

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.

SuperH™ Family E10A Emulator

Additional Document for User's Manual

SH7750 E10A HS7750KCM02HE

Renesas Microcomputer

Development Environment

System

SuperH™ Family / SH7750 Series

Specific Guide for the SH7750

E10A Emulator

Cautions

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

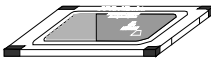
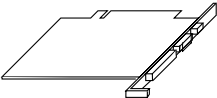
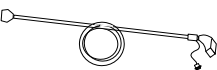
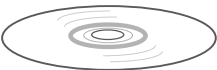
Section 1	Connecting the Emulator with the User System	1
1.1	Components of the Emulator	1
1.2	Connecting the E10A Emulator with the User System	2
1.3	Installing the H-UDI Port Connector on the User System	2
1.4	Pin Assignments of the H-UDI Port Connector	2
1.5	Recommended Circuit between the H-UDI Port Connector and the MPU	4
1.5.1	Recommended Circuit	4
Section 2	Specifications of the SH7750 E10A Emulator's Software	5
2.1	Differences between the SH7750 and SH7750S and the Emulator.....	5
2.2	Specific Functions for the SH7750 E10A Emulator	9
2.2.1	Emulator Driver Selection	9
2.2.2	Break Condition Functions	10
2.2.3	Trace Functions.....	12
2.2.4	Notes on Using the JTAG Clock (TCK) and AUD Clock (AUDCK).....	16
2.2.5	Notes on Setting the [Breakpoint] Dialog Box	16
2.2.6	Notes on Setting the [Break Condition] Dialog Box and the BREAKCONDITION_ SET Command	18
2.2.7	Notes on Setting the UBC_MODE Command.....	18
2.2.8	Performance Measurement Function	18
2.2.9	Note on Using the Profile Function	25
2.2.10	Interrupts	25

Section 1 Connecting the Emulator with the User System

1.1 Components of the Emulator

The SH7750 E10A emulator supports the SH7750 and SH7750S. Table 1.1 lists the components of the emulator.

Table 1.1 Components of the Emulator (HS7750KCM01H or HS7750KCI01H)

Classification	Component	Appearance	Quantity	Remarks
Hardware	Card emulator	 <p>(PCMCIA)</p>	1	HS7750KCM01H (PCMCIA: 14-pin type): Depth: 85.6 mm, Width: 54.0 mm, Height: 5.0 mm, Weight: 27.0 g HS7750KCI01H (PCI: 14-pin type): Depth: 144.0 mm, Width: 105.0 mm, Mass: 93.0 g
		<p>or</p>  <p>(PCI)</p>		
	User system interface cable		1	HS7750KCM01H (PCMCIA: 14-pin type): Length: 80 cm, Weight: 45.0 g HS7750KCI01H (PCI: 14-pin type): Length: 150 cm, Mass: 86.0 g
Software	SH7750 E10A emulator setup program, SuperH™ Family E10A Emulator User's Manual, and Specific Guide for the SH7750 E10A Emulator		1	HS7750KCM01SR, HS0005KCM01HJ, HS0005KCM01HE, HS7750KCM02HJ, and HS7750KCM02HE (provided on a CD-R)

1.2 Connecting the E10A Emulator with the User System

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

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

Table 1.2 shows the recommended H-UDI port connectors for the emulator.

Table 1.2 Recommended H-UDI Port Connectors

Connector	Type Number	Manufacturer	Specifications
14-pin connector	2514-6002	Minnesota Mining & Manufacturing Ltd.	14-pin straight type

Note: Do not install any components within 3 mm of the H-UDI port connector.

1.4 Pin Assignments of the H-UDI Port Connector

Figure 1.1 shows the pin assignments of the H-UDI port connector.

