
Compatibility	Incompatible with existing models.

:DISPlay:RESult:EDIagram:TEMPlate:SCALe?

Response	<max> = <NR2 NUMERIC RESPONSE DATA> <step> = <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?` `< 4000,800`
Compatibility	Incompatible with existing models.

:DISPlay:RESult:EDIagram:TEMPlate:PSCale <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 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:PSCale 1000,10`
Compatibility	Incompatible with existing models.

:DISPlay:RESult:EDIagram:TEMPlate:PSCale?

Response	<max> = <NR2 NUMERIC RESPONSE DATA> <step> = <NR2 NUMERIC RESPONSE DATA>
Function	Queries the maximum value and the grid resolution in the direction of the Phase of the Mask Edit graph.
Example	`> :DISPlay:RESult:EDIagram:TEMPlate:PSCale?` `< 1000,10`
Compatibility	Incompatible with existing models.

:DISPlay:RESult:EDIagram:TEMPlate:SAMPle <string>

Parameter	<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	`> :DISPlay:RESult:EDIagram:TEMPlate:SAMPle` `"C:\…\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, 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	<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 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.

7.6.3.2.7 File menu setting commands

Table 7.6.3.2.7-1 File menu setting commands

No.	Setting Items	Commands
[1]	Open	:SYSTem:MMEMory:DIAGram:RECall
[2]	Save	:SYSTem:MMEMory:DIAGram:STORe
[3]	Print	:SYSTem:PRINt:DIAGram

: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.

: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.

7.6.4 Q measurement

This section describes the commands for Q measurement.

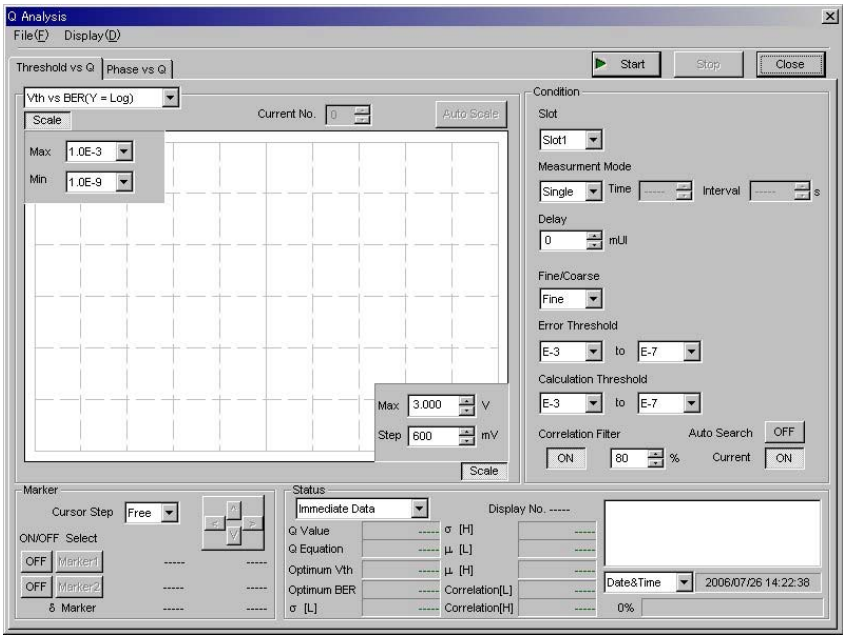


Figure 7.6.4-1 Q measurement screen

7.6.4.1 Graph display setting commands

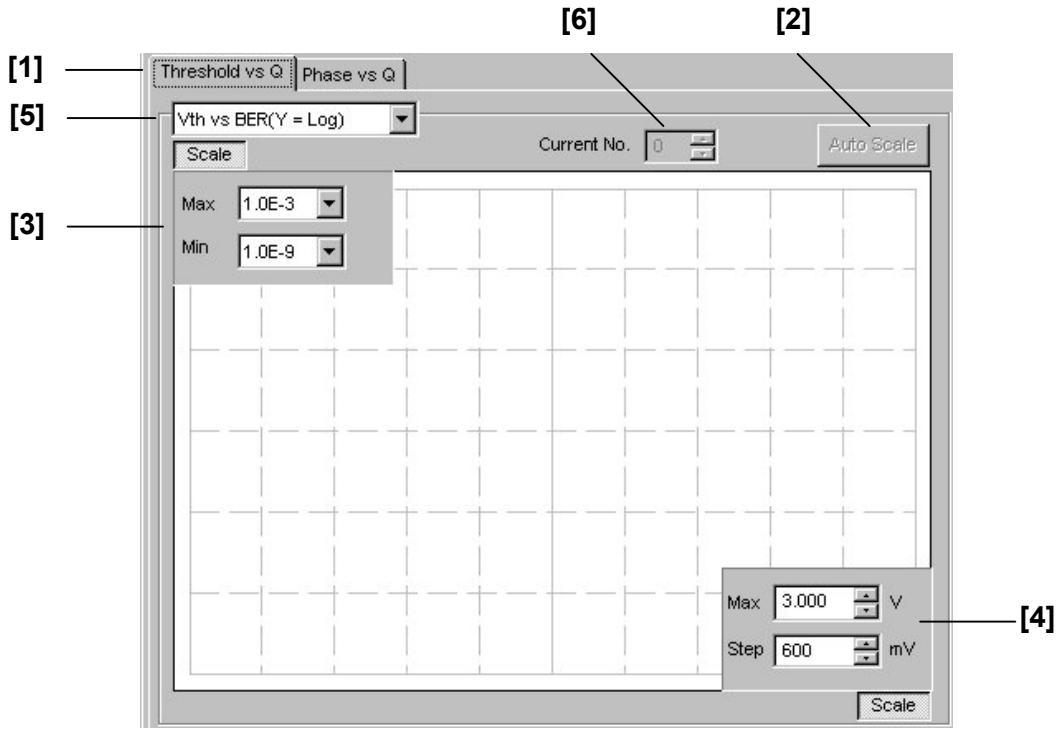


Figure 7.6.4.1-1 Graph display screen (Threshold vs Q tab)

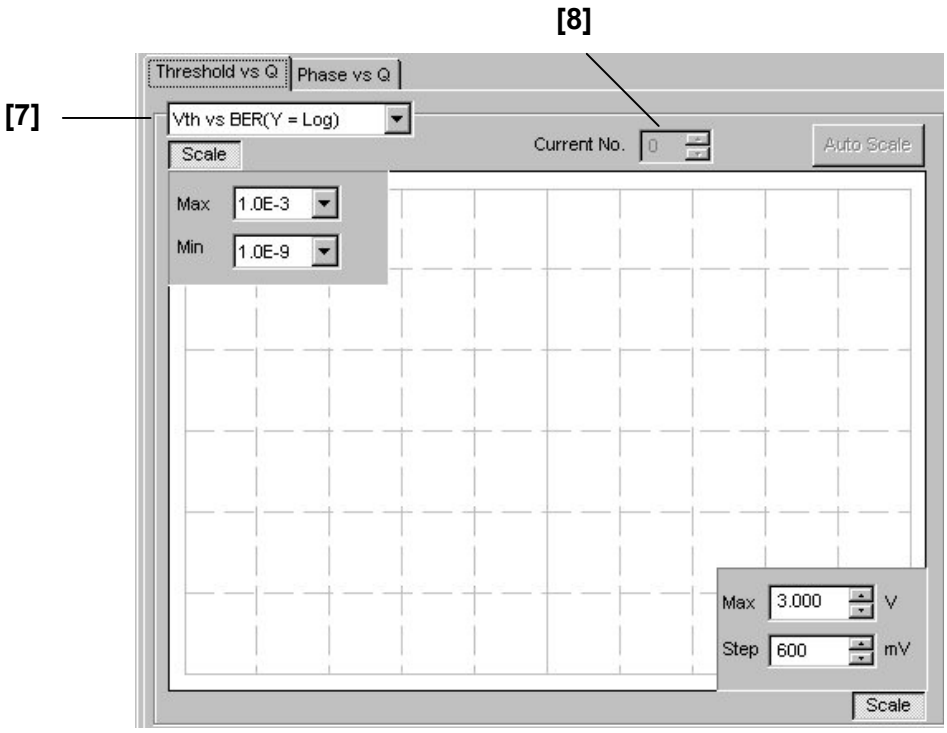


Figure 7.6.4.1-2 Graph display screen (Phase vs Q tab)

Table 7.6.4.1-1 Graph display setting commands

No.	Setting Items	Commands
[1]	Measurement mode switching	:DISPlay:RESult:QANalysis:MODE
		:DISPlay:RESult:QANalysis:MODE?
[2]	Auto Scale	:DISPlay:RESult:QANalysis:SCALe:ASCale
[3]	Max	:DISPlay:RESult:QANalysis:SCALe:VERTical
	Min	:DISPlay:RESult:QANalysis:SCALe:VERTical?
[4]	Max	:DISPlay:RESult:QANalysis:SCALe:HORizontal
	Step	:DISPlay:RESult:QANalysis:SCALe:HORizontal?
[5]	Threshold vs. Q Display graph selection	:DISPlay:RESult:QANalysis:THReshold:ITEM
		:DISPlay:RESult:QANalysis:THReshold:ITEM?
[6]	Threshold vs. Q Current No.	:DISPlay:RESult:QANalysis:THReshold:MNUMber
		:DISPlay:RESult:QANalysis:THReshold:MNUMber?
[7]	Phase vs. Q Display graph selection	:DISPlay:RESult:QANalysis:PHASe:ITEM
		:DISPlay:RESult:QANalysis:PHASe:ITEM?
[8]	Phase vs. Q Current No.	:DISPlay:RESult:QANalysis:PHASe:MNUMber
		:DISPlay:RESult:QANalysis:PHASe:MNUMber?

: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 7.6.4.1-2 Parameters for Threshold vs. Q tab

Format	<max>	<step>	Remarks
Vth vs BER (Y = Log)	2 to 7 (E–2 to E–7), in single steps	Inputs the minimum value. 8 to 14 (E–8 to E–14), in single steps	A difference between <max> and <step> must be 7 or greater even number.
Vth vs BER (Y = Log (–Ln))	2 to 7 (E–2 to E–7), in single steps	Inputs the minimum value. 8 to 14 (E–8 to E–14), in single steps	A difference between <max> and <step> must be 7 or greater even number.
Vth vs Q	10 to 1000, in 10 steps	1 to 100, in single steps	When Linear
	–40 to 60, in 10 steps	1 to 11, in single steps	When Log
Times vs Q	10 to 1000, in 10 steps	1 to 100, in single steps	When Linear
	–40 to 60, in 10 steps	1 to 11, in single steps	When Log
Histogram	50 to 1000, in 50 steps	Omitted	

Table 7.6.4.1-3 Parameters for Phase vs. Q tab

Format	<max>	<step>	Remarks
Vth vs BER (Y = Log)	2 to 7 (E–2 to E–7), in single steps	Inputs the minimum value. 8 to 14 (E–8 to E–14), in single steps	A difference between <max> and <step> must be 7 or greater even number.
Vth vs BER (Y = Log (–Ln))	2 to 7 (E–2 to E–7), in single steps	Inputs the minimum value. 8 to 14 (E–8 to E–14), in single steps	A difference between <max> and <step> must be 7 or greater even number.
Vth vs Q	10 to 1000, in 10 steps	1 to 100, in single steps	When Linear
	–40 to 60, in 10 steps	1 to 11, in single steps	When Log
Phase vs Q	10 to 1000, in 10 steps	1 to 100, in single steps	When Linear
	–40 to 60, in 10 steps	1 to 11, in single steps	When Log
Phase vs σ	0.0010 to 1.0000, in 0.0010 steps	0.001 to 0.1000, 0.0001 steps	
Phase vs μ	–3.990 to 4.000 (V), in 0.010 V steps	0.001 to 0.800 (V), in 0.001 V steps	
Phase vs Opt BER	10 to 1 (E–10 to E–1), in single steps	Inputs the minimum value. 199 to 20 (E–199 to E–20), in single steps	
Phase vs Opt Threshold	–3.990 to 4.000 (V), in 0.010 V steps	0.001 to 0.800 (V), in 0.001 V steps	
Phase vs Correlation	10 to 100 (%), in 10 % steps	0 to 90 (%), in 10 % steps	A difference between <max> and <step> must be 10 or greater

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

Format	<max>	<step>	Remarks
Vth vs BER (Y = Log)	–3.990 to 4.000 (V), in 0.010 V steps	0.001 to 0.800 (V), in 0.001 V steps	
Vth vs BER (Y = Log (–Ln))	–3.990 to 4.000 (V), in 0.010 V steps	0.001 to 0.800 (V), in 0.001 V steps	
Vth vs Q	–3.990 to 4.000 (V), in 0.010 V steps	0.001 to 0.800 (V), in 0.001 V steps	
Times vs Q	100 to 1000, in 50 steps	10 (fixed)	
Histogram	0.16 to 1000.00, in 0.01 steps	16/32/64/128/256	When Linear <max> must be greater than <step>.
	–49.84 to 60.00 (dB), in 0.01 steps	16/32/64	When Log

Table 7.6.4.1-5 Parameters for Phase vs. Q tab

Format	<max>	<step>	Remarks
Vth vs BER (Y = Log)	–3.990 to 4.000 (V), in 0.010 V steps	0.001 to 0.800 (V), in 0.001 V steps	
Vth vs BER (Y = Log (–Ln))	–3.990 to 4.000 (V), in 0.010 V steps	0.001 to 0.800 (V), in 0.001 V steps	
Vth vs Q	–3.990 to 4.000 (V), in 0.010 V steps	0.001 to 0.800 (V), in 0.001 V steps	
Phase vs Q	–900 to 1000 (mUI), in 100 mUI steps	10 to 200 (mUI), in 10 mUI steps	
Phase vs σ	–900 to 1000 (mUI), in 100 mUI steps	10 to 200 (mUI), in 10 mUI steps	
Phase vs μ	–900 to 1000 (mUI), in 100 mUI steps	10 to 200 (mUI), in 10 mUI steps	
Phase vs Opt BER	–900 to 1000 (mUI), in 100 mUI steps	10 to 200 (mUI), in 10 mUI steps	
Phase vs Opt Threshold	–900 to 1000 (mUI), in 100 mUI steps	10 to 200 (mUI), in 10 mUI steps	
Phase vs Correlation	–900 to 1000 (mUI), in 100 mUI steps	10 to 200 (mUI), in 10 mUI steps	

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
```

Compatibility: Incompatible with existing models.

: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: `> :DISPlay:RESult:QANalysis: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	`> :DISPlay:RESult:QANalysis:THReshold:MNUMber?` `< 100`
Compatibility	Incompatible with existing models.

:DISPlay:RESult:QANalysis: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. σ
	PHAMu	Phase vs. μ
	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: `> :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.

7.6.4.2 Vth vs. Q measurement commands

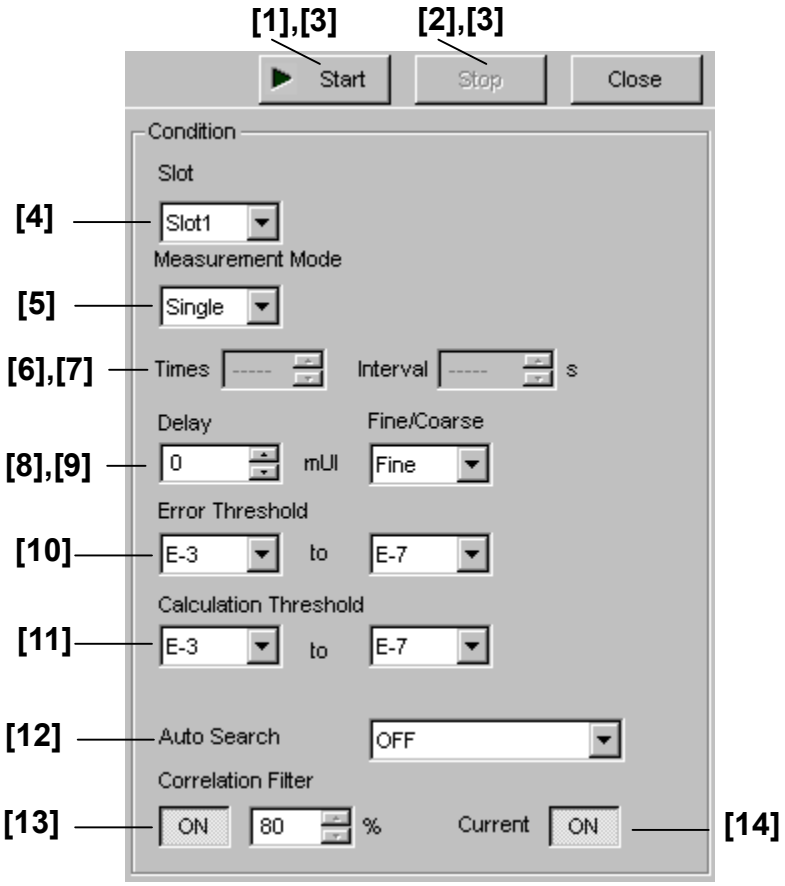


Figure 7.6.4.2-1 Vth vs. Q measurement setting field

Table 7.6.4.2-1 Vth vs. Q measurement setting commands

No.	Setting Items	Commands
[1]	Start	:SENSe:MEASure:QANalysis:THReshold:STARt
[2]	Stop	:SENSe:MEASure:QANalysis:THReshold:STOP
[3]	Query for measurement progress state	:SENSe:MEASure:QANalysis:STATe?
[4]	Slot	:SENSe:MEASure:QANalysis:THReshold:SLOT
		:SENSe:MEASure:QANalysis:THReshold:SLOT?
[5]	Meas Mode	:SENSe:MEASure:QANalysis:THReshold:MODE
		:SENSe:MEASure:QANalysis:THReshold:MODE?
[6]	Times	:SENSe:MEASure:QANalysis:THReshold:TIMEs
		:SENSe:MEASure:QANalysis:THReshold:TIMEs?
[7]	Interval	:SENSe:MEASure:QANalysis:THReshold:INTerval
		:SENSe:MEASure:QANalysis:THReshold:INTerval?
[8]	Delay	:SENSe:MEASure:QANalysis:THReshold:DELay
		:SENSe:MEASure:QANalysis:THReshold:DELay?
[9]	Fine/Coarse	:SENSe:MEASure:QANalysis:THReshold:RESolution
		:SENSe:MEASure:QANalysis:THReshold:RESolution?

Table 7.6.4.2-1 Vth vs. Q measurement setting commands (Cont'd)

No.	Setting Items	Commands
[10]	Error Threshold	:SENSe:MEASure:QANalysis:THReshold:ERANge
		:SENSe:MEASure:QANalysis:THReshold:ERANge?
[11]	Calculation Threshold	:SENSe:MEASure:QANalysis:THReshold:CALRange
		:SENSe:MEASure:QANalysis:THReshold:CALRange?
[12]	Auto Search	:SENSe:MEASure:QANalysis:THReshold:ASEarch
		:SENSe:MEASure:QANalysis:THReshold:ASEarch?
[13]	Correlation Filter	:SENSe:MEASure:QANalysis:THReshold:CFILter
		:SENSe:MEASure:QANalysis:THReshold:CFILter?
[14]	Current	:SENSe:MEASure:QANalysis:THReshold:CURRent
		:SENSe:MEASure:QANalysis:THReshold:CURRent?

: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?
< 1
```

Compatibility: Incompatible with existing models.

: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?

Response	<mode> = <CHARACTER RESPONSE DATA> SING, REP, UNT
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>

Parameter	<numeric> = <DECIMAL NUMERIC PROGRAM DATA> 2 to 1000 2 to 1000 times, in 1-time steps
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?

Response	<numeric> = <NR1 NUMERIC RESPONSE DATA>
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 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.

: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.

:SENSe:MEASure:QANalysis: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:QANalysis:THReshold:ASEarch THReshold`	
Compatibility	Incompatible with existing models.	

:SENSe:MEASure:QANalysis: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:QANalysis:THReshold:ASEarch?` `< THR`
Compatibility	Incompatible with existing models.

: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?

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

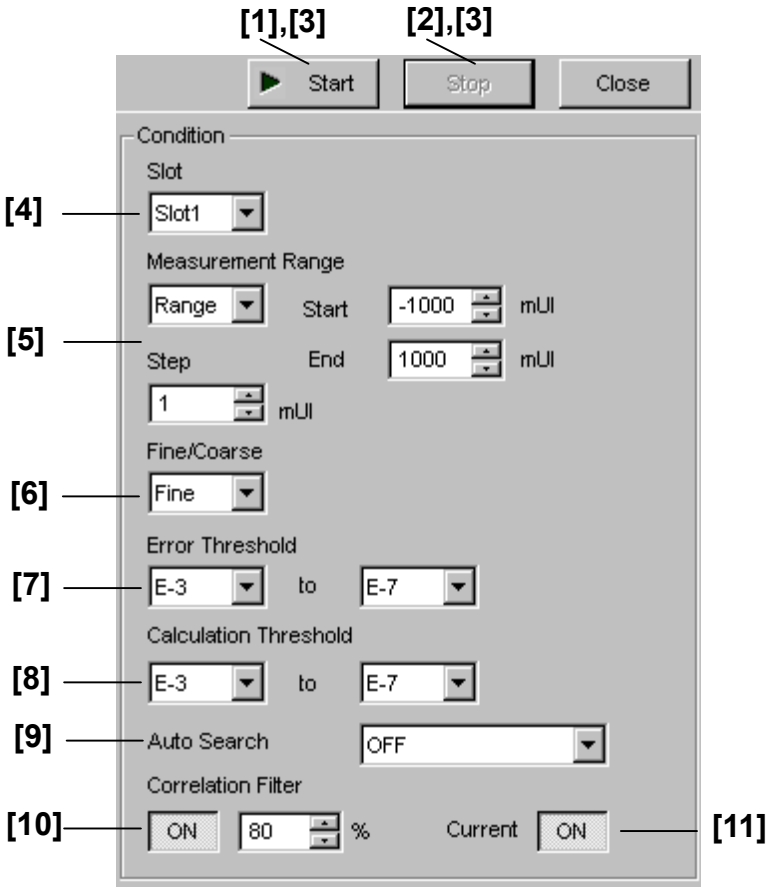


Figure 7.6.4.3-1 Phase vs. Q measurement setting field

Table 7.6.4.3-1 Phase vs. Q measurement setting commands

No.	Setting Items		Commands
[1]	Start		:SENSe:MEASure:QANalysis:PHASe:STARt
[2]	Stop		:SENSe:MEASure:QANalysis:PHASe:STOP
[3]	Query for measurement state		:SENSe:MEASure:QANalysis:STATe?
[4]	Slot		:SENSe:MEASure:QANalysis:PHASe:SLOT
			:SENSe:MEASure:QANalysis:PHASe:SLOT?
[5]	Measurement Range	Start End	:SENSe:MEASure:QANalysis:PHASe:RANGe
			:SENSe:MEASure:QANalysis:PHASe:RANGe?
		Center Span	:SENSe:MEASure:QANalysis:PHASe:WIDTh
			:SENSe:MEASure:QANalysis:PHASe:WIDTh?
[6]	Fine/Coarse		:SENSe:MEASure:QANalysis:PHASe:RESolution
			:SENSe:MEASure:QANalysis:PHASe:RESolution?
[7]	Error Threshold		:SENSe:MEASure:QANalysis:PHASe:ERANge
			:SENSe:MEASure:QANalysis:PHASe:ERANge?
[8]	Calculation Threshold		:SENSe:MEASure:QANalysis:PHASe:CALRange
			:SENSe:MEASure:QANalysis:PHASe:CALRange?

Table 7.6.4.3-1 Phase vs. Q measurement setting commands (Cont'd)

No.	Setting Items	Commands
[9]	Auto Search	:SENSe:MEASure:QANalysis:PHASe:ASEarch
		:SENSe:MEASure:QANalysis:PHASe:ASEarch?
[10]	Correlation Filter	:SENSe:MEASure:QANalysis:PHASe:CFILter
		:SENSe:MEASure:QANalysis:PHASe:CFILter?
[11]	Current	:SENSe:MEASure:QANalysis:PHASe:CURRent
		:SENSe:MEASure:QANalysis:PHASe:CURRent?

: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.

:SENSe:MEASure:QANalysis: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:QANalysis:PHASe:RANGe?` `< -200,300,10`
Compatibility	Incompatible with existing models.

:SENSe:MEASure:QANalysis:PHASe:WIDTh ,,<step>

Parameter	= <DECIMAL NUMERIC PROGRAM DATA> –1000 to 1000 –1000 to 1000 mUI, in 1-mUI steps = <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:QANalysis:PHASe:WIDTh 100,200,10`
Compatibility	Incompatible with existing models.

:SENSe:MEASure:QANalysis:PHASe:WIDTh?

Response	= <NR1 NUMERIC RESPONSE DATA> = <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 <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.

: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
	Omitted when <boolean> is "0" or "OFF".
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?

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	`> :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:
	`> :SENSe:MEASure:QANalysis:PHASe:CURRent 1`
Compatibility	Incompatible with existing models.

:SENSe:MEASure:QANalysis: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:QANalysis:PHASe:CURRent?` `< 1`
Compatibility	Incompatible with existing models.

7.6.4.4 Measurement results and status query commands

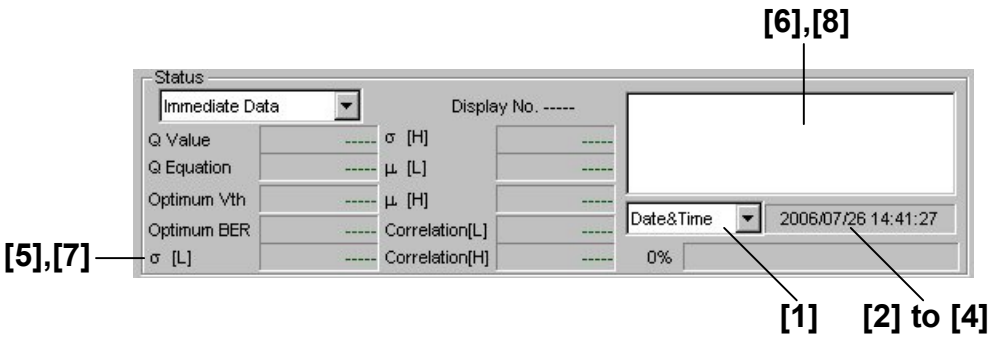


Figure 7.6.4.4-1 Measurement results and status display field

Table 7.6.4.4-1 Measurement results and status query commands

No.	Setting Items	Commands
[1]	No label (Setting of time display)	:SENSe:MEASure:QANalysis:TIME
		:SENSe:MEASure:QANalysis:TIME?
[2]	Query for current time	:SENSe:MEASure:QANalysis:DTIMe?
[3]	Query for measurement start time	:SENSe:MEASure:QANalysis:STARt?
[4]	Query for measurement elapsed time	:SENSe:MEASure:QANalysis:ELAPsed?
[5]	Query for Vth vs. Q measurement result	:CALCulate:DATA:QANalysis:THReshold?
[6]	Query for Vth vs. Q measurement status	:CALCulate:DATA:QANalysis:TSTatus?
[7]	Query for Phase vs. Q measurement result	:CALCulate:DATA:QANalysis:PHASe?
[8]	Query for Phase vs. Q measurement status	:CALCulate:DATA:QANalysis:PSTatus?

: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.	

: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 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 Q measurement.
Example	> :SENSe:MEASure:QANalysis:DTIMe? < 2005,12,31,23,59,59
Compatibility	Incompatible with existing models.

: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
```
> :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
```
> :SENSe:MEASure:QANalysis:ELAPsed?
< 31,23,59,59
```

Compatibility
Incompatible with existing models.

: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 7.6.4.4-2 Vth vs. Q measurement result <result1>

Items		<result1>	Response Format
Top side Vth/Error Rate	Number of measurements 1 to 1000	"VTHQ:TOP,1"..."VTHQ:TOP,1000"	Form1
Bottom side Vth/Error Rate		"VTHQ:BTM,1"..."VTHQ:BTM,1000"	Form1

The measurement result for each Vth is delimited with a comma (,) and returned.

Table 7.6.4.4-3 Vth vs. Q immediate data <result2>

Items		<result2>	Response Format
Q value	Number of measurements 1 to 1000	"QVALue:1"... "QVALue:1000"	Form2
Optimum BER		"OPTBer:1"... "OPTBer:1000"	Form3
Optimum Vth		"OPTVth:1"... "OPTVth:1000"	Form4
Correlation bottom		"CORBtm:1"... "CORBtm:1000"	Form2
Correlation top		"CORTop:1"... "CORTop:1000"	Form2
σ bottom		"SGMBtm:1"... "SGMBtm:1000"	Form2
σ top		"SGMTop:1"... "SGMTop:1000"	Form2
μ bottom		"MUBtm:1"... "MUBtm:1000"	Form4
μ top		"MUTop:1"... "MUTop:1000"	Form4
Q Equation		"QEQ:1"... "QEQ:1000"	Form2

Table 7.6.4.4-4 Vth vs. Q measurement statistic data <result3>

Items	<Result3>	Response Format
Total Data	"TOTal"	Form5
Valid Data	"VALid"	Form5
Q Max	"QMAX"	Form2
Q Min	"QMIN"	Form2
Q Mean	"QMEan"	Form2
Q σ	"QSGM"	Form2
Q-5 σ	"QSGM5"	Form2

Response <string> = <STRING RESPONSE DATA>

Table 7.6.4.4-5 Response format

Form	Format	Description
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 1.0000E000
	"--------"	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:

```
> :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> = <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 Vth vs. Q measurement status.

Example

```
> :CALCulate:DATA:QANalysis:TSTatus?
< "Sync Loss"
```

Compatibility Incompatible with existing models.

: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>

Items		<result1>	Response Format
Top side Vth-Error Rate	Phase –1000 to 1000 mUI	"VTHQ:TOP,XXXXX"	Form1
Bottom side Vth-Error Rate		"VTHQ:BTM,XXXXX"	Form1

Table 7.6.4.4-7 Phase vs. Q immediate data <result2>

Items		<result2>	Response Format
Q value	Phase –1000 to 1000	"QVALue:–1000"… "QVALue:1000"	Form2
Optimum BER		"OPTBer:–1000"… "OPTBer:1000"	Form3
Optimum Vth		"OPTVth:–1000"… "OPTVth:1000"	Form4
Correlation bottom		"CORBtm:–1000"… "CORBtm:1000"	Form2
Correlation top		"CORTop:–1000"… "CORTop:1000"	Form2
Σ bottom		"SGMBtm:–1000"… "SGMBtm:1000"	Form2
Σ top		"SGMTop:–1000"… "SGMTop:1000"	Form2
M bottom		"MUBtm:–1000"… "MUBtm:1000"	Form4
M top		"MUTop:–1000"… "MUTop:1000"	Form4
Q equation		"QEQ:–1000"… "QEQ:1000"	Form2

Response

<string> = <STRING RESPONSE DATA>

Table 7.6.4.4-8 Response format

Form	Format	Description
Form1 Vth-ER 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 1.0000E000
	"--------"	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.

Function

Queries the Phase vs. Q measurement result.

Example

To query the TOP side measurement result:

```
> :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"
```

7.6.4.5 File menu setting commands

Table 7.6.4.5-1 File menu setting commands

No.	Setting Items	Commands
[1]	Open	:SYSTem:MMEMory:QANalysis:RECall
[2]	Save	:SYSTem:MMEMory:QANalysis:STORe
[3]	Print	:SYSTem:PRINt:QANalysis
[4]	Q Unit	:SENSe:MEASure:QANalysis:THReshold:LSCale
		:SENSe:MEASure:QANalysis:THReshold:LSCale?
		:SENSe:MEASure:QANalysis:PHASe:LSCale
		:SENSe:MEASure:QANalysis:PHASe:LSCale?

: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 <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 by specifying the save destination, file name, and file format.

```
> :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:

```
:SYSTem:PRINt:QANalysis VTQ
```

Compatibility

Incompatible with existing models.

:SENSe:MEASure:QANalysis:THReshold:LSCale <scale>

Parameter	<scale> = <CHARACTER PROGRAM DATA> LINear 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:QANalysis:THReshold:LSCale LOG`
Compatibility	Incompatible with existing models.

:SENSe:MEASure:QANalysis:THReshold:LSCale?

Response	<scale> = <CHARACTER RESPONSE DATA> LIN, LOG
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>

Parameter	<scale> = <CHARACTER PROGRAM DATA> LINear 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:QANalysis:PHASe:LSCale LOG`
Compatibility	Incompatible with existing models.

:SENSe:MEASure:QANalysis:PHASe:LSCale?

Response	<scale> = <CHARACTER RESPONSE DATA> LIN, LOG
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.

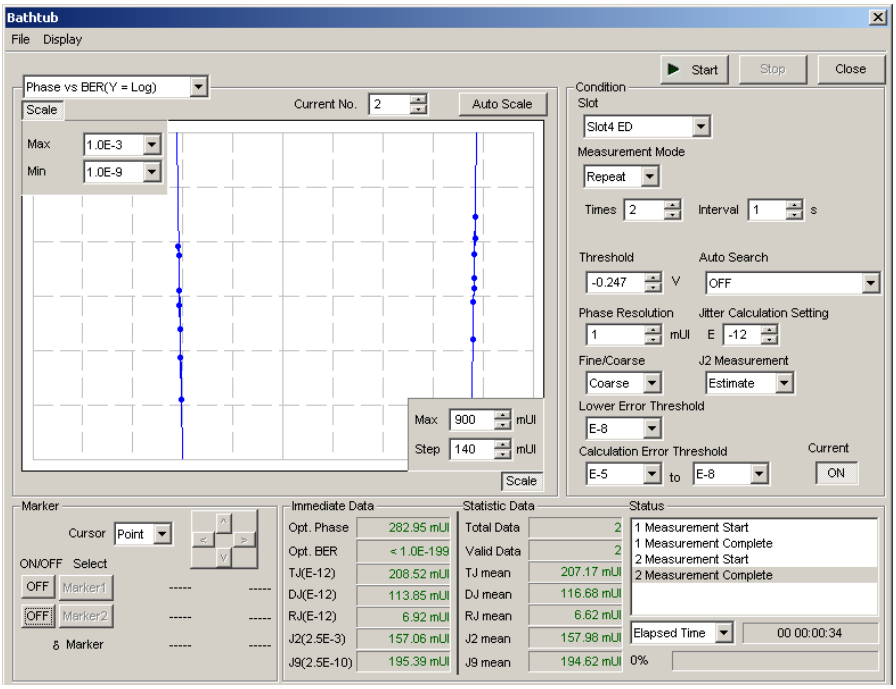


Figure 7.6.5-1 Bathtub measurement screen

7.6.5.1 Measurement setting commands

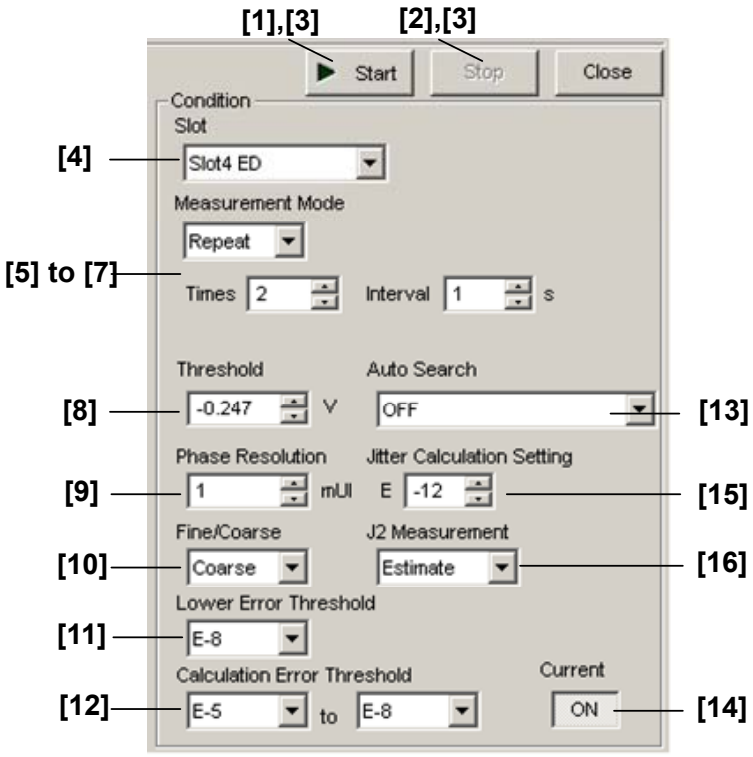


Figure 7.6.5.1-1 Measurement setting field

Table 7.6.5.1-1 Measurement setting commands

No.	Setting Items	Commands
[1]	Start	:SENSe:MEASure:BATHtub:STARt
[2]	Stop	:SENSe:MEASure:BATHtub:STOP
[3]	Query for measurement status	:SENSe:MEASure:BATHtub:STATe?
[4]	Slot	:SENSe:MEASure:BATHtub:SLOT
		:SENSe:MEASure:BATHtub:SLOT?
[5]	Measurement Mode	:SENSe:MEASure:BATHtub:MODE
		:SENSe:MEASure:BATHtub:MODE?
[6]	Time	:SENSe:MEASure:BATHtub:TIME
		:SENSe:MEASure:BATHtub:TIME?
[7]	Interval	:SENSe:MEASure:BATHtub:INTerval
		:SENSe:MEASure:BATHtub:INTerval?
[8]	Threshold	:SENSe:MEASure:BATHtub:DATA
		:SENSe:MEASure:BATHtub:DATA?
[9]	Phase Resolution	:SENSe:MEASure:BATHtub:RESolution:MUI
		:SENSe:MEASure:BATHtub:RESolution:MUI?
[10]	Fine/Coarse	:SENSe:MEASure:BATHtub:ECOunt
		:SENSe:MEASure:BATHtub:ECOunt?
[11]	Lower Error Threshold	:SENSe:MEASure:BATHtub:RANGe
		:SENSe:MEASure:BATHtub:RANGe?
[12]	Calculation Error Threshold	:SENSe:MEASure:BATHtub:CALRange
		:SENSe:MEASure:BATHtub:CALRange?
[13]	Auto Search	:SENSe:MEASure:BATHtub:ASEarch
		:SENSe:MEASure:BATHtub:ASEarch?
[14]	Current	:SENSe:MEASure:BATHtub:CURRent
		:SENSe:MEASure:BATHtub:CURRent?
[15]	Jitter Calculation Setting	:SENSe:MEASure:BATHtub:JCALculation
		:SENSe:MEASure:BATHtub:JCALculation?
[16]	J2 Measurement	:SENSe:MEASure:BATHtub:J2Meas
		:SENSe:MEASure:BATHtub:J2Meas?

: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?

Response	<numeric> = <NR1 NUMERIC RESPONSE DATA>
	0 During measurement stop
	1 During measurement
Function	Queries the measurement processing state during Bathtub measurement.
Example	`> :SENSe:MEASure:BATHtub:STATe?`
	`< 1`
Compatibility	Incompatible with existing models.

:SENSe:MEASure:BATHtub: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 Bathtub measurement target slot.
Example	To set the target slot to slot 3.
	`> :SENSe:MEASure:BATHtub:SLOT 3`
Compatibility	Incompatible with existing models.

:SENSe:MEASure:BATHtub: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>
Function	Queries the Bathtub measurement target slot.
Example	`> :SENSe:MEASure:BATHtub:SLOT?`
	`< 3`
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.

: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> = <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.

:SENSe:MEASure:BATHtub:RESolution:MUI <numeric>

Parameter	<numeric> = <DECIMAL NUMERIC PROGRAM DATA>
	1 to 100 1 to 100 mUI, in 1-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_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 RESONSE 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	Auto search OFF
	PHASe	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.

: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.

7.6.5.2 Graph display setting commands

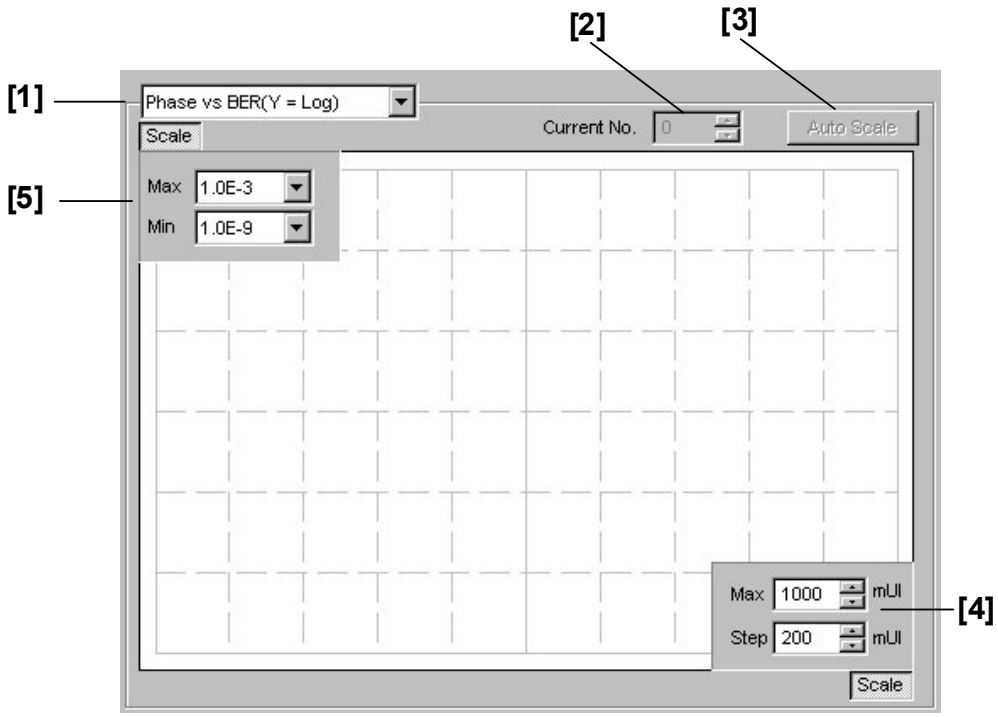


Figure 7.6.5.2-1 Graph display screen

Table 7.6.5.2-1 Graph display setting commands

No.	Setting Items	Commands
[1]	Display graph selection	:DISPlay:RESult:BATHtub:ITEM
		:DISPlay:RESult:BATHtub:ITEM?
[2]	Current No.	:DISPlay:RESult:BATHtub:MNUMber
		:DISPlay:RESult:BATHtub:MNUMber?
[3]	Auto Scale	:DISPlay:RESult:BATHtub:SCALe:ASCale
[4]	Horizontal axis scale setting	:DISPlay:RESult:BATHtub:SCALe:HORizontal
		:DISPlay:RESult:BATHtub:SCALe:HORizontal?
[5]	Vertical axis scale setting	:DISPlay:RESult:BATHtub:SCALe:VERTical
		:DISPlay:RESult:BATHtub:SCALe:VERTical?

: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.	

: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.

:DISPlay:RESult:BATHtub:SCALe:HORizontal <max>[,<step>]

Parameter <max>,<step> = <DECIMAL NUMERIC PROGRAM DATA>

Table 7.6.5.2-2 Parameter

Form	<max>	<step>
Phase type	–900 to 1000 (mUI), in 100-mUI steps	10 to 200 (mUI), in 10-mUI steps
Histogram type	16,32,64,128,256	Omitted

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

Form	<max>	<min>
Histogram type	6 to 14 (E+6 to E+14)	Input 0
Error Rate type	2 to 7 (E–2 to E–7)	8 to 15 (E–8 to E–15)

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.

7.6.5.3 Measurement result and status query commands

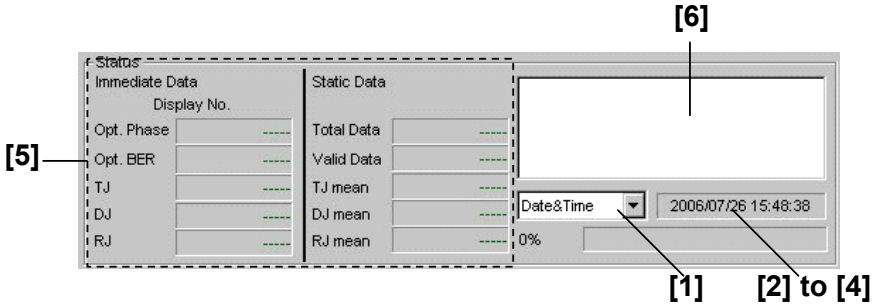


Figure 7.6.5.3-1 Measurement result and status display

Table 7.6.5.3-1 Measurement result and status query commands

No.	Setting Items	Commands
[1]	Display time selection	:SENSe:MEASure:BATHtub:DTIME
		:SENSe:MEASure:BATHtub:DTIME?
[2]	Query for current time	:SENSe:MEASure:BATHtub:TIME:DTIMe?
[3]	Query for measurement start time	:SENSe:MEASure:BATHtub:TIME:STARt?
[4]	Query for measurement elapsed time	:SENSe:MEASure:BATHtub:TIME:ELAPsed?
[5]	Query for result	:CALCulate:DATA:BATHtub?
[6]	Query for status	:CALCulate:DATA:BATHtub:STATus?

: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
```
> :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
```
> :SENSe:MEASure:BATHtub:TIME:STARt?
< 2005,12,31,23,59,59
```

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 elapsed 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>

<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 (Phase/Error Rate)	Number of measurements 1 to 1000	"BATH:1"…"BATH:1000"	Form1
Histogram (Phase/Error Count)		"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

Response <string> = <STRING RESPONSE DATA>

Table 7.6.5.3-5 Response format

Form	Format	Description
Form1 Phase-BER type	"XXXX.XX, Y.YYYYE-YYY"	XXXX.XX: Phase (mUI or PS) Y.YYYYE-YYY: Error Rate
	"-------,-------"	When no data corresponds to the query.
Form2 Histogram type	"XXXX.XX, Y.YYYYE-YYY"	XXXX.XX: Phase (mUI or PS) Y.YYYYE-YYY: Error Count
	"-------,-------"	When no data corresponds to the query.
Form3 Fraction type 1	"X.XXXXE-XXX"	When 0.0000E–016 to 1.0000E000
	"--------"	When no data corresponds to the query.
	"< 1.0E-199"	When E-199 or less
Form4 Fraction type 2	"XXXX.XX"	When –9999.99 to 9999.99
	"--------"	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 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?

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.
Illegal Error > Frequency NG > Clock Loss (CR Unlock) > Sync Loss > Out or range

Function Queries the Q measurement status

Example

```
> :CALCulate:DATA:BATHtub:STATus?
< "Sync Loss"
```

7.6.5.4 File menu setting commands

Table 7.6.5.4-1 File menu setting commands

No.	Setting Items	Commands
[1]	Open	:SYSTem:MMEMory:BATHtub:RECall
[2]	Save	:SYSTem:MMEMory:BATHtub:STORe
[3]	Print	:SYSTem:PRINt:BATHtub

: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>

Parameter	<list> = <CHARACTER PROGRAM DATA> PSB Phase vs. BER Result
Function	Prints the Bathtub measurement result data.
Example	> :SYSTem:PRINt:BATHtub
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

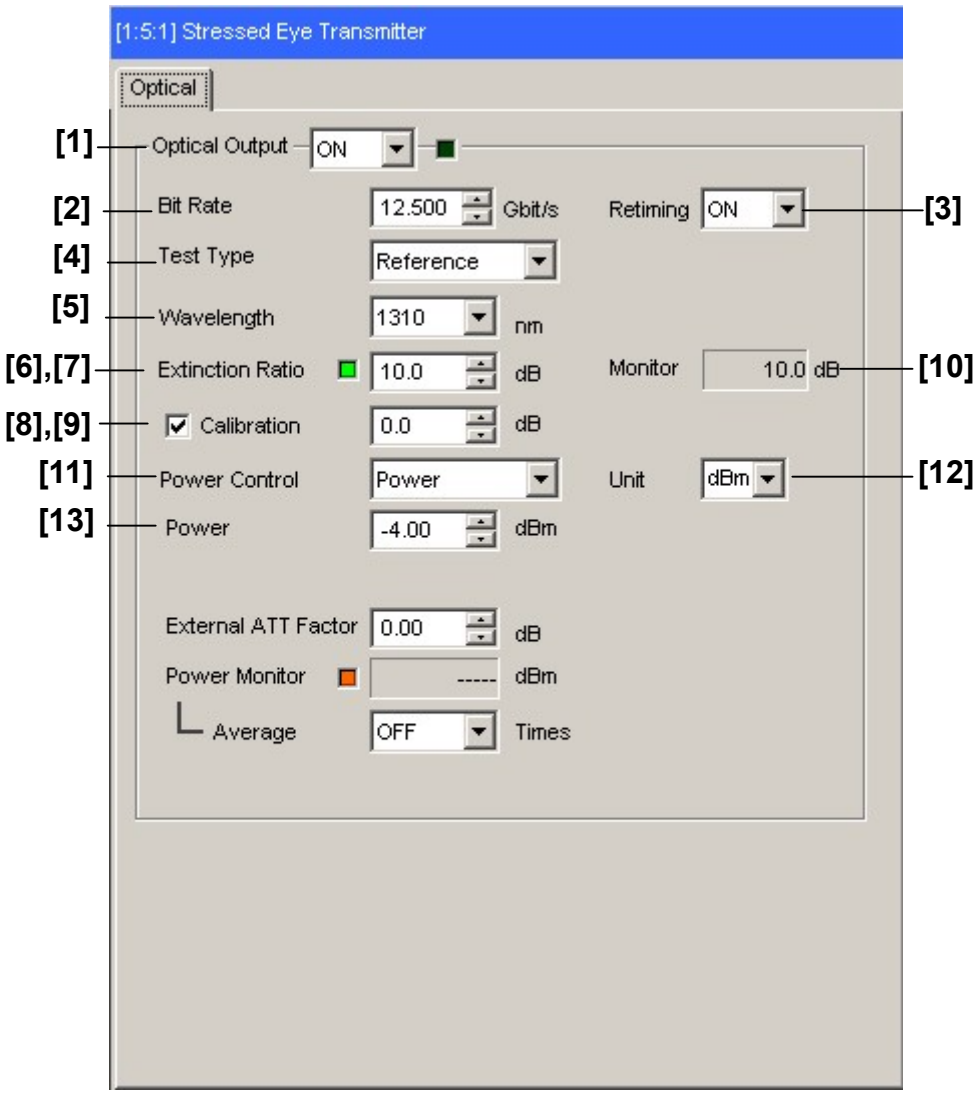


Figure 7.7.1.1-1 Optical Output setting(when Power Control is set to Power)

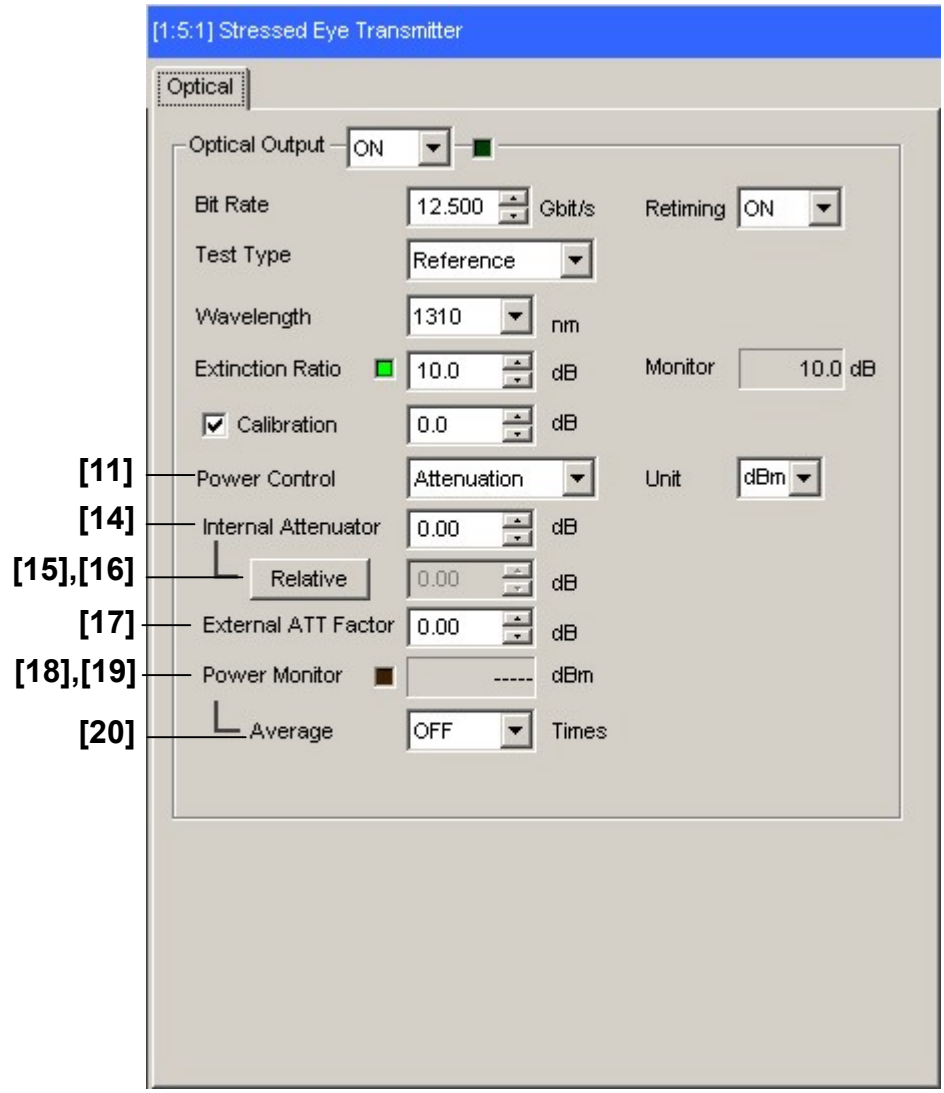


Figure 7.7.1.1-2 Optical Output setting (when Power Control is set to Attenuation)

Table 7.7.1.1-1 Optical Output setting commands

No.	Setting Items	Commands
[1]	Optical Output	:TRANsmitter:OPTical:OUTPut
		:TRANsmitter:OPTical:OUTPut?
[2]	Bit Rate	:TRANsmitter:OPTical:OUTPut:BITRate
		:TRANsmitter:OPTical:OUTPut:BITRate?
[3]	Retiming	:TRANsmitter:OPTical:OUTPut:RETiming
		:TRANsmitter:OPTical:OUTPut:RETiming?
[4]	Test Type	:TRANsmitter:OPTical:OUTPut:TESTtype
		:TRANsmitter:OPTical:OUTPut:TESTtype?
[5]	Wavelength	:TRANsmitter:OPTical:OUTPut:WAVelength
		:TRANsmitter:OPTical:OUTPut:WAVelength?

Table 7.7.1.1-1 Optical Output setting commands (Cont'd)

No.	Setting Items	Commands
[6]	Extinction Ratio LED display	:TRANsmitter:OPTical:OUTPut:EXTinction:OVERrange?
[7]	Extinction Ratio	:TRANsmitter:OPTical:OUTPut:EXTinction:RATio
		:TRANsmitter:OPTical:OUTPut:EXTinction:RATio?
[8]	Calibration check box	:TRANsmitter:OPTical:OUTPut:EXTinction:OCALibration
		:TRANsmitter:OPTical:OUTPut:EXTinction:OCALibration?
[9]	Calibration	:TRANsmitter:OPTical:OUTPut:EXTinction:CALibration
		:TRANsmitter:OPTical:OUTPut:EXTinction:CALibration?
[10]	Monitor	:TRANsmitter:OPTical:OUTPut:EXTinction:MONitor?
[11]	Power Control	:TRANsmitter:OPTical:OUTPut:CONTrol
		:TRANsmitter:OPTical:OUTPut:CONTrol?
[12]	Unit	:TRANsmitter:OPTical:OUTPut:CONTrol:UNIT
		:TRANsmitter:OPTical:OUTPut:CONTrol:UNIT?
[13]	Power	:TRANsmitter:OPTical:OUTPut:POWer
		:TRANsmitter:OPTical:OUTPut:POWer?
[14]	Internal Attenuator	:TRANsmitter:OPTical:OUTPut:ATTenuator
		:TRANsmitter:OPTical:OUTPut:ATTenuator?
[15]	Relative	:TRANsmitter:OPTical:OUTPut:RELative
		:TRANsmitter:OPTical:OUTPut:RELative?
[16]	Relative display value	:TRANsmitter:OPTical:OUTPut:RATTenuator
		:TRANsmitter:OPTical:OUTPut:RATTenuator?
[17]	External ATT Factor	:TRANsmitter:OPTical:OUTPut:ATTFactor
		:TRANsmitter:OPTical:OUTPut:ATTFactor?
[18]	Power Monitor LED display	:TRANsmitter:OPTical:OUTPut:PMONitor:CALCulation?
[19]	Power Monitor	:TRANsmitter:OPTical:OUTPut:PMONitor?
[20]	Average	:TRANsmitter:OPTical:OUTPut:AVERage
		:TRANsmitter:OPTical:OUTPut:AVERage?

:TRANsmitter:OPTical: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. 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?

Response	<boolean> = <NR1 NUMERIC RESPONSE DATA>	
	0	Use of internal D-FF is OFF
	1	Use of internal D-FF is ON
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>

Parameter	<route> = <CHARACTER PROGRAM DATA>	
	REFerence	Reference
	STRessedeye	StressedEye (when any of MU181620A-x11, x12, and x13 is installed.)
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?

Response	<route> = <CHARACTER RESPONSE DATA>
	REF, STR
Function	Queries the optical output route.
Example	`> :TRANsmitter:OPTical:OUTPut:TESTtype?`
	`< REF`
Compatibility	Incompatible with existing models.

:TRANsmitter:OPTical:OUTPut:WAVelength <length>

Parameter	<length> = <CHARACTER PROGRAM DATA>	
	W_1310	1310 nm (when MU181620A-x01, x03, x11, or x13 is installed.)
	W_1550	1550 nm (when MU181620A-x02, x03, x11, or x13 is installed.)
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.	

:TRANsmitter:OPTical:OUTPut:WAVelength?

Response <length> = <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 <numeric> = <NR1 NUMERIC RESPONSE DATA>
0 Within specifications (LED: green)
1 Beyond specifications (LED: red)

Table 7.7.1.1-2 Specification Conformance

Wavelength[nm]	Test Type	Extinction Ratio[dB]	Specification Conformance
1310	Reference	4.0 to 10.0	0: Within specifications
		2.0 to 3.9, 10.1 to 11.0	1: Beyond specifications
1310	Stressed Eye	2.0 to 6.0	0: Within specifications
		1.0 to 1.9, 6.1 to 7.0	1: Beyond specifications
1550	Reference	6.0 to 10.0	0: Within specifications
		2.0 to 5.9, 10.1 to 11.0	1: Beyond specifications
1550	Stressed Eye	2.0 to 5.0	0: Within specifications
		1.0 to 1.9, 5.1 to 7.0	1: Beyond specifications

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 <numeric> = <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.

:TRANsmitter:OPTical:OUTPut:EXTinction:RATio?

Response	<numeric> = <NR2 NUMERIC RESPONSE DATA>
	1.0 to 11.0 1.0 to 11.0 dB
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>

Parameter	<boolean> = <BOOLEAN PROGRAM DATA>
	OFF or 0 Offset OFF
	ON or 1 Offset ON
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?

Response	<boolean> = <NR1 NUMERIC RESPONSE DATA>
	0 Offset OFF
	1 Offset ON
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.

: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

Format	Description
"XX.X"	0.1 to 14.0 dB / 0.1 dB Step
"-----"	When no data corresponds to the query

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.

:TRANsmitter:OPTical:OUTPut:CONTrol <power>

Parameter	<power> = <CHARACTER PROGRAM DATA>	
	OFF	OFF
	POWer	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.

:TRANsmitter:OPTical:OUTPut:POWer <numeric>

Parameter	<numeric> = <DECIMAL NUMERIC PROGRAM DATA> When Wavelength is set to 1310 nm while Unit is set to dBm: –20.00 to –4.00 –20.00 to –4.00 dBm / 0.01 dBm Step When Wavelength is set to 1310 nm while Unit is set to μW: 10 to 398 10 to 398 μW / 1 μW Step When Wavelength is set to 1550 nm while Unit is set to dBm: –20.00 to –2.00 –20.00 to –2.00 dBm / 0.01 dBm Step When Wavelength is set to 1550 nm while Unit is set to μW: 10 to 630 10 to 630 μW / 1 μW Step
Function	Sets the optical output power. Valid only when the control method of optical output power is set to 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?

Response	<numeric> = <NR2 NUMERIC RESPONSE DATA> When Unit is set to dBm: –20.00 to –2.00 –20.00 to –2.00 dBm When Unit is set to μW: 10 to 630 10 to 630 μW
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.

: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?

Response	<boolean> = <NR1 NUMERIC RESPONSE DATA>
	0 Attenuation reference OFF
	1 Attenuation reference ON
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>

Parameter	<numeric> = <DECIMAL NUMERIC PROGRAM DATA>
	When Wavelength is set to 1310 nm:
	–16.00 to 16.00 –16.00 to 16.00 dB / 0.01 dB Step
	When Wavelength is set to 1550 nm:
	–18.00 to 18.00 –18.00 to 18.00 dB / 0.01 dB Step
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?

Response	<numeric> = <NR2 NUMERIC RESPONSE DATA>
	–18.00 to 18.00 –18.00 to 18.00 dB
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>
	0.00 to 40.00 0.00 to 40.00 dB/0.01 dB Step
Function	Sets the attenuation when an optical output signal passes through an external attenuator.
Example	To set the attenuation of the external attenuator to 21.16 dB: `> :TRANsmitter:OPTical:OUTPut:ATTFactor 21.16`
Compatibility	Incompatible with existing models.

:TRANsmitter:OPTical:OUTPut:ATTFactor?

Response	<numeric> = <NR2 NUMERIC RESPONSE DATA>
	0.00 to 40.00 0.00 to 40.00 dB
Function	Queries the attenuation of the external attenuator.
Example	`> :TRANsmitter:OPTical:OUTPut:ATTFactor?` `< 21.16`
Compatibility	Incompatible with existing models.

:TRANsmitter:OPTical:OUTPut:PMONitor:CALCulation?

Response	<numeric> = <NR1 NUMERIC RESPONSE DATA>
	0 Average value calculation is not being performed (LED off)
	1 Average value is being calculated (LED on)
Function	Queries whether the Power Monitor value is being averaged.
Example	`> :TRANsmitter:OPTical:OUTPut:PMONitor:CALCulation?` `< 1`
Compatibility	Incompatible with existing models.

:TRANsmitter:OPTical:OUTPut:PMONitor?

Response <string> = <STRING RESPONSE DATA>

Table 7.7.1.1-4 Response

Format	Description
"XXX.XX"	XXX.XX dBm
"XXX"	XXX μW
"0>"	When Unit is set to dBm while the value is larger than 0.0 dBm
"<-18"	When Unit is set to dBm while the value is less than –18.0 dBm
"<15"	When Unit is set to μW while the value is less than 15 μW
"1000>"	When Unit is set to μW while the value is larger than 1000 μW
"-----"	When no data corresponds to the query

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

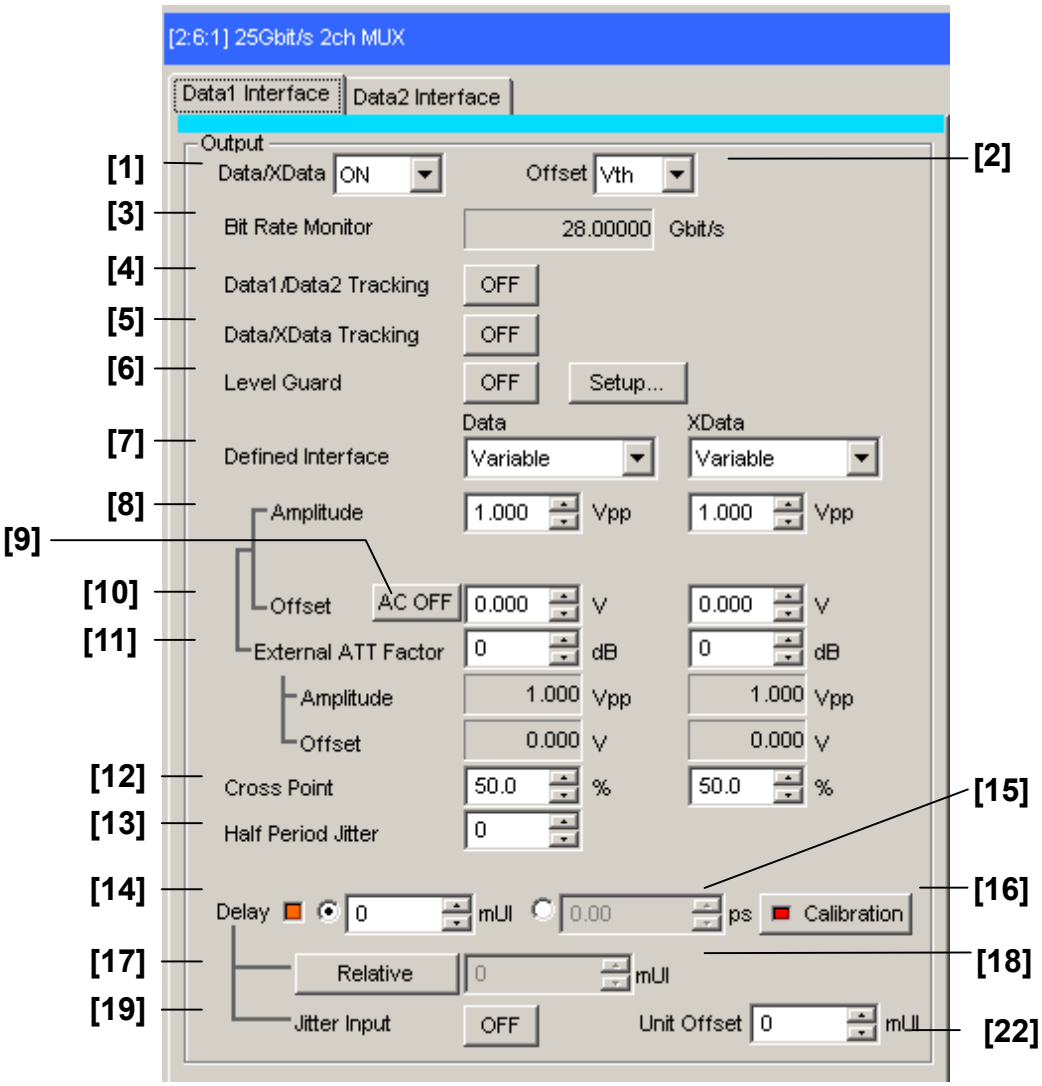


Figure 7.8.1.1-1 Data Interface setting field

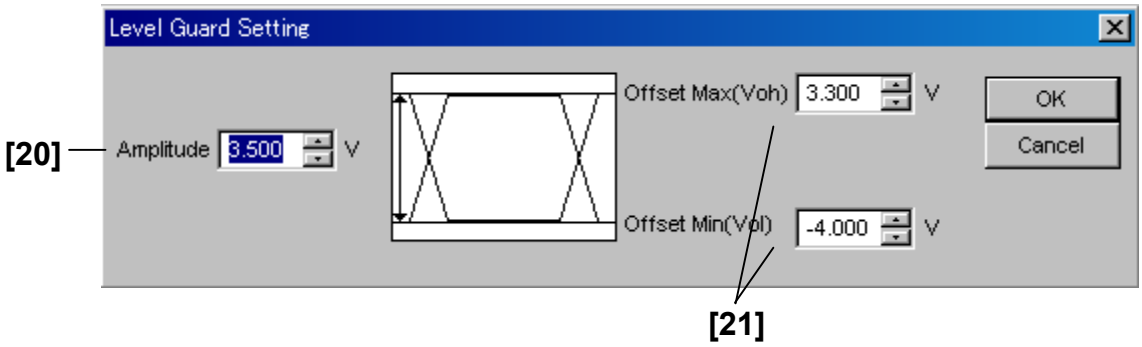


Figure 7.8.1.1-2 Level Guard setting field

Table 7.8.1.1-1 Data Interface setting commands

No.	Setting Items	Commands
[1]	Data/Xdata ON/OFF	:MUX:DATA:OUTPut
		:MUX:DATA:OUTPut?
[2]	Offset	:MUX:OUTPut:OFFSet
		:MUX:OUTPut:OFFSet?
[3]	Bit Rate Monitor	:MUX:OUTPut:BMONitor?
[4]	Data1/Data2 Tracking	:MUX:DATA:INTerface:TRACking
		:MUX:DATA:INTerface:TRACking?
[5]	Data/XData Tracking	:MUX:DATA:TRACking
		:MUX:DATA:TRACking?
[6]	Level Guard	:MUX:DATA:LEVGuard
		:MUX:DATA:LEVGuard?
[7]	Defined Interface	:MUX:DATA:LEVel
		:MUX:DATA:LEVel?
[8]	Amplitude	:MUX:DATA:AMPLitude
		:MUX:DATA:AMPLitude?
[9]	AC ON/OFF	:MUX:DATA:AOFFset
		:MUX:DATA:AOFFset?
[10]	Offset	:MUX:DATA:OFFSet
		:MUX:DATA:OFFSet?
[11]	External ATT Factor	:MUX:DATA:ATTFactor
		:MUX:DATA:ATTFactor?
[12]	Cross Point	:MUX:DATA:CPOint
		:MUX:DATA:CPOint?
[13]	Half Period Jitter	:MUX:DATA:HPJitter
		:MUX:DATA:HPJitter?
[14]	Delay (mUI settings)	:MUX:DATA:UIPadjust
		:MUX:DATA:UIPadjust?
[15]	Delay (ps settings)	:MUX:DATA:PADJust
		:MUX:DATA:PADJust?
[16]	Calibration	:MUX:DATA:PCALibration

Table 7.8.1.1-1 Data Interface setting commands (Cont'd)

No.	Setting Items	Commands
[17]	Relative	:MUX:DATA:RELative
		:MUX:DATA:RELative?
[18]	No label (Settings of Relative values)	:MUX:DATA:RDELay
		:MUX:DATA:RDELay?
[19]	Jitter Input	:MUX:DATA:JINPut
		:MUX:DATA:JINPut?
[20]	Amplitude	:MUX:DATA:LIMitter:AMPLitude
		:MUX:DATA:LIMitter:AMPLitude?
[21]	Offset Max/Min	:MUX:DATA:LIMitter:OFFSet
		:MUX:DATA:LIMitter:OFFSet?
[22]	Unit Offset	:MUX:DATA:UDELay:OFFSet
		MUX:DATA:UDELay:OFFSet?

: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 Data1Interface to ON:

```
> :MUX:DATA:OUTPut ON,1
```

Compatibility
Not backwards compatible

:MUX:DATA:OUTPut? [<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 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 Data1Interface 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 7.8.1.1-2 Response format

Format	Description
"XX.XXXXX"	XX.XXXXX Gbit/s
"-----"	When no data corresponds to the query

Function	Queries BitRate.

Example	To query BitRate of Data1Interface:
	`> :MUX:OUTPut:BMONitor? 1`
	`< "28.00000"`

Compatibility	Not backwards compatible

:MUX:DATA:INTerface:TRACking <boolean>

Parameter	<boolean> = <BOOLEAN PROGRAM DATA>
	OFF or 0, ON or 1
Function	Sets shared settings for Data Interface ON/OFF.
Example	To set shared settings for Data Interface to ON: `> :MUX:DATA:INTerface:TRACking ON`
Compatibility	Not backwards compatible

:MUX:DATA:INTerface:TRACking?

Response	<boolean> = <NR1 NUMERIC RESPONSE DATA>
	0 Shared settings for Data Interface are OFF
	1 Shared settings for Data Interface are ON
Function	Queries shared ON/OFF settings for Data Interface.

Example	To query shared ON/OFF settings for Data Interface:
	`> :MUX:DATA:INTerface:TRACking?`
	`< 1`

Compatibility	Not backwards compatible

:MUX:DATA:TRACking <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 shared settings for Data/XData output ON or OFF.
Example	To set shared settings for Data/XData of Data1Interface to ON: `> :MUX:DATA:TRACking ON,1`
Compatibility	Not backwards compatible

: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>]

Parameter	<port> = <CHARACTER PROGRAM DATA>	
	DATA, XDATa	
	<level> = <CHARACTER PROGRAM DATA>	
	VARiable	Variable (MU182020A-x10, x11, x12, x13, MU182021A-x10, x11, x12, x13)
	PCML	PCML level (MU182020A-x10, x11, x12, x13, MU182021A-x10, x11, x12, x13)
	NCML	NCML level (MU182020A-x10, x11, x12, x13, MU182021A-x10, x11, x12, x13)
	SCFL	SCFL level (MU182020A-x10, x11, x12, x13, MU182021A-x10, x11, x12, x13)
	NECL	NECL level (MU182020A-x10, x11, x12, x13, MU182021A-x10, x11, x12, x13)
	LVPecl	LVPECL level (MU182020A-x10, x11, x12, x13, MU182021A-x10, x11, x12, x13)
	LVDS400	LVDS 400Mv (MU182020A-x10, MU182021A-x10)
	<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 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	

: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

: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	<numeric> = <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

:MUX:DATA:AOFFset? [<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 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

:MUX:DATA:OFFSet <port>,<numeric>[,<interface>]

Parameter

<port> = <CHARACTER PROGRAM DATA>

DATA, XDATa

<numeric> = <DECIMAL NUMERIC PROGRAM DATA>

–2.000 to 3.300	–2.000 to 3.300 Voh/0.001V step (MU182020A-x10, x11, x12, x13, MU182021A-x10, x11, x12, x13)
–2.875 to 3.175	–2.875 to 3.175 Vth/0.001V step (MU182020A-x10, MU182021A-x10)
–3.000 to 3.050	–3.000 to 3.050 Vth/0.001V step (MU182020A-x11, MU182021A-x11)
–3.000 to 3.050	–3.000 to 3.050 Vth/0.001V step (MU182020A-x12, MU182021A-x12)
–3.000 to 3.050	–3.000 to 3.050 Vth/0.001V step (MU182020A-x13, MU182021A-x13)
–3.750 to 3.050	–3.750 to 3.050 Vol/0.001V step (MU182020A-x10, MU182021A-x10)
–4.000 to 2.800	–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

: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

: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

:MUX:DATA:HPJitter? [<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 data output crosspoint for specific port.

Example	To query Half Period Jitter of Data1Interface:
	`> :MUX:DATA:HPJitter? 1`
	`< 10`

Compatibility	Not backwards compatible

: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

:MUX:DATA:PADJust? [<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 shared settings for Data and Xdata output in ps units.	
Example	To query shared settings for Data output of Data1Interface:	
	`> :MUX:DATA:UIPadjust? 1`	
	`< 1000`	
Compatibility	Not backwards compatible	

:MUX:DATA:PCALibration [<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 Data and XData output phase.	
Example	To calibrate Data and XData output phase for Data1Interface:	
	`> :MUX:DATA:PCALibration 1`	
Compatibility	Not backwards compatible	

:MUX:DATA:RELative <boolean>[,<interface>]

Parameter	<boolean> = <BOOLEAN PROGRAM DATA>	
	OFF or 0	Reference OFF (AC output)
	ON or 1	Reference ON (DC 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 delay reference ON or OFF.	
Example	To set Data output delay reference for Data1Interface 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

:MUX:DATA:RDELay? [<unit>][,<interface>]

Parameter	<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.
Response	<numeric> = <NR2 NUMERIC RESPONSE DATA>	
Function	Queries data output delay reference value.	
Example	To query data output delay reference value for Data1Interface in mUI units:	
	`> :MUX:DATA:RDELay? UI,1`	
	`< -1000`	
Compatibility	Not backwards compatible	

:MUX:DATA:JINPut <boolean>[,<interface>]

Parameter	<boolean> = <BOOLEAN PROGRAM DATA>	
	OFF or 0	Jitter Input OFF
	ON or 1	Jitter Input 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 Jitter Input addition status.	
Example	To set Jitter Input addition status for Data1Interface to ON:	
	`> :MUX:DATA:JINPut ON,1`	
Compatibility	Not backwards compatible	

:MUX:DATA:JINPut? [<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 Jitter Input addition status.	
Example	To query Jitter Input addition status of Data1Interface:	
	`> :MUX:DATA:JINPut? 1`	
	`< ON`	
Compatibility	Not backwards compatible	

:MUX:DATA:LIMitter:AMPLitude <numeric>[,<interface>]

Parameter	<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/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	

:MUX:DATA:LIMitter:AMPLitude? [<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 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	

:MUX:DATA:LIMitter:OFFSet <max>,<min>[,<interface>]

Parameter	<max>,<min> = <DECIMAL NUMERIC PROGRAM DATA>	
	–2.000 to 3.300	–2.000 to 3.300 Voh/0.001V step (MU182020A-x10, x11, x12, x13, MU182021A-x10, x11, x12, x13)
	–3.750 to 3.050	–3.750 to 3.050 Vol/0.001V step (MU182020A-x10, MU182021A-x10)
	–4.000 to 2.800	–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/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:LIMitter:OFFSet 3.000,-2.000,1`	
Compatibility	Not backwards compatible	

:MUX:DATA:LIMitter:OFFSet? [<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	<max> = <NR2 NUMERIC RESPONSE DATA>	
	<min> = <NR2 NUMERIC RESPONSE DATA>	
Function	Queries Data/XData output offset limit.	
Example	To query Data/XData output offset limit of Data1Interface:	
	`> :MUX:DATA:LIMitter:OFFSet? 1`	
	`< 3.000, -2.000`	
Compatibility	Not backwards compatible	

:MUX:DATA:UDELay:OFFSet <numeric>

Parameter	<numeric> = <DECIMAL NUMERIC PROGRAM DATA> –64000 to 64000 –64000 to 64000 mUI/2 mUI Step
Function	Sets the unit of the Delay offset value.
Example	To set the Delay offset to 500 mUI: `> :MUX:DATA:UDELay:OFFSet 500`
Compatibility	Not backwards compatible

:MUX:DATA:UDELay:OFFSet?

Response	<numeric> = <NR1 NUMERIC RESPONSE DATA>
Function	Queries the Delay offset value.
Example	`> :MUX:DATA:UDELay:OFFSet?` `< 500`
Compatibility	Not backwards compatible

7.8.2 Commands related to Clock Interface tab

7.8.2.1 Clock Output setting field

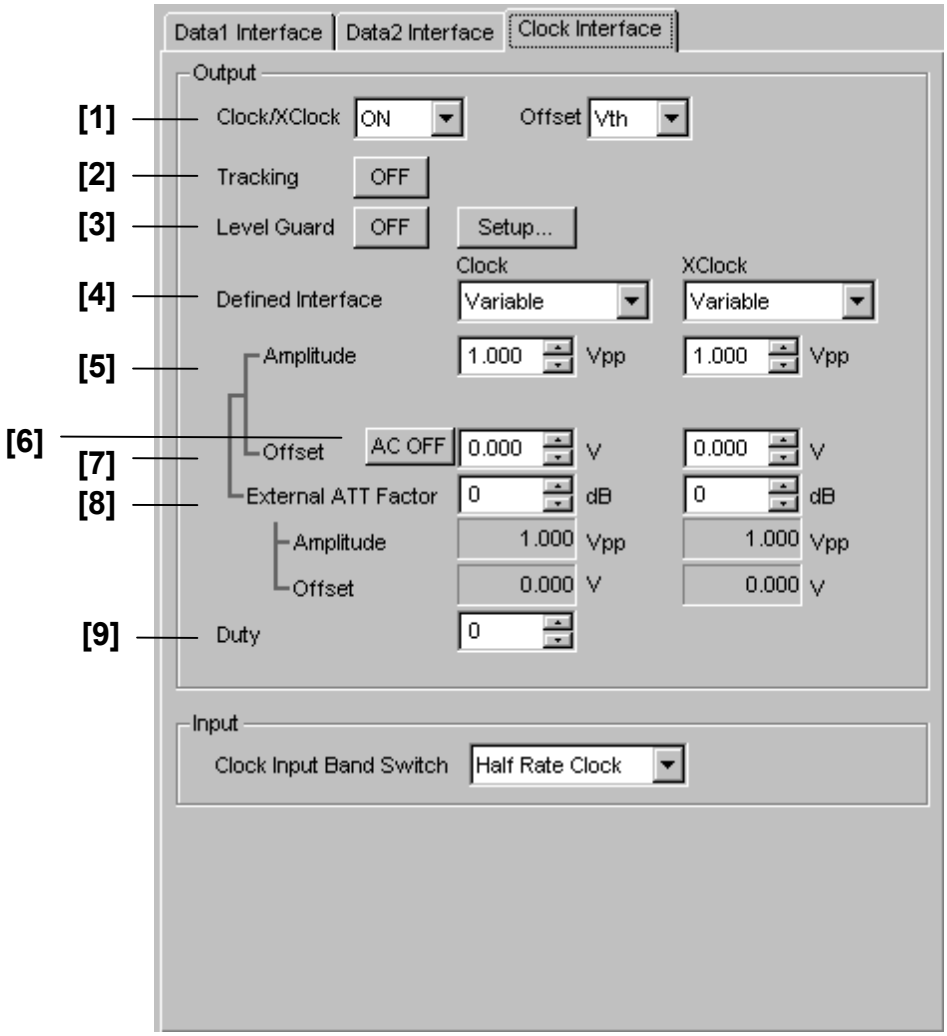


Figure 7.8.2.1-1 Clock Interface setting field

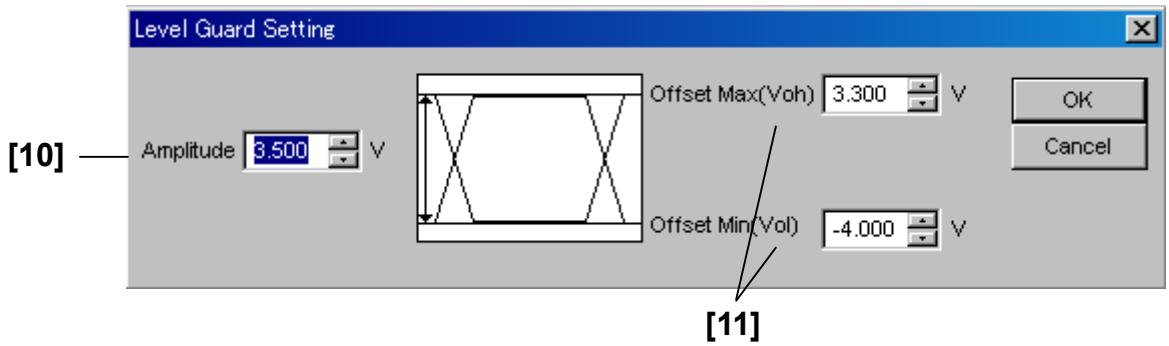


Figure 7.8.2.1-2 Level Guard setting field

Table 7.8.2.1-1 Clock Interface setting commands

No.	Setting Items	Commands
[1]	Clock/Xclock ON/OFF	:MUX:CLOCk:OUTPut
		:MUX:CLOCk:OUTPut?
[2]	Tracking	:MUX:CLOCk:TRACking
		:MUX:CLOCk:TRACking?
[3]	Level Guard	:MUX:CLOCk:LEVGuard
		:MUX:CLOCk:LEVGuard?
[4]	Defined Interface	:MUX:CLOCk:LEVel
		:MUX:CLOCk:LEVel?
[5]	Amplitude	:MUX:CLOCk:AMPLitude
		:MUX:CLOCk:AMPLitude?
[6]	AC ON・OFF	:MUX:CLOCk:AOFFset
		:MUX:CLOCk:AOFFset?
[7]	Offset (Specific Port)	:MUX:CLOCk:OFFSet
		:MUX:CLOCk:OFFSet?
[8]	External ATT Factor	:MUX:CLOCk:ATTFactor
		:MUX:CLOCk:ATTFactor?
[9]	Duty	:MUX:CLOCk:DUTY
		:MUX:CLOCk:DUTY?
[10]	Amplitude	:MUX:CLOCk:LIMitter:AMPLitude
		:MUX:CLOCk:LIMitter:AMPLitude?
[11]	Offset Max/Min	:MUX:CLOCk:LIMitter:OFFSet
		:MUX:CLOCk:LIMitter:OFFSet?

: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

:MUX:CLOCk: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:CLOCk:OUTPut?`
	`< 1`

Compatibility	Not backwards compatible

:MUX:CLOCk: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:CLOCk:TRACking ON`
Compatibility	Not backwards compatible

:MUX:CLOCk: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:CLOCk:TRACking?`
	`< 1`

Compatibility	Not backwards compatible

:MUX:CLOCk: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:CLOCk:LEVGuard ON`
Compatibility	Not backwards compatible

:MUX:CLOCk:LEVGuard?

Response	<numeric> = <NR1 NUMERIC RESPONSE DATA>	
	0	Clock/XClock output range limit OFF
	1	Clock/XClock output range limit ON
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	<port> = <CHARACTER PROGRAM DATA>	
	CLOCk, XCLock	
	<level> = <CHARACTER PROGRAM DATA>	
	VARiable	Variable
	PCML	PCML level
	NCML	NCML level
	SCFL	SCFL level
	NECL	NECL level
	LVPecl	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	<port> = <CHARACTER PROGRAM DATA>	
	CLOCk, XCLock	
Response	<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>

Parameter	<port> = <CHARACTER PROGRAM DATA>
	CLOCk, XCLock
	<numeric> = <DECIMAL NUMERIC PROGRAM DATA>
	0.500 to 2.000 0.500 to 2.000 V/0.002 V step
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>

Parameter	<port> = <CHARACTER PROGRAM DATA>
	CLOCk, XCLock
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>

Parameter	<boolean> = <BOOLEAN PROGRAM DATA>
	OFF or 0 Offset OFF (DC output)
	ON or 1 Offset ON (AC output)
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?

Response	<numeric> = <NR1 NUMERIC RESPONSE DATA>
	0 Offset OFF
	1 Offset ON
Function	Queries clock output offset ON/OFF.
Example	To query clock output offset of Clock Interface:
	`> :MUX:CLOCk:AOFFset?`
	`< 1`
Compatibility	Not backwards compatible

:MUX:CLOCk:OFFSet <port>,<numeric>

Parameter	<port> = <CHARACTER PROGRAM DATA> CLOCk, XCLock <numeric> = <DECIMAL NUMERIC PROGRAM DATA> –2.000 to 3.300 –2.000 to 3.300 Voh/0.001 V step –3.000 to 3.050 –3.000 to 3.050Vth/0.001 V step –4.000 to 2.800 –4.000 to 2.800Vol/0.001 V step
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 0 to 40 dB/1 dB Step
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

: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

:MUX:CLOCk:LIMitter: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:LIMitter:AMPLitude?`
	`< 2.000`

Compatibility	Not backwards compatible

:MUX:CLOCk:LIMitter: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:LIMitter:OFFSet 3.000,-2.000`
Compatibility	Not backwards compatible

:MUX:CLOCk:LIMitter: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:LIMitter:OFFSet?`
	`< 3.000,-2.000`

Compatibility	Not backwards compatible

7.8.2.2 Clock Input setting field

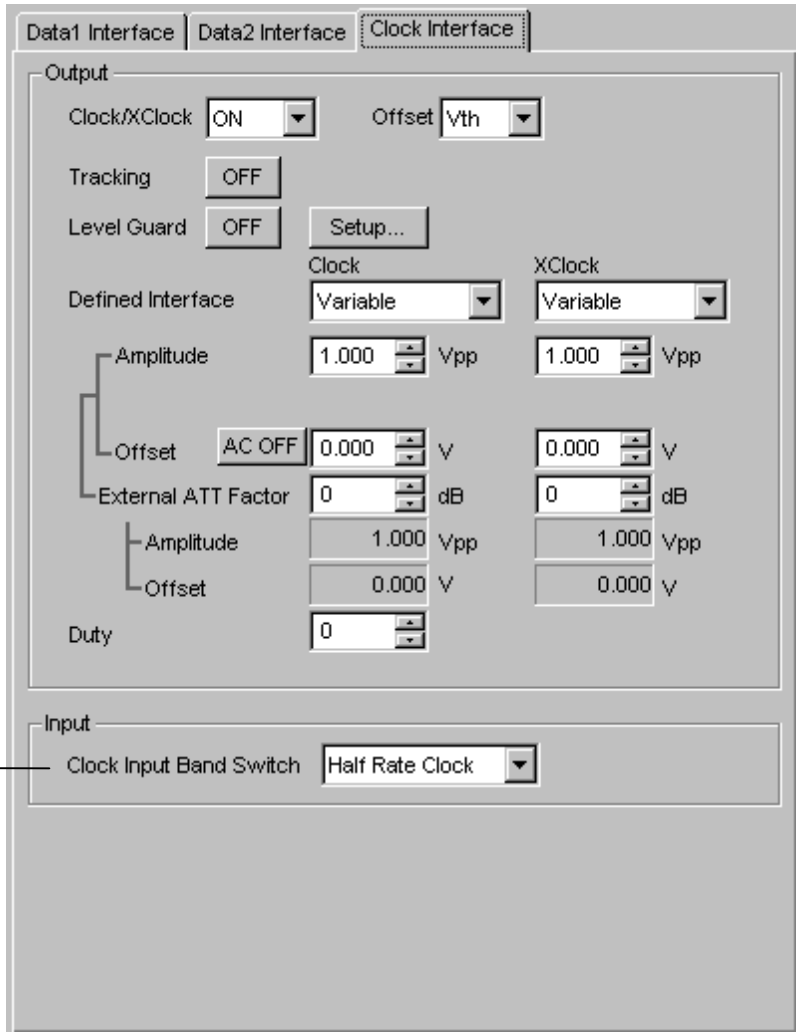


Figure 7.8.2.2-1 Clock Input setting field

Table 7.8.2.2-1 Clock Input setting commands

No.	Setting Items	Commands
[1]	Clock Input Band Switch	:MUX:CLOCk:BAND
		:MUX:CLOCk:BAND?

: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	`> :MUX:CLOCk:BAND?` `< HALF`
Compatibility	Not backwards compatible

7.8.3 Commands related to Emphasis tab

7.8.3.1 Emphasis setting field

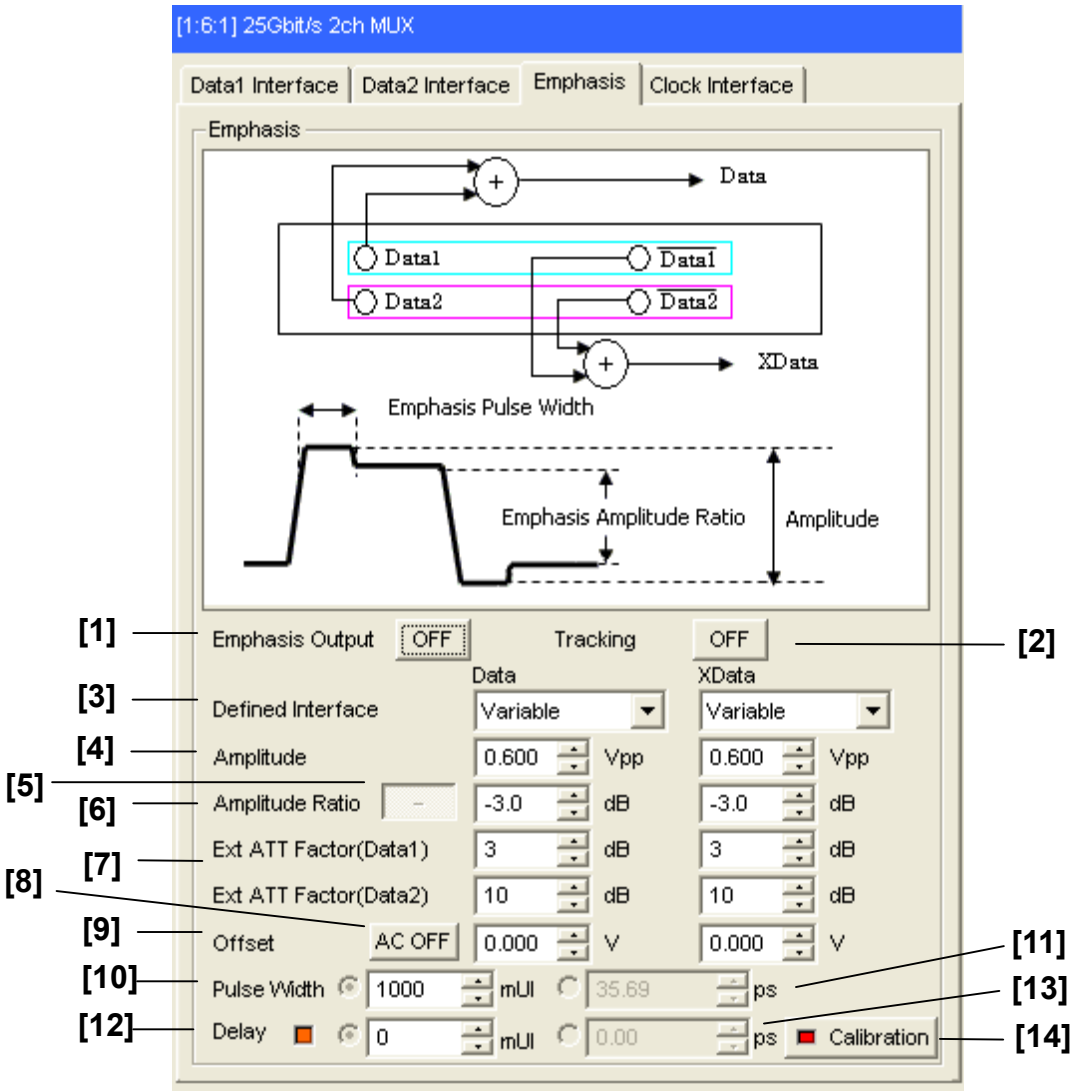


Figure 7.8.3.1-1 Emphasis setting

Table 7.8.3.1-1 Emphasis setting commands

No.	Setting Items	Commands
[1]	Emphasis output ON/OFF	:MUX:EMPHasis:SET
		:MUX:EMPHasis:SET?
[2]	Tracking	:MUX:EMPHasis:TRACking
		:MUX:EMPHasis:TRACking?
[3]	Defined Interface	:MUX:EMPHasis:LEVel
		:MUX:EMPHasis:LEVel?
[4]	Amplitude	:MUX:EMPHasis:AMPLitude
		:MUX:EMPHasis:AMPLitude?

Table 7.8.3.1-1 Emphasis setting commands (Cont'd)

No.	Setting Items	Commands
[5]	Amplitude Ratio ± setting button	:MUX:EMPHasis:RATio:SIGN
		:MUX:EMPHasis:RATio:SIGN?
[6]	Amplitude Ratio	:MUX:EMPHasis:AMPLitude:RATio
		:MUX:EMPHasis:AMPLitude:RATio?
[7]	External ATT Factor (Data1/Data2)	:MUX:EMPHasis:ATTFactor
		:MUX:EMPHasis:ATTFactor?
[8]	AC ON/OFF	:MUX:EMPHasis:AOFFset
		:MUX:EMPHasis:AOFFset?
[9]	Offset	:MUX:EMPHasis:OFFSet
		:MUX:EMPHasis:OFFSet?
[10]	Pulse Width (mUI setting)	:MUX:EMPHasis:PULSewidth:UIPadjust
		:MUX:EMPHasis:PULSewidth:UIPadjust?
[11]	Pulse Width (ps setting)	:MUX:EMPHasis:PULSewidth:PADJust
		:MUX:EMPHasis:PULSewidth:PADJust?
[12]	Delay (mUI setting)	:MUX:EMPHasis:UIPadjust
		:MUX:EMPHasis:UIPadjust?
[13]	Delay (ps setting)	:MUX:EMPHasis:PADJust
		:MUX:EMPHasis:PADJust?
[14]	Calibration	:MUX:EMPHasis:PCALibration

:MUX:EMPHasis:SET <boolean>

Parameter
<boolean> = <BOOLEAN PROGRAM DATA>
OFF or 0 Emphasis signal OFF
ON or 1 Emphasis signal ON

Function
Sets Emphasis signal ON/OFF.

Example
To set Emphasis signal to ON:

```
> :MUX:EMPHasis:SET ON
```

Compatibility
Not backwards compatible

:MUX:EMPHasis:SET?

Response
<numeric> = <NR1 NUMERIC RESPONSE DATA>
0 Emphasis signal OFF
1 Emphasis signal ON

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 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

:MUX:EMPHasis:AMPLitude <port>,<numeric>

Parameter	<port> = <CHARACTER PROGRAM DATA> DATA, XDATa <numeric> = <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	<numeric> = <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

: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 queries Emphasis value for Emphasis Data:
	`> :MUX:EMPHasis:AMPLitude:RATio? 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 Data1Interface:
	`> :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

: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>

Parameter	<numeric> = <DECIMAL NUMERIC PROGRAM DATA> –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.
Function	Sets Emphasis output phase 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 output phase to 80 ps: `> :MUX:EMPHasis:PADjust 80`
Compatibility	Not backwards compatible

:MUX:EMPHasis:PADJust?

Response	<numeric> = <NR2 NUMERIC RESPONSE DATA>
Function	Queries Emphasis output phase in ps units.
Example	`> :MUX:EMPHasis:UIPadjust?` `< 80`
Compatibility	Not backwards compatible

:MUX:EMPHasis:PCALibration

Parameter	None
Function	Calibrates Emphasis Delay.
Example	`> :MUX:EMPHasis:PCALibration`
Compatibility	Not backwards compatible

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

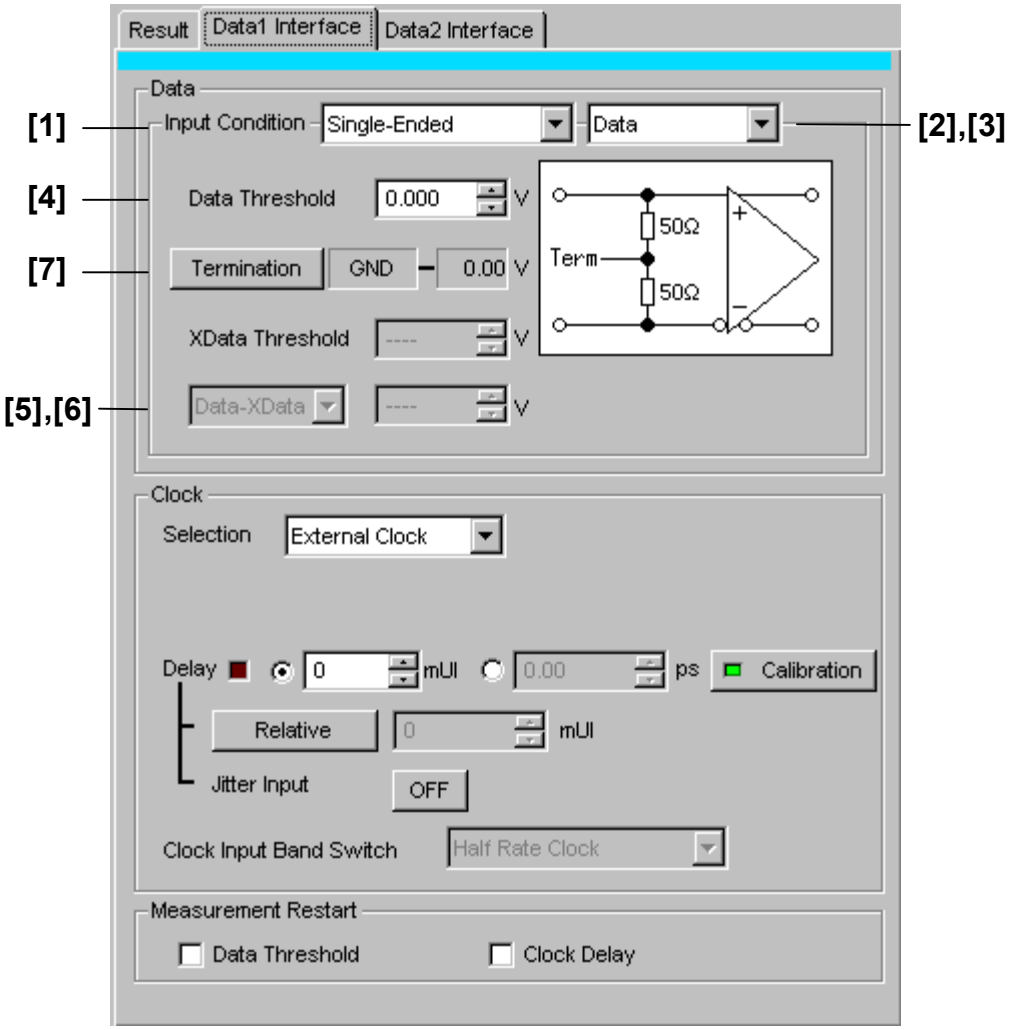


Figure 7.9.1.1-1 Data Interface setting field

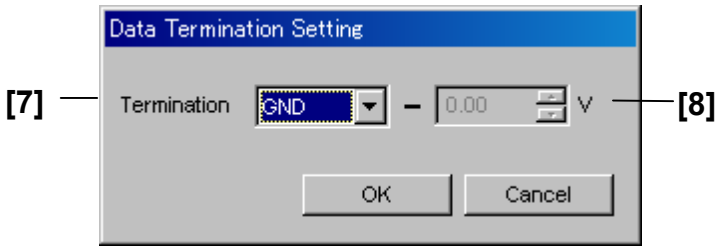


Figure 7.9.1.1-2 Termination Setting

Table 7.9.1.1-1 Data Interface setting commands

No.	Setting Items	Commands
[1]	Input Condition	:DEMux:DATA:INTerface
		:DEMux:DATA:INTerface?
[2]	No label (Differential setting)	:DEMux:DATA:DIFFerential
		:DEMux:DATA:DIFFerential?
[3]	No label (Differential reference signal setting)	:DEMux:DATA:SINGle
		:DEMux:DATA:SINGle?
[4]	Data Threshold X Data Threshold	:DEMux:DATA:THReshold
		:DEMux:DATA:THReshold?
[5]	No label (Differential reference signal setting)	:DEMux:DATA:DIFFerential:BASis
		:DEMux:DATA:DIFFerential:BASis?
[6]	No label (Differential data threshold setting)	:DEMux:DATA:DIFFerential:THReshold
		:DEMux:DATA:DIFFerential:THReshold?
[7]	Termination	:DEMux:DATA:TERMination
		:DEMux:DATA:TERMination?
[8]	No label (Termination voltage setting)	:DEMux:DATA:TLEVel
		:DEMux:DATA:TLEVel?

: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:

```
> :DEMux:DATA:INTerface SINGle,1
```

Compatibility

Not backwards compatible

: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	

:DEMux:DATA:THReshold? <port>[,<interface>]

Parameter	<port> = <CHARACTER PROGRAM DATA> DATA, XDATa (At Single, input the selected port.) <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 date input threshold value for specific port.

Example	To query threshold value for Data1Interface input port:
	`> :DEMux:DATA:THReshold? DATA,1`
	`< -3.000`

Compatibility	Not backwards compatible

:DEMux:DATA:DIFFerential:BASis <basis>[,<interface>]

Parameter	<basis> = <CHARACTER PROGRAM DATA> DATA Data set as reference XDATa XData set as reference <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 signal for data input threshold value.
Example	To set differential reference signal for Data1Interface threshold value to Data: `> :DEMux:DATA:DIFFerential:BASis DATA,1`
Compatibility	Not backwards compatible

:DEMux:DATA:DIFFerential:BASis? [<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	<basis> = <CHARACTER RESPONSE DATA> 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:
	`> :DEMux:DATA:DIFFerential:BASis? 1`
	`< DATA`

Compatibility	Not backwards compatible

:DEMux:DATA:DIFFerential:THReshold <numeric>[,<interface>]

Parameter	<numeric> = <DECIMAL NUMERIC PROGRAM DATA>	
	–1.500 to 1.500	–1.500 to 1.500 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 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	

:DEMux:DATA:DIFFerential:THReshold? [<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 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	

: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

7.9.2 Commands related to Clock Interface tab

7.9.2.1 Clock Interface setting field

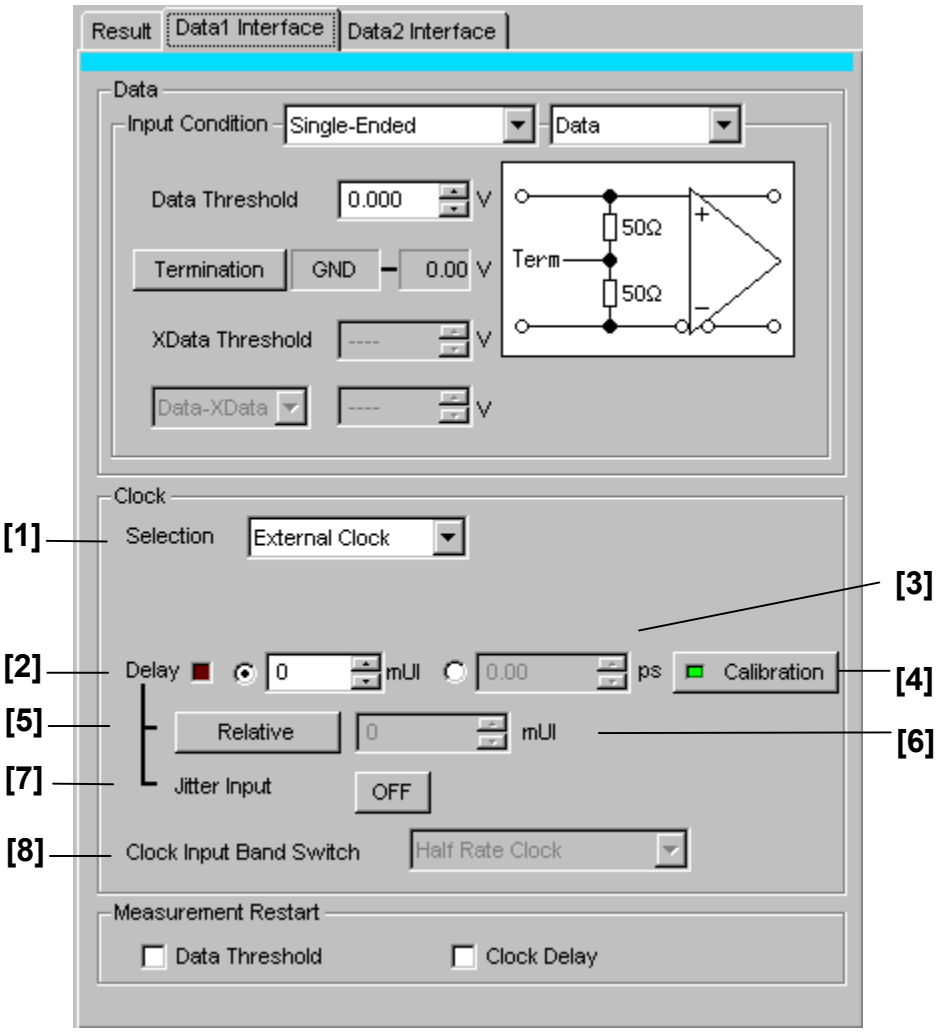


Figure 7.9.2.1-1 Clock Interface setting field

Table 7.9.2.1-1 Clock Interface setting commands

No.	Setting Items	Commands
[1]	Selection	:DEMux:CLOCk:SELection
		:DEMux:CLOCk:SELection?
[2]	Delay (mUI units)	:DEMux:CLOCk:UIDelay
		:DEMux:CLOCk:UIDelay?
[3]	Delay (ps units)	:DEMux:CLOCk:PDELay
		:DEMux:CLOCk:PDELay?
[4]	Calibration	:DEMux:CLOCk:CALibration
[5]	Relative	:DEMux:CLOCk:RELative
		:DEMux:CLOCk:RELative?

Table 7.9.2.1-1 Clock Interface setting commands (Cont'd)

No.	Setting Items	Commands
[6]	No label (Settings of relative values)	:DEMux:CLOCk:RDELay
		:DEMux:CLOCk:RDELay?
[7]	Jitter Input	:DEMux:CLOCk:JINPut
		:DEMux:CLOCk:JINPut?
[8]	Clock Input Band Switch	:DEMux:CLOCk:BAND
		:DEMux:CLOCk:BAND?

:DEMux:CLOCk:SELection <sel>

Parameter
<sel> = <CHARACTER PROGRAM DATA>
EXTernal External Clock

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>
EXT External Clock

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> 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 in mUI units.

Example	To query Data1Interface clock phase variation in mUI units:
	`> :DEMux:CLOCk:DELay? 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> 1 to 2 Data Interface for Data1, and Data2 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> 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 and units.

Example	To query Data1Interface clock phase variation in ps units:
	`> :DEMux:CLOCk:PDELay? 1`
	`< -40`

Compatibility	Not backwards compatible

: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

:DEMux:CLOCk:JINPut <boolean>[,<interface>]

Parameter	<boolean> = <BOOLEAN PROGRAM DATA>	
	0 or OFF	Jitter Input OFF
	1 or ON	Jitter Input 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 Jitter Input ON/OFF.	
Example	To set Data1Interface JitterInput to ON:	
	`> :DEMux:Clock:JINPut ON,1`	
Compatibility	Not backwards compatible	

:DEMux:CLOCk:JINPut? [<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	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>

Parameter	<sw> = <CHARACTER PROGRAM DATA>	
	HALFrate	Selects Half Clock Rate
	FULLrate	Selects Full Clock Rate
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?

Response	<sw> = <CHARACTER RESPONSE DATA>	
	HALFrate	Selects Half Clock Rate
	FULLrate	Selects Full Clock Rate
Function	Queries Clock Input Band Switch.	
Example	To query Clock Input Band Switch:	
	> :DEMux:CLOCk:BAND?	
	< HALF	
Compatibility	Not backwards compatible	

7.9.2.2 Measurement Restart setting field

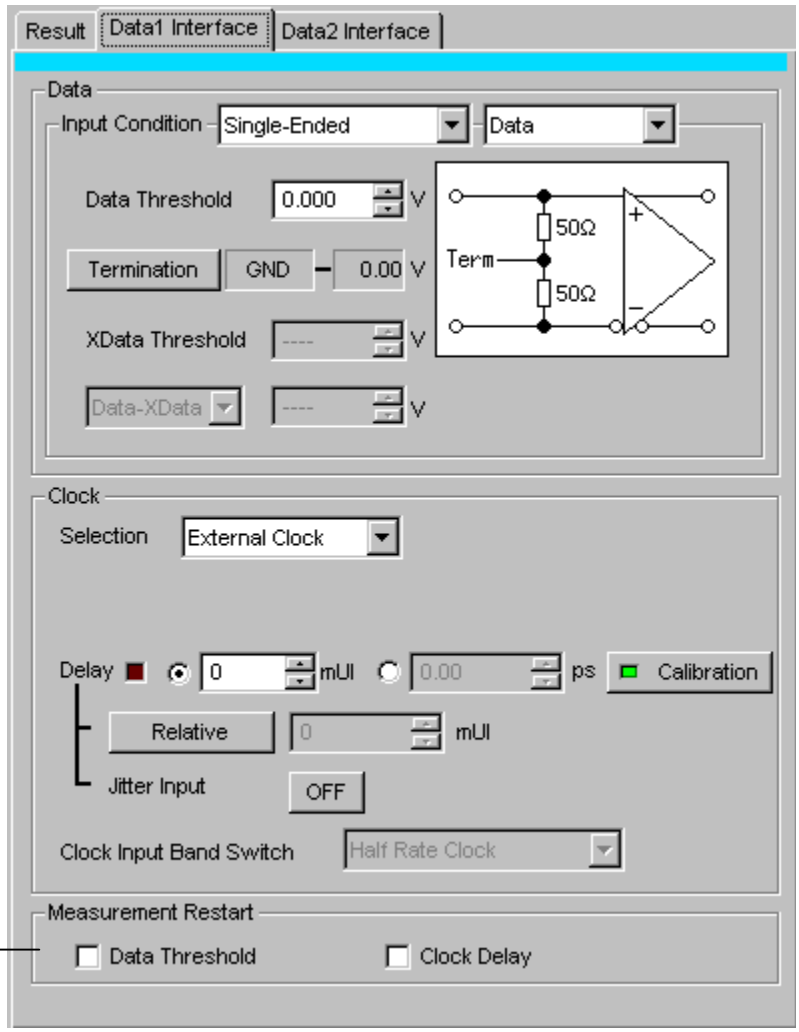


Figure 7.9.2.2-1 Measurement Restart setting field

Table 7.9.2.2-1 Measurement Restart setting commands

No.	Setting Items	Commands
[1]	Measurement Restart	:DEMux:MEASure:MREStart
		:DEMux:MEASure:MREStart?

:DEMux:MEASure:MREStart <data>,<clock> [,<interface>]

Parameter	<data> = < BOOLEAN PROGRAM DATA >	
	0 or OFF	Set measurement restart at Data Threshold change to OFF
	1 or ON	Set measurement restart at Data Threshold change to ON
	<clock> = < BOOLEAN PROGRAM DATA >	
	0 or OFF	Set measurement restart at Clock Delay change to OFF
	1 or ON	Set measurement restart at Clock Delay change to 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 measurement restart conditions.	
Example	To set Data1Interface measurement restart conditions: `> :DEMux:MEASure:MREStart 0,0,1`	
Compatibility	Not backwards compatible	

:DEMux:MEASure:MREStart? [<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	<data> = <NR1 NUMERIC RESPONSE DATA>	
	<clcok> = <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:MEASure:MREStart?`
	`< 0,0`	

Compatibility	Not backwards compatible	

7.9.3 Commands related to Result tab

7.9.3.1 Result setting field

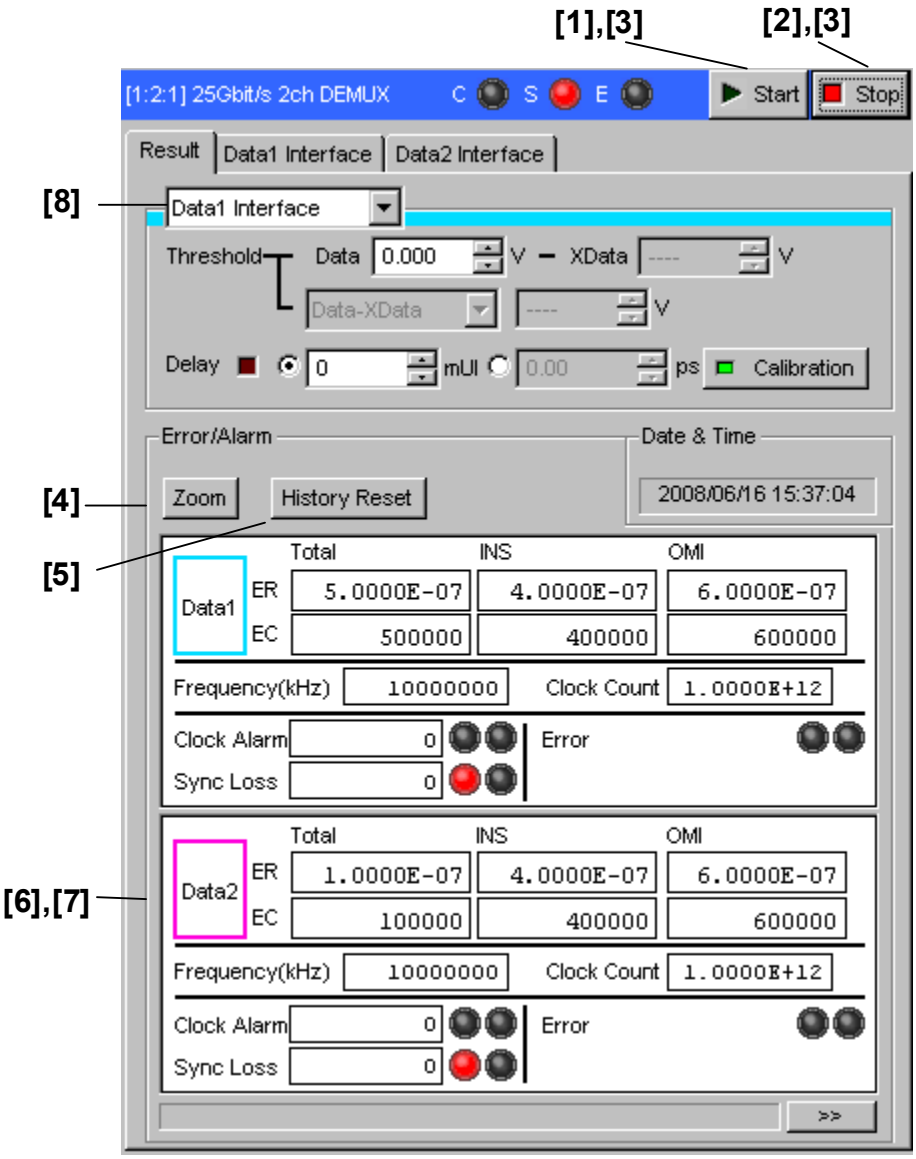


Figure 7.9.3.1-1 Result tab

Table 7.9.3.1-1 Result setting commands

No.	Setting Items	Commands
[1]	Start	:DEMux:MEASure:STARt
[2]	Stop	:DEMux:MEASure:STOP
[3]	Query measurement conditions	:DEMux:MEASure:EALarm:STATe?
[4]	Zoom	:DEMux:RESult:ZOOM
		:DEMux:RESult:ZOOM?
[5]	History Reset	:DEMux:RESult:EALarm:HRESet
[6]	No label (Query measurement data of error alarm)	:DEMux:CALCulate:DATA:EALarm?
[7]	No label (Query window items)	:DEMux:CALCulate:DATA:MONitor?
[8]	Display measurement items	:DEMux:DISPlay:SETTing

:DEMux:MEASure:STARt

Function — Starts MU181040A BER measurement in same unit and restarts during measurement.

Example — To start MU181040A BER measurement in same unit:

```
> :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:

```
> :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:

```
> :DUMux:MEASure:EALarm:STATe?
< 0
```

Compatibility — Not backwards compatible

: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

: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.

