Everywhere you imagine.

RENESAS

SH7046 E6000H Emulator
User's Manual

Renesas Microcomputer Development Environment System
SuperH™ Family / SH7046 Series
SuperH™ Family / SH7047 Series
SuperH™ Family / SH7144 Series
HS7046EPH60HE

User's Manual

Rev.4.00
Revision Date: Oct. 21, 2005

Renesas Technology
www.renesas.com

IMPORTANT INFORMATION

READ FIRST

• **READ this user's manual before using this emulator product.**

• **<u>KEEP the user's manual handy for future reference.</u>**

Do not attempt to use the emulator product until you fully understand its mechanism.

Emulator Product:

Throughout this document, the term "emulator product" shall be defined as the following products produced only by Renesas Technology Corp. excluding all subsidiary products.

- Emulator station
- PC interface board
- User system interface board
- Cable

The user system or a host computer is not included in this definition.

Purpose of the Emulator Product:

This emulator product is a software and hardware development tool for systems employing the Renesas microcomputer. This emulator product must only be used for the above purpose.

Limited Applications:

This emulator product is not authorized for use in MEDICAL, atomic energy, aeronautical or space technology applications without consent of the appropriate officer of a Renesas sales company. Such use includes, but is not limited to, use in life support systems. Buyers of this emulator product must notify the relevant Renesas sales offices before planning to use the product in such applications.

Improvement Policy:

Renesas Technology Corp. (including its subsidiaries, hereafter collectively referred to as Renesas) pursues a policy of continuing improvement in design, performance, and safety of the emulator product. Renesas reserves the right to change, wholly or partially, the specifications, design, user's manual, and other documentation at any time without notice.

Target User of the Emulator Product:

This emulator product should only be used by those who have carefully read and thoroughly understood the information and restrictions contained in the user's manual. Do not attempt to use the emulator product until you fully understand its mechanism.

It is highly recommended that first-time users be instructed by users that are well versed in the operation of the emulator product.

RENESAS

LIMITED WARRANTY

 Renesas warrants its emulator products to be manufactured in accordance with published specifications and free from defects in material and/or workmanship. Renesas, at its option, will repair or replace any emulator products returned intact to the factory, transportation charges prepaid, which Renesas, upon inspection, determine to be defective in material and/or workmanship. The foregoing shall constitute the sole remedy for any breach of Renesas' warranty. See the Renesas warranty booklet for details on the warranty period. This warranty extends only to you, the original Purchaser. It is not transferable to anyone who subsequently purchases the emulator product from you. Renesas is not liable for any claim made by a third party or made by you for a third party.

DISCLAIMER

 RENESAS MAKES NO WARRANTIES, EITHER EXPRESS OR IMPLIED, ORAL OR WRITTEN, EXCEPT AS PROVIDED HEREIN, INCLUDING WITHOUT LIMITATION THEREOF, WARRANTIES AS TO MARKETABILITY, MERCHANTABILITY, FITNESS FOR ANY PARTICULAR PURPOSE OR USE, OR AGAINST INFRINGEMENT OF ANY PATENT. IN NO EVENT SHALL RENESAS BE LIABLE FOR ANY DIRECT, INCIDENTAL OR CONSEQUENTIAL DAMAGES OF ANY NATURE, OR LOSSES OR EXPENSES RESULTING FROM ANY DEFECTIVE EMULATOR PRODUCT, THE USE OF ANY EMULATOR PRODUCT, OR ITS DOCUMENTATION, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. EXCEPT AS EXPRESSLY STATED OTHERWISE IN THIS WARRANTY, THIS EMULATOR PRODUCT IS SOLD "AS IS ", AND YOU MUST ASSUME ALL RISK FOR THE USE AND RESULTS OBTAINED FROM THE EMULATOR PRODUCT.

RENESAS

State Law:

Some states do not allow the exclusion or limitation of implied warranties or liability for incidental or consequential damages, so the above limitation or exclusion may not apply to you. This warranty gives you specific legal rights, and you may have other rights which may vary from state to state.

The Warranty is Void in the Following Cases:

Renesas shall have no liability or legal responsibility for any problems caused by misuse, abuse, misapplication, neglect, improper handling, installation, repair or modifications of the emulator product without Renesas' prior written consent or any problems caused by the user system.

All Rights Reserved:

This user's manual and emulator product are copyrighted and all rights are reserved by Renesas. No part of this user's manual, all or part, may be reproduced or duplicated in any form, in hard-copy or machine-readable form, by any means available without Renesas' prior written consent.

Other Important Things to Keep in Mind:

1. Circuitry and other examples described herein are meant merely to indicate the characteristics and performance of Renesas' semiconductor products. Renesas assumes no responsibility for any intellectual property claims or other problems that may result from applications based on the examples described herein.

2. No license is granted by implication or otherwise under any patents or other rights of any third party or Renesas.

Figures:

Some figures in this user's manual may show items different from your actual system.

Limited Anticipation of Danger:

Renesas cannot anticipate every possible circumstance that might involve a potential hazard. The warnings in this user's manual and on the emulator product are therefore not all inclusive. Therefore, you must use the emulator product safely at your own risk.

RENESAS

SAFETY PAGE

READ FIRST

• **READ this user's manual before using this emulator product.**

• **KEEP the user's manual handy for future reference.**

Do not attempt to use the emulator product until you fully understand its mechanism.

DEFINITION OF SIGNAL WORDS

This is the safety alert symbol. It is used to alert you to potential personal injury hazards. Obey all safety messages that follow this symbol to avoid possible injury or death.

⚠ DANGER **DANGER** indicates an imminently hazardous situation which, if not avoided, will result in death or serious injury.

⚠ WARNING **WARNING** indicates a potentially hazardous situation which, if not avoided, could result in death or serious injury.

⚠ CAUTION **CAUTION** indicates a potentially hazardous situation which, if not avoided, may result in minor or moderate injury.

CAUTION **CAUTION** used without the safety alert symbol indicates a potentially hazardous situation which, if not avoided, may result in property damage.

NOTE emphasizes essential information.

RENESAS

⚠ WARNING

Observe the precautions listed below. Failure to do so will result in a FIRE HAZARD and will damage the user system and the emulator product or will result in PERSONAL INJURY. The USER PROGRAM will be LOST.

1. Carefully handle the emulator product to prevent receiving an electric shock because the emulator product has a DC power supply. Do not repair or remodel the emulator product by yourself for electric shock prevention and quality assurance.

2. Always switch OFF the emulator and user system before connecting or disconnecting any CABLES or PARTS.

3. Always before connecting, make sure that pin 1 on both sides are correctly aligned.

4. Supply power according to the power specifications and do not apply an incorrect power voltage. Use only the provided AC power cable. Use only the specified type of fuse.

RENESAS

Warnings on Emulator Usage

Warnings described below apply as long as you use this emulator. Be sure to read and understand the warnings below before using this emulator. Note that these are the main warnings, not the complete list.

⚠ WARNING

Always switch OFF the emulator and user system before connecting or disconnecting any CABLES or PARTS. Failure to do so will result in a FIRE HAZARD and will damage the user system and the emulator product or will result in PERSONAL INJURY. The USER PROGRAM will be LOST.

RENESAS

CAUTION

Place the emulator station and evaluation chip board so that the trace cables are not bent or twisted. A bent or twisted cable will impose stress on the user interface leading to connection or contact failure. Make sure that the emulator station is placed in a secure position so that it does not move during use nor impose stress on the user interface.

RENESAS

RENESAS

Preface

Thank you for purchasing the E6000H emulator.

CAUTION

Read this manual before using the emulator product. Incorrect operation or connection will damage the user system, the emulator product, and the user program.

The E6000H emulator is an efficient software and hardware development support tool for application systems using the microcomputer developed by Renesas Technology Corp.

The E6000H emulator can either be used without a user system, for developing and debugging software, or connected via a user system interface cable to a user system, for debugging user hardware.

The emulator provides the following features:

1. Realtime emulation of the MCU
2. Efficient debugging enabled by variable break functions and a mass-storage trace memory (128-kcycles)
3. Parallel access with a command execution during emulation, for example
   - Trace data display
   - Emulation memory display and modification
4. Performance analysis
   Measurement of subroutine execution time and count for evaluating the execution efficiency of user programs
5. Graphical User Interface by the High-performance Embedded Workshop that runs on Windows® operating systems

High-performance Embedded Workshop is a Graphical User Interface intended to ease the development and debugging of applications written in C/C++ programming language and assembly language. Its aim is to provide a powerful yet intuitive way of accessing, observing and modifying the debugging platform in which the application is running.

High-performance Embedded Workshop is a powerful development environment for embedded applications targeted at microcontrollers. The main features are:

- A configurable build engine that allows you to set-up compiler, assembler and linker options via an easy to use interface.
- An integrated text editor with user customizable syntax coloring to improve code readability.
- A configurable environment to run your own tools.
- An integrated debugger which allows you to build and debug in the same application.
- Version control support.

The High-performance Embedded Workshop has been designed with two key aims; firstly to provide you, the user, with a set of powerful development tools and, secondly, to unify and present them in a way that is easy to use.

RENESAS

About This Manual

This manual is comprised of two parts: Hardware Part and Debugger Part.

Hardware Part: Preparation before use, hardware specifications, and troubleshooting procedure.

Debugger Part: A peculiar debugging function to the emulator, tutorial, Emulator software specification, and notes.

This manual describes the debugging function for SH7046 E6000H Emulator debugger that used with the High-performance Embedded Workshop.

For detailed information on the basic "look and feel" of the High-performance Embedded Workshop and customizing the High-performance Embedded Workshop environment and the build and the debugging functions common to the High-performance Embedded Workshop products, refer to the High-performance Embedded Workshop user's manual.

This manual does not intend to explain how to write C/C++ or assembly language programs, how to use any particular operating system or how best to tailor code for the individual devices. These issues are left to the respective manuals.

Microsoft, MS-DOS, Windows, Windows NT are registered trademarks of Microsoft Corporation.

Visual SourceSafe is a trademark of Microsoft Corporation.

IBM is a registered trademark of International Business Machines Corporation.

All brand or product names used in this manual are trademarks or registered trademarks of their respective companies or organizations.

Document Conventions

This manual uses the following typographic conventions:

Table 1   Typographic Conventions

Convention	Meaning
[Menu->Menu Option]	Bold text with '->' is used to indicate menu options (for example, **[File->Save As...]**).
FILENAME.C	Uppercase names are used to indicate filenames.
"enter this string"	Used to indicate text that must be entered (excluding the "" quotes).
Key + Key	Used to indicate required key presses. For example, **CTRL+N** means press the **CTRL** key and then, whilst holding the **CTRL** key down, press the **N** key.
↻ (The "how to" symbol)	When this symbol is used, it is always located in the left-hand margin. It indicates that the text to its immediate right is describing "how to" do something.

Components

Check all the components described in the component list unpacking. If the components are not complete, contact a Renesas sales office.

RENESAS

Contents

RENESAS

Debugger Part

RENESAS

RENESAS

RENESAS

Hardware Part

Section 1   Overview

1.1   Notes on Usage

```
CAUTION

READ the following warnings before using the emulator product.
Incorrect operation will damage the user system and the emulator
product. The USER PROGRAM will be LOST.
```

1. Check all components with the component list after unpacking the emulator.
2. Never place heavy objects on the casing.
3. Observe the following conditions in the area where the emulator is to be used:
    - Make sure that the internal cooling fans on the sides of the emulator must be at least 20 cm (8") away from walls or other equipment.
    - Keep out of direct sunlight or heat. Refer to section 3.1, Environmental Conditions.
    - Use in an environment with constant temperature and humidity.
    - Protect the emulator from dust.
    - Avoid subjecting the emulator to excessive vibration. Refer to section 3.1, Environmental Conditions.
4. Protect the emulator from excessive impacts and stresses.
5. Before using the emulator's power supply, check its specifications such as power voltage and frequency.
6. When moving the emulator, take care not to vibrate or otherwise damage it.
7. After connecting the cable, check that it is connected correctly. For details, refer to section 2, Preparation before Use.
8. Supply power to the emulator and connected parts after connecting all cables. Cables must not be connected or removed while the power is on.
9. For details on differences between the target MCU and the emulator, refer to section 3.5, Support of the Target MCU.

1.2 Emulator Hardware Components

The emulator consists of an E6000H station and an evaluation chip board. By installing a user system interface board (option) on your host computer, the emulator can be connected in the same package as the device. PC interface (option) includes a PC interface board (PCI bus and PC card bus), a LAN adapter (connected with the network), and a USB adapter (connected with the USB interface). By connecting the emulator to the host computer via those interfaces, the High-performance Embedded Workshop can be used for debugging. For details on PC interface boards (available for PCI bus and PC card bus specifications), LAN adapter, and USB adapter, refer to their description notes.
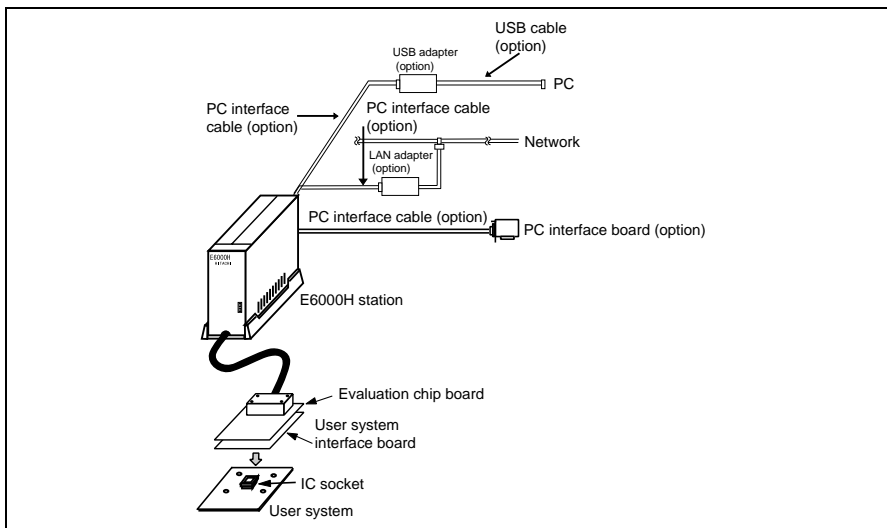
Figure 1.1 Emulator Hardware Components

RENESAS

1.2.1 E6000H Station Components (A Part of Photos may be Different from Real Appearances)

The names of the components on the front/rear panel of the E6000H station are listed below.

Front Panel:
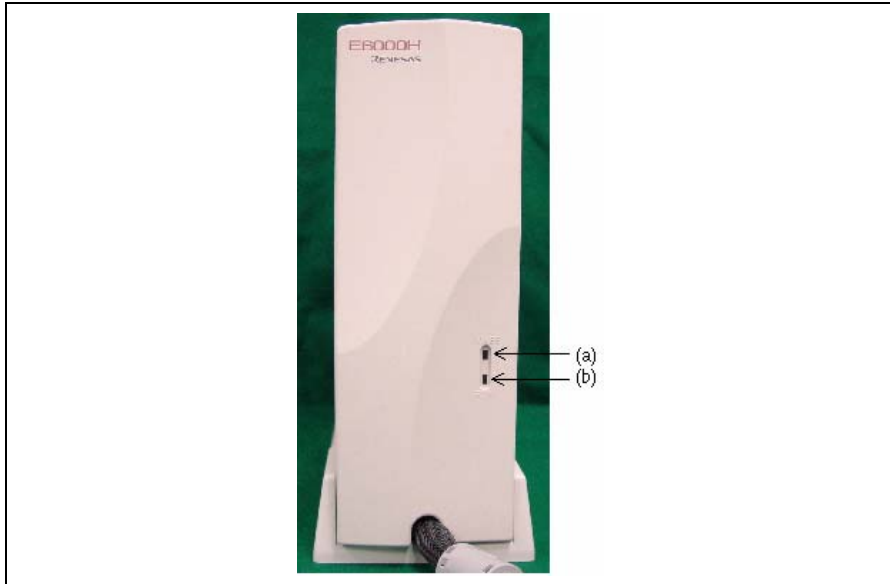
Figure 1.2   E6000H Station: Front Panel

(a) POWER lamp: Is lit up while the E6000H station is supplied with power.
(b) RUN lamp: Is lit up while the user program is running.

Rear Panel:

Figure 1.3   E6000H Station: Rear Panel

(a)	Power switch:	Turning this switch to I (input) supplies power to the emulator (E6000H station and evaluation chip board).
(b)	AC power connector:	For an AC 100-V to 240-V power supply.
(c)	PC interface cable connector:	For the PC interface cable that connects the host computer to the E6000H station. A PC interface board, PC card interface, LAN adapter, or USB adapter can be connected. Marked PC/IF.

RENESAS

1.2.2   Evaluation Chip Board Configuration

The names of the components on the evaluation chip board of the emulator are listed below.
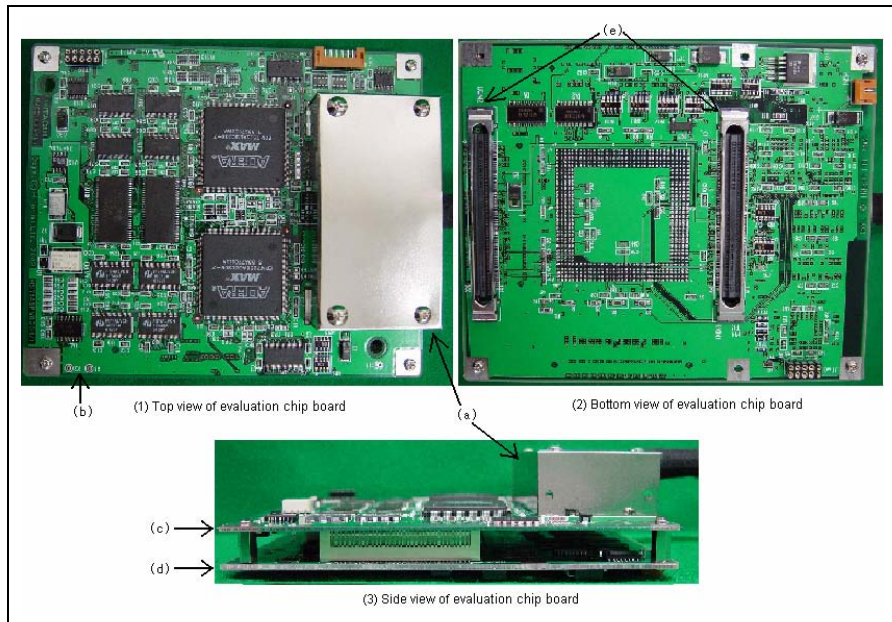
Figure 1.4   Evaluation Chip Board

(a) Station to evaluation chip board interface connector cover: This is a cover for protecting the connector that connects the E6000H station to the evaluation chip board.

(b) Crystal oscillator terminals: For installing a crystal oscillator to be used as an external clock source for the target MCU.

(c) HS7145PWB20H board: Connector to the trace cable is attached.

(d) HS7145PWB30H board: An evaluation chip is installed and a dedicated connector to the user system interface board or user system is attached.

(e) User system interface board connector: For connecting the user system interface board or user system.

Note: (a) to (e) listed above are referred to as the evaluation chip board.

RENESAS

1.2.3 Configuration of User System Interface Board

The names of the components of the user system interface board are given below.

(1) Top view of HS7145ECH61H board   (2) Bottom view of HS7145ECH61H board
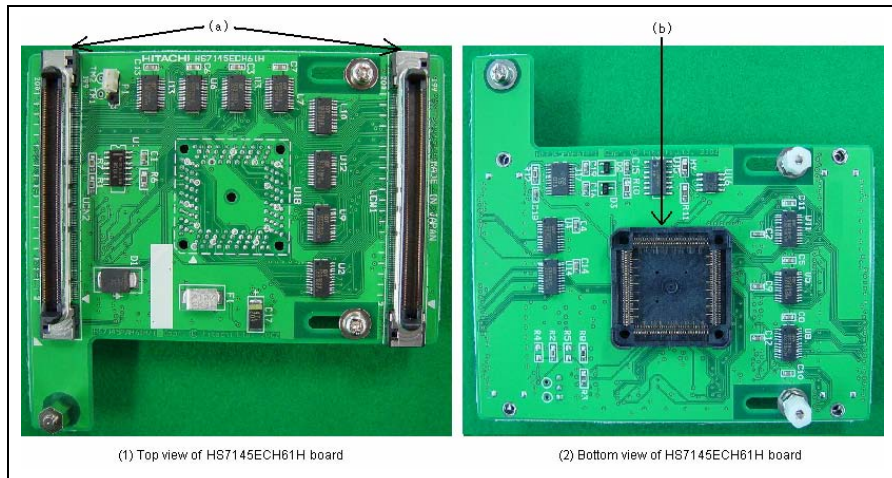
Figure 1.5   Configuration of User System Interface Board

(a) Connector for the evaluation chip   For connection to the evaluation chip board.
    board:

(b) Connector for the user system: For connection to the user system.

RENESAS

1.3 System Configuration

The emulator must be connected to a host computer (via the selected PC interface board).

1.3.1 System Configuration Using a PC Interface Board

The emulator can be connected to a host computer via a PC interface board (option: PCI bus or PC card bus). Install the PC interface board to the expansion slot for the interface board in the host computer, and connect the interface cable supplied with the PC interface board to the emulator. A LAN adapter can be used to connect the emulator to a host computer as a network. A USB adapter can be used to connect the emulator to a host computer with the USB interface. For details on PC interface boards (available for PCI bus and PC card bus specifications), LAN adapter, and USB adapter, refer to their description notes. Figure 1.6 shows the configuration of a system in which the PC interface board is used. Figure 1.7 shows the configuration of a system in which the LAN adapter is used. Figure 1.8 shows the configuration of a system in which the USB adapter is used.
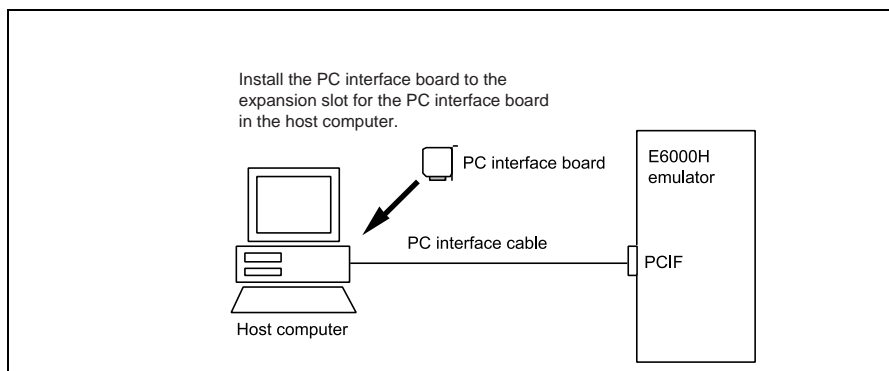
Figure 1.6   System Configuration Using a PC Interface Board
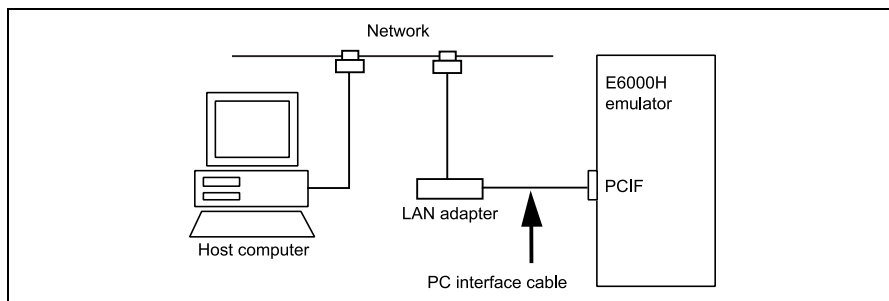
Figure 1.7   System Configuration Using a LAN Adapter

RENESAS
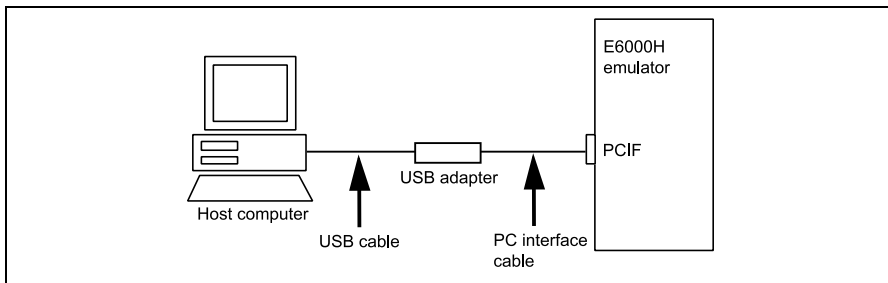
Figure 1.8   System Configuration Using a USB Adapter

RENESAS

Section 2   Preparation before Use

2.1   Description on Emulator Usage

This section describes the preparation before use of the emulator. Figure 2.1 is a flowchart on preparation before debugging with the emulator.

CAUTION

Read this section and understand its contents before preparation.
Incorrect operation will damage the user system and the emulator.
The USER PROGRAM will be LOST.

Reference:

Connect the emulator.	Described in this section.
Set up the emulator on each OS.	See the E6000H emulator setup guide.
Start debugging.	See the Debugger Part.
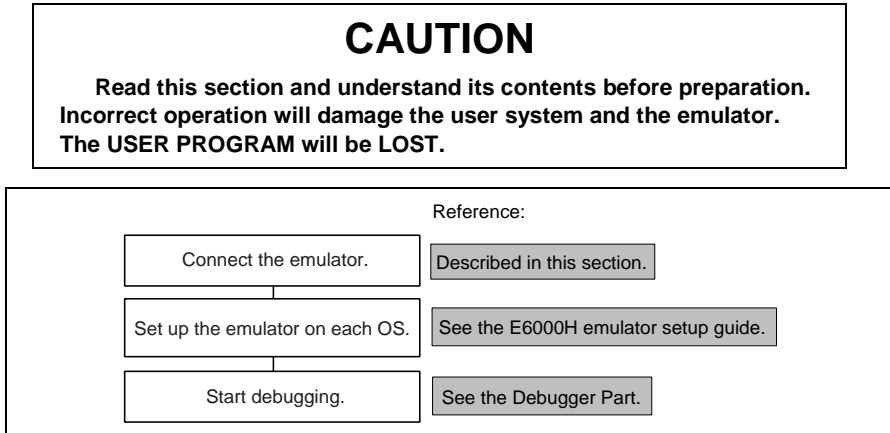
Figure 2.1   Emulator Preparation Flowchart

2.2 Emulator Connection

2.2.1 Connecting the User System

⚠️ **WARNING**

Always switch OFF the emulator and user system before connecting or disconnecting any CABLES. Failure to do so will result in a FIRE HAZARD and will damage the user system and the emulator or will result in PERSONAL INJURY. The USER PROGRAM will be LOST.

---

1. Check that the emulator power switch is turned off. Ensure that the power lamp on the right side of the E6000H station's front panel is not lit.

2. Remove the AC power cable of the E6000H station from the outlet (if the cable is connected to the outlet).

3. Connect pin 1 on the user system connector to the connector installed at the bottom of the E6000H user system interface board. When connecting the connector, prevent the upper or lower side of the board from lifting off the connector. Alternately tighten the screws on both sides of the board.

RENESAS

2.2.2 Connecting the User System Interface Board

⚠ WARNING

   Always switch OFF the emulator and user system and check pin numbers on the connectors and IC socket before connecting or disconnecting the USER SYSTEM INTERFACE BOARD. Connection with the power on or incorrect connection will damage the emulator, user system interface board, and user system, and result in a FIRE HAZARD.

For details on the method of connecting the user system interface board, refer to the descriptions of the user system interface boards for individual SH7046 E6000H-series products.
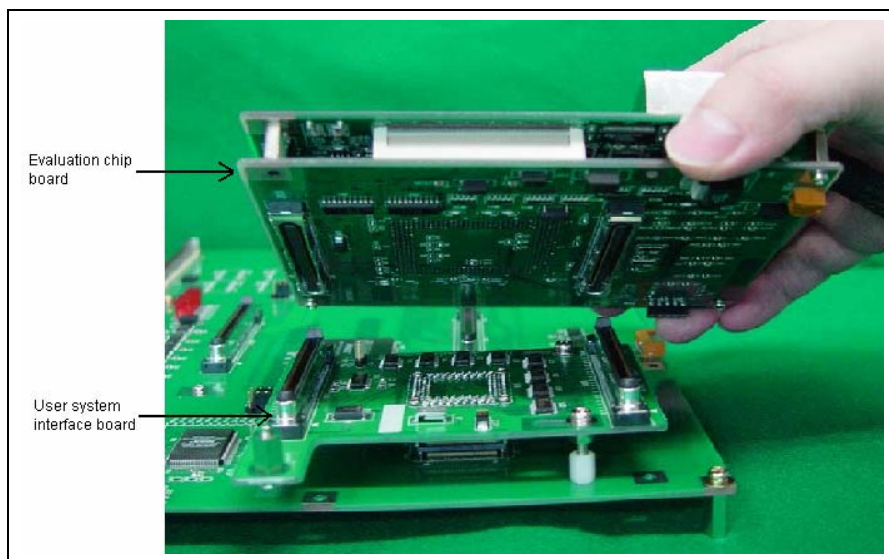
Figure 2.2   Connecting the User System Interface Board

CAUTION

Check the external probe direction and connect the external probe to the emulator station correctly. Incorrect connection will damage the probe or connector.

When an external probe is connected to the emulator probe connector on the E6000H evaluation chip board's rear panel, it enables external signal tracing and multibreak detection. Figure 2.3 shows the external probe connector.

Pin No.	Probe Name	Signal Name	Note
1	1	Probe input 1	Synchronous break
2	2	Probe input 2	input pin
3	3	Probe input 3	
4	4	Probe input 4	
5	G	GND	GND connection pin
6	TUV	Trigger output	Trigger mode output pin
7	G	GND	GND connection pin
8	T5V	RUN or break output	High-level output at RUNSTEP
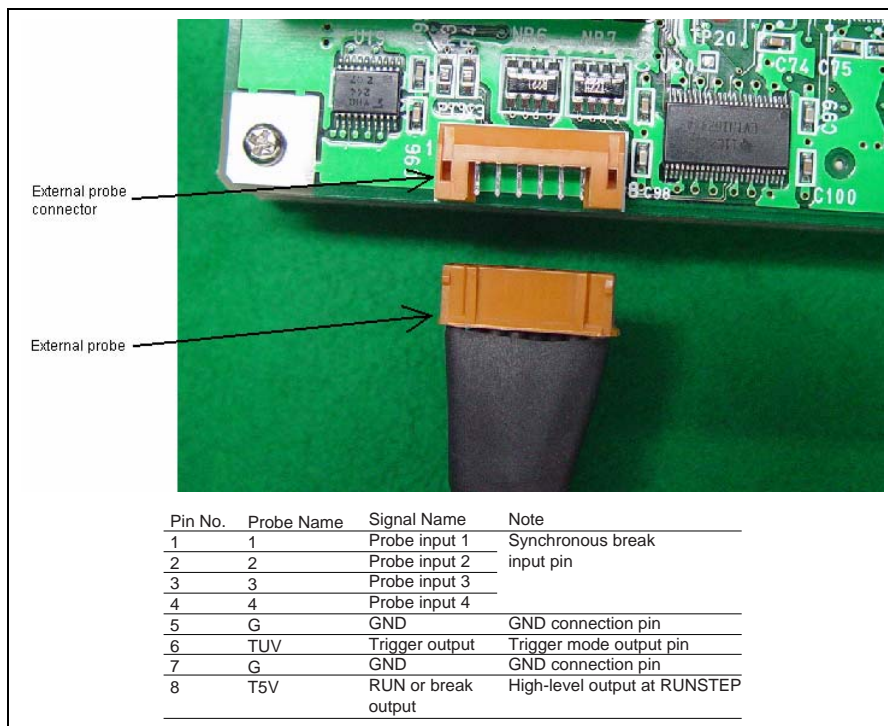
Figure 2.3   External Probe Connector

RENESAS

2.2.4 Selecting the Clock

This emulator supports three types of clock for the target MCU: a crystal oscillator attached on the evaluation chip board, external clock input from the user system, and the emulator internal clock. The clock is specified with the [Configuration] dialog box.

This emulator can use a clock source (ϕ) running at up to 50.0 MHz (four times the external clock frequency of 12.5 MHz) as the MCU clock input. Select one of the followings:

Target	External clock (clock signal supplied from the user system to the EXTAL pin): 4.0 to 12.5 MHz
Xtal	Crystal oscillator: 5.0 to 12.5 MHz
12.5 MHz	Emulator internal clock
10 MHz	Emulator internal clock
8 MHz	Emulator internal clock
6 MHz	Emulator internal clock
4 MHz	Emulator internal clock

Crystal Oscillator: A crystal oscillator is not supplied with the emulator. Prepare and use one that has the same frequency as that of the user system. When using a crystal oscillator as the target MCU clock source, the frequency range must be from 5.0 to 12.5 MHz.

CAUTION

Always switch OFF the emulator and user system before connecting or disconnecting the CRYSTAL OSCILLATOR. Otherwise, the USER PROGRAM will be LOST.

Follow the procedure listed below to install the crystal oscillator:

1. Check that the emulator power switch is turned off. (Check that the power LED is not lit.)
2. Attach the crystal oscillator into the terminals on the evaluation chip board (figure 2.4).
3. Turn on the user system power and then the emulator power. Then the crystal oscillator will be automatically set. This function will allow the execution of the user program at the operating frequency of the user system even when the user system is not connected to the emulator.

RENESAS
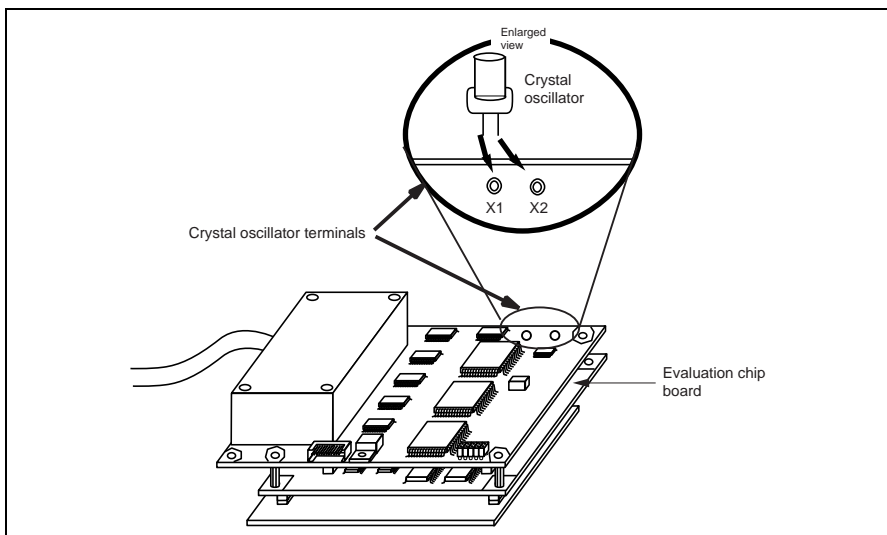
Figure 2.4   Installing the Crystal Oscillator

External Clock: Follow the procedure listed below to select the external clock.

1. Check that the emulator power switch is turned off. (Check that the power LED is not lit.)
2. Connect the evaluation chip board to the user system and supply a clock through the EXTAL pin from the user system.
3. Turn on the user system power and then the emulator power. The external clock will then be automatically set.

Emulator Internal Clock: Specify an emulator internal clock in the [Configuration] dialog box.

Reference:

When the emulator system program is initiated, the emulator automatically selects the MCU clock source according to the following priority:

1. User system's clock when an external clock is supplied from the user system
2. Crystal oscillator when attached to the evaluation chip board
3. Emulator internal clock

RENESAS

2.2.5 Connecting the System Ground

CAUTION

Separate the frame ground from the signal ground at the user system. When the frame ground is connected to the signal ground and the emulator is then connected to the user system, the emulator will malfunction.

The emulator's signal ground is connected to the user system's signal ground via the evaluation chip board. In the E6000H station, the signal ground and frame ground are connected (figure 2.5). At the user system, connect the frame ground only; do not connect the signal ground to the frame ground.

If it is difficult to separate the frame ground from the signal ground in the user system, ground the frame to the same outlet as the 100-V to 240-V power supply of the emulator station (figure 2.6) so that the ground potentials become even.
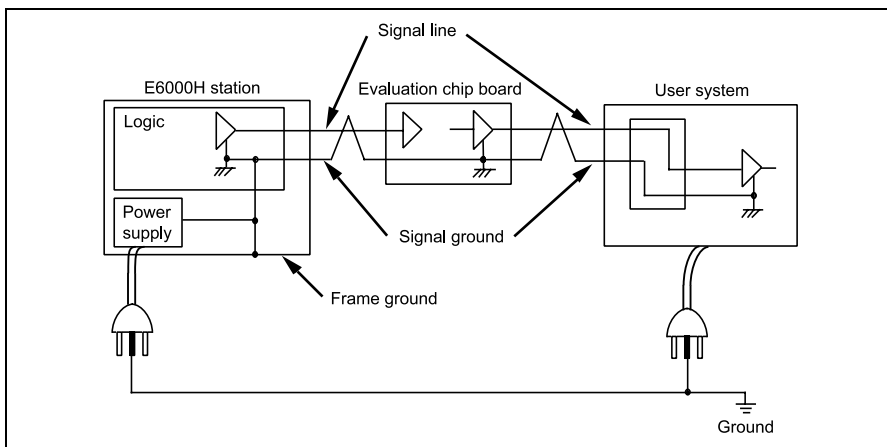
Figure 2.5   Connecting the System Ground

⚠ WARNING

Always switch OFF the emulator and user system before connecting or disconnecting any CABLES. Failure to do so will result in a FIRE HAZARD and will damage the user system and the emulator or will result in PERSONAL INJURY. The USER PROGRAM will be LOST.

The user system must be connected to an appropriate ground so as to minimize noise and the adverse effects of ground loops. When connecting the evaluation chip board and the user system, confirm that the ground pins of the evaluation chip board are firmly connected to the user system's ground.

15

RENESAS
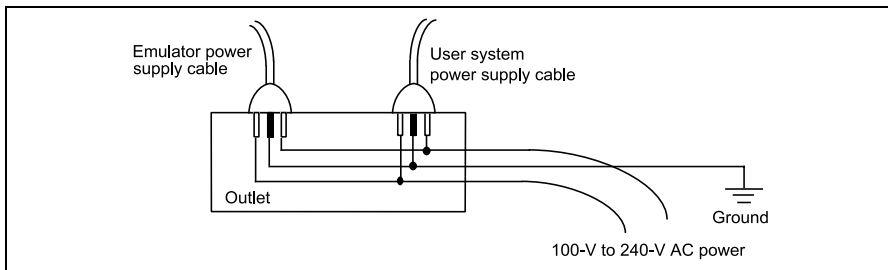
Figure 2.6 Connecting the Frame Ground

2.2.6 PC Interface Board Specifications

For details on the PC interface board, LAN adapter, or USB adapter, refer to their description notes.

Section 3   Hardware Specifications

3.1   Environmental Conditions

<div style="border:1px solid black; padding:1em;">

CAUTION

Observe the conditions listed in table 3.1 when using the emulator. The following environmental conditions must be satisfied, otherwise the user system and the emulator will not operate normally. The USER PROGRAM will be LOST.

</div>

Table 3.1   Environmental Conditions

Item	Specifications	
Temperature	Operating:	+10 to +35°C
	Storage:	−10 to +50°C
Humidity	Operating:	35 to 80% RH, no condensation
	Storage:	35 to 80% RH, no condensation
Vibration	Operating:	2.45 m/s^2 max.
	Storage:	4.9 m/s^2 max.
	Transportation:	14.7 m/s^2 max.
AC input power	Voltage:	100 V to 240 V AC
	Frequency:	50/60 Hz
	Power consumption: 75 W	
Ambient gases	There must be no corrosive gases present.	

3.2   Emulator External Dimensions and Mass

Figures 3.1 shows the external dimensions and mass of the E6000H emulator.
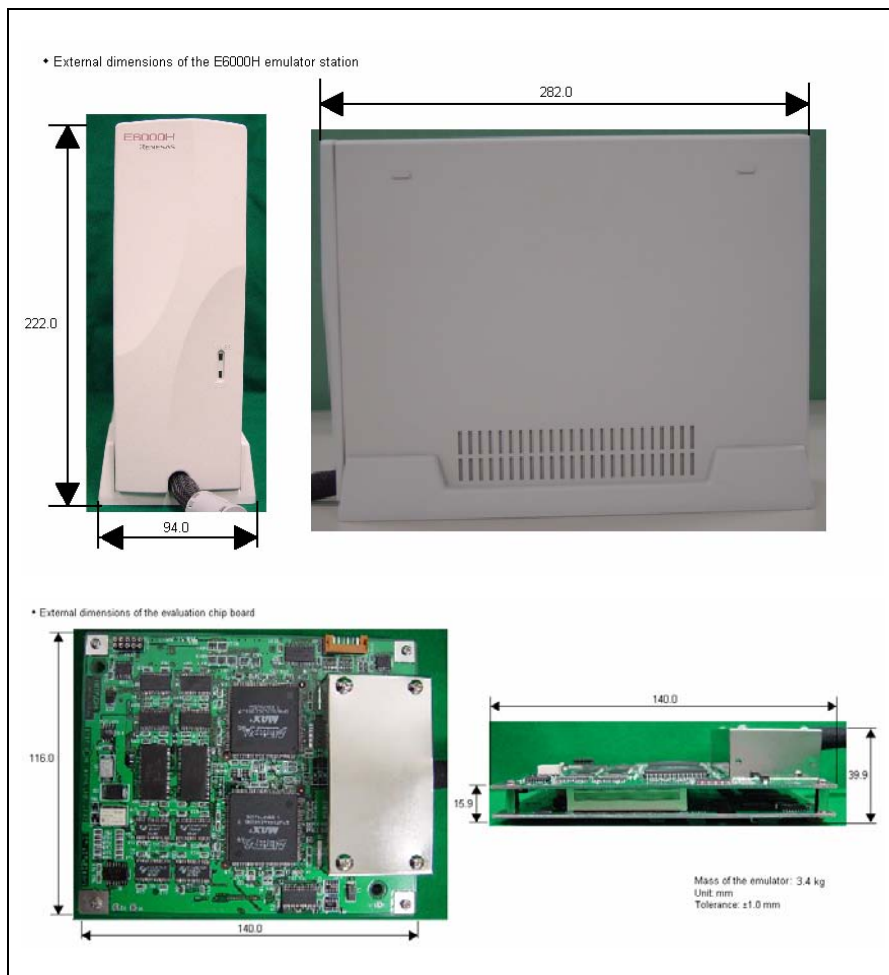
Figure 3.1   External Dimensions and Mass of the Emulator

RENESAS

3.3　User System Interface Circuit

3.3.1　User System Interface Circuit

The circuits that interface the MCU in the emulator to the user system include buffers and resistors. When connecting the emulator to a user system, adjust the user system hardware compensating for FANIN, FANOUT, and propagation delays.

The AC timing values when using the emulator are shown in table 3.2.

Note: The values with the emulator connected, in table 3.2, are measurements for reference and are not guaranteed values.

Table 3.2　Bus Timing when Using the Emulator (Bus Clock: 50.0 MHz)

Item	MCU Specifications (ns)		Values with Emulator Connected (ns)	
	Min	Max	Min	Max
tRDS	15	—	16	—
tACC	t_{cyc} x (n + 2) – 35 (n is the number of waits)	—	t_{cyc} x (n + 2) – 37 (n is the number of waits)	—

The basic bus cycle (software wait) is shown in figure 3.2. The user system interface circuits connected to the user system are shown in figures 3.3 to 3.8.

RENESAS

Note: tRDH: Specified from the earliest negation timing from A17 to A0, _CSn, or _RD.
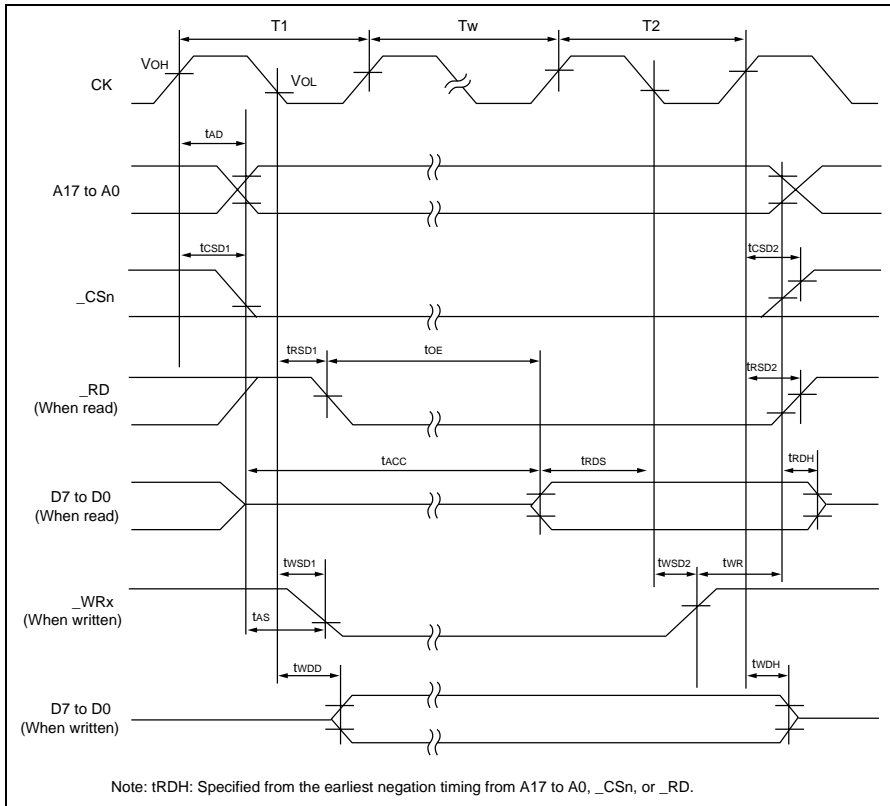
Figure 3.2   Basic Bus Cycle (Software Wait)
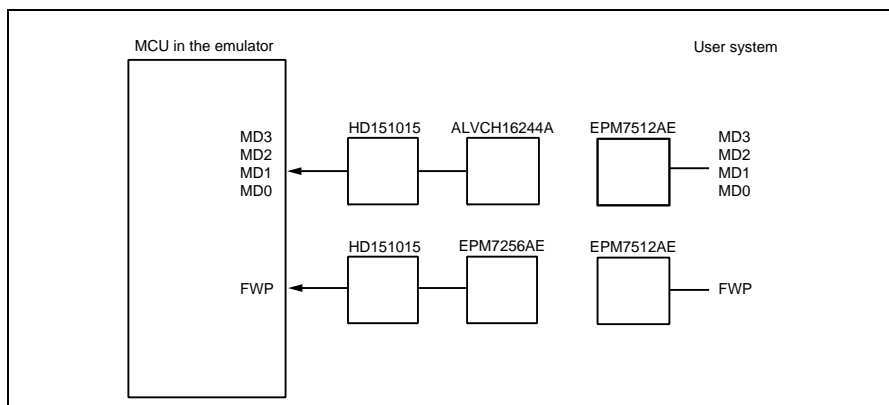
Figure 3.3   User System Interface Circuits (1)

RENESAS
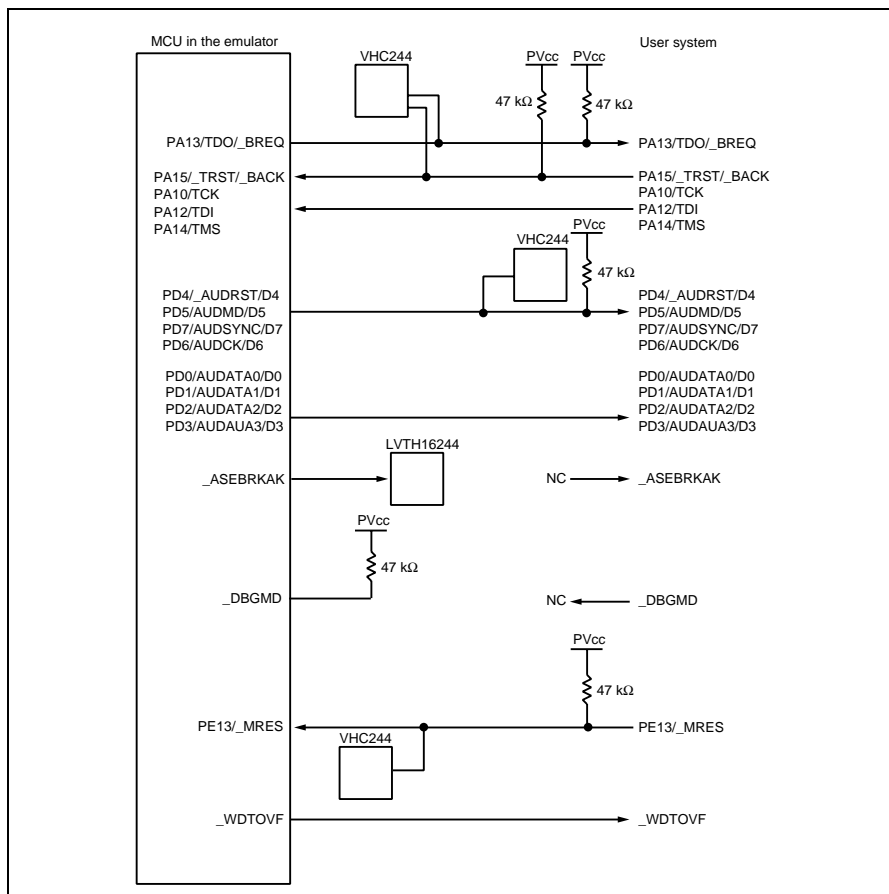
Figure 3.4   User System Interface Circuits (2)
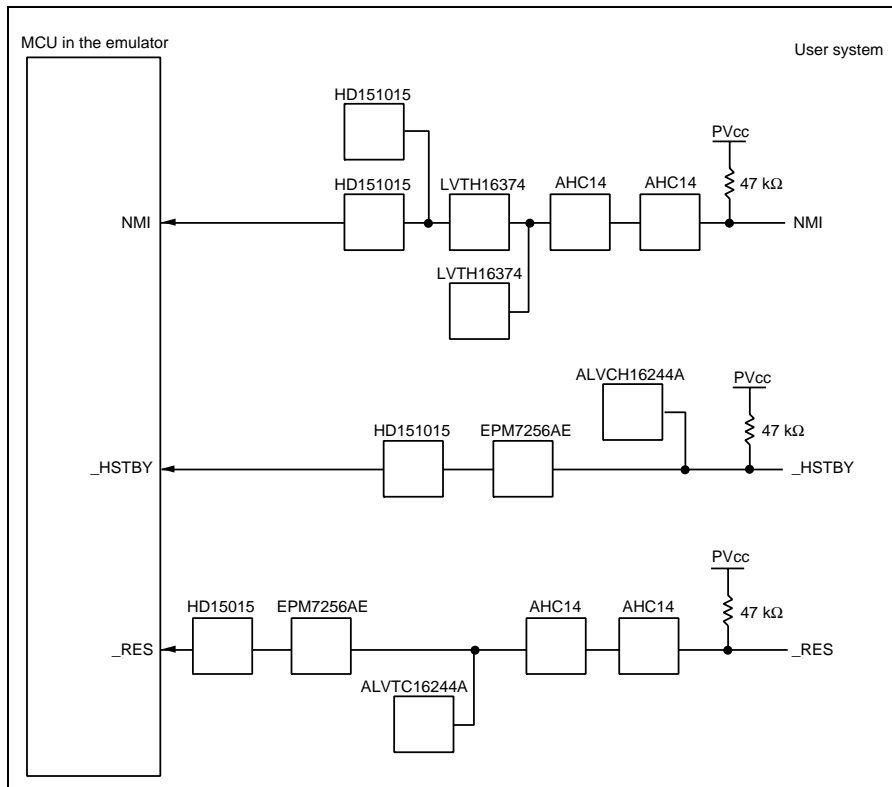
Figure 3.5 User System Interface Circuits (3)
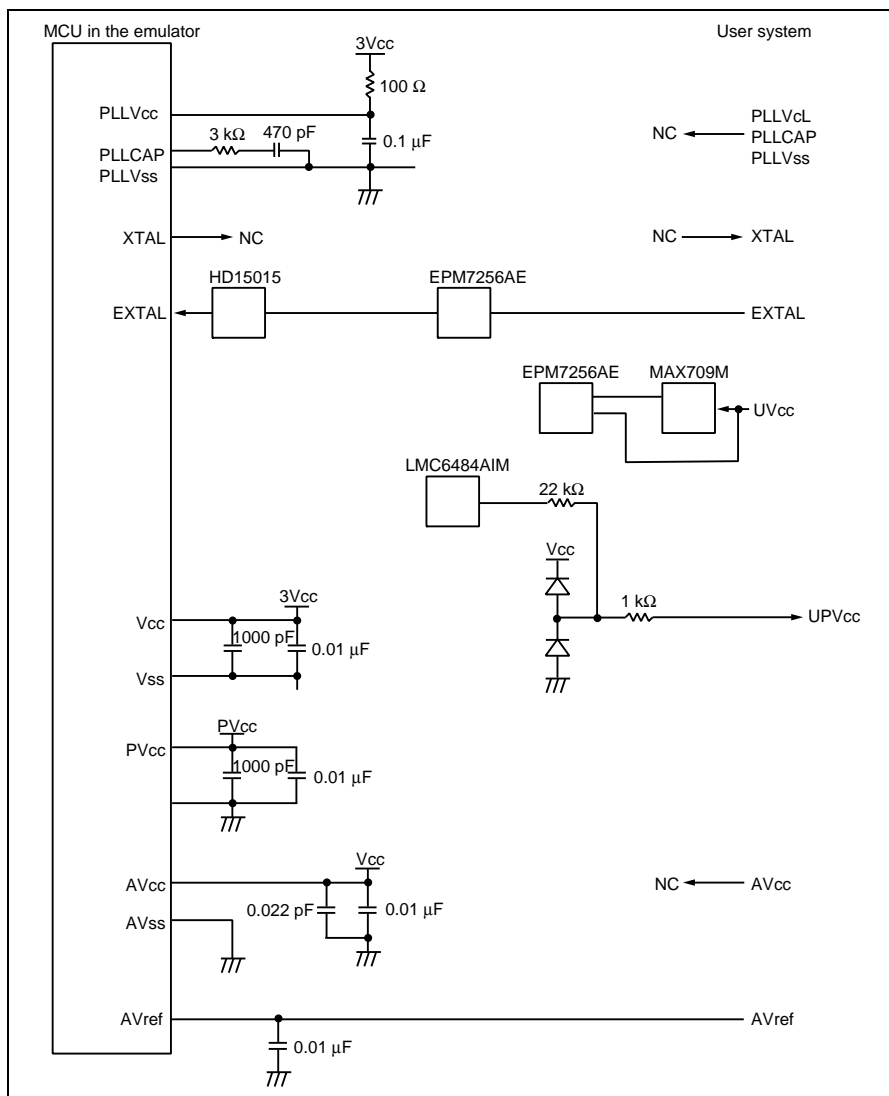
Figure 3.6   User System Interface Circuits (4)

RENESAS
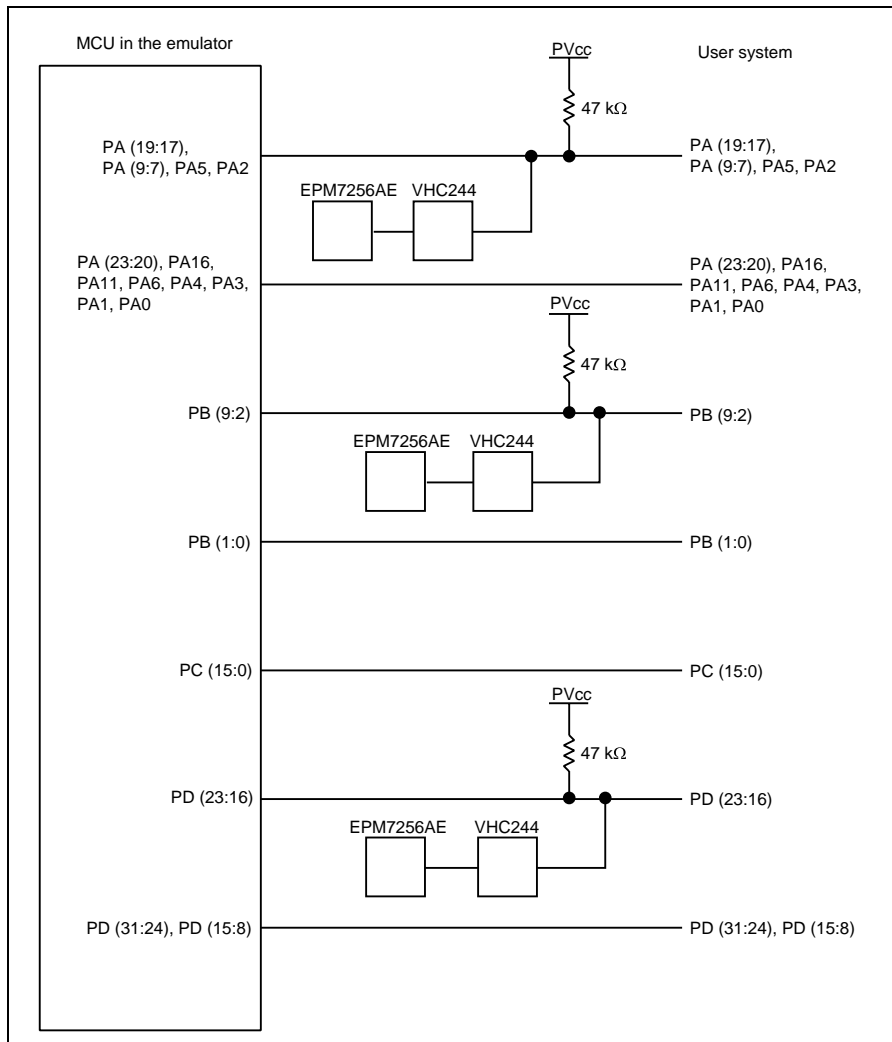
Figure 3.7   User System Interface Circuits (5)
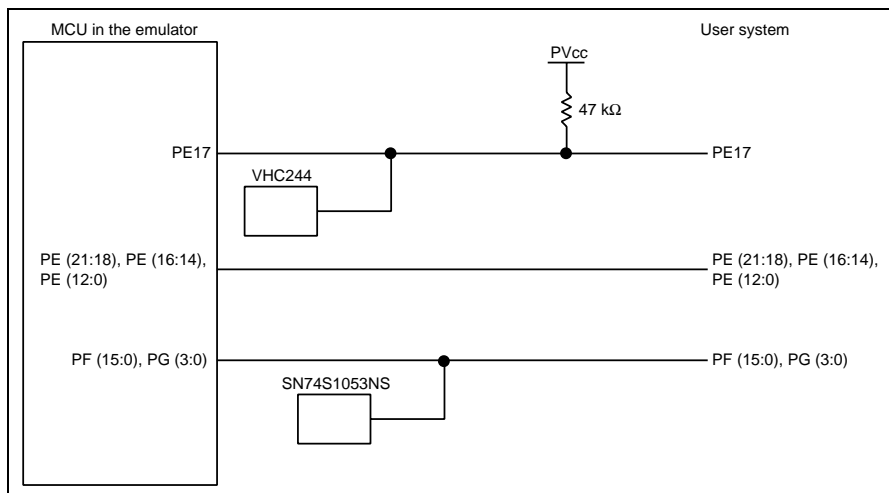
Figure 3.8   User System Interface Circuits (6)

3.3.2   Delay Time with the User System Interface

The delay time is generated on the timing of the _RES signal when it is input to the MCU from the user system, as shown in table 3.3, because this connection for this signal is via logic circuit on the evaluation chip board.

