

SuperH™ Family E10A Emulator

Additional Document for User's Manual

SH7630 E10A

HS7630KCM02HE

Renesas Microcomputer Development Environment System

SuperH™ Family / SH7600 Series

Specific Guide for the SH7630 E10A Emulator

User's Manual

Cautions

Keep safety first in your circuit designs!

1. Renesas Technology Corporation puts the maximum effort into making semiconductor products better and more reliable, but there is always the possibility that trouble may occur with them. Trouble with semiconductors may lead to personal injury, fire or property damage. Remember to give due consideration to safety when making your circuit designs, with appropriate measures such as (i) placement of substitutive, auxiliary circuits, (ii) use of nonflammable material or (iii) prevention against any malfunction or mishap.


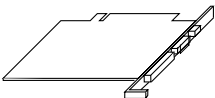
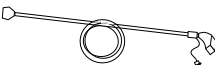
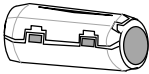
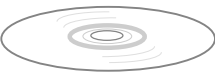
Notes regarding these materials

1. These materials are intended as a reference to assist our customers in the selection of the Renesas Technology Corporation product best suited to the customer's application; they do not convey any license under any intellectual property rights, or any other rights, belonging to Renesas Technology Corporation or a third party.
2. Renesas Technology Corporation assumes no responsibility for any damage, or infringement of any third-party's rights, originating in the use of any product data, diagrams, charts, programs, algorithms, or circuit application examples contained in these materials.
3. All information contained in these materials, including product data, diagrams, charts, programs and algorithms represents information on products at the time of publication of these materials, and are subject to change by Renesas Technology Corporation without notice due to product improvements or other reasons. It is therefore recommended that customers contact Renesas Technology Corporation or an authorized Renesas Technology Corporation product distributor for the latest product information before purchasing a product listed herein. The information described here may contain technical inaccuracies or typographical errors. Renesas Technology Corporation assumes no responsibility for any damage, liability, or other loss rising from these inaccuracies or errors. Please also pay attention to information published by Renesas Technology Corporation by various means, including the Renesas Technology Corporation Semiconductor home page (<http://www.renesas.com>).
4. When using any or all of the information contained in these materials, including product data, diagrams, charts, programs, and algorithms, please be sure to evaluate all information as a total system before making a final decision on the applicability of the information and products. Renesas Technology Corporation assumes no responsibility for any damage, liability or other loss resulting from the information contained herein.
5. Renesas Technology Corporation semiconductors are not designed or manufactured for use in a device or system that is used under circumstances in which human life is potentially at stake. Please contact Renesas Technology Corporation or an authorized Renesas Technology Corporation product distributor when considering the use of a product contained herein for any specific purposes, such as apparatus or systems for transportation, vehicular, medical, aerospace, nuclear, or undersea repeater use.
6. The prior written approval of Renesas Technology Corporation is necessary to reprint or reproduce in whole or in part these materials.
7. If these products or technologies are subject to the Japanese export control restrictions, they must be exported under a license from the Japanese government and cannot be imported into a country other than the approved destination. Any diversion or reexport contrary to the export control laws and regulations of Japan and/or the country of destination is prohibited.
8. Please contact Renesas Technology Corporation for further details on these materials or the products contained therein.

Contents

Section 1	Connecting the Emulator with the User System	1
1.1	Components of the Emulator	1
1.2	Connecting the E10A Emulator with the User System	4
1.3	Installing the H-UDI Port Connector on the User System	5
1.4	Pin Arrangement of the H-UDI Port Connector.....	5
1.5	Recommended Circuit between the H-UDI Port Connector and the MPU.....	8
1.5.1	Recommended Circuit (36-Pin Type)	8
1.5.2	Recommended Circuit (14-Pin Type)	10
Section 2	Specifications of the SH7630 E10A Emulator's Software	13
2.1	Differences between the SH7630 and the Emulator	13
2.2	Specific Functions for the SH7630 E10A Emulator	16
2.2.1	Emulator Driver Selection	16
2.2.2	Break Condition Functions	16
2.2.3	Trace Functions.....	18
2.2.4	Notes on Using the JTAG Clock (TCK) and AUD Clock (AUDCK)	24
2.2.5	Notes on Setting the [Breakpoint] Dialog Box	24
2.2.6	Notes on Setting the [Break Condition] Dialog Box and BREAKCONDITION_SET Command	25
2.2.7	Notes on Setting the UBC_MODE Command.....	26
2.2.8	Performance Measurement Function	26

Table 1.1 Components of the Emulator (HS7630KCM01H, HS7630KCM02H, HS7630KCI01H, or HS7630KCI02H)

Classification	Component	Appearance	Quantity	Remarks
Hard-ware	Card emulator		1	HS7630KCM01H (PCMCIA: 14-pin type): Depth: 85.6 mm, Width: 54.0 mm, Height: 5.0 mm, Mass: 27.0 g
		(PCMCIA)		HS7630KCM02H (PCMCIA: 36-pin type): Depth: 85.6 mm, Width: 54.0 mm, Height: 5.0 mm, Mass: 28.0 g
		or		HS7630KCI01H (PCI: 14-pin type): Depth: 122.0 mm, Width: 96.0 mm, Mass: 80.0 g
				HS7630KCI02H (PCI: 36-pin type): Depth: 122.0 mm, Width: 96.0 mm, Mass: 90.0 g
		(PCI)		
User system interface cable			1	HS7630KCM01H (PCMCIA: 14-pin type): Length: 80 cm, Mass: 45.0 g
				HS7630KCM02H (PCMCIA: 36-pin type): Length: 30 cm, Mass: 55.0 g
				HS7630KCI01H (PCI: 14-pin type): Length: 150 cm, Mass: 86.0 g
				HS7630KCI02H (PCI: 36-pin type): Length: 80 cm, Mass: 69.0 g
Ferrite core (connected with the user interface cable)		1	Countermeasure for EMI* (only for HS7630KCM02H and HS7630KCI02H)	
Soft-ware	SH7630 E10A emulator setup program, SH Family E10A Emulator User's Manual, and Specific Guide to the SH7630 E10A Emulator		1	HS7630KCM01SR, HS0005KCM01HJ, HS0005KCM01HE, HS7630KCM02HJ, and HS7630KCM02HE (provided on a CD-R)

Note: The EMI is an abbreviation of the Electrical Magnetic Interference.

For EMI countermeasure, use the ferrite core by connecting the user interface cable. When the user interface cable is connected with the emulator or user system, connect the ferrite core in the user system as shown in figure 1.1.

