REJ10B0186-0100

Everywhere you imagine.

SuperH™ Family E10A Emulator

Additional Document for User's Manual

Specific Guide for the SH7710 E10A Emulator

Renesas Microcomputer Development Environment System

SuperH™ Family / SH7700 Series

SH7710 E10A HS7710KCM02HE

User's Manual

Rev.1.00
Revision Date: Mar. 01, 2005

Renesas Technology
www.renesas.com

Keep safety first in your circuit designs!

1. Renesas Technology Corp. 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 Corp. 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 Corp. or a third party.
2. Renesas Technology Corp. 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 Corp. without notice due to product improvements or other reasons. It is therefore recommended that customers contact Renesas Technology Corp. or an authorized Renesas Technology Corp. 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 Corp. 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 Corp. by various means, including the Renesas Technology Corp. 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 Corp. assumes no responsibility for any damage, liability or other loss resulting from the information contained herein.
5. Renesas Technology Corp. 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 Corp. or an authorized Renesas Technology Corp. 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 Corp. 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 Corp. for further details on these materials or the products contained therein.

Contents

Section 1 Connecting the Emulator with the User System

1.1 Components of the Emulator

The SH7710 E10A emulator supports the SH7710. Table 1.1 lists the components of the emulator.

Table 1.1 Components of the Emulator (HS7710KCM01H, HS7710KCM02H, HS7710KCI01H, or HS7710KCI02H)

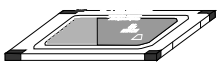
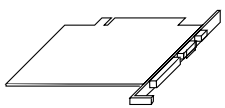
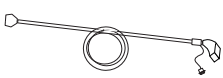
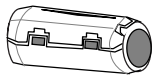
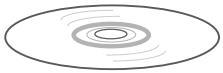
Classi-fication	Component	Appearance	Quan-tity	Remarks
Hard-ware	Card emulator	 (PCMCIA) or  (PCI)	1	HS7710KCM01H (PCMCIA: 14-pin type): Depth: 85.6 mm, Width: 54.0 mm, Height: 5.0 mm, Mass: 27.0 g HS7710KCM02H (PCMCIA: 36-pin type): Depth: 85.6 mm, Width: 54.0 mm, Height: 5.0 mm, Mass: 28.0 g HS7710KCI01H (PCI: 14-pin type): Depth: 144.0 mm, Width: 105.0 mm, Mass: 93.0 g HS7710KCI02H (PCI: 36-pin type): Depth: 122.0 mm, Width: 96.0 mm, Mass: 90.0 g
	User system interface cable		1	HS7710KCM01H (PCMCIA: 14-pin type): Length: 80 cm, Mass: 45.0 g HS7710KCM02H (PCMCIA: 36-pin type): Length: 30 cm, Mass: 55.0 g HS7710KCI01H (PCI: 14-pin type): Length: 150 cm, Mass: 86.0 g HS7710KCI02H (PCI: 36-pin type): Length: 80 cm, Mass: 69.0 g

Table 1.1 Components of the Emulator (HS7710KCM01H, HS7710KCM02H, HS7710KCI01H, or HS7710KCI02H) (cont)

Classi-fication	Component	Appearance	Quan-tity	Remarks
Hard-ware	Ferrite core (connected with the user interface cable)		1	Countermeasure for EMI* (only for HS7710KCM02H and HS7710KCI02H)
Soft-ware	SH7710 E10A emulator setup program, SuperH™ Family E10A Emulator User's Manual, and Specific Guide for the SH7710 E10A Emulator		1	HS7710KCM01SR, HS0005KCM01HJ, HS0005KCM01HE, HS7710KCM02HJ, and HS7710KCM02HE (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.

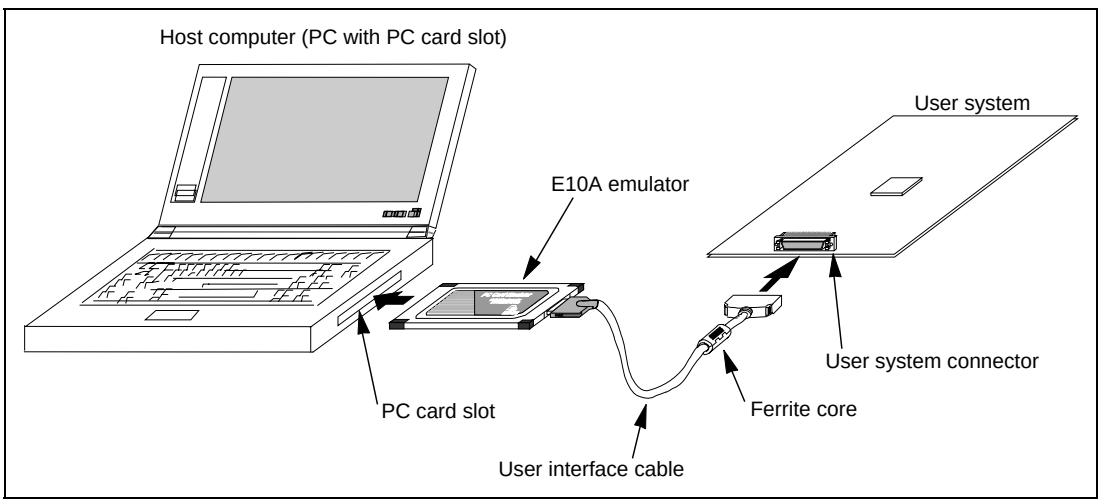


Figure 1.1 Connecting Ferrite Core

1.2 Connecting the 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 an example of recommended connection between the connector and the MPU shown in this manual. 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 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
HS7710KCM02H, HS7710KCI02H	36-pin connector	Available
HS7710KCM01H, HS7710KCI01H	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 window trace function is also supported for acquiring memory access in the specified range (memory access address or memory access data) 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 designing the 36-pin connector layout on the user board, do not connect any components under the H-UDI connector. When designing the 14-pin connector layout on the user board, do not place any components within 3 mm of the H-UDI port connector.