Table 3.3   Delay Time for Signal Connected via the Evaluation Chip Board

Signal Name	Delay Time (ns)
_RES	15.0

RENESAS

3.4 Connecting the Emulator to the User System

3.4.1 Connecting to the User System

<div style="border:2px solid black; text-align:center;">

⚠ WARNING

Always switch OFF the emulator and user system before connecting or disconnecting any CABLES.
Failure to do so will result in a FIRE HAZARD, and will damage the user system or emulator or result in PERSONAL INJURY. Also, the USER PROGRAM will be LOST.

</div>

The emulator is connected to the user system by using the user system interface board.

Table 3.4 User System Interface Board and User Interfaces

User System Interface Board	User Interface
HS7046ECH61H	FP-80Q (IC149_080_017_B5)[1]
HS7047ECH61H	FP-100M (IC149_100_054_B51)[1]
HS7144ECH61H	FP-112B (IC149_112C15333-0B)[1]
HS7145ECH61H	FP-144F (NQPACK144SD)[2]

Notes: 1. The IC149 series is manufactured by YAMAICHI ELECTRONICS Co., LTD.
2. The NQPACK series is manufactured by Tokyo Eletech Corporation.

RENESAS

Installing IC Socket

1. Installing IC Socket

   Install the IC socket for each package to the user system. After checking the location of pin 1 on the IC socket, apply epoxy resin adhesive to the bottom of the IC, and fasten it to the user system before soldering.

2. Soldering IC Socket

   After fastening, solder the IC socket to the user system. Be sure to completely solder the leads so that the solder slops gently over the leads and forms solder fillets. (Use slightly more solder than the MCU.)

Connection Using the HS7046ECH61H

⚠ WARNING

   Always switch OFF the emulator and user system before connecting or disconnecting any CABLES.
Failure to do so will result in a FIRE HAZARD, and will damage the user system or emulator or result in PERSONAL INJURY.
Also, the USER PROGRAM will be LOST.

Notes: 1. For more details on the HS7046ECH61H, refer to the user's manual supplied with the emulator.

2. This user system interface board can only be used in combination with the specified QFP socket (IC149-080-017-B5).

Install the FP-80Q IC socket (IC149-080-017-B5 manufactured by YAMAICHI ELECTRONICS Co., LTD.) on the user system to connect the emulator. Since the pin arrangement is the same as that of the actual MCU, refer to the hardware manual.

Figure 3.9 shows the connection of the HS7046ECH61H, figure 3.10 shows the size restrictions for the installed components of the HS7046ECH61H, and figure 3.11 shows the recommended mount pad dimensions of the user system IC socket.

RENESAS
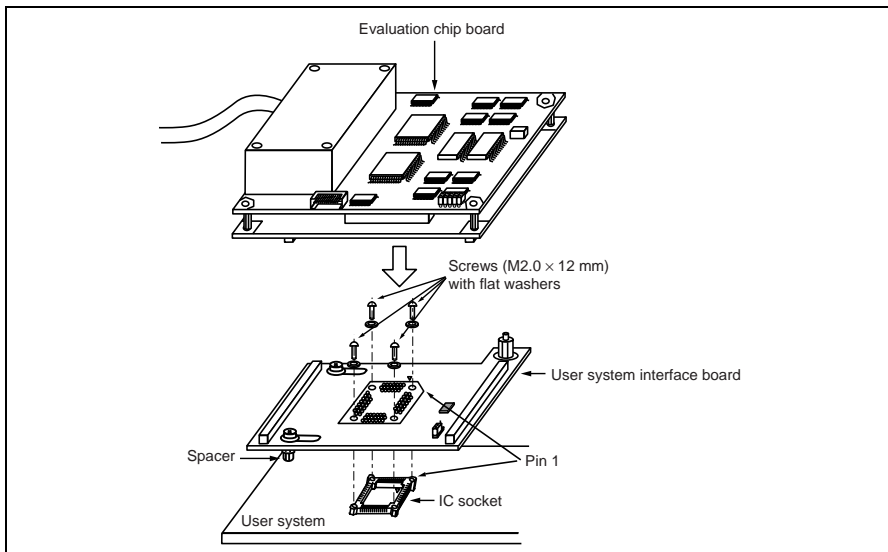
Figure 3.9   Connection Using the HS7046ECH61H
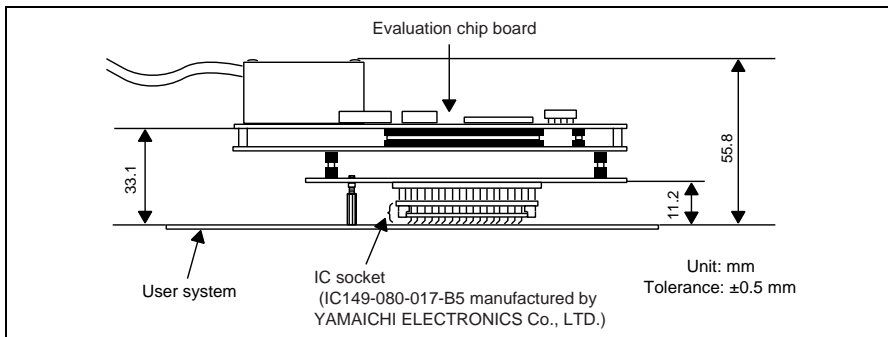
Figure 3.10   Restrictions on Component Installation

RENESAS
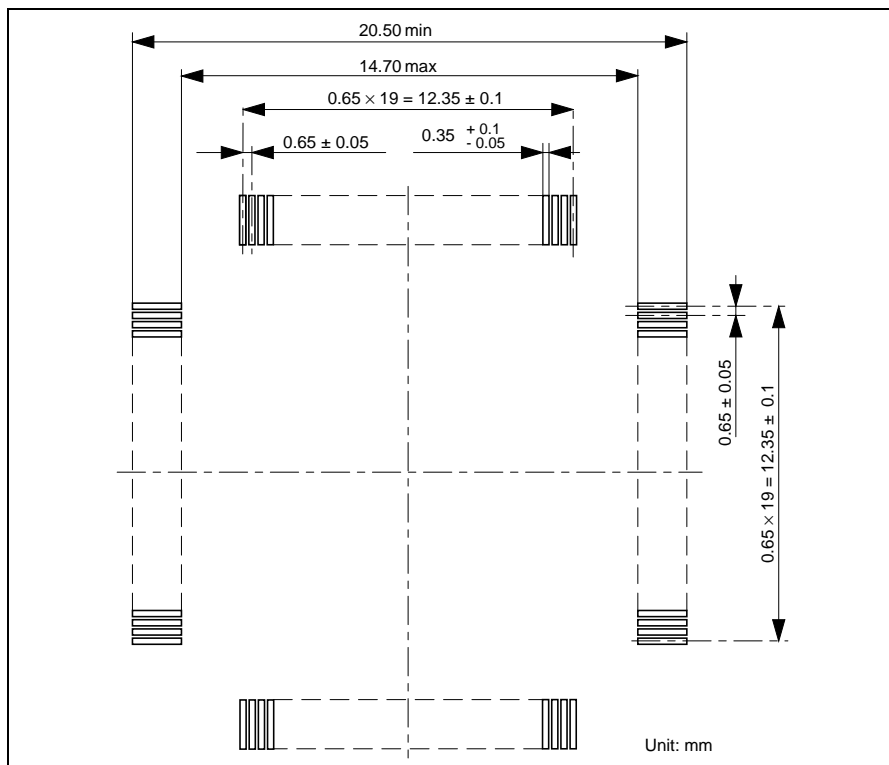
Figure 3.11   Recommended Mount Pad Dimensions of the User System IC Socket

Connection Using the HS7047ECH61H

⚠ WARNING

Always switch OFF the emulator and user system before
connecting or disconnecting any CABLES.
Failure to do so will result in a FIRE HAZARD, and will damage
the user system or emulator or result in PERSONAL INJURY.
Also, the USER PROGRAM will be LOST.

Notes: 1. For more details on the HS7047ECH61H, refer to the user's manual supplied with the emulator.
2. This user system interface board can only be used in combination with the specified QFP socket (IC149-100-054-B51).

Install the FP-100M IC socket (IC149-100-054-B51 manufactured by YAMAICHI ELECTRONICS Co., LTD.) on the user system to connect the emulator. Since the pin arrangement is the same as that of the actual MCU, refer to the hardware manual.

Figure 3.12 shows the connection of the HS7047ECH61H, figure 3.13 shows the size restrictions for the installed components of the HS7047ECH61H, and figure 3.14 shows the recommended mount pad dimensions of the user system IC socket.

RENESAS
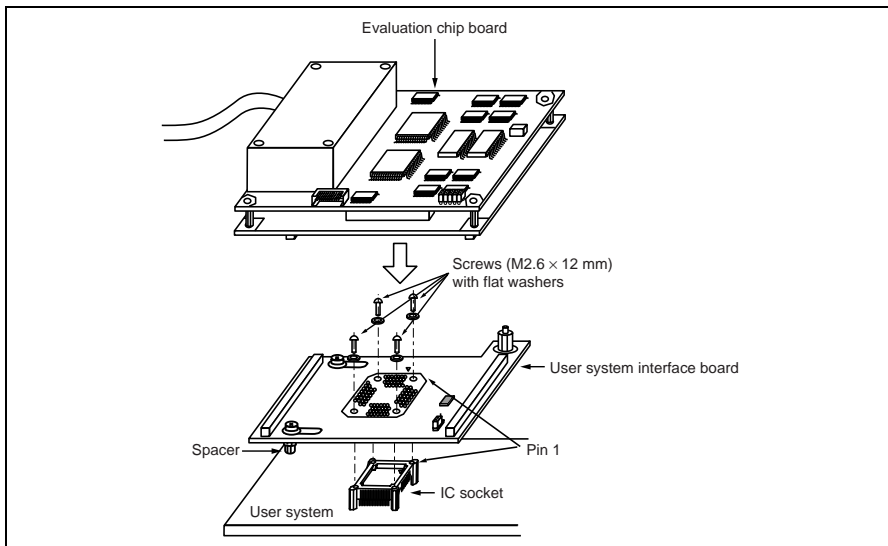
Figure 3.12   Connection Using the HS7047ECH61H
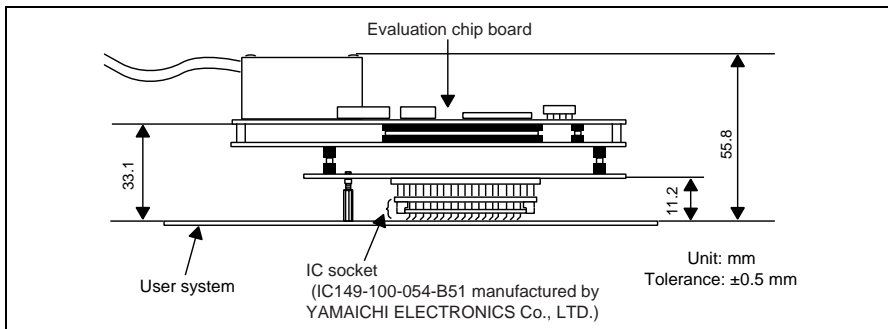
Figure 3.13   Restrictions on Component Installation

RENESAS

17.10 min

13.80 max

0.50 × 24 = 12.00 ± 0.05

0.50 ± 0.05

0.30 ± 0.05

0.50 ± 0.05

0.50 × 24 = 12.00 ± 0.05

Unit: mm
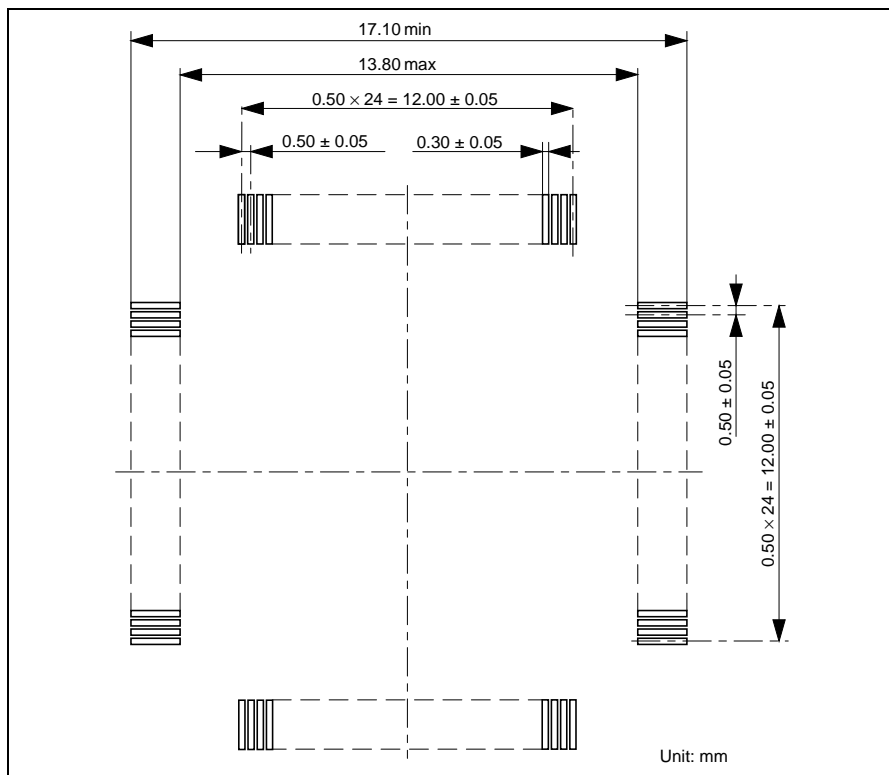
Figure 3.14   Recommended Mount Pad Dimensions of the User System IC Socket

Connection Using the HS7144ECH61H

⚠ WARNING

Always switch OFF the emulator and user system before connecting or disconnecting any CABLES.
Failure to do so will result in a FIRE HAZARD, and will damage the user system or emulator or result in PERSONAL INJURY.
Also, the USER PROGRAM will be LOST.

Notes: 1. For more details on the HS7144ECH61H, refer to the user's manual supplied with the emulator.
2. This user system interface board can only be used in combination with the specified QFP socket (IC149-112C15333-0B).

Install the FP-112B IC socket (IC149-112C15333-0B manufactured by YAMAICHI ELECTRONICS Co., LTD.) on the user system to connect the emulator. Since the pin arrangement is the same as that of the actual MCU, refer to the hardware manual.

Figure 3.15 shows the connection of the HS7144ECH61H, figure 3.16 shows the size restrictions for the installed components of the HS7144ECH61H, and figure 3.17 shows the recommended mount pad dimensions of the user system IC socket.

RENESAS
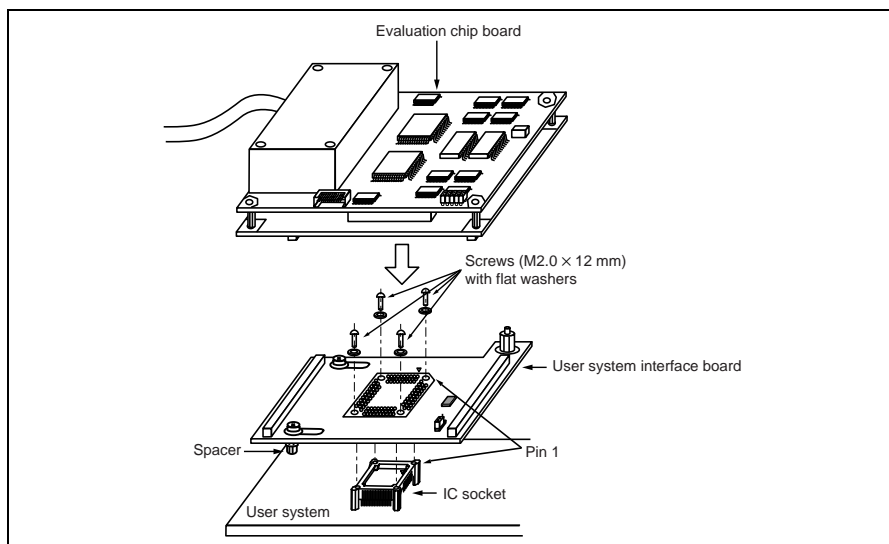
Figure 3.15   Connection Using the HS7144ECH61H
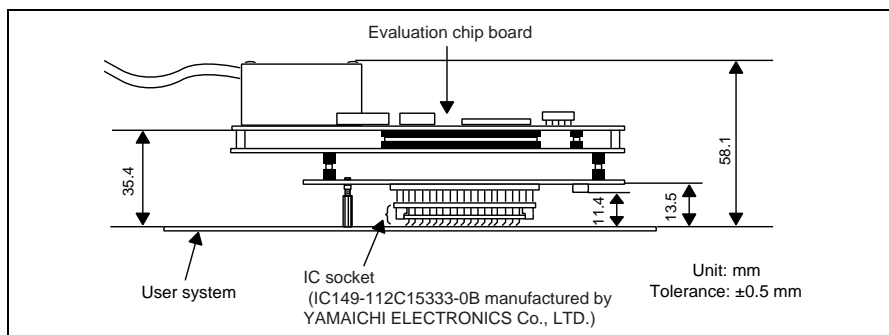
Figure 3.16   Restrictions on Component Installation

25.60 (min.)

20.80 (max.)

0.65 × 27 = 17.55 ± 0.10

0.65 ± 0.05

0.35 $^{+0.10}_{-0.05}$

0.65 ± 0.05

0.65 × 27 = 17.55 ± 0.10

Unit: mm
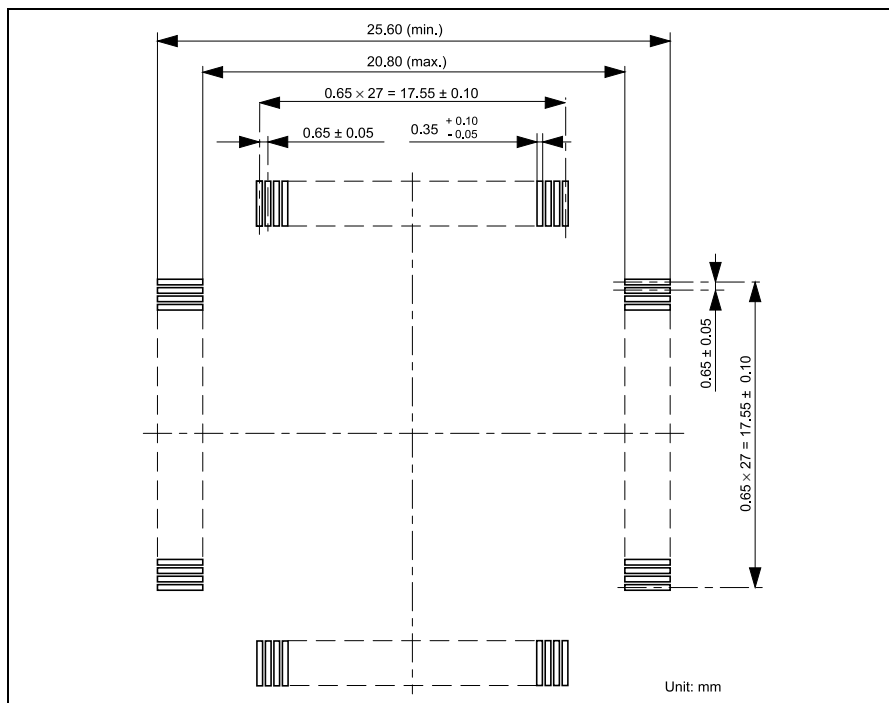
Figure 3.17   Recommended Mount Pad Dimensions of the User System IC Socket

RENESAS

Connection Using the HS7145ECH61H

<div style="border:1px solid black">

⚠ WARNING

   Always switch OFF the emulator and user system before connecting or disconnecting any CABLES.
Failure to do so will result in a FIRE HAZARD, and will damage the user system or emulator or result in PERSONAL INJURY.
Also, the USER PROGRAM will be LOST.

</div>

Notes: 1. For more details on the HS7145ECH61H, refer to the user's manual supplied with the emulator.
2. This user system interface board can only be used in combination with the specified QFP socket (NQPACK144SD).

Install the FP-144F IC socket (NQPACK144SD manufactured by Tokyo Eletech Corporation) on the user system to connect the emulator. Since the pin arrangement is the same as that of the actual MCU, refer to the hardware manual.

Figure 3.18 shows the connection of the HS7145ECH61H, figure 3.19 shows the size restrictions for the installed components of the HS7145ECH61H, and figure 3.20 shows the recommended mount pad dimensions of the user system IC socket.

RENESAS
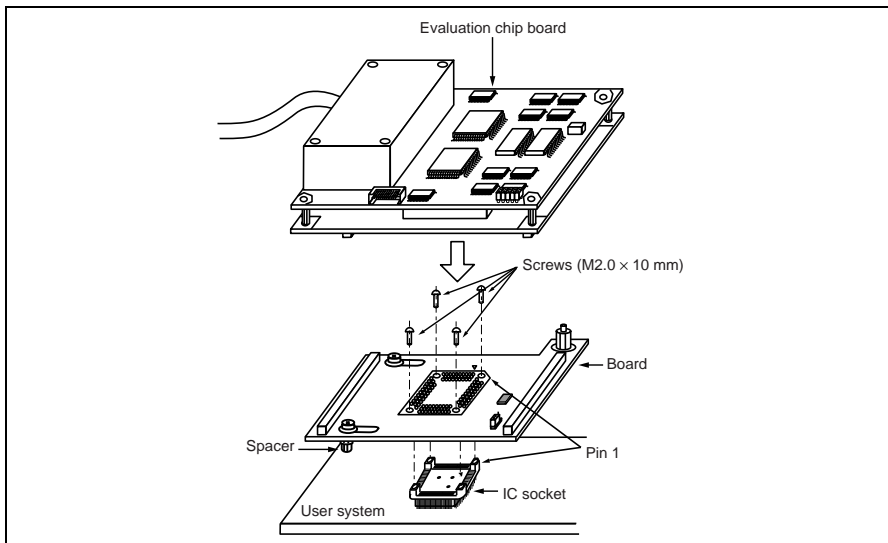
Figure 3.18   Connection Using the HS7145ECH61H
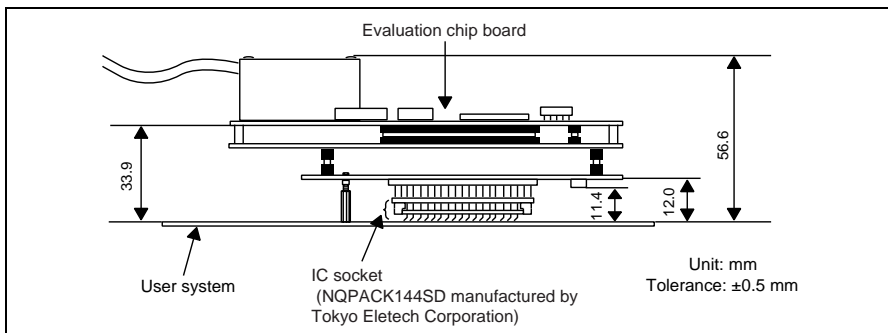
Figure 3.19   Restrictions on Component Installation

Figure 3.20   Recommended Mount Pad Dimensions of the User System IC Socket
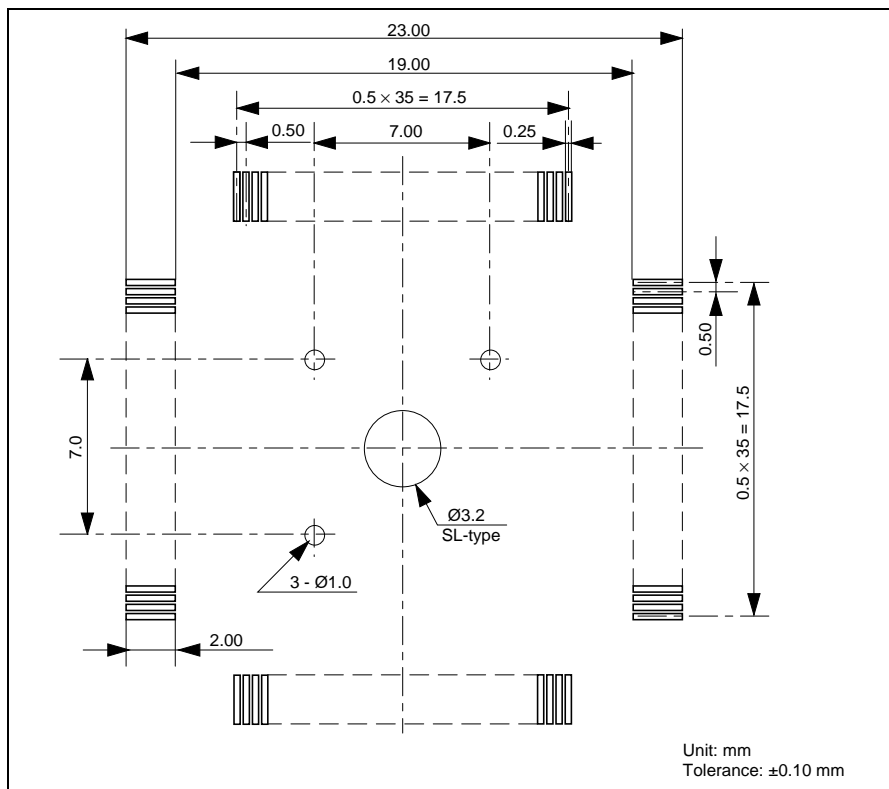
Connection Using the Dedicated Connector

⚠ WARNING

Always switch OFF the emulator and user system before connecting or disconnecting any CABLES.
Failure to do so will result in a FIRE HAZARD, and will damage the user system or emulator or result in PERSONAL INJURY.
Also, the USER PROGRAM will be LOST.

Note: This evaluation chip board can only be used in combination with the specified dedicated connector (WD-200P-VF85-N).

Install the dedicated connector (WD-200P-VF85-N manufactured by Japan Aviation Electronics Industry, Ltd.) on the user system to connect the emulator. Figures 3.21, 3.22, and 3.23 show connection using the dedicated connector, size restrictions for the installed components, and the location for mounting the connector in the user system, respectively.
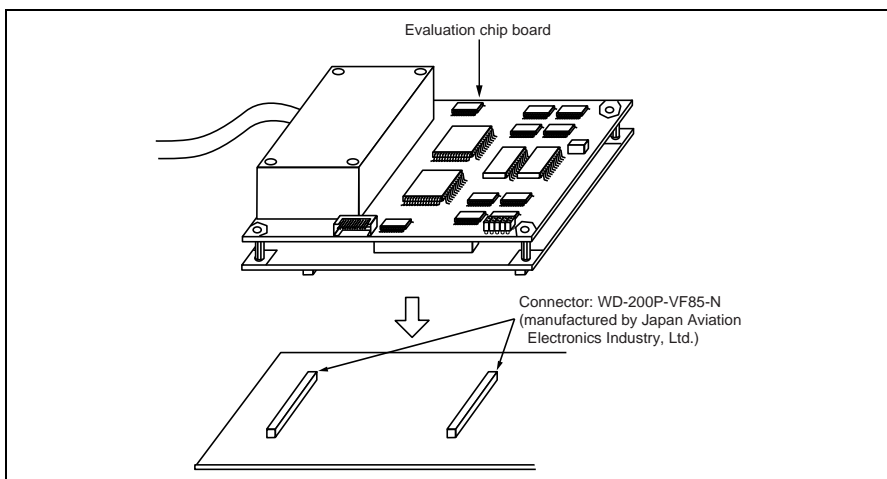
Figure 3.21   Connection Using the Dedicated Connector
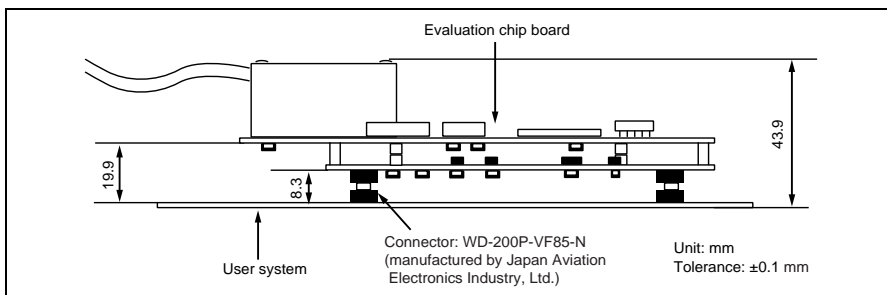
Figure 3.22   Size Restrictions for the Installed Components

RENESAS
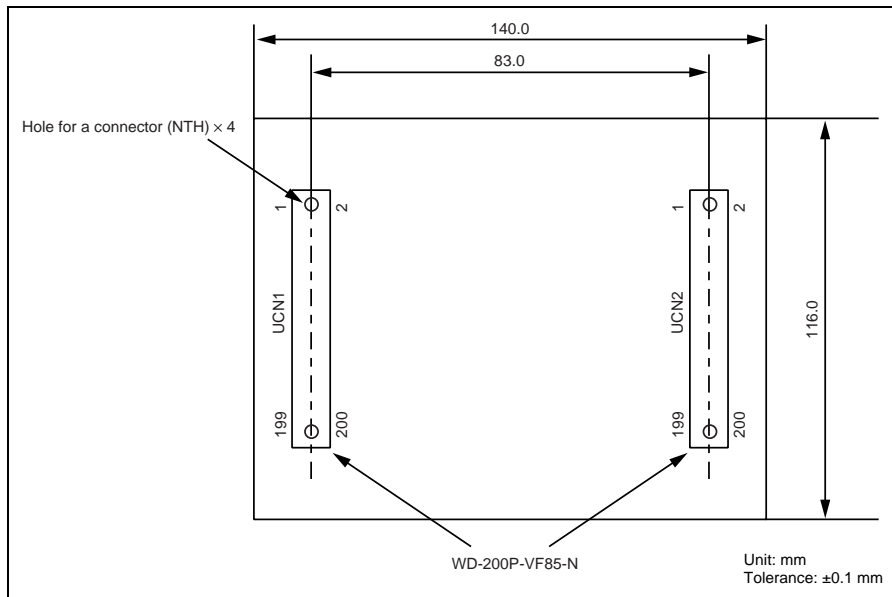
Figure 3.23 Location for Mounting the Connector in the User System

To design the foot pattern, refer to the catalog on WD-200P-VF85-N for dimensions.

3.4.2 Pin Arrangement on the User System Interface Connector

Table 3.5 lists the pin arrangement on the user system interface connector of HS7046EPH60H.

Table 3.5 Pin Arrangement on HS7046EPH60H

User I/F 1	Pin No.	Signal Name	User I/F 1	Pin No.	Signal Name
UCN1	1	GND	UCN1	25	N.C. (SMSB0_N)
	2	GND		26	GND
	3	GND		27	GND
	4	GND		28	N.C. (SDB7_P)
	5	GND		29	N.C. (SE_N)
	6	GND		30	N.C. (SDB6_P)
	7	GND (TGBON1)		31	N.C. (SMSPT_N)
	8	N.C. (SDB15_P)		32	N.C. (SDB5_P)
	9	N.C. (SMS3_N)		33	N.C. (SMSE_N)
	10	N.C. (SDB14_P)		34	N.C. (SDB4_P)
	11	N.C. (SMS2_N)		35	N.C. (SIORD_N)
	12	N.C. (SDB13_P)		36	GND
	13	N.C. (SMS1_N)		37	GND
	14	N.C. (SDB12_P)		38	N.C. (SDB3_P)
	15	N.C. (SMS0_N)		39	N.C. (SSSTBY_N)
	16	GND		40	N.C. (SDB2_P)
	17	GND		41	N.C. (SIOWORD_N)
	18	N.C. (SDB11_P)		42	N.C. (SDB1_P)
	19	N.C. (SMSB3_N)		43	N.C. (SWAIT_N)
	20	N.C. (SDB10_P)		44	N.C. (SDB0_P)
	21	N.C. (SMSB2_N)		45	N.C. (SIOWR_N)
	22	N.C. (SDB9_P)		46	GND
	23	N.C. (SMSB1_N)		47	GND
	24	N.C. (SDB8_P)		48	N.C. (SHCANSTART_P)

RENESAS

Table 3.5   Pin Arrangement on HS7046EPH60H (cont)

User I/F 1	Pin No.	Signal Name	User I/F 1	Pin No.	Signal Name
UCN1	49	N.C.	UCN1	75	N.C.
	50	N.C.		76	N.C.
	51	N.C.		77	GND
	52	N.C.		78	GND
	53	N.C.		79	PE21/PWOBB/SCK4/A15
	54	N.C.		80	N.C.
	55	N.C.		81	PE20/PVOB/TXD4/A14
	56	GND		82	PE13/TIOC4B/_MRES
	57	GND		83	PE19/PUOB/RXD4/A13
	58	N.C.		84	PE12/TIOC4A/TXD1/TXD3
	59	N.C.		85	PE18/PWOA/-/A12
	60	N.C.		86	PE11/TIOC3D/RXD1/RXD3
	61	N.C.		87	GND
	62	N.C.		88	GND
	63	N.C.		89	PE17/PVOA/_WAIT/A11
	64	N.C.		90	PE10/TIOC3C/TXD2/_WRL/ HRxD0&HRxD
	65	GND		91	PE16/PUOA/_UBCTRG/A10
	66	N.C.		92	PE9/TIOC3B/SCK1/SCk3/ HTxD0&HTxD1
	67	N.C.		93	PE15/TIOC4D/DACK1/ _IRQOUT
	68	GND		94	PE8/TIOC3A/SCK2/_WRH
	69	N.C.		95	PE14/TIOC4C/DACK0
	70	N.C.		96	PE7/TIOC2B/RXD2/A9
	71	N.C.		97	N.C.
	72	N.C.		98	GND
	73	N.C.		99	3VCC
	74	N.C.		100	PE6/TIOC2A/SCK3/A8

RENESAS

Table 3.5 Pin Arrangement on HS7046EPH60H (cont)

User I/F 1	Pin No.	Signal Name	User I/F 1	Pin No.	Signal Name
UCN1	101	3VCC	UCN1	127	GND
	102	PE5/TIOC1B/TXD3/A7		128	GND
	103	3VCC		129	GND
	104	PE4/TIOC1A/RXD3/A6		130	PD5/D5/-/AUDMD
	105	3VCC		131	PB9/_IRQ7/A21/_ADTRG/-/-/-/-
	106	PE3/TIOC0D/DRAK1/_CS3		132	PD6/D6/-/AUDCK
	107	3VCC		133	PB8/_IRQ6/A20/_WAIT/-/-/-/-
	108	GND		134	PD7/D7/-/_AUDSYNC
	109	3VCC		135	GND
	110	PE2/TIOC0C/_DREQ1/_CS2		136	PD8/D8/-/_UBCTRG
	111	5VCC		137	PB7/_IRQ5/A19/_BREQ/-/-/-/-
	112	PE1/TIOC0B/_DRAK0/_CS1		138	GND
	113	5VCC		139	PB6/_IRQ4/A18/_BACK/-/-/-/-
	114	PE0/TIOC0A/_DREQ0/_CS0		140	PD9/D9/-/-
	115	5VCC		141	PB5/_IRQ3/_POE/-/SDA1/CK/ HRxD2/-
	116	PD0/D0/RXD2/AUDATA0		142	PD10/D10/-/-
	117	5VCC		143	PB4/_IRQ2/_POE2/-/SCL1/ _POEHIZ/HTxD2/SCK4
	118	GND		144	PD11/D11/-/-
	119	5VCC		145	PB3/_IRQ1/_POE1/-/SDA0/ _VCOSTOP/-/TXD4
	120	PD1/D1/TXD2/AUDATA1		146	PD12/D12/-/-
	121	5VCC		147	N.C.
	122	PD2/D2/SCK2/AUDATA2		148	GND
	123	N.C.		149	PB2/_IRQ0/_POE0/-/SCL0/ _OSCER/-/RXD4
	124	PD3/D3/-/AUDATA3		150	PD13/D13/-/-
	125	GND		151	PB1/A17/HRxD0/HRxD1/-/-/-/ SCK4
	126	PD4/D4/-/AUDRET		152	PD14/D14/-/-

Table 3.5   Pin Arrangement on HS7046EPH60H (cont)

User I/F 1	Pin No.	Signal Name	User I/F 1	Pin No.	Signal Name
UCN1	153	PB0/A16/HTxD0/HTxD1/-/-/-/-	UCN1	177	GND
	154	PD15/D15/-/-		178	PD24/D24/_DREQ0/_UBCTRG
	155	PC15/A15/-/-		179	PC6/A6/-/-
	156	PD16/D16/_IRQ0/AUDATA0		180	PD25/D25/_DREQ1/-
	157	GND		181	PC5/D5/-/-
	158	GND		182	PD26/D26/DACK0/-
	159	PC14/A14/-/-		183	PC4/A4/-/-
	160	PD17/D17/_IRQ1/AUDATA1		184	PD27/D27/DACK1/-
	161	PC13/A13/-/-		185	GND
	162	PD18/D18/_IRQ2/AUDATA2		186	GND
	163	PC12/A12/-/-		187	PC3/A3/-/-
	164	PD19/D19/_IRQ3/AUDATA3		188	PD28/D28/_CS2/-
	165	PC11/A11/-/-		189	PC2/A2/-/-
	166	GND		190	PD29/D29/_CS3/-
	167	GND		191	PC1/A1/-/-
	168	PD20/D20/_IRQ4/_AUDRST		192	PD30/D30/_IRQOUT/-
	169	PC10/A10/-/-		193	PC0/A0/-/-/HRxD1
	170	PD21/D21/_IRQ5/AUDMD		194	PD31/D31/_ADTRG/-
	171	PC9/A9/-/-		195	GND
	172	PD22/D22/_IRQ6/AUDCK		196	GND
	173	PC8/A8/-/-		197	GND
	174	PD23/D23/_IRQ7/_AUDSYNC		198	GND
	175	PC7/A7/-/-		199	GND
	176	GND		200	GND

RENESAS

Table 3.5 Pin Arrangement on HS7046EPH60H (cont)

User I/F 2	Pin No.	Signal Name	User I/F 2	Pin No.	Signal Name
UCN2	1	GND	UCN2	31	N.C. (SAB1_P)
	2	GND		32	MD3
	3	GND		33	N.C. (SAB2_P)
	4	GND		34	MD2
	5	GND		35	N.C. (SAB3_P)
	6	GND		36	MD1
	7	GND		37	GND
	8	GND		38	GND
	9	N.C. (S0CLR0_N)		39	N.C. (SAB4_P)
	10	N.C. (S1CLR0_N)		40	MD0
	11	N.C. (S0CLR1_N)		41	N.C. (SAB5_P)
	12	N.C. (S1CLR1_N)		42	NMI
	13	N.C. (S0CLR2_N)		43	N.C. (SAB6_P)
	14	N.C. (S1CLR2_N)		44	_HSTBY
	15	N.C. (S0CLR3_N)		45	N.C. (SAB7_P)
	16	N.C. (S1CLR3_N)		46	N.C.
	17	GND		47	GND
	18	GND		48	VCC
	19	N.C. (S2CLR0_N)		49	N.C. (SAB8_P)
	20	N.C. (S3CLR0_N)		50	VCC
	21	N.C. (S2CLR1_N)		51	N.C. (SAB9_P)
	22	N.C. (S3CLR1_N)		52	VCC
	23	N.C. (S2CLR2_N)		53	N.C. (SAB10_P)
	24	N.C. (S3CLR2_N)		54	PVCC
	25	N.C. (S2CLR3_N)		55	N.C. (SAB11_P)
	26	N.C. (S3CLR3_N)		56	PVCC
	27	GND		57	GND
	28	GND		58	PVCC
	29	N.C. (SAB0_P)		59	N.C. (SVID4_N)
	30	MD4		60	PVCC

RENESAS

Table 3.5 Pin Arrangement on HS7046EPH60H (cont)

User I/F 2	Pin No.	Signal Name	User I/F 2	Pin No.	Signal Name
UCN2	61	N.C. (SVID3_N)	UCN2	90	N.C. (S2REQ3_N)
	62	PVCC		91	N.C. (S3REQ0_N)
	63	N.C. (SVID2_N)		92	N.C.
	64	PVCC		93	N.C. (S3REQ1_N)
	65	N.C. (SVID1_N)		94	N.C.
	66	N.C.		95	N.C. (S3REQ2_N)
	67	GND		96	N.C.
	68	GND		97	N.C. (S3REQ3_N)
	69	N.C. (SVID0_N)		98	N.C.
	70	_RES		99	N.C. (S3REQ4_N)
	71	N.C. (S0REQ0_N)		100	N.C.
	72	N.C. (SADTRG2_N)		101	GND
	73	N.C. (S0REQ1_N)		102	AVCC
	74	N.C. (SADTRG1_N)		103	N.C.
	75	N.C. (S0REQ2_N)		104	AVCC
	76	N.C. (SADTRG0_N)		105	N.C. (EVAVCC)
	77	GND		106	AVCC
	78	GND		107	N.C. (EVAVCC)
	79	N.C. (S0REQ3_N)		108	AVCC
	80	N.C. (SMTUADTRG_N)		109	N.C. (EVAVCC)
	81	N.C. (S1REQ0_N)		110	AVref
	82	N.C. (SMMTADTRG_N)		111	N.C.
	83	N.C. (S1REQ1_N)		112	AVref
	84	N.C. (S2REQ0_N)		113	N.C.
	85	N.C. (S1REQ2_N)		114	AVSS
	86	N.C. (S2REQ1_N)		115	N.C.
	87	GND		116	AVSS
	88	N.C. (S2REQ2_N)		117	N.C.
	89	N.C. (S1REQ3_N)		118	AVSS

RENESAS

Table 3.5 Pin Arrangement on HS7046EPH60H (cont)

User I/F 2	Pin No.	Signal Name	User I/F 2	Pin No.	Signal Name
UCN2	119	N.C.	UCN2	140	GND
	120	AVSS		141	GND
	121	GND		142	PF8/AN8
	122	PF0/AN0		143	PA6/TCLKA/_CS2/_RD/-/RXD2/-/-
	123	N.C.		144	PF9/AN9
	124	PF1/AN1		145	PA7/TCLKB/_CS3/_WAIT/-/TXD2/-/-
	125	N.C.		146	PF10/AN10
	126	PF2/AN2		147	PA8/TCLKC/_IRQ2/D0/-/RXD3/-/-
	127	PA0/RXD0/-/-/A0/_POE0/RXD2/-		148	PF11/AN11
	128	PF3/AN3		149	PA9/TCLKD/_IRQ3/D1/-/TXD3/-/-
	129	PA1/TXD0/-/-/A1/_POE1/TXD2/-		150	GND
	130	GND		151	GND
	131	GND		152	PF12/AN12
	132	PF4/AN4		153	PA10/_CS0/_RD/D2/TCK/SCK2/-/-
	133	PA2/SCK0/_DREQ0/IRQ0/A2/PCIO/ACK2/-		154	PF13/AN13
	134	PF5/AN5		155	PA11/_CS1/_ADTRG/D3/-/SCK3/-/-
	135	PA3/RXD1/-/_WRL/A3/_POE4/RXD3/-		156	PF14/AN14
	136	PF6/AN6		157	PA12/_WRL/_UBCTRG/D4/TDI/-/-/-
	137	PA4/TXD1/-		158	PF15/AN15
	138	PF7/AN7		159	PA13/_WRH/POE4/D5/TDO/_BREQ/-/-
	139	PA5/SCK1/_DREQ1/_IRQ1/A5/_POE6/ACK3/-		160	GND

RENESAS

Table 3.5   Pin Arrangement on HS7046EPH60H (cont)

User I/F 2	Pin No.	Signal Name	User I/F 2	Pin No.	Signal Name
UCN2	161	GND	UCN2	181	GND
	162	PG0/AN16		182	EXTAL
	163	PA14/_RD/POE5/D6/ TMS/-/-/-		183	PA22/_WRHL/-/-/-/-/-/-
	164	PG1/AN17		184	GND
	165	PA15/CK/_POE6/D7/ _TRES/_BACK/-/-		185	PA23/_WRHH/-/-/-/-/-/-
	166	PG2/AN18		186	N.C. (CKEO_P)
	167	PA16/-/_POE7/-/-/-/-/-		187	_WDTOVF
	168	PG3/AN19		188	GND
	169	PA17/_WAIT/PCIO/-/-/-/-		189	PINMD1*
	170	N.C.		190	FWP
	171	GND		191	PINMD0*
	172	AVref		192	N.C. (SVRES_N)
	173	PA18/_BREQ/DRAK0/-/-/-/-/-		193	GND
	174	AVref		194	GND (TGBON2)
	175	PA19/_BACK/DRAK1/-/-/-/-/-		195	GND
	176	AVref		196	GND
	177	PA20/-/-/-/-/-/-/-		197	GND
	178	AVref		198	GND
	179	PA21/-/-/-/-/-/-/-		199	GND
	180	N.C.		200	GND

Note: Signals PINMD1 and PINMD0 are used for recognition of the target MCUs. Pin processing must be handled according to the table below.

Target MCU	PINMD1	PINMD0
SH7046	GND	GND
SH7047	GND	Vcc (pulled up at 1 kΩ)
SH7144	Vcc (pulled up at 1 kΩ)	GND
SH7145	Vcc (pulled up at 1 kΩ)	Vcc (pulled up at 1 kΩ)

RENESAS

3.4.3 Precautions on Connecting the User System

When connecting the evaluation chip board to the user system, note the followings:

1. Secure the E6000H station location.
   Place the E6000H station and evaluation chip board so that the station to trace cable is not bent or twisted, as shown below. A bent or twisted cable will impose stress on the user interface, leading to connection or contact failure. Make sure that the E6000H station is placed in a secure position so that it does not move and impose stress on the user interface during use.
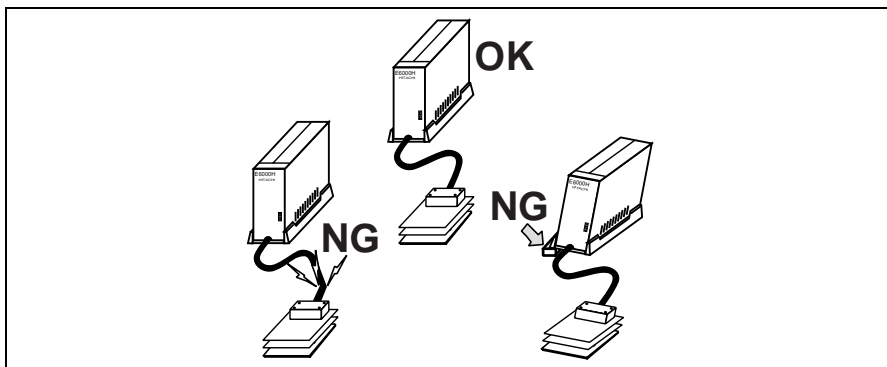
2. Make sure the power supply is off.
   Before connecting the evaluation chip board to the user system, check that the emulator and the user system are turned off.

3. Connect the UVCC to the user system power.
   The emulator monitors and determines whether the user system is turned on or off by the UVCC pins (HS7046EPH60H: pins 48, 50, 52 (VCC), 54, 56, 58, 60, 62, and 64 (PVCC)) on the user system interface connector (UCN2).
   Accordingly, after connecting the user system to the emulator, be sure to supply power to the VCC pins. Otherwise, the emulator assumes that the user system is not connected.
   When the user system is connected, check that the power of the user system is supplied to these pins.

RENESAS

3.5   Support of the Target MCU

3.5.1   Memory Space

The MCU has a 4-Gbyte memory space in its architecture.

On-Chip ROM Area

- Access to the on-chip ROM area

  The emulator includes substitute RAM for the MCU's on-chip ROM. The substitute RAM is accessed if an attempt is made to access the on-chip ROM. The on-chip ROM area access differs between user program execution and the emulator commands.

  Only read access is enabled during execution of the user program. A break occurs if the on-chip ROM area is written to. However, if write access to the on-chip ROM area has been selected in the [Configuration] dialog box, both reading and writing are enabled.

  For access with emulator functions (Memory window or loading), read and write are always enabled.

  The on-chip ROM area is read in one state and written to in six states.

- Flash memory

  The emulator does not support the flash memory.

- Capacitance between on-chip ROM and on-chip RAM

  In this emulator, note that the capacitance between on-chip ROM and on-chip RAM differs from that of the product MCU. The display of the [Status] window applies to the product MCU.

Table 3.6   Capacitance between On-Chip ROM and On-Chip RAM

Item	Emulator	Product MCU
Capacitance of on-chip ROM	1 Mbyte (H'00000000 to H'0000FFFF)	Refer to the MCU's hardware manual.
Capacitance of on-chip RAM	16 kbytes (H'FFFF0000 to H'FFFF3FFF)	
	16 kbytes (H'FFFFC000 to H'FFFFFFFF)	

On-Chip I/O Area

If an attempt is made to access the on-chip I/O area, the on-chip I/O area in the MCU installed in the emulator is accessed. To break the user program when the on-chip I/O area is written to or accessed, use the hardware break or internal break.

External Memory Area

The MCU's external memory area can be set with all memory attributes that the emulator supports.

RENESAS

3.5.2   Low Power-Consumption Mode (Sleep, Software Standby, and Hardware Standby)

For reduced power consumption, the MCU has sleep, software standby, and hardware standby modes.

Hardware Standby Mode

Since the _HSTBY signal from the user system is not input to the MCU in the emulator, the emulator does not support this mode.

Sleep and Software Standby Modes

- Break

    The sleep and software standby modes can be cleared with either the normal clearing function or with the break condition satisfaction (forced break), and the program breaks. When restarting after a break, the user program will restart at the instruction following the SLEEP instruction.

- Trace

    Trace information is not acquired in these modes.

- Memory access with emulator functions

    For information on displaying and modifying the contents of memory in the sleep and software standby modes, refer to section 5.4, Displaying and Modifying the Contents of Memory.