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	SH7750 Pin No.	
			BGA256	HQFP208
1	TCK	Input	A-5	198
2*2	/TRST	Input	C-4	200
3	TDO	Output	A-6	194
4*2	/ASEBRK BRKACK	Input/ Output	B-7	193
5	TMS	Input	B-6	197
6	TDI	Input	B-5	199
7*2	/RESET	Output	B-1	2
11	Not connected	—	—	—
8 to 10 and 12 to 13	GND	—	—	—
14*3	GND	Output	—	—

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

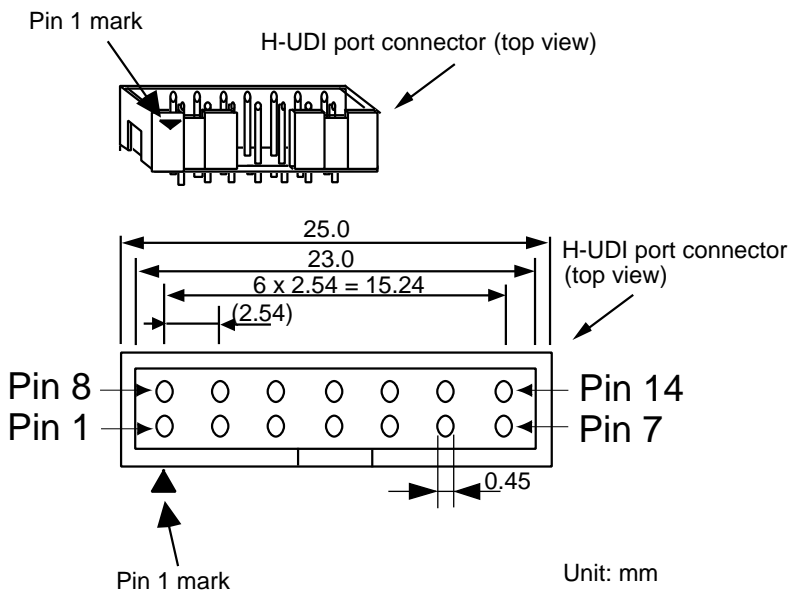


Figure 1.1 Pin Assignments of the H-UDI Port Connector

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

1.5.1 Recommended Circuit

Figure 1.2 shows a recommended circuit for connection 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. When a network resistance is used for pull-up, it may be affected by a noise. Separate TCK from other resistances.
 3. The reset signal in the user side is input to the /RESET pin (pin 198) of the SH7750. Connect this pin to the H-UDI port connector as the output from the user system.
 4. 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.
 5. The resistance values shown in figure 1.2 are recommended.
 6. For the pin processing in cases where the emulator is not used, refer to the hardware manual of the related device.

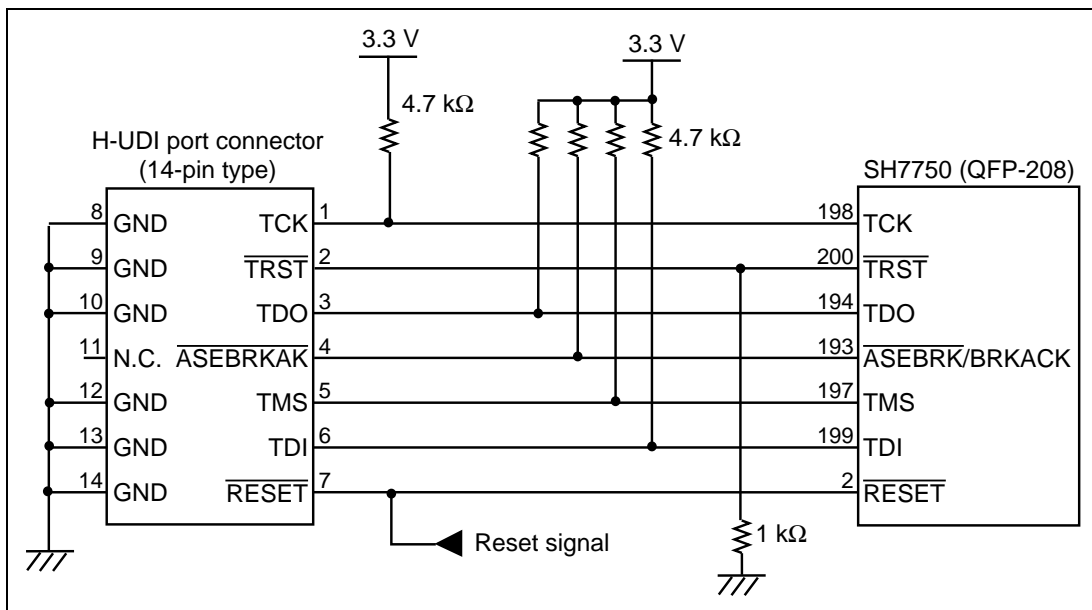


Figure 1.2 Recommended Circuit for Connection between the H-UDI Port Connector and MPU

Section 2 Specifications of the SH7750 E10A Emulator's Software

2.1 Differences between the SH7750 and SH7750S 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.

Table 2.1 Register Initial Values at Emulator Link Up

Register	Emulator at Link Up
R0 to R14	H'00000000
R15 (SP)	H'00000000
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

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

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

For low-power consumption, the SH7750 and SH7750S have sleep, standby, and module standby modes.