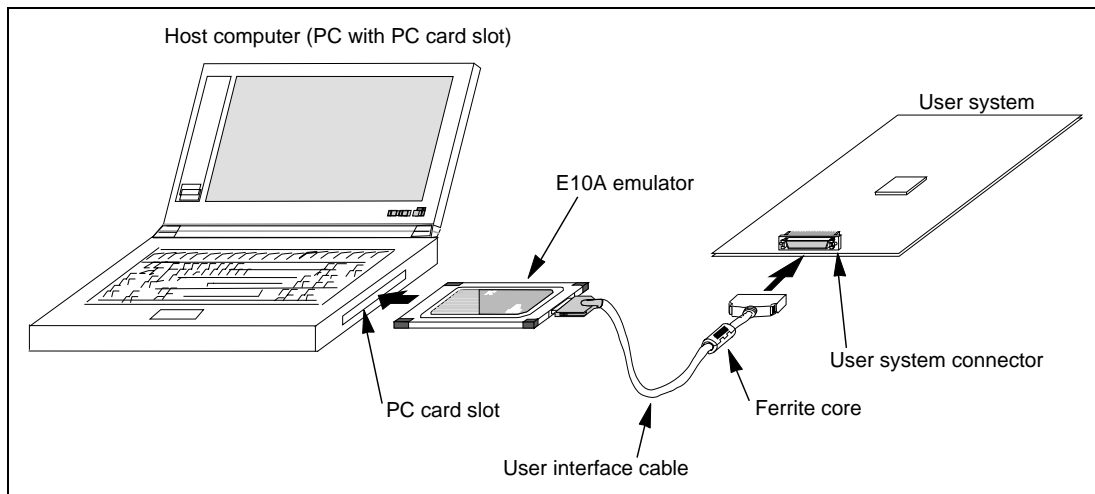


Figure 1.1 Connecting Ferrite Core

1.2 Connecting the E10A Emulator with the User System

To connect the E10A emulator (hereinafter referred to as the emulator), the H-UDI port connector must be installed on the user system to connect the user system interface cable. When designing the user system, refer to the recommended circuit between the H-UDI port connector and the MCU. In addition, read the E10A emulator user's manual and hardware manual for the related device.

Table 1.2 shows the type number of the E10A emulator, the corresponding connector type, and the use of AUD function.

Table 1.2 Type Number, AUD Function, and Connector Type

Type Number	Connector	AUD Function
HS7630KCM02H, HS7630KCI02H	36-pin connector	Available
HS7630KCM01H, HS7630KCI01H	14-pin connector	Not available

The H-UDI port connector has the 36-pin and 14-pin types as described below. Use them according to the purpose of the usage.

1. 36-pin type (with AUD function)

The AUD trace function is supported. A large amount of trace information can be acquired in realtime. The E10A emulator supports the window trace function that memory access (memory access address or memory access data) in the specified range can be acquired by tracing.

2. 14-pin type (without AUD function)

The AUD trace function cannot be used because only the H-UDI function is supported. For tracing, only the internal trace function is supported. Since the 14-pin type connector is smaller than the 36-pin type (1/2.5), the area where the connector is installed on the user system can be reduced.

1.3 Installing the H-UDI Port Connector on the User System

Table 1.3 shows the recommended H-UDI port connectors for the emulator.

Table 1.3 Recommended H-UDI Port Connectors

Connector	Type Number	Manufacturer	Specifications
36-pin connector	DX10M-36S	Hirose Electric Co., Ltd.	Screw type
	DX10M-36SE, DX10G1M-36SE		Lock-pin type
14-pin connector	2514-6002	Minnesota Mining & Manufacturing Ltd.	14-pin straight type

Note: When the 36-pin connector is used, do not connect any components under the H-UDI connector. When the 14-pin connector is used, do not install any components within 3 mm of the H-UDI port connector.

1.4 Pin Arrangement of the H-UDI Port Connector

Figures 1.2 and 1.3 show the pin arrangement of the 36-pin and 14-pin H-UDI port connectors, respectively.

Note: Note that the pin number assignment of the H-UDI port connector shown below differs from that of the connector manufacturer.

Pin No.	Signal	Input/ Output *1	SH7630 Pin No.	Note	Pin No.	Signal	Input/ Output *1	SH7630 Pin No.	Note
1	AUDCK	Output	85		19	TMS	Input	28	
2	GND	—			20	GND	—		
3	AUDATA0	Output	86		21 ^{*2}	/TRST	Input	24	
4	GND	—			22	GND	—		
5	AUDATA1	Output	87		23	TDI	Input	26	
6	GND	—			24	GND	—		
7	AUDATA2	Output	90		25	TDO	Output	25	
8	GND	—			26	GND	—		
9	AUDATA3	Output	91		27 ^{*2}	/ASEBRKAK	Output	7	
10	GND	—			28	GND	—		
11 ^{*2}	/AUDSYNC	Output	92		29	NC	—		Pulled-up
12	GND	—			30	GND	—		
13	NC	—			31 ^{*2}	/RESETP	Output	141	User reset
14	GND	—			32	GND	—		
15	NC	—			33 ^{*3}	GND	Output		
16	GND	—			34	GND	—		
17	TCK	Input	27		35	NC	—		
18	GND	—			36	GND	—		

- Notes: 1. Input to or output from the user system.
2. The slash (/) means that the signal is active-low.
3. The emulator monitors the GND signal of the user system and detects whether or not the user system is connected.

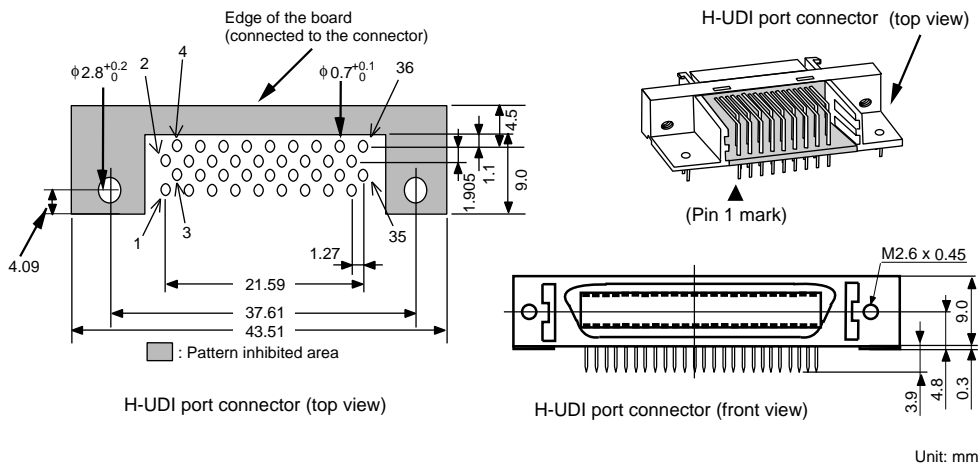


Figure 1.2 Pin Arrangement of the H-UDI Port Connector (36 Pins)

Pin No.	Signal	Input/ Output*1	SH7630 Pin No.	Note
1	TCK	Input	27	
2*2	/TRST	Input	24	
3	TDO	Output	25	
4*2	/ASEBRKAK	Output	7	
5	TMS	Input	28	
6	TDI	Input	26	
7*2	/RESETP	Output	141	
11	Not connected	—	—	Pulled-up
8 to 10	GND	—	—	
12 to 13		—	—	
14*3	GND	Output	—	

- Notes: 1. Input to or output from the user system.
2. The slash (/) means that the signal is active-low.
3. The emulator monitors the GND signal of the user system and detects whether the user system is connected or not.