3.5.3   Interrupts

During execution and step execution, the user can interrupt the MCU.

During halting emulation (break mode), the interrupt source is retained. The mode transits the interrupt processing immediately after emulation is restarted.

3.5.4   Control Input Signals (_RES, _BREQ, and _WAIT)

The MCU control input signals are _RES, _BREQ, and _WAIT.

The _RES signal is only valid when emulation has been started with normal program execution (i.e., the _RES signal is invalid when emulation has been started with step execution). While emulation is being halted (break), the input of the _RES signal to the MCU is not possible. The input of the _RES signal during execution or step execution is disabled by a setting in the [Configuration Properties] dialog box.

The _BREQ and _WAIT signals are always valid during emulation or emulation halted (break). The input of those signals is disabled by a setting in the [Configuration Properties] dialog box.

3.5.5   Bus State Controller (BSC)

The wait state controller has a programmable wait mode and a WAIT pin input mode. The programmable wait mode is valid when the emulation memory or user external memory is accessed, but input to the user WAIT pin is only valid when user external memory is accessed.

3.5.6   Watchdog Timer (WDT)

While emulation is being halted (during break), counting up the WDT timer counter (TCNT) is suspended, and restarted when emulation is executed again (user mode).

During break mode, a prescaler, which supplies a clock to TCNT, operates continuously. Since the phase of the prescaler may be unmatched before or after emulation transits the break mode, the period before an overflow occurs will differ by ±1 cycle in the prescaler's clock cycle.

RENESAS

3.5.7 A/D Converter

The A/D converter has AVcc, AVss, Avref, and _ADTRG pins as well as the analog input pins. As the A/D converter operates with an independent power supply, connect AVcc (the power supply pin) to the A/D power supply on the user system.

Notes: 1. When not using the A/D converter, connect AVcc to Vcc.
2. As the user system interface cable, printed circuit boards, and protective circuits are connected between the MCU and the user system in the emulator pod, the conversion precision is lower than that of the SH7046 series or SH7144 series MCUs. At final debugging of the user system using the A/D converter, use the actual SH7046 series or SH7144 series (F-ZTAT microcomputer) MCUs.

3.5.8 Emulator State and On-Chip Modules

Some on-chip modules do not operate when the emulator is in break mode. Table 3.7 shows the relation between the emulator's state and operation of the on-chip modules.

Table 3.7 Emulator State and Operation of On-Chip Modules

On-Chip Module	Operation During Emulation Halted (Break)	Operation During Emulation (Execution or Step Execution)
WDT (watchdog timer)	No	Yes
MTU (multi-function timer pulse unit)	Yes	Yes
MMT (motor management unit)	Yes	Yes
CMT (compare-match timer)	Yes	Yes
SCI (serial communication interface)	Yes	Yes
DTC (data transfer controller)	Yes[1]	Yes
HCAN2	Yes	Yes
UBC (user break controller)	No	Yes
AUD (advanced user debugger)	Yes	Yes
I/O port	Yes	Yes
A/D converter	Yes	Yes
H-UDI (user debugging interface)	Not available[2]	Not available[2]

Notes: 1. If a break occurs during a DTC cycle (vector read, read/write of transferred information, or data read/write), the DTC continues operation until the DTC cycle is complete. The DTC resumes operation after it returns to emulation.
2. Not supported by the emulator.

3.5.9　Pin Functions

This emulator supports the extended functions that are not included in the product MCU. Therefore, the pin status, which does not exist in the SH7046, SH7047, SH7144, or SH7145, may be displayed. Table 3.8 shows the difference between the pins of which status is monitored on the emulator and the product MCU.

Table 3.8　Pin Functions

Pin Name	Emulator	SH7046F	SH7047F	SH7144F	SH7145F
_RES	Yes	Yes	Yes	Yes	Yes
NMI	Yes	Yes	Yes	Yes	Yes
_IRQ0	Yes	Yes	Yes	Yes	Yes
_IRQ1	Yes	Yes	Yes	Yes	Yes
_IRQ2	Yes	Yes	Yes	Yes	Yes
_IRQ3	Yes	Yes	Yes	Yes	Yes
_IRQ4	Yes	No	No	Yes	Yes
_IRQ5	Yes	No	No	Yes	Yes
_IRQ6	Yes	No	No	Yes	Yes
_IRQ7	Yes	No	No	Yes	Yes
_WAIT	Yes	No	Yes	Yes	Yes
_BREQ	Yes	No	Yes	Yes	Yes
_HSTBY	Yes	No	Yes	No	No

When the emulator monitors the status of multiplexed pins IRQ0 to IRQ7, WAIT, and BREQ, the corresponding ports must be correctly set on the [Pin Select Registers] page of the [Configuration Properties] dialog box.

RENESAS

3.5.10 Different Initial Values of Registers in the Emulator

Note that the emulator initializes some general or control registers whenever the system is activated or the MCU is reset by commands.

Table 3.9 Initial Values of Registers in the MCU and the Emulator

Register Name	Emulator		MCU (Reset)
	Power On	Reset (Reset CPU)	
PC	Power-on reset vector PC value	Power-on reset vector PC value	Power-on reset vector PC value
R0 to R14	H'00000000	Value before reset	Undefined
R15 (SP)	Power-on reset vector SP value	Power-on reset vector SP value	Power-on reset vector SP value
SR	H'000000F0	H'000000F0	H'00000XFX*
GBR	H'00000000	Value before reset	Undefined
VBR	H'00000000	H'00000000	H'00000000
MACH	H'00000000	Value before reset	Undefined
MACL	H'00000000	Value before reset	Undefined
PR	H'00000000	Value before reset	Undefined

Note: X indicates an undefined value.

Section 4   Diagnostic Test Procedure

This section describes the diagnostic test procedure using the E6000H diagnostic program.

4.1   System Set-Up for Diagnostic Program Execution

To execute the diagnostic program, use the following hardware; do not connect the user system interface board and user system.

- E6000H (HS7046EPH60H)
- Host computer
- The E6000 PC interface board which will be one of the following boards:

  PCI bus interface board (HS6000EIC01H or HS6000EIC02H)

  PC card interface (HS6000EIP01H)

  LAN adapter (HS6000ELN01H)

  USB adapter (HS6000EIU01H or HS6000EIU02H)

1. Install the E6000 PC interface board in the host computer and connect the supplied PC interface cable to the board.
2. Connect the PC interface cable to the E6000H.
3. Connect the supplied AC power cable to the E6000H.
4. Initiate the host computer to make it enter the command input wait state of the DOS prompt (Windows® 98SE or Windows® Me) or command prompt (Windows NT®, Windows® 2000, or Windows® XP). If the property of the prompt window is not a mode for displaying the whole screen, press the [Alt + Enter] key to switch the mode. To set back the mode, press the [Alt + Enter] key. The display of the screen is switched regardless of the OS being used.

Note: In the MS-DOS prompt, if the display of the screen is not switched after pressing the [Alt + Enter] key, mark the [Alt+Enter] check box in [Windows shortcut keys] of the [Misc] page on the [MS-DOS Prompt Properties] dialog box and click the [Apply] button to update the settings, as shown in figure 4.1.

RENESAS
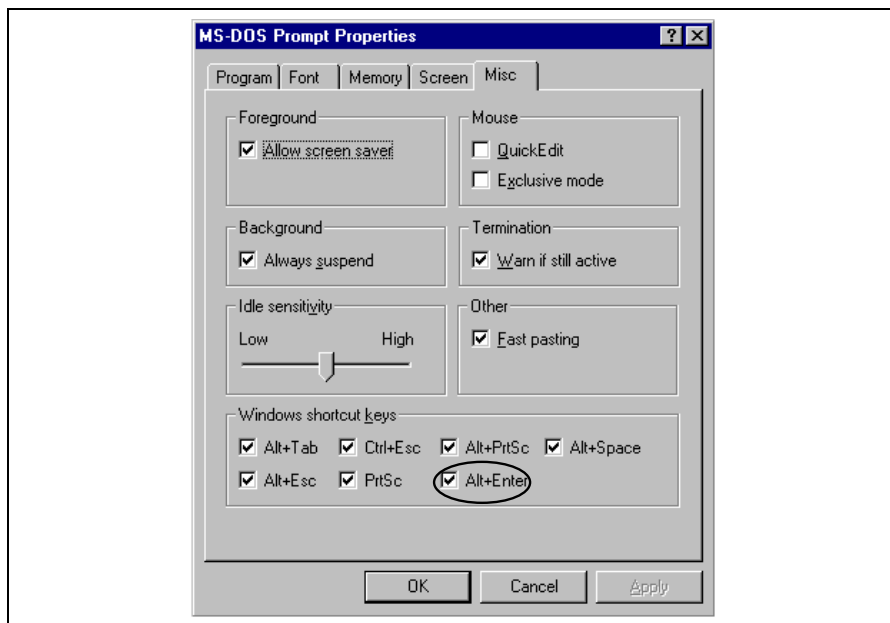
Figure 4.1   [Misc] Page

5.   Turn on the E6000H emulator switch.

Note: To execute the diagnostic program, firstly turn on the power of the emulator. In the diagnostic program, the initial state of hardware is checked. Therefore, after the power is turned on, do not activate the High-performance Embedded Workshop before executing the diagnostic program.

4.2   Test Item of the Diagnostic Program

Table 4.1 shows the test items of this diagnostic program.

Table 4.1   Test Items of the Diagnostic Program

Test No.	Test Item	Description
1	Main Board Access	Register test in the E6000H main board
2	Emulation Board Access	Register test in the E6000H emulation board
3	Evaluation Chip Board Access	Register test in the E6000H evaluation chip board
4	Basic Function	Test for the basic function
5	GO to BREAK Time Measurement	Test for the execution time measurement function
6	Emulation Monitor	Test for emulation monitor
7	G/A Break Function	Test for the G/A break function
8	G/A Performance Analysis Function	Test for the G/A performance measurement function
9	G/A Monitor Function	Test for the G/A monitoring function
10	G/A Parallel RAM Monitor	Test for the G/A parallel RAM monitoring function
11	G/A Trace Function	Test for the G/A trace function
12	Combination	Test for combination of each function
13	Parallel Access	Test for the parallel access function
14	PC Coverage	Test for the PC coverage function

RENESAS

4.3 Diagnostic Test Procedure Using the Diagnostic Program

Insert the CD-R (HS7046EPH60SR supplied with the E6000H) into the CD-ROM drive of the host computer, move the current directory to <Drive>:\Diag with a command prompt, and enter one of the following commands according to the PC interface board used to initiate the diagnostic program:

1. PCI bus interface board (HS6000EIC01H or HS6000EIC02H)
   > TM7046 –PCI (RET)

2. PC card interface (HS6000EIP01H)
   > TM7046 –PCCD (RET)

3. LAN adapter (HS6000ELN01H)
   > TM7046 –ELN (RET)

4. USB adapter (HS6000EIU01H or HS6000EIU02H)
   > TM7046 –USB (RET)

The High-performance Embedded Workshop must be installed before the test program is executed.

Be sure to initiate the diagnostic program from <Drive>:\Diag. Do not initiate it from a directory other than <Drive>:\Diag, such as > <Drive>:\Diag\TM7046 –PCI (RET). If the diagnostic program is initiated when the current directory is not <Drive>:\Diag, the diagnostic program will not operate correctly.

When –S is added to the command line such as > TM7046 –PCI –S (RET), steps 1 to 14 will be repeatedly executed. To stop the execution, enter Q.

Notes: 1.   <Drive> is a drive name for the CD-ROM drive.
       2.   Do not remove the CD-R from the CD-ROM drive during test program execution.

RENESAS

The following messages are displayed during the test. There are 14 steps in this test (when a PCI interface board is used, the time for the test will be about three minutes).

Message	Description
E6000H SH7046 Emulator Tests Vx.x.xx.xxx	Test program start message. x.x shows the version number.
Loading driver ....OK (Use PCI)	Shows that the PC interface board is correctly installed in the host computer.
Initializing driver ....OK Searching for interface card ....OK Checking emulator is connected ....OK	Shows that the E6000H emulator is correctly connected to the host computer.
Emulator board information: Main board ID: H'0 Emulation board ID: H'002	
	Shows the ID number of the E6000H emulator.
Normal started at Wed Jan 22 09:43:43 2003	Shows the time when the diagnostic program has started.

```
******* NORMAL TEST - Press 'Q' to stop *******    (COUNT=0001)
1. Main Board Access
    01) Registers Initial Value Check .............................OK
    02) Registers Write/Verify ....................................OK
    03) DPRAM Address Decode Test ...............................SKIP
    04) DPRAM Marching Test .....................................SKIP
    05) Trace Memory Address Decode Test ..........................OK
    06) Trace Memory Marching Test ................................OK
    07) G/A Registers Initial Value Check .........................OK
    08) G/A Registers Write/Verify ................................OK
2. Emulation Board Access
    01) Registers Initial Value Check .............................OK
    02) Registers Write/Verify ....................................OK
    03) H-UDI Interface Registers Initial Value Check .............OK
    04) H-UDI Interface Registers Write/Verify ....................OK
    05) MAPR Interface Registers Initial Value Check ..............OK
    06) MAPR Interface Registers Write/Verify .....................OK
    07) PRALR Interface Registers Initial Value Check .............OK
    08) PRALR Interface Registers Write/Verify ....................OK
```

RENESAS

```
3. Evaluation Board Access
     01) Registers Initial Value Check .............................OK
     02) Registers Write/Verify ...................................OK
     03) H-UDI IDCODE Check .......................................OK
     04) Firmware BOOT ............................................OK
     05) Configuration Set ........................................OK
     06) Emulation Memory CS0 Test ................................OK
     07) Emulation Memory CS1 Test ................................OK
     08) Emulation Memory CS2 Test ................................OK
     09) Emulation Memory CS3 Test ................................OK
     10) Emulation Memory Test ....................................OK
     11) INROM Test ...............................................OK
     12) INRAM Test ...............................................OK
4. Basic Function
     01) GO to BREAK ..............................................OK
     02) RESET GO .................................................OK
     03) STEP .....................................................OK
     04) KEYBREAK .................................................OK
     05) BRKCONT ..................................................OK
     06) ERAM WRITE PROTECT Test ..................................OK
     07) INROM WRITE PROTECT Test .................................OK
     08) ERAM GOD Test ............................................OK
5. GO to BREAK Time Measurement
     01) Counter Test Mode ........................................OK
     02) EMU 10MHz MPU 20MHz Sampling 20ns  .......................OK
     03) EMU 10MHz MPU 40MHz Sampling 1.6u ........................OK
     04) EMU 10MHz MPU 40MHz Sampling 52u .........................OK
     05) EMU 10MHz MPU 40MHz Sampling MPU .........................OK
     06) EMU 10MHz MPU 40MHz Sampling MPU/2 .......................OK
     07) EMU 10MHz MPU 40MHz Sampling MPU/4 .......................OK
     08) EMU 10MHz MPU 40MHz Sampling MPU/8 .......................OK
     09) EMU 10MHz MPU 20MHz Sampling 100ns .......................OK
6. Emulation Monitor
     01) TRES .....................................................OK
     02) ASEST3 - 0 ...............................................OK
     03) VCC3VNG ..................................................OK
     04) PVCC-NG ..................................................OK
```

RENESAS

7. G/A Break Function

01) Address Condition ....OK

02) Data Condition ....OK

03) Control Signal Condition (ASEDSHH/HL/LH/HL) ....OK

04) Function Code Condition (ASEAST3-0) ....OK

05) Control Signal Condition (ASEAA3-0) ....OK

06) Control Signal Condition (DMA) ....OK

8. G/A Performance Analysis Function

01) Time Measurement (20ns Sampling) ....OK

02) Pckcr Emclk Test 4MHz ....OK

03) Pckcr Emclk Test 12MHz ....OK

04) Pckcr Emclk Test 20MHz ....OK

05) Pckcr Emclk Test 25MHz ....OK

06) Pckcr Emclk Test 32MHz ....OK

07) Pckcr Emclk Test 50MHz ....OK

9. G/A Monitor Function

01) RUN ....OK

02) VCCDOWN ....OK

03) NOCLK ....OK

04) TIMEOUT ....OK

10. G/A Parallel RAM Monitor

01) PRAM Monitor (BYTE) ....OK

02) PRAM Monitor (WORD) ....OK

03) PRAM Monitor (LONG WORD) ....OK

04) PRAM Monitor (ERAM CS0(8Bit) LONG WORD) ....OK

05) PRAM Monitor (ERAM CS1(16Bit) WORD) ....OK

06) PRAM Monitor (ERAM CS2(32Bit) BYTE) ....OK

07) PRAM Monitor (ERAM CS3(32Bit) LONG WORD) ....OK

11. G/A Trace Function

01) Free Trace ....OK

02) Trace Stop ....OK

03) Time Stamp ....OK

04) Trace Suppress ....OK

05) Range Trace ....OK

06) Sequential Trace Stop ....OK

07) TBM Overflow Trace ....OK

08) Timeout Trace Stop ....OK

09) Subroutine Trace ....OK

10) INROM Trace ....OK

RENESAS

```
12. Combination
    01) B to A Time Measurement( FPGA counter ) ...................OK
    02) B to A Time Measurement( G/A counter ) ....................OK
    03) D to C Time Measurement( G/A counter ) ....................OK
13. Parallel Access
    01) INROM Parallel Read Access(Byte/Long) .....................OK
    02) INROM Parallel Write Access(Byte/Long) ....................OK
    03) ERAM Parallel Read Access(Byte/Long) ......................OK
    04) ERAM Parallel Write Access(Byte/Long) .....................OK
    05) ERAM Parallel Write Access(Byte/Word/Long) ................OK
    06) ERAM Parallel Read Access(Byte/Word/Long) .................OK
    07) INROM Parallel Write Access(Byte/Word/Long) ...............OK
    08) INROM Parallel Read Access(Byte/Word/Long) ................OK
14. PC Coverage
    01) PC Coverage ...............................................OK
```

`Normal stopped at Wed Jan 22 09:46:55 2003`	Shows the time when the diagnostic program has ended.
`Tests run for 0H:3M:12S`	Shows the execution time of the diagnostic program.

```
Summary:
    Tests performed 1 time(s).
  1. Main Board Access           : 0 Error(s)
  2. Emulation Board Access      : 0 Error(s)
  3. Evaluation Board Access     : 0 Error(s)
  4. Basic Function              : 0 Error(s)
  5. GO to BREAK Time Measurement : 0 Error(s)
  6. Emulation Monitor           : 0 Error(s)
  7. G/A Break Function          : 0 Error(s)
  8. G/A Performance Analysis Function : 0 Error(s)
  9. G/A Monitor Function        : 0 Error(s)
 10. G/A Parallel RAM Monitor    : 0 Error(s)
 11. G/A Trace Function          : 0 Error(s)
 12. Combination                 : 0 Error(s)
 13. Parallel Access             : 0 Error(s)
 14. PC Coverage                 : 0 Error(s)
```

Shows the total of the number of errors occurred in each test item.

RENESAS

Debugger Part

Section 1   Overview

The Debugger Part includes the following information.

Table 1.1 Debugger Part Contents

| Section	Title	Content
2	Preparation before Use	This section starts with creation of a workspace and ends with connection to the emulator.
3	Debugging	This section describes this emulator 's peculiar debugging operation and the associated windows and dialog boxes.
		Refer to the High-performance Embedded Workshop user's manual about High-performance Embedded Workshop common functions as below.
		— Preparations for Debugging
		— Viewing a Program
		— Operating Memory
		— Displaying Memory Contents as Waveforms
		— Displaying Memory Contents as an Image
		— Modifying the variables
		— Viewing the I/O Memory
		— Looking at Registers
		— Executing Your Program
		— Viewing the Function Call History
		— Debugging with the Command Line Interface
		— Elf/Dwarf2 Support
		— Looking at Labels
4	Tutorial	This section describes how to use the emulator functions by using a tutorial program provided with the emulator.
5	Software Specifications and Notes Specific to This Product	This section describes software specifications and notes regarding the emulator.
6	Error Messages	This section describes the contents of error messages that may occur while the emulator is in use, and solutions to them.

RENESAS

Section 2   Preparation before Use

2.1   Method for Activating High-performance Embedded Workshop

To activate the High-performance Embedded Workshop, follow the procedure listed below.

1. Connect the emulator to the host computer.
2. Connect the user system interface cable to the connector of the emulator if you use the user system interface cable. This is not necessary when you do not use the user system interface cable.
    Turn on the emulator. Be sure to turn on the user system before supplying power to the emulator if you use the user system.
3. Activate the High-performance Embedded Workshop from [Programs] in the [Start] menu.
4. The [Welcome!] dialog box is displayed.
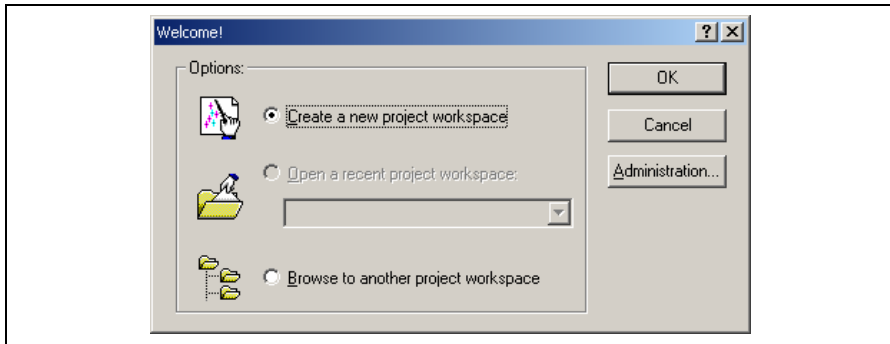
Figure 2.1   [Welcome!] Dialog Box

[Create a new project workspace] radio button: Creates a new workspace.

[Open a recent project workspace] radio button: Uses an existing workspace and displays the history of the opened workspace.

[Browse to another project workspace] radio button: Uses an existing workspace; this radio button is used when the history of the opened workspace does not remain.

In this section, we describe the following three ways to start up the High-performance Embedded Workshop:

•   [Create a new project workspace] - a toolchain is not in use

•   [Create a new project workspace] - a toolchain is in use

•   [Browse to another project workspace]

The method to create a new workspace depends on whether a toolchain is or is not in use. Note that this emulator product does not include a toolchain. Use of a toolchain is available in an environment where the H8S, H8/300 series C/C++ compiler package or the SuperH™ RISC engine C/C++ compiler package has been installed. For details on this, refer to the manual attached to the H8S, H8/300 series C/C++ compiler package or the SuperH™ RISC engine C/C++ compiler package.

RENESAS

2.1.1   Creating a New Workspace (Toolchain Not Used)

1. In the [Welcome!] dialog box that is displayed when the High-performance Embedded Workshop is activated, select [Create a new project workspace] radio button and click the [OK] button.
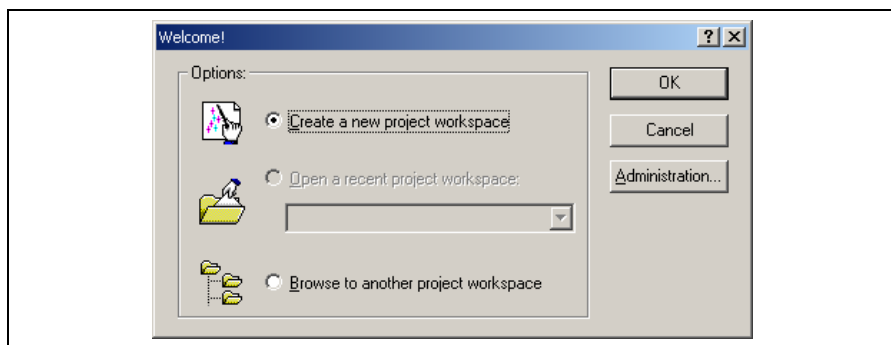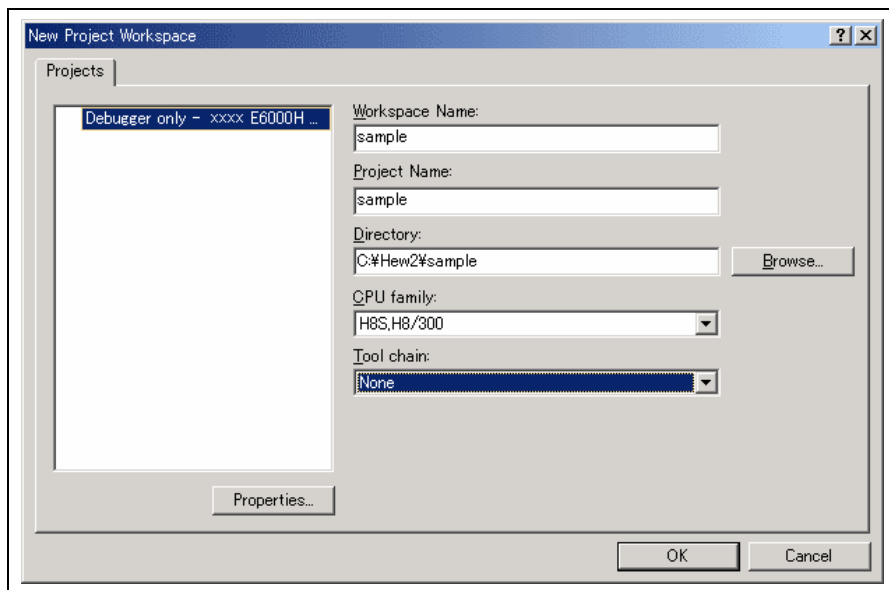

Figure 2.2   [Welcome!] Dialog Box

2. Creation of a new workspace is started. The following dialog box is displayed.

Figure 2.3 [New Project Workspace] Dialog Box

[Workspace Name] edit box: Enter the new workspace name.

[Project Name] edit box: Enter the project name. When the project name is the same as the workspace name, it needs not be entered.

[Directory] edit box: Enter the directory name in which the workspace will be created. Click the [Browse…] button to select a directory.

[CPU family] combo box: Select the target CPU family.

Other list boxes are used for setting the toolchain; the fixed information is displayed when the toolchain has not been installed.

Click the [OK] button.

3. Select the target platform of the session file. The following dialog box is displayed.
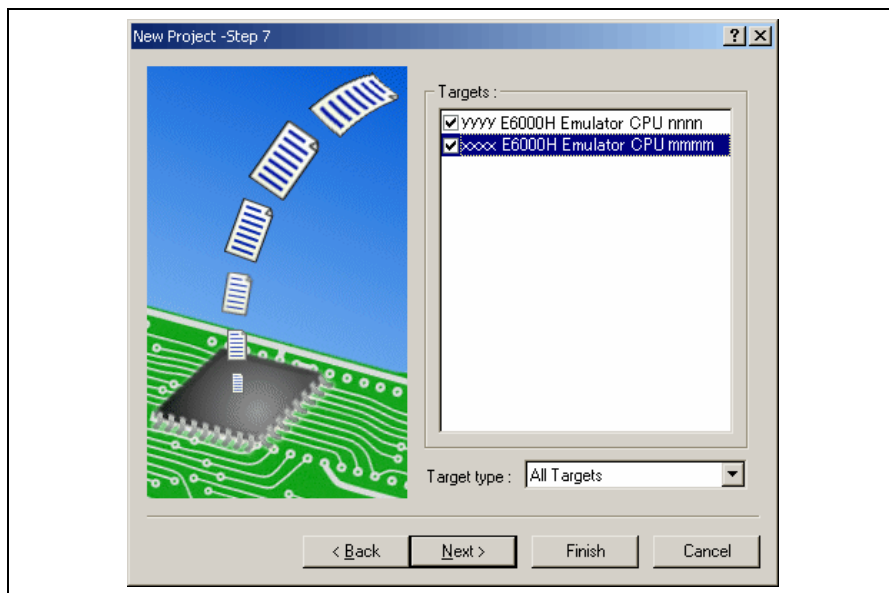
Figure 2.4   [New Project – Step 7] Dialog Box

The target platform for the session file used when the High-performance Embedded Workshop is activated must be selected here. Check the box against the target platform and then click the [Next] button.

RENESAS

4. Set the configuration file name. The configuration file saves the state of High-performance Embedded Workshop except for the emulator.
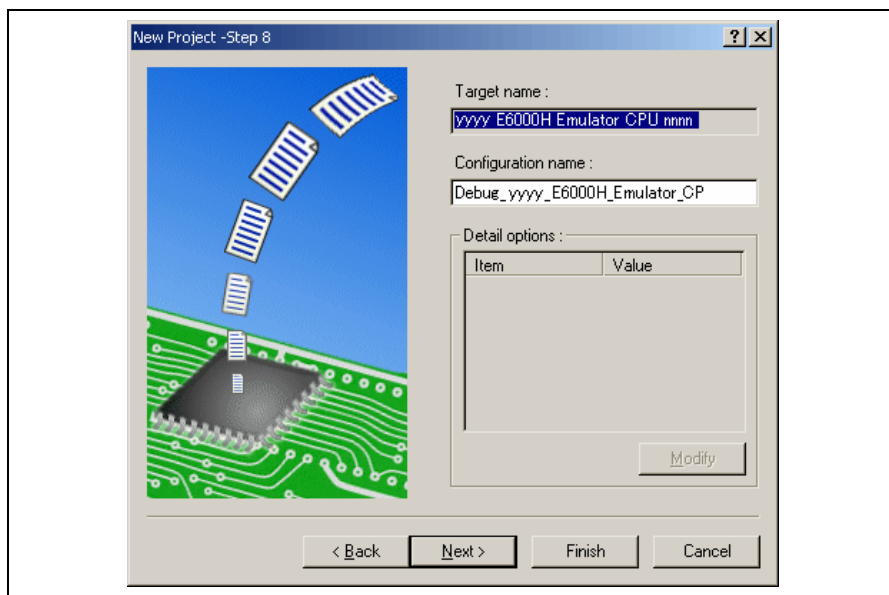
Figure 2.5   [New Project – Step 8] Dialog Box

If multiple target platforms were selected in the [New Project – Step 7] dialog box shown in figure 2.5, set the name of a configuration file for each of them, each time clicking the [Next] button to proceed to the next target.

Setting of the configuration file name is the end of the emulator settings.

Click the [Finish] button to display the [Summary] dialog box. Clicking the [OK] button activates the High-performance Embedded Workshop.

5. After the High-performance Embedded Workshop has been activated, the emulator is automatically connected. The message "Connected" is displayed on the [Debug] tab in the [Output] window to indicate the completion of connection.

RENESAS

2.1.2 Creating a New Workspace (Toolchain Used)

1. In the [Welcome!] dialog box that is displayed when the High-performance Embedded Workshop is activated, select the [Create a new project workspace] radio button and click the [OK] button.
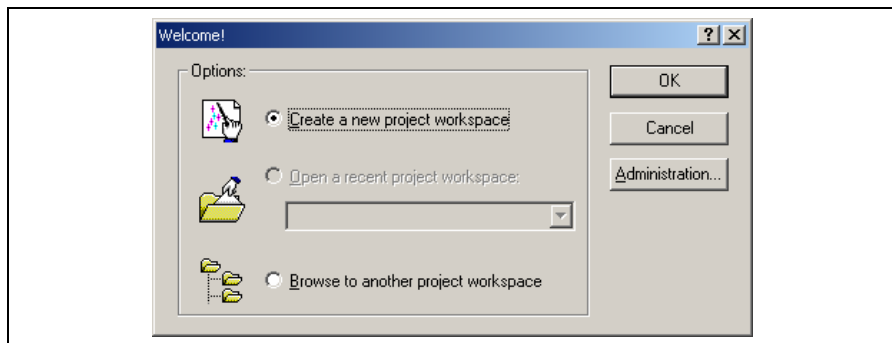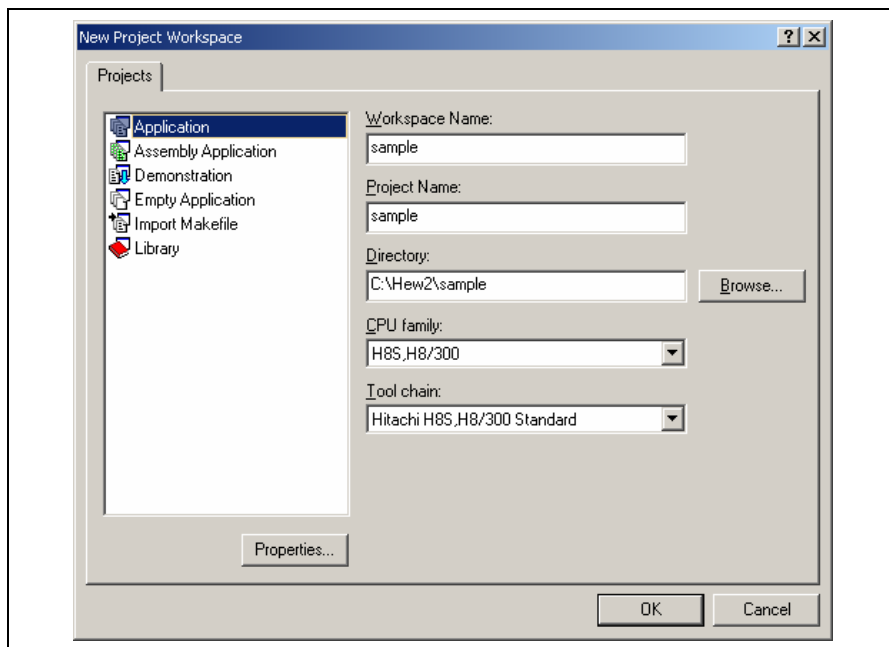

Figure 2.6   [Welcome!] Dialog Box

2. Creation of a new workspace is started. The following dialog box is displayed.

Figure 2.7   [New Project Workspace] Dialog Box

[Workspace Name] edit box: Enter the new workspace name.

[Project Name] edit box: Enter the project name. When the project name is the same as the workspace name, it needs not be entered.

[Directory] edit box: Enter the directory name in which the workspace will be created. Click the [Browse…] button to select a directory.

[CPU family] combo box: Select the target CPU family.

[Tool chain] combo box: Select the target toolchain name when using the toolchain. Otherwise, select [None].

[Project type] list box: Select the project type to be used.

Notes: When [Demonstration] is selected in the emulator, note the followings:
The [Demonstration] is a program for the simulator attached to the H8S, H8/300 compiler package or the SuperH™ RISC engine C/C++ compiler package. To use the generated source file, delete the printf statement in the source file.

RENESAS

3. Make the required setting for the toolchain. When the setting has been completed, the following dialog box is displayed.
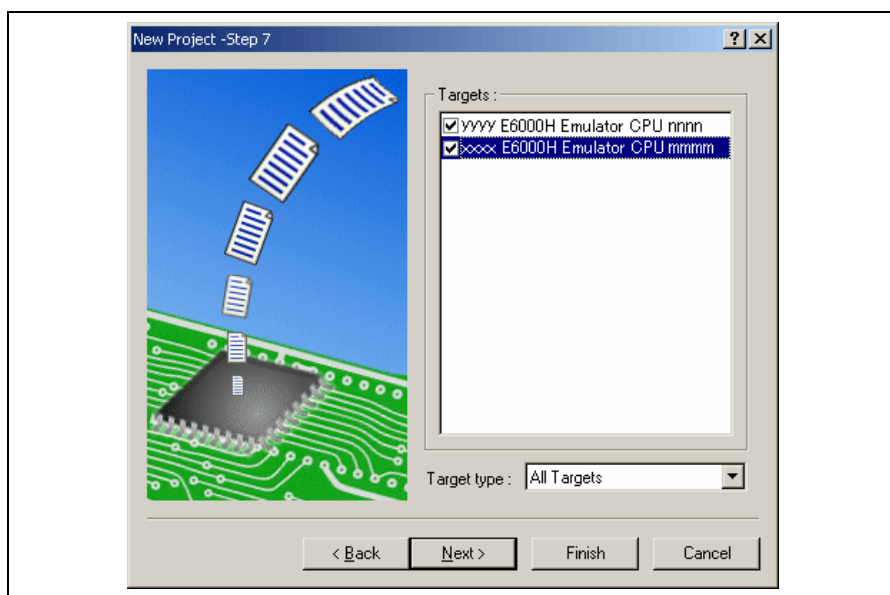
Figure 2.8 [New Project – Step 7] Dialog Box

The target platform for the session file used when the High-performance Embedded Workshop is activated must be selected here. Check the box against the target platform and then click the [Next] button.

RENESAS

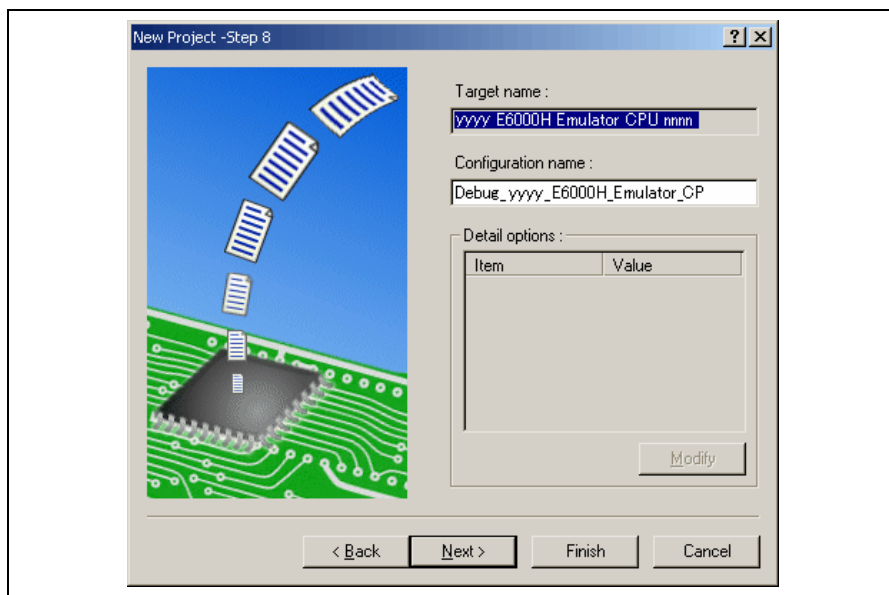4. Set the configuration file name. The configuration file saves the state of High-performance Embedded
    Workshop except for the emulator.

Figure 2.9   [New Project – Step 8] Dialog Box

If multiple target platforms were selected in the [New Project – Step 7] dialog box shown in figure 2.9, set
the configuration file name for each of them, each time clicking the [Next] button to proceed to the next
target.

Setting of the configuration file name is the end of the emulator settings.

Complete the creation of a new workspace according to the instructions on the screen. This activates the
High-performance Embedded Workshop.

5. After the High-performance Embedded Workshop has been activated, connect the emulator. However, it is
    not necessary to connect the emulator immediately after the High-performance Embedded Workshop has
    been activated.

    Select either of the following two ways to connect the emulator: connecting the emulator after the setting at
    emulator activation or without the setting at emulator activation. For details on the connection of the
    emulator, refer to section 2.2, Connecting the Emulator.

RENESAS

2.1.3 Selecting an Existing Workspace

1. In the [Welcome!] dialog box that is displayed when the High-performance Embedded Workshop is activated, select [Browse to another project workspace] radio button and click the [OK] button.
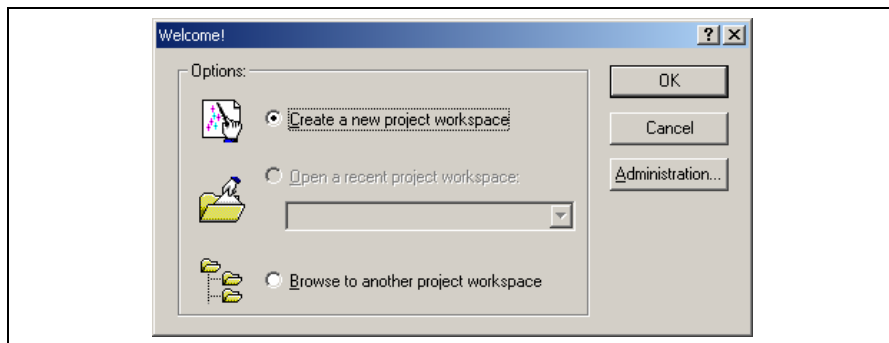
Figure 2.10   [Welcome!] Dialog Box

2. The [Open Workspace] dialog box is displayed. Select a directory in which you have created a workspace. After that, select the workspace file (.hws) and click the [Open] button.
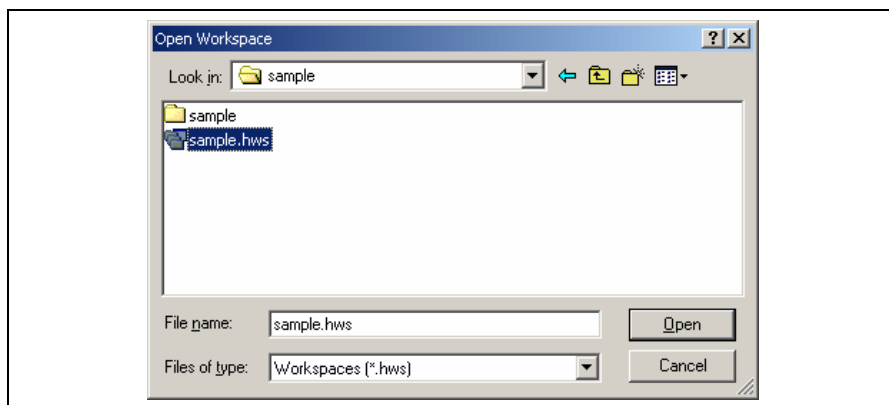
Figure 2.11   [Open Workspace] Dialog Box

3. This activates the High-performance Embedded Workshop and recovers the state of the selected workspace at the time it was saved.
When the saved state information of the selected workspace includes connection to the emulator, the emulator will automatically be connected. To connect the emulator when the saved state information does not include connection to the emulator, refer to section 2.2, Connecting the Emulator.

2.2   Connecting the Emulator

Select either of the following two ways to connect the emulator:

(a) Connecting the emulator after the setting at emulator activation

Select [Debug -> Debug Settings…] to open the [Debug Settings] dialog box. It is possible to register the download module or the command chain that is automatically executed at activation.

When the dialog box is closed after setting the [Debug Settings] dialog box, the emulator will automatically be connected.

(b) Connecting the emulator without the setting at emulator activation

Connect the emulator by simply switching the session file to one in which the setting for the emulator use has been registered.
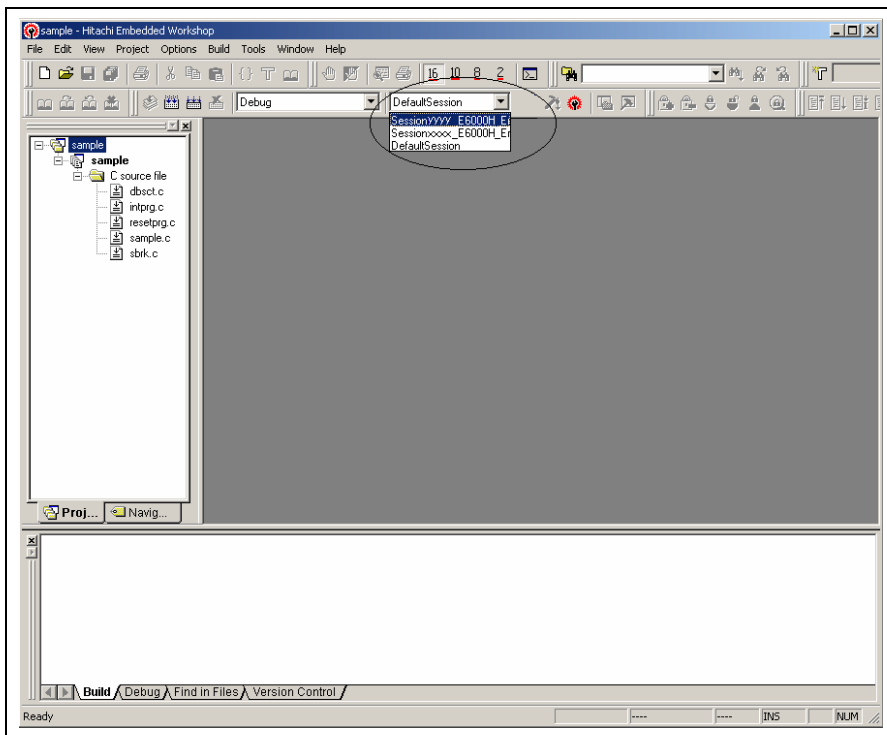

Figure 2.12   Selecting the Session File

In the list box that is circled in figure 2.12, select the session file name including the character string that has been set in the [Target name] text box in figure 2.9, [New Project – Step 8] dialog box. The setting for using the emulator has been registered in this session file.

Selecting [Debug -> Connect] connects the emulator.

2.3   Re-connecting the Emulator

When the emulator is disconnected, re-connection is possible by using the following methods.

Select [Debug -> Connect] or click the [Connect] toolbar button (![button]) to re-connect the emulator.

Note: When re-connecting the emulator, the load module must be registered to the High-performance
Embedded Workshop beforehand.

2.4   Ending the Emulator

The emulator can be exited by using the following two methods:

(a) Canceling the connection of the emulator being activated
Select [Debug -> Disconnect] or click on the [Disconnect] toolbar button (![button]).

(b) Exiting the High-performance Embedded Workshop

1.   Select [File -> Exit].
2.   A message box is displayed. If necessary, click the [Yes] button to save a session. After saving a session,
the High-performance Embedded Workshop exits.
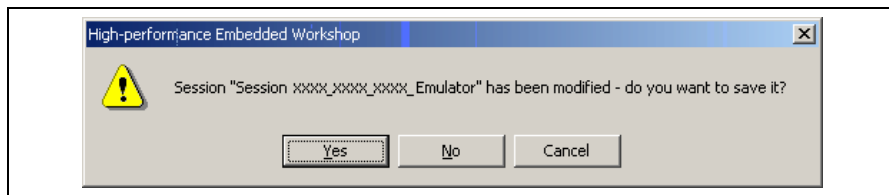
Figure 2.13   [Session has been modified] Message Box

RENESAS

Section 3   Debugging

This section describes the debugging operations and their related windows and dialog boxes.

3.1   Setting the Environment for Emulation

The method for setting the environment for emulation is described here. This environment must be set correctly before debugging is started.

3.1.1   Opening the [Configuration Properties] Dialog Box

Selecting [Setup -> Emulator -> System…] or clicking the [Emulator System] toolbar button (🔧) opens the [Configuration Properties] dialog box.

Figure 3.1   [Configuration Properties] Dialog Box ([General] Page)

RENESAS

[General] page

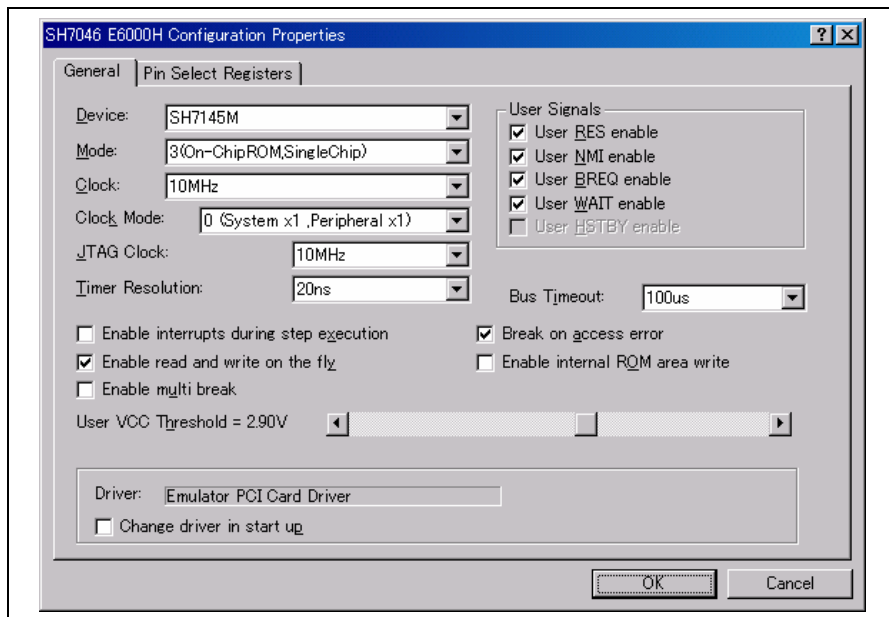|	
[Device]	Selects the target MCU to be emulated. See the hardware manual for details.
[Mode]	Selects the operating mode of the target MCU.
[Clock]	Selects the speed of the input clock of the target MCU. Select one of the followings: Emulator Clock (x MHz): Clock signal of the emulator (x: 4, 6, 8, 10, or 12.5) Target: Clock signal from the user system Xtal: Crystal oscillator on the evaluation chip board
[Clock Mode]	Sets the operating clock mode of the target MCU.
[JTAG Clock]	Sets the JTAG clock frequency.
[Timer Resolution]	Selects the resolution of the timer for use in execution time measurement and performance analysis. Select one of the following values. Runtime measurement: 52us, 1.6us, or 20ns Clock counter measurement: CLOCK, CLOCK/2, CLOCK/4, or CLOCK/8
[Enable interrupts during step execution]	When this box is checked, interrupts are accepted during step execution.
[Break on access error]	When this box is checked, a break (the user program stops) occurs if your program accesses a guarded memory area or writes to a write-protected area.
[Enable read and write on the fly]	Selects whether or not to enable an access to memory during user program execution.
[Enable internal ROM area write]	When this box is checked, writing to the on-chip ROM area is enabled.
[User VCC Threshold]	Sets the voltage level for the user system. [Down] will be displayed in [User PVcc] of the [Extended Monitor] window when the actual user Vcc of the target system is lower than the specified value.
[User Signals]	When this box is checked, the selected signal from the user system is enabled.
[Bus Timeout]	Select the bus timeout detection time. 100us, 1.6ms, 13ms, or 210ms can be selected.
[Enable multi break]	When this box is checked, the multibreak function is enabled. This allows breaking the program for several E6000H emulators at the same time by using a trigger input and probe pins.
[Driver]	Displays the E6000H driver that is currently installed.
[Change driver in start up]	When this box is checked, selection of a driver will be available next time the emulator is connected.

Note: The system clock (ϕ) is set as the input for the clock counter setting.

The debugging function in the E6000H emulator is realized by serial communication with the target MCU. The JTAG clock is used as the input clock in this serial communication. Set the JTAG clock frequency as high as possible to improve performance in downloading and reading memory. Note that, however, the JTAG clock frequency must be lower than that of the peripheral modules clock (Pϕ) of the target MCU.

RENESAS

3.1.2   Settings Associated with the Pin Function Controller

The [Pin Select Registers] page of the [Configuration Properties] dialog box allows setting of the pins selected by the pin function controller (PFC). The emulator correctly sets these pins that correspond to the signals for use in the following functions.

- External interrupt signal conditions for hardware breaks and tracing

- Detecting the WAIT signal for display in the emulation-state display

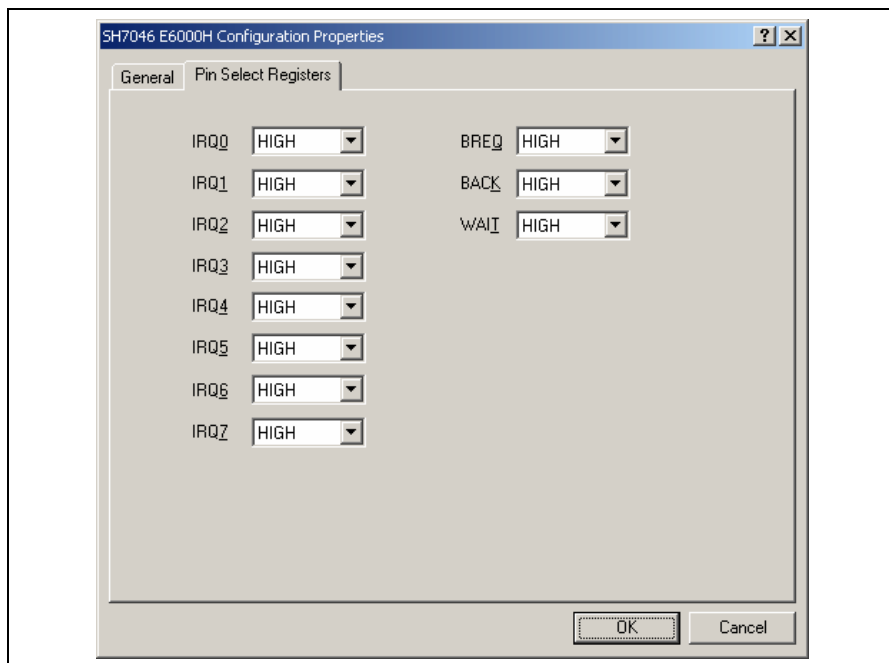Select the pins that have been set as the PFC by the user program.

**Figure 3.2   [Configuration] Dialog Box ([Pin Select Registers] Page)**

[Pin Select Registers] page

Option	Description
[IRQ0]	Sets the pins (PA2, PB2, PD16, HIGH) that correspond to the IRQ0 signal.
[IRQ1]	Sets the pins (PA5, PB3, PD17, HIGH) that correspond to the IRQ1 signal.
[IRQ2]	Sets the pins (PA8, PB4, PD18, HIGH) that correspond to the IRQ2 signal.
[IRQ3]	Sets the pins (PA9, PB5, PD19, HIGH) that correspond to the IRQ3 signal.
[IRQ4]	Sets the pins (PB6, PD20, HIGH) that correspond to the IRQ4 signal.
[IRQ5]	Sets the pins (PB7, PD21, HIGH) that correspond to the IRQ5 signal.
[IRQ6]	Sets the pins (PB8, PD22, HIGH) that correspond to the IRQ6 signal.
[IRQ7]	Sets the pins (PB9, PD23, HIGH) that correspond to the IRQ7 signal.
[BREQ]	Sets the pins (PA13, PA18, PB7, HIGH) that correspond to the BREQ signal.
[BACK]	Sets the pins (PA15, PA19, PB6, HIGH) that correspond to the BACK signal.
[WAIT]	Sets the pins (PA7, PA17, PB8, PE17, HIGH) that correspond to the WAIT signal.

Note: The initial values for all of these settings are HIGH. When signals are specified as HIGH, the emulator displays HIGH, regardless of the state of the signals. Note that this has no effect on the pin state of the target MCU.

