To our customers,

Old Company Name in Catalogs and Other Documents

On April 1st, 2010, NEC Electronics Corporation merged with Renesas Technology Corporation, and Renesas Electronics Corporation took over all the business of both companies. Therefore, although the old company name remains in this document, it is a valid Renesas Electronics document. We appreciate your understanding.

Renesas Electronics website: http://www.renesas.com

April 1st, 2010
Renesas Electronics Corporation

Issued by: Renesas Electronics Corporation (http://www.renesas.com)

Send any inquiries to http://www.renesas.com/inquiry.

Notice

1. All information included in this document is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas Electronics products listed herein, please confirm the latest product information with a Renesas Electronics sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas Electronics such as that disclosed through our website.
2. Renesas Electronics does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties by or arising from the use of Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
3. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part.
4. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
5. When exporting the products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations. You should not use Renesas Electronics products or the technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. Renesas Electronics products and technology may not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations.
6. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
7. Renesas Electronics products are classified according to the following three quality grades: "Standard", "High Quality", and "Specific". The recommended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below. You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application categorized as "Specific" without the prior written consent of Renesas Electronics. Further, you may not use any Renesas Electronics product for any application for which it is not intended without the prior written consent of Renesas Electronics. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for an application categorized as "Specific" or for which the product is not intended where you have failed to obtain the prior written consent of Renesas Electronics. The quality grade of each Renesas Electronics product is "Standard" unless otherwise expressly specified in a Renesas Electronics data sheets or data books, etc.

   "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; and industrial robots.

   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; safety equipment; and medical equipment not specifically designed for life support.

   "Specific": Aircraft; aerospace equipment; submersible repeaters; nuclear reactor control systems; medical equipment or systems for life support (e.g. artificial life support devices or systems), surgical implantations, or healthcare intervention (e.g. excision, etc.), and any other applications or purposes that pose a direct threat to human life.
8. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

(Note 1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its majority-owned subsidiaries.

(Note 2) "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics.

User's Manual

SuperH™ Family E10A-USB Emulator for Multi-core Microcomputers
Additional Document for User's Manual Supplementary Information on Using the SH7776

Renesas Microcomputer Development Environment System

SuperH™ Family

E10A-USB for SH7776 HS7776KCU04HE

Renesas Electronics
www.renesas.com

Rev.1.00 2009.11

Notes regarding these materials

1. This document is provided for reference purposes only so that Renesas customers may select the appropriate Renesas products for their use. Renesas neither makes warranties or representations with respect to the accuracy or completeness of the information contained in this document nor grants any license to any intellectual property rights or any other rights of Renesas or any third party with respect to the information in this document.
2. Renesas shall have no liability for damages or infringement of any intellectual property or other rights arising out of the use of any information in this document, including, but not limited to, product data, diagrams, charts, programs, algorithms, and application circuit examples.
3. You should not use the products or the technology described in this document for the purpose of military applications such as the development of weapons of mass destruction or for the purpose of any other military use. When exporting the products or technology described herein, you should follow the applicable export control laws and regulations, and procedures required by such laws and regulations.
4. All information included in this document such as product data, diagrams, charts, programs, algorithms, and application circuit examples, is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas products listed in this document, please confirm the latest product information with a Renesas sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas such as that disclosed through our website. (http://www.renesas.com)
5. Renesas has used reasonable care in compiling the information included in this document, but Renesas assumes no liability whatsoever for any damages incurred as a result of errors or omissions in the information included in this document.
6. When using or otherwise relying on the information in this document, you should evaluate the information in light of the total system before deciding about the applicability of such information to the intended application. Renesas makes no representations, warranties or guaranties regarding the suitability of its products for any particular application and specifically disclaims any liability arising out of the application and use of the information in this document or Renesas products.
7. With the exception of products specified by Renesas as suitable for automobile applications, Renesas products are not designed, manufactured or tested for applications or otherwise in systems the failure or malfunction of which may cause a direct threat to human life or create a risk of human injury or which require especially high quality and reliability such as safety systems, or equipment or systems for transportation and traffic, healthcare, combustion control, aerospace and aeronautics, nuclear power, or undersea communication transmission. If you are considering the use of our products for such purposes, please contact a Renesas sales office beforehand. Renesas shall have no liability for damages arising out of the uses set forth above.
8. Notwithstanding the preceding paragraph, you should not use Renesas products for the purposes listed below:
   (1) artificial life support devices or systems
   (2) surgical implantations
   (3) healthcare intervention (e.g., excision, administration of medication, etc.)
   (4) any other purposes that pose a direct threat to human life

   Renesas shall have no liability for damages arising out of the uses set forth in the above and purchasers who elect to use Renesas products in any of the foregoing applications shall indemnify and hold harmless Renesas Technology Corp., its affiliated companies and their officers, directors, and employees against any and all damages arising out of such applications.
9. You should use the products described herein within the range specified by Renesas, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas shall have no liability for malfunctions or damages arising out of the use of Renesas products beyond such specified ranges.
10. Although Renesas endeavors to improve the quality and reliability of its products, IC products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Please be sure to implement safety measures to guard against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other applicable measures. Among others, since the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
11. In case Renesas products listed in this document are detached from the products to which the Renesas products are attached or affixed, the risk of accident such as swallowing by infants and small children is very high. You should implement safety measures so that Renesas products may not be easily detached from your products. Renesas shall have no liability for damages arising out of such detachment.
12. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written approval from Renesas.
13. Please contact a Renesas sales office if you have any questions regarding the information contained in this document, Renesas semiconductor products, or if you have any other inquiries.

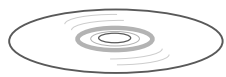
Contents

Section 1 Connecting the Emulator with the User System

1.1 Components of the Emulator

The emulator supports the SH7776. Table 1.1 lists the components of the emulator.

Table 1.1 Components of the Emulator

Classi-fication	Component	Appearance	Quan-tity	Remarks
Hard-ware	Emulator box		1	Depth: 68.0 mm, Width: 101.5 mm, Height: 22.7 mm, Mass: 66.9 g
	User system interface cable		1	14-pin type: Length: 17 cm, Mass: 12.3 g
	User system interface cable		1	38-pin type: Length: 20 cm, Mass: 10.8 g
	USB cable		1	Length: 150 cm, Mass: 53.0 g
Soft-ware	E10A-USB emulator setup program, SuperH™ Family E10A-USB Emulator for Multi-core User's Manual, Supplementary Information on Using the SH7776*, and Test program manual for HS0005KCU04H		1	HS0005KCU04SR, HS0005KCU04HJ, HS0005KCU04HE, HS7776KCU01HJ, HS7776KCU01HE, HS0005TM04HJ, and HS0005TM04HE (provided on a CD-R)

Note: Additional document for the MPUs supported by the emulator is included. Check the target MPU and refer to its additional document.

1.2 Connecting the Emulator with the User System

To connect the E10A-USB 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-USB emulator user's manual and hardware manual for the relevant 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
HS0005KCU04H	14-pin connector	Not available
HS0005KCU04H	38-pin connector	Available

The H-UDI port connector has the 14-pin and 38-pin types as described below. Use them according to the purpose of the usage.

1. 14-pin type (without AUD function)
   The AUD trace function cannot be used because only the H-UDI function is supported.
2. 38-pin type (with AUD function)
   The 38-pin connector supports AUD tracing, making the acquisition of large amounts of trace information in realtime possible. Window tracing, i.e. the acquisition of trace information (addresses and data) for memory access within a specified range also becomes possible.

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
14-pin connector	2514-6002	Minnesota Mining & Manufacturing Ltd.	14-pin straight type
38-pin connector	2-5767004-2	Tyco Electronics AMP K.K.	38-pin Mictor type

Note: 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. When designing the 38-pin connector layout on the user board, reduce cross-talk noise etc. by keeping other signal lines out of the region where the H-UDI port connector is situated. As shown in figure 1.1, an upper limit (5 mm) applies to the heights of components mounted around the user system connector.

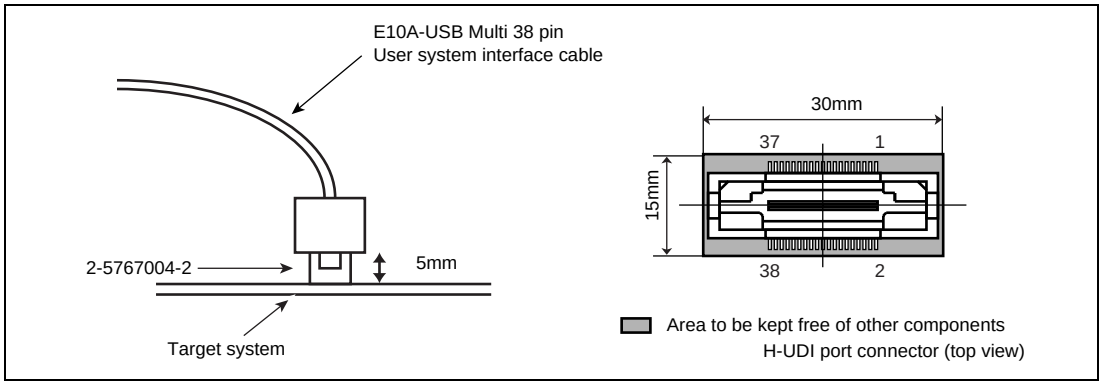


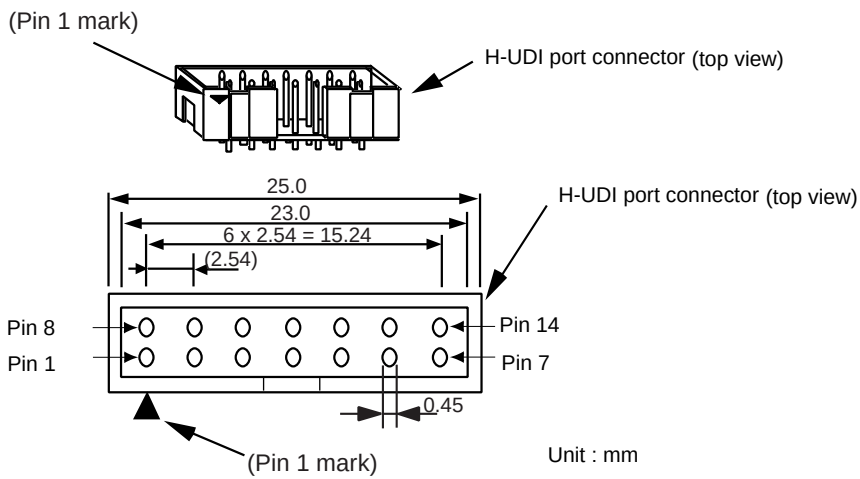
Figure 1.1 Restriction on Component Mounting

1.4 Pin Assignments of the H-UDI Port Connector

Figures 1.2 and 1.3 show the pin assignments of the 14-pin and 38-pin H-UDI port connectors, respectively.