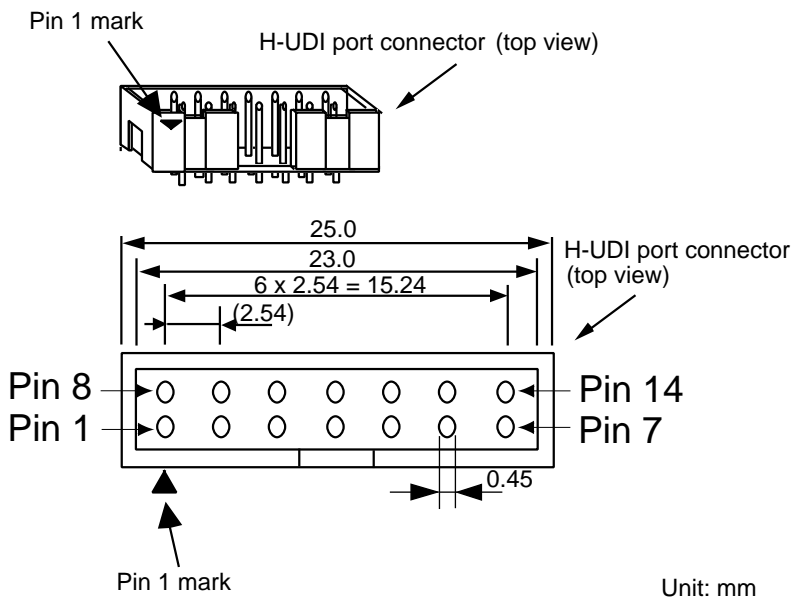


Figure 1.3 Pin Arrangement of the H-UDI Port Connector (14 Pins)

1.5 Recommended Circuit between the H-UDI Port Connector and the MPU

1.5.1 Recommended Circuit (36-Pin Type)

Figure 1.4 shows a recommended circuit between the H-UDI port connector (36 pins) and the MPU.

- Notes:
1. Do not connect anything to the N.C. pin of the H-UDI port connector.
 2. The processing of the /ASEMD0 pin differs depending on whether the emulator is used or not. As the emulator does not control this pin, it must be controlled by a switch on the board.
 - (1) When the emulator is used: /ASEMD0 = low (ASE mode)
 - (2) When the emulator is not used: /ASEMD0 = high (normal mode)
 3. The reset signal in the user system is input to the /RESETP pin of the MPU. Connect this signal to the H-UDI port connector as the output from the user system.
 4. When a joined resistance is used for pull-up, it may be affected by a noise. Separate TCK from other resistances.
 5. When the emulator is in use, pin 29 of the H-UDI connector must be pulled up by a resistance of several kilo-ohms.
 6. The pattern between the H-UDI connector and the MPU must be as short as possible. Do not connect the signal lines to other components on the board.
 7. The resistance values shown in figure 1.4 are recommended.
 8. For the pin processing in cases where the emulator is not used, refer to the hardware manual of the related device.

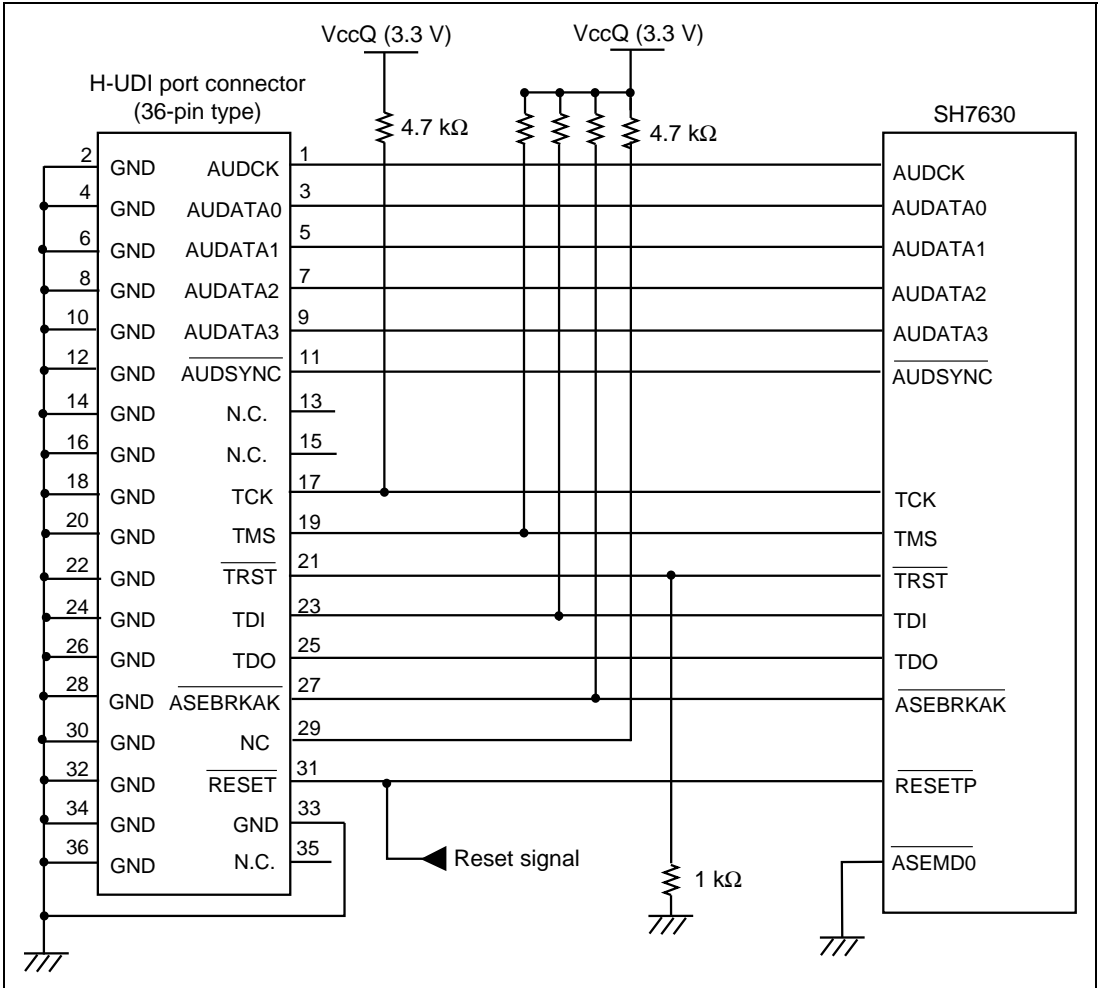


Figure 1.4 Recommended Circuit for Connection between the H-UDI Port Connector and MPU (36-Pin Type)

1.5.2 Recommended Circuit (14-Pin Type)

Figure 1.5 shows a recommended circuit between the H-UDI port connector and the MPU.

