To our customers,

# Old Company Name in Catalogs and Other Documents

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

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

April 1st, 2010
Renesas Electronics Corporation

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

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

RENESAS

## Notice

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

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

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

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

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

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

To all our customers

# Regarding the change of names mentioned in the document, such as Hitachi Electric and Hitachi XX, to Renesas Technology Corp.

The semiconductor operations of Mitsubishi Electric and Hitachi were transferred to Renesas Technology Corporation on April 1st 2003. These operations include microcomputer, logic, analog and discrete devices, and memory chips other than DRAMs (flash memory, SRAMs etc.) Accordingly, although Hitachi, Hitachi, Ltd., Hitachi Semiconductors, and other Hitachi brand names are mentioned in the document, these names have in fact all been changed to Renesas Technology Corp. Thank you for your understanding. Except for our corporate trademark, logo and corporate statement, no changes whatsoever have been made to the contents of the document, and these changes do not constitute any alteration to the contents of the document itself.

Renesas Technology Home Page: http://www.renesas.com

Renesas Technology Corp.
Customer Support Dept.
April 1, 2003

## Cautions

Keep safety first in your circuit designs!

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

Notes regarding these materials

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

User's Manual

# SH7058 E6000H Emulator HS7058EPH60H

## User's Manual

Renesas Electronics
www.renesas.com

Rev.2.0 2002.10

# Cautions

1. Hitachi neither warrants nor grants licenses of any rights of Hitachi's or any third party's patent, copyright, trademark, or other intellectual property rights for information contained in this document. Hitachi bears no responsibility for problems that may arise with third party's rights, including intellectual property rights, in connection with use of the information contained in this document.
2. Products and product specifications may be subject to change without notice. Confirm that you have received the latest product standards or specifications before final design, purchase or use.
3. Hitachi makes every attempt to ensure that its products are of high quality and reliability. However, contact Hitachi's sales office before using the product in an application that demands especially high quality and reliability or where its failure or malfunction may directly threaten human life or cause risk of bodily injury, such as aerospace, aeronautics, nuclear power, combustion control, transportation, traffic, safety equipment or medical equipment for life support.
4. Design your application so that the product is used within the ranges guaranteed by Hitachi particularly for maximum rating, operating supply voltage range, heat radiation characteristics, installation conditions and other characteristics. Hitachi bears no responsibility for failure or damage when used beyond the guaranteed ranges. Even within the guaranteed ranges, consider normally foreseeable failure rates or failure modes in semiconductor devices and employ systemic measures such as fail-safes, so that the equipment incorporating Hitachi product does not cause bodily injury, fire or other consequential damage due to operation of the Hitachi product.
5. This product is not designed to be radiation resistant.
6. No one is permitted to reproduce or duplicate, in any form, the whole or part of this document without written approval from Hitachi.
7. Contact Hitachi's sales office for any questions regarding this document or Hitachi semiconductor products.

# 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 Hitachi, Ltd. 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 Hitachi microcomputer HD64F7058 (hereafter referred to as MCU). 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 Hitachi sales company. Such use includes, but is not limited to, use in life support systems. Buyers of this emulator product must notify the relevant Hitachi sales offices before planning to use the product in such applications.

### Improvement Policy:

Hitachi, Ltd. (including its subsidiaries, hereafter collectively referred to as Hitachi) pursues a policy of continuing improvement in design, performance, and safety of the emulator product. Hitachi may 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.

# LIMITED WARRANTY

Hitachi warrants its emulator products to be manufactured in accordance with published specifications and free from defects in material and/or workmanship. Hitachi, at its option, will repair or replace any emulator products returned intact to the factory, transportation charges prepaid, which Hitachi, upon inspection, determine to be defective in material and/or workmanship. The foregoing shall constitute the sole remedy for any breach of Hitachi's warranty. See the Hitachi 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. Hitachi is not liable for any claim made by a third party or made by you for a third party.

# DISCLAIMER

HITACHI 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 HITACHI 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.

### 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:

Hitachi 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 Hitachi's prior written consent or any problems caused by the user system.

### All Rights Reserved:



### Other Important Things to Keep in Mind:

1. Circuitry and other examples described herein are meant merely to indicate the characteristics and performance of Hitachi's semiconductor products. Hitachi 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 Hitachi.

### Figures:

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

### Limited Anticipation of Danger:

Hitachi 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.

# SAFETY PAGE

## 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.**

## 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.

## ⚠ 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.**

# Warnings on Emulator Usage

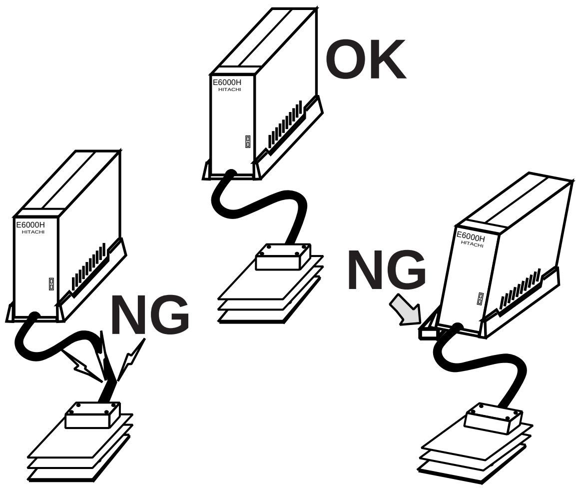
Warnings described below apply as long as you use this emulator. Be sure to read and understand the warnings below before using this emulator. Note that these are the main warnings, not the complete list.

**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.**

# 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.



# CAUTION

**This equipment has been tested and found to comply with the limits for a Class A digital device, pursuant to part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference in which case the user will be required to correct the interference at his own expense.**

# Preface

Thank you for purchasing the emulator for the Hitachi microcomputer SH7058.

| |
|---|
| **CAUTION** |
| **Read section 3, Preparation before Use before using the emulator product. Incorrect operation or connection will damage the user system, the emulator product, and the user program.** |

The SH7058 E6000H emulator (hereinafter referred to as the emulator) is an efficient software and hardware development tool for systems based on Hitachi microcomputer SH7058. The emulator is operated by using the Hitachi Debugging Interface (hereafter referred to as HDI). This interface program is supported by Windows® 98, Windows® Me, Windows NT® 4.0, and Windows® 2000.

This manual describes the emulator functions and operations. Please read this manual carefully before use, in particular section 1.1, Notes on Usage. A CD-R for the E6000H emulator is packaged with the emulator. For details, refer to section 3, Preparation before Use.

**Related Manuals:**

Description Notes on Using the PC Card Interface (HS6000EIP01H) for the E6000/E8000 Emulator
Description Notes on Using the PCI Interface Board (HS6000EIC01H) for the E6000/E8000 Emulator
Description Notes on Using the PCI Interface Board (HS6000EIC02H) for the E6000/E8000 Emulator
Description Notes on Using the LAN Adapter (HS6000ELN01H) for the E6000/E8000 Emulator
Description Notes on Using the USB Adapter (HS6000EIU01H) for the E6000/E8000 Emulator
Hitachi Embedded Workshop User’s Manual
SuperH™ RISC engine C/C++ Compiler User's Manual
SuperH™ RISC engine Assembler User's Manual
H Series Linkage Editor, Librarian, Object Converter User's Manual
Hitachi Debugging Interface User's Manual
Hardware Manual supporting each MCU
Programming Manual supporting each MCU

Notes: 1. IBM PC is a registered trademark of International Business Machines Corporation.

2. Microsoft®, Windows®, and Windows NT® 4.0 are registered trademarks of Microsoft Corporation in the United States and/or in other countries.

3. Pentium® is a registered trademark of Intel Corporation in the United States.

Abbreviation: 1. Windows® 98 is an abbreviation for Microsoft® Windows® 98 operating system.

2. Windows® Me is an abbreviation for Microsoft® Windows® Millennium Edition.

3. Windows NT® 4.0 is an abbreviation for Microsoft® Windows NT® 4.0 operating system.

4. Windows® 2000 is an abbreviation for Microsoft® Windows® 2000 operating system.

# Contents

## Figures

## Tables

# Section 1 Overview

This system is an efficient software and hardware development support tool for application systems using the HD64F7058 (hereafter referred to as SH7058) microcomputer developed by Hitachi, Ltd.

As peripheral functions for use with its high-speed CPU, the SH7058 incorporates a floating-point unit (FPU), direct memory access controller (DMAC), advanced timer unit (ATU-II), advanced pulse unit (APU), watchdog timer (WDT), compare-match timer (CMT), serial communications interface (SCI), Hitachi controller area network (HCAN), A/D converter, interrupt controller (INTC), I/O ports, memory, etc.

The emulator operates in place of the SH7058 and performs realtime emulation of the user system. The emulator also provides functions for efficient hardware and software debugging.

The emulator package consists of the SH7058 E6000H station and a user system interface board. The emulator is connected to the user system via the dedicated connector on the evaluation chip board or the user system interface board.

PC interface (option) includes a PC interface board (PCI bus and PC card bus; installed on the host computer), 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 SH7058 E6000H Hitachi Debugging Interface (hereafter referred to as HDI) 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.

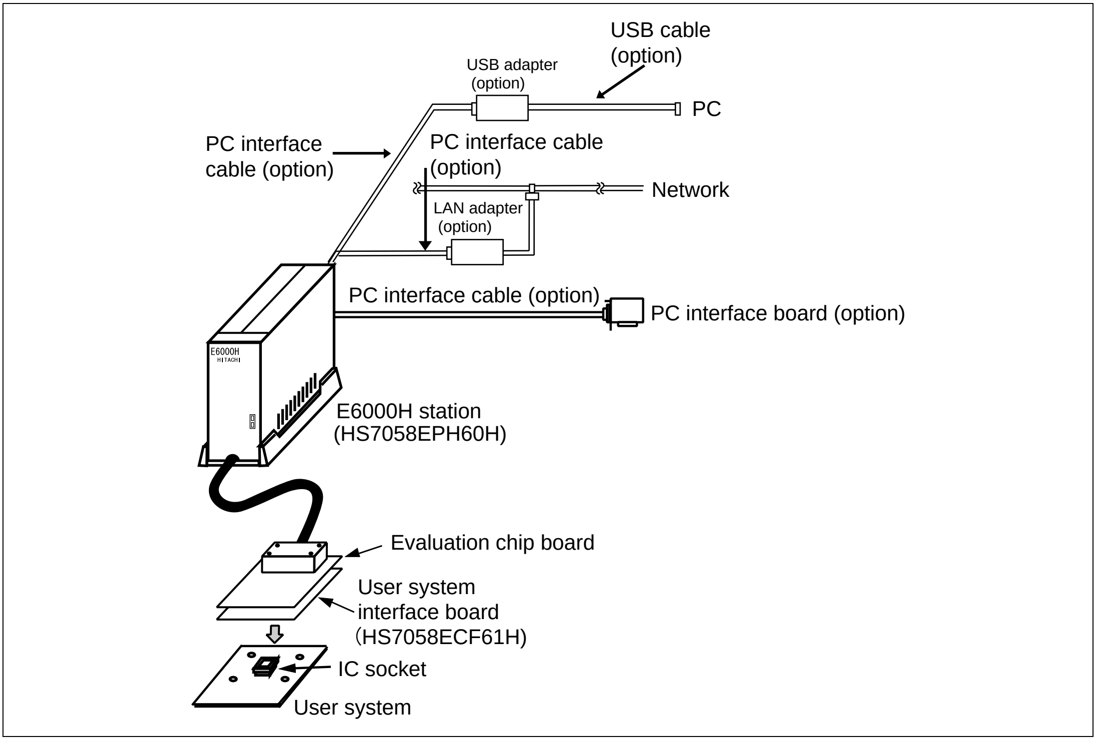



**Figure 1.1 Configuration of the Emulator for the SH7058**

The emulator provides the following features:

1. Realtime emulation of the MCU at 80 MHz
2. A wide selection of emulation commands, promoting efficient system development
3. On-line help functions to facilitate command usage without a manual
4. Efficient debugging enabled by variable break functions and a mass-storage trace memory (128-kcycles)
5. Parallel access with a command execution during emulation, for example
   - Trace data display
   - Emulation memory display and modification
6. Performance analysis
   Measurement of subroutine execution time and count for evaluating the execution efficiency of user programs
7. A LAN adapter for connecting the emulator to a host computer via a LAN interface (10BASE-T or 100BASE-TX), allowing loading of the SH7058 E6000H Hitachi Debugging Interface into the host computer. This enables graphic display operations in a multi-window environment, and source-level debugging.
8. A USB adapter for connecting the emulator to a host computer via a USB interface, allowing loading of the SH7058 E6000H Hitachi Debugging Interface into the host computer. This enables graphic display operations in a multi-window environment, and source-level debugging.
9. A PC interface board (for the PCI bus or PC card bus) connected to the host computer through the PC interface cable. The HDI can be loaded into the host computer to enable:
   - Graphic display operations in a multi-window environment
   - Source-level debugging

Note: Ethernet® is a registered trademark of Xerox Corporation (USA).

## 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 1.2, 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 1.2, 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 3, 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 MCU and the emulator, refer to appendix E.5, Differences between the Emulator and the MCU.

## 1.2 Environmental Conditions

> **CAUTION**
>
> **Observe the conditions listed in table 1.1 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.**

**Table 1.1 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. | |

Details of the operating environment are listed in table 1.2.

**Table 1.2 Operating Environment**

| Item | Operating Environment |
|---|---|
| Host computer | IBM PCs and compatible machines that contain Pentium® III processors (800 MHz or faster is recommended) |
| Operating system | Windows® 98, Windows® Me, Windows NT® 4.0, or Windows® 2000 |
| Minimum memory capacity for operation | 32 Mbytes (more than twice the size of the load module is recommended) |
| Display | Resolution better than 800 × 600 (SVGA) is recommended |
| Empty space in a hard disk | Disk capacity required for installation: 40 Mbytes or more<br>Take the swap area into account when ensuring that there is enough space on your system (more than four times the size of the memory is recommended). |
| Supported interfaces | PCI bus slot, PC card (PCMCIA), LAN adapter (conforming to IEEE802.3, with 10BASE-T or 100BASE-TX), USB adapter (conforming to V1.1) |
| Pointing device such as a mouse | A pointing device such as a mouse, which can be connected to the host computer and is supported by Windows® 98, Windows® Me, Windows NT® 4.0, and Windows® 2000 |
| CD-ROM drive | Required for installing the emulator software or referring to the user's manual |

## 1.3 Components

The emulator components are listed below. Check all components after unpacking. If any component is missing, contact the sales agency from which the emulator was purchased.

### 1.3.1 Emulator

Table 1.3 lists the emulator components.

**Table 1.3 Emulator Components (HS7058EPH60H)**

| Classification | Item | Quantity | Remarks |
|---|---|---|---|
| Hardware | E6000H station | 1 | |
| | Trace cable | 1 | Connected to the emulator station |
| | SH7058 evaluation chip board | 1 | Two boards connected via a trace cable |
| | External probe | 1 | Probe input: four,<br>RUN or break output: one<br>Trigger output: one<br>GND: two |
| | AC power cable | 1 | |
| Software | SH7058 E6000H emulator | 1 | CD-R<br>HS7058EPH60SR |
| Manual | Description Notes on Using the SH7058 E6000H Emulator | 1 | HS7058EPH60HE-P(*) |

Note: ‘*’ indicates a manual revision.

### 1.3.2 Options

In addition to the E6000H station, the options listed in table 1.4 are also available. Refer to each option manual for details on these optional components.

**Table 1.4 Optional Component Specifications**

| Item | Model Name | Specifications |
|---|---|---|
| User system interface board for the SH7058 (FP-256H) | HS7058ECF61H | For the SH7058 (FP-256H) |
| PCI interface board | HS6000EIC01H<br>HS6000EIC02H | PCI bus |
| PC interface card | HS6000EIP01H | PC card bus |
| LAN adapter | HS6000ELN01H | • TCP/IP communications protocol<br>• 10BASE-T<br>• 100BASE-TX |
| USB adapter | HS6000EIU01H | Conforms to V1.1 |

# Section 2 Components

## 2.1 Emulator Hardware Components

The emulator consists of an E6000H station and an SH7058 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; installed on the host computer), 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 SH7058 E6000H Hitachi Debugging Interface (hereafter referred to as HDI) 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.

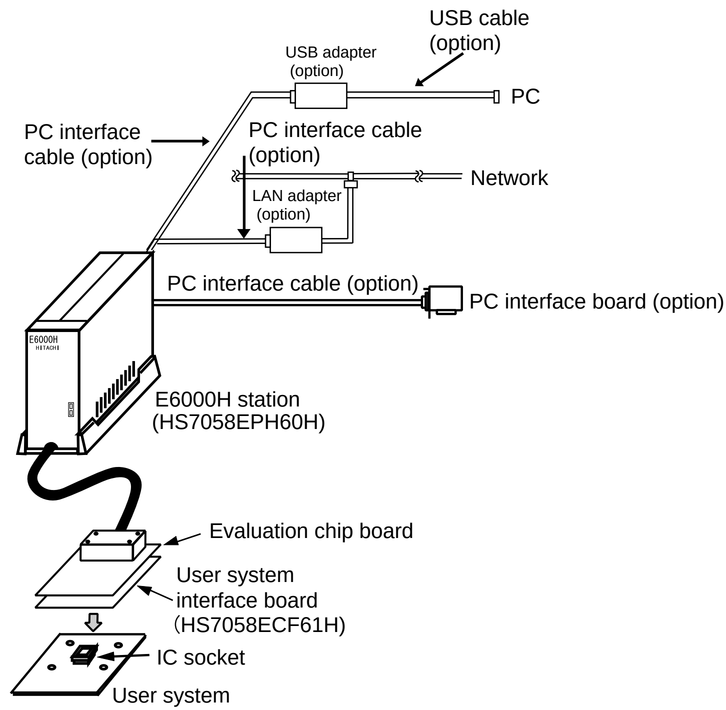



**Figure 2.1 Emulator Hardware Components**

### 2.1.1 E6000H Station Components

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

**Front Panel:**

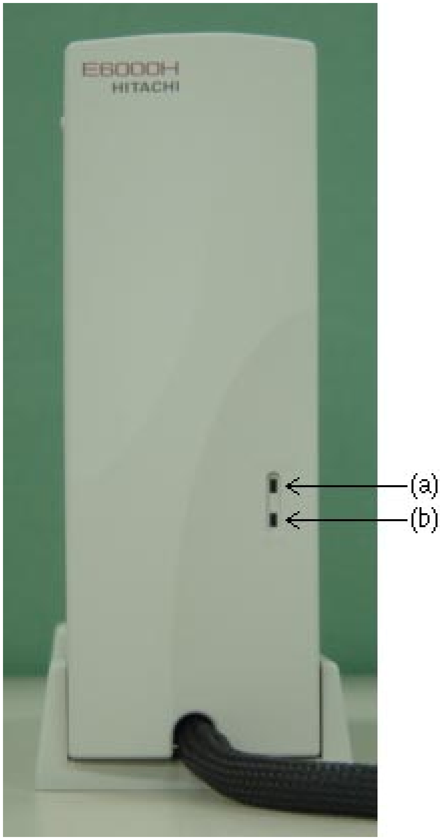



**Figure 2.2 E6000H Station: Front Panel**

1. POWER lamp: Is lit up while the E6000H station is supplied with power.
2. RUN lamp: Is lit up while the user program is running.

**Rear Panel:**

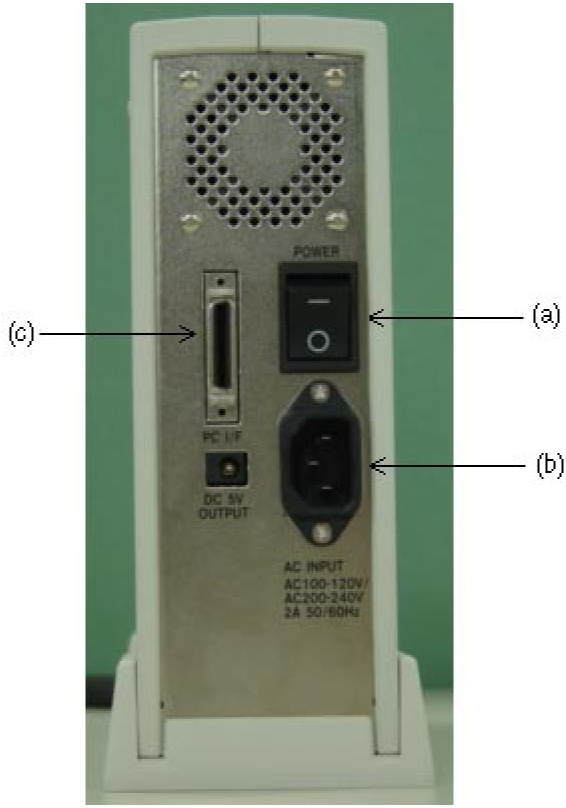



**Figure 2.3 E6000H Station: Rear Panel**

| | | |
|---|---|---|
| (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. |

### 2.1.2 Evaluation Chip Board Configuration

The names of the components on the evaluation chip board of the emulator are listed below.

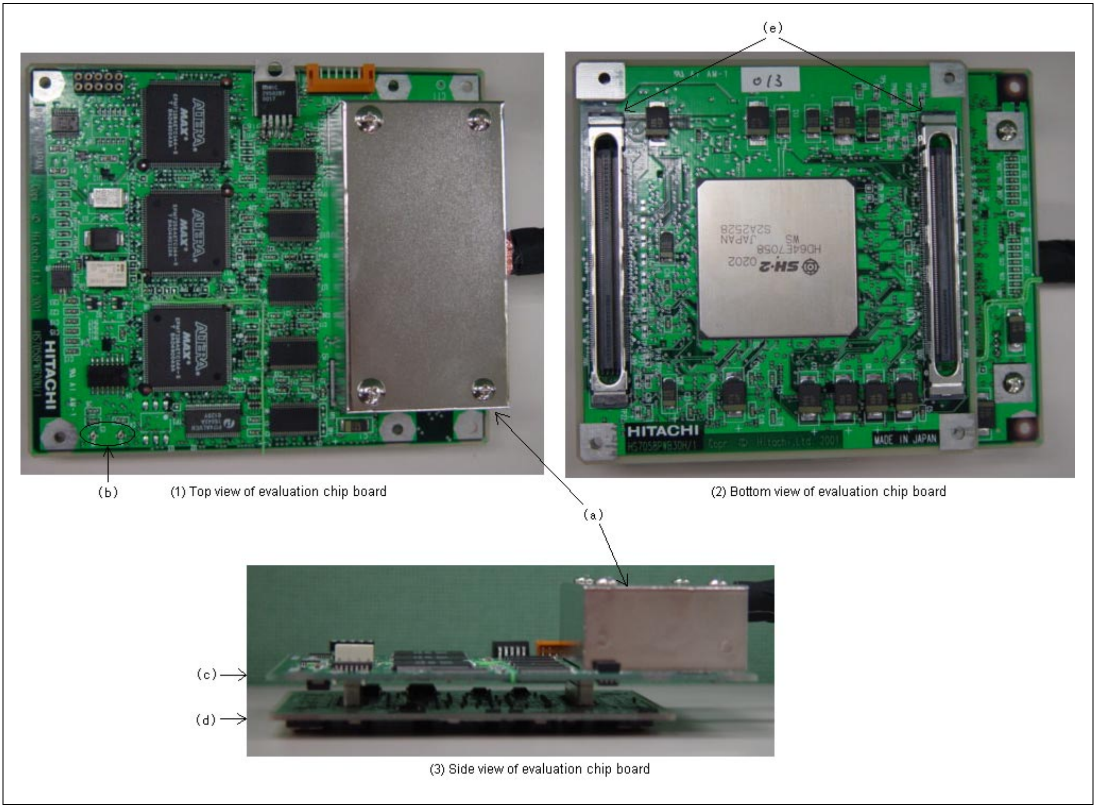



**Figure 2.4 Evaluation Chip Board (HS7058EPH60H)**

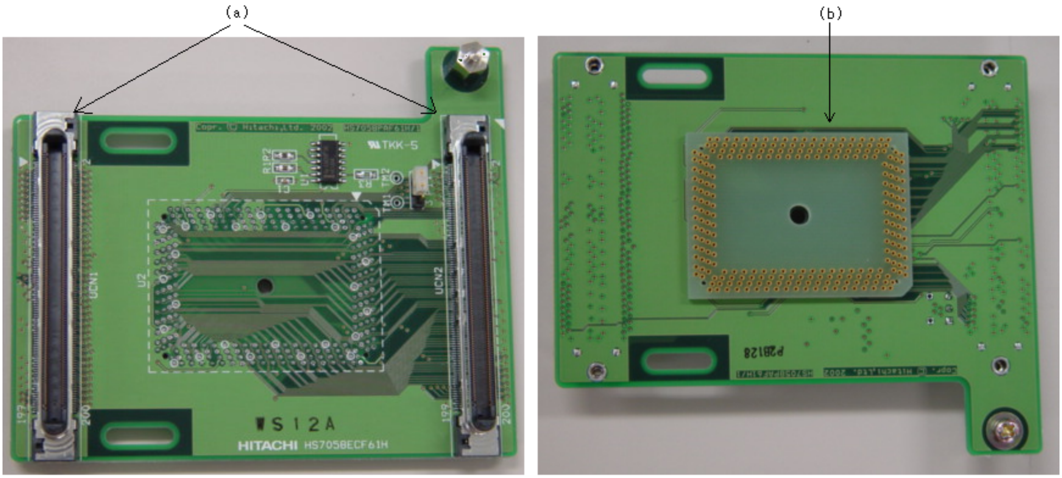
| | |
|---|---|
| (a) Station to evaluation chip board interface connector cover: | This is a cover for protecting the connector that connects the E6000H station to the evaluation chip board. |
| (b) Crystal oscillator terminals: | For installing a crystal oscillator to be used as an external clock source for the MCU. |
| (c) HS7058PWB20H board: | Connector to the trace cable is attached. |
| (d) HS7058PWB30H 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 evaluation chip board.

### 2.1.3 Configuration of User System Interface Board

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




(1) Top view of HS7058ECF61H board

(2) Bottom view of HS7058ECF61H board

**Figure 2.5 Configuration of User System Interface Board**

| | |
|---|---|
| (a) Connector for the evaluation chip board | For connection to the evaluation chip board. |
| (b) Connector for the user system | For connection to the user system. |

## 2.2 Configuration of the Provided CD-R

The provided CD-R contains software for the SH7058 E6000H emulator and user’s manuals. Table 2.1 shows the configuration of the CD-R. All directories listed in this table are under the directory 7058 in the root directory.

**Table 2.1 Contents of CD-R**

| Directory Name | File Name | Contents | Note |
|---|---|---|---|
| | setup.exe | Installer | |
| | Hdi.exe [*1] | HDI program for the SH7058 E6000H emulator | |
| \Diag | Tm7058.exe [*1] | Diagnostic and maintenance program | |
| \Drivers\Pci\95 | pcihei.inf | Setup information (PCI) | For Windows® 98 and Windows® Me |
| \Drivers\Pci\95 | pcihei.vxd | Virtual driver (PCI) | For Windows® 98 and Windows® Me |
| \Drivers\Pci\Nt | pcihei.sys | System file (PCI) | For Windows NT® |
| \Drivers\Pci\2000 | pcihei2k.sys | System file (PCI) | For Windows® 2000 |
| \Drivers\Pci\2000 | pcihei2k.inf | Setup information (PCI) | For Windows® 2000 |
| \Drivers\Pcmcia\95 | ulepcc.inf | Setup information (PCMCIA) | For Windows® 98 |
| \Drivers\Pcmcia\95 | ulepcc.vxd | Virtual driver (PCMCIA) | For Windows® 98 |
| \Drivers\Pcmcia\Nt | ulepccnt.sys | System file (PCMCIA) | For Windows NT® |
| \Drivers\Pcmcia\2000 | ulepcc2k.sys | System file (PCMCIA) | For Windows® 2000 |
| \Drivers\Pcmcia\2000 | ulepcc2k.inf | Setup information (PCMCIA) | For Windows® 2000 |
| \Manuals\Japanese | HS6400DIIW5SJ.pdf [*2] | Hitachi Debugging Interface user's manual | PDF documents in Japanese [*4] |
| \Manuals\Japanese | HS7058EPH60HJ.pdf [*2] | SH7058 E6000H emulator user's manual | PDF documents in Japanese [*4] |

**Table 2.1 Contents of CD-R (cont)**

| Directory Name | File Name | Contents | Notes |
|---|---|---|---|
| \Manuals\English | HS6400DIIW5SE.pdf [*3] | Hitachi Debugging Interface user's manual | PDF documents in English [*4] |
| \Manuals\English | HS7058EPH60HE.pdf [*3] | SH7058 E6000H emulator user's manual | PDF documents in English [*4] |
| \Pdf_read\Japanese | Ar50jpn.exe | Acrobat® Reader 5.0 installer | Japanese version |
| \Pdf_read\English | Ar50eng.exe | Acrobat® Reader 5.0 installer | English version |
| \Tutorial | | Sample program | |

Notes: 1. Installed on your host computer by the installer.

2. This is the Japanese version of the manual. It cannot be installed by the English-language version of the installer.

3. This is the English version of the manual. It cannot be installed by the Japanese-language version of the installer.

4. Use the Acrobat® Reader to see PDF documents.

## 2.3 System Configuration

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

### 2.3.1 System Configuration Using a PC Interface Board

The emulator can be connected to a host computer via a PC interface board (options: 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 2.6 shows the configuration of a system in which the PC interface board is used. Figure 2.7 shows the configuration of a system in which the LAN adapter is used. Figure 2.8 shows the configuration of a system in which the USB adapter is used.

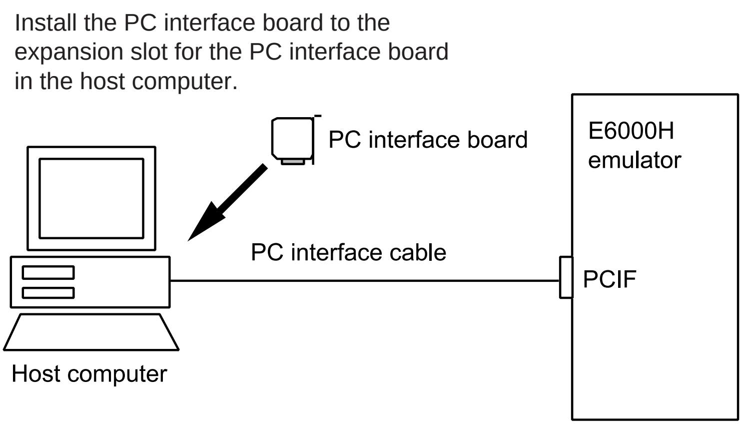



**Figure 2.6 System Configuration Using a PC Interface Board**

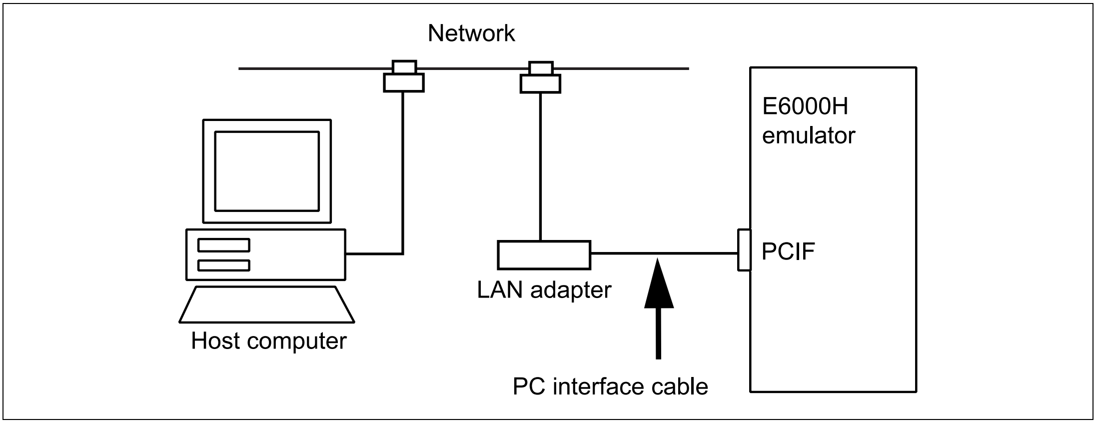



**Figure 2.7 System Configuration Using a LAN Adapter**

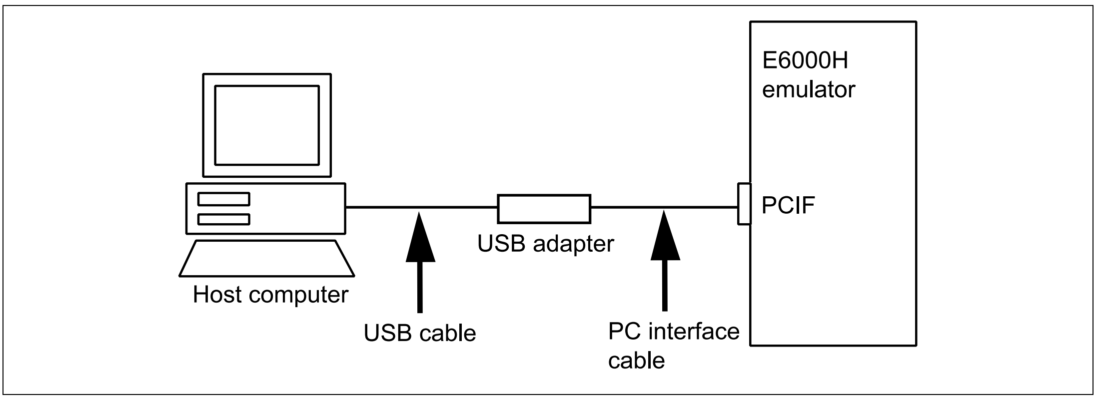



**Figure 2.8 System Configuration Using a USB Adapter**

# Section 3 Preparation before Use

## 3.1 Description on Emulator Usage

This section describes the preparation before emulator usage. Figure 3.1 is a flowchart on preparation before the usage of 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.** |

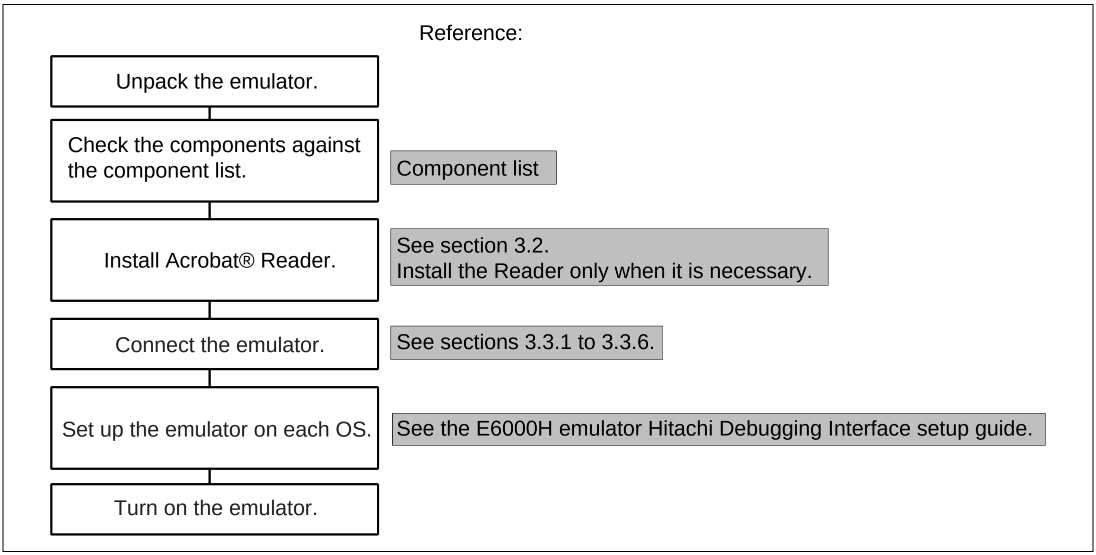



**Figure 3.1 Emulator Preparation Flowchart**

## 3.2 Installing the Acrobat® Reader

Acrobat® Reader 5.0 is required to view the online help. Acrobat® Reader is provided on the CD-R of this product. The installation of Acrobat® Reader is described below. If you have already installed Acrobat® Reader, do not carry out this procedure.