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

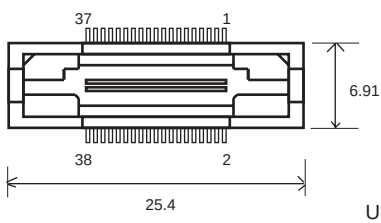
Pin No.	Signal	Input/ Output*1	SH7776 Pin No.	Note
1	TCK	Input	AG5	
2	TRST# *2	Input	AE6	
3	TDO	Output	AG6	
4	ASEBRK# /BRKACK *2	Input/ output	AH5	
5	TMS	Input	AH6	
6	TDI	Input	AF5	
7	RES# *2	Output	B28	User reset
8	N.C.	—		
9	(GND) *4	—		
11	UVCC	Output		
10,12,13	GND	—		
14	GND *3	Output		



Notes: 1. Input to or output from the user system.
2. The symbol (#) 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.
4. When the user system interface cable is connected to this pin and the MPMD pin is set to 0, do not connect to GND but to the MPMD pin directly.

Figure 1.2 Pin Assignments of the H-UDI Port Connector (14 Pins)

Pin No.	Signal	Input/ Output *1	SH7776 Pin No.	Note	Pin No.	Signal	Input/ Output *1	SH7776 Pin No.	Note
1	N.C.	—			20	N.C.	—		
2	N.C.	—			21	TRST# *2	Input	AE6	
3	MPMD(GND) *4	—			22	N.C.	—		
4	N.C.	—			23	N.C.	—		
5	UCON#(GND) *3	—			24	AUDATA3	Output	AH3	
6	AUDCK	Output	AG1		25	N.C.	—		
7	N.C.	—			26	AUDATA2	Output	AF3	
8	ASEBRK#/ BRKACK *2	Input/ output	AH5		27	N.C.	—		
9	RES# *2	Output	B28	User reset	28	AUDATA1	Output	AH2	
10	N.C.	—			29	N.C.	—		
11	TDO	Output	AG6		30	AUDATA0	Output	AG2	
12	UVCC_AUD	Output			31	N.C.	—		
13	N.C.	—			32	AUDSYNC	Output	AF1	
14	UVCC	Output			33	N.C.	—		
15	TCK	Intput	AG5		34	N.C.	—		
16	N.C.	—			35	N.C.	—		
17	TMS	Intput	AH6		36	N.C.	—		
18	N.C.	—			37	N.C.	—		
19	TDI	Intput	AF5		38	N.C.	—		



Unit: mm

H-UDI port connector (top view)

Notes: 1. Input to or output from the user system.
2. The symbol (#) 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.
4. When the user system interface cable is connected to this pin and the MPMD pin is set to 0, do not connect to GND but to the MPMD pin directly.
5. The GND bus lead at the center of the H-UDI port connector must be grounded.

Figure 1.3 Pin Assignments of the H-UDI Port Connector (38 Pins)

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

1.5.1 Recommended Circuit (14-Pin Type)

Figure 1.4 shows a recommended circuit for connection between the H-UDI port connector (14 pins) and the MPU when the emulator is in use.

Notes: 1. Do not connect anything to the N.C. pins of the H-UDI port connector.

2. The MPMD pin must be 0 when the emulator is connected and 1 when the emulator is not connected, respectively.
   (1) When the emulator is used: MPMD = 0
   (2) When the emulator is not used: MPMD = 1
   Figure 1.4 shows an example of circuits that allow the MPMD pin to be GND (0) whenever the emulator is connected by using the user system interface cable.
   When the MPMD pin is changed by switches, etc., ground pin 9. Do not connect this pin to the MPMD pin.
3. When a network resistance is used for pull-up, it may be affected by a noise. Separate TCK from other resistances.
4. When the power is supplied, the TRST# pin must be low during a specified period regardless of whether or not the H-UDI is used.
5. The pattern between the H-UDI port connector and the MPU must be as short as possible. Do not connect the signal lines to other components on the board.
6. The operating voltage for the H-UDI of the MPU is the VccQ voltage, so only supply this voltage to the UVCC pin.
7. The resistance values shown in figure 1.4 are for reference.
8. For the pin processing in cases where the emulator is not used, refer to the hardware manual of the related MPU.

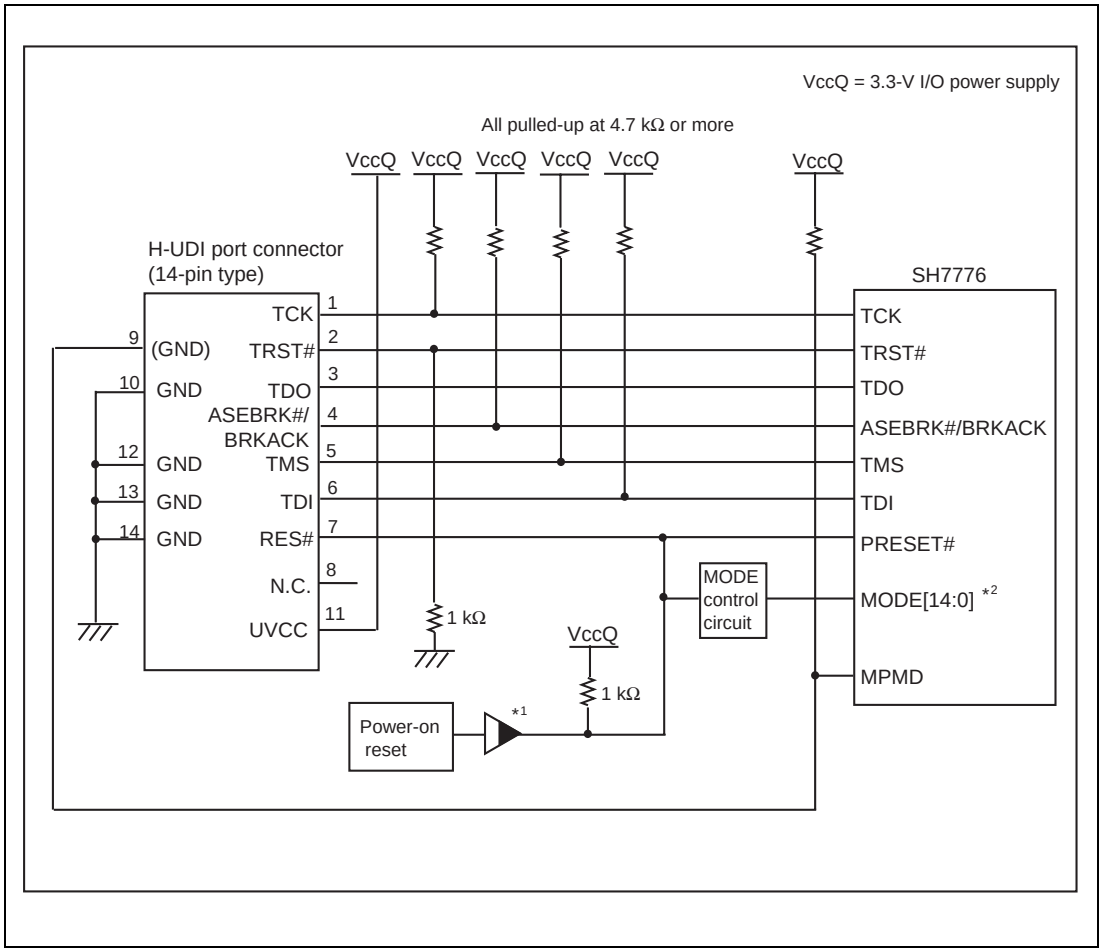


Figure 1.4 Recommended Circuit for Connection between the H-UDI Port Connector and MPU when the Emulator is in Use (14-Pin Type)

Notes: 1. Open drain buffer.

2. The MODE pins of the SH7776 are only effective during the period of a reset. To set the operating mode correctly, ensure that the settings of the MODE[14:0] pins are reflected in the output signals from the H-UDI port connector.

1.5.2 Recommended Circuit (38-Pin Type)

Figure 1.5 shows a recommended circuit for connection between the H-UDI and AUD port connectors (38 pins) and the MPU when the emulator is in use.

Notes:
1. Do not connect anything to the N.C. pins of the H-UDI port connector.
2. The MPMD pin must be 0 when the emulator is connected and 1 when the emulator is not connected, respectively.
   (1) When the emulator is used: MPMD = 0
   (2) When the emulator is not used: MPMD = 1
   Figure 1.5 shows an example of circuits that allow the MPMD pin to be GND (0) whenever the emulator is connected by using the user system interface cable.
   When the MPMD pin is changed by switches, etc., ground pin 3. Do not connect this pin to the MPMD pin.
3. When a network resistance is used for pull-up, it may be affected by a noise. Separate TCK from other resistances.
4. The pattern between the H-UDI port connector and the MPU must be as short as possible. Do not connect the signal lines to other components on the board.
5. The AUD signals (AUDCK, AUDATA3 to AUDATA0, and AUDSYNC) operate in high speed. Isometric connection is needed if possible. Do not separate connection nor connect other signal lines adjacently.
6. The operating voltages for the H-UDI and AUD of the MPU are the VccQ and VccQ-DDR voltages, respectively, so supply the VccQ voltage to the UVCC pin and the VccQ-DDR voltage to the UVCC_AUD pin.
7. The resistance values shown in figure 1.5 are for reference.
8. For the AUDCK pin, guard the pattern between the H-UDI port connector and the MPU at GND level.
9. When the power is supplied, the TRST# pin must be low during a specified period regardless of whether or not the H-UDI is used.
10. The GND bus lead at the center of the H-UDI port connector must be grounded.
11. For the pin processing in cases where the emulator is not used, refer to the hardware manual of the related MPU.