- Notes:
1. Do not connect anything to the N.C. pin of the H-UDI port connector.
 2. The processing of the /ASEMD0 pin differs depending on whether the emulator is used or not. As the emulator does not control this pin, it must be controlled by a switch on the board.
 - (1) When the emulator is used: /ASEMD0 = low (ASE mode)
 - (2) When the emulator is not used: /ASEMD0 = high (normal mode)
 3. The reset signal in the user system is input to the /RESETP pin of the MPU. Connect this signal to the H-UDI port connector as the output from the user system.
 4. When a joined resistance is used for pull-up, it may be affected by a noise. Separate TCK from other resistances.
 5. When the emulator is in use, pin 11 of the H-UDI connector must be pulled up by a resistance of several kilo-ohms.
 6. The pattern between the H-UDI connector and the MPU must be as short as possible. Do not connect the signal lines to other components on the board.
 7. The resistance values shown in figure 1.5 are recommended.
 8. For the pin processing in cases where the emulator is not used, refer to the hardware manual of the related device.

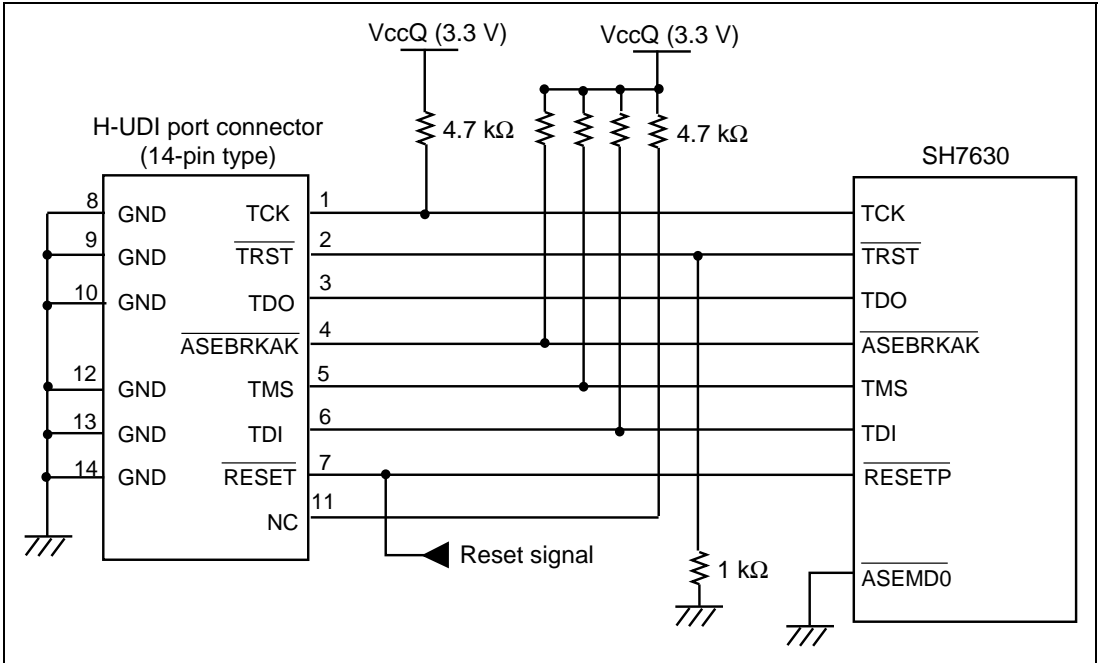


Figure 1.5 Recommended Circuit for Connection between the H-UDI Port Connector and MPU (14-Pin Type)

Section 2 Specifications of the SH7630 E10A Emulator's Software

2.1 Differences between the SH7630 and the Emulator

- When the emulator system is initiated, it initializes the general registers and part of the control registers as shown in table 2.1. The initial values of the actual SH7630 registers are undefined.

Table 2.1 Register Initial Values at Emulator Power-On

Register	Emulator at Power-on
R0 to R14	H'00000000
R15 (SP)	H'A0000000
R0_BANK to R7_BANK	H'00000000
PC	H'A0000000
SR	H'700000F0
GBR	H'00000000
VBR	H'00000000
MACH	H'00000000
MACL	H'00000000
PR	H'00000000
SPC	H'00000000
SSR	H'000000F0

- The emulator uses the H-UDI; do not access the H-UDI.
- Low-Power States (Sleep, Software Standby, and Module Standby)

For low-power consumption, the SH7630 has sleep, software standby, and module standby states.

The sleep, software standby, and module standby states are switched using the SLEEP instruction. When the emulator is used, only the sleep state can be cleared with either the normal clearing function or with the [Stop] button, and a break will occur.

Note: The memory must not be accessed or modified in sleep state.

- Reset Signals

The SH7630 reset signals are only valid during emulation started with clicking the GO or STEP-type button. If these signals are input from the user system in command input wait state, they are not sent to the SH7630.

Note: Do not break the user program when the /RESETP and /WAIT signals are being low. A TIMEOUT error will occur. If the /WAIT signal is fixed to low during break, a TIMEOUT error will occur at memory access.

- Direct Memory Access Controller (DMAC)

The DMAC operates even when the emulator is used. When a data transfer request is generated, the DMAC executes DMA transfer.

- Memory Access during User Program Execution

When a memory is accessed from the memory window, etc. during user program execution, the user program is resumed after it has stopped in the E10A emulator to access the memory. Therefore, realtime emulation cannot be performed.

The stopping time of the user program is as follows:

Environment:

Host computer: 650 MHz (Pentium® III)

SH7630: 60 MHz

JTAG clock: 3.75 MHz

When a one-byte memory is read from the command-line window, the stopping time will be about 20 ms.

- Memory Access during User Program Break

The emulator can download the program for the flash memory area. Other memory write operations are enabled for the RAM area. Therefore, an operation such as memory write or BREAKPOINT should be set only for the RAM area.

- Cache Operation during User Program Break

When cache is enabled, the emulator accesses the memory by the following methods:

- At memory write: Writes through the cache, then writes to the memory.
- At memory read: Does not change the cache write mode that has been set.

Therefore, when memory read or write is performed during user program break, the cache state will be changed.

- UBC

When [User] is specified in the [UBC mode] list box in the [Configuration] dialog box, the UBC can be used in the user program.

Do not use the UBC in the user program as it is used by the E10A emulator when [EML] is specified in the [UBC mode] list box in the [Configuration] dialog box.

- Loading Sessions

Information in [JTAG clock] of the [Configuration] dialog box cannot be recovered by loading sessions. Thus the TCK value will be as follows:

When HS7630KCI01H or HS7630KCI02H is used: TCK = 4.125 MHz

When HS7630KCM01H or HS7630KCM02H is used: TCK = 3.75 MHz

- [IO] Window

— Display and modification

Do not change values of the User Break Controller because it is used by the emulator.