1.4 Pin Assignments of the H-UDI Port Connector

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

Note: Note that the pin number assignments of the H-UDI port connector shown below differ from those of the connector manufacturer.

Pin No.	Signal	Input/ Output*1	SH7710 Pin No.	Note	Pin No.	Signal	Input/ Output*1	SH7710 Pin No.	Note
1	AUDCK	Output	205		19	TMS	Input	199	
2	GND	——			20	GND	——		
3	AUDATA0	Output	213		21*2	/TRST	Input	201	
4	GND	——			22*5	(GND)	——		
5	AUDATA1	Output	212		23	TDI	Input	198	
6	GND	——			24	GND	——		
7	AUDATA2	Output	211		25	TDO	Output	200	
8	GND	——			26	GND	——		
9	AUDATA3	Output	210		27*2	/ASEBRKAK	Output	203	
10	GND	——			28	GND	——		
11*2	/AUDSYNC	Output	204		29*4	UVCC	Output		
12	GND	——			30	GND	——		
13	NC	——			31*2	/RESETP	Output	215	User reset
14	GND	——			32	GND	——		
15	NC	——			33*3	GND	Output		
16	GND	——			34	GND	——		
17	TCK	Input	202		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.

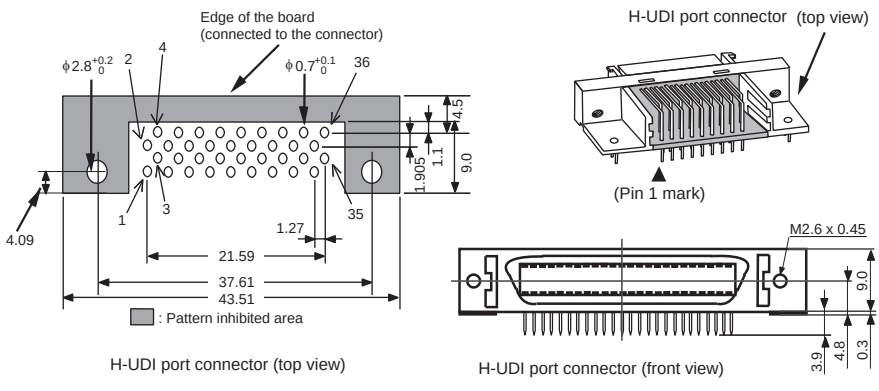


Figure 1.2 Pin Assignments of the H-UDI Port Connector (36 Pins)

Pin No.	Signal		Input/ Output*1	SH7710 Pin No.	Note
1	TCK		Input	202	
2	/TRST	*2	Input	201	
3	TDO		Output	200	
4	/ASEBRKAK	*2	Output	203	
5	TMS		Input	199	
6	TDI		Input	198	
7	/RESETP	*2	I/O	215	User reset
8	N.C.		—	—	
9	(GND)	*5	—	—	
11	UVCC	*4	Output	—	
10, 12, and 13	GND		—	—	
14	GND	*3	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 or not the user system is connected.

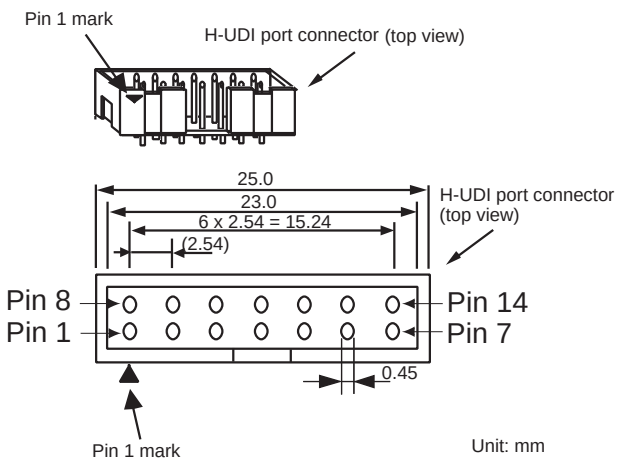


Figure 1.3 Pin Assignments of the H-UDI Port Connector (14 Pins)

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

1.5.1 Recommended Circuit (36-Pin Type)

Figure 1.5 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. pins 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
   (2) When the emulator is not used: /ASEMD0 = high
3. The reset signal in the user system is input to the /RESETP pin (pin 215) of the MPU. Connect this signal to the H-UDI port connector as the output from the user system.
4. When a network resistance is used for pull-up, it may be affected by a noise. Separate TCK from other resistances.
5. 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.
6. When the power of the emulator is turned on (make sure that the emulator is inserted into the host computer and the power supply is turned on), and the target system (such as the system mounting the target device) is connected to the emulator (the power supply of the board is turned off), the power-supply voltage of the target device (target system) may become higher around 1.2 V to 1.4 V than the normal voltage due to the leakage current from the emulator. This is because TMS and TRST are driven to high by the emulator; the leakage current occurs from a TMS line before starting the emulator software (HDI or HEW), and from TMS and TRST lines through the CPU after starting the emulator software. This phenomenon only occurs in the SuperH™ family E10A emulator.
   Although the CPU will not be degraded or damaged, a power-on reset may be disabled. The power-supply voltage raised by the leakage current can be reduced to 0.2 V by inserting a diode in the output pins (TMS and TRST) of the emulator. For the type (type number) and the inserting direction of the diode, see figure 1.4.