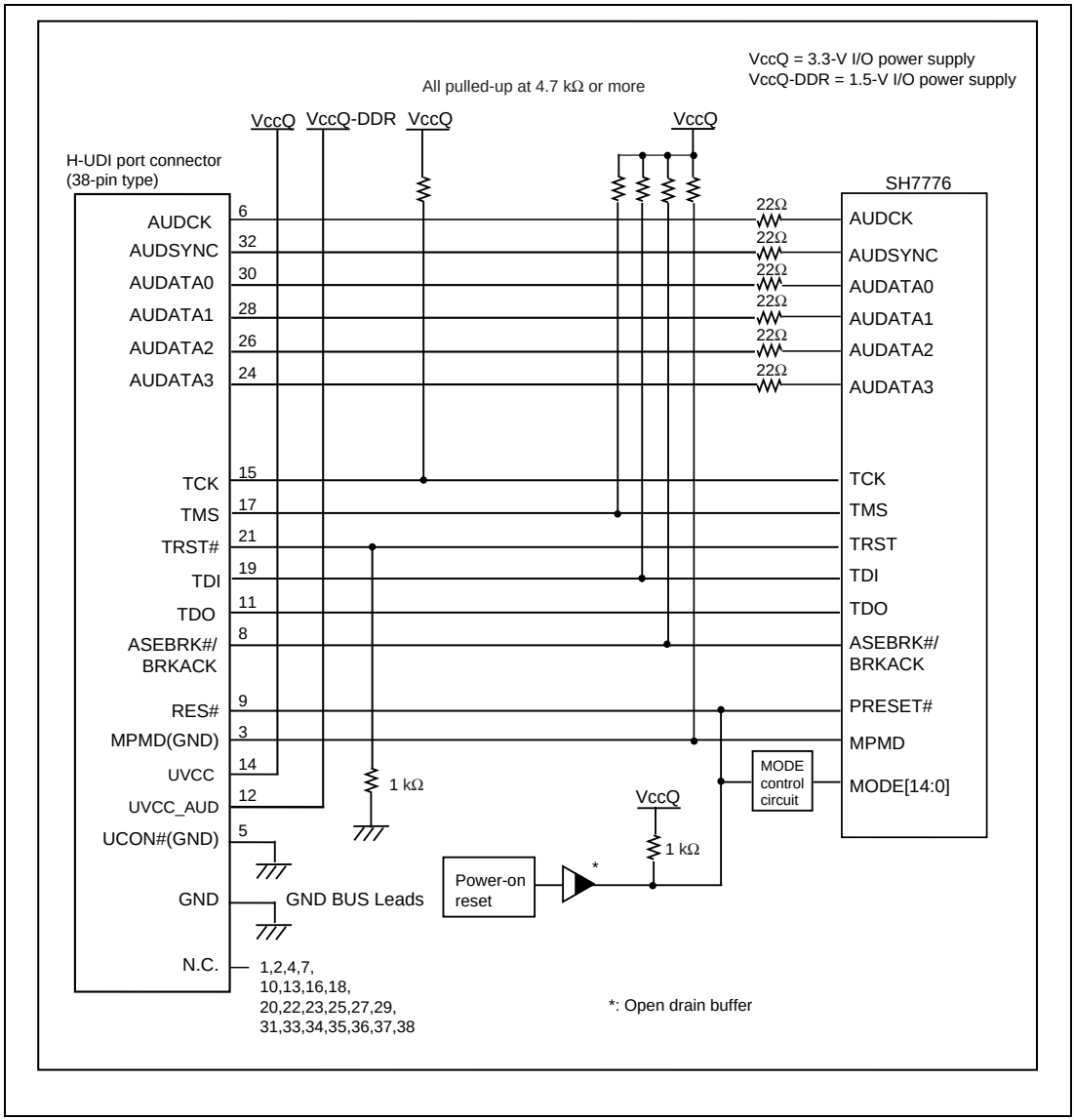


Figure 1.5 Recommended Circuit for Connection between the H-UDI Port Connector and MPU when the Emulator is in Use (38-Pin Type)

Notes: 1. Open drain buffer.

2. The MODE pins of the SH7776 are only effective during the period of a reset. To set the operating mode correctly, ensure that the settings of the MODE[14:0] pins are reflected in the output signals from the H-UDI port connector.

Section 2 Software Specifications when Using the SH7776

2.1 Differences between the SH7776 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 SH7776 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
DBR	H'00000000
SGR	H'00000000
SPC	H'00000000
SSR	H'000000F0
FPUL	H'00000000
FPSCR	H'00040001
FR0 to FR15	H'00000000
XF0 to XF15	H'00000000
DR0 to DR14	H'00000000
XD0 to XD15	H'00000000

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

3. Low-Power States (Sleep, Light sleep, Module standby, and DDR3-SDRAM power-supply back-up).

   For low-power consumption, the SH7776 has sleep, light sleep, module standby, and DDR3-SDRAM power supply back-up states.

   The SLEEP instruction is used for switching to the sleep and light sleep states. When the emulator is in use, release from the sleep and light sleep states is either by the normal release function or by the [STOP] button, in which case a break will occur.

   Debugging is not supported in the power supply back-up state for DDR3-SDRAM.

Note: During execution of a SLEEP instruction, do not look up or alter contents of memory and do not use [Internal mode] to switch the target CPU. On release from the low-power state, instruction execution starts from the PC value which follows that of the SLEEP instruction. Access to memory during the execution of SLEEP instructions can be stopped by deselecting the [Sleep] check box under [Memory access option] in the [Configuration] dialog box.

4. Reset Signals

   Do not assert the RESET signal during a break. Attempting to do so will lead to a TIMEOUT error.

Note: Do not break the user program when the RESET signal is being low and the WAIT control signal is being active. A TIMEOUT error will occur. If the WAIT control signal is fixed to active 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 emulator to access the memory. Therefore, realtime emulation cannot be performed.

   The stopping time of the user program is as follows:

   Environment:

   Host computer: 3 GHz (Pentium® IV)
   SH7776 (with only CPU0 in use): 533 MHz (CPU clock)
   JTAG clock: 30 MHz (TCK clock)

   When a one-byte memory is read from the command-line window, the stopping time will be about 40 ms.

7. Memory Access during User Program Break

   The emulator can download the program for the flash memory area (for details, refer to section 7.24, Download Function to the Flash Memory Area, in the SuperH™ Family E10A-USB Emulator for Multi-core 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 issues a single write to outside. The LRU is not updated.

   — At memory read: Reads memory from the cache. The LRU is not updated.

   Therefore, when memory read or write is performed during user program break, the cache state does not change.

   — At breakpoint set: Disables the instruction cache.

   — Writing to memory that causes a cache hit for the same area for both CPUs: The instruction cache is disabled for both CPUs.

9. UBC

   When [User] is specified in the [UBC mode] list box on the [Common Setting] page 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 on the [Common Setting] page 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] on the [General] page 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] on the [Common Setting] page in the [Configuration] dialog box cannot be recovered by loading sessions. Thus the TCK value will be 5 MHz.
    If the [Search the best JTAG clock] option is used when the emulator is initiated, the TCK value will be initialized as a value that has been automatically acquired.

12. [IO] Window

— Display and modification

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

The watchdog timer only operates when both CPUs are executing the user program. 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. After the I/O-register definition file is created, the MPU's specifications 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 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.

14. [Reset CPU] and [Reset Go] in the [Debug] Menu

When [Reset CPU] or [Reset Go] is selected, a H-UDI reset is issued and both CPUs are always reset.
For the H-UDI reset, the watchdog timer except for the overflow counter and the clock oscillator is not initialized.
If both CPUs are handling processing during a break when [Reset CPU] is selected, CPU1 is released from the module stop state and enters the break state. If one CPU is executing the user program, if the operation is performed from the CPU where a break is occurring, the CPU which executing the user program, the module stop state follow the initial value of the device.
The [Reset Go] function is only effective when the synchronous executing function is enabled and the operation is from the master CPU. When [Reset Go] is selected, CPU1 enters the module stop state. Release CPU0 from within the user program.

2.2 Specific Emulator Functions for Use with the SH7776

2.2.1 Functions for Synchronized Debugging

For use with the SH7776, the following facilities for synchronization of debugging are available.

Reset	Operations in response to the [Reset CPU] and [Reset Go] functions are synchronized. For synchronization of the response to the [Reset Go] function, the [Go] check box must also should be checked. Response to the reset function is always synchronized, and non-synchronized operations are not available.
Go	Operations in response to the [Go] and [Reset Go] functions are synchronized. For synchronization of the response to the [Reset Go] function, the [Reset] check box must also should be checked. Synchronized or non-synchronized operation can be set as desired.
Break/Halt	Operations in response to a device break and selection of the [Halt Program] function are synchronized. Selection of synchronized or non-synchronized operation in response to individual types of break is not possible. Synchronized or non-synchronized operation in response to breaks in general and to selection of the [Halt Program] can be set as desired.
Step	Operations in response to the various functions for step execution are synchronized. When synchronized stepping is being performed while the other core is executing the user program, operation of the other core at the end of step execution depends on the setting for [Break/Halt]. Synchronized or non-synchronized operation can be set as desired.
Connect	Operation of the emulator in response to [Connect] is synchronized in all sessions. Connection is always synchronized. Non-synchronized connection is not available.
Download Modules	Operation of the emulator in response to [Download Modules] is synchronized in all sessions. Downloading is always synchronized. Non-synchronized downloading is not available.
Initialize	Operation of the emulator in response to [Initialize] is synchronized in all sessions. Initialization is always synchronized. Non-synchronized initialization is not available.

The emulator is used to set 10 CPU event conditions and software trace conditions (Ch. 1 to Ch. 10) per CPU, and 6 system-bus events as common conditions (Ch1 to Ch6) can be set. Table 2.4 lists the conditions that can be used to set up Event Conditions.