The sleep and standby modes are switched using the SLEEP instruction. When the emulator is used, the sleep mode can be cleared by either normal clearing or by the satisfaction of a break condition (including BREAK key input). In the latter case, the user program breaks. The standby mode can be cleared with the normal clearing function or BREAK key input, and after the standby mode is cleared, the user program operates correctly. Note, however, that if a command has been entered in standby mode or module standby mode, no commands can be used from the emulator after the standby mode is cleared.

Note: After the sleep mode is cleared by a break, execution restarts at the instruction following the SLEEP instruction.

If the memory is accessed or modified in sleep mode, the sleep mode is cleared and execution starts at the instruction following the SLEEP instruction.

Although the SH7750S supports the hardware standby function, if the SH7750 E10A emulator enters the hardware standby mode, a TIMEOUT error will occur.

When the SLEEP instruction is executed by a step command and [Step...] in the [Run] menu is used, set [Rate] as 6. If 5 or lower value is set, a communication timeout error will occur.

4. Reset Signals

The SH7750 and SH7750S 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 SH7750 or SH7750S.

Note: Do not start user program execution when the /RESET, /BREQ, or /RDY signal is being low. A TIMEOUT error will occur.

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: 1 GHz (Pentium® III)

OS: Windows® 2000

SH7750: 50 MHz (CPU clock)

JTAG clock: 15 MHz

When a one-byte memory is read from the command-line window, the stopping time will be about 45 ms.

7. Interrupt

When the NMIB bit in the ICR register is 1, the NMI interrupt is accepted during break and the program is executed from the NMI interrupt vector. If the program cannot return normally from the NMI interrupt routine or the value in the general-purpose register is not guaranteed, a communication timeout error will occur.

8. 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.

When the memory area can be written by the MMU, do not perform memory write, BREAKPOINT break, or downloading.

9. 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.

10. 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 HS7750KCI01H is used: TCK = 4.125 MHz
- When HS7750KCM01H is used: TCK = 3.75 MHz

11. [IO] Window

- Display and modification

When [End] is set in the [UBC Mode] list box of the [Configuration] dialog box, 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 (SH7750.IO) and then activate the HEW. After the I/O-register definition 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.

Note: As default, SDMR2 and SDMR3 are specified in the I/O-register definition file as the SDMR registers in areas 2 and 3, respectively.

When the SH7750S is used, the IPRD register is not displayed in the [IO] window. To get it to display, edit the I/O-register definition file (SH7750.IO) as follows and start the HEW:

In SH7750.IO, locate 'IPRC = 0xFFD0000C W A'. Under this, add 'IPRD = 0xFFD00010 W A' and save the file.

- Verify

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

12. 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 SH7750 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
HS7750KCM01H	E10A PC Card Driver 3
HS7750KCI01H	E10A PCI Card Driver 3

2.2.2 Break Condition Functions

In addition to BREAKPOINT functions, the emulator has Break Condition functions. Five types of conditions can be set under Break Condition 1, 2, 3, 4, 5. 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 SH7750 or SH7750S address bus value or the program counter value matches the specified value.
Data bus condition (Data)	Breaks when the SH7750 or SH7750S data bus value matches the specified value. Byte, word, or longword can be specified as the access data size.
ASID condition (ASID)	Breaks when the SH7750 or SH7750S ASID value matches the specified condition.
Bus state condition (Bus State)	There are two bus state condition settings: Read/write condition: Breaks in the read or write cycle of the SH7750 or SH7750S. Bus state condition: Breaks when the operating state in an SH7750 or SH7750S bus cycle matches the specified condition.
LDTLB instruction break condition	Breaks when the SH7750 or SH7750S executes the LDTLB instruction.
Internal I/O break condition	Breaks when the SH7750 or SH7750S accesses the internal I/O.

Note: For the window function or command-line syntax, refer to the online help.