1. Insert the CD-R of this product in the CD-ROM drive.
2. Click [Run…] from the [Start] menu.
3. Specify Ar50eng.exe in Pdf_Read\English directory in the [Run] dialog box (e.g. D:\7058\Pdf_Read\English\Ar50eng.exe) then click the [OK] button.
4. Install according to the instructions displayed in the screen.

## 3.3 Emulator Connection

The following description is given on emulator connection.

### 3.3.1 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.

### 3.3.2 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 SH7058 E6000H-series products.

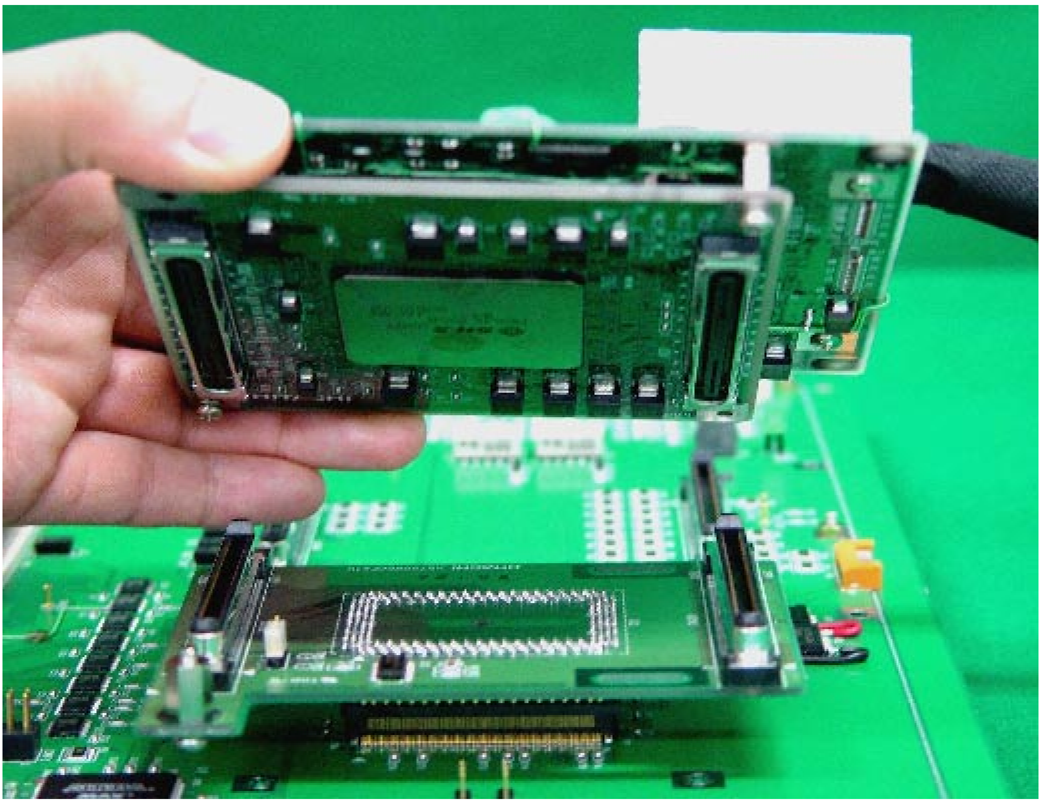


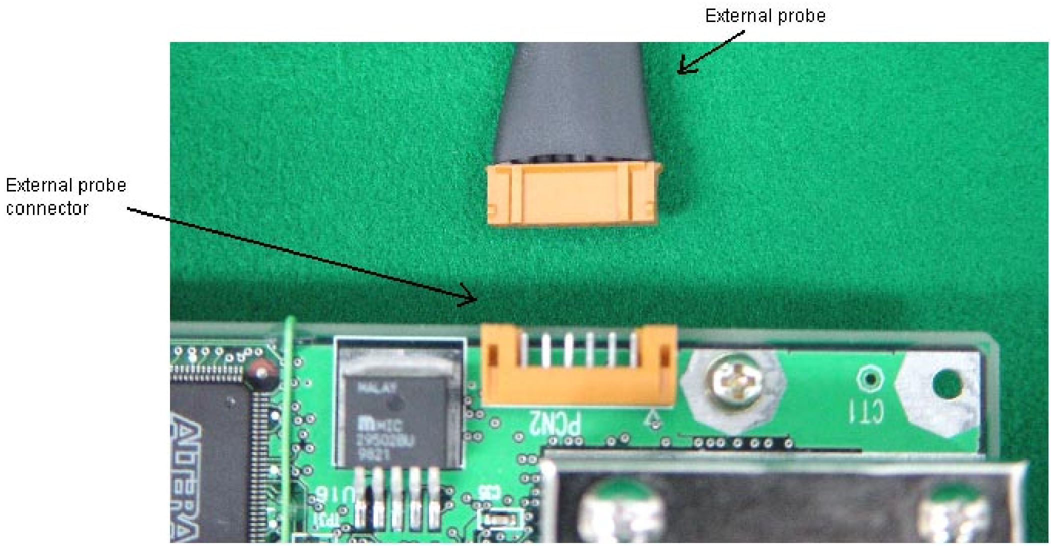
**Figure 3.2 Connecting the User System Interface Board**

### 3.3.3 Connecting the External Probe

# 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 3.3 shows the external probe connector.




| Pin No. | Probe Name | Signal Name | Note |
|---|---|---|---|
| 1 | 1 | Probe input 0 | Synchronous break input pin |
| 2 | 2 | Probe input 1 | |
| 3 | 3 | Probe input 2 | |
| 4 | 4 | Probe input 3 | |
| 5 | 5 | GND | GND connection pin |
| 6 | 6 | Trigger output | Trigger mode output pin |
| 7 | 7 | GND | GND connection pin |
| 8 | 8 | RUN or break output | Low-level output at RUNSTEP |

**Figure 3.3 External Probe Connector**

### 3.3.4 Selecting the Clock

This emulator supports three types of clock for the 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 or the CLOCK command.

This emulator can use a clock source (ϕ) running at up to 80.0 MHz (eight times the external clock frequency of 10.0 MHz) as the MCU clock input.

CLOCK command
- XTAL (Crystal oscillator: 5.0 to 10.0 MHz)
- USER (External clock: 5.0 to 10.0 MHz)
- 10 (Emulator internal clock: 10.0 MHz)
- 5 (Emulator internal clock: 5.0 MHz)

**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 MCU clock source, the frequency range must be from 5.0 to 10.0 MHz.

**CAUTION**

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

Use the following procedure 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 3.4).
3. Turn on the user system power and then the emulator power. Then crystal oscillator will be automatically set and setup. 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.

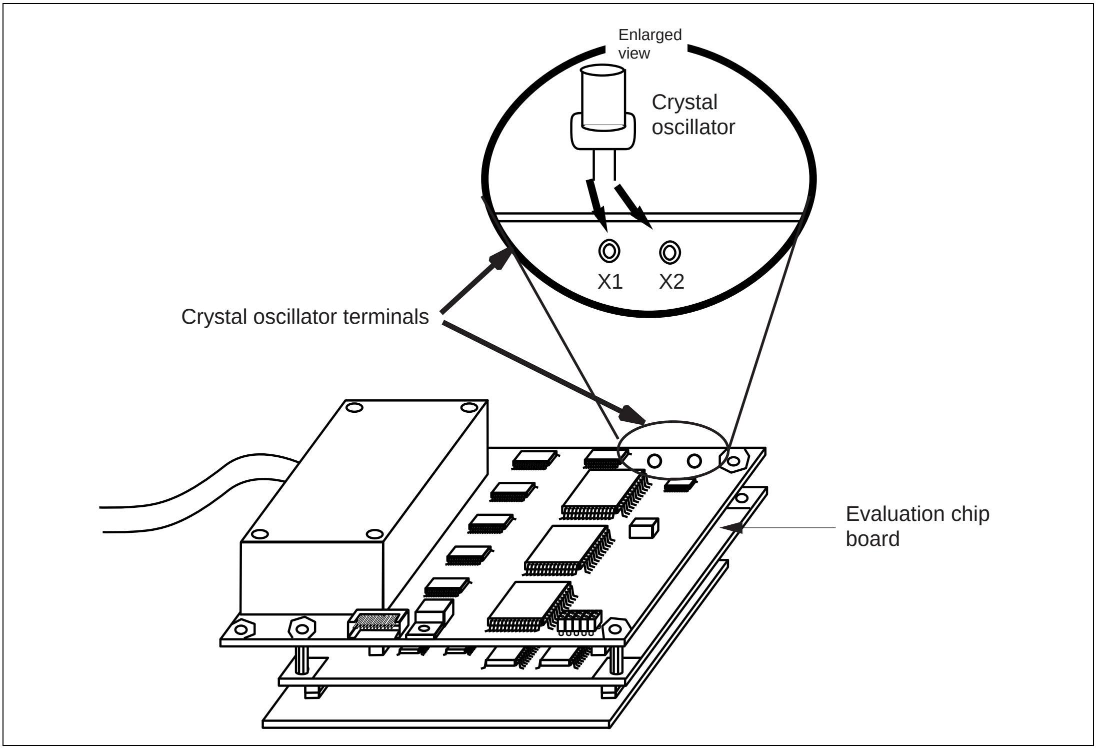



**Figure 3.4 Installing the Crystal Oscillator**

**External Clock:** Use the following procedure 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. USER (the external clock) will then be automatically specified by a CLOCK command.

**Emulator Internal Clock:** Specify 5 (5.0 MHz) or 10 (10.0 MHz) in the [Configuration] dialog box or with the CLOCK command.

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

### 3.3.5 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 3.5). 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 3.6) so that the ground potentials become even.

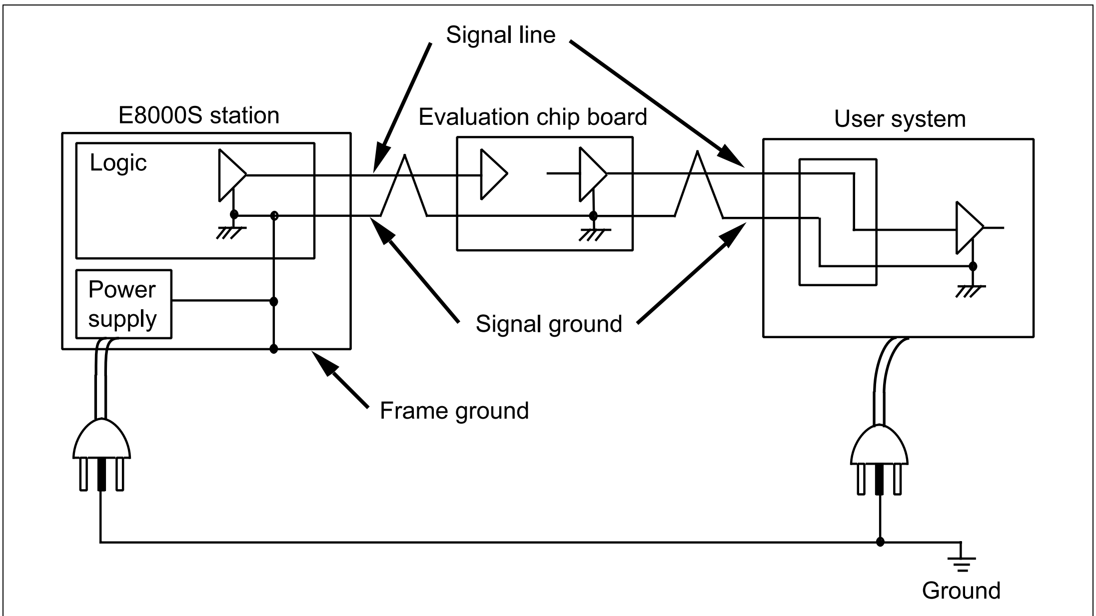



**Figure 3.5 Connecting the System Ground**

# ⚠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.

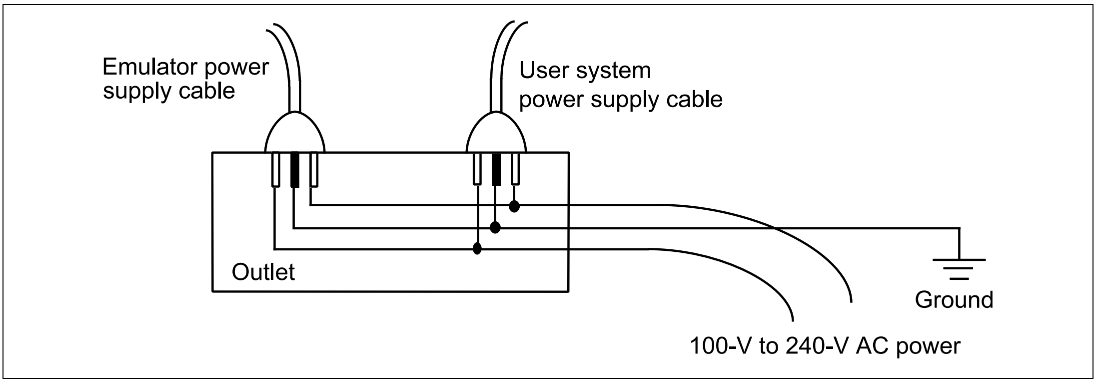



**Figure 3.6 Connecting the Frame Ground**

### 3.3.6 PC Interface Board Specifications (PCI Bus Specifications)

**PC Interface Board Specifications:** Table 3.1 lists the PCI-bus PC interface board specifications. For details on other interface boards, refer to their description notes.

**Table 3.1 PC Interface Board Specifications**

| Item | Specifications |
|---|---|
| Host computer that can be used | PC with a PCI slot, or compatible machine |
| OS | Microsoft® Windows NT® 4.0, Windows® 98, Windows® Me, or Windows® 2000 operating system |
| Memory area | 16 kbytes |
| PCI-bus specifications | Conforms to revision 2.1 |

**Installing the PC Interface Board:**

**⚠WARNING**

**Always switch OFF the host computer and peripheral devices connected to the host computer before installing the PC interface board. Failure to do so will result in a FIRE HAZARD and will damage the host computer, interface board, and peripheral devices, or will result in PERSONAL INJURY.**

Remove the cover of the host computer and install the PC interface board in the PCI-bus specification extension slot. Tighten the screw after confirming that the PC interface cable can be connected to the board.

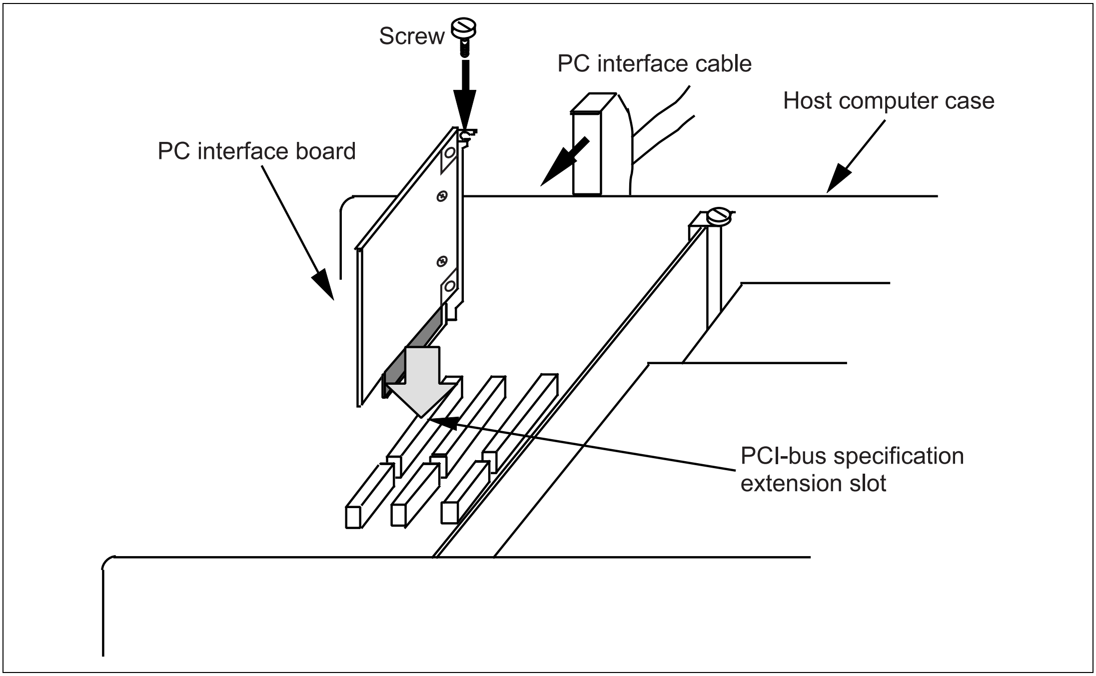



**Figure 3.7 Installing the PCI Interface Board**

**Connecting the E6000H Station to the PC Interface Board:**

**⚠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.**

Before using the emulator, connect the E6000H station to the PC interface board with the PC interface cable supplied, as shown in figure 3.8.

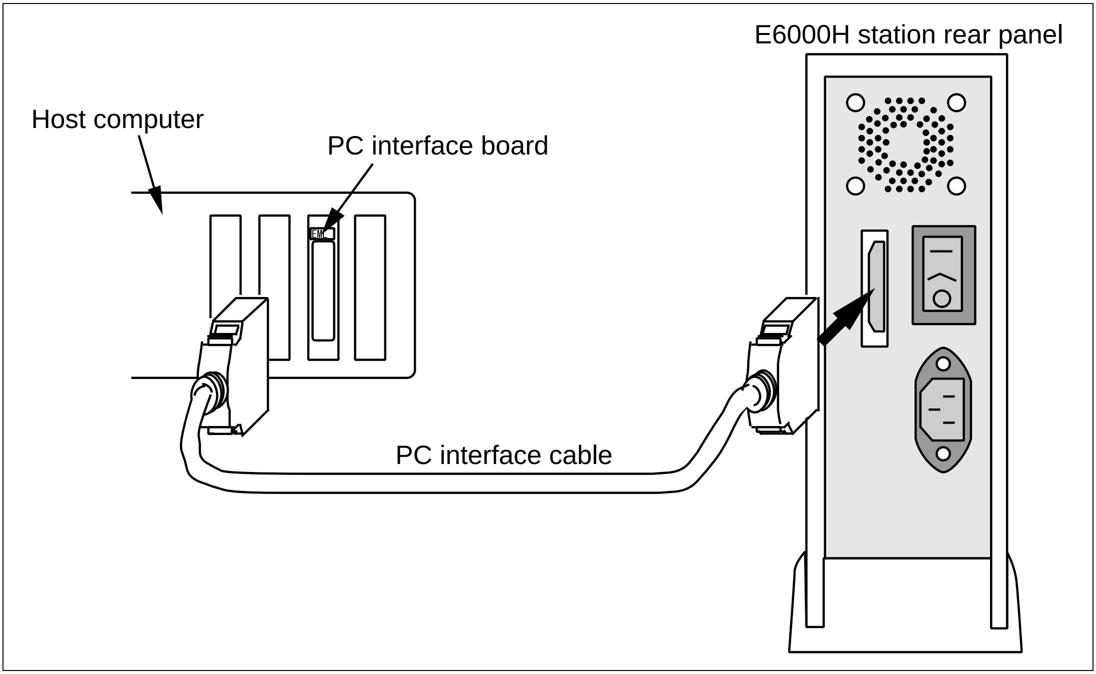



**Figure 3.8 Connecting the E6000H Station to the PC Interface Board**

# Section 4 Tutorial

## 4.1 Introduction

The following describes the main functions of the HDI by using a sample program for sorting random data. For more complicated usage, refer to section 5, Emulator Functions. This tutorial assumes that the user system is not connected.

The sample program performs the following actions:

- The `main` function generates 10 pieces of random data to be sorted.
- The `sort` function generates the array and inputs the random data in the array, and sorts the random data in ascending order.
- The `change` function inputs the array generated by the `sort` function, and changes the data in descending order.

Note: This sample program is created by using the SuperH RISC engine C/C++ compiler Package (V6.0A).

## 4.2 Running the HDI

- To run the HDI, select the [SH7058 E6000H Emulator Software]-[Hitachi Debugging Interface] from the [Start] menu.

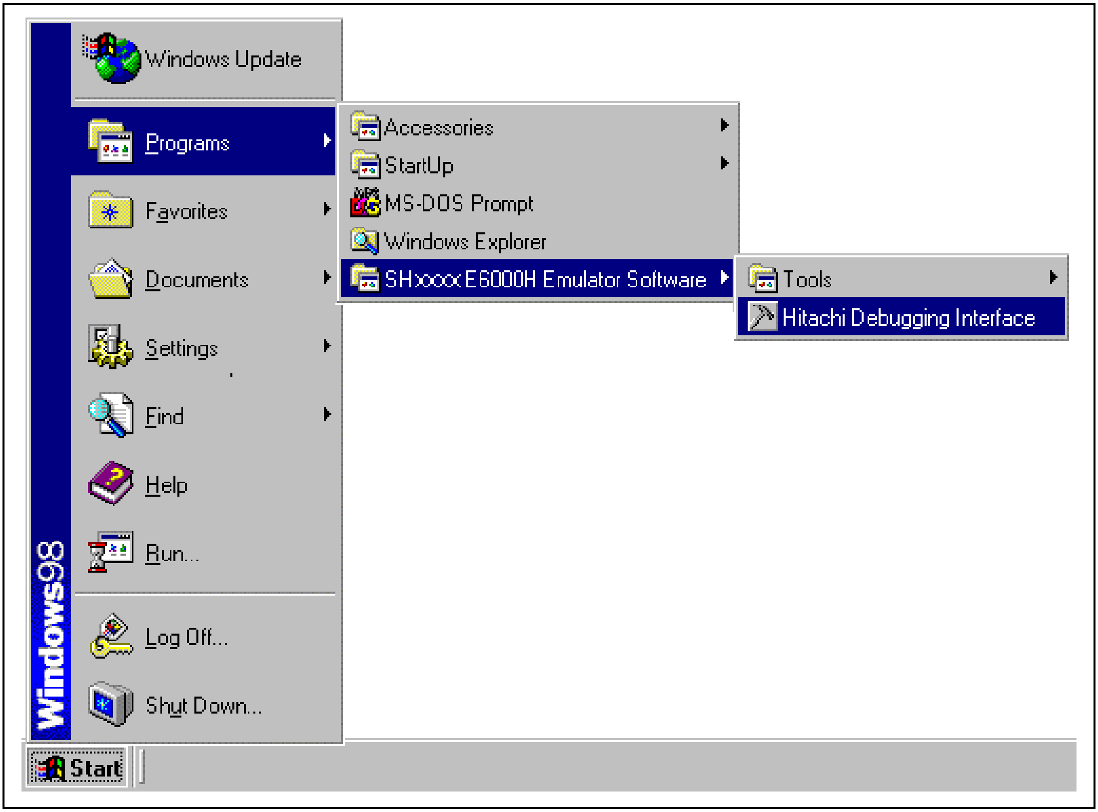



**Figure 4.1 [Start] Menu**

The HDI window as shown in figure 4.2 is displayed. Here the [Select Session] dialog box is displayed. Select the target MCU name of the installed HDI and click the [OK] button.

The settings related to the MCU, such as the MCU operating mode, can be performed on the [General] page in the [Configuration] dialog box. Table 4.1 lists the settings when running the sample program.

**Table 4.1 [General] Page in the [Configuration] Dialog Box Setting Example: Configuration of the [General] Page**

| Option | Setting Value |
|---|---|
| [Device] | SH7058 |
| [Mode] | 3 (On-chip ROM or Single Chip) |
| [Clock] | 5 MHz |
| [H-UDI (JTAG) Clock] | 10 MHz |
| [Timer Resolution] | 20 ns |
| [Input Frequency] | Input prohibited |
| [Flash Load Option] | Disabled |
| [User Signals] | All enabled |
| [Bus timeout] | 100 us |
| [Enable interrupts during step execution] | Disabled |
| [Enable select AUD to Emulator] | Enabled |
| [Enable multi break] | Disabled |
| [User VCC Threshold] | 4.4 V |

The settings can be checked on the [General] page in the [Configuration] dialog box.

For details on the [General] page in the [Configuration] dialog box, refer to section 5.2, Setting the Emulator's Operating Conditions.

The HDI window is shown in figure 4.2.

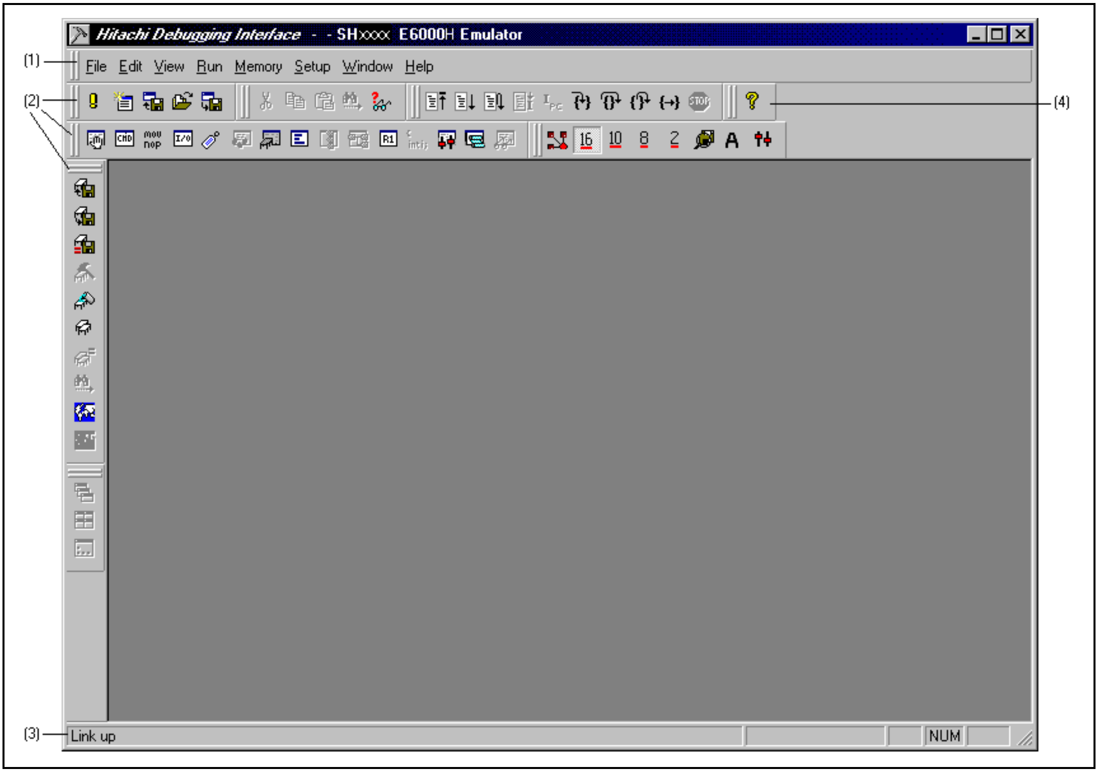



**Figure 4.2 HDI Window**

Numbers in figure 4.2 indicate the following:

1. Menu bar

   Indicates the HDI command menus for the use of the HDI.
2. Toolbar

   Contains convenient buttons as shortcuts of menu commands.
3. Status bar

   Indicates the state of the emulator and progress information about downloading.
4. [Help] button

   Activates the HDI on-line help.

## 4.3 Setting the Memory Map

In the next step, allocate the emulation memory.

- Select [Configure platform…] from the [Setup] menu to display the current memory map.

The [Configuration] dialog box is displayed. Click the [ERam] tab to display the [ERam] page.

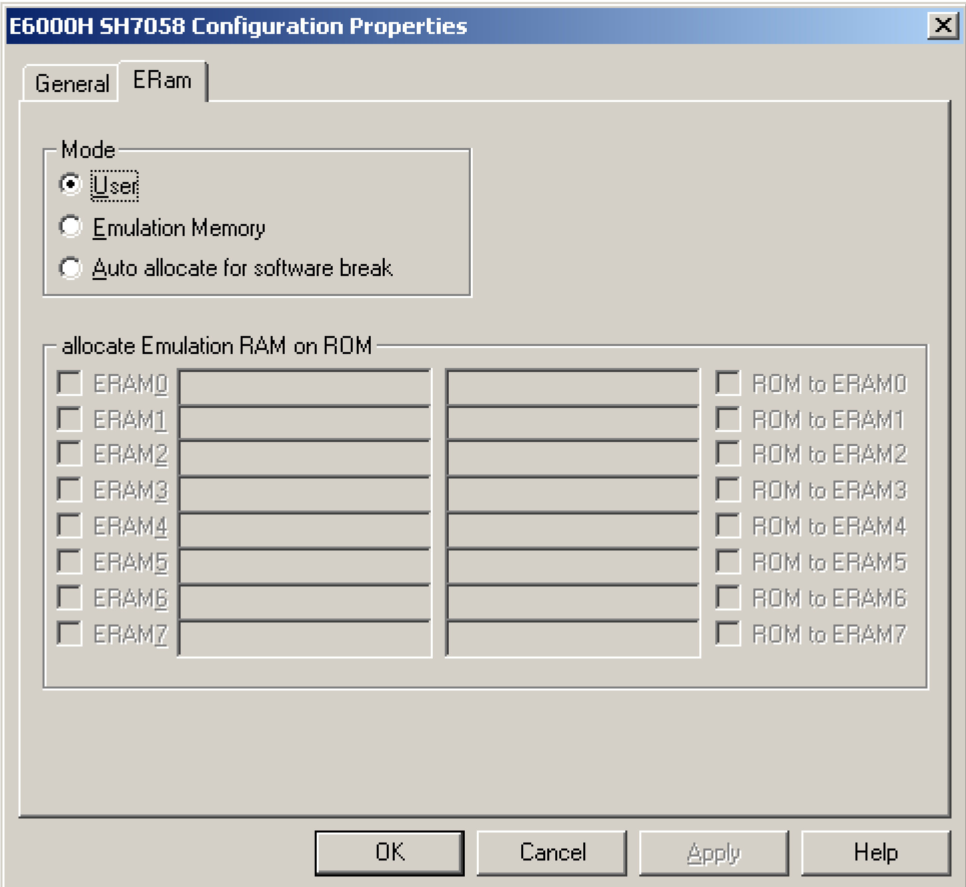



**Figure 4.3 [ERam] Page (before Setting)**

The emulator can allocate emulation memory to eight ROM areas in 4-kbyte units.

When [Emulation Memory] in the [Mode] group box is selected, the [allocate Emulation RAM on ROM] group box is enabled. In the tutorial program, allocate H’00000000 to H’0000FFF and H’00001000 to H’00001FFF as the emulation memory area.