3.1.3 Selecting the Interface to be Connected

Checking [Change driver in start up] on the [Configuration Properties] dialog box allows a selection of the driver next time the emulator is connected.
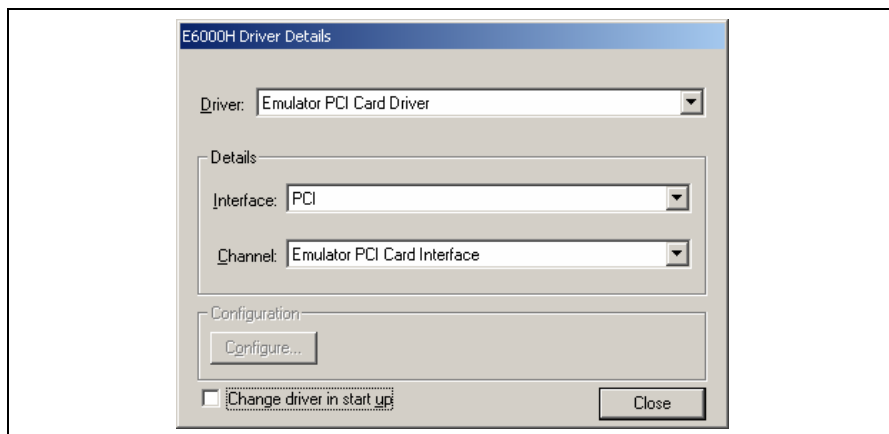
Figure 3.3 [Driver Details] Dialog Box

[Driver]: Selects the driver that connects the High-performance Embedded Workshop and the emulator.

[Details]: Sets the details of the driver being connected.

    [Interface]: The name of the interface to be connected. This should not be changed in this emulator.

   [Channel]: Channel for the selected interface. This should not be changed in this emulator.

[Configuration]: Driver setting.

    [Configure…]: A dialog box for setting will be displayed when the driver supports the configuration dialog. Note that this item is not available with this emulator.

[Change driver in start up]:
    Checking this box selects the driver when the emulator is connected the next time.

RENESAS

3.1.4 Opening the [Memory Mapping] Dialog Box

Selecting [Setup -> Emulator -> Memory Resource…] or clicking the [Emulator Memory Resource] toolbar button (🔳) opens the [Memory Mapping] dialog box.
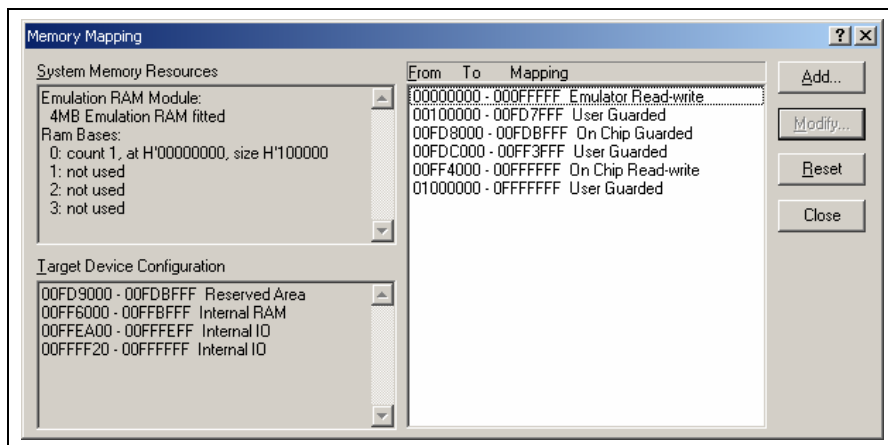
Figure 3.4 [Memory Mapping] Dialog Box

[Add...]:   Displays the [Edit Memory Mapping] dialog box, allowing the user to modify the address range and attributes of a memory map.

[Modify...]: Displays the [Edit Memory Mapping] dialog box, allowing the user to modify the address range and attributes of a memory map.

[Reset]:   Resets the map memory to its default settings.

[Close]:   Closes the dialog box.

The memory configuration of the target MCU being emulated is displayed by the [Memory] sheet in the [Status] window.

Note: For details, refer to section 5.12, Memory Map. The content displayed in the [Memory Mapping] dialog box differs depending on the product.

RENESAS

3.1.5   Changing the Memory Map Setting

Clicking the [Add…] button on the [Memory Mapping] dialog box or clicking the [Modify…] button after selecting the information on the memory map setting you want to change opens the [Edit Memory Mapping] dialog box.
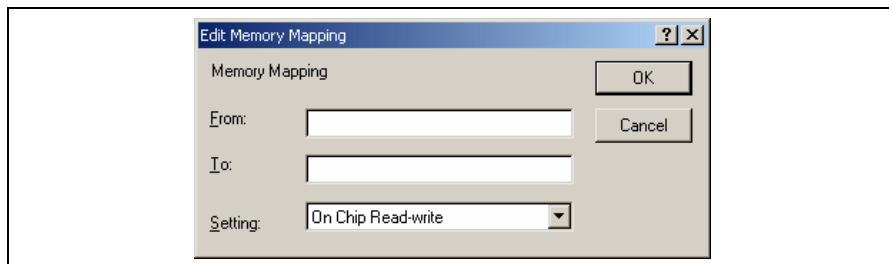
Figure 3.5   [Edit Memory Mapping] Dialog Box

Use this dialog box to change the address range and attributes of a memory map.

[From]:   Enter the start address of the map range.

[To]:   Enter the end address of the map range.

[Setting]:   Enter the memory map setting.
                     The choices given are listed below. The User (external memory) and Emulator (emulation memory) attributes can be modified.

- User Read-write: Selects an external area that is readable/writable

- User Read-only: Selects an external area that is write-protected

- User Guarded: Selects an external area that is access-prohibited

- Emulator Read-write: Sets readable/writable emulation memory as an external area

- Emulator Read-only: Sets write-protected emulation memory as an external area

- Emulator Guarded: Sets access-prohibited emulation memory as an external area

- No-access Area: Space that includes no memory

Emulation memory can be used as replacement of a ROM or flash memory on the user system. When the user program accesses to an access-prohibited area or writes to a write-protected area, a break occurs due to the illegal access. The user can set the [Configuration Properties] dialog box so that an illegal access will cause a break. Accesses to memory with the debugging function in use, as in the [Memory] window, are available regardless of the attribute that has been selected.

Note: The setting cannot be changed in the single-chip mode.

3.2   Downloading a Program

This section describes how to download a program and view it as source code or assembly-language mnemonics.

Note: After a break has occurred, the [Editor] window displays the location of the present program counter
        (PC). In most cases, for example if an Elf/Dwarf2-based project is moved from its original path, the
        source file may not be automatically found. In this case, a source file browser dialog box is displayed to
        allow you to manually locate the source file.

3.2.1   Downloading a Program

A load module to be debugged must be downloaded.

To download a program, select the load module from [Debug -> Download] or select [Download] from the
popup menu opened by clicking the right-hand mouse button on the load module in [Download modules] of the
[Workspace] window.

Note: Before downloading a program, it must be registered to the High-performance Embedded Workshop as a
        load module.

3.2.2   Viewing the Source Code

To view a source file's code, double-click on its icon in the file tree, or right-click on the source file and select
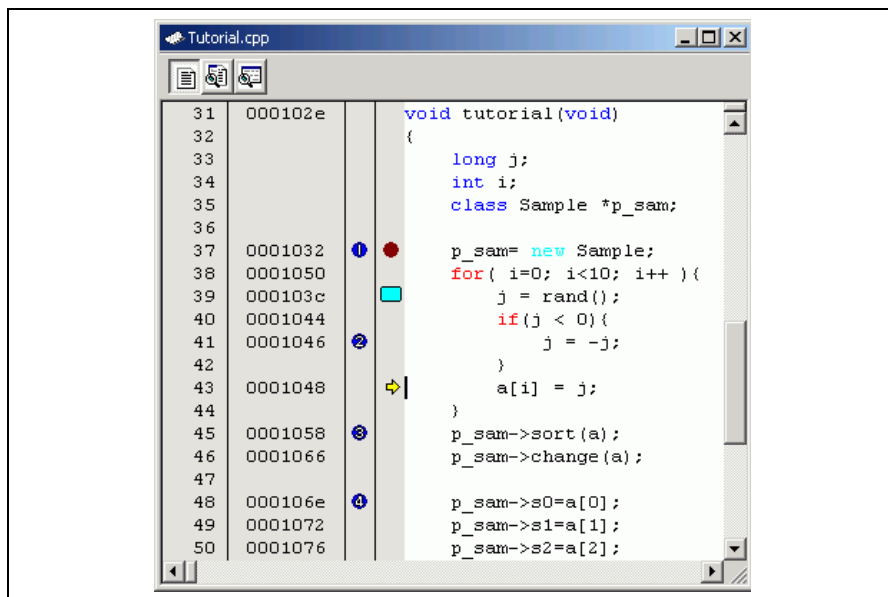the [Open] option on the pop-up menu. The [Editor] window is displayed.

Figure 3.6   Editor Window

RENESAS

In this window, the following items are shown on the left as information on lines.

- 1st column (Line Number column): A line number for the source file
- 2nd column (Source Address column): Address information for the source line
- 3rd column (On Chip Break column): On-chip breaks
- 4th column (S/W Breakpoints column): PC, bookmark, and breakpoint information

The text area is displayed in the right part of the [Editor] window.

Line Number column

This column displays the line number for the source file.

Source Address column

When a program is downloaded, an address for the current source file is displayed on the Source address column. These addresses are helpful when setting the PC value or a breakpoint.

On Chip Break column

The On Chip Break column displays the following items:

❶: On-chip break channel 1

❷: On-chip break channel 2

❸: On-chip break channel 3

❹: On-chip break channel 4

Ⓡ: On-chip break reset point

These are also set by using the popup menu.

S/W Breakpoints column

This column displays the following items:

▭: A bookmark is set.

●: A PC breakpoint is set.

➪: PC location

●   To switch off a column in all source files

1. Click the right-hand mouse button on the [Editor] window or select the [Edit] menu.
2. Click the [Define Column Format…] menu item.
3. The [Global Editor Column States] dialog box is displayed.
4. A check box indicates whether the column is enabled or not. If it is checked, the column is enabled. If the check box is gray, the column is enabled in some files and disabled in others. Deselect the check box of a column you want to switch off.
5. Click the [OK] button for the new column settings to take effect.
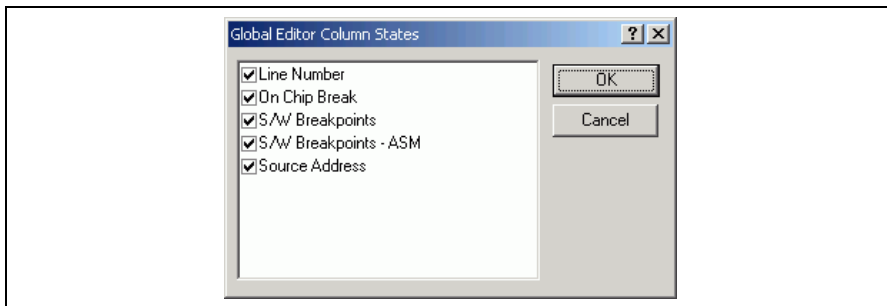
Figure 3.7   [Global Editor Column States] Dialog Box

● To switch off a column in one source file
1. Open the source file which contains the column you want to remove and click the [Edit] menu.
2. Click the [Columns] menu item to display a cascaded menu item. The columns are displayed in this popup menu. If a column is enabled, it has a tick mark next to its name. Clicking the entry will toggle whether the column is displayed or not.

RENESAS

3.2.3 Viewing the Assembly-Language Code

If you have a source file open, right-click to open the pop-up menu and select the [View Disassembly] option to open a Disassembly view at the same address as the current Source view.

It is also possible to view the disassembly using the new integrated [Disassembly view] in the source file.

If you do not have a source file, but wish to view code at assembly-language level, then select one of the following operations:

Click on the View Disassembly toolbar button (⊞).

Choose the [View -> Disassembly…] menu option.

Press Ctrl + D.

The [Disassembly] window opens at the current PC location.
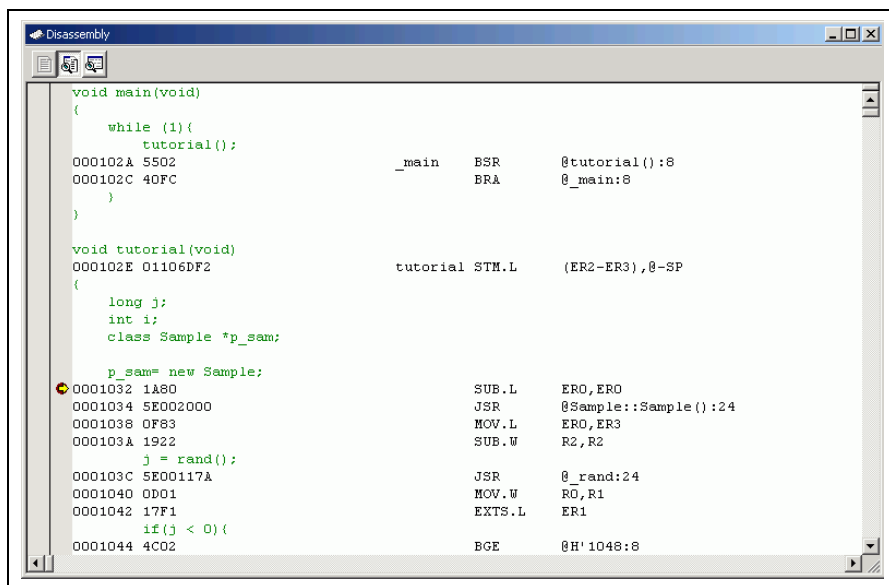
Figure 3.8 [Disassembly] Window

In this window, the following information is shown on the left as information lines.

- First column (On-chip break column): On-chip breaks

- Second column (S/W Breakpoints - ASM column): PC and breakpoint information

This window is used in the same way as the source code window.

3.2.4   Modifying the Assembly-Language Code

You can modify the assembly-language code by double-clicking on the instruction that you want to change. The [Assembler] dialog box will be opened.

Figure 3.9   [Assembler] Dialog Box

The address, instruction code, and mnemonic are displayed. Enter a new instruction or edit the old instruction in the [Mnemonics] field. Pressing the [Enter] key will replace the memory content with the new instruction and move on to the next instruction. Clicking the [OK] button will replace the memory content with the new instruction and close the dialog box. Clicking the [Cancel] button or pressing the [Esc] key will close the dialog box without modifying the memory contents.

Note: The assembly-language code being displayed is the current memory content. If the memory contents are changed the [Assembler] dialog box and the [Disassembly] window will show the new assembly-language code, but the source file displayed in the [Editor] window will be unchanged. This is the same even if the source file contains assembly codes.

3.2.5   Viewing a Specific Address

When you are viewing your program in the [Disassembly] window, you may wish to look at another area of your program's code. Rather than scrolling through a lot of code in the program, you can go directly to a specific address. Select [Set Address…] from the popup menu, and the dialog box shown in figure 3.10 is displayed.
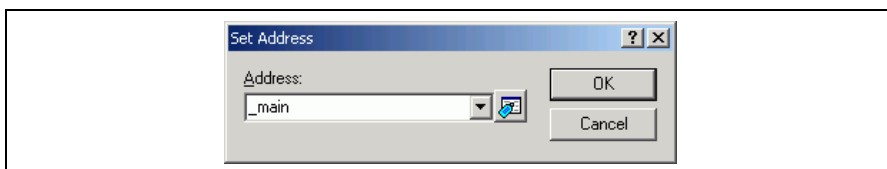
Figure 3.10   [Set Address] Dialog Box

Enter the address in the [Address] edit box and either click on the [OK] button or press the Enter key. A label name can also be specified as the address. When an overloaded function or a class name is entered, the [Select Function] dialog box opens for you to select a function.

3.2.6   Viewing the Current Program Counter Address

Wherever you can enter an address or value into the High-performance Embedded Workshop, you can also enter an expression. If you enter a register name prefixed by the hash character, the contents of that register will be used as the value in the expression. Therefore, if you enter the expression #pc in the [Set Address] dialog box, the [Editor] or [Disassembly] window will display the current PC address. It allows the offset of the current PC to be displayed by entering an expression with the PC register plus an offset, e.g., #PC+0x100.

RENESAS

3.3 Viewing the Current Status

Choose [View -> CPU -> Status] or click the [View Status] toolbar button ( ) to open the [Status] window and see the current status of the debugging platform.
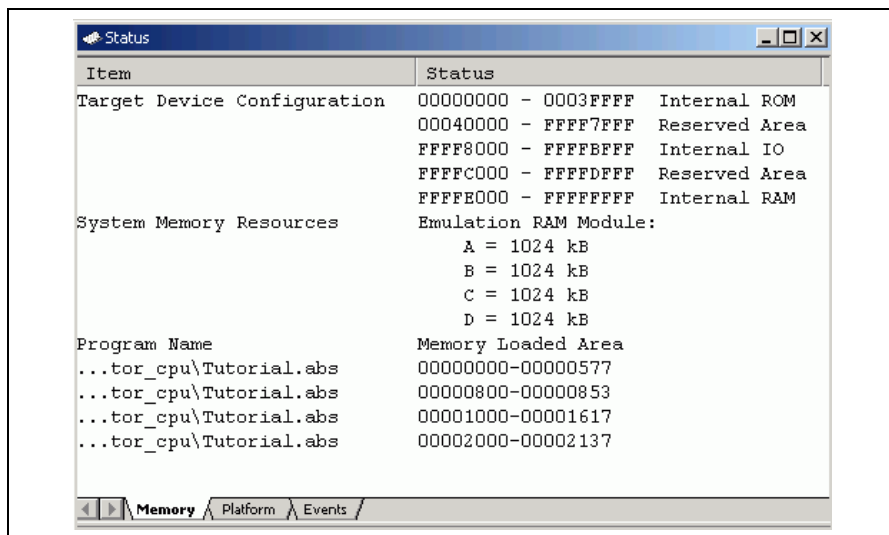
Figure 3.11 [Status] Window

The [Status] window has following three sheets:

- [Memory] sheet

  Displays information about the current memory status including the memory mapping resources and the areas used by the currently loaded object file.

- [Platform] sheet

  Displays information about the environment for emulation, typically including CPU type and emulation mode.

- [Events] sheet

  Displays information about the current event (breakpoint) status, including resource information.

Note: The items that can be set in this window depend on the emulator in use. For details, refer to the online help.

3.4   Reading and Displaying the Emulator Information Regularly

Use the [Extended Monitor] window to know the changing information on the emulator no matter the user program is running or halted.

Note: The extended monitor function does not affect the execution of the user program since it monitors the user system or the signal output from the target MCU in the emulator by using the emulator's hardware circuit.

3.4.1   Opening the [Extended Monitor] Window

Selecting [View -> CPU -> Extended Monitor] or clicking the [Extended Monitor] toolbar button ( ) displays this window. The interval of updating the display is approximately 1,000 ms during user program execution or 5,000 ms while breaking, respectively.

Item	Value
User RES	Inactive
User NMI	Active
User BREQ	Inactive
User WAIT	Inactive
User HSTBY	Inactive
User BACK	Inactive
User MRES	Inactive
User FWP	Inactive
User IRQ7-0	11111111
User VCC	Ignore
User PVCC	Ignore
User Target	Not Connected
Eva Chip Board	Attached
Coverage Board	Attached
Running status	Break = User Break
Trace Count	131072
Trace Timestamp OverFlow	None
Target Mode	Ignore
Target ClockMode	Ignore
Target Clock	Ignore
Pin Counts	80

Figure 3.12   [Extended Monitor] Window

RENESAS

3.4.2   Selecting Items to be Displayed

Selecting [Properties…] from the popup menu displays the [Extended Monitor Configuration] dialog box.
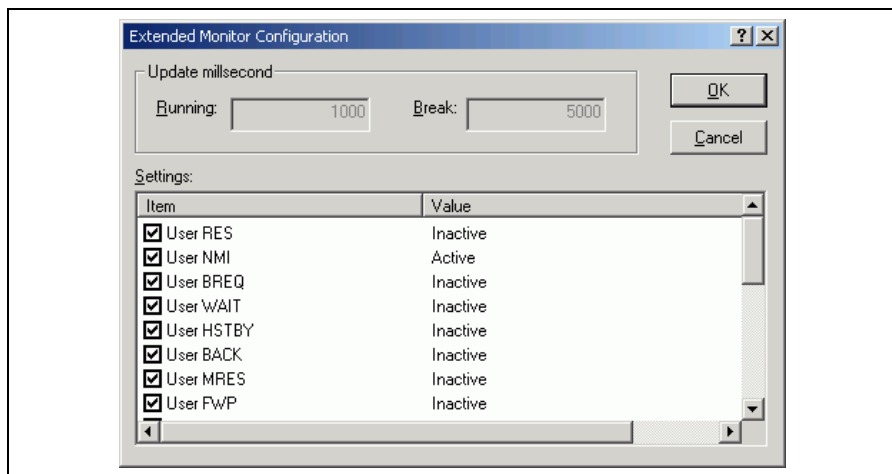
Figure 3.13   [Extended Monitor Configuration] Dialog Box

This dialog box allows the user to set the items to be displayed in the [Extended Monitor] window.

Note: The items that can be set in this window depend on the emulator in use. For details, refer to the online help.

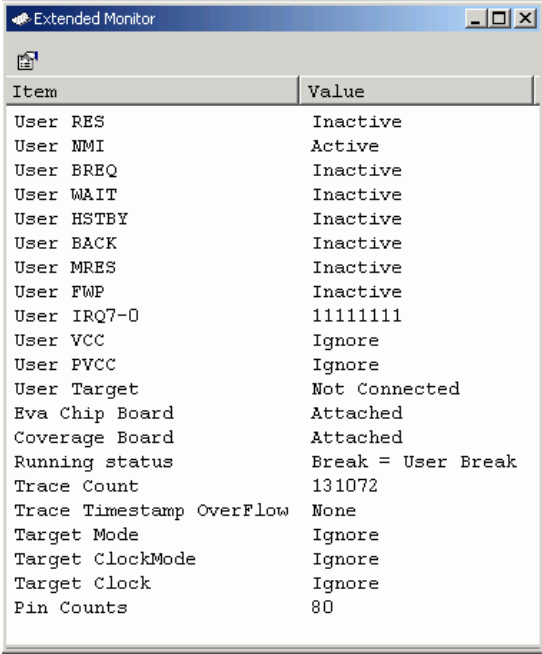## 3.5 Displaying Memory Contents in Realtime

Use the [Monitor] window to monitor the memory contents during user program execution. In the Monitor function, the realtime operation is retained since the bus monitoring circuit of the emulator sets the read/write signal of the MCU as a trigger and holds the address bus and data bus values to update the displayed contents of the memory.

Up to eight points or 256 bytes in total can be set by using the eight monitoring channels on the bus monitoring circuit. It is possible that a part or all of monitoring ranges is overlapped.

Notes: 1. Monitoring is impossible for an area, such as an on-chip timer counter, where no internal write signal is generated to update a value.
2. The procedure to display or modify the contents of memory differs depending on the product. If the display of memory contents is updated during execution of the user program, realtime emulation may not be available. For details, refer to section 5.3, Displaying and Modifying the Contents of Memory.

3.5.1 Opening the [Monitor] Window

To open the [Monitor] window, select [View -> CPU -> Monitor -> Monitor Setting...] or click the [Monitor] toolbar button ( ) to display the [Monitor Settings] dialog box.

Figure 3.14 [Monitor Setting] Dialog Box

[Name]:   Decides the name of the monitor window.

[Options]:   Sets monitor conditions.

    [Address]:   Sets the start address for monitoring.

    [Size]:   Sets the range for monitoring.

    [Access]:   Sets the access size to be displayed in the monitor window.

    [Auto-Refresh at rate]:   Sets the interval for acquisition by monitoring (500 ms at minimum).

    [Reading the Initial Value]: Selects reading of the values in the monitored area when the monitor window is opened.

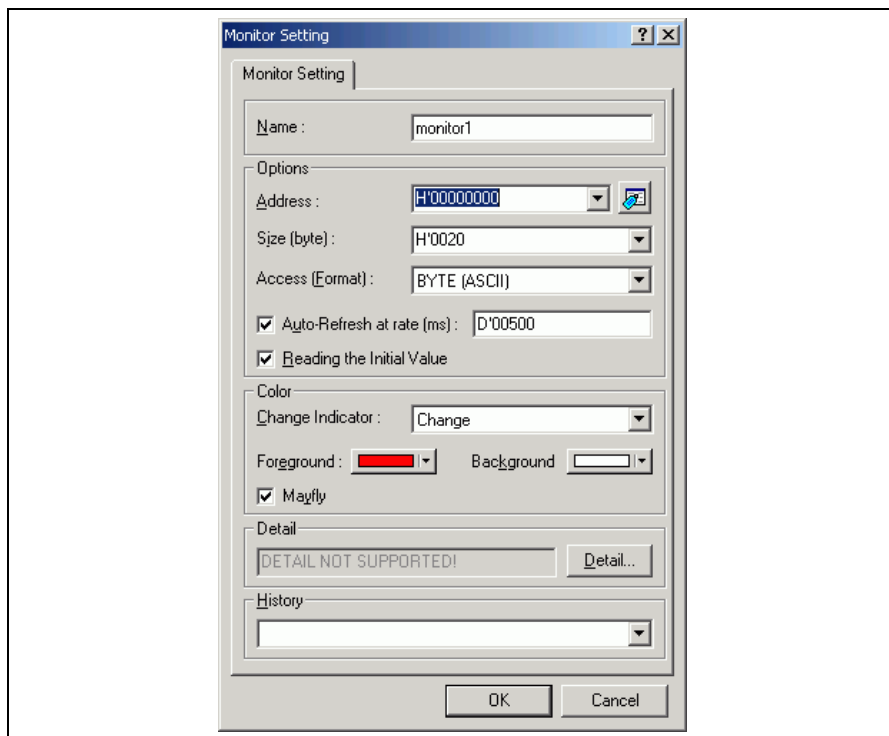[Color]:   Sets the method to update monitoring and the attribute of colors.

    [Change Indicator]:   Selects how to display the values that have changed during monitoring (available when [Reading the Initial Value] has been selected).

        No change: No color change.

        Change: Color is changed according to the [Foreground] and [Background] options.

        Gray: Those data with values that have not been changed are displayed in gray.

        Appear: A value is only displayed after changed.

    [Foreground]:   Sets the color used for display (available when [Change] has been selected).

    [Background]:   Sets the background color (available when [Change] has been selected).

    [Mayfly]:   A check in this box selects restoration of the color of those data which have not been updated in a specified interval to the color selected in the [Background] option. The specified interval is the interval for monitor acquisition (available when [Change], [Gray], or [Appear] has been selected).

[Detail]:   Sets the items specific to the emulator.

[History]:   Displays the previous settings.

Notes: 1. In this emulator, odd addressees cannot be specified as the start addresses for monitoring.
      2. Selection of the foreground or background color may not be available depending on the operating system in use.

After setting, clicking the [OK] button displays the [Monitor] window.
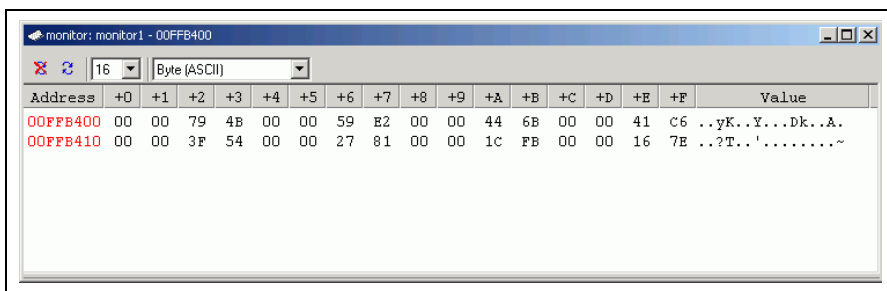
Figure 3.15 [Monitor] Window

During user program execution, the display is updated according to the setting value of the auto-update interval.

Note: Select [Refresh] from the popup menu when data is not displayed correctly after changing the address or content of memory.

3.5.2 Changing the Monitor Settings

Selecting [Monitor Setting…] from the popup menu of the [Monitor] window displays the [Monitor Setting] dialog box, which allows the settings to be changed.

Colors, the size of accesses, and the display format can be easily changed from [Color] or [Access] of the popup menu.

3.5.3 Temporarily Stopping Update of the Monitor

During user program execution, the display of the [Monitor] window is automatically updated according to the auto-update interval. Select [Lock Refresh] from the popup menu of the [Monitor] window to stop the update of display. The characters in the address section are displayed in black, and the update of display is stopped.

Selecting [Lock Refresh] again from the popup menu cancels the stopped state.

3.5.4 Deleting the Monitor Settings

Selecting [Close] from the popup menu of the [Monitor] window to be deleted closes the [Monitor] window and deletes the monitor settings.

3.5.5 Monitoring Variables

Using the [Watch] window refers to the value of any variables.

When the address of the variable registered in the [Watch] window exists within the monitoring range that has been set by the Monitor function, the value of the variable can be updated and displayed.

This function allows checking the content of a variable without affecting the realtime operation.

RENESAS

3.5.6 Hiding the [Monitor] Window

When using the Monitor function to monitor the value of a variable from the [Watch] window, hide the [Monitor] window for the effective use of the screen.

The current monitoring information is listed as the submenu when selecting [Display -> CPU -> Monitor]. The list consists of the [Monitor] window name and the address to start monitoring.

When the left of the list is checked, the [Monitor] window is being displayed.

Selecting items of the [Monitor] window you want to hide from the monitor setting list displays no [Monitor] window and removes the check mark at the left of the list.

To display the [Monitor] window again, select the hidden the [Monitor] window.
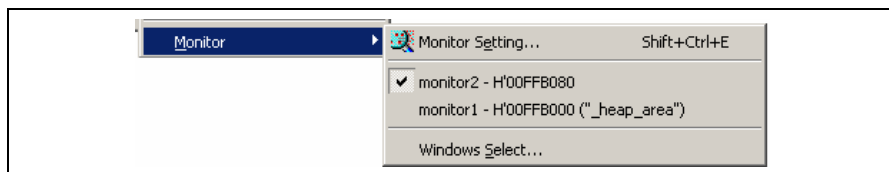
Figure 3.16   Monitor Setting List

3.5.7   Managing the [Monitor] Window

Selecting [Display -> CPU -> Monitor -> Windows Select…] displays the [Windows Select] dialog box. In this window, the current monitoring condition is checked and the new monitoring condition is added, edited, and deleted in succession.

Selecting multiple monitoring conditions enables a temporary stop of update, hiding, and deletion.
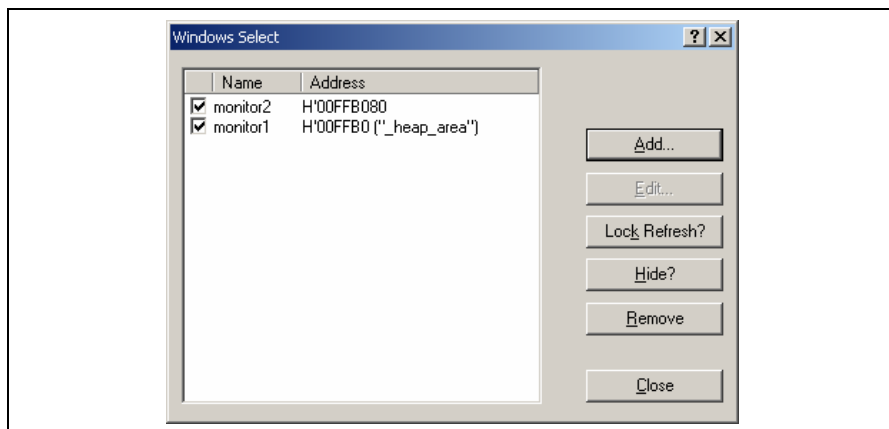
Figure 3.17   [Windows Select] Dialog Box

[Add]:   Adds a new monitoring condition.

[Edit]:   Changes the settings of the selected [Monitor] window (disabled when selecting multiple items).

[Lock Refresh/Unlock Refresh]:
                  Automatically updates or stops updating the display of the selected [Monitor] window.

[Hide/UnHide]:   Displays or hides the selected [Monitor] window.

[Remove]:   Removes the selected monitoring conditions.

[Close]:   Closes this dialog box.

RENESAS

3.6 Looking at Variables

This section describes how you can look at variables in the source program.

3.6.1 [Watch] Window

You can view any value in the [Watch] window.
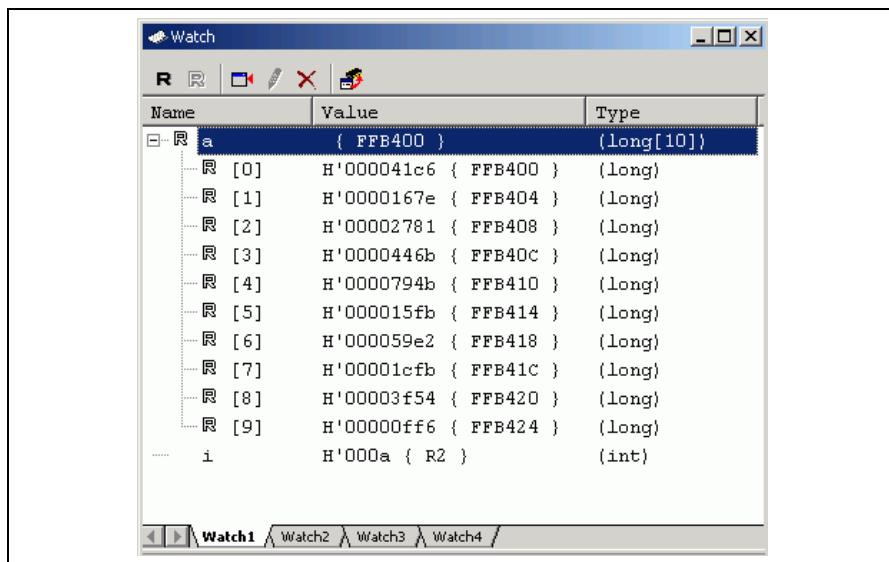
Figure 3.18 [Watch] Window

The [R] mark shows that the value of the variable can be updated during user program execution.

It is possible to recognize the method for updating the value during user program execution according to the color of the [R] mark.

Blue-outline [R]: The address of the variable is within the range that has been set for the monitoring function and the data is readable by using the monitoring function.

Blue [R]: An updated value of the data at this location has been read by the monitoring function.

Black-outline [R]: The address of the variable is outside the range that has been set for the monitoring function and the data is not readable by using the monitoring function.

Black [R]: A value has been updated by reading the normal data.

RENESAS

Notes: 1. This function can be set per variable or per element or body for structures of data.

2. The color of an [R] in the [Name] column changes according to the monitoring settings.

3. A variable that is allocated to a register cannot be selected for monitoring.

4. The procedure to display or modify the contents of memory differs depending on the product. If the display of memory contents is updated during execution of the user program, realtime emulation may not be available. For details, refer to section 5.3, Displaying and Modifying the Contents of Memory.

3.7 Using the Event Points

The emulator has the event point function to support breakpoints of the following three types.

Software breakpoints: Execution of the user program stops when the instruction at the specified address is fetched. Up to 255 software breakpoints can be set. Any content at the specified address is replaced by a break instruction (a dedicated instruction for use with the emulator). The software breakpoint cannot be set in the write-protected area such as ROM area or flash memory area on the user system. The user can set a software breakpoint in the [Editor] or [Disassembly] window.

On-chip breakpoints: These break functions built in the MCU. Conditions on the address bus, data bus, bus/area, and satisfaction count can be set. The on-chip breakpoint can be set even in the ROM area or flash memory area on the user system. It is also possible to set a sequential breakpoint consisted of several on-chip breakpoints. The user can set an on-chip breakpoint in the [Editor] or [Disassembly] window.

On-emulator breakpoints: On-emulator break functions are implemented by dedicated hardware in the E6000H station. Conditions on the address bus, data bus, bus/area, external probe signals, external interrupt signals, and satisfaction count can be set. As the emulator hardware provides this function, several cycles may be required until a break occurs after satisfaction of a condition.

Software, on-chip, and on-emulator breakpoints can be set in the [Event] window.

Select [View -> Code -> Eventpoints] or click the [Eventpoints] toolbar button (🔳) to open the [Event] window.
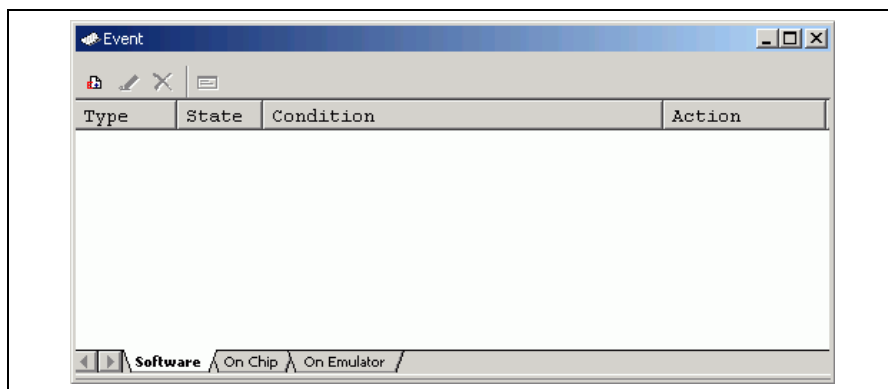
Figure 3.19   [Event] Window

The [Event] window has the following three sheets:

[Software] sheet: Displays the settings made for software breakpoints. It is also possible to set, modify, and cancel software breakpoints.

[On Chip] sheet: Displays or sets on-chip breakpoints.

[On Emulator] sheet: Displays or sets on-emulator breakpoints.

Note: For notes on event points, refer o section 5.5, Event Functions.

3.7.1   Setting a Software Breakpoint

Use the [Software] sheet on the [Event] window to display, change, or add settings for software breakpoints.

Select [Add...] or [Edit…] from the popup menu displayed on the [Software] sheet. The [Breakpoint Properties] dialog box (the [Software Break] page) will appear.

Figure 3.20   [Breakpoint Properties] Dialog Box ([Software Break] Page)

RENESAS

To add a new software breakpoint, select an empty line from the list box on the [Software Break] page and click the [Edit…] button. To change existing settings, select the software breakpoint that you want to change from the list box and click the [Edit…] button. The [Software Break] dialog box is displayed.

Figure 3.21   [Software] Dialog Box ([Address] Page)

Specify the breakpoint's address in the [Address] edit box and click the [OK] button.

3.7.2 Setting an On-Chip Breakpoint

Use the [On Chip] sheet on the [Event] window to display, change, or add settings for on-chip breakpoints.

Select [Add...] or [Edit…] from the popup menu displayed on the [On Chip] sheet. The [Breakpoint Properties] dialog box (the [On Chip Break] page) will appear.
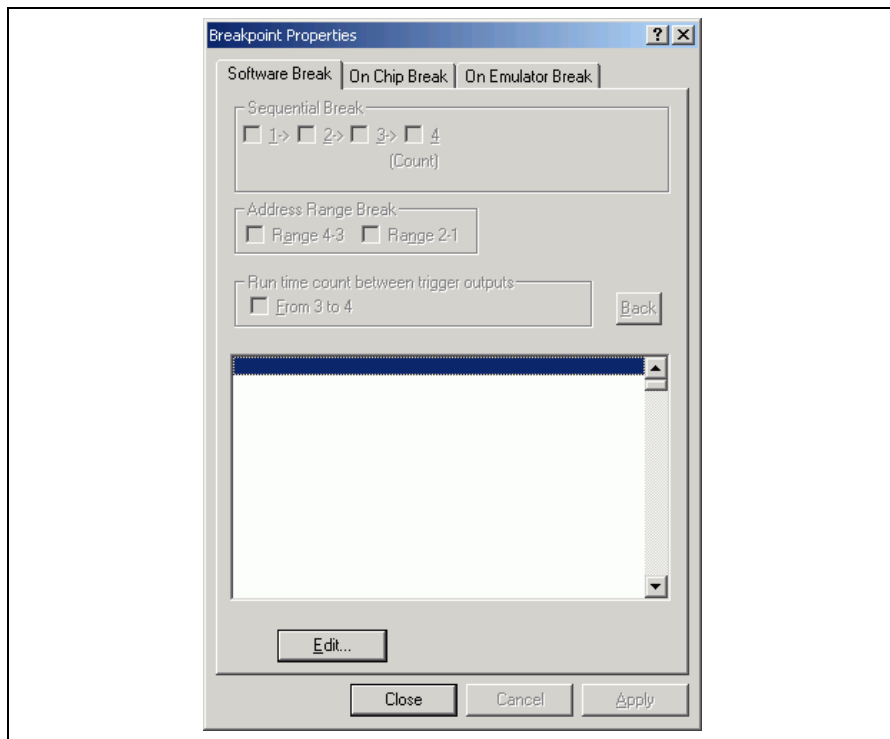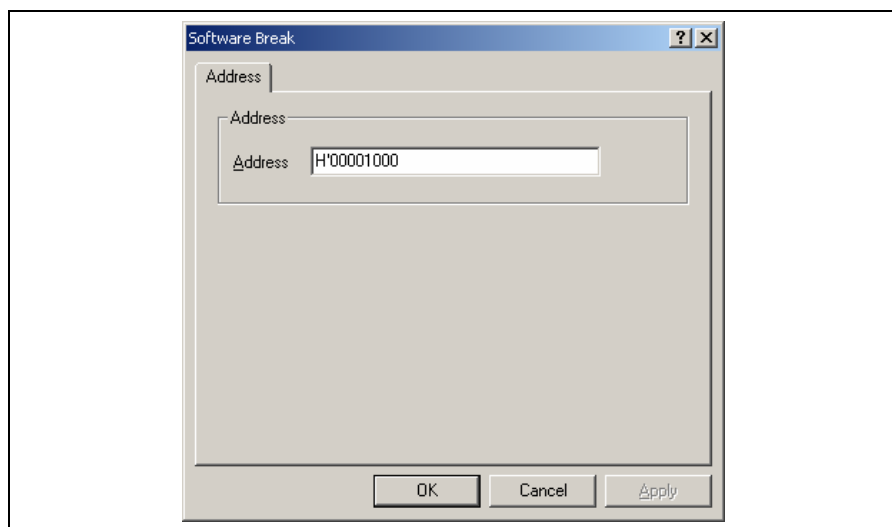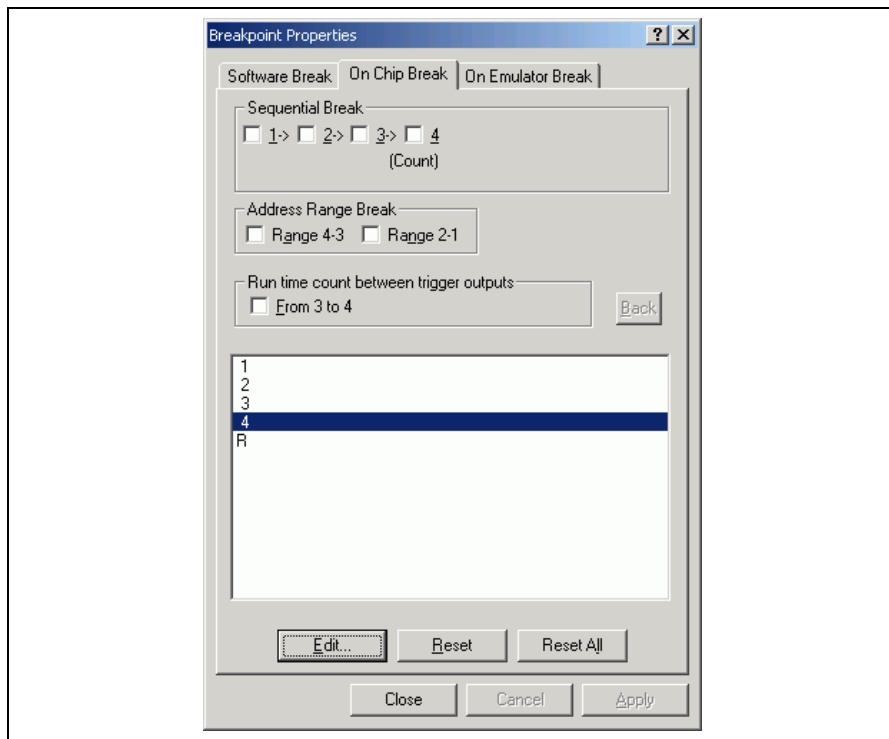

Figure 3.22   [Breakpoint Properties] Dialog Box ([On Chip Break] Page)

[Sequential Break]: Use this item to specify a sequential break consisted of on-chip breakpoints. A sequential break allows breaking the program when the conditions of several channels are satisfied sequentially. Two to four points can be selected for this function. The conditions are satisfied from 1 to 4, in order. Check a box to use the sequential break function (select 3 to use two points, 2 to use three points, and 1 to use four points).

[Address Range Break]: Selecting a combination of channels specifies the break range. Select either of the followings:
4-3: Channels 4 and 3 are used to specify the break range.
2-1: Channels 2 and 1 are used to specify the break range.

[Run time count between trigger outputs]:
Measures the time between two points by using channels 3 and 4. After channel 3 has been satisfied, the time is measured when channel 4 is satisfied. The result is displayed in [RunTime Count] on the [Platform] sheet of the [Status] window.

[Back]:	Puts the setting back to the previous state at the time the dialog box has been displayed.
List box:	Displays the current settings for each of the channels. R shown as the channel number indicates that a reset point is set for a sequential break. If no setting has been made for a channel, only the channel number is displayed here. When a channel is used for the sequential break function, S is displayed next to the channel number.
[Edit…]:	Clicking this button opens the [On Chip Break Channel n] dialog box (n: channel number), which allows the user to set a break condition for a selected channel.
[Reset]:	Clears the settings made for the selected channel.
[Reset All]:	Clears the settings made for all of the channels.

The user can set more complex break conditions in the [On Chip Break Channel n] dialog box by a combination of conditions provided on pages [Address], [Data], [Bus/Area], [Count], and [Action].
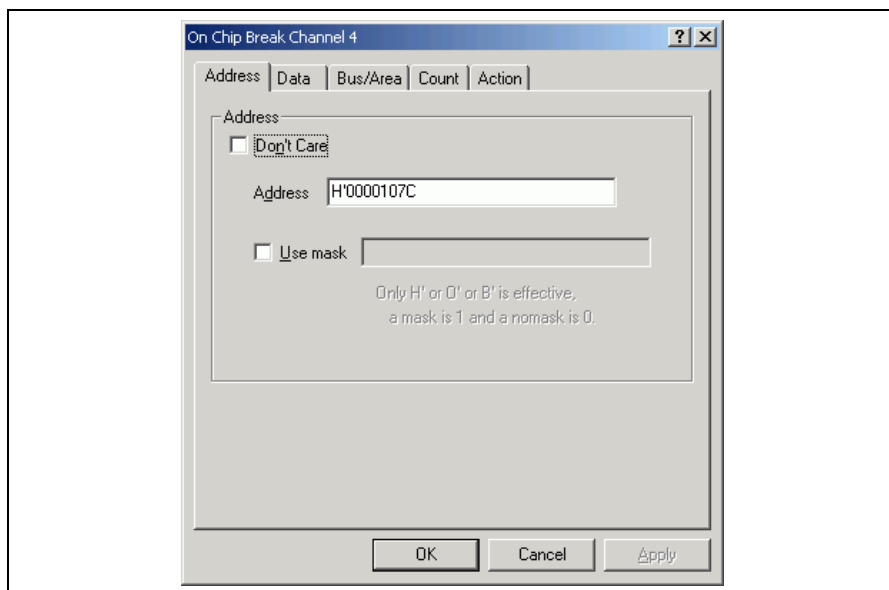
Figure 3.23   [On Chip Break Channel n] Dialog Box

RENESAS

[Address]:   Sets address bus conditions.

    [Don't Care]:   Selects no address bus condition.

    [Address]:   Sets an address bus value. It is also possible to specify a label as the address value.

    [Use mask]:   Sets mask conditions. Set the mask bits if [Use mask] is selected. Masked bits satisfy this break condition regardless of their values.

[Data]: Sets data bus conditions.

    [Don't Care]:   Selects no data bus condition.

    [Value]:   Sets a data bus value.

    [Use mask]:   Sets mask conditions. Set the mask bits if [Use mask] is selected. Masked bits satisfy this break condition regardless of their values.

    [Access Size]:   Selects the data-access size.

[Bus/Area]:   Sets access type, bus status, and read/write cycle conditions.

    [Access type]:   Sets access type conditions.

    [Bus State]:   Sets bus status conditions. When [Don't Care] has been selected, no bus status condition can be set.

    [Read/Write]:   Sets read/write conditions. When [Don't Care] has been selected, no read/write condition can be set.

[Count]:   Sets a satisfaction count of the condition. When [Don't Care] has been selected, the satisfaction count is defined as 1.

[Action]

    [Break]:   Halts execution when the selected condition has been satisfied.

        [After execution]:   Halts execution after the address at which the condition has been satisfied.

        [Before execution]:   Halts execution before the address at which the condition is satisfied.

    [Output Trigger]: Outputs a trigger when the selected condition has been satisfied.

3.7.3 Settings an On-Emulator Breakpoint

Use the [On Emulator] sheet on the [Event] window to display, change, or add settings for on-emulator breakpoints.

Select [Add...] or [Edit…] from the popup menu displayed on the [On Chip] sheet. The [Breakpoint Properties] dialog box (the [On Emulator Break] page) will appear.
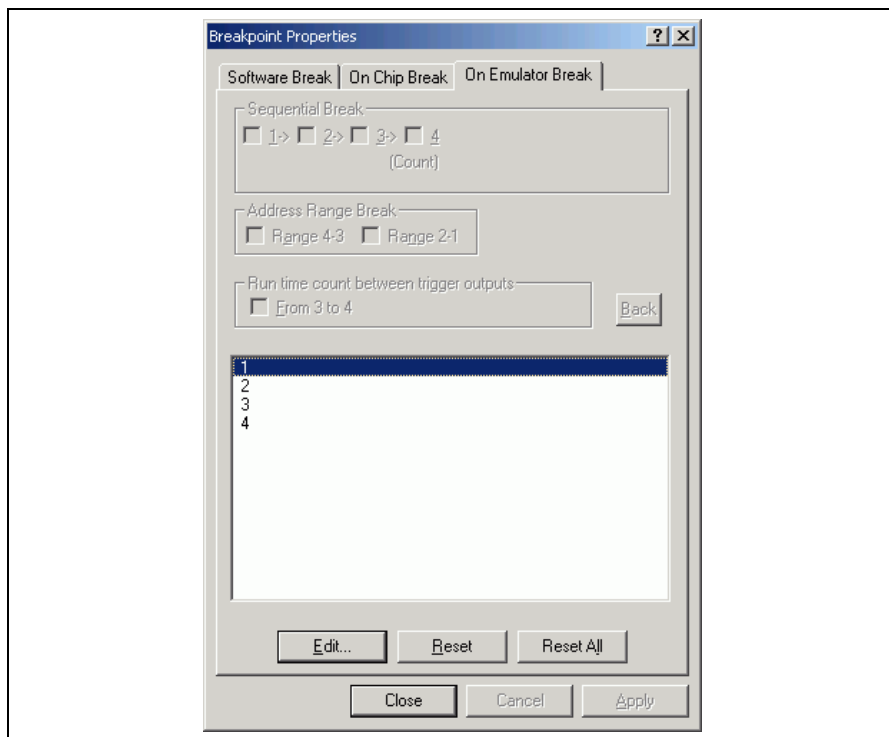
Figure 3.24　[Breakpoint Properties] Dialog Box ([On Emulator Break] Page)

List box:　　　　　　　　Displays the current settings for each of the channels. If no setting has been made for a channel, only the channel number is displayed here.

[Edit…]:　　　　　　　　Clicking this button opens the [On Emulator Break Channel n] dialog box (n: channel number), which allows the user to set a break condition for a selected channel.

[Reset]:　　　　　　　　Clears the settings made for the selected channel.

[Reset All]:　　　　　　Clears the settings made for all of the channels.

RENESAS

The user can set more complex break conditions in the [On Emulator Break Channel n] dialog box by a combination of conditions provided on pages [Address], [Data], [Bus/Area], [Probe], [Interrupt], and [Count].

Figure 3.25   [On Emulator Break Channel n] Dialog Box

RENESAS

[Address]:   Sets address conditions. It is also possible to specify a label as the address value.

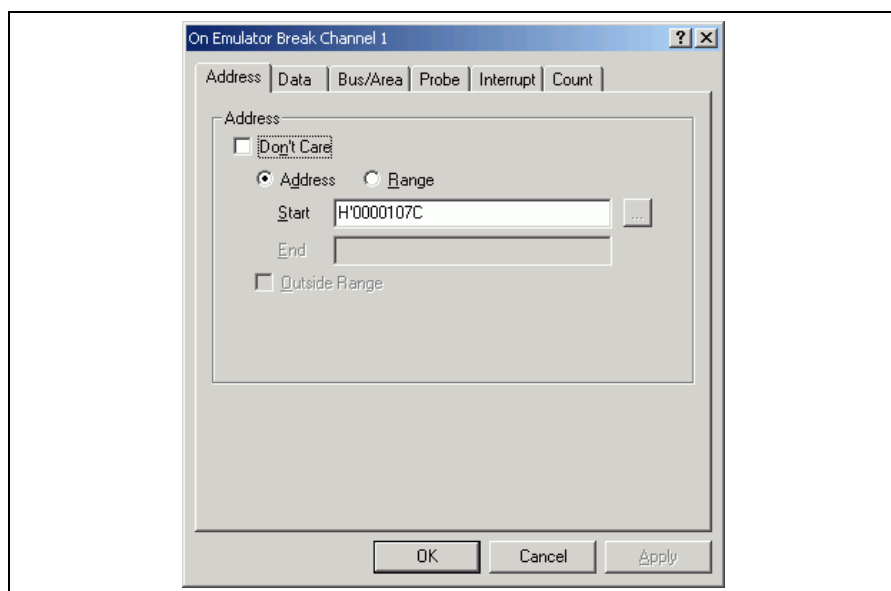   [Don't Care]:   Selects no address bus condition.

    [Address]:   Select this button to set the address bus value specified in [Start] as the break condition.

    [Range]:   A break occurs in the range of the address bus values specified from [Start] (start address) to [End] (end address).

    [Outside Range]:   Select this option to generate a break with an address bus outside the range set in [Range].

    […]:   The address range of a function can be set by [Start] and [End]. For details, refer to section 5.10, Input Format.

[Data]: Sets data conditions.

    [Don't Care]:   Selects no data bus condition.

    [Value]:   Sets a data bus value.

    [Use mask]:   Sets mask conditions. Set the mask bits if [Use mask] is selected. Masked bits satisfy this break condition regardless of their values.

    [Except this value]:   Sets a value other than that has been specified as the data bus condition.

    [Access Size]:   Selects the data-access size.

    [Position]:   Sets a data bus value as a number. The position of the valid data bus is specified.

          [Long]:   None

          [Word]:   4n: Upper word
                             4n + 2: Lower word

          [Byte]:   4n: Upper byte of the upper word
                             4n + 1: Lower byte of the upper word
                             4n + 2: Upper byte of the lower word
                             4n + 3: Lower byte of the lower word

[Bus/Area]:   Sets access type, bus status, and read/write cycle conditions.