Table 2.4 Types of Event Condition

Event Condition Type	Description
Address bus condition (Address)	Breaks when the SH7776 address bus value or the program counter value matches the specified value.
Data bus condition (Data)	Breaks when the SH7776 data bus value matches the specified value. Byte, word, or longword can be specified as the access data size.
Bus state condition (Bus State)	There are two bus state condition settings: Bus state condition: Breaks or acquires a trace when the data bus or the X-Bus or Y-Bus address bus of the SH7776 is matched. Read/Write condition: Breaks or acquires a trace when the specified read/write condition is matched.
Window address condition	Breaks or acquires a trace when the data in the specified memory range is accessed.
LDTLB instruction event condition	Breaks when the SH7776 executes the LDTLB instruction.
Count	Breaks when the conditions set are satisfied the specified number of times.
Branch trace condition (Branch trace)	Breaks or acquires a trace when a branch occurs with the condition specified by the SH7776. (By default, trace acquisition is enabled).
Software trace	Selects whether or not the software trace is acquired.
System bus	Breaks or acquires a trace when the address or data on the system bus is matched.
Action	Selects the operation when a condition, such as setting a break, trace, or performance start or end, is matched.

Table 2.5 lists the combinations of conditions that can be set under CPU Event (Ch1 to Ch10) and the software trace.

Table 2.5 Dialog Boxes for Setting Event Conditions

	Function									
Dialog Box	Address Bus Condition (Address)	Data Bus Condition (Data)	ASID Condition (ASID)	Bus State Condition (Bus Status)	Window Address Condition (Window address)	LDTLB Instruction Break	Count Condition (Count)	Branch Condition (Branch Trace)	Software Trace	Action
[CPU Event 1] dialog box	O	X	O	O	X	X	X	X	X	O (B and P)
[CPU Event 2] dialog box	O	O	O	O	X	X	O	X	X	O (B and P)
[CPU Event 3] dialog box	O	X	O	X	X	X	X	X	X	O (B and P)
[CPU Event 4] dialog box	O	X	O	X	X	X	X	X	X	O (B and P)
[CPU Event 5] dialog box	X	X	O	O	O	X	X	X	X	O (B, T, and P)
[CPU Event 6] dialog box	X	X	O	O	O	X	X	X	X	O (B, T, and P)
[CPU Event 7] dialog box	X	X	X	X	X	O	X	X	X	Break fixed
[CPU Event 8] dialog box	O	X	O	O	X	X	X	X	X	O (B and P)
[CPU Event 9] dialog box	O	O	O	O	X	X	O	X	X	O (B and P)
[CPU Event 10] dialog box	X	X	X	X	X	X	X	O	X	O (B, T, and P)
[Software Trace] dialog box	X	X	X	X	X	X	X	X	O	Tracing only

Notes: 1. O: Can be set in the dialog box.
X: Cannot be set in the dialog box.

2. For the Action item,
B: Setting a break is enabled.
T: Setting a trace is enabled.
P: Setting a performance start or end condition is enabled.

Sequential Setting: In the emulator, sequential setting of an Event Condition is enabled.

Table 2.6 Sequential Break Conditions ([CPU Sequential setting] Dialog Box)

Dialog Box	Type	Event Condition	Description
[CPU Sequential Event] Page	2 Channel Sequential	Ch2 -> 1	Halts a program when a condition is satisfied in the order of CPU Event 2, 1. An break condition must be set for Ch2 and Ch1.
		Ch4 -> 3	Halts a program when a condition is satisfied in the order of CPU Event 4, 3. An break condition must be set for Ch4 and Ch3.
		Ch6 -> 5	Halts a program when a condition is satisfied in the order of CPU Event 6, 5. An break condition must be set for Ch6 and Ch5.
		Ch9 -> 8	Halts a program when a condition is satisfied in the order of CPU Event 9, 8. An break condition must be set for Ch9 and Ch8.
	Many Channel Sequential	Ch3 -> 2 -> 1	Halts a program when a condition is satisfied in the order of CPU Event 3, 2, 1. An break condition must be set for Ch3, Ch2, and Ch1.
		Ch4 -> 3-> 2 -> 1	Halts a program when a condition is satisfied in the order of CPU Event 4, 3, 2, 1. An break condition must be set for Ch4, Ch3, Ch2, and Ch1.
		Ch5 -> 4 -> 3-> 2 -> 1	Halts a program when a condition is satisfied in the order of CPU Event 5, 4, 3, 2, 1. An break condition must be set for Ch5, Ch4, Ch3, Ch2, and Ch1.
		Ch6 -> 5 -> 4 -> 3-> 2 -> 1	Halts a program when a condition is satisfied in the order of CPU Event 6, 5, 4, 3, 2, 1. An break condition must be set for Ch6, Ch5, Ch4, Ch3, Ch2, and Ch1.
		Ch8 -> 6 -> 5 -> 4 -> 3-> 2 -> 1	Halts a program when a condition is satisfied in the order of CPU Event 8, 6, 5, 4, 3, 2, 1. An break condition must be set for Ch8, Ch6, Ch5, Ch4, Ch3, Ch2, and Ch1.
		Ch9 -> 8 -> 6 -> 5 -> 4 -> 3-> 2 -> 1	Halts a program when a condition is satisfied in the order of CPU Event 9, 8, 6, 5, 4, 3, 2, 1. An break condition must be set for Ch9, Ch8, Ch6, Ch5, Ch4, Ch3, Ch2, and Ch1.

Table 2.6 Sequential Event Conditions (cont)

Dialog Box	Type	Event Condition	Description
[CPU Sequential Event] Page (cont)	CPU Extended		Expands the [CPU Sequential Extend] page. The sequential setting is enabled with any combination. For details, refer to section 2.2.1, Sequential Break Extension Setting, in this manual.
[System Bus Sequential Event] Page	System Bus Channel	Ch2 -> 1	Halts a program when a condition is satisfied in the order of Systembus Event 2, 1. An break condition must be set for Ch2 and Ch1.
		Ch1 -> 2	Halts a program when a condition is satisfied in the order of Systembus Event 1, 2. An break condition must be set for Ch1 and Ch2.
		Ch4 -> 3	Halts a program when a condition is satisfied in the order of Systembus Event 4, 3. An break condition must be set for Ch4 and Ch3.
		Ch3 -> 4	Halts a program when a condition is satisfied in the order of Systembus Event 3, 4. An break condition must be set for Ch3 and Ch4.
		Ch6 -> 5	Halts a program when a condition is satisfied in the order of Systembus Event 6, 5. An break condition must be set for Ch6 and Ch5.
		Ch5 -> 6	Halts a program when a condition is satisfied in the order of Systembus Event 5, 6. An break condition must be set for Ch5 and Ch6.
	System Bus Extended		Expands the [CPU Sequential Extend] page. The sequential setting is enabled with any combination. For details, refer to section 2.2.1, Sequential Break Extension Setting, in this manual.

Sequential Break Extension Setting:

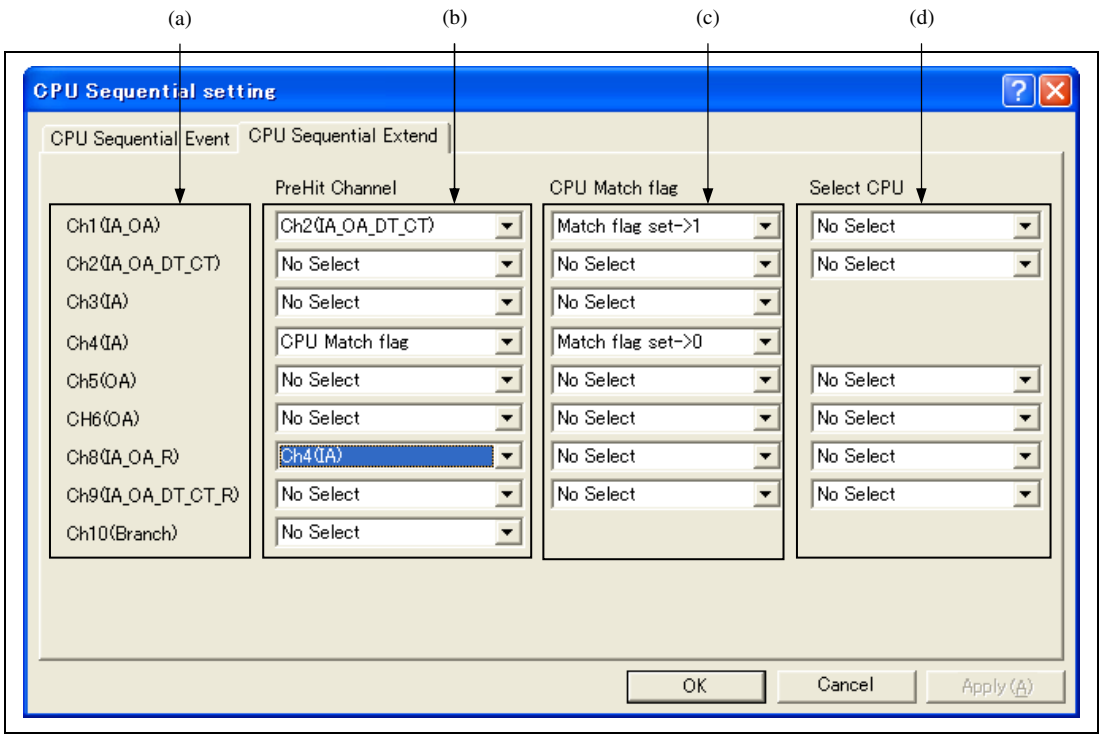


Figure 2.1 [CPU Sequential Extend] Page

(a) Indicates the channel name for setting conditions.

(b) Selects a condition that is satisfied before the channel which sets up conditions.
When a channel name is selected, it is required that the condition of the channel selected here must have already been satisfied.
When [CPU Match flag] is selected, the CPU match flag must be set.
When a condition is selected by the channel selected here, no break will occur.

(c) When a condition is satisfied, the CPU match flag is set or cleared.
When a program breaks, the CPU match flag is initialized.

(d) If a channel for the other CPU was specified in column (b), the CPU number is selected here.
If No Select is selected, the setting in column (b) is for the target CPU of the given debugger session.

Note: When setting a sequential break condition that spans the CPUs, remove the check mark from the [Acquire break] check box on the [Action] page for the channel along the way to doing so.

Set the event condition for each channel in the [Event Condition] dialog box; this also applies to the [System Bus Sequential Extend] page.

Usage Example of Sequential Break Extension Setting: A tutorial program provided for the product is used as an example. For the tutorial program, refer to section 7, Tutorial, SH-4A in the SuperH™ Family E10A-USB Emulator User's Manual.