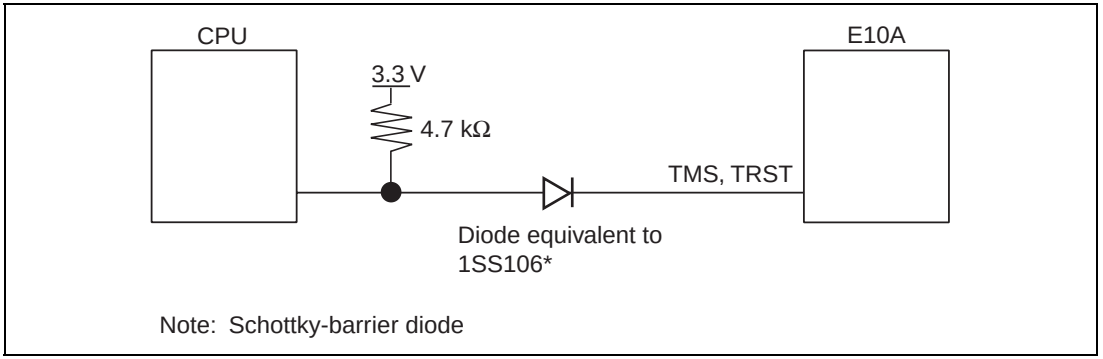


Figure 1.4 Countermeasure against the Leakage Current in the Emulator

The result above differs depending on the circuit and can only be used as a reference.

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.
9. For the AUDCK pin, guard the pattern between the H-UDI port connector and the MPU at GND level.

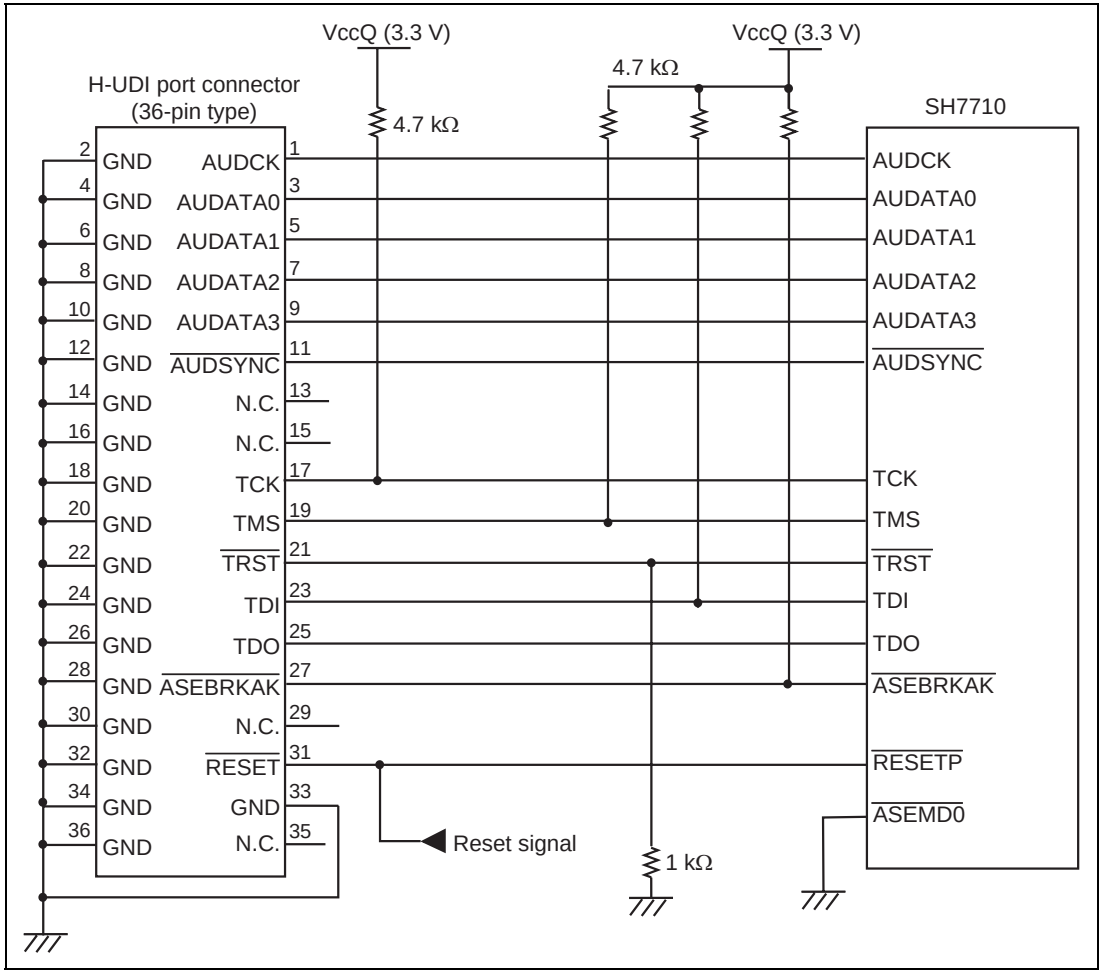


Figure 1.5 Recommended Circuit for Connection between the H-UDI Port Connector and MPU (36-Pin Type)

1.5.2 Recommended Circuit (14-Pin Type)

Figure 1.7 shows a recommended circuit between the H-UDI port connector and the MPU.