Table 2.5 lists the combinations of conditions that can be set under Break Condition 1, 2, 3, 4, 5, 6, 7, 8.

Table 2.5 Dialog Boxes for Setting Break Conditions

	Dialog Box		
	[Break Condition 1] Dialog Box	[Break Condition 2, 3, 4] Dialog Box	[Break Condition 5] Dialog Box
Address bus condition (Address)	O	O	X
Data bus condition (Data)	O	X	X
ASID condition (ASID)	O	O	X
Read/write specification	O	O	X
Data access	O	O	X
Before/after execution	O	O	X
Sequential break	O	O	X
LDTLB instruction break	X	X	O
Internal I/O access break	X	X	O

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

- Notes:
1. If the BL bit of the SR register is 1, do not use BREAKPOINTS.
 2. If a break is specified for an address that is close to an address whose instruction generates a manual reset, a manual reset may be generated instead of a break. Therefore, to ensure the performance of a break, specify a break for an address that is four addresses before the address whose instruction generates an exception.

The SH7750 E10A emulator has sequential break functions. Table 2.6 shows the sequential break conditions.

Table 2.6 Sequential Break Conditions

Break Condition	Description
Sequential break condition 2-1	Program is halted when Break Condition 2 and Break Condition 1 are satisfied in that order. Break Condition 2,1 should be set.
Sequential break condition 3-2-1	Program is halted when Break Condition 3, Break Condition 2, and Break Condition 1 are satisfied in that order. Break Condition 3,2,1 should be set.
Sequential break condition 4-3-2-1	Program is halted when Break Condition 4, Break Condition 3, Break Condition 2, and Break Condition 1 are satisfied in that order. Break Condition 4,3,2,1 should be set.

Note: Sequential breaks can be specified by the [Configuration] dialog box.

2.2.3 Trace Functions

The SH7750 E10A emulator does not support the AUD function.

Table 2.7 shows the internal trace functions.

Table 2.7 Internal Trace Functions

Function	Description
Branch instruction trace	<p>Traces and displays the branch instructions. The branch source address and branch destination address for the eight latest branch instructions are displayed. There are three kinds of branch instruction trace:</p> <ul style="list-style-type: none">• Normal branch instruction trace Traces and displays the normal branch instructions. The normal branch instructions are the BF, BF/S, BT/S, BRA, BRAF, and JMP instructions. To use this function, select the [Acquire normal branch instruction trace] check box of the [Branch trace] page.• Subroutine branch instruction trace Traces and displays the subroutine branch instructions. The subroutine branch instructions are the BSR, BSRF, JSR, and RTS instructions. To use this function, select the [Acquire subroutine branch instruction trace] check box of the [Branch trace] page• Exception branch instruction trace Traces and displays the exception branch instruction. The exception branch instruction is the RTE instruction. In addition, all the exception and interrupt operations are traced. To use this function, select the [Acquire exceptional branch instruction trace] check box of the [Branch trace] page.
Continous trace	<p>Acquires the trace information continuously. This is called continuous trace. For the branch instruction trace, eight-branch information can be repeatedly acquired a maximum of four times. Select the [Acquire continuous trace] check box of the [Branch trace] page. If continuous trace is selected, realtime trace cannot be performed.</p>
Internal I/O trace	<p>Traces and displays the address and data that access the internal I/O area. To use this function, select the [Get trace information of internal I/O Area] radio button in the [Break Condition 5] dialog box and the [Acquire continuous trace] check box of the [Branch trace] page.</p>
LDTLB instruction execution trace	<p>Traces and displays the address that executes the LDTLB instruction. To use this function, select the [Get trace information of LDTLB instruction] radio button in the [Break Condition 5] dialog box and the [Acquire continuous trace] check box of the [Branch trace] page.</p>

- Notes:
1. For the SH7750 E10A emulator, trace acquisition of the eight latest branch instructions is enabled.
 2. 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 ***
 3. If a TLB error occurs while acquired trace information is displayed, the following error message will be displayed.



Figure 2.1 Message Box for Clearing a TLB Error

After a TLB error, trace acquisition cannot be performed.