<interface> = <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

Form	Format	Description
Form1 Integer type	"XXXXXXX"	For 0 to 9999999
	"X.XXXXEXX"	For 1.0000E07 to 9.9999E16
	"-------"	When no data corresponds to the query
Form2 Fraction type	"X.XXXXE-XX"	For 0.0000E–16 to 1.0000E00
	"-------"	When no data corresponds to the query
Form3 % type	"XXX.XXXXX"	For 0.0000 to 100.0000
	"-------"	When no data corresponds to the query
Form4 Frequency type	"XXXXXXXXX"	For 0 to MAX(Hz)
	"---------"	When no data corresponds to the query
Form5 Voltage type	"XX.XXX,XX.XXX"	For –4.000 to +4.000(V)
	"----,-----"	When no data corresponds to the query
Form6 Phase type	"xxxxx,xxx.xx"	mUI units, ps units
	"-----,----"	When no data corresponds to the query

Function

Queries measured data for parameter.

Example

To query Data Interface1 current measured data total error rate:

```
> :DEMux:CALCulate:DATA:EALarm? "CURRent:ER:TOTal",1
< : "0.0000E-16"
```

To query 25G × 2 chTotal current data Threshold measured data

```
> :DEMux:CALCulate:DATA:EALarm? "CURRent:THReshold",3
< "-4.000,------ "
```

(in case of Single-end)

Compatibility

Not backwards compatible

Table 7.9.3.1-3 Parameter

Form		<result>	Format
Error Rate	Total INS OMI	ER:TOTal ER:INSertion ER:OMISsion	Form2
Error Count	Total INS OMI	EC:TOTal EC:INSertion EC:OMISsion	Form1
Error Rate	Transition Non Transition	ER:TRANsition ER:NONTransition	Form2
Error Count	Transition Non Transition	EC:TRANsition EC:NONTransition	Form1
%EFI	Total	EFI:TOTal	Form3
EI	Total	EI:TOTal	Form1
Frequency(kHz)		FREQuency	Form4
Clock Count		CC:TOTal	Form1
Clock Alarm		AINTerval:CALarm	Form1
Sync Loss		AINTerval:PSLoss	Form1
Data Threshold		THReshold	Form5
Data Phase		PHASe	Form6
Threshold EI	>1.0E-3	THReshold:EI:TOTal:E_3	Form1
	>1.0E-4	THReshold:EI:TOTal:E_4	
	>1.0E-5	THReshold:EI:TOTal:E_5	
	>1.0E-6	THReshold:EI:TOTal:E_6	
	>1.0E-7	THReshold:EI:TOTal:E_7	
	>1.0E-8	THReshold:EI:TOTal:E_8	
	≤1.0E-8	THReshold:EI:TOTal:UE_8	
Threshold %EFI	>1.0E-3	THReshold:EFI:TOTal:E_3	Form3
	>1.0E-4	THReshold:EFI:TOTal:E_4	
	>1.0E-5	THReshold:EFI:TOTal:E_5	
	>1.0E-6	THReshold:EFI:TOTal:E_6	
	>1.0E-7	THReshold:EFI:TOTal:E_7	
	>1.0E-8	THReshold:EFI:TOTal:E_8	
	≤1.0E-8	THReshold:EFI:TOTal:UE_8	
G.821	ES	G821:ES2:TOTal	Form1
	EFS	G821:EFS2:TOTal	
	SES	G821:SES2:TOTal	
	DM	G821:DM2:TOTal	
	US	G821:US2:TOTal	
	%ES	G821:ES:TOTal	Form3
	%EFS	G821:EFS:TOTal	
	%SES	G821:SES:TOTal	
	%DM	G821:DM:TOTal	
	%US	G821:US:TOTal	

:DEMux:DISPlay:SETTing <setting>

Parameter
<setting> = <CHARACTER PROGRAM DATA>

DATA1	Display Data1 Interface items
DATA2	Display Data2 Interface items
AUToadjust	Display AutoAdjust items

Function
Sets displayed items on the Result tab:

Example
To display the Data1 Interface Items:

```
> :DEMux:DISPlay:SETTing DATA1
```

Compatibility
Not backwards compatible

:DEMux:CALCulate:DATA:MONitor? <item>

Parameter
<item> = <STRING PROGRAM DATA>

"BIT:TOTal"	Bit Error (Total Error)
"PSLoss"	Pattern Sync Loss
"CALarm"	Clock Alarm

Response
<string> = <STRING RESPONSE DATA>

Table 7.9.3.1-4 Response format

Form	Format	Description
Form1	"Occur"	When an alarm occurs
	"Not Occur"	When no alarm occurs
	"-------"	When no data corresponds to the query

Function
Queries real-time occurrence status of monitor items specified by parameters.

Example
To query Bit Error occurrence status:

```
> :DEMux:CALCulate:DATA:MONitor? "BIT TOTal"
< "Occur"
```

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 7.10.1-1 MU181500B Command List

No.	Command Header 1	Command Header 2	Command Header 3	Command Header 4	Command/ Query	Remarks
1	:OUTPut	:AUX	:JCONdition		Q	
2			:REFClock		C/Q	
3				:MONitor	Q	
4			:SELect		C/Q	
5			:SUBRateclock		C/Q	
6				:AMPLitude	C/Q	
7				:MONitor	Q	
8		:CLOCk	:FREQuency		C/Q	Synthesizer compatible
9			:OFFset	:PPM	C/Q	Synthesizer compatible
10		:RCLock	:SELect		C/Q	Synthesizer compatible
11	:SOURce	:JITTer	:BUJ	:AMPLitude	C/Q	
12				:BITRate	C/Q	
13				:ENABle	C/Q	
14				:LPFilter	C/Q	
15				:PRBS	C/Q	
16			:EXTJitter	:ENABle	C/Q	
17			:RJ	:AMPLitude	C/Q	
18				:DEFault	C	
19				:ENABle	C/Q	
20				:FILTer	C/Q	
21				:HFAMplitude	C/Q	
22				:HPFilter	C/Q	
23				:LFAMplitude	C/Q	
24				:LPFilter	C/Q	
25				:MONitor	Q	

Table 7.10.1-1 MU181500B Command List (Cont'd)

No.	Command Header 1	Command Header 2	Command Header 3	Command Header 4	Command/ Query	Remarks
26	:SOURce	:JITTer	:SJ[2]	:AMPLitude	C/Q	
27				:ENABle	C/Q	
28				:FREQuency	C/Q	
29			:SSC	:DEViation	C/Q	
30				:ENABle	C/Q	
31				:FREQuency	C/Q	
32				:TYPE	C/Q	
33		:OUTPut	:PATA	:JOVerload	Q	
34				:MONitor	Q	
35				:SELect	C/Q	
36	:SYSTem	:INPut	:CSELect		C/Q	
37				:MODule	Q	
38		:MMEMory	:RECall		C	
39			:STORe		C	

7.10.2 Clock Source Settings Commands

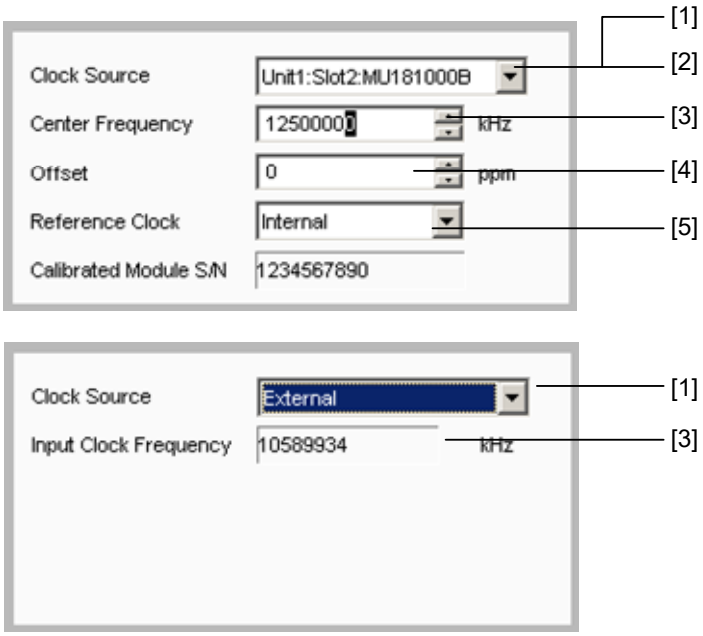


Figure 7.10.2-1 Clock Source Settings

Table 7.10.2-1 Clock Source Settings Commands

No.	Setting item	Command
[1]	Clock Source	:SYSTem:INPut:CSELect
		:SYSTem:INPut:CSELect?
[2]	Queries system built-in synthesizer	:SYSTem:INPut:CSELect:MODule?
[3]	Center Frequency/Input Clock Frequency	:OUTPut:CLOCk:FREQuency
		:OUTPut:CLOCk:FREQuency?
[4]	Offset	:OUTPut:CLOCk:OFFSet:PPM
		:OUTPut:CLOCk:OFFSet:PPM?
[5]	Reference Clock	:OUTPut:RCLock:SELect
		:OUTPut:RCLock:SELect?

: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.	

: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 100,000 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

: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

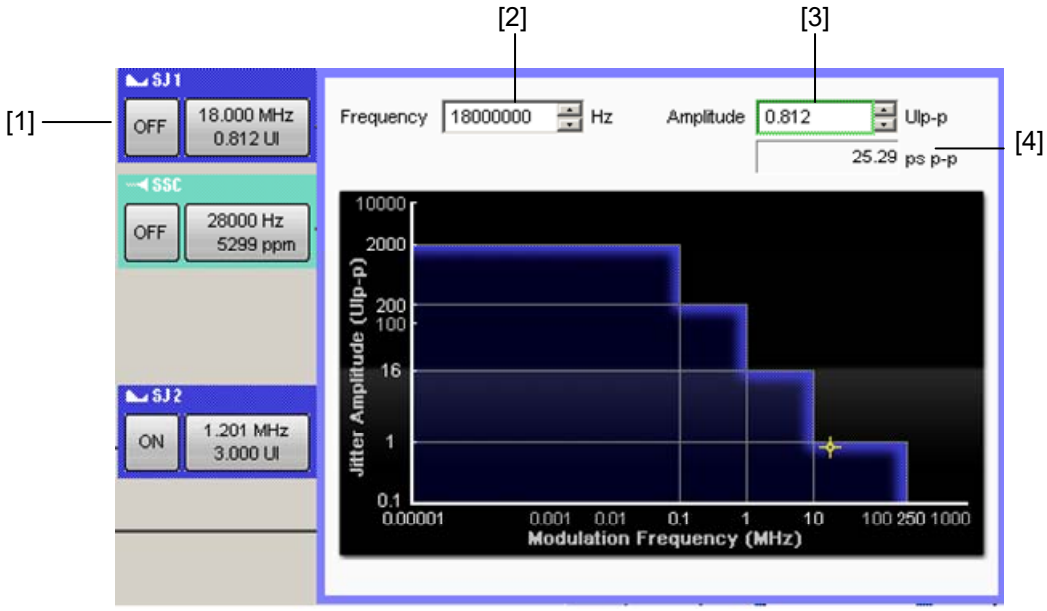


Figure 7.10.3.1-1 SJ/SJ2 Settings

Table 7.10.3.1-1 SJ/SJ2 Setting Commands

No.	Setting item	Command
[1]	SJ/SJ2 ON/OFF	:SOURce:JITTer:SJ[2]:ENABle :SOURce:JITTer:SJ[2]:ENABle?
[2]	SJ/SJ2 Modulation Frequency	:SOURce:JITTer:SJ[2]:FREQuency :SOURce:JITTer:SJ[2]:FREQuency?
[3]	SJ/SJ2 Jitter Amplitude	:SOURce:JITTer:SJ[2]:AMPLitude :SOURce:JITTer:SJ[2]:AMPLitude?
[4]	SJ/SJ2 Jitter Amplitude Query	:SOURce:JITTer:SJ[2]:PSAMplitude?

: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:	
	:If [2] of SJ[2] is omitted, SJ setting is assumed.	
Example	To set SJ setting to ON:	
	`> :SOURce:JITTer:SJ:ENABle ON`	
Compatibility	Incompatible with existing models.	

:SOURce:JITTer:SJ[2]:ENABle?

Response	<numeric> = <NR1 NUMERIC RESPONSE DATA>	
	0	SJ/SJ2 OFF
	1	SJ/SJ2 ON
Function	Queries SJ or SJ2 ON/OFF setting.	
Example	To query SJ ON/OFF setting:	
	`> :SOURce:JITTer:SJ:ENABle?`	
	`< 1`	
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.	

: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>

Data Generation / Modulation Frequency	Half-rate(MUX)		Full-rate(PPG), Full-rate(MUX)		Quarter-rate (MUX)	
	Setting Range (UIp-p)	Step (UI)	Setting Range (UIp-p)	Step (UI)	Setting Range (UIp-p)	Step (UI)
10 Hz to 1 MHz	0 to 50.000	0.002	0 to 40.000	0.001	0 to 50.000	0.004
1.001 MHz to 10 MHz	0 to 10.000	0.002	0 to 8.000	0.001	0 to 10.000	0.004
10.01 MHz to 250 MHz	0 to 0.550	0.002	0 to 0.500	0.001	0 to 0.548	0.004

Data Generation / Modulation Frequency	32G PPG		64G MUX	
	Setting Range (UIp-p)	Setting Range (UIp-p)	Setting Range (UIp-p)	Setting Range (UIp-p)
10 Hz to 100 kHz	0 to 2000.000	0.002	0 to 2000.000	0.002
100.1 kHz to 1 MHz	0 to 200.000	0.002	0 to 200.000	0.002
1.001 MHz to 10 MHz	0 to 16.000	0.002	0 to 16.000	0.002
10.01 MHz to 250 MHz	0 to 1.0	0.002	0 to 1.0	0.002

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.

:SOURce:JITTer:SJ[2]:AMPLitude?

Response	<numeric> = <NR2 NUMERIC RESPONSE DATA> 0 to 2000.000 0 to 2000.000 UI
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?

Response	<numeric> = <NR2 NUMERIC RESPONSE DATA> 0 to xx.xxx 0 to xx.xxx ps p-p
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

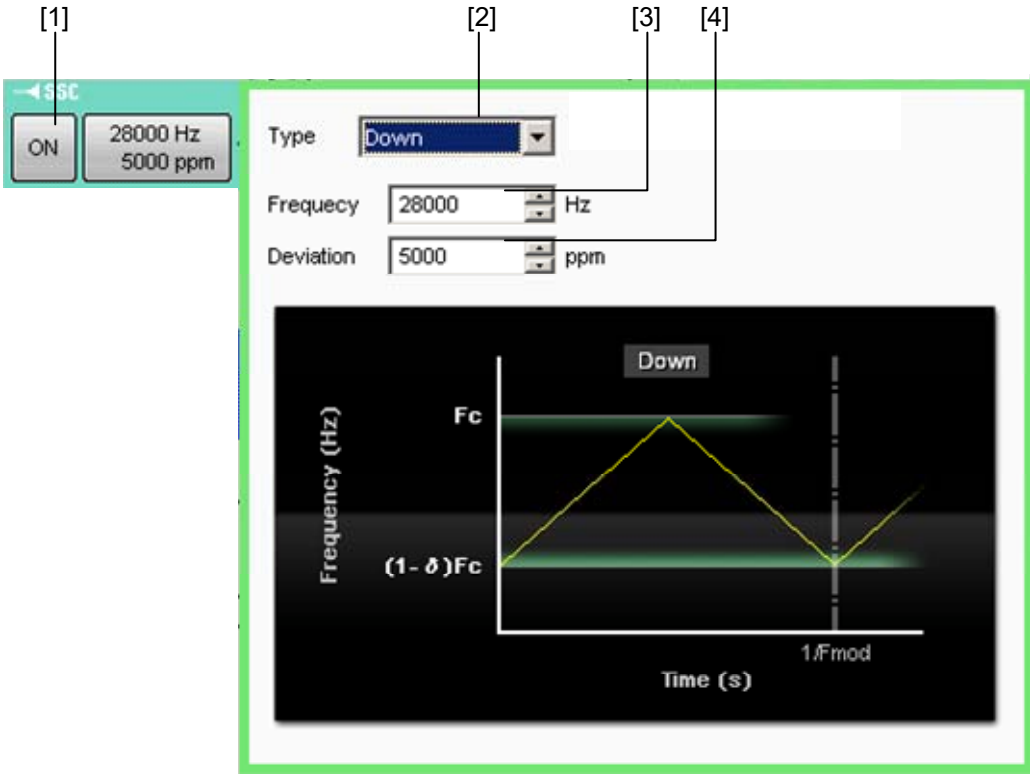


Figure 7.10.3.2-1 SSC Settings

Table 7.10.3.2-1 SSC Setting Commands

No.	Setting item	Command
[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?

: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.

: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

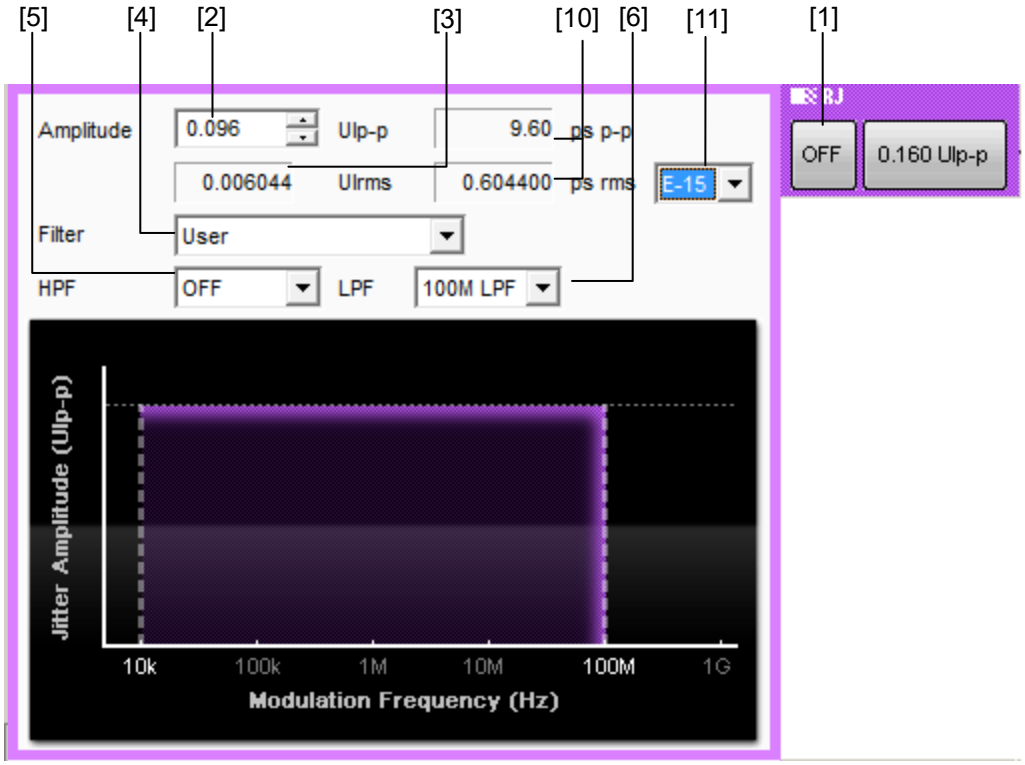


Figure 7.10.3.3-1 RJ Settings

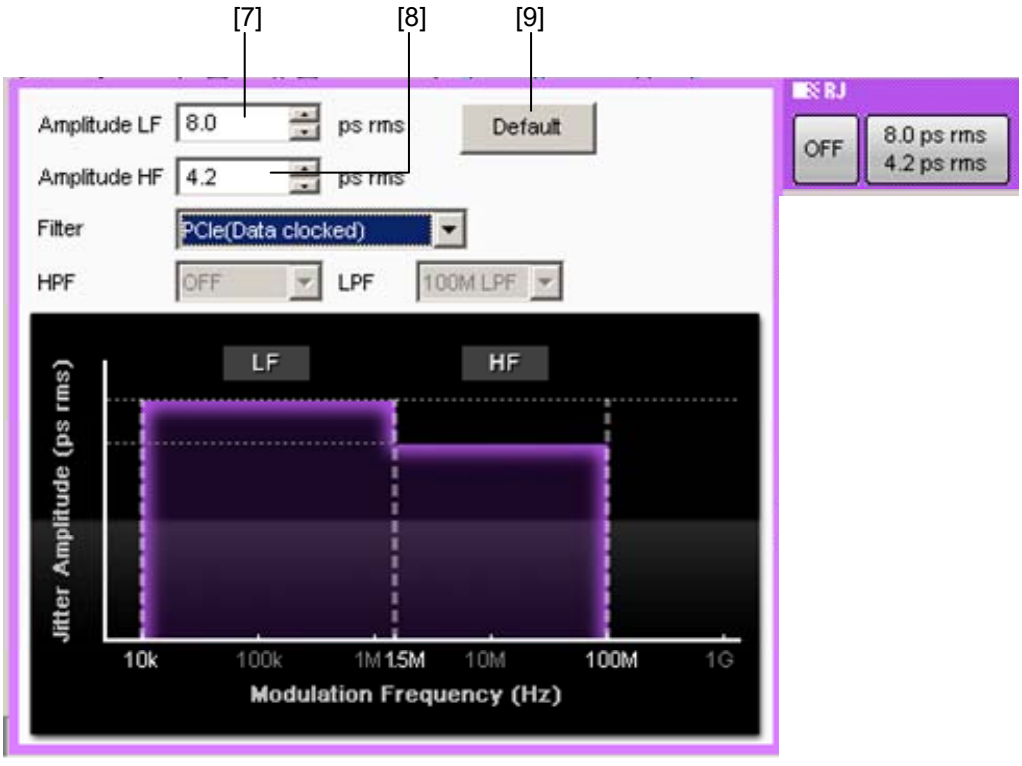


Figure 7.10.3.3-2 RJ Settings (2)

Table 7.10.3.3-1 RJ Setting Commands

No.	Setting item	Command
[1]	RJ ON/OFF	:SOURce:JITTer:RJ:ENABle
		:SOURce:JITTer:RJ:ENABle?
[2]	RJ Jitter Amplitude (UIp-p)	:SOURce:JITTer:RJ:AMPLitude
		:SOURce:JITTer:RJ:AMPLitude?
[3]	RJ Jitter Amplitude (UIrms)	:SOURce:JITTer:RJ:MONitor?
[4]	RJ Filter	:SOURce:JITTer:RJ:FILTer
		:SOURce:JITTer:RJ:FILTer?
[5]	RJ High Pass Filter	:SOURce:JITTer:RJ:HPFilter
		:SOURce:JITTer:RJ:HPFilter?
[6]	RJ Low Pass Filter	:SOURce:JITTer:RJ:LPFilter
		:SOURce:JITTer:RJ:LPFilter?
[7]	RJ Jitter Amplitude LF (ps rms)	:SOURce:JITTer:RJ:LFAMplitude
		:SOURce:JITTer:RJ:LFAMplitude?
[8]	RJ Jitter Amplitude HF (ps rms)	:SOURce:JITTer:RJ:HFAMplitude
		:SOURce:JITTer:RJ:HFAMplitude?
[9]	Amplitude LF/HF Default	:SOURce:JITTer:RJ:DEFault
[10]	RJ Jitter Amplitude Query	:SOURce:JITTer:RJ:PSAMplitude?
[11]	RJ p-p/rms Conversion Coefficient	:SOURce:JITTer:RJ:RMSConvert
		:SOURce:JITTer:RJ:RMSConvert?

: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 <numeric> = <DECIMAL NUMERIC PROGRAM DATA>

Data Generator / Clock Frequency	Half-rate(MUX)		Full-rate(PPG), Full-rate(MUX)		Quarter-rate(MUX)	
	Setting Range (UIp-p)	Step (UI)	Setting Range (UIp-p)	Step (UI)	Setting Range (UIp-p)	Step (UI)
< 2.5 GHz	0 to 0.2f	0.004	0 to 0.2f	0.002	0 to 0.2f	0.008
≥ 2.5 GHz	0 to 0.500	0.004	0 to 0.500	0.002	0 to 0.496	0.008

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.

:SOURce:JITTer:RJ:FILTer <type>

Parameter	<type> = <CHARACTER PROGRAM DATA>	
	USER	User settings
	PCIexpress1	PCI-express Data clocked settings
	PCIexpress2	PCI-express Common Ref.clock settings
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?

Response	<type> = <CHARACTER RESPONSE DATA>
	USER, PCI1, PCI2
Function	Queries RJ filter standard.
Example	`> :SOURce:JITTer:RJ:FILTer?`
	`< USER`
Compatibility	Incompatible with existing models.

:SOURce:JITTer:RJ:HPFilter <setting>

Parameter	<setting> = <CHARACTER PROGRAM DATA>	
	OFF	OFF
	M_10	10 MHz High-pass filter
	M_20	20 MHz High-pass filter
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?

Response	<setting> = <CHARACTER RESPONSE DATA>
	OFF, M_10, M_20
Function	Queries high-pass filter for use at RJ.
Example	`> :SOURce:JITTer:RJ:HPFilter?`
	`< M_10`
Compatibility	Incompatible with existing models.

: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.	

: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.

:SOURce:JITTer:RJ:RMSConvert <numeric>

Parameter	<numeric> = <DECIMAL NUMERIC PROGRAM DATA> 10 to 16 E–10 to E–16 BER of p-p/rms Conversion Coefficient
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?

Response	<numeric> = <NR1 NUMERIC RESPONSE DATA> 10 to 16 E–10 to E–16 BER of p-p/rms Conversion Coefficient
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

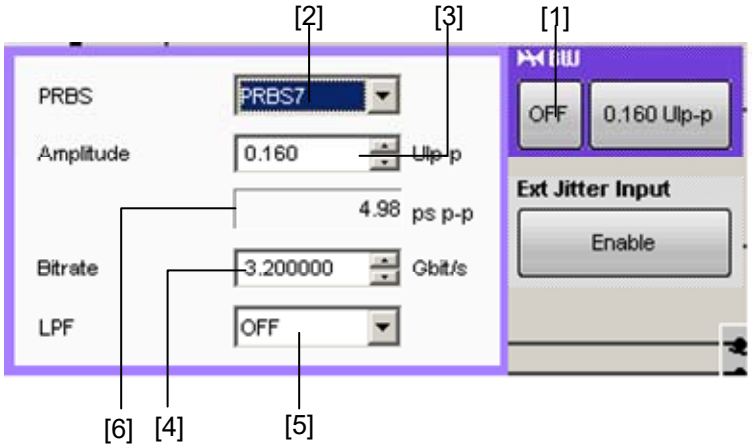


Figure 7.10.3.4-1 BUJ Settings

Table 7.10.3.4-1 BUJ Setting Commands

No.	Setting item	Command
[1]	BUJ ON/OFF	:SOURce:JITTer:BUJ:ENABle
		:SOURce:JITTer:BUJ:ENABle?
[2]	BUJ PRBS	:SOURce:JITTer:BUJ:PRBS
		:SOURce:JITTer:BUJ:PRBS?
[3]	BUJ Jitter Amplitude	:SOURce:JITTer:BUJ:AMPLitude
		:SOURce:JITTer:BUJ:AMPLitude?
[4]	BUJ Bitrate	:SOURce:JITTer:BUJ:BITRate:
		:SOURce:JITTer:BUJ:BITRate?
[5]	BUJ Low Pass Filter	:SOURce:JITTer:BUJ:LPFilter
		:SOURce:JITTer:BUJ:LPFilter?
[6]	BUJ Jitter Amplitude Query	:SOURce:JITTer:BUJ:PSAMplitude?

: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^n-1 (n = 7)
	9	PRBS2^n-1 (n = 9)
	11	PRBS2^n-1 (n = 11)
	15	PRBS2^n-1 (n = 15)
	23	PRBS2^n-1 (n = 23)
	31	PRBS2^n-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.

:SOURce:JITTer:BUJ:AMPLitude <numeric>

Parameter <numeric> = <DECIMAL NUMERIC PROGRAM DATA>

Data Generator / Clock Frequency	Half-rate(MUX)		Full-rate(PPG), Full-rate(MUX)		Quarter-rate(MUX)	
	Setting Range (UIp-p)	Step (UI)	Setting Range (UIp-p)	Step (UI)	Setting Range (UIp-p)	Step (UI)
< 2.5 GHz	0 to 0.2f	0.004	0 to 0.2f	0.002	0 to 0.2f	0.008
≥ 2.5 GHz	0 to 0.500	0.004	0 to 0.500	0.002	0 to 0.496	0.008

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:

```
> :SOURce:JITTer:BUJ:AMPLitude 0.5
```

Compatibility Incompatible with existing models.

:SOURce:JITTer:BUJ:AMPLitude?

Response <numeric> = <NR2 NUMERIC RESPONSE DATA>

0 to 0.500 0 to 0.500 UIp-p

Function Queries BUJ modulation amount.

Example

```
> :SOURce:JITTer:BUJ:AMPLitude?
< 0.500
```

Compatibility Incompatible with existing models.

:SOURce:JITTer:BUJ:BITRate <numeric>

Parameter <numeric> = <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:

```
> :SOURce:JITTer:BUJ:BITRate 1.0
```

Compatibility Incompatible with existing models.

: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	<numeric> = <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

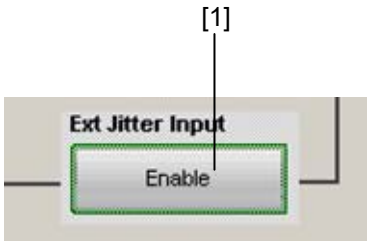


Figure 7.10.3.5-1 Ext. Jitter Input Setting

Table 7.10.3.5-1 Ext. Jitter Setting Command

No.	Setting item	Command
[1]	Ext. Jitter Input ON/OFF	:SOURce:JITTer:EXTJitter:ENABle
		:SOURce:JITTer:EXTJitter:ENABle?

: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

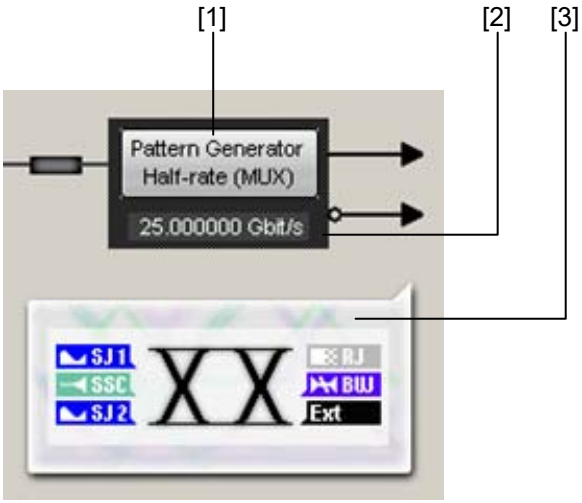


Figure 7.10.4-1 Output Settings

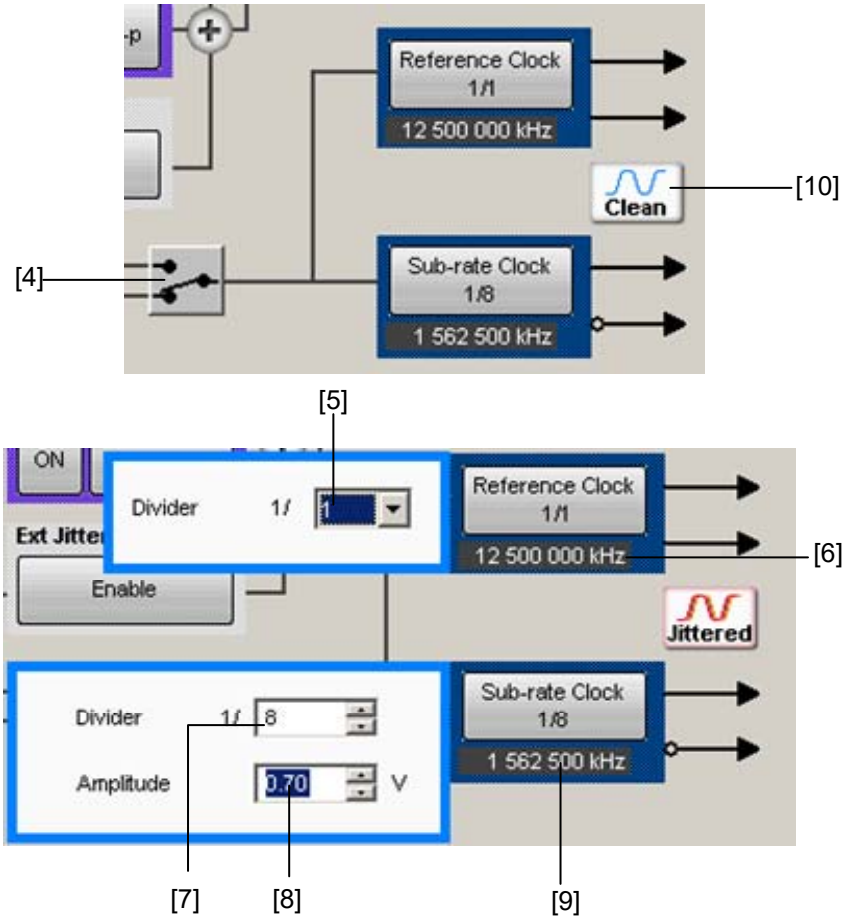


Figure 7.10.4-2 Auxiliary Output Settings

Table 7.10.4-1 Auxiliary Input/Output Setting Commands

No.	Setting item	Command
[1]	Data Generator Select	:SOURce:OUTPut:DATA:SELect
		:SOURce:OUTPut:DATA:SELect?
[2]	Bit-rate Monitor	:SOURce:OUTPut:DATA:MONitor?
[3]	Jitter Overload	:SOURce:OUTPut:DATA:JOVerload?
[4]	AUX Selector	:OUTPut:AUX:SELect
		:OUTPut:AUX:SELect?
[5]	Reference Clock	:OUTPut:AUX:REFClock
		:OUTPut:AUX:REFClock?
[6]	Reference Clock Monitor	:OUTPut:AUX:REFClock:MONitor?
[7]	Sub-rate Clock	:OUTPut:AUX:SUBRateclock
		:OUTPut:AUX:SUBRateclock?
[8]	Sub-rate Clock Amplitude	:OUTPut:AUX:SUBRateclock:AMPLitude
		:OUTPut:AUX:SUBRateclock:AMPLitude?
[9]	Sub-rate Clock Monitor	:OUTPut:AUX:SUBRateclock:MONitor?
[10]	Jitter condition	:OUTPut:AUX:JCONdition?

: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.

:SOURce:OUTPut:DATA:MONitor?

Response <string> = <STRING RESPONSE DATA>

Format	Description
"XX.XXXXX"	0.050000 to 66.000000 0.050000 to 66.000000 Gbit/s
"-----"	When no data corresponds to the query (Out of range)

Function Queries BitRate of data output.

Example
```
> :SOURce:OUTPut:DATA:MONitor?
< "12.5000000"
```

Compatibility Incompatible with existing models.

:SOURce:OUTPut:DATA:JOVerload?

Response <numeric> = <NR1 NUMERIC RESPONSE DATA>

0 Not overloaded

1 Overloaded

Function Queries whether jitter impressed on data output is Overload condition.

Example
```
> :SOURce:OUTPut:DATA:JOVerload?
< 1
```

Compatibility Incompatible with existing models.

:OUTPut:AUX:SELect <select>

Parameter <select> = <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:
```
> :OUTPut:AUX:SELect INTernal
```

Compatibility Incompatible with existing models.

:OUTPut:AUX:SELect?

Response <numeric> = <CHARACTER RESPONSE DATA>

INT, EXT

Function Queries signal source used as auxiliary output.

Example
```
> :OUTPut:AUX:SELect?
< INT
```

Compatibility Incompatible with existing models.

:OUTPut:AUX:REFClock <numeric>

Parameter
<numeric> = <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
<numeric> = <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>

Format	Description
"XXXXXXX"	12500 to 16250000 12,500 to 16,250,000 kHz
"-----"	When no data corresponds to the query

Function
Queries reference clock output frequency.

Example

```
> :OUTPut:AUX:REFClock:MONitor?
< "100000"
```

Compatibility
Incompatible with existing models.

:OUTPut:AUX:SUBRateclock <numeric>

Parameter
<numeric> = <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.

: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>

Format	Description
"XXXXXXX"	195 to 4062500 195 to 4,062,500 kHz
"-----"	When no data corresponds to the query

Function Queries sub-rate clock output frequency.

Example
```
> :OUTPut:AUX:SUBRateclock:MONitor?
< "100000"
```

Compatibility Incompatible with existing models.

:OUTPut:AUX:JCONdition?

Response	<numeric> = <NR1 NUMERIC RESPONSE DATA> 0 No jitter 1 Jitter added
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 7.10.5-1 File Menu Setting Commands

No.	Setting item	Command
[1]	Open	:SYSTem:MMEMory:RECall
[2]	Save	:SYSTem:MMEMory:STORe

: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>
JMS Jitter Modulation Source Setup

Function

Loads MP181500B setting data.

Example

```
> :SYSTem:MMEMory:RECall "C:\Test\example","1:1:1
 MU181500B",JMS
```

Compatibility

Incompatible with existing models.

: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 7.11.1.1-1 Common Setting Commands

Setting Items	Commands
Number of the unit to be operated	:UENTry:ID
	:UENTry:ID?
Number of the module to be operated (slot position)	:MODule:ID
	:MODule:ID?
Number of the port to be operated	:PORT:ID
	:PORT:ID?
Number of the data interface to be operated	:INTerface:ID
	:INTerface:ID?
Query for error message	:SYSTem:ERRor?
Query for SCPI version	:SYSTem:VERSion?
Query for software status	:SYSTem:CONDition?
Query for hardware system configuration	:SYSTem:ORGanization:HARDware?
Query for system error	:SYSTem:INFormation:ERRor?
Terminator type	:SYSTem:TERMination
Query for model name of mainframe and module	:SYSTem:CONDition:UNITs?
Query for mainframe information	:SYSTem:UNIT?
Query for module information	:SYSTem:MODule?
Module screen display	:DISPlay:ACTive

: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

: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: `> :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	`> :PORT:ID?` `< 1`
Compatibility	Incompatible with existing models

: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).
Example	To set the number of data interface to be operated to 4: `> :INTerface:ID 4`
Compatibility	Incompatible with existing models

:INTerface:ID?

Response	<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

: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>

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 7.11.1.1-2 "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>

XXXXXXXXX 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>

XXXXXX/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.

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 7.11.1.1-2 Option Character Correspondence Table

Model/Name	Option Number	Option Name
MP1800A Signal Quality Analyzer	OPT014	2-Slot PPG/ED
	OPT015	4-Slot PPG/ED
	OPT016	6-Slot PPG/ED
	OPT032	32Gbit/s PPG/ED
MT1810A 4 Slot Chassis	OPT014	2-Slot PPG/ED
	OPT015	4-Slot PPG/ED
	OPT032	32Gbit/s PPG/ED
MX180000A Signal Quality Analyzer Control Software	—	—
MU181000A 12.5GHz Synthesizer	OPTx01	Jitter Modulation
MU181000B 12.5GHz 4port Synthesizer	OPTx01	Jitter Modulation
MU181020A 12.5Gbit/s PPG	OPT001	9.8 to 12.5Gbit/s
	OPT002	0.1 to 12.5Gbit/s
	OPTx10	Variable Data Output (0.05 to 0.8 Vp-p)
	OPTx11	Variable Data Output (0.25 to 2.5 Vp-p)
	OPTx12	High Performance Data Output (0.05 to 2.0 Vp-p)
	OPTx13	Variable Data Output (0.5 to 3.5 Vp-p)
	OPTx21	Differential Clock Output (0.1 to 2.0 Vp-p)
	OPTx30	Variable Data Delay
MU181040A 12.5Gbit/s ED	OPT001	9.8 to 12.5Gbit/s
	OPT002	0.1 to 12.5Gbit/s
	OPTx20	Clock Recovery
	OPTx30	Variable Clock Delay
MU181600A Optical Transceiver (XFP)	—	—
MU181601A Optical Transceiver (SFP)	—	—
MU181620A Stressed Eye Transmitter	OPTx01	1310nm Reference
	OPTx02	1550nm Reference
	OPTx03	1310nm/1550nm Reference
	OPTx11	1310nm Stressed Eye
	OPTx12	1550nm Stressed Eye
	OPTx13	1310nm/1550nm Stressed Eye
MU181640A Optical Receiver	OPTx04	Band Width 8.5GHz
MU181800A 12.5GHz Clock Distributor	—	—

Table 7.11.1.1-2 Option Character Correspondence Table (Cont'd)

Model/Name	Option Number	Option Name
MU181020B 14Gbit/s PPG	OPT002	0.1 to 14.Gbit/s
	OPTx11	Variable Data Output (0.25 to 2.5 Vp-p)
	OPTx12	High Performance Data Output (0.05 to 2.0 Vp-p)
	OPTx13	Variable Data Output (0.5 to 3.5 Vp-p)
	OPTx21	Differential Clock Output (0.1 to 2.0 Vp-p)
	OPTx30	Variable Data Delay
MU181040B 14Gbit/s ED	OPT002	0.1 to 14.Gbit/s
	OPTx20	Clock Recovery
	OPTx30	Variable Clock Delay
MU182020A 25Gbit/s 1ch MUX MU182021A 25Gbit/s 2ch MUX	OPTx01	28Gbit/s Extension
	OPTx02	Clock Input Band Switch
	OPTx10	Variable Data Output (0.25 to 1.75 Vp-p)
	OPTx11	Variable Data Output (0.5 to 2.5 Vp-p)
	OPTx12	Variable Data Output (0.5 to 2.0 Vp-p)
	OPTx13	Variable Data Output (0.5 to 3.5 Vp-p)
	OPTx21	Clock output (0.5 to 2.0 Vp-p)
	OPTx30	25Gbit/s Variable Data Delay
	OPTx31	28Gbit/s Variable Data Delay
MU182040A 25Gbit/s 1ch DEMUX MU182041A 25Gbit/s 2ch DEMUX	OPTx01	28Gbit/s Extension
	OPTx02	Clock Input Band Switch
	OPTx30	25GHz Variable Clock Delay
	OPTx31	28GHz Variable Clock Delay
MU183020A 28G/32G bit/s PPG	OPTx01	32G bit/s Extension
	OPTx12	1ch 2V Data Output
	OPTx13	1ch 3.5V Data Output
	OPTx30	1ch Data Delay
	OPTx22	2ch 2V Data Output
	OPTx23	2ch 3.5V Data Output
	OPTx31	2ch Data Delay
MU183021A 28G/32G bit/s 4ch PPG	OPTx01	32G bit/s Extension
	OPTx12	4ch 2V Data Output
	OPTx13	4ch 3.5V Data Output
	OPTx30	4ch Data Delay
MU183040A 28G/32G bit/s ED	OPTx01	32G bit/s Extension
	OPTx10	1ch ED
	OPTx20	2ch ED
MU183041A 28G/32G bit/s 4ch ED	OPTx01	32G bit/s Extension

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>

XXXXXXXXX	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>

XXXXXX/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

: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>

XXXXXXXXX Mainframe name (e.g.,: MP1800A)
See Table 7.11.1.1-2 "Option Character Correspondence Table".

Note:

NONE is output for mainframe2 to 4, if no mainframe is connected.

<module1> to <module64> = <STRING RESPONSE DATA>

XXXXXXXXX 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.

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

: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> =

<unit>,<serial>,<mver>,<hver>,<opt1>,<sbver>,<saver>,<opt2>

<unit> = <STRING RESPONSE DATA>

XXXXXXXXX Mainframe name (e.g., : MP1800A/MT1810A)
See Table 7.11.1.1-2 "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>

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 7.11.1.1-2 "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>

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>

Parameter

<numeric> = <NR1 NUMERIC PROGRAM DATA>

1 to 6 Slot

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>

XXXXXXXXX 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>

XXXXXX/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,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.

Tab ID	MU183020A-x12/x13	MU183020A-x22/x23, MU183021A
1	Output	Output
2	Pattern	Pattern
3	Error Addition	Error Addition
4	Misc 1	Precode
5	Misc 2	Misc 1
6		Misc 2

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

Setting Items	Commands
Quick Open	:SYSTem:MMEMory:QRECall
Quick Save	:SYSTem:MMEMory:QSTore
Open	:SYSTem:MMEMory:RECall
Save	:SYSTem:MMEMory:STORe
Screen Copy Execute	:SYSTem:PRINt:COPY
Printer Setup	:SYSTem:PRINt:PRINter:SET
	:SYSTem:PRINt:PRINter:SET?
Combination Setting	:COMBination:OPERation:ABILity:CHSYnc?
	:COMBination:OPERation:CHSetting
	:COMBination:OPERation:PPGChsync
	:COMBination:OPERation:USYNcout
	:COMBination:OPERation:USYNcout?
Initialize	:SYSTem:MEMory:INITialize

: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.

: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

<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, 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
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)

Function

Opens module settings and result data.

Example

To read a file for the PPG module:

```
> :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>

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, 2, 3, 4
<slot> = 1, 2, 3, 4..., 6
<port> = 1
<module> = Module model name
<data_type> = <CHARACTER PROGRAM DATA>

PPG	PPG Setup
ED	ED Setup
SYN	Synthesizer Setup
XFP	XFP Setup
SFP	SFP Setup
ERA	Error/Alarm Result
OTX	Transmitter Setup
MX1	MUX Setup
MX2	MUX Setup
DX1	DEMUX Setup
DX2	DEMUX Setup
DER	DEMUX Error/Alarm Result
P32	28G/32G PPG Setup
P32X	28G/32G 4ch PPG Setup
E32	28G/32G ED Setup
E32X	28G/32G 4ch ED Setup

<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:

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	<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.	

: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.

: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>

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

Response	<numeric> = <NR1 NUMERIC RESPONSE DATA>
	0 Unit Sync OFF
	1 Unit Sync ON
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.

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

Setting Items	Commands
Output ON/OFF	:SOURce:OUTPut:ASET
	:SOURce:OUTPut:ASET?
Error Addition ON/OFF	:SOURce:PATTern:EADDition:ASET
	:SOURce:PATTern:EADDition:ASET?
Single Error Addition	:SOURce:PATTern:EADDition:ASINgle
Open	Refer to 7.11.1.2 “File menu commands”.
Save	Refer to 7.11.1.2 “File menu commands”.
Print	Refer to 7.11.1.2 “File menu commands”.
System Error ON/OFF	:SYSTem:BEEPer:SYSTem:SET
	:SYSTem:BEEPer:SYSTem:SET?
System Error	:SYSTem:BEEPer:SYSTem:TYPE
	:SYSTem:BEEPer:SYSTem:TYPE?
Unit Sync Output	:SOURce:PATTern:USYNcout

: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.

: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.	

: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.

7.11.1.4 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	<numeric> = <DECIMAL NUMERIC PROGRAM DATA>	
	0 to 15	Sum total of the bit of transition filter (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	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: `> :INSTrument:PPGG32:PTRansition 4`	

:INSTrument:PPGG32:PTRansition?

Response	<numeric> = <NR1 NUMERIC RESPONSE DATA>	
	0 to 15	Sum total of the bit of transition filter (DECIMAL)
Function	Queries contents of transition filter (positive direction transition) at 28G/32G bit/s PPG status.	
Example	`> :INSTrument:PPGG32:PTRansition?` `< 4`	

:INSTrument:PPGG32:NTRansition <numeric>

Parameter	<numeric> = <DECIMAL NUMERIC PROGRAM DATA>	
	0 to 15	Sum total of the bit of transition filter (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	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: `> :INSTrument:PPGG32:NTRansition 4`	

:INSTrument:PPGG32:NTRansition?

Response	<numeric> = <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

Figure 7.11.1.5-1 Data Interface Select

Table 7.11.1.5-1 Data Interface Select Setting Commands

Setting Item	Command
Data Interface Select	:INTerface:ID
	:INTerface:ID?

: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
```

:INTerfaceme: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
```

7.11.1.6 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 7.11.1.6-1 Pattern Editor Setting Commands

Setting Item	Command
Open	:SYSTem:MMEMory:PATTern:RECall
Save	:SYSTem:MMEMory:PATTern:STORe

: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>

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:MMEMory:PATTern:STORe "C:\Test\example",TXT
```

Compatibility

Incompatible with existing models.

7.11.2 Output Tab

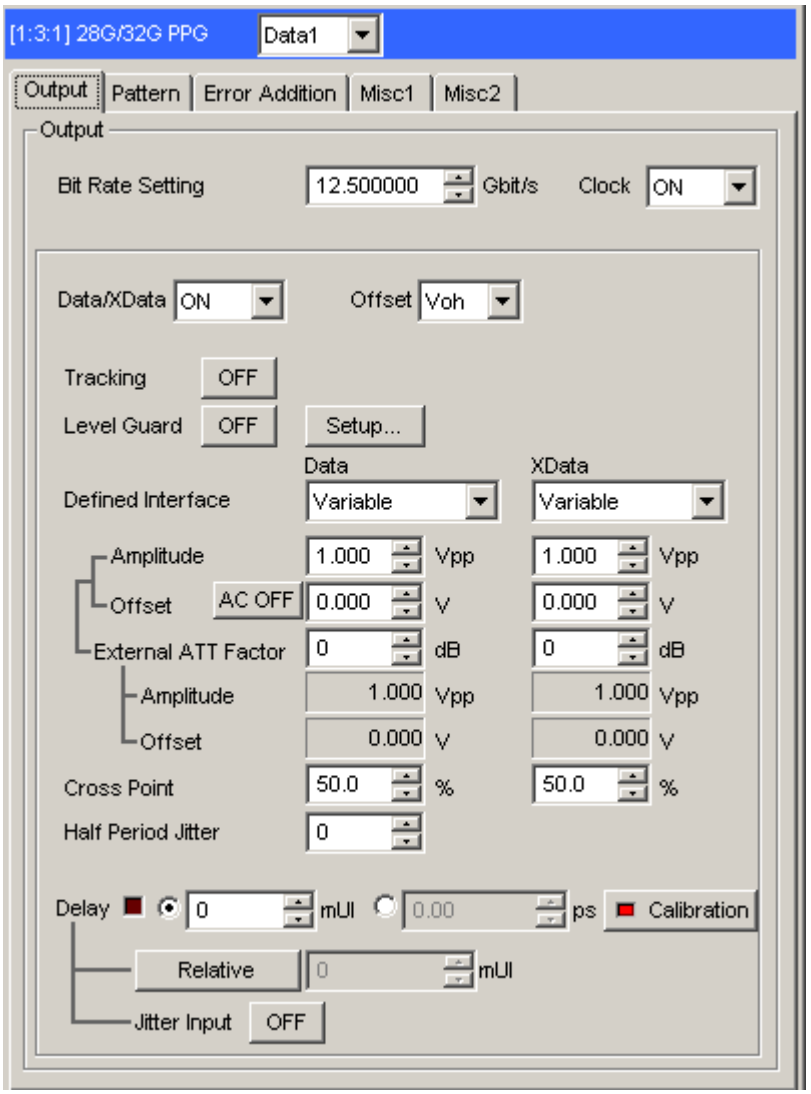


Figure 7.11.2-1 Output Tab

7.11.2.1 Output Setting Commands

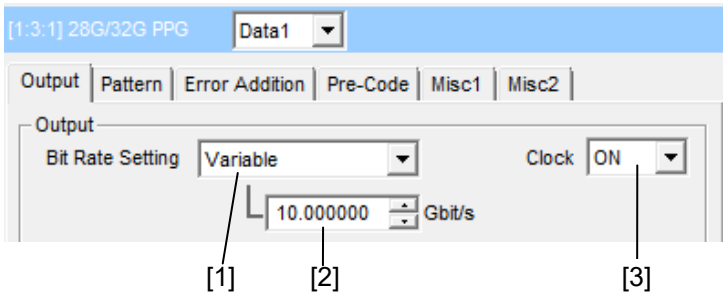


Figure 7.11.2.1-1 Output Settings

Table 7.11.2.1-1 Output Setting Commands

No.	Setting Items	Commands
[1]	Bit Rate Setting Preset Standard Bit Rate	:OUTPut:DATA:STANdard
		:OUTPut:DATA:STANdard?
[2]	Bit Rate Setting/ Bit Rate Monitor	:OUTPut:DATA:BITRate
		:OUTPut:DATA:BITRate?
[3]	Clock	:OUTPut:CLOCk:OUTPut
		:OUTPut:CLOCk:OUTPut?

:OUTPut:DATA:STANdard <freq>

Parameter

<freq> = <STRING PROGRAM DATA>

"Variable"	Variable	Variable*
"PCI_EX1"	PCI Express Gen1	2.500000 Gbit/s
"PCI_EX2"	PCI Express Gen2	5.000000 Gbit/s
"USB3"	USB3.0	5.000000 Gbit/s
"PCI_EX3"	PCI Express Gen3	8.000000 Gbit/s
"8G_FC"	8GFC	8.500000 Gbit/s
"OC_192"	OC192/STM64	9.953280 Gbit/s
"IB_QDR"	Infiniband QDR	10.000000 Gbit/s
"USB3_1"	USB3.1	10.000000 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
"16G_FC"	16GFC	14.025000 Gbit/s
"IB_FDR"	Infiniband_FDR	14.062500 Gbit/s
"PCI_EX4"	PCI Express Gen4	16.000000 Gbit/s
"SAS"	SAS	24.000000 Gbit/s
"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

*: 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_":

```
> :OUTPut:DATA:STANdard "USB3_1"
```

Compatibility

Incompatible with the existing modules.

:OUTPut:DATA:STANdard?

Response	<freq> = <STRING RESPONSE DATA> "Variable","PCI_EX1", "PCI_EX2", "USB3","PCI_EX3", "8G_FC", "OC_192","IB_QDR", "USB3_1", "10G_FC ","OC_192_G975", "OC_192_G709", "10GbE_G709", "10G_FC_G709", "16G_FC", "IB_FDR", "PCI_EX4", "SAS","IB_EDR", "100GbE ", "100G_OTU", "32G_FC","100G_ULH"
Function	Queries the preset bit rate that the 32G PPG module outputs.
Example	`> :OUTPut:DATA:STANdard?` `< "USB3_1"`
Compatibility	Incompatible with the existing modules.

: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.

: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	`> :OUTPut:DATA:BITRate?` `< 32.100000`
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

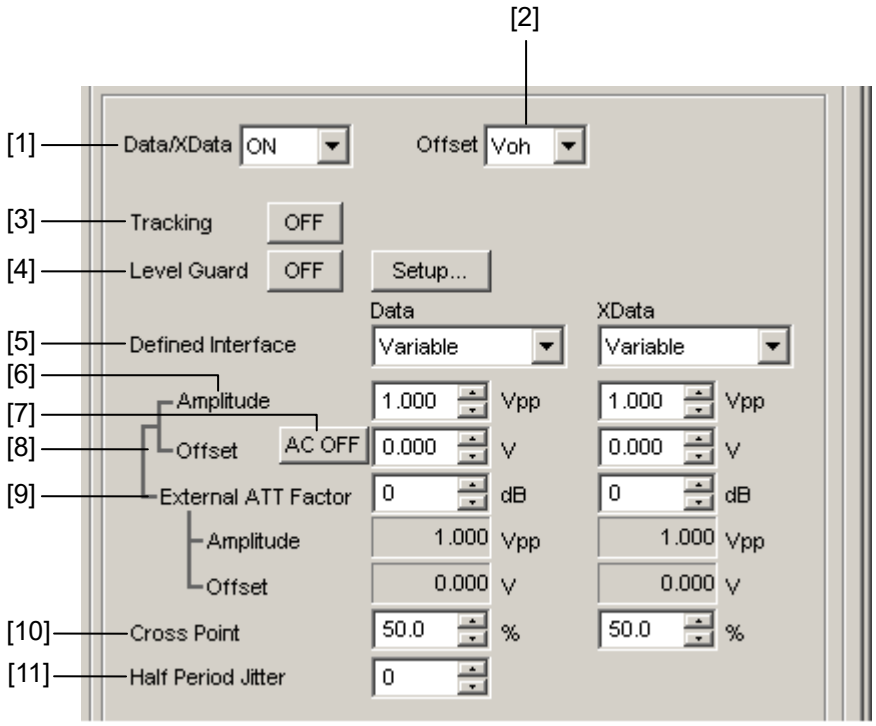


Figure 7.11.2.2-1 Data/XData Settings

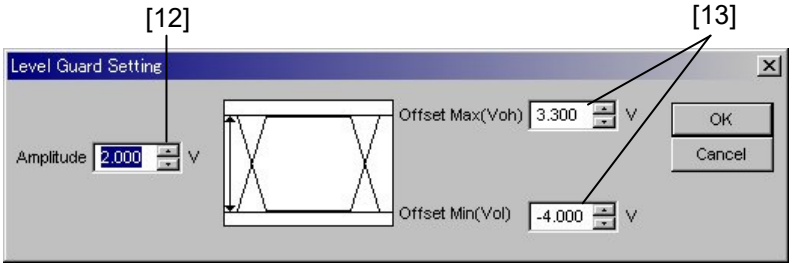


Figure 7.11.2.2-2 Level Guard Settings Dialog Box

Table 7.11.2.2-1 Data/XData Setting Commands

No.	Setting Items	Commands
[1]	Data/XData	:OUTPut:DATA:OUTPut
		:OUTPut:DATA:OUTPut?
[2]	Offset	:OUTPut:OFFset
		:OUTPut:OFFset?
[3]	Tracking	:OUTPut:DATA:TRACking
		:OUTPut:DATA:TRACking?
[4]	Level Guard	:OUTPut:DATA:LEVGuard
		:OUTPut:DATA:LEVGuard?
[5]	Defined Interface	:OUTPut:DATA:LEVel
		:OUTPut:DATA:LEVel?

Table 7.11.2.2-1 Data/XData Setting Commands (Cont'd)

No.	Setting Items	Commands
[6]	Amplitude	:OUTPut:DATA:AMPLitude
		:OUTPut:DATA:AMPLitude?
[7]	AC ON/OFF	:OUTPut:DATA:AOFFset
		:OUTPut:DATA:AOFFset?
[8]	Offset	:OUTPut:DATA:OFFSet
		:OUTPut:DATA:OFFSet?
[9]	External ATT Factor	:OUTPut:DATA:ATTFactor
		:OUTPut:DATA:ATTFactor?
[10]	Cross Point	:OUTPut:DATA:CPOint
		:OUTPut:DATA:CPOint?
[11]	Half Period Jitter	:OUTPut:DATA:HPJitter
		:OUTPut:DATA:HPJitter?
[12]	Amplitude	:OUTPut:DATA:LIMitter:AMPLitude
		:OUTPut:DATA:LIMitter:AMPLitude?
[13]	Offset Max/Min	:OUTPut:DATA:LIMitter:OFFSet
		:OUTPut:DATA:LIMitter:OFFSet?
[14]	Without label (Queries the status of data output setting completion)	:OUTPut:CHANge:CSTate?

: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	<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	Compatible with MU181020A/B.	

: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.	

:OUTPut:DATA:LEVel? <port>

Parameter	<port> = <CHARACTER PROGRAM DATA> DATA, XDATa
Response	<level> = <CHARACTER RESPONSE DATA> 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	<port> = <CHARACTER PROGRAM DATA>
	DATA — DATA output
	XDATa — XData output
	<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 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	<port> = <CHARACTER PROGRAM DATA> DATA, XDATa
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 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.

: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.

: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 –20 to 20
Function	Queries the half period jitter.
Example	`> :OUTPut:DATA:HPJitter?`
	`< 10`
Compatibility	Incompatible with existing models.

: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-x12/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:LIMitter:OFFSet 3.000,-2.000`	
Compatibility	Compatible with MU181020A/B.	

: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

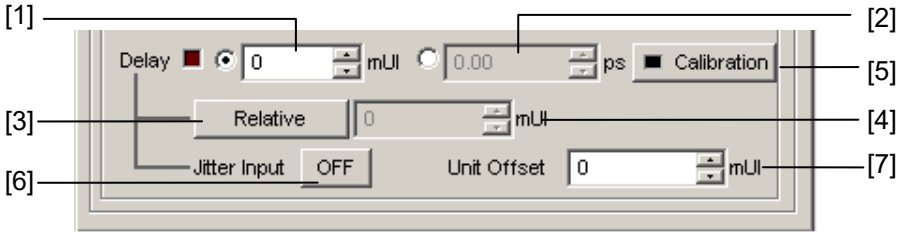


Figure 7.11.2.3-1 Delay Setting

Table 7.11.2.3-1 Delay Setting Commands

No.	Setting Items	Commands
[1]	Delay (mUI setting)	:OUTPut:DATA:UIPadjust
		:OUTPut:DATA:UIPadjust?
[2]	Delay (ps setting)	:OUTPut:DATA:PADJust
		:OUTPut:DATA:PADJust?
[3]	Relative	:OUTPut:DATA:RELative
		:OUTPut:DATA:RELative?
[4]	No label (Relative value setting)	:OUTPut:DATA:RDELay
		:OUTPut:DATA:RDELay?
[5]	Calibration	:OUTPut:DATA:PCALibration
[6]	Jitter Input	:OUTPut:DATA:JINPut
		:OUTPut:DATA:JINPut?
[7]	Unit Offset	:OUTPut:DATA:UDELay:OFFSet
		:OUTPut:DATA:UDELay:OFFSet?

:OUTPut:DATA:UIPadjust <numeric>

Parameter
<numeric> = <DECIMAL NUMERIC PROGRAM DATA>
–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 –1000 to 1000 mUI/2 mUI Step
	–64000 to 64000 –64000 to 64000 mUI/2 mUI Step (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	Compatible with MU181020A/B.

:OUTPut:DATA:PADJust <numeric>

Parameter	<numeric> = <DECIMAL NUMERIC PROGRAM DATA>
	–416 to 416 –416 to 416 ps/0.2 ps (at 2.4 Gbit/s)
	–80 to 80 –80 to 80 ps/0.08 ps (at 12.5 Gbit/s)
	–31.14 to 31.14 –31.14 to 31.14 ps/0.06 ps (at 32.1 Gbit/s)
	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	Compatible with MU181020A/B.

:OUTPut:DATA:PADJust?

Response	<numeric> = <NR2 NUMERIC RESPONSE DATA>
	–416 to 416 –416 to 416 ps/0.2 ps (at 2.4 Gbit/s)
	–80 to 80 –80 to 80 ps/0.08 ps (at 12.5 Gbit/s)
	–31.14 to 31.14 –31.14 to 31.14 ps/0.06 ps (at 32.1 Gbit/s)
	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	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.	

: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.

7.11.3 Pattern Tab

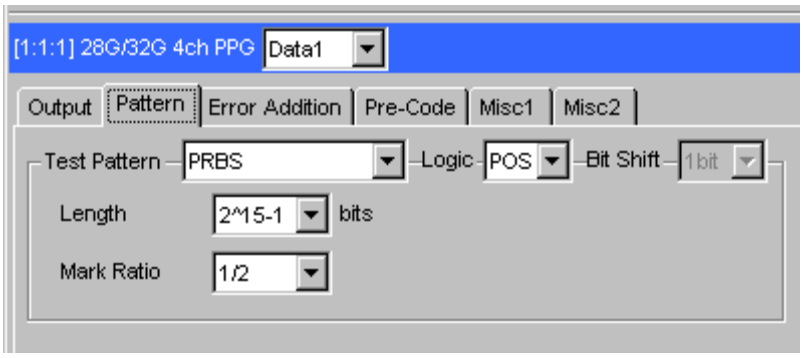


Figure 7.11.3-1 Pattern Tab

7.11.3.1 Test Pattern Setting Commands

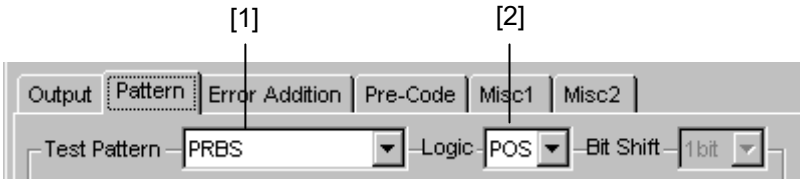


Figure 7.11.3.1-1 Test Pattern Setting

Table 7.11.3.1-1 Test Pattern Setting Commands

No.	Setting Items	Commands
[1]	Test Pattern	:SOURce:PATTern:TYPE
		:SOURce:PATTern:TYPE?
[2]	Logic	:SOURce:PATTern:LOGic
		:SOURce:PATTern:LOGic?

: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`	
Compatibility	Compatible with MU181020A/B.	

:SOURce:PATTern:TYPE?

Response	<type> = <CHARACTER RESPONSE DATA> PRBS, ZSUB, DATA, MIXD
Function	Queries the type of the test pattern.
Example	> :SOURce:PATTern:TYPE? < MIXD
Compatibility	Compatible with MU181020A/B.

:SOURce: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 negative logic (NEGative): > :SOURce:PATTern:LOGic NEGative
Compatibility	Compatible with MU181020A/B.

:SOURce:PATTern:LOGic?

Response	<logic> = <CHARACTER RESPONSE DATA> POS, NEG
Function	Queries the logic of the test pattern.
Example	> :SOURce:PATTern:LOGic? < NEG
Compatibility	Compatible with MU181020A/B.

7.11.3.2 PRBS pattern setting commands

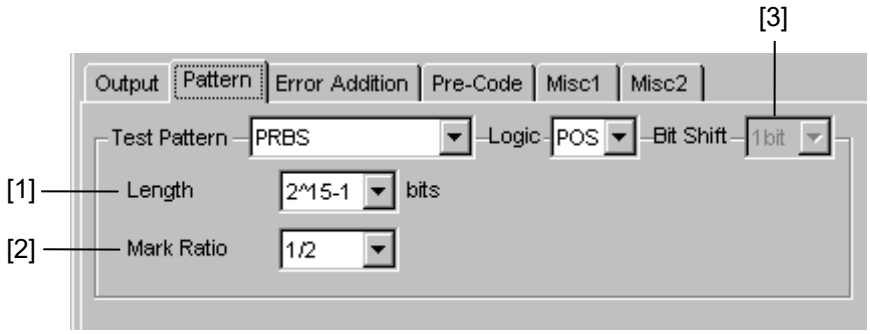


Figure 7.11.3.2-1 Test Pattern Setting (When PRBS Is Selected)

Table 7.11.3.2-1 PRBS Pattern Setting Commands

No.	Setting Items	Commands
[1]	Length	:SOURce:PATTern:PRBS:LENGth
		:SOURce:PATTern:PRBS:LENGth?
[2]	Mark Ratio	:SOURce:PATTern:PRBS:MRATio
		:SOURce:PATTern:PRBS:MRATio?
[3]	Bit Shift	:SOURce:PATTern:PRBS:BSHift?

:SOURce:PATTern: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 (2^n-1 (n = 7, 9, 10, 11, 15, 20, 23, or 31)) during PRBS pattern generation.
Example	To set PRBS15: `> :SOURce:PATTern:PRBS:LENGth 15`
Compatibility	Compatible with MU181020A/B.

: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

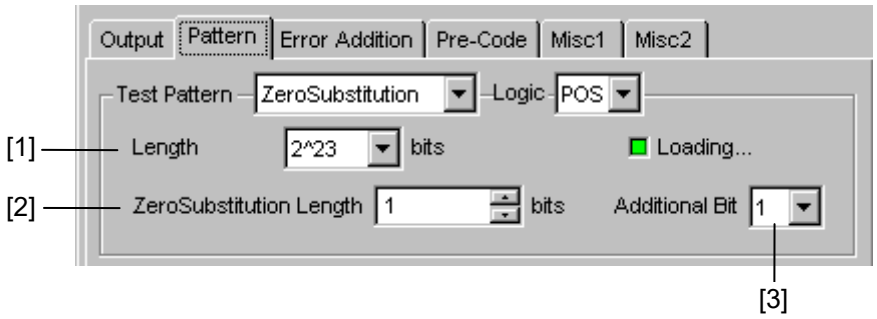


Figure 7.11.3.3-1 Test Pattern Setting (When Zero-Substitution Is Selected)

Table 7.11.3.3-1 Zero-Substitution Pattern Setting Commands

No.	Setting Items	Commands
[1]	Length	:SOURce:PATTern:ZSUBstitute:LENGth
		:SOURce:PATTern:ZSUBstitute:LENGth?
[2]	Zero-Substitution Length	:SOURce:PATTern:ZSUBstitute:ZLENgth
		:SOURce:PATTern:ZSUBstitute:ZLENgth?
[3]	Additional Bit	:SOURce:PATTern:ZSUBstitute:ADDBit
		:SOURce:PATTern:ZSUBstitute:ADDBit?

: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.

:SOURce: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 generation.
Example	`> :SOURce:PATTern:ZSUBstitute:LENGth?` `< 15`
Compatibility	Compatible with MU181020A/B.

:SOURce: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) 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	<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 generation.
Example	`> :SOURce:PATTern:ZSUBstitute:ZLENgth?` `< 10`
Compatibility	Compatible with MU181020A/B.

:SOURce:PATTern:ZSUBstitute:ADDBit <numeric>

Parameter	<numeric> = <DECIMAL NUMERIC PROGRAM DATA> 0 The 2^nth bit is set to "0". 1 The 2^nth bit is set to "1". (same as existing models).
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:PATTern:ZSUBstitute:ADDBit?

Response	<numeric> = <NR1 NUMERIC RESPONSE DATA> 0 The 2^nth bit is set to "0". 1 The 2^nth bit is set to "1". (same as existing models).
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

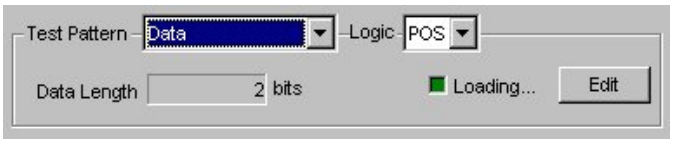


Figure 7.11.3.4-1 Test Pattern Setting (When Data Is Selected)

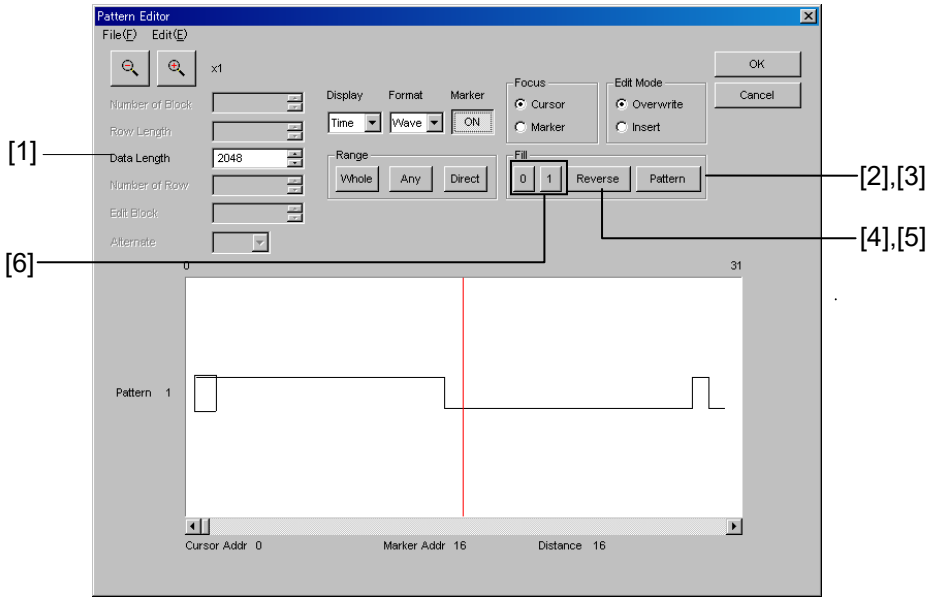


Figure 7.11.3.4-2 Data Pattern Setting (Pattern Editor)

Table 7.11.3.4-1 Data Pattern Setting Commands

No.	Setting Items	Commands
[1]	Data Length	:SOURce:PATTern:DATA:LENGth
		:SOURce:PATTern:DATA:LENGth?
[2]	No label (Pattern setting)	:SOURce:PATTern:DATA:WHOLe
		:SOURce:PATTern:DATA:WHOLe?
[3]	No label (Pattern setting)	:SOURce:PATTern:BDATa:WHOLe
		:SOURce:PATTern:BDATa:WHOLe?
[4]	No label (Data inversion)	:SOURce:PATTern:DREVerse:ADDRess
[5]	No label (Data inversion)	SOURce:PATTern:DREVerse:DELTa
[6]	No label (All 0 or All 1)	:SOURce:PATTern:DATA:FILL

:SOURce:PATTern:DATA:LENGth <numeric>

Parameter	<numeric> = <DECIMAL NUMERIC PROGRAM DATA>		
	Independent	2 to 268435456	2 to 268,435,456 bits/1 bit Step
	2 ch Combination	4 to 536870912	4 to 536,870,912 bits/2 bit Step
	4 ch Combination	8 to 1073741824	8 to 1,073,741,82 bits/4 bit Step
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>

Parameter

<start> = <NON-DECIMAL PROGRAM DATA>

#H0 #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 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>

Parameter

<start> = <NON-DECIMAL PROGRAM DATA>

#H0 to #HFFFFFFFF	0 to FFFFFFF 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 × 4 bits)

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:

```
> :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 #HFFFFFFFF — 0 to FFFFFFF bits/1 bit Step (Specify in hexadecimal)

<end> = <NON-DECIMAL PROGRAM DATA>

#H0 to #HFFFFFFFF — 0 to FFFFFFF 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.)

:SOURce:PATTern:BDATa:WHOLe? <start>[,<size>]

Parameter

<start> = <NON-DECIMAL PROGRAM DATA>

#H0 to #HFFFFFFFF	0 to FFFFFFF 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.

[<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.

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 binary data of the setting pattern from the <start> address of the pattern data during Data pattern generation:

```
> :SOURce:PATTern:BDATa:WHOLe? #H0
< #213AA...
```

To query binary data of 2 bytes from the <start> address of the pattern data during Data pattern generation:

```
> :SOURce:PATTern:BDATa:WHOLe? #H0,2
< #12AA
```

Compatibility

Partially compatible with MU181020A/B.

:SOURce:PATTern:DREVerse:ADDRess <start>,<end>

Parameter

<start> = <NON-DECIMAL PROGRAM DATA>

#H0 to #HFFFFFFFF	0 to FFFFFFF bits/1 bit Step (Specify in hexadecimal)

<end> = <NON-DECIMAL PROGRAM DATA>

#H0 to #HFFFFFFFF	0 to FFFFFFF 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.

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

<start> = <NON-DECIMAL PROGRAM DATA>

#H0 to #HFFFFFFFF	0 to FFFFFFF 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 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).

: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

$$\text{Max} = \frac{\text{Pattern Length}}{128} \text{ (rounding up fractions)}$$

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

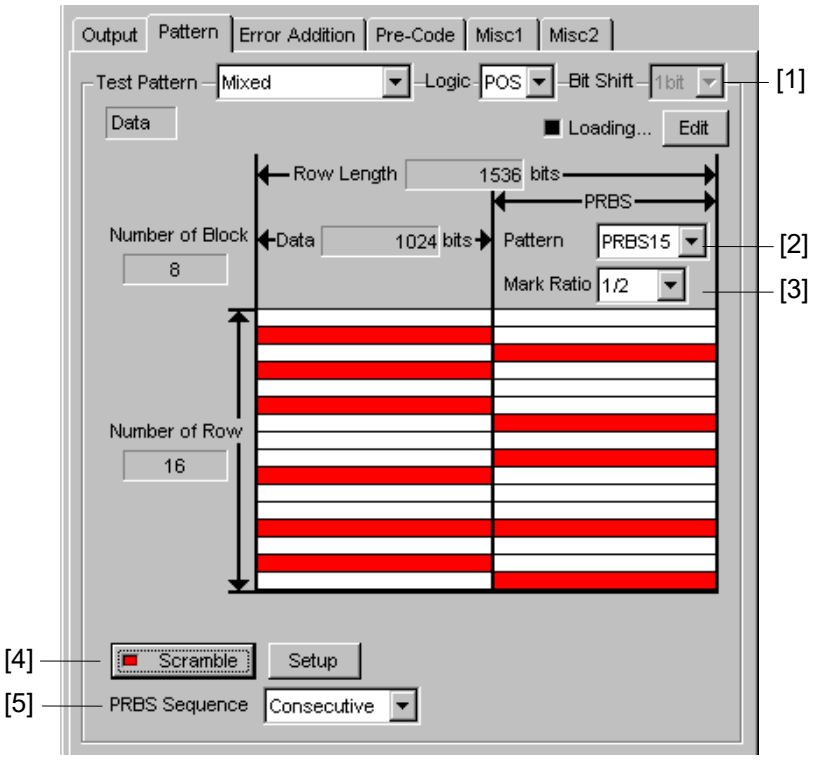


Figure 7.11.3.5-1 Test Pattern Setting (When Mixed Is Selected)

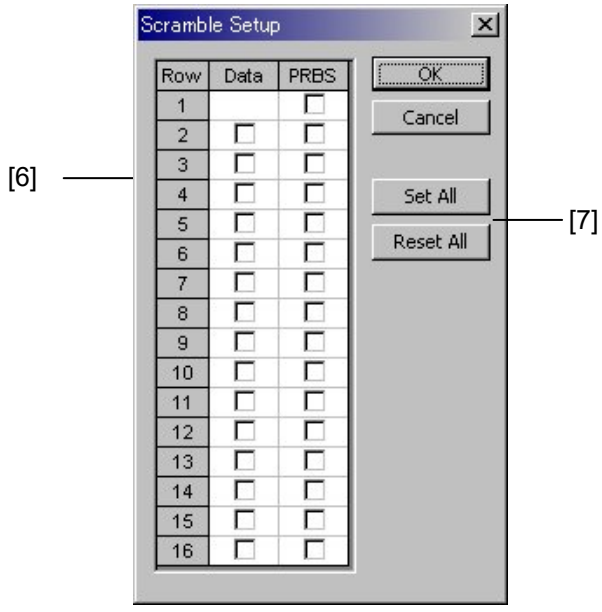


Figure 7.11.3.5-2 Scramble Setup Dialog Box

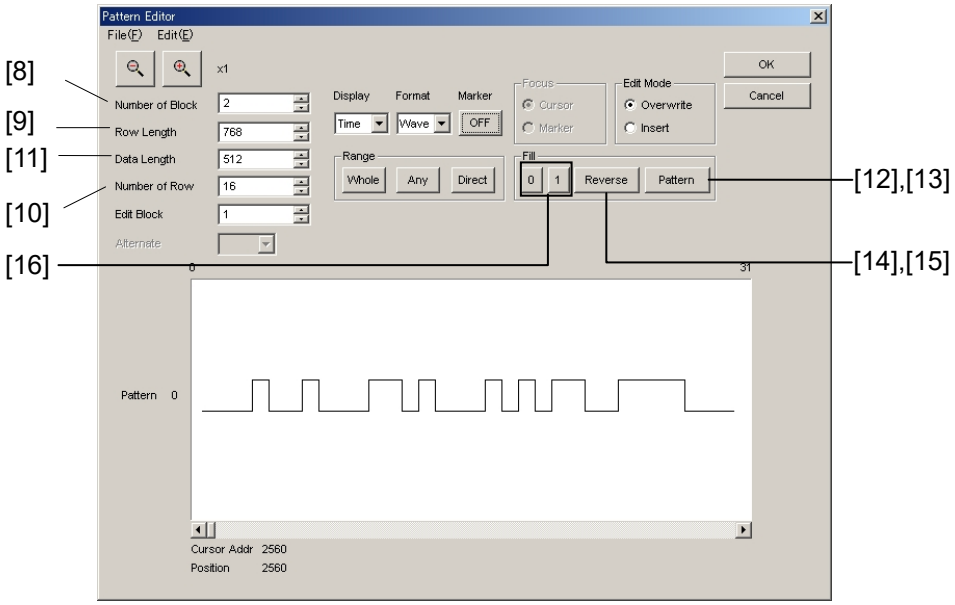


Figure 7.11.3.5-3 Mixed Data Pattern Setting (Pattern Editor)

Table 7.11.3.5-1 Mixed Data Mixed Data Pattern Setting Commands

No.	Setting Items	Commands
[1]	Bit Shift	:SOURce:PATTern:MIXData:PRBS:BITShift?
[2]	Pattern	:SOURce:PATTern:MIXData:PRBS:LENGth
		:SOURce:PATTern:MIXData:PRBS:LENGth?
[3]	Mark Ratio	:SOURce:PATTern:MIXData:PRBS:MRATio
		:SOURce:PATTern:MIXData:PRBS:MRATio?
[4]	Scramble	:SOURce:PATTern:MIXData:SCRamble
		:SOURce:PATTern:MIXData:SCRamble?
[5]	PRBS Sequence	:SOURce:PATTern:MIXData:PRBS:SEQuence
		:SOURce:PATTern:MIXData:PRBS:SEQuence?
[6]	Setup	:SOURce:PATTern:MIXData:SRSetting
		:SOURce:PATTern:MIXData:SRSetting?
[7]	Set All/Reset All	:SOURce:PATTern:MIXData:SCRamble:ALLSet
[8]	Number of Block	:SOURce:PATTern:MIXData:BLOCk
		:SOURce:PATTern:MIXData:BLOCk?
[9]	Row Length	:SOURce:PATTern:MIXData:RLENgth
		:SOURce:PATTern:MIXData:RLENgth?
[10]	Number of Row	:SOURce:PATTern:MIXData:ROW
		:SOURce:PATTern:MIXData:ROW?
[11]	Data Length	:SOURce:PATTern:MIXData:DATA:LENGth
		:SOURce:PATTern:MIXData:DATA:LENGth?
[12]	No label (Pattern setting)	:SOURce:PATTern:MIXData:DATA:WHOLe
		:SOURce:PATTern:MIXData:DATA:WHOLe?
[13]	No label (Pattern setting)	:SOURce:PATTern:MIXData:BDATa:WHOLe
		:SOURce:PATTern:MIXData:BDATa:WHOLe?

Table 7.11.3.5-1 Mixed Data Mixed Data Pattern Setting Commands (Cont'd)

No.	Setting Items	Commands
[14]	No label (Data inversion)	:SOURce:PATTern:MIXData:DREVerse:ADDRess
[15]	No label (Data inversion)	:SOURce:PATTern:MIXData:DREVerse:DELTa
[16]	No label (All 0 or All 1)	:SOURce:PATTern:MIXData:DATA:FILL

: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.

: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	<numeric> = <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.	

: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.	

: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>

Parameter
<numeric> = <DECIMAL NUMERIC PROGRAM DATA>

Independent	1536 to 2415919104	1,536 to 2,415,919,104 bits/ 256 bit Step
2 ch Combination	3072 to 4831838208	3,072 to 4,831,838,208 bits/ 512 bit Step
4 ch Combination	6144 to 9663676416	6,144 to 9,663,676,416 bits/ 1,024 bit Step

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?

Response
<numeric> = <NR1 NUMERIC RESPONSE DATA>

Independent	1536 to 2415919104	1,536 to 2,415,919,104 bits/ 256 bit Step
2 ch Combination	3072 to 4831838208	3,072 to 4,831,838,208 bits/ 512 bit Step
4 ch Combination	6144 to 9663676416	6,144 to 9,663,676,416 bits/ 1,024 bit Step

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>

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
Partially compatible with MU181020A/B.

: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	Partially compatible with MU181020A/B.

:SOURce:PATTern:MIXData:DATA:LENGth <numeric>

Parameter	<numeric> = <DECIMAL NUMERIC PROGRAM DATA>

Independent	1024 to 268435456	1,024 to 268,435,456 bits/ 1 bit Step
2 ch Combination	2048 to 536870912	2,048 to 536,870,912 bits/ 2 bit Step
4 ch Combination	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 `> :SOURce:PATTern:MIXData:DATA:LENGth 1024`
Compatibility	Partially compatible with MU181020A/B (the parameter range varies depending on the settings).

:SOURce:PATTern:MIXData:DATA:LENGth?

Response	<numeric> = <NR1 NUMERIC RESPONSE DATA>

Independent	1024 to 268435456	1,024 to 268,435,456 bits/ 1 bit Step
2 ch Combination	2048 to 536870912	2,048 to 536,870,912 bits/ 2 bit Step
4 ch Combination	4096 to 1073741824	4,096 to 1,073,741,8244 bits/ 4 bit Step

Function	Queries the pattern length on the pattern data side that is edited during Mixed Data pattern generation.
Example	`> :SOURce:PATTern:MIXData:DATA:LENGth?` `< 1024`
Compatibility	Partially compatible with MU181020A/B.

:SOURce: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 #HFFFFFFFF 0 to FFFFFFF bits/1 bit Step (Specify in hexadecimal)

<end> = <NON-DECIMAL PROGRAM DATA>

#H0 to #HFFFFFFFF 0 to FFFFFFF 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.

<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

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 1 to 511 Block/1 Step

<start> = <NON-DECIMAL PROGRAM DATA>

#H0 to #HFFFFFFFF 0 to FFFFFFF 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 × 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:

```
> :SOURce:PATTern:MIXData:DATA:WHOLe? 1,#H0
< "H1111"
```

Compatibility

Partially compatible with MU181020A/B.

:SOURce:PATTern:MIXData:BDATa:WHOLe <block>,<start>,<end>,<bdata>

Parameter

<block> = <DECIMAL NUMERIC PROGRAM DATA>

1 to 511 1 to 511 Block/1 Step

<start> = <NON-DECIMAL PROGRAM DATA>

#H0 to #HFFFFFFFF 0 to FFFFFFF bits/1 bit Step
(Specify in hexadecimal)

<end> = <NON-DECIMAL PROGRAM DATA>

#H0 to #HFFFFFFFF 0 to FFFFFFF 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 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

<block> = <DECIMAL NUMERIC PROGRAM DATA>

1 to 511 1 to 511 Block/1 Step

<start> = <NON-DECIMAL PROGRAM DATA>

#H0 to #HFFFFFFFF 0 to FFFFFFF 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.

[<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: 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 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>

Parameter

<block> = <DECIMAL NUMERIC PROGRAM DATA>

1 to 511 1 to 511 Block/1 Step

<start> = <NON-DECIMAL PROGRAM DATA>

#H0 to #HFFFFFFFF 0 to FFFFFFF bits/1 bit Step
(Specify in hexadecimal)

<end> = <NON-DECIMAL PROGRAM DATA>

#H0 to #HFFFFFFFF 0 to FFFFFFF 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.

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>

Parameter

<block> = <DECIMAL NUMERIC PROGRAM DATA>

1 to 511 1 to 511 Block/1 Step

<start> = <NON-DECIMAL PROGRAM DATA>

#H0 to #FFFFFFF 0 to FFFFFFF bits/1 bit Step
(Specify in hexadecimal)

<delta> = <DECIMAL NUMERIC PROGRAM DATA>

1 to 134217728 1 to 134,217,728 bits/1 bit Step

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).

: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

$$\text{Max} = \frac{\text{Pattern Length}}{128} \text{ (rounding up fractions)}$$

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

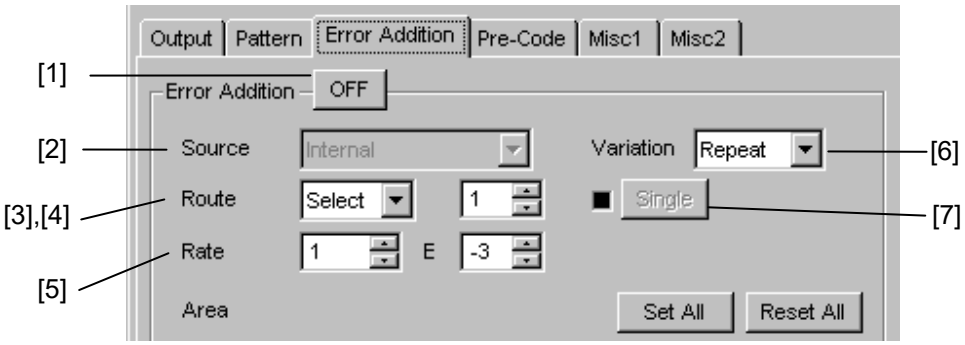


Figure 7.11.4-1 Error Addition Settings (PRBS, Zero-Substitution, and Data Patterns)

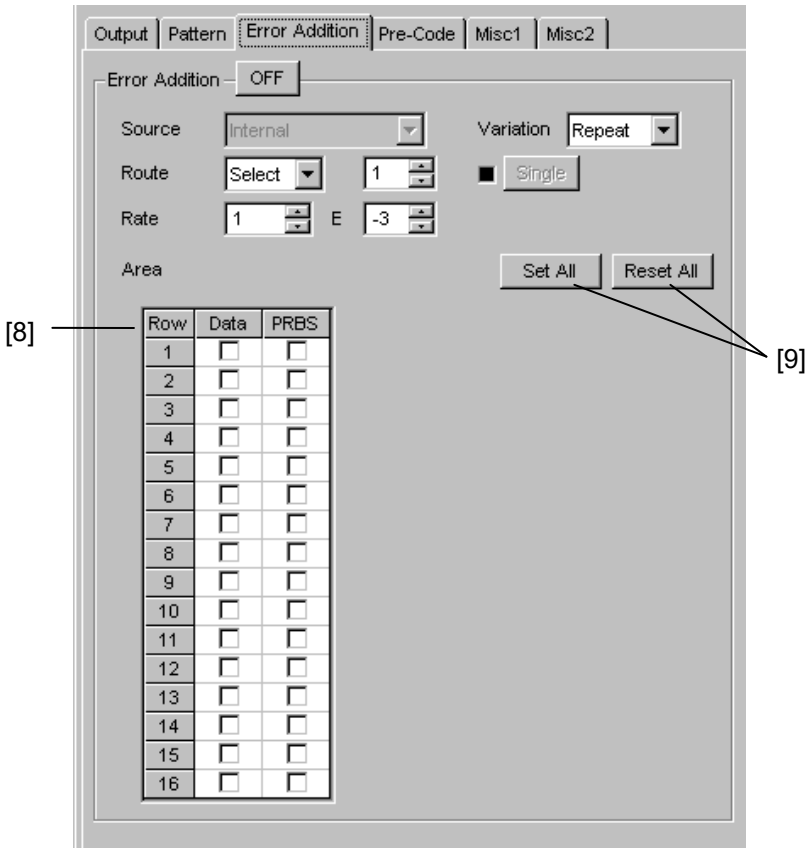


Figure 7.11.4-2 Error Addition Settings (Mixed Pattern)

Table 7.11.4-1 Error Addition Setting Commands

No.	Setting Items	Commands
[1]	Error Addition ON/OFF	:SOURce:PATTern:EADDition:SET
		:SOURce:PATTern:EADDition:SET?
[2]	Source	:SOURce:PATTern:EADDition:SOURce
		:SOURce:PATTern:EADDition:SOURce?
[3]	Route	:SOURce:PATTern:EADDition:SELect
		:SOURce:PATTern:EADDition:SELect?
[4]	Error Addition Route setting	:SOURce:PATTern:EADDition:ROUTe
		:SOURce:PATTern:EADDition:ROUTe?
[5]	Rate	:SOURce:PATTern:EADDition:RATE
		:SOURce:PATTern:EADDition:RATE?
[6]	Variation	:SOURce:PATTern:EADDition:VARiation
		:SOURce:PATTern:EADDition:VARiation?
[7]	Single	:SOURce:PATTern:EADDition:SINGle
[8]	Error Addition Area setting	:SOURce:PATTern:EADDition:AREA
		:SOURce:PATTern:EADDition:AREA?
[9]	Set All/Reset All	:SOURce:PATTern:EADDition:ALLSet

: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	Disables 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

: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.

: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	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.

: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
	<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? 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.

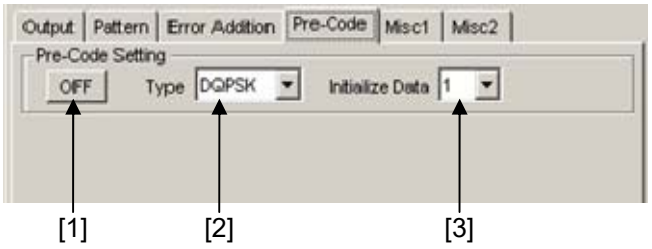


Figure 7.11.5-1 Pre-Code Tab

Table 7.11.5-1 Pre-Code Setting Commands

No.	Setting Items	Commands
[1]	Pre-Code ON/OFF	:SOURce:PRECode:SET
		:SOURce:PRECode:SET?
[2]	Type	:SOURce:PRECode:TYPE
		:SOURce:PRECode:TYPE?
[3]	Initialize Data	:SOURce:PRECode:INITialize
		:SOURce:PRECode:INITialize?

:SOURce:PRECode:SET <boolean>

Parameter <boolean> = <BOOLEAN PROGRAM DATA>
OFF to 0 Pre-Code OFF
ON or 1 Pre-Code ON

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 Pre-Code OFF
1 Pre-Code ON

Function Queries the Pre-Code Setting ON or OFF.

Example

```
> :SOURce:PRECode:SET?
< 1
```

Compatibility Compatible with MU181020A/B.

: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

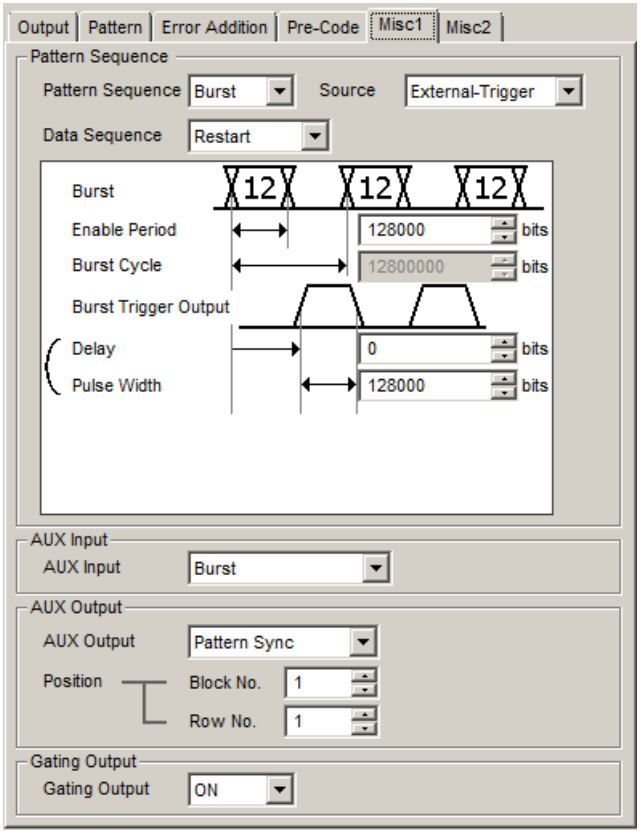


Figure 7.11.6-1 Misc1 Tab

7.11.6.1 Pattern Generation Setting Commands

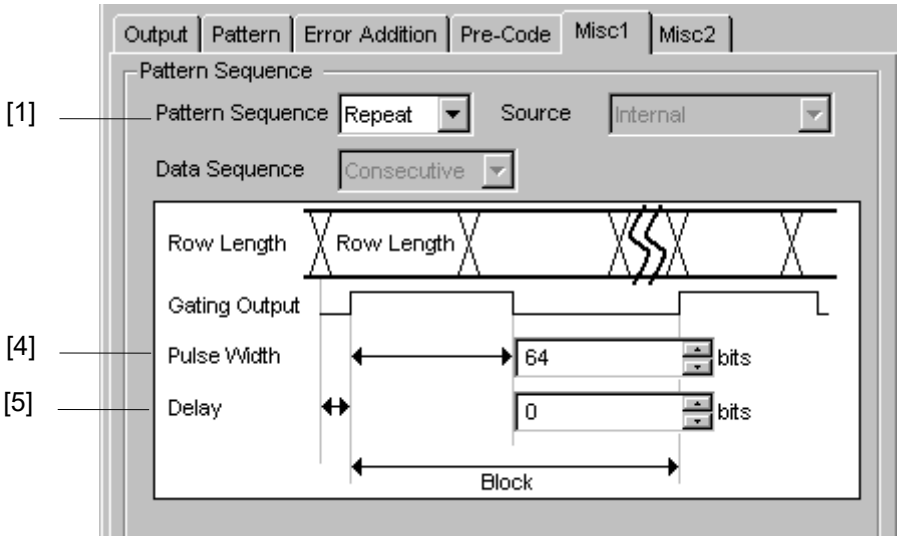


Figure 7.11.6.1-1 Pattern Sequence Setting (When Generating Repeat Signal)

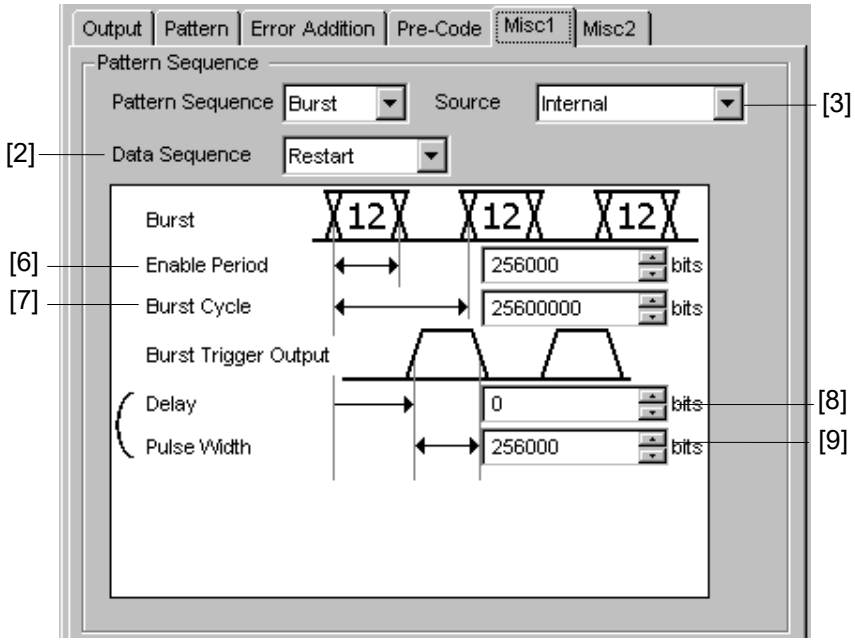


Figure 7.11.6.1-2 Pattern Sequence Setting (When Generating Burst Signal)

Table 7.11.6.1-1 Pattern Generation Setting Commands

No.	Setting Items	Commands
[1]	Pattern Sequence	:SOURce:PATTern:OMODe
		:SOURce:PATTern:OMODe?
[2]	Data Sequence	:SOURce:PATTern:BURSt:SEQuence
		:SOURce:PATTern:BURSt:SEQuence?
[3]	Source	:SOURce:PATTern:BURSt:MODE
		:SOURce:PATTern:BURSt:MODE?
[4]	Pulse Width	:SOURce:PATTern:REPeat:PULSewidth
		:SOURce:PATTern:REPeat:PULSewidth?
[5]	Delay (For Repeat)	:SOURce:PATTern:REPeat:TRIGdelay
		:SOURce:PATTern:REPeat:TRIGdelay?
[6]	Enable Period	:SOURce:PATTern:BURSt:ELENgth
		:SOURce:PATTern:BURSt:ELENgth?
[7]	Burst Cycle	:SOURce:PATTern:BURSt:CYCLe
		:SOURce:PATTern:BURSt:CYCLe?
[8]	Delay (For Burst)	:SOURce:PATTern:BURSt:TRIGdelay
		:SOURce:PATTern:BURSt:TRIGdelay?
[9]	Pulse Width	:SOURce:PATTern:BURSt:PULSewidth
		:SOURce:PATTern:BURSt:PULSewidth?

: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	Partially compatible with MU181020A/B.

:SOURce:PATTern:OMODe?

Response	<mode> = <CHARACTER RESPONSE DATA> REP, BURS
Function	Queries the signal generation method.
Example	`> :SOURce:PATTern:OMODe?` `< BURS`
Compatibility	Partially compatible with MU181020A/B.

: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	Partially compatible with MU181020A/B.

: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	Partially compatible with MU181020A/B.

:SOURce:PATTern:BURSt:MODE <mode>

Parameter	<mode> = <CHARACTER PROGRAM DATA>
	INTernal — Generates based on internal reference signal
	EXTTrig — Generates based on external signal trigger edge
	EXTernal — Generates during the high level interval of external signal trigger input
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	<numeric> = <DECIMAL NUMERIC PROGRAM DATA>

	PRBS/ZeroSub/Data	Mixed
Independent	0 to 34359738240 bits/ 8 bits Step	0 to 2415918976/ 8 bits Step
2 ch Combination	0 to 68719476480/ 16 bits Step	0 to 4831837952/ 16 bits Step
4 ch Combination	0 to 137438952960/ 32 bits Step	0 to 9663675904/ 32 bits Step

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
<numeric> = <NR1 NUMERIC RESPONSE DATA>
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:
```
> :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
<numeric> = <DECIMAL NUMERIC PROGRAM DATA>

	PRBS/ZeroSub/Data	Mixed
Independent	0 to 34359738240 bits/ 8 bits Step	0 to 2415918976/ 8 bits Step
2 ch Combination	0 to 68719476480/ 16 bits Step	0 to 4831837952/ 16 bits Step
4 ch Combination	0 to 137438952960/ 32 bits Step	0 to 9663675904/ 32 bits Step

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:
```
> :SOURce:PATTern:REPeat:TRIGdelay 256
```

Compatibility
Partially compatible with MU181020A/B (the parameter range varies depending on the settings).

:SOURce:PATTern:REPeat:TRIGdelay?

Response
<numeric> = <NR1 NUMERIC RESPONSE DATA>
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
```
> :SOURce:PATTern:REPeat:TRIGdelay?
< 256
```

Compatibility
Partially compatible with MU181020A/B (the parameter range varies depending on the settings).

:SOURce:PATTern:BURSt:ELENgth <numeric>

Parameter

<numeric> = <DECIMAL NUMERIC PROGRAM DATA>

	Internal	External
Independent	1024 to 2147483392 bits /256 bits Step	1,024 to 2147483648 bits /256 bits Step
2 ch Combination	2048 to 4294966784 bits /512 bits Step	2,048 to 4294967296 bits /512 bits Step
4 ch Combination	4096 to 8589933568 bits /1024 bits Step	4,096 to 8589934592 bits /1024 bits Step

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

<numeric> = <NR1 NUMERIC RESPONSE DATA>

1024 to 8589934592 1,0240 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>

Parameter

<numeric> = <DECIMAL NUMERIC PROGRAM DATA>

Independent	1,536 to 2,147,483,648 bits/256 bits Step
2 ch Combination	3,072 to 4,294,967,296 bits/512 bits Step
4 ch Combination	6,144 to 8,589,934,592 bits/1024 bits Step

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).

: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).

: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>

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 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

<port> = <CHARACTER PROGRAM DATA>

BURStout1	Burst Output
BURStout2	Burst Output2

Response

Valid only when AUX Output is set to Burst Output2.

<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? BURS1
< 1024
```

Compatibility

Partially compatible with MU181020A/B (the parameter range varies depending on the settings).

7.11.6.2 AUX Input setting commands

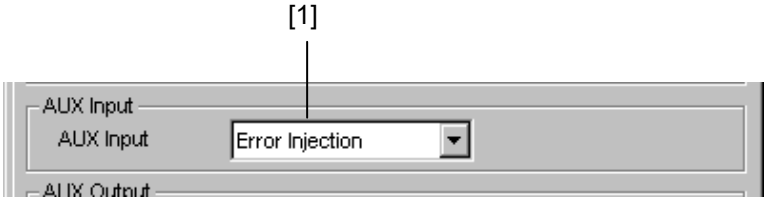


Figure 7.11.6.2-1 AUX Input Setting

Table 7.11.6.2-1 AUX Input Setting Commands

No.	Setting Items	Commands
[1]	AUX Input	:INPut:INPut:FUNCtion
		:INPut:INPut:FUNCtion?

: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

Figure 7.11.6.3-1 AUX Output Setting (When 1/N Clock Is Selected)

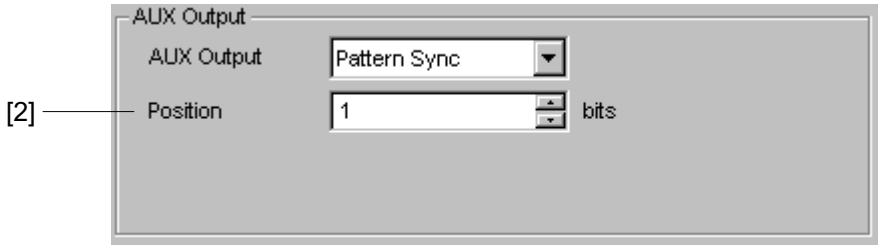


Figure 7.11.6.3-2 AUX Output Setting (When Pattern Sync Is Selected: PRBS, Zero-Substitution, Data Pattern)

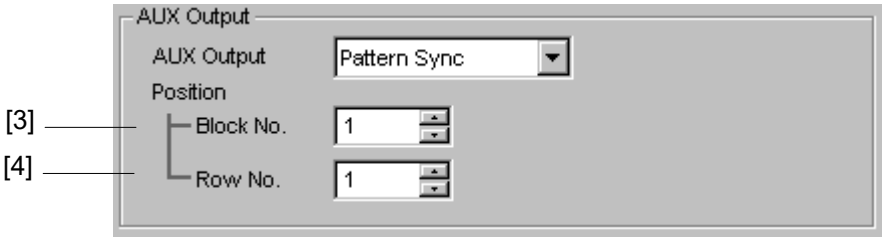


Figure 7.11.6.3-3 AUX Output Setting (When Pattern Sync Is Selected: Mixed Data Pattern)

Table 7.11.6.3-1 AUX Output Setting Commands

No.	Setting Items	Commands
[1]	AUX Output	:OUTPut:SYNC:SOURce
		:OUTPut:SYNC:SOURce?
[2]	Position (PRBS, Zero-Substitution, Data)	:OUTPut:SYNC:POSition
		:OUTPut:SYNC:POSition?
[3]	Block No. (Mixed-Data)	:OUTPut:SYNC:MIXData:BLOCk
		:OUTPut:SYNC:MIXData:BLOCk?
[4]	Row No. (Mixed-Data)	:OUTPut:SYNC:MIXData:ROW
		:OUTPut:SYNC:MIXData:ROW?

: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
	Omit <numeric> when NCLock (1/n clocks) is not selected.
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>
	Omit <numeric> for other than NCL (1/N clocks).
	4 to 512
Function	Queries the output signal for synchronization output.

Example	`> :OUTPut:SYNC:SOURce?`
	`< PATT`
	`> :OUTPut:SYNC:SOURce?`
	`< NCL,512`

Compatibility	Partially compatible with MU181020A/B (the parameter range varies depending on the settings).

:OUTPut:SYNC:POSition <numeric>

Parameter	<numeric> = <DECIMAL NUMERIC PROGRAM DATA>
	Independent: 1 to 34359738105/ 8 bits Step — 1 to 34,359,738,105 bits
	2 ch Combination: 1 to 68719476209/ 16 bits Step — 1 to 68,719,476,209 bits
	4 ch Combination: 1 to 137438952417/ 32 bits Step — 1 to 137,438,952,417 bits
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 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	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 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	Compatible with MU181020A/B.

: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

Figure 7.11.6.4-1 Gating Output Setting

Table 7.11.6.4-1 Gating Output Setting Commands

No.	Setting Items	Commands
[1]	Gating Output	:OUTPut:SYNC:GATing:SOURce
		:OUTPut:SYNC:GATing:SOURce?

: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
```

7.11.7 Misc 2 Tab

7.11.7.1 Clock Setting commands

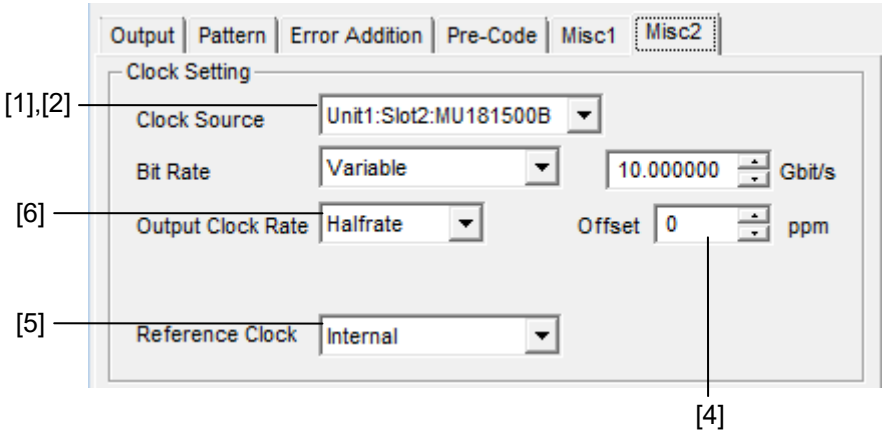


Figure 7.11.7.1-1 Clock Setting (When MU181000B Is Selected)

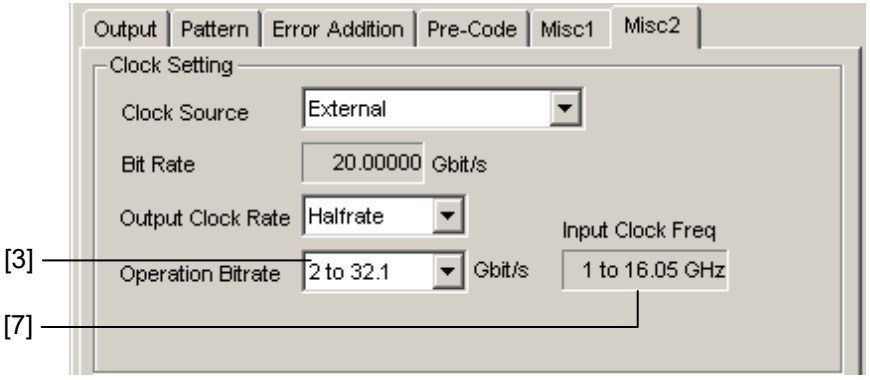


Figure 7.11.7.1-2 Clock Setting (When External Is Selected)

Table 7.11.7.1-1 Reference Clock Setting Commands

No.	Setting Items	Commands
[1]	Clock Source	:SYSTem:INPut:CSELect
		:SYSTem:INPut:CSELect?
[2]	Queries system built-in synthesizer	:SYSTem:INPut:CSELect:MODule?
[3]	Operation Bitrate	:SYSTem:OUTPut:BITRate
		:SYSTem:OUTPut:BITRate?
[4]	Clock Offset	:OUTPut:CLOCK:OFFSet:PPM
		:OUTPut:CLOCK:OFFSet:PPM?
[5]	Reference Clock Source	:OUTPut:RCLock:SELect
		:OUTPut:RCLock:SELect?
[6]	Output Clock Rate	:SYSTem:OUTPut:CRATe
		:SYSTem:OUTPut:CRATe?
[7]	Input Clock Freq.	:SYSTem:INPut:CFReq?

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

<input> = <CHARACTER PROGRAM DATA>

EXTernal	External synthesizer
INTernal[1 to 7]	Built-in synthesizer or jitter generation source

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

<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 model name	MU181000A/B or MU181500B

Function

Queries the synthesizer of the selecting clock input source.

Example

```
> :SYSTem:INPut:CSELect?
< INT2,"1:4 MU181500B"
```

Compatibility

Compatible with MU181500B.

: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 model name	MU181000A, MU181000B, or MU181500B

NONE is returned when built-in synthesizer and jitter signal source are not present.

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>

Parameter
<input> = <CHARACTER PROGRAM DATA>
When using the external clock:

HIGH	20 to 28.1 G (32.1G) bit/s (when setting Fullrate clock)
MIDDle	16 to 20.4 Gbit/s (when setting Fullrate clock)
LOW	2.4 to 16 Gbit/s (when setting Fullrate clock)
HALF	2.4 to 28.1 (32.1) G bit/s (when setting Halfrate clock)
HIGH1	25 to 28.1 (32.1) G bit/s (when setting Fullrate/Halfrate clock)

When using the eternal clock and MU181500B Jitter Modulation Source:

HIGH	25to 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.

: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?

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 MU181500B and MU181000A/B.

:SYSTem:OUTPut:CRATe <rate>

Parameter	<rate> = <CHARACTER PROGRAM DATA> HALFrate 1/2 clock output to data FULLrate 1/1 clock output to data
Function	Selects output data and clock correlation.
Example	To set output data/clock to Half Rate: > :SYSTem:OUTPut:CRATe HALF
Compatibility	Incompatible with existing models.

:SYSTem:OUTPut:CRATe?

Response	<rate> = <CHARACTER RESPONSE DATA> HALF, FULL
Function	Queries the relationship between output data and clock.
Example	> :SYSTem:OUTPut:CRATe? < HALF
Compatibility	Incompatible with existing models.

:SYSTem:INPut:CFReq?

Response	<string> = <STRING RESPONSE DATA>
	When using the external clock:
	"10 to 16.05 GHz" (when setting Fullrate clock)
	"10 to 14.05 GHz" Without Opt-x01
	"8 to 10.02 GHz" (when setting Fullrate clock)
	"2.4 to 16 GHz" (when setting Fullrate clock)
	"1.2 to 16.05 GHz" (when setting Halfrate clock)
	"1.2 to 14.05 GHz" Without Opt-x01
	"6.25 to 8.025 GHz" (when setting Fullrate/Halfrate clock)
	"6.25 to 7.025 GHz" Without Opt-x01
	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	`> :SYSTem:INPut:CFReq?`
	`< "1.2 to 16.05 GHz"`
Compatibility	Incompatible with the existing modules.

7.11.7.2 Combination setting commands

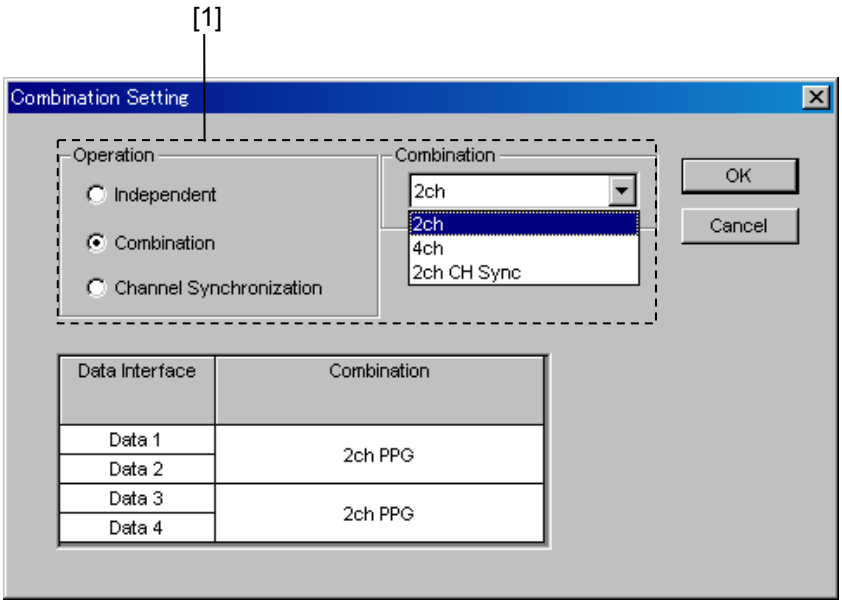


Figure 7.11.7.2-1 Combination Setting Commands (Combination)

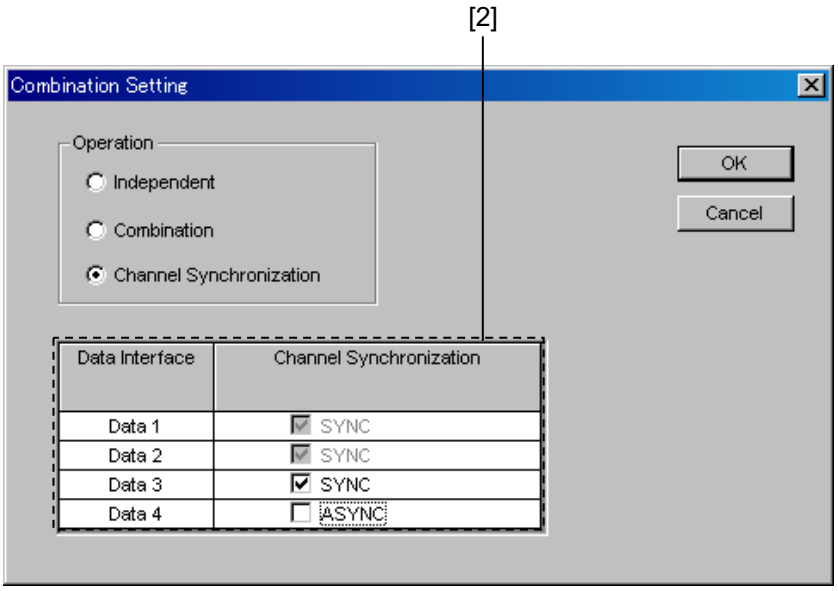


Figure 7.11.7.2-2 Combination Setting Commands (Channel Synchronization)

Table 7.11.7.2-1 Combination Setting Commands

No.	Setting Items	Commands
[1]	Operation	:MCOMbination:OPERation:SETTing
		:MCOMbination:OPERation:SETTing?
[2]	Combination	:MCOMbination:OPERation:CHSetting
		:MCOMbination:OPERation:CHSetting?

:MCOMbination:OPERation:SETTing <numeric>

Parameter	<numeric> = <DECIMAL NUMERIC PROGRAM DATA>	
	0	Independent
	1	Channel Synchronization
	2	2ch Combination
	3	2ch Combination CH Sync
	4	4ch Combination
	5	64G × 2ch Combination
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	<numeric> = <NR1 NUMERIC RESPONSE DATA>	
	0	Independent
	1	Channel Synchronization
	2	2ch Combination
	3	2ch Combination CH Sync
	4	4ch Combination
	5	64G × 2ch Combination
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	<interface> = <DECIMAL NUMERIC PROGRAM DATA>	
	2 to 14	Total number of Data Interface that can configure channel synchronization (decimal)
	Available bits	
	2 (Bit1)	Selects Data2
	4 (Bit2)	Selects Data3
	8 (Bit3)	Selects Data4
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?