Notes: 1. Do not connect anything to the N.C. pins 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
   (2) When the emulator is not used: /ASEMD0 = high
3. The reset signal in the user system is input to the /RESETP pin (pin 215) of the MPU. Connect this signal to the H-UDI port connector as the output from the user system.
4. When a network resistance is used for pull-up, it may be affected by a noise. Separate TCK from other resistances.
5. 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.
6. When the power of the emulator is turned on (make sure that the emulator is inserted into the host computer and the power supply is turned on), and the target system (such as the system mounting the target device) is connected to the emulator (the power supply of the board is turned off), the power-supply voltage of the target device (target system) may become higher around 1.2 V to 1.4 V than the normal voltage due to the leakage current from the emulator. This is because TMS and TRST are driven to high by the emulator; the leakage current occurs from a TMS line before starting the emulator software (HDI or HEW), and from TMS and TRST lines through the CPU after starting the emulator software. This phenomenon only occurs in the SuperH™ family E10A emulator.
   Although the CPU will not be degraded or damaged, a power-on reset may be disabled. The power-supply voltage raised by the leakage current can be reduced to 0.2 V by inserting a diode in the output pins (TMS and TRST) of the emulator. For the type (type number) and the inserting direction of the diode, see figure 1.6.

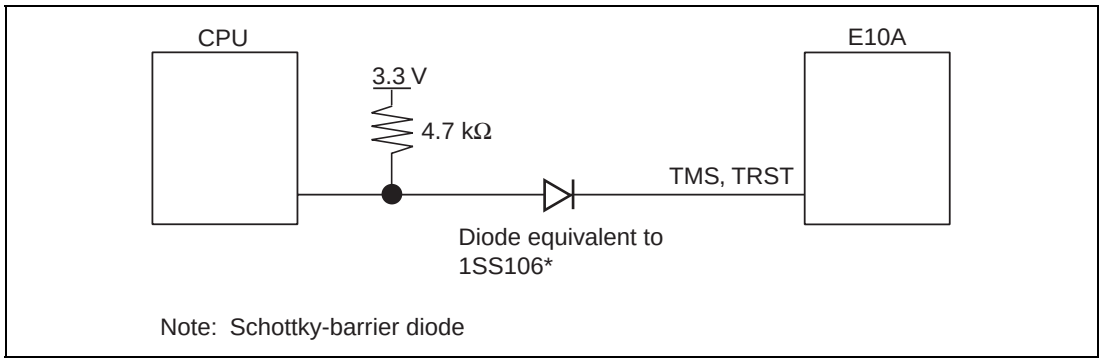


Figure 1.6 Countermeasure against the Leakage Current in the Emulator

The result above differs depending on the circuit and can only be used as a reference.

7. The resistance values shown in figure 1.7 are recommended.
8. For the pin processing in cases where the emulator is not used, refer to the hardware manual of the related device.

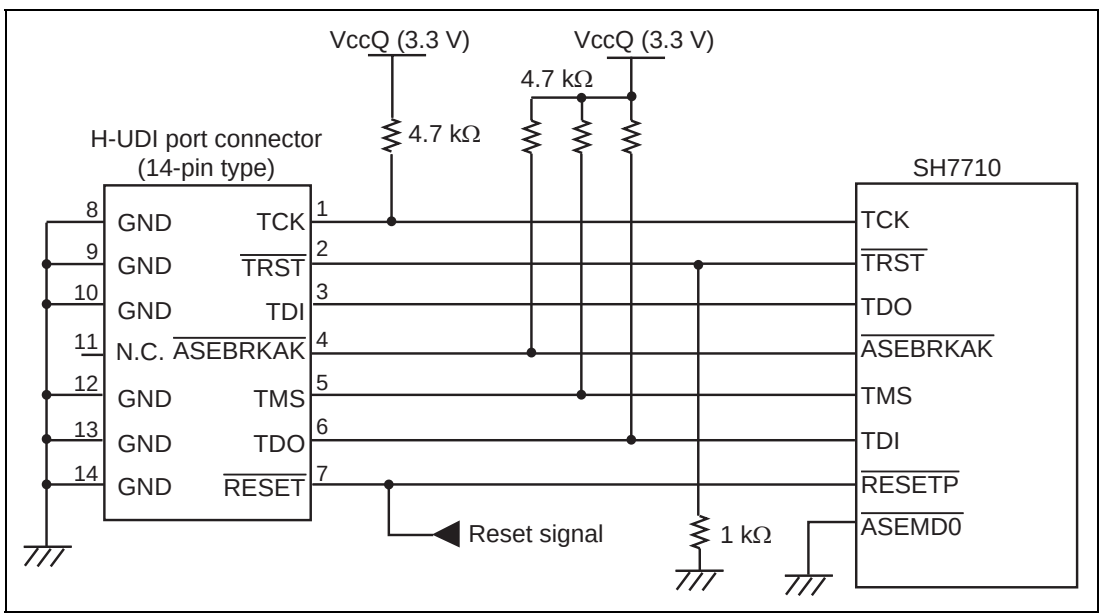


Figure 1.7 Recommended Circuit for Connection between the H-UDI Port Connector and MPU (14-Pin Type)

Section 2 Specifications of the SH7710 E10A Emulator's Software

2.1 Differences between the SH7710 and the Emulator

1. 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 SH7710 registers are undefined.