4. When MMU settings are modified or when a user program is changed between GO command completion and trace display, the displayed mnemonics or operand may not be correct.
 5. If a completion-type exception occurs during exception branch acquisition, the next address to the address in which an exception occurs is acquired.
 6. When a user interrupt is enabled by the INTERRUPT command during the emulator command wait state or user program execution, an interrupt that is generated at the program execution start or end, including a step operation, can be traced in realtime.
1. Notes on the [Trace] Window
- (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 TLB error occurs while acquired trace information is displayed, the following error message will be displayed.



Figure 2.2 Message Box for Clearing a TLB Error

After a TLB error, trace acquisition cannot be performed.

- (3) When MMU settings are modified or when a user program is changed between GO command completion and trace display, the displayed mnemonics or operand may not be correct.
 - (4) If a completion-type exception occurs during exception branch acquisition, the next address to the address in which an exception occurs is acquired.
 - (5) When a user interrupt is enabled by the INTERRUPT command during the emulator command wait state or user program execution, an interrupt that is generated at the program execution start or end, including a step operation, can be traced in realtime.
2. Notes on Setting the [Trace Acquisition] Window
- (1) When the [Acquire continuous trace] check box is selected, do not perform memory access during emulation.
 - (2) When internal I/O trace or LDTLB instruction trace is performed, select the [Acquire continuous trace] check box.
 - (3) When the [Acquire continuous trace] check box is selected, 32 trace information data can be acquired. In this case, however, since the user program stops at constant intervals, the processing speed is decreased compared with the case where the [Acquire continuous trace] check box is not selected.
 - (4) 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
 - (5) When the [Acquire continuous trace] check box is selected, and when either the [Get trace information of internal I/O area] radio button (internal I/O trace enabled) or the [Get trace information of LDTLB instruction] radio button is selected (LDTLB instruction trace enabled) with the [Break Condition 5] dialog box,
 - An internal I/O trace cannot be made with the Step In command.
 - The LDTLB instruction and internal I/O trace cannot be performed with the Step Over command.

- (6) Do not use the continuous trace for a program in which an SGR value is referred to with the interrupt handler. In the SH7750 E10A emulator, the contents of the SGR register are lost when the user program breaks. Since the user program execution stops at constant intervals while the continuous trace is selected, the contents of the SGR register will be lost.
- (7) When continuous trace is used, do not enable user interrupt by the INTERRUPT command during the emulator command wait state or user program execution.

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

Set the JTAG clock (TCK) frequency to lower than the frequency of the SH7750 or SH7750S peripheral module clock (CKP).

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. Accordingly, it can be set only to the internal RAM area. However, a BREAKPOINT cannot be set to the following addresses:
 - An address whose memory content is H'003B
 - An area other than the CS0 to CS6 areas and the internal RAM area
 - An instruction in which Break Condition 4 is satisfied
 - A slot instruction of a delayed branch instructionIn addition, do not perform memory write, BREAKPOINT, or download even if the memory space can only be written by the MMU.
3. During step operation, a BREAKPOINT is disabled.
4. Conditions set at Break Condition 3 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 3 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 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 SH7750 or SH7750S MMU state during command input when the VPMAP_SET command setting is disabled. The ASID value of the SH7750 or SH7750S 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 SH7750 or SH7750S MMU state during command input. Even when the VP_MAP table is modified after BREAKPOINT setting, the address translated at BREAKPOINT setting is valid.

8. 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 SH7750 or SH7750S MMU during 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 [System Status] window is ILLEGAL INSTRUCTION, not BREAKPOINT.
9. 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 SH7750 or SH7750S MMU during 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.
10. If a TLB error occurs during virtual address setting, the following message box will be displayed.



Figure 2.3 Message Box for Clearing a TLB-Error

If a program is executed again without clearing the BREAKPOINT set at the address in which the TLB error occurs, a TLB error will occur again. Accordingly, clear the BREAKPOINT before execution.

11. 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 [System Status] window is ILLEGAL INSTRUCTION, not BREAKPOINT.

12. 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.
13. While a BREAKPOINT is set, the contents of the instruction cache are disabled at execution completion.

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

1. When [Go to cursor], [Step In], [Step Over], or [Step Out] is selected, the settings of Break Condition 4 are disabled.
2. Break Condition 4 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 4.
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.

2.2.7 Notes 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 SH7750 E10A emulator supports the performance measurement function.

1. Setting the performance measurement conditions
To set the performance measurement conditions, use the [CPU Performance] 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 [CPU Performance] dialog box is displayed by selecting [Setting].