For each Watchdog Timer register, there are two registers to be separately used for write and read operations.

Table 2.2 Watchdog Timer Register

Register Name	Usage	Register
WTCSR(W)	Write	Watchdog timer control/status register
WTCNT(W)	Write	Watchdog timer counter
WTCSR(R)	Read	Watchdog timer control/status register
WTCNT(R)	Read	Watchdog timer counter

- The watchdog timer operates only when the user program is executed. Do not change the value of the frequency change register in the [IO] window or [Memory] window.

- The internal I/O registers can be accessed from the [IO] window. However, note the following when accessing the SDMR register of the bus-state controller. Before accessing the SDMR register, specify addresses to be accessed in the I/O-register definition file (SH7630.IO) and then activate the HEW. For details on I/O-register definition file, refer to the Hitachi Debugging Interface User's Manual. Note that, however, the E10A emulator does not support the bit-field function described in the Hitachi Debugging Interface User's Manual.

- Verify

In the [IO] window, the verify function of the input value is disabled.

- Illegal Instructions

If illegal instructions are executed by STEP-type commands, the emulator cannot go to the next program counter.

2.2 Specific Functions for the SH7630 E10A Emulator

The SH7630 E10A emulator does not support the following function:

- MMU-related functions (The SH7630 does not mount the MMU.)
 - VPMAP-related command
 - Virtual and Physical specification in the [Configuration] window
 - Virtual and Physical specification on the command-line function
 - Virtual and Physical specification in the [Breakpoint] window
 - LDTLB instruction execution break function
 - MEMORYAREA_SET command

2.2.1 Emulator Driver Selection

Table 2.3 shows drivers which are selected in the [E10A Driver Details] dialog box.

Table 2.3 Type Number and Driver

Type Number	Driver
HS7630KCM01H	E10A PC Card Driver 7
HS7630KCM02H	E10A PC Card Driver 8
HS7630KCI01H	E10A PCI Card Driver 7
HS7630KCI02H	E10A PCI Card Driver 8

2.2.2 Break Condition Functions

In addition to BREAKPOINT functions, the emulator has Break Condition functions. Three types of conditions can be set under Break Condition 1, 2, 3. Table 2.4 lists these conditions of Break Condition.

Table 2.4 Types of Break Conditions

Break Condition Type	Description
Address bus condition (Address)	Breaks when the SH7630 address bus value or the program counter value matches the specified value.
Data bus condition (Data)	Breaks when the SH7630 data bus value matches the specified value. Byte, word, or longword can be specified as the access data size.
Bus state condition (Bus State)	<p>There are two bus state condition settings:</p> <p>Read/Write condition: Breaks when the SH7630 RD or RDWR signal level matches the specified condition.</p> <p>Bus state condition: Breaks when the operating state in an SH7630 bus cycle matches the specified condition.</p> <p>Types of buses that can be specified are listed below.</p> <ul style="list-style-type: none"> • L-bus (CPU-ALL): Indicates an instruction fetch and data access, including a hit to the cache memory. • L-bus (CPU-Data): Indicates a data access by the CPU, including a hit to the cache memory. • I-bus (CPU.DMA): Indicates a CPU cycle when the cache memory is not hit, and a data access by the DMA.
Internal I/O break condition	Breaks when the SH7630 accesses the internal I/O.
Count	Breaks when the conditions set are satisfied the specified number of times.

Note: When U-RAM is accessed from the P0 space, the I-bus must be selected, and when accessed from the P2 space, the L-bus must be selected. When cache fill cycle is acquired, the I-bus must be selected.

Table 2.5 lists the combinations of conditions that can be set under Break Condition 1, 2, 3.

Table 2.5 Dialog Boxes for Setting Break Conditions

Dialog Box	Type				
	Address Bus Condition (Address)	Data Bus Condition (Data)	Bus State Condition (Bus Status)	Count Condition (Count)	Internal I/O Break
[Break Condition 1] dialog box	O	O	O	O	X
[Break Condition 2] dialog box	O	X	O	X	X
[Break Condition 3] dialog box	X	X	X	X	O

Note: O: Can be set in the dialog box.
X: Cannot be set in the dialog box.

2.2.3 Trace Functions

The SH7630 E10A emulator supports the trace functions listed in table 2.6.

Table 2.6 Trace Functions

Function	Internal Trace	AUD Trace
Branch trace	Supported (eight branches)	Supported
Range memory access trace	Not supported	Supported
Software trace	Not supported	Supported

Table 2.7 shows the type numbers that the AUD function can be used.

Table 2.7 Type Number and AUD Function

Type Number	AUD Function
HS7630KCM01H, HS7630KCI01H	Not supported
HS7630KCM02H, HS7630KCI02H	Supported

AUD Trace Functions: This function is operational when the AUD pin of the device is connected to the emulator. Table 2.8 shows the AUD trace acquisition mode that can be set in each trace function.

Table 2.8 AUD Trace Acquisition Mode

Type	Mode	Description
Continuous trace occurs	Realtime trace	When the next branch occurs while the trace information is being output, the trace information being output is output but the next trace information is not output. The user program can be executed in realtime, but some trace information may be lost.
	Non realtime trace	When the next branch occurs while the trace information is being output, the CPU stops operations until the information is output. The user program is not executed in realtime.
Trace buffer full	Trace continue	This function overwrites the oldest trace information to store the latest trace information.
	Trace stop	After the trace buffer becomes full, the trace information is no longer acquired. (The user program is continuously executed.)

(a) Branch Trace Function

The branch source and destination addresses and their source lines are displayed.

Branch trace can be acquired by selecting the [Branch trace] check box in the [AUD function] group box of the [Trace mode] page.

The branch type can be selected in the [AUD Branch trace] page.