    [Access type]:   Sets access type conditions. When [Don't Care] has been selected, no access type condition can be set.

    [Bus State]:   Sets bus status conditions. When [Don't Care] has been selected, no bus status condition can be set.

    [Area]:   Sets area conditions. When [Don't Care] has been selected, no area condition can be set.

        [On chip ROM]: Selects the on-chip ROM area as the condition.

        [On chip RAM]: Selects the on-chip RAM area as the condition.

        [On chip I/O]:   Selects the on-chip I/O area as the condition.

RENESAS

[External]:	Selects an external area where no emulation memory is allocated as the condition.	
[Emulator]:	Selects an external area where emulation memory is allocated as the condition.	

[Read/Write]: Sets read/write conditions. When [Don't Care] has been selected, no read/write condition can be set.

[Probe]: Sets the levels (high or low) of the external probe signals (PRB1 to PRB4) as the condition. When [Don't Care] has been selected, the level of the selected probe signal cannot be set as the condition.

[Interrupt]: Sets the levels (high or low) of the IRQ and NMI signals as the condition. When [Don't Care] has been selected, the level of the IRQ or NMI signal cannot be set as the condition.

[Count]: Sets a satisfaction count condition. When [Don't Care] has been selected, the satisfaction count is defined as 1.

3.7.4   Editing Event Points

Handlings for settings other than software breakpoints, on-chip breakpoints, and on-emulator breakpoints are common.

3.7.5   Modifying Event Points

Select an event point to be modified, and choose [Edit...] from the popup menu to open the dialog box that corresponds the event, which allows the user to modify the event conditions. The [Edit...] menu is only available when one event point is selected.

3.7.6   Enabling an Event Point

Select an event point and choose [Enable] from the popup menu to enable the selected event point.

3.7.7   Disabling an Event Point

Select an event point and choose [Disable] from the popup menu to disable the selected event point. When an event point is disabled, the event point will remain in the list, but an event will not occur when the specified conditions have been satisfied.

3.7.8   Deleting an Event Point

Select an event point and choose [Delete] from the popup menu to remove the selected event point. To retain the event point but not have it cause an event when its conditions are met, use the [Disable] option (see section 3.15.7, Disabling an Event Point).

RENESAS

3.7.9 Deleting All Event Points

Choose [Delete All] from the popup menu to remove all event points.

3.7.10 Viewing the Source Line for an Event Point

Select an event point and choose [Go to Source] from the popup menu to open the [Editor] or [Disassembly] window at the address of the event point. The [Go to Source] menu is only available when one event point that has the corresponding source file is selected.

3.8 Viewing the Trace Information

The emulator acquires the results of each instruction execution into the trace buffer as trace information and displays it in the [Trace] window. The conditions for the trace information acquisition can be specified in the [Trace Acquisition] dialog box.

Since trace information in bus-cycles is acquired by the hardware circuit and stored in the trace buffer, the realtime operation is retained. The [Trace] window displays the content of the trace buffer, which records up to 128-k bus cycles from the last program run and is always updated.

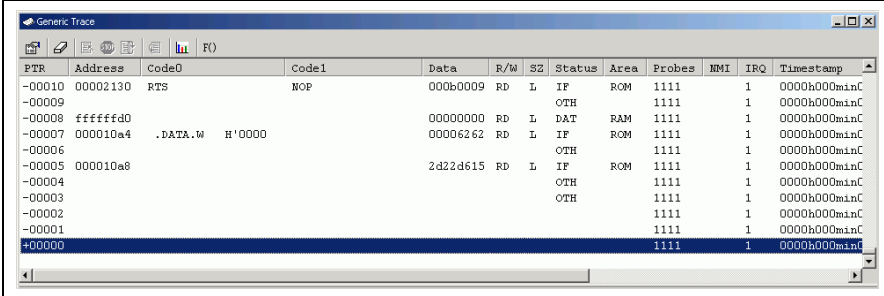Note: For notes on the trace functions, refer to section 5.6, Trace Functions.

3.8.1 Opening the [Trace] Window

To open the [Trace] window, choose [View -> Code -> Trace] or click the [Trace] toolbar button ( ).

3.8.2 Acquiring Trace Information

When the emulator does not set the acquisition condition of the trace information, all bus cycles are acquired by default without any condition (free trace mode).

In the free trace mode, trace acquisition is started with the execution of the user program and stopped by halting the user program. The acquired trace information is displayed in the [Trace] window.

Figure 3.26   [Trace] Window

This window displays the following trace information items:

[PTR]:	Cycle number in the trace buffer. When the most recent record is record 0, earlier record numbers go backwards (-1, -2, ...). If a delay count has been set, the cycle number where the trace stop condition has been satisfied is record 0. For the cycle (during delay) executed until the trace has stopped, earlier record numbers go forward (+1, +2, ...) the most recent record.
[Address]:	Address bus value (8-digit hexadecimal)
[Code0]:	Upper 16 bits of the executed instruction code
[Code1]:	Lower 16 bits of the executed instruction code
[Data]:	Data in byte, word, or longword units, displayed as 2-digit, 4-digit, or 8-digit hexadecimal

RENESAS

[R/W]:	Whether an access was read (RD) or write (WR)

[SZ]:   Selects the size of an access as B (byte), W (word), or L (longword).

[Status]:   Bus state; DMA (internal DMAC/DTC execution cycle), AUD (AUD cycle), BRL (user bus release), DAT (CPU data access cycle except for PC-relative data access), IF (CPU instruction fetch cycle including PC-relative data access), or OTH (other cases).

[Area]:   Types of access areas; U32, U16, and U08 (32-, 16-, and 8-bit external memory area, respectively), E32, E16, and E08 (32-, 16-, and 8-bit emulation memory area, respectively), ROM, I/O, and RAM (on-chip ROM, on-chip I/O, and on-chip RAM, respectively).

[Probes]:   A 4-bit binary number showing the four probe pins in the order of Probe 4, Probe 3, Probe 2, and Probe 1 from the left.

[NMI]:   Status of NMI input signal

[IRQ]:   Status of IRQ input signals

[Timestamp]:   Time stamp of the record. Time stamps start from zero each time the user program is executed. The minimum unit used in time measurement depends on the time stamp clock rate selected in the [Trace Acquisition] dialog box.

[Source]:   Source program of the executed instruction address

[Label]:   Label information of the address (if defined)

[Timestamp-Difference]: Difference from the timestamp value shown on the previous line

It is possible to hide any column not necessary in the [Trace] window. Selecting a column you want to hide from the popup menu displayed by clicking the right-hand mouse button on the header column hides that column. To display the hidden column, select the column from the said popup menu again.

RENESAS

3.8.3   Specifying Trace Acquisition Conditions

The capacity of the trace buffer is limited. When the buffer becomes full, the oldest trace information is overwritten. Setting the trace acquisition condition allows acquisition of useful trace information and effective use of the trace buffer.

There are the following types of trace acquisition conditions.

Free trace: Acquires trace information continuously from the start of the user program execution to the occurrence of a break (only when no trace acquisition condition is set).

Sequential trace stop: Specifies the order of trace acquisition conditions to be satisfied and stops trace acquisition when all of the conditions are satisfied. It is possible to set up to seven pass points and one reset point. No break will occur even when the trace acquisition stops.

Trace stop due to trace buffer overflow: Stops trace acquisition when the trace buffer in the emulator station overflows. No break will occur even when the trace acquisition stops.

Trace stop: Stops trace acquisition when the specified conditions are satisfied. In this mode, trace acquisition stops without stopping the user program execution. Up to 12 points can be set independently as trace stop conditions. No break will occur even when the trace acquisition stops.

Address range trace: Acquires trace information of instructions or operands accessed in the range (subroutine) between the start and end addresses. Note that, however, when the selected subroutine calls another subroutine, no trace information will be acquired from the called subroutine. Up to 12 points can be set independently as the address ranges.

Conditional trace: Only acquires trace information from the points where the specified conditions are satisfied. Up to 12 points can be set independently as the conditions.

Address range conditional trace: Accesses instructions or operands in the range (subroutine) between the start and end addresses and only acquires trace information in the bus cycles that satisfy the conditions. This mode is a combination of address range trace and conditional trace. Up to six points can be set independently as the address ranges with conditions.

Point to Point trace: Acquires trace information from the satisfaction of the address condition set as a start condition to that of the address condition set as an end condition.

Execution time measurement: Measures execution time between two points by using the trace acquisition conditions.

Trigger output: Outputs a pulse from trigger pins when the specified conditions are satisfied.

The trace acquisition condition is set in the [Trace Acquisition] dialog box that is displayed by selecting [Acquisition…] from the popup menu.

The [Trace Acquisition Properties] dialog box has the pages [Condition] and [Other].
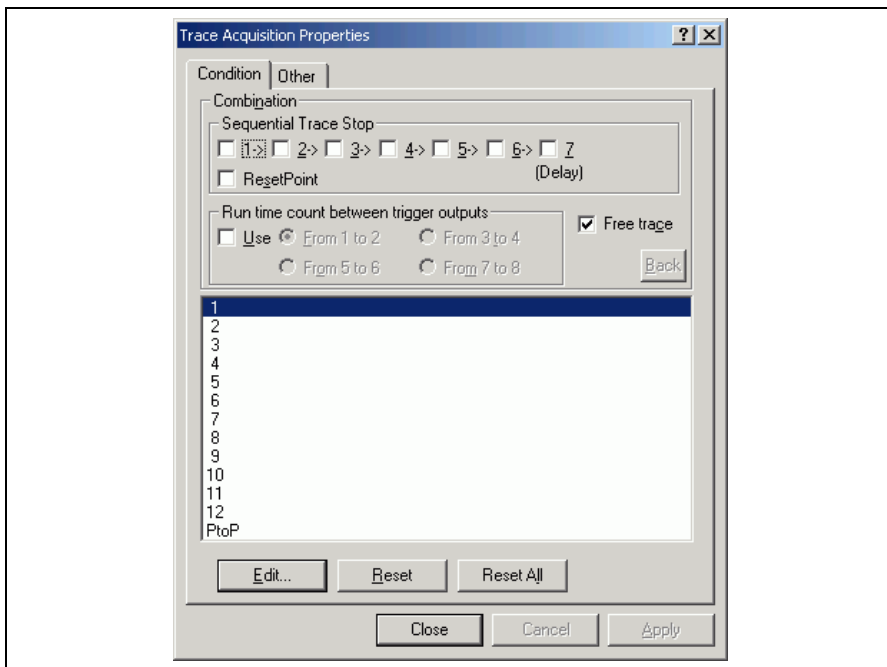
(1) [Condition] page

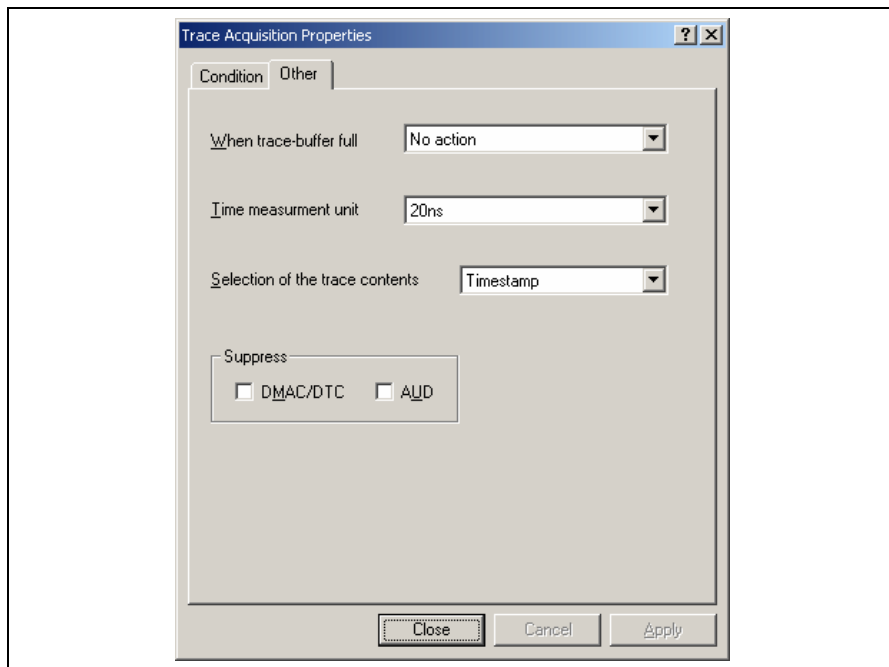**Figure 3.27   [Trace Acquisition Properties] Dialog Box ([Condition] Page)**

[Sequential Trace Stop]:   Use this option to set a sequential trace stop by using channels 1 to 7. The sequential trace stop function allows trace acquisition to stop when the conditions of several channels are satisfied in the specified order. Two to seven pass points and one reset point are selectable as sequential trace stop conditions. The conditions are satisfied in the order of 1 to 7. To use a sequential trace stop, select the checkbox of the channel. To set a reset condition, select the [Reset Point] checkbox. Channel 8 is used for a reset condition. When a reset condition is satisfied, all the sequential trace stop conditions that have been satisfied will be cleared and the emulator starts checking the first condition again. When a sequential trace stop is enabled, no setting is available for the channels (out of 1 to 7) that are not used for the sequential trace stop function.

[Run time count between trigger outputs]:
                          Selects channels for use in execution time measurement. Clicking [Use] allows measurement of time in tracing. There are four types of channel combinations consisted of those for the start and the end of measurement: 1-2, 3-4, 5-6, and 7-8.

[Free trace]:   Selects the free trace mode. When [Free trace] is enabled, any trace acquisition condition set will be ignored.

[Back]:   Puts the setting back to the previous state at the time the dialog box has been displayed.

51

List box:

Displays the current settings for each of the channels. If no setting has been made for a channel, only the channel number is displayed here. When a channel is used for the sequential trace stop function, S is displayed next to the channel number. When a reset condition for a sequential trace stop is enabled, R is displayed next to channel 8. PtoP is for use in the Point to Point trace. UNUSED is displayed next to the channel number if that channel is not available.

[Edit…]:

Clicking this button opens the [Trace Acquisition Condition Channel n] dialog box (n: channel number or PtoP), which allows the user to set a break condition for a selected channel.

[Reset]:

Clears the settings made for the selected channel.

[Reset All]:

Clears the settings made for all of the channels.

RENESAS

(2) [Other] page

Figure 3.28 [Trace Acquisition] Dialog Box ([Other] Page)

[When trace-buffer full]:   Selects an action to take when the trace buffer becomes full.

[No action]:   Overwrites the oldest information in the trace buffer.

[Stop trace]:   Stops trace acquisition without stopping the user program execution.

[Stop execution and trace]: Stops the user program execution.

[Time measurement unit]: Selects the minimum time unit for the time stamping of the bus trace information.

[52us]:   Time stamping is in minimum time units of 52 μs.

[1.6us]:   Time stamping is in minimum time units of 1.6 μs.

[20ns]:   Time stamping is in minimum time units of 20 ns.

[Clock]:   Time stamping is in terms of the number of bus-clock cycles, i.e., is synchronized with the cycles of the system clock signal (ϕ).

[Clock/2]:   Time stamping is in terms of the number of bus-clock cycles, i.e., is synchronized with 1/2 cycle of the system clock signal (ϕ).

[Clock/4]:   Time stamping is in terms of the number of bus-clock cycles, i.e., is synchronized with 1/4 cycle of the system clock signal (ϕ).

RENESAS

[Clock/8]: Time stamping is in terms of the number of bus-clock cycles, i.e., is synchronized with 1/8 cycle of the system clock signal (ϕ).

[Selection of the trace contents]:
Selects whether or not to display time stamps, IRQs, or executed instructions.

[Time stamp]: Displays IRQs if the level of any of IRQ7-0 is low. This allows any 32-bit time stamp information to be displayed.

[IRQ7-0 all indications]: Displays the status of IRQ7-0 in bit units. Lower 16 bits of time stamps are fixed to 0.

[Suppress]: Acquires no DMAC/DTC cycles or AUD cycles.

RENESAS

(3) [Trace Acquisition Condition Channel n] dialog box

Use this dialog box to set pass points and a reset point for a sequential trace stop, and conditions in the address range trace, address range conditional trace, conditional trace, Point to Point trace, execution time measurement, and a trigger output.
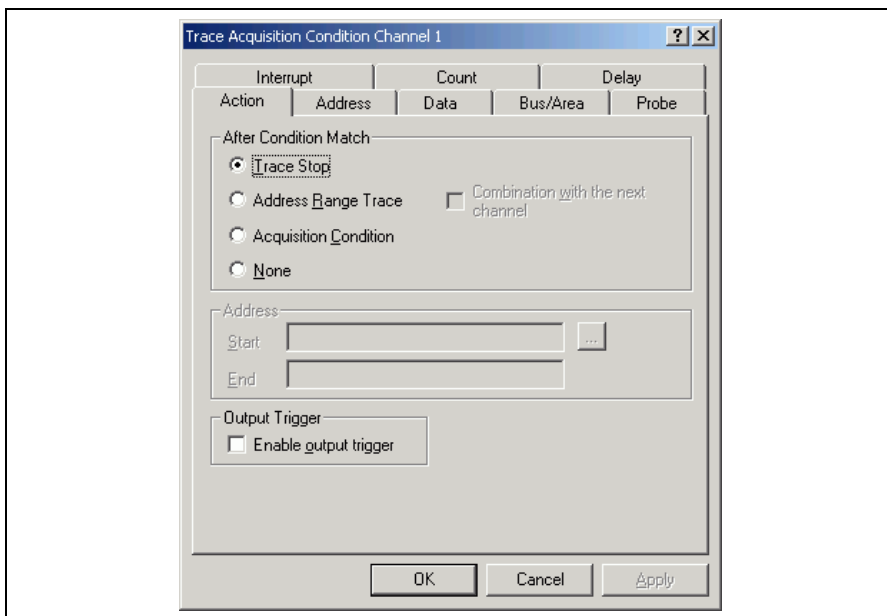
Figure 3.29   [Trace Acquisition Condition Channel n] Dialog Box

The [Trace Acquisition Condition Channel n] dialog box has pages [Action], [Address], [Data], [Bus/Area], [Probe], [Interrupt], [Count], and [Delay]. The user can make more complex settings by a combination of conditions provided on these pages.

[Action]

[After Condition Match]:   Selects an action to take when a condition is satisfied.

[Trace Stop]:   Selects a trace stop.

[Address Range Trace]:   Selects an address range trace. Selecting this option and checking [Combination with the next channel] allows an address range conditional trace.

[Acquisition Condition]:   Selects a conditional trace.

[None]:   Select this option if you do not want to take any of the actions listed above. This is useful for a trigger output or execution time measurement.

[Address]:   Sets the start and end addresses of the range in the address range trace, address range conditional trace, or Point to Point trace.

RENESAS

[Start]:   Set the start address.

[End]:   Set the end address.

[f()...]:   The address range of a function can be set by [Start] and [End].

[Output Trigger]:   Outputs a trigger after the satisfaction of a trace condition.

[Address]:   Sets address conditions.

[Data]:   Sets data conditions.

[Bus/Area]:   Sets access type, bus status, and read/write cycle conditions.

[Probe]:   Sets the levels (high or low) of the external probe signals (PRB1 to PRB4) as the condition.

[Interrupt]:   Sets the levels (high or low) of the IRQ and NMI signals as the condition.

[Count]:   Sets a satisfaction count condition.

[Delay]:   Sets the number of bus cycles delayed after the satisfaction of a trace condition. This function allows you to check the trace information before/after any of the specified conditions are satisfied. When [Don't Care] has been selected, there is no delay.

Notes: 1. The settings to be made on pages [Address], [Data], [Bus/Area], [Probe], [Interrupt], and [Count] are the same as those for on-emulator break conditions. For details on the on-emulator break conditions, refer to section 3.7, Using the Event Points.

   2. Set the range in the address range trace so that value of the end address will be larger than that of the start address.

   3. Two channels are used in the address range conditional trace. To perform the address range conditional trace, select an odd-numbered channel (2n + 1) for the address range trace and an even-numbered channel (2n) for the conditional trace, respectively, and then check [Combination with the next channel].

3.8.4 Searching for a Trace Record

Use the [Trace Find] dialog box to search for a trace record. To open this dialog box, choose [Find...] from the popup menu.
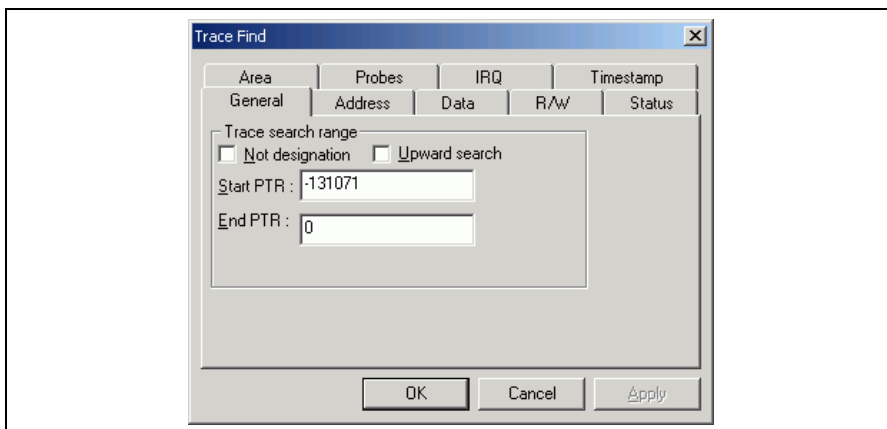
Figure 3.30 [Trace Find] Dialog Box

The [Trace Find] dialog box has the following options:

[General]: Sets the range for searching.

    [Not designation]: Searches for information that does not match the conditions set in other pages when this box is checked.

    [Upward search]: Searches upwards when this box is checked.

    [Start PTR]: Enters a PTR value to start a search.

    [End PTR]: Enters a PTR value to end a search.

[Address]: Sets an address condition.

    [Don't care]: Detects no address when this box is checked.

    [Value]: Enter the address value (not available when [Don't care] has been checked).

[Data]: Sets a data condition.

    [Don't care]: Detects no data when this box is checked.

    [Value]: Enter the data value (not available when [Don't care] has been checked).

[R/W]: Selects the type of access cycles.

    [Don't care]: Detects no read/write condition when this box is checked.

    [Setting]: Detects the specified read/write condition.
                             RD: Read cycle
                             WR: Write cycle

RENESAS

[Status]:   Selects the status of a bus as the condition.

    [Don't care]:   Detects no bus condition when this box is checked.

    [Setting]:   Detects the specified bus condition.

[Area]:   Selects the area being accessed.

    [Don't care]:   Detects no area condition when this box is checked.

    [Setting]:   Detects the specified area condition.

[Probes]:   Sets a probe signal condition.

    [Don't care]:   Detects no probe signal condition when this box is checked.

    [Setting]:   Detects the specified probe signal condition.
                         Don't care: Detects no selected probe signal condition.
                         High: The status of the probe signal is high.
                         Low: The status of the probe signal is low.

[IRQ]:   Sets an IRQ signal condition.

    [Don't care]:   Detects no IRQ signal condition when this box is checked.

    [Setting]:   Detects the specified IRQ signal condition.
                         Don't care: Detects no selected IRQ signal condition.
                         High: The status of the IRQ signal is high.
                         Low: The status of the IRQ signal is low.

[Timestamp]:   Sets the time stamp condition.

    [Don't care]:   Detects no time stamp value when this box is checked.

    [Value]:   Enter the time stamp value.
                         The format is as follows:
                         hour: h, minute: min, second: s, millisecond: ms, microsecond: us, nanosecond: ns

Clicking the [OK] button after setting conditions in those pages stores the settings and starts searching. Clicking the [Cancel] button closes this dialog box without setting of conditions.

When a trace record that matches the search conditions is found, the line for the trace record will be highlighted. When no matching trace record is found, a message dialog box will appear.

Only the trace information that satisfies all the conditions set in above pages will be searched.

If a find operation is successful, selecting [Find Next] from the popup menu will move to the next found item.

3.8.5   Clearing the Trace Information

Select [Clear] from the popup menu to empty the trace buffer that stores the trace information. If several [Trace] windows are open, all [Trace] windows will be cleared as they all access the same buffer.

RENESAS

3.8.6   Saving the Trace Information in a File

Select [Save...] from the popup menu to open the [Save As] file dialog box, which allows the user to save the information displayed in the [Trace] window as a text file. A range can be specified based on the [PTR] number (saving the complete buffer may take several minutes). Note that this file cannot be reloaded into the [Trace] window.

Note: In filtering of trace information, the range to be saved cannot be selected. All the trace information displayed in the [Trace] window after filtering will be saved. Select a filtering range on the [General] page in the [Trace Filter] dialog box if you want to save the selected range. For details on the filtering function, refer to section 3.8.11, Extracting Records from the Acquired Information.

3.8.7   Viewing the [Editor] Window

The [Editor] window corresponding to the selected trace record can be displayed in the following two ways:

- Select a trace record and choose [View Source] from the popup menu.
- Double-click a trace record

The [Editor] or [Disassembly] window opens and the selected line is marked with a cursor.

3.8.8   Trimming the Source

Choose [Trim Source] from the popup menu to remove the white space from the left side of the source.

When the white space is removed, a check mark is shown to the left of the [Trim Source] menu. To restore the white space, choose [Trim Source] while the check mark is shown.

3.8.9   Temporarily Stopping Trace Acquisition

To temporarily stop trace acquisition during execution of the user program, select [Halt] from the popup menu. This stops trace acquisition and updates the trace display. Use this method to check the trace information without stopping execution of the user program.

3.8.10   Restarting Trace Acquisition

To restart trace acquisition being stopped during execution of the user program, select [Restart] from the popup menu.

3.8.11 Extracting Records from the Acquired Information

Use the filtering function to extract the records you need from the acquired trace information. The filtering function allows the trace information acquired by hardware to be filtered by software. Unlike the settings made in the [Trace Acquisition] dialog box for acquiring trace information by conditions, changing the settings for filtering several times to filter the acquired trace information allows easy extraction of necessary information, which is useful for analysis of data. The content of the trace buffer will not be changed even when the filtering function is used. Acquiring useful information as much as possible by the [Trace Acquisition] settings improves the efficiency in analysis of data because the capacity of the trace buffer is limited.

Use the filtering function in the [Trace Filter] dialog box to select a range for filtering.
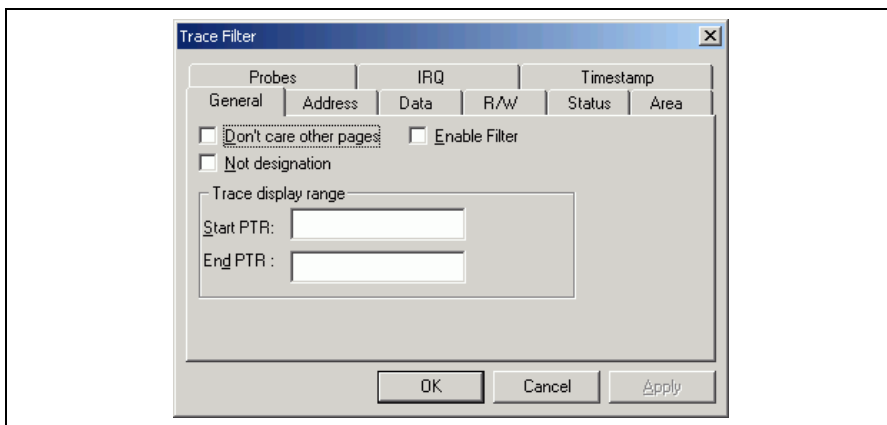
Figure 3.31 [Trace Filter] Dialog Box ([General] Page)

To open the [Trace Filter] dialog box, select [Filter…] from the popup menu.

The [Trace Filter] dialog box has the following pages:

[General]: Sets the range for filtering.

    [Don't care other pages]: Only selects the cycle number when this box is checked. Other options become invalid.

    [Enable Filter]: Enables the filter when this box is checked.

    [Not designation]: Filters information that does not match the conditions set in those pages when this box is checked.

    [Trace display range]: Sets the range for filtering.

        [Start PTR]: Enters a PTR value to start filtering.

        [End PTR]: Enters a PTR value to end filtering.

[Address]: Sets address conditions.

    [Don't care]: Detects no address when this box is checked.

    [Setting]: Detects the specified address.

	[Point]:	Specifies a single address.
	[Range]:	Specifies an address range.
	[From]:	Enter a single address value or the start value of the address range.
	[To]:	Enter the end value of the address range.

[Data]: Sets data conditions.

[Don't care]:		Detects no data when this box is checked.
[Setting]:		Detects the specified data.
	[Point]:	Specifies single data.
	[Range]:	Specifies a data range.
	[From]:	Enter a single value or the minimum value of the data range.
	[To]:	Enter the maximum value of the data range.

[R/W]: Selects the type of access cycles.

[Don't care]:		Detects no read/write condition when this box is checked.
[Setting]:		Detects the specified read/write condition.
	[RD]:	Detects read cycles when this box is checked.
	[WR]:	Detects write cycles when this box is checked.

[Status]: Selects the status of a bus as the condition (multiple conditions can be selected here).

[Don't care]: Detects no bus condition when this box is checked.

[Setting]: Detects the specified bus condition.

[Area]: Selects the area being accessed.

[Don't care]: Detects no area condition when this box is checked.

[Setting]: Detects the specified area condition (not available when [Don't care] has been checked).

[Probes]: Sets a probe signal condition (multiple conditions can be selected here).

[Don't care]: Detects no probe signal condition when this box is checked.

[Setting]: Detects the specified probe signal condition.
Don't care: Detects no selected probe signal condition.
High: The status of the probe signal is high.
Low: The status of the probe signal is low.

[IRQ]: Sets an IRQ signal condition.

[Don't care]: Detects no IRQ signal condition when this box is checked.

[Setting]: Detects the specified IRQ signal condition.
Don't care: Detects no selected IRQ signal condition.

RENESAS

High: The status of the IRQ signal is high.
Low: The status of the IRQ signal is low.

[Timestamp]: Sets the time stamp condition.

[Don't care]: Detects no time stamp value when this box is checked.

[Setting]: Detects the specified time stamp value.

[Point]: Specifies a single time stamp.

[Range]: Specifies a time stamp range.

[From]: Enter a single time stamp value or the minimum value of the time stamp range.
The format is as follows:
hour: h, minute: min, second: s, millisecond: ms, microsecond: us, nanosecond: ns

[To]: Enter the maximum value of the time stamp range.
The format is the same as that in [From].

Set filtering conditions and then press the [OK] button. This starts filtering according to the conditions. Clicking the [Cancel] button closes the [Trace Filter] dialog box, which holds the settings at the time when the dialog box was opened.

In filtering, only the trace information that satisfies one or more filtering conditions set in the above pages will be displayed in the [Trace] window.

Filtering conditions can be changed several times to analyze data because the content of the trace buffer is not changed by filtering.

RENESAS

3.8.12 Calculating the Difference in Time Stamping

Select [Timestamp Difference…] from the popup menu to calculate the time difference between the two points selected by the result of tracing in acquisition of time stamp information.
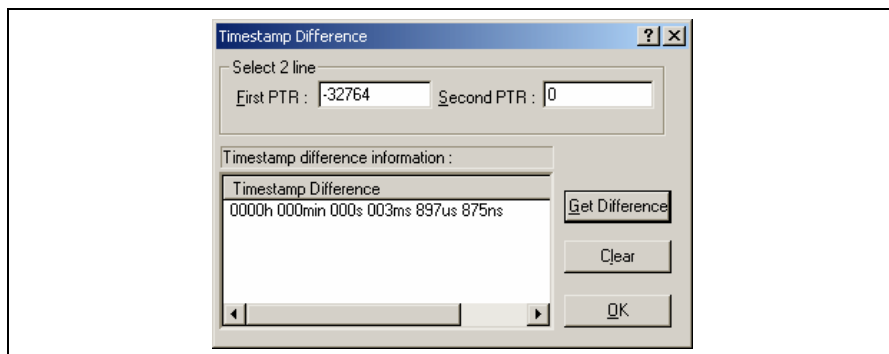
Figure 3.32   [Timestamp Difference] Dialog Box

[Select 2 line]:   Select trace records to calculate the time stamp difference.

    [First PTR]:   Specifies the first pointer to measure the difference. The pointer of the line selected on the [Trace] window is displayed by default.

    [Second PTR]:   Specifies the second pointer to measure the difference.

[Timestamp Difference]:   Displays the results of calculation.

[Get Difference]:   Calculates the difference between the specified two points and display its result in the [Timestamp Difference] list.

[Clear]:   Clears all the results in the [Timestamp Difference] list.

[OK]:   Closes the dialog box. All the results in the [Timestamp Difference] list are cleared.

3.8.13 Analyzing Statistical Information

Choose [Statistic] from the popup menu to open the [Statistic] dialog box and analyze statistical information under the specified conditions.
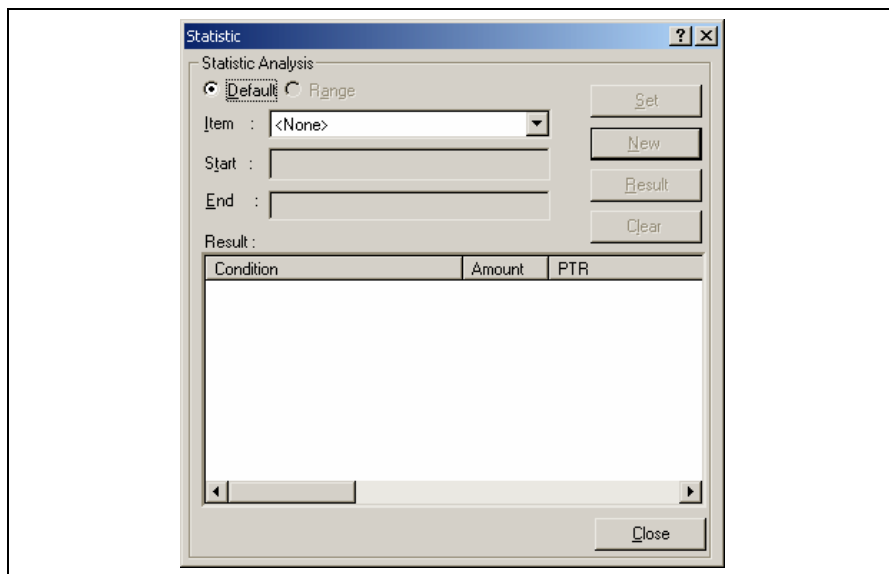
Figure 3.33 [Statistic] Dialog Box

[Statistic Analysis]:   Setting required for analysis of statistical information.

    [Default]:   Sets a single input value or character string.

    [Range]:   Sets the input value or character string as a range.

    [Item]:   Sets the item for analysis.

    [Start]:   Sets the input value or character string. To set a range, the start value must be specified here.

    [End]:   Specify the end value if a range has been set (only available when [Range] has been selected).

    [Set]:   Adds a new condition to the current one.

    [New]:   Creates a new condition.

    [Result]:   Obtains the result of statistical information analysis.

    [Clear]:   Clears all conditions and results of statistical information analysis.

    [Close]:   Closes this dialog box. All the results displayed in the [Result] list will be cleared.

This dialog box allows the user to analyze statistical information concerning the trace information. Set the target of analysis in [Item] and the input value or character string by [Start] and [End]. Click the [Result] button after setting a condition by pressing the [New] or [Add] button to analyze the statistical information and display its result in the [Result] list.

Note: In this emulator, only [PTR] can be set as a range. Each of other items must be specified as a character string. In analysis of statistical information, character strings are compared with those displayed in the [Trace] window. Only those that completely match are counted. Note, however, that this test is not case sensitive. The number of blanks will not be cared either.

3.8.14 Extracting Function Calls from the Acquired Trace Information

To extract function calls from the acquired trace information, select [Function Call…] from the popup menu. The [Function Call Display] dialog box will be displayed.
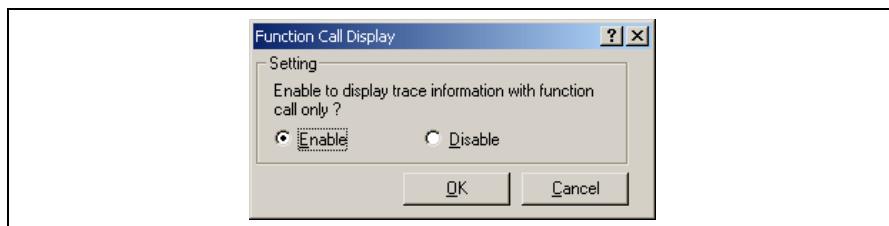
Figure 3.34 [Function Call Display] Dialog Box

[Setting]:   Selects whether or not to extract function calls.

    [Enable]:   Extracts function calls.

    [Disable]:   Does not extract function calls.

When [Enable] is selected, only the cycles that include function calls are extracted for display from the acquired trace information. The content of the trace buffer is not changed by extraction of function calls. Using this function for the result of the free trace or the trace information that includes function calls allows the user to know the order of function calls.

3.9 Analyzing Performance

Use the performance analysis function to measure the rate of execution time. The performance analysis function does not affect the realtime operation because it measures the rate of execution time in the specified range by using the circuit for measurement of hardware performance included in the emulator.

Select one of the following five modes according to the purpose of measurement.

Table 3.1   Available Measurement Modes

Mode	Description	Purpose
Time Of Specified Range Measurement	Measures the execution time and execution count in the specified range.	Measurement of time taken for processing of functions except for that required for child functions called from the functions.
Start Point To End Point Measurement	Measures the execution time and execution count between the specified addresses.	Measurement of time taken for processing of functions.
Start Range To End Range Measurement	Measures the execution time from a specified range to another specified range.	Measurement of execution time spent from calling of any of sequential subroutines to calling of any of other sequential subroutines in a program that includes subroutines in sequence, such as an assembly program.
Access Count Of Specified Range Measurement	Measures the number of times a specified range is accessed from another specified range.	Measurement of the number of times a global variable is accessed from a specific function.
Called Count Of Specified Range Measurement	Measures the number of times a specified range has called another specified range.	Measurement of the number of times a function is called from a specific function.

Use eight performance channels installed on the circuit for measurement of hardware performance in the emulator for setting of conditions for measurement. Up to eight points can be set.

Note, however, that up to four points can be set in Start Range To End Range Measurement, Access Count Of Specified Range Measurement, or Called Count Of Specified Range Measurement because two sequential points are used for setting a condition in these modes.

RENESAS

Table 3.2   Mode Settings for Measurement

Measurement Mode	Point							
	1	2	3	4	5	6	7	8
Time Of Specified Range Measurement	O	O	O	O	O	O	O	O
Start Point To End Point Measurement	O	O	O	O	O	O	O	O
Start Range To End Range Measurement	O	—	O	—	O	—	O	—
Access Count Of Specified Range Measurement	O	—	O	—	O	—	O	—
Called Count Of Specified Range Measurement	O	—	O	—	O	—	O	—

Note: O: Available
        —: Not available

Note: Only one point is used in Time Of Specified Range Measurement and Start Point To End Point
        Measurement, while two sequential points are used in Start Range To End Range Measurement, Access
        Count Of Specified Range Measurement, and Called Count Of Specified Range Measurement. The
        conditions that have been set will be canceled when switching these modes of different types.

RENESAS

3.9.1   Opening the [Performance Analysis] Window

Choose [View -> Performance -> Performance Analysis] or click the [PA] toolbar button (⊞) to open the [Select Performance Analysis Type] dialog box.
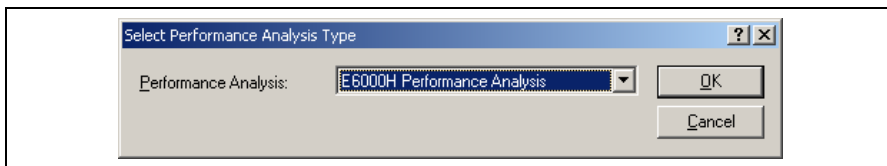
Figure 3.35   [Select Performance Analysis Type] Dialog Box

Select [E6000H Performance Analysis] and then click the [OK] button to open the [Performance Analysis] window.
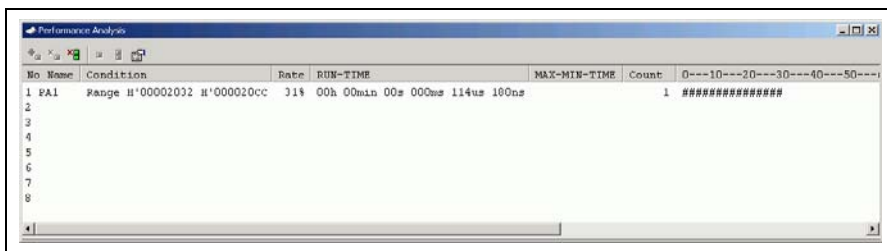
Figure 3.36   [Performance Analysis] Window

This window displays the rate of execution time in the area selected by the user during the last program run in percentages, histogram, or numerical values.

It is possible to hide any column not necessary in the [Performance Analysis] window. Selecting a column you want to hide from the popup menu displayed by clicking the right-hand mouse button on the header column hides that column. To display the hidden column, select the column from the said popup menu again.

RENESAS

3.9.2 Setting Conditions for Measurement

Conditions for measurement can be displayed and changed in the [Performance Analysis] window. Select a point where a condition is to be set, and then select [Set…] from the popup menu to display the [Performance Analysis Properties] dialog box.

Select either from the following five modes as the condition by the [Measurement Method] option:

Table 3.3   Conditions for Measurement (Measurement Method)

[Measurement Method] Option
Time Of Specified Range Measurement
Start Point To End Point Measurement
Start Range To End Range Measurement
Access Count Of Specified Range Measurement
Called Count Of Specified Range Measurement

Set a condition for measurement according to the mode being selected. The parameters to be set depend on the modes.

The [Performance Analysis Properties] window has a support function to enter the address range of a function automatically if the name of the function is entered to set an address range. Entering a function name in the [Input Function Range] dialog box displayed by clicking the […] button on the [Performance Analysis Properties] dialog box automatically enters the address range of the function.
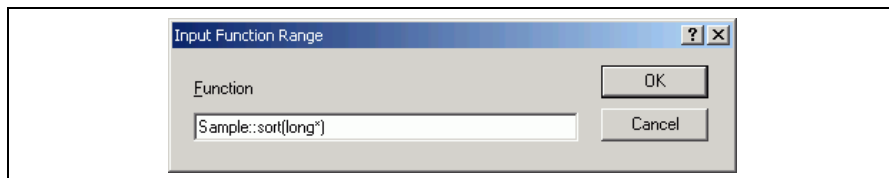
Figure 3.37   [Input Function Range] Window

Notes: 1. Entering the name of an overload function or a class opens the [Select Function] dialog box. Select a function in this dialog box.

2. The addresses figured out are just for reference. In some cases, the end address of a function may be different. Check the last instruction of the function in the [Disassembly] window to correct the value set in [End Address] so that it will be the address of the last instruction (in general, the last instruction of a function is a RTS instruction). A label name or an expression can be entered instead of an address value in boxes where an address should be entered.

(1) Time Of Specified Range Measurement
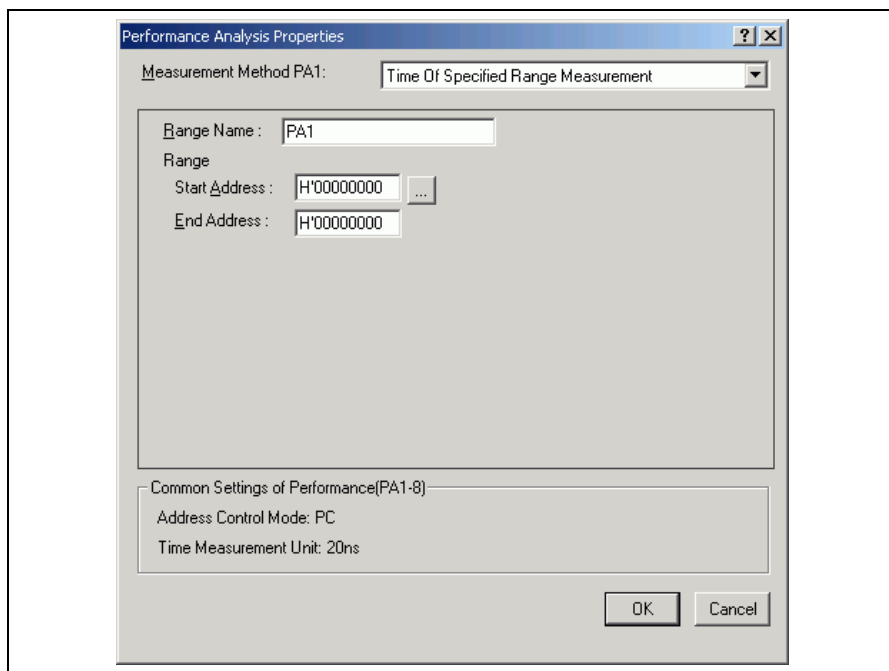
Figure 3.38 Time Of Specified Range Settings

[Range Name]: The name of the range to be measured

[Range]: The range for the Time Of Specified Range Measurement

[Start Address]: Address to start measurement

[End Address]: Address to end measurement

Measures the execution time and the execution count in the range between the start address and end address. Starts measurement with a detected program prefetch in the range specified between the start and end addresses, and then stops with a detected program prefetch out of the specified range. Measurement can be restarted with a detected program prefetch in the specified range. The execution count is incremented every time the program is prefetched at the end address of the specified range. The execution time measured does not include the time spent while being called from the specified range.

(2) Start Point To End Point Measurement
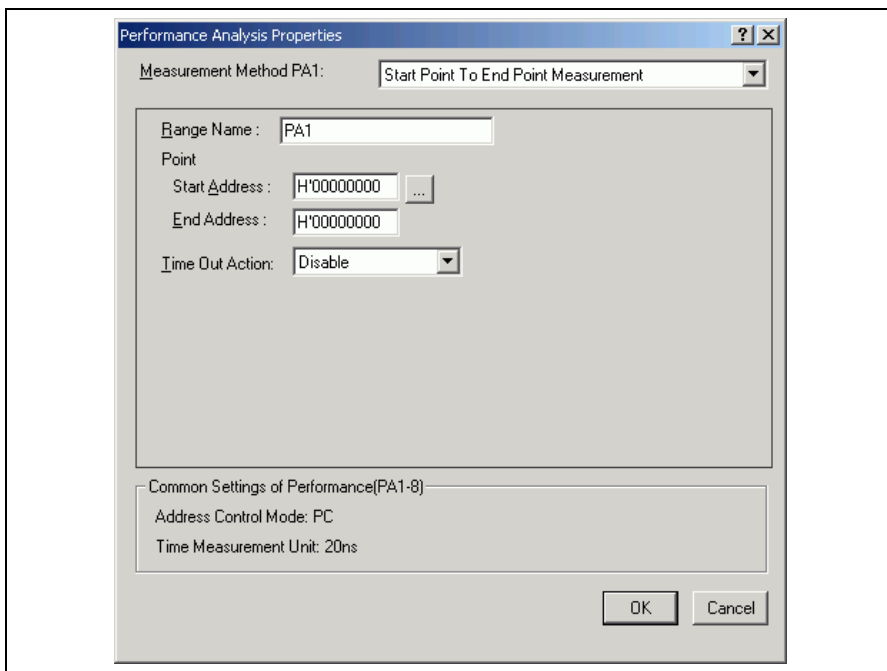
Figure 3.39   Start Point To End Point Measurement Settings

[Range Name]:   The name of the range to be measured

[Point]:   The range for the Start Point To End Point Measurement

    [Start Address]:   Address to start measurement

    [End Address]:   Address to end measurement

[Time Out Action]:   The action to take when a timeout or count-out occurs.

    Disable: Disables setting of a timeout or count-out value.
    Enable: Stops the user program execution when a timeout or count-out occurs.
    Trace Stop: Stops trace acquisition when a timeout or count-out occurs.

    This is only available for channel 1.

[Time Out]:   The timeout value to finish measurement. When the minimum time for measurement
    is 160 ns, 40 ns, or 20 ns, enter the value as follows.
    Example: 1h 2min 3s 123ms 456us 789ns

    If the CPU operating mode is target, enter a hexadecimal number in 10 digits.
    Example: 123456789A

    A break occurs every time a value measured in the specified range exceeds the
    timeout value (not the total time). This is only available for channel 1.

71

[Count]:   The count-up value used in measurement of the execution count. A break occurs
                            every time the execution count exceeds the count-up value. This is only available for
                            channel 1.

Measures the execution time and the execution count in the range between start address and end address. Starts
measurement with a detected program prefetch at the start address, and then stops with a detected program
prefetch at the end address. The execution count is incremented every time the program is prefetched at the end
address of the specified range. The execution time measured includes the time spent while being called from the
specified range. When either from one to four points is selected, the maximum and minimum time in the
specified range can be measured.

(3) Start Range To End Range Measurement
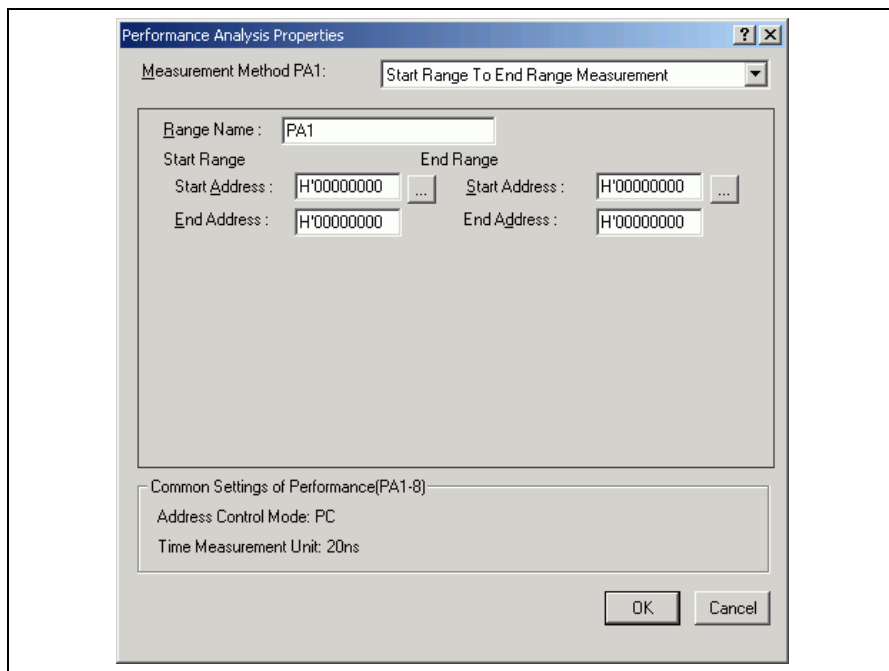
Figure 3.40   Start Range To End Range Measurement Settings

[Range Name]:   The name of the range to be measured

[Start Range]:   The start range for the Start Range To End Range Measurement

    [Start Address]:   Start address

    [End Address]:   End address

[End Range]:   The end range for the Start Range To End Range Measurement

    [Start Address]:   Start address

    [End Address]:   End address

Starts measurement with a detected prefetch cycle in the specified start address range, and then stops with a detected prefetch cycle in the specified end address range. The execution count is incremented every time the program passes the end address range.

RENESAS

(4) Access Count Of Specified Range Measurement
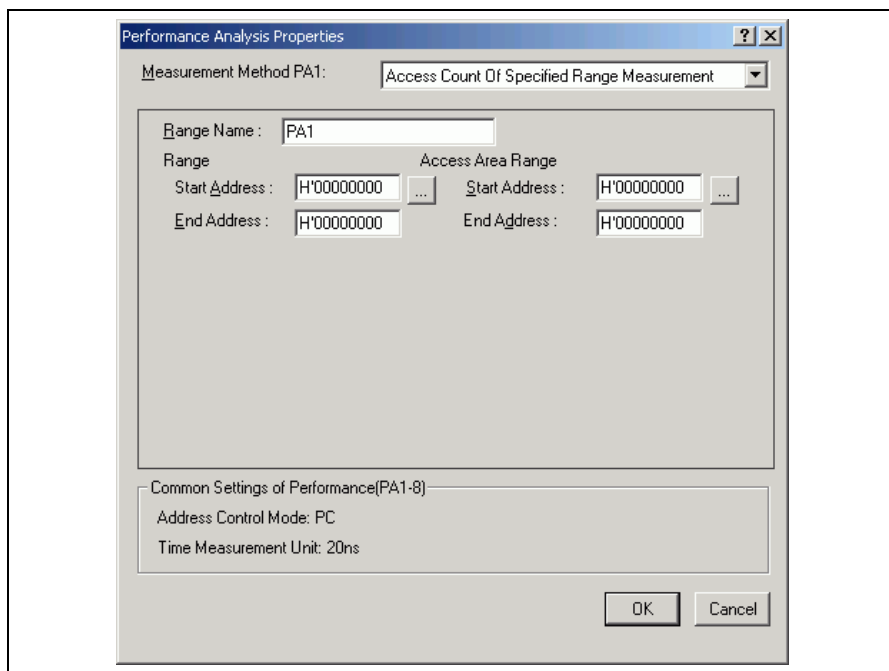
Figure 3.41   Access Count Of Specified Range Measurement Settings

[Range Name]:   The name of the range to be measured

[Range]:   The range for the Access Count Of Specified Range Measurement

        [Start Address]:   Start address

        [End Address]:   End address

[Access Area Range]:   The access range for the Access Count Of Specified Range Measurement

        [Start Address]:   Start address

        [End Address]:   End address

Measures the number of times the range specified as the access range is accessed from the range specified by the start and end addresses. The execution count in the range is measured with Time Of Specified Range Measurement mode.

(5) Called Count Of Specified Range Measurement
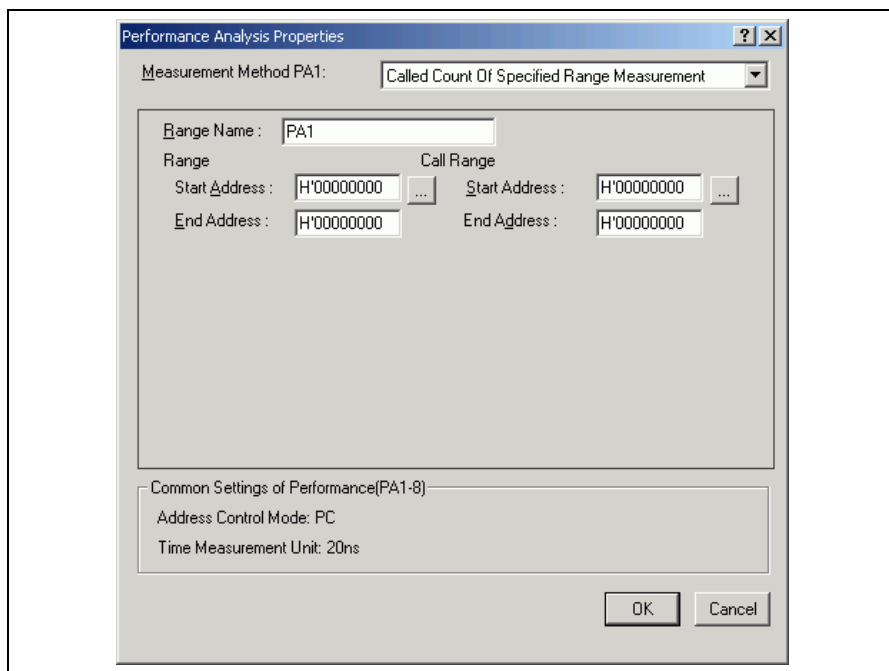
Figure 3.42 Called Count Of Specified Range Measurement Settings

[Range Name]:   The name of the range to be measured

[Range]:   The range for the Called Count Of Specified Range Measurement

    [Start Address]:   Start address

    [End Address]:   End address

[Call Range]:   The range for the Called Count Of Specified Range Measurement. As the call range, specify the start and end addresses of the selected subroutine.