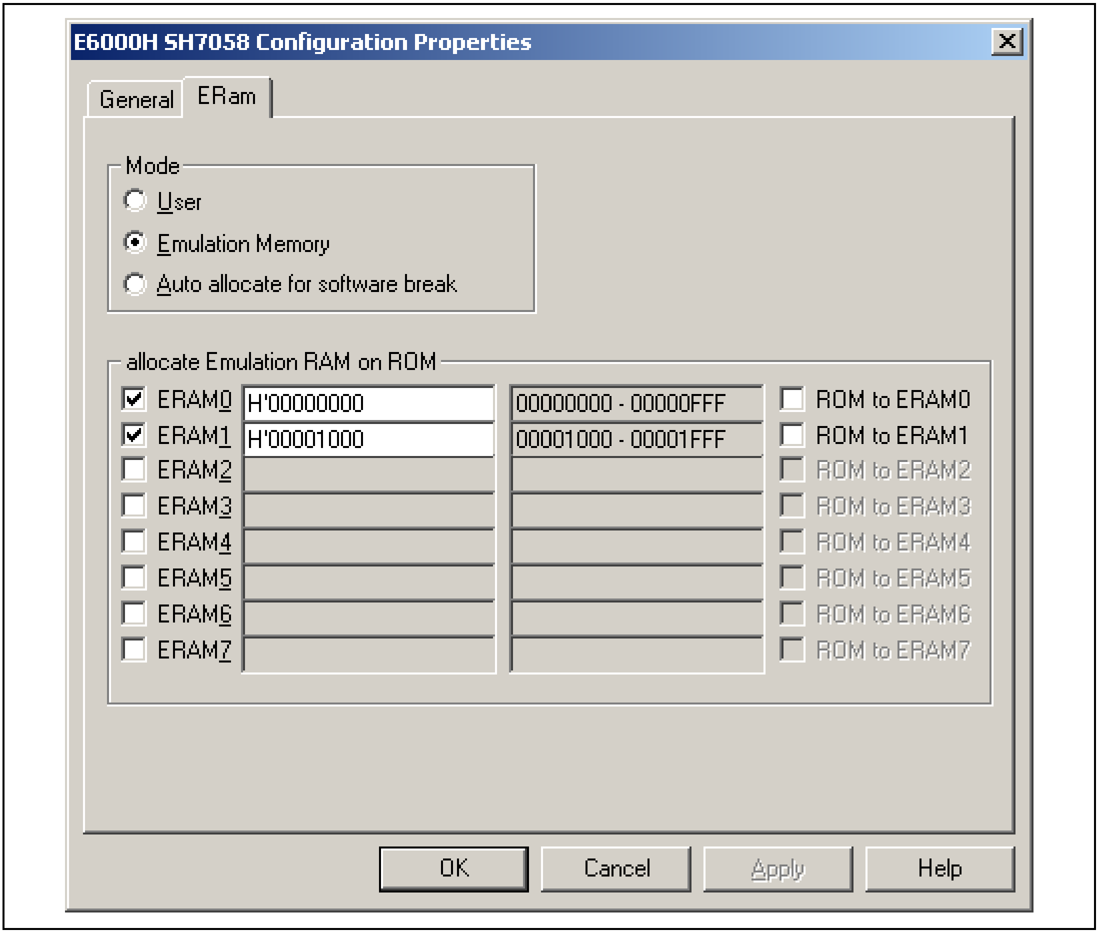



**Figure 4.4 Allocating the Emulation Memory**

- Check the [ERAM0] check box and enter H'00000000 in the address box. Similarly, check the [ERAM1] check box and enter H'00001000 in the address box. The value in the gray address box shows the range that the memory has been allocated.

- Click the [OK] button after values have been entered.

For details on the allocation of emulation memory, refer to the [ERam] Page in section 5.2.1.

## 4.4 Downloading

### 4.4.1 Downloading the Sample Program

Download the sample program in the ELF/DWARF2 format to be debugged.

- Select [Load Program...] from the [File] menu. The [Load Program] dialog box is displayed.
- Click the [Browse…] button. The [Open] dialog box will be displayed.
- Select the file TUTORIAL.ABS, and click the [Open] button.

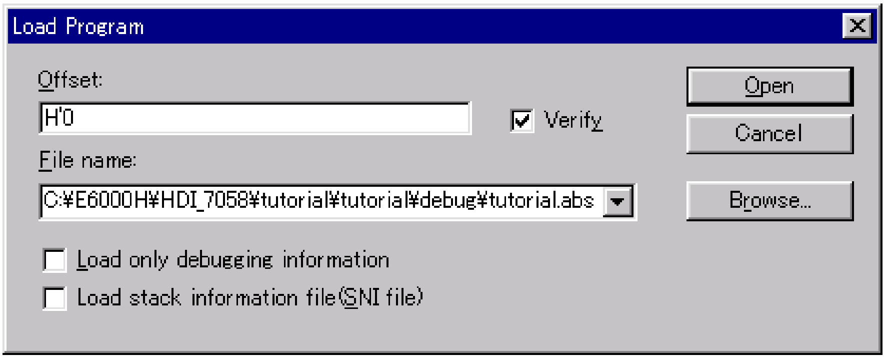



**Figure 4.5 [Load Program] Dialog Box**

- Click the [Open] button in the [Load Program] dialog box.

The following dialog box will be displayed when the program completes loading. In the dialog box, the address where the program was loaded is displayed.

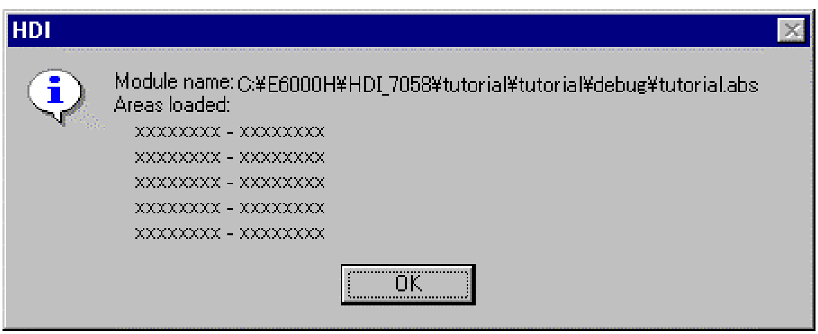



**Figure 4.6 [HDI] Dialog Box**

- Click the [OK] button.

### 4.4.2 Displaying the Source Program

The [Source] window allows the user to display the C/C++ language source program, set breakpoints, execute the program, and select variables, so the user can debug a program at the source level.

- Select [Source...] from the [View] menu.

The [Open] dialog box is displayed.

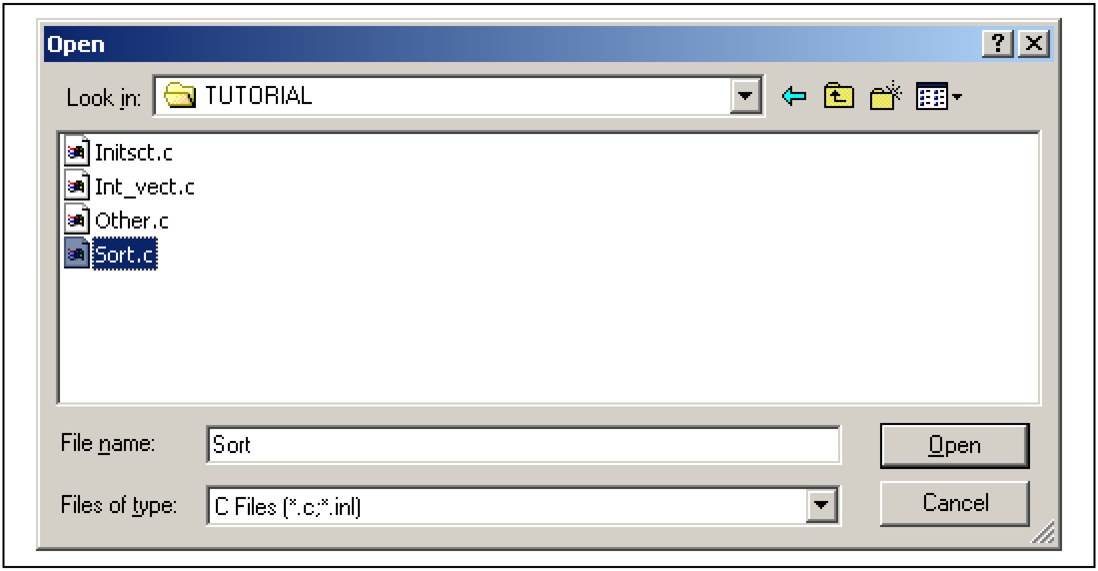



**Figure 4.7 [Open] Dialog Box**

- Select [Sort.c] and click the [Open] button. The [Source] window is displayed. If necessary, select whatever font or size you like, by selecting the [Font…] option from the [Customize] submenu in the [Setup] menu.

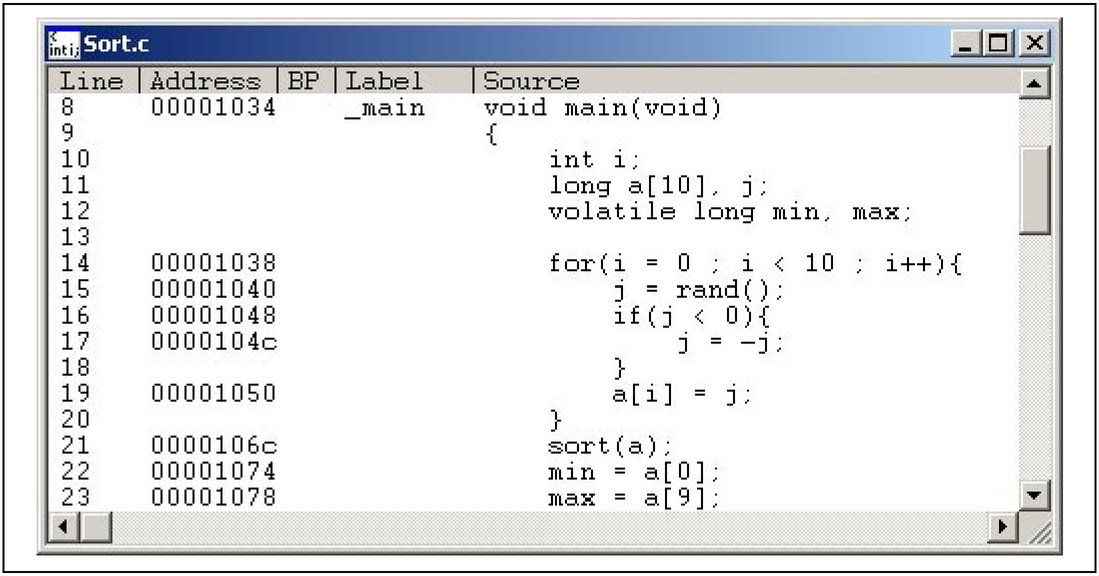



**Figure 4.8 [Source] Window (Displaying the Source Program)**

## 4.5 Setting the Program Counter Breakpoints

A breakpoint is one of the debugging functions.

The [Source] window provides a very simple way of setting program counter (PC) breakpoints. For example, to set a breakpoint at the line that contains the `sort` function call:

- Double-click the [BP] column on the line containing the `sort` function call.

The ● will be displayed on the line containing the `sort` function to show that a software breakpoint is set at that address.

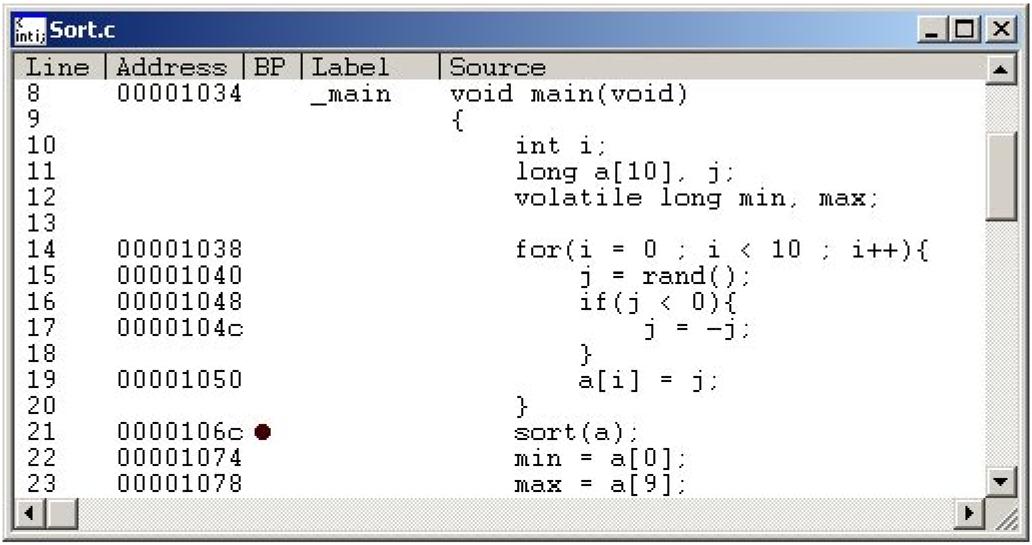



**Figure 4.9 [Source] Window (Setting a PC Breakpoint)**

The emulator has many break functions. For details, refer to section 5.6, Break Functions.

## 4.6 Executing the Program

- To execute the program, select [Reset Go] from the [Run] menu, or click the [Reset Go] button on the toolbar.

The program will be executed up to the breakpoint that has been set, and will then stop. The line where the program has halted will be highlighted in the [Source] window.

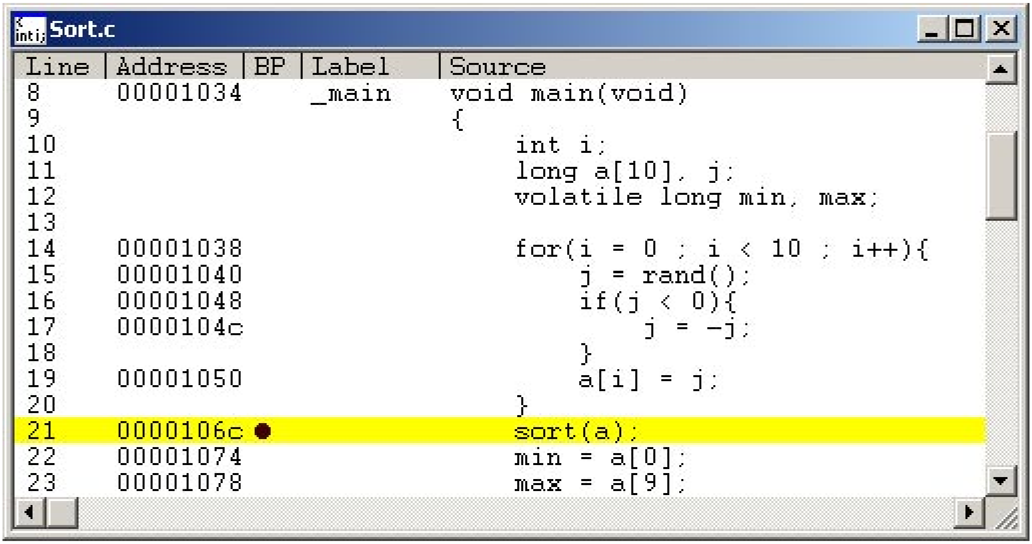



**Figure 4.10 [Source] Window (Break State)**

The user can see the cause of the last break through the [Platform] sheet in the [System Status] window.

- Select [Status] from the [View] menu. The [System Status] window is displayed.
- Select [Platform] sheet from the [System Status] window.

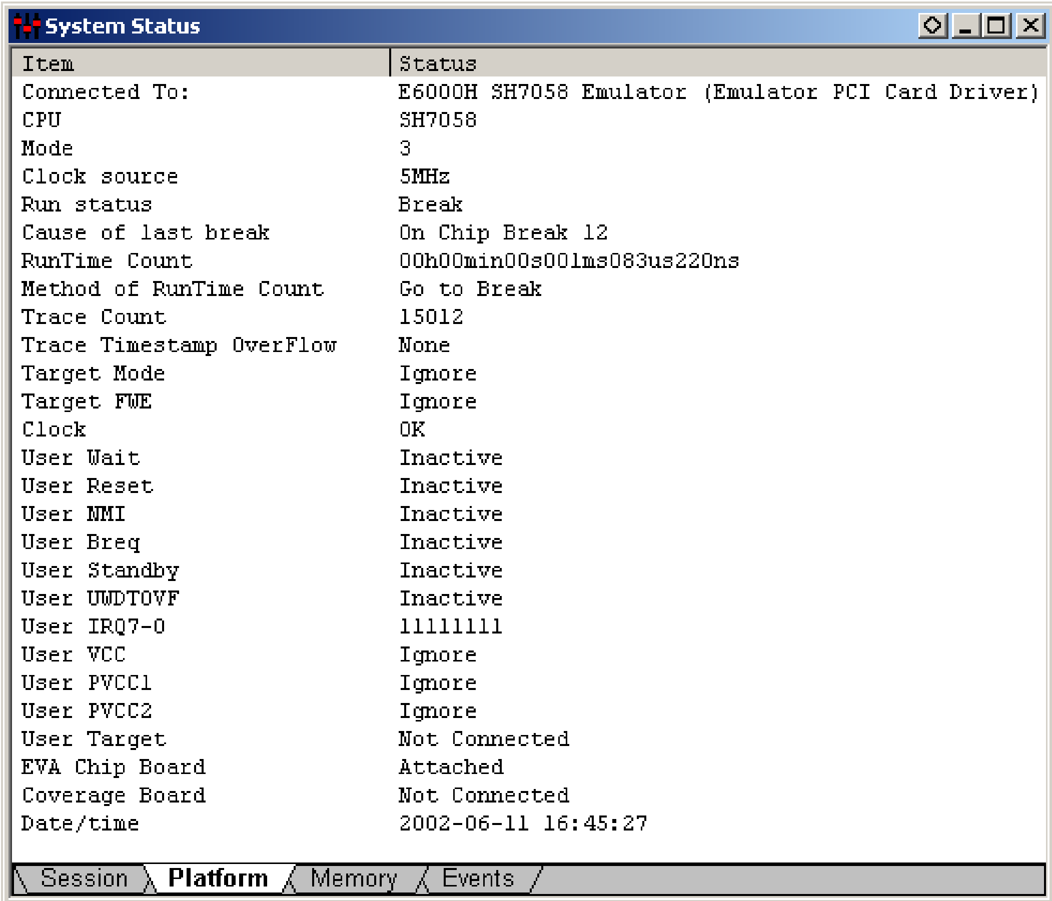



**Figure 4.11 [System Status] Window**

The [Cause of last break] line shows that the cause of the break is `On Chip Break 12`.

For details on program execution, refer to section 5.3, Program Execution.

## 4.7 Reviewing Breakpoints

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

- Select [Breakpoints] from the [View] menu. The [Breakpoints] window is displayed. The contents of the breakpoint set will be displayed. A ● will be displayed in the [Enable] column.

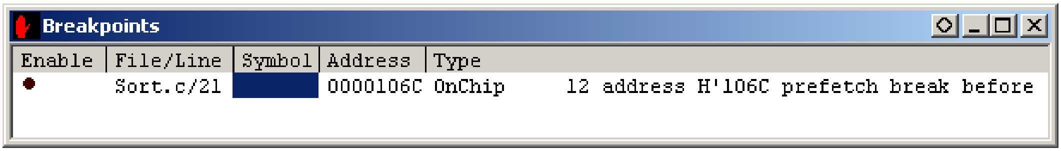



**Figure 4.12 [Breakpoints] Window**

The [Breakpoints] window also allows the user to change breakpoints, set new breakpoints, and delete breakpoints.

- Close the [Breakpoints] window.

## 4.8 Viewing Memory

The user can view the contents of a memory block in the [Memory] window. For example, to view the memory contents corresponding to the external variable array a:

- Select [Memory...] from the [View] menu. The [Open Memory Window] dialog box is displayed.
- Input a in the [Address] edit box, and set the [Format] combo box as [Long Word].

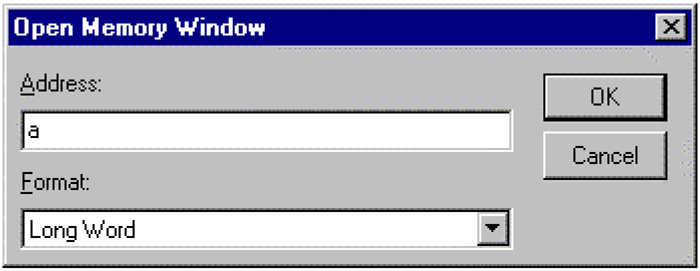



**Figure 4.13 [Open Memory Window] Dialog Box**

- Click the [OK] button. The [Long Memory] window showing the specified area of memory is displayed.

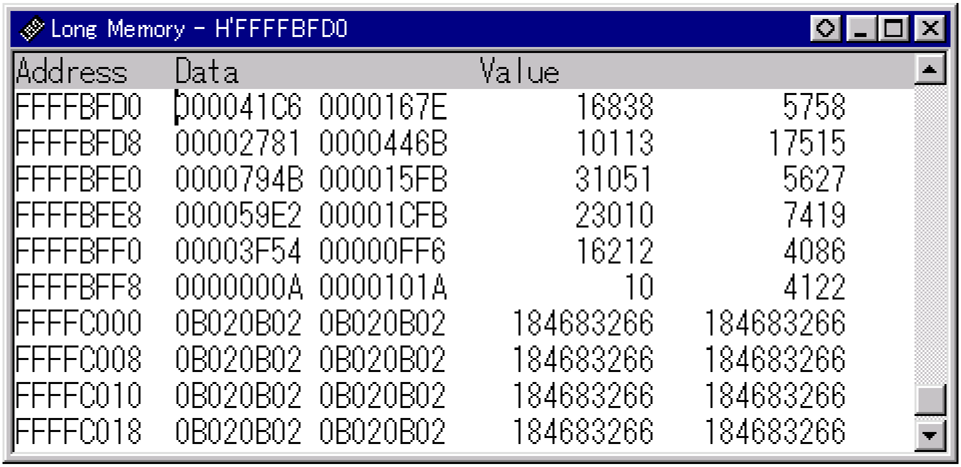



**Figure 4.14 [Long Memory] Window**

## 4.9 Watching Variables

As the user steps through a program, it is possible to watch the values of variables used in the program. For example, to check the contents of the long-type array a declared at the beginning of the program, use the following procedure:

- Click the left of array a displayed in the [Source] window to position the cursor.
- Click the [Source] window with the right mouse button, and select [Instant Watch...] from a pop-up menu.

The [Instant Watch] dialog box is displayed.

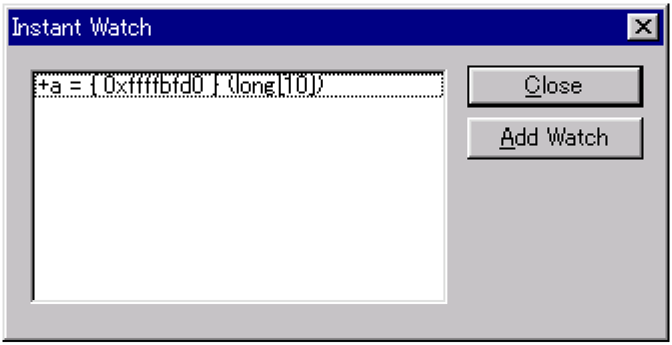



**Figure 4.15 [Instant Watch] Dialog Box**

- Click the [Add Watch] button to add a variable to the [Watch Window] window.

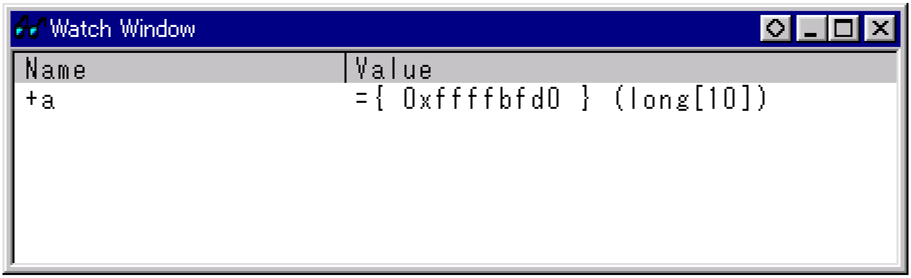



**Figure 4.16 [Watch Window] Window (Displaying the Array)**

The user can also add a variable to the [Watch Window] window by specifying its name.

- Click the [Watch Window] window with the right mouse button and select [Add Watch…] from the pop-up menu.

The [Add Watch] dialog box is displayed.

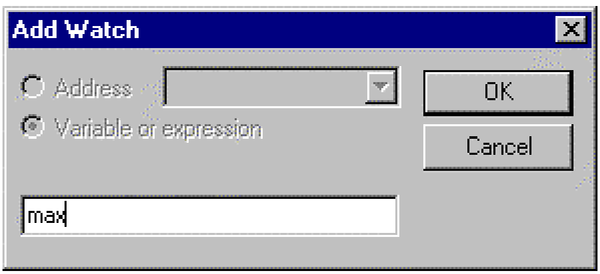



**Figure 4.17 [Add Watch] Dialog Box**

- Input variable ***max*** and click the [OK] button.

The [Watch Window] window will now also show the Volatile long-type variable `max`.

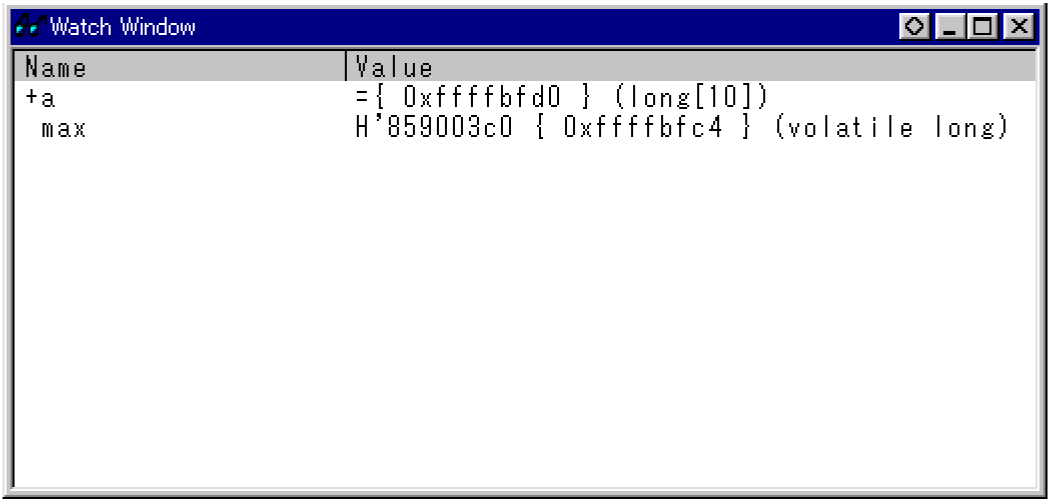



**Figure 4.18 [Watch Window] Window (Displaying the Variable)**

- Double-click the + symbol to the left of array a in the [Watch Window] window to expand the variable and watch all the elements in the array.

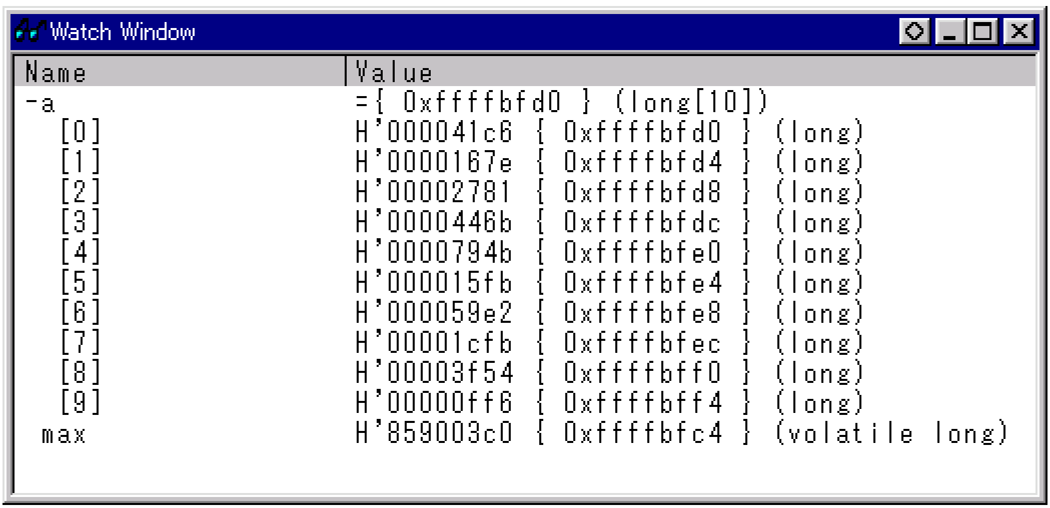



**Figure 4.19 [Watch Window] Window (Displaying Array Elements)**

## 4.10 Stepping Through a Program

The HDI provides various step commands that allow efficient program debugging. For details on step function, refer to section 5.5, Step Functions.

**Table 4.2 Step Command**

| Command | Description |
|---|---|
| Step In | Steps through the statements in a function by each line, or steps through assembly statements by each instruction. (For a line that calls a function, execution stops at the first line of the called function.) |
| Step Over | Steps through the statements in a function by each line, or steps through assembly statements by each instruction. (For a line that calls a function, the whole of the called function is executed in a single step.) |
| Step Out | Steps out of a function, and stops at the next line that calls the function in the program. |
| Step… | Steps the specified counts repeatedly at a specified rate. |

Before executing program stepping, confirm that the program is executed up to the `sort` function line at address H’0000106c.

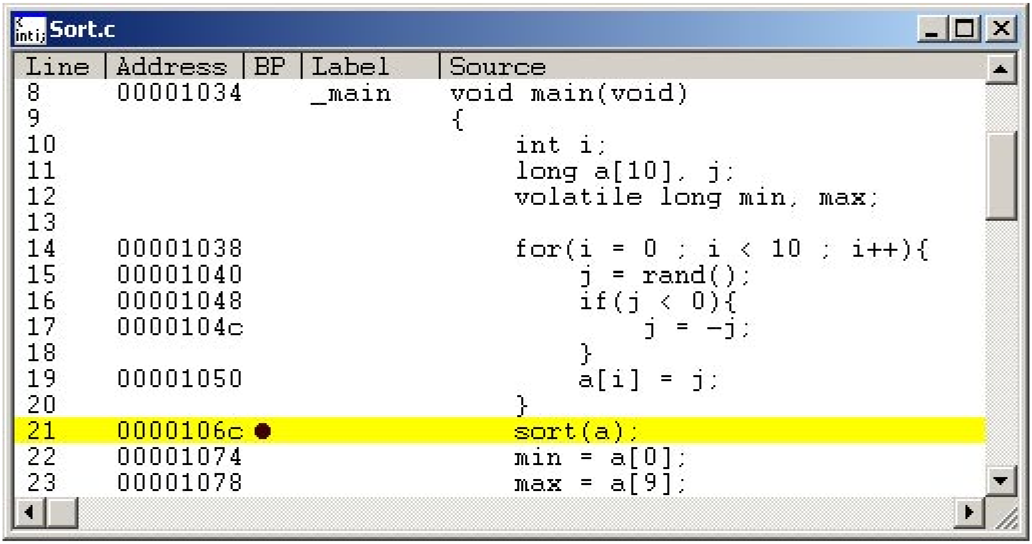



**Figure 4.20 [Source] Window (Step Execution)**

### 4.10.1 Executing [Step In] Command

The [Step In] command steps into the called function and stops at the first line of the function.

- To step into the `sort` function, select [Step In] from the [Run] menu, or click the [Step] button  in the toolbar.

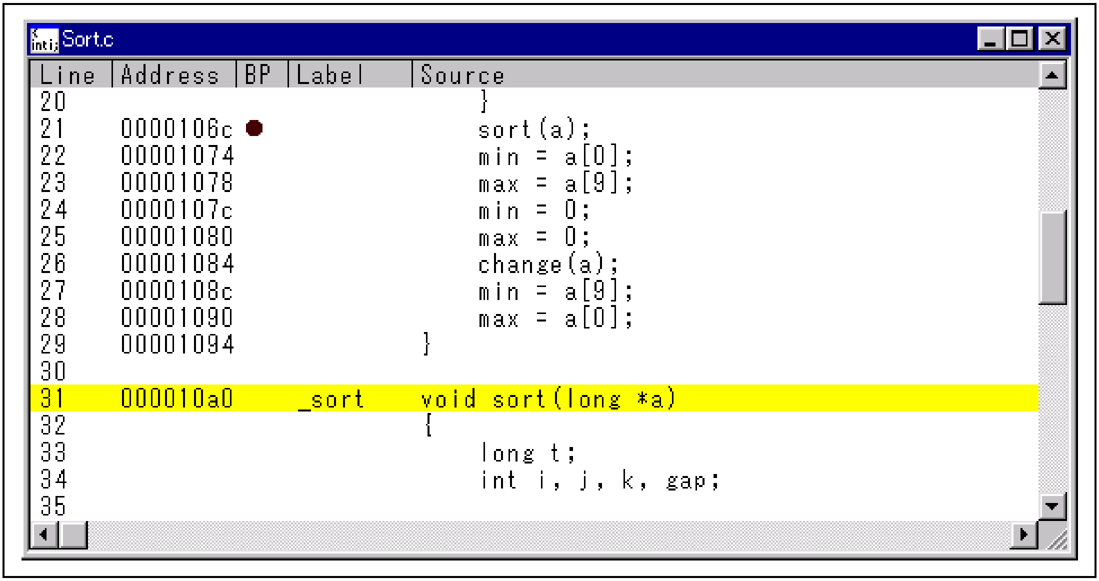



**Figure 4.21 [Source] Window (Step In)**

The highlighted line moves to the first line of the `sort` function in the [Source] window.

### 4.10.2 Executing [Step Out] Command

The [Step Out] command steps out of the called function and stops at the next line that called the function in the program.

- To step out of the `sort` function, select [Step Out] from the [Run] menu, or click the [Step Out] button in the toolbar.

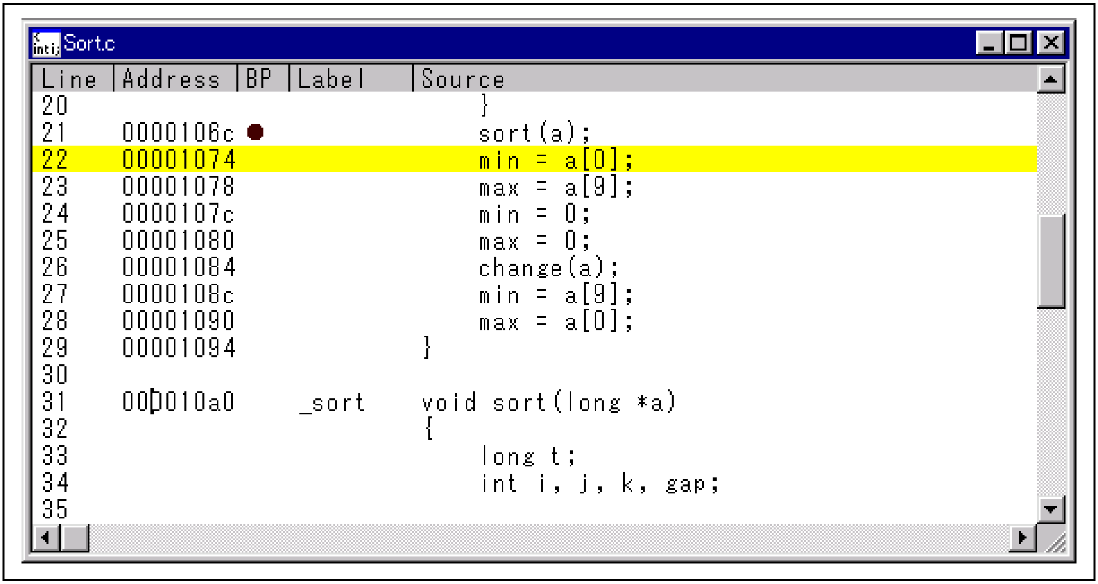



**Figure 4.22 [Source] Window (Step Out)**

The data of array `a` displayed in the [Watch Window] window is sorted in ascending order.