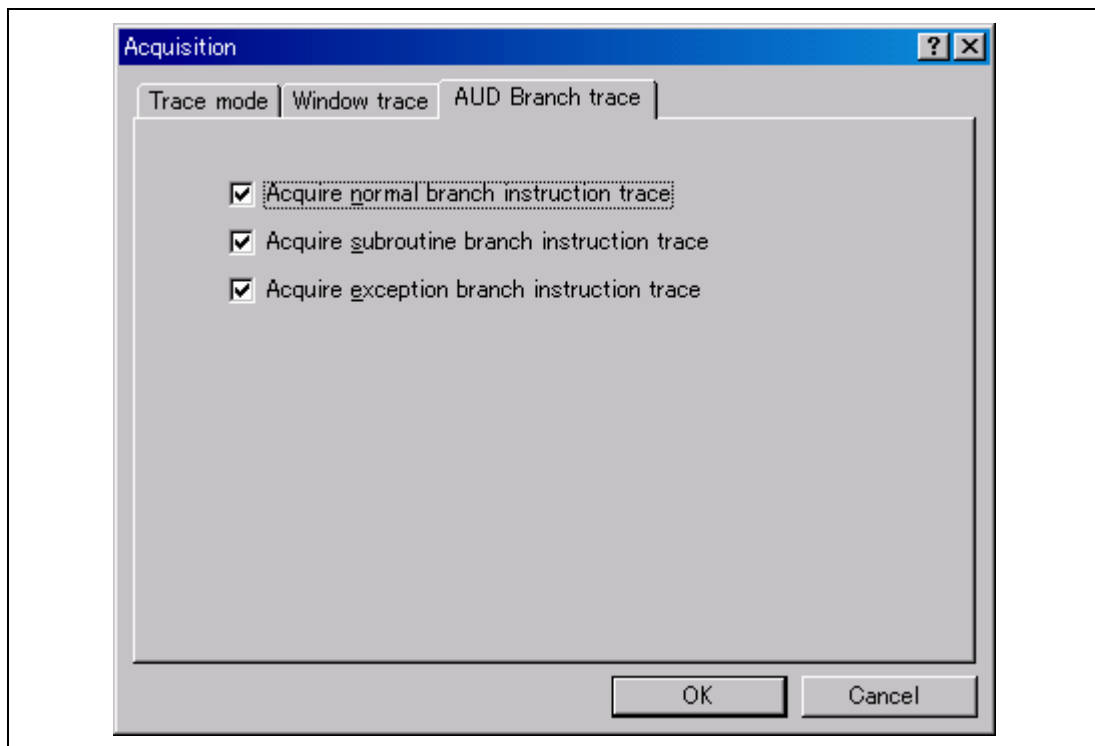


Figure 2.2 [AUD Branch trace] Page

(b) Window Trace Function

Memory access in the specified range can be acquired by trace.

Two memory ranges can be specified for channels A and B. The read, write, or read/write cycle can be selected as the bus cycle for trace acquisition.

[Setting Method]

- (i) Select the [Channel A] and [Channel B] check boxes in the [AUD function] group box of the [Trace mode] page. Each channel will become valid.
- (ii) Open the [Window trace] page and specify the bus cycle and memory range that are to be set for each channel.

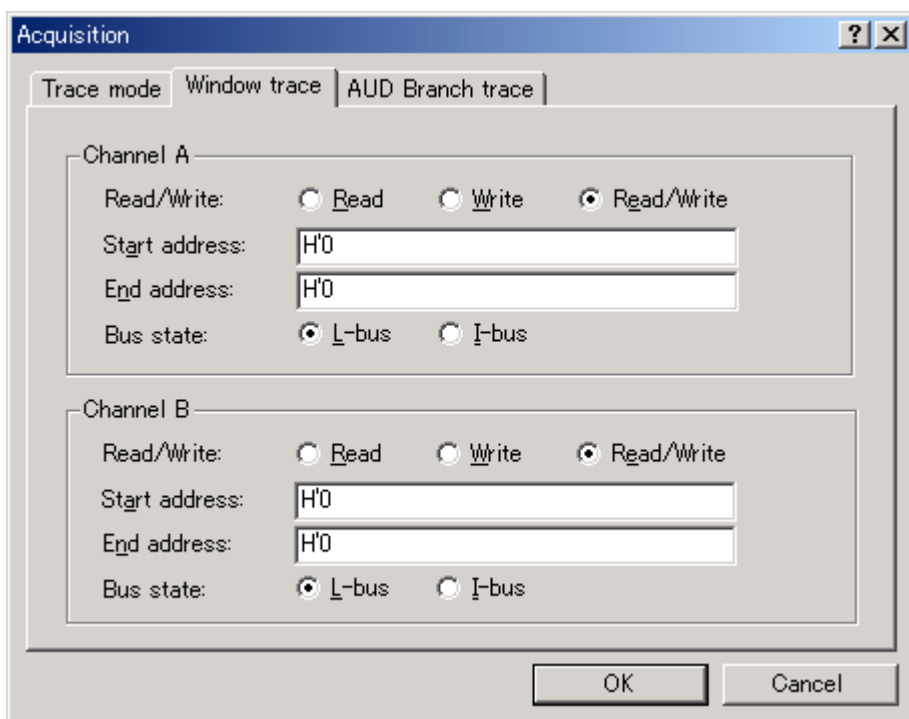


Figure 2.3 [Window trace] Page

Note: When the [L-bus] or [I-bus] radio button is selected, the following bus cycles will be traced.

L-bus: A bus cycle generated by the CPU is acquired. A bus cycle is also acquired when the cache has been hit.

I-bus: A bus cycle generated by the CPU or DMA is acquired. A bus cycle is not acquired when the cache has been hit. The address information acquired by the I-bus is 28 bits and the upper 4 bits are displayed as '*'. The source cannot be displayed in the [Trace] window.

When U-RAM is accessed from the P0 space, the I-bus must be selected, and when accessed from the P2 space, the L-bus must be selected. When a cache fill cycle is acquired, I-bus must be selected.

(c) Software Trace Function

Note: This function can be supported with SHC compiler V7.0 and later.

When a specific instruction is executed, the PC value at execution and the contents of one general register are acquired by trace. Describe the Trace(x) function (x is a variable name) to be compiled and linked beforehand. For details, refer to the SHC manual.

When the load module is loaded on the emulator and a valid software trace function is executed, the PC value that has executed the Trace(x) function, the general register value for x, and the source lines are displayed.

To activate the software trace function, select the [Software trace] check box in the [AUD function] group box of the [Trace mode] page.

Notes on AUD Trace:

1. When the trace display is performed during user program execution, the mnemonics, operands, or source is not displayed.
2. The AUD trace function outputs the differences between newly output branch source addresses and previously output branch source addresses. The window trace function outputs the differences between newly output addresses and previously output addresses. If the previous branch source address is the same as the upper 16 bits, the lower 16 bits are output. If it matches the upper 24 bits, the lower 8 bits are output. If it matches the upper 28 bits, the lower 4 bits are output.
The emulator regenerates the 32-bit address from these differences and displays it in the [Trace] window. If the emulator cannot display the 32-bit address, it displays the difference from the previously displayed 32-bit address.
3. If the 32-bit address cannot be displayed, the source line is not displayed.
4. In the SH7630 E10A emulator, when multiple loops are performed to reduce the number of AUD trace displays, only the IP counts up.
5. In the SH7630 E10A emulator, the maximum number of trace display pointers is as follows:
When HS7630KCM02H is used: D'8191 to -0
When HS7630KCI02H is used: D'32767 to -0
However, the maximum number of trace display pointers differs according to the AUD trace information to be output. Therefore, the above pointers cannot be always acquired.
6. When [User] is specified in the [UBC mode] list box in the [Configuration] window, the AUD trace is not acquired. In this case, exit the [Trace] window.
7. When the branch information for the odd address is acquired, it may be displayed as the software trace.
8. If a completion-type exception occurs during exception branch acquisition, the next address to the address in which an exception occurs is acquired.
9. For the AUD non-realtime trace, the written access may be executed again. If this is a problem on the user system, do not use the non-realtime trace.