Table 2.1 Register Initial Values at Emulator Link Up

Register	Emulator at Link Up
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
RS	H'00000000
RE	H'00000000
MOD	H'00000000
A0G, A1G	H'00000000
A0, A1	H'00000000
X0, X1	H'00000000
Y0, Y1	H'00000000
M0, M1	H'00000000
DSR	H'00000000

2. The emulator uses the H-UDI; do not access the H-UDI.

3. Low-Power States (Sleep, Software Standby, and Module Standby)

   For low-power consumption, the SH7710 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.

4. Reset Signals

   The SH7710 reset signals are only valid during emulation started with clicking the GO or STEP-type button. If these signals are enabled on the user system in command input wait state, they are not sent to the SH7710.

Note: Do not break the user program when the /RESETP, /BREQ, or /WAIT signal is 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.

5. 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.

6. 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)
   SH7710: 60 MHz (CPU clock)
   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.

7. Memory Access during User Program Break

   The emulator can download the program for the flash memory area (refer to section 6.22, Download Function to the Flash Memory Area, in the Debugger Part of the SuperH™ Family E10A Emulator User's Manual). 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.

8. 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.

9. 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 emulator when [EML] is specified in the [UBC mode] list box in the [Configuration] dialog box.

10. Memory Access during Break

    In the enabled MMU, when a memory is accessed and a TLB error occurs during break, it can be selected whether the TLB exception is controlled or the program jumps to the user exception handler in [TLB Mode] in the [Configuration] dialog box. When [TLB miss exception is enable] is selected, a Communication Timeout error will occur if the TLB exception handler does not operate correctly. When [TLB miss exception is disable] is selected, the program does not jump to the TLB exception handler even if a TLB exception occurs. Therefore, if the TLB exception handler does not operate correctly, a Communication Timeout error will not occur but the memory contents may not be correctly displayed.

11. 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 HS7710KCI01H or HS7710KCI02H is used: TCK = 4.125 MHz

    — When HS7710KCM01H or HS7710KCM02H is used: TCK = 3.75 MHz

12. [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 (SH7710.IO) and then activate the HEW. After the I/O-register file is created, the MPU's specification may be changed. If each I/O register in the I/O-register definition file differs from addresses described in the hardware manual, change the I/O-register definition file according to the description in the hardware manual. The I/O-register definition file can be customized depending on its format. Note that, however, the E10A emulator does not support the bit-field function.

— Verify

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

13. 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 SH7710 E10A Emulator

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
HS7710KCM01H	E10A PC Card Driver 3
HS7710KCM02H	E10A PC Card Driver 4
HS7710KCI01H	E10A PCI Card Driver 3
HS7710KCI02H	E10A PCI Card Driver 4

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 SH7710 address bus value or the program counter value matches the specified value.
Data bus condition (Data)	Breaks when the SH7710 data bus value matches the specified value. Byte, word, or longword can be specified as the access data size.
X-Bus or Y-Bus condition (Address and data)	Breaks when the X-Bus or Y-Bus address bus or data bus matches the specified value.
Bus state condition (Bus State)	There are two bus state condition settings: Read/Write condition: Breaks when the SH7710 RD or RDWR signal level matches the specified condition. Bus state condition: Breaks when the operating state in an SH7710 bus cycle matches the specified condition. Types of buses that can be specified are listed below. • 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 SH7710 accesses the internal I/O.
LDTLB instruction break condition	Breaks when the SH7710 executes the LDTLB instruction.
Count	Breaks when the conditions set are satisfied the specified number of times.

Note: When X/Y-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

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

Notes: 1. O: Can be set in the dialog box.
X: Cannot be set in the dialog box.
2. For Break Condition 2, X-bus and Y-bus conditions cannot be specified.

2.2.3 Trace Functions

The SH7710 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**
HS7710KCM01H, HS7710KCI01H	Not supported
HS7710KCM02H, HS7710KCI02H	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.)

To set the AUD trace acquisition mode, click the [Trace] window with the right mouse button and select [Setting] from the pop-up menu to display the [Acquisition] dialog box. The AUD trace acquisition mode can be set in the [AUD mode1] or [AUD mode2] group box in the [Trace mode] page of the [Acquisition] dialog box.