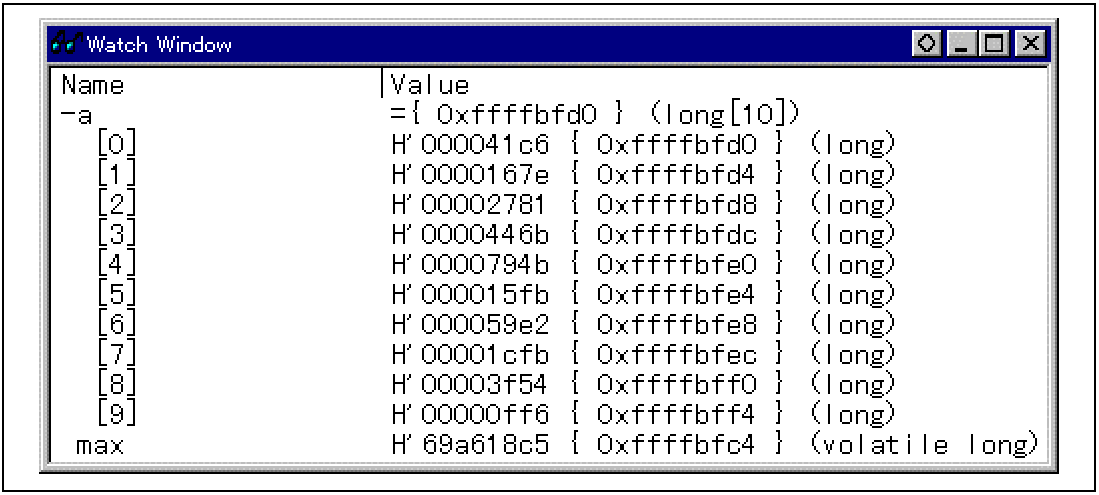



**Figure 4.23 [Watch Window] Display Example (1)**

To execute two steps, use the [Step In] command twice.

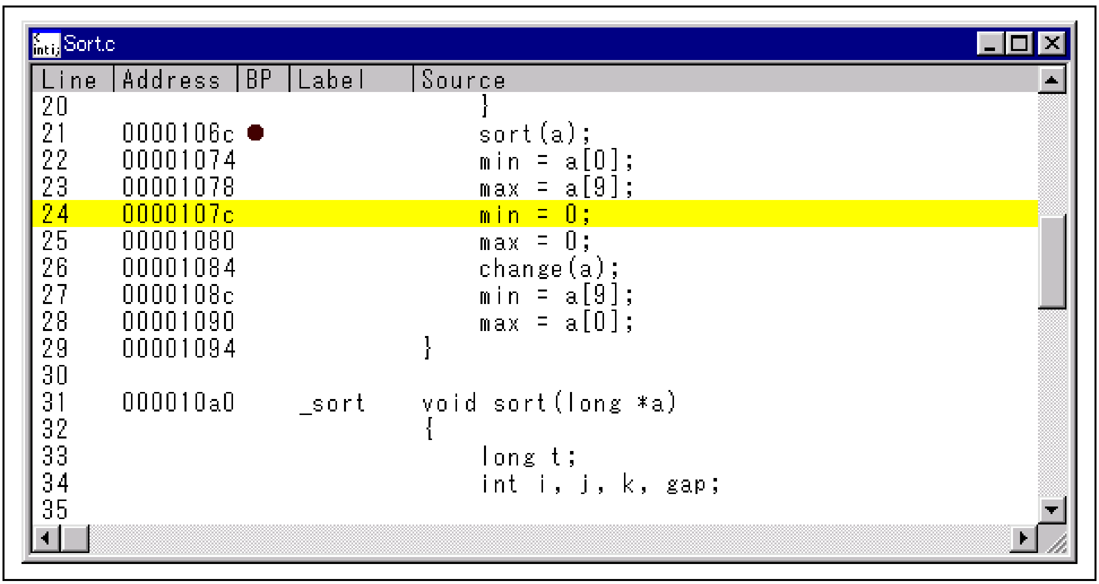



**Figure 4.24 [Source] Window (Step Out → Step In)**

The value of variable `max` displayed in the [Watch Window] window is changed to the maximum data value.

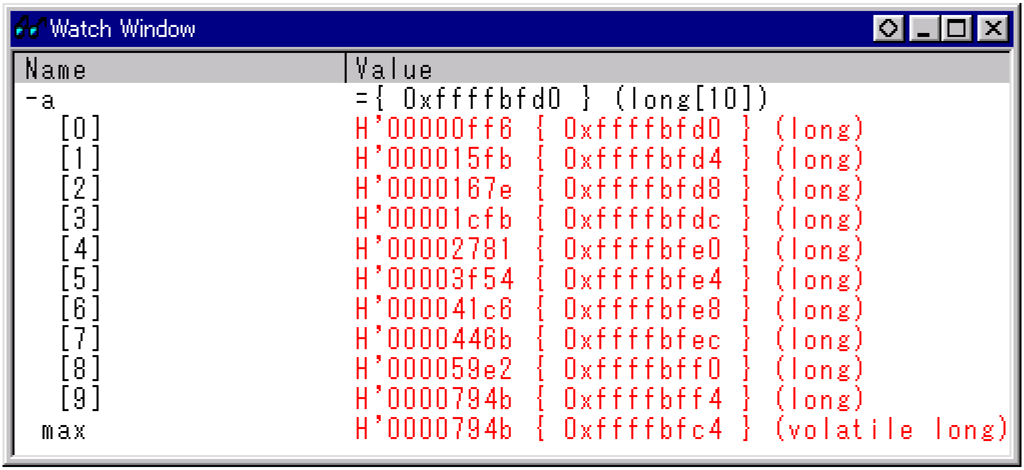



**Figure 4.25 [Watch Window] Display Example (2)**

### 4.10.3 Executing [Step Over] Command

The [Step Over] command executes a line that calls a function as a single step.

- Before executing the [Step Over] command, execute two steps up to a line that calls the `change` function by using the [Step In] command twice.

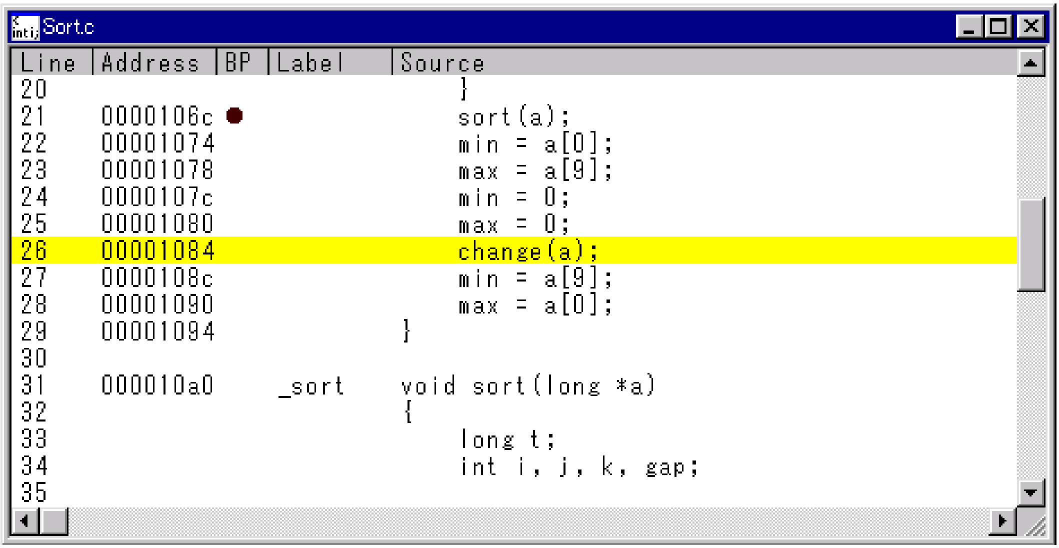



**Figure 4.26 [Source] Window (Before Step Over Execution)**

- Select [Step Over] from the [Run] menu, or click the [Step Over] button in the toolbar.

A line that calls the `change` function is executed as a single step, and execution stops at the next line in the program.

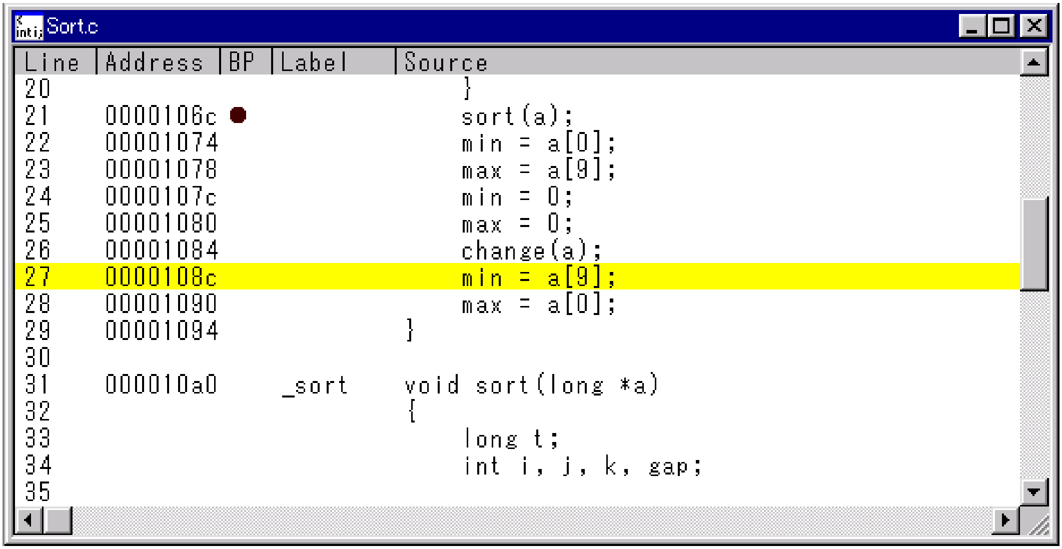



**Figure 4.27 [Source] Window (After Step Over Execution)**

When the last line of the `change` function is executed, the data of array `a`, which is displayed in the [Watch Window] window, is sorted in descending order.

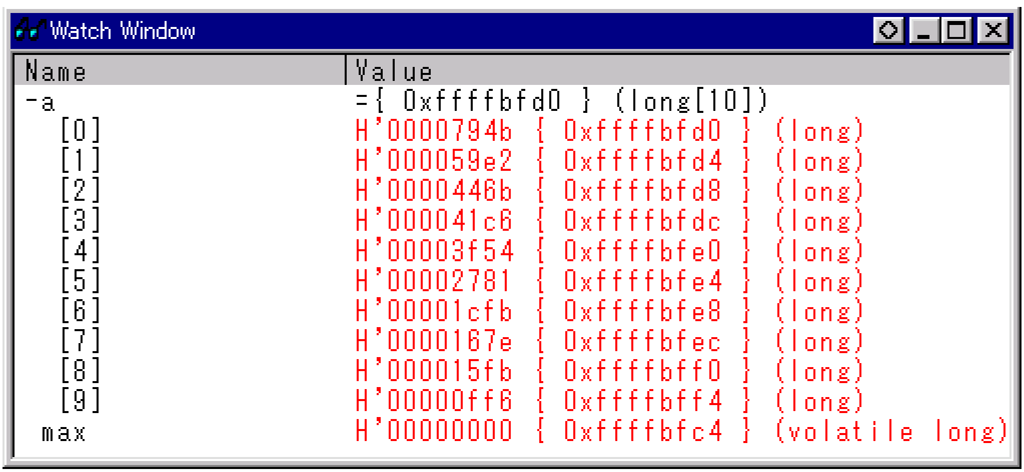



**Figure 4.28 [Watch Window] Display Example (3)**

## 4.11 Displaying Local Variables

The user can see local variables in a function using the [Locals] window. For example, the local variables in the `main` function, which declares five local variables; a, min, max, j, and i will be examined.

Select [Locals] from the [View] menu. The [Locals] window is displayed. When no local variables exist, the [Locals] window is empty.

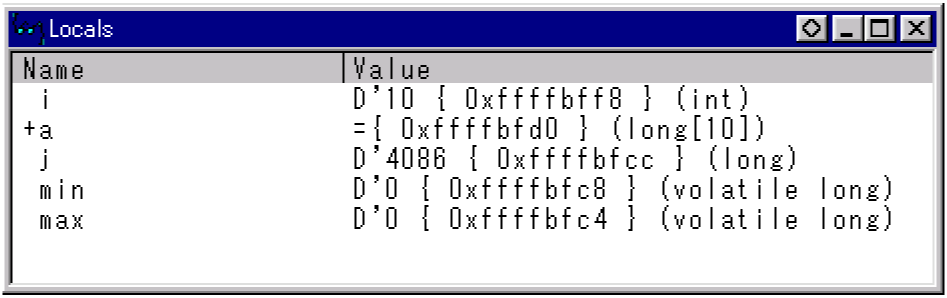



**Figure 4.29 [Locals] Window**

Double-click the + symbol to the left of array a in the [Locals] window to expand the variable and watch all the elements in the array.

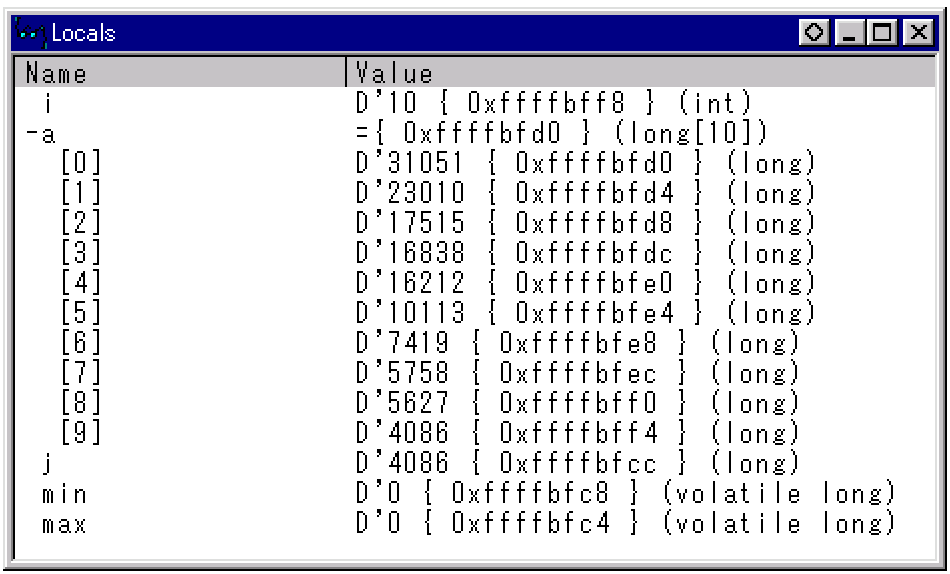



**Figure 4.30 [Locals] Window (Displaying Array a Elements)**

## 4.12 Saving and Loading the Session

The information set to the HDI windows and dialog boxes can be saved as a session file. Loading this session file at HDI initialization will allow debugging to be resumed from the same state as the last session.

To save the session file, select [Save Session As...] from the [File] menu. At this time, the window for specifying the file name is displayed. Input the session file name in the window and click the [Save] button.

To load the session file, select [Load Session...] from the [File] menu.

A session file can be automatically saved and loaded by setting the [HDI Options] dialog box of [Options...] in the [Setup] menu.

To automatically save the session file, click on the [Save session automatically] radio button in the [Session] page. The dialog box for specifying the file at HDI termination is then displayed. Specifying the file name enables session information to be automatically saved to the file from the following HDI termination.

To automatically load the session file, enable [Load last session on startup] check box in the [Session] page. The session information is automatically loaded.

For more details on sessions and a setting method, refer to the Hitachi Debugging Interface User's Manual on the CD-R.

# Section 5 Emulator Functions

## 5.1 Introduction

The following is a full description of the emulator's functions, including those that were not described in section 4, Tutorial.

Table 5.1 is a list of the emulator functions that are described in this section.

**Table 5.1 Emulator Functions**

| Section | Function | Description |
|---|---|---|
| 5.2 | Operating mode setting | Sets the operating conditions for the emulator |
| 5.3 | Execution | Emulation |
| 5.4 | Display of cause for termination and operating status | Displays the cause for emulation termination and the MCU's operating status |
| 5.5 | Step execution | Emulation with step execution |
| 5.6 | Break | Break functions |
| 5.7 | Realtime trace | Acquires, searches for, and displays tracing information |
| 5.8 | Execution time measurement | Measures the total execution time of the user program |
| 5.9 | Performance analysis | Performance and profile data measurement functions |
| 5.10 | Information display | Displays the various items set in each dialog box |
| 5.11 | Trigger output | Trigger outputs |
| 5.12 | Stack trace | Displays the history of called functions |
| 5.13 | Displaying memory and auto-update memory | Displays memory in one of two ways: by parallel access or the auto-update memory display function (the contents of a range of locations is displayed and the display is updated every 500 ms during execution of the user program) |
| 5.14 | Entering masks | Formats for the input of masks |
| 5.15 | Extended function for [Source] window | Extended functions for the [Source] window of the HDI |

## 5.2 Setting the Emulator's Operating Conditions

The user must set the operating conditions before using the emulator. Table 5.2 lists the settings.

**Table 5.2 Setting the Emulator's Operating Conditions**

| Setting | Item | Description |
|---|---|---|
| Emulator settings<br>(Note: Use the [Configuration] dialog box to make these settings.) | MCU setting | Selects the MCU to be used. |
| | Operating mode setting | Selects the MCU operating mode. |
| | Clock setting for MCU | Selects the clock to be supplied to the MCU. |
| | H-UDI (JTAG) clock setting | Selects the clock to be supplied to the H-UDI (JTAG) clock. |
| | Minimum measurement time setting for the execution-time measurement counter | Selects the measurement unit time for the measurement counter of the execution time. |
| | Operation setting when using the on-chip flash memory | Sets the operation when the on-chip flash memory is used. |
| | Operation setting of the emulator function | Enables the use of the emulator function.<br>Interrupt during step execution<br>Selection of the use of the AUD<br>Multibreak mode |
| | User signal input setting | Enables the use of the user signal input.<br>Enabling the input of the RES signal<br>Enabling the input of the NMI signal<br>Enabling the input of BREQ/WAIT |
| | Threshold setting for Bus Time Out | Sets the threshold of Bus Time Out. |
| | Threshold setting for user Vcc | Sets the threshold of user Vcc. |
| | Driver setting | Opens the dialog box to change the driver. |
| ERAM setting<br>(Note: Use the [ERam] page to make this setting.) | Mode selection | Selects the mode for the emulation memory.<br>Selecting the user RAM mode<br>Selecting the emulation memory mode<br>Selecting the software-break automatic setting mode for the on-chip ROM area (on-chip flash memory area) |

### 5.2.1 [Configuration] Dialog Box

Select [Configure Platform...] from the [Setup] menu to open the [Configuration] dialog box.

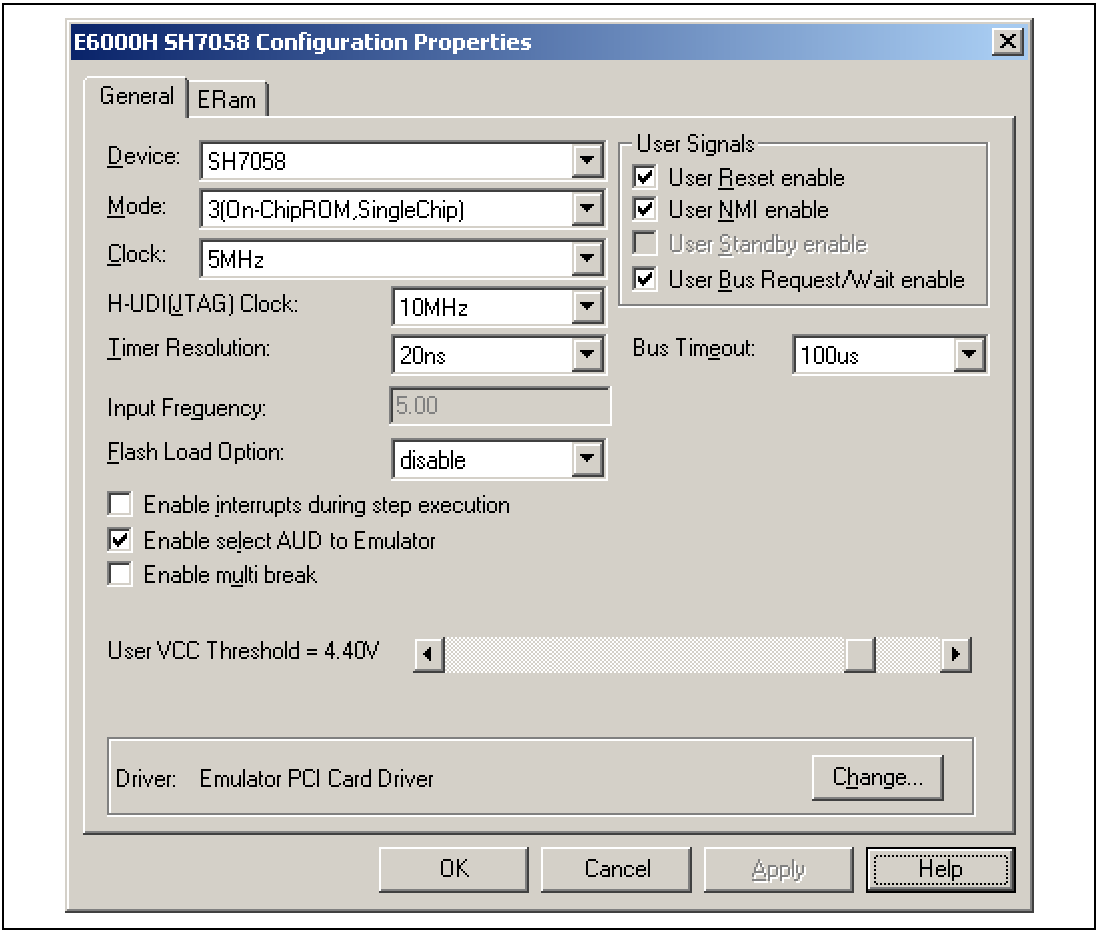



**Figure 5.1 [Configuration] Dialog Box**

Use this dialog box to set the emulation conditions for the emulator.

**Table 5.3 [Configuration] Dialog Box**

| Page | Description |
|---|---|
| [General] | Selects the MCU, operating clock, H-UDI (JTAG) clock, the minimum measurement time for the execution-time measurement counter, operation when using the on-chip flash memory, and thresholds of bus time out and user Vcc, sets the mode and the clock when programming the on-chip flash memory, enables emulator operation and the input of the user signal, and displays the dialog box to change the driver |
| [ERam] | Selects the mode and sets the address for allocating the emulation memory and automatic copy of the on-chip flash memory to the emulation memory |

Each page of the [Configuration] dialog box is described below.

**[General] Page:**

Use this page to select the MCU, operating clock, H-UDI (JTAG) clock, the minimum measurement time for the execution-time measurement counter, and threshold of user Vcc, set the mode, enable emulator operation and the input of the user signal, and display the dialog box to change the driver.

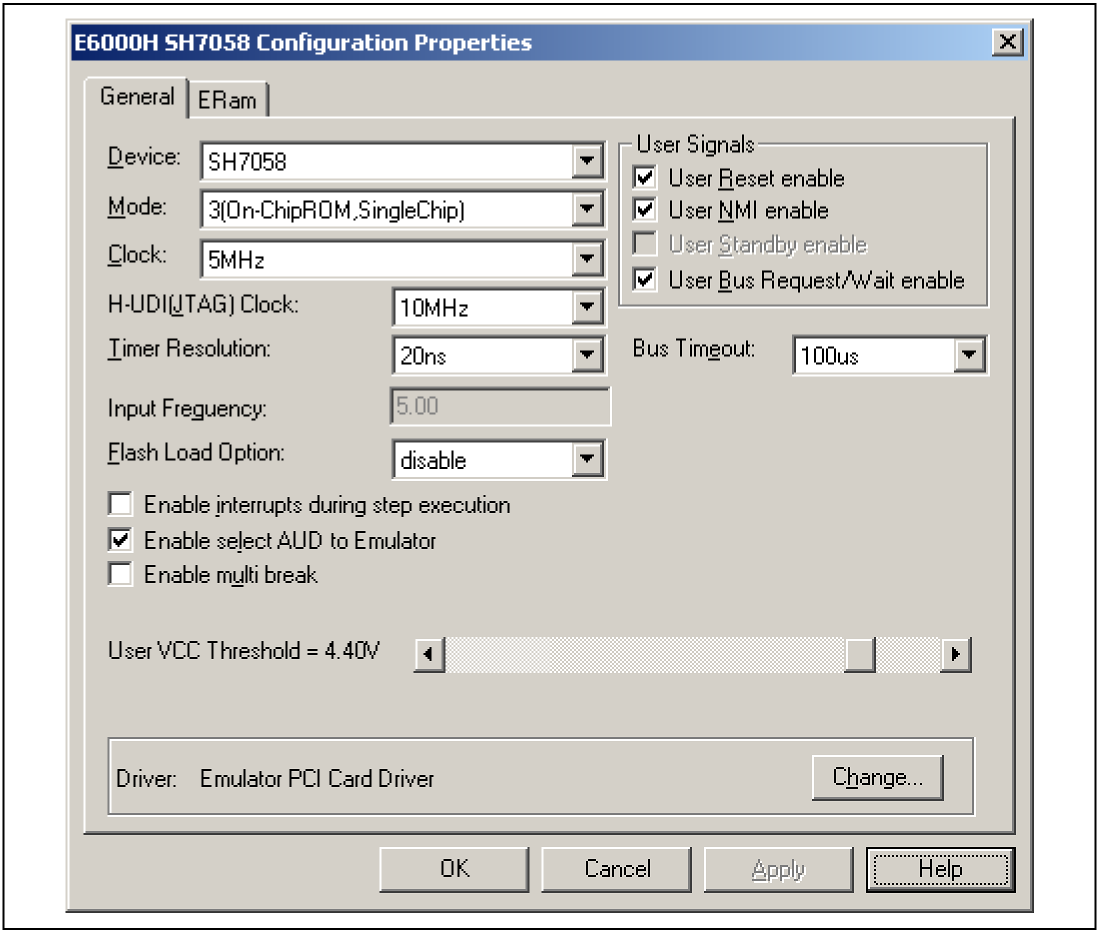



**Figure 5.2 [Configuration] Dialog Box ([General] Page)**

**Table 5.4 [General] Page**

| Option | Item | Description |
|---|---|---|
| [Device] | Selects the MCU | |
| | SH7xxx | Selects the MCU for emulation |
| [Mode] | Selects the operating mode for emulation | |
| | 0 (8-bit bus) | MCU expansion mode 0 |
| | 1 (16-bit bus) | MCU expansion mode 1 |
| | 2 (on-chip ROM) | MCU expansion mode 2 |
| | 3 (on-chip ROM, single chip) | MCU single chip mode |
| | Target | Mode that has been set on the user system |
| [Clock] | Selects the clock to be supplied to the MCU | |
| | Emulator Clock (x MHz) | Internal clock (x = 5 MHz or 10 MHz) |
| | Target clock | Clock signal for the user system |
| | X’TAL | Crystal oscillator on the evaluation chip |
| [H-UDI (JTAG) Clock] | Selects the clock to be input to the H-UDI | |
| | Frequency | Frequency x (x = 10 MHz, 15 MHz, or 20 MHz) |
| [Timer Resolution] | Selects the execution time or the minimum measurement time width of the counter. The value set here affects the display of the runtime and performance. | |
| | Runtime measurement | 52 us, 1.6 us, or 20 ns |
| | Clock counter measurement | CLOCK, CLOCK/2, CLOCK/4, or CLOCK/8 |
| [Input Frequency] | Sets the clock when programming the on-chip flash memory. The clock is input when X’tal or Target is specified for [Clock]. The value must be input up to two places of decimals between 5.00 MHz to 10.00 MHz. | |
| [Flash Load Option] | Sets the operation when programming the on-chip flash memory.<br>disable: Programming disabled<br>update: Programs the on-chip flash memory without initialization<br>erase: Initializes and programs the on-chip flash memory of the target block (4 kbytes)<br>all erase: Initializes and programs all the on-chip flash memory | |

**Table 5.4 [General] Page (cont)**

| Option | Item | Description |
|---|---|---|
| [User Signals] | User Reset enable | Enables the RES signal from the user system |
| | User NMI enable | Enables the NMI signal from the user system |
| | User Bus Request/Wait enable | Enables BREQ or Wait from the user system |
| [Enable interrupts during step execution] | Specifies whether or not the interrupt is generated during step execution. | |
| [Enable select AUD to Emulator] | Specifies this option when the AUD is used by the emulator. Remove the check mark when used by the user. | |
| [Enable multi break] | Specifies whether or not the multibreak is enabled. | |
| [User VCC Threshold] | Detects the lowered user-system voltage. When the mode is Target, the System Status window informs that the user VCC is lowered than the specified value. When the mode is selected, the initial value is changed. This option also informs when the recommended voltage range is exceeded depending on the mode. | |
| [Driver…Change] | Specifies the emulator driver. When this dialog box is opened, the emulator is initialized even if the setting is not changed. | |

Note: When the check mark for [Enable select AUD to Emulator] is removed and the AUD function is used by the user, memory access is disabled during program execution.

**[ERam] Page:**

Use this page to select the used mode of emulation memory and to allocate the emulation memory. The emulation memory is mapped to addresses from H’FFFE8000 to H’FFFEFFFF, and used for on-chip ROM emulation. The addresses H’FFFE8000 to H’FFFEFFFF can also be used for on-chip RAM. In the actual MCU, the emulation memory is reserved and not installed.

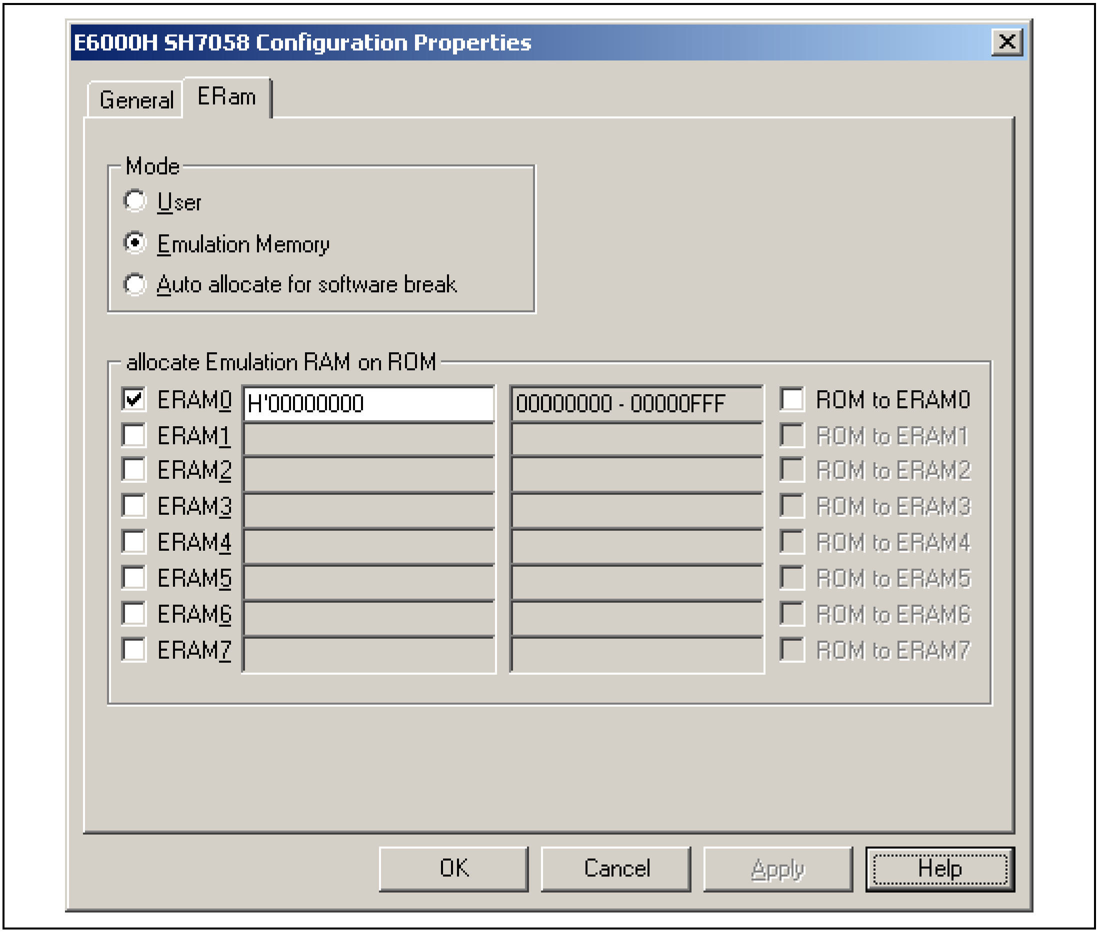



**Figure 5.3 [Configuration] Dialog Box ([ERam] Page)**

**Table 5.5 [ERam] Page**

| Option | Description | |
|---|---|---|
| [Mode] | Selects the ERAM mode to be used. | |
| | [User] | Uses the emulation memory for on-chip RAM. Addresses from H'FFFE8000 to H'FFFEFFFF can be used as the on-chip RAM. |
| | [Emulation Memory] | Performs on-chip ROM emulation. The ROM area (on-chip flash memory area) to be emulated is allocated to the address by the [Emulation RAM]. |
| [Auto allocate for software break] | Sets the mode to automatically emulate the on-chip ROM when a software break is set on the on-chip ROM. | |
| [allocate Emulation RAM on ROM] | When the emulation memory is selected with [Mode], selects the ERAM area and sets the ROM area address to be emulated. | |
| | ERAM 0 to 7 | Selects the ERAM to be allocated. |
| | Address input | Inputs the start address of ROM to be emulated. When a value is set outside the 4-kbyte boundary, round down the value below the boundary. |
| | Set range address display | Displays the address range of ROM to be emulated. |
| | Checkbox for automatic writing to the on-chip flash memory | When this checkbox is checked, the contents of the on-chip flash memory are read and written to the emulation memory when the emulation memory is allocated. When the user program is downloaded to the on-chip flash memory in this state, the program is also downloaded to the emulation memory. |

Note: Stop the DMAC operation when the emulation memory is set.