Parameter	<interface> = <NR1 NUMERIC RESPONSE DATA>	
	2 to 14	Total number of Data Interface that can configure channel synchronization (decimal)
	Available bits	
	2 (Bit1)	Data2
	4 (Bit2)	Data3
	8 (Bit3)	Data4
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

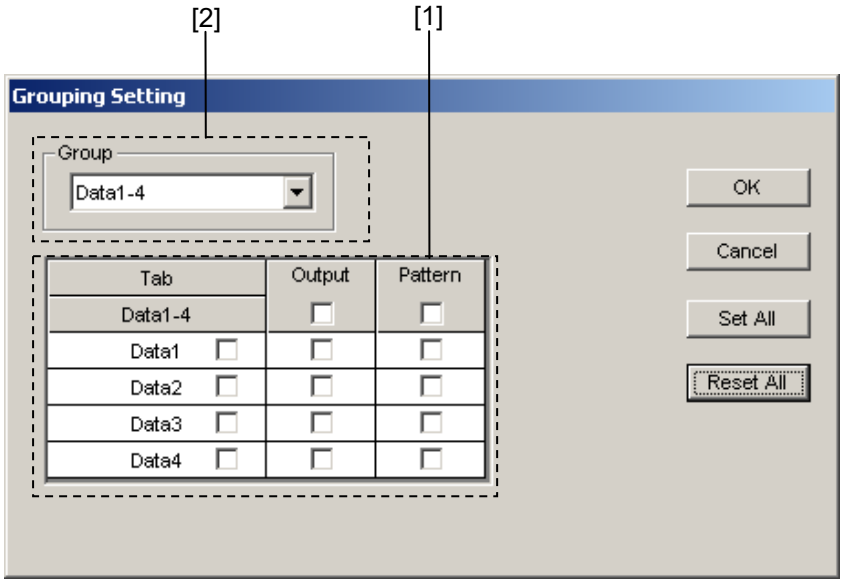


Figure 7.11.7.3-1 Grouping Setting Dialog Box

Table 7.11.7.3-1 Grouping Setting Commands

No.	Setting Items	Commands
[1]	Selects tabs to be grouped together.	:SOURce:GROup:SETTing
		:SOURce:GROup:SETTing?
[2]	Selects Data Ifs to be grouped together.	:SOURce:GROup:SELect
		:SOURce:GROup:SELect?

: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 Data 2 together:

```
> :SOURce:GROup:SETTing PATTern,3
```

To group the Pattern tabs of Data1 to Data 4 together:

```
> :SOURce:GROup:SETTing OUTPut,15
```

Compatibility

Incompatible with existing models.

: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	<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	`> :SOURce:GROup:SELect?`	
	`< 2`	
Compatibility	Incompatible with existing models.	

7.12 28G/32G bit/s ED Commands

This section describes the setting and query commands for the MU183040A 28G/32G bit/s ED, MU183041A 28G/32G bit/s 4ch ED, MU183040B 28G/32G bit/s High Sensitivity ED, MU183041B 28G/32G bit/s 4ch High Sensitivity ED (hereafter, MU183040A/40B/41A/41B). 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 7.12.1.1-1 Common Setting Commands

Setting Items	Commands
Number of the unit to be operated	:UENTry:ID
	:UENTry:ID?
Number of the module to be operated (slot position)	:MODule:ID
	:MODule:ID?
Number of the data interface to be operated	:INTerface:ID
	:INTerface:ID?
Setting of automatic measurement to be operated	:SYSTem:CFUNction
	:SYSTem:CFUNction?
Query for error message	:SYSTem:ERRor?
Query for SCPI version	:SYSTem:VERSion?
Query for software status	:SYSTem:CONDition?
Query for hardware system configuration	:SYSTem:ORGanization:HARDware?
Query for system error	:SYSTem:INFormation:ERRor?
Terminator type	:SYSTem:TERMination
	:SYSTem:TERMination?
Query for system error	:SYSTem:CONDition:UNITs?
Query for mainframe information	:SYSTem:UNIT?
Query for module information	:SYSTem:MODule?
Setting of measurement result drawing	:SYSTem:DISPlay:RESult
	:SYSTem:DISPlay:RESult?
Module screen display	:DISPlay:ACTive

: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.

:INTerface:ID <interface_number>

Parameter
<interface_number> = <DECIMAL NUMERIC PROGRAM DATA>
1 to 4 Data interface number
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?

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

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>

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
ASE32	Auto Search (MU183040A/41A/40B/41B)
EMAR32	Eye Margin measurement (MU183040A/41A/40B/41B)
EDI32	Eye Diagram measurement (MU183040A/41A/40B/41B)
BTUB32	Bathtub measurement (MU183040A/41A/40B/41B)
AADJ32	Auto Adjust (MU183040A/41A/40B/41B)
QAN32	Q measurement (MU183040A/41A/40B/41B)
PAMB	PAM BER measurement (MU183040B/41B)
ECT	Eye Contour measurement (MU183040A/41A/40B/41B)

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?

Response	<function> = <CHARACTER RESPONSE 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
	ASE32	Auto Search (MU183040A/41A/40B/41B)
	EMAR32	Eye Margin measurement (MU183040A/41A/40B/41B)
	EDI32	Eye Diagram measurement (MU183040A/41A/40B/41B)
	BTUB32	Bathtub measurement (MU183040A/41A/40B/41B)
	AADJ32	Auto Adjust (MU183040A/41A/40B/41B)
	QAN32	Q measurement (MU183040A/41A/40B/41B)
	PAMB	PAM BER measurement (MU183040B/41B)
	ECT	Eye Contour measurement (MU183040A/41A/40B/41B)
Function	Queries the automatic measurement function being performed.	
Example	`> :SYSTem:CFUNction?` `< ASE32`	
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. 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 YYYY: 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>

XXXXXXXXX	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

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
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 7.12.1.1-2 Option Character Correspondence Table

Model/Name	Option Number	Option Name
MP1800A Signal Quality Analyzer	OPT014	2-Slot PPG/ED
	OPT015	4-Slot PPG/ED
	OPT016	6-Slot PPG/ED
	OPT032	32Gbit/s PPG/ED support
MT1810A 4 Slot Chassis	OPT014	2-Slot PPG/ED
	OPT015	4-Slot PPG/ED
	OPT032	32Gbit/s PPG/ED support
MX180000A Signal Quality Analyzer Control Software	–	–
MU181000A 12.5GHz Synthesizer	OPTx01	Jitter Modulation
MU181000B 12.5GHz 4port Synthesizer	OPTx01	Jitter Modulation
MU181020A 12.5Gbit/s PPG	OPT001	9.8 to 12.5 Gbit/s
	OPT002	0.1 to 12.5 Gbit/s
	OPTx10	Variable Data Output (0.05 to 0.8 Vp-p)
	OPTx11	Variable Data Output (0.25 to 2.5 Vp-p)
	OPTx12	High Performance Data Output (0.05 to 2.0 Vp-p)
	OPTx13	Variable Data Output (0.5 to 3.5 Vp-p)
	OPTx21	Differential Clock Output (0.1 to 2.0 Vp-p)
	OPTx30	Variable Data Delay
MU181040A 12.5Gbit/s ED	OPT001	9.8 to 12.5 Gbit/s
	OPT002	0.1 to 12.5 Gbit/s
	OPTx20	Clock Recovery
	OPTx30	Variable Clock Delay
MU181600A Optical Transceiver (XFP)	–	–
MU181601A Optical Transceiver (SFP)	–	–

Table 7.12.1.1-2 Option Character Correspondence Table (Cont'd)

Model/Name	Option Number	Option Name
MU181620A Stressed Eye Transmitter	OPTx01	1310nm Reference
	OPTx02	1550nm Reference
	OPTx03	1310nm/1550nm Reference
	OPTx11	1310nm Stressed Eye
	OPTx12	1550nm Stressed Eye
	OPTx13	1310nm/1550nm Stressed Eye
MU181640A Optical Receiver	OPTx04	Band Width 8.5 GHz
MU181800A 12.5GHz Clock Distributor	–	–
MU181020B 14Gbit/s PPG	OPT002	0.1 to 14 Gbit/s
	OPTx11	Variable Data Output (0.25 to 2.5 Vp-p)
	OPTx12	High Performance Data Output (0.05 to 2.0 Vp-p)
	OPTx13	Variable Data Output (0.5 to 3.5 Vp-p)
	OPTx21	Differential Clock Output (0.1 to 2.0 Vp-p)
	OPTx30	Variable Data Delay
MU181040B 14Gbit/s ED	OPT002	0.1 to 14 Gbit/s
	OPTx20	Clock Recovery
	OPTx30	Variable Clock Delay
MU182020A 25Gbit/s 1ch MUX MU182021A 25Gbit/s 2ch MUX	OPTx01	28Gbit/s Extension
	OPTx02	Clock Input Band Switch
	OPTx10	Variable Data Output (0.25 to 1.75Vp-p)
	OPTx11	Variable Data Output (0.5 to 2.5Vp-p)
	OPTx12	Variable Data Output (0.5 to 2.0Vp-p)
	OPTx13	Variable Data Output (0.5 to 3.5Vp-p)
	OPTx21	Differential Clock Output (0.5 to 2.0Vp-p)
	OPTx30	25Gbit/s Variable Data Delay
	OPTx31	28Gbit/s Variable Data Delay
MU182040A 25Gbit/s 1ch DEMUX MU182041A 25Gbit/s 2ch DEMUX	OPTx01	28Gbit/s Extension
	OPTx02	Clock Input Band Switch
	OPTx30	25GHz Variable Clock Delay
	OPTx31	28GHz Variable Clock Delay
MU183020A 28G/32G bit/s PPG	OPTx01	32G bit/s Extension
	OPTx10	1ch 1V Data Output
	OPTx12	1ch 2V Data Output
	OPTx13	1ch 3.5V Data Output
	OPTx30	1ch Data Delay
	OPTx20	2ch 1V Data Output
	OPTx22	2ch 2V Data Output
	OPTx23	2ch 3.5V Data Output
	OPTx31	2ch Data Delay

Table 7.12.1.1-2 Option Character Correspondence Table (Cont'd)

Model/Name	Option Number	Option Name
MU183021A 28G/32G bit/s 4ch PPG	OPTx01	32G bit/s Extension
	OPTx10	4ch 1V Data Output
	OPTx12	4ch 2V Data Output
	OPTx13	4ch 3.5V Data Output
	OPTx30	4ch Data Delay
MU183040A 28G/32G bit/s ED	OPTx01	32 Gbit/s Extension
	OPTx10	1ch ED
	OPTx20	2ch ED
MU183041A 28G/32G bit/s 4ch ED	OPTx01	32 Gbit/s Extension
MU183040B 28G/32G bit/s High Sensitivity ED	OPTx01	32 Gbit/s Extension
	OPTx10	1ch ED
	OPTx20	2ch ED
MU183041B 28G/32G bit/s 4ch High Sensitivity ED	OPTx01	32 Gbit/s Extension

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>

XXXXXXXXX	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,OPT2
20,MU181040A,6201234571,1.00.00,1.00.00,1.00.00,OPT002,O
PT220
```

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.

: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>
XXXXXXXXX Mainframe name (e.g.,: MP1800A)
See Table 7.12.1.1-2 "Option Character Correspondence Table".

Note:

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

<module1> to <module64> = <STRING RESPONSE DATA>
XXXXXXXXX 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.

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.

: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> =

<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>

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>

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>

Parameter
<numeric> = <NR1 NUMERIC PROGRAM DATA>
1 to 6 Slot
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>
XXXXXXXXX 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

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.

Tab ID	Tab Name
1	Result
2	Measurement
3	Pattern
4	Input
5	Capture
6	Misc 1
7	Misc 2

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 unit1 slot4)

```
> :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

Setting Items	Commands
Quick Open	:SYSTem:MMEMory:QRECall
Quick Save	:SYSTem:MMEMory:QSTore
Open	:SYSTem:MMEMory:RECall
Save	:SYSTem:MMEMory:STORe
Screen Copy Execute	:SYSTem:PRINt:COPY
Screen Copy Setup	:SYSTem:PRINt:COPYsetup:SET
	:SYSTem:PRINt:COPYsetup:SET?
Print	:SYSTem:PRINt:EALarm
Printer Setup	:SYSTem:PRINt:PRINter:SET
	:SYSTem:PRINt:PRINter:SET?
Initialize	:SYSTem:MEMory:INITialize

: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>
"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.

: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.

: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	

: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:	
	`> :SYSTem:PRINt:EALarm ERA_C2,"1:4:1 MU183040A"`	
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

Figure 7.12.1.3-1 Common Function Buttons

Table 7.12.1.3-1 Common Function Button Setting Commands

Setting Items	Commands
Meas. Start	:SENSe:MEASure:ASTRt
Meas. Stop	:SENSe:MEASure:ASTP
(Query for measurement status)	:SENSe:MEASure:ASTate?
Open	Refer to 7.12.1.2 “Commands for File menu”.
Save	Refer to 7.12.1.2 “Commands for File menu”.
Print	Refer to 7.12.1.2 “Commands for File menu”.
Error ON/OFF	:SYSTem:BEEPer:ERRor:SET
	:SYSTem:BEEPer:ERRor:SET?
Alarm ON/OFF	:SYSTem:BEEPer:ALARm:SET
	:SYSTem:BEEPer:ALARm:SET?
System Error ON/OFF	:SYSTem:BEEPer:SYSTem:SET
	:SYSTem:BEEPer:SYSTem:SET?
System Error	:SYSTem:BEEPer:SYSTem:TYPE
	:SYSTem:BEEPer:SYSTem:TYPE?

: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.

: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	
	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.	

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]?

Response

<numeric> = <NR1 NUMERIC RESPONSE DATA>

0 to 16383	Sum total of the bit of event register (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

Queries event at 28G/32G bit/s ED status.

Example

```
> :INSTrument:EDG32:EVENt?
```

or

```
> :INSTrument:EDG32?
< 1
```

:INSTrument:EDG32:CONDition?

Response	<numeric> = <NR1 NUMERIC RESPONSE DATA>	
	0 to 16383	Sum total of the bit of condition register (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	Queries condition at 28G/32G bit/s ED status.	
Example	`> :INSTrument:EDG32:CONDition?`	
	`< 1`	

:INSTrument:EDG32:PTRansition <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 (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	<numeric> = <NR1 NUMERIC RESPONSE DATA>	
	0 to 16383	Sum total of the bit of transition filter (DECIMAL)
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:RESet

Function

Initializes event at 28G/32G bit/s ED status.

Example

```
> :INSTrument:EDG32:RESet
```

7.12.1.5 Data Interface setting commands

Figure 7.12.1.5-1 Data Interface Select

Table 7.12.1.5-1 Data Interface Setting Commands

Setting Items	Commands
Data Interface Select	:INTerface:ID
	:INTerface:ID?

: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

:INTerfaceme: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

Setting Items	Commands
Open	:SYSTem:MMEMory:PATTern:RECall
Save	:SYSTem:MMEMory:PATTern:STORe

: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>

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:MMEMory:PATTern:STORe "C:\Test\example",TXT
```

Compatibility

Incompatible with existing models.

7.12.2 Commands Related to Result Tab

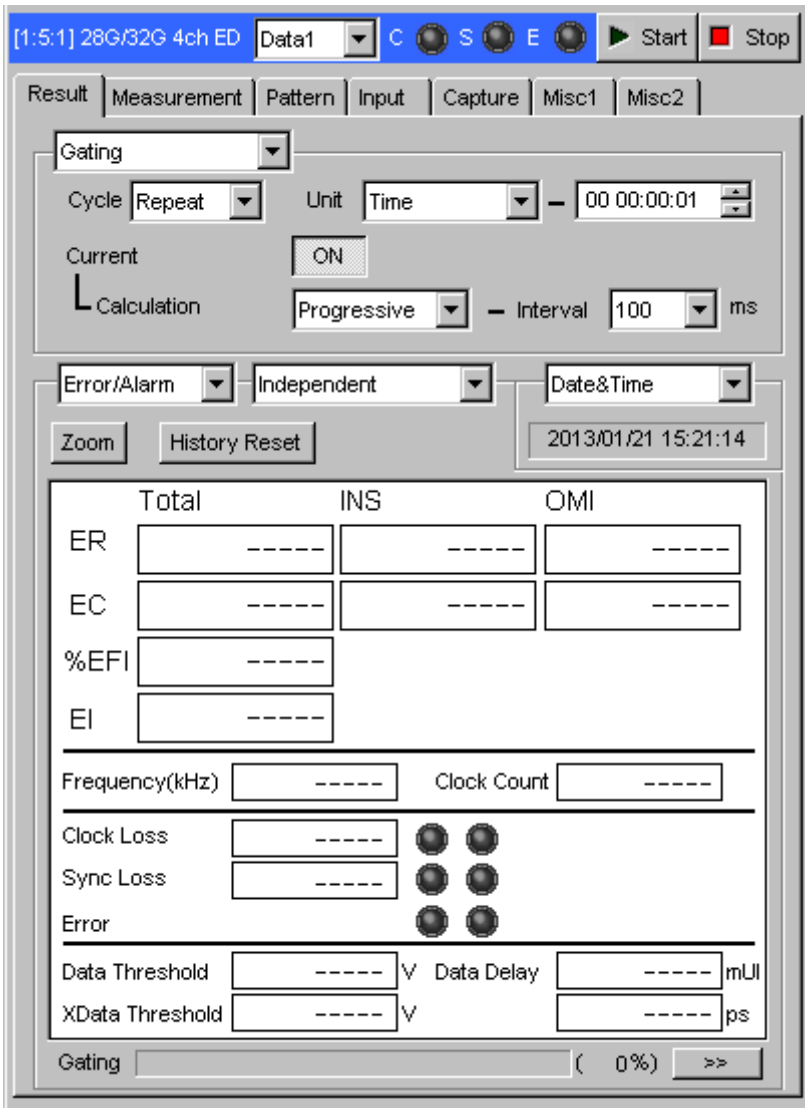


Figure 7.12.2-1 Result Tab

7.12.2.1 Result setting commands

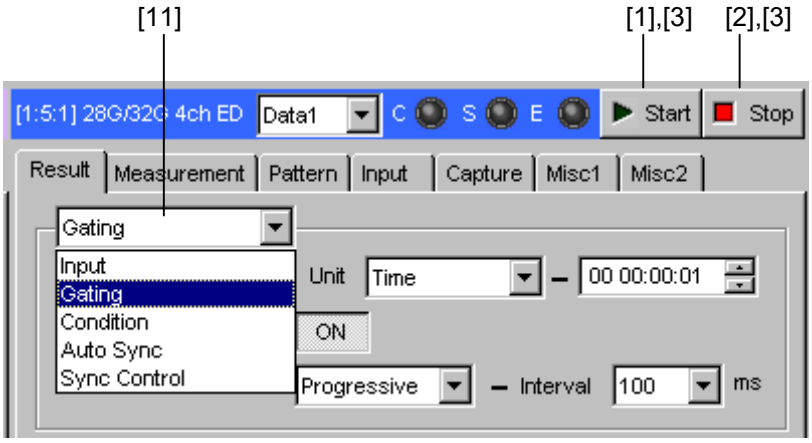


Figure 7.12.2.1-1 Start/Stop Settings

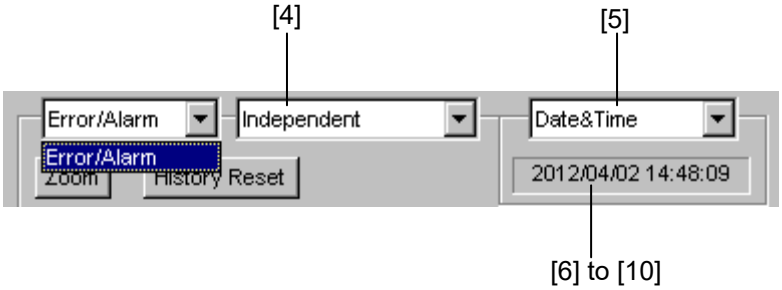


Figure 7.12.2.1-2 Result Settings

Table 7.12.2.1-1 Result Setting Commands

No.	Setting Items	Commands
[1]	Start	:SENSe:MEASure:STARt
[2]	Stop	:SENSe:MEASure:STOP
[3]	Measurement state query	:SENSe:MEASure:EALarm:STATe?
[4]	Selection of result combination status	:DISPlay:RESult:COMBination
		:DISPlay:RESult:COMBination?
[5]	Display time setting	:DISPlay:RESult:TIME
		:DISPlay:RESult:TIME?
[6]	Query for measurement start time	:SENSe:MEASure:EALarm:STARt?
[7]	Query for measurement stop time	:SENSe:MEASure:EALarm:STOP?
[8]	Query for measurement elapsed time	:SENSe:MEASure:EALarm:ELAPsed?
[9]	Query for measurement remaining time	:SENSe:MEASure:EALarm:TIMed?
[10]	Query for alarm occurrence time	:SENSe:MEASure:EALarm:AOCCur?
[11]	Display measurement items	:SENSe:DISPlay:SETTing

: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

:DISPlay:RESult:COMBination?

Response	<item> = <NR1 NUMERIC RESPONSE DATA>	
	1	Independent
	21	2ch Combination1-2
	22	2ch Combination3-4
	23	Simultaneous display of 2ch Combination Data1/2 and Data3/4.
	4	4ch Combination
Function	Queries the status of the displayed combination.	
Example	`> :DISPlay:RESult:COMBination?`	
	`< 1`	
Compatibility	Incompatible with existing models	

: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 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: <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 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: <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 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.

: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>		
	<da<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 <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 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

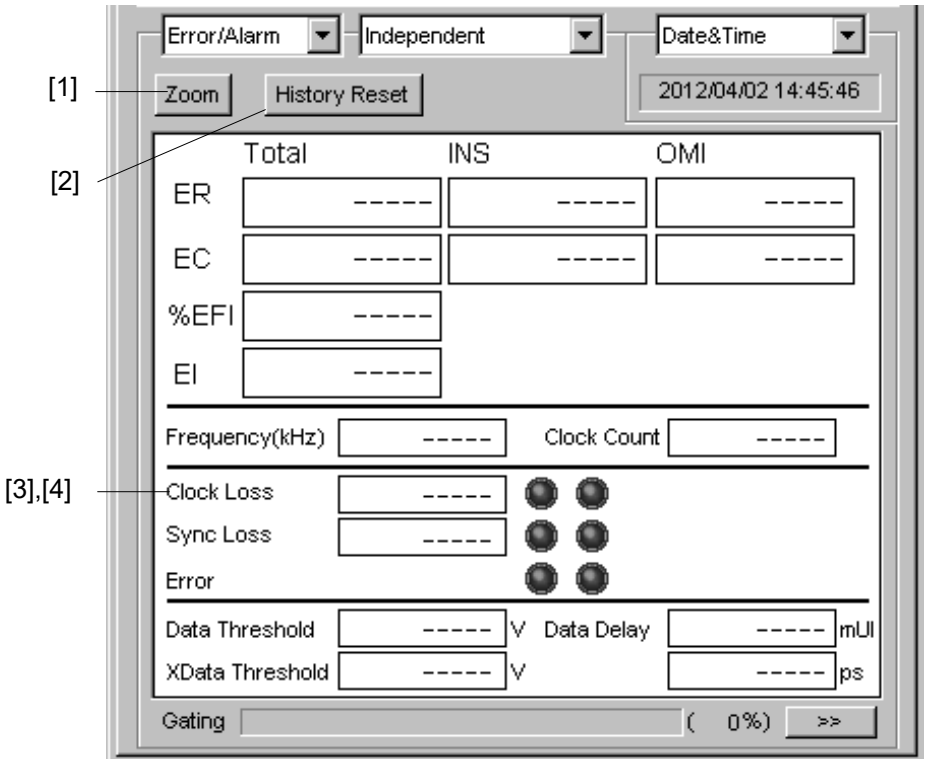


Figure 7.12.2.2-1 Error/Alarm Result Display Area

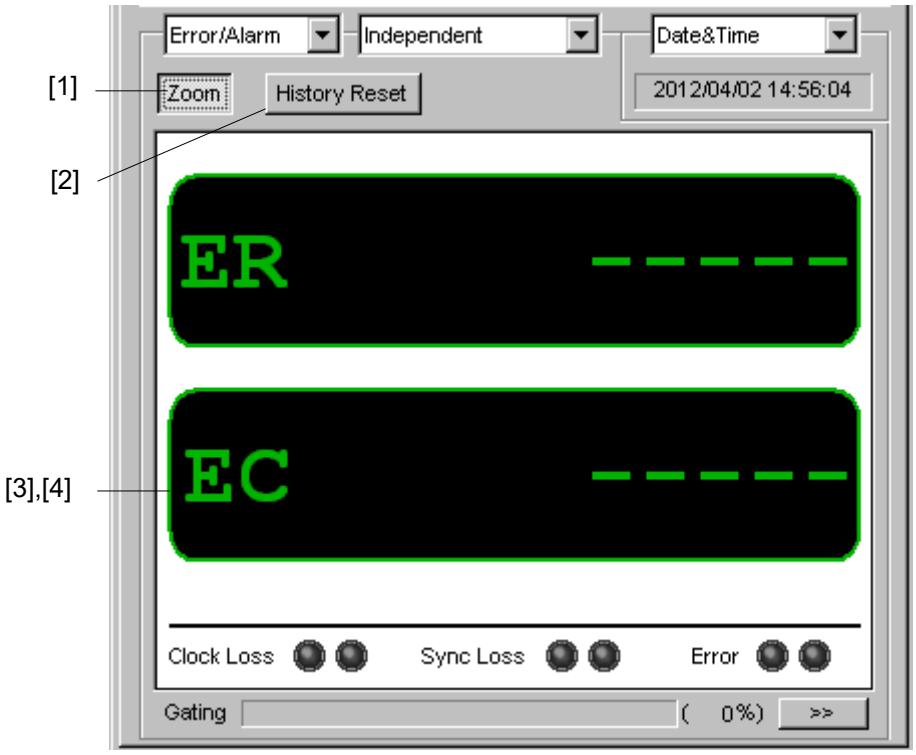


Figure 7.12.2.2-2 Error/Alarm Result Display Area (When Zoom Is Clicked)

Table 7.12.2.2-1 Error/Alarm Setting Commands

No.	Setting Items	Commands
[1]	Zoom	:DISPlay:RESult:ZOOM
		:DISPlay:RESult:ZOOM?
[2]	History Reset	:DISPlay:RESult:EALarm:HRESet
[3]	No label (Query for error alarm measurement data)	:CALCulate:DATA:EALarm?
[4]	No label (Query for monitoring item)	:CALCulate:DATA:MONitor?
[5]	No label (Query for Combination measurement data)	:CALCulate:CDATa:EALarm?
[6]	No label (Query for Combination monitoring item)	:CALCulate:CDATa:MONitor?

: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.

: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

Form	Format	Description
Form1 Integer type	"XXXXXXX"	For 0 to 9999999
	"X.XXXXEXX"	For 1.0000E07 to 9.9999E17
	"-------"	When no data corresponds to the query
Form2 Fraction type	"X.XXXXE-XX"	For 0.0001E–18 to 1.0000E00
	"-------"	When no data corresponds to the query
Form3 % type	"XXX.XXXXX"	For 0.0000 to 100.0000
	"-------"	When no data corresponds to the query
Form4 Frequency type	"XXXXXXXXX"	For 0 to MAX (Hz)
	"-------"	When no data corresponds to the query
Form5 Voltage type	"X.XXX, XX.XXX"	For –4.000 to + 4.000
	"-------, -------"	When no data corresponds to the query
Form6 Phase type	"XXXX, XXXX"	For –1000 to +1000, –10000 to +10000
	"-----, -------"	When no data corresponds to the query

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

Items		<result>	Form
Error Rate	Total	ER:TOTal	Form2
	INS	ER:INSertion	
	OMI	ER:OMISsion	
	Transition	ER:TRANsition	
	Non Transition	ER:NONTransition	
Error Count	Total	EC:TOTal	Form1
	INS	EC:INSertion	
	OMI	EC:OMISsion	
	Transition	EC:TRANsition	
	Non Transition	EC:NONTransition	
%EFI	Total	EFI:TOTal	Form3
EI	Total	EI:TOTal	Form1
Frequency(kHz)		FREQuency	Form4
Clock Count		CC:TOTal	Form1
Clock Loss		AINTerval:CLOSs	Form1
Sync Loss		AINTerval:PSLoss	Form1
Data Threshold		THReshold	Form5
Data Phase		PHASe	Form6

:CALCulate:DATA:MONitor? <item>

Parameter

<item> = <STRING PROGRAM DATA>

"BIT:TOTal"	Bit Error (Total Error)
"CLOSs"	Clock Loss
"PSLoss"	Pattern Sync Loss

Response

<string> = <STRING RESPONSE DATA>

Table 7.12.2.2-4 Response

Format	Description
"Occur"	When an alarm occurs
"Not Occur"	When no alarm occurs
"------"	When no data corresponds to the query

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 7.12.2.2-5 Response Format

Form	Format	Description
Form1 Integer type	"XXXXXXX"	For 0 to 9999999
	"X.XXXXEXX"	For 1.0000E07 to 9.9999E17
	"-------"	When no data corresponds to the query
Form2 Fraction type	"X.XXXXE-XX"	For 0.0001E–18 to 1.0000E00
	"-------"	When no data corresponds to the query
Form3 % type	"XXX.XXXXX"	For 0.0000 to 100.0000
	"-------"	When no data corresponds to the query
Form4 Frequency type	"XXXXXXXXX"	For 0 to MAX (Hz)
	"-------"	When no data corresponds to the query
Form5 Voltage type	"X.XXX, XX.XXX"	For –4.000 to +4.000
	"-------, -------"	When no data corresponds to the query
Form6 Phase type	"XXXX, XXXX"	For –1000 to +1000, –10000 to +10000
	"-----, -------"	When no data corresponds to the query

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

Items		<result>	Form
Error Rate	Total	ER:TOTal	Form2
	INS	ER:INSertion	
	OMI	ER:OMISsion	
Error Count	Total	EC:TOTal	Form1
	INS	EC:INSertion	
	OMI	EC:OMISsion	
%EFI	Total	EFI:TOTal	Form3
EI	Total	EI:TOTal	Form1
Frequency(kHz)		FREQuency	Form4
Clock Count		CC:TOTal	Form1
Clock Loss		AINTerval:CLOSs	Form1
Sync Loss		AINTerval:PSLoss	Form1
Data Threshold		THReshold	Form5
Data Phase		PHASe	Form6

: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

Format	Description
"Occur"	When an alarm occurs
"Not Occur"	When no alarm occurs
"------"	When no data corresponds to the query

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

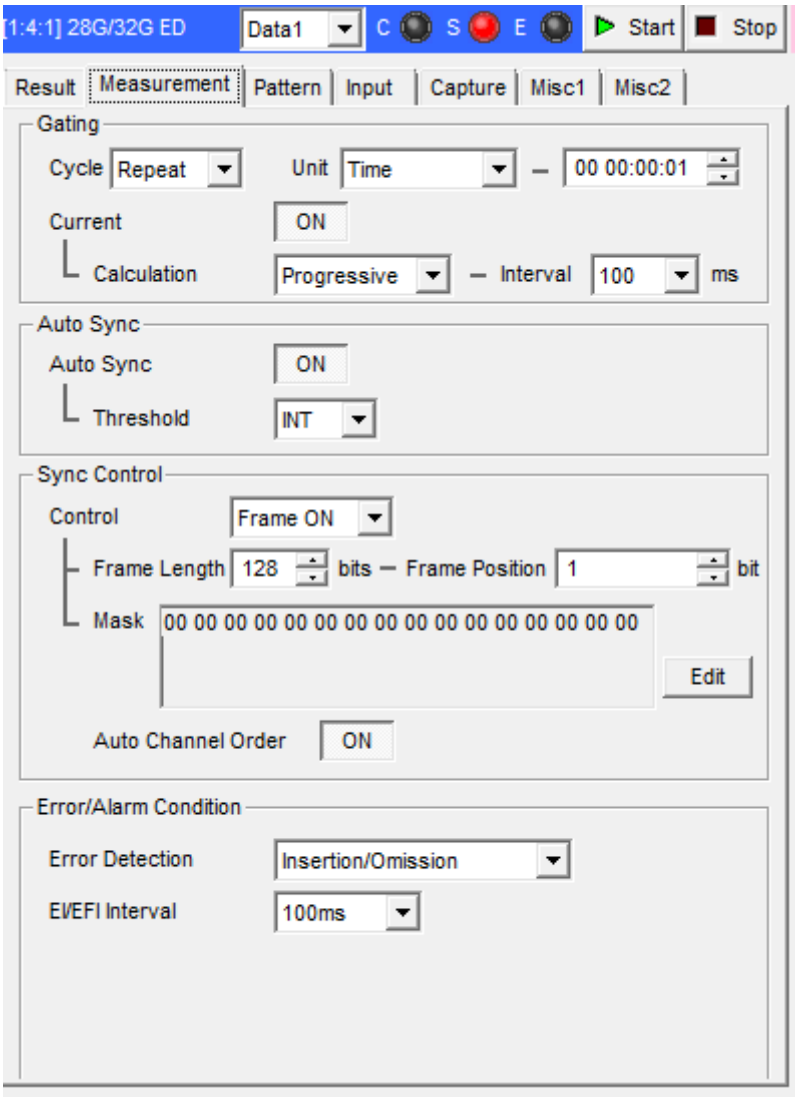


Figure 7.12.3-1 Measurement Tab

7.12.3.1 Gating setting commands

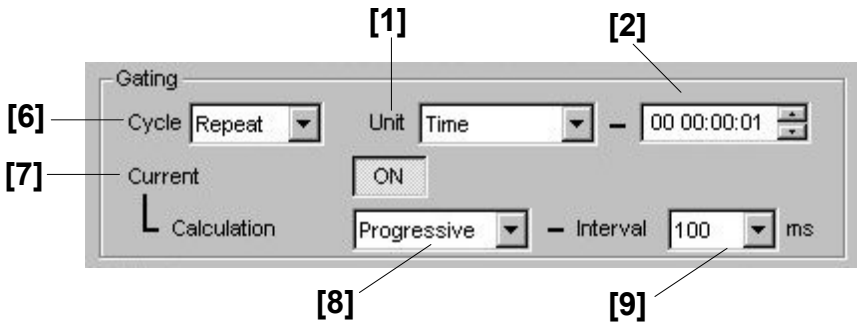


Figure 7.12.3.1-1 Gating Setting (Time)

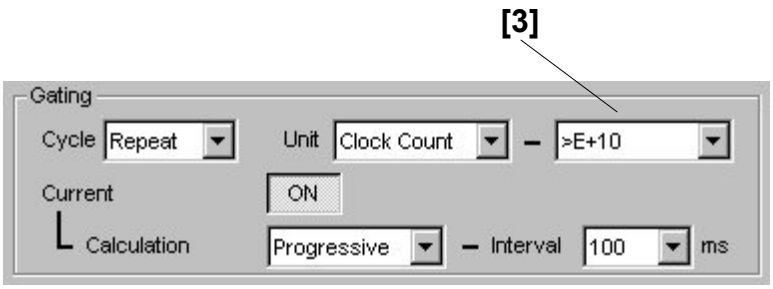


Figure 7.12.3.1-2 Gating Setting (Clock Count)

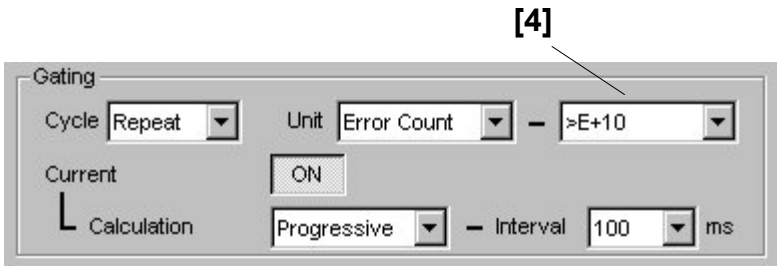


Figure 7.12.3.1-3 Gating Setting (Error Count)

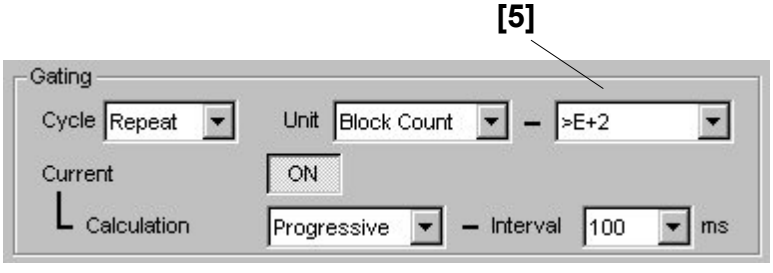


Figure 7.12.3.1-4 Gating Setting (Block Count)

Table 7.12.3.1-1 Gating Setting Commands

No.	Setting Items	Commands
[1]	Unit	:SENSe:MEASure:EALarm:UNIT
		:SENSe:MEASure:EALarm:UNIT?
[2]	No label (Measurement time)	:SENSe:MEASure:EALarm:PERiod
		:SENSe:MEASure:EALarm:PERiod?
[3]	No label (Clock count)	:SENSe:MEASure:EALarm:CLOCkcnt
		:SENSe:MEASure:EALarm:CLOCkcnt?
[4]	No label (Error count)	:SENSe:MEASure:EALarm:ERRorcnt
		:SENSe:MEASure:EALarm:ERRorcnt?
[5]	No label (Block count)	:SENSe:MEASure:EALarm:BLOCkcnt
		:SENSe:MEASure:EALarm:BLOCkcnt?
[6]	Cycle	:SENSe:MEASure:EALarm:MODE
		:SENSe:MEASure:EALarm:MODE?
[7]	Current	:DISPlay:RESult:EALarm:MODE
		:DISPlay:RESult:EALarm:MODE?
[8]	Calculation	:DISPlay:RESult:EALarm:CALCuration
		:DISPlay:RESult:EALarm:CALCuration?
[9]	Interval	:DISPlay:RESult:EALarm:INTerval
		:DISPlay:RESult:EALarm:INTerval?

:SENSe:MEASure:EALarm:UNIT <unit>

Parameter <unit> = <CHARACTER PROGRAM DATA>

TIME	Time
CLOCk	Clock Count
ERRor	Error Count
BLOCk	Block Count

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 <unit> = <CHARACTER RESPONSE DATA>

TIME, CLOC, ERR, BLOC

Function Queries the unit of measurement cycle.

Example

```
> :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	<count> = <CHARACTER PROGRAM DATA>	
	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
	E_15	>1 E + 15
	E_16	>1 E + 16
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	<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 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	<count> = <CHARACTER PROGRAM DATA>	
	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
	E_15	>1 E + 15
	E_16	>1 E + 16
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	`> :SENSe:MEASure:EALarm:ERRorcnt?` `< E_4`
Compatibility	Compatible with MU181040A/B.

:SENSe:MEASure:EALarm:BLOCkcnt <count>

Parameter	<count> = <CHARACTER PROGRAM DATA>	
	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	<count> = <CHARACTER RESPONSE DATA>
	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	<mode> = <CHARACTER PROGRAM DATA>	
	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:	

```
> :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	Turns off the actual value update of the measurement result data display.
1 or ON	Turns on the actual value update of the measurement result data display.

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>

Parameter	<mode> = <CHARACTER PROGRAM DATA>
	PROGressive Cumulative processing.
	IMMediate Immediate processing.
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	<mode> = <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>

Parameter	<interval> = <CHARACTER PROGRAM DATA>
	I100 Updates display in 100-ms period.
	I200 Updates display in 200-ms period.
	I500 Updates display in 500-ms period.
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	<interval> = <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

Figure 7.12.3.2-1 Auto Sync Setting

Table 7.12.3.2-1 Auto Sync Setting Commands

No.	Setting Items	Commands
[1]	Auto Sync	:SENSe:PATTern:SYNC:ASYNc
		:SENSe:PATTern:SYNC:ASYNc?
[2]	Threshold	:SENSe:PATTern:SYNC:THReshold
		:SENSe:PATTern:SYNC:THReshold?

: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.

:SENSe:PATTern:SYNC:THReshold <thre>

Parameter	<thre> = <CHARACTER PROGRAM DATA>
	INT Internal
	E_2 E–2
	E_3 E–3
	E_4 E–4
	E_5 E–5
	E_6 E–6
	E_7 E–7
	E_8 E–8
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?

Response	<thre> = <CHARACTER RESPONSE DATA> INT, E_2, E_3, E_4, E_5, E_6, E_7, E_8
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

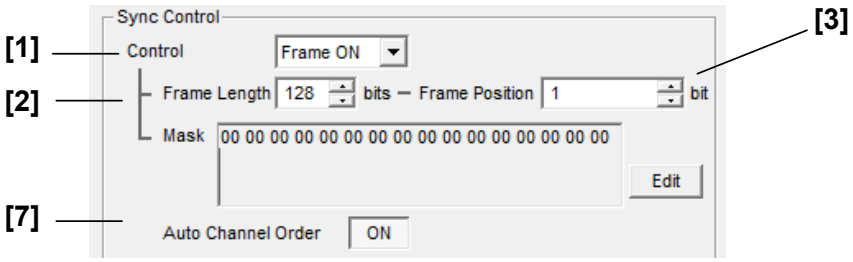


Figure 7.12.3.3-1 Sync Control Setting

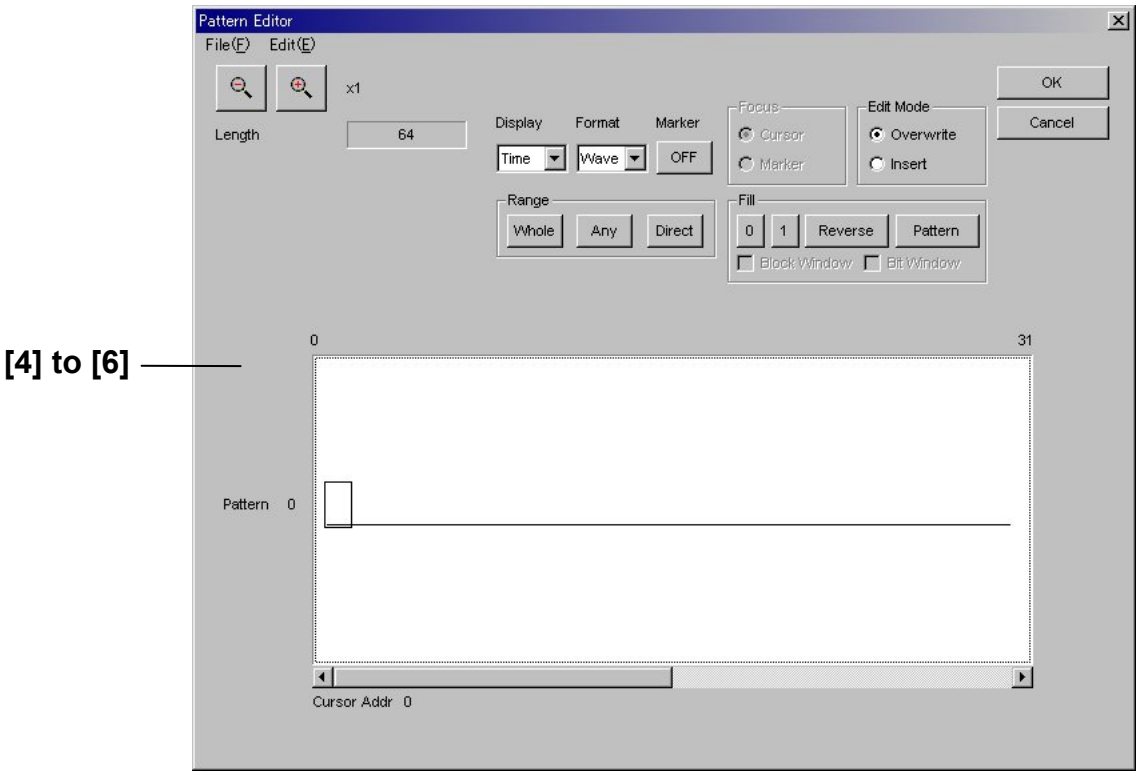


Figure 7.12.3.3-2 Pattern Editor Dialog Box

Table 7.12.3.3-1 Sync Control Setting Commands

No.	Setting Items	Commands
[1]	Control	:SENSe:PATTern:SYNC:PSMode
		:SENSe:PATTern:SYNC:PSMode?
[2]	Frame Length	:SENSe:PATTern:SYNC:FLENgth
		:SENSe:PATTern:SYNC:FLENgth?
[3]	Frame Position	:SENSe:PATTern:SYNC:FPOSition
		:SENSe:PATTern:SYNC:FPOSition?
[4]	Mask (Mask pattern setting)	:SENSe:PATTern:SYNC:DATA:WHOLe
		:SENSe:PATTern:SYNC:DATA:WHOLe?
[5]	No label (Data inversion)	:SENSe:PATTern:SYNC:DREVerse:ADDRess
[6]	No label (All 0 or 1)	:SENSe:PATTern:SYNC:PROGram:FILL
[7]	Auto Channel Order	:SENSe:PATTern:SYNC:ACORder
		:SENSe:PATTern:SYNC:ACORder?

: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	<length> = <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: `> :SENSe:PATTern:SYNC:FLENgth 4`
Compatibility	Partially compatible with MU181040A/B.

:SENSe:PATTern:SYNC:FLENgth?

Response	<length> = <NR1 NUMERIC RESPONSE DATA>
Function	Queries the unique pattern length for the synchronization control with Frame detection ON.
Example	`> :SENSe:PATTern:SYNC:FLENgth?` `< 4`
Compatibility	Partially compatible with MU181040A/B.

:SENSe:PATTern:SYNC:FPOSition <length>

Parameter	<length> = <DECIMAL NUMERIC PROGRAM DATA> 1 to 268435453 1 to 268,435,453 bits/1 bit Step Allowable parameter setting: Pattern length − Frame length + 1 (bits) At 2ch Combination Setting, 1 to 536870905 1 to 536,870,905 bits/2 bit Step At 4ch Combination Setting, 1 to 1073741809 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: `> :SENSe:PATTern:SYNC:FPOSition 65`
Compatibility	Partially compatible with MU181040A/B.

: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.
	Up to 64 bits hexadecimal data is returned.	
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>

Parameter
<start> = <NON-DECIMAL PROGRAM DATA>
#H0 to #H3F 0 to 3F bits/1 bit Step (Specify in hexadecimal.)
<end> = <NON-DECIMAL PROGRAM DATA>
#H0 to #H3F 0 to 3F bits/1 bit Step (Specify in hexadecimal.)

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>

Parameter
<start> = <NON-DECIMAL PROGRAM DATA>
#H0 to #H3F 0 to 3F bits/1 bit Step (Specify in hexadecimal.)
<end> = <NON-DECIMAL PROGRAM DATA>
#H0 to #H3F 0 to 3F bits/1 bit Step (Specify in hexadecimal.)
<data> = <DECIMAL NUMERIC PROGRAM DATA>
0 Fills the specified range with 0s.
1 Fills the specified range with 1s.

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.

: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	<numeric>=<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

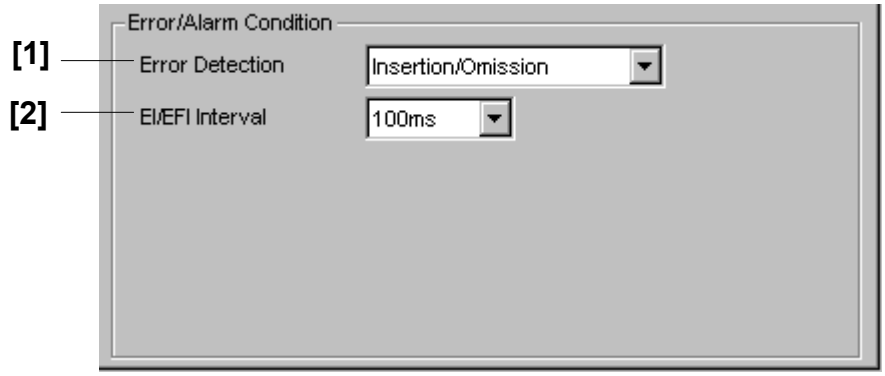


Figure 7.12.3.4-1 Error/Alarm Condition Setting

Table 7.12.3.4-1 Error/Alarm Condition Setting Commands

No.	Setting Items	Commands
[1]	Error Detection	:CALCulate:EALarm:ERRor:EDETect
		:CALCulate:EALarm:ERRor:EDETect?
[2]	EI/EFI Interval	:CALCulate:EALarm:ERRor:INTerval
		:CALCulate:EALarm:ERRor:INTerval?

: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.

: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

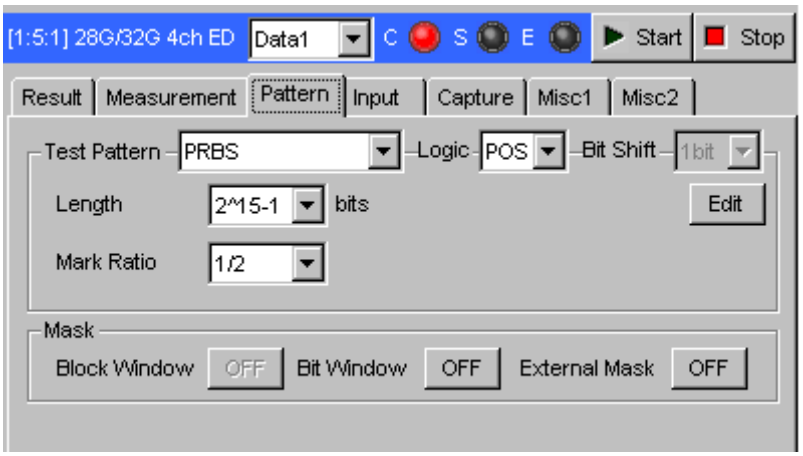


Figure 7.12.4-1 Pattern Tab

7.12.4.1 Test Pattern setting commands

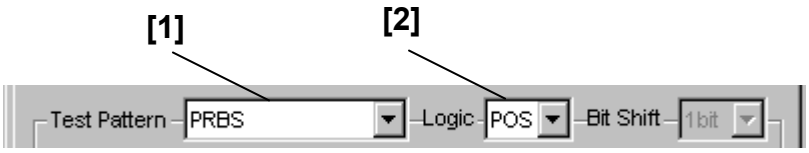


Figure 7.12.4.1-1 Test Pattern Area

Table 7.12.4.1-1 Test Pattern Setting Commands

No.	Setting Items	Commands
[1]	Test Pattern	:SENSe:PATTern:TYPE
		:SENSe:PATTern:TYPE?
[2]	Logic	:SENSe:PATTern:LOGic
		:SENSe:PATTern:LOGic?

: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

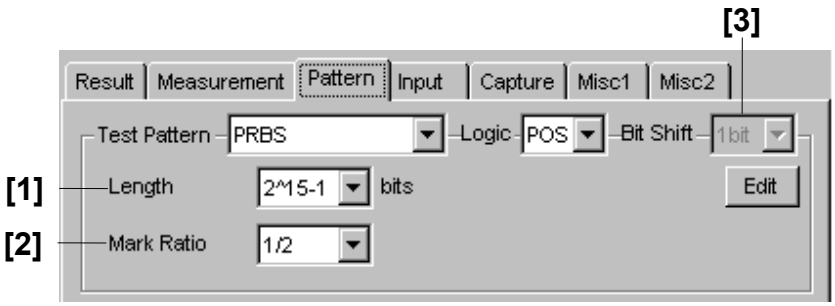


Figure 7.12.4.2-1 Test Pattern Setting (When PRBS Is Selected)

Table 7.12.4.2-1 PRBS Pattern Setting Commands

No.	Setting Items	Commands
[1]	Length	:SENSe:PATTern:PRBS:LENGth
		:SENSe:PATTern:PRBS:LENGth?
[2]	Mark Ratio	:SENSe:PATTern:PRBS:MRATio
		:SENSe:PATTern:PRBS:MRATio?
[3]	Bit Shift	:SENSe:PATTern:PRBS:BSHift?

:SENSe:PATTern: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 (2^n-1 (n = 7, 9, 10, 11, 15, 20, 23, or 31)) for PRBS pattern reception.

Example

To set the number of stages for PRBS pattern reception to 2^7-1:

```
> :SENSe:PATTern:PRBS:LENGth 7
```

Compatibility

Compatible with MU181040A/B.

: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

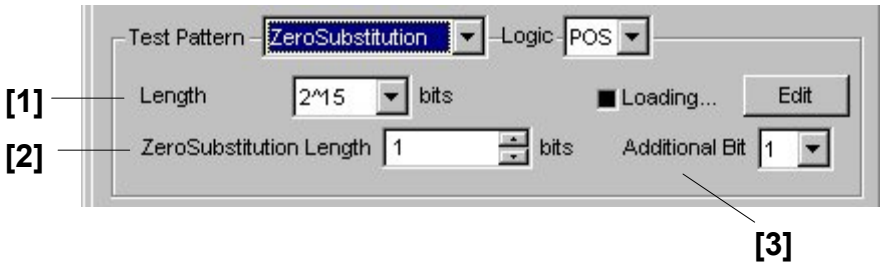


Figure 7.12.4.3-1 Test Pattern Setting (When Zero Substitution Is Selected)

Table 7.12.4.3-1 Zero Substitution Pattern Setting Commands

No.	Setting Items	Commands
[1]	Length	:SENSe:PATTern:ZSUBstitute:LENGth
		:SENSe:PATTern:ZSUBstitute:LENGth?
[2]	Zero-Substitution Length	:SENSe:PATTern:ZSUBstitute:ZLENgth
		:SENSe:PATTern:ZSUBstitute:ZLENgth?
[3]	Additional Bit	:SENSe:PATTern:ZSUBstitute:ADDBit
		:SENSe:PATTern:ZSUBstitute:ADDBit?

:SENSe: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 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.

: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. Available parameters vary depending on the setting conditions.
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^nth bit is set to "0". 1 The 2^nth 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^nth bit is set to "0". 1 The 2^nth 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)

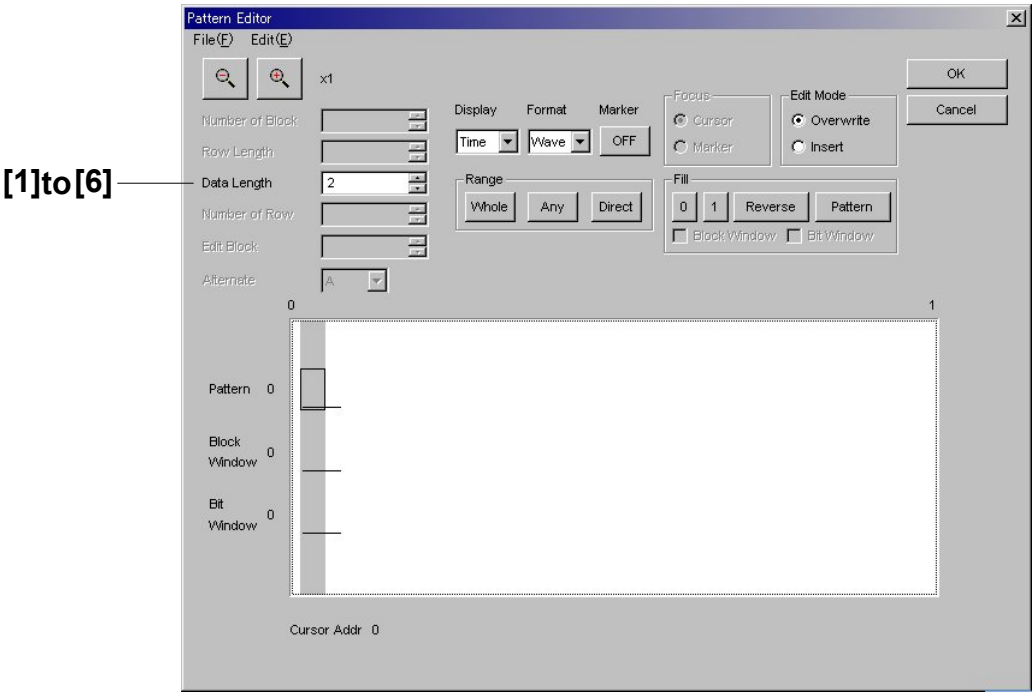


Figure 7.12.4.4-2 Data Pattern Setting (Pattern Editor Dialog Box)

Table 7.12.4.4-1 Data Pattern Setting Commands

No.	Setting Items	Commands
[1]	Data Length	:SENSe:PATTern:DATA:LENGth
		:SENSe:PATTern:DATA:LENGth?
[2]	No label (Pattern setting)	:SENSe:PATTern:DATA:WHOLe
		:SENSe:PATTern:DATA:WHOLe?
[3]	No label (Pattern setting)	:SENSe:PATTern:BDATa:WHOLe
		:SENSe:PATTern:BDATa:WHOLe?
[4]	No label (Data inversion)	:SENSe:PATTern:DREVerse:ADDRess
[5]	No label (Data inversion)	:SENSe:PATTern:DREVerse:DELTa
[6]	No label (All 0 or 1)	:SENSe:PATTern:DATA:FILL

:SENSe:PATTern:DATA:LENGth <numeric>

Parameter
<numeric> = <DECIMAL NUMERIC PROGRAM DATA>
2 to 268435456 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:

```
> :SENSe:PATTern:DATA:LENGth 2
```

Compatibility
Partially compatible with MU181040A/B.

: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
Partially compatible with MU181040A/B.

:SENSe:PATTern:DATA:WHOLe <start>,<end>,<data>

Parameter
<start> = <NON-DECIMAL PROGRAM DATA>
#H0 to #H7FFFFFF 0 to 7FFFFFF bits/1 bit Step
(Specify in hexadecimal.)
<end> = <NON-DECIMAL PROGRAM DATA>
#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.

<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 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:PATTern: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 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 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 FFFFFFF bits/1 bit Step (Specify in hexadecimal.)

<end> = <NON-DECIMAL PROGRAM DATA>

#H0 to #HFFFFFFFF — 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.

<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 FFFFFFF bits addresses during Data pattern reception:

```
> :SENSe:PATTern:BDATa:WHOLe #H0,#HFFFFFFF,#10011
```

Compatibility: Partially compatible with MU181040A/B.

:SENSe:PATTern:BDATa:WHOLe? <start> [,<size>]

Parameter:

<start> = <NON-DECIMAL PROGRAM DATA>

#H0 to #HFFFFFFF	0 to FFFFFFF 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:ADDRess <start>,<end>

Parameter

<start> = <NON-DECIMAL PROGRAM DATA>

#H0 to #HFFFFFFFF 0 to FFFFFFF bits/1 bit Step
(Specify in hexadecimal.)

<end> = <NON-DECIMAL PROGRAM DATA>

#H0 to #HFFFFFFFF 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.

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 7FFFFFF bit address of the program at Data pattern reception:

```
> :SENSe:PATTern:DREVerse:ADDRess #H0,#H7FFFFFF
```

Compatibility

Partially compatible with MU181040A/B.

:SENSe:PATTern:DREVerse:DELTa <start>,<delta>

Parameter

<start> = <NON-DECIMAL PROGRAM DATA>

#H0 to #HFFFFFFFF 0 to FFFFFFF 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:PATTern:DREVerse:DELTa #H0,1
```

Compatibility

Partially compatible with MU181040A/B.

: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

$$\text{Max} = \frac{\text{Pattern Length}}{128} \text{ (rounding up fractions)}$$

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

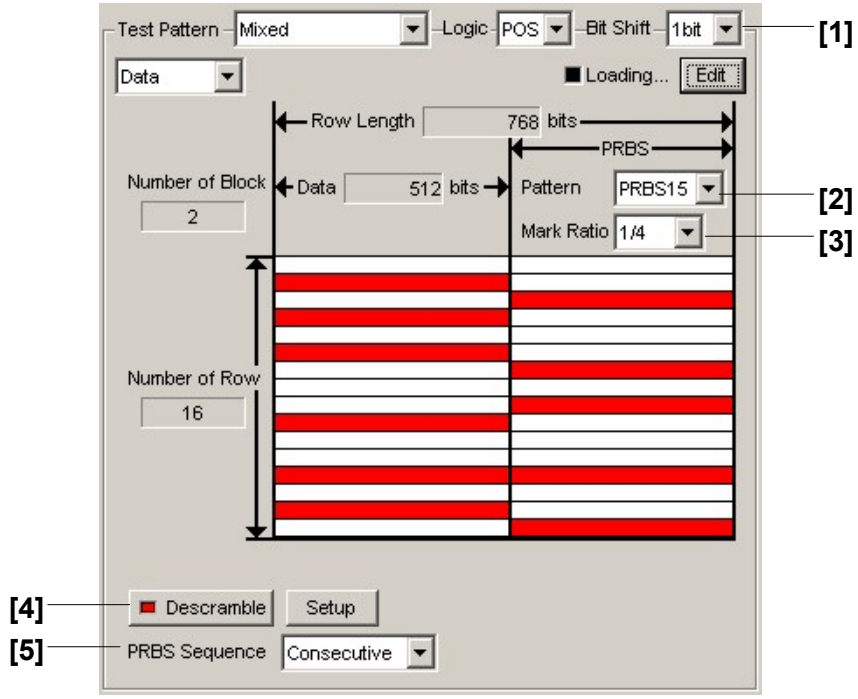


Figure 7.12.4.5-1 Test Pattern Setting (When Mixed Is Selected)

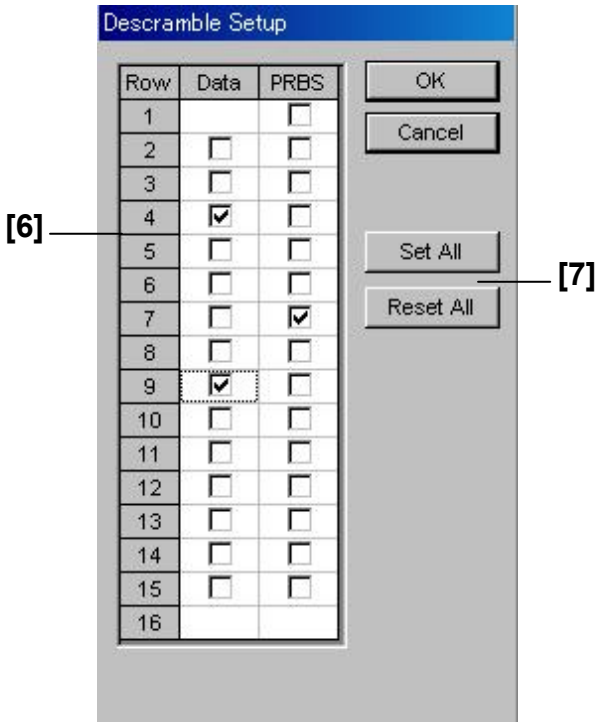


Figure 7.12.4.5-2 Descramble Setup Dialog Box

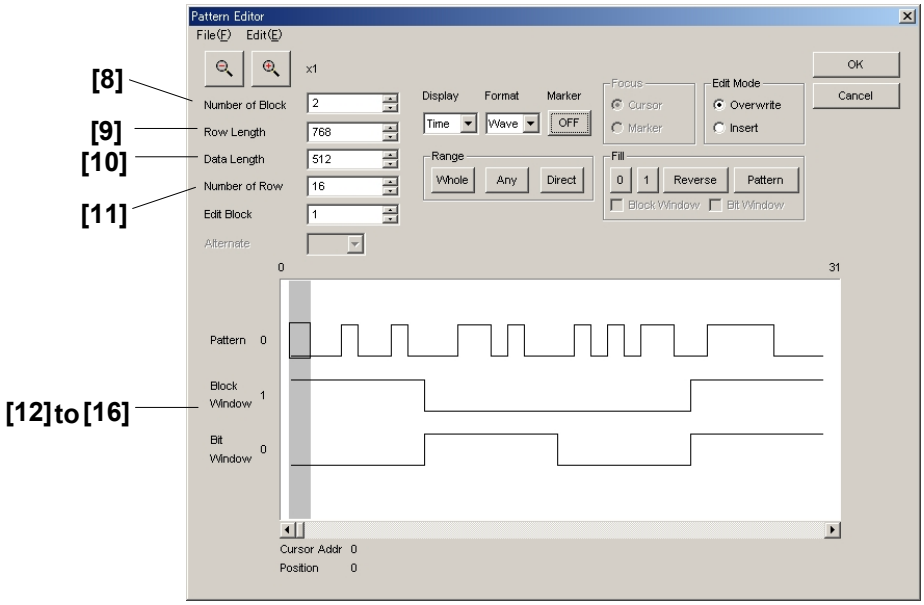


Figure 7.12.4.5-3 Mixed Data Pattern Setting (Pattern Editor Dialog Box)

Table 7.12.4.5-1 Mixed Data pattern Setting Commands

No.	Setting Items	Commands
[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?

Table 7.12.4.5-1 Mixed Data pattern Setting Commands (Cont'd)

No.	Setting Items	Commands
[13]	No label (Pattern setting)	:SENSe:PATTern:MIXData:BDATa:WHOLe
		:SENSe:PATTern:MIXData:BDATa:WHOLe?
[14]	No label (Data inversion)	:SENSe:PATTern:MIXData:DREVerse:ADDRess
[15]	No label (Data inversion)	:SENSe:PATTern:MIXData:DREVerse:DELTa
[16]	No label (Data inversion)	:SENSe:PATTern:MIXData:DATA:FILL

: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^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 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 Compatible with MU181040A/B.

: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 1/2INVT
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	<numeric> = <NR1 NUMERIC RESPONSE DATA> 0 Descramble OFF 1 Descramble ON
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	<sequence> = <CHARACTER PROGRAM DATA> RESTart Inconsecutive PRBS pattern CONSecutive Consecutive PRBS pattern
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	<sequence> = <CHARACTER RESPONSE DATA> REST, CONS
Function	Queries the PRBS pattern sequence during Mixed Data pattern reception.
Example	`> :SENSe:PATTern:MIXData:PRBS:SEQuence?` `< REST`
Compatibility	Compatible with MU181040A/B.

: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
	Note, however, that Data of 1 Row is always Descramble OFF.
	<prbs> = <BOOLEAN PROGRAM DATA>
	0 or OFF Descramble OFF
	1 or ON Descramble ON
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 1 to 511 Block/1 Step

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 1 to 511 Block

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

: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? < 1536
Compatibility	Partially compatible with MU181040A/B.

:SENSe:PATTern:MIXData:DATA:LENGth <numeric>

Parameter	<numeric> = <DECIMAL NUMERIC PROGRAM DATA> 1024 to 268435456 1,024 to 268,435,456 bits/1 bit Step ***Note:*** At 2ch Combination, the setting range and Step are doubled. 2048 to 536870912 2,048 to 536,870,912 bits/2 bit Step At 4ch Combination, the setting range and Step are quadrupled. 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 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?

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? < 2048
Compatibility	Partially compatible with MU181040A/B.

: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 #HFFFFFFFF 0 to FFFFFFF bits/1 bit Step
(Specify in hexadecimal.)

<end> = <NON-DECIMAL PROGRAM DATA>

#H0 to #HFFFFFFFF 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> = <STRINGP ROGRAM 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 7FFFFFF bits of 3 blocks in hexadecimal during Mixed Data pattern reception:

```
> :SENSe:PATTern:MIXData:DATA:WHOLe 3,#H0,#H7FFFFFF,
"H001"
```

Compatibility

Partially compatible with MU181040A/B.

:SENSe:PATTern: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 #HFFFFFFFF 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:PATTern:MIXData:DATA:WHOLe? 3,#H0
< "H001"
```

Compatibility

Partially compatible with MU181040A/B.

:SENSe:PATTern:MIXData:BDATa:WHOLe <block>,<start>,<end>,<bdata>

Parameter

<block> = <DECIMAL NUMERIC PROGRAM DATA>

1 to 511 1 to 511 Block/1 Step

<start> = <NON-DECIMAL PROGRAM DATA>

#H0 to #HFFFFFFFF 0 to FFFFFFF bits/1 bit Step
(Specify in hexadecimal.)

<end> = <NON-DECIMAL PROGRAM DATA>

#H0 to #HFFFFFFFF 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.

<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 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,#H7FFFFF,#10011
```

Compatibility

Partially compatible with MU181040A/B.

:SENSe:PATTern:MIXData:BDATa:WHOLe? <block>,<start>[,<size>]

Parameter

<block> = <DECIMAL NUMERIC PROGRAM DATA>

1 to 511 1 to 511 Block/1 Step

<start> = <NON-DECIMAL PROGRAM DATA>

#H0 to #HFFFFFFFF 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.

[<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: 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:

```
> :SENSe:PATTern:MIXData:BDATa:WHOLe? 3,#H0
< #10011
```

Compatibility

Partially compatible with MU181040A/B.

: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-DECIMALPROGRAMDATA>

#H0 to #HFFFFFFFF 0 to FFFFFFF bits/1 bit Step
(Specify in hexadecimal.)

<end> = <NON-DECIMAL PROGRAM DATA>

#H0 to #HFFFFFFFF 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.

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,#H7FFFFFF
```

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 #HFFFFFFFF 0 to FFFFFFF 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 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

<block> = <DECIMAL NUMERIC PROGRAM DATA>

1 to 511 1 to 511 Block/1 Step

<range> = <CHARACTER PROGRAM DATA>

PAGE Specifies pages (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) page/1 page Step

$$\text{Max} = \frac{\text{Pattern Length}}{128} \text{ (rounding up fractions)}$$

Specify "0" when <range> is set to ALL.

<data> = <DECIMAL NUMERIC PROGRAM DATA>

0 Fills "0" in specified range.

1 Fills "1" in specified range.

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

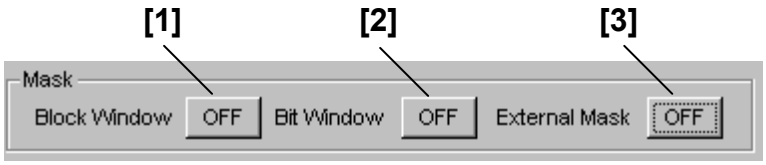


Figure 7.12.4.6-1 Mask ON/OFF Setting

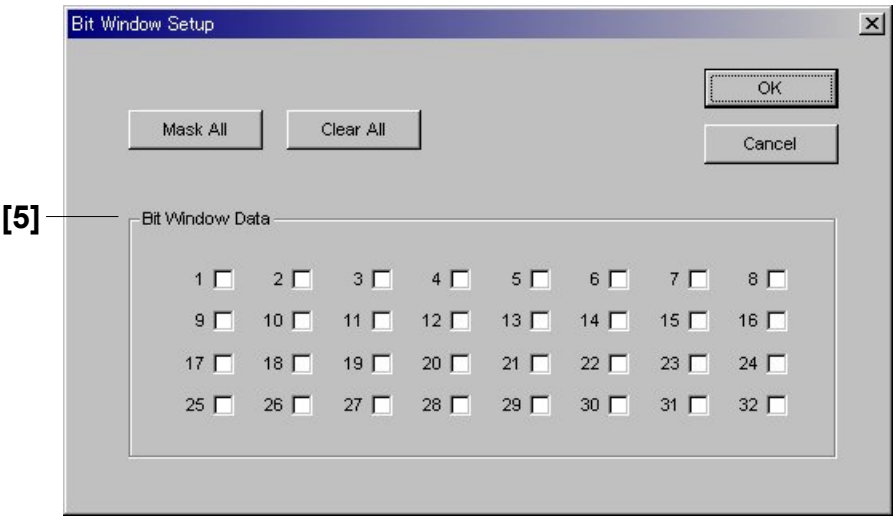


Figure 7.12.4.6-2 Bit Window Setup Dialog Box (For PRBS Pattern)

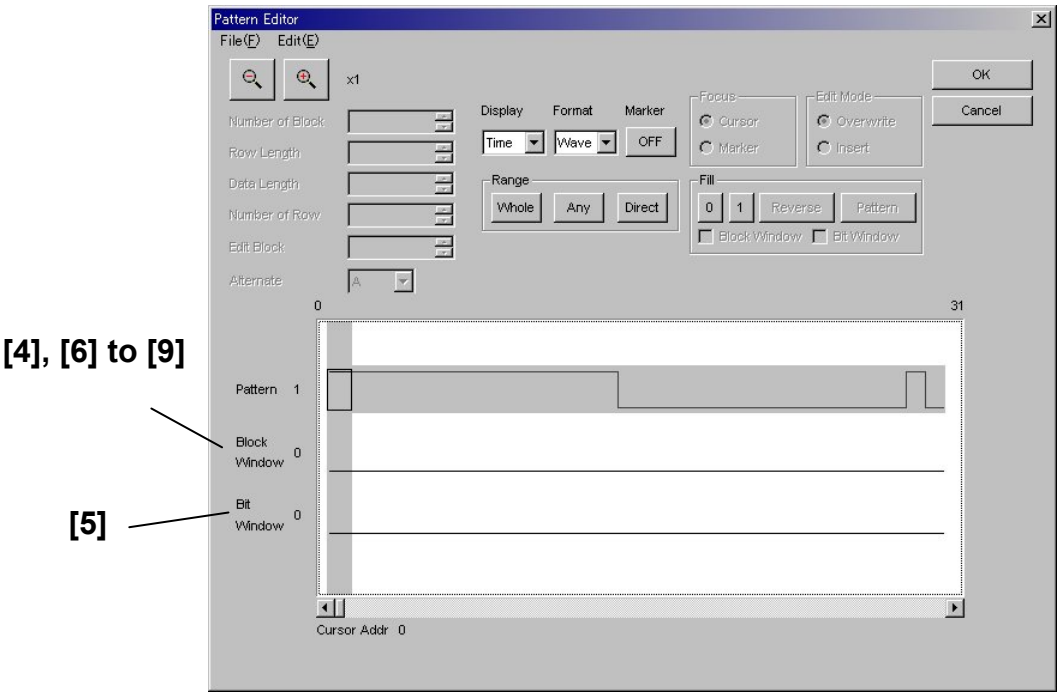


Figure 7.12.4.6-3 Bit Window and Block Window Settings in the Pattern Editor Dialog Box (For Zero Substitution, Data Patterns)

Table 7.12.4.6-1 Mask Setting Commands

No.	Setting Items	Commands
[1]	Block Window ON/OFF	:SENSe:PATTern:MASK:BLOCk
		:SENSe:PATTern:MASK:BLOCk?
[2]	Bit Window ON/OFF	:SENSe:PATTern:MASK:BIT
		:SENSe:PATTern:MASK:BIT?
[3]	External Mask ON/OFF	:SENSe:PATTern:MASK:EXTernal
		:SENSe:PATTern:MASK:EXTernal?
[4]	Block Window (Zero-Substitution, Data)	:SENSe:PATTern:MASK:BLKWindow
		:SENSe:PATTern:MASK:BLKWindow?
[5]	Bit Window	:SENSe:PATTern:MASK:BTWindow
		:SENSe:PATTern:MASK:BTWindow?
[6]	Block Window (Setting pattern)	:SENSe:PATTern:MASK:BDATa:WHOLe
		:SENSe:PATTern:MASK:BDATa:WHOLe?
[7]	Block Window (Inverting data)	:SENSe:PATTern:MASK:DREVerse:ADDRess
[8]	Block Window (Inverting data)	:SENSe:PATTern:MASK:DREVerse:DELTa
[9]	Block Window (All 0 or 1)	:SENSe:PATTern:MASK:DATA:FILL

:SENSe:PATTern: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:PATTern:MASK:BLOCk ON
```

Compatibility Compatible with MU181040A/B.

:SENSe:PATTern: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

```
> :SENSe:PATTern:MASK:BLOCk?
< 1
```

Compatibility Compatible with MU181040A/B.

: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.

Pattern Length setting			Block Window step
2*N	to	2,097,152*N bits	1*N bits
2,097,153*N	to	4,194,304*N bits	2*N bits
4,194,305*N	to	8,388,608*N bits	4*N bits
8,388,609*N	to	16,777,216*N bits	8*N bits
16,777,217*N	to	33,554,432*N bits	16*N bits
33,554,433*N	to	67,108,864*N bits	32*N bits
67,108,864*N	to	134,217,728*N bits	64*N bits
134,217,729*N	to	268,435,456*N bits	128*N bits

In addition, relation between pattern length and step is increased by N times according to the Combination setting.

N = 1, 2, 4

Independent: N = 1

2ch Combination: N = 2

4ch Combination: N = 4

:SENSe:PATTern:MASK:BLKWindow? <start>

Parameter	<start> = <NON-DECIMAL PROGRAM DATA>	
	#H0 to #H1FFFFFF	0 to 1FFFFF bits/1 bit Step (Specify in hexadecimal.)
Response	<data> = <STRING RESPONSE DATA>	
	"H***…*"	Returns the pattern data in hexadecimal format. 4 to 400 characters (pattern data of 400 × 4 bits)
Function	Queries the mask pattern data at the time of a Block Window setup by 400 × 4 bits from the <start>address.	

Example	To query the mask pattern data at the time of a Block Window setup by 400 × 4 bits from the 0 bit address:
	`> :SENSe:PATTern:MASK:BLKWindow? #H0`
	`< "H001"`	

Compatibility	Incompatible with existing models MU181040A/B.	

: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 Mask pattern data of Bit Window setting.	
Example	To set the mask pattern data of Bit Window setting in hexadecimal: `> :SENSe:PATTern:MASK:BTWindow "H001"`	
Compatibility	Incompatible with existing models MU181040A/B.	

:SENSe:PATTern:MASK:BTWindow?

Response	<data> = <STRING RESPONSE DATA>	
	"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	`> :SENSe:PATTern:MASK:BTWindow?` `< "H00100100"`	
Compatibility	Incompatible with existing models MU181040A/B.	

:SENSe:PATTern:MASK:BDATa:WHOLe <start>,<end>,<bdata>

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.)

<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 byte max.

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 7FFFF when **Block Window** is **ON**:

```
> :SENSe:PATTern:MASK:BDATa:WHOLe #H0,#H7FFFF,#11A
```

Compatibility

Incompatible with existing models MU181040A/B.

: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>

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) page/1page step

$$\text{Max} = \frac{\text{Pattern Length}}{128} \text{ (round up)}$$

Specify “0” when <range> is ALL.
<data> = <DECIMAL NUMERIC PROGRAM DATA>
0 Fills the specified range with 0s.
1 Fills the specified range with 1s.

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

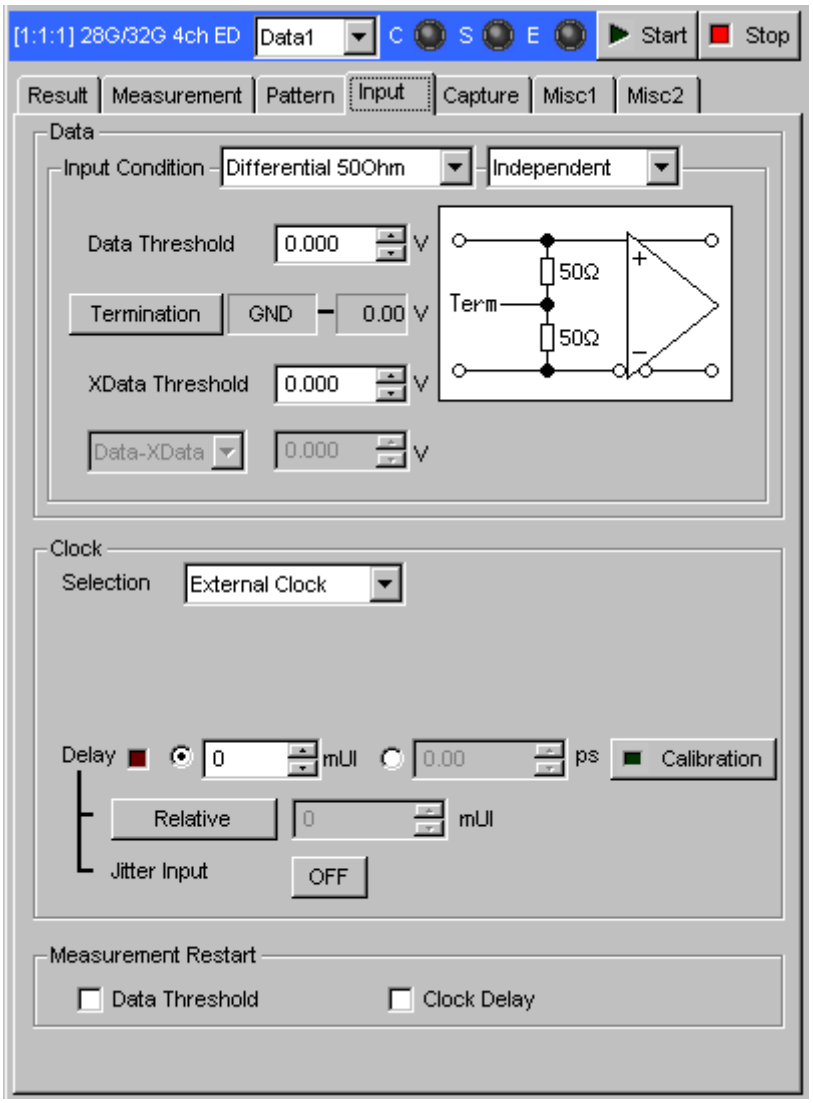


Figure 7.12.5-1 Input Tab

7.12.5.1 Data Input setting commands

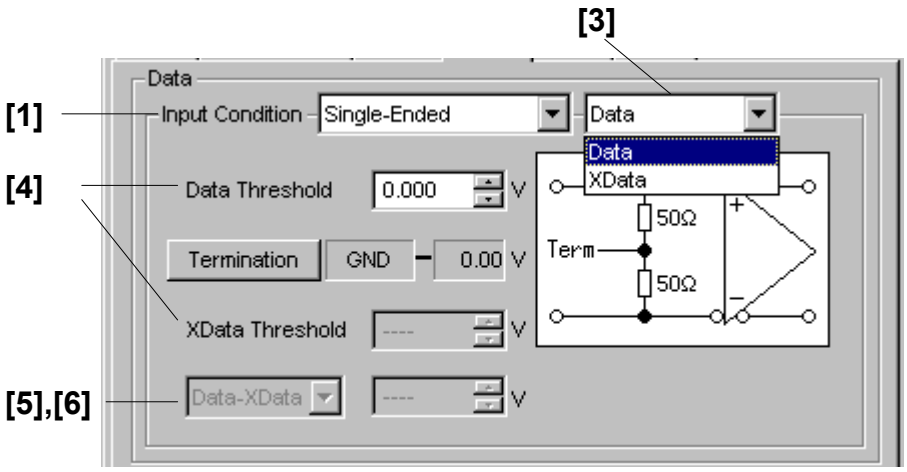


Figure 7.12.5.1-1 Data Setting (When Single-Ended Is Selected)

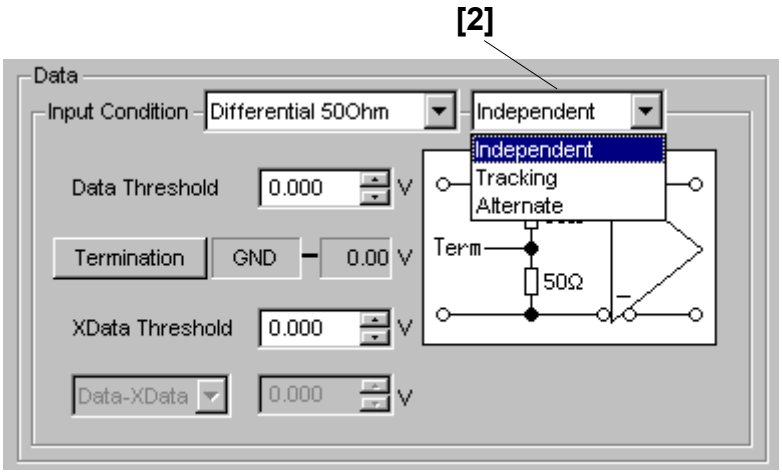


Figure 7.12.5.1-2 Data Setting (When Differential 50 Ohm Is Selected)

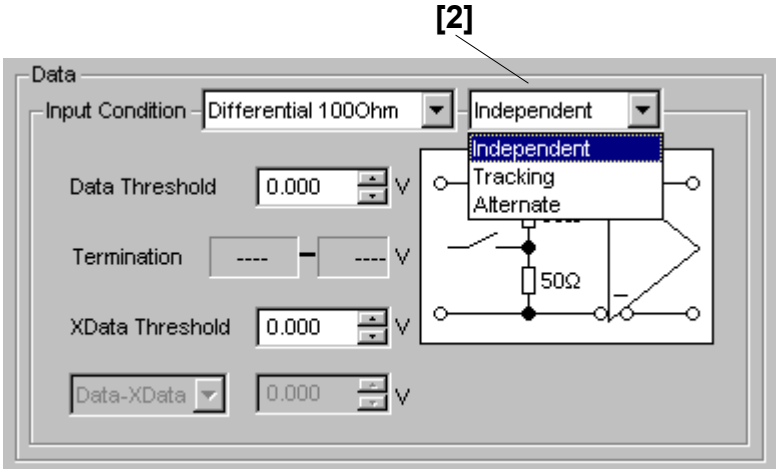


Figure 7.12.5.1-3 Data Setting (When Differential 100 Ohm Is Selected)

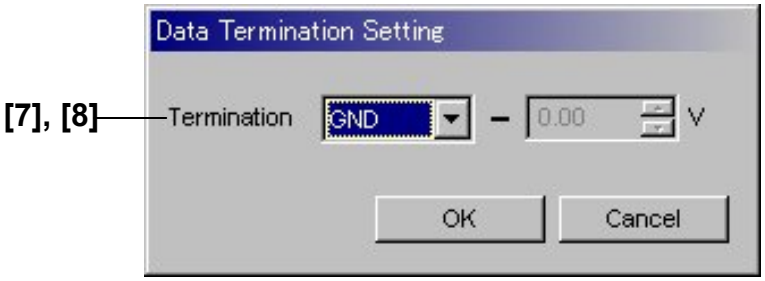


Figure 7.12.5.1-4 Data Termination Setting Dialog Box

Table 7.12.5.1-1 Data Input Setting Commands

No.	Setting Items	Commands
[1]	Input Condition	:INPut:DATA:INTerface
		:INPut:DATA:INTerface?
[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?
[4]	Data Threshold XData Threshold	:INPut:DATA: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?
[7]	Termination	:INPut:DATA:TERMination
		:INPut:DATA:TERMination?
[8]	No label (Termination level setting)	:INPut:DATA:TLEVel
		:INPut:DATA:TLEVel?

:INPut:DATA:INTerface <interface>

Parameter	<interface> = <CHARACTER PROGRAM DATA>	
	SINGle	Single Ended
	DIF50ohm	Differential 50 Ω
	DIF100ohm	Differential 100 Ω
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?

Response	<interface> = <CHARACTER RESPONSE DATA>
	SING, DIF50, DIF100
Function	Queries the data input interface.
Example	`> :INPut:DATA:INTerface?`
	`< SING`
Compatibility	Compatible with MU181040A/B.

:INPut:DATA:DIFFerential <item>

Parameter	<item> = <CHARACTER PROGRAM DATA>	
	INDependent	Data/Xdata independent setting
	TRACking	Data/Xdata common setting
	ALTernate	Data/Xdata differential setting
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?

Response	<item> = <CHARACTER RESPONSE DATA>
	IND, TRAC, ALT
Function	Queries the differential type when the data input interface is set to Differential.
Example	`> :INPut:DATA:DIFFerential?`
	`< IND`
Compatibility	Compatible with MU181040A/B.

: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.
	Be sure to input the selected port when Single is selected.
	<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.

:INPut:DATA:DIFFerential:BASis <basis>

Parameter	<basis> = <CHARACTER PROGRAM DATA>
	DATA Sets the differential reference signal to Data.
	XDATa Sets the differential reference signal to 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 DATA: `> :INPut:DATA:DIFFerential:BASis DATA`
Compatibility	Compatible with MU181040A/B.

:INPut:DATA:DIFFerential:BASis?

Response	<basis> = <CHARACTER RESPONSE DATA> DATA, XDAT
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>
	–3.000 to 3.000 –3.000 to 3.000 V/0.001 V Step
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

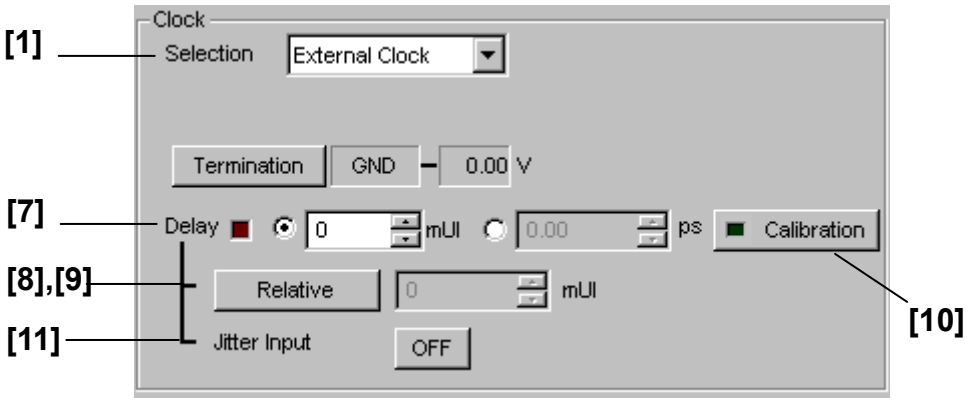


Figure 7.12.5.2-1 Clock Setting (When External Clock Is Selected)

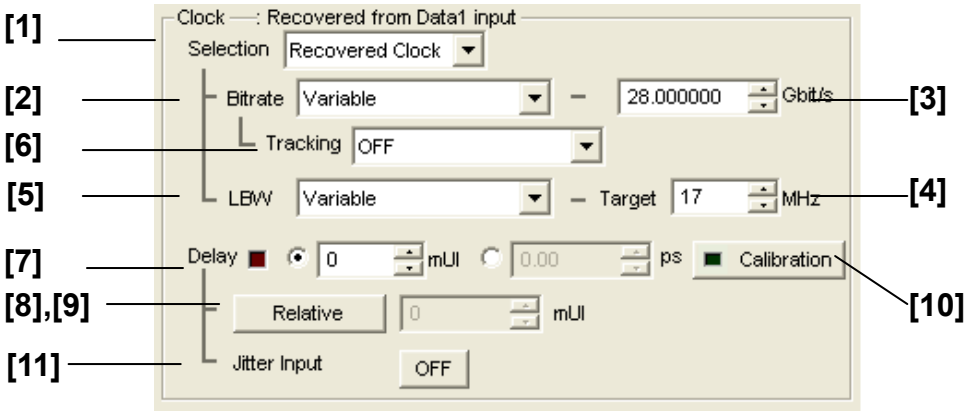


Figure 7.12.5.2-2 Clock Setting (Option 22 When Recovered Clock Is Selected)

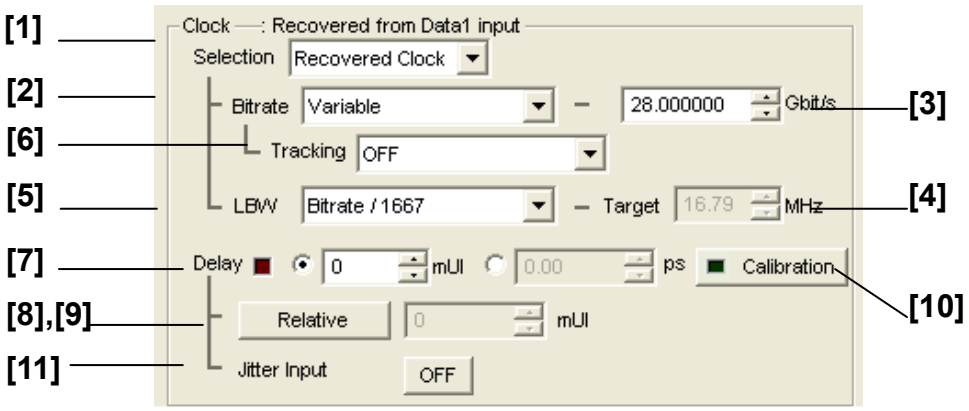


Figure 7.12.5.2-3 Clock Setting (Option 23 When Recovered Clock Is Selected)

Table 7.12.5.2-1 Clock Input Setting Commands

No.	Setting Items	Commands
[1]	Selection	:INPut:CLOCk:SELection
		:INPut:CLOCk:SELection?
[2]	Setting of preset frequency for Clock Recovery Preset	:INPut:CLOCk:RECovery
		:INPut:CLOCk:RECovery?
[3]	Setting of frequency for Clock Recovery Operation Bitrate	:INPut:CLOCk:CRFReq
		:INPut:CLOCk:CRFReq?
[4]	Target LBW	:INPut:CLOCk:LBANdwidth
		:INPut:CLOCk:LBANdwidth?
[5]	Selection of a loop band setting	:INPut:CLOCk:LBW
		:INPut:CLOCk:LBW?
[6]	PPG Bitrate Tracking	:INPut:CLOCk:PTRacking
		:INPut:CLOCk:PTRacking?
[7]	Delay	:INPut:CLOCk:DELay
		:INPut:CLOCk:DELay?
[8]	Relative	:INPut:CLOCk:RELative
		:INPut:CLOCk:RELative?
[9]	No Label (Relative value setting)	:INPut:CLOCk:RDELay
		:INPut:CLOCk:RDELay?
[10]	Calibration	:INPut:CLOCk:CALibration
[11]	Jitter Input	:INPut:CLOCk:JINPut
		:INPut:CLOCk:JINPut?

: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>

Parameter

<freq> = <STRING PROGRAM DATA>

When the Option-22 is installed:

"Variable"	Variable:	Variable
"OC_48"	OC48/STM16:	2.488320 Gbit/s
"PCI_EX1"	PCI Express Gen1:	2.500000 Gbit/s
"IB_SDR"	Infiniband SDR:	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 Gen2:	5.000000 Gbit/s
"USB3"	USB3.0:	5.000000 Gbit/s
"IB_DDR"	Infiniband DDR:	5.000000 Gbit/s
"SATA6"	SATA 6Gb/s:	6.000000 Gbit/s
"HSBI"	HSBI:	6.250000 Gbit/s
"PCI_EX3"	PCI Express Gen3:	8.000000 Gbit/s
"8G_FC"	8GFC:	8.500000 Gbit/s
"OC_192"	OC192/STM64:	9.953280 Gbit/s
"IB_QDR"	Infiniband QDR:	10.000000 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
"16G_FC"	16GFC:	14.025000 Gbit/s
"IB_FDR"	Infiniband_FDR:	14.062500 Gbit/s
"PCI_EX4"	PCI Express Gen4:	16.000000 Gbit/s
"SAS"	SAS:	24.000000 Gbit/s
"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

When the Option-23 is installed:

"Variable"	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.

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 ", "OC_192_G975", "OC_192_G709", "10GbE_G709", "10G_FC_G709", "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 28.100000 Gbit/s /1 kbit/s Step When the Option-23 is installed: 25.500000 to 32.100000 25.500000 to 32.100000 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.

:INPut:CLOCk:CRFReq?

Response	<numeric> = <NR2 NUMERIC RESPONSE DATA> When the Option-22 is installed: 2.400000 to 28.100000 2.40000 to 28.100000 Gbit/s /1 kbit/s Step When the Option-23 is installed: 25.500000 to 32.100000 25.500000 to 32.100000 Gbit/s /1 kbit/s Step
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: Operation Bitrate [Gbit/s] Range [MHz] (Step: 1 MHz) 2.400 000 to 5.500 000 Fixed to 3 MHz 5.500 001 to 7.500 000 3 to 4 MHz 7.500 001 to 9.500 000 3 to 5 MHz 9.500 001 to 10.500 000 3 to 6 MHz 10.500 001 to 12.500 000 3 to 7 MHz 12.500 001 to 14.500 000 3 to 8 MHz 14.500 001 to 15.500 000 3 to 9 MHz 15.500 001 to 17.500 000 3 to 10 MHz 17.500 001 to 19.500 000 3 to 11 MHz 19.500 001 to 20.500 000 3 to 12 MHz 20.500 001 to 22.500 000 3 to 13 MHz 22.500 001 to 24.500 000 3 to 14 MHz 24.500 001 to 25.500 000 3 to 15 MHz 25.500 001 to 27.500 000 3 to 16 MHz 27.500 001 to 28.100 000 3 to 17 MHz 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.

: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 Does not track.
INTernal[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	–1000 to 1000 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 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	–1000 to 1000 mUI
	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.	

: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

Figure 7.12.5.3-1 Measurement Restart Setting

Table 7.12.5.3-1 Measurement Restart Setting Commands

No.	Setting Item	Commands
[1]	Measurement Restart	:SENSe:MEASure:MREStart
		:SENSe:MEASure:MREStart?

: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

```
> :SENSe:MEASure:MREStart?
< 1,1
```

Compatibility
Compatible with MU181040A/B.

7.12.6 Commands Related to Capture Tab

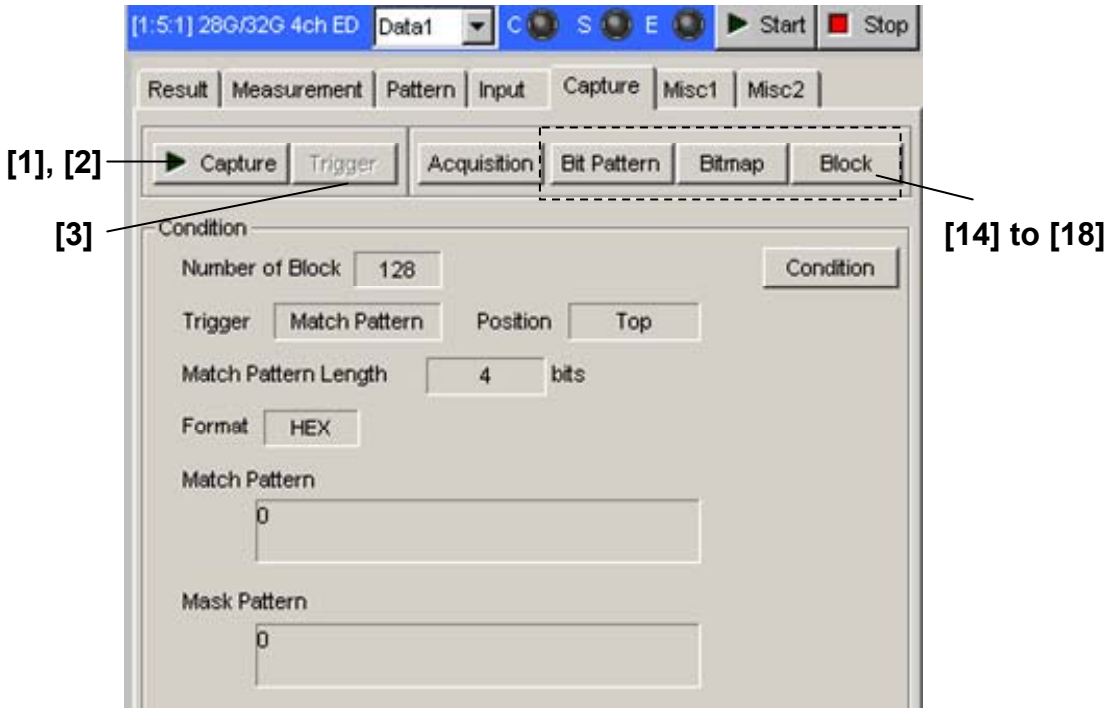


Figure 7.12.6-1 Capture Tab

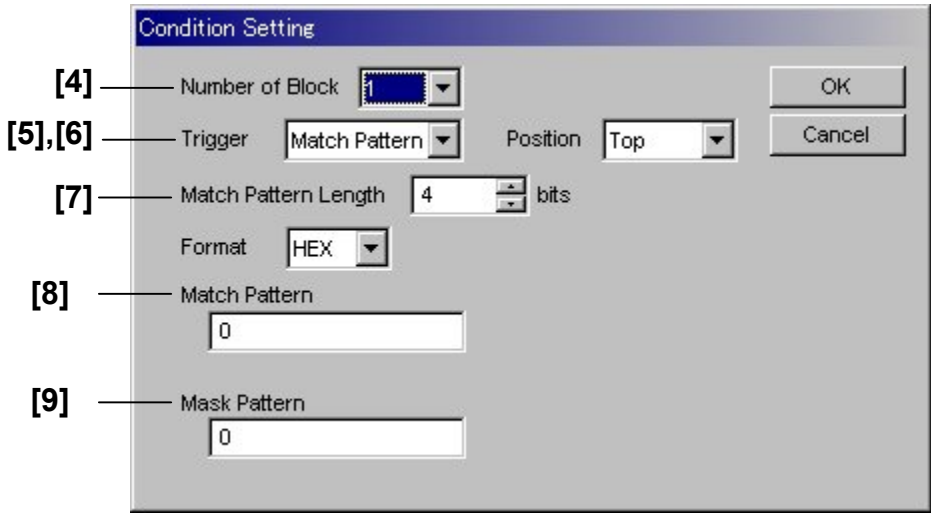


Figure 7.12.6-2 Condition Setting Dialog Box

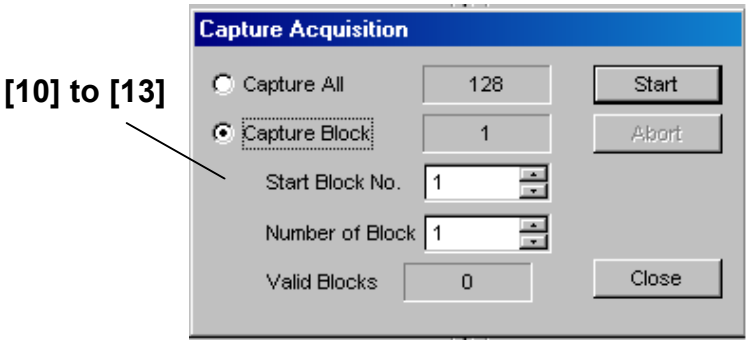


Figure 7.12.6-3 Capture Acquisition Dialog Box

Table 7.12.6-1 Capture Tab Setting Commands

No.	Setting Items	Commands
[1]	Capture Start	:SENSe:CAPTure:STARt
[2]	Capture Stop	:SENSe:CAPTure:STOP
[3]	Trigger	:SENSe:CAPTure:TRIGger
[4]	Number of Block	:SENSe:CAPTure:CONDition:BLOCk
		:SENSe:CAPTure:CONDition:BLOCk?
[5]	Trigger	:SENSe:CAPTure:CONDition:TRIGger
		:SENSe:CAPTure:CONDition:TRIGger?
[6]	Position	:SENSe:CAPTure:CONDition:POSition
		:SENSe:CAPTure:CONDition:POSition?
[7]	Match Pattern Length	:SENSe:CAPTure:CONDition:MPLength
		:SENSe:CAPTure:CONDition:MPLength?
[8]	Match Pattern	:SENSe:CAPTure:CONDition:MPEDit
		:SENSe:CAPTure:CONDition:MPEDit?
[9]	Mask Pattern	:SENSe:CAPTure:CONDition:MASKedit
		:SENSe:CAPTure:CONDition:MASKedit?
[10]	Query for capture data block count	:SENSe:CAPTure:ACQuisition:BNUMber?
[11]	Start of capture data acquisition	:SENSe:CAPTure:ACQuisition:STARt
[12]	Query for capture data acquisition state	:SENSe:CAPTure:ACQuisition:STATe?
[13]	Cancel of capture data acquisition	:SENSe:CAPTure:ACQuisition:CANCel
[14]	Query for capture data length per block	:SENSe:CAPTure:BPATtern:LENGth?
[15]	Query for capture result data	:SENSe:CAPTure:BPATtern:DATA:WHOLe?
[16]	Query for capture result data	:SENSe:CAPTure:BPATtern:BDATa:WHOLe?
[17]	Query for capture result error position	:SENSe:CAPTure:BPATtern:ERRor:WHOLe?
[18]	Query for capture result error position	:SENSe:CAPTure:BPATtern:BERRor:WHOLe?

:SENSe:CAPTure:STARt

Function	Starts capture result acquisition.
Example	`> :SENSe:CAPTure:STARt`
Compatibility	Compatible with MU181040A/B.

:SENSe:CAPTure:STOP

Function	Stops capture result acquisition.
Example	`> :SENSe:CAPTure:STOP`
Compatibility	Compatible with MU181040A/B.

:SENSe:CAPTure:TRIGger

Function	Generates a capture manual trigger.
Example	`> :SENSe:CAPTure:TRIGger`
Compatibility	Compatible with MU181040A/B.

:SENSe:CAPTure:CONDition:BLOCk <block>

Parameter	<block> = <CHARACTER PROGRAM DATA>	
	B1	1 Block
	B2	2 Block
	B4	4 Block
	B8	8 Block
	B16	16 Block
	B32	32 Block
	B64	64 Block
	B128	128 Block
Function	Sets the capture block division number.	
Example	To set the capture block division number to 8 blocks: `> :SENSe:CAPTure:CONDition:BLOCk B8`	
Compatibility	Compatible with MU181040A/B.	

:SENSe:CAPTure:CONDition:BLOCk?

Response	<block> = <CHARACTER RESPONSE DATA>	
	B1	1 Block
	B2	2 Block
	B4	4 Block
	B8	8 Block
	B16	16 Block
	B32	32 Block
	B64	64 Block
	B128	128 Block
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>

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	Compatible with MU181040A/B.	

:SENSe:CAPTure:CONDition:TRIGger?

Response	<trigger> = <CHARACTER RESPONSE DATA>
	EDET, PATT, MAN, EXT
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.

: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.	

:SENSe:CAPTure:ACQuisition:BNUMber?

Parameter	<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	Compatible with MU181040A/B.	

: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	1 to 128 blocks, in 1-block steps, acquisition start block
	[<number>] = <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 <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	Compatible with MU181040A/B.	

:SENSe:CAPTure:ACQuisition:STATe?

Response	<state> = <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: <length> = <NR1 NUMERIC RESPONSE DATA>
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:
<block> = <DECIMAL NUMERIC PROGRAM DATA>
1 to 128 — 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)

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 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: Partially compatible with MU181040A/B (the parameter range varies depending on the settings).

: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 7FFFFFF 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 #H7FFFFF	0 to 7FFFFF 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 — 1 to 128 blocks, in 1 steps

<start> = <NON-DECIMAL PROGRAM DATA>

#H0 to #H7FFFFF — 0 to 7FFFFF 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 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

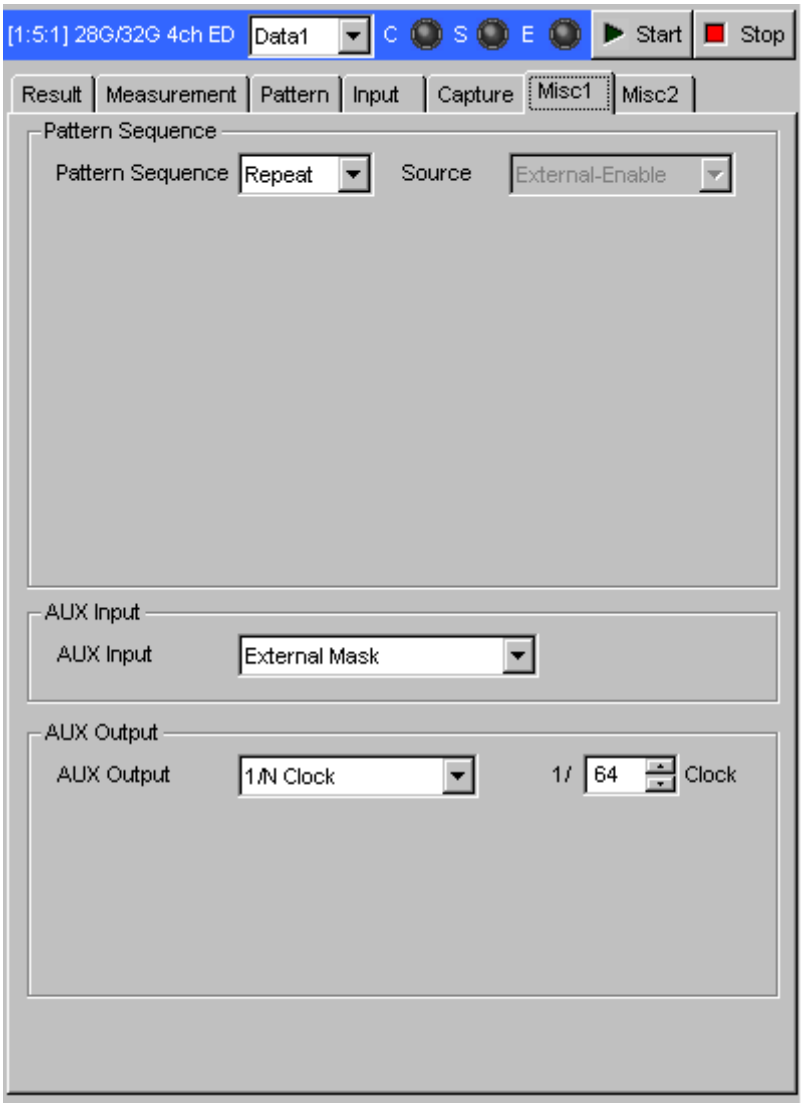


Figure 7.12.7-1 Misc1 Tab

7.12.7.1 Pattern Sequence setting commands

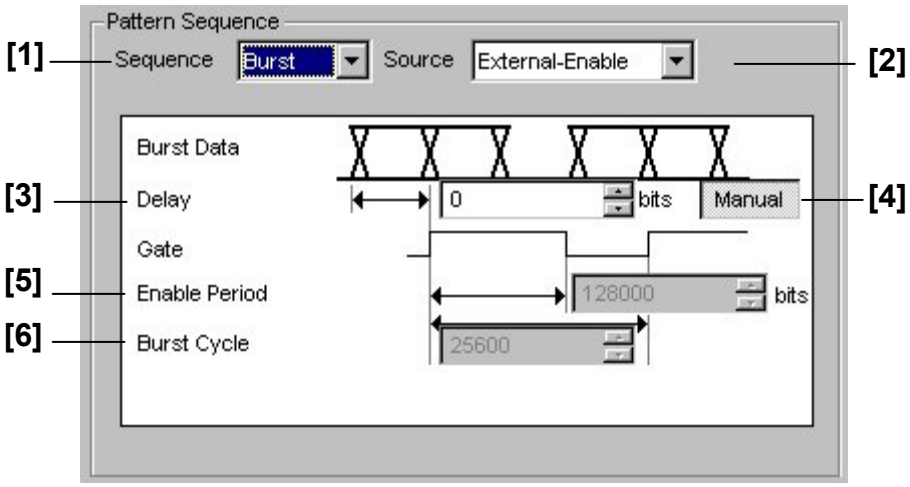


Figure 7.12.7.1-1 Pattern Sequence Setting

Table 7.12.7.1-1 Pattern Sequence Setting Commands

No.	Setting Items	Commands
[1]	Sequence	:SENSe:PATTern:IMODe
		:SENSe:PATTern:IMODe?
[2]	Source	:SENSe:PATTern:BURSt:MODE
		:SENSe:PATTern:BURSt:MODE?
[3]	Delay	:SENSe:PATTern:BURSt:DELay
		:SENSe:PATTern:BURSt:DELay?
[4]	Delay Manual/Auto	:SENSe:PATTern:BURSt:ADJust
		:SENSe:PATTern:BURSt:ADJust?
[5]	Enable Period	:SENSe:PATTern:BURSt:ELENgth
		:SENSe:PATTern:BURSt:ELENgth?
[6]	Burst Cycle	:SENSe:PATTern:BURSt:CYCLe
		:SENSe:PATTern:BURSt:CYCLe?

: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.

: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 EXTTrig 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.

: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

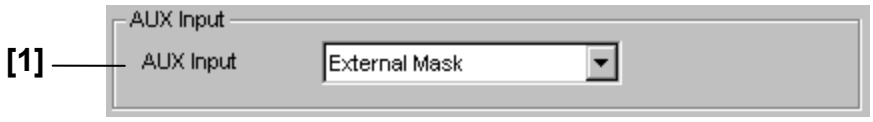


Figure 7.12.7.2-1 AUX Input Setting

Table 7.12.7.2-1 AUX Input Setting Commands

No.	Setting Items	Commands
[1]	AUX Input	:INPut:AUX:USAGe
		:INPut:AUX:USAGe?

: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	`> :INPut:AUX:USAGe?` `< BURS`
Compatibility	Compatible with MU181040A/B.

7.12.7.3 AUX Output setting commands

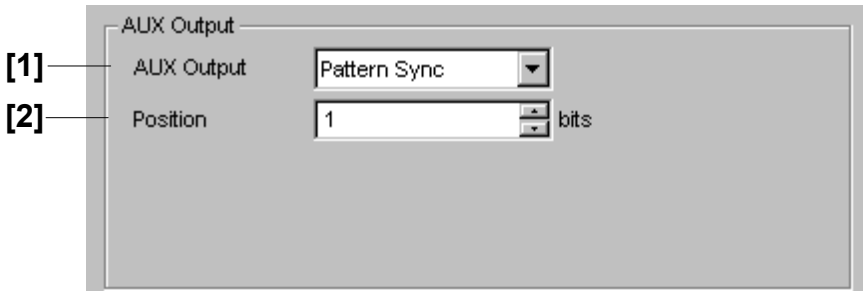


Figure 7.12.7.3-1 AUX Output Setting
(For PRBS, Zero Substitution, and Data Patterns)

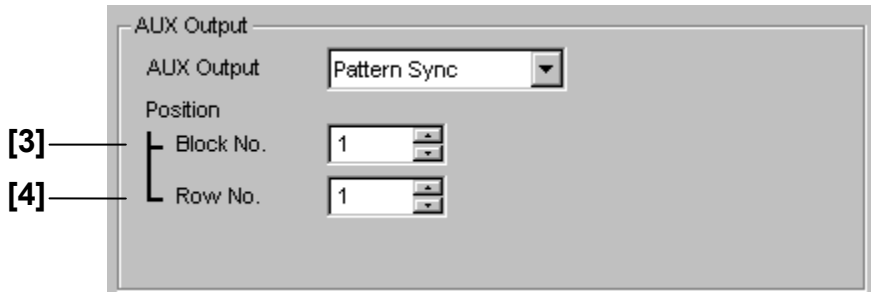


Figure 7.12.7.3-2 AUX Output Setting (For Mixed Pattern)

Table 7.12.7.3-1 AUX Output Setting Commands

No.	Setting Items	Commands
[1]	AUX Output	:OUTPut:SYNC:SOURce
		:OUTPut:SYNC:SOURce?
[2]	Position (PRBS, Zero-Substitution, Data)	:OUTPut:SYNC:POSition
		:OUTPut:SYNC:POSition?
[3]	Block No. (Mixed Data)	:OUTPut:SYNC:MIXData:BLOCk
		:OUTPut:SYNC:MIXData:BLOCk?
[4]	Row No. (Mixed Data)	:OUTPut:SYNC:MIXData:ROW
		:OUTPut:SYNC:MIXData:ROW?

:OUTPut:SYNC:SOURce <source>[,<numeric>]

Parameter	<source> = <CHARACTER PROGRAM DATA>	
	CLOCk8	1/8 clock
	CLOCk16	1/16 clock
	CLOCk32	1/32 clock
	CLOCk64	1/64 clock
	PATTern	Pattern Sync (Variable)
	SGAin	Sync Gain
	ERRorout	Error Output
	NCLock	1/n clock
	[<numeric>] = <DECIMAL NUMERIC PROGRAM DATA>	
	Omits <numeric> when NCLock (1/n clocks) is not selected.	
	4 to 512	4,6,8,10,…,510,512/2 Step
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?

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).
	4 to 512
Function	Queries the output signal for synchronization output.
Example	`> :OUTPut:SYNC:SOURce?`
	`< CLOC4`
	`> :OUTPut:SYNC:SOURce?`
	`< NCL,510`
Compatibility	Partially compatible with MU181040A/B.

:OUTPut:SYNC:POSition <numeric>

Parameter	<numeric> = <DECIMAL NUMERIC PROGRAM DATA> 1 to 34359738105 1 to 34,359,738,105 bits/8bits Step ***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?

Response	<numeric> = <NR1 NUMERIC RESPONSE DATA>
Function	Queries the output position for synchronization output for Data, PRBS, and Zero Substitution patterns.
Example	`> :OUTPut:SYNC:POSition?` `< 1`
Compatibility	Partially compatible with MU181040A/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 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?

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	Compatible with MU181040A/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 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	Compatible with MU181040A/B.

:OUTPut:SYNC:MIXData:ROW?

Response	<numeric> = <NR1 NUMERIC RESPONSE DATA> 1 to 16 1 to 16 Row
Function	Queries the output Row position for synchronization output for Mixed Data pattern.
Example	`> :OUTPut:SYNC:MIXData:ROW?` `< 1`
Compatibility	Compatible with MU181040A/B.

7.12.8 Commands Related to Misc2 Tab

7.12.8.1 Combination setting commands

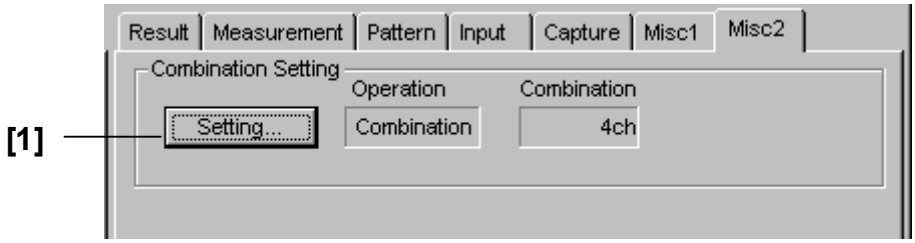


Figure 7.12.8.1-1 Misc2 Tab

Table 7.12.8.1-1 Combination Setting Commands

No.	Setting Item	Commands
[1]	Combination Setting	:MCOMBination:OPERation:SETTing
		:MCOMBination:OPERation:SETTing?