The conditions of Event Condition are set as follows:

1. Ch1

   Breaks address H'0000108C when the condition [Prefetch address break after executing] is satisfied.
2. Ch2

   Breaks address H'0000107A when the condition [Prefetch address break after executing] is satisfied.
3. Ch4

   Breaks address H'000010AC when the condition [Prefetch address break after executing] is satisfied.
4. Ch8

   Breaks address H'000010BE when the condition [Prefetch address break after executing] is satisfied.
   Note: Do not set other channels.
5. Set the [CPU Sequential Extend] page as shown in figure 2.1.

Then, set the program counter and stack pointer (PC = H'00000800, R15 = H'00010000) in the [Registers] window and click the [Go] button. If this does not execute normally, issue a reset and execute the above procedures.

The program is executed up to the condition of Ch8 and halted. Here, the condition is satisfied in the order of Ch2 -> 1 -> 4 -> 8.

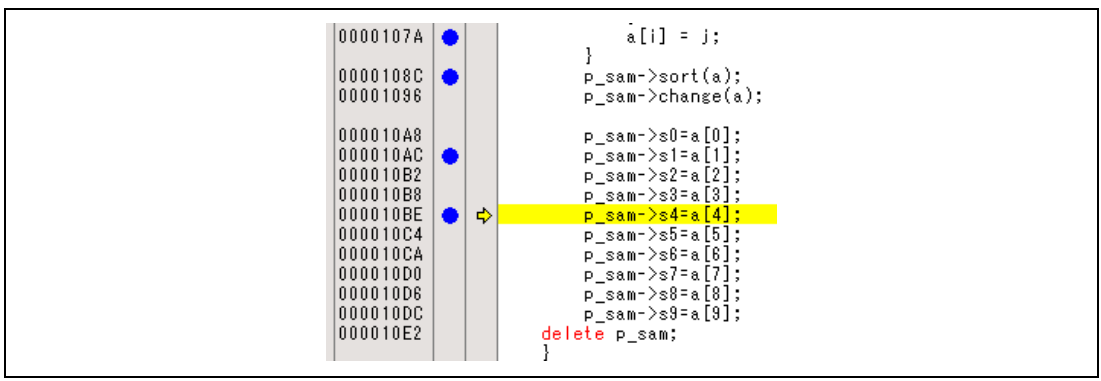


Figure 2.2 [Source] Window at Execution Halted (Sequential Break)

2.2.2 Trace Functions

The emulator supports the trace functions listed in table 2.7.

Table 2.7 Trace Functions

Function	Internal Trace	AUD Trace	Memory Output Trace
Branch trace	Supported (60 branches)	Supported	Supported
Range memory access trace	Supported (60 events)	Supported	Supported
Software trace	Supported (60 events)	Supported	Supported

Branch Trace Functions: The branch source and destination addresses, their source lines, branch types, and types of accessed bus masters are displayed.

[Setting Method]
Open the [CPU Event 10] dialog box by double-clicking on the Ch10 (Branch) column of the [CPU Event] sheet of the [Event point] window, select the [Branch event] page, and select or deselect check boxes in the [Branch] group box as required to set conditions for branch acquisition.

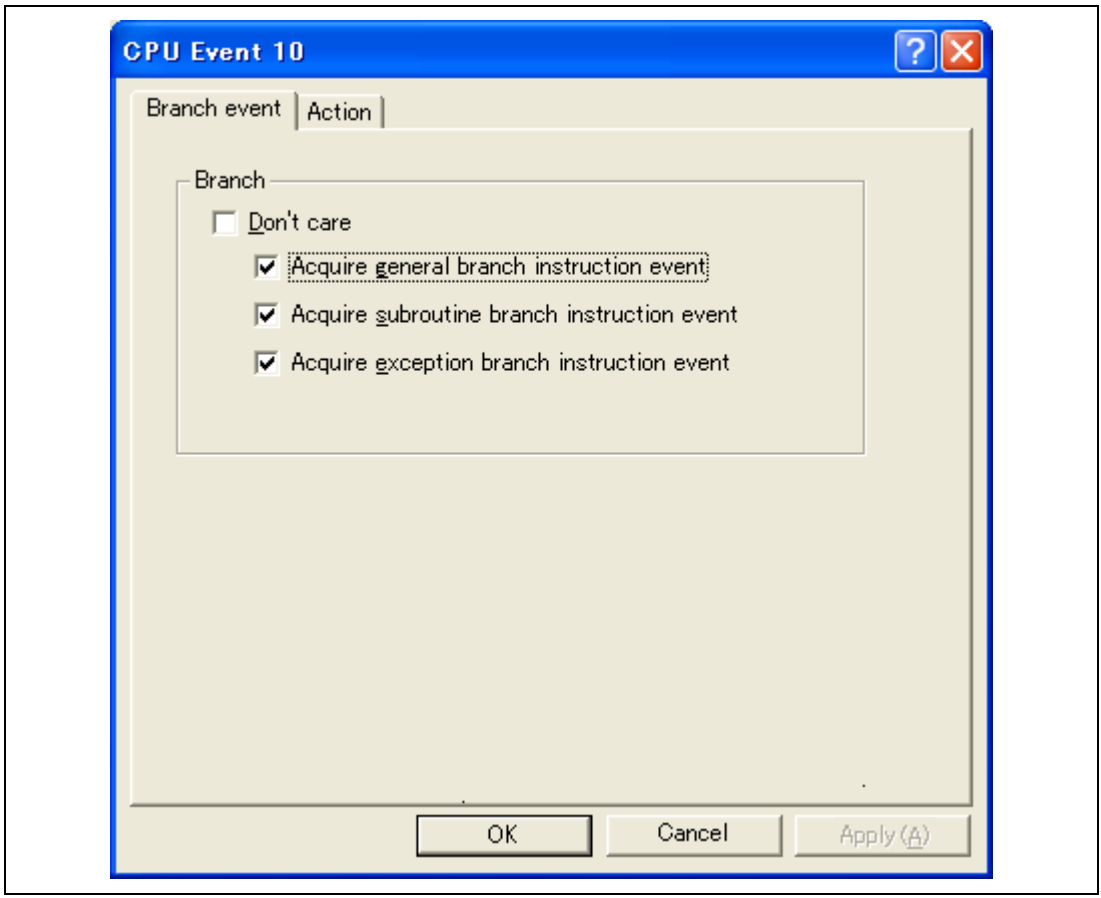


Figure 2.3 [Branch trace] Dialog Box

A branch trace can be acquired by selecting the [Acquire trace] check box of the [Action] page.

Note: To cancel settings, select [Delete] from the popup menu that is opened by clicking on the Ch10 (Branch) column with the right-mouse button.

Range Memory Access Trace Functions: The memory access within the specified range is acquired by a trace. The read cycle, write cycle, or read/write cycle can be selected as the bus type, ASID value, or bus cycle for trace acquisition.

[Setting Method]

(i) To open the [CPU Event 5] or [CPU Event 6] dialog box, double-click on the Ch5 (OA) or Ch6 (OA) column the [CPU Event] sheet of the [Eventpoint] window.

(ii) Remove the check mark of the [Don't care] check box in the [Window address] page and enter the memory range to be set.

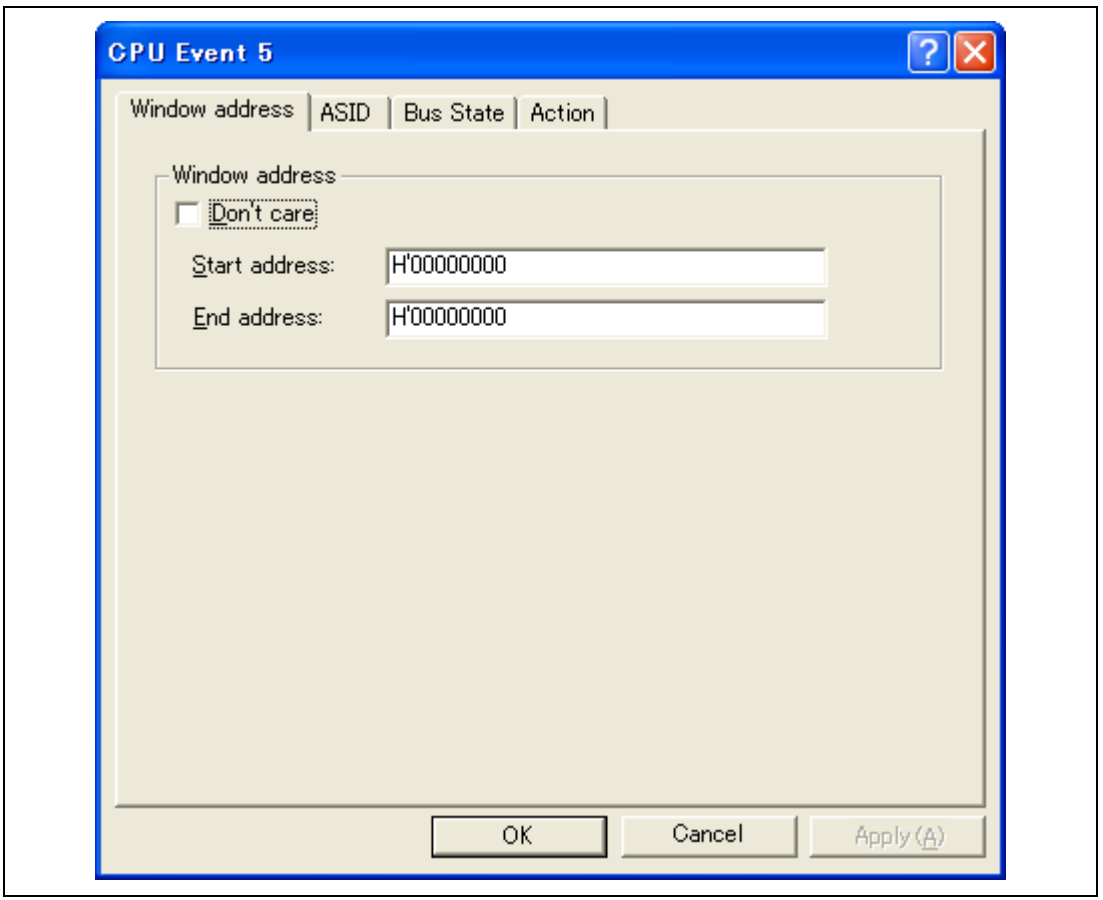


Figure 2.4 [Window address] Page

(iii) Open the [ASID] page, remove the check mark of the [Don't care] check box, and enter the ASID value to be set.
When the ASID value is not set as a condition, do not remove the check mark of the [Don't care] check box.