**[User] Mode:**

The emulation memory is used as the on-chip RAM. The addresses from H'FFFE8000 to H'FFFEFFFF can be used as the on-chip RAM. Figure 5.4 shows the selection in [User] mode, and figure 5.5 shows the emulation memory diagram when the [User] mode is selected.

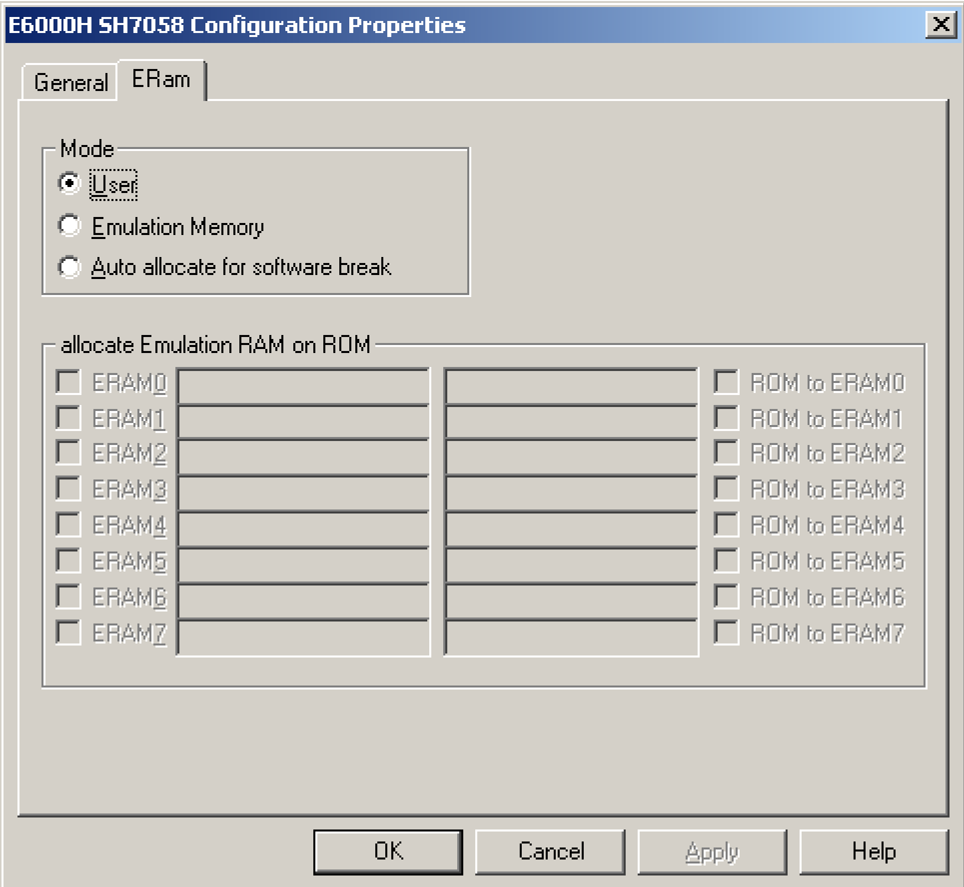



**Figure 5.4 [User] Mode in [ERam] Page**

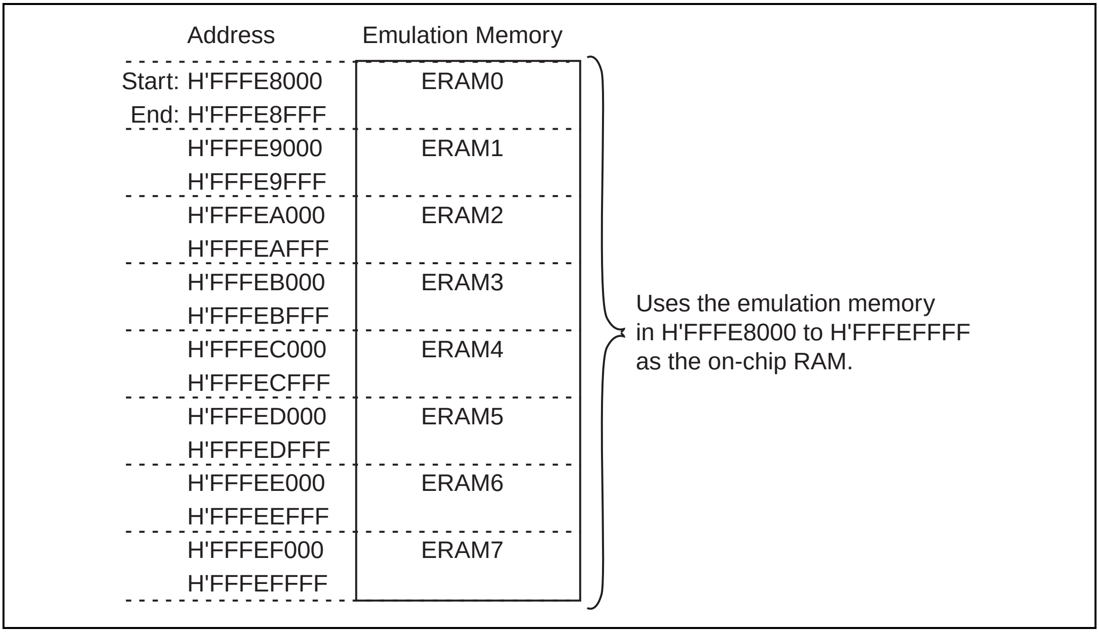



**Figure 5.5 Diagram for Emulation Memory in [User] Mode**

**[Emulation Memory] Mode:**

The ERAM area can be used for the emulation of the ROM area (on-chip flash memory area). The emulation memory can be allocated in 4-kbyte boundaries. The ROM contents can be downloaded to ERAM automatically, and the ERAM contents can also be downloaded to the ROM area (on-chip flash memory) automatically.

(a) Emulation Memory Allocation

Figure 5.6 shows the display when the ROM area from H’00000000 to H’00000FFF, and H’00001000 to H’00001FFF are allocated to ERAM0 and ERAM1, respectively. Figure 5.7 shows the diagram.

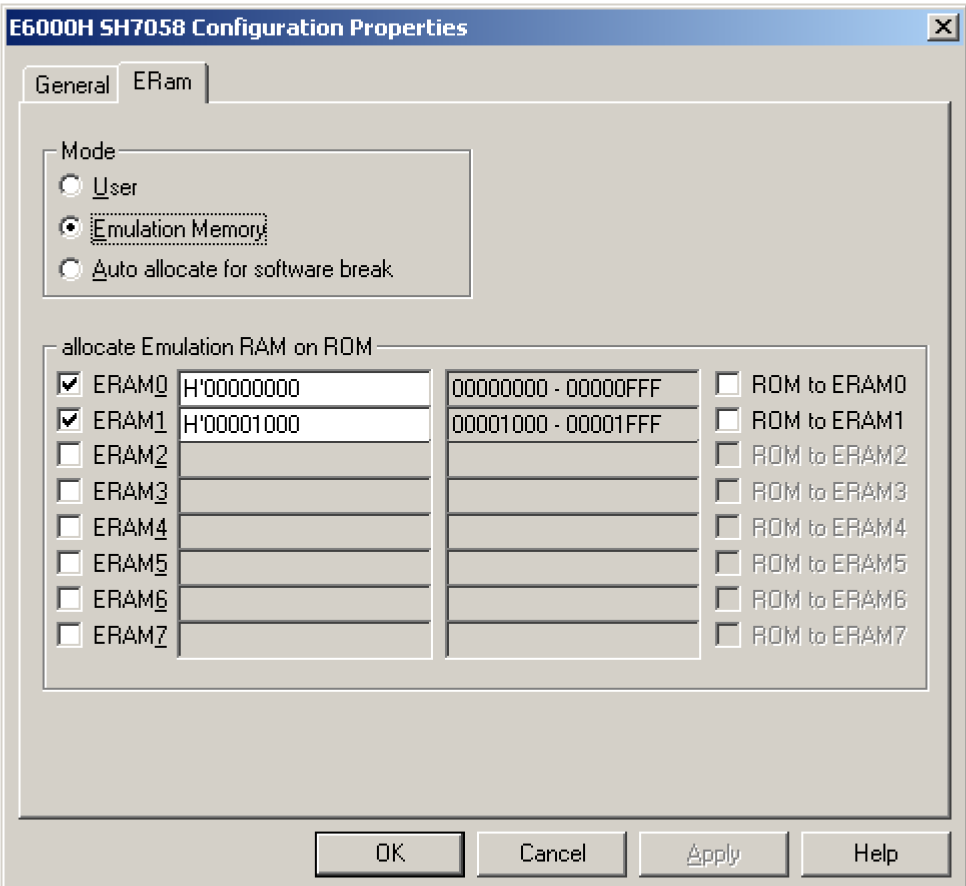



**Figure 5.6 Allocation to ERAM0 and ERAM1 Display**

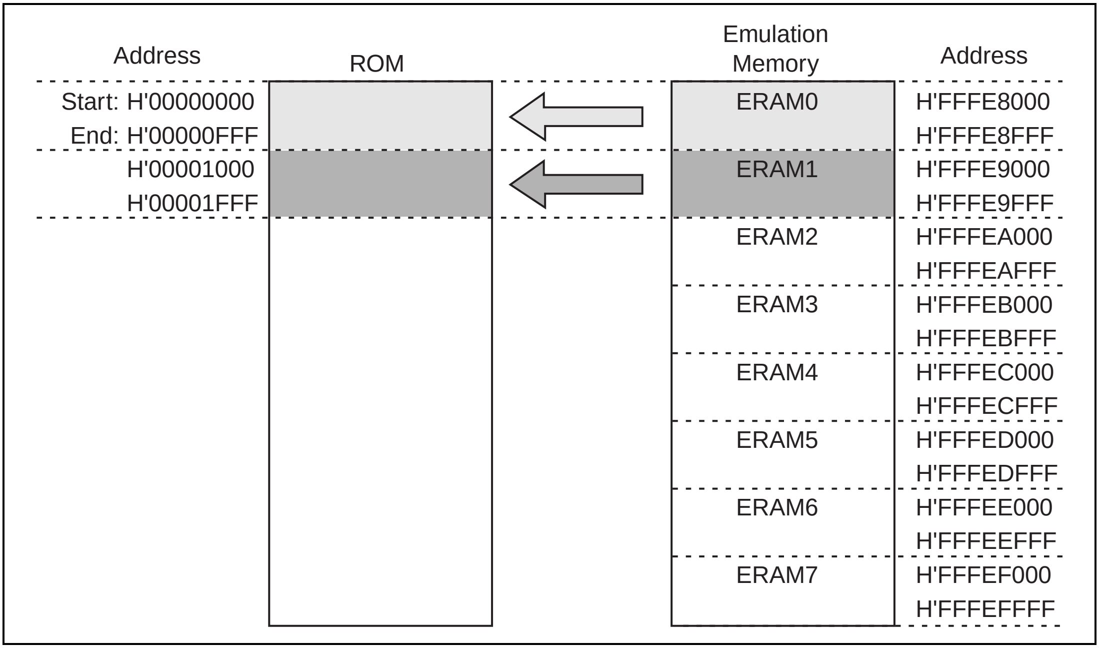



**Figure 5.7 Diagram for Allocation to ERAM0 and ERAM1**

Figure 5.8 shows the display when the ROM area from H'00000000 to H'00000FFF is allocated to ERAM4, and H'00002000 to H'00002FFF to ERAM0, respectively. Figure 5.9 shows the diagram.

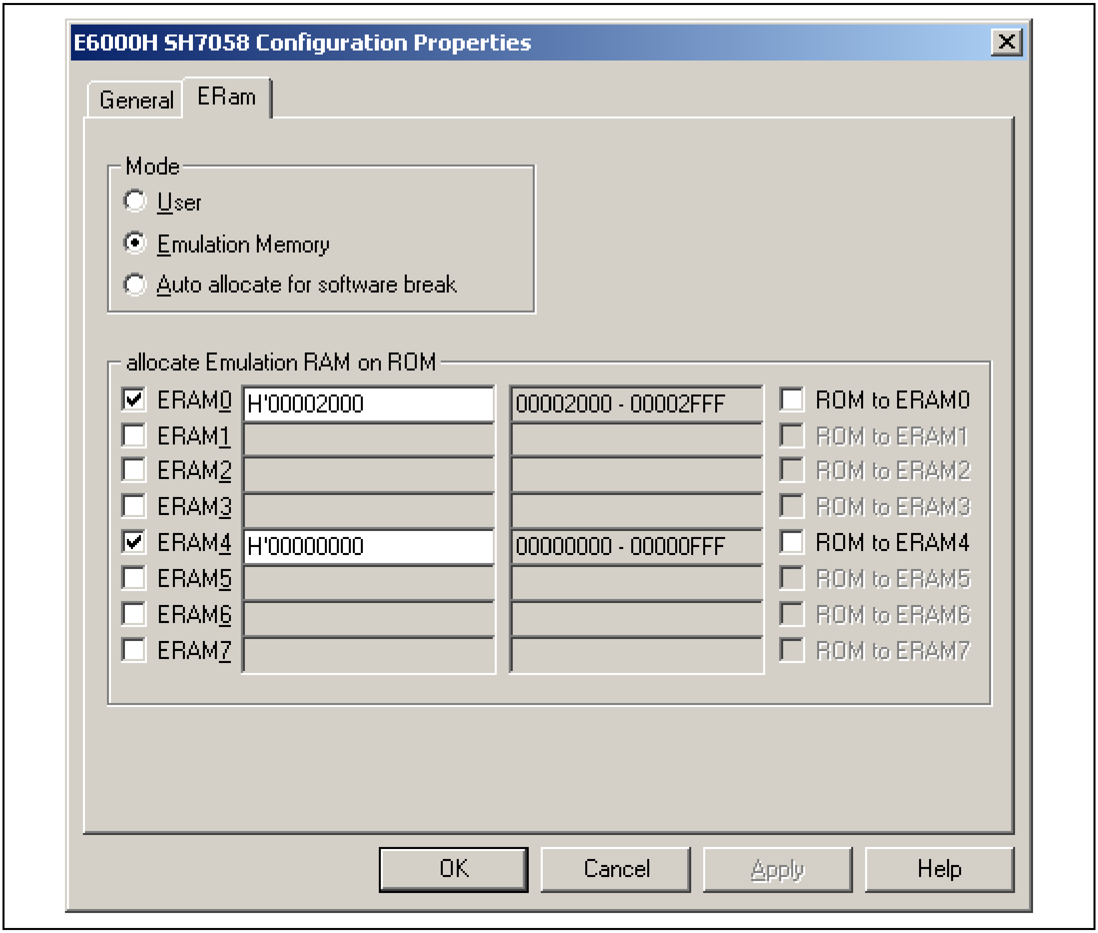



**Figure 5.8 Allocation to ERAM4 and ERAM0 Display**

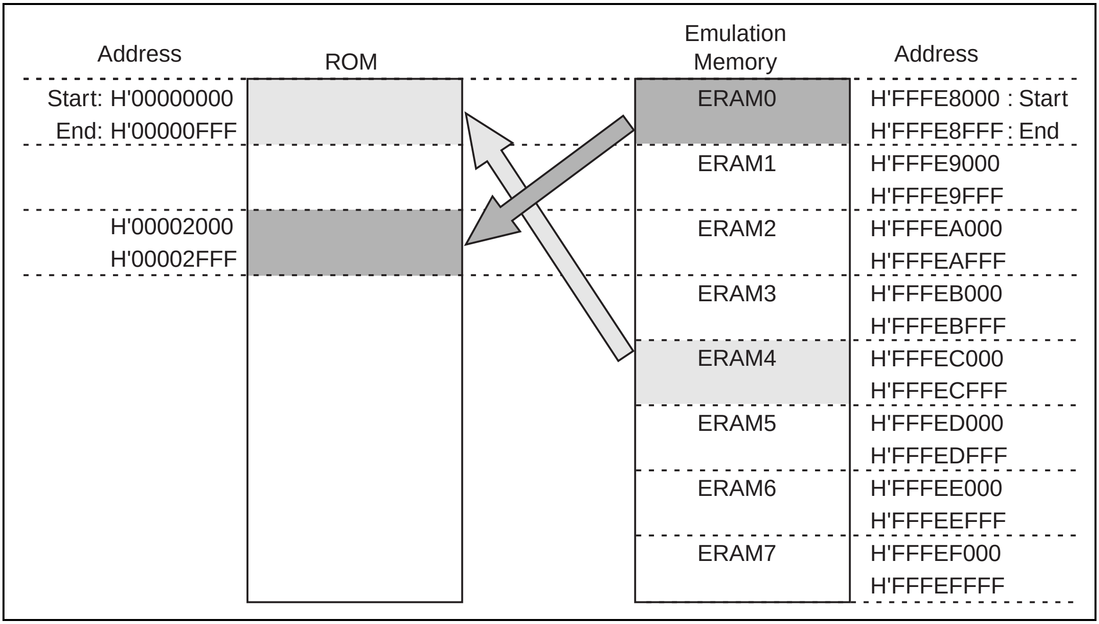



**Figure 5.9 Diagram for Allocation to ERAM4 and ERAM0**

(b) Automatically Copying ROM Contents to ERAM

Figure 5.10 shows the setting when the ROM area from addresses H’00000000 to H’00000FFF and H’00001000 to H’00001FFF are allocated to ERAM0 and ERAM1, respectively, then automatically copying the ROM area from addresses H’00000000 to H’00000FFF contents to ERAM0.

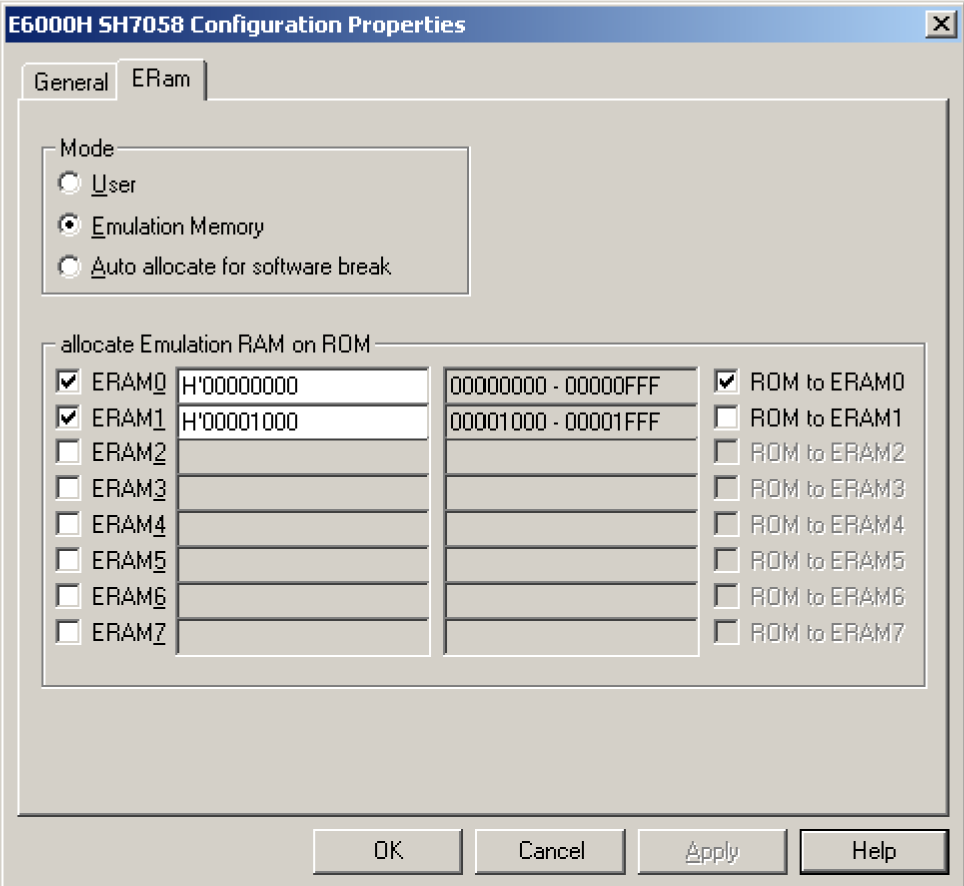



**Figure 5.10 Automatically Copying ROM Contents to ERAM0**

As shown in figure 5.10, when the [ROM to ERAM*n* (*n* = 0 to 7)] checkbox on the right side of the [allocate Emulation RAM on ROM] group box is checked and [OK] is clicked, the contents of the on-chip ROM are written to the area that has been emulated by the specified ERAM.

(c) Automatically Writing (Copying) ERAM Contents to the ROM Area (On-chip Flash Memory)

When the ERAM is released to the user, the automatic writing (copying) ERAM contents to ROM area (on-chip flash memory) is enabled. As shown in figure 5.6, when the ROM area from addresses H’00000000 to H’00000FFF and H’00001000 to H’00001FFF are allocated to ERAM0 and ERAM1, respectively, and ERAM0 and ERAM1 is released, the display is as shown in figure 5.11. Figure 5.12 shows the selection for automatically writing (copying) ERAM contents to ROM area (on-chip flash memory).

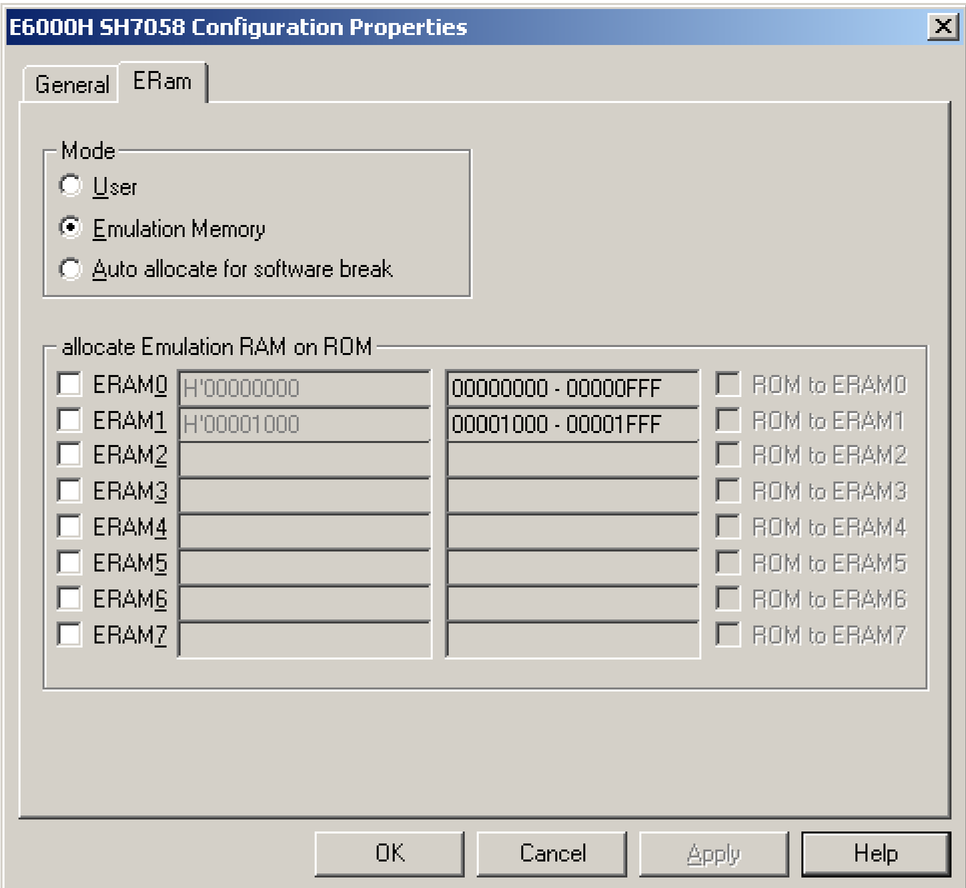



**Figure 5.11 Releasing ERAM0 and ERAM1**

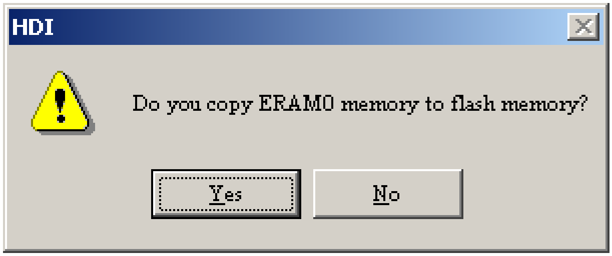



**Figure 5.12 Selection of Automatic Downloading of ERAM contents to ROM Area (On-chip Flash Memory)**

As shown in figure 5.11, remove the checking in the checkbox on the left side of the [allocate Emulation RAM on ROM] group box, and click [OK]. When the check is removed, the display shows the selection whether or not to write the ERAM contents to on-chip ROM (on-chip flash memory) for each ERAM area, so click [Yes] or [No].

**[Auto allocate for software break] Mode:**

When a software break is set for the ROM area in the [Auto allocate for software break] mode, the ERAM area is automatically used to emulate the ROM area including the address where the software break has been set. This function enables automatically setting the software break to the ROM area without setting the area to be emulated. The emulation memory is allocated in 4-kbyte boundaries. Figure 5.13 shows the setting of the [Auto allocate for software break] mode.

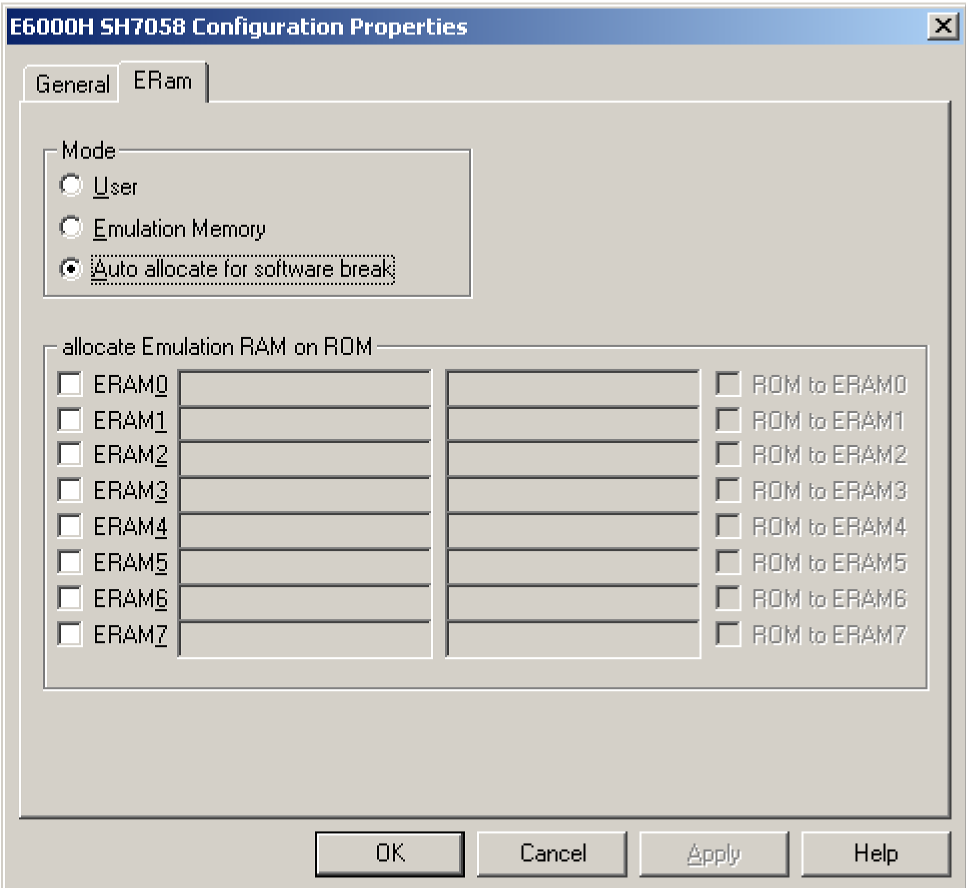



**Figure 5.13 Selection of the [Auto allocate for software break] Mode in [ERam] Page**

## 5.3 Program Execution

### 5.3.1 Execution

Table 5.6 shows the main forms of program execution.

**Table 5.6 Program Execution**

| Form | Function | Procedure |
|---|---|---|
| Normal execution | Executes the user program from the current PC (program counter) address. | Click the [Go] button |
| | | Select [Go] from the [Run] menu |
| Execution from the reset vector | Inputs the RES signal to the MCU, then executes the user program from the reset vector. | Click the [Reset Go] button |
| | | Select [Reset Go] from the [Run] menu |
| Setting a program counter to a specified address | Sets the program counter to the specified address. | Place the mouse cursor on the [Source] window. Then click the [Set PC Here] button or select [Set PC to Cursor] from the [Run] menu |
| Execution to a specified address | Executes the user program to the specified address. | Place the mouse cursor on the [Source] window. Then click the [Go to Cursor] button or select [Go to Cursor] from the [Run] menu |
| | Specifies the end address, and executes the user program up to that address. | Select [Run…] from the [Run] menu, specify a start address for [Program Counter] in the [Run] dialog box, specify an end address for [Stop At] in the [Run] dialog box, then click the [Go PC] button. More than one address can be specified as an end address. |

Note: The function, which is executed by placing the mouse cursor on the specified address in the [Source] window and then selecting the [Go to Cursor] button, or by selecting [Go to Cursor] from the [Run] menu, uses a channel for the on-chip break. Therefore, when 12 channels of the on-chip break are used with the break condition, this function cannot be used.

## 5.4 Display of Cause for Termination and Operating Status

**Cause for Termination:** When emulation is terminated, the cause of termination is displayed as the [Cause of last break] on the [Platform] sheet in the [System Status] window, and on the HDI window's status bar.

Table 5.7 is a list of the messages that indicate the various causes for termination.

**Table 5.7 Causes for Termination**

| Display | Meaning |
|---|---|
| User Break | A forced break has been issued via the [STOP] button or [Halt] from the [Run] menu. |
| Software Break | The break was triggered by a software break. |
| On Chip Channel *n* (*n* = 1 to 12) | The break was triggered by an on-chip break. |
| On Emulator Break | The break was triggered by an on-emulator break. |
| Stepping Completed | A step execution has been completed. |
| Stepping Aborted | A step execution has been aborted. |
| Performance Break | The break was triggered by performance analysis. |
| Invalid breakpoint | The break was triggered by an instruction other than PC Break. |

**Operating Status Display:** While the user program is in execution, the MCU's operating status is monitored and displayed on the HDI window's status bar. This function allows the user to observe the progress of the program. The display is only updated when the status changes.

Table 5.8 is a list of the operating status messages.

**Table 5.8 Operating Status Display**

| Display | Meaning |
|---|---|
| Address=xxxxxxxx | During execution of the user program, the address from which operations are fetched is displayed here. |
| Reset | The MCU has been reset. The RES signal is low. |
| Running | Execution of the user program has been initiated. This message is displayed once the execution has been started or restarted. Note that this message is deleted when Address=xxxxxxxx starts to be displayed. |
| Status=SLEEP | The MCU is in its sleep mode |
| Status=Bus Timeout WAIT signal = Low | The WAIT signal is low. |
| Status=BREQ | This status is displayed when the BREQ signal is low. |
| Status=BUSREL | The MCU is in its bus release mode. |
| Status=STANBY | The MCU is in its hardware or software standby mode. |
| Status=Bus Timeout RES signal = Low | The RESET pin is low. |
| Status=VCC Down | The VCC is lower than the value that has been set. |

## 5.5 Step Functions

### 5.5.1 Step Execution

Several types of step execution are available, and are shown in table 5.9.

**Table 5.9 Step Execution**

| Type | Description | Procedure |
|---|---|---|
| Executing each instruction of a function as a single step | Executes each line or instruction as one step. When a function is called, the call is executed, and execution stops at the first line or instruction of the called function. | Click the [Step] button. |
| | | Select [Step In] from the [Run] menu. |
| Executing all instructions of a function as a single step | Executes each line or instruction as one step. When a function is called, all instructions of the called function are executed as a single step, and execution stops at the line or instruction immediately after the calling line or instruction. The instructions are executed only in the RAM area on the user system and the area that the emulation memory has been allocated. | Click the [Step Over] button. |
| | | Select [Step Over] from the [Run] menu. |
| Executing a specified number of steps | Executes the specified number of steps.<br>Note that the specified address must be the start of an instruction. If, for example, the address of the second byte of an instruction is specified, execution will not stop, and the specified number of steps will still be executed. | Click the [Step dialog] button, specify the number of steps in [Steps] in the [Step Program] dialog box, and start execution. Selecting [Step Over Calls] allows a function call to be executed as a single step. |
| | | Select [Step…] from the [Run] menu. The settings are the same as above. |
| Stopping function execution | Steps out of a function. Execution stops at the line after the calling line in the program. | Click the [Step Out] button. |
| | | Select [Step Out] from the [Run] menu. |

Note: Breaks become disabled during step execution. However, the trigger is output.

### 5.5.2 Accepting Interrupts during Step Execution

Interrupts cannot normally be accepted during step execution. Select [Enable interrupts during step execution] from the [General] page of the [Configuration] dialog box if you want interrupts to be accepted during step execution.

## 5.6 Break Functions

The emulator provides break function shown in table 5.10. The HDI displays a list of breakpoints in the [Breakpoints] window, and the break conditions are specified in the dialog boxes for break functions.