    [Start Address]:   Start address

    [End Address]:   End address

Measures the number of times the range specified as the call range is called from the range specified by the start and end addresses. The execution time in the specified range can be measured with Time Of Specified Range Measurement mode. As the call range, specify the start and end addresses of the selected subroutine.

RENESAS

3.9.3   Starting Performance Data Acquisition

Executing the user program clears the result of previous measurement and automatically starts measuring the rate of execution time according to the conditions that have been set. Stopping the user program displays the result of measurement in the [Performance Analysis] window.

3.9.4   Deleting a Measurement Condition

Select [Reset] from the popup menu with a measurement condition selected to delete the condition.

3.9.5   Deleting All Measurement Conditions

Choose [Reset All] from the popup menu to delete all the conditions that have been set.

3.10　Profiling Function

3.10.1　Enabling the Profile

Choose [View->Performance->Profile] to open the [Profile] window. Choose [Enable Profiler] from the popup menu of the [Profile] window. The item on the menu will be checked.

3.10.2　Specifying Measuring Mode

You can specify whether to trace functions calls while profile data is acquired. When function calls are traced, the relations of function calls during user program execution are displayed as a tree diagram. When not traced, the relations of function calls cannot be displayed, but the time for acquiring profile data can be reduced.

To stop tracing function calls, choose [Disable Tree (Not traces function call)] from the popup menu in the [Profile] window (a check mark is shown to the left of the menu item).
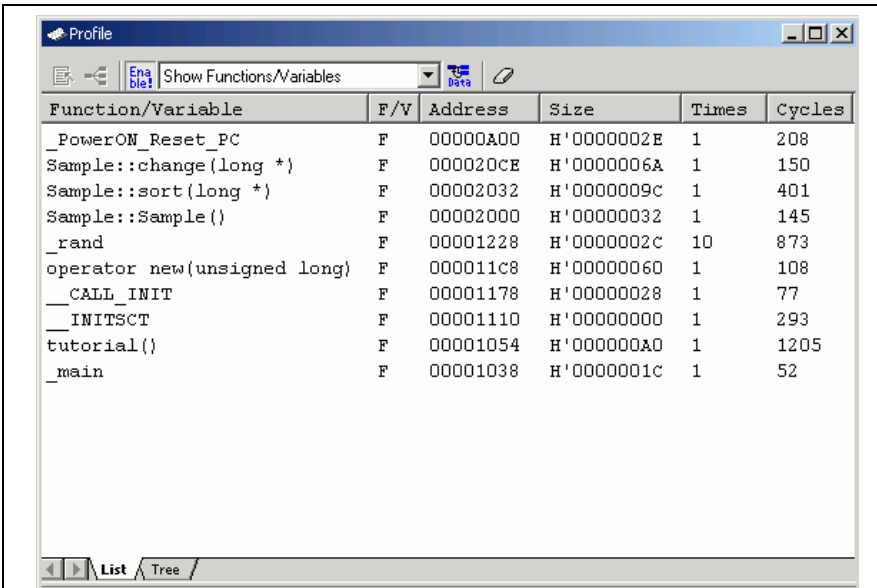
When acquiring profile data of the program in which functions are called in a special way, such as task switching in the OS, stop tracing function calls.

3.10.3　Executing the Program and Checking the Results

After the user program has been executed and execution has been halted, the results of measurement are displayed in the [Profile] window.

The [Profile] window has two sheets; a [List] sheet and a [Tree] sheet.

3.10.4   [List] Sheet

Figure 3.43   [Profile] Window ([List] Sheet)
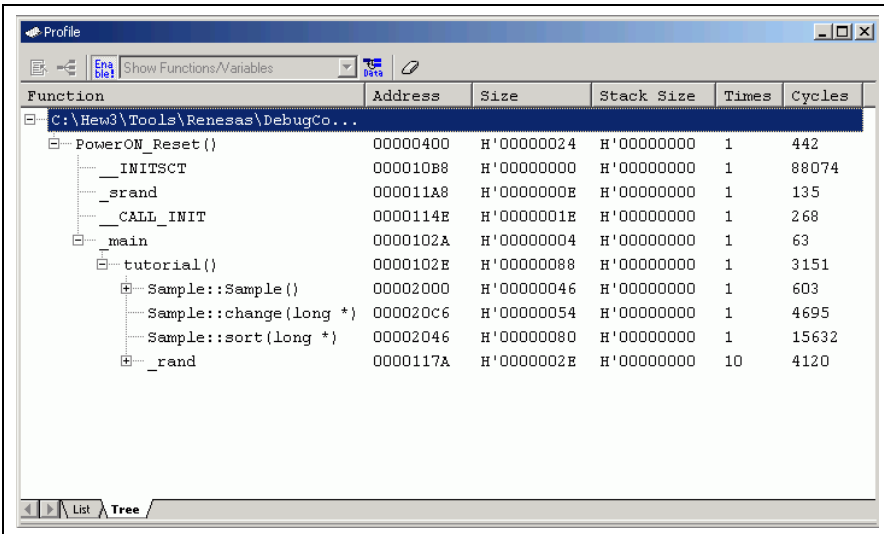
This window displays the address and size of a function or a global variable, the number of times the function is called or the global variable is accessed, and profile data.

When the column header is clicked, data are sorted in alphabetic or numeric ascending/descending order.

Double-clicking the [Function/Variable] or [Address] column displays the source program of the address in the line. Right-clicking on the mouse within the window displays a popup menu. For details on this popup menu, refer to section 3.10.5, [Tree] Sheet.

Note: For notes on the profiling function, refer to section 5.9, Profiling Function.

RENESAS

3.10.5 [Tree] Sheet

Figure 3.44 [Profile] Window ([Tree] Sheet)

This window displays the relation of function calls in a tree structure. Displayed contents are the address, size, stack size, and number of function calls and execution cycles. The stack size and number of function calls are values when the function is called.

The [Tree] sheet is only available when [Not trace the function call] is not checked in the popup menu of the [Profile] window.

Double-clicking a function in the [Function] column expands or reduces the tree structure display. The expansion or reduction is also provided by the "+" or "-" key. Double-clicking the [Address] column displays the source program of the specific address.

Right-clicking on the mouse within the window displays a popup menu. Supported menu options are described in the following sections:

- **View Source**

Displays the source program or disassembled memory contents for the address in the selected line.

- **View Profile-Chart**

Displays the [Profile-Chart] window focused on the function in the specified line.

- **Enable Profiler**

Toggles acquisition profile data. When profile data acquisition is active, a check mark is shown to the left of the menu text.

- **Not trace the function call**

Stops tracing function calls while profile data is acquired. This menu is used when acquiring profile data of the program in which functions are called in a special way, such as task switching in the OS.

To display the relation of function calls in the [Tree] sheet of the [Profile] window, acquire profile data without selecting this menu. In addition, do not select this menu when optimizing the program by the optimizing linkage editor using the acquired profile information file.

- **Find…**

Displays the [Find Text] dialog box to find a character string in the [Function] column. Search is started by inputting a character string to be found in the edit box and clicking [Find Next] or pressing the Enter key.

- **Clear Data**

Clears the number of times functions are called and profile data. Data in the [Profile] window's [List] sheet and the [Profile-Chart] window are also cleared.

- **Output Profile Information Files…**

Displays the [Save Profile Information Files] dialog box. Profiling results are saved in a profile information file (.pro extension). The optimizing linkage editor optimizes user programs according to the profile information in this file. For details of the optimization using the profile information, refer to the manual of the optimizing linkage editor.

Note: If profile information has been acquired by selecting the [Not trace the function call] menu, the program cannot be optimized by the optimizing linkage editor.

- **Output Text File…**

Displays the [Save Text of Profile Data] dialog box. Displayed contents are saved in a text file.

- **Setting**

This menu has the following submenus (the menus available only in the [List] sheet are also included).

1.   Show Functions/Variables

Displays both functions and global variables in the [Function/Variable] column.

2.   Show Functions

Displays only functions in the [Function/Variable] column.

3.   Show Variables

Displays only global variables in the [Function/Variable] column.

4.   Only Executed Functions

Only displays the executed functions. If a stack information file (.sni extension) output from the optimizing linkage editor does not exist in the directory where the load module is located, only the executed functions are displayed even if this check box is not checked.

5. Include Data of Child Functions

Sets whether or not to display information for a child function called in the function as profile data.

- **Properties...**

This popup menu is unavailable in the SH7046 E6000H emulator.
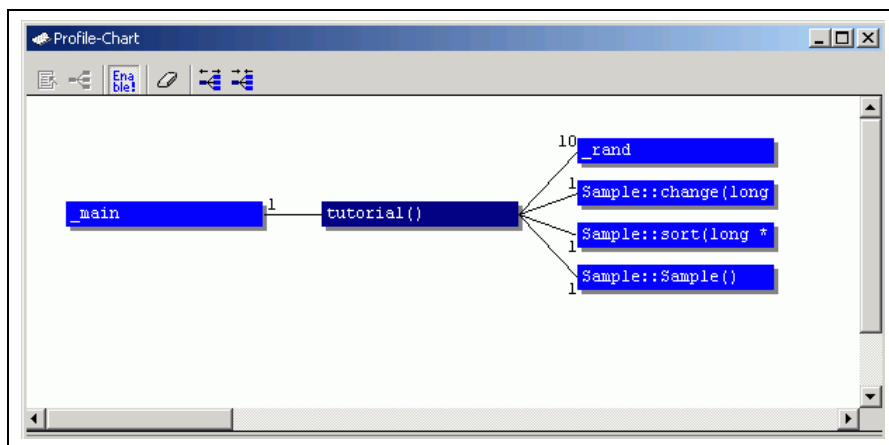
3.11 [Profile-Chart] Window

Figure 3.45 [Profile-Chart] Window

This window displays the relation of calls for a specific function. This window displays the calling relation for the function specified in the [List] sheet or [Tree] sheet in the [Profile] window. The specified function is displayed in the middle, the calling function on the left side, and the called function on the right side. Values beside the calling and called functions show the number of times the function has been called.

Right-clicking on the mouse within the window displays a popup menu. Supported menu options are described in the following sections.

- **View Source**

Displays the source program or disassembled memory contents for the address of the function on which the cursor is placed when the right-hand mouse button is clicked. If the cursor is not placed on a function when the right-hand mouse button is clicked, this menu option is displayed in gray characters.

- **View Profile-Chart**

Displays the [Profile-Chart] window for the specific function on which the cursor is placed when the right-hand mouse button is clicked. If the cursor is not placed on a function when the right-hand mouse button is clicked, this menu option is displayed in gray characters.

- **Enable Profiler**

Toggles acquisition of profile data. When profile data acquisition is active, a check mark is shown to the left of the menu text.

- **Clear Data**

Clears the number of times functions are called and profile data. Data in the [List] sheet and [Tree] sheet in the [Profile] window are also cleared.

RENESAS

- **Multiple View**

If the [Profile-Chart] window is going to be opened when it has already been opened, selects whether another window is to be opened or the same window is to be used to display data. When a check mark is shown to the left side of the menu text, another window is opened.

- **Output Profile Information File…**

Displays the [Save Profile Information File] dialog box. Profiling results are saved in a profile information file (.pro extension). The optimizing linkage editor optimizes user programs according to the profile information in this file. For details of the optimization using the profile information, refer to the manual of the optimizing linkage editor.

- **Expands Size**

Expands spaces between each function. The "+" key can also be used to expand spaces.

- **Reduces Size**

Reduces spaces between each function. The "-" key can also be used to reduce spaces.

RENESAS

Section 4   Tutorial

4.1   Introduction

This section describes the main functions of the emulator by using a tutorial program.

The tutorial program is based on the C++ program that sorts ten random data items in ascending or descending order. The tutorial program performs the following actions:

- The `main` function repeatedly calls the `tutorial` function to repeat sorting.
- The `tutorial` function generates random data to be sorted and calls the `sort` and `change` functions in that order.
- The `sort` function enters the array where the random data generated by the `tutorial` function are stored, and sorts them in the ascending order.
- The `change` function then sorts the array, which was sorted in ascending order by the `sort` function, in descending order.

The file `tutorial.cpp` contains the source code for the tutorial program. The file `Tutorial.abs` is a compiled load module in the Dwarf2 format.

Notes: 1. After recompilation, the addresses may differ from those given in this section.

      2. This section describes general usage examples for the emulator. For the particular specifications of each product, refer to section 3, Debugging, or the online help.

      3. The operation address of `Tutorial.abs` attached to each product differs depending on the product. Replace the address used in this section with the correct address in each product after checking that it is placed on the corresponding line of the source program.

      4. In this tutorial, the SH7046 E6000H emulator is taken as an example. File paths or the appearance of figures differ depending on the product.

4.2   Running the High-performance Embedded Workshop

Open a workspace by following the procedure listed in section 2.1.3, Selecting an Existing Workspace.

Select the following directory.

OS installation drive \Workspace\Tutorial\E6000H\7046

Note: The directory mentioned above cannot be specified depending on the version of the software. In such
            cases, specify the following directory instead.

High-performance Embedded Workshop installation destination directory
\Tools\Renesas\DebugComp\Platform\E6000H\7046\Tutorial

Then select the file indicated below.
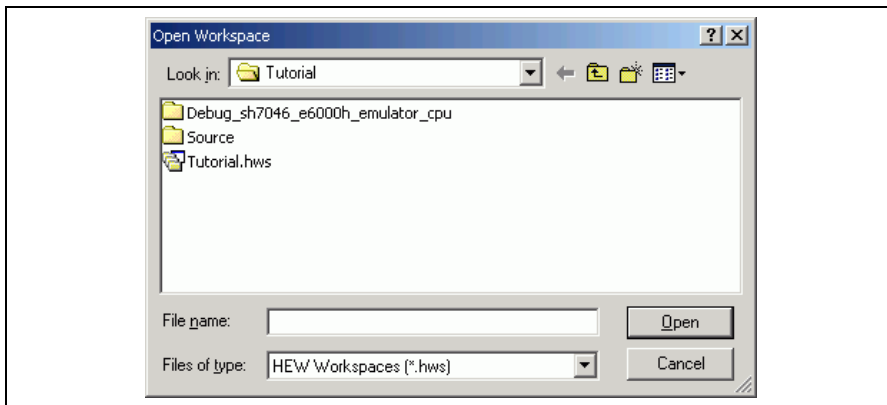
Figure 4.1   [Open Workspace] Dialog Box

RENESAS

4.3 Downloading the Tutorial Program

4.3.1 Downloading the Tutorial Program

Download the object program to be debugged.

- Select [Download module] from [Tutorial.abs] of [Download modules].
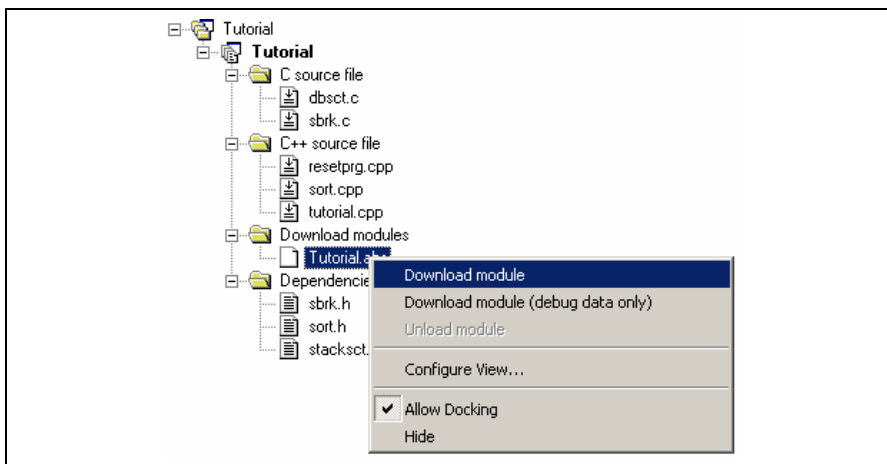
Figure 4.2   Downloading the Tutorial Program

4.3.2   Displaying the Source Program

The High-performance Embedded Workshop allows the user to debug a user program at the source level.

- Double-click [Tutorial.cpp] under [C++ source file].
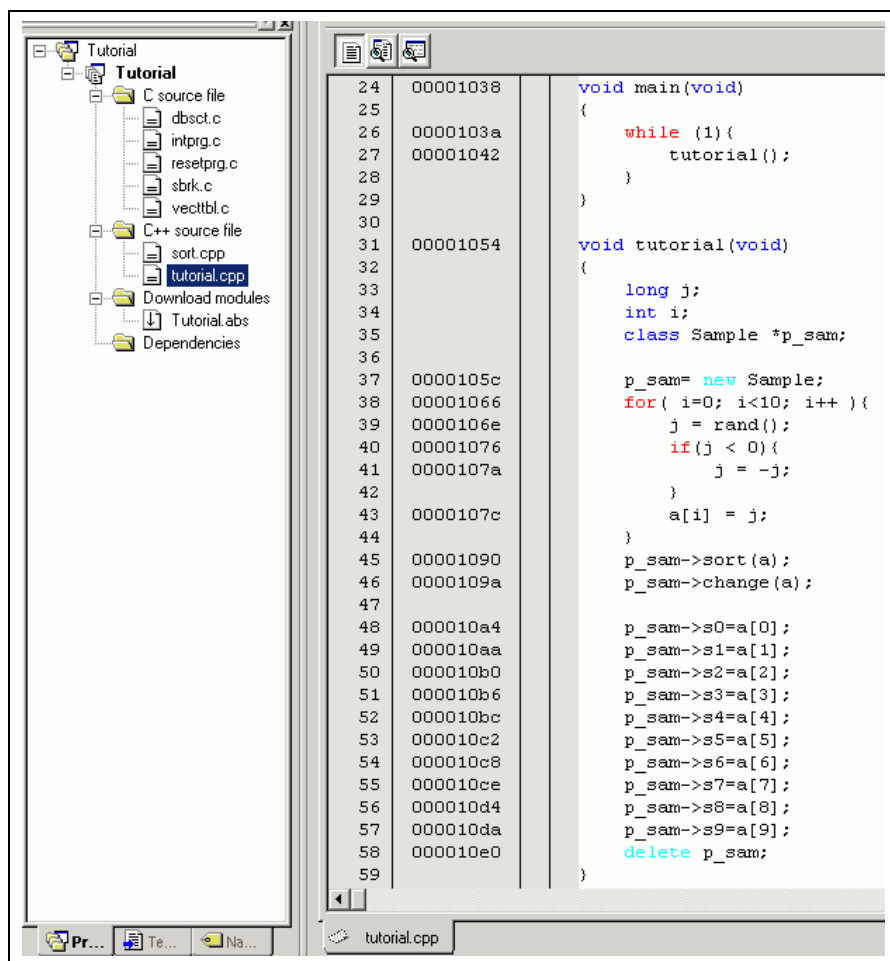
Figure 4.3   [Editor] Window (Displaying the Source Program)

- Select a font and size that are legible if necessary.

Initially the [Editor] window shows the start of the user program, but the user can use the scroll bar to scroll through the user program and look at the other statements.

RENESAS

4.4 Setting a Software Breakpoint

A software breakpoint is a simple debugging function.

The [Editor] window provides a very simple way of setting a software breakpoint at any point in a program. For example, to set a software breakpoint where the sort function is called:

- Select by double-clicking the [S/W Breakpoints] column on the line containing the sort function call.
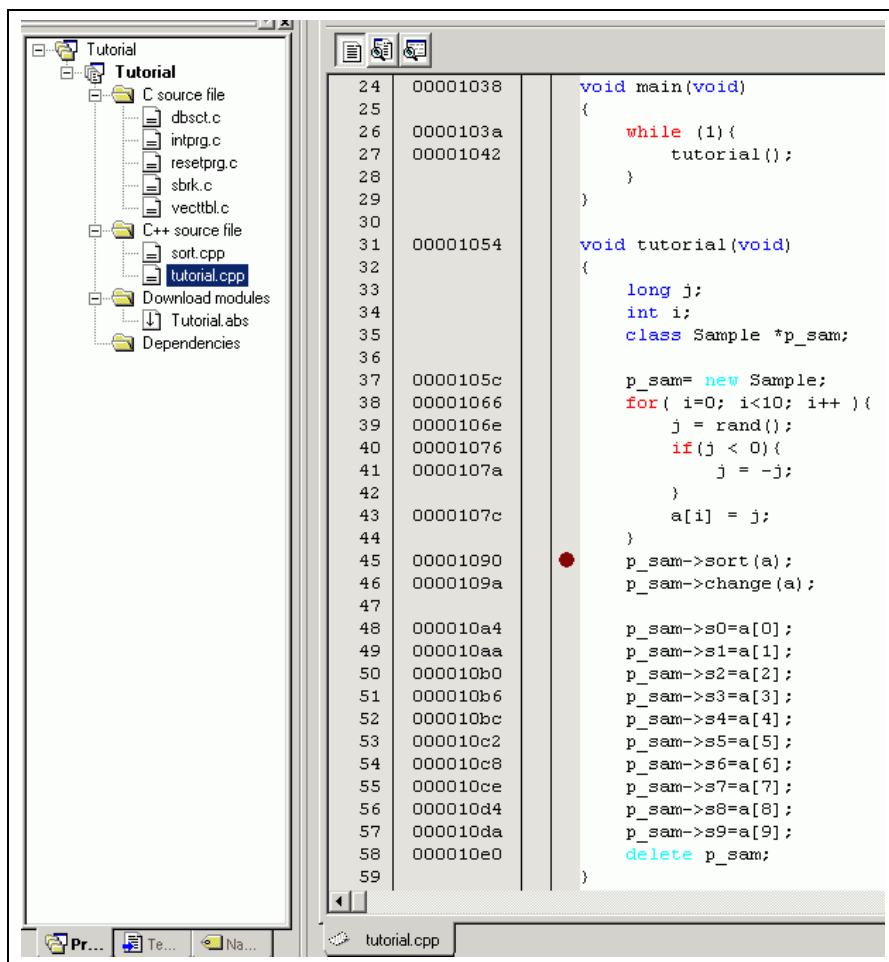
Figure 4.4 [Editor] Window (Setting a Software Breakpoint)

The symbol ● will appear on the line containing the sort function. This shows that a software breakpoint has been set.

RENESAS

4.5 Setting Registers

Set a value in the program counter before executing the program.

- Select [Registers] from the [CPU] submenu of the [View] menu or click the [Register] toolbar button (⊞) to display the [Register] window.
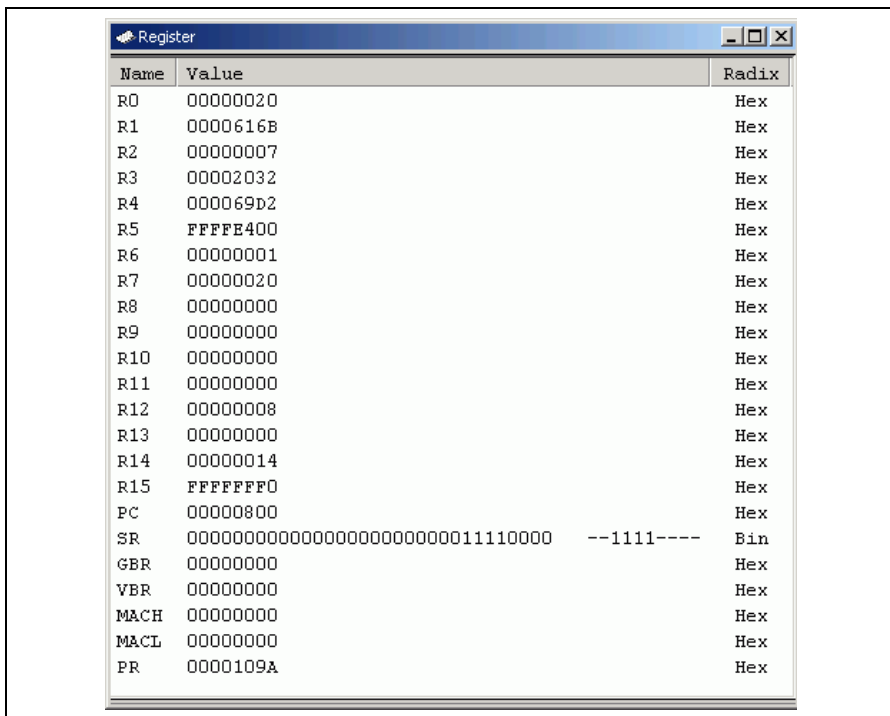
Figure 4.5   [Register] Window

- To change the value of the program counter (PC), double-click on the PC value area in the [Register] window with the mouse. The following dialog box is then displayed, and the value can be changed. Set the program counter to H'00000800 in this tutorial program, and click the [OK] button.
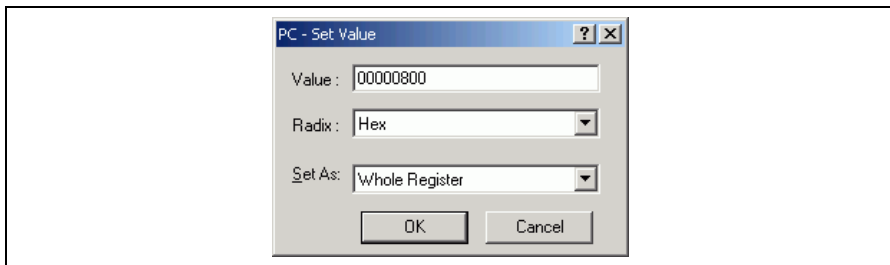
Figure 4.6   [Register] Dialog Box (PC)

RENESAS

4.6   Executing the Program

Execute the program as described in the following:

- To execute the program, select [Go] from the [Debug] menu, or click the [Go] button on the toolbar.
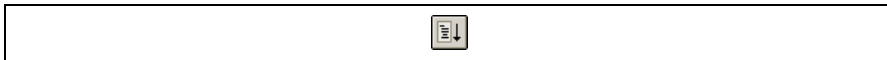
Figure 4.7   [Go] Button

While the program is executed, the current address bus value and the operating state of the MCU are displayed on the status bar.

The program will be executed up to the software breakpoint that has been set, and an arrow will appear on the [S/W Breakpoints] column in the [Editor] window to show the position where the program has halted, with the message [`Break = Software Break`] in the status bar.

Notes: When the source file is displayed after a break, a path of the source file may be inquired. The location of the source file is as follows:

OS installation drive \Workspace\Tutorial\E6000H\7046\Source

The directory mentioned above cannot be specified depending on the version of the software. In such cases, specify the following directory instead.

High-performance Embedded Workshop installation destination directory
\Tools\Renesas\DebugComp\Platform\E6000H\7046\Source

```
24    00001038        void main(void)
25                    {
26    0000103a            while (1){
27    00001042                tutorial();
28                        }
29                    }
30
31    00001054        void tutorial(void)
32                    {
33                        long j;
34                        int i;
35                        class Sample *p_sam;
36
37    0000105c            p_sam= new Sample;
38    00001066            for( i=0; i<10; i++ ){
39    0000106e                j = rand();
40    00001076                if(j < 0){
41    0000107a                    j = -j;
42                            }
43    0000107c                a[i] = j;
44                        }
45    00001090    ⬤      p_sam->sort(a);
46    0000109a            p_sam->change(a);
47
48    000010a4            p_sam->s0=a[0];
49    000010aa            p_sam->s1=a[1];
50    000010b0            p_sam->s2=a[2];
51    000010b6            p_sam->s3=a[3];
52    000010bc            p_sam->s4=a[4];
53    000010c2            p_sam->s5=a[5];
54    000010c8            p_sam->s6=a[6];
55    000010ce            p_sam->s7=a[7];
56    000010d4            p_sam->s8=a[8];
57    000010da            p_sam->s9=a[9];
58    000010e0            delete p_sam;
59                    }
```

tutorial.cpp
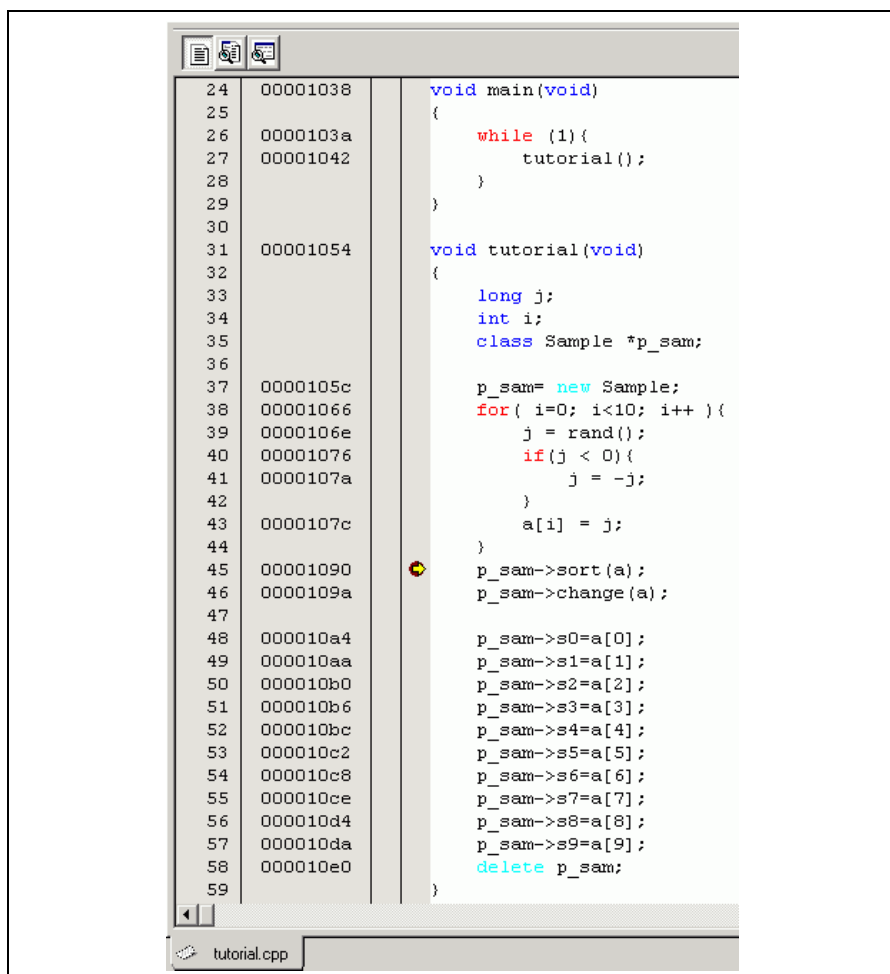
Figure 4.8   [Editor] Window (Break Status)

RENESAS

The user can see the cause of the break that occurred last time in the [Status] window.

- Select [Status] from the [CPU] submenu of the [View] menu or click the [Status] toolbar button (). After the [Status] window is displayed, open the [Platform] sheet, and check the `Status` of `Cause of last break`.
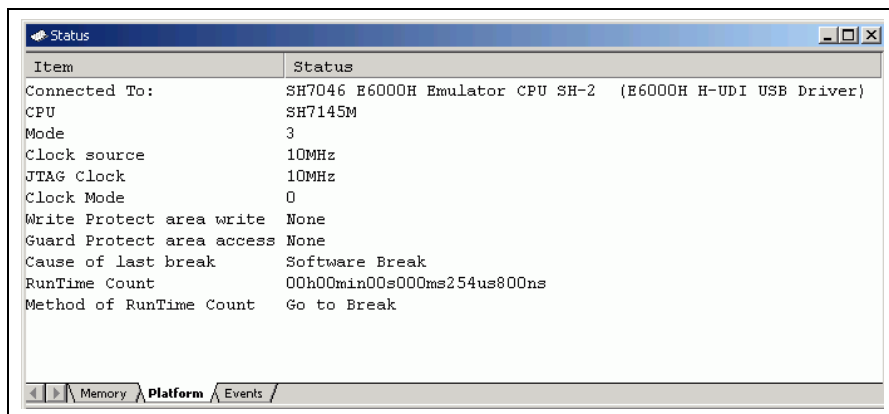
Figure 4.9 [Status] Window

Note: The items that can be displayed in this window differ depending on the product. For the items that can be displayed, refer to section 3, Debugging, or the online help.

4.7　Reviewing Breakpoints

The user can see all the breakpoints set in the program in the [Event] window.

- Select [Eventpoints] from the [Code] submenu of the [View] menu or click the [Eventpoints] toolbar button
  (). The [Event] window is displayed. Select the [Software] sheet.
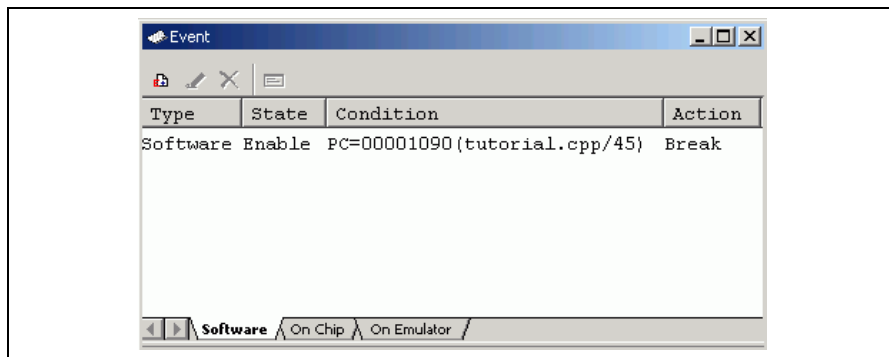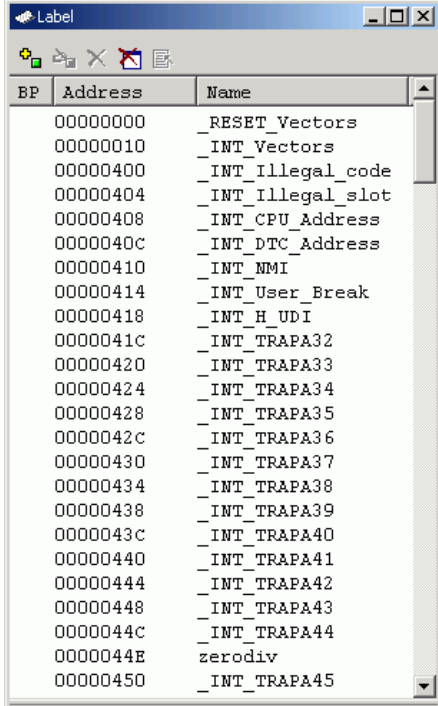
Figure 4.10　[Event] Window

The popup menu, opened by clicking the [Event] window with the right-hand mouse button, allows the user
to set or change breakpoints, define new breakpoints, and delete, enable, or disable breakpoints.

RENESAS

4.8 Referring to Symbols

The [Label] window can be used to display the information on symbols in modules.

Select [Label] from the [Symbol] submenu of the [View] menu. The [Label] window is displayed so that the user can refer to the addresses of symbols in modules.

Figure 4.11   [Label] Window

4.9 Viewing Memory

When the label name is specified, the user can view the memory contents that the label has been registered in the [Memory] window. For example, to view the memory contents corresponding to _main in byte size:

- Select [Memory …] from the [CPU] submenu of the [View] menu or click the [View Memory] toolbar button ( ) to open the [Format] dialog box. Enter **_main** in the [Begin] edit box and **+ff** in the [End] edit box, respectively, and select Byte in the [Format] combo box.
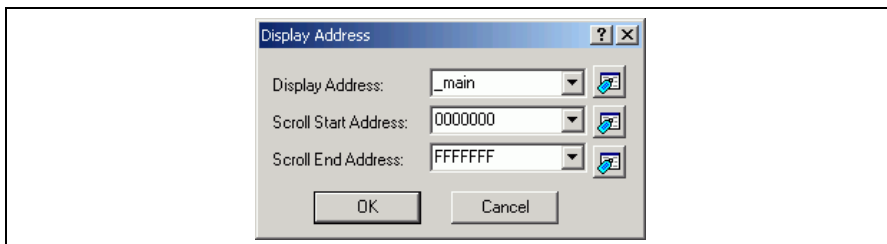
Figure 4.12   [Format] Dialog Box

- Click the [OK] button. The [Memory] window showing the selected area of memory is displayed.
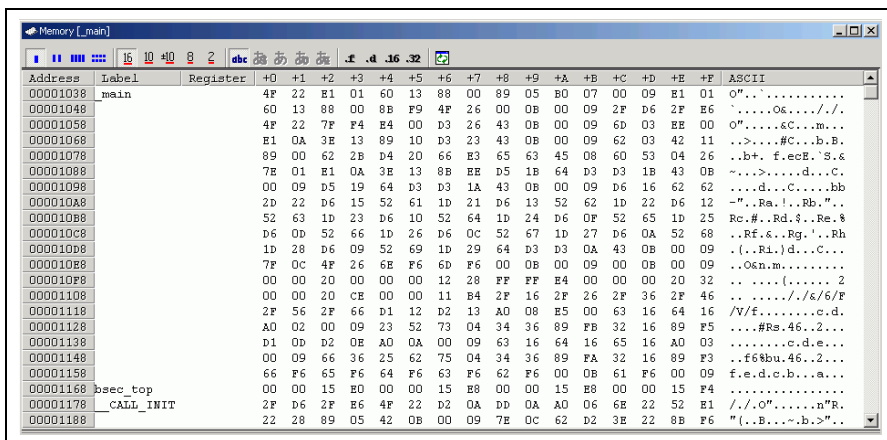
Figure 4.13   [Memory] Window

RENESAS

4.10 Watching Variables

As the user steps through a program, it is possible to watch that the values of variables used in the user program are changed. For example, set a watch on the long-type array a declared at the beginning of the program, by the following procedure:

- Click the left of displayed array a in the [Editor] window to place the cursor.

- Select [Instant Watch...] with the right-hand mouse button.

The following dialog box will be displayed.
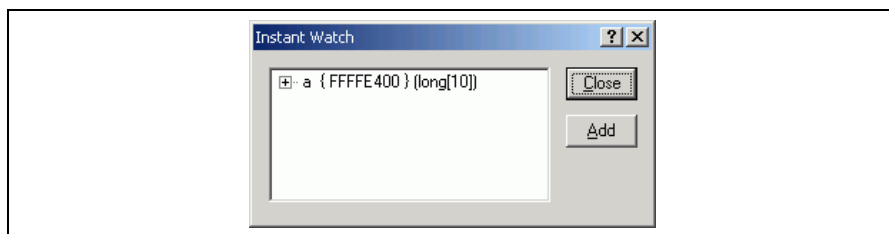
Figure 4.14   [Instant Watch] Dialog Box

- Click the [Add] button to add a variable to the [Watch] window.
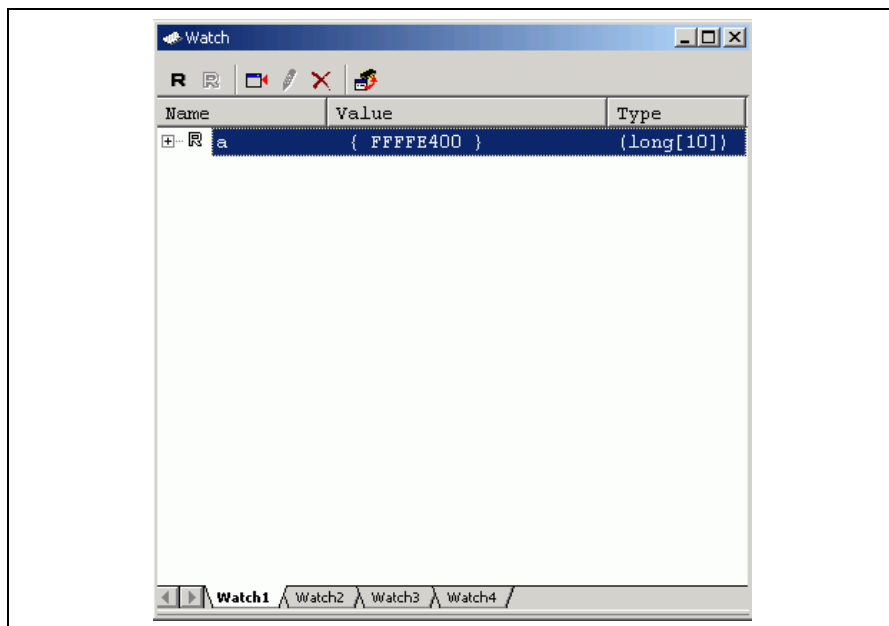
Figure 4.15   [Watch] Window (Displaying the Array)

RENESAS

The user can also add a variable to the [Watch] window by specifying its name.

- Click the [Watch] window with the right-hand mouse button and select [Add Watch…] from the popup menu.

The following dialog box will be displayed.
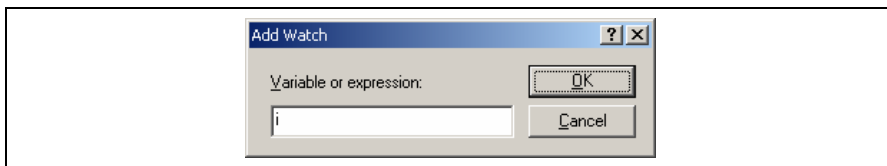
Figure 4.16 [Add Watch] Dialog Box

- Enter variable *i* to [Variable or expression] edit box and click the [OK] button.

The [Watch] window will now also show the int-type variable i.
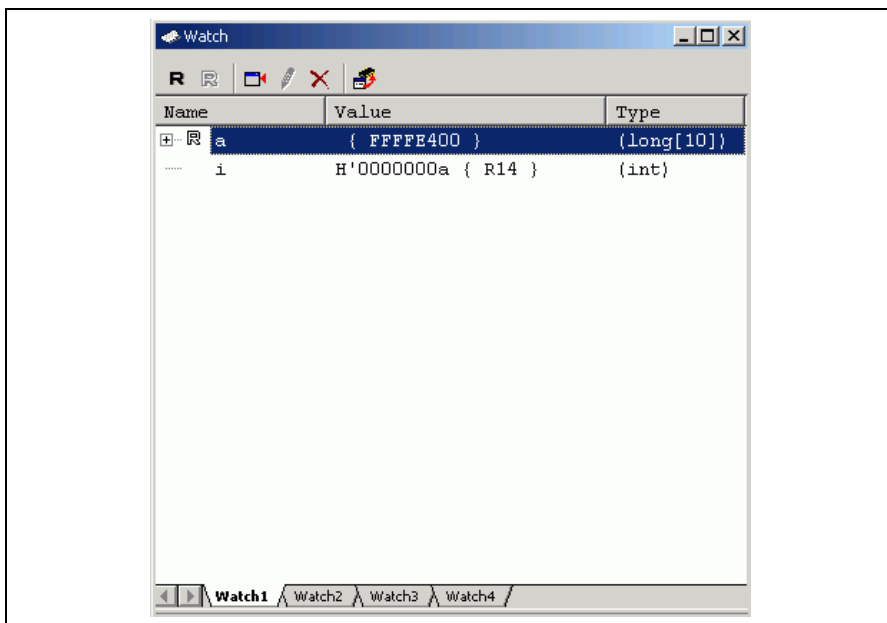
Figure 4.17 [Watch] Window (Displaying the Variable)

The user can click mark '+' at the left side of array a in the [Watch] window to watch all the elements.
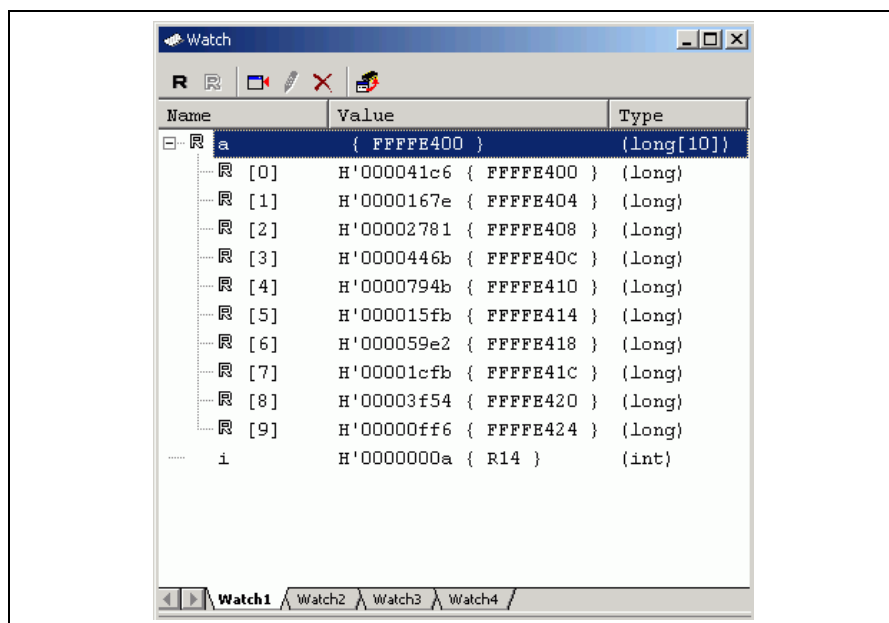
Figure 4.18   [Watch] Window (Displaying Array Elements)

4.11 Displaying Local Variables

The user can display local variables in a function by using the [Locals] window. For example, we will examine the local variables in the tutorial function, which declares local variables j, i, and p_sam.

- Select [Locals] from the [Symbol] submenu of the [View] menu. The [Locals] window is displayed.

The [Locals] window shows the local variables in the function currently pointed to by the program counter, along with their values. Note, however, that the [Locals] window is initially empty because local variables are yet to be declared.
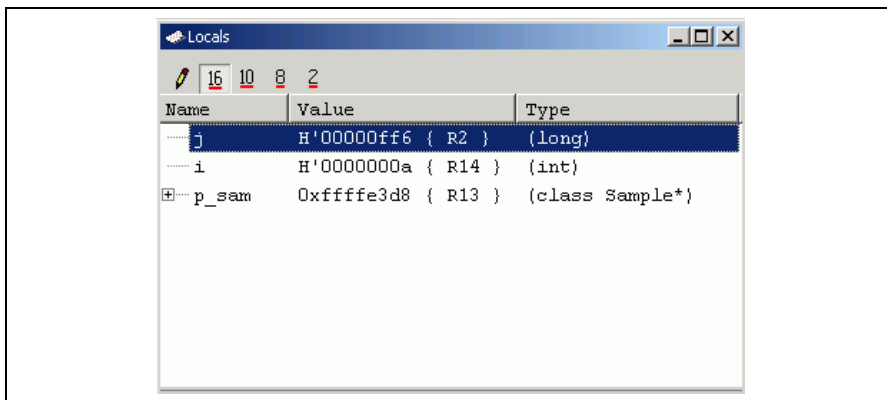
Figure 4.19 [Locals] Window

The user can click mark '+' at the left side of class instance p_sam in the [Locals] window to watch all the elements. View the elements of class instance p_sam before and after the execution of the sort function and check that the random data is sorted in the descending order.

4.12 Stepping Through a Program

The High-performance Embedded Workshop provides a range of step menu commands that allow efficient program debugging.

Table 4.2 Step Options

Menu Command	Description
Step In	Executes each statement, including statements within functions.
Step Over	Executes a function call in a single step.
Step Out	Steps out of a function, and stops at the statement following the statement in the program that called the function.
Step…	Steps the specified times repeatedly at a specified rate.

4.12.1 Executing the [Step In] Command

The [Step In] command steps into the called function and stops at the first statement of the called function.

- To step through the `sort` function, select [Step In] from the [Debug] menu, or click the [Step In] button on the toolbar.
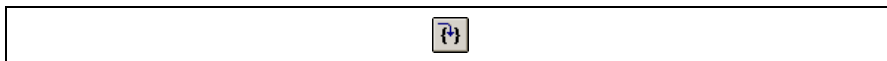

Figure 4.20   [Step In] Button
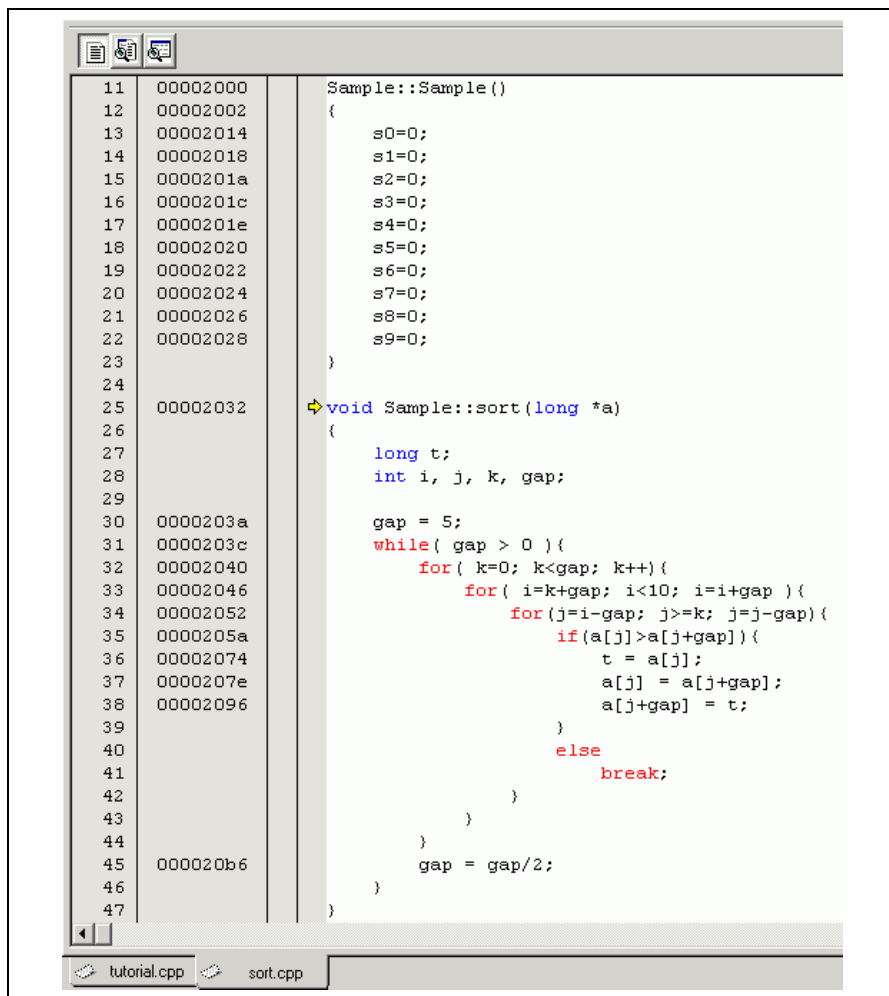
Figure 4.21   [Editor] Window (Step In)

- The highlighted line moves to the first statement of the sort function in the [Editor] window.

RENESAS

4.12.2 Executing the [Step Out] Command

The [Step Out] command steps out of the called function and stops at the next statement of the calling statement.

- To step out of the `sort` function, select [Step Out] from the [Debug] menu, or click the [Step Out] button in the toolbar.

Figure 4.22 [Step Out] Button
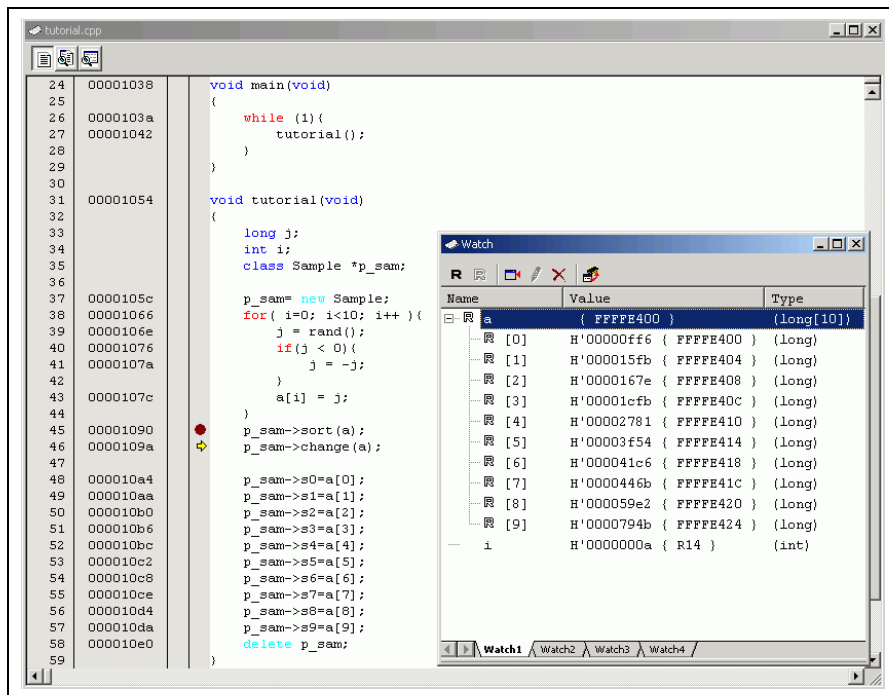
Figure 4.23 [High-performance Embedded Workshop] Window (Step Out)

The data of variable `a` displayed in the [Watch] window is sorted in the ascending order.

4.12.3 Executing the [Step Over] Command

The [Step Over] executes a function call in a single step and stops at the next statement of the main program.

- To step through all statements in the change function in a single step, select [Step Over] from the [Debug] menu, or click the [Step Over] button on the toolbar.