: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

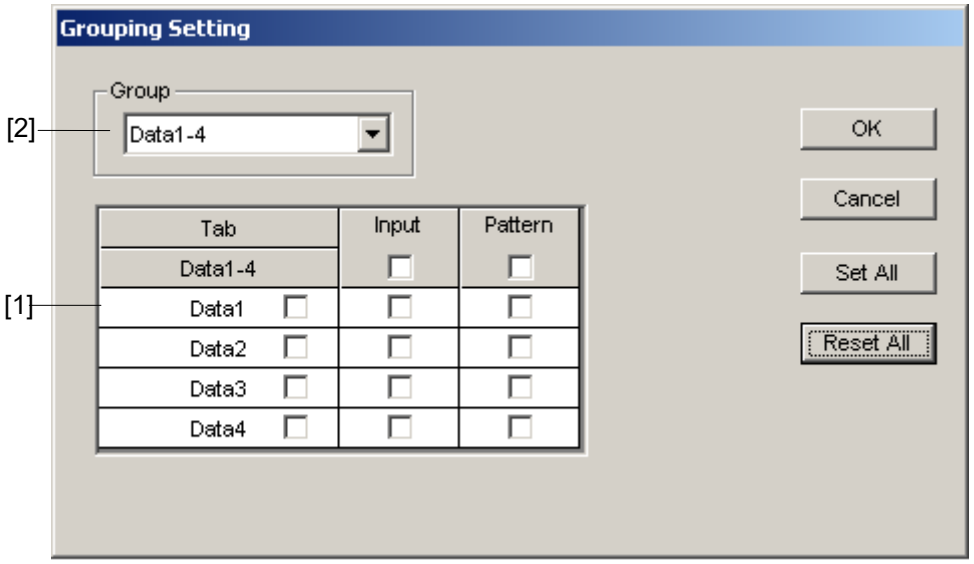


Figure 7.12.8.2-1 Grouping Setting Dialog Box for Misc 2 Tab

Table 7.12.8.2-1 Combination Setting Commands

No.	Setting Items	Commands
[1]	Selects tabs to be grouped together.	:SENSe:GROup:SETTing
		:SENSe:GROup:SETTing?
[2]	Selects Data IFs to be grouped together.	:SENSe:GROup:SELect
		:SENSe:GROup:SELect?

: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.

:SENSe:GROup:SETTing? <tab>

Parameter	<tab> = <CHARACTER PROGRAM DATA>	
	PATTern	Pattern tab
	INPut	Input 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 Input tabs are grouped together:	
Example	`> :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

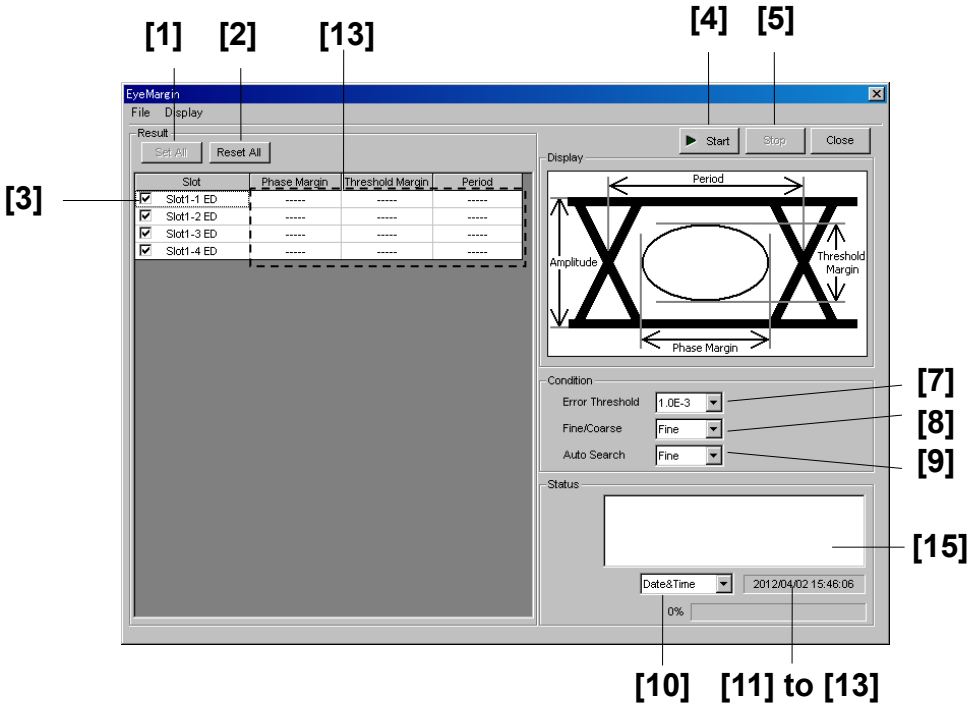


Figure 7.12.9.1-1 Eye Margin Window

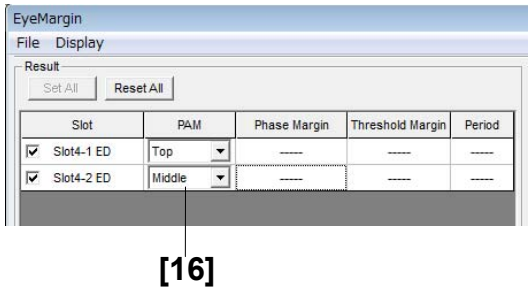


Figure 7.12.9.1-2 Eye Margin Window (PAM mode)

Table 7.12.9.1-1 Eye Margin measurement Setting Commands

No.	Setting Items	Commands
[1]	Set All	:SENSe:MEASure:EMARgin:MARGin:ASLot
[2]	Reset All	:SENSe:MEASure:EMARgin:MARGin:RSLot
[3]	Slot	:SENSe:MEASure:EMARgin:MARGin:SLOT
		:SENSe:MEASure:EMARgin:MARGin:SLOT?
[4]	Start	:SENSe:MEASure:STARt
[5]	Stop	:SENSe:MEASure:STOP
[6]	No label (Query for measurement state)	:SENSe:MEASure:EMARgin:STATe?
[7]	Error Threshold	:SENSe:MEASure:EMARgin:MARGin:THReshold
		:SENSe:MEASure:EMARgin:MARGin:THReshold?
[8]	Fine/Coarse	:SENSe:MEASure:EMARgin:MARGin:RESolution
		:SENSe:MEASure:EMARgin:MARGin:RESolution?
[9]	Auto Search	:SENSe:MEASure:EMARgin:MARGin:ASEarch
		:SENSe:MEASure:EMARgin:MARGin:ASEarch?
[10]	Selection of measurement time display	:SENSe:MEASure:EMARgin:TIME
		:SENSe:MEASure:EMARgin:TIME?
[11]	Query for date and time	:SENSe:MEASure:EMARgin:DTIMe?
[12]	Query for measurement start time	:SENSe:MEASure:EMARgin:STARt?
[13]	Query for measurement elapsed time	:SENSe:MEASure:EMARgin:ELAPsed?
[14]	Query for measurement result	:CALCulate:DATA:EMARgin?
[15]	No label (Query for measurement status)	:CALCulate:DATA:EMARgin:STATus?
[16]	Selection of search target threshold (In PAM mode only)	:SENSe:MEASure:EMARgin:MARGin:ASEarch:PAMSet
		:SENSe:MEASure:EMARgin:MARGin:ASEarch:PAMSet?

: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>,<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 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

: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>

Parameter
<type> = <CHARACTER PROGRAM DATA>

FINE	FINE mode
COARse	COARse mode

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?

Response
<type> = <CHARACTER RESPONSE DATA>
FINE, COAR

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>]

Parameter
<boolean> = <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 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

: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 STARt`	
Compatibility	Compatible with the MP1632C.	

:SENSe:MEASure:EMARgin:TIME?

Response	<type> = <CHARACTER RESPONSE DATA>	
	DTIM	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
<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 Eye Margin measurement.

Example
```
> :SENSe:MEASure:EMARgin:DTIMe?
< 2012,4,1,23,59,59
```

Compatibility
Incompatible with existing models

:SENSe:MEASure:EMARgin:STARt?

Response
<year> = <NR1 NUMERIC RESPONSE DATA>
0, 2000 to 2036 Year 2000 to 2036
<month> = <NR1 NUMERIC RESPONSE DATA>
0 to 12 0 to December
<day> = <NR1 NUMERIC RESPONSE DATA>
0 to 31 0 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 Eye Margin measurement start time.

Example
```
> :SENSe:MEASure:EMARgin:STARt?
< 2012,4,1,23,59,59
```

Compatibility
Compatible with the MP1632C.

: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>

Items	<string>	Response Format
Phase margin	"PHASe"	Form1
Threshold margin	"THReshold"	Form2
Period	"PERiod"	Form1

Response

<result> = <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 <result>

Form	Format	Description
Form1 Phase type	" XXXX"	For 0 to MAX (mUI/ps) Displays in the selected unit.
	"Failed"	For measurement failure
	"------"	When no data corresponds to the query.
Form2 Threshold type	" XXXX"	0 to MAX (mVpp)
	"Failed"	For measurement failure
	"------"	When no data corresponds to the query.

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 <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 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 7.12.9.2-1 File Menu Setting Commands

No.	Setting Items	Commands
[1]	Open	:SYSTem:MMEMory:MARGin:RECall
[2]	Save	:SYSTem:MMEMory:MARGin:STORe

: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

7.12.10 Eye Diagram Measurement

This section describes the commands for Eye Diagram measurement. Commands are described for each tab.

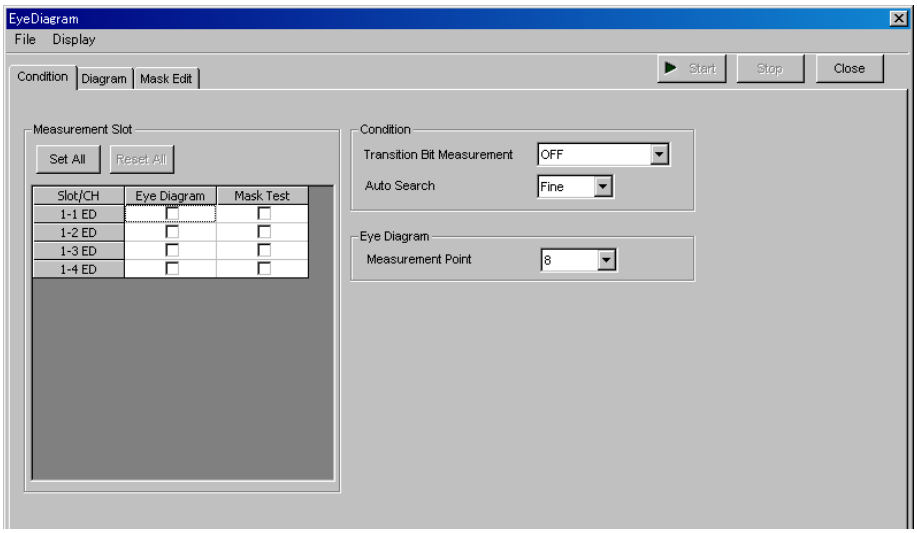


Figure 7.12.10-1 Eye Diagram Window

7.12.10.1 Condition tab

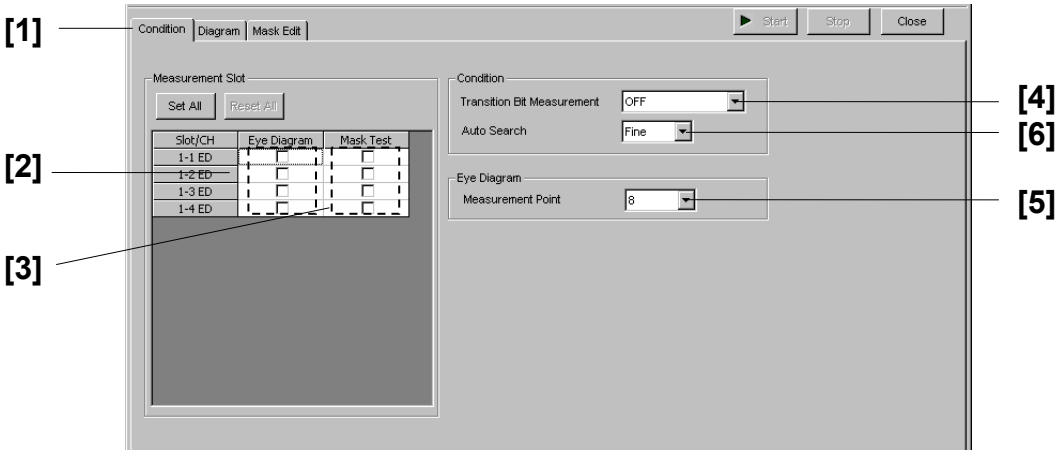


Figure 7.12.10.1-1 Condition Tab

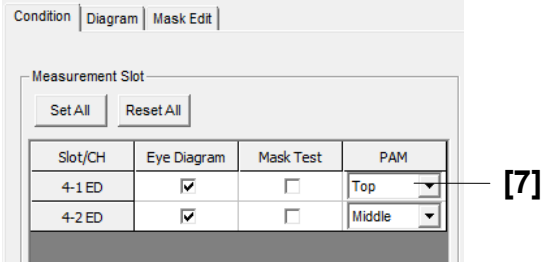


Figure 7.12.10.1-2 Condition Tab (PAM mode)

Table 7.12.10.1-1 Condition Tab Setting Commands

No.	Setting Items	Commands
[1]	Tab Selection	:DISPlay:RESult:EDIagram:TABSelect
		:DISPlay:RESult:EDIagram:TABSelect?
[2]	Slot Eye Diagram	:SENSe:MEASure:EDIagram:DIAGram:SLOT
		:SENSe:MEASure:EDIagram:DIAGram:SLOT?
[3]	Slot Mask Test	:SENSe:MEASure:EDIagram:MTESt:SLOT
		:SENSe:MEASure:EDIagram:MTESt:SLOT?
[4]	Transition Bit Measurement	:SENSe:MEASure:EDIagram:DIAGram:TRANsition
		:SENSe:MEASure:EDIagram:DIAGram:TRANsition?
[5]	Measurement Point	:SENSe:MEASure:EDIagram:DIAGram:TYPE
		:SENSe:MEASure:EDIagram:DIAGram:TYPE?
[6]	Auto Search	:SENSe:MEASure:EDIagram:DIAGram:ASEarch
		:SENSe:MEASure:EDIagram:DIAGram:ASEarch?
[7]	Selection of search target threshold (In PAM mode only)	:SENSe:MEASure:EDIagram:DIAGram:ASEarch:PAMSet
		:SENSe:MEASure:EDIagram:DIAGram:ASEarch:PAMSet?

: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

: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
< 1
```

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:EDIagram:MTESt:SLOT SLOT1,1,ON
```

Compatibility
Incompatible with existing models

:SENSe:MEASure:EDIagram: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:EDIagram:MTESt:SLOT? SLOT1,1
< 1
```

Compatibility
Incompatible with existing models

:SENSe:MEASure:EDIagram:DIAGram:TRANsition <type>

Parameter	<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:	
	`> :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	`> :SENSe:MEASure:EDIagram:DIAGram:TRANsition?`
	`< TRAN`
Compatibility	Incompatible with existing models

:SENSe:MEASure:EDIagram:DIAGram:TYPE <type>

Parameter	<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:	
	`> :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, POIN128
Function	Queries the number of measurement points during Eye Diagram measurement.
Example	`> :SENSe:MEASure:EDIagram:DIAGram:TYPE?`
	`< 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

: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

7.12.10.2 Diagram tab

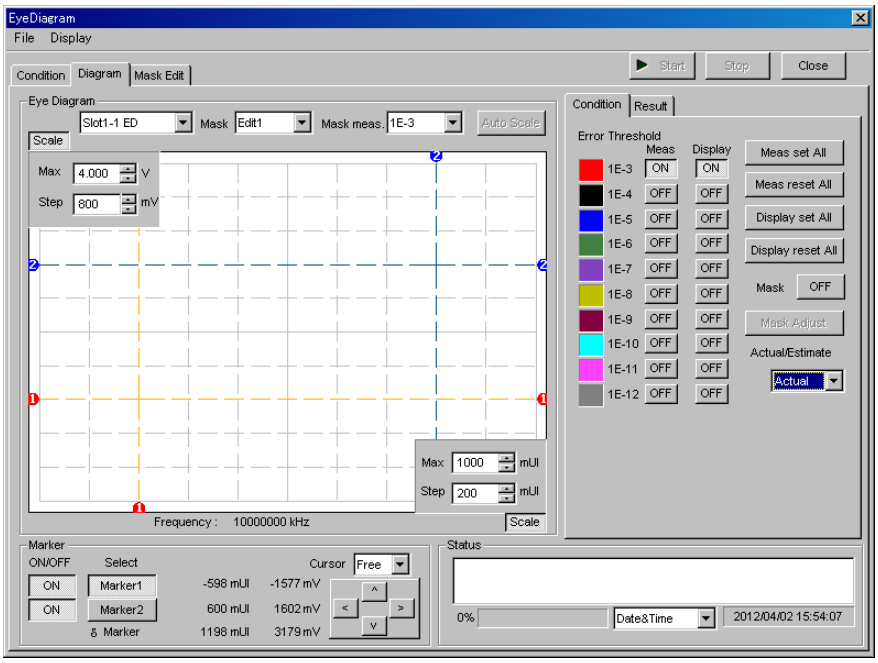


Figure 7.12.10.2-1 Diagram Tab

7.12.10.2.1 Graph display setting commands

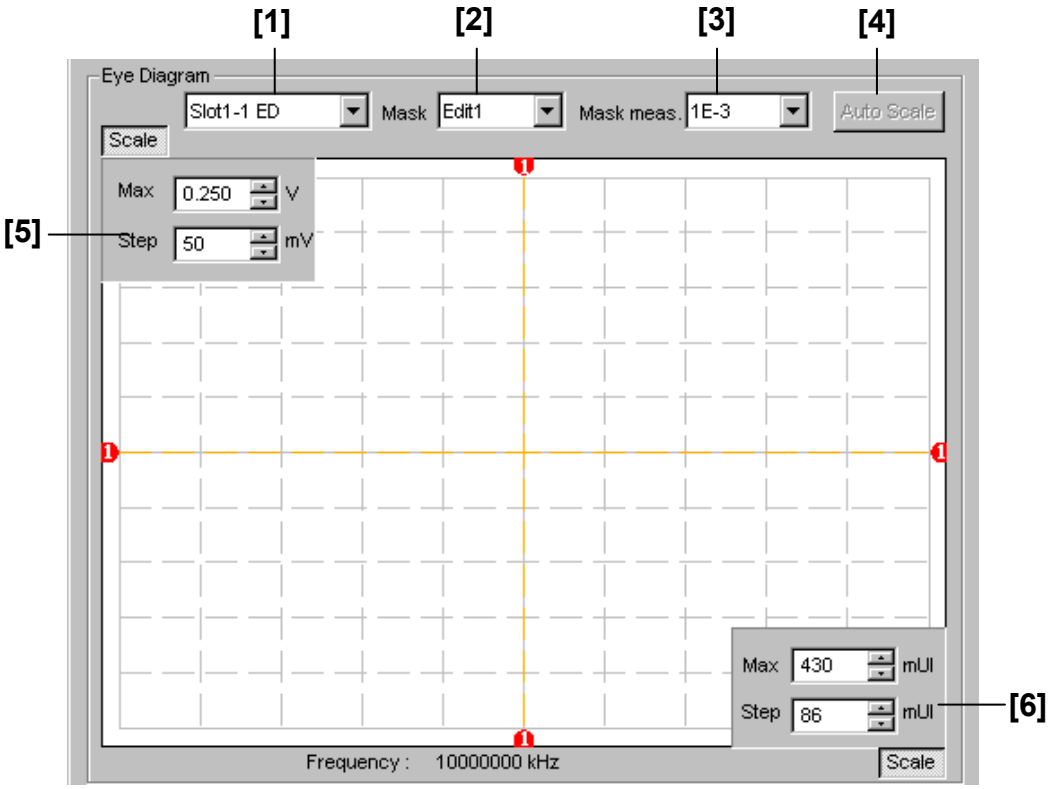


Figure 7.12.10.2.1-1 Graph Display Field

Table 7.12.10.2.1-1 Graph Display Setting Commands

No.	Setting Items	Commands
[1]	No label (Result display slot setting)	:DISPlay:RESult:EDIagram:SLOT
		:DISPlay:RESult:EDIagram:SLOT?
[2]	Mask	:DISPlay:RESult:EDIagram:TEMPlate:SELect
		:DISPlay:RESult:EDIagram:TEMPlate:SELect?
[3]	Mask meas.	:DISPlay:RESult:EDIagram:TEMPlate:THReshold
		:DISPlay:RESult:EDIagram:TEMPlate:THReshold?
[4]	Auto Scale	:DISPlay:RESult:EDIagram:SCALe:ASCale
[5]	Max/Step	:DISPlay:RESult:EDIagram:SCALe:VOLTage
		:DISPlay:RESult:EDIagram:SCALe:VOLTage?
[6]	Max/Step	:DISPlay:RESult:EDIagram:SCALe:PHASe
		:DISPlay:RESult:EDIagram:SCALe:PHASe?

: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 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

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	`> :DISPlay:RESult:EDIagram:SLOT?`
	`< SLOT1,1`
Compatibility	Incompatible with existing models

:DISPlay:RESult:EDIagram:TEMPlate:SELect <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:TEMPlate:SELect E2`
Compatibility	Incompatible with existing models

:DISPlay:RESult:EDIagram:TEMPlate:SELect?

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:TEMPlate:SELect?`
	`< E2`
Compatibility	Incompatible with existing models

:DISPlay:RESult:EDIagram:TEMPlate:THReshold <thre>

Parameter	<thre> = <CHARACTER PROGRAM DATA> E_3 to E_12 1E–3 to 1E–12 /1 Step
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:TEMPlate:THReshold E_3`
Compatibility	Incompatible with existing models

:DISPlay:RESult:EDIagram:TEMPlate:THReshold?

Response	<thre> = <CHARACTER RESPONSE DATA>
Function	Queries the Error Threshold for performing the Eye Diagram mask test.
Example	`> :DISPlay:RESult:EDIagram:TEMPlate:THReshold?` `< E_3`
Compatibility	Incompatible with existing models

: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 –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 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 –3.990 to 4.000 V <step> = <NR2 NUMERIC RESPONSE DATA> 0.001 to 0.800 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

:DISPlay:RESult:EDIagram:SCALe:PHASe <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 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> = <NR1 NUMERIC RESPONSE DATA> <step> = <NR1 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

7.12.10.2.2 Condition setting commands

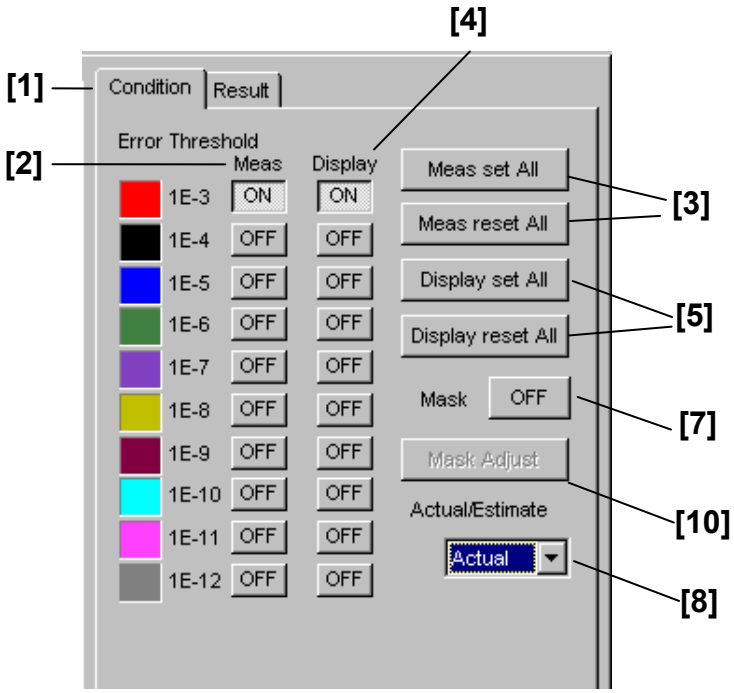


Figure 7.12.10.2.2-1 Condition Tab (Actual)

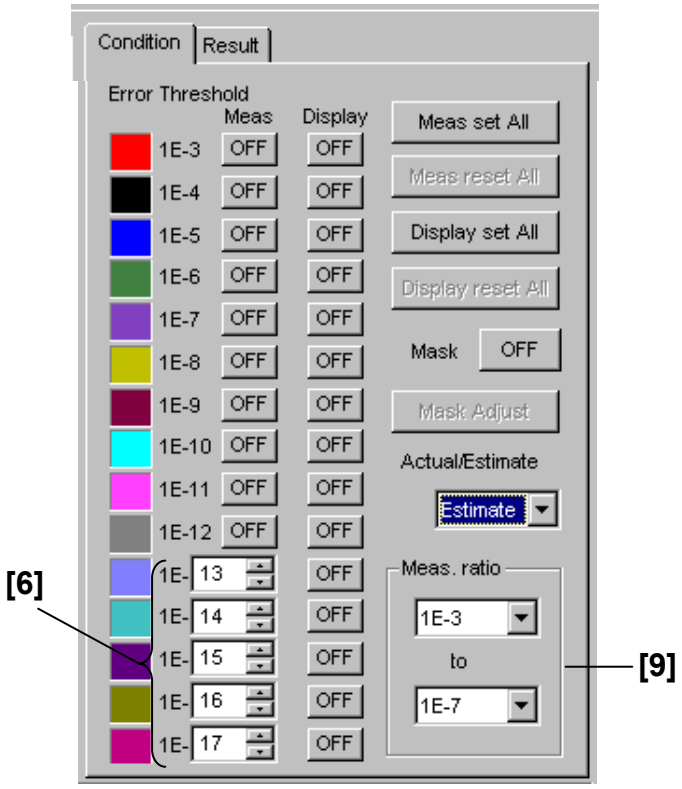


Figure 7.12.10.2.2-2 Condition Tab (Estimate)

Table 7.12.10.2.2-1 Condition Tab Setting Commands

No.	Setting Items	Commands
[1]	Condition Detail Result	:DISPlay:RESult:EDIagram:DIAGram:TABSelect
		:DISPlay:RESult:EDIagram:DIAGram:TABSelect?
[2]	Meas	:SENSe:MEASure:EDIagram:DIAGram:THReshold
		:SENSe:MEASure:EDIagram:DIAGram:THReshold?
[3]	Meas set All Meas reset All	:SENSe:MEASure:EDIagram:DIAGram:ATHReshold
[4]	Display	:DISPlay:RESult:EDIagram:ERATe
		:DISPlay:RESult:EDIagram:ERATe?
[5]	Display set All Display reset All	:DISPlay:RESult:EDIagram:AERate
[6]	Setting of error rate for Estimate measurement	DISPlay:RESult:EDIagram:EERate
		DISPlay:RESult:EDIagram:EERate?
[7]	Mask	:DISPlay:RESult:EDIagram:DISPlay:MASK
		:DISPlay:RESult:EDIagram:DISPlay:MASK?
[8]	Actual/Estimate	:DISPlay:RESult:EDIagram:ESTimate
		:DISPlay:RESult:EDIagram:ESTimate?
[9]	Meas.ratio	:SENSe:MEASure:EDIagram:DIAGram:BTHReshold
		:SENSe:MEASure:EDIagram:DIAGram:BTHReshold?
[10]	Mask Adjust	:DISPlay:RESult:EDIagram:ADJust
[11]	Query for measurement point	:SENSe:MEASure:EDIagram:DIAGram:POINt?
[12]	Query for measurement result	:CALCulate:DATA:DIAGram?

: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

: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
	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	

: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
<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

: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

: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>
<string> 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>

Item		<string>
Eye Map type	1E–3	"MAP:E_3:POINt1",...,"MAP:E_3:POINt128"
	1E–4	"MAP:E_4:POINt1",...,"MAP:E_4:POINt128"
	|	|
	1E–11	"MAP:E_11:POINt1",...,"MAP:E_11:POINt128"
	1E–12	"MAP:E_12:POINt1",...,"MAP:E_12:POINt128"

[<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

Type	Format	Description
Form1 Eye Map type	"XXXXX,YYYYY"	XXXXX: Phase (mUI) XXXXX: Phase (ps) (Displays in the selected unit.) YYYYY: Threshold (mV)
	"-----,-----"	When no data corresponds to the query.

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:

```
>:CALCulate:DATA:DIAGram? SLOT1,2,"MAP:E_3:POINt1"
<" 100, 1000"
```

Compatibility

Incompatible with existing models

7.12.10.2.3 Result query command

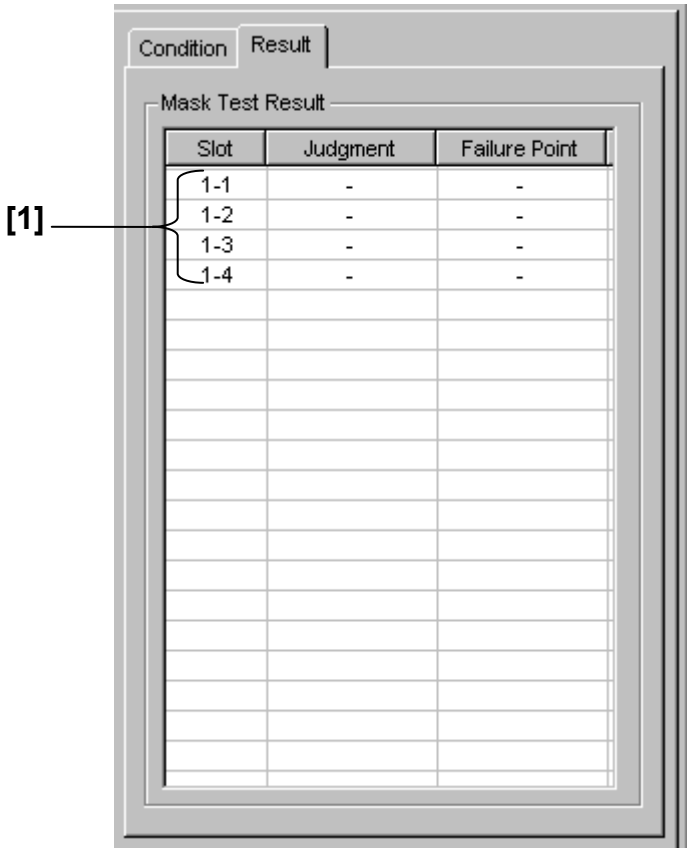


Figure 7.12.10.2.3-1 Result Tab

Table 7.12.10.2.3-1 Result Query Command

No.	Setting Item	Command
[1]	Query for Mask Test result	:CALCulate:DATA:EDIagram:MASK?

: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

Type	Format	Description
Form1 EyeResult type	"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 Mainframe No.3 Slot1 Data 1:

```
>: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

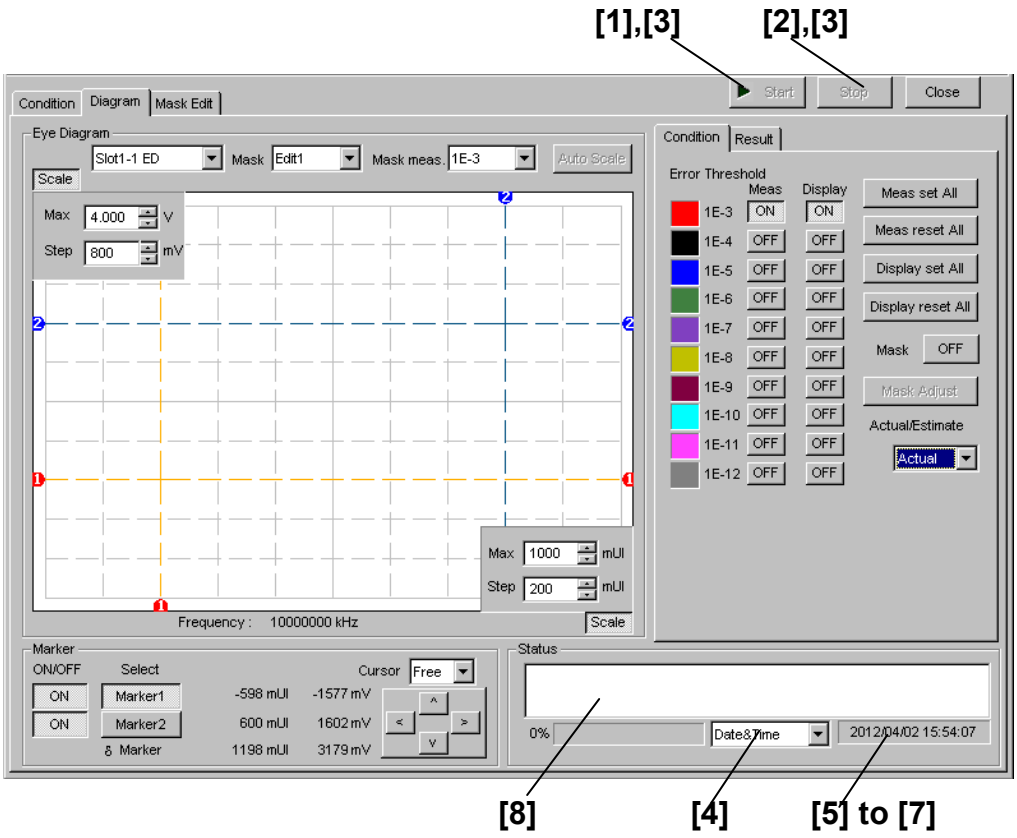


Figure 7.12.10.2.4-1 Diagram Tab, Start and Stop Buttons, and Status Field

Table 7.12.10.2.4-1 Diagram Tab (Status) Setting Commands

No.	Setting Items	Commands
[1]	Start	:SENSe:MEASure:EDIagram:DIAGram:STARt
[2]	Stop	:SENSe:MEASure:EDIagram:DIAGram:STOP
[3]	Query for measurement state	:SENSe:MEASure:EDIagram:STATe?
[4]	No label (Setting of time display)	:SENSe:MEASure:EDIagram:TIME
		:SENSe:MEASure:EDIagram:TIME?
[5]	Query for current time	:SENSe:MEASure:EDIagram:DTIMe?
[6]	Query for measurement start time	:SENSe:MEASure:EDIagram:STARt?
[7]	Query for measurement elapsed time	:SENSe:MEASure:EDIagram:ELAPsed?
[8]	Query for measurement status	:CALCulate:DATA:EDIagram:STATus?

:SENSe:MEASure:EDIagram:DIAGram:STARt

Function	Starts the Eye Diagram measurement.
Example	`> :SENSe:MEASure:EDIagram:DIAGram:STARt`
Compatibility	Incompatible with existing models

: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	<numeric> = <NR1 NUMERIC RESPONSE DATA>	
	1	Measurement has been started.
	0	Measurement has been stopped.
	–1	Measurement has failed.
Function	Queries the Eye Diagram measurement status.	
Example	`> :SENSe:MEASure:EDIagram:STATe?`	
	`< 1`	
Compatibility	Incompatible with existing models	

:SENSe:MEASure:EDIagram: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 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	<type> = <CHARACTER RESPONSE DATA>	
	DTIM	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 Diagram measurement time display type.	
Example	`> :SENSe:MEASure:EDIagram:TIME?`	
	`< STAR`	
Compatibility	Incompatible with existing models	

:SENSe:MEASure:EDIagram:DTIMe?

Response
<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 Eye Diagram measurement.

Example
```
> :SENSe:MEASure:EDIagram:DTIMe?
< 2012,4,1,23,59,59
```

Compatibility
Incompatible with existing models

:SENSe:MEASure:EDIagram:STARt?

Response
<year> = <NR1 NUMERIC RESPONSE DATA>
0, 2000 to 2036 Year 2000 to 2036
<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 Eye Diagram measurement start time (Start Time).

Example
```
> :SENSe:MEASure:EDIagram:STARt?
< 2012,4,1,23,59,59
```

Compatibility
Incompatible with existing models

:SENSe:MEASure:EDIagram:ELAPsed?

Response

<day> = <NR1 NUMERIC RESPONSE DATA>

0, 1 to 99	1 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 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

<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 or range

Function

Queries the status condition during Eye Diagram measurement.

Example

```
> :CALCulate:DATA:EDIagram:STATus?
< "Sync Loss"
```

Compatibility

Incompatible with existing models

7.12.10.2.5 Mask Edit tab setting commands

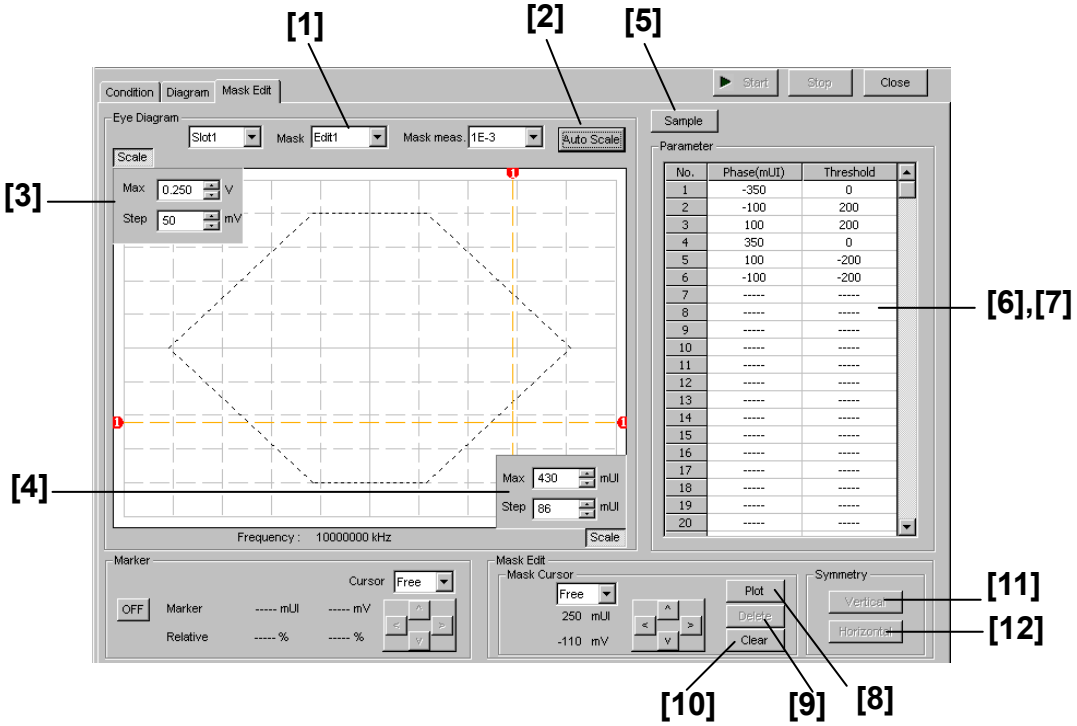


Figure 7.12.10.2.5-1 Mask Edit Tab

Table 7.12.10.2.5-1 Mask Edit Tab Setting Commands

No.	Setting Items	Commands
[1]	Mask	:DISPlay:RESult:EDIagram:TEMPlate:PATTern
		:DISPlay:RESult:EDIagram:TEMPlate:PATTern?
[2]	Auto Scale	:DISPlay:RESult:EDIagram:TEMPlate:ASCale
[3]	Max Step	:DISPlay:RESult:EDIagram:TEMPlate:SCALe
		:DISPlay:RESult:EDIagram:TEMPlate:SCALe?
[4]	Max Step	:DISPlay:RESult:EDIagram:TEMPlate:PSCale
		:DISPlay:RESult:EDIagram:TEMPlate:PSCale?
[5]	Sample	:DISPlay:RESult:EDIagram:TEMPlate:SAMPle
[6]	Query for mask point count	:DISPlay:RESult:EDIagram:TEMPlate:PNUMber?
[7]	Phase(mUI) Threshold	:DISPlay:RESult:EDIagram:TEMPlate:EDIT
		:DISPlay:RESult:EDIagram:TEMPlate:EDIT?
[8]	Plot	:DISPlay:RESult:EDIagram:TEMPlate:PLOT
[9]	Delete	:DISPlay:RESult:EDIagram:TEMPlate:DELete
[10]	Clear	:DISPlay:RESult:EDIagram:TEMPlate:CLEar
[11]	Vertical	:DISPlay:RESult:EDIagram:TEMPlate:VERTical
[12]	Horizontal	:DISPlay:RESult:EDIagram:TEMPlate:HORizontal

: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:PATTern?

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:PATTern?` `< 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:SCALe <max>,<step>

Parameter	<max> = <DECIMAL NUMERIC PROGRAM DATA>
	–3.990 to 4.000 –3.990 to 4.000 V/0.001 V Step
	<step> = <DECIMAL NUMERIC PROGRAM DATA>
	1 to 800 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:SCALe 4.000,800`
Compatibility	Incompatible with existing models

:DISPlay:RESult:EDIagram:TEMPlate:SCALe?

Response	<max> = <NR2 NUMERIC RESPONSE DATA> <step> = <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?` `< 4000,800`
Compatibility	Incompatible with existing models

:DISPlay:RESult:EDIagram:TEMPlate:PSCale <max>,<step>

Parameter	<max> = <DECIMAL NUMERIC PROGRAM DATA> –990 to 1000 –990 to 1000 mUI/1 mUI Step <step> = <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: `> :DISPlay:RESult:EDIagram:TEMPlate:PSCale 1000,10`
Compatibility	Incompatible with existing models

:DISPlay:RESult:EDIagram:TEMPlate:PSCale?

Response	<max> = <NR2 NUMERIC RESPONSE DATA> <step> = <NR2 NUMERIC RESPONSE DATA>
Function	Queries the maximum value and the grid resolution in the direction of the Phase of the Mask Edit graph.
Example	`> :DISPlay:RESult:EDIagram:TEMPlate:PSCale?` `< 1000,10`
Compatibility	Incompatible with existing models

:DISPlay:RESult:EDIagram:TEMPlate:SAMPle <string>

Parameter	<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	`> :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

:DISPlay:RESult:EDIagram:TEMPlate:DELete <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 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

No.	Setting Items	Commands
[1]	Open	:SYSTem:MMEMory:DIAGram:RECall
[2]	Save	:SYSTem:MMEMory:DIAGram:STORe

: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

: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.

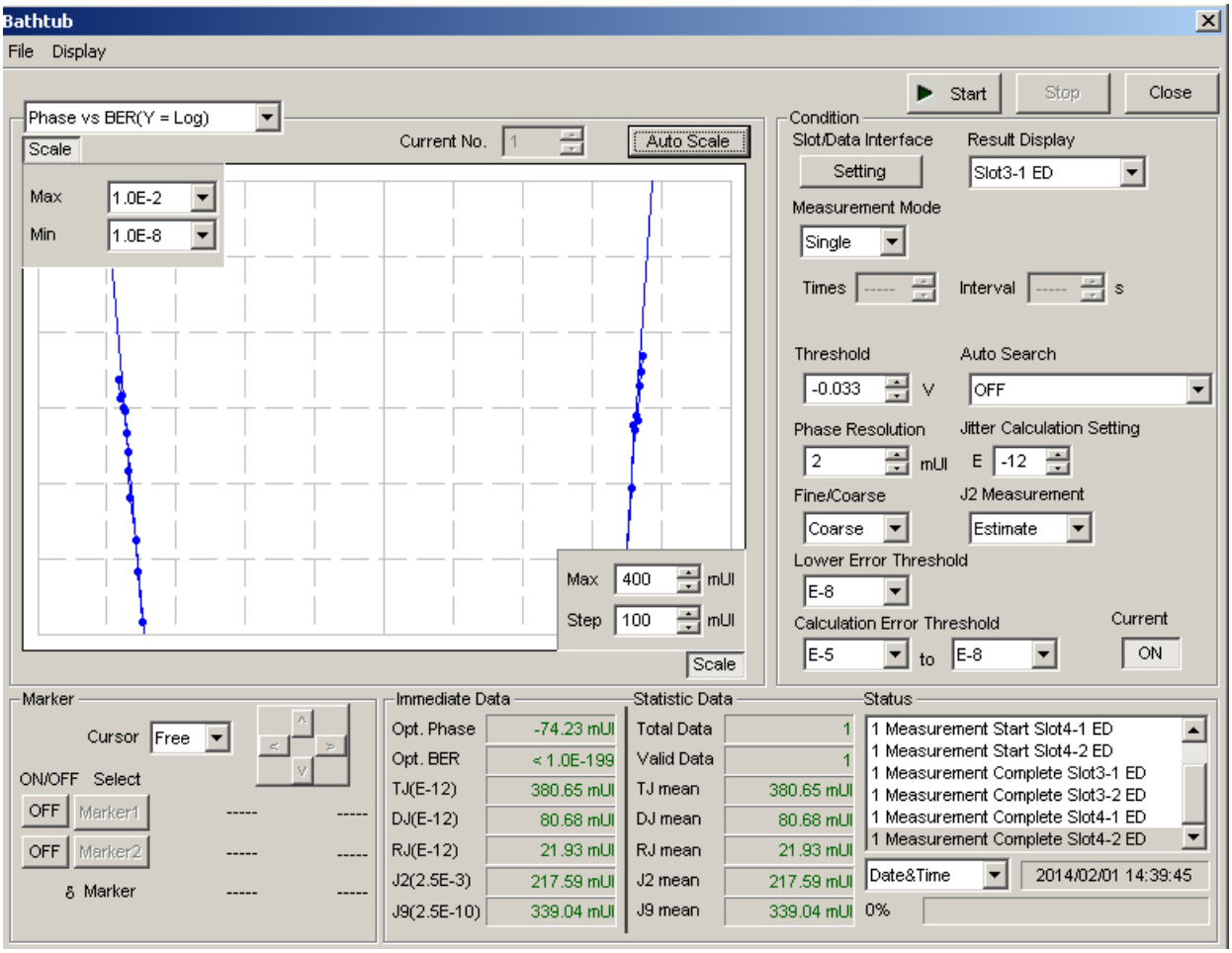


Figure 7.12.11-1 Bathtub Window

7.12.11.1 Measurement setting commands

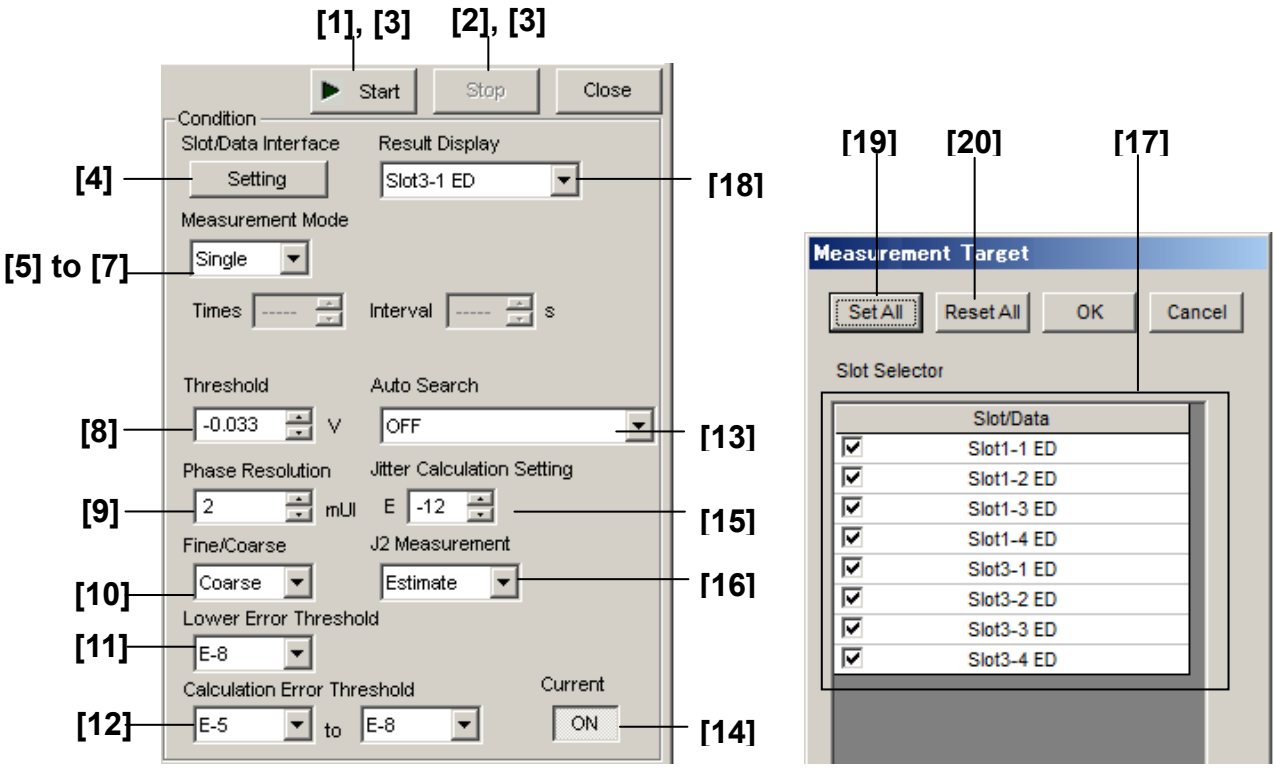


Figure 7.12.11.1-1 Condition Field

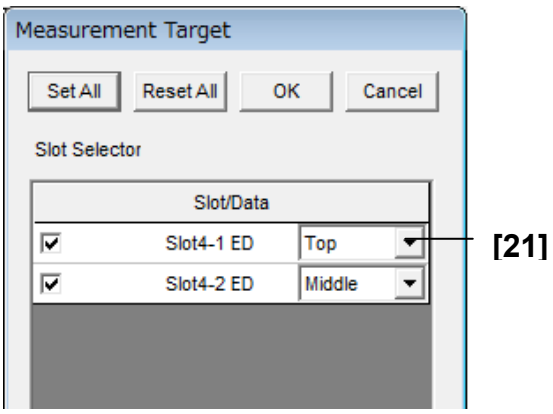


Figure 7.12.11.1-2 Measurement Target dialog box (PAM mode)

Table 7.12.11.1-1 Measurement Setting Commands

No.	Setting Items	Commands
[1]	Start	:SENSe:MEASure:BATHtub:STARt
[2]	Stop	:SENSe:MEASure:BATHtub:STOP
[3]	Measurement status query	:SENSe:MEASure:BATHtub:STATe?
[4]	Slot	:SENSe:MEASure:BATHtub:SLOT
		:SENSe:MEASure:BATHtub:SLOT?
[5]	Measurement Mode	:SENSe:MEASure:BATHtub:MODE
		:SENSe:MEASure:BATHtub:MODE?
[6]	Time	:SENSe:MEASure:BATHtub:TIMes
		:SENSe:MEASure:BATHtub:TIMes?
[7]	Interval	:SENSe:MEASure:BATHtub:INTerval
		:SENSe:MEASure:BATHtub:INTerval?
[8]	Threshold	:SENSe:MEASure:BATHtub:DATA
		:SENSe:MEASure:BATHtub:DATA?
[9]	Phase Resolution	:SENSe:MEASure:BATHtub:RESolution:MUI
		:SENSe:MEASure:BATHtub:RESolution:MUI?
[10]	Fine/Coarse	:SENSe:MEASure:BATHtub:ECOunt
		:SENSe:MEASure:BATHtub:ECOunt?
[11]	Lower Error Threshold	:SENSe:MEASure:BATHtub:RANGe
		:SENSe:MEASure:BATHtub:RANGe?
[12]	Calculation Error Threshold	:SENSe:MEASure:BATHtub:CALRange
		:SENSe:MEASure:BATHtub:CALRange?
[13]	Auto Search	:SENSe:MEASure:BATHtub:ASEarch
		:SENSe:MEASure:BATHtub:ASEarch?
[14]	Current	:SENSe:MEASure:BATHtub:CURRent
		:SENSe:MEASure:BATHtub:CURRent?
[15]	Jitter Calculation Setting	:SENSe:MEASure:BATHtub:JCALculation
		:SENSe:MEASure:BATHtub:JCALculation?
[16]	J2 Measurement	:SENSe:MEASure:BATHtub:J2Meas
		:SENSe:MEASure:BATHtub:J2Meas?
[17]	Selection of Slot(s) to be measured	:SENSe:MEASure:BATHtub:SELSlot
		:SENSe:MEASure:BATHtub:SELSlot?
[18]	Selection of Slot(s) you want to view measurement results	:SENSe:MEASure:BATHtub:DISPlay
[19]	Set All	:SENSe:MEASure:BATHtub:SLASet
[20]	Reset All	:SENSe:MEASure:BATHtub:SLAReset
[21]	Selection of search target threshold. (In PAM mode only)	:SENSe:MEASure:BATHtub:ASEarch:PAMSet
		:SENSe:MEASure:BATHtub:ASEarch:PAMSet?

: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?

Response	<numeric> = <NR1 NUMERIC RESPONSE DATA>
	0 During measurement stop
	1 During measurement
Function	Queries the measurement processing state during Bathtub measurement.
Example	`> :SENSe:MEASure:BATHtub:STATe?` `< 1`
Compatibility	Incompatible with existing models

:SENSe:MEASure:BATHtub:SLOT <numeric>,<interface>[,<unit>]

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 Bathtub measurement target slot.
Example	To set the target slot to No.3, Data1: `> :SENSe:MEASure:BATHtub:SLOT 3,1`
Compatibility	Incompatible with existing models

: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

: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 RESONSE 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 Auto search OFF
	PHASe Phase auto search (Coarse)
	PTHReshold Phase & threshold auto search (Coarse)
	FPHase Phase auto search (Fine)
	FPTHreshold Phase & Threshold auto search (Fine)
	PPTHreshold Phase & Threshold auto search (PAM Coarse)
	PFPThreshold Phase & Threshold auto search (PAM 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, PPTH, PFPT
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

: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
	<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.
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

: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

Parameter	None
Function	Sets all Slot and Data Interface combinations as the Bathtub measurement objects.
Example	`> :SENSe:MEASure:BATHtub:SLASet`
Compatibility	Incompatible with existing models

:SENSe:MEASure:BATHtub:SLAReset

Parameter	None
Function	Resets all Slot and Data Interface combinations set as the Bathtub measurement objects.
Example	`> :SENSe:MEASure:BATHtub:SLAReset`
Compatibility	Incompatible with existing models

:SENSe:MEASure:BATHtub:ASEarch:PAMSet <slot>,<interface>,<threshold>[,<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 <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 Bathtub measurement.
Example	To set Data 3 of Slot No.1 for Top threshold search: `> :SENSe:MEASure:BATHtub:ASEarch:PAMSet SLOT1,3,TOP`
Compatibility	Incompatible with existing models

: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

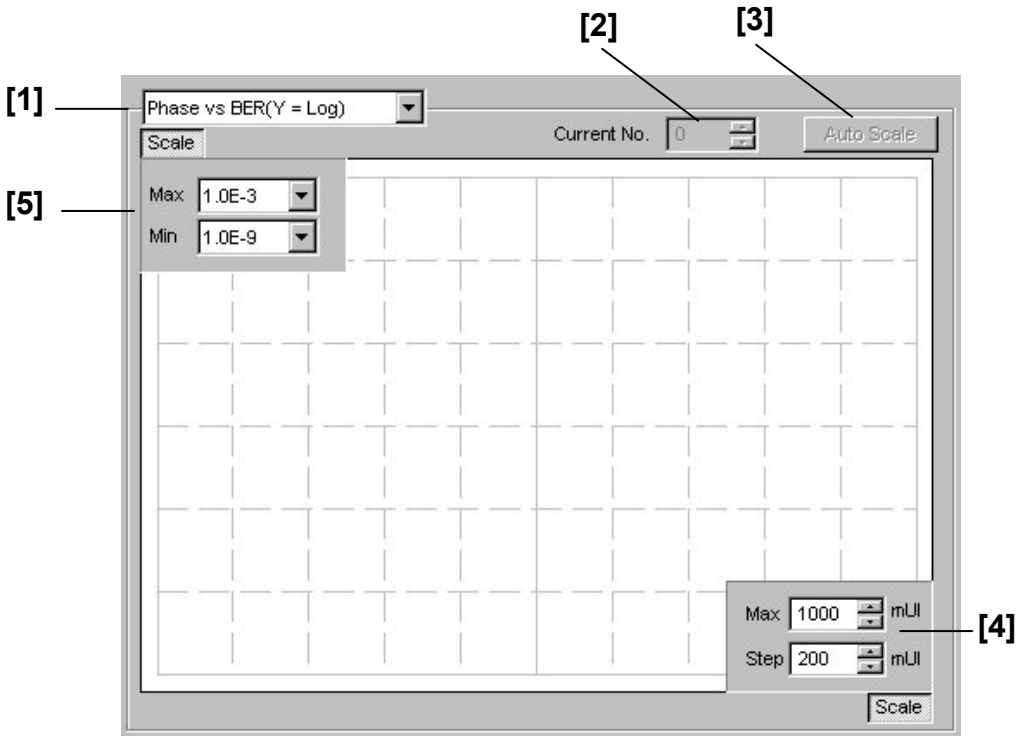


Figure 7.12.11.2-1 Graph Display Field

Table 7.12.11.2-1 Graph Display Setting Commands

No.	Setting Items	Commands
[1]	Display graph selection	:DISPlay:RESult:BATHtub:ITEM
		:DISPlay:RESult:BATHtub:ITEM?
[2]	Current No.	:DISPlay:RESult:BATHtub:MNUMber
		:DISPlay:RESult:BATHtub:MNUMber?
[3]	Auto Scale	:DISPlay:RESult:BATHtub:SCALe:ASCale
[4]	Horizontal axis scale setting	:DISPlay:RESult:BATHtub:SCALe:HORizontal
		:DISPlay:RESult:BATHtub:SCALe:HORizontal?
[5]	Vertical axis scale setting	:DISPlay:RESult:BATHtub:SCALe:VERTical
		:DISPlay:RESult:BATHtub:SCALe:VERTical?

: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	

: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

:DISPlay:RESult:BATHtub:SCALe:HORizontal <max>[,<step>]

Parameter <max>,<step> = <DECIMAL NUMERIC PROGRAM DATA>

Table 7.12.11.2-2 Parameters

Form	<max>	<step>
Phase type	–900 to 1000 (mUI), in 100-mUI steps	10 to 200 (mUI), in 10-mUI steps
Histogram type	16,32,64,128,256	Omitted

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 7.12.11.2-3 Parameters

Form	<max>	<min>
Histogram Type	6 to 15 (E + 6 to E + 15)	Input 0
Error Rate Type	2 to 7 (E–2 to E–7)	8 to 15 (E–8 to E–15)

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

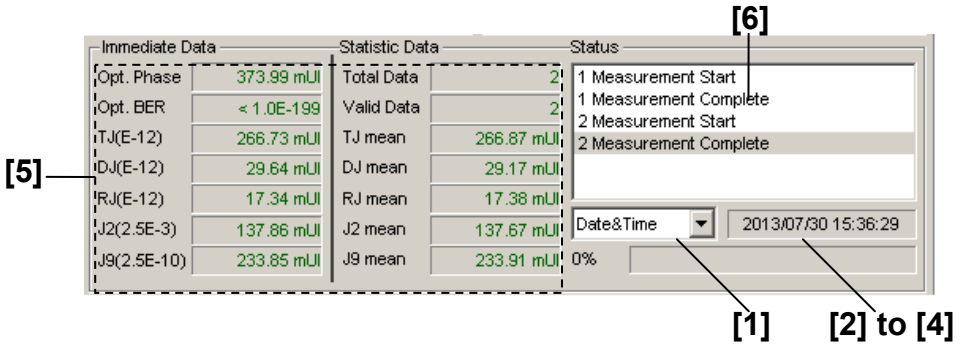


Figure 7.12.11.3-1 Measurement Result and Status Display

Table 7.12.11.3-1 Measurement Result and Status Query Commands

No.	Setting Items	Commands
[1]	Display time selection	:SENSe:MEASure:BATHtub:DTIME
		:SENSe:MEASure:BATHtub:DTIME?
[2]	Query for current time	:SENSe:MEASure:BATHtub:TIME:DTIMe?
[3]	Query for measurement start time	:SENSe:MEASure:BATHtub:TIME:STARt?
[4]	Query for measurement elapsed time	:SENSe:MEASure:BATHtub:TIME:ELAPsed?
[5]	Query for result	:CALCulate:DATA:BATHtub?
[6]	Query for status	:CALCulate:DATA:BATHtub:STATus?

: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 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 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	<year> = <NR1 NUMERIC RESPONSE DATA>
	0, 2000 to 2036 Year 2000 to 2036
	<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 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>

Items		<Result1>	Response Format
Phase vs BER (Phase/Error Rate)	Number of measurements 1 to 1000	"BATH:1"…"BATH:1000"	Form1
Histogram (Phase/Error Count)		"HIST:1"…"HIST:1000"	Form2

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>

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 7.12.11.3-4 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

Response <string> = <STRING RESPONSE DATA>

Table 7.12.11.3-5 Response Format

Form	Format	Description
Form1 Phase-BER type	"XXXX.XX, Y.YYYYE-YYY"	XXXX.XX: Phase (mUI or PS) Y.YYYYE-YYY: Error Rate
	"-------,-------"	When no data corresponds to the query.
Form2 Histogram type	"XXXX.XX, Y.YYYYE-YYY"	XXXX.XX: Phase (mUI or PS) Y.YYYYE-YYY: Error Count
	"-------,-------"	When no data corresponds to the query.
Form3 Fraction type 1	"X.XXXXE-XXX"	When 0.0000E–016 to 1.0000E000
	"--------"	When no data corresponds to the query.
	"< 1.0E-199"	When E–199 or less
Form4 Fraction type 2	"XXXX.XX"	When –9999.99 to 9999.99
	"--------"	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 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?

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:

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"
```

7.12.11.4 File menu setting commands

Table 7.12.11.4-1 File Menu Setting Commands

No.	Setting Items	Commands
[1]	Open	:SYSTem:MMEMory:BATHtub:RECall
[2]	Save	:SYSTem:MMEMory:BATHtub:STORe

: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

7.12.12 Auto Search

This chapter describes the commands that are used after Auto Search is set by the :SYSTem:CFUNction command.

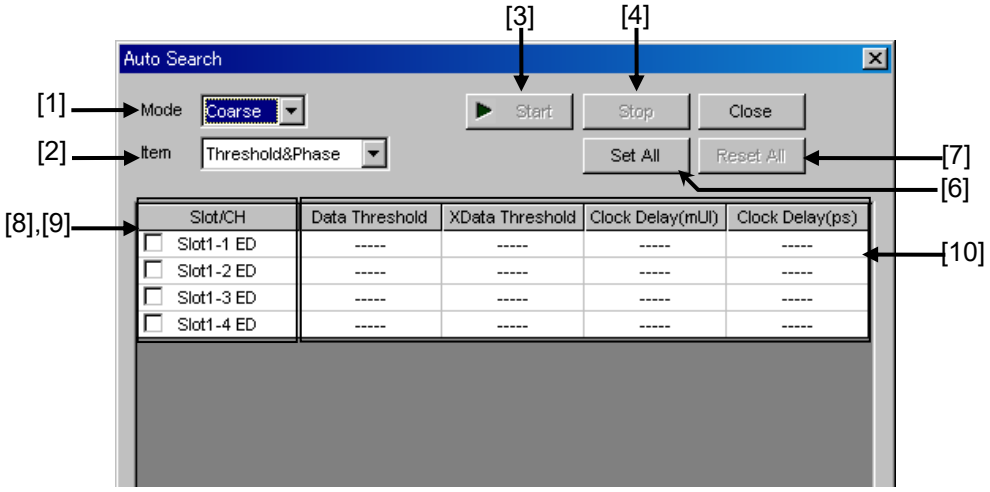


Figure 7.12.12-1 Auto Search Dialog Box

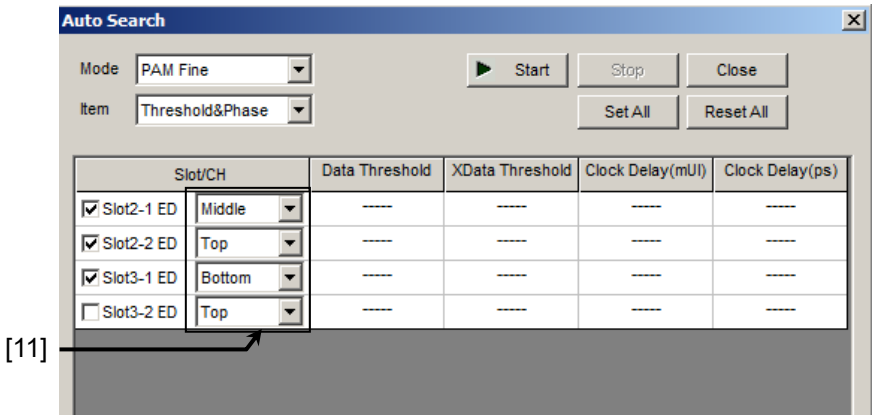


Figure 7.12.12-2 Auto Search Dialog Box (PAM Mode)

Table 7.12.12-1 Auto Search Setting Commands

No.	Setting Items	Commands
[1]	Mode	:SENSe:MEASure:ASEarch:SMODe
		:SENSe:MEASure:ASEarch:SMODe?
[2]	Item	:SENSe:MEASure:ASEarch:MODE
		:SENSe:MEASure:ASEarch:MODE?
[3]	Start	:SENSe:MEASure:ASEarch:STARt
[4]	Stop	:SENSe:MEASure:ASEarch:STOP
[5]	No label (Query for function state)	:SENSe:MEASure:ASEarch:STATe?
[6]	Set All	:SENSe:MEASure:ASEarch:SLASet
[7]	Reset All	:SENSe:MEASure:ASEarch:SLAReset
[8]	Slot Select	:SENSe:MEASure:ASEarch:SELSlot
		:SENSe:MEASure:ASEarch:SELSlot?

Table 7.12.12-1 Auto Search Setting Commands (Cont'd)

No.	Setting Items	Commands
[9]	No label (Query for slot state)	:SENSe:MEASure:ASEarch:SLOT?
[10]	No label (Query for auto search results)	:CALCulate:DATA:ASEarch?
[11]	Search target threshold (In PAM mode only)	:SENSe:MEASure:ASEarch:PAMSet
		:SENSe:MEASure:ASEarch:PAMSet?

:SENSe:MEASure:ASEarch:SMODe <mode>

Parameter	<mode> = <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> = <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> = <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	> :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	<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	> :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

: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 NUMERICRESPONSE 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

: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:

```
> :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 <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 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:

```
> :SENSe:MEASure:ASEarch:PAMSet SLOT1,3,BOTTom
```

Compatibility
Incompatible with existing models.

:SENSe:MEASure: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 of the slot for running PAM Auto Search.

Example
To query search target threshold of Data 3 of Slot No. 1:

```
> :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.

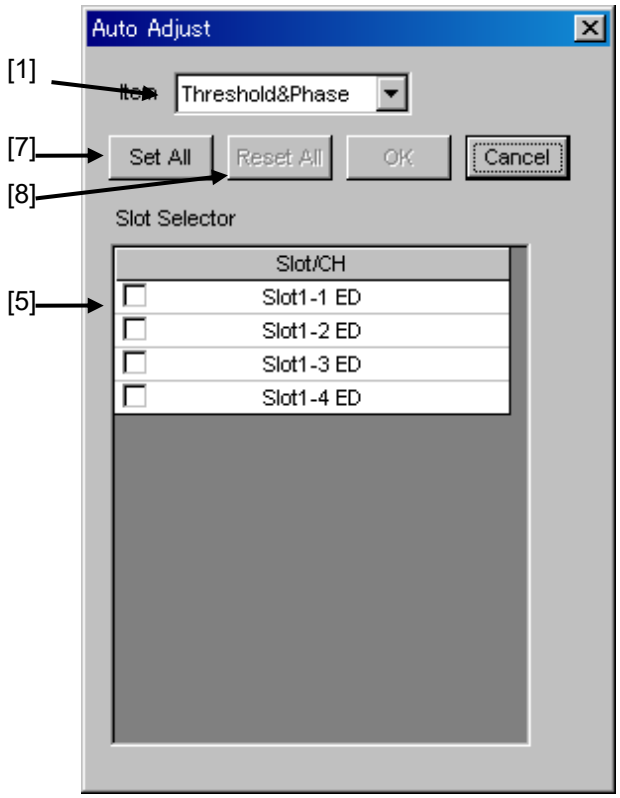


Figure 7.12.13-1 Auto Adjust Dialog Box

Table 7.12.13-1 Auto Adjust Setting Commands

No.	Setting Items	Commands
[1]	Item	:SENSe:MEASure:AADJust32:ITEM
		:SENSe:MEASure:AADJust32:ITEM?
[2]	Start	:SENSe:MEASure:AADJust32:STARt
[3]	Stop	:SENSe:MEASure:AADJust32:STOP
[4]	No label (Query for function state)	:SENSe:MEASure:AADJust32:STATe?
[5]	Slot Select	:SENSe:MEASure:AADJust32:SELSlot
		:SENSe:MEASure:AADJust32:SELSlot?
[6]	No label (Query for slot state)	:SENSe:MEASure:AADJust32:SLOT?
[7]	Set All	:SENSe:MEASure:AADJust32:SLASet
[8]	Reset All	:SENSe:MEASure:AADJust32:SLAReset

: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	To set the execution item of Auto Adjust to Phase & Threshold:	
	`> :SENSe:MEASure:AADJust32:ITEM PTHReshold`	
Compatibility	Incompatible with existing models	

:SENSe:MEASure:AADJust32:ITEM?

Response	<item> = <CHARACTER RESPONSE DATA>	
	PTHR	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	

: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>]

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 ***Note:*** The above response returns all slots for which Auto Adjust can be executed by delimiting with commas (,). <interface> = <NR1 NUMERIC RESPONSE DATA> 1 to 4 Data interface No.1 to 4
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:
	`> :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	`> :SENSe:MEASure:AADJust32:SLASet`
Compatibility	Incompatible with existing models

:SENSe:MEASure:AADJust32:SLAReset

Function	Deselect all slots for Auto Adjust execution.
Example	`> :SENSe:MEASure:AADJust32:SLAReset`
Compatibility	Incompatible with existing models

7.12.14 Q measurement

This section describes the commands for Q measurement.

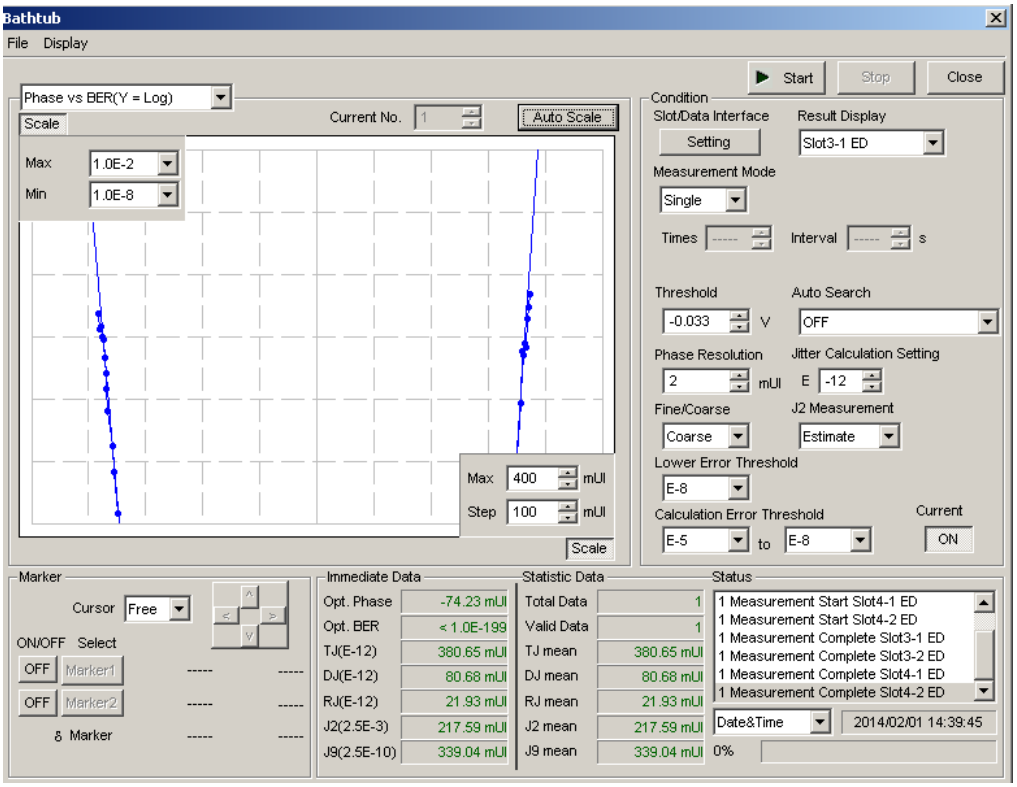


Figure 7.12.14-1 Q measurement screen

7.12.14.1 Vth vs. Q measurement commands

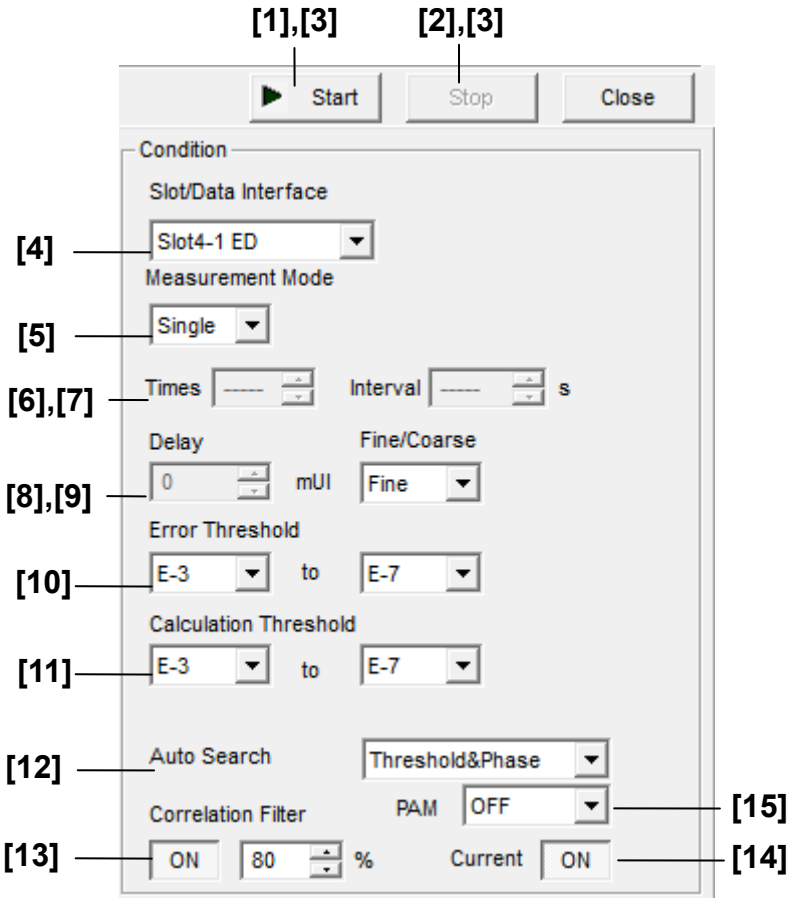


Figure 7.12.14.1-1 Vth vs. Q measurement setting field

Table 7.12.14.1-1 Vth vs. Q measurement setting commands

No.	Setting Items	Commands
[1]	Start	:SENSe:MEASure:QANalysis:THReshold:STARt
[2]	Stop	:SENSe:MEASure:QANalysis:THReshold:STOP
[3]	Query for measurement progress state	:SENSe:MEASure:QANalysis:STATe?
[4]	Slot	:SENSe:MEASure:QANalysis:THReshold:SLOT
		:SENSe:MEASure:QANalysis:THReshold:SLOT?
[5]	Meas Mode	:SENSe:MEASure:QANalysis:THReshold:MODE
		:SENSe:MEASure:QANalysis:THReshold:MODE?
[6]	Times	:SENSe:MEASure:QANalysis:THReshold:TIMEs
		:SENSe:MEASure:QANalysis:THReshold:TIMEs?
[7]	Interval	:SENSe:MEASure:QANalysis:THReshold:INTerval
		:SENSe:MEASure:QANalysis:THReshold:INTerval?
[8]	Delay	:SENSe:MEASure:QANalysis:THReshold:DELay
		:SENSe:MEASure:QANalysis:THReshold:DELay?

Table 7.12.14.1-1 Vth vs. Q measurement setting commands (Cont'd)

No.	Setting Items	Commands
[9]	Fine/Coarse	:SENSe:MEASure:QANalysis:THReshold:RESolution
		:SENSe:MEASure:QANalysis:THReshold:RESolution?
[10]	Error Threshold	:SENSe:MEASure:QANalysis:THReshold:ERANge
		:SENSe:MEASure:QANalysis:THReshold:ERANge?
[11]	Calculation Threshold	:SENSe:MEASure:QANalysis:THReshold:CALRange
		:SENSe:MEASure:QANalysis:THReshold:CALRange?
[12]	Auto Search	:SENSe:MEASure:QANalysis:THReshold:ASEarch
		:SENSe:MEASure:QANalysis:THReshold:ASEarch?
[13]	Correlation Filter	:SENSe:MEASure:QANalysis:THReshold:CFILter
		:SENSe:MEASure:QANalysis:THReshold:CFILter?
[14]	Current	:SENSe:MEASure:QANalysis:THReshold:CURRent
		:SENSe:MEASure:QANalysis:THReshold:CURRent?
[15]	Selection of PAM search target threshold	:SENSe:MEASure:QANalysis:THReshold:ASEarch:PAMSet
		:SENSe:MEASure:QANalysis:THReshold:ASEarch:PAMSet?

: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?`	
	`< 1`	
Compatibility	Incompatible with existing models.	

:SENSe:MEASure:QANalysis:THReshold:SLOT <numeric>,<interface>[,<unit>]

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: `> :SENSe:MEASure:QANalysis:THReshold:SLOT 2,1`
Compatibility	Incompatible with existing models.

:SENSe:MEASure:QANalysis:THReshold: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 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>

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?

Response	<mode> = <CHARACTER RESPONSE DATA>
	SING, REP, UNT
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>

Parameter	<numeric> = <DECIMAL NUMERIC PROGRAM DATA>	
	2 to 1000	2 to 1000 times, in 1-time steps
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?

Response	<numeric> = <NR1 NUMERIC RESPONSE DATA>
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.

: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.

: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.

:SENSe:MEASure:QANalysis:THReshold:ASEarch <item>

Parameter	<item> = <CHARACTER PROGRAM DATA>
	OFF Auto search OFF
	THReshold Threshold auto search
	PTHReshold 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:THReshold:ASEarch THReshold`
Compatibility	Incompatible with existing models.

:SENSe:MEASure:QANalysis: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:QANalysis:THReshold:ASEarch?`
	`< THR`
Compatibility	Incompatible with existing models.

: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?

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.

:SENSe:MEASure:QANalysis:THReshold: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 Vth vs Q.
Example	To set to Top threshold search:
	`> :SENSe:MEASure:QANalysis:THReshold:ASEarch:PAMSet TOP`
Compatibility	Incompatible with existing models.

:SENSe:MEASure:QANalysis:THReshold:ASEarch:PAMSet?

Response	<item> = <CHARACTER RESPONSE DATA>
	OFF, TOP, MIDD, BOTT
Function	Queries the search target threshold for PAM Auto Search when measuring Vth vs Q.
Example	`> :SENSe:MEASure:QANalysis:THReshold:ASEarch:PAMSet?`
	`< TOP`
Compatibility	Incompatible with existing models.

7.12.14.2 Phase vs. Q measurement setting commands

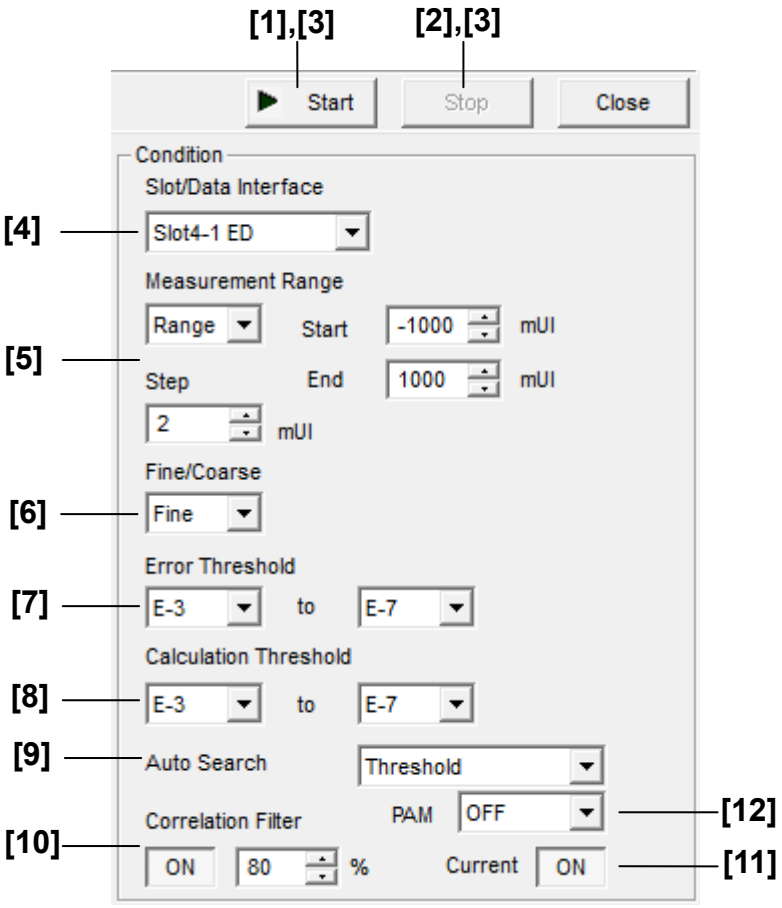


Figure 7.12.14.2-1 Phase vs. Q measurement setting field

Table 7.12.14.2-1 Phase vs. Q measurement setting commands

No.	Setting Items		Commands
[1]	Start		:SENSe:MEASure:QANalysis:PHASe:STARt
[2]	Stop		:SENSe:MEASure:QANalysis:PHASe:STOP
[3]	Query for measurement state		:SENSe:MEASure:QANalysis:STATe?
[4]	Slot		:SENSe:MEASure:QANalysis:PHASe:SLOT
			:SENSe:MEASure:QANalysis:PHASe:SLOT?
[5]	Measurement Range	Start	:SENSe:MEASure:QANalysis:PHASe:RANGe
		End	:SENSe:MEASure:QANalysis:PHASe:RANGe?
		Center	:SENSe:MEASure:QANalysis:PHASe:WIDTh
		Span	:SENSe:MEASure:QANalysis:PHASe:WIDTh?
[6]	Fine/Coarse		:SENSe:MEASure:QANalysis:PHASe:RESolution
			:SENSe:MEASure:QANalysis:PHASe:RESolution?
[7]	Error Threshold		:SENSe:MEASure:QANalysis:PHASe:ERANge
			:SENSe:MEASure:QANalysis:PHASe:ERANge?

Table 7.12.14.2-1 Phase vs. Q measurement setting commands (Cont'd)

No.	Setting Items	Commands
[8]	Calculation Threshold	:SENSe:MEASure:QANalysis:PHASe:CALRange
		:SENSe:MEASure:QANalysis:PHASe:CALRange?
[9]	Auto Search	:SENSe:MEASure:QANalysis:PHASe:ASEarch
		:SENSe:MEASure:QANalysis:PHASe:ASEarch?
[10]	Correlation Filter	:SENSe:MEASure:QANalysis:PHASe:CFILter
		:SENSe:MEASure:QANalysis:PHASe:CFILter?
[11]	Current	:SENSe:MEASure:QANalysis:PHASe:CURRent
		:SENSe:MEASure:QANalysis:PHASe:CURRent?
[12]	Selection of PAM search target threshold	:SENSe:MEASure:QANalysis:PHASe:ASEarch:PAMSet
		:SENSe:MEASure:QANalysis:PHASe:ASEarch:PAMSet?

: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>,<interface>[,<unit>]

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	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>]

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 Phase vs. Q measurement target slot.
Example	`> :SENSe:MEASure:QANalysis:PHASe:SLOT?`
	`< 2,1`
Compatibility	Incompatible with existing models.

:SENSe:MEASure:QANalysis:PHASe:RANGe <start>,<end>,<step>

Parameter
<start> = <DECIMAL NUMERIC PROGRAM DATA>
–1000 to 998 –1000 to 998mUI/2 mUI Step
<end> = <DECIMAL NUMERIC PROGRAM DATA>
–998 to 1000 –998 to 1000 mUI/2 mUI Step
<step> = <DECIMAL NUMERIC PROGRAM DATA>
2 to 200 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 –200 to 300 mUI and measurement step to 10 mUI:

```
> :SENSe:MEASure:QANalysis:PHASe:RANGe -200,300,10
```

Compatibility
Incompatible with existing models.

:SENSe:MEASure:QANalysis: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:QANalysis:PHASe:RANGe?
< -200,300,10
```

Compatibility
Incompatible with existing models.

:SENSe:MEASure:QANalysis:PHASe:WIDTh ,,<step>

Parameter
 = <DECIMAL NUMERIC PROGRAM DATA>
–998 to 998 –998 to 998 mUI/2 mUI Step
 = <DECIMAL NUMERIC PROGRAM DATA>
4 to 2000 4 to 2000 mUI/4 mUI Step
<step> = <DECIMAL NUMERIC PROGRAM DATA>
2 to 200 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 100 mUI, centered at 200 mUI, in 10-mUI steps:

```
> :SENSe:MEASure:QANalysis:PHASe:WIDTh 100,200,10
```

Compatibility
Incompatible with existing models.

:SENSe:MEASure:QANalysis:PHASe:WIDTh?

Response	= <NR1 NUMERIC RESPONSE DATA> = <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 <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.

: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 Omitted when <boolean> is "0" or "OFF".
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?

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	`> :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:	
	`> :SENSe:MEASure:QANalysis:PHASe:CURRent 1`	
Compatibility	Incompatible with existing models.	

:SENSe:MEASure:QANalysis: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:QANalysis:PHASe:CURRent?`	
	`< 1`	
Compatibility	Incompatible with existing models.	

:SENSe:MEASure:QANalysis: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:QANalysis:PHASe:ASEarch:PAMSet TOP`	
Compatibility	Incompatible with existing models.	

:SENSe:MEASure:QANalysis: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:QANalysis:PHASe:ASEarch:PAMSet?`	
	`< TOP`	
Compatibility	Incompatible with existing models.	

7.12.14.3 Graph display setting commands

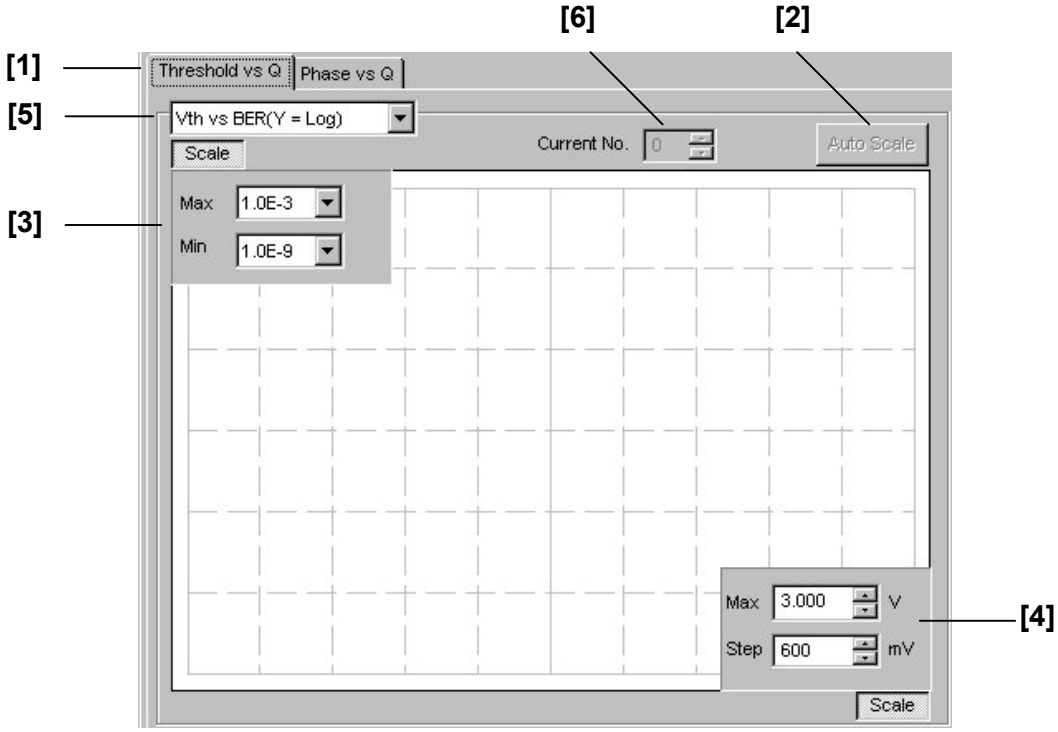


Figure 7.12.14.3-1 Graph display screen (Threshold vs Q tab)

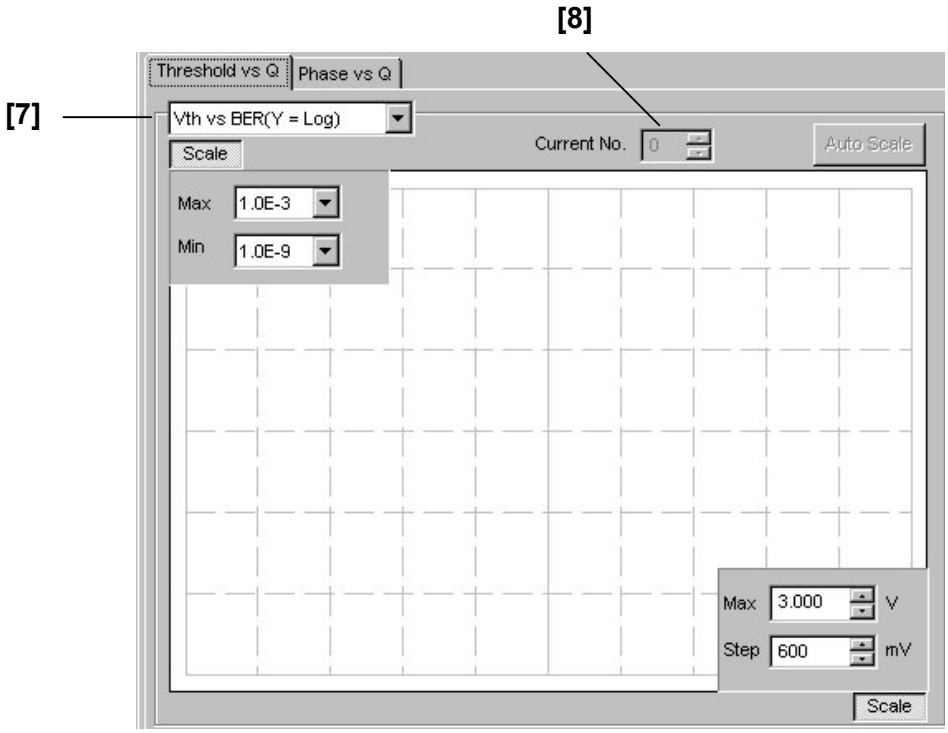


Figure 7.12.14.3-2 Graph display screen (Phase vs Q tab)

Table 7.12.14.3-1 Graph display setting commands

No.	Setting Items	Commands
[1]	Measurement mode switching	:DISPlay:RESult:QANalysis:MODE
		:DISPlay:RESult:QANalysis:MODE?
[2]	Auto Scale	:DISPlay:RESult:QANalysis:SCALe:ASCale
[3]	Max	:DISPlay:RESult:QANalysis:SCALe:VERTical
	Min	:DISPlay:RESult:QANalysis:SCALe:VERTical?
[4]	Max	:DISPlay:RESult:QANalysis:SCALe:HORizontal
	Step	:DISPlay:RESult:QANalysis:SCALe:HORizontal?
[5]	Threshold vs. Q	:DISPlay:RESult:QANalysis:THReshold:ITEM
	Display graph selection	:DISPlay:RESult:QANalysis:THReshold:ITEM?
[6]	Threshold vs. Q	:DISPlay:RESult:QANalysis:THReshold:MNUMber
	Current No.	:DISPlay:RESult:QANalysis:THReshold:MNUMber?
[7]	Phase vs. Q	:DISPlay:RESult:QANalysis:PHASe:ITEM
	Display graph selection	:DISPlay:RESult:QANalysis:PHASe:ITEM?
[8]	Phase vs. Q	:DISPlay:RESult:QANalysis:PHASe:MNUMber
	Current No.	:DISPlay:RESult:QANalysis:PHASe:MNUMber?

: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

Format	<max>	<step>	Remarks
Vth vs BER(Y = Log)	2 to 7 (E–2 to E–7), in single steps	Inputs the minimum value. 8 to 14 (E–8 to E–14), in single steps	A difference between <max> and <step> must be 7 or greater even number.
Vth vs BER(Y = Log(–Ln))	2 to 7 (E–2 to E–7), in single steps	Inputs the minimum value. 8 to 14 (E–8 to E–14), in single steps	A difference between <max> and <step> must be 7 or greater even number.
Vth vs Q	10 to 1000, in 10 steps	1 to 100, in single steps	When Linear
	–40 to 60, in 10 steps	1 to 11, in single steps	When Log
Times vs Q	10 to 1000, in 10 steps	1 to 100, in single steps	When Linear
	–40 to 60, in 10 steps	1 to 11, in single steps	When Log
Histogram	50 to 1000, in 50 steps	Omitted	

Table 7.12.14.3-3 Parameters for Phase vs. Q tab

Format	<max>	<step>	Remarks
Vth vs BER(Y = Log)	2 to 7 (E–2 to E–7), in single steps	Inputs the minimum value. 8 to 14 (E–8 to E–14), in single steps	A difference between <max> and <step> must be 7 or greater even number.
Vth vs BER(Y = Log(–Ln))	2 to 7 (E–2 to E–7), in single steps	Inputs the minimum value. 8 to 14 (E–8 to E–14), in single steps	A difference between <max> and <step> must be 7 or greater even number.
Vth vs Q	10 to 1000, in 10 steps	1 to 100, in single steps	When Linear
	–40 to 60, in 10 steps	1 to 11, in single steps	When Log
Phase vs Q	10 to 1000, in 10 steps	1 to 100, in single steps	When Linear
	–40 to 60, in 10 steps	1 to 11, in single steps	When Log
Phase vs σ	0.0010 to 1.0000, in 0.0010 steps	0.001 to 0.1000, 0.0001 steps	
Phase vs μ	–3.990 to 4.000 (V), in 0.010 V steps	0.001 to 0.800 (V), in 0.001 V steps	
Phase vs Opt BER	10 to 1 (E–10 to E–1), in single steps	Inputs the minimum value. 199 to 20 (E–199 to E–20), in single steps	
Phase vs Opt Threshold	–3.990 to 4.000 (V), in 0.010 V steps	0.001 to 0.800 (V), in 0.001 V steps	
Phase vs Correlation	10 to 100 (%), in 10 % steps	0 to 90 (%), in 10 % steps	A difference between <max> and <step> must be 10 or greater

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 7.12.14.3-4 Parameters for Threshold vs. Q tab

Format	<max>	<step>	Remarks
Vth vs BER(Y = Log)	–3.990 to 4.000 (V), in 0.010 V steps	0.001 to 0.800 (V), in 0.001 V steps	
Vth vs BER(Y = Log(–Ln))	–3.990 to 4.000 (V), in 0.010 V steps	0.001 to 0.800 (V), in 0.001 V steps	
Vth vs Q	–3.990 to 4.000 (V), in 0.010 V steps	0.001 to 0.800 (V), in 0.001 V steps	
Times vs Q	100 to 1000, in 50 steps	10 (fixed)	
Histogram	0.16 to 1000.00, in 0.01 steps	16/32/64/128/256	When Linear <max> must be greater than <step>.
	–49.84 to 60.00 (dB), in 0.01 steps	16/32/64	When Log

Table 7.12.14.3-5 Parameters for Phase vs. Q tab

Format	<max>	<step>	Remarks
Vth vs BER(Y = Log)	–3.990 to 4.000 (V), in 0.010 V steps	0.001 to 0.800 (V), in 0.001 V steps	
Vth vs BER(Y = Log(–Ln))	–3.990 to 4.000 (V), in 0.010 V steps	0.001 to 0.800 (V), in 0.001 V steps	
Vth vs Q	–3.990 to 4.000 (V), in 0.010 V steps	0.001 to 0.800 (V), in 0.001 V steps	
Phase vs Q	–900 to 1000 (mUI), in 100 mUI steps	10 to 200 (mUI), in 10 mUI steps	
Phase vs σ	–900 to 1000 (mUI), in 100 mUI steps	10 to 200 (mUI), in 10 mUI steps	
Phase vs μ	–900 to 1000 (mUI), in 100 mUI steps	10 to 200 (mUI), in 10 mUI steps	
Phase vs Opt BER	–900 to 1000 (mUI), in 100 mUI steps	10 to 200 (mUI), in 10 mUI steps	
Phase vs Opt Threshold	–900 to 1000 (mUI), in 100 mUI steps	10 to 200 (mUI), in 10 mUI steps	
Phase vs Correlation	–900 to 1000 (mUI), in 100 mUI steps	10 to 200 (mUI), in 10 mUI steps	

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
```

Compatibility Incompatible with existing models.

: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: `> :DISPlay:RESult:QANalysis: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	`> :DISPlay:RESult:QANalysis:THReshold:MNUMber?` `< 100`
Compatibility	Incompatible with existing models.

:DISPlay:RESult:QANalysis: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. σ
	PHAMu	Phase vs. μ
	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: `> :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

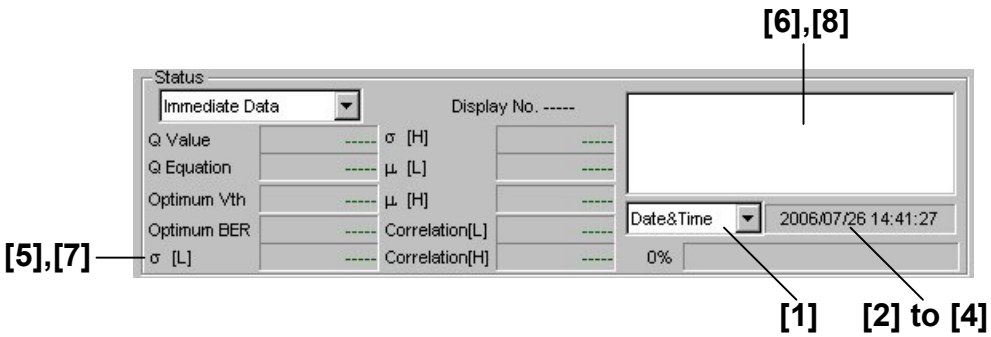


Figure 7.12.14.4-1 Measurement results and status display field

Table 7.12.14.4-1 Measurement results and status query commands

No.	Setting Items	Commands
[1]	No label (Setting of time display)	:SENSe:MEASure:QANalysis:TIME
		:SENSe:MEASure:QANalysis:TIME?
[2]	Query for current time	:SENSe:MEASure:QANalysis:DTIMe?
[3]	Query for measurement start time	:SENSe:MEASure:QANalysis:STARt?
[4]	Query for measurement elapsed time	:SENSe:MEASure:QANalysis:ELAPsed?
[5]	Query for Vth vs. Q measurement result	:CALCulate:DATA:QANalysis:THReshold?
[6]	Query for Vth vs. Q measurement status	:CALCulate:DATA:QANalysis:TSTatus?
[7]	Query for Phase vs. Q measurement result	:CALCulate:DATA:QANalysis:PHASe?
[8]	Query for Phase vs. Q measurement status	:CALCulate:DATA:QANalysis:PSTatus?

: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.	

: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.

:SENSe:MEASure:QANalysis:STARt?

Response

<year> = <NR1 NUMERIC RESPONSE DATA>
0, 2000 to 2036 Year 2000 to 2036
<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

```
> :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

```
> :SENSe:MEASure:QANalysis:ELAPsed?
< 31,23,59,59
```

Compatibility

Incompatible with existing models.

:CALCulate:DATA:QANalysis:THReshold? <string>

Parameter

<string> = <STRING PROGRAM DATA>

<result1> For the contents of measurement data <result1> see Table 7.12.14.4-2.

<result2> For the contents of immediate data <result2> see Table 7.12.14.4-3.

<result3> For the contents of statistic data <result3> see Table 7.12.14.4-4.

Table 7.12.14.4-2 Vth vs. Q measurement result <result1>

Items		<result1>	Response Format
Top side Vth/Error Rate	Number of measurements 1 to 1000	"VTHQ:TOP,1" … "VTHQ:TOP,1000"	Form1
Bottom side Vth/Error Rate		"VTHQ:BTM,1" … "VTHQ:BTM,1000"	Form1

The measurement result for each Vth is delimited with a comma (,) and returned.

Table 7.12.14.4-3 Vth vs. Q immediate data <result2>

Items		<result2>	Response Format
Q value	Number of measurements 1 to 1000	"QVALue:1"… "QVALue:1000"	Form2
Optimum BER		"OPTBer:1"… "OPTBer:1000"	Form3
Optimum Vth		"OPTVth:1"… "OPTVth:1000"	Form4
Correlation bottom		"CORBtm:1"… "CORBtm:1000"	Form2
Correlation top		"CORTop:1"… "CORTop:1000"	Form2
σ bottom		"SGMBtm:1"… "SGMBtm:1000"	Form2
σ top		"SGMTop:1"… "SGMTop:1000"	Form2
μ bottom		"MUBtm:1"… "MUBtm:1000"	Form4
μ top		"MUTop:1"… "MUTop:1000"	Form4
Q Equation		"QEQ:1"… "QEQ:1000"	Form2

Table 7.12.14.4-4 Vth vs. Q measurement statistic data <result3>

Items	<Result3>	Response Format
Total Data	"TOTal"	Form5
Valid Data	"VALid"	Form5
Q Max	"QMAX"	Form2
Q Min	"QMIN"	Form2
Q Mean	"QMEan"	Form2
Q σ	"QSGM"	Form2
Q-5 σ	"QSGM5"	Form2

Response <string> = <STRING RESPONSE DATA>

Table 7.12.14.4-5 Response format

Form	Format	Description
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 1.0000E000
	"--------"	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:

```
> :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> = <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 Vth vs. Q measurement status.

Example

```
> :CALCulate:DATA:QANalysis:TSTatus?
< "Sync Loss"
```

Compatibility

Incompatible with existing models.

: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>

Items		<result1>	Response Format
Top side Vth-Error Rate	Phase –1000 to 1000 mUI	"VTHQ:TOP,XXXXX"	Form1
Bottom side Vth-Error Rate		"VTHQ:BTM,XXXXX"	Form1

Table 7.12.14.4-7 Phase vs. Q immediate data <result2>

Items		<result2>	Response Format
Q value	Phase –1000 to 1000	"QVALue:–1000"… "QVALue:1000"	Form2
Optimum BER		"OPTBer:–1000"… "OPTBer:1000"	Form3
Optimum Vth		"OPTVth:–1000"… "OPTVth:1000"	Form4
Correlation bottom		"CORBtm:–1000"… "CORBtm:1000"	Form2
Correlation top		"CORTop:–1000"… "CORTop:1000"	Form2
Σ bottom		"SGMBtm:–1000"… "SGMBtm:1000"	Form2
Σ top		"SGMTop:–1000"… "SGMTop:1000"	Form2
M bottom		"MUBtm:–1000"… "MUBtm:1000"	Form4
M top		"MUTop:–1000"… "MUTop:1000"	Form4
Q equation		"QEQ:–1000"… "QEQ:1000"	Form2

Response

<string> = <STRING RESPONSE DATA>

Table 7.12.14.4-8 Response format

Form	Format	Description
Form1 Vth-ER 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 1.0000E000
	"--------"	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.

Function Queries the Phase vs. Q measurement result.

Example To query the TOP side measurement result:

```
> :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

```
> :CALCulate:DATA:QANalysis:PSTatus?
< "Sync Loss"
```

7.12.14.5 File menu setting commands

Table 7.12.14.5-1 File menu setting commands

No.	Setting Items	Commands
[1]	Open	:SYSTem:MMEMory:QANalysis:RECall
[2]	Save	:SYSTem:MMEMory:QANalysis:STORe
[3]	Print	:SYSTem:PRINt:QANalysis
[4]	Q Unit	:SENSe:MEASure:QANalysis:THReshold:LSCale
		:SENSe:MEASure:QANalysis:THReshold:LSCale?
		:SENSe:MEASure:QANalysis:PHASe:LSCale
		:SENSe:MEASure:QANalysis:PHASe:LSCale?

: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 <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: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:

```
> :SYSTem:PRINt:QANalysis VTQ
```

Compatibility

Incompatible with existing models.

:SENSe:MEASure:QANalysis:THReshold:LSCale <scale>

Parameter	<scale> = <CHARACTER PROGRAM DATA> LINear 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:QANalysis:THReshold:LSCale LOG`
Compatibility	Incompatible with existing models.

:SENSe:MEASure:QANalysis:THReshold:LSCale?

Response	<scale> = <CHARACTER RESPONSE DATA> LIN, LOG
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>

Parameter	<scale> = <CHARACTER PROGRAM DATA> LINear 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:QANalysis:PHASe:LSCale LOG`
Compatibility	Incompatible with existing models.

:SENSe:MEASure:QANalysis:PHASe:LSCale?

Response	<scale> = <CHARACTER RESPONSE DATA> LIN, LOG
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.

7.12.15 PAM BER measurement

This section describes PAM BER measurement commands.

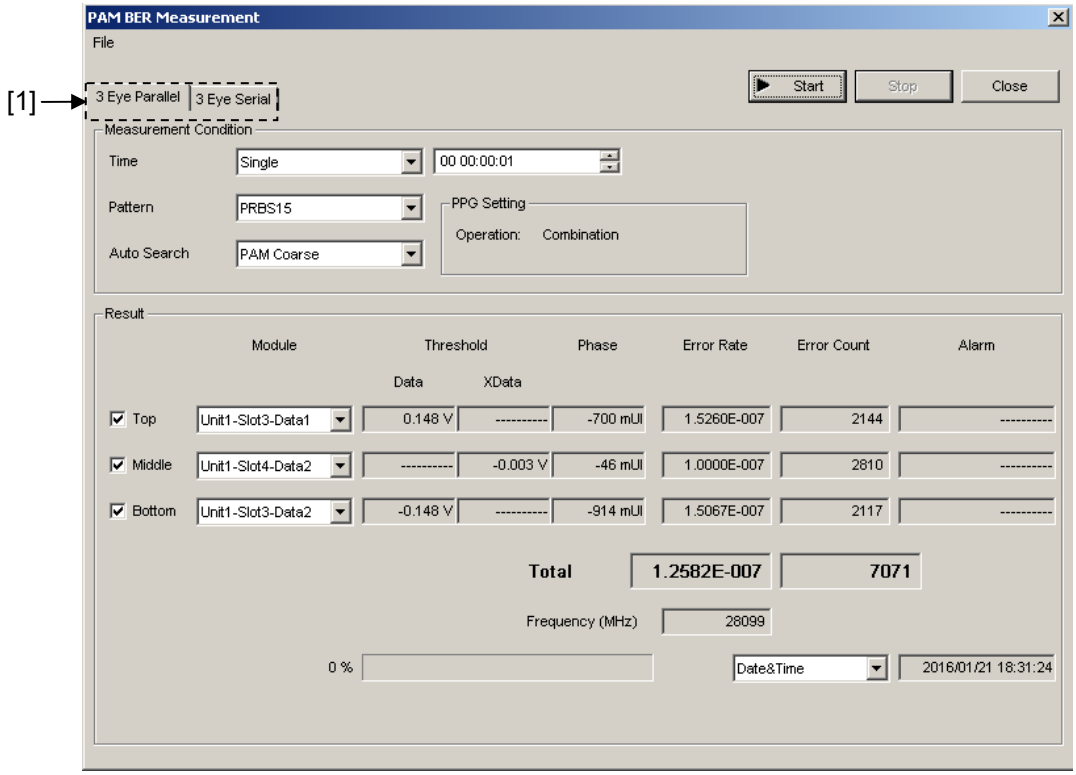


Figure 7.12.15-1 PAM BER Measurement Screen

Table 7.12.15-1 PAM BER Measurement Commands

No.	Setting Items	Commands
[1]	Tab Selection	:DISPlay:RESult:PAMBer:TABSelect
		:DISPlay:RESult:PAMBer:TABSelect?

:DISPlay:RESult:PAMBer:TABSelect <tab>

Parameter	<tab> = <CHARACTER PROGRAM DATA> PBSerial Selects the 3 Eye Serial tab. PBParallel Selects the 3 Eye Parallel tab.
Function	Selects the PAM BER measurement type. Selects serial measurement using 3 Eye Serial or the 3 Eye Parallel measurement tab using 3ch.
Example	To select the 3Lvl Serial tab: `> :DISPlay:RESult:PAMBerTABSelect PBSerial`
Compatibility	Incompatible with existing models.

:DISPlay:RESult:PAMBer:TABSelect?

Response	<tab> = <CHARACTER RESPONSE DATA> PBS, PBP
Function	Queries the tab displayed for PAM BER measurement.
Example	`> :DISPlay:RESult:PAMBerTABSelect?` `< PBS`
Compatibility	Incompatible with existing models.

7.12.15.1 3 Eye Serial Measurement Commands

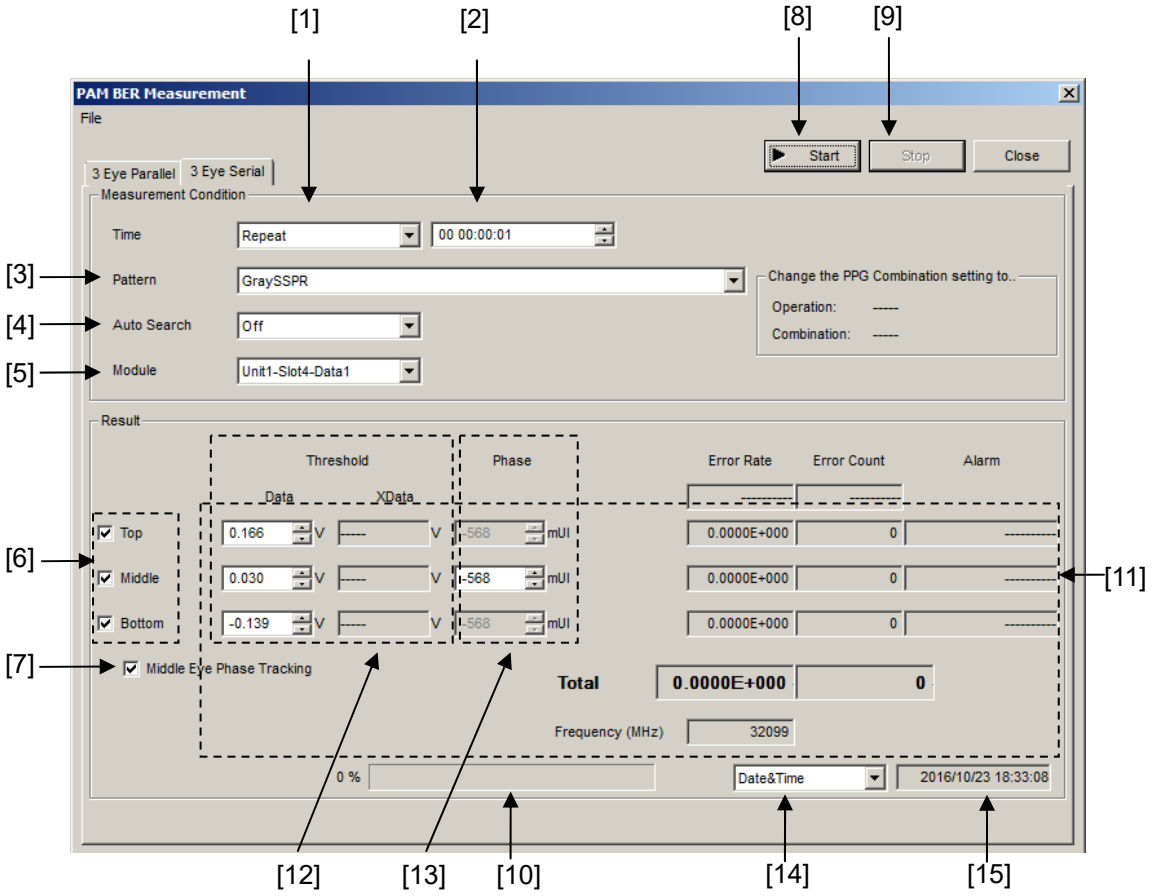


Figure 7.12.15.1-1 3 Eye Serial Measurement Screen

Table 7.12.15.1-1 3 Eye Serial Measurement Commands

No.	Setting Items	Commands
[1]	Mode	:SENSe:MEASure:PBSerial:MODE
		:SENSe:MEASure:PBSerial:MODE?
[2]	Time	:SENSe:MEASure:PBSerial:PERiod
		:SENSe:MEASure:PBSerial:PERiod?
[3]	Pattern	:SENSe:PATTern:PBSerial:TYPE
		:SENSe:PATTern:PBSerial:TYPE?
[4]	Auto Search	:SENSe:MEASure:PBSerial:ASEarch
		:SENSe:MEASure:PBSerial:ASEarch?
[5]	Module	:SENSe:MEASure:PBSerial:SLOT
		:SENSe:MEASure:PBSerial:SLOT?
[6]	Measurement PAM Threshold	:SENSe:MEASure:PBSerial:PAMSet
		:SENSe:MEASure:PBSerial:PAMSet?
[7]	Middle Eye Phase Tracking	:SENSe:MEASure:PBSerial:MEPHase
		:SENSe:MEASure:PBSerial:MEPHase?

Table 7.12.15.1-1 3 Eye Serial Measurement Commands (Cont'd)

No.	Setting Items	Commands
[8]	Start	:SENSe:MEASure:PBSerial:STARt
[9]	Stop	:SENSe:MEASure:PBSerial:STOP
[10]	State	:SENSe:MEASure:PBSerial:STATe?
[11]	Result	:CALCulate:DATA:PBSerial?
[12]	Threshold	:SENSe:MEASure:PBSerial:DATA:THReshold
		:SENSe:MEASure:PBSerial:DATA:THReshold?
[13]	Delay	:SENSe:MEASure:PBSerial:CLOCk:DELay
		:SENSe:MEASure:PBSerial:CLOCk:DELay?
[14]	Display time selection	:SENSe:MEASure:PBSerial:TIME
		:SENSe:MEASure:PBSerial:TIME?
[15]	Query for current time	:SENSe:MEASure:PBSerial:TIME:DTIMe?
	Query for measurement start time	:SENSe:MEASure:PBSerial:TIME:STARt?
	Query for measurement elapsed time	:SENSe:MEASure:PBSerial:TIME:ELAPsed?

: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:

```
> :SENSe:MEASure:PBSerial:MODE REPeat
```

Compatibility Incompatible with existing models.

:SENSe:MEASure:PBSerial:MODE?

Response <mode> = <CHARACTER RESPONSE DATA>

REP, SING

Function Queries the measurement repeat setting.