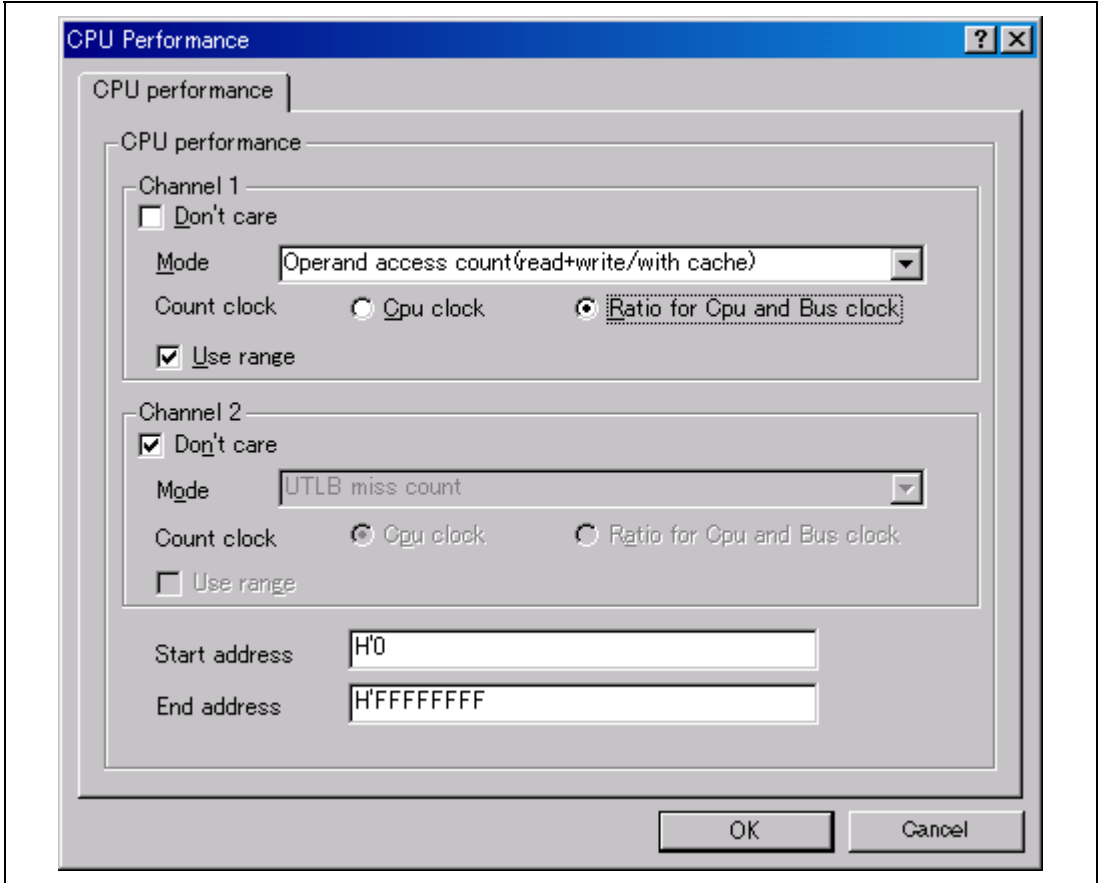


Figure 2.4 [CPU Performance] Dialog Box

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

The emulator measures how many times the conditions of the user program specified with the performance analysis function are satisfied. For this function, two events can be measured simultaneously and the following conditions can be specified. When the PC value is set, the original UBC function is not available because the UBC is used to specify the measurement start and end PC values.

(a) Measurement range

One of the following ranges can be specified by either of measurement channels 1 and 2.

1. From the start to the end of the user program execution
2. From the satisfaction of the condition set in [Start address] to the satisfaction of the condition set in [End address]

When the first range is specified, the measurement result includes a several-cycle error for one user program execution. Therefore, do not specify this range when the step is to be executed. In addition, the user program execution stops when continuous trace is used; again, do not specify the first range in this case.

Note: When the range is specified, be sure to select the [Use range] check box, set the measurement start and end conditions for [Start address] and [End address], respectively, and then execute the user program.