(iv) Open the [Bus state] page and specify the bus type and bus cycle that are to be set.

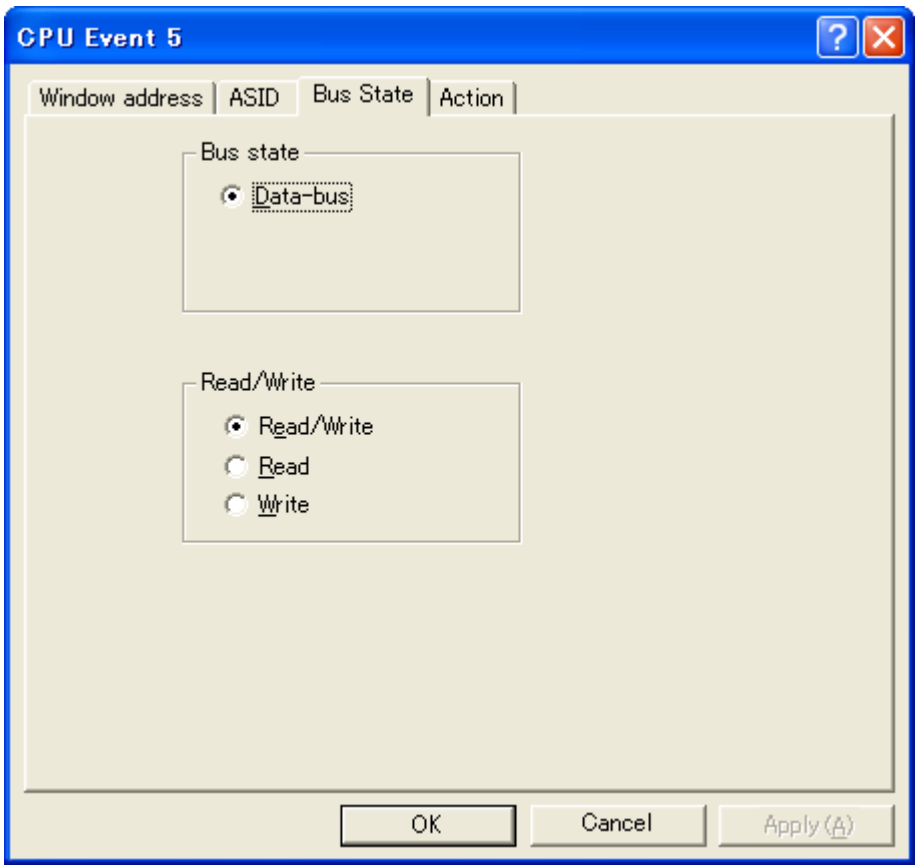


Figure 2.5 [Bus State] Page

(v) Selecting the [Acquire trace] check box in the [Action] page enables acquiring memory access within the range.

Note: To cancel settings, select the popup menu that is opened by clicking on the Ch5 (OA) or Ch6 (OA) column with the right-mouse button.

Software Trace Function:

Note: This function can be supported with SHC/C++ compiler (manufactured by Renesas Technology Corp.; including OEM and bundle products) V6.0 or later.
However, SHC/C++ compiler (including OEM and bundle products) V8.0 or later is needed when instructions other than those compatible with SH4 are output.

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 SuperH™ RISC engine C/C++ Compiler, Assembler, Optimizing Linkage Editor User's Manual.

When the load module is downloaded on the emulator and is executed while a software trace function is valid, 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 [Acquire Software trace] radio button in the [Software trace] dialog box that is opened by double-clicking on the software Trace column of the [CPU Event] sheet of the [Eventpoint] window.

Note: To cancel settings, select the [Don't care] radio button in the [Software trace] dialog box or select [Delete] from the popup menu that is opened by clicking on the software Trace column with the right-mouse button.

Internal Trace Function: Open the [Acquisition] dialog box by right-clicking on the [Trace] window and then selecting [Setting] from the pop up menu. Enable the internal trace function by selecting the [Internal trace] radio button in the [Trace type] group box on the [Trace mode] page of the [Acquisition] dialog box. Use this function with the required trace condition settings.

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. Use of the synchronized break function so that breaks occur at the same time for both CPUs is recommended. Although information for both CPUs can be acquired at the same time, the results of tracing are not displayed in the trace window for a CPU for which a break has occurred while the other CPU is proceeding with execution. Also, when breaks in execution are not synchronized, filling up is possible according to the trace information for the CPU that is proceeding with execution.

AUD Trace Functions: Open the [Acquisition] dialog box by right-clicking on the [Trace] window and then selecting [Setting] from the pop up menu. Enable the AUD trace function by selecting the [AUD trace] radio button in the [Trace type] group box on the [Trace mode] page of the [Acquisition] dialog box. Use this function with the required trace condition settings.

Table 2.9 shows the AUD trace acquisition mode that can be set in each trace function.

Table 2.9 AUD Trace Acquisition Mode

Type	Mode	Description
Continuous trace occurs	Realtime trace	When the next branch occurs while the trace information is being output, all the information may not be output. The user program can be executed in realtime, but some trace information will 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.
AUD pin	4 bits	The trace data is acquired from the 4-bit AUDATA pin.
	8 bits	The trace data is acquired from the 8-bit AUDATA pin. This mode is not available when the SH7776 is used.
AUD trace memory size	0.25 to 16 Mbytes	Specify a size in the range from 0.25 to 16 Mbytes for the trace buffer of the emulator.

Note: When the AUD trace is enabled, the emulator forcibly changes the pin functions of the specified port as the AUD functions.

Settings for the AUD trace acquisition mode are made in the [Trace mode1], [Trace mode2], [AUD mode], and [AUD Trace Memory Size] group boxes on the [Trace mode] page of the [Acquisition] dialog box.

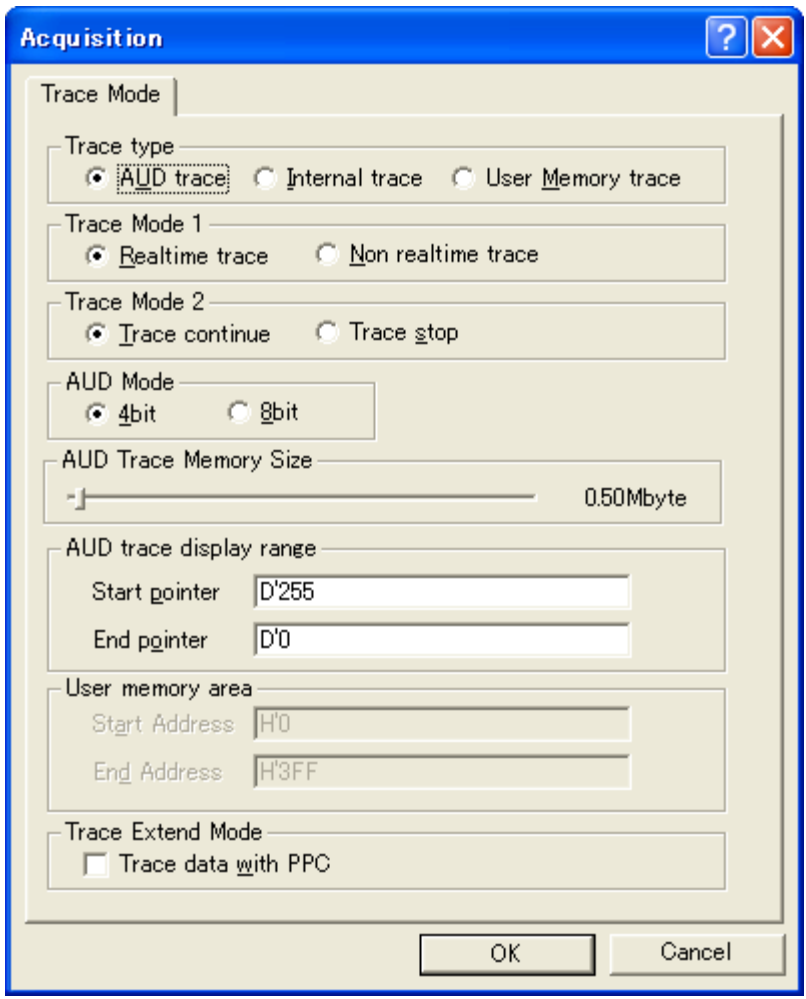


Figure 2.6 [Trace Mode] Page

Notes on AUD Trace:

1. When the trace display is performed during user program execution, the mnemonics, operands are not displayed.
2. The AUD branch 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 previously output 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 emulator, when multiple loops are performed to reduce the number of AUD trace displays, only the IP counts up.
5. In the emulator, the maximum number of lines for the trace display is 131070 (65535 branches).
   If more than 131070 lines are to be acquired, modify the values of the start pointer and end pointer for [AUD trace display range] on the [Trace mode] page of the [Acquisition] dialog box.
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. If a completion-type exception occurs during exception branch acquisition, the next address to the address in which an exception occurs is acquired.
8. Use of the synchronized break function so that breaks occur at the same time for both CPUs is recommended. Also, when breaks in execution are not synchronized, filling up is possible according to the trace information for the CPU that is proceeding with execution.

Memory Output Trace Functions: Open the [Acquisition] dialog box by right-clicking on the [Trace] window and then selecting [Setting] from the pop up menu. Enable the memory output trace function by selecting the [User Memory trace] radio button in the [Trace type] group box on the [Trace mode] page of the [Acquisition] dialog box.
In this function, write the trace data in the specified user memory range.
Specify the start address to output a trace for the [Start] edit box in the [User memory area] group box, and the end address for the [End Address] edit box. Set the trace condition to be used.

Table 2.10 shows the memory-output trace acquisition mode that can be set in each trace function.

Table 2.10 Memory-Output Trace Acquisition Mode

Type	Mode	Description
Continuous trace occurs	Realtime trace	When the next branch occurs while the trace information is being output, all the information may not be output. The user program can be executed in realtime, but some trace information will 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 memory-output 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 [Trace mode1] or [Trace mode2] group box in the [Trace mode] page of the [Acquisition] dialog box.