Figure 4.24   [Step Over] Button
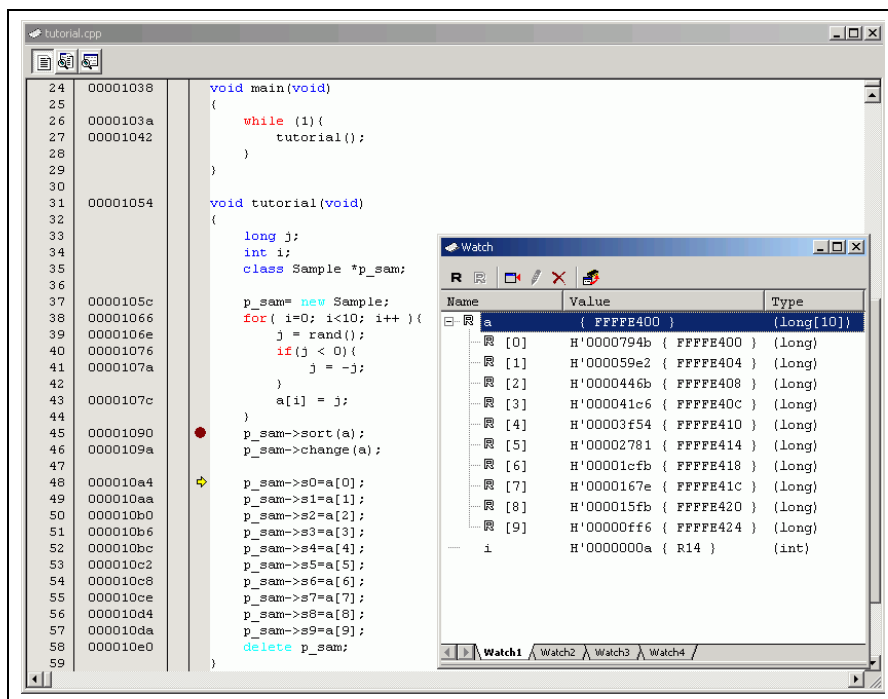
Figure 4.25   [High-performance Embedded Workshop] Window (Step Over)

The data of variable a displayed in the [Watch] window is sorted in the descending order.

4.13　Forced Breaking of Program Executions

The High-performance Embedded Workshop can force a break during the execution of a program.

- Cancel all the breakpoints.

- To execute the remaining sections of the `tutorial` function, select [Go] from the [Debug] menu or the [Go] button on the toolbar.
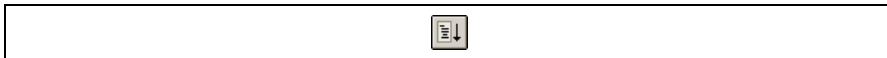
Figure 4.26　[Go] Button

- The program goes into an endless loop. To force a break during execution, select [Halt] from the [Debug] menu or the [Halt] button on the toolbar.

Figure 4.27　[Halt] Button

4.14　Resetting the Target MCU

Resetting the target MCU initializes the on-chip I/O registers and makes the program counter jump to the address set in the reset vector.

- To reset the target MCU, select [Reset CPU] from the [Debug] menu or the [Reset CPU] button on the toolbar.
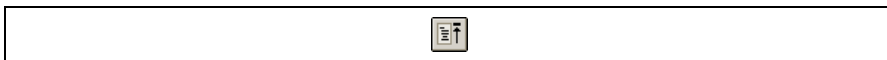
Figure 4.28　[Reset CPU] Button

- To execute the user program immediately after a reset, select [Reset Go] from the [Debug] menu or the [Reset Go] button on the toolbar.
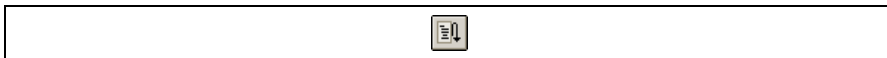

Figure 4.29　[Reset Go] Button

Note:　This tutorial program is executable from the reset vector.

4.15 Break Function

The emulator provides break functions by software breaks, on-chip breaks, and on-emulator breaks. Software breakpoints, on-chip breakpoints, and on-emulator breakpoints can be set in the High-performance Embedded Workshop's [Event] window.

An overview and setting of the break function are described below.

4.15.1 Software Break Function

The emulator can set up to 255 software breakpoints.

- Select [Eventpoints] from the [Code] submenu of the [View] menu or click the [Eventpoints] toolbar button (). The [Event] window is displayed.
- Select the [Software] sheet.
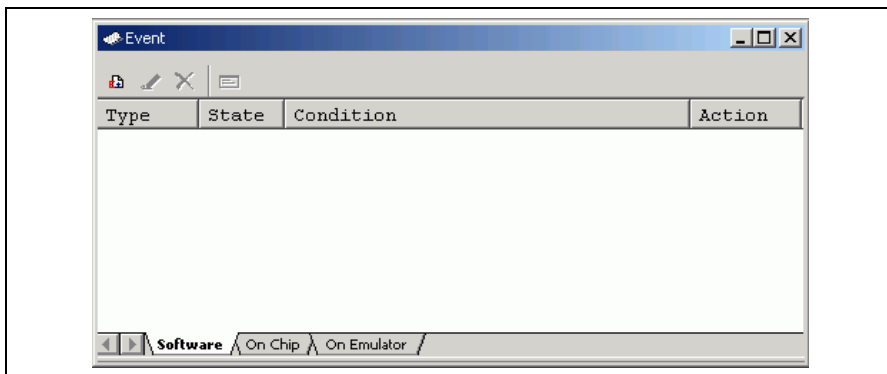
Figure 4.30   [Event] Window (Before Setting a Software Breakpoint)

- Click the [Event] window with the right-hand mouse button and select [Add…] from the popup menu.
- The [Breakpoint Properties] dialog box ([Software Break] page) is displayed.

RENESAS
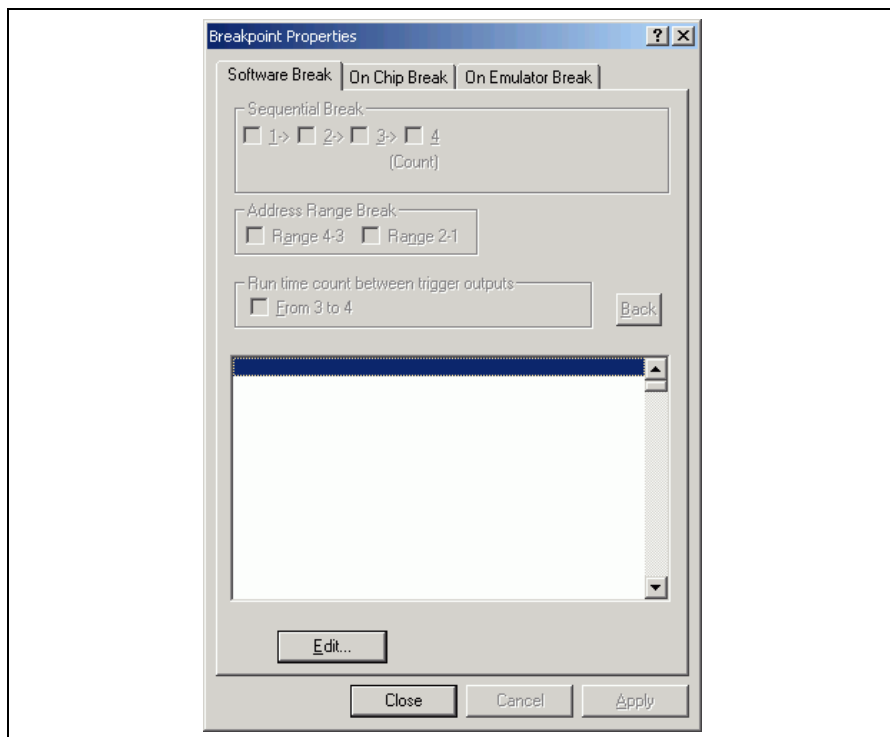
Figure 4.31 [Breakpoint Properties] Dialog Box

- Click the [Edit…] button to display the [Software Break] dialog box.
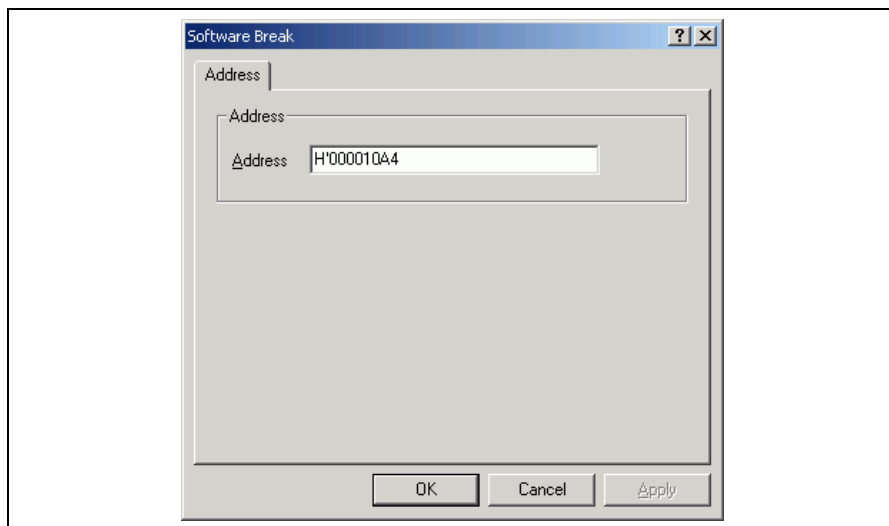

Figure 4.32   [Software Break] Dialog Box

- Use the [Editor] window to refer to the address on the line that has 'p_sam->s0=a[0];' within the `tutorial` function and enter this address in the [Address] edit box. In this example, enter *H'000010A4*.

Note: This dialog box differs depending on the product. For the items of each product, refer to section 3, Debugging, or the online help.

- Click the [OK] button. Then click the [Close] button on the [Breakpoint Properties] dialog box.

The software breakpoint that has been set is displayed in the [Event] window.
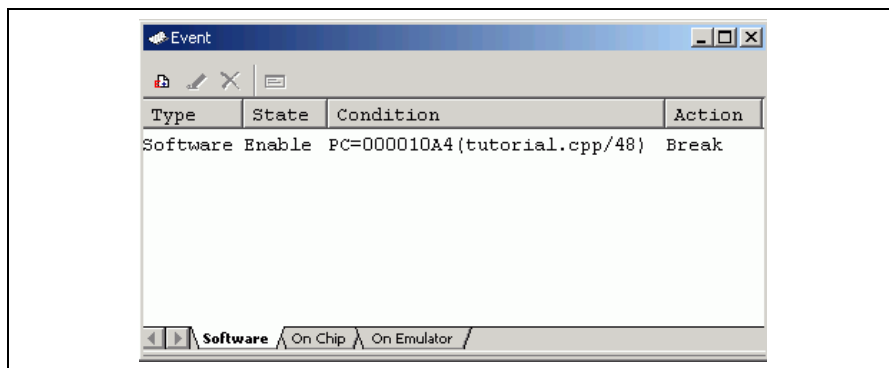
Figure 4.33   [Event] Window (Software Breakpoint Setting)

Note: The items that can be displayed in this window differ depending on the product. For the items that can
be displayed, refer to section 3, Debugging, or the online help.

- Close the [Event] window.
- To stop the tutorial program at the breakpoint, select [Reset Go] from the [Debug] menu.

RENESAS

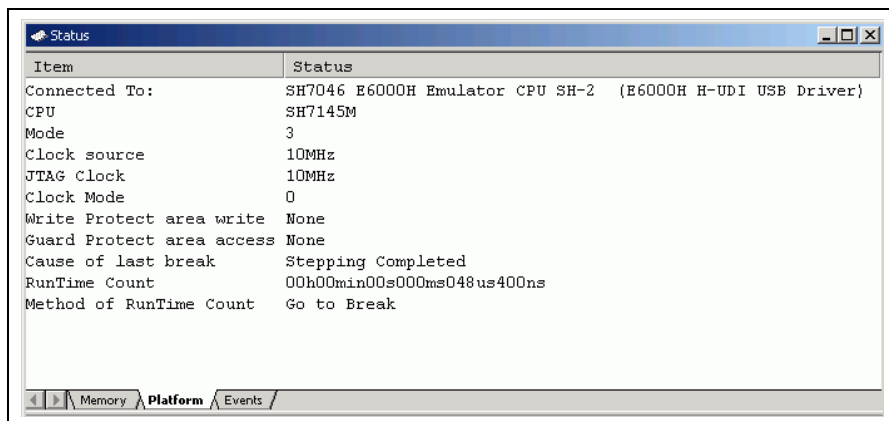The program runs until it stops at the breakpoint that has been set.
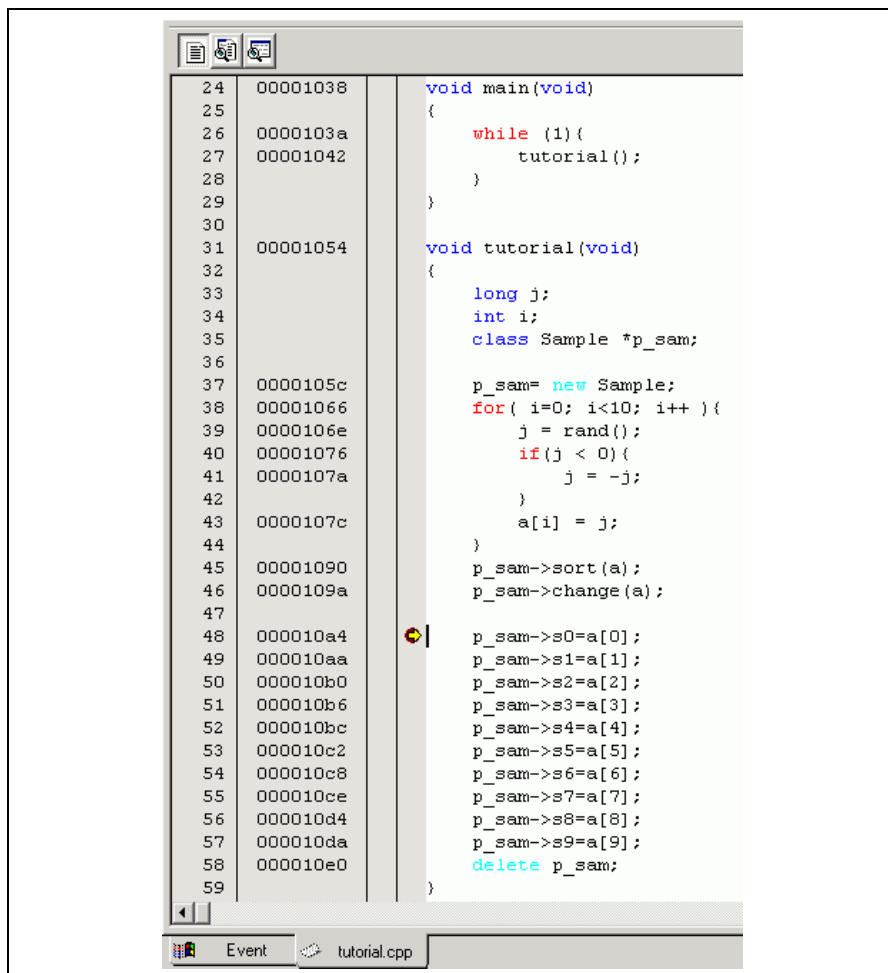
Figure 4.34   [Editor] Window at Execution Stop (Software Break)

RENESAS

The [Status] window displays the following contents:

Figure 4.35   Displayed Contents of the [Status] Window (Software Break)

Note: The items that can be displayed in this window differ depending on the product. For the items that can be displayed, refer to section 3, Debugging, or the online help.

4.15.2 On-Chip Break Function

Setting of an on-chip breakpoint on channel 4 such that a break is triggered when the break condition has been satisfied five times is explained as an example of the use of on-chip breakpoints.

Note: The channels on which the satisfaction count can be specified differ depending on the product. For details on each product, refer to section 3, Debugging, or the online help.

- Select [Eventpoints] from the [Code] submenu of the [View] menu or click the [Eventpoints] toolbar button (![EP]). The [Event] window is displayed.
- The software breakpoint that has been previously set must be deleted. Click the [Software] sheet of the [Event] window with the right-hand mouse button and select [Delete All] from the popup menu to delete all the software breakpoints that have been set.
- Click the [On Chip] tab of the [Event] window.
- Click the [Event] window with the right-hand mouse button and select [Add…] from the popup menu.
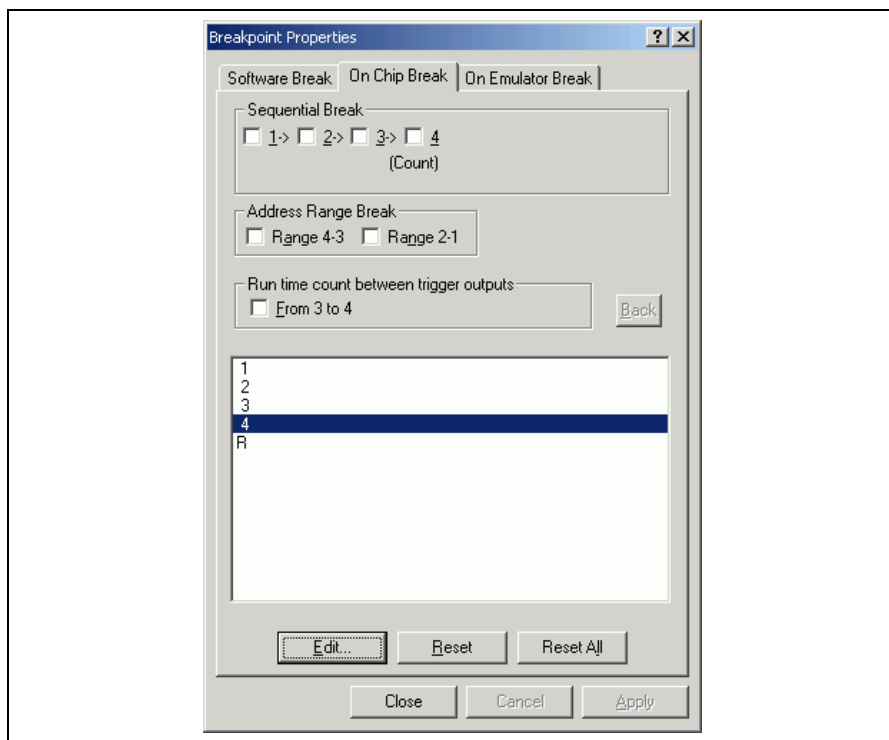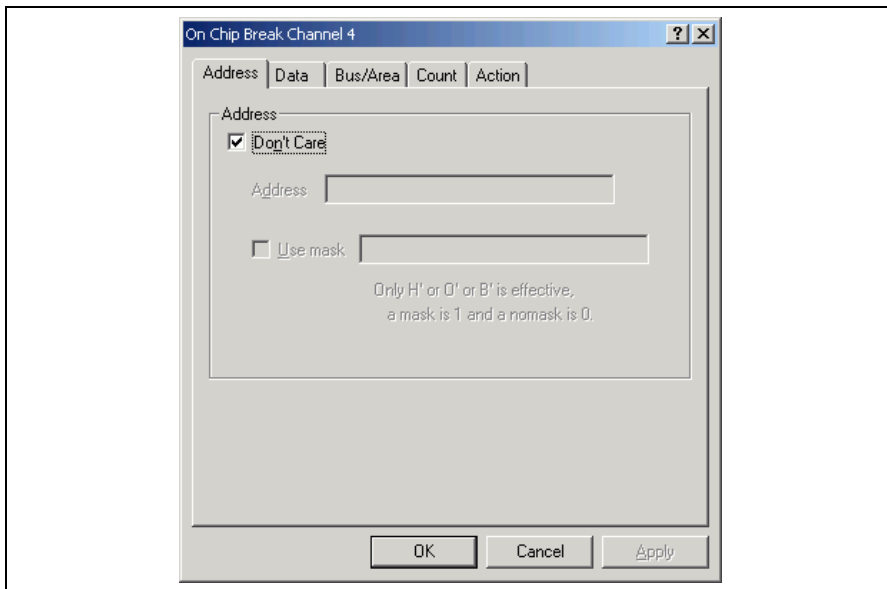- The [Breakpoint Properties] dialog box ([On Chip Break] page) is displayed.

Figure 4.36   [Breakpoint Properties] Dialog Box ([On Chip Break] Page)

- Select [4] in the list box and click the [Edit…] button. The [On Chip Break Channel 4] dialog box is displayed.

RENESAS

- Make the following settings in the group boxes on the [Address] page:
  Uncheck the [Don't Care] checkbox.
  Then use the [Editor] window to refer to the address on the line that has 'a[i]=j;' within the `tutorial` function and enter this address in the [Address] edit box. In this example, enter *H'0000107C*.
- Make the following settings in the boxes on the [Count] page:
  Uncheck the [Don't Care] checkbox.
  Enter *D' 5* in the [Count] edit box.

Note: The content of this dialog box differs depending on the product. For details on each product, refer to section 3, Debugging, or the online help.

Figure 4.37   [On Chip Break Channel 4] Dialog Box

- Click the [OK] button. Then click the [Close] button on the [Breakpoint Properties] dialog box.

The on-chip breakpoint that has been set is displayed in the [Event] window.

Note: The items that can be displayed in this window differ depending on the product. For the items that can be displayed, refer to section 3, Debugging, or the online help.

Close the [Event] window. Then select [Reset Go] from the [Debug] menu to stop the tutorial program at on-chip breakpoints. The program runs and then stops at the breakpoint that has been set. The cause of a break can be checked in [Cause of last break] of the [Status] window.

Refer to the [Watch] window for the value of variable `i`. The value is 4, indicating that the break occurred after the condition had been satisfied five times.

Then delete the on-chip breakpoint. Clicking the right-hand mouse button on the [Event] window displays a popup menu. Select [Delete All] from this menu to delete all the on-chip breakpoints.

RENESAS

4.16 Trace Functions

The trace functions of the emulator use the realtime trace buffer, which can store the information of up to 128-k bus cycles. The content of this buffer, which is constantly updated during execution, is displayed in the [Trace] window.

Select [Trace] from the [Code] submenu of the [View] menu or click the [Trace] toolbar button () to display the [Trace] window.
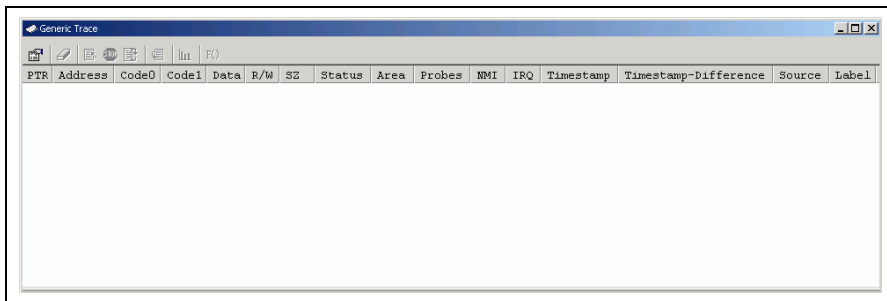
Figure 4.38   [Trace] Window

When trace information is displayed in the [Trace] window, clicking the right-hand mouse button on the [Trace] window displays a popup menu. Select [Clear] from this menu to clear the trace information.

The following sections give an overview of the trace functions and the settings.

4.16.1 Displaying Trace Information by the Free Trace Function

The free trace function allows continuous acquisition of trace information from the start of user program execution to the occurrence of a break.

(1) All break conditions must be deleted. Clicking the right-hand mouse button on the [Trace] window displays a popup menu. Select [Acquisition…] from this menu to display the [Trace Acquisition Properties] dialog box. Ensure that [Free Trace] is checked and then click the [Close] button.
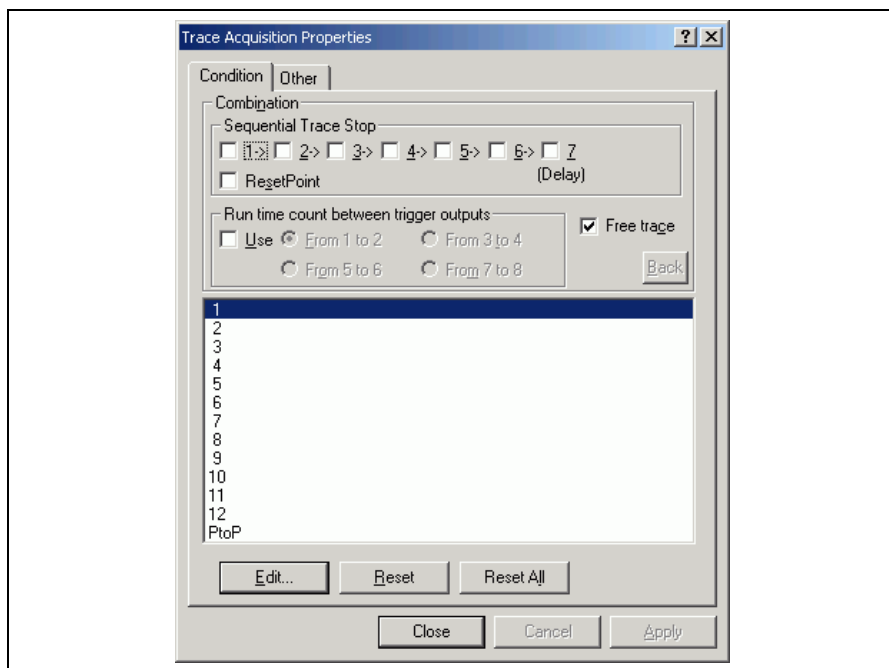

Figure 4.39   [Trace Acquisition Properties] Dialog Box (Free Trace)

RENESAS

(2) Set a software breakpoint at the address on the line that has 'p_sam->s0=a[0];' within the `tutorial` function (refer to section 4.15.1, Software Break Function).

(3) Select [Reset Go] from the [Debug] menu. Execution stops when the break condition is satisfied, and the [Trace] window then displays the trace information.

Figure 4.40 [Trace] Window (Free Trace)

RENESAS

4.16.2 Displaying Trace Information by the Trace Stop Function

While the trace stop function is in use, acquisition of trace information stops when a specified condition is satisfied. The user can check the program flow by the trace information without breaking the user program execution.

(1) Delete all the break conditions that have been set. Uncheck [Free Trace] on the [Condition] page of the [Trace Acquisition Properties] dialog box (otherwise, the free trace mode will be selected).
(2) Select [1] from the list box on the [Condition] page of the [Trace Acquisition Properties] dialog box and then click [Edit…]. The [Trace Acquisition Condition Channel 1] dialog box is displayed. Select the [Trace Stop] radio button in the [After Condition Match] group box on the [Action] page.
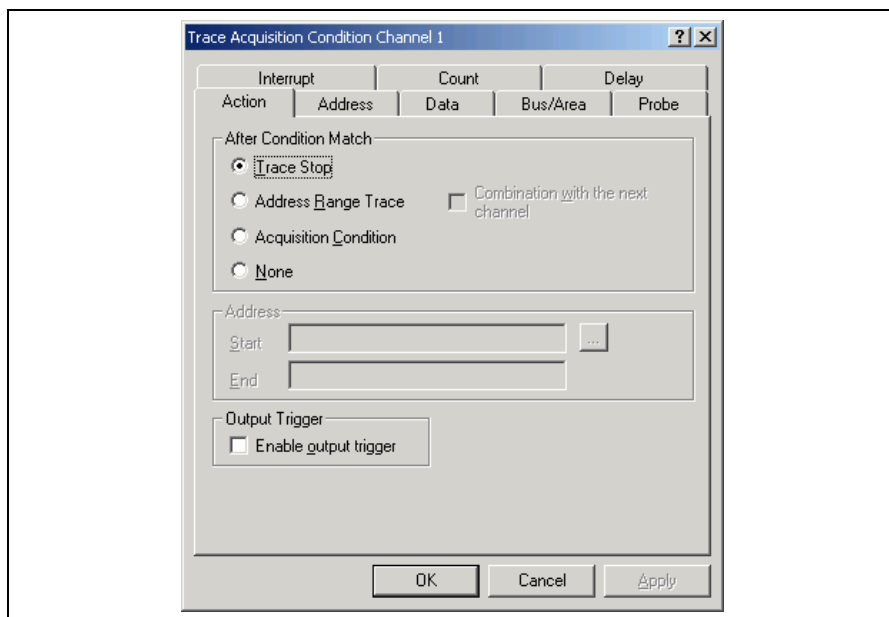
Figure 4.41 [Trace Acquisition Condition Channel 1] Dialog Box (Trace Stop)

(3) An address must be set as the condition. Uncheck [Don't Care] on the [Address] page of the [Trace Acquisition Condition Channel 1] dialog box. Then use the [Editor] window to refer to the address on the line that includes 'a[i]=j;' within the `tutorial` function and enter this address in the [Start] edit box. In this example, enter *H'0000107C*. This completes the setting of the address. Click the [OK] button to close the [Trace Acquisition Condition Channel 1] dialog box.
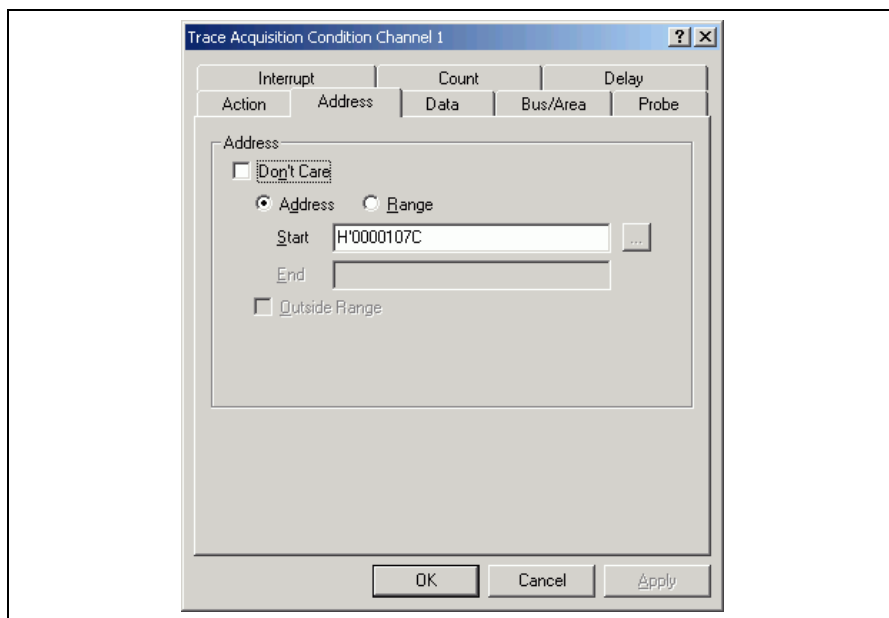
Figure 4.42   [Trace Acquisition Condition Channel 1] Dialog Box ([Address] Page)

(4) Items that have been set are displayed in the list box on the [Condition] page of the [Trace Acquisition Properties] dialog box. Click the [Close] button on this dialog box.
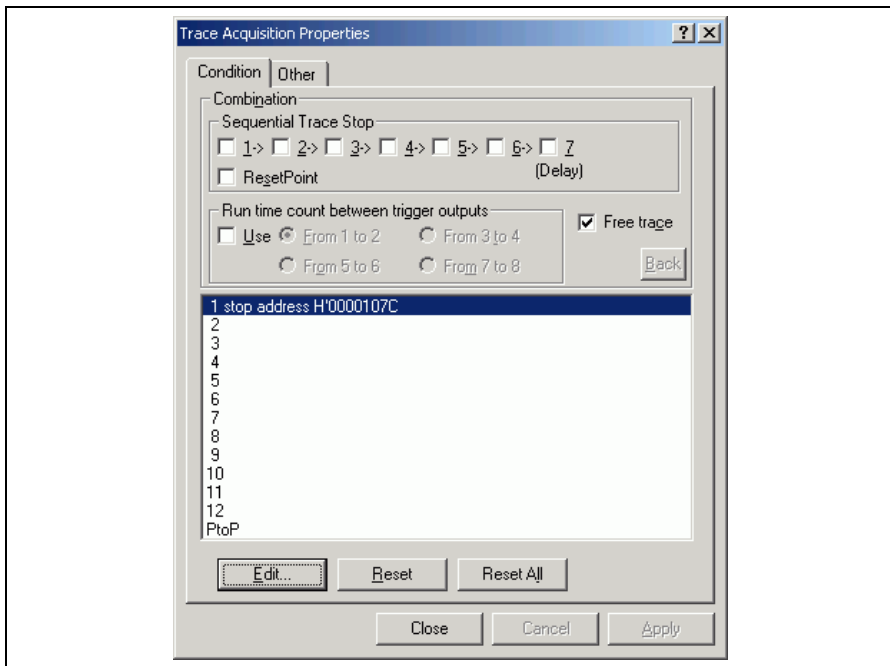
Figure 4.43   [Trace Acquisition Properties] Dialog Box (Trace Stop)

(5) Select [Reset Go] from the [Debug] menu. The trace condition is satisfied, and the [Trace] window then displays the following contents.

Figure 4.44   [Trace] Window (Trace Stop)

RENESAS

4.16.3 Displaying Trace Information by the Conditional Trace Function

The conditional trace function only acquires trace information at the address where a specified condition has been satisfied. This is useful for analyzing a program focused on reading from or writing to a specific address (e.g. a global variable or memory mapped I/O).

(1) If the user program is running, select [Halt Program] from the [Debug] menu to halt the program.

(2) Delete all the break conditions that have been set. Uncheck [Free Trace] on the [Condition] page of the [Trace Acquisition Properties] dialog box (otherwise, the free trace mode will be selected).

(3) Select [1] from the list box on the [Condition] page of the [Trace Acquisition Properties] dialog box and then click [Edit…]. The [Trace Acquisition Condition Channel 1] dialog box is displayed. Select the [Acquisition Condition] radio button in the [After Condition Match] group box on the [Action] page.

(4) An address must be set as the condition. Uncheck [Don't Care] on the [Address] page of the [Trace Acquisition Condition Channel 1] dialog box. Then use the [Watch] window to refer to the address on the line that includes 'a[0]' and enter this address in the [Start] edit box. In this example, enter *H'FFFFE400*. This completes the setting of an address. Click the [OK] button to close the [Trace Acquisition Condition Channel 1] dialog box.

(5) Items that have been set are displayed in the list box on the [Condition] page of the [Trace Acquisition Properties] dialog box. Click the [Close] button on this dialog box.
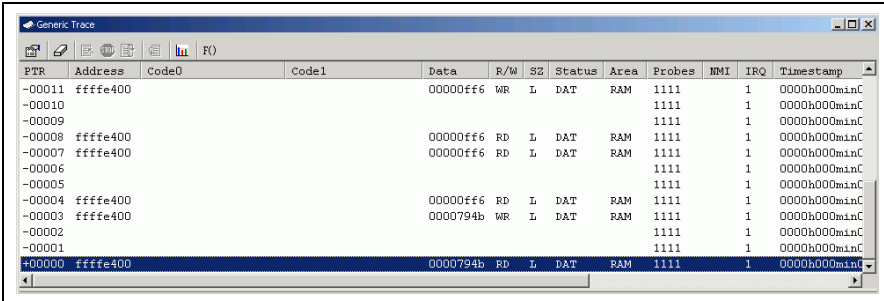
(6) Set a software breakpoint at the address on the line that has 'delete p_sam;' within the `tutorial` function (*H'000010E0.* in this example) (for details, refer to section 4.15.1, Software Break Function).

(7) Select [Reset Go] from the [Debug] menu. Execution stops when the break condition is satisfied, and the [Trace] window then displays the following contents.

Figure 4.45   [Trace] Window (Conditional Trace)

RENESAS

4.16.4　Statistics

The number of times the on-chip RAM has been written to can be included in the acquired trace information.

(1) Delete all the break conditions that have been set. Click [Reset All] on the [Condition] page of the [Trace Acquisition Properties] dialog box to cancel trace conditions. Check [Free Trace] on the [Condition] page of the [Trace Acquisition Properties] dialog box.

(2) Make the setting so that a break occurs at the address on the line that has 'p_sam->s0=a[0];' within the `tutorial` function (***H'000010A4*** in this example) (for details on, refer to section 4.15.1, Software Break Function).

(3) Select [Reset Go] from the [Debug] menu. Execution stops when the break condition is satisfied, and the [Trace] window then displays the trace information.

(4) Select [Statistic…] from the popup menu that is displayed when you click the right-hand mouse button on the [Trace] window. A message box appears, indicating that the trace data is being loaded, and the [Statistic] dialog box will be displayed.
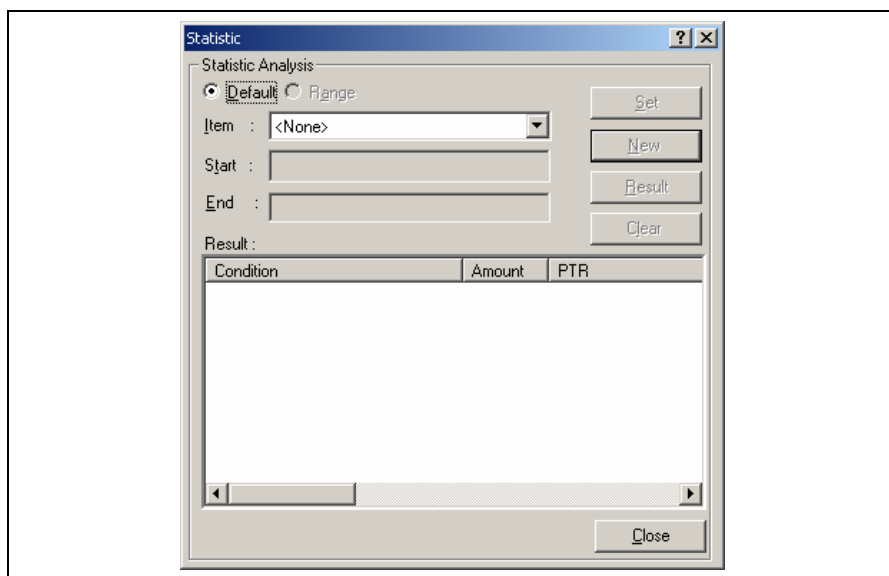
Figure 4.46　[Statistic] Dialog Box

(5) Select [R/W] in the [Item] combo box and enter *WR* in the [Start] edit box. Then, click the [New] button. "`R/W=WR`" will be displayed in the [Condition] column of the [Result] list box.
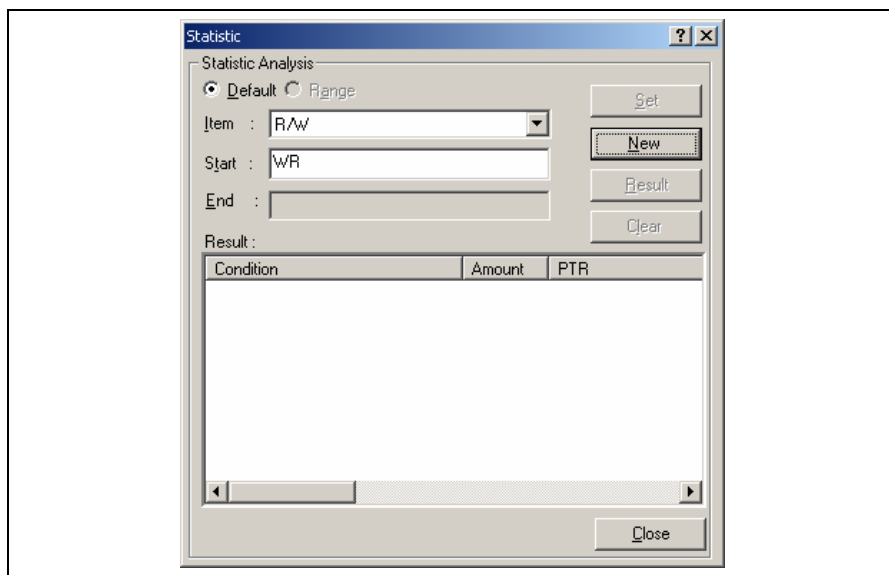
Figure 4.47   [Statistic] Dialog Box (New Condition)

(6) Then, select [Area] from the [Item] combo box and enter **RAM** in the [Start] edit box. Then, click the [Add] button; the new condition is now added to the "R/W=WR" display in the [Condition] column of the [Result] list box, so that it now shows "R/W=WR & Area=RAM". This completes the setting of the conditions.
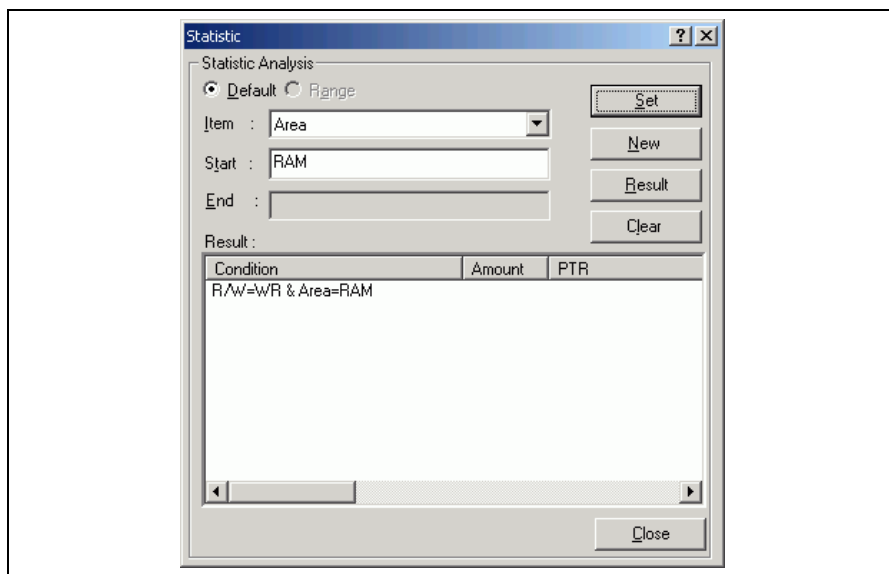
Figure 4.48   [Statistic] Dialog Box (Condition Added)

(7) To start statistical analysis of the specified condition, press the [Result] button. The number of write operations that satisfies the conditions and the PTR values will be displayed.
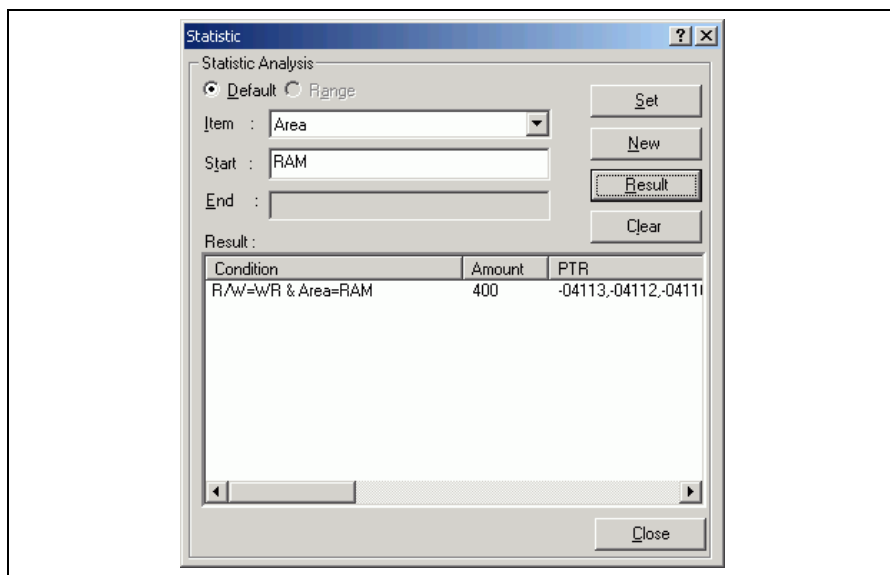
Figure 4.49   [Statistic] Dialog Box (Result of Analysis)

(8) Click the [Close] button to close the [Statistic] dialog box.

(9) Delete the event points that have been set and clear the trace information. Clicking the right-hand mouse button on the [Event] window displays a popup menu. Select [Delete All] from this menu to delete all the event points that have been set. Clicking the right-hand mouse button on the [Trace] window displays a further popup menu. Select [Clear] from this menu to clear the trace information.

4.16.5 Function Calls

This mechanism is only used to collect trace information on the function calls.

(1) Make the setting so that a break occurs at the address on the line that has 'p_sam->s0=a[0];' within the `tutorial` function (**H'000010A4** in this example) (for details, refer to section 4.15.1, Software Break Function).

(2) Select [Reset Go] from the [Debug] menu. Execution stops when the break condition is satisfied, and the [Trace] window then displays the trace information.

(3) Select [Function Call…] from the popup menu displayed by clicking the right-hand mouse button on the [Trace] window. The [Function Call Display] dialog box will be displayed.
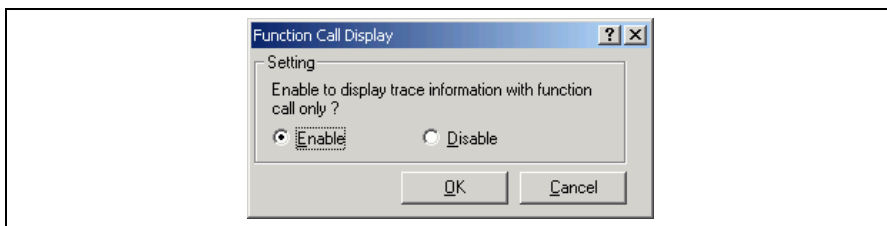
Figure 4.50 [Function Call Display] Dialog Box

(4) Click the [Enable] radio button and then the [OK] button. Only the information on function calls is now displayed in the [Trace] window.
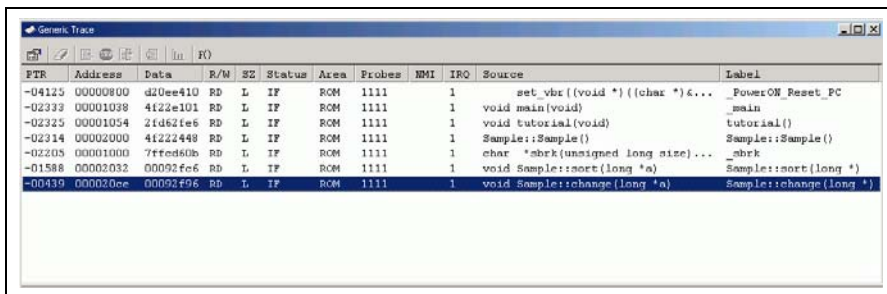
Figure 4.51 [Trace] Window (Function Calls)

(5) To return the display in the [Trace] window to its previous state, follow the procedure in (3) to display the [Function Call Display] dialog box. Click the [Disable] button and then the [OK] button.

(6) Delete the event points that have been set and clear the trace information. Clicking the right-hand mouse button on the [Event] window displays a popup menu. Select [Delete All] from this menu to delete all the event points that have been set. Clicking the right-hand mouse button on the [Trace] window displays a further popup menu. Select [Clear] from this menu to clear the trace information.

RENESAS

4.17 Stack Trace Function

The emulator uses the information on the stack to display the function call history.

Notes: 1. This function can be used only when the load module that has the Dwarf2-type debugging information is loaded. Such load modules are supported in H8S, H8/300 C/C++ compiler V4.0 or later.

2. For details on the stack trace function, refer to the online help.

- Double-click the [S/W Breakpoints] column in the sort function and set a software breakpoint.
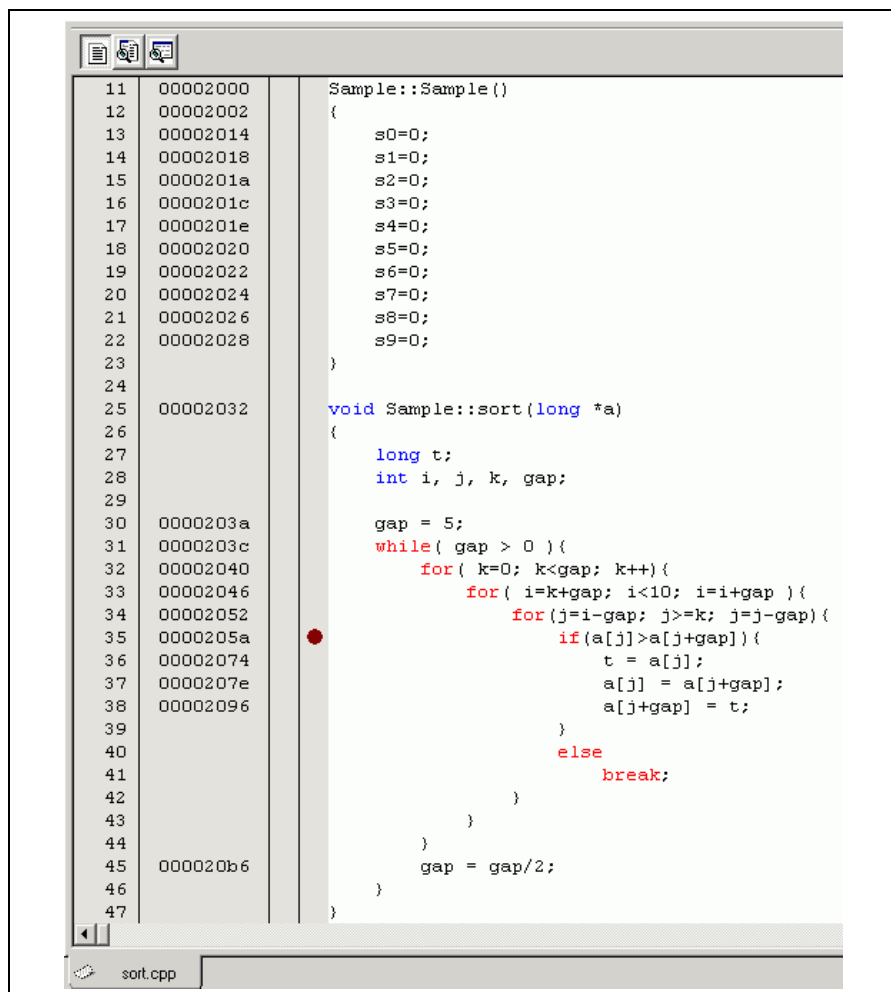
Figure 4.52 [Editor] Window (Software Breakpoint Setting)

- Select [Reset Go] from the [Debug] menu.
- After the break in program execution, select [Stack Trace] from the [Code] submenu of the [View] menu to open the [Stack Trace] window.
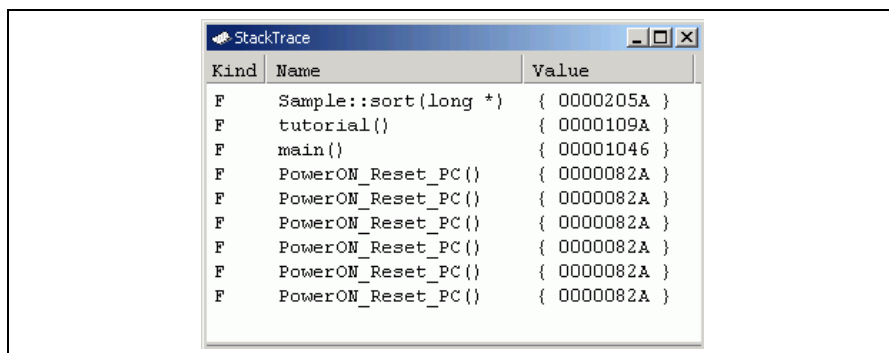
Figure 4.53   [Stack Trace] Window

Figure 4.53 shows that the position of the program counter is currently at the selected line of the `sort()` function, and that the `sort()` function is called from the `tutorial()` function.

To delete the software breakpoint, double-click the [S/W Breakpoints] column in the `sort` function again.

4.18 Performance Analysis Function

Performance analysis by the emulator is available in the following modes:

- Time Of Specified Range Measurement
- Start Point To End Point Measurement
- Start Range To End Range Measurement
- Access Count Of Specified Range Measurement
- Called Count Of Specified Range Measurement

In this tutorial, we describe the Time Of Specified Range Measurement.

4.18.1 Time Of Specified Range Measurement

(1) Select [Performance Analysis] from the [Performance] submenu of the [View] menu to display the [Select Performance Analysis Type] dialog box.
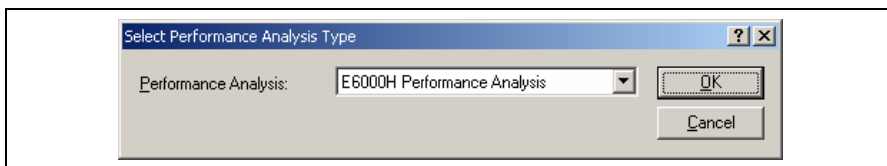
Figure 4.54 [Select Performance Analysis Type] Dialog Box

(2) Select "E6000H Performance Analysis" from the [Performance Analysis] combo box in the [Select Performance Analysis Type] dialog box and click the [OK] button. The [Performance Analysis] window will be displayed.
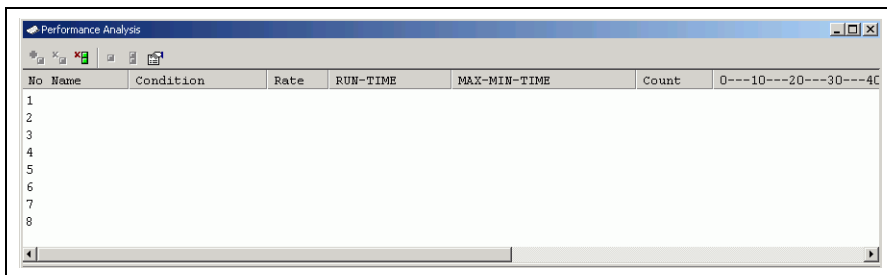
Figure 4.55 [Performance Analysis] Window

RENESAS

(3) Select the line in the [Performance Analysis] window that has 1 in its [No] column and click the right-hand mouse button to display a popup menu. Select [Set…] from this popup menu to display the [Performance Analysis Properties] dialog box.
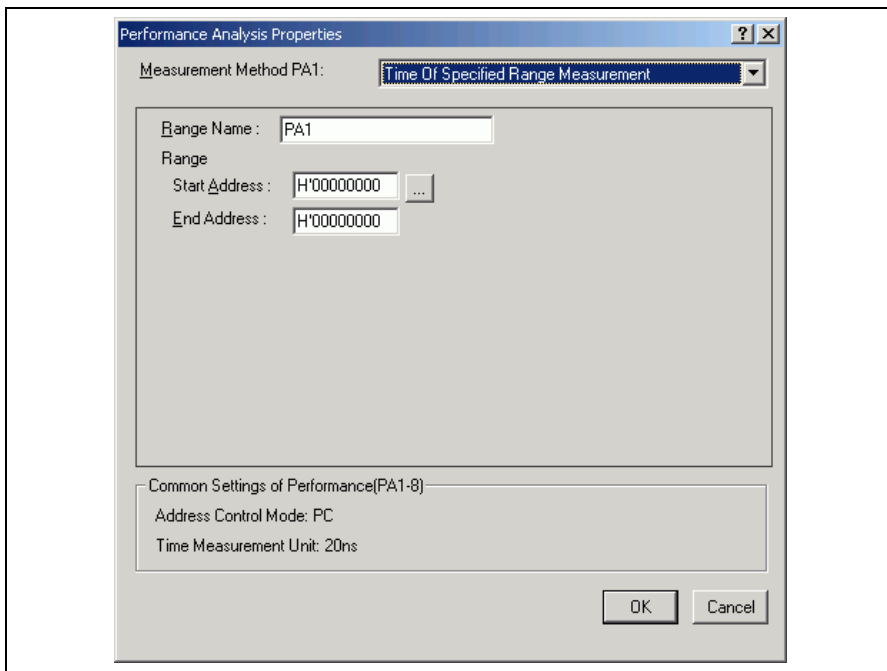
Figure 4.56 [Performance Analysis Properties] Dialog Box

(4)Select `Time Of Specified Range Measurement` from the [Measurement Method PA1] combo box.