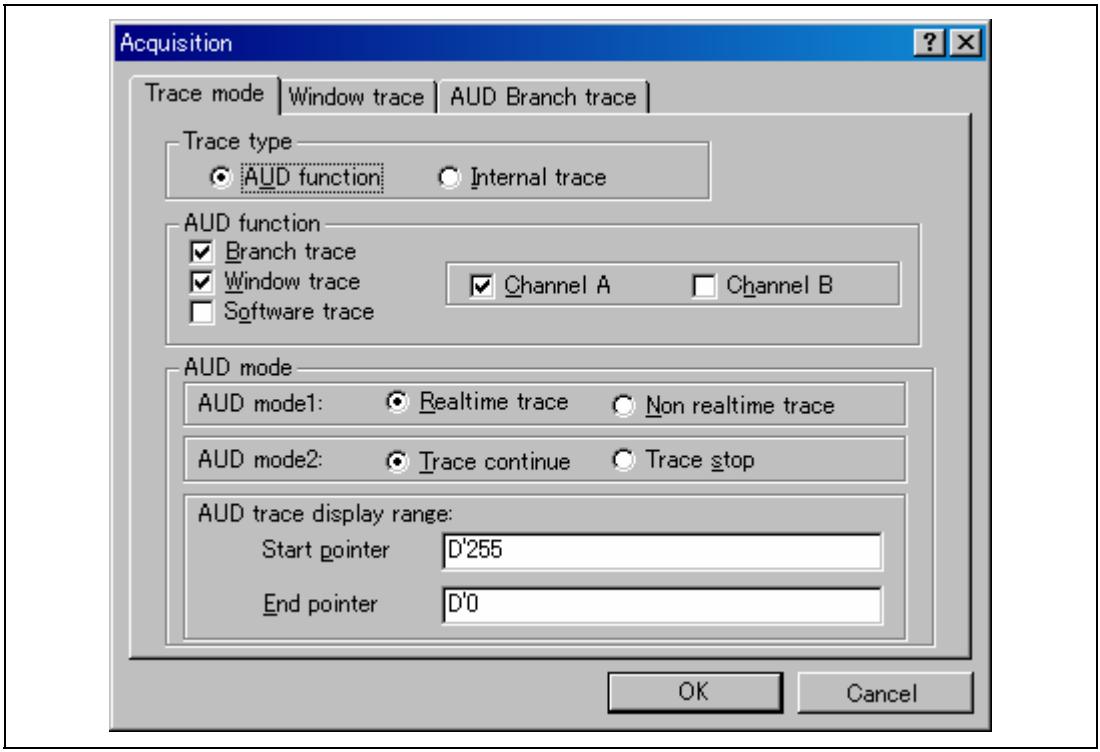


Figure 2.1 [Trace mode] Page

When the AUD trace function is used, select the [AUD function] radio button in the [Trace type] group box of the [Trace mode] page.

(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.

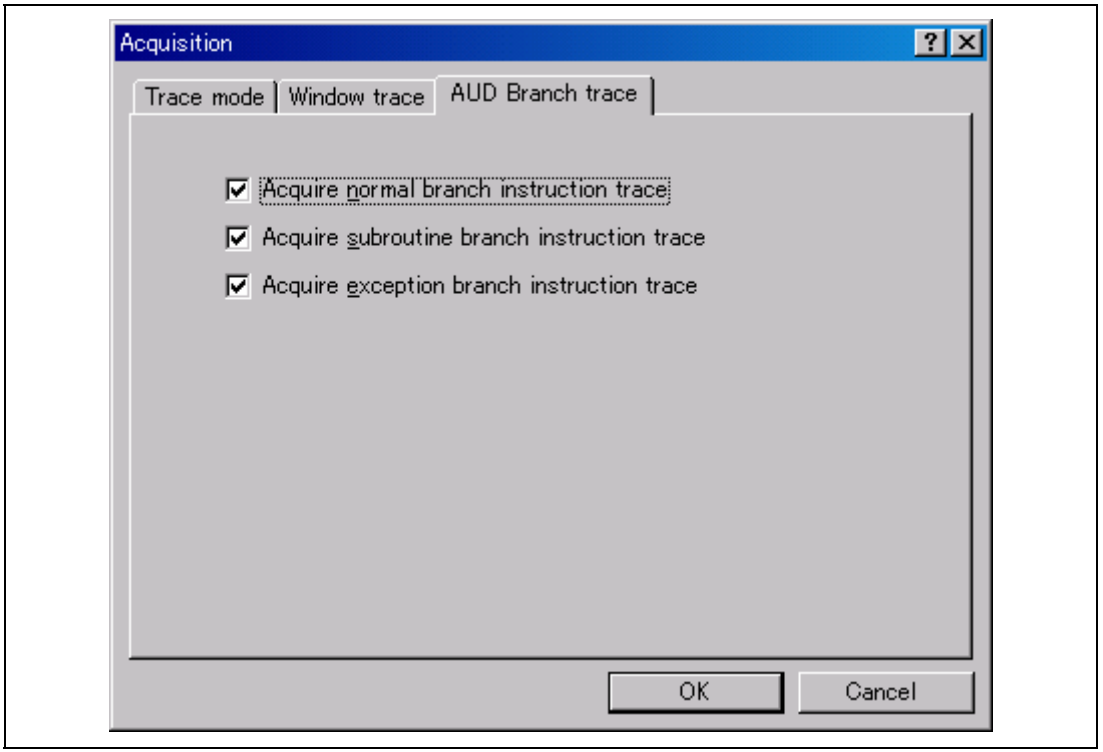


Figure 2.2 [AUD Branch trace] Page

(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.