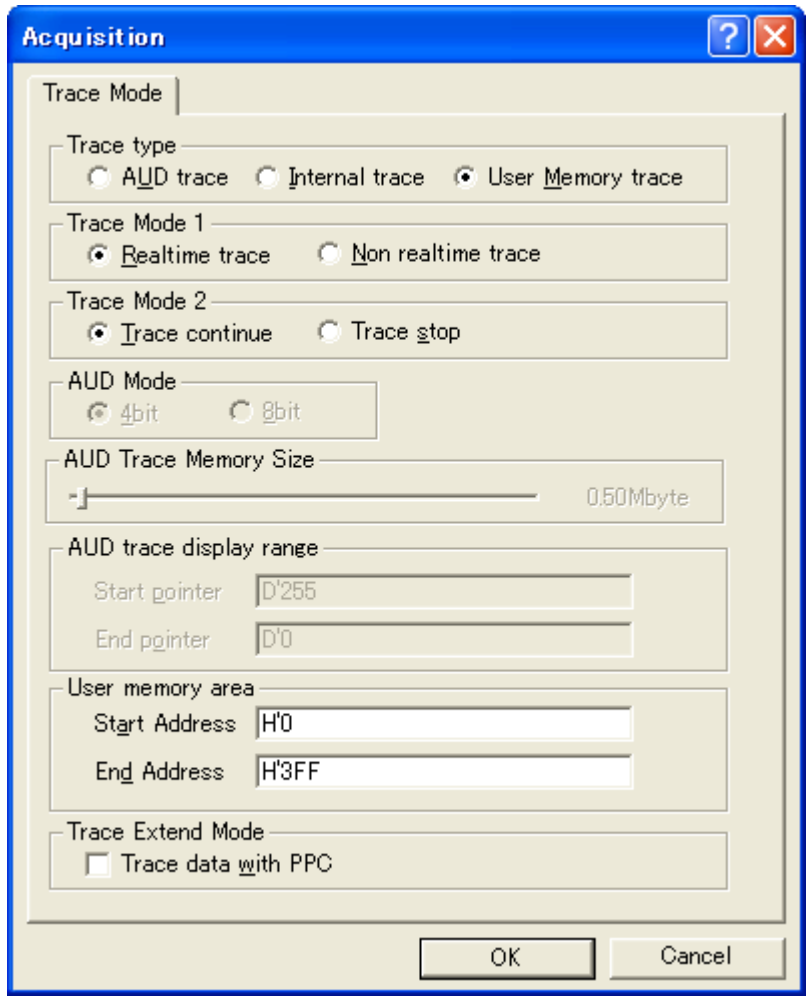


Figure 2.7 [Trace Mode] Page

Notes: 1. The memory range for which trace is output is the address on the system bus and not supported for the MMU or cache.

2. In the memory range for output, do not specify the ranges that the user program has been downloaded or the user program accesses.
3. Do not specify the internal RAM area for the output range.
4. The range for trace output must be 1 MB or less.
5. Use of the synchronized break function so that breaks occur at the same time for both CPUs is recommended. Also, when breaks in execution are not synchronized, filling up is possible according to the trace information for the CPU that is proceeding with execution.

2.2.3 Notes on Using the JTAG (H-UDI) Clock (TCK) and AUD Clock (AUDCK)

1. Set the JTAG clock (TCK) frequency to lower than the frequency of the SH7776 peripheral module clock (CKP).
2. Set the AUD clock (AUDCK) frequency to 50 MHz or lower. If the frequency is higher than 50 MHz, the emulator will not operate normally.
3. The set value of the JTAG clock (TCK) is initialized by executing [Reset CPU] or [Reset Go]. Thus the TCK value will be 5 MHz.
   If the [Search the best JTAG clock] option is used when the emulator is initiated, the TCK value will be initialized as a value that has been automatically acquired.

2.2.4 Notes on Setting the [Breakpoint] Dialog Box

1. A breakpoint can be set only when both debuggers have been synchronized by selecting [Synchronization style], execution is by the [Synchronization options] , and the Break/Halt checkbox has been selected.
2. When an odd address is set, the next lowest even address is used.
   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. A BREAKPOINT cannot be set to the following addresses:
   - ROM areas in CS0 to CS6
   - Areas other than CS0 to CS6 except for the internal RAM
   - A slot instruction of a delayed branch instruction
   - An area that can be only read by MMU
3. During step operation, BREAKPOINTs are disabled.
4. 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.

5. 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.
6. 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 SH7776 MMU status during command input when the VPMAP_SET command setting is disabled. The ASID value of the SH7776 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 SH7776 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.
7. 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 SH7776 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.
8. 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 SH7776 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.
9. 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.
10. 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 [Source] 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 event condition, the mark ● disappears.

2.2.5 Notes on Setting the [CPU Event] Dialog Box and the BREAKCONDITION_SET Command

1. When [Go to cursor], [Step In], [Step Over], or [Step Out] is selected, the settings of CPU Event 3 are disabled.
2. When a CPU Event is satisfied, emulation may stop after two or more instructions have been executed.
3. 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.
4. To restart execution from the address where it stopped due to a CPU Event, if the instruction at that address is executed as a single step and execution is continuous from the next PC value, the real-time characteristic is lost.

2.2.6 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, Ch8 (IA_OA_R) and Ch9 (IA_OA_DT_CT_R) of CPU Event cannot be used.

2.2.7 Note on Setting the PPC_MODE Command

In the [Configuration] dialog box, if [User] is set while the [PPC mode] list box has been set, Ch1 and Ch2 of the performance analysis function cannot be used.

2.2.8 Performance Measurement Function

The emulator supports four channels (Ch1 to Ch4) of 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 a channel 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].

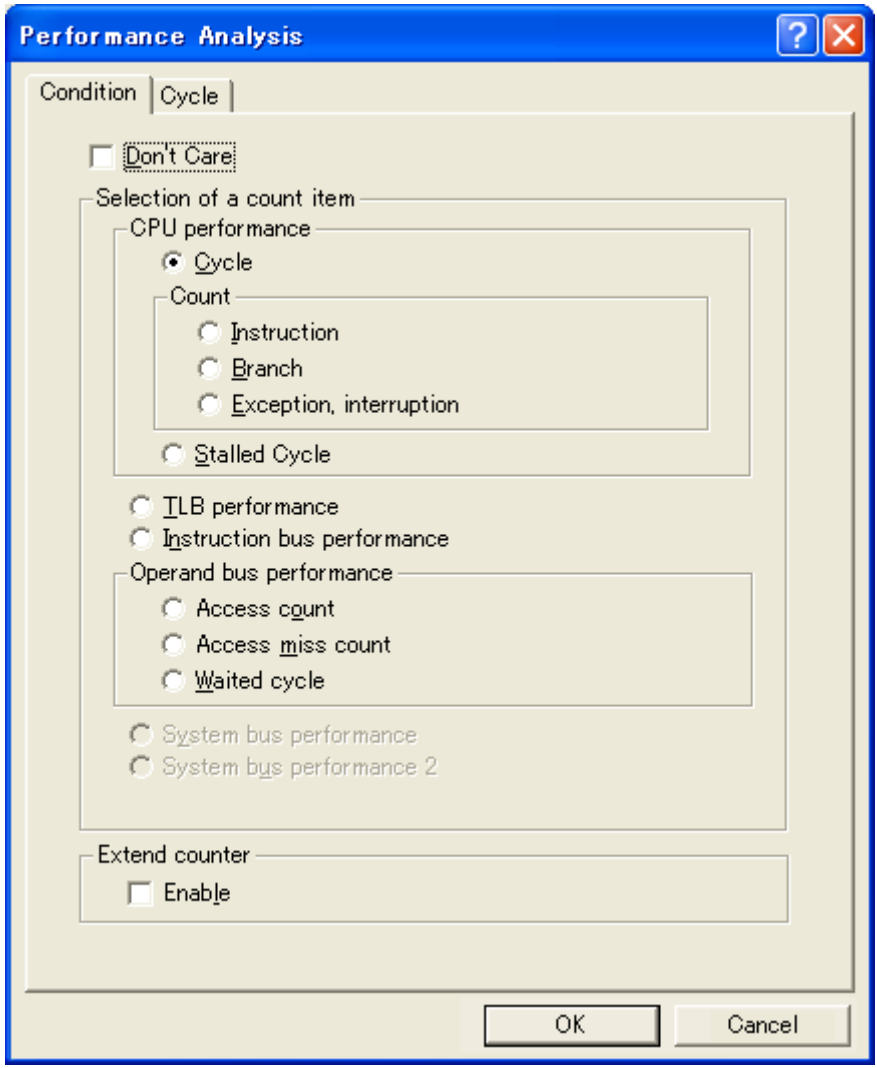


Figure 2.8 [Performance Analysis] Dialog Box

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

(a) Specifying the measurement start/end conditions

Set the performance measurement conditions in the [Action] page after conditions have been set in the [CPU Event] dialog box that is opened by double-clicking Ch1 to Ch6 and Ch8 to Ch10 on the [CPU Event] sheet of the [Action] page.

Notes: 1. When no measurement start/end conditions are specified, measurement is started by executing a program and ended when an event condition is satisfied.

2. When only the measurement start or end condition is specified, performance cannot be measured. Be sure to specify both of the measurement start and end conditions.
3. Step is not possible while start and end conditions for measurement are specified. Also, when execution is restarted from the address where it stopped due to a breakpoint or CPU event break condition, operation is not possible since this requires stepped execution. Restart execution after removing the breakpoint or CPU event break condition.
4. The use of one channel as both a break condition and a start or end condition for measurement is not possible. After specification as a start or end condition for measurement, setting as a break condition is ineffective.

Table 2.11 Conditions Specified in the [Action] Page

Item		Description
PA1	pa1_start_point	Specifies the conditions of CPU Event that has been set as the measurement start condition of performance channel 1.
	pa1_end_point	Specifies the conditions of CPU Event that has been set as the measurement end condition of performance channel 1.
PA2	pa2_start_point	Specifies the conditions of CPU Event that has been set as the measurement start condition of performance channel 2.
	pa2_end_point	Specifies the conditions of CPU Event that has been set as the measurement end condition of performance channel 2.
PA3	pa3_start_point	Specifies the conditions of CPU Event that has been set as the measurement start condition of performance channel 3.
	pa3_end_point	Specifies the conditions of CPU Event that has been set as the measurement end condition of performance channel 3.
PA4	pa4_start_point	Specifies the conditions of CPU Event that has been set as the measurement start condition of performance channel 4.
	pa4_end_point	Specifies the conditions of CPU Event that has been set as the measurement end condition of performance channel 4.