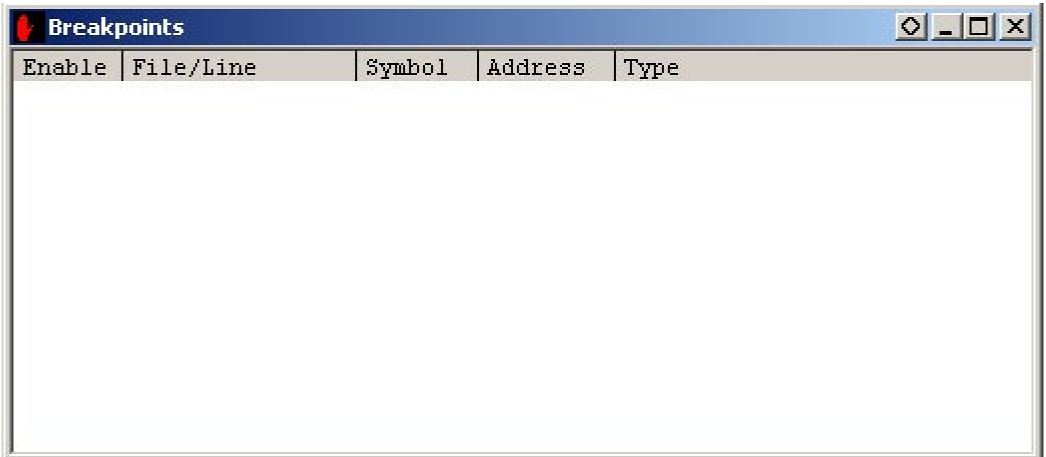



**Figure 5.14 [Breakpoints] Window**

For details on the [Breakpoints] window, refer to the Hitachi Debugging Interface User's Manual (on the CD-R).

**Table 5.10 Break Functions**

| Type | Description |
|---|---|
| On-chip break | A break function incorporated in the MCU. When a condition set for On Chip Break Channel *n* (*n* = 1 to 12) has been satisfied, a break occurs. "OnChip" is displayed under Type in the [Breakpoints] window. |
| On-chip sequential break | When all of the conditions have been satisfied in the specified order of the on-chip break conditions, a break occurs. On Chip Break Channel 1 to 8 can be used for four- or eight-level sequential break conditions. On Chip Break Channel Reset is used as a reset point for the sequential break condition (can be set only for On Chip Break Channel 8). |
| On-emulator break | This type of break is generated by the dedicated hardware in the emulator. Conditions can be set to On Emulator Break, and when one of these conditions has been satisfied, a break occurs. "OnEmulator" is displayed under Type in the [Breakpoints] window. |
| Software break | The contents of the specified address are replaced by a break instruction (a dedicated instruction for use with the emulator), and the program is then executed. When the break instruction is executed, a break occurs. "Software" is displayed under Type in the [Breakpoints] window. |
| Forced break | This is the break for the forcible termination of a program that is issued when the [STOP] button on the toolbar is pressed. |
| Break due to trace buffer overflow | This break occurs when the trace buffer in the emulator overflows during trace acquisition. |

Notes: 1. The [Enable] column in the [Breakpoints] window displays ● only when On Chip Break is selected and the Bus/Area condition is Instruction prefetch. When BREAKPOINT is set in other conditions, ❍ is displayed.

2. During user program execution, it is not possible to jump the corresponding source line (or address line) on the [Source] or [Disassemble] window from the breakpoint by using [Go to Source] in the pop-up menu on the [Breakpoints] window.

### 5.6.1 On-Chip Break

**Overview:** These break functions built in the MCU. The on-chip break conditions are shown in table 5.11. These conditions are satisfied when all of its specified conditions are satisfied (an AND condition).

**Table 5.11 On-Chip Break Conditions**

| Break Condition | Description |
|---|---|
| Address bus | Satisfied when the address value matches the specified value. |
| Data bus | Satisfied when the data value matches the specified value. |
| Bus/Area | Satisfied when the specified access type, bus status, and read/write cycle conditions are matched.<br>When no condition is set, all bus cycles, including program-fetch cycles, satisfy the condition. |
| Satisfaction count | This condition is specified in combination with some other conditions. The break occurs when the specified condition has been satisfied the specified number of times. |
| Reset point | Specifies the reset point condition. This condition is available only for On Chip Break Channel 8. |
| Operating | Specifies the operation when the condition has been satisfied. |

The on-chip break uses 12 channels independently and is used as the PC breakpoint that is set by double-clicking the [Source], [Disassembly], or [Label] window. When there is an empty channel, the PC breakpoint that is set by double-clicking is set for that channel. When there is no empty channel, the empty channel for the software break is used as the PC breakpoint. However, in this case, the PC breakpoint cannot be set in the on-chip ROM area (on-chip flash memory area).

When the software breakpoint is set in the on-chip ROM area, emulate the ROM area by using the ERAM.

**Setting an On-Chip Break:** The setting of On Chip Break 8 is taken as an example.

Select [Edit...] from the pop-up menu in the [Breakpoints] window, and the dialog box (figure 5.15) will appear. Click the [On Chip Break] tab to select the [On Chip Break] page.

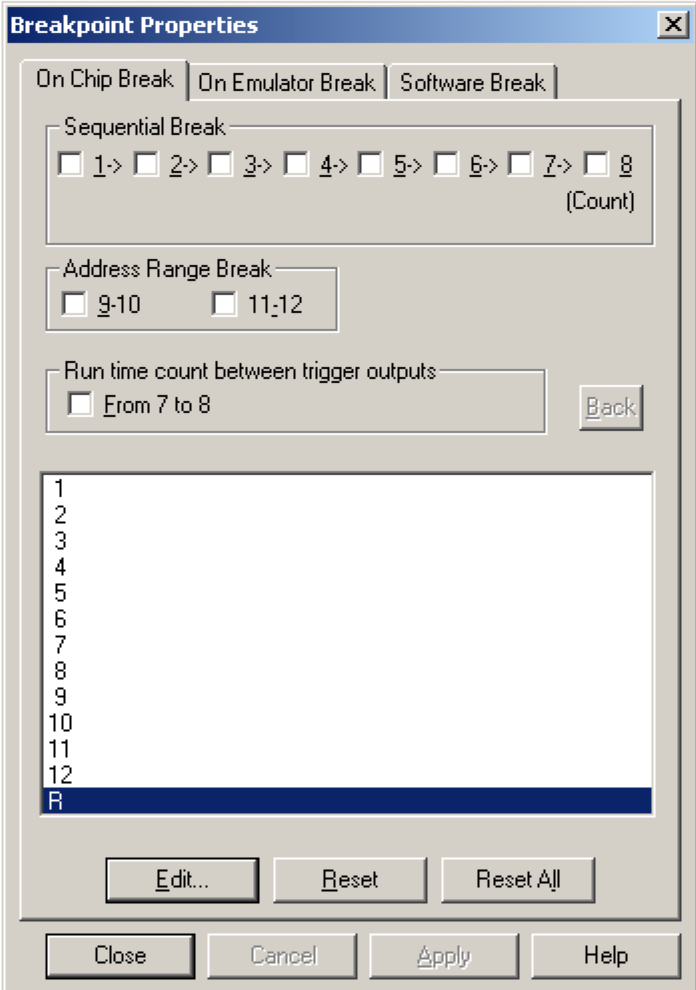



**Figure 5.15 [Breakpoint Properties] Dialog Box ([On Chip Break] Page)**

**Table 5.12 [On Chip Break] Dialog Box Options**

| Option | Description |
|---|---|
| [Sequential Break] | Specifies On Chip Break 1 to 8 and the order of halting a breakpoint. Channels are specified in a sequential order. |
| [Address Range Break] | Specifies the combination of channels and the range of halting a break. For channels, select the following:<br>9-10: Channels 9 and 10 are used to specify the range of break.<br>11-12: Channels 11 and 12 are used to specify the range of break. |
| [Point To Point Time Measurement] | Measures a time between two points by using channels 7 and 8.<br>After channel 7 has been satisfied, the time is measured when channel 8 is satisfied.<br>The measured result is displayed in PtoP Time Count in the [System Status] window, which is EORed to RunTime Count. |
| [Back] | Puts the setting back when the dialog box has been displayed. |
| [Edit...] | Modifies the setting selected in the [Condition] list box. Clicking this button opens the [On Chip Break *n*] dialog box. (*n*: channel number.) |
| [Reset] | Clears the settings on the On Chip Break page selected in the [Condition] list box. |
| [Reset All] | Clears all settings on the On Chip Break page in the [Condition] list box. |

Double-click 8 in the list box by using a mouse. Click the [Edit...] button to open the [On Chip Break 8] dialog box.

Close the [On Chip Break 8] dialog box and the display returns to the [On Chip Break] dialog box. The contents of the on-chip break condition set to 8 in the list box are displayed. Click the [OK] button to close the [On Chip Break] dialog box.

On-chip break conditions are specified for the other channels in the same way.

The following sections describe each page.

(a) [Address] Page

Use this page to specify the address bus conditions.

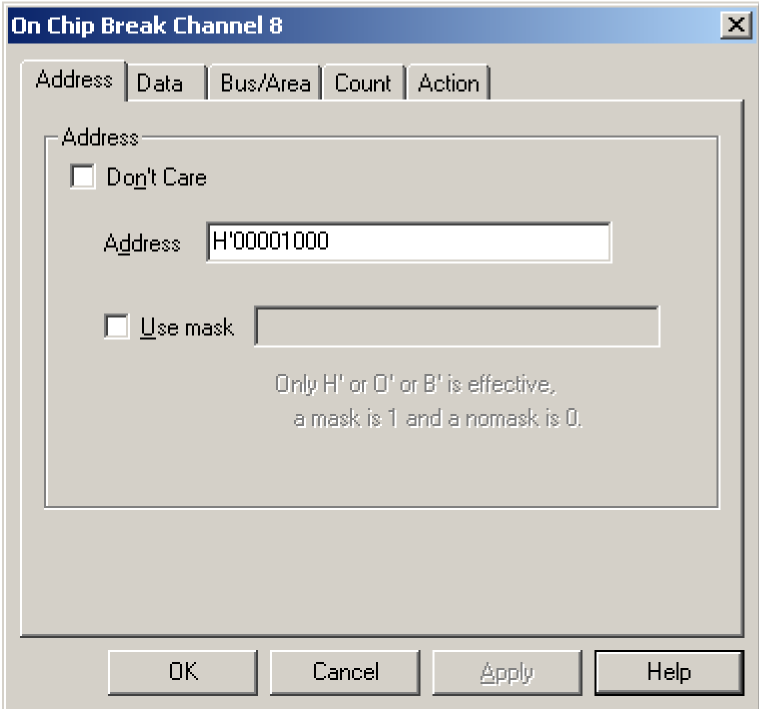



**Figure 5.16 [On Chip Break Channel 8] Dialog Box ([Address] Page)**

**Table 5.13 [Address] Page Options**

| Option | Description |
|---|---|
| [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. |
| [User mask] | Sets mask conditions.<br>Sets the mask bits if [User mask] is selected. Masked bits satisfy this break condition regardless of their values. |

Notes: 1. Set a multiple of two as the address value when the 16-bit area is accessed.
2. Set a multiple of four as the address value when the 32-bit area is accessed.

(b) [Data] Page

Use this page to specify the data bus conditions.

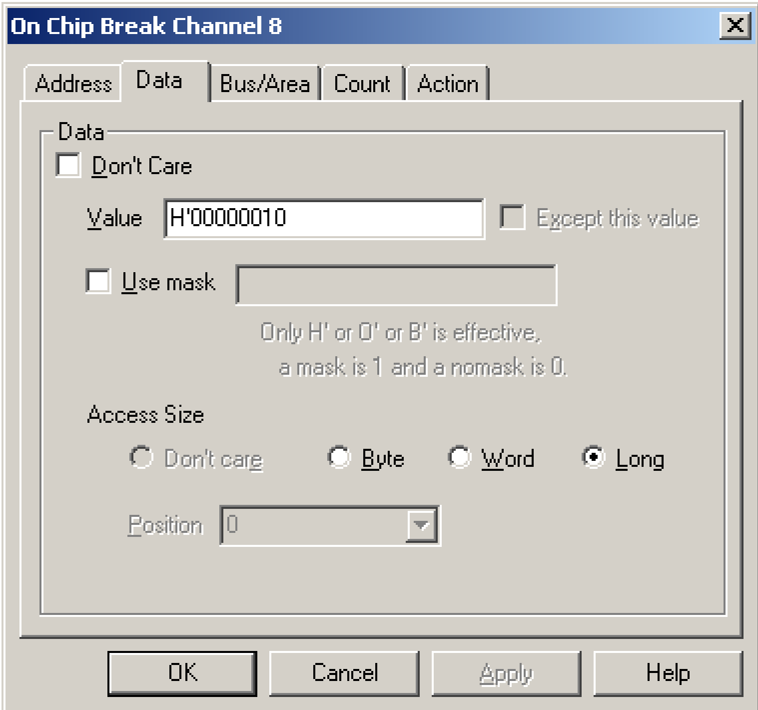



**Figure 5.17 [On Chip Break Channel 8] Dialog Box ([Data] Page)**

**Table 5.14 [Data] Page Options**

| Option | Description |
|---|---|
| [Don't Care] | Sets no data bus condition. |
| [Value] | Sets a data bus value as a number. |
| [Access Size] | Specifies the data-access size.<br>[Byte]: Sets byte-data-access cycles.<br>[Word]: Sets word-data-access cycles.<br>[Long]: Sets longword-data-access cycles.<br>[Position]: Sets a data bus value as a number. The position of the valid data bus is specified.<br>[Long]: None<br>[Word]: 4n: Upper word<br>4n + 2: Lower word<br>[Byte]: 4n: Upper byte of the upper word<br>4n + 1: Lower byte of the upper word<br>4n + 2: Upper byte of the lower word<br>4n + 3: Lower byte of the lower word |
| [Use Mask] | Sets mask conditions. Sets the mask bits if [User mask] is selected. Masked bits on the data bus satisfy this break condition regardless of their values. |

(c) [Bus State] Page

Use this page to specify conditions for the read or write cycle and access type.

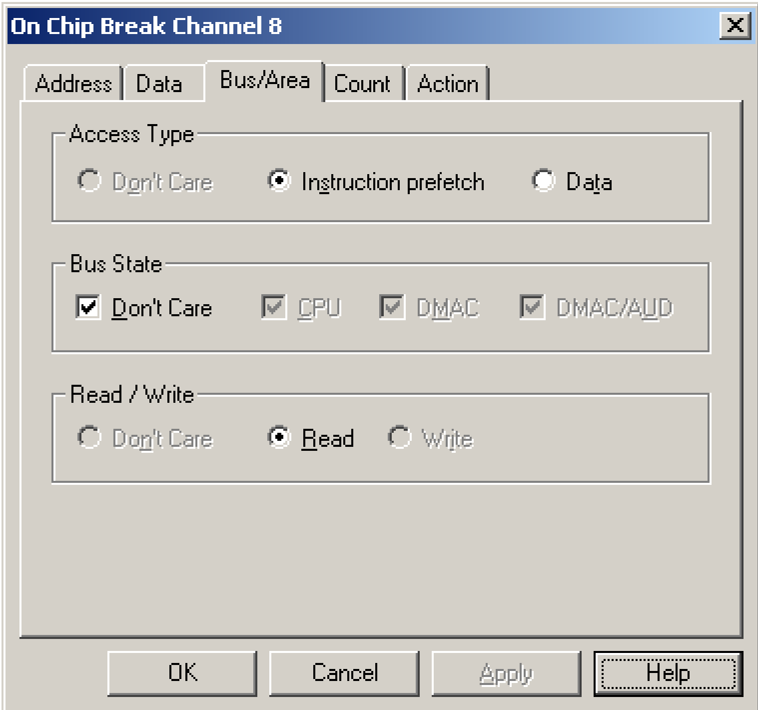



**Figure 5.18 [On Chip Break Channel 8] Dialog Box ([Bus State] Page)**

**Table 5.15 [Bus State] Page Options**

**[Access Type] Group Box**

| Option | Description |
|---|---|
| [Instruction prefetch] | Instruction prefetch cycles satisfy this condition. |
| [Data] | Data access cycles satisfy this condition. |

### [Bus State] Group Box

| Option | Description |
|---|---|
| [Don’t Care] | All cycles satisfy this condition. |
| [CPU] | CPU cycles satisfy this condition. |
| [DMAC] | DMAC cycles satisfy this condition. |

### [Read/Write] Group Box

| Option | Description |
|---|---|
| [Don’t Care] | Both of read and write cycles satisfy this condition. |
| [Read] | Read cycles satisfy this condition. |
| [Write] | Write cycles satisfy this condition. |

(d) [Count] Page

Use this page to specify a satisfaction count condition.

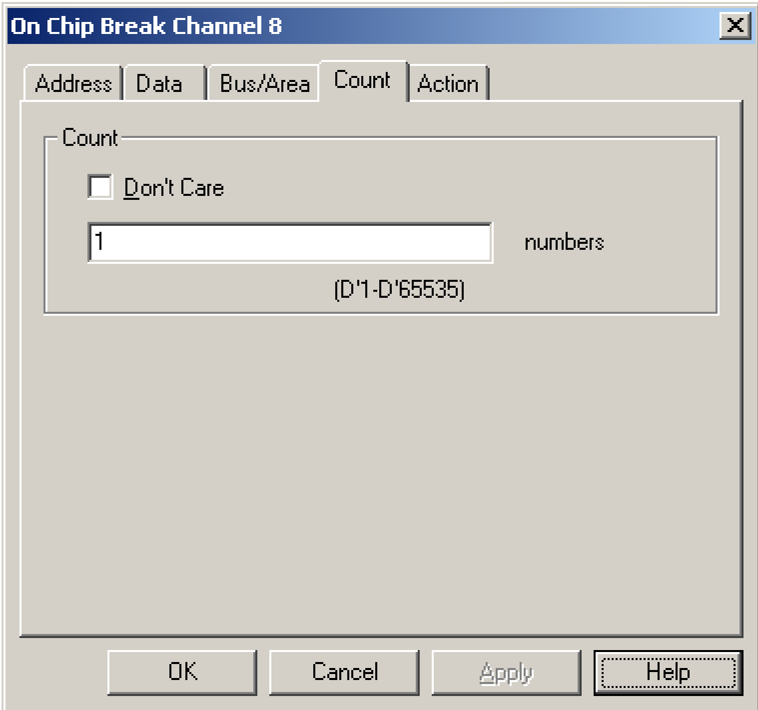



**Figure 5.19 [On Chip Break Channel 8] Dialog Box ([Count] Page)**

**Table 5.16 [Count] Page Options**

| Option | Description |
|---|---|
| [Don't Care] | Selects no satisfaction count condition. |
| Input area | Sets a value for the satisfaction count condition as a numeric. The default is D'1. Any value in the range from D'1 to D'65535 can be set here. |

Note: The satisfaction count condition can be set only for channel 8.

(e) [Action] Page

Use this page to specify an operating condition when the setting condition has been satisfied.

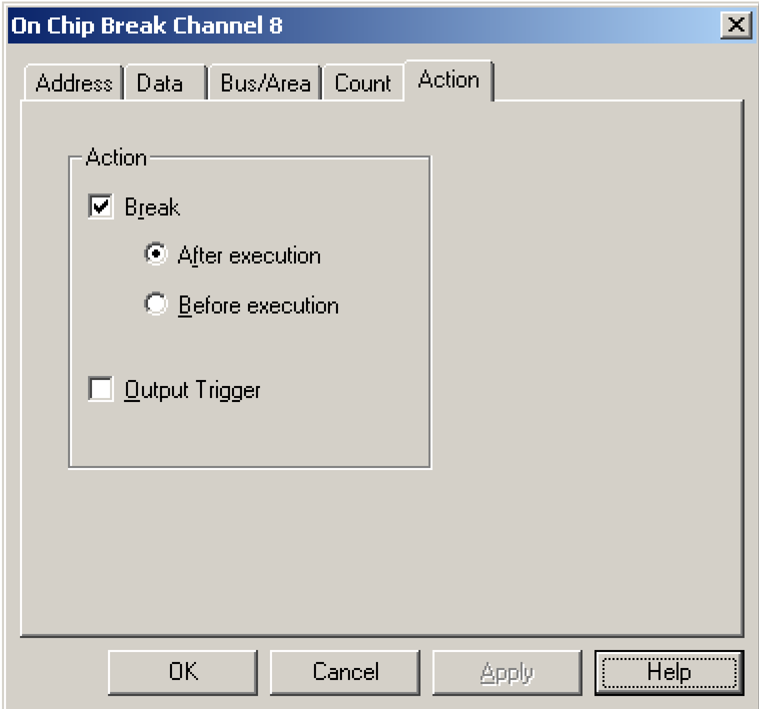



**Figure 5.20 [On Chip Break Channel 8] Dialog Box ([Action] Page)**

**Table 5.17 [Action] Page Options**

| Option | Description |
|---|---|
| [Break] | Halts execution when the setting condition has been satisfied.<br>[After execution]: Halts execution after the address at which the condition has been satisfied was executed.<br>[Before execution]: Halts execution before the address at which the condition is satisfied is executed. |
| [Output Trigger] | Outputs a trigger when the setting condition has been satisfied. |

Note: The trigger output when the setting condition has been satisfied can be used only in channels 1 to 8.

(f) [Reset] Page

Use this page to specify the reset point conditions.

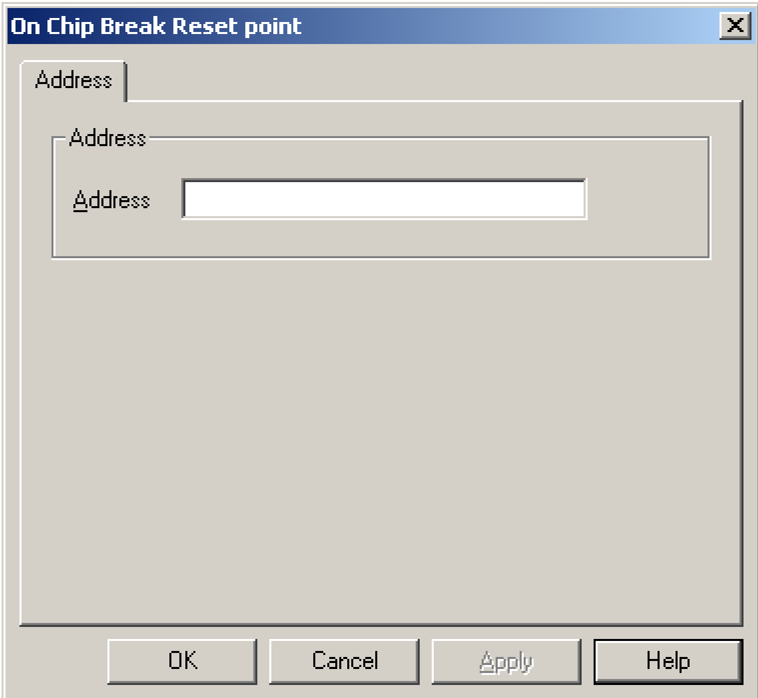



**Figure 5.21 [On Chip Break Reset] Dialog Box ([Reset] Page)**

**Table 5.18 [On Chip Break Reset] Dialog Box Option**

| Option | Description |
|---|---|
| [Address] | Specifies the reset point with the address conditions. |

### 5.6.2 On-Chip Sequential Break

**Overview:** An on-chip sequential break set up with on-chip break occurs when on-chip break conditions are satisfied in one of three specified orders.

As shown in table 5.19, there are two modes for the on-chip sequential break that uses On Chip Break Channel 1 to 8 and Reset. When a reset point condition is satisfied before satisfaction of On Chip Break Channel 8, all the satisfied conditions are cleared. The emulator then restarts checking for satisfaction of the on-chip sequential break conditions from the first condition.

**Table 5.19 On-Chip Sequential Break Modes (On Chip Break Channel 1 to 8)**

| Mode | Description |
|---|---|
| 8-level on-chip sequential break | On Chip Break Channel 1 to 8 are used for sequential break. In this mode, up to eight sequential break conditions can be set.<br>The normal On Chip Break condition can be independently set for On Chip Break Channel 9 to 12. |
| 4-level on-chip sequential break | On Chip Break Channel 1 to 4 are used for sequential break. In this mode, up to four sequential break conditions can be set.<br>The normal On Chip Break condition can be independently set for On Chip Break Channel 1 to 4 and 9 to 12. |

Note: When the same sequential break conditions are continued, only the first condition is considered as satisfied. In this case, other conditions are not satisfied and no sequential break occurs.

**Setting an On-Chip Sequential Break:** Set the on-chip break conditions. To specify the on-chip sequential break mode, which uses On Chip Break Channel 1 to 8, select [Sequential Break] on the [On Chip Break] page of the [Breakpoint Properties] dialog box. In the example shown in figure 5.22, 8-level on-chip sequential break mode is selected.

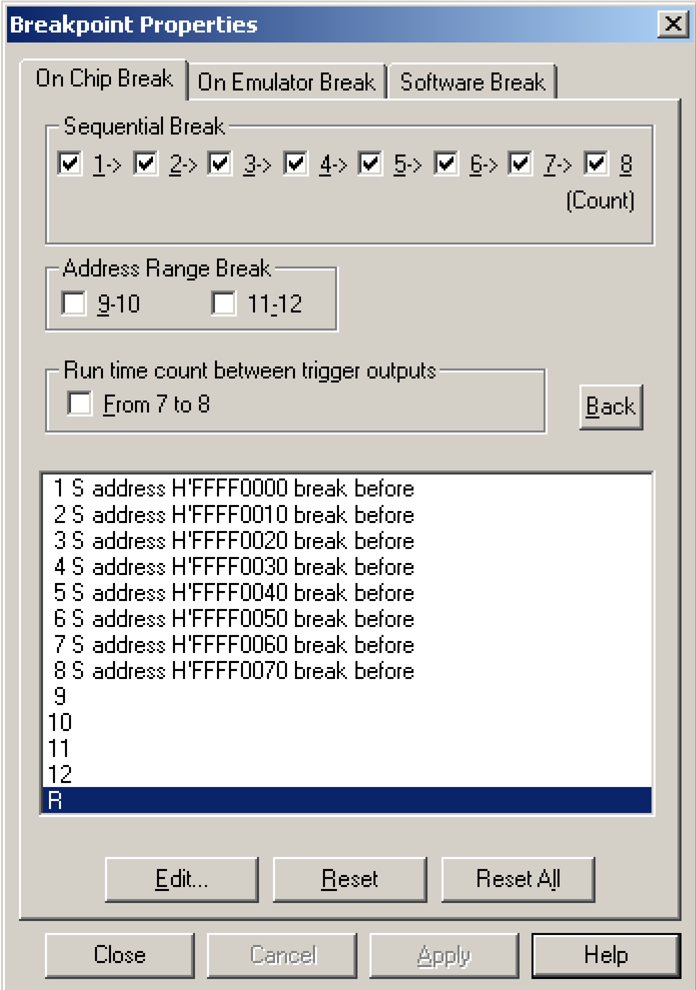



**Figure 5.22 [Breakpoint Properties] Dialog Box ([On Chip Break] Page)**

Table 5.20 shows the items selected in [Sequential Break] when the on-chip sequential break is specified using On Chip Break Channel 1 to 8.

**Table 5.20 [Sequential Break] Options (On Chip Break Channel 1 to 8)**

| [Sequential Break] Item | Description |
|---|---|
| 8-level on-chip sequential break mode | When either of 1, 2, 3, or 4 is selected, Channels 1 to 8 are always used as a sequential break. Set the different conditions for all of Channels 1 to 8. |
| 4-level on-chip sequential break mode | When either of 5, 6, 7, or 8 is selected, Channels 5 to 8 are always used as a sequential break. Set the different conditions for all of Channels 5 to 8. |

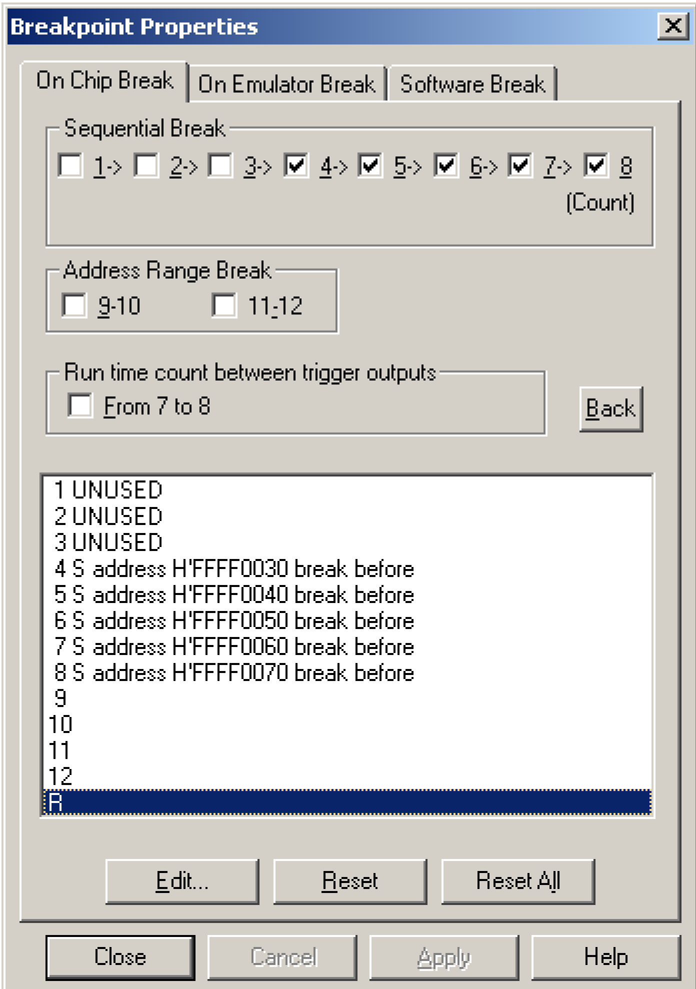



**Figure 5.23 Selecting 4 for [Sequential Break]**

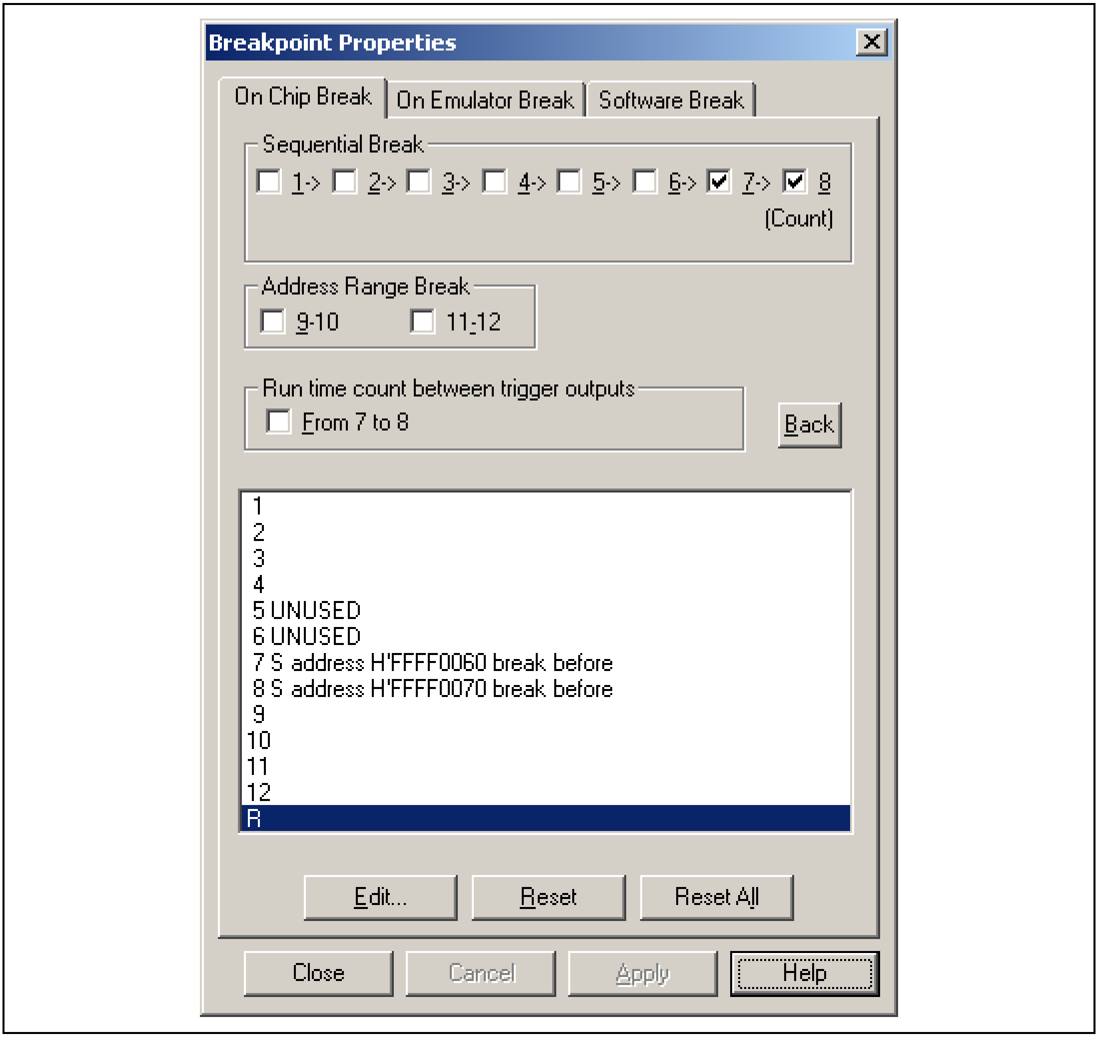



**Figure 5.24 Selecting 7 for [Sequential Break]**

### 5.6.3 On-Emulator Break

**Overview:** On-emulator break functions are implemented by dedicated hardware in the E6000H station. The on-emulator break conditions have four channels (On Emulator Break 1 to 4). On-emulator break occurs when all of the specified conditions (an AND condition) are satisfied.

**Table 5.21 On-Emulator Break Conditions**

| Condition | Description |
|---|---|
| Address bus | Satisfied when the value on the address bus matches the specified value. |
| Data bus | Satisfied when the value on the data bus matches the specified value. |
| Bus/Area | Satisfied when the specified access type, bus status, and read/write cycle conditions are matched.<br>When no condition is set, all bus cycles, including program-fetch cycles, satisfy the condition. |
| External probe | Satisfied when the external probe (PRB) signal levels match a specification. |
| External interrupt signal | Satisfied when the external interrupt signal levels match a specification. |
| Satisfaction count | The break occurs when the above conditions are satisfied the specified number of times. |

**Setting an On-Emulator Break:** The setting of On Emulator Break 1 is taken as an example.

Place the cursor in the [Breakpoints] window then click the right-hand mouse button to display the window's pop-up menu. Select [Edit...] from the menu, and the dialog box will appear (figure 5.15). Click the [On Emulator Break] tab to select the [On Emulator Break] page (figure 5.25).