Example

```
> :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.	

: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>]

Parameter
<slot> = <DECIMAL 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
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>]

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 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 target slot and data interface for 3 Eye Serial PAM BER measurement.

Example

```
> :SENSe:MEASure:PBSerial:SLOT?
< 4,1
```

Compatibility
Incompatible with existing models.

: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>

Response	<boolean> = <NR1 NUMERIC RESPONSE DATA> 1 ON 0 OFF
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?

Response	<numeric> = <NR1 NUMERIC RESPONSE DATA> 0 Measurement stopped. 1 Measurement in progress
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

<string> = <STRING PROGRAM DATA>
"TOP:<result>"
"MIDDle:<result>"
"BOTTom:<result>"
"TOTal:<result>"

Table 7.12.15.1-2 3 Eye Serial Measurement Results <result>

Item	<result>	Response Format
Error Count	EC	Form1
Error Rate	ER	Form2
Frequency(Hz)	FREQuency	Form3
Data Threshold	THReshold	Form4
Data Phase	PHASe	Form5

Response

<string> = <STRING RESPONSE DATA>

Table 7.12.15.1-3 Response Format

Type	Format	Description
Form1 Integer type	"XXXXXXX"	0 to 9999999
	"X.XXXEXX"	1.0000E07 to 9.9999E17
	"-----"	When no data corresponds to the query
Form2 Fraction type	"X.XXXE-XX"	0.0001E–18 to 1.0000E00
	"-----"	When no data corresponds to the query
Form3 Frequency type	"XXXXXXXXX"	For 0 to MAX (Hz)
	"-----"	When no data corresponds to the query
Form4 Voltage type	"X.XXX,XX.XXX"	For –4.000 to +4.000
	"-----"	When no data corresponds to the query
Form5 Phase type	"XXXX"	For –1000 to +1000, –10000 to +10000
	"-----"	When no data corresponds to the query

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:	
	`> :SENSe:MEASure:PBSerial:DATA:THReshold? TOP,DATA`	
	`< 0.100`	
Compatibility	Incompatible with existing models.	

:SENSe:MEASure:PBSerial:CLOCk:DELay <threshold>,<numeric>

Parameter	<threshold> = <CHARACTER PROGRAM DATA>	
	TOP	Top level
	MIDDle	Middle level
	BOTTom	Bottom level
	<numeric> = <DECIMAL NUMERIC PROGRAM DATA>	
	–1000 to 1000	–1000 to 1000 mUI/2 mUI Step
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>

Parameter	<threshold> = <CHARACTER PROGRAM DATA>	
	TOP	Top level
	MIDDle	Middle level
	BOTTom	Bottom level
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	
	`> :SENSe:MEASure:PBSerial:CLOCk:DELay? TOP`	
	`< 300`	
Compatibility	Incompatible with existing models.	

:SENSe:MEASure:PBSerial: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 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.	

: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.

:SENSe:MEASure:PBSerial:TIME:STARt?

Response	<year>,<month>,<day>,<hour>,<min>,<second>
	<year> = <NR1 NUMERIC RESPONSE DATA>
	2000 to 2036 Year 2000 to 2036
	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 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	<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 Serial PAM BER measurement.
Example	`> :SENSe:MEASure:PBSerial:TIME:ELAPsed?`
	`< 23,59,59`
Compatibility	Incompatible with existing models.

7.12.15.2 3 Eye Parallel Measurement Commands

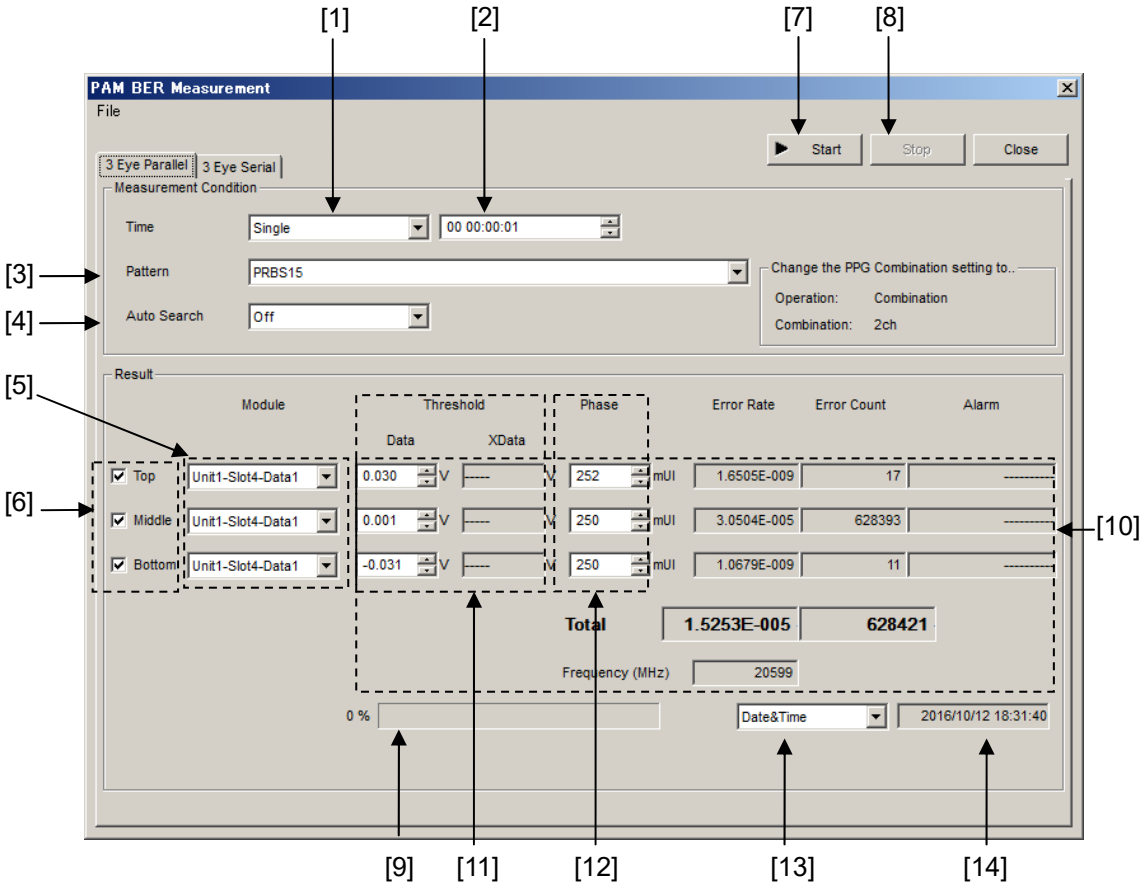


Figure 7.12.15.2-1 3 Eye Parallel Measurement Screen

Table 7.12.15.2-1 3 Eye Parallel Measurement Commands

No.	Setting Items	Commands
[1]	Mode	:SENSe:MEASure:PBParallel:MODE
		:SENSe:MEASure:PBParallel:MODE?
[2]	Time	:SENSe:MEASure:PBParallel:PERiod
		:SENSe:MEASure:PBParallel:PERiod?
[3]	Pattern	:SENSe:PATTern:PBParallel:TYPE
		:SENSe:PATTern:PBParallel:TYPE?
[4]	Auto Search	:SENSe:MEASure:PBParallel:ASEarch
		:SENSe:MEASure:PBParallel:ASEarch?
[5]	Module	:SENSe:MEASure:PBParallel:SLOT
		:SENSe:MEASure:PBParallel:SLOT?
[6]	Measurement PAM Threshold	:SENSe:MEASure:PBParallel:PAMSet
		:SENSe:MEASure:PBParallel:PAMSet?
[7]	Start	:SENSe:MEASure:PBParallel:STARt
[8]	Stop	:SENSe:MEASure:PBParallel:STOP
[9]	State	:SENSe:MEASure:PBParallel:STATe?

Table 7.12.15.2-1 3 Eye Parallel Measurement Commands (Cont'd)

No.	Setting Items	Commands
[10]	Result	:CALCulate:DATA:PBParallel?
[11]	Threshold	:SENSe:MEASure:PBParallel:DATA:THReshold
		:SENSe:MEASure:PBParallel:DATA:THReshold?
[12]	Delay	:SENSe:MEASure:PBParallel:CLOCk:DELay
		:SENSe:MEASure:PBParallel:CLOCk:DELay?
[13]	Display time selection	:SENSe:MEASure:PBParallel:TIME
		:SENSe:MEASure:PBParallel:TIME?
[14]	Query for current time	:SENSe:MEASure:PBParallel:TIME:DTIMe?
	Query for measurement start time	:SENSe:MEASure:PBParallel:TIME:STARt?
	Query for measurement elapsed time	:SENSe:MEASure:PBParallel:TIME:ELAPsed?

: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

```
> :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.

:SENSe:MEASure:PBParallel: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 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	<item> = <CHARACTER RESPONSE DATA> OFF, PAMF, PAMC
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	<threshold> = <CHARACTER PROGRAM DATA>	
	TOP	Top threshold
	MIDDle	Middle threshold
	BOTTom	Bottom threshold
	<slot> = <DECIMAL 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	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.

: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>

Items	<result>	Response Format
Error Count	EC	Form1
Error Rate	ER	Form2
Frequency (Hz)	FREQuency	Form3
Data Threshold	THReshold	Form4
Data Phase	PHASe	Form5

Response

<string> = <STRING RESPONSE DATA>

Table 7.12.15.2-3 Response Format

Form	Format	Description
Form1 Integer type	"XXXXXXX"	0 to 9999999
	"X.XXXEXX"	1.0000E07 to 9.9999E17
	"-----"	When no data corresponds to the query
Form2 Fraction type	"X.XXXE-XX"	0.0001E–18 to 1.0000E00
	"-----"	When no data corresponds to the query
Form3 Frequency type	"XXXXXXXXX"	For 0 to MAX (Hz)
	"-----"	When no data corresponds to the query
Form4 Voltage type	"X.XXX,XX.XXX"	For –4.000 to +4.000
	"-----"	When no data corresponds to the query
Form5 Phase type	"XXXX,XXXX"	For –1000 to +1000, –10000 to +10000
	"-----"	When no data corresponds to the query

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>

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 Parallel PAM BER measurement.	
Example	To set the measurement threshold to Data input, 0.100 V: `> :SENSe:MEASure:PBParallel:DATA:THReshold TOP,DATA,0.100`	
Compatibility	Incompatible with existing models.	

:SENSe:MEASure:PBParallel: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 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:
	`> :SENSe:MEASure:PBParallel:DATA:THReshold? TOP,DATA`
	`< 0.100`	

Compatibility	Incompatible with existing models.	

:SENSe:MEASure:PBParallel:CLOCk:DELay <threshold>,<numeric>

Parameter	<threshold> = <CHARACTER PROGRAM DATA>	
	TOP	Top level
	MIDDle	Middle level
	BOTTom	Bottom level
	<numeric> = <DECIMAL NUMERIC PROGRAM DATA>	
	–1000 to 1000	–1000 to 1000 mUI/2 mUI Step
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.	

:SENSe:MEASure:PBParallel:CLOCk:DELay? <threshold>

Parameter	<threshold> = <CHARACTER PROGRAM DATA>	
	TOP	Top level
	MIDDle	Middle level
	BOTTom	Bottom level
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 Parallel PAM BER measurement.	
Example	To query the measurement threshold of Top level, data input:	
	`> :SENSe:MEASure:PBParallel:CLOCk:DELay? TOP`	
	`< 300`	
Compatibility	Incompatible with existing models.	

:SENSe:MEASure:PBParallel: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 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.	

: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	<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	<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

No.	Setting Items	Commands
[1]	3 Eye serial measurement file open	:SYSTem:MMEMory:PBSerial:RECall
[2]	3 Eye serial measurement file save	:SYSTem:MMEMory:PBSerial:STORe
[3]	3 Eye parallel measurement file open	:SYSTem:MMEMory:PBParallel:RECall
[4]	3 Eye parallel measurement file save	:SYSTem:MMEMory:PBParallel:STORe

: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.

:SYSTem:MMEMory: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:MMEMory:PBParallel:RECall "C:\Test\example"`
Compatibility	Incompatible with existing models.

:SYSTem:MMEMory: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:MMEMory: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

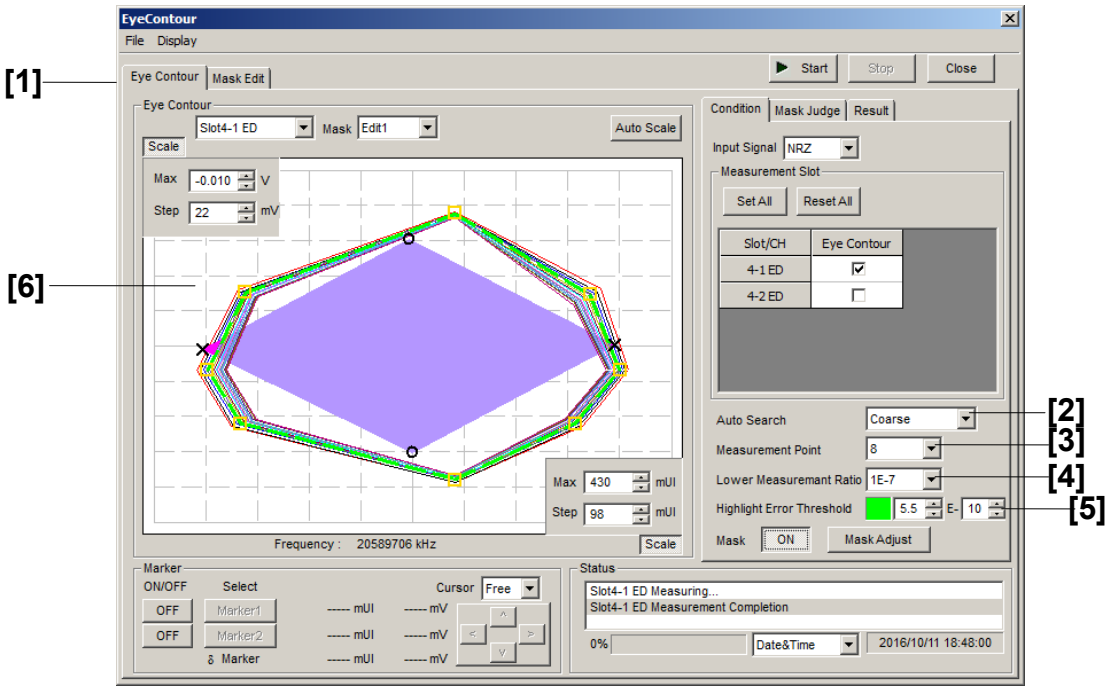


Figure 7.12.16.1-1 Eye Contour Measurement Screen

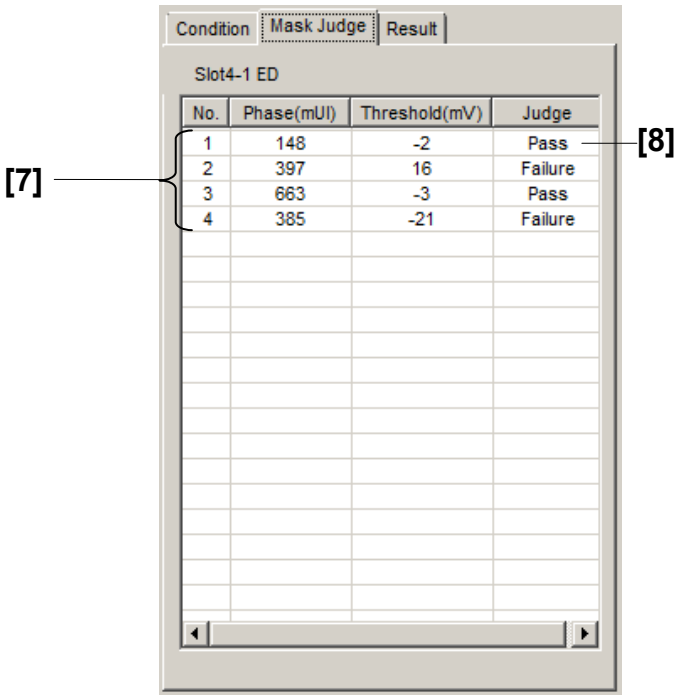


Figure 7.12.16.1-2 Eye Contour Mask Judge Screen

Table 7.12.16.1-1 Eye Contour Tab Setting Commands

No.	Setting Items	Commands
[1]	Tab Selection	:DISPlay:RESult:EDIagram:TABSelect
		:DISPlay:RESult:EDIagram:TABSelect?
[2]	Auto Search	:SENSe:MEASure:EDIagram:DIAGram:ASEarch
		:SENSe:MEASure:EDIagram:DIAGram:ASEarch?
[3]	Measurement Point	:SENSe:MEASure:EDIagram:DIAGram:TYPE
		:SENSe:MEASure:EDIagram:DIAGram:TYPE?
[4]	Lower Measurement Ratio	:SENSe:MEASure:EDIagram:DIAGram:LMRatio
		:SENSe:MEASure:EDIagram:DIAGram:LMRatio?
[5]	Highlight Error Threshold	:DISPlay:RESult:EDIagram:HETHreshold
		:DISPlay:RESult:EDIagram:HETHreshold?
[6]	Query for measurement results	:CALCulate:DATA:DIAGram?
[7]	Query Mask point number	:CALCulate:DATA:EDIagram:DETail:PNUMber?
[8]	Query Mask Judge results	:CALCulate:DATA:EDIagram:DETail:POINt?

: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?

Response	<setting> = <DECIMAL NUMERIC RESPONSE DATA> 0 OFF 1 ON (Coarse) 3 ON (PAM Coarse) 4 ON (PAM4 Fine)
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>

Parameter	<type> = <CHARACTER PROGRAM DATA> POINt4 4 Point POINt8 8 Point POINt16 16 Point
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:TYPE?

Response	<type> = <CHARACTER RESPONSE DATA> POIN4, POIN8, POIN16
Function	Queries the number of measurement points for Eye Diagram measurement.
Example	`> :SENSe:MEASure:EDIagram:DIAGram:TYPE?` `< POIN8`
Compatibility	Some parameters are compatible with Eye Diagram measurement (32G).

:SENSe:MEASure:EDIagram:DIAGram:LMRatio <range>

Parameter	<range> = <CHARACTER PROGRAM DATA> E_7 to E_12 E–7 to E–12
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.

:SENSe:MEASure:EDIagram:DIAGram:LMRatio?

Response	<range> = <CHARACTER RESPONSE DATA>
	E_7 to E_12 E–7 to E–12
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>

Parameter	<range> = <CHARACTER PROGRAM DATA>
	E_6 to E_20 E–6 to E–20
	<numeric1> = <DECIMAL NUMERIC PROGRAM DATA>
	1.0 to 9.9 Mantissa
	Sets two separated by a comma (,).
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?

Response	<range> = <CHARACTER RESPONSE DATA>
	E_6 to E_20 E–6 to E–20
	<numeric1> = <DECIMAL NUMERIC RESPONSE DATA>
	1.0 to 9.9 Mantissa
	Returns two separated by a comma (,).
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 7.12.16.1-2 Examples specifying a measurement point

Error rate	<string>
1E–6	"MAP:E_6:POINt1"
1E–7	"MAP:E_7:POINt1
:	:
1E–19	"MAP:E_19:POINt1"
1E–20	"MAP:E_20:POINt1"
Highlight Error Threshold	"MAP:HET:POINt1"

[<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 7.12.16.1-3.

Table 7.12.16.1-3 Response Format

Form	Format	Description
Form1 Eye Map type	"XXXXX,YYYYY"	XXXXX: Phase (mUI) XXXXX: Phase (ps) (Displays in the selected unit.) YYYYY: Threshold (mV)
	"-----,-----"	When no data corresponds to the query

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).

: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

Form	Format	Description
Form1 Eye Map type	"XXXXX,YYYYY,ZZ"	XXXXX: Phase (mUI) XXXXX: Phase (ps) ***Note:*** Displays in the specified unit. YY.YYY : Threshold (mV) ZZ : OK, NG
	"-----,--.---,--"	When no data corresponds to the query

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>

Items		<result>	Response Format
Judgment type	Point1 Point2 | Point31 Point32	"JDG:POINt1" "JDG:POINt2" | "JDG:POINt31" "JDG:POINt32"	Form1

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

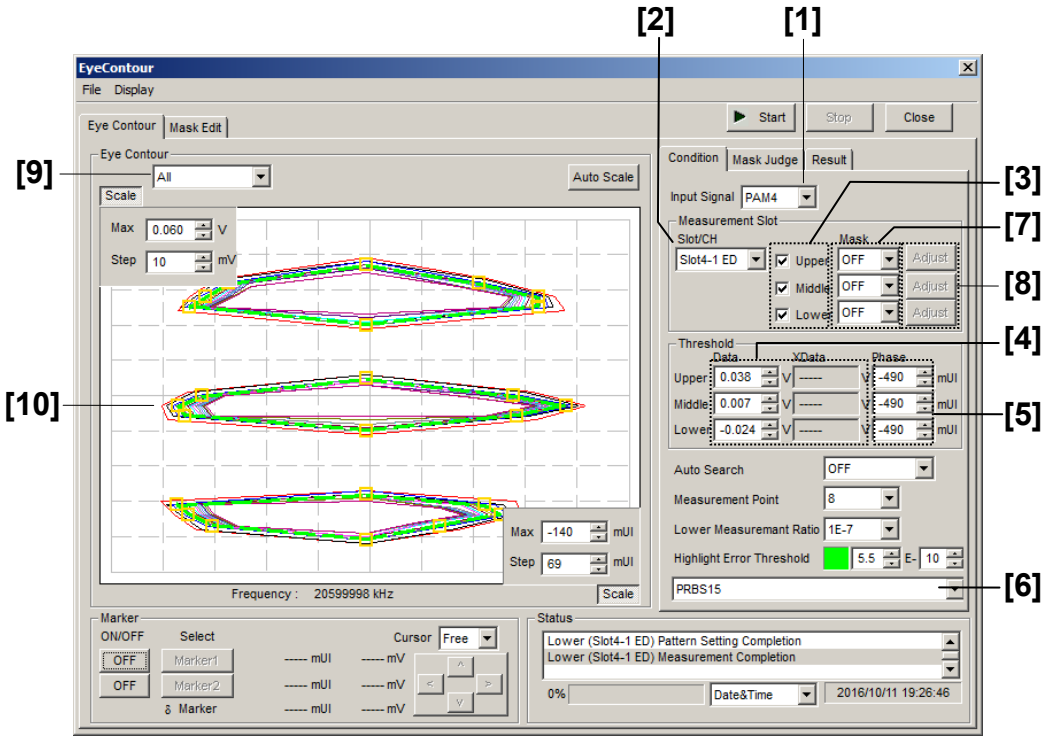


Figure 7.12.16.2-1 PAM4 Eye Contour Measurement Screen

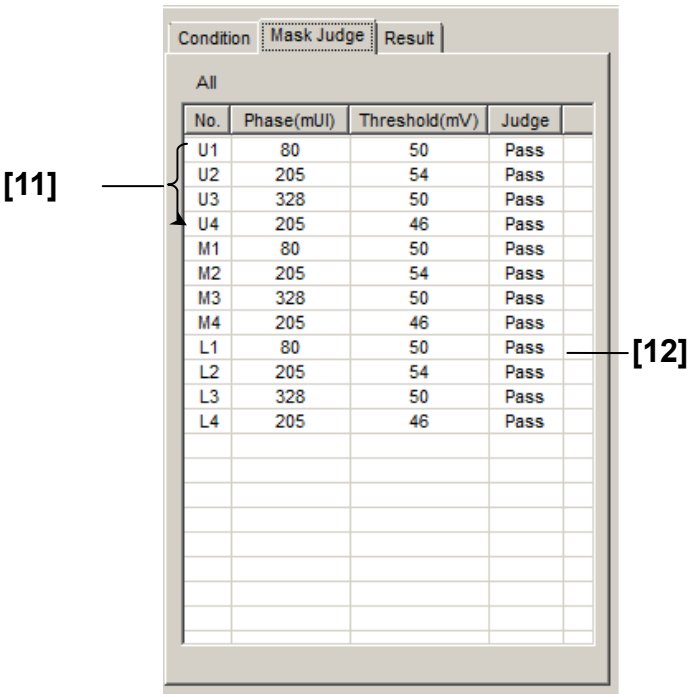


Figure 7.12.16.2-2 PAM4 Eye Contour Mask Judge Screen

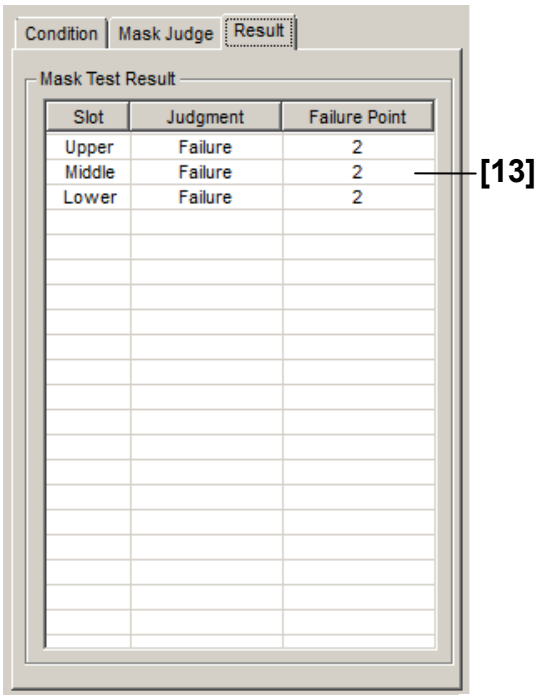


Figure 7.12.16.2-3 PAM4 Eye Contour Result Screen

Table 7.12.16.2-1 PAM4 Eye Contour Tab Setting Commands

No.	Setting Items	Commands
[1]	Input Signal	:SENSe:MEASure:EDIagram:DIAGram:ISIGnal
		:SENSe:MEASure:EDIagram:DIAGram:ISIGnal?
[2]	Selection of slot(s) and channel(s) to be measured	:SENSe:MEASure:EDIagram:DIAGram:PAM:SLOT
		:SENSe:MEASure:EDIagram:DIAGram:PAM:SLOT?
[3]	Selection of threshold(s) to be measured (Upper/Middle/Lower)	:SENSe:MEASure:EDIagram:DIAGram:PAM:PAMSet
		:SENSe:MEASure:EDIagram:DIAGram:PAM:PAMSet?
[4]	Threshold	:SENSe:MEASure:EDIagram:DIAGram:PAM:THReshold
		:SENSe:MEASure:EDIagram:DIAGram:PAM:THReshold?
[5]	Phase	:SENSe:MEASure:EDIagram:DIAGram:PAM:PHASe
		:SENSe:MEASure:EDIagram:DIAGram:PAM:PHASe?
[6]	Selection of pattern	:SENSe:MEASure:EDIagram:DIAGram:PAM:TYPE
		:SENSe:MEASure:EDIagram:DIAGram:PAM:TYPE?
[7]	Selection of mask(s)	:DISPlay:RESult:EDIagram:TEMPlate:PAM:SELect
		:DISPlay:RESult:EDIagram:TEMPlate:PAM:SELect?
[8]	Mask Adjust	:DISPlay:RESult:EDIagram:PAM:ADJust
[9]	Selection of threshold(s) to be displayed on PAM4 measurement result graph	:DISPlay:RESult:EDIagram:PAM:THReshold
		:DISPlay:RESult:EDIagram:PAM:THReshold?
[10]	Query for measurement results	:CALCulate:DATA:DIAGram:PAM4?
[11]	Query Mask point number	:CALCulate:DATA:EDIagram:DETail:PNUMber:PAM4?
[12]	Query Mask Judge results	:CALCulate:DATA:EDIagram:DETail:POINt:PAM4?
[13]	Query for Mask Test result	:CALCulate:DATA:EDIagram:MASK:PAM4?

: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.

:SENSe:MEASure:EDIagram:DIAGram:PAM: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
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>

Parameter	<thre> = <CHARACTER PROGRAM DATA>	
	UPPer	Upper threshold
	MIDDle	Middle threshold
	LOWer	Lower threshold
Response	<boolean> = <NR1 NUMERIC RESPONSE DATA>	
	0	OFF
	1	ON
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>

Parameter	<thre> = <CHARACTER PROGRAM DATA>	
	UPPer	Upper threshold
	MIDDle	Middle threshold
	LOWer	Lower threshold
	<port> = <CHARACTER PROGRAM DATA>	
	DATA	Data
	XDATa	XData
	<numeric> = <DECIMAL NUMERIC PROGRAM DATA>	
	–3.500 to 3.300	–3.500 to 3.300 V/0.001 V Step
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.	

:SENSe:MEASure:EDIagram:DIAGram:PAM:THReshold? <thre>,<port>

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	–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	–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.	

:SENSe:MEASure:EDIagram:DIAGram:PAM:PHASe? <thre>

Parameter	<thre> = <CHARACTER PROGRAM DATA>	
	UPPer	Upper threshold
	MIDDle	Middle threshold
	LOWer	Lower threshold
Response	<numeric> = <NR1 NUMERIC RESPONSE DATA>	
	–1000 to 1000	–1000 to 1000 mUI/2 mUI Step
Function	Queries the phase at which PAM4 Eye Contour measurement starts.	
Example	To query the phase set for the Upper threshold at which PAM4 Eye Contour measurement starts.	
	`> :SENSe:MEASure:EDIagram:DIAGram:PAM:PHASe? UPPer`	
	`< 300`	
Compatibility	Incompatible with existing models.	

:SENSe:MEASure:EDIagram:DIAGram:PAM: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
	"PRQS10"	PRQS10
	"SSPR"	SSPR
	"QPRBS13-CEI"	QPRBS13-CEI
Function	Sets the test pattern for PAM4 Eye Contour measurement.	
Example	To set "QPRBS13-CEI" for the test pattern:	
	`> :SENSe:MEASure:EDIagram:DIAGram:PAM:TYPE "QPRBS13-CEI"`	
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.	

:DISPlay:RESult:EDIagram:TEMPlate:PAM:SELect? <thre>

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:
	`> :DISPlay:RESult:EDIagram:TEMPlate:PAM:SELect? LOWer`
	`< E3`	

Compatibility	Incompatible with existing models.	

: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? <thresholde>,<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 7.12.16.2-2 Examples specifying a measurement point

Error rate	<string>
1E–6	"MAP:E_6:POINt1"
1E–7	"MAP:E_7:POINt1
:	:
1E–19	"MAP:E_19:POINt1"
1E–20	"MAP:E_20:POINt1"
Highlight Error Threshold	"MAP:HET:POINt1"

[<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 7.12.16.2-3 Response Format

Form	Format	Description
Form1 Eye Map type	"XXXXX,YYYYY"	XXXXX: Phase (mUI) XXXXX: Phase (ps) (Displays in the selected unit.) YYYYY: Threshold (mV)
	"-----,-----"	When no data corresponds to the query

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:

```
>:CALCulate:DATA:DIAGram? UPPer,"MAP:HET:POINt1"
<" 100,1000"
```

Compatibility

Incompatible with existing models.

:CALCulate:DATA:EDIagram:DETail:PNUMber:PAM4? <threshold>

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	<numeric> = <NR1 NUMERIC RESPONSE DATA>	
	0 to 32	Result point number
Function	Queries the result point number of Mask Judge for Middle Eye.	
Example	`> :CALCulate:DATA:EDIagram:DETail:PNUMber:PAM4? MIDDle`	
	`< 6`	
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>

<result>	Measurement data

For the format of <result>, refer to Table 7.12.16.2-5.

Response

<string> = <STRING RESPONSE DATA>

Table 7.12.16.2-4 Response Format

Form	Format	Description
Form1 Eye Map type	"XXXXX,YYYYY,ZZ"	XXXXX: Phase (mUI) XXXXX: Phase (ps) ***Note:*** Displays in the specified unit. YY.YYY: Threshold (mV) ZZ: Pass, Failure
	"-----,--.---,--"	When no data corresponds to the query

Function

Queries the measurement results of the Eye Contour Mask Judge measurement.

Table 7.12.16.2-5 Eye Contour Mask Judge Measurement Item <result>

Items		<result>	Response Format
Judgment type	Point1 Point2 | Point31 Point32	"JDG:POINt1" "JDG:POINt2" | "JDG:POINt31" "JDG:POINt32"	Form1

Example

To query the Point 1 measurement result of Mask Judge for Middle Eye Contour:

```
> :CALCulate:DATA:EDIagram:DETail:POINt:PAM4?
MIDDle,"JDG:POINt1"
< "    0, 4000,Pass"
```

Compatibility

Incompatible with existing models.

:CALCulate:DATA:EDIagram:MASK:PAM4? <threshold>

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 7.12.16.2-6 Response Format

Form	Format	Description
Form1 EyeResult type	"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:

```
>:CALCulate:DATA:EDIagram:MASK:PAM4? Upper
<"Failure,1,2,3,4,32"
<"Pass"
```

Compatibility

Incompatible with existing models.

7.12.16.3 Compatible Commands with Eye Diagram Measurement

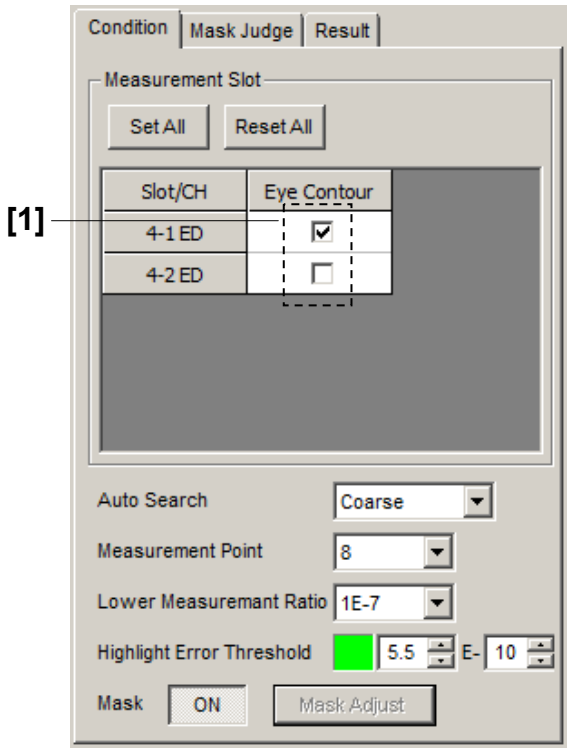


Figure 7.12.16.3-1 Condition Tab

Table 7.12.16.3-1 Condition Tab Setting Commands

No.	Setting Items	Commands
[1]	Slot Eye Diagram	:SENSe:MEASure:EDIagram:DIAGram:SLOT
		:SENSe:MEASure:EDIagram:DIAGram:SLOT?

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

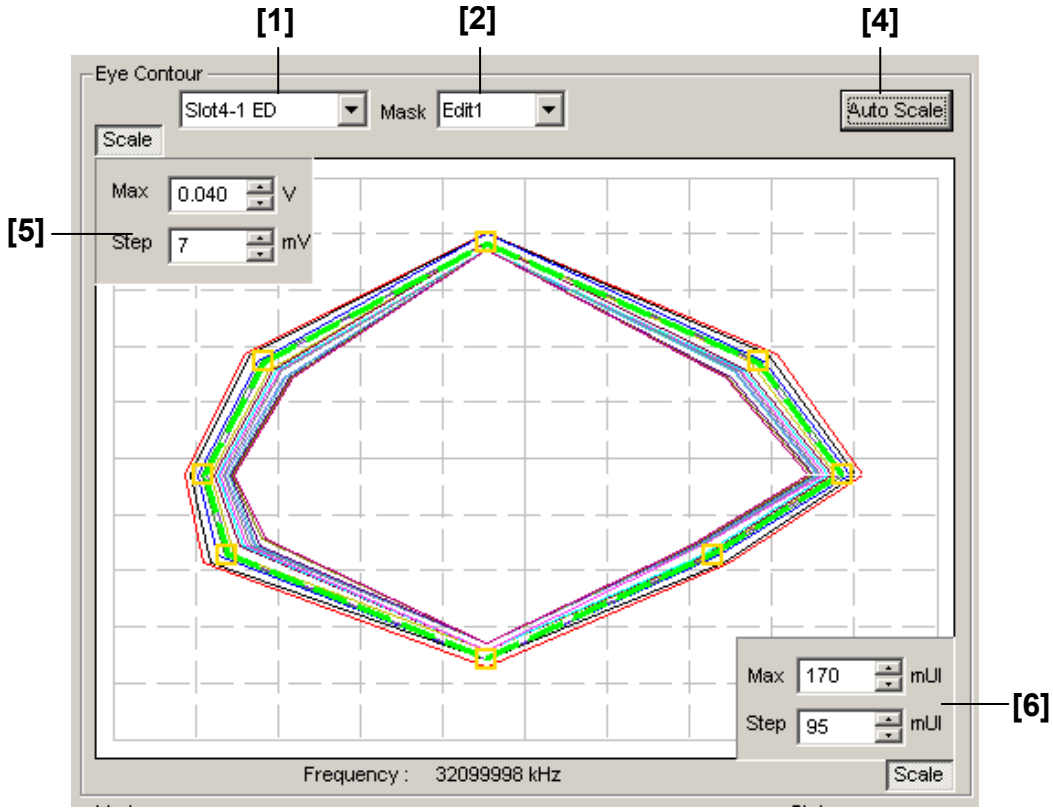


Figure 7.12.16.3-2 Graph Display Field

Table 7.12.16.3-2 Graph Display Setting Commands

No.	Setting Items	Commands
[1]	No label (Result display slot setting)	:DISPlay:RESult:EDIagram:SLOT
		:DISPlay:RESult:EDIagram:SLOT?
[2]	Mask	:DISPlay:RESult:EDIagram:TEMPlate:SELect
		:DISPlay:RESult:EDIagram:TEMPlate:SELect?
[4]	Auto Scale	:DISPlay:RESult:EDIagram:SCALe:ASCale
[5]	Max/Step	:DISPlay:RESult:EDIagram:SCALe:VOLTage
		:DISPlay:RESult:EDIagram:SCALe:VOLTage?
[6]	Max/Step	:DISPlay:RESult:EDIagram:SCALe:PHASe
		:DISPlay:RESult:EDIagram:SCALe:PHASe?

For commands description, refer to 7.12.10.2.1 “Graph display setting commands”.

7.12.16.3.1 Condition setting commands

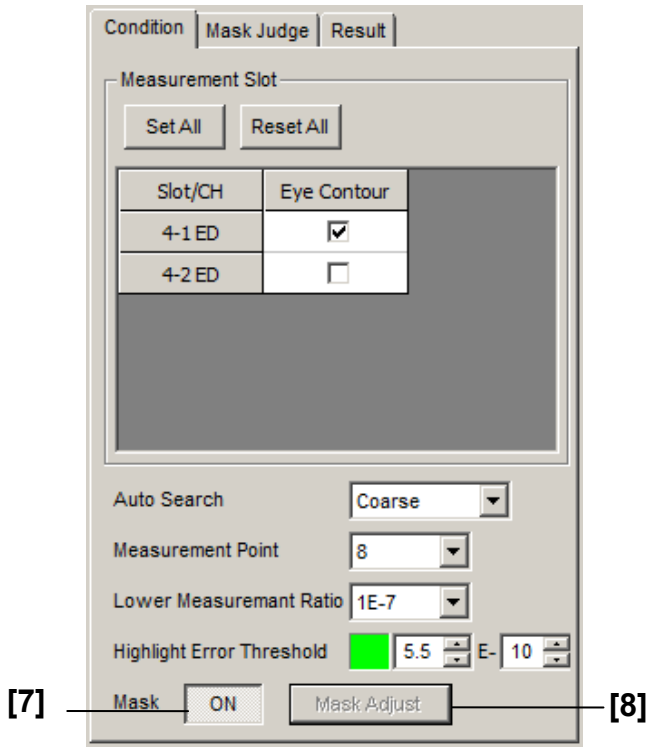


Figure 7.12.16.3.1-1 Condition Tab

Table 7.12.16.3.1-1 Condition Tab Setting Commands

No.	Setting Items	Commands
[7]	Mask	:DISPlay:RESult:EDIagram:DISPlay:MASK
		:DISPlay:RESult:EDIagram:DISPlay:MASK?
[8]	Mask Adjust	:DISPlay:RESult:EDIagram:ADJust

For commands description, refer to 7.12.10.2.2 “Condition setting commands”.

7.12.16.3.2 Eye Contour tab (Status) setting commands

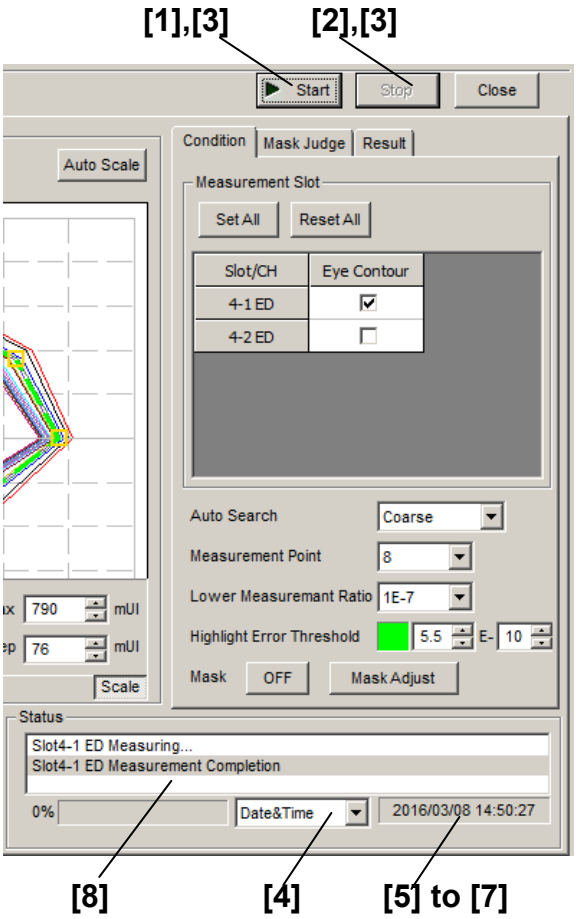


Figure 7.12.16.3.2-1 Diagram Tab, Start and Stop Buttons, and Status Field

Table 7.12.16.3.2-1 Diagram Tab (Status) Setting Commands

No.	Setting Items	Commands
[1]	Start	:SENSe:MEASure:EDIagram:DIAGram:STARt
[2]	Stop	:SENSe:MEASure:EDIagram:DIAGram:STOP
[3]	Query for measurement state	:SENSe:MEASure:EDIagram:STATe?
[4]	No label (Setting of time display)	:SENSe:MEASure:EDIagram:TIME
		:SENSe:MEASure:EDIagram:TIME?
[5]	Query for current time	:SENSe:MEASure:EDIagram:DTIMe?
[6]	Query for measurement start time	:SENSe:MEASure:EDIagram:STARt?
[7]	Query for measurement elapsed time	:SENSe:MEASure:EDIagram:ELAPsed?
[8]	Query for measurement status	:CALCulate:DATA:EDIagram:STATus?

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

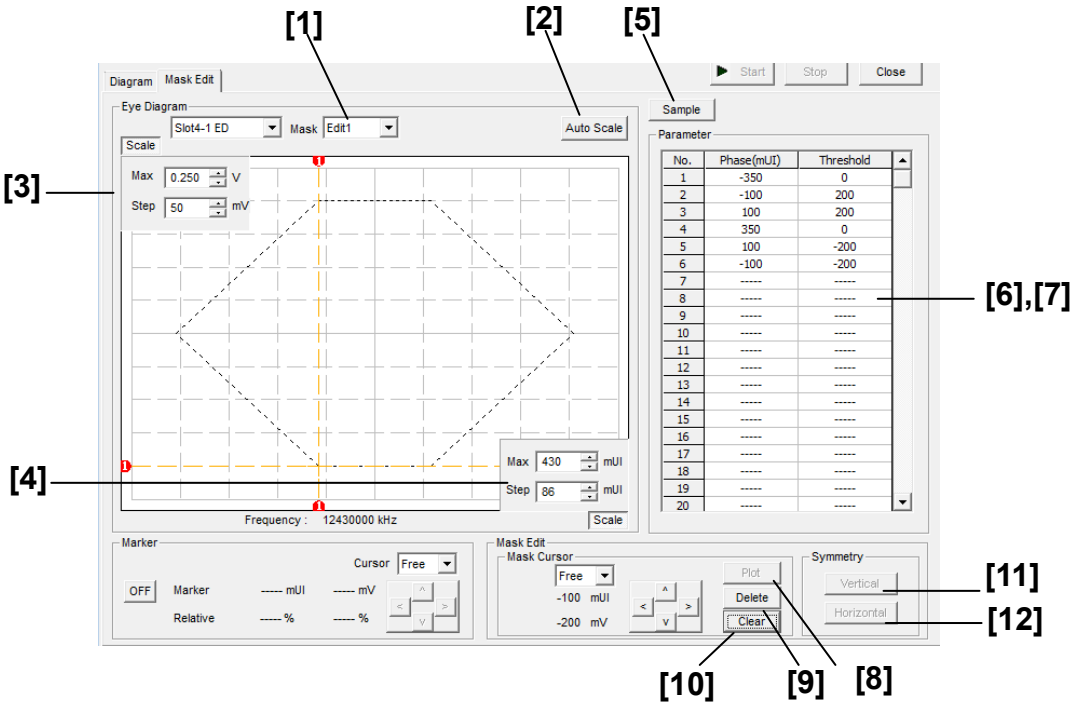


Figure 7.12.16.3.3-1 Mask Edit Tab

Table 7.12.16.3.3-1 Mask Edit Tab Setting Commands

No.	Setting Items	Commands
[1]	Mask	:DISPlay:RESult:EDIagram:TEMPlate:PATTern
		:DISPlay:RESult:EDIagram:TEMPlate:PATTern?
[2]	Auto Scale	:DISPlay:RESult:EDIagram:TEMPlate:ASCale
[3]	Max Step	:DISPlay:RESult:EDIagram:TEMPlate:SCALe
		:DISPlay:RESult:EDIagram:TEMPlate:SCALe?
[4]	Max Step	:DISPlay:RESult:EDIagram:TEMPlate:PSCale
		:DISPlay:RESult:EDIagram:TEMPlate:PSCale?
[5]	Sample	:DISPlay:RESult:EDIagram:TEMPlate:SAMPle
[6]	Query for mask point count	:DISPlay:RESult:EDIagram:TEMPlate:PNUMber?
[7]	Phase (mUI) Threshold	:DISPlay:RESult:EDIagram:TEMPlate:EDIT
		:DISPlay:RESult:EDIagram:TEMPlate:EDIT?
[8]	Plot	:DISPlay:RESult:EDIagram:TEMPlate:PLOT
[9]	Delete	:DISPlay:RESult:EDIagram:TEMPlate:DELete
[10]	Clear	:DISPlay:RESult:EDIagram:TEMPlate:CLEar
[11]	Vertical	:DISPlay:RESult:EDIagram:TEMPlate:VERTical
[12]	Horizontal	:DISPlay:RESult:EDIagram:TEMPlate:HORizontal

For commands description, refer to 7.12.10.2.5 “Mask Edit tab setting commands”.

7.12.16.3.4 File menu setting commands

Table 7.12.16.3.4-1 File Menu Setting Commands

No.	Setting Items	Commands
[1]	Open	:SYSTem:MMEMory:DIAGram:RECall
[2]	Save	:SYSTem:MMEMory:DIAGram:STORe

For commands description, refer to 7.12.10.2.6 “File menu setting commands”.

7.12.16.3.5 Result Query Command

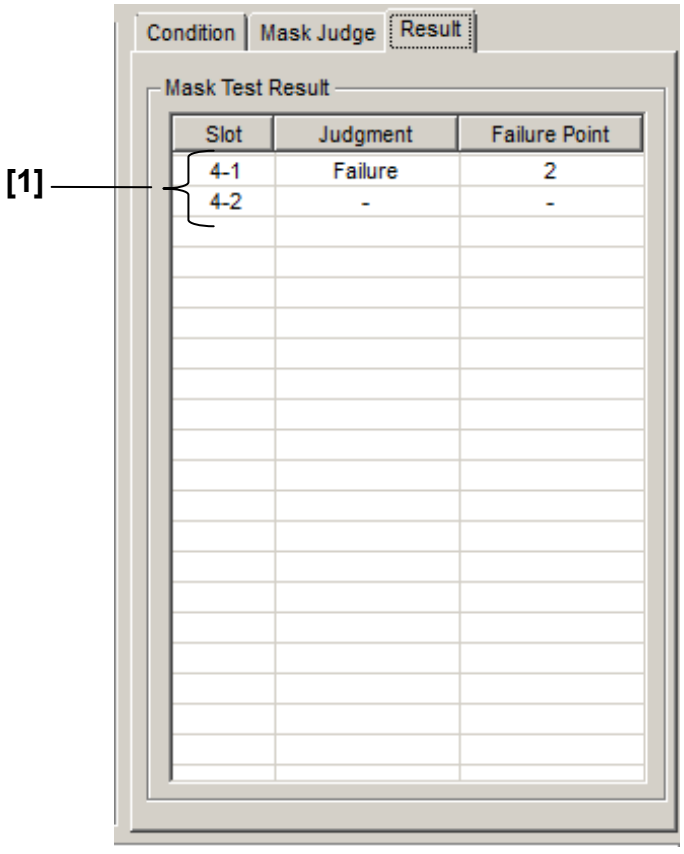


Figure 7.12.16.3.5-1 Result Tab

Table 7.12.16.3.5-1 Result Query Command

No.	Setting Item	Command
[1]	Query Mask Test result	:CALCulate:DATA:EDIagram:MASK?

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

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 8.1.1-1 Commands for common settings

Setting Items	Commands
Query for software status	CND?
Query for hardware system configuration	HDR?
Query for system error	INF?
Terminator type	TRM?
Query for model name of mainframe and module	CUN?
Query for mainframe information	UNT?
Query for module information	MDN?

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.

<module> = <STRING RESPONSE DATA>
XXXXXXXXX (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>
XXXXXXXXXX (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>
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 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.
Example	> CND? < CND 6201234567, 1.00.00, 1.00.20,OPT301,OPT302, 1.00.00, 1.00.00,OPT 12, 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	Incompatible with existing models.

Table 8.1.1-2 Option character correspondence table

Model/Name	Option Number	Option Name
MP1800A Signal Quality Analyzer	OPT014	2-Slot PPG/ED
	OPT015	4-Slot PPG/ED
	OPT016	6-Slot PPG/ED
MT1810A 4 Slot Chassis	OPT014	2-Slot PPG/ED
	OPT015	4-Slot PPG/ED
MX180000A Signal Quality Analyzer Control Software	–	–
MU181000A 12.5 GHz Synthesizer	OPTx01	Jitter Modulation
MU181000B 12.5 GHz 4port Synthesizer	OPTx01	Jitter Modulation
MU181020A 12.5 Gbit/s PPG	OPT001	9.8 to 12.5 Gbit/s
	OPT002	0.1 to 12.5 Gbit/s
	OPTx10	Variable Data Output (0.05 to 0.8 Vp-p)
	OPTx11	Variable Data Output (0.25 to 2.5 Vp-p)
	OPTx12	High Performance Data Output (0.05 to 2.0 Vp-p)
	OPTx13	Variable Data Output (0.5 to 3.5 Vp-p)
	OPTx21	Differential Clock Output (0.1 to 2.0 Vp-p)
	OPTx30	Variable Data Delay
MU181040A 12.5 Gbit/s ED	OPT001	9.8 to 12.5 Gbit/s
	OPT002	0.1 to 12.5 Gbit/s
	OPTx20	Clock Recovery
	OPTx30	Variable Clock Delay
MU181600A Optical Transceiver (XFP)	–	–
MU181601A Optical Transceiver (SFP)	–	–
MU181620A Stressed Eye Transmitter	OPTx01	1310nm Reference
	OPTx02	1550nm Reference
	OPTx03	1310nm/1550nm Reference
	OPTx11	1310nm Stressed Eye
	OPTx12	1550nm Stressed Eye
	OPTx13	1310nm/1550nm Stressed Eye
MU181640A Optical Receiver	OPTx04	Band Width 8.5GHz
MU181800A 12.5GHz Clock Distributor	–	–

Table 8.1.1-2 Option character correspondence table (Cont'd)

Model/Name	Option Number	Option Name
MU181020B 14 Gbit/s PPG	OPT002	0.1 to 14 Gbit/s
	OPTx11	Variable Data Output (0.25 to 2.5 Vp-p)
	OPTx12	High Performance Data Output (0.05 to 2.0 Vp-p)
	OPTx13	Variable Data Output (0.5 to 3.5 Vp-p)
	OPTx21	Differential Clock Output (0.1 to 2.0 Vp-p)
	OPTx30	Variable Data Delay
MU181040B 14 Gbit/s ED	OPT002	0.1 to 14 Gbit/s
	OPTx20	Clock Recovery
	OPTx30	Variable Clock Delay
MU182020A 25 Gbit/s 1ch MUX MU182021A 25 Gbit/s 2ch MUX	OPTx01	28 Gbit/s Extension
	OPTx02	Clock Input Band Switch
	OPTx10	Variable Data Output (0.25 to 1.75Vp-p)
	OPTx11	Variable Data Output (0.5 to 2.5Vp-p)
	OPTx12	Variable Data Output (0.5 to 2.0Vp-p)
	OPTx13	Variable Data Output (0.5 to 3.5Vp-p)
	OPTx21	Differential Clock Output (0.5 to 2.0Vp-p)
	OPTx30	25Gbit/s Variable Data Delay
	OPTx31	28Gbit/s Variable Data Delay
MU182040A 25 Gbit/s 1ch DEMUX MU182041A 25 Gbit/s 2ch DEMUX	OPTx01	28Gbit/s Extension
	OPTx02	Clock Input Band Switch
	OPTx30	25GHz Variable Clock Delay
	OPTx31	28GHz Variable Clock Delay

Note:

The option names of the MU181020B-003 and MU181040B-003 are not output.

HDR?

Response

<slot1>,...,<slot6>
<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>

XXXXXXXXX (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.

<serial> = <STRING RESPONSE DATA>

XXXXXXXXXX (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.

<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 lower 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 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.

Function
Queries the hardware system configuration of the MP1800A/MT1810A.

Example
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,OP
T001,
OPT220,
MU181040A,6201234571,0001.00.00,0001.00.00,0001.00.00,OP
T002,
OPT220"
```

Compatibility
Partially compatible with the MP1632C and MP1776A.

INF? <unit>

Parameter	<unit> = <DECIMAL NUMERIC PROGRAM DATA>	
	1 to 4	Mainframe number
Response	<numeric> = <DEFINITE LENGTH ARBITORARY BLOCK RESPONSE DATA>	
	#B0000	NONE
	#B100	PLL Unlock
	#B0100	Temperature
	#B0010	Fan
Function	Queries the content of the system error that has currently occurred.	
Example	To query the content of the system error that has occurred in Mainframe 1:	
	`> INF? 1`	
	`< INF #B1000`	
Compatibility	Incompatible with existing models.	

TRM <numeric>

Parameter	<numeric> = <DECIMAL NUMERIC PROGRAM DATA>	
	0	Switches the terminator type to LF+EOI.
	1	Switches the terminator type to CR+LF+EOI.
Function	Sets the terminator type of the response data.	
Example	To set the terminator type to LF+EOI:	
	`> TRM 0`	
Compatibility	Compatible with the MP1632C.	

TRM?

Response	<numeric> = <NR1 NUMERIC RESPONSE DATA> (FIX1)	
	0	Terminator type LF+EOI
	1	Terminator type CR+LF+EOI
Function	Queries the terminator type of the response data.	
Example	`> TRM?`	
	`< TRM 0`	
Compatibility	Compatible with the MP1632C.	

CUN?

Response

<unit1>,...,<unit4>,<slot1>,...,<slot6>
="<mainframe1>,...,<mainframe4>,<module1>,...,<module6>"
<mainframe1> to <mainframe4> = <STRING RESPONSE DATA>

XXXXXXXXX	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>

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.

Function

Queries the model name of the MP1800A/MT1810A and module.

Example

```
> CUN?
< NONE,MU181000A,NONE,MU181020A,
MU181040A,NONE,...,MU181600A,MU181601A,MU181620A,NONE,
MU181640A,NONE,...,NONE
```

Example

To query the model name of the MP1800A/MT1810A and module for the MP1800A (1-box type):

```
> CUN?
< MP1800A,NONE,NONE,NONE,MU181000A,NONE,MU181020A,
MU181040A,NONE,MU181600A,NONE,...,NONE
```

Compatibility

Incompatible with existing models.

UNT? <numeric>

Parameter

<numeric> = <NR1 NUMERIC PROGRAM DATA>

1 to 4 Mainframe number

For the MP1800A set to "1" and for the MT1810A set to 1 to 4.

Response

"<mainframe>",

<mainframe> = <module>,<serial>,<mver>,<hver>,<opt1>,<sbver>,<saver>,<opt2>

<module> = <STRING RESPONSE DATA>

XXXXXXXXX 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.

<serial> = <STRING RESPONSE DATA>

XXXXXXXXXX 0000000000 to 9999999999
Main frame 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 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 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)

Function

Queries the mainframe information including model and serial number.

Example

To query the Mainframe 1 information:

```
> UNT? 1
<
"MP1800A,6201234568,1.00.00,1.00.00,OPT01,1.00.00,1.00.0
0"
```

Compatibility

Incompatible with existing models.

MDN? <numeric>

Parameter

<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

Response

<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>

XXXXXXXXXX 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>

XXXXXX/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 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.

Function

Queries the module information on the specified slot.

Example

To query the module information on Slot 3:

```
> MDN? 3
< MDN
"MU181020A,6201234568,1.00.00,-----,1.00.00,1.00.00,OPT0
02,
 OPT211, OPT220, "
```

Compatibility

Incompatible with existing models.

8.1.2 Commands related to File menu

Table 8.1.2-1 File Menu Setting Command

Setting Items	Commands
Screen Copy	HCP
Combination Setting	CST?
	CMA?
	CST
	CHA?
	CHS
Initialize	INI

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:

```
> 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>]

Parameter	<configuration> = <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
	[<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>]

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 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:	

```
> 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

Figure 8.1.3-1 Common function button

Table 8.1.3-1 Common Function Button Setting Commands

Setting Items	Commands
Data/Clock Output ON/OFF	OON
	OON?
Error Addition ON/OFF	EAS
	EAS?
Single Error Addition	EAI
Meas. Start	SAT
Meas. Stop	SOT
(Query for measurement status)	STT?
Buzzer Setting Error ON/OFF	MON
	MON?
Buzzer Setting Alarm ON/OFF	ALM
	ALM?
Buzzer Setting System Error ON/OFF	SYS
	SYS?
Buzzer Setting System Error	SYT
	SYT?

OON <numeric>

Parameter	<numeric> = <DECIMAL NUMERIC PROGRAM DATA> 0 Data and Clock output OFF 1 Data and Clock output ON
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?

Response	<numeric> = <NR1 NUMERIC RESPONSE DATA> (FIX)
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>

Parameter	<numeric> = <DECIMAL NUMERIC PROGRAM DATA> 0 Error addition OFF 1 Error addition ON
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?

Response	<numeric> = <NR1 NUMERIC RESPONSE DATA> (FIX)
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.

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	<numeric> = <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	<numeric> = <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	<numeric> = <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>

Parameter	<numeric> = <DECIMAL NUMERIC PROGRAM DATA>
	0 Buzzer OFF
	1 Buzzer ON
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?

Response	<numeric> = <NR1 NUMERIC RESPONSE DATA> (FIX1)
Function	Queries the buzzer ON/OFF state at alarm occurrence.
Example	`> ALM?` `< ALM 0`
Compatibility	Compatible with the MP1632C and MP1762D.

SYS <numeric>

Parameter	<numeric> = <DECIMAL NUMERIC PROGRAM DATA>
	0 Buzzer OFF
	1 Buzzer ON
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?

Response	<numeric> = <NR1 NUMERIC RESPONSE DATA> (FIX1)
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

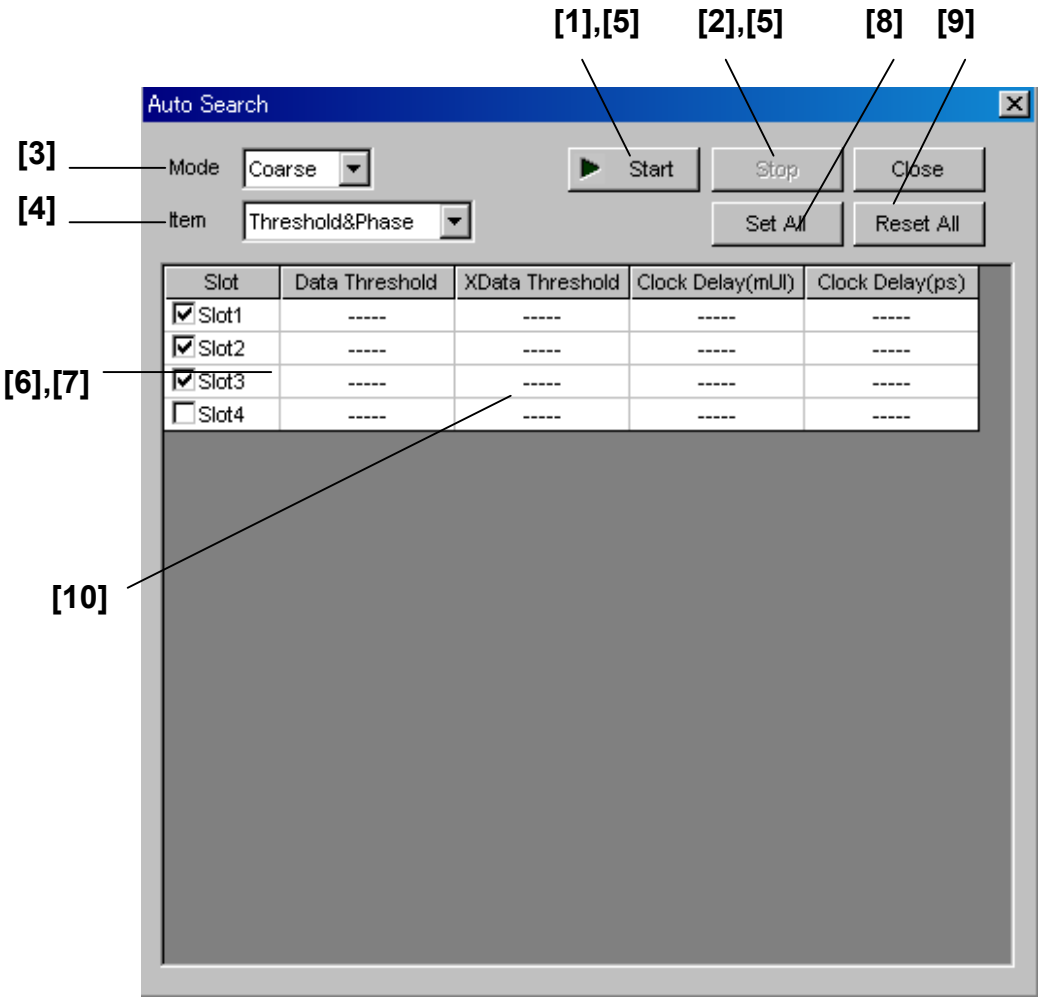


Figure 8.1.4-1 Auto Search dialog box

Table 8.1.4-1 Auto Search Setting Commands

No.	Setting Items	Commands
[1]	Start	AST
[2]	Stop	ASO
[3]	Mode	AMS
		AMS?
[4]	Item	ASM
		ASM?
[5]	Start/Stop	SRH
		SRH?
[6]	No label (Query for slot state)	ASL?
[7]	Slot Select	ASE
		ASE?
[8]	Set All	ASA
[9]	Reset All	ASR
[10]	No label (Query for results)	ARS?

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: <mode> = <DECIMAL NUMERIC PROGRAM DATA>

0	Fine mode
1	Coarse mode

Function: Sets the Auto Search mode.

Example: To set the Auto Search mode to Fine mode:
```
> AMS 0
```

Compatibility: Incompatible with existing models.

AMS?

Response	<mode> = <NR1 NUMERIC RESPONSE DATA> (FIX1)
Function	Queries the Auto Search mode.
Example	> AMS? < AMS 0
Compatibility	Incompatible with existing models.

ASM <mode>

Parameter	<mode> = <DECIMAL NUMERIC PROGRAM DATA> 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> = <NR1 NUMERIC RESPONSE DATA> (FIX1)
Function	Queries the execution item of Auto Search.
Example	> ASM? < ASM 0
Compatibility	Compatible with the MP1632C.

SRH <numeric>

Parameter	<numeric> = <NR1 NUMERIC PROGRAM DATA> 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.

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:

> 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.

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.

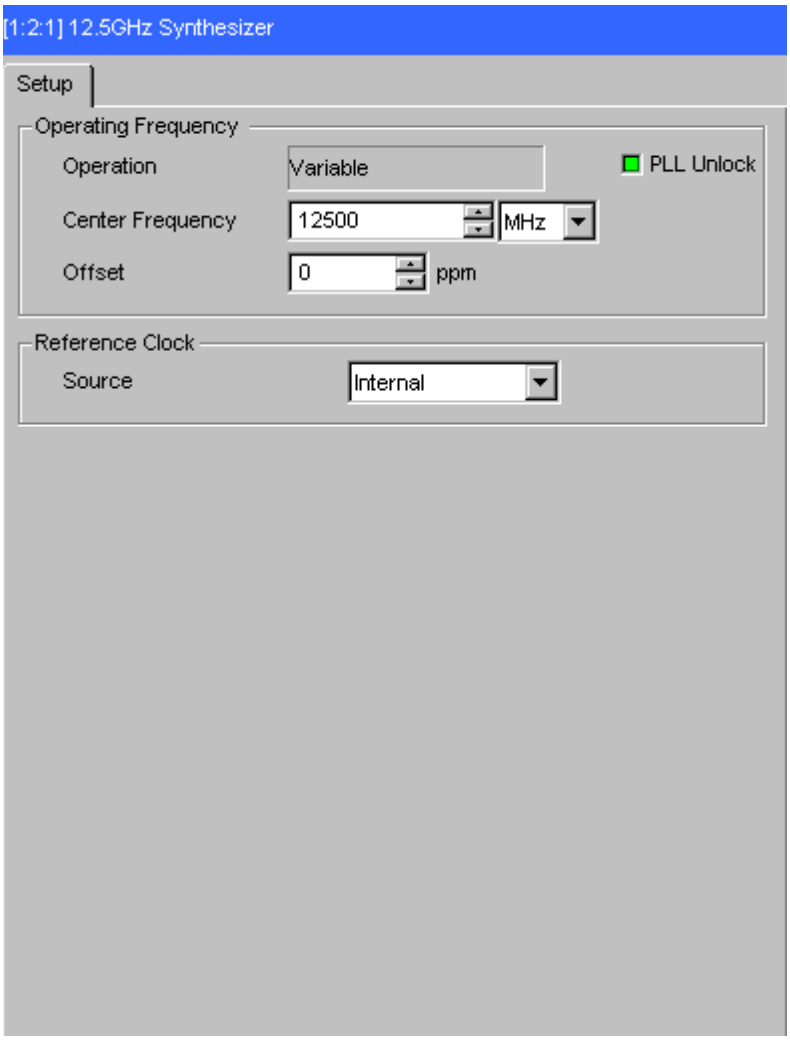


Figure 8.2-1 Synthesizer Window

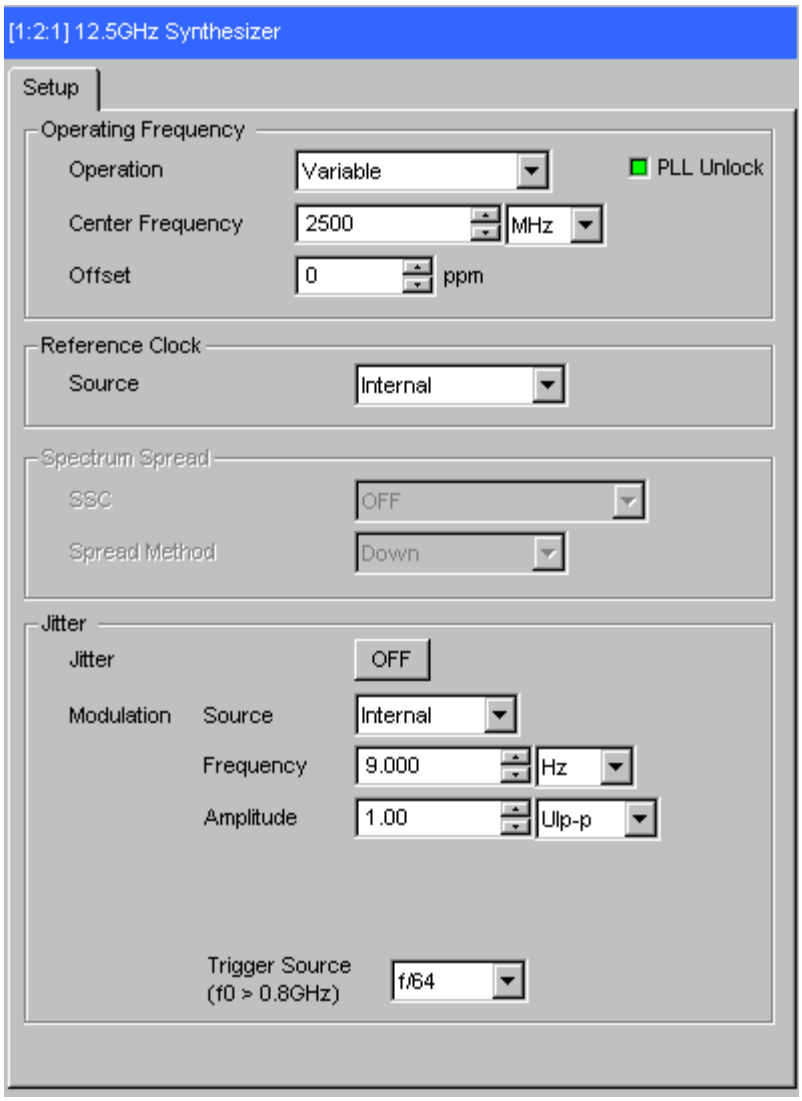


Figure 8.2-2 12.5 GHz Synthesizer Window (Option x01)

8.2.1 Operation Frequency setting commands

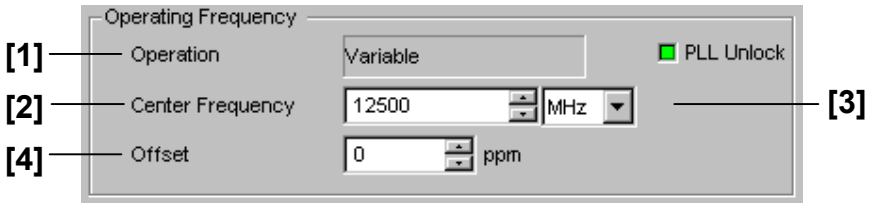


Figure 8.2.1-1 Operation Frequency setting field

Table 8.2.1-1 Operation Frequency setting commands

No.	Setting Items	Commands
[1]	Operation	OPE
		OPE?
[2]	Center Frequency	FRQ
		FRQ?
[3]	Center Frequency Unit	RES
		RES?
[4]	Offset	COP
		COP?

OPE <numeric>

Parameter	<numeric> = <DECIMAL PROGRAM DATA>
	0 Variable
	1 PCIe-Gen I (2.5 GHz)
	2 PCIe-Gen II (5 GHz)
	3 Serial-ATA (6 GHz)
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	<numeric> = <NR1 RESPONSE DATA> (FIX1)
	0 Variable
	1 PCIe-Gen I (2.5 GHz)
	2 PCIe-Gen II (5 GHz)
	3 Serial-ATA (6 GHz)
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.

FRQ <numeric>

Parameter
<numeric> = <DECIMAL NUMERIC PROGRAM DATA>
When the operation frequency setting resolution unit is kHz:
100000 to 12500000 100,000 to 12,500,000 kHz, in 1 kHz steps
When the operation frequency setting resolution unit is MHz:
100 to 12500 100 to 12,500 MHz, in 1 MHz steps

Function
Sets the operation frequency of the synthesizer.

Example
To set the operation frequency of the synthesizer to 12500000 kHz:

```
> FRQ 12500000
```

To set the operation frequency of the synthesizer to 100 MHz:

```
> FRQ   100
```

Compatibility
Partially compatible with the MP1632C and MP1761C.
Compatible with the MP1775A Pulse Pattern Generator (hereinafter, referred to as "MP1775A").

FRQ?

Response
<numeric> = <NR1 NUMERIC RESPONSE DATA>
When the operation frequency setting resolution unit is kHz (FIX8):
100000 to 12500000 100,000 to 12,500,000 kHz
When the operation frequency setting resolution unit is MHz (FIX5):
100 to 12500 100 to 12,500 MHz

Function
Queries the operating frequency of the synthesizer.

Example
To query the operating frequency of the synthesizer in kHz units:

```
> FRQ?
< FRQ 12500000
< FRQ   100000
```

To query the operating frequency of the synthesizer in MHz units:

```
> FRQ?
< FRQ   100
```

Compatibility
Partially compatible with the MP1632C and MP1761C.
Compatible with the MP1775A.

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.

8.2.2 Reference Clock setting commands

Figure 8.2.2-1 Reference Clock setting field

Table 8.2.2-1 Reference Clock setting commands

No.	Setting Item	Commands
[1]	Source	RFC
		RFC?

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

No.	Setting Items	Commands
[1]	SCC	RSU
		RSU?
[2]	Spread Method	RSM
		RSM?

RSU <numeric>

Parameter
<numeric> = <DECIMAL NUMERIC PROGRAM DATA>

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)

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:

```
> RSU 1
```

Compatibility
Incompatible with existing models.

RSU?

Response
<numeric> = <NR1 NUMERIC RESPONSE DATA (FIX1)

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)

Function
Queries setting of SSC (Spread Spectrum Clocking) function of the synthesizer.

Example
```
> RSU?
< RSU 1
```

Compatibility
Incompatible with existing models.

RSM <numeric>

Parameter
<numeric> = <DECIMAL NUMERIC PROGRAM DATA>

0	Changes frequency within the range from 0 to –0.5%
1	Changes frequency within the range of ±0.25%

Function
Sets spectrum spreading method for SSC (Spread Spectrum Clocking) using 100 MHz reference signal.

Example
To set spectrum spreading method for SSC using 100 MHz reference signal to DOWN:
```
> RSM 0
```

Compatibility
Incompatible with existing models.

RSM?

Response
<numeric> = <NR1 NUMERIC RESPONSE DATA> (FIX1)

0	Changes frequency within the range from 0 to –0.5%
1	Changes frequency within the range of ±0.25%

Function
Queries spectrum spreading method for SSC (Spread Spectrum Clocking) using 100 MHz reference signal.

Example
```
> RSM?
< RSM 0
```

Compatibility
Incompatible with existing models.

8.2.4 Jitter setting commands

The following commands are enabled only when the Option MU181000A-x01 or MU181000B-x01 is available.

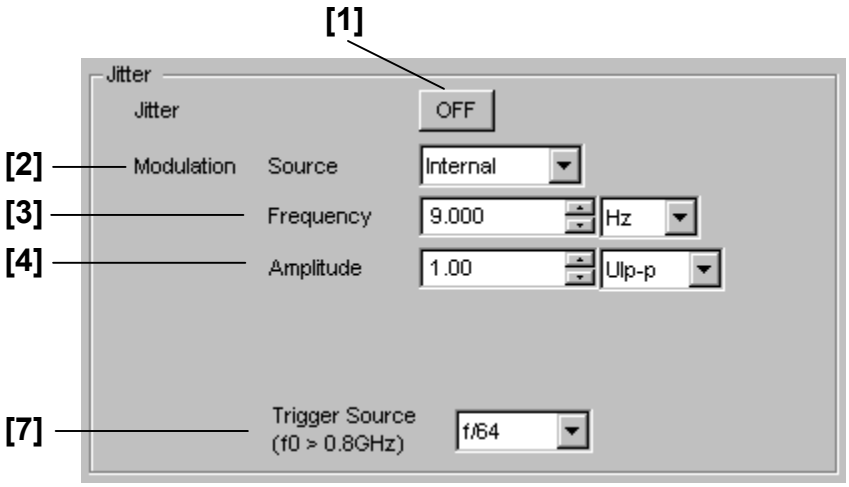


Figure 8.2.4-1 Jitter setting (Modulation Source: Internal)

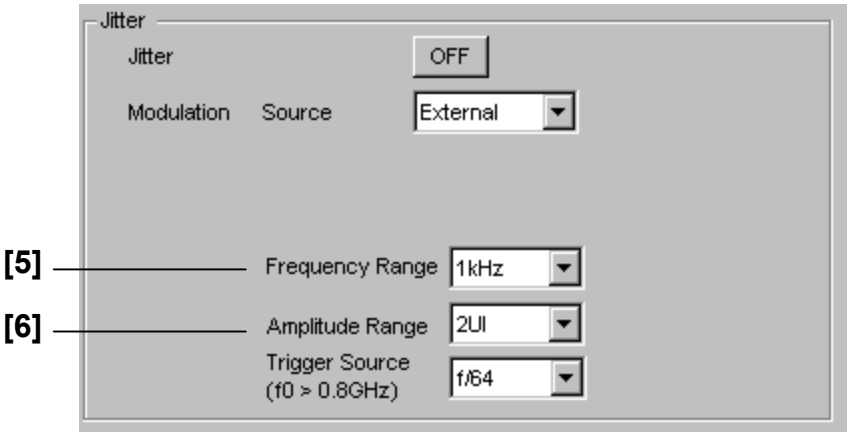


Figure 8.2.4-2 Jitter setting (Modulation Source: External)

Table 8.2.4-1 Jitter setting commands

No.	Setting Items	Commands
[1]	Jitter ON/OFF	JOU
		JOU?
[2]	Modulation Source	JMS
		JMS?
[3]	Frequency	JMF
		JMF?
[4]	Amplitude	JMP
		JMP
[5]	Frequency Range	JMG
		JMG?
[6]	Amplitude Range	JMA
		JMA?
[7]	Trigger Source	JTS
		JTS?

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>

Parameter	<numeric> = <DECIMAL NUMERIC PROGRAM DATA>
	0 Internal modulation source
	1 External modulation source
	2 External modulation source (IQ)
Function	Sets modulation signal source.
Example	To set modulation signal source to internal modulation source: `> JMS 0`
Compatibility	Incompatible with existing models.

JMS?

Response	<numeric> = <NR1 NUMERIC RESPONSE DATA> (FIX1)
	0 Internal modulation source
	1 External modulation source
	2 External modulation source (IQ)
Function	Queries modulation signal source.
Example	`> JMS?` `< JMS 0`
Compatibility	Incompatible with existing models.

JMF <numeric1>,<numeric2>

Parameter

<numeric> = <DECIMAL NUMERIC PROGRAM DATA>

When Hz is selected for <numeric2>

9.000 to 1000.0	9.000 to 1000.0 Hz Setting steps 9.000[Hz] ≤ fmod ≤10.000 [Hz]: In 0.001-Hz steps 10.01[Hz] ≤ fmod ≤100.00 [Hz]: In 0.01-Hz steps 100.1[Hz] ≤ fmod ≤1000.0 [Hz]: In 0.1-Hz steps

When kHz is selected for <numeric2>

1.001 to 1000.0	1.00 to 1000.0 kHz Setting steps 1.001[kHz] ≤ fmod ≤10.000 [kHz]: In 0.001-kHz steps 10.01[kHz] ≤ fmod ≤100.00 [kHz]: In 0.01-kHz steps 100.1[kHz] ≤ fmod ≤1000.0 [kHz]: In 0.1-kHz steps

When MHz is selected for <numeric2>

1.001 to 80.00	1.00 to 80 MHz Setting steps 1.001[MHz] ≤ fmod ≤10.000 [MHz]: In 0.001-MHz step 10.01[MHz] ≤ fmod ≤80.00 [MHz]: In 0.01-MHz steps

<numeric2> = <CHARACTER PROGRAM DATA>

0	Hz units
1	kHz units
2	MHz units

Function

Sets internal modulation frequency and units.

Example

To set internal modulation frequency to 10 kHz:

```
> JMF  10.00,1
```

Compatibility

Incompatible with existing models.

JMF?

Response	<numeric1> = <NR2 NUMERIC RESPONSE DATA> (FIX6)	
	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)
	<numeric2> = <NR1 NUMERIC RESPONSE DATA> (FIX1)	
	0	Hz units
	1	kHz units
	2	MHz units
Function	Queries internal modulation frequency and units.	
Example	`> JMF?` `< JMF 10.000,1`	
Compatibility	Incompatible with existing models.	

JMP <numeric1>,<numeric2>

Parameter	<numeric1> = <DECIMAL NUMERIC PROGRAM DATA>	
	0 to 999	0 to 999 mUIp-p, in 1-mUp-p steps
	1.00 to 32.00	1.00 to 32.00 UIp-p, in 0.01-UIp-p steps
	32.1 to 256.0	32.1 to 256.0 UIp-p, in 0.1-UIp-p steps
	257 to 2049	257 to 2049 UIp-p, in 1-UIp-p steps
	2050 to 4000	2050 to 4000 UIp-p, in 10-UIp-p steps
	<numeric2> = <DECIMAL NUMERIC PROGRAM DATA>	
	0	mUIp-p units
	1	UIp-p units
Function	Sets internal modulation amplitude and units.	
Example	To set internal modulation amplitude to 1.100 UIp-p: `> JMP 1.100,1`	
Compatibility	Incompatible with existing models.	

JMP?

Response	<numeric1> = <NR2 NUMERIC RESPONSE DATA>	
	0 to 999	0 to 999 mUIp-p
	1.00 to 32.00	1.00 to 32.00 UIp-p
	32.1 to 256.0	32.1 to 256.0 UIp-p
	257 to 2049	257 to 2049 UIp-p
	2050 to 4000	2050 to 4000 UIp-p
	<numeric2> = <DECIMAL NUMERIC PROGRAM DATA> (FIX1)	
	0	mUIp-p units
	1	UIp-p units
Function	Queries internal modulation amplitude and units.	
Example	`> JMP?`	
	`< JMP 1.100,1`	
Compatibility	Incompatible with existing models.	

JMG <numeric>

Parameter	<numeric> = <DECIMAL NUMERIC PROGRAM DATA>	
	0	275 Hz
	1	550 Hz
	2	1 kHz
	3	1.1 kHz
	4	2.2 kHz
	5	2.75 kHz
	6	4.4 kHz
	7	5.5 kHz
	8	11 kHz
	9	22 kHz
	10	27.5 kHz
	11	44 kHz
	12	55 kHz
	13	100 kHz
	14	110 kHz
	15	220 kHz
	16	250 kHz
	17	440 kHz
	18	500 kHz
	19	1 MHz
	20	2 MHz
	21	4 MHz
	22	80 MHz
	23	500 MHz
	24	1 GHz
	27	All range
Function	Sets external modulation frequency range.	
Example	To set external modulation frequency range to 275 kHz:	
	`> JMG 0`	
Compatibility	Incompatible with existing models.	

JMG?

Response	<numeric> = <NR1 NUMERIC RESPONSE DATA> (FIX2)
Function	Queries external modulation frequency range.
Example	`> JMG?`
	`< JMG 0`
Compatibility	Incompatible with existing models.

JMA<numeric>

Parameter	<numeric> = <DECIMAL NUMERIC PROGRAM DATA>	
	0	0.22 UI
	1	2.00 UI
	2	20 UI
	3	200 UI
	4	4,000 UI
Function	Sets external modulation amplitude.	
Example	To set external modulation amplitude to 0.22 UI:	
	`> JMA 0`	
Compatibility	Incompatible with existing models.	

JMA?

Response	<numeric> = <NR1 NUMERIC RESPONSE DATA> (FIX1)	
	0	0.22 UI
	1	2.00 UI
	2	20 UI
	3	200 UI
	4	4,000 UI
Function	Queries external modulation amplitude.	
Example	`> JMA?`	
	`< JMA 0`	
Compatibility	Incompatible with existing models.	

JTS<numeric>

Parameter	<numeric> = <DECIMAL NUMERIC PROGRAM DATA>	
	0	f/1
	1	f/64
Function	Selects trigger output division ratio.	
Example	To set trigger output division ratio to f/1:	
	`> JTR 0`	
Compatibility	Incompatible with existing models.	

JTS?

Response	<numeric> = <NR1 NUMERIC RESPONSE DATA> (FIX1)
	0 f/1
	1 f/64
Function	Queries trigger output division ratio.
Example	`> 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

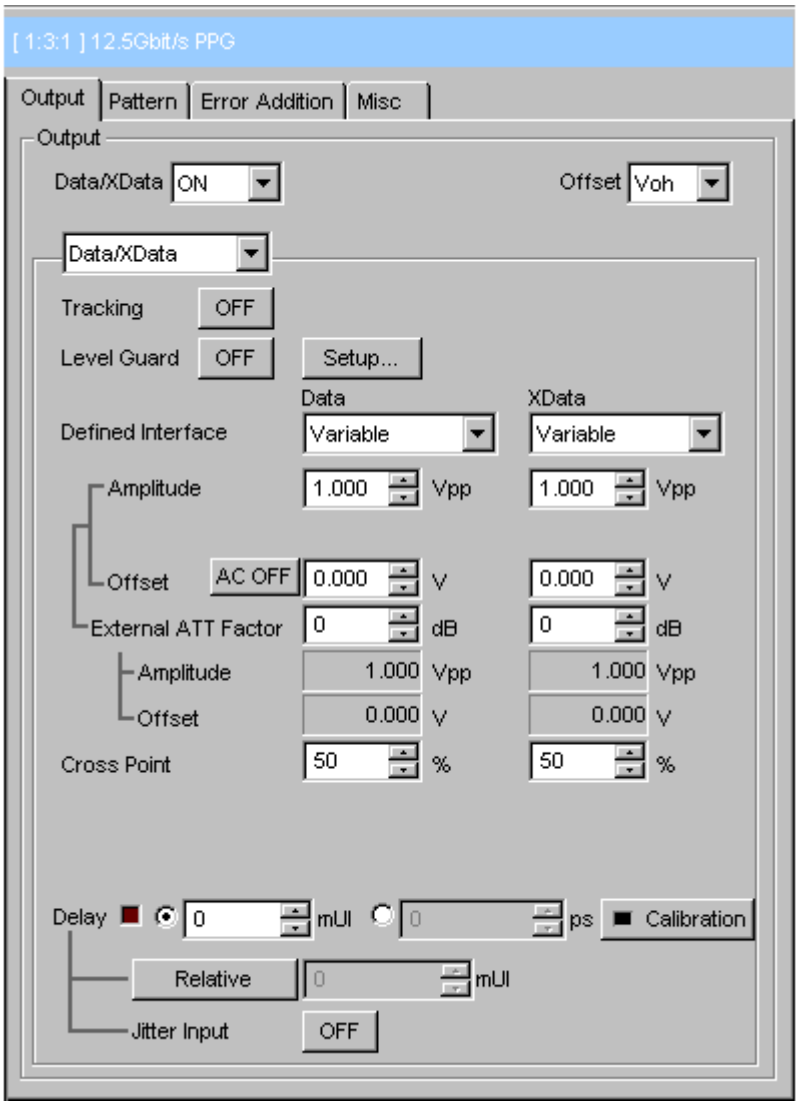


Figure 8.3.1-1 Output tab

8.3.1.1 Output setting commands

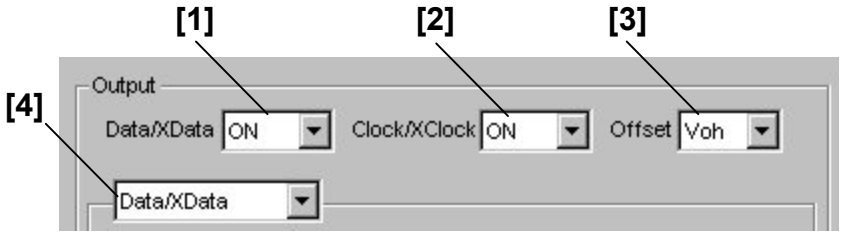


Figure 8.3.1.1-1 Output setting field

Table 8.3.1.1-1 Output setting commands

No.	Setting Items	Commands
[1]	Data/XData	DON
		DON?
[2]	Clock/XClock	CON
		CON?
[3]	Offset	OFS
		OFS?
[4]	No label (Setting item selection)	OOF
		OOF?

DON <numeric>

Parameter	<numeric> = <DECIMAL NUMERIC PROGRAM DATA>
	0 Output OFF
	1 Output ON
Function	Sets data output ON or OFF.
Example	To set data output ON: `> DON 1`
Compatibility	Compatible with the MP1632C.

DON?

Response	<numeric> = <NR1 NUMERIC RESPONSE DATA> (FIX1)
Function	Queries the data output ON or OFF state.
Example	`> DON?` `< DON 1`
Compatibility	Compatible with the MP1632C.

CON <numeric>

Parameter	<numeric> = <DECIMAL NUMERIC PROGRAM DATA> 0 Output OFF 1 Output ON
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	<offset> = <DECIMAL NUMERIC PROGRAM DATA> 0 VOH High level 1 VTH Center value between High and Low levels 2 VOL Low level
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.

OOF <numeric>

Parameter	<numeric> = <DECIMAL NUMERIC PROGRAM DATA>
	0 Data/XData setting screen
	1 Clock/XClock setting screen
	2 CMU Bitrate setting screen
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?

Response	<numeric> = <NR1 NUMERIC RESPONSE DATA> (FIX1)
Function	Queries the selected setting screen.
Example	`> OOF?` `< OOF 0`
Compatibility	Incompatible with existing models.

8.3.1.2 Data/XData setting commands

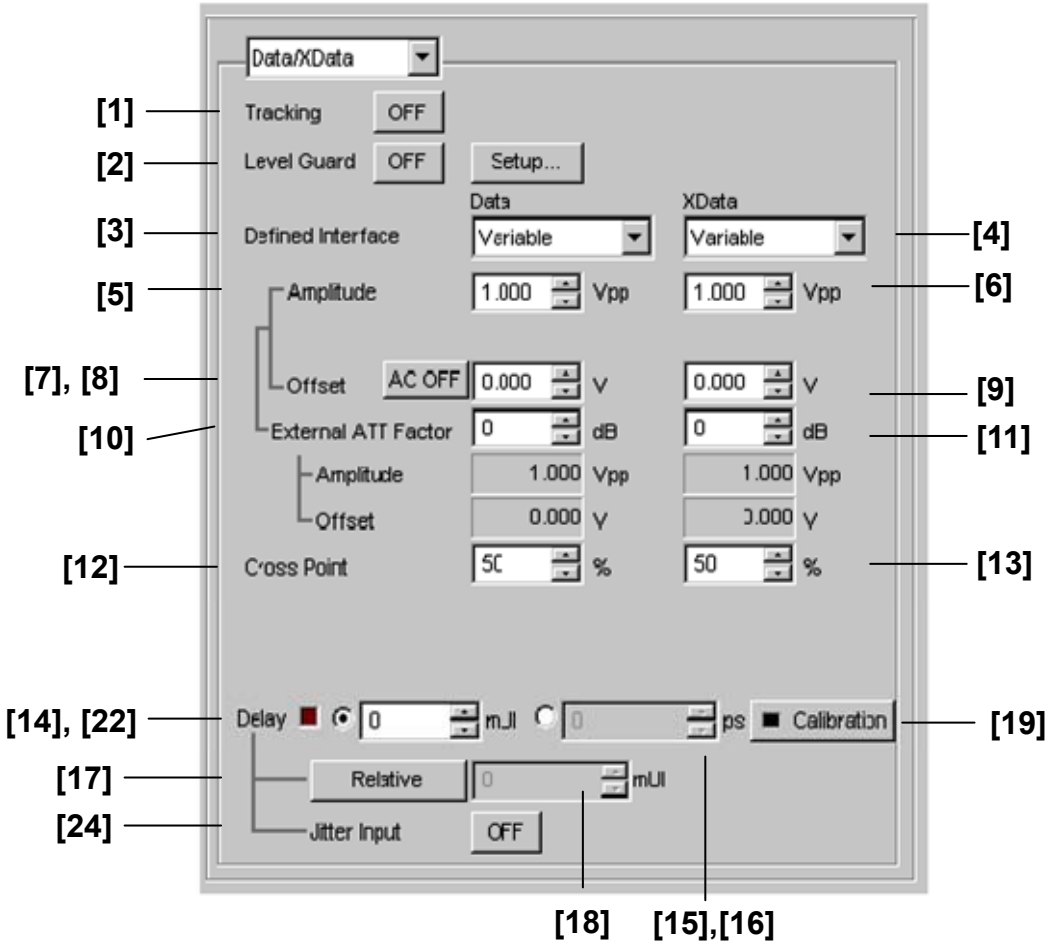


Figure 8.3.1.2-1 Data/XData setting field

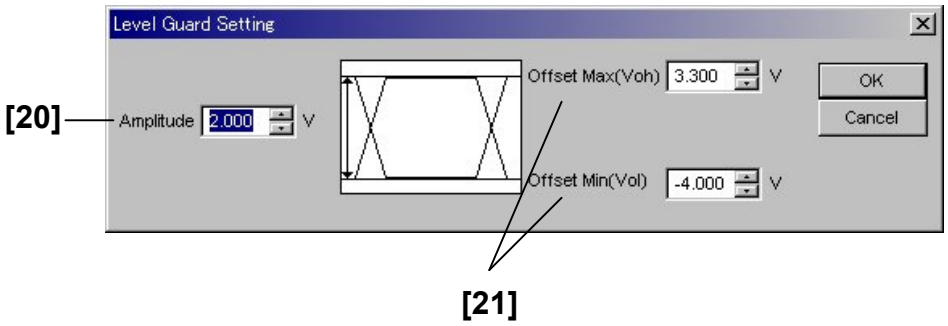


Figure 8.3.1.2-2 Level Guard Setting dialog box

Table 8.3.1.2-1 Data/XData setting commands

No.	Setting Items	Commands
[1]	Tracking	TRK
		TRK?
[2]	Level Guard	DGD
		DGD?
[3]	Defined Interface (Data)	DAL
		DAL?
[4]	Defined Interface (XData)	XDL
		XDL?
[5]	Amplitude (Data)	DAP
		DAP?
[6]	Amplitude (XData)	NAP
		NAP?
[7]	AC ON/OFF	DAD
		DAD?
[8]	Offset (Data)	DOS
		DOS?
[9]	Offset (XData)	NOS
		NOS?
[10]	External ATT Factor (Data)	DAT
		DAT?
[11]	External ATT Factor (XData)	XAT
		XAT?
[12]	Cross Point (Data)	DCR
		DCR?
[13]	Cross Point (XData)	XDC
		XDC?
[14]	Phase Adjust	PAU
		PAU?
[15]	Phase Adjust	PAD
		PAD?
[16]	Phase Adjust	CDL
		CDL?
[17]	Relative 0	PRO
		PRO?
[18]	No label (Relative value setting)	PRE
		PRE?
[19]	Calibration	PCA

Table 8.3.1.2-1 Data/XData setting commands (Cont'd)

No.	Setting Items	Commands
[20]	Amplitude max	DLM
		DLM?
[21]	Offset max/min	DOF
		DOF?
[22]	No label (Delay status query)	DLY?
[23]	No label (Query for data/clock output setting completion status)	OCS?
[24]	Jitter Input	PJO
		PJO?

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>

Parameter	<numeric> = <DECIMAL NUMERIC PROGRAM DATA>
	0 OFF
	1 ON
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?

Response	<numeric> = <NR1 NUMERIC RESPONSE DATA> (FIX1)
Function	Queries the data output range limitation ON or OFF state.
Example	`> DGD?` `< DGD 1`
Compatibility	Incompatible with existing models.

DAL(XDL) <level>

Parameter	<level> = <DECIMAL NUMERIC PROGRAM DATA>
	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 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)?

Response	<level> = <NR1 NUMERIC RESPONSE DATA> (FIX1)
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>

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 amplitude of the Data (XData) output.	
Example	To set the amplitude of the Data output to 1.000 V:	
	`> DAP 1.000`	
Compatibility	Compatible with the MP1632C, MP1761C and MP1775A (0.50 to 2.00 V).	

DAP(NAP)?

Response	<numeric> = <NR2 NUMERIC RESPONSE DATA> (FIX5)
Function	Queries the amplitude of the Data (XData) output.
Example	To query the amplitude of the Data output:
	`> DAP?`
	`< DAP 1.000`
Compatibility	Compatible with the MP1632C, MP1761C and MP1775A.

DAD <numeric>

Parameter	<numeric> = <DECIMAL NUMERIC PROGRAM DATA>	
	0	Offset OFF (DC output)
	1	Offset ON (AC output)
Function	Sets the data output offset ON or OFF	
Example	To set the data output offset ON:	
	`> DAD 1`	
Compatibility	Incompatible with existing models.	

DAD?

Response	<numeric> = <NR1 NUMERIC RESPONSE DATA> (FIX1)
Function	Queries the data output offset ON or OFF state.
Example	`> DAD?`
	`< DAD 1`
Compatibility	Incompatible with existing models.

DOS(NOS) <numeric>

Parameter	<numeric> = <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, x14)
	–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 (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)?

Response	<numeric> = <NR2 NUMERIC RESPONSE DATA> (FIX6)
Function	Queries the Data (XData) output offset.

Example	To query the XData output offset:
	`> NOS?`
	`< NOS  1.000`
	`< NOS -1.000`

Compatibility	Compatible with the MP1632C, MP1761C and MP1775A.

DAT(XAT) <numeric>

Parameter	<numeric> = <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: `> DAT 20`
Compatibility	Incompatible with existing models.

DAT(XAT)?

Response	<numeric> = <NR1 NUMERIC RESPONSE DATA> (FIX2)
Function	Queries the External ATT Factor of the Data (XData) output.

Example	To query the External ATT Factor of Data output:
	`> DAT?`
	`< DAT 20`
	`< DAT 0`

Compatibility	Incompatible with existing models.

DCR(XDC) <numeric>

Parameter	<numeric> = <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%: `> XDC 60`
Compatibility	Partially compatible with the MP1632C parameters (25 to 75%, in 1% steps).

DCR(XDC)?

Response	<numeric> = <NR1 NUMERIC RESPONSE DATA> (FIX2) <numeric> = <NR2 NUMERIC RESPONSE DATA> (FIX4)
Function	Queries the Data (XData) output crosspoint.

Example	To query the crosspoint of the XData output:
	`> XDC?`
	`< XDC 60`

Compatibility	Partially compatible with the MP1632C parameters (25 to 75%, in 1% steps).

PAU <numeric>

Parameter	<numeric> = <DECIMAL NUMERIC PROGRAM DATA>
	–1000 to 1000 –1,000 to 1,000 mUI, in 1-mUI steps
	In the case of Combination
	–64000 to 64000 –64,000 to 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 to 640,000 ps, in 10-ps steps (at 0.1 GHz)
	–64000.0 to 64000.0 –64,000 to 64,000 ps, in 1-ps steps (at 1 GHz)
	–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:
	`> PAD 100`
Compatibility	Incompatible with existing models.

PAD?

Response	<numeric> = <NR1 NUMERIC RESPONSE DATA> <NR2 NUMERIC RESPONSE DATA> (FIX8)
Function	Queries the data output delay in ps units.
Example	`> PAD?` `< PAD  -10000` (at 0.1 GHz) `< PAD -64000.0` (at 1 GHz)
Compatibility	Incompatible with existing models.

CDL <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 to 640,000 ps, in 10-ps steps (at 0.1 GHz)
	–64000.0 to 64000.0 –64,000 to 64,000 ps, in 1-ps steps (at 1 GHz)
	–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: `> CDL 100`
Compatibility	Compatible with the MP1632C, MP1761C and MP1775A.

CDL?

Response	<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	`> CDL?` `< CDL  -10000` (at 0.1 GHz) `< CDL -5120.00` (at 12.5 GHz)
Compatibility	Compatible with the MP1632C, MP1761C and MP1775A.

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>]

Parameter

<numeric> = <DECIMAL NUMERIC PROGRAM DATA>

(mUI units) –2000 to 2000 –2,000 to 2,000 mUI, in 1-mUI steps

(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)

In the case of Combination

(mUI units)–64000 to 64000 –64,000 to 64,000 mUI, in 1-mUI steps

(ps units) Converted from the calculated frequency value, based on the setting resolution of mUI.

–640000 to 640000	–640,000 to 640,000 ps, in 10-ps steps (at 0.1 GHz)
–64000.0 to 64000.0	–64,000 to 64,000 ps, in 1-ps steps (at 1 GHz)
–5120.00 to 5120.00	–5120 to 5120 ps, in 0.08-ps steps (at 12.5 GHz)

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

0	mUI units
1	ps units (The mUI 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. 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.

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.

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	<numeric> = <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	`> 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: `> 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	`> PJO?`
	`< PJO 1`
Compatibility	Incompatible with existing models.

8.3.1.3 Clock/XClock setting commands

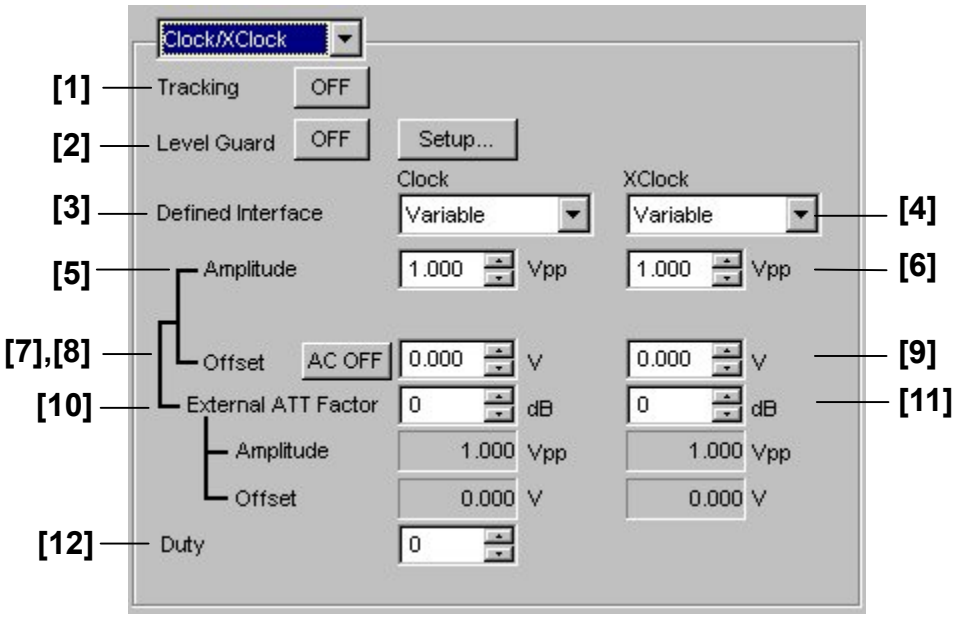


Figure 8.3.1.3-1 Clock/XClock setting field

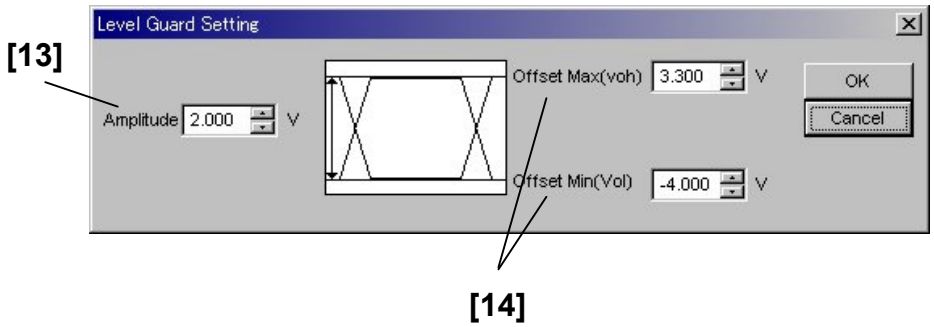


Figure 8.3.1.3-2 Level Guard Setting dialog box

Table 8.3.1.3-1 Clock/XClock setting commands

No.	Setting Items	Commands
[1]	Tracking	CGR
		CGR?
[2]	Level Guard	CLG
		CLG?
[3]	Defined Interface (Clock)	CLV
		CLV?
[4]	Defined Interface (XClock)	XLV
		XLV?
[5]	Amplitude (Clock)	CAP
		CAP?
[6]	Amplitude (XClock)	XCA
		XCA?
[7]	AC ON/OFF	CLA
		CLA?
[8]	Offset (Clock)	COS
		COS?
[9]	Offset (XClock)	XCF
		XCF?
[10]	External ATT Factor (Clock)	CAF
		CAF?
[11]	External ATT Factor (XClock)	XAF
		XAF?
[12]	Duty (Clock)	CDT
		CDT?
[13]	Amplitude	CLM
		CLM?
[14]	Offset Max/Min	CLO
		CLO?
[15]	No label (Query for data/clock output setting completion status)	OCS? Refer to 8.3.1.2 "Data/XData setting commands" for details on this command.

CGR <numeric>

Parameter	<numeric> = <DECIMAL NUMERIC PROGRAM DATA> 0 OFF 1 ON
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>

Parameter	<numeric> = <DECIMAL NUMERIC PROGRAM DATA> 0 OFF 1 ON
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.

CLV(XLV)<level>

Parameter	<level> = <DECIMAL NUMERIC PROGRAM DATA>
	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.

CLA <numeric>

Parameter	<numeric> = <DECIMAL NUMERIC PROGRAM DATA>
	0 Offset OFF (DC output)
	1 Offset ON (AC output)
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?

Response	<numeric> = <NR1 NUMERIC RESPONSE DATA> (FIX1)
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>
	–2.000 to 3.300 –2.000 to 3.300 Voh, in 0.001-V steps
	–3.000 to 3.250 –3.000 to 3.250 Vth, in 0.001-V steps
	–4.000 to 3.200 –4.000 to 3.200 Vol, in 0.001-V steps
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)?

Response	<numeric> = <NR2 NUMERIC RESPONSE DATA> (FIX6)
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.

CAF(XAF)<numeric>

Parameter	<numeric> = <DECIMAL NUMERIC PROGRAM DATA>
	0 to 40 0 to 40 dB, in 1-dB steps
Function	Sets the External ATT Factor of the Clock (XClock) output.
Example	To set the External ATT Factor of the Clock output to 20 dB:
	`> CAF 20`
Compatibility	Incompatible with existing models.

CAF(XAF)?

Response	<numeric> = <NR1 NUMERIC RESPONSE DATA> (FIX2)
Function	Queries the External ATT Factor of the Clock (XClock) output.
Example	To query the External ATT Factor of the Clock output:
	`> CAF?`
	`< CAF 20`
	`< CAF 0`
Compatibility	Incompatible with existing models.

CDT <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:
	`> CDT -25`
Compatibility	Compatible with the MP1632C.

CDT?

Response	<numeric> = <NR1 NUMERIC RESPONSE DATA> (FIX3)
Function	Queries the Duty of the Clock output.
Example	`> CDT?`
	`< CDT -25`
	`< CDT -5`
Compatibility	Compatible with the MP1632C.

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`
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

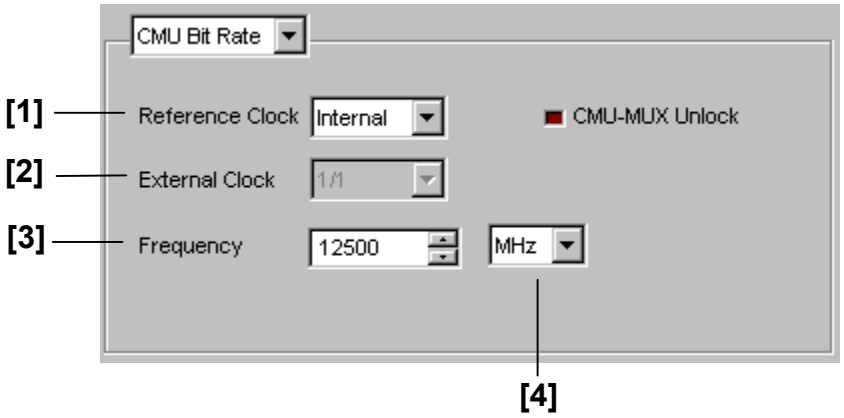


Figure 8.3.1.4-1 CMU Bit Rate setting field (for Internal)

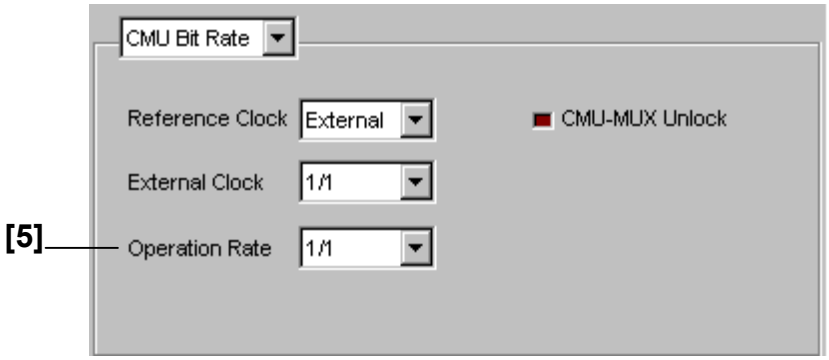


Figure 8.3.1.4-2 CMU Bit Rate setting field (for External)

Table 8.3.1.4-1 CMU Bit Rate setting commands

No.	Setting Items	Commands
[1]	Reference Clock	CRE
		CRE?
[2]	External Clock	CEC
		CEC?
[3]	Frequency	CRF
		CRF?
[4]	No label (Frequency unit setting)	CRS
		CRS?
[5]	Operation Rate	COR
		COR?

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 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 1562500 1225000 to 1562500 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
	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.

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.

8.3.2 Commands related to Pattern tab

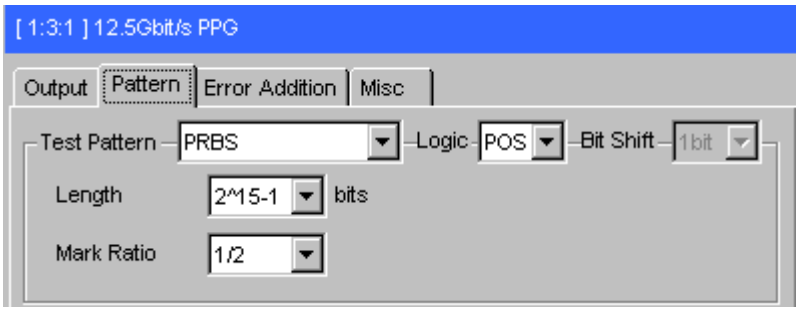


Figure 8.3.2-1 Pattern tab

8.3.2.1 Test Pattern setting commands

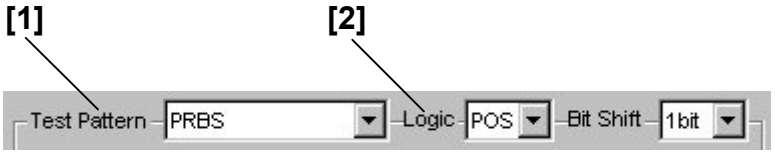


Figure 8.3.2.1-1 Test Pattern setting field

Table 8.3.2.1-1 Test Pattern setting commands

No.	Setting Items	Commands
[1]	Test Pattern	PTS
		PTS?
[2]	Logic	LGC
		LGC?

PTS <type>

Parameter	<type> = <DECIMAL NUMERIC PROGRAM DATA>	
	0	Alternate pattern
	1	Data pattern
	2	Zero-Substitution pattern
	3	PRBS pattern
	4	Mixed Data pattern
	5	Mixed Alternate pattern
	6	Sequence pattern
Function	Sets the type of the test pattern.	
Example	To set the test pattern type to the Mixed Data pattern:	
	`> PTS 4`	
Compatibility	Compatible with the MP1761C.	

PTS?

Response	<type> = <NR1 NUMERIC RESPONSE DATA> (FIX1)
Function	Queries the type of the test pattern.
Example	`> PTS?`
	`< PTS 4`
Compatibility	Compatible with the MP1761C.

LGC <logic>

Parameter	<logic> = <DECIMAL NUMERIC PROGRAM DATA>	
	0	Positive (positive logic)
	1	Negative (negative logic)
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):	
	`> LGC 1`	
Compatibility	Compatible with the MP1632C, MP1761C and MP1775A.	

LGC?

Response	<logic> = <NR1 NUMERIC RESPONSE DATA> (FIX1)
Function	Queries the logic at test pattern generation.
Example	`> LGC?`
	`< LGC 1`
Compatibility	Compatible with the MP1632C, MP1761C and MP1775A.

8.3.2.2 PRBS pattern setting commands

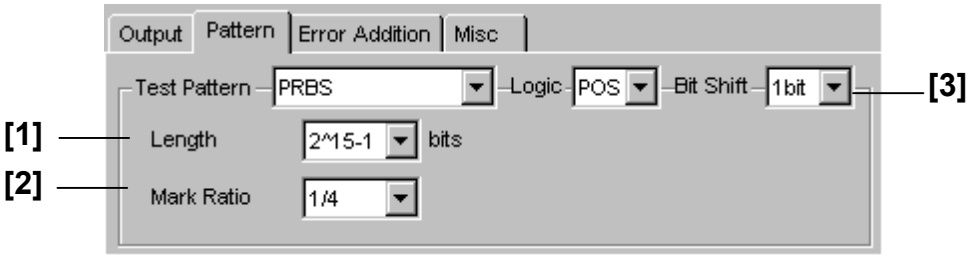


Figure 8.3.2.2-1 PRBS pattern setting field

Table 8.3.2.2-1 PRBS pattern setting commands

No.	Setting Items	Commands
[1]	Length	PTN
		PTN?
[2]	Mark Ratio	MRK
		MRK?
[3]	Bit Shift	SFT
		SFT?

PTN <numeric>

Parameter

<numeric> = <DECIMAL NUMERIC PROGRAM DATA>

	PRBS		Zero-Substitution
2	2^7-1	2	2^7
3	2^9-1	3	2^9
4	$2^{10}-1$	4	2^{10}
5	$2^{11}-1$	5	2^{11}
6	$2^{15}-1$	6	2^{15}
7	$2^{20}-1$	7	2^{20}
8	$2^{23}-1$	8	2^{23}
9	$2^{31}-1$	–	
		12	2^7-1
		13	2^9-1
		14	$2^{10}-1$
		15	$2^{11}-1$
		16	$2^{15}-1$
		17	$2^{20}-1$
		18	$2^{23}-1$

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?

Response

<numeric> = <NR1 NUMERIC RESPONSE DATA> (FIX2)

Function

Queries the number of stages during PRBS pattern or Zero-Substitution pattern generation.

Example

```
> PTN?
< PTN  6
< PTN 18
```

Compatibility

Compatible with the MP1632C, MP1761C and MP1775A.

MRK <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 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>	
	0	1 bit
	1	3 bits
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

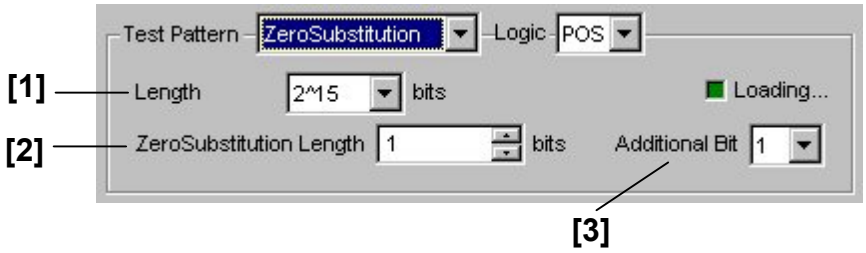


Figure 8.3.2.3-1 Zero-Substitution pattern setting field

Table 8.3.2.3-1 Zero-Substitution pattern setting commands

No.	Setting Items	Commands
[1]	Length	PTN Refer to 8.3.2.2 "PRBS pattern setting commands" for detail on this command.
		PTN? Refer to 8.3.2.2 "PRBS pattern setting commands" for detail on this command.
[2]	Zero-Substitution Length	ZLN
		ZLN?
[3]	Additional Bit	ZAD
		ZAD?

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.

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> 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.
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

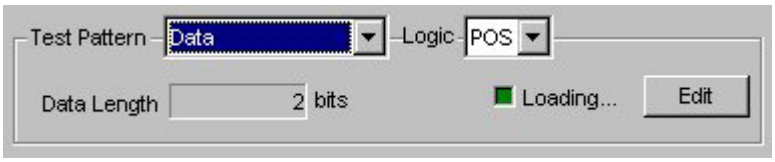


Figure 8.3.2.4-1 Data pattern setting field

[1] to [6]

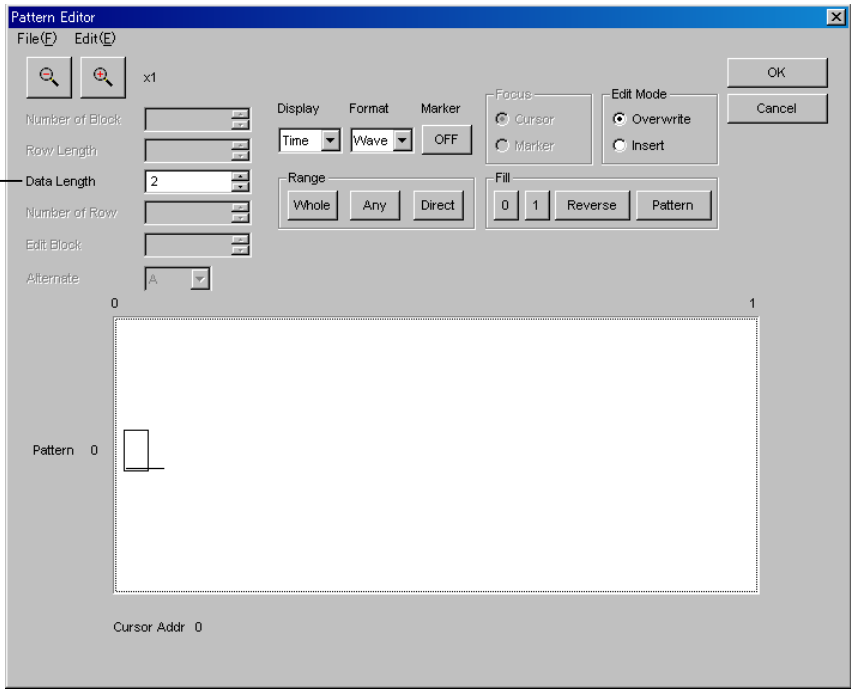


Figure 8.3.2.4-2 Data pattern setting (Pattern Editor screen)

Table 8.3.2.4-1 Data pattern setting commands

No.	Setting Items	Commands
[1]	Data Length	DLN
		DLN?
[2]	No label (Pattern setting)	PDT
		PDT?
[3]	No label (Pattern setting)	BDT
		BDT?
[4]	No label (Data inversion)	DRA
[5]	No label (Data inversion)	DRD
[6]	No label (All 0 or All 1)	ALL
[7]	No label (Pattern setting)	WRT
[8]	No label (Pattern reading)	RED?

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.

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: ABCABCAB
- <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

- <start> = #H0, <end> = #H3, <data> = #B01100110
  Setting data: 0110

PDT? <start>

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.

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 Data pattern generation.

Example

To query the pattern data from address 0:

```
> PDT? #H0
< PDT #H0123456789ABCDEF… (1 to 400 characters in hexadecimal)
< PDT #H0
```

Compatibility

Compatible with the MP1632C.

BDT <start>,<end>,<bdata>

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.

<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 value is the pattern length.

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.

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

the inversion range by the <start> and <end> addresses.

Example
To invert addresses 0 to 4F of the pattern data:

```
> DRA #H0,#H4F
```

Compatibility
Compatible with the MP1632C.

DRD <start>,<delta>

Parameter
<start> = <NON-DECIMAL NUMERIC PROGRAM DATA>

#H0 to #H7FFFFFFF	0 to 7FFFFFFF bits (Specify in hexadecimal)

<delta> = <DECIMAL NUMERIC PROGRAM DATA>

1 to 134217728	1 to 134,217,728 bit

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>

Parameter
<data> = <DECIMAL NUMERIC PROGRAM DATA>

0	Fills all bits with 0s.
1	Fills all bits with 1s.

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.

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 s pattern input tart address for binary transfer during Data pattern generation.

Example

To transfer 16-byte data from address 0:

```
> WRT 16,0
< ABCDEFGHIJKLMNOP (Binary data of 16 bytes)
```

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

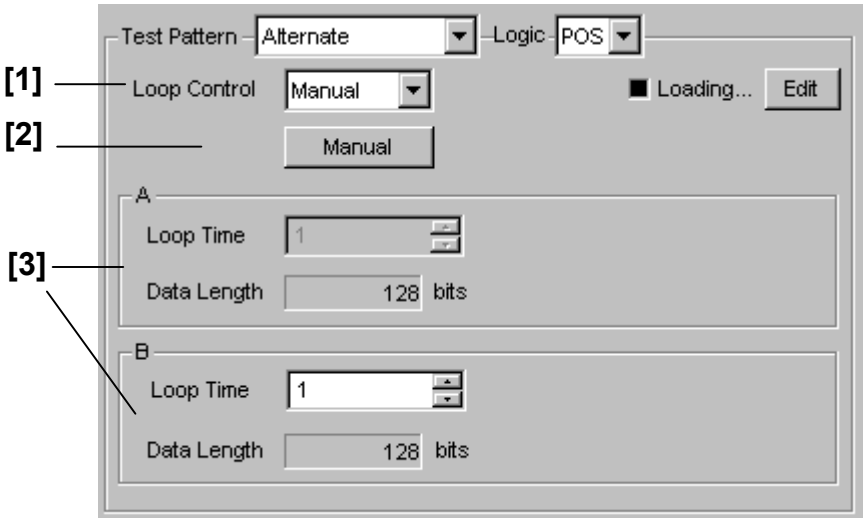


Figure 8.3.2.5-1 Alternate pattern setting field

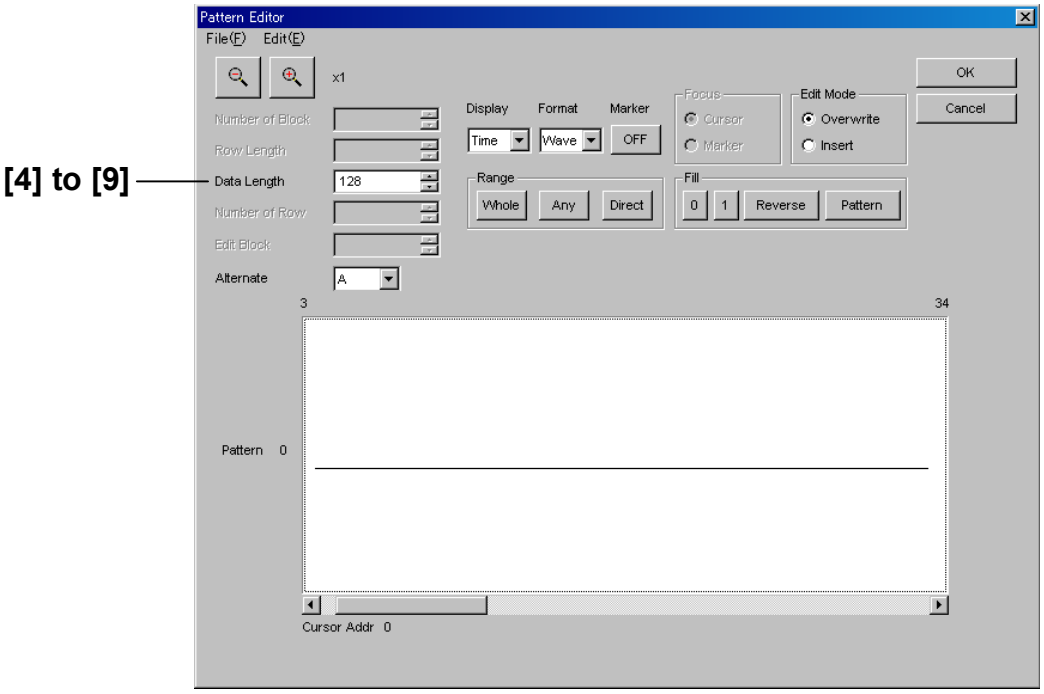


Figure 8.3.2.5-2 Alternate pattern setting (Pattern Editor screen)

Table 8.3.2.5-1 Alternate pattern setting commands

No.	Setting Items	Commands
[1]	Loop Control	APS
		APS?
[2]	Manual	AMN
[3]	A/B LoopTime	LPT
		LPT?
[4]	Data Length/Alternate	ALE
		ALE?
[5]	No label (Pattern setting)	ADW
		ADW?
[6]	No label (Pattern setting)	ABW
		ABW?
[7]	No label (Data inversion)	ARA
[8]	No label (Data inversion)	ARD
[9]	No label (All 0 or All 1)	ADF

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:

```
> 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

```
> APS?
< APS 1
```

Compatibility Compatible with the MP1761C.