Figure 2.9 [Action] Page

(b) Measurement tolerance

— The measured value includes tolerance.

— Tolerance will be generated before or after a break.

For details, see table 2.14.

(c) Measurement items

Items are measured in the [Performance Analysis] dialog box for each channel from Ch1 to Ch4. A maximum of four conditions can be specified at the same time. Table 2.12 shows the measurement items. (Options in table 2.12 are parameters for <mode> of the PERFORMANCE_SET command. They are displayed in CONDITION of the [Performance Analysis] window.)

Table 2.12 Measurement Items

Classification	Type	Measurement Item	Option	Note
Disabled			None	Not measured.
CPU performance	Cycle	Elapsed cycles	AC	Except for power-on period; counted by the CPU clock.
		Cycles executed in privileged mode	PM	The number of privileged-mode cycles among the number of elapsed cycles.
		Cycles for asserting the SR.BL bit	BL	The number of cycles when the SR.BL bit = 1 among the number of elapsed cycles.
	Instruction	Number of effective instructions issued	I	The number of execution instructions = number of valid instructions issued + number of cases of simultaneous execution of two instructions. The number of valid instructions means the number of completed instructions.
		Number of 2 instruction executed simultaneously	2I	The number of times that two instructions are executed simultaneously among the valid instructions issued.
	Branch	Number of unconditional branch	BT	The number of unconditional branches other than branches occurring after an exception. However, RTE is counted.
	Exception, interruption	Number of exceptions accepted	EA	Interrupts are included.
		Number of interrupts accepted	INT	NMI is included.
		Number of UBC channel hit	UBC	Performs OR to count the number of channel-hits in the CPU.

Table 2.12 Measurement Items (cont)

Classification	Type	Measurement Item	Option	Note
CPU performance (cont)	Stalled cycle	Cycles stalled in full-trace mode (with multi-counts)	SFM	All items are counted independently.
		Cycles stalled in full-trace mode (without multi-counts)	SF	This item is not counted if the stall cycle is generated simultaneously with a stall cycle that has occurred due to instruction execution.
TLB performance	TLB	Number of UTLB miss for instruction fetch	UMI	The number of TLB-miss exceptions generated by an instruction fetch (number of EXPEVT sets).
		Number of UTLB miss for operand fetch	UMO	The number of TLB-miss exceptions generated by an operand access (number of EXPEVT sets).
		Number of ITLB miss	IM	The number of ITLB misses for valid accesses (does not include UTLB hits or misses).
Instruction bus performance	Instruction	Number of memory accesses for instruction fetch	MIF	The number of memory accesses by an instruction fetch. Accesses canceled by an instruction-fetch bus are not counted. Instruction fetches, which have been fetched in anticipation of a branch but not actually executed, are counted. Accesses by the PREFI instruction are included.
		Number of instruction cache access	IC	The number of accesses for an instruction cache during memory access of the opcode.

Table 2.12 Measurement Items (cont)

Classification	Type	Measurement Item	Option	Note
Instruction bus performance (cont)	Instruction (cont)	Number of instruction cache miss	ICM	The number of cache misses by an instruction cache access (the number of accesses to the outside of the CPU core due to a cache miss).
		Number of internal-RAM access for instruction fetch (XY-RAM or O-L memory)	XL	The number of accesses for the XY or O-L memory in the SH7776 during memory accesses of the opcode.
		Number of I-L memory access for instruction fetch	ILIF	The number of accesses for the I-L memory in the SH7776 during memory accesses of the opcode.
Operand bus performance	Access count	Number of memory access for operand fetch (READ)	MR	The number of memory accesses by an operand read (equal to loading on the operand bus). Accesses by the PREF instruction or canceled accesses are not included.
		Number of memory access for operand fetch (WRITE)	MW	The number of memory accesses by an operand write (equal to storing memory on the operand bus). Canceled accesses are not included.
		Number of operand cache access (READ)	CR	The number of operand-cache reads during memory access (read) of an operand.
		Number of operand cache access (WRITE)	CW	The number of operand-cache reads during memory access (write) of an operand.

Table 2.12 Measurement Items (cont)

Classification	Type	Measurement Item	Option	Note
Operand bus performance (cont)	Access count (cont)	Number of internal-RAM access for operand fetch (READ) (XY-RAM or O-L memory)	XLR	The number of accesses to XY or O-L memory in the SH7776 during memory access (read) of an operand. (Accesses via the XY bus and the operand bus are included. When MOVX and MOVY are executed simultaneously, it increments one count regardless of the read or write.)
		Number of internal-RAM access for operand fetch (WRITE) (XY-RAM or O-L memory)	XLW	The number of accesses to XY or O-L memory in the SH7776 during memory access (write) of an operand. (Accesses via the XY bus and the operand bus are included. When MOVX and MOVY are executed simultaneously, it increments one count regardless of the read or write.)
		Number of I-L memory access for operand fetch (READ/WRITE)	ILRW	The number of accesses to I-L memory in the SH7776 during memory access (read/write) of an operand.
	Access miss count	Number of operand cache miss (READ)	CMR	The number of cache misses by an operand cache access (read) (number of accesses to the outside of the CPU core due to a cache miss). Cache misses are not counted by the PREF instruction.
		Number of operand cache miss (WRITE)	CMW	The number of cache misses by an operand cache access (write) (number of accesses to the outside of the CPU core due to a cache miss). Write-through accesses are not counted. Cache misses are not counted by the PREF instruction.

Table 2.12 Measurement Items (cont)

Classification	Type	Measurement Item	Option	Note
Operand bus performance (cont)	Waited cycle	Waited cycles for operand fetch (READ)	WOR	The number of waited cycles by a memory access (read) of an operand.
		Waited cycles for operand fetch (WRITE)	WOW	The number of waited cycles by a memory access (write) of an operand.
		Waited cycles for operand cache miss (READ)	WCMR	The number of waited cycles by an operand cache miss (read) (however, the number of waited cycles of cache fill is included due to contention).
		Waited cycles for operand cache miss (WRITE)	WCMW	The number of waited cycles by an operand cache miss (write).
		Number of waited cycles by an I-L memory access for operand fetch (READ)	WILR	The number of waited cycles by an I-L memory access (read) of an operand.
		Number of waited cycles by an I-L memory access for operand fetch (WRITE)	WILW	The number of waited cycles by an I-L memory access (write) of an operand.

Table 2.13 shows the measurement items and methods that are mainly used.

Table 2.13 Main Measurement Items

Main Measurement Item	Measurement Method
Elapsed time	Number of elapsed cycles x CPU clock cycles
Number of execution instructions	Number of valid instructions issued + number of cases of simultaneous execution of two instructions
Number of interrupts accepted	Number of exceptions accepted
Number of instruction fetches (for both cache and non-cache)	Number of memory accesses in an opcode
Instruction-cache hit ratio	(Number of instruction-cache accesses– instruction-cache miss counts)/instruction-cache access counts
Number of operand accesses (for both cache and non-cache)	Number of memory accesses in an operand (read) + number of memory accesses in an operand (write)
Operand-cache hit ratio (read)	(Number of operand-cache accesses (read) – number of operand-cache misses (read))/number of operand-cache accesses (read)
Operand-cache hit ratio (write)	(Number of operand-cache accesses (write) – number of operand-cache misses (write))/ number of operand-cache accesses (write)
Operand-cache hit ratio	(Number of operand-cache accesses (read) + number of operand-cache accesses (write) – number of operand-cache misses (read) – number of operand-cache misses (write))/(number of operand-cache accesses (read) + number of operand-cache accesses (write))

Each measurement condition is also counted when conditions in table 2.14 are generated.

Table 2.14 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 and memory output 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.

(d) Extension setting of the performance-result storing counter

The 32-bit counter stores the result of performance, and two counters can be used as a 64-bit counter.

To set a 64-bit counter, check the [Enable] check box in the [Extend counter] group box of the [Performance Analysis] dialog box for Ch1 and Ch3.

2. Displaying the result of performance

The result of performance is displayed in the [Performance Analysis] window or the PERFORMANCE_ANALYSIS command in hexadecimal (32 bits).

However, when the extension counter is enabled, it is displayed in hexadecimal (64 bits).

Note: If a performance counter overflows as a result of measurement, "*" will be displayed for upper bits.

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.

Super™ Family E10A-USB Emulator
for Multi-core Microcomputers
Additional Document for User's Manual
Supplementary Information on Using the SH7776

Publication Date: Rev.1.00, November 19, 2009
Published by: Sales Strategic Planning Div.
Renesas Technology Corp.
Edited by: Customer Support Department
Global Strategic Communication Div.
Renesas Solutions Corp.

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 (Shanghai) Co., Ltd.
Unit 204, 205, AZIACenter, No.1233 Lujiazui Ring Rd, Pudong District, Shanghai, China 200120
Tel: <86> (21) 5877-1818, Fax: <86> (21) 6887-7858/7898

Renesas Technology Hong Kong Ltd.
7th Floor, North Tower, World Finance Centre, Harbour City, Canton Road, Tsimshatsui, Kowloon, Hong Kong
Tel: <852> 2265-6688, Fax: <852> 2377-3473

Renesas Technology Taiwan Co., Ltd.
10th Floor, No.99, Fushing North Road, Taipei, Taiwan
Tel: <886> (2) 2715-2888, Fax: <886> (2) 3518-3399

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

Renesas Technology Korea Co., Ltd.
Kukje Center Bldg. 18th Fl., 191, 2-ka, Hangang-ro, Yongsan-ku, Seoul 140-702, Korea
Tel: <82> (2) 796-3115, Fax: <82> (2) 796-2145

Renesas Technology Malaysia Sdn. Bhd
Unit 906, Block B, Menara Amcorp, Amcorp Trade Centre, No.18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: <603> 7955-9390, Fax: <603> 7955-9510

SuperH™ Family E10A-USB Emulator for Multi-core Microcomputers

Additional Document for User's Manual Supplementary Information on Using the SH7776

Renesas Electronics Corporation
1753, Shimonumabe, Nakahara-ku, Kawasaki-shi, Kanagawa 211-8668 Japan

REJ10J2083-0100