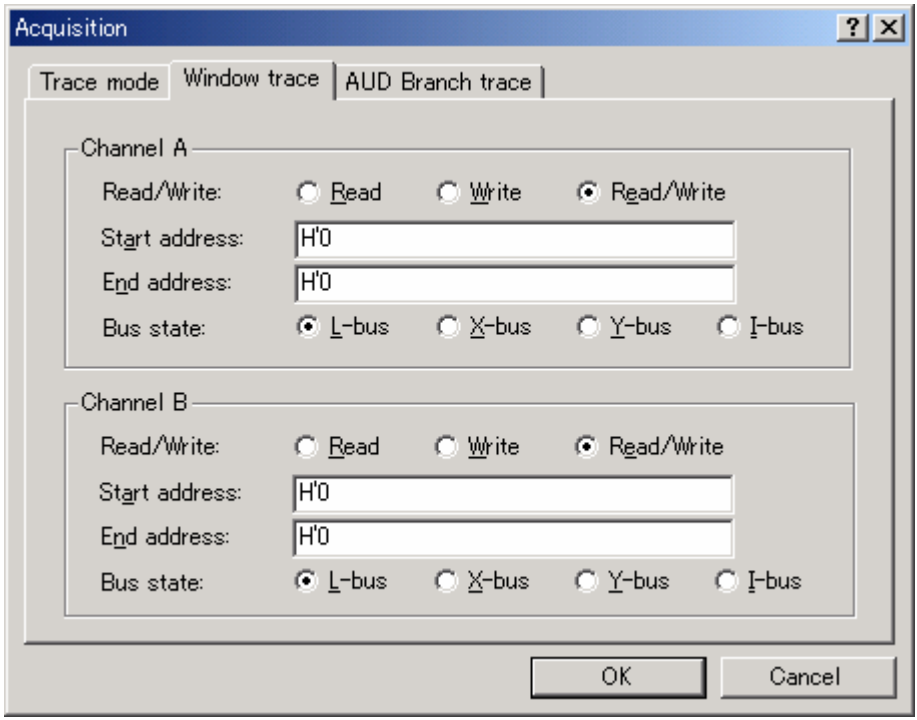


Figure 2.3 [Window trace] Page

Notes: 1. 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 or X/Y-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.

2. Address setting when X/Y-bus is selected
To trace both the X/Y-bus when the X/Y-bus is accessed at the same time, the X-bus condition must be set in channel A, and the Y-bus condition must be set in channel B.

(c) Software Trace Function

Note: This function can be supported with SHC/C++ compiler (manufactured by Renesas Technology Corp.; including OEM and bundle products) V7.0 or 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 SH7710 E10A emulator, when multiple loops are performed to reduce the number of AUD trace displays, only the IP counts up.
5. In the SH7710 E10A emulator, the maximum number of trace display pointers is as follows:
   When HS7710KCM02H is used: D'8191 to -0
   When HS7710KCI02H 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. The AUD trace acquisition is not available when [User] is selected in the [UBC mode] list box of the [Configuration] dialog box. In this case, close the [Trace] window.
7. Do not use the AUD full-trace mode for the VIO function.
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. This function traces and displays the branch instructions. The branch source address and branch destination address for the eight latest branch instructions are displayed. See figure 2.1, [Trace mode] Page.

Notes: 1. If an interrupt is generated at the program execution start or end, including a step operation, 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. The internal trace acquisition is not available when [User] is selected in the [UBC mode] list box of the [Configuration] dialog box. In this case, close the [Trace] window.

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

1. Set the JTAG clock (TCK) frequency to lower than the frequency of the SH7710 peripheral module clock (CKP).
2. Set the AUD clock (AUDCK) frequency to 50 MHz or lower 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 RAM areas in CS0 to CS6 and the internal RAM areas. However, a BREAKPOINT cannot be set to the following addresses:
   - ROM areas in CS0 to CS6
   - Areas other than CS0 to CS6
   - Areas other than the internal RAM
   - An instruction in which Break Condition 2 is satisfied
   - A slot instruction of a delayed branch instruction

— An area that can be only read by MMU

3. During step operation, 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 operation 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. When the [Normal] option is selected in the [Memory area] group box in the [General] page of the [Configuration] dialog box, a BREAKPOINT is set to a physical address or a virtual address according to the SH7710 MMU status during command input when the VPMAP_SET command setting is disabled. The ASID value of the SH7710 PTEH register during command input is used. When VPMAP_SET command setting is enabled, a BREAKPOINT is set to a physical address into which address translation is made according to the VP_MAP table. However, for addresses out of the range of the VP_MAP table, the address to which a BREAKPOINT is set depends on the SH7710 MMU status during command input. Even when the VP_MAP table is modified after BREAKPOINT setting, the address translated when the BREAKPOINT is set valid.
10. When the [Physical] option is selected in the [Memory area] group box in the [General] page of the [Configuration] dialog box, a BREAKPOINT is set to a physical address. A BREAKPOINT is set after disabling the SH7710 MMU upon program execution. After setting, the MMU is returned to the original state. When a break occurs at the corresponding virtual address, the cause of termination displayed in the status bar and the [Output] window is ILLEGAL INSTRUCTION, not BREAKPOINT.
11. When the [Virtual] option is selected in the [Memory area] group box in the [General] page of the [Configuration] dialog box, a BREAKPOINT is set to a virtual address. A BREAKPOINT is set after enabling the SH7710 MMU upon program execution. After setting, the MMU is returned to the original state. When an ASID value is specified, the BREAKPOINT is set to the virtual address corresponding to the ASID value. The emulator sets the BREAKPOINT after rewriting the ASID value to the specified value, and returns the ASID value to its original

value after setting. When no ASID value is specified, the BREAKPOINT is set to a virtual address corresponding to the ASID value at command input.