AMN

Function Switches the Alternate pattern A/B manually.

Example

```
> 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.

ALE? <content>

Parameter
<content> = <DECIMAL NUMERIC PROGRAM DATA>
0 Pattern A
1 Pattern B

Response
<numeric> = <NR1 NUMERIC RESPONSE DATA> (FIX8)

Function
Queries the Alternate pattern length.

Example
To query the pattern length of the Alternate pattern B:

```
> ALE? 1
< ALE     128
< ALE 67108864
```

Compatibility
Incompatible with existing models.

ADW <content>,<start>,<end>,<data>

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.

<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 Alternate pattern generation.
The set data overwrites the specified range.

Example
To set the addresses 0 to 1F of the pattern A to AA:

```
> ADW 0,#H0,#H1F,#HAA
```

Compatibility
Incompatible with existing models.

ADW? <content>,<start>

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 (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 Alternate pattern generation.

Example

To query the pattern A data from address 0:

```
> ADW? 0,H0
< ADW #H0123456789ABCDEF… (1 to 400 characters in Hex.)
< ADW #H0
```

Compatibility

Incompatible with existing models.

ABW <content>,<start>,<end>,<bdata>

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.

<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:

```
> ABW 0,#H0,#H1F,#11B (B=42)
```

Compatibility

Incompatible with existing models.

ABW? <content>,<start>[,<size>]

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)

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>

#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:

```
> 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>

Parameter	<content> = <DECIMAL NUMERIC PROGRAM DATA>	
	0	Pattern A
	1	Pattern B
	<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 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 ALL.	
	<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 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

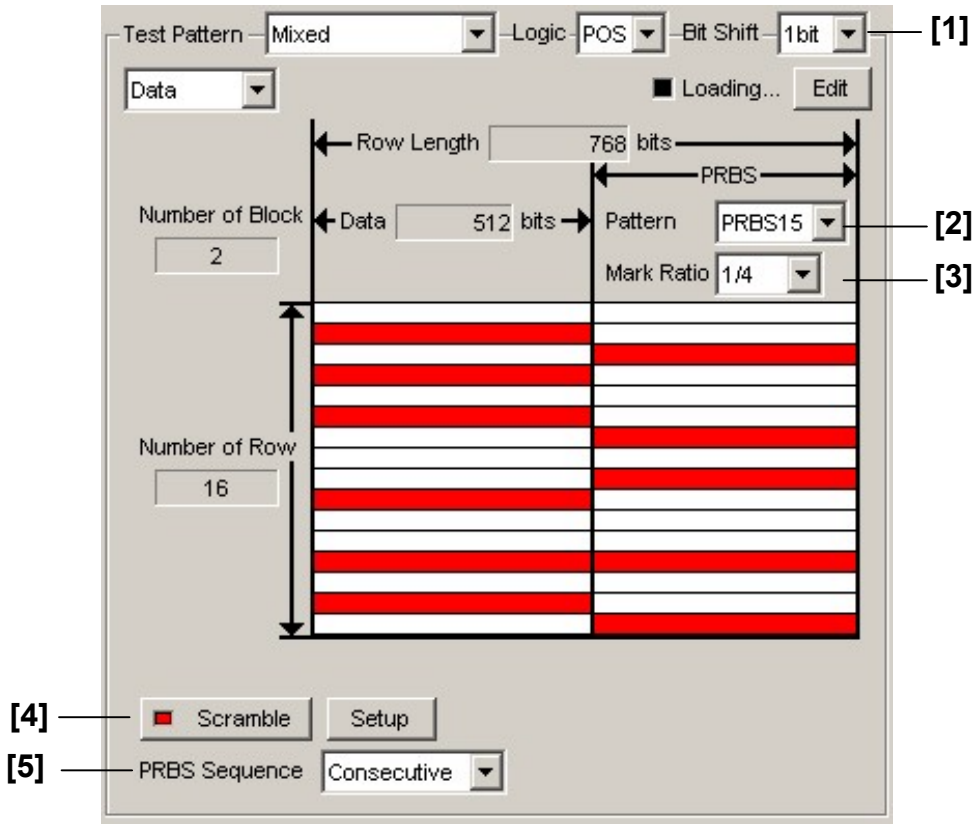


Figure 8.3.2.6-1 Mixed Data pattern setting field

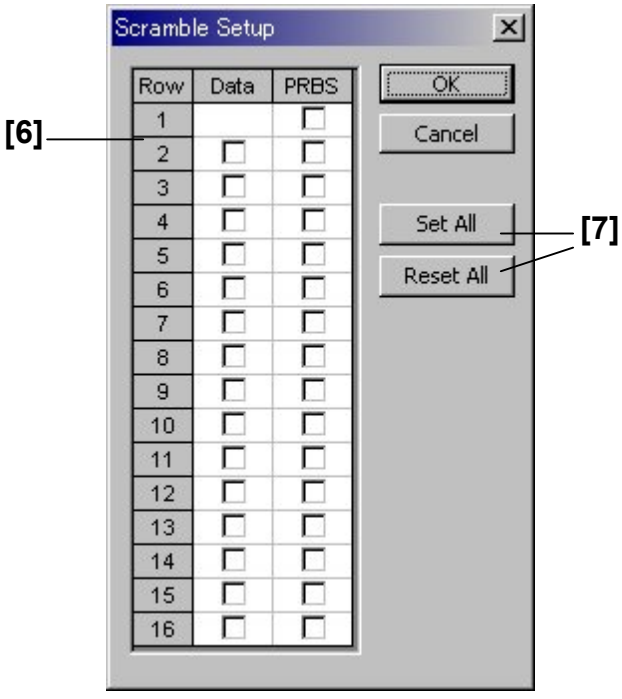


Figure 8.3.2.6-2 Descramble Setup dialog box

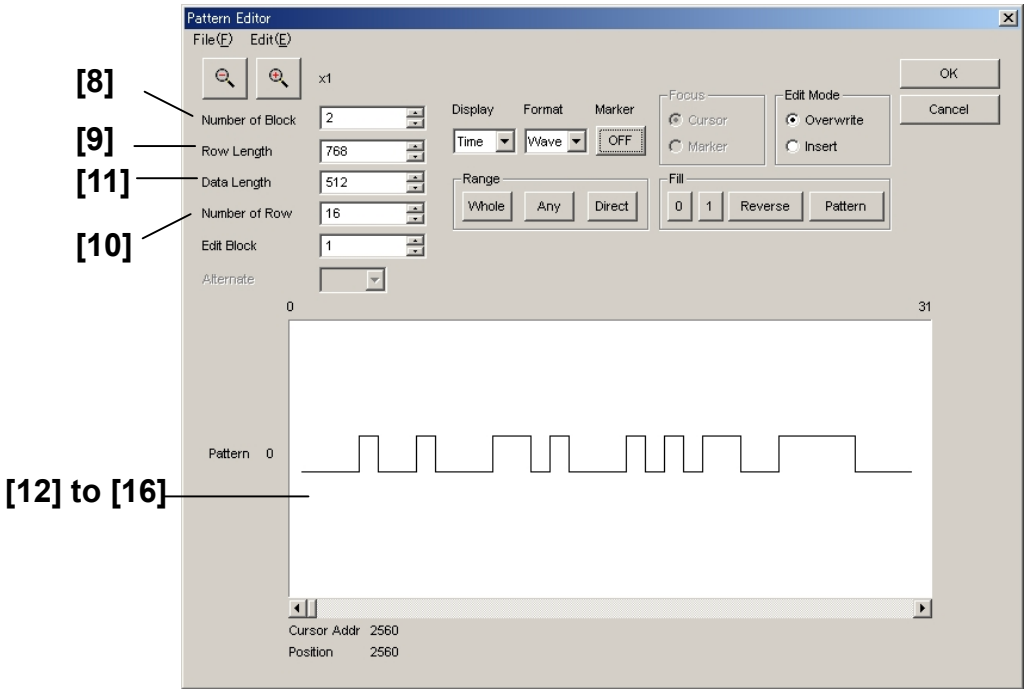


Figure 8.3.2.6-3 Mixed Data pattern setting (Pattern Editor screen)

Table 8.3.2.6-1 Mixed Data pattern setting commands

No.	Setting Items	Commands
[1]	Bit Shift	MPB
		MPB
[2]	PRBS Pattern	MPP
		MPP?
[3]	PRBS Mark Ratio	MPM
		MPM?
[4]	Scramble ON/OFF	MSC
		MSC?
[5]	PRBS Sequence	MPS
		MPS?
[6]	Scramble Setup	MSS
		MSS?
[7]	Set All/Reset All	MSL
[8]	Number of Block	MBL
		MBL?
[9]	Row Length	MRL
		MRL?
[10]	Data Length	MRW
		MRW?
[11]	Number of Row	MDL
		MDL?

Table 8.3.2.6-1 Mixed Data pattern setting commands (Cont'd)

No.	Setting Items	Commands
[12]	No label (Pattern setting)	MDW
		MDW?
[13]	No label (Pattern setting)	MBW
		MBW?
[14]	No label (Data inversion)	MDA
[15]	No label (Data inversion)	MDD
[16]	No label (All 0 or All 1)	MDF

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>	
	2	2^7-1
	3	2^9-1
	4	$2^{10}-1$
	5	$2^{11}-1$
	6	$2^{15}-1$
	7	$2^{20}-1$
	8	$2^{23}-1$
	9	$2^{31}-1$
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>		
		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 generation.		
Example	To set the mark ratio for the PRBS pattern to 1/4: `> MPM 2`		
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 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.

MPS?

Response	<sequence> = <NR1 NUMERIC RESPONSE DATA> (FIX1)
Function	Queries the PRBS pattern sequence during Mixed Data pattern generation.
Example	> MPS? < MPS 0
Compatibility	Incompatible with existing models.

MSS <row>,<data>,<prbs>

Parameter	<row> = <DECIMAL NUMERIC PROGRAM DATA> 1 to 16 1 to 16 rows, in 1-row steps <data> = <DECIMAL NUMERIC PROGRAM DATA> 0 Scramble OFF 1 Scramble ON Note, however, that Scramble is always OFF only for Data of 1 Row. <prbs> = <DECIMAL NUMERIC PROGRAM DATA> 0 Scramble OFF 1 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: > MSS 2,1,0
Compatibility	Incompatible with existing models.

MSS? <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> (FIX1)
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: > MSS? 2 < MSS 1,0
Compatibility	Incompatible with existing models.

MSL <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: `> 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 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>

Parameter	<numeric> = <DECIMAL NUMERIC PROGRAM DATA> 768 to 2281701376 768 to 2,281,701,376 bit, 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: `> MRL 768`
Compatibility	Incompatible with existing models.

MRL?

Response	<numeric> = <NR1 NUMERIC RESPONSE DATA> (FIX11)
Function	Queries the pattern length of 1 Row that is edited during Mixed Data pattern generation.
Example	> MRL? < MRL 768 < MRL 2281701376 < MRL 9126805504
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 generation.
Example	To set the number of rows within one block to 16: > MRW 16
Compatibility	Incompatible with existing models.

MRW?

Response	<numeric> = <NR1 NUMERIC RESPONSE DATA> (FIX2)
Function	Queries the number of rows within one block that are edited during Mixed Data pattern generation.
Example	> MRW? < MRW 16 < MRW 1
Compatibility	Incompatible with existing models.

MDL <numeric>

Parameter	<numeric> = <DECIMAL NUMERIC PROGRAM DATA> 512 to 134217728 512 to 134,217,728 bit, in 1-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 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.

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 × 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 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)

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 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>

#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 1 pattern data to 43:

```
> MBW 1,#H0,#H1F,#11C (C=43)
```

Compatibility

Incompatible with existing models.

MBW? <block>,<start>[,<size>]

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 #H1FFFFFFF 0 to 1FFFFFFF 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>] = <DECIMAL NUMERIC PROGRAM DATA>

1 to 16777216 1 to 16,777,216, 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 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.

MDA <block>,<start>,<end>

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.

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 — 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)

<delta> = <DECIMAL NUMERIC PROGRAM DATA>

1 to 134217728 — 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. (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 = 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

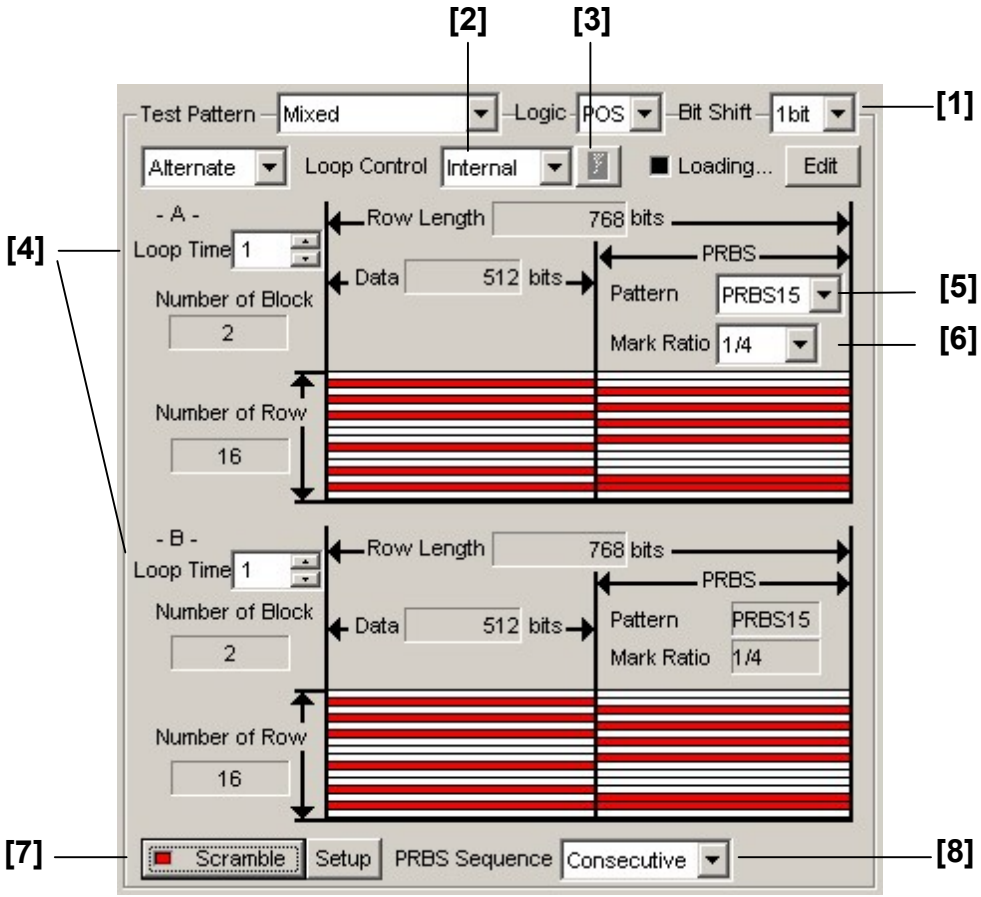


Figure 8.3.2.7-1 Mixed Alternate pattern setting field

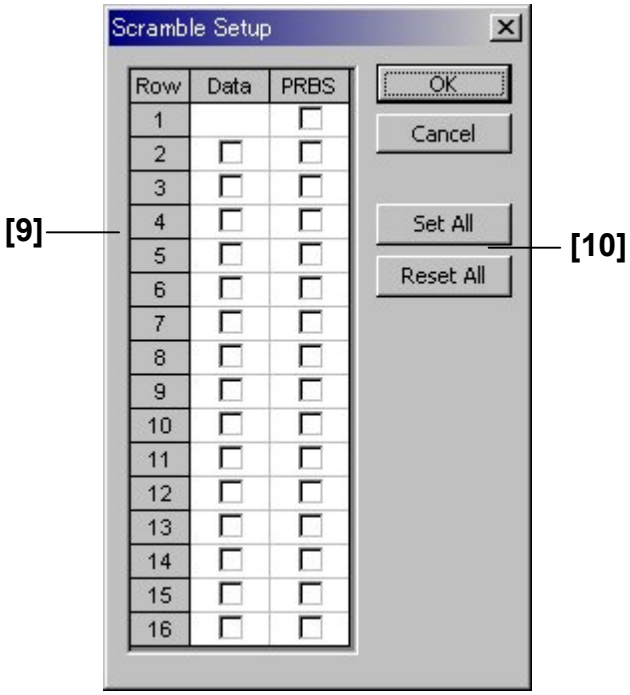


Figure 8.3.2.7-2 Descramble Setup dialog box

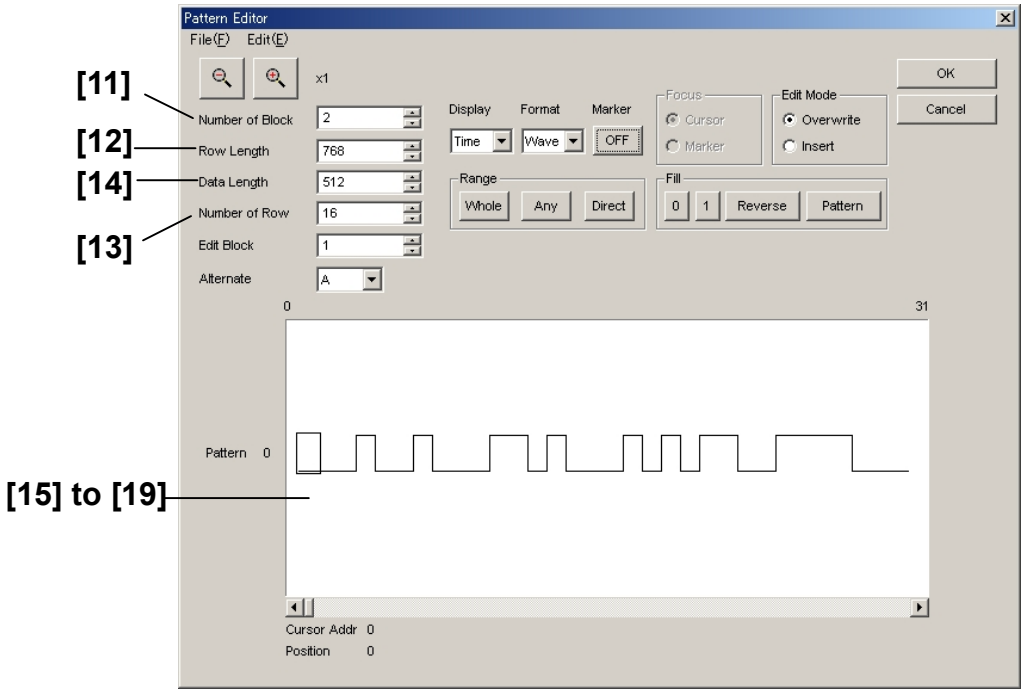


Figure 8.3.2.7-3 Mixed Alternate pattern setting (Pattern Editor screen)

Table 8.3.2.7-1 Mixed Alternate pattern setting commands

No.	Setting Items	Commands
[1]	Bit Shift	XPB
		XPB?
[2]	Loop Control	XLC
		XLC?
[3]	Manual button	XMN
[4]	Loop Time	XPT
		XPT?
[5]	Pattern	XPP
		XPP?
[6]	Mark Ratio	XPM
		XPM?
[7]	Scramble ON/OFF	XSC
		XSC?
[8]	PRBS Sequence	XPS
		XPS?
[9]	Scramble Setup	XSR
		XSR?
[10]	Set All/Reset All	XSA
[11]	Number of Block	XBL
		XBL?
[12]	Row Length	XRL
		XRL?
[13]	Data Length	XML
		XML?
[14]	Number of Row	XRW
		XRW?
[15]	No label (Pattern setting)	XDW
		XDW?
[16]	No label (Pattern setting)	XBW
		XBW?
[17]	No label (Data inversion)	XDA
[18]	No label (Data inversion)	XDD
[19]	No label (All 0 or All 1)	XDF

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

Function	Generates the pattern B for the specified number of Loop times during Mixed Alternate pattern generation.
Example	`> XMN`
Compatibility	Incompatible with existing models.

XPT <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 Mixed Alternate pattern A or B.
Example	To set the number of pattern B generation times to 20: `> XPT 1,20`
Compatibility	Incompatible with existing models.

XPT? <content>

Parameter	<content> = <DECIMAL NUMERIC PROGRAM DATA> (FIX3) 0 Pattern A 1 Pattern B
Response	<numeric> = <NR1 NUMERIC RESPONSE DATA>
Function	Queries the number of generation times of the Mixed Alternate pattern A or B.

Example	To set the number of pattern B generation times:
	`> XPT? 1`
	`< XPT 1`
	`< XPT 511`

Compatibility	Incompatible with existing models.

XPP <numeric>

Parameter

<numeric> = <DECIMAL NUMERIC PROGRAM DATA>

2	2^7-1
3	2^9-1
4	$2^{10}-1$
5	$2^{11}-1$
6	$2^{15}-1$
7	$2^{20}-1$
8	$2^{23}-1$
9	$2^{31}-1$

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:

```
> XPP 2
```

Compatibility

Incompatible with existing models.

XPP?

Response

<numeric> = <NR1 NUMERIC RESPONSE DATA> (FIX1)

Function

Queries the number of stages of the PRBS pattern during Mixed Alternate pattern generation.

Example

```
> XPP?
< XPP 2
```

Compatibility

Incompatible with existing models.

XPM <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 Alternate pattern generation.

Example

To set the mark ratio for the PRBS pattern to 1/8:

```
> XPM 1
```

Compatibility

Incompatible with existing models.

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.

XPS?

Response	<sequence> = <NR1 NUMERIC RESPONSE DATA> (FIX1)
Function	Queries the PRBS pattern sequence during Mixed Alternate pattern generation.
Example	> XPS? < XPS 0
Compatibility	Incompatible with existing models.

XSR <row>,<data>,<prbs>

Parameter	<row> = <DECIMAL NUMERIC PROGRAM DATA> 1 to 16 1 to 16 rows, in 1-row steps <data> = <DECIMAL NUMERIC PROGRAM DATA> 0 Scramble OFF 1 Scramble ON Note, however, that Data of 1 Row Scramble is fixed to Scramble OFF. <prbs> = <DECIMAL NUMERIC PROGRAM DATA> 0 Scramble OFF 1 Scramble ON
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: > XSR 3,1,0
Compatibility	Incompatible with existing models.

XSR? <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> (FIX1)
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: > XSR? 3 < XSR 1,0
Compatibility	Incompatible with existing models.

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	<numeric> = <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	<numeric> = <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	<numeric> = <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	<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: > XRW 8
Compatibility	Incompatible with existing models.

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 × 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 × 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 B data from address 0:

```
> 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 <block>,<content>,<start>,<end>

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)

<end> = <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.

Function

Inverts the data in the program during Mixed Alternate pattern generation. Specify the inversion range by the <start> and <end> addresses.

Example

To invert addresses 0 to 1F of the block 1 pattern B:

```
> XDA 1,1,#H0,#H1F
```

Compatibility

Incompatible with existing models.

XDD <block>,<content>,<start>,<delta>

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)

<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 Mixed 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 block 1 pattern B from address 0:

```
> XDD 1,1,#H0,64
```

Compatibility

Incompatible with existing models.

XDF <block>,<content>,<range>,<page>,<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

<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 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

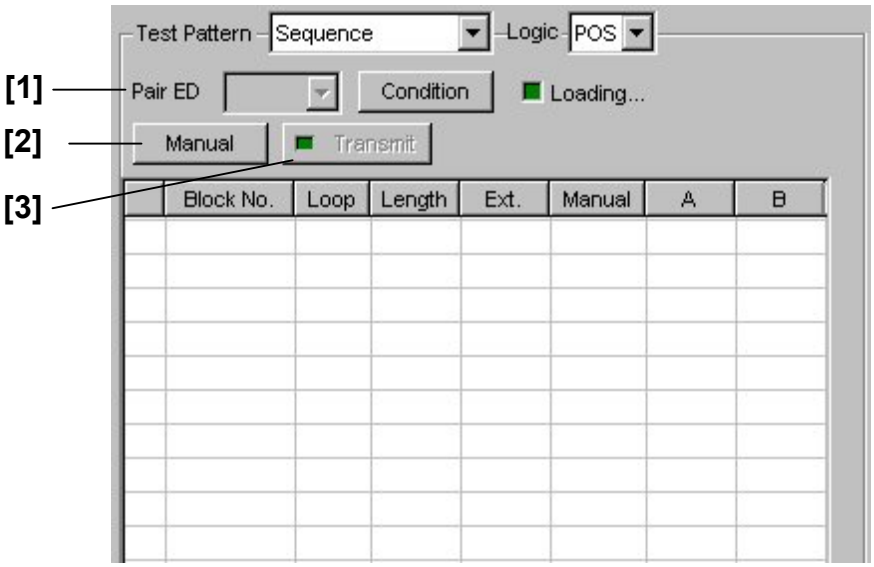


Figure 8.3.2.8-1 Sequence pattern setting field

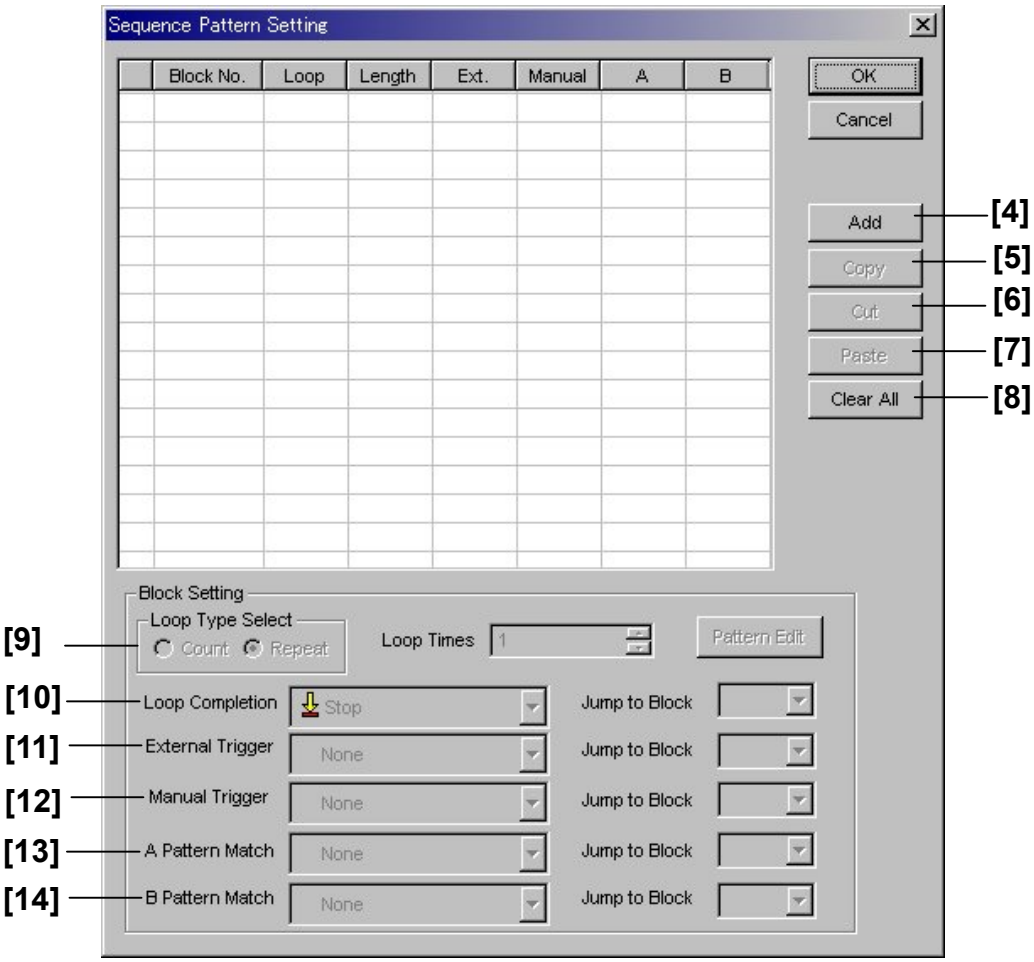


Figure 8.3.2.8-2 Sequence Pattern Setting screen

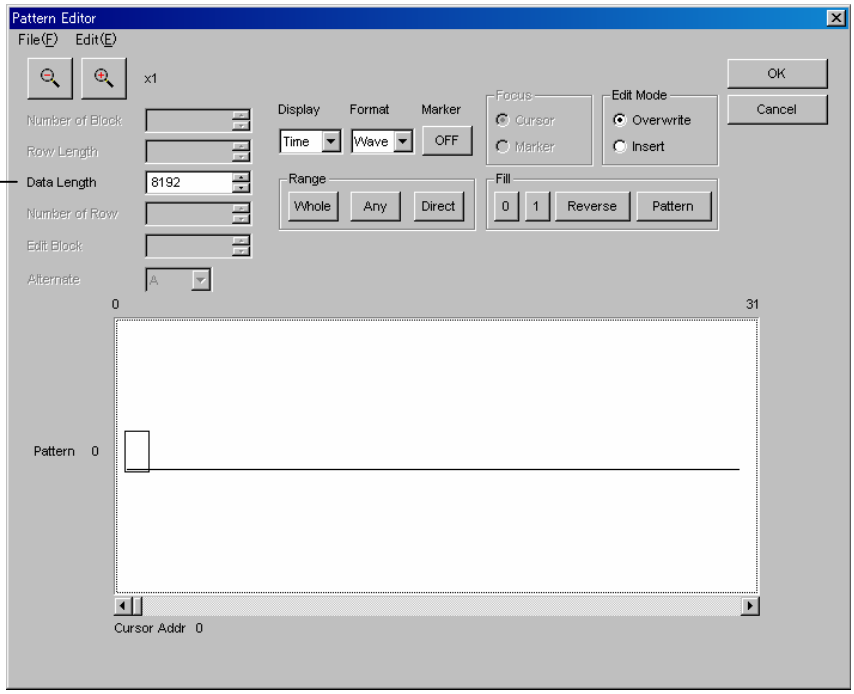


Figure 8.3.2.8-3 Sequence pattern setting (Pattern Editor screen)

Table 8.3.2.8-1 Sequence pattern setting commands

No.	Setting Items	Commands
[1]	Pair ED	SPR
		SPR?
[2]	Manual	SMA
[3]	Transmit	SRE
[4]	Add	SAD
		SAD?
[5]	Copy	SCO
[6]	Cut	SCU
[7]	Paste	SPA
[8]	Clear All	SCD
[9]	Loop times	SLP
		SLP?
[10]	Loop Completion	SCL
		SCL?
[11]	External Trigger	SCE
		SCE?
[12]	Manual Trigger	SCM
		SCM?
[13]	A Pattern Match	SCA
		SCA?
[14]	B Pattern Match	SCB
		SCB?

Table 8.3.2.8-1 Sequence pattern setting commands (Cont'd)

No.	Setting Items	Commands
[15]	Data Length	SLG
		SLG?
[16]	No label (Pattern setting)	SDW
		SDW?
[17]	No label (Pattern setting)	SBW
		SBW?
[18]	No label (Data inversion)	SDA
[19]	No label (Data inversion)	SDD
[20]	No label (All 0 or All 1)	SDF

SPR <numeric>

Parameter <numeric> = <DECIMAL NUMERIC PROGRAM DATA>
0 to 6 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:
```
> 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
```
> SPR?
< SPR 2
```

Compatibility Incompatible with existing models.

SMA

Function Generates a Sequence pattern that is set manually.

Example
```
> SMA
```

Compatibility Incompatible with existing models.

SRE

Function Restarts the Sequence pattern generation.

Example
```
> SRE
```

Compatibility Incompatible with existing models.

SAD <block>

Parameter	<block> = <DECIMAL NUMERIC PROGRAM DATA> 1 to 128 1 to 128 blocks, in 1-block steps
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?

Response	<block> = <NR1 NUMERIC RESPONSE DATA> (FIX3)
Function	Queries the number of blocks that are edited during Sequence pattern generation.

Example	`> SAD?`
	`< SAD 1`
	`< SAD 128`

Compatibility	Incompatible with existing models.

SCO <block>

Parameter	<block> = <DECIMAL NUMERIC PROGRAM DATA> 1 to 128 1 to 128 blocks, in 1-block steps
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>

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: `> SCU 15`
Compatibility	Incompatible with existing models.

SPA <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 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	<block> = <DECIMAL NUMERIC PROGRAM DATA> 1 to 128 1 to 128 blocks, in 1-block steps <type> = <DECIMAL NUMERIC PROGRAM DATA> 0 Specifies the number of repetition times. 1 Does not specify the number of repetition times. [<numeric>] = <DECIMAL NUMERIC PROGRAM DATA> 1 to 1024 1 to 1,024 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.

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
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 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.

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.

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.

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 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 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 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 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>

Parameter	<block> = <DECIMAL NUMERIC PROGRAM DATA>
	1 to 128 1 to 128 blocks, in 1-block steps
Response	<b> = <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 pattern B match condition of each block during Sequence pattern generation.
Example	To query the pattern B match condition of block 7:
	`> SCB? 7`
	`< SCB 2`
	`< SCB 1, 10`
	`< SCB 1,128`
Compatibility	Incompatible with existing models.

SLG <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 8,192 to 1,048,576 bits, in 128-bit steps
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:
	`> SLG 3,8192`
Compatibility	Incompatible with existing models.

SLG? <block>

Parameter	<block> = <DECIMAL NUMERIC PROGRAM DATA>
	1 to 128 1 to 128 blocks, in 1-block steps
Response	<numeric> = <NR1 NUMERIC RESPONSE DATA> (FIX7)
	8192 to 1048576 8,192 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:
	`> SLG? 3`
	`< SLG 8192`
	`< 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 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 × 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 #HFFFFF 0 to FFFFF bits, 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 #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.

<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,#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 NUMERIC PROGRAM DATA>

#H0 to #HFFFFF 0 to FFFFF 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 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.

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.

SDD <block>,<start>,<delta>

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)

<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-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 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

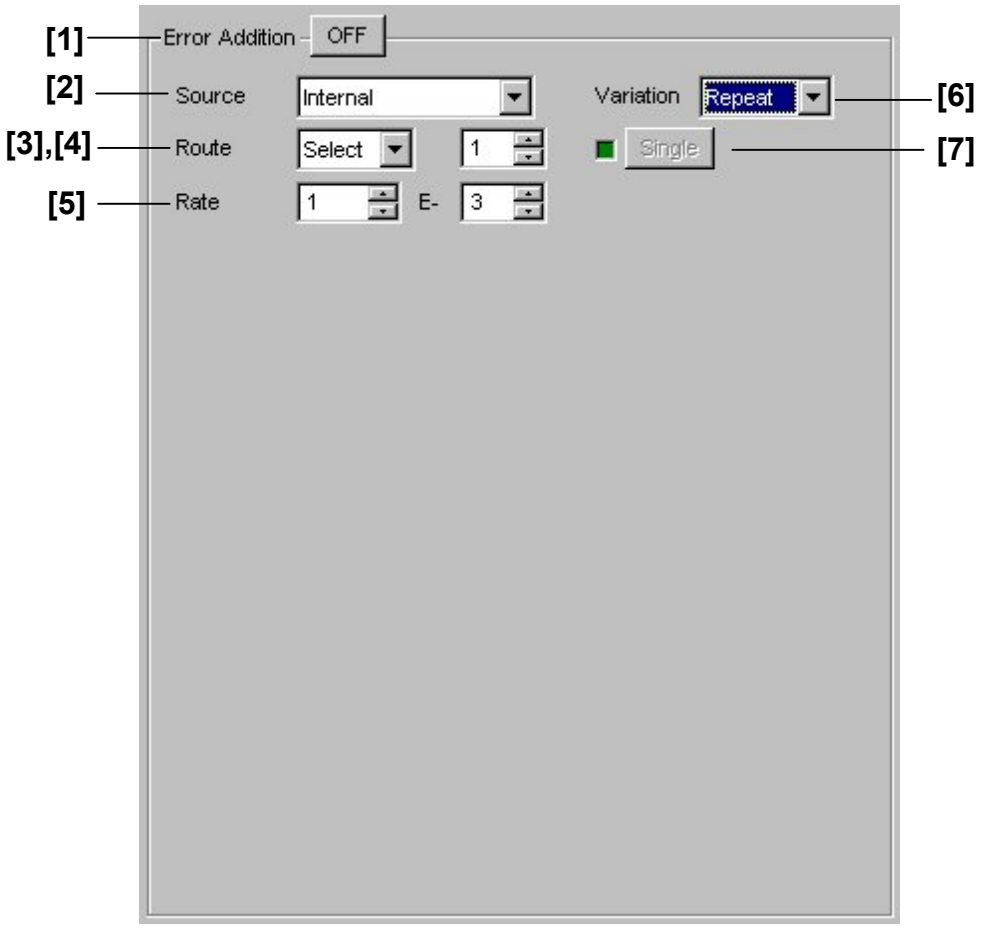


Figure 8.3.3-1 Error Addition setting field (For PRBS, Zero-Substitution, Data patterns)

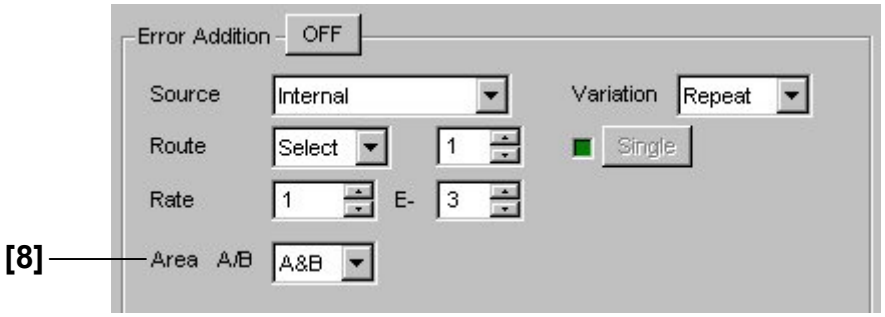


Figure 8.3.3-2 Error Addition setting field (For Alternate pattern)

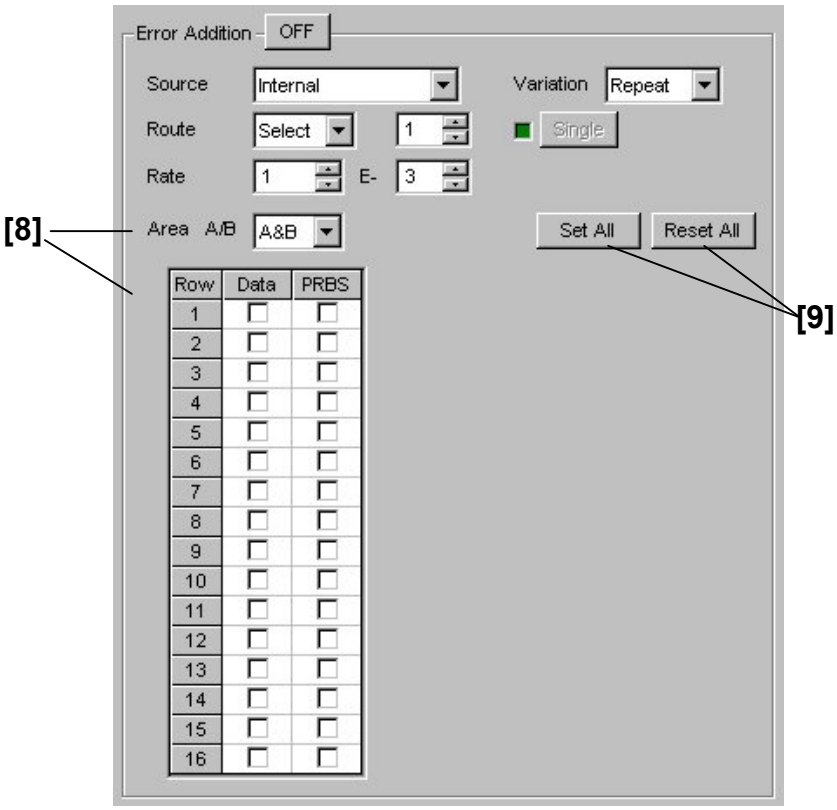


Figure 8.3.3-3 Error Addition setting field (For Mixed pattern)

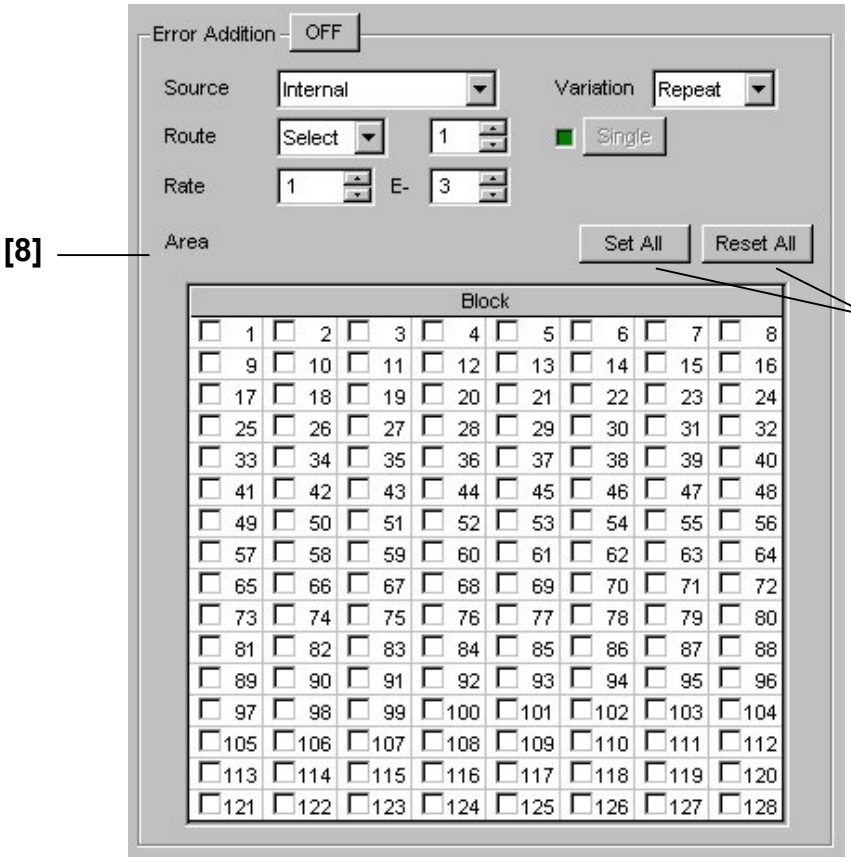


Figure 8.3.3-4 Error Addition setting field (For Sequence pattern)

Table 8.3.3-1 Error Addition tab setting commands

No.	Setting Items	Commands
[1]	Error Addition ON, OFF	EAD
		EAD?
[2]	Source	EEI
		EEI?
[3]	Route	ESL
		ESL?
[4]	Error Addition Route setting	ECH
		ECH?
[5]	Rate	ERT
		ERT?
[6]	Variation	EAV
		EAV?
[7]	Single	ESI
[8]	Area A/B (Error addition area setting)	EAR
		EAR?
[9]	Set All/Reset All	EAA

EAD <numeric>

Parameter	<numeric> = <DECIMAL NUMERIC PROGRAM DATA>
	0 Bit error addition OFF
	1 Bit error addition ON
	7 Single error addition
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?

Response	<numeric> = <NR1 NUMERIC RESPONSE DATA> (FIX1)
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>

Parameter	<source> = <DECIMAL NUMERIC PROGRAM DATA>
	0 Internal: Internal signal
	1 External Trigger: Rising of external signal trigger
	2 External Disable: Disables 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: `> EEI 0`
Compatibility	Compatible with the MP1761C.

EEI?

Response	<source> = <NR1 NUMERIC RESPONSE DATA> (FIX1)
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>

Parameter	<select> = <DECIMAL NUMERIC PROGRAM DATA>		
	0	Scan:	Changes the route for which 32 Demux of 1/1 signal is performed every time when an error is added.
	1	Select:	Adds an error to the specified route.
Function	Sets the route addition method for bit error addition.		
Example	To set the route addition method to Scan: `> ESL 0`		
Compatibility	Incompatible with existing models.		

ESL?

Response	<select> = <NR1 NUMERIC RESPONSE DATA> (FIX1)
Function	Queries the route addition method for bit error addition.
Example	`> ESL?` `< ESL 0`
Compatibility	Incompatible with existing models.

ECH <numeric>

Parameter	<numeric> = <DECIMAL NUMERIC PROGRAM DATA>	
	1 to 32	1 to 32, in single steps
Function	Sets the route to add a bit error.	
Example	To set Route 2 as the route to add a bit error: `> ECH 2`	
Compatibility	Compatible with the MP1632C (Routes 1 to 8 only). Compatible with the MP1761C and MP1775A.	

ECH?

Response	<numeric> = <NR1 NUMERIC RESPONSE DATA> (FIX2)
Function	Queries the route to add a bit error.

Example	`> ECH?`
	`< ECH  1`
	`< ECH 32`

Compatibility	Compatible with the MP1632C (Routes 1 to 8 only). Compatible with the MP1761C and MP1775A.

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	`> 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.

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:
	`> EAR? 0,1`
	`< EAR 1,1`
	To query the error addition ON/OFF state of Pattern A, for Alternate pattern:
	`> 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: `> EAA 1`
Compatibility	Incompatible with existing models.

8.3.4 Commands related to Misc tab

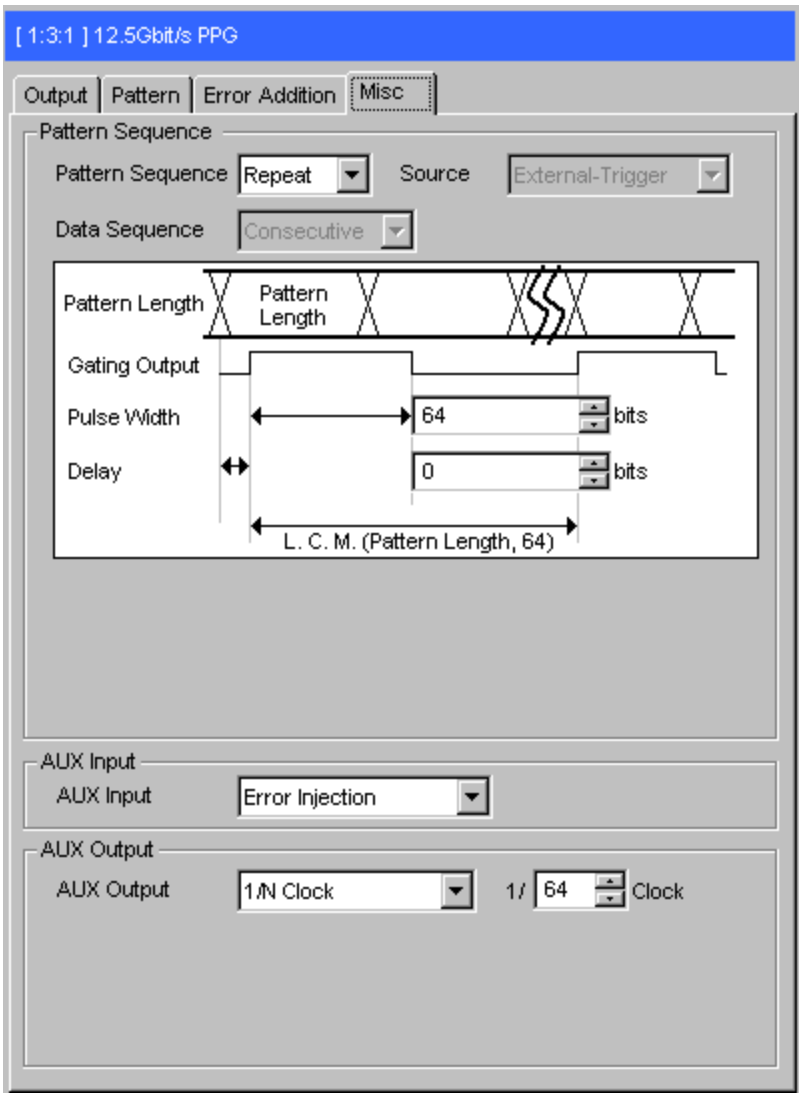


Figure 8.3.4-1 Misc tab

8.3.4.1 Pattern Generation setting commands

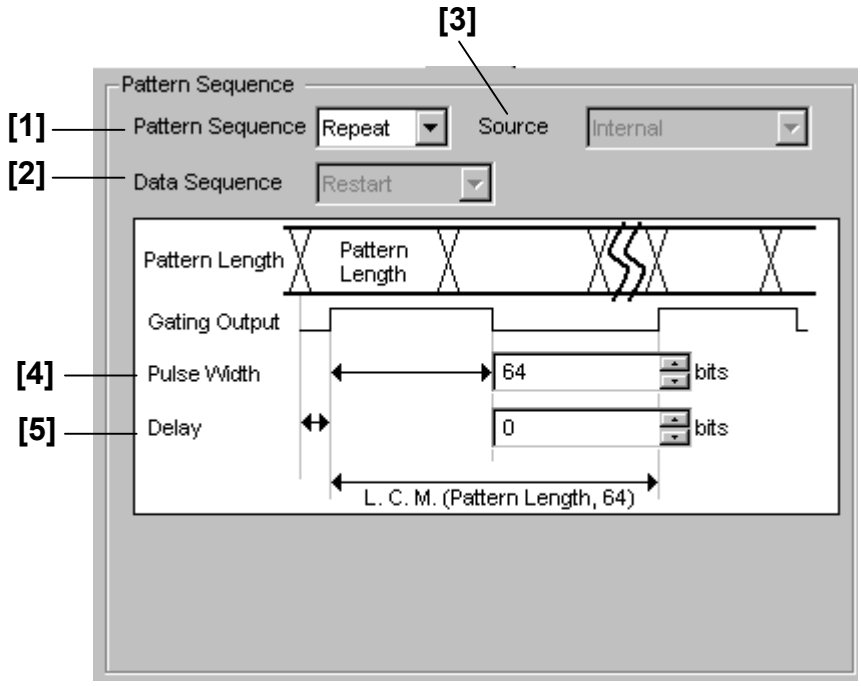


Figure 8.3.4.1-1 Pattern Generation setting field (for Repeat signal generation)

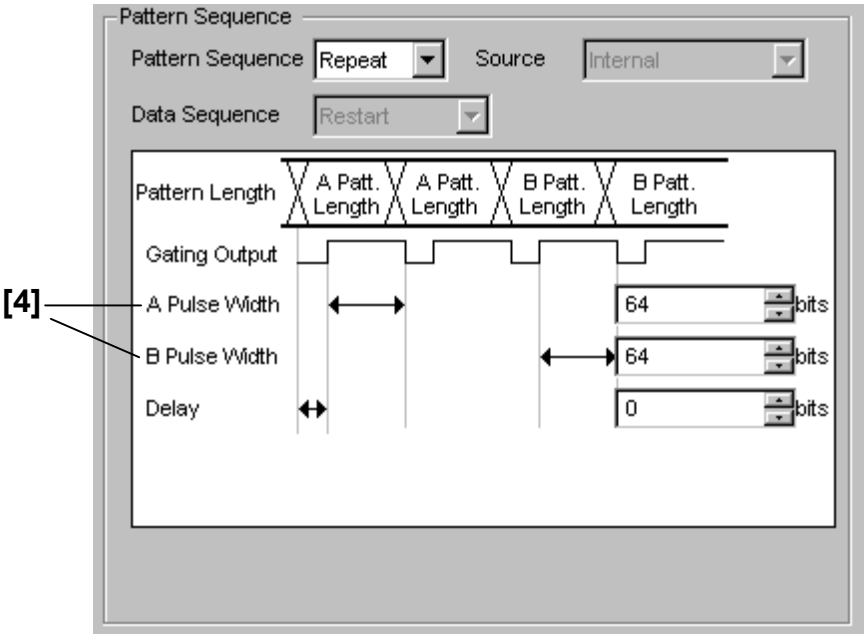


Figure 8.3.4.1-2 Pattern Generation setting field (for Repeat signal generation, Alternate pattern)

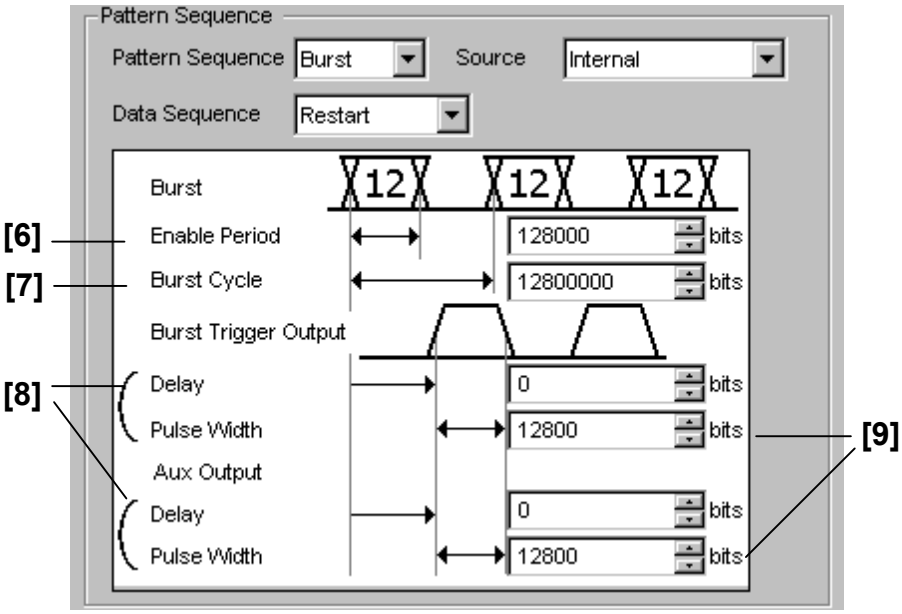


Figure 8.3.4.1-3 Pattern Generation setting field (for Burst signal generation)

Table 8.3.4.1-1 Pattern Generation setting commands

No.	Setting Items	Commands
[1]	Pattern Sequence	PTO
		PTO?
[2]	Data Sequence	BSQ
		BSQ?
[3]	Source	BRM
		BRM?
[4]	Pulse Width	RPW
		RPW?
[5]	Delay (For Repeat signal generation)	RTD
		RTD?
[6]	Enable Period	BRE
		BRE?
[7]	Burst Cycle	BRC
		BRC?
[8]	Delay (For Burst signal generation)	TGD
		TGD?
[9]	Enable Pulse Width	PWD
		PWD?

PTO <mode>

Parameter	<mode> = <DECIMAL NUMERIC PROGRAM DATA>
	0 Repeat signal
	1 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: `> PTO 1`
Compatibility	Compatible with the MP1632C.

PTO?

Response	<mode> = <NR1 NUMERIC RESPONSE DATA> (FIX1)
Function	Queries the signal generation method.
Example	`> PTO?` `< PTO 1`
Compatibility	Compatible with the MP1632C.

BSQ <mode>

Parameter	<mode> = <DECIMAL NUMERIC 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 signal.
Example	To set the output sequence to Restart: `> BSQ 0`
Compatibility	Incompatible with existing models.

BSQ?

Response	<mode> = <NR1 NUMERIC RESPONSE DATA> (FIX1)
Function	Queries the data output sequence for the Burst signal.
Example	`> BSQ?` `< BSQ 0`
Compatibility	Incompatible with existing models.

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>]

Parameter
[<content>] = <DECIMAL NUMERIC PROGRAM DATA> (FIX12)
0 Pattern A
1 Pattern B
Note:
For patterns other than the Alternate pattern, <content> cannot be set and should be omitted.

Response
<numeric> = <NR1 NUMERIC RESPONSE DATA>

Function
Queries the pulse width of the timing signal to be output during the timing signal period.

Example
```
> RPW?
< RPW         64
< RPW 2281701312
```

Compatibility
Incompatible with existing models.

RTD <numeric>

Parameter
<numeric> = <DECIMAL NUMERIC PROGRAM DATA>
0 to 68719476672 0 to 68,719,476,672 bits, in 16-bit steps
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.

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:
```
> RTD 256
```

Compatibility
Incompatible with existing models.

RTD?

Response
<numeric> = <NR1 NUMERIC RESPONSE DATA> (FIX12)

Function
Queries the delay value for the timing signal pulse to be output during the timing signal period.

Example
```
> RTD?
< RTD          16
< RTD 2281701312
```

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.

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 8589934592`

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.

PWD <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 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>

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)

0 to 21474835848 0 to 2,147,483,584 bit

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

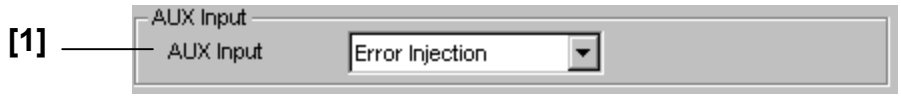


Figure 8.3.4.2-1 AUX Input setting field

Table 8.3.4.2-1 AUX Input setting commands

No.	Setting Item	Commands
[1]	AUX Input	FNC
		FNC?

FNC <function>

Parameter	<function> = <DECIMAL NUMERIC PROGRAM DATA> 0 Burst Source 1 Error Injection 2 Sequence Control 3 Alternate Control Set Error Injection in the case of Repeat.
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> = <NR1 NUMERIC RESPONSE DATA> (FIX1)
Function	Queries the function of AUX Input.
Example	`> FNC?` `< FNC 1`
Compatibility	Incompatible with existing models.

8.3.4.3 AUX Output setting commands

Figure 8.3.4.3-1 AUX Output setting field

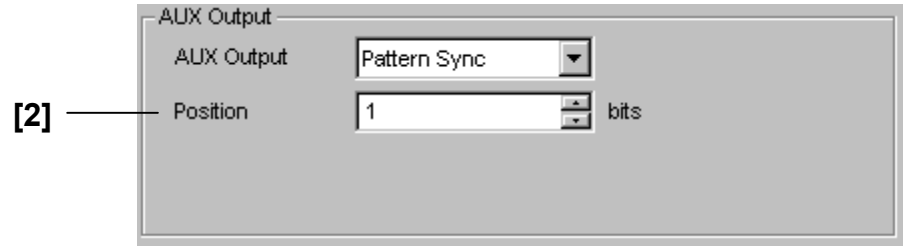


Figure 8.3.4.3-2 AUX Output setting field

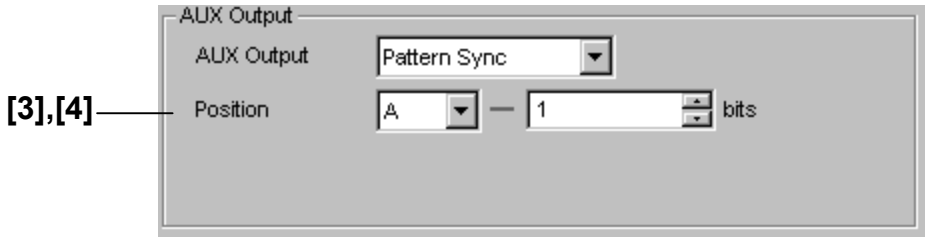


Figure 8.3.4.3-3 AUX Output setting field
(For Alternate pattern)

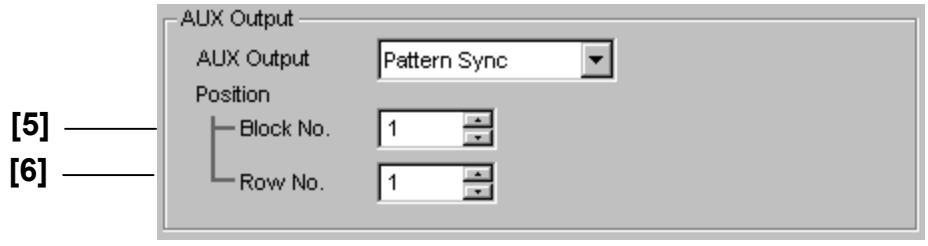


Figure 8.3.4.3-4 AUX Output setting field
(For Mixed Data pattern)

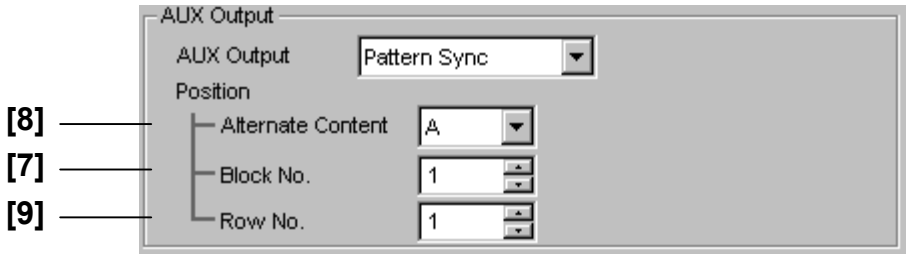


Figure 8.3.4.3-5 AUX Output setting field (For Mixed Alternate pattern)

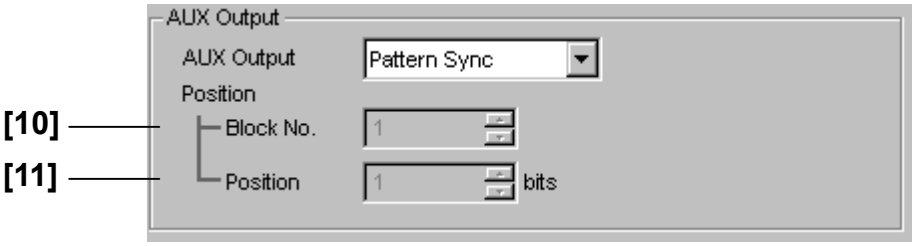


Figure 8.3.4.3-6 AUX Output setting field (For Sequence pattern)

Table 8.3.4.3-1 AUX Output setting commands

No.	Setting Items	Commands
[1]	AUX Output	SOP
		SOP?
[2]	Position	PSP
		PSP?
[3]	A/B switching (Alternate)	SCN
		SCN?
[4]	Position	APO
		APO?
[5]	Block No. (Mixed Data)	SBL
		SBL?
[6]	Row No. (Mixed Data)	SRW
		SRW?
[7]	Block No. (Mixed Alternate)	SML
		SML?
[8]	Alternate Content (Mixed Alternate)	SXO
		SXO?
[9]	Row No. (Mixed Alternate)	SXW
		SXW?
[10]	Block No. (Sequence)	SQB
		SQB?
[11]	Position (Sequence)	SQP
		SQP?

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.

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?

Response	<content> = <NR1 NUMERIC RESPONSE DATA> (FIX1)
Function	Queries the output position for synchronization output for Alternate pattern.
Example	> SCN? < SCN 0
Compatibility	Incompatible with existing models.

APO <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: > APO 33
Compatibility	Incompatible with existing models.

APO?

Response	<numeric> = <NR1 NUMERIC RESPONSE DATA> (FIX8)
Function	Queries the output position for synchronization output for Alternate pattern.
Example	> APO? < APO 1 < APO 67108785
Compatibility	Incompatible with existing models.

SBL <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: > SBL 10
Compatibility	Incompatible with existing models.

SBL?

Response	<numeric> = <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	<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 Data pattern.
Example	To set the synchronization output Row to 1 Row: > SRW 1
Compatibility	Incompatible with existing models.

SRW?

Response	<numeric> = <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	<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 Alternate pattern.
Example	To set the synchronization output block to block 10: > SML 10
Compatibility	Incompatible with existing models.

SML?

Response	<numeric> = <NR1 NUMERIC RESPONSE DATA> (FIX3)
Function	Queries the output block position for synchronization output for Mixed Alternate pattern.
Example	> SML? < SML 1 < SML 511
Compatibility	Incompatible with existing models.

SXO <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 Mixed Alternate pattern.
Example	To set the synchronization output position to the top of Pattern A: > SXO 0
Compatibility	Incompatible with existing models.

SXO?

Response	<content> = <NR1 NUMERIC RESPONSE DATA> (FIX1)
Function	Queries the output position for synchronization output for Mixed Alternate pattern.
Example	> SXO? < SXO 0
Compatibility	Incompatible with existing models.

SXW <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: > SXW 2
Compatibility	Incompatible with existing models.

SXW?

Response	<numeric> = <NR1 NUMERIC RESPONSE DATA> (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	<numeric> = <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	<numeric> = <NR1 NUMERIC RESPONSE DATA> (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	<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 1,048,497: > SQP 1048497
Compatibility	Incompatible with existing models.

SQP?

Response	<numeric> = <NR1 NUMERIC RESPONSE DATA> (FIX7)
Function	Queries the output position for synchronization output for Sequence pattern.
Example	> SQP? < SQP 1 < SQP 1048497
Compatibility	Incompatible with existing models.

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

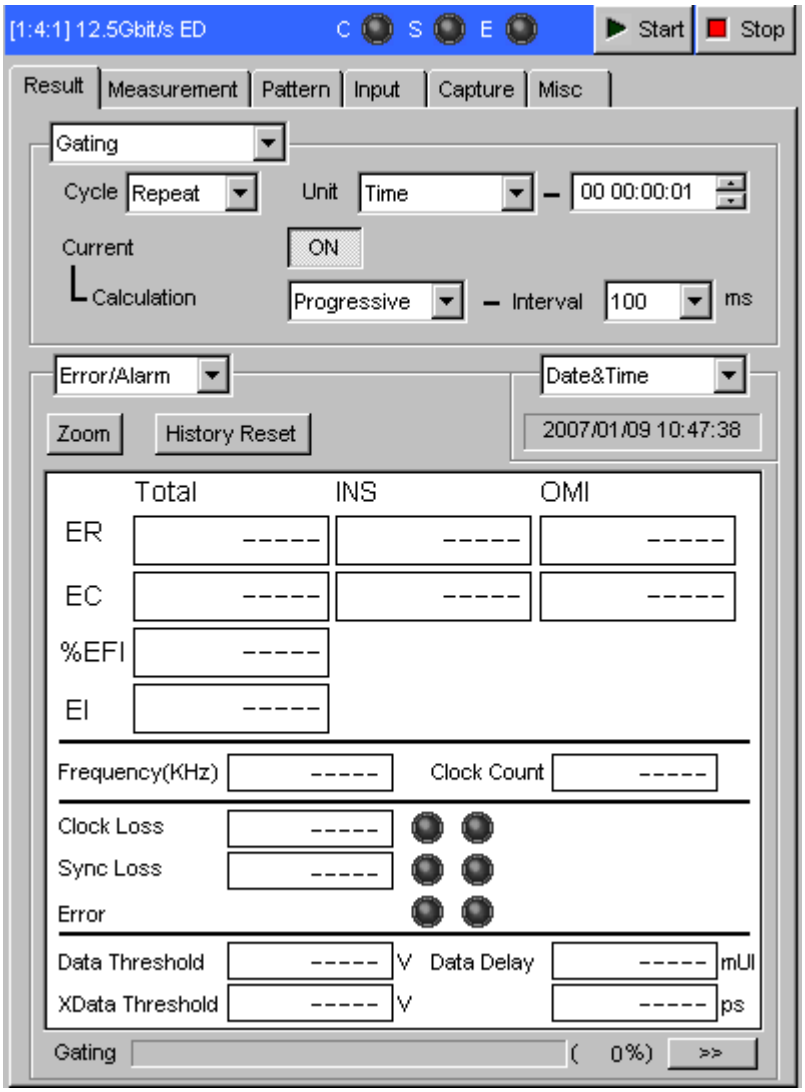


Figure 8.4.1-1 Result tab

8.4.1.1 Result setting commands

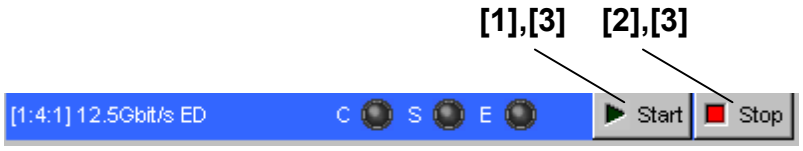


Figure 8.4.1.1-1 Start/Stop setting

Figure 8.4.1.1-2 Result setting field

Table 8.4.1.1-1 Result setting commands

No.	Setting Items	Commands
[1]	Start	STA
[2]	Stop	STO
[3]	Query for measurement status	MSR?
[4]	Switching of measurement result display	SET
		SET?
[5]	Switching of time display	TIM
		TIM?
[6]	Query for measurement start time	MSA?
[7]	Query for measurement end time	MSO?
[8]	Query for measurement elapsed time	MLP?
[9]	Query for measurement remaining time	ETI?
[10]	Query for intermediate data creation time	INT?
[11]	Query for alarm occurrence time	AMD?

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	<time> = <DECIMAL NUMERIC PROGRAM DATA>	
	0	Displays the current date and time (Date&Time).
	1	Displays the current date and time (Date&Time).
	3	Displays the remaining time based on the measurement period (Remain Time).
	4	Displays the elapsed time based on the measurement period (Elapsed Time).
	5	Displays the measurement start time (Start Time).
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) 0, 3, 4, 5
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	`> 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	`> ETI?`		
	`< ETI 00,00,00,00`		
Compatibility	Incompatible with existing models.		

INT?

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 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	<alarm> = <DECIMAL NUMERIC PROGRAM DATA>		
	0	Outputs all item data	
	3	Clock Loss occurrence time	
	4	Clock Loss recovery time	
	5	Pattern Sync Loss occurrence time	
	6	Pattern Sync Loss recovery time	
	7	CR Unlock occurrence time	
	8	CR Unlock recovery time	
Response	<year>,<month>,<day>,<hour>,<min>,<second> = <NR1 NUMERIC RESPONSE DATA>		
	<year>	00 to 36	Year 2000 to 2036 (FIX2)
	<month>	01 to 12	January to December (FIX2)
	<day>	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 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`		

```
< 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 01:01:01,06-01-01 02:01:01
```

Compatibility

Compatible with the MP1762D.

8.4.1.2 Error/Alarm setting commands

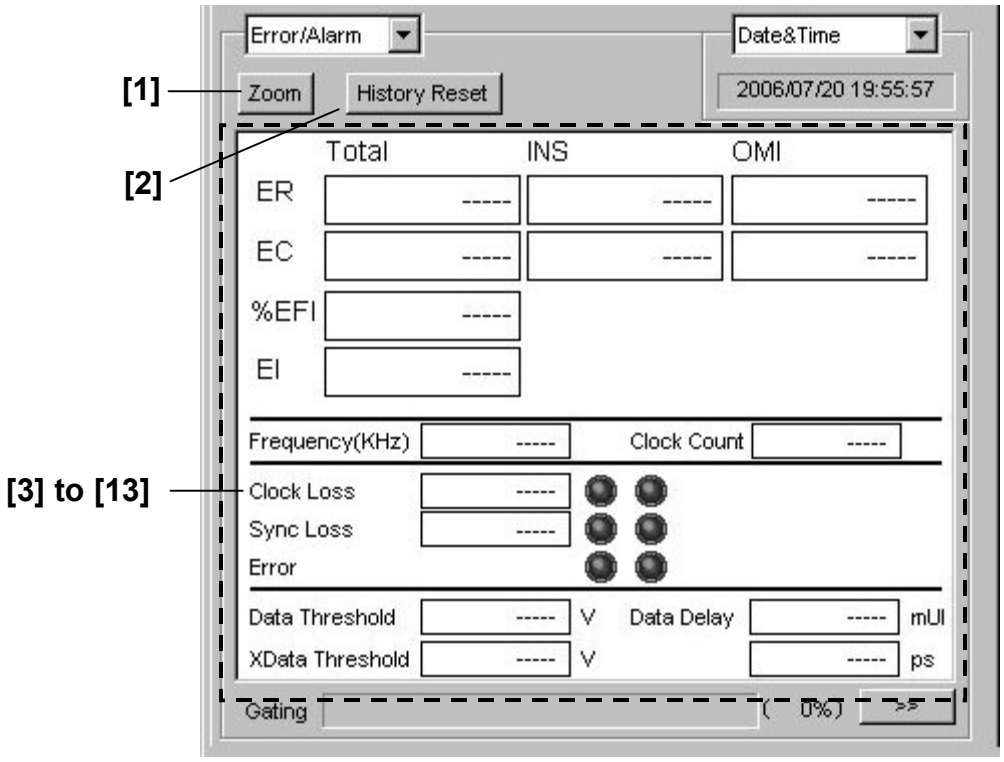


Figure 8.4.1.2-1 Error/Alarm result display screen

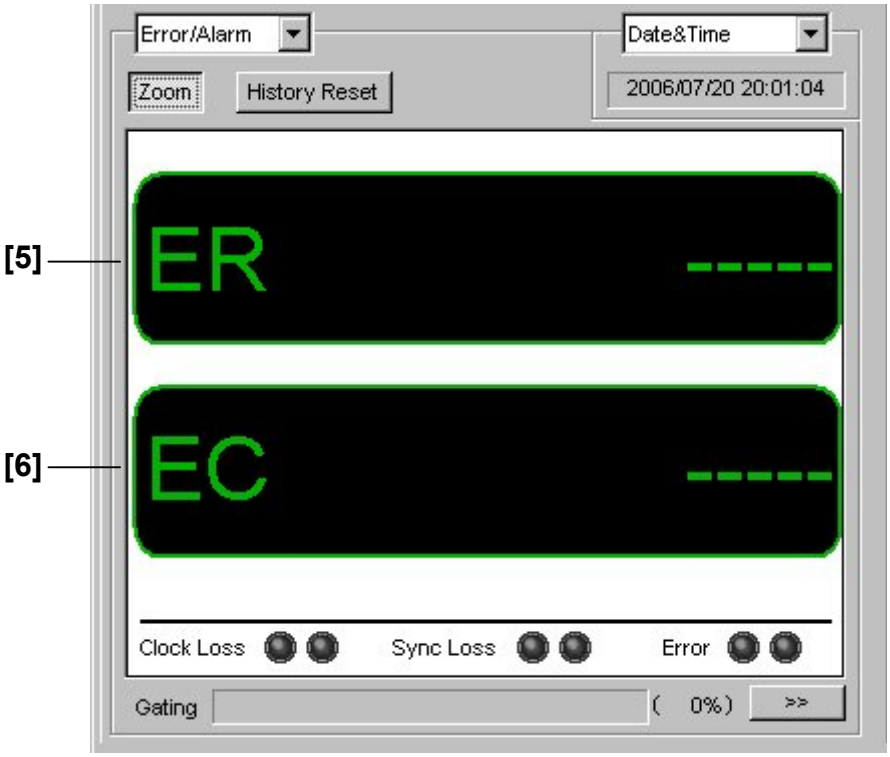


Figure 8.4.1.2-2 Error/Alarm result display screen (when Zoom is selected)

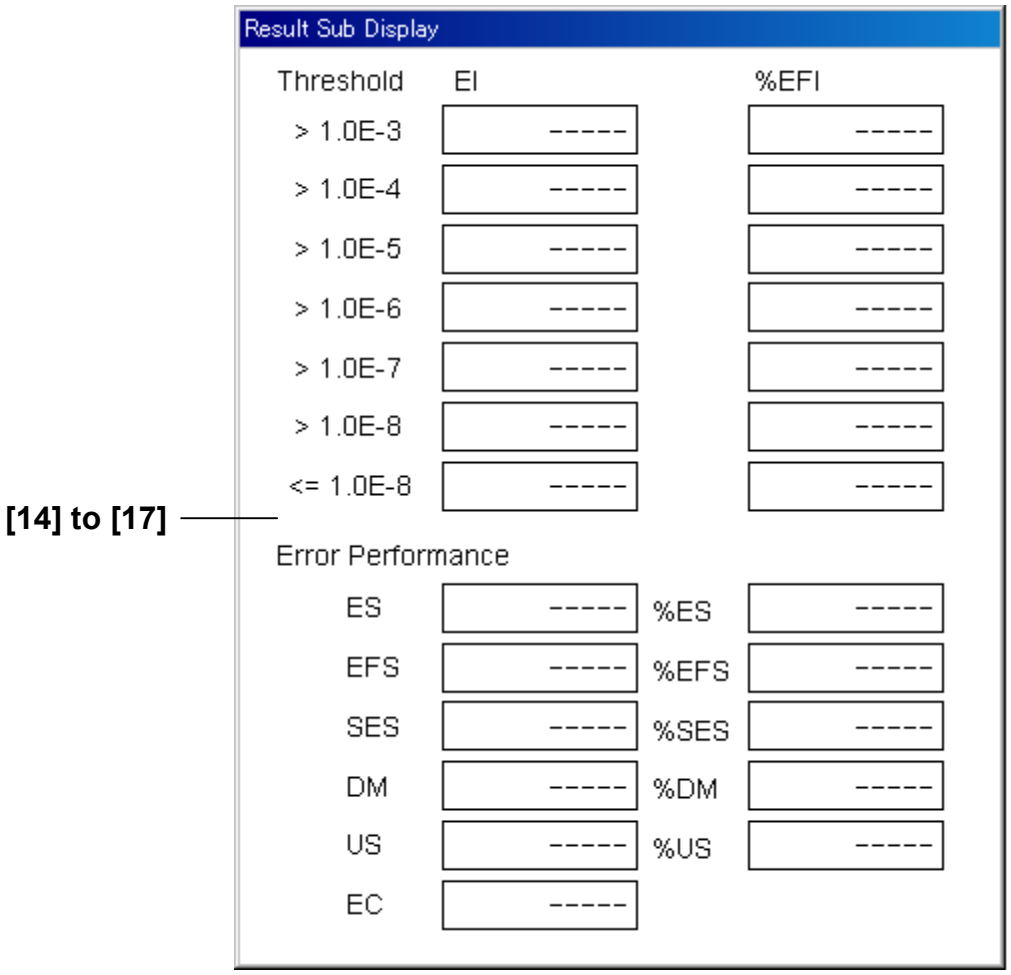


Figure 8.4.1.2-3 Error/Alarm result display screen (Result Sub Display)

Table 8.4.1.2-1 Error/Alarm setting commands

No.	Setting Items	Commands
[1]	Zoom	ZOM
		ZOM?
[2]	History Reset	HRE
[3]	No label (Query for measurement data)	END?
[4]	No label (Query for monitoring item)	MTR?
[5]	ER (Total)	ER?
[6]	EC (Total)	EC?
[7]	%EFI	EFI?
[8]	EI	EI?
[9]	Frequency (kHz)	FRQ?
[10]	Clock Count	CC?
[11]	Clock Loss	CLI?
[12]	Sync Loss	SLI?
[13]	Error	ERS?
[14]	Error Performance	PFP?
[15]	Error Performance %	PFC?
[16]	Threshold EI	THE?
[17]	Threshold %EFI	THF?
[18]	No label (Query for alarm interval measurement data)	AIN?
[19]	No label (Query for 1 second data)	OSD?
[20]	No label (Query for intermediate data)	IMD?
[21]	No label (Query for error rate detection status)	BES?

ZOM <numeric>

Parameter	<numeric> = <DECIMAL NUMERIC PROGRAM DATA> 0 Zoom out 1 Zoom in
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?

Response	<numeric> = <NR1 NUMERIC RESPONSE DATA> (FIX1)
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.

END? <result>,<item>

Parameter

<result> = <DECIMAL NUMERIC PROGRAM DATA>

0	Time data
1	Alarm measurement data
2	Error measurement data
3	Threshold EI,EFI data
4	Error performance

<item> = <DECIMAL NUMERIC PROGRAM DATA>

0	Outputs all items
1 to 13	Outputs data of 1 item

For the display result items corresponding to each <result> see Table 8.4.1.2-2.

Response

<string> = <STRING RESPONSE DATA>

Table 8.4.1.2-2 Response Format

Form	Format	Description
Form1 Integer type	XXXXXXX	For 0 to 9999999
	X.XXXXEXX	For 1.0000E07 to 9.9999E17
	ERR	When no data corresponds to the query
Form2 Fraction type	X.XXXXE-XX	For 0.0000E–18 to 1.0000E00
	ERR	When no data corresponds to the query
Form3 % type	XXX.XXXXX	For 0.0000 to 100.0000
	ERR	When no data corresponds to the query
Form4 Frequency type	XXXXXXXXX	For 0 to MAX (Hz)
	ERR	When no data corresponds to the query
Form5 Time type	XX-XX-XX XX:XX:XX	For year-month-day hour:minute:second
	X XX:XX:XX	For day hour:minute:second
	ERR	When no data corresponds to the query
Form6 Mixed type	XXXXXXXX,XXX.XXXX	For less than 1E+8
	X.XXXXEXX,XXX.XXXX	For 1E+8 and above
	ERR	When no data corresponds to the query

Function

Queries the measurement data corresponding to the parameter.

Table 8.4.1.2-3 Parameter

<result>	<item>	Items	Format
0 Time data	1	Measurement start time	Form5
	2	Measurement end time	Form5
	3	Measurement elapsed time	Form5
	4	Measurement remaining time	Form5
1 Alarm measurement data	2	Clock Loss interval counts	Form1
	3	Sync Loss interval counts	Form1
	4	CR Unlock interval counts	Form1
2 Error measurement data	1	Error Rate (total)	Form2
	2	Error Count (total)	Form1
	3	Clock Count	Form1
	4	EI	Form1
	5	%EFI	Form3
	6	Error Rate (INS)	Form2
	7	Error Rate (OMI)	Form2
	8	Error Count (INS)	Form1
	9	Error Count (OMI)	Form1
	10	Error Rate (Transition)	Form2
	11	Error Rate (Non Transition)	Form2
	12	Error Count (Transition)	Form1
	13	Error Count (Non Transition)	Form1
3 Threshold EI,EFI data	1	>1.0E–3	Form6
	2	>1.0E–4	Form6
	3	>1.0E–5	Form6
	4	>1.0E–6	Form6
	5	>1.0E–7	Form6
	6	>1.0E–8	Form6
	7	≤1.0E–8	Form6
4 Error performance data	1	ES	Form1
	2	EFS	Form1
	3	SES	Form1
	4	DM	Form1
	5	US	Form1
	6	%ES	Form3
	7	%EFS	Form3
	8	%SES	Form3
	9	%DM	Form3
	10	%US	Form3

Example

```
> END? 0,1
< 05-11-30 18:00:00
```

Compatibility

Partially compatible with the MP1762D.

MTR? <item>

Parameter	<item> = <DECIMAL NUMERIC PROGRAM DATA>
	0 Bit Error (Total Error)
	3 Clock Loss or CR Unlock
	4 Pattern Sync Loss
Response	<numeric> = <NR1 NUMERIC RESPONSE DATA> (FIX1)
	0 When an alarm occurs
	1 When no alarm occurs
	2 When no data corresponds to the query
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?

Response	<numeric> = <NR3 NUMERIC RESPONSE DATA>
	0.0000E–18 to 1.0000E–00 Error rate
Function	Outputs the error rate measurement result according to the output format. "0.0000E–00" is returned if no data exists.

Example	`> ER?`
	`< ER 0.0000E-00`
	`< ER 1.9999E-1`

Compatibility	Compatible with the MP1632C and MP1762D.

EC?

Response	<numeric> = <NR1 or NR3 NUMERIC RESPONSE DATA>
	1.0000E07 to 9.9999E17 Error count
	0 to 9999999
Function	Outputs the error count measurement results according to the output format. "1.0000E–99" is returned if no data exists.

Example	`> EC?`
	`< EC 1.0000E-99`
	`< EC 1.0000E09`
	`< EC 9999999`

Compatibility	Compatible with the MP1632C and MP1762D.

EFI?

Response
<numeric> = <NR2 NUMERIC RESPONSE DATA>
0.0000 to 100.0000 Error free interval rate

Function
Outputs the measurement result of the error free interval rate according to the output format.
"999.9999" is returned if no data exists.

Example
```
> EFI?
< EFI   999.9999
< EFI    20.0088
< EFI   100.0000
```

Compatibility
Compatible with the MP1632C and MP1762D.

EI?

Response
<numeric> = <NR1 or NR3 NUMERIC RESPONSE DATA>
1.0000E07 to 9.9999E17 Error interval
0 to 9999999

Function
Outputs the measurement result of the error interval count according to the output format.
"1.0000E–99" is returned if no data exists.

Example
```
> EI?
< EI  1.0000E-99
< EI  1.0000E09
< EI    1999984
```

Compatibility
Compatible with the MP1632C and MP1762D.

FRQ?

Response	<numeric> = <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	> FRQ? < FRQ 0.000 < FRQ 12500.000
Compatibility	Compatible with the MP1632C and MP1762D.

CC?

Response	<numeric> = <NR1 or NR3 NUMERIC RESPONSE DATA> 1.0000E07 to 9.9999E17 Clock count 0 to 9999999
Function	Outputs the measurement results of the clock count according to the output format. "1.0000E–99" is returned if no data exists.
Example	> CC? < CC 1.0000E-99 < CC 1.0000E-09 < CC 5000256
Compatibility	Compatible with the MP1632C and MP1762D.

CLI?

Response	<numeric> = <NR1 NUMERIC RESPONSE DATA> (FIX1) 0 Non Clock Loss status 1 Clock Loss status
Function	Displays the clock input status.
Example	> CLI? < 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,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.

PFC?

Response	<numeric> = <NR1 or NR3 NUMERIC RESPONSE DATA> (FIX10) XXXXXXXXXX ES data <numeric> = <NR1 or NR3 NUMERIC RESPONSE DATA> (FIX10) XXXXXXXXXX EFS data <numeric> = <NR1 or NR3 NUMERIC RESPONSE DATA> (FIX10) XXXXXXXXXX SFS data <numeric> = <NR1 or NR3 NUMERIC RESPONSE DATA> (FIX10) XXXXXXXXX US data <numeric> = <NR1 or NR3 NUMERIC RESPONSE DATA> (FIX10) XXXXXXXXXX DM data <numeric> = <NR1 or NR3 NUMERIC RESPONSE DATA> (FIX10) XXXXXXXXXX EC data Displays the above items delimited with a comma (,) (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	`> PFC?`
	`< PFC`
	`1.0000E-99,1.0000E-99,1.0000E-99,1.0000E-99,1.0000E-99,`
	`1.0000E-99`
	`< PFC 1800, 1800, 900, 59,`
	`100,1.0000E-99`

Compatibility	Compatible with the MP1632C.

THE?

Response	<numeric> = <NR1 or NR3 NUMERIC RESPONSE DATA> (FIX10) XXXXXXXXXX >1E–03 data <numeric> = <NR1 or NR3 NUMERIC RESPONSE DATA> (FIX10) XXXXXXXXXX >1E–04 data <numeric> = <NR1 or NR3 NUMERIC RESPONSE DATA> (FIX10) XXXXXXXXXX >1E–05 data <numeric> = <NR1 or NR3 NUMERIC RESPONSE DATA> (FIX10) XXXXXXXXXX >1E–06 data <numeric> = <NR1 or NR3 NUMERIC RESPONSE DATA> (FIX10) XXXXXXXXXX >1E–07 data <numeric> = <NR1 or NR3 NUMERIC RESPONSE DATA> (FIX10) XXXXXXXXXX >1E–08 data <numeric> = <NR1 or NR3 NUMERIC RESPONSE DATA> (FIX10) XXXXXXXXXX ≤1E–08 data Displays the above items delimited with a comma (,) (THE XX,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?

Response	<numeric> = <NR2 or NR3 NUMERIC RESPONSE DATA> (FIX10) XXX.XXXX >1E–03 data <numeric> = <NR2 or NR3 NUMERIC RESPONSE DATA> (FIX10) XXX.XXXX >1E–04 data <numeric> = <NR2 or NR3 NUMERIC RESPONSE DATA> (FIX10) XXX.XXXX >1E–05 data <numeric> = <NR2 or NR3 NUMERIC RESPONSE DATA> (FIX10) XXX.XXXX >1E–06 data <numeric> = <NR2 or NR3 NUMERIC RESPONSE DATA> (FIX10) XXX.XXXX >1E–07 data <numeric> = <NR2 or NR3 NUMERIC RESPONSE DATA> (FIX10) XXX.XXXX >1E–08 data <numeric> = <NR2 or NR3 NUMERIC RESPONSE DATA> (FIX10) XXX.XXXX ≤1E–08 data Displays the above items delimited with a comma (,) (THF XX,XX,XX,XX,XX,XX,XX). "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?

Response	<numeric> = <NR1 or NR3 NUMERIC RESPONSE DATA> (FIX10) XXXXXXXXXX Pattern Sync Loss data <numeric> = <NR1 or NR3 NUMERIC RESPONSE DATA> (FIX10) XXXXXXXXXX Clock Loss data <numeric> = <NR1 or NR3 NUMERIC RESPONSE DATA> (FIX10) XXXXXXXXXX CR Unlock data 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?

Response	<numeric> = <NR1 or NR3 NUMERIC RESPONSE DATA> Average error rate for 1 second, average error count for 1 second X.XXXXE–XX, XXXXXXXX (average error count for 1 second is less than 1E+8) X.XXXXE–XX,X.XXXXEXX (average error count for 1 second is 1E+8 and above)
Function	Outputs the measurement results for 1 second according to the output format. "0.0000E–00,1.0000E–99" is returned if no data exists.

Example	To output the measurement results for 1 second according to the output format:
	`> OSD?`
	`< OSD 0.0000E-00,1.0000E-99`
	`< OSD 2.0000E-01,1.0000E09`

Compatibility	Compatible with the MP1762D.

IMD? <result>,<item>

Parameter	<result> = <DECIMAL NUMERIC PROGRAM DATA> Same as END?
Response	<string> = <STRING RESPONSE DATA> Same as END?
Function	Displays the measurement intermediate data.
Example	`> IMD? 0,1` `< 05-11-30 18:00:00`
Compatibility	Compatible with the MP1762D.

BES? <error>

Parameter	<error> = < DECIMAL NUMERIC PROGRAM DATA>	
	0	Total Error
	1	Insertion Error
	2	Omission Error
	3	Transition Error
	4	Non-Transition Error
Response	<numeric> = <NR1 NUMERIC RESPONSE DATA> (FIX1)	
	0	Error detection
	1	No error detection
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

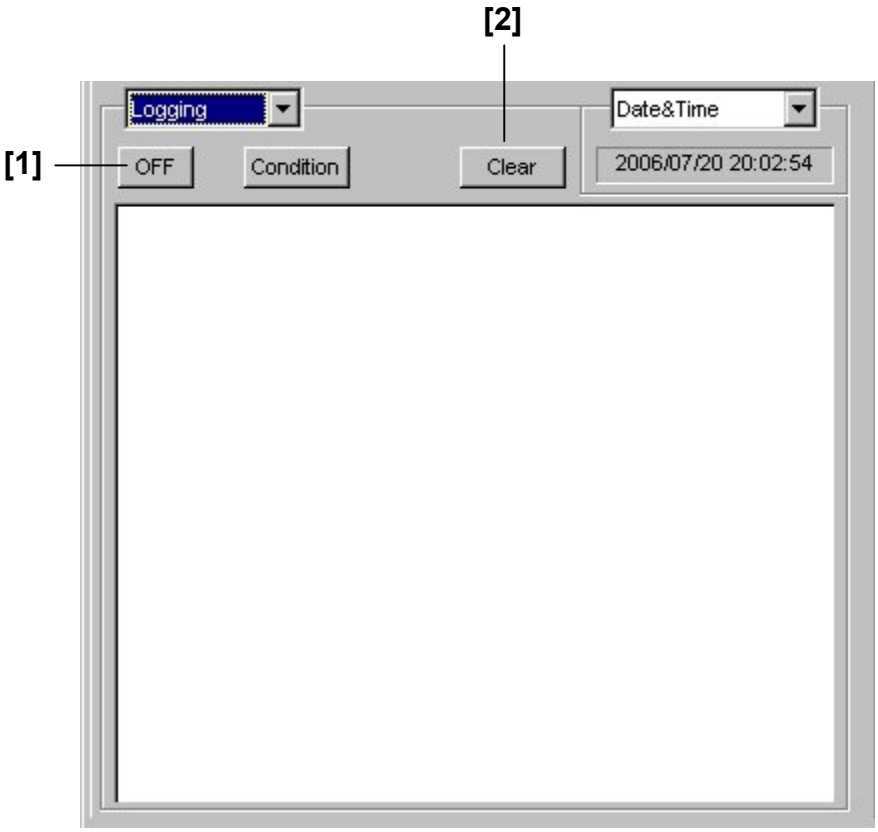


Figure 8.4.1.3-1 Logging setting

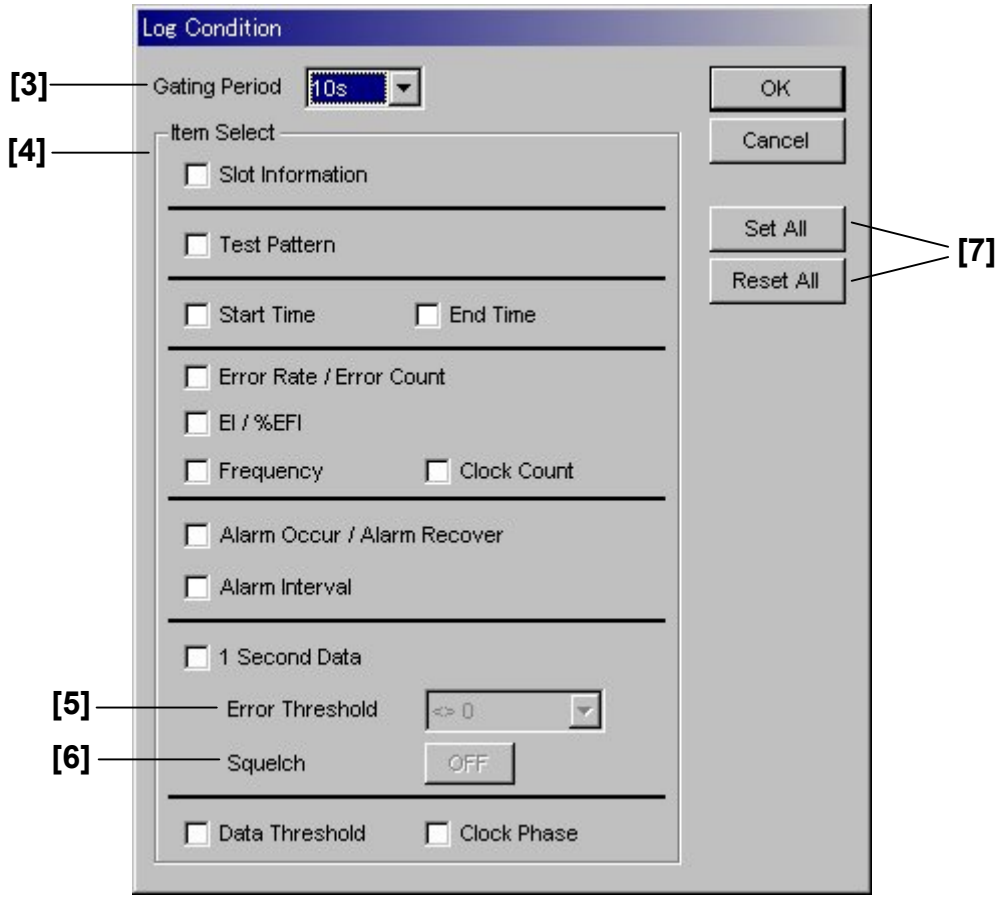


Figure 8.4.1.3-2 Logging Condition

Table 8.4.1.3-1 Logging setting commands

No.	Setting Items	Commands
[1]	Logging ON/OFF	LON
		LON?
[2]	Clear	LCR
[3]	Gating Period	LCP
		LCP?
[4]	Item Select	LCD
		LCD?
[5]	Error Threshold	LCT
		LCT?
[6]	Squelch	LCS
		LCS?
[7]	Set All/Reset All	LAS

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>

Parameter	<item> = <DECIMAL NUMERIC PROGRAM DATA>
	0 to 12 Each logging item
Response	<numeric> = <NR1 NUMERIC RESPONSE DATA> (FIX1)
	0 OFF
	1 ON
Function	Sets the specified logging measurement item ON/OFF.
Example	`> LCD? 0`
	`< LCD 1`
Compatibility	Incompatible with existing models.

LCT <thre>

Parameter	<thre> = <DECIMAL NUMERIC PROGRAM DATA>
	0 <>0
	1 >1E–3
	2 >1E–4
	3 >1E–5
	4 >1E–6
	5 >1E–7
	6 >1E–8
	7 >1E–9
Function	Sets the logging error rate threshold value.
Example	To set the logging error rate threshold value to 0:
	`> LCT 0`
Compatibility	Incompatible with existing models.

LCT?

Response	<thre> = <NR1 NUMERIC RESPONSE DATA> (FIX1)
Function	Queries the logging error rate threshold value.
Example	`> LCT?`
	`< LCT 0`
Compatibility	Incompatible with existing models.

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

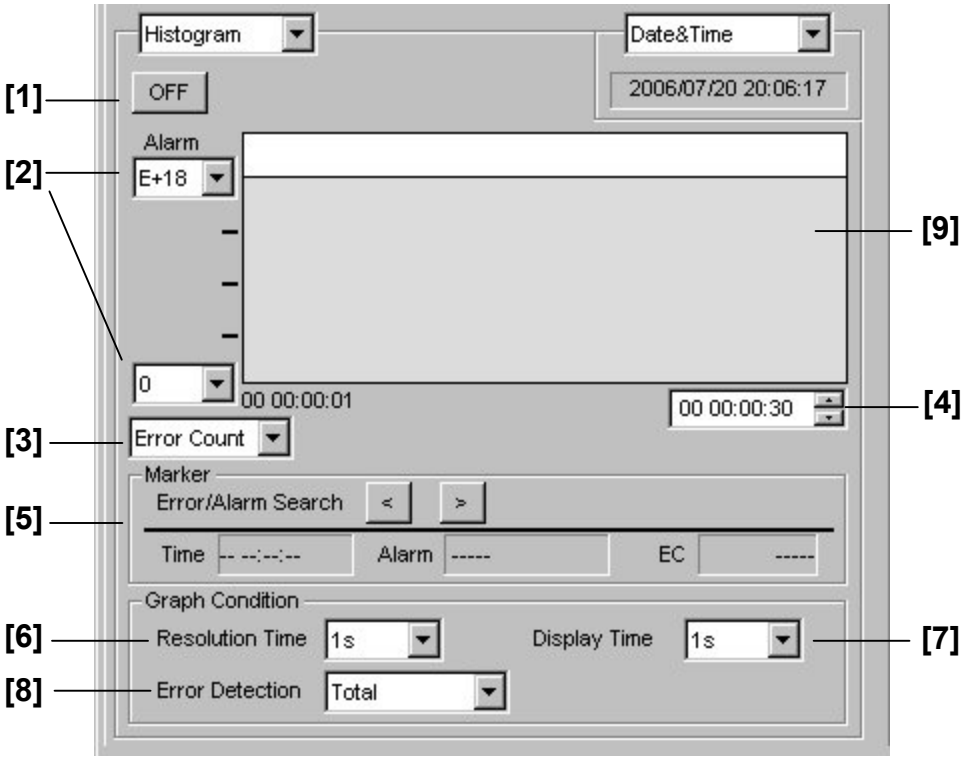


Figure 8.4.1.4-1 Histogram setting

Table 8.4.1.4-1 Histogram setting commands

No.	Setting Items	Commands
[1]	Histogram ON/OFF	HON
		HON?
[2]	Scale setting	HAL
		HAL?
[3]	Measurement item setting	HTP
		HTP?
[4]	Display time setting	HTS
		HTS?
[5]	Marker	HMK
		HMK?
[6]	Resolution Time	HRS
		HRS?
[7]	Display Time	HDT
		HDT?
[8]	Error Detect	HED
		HED?
[9]	Query for measurement result	HIS?

HON <numeric>

Parameter	<numeric> = <DECIMAL NUMERIC PROGRAM DATA>
	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.

HAL?

Response	<top> = <NR1 NUMERIC RESPONSE DATA> (FIX2) <bottom> = <NR1 NUMERIC RESPONSE DATA> (FIX2)
Function	Queries the vertical axis scale of the histogram.

Example	`> HAL?`
	`< HAL  5, 0`
	`< HAL 19,17`

Compatibility	Incompatible with existing models.

HTP <type>

Parameter	<type> = <DECIMAL NUMERIC PROGRAM DATA>
	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): `> HTP 0`
Compatibility	Incompatible with existing models.

HTP?

Response	<type> = <NR1 NUMERIC RESPONSE DATA> (FIX1)
Function	Queries the measurement item counting type displayed in the histogram.
Example	`> HTP?` `< 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 The minimum setting value is 0, 0, 0, 30.
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.

HMK <position>

Parameter
<position> = <DECIMAL NUMERIC PROGRAM DATA>

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:

```
> HMK 1
```

Compatibility
Incompatible with existing models.

HMK?

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

"--" is displayed if there is no alarm and error.
<alarm> = <NR1 NUMERIC RESPONSE DATA> (FIX1)

0	Clock Loss
1	Sync Loss
2	CR Unlock
3	No alarm

<result> = <STRING RESPONSE DATA>
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

Form	Format	Description
Form1 Integer type	XXXXXXX	For 0 to 9999999
	X.XXXXEXX	For 1.0000E07 to 9.9999E17
	1.0000E-99	When no data corresponds to the query
Form2 Fraction type	X.XXXXE-XX	For 0.0001E–18 to 1.0000E00
	1.0000E-99	When no data corresponds to the query

Function
Queries the marker setting time, alarm, and error rate.

Example

```
> HMK?
< HMK 00,01,30,30,3,1.0000E07
```

Compatibility
Incompatible with existing models.

HRS <numeric>

Parameter	<numeric> = <DECIMAL NUMERIC PROGRAM DATA>	
	0	1 s
	1	10 s
	2	30 s
	3	1 min
	4	10 min
	5	30 min
	6	1 hour
Function	Sets the histogram calculation resolution.	
Example	To set the histogram calculation resolution to 1 second:	
	`> HRS 0`	
Compatibility	Incompatible with existing models.	

HRS?

Response	<numeric> = <NR1 NUMERIC RESPONSE DATA> (FIX1)
Function	Queries the histogram calculation resolution.
Example	`> HRS?`
	`< HRS 0`
Compatibility	Incompatible with existing models.

HDT <numeric>

Parameter	<numeric> = <DECIMAL NUMERIC PROGRAM DATA>	
	0	1 s
	1	10 s
	2	30 s
	3	1 min
	4	10 min
	5	30 min
	6	1 hour
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	<numeric> = <NR1 NUMERIC RESPONSE DATA> (FIX1)
Function	Queries the histogram display resolution.
Example	> HDT? < HDT 0
Compatibility	Incompatible with existing models.

HED <mode>

Parameter	<mode> = <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: > HED 0
Compatibility	Incompatible with existing models.

HED?

Response	<mode> = <NR1 NUMERIC RESPONSE DATA> (FIX1)
Function	Queries the bit error alarm measurement method and error detection method for histogram calculation.
Example	> HED? < HED 0
Compatibility	Incompatible with existing models.

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.

<rate> = <STRING RESPONSE DATA>

When the measurement item is Error Rate, the response is returned in Form2.

Table 8.4.1.4-3 Response format

Form	Format	Description
Form1 Integer type	XXXXXXX	For 0 to 9999999
	X.XXXXEXX	For 1.0000E07 to 9.9999E17
	1.0000E-99	When no data corresponds to the query
Form2 Fraction type	X.XXXXE-XX	For 0.0001E–18 to 1.0000E00
	1.0000E-99	When no data corresponds to the query

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

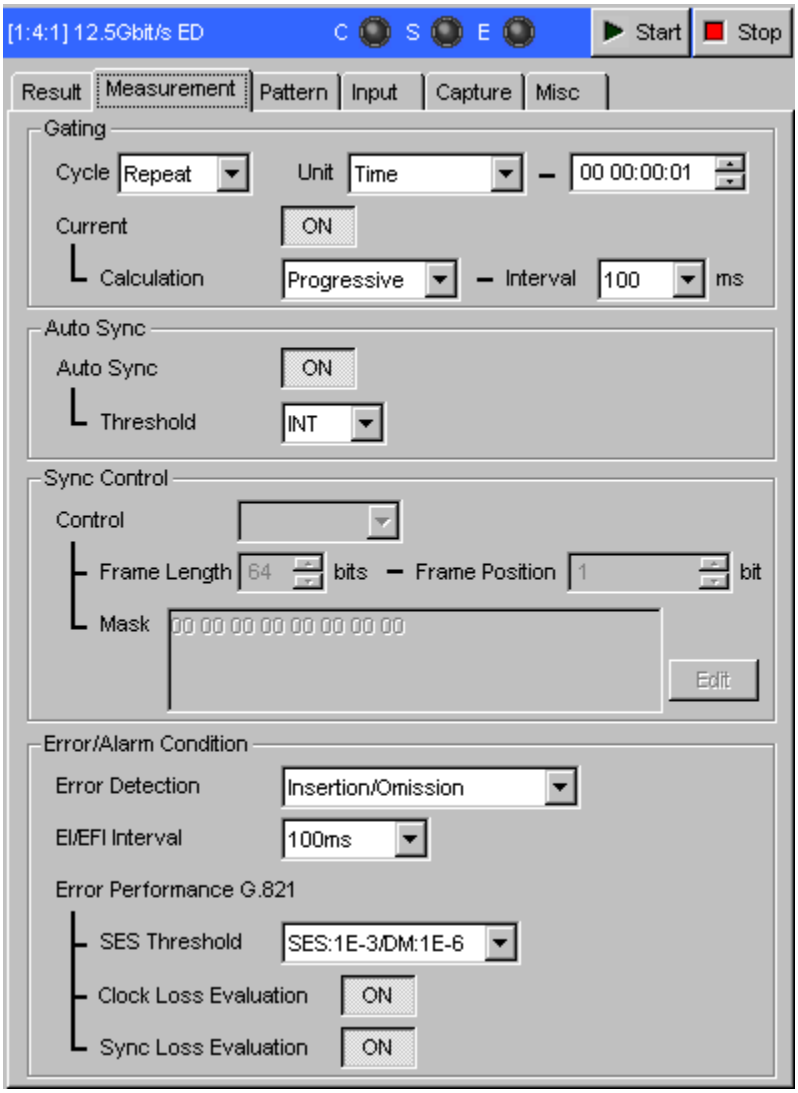


Figure 8.4.2-1 Measurement tab

8.4.2.1 Gating setting commands

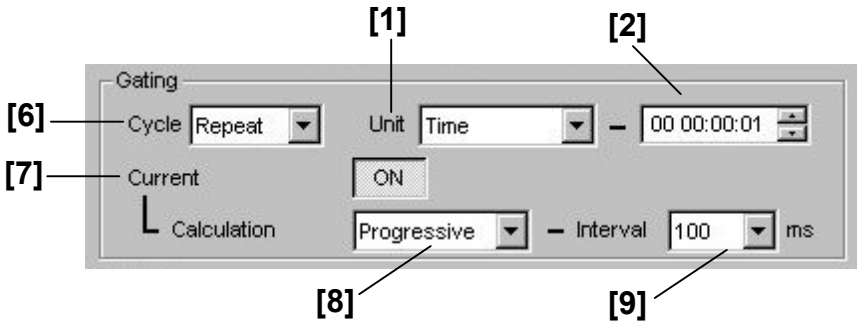


Figure 8.4.2.1-1 Gating setting (Time)

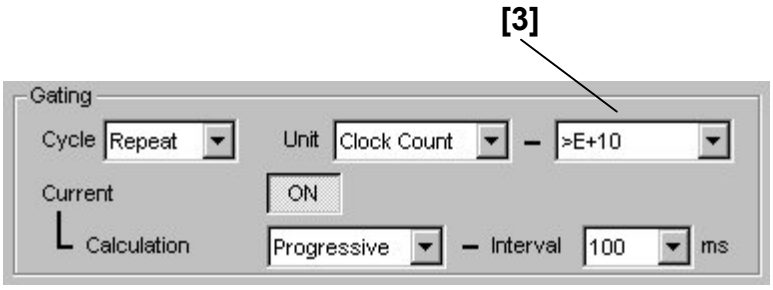


Figure 8.4.2.1-2 Gating setting (Clock Count)

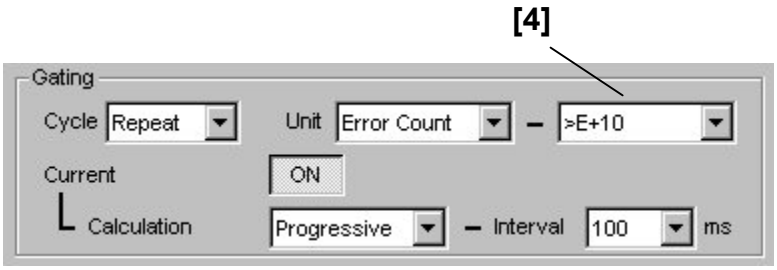


Figure 8.4.2.1-3 Gating setting (Error Count)

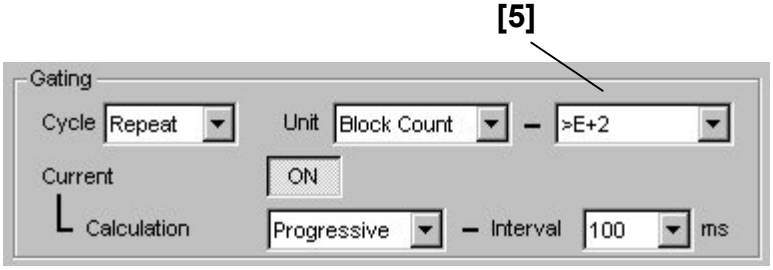


Figure 8.4.2.1-4 Gating setting (Block Count)

Table 8.4.2.1-1 Measurement tab setting commands

No.	Setting Items	Commands
[1]	Unit	UNT
		UNT?
[2]	No label (Measurement time setting)	PRD
		PRD?
[3]	No label (Clock count setting)	CLC
		CLC?
[4]	No label (Error count setting)	ERC
		ERC?
[5]	No label (Block count setting)	BLC
		BLC?
[6]	Cycle	MOD
		MOD?
[7]	Current	CUR
		CUR?
[8]	Calculation	CAL
		CAL?
[9]	Interval	ITV
		ITV?

UNI <unit>

Parameter	<unit> = <DECIMAL NUMERIC PROGRAM DATA>	
	0	Time
	1	Clock Count
	2	Error Count
	3	Block Count
Function	Sets the measurement period unit.	
Example	To set the measurement period unit to Time:	
	`> UNI 0`	
Compatibility	Incompatible with existing models.	

UNI?

Response	<unit> = <NR1 NUMERIC RESPONSE DATA> (FIX1)
Function	Queries the measurement period unit.
Example	`> UNI?`
	`< UNI 0`
Compatibility	Incompatible with existing models.

PRD <day>,<hour>,<min>,<second>

Parameter	<day>,<hour>,<min>,<second> = <DECIMAL NUMERIC PROGRAM DATA>		
	<day>	0 to 99	0 to 99 days, in 1-day steps
	<hour>	0 to 23	0 to 23 hours, in 1-hour steps
	<min>	0 to 59	0 to 59 minutes, in 1-minute steps
	<second>	0 to 59	0 to 59 seconds, in 1-second steps
	(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	<day>,<hour>,<min>,<second> = <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	> PRD?		
	< PRD 00,01,00,00		
Compatibility	Compatible with the MP1632C and MP1762D.		

CLC<count>

Parameter	<count> = <DECIMAL NUMERIC PROGRAM DATA>	
	0	>1E+4
	1	>1E+5
	2	>1E+6
	3	>1E+7
	4	>1E+8
	5	>1E+9
	6	>1E+10
	7	>1E+11
	8	>1E+12
	9	>1E+13
	10	>1E+14
	11	>1E+15
	12	>1E+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 1E+4:	
	> CLC 0	
Compatibility	Incompatible with existing models.	

CLC?

Response	<count> = <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	<count> = <DECIMAL NUMERIC PROGRAM DATA>

0	>1E+4
1	>1E+5
2	>1E+6
3	>1E+7
4	>1E+8
5	>1E+9
6	>1E+10
7	>1E+11
8	>1E+12
9	>1E+13
10	>1E+14
11	>1E+15
12	>1E+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 1E+4: `> ERC 0`
Compatibility	Incompatible with existing models.

ERC?

Response	<count> = <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	<count> = <DECIMAL NUMERIC PROGRAM DATA>	
	0	>1E+2
	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
	11	>1E+13
	12	>1E+14
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: `> BLC 0`	
Compatibility	Incompatible with existing models.	

BLC?

Response	<count> = <NR1 NUMERIC RESPONSE DATA> (FIX2)
Function	Queries the block count threshold when the measurement period is set to Block Count.

Example	`> BLC?`
	`< BLC 0`
	`< BLC 12`

Compatibility	Incompatible with existing models.

MOD <mode>

Parameter	<mode> = <DECIMAL NUMERIC PROGRAM DATA>	
	0	Repeat
	1	Single
	2	Untimed
Function	Sets the bit error alarm measurement processing mode.	
Example	To set the bit error alarm measurement processing mode to Repeat (performing measurement repeatedly): `> MOD 0`	
Compatibility	Compatible with the MP1632C and MP1762D.	

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

Table 8.4.2.2-1 Auto Sync setting commands

No.	Setting Items	Commands
[1]	Auto Sync	SYN
		SYN?
[2]	Threshold Gain	SYE
		SYE?

SYN <numeric>

Parameter	<numeric> = <DECIMAL NUMERIC PROGRAM DATA>
	0 Auto Sync OFF
	1 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): `> SYN 1`
Compatibility	Compatible with the MP1632C and MP1762D.

SYN?

Response	<numeric> = <NR1 NUMERIC RESPONSE DATA> (FIX1)
Function	Queries the Auto Sync ON/OFF state.
Example	`> SYN?` `< SYN 1`
Compatibility	Compatible with the MP1632C and MP1762D.