Internal Trace Function: This function is activated by selecting the [Internal trace] radio button in the [Trace type] group box of the [Trace mode] page. See figure 2.1, [Trace mode] Page. The internal trace functions are also activated by selecting each check box on the [Branch trace] page.

Notes: 1. If an interrupt is generated at the program execution start or end, including a step execution, the emulator address may be acquired. In such a case, the following message will be displayed. Ignore this address because it is not a user program address.

*** EML ***

2. If a completion-type exception occurs during exception branch acquisition, the next address to the address in which an exception occurs is acquired.
3. Trace information cannot be acquired for the following branch instructions:
 - The BF and BT instructions whose displacement value is 0
 - Branch to H'A0000000 by reset
4. When [User] is specified in the [UBC mode] list box in the [Configuration] window, the internal trace is not acquired. In this case, exit the [Trace] window.

2.2.4 Notes on Using the JTAG Clock (TCK) and AUD Clock (AUDCK)

The JTAG clock (TCK) and AUD clock (AUDCK), which can be set in the [Configuration] window, have notes as follows.

Set the JTAG clock (TCK) frequency to less than the frequency of the SH7630 peripheral module clock (CKP).

Set the AUD clock (AUDCK) frequency 50 MHz or below for PCMCIA and PCI cards.

2.2.5 Notes on Setting the [Breakpoint] Dialog Box

1. When an odd address is set, the next lowest even address is used.
2. A BREAKPOINT is accomplished by replacing instructions of the specified address. Accordingly, it can be set only to the internal RAM area. However, a BREAKPOINT cannot be set to the following addresses:
 - An area other than CS0 to CS6 and the internal RAM
 - An instruction in which Break Condition 2 is satisfied
 - A slot instruction of a delayed branch instruction
3. During step execution, a BREAKPOINT is disabled.
4. Conditions set at Break Condition 2 are disabled when an instruction to which a BREAKPOINT has been set is executed. Do not set a BREAKPOINT to an instruction in which Break Condition 2 is satisfied.
5. When execution resumes from the address where a BREAKPOINT is specified, single-step execution is performed at the address before execution resumes. Therefore, realtime operation cannot be performed.

6. When a BREAKPOINT is set to the slot instruction of a delayed branch instruction, the PC value becomes an illegal value. Accordingly, do not set a BREAKPOINT to the slot instruction of a delayed branch instruction.
7. When a BREAKPOINT is set to the cacheable area, the cache block containing the BREAKPOINT address is filled immediately before and after user program execution.
8. Note on DSP repeat loop:
A BREAKPOINT is equal to a branch instruction. In some DSP repeat loops, branch instructions cannot be set. For these cases, do not set BREAKPOINTS. Refer to the hardware manual for details.
9. If an address of a BREAKPOINT cannot be correctly set in the ROM or flash memory area, a mark ● will be displayed in the [BP] area of the address on the [Editor] or [Disassembly] window by refreshing the [Memory] window, etc. after Go execution. However, no break will occur at this address. When the program halts with the break condition, the mark ● disappears.

2.2.6 Notes on Setting the [Break Condition] Dialog Box and BREAKCONDITION_SET Command

1. Break Condition 2 is disabled during step execution.
2. Break Condition 2 is disabled when an instruction to which a BREAKPOINT has been set is executed. Accordingly, do not set a BREAKPOINT to an instruction which satisfies Break Condition 2.
3. When a Break Condition is satisfied, emulation may stop after two or more instructions have been executed.
4. If a PC break address condition is set to the slot instruction after a delayed branch instruction, user program execution cannot be terminated before the slot instruction execution; execution stops before the branch destination instruction.
5. Break Condition 1,2 is used as the measurement range in the performance measurement function when [PA-1 start point] and [PA-1 end point] are displayed on the [Action] part in the [Break condition] sheet of the [Event] window. This applies when the Break Condition is displayed with the BREAKCONDITION_DISPLAY command in the command-line function. In this case, a break does not occur when Break Condition 1,2 is satisfied.
6. Note that a break occurs with a break satisfaction condition by an instruction that has been cancelled due to the generation of an exception.
7. Use the sequential break or count break with the L-bus condition. If such break is used with the I-bus condition, it will not operate correctly.
8. A break will not occur with the execution counts specified on the execution of the multi-step instruction.

2.2.7 Notes on Setting the UBC_MODE Command

In the [Configuration] window, if [User] is set while the [UBC mode] list box has been set, the STEP-type commands that use Break Condition 2 for implementation cannot be used.

2.2.8 Performance Measurement Function

The SH7630 E10A emulator supports the performance measurement function.

1. Setting the performance measurement conditions

To set the performance measurement conditions, use the [Performance Analysis] dialog box and the PERFORMANCE_SET command. When any line on the [Performance Analysis] window is clicked with the right mouse button, the popup menu is displayed and the [Performance Analysis] dialog box is displayed by selecting [Setting].

Note: For the command line syntax, refer to the online help.

— Specifying the measurement start/end conditions

The measurement start/end conditions are specified in [Mode] in the [Performance Analysis] dialog box. Three conditions can be set as shown in table 2.9.

Table 2.9 Conditions Specified in [Mode]

Item	Description
Normal break	Measurement is started by executing a program and ended when a break condition is satisfied.
Break Condition 1 -> 2	Measurement is started from the satisfaction of the condition set in Break Condition 1 to the satisfaction of the condition set in Break Condition 2.
Break Condition 2 -> 1	Measurement is started from the satisfaction of the condition set in Break Condition 2 to the satisfaction of the condition set in Break Condition 1.

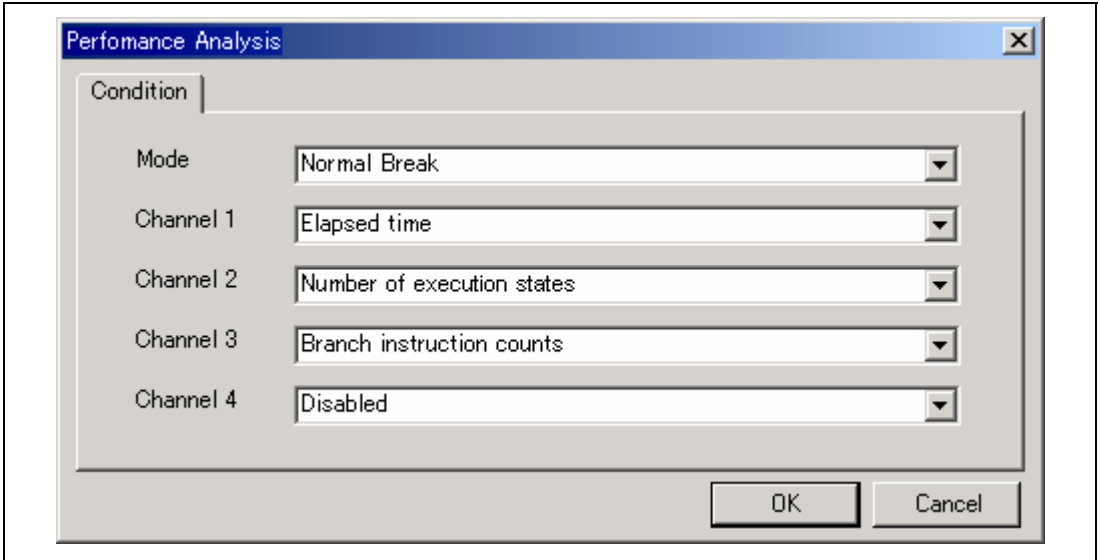


Figure 2.4 [Performance Analysis] Dialog Box

— Measurement range

One of the following ranges can be specified. This depends on the item selected for [Mode] in the [Performance Analysis] dialog box.