12. An address (physical address) to which a BREAKPOINT is set is determined when the BREAKPOINT is set. Accordingly, even if the VP_MAP table is modified after BREAKPOINT setting, the BREAKPOINT address remains unchanged. When a BREAKPOINT is satisfied with the modified address in the VP_MAP table, the cause of termination displayed in the status bar and the [Output] window is ILLEGAL INSTRUCTION, not BREAKPOINT.
13. 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 the BREAKCONDITION_SET Command

1. When [Step In], [Step Over], or [Step Out] is selected, the settings of Break Condition 2 are disabled.
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 Note on Setting the UBC_MODE Command

In the [Configuration] dialog box, 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 SH7710 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.

(a) Specifying the measurement start/end conditions

The measurement start/end conditions are specified in the [Mode] drop-down list box 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 and ended with 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 and ended with the satisfaction of the condition set in Break Condition 1.

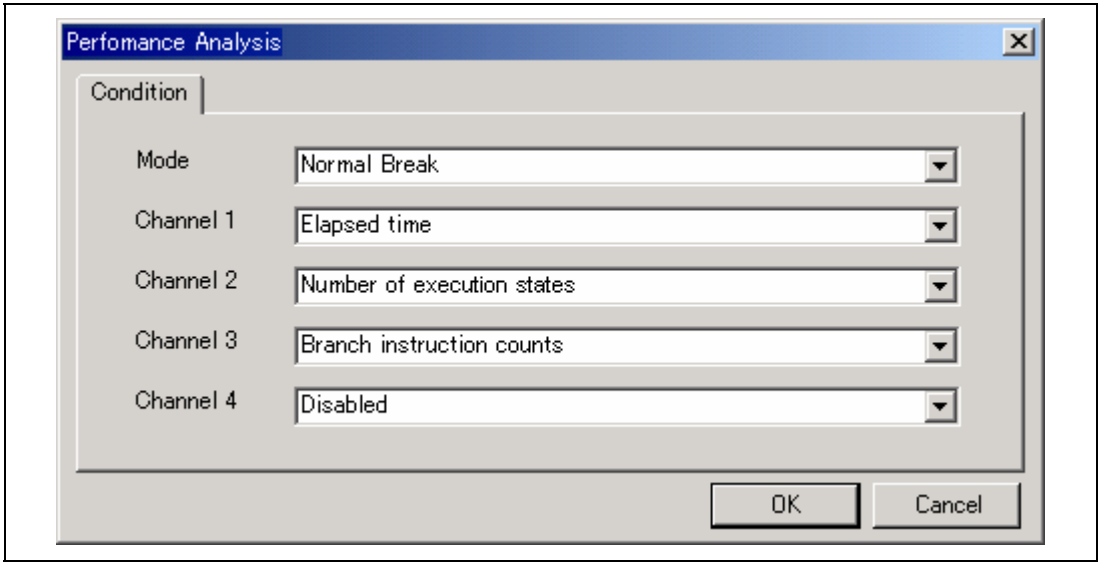


Figure 2.4 [Performance Analysis] Dialog Box

(b) 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 operation 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.

(c) 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	XYS (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

Table 2.10 Measurement Item (cont)

Selected Name	Option
Non-cacheable area data access cycle	NCD
Cacheable area access cycle	CC
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
No caching due to the settings of TLB cacheable bit	Counted for accessing the cacheable area.
Cache-on counting	Accessing the non-cacheable area is counted less than the actual number of cycles and counts. Accessing the cacheable, X/Y-RAM, and U-RAM 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 hexadecimal (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 SH7710 E10A Emulator

Publication Date: Rev.1.00, March 1, 2005
Published by: Sales Strategic Planning Div.
Renesas Technology Corp.
Edited by: Technical Documentation & Information Department
Renesas Kodaira Semiconductor Co., Ltd.

Renesas Technology Corp. Sales Strategic Planning Div. Nippon Bldg., 2-6-2, Ohte-machi, Chiyoda-ku, Tokyo 100-0004, Japan

RENESAS SALES OFFICES

http://www.renesas.com

Refer to "**http://www.renesas.com/en/network**" for the latest and detailed information.

Renesas Technology America, Inc.
450 Holger Way, San Jose, CA 95134-1368, U.S.A
Tel: <1> (408) 382-7500, Fax: <1> (408) 382-7501

Renesas Technology Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K.
Tel: <44> (1628) 585-100, Fax: <44> (1628) 585-900

Renesas Technology Hong Kong Ltd.
7th Floor, North Tower, World Finance Centre, Harbour City, 1 Canton Road, Tsimshatsui, Kowloon, Hong Kong
Tel: <852> 2265-6688, Fax: <852> 2730-6071

Renesas Technology Taiwan Co., Ltd.
10th Floor, No.99, Fushing North Road, Taipei, Taiwan
Tel: <886> (2) 2715-2888, Fax: <886> (2) 2713-2999

Renesas Technology (Shanghai) Co., Ltd.
Unit2607 Ruijing Building, No.205 Maoming Road (S), Shanghai 200020, China
Tel: <86> (21) 6472-1001, Fax: <86> (21) 6415-2952

Renesas Technology Singapore Pte. Ltd.
1 Harbour Front Avenue, #06-10, Keppel Bay Tower, Singapore 098632
Tel: <65> 6213-0200, Fax: <65> 6278-8001

Colophon 2.0

SuperH™ Family E10A Emulator Additional Document for User's Manual Specific Guide for the SH7710 E10A Emulator

Renesas Technology Corp.
2-6-2, Ote-machi, Chiyoda-ku, Tokyo, 100-0004, Japan