SYE <thre>

Parameter	<thre> = <DECIMAL NUMERIC PROGRAM DATA>
	0 E–2
	1 E–3
	2 E–4
	3 E–5
	4 E–6
	5 E–7
	6 E–8
	8 Internal
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?

Response	<thre> = <NR1 NUMERIC RESPONSE DATA> (FIX1)
Function	Queries the synchronization detection threshold for resynchronization.
Example	`> SYE?` `< SYE 0`
Compatibility	Compatible with the MP1762D.

8.4.2.3 Sync Control setting commands

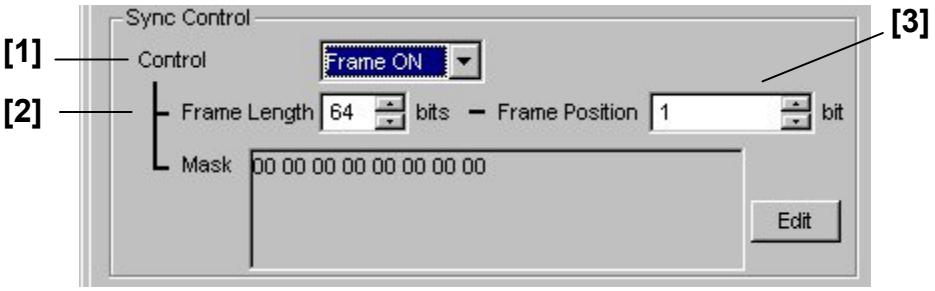


Figure 8.4.2.3-1 Sync Control setting commands

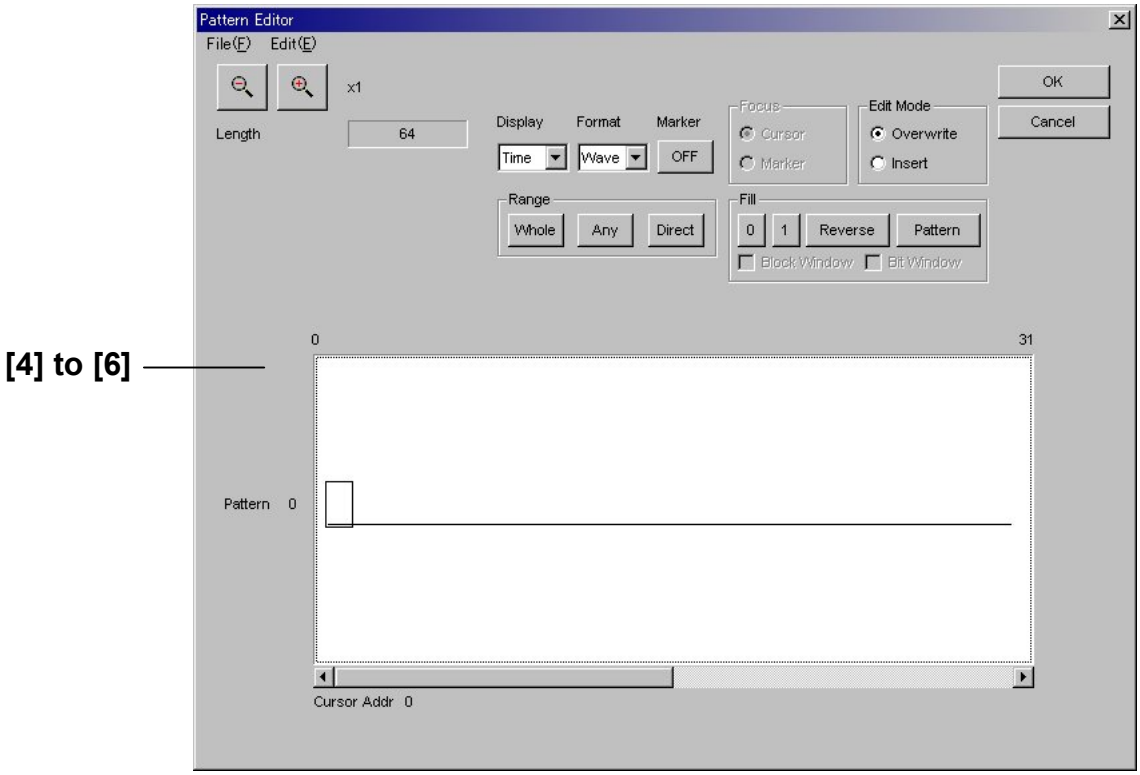


Figure 8.4.2.3-2 Sync Control setting (Pattern Editor screen)

Table 8.4.2.3-1 Sync Control setting commands

No.	Setting Items	Commands
[1]	Control	SYM
		SYM?
[2]	Frame Length	FLN
		FLN?
[3]	Frame Position	FPS
		FPS?
[4]	Mask (Setting of mask pattern)	PDW
		PDW?
[5]	No label (Data inversion)	PSA
[6]	No label (All 0s or 1s)	PFL

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

```
> SYM?
< SYM 0
```

Compatibility Compatible with the MP1762D.

FLN <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 detection ON.
Example	To set the unique pattern length to 4 bits when the synchronization mode is set to frame detection ON: `> FLN 4`
Compatibility	Compatible with the MP1632C and MP1762D.

FLN?

Response	<length> = <NR1 NUMERIC RESPONSE DATA> (FIX3)
Function	Queries the unique pattern length when the synchronization mode is set to frame detection ON.

Example	`> FLN?`
	`< FLN 4`
	`< FLN 64`

Compatibility	Compatible with the MP1632C and MP1762D.

FPS <length>

Parameter	<length> = <DECIMAL NUMERIC PROGRAM DATA> 1 to 134217725 1 to 134,217,725 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: `> FPS 1`
Compatibility	Incompatible with existing models.

FPS?

Response	<length> = <NR1 NUMERIC RESPONSE DATA>
Function	Queries the frame synchronization start position when the synchronization mode is set to 1 (frame detection ON).

Example	`> FPS?`
	`< FPS 1`
	`< FPS 134217665`

Compatibility	Incompatible with existing models.

PDW <data>

Parameter

<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)

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?

Response

<data> = <HEXADECIMAL NUMERIC RESPONSE DATA>

#H**...*	1 to 16 characters (Specify pattern data in hexadecimal format)

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

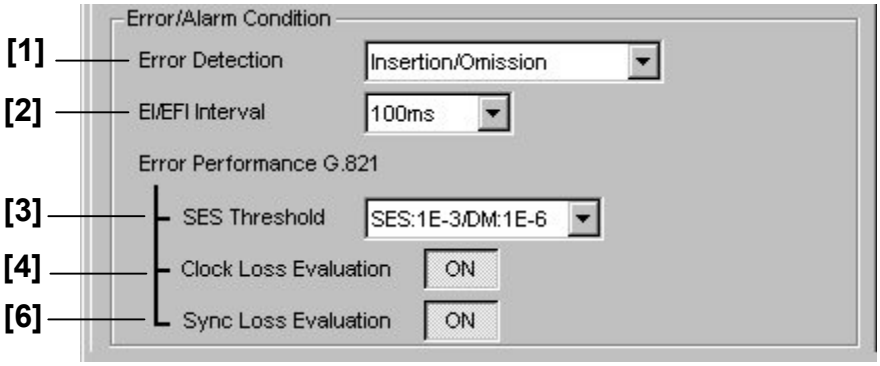


Figure 8.4.2.4-1 Error/Alarm Condition

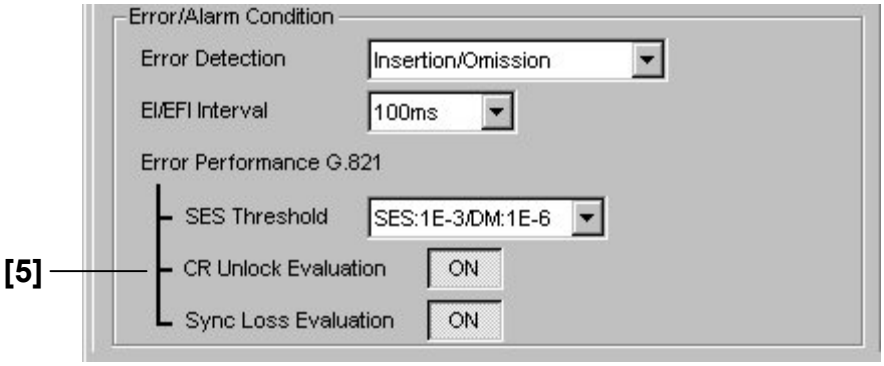


Figure 8.4.2.4-2 Error/Alarm Condition setting (Recovered Clock)

Table 8.4.2.4-1 Error/Alarm Condition setting commands

No.	Setting Items	Commands
[1]	Error Detect	ERD
		ERD?
[2]	EI/EFI Interval	EIT
		EIT?
[3]	SES Threshold	ETH
		ETH?
[4]	Clock Loss Evaluation	CLS
		CLS?
[5]	CR Unlock Evaluation	CRU
		CRU?
[6]	Sync Loss Evaluation	SLS
		SLS?

ERD <mode>

Parameter	<mode> = <DECIMAL NUMERIC PROGRAM DATA>
	0 INS/OMI
	1 Transition/Non Transition
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: `> ERD 0`
Compatibility	Incompatible with existing models.

ERD?

Response	<mode> = <NR1 NUMERIC RESPONSE DATA> (FIX1)
Function	Queries the error detection method during bit error alarm measurement.
Example	`> ERD?` `< ERD 0`
Compatibility	Incompatible with existing models.

EIT <numeric>

Parameter	<numeric> = <DECIMAL NUMERIC PROGRAM DATA>
	0 1 ms
	1 10 ms
	2 100 ms
	3 1 s
Function	Sets the interval for EI and %EFI measurement.
Example	To set the interval for EI and %EFI measurement to 1 ms: `> EIT 0`
Compatibility	Compatible with the MP1762D.

EIT?

Response	<numeric> = <NR1 NUMERIC RESPONSE DATA> (FIX1)
Function	Queries the interval for EI and %EFI measurement.
Example	`> EIT?` `< EIT 0`
Compatibility	Compatible with the MP1762D.

ETH <thre>

Parameter	<thre> = <DECIMAL NUMERIC PROGRAM DATA> 0 SES: 1E–3, DM: 1E–6 1 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: `> ETH 0`
Compatibility	Compatible with the MP1632C and MP1762D.

ETH?

Response	<thre> = <NR1 NUMERIC RESPONSE DATA> (FIX1)
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>

Parameter	<numeric> = <DECIMAL NUMERIC PROGRAM DATA> 0 Out of evaluation target 1 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: `> CLS 1`
Compatibility	Compatible with the MP1632C and MP1762D.

CLS?

Response	<numeric> = <NR1 NUMERIC RESPONSE DATA> (FIX1)
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.

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

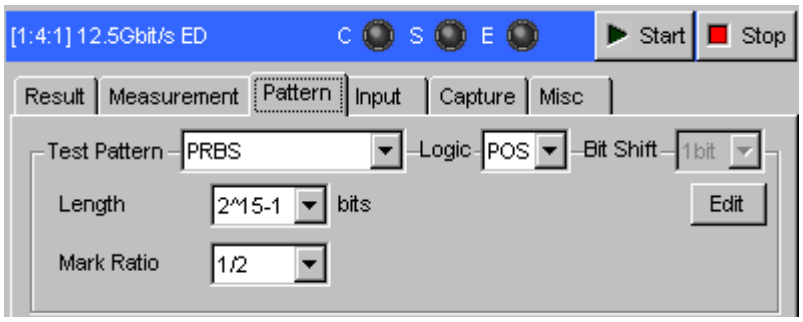


Figure 8.4.3-1 Pattern setting tab

8.4.3.1 Test Pattern setting commands

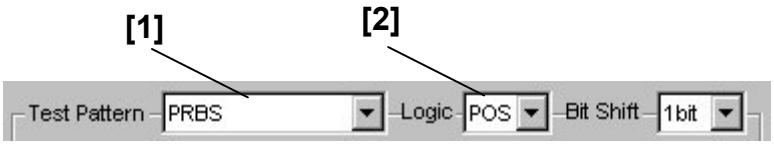


Figure 8.4.3.1-1 Test Pattern setting field

Table 8.4.3.1-1 Test Pattern setting commands

No.	Setting Items	Commands
[1]	Test Pattern	PTS
		PTS?
[2]	Logic	LGC
		LGC?

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: `> PTS 1`
Compatibility	Compatible with the MP1762D.

PTS?

Response	<type> = <NR1 NUMERIC RESPONSE DATA> (FIX1)
Function	Queries the type of the test pattern.
Example	`> 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: `> 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	`> LGC?` `< LGC 0`
Compatibility	Compatible with the MP1632C and MP1762D.

8.4.3.2 PRBS pattern setting commands

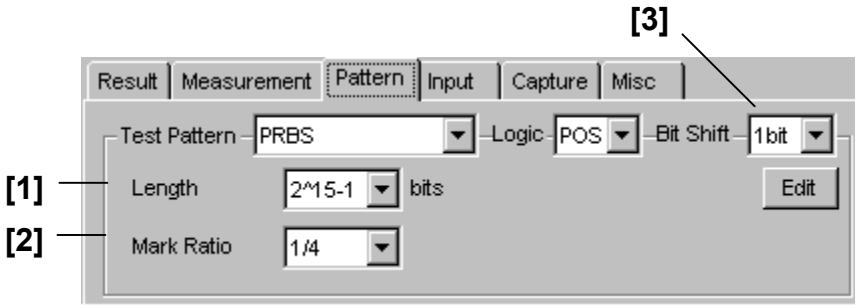


Figure 8.4.3.2-1 PRBS pattern setting

Table 8.4.3.2-1 PRBS pattern setting commands

No.	Setting Items	Commands
[1]	Length	PTN
		PTN?
[2]	Mark Ratio	MRK
		MRK?
[3]	Bit Shift	SFT
		SFT?

PTN <numeric>

Parameter

<numeric> = <DECIMAL NUMERIC PROGRAM DATA>

	PRBS		Zero-Substitution
2	2^7-1	2	2^7
3	2^9-1	3	2^9
4	$2^{10}-1$	4	2^{10}
5	$2^{11}-1$	5	2^{11}
6	$2^{15}-1$	6	2^{15}
7	$2^{20}-1$	7	2^{20}
8	$2^{23}-1$	8	2^{23}
9	$2^{31}-1$	–	
		12	2^7-1
		13	2^9-1
		14	$2^{10}-1$
		15	$2^{11}-1$
		16	$2^{15}-1$
		17	$2^{20}-1$
		18	$2^{23}-1$

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:

```
> 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

```
> 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>

	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 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

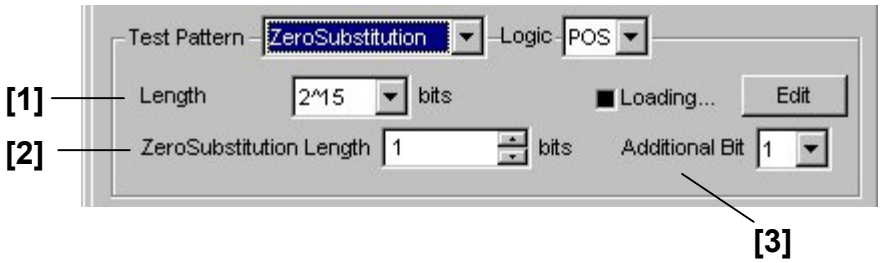


Figure 8.4.3.3-1 Zero-Substitution pattern setting

Table 8.4.3.3-1 Zero-Substitution pattern setting commands

No.	Setting Items	Commands
[1]	Length	PTN Refer to Section 8.4.3.2 "PRBS pattern setting commands" for detail on this command.
		PTN? Refer to Section 8.4.3.2 "PRBS pattern setting commands" for detail on this command.
[2]	Zero Substitution length	ZLN
		ZLN?
[3]	Additional Bit	ZAD
		ZAD?

ZLN <numeric>

Parameter
<numeric> = <DECIMAL NUMERIC PROGRAM DATA>

1 to 2^n–1	1 to 2^n–1, in 1-bit steps (when ZSUBlength, 2n is set)
1 to 2^n–2	1 to 2^n–2, in 1-bit steps (when ZSUBlength, 2^n–1 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.

ZLN?

Response	<numeric> = <NR1 NUMERIC RESPONSE DATA> (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	<numeric> = <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	<numeric> = <NR1 NUMERIC RESPONSE DATA> (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

Figure 8.4.3.4-1 Data pattern setting

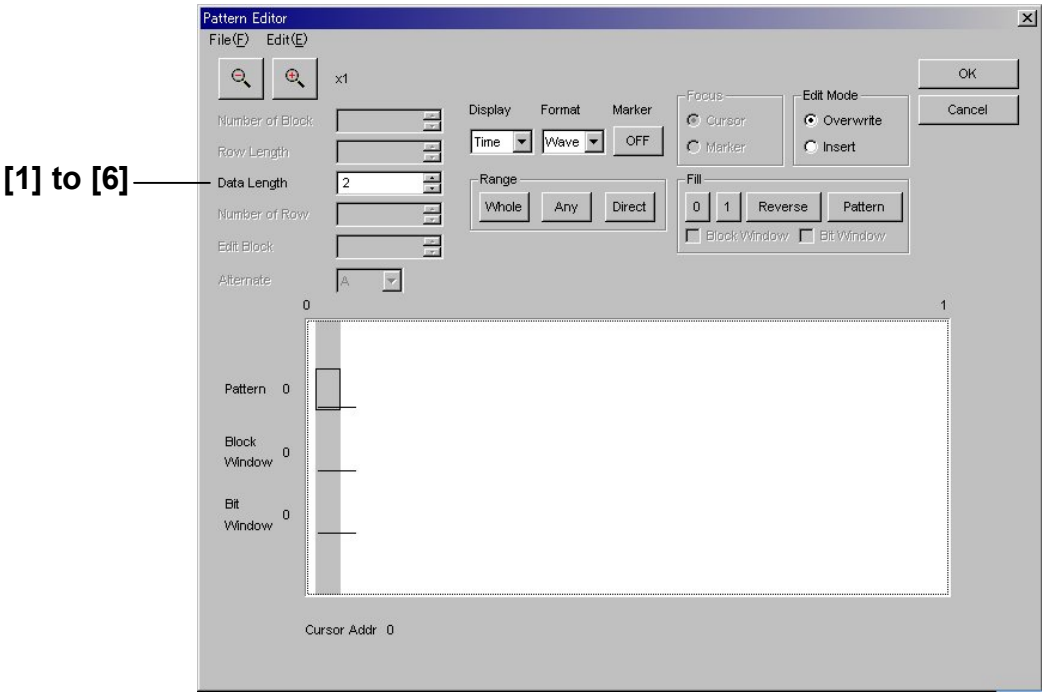


Figure 8.4.3.4-2 Data pattern setting (Pattern Editor)

Table 8.4.3.4-1 Data pattern setting commands

No.	Setting Items	commands
[1]	Data Length	DLN
		DLN?
[2]	No label (Pattern setting)	PDT
		PDT?
[3]	No label (Pattern setting)	BDT
		BDT?
[4]	No label (Data inversion)	DRA
[5]	No label (Data inversion)	DRD
[6]	No label (All 0s or 1s)	ALL
[7]	No label (Pattern setting)	WRT
[8]	No label (Pattern reading)	RED?

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.

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 × 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 × 4 bits)

Function

Queries 400 × 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>

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.

<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.

During 4ch Combination: 1 to 67,108,864 bytes
<binary>: Binary data up to maximum 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 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

<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 is doubled.
At 4 ch Combination, the setting range is quadrupled.

Response

<bdata> = <DEFINITE LENGTH ARBITRARY BLOCK RESPONSE DATA>

Queries binary data of up to the setting pattern when [<size>] is omitted.

Function

Queries binary data of bytes specified by <size> from <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:

```
> BDT? #H0,2
< BDT #12AA
```

To query the setting pattern of binary data from 0 address of the pattern data during Data pattern generation:

```
> BDT? #H0
< BDT #516000AA...
```

(Binary data of 16,777,216 bytes)

Compatibility

Compatible with the MP1632C.

DRA <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 reception.
Specify the inversion range by the <start> and <end> addresses.

Example

To invert 0 to 4F addresses of the pattern data:

```
> DRA #H0,#H4F
```

Compatibility

Compatible with the MP1632C.

DRD <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 128 bits of data from 0 address of the pattern data: `> DRD #H0,128`
Compatibility	Compatible with the MP1632C.

ALL <data>

Parameter	<data> = <DECIMAL NUMERIC PROGRAM DATA> 0 Fills all bits with 0s. 1 Fills all bits with 1s.
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.

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

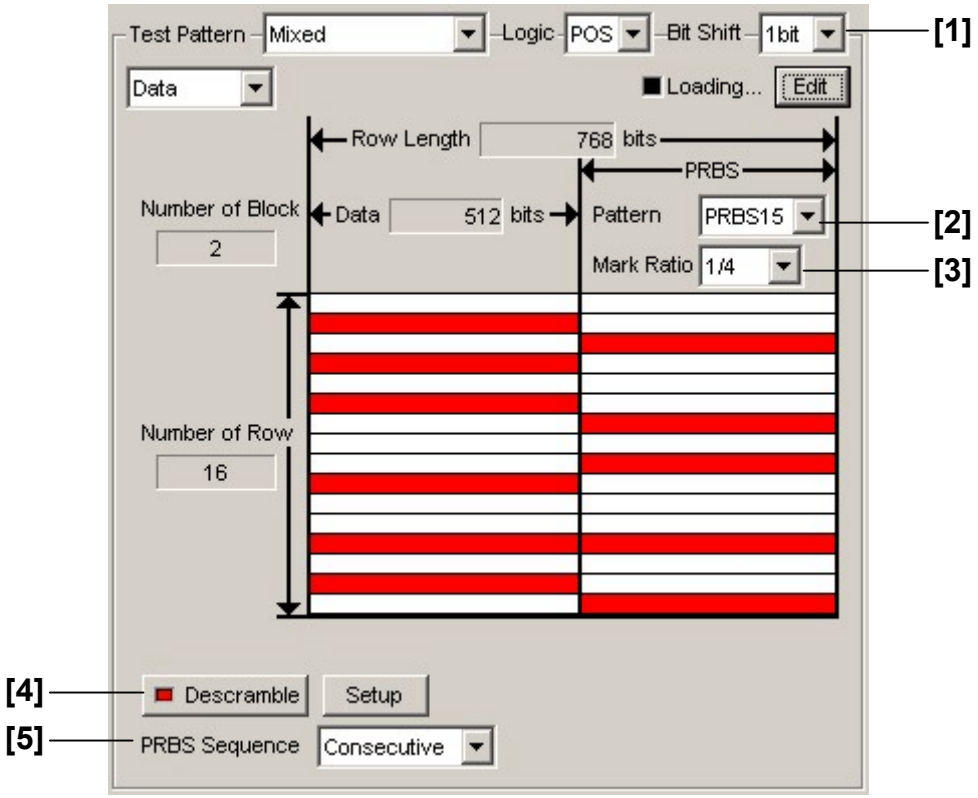


Figure 8.4.3.5-1 Mixed-Data pattern setting

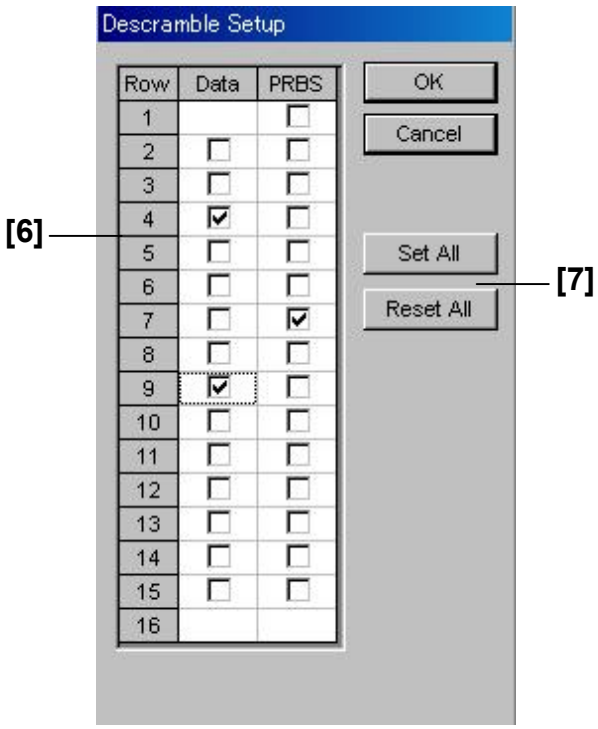


Figure 8.4.3.5-2 Descramble Setup

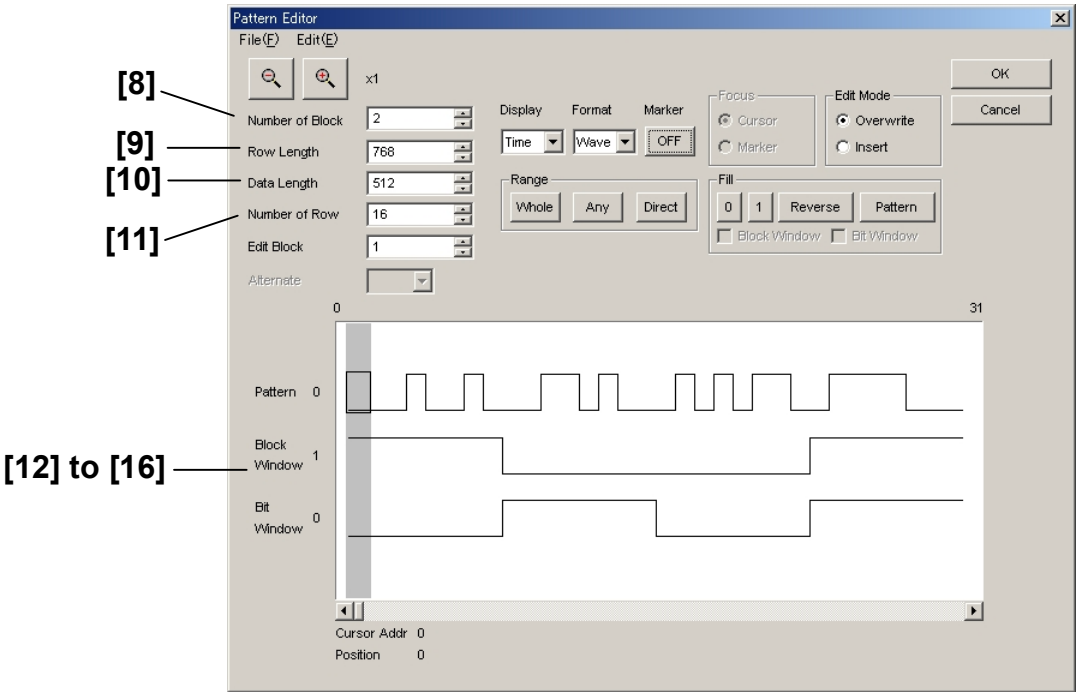


Figure 8.4.3.5-3 Mixed-Data pattern setting (Pattern Editor)

Table 8.4.3.5-1 Mixed-Data pattern setting commands

No.	Setting Items	Commands
[1]	Bit Shift	MPB
		MPB?
[2]	PRBS Pattern	MPP
		MPP?
[3]	PRBS Marl Ratio	MPM
		MPM?
[4]	Descramble ON/OFF	MSC
		MSC?
[5]	PRBS Sequence	MPS
		MPS?
[6]	Setup	MSS
		MSS?
[7]	Set All/Reset All	MSL
[8]	Number of Block	MBL
		MBL?
[9]	Row Length	MRL
		MRL?
[10]	Data Length	MDL
		MDL?
[11]	Number of Row	MRW
		MRW?
[12]	No label (Pattern setting)	MDW
		MDW?
[13]	No label (Pattern setting)	MBW
		MBW?
[14]	No label (Data inversion)	MDA
[15]	No label (Data inversion)	MDD
[16]	No label (All 1s or 0s)	MDF

MPB <numeric>

Parameter	<numeric> = <DECIMAL NUMERIC PROGRAM DATA>
	0 1-bit
	1 3-bit
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?

Response	<numeric> = <NR1 NUMERIC RESPONSE DATA> (FIX1)
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>

Parameter	<numeric> = <DECIMAL NUMERIC PROGRAM DATA>
	2 2^7-1
	3 2^9-1
	4 $2^{10}-1$
	5 $2^{11}-1$
	6 $2^{15}-1$
	7 $2^{20}-1$
	8 $2^{23}-1$
	9 $2^{31}-1$
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.

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.

MSC <numeric>

Parameter	<numeric> = <DECIMAL NUMERIC PROGRAM DATA>
	0 Descramble OFF
	1 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: `> MSC 1`
Compatibility	Incompatible with existing models.

MSC?

Response	<numeric> = <NR1 NUMERIC RESPONSE DATA> (FIX1)
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>

Parameter	<sequence> = <DECIMAL NUMERIC PROGRAM DATA>
	0 Restart: Inconsecutive PRBS pattern
	1 Consecutive: Consecutive PRBS pattern
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?

Response	<sequence> = <NR1 NUMERIC RESPONSE DATA> (FIX1)
Function	Queries the PRBS pattern sequence during Mixed Data pattern reception.
Example	`> MPS?` `< MPS 0`
Compatibility	Incompatible with existing models.

MSS <row>,<data>,<prbs>

Parameter	<row> = <DECIMAL NUMERIC PROGRAM DATA>
	1 to 16 1 to 16 rows, in 1-row steps
	<data> = <DECIMAL NUMERIC PROGRAM DATA>
	0 Descramble OFF
	1 Descramble ON
	Note, however, that Descramble OFF is fixed for data of 1 row.
	<prbs> = <DECIMAL NUMERIC PROGRAM DATA>
	0 Descramble OFF
	1 Descramble ON
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>

Parameter	<row> = <DECIMAL NUMERIC PROGRAM DATA>
	1 to 16 1 to 16 rows, in 1-row steps
Response	<data>,<prbs> = <NR1 NUMERIC RESPONSE DATA> (FIX1)
Function	Queries the descramble setting (ON/OFF) for the Data and PRBS of the specified Row.
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.

MRW?

Response	<numeric> = <NR1 NUMERIC RESPONSE DATA> (FIX2)
Function	Queries the number of rows within one block that are edited during Mixed Data pattern reception.
Example	> MRW? < MRW 1 < MRW 16
Compatibility	Incompatible with existing models.

MRL <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 reception.
Example	To set pattern length to 768 bits: > MRL 768
Compatibility	Incompatible with existing models.

MRL?

Response	<numeric> = <NR1 NUMERIC RESPONSE DATA> (FIX11)
Function	Queries the pattern length of 1 row that is edited during Mixed Data pattern reception.
Example	> MRL? < MRL 768 < MRL 2281701376
Compatibility	Incompatible with existing models.

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 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 × 4 bits)

Function

Queries 400 × 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:

```
> MDW? 1,#H0
< MDW #H0123456789ABCDEF...
```

(minimum: 1, maximum: 400 in hexadecimal)

```
< MDW #H0
```

Compatibility

Incompatible with existing models.

MBW <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 #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>]

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)

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.

8.4.3.6 Sequence pattern setting commands

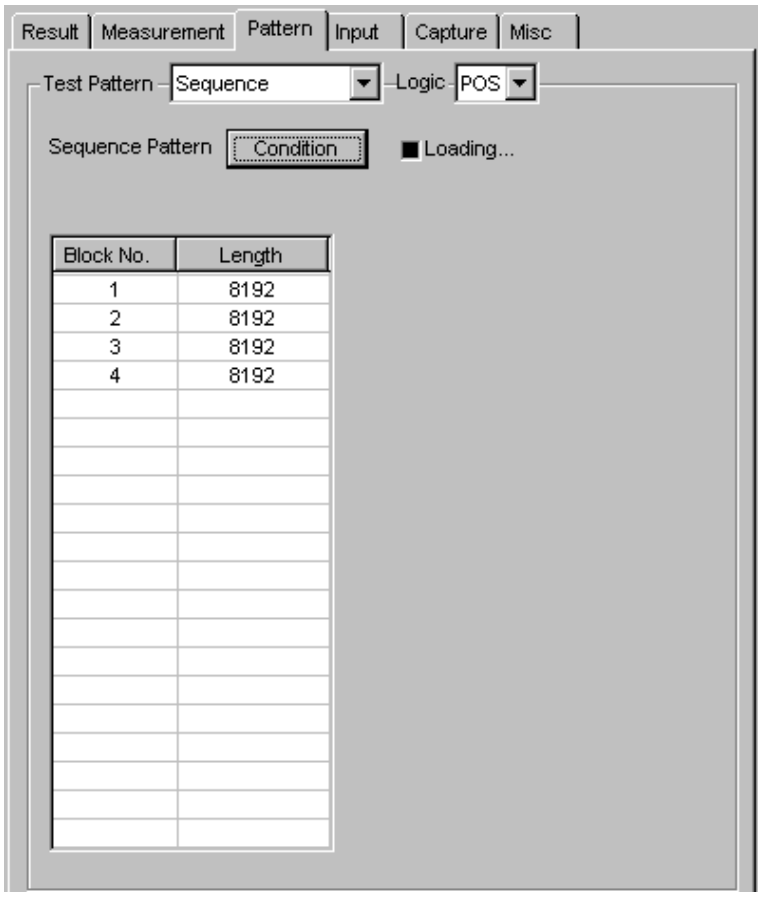


Figure 8.4.3.6-1 Sequence pattern setting

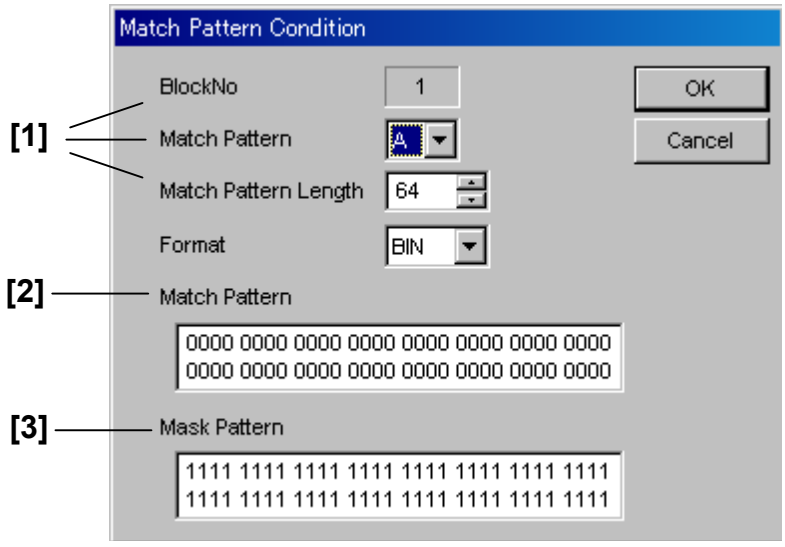


Figure 8.4.3.6-2 Match Pattern Condition

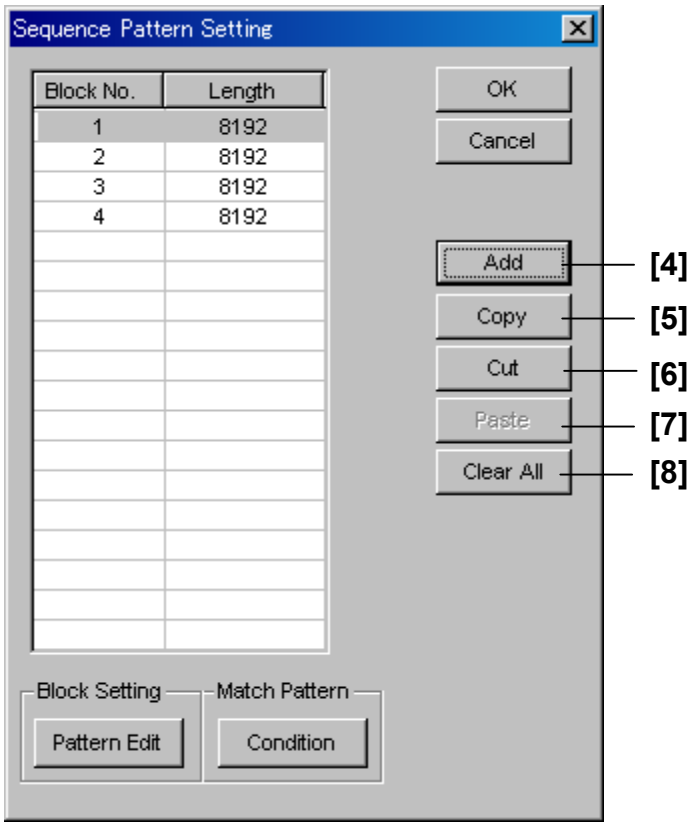


Figure 8.4.3.6-3 Sequence Pattern Setting

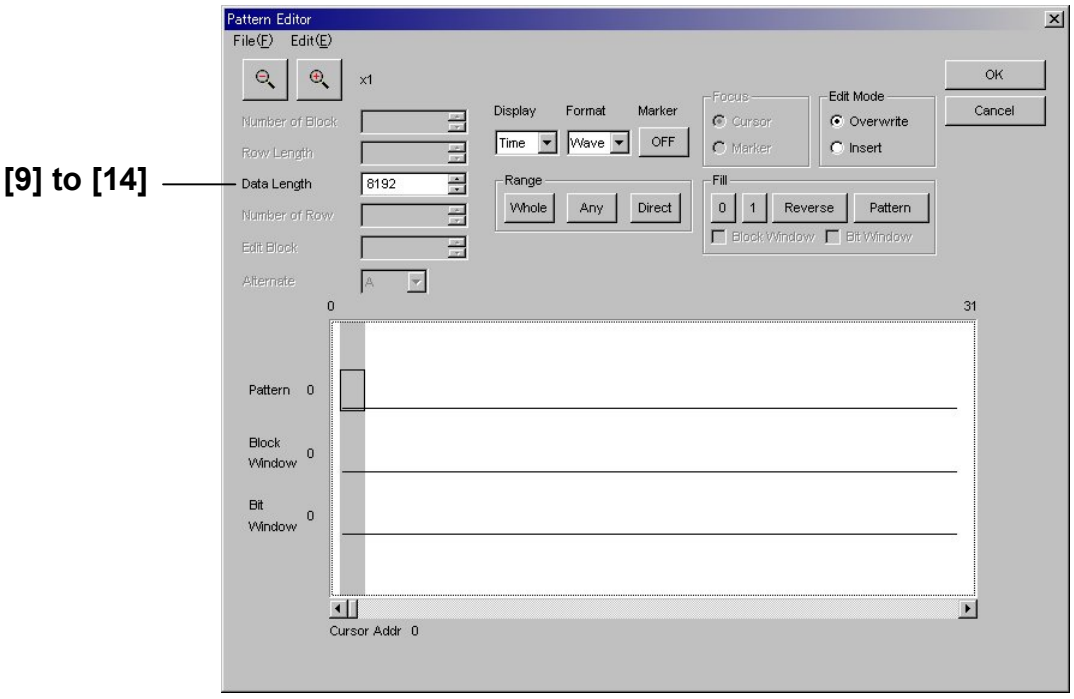


Figure 8.4.3.6-4 Sequence pattern setting (Pattern Editor screen)

Table 8.4.3.6-1 Sequence pattern setting commands

No.	Setting Items	Commands
[1]	BlockNo Match Pattern Match Pattern Length	SQL
		SQL?
[2]	Match Pattern	SQM
		SQM?
[3]	Mask Pattern	SQE
		SQE?
[4]	Add	SAD
		SAD?
[5]	Copy	SCO
[6]	Cut	SCU
[7]	Paste	SPA
[8]	Clear All	SCD
[9]	Number of Block	SLG
		SLG?
[10]	No label (Pattern setting)	SDW
		SDW?
[11]	No label (Pattern setting)	SBW
		SBW?
[12]	No label (Data inversion)	SDA
[13]	No label (Data inversion)	SDD
[14]	No label (All 0s or 1s)	SDF

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 — 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 — 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	<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 reception.
Example	To set the number of blocks that are edited to 10: `> SAD 10`
Compatibility	Incompatible with existing models

SAD?

Response	<block> = <NR1 NUMERIC RESPONSE DATA> (FIX3)
Function	Queries the number of blocks that are edited during Sequence pattern reception.

Example	`> SAD?`
	`< SAD 10`
	`< SAD 128`

Compatibility	Incompatible with existing models

SCO <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 reception.
Example	To copy the settings in block 1 during Sequence pattern reception: `> SCO 1`
Compatibility	Incompatible with existing models.

SCU <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 reception.
Example	To cut the settings in block 1 during Sequence pattern reception: `> SCU 1`
Compatibility	Incompatible with existing models.

SPA<block>

Parameter	<block> = <DECIMAL NUMERIC PROGRAM DATA> 1 to 128 1 to 128 blocks, in 1-block steps
Function	Pastes the settings in the specified block during Sequence pattern reception.
Example	To paste the settings in block 1 during Sequence pattern reception: `> SPA 1`
Compatibility	Incompatible with existing models.

SCD

Function	Clears all the settings during Sequence pattern reception.
Example	`> SCD`
Compatibility	Incompatible with existing models.

SLG <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 8,192 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 length of the specified block during Sequence pattern reception.
Example	To set the length of block 1 to 1 bit during Sequence pattern reception: `> SLG 1,8192`
Compatibility	Incompatible with existing models.

SLG? <block>

Parameter	<block> = <DECIMAL NUMERIC PROGRAM DATA> 1 to 128 1 to 128 blocks, in 1-block steps
Response	<numeric> = <NR1 NUMERIC RESPONSE DATA> (FIX7) 8192 to 1048576 8,192 to 1,048,576 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:
	`> SLG? 1`
	`< SLG 8192`
	`< 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 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> = <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 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 FFFFF addresses 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 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.

Response

<data> = HEXADECIMAL NUMERIC RESPONSE DATA>

#H***...*	Specify 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 reception.

Example

To query 400 × 4 bits of pattern data from 0 address:

```
> SDW? 1,#H0
< SDW #H0123456789ABCDEF...
```

(minimum: 1, maximum: 400 in hexadecimal)

```
< 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 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 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)

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

Queries 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 during Sequence pattern reception.

Example

To query 2 bytes of block 1 data from 0 address:

```
> SBW? 1,#H0,2
< SBW #12DD
```

To query the setting pattern of block 1 data from 0 address:

```
> SBW? 1,#H0
< SBW #516000AA...
```

(16,000 bytes of binary data)

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 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

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>

Parameter	<block> = <DECIMAL NUMERIC PROGRAM DATA>
	1 to 128 — 1 to 128 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 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 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 in block 1:
	`> SDF 1,0,1,0`
Compatibility	Incompatible with existing models.

8.4.3.7 Mask setting commands

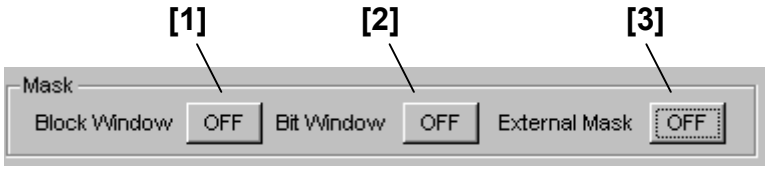


Figure 8.4.3.7-1 Mask ON/OFF setting

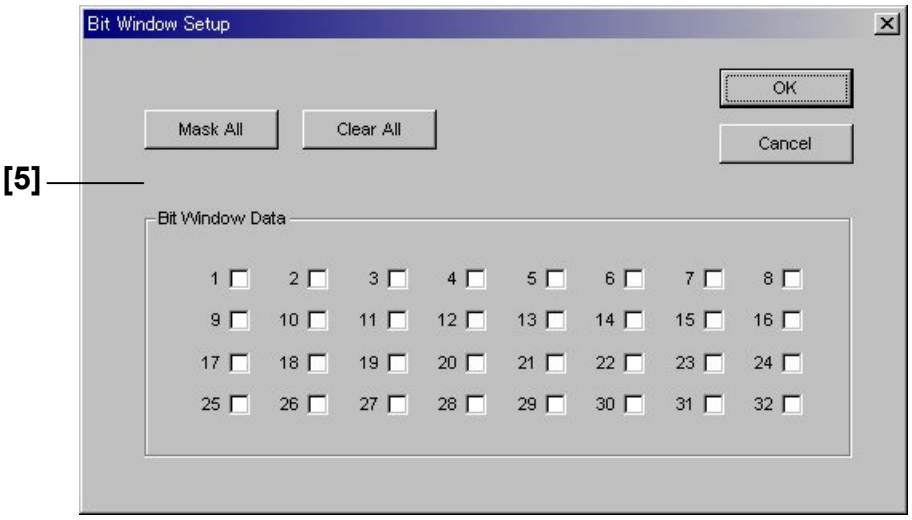


Figure 8.4.3.7-2 Bit Window setting (PRBS pattern)

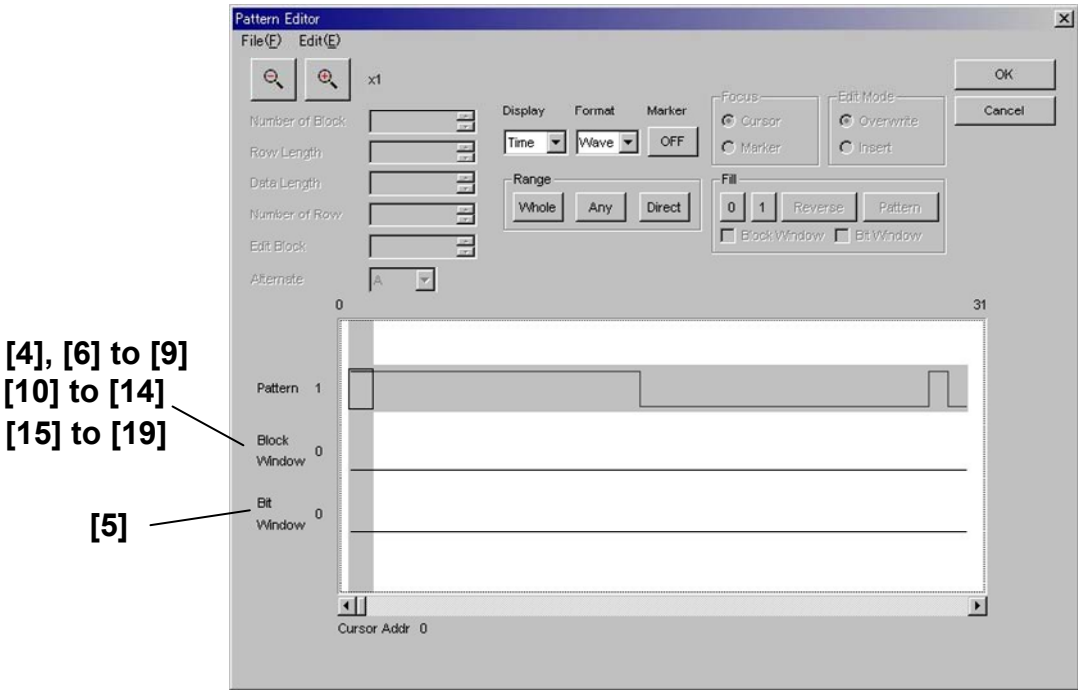


Figure 8.4.3.7-3 Bit Window, Block Window setting
(Zero-Substitution, Data, Mixed-Data, Sequence pattern)

Table 8.4.3.7-1 Mask setting commands

No.	Setting Items	Commands
[1]	Block Window ON/OFF	MGE
		MGE?
[2]	Bit Window ON/OFF	MSE
		MSE?
[3]	External Mask ON/OFF	MEX
		MEX?
[4]	Block Window (Zero-Substitution, Data)	MGB
		MGB?
[5]	Bit Window	CHM
		CHM?
[6]	Block Window (Pattern setting)	BDM
		BDM?
[7]	Block Window (Data inversion)	DRM
[8]	Block Window (Data inversion)	DBM
[9]	Block Window (All 0s or 1s)	ALB
[10]	Block Window (Mixed Data)	MDB
		MDB?
[11]	Block Window (Pattern setting)	MBM
		MBM?
[12]	Block Window (Data inversion)	MDM
[13]	Block Window (Data inversion)	MMB
[14]	Block Window (All 0s or 1s)	MFB
[15]	Block Window (Sequence)	MSB
		MSB?
[16]	Block Window (Pattern setting)	SBM
		SBM?
[17]	Block Window (Data inversion)	SDM
[18]	Block Window (Data inversion)	SDB
[19]	Block Window (All 0s or 1s)	SMF

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>

Parameter	<numeric> = <DECIMAL NUMERIC PROGRAM DATA>
	0 External Mask OFF
	1 External Mask ON
Function	Sets External Mask ON/OFF to the received pattern.
Example	To set External Mask ON to the received pattern: `> MEX 1`
Compatibility	Incompatible with existing models.

MEX?

Response	<numeric> = <NR1 NUMERIC RESPONSE DATA> (FIX1)
Function	Queries the External Mask ON/OFF state of the received pattern.
Example	`> MEX?` `< MEX 1`
Compatibility	Incompatible with existing models.

MGB <start>,<end>,<data>

Parameter

<start> = <NON-DECIMAL PROGRAM DATA>

#H0 to #HF7FFFFFF	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 #HF7FFFFFF	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 × 4 bits)

Function

Queries 400 × 4 bits of mask pattern data from the <start> address during Block Window setting.

Example

To query 400 × 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.

CHM?

Response	<data> = <HEXADECIMAL NUMERIC RESPONSE DATA>
Function	Queries the mask pattern data during Bit Window setting

Example	`> CHM?`
	`< CHM #H01234567`
	`< CHM #H00000000`

Compatibility	Incompatible with existing models.

BDM <start>,<end>,<bdata>

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. <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>: Binarydata up to 16,777,216 bytes
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 7FFFFFF addresses during Block Window setting: `> BDM #H0,#H7FFFFFF,#11A`
Compatibility	Incompatible with existing models.

BDM? <start>[,<size>]

Parameter

<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 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 #516000AAAAAA...
```

(binary data of 16,000 bytes)

Compatibility

Incompatible with existing models.

DRM <start>,<end>

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 >

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:

```
> DRM #H0,#H7FFFFFF
```

Compatibility

Incompatible with existing models.

DBM <start>,<delta>

Parameter

<start> = <NON-DECIMAL PROGRAM DATA>

#H0 to #H7FFFFFF 0 to 7FFFFFF 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 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:

```
> DBM #H0,1
```

Compatibility

Incompatible with existing models.

ALB <range>,<page>,<data>

Parameter	<range> = <CHARACTER 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 = 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 mask pattern data during Block Window setting.	
Example	To set 0s for the first page of the mask pattern data during Block Window setting: `> ALB 0,1,0`	
Compatibility	Incompatible with existing models.	

MDB <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 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.

MDB? <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 #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>

Function

Queries 400 × 4 bits of Block Window mask pattern data from the <start> to <end> addresses during Mixed Data pattern setting.

Example

To query 400 × 4 bits of mask pattern data from bit 0 address during Block Window setting:

```
> MDB? 1,#H0
< MDB #H1
```

Compatibility

Incompatible with existing models.

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 7FFFFF bits, in 1-bit steps (Specify in hexadecimal)

<end> = <NON-DECIMAL PROGRAM DATA >

#H0 to #H7FFFFFF 0 to 7FFFFF 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 7FFFFFF addresses during Mixed Data pattern setting or Block Window setting:

```
> MBM 1,#H0,#H7FFFFFF,#11A
```

Compatibility

Incompatible with existing models.

MBM? <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 7FFFFF 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:

```
> 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):

```
> MBM? 1,#H0
< MBM #516000AAAAAA...  (binary data of 16,000 bytes)
```

Compatibility

Incompatible with existing models.

MDM <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 #H7FFFFFF 0 to 7FFFFF bits, in 1-bit steps (Specify in hexadecimal)

<end> = <NON-DECIMAL PROGRAM DATA >

#H0 to #H7FFFFFF 0 to 7FFFFF 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 7FFFFFF 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 1 to 511 blocks, in 1-block steps

<start> = <NON-DECIMAL PROGRAM DATA >

#H0 to #H7FFFFFF 0 to 7FFFFF 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 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>

Parameter

<block> = <DECIMAL NUMERIC PROGRAM DATA>

1 to 511	1 to 511 blocks, in 1-block steps

<range> = <CHARACTER 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 = 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 Block Window mask pattern during Mixed Data pattern setting.

Example

To set 0s for the first page of the Block Window mask pattern during Mixed Data pattern setting:

```
> MFB 1,0,1,0
```

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,#HFFFFF,#H1
```

Compatibility

Incompatible with existing models.

MSB? <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 FFFFF bits, in 1-bit steps (Specify in hexadecimal)

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 #H1
```

Compatibility

Incompatible with existing models.

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 >

#XYYY<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 7FFFFFF 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:

```
> 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):

```
> 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>

Parameter

<block> = < DECIMAL NUMERIC PROGRAM DATA>

1 to 128	1 to 128 Blocks, in 1-block steps

<range> = <CHARACTER 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 8192 (Max.) pages, in 1-page steps

$$\text{Max} = \frac{\text{Pattern Length}}{128} \quad \text{(rounding up fractions)}$$

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 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

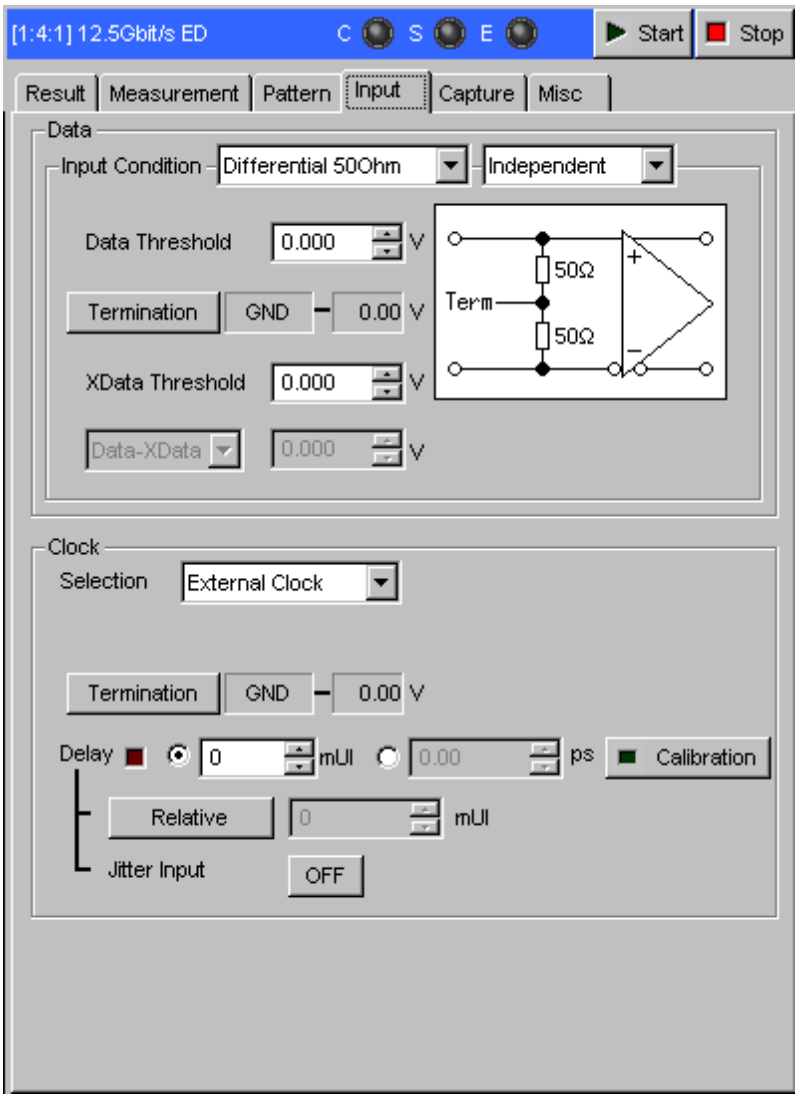


Figure 8.4.4-1 Input tab setting

8.4.4.1 Data Input setting commands

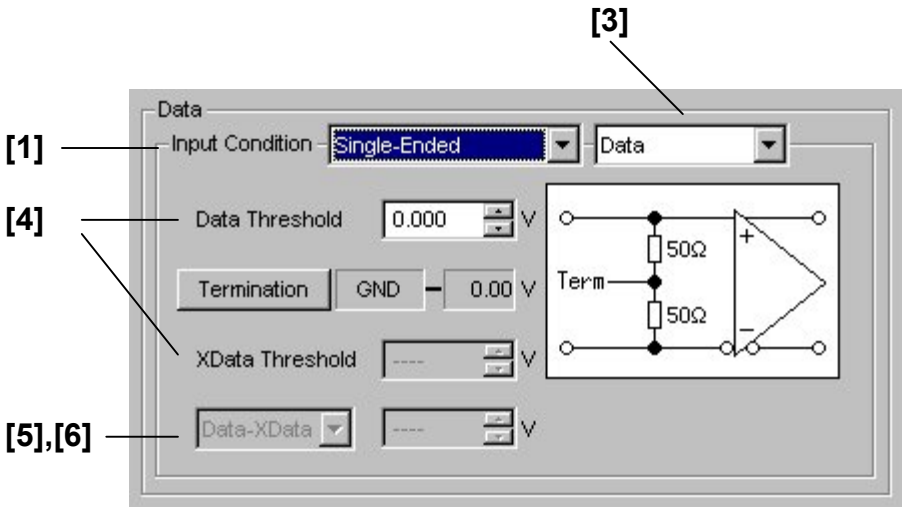


Figure 8.4.4.1-1 Single-End Data

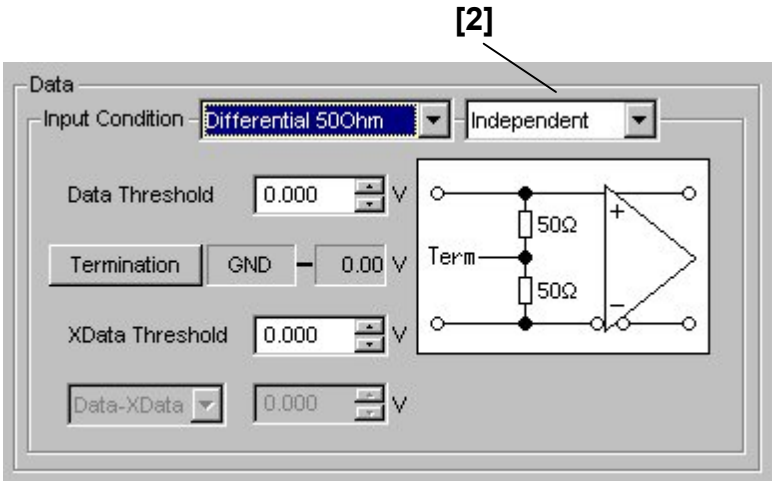


Figure 8.4.4.1-2 Differential 50 Ohm

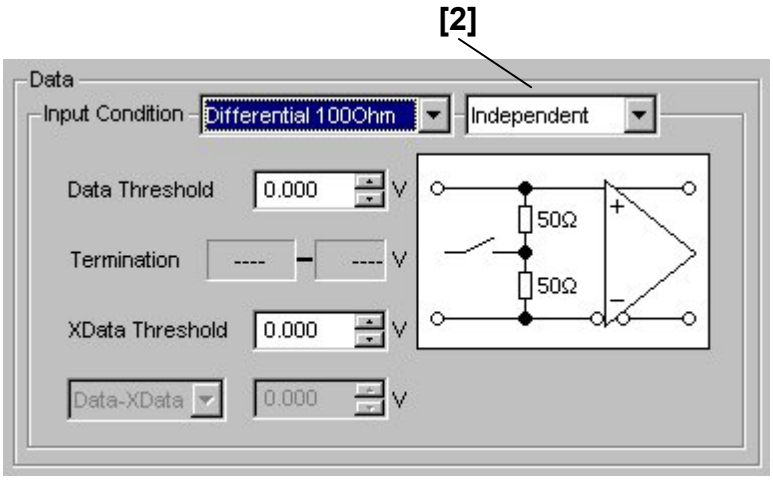


Figure 8.4.4.1-3 Differential 100 Ohm

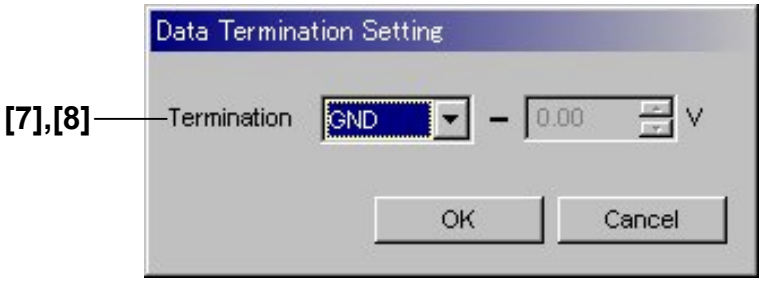


Figure 8.4.4.1-4 Termination Setting

Table 8.4.4.1-1 Data Input setting commands

No.	Setting Items	Commands
[1]	Input Condition	DSD
		DCD?
[2]	Differential type setting	DAC
		DCD?
[3]	Data input port setting	DSL
		DCD?
[4]	Data Threshold XData Threshold	DTH
		DTH?
[5]	Differential reference signal setting	DDB
		DDB?
[6]	Differential data input threshold setting	DDT
		DDT?
[7]	Termination	DTM
		DTM?
[8]	No label (Termination voltage setting)	DTL
		DTL?

DSD <interface>

Parameter	<interface> = <DECIMAL NUMERIC PROGRAM DATA>
	0 Single Ended
	1 Differential 50 Ω
	2 Differential 100 Ω
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>

Parameter	<tracking> = <DECIMAL NUMERIC PROGRAM DATA>
	0 Independent
	1 Tracking
	2 Alternate
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>

Parameter	<port> = <DECIMAL NUMERIC PROGRAM DATA>
	0 Data
	1 XData
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.

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	`> DCD?`	
	`< DCD 0,0,0`	
Compatibility	Incompatible with existing models.	

DTH <numeric>

Parameter	<numeric> = <DECIMAL NUMERIC PROGRAM DATA>	
	–3.500 to 3.300	–3.500 to 3.300 V, in 0.001-V steps (when MU181040A-002 and MU181040B-002 are installed)
	–0.350 to 0.350	–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:	
	`> 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	`> 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	<basis> = <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: `> DDB 0`
Compatibility	Incompatible with existing models.

DDB?

Response	<basis> = <NR1 NUMERIC RESPONSE DATA> (FIX1)
Function	Queries the differential reference signal for the data input threshold value.
Example	`> DDB?` `< DDB 0`
Compatibility	Incompatible with existing models.

DDT <numeric>

Parameter	<numeric> = <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: `> DDT -3.000`
Compatibility	Incompatible with existing models.

DDT?

Response	<numeric> = <NR2 NUMERIC RESPONSE DATA> (FIX6)
Function	Queries the data input threshold value for differential input.

Example	`> DDT?`
	`< DDT -3.000`
	`< DDT  3.000`

Compatibility	Incompatible with existing models.

DTM <term>

Parameter	<term> = <DECIMAL NUMERIC PROGRAM DATA>
	0 GND (50 Ω to GND)
	1 NECL (50 Ω to –2 V)
	3 LVPECL (50 Ω to 1.3 V)
	4 VARIABLE (50 Ω to Variable Voltage)
	5 PCML (50 Ω to 3.3 V)
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	`> DTM?` `< DTM 0`
Compatibility	Partially compatible with the MP1762D (0: GND).

DTL <numeric>

Parameter	<numeric> = <DECIMAL 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 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	`> DTL?`
	`< DTL -2.00`
	`< DTL  3.50`

Compatibility	Incompatible with existing models.

8.4.4.2 Clock Input setting commands

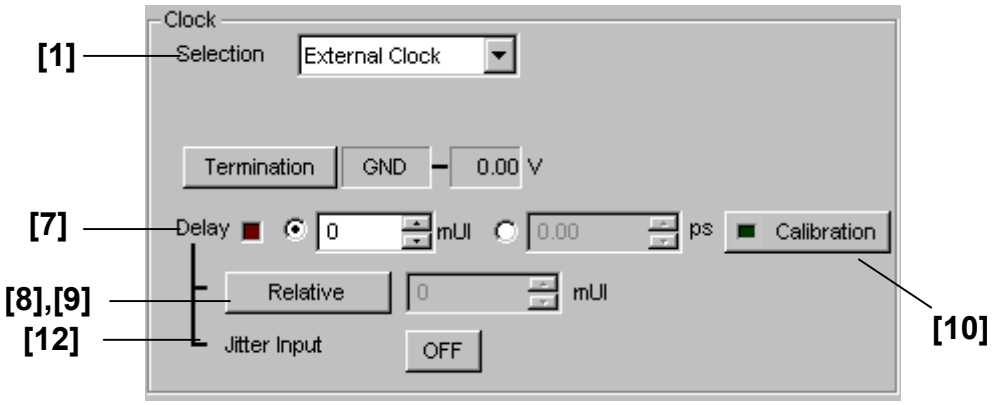


Figure 8.4.4.2-1 External Clock setting

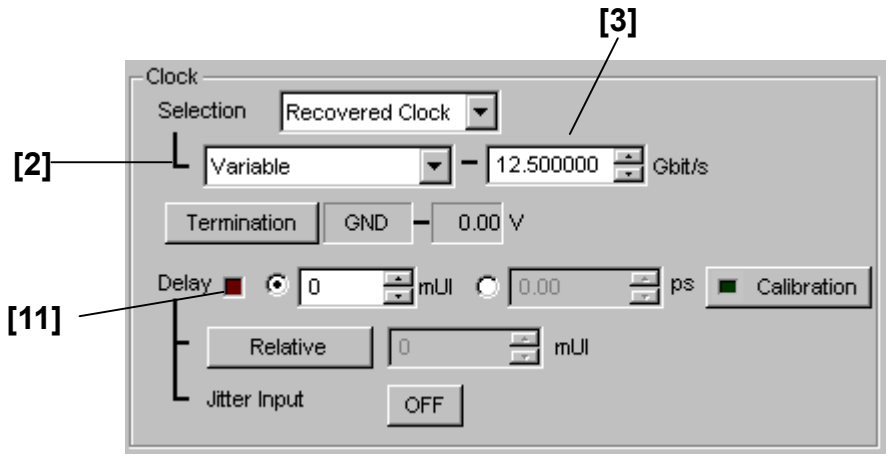


Figure 8.4.4.2-2 Recovered Clock setting

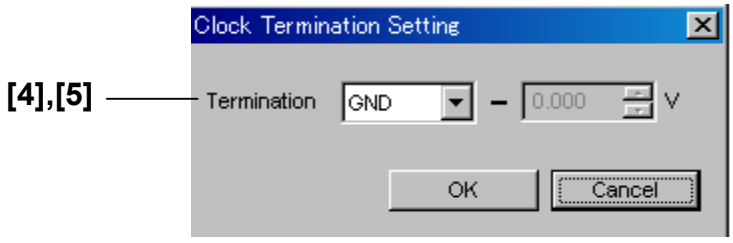


Figure 8.4.4.2-3 Clock Termination setting dialog box

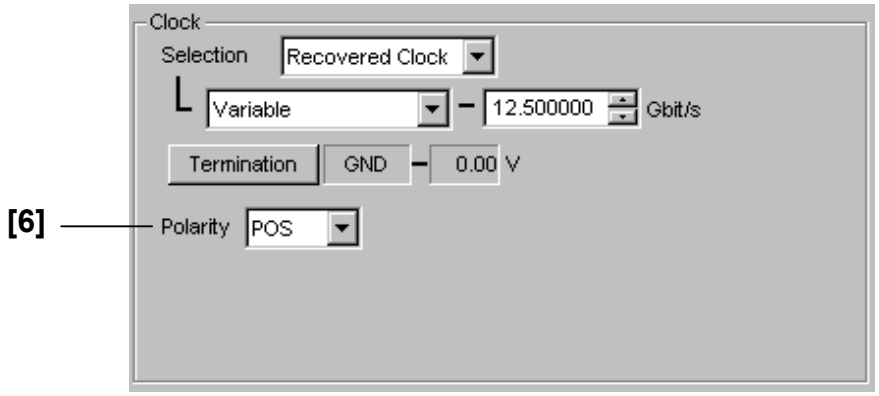


Figure 8.4.4.2-4 Clock Input setting (For no delay option)

Table 8.4.4.2-1 Clock Input setting commands

No.	Setting Items	commands
[1]	Selection	RRC
[2]	Setting for specific frequency of clock recovery	RFQ
		DRC?
[3]	Setting for clock recovery frequency	CFQ
		CFQ?
[4]	Termination	CTM
		CTM?
[5]	No label (Termination voltage setting)	CTL
		CTL?
[6]	Polarity	CRC
		CRC?
[7]	Delay (mUI setting)	CPA
		CPA?
	Delay (ps setting)	CPU
		CPU?
[8]	Relative	CRO
		CRO?
[9]	No label (Relative value setting)	CRD
		CRD?
[10]	Calibration	CCA
[11]	No label (LED)	DLY?
[12]	Jitter Input	CJO
		CJO?

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	SATA1.5Gb/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
	13	SATA 6Gb/s	6.000000 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.		

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 4.900000 to 6.250000 Gbit/s /1 kbit/s Step 9.800000 to 12.500000 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.

CFQ?

Response	<numeric> = <NR2 NUMERIC RESPONSE DATA> (FIX9)
Function	Queries the frequency for the clock data recovery.
Example	> CFQ? < CFQ 0.100000 < CFQ 12.500000
Compatibility	Incompatible with existing models.

CTM <term>

Parameter	<term> = <NR1 NUMERIC PROGRAM DATA>
	0 GND (50 Ω to GND)
	1 NECL (50 Ω to –2 V)
	3 LVPECL (50 Ω to 1.3 V)
	4 VARIABLE (50 Ω to Variable Voltage)
	5 PCML (50 Ω to 3.3 V)
Function	Sets the clock input termination condition.
Example	To set the clock input termination condition to GND: > CTM 0
Compatibility	Partially compatible with the MP1632C and MP1762D (0: GND, 1: ECL).

CTM?

Response	<term> = <NR1 NUMERIC RESPONSE DATA> (FIX1)
Function	Queries the clock input termination condition.
Example	> CTM? < CTM 0
Compatibility	Partially compatible with the MP1632C and MP1762D (0:GND, 1: ECL).

CTL <numeric>

Parameter	<numeric> = <DICIMAL NUMERIC PROGRAM DATA>
	–2.50 to 3.50 –2.50 to 3.50 V, in 0.01 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 –2.00 V: > CTL -2.00
Compatibility	Incompatible with existing models.

CTL?

Response	<numeric> = <NR2 NUMERIC RESPONSE DATA> (FIX5)
Function	Queries the termination voltage when the input termination condition is set to Variable.

Example	`> CTL?`
	`< CTL -2.00`
	`< CTL 3.00`

Compatibility	Incompatible with existing models.

CRC <logic>

Parameter	<logic> = <NR1 NUMERIC PROGRAM DATA> 0 Positive 1 Negative
Function	Selects the phase unit for the clock recovery clock.
Example	To set the phase unit for the clock recovery clock to Positive: `> CRC 0`
Compatibility	Incompatible with existing models.

CRC?

Response	<logic> = <NR1 NUMERIC RESPONSE DATA> (FIX1)
Function	Queries the phase unit for the clock recovery clock.
Example	`> CRC?` `< CRC 0`
Compatibility	Incompatible with existing models.

CPA <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) 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: `> CPA -10000`
Compatibility	Compatible with the MP1632C and MP1762D.

CPA?

Response	<numeric> = <NR1 NUMERIC RESPONSE DATA> <NR2 NUMERIC RESPONSE DATA> (FIX7)
Function	Queries value of the clock input phase in ps units.
Example	> CPA? < CPA -1000.0 > CPA 10000
Compatibility	Compatible with the MP1632C and MP1762D.

CPU <numeric>

Parameter	<numeric> = <DECIMAL NUMERIC PROGRAM DATA> –1000 to 1000 –1,000 to 1,000 mUI, in 1-mUI steps
Function	Sets the value of the clock input phase in mUI units (only when MU181040A-x30 and MU181040B-x30 are installed).
Example	To set the clock input phase to –1,000 mUI in mUI units: > CPU -1000
Compatibility	Incompatible with existing models.

CPU?

Response	<numeric> = <NR1 NUMERIC RESPONSE DATA> (FIX5)
Function	Queries the value of the clock input phase in mUI units.
Example	> CPU? < CPU -1000 < CPU 1000
Compatibility	Incompatible with existing models.

CRO <numeric>

Parameter	<numeric> = <DECIMAL NUMERIC PROGRAM DATA> 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.

CRO?

Response	<numeric> = <NR1 NUMERIC RESPONSE DATA> (FIX1)
Function	Queries the clock input phase variable reference setting.
Example	`> CRO?` `< CRO 1`
Compatibility	Incompatible with existing models.

CRD <numeric>[,<unit>]

Parameter	<numeric> = <DECIMAL NUMERIC PROGRAM DATA>
	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)
	[<unit>] = <NR1 NUMERIC PROGRAM DATA>
	0 mUI unit
	1 ps unit (The ps unit is selected when <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: `> CRD -1000,0`
Compatibility	Incompatible with existing models.

CRD? [<unit>]

Parameter	[<unit>] = <NR1 NUMERIC PROGRAM DATA>
	0 mUI unit
	1 ps unit (The ps unit is selected when <unit> is omitted.)
Response	<numeric> = <NR1 NUMERIC RESPONSE DATA> <NR2 NUMERIC RESPONSE DATA> (FIX7)
Function	Queries the value of the clock input phase variable reference.

Example	`> CRD? 0`
	`< CRD -1000`
	`< 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	<numeric> = <NR1 NUMERIC RESPONSE DATA> (FIX1) 0 Ready status 1 Busy status
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	<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: `> CJO 1`
Compatibility	Incompatible with existing models.

CJO?

Response	<numeric> = <NR1 NUMERIC RESPONSE DATA> (FIX1) 0 Jitter input OFF setting 1 Jitter input ON setting
Function	Queries the Jitter Input button setting.
Example	`> CJO?` `< CJO 1`
Compatibility	Incompatible with existing models.

8.4.5 Commands related to Capture tab

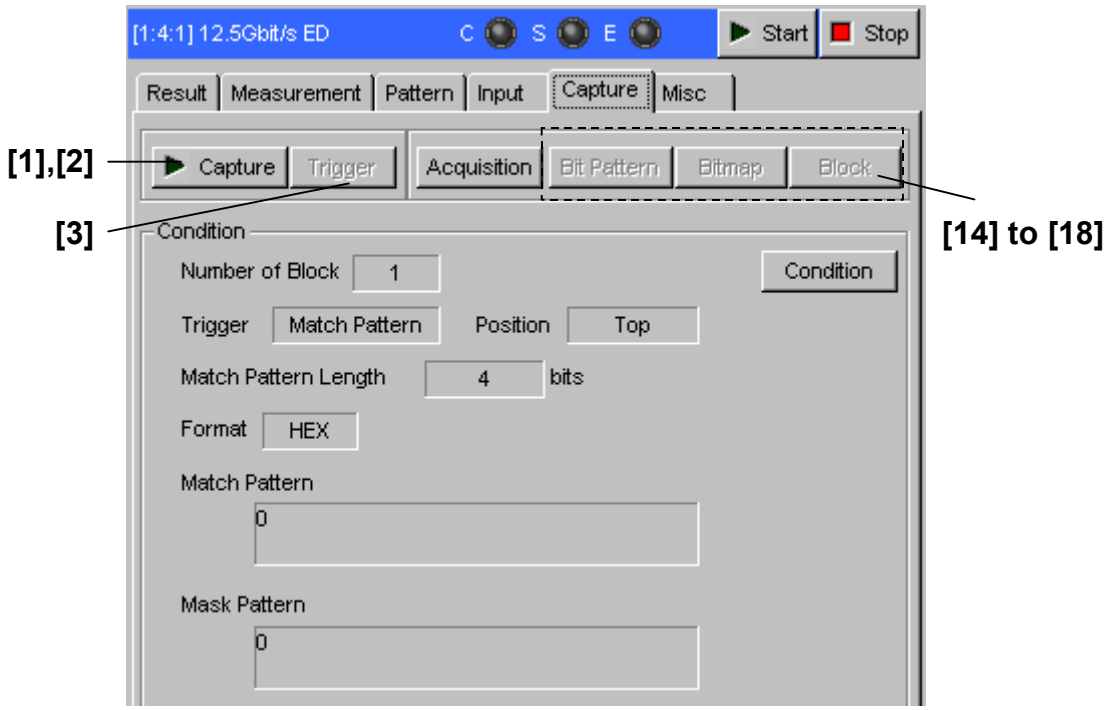


Figure 8.4.5-1 Capture tab (initial state)

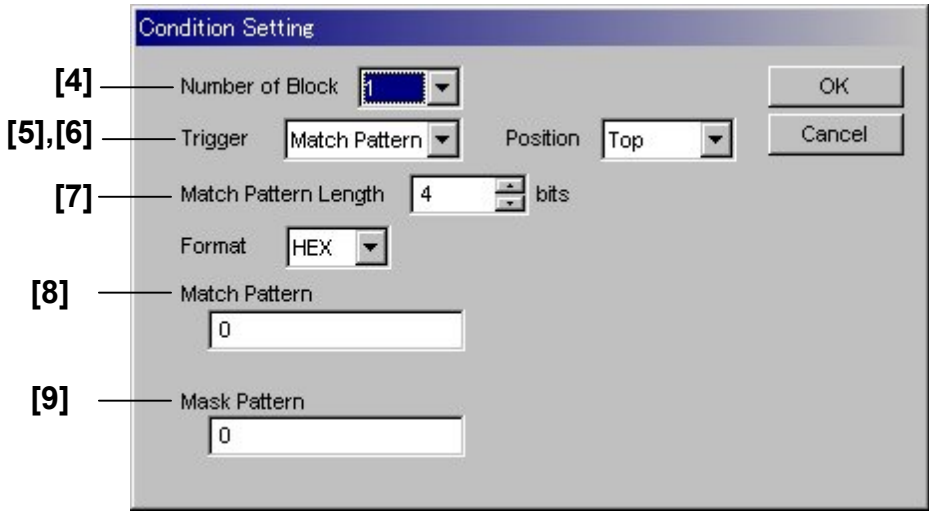


Figure 8.4.5-2 Condition Setting

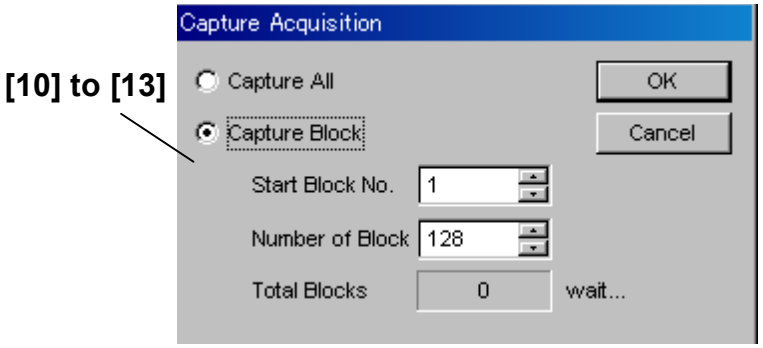


Figure 8.4.5-3 Setting for Capture Data reading

Table 8.4.5-1 Capture tab setting commands

No.	Setting Items	Commands
[1]	Capture	CSA
[2]	Capture	CSP
[3]	Trigger	CTG
[4]	Number of Block	CBC
		CBC?
[5]	Trigger	BTG
		BTG?
[6]	Position	CPS
		CPS?
[7]	Match Pattern Length	CMP
		CMP?
[8]	Match Pattern	MPE
		MPE?
[9]	Mask Pattern	CMK
		CMK?
[10]	Query for capture data block count	CAB?
[11]	Start of capture data acquisition	CAS
[12]	Query for capture data acquisition state	CAT?
[13]	Cancel of capture data acquisition	CAC
[14]	Query for capture data length per block	CBL
[15]	Query for capture result data	CDW?
[16]	Query for capture result data	CBW?
[17]	Query for error position in capture result	CED?
[18]	Query for error position in capture result	CEB?

CSA

Function	Starts capture result acquisition.
Example	`> CSA`
Compatibility	Incompatible with existing models.

CSP

Function	Stops capture result acquisition.
Example	`> CSP`
Compatibility	Incompatible with existing models.

CTG

Function	Generates a capture manual trigger.
Example	`> CTG`
Compatibility	Incompatible with existing models.

CBC <numeric>

Parameter	<numeric> = <DECIMAL NUMERIC PROGRAM DATA> 1, 2, 4, 8, 16, 32, 64, 128 Capture block division number
Function	Sets the capture block division number.
Example	To set the capture block division number to 1: `> CBC 1`
Compatibility	Incompatible with existing models.

CBC?

Response	<numeric> = <NR1 NUMERIC RESPONSE DATA> (FIX3)
Function	Queries the capture block division number.

Example	`> CBC?`
	`< CBC 1`
	`< CBC 128`

Compatibility	Incompatible with existing models.

BTG <trigger>

Parameter	<trigger> = <DECIMAL NUMERIC PROGRAM DATA>	
	0	Error Detect: When an error is detected
	1	When matches the match pattern
	2	When a manual trigger is generated
	4	When external trigger is selected
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: `> BTG 0`	
Compatibility	Incompatible with existing models.	

BTG?

Response	<trigger> = <NR1 NUMERIC RESPONSE DATA> (FIX1)
Function	Queries the trigger to store the data to memory for the capture function.
Example	`> BTG?` `< BTG 0`
Compatibility	Incompatible with existing models.

CPS <pos>

Parameter	<pos> = <DECIMAL NUMERIC PROGRAM DATA>	
	0	Store the data to the top of the memory.
	1	Store the data in the middle of the memory.
	2	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 for the capture function to the top of the memory: `> CPS 0`	
Compatibility	Incompatible with existing models.	

CPS?

Response	<pos> = <NR1 NUMERIC RESPONSE DATA> (FIX1)
Function	Queries the data storage memory position for the capture function.
Example	`> CPS?` `< CPS 0`
Compatibility	Incompatible with existing models.

CMP <numeric>

Parameter	<numeric> = <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 match pattern length to 4 bits: `> CMP 4`
Compatibility	Incompatible with existing models.

CMP?

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>

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 match pattern.
Example	To edit one character of the capture trigger match pattern to "A": `> MPE #HA`
Compatibility	Incompatible with existing models.

MPE?

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.

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
<length> = <NR1 NUMERIC RESPONSE DATA> <FIX10>
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 4 ch Combination.

Function
Queries capture data length per block.

Example
To query capture data length when the number of blocks is 32:

```
> CBL?
< CBL    4194304
```

Compatibility
Incompatible with existing models.

CDW? <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 #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 data of 400 × 4 bits from <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 × 4 bits from address 0 in block 3 acquired by capture function:

```
> CDW? 3,#H0
< CDW #H001
```

Compatibility
Incompatible with existing models.

CBW? <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 #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 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 binary data of 16,777,216 bytes from <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	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.

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 #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 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

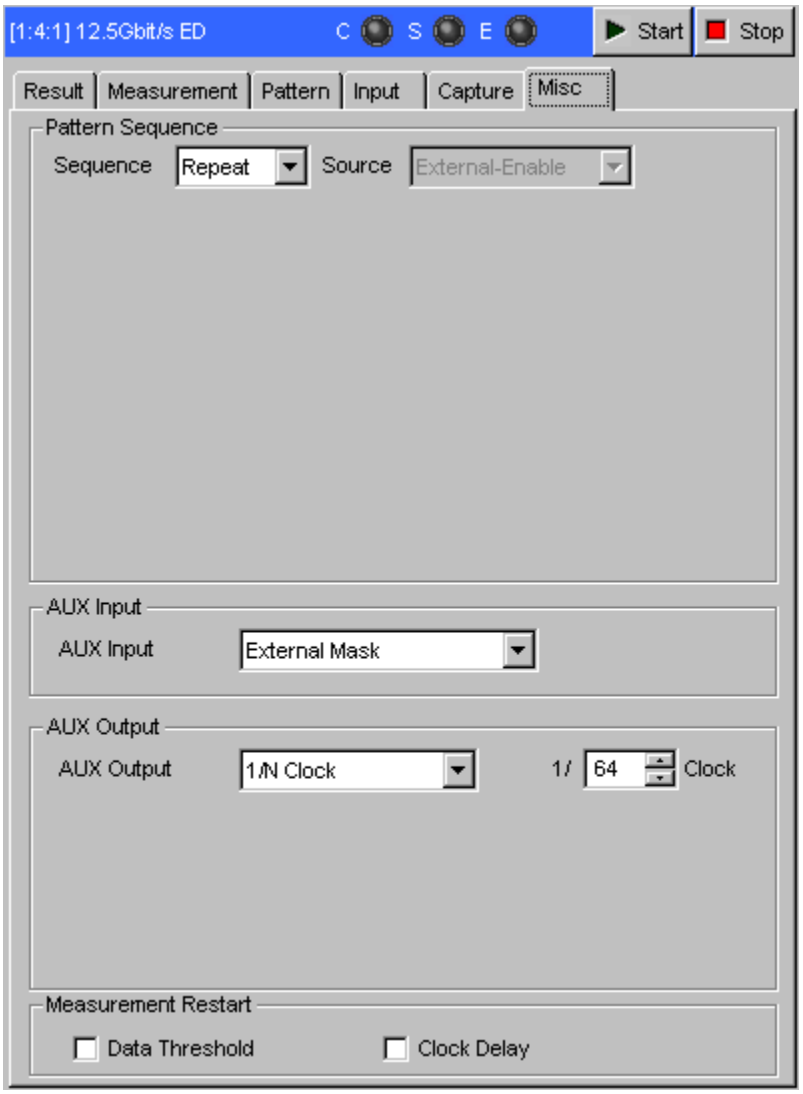


Figure 8.4.6-1 Misc tab

8.4.6.1 Pattern Sequence setting commands

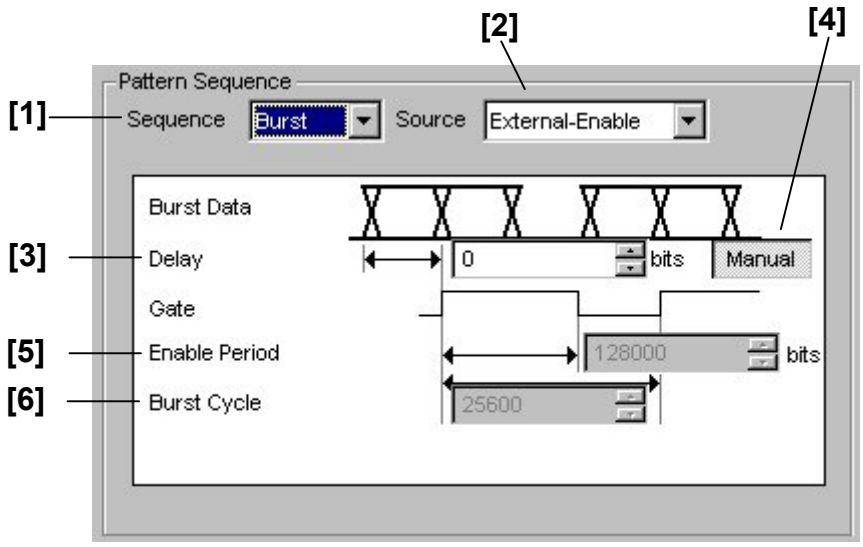


Figure 8.4.6.1-1 Pattern Sequence setting

Table 8.4.6.1-1 Pattern Sequence setting commands

No.	Setting Items	Commands
[1]	Sequence	BST(PTI)
		BST?(PTI?)
[2]	Source	BSM
		BSM?
[3]	Burst Trigger Delay	ABD
		ABD?
[4]	No label (Automatic Adjustment)	BAD
		BAD?
[5]	Enable Period	BEL
		BEL?
[6]	Burst Cycle	BCY
		BCY?

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.

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	<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 for Burst signal reception to 12,800 bits: `> BEL 12800`
Compatibility	Incompatible with existing models.

BEL?

Response	<numeric> = <NR1 NUMERIC RESPONSE DATA> (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	<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: `> BCY 25600`
Compatibility	Incompatible with existing models.

BCY?

Response	<numeric> = <NR1 NUMERIC RESPONSE DATA> (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

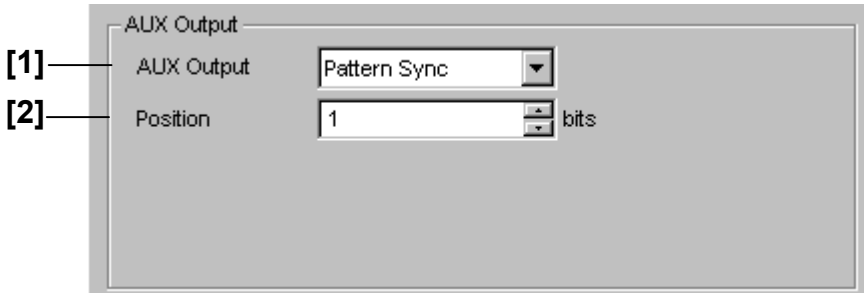


Figure 8.4.6.2-1 Pattern Sync setting
(For PRBS, Zero Substitution, and Data patterns)

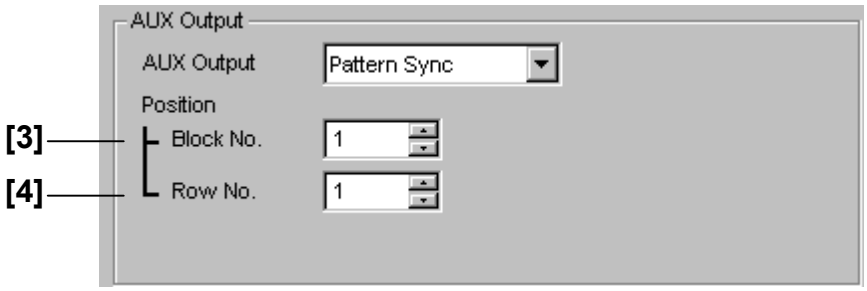


Figure 8.4.6.2-2 Pattern Sync setting (For Mixed pattern)

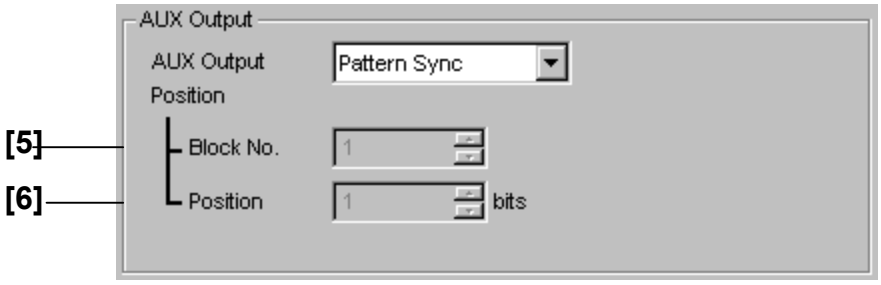


Figure 8.4.6.2-3 Pattern Sync setting (For Sequence pattern)

Table 8.4.6.2-1 AUX Output setting commands

No.	Setting Items	Commands
[1]	AUX Output	SOP
		SOP?
[2]	Position (PRBS, Zero Substitution, Data)	PSP
		PSP?
[3]	Block No. (Mixed Data)	SMB
		SMB?
[4]	Row No. (Mixed Data)	SMR
		SMR?
[5]	Block No. (Sequence)	SSB
		SSB?
[6]	Position (Sequence)	SSP
		SSP?

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).

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	`> 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: `> 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	`> PSP?`
	`< PSP  1`
	`< PSP  2147483633`

Compatibility	Partially compatible with the MP1632C and MP1762D (16-bit resolution (steps) compatible).

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

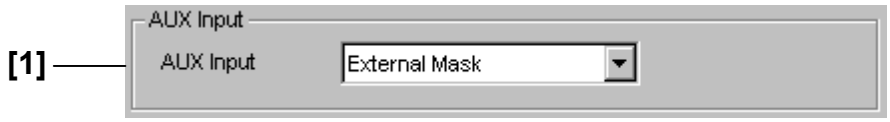


Figure 8.4.6.3-1 AUX Input setting

Table 8.4.6.3-1 AUX Input setting commands

No.	Setting Items	Commands
[1]	AUX Input	AUS
		AUS?

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

Figure 8.4.6.4-1 Measurement Restart setting

Table 8.4.6.4-1 Measurement Restart setting commands

No.	Setting Items	Commands
[1]	Measurement Restart	MRS
		MRS?

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.

MRS?

Response

<data>,<clock> = <NR1 NUMERIC RESPONSE DATA> (FIX1)

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> = <NR1 NUMERIC PROGRAM DATA> (FIX1)

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

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

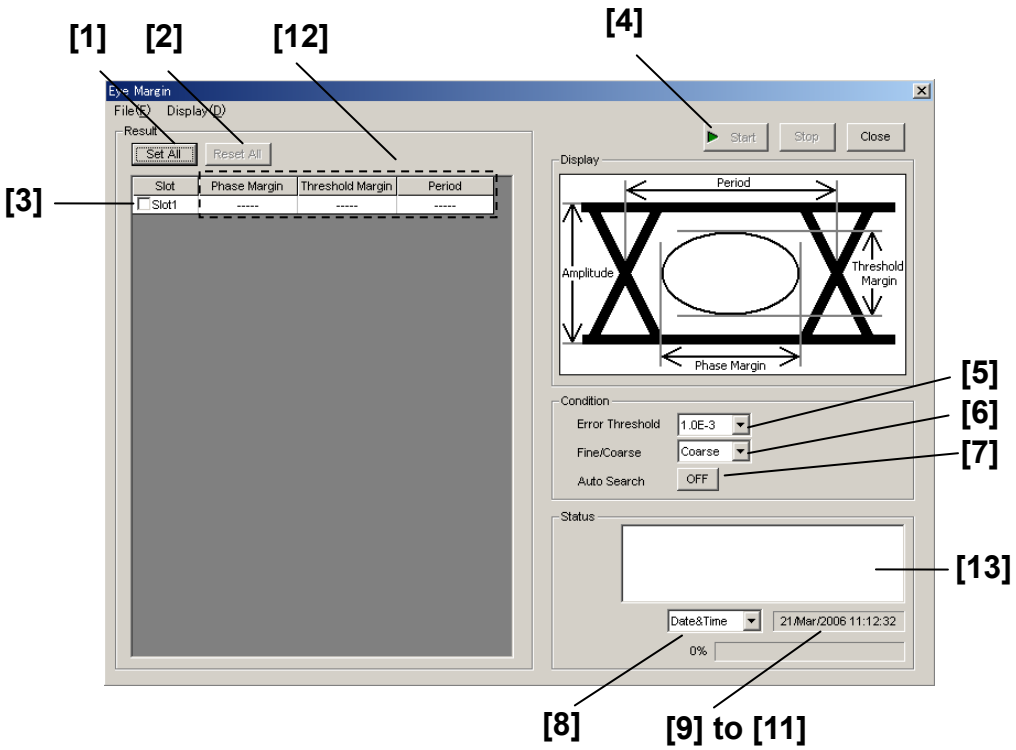


Figure 8.5.1-1 Eye Margin measurement screen

Table 8.5.1-1 Eye Margin measurement setting commands

No.	Setting Items	Commands
[1]	Set All	MEA
[2]	Reset All	MER
[3]	Slot	MES
		MES?
[4]	Start	EST
		EST?
[5]	Error Threshold	EYT
		EYT?
[6]	Fine/Coarse	MMR
		MMR?
[7]	Auto Search ON/OFF	EMA
		EMA?
[8]	Selection of measurement time display	MTI
		MTI?
[9]	Setting of time display	MDT?
[10]	Query for measurement start time	ESA?
[11]	Query for measurement elapsed time	MLP?
[12]	Query for measurement result	EMM?
[13]	Query for measurement status	EMW?

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.

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.

EST?

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>

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?

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.

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.

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 star 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.

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.

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

Form	Format	Description
Form1 Phase type	XXXXXXX	When 0 to MAX (mUI/ps) Displays in the selected unit
	Failed	For measurement failure
	------	When no data corresponds to the query
Form2 Threshold type	XXXXX	For 0 to MAX (mVpp)
	Failed	For measurement failure
	------	When no data corresponds to the query

Function

Queries the Eye Margin measurement result.

Table 8.5.1-3 Parameters

Items	<result>	Response Format
Phase margin	0	Form1
Threshold margin	1	Form2
Period	2	Form1

Example

Queries the phase margin result of the Eye Margin measurement for Slot1.

```
> EMM? 1,0
< EMM    1000
```

Compatibility

Incompatible with existing models.

EMW?

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 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.

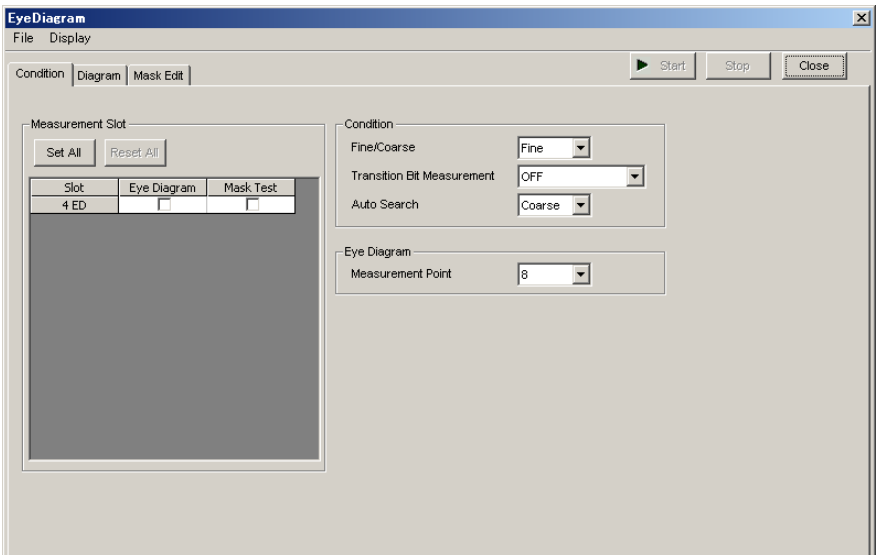


Figure 8.5.2-1 Eye Diagram measurement screen

8.5.2.1 Condition tab setting commands

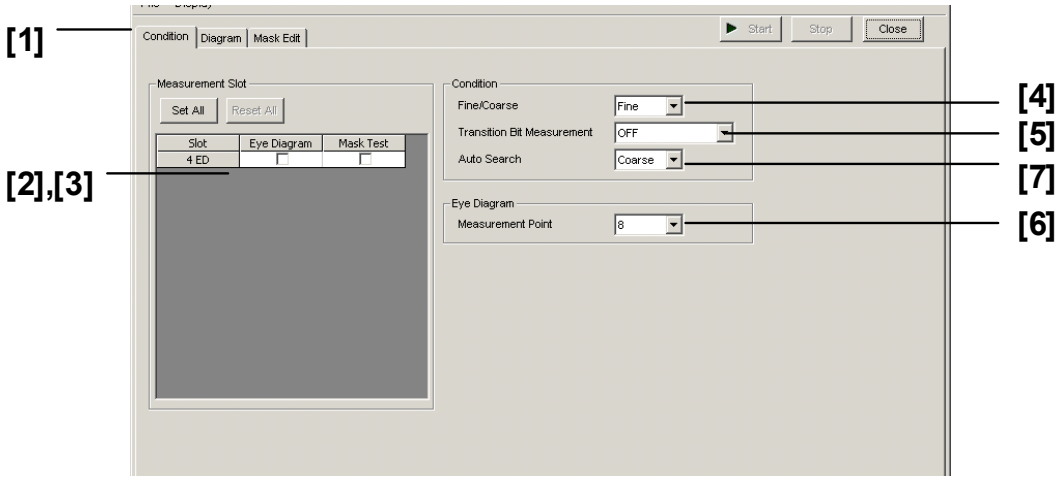


Figure 8.5.2.1-1 Condition tab

Table 8.5.2.1-1 Condition tab setting commands

No.	Setting Items	Commands
[1]	Tab selection	ETB
		ETB?
[2]	Slot Eye Diagram	EDS
		EDS?
[3]	Slot Mask Test	EMS
		EMS?
[4]	Fine/Coarse	EMR
		EMR?
[5]	Transition Bit Measurement	EMT
		EMT?
[6]	Measurement Point	EYP
		EYP?
[7]	Auto Search	EDA
		EDA?

ETB <tab>

Parameter	<tab> = <DECIMAL NUMERIC PROGRAM DATA>
	0 Condition tab
	1 Diagram tab
	2 Mask Edit tab
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?

Response	<tab> = <NR1 NUMERIC RESPONSE DATA> (FIX1) 0 to 2
Function	Queries the tab currently active in the Eye Diagram measurement screen.
Example	`> ETB?` `< ETB 0`
Compatibility	Incompatible with existing models.

EDS <slot>,<numeric>[,<unit>]

Parameter	<slot> = <DECIMAL NUMERIC PROGRAM DATA>
	1 to 6 Slot No. 1 to 6
	<numeric> = <DECIMAL NUMERIC PROGRAM DATA>
	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 Diagram measurement slot.
Example	Eye Diagram measurement To set Slot1 to ON: `> EDS 1,1`
Compatibility	Incompatible with existing models.

EDS? <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	To query whether the specified slot is targeted for the Eye Diagram measurement:
Example	`> EDS? 1` `< EDS 1`
Compatibility	Incompatible with existing models.

EMS <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 DATA> 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 slot for Mask Test measurement.
Example	To set Slot1 for Mask Test measurement to ON: `> EMS 1,1`
Compatibility	Incompatible with existing models.

EMS? <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	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>

Parameter	<type> = <DECIMAL NUMERIC PROGRAM DATA> 0 Coarse mode 1 Fine mode
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

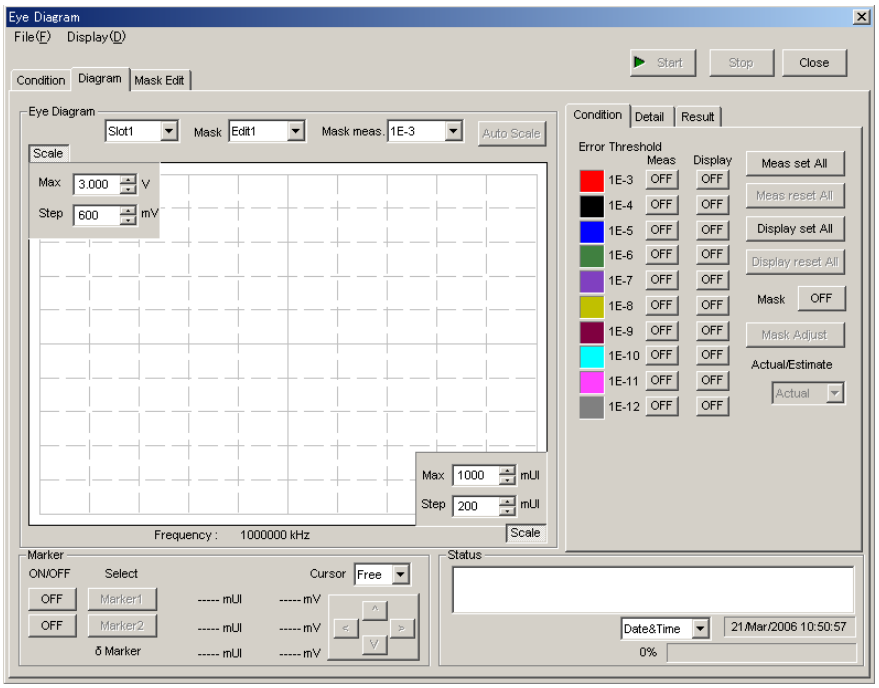


Figure 8.5.2.2-1 Diagram tab

8.5.2.2.1 Graph display setting commands

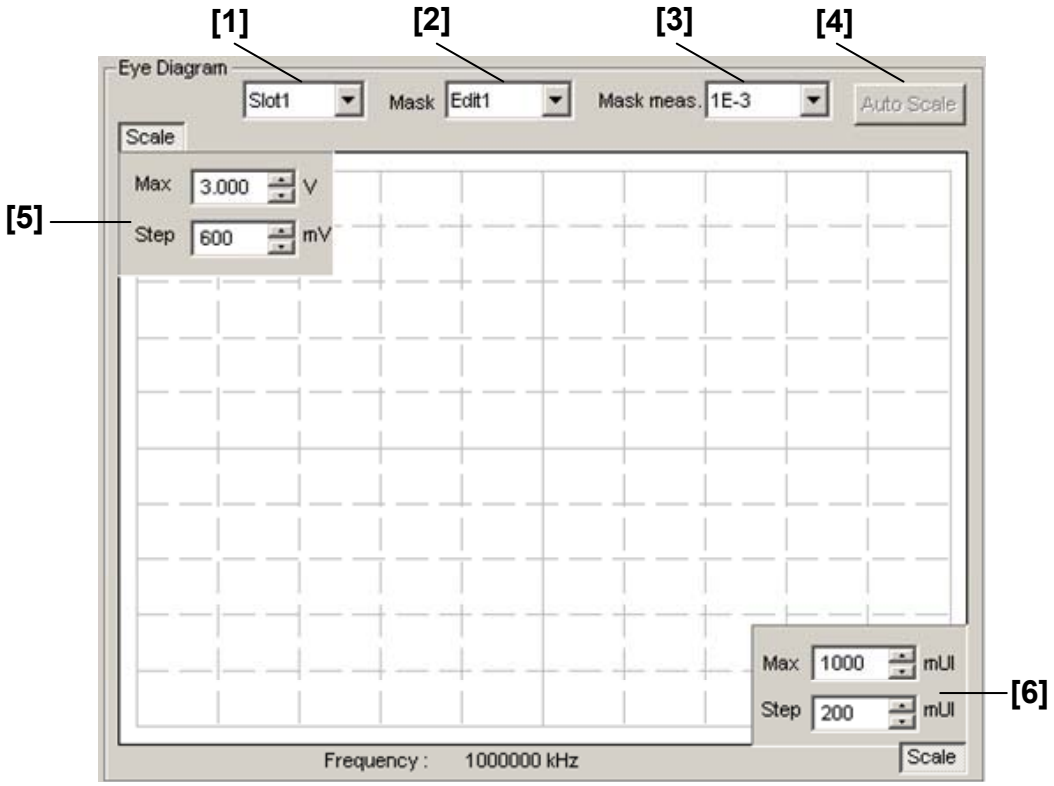


Figure 8.5.2.2.1-1 Eye Diagram field on Diagram tab

Table 8.5.2.2.1-1 Diagram tab setting commands

No.	Setting Items	Commands
[1]	No label (Setting of result display slot)	DPS
		DPS?
[2]	Mask	ETS
		ETS?
[3]	Mask meas	ETT
		ETT?
[4]	Auto Scale	ASC
[5]	Max Step (Threshold)	VSC
		VSC?
[6]	Max Step (Phase)	PSC
		PSC?

DPS <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 Eye Diagram result display target slot.
Example	To set the Eye Diagram result display target slot to Slot1:
	`> DPS 1`
Compatibility	Incompatible with existing models.

DPS? [<unit>]

Parameter	[<unit>] = <DECIMAL NUMERIC PROGRAM DATA>
	1 to 4 Mainframe No. 1 to 4
	1 to 6 when using the MP1800A, 1 to 4 when using the MT1810A.
	Specify the mainframe No. when connected to the MT1810A.
	<unit> is optional. When omitted, 1 is specified.
Response	<slot> = <NR1 NUMERIC RESPONSE DATA> (FIX2)
Function	Queries the Eye Diagram result display target slot.
Example	`> DPS?`
	`< DPS 1`
	`< DPS 64`
Compatibility	Incompatible with existing models.

ETS <type>

Parameter	<type> = <DECIMAL NUMERIC PROGRAM DATA>
	0 Edit1
	1 Edit2
	2 Edit3
	3 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 Edit4:
	`> ETS 3`
Compatibility	Incompatible with existing models.

ETS?

Response	<type> = <NR1 NUMERIC RESPONSE DATA> (FIX1)
Function	Queries the mask pattern for performing the Eye Diagram mask test.
Example	> ETS? < ETS 3
Compatibility	Incompatible with existing models.

ETT <thre>

Parameter	<thre> = <DECIMAL NUMERIC PROGRAM DATA> 3 to 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: > ETT 3
Compatibility	Incompatible with existing models.

ETT?

Response	<thre> = <NR1 NUMERIC RESPONSE DATA> (FIX2)
Function	Queries the Error Threshold for performing the Eye Diagram mask test.
Example	> ETT? < ETT 3 < ETT 12
Compatibility	Incompatible with existing models.

ASC

Function	Automatically adjusts the Eye Diagram measurement graph scale.
Example	> ASC
Compatibility	Incompatible with existing models.

VSC <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> 0.001 to 0.800 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 –3.990 to 4.000 V <step> = <NR2 NUMERIC RESPONSE DATA>(FIX5) 0.001 to 0.800 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 –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 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`
Compatibility	Incompatible with existing models.

PSC?

Response	<min> = <NR1 NUMERIC RESPONSE DATA> (FIX4) –990 to 1000 –990 to 1000 mUI <step> = <NR1 NUMERIC RESPONSE DATA> (FIX3) 1 to 200 1 to 200 mUI
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

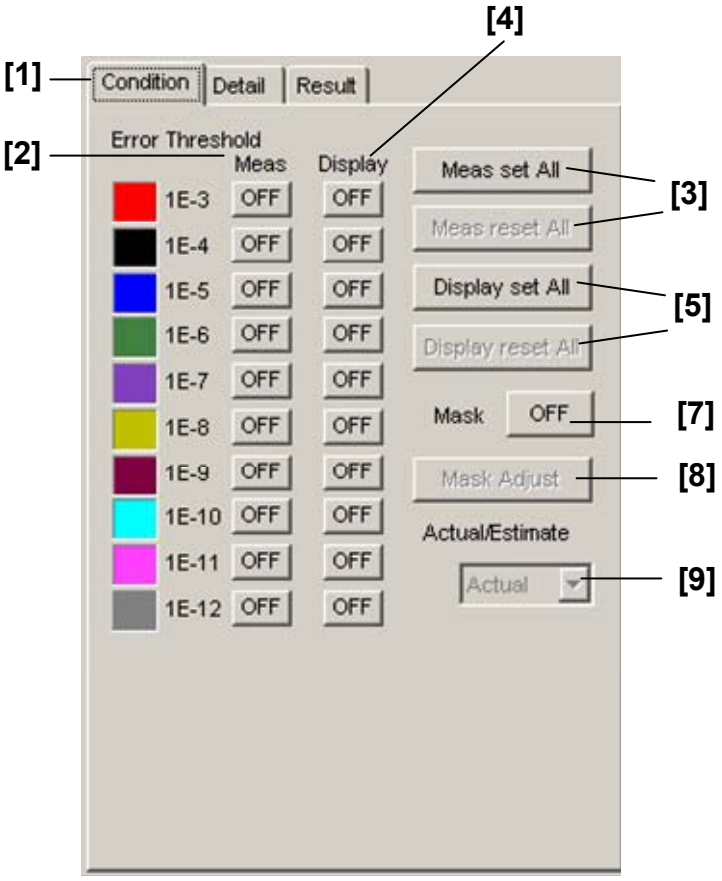


Figure 8.5.2.2.2-1 Condition, Detail and Result tab selection on Diagram tab

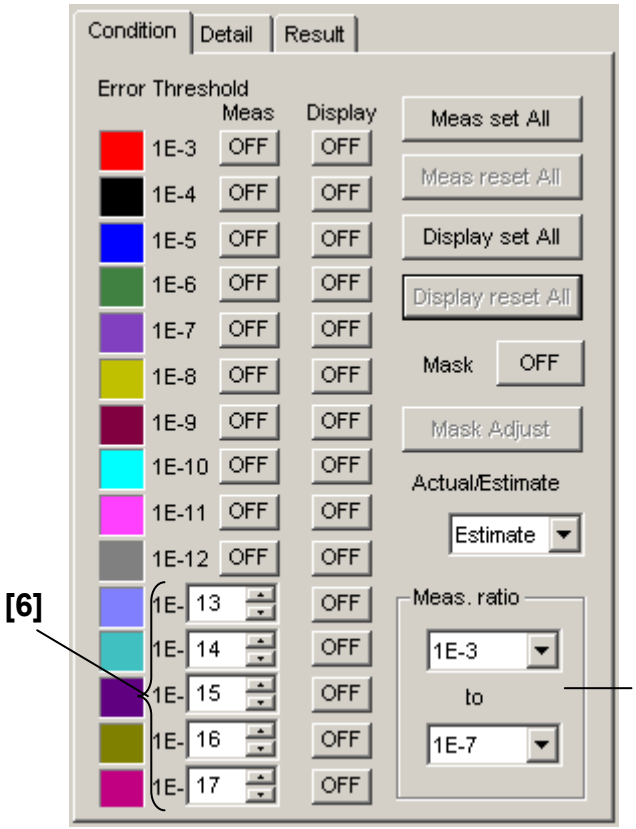


Figure 8.5.2.2.2-2 Condition setting screen (Estimate)

Table 8.5.2.2.2-1 Diagram tab setting commands

No.	Setting Items	Commands
[1]	Select tab	DTB
		DTB?
[2]	Meas	ETR
		ETR?
[3]	Meas set All Meas reset All	DAS
[4]	Display	DER
		DER?
[5]	Display set All Display reset All	DAR
[6]	Setting of error rate for Estimate measurement	DES
		DES?
[7]	Mask ON/OFF	DPM
		DPM?
[8]	Mask Adjust	DPA
[9]	Actual/Estimate	DPE
		DPE?
[10]	Meas.ratio	DBT
		DBT?
[11]	Query for measurement point	EDP?
[12]	Query for measurement result	EDM?

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:

```
> 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

```
> DTB?
< DTB 2
```

Compatibility Incompatible with existing models.

ETR <thre>

Parameter

<thre> = <BINARY NUMERIC PROGRAM DATA>

B1000000000	1E–3
B0100000000	1E–4
B0010000000	1E–5
B0001000000	1E–6
B0000100000	1E–7
B0000010000	1E–8
B0000001000	1E–9
B0000000100	1E–10
B0000000010	1E–11
B0000000001	1E–12

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?

Response

<thre> = <NUMERIC RESPONSE DATA> (FIX11)

Function

Queries the Eye Diagram measurement target error rate.

Example

```
> ETR?
< ETR B1000100001
```

Compatibility

Incompatible with existing models.

DAS <numeric>

Parameter

<numeric> = <DECIMAL NUMERIC PROGRAM DATA>

0	OFF
1	ON

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

<thre> = <BINARY NUMERIC PROGRAM DATA>

B100000000000000	1E–3
B010000000000000	1E–4
B001000000000000	1E–5
B000100000000000	1E–6
B000010000000000	1E–7
B000001000000000	1E–8
B000000100000000	1E–9
B000000010000000	1E–10
B000000001000000	1E–11
B000000000100000	1E–12
B000000000010000	1E–13
B000000000001000	1E–14
B000000000000100	1E–15
B000000000000010	1E–16
B000000000000001	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 E–17 to ON:

```
> DER B000000000100001
```

Compatibility

Incompatible with existing models.

DER?

Response

<thre> = <NUMERIC RESPONSE DATA> (FIX16)

Function

Queries the Eye Diagram measurement result display target.

Example

```
> DER?
< DER B000000000100001
```

Compatibility

Incompatible with existing models.

DAR <numeric>

Parameter

<numeric> = <DECIMAL NUMERIC PROGRAM DATA>

0	OFF
1	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> 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: `>DES 15,100,150,199,180`
Compatibility	Incompatible with existing models.

DES?

Response	<numeric> = <NR1 NUMERIC RESPONSE DATA> (FIX3) (FIX3) (FIX3) (FIX3) (FIX3)
Function	Queries the result display error rate in Estimate measurement from E–13 to E–199.

Example	`> DES?`
	`< DES 15,100,150,199,180`
	`< DES 15, 30, 70, 80, 99`

Compatibility	Incompatible with existing models.

DPM <numeric>

Parameter	<numeric> = <DECIMAL NUMERIC PROGRAM DATA> 1 Display 0 Clear
Function	Selects between displaying and clearing the mask during Eye Diagram measurement.
Example	To display the mask during Eye Diagram measurement: `> DPM 1`
Compatibility	Incompatible with existing models.

DPM?

Response	<numeric> = <NR1 NUMERIC RESPONSE DATA> (FIX1)
Function	Queries the setting whether to display or clear the mask during Eye Diagram measurement.
Example	`> DPM?` `< DPM 1`
Compatibility	Incompatible with existing models.

DPA

Function	Automatically adjusts the mask during Eye Diagram measurement.
Example	`> DPA`
Compatibility	Incompatible with existing models.

DPE <type>

Parameter	<type> = <DECIMAL NUMERIC PROGRAM DATA> 0 Actual measurement 1 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: `> DPE 0`
Compatibility	Incompatible with existing models.

DPE?

Response	<type> = <NR1 NUMERIC RESPONSE DATA> (FIX1)
Function	Queries the Actual/Estimate measurement state during Eye Diagram measurement.
Example	`> DPE?` `< DPE 0`
Compatibility	Incompatible with existing models.

DBT <upper>,<lower>

Parameter	<upper> = <DECIMAL NUMERIC PROGRAM DATA> 3 to 7 1E–3 to 1E–7, in single steps <lower> = <DECIMAL NUMERIC PROGRAM DATA> 7 to 12 1E–7 to E–12, in single steps ***Note:*** The difference between the <upper> and <lower> value must be 3 or greater.
Function	Sets the reference error rate during Eye Diagram Estimate.
Example	To set the reference error rate during Eye Diagram Estimate to E–3 or E–12: `> DBT 3,12`
Compatibility	Incompatible with existing models.