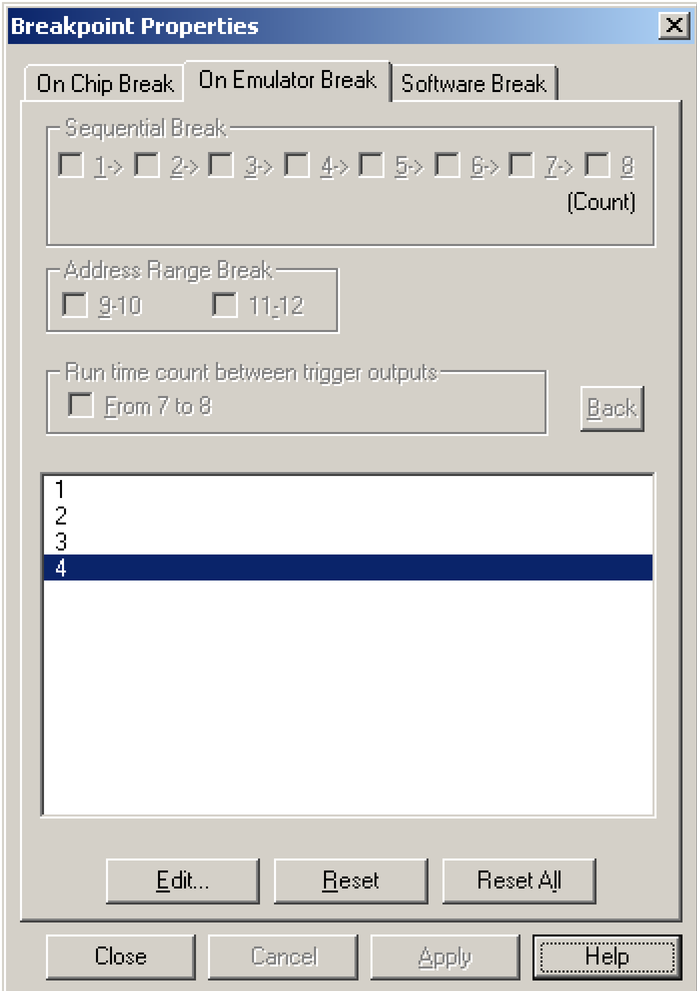



**Figure 5.25 [Breakpoint Properties] Dialog Box ([On Emulator Break] Page)**

**Table 5.22 Options in the [On Emulator Break] Page**

| Option | Description |
|---|---|
| [Edit...] | Modifies the settings on On Emulator Break 1 to 4 selected in the list box. Clicking this button opens the [On Emulator Break Channel *n*] dialog box. (*n*: 1 to 4) |
| [Reset] | Clears the settings on On Emulator Break 1 to 4 selected in the list box |
| [Reset All] | Clears all settings on On Emulator Break 1 to 4 in the list box. |

The following sections describe each page.

(a) [Address] Page

Use this page to specify the address bus conditions.

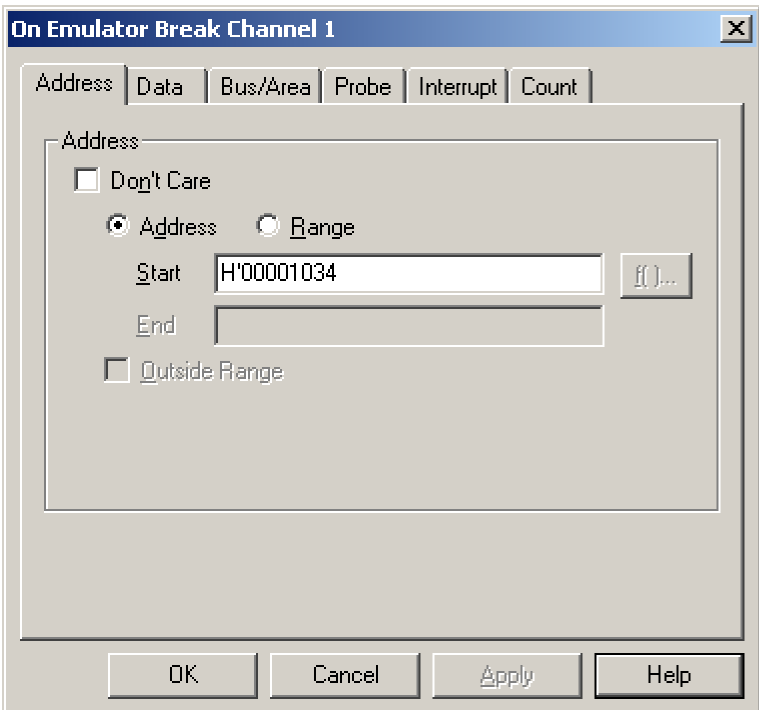



**Figure 5.26 [On Emulator Break Channel 1] Dialog Box ([Address] Page)**

**Table 5.23 [Address] Page Options**

| Option | Description |
|---|---|
| [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] to [End].<br>[Start]: Sets the (start) address bus value as a numeric or a symbol.<br>[End]: Sets the (end) address bus value as a numeric or a symbol. |
| [Outside Range] | Selects this option to generate a break with the value set in [Start] or an address bus outside the range set from [Start] to [End]. |

Notes: 1. Set a multiple of two as the address value when the 16-bit area is accessed.

2. Set a multiple of four as the address value when the 32-bit area is accessed.

(b) [Data] Page

Use this page to specify the data bus conditions. The data bus condition must be set considering the address bus, data bus width, and access size.

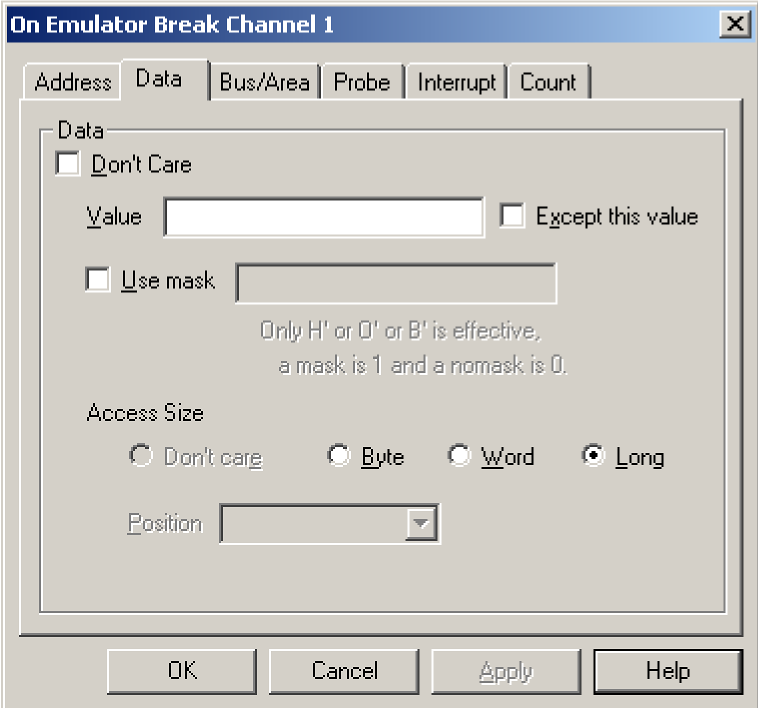



**Figure 5.27 [On Emulator Break Channel 1] Dialog Box ([Data] Page)**

**Table 5.24 [Data] Page Options**

| Option | Description |
|---|---|
| [Don’t Care] | Sets no data bus condition. |
| [Value] | Sets a data bus value as a number. |
| [Access Size] | Specifies the data-access size.<br>[Byte]: Sets byte-data-access cycles.<br>[Word]: Sets word-data-access cycles.<br>[Long]: Sets longword-data-access cycles.<br>[Position]: Sets a data bus value as a number. The position of the valid data bus is specified.<br>[Long]: None<br>[Word]: 4n: Upper word<br>4n + 2: Lower word<br>[Byte]: 4n: Upper byte of the upper word<br>4n + 1: Lower byte of the upper word<br>4n + 2: Upper byte of the lower word<br>4n + 3: Lower byte of the lower word |
| [Except this value] | Sets a value other than that has been specified as the data bus condition. |
| [Use mask] | Sets mask conditions. Sets the mask bits if [User mask] is selected. Masked bits on the data bus satisfy this break condition regardless of their values. |

(c) [Bus/Area] Page

Use this page to specify conditions for the read or write cycle and access type.

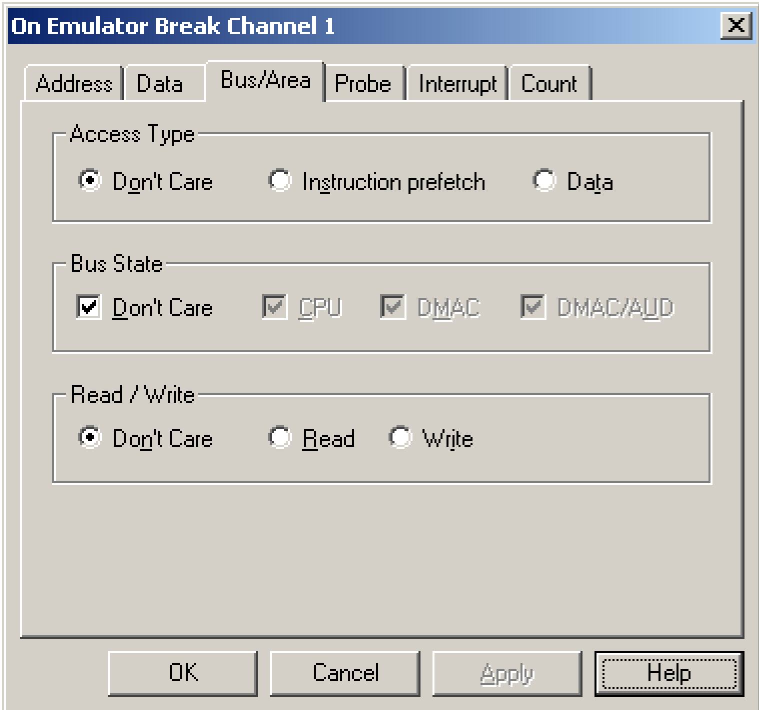



**Figure 5.28 [On Emulator Break Channel 1] Dialog Box ([Bus/Area] Page)**

**Table 5.25 [Bus/Area] Page Options**

**[Access Type] Group Box**

| Option | Description |
|---|---|
| [Don’t Care] | All access types satisfy this condition. |
| [Instruction prefetch] | Instruction prefetch cycles satisfy this condition. |
| [Data] | Data access cycles satisfy this condition. |

### [Bus State] Group Box

| Option | Description |
|---|---|
| [Don’t Care] | All cycles satisfy this condition. |
| [CPU] | CPU cycles satisfy this condition. |
| [DMAC/AUD] | DMAC and AUD cycles satisfy this condition. |

### [Read/Write] Group Box

| Option | Description |
|---|---|
| [Don’t Care] | Both of read and write cycles satisfy this condition. |
| [Read] | Read cycles satisfy this condition. |
| [Write] | Write cycles satisfy this condition. |

(d) [Probe] Page

Use this page to specify external probe signal (PRB1 to PRB4) conditions.

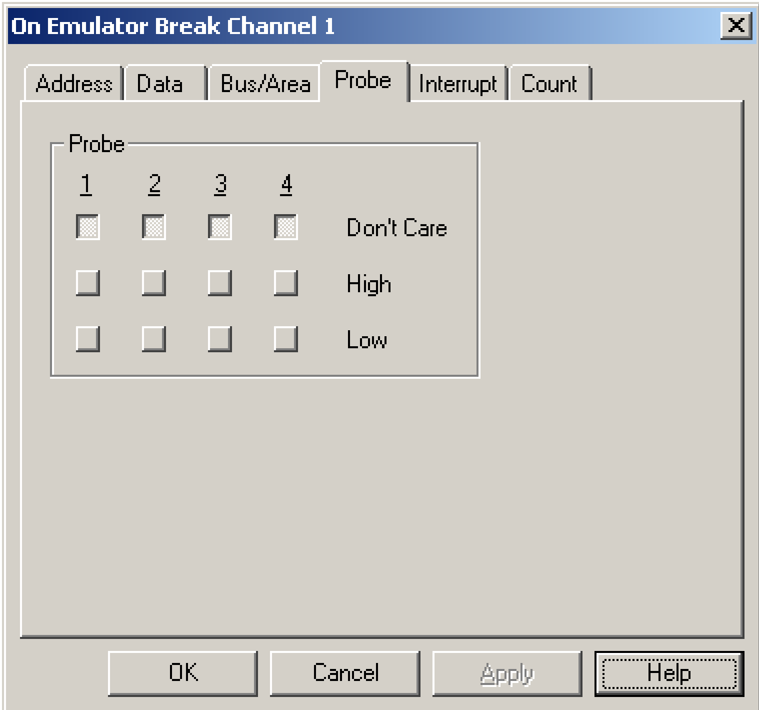



**Figure 5.29 [On Emulator Break Channel 1] Dialog Box ([Probe] Page)**

**Table 5.26 [Probe] Page Options**

| Option | Description |
|---|---|
| [Don't Care] | Selects no external probe signal state condition. |
| [High] | Sets the high level of this external probe signal as a break condition. |
| [Low] | Sets the low level of this external probe signal as a break condition. |

(e) [Interrupt] Page

Use this page to specify external interrupt (IRQ0 to IRQ7) and NMI signal conditions.

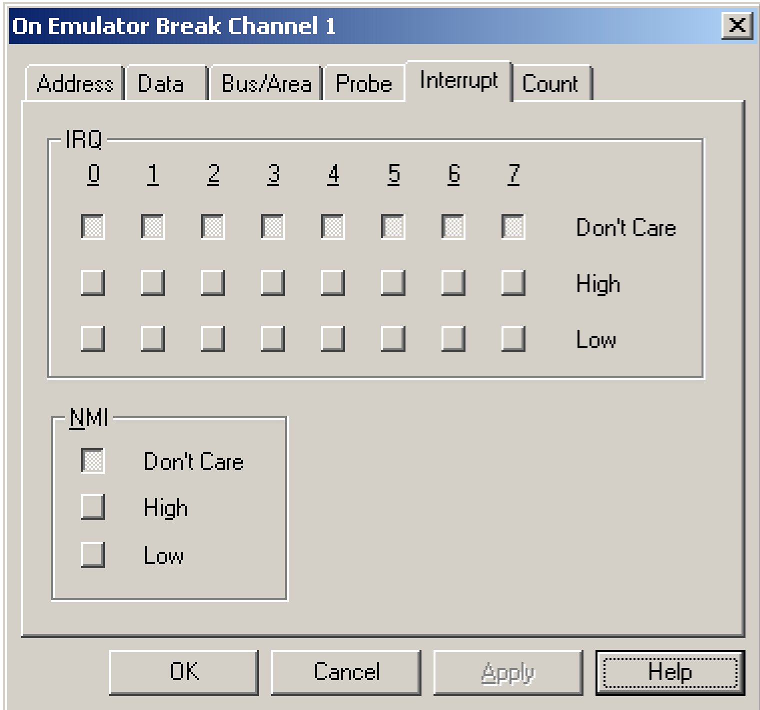



**Figure 5.30 [On Emulator Break Channel 1] Dialog Box ([Interrupt] Page)**

**Table 5.27 [Interrupt] Page Options**

**[IRQ0 to IRQ7] Group Box**

| Option | Description |
|---|---|
| [Don't Care] | Selects no external interrupt signal state condition. |
| [High] | Sets the high level of this external interrupt signal as a break condition. |
| [Low] | Sets the low level of this external interrupt signal as a break condition. |

### [NMI] Group Box

| Option | Description |
| --- | --- |
| [Don't Care] | Selects no NMI signal-state condition. |
| [High] | Sets the high level of the NMI signal as a break condition. |
| [Low] | Sets the low level of the NMI signal as a break condition. |

(f) [Count] Page

Use this page to specify a satisfaction count condition.

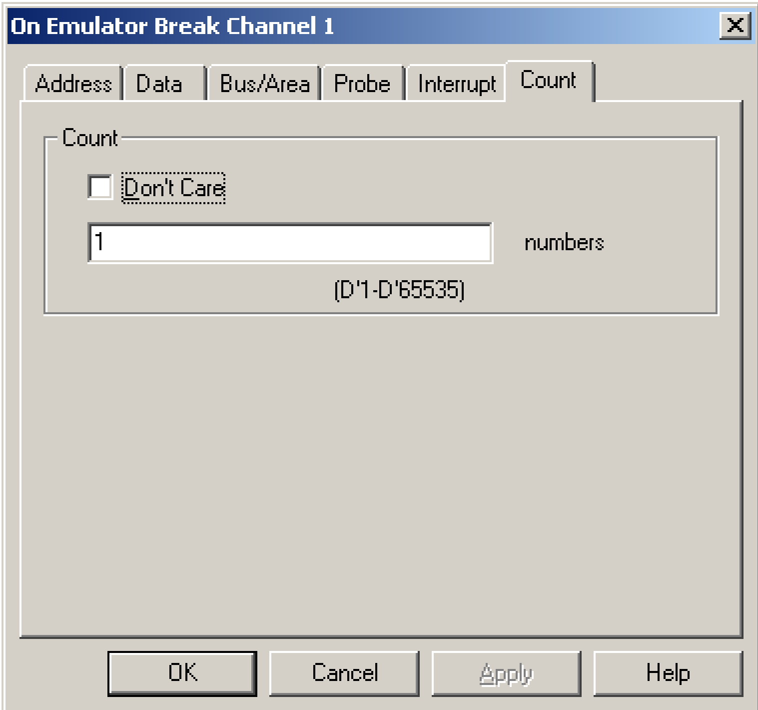



**Figure 5.31 [On Emulator Break Channel 1] Dialog Box ([Count] Page)**

**Table 5.28 [Count] Page Options**

| Option | Description |
|---|---|
| [Don't Care] | Selects no satisfaction count condition. |
| Input area | Sets a value for the satisfaction count condition as a numeric. The default is D'1. Any value in the range from D'1 to D'65535 can be set here. |

### 5.6.4 Software Break

**Overview:** Any content at the specified address is replaced by a break instruction (a dedicated instruction for use with the emulator). Execution of the user program stops when the break instruction is executed. The instruction that had been at the address is not executed, so the result is a break before execution.
It is possible to specify up to 255 software breakpoints.

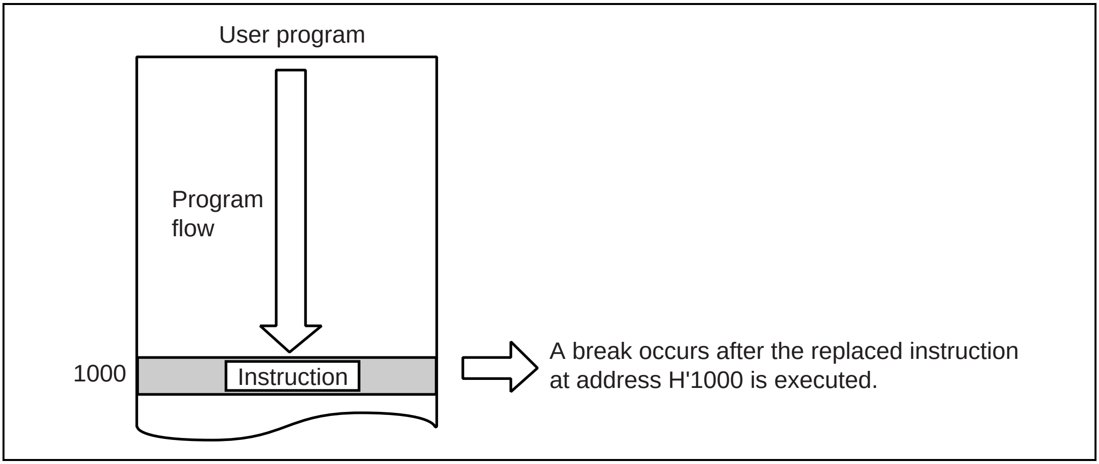



**Figure 5.32 Example of a Software Break Instruction**

The software break cannot be set in the on-chip ROM area (on-chip flash memory area). To set the software break in the on-chip ROM area, use the ERAM and emulate the ROM area.

**Setting a Software Break:** Place the cursor in the [Breakpoints] window and click the right-hand mouse button to display the pop-up menu. Select [Add...] from the menu, and the [Breakpoint Properties] dialog box (figure 5.15) will appear. Click the [Software Break] tab and select the [Software Break] page (figure 5.33).

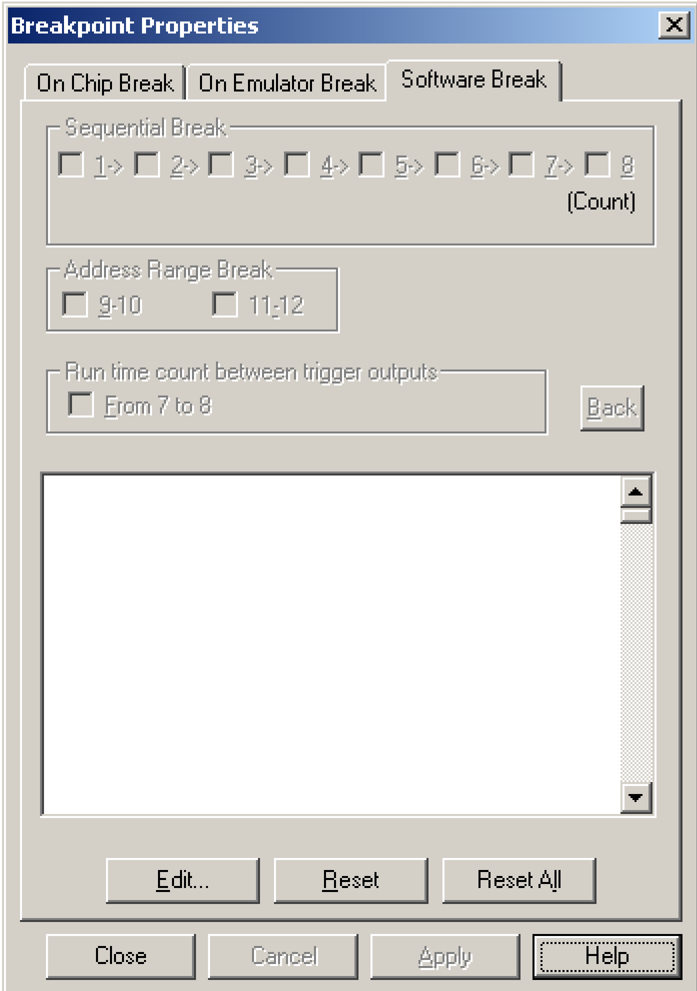



**Figure 5.33 [Breakpoint Properties] Dialog Box ([Software Break] Page)**

**Table 5.29 [Software Break] Page Options**

| Option | Description |
|---|---|
| [Edit…] | Allows the user to modify the software breakpoint settings. Clicking the [Edit...] button by selecting the list box displays the [Address] dialog box. |
| [Reset] | Clears the software breakpoint settings selected in the list box. |
| [Reset All] | Clears all software breakpoints. |

Clicking the [Edit…] button by selecting the list box in the [Software Break] page displays the [Software Break] dialog box. For the software break, only the [Address] page is displayed. Specify the breakpoint's address, then click the [OK] button.

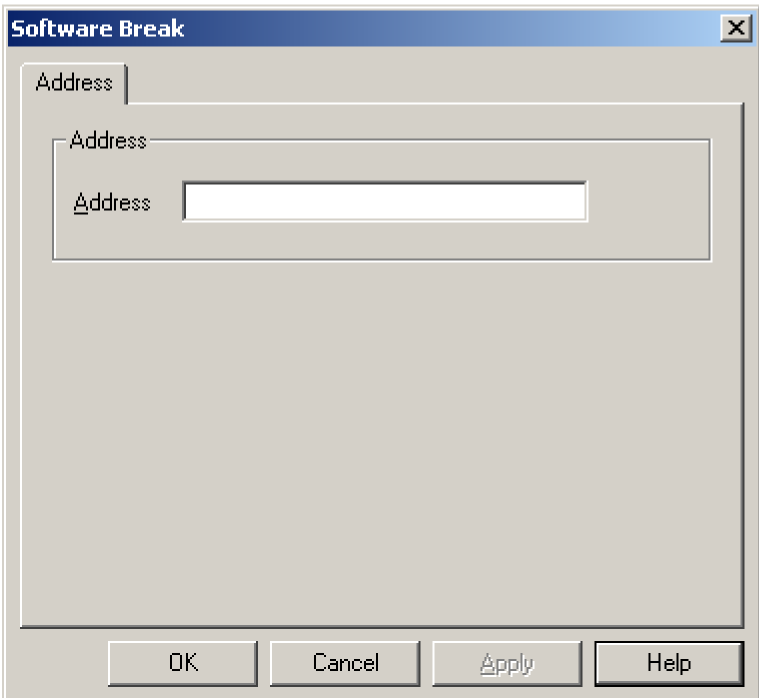



**Figure 5.34 [Software Break] Dialog Box ([Address] Page)**

The display returns to the [Software Break] page. The list box now displays the specified address. Click the [OK] button to close the [Breakpoint Properties] dialog box.

**Table 5.30 [Software Break] Dialog Box Option**

| Option | Description |
|---|---|
| [Address] | Sets the breakpoint's value as a numeric. |

When a software break is set, the instruction at the specified address is replaced. It is only possible to set a software break in the RAM area (including the emulation memory). However, it is not possible to set a software breakpoint at an address that satisfies any of these conditions:

- The address holds H'0000
- The address is in any area other than CS area (except the on-chip ROM/RAM area)
- The address of the delay slot for a delayed-branch instruction

Notes: 1. The maximum number of software breakpoints and [Stop At] settings allowed in the [Run Program] dialog box is 255. Therefore, when 255 software breakpoints have been set, any further specification made by using the [Stop At] item in the [Run Program] dialog box is invalid. Ensure that the total number of software breakpoints and settings made by using the [Stop At] item in the [Run Program] dialog box is 255 or less.

2. When a disabled breakpoint address is specified as a [Stop At] item in the [Run Program] dialog box, the breakpoint becomes enabled after the first time that execution subsequently stops at that address.
3. If it is not possible to correctly set a breakpoint when a session file is loaded, the breakpoint is registered as DISABLE in the [Breakpoints] window.
4. Software breakpoints are ignored during step execution.
5. Do not set a break immediately after the delayed branch instruction (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.
6. Do not allow the user program to modify memory at a breakpoint address.
7. The contents of the specified address where a breakpoint has been set are replaced by a break instruction during emulation.

### 5.6.5 Forced Break

A user program can be forcibly terminated by clicking the [STOP] button or by selecting [Halt] from the [Run] menu.

### 5.6.6 Break Due to Trace-Buffer Overflow

A break occurs when the trace buffer in the E6000H station overflows during trace acquisition.

Select [Stop execution and stop trace] in [When trace-buffer full] on the [Other] page of the [Trace Acquisition Properties] dialog box.

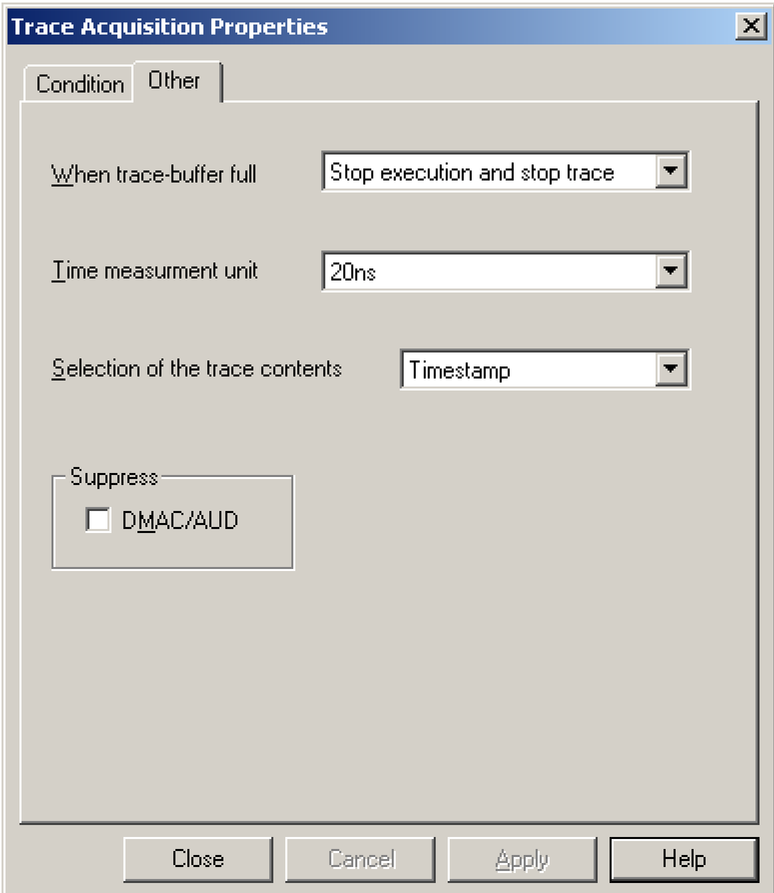



**Figure 5.35 [Trace Acquisition Properties] Dialog Box ([Other] Page)**

## 5.7 Realtime Trace Functions

The emulator allows realtime trace acquisition of up to 131,070 cycles of the SH7058 bus information. This has no effect on the user system during emulation.

A total of up to 65,535 lines of information can be displayed in the HDI's [Trace] window. For details on the [Trace] window, refer to the Hitachi Debugging Interface User's Manual (on the CD-R).

### 5.7.1 Trace Acquisition Condition

Trace Conditions 1 to 12 are available.

Specify a trace acquisition condition on each page of the [Trace Acquisition Properties] dialog box which will be displayed by clicking [Acquisition…] in the pop-up menu of the [Trace] window and the [Action] page in the [Trace Acquisition Condition Channel *n*] (*n*: integers 1 to 12).

The trace acquisition modes are shown in table 5.31.

**Table 5.31 Trace Acquisition Modes**

| Acquisition Mode | Description |
|---|---|
| Free trace | Trace acquisition is continuous; from the start of user-program execution until any of the break conditions is satisfied. |
| Sequential trace stop | An order in which trace conditions must be satisfied can be specified. When all of the conditions are satisfied in the specified order, trace acquisition will stop. Can be specified for Trace Conditions.* |
| Trace stop due to an overflow of trace buffer | Trace acquisition stops when the trace buffer in the emulator overflows.* |
| Trace stop | Trace acquisition stops when a specified condition is satisfied. In this mode, trace acquisition is stopped without halting user program execution.* |
| Address range trace | Instruction and operand accesses are traced during execution in the range between the start address and end address of a specified subroutine, or of a subroutine specified by its start and end addresses. However, when the specified subroutine calls other subroutines, the called subroutine will not be traced. |
| Conditional address range trace | In this mode, trace information is only gathered when access is to instructions and operands in the range specified by the start address and end address, during those bus cycles in which the specified conditions are matched. |
| Conditional trace | Trace is acquired only in the position where the specified conditions are satisfied. |
| Trigger output | A pulse is output from the trigger pin when the specified conditions are satisfied. |

Note: It takes 10 to 11 cycles from the satisfaction of conditions to the halt of trace acquisition.

**Free Trace Mode:** Trace information is acquired continuously from the start of user program execution until any of the break conditions is satisfied. The free trace mode is the default when [Free trace] in the [Trace Acquisition Properties] dialog box is selected regardless of the settings of other conditions.

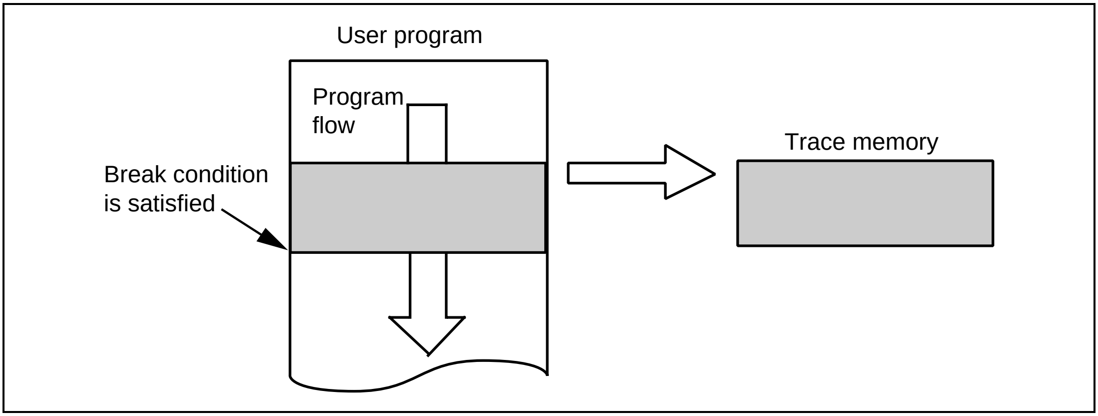



**Figure 5.36 Trace Acquisition in Free Trace Mode**

**Trace-Stop Mode:**

(a) Overview

Trace acquisition stops when the specified conditions are satisfied.

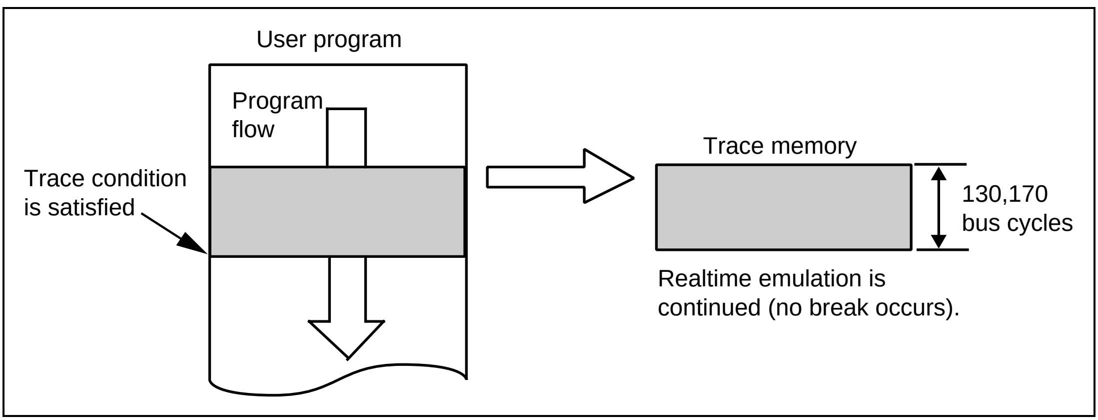



**Figure 5.37 Trace Acquisition in Trace-Stop Mode**

In this mode, execution of the user program will not be suspended but the emulator enters the trace-halt mode.