(5) The parameter settings are as follows.

- Enter **_sort_** in the [Range Name] edit box.
- Click the […] button on the right of the [Start Address] edit box to display the [Input Function Range] dialog box. Enter the function name **_sort_** in the [Function] edit box in this dialog box and then click the [OK] button. The addresses for the function `Sample::sort(long*)` will now be set in the [Start Address] and [End Address] edit boxes.
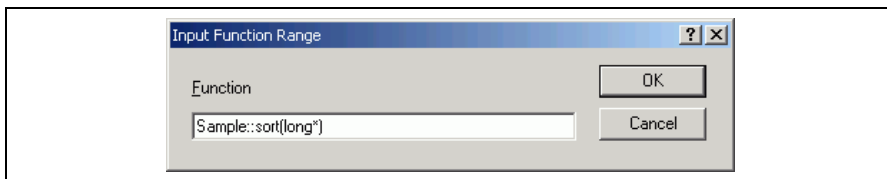
Figure 4.57 [Input Function Range] Dialog Box

RENESAS

Note: The addresses figured out in the [Input Function Range] dialog box are just for reference. In some cases, the end address of a function may be incorrect. Check the last instruction of the function in the [Disassembly] window to correct the value set in [End Address] so that it will be the address of the last instruction (in general, the last instruction of a function is a RTS instruction). A label name or an expression can be entered instead of an address value in boxes where an address should be entered.

(6) Click the [OK] button to display the contents that has been set for line 1 of the [No] column in the [Performance Analysis] window. This completes the settings for measuring the time within the specified range.
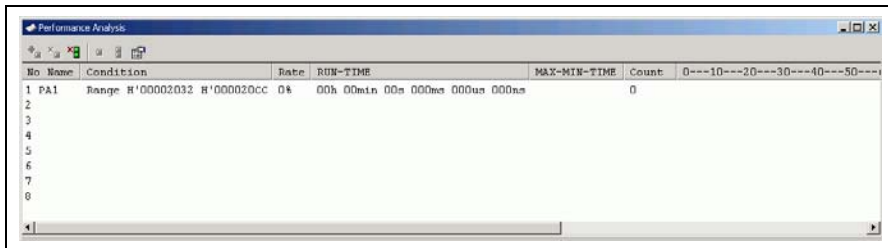
Figure 4.58 [Performance Analysis] Dialog Box (Setting Completed)

(7) Set a software breakpoint at the address on the line that has 'p_sam->change(a);' within the `tutorial` function (`H'0000109A` in this example). Refer to section 4.15.1, Software Break Function.

(8) Select [Reset Go] from the [Debug] menu. Execution stops when the break condition is satisfied, and the [Performance Analysis] window then displays the information shown below. The value shown in the [Count] column is 1, which indicates that the `sort` function has been executed once, and the execution time is also displayed. In this tutorial, the minimum unit for time measurement is defined as 20 ns. This value can be changed in the [Configuration Properties] dialog box.
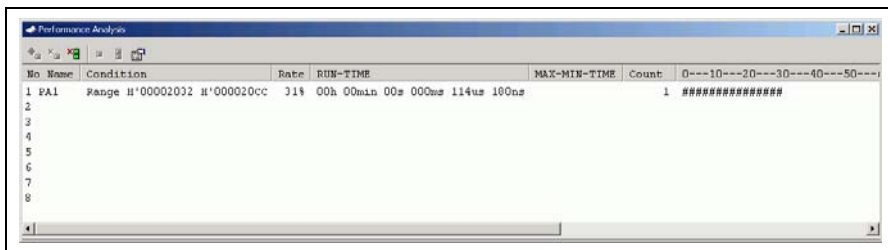
Figure 4.59 [Performance Analysis] Dialog Box (Displaying the Result)

(9) Delete the settings for performance analysis and delete the event points. Click the right-hand mouse button on the [Performance Analysis] window to display a popup menu. Select [Reset All] from this popup menu to clear all of the settings. Clicking the right-hand mouse button on the [Event] window also displays a popup menu. Select [Delete All] from this popup menu to delete all the event points that have been set.

RENESAS

4.19 Monitor Function

The emulator allows monitoring of the contents of specified addresses in memory during execution of the user program. In this example, we monitor the content of the address range where variable a of the `tutorial` function is stored.

(1) Select the [CPU] submenu from the [View] menu. Selecting [Monitor Setting…] from the [Monitor] submenu displays the [Monitor Setting] dialog box.
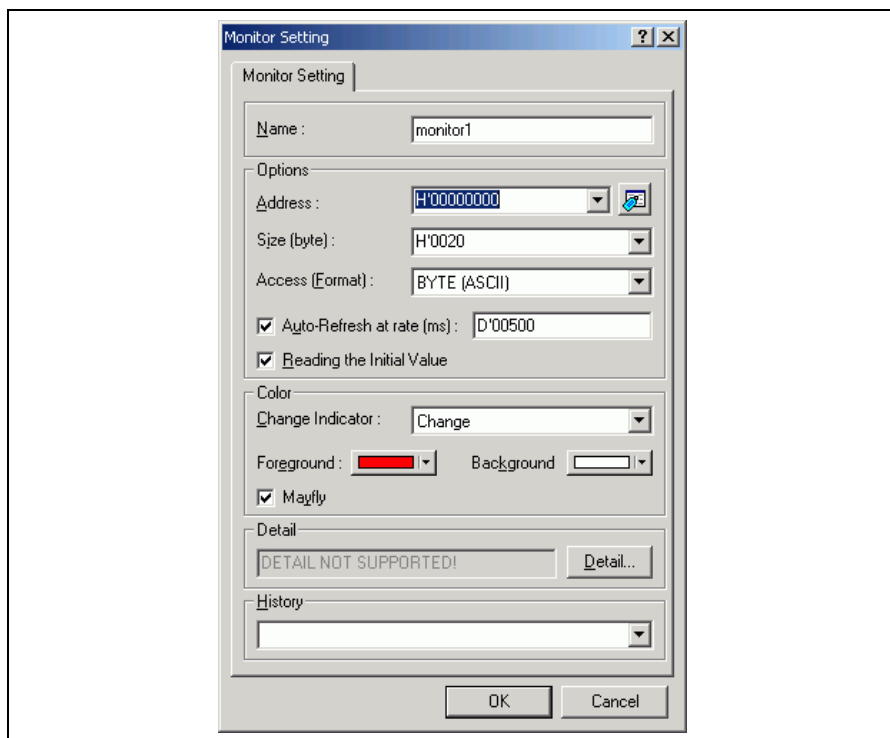
Figure 4.60   [Monitor Setting] Dialog Box

(2) Set the items in the [Monitor Setting] dialog box as follows:

- Enter **monitor1** in the [Name] edit box.
- Set the parameters in the [Options] group box as follows:
  - (a) Use the [Watch] window to refer to the address on the line where variable a, which is defined within the tutorial function, is allocated and enter this address in the [Address] edit box. In this example, **H'FFFFE400** is entered.
  - (b) Enter **H'20** in the [Size (byte)] combo box.
  - (c) Select BYTE (ASCII) from the [Access (Format)] combo box.
  - (d) Check the [Auto-Refresh at rate (ms)] check box and enter **D'00500** in the edit box.
  - (e) Check the [Reading the Initial Value] check box.
- Set the parameters in the [Color] group box as follows:
  - (a) Select Change from the [Change Indicator] combo box.
  - (b) Select red and white in the [Foreground] and [Background] combo boxes, respectively.
  - (c) Check the [Mayfly] check box.

Note: Depending on the operating system in use, the foreground and background colors may not be selectable.
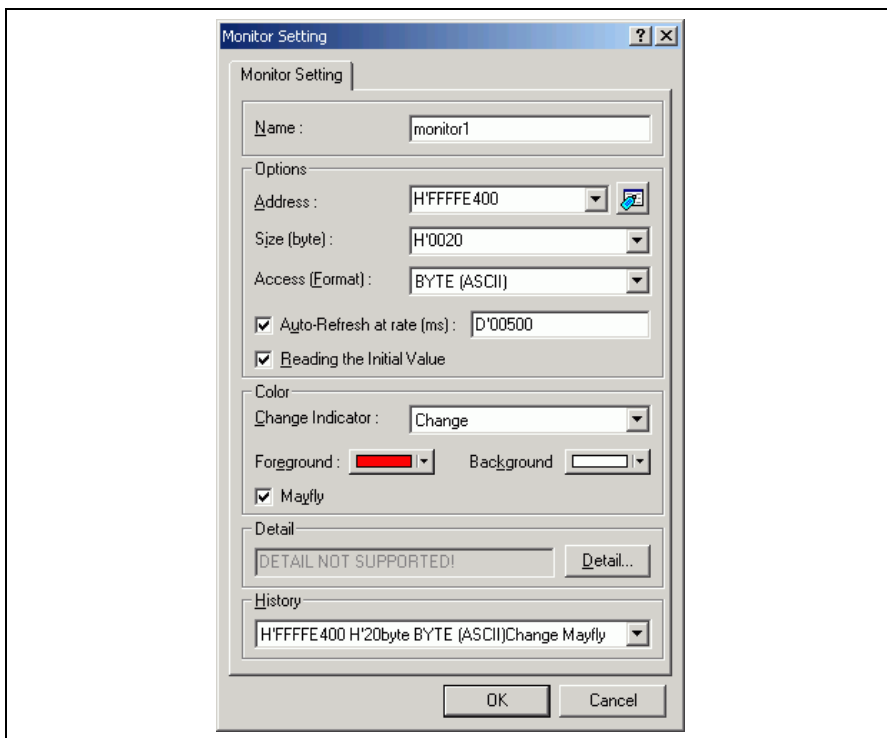
Figure 4.61   [Monitor Setting] Dialog Box (Setting Completed)

RENESAS

(3) Click the [OK] button to open the [Monitor] window.
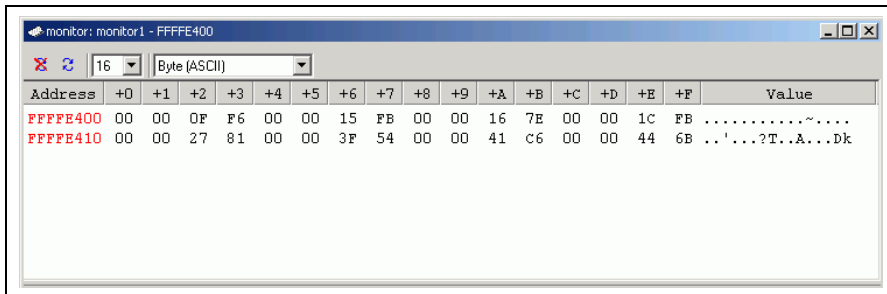
Figure 4.62   [Monitor] Window

(4) Select [Reset Go] from the [Debug] menu. When the contents of the address range changes by execution, the updated values are in red (i.e. the color that was selected in the [Foreground] and [Background] combo boxes). Values will be displayed in black if they have not been updated or a certain period of time has elapsed since the last update.
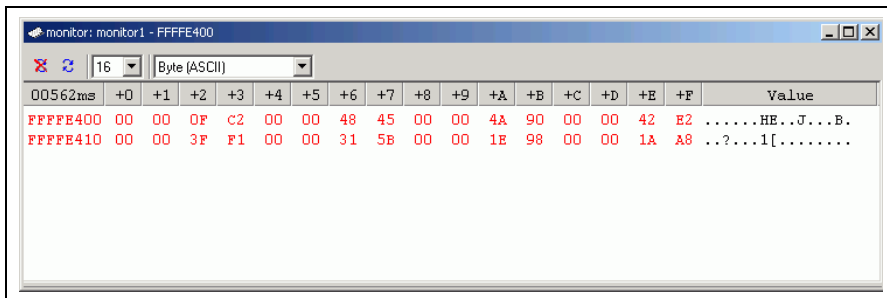
Figure 4.63   [Monitor] Window (during Execution)

(5) After you have finished checking the states in the [Monitor] window, select [Halt Program] from the [Debug] menu to halt the program's execution.

4.20   What Next?

In this tutorial, some of the main features of the emulator and the High-performance Embedded Workshop operation have been given. By using the emulation functions provided by the emulator, a high-level debugging is possible. The conditions caused by hardware and software can be accurately classified and the users can investigate the problems effectively.

RENESAS

Section 5   Software Specifications and Notes Specific to This Product

This section describes the software specifications and notes specific to the E6000H emulator.

5.1   Supported Hardware

This emulator software is specialized for the SH7046 E6000H emulator (HS7046EPH60H).

5.2   Debugging Platform

The following debugging platform can be selected in this emulator. The target MCUs to be emulated depend on the selected debugging platform.

Table 5.1   Selectable Target Platform

Debugging Platform	Description
SH7046 E6000H Emulator CPUSH-2	The SH7046, SH7047, SH7144, or SH7145 group microcomputers can be emulated.

5.3   Displaying and Modifying the Contents of Memory

5.3.1   Displaying and Modifying the Contents of Memory during Execution

The emulator accesses memory in the following three ways to display and modify the contents of memory during user program execution.

Table 5.2   Access Types for Displaying and Modifying Contents of Memory

Access Type	Description	Period Suspended	Display	Modification
Monitor function	Automatically updates the display of the memory contents without stopping the user program execution	None	Enabled	Disabled
Parallel access function	Temporarily stops the user program execution	Short	Enabled	Enabled
Short break function	Temporarily stops the user program execution	Long	Enabled	Enabled

These access types have the following characteristics.

Table 5.3   Characteristics of Displaying and Modifying Contents of Memory

Access Type	Target Window	Target Memory Area
Monitor function	Realtime update of display in the [Monitor] window and in the [Watch] window when the monitor function is used	Specified eight points or up to 256 bytes of the areas that the user program is allowed to access
Parallel access function	Windows that display the memory contents other than the [Monitor] window	On-chip ROM and emulation memory area
Short break function	Windows that display the memory contents other than the [Monitor] window	On-chip RAM, on-chip I/O, and user memory area

Notes: 1. If the setting in the [Watch] window allows a realtime update of a value by reading the normal data, access types will be different depending on the target memory area that has been set.
2. The parallel access or short break function cannot be used to display or modify the memory contents while in the sleep mode or standby mode.
3. When the parallel access function or short break function is in use, the program is not executed in realtime.

5.3.2 Reference Values for Parallel Access Function Termination Period

Table 5.4 shows the reference value for displaying and modifying the memory contents during user program execution.

Table 5.4   Suspension of Program Execution: Reference Values

Access Type	Condition		Period Suspended
Short break	Read	Displaying 256 bytes in the on-chip RAM	3.5 ms
		Reading a longword from the on-chip RAM	1.8 ms
	Write	Writing a longword to the on-chip RAM	1.8 ms
Parallel access	Read	Reading a longword from the emulation memory	220 ns
	Write	Writing a longword to the emulation memory	220 ns
Auto update Memory	Execution of the user program does not stop		

Table 5.5 shows the environment in which these measurements were obtained.

Table 5.5   Measurement Environment

Item		Setting
Settings of E6000H	System clock (ϕ)	10.0 MHz
	Clock mode	Clock mode 3
	JTAG clock	10.0 MHz
Host computer	CPU	Pentium® IV 1.5 GHz
	PC interface	PCI interface

5.3.3 Monitor Function

- Up to eight points or 256 bytes in total can be specified for the monitor function.
- The monitor function is implemented by eight 32-byte hardware channels. The address range specified for one channel must be aligned to a 32-byte boundary; two channels should be used to specify a range across a 32-byte boundary. Accordingly, when multiple ranges are specified across 32-byte boundaries, the total specifiable size is less than 256 bytes.
- When monitor function conditions are set or modified during user program execution, the program is not executed in realtime.
- When [Access Size and Display Format] are modified during user program execution, the program is not executed in realtime.

RENESAS

5.4　Executing Your Program

5.4.1　Step Execution

Break conditions are ignored during step execution, but trigger pulses will be output.

5.5　Event Functions

5.5.1　Software Breakpoints

- A software breakpoint is realized by replacing the instruction at the specified address with a special instruction. Accordingly, it can only be set to the area including the emulation RAM. Note that it cannot be set to the following addresses:
  - — Addresses whose memory content is H'0000
- Do not modify the contents of the software breakpoints addresses by the user program.
- The content of a software breakpoint address is replaced by a break instruction during user program execution.
- The maximum number of software breakpoints and temporary PC breakpoints in [Temporary PC Breakpoints] of the [Run Program] dialog box is 255 in total. Therefore, when 255 software breakpoints have been set, no temporary breakpoint set in [Temporary PC Breakpoints] of the [Run Program] dialog box is valid. Ensure that the total number of software breakpoints and temporary PC breakpoints are 255 or less.
- Do not set a breakpoint immediately after a delayed branch instruction (at a slot instruction). If this is attempted, a slot illegal instruction interrupt will occur when the delayed branch instruction is executed, and the break will not occur.

5.5.2　On-Chip Break

- The satisfaction count can only be set for channel 4.
- A reset point is only valid when a sequential break is enabled.
- If a sequential break and a break range ([Address Range Break]) have been specified together, only a sequential break is enabled.
- The address and data conditions are satisfied on the bus cycles where the values on the address bus or data bus match. Consider the following points when setting these conditions.
  - — Longword access

    Longword data is read and written in a single bus cycle. A data condition is only valid for a longword access when specified as longword. The specified address must be a multiple of four. Note that longword data is only valid as the size of an access.
  - — Word access

    Word data is read and written in a single bus cycle. The specified address must be a multiple of two. Word data is only valid as the size of an access.
  - — Byte access

    Byte data is read and written in a single bus cycle. A data condition is only valid for a byte access when specified as byte. Any address condition, both an even and odd address, are valid.

5.5.3 On-Emulator Break

- A break will occur several cycles after a condition is satisfied.
- The address bus and data bus conditions are satisfied on the bus cycles where the values on the address bus or data bus match. Consider the following points when setting these conditions.
    - 32-bit bus area
        - Longword access

          Longword data is read and written in a single bus cycle. A data condition is only valid for a longword access when specified as longword. An address condition is only valid for a longword access when specified as a multiple of four.
        - Word access

          Word data is read and written in a single bus cycle. A data condition is only valid for a word access when specified as word. An address condition is only valid for a word access when specified as a multiple of two.
        - Byte access

          Byte data is read and written in a single bus cycle. A data condition is only valid for a byte access when specified as byte. Any address condition, whether an even or odd address, is valid.
    - 16-bit bus area
        - Longword access

          Longword data is read and written in two bus cycles. A data condition is only valid for a longword access when specified as word. An address condition is only valid for a longword access when specified as a multiple of two.
        - Word access

          Word data is read and written in a single bus cycle. A data condition is only valid for a word access when specified as word. Any multiple of two is a valid address condition.
        - Byte access

          Byte data is read and written in a single bus cycle. A data condition is only valid for a byte access when specified as byte. Any address condition, whether an even or odd address, is valid.
    - 8-bit bus area

      All accesses are done in bytes in this area (a longword is accessed in four byte cycles, and a word in two byte cycles). Any address condition, whether an even or odd address, is valid. A data condition is only valid when specified as byte.

RENESAS

5.6 Trace Functions

5.6.1 Displaying the Trace Information

- The same hardware resource is used for acquisition of time stamps and the IRQ signals in the trace function. Accordingly, the trace contents differ depending on the [Selection of the trace contents] setting in the [Trace Acquisition Properties] dialog box.

Table 5.7 Trace Contents

Setting	Description
When [Timestamp] is selected	Acquires and displays timestamp in 32 bits. The IRQ signals are not acquired.
When [IRQ7-0 all indications] is selected	Acquires and displays the IRQ signals. The time stamp is displayed with the lower 16 bits fixed to 0.

- When the user clock (system clock signal φ) has been selected for [Time measurement unit] in the [Trace acquisition] dialog box, the time stamp is displayed for 32 bits in hexadecimal.

5.6.2 Specifying Trace Acquisition Conditions

- The trace will stop several cycles after a condition is satisfied.
- Six or more bus cycles are required between pass points of sequential trace stop conditions and reset condition.
- Six or more bus cycles are required between the start and end of measurement when [1-2], [3-4], or [5-6] is specified in [Run time count between trigger outputs] of the [Trace Acquisition Properties] dialog box.
- Fifteen or more bus cycles are required between the start and end of measurement when [7-8] is specified in [Run time count between trigger outputs] of the [Trace Acquisition Properties] dialog box.
- Six or more bus cycles are required from the start of execution to satisfaction of a trace stop mode condition.
- A sequential break or a trace stop may be incorrect when the user program is executed after the specified address condition has been applied as the PC address to start execution.
- The Point to Point trace mode is not available when channel 1 is used for the performance analysis function.

5.6.3 Searching for a Trace Record

- When the range for searching is specified in the [General] page, a PTR value to end the search can be specified in the [Start PTR] option, and a PTR value to start the search can be specified in the [End PTR].
- When the user clock (i.e. system clock signal φ) has been selected in [Time measurement unit] of the [Trace Acquisition Properties] dialog box, no time stamp information will be searched.

5.6.4 Filtering Trace Records

- After the trace information is filtered, all trace information displayed in the [Trace] window is saved; a range for saving trace information cannot be specified. To save a specific range of trace information, the filter range must be specified in the [General] page of the [Trace Filter] dialog box.
- When the user clock (i.e. system clock signal φ) has been selected in [Time measurement unit] of the [Trace Acquisition Properties] dialog box, no time stamp information will be filtered.

5.7   Monitor Function

The foreground and background colors cannot be changed in some operating systems.

5.8   Performance Analysis Function

5.8.1   Errors

An error will be included in the measured performance as follows:

- ±one-resolution error (±20-ns error when the measurement resolution is 20 ns)
  This error may occur when the user program execution starts or stops (breaks) or when the measurement start or end condition is satisfied.
- Frequency stability of the crystal oscillating module for performance analysis: ±0.01%

5.8.2   Notes

- In all measurement modes, the interval between the end condition satisfaction and the next start condition satisfaction must be longer than one-measurement-resolution time. If the interval is shorter than that, the interval itself is included in the measured time.
- In [Time Of Specified Range Measurement], measurement stops when an instruction is fetched outside the specified range. In [Start Point To End Point Measurement] and [Start Range To End Range Measurement], measurement stops when the specified end condition is satisfied. When the same addresses are specified for these modes, the time measured in [Time Of Specified Range Measurement] is longer than that measured in [Start Point To End Point Measurement] or [Start Range To End Range Measurement].
- Execution time is measured by using address bus values in prefetch cycles. If the end address condition is specified at an address near the instruction following a branch instruction or delayed slot instruction, correct time cannot be measured. Check the bus trace display for the operation after the branch instruction prefetch cycle, and specify the end address condition at the address in a prefetch cycle which will not be executed by the branching.
- Channel 1 is not available for performance analysis when the Point to Point trace mode is selected.
- The resolution for the performance analysis function can be set in [Timer Resolution] of the [Configuration Properties] dialog box. If the clock counter value is set as the resolution, the value shown in [RUN-TIME] and [MAX-MIN-TIME] will be that of the clock counter (displayed in hexadecimal).
- The counter for measurement has 24 bits, and the maximum measurement time is as given below depending on the value set in [Timer Resolution].

Table 5.8 Maximum Measurement Time

Value of [Timer Resolution]	Maximum Measurement Time
52 us	Approximately 14 minutes
1.6 us	Approximately 26 seconds
20 ns	Approximately 0.33 second

- The maximum measurement count for Access Count of Specified Range Measurement and Called Count of Specified Range Measurement is 65,535.

RENESAS

5.9 Profiling Function

- If there is no stack information file (extension is '.sni') that is output from the optimizing linkage editor, only the functions that have been executed during the profiling data measurement are displayed. For details of the stack information file, refer to the manual of the optimizing linkage editor.
- The stack size differs from the actual value. It should be used as a reference value during a function call. If there is no stack information file (extension is '.sni') that is output from the optimizing linkage editor, the stack size is not displayed.
- While the profiling function is used, software break and on-emulator break, which are event functions, are not available.
- While the profiling function is used, the parallel access function during user program execution is not available.
- Since the profiling function internally breaks user program execution, the program is not executed in realtime. The measured value includes an error.
- [Cycle] displays a decimal value of the counter for measurement of execution time. The resolution for the counter for measurement of execution time can be set in [Timer Resolution] of the [Configuration Properties] dialog box. The expression for the execution time of each function is as follows:

    Execution time = Value of the [Cycle] item × Value of [Time Resolution]
- To enable the profiling function, [Enable read and write on the fly] must be unchecked in the [Configuration Properties] dialog box.

5.10 Input Format

5.10.1 Entering Masks

Address bus conditions and data bus conditions can be input with masks. Addresses can be masked in 1-, 3-, or 4-bit units. When a bit is masked, it always satisfies the condition.

To specify a mask for an address bus condition, specify the mask value in the [Mask] area.

The mask for data conditions is similarly specified in the [Mask] area.

To specify any further mask, specify 1 for the digits to be ignored. Examples of mask specification are listed below.

Table 5.9　Address Mask Specification

Input Value	Mask Unit	Example	Masked Bits
Binary	1 bit	B'00000111	Masks bits 0 to 2
Octal	3 bits	O'000017	Masks bits 0 to 3
Hexadecimal	4 bits	H'07FF	Masks bits 0 to 10

5.11　Tutorial Program

5.11.1　Notes on Operating the Tutorial Program

To operate the High-performance Embedded Workshop according to the description of section 4, Tutorial, the following procedures must be added or modified.

Table 5.11　Notes on Operating the Tutorial Program

4.6　Executing the Program	Reset the target MCU before executing the program. Refer to section 4.14, Resetting the Target MCU for how to reset the target MCU.
4.12.3　Executing the [Step Over] Command	In this product, after execution of [Step Out], execution stops at the statement where the `sort` function is called. Before executing the [Step Over] command, execute the [Step In] command once so that the execution stops at the statement where the `change` function is called.

5.12　Memory Map

5.12.1　Emulation Memory

1. The emulator manages areas in the memory blocks shown in figures 5.1 and 5.2. Emulation memory and user memory cannot coexist in a single block.
2. When emulation memory is used in 50-MHz operation, one or more cycles are required as wait-state cycles for access to those areas to which emulation memory is allocated. When emulation memory is used, refer to the table below and set the number of wait cycles for the bus-state controller.
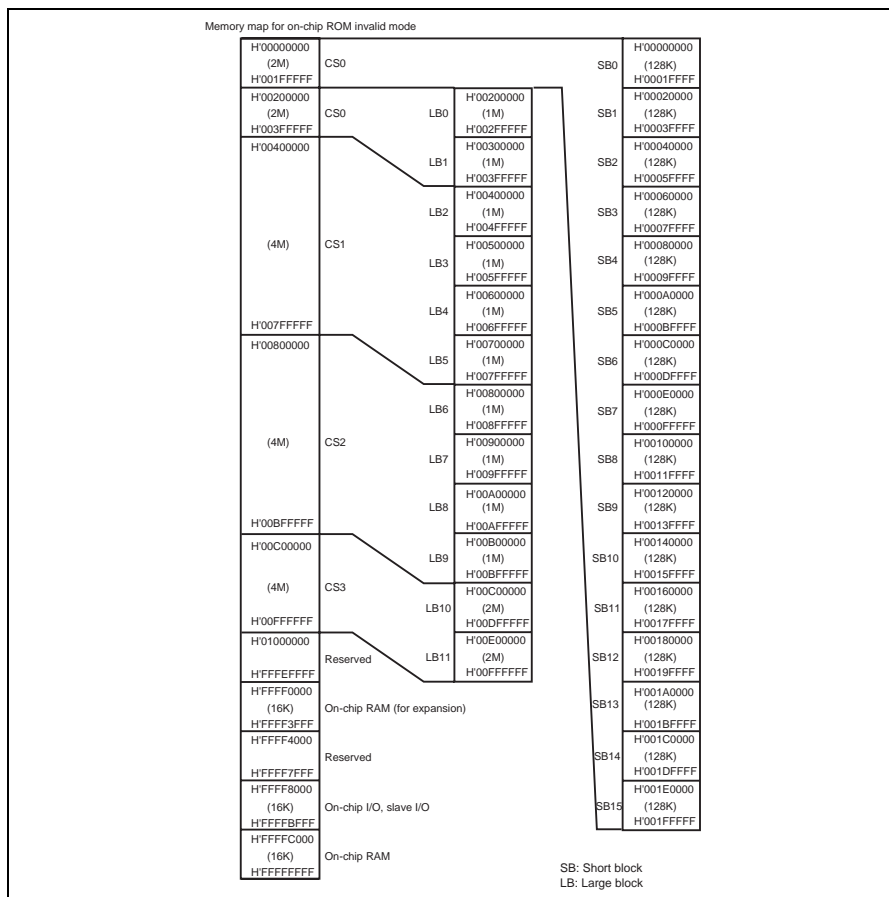
External Operating Frequency	Number of Required Wait Cycles
50 MHz	One or more
40 MHz or lower	0

5.12.2　Controlling Memory Map

The memory map is divided into the 128-kbyte, 1-Mbyte, or 2-Mbyte blocks in each CS area. The accessing attribute can be changed in a unit of each block. For the size of blocks used for each CS area, refer to figure 5.1 or 5.2.

This emulator incorporates the 4-Mbyte emulation memory, which can be divided into 128 kbytes (x 16) and 1 Mbyte (x 2).

The emulation memory can also be set in a unit of each block (figure 5.1 or 5.2), as well as in the accessing attribute. Some 128-kbyte emulation memory is set for the 1-Mbyte or 2-Mbyte blocks. However, the remaining area of the allocated block cannot be used. This also applies to a case when one 1-Mbyte emulation memory is allocated to the 2-Mbyte blocks.

Memory map for on-chip ROM invalid mode

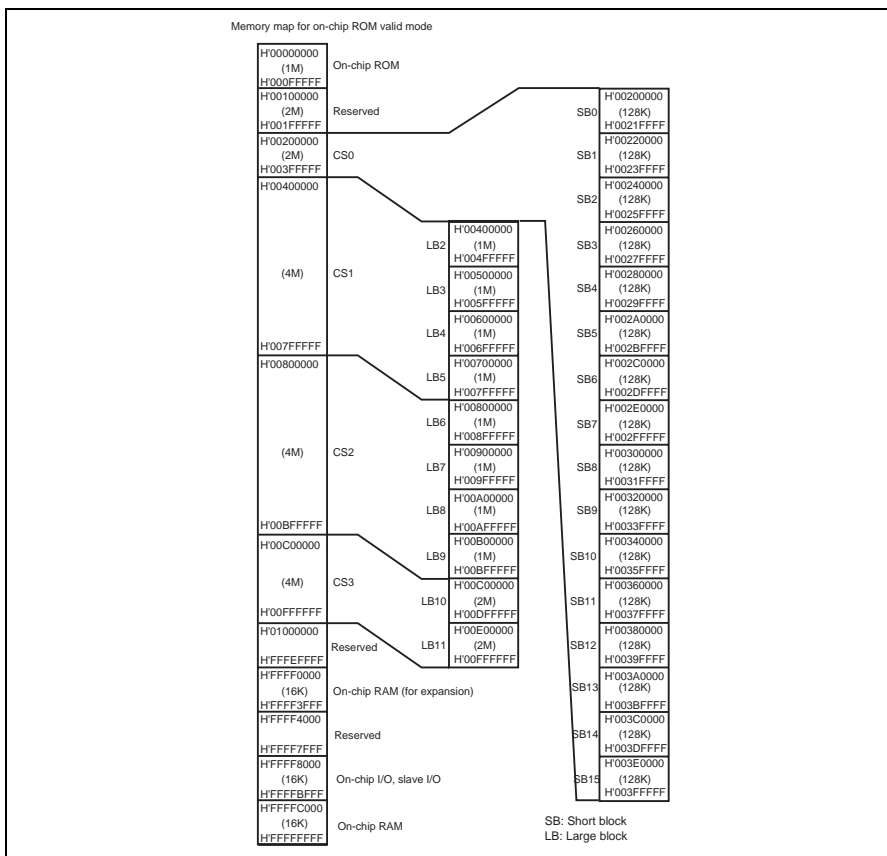|						
H'00000000 (2M) H'001FFFFF	CS0				SB0	H'00000000 (128K) H'0001FFFF
H'00200000 (2M) H'003FFFFF	CS0	LB0	H'00200000 (1M) H'002FFFFF		SB1	H'00020000 (128K) H'0003FFFF
H'00400000		LB1	H'00300000 (1M) H'003FFFFF		SB2	H'00040000 (128K) H'0005FFFF
		LB2	H'00400000 (1M) H'004FFFFF		SB3	H'00060000 (128K) H'0007FFFF
(4M)	CS1	LB3	H'00500000 (1M) H'005FFFFF		SB4	H'00080000 (128K) H'0009FFFF
H'007FFFFF		LB4	H'00600000 (1M) H'006FFFFF		SB5	H'000A0000 (128K) H'000BFFFF
H'00800000		LB5	H'00700000 (1M) H'007FFFFF		SB6	H'000C0000 (128K) H'000DFFFF
		LB6	H'00800000 (1M) H'008FFFFF		SB7	H'000E0000 (128K) H'000FFFFF
(4M)	CS2	LB7	H'00900000 (1M) H'009FFFFF		SB8	H'00100000 (128K) H'0011FFFF
		LB8	H'00A00000 (1M) H'00AFFFFF		SB9	H'00120000 (128K) H'0013FFFF
H'00BFFFFF		LB9	H'00B00000 (1M) H'00BFFFFF		SB10	H'00140000 (128K) H'0015FFFF
H'00C00000		LB10	H'00C00000 (2M) H'00DFFFFF		SB11	H'00160000 (128K) H'0017FFFF
(4M)	CS3	LB11	H'00E00000 (2M) H'00FFFFFF		SB12	H'00180000 (128K) H'0019FFFF
H'00FFFFFF					SB13	H'001A0000 (128K) H'001BFFFF
H'01000000 H'FFFEFFFF	Reserved				SB14	H'001C0000 (128K) H'001DFFFF
H'FFFFF000 (16K) H'FFFFF3FFF	On-chip RAM (for expansion)				SB15	H'001E0000 (128K) H'001FFFFF
H'FFFFF4000 H'FFFFF7FFF	Reserved					
H'FFFFF8000 (16K) H'FFFFFBFFF	On-chip I/O, slave I/O					
H'FFFFFC000 (16K) H'FFFFFFFFF	On-chip RAM					

SB: Short block
LB: Large block

Figure 5.1   Memory Map in the Invalid On-Chip ROM Mode

RENESAS

Memory map for on-chip ROM valid mode

H'00000000 (1M) H'000FFFFF	On-chip ROM				
H'00100000 (2M) H'001FFFFF	Reserved		SB0	H'00200000 (128K) H'0021FFFF	
H'00200000 (2M) H'003FFFFF	CS0		SB1	H'00220000 (128K) H'0023FFFF	
H'00400000			SB2	H'00240000 (128K) H'0025FFFF	
		LB2	H'00400000 (1M) H'004FFFFF	SB3	H'00260000 (128K) H'0027FFFF
(4M)	CS1	LB3	H'00500000 (1M) H'005FFFFF	SB4	H'00280000 (128K) H'0029FFFF
		LB4	H'00600000 (1M) H'006FFFFF	SB5	H'002A0000 (128K) H'002BFFFF
H'007FFFFF		LB5	H'00700000 (1M) H'007FFFFF	SB6	H'002C0000 (128K) H'002DFFFF
H'00800000		LB6	H'00800000 (1M) H'008FFFFF	SB7	H'002E0000 (128K) H'002FFFFF
(4M)	CS2	LB7	H'00900000 (1M) H'009FFFFF	SB8	H'00300000 (128K) H'0031FFFF
		LB8	H'00A00000 (1M) H'00AFFFFF	SB9	H'00320000 (128K) H'0033FFFF
H'00BFFFFF H'00C00000		LB9	H'00B00000 (1M) H'00BFFFFF	SB10	H'00340000 (128K) H'0035FFFF
(4M)	CS3	LB10	H'00C00000 (2M) H'00DFFFFF	SB11	H'00360000 (128K) H'0037FFFF
H'00FFFFFF H'01000000		LB11	H'00E00000 (2M) H'00FFFFFF	SB12	H'00380000 (128K) H'0039FFFF
H'FFFEFFFF	Reserved		SB13	H'003A0000 (128K) H'003BFFFF	
H'FFFF0000 (16K) H'FFFF3FFF	On-chip RAM (for expansion)		SB14	H'003C0000 (128K) H'003DFFFF	
H'FFFF4000 (16K) H'FFFF7FFF	Reserved		SB15	H'003E0000 (128K) H'003FFFFF	
H'FFFF8000 (16K) H'FFFFBFFF	On-chip I/O, slave I/O				
H'FFFFC000 (16K) H'FFFFFFFF	On-chip RAM				

SB: Short block
LB: Large block

Figure 5.2   Memory Map in the Valid On-Chip ROM Mode

RENESAS

Section 6   Error Messages

6.1   Error Messages of the Emulator

6.1.1   Error Messages at Emulator Initiation

The emulator displays error messages in the format below if an error occurs at emulator initiation in the dedicated message dialog box when the High-performance Embedded Workshop is used.

Table 6.1 lists error messages at emulator initiation.

Table 6.1   Error Messages at Initiation

Error Message	Description and Solution
There is no configuration file.	The configuration file that is required to initiate the emulator cannot be found. Exit and re-install the High-performance Embedded Workshop. Then re-connect the user system interface cable, turn on the power of the emulator, and re-initiate the High-performance Embedded Workshop. If the problem is not solved, contact us and describe the error occurrence in detail.
The contents of the configuration file are incorrect.	The configuration file that is required to initiate the emulator is invalid. Exit and re-install the High-performance Embedded Workshop. Then re-connect the user system interface cable, turn on the power of the emulator, and re-initiate the High-performance Embedded Workshop. If the problem is not solved, contact us and describe the error occurrence in detail.
Main Board not Support (XX XX XX) Emulator is switched off or not connected	The emulator power is not turned on, or the user system interface cable is not connected. Exit the High-performance Embedded Workshop, re-connect the user system interface cable, turn on the power of the emulator, and re-initiate the High-performance Embedded Workshop. If the problem is not solved, contact us and describe the error occurrence in detail.
Emulation Board not Support (XX XX XX) Emulator is switched off or not connected	The emulator power is not turned on, or the user system interface cable is not connected. Exit the High-performance Embedded Workshop, re-connect the user system interface cable, turn on the power of the emulator, and re-initiate the High-performance Embedded Workshop. If the problem is not solved, contact us and describe the error occurrence in detail.
EVA chip Board not Support (XX XX XX) Emulator is switched off or not connected	The emulator power is not turned on, or the user system interface cable is not connected. Exit the High-performance Embedded Workshop, re-connect the user system interface cable, turn on the power of the emulator, and re-initiate the High-performance Embedded Workshop. If the problem is not solved, contact us and describe the error occurrence in detail.
Can't initialize G/A registers	An error occurred during the initialization of the emulator. Exit the High-performance Embedded Workshop, re-connect the user system interface cable, turn on the power of the emulator, and re-initiate the High-performance Embedded Workshop. If the problem is not solved, contact us and describe the error occurrence in detail.
There is no effective clock source	A valid clock source cannot be found. Connect a valid clock source.
This mode can not specify	The state of mode pins for the target board is incorrect. Initiation is only possible in mode 4. Set the mode pins correctly.

RENESAS

Table 6.1   Error Messages at Initiation (cont)

Error Message	Description and Solution
Can't find firmware file Firmware open Error Firmware Download Error Firmware Name Error	There is an error in the file that is required at emulator initiation. Exit the High-performance Embedded Workshop, re-connect the user system interface cable, turn on the power of the emulator, and re-initiate the High-performance Embedded Workshop. If the problem is not solved, contact us and describe the error occurrence in detail.
Failed to receive a firmware initialization command.	Initiation of the emulator firmware has failed. Exit the High-performance Embedded Workshop, re-connect the user system interface cable, turn on the power of the emulator, and re-initiate the High-performance Embedded Workshop. If the problem is not solved, contact us and describe the error occurrence in detail.
Target system is Vcc down	The value of Vcc is lower than the specified threshold value.
JTAG Timeout Srval Error JTAG Packet Receive Error	Exit the High-performance Embedded Workshop, re-connect the user system interface cable, turn on the power of the emulator, and re-initiate the High-performance Embedded Workshop.

6.1.2   Error Messages during Emulation

The emulator displays error messages if an error occurs during emulation in the dedicated message dialog box when the High-performance Embedded Workshop is used, and on the status bar. Table 6.2 lists error messages during emulation.

Table 6.2   Error Messages during Emulation

| Error Message	Description and Solution
Communication DLL error. Communication Timeout error.	The power of the emulator is turned off or there is a communication error. Exit the High-performance Embedded Workshop, re-connect the user system interface cable, turn on the power of the emulator, and re-initiate the High-performance Embedded Workshop. If the problem is not solved, contact us and describe the error occurrence in detail.
Parallel Access Error	An error has occurred during a parallel access. Parallel accesses are disabled until a break occurs.

RENESAS

Appendix A   Menus

Table A.1 shows GUI menus.

Table A.1   GUI Menus

Menu	Option		Shortcut	Toolbar Button	Remarks
View	Command Line		Ctrl + L		Opens the [Command Line] window.
	Workspace		Alt + K		Opens the [Workspace] window.
	Output		Alt + U		Opens the [Output] window.
	Disassembly		Ctrl + D		Opens the [Disassembly] window.
	CPU	Registers	Ctrl + R		Opens the [Register] window.
		Memory…	Ctrl + M		Opens the [Memory] window.
		IO	Ctrl + I		Opens the [IO] window.
		Status	Ctrl + U		Opens the [Status] window.
		Extended Monitor			Opens the [Extended Monitor] window.
		Monitor	Monitor Setting…	Shift + Ctrl + E		Opens the [Monitor] window.
			Windows Select…			Opens the [Windows Select] dialog box to list, add, or edit the [Monitor] window.
	Symbol	Labels	Shift + Ctrl + A		Opens the [Labels] window.
		Watch	Ctrl + W		Opens the [Watch] window.
		Locals	Shift + Ctrl + W		Opens the [Locals] window.
	Code	Eventpoints	Ctrl + E		Opens the [Event] window.
		Trace	Ctrl + T		Opens the [Trace] window.
		Code Coverage...	Shift + Ctrl + H		Opens the [Code Coverage] window.
		Data coverage...	Shift + Ctrl + Z		Opens the [Data Coverage] window.
		Stack Trace	Ctrl + K		Opens the [Stack Trace] window.

Table A.1 GUI Menus (cont)

Menu	Option		Shortcut	Toolbar Button	Remarks
View (cont)	Gra-phic	Image…	Shift + Ctrl + G		Opens the [Image] window.
		Waveform…	Shift + Ctrl + V		Opens the [Waveform] window.
	Per-form-ance	Performance Analysis	Shift + Ctrl + P		Opens the [Performance Analysis] window.
		Profile	Shift + Ctrl + F		Opens the [Profile] window.

Menu	Option	Shortcut	Toolbar Button	Remarks
Debug	Debug Sessions…			Opens the [Debug Sessions] dialog box to list, add, or remove the debug session.
	Debug Settings…			Opens the [Debug Settings] dialog box to set the debugging conditions or download modules.
	Reset CPU			Resets the target hardware and sets the PC to the reset vector address.
	Go	F5		Starts executing the user program at the current PC.
	Reset Go	Shift + F5		Resets the target hardware and executes the user program from the reset vector address.
	Go To Cursor			Starts executing the user program at the current PC until the PC reaches the address indicated by the current text cursor position.
	Set PC To Cursor			Sets the PC to the address at the row of the text cursor.
	Run…			Launches the [Run Program] dialog box allowing the user to enter the PC or PC breakpoint during executing the user program.
	Display PC	Shift+Ctrl +Y		Opens the [Editor] or [Disassembly] window at the address of the PC.

RENESAS

Table A.1   GUI Menus (cont)

Menu	Option		Shortcut	Toolbar Button	Remarks
Debug (cont)	Step In		F11		Executes a block of user program before breaking.
	Step Over		F10		Executes a block of user program before breaking. If a subroutine call is reached, then the subroutine will not be entered.
	Step Out		Shift + F11		Executes the user program to reach the end of the current function.
	Step…				Launches the [Step Program] dialog box allowing the user to modify the settings for stepping.
	Step Mode	Auto			Steps only one source line when the [Editor] window is active. When the [Disassembly] window is active, stepping is executed in a unit of assembly instructions.
		Assembly			Executes stepping in a unit of assembly instructions.
		Source			Steps only one source line.
	Halt Program		Esc		Stops the execution of the user program.
	Initialize				Disconnects the emulator and connects it again.
	Connect				Connects the emulator.
	Disconnect				Disconnects the emulator.
	Save Memory…				Saves the specified memory area data to a file.
	Verify Memory…				Verifies file contents against memory contents.
	Configure Overlay…				Selects the target section group when the overlay function is used.
	Download Modules				Downloads the object program.
	Unload Modules				Unloads the object program.

RENESAS

Table A.1   GUI Menus (cont)

Menu	Option		Shortcut	Toolbar Button	Remarks
Setup	Customize…				Customize the High-performance Embedded Workshop application.
	Options…				Sets option of the High-performance Embedded Workshop application.
	Format Views…				Configure fonts, colors, keywords and so on, for the window.
	Radix	Hexadecimal		16	Uses a hexadecimal for displaying a radix in which the numerical values will be displayed and entered by default.
		Decimal		10	Uses a decimal for displaying a radix in which the numerical values will be displayed and entered by default.
		Octal		8	Uses an octal for displaying a radix in which the numerical values will be displayed and entered by default.
		Binary		2	Uses a binary for displaying a radix in which the numerical values will be displayed and entered by default.
	Emu-lator	System…		📶	Opens the [Configuration Properties] dialog box allowing the user to modify the emulator settings.
		Memory Resource…		🖼	Opens the [Memory Mapping] dialog box allowing the user to view and edit the emulator 's current memory map.

RENESAS

Appendix B   Command Lines

Table B.1 lists the High-performance Embedded Workshop commands.

Table B.1   High-performance Embedded Workshop Commands

No.	Command Name	Abbreviation	Function
1	!	-	Comment
2	ADD_FILE	AF	Adds a file to the current project
3	ANALYSIS	AN	Enables or disables performance analysis
4	ANALYSIS_RANGE	AR	Sets or displays a performance analysis range
5	ANALYSIS_RANGE_ DELETE	AD	Deletes a performance analysis range
6	ASSEMBLE	AS	Assembles instructions into memory
7	ASSERT	-	Checks if an expression is true or false
8	AUTO_COMPLETE	AC	Enables or disables the auto-complete function
9	BREAKPOINT_ONCHIP	BC	Displays on-chip breakpoints, sets sequential breaks, and sets PtoP time measurement
10	BREAKPOINT_ONCHIPn	BCn	Sets on-chip breakpoint of each channel
11	BREAKPOINT_ONCHIP _CLEAR	BCC	Clears on-chip breakpoints
12	BREAKPOINT_ONCHIP _ENABLE	BCE	Enables or disables an on-chip breakpoint
13	BREAKPOINT_ ONEMULATOR	BE	Displays on-emulator breakpoints
14	BREAKPOINT_ ONEMULATORn	BEn	Sets on-emulator breakpoint of each channel
15	BREAKPOINT_ ONEMULATOR_CLEAR	BEC	Clears on-emulator breakpoints
16	BREAKPOINT_ ONEMULATOR_ ENABLE	BEE	Enables or disables an on-emulator breakpoint
17	BREAKPOINT_ SOFTWARE	BS	Sets a software breakpoint
18	BREAKPOINT_ SOFTWARE_CLEAR	BSC	Clears software breakpoints
19	BREAKPOINT_ SOFTWARE_ENABLE	BSE	Enables or disables a software breakpoint
20	BUILD	BU	Performs a build on the current project
21	BUILD_ALL	BL	Performs a build all on the current project
22	CHANGE_ CONFIGURATION	CC	Sets the current configuration
23	CHANGE_PROJECT	CP	Sets the current project
24	CHANGE_SESSION	CS	Changes the current session
25	CLOSE_WORKSPACE	CW	Close the current workspace
26	CLOCK_MODE	CKM	Sets and displays the clock mode
27	CONFIGURE_ PLATFORM	CPF	Sets the debugging environment for the emulator
28	COVERAGE_DISPLAY	CVD	Displays coverage information
29	COVERAGE_CLEAR	CVC	Initializes coverage information
30	COVERAGE_SET	CVS	Sets or modifies coverage information

RENESAS

Table B.1   High-performance Embedded Workshop Commands (cont)

No.	Command Name	Abbreviation	Function
31	DEFAULT_OBJECT_ FORMAT	DO	Sets the default object (program) format
32	DEVICE_TYPE	DE	Selects a device type to emulate
33	DISASSEMBLE	DA	Disassembles memory contents
34	EMULATOR_CLOCK	ECK	Selects the clock rate of the target MCU for the emulator
35	ERASE	ER	Clears the [Command Line] window
36	EVALUATE	EV	Evaluates an expression
37	EXMONITOR_DISPLAY	EXMD	Displays the content of the expansion monitor
38	EXMONITOR_SET	EXMS	Selects whether or not to display the items in the expansion monitor
39	EXMONITOR_ SETRATE	EXMSR	Sets the time to update the expansion monitor during emulation or a break
40	FILE_LOAD	FL	Loads an object (program) file
41	FILE_SAVE	FS	Saves memory to a file
42	FILE_UNLOAD	FU	Unloads a file
43	FILE_VERIFY	FV	Verifies file contents against memory
44	GENERATE_MAKE_ FILE	GM	Creates a makefile to be built outside the High-performance Embedded Workshop
45	GO	GO	Executes user program
46	GO_RESET	GR	Executes user program from reset vector
47	GO_TILL	GT	Executes user program until temporary breakpoint
48	HALT	HA	Halts the user program
49	HELP	HE	Displays the command line help
50	INITIALIZE	IN	Initializes the debugging platform
51	JTAG_CLOCK	JCK	Sets and displays the JTAG clock (TCK)
52	LOG	LO	Controls command output logging
53	MAP_DISPLAY	MA	Displays memory mapping
54	MAP_SET	MS	Sets memory mapping
55	MEMORY_COMPARE	MC	Compares memory contents
56	MEMORY_DISPLAY	MD	Displays memory contents
57	MEMORY_EDIT	ME	Modifies memory contents
58	MEMORY_FILL	MF	Modifies the content of a memory area by specifying data
59	MEMORY_FIND	MI	Searches for data within the memory range
60	MEMORY_MOVE	MV	Moves a block of memory
61	MEMORY_TEST	MT	Tests a block of memory
62	MODE	MO	Sets or displays the MCU mode
63	MONITOR_CLEAR	MOC	Deletes a monitor point
64	MONITOR_DISPLAY	MOD	Displays the content of the monitor
65	MONITOR_REFRESH	MOR	Controls an automatic update of the content of the monitor
66	MONITOR_SET	MOS	Sets or displays a monitor point
67	OPEN_WORKSPACE	OW	Opens a workspace

RENESAS

Table B.1   High-performance Embedded Workshop Commands (cont)

No.	Command Name	Abbreviation	Function
68	PINSEL	PSL	Displays the current settings for selection of pins
69	PINSEL_xxxx	PSLxxxx	Allows settings for selection of pins (xxxx: Pin name)
70	PROFILE	PR	Enables or disables the profile
71	PROFILE_DISPLAY	PD	Displays profiling results
72	PROFILE_SAVE	PS	Saves profiling results
73	QUIT	QU	Exits High-performance Embedded Workshop
74	RADIX	RA	Sets default input radix
75	REFRESH	RF	Updates windows related to memory
76	REGISTER_DISPLAY	RD	Displays CPU register values
77	REGISTER_SET	RS	Sets CPU register contents
78	REMOVE_FILE	REM	Deletes the specified file from the current project
79	RESET	RE	Resets CPU
80	SAVE_SESSION	SE	Saves the current session
81	SLEEP	-	Delays command execution
82	STEP	ST	Steps program (by instructions or source lines)
83	STEP_MODE	SM	Sets the step mode
84	STEP_OUT	SP	Steps out of the current function
85	STEP_OVER	SO	Steps program, not stepping into functions
86	STEP_RATE	SR	Sets or displays rate of stepping
87	SUBMIT	SU	Executes a command file
88	SYMBOL_ADD	SA	Defines a symbol
89	SYMBOL_CLEAR	SC	Deletes a symbol
90	SYMBOL_LOAD	SL	Loads a symbol information file
91	SYMBOL_SAVE	SS	Saves a symbol information file
92	SYMBOL_VIEW	SV	Displays symbols
93	STATUS	STS	The content of the [Platform] sheet in the [Status] window is displayed.
94	SAVE_WORKSPACE	SW	Saves the current workspace
95	TCL	-	Enables or disables the TCL
96	TIMER	TI	Sets or displays the timer resolution
97	TOOL_INFORMATION	TO	Outputs information on the currently registered tool to a file
98	TRACE	TR	Displays trace information
99	TRACE_ACQUISITION	TA	Sets or displays trace acquisition parameters
100	TRACE_ACQUISITIONn	TAn	Sets PtoP point and each channel for trace acquisition conditions
101	TRACE_ACQUISITION _CLEAR	TAC	Deletes trace acquisition parameters
102	TRACE_BINARY_ COMPARE	TBC	Compares a trace binary file with the current trace information
103	TRACE_BINARY_SAVE	TBV	Outputs trace information into a binary file

RENESAS

Table B.1   High-performance Embedded Workshop Commands (cont)

No.	Command Name	Abbreviation	Function
104	TRACE_FILTER	TF	Filters trace information
105	TRACE_SAVE	TV	Outputs trace information into a file
106	TRACE_STATISTIC	TST	Analyzes statistic information
107	UPDATE_ALL_ DEPENDENCIES	UD	Updates dependencies of the current project
108	USER_SIGNALS	US	Enables or disables the user signal information
109	WATCH_ADD	WA	Adds a watch item
110	WATCH_AUTO_UPDATE	WU	Selects or cancels automatic update of watch items
111	WATCH_DELETE	WD	Deletes a watch item
112	WATCH_DISPLAY	WI	Displays the contents of the Watch window
113	WATCH_EDIT	WE	Edits the value of a watch item
114	WATCH_EXPAND	WX	Expands or collapses a watch item
115	WATCH_RADIX	WR	Changes the radix of a watch item to be displayed
116	WATCH_SAVE	WS	Saves the contents of the Watch window to a file

For the syntax of each command, refer to the online help.

RENESAS

SH7046 E6000H Emulator
User's Manual

RENESAS

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