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 Slot No. 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 1E–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

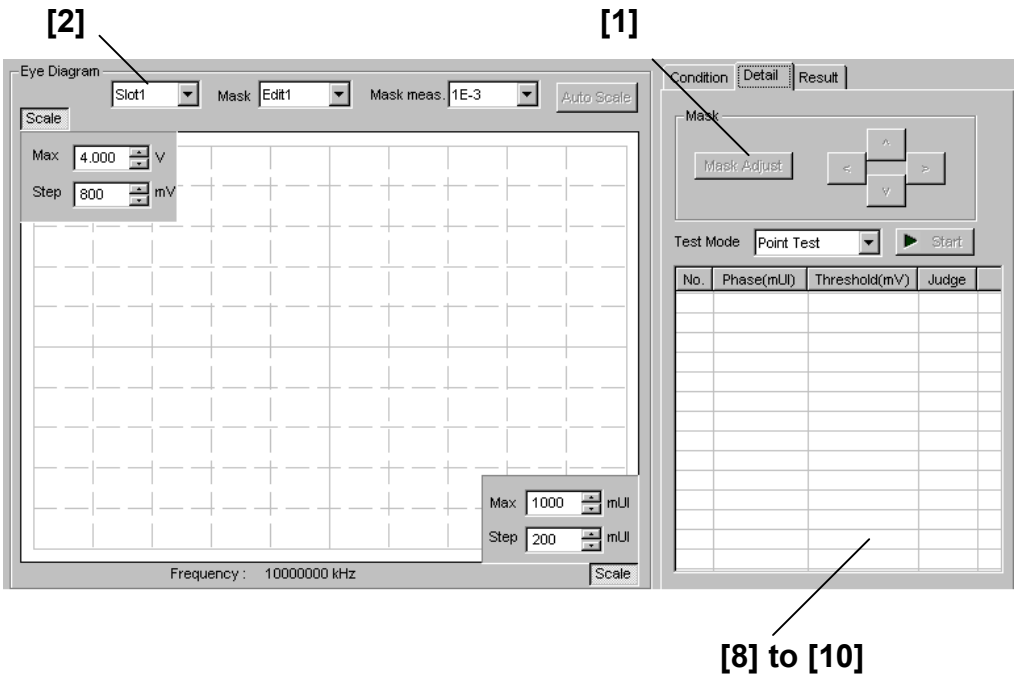


Figure 8.5.2.2.3-1 Detail Mask Point Test screen

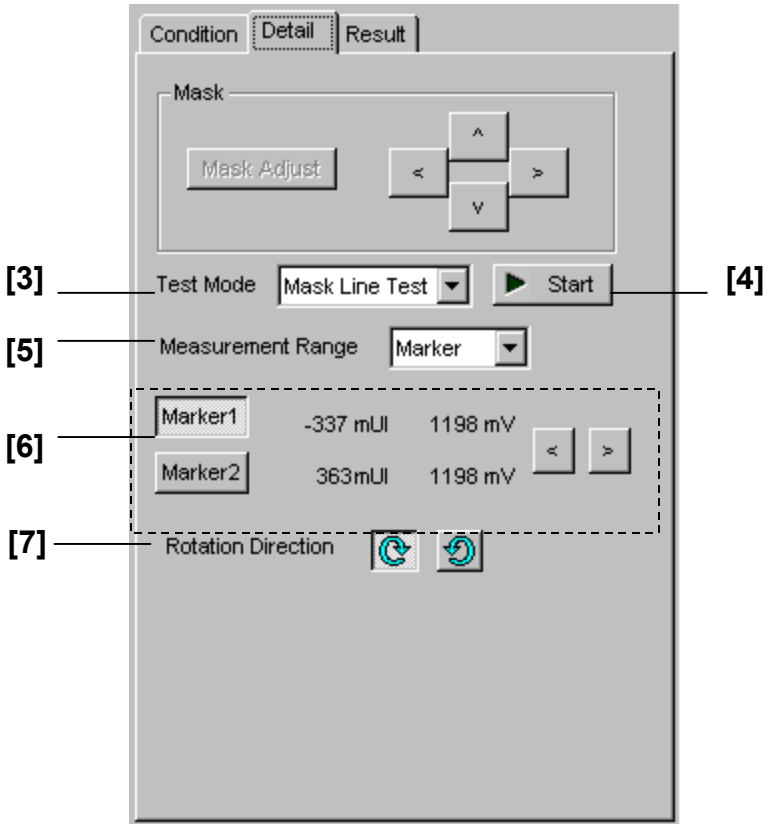


Figure 8.5.2.2.3-2 Detail Mask Line Test screen

Table 8.5.2.2.3-1 Detail setting commands

No.	Setting Items	Commands
[1]	Mask Adjust	DMK
[2]	Slot selection	DTS
		DTS?
[3]	Test Mode	DMO
		DMO?
[4]	Start/Stop	DST
		DSP
		DSA?
[5]	Measurement Range	DLR
		DLR?
[6]	Marker1 Marker2	DMR
		DMR?
[7]	Rotation Direction	DLT
		DLT?
[8]	Query for measurement point	DPN?
[9]	Query for Point test result	DDP?
[10]	Query for Line test result	DDL?

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>]

Parameter	[<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	<slot> = <NR1 NUMERIC RESPONSE DATA> (FIX2)
Function	Queries the Mask pass/fail measurement target slot during Eye Diagram measurement.
Example	> DTS? < DTS 1
Compatibility	Incompatible with existing models.

DMO <mode>

Parameter	<mode> = <DECIMAL NUMERIC PROGRAM DATA> 0 Point Test 1 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: > DMO 0
Compatibility	Incompatible with existing models.

DMO?

Response	<mode> = <NR1 NUMERIC RESPONSE DATA> (FIX1)
Function	Queries the measurement mode during Eye Diagram Detail measurement.
Example	> DMO? < DMO 0
Compatibility	Incompatible with existing models.

DST

Function	Starts the Mask pass/fail judgment during Eye Diagram Detail measurement.
Example	> DST
Compatibility	Incompatible with existing models.

DSP

Function	Stops the Mask pass/fail judgment during Eye Diagram Detail measurement.
Example	`> DSP`
Compatibility	Incompatible with existing models.

DSA?

Response	<numeric> = <NR1 NUMERIC RESPONSE DATA> (FIX1) 1 Start 0 Stop
Function	Queries the Mask pass/fail judgment status during Eye Diagram Detail measurement.
Example	`> DSA?` `< DSA 1`
Compatibility	Incompatible with existing models.

DLR <range>

Parameter	<range> = <DECIMAL NUMERIC PROGRAM DATA> 0 ALL 1 Marker
Function	Selects the measurement range during Eye Diagram Detail Line measurement.
Example	To set the measurement range during Eye Diagram Detail Line measurement to Marker: `> DLR 1`
Compatibility	Incompatible with existing models.

DLR?

Response	<range> = <NR1 NUMERIC RESPONSE DATA> (FIX1)
Function	Queries the measurement range during Eye Diagram Detail Line measurement.
Example	`> DLR?` `< DLR 1`
Compatibility	Incompatible with existing models.

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)(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.

DLT <type>

Parameter	<type> = <DECIMAL NUMERIC PROGRAM DATA> 0 Counterclockwise 1 Clockwise
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?

Response	<type> = <NR1 NUMERIC RESPONSE DATA> (FIX1)
Function	Queries the Mask pass/fail judgment direction during Eye Diagram Detail Line measurement.
Example	`> DLT?` `< DLT 1`
Compatibility	Incompatible with existing models.

DPN?

Response	<numeric> = <NR1 NUMERIC RESPONSE DATA> (FIX5) 1 to 20000 Detail measurement point
Function	Queries the number of points during Eye Diagram Detail measurement.

Example	`> DPN?`
	`< DPN 1000`
	`< DPN 20000`

Compatibility	Incompatible with existing models.

DDP? <point>

Parameter

<point> = <DECIMAL NUMERIC PROGRAM DATA>

1 to 32	Measurement point

Response

<numeric1> = <NR1 NUMERIC RESPONSE DATA> (FIX5)

–1000 to 1000	Phase data (mUI)
XXXXX	Phase data (ps)

Note:

Displays in the selected unit.

<numeric2> = <NR1 NUMERIC RESPONSE DATA> (FIX5)

–4000 to 4000	Threshold data (mV)

<numeric3> = <NR1 NUMERIC RESPONSE DATA> (FIX1)

0	OK
1	NG

Function

Queries the measurement result of the Eye Diagram Detail Point Test.

Example

To query the measurement result of the Eye Diagram Detail Point Test at measurement point 1:

```
> DDP? 1
< DDP -1000,-4000,0
< DDP    0,    0,0
< DDP  1000, 4000,1
```

Compatibility

Incompatible with existing models.

DDL? <point>

Parameter

<point> = <DECIMAL NUMERIC PROGRAM DATA>

1 to 20000 Measurement point

Response

<numeric1> = <NR1 NUMERIC RESPONSE DATA> (FIX5)

–1000 to 1000 Phase data (mUI)

XXXXX Phase data (ps)

Note:

Displays in the selected unit.

When no data exists, "------" is displayed.

<numeric2> = <NR1 NUMERIC RESPONSE DATA> (FIX5)

–4000 to 4000 Threshold data (mV)

When no data exists, "------" is displayed.

<numeric3> = <NR1 NUMERIC RESPONSE DATA> (FIX1)

0 OK

1 NG

Function

Queries the measurement result of the Eye Diagram Detail Mask Line Test.

Example

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
```

Compatibility

Incompatible with existing models.

8.5.2.2.4 Result query commands

Figure 8.5.2.2.4-1 Result tab

Table 8.5.2.2.4-1 Result query command

No.	Setting Items	Command
[1]	Query for Mask Test result	MSK?

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

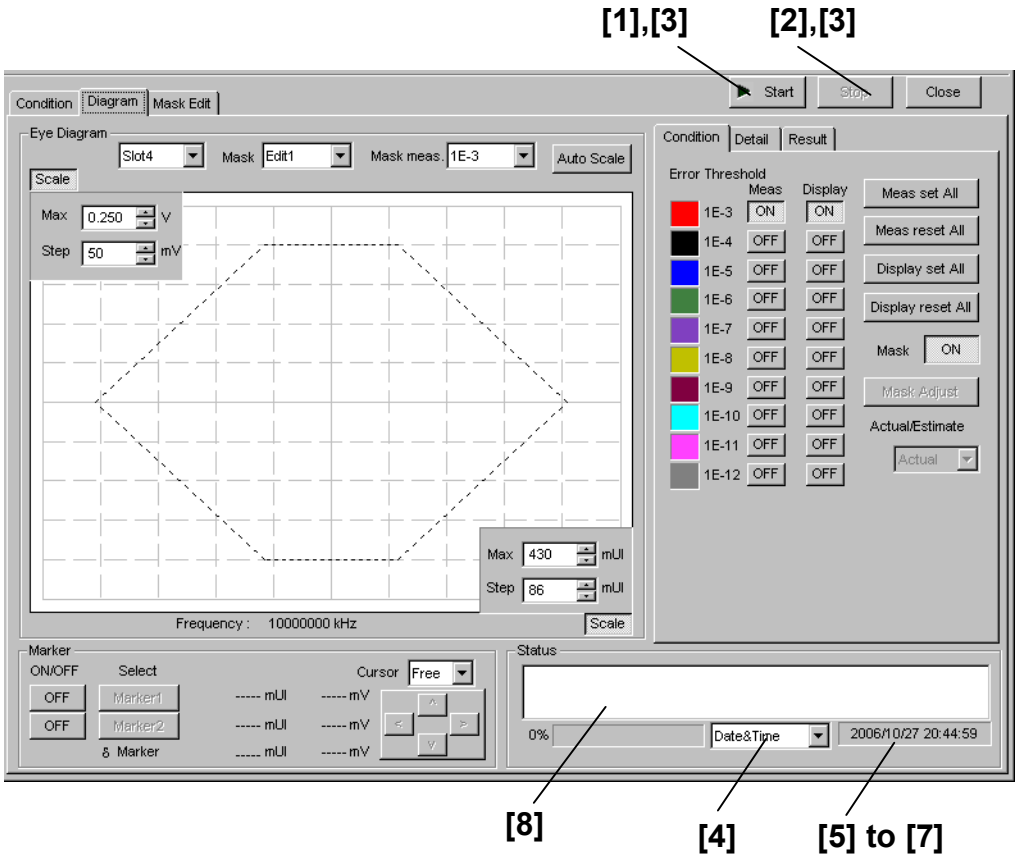


Figure 8.5.2.2.5-1 Diagram tab Start/Stop buttons and Status field

Table 8.5.2.2.5-1 Diagram tab (Status) setting commands

No.	Setting Items	Commands
[1]	Start	DTA
[2]	Stop	DTO
[3]	Query for measurement state	MSR?
[4]	No label (Setting of time display)	ETM
		ETM?
[5]	Query for current time	EDT?
[6]	Query for measurement start time	MSA?
[7]	Query for measurement elapsed time	ELP?
[8]	Query for measurement status	EDW?

DTA

Function	Starts the Eye Diagram measurement.
Example	`> DTA`
Compatibility	Incompatible with existing models.

DTO

Function	Stops the Eye Diagram measurement.
Example	`> DTO`
Compatibility	Incompatible with existing models.

MSR?

Response	<numeric> = <NR1 NUMERIC RESPONSE DATA> (FIX1)
	1 During measurement
	0 Measurement has been stopped.
	2 Measurement has failed.
Function	Queries the Eye Diagram measurement status.
Example	`> MSR?`
	`< MSR 1`
Compatibility	Incompatible with existing models.

ETM <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 Diagram measurement time display type.
Example	To set the Eye Diagram measurement time display type to measurement start time (Start Time):
	`> ETM 1`
Compatibility	Incompatible with existing models.

ETM?

Response	<type> = <NR1 NUMERIC RESPONSE DATA> (FIX1)
Function	Queries the Eye Diagram measurement time display type.
Example	`> ETM?`
	`< ETM 1`
Compatibility	Incompatible with existing models.

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.

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.

8.5.2.2.6 Mask Edit tab setting commands

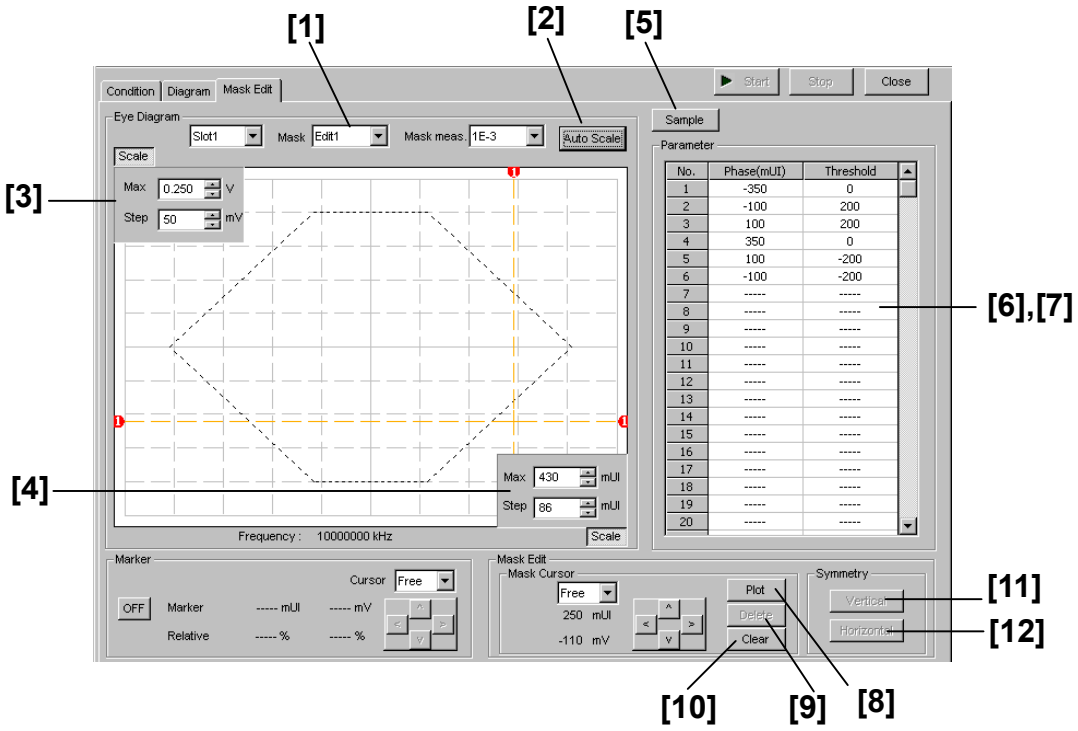


Figure 8.5.2.2.6-1 Mask Edit tab

Table 8.5.2.2.6-1 Mask Edit tab setting commands

No.	Setting Items	Commands
[1]	Mask	TTP
		TTP?
[2]	Auto Scale	MAS
[3]	Max Step (Threshold)	MVS
		MVS?
[4]	Max Step (Phase)	MPH
		MPH?
[5]	Sample	SAM
[6]	Query for mask point count	TPN?
[7]	Phase (mUI) Threshold	TPV
		TPV?
[8]	Plot	TPL
[9]	Delete	TDL
[10]	Clear	TCL
[11]	Vertical	TVE
[12]	Horizontal	THO

TTP <type>

Parameter	<type> = <DECIMAL NUMERIC PROGRAM DATA>
	0 Edit1
	1 Edit2
	2 Edit3
	3 Edit4
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?

Response	<type> = <NR1 NUMERIC RESPONSE DATA> (FIX1)
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.

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.

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.

TPV <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 (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 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 –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): `> 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.

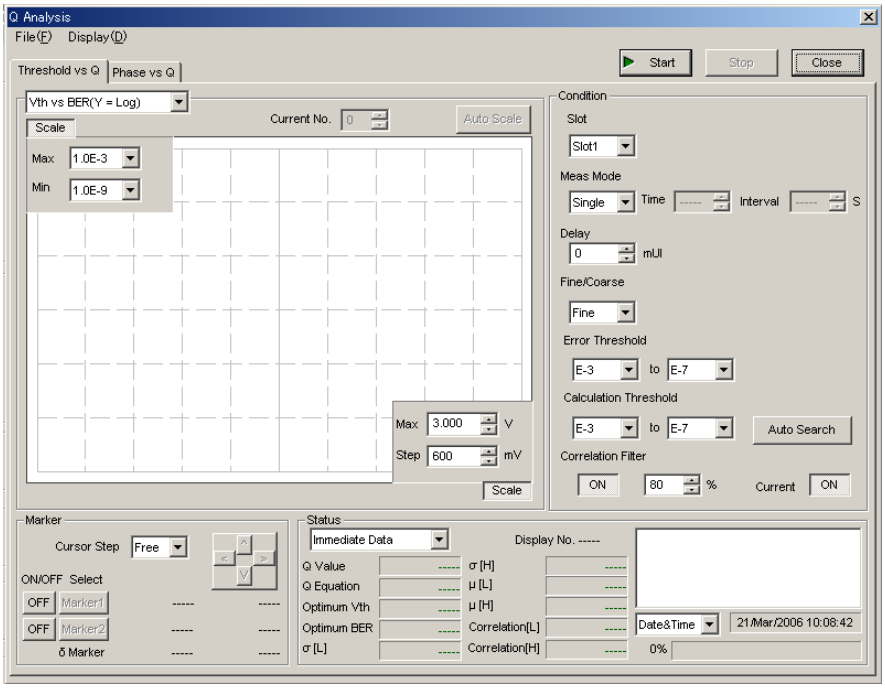


Figure 8.5.3-1 Q measurement screen

8.5.3.1 Graph display setting commands

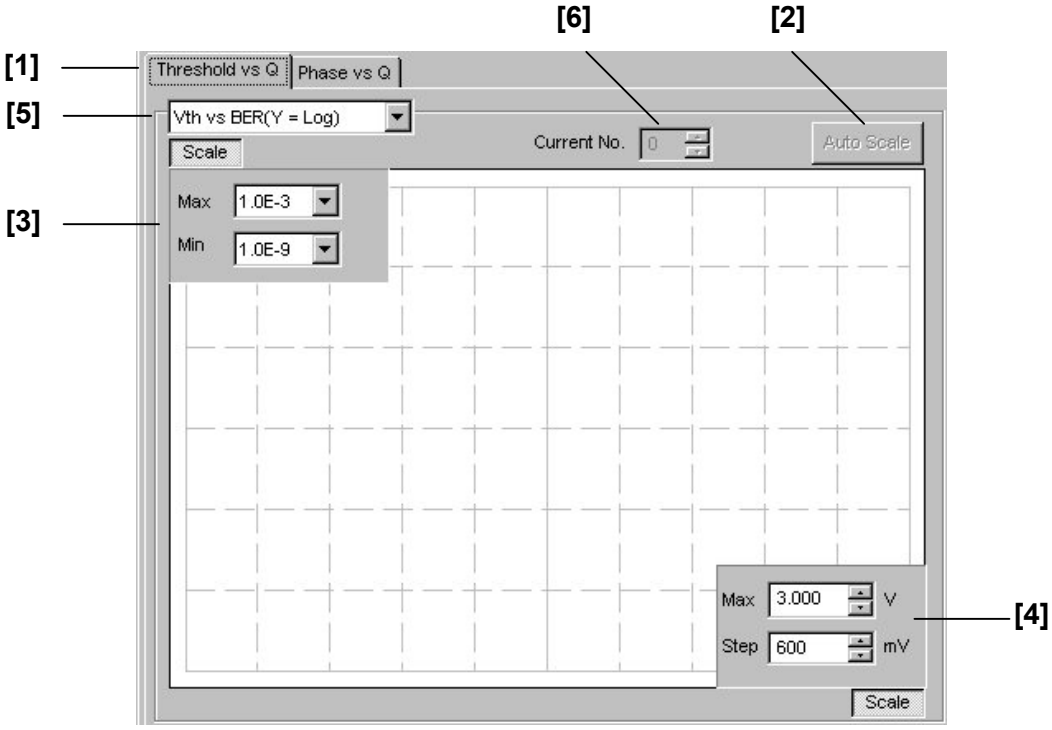


Figure 8.5.3.1-1 Graph display screen (Threshold vs Q tab)

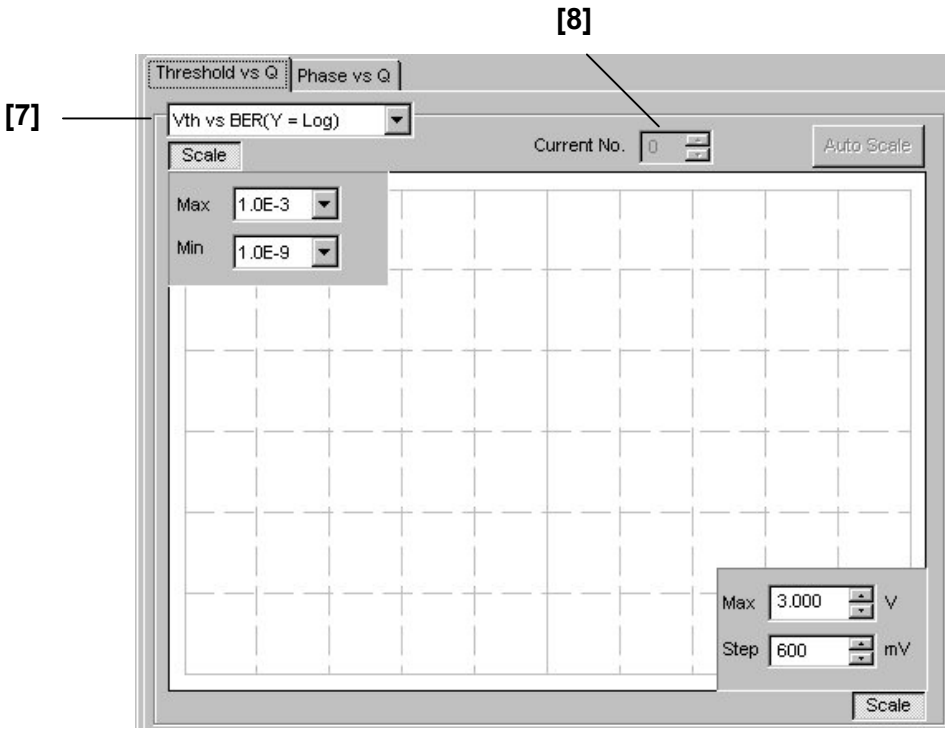


Figure 8.5.3.1-2 Graph display screen (Phase vs Q tab)

Table 8.5.3.1-1 Graph display setting commands

No.	Setting Items	Commands
[1]	Measurement mode switching	QMO
		QMO?
[2]	Auto Scale	QSC
[3]	Max	QSV
	Min	QSV?
[4]	Max	QSH
	Step	QSH?
[5]	Threshold vs. Q	QTI
	Display graph selection	QTI?
[6]	Threshold vs. Q	QMN
	Current No.	QMN?
[7]	Phase vs. Q	QPI
	Display graph selection	QPI?
[8]	Phase vs. Q	QPM
	Current No.	QPM?

QMO <mode>

Parameter <mode> = <DECIMAL NUMERIC PROGRAM DATA>
0 Vth vs. Q measurement
1 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
```
> QMO?
< QMO 1
```

Compatibility Incompatible with existing models.

QSC

Function Automatically adjusts the graph display during Q measurement.

Example
```
> QSC
```

Compatibility Incompatible with existing models.

QSV <max>[,<step>]

Parameter <max> [,<step>] = <DECIMAL NUMERIC PROGRAM DATA>

Table 8.5.3.1-2 Parameters for Threshold vs. Q tab

Format	<max>	<step>	Remarks
Vth vs BER(Y = Log)	2 to 7 (E–2 to E–7), in single steps	Inputs the minimum value. 8 to 14 (E–8 to E–14), in single steps	A difference between <max> and <step> must be 8 or greater even number.
Vth vs BER(Y = Log(–Ln))	2 to 7 (E–2 to E–7), in single steps	Inputs the minimum value. 8 to 14 (E–8 to E–14), in single steps	A difference between <max> and <step> must be 8 or greater even number.
Vth vs Q	10 to 1000, in 10 steps	1 to 100, in single steps	When Linear
	–40 to 60, in 10 steps	1 to 11, in single steps	When Log
Times vs Q	10 to 1000, in 10 steps	1 to 100, in single steps	When Linear
	–40 to 60, in 10 steps	1 to 11, in single steps	When Log
Histogram	50 to 1000, in 50 steps	Omitted	

Table 8.5.3.1-3 Parameters for Phase vs. Q tab

Format	<max>	<step>	Remarks
Vth vs BER(Y = Log)	2 to 7 (E–2 to E–7), in single steps	Inputs the minimum value. 8 to 14 (E–8 to E–14), in single steps	A difference between <max> and <step> must be 8 or greater even number.
Vth vs BER(Y = Log(–Ln))	2 to 7 (E–2 to E–7), in single steps	Inputs the minimum value. 8 to 14 (E–8 to E–14), in single steps	A difference between <max> and <step> must be 8 or greater even number.
Vth vs Q	10 to 1000, in 10 steps	1 to 100, in single steps	When Linear
	–40 to 60, in 10 steps	1 to 11, in single steps	When Log
Phase vs Q	10 to 1000, in 10 steps	1 to 100, in single steps	When Linear
	–40 to 60, in 10 steps	1 to 11, in single steps	When Log
Phase vs σ	0.0010 to 1.0000, in 0.0010 steps	0.001 to 0.1000, 0.0001 steps	
Phase vs μ	–3.990 to 4.000 (V), in 0.010 V steps	0.001 to 0.800 (V), in 0.001 V steps	
Phase vs Opt BER	10 to 1 (E–10 to E–1), in single steps	Inputs the minimum value. 199 to 20 (E–199 to E–20), in single steps	
Phase vs Opt Threshold	–3.990 to 4.000 (V), in 0.010 V steps	0.001 to 0.800 (V), in 0.001 V steps	
Phase vs Correlation	10 to 100 (%), in 10 % steps	0 to 90 (%), in 10 % steps	A difference between <max> and <step> must be 10 or greater

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. Input the minimum value instead of the step width, only when the vertical axis displays the error rate.

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?

Response — <max>,<step> = <NR2 NUMERIC RESPONSE DATA>

Function — Queries the vertical scale of the graph display during Q measurement.

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

Format	<max>	<step>	Remarks
Vth vs BER(Y = Log)	–3.990 to 4.000 (V), in 0.010 V steps	0.001 to 0.800 (V), in 0.001 V steps	
Vth vs BER(Y = Log(–Ln))	–3.990 to 4.000 (V), in 0.010 V steps	0.001 to 0.800 (V), in 0.001 V steps	
Vth vs Q	–3.990 to 4.000 (V), in 0.010 V steps	0.001 to 0.800 (V), in 0.001 V steps	
Times vs Q	100 to 1000, in 50 steps	10 (fixed)	
Histogram	0.16 to 1000.00, in 0.01 steps	16/32/64/128/256	When Linear <max> must be greater than <step>.
	–49.84 to 60.00 (dB), in 0.01 steps	16/32/64	When Log

Table 8.5.3.1-5 Parameters for Phase vs. Q tab

Format	<max>	<step>	Remarks
Vth vs BER(Y = Log)	–3.990 to 4.000 (V), in 0.010 V steps	0.001 to 0.800 (V), in 0.001 V steps	
Vth vs BER(Y = Log(–Ln))	–3.990 to 4.000 (V), in 0.010 V steps	0.001 to 0.800 (V), in 0.001 V steps	
Vth vs Q	–3.990 to 4.000 (V), in 0.010 V steps	0.001 to 0.800 (V), in 0.001 V steps	
Phase vs Q	–900 to 1000 (mUI), in 100 mUI steps	10 to 200 (mUI), in 10 mUI steps	
Phase vs σ	–900 to 1000 (mUI), in 100 mUI steps	10 to 200 (mUI), in 10 mUI steps	
Phase vs μ	–900 to 1000 (mUI), in 100 mUI steps	10 to 200 (mUI), in 10 mUI steps	
Phase vs Opt BER	–900 to 1000 (mUI), in 100 mUI steps	10 to 200 (mUI), in 10 mUI steps	
Phase vs Opt Threshold	–900 to 1000 (mUI), in 100 mUI steps	10 to 200 (mUI), in 10 mUI steps	
Phase vs Opt Correlation	–900 to 1000 (mUI), in 100 mUI steps	10 to 200 (mUI), in 10 mUI steps	

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

0.100-steps:
```
> QSH -1.000,0.100
```

Compatibility: Incompatible with existing models.

QSH?

Response: <max>,<step> = <NR2 NUMERIC RESPONSE DATA>

Function: Queries the horizontal scale of the graph display during Q measurement.

Example:
```
> QSH?
< QSH -1.000,0.100
```

Compatibility: Incompatible with existing models.

QTI <mode>

Parameter: <mode> = <DECIMAL NUMERIC PROGRAM DATA>

0	Vth vs. BER (Y axis Log)
1	Vth vs. BER (Y axis Log (–Ln))
2	Vth vs. Q
3	Times vs. Q
4	Histogram

Function: Sets the item to be displayed in the graph in Vth vs. Q measurement.

Example:
```
> QTI 2
```

Compatibility: Incompatible with existing models.

QTI?

Response: <mode> = <NR1 NUMERIC RESPONSE DATA> (FIX1)

Function: Queries the item to be displayed in the graph in Vth vs. Q measurement.

Example:
```
> QTI?
< QTI 2
```

Compatibility: Incompatible with existing models.

QMN <numeric>

Parameter	<numeric> = <DECIMAL NUMERIC 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 2: `> QMN 2`
Compatibility	Incompatible with existing models.

QMN?

Response	<numeric> = <NR1 NUMERIC RESPNSE DATA> (FIX4)
Function	Queries the measurement number for which the Vth vs. Q measurement result is displayed in a graph.

Example	`> QMN?`
	`< QMN 2`
	`< QMN 1000`

Compatibility	Incompatible with existing models.

QPI <mode>

Parameter	<mode> = <DECIMAL NUMERIC PROGRAM DATA> 0 Vth vs. BER (Y axis Log) 1 Vth vs. BER (Y axis Log (–Ln)) 2 Vth vs. Q 3 Phase vs. Q 4 Phase vs. σ 5 Phase vs. μ 6 Phase vs. Optimum BER 7 Phase vs. Optimum Vth 8 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: `> QPI 3`
Compatibility	Incompatible with existing models.

QPI?

Response	<mode> = <NR1 NUMERIC RESPONSE DATA> (FIX1)

Example	Queries the item to be displayed in the graph during Phase vs. Q measurement.
	`> QPI?`
	`< QPI 3`

Compatibility	Incompatible with existing models.

QPM <numeric>

Parameter	<numeric> = <DECIMAL NUMERIC PROGRAM DATA> 1 to 1000 Measurement number (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 2: `> QPM 2`
Compatibility	Incompatible with existing models.

QPM?

Response	<numeric> = <NR1 NUMERIC RESPNSE DATA> (FIX4)
Function	Queries the measurement number for which the Phase vs. Q measurement result is displayed in a graph.

Example	`> QPM?`
	`< QPM 2`
	`< QPM 1000`

Compatibility	Incompatible with existing models.

8.5.3.2 Vth vs. Q measurement commands

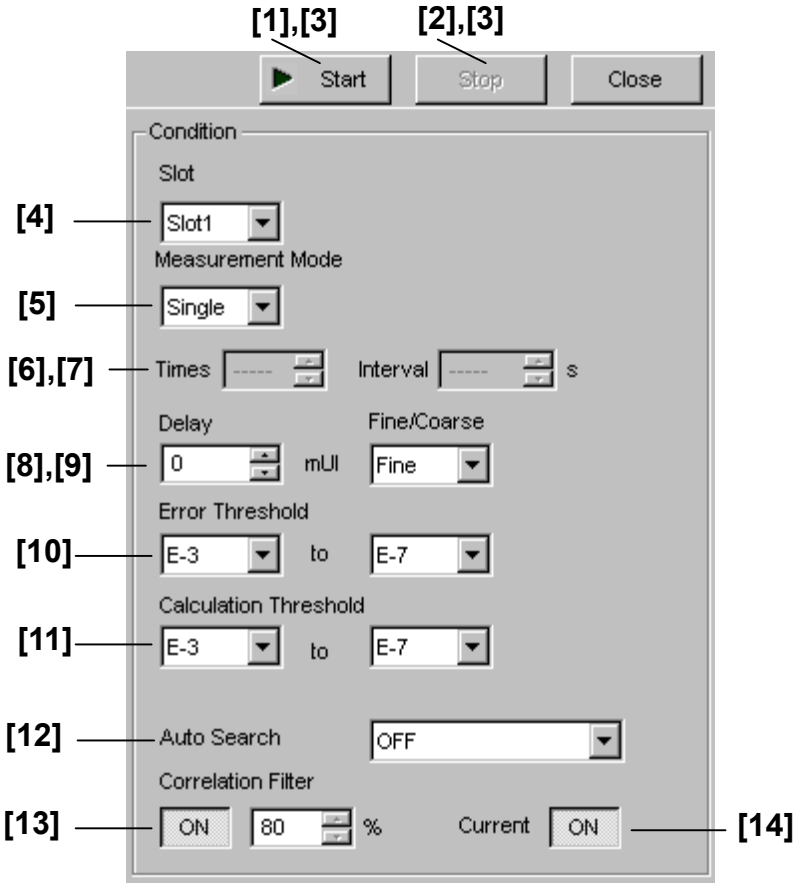


Figure 8.5.3.2-1 Vth vs. Q measurement setting field

Table 8.5.3.2-1 Vth vs. Q measurement setting commands

No.	Setting Items	Commands
[1]	Start	QTT
[2]	Stop	QTP
[3]	Query for measurement progress state	QAS?
[4]	Slot	QSL
		QSL?
[5]	Meas Mode	QTO
		QTO?
[6]	Time	QTM
		QTM?
[7]	Interval	QTN
		QTN?
[8]	Delay	QTD
		QTD?
[9]	Fine/Coarse	QTR
		QTR?
[10]	Error Threshold	QTE
		QTE?
[11]	Calculation Threshold	QTC
		QTC?
[12]	Auto Search	QTA
		QTA?
[13]	Correlation Filter	QTF
		QTF?
[14]	Current	QTU
		QTU?

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.

QAS?

Response	<numeric> = <NR1 NUMERIC RESPONSE DATA> (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	<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.
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>] = <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 RESPNSE DATA> (FIX2)
Function	Queries the Vth vs. Q measurement target slot.
Example	> QSL?
	< QSL 2
Compatibility	Incompatible with existing models.

QTO <mode>

Parameter	<mode> = <DECIMAL NUMERIC PROGRAM DATA>
	0 Single
	1 Repeat
	2 Untimed
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?

Response	<mode> = <NR1 NUMERIC RESPONSE DATA> (FIX1)
Function	Queries the measurement processing mode during Vth vs. Q measurement.
Example	`> QTO?`
	`< QTO 0`
Compatibility	Incompatible with existing models.

QTM <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 Vth vs. Q measurement.
Example	To set the number of measurements to 100 times:
	`> QTM 100`
Compatibility	Incompatible with existing models.

QTM?

Response	<NUMERIC> = <NR1 NUMERIC RESPONSE DATA> (FIX4)
Function	Queries the number of measurements during Vth vs. Q measurement.
Example	`> QTM?`
	`< QTM 2`
	`< QTM 1000`
Compatibility	Incompatible with existing models.

QTN <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 s: `> QTN 50`
Compatibility	Incompatible with existing models.

QTN?

Response	<NUMERIC> = <NR1 NUMERIC RESPONSE DATA> (FIX4)
Function	Queries the measurement interval time during Vth vs. Q measurement.

Example	`> QTN?`
	`< QTN 50`
	`< QTN 9999`

Compatibility	Incompatible with existing models.

QTD <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: `> QTD -100`
Compatibility	Incompatible with existing models.

QTD?

Response	<numeric> = <NR1 NUMERIC RESPONSE DATA> (FIX5)
Function	Queries the measurement start delay position during Vth vs. Q measurement.

Example	`> QTD?`
	`< QTD -1000`
	`< QTD -100`
	`< QTD 1000`

Compatibility	Incompatible with existing models.

QTR <mode>

Parameter	<mode> = <DECIMAL NUMERIC PROGRAM DATA>
	0 Fine mode
	1 Coarse mode
Function	Sets the measurement resolution (Fine/Coarse mode) during Vth vs. Q measurement.
Example	To set the measurement resolution to Fine mode: `> 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	`> QTR?` `< QTR 0`
Compatibility	Incompatible with existing models.

QTE <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 measurement error threshold range during Vth vs. Q measurement.
Example	To set the measurement error threshold range to E–3 to E–12: `> 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	`> 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 hase & 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.

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

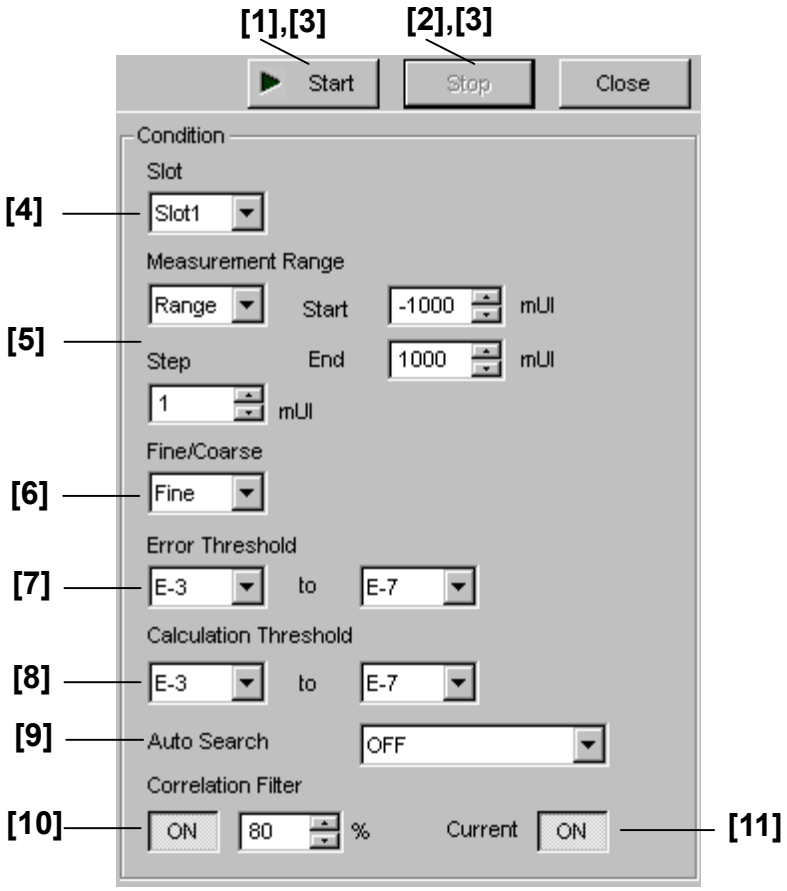


Figure 8.5.3.3-1 Phase vs. Q measurement setting field

Table 8.5.3.3-1 Phase vs. Q measurement setting commands

No.	Setting Items	Commands
[1]	Start	QPT
[2]	Stop	QPP
[3]	Query for measurement progress state	QAS?
[4]	Slot	QSP
		QSP?
[5]	Measurement Range	QMR
		QMR?
	Start End	QPO
		QPO?
	Center Span	QPW
		QPW?
[6]	Fine/Coarse	QPR
		QPR?
[7]	Error Threshold	QPE
		QPE?
[8]	Calculation Threshold	QPC
		QPC?
[9]	Auto Search	QPA
		QPA?
[10]	Correlation Filter	QPF
		QPF?
[11]	Current	QPU
		QPU?

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.

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.
	<unit> 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.
	<unit> is optional. When omitted, 1 is specified.
Response	<numeric> = <NR1 NUMERIC RESPNSE DATA> (FIX2)
Function	Queries the Phase vs. Q measurement target slot.
Example	> QSP?
	< QSP 2
	< QSP 64
Compatibility	Incompatible with existing models.

QMR <mode>

Parameter	<mode> = <DECIMAL NUMERIC PROGRAM DATA>	
	0	Range: Specifies the start position and end positions of the measurement range.
	1	Width: Specifies the center position and width of the measurement range.
Function	Selects the method to specify the Phase vs. Q measurement range.	
Example	To set the method to specify the measurement range to Width: `> QMR 1`	
Compatibility	Incompatible with existing models.	

QMR?

Response	<mode> = < NR1 NUMERIC RESPONSE DATA> (FIX1)
Function	Queries the method to specify the Phase vs. Q measurement range.
Example	`> QMR?` `< QMR 1`
Compatibility	Incompatible with existing models.

QPO <start>,<end>,<step>

Parameter	<start> = <DECIMAL NUMERIC PROGRAM DATA>	
	–1000 to 1000	–1000 to 1000 mUI, in 1-mUI steps
	<end> = <DECIMALNUMERICPROGRAMDATA>	
	–1000 to 1000	–1000 to 1000 mUI, in 1-mUI steps
	<step> = <DECIMALNUMERICPROGRAMDATA>	
	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 from –200 to 300 mUI, in 10-mUI steps: `> 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 ,,<step>

Parameter	= <DECIMAL NUMERIC PROGRAM DATA> –1000 to 1000 –1000 to 1000 mUI, in 1-mUI steps = <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	= <NR1 NUMERIC RESPONSE DATA> (FIX5) = <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.

QPR <mode>

Parameter	<mode> = <DECIMAL NUMERIC PROGRAM DATA>
	0 Fine mode
	1 Coarse mode
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?

Response	<mode> = <NR1 NUMERIC RESPONSE DATA> (FIX1)
Function	Queries the measurement resolution (Fine/Coarse mode) during Phase vs.Q measurement.
Example	`> QPR?` `< QPR 0`
Compatibility	Incompatible with existing models.

QPE <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 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?

Response	<upper> = <NR1 NUMERIC RESPONSE DATA> (FIX1) <lower> = <NR1 NUMERIC RESPONSE DATA> (FIX2)
Function	Queries the measurement error threshold range during Phase vs. Q measurement.

Example	`> QPE?`
	`< QPE 3, 7`
	`< QPE 5,12`

Compatibility	Incompatible with existing models.

QPC <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 Phase vs. Q measurement recalculation.
Example	To set the error threshold range during recalculation to E–5 to E–10: `> QPC 5,10`
Compatibility	Incompatible with existing models.

QPC?

Response	<upper> = <NR1 NUMERIC RESPONSE DATA> (FIX1) <lower> = <NR1 NUMERIC RESPONSE DATA> (FIX2)
Function	Queries the error threshold range during Phase vs. Q measurement recalculation.

Example	`> QPC?`
	`< QPC 3, 7`
	`< QPC 5,10`

Compatibility	Incompatible with existing models.

QPA <numeric>

Parameter	<numeric> = <DECIMAL NUMERIC PROGRAM DATA>
	0 Auto Search OFF
	1 Threshold auto search (Coarse)
	2 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: `> QPA 1`
Compatibility	Incompatible with existing models.

QPA?

Response	<numeric> = <NR1 NUMERIC RESPONSE DATA> (FIX1)
Function	Queries whether to execute auto search during Phase vs. Q measurement.
Example	`> QPA?` `< QPA 1`
Compatibility	Incompatible with existing models.

QPF <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 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?

Response	<numeric1> = <NR2 NUMERIC RESPONSE DATA> (FIX1) <numeric2> = <NR2 NUEMRIC RESPONSE DATA> (FIX3)
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>

Parameter	<numeric> = <DECIMAL NUMERIC PROGRAM DATA>
	0 OFF: Updates at measurement termination
	1 ON: Updates every second
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?

Response	<numeric> = <NR1 NUMERIC RESPONSE DATA> (FIX1)
Function	Queries the graph display update period during Phase vs. Q measurement.
Example	> QPU? < QPU 1
Compatibility	Incompatible with existing models.

8.5.3.4 Measurement results and status query commands

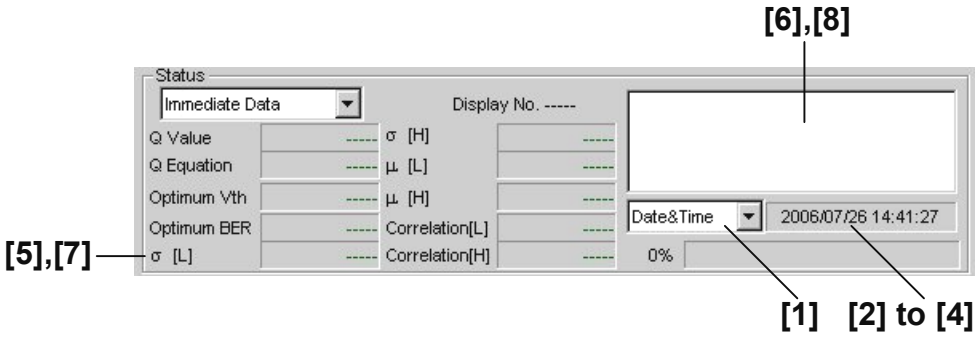


Figure 8.5.3.4-1 Measurement results and status display field

Table 8.5.3.4-1 Measurement results and status query commands

No.	Setting Items	Commands
[1]	No label (Setting of time display)	QAT
		QAT?
[2]	Query for current time	QSO?
[3]	Query for measurement start time	QST?
[4]	Query for measurement elapsed time	QEL?
[5]	Query for Vth vs. Q measurement result	QTH?
[6]	Query for Vth vs. Q measurement status	QTW?
[7]	Query for Phase vs. Q measurement result	QPH?
[8]	Query for Phase vs. Q measurement status	QSW?

QAT <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 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.

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.

QST?

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:
QST 0000,00,00,00,00,00

Function

Queries the measurement start time (Start Time) during Q measurement.

Example

```
> QST?
< QST 2005,12,31,23,59,59
```

Compatibility

Incompatible with existing models.

QEL?

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:
QEL 00,00,00,00

Function

Queries the measurement elapsed time (Elapsed time) during Q measurement.

Example

```
> QEL?
< 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 8.5.3.4-2 Vth vs. Q measurement data <result1>

Items		<numeric1>,<numeric2>	Response Format
Top side Vth/Error Rate	Number of measurements 1 to 1000	0,1,…0,1000	Form1
Bottom side Vth/Error Rate		1,1,…1,1000	Form1

The measurement result for each Vth is delimited with a comma (,) and returned.

Table 8.5.3.4-3 Vth vs. Q immediate data <result2>

Items		<numeric1>,<numeric2>	Response Format
Q value	Number of measurements 1 to 1000	2,1,…2,1000	Form2
Optimum BER		3,1,…3,1000	Form3
Optimum Vth		4,1,…4,1000	Form4
Correlation bottom		5,1,…5,1000	Form2
Correlation top		6,1,…6,1000	Form2
σ bottom		7,1,…7,1000	Form2
σ top		8,1,…8,1000	Form2
μ bottom		9,1,…9,1000	Form4
μ top		10,1,…10,1000	Form4
Q Equation		11,1,…11,1000	Form2

Table 8.5.3.4-4 Vth vs. Q measurement statistic data <result3>

Items	<numeric1>	Response Format
Total Data	12	Form5
Valid Data	13	Form5
Q Max	14	Form2
Q Min	15	Form2
Q Mean	16	Form2
Q σ	17	Form2
Q-5 σ	18	Form2

<numeric2> is omitted for statistic data.

Response

<string> = <STRING RESPONSE DATA>

Table 8.5.3.4-5 Response format

Form	Format	Description
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.

QTW?

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 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>

Items		<numeric1>,<numeric2>	Response Format
Top side Vth-Error Rate	Phase –1000 to 1000 mUI	1,–1000,…0,1000	Form1
Bottom side Vth-Error Rate		1,–1000,…1,1000	Form1

Table 8.5.3.4-7 Vth vs. Q immediate data <result2>

Items		<numeric1>,<numeric2>	Response Format
Q value	Phase –1000 to 1000 mUI	2,–1000,…2,1000	Form2
Optimum BER		3,–1000,…3,1000	Form3
Optimum Vth		4,–1000,…4,1000	Form4
Correlation bottom		5,–1000,…5,1000	Form2
Correlation top		6,–1000,…6,1000	Form2
σ bottom		7,–1000,…7,1000	Form2
σ top		8,–1000,…8,1000	Form2
μ bottom		9,–1000,…9,1000	Form4
μ top		10,–1000,…10,1000	Form4
Q Equation		11,–1000,…11,1000	Form2

Response

<string> = <STRING RESPONSE DATA>

Table 8.5.3.4-8 Response format

Form	Format	Description
Form1 Vth-ER 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.

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.

8.5.3.5 File menu setting commands

Table 8.5.3.5-1 File menu setting commands

No.	Setting Items	Commands
[1]	Q Unit	QTS
		QTS?
		QPS
		QPS?

QTS <numeric>

Parameter	<numeric> = <DECIMAL NUMERIC PROGRAM DATA> 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: `> QTS 1`
Compatibility	Incompatible with existing models.

QTS?

Response	<numeric> = <NR1 NUMERIC RESPONSE DATA> (FIX1)
Function	Queries whether the Q value display scale during Vth vs. Q measurement is set to Linear or Log display.
Example	`> QTS?` `< QTS 1`
Compatibility	Incompatible with existing models.

QPS <scale>

Parameter	<scale> = <DECIMAL NUMERIC PROGRAM DATA> 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: `> QPS 1`
Compatibility	Incompatible with existing models.

QPS?

Response	<scale> = <NR1 NUMERIC RESPONSE DATA> (FIX1)
Function	Queries whether the Q value display scale during Phase vs. Q measurement is set to Linear or Log display.
Example	> QPS? < 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

- 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).

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

Code	Message	Error detection condition
–101	Invalid character	An invalid character is included in a header or parameter. ***Notes***: • The top of mnemonic should be stated from alphabet. After this, alphabet character "A" to "Z", "a" to "z", underline "_", number "0" to "9" can be followed. • A rule of character data is similar to program mnemonic. Ex. : Case where a header includes #: :OUTPut:CLOCk:#REQuency 200 Case where a parameter include *: :OUTPut:CLOCk:FREQuency:UNIT MHZ*
–102	Syntax error	Case that the command excepted a <white space> is only "*".
–104	Data type error	The parameter type differs from the specified type. ***Notes***: • The character data is send as parameter although it should be sent a numeric parameter. • The numeric data is send as parameter although it should be a character parameter. Ex. : Case where a parameter sends data of character type to a command that accepts only numeric data: :SENSe:MEASure:EALarm:PERiod 1A,D,M,S

Table 9.2.1-1 Error messages for command errors (Cont'd)

Code	Message	Error detection condition
–108	Parameter not allowed	The number of parameters does not agree with the defined (required) number. 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:PATTern: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

Code	Message	Error detection condition
–220	Parameter error	An error is found in the parameter.
–221	Setting conflict	The parameter is correct but cannot be set because of the MP1800A/MT1810A condition. Ex. : Although the Pattern generation setting is not "Burst", then the command to change the data output sequence for a Burst data is sent

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 9.2.3-1 Error messages for device specific errors

Code	Message	Error detection condition
–310	System error	A system error occurred.

9.2.4 Supplement / Error No. and the event status register

Table 9.2.4-1 Error No. and the event status register

Error No.	Error type	Event Status Register to Be Set	Bit No.
–100 to –199	Command error	CME_BIT	5
–200 to –299	Execution error	EXE_BIT	4
–300 to –399	Device specific error	DDE_BIT	3

Appendix A Command Compatibility List

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 THE 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)

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

Function	Control Message	Data Request Message	Compatibility
Pattern Output Mode	PTO	PTO?	A
Transmission Pattern	PTN	PTN?	C
Transmission Pattern Mark Ratio	MRK	MRK?	A
Transmission Pattern Logic	LGC	LGC?	A
Bit Shift	SFT	SFT?	A
Zero-Subst Pattern Length	ZPL	ZPL?	A
Length of Consecutive Zeros of Zero Subst Pattern	ZLN	ZLN?	A
PRGM Pattern Length	DLN	DLN?	A
Burst Mode	BRM	BRM?	A
Burst Cycle	BRC	BRC?	C
Burst Enable Length	BRE	BRE?	C
Number of Pattern Pages	PAG ADR	PAG? ADR?	– –
Pattern Bit	BIT	BIT?	–
Pattern Data	PDT	PDT?	A
Pattern Binary Data	BDT	BDT?	A
Reversion of Transmission Pattern Specification Address Data (address specification)	DRA	–	A
Reversion of Transmission Pattern Specification Address Data (delta specification)	DRD	–	A
Pattern Data Input Byte Number	WRT	–	A
Pattern Data Output Byte Number	–	RED	A
ALL Preset (All Pages)	ALL	–	A
ALL Preset (One Pages)	PST	–	–
Error Insertion	EAD	EAD?	B
Error Insertion Route	ECH	ECH?	A
Level of PRBS 1	PML	PML?	–
Level of PRGM 1	GML	GML?	–

Table A.1.1-1 MP1632C PPG commands (Cont'd)

Function	Control Message	Data Request Message	Compatibility
On/Off of Output	OON	OON?	C
On/Off of Clock Output	CON	CON?	A
On/Off of XClock Output	XCO	XCO?	A
Setting of Clock Delay	CDL	CDL?	A
Clock Output Level	CLL	CLL?	–
XClock Output Level	XCL	XCL?	–
Clock Output Amplitude	CAP	CAP?	A
XClock Output Amplitude	XCA	XCA?	A
Clock Output Offset	COS	COS?	B
XClock Output Offset	XCF	XCF?	A
Offset Reference Port Selection	OPS	OPS?	A
Offset Reference Value	OFS	OFS?	A
Setting of Clock Duty	CDT	CDT?	A
Setting of XClock Duty	XDT	XDT?	A
Setting of Clock DC Impedance	CIM	CIM?	–
ON/OFF of Data Output	DON	DON?	A
ON/OFF of XData Output	XDO	XDO?	C
Setting of Data DC Impedance	DIM	DIM?	–
Data Cross Point	DCR	DCR?	B
XData Cross Point	XDC	XDC?	B
Data Output Level	DAL	DAL?	A
XData Output Level	XDL	XDL?	A
Data Output Amplitude	DAP	DAP?	A
XData Output Amplitude	NAP	NAP?	A
Data Output Offset	DOS	DOS?	A
XData Output Offset	NOS	NOS?	A
Selection of Sync Output Signal	SOP	SOP?	B
Selection of Pattern Sync Output Position	PSP	PSP?	B
PPG Clock Loss	–	PCL?	A
Delay Trouble3	–	DTR?	A
Delay Setting State	–	DLY?	A
Setting of Clock/XClock Grouping	CGR	CGR?	A
Setting of Data/XData Grouping	TRK	TRK?	A
PPG/ED Pattern Common Setting	PCO	PCO?	–

Table A.1.1-2 MP1632C Internal Synthesizer commands

Function	Control Message	Data Request Message	Compatibility
Setting of Clock Frequency	FRQ	FRQ?	B
Clock Reference Signal	RFC	RFC?	A
PLL Lock State	–	PLL?	–

Table A.1.1-3 MP1632C ED commands

Function	Control Message	Data Request Message	Compatibility
Pattern Reception Mode (Pattern input mode)	PTI	PTI?	A
Reception Pattern	PTN	PTN?	B
Reception Pattern Mark Ratio	MRK	MRK?	A
Reception Pattern Logic	LGC	LGC?	A
Bit Shift	SFT	SFT?	A
Zero-Subst Pattern Length	ZPL	ZPL?	–
Length of Consecutive Zeros of Zero Subst Pattern	ZLN	ZLN?	B
PRGM Pattern Length	DLN	DLN?	A
Automatic Synchronization Function	SYN	SYN?	A
Synchronization Loss Threshold	LTH	LTH?	–
Synchronization Gain Threshold	GTH	GTH?	–
Internal Synchronization Threshold	ITH	ITH?	–
Frame Synchronization Function	FSY	FSY?	–
Frame Length	FLN	FLN?	A
Number of Pattern Pages	PAG ADR	PAG? ADR?	– –
Pattern Bit	BIT	BIT?	–
Pattern Data	PDT	PDT?	A
Pattern Binary Data	BDT	BDT?	A
Reversion of Transmission Pattern Specification Address Data (address specification)	DRA	–	A
Reversion of Transmission Pattern Specification Address Data (delta specification)	DRD	–	A
Pattern Data Input Byte Number	WRT	–	–
Pattern Data Output Byte Number	–	RED	–
ALL Preset (All Pages)	ALL	–	A
ALL Preset (One Pages)	PST	–	–
Level of PRBS 1	PML	PML?	–
Level of PRGM 1	GML	GML?	–
Setting of Measurement Item	TIT	TIT?	–

Table A.1.1-3 MP1632C ED commands (Cont'd)

Function	Control Message	Data Request Message	Compatibility
Measurement Start/Restart	STA	–	A
Measurement Stop	STO	–	A
Measurement restart setting	MRS	MRS?	C
Setting of Measurement Mode	MOD	MOD?	A
Measurement Period Setting	PRD	PRD?	A
Timed Start Setting	TSS	TSS?	–
Timed Start Time Setting	STI	STI?	–
Error Detection Mode Selection	ETY	ETY?	–
Measurement Mask Route Selection	SCH	SCH?	–
Measurement Start Time	–	MSA?	–
Measurement Stop Time	–	MSO?	A
Measurement State	–	MSR?	A
Measurement Elapsed Time	–	MLP?	B
Measurement Residual Time	–	ETI?	A
Intermediate Data Creation Time	–	INT?	–
Alarm Occurrence Time	–	AOT?	–
Alarm Recovery Time	–	ART?	–
Setting of Eye Margin Measurement Mode	EMD	EMD?	–
Eye Margin Measurement Item Setting	EYT	EYT?	C
Eye Margin Measurement Threshold Setting	EMT	EMT?	C
Eye Margin Measurement Resolution Setting	EMR	EMR?	C
Eye Diagram Measurement Threshold Setting	EDT	EDT?	C
Eye Diagram Measurement Effective Points	–	EDP?	A
Setting of Auto search Mode	ASM	ASM?	B
Auto search Function	SRH	SRH?	B
PRBS pattern Search Function	PSH	PSH?	–
Clock Input Polarity	CPL	CPL?	–
Clock Input Phase Setting	CPA	CPA?	B
Clock Input Terminal Voltage	CTM	CTM?	–
Data Input Level	DAL	DAL?	B
Data Input Threshold Setting	DTH	DTH?	A
Setting of Data Input Terminal Voltage	DTM	DTM?	B
Clock Loss Process Option	CLS	CLS?	A
Sync Loss Process Option	SLS	SLS?	A

Table A.1.1-3 MP1632C ED commands (Cont'd)

Function	Control Message	Data Request Message	Compatibility
Measurement Interval Time Selection	ITV	ITV?	B
Error Performance Data Measurement ON/OFF	PRF	PRF?	–
Error Performance Threshold Setting	ETH	ETH?	A
Threshold EI/%EFI Measurement ON/OFF	TEI	TEI?	–
Power Fail Measurement ON/OFF	PFM	PFM?	–
Error Ratio Measurement Result	–	ER?	A
Error Count Measurement Result	–	EC?	A
Clock Count Measurement Result	–	CC?	A
Error Interval Count Measurement Result	–	EI?	A
Error Free Interval Rate Measurement Result	–	EFI?	A
Clock Frequency Measurement Result	–	FRQ?	A
Performance % Measurement Result	–	PFP?	A
Performance Count Measurement Result	–	PFC?	A
Threshold EI Measurement Result	–	THE?	A
Threshold %EFI Measurement Result	–	THF?	A
Alarm Interval Measurement Result	–	AIN?	A
1 Second Average Error Ratio Measurement Result	–	OER?	–
1 Second Average Error Count Measurement Result	–	OEC?	–
Data Output Format	FMT	FMT?	–
Intermediate Measurement Result Output Function	–	IMD?	A
Eye Margin Measurement Result	–	EMM?	–
Eye Diagram Measurement Result	–	EDM?	C
Measurement Result Store Function	BST	–	A
Measurement Result Store Buffer Clear Function	BCL	–	–
Clock Loss Status	–	CLI?	A
Sync Loss Status	–	SLI?	A
Error Detection Status	–	ERS?	A
Error Detection Status	–	BES?	A
Power Fail Status	–	POF?	–
Delay Trouble	–	DTR?	–
Delay Setting Status	–	DLY?	A
Synchronous Signal Output Selection	SOP	SOP?	B
PPG/ED Pattern Common Setting	PCO	PCO?	–

Table A.1.1-4 MP1632C Main Frame commands

Function	Control Message	Data Request Message	Compatibility
Child window Open/Close	WOC	WOC?	–
System Child window panel selection	SYW	SYW?	–
System Child window panel selection	SEW	SEW?	–
Test Menu System Child window panel selection	TEW	TEW?	–
Real Time/Measurement Time display selection	TIM	TIM?	A
Result Child window Sub window selection	REW	REW?	–
Intermediate Result display Function	CUR	CUR?	A
Performance Result Display Function	PFD	PFD?	–
Zoom window display ON/OFF setting	ZOM	ZOM?	A
Zoom window display Item setting	ZOI	ZOI?	–
Zoom window Performance item setting	ZOP	ZOP?	–
Zoom window Threshold EI/%EFI setting	ZOT	ZOT?	–
History Lamp Reset	HRE	–	A
Eye Margin Measurement Display Error Rate	DER	DER?	C
Eye Diagram Display window selection	EDD	EDD?	–
Eye diagram display Auto Scale	ASC	–	C
Eye diagram Voltage Scale Setting	VSC	VSC?	C
Eye diagram Phase Scale Setting	PSC	PSC?	C
Eye diagram Marker Display ON/OFF	EMA	EMA?	–
Marker Operation Mode	MKM	MKM?	–
Marker Specification Threshold	MKT	MKT?	–
Marker Move Specification	MMV	–	–
Marker Position	MKP	MKP?	–
Customize window display Slot setting	CUS	CUS?	–
Customize window display Item setting	CUI	CUI?	–
Customize window display Item Port setting	CUP	CUP?	–
Customize window Pattern Setting Unit	CPT	CPT?	A
Customize Edit Pattern Position	CPF	CPF?	–
Customize window Measurement display Item	CRI	CRI?	–
System Error Monitor Buzzer ON/OFF	SYS	SYS?	A
Error Monitor Buzzer ON/OFF	MON	MON?	A
Alarm Monitor Buzzer ON/OFF	ALM	ALM?	A
System Error Type	SYT	SYT?	B
Internal Timer Setting	RTM	RTM?	–

Table A.1.1-4 MP1632C Main Frame commands (Cont'd)

Function	Control Message	Data Request Message	Compatibility
Input/Output Buffer Size	BUF	BUF?	–
Hardware Organization	–	HDR?	C
Software Organization	–	SFR?	–
Data Recall	RCL	–	–
Data Save	SAV	–	–
Quick Recall	QRC	–	–
Quick Save	QSA	–	–
File No./Directory Mode Selection	FIL	FIL?	–
File Catalog	–	CAT?	–
Floppy Access Status	–	MAC?	–
Initialize	INI	–	A
Hard Copy	HCP	–	A
Prints Text	PTX	–	–
Threshold EI/%EFI print selection	THR	THR?	–
Performance Data print selection	EPF	EPF?	–
Error Measurement print selection	ERP	ERP?	–
Alarm Interval print selection	ALP	ALP?	–
Frequency Data print selection	FRP	FRP?	–
Manual Print Start	PSA	–	–
Addressed Pattern Print	PAD	–	–
All Pattern Print	PAL	–	–
Eye Margin Measurement Result Print	PEM	–	–
Recall Eye Diagram Print	PER	–	–
LSB/MSB Swap Command	SWP	SWP?	–
Termination Select	TRM	TRM?	A
Service Request Enable Register (ED)	SRQ	SRQ?	C
Status Byte Register (ED)	STB	STB?	C
Standard Event Status Enable Register (ED)	ESE	ESE?	C
Event Status Register (ED)	ESR	ESR?	C
Extended Event Status Enable Register (ED)(ED)	EES	EES?	C
Event Status Register (ED)	EER	EER?	C
Service Request Enable Register (PPG/Synthe)	SRQ	SRQ?	C
Status Byte Register (PPG/Synthe)	STB	STB?	C
Standard Event Status Enable Register (PPG/Synthe)	ESE	ESE?	C

Table A.1.1-4 MP1632C Main Frame commands (Cont'd)

Function	Control Message	Data Request Message	Compatibility
Event Status Register (PPG/Synthe)	ESR	ESR?	C
Extended Event Status Enable Register (PPG/Synthe)	EES	EES?	C
Event Status Register (PPG/Synthe)	EER	EER?	C

Table A.1.1-5 MP1632C SOURce Subsystem

Commands	Compatibility
:SOURce:PATTern:OMODe <mode>	A
:SOURce:PATTern:OMODe?	A
:SOURce:PATTern:TYPE <type>	B
:SOURce:PATTern:TYPE?	B
:SOURce:PATTern:PRBS:MRATio <mratio>	A
:SOURce:PATTern:PRBS:MRATio?	A
:SOURce:PATTern:PRBS:BSHift <numeric>	A
:SOURce:PATTern:PRBS:BSHift?	A
:SOURce:PATTern:ZSUBstitute:LENGth <numeric>	A
:SOURce:PATTern:ZSUBstitute:LENGth?	A
:SOURce:PATTern:ZSUBstitute:ZLENgth <numeric>	A
:SOURce:PATTern:ZSUBstitute:ZLENgth?	A
:SOURce:PATTern:ZSUBstitute:LOGic <logic>	–
:SOURce:PATTern:ZSUBstitute:LOGic?	–
:SOURce:PATTern:PROGram:LENGth <numeric>	–
:SOURce:PATTern:PROGram:LENGth?	–
:SOURce:PATTern:PROGram:LOGic <logic>	–
:SOURce:PATTern:PROGram:LOGic?	–
:SOURce:PATTern:BURSt:MODE <mode>	A
:SOURce:PATTern:BURSt:MODE?	A
:SOURce:PATTern:BURSt:CYCLe <numeric>	C
:SOURce:PATTern:BURSt:CYCLe?	C
:SOURce:PATTern:BURSt:ELENgth <numeric>	C
:SOURce:PATTern:BURSt:ELENgth?	C
:SOURce:PATTern:DATA:WHOLe <start>,<end>,<data>	A
:SOURce:PATTern:DATA:WHOLe? <start>	A

Table A.1.1-5 MP1632C SOURce Subsystem (Cont'd)

Commands	Compatibility
:SOURce:PATTern:BDATa:WHOLe <start>,<end>,<data>	A
:SOURce:PATTern:BDATa:WHOLe? <start>	A
:SOURce:PATTern:DREVerse:ADDRess <start>,<end>	A
:SOURce:PATTern:DREVerse:DELTa <start>,<delta>	A
:SOURce:PATTern:EADDition:SET <boolean>	C
:SOURce:PATTern:EADDition:SET?	C
:SOURce:PATTern:EADDition:SINGle	A
:SOURce:PATTern:EADDition:RATE <mode>	B
:SOURce:PATTern:EADDition:RATE?	B
:SOURce:PATTern:EADDition:ROUTe <route>	A
:SOURce:PATTern:EADDition:ROUTe?	A
:SOURce:PATTern:LOGic:PRBS <mark>	–
:SOURce:PATTern:LOGic:PRBS?	–
:SOURce:PATTern:LOGic:PRGM <mark>	–
:SOURce:PATTern:LOGic:PRGM?	–

Table A.1.1-6 MP1632C OUTPut Subsystem

Commands	Compatibility
:OUTPut:SET	C
:OUTPut:SET?	C
:OUTPut:CLOCk:OUTPut <port>,<boolean>	C
:OUTPut:CLOCk:OUTPut? <port>	C
:OUTPut:CLOCk:DELay <numeric>	–
:OUTPut:CLOCk:DELay?	–
:OUTPut:CLOCk:LEVel <port>,<level>	–
:OUTPut:CLOCk:LEVel? <port>	–
:OUTPut:CLOCk:AMPLitude <port>,<numeric>	A
:OUTPut:CLOCk:AMPLitude? <port>	A
:OUTPut:CLOCk:OFFSet <port>,<numeric>	B
:OUTPut:CLOCk:OFFSet? <port>	B
:OUTPut:CLOCk:OREFerence <port>,<offset>	–
:OUTPut:CLOCk:OREFerence? <port>	–
:OUTPut:CLOCk:Duty <port>,<numeric>	A
:OUTPut:CLOCk:Duty? <port>	A

Table A.1.1-6 MP1632C OUTPut Subsystem (Cont'd)

Commands	Compatibility
:OUTPut:CLOCk:IMPedance <impedance>	–
:OUTPut:CLOCk:IMPedance?	–
:OUTPut:DATA:OUTPut <port>,<boolean>	C
:OUTPut:DATA:OUTPut? <port>	C
:OUTPut:DATA:IMPedance <impedance>	–
:OUTPut:DATA:IMPedance?	–
:OUTPut:DATA:CPOint <port>,<numeric>	B
:OUTPut:DATA:CPOint? <port>	B
:OUTPut:DATA:LEVel <port>,<level>	A
:OUTPut:DATA:LEVel? <port>	A
:OUTPut:DATA:AMPLitude <port>,<numeric>	A
:OUTPut:DATA:AMPLitude? <port>	A
:OUTPut:DATA:OFFSet <port>,<numeric>	A
:OUTPut:DATA:OFFSet? <port>	A
:OUTPut:DATA:OREFerence <port>,<offset>	–
:OUTPut:DATA:OREFerence? <port>	–
:OUTPut:SYNC:SOURce <source>	A
:OUTPut:SYNC:SOURce?	A
:OUTPut:SYNC:POSition <address>	B
:OUTPut:SYNC:POSition?	B
:OUTPut:SYNC:SOURce <source>	A
:OUTPut:SYNC:SOURce?	A
:OUTPut:CLOCk:FREQuency <numeric>	B
:OUTPut:CLOCk:FREQuency?	B
:OUTPut:RCLock.:SELect <clock>	B
:OUTPut:RCLock.:SELect?	B

Table A.1.1-7 MP1632C CALCulate Subsystem

Commands	Compatibility
:CALCulate:DATA:MONitor?	C

Table A.1.1-8 MP1632C INSTrument Subsystem

Commands	Compatibility
:INSTrument:COUPle:CLOCk:TRACking <boolean>	–
:INSTrument:COUPle:CLOCk:TRACking?	–
:INSTrument:COUPle:DATA:TRACking <boolean>	–
:INSTrument:COUPle:DATA:TRACking?	–
:INSTrument:COUPle:PATTern:SET <boolean>	–
:INSTrument:COUPle:PATTern:SET?	–
:INSTrument:COUPle:PATTern:SET <boolean>	–
:INSTrument:COUPle:PATTern:SET?	–

Table A.1.1-9 MP1632C SENSe Subsystem

Commands	Compatibility
:SENSe:PATTern:IMODe <mode>	A
:SENSe:PATTern:IMODe?	A
:SENSe:PATTern:TYPE <type>	B
:SENSe:PATTern:TYPE?	B
:SENSe:PATTern:PRBS:MRATio <mratio>	A
:SENSe:PATTern:PRBS:MRATio?	A
:SENSe:PATTern:PRBS:BSHift <numeric>	A
:SENSe:PATTern:PRBS:BSHift?	A
:SENSe:PATTern:ZSUBstitute:LENGth <numeric>	A
:SENSe:PATTern:ZSUBstitute:LENGth?	A
:SENSe:PATTern:ZSUBstitute:ZLENgth <numeric>	B
:SENSe:PATTern:ZSUBstitute:ZLENgth?	B
:SENSe:PATTern:ZSUBstitute:LOGic <logic>	–
:SENSe:PATTern:ZSUBstitute:LOGic?	–
:SENSe:PATTern:PROGram:LENGth <numeric>	–
:SENSe:PATTern:PROGram:LENGth?	–
:SENSe:PATTern:PROGram:LOGic <logic>	–
:SENSe:PATTern:PROGram:LOGic?	–
:SENSe:PATTern:SYNC:ASYNc <boolean>	A
:SENSe:PATTern:SYNC:ASYNc?	A
:SENSe:PATTern:SYNC:THReshold:LOSS <thre>	–
:SENSe:PATTern:SYNC:THReshold:LOSS?	–

Table A.1.1-9 MP1632C SENSe Subsystem (Cont'd)

Commands	Compatibility
:SENSe:PATTern:SYNC:THReshold:GAIN <thre>	C
:SENSe:PATTern:SYNC:THReshold:GAIN?	C
:SENSe:PATTern:SYNC:THReshold:INTernal <boolean>	–
:SENSe:PATTern:SYNC:THReshold:INTernal?	–
:SENSe:PATTern:SYNC:PSMode <mode>	A
:SENSe:PATTern:SYNC:PSMode?	A
:SENSe:PATTern:SYNC:FLENgth <numeric>	A
:SENSe:PATTern:SYNC:FLENgth?	A
:SENSe:PATTern:DATA:WHOLe <start>,<end>,<data>	A
:SENSe:PATTern:DATA:WHOLe? <start>	A
:SENSe:PATTern:BDATa:WHOLe <start>,<end>,<bdata>	A
:SENSe:PATTern:BDATa:WHOLe? <start>	A
:SENSe:PATTern:DREVerse:ADDRess <start>,<end>	A
:SENSe:PATTern:DREVerse:DELTa <start>,<delta>	A
:SENSe:PATTern:LOGic:PRBS <mark>	–
:SENSe:PATTern:LOGic:PRBS?	–
:SENSe:PATTern:LOGic:PRGM <mark>	–
:SENSe:PATTern:LOGic:PRGM?	–
:SENSe:MEASure:TEST?	–
:SENSe:MEASure:STARt	A
:SENSe:MEASure:STOP	A
:SENSe:MEASure:MREStart <boolean>	A
:SENSe:MEASure:MREStart?	A
:SENSe:MEASure:EALarm:MODE <mode>	A
:SENSe:MEASure:EALarm:MODE?	A
:SENSe:MEASure:EALarm:PERiod <d>,<h>,<m>,<s>	A
:SENSe:MEASure:EALarm:PERiod?	A
:SENSe:MEASure:EALarm:BTIMe:SET <boolean>	–
:SENSe:MEASure:EALarm:BTIMe:SET?	–
:SENSe:MEASure:EALarm:BTIMe:STARt <y>,<m>,<d>,<h>,<m>,<s>	–
:SENSe:MEASure:EALarm:BTIMe:STARt?	–
:SENSe:MEASure:EALarm:ERRor:TYPE?	–
:SENSe:MEASure:EALarm:MASK:ROUTe <route>,<boolean>	–
:SENSe:MEASure:EALarm:MASK:ROUTe?	–
:SENSe:MEASure:EALarm:STARt?	A
:SENSe:MEASure:EALarm:STOP?	A
:SENSe:MEASure:EALarm:STATe?	A
:SENSe:MEASure:EALarm:ELAPsed?	A
:SENSe:MEASure:EALarm:TIMed?	A
:SENSe:MEASure:EALarm:ITIMe?	A
:SENSe:MEASure:EALarm:AOCCur? <alarm>	A
:SENSe:MEASure:EALarm:ARECver? <alarm>	A

Table A.1.1-9 MP1632C SENSe Subsystem (Cont'd)

Commands	Compatibility
:SENSe:MEASure:EMARgin:MODE <mode>	–
:SENSe:MEASure:EMARgin:MODE?	–
:SENSe:MEASure:EMARgin:TYPE?	–
:SENSe:MEASure:EMARgin:TYPE <type>	–
:SENSe:MEASure:EMARgin:MARGin:THReshold <thre>	A
:SENSe:MEASure:EMARgin:MARGin:THReshold?	A
:SENSe:MEASure:EMARgin:MARGin:RESolution <type>	–
:SENSe:MEASure:EMARgin:MARGin:RESolution?	–
:SENSe:MEASure:EMARgin:DIAGram:THReshold <thre>,<boolean>	–
:SENSe:MEASure:EMARgin:DIAGram:THReshold?	–
:SENSe:MEASure:EMARgin:DIAGram:POINt? <thre>	–
:SENSe:MEASure:EMARgin:STATe?	–
:SENSe:MEASure:EMARgin:STARt?	–
:SENSe:MEASure:EMARgin:STOP?	–
:SENSe:MEASure:EMARgin:ELAPsed?	–
:SENSe:MEASure:ASEarch:MODE <mode>	B
:SENSe:MEASure:ASEarch:MODE?	B
:SENSe:MEASure:ASEarch:STARt	A
:SENSe:MEASure:ASEarch:STOP	A
:SENSe:MEASure:ASEarch:PATTern <boolean>	–
:SENSe:MEASure:ASEarch:PATTern?	–
:SENSe:MEASure:ASEarch:STATe?	A

Table A.1.1-10 MP1632C INPut Subsystem

Commands	Compatibility
:INPut:CLOCk:POLarity <pol>	–
:INPut:CLOCk:POLarity?	–
:INPut:CLOCk:DELay <numeric>	A
:INPut:CLOCk:DELay?	A
:INPut:CLOCk:TERMination <term>	A
:INPut:CLOCk:TERMination?	A
:INPut:DATA:LEVel <level>	C
:INPut:DATA:LEVel?	C
:INPut:DATA:THReshold <numeric>	B
:INPut:DATA:THReshold?	B
:INPut:DATA:TERMination <term>	C
:INPut:DATA:TERMination?	C

Table A.1.1-11 MP1632C CALCulate Subsystem

Commands	Compatibility
:CALCulate:EALarm:CLEValuation <boolean>	A
:CALCulate:EALarm:CLEValuation?	A
:CALCulate:EALarm:SLEValuation <boolean>	A
:CALCulate:EALarm:SLEValuation?	A
:CALCulate:EALarm:ERRor:INTerval <numeric>,<suffix>	A
:CALCulate:EALarm:ERRor:INTerval?	A
:CALCulate:EALarm:PERFormance:MEASurement <boolean>	–
:CALCulate:EALarm:PERFormance:MEASurement?	–
:CALCulate:EALarm:PERFormance:THReshold <thre>	A
:CALCulate:EALarm:PERFormance:THReshold?	A
:CALCulate:EALarm:TEINterval:MEASurement <boolean>	–
:CALCulate:EALarm:TEINterval:MEASurement?	–
:CALCulate:EALarm:PFAil <boolean>	–
:CALCulate:EALarm:PFAil?	–
:CALCulate:DATA:EALarm? <string>	B
:CALCulate:DATA:EMARgin? <string>	–
:CALCulate:DATA:STORe <string>	–
:CALCulate:DATA:CLEar <string>	–
:CALCulate:DATA:MONitor? <item>	B
:CALCulate:DATA:MONitor? <string>	A

Table A.1.1-12 MP1632C DISPlay Subsystem

Commands	Compatibility
:DISPlay:WINDow:OPEN <disp>	–
:DISPlay:WINDow:CLOSe <disp>	–
:DISPlay:SYSTem[:NAME] <name>	–
:DISPlay:SYSTem[:NAME]?	–
:DISPlay:SETup[:NAME] <name>	–
:DISPlay:SETup[:NAME]?	–
:DISPlay:TEST[:NAME] <name>	–
:DISPlay:TEST[:NAME]?	–
:DISPlay:RESult:TIME <time>	A
:DISPlay:RESult:TIME?	A
:DISPlay:RESult:EALarm[:NAME] <name>	–
:DISPlay:RESult:EALarm[:NAME]?	–
:DISPlay:RESult:EALarm:MODE <mode>	C
:DISPlay:RESult:EALarm:MODE?	C
:DISPlay:RESult:EALarm:ALL:PTYPE <ptype>	–
:DISPlay:RESult:EALarm:ALL:PTYPE?	–
:DISPlay:RESult:EALarm:ZOOM1|:ZOOM2|:ZOOM3|:ZOOM4:SET <boolean>	–
:DISPlay:RESult:EALarm:ZOOM1|:ZOOM2|:ZOOM3|:ZOOM4:SET?	–
:DISPlay:RESult:EALarm:ZOOM1|:ZOOM2|:ZOOM3|:ZOOM4:ITEM <item>	–
:DISPlay:RESult:EALarm:ZOOM1|:ZOOM2|:ZOOM3|:ZOOM4:ITEM?	–
:DISPlay:RESult:EALarm:ZOOM1|:ZOOM2|:ZOOM3|:ZOOM4:PITem <pitem>	–

Table A.1.1-12 MP1632C DISPlay Subsystem (Cont'd)

Commands	Compatibility
:DISPlay:RESult:EALarm:ZOOM1|:ZOOM2|:ZOOM3|:ZOOM4:PITem?	–
:DISPlay:RESult:EALarm:ZOOM1|:ZOOM2|:ZOOM3|:ZOOM4:TITem <titem>	–
:DISPlay:RESult:EALarm:ZOOM1|:ZOOM2|:ZOOM3|:ZOOM4:TITem?	–
:DISPlay:RESult:EALarm:HRESet	A
:DISPlay:RESult:EMARgin:ERATe <erate>,<boolean>	–
:DISPlay:RESult:EMARgin:ERATe?	–
:DISPlay:RESult:EMARgin:SCALe:ASCale	–
:DISPlay:RESult:EMARgin:SCALe:VOLTage <min>,<step>	–
:DISPlay:RESult:EMARgin:SCALe:VOLTage?	–
:DISPlay:RESult:EMARgin:SCALe:PHASe <min>,<step>	–
:DISPlay:RESult:EMARgin:SCALe:PHASe?	–
:DISPlay:RESult:EMARgin:MARKer:SET <marker>,<boolean>	–
:DISPlay:RESult:EMARgin:MARKer:SET? <marker>	–
:DISPlay:RESult:EMARgin:MARKer:MODE <mode>	–
:DISPlay:RESult:EMARgin:MARKer:MODE?	–
:DISPlay:RESult:EMARgin:MARKer:ERATe <erate>	–
:DISPlay:RESult:EMARgin:MARKer:ERATe?	–
:DISPlay:RESult:EMARgin:MARKer:MOVE <marker>,<mode>	–
:DISPlay:RESult:EMARgin:MARKer:POSition? <marker>	–
:DISPlay:RESult:EMARgin:TEMPlate:SELect <select>,<boolean>	–
:DISPlay:RESult:EMARgin:TEMPlate:SELect?	–
:DISPlay:RESult:EMARgin:TEMPlate:MOVE <template>,<mode>	–
:DISPlay:RESult:EMARgin:TEMPlate:PNUMber <template>,<point>	–
:DISPlay:RESult:EMARgin:TEMPlate:PNUMber? <template>	–
:DISPlay:RESult:EMARgin:TEMPlate:OFFSet:VOLTage <template>,<voltage>	–
:DISPlay:RESult:EMARgin:TEMPlate:OFFSet:VOLTage? <template>	–
:DISPlay:RESult:EMARgin:TEMPlate:OFFSet:PHASe <template>,<phase>	–
:DISPlay:RESult:EMARgin:TEMPlate:OFFSet:PHASe? <template>	–
:DISPlay:RESult:EMARgin:TEMPlate:POINt:VOLTage <template>,<point>,<voltage>	–
:DISPlay:RESult:EMARgin:TEMPlate:POINt:VOLTage? <template>,<point>	–
:DISPlay:RESult:EMARgin:TEMPlate:POINt:PHASe <template>,<point>,<phase>	–
:DISPlay:RESult:EMARgin:TEMPlate:POINt:PHASe? <template>,<point>	–
:DISPlay:RESult:EMARgin:TEMPlate:CONNect <template>,<boolean>	–
:DISPlay:RESult:EMARgin:TEMPlate:SELect? <template>	–
:DISPlay:RESult:CUSTomize:SETup1|:SETup2|:SETup3|:SETup4|:SETup5|:SETup6:UNIT <unit>	–
:DISPlay:RESult:CUSTomize:SETup1|:SETup2|:SETup3|:SETup4|:SETup5|:SETup6:UNIT?	–
:DISPlay:RESult:CUSTomize:SETup1|:SETup2|:SETup3|:SETup4|:SETup5|:SETup6:ITEM <string>	–
:DISPlay:RESult:CUSTomize:SETup1|:SETup2|:SETup3|:SETup4|:SETup5|:SETup6:ITEM?	–
:DISPlay:RESult:CUSTomize:SETup1|:SETup2|:SETup3|:SETup4|:SETup5|:SETup6:PORT <port>	–

Table A.1.1-12 MP1632C DISPlay Subsystem (Cont'd)

Commands	Compatibility
:DISPlay:RESult:CUSTomize:SETup1|:SETup2|:SETup3|:SETup4|:SETup5|:SETup6:PORT?	–
:DISPlay:RESult:CUSTomize:PATTern:UNIT <unit>	–
:DISPlay:RESult:CUSTomize:PATTern:UNIT?	–
:DISPlay:RESult:CUSTomize:PATTern:OFFSet <numeric>	–
:DISPlay:RESult:CUSTomize:PATTern:OFFSet?	–
:DISPlay:RESult:CUSTomize:RESult:ITEM <item>	–
:DISPlay:RESult:CUSTomize:RESult:ITEM?	–

Table A.1.1-13 MP1632C SYSTem Subsystem

Commands	Compatibility
:SYSTem:BEEPer:ERRor:SET <boolean>	A
:SYSTem:BEEPer:ERRor:SET?	A
:SYSTem:BEEPer:ALARm:SET <boolean>	A
:SYSTem:BEEPer:ALARm:SET?	A
:SYSTem:BEEPer:SYSTem:SET <boolean>	A
:SYSTem:BEEPer:SYSTem:SET?	A
:SYSTem:BEEPer:SYSTem:TYPE <type>,<boolean>	B
:SYSTem:BEEPer:SYSTem:TYPE?	A
:SYSTem:MODE?	–
:SYSTem:DATE <year>,<month>,<day>	–
:SYSTem:DATE?	–
:SYSTem:TIME <hour>,<min>,<sec>	–
:SYSTem:TIME?	–
:SYSTem:BSIZe? <mode>	–
:SYSTem:ERRor?	A
:SYSTem:VERSion?	A
:SYSTem:ORGanization:HARDware?	C
:SYSTem:ORGanization:SOFTware?	–
:SYSTem:MMEMory:RECall <type>,<file_name>,<unit>	C
:SYSTem:MMEMory:RECall "EYE:TEMPlate1", <file_name>,0	–
:SYSTem:MMEMory:STORe <type>,<file_name>,<unit>	C
:SYSTem:MMEMory:STORe "EYE:TEMPlate1", <file_name>,0	–
:SYSTem:MMEMory:QRECall <file_name>	C
:SYSTem:MMEMory:QSTore <file_name>,<comment>	C
:SYSTem:MMEMory:CATalog? <drv_dir>,<type>	–
:SYSTem:MEMory:INITialize	A

Table A.1.1-13 MP1632C SYSTem Subsystem (Cont'd)

Commands	Compatibility
:SYSTem:PRINt:COPY	A
:SYSTem:PRINt:TEXT <string>	–
:SYSTem:PRINt:EALarm <item>	C
:SYSTem:PRINt:PATTern:TABLe:ADDRess <unit>,<form>,<start>,<end>	–
:SYSTem:PRINt:PATTern:TABLe:ALL <unit>,<form>	–
:SYSTem:PRINt:CANCel	–
:SYSTem:PRINt:EMARgin <boolean>	–
:SYSTem:PRINt:REMARgin <boolean>	–
:SYSTem:TERMination	A
:SYSTem:TERMination?	A

Table A.1.1-14 MP1632C STATus Subsystem

Commands	Compatibility
:STATus:PRESet	A
:STATus:OPERation[:EVENt]?	C
:STATus:OPERation:CONDition?	C
:STATus:OPERation:ENABle <numeric>	C
:STATus:OPERation:ENABle?	C
:STATus:OPERation:PTRansition <numeric>	C
:STATus:OPERation:PTRansition?	C
:STATus:OPERation:NTRansition <numeric>	C
:STATus:OPERation:NTRansition?	C
:STATus:OPERation:INSTrument[:EVENt]?	–
:STATus:OPERation:INSTrument:ENABle <numeric>	–
:STATus:OPERation:INSTrument:ENABle?	–
:STATus:OPERation:INSTrument:PTRansition <numeric>	–
:STATus:OPERation:INSTrument:PTRansition?	–
:STATus:OPERation:INSTrument:NTRansition <numeric>	–
:STATus:OPERation:INSTrument:NTRansition?	–
:STATus:QUEStionable[:EVENt]?	–
:STATus:QUEStionable:CONDition?	–
:STATus:QUEStionable:ENABle <numeric>	–
:STATus:QUEStionable:ENABle?	–

Table A.1.1-14 MP1632C STATus Subsystem (Cont'd)

Commands	Compatibility
:STATus:QUEStionable:PTRansition <numeric>	–
:STATus:QUEStionable:PTRansition?	–
:STATus:QUEStionable:NTRansition <numeric>	–
:STATus:QUEStionable:NTRansition?	–
:STATus:QUEStionable:MONitor[:EVENt]?	–
:STATus:QUEStionable:MONitor:CONDition?	–
:STATus:QUEStionable:MONitor:ENABle <numeric>	–
:STATus:QUEStionable:MONitor:ENABle?	–
:STATus:QUEStionable:MONitor:PTRansition <numeric>	–
:STATus:QUEStionable:MONitor:PTRansition?	–
:STATus:QUEStionable:MONitor:NTRansition <numeric>	–
:STATus:QUEStionable:MONitor:NTRansition?	–
:STATus:QUEStionable:MONitor:SLOT1|:SLOT3|:SLOT4[:EVENt]?	–
:STATus:QUEStionable:MONitor:SLOT1|:SLOT3|:SLOT4:CONDition?	–
:STATus:QUEStionable:MONitor:SLOT1|:SLOT3|:SLOT4:ENABle <numeric>	–
:STATus:QUEStionable:MONitor:SLOT1|:SLOT3|:SLOT4:ENABle?	–
:STATus:QUEStionable:MONitor:SLOT1|:SLOT3|:SLOT4:PTRansition <numeric>	–
:STATus:QUEStionable:MONitor:SLOT1|:SLOT3|:SLOT4:PTRansition?	–
:STATus:QUEStionable:MONitor:SLOT1|:SLOT3|:SLOT4:NTRansition <numeric>	–
:STATus:QUEStionable:MONitor:SLOT1|:SLOT3|:SLOT4:NTRansition?	–
:STATus:QUEStionable:MONitor:SLOT3|:SLOT4:G32P|:G32E.[:EVENt]?	–
:STATus:QUEStionable:MONitor:SLOT3|:SLOT4:G32P|:G32E.:CONDition?	–
:STATus:QUEStionable:MONitor:SLOT3|:SLOT4:G32P|:G32E.:ENABle <numeric>	–
:STATus:QUEStionable:MONitor:SLOT3|:SLOT4:G32P|:G32E.:ENABle?	–
:STATus:QUEStionable:MONitor:SLOT3|:SLOT4:G32P|:G32E:PTRansition <numeric>	–
:STATus:QUEStionable:MONitor:SLOT3|:SLOT4:G32P|:G32E:PTRansition?	–
:STATus:QUEStionable:MONitor:SLOT3|:SLOT4:G32P|:G32E:NTRansition <numeric>	–
:STATus:QUEStionable:MONitor:SLOT3|:SLOT4:G32P|:G32E:NTRansition?	–

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 A.1.2-1 MP1761A/B/C device message

Section	Function	Control Message	Data Request Message	Compatibility
INTERNAL CLOCK	Internal clock frequency	FRQ	FRQ?	B
	Internal clock resolution switching	RES	RES?	A
MEMORY	File No./Directory mode switching	FIL	FIL?	–
	FD data recall	RCL	–	–
	FD data delete	DEL	–	–
	FD data save	SAV	–	–
	FD data resave	RSV	–	–
	Memory mode switch	MEM	MEM?	C
	FD format	FDF	–	–
	File contents search	–	FSH?	–
	Memory FD mode	–	FMD?	–
	FD access status	–	MAC?	–
	FD error message	–	FDE?	–
PATTERN	Pattern logic	LGC	LGC?	C
	Generation pattern switch	PTS	PTS?	B
	Zero Subst/PRBS stage	PTN	PTN?	C
	PRBS mark ratio	MRK	MRK?	C
	Alternate A/B display switch	ALT	ALT?	–
	Error insertion	EAD	EAD?	C
	Alternate A/ B loop times	LPT	LPT?	C
	Data length	DLN	DLN?	C
	Zero Subst length	ZLN	ZLN?	C
	Page number	ADR PAG	ADR? PAG?	– –
	Pattern bit	BIT	BIT?	–
	Pattern data preset (All pages ,All bits)	ALL	–	C
	Pattern data preset (1 page ,All bits)	PST	–	–
	Pattern Sync trigger position	PSP	PSP?	C

Table A.1.2-1 MP1761A/B/C device message (Cont'd)

Section	Function	Control Message	Data Request Message	Compatibility
PATTERN (Cont'd)	Page No./ Pattern sync trigger position display switch	PPD	PPD?	–
OUTPUT	Data output termination	DTM	DTM?	A
	Clock1 output termination	CTM	CTM?	A
	Offset reference value	OFS	OFS?	C
	Data output amplitude	DAP	DAP?	C
	XData output amplitude	NAP	NAP?	–
	Data output offset	DOS	DOS?	C
	XData output offset	NOS	NOS?	–
	Clock1 output amplitude	CAP	CAP?	C
	Clock1 output delay time	CDL	CDL?	A
	Clock1 output offset	COS	COS?	B
	Output ON/OFF	OON	OON?	C
	Data/XData display switch	DDS	DDS?	A
	Data/XData tracking	TRK	TRK?	C
	1/1 SPEED, 1/4 SPEED switch	SPD	SPD?	–
Front panel	Sync signal output selection	SOP	SOP?	C
Rear panel	Error insertion channel	ECH	ECH?	C
Function switch	Mark ratio and bit shift number	SFT	SFT?	C
	External error insertion	EEI	EEI?	C
	Alternate pattern A/B switch signal selection	APS	APS?	C
Other	Initialize	INI	–	A
	Pattern data input byte number	WRT	–	A
	Pattern data output byte number	–	RED?	A
	Internal timer setting	RTM	RTM?	–
	Internal synthesizer PLL	–	PLL?	–
	Power cut, recovery status	–	PWI?	–
	Delay status	–	DLY?	A
	Termination code selection	TRM	TRM?	A

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 A.1.3-1 MP1762A/C/D device message

Section	Function	Control Message	Data Request Message	Compatibility
INPUT	Data input threshold voltage	DTH	DTH?	A
	Eye margin measurement result (Threshold)	–	THM?	–
	Eye margin measurement result (Error ratio)	EYT	EYT?	C
	Eye margin measurement start	EST	EST?	–
	Eye margin measurement display	EME	EME?	–
	Clock input phase	CPA	CPA?	B
	Clock input polarity	CPL	CPL?	–
	Clock input termination voltage	CTM	CTM?	A
	Eye margin measurement result (Phase)	–	PHM?	–
	Data/XData tracking	DAC	DCD?	B
	Data input termination condition	DTC	DTC?	B
	Recovery frequency	RFQ	DRC?	C
	PLL reset	RPL	RPL?	–
	Input data select	DSL	DCD?	A
	Clock select	RRC	DRC?	A
	Single-ended/Differential select	DSD	DCD?	B
	Delay status	–	DLY?	A
	Data input termination voltage	DTM	DTM?	B
	Automatic phase threshold search	SRH	SRH?	–
MEMORY	File contents search	–	FSH?	–
	FD data delete	DEL	–	–
	FD error message	–	FDE?	–
	FD format	FDF	–	–
	FD data recall	RCL	–	–
	FD access status	–	MAC?	–
	File No./Directory mode switching	FIL	FIL?	–
	FD data save	SAV	–	–
	Memory FD mode	–	FMD?	–
	FD data resave	RSV	–	–
	Memory mode switch	MEM	MEM?	–

Table A.1.3-1 MP1762A/C/D device message (Cont'd)

Section	Function	Control Message	Data Request Message	Compatibility
PATTERN	Pattern logic	LGC	LGC?	A
	Generation pattern switch	PTS	PTS?	A
	Zero Subst/PRBS stage	PTN	PTN?	A
	PRBS mark ratio	MRK	MRK?	A
	Synchronous method	SYM	SYM?	A
	Display selection	DSP	DSP?	–
	Page number	PAG ADR	PAG? ADR?	– –
	Pattern bit	BIT	BIT?	–
	Frame length	FLN	FLN?	A
	Bit window pattern	CHM	CHM?	C
	Alternate A/B display switching	ALT	ALT?	–
	Alternate pattern A/B switch signal selection	APS	APS?	–
	Error analysis data	–	EAB?	–
	Error analysis page	EAP	EAP?	–
	Error analysis trigger	EAT	EAT?	–
	Number of pattern data input bytes	WRT	–	A
	Block window ON/OFF	MGE	MGE?	A
	Block window pattern	MGB	MGB?	C
	Bit window ON/OFF	MSE	MSE?	A
	Bit window page	MSK	MSK?	–
	Bit window preset (all pages, all bits)	HAL	–	C
	Bit window preset (1 page, all bits)	HPS	–	–
	Pattern data preset (1 pages, all bits)	PST	–	–
	Pattern data preset (all pages, all bits)	ALL	–	A
	Block window preset (1 page, all bits)	MPS	–	–
	Number of bytes of block window data output	–	MRD?	–
	Block window preset (all pages, all bits)	MAL	MAL?	–
	Number of bytes of block window data input	MWT	–	–
	Page/pattern sync trigger position display switch	PPD	PPD?	A
	Pattern sync position	PSP	PSP?	B
	Zero Subst length	ZLN	ZLN?	B
	Number of pattern data output bytes	–	RED?	–

Table A.1.3-1 MP1762A/C/D device message (Cont'd)

Section	Function	Control Message	Data Request Message	Compatibility
MEASURE-MENT	Measurement result for clock count	–	CC?	A
	Clock Loss status	–	CLI?	A
	Intermediate result display	CUR	CUR?	A
	Data length	DLN	DLN?	A
	Measurement result display mode	DMS	DMS?	–
	Measurement start/restart	STA	–	A
	Measurement stop	STO	–	A
	Automatic synchronous threshold	SYE	SYE?	A
	Automatic synchronization	SYN	SYN?	A
	Clear data from buffer	EDC	–	–
	Store data in buffer	EDS	–	–
	EI measurement result	–	EI?	A
	EI/EF interval time	EIT	EIT?	A
	1s measurement result	–	OSD?	A
	Clear intermediate data	IMC	–	–
	Output intermediate data	–	IMD?	A
	Store intermediate data in buffer	IMS	–	–
	Internal timer setting	RTM	RTM?	–
	Measurement mode	MOD	MOD?	A
	Measurement period setting	PRD	PRD?	A
	Measurement condition	–	MSR?	A
	Error detection status	–	ERS?	A
	Alarm measurement result	–	AMD?	–
	Error count result	–	EC?	A
	Clock frequency data	–	FRQ?	A
	Synchronous loss status	–	SLI?	A
	Synchronous signal output	SOP	SOP?	A
PATTERN	Real time /measurement time display switching	TIM	TIM?	A
	Measurement termination data	–	END?	B
	Error rate measurement result	–	ER?	A
Front panel	Printer function	PRN	PRN?	–
	Manual print	PSA	–	–
	Alarm monitor	ALM	ALM?	A
	Error monitor	MON	MON?	A
	Sync output selection	SOP	SOP?	A
Rear panel	GPIB2 address	GPA	GPA?	–
Rear panel function switch	Mark ratio and bit shift	SFT	SFT?	A
	Clock loss process	CLS	CLS?	A
	Sync loss process	SLS	SLS?	A
	Burst measurement mode	BST	BST?	A
	Error performance data threshold selection	ETH	ETH?	A

Table A.1.3-1 MP1762A/C/D device message (Cont'd)

Section	Function	Control Message	Data Request Message	Compatibility
Rear panel function switch (Cont'd)	Intermediate data calculation	CAL	CAL?	A
	Error detection mode	ETY	ETY?	–
	1s data print threshold	DOT	DOT?	–
	1s data print	OSC	OSC?	–
	Intermediate data print	ITM	ITM?	–
	Measurement interval time	ITV	ITV?	A
	%EFI measurement result	–	EFI?	A
	Error performance print	EPF	EPF?	–
	Data print format	FMT	FMT?	–
	Paper saving function	PSV	PSV?	–
	Threshold EI,EFI data print	THR	THR?	–
	Termination code selection	TRM	TRM?	A

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 A.1.4-1 MP1775A device message

Section	Function	Control Message	Data Request Message	Compatibility
Clock	Operating clock	CLK	CLK?	–
	Internal clock frequency resolution	RES	RES?	A
	Internal clock frequency	FRQ	FRQ?	B
Pattern	Output pattern logic	LGC	LGC?	A
	Output pattern	PTS	PTS?	C
	Zero Subst, PRBS pattern stages	PTN	PTN?	A
	PRGM data channel setting	PCH	PCH?	–
	Data length	DLN	DLN?	A
	Number of page	PAG ADR	PAG? ADR?	– –
	Pattern bits	BIT BIT	BIT? BIT?	– –
	Pattern data input bytes	WRT	–	A
	Pattern data output bytes	–	RED?	A
	Preset (all bits/channel)	ALL	–	A
	Preset (all bits/page)	PST	–	–
	0 continuous bit length	ZLN	ZLN?	C
	Error addition	EAD	EAD?	C
	Error addition channel	ECH	ECH?	A
Output	Output offset reference value	OFS	OFS?	A
	Data output termination selection	DTM	DTM?	–
	Data output display setting	SCD	SCD?	C
	Data output common setting	DCM	DCM?	–
	Data output amplitude	DAP	DAP?	–
	Data output offset	DOS	DOS?	–
	Clock output termination selection	CTM	CTM?	–
	Clock output display selection	SCC	SCC?	–
	Clock output common setting	CCM	CCM?	C
	Clock output amplitude	CAP	CAP?	–
	Clock output offset	COS	COS?	B
	Clock output delay	CDL	CDL?	A
	Data/Clock output ON/OFF	OON	OON?	C
	Data Cross point	DCP	DCP?	C
	Clock Duty	CDT	CDT?	A
	Synchronous output selection	SOP	SOP?	C

Table A.1.4-1 MP1775A device message (Cont'd)

Section	Function	Control Message	Data Request Message	Compatibility
Memory	Memory function selection	MEM	MEM?	–
	File No./ Directory mode selection	FIL	FIL?	–
	Data recall	RCL	–	–
	Data save	SAV	–	–
	Data resave	RSV	–	–
	Data delete	DEL	–	–
	FD format	FDF	–	–
	File content search	–	FSH?	–
	FD access status	–	MAC?	–
	FD error message	–	FDE?	–
Other	Initialize	INI	–	A
	Internal timer setting	RTM	RTM?	–
	PLL lock state	–	PLL?	–
	Delay state	–	DLY?	A
	Response data termination code	TRM	TRM?	A
	END event status enable register	ESE1	ESE1?	C
	END event status register	–	ESR1?	C
	ERROR event status enable register	ESE2	ESE2?	C
	ERROR event status register	–	ESR2?	C

A.1.5 Compatibility with MP1776A

This section describes the compatibility between the MP1800A or the MT1810A commands and the MP1776A Error Detector

Table A.1.5-1 MP1776A SCPI SENSe Subsystem

Commands	Compatibility
:SENSe1|2|3|4:PATTern:IMODe <mode>	A
:SENSe1|2|3|4:PATTern:IMODe?	A
:SENSe1|2|3|4:PATTern:TYPE <type>	B
:SENSe1|2|3|4:PATTern:TYPE?	B
:SENSe1|2|3|4:PATTern:PRBS:LOGic <logic>	–
:SENSe1|2|3|4:PATTern:PRBS:LOGic?	–
:SENSe1|2|3|4:PATTern:ZSUBstitute:LENGth <numeric>	A
:SENSe1|2|3|4:PATTern:ZSUBstitute:LENGth?	A
:SENSe1|2|3|4:PATTern:ZSUBstitute:ZLENgth <numeric>	B
:SENSe1|2|3|4:PATTern:ZSUBstitute:ZLENgth?	B
:SENSe1|2|3|4:PATTern:ZSUBstitute:LOGic <logic>	–
:SENSe1|2|3|4:PATTern:ZSUBstitute:LOGic?	–
:SENSe1|2|3|4:PATTern:DATA:WHOLe <start>,<end>,<data>	A
:SENSe1|2|3|4:PATTern:DATA:WHOLe? <start>	A
:SENSe1|2|3|4:PATTern:BDATA:WHOLe <start>,<end>,<bdata>	A
:SENSe1|2|3|4:PATTern:BDATA:WHOLe? <start>	A
:SENSe1|2|3|4:PATTern:PROGram:LENGth <numeric>	–
:SENSe1|2|3|4:PATTern:PROGram:LENGth?	–
:SENSe1|2|3|4:PATTern:PROGram:LOGic <logic>	–
:SENSe1|2|3|4:PATTern:PROGram:LOGic?	–
:SENSe1|2|3|4:PATTern:PROGram:DATA <page>,<start>,<data>	–
:SENSe1|2|3|4:PATTern:PROGram:DATA ? <page>	–
:SENSe1|2|3|4:PATTern:PROGram:FILL <range>,<page>,<data>	–
:SENSe1|2|3|4:PATTern:SYNC:ASYNc <boolean>	A
:SENSe1|2|3|4:PATTern:SYNC:ASYNc?	A
:SENSe1|2|3|4:PATTern:SYNC:THReshold <thre>	A
:SENSe1|2|3|4:PATTern:SYNC:THReshold?	A
:SENSe1|2|3|4:PATTern:SYNC:PSMode <mode>	A
:SENSe1|2|3|4:PATTern:SYNC:PSMode?	A
:SENSe1|2|3|4:PATTern:SYNC:FLENgth <numeric>	A
:SENSe1|2|3|4:PATTern:SYNC:FLENgth?	A
:SENSe1|2|3|4:MEASure:STARt	A
:SENSe1|2|3|4:MEASure:STOP	A
:SENSe1|2|3|4:MEASure:EALarm:MODE <mode>	A
:SENSe1|2|3|4:MEASure:EALarm:MODE?	A
:SENSe1|2|3|4:MEASure:EALarm:PERiod <d>,<h>,<m>,<s>	A
:SENSe1|2|3|4:MEASure:EALarm:PERiod?	A
:SENSe1|2|3|4:MEASure:EALarm:ERRor:MODE <mode>	–
:SENSe1|2|3|4:MEASure:EALarm:ERRor:MODE?	–
:SENSe1|2|3|4:MEASure:EALarm:STARt?	–

Table A.1.5-1 MP1776A SCPI SENSe Subsystem (Cont'd)

Commands	Compatibility
:SENSe1|2|3|4:MEASure:EALarm:STOP?	–
:SENSe1|2|3|4:MEASure:EALarm:STATe?	–
:SENSe1|2|3|4:MEASure:EALarm:ELAPsed?	–
:SENSe1|2|3|4:MEASure:EALarm:TIMed?	–
:SENSe1|2|3|4:MEASure:EALarm:ITIMe?	–
:SENSe1|2|3|4:MEASure:EALarm:ALARm? <alarm>	–
:SENSe1|2|3|4:MEASure:ASEarch:MODE <mode>	–
:SENSe1|2|3|4:MEASure:ASEarch:MODE?	–
:SENSe1|2|3|4:MEASure:ASEarch:SET <boolean>	–
:SENSe1|2|3|4:MEASure:ASEarch:SET?	–
:SENSe1|2|3|4:MEASure:ASEarch:STARt	A
:SENSe1|2|3|4:MEASure:ASEarch:STOP	A
:SENSe1|2|3|4:MEASure:ASEarch:STATe?	A

Table A.1.5-2 MP1776A INPut Subsystem

Commands	Compatibility
:INPut1|2|3|4:DATA:LEVel <level>	–
:INPut1|2|3|4:DATA:LEVel?	–
:INPut1|2|3|4:DATA:THReshold <numeric>	B
:INPut1|2|3|4:DATA:THReshold?	B
:INPut1|2|3|4:DATA:TERMination <term>	–
:INPut1|2|3|4:DATA:TERMination?	–
:INPut1|2|3|4:CLOCk:POLarity <pol>	–
:INPut1|2|3|4:CLOCk:POLarity?	–
:INPut1|2|3|4:CLOCk:DELay <numeric>	A
:INPut1|2|3|4:CLOCk:DELay?	A
:INPut1|2|3|4:CLOCk:TERMination <term>	–
:INPut1|2|3|4:CLOCk:TERMination?	–

Table A.1.5-3 MP1776A CALCulate Subsystem

Commands	Compatibility
:CALCulate1|2|3|4:EALarm:CLEValuation <boolean>	A
:CALCulate1|2|3|4:EALarm:CLEValuation?	A
:CALCulate1|2|3|4:EALarm:SLEValuation <boolean>	A
:CALCulate1|2|3|4:EALarm:SLEValuation?	A
:CALCulate1|2|3|4:EALarm:ERRor:INTerval <numeric>,<suffix>	A
:CALCulate1|2|3|4:EALarm:ERRor:INTerval?	A
:CALCulate1|2|3|4:EALarm:PERFormance:THReshold <string>	A
:CALCulate1|2|3|4:EALarm:PERFormance:THReshold?	A
:CALCulate1|2|3|4:DATA:EALarm? <data>,<string>:DATA	C
:CALCulate1|2|3|4:DATA:MONitor? <item>	B
:CALCulate1|2|3|4:DATA:STORe <data>	–
:CALCulate1|2|3|4:DATA:RBUFfer? <data>,<item>	–
:CALCulate1|2|3|4:DATA:CLEar <data>	–
:CALCulate1|2|3|4:DATA:OSECond? <ch>	–

Table A.1.5-4 MP1776A OUTPut Subsystem

Commands	Compatibility
:OUTPut1|2|3|4:SYNC:SOURce <source>	B
:OUTPut1|2|3|4:SYNC:SOURce?	B

Table A.1.5-5 MP1776A INSTrument Subsystem

Commands	Compatibility
:INSTrument1|2|3|4:COUPle <mode>	–
:INSTrument1|2|3|4:COUPle?	–

Table A.1.5-6 MP1776A DISPlay Subsystem

Commands	Compatibility
:DISPlay:WINDow:TILE	–
:DISPlay:WINDow:CASCade	–
:DISPlay:WINDow:OPEN <disp>	–
:DISPlay:WINDow:OPEN?	–
:DISPlay:WINDow:CLOSe <disp>	–
:DISPlay:MFRame[:NAME] <name>	–
:DISPlay:MFRame[:NAME]?	–
:DISPlay:RESult:TIME <type>	A
:DISPlay:RESult:TIME?	A
:DISPlay:RESult:EALarm:MODE <mode>	B
:DISPlay:RESult:EALarm:MODE?	B
:DISPlay:RESult:EALarm:CH1|:CH2|:CH3|:CH4:ITEM <item>	–
:DISPlay:RESult:EALarm:CH1|:CH2|:CH3|:CH4:ITEM?	–
:DISPlay:RESult:EALarm:CH1|:CH2|:CH3|:CH4:PITem <pitem>	–
:DISPlay:RESult:EALarm:CH1|:CH2|:CH3|:CH4:PITem?	–
:DISPlay:RESult:EALarm:CH1|:CH2|:CH3|:CH4:TITem <titem>	–
:DISPlay:RESult:EALarm:CH1|:CH2|:CH3|:CH4:TITem?	–
:DISPlay:RESult:EALarm:HRESet	A

Table A.1.5-7 MP1776A SYSTem Subsystem

Commands	Compatibility
SYSTem:BEEPer:ERRor:SET <boolean>	A
SYSTem:BEEPer:ERRor:SET?	A
SYSTem:BEEPer:ERRor:CHANnel <ch>,<boolean>	–
SYSTem:BEEPer:ERRor:CHANnel? <ch>	–
SYSTem:BEEPer:ALARm:SET <boolean>	A
SYSTem:BEEPer:ALARm:SET?	A
SYSTem:BEEPer:ALARm:CHANnel <ch>,<boolean>	–
SYSTem:BEEPer:ALARm:CHANnel? <ch>	–
SYSTem:BEEPer:SYSTem:SET <boolean>	A
SYSTem:BEEPer:SYSTem:SET?	A
SYSTem:BEEPer:SYSTem:TYPE <type>,<boolean>	–
SYSTem:BEEPer:SYSTem:TYPE?	–
SYSTem:DATE <year>,<month>,<day>	–
SYSTem:DATE?	–
SYSTem:TIME <hour>,<min>,<sec>	–
SYSTem:TIME?	–
SYSTem:ERRor?	A
SYSTem:VERSion?	A
SYSTem:ORGanization:HARDware?	C
SYSTem:ORGanization:SOFTware?	–
SYSTem:MMEMory:RECall <type>,<file_name>,<ch>	C
SYSTem:MMEMory:STORe <type>,<file_name>,<ch>	C
SYSTem:MMEMory:STORe:CATalog? <drv_dir>,<type>	–
SYSTem:MEMory:INITialize	A
SYSTem:PRINt:COPY	A
SYSTem:PRINt:EALarm <item>	C
SYSTem:PRINt:PATTern <ch>,<form>,<start>,<end>	–
SYSTem:PRINt:CANCel	–

Table A.1.5-8 MP1776A STATus Subsystem

Commands	Compatibility
STATus:PRESet	A
STATus:OPERation[:EVENt]?	C
STATus:OPERation:CONDtion?	C
STATus:OPERation:ENABle <numeric>	C
STATus:OPERation:ENABle?	C
STATus:OPERation:PTRansition <numeric>	C
STATus:OPERation:PTRansition?	C
STATus:OPERation:NTRansition <numeric>	C
STATus:OPERation:NTRansition?	C
STATus:OPERation:INSTrument[:EVENt]?	–
STATus:OPERation:INSTrument:CONDtion?	–
STATus:OPERation:INSTrument:ENABle <numeric>	–
STATus:OPERation:INSTrument:ENABle?	–
STATus:OPERation:INSTrument:PTRansition <numeric>	–
STATus:OPERation:INSTrument:PTRansition?	–
STATus:OPERation:INSTrument:NTRansition <numeric>	–
STATus:OPERation:INSTrument:NTRansition?	–
STATus:QUEStionable[:EVENt]?	–
STATus:QUEStionable:ENABle <numeric>	–
STATus:QUEStionable:ENABle?	–
STATus:QUEStionable:PTRansition <numeric>	–
STATus:QUEStionable:PTRansition?	–
STATus:QUEStionable:NTRansition <numeric>	–
STATus:QUEStionable:NTRansition?	–
STATus:QUEStionable:MONitor[:EVENt]?	–
STATus:QUEStionable:MONitor:CONDtion?	–

Table A.1.5-8 MP1776A STATus Subsystem (Cont'd)

Commands	Compatibility
STATus:QUEStionable:MONitor:ENABle <numeric>	–
STATus:QUEStionable:MONitor:ENABle?	–
STATus:QUEStionable:MONitor:PTRansition <numeric>	–
STATus:QUEStionable:MONitor:PTRansition?	–
STATus:QUEStionable:MONitor:NTRansition <numeric>	–
STATus:QUEStionable:MONitor:NTRansition?	–
STATus:QUEStionable:MONitor:CH1|:CH2|:CH3|:CH4|:CH1_3|:CH2_4|:CH1_4[:EVENt]?	–
STATus:QUEStionable:MONitor:CH1|:CH2|:CH3|:CH4|:CH1_3|:CH2_4|:CH1_4:CONDtion?	–
STATus:QUEStionable:MONitor:CH1|:CH2|:CH3|:CH4|:CH1_3|:CH2_4|:CH1_4:ENABle <numeric>	–
STATus:QUEStionable:MONitor:CH1|:CH2|:CH3|:CH4|:CH1_3|:CH2_4|:CH1_4:ENABle?	–
STATus:QUEStionable:MONitor:CH1|:CH2|:CH3|:CH4|:CH1_3|:CH2_4|:CH1_4:PTRansition <numeric>	–
STATus:QUEStionable:MONitor:CH1|:CH2|:CH3|:CH4|:CH1_3|:CH2_4|:CH1_4:PTRansition?	–
STATus:QUEStionable:MONitor:CH1|:CH2|:CH3|:CH4|:CH1_3|:CH2_4|:CH1_4:NTRansition <numeric>	–
STATus:QUEStionable:MONitor:CH1|:CH2|:CH3|:CH4|:CH1_3|:CH2_4|:CH1_4:NTRansition?	–