(b) Measurement item

Items are measured with [Channel 1 to 2] in the [CPU Performance] dialog box. Maximum two conditions can be specified at the same time. Table 2.8 shows the measurement items (Options in table 2.8 are parameters for <mode> of the PERFORMANCE_SET command. They are displayed for NAME in the [Performance Analysis] window).

Table 2.8 Measurement Items

Event	Keyword	Description
Operand access count (read and write/with cache)	OARW*	The number of times the operand access is performed on the cacheable area when the cache is enabled (both read and write accesses).
Internal RAM operand access count	OARAM	The number of times the internal RAM area is accessed.
All operand access count	OA	The number of all operand accesses.
Internal I/O area access count	IOA	The number of times the internal I/O area is accessed.
Operand cache read and write miss count	DCRW	The number of times operand cache misses occur at data reading or writing.
Instruction cache miss count	EC	The number of times instruction cache misses.
UTLB miss count	DT	The number of times UTLB misses occur at data access.
Instruction TLB miss count (ITLB and UTLB misses)	ET	The number of times UTLB and ITLB misses occur at instruction access.
Instruction fetch count	EF*	The number of times instructions are fetched from the cacheable area when the cache is enabled.
All instruction fetch count	EA	The number of times all instructions are fetched.
Branch instruction execution count	B	The number of times branch instructions are issued (instructions to be counted: BF (other than displacement 0), BF/S and BT (other than displacement 0), BT/S, BRA, BRAF, and JMP).
Branch taken count	BT	The number of times branches are taken (branches to be counted are the same as mode B).
Instruction execution count	E	The number of times instructions are issued.

Table 2.8 Measurement Items (cont)

Event	Keyword	Description
Two-instruction concurrent execution count	E2	The number of times two instructions are issued at the same time.
FPU instruction execution count	EFP	The number of times FPU instruction is issued.
TRAPA instruction execution count	ETR	The number of times the TRAPA instruction is executed.
Interrupt count (normal)	INT	The number of interrupts (generally except for NMI).
Interrupt count (NMI)	NMI	The number of NMI interrupts.
Instruction cache-fill cycle	ECF	The number of instruction cache-fill cycles.
Operand cache-fill cycle	OCF	The number of operand cache-fill cycles.
Elapsed-time cycle	TM	The number of cycles for elapsed time.

Note: For the non-cache operand accesses due to the PREF instruction or TLB.c=0, the correct value cannot be counted.

The events can be counted even in the conditions shown in table 2.9, in addition to the normal count conditions.

Table 2.9 Performance Count Conditions

Event	Count Condition	Target Mode
All count conditions	When the event to be counted up is canceled by an exception.	All
Instruction cache miss count	<ul style="list-style-type: none"> Includes instruction fetch for the cache-off area to count the number of times the instruction has not been fetched in one cycle. When a cache miss occurs during an overrun fetch generated at exception. 	EC
TLB miss count	When the TLB miss is canceled by an exception having a higher priority than that of the TLB miss	DT and ET
Instruction fetch count	<ul style="list-style-type: none"> When the instruction fetch request by the CPU is accepted. Does not count when the cache is bypassed from the external bus to supply the instruction to the CPU at instruction cache miss. 	EF and EA
Instruction issue count	Counts one when two instructions are issued at the same time.	E
	Counts one to three when instruction fetch exception (instruction address error, instruction TLB miss exception, or instruction TLB protection violation exception) occurs.	E and E2
FPU instruction issue count	<ul style="list-style-type: none"> Counts one when two instructions are issued at the same time. The following shows the FPU instructions: LDS Rm, FPUL, LDS.L @Rm+, FPUL, LDS Rm, FPSCR, LDS.L @Rm+, FPSCR, STS FPUL, Rn, STS.L FPUL, @-Rn, STS FPSCR, Rn, STS.L FPSCR, @-Rn Others: instructions that the instruction code is H'Fxxx 	EFP

(c) Counting method

One of the following methods can be specified by each of measurement channels 1 and 2.

1. Counted by the CPU operating clock
2. Counted by the ratio of the CPU operating clock to the bus clock

When the above method 1 is specified, one CPU operating clock cycle is counted as one.

When method 2 is specified, the count is incremented by 3, 4, 6, 8, 12, or 24, according to the clock frequency ratio (ratio of the CPU clock to the bus clock). In this case, the execution time can be calculated by the following expression:

$$T = C \times B / 24 \quad (T: \text{Execution time}; B: \text{Time of one bus clock cycle}; C: \text{Count})$$

When the ratio of the CPU clock to the bus clock is changed in the user program, it is recommended to select method 2, above, to count the number of cycles.