The trace stop conditions are shown in table 5.32. When all of the specified conditions (an AND condition) are satisfied, trace acquisition will stop.

**Table 5.32 Trace Stop Conditions**

| Condition | Description |
|---|---|
| Address bus | The condition is satisfied when the value on the address bus matches the specified value. |
| Data bus | The condition is satisfied when the value on the data bus matches the specified value. |
| Bus | The condition is satisfied when the specified access type and read/write cycle conditions match the specification. |
| External probe signal | The condition is satisfied when the external probe (PRB) signal levels match a specification. |
| External interrupt signal | The condition is satisfied when the external interrupt signal levels match a specification. |
| Satisfaction count | Trace acquisition stops when the above conditions are satisfied the specified number of times. |
| Delay | Trace acquisition stops the specified number of bus cycles after the above conditions have been satisfied. This condition can be set only for Trace Condition 7 and 11. |

(b) Setting Trace-Stop Mode Conditions

Trace Condition 7 is taken as an example of setting a trace-stop mode condition.

Place the cursor in the [Trace] window then click the right-hand mouse button to display the pop-up menu. Select [Acquisition] from the menu, and the [Trace Acquisition Properties] dialog box will appear. Select channel 7 from the list box.

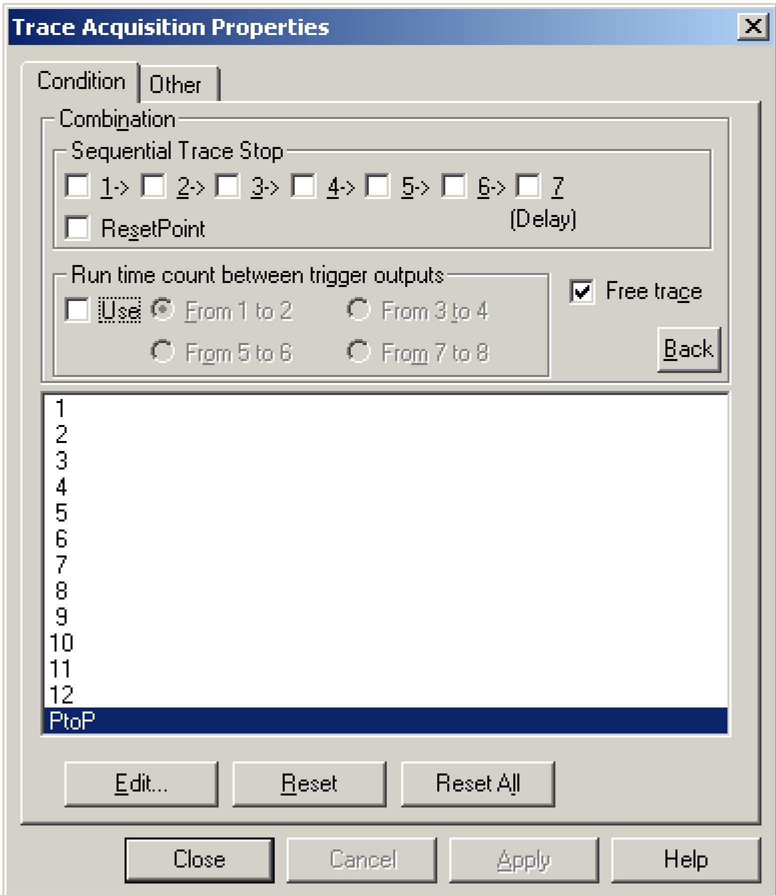



**Figure 5.38 [Trace Acquisition Properties] Dialog Box**

Table 5.33 shows the [Trace Acquisition Properties] options.

**Table 5.33 [Trace Acquisition Properties] Dialog Box Options**

| Option | Description |
|---|---|
| [Sequential Trace Stop] | Specifies channels (1 to 7) sequentially and the order of halting trace. A reset point is specified for ResetPoint. The trace is stopped at reset. |
| [Time measurement between trigger outputs] | Specifies channels that measure the trace time. The time is measured by clicking Use. Channels are used in combination of starting measurement and ending measurement; i.e. 1-2, 3-4, 5-6, and 7-8. |
| [Free Trace] | Selects free trace. |
| [Back] | Puts the contents set in the Combination group back when the dialog box has been displayed. |
| List box | Displays the setting status of Trace Condition.<br>The contents of conditions that are currently set are displayed.<br>Channels 1 to 12 and PtoP (Point to Point measurement) setting channel. |
| [Edit...] | Modifies the Trace Condition settings selected in the list box. Clicking this button opens the [Trace Condition X] dialog box. (X: 1, 2, …, 11, 12.) |
| [Reset] | Clears the Trace Condition settings selected in the [Condition] list box. |
| [Reset All] | Clears all Trace Condition settings. |

Notes: 1. When a combination of 1-2, 3-4, or 5-6 is selected for [Time measurement between trigger outputs], the period from starting measurement to ending measurement must be set with an interval of six or more bus cycles.

2. When a combination of 7-8 is selected for [Time measurement between trigger outputs], the period from starting measurement to ending measurement must be set with an interval of 15 or more bus cycles.

Double-click 7 in the list box on the [Condition] page by using a mouse. Click the [Edit...] button to open the [Trace Acquisition Condition Channel 7] dialog box. The trace acquisition mode is specified on the [Action] page.

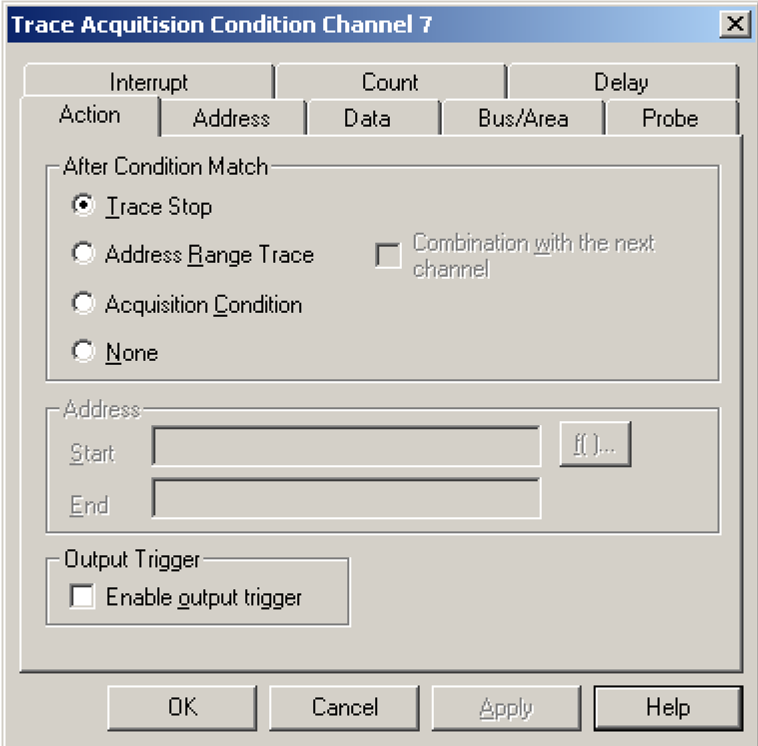



**Figure 5.39 [Action] Page**

**Table 5.34 [Action] Page Options**

<table>
<tr><th>Option</th><th colspan="3">Description</th></tr>
<tr><td rowspan="6">[After Condition Match]</td><td colspan="3">Specifies the processing after the trace condition has been satisfied. Select the following items.</td></tr>
<tr><td>[Trace Stop]</td><td colspan="2">Sets the trace halt mode.<br>Halting trace is selected after the trace condition has been satisfied.</td></tr>
<tr><td rowspan="2">[Address Range Trace]</td><td colspan="2">Sets the address range trace mode.<br>The trace is acquired by specifying the address range.</td></tr>
<tr><td>Combination with the next channel (selected for channels 1, 3, 5, and 7)</td><td>Uses the next channel to additionally set other acquisition conditions in the range trace and sets the conditional address range trace mode.</td></tr>
<tr><td>[Acquisition Condition]</td><td colspan="2">Sets the conditional trace mode.</td></tr>
<tr><td>[None]</td><td colspan="2">Selects this option when the starting measurement and ending measurement conditions of [Time measurement between trigger outputs] are set, which does not affect the acquired trace.</td></tr>
<tr><td>[Address]</td><td colspan="3">Sets the address range when Address Range Trace is selected.</td></tr>
<tr><td>[Output Trigger]</td><td colspan="3">Outputs a trigger externally after the trace condition has been satisfied.</td></tr>
</table>

Table 5.35 shows the dialog box options for each channel.

**Table 5.35 [Trace Acquisition Condition Channel] Dialog Box Pages**

| Dialog Box | Page | Description |
|---|---|---|
| Channels 1, 2, 3, 4, 5, 6, 8, 9, 10, and 12 | [Action] | Selects the processing after the trace condition has been satisfied. |
| | [Address] | Sets address bus conditions. |
| | [Data] | Sets data bus conditions. |
| | [Bus/Area] | Sets access type and read/write cycle conditions. |
| | [Probe] | Sets external probe signal conditions. |
| | [Interrupt] | Sets external interrupt signal conditions. |
| | [Count] | Sets satisfaction count conditions. |
| Channels 7 and 11 | [Action] | Selects the processing after the trace condition has been satisfied. |
| | [Address] | Sets address bus conditions. |
| | [Data] | Sets data bus conditions. |
| | [Bus/Area] | Sets access type and read/write cycle conditions. |
| | [Probe] | Sets external probe signal conditions. |
| | [Interrupt] | Sets external interrupt signal conditions. |
| | [Count] | Sets satisfaction count conditions. |
| | [Delay] | Sets delay conditions. |

Specify the required conditions on the corresponding pages, then click the [OK] button.

The options on each page are the same as those of the on-emulator break, except the [Action] and [Delay] pages. For details of the options on each page, refer to section 5.6.3, On-Emulator Break. Figure 5.40 shows the setting of the [Delay] page and table 5.36 shows the options.

Notes: 1. Set a multiple of two as the address value when the 16-bit area is accessed.

2. Set a multiple of four as the address value when the 32-bit area is accessed.

3. The trace-stop mode condition must be set with an interval of six or more bus cycles.

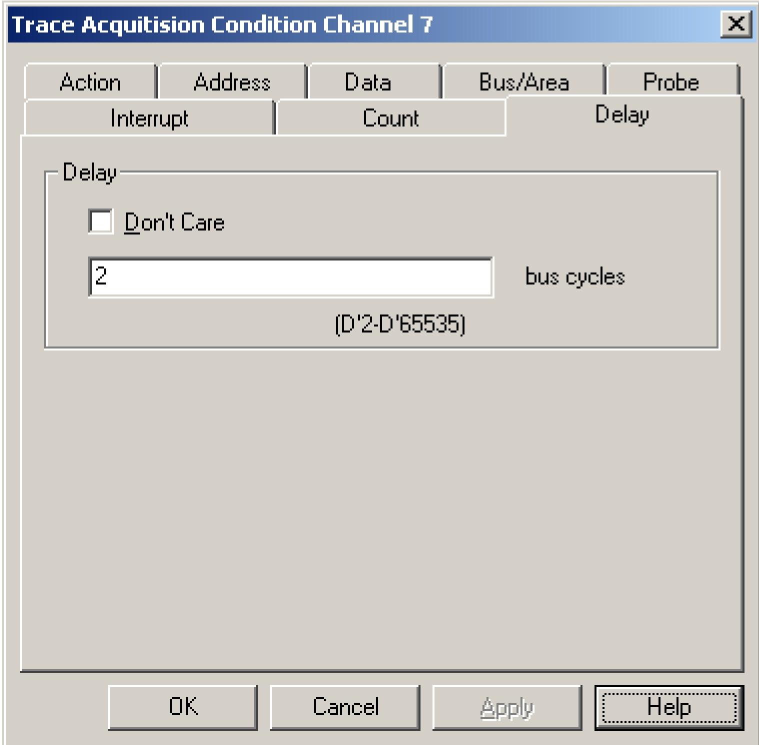



**Figure 5.40 [Delay] Page**

**Table 5.36 [Delay] Page Options**

| Option | Description |
|---|---|
| [Don't Care] | Specifies no satisfaction count conditions. |
| Input area | Sets a value for the satisfaction count condition as a numeric. The default is D'1. Values D'2 to D'65535 can be set. |

The [Trace Acquisition Condition Channel 7] dialog box closes and the display returns to the [Condition] page. The specified trace-stop conditions will now be displayed as condition 7 in the list box. Click the [Close] button to close the [Trace Acquisition Properties] dialog box.

Trace stop conditions for the other channels are specified in the same way.

Note: It takes 10 to 11 cycles while the delay condition has been satisfied until the trace acquisition is stopped. Set the delay condition considering the cycle before the trace acquisition is stopped.

**Sequential Trace Stop:**

(a) Overview

A sequential trace stop occurs after a set of channels of a trace-stop condition has been satisfied in a specified order.

This function uses the trace condition; a sequence of up to seven trace-stop conditions and one reset point can be specified as the trace condition.

The user must consider the order of satisfaction in specifying trace-stop conditions for the trace condition; tracing only stops when the trace-stop conditions have all been satisfied in the order specified by the user.

When the reset point is passed, the record of sequential trace-stop conditions that have been satisfied to that point is cleared, and the emulator restarts checking for satisfaction of the sequential trace-stop conditions from the first condition.

(b) Setting a Sequential Trace Stop

The 5-level sequential trace is taken as an example of using the sequential trace stop.

Place the cursor in the [Trace] window then click the right-hand mouse button to display the pop-up menu. Select [Acquisition].

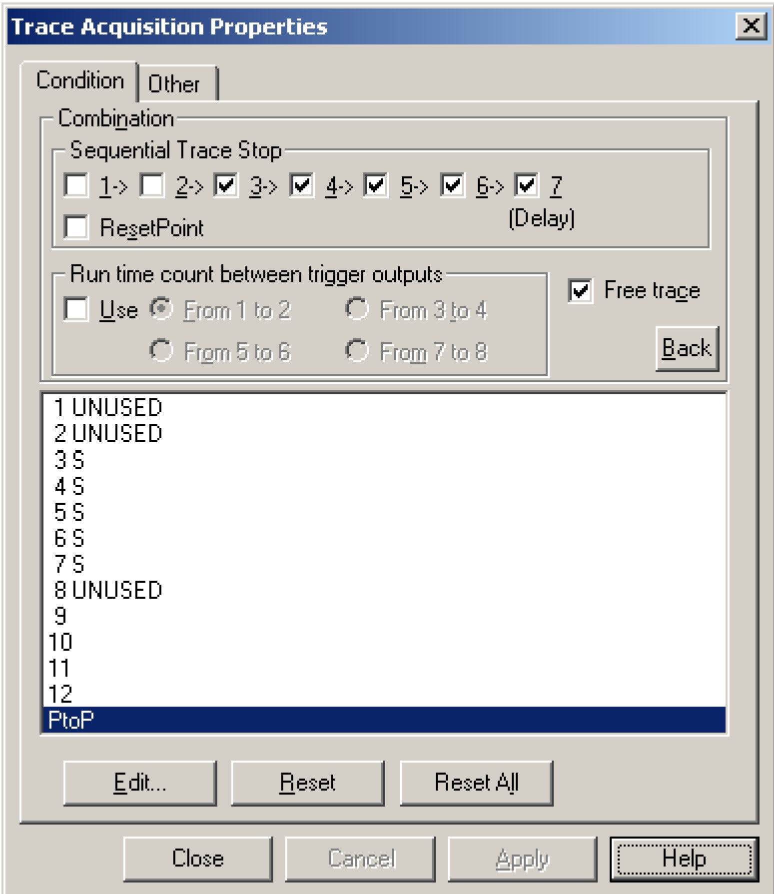



**Figure 5.41 [Trace Acquisition Properties] Dialog Box (Sequential Trace Stop)**

Checking 5 for [Sequential Trace Stop] enables using channels 3 to 7 as a sequential trace stop. Conditions for channels 3 to 7 before setting the sequential trace stop are maintained as the sequential trace conditions. Set the trace stop condition for channels 3 to 7.

Notes: 1. Set a multiple of two as the address value when the 16-bit area is accessed.

2. Set a multiple of four as the address value when the 32-bit area is accessed.

3. The sequential trace stop condition must be set with an interval of six or more bus cycles.

**Trace Stop Due to Trace Buffer Overflow:** A break occurs when the trace buffer in the E6000H station overflows during trace acquisition.

Select [Stop trace] in [When trace-buffer full] on the [Other] page of the [Trace Acquisition Properties] dialog box.

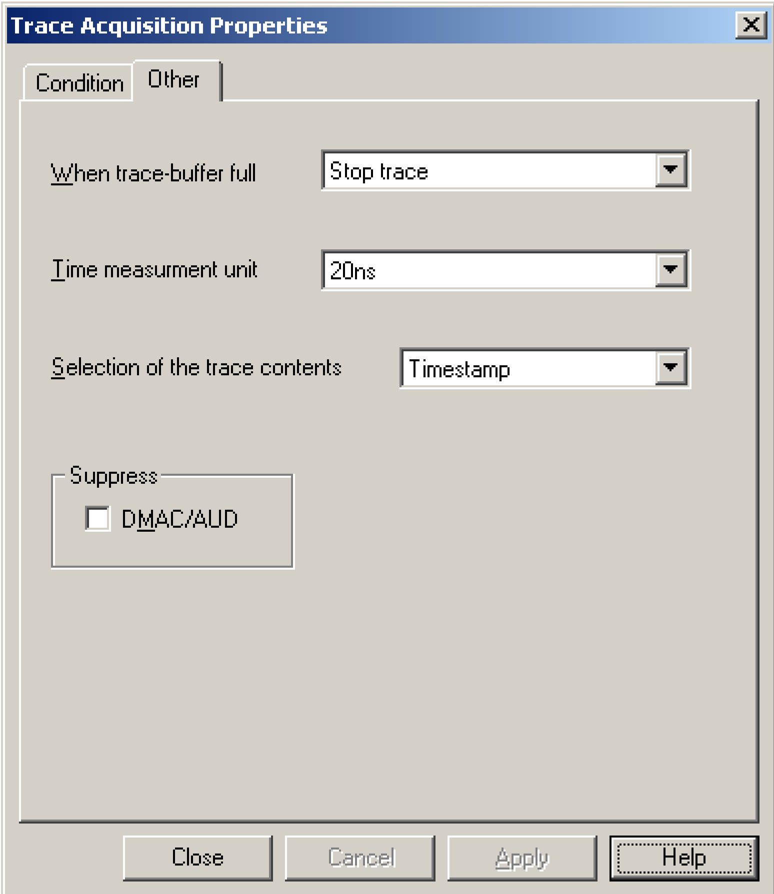



**Figure 5.42 [Trace Acquisition Properties] Dialog Box ([Other] Page)**

**Subroutine Trace Mode:**

(a) Overview

Instruction and operand accesses are traced during execution in the range between the start address and end address of a specified subroutine, or of a subroutine specified by its start and end addresses. However, when the specified subroutine calls other subroutines, the called subroutine will not be traced.

The specification of a subroutine trace mode is described (Trace Condition 1 is taken as an example). Select the [Trace] window and click the right-hand mouse button. Select [Acquisition] from the menu, then the [Trace Acquisition Properties] dialog box is displayed. Select Channel 1 from the list box. Selecting [Address Range Trace] on the [Action] page sets the subroutine trace mode.

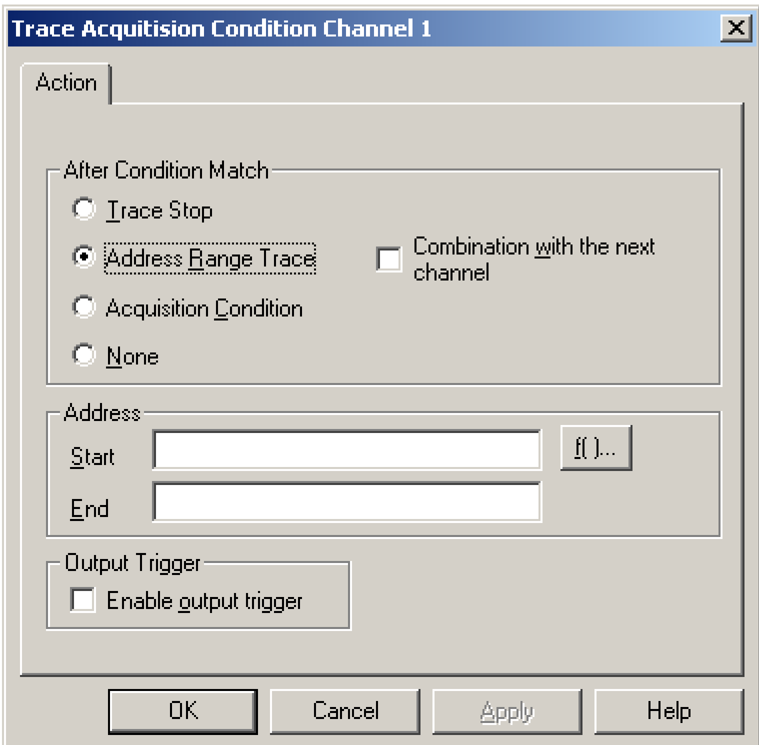



**Figure 5.43 Setting the Subroutine Trace Mode ([Action] Page)**

When [Combination with the next channel] is selected and each page of the next channel is set, the trace acquisition condition can be set for the subroutine.

When [Combination with the next channel] is selected (only the odd channels can be selected), set each page of the next channel (even channel) corresponding to the selected channels, and click the [OK] button to set the trace acquisition condition.

The [Trace Condition 1] dialog box closes and the display returns to the [Trace Acquisition Properties] dialog box. The specified trace acquisition conditions will now be displayed as channel 1 in the list box. Click the [OK] button to close the [Trace Acquisition Properties] dialog box.

Subroutine trace modes are specified in the same way.

The settings on each page are the same as those of the dialog box for the on-emulator break, except the [Action] page. For details on each page, refer to section 5.6.3, On-Emulator Break.

Notes: 1. Set a multiple of two as the address value when the 16-bit area is accessed.
2. Set a multiple of four as the address value when the 32-bit area is accessed.
3. The subroutine trace mode condition must be set with an interval of six or more bus cycles.

**Conditional Trace Mode:**

(a) Overview

Cycles are only acquired where the specified conditions are satisfied.

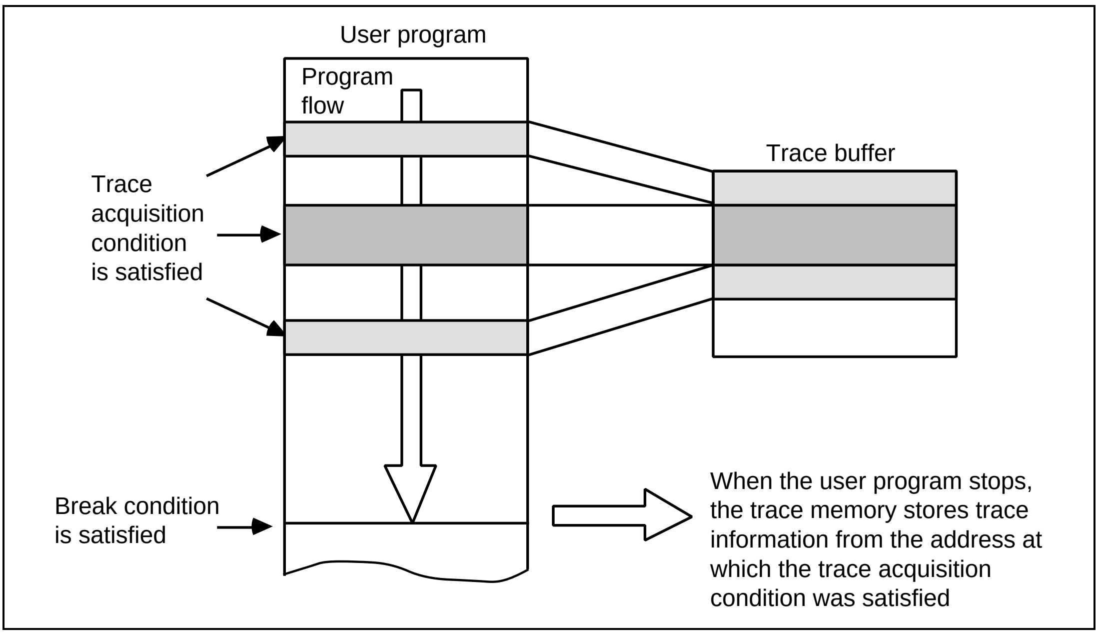



**Figure 5.44 Example of Conditional Trace Mode**

The conditions for conditional trace mode are shown in table 5.37. Information is acquired when all of the specified conditions (an AND condition) are satisfied.

**Table 5.37 Conditional Trace Modes**

| Condition | Description |
|---|---|
| Address bus | Satisfied when the value on the address bus matches the specified value. |
| Data bus | Satisfied when the value on the data bus matches the specified value. |
| Bus/access type | Satisfied when the bus status condition is as specified. |
| External interrupt signal | Satisfied when the external interrupt signal levels match the specified conditions. |
| External probe signal | Satisfied when the external probe (PRB) signal levels match the specified conditions. |
| Satisfaction count | Satisfied when the specified conditions match the satisfied counts. |

Notes: 1. Set a multiple of two as the address value when the 16-bit area is accessed.
2. Set a multiple of four as the address value when the 32-bit area is accessed.
3. The conditional trace mode condition must be set with an interval of six or more bus cycles.

Trace Condition 1 is taken as an example of setting a conditional trace mode.

Place the cursor in the [Trace] window then click the right-hand mouse button to display the pop-up menu. Select [Acquisition…] from the menu, and the [Trace Acquisition Properties] dialog box will appear. Select Channel 1 from the list box. Select [Acquisition Condition] on the [Action] page to set the conditional trace mode.

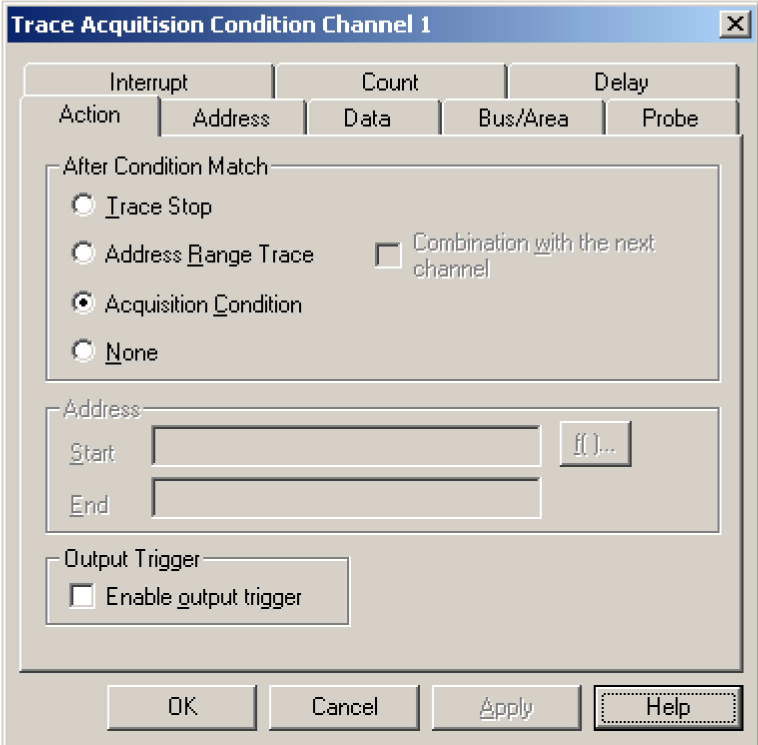



**Figure 5.45 Setting the Conditional Trace Mode ([Action] Page)**

**Trigger Output:** A low-level pulse is output from the trigger-output probe when the conditions specified for Trace Condition are satisfied during execution of the user program. For details, refer to figure 5.39 and table 5.34.

**Other Conditions:** In the [Other] page of the [Trace Acquisition Properties] dialog box, tracing can be specified to stop on the overflow of the trace buffer, suppressing of DMAC or AUD cycles can be selected, and the minimum period for time stamping of acquired bus-tracing information can be specified.

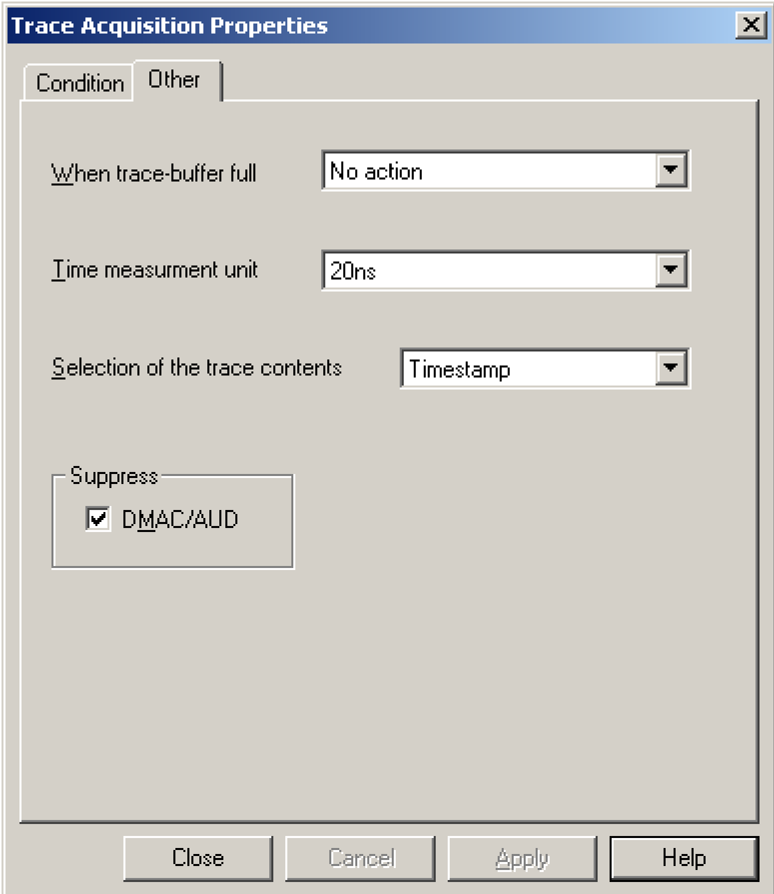



**Figure 5.46 [Trace Acquisition Properties] Dialog Box ([Other] Page)**

(a) [Suppress]

Cycles that are suppressed (not acquired) are selected. The DMAC and AUD cycles are not acquired when [DMAC/AUD] is checked.

(b) [Time measurement unit]

The minimum period for time stamping is specified.

**Table 5.38 [Time measurement unit] Group Box Option**

| Option | Description |
|---|---|
| [Time measurement unit] | Selects the minimum time for the time stamping of the measured bus trace information from among the values listed below.<br>20ns: Time stamping is in minimum time units of 20 ns (default).<br>1.6us: Time stamping is in minimum time units of 1.6 µs.<br>52us: Time stamping is in minimum time units of 52 µs.<br>CLOCK: Time stamping is in terms of the number of bus-clock cycles, i.e., is synchronized with the cycles of the system clock ($\phi$) signal.<br>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 ($\phi$) signal.<br>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 ($\phi$) signal.<br>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 ($\phi$) signal. |

Click the [Apply] button to set the minimum time, then click the [Close] button.

### 5.7.2 External Bus Trace Timing

The acquisition of trace information is synchronized with the rising edge of T2 cycles of the CK signal.

In each bus cycle, the number of cycles between the end of the previous bus cycle and the end of the current bus cycle is measured.

An example of a bus-trace timing for the external probe signal is shown in figure 5.47.

Note: When the external probe signal information is traced, changes in the signal may not be traced if its level changes with certain timings. This is because the signal is not synchronized with the CK signal.

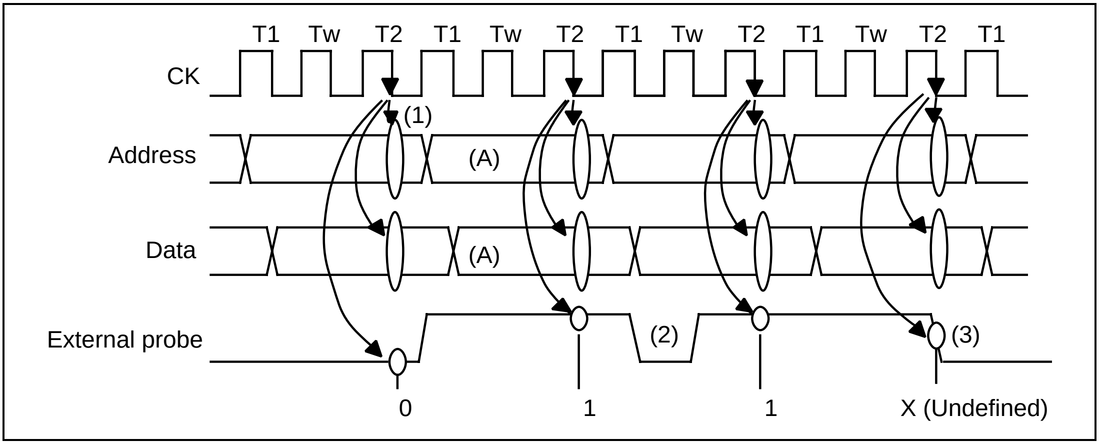



**Figure 5.47 Tracing External Probe Signal**

(1) External probe signal

(a) Information is traced on the falling edge of the T2 cycle of the CK signal (figure 5