1. From the start to the end of the user program execution (When Normal Break is selected for [Mode])
2. From the satisfaction of the condition set in Break Condition 1 to the satisfaction of the condition set in Break Condition 2 (When Break condition 1->2 is selected for [Mode])
3. From the satisfaction of the condition set in Break Condition 2 to the satisfaction of the condition set in Break Condition 1 (When Break condition 2->1 is selected for [Mode])

(In the second and third ranges, [PA-1 start point] and [PA-1 end point] are displayed on the [Action] part in the [Break condition] sheet of the [Event] window.)

For measurement errors,

- The measured value includes errors.
- Error will occur before or after a break.

- Notes: 1. When the second and third ranges are specified, execute the user program after the measurement start condition is set to Break Condition 1 (or Break Condition 2) and the measurement end condition to Break Condition 2 (or Break Condition 1).

2. Step execution is not possible when Break condition 1->2 or Break condition 2->1 is selected for the PERFORMANCE_SET command or in [Mode] of the [Performance Analysis] dialog box.
3. When Break condition 1->2 or Break condition 2->1 is selected in [Mode] of the [Performance Analysis] dialog box, specify one or more items for measurement. When there is no item, the error message “Measurement item does not have specification. Please set up a measurement item.” will be displayed. When no item is specified for the PERFORMANCE_SET command, the settings of Break condition 1->2 or Break condition 2->1 will be an error.

— Measurement item

Items are measured with [Channel 1 to 4] in the [Performance Analysis] dialog box. Maximum four conditions can be specified at the same time. Table 2.10 shows the measurement items (Options in table 2.10 are parameters for <mode> of the PERFORMANCE_SET command. They are displayed for CONDITION in the [Performance Analysis] window).

Table 2.10 Measurement Item

Selected Name	Option
Disabled	None
Elapsed time	AC
Number of execution states	VS
Branch instruction counts	BT
Number of execution instructions	I
DSP-instruction execution counts	DI (Devices incorporating the DSP function can only be measured.)
Instruction/data conflict cycle	MAC
Other conflict cycles than instruction/data	OC
Exception/interrupt counts	EA
Data-TLB miss cycle	MTS (Devices incorporating the MMU function can only be measured.)
Instruction-TLB miss cycle	ITS (Devices incorporating the MMU function can only be measured.)
Interrupt counts	INT
Number of BL=1 instructions	BL1
Number of MD=1 instructions	MD1
Instruction cache-miss counts	IC
Data cache-miss counts	DC
Instruction fetch stall	IF
Data access stall	DA
Instruction cache-miss stall	ICS
Data cache-miss stall	DCS
Cacheable access stall	CS
X/Y-RAM access stall	XYX (Devices incorporating the X/Y memory can only be measured.)
URAM access stall	US (Devices incorporating the U memory can only be measured.)
Instruction/data access stall cycle	MA
Other access cycles than instruction/data	NMA
Non-cacheable area access cycle	NCC
Non-cacheable area instruction access cycle	NCI
Non-cacheable area data access cycle	NCD
Cacheable area access cycle	CC

Table 2.10 Measurement Item (cont)

Selected Name	Option
Cacheable area instruction access cycle	CIC
Cacheable area data access cycle	CDC
Access counts other than instruction/data	NAM
Non-cacheable area access counts	NCN
Non-cacheable area instruction access counts	NCIN
Non-cacheable area data access counts	NCDN
Cacheable area access counts	CN
Cacheable area instruction access counts	CIN
Cacheable area data access counts	CDN

Each measurement condition is also counted when conditions in table 2.11 are generated.

Table 2.11 Performance Measurement Conditions to be Counted

Measurement Condition	Notes
Cache-on counting	Accessing the non-cacheable area is counted less than the actual number of cycles and counts. Accessing the cacheable and U memory areas is counted more than the actual number of cycles and counts.
Branch count	The counter value is incremented by 2. This means that two cycles are valid for one branch.

- Notes: 1. In the non-realtime trace mode of the AUD trace, normal counting cannot be performed because the generation state of the stall or the execution cycle is changed.
2. Since the clock source of the counter is the CPU clock, counting also stops when the clock halts in the sleep mode.

2. Displaying the measured result

The measured result is displayed in the [Performance Analysis] window or the PERFORMANCE_ANALYSIS command with 32 bits.

Note: If a performance counter overflows as a result of measurement, “*****” will be displayed.

3. Initializing the measured result

To initialize the measured result, select [Initialize] from the popup menu in the [Performance Analysis] window or specify INIT with the PERFORMANCE_ANALYSIS command.

**SuperH™ Family E10A Emulator
Additional Document for User's Manual
Specific Guide for the SH7630 E10A Emulator**

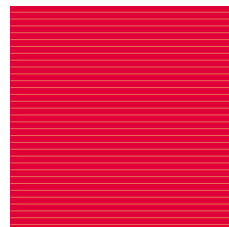
Publication Date: Rev.1.00, July 3, 2003

Published by: Sales Strategic Planning Div.
Renesas Technology Corp.

Edited by: Technical Documentation & Information Department
Renesas Kodaira Semiconductor Co., Ltd.

©2003 Renesas Technology Corp. All rights reserved. Printed in Japan.

SuperH™ Family E10A Emulator Additional Document for User's Manual



Renesas Technology Corp.

2-6-2, Ote-machi, Chiyoda-ku, Tokyo, 100-0004, Japan

REJ10B0015-0100H

Download from www.somanuals.com. All Manuals Search And Download.

Free Manuals Download Website

<http://myh66.com>

<http://usermanuals.us>

<http://www.somanuals.com>

<http://www.4manuals.cc>

<http://www.manual-lib.com>

<http://www.404manual.com>

<http://www.luxmanual.com>

<http://aubethermostatmanual.com>

Golf course search by state

<http://golfingnear.com>

Email search by domain

<http://emailbydomain.com>

Auto manuals search

<http://auto.somanuals.com>

TV manuals search

<http://tv.somanuals.com>