The following shows examples to measure the performance of the user program by the performance measurement function.

(i) Measuring cache hit ratio

Specify measurement channel 1 to count the cache misses (for data read and write) and specify measurement channel 2 to count operand accesses (read and write) to the cacheable area while the cache is enabled. Specify, with both the channels, the measurement from the start to the end of user program execution.

With the above command settings, the cache miss count and the access count to the cacheable area can be measured, and the cache hit ratio in the executed user program can be obtained.

(ii) Measuring ratio of execution time in specified program area to total execution time

Specify measurement channel 1 to measure the elapsed cycle count from the start to the end of user program execution. Specify measurement channel 2 to measure the elapsed cycle count during execution from the specified start PC to the specified end PC.

With both the channels, the total elapsed cycle and the elapsed cycle for the specified program area can be measured, and the ratio of the execution time in the specified program area to the total execution time can be obtained.

- Notes:
1. The counter for performance measurement has 48 bits. A maximum of $2^{48} = 2.8 \times 10^{14}$ counts and 16.3-day cycles (when the CPU operating frequency is 200 MHz) can be measured. If a counter overflow occurs, the count becomes invalid.
 2. When performance measurement conditions are set, canceled, or initialized, the settings in the UBC are not guaranteed.
 3. Set the same start and end PC values for both channels 1 and 2. If different PC values are set, the last settings become valid.
 4. When the start and end PC values are set with this command, the value that has been previously set for UBC becomes invalid.
 5. For details on command-line syntax, refer to the online help.

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.

2.2.9 Note on Using the Profile Function

While the profile function is being used, in addition to the description in section 5.19.12 in the Debugger Part of the SuperH™ Family E10A Emulator User's Manual, the following functions cannot be used.

1. Continuous trace function

When the profile function is enabled, do not use the continuous trace function that can be used in the internal trace function. The profile data cannot be measured correctly.

2. Internal trace function

When the profile function is enabled, mode selection of the internal trace is disabled since all items of the internal trace modes are selected in the emulator.

3. Halt function

When the profile function is enabled, do not use the halt function for the internal or AUD trace.

2.2.10 Interrupts

During emulation, any interrupt to the SH7750 or SH7750S can be used. Whether or not to process interrupts during emulator command execution or in command input wait state can be specified.

— When no interrupt is processed during user program execution or in command input wait state

While the emulator is executing the user program or is in command input wait state, interrupts are not processed generally. However, if an internal interrupt or an edge sensitive external interrupt occurs in command input wait state, the emulator holds the interrupt and executes the interrupt processing routine when the GO command is entered.

— When interrupts are processed during user program execution or in command input wait state

To process NMIs and peripheral module interrupts during emulator command execution and in command input wait state, use the INTERRUPT command. Input the INTERRUPT command into the command line window.

- To process only NMIs

Specify "nonmask" in the <interrupt_enable> option of the INTERRUPT command.

- To process NMIs and peripheral module interrupts
Specify "all" in the <interrupt_enable> option of the INTERRUPT command.
- To switch to the mode in which no interrupt is processed:
Specify "disable" in the <interrupt_enable> option of the INTERRUPT command.

- Notes:
1. Check that the interrupt handler operates normally before using this function. In addition, do not execute a non-limited loop or the sleep instruction in the interrupt handler. If the processing of the handler does not end, the emulator generates a Communication Timeout error.
 2. When interrupts are accepted during user program execution and emulator command execution state, user interrupt processing is not traced. In this case, continuous trace is not enabled.
 3. Use the NOP instruction at the delay slot after the RTE instruction in the interrupt handler.
 4. If a user interrupt is inserted while the user program breaks until the processing ends, do not set a BREAKPOINT in the interrupt handler. The emulator may generate a Communication Timeout error. Use the Break Condition function.
 5. For details on window function and command-line syntax, refer to the online help.

SuperH™ Family E10A Emulator
Additional Document for User's Manual
Specific Guide for the SH7750 E10A Emulator

Publication Date: Rev.1.00, February 17, 2004

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

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

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



Renesas Electronics Corporation

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

REJ10B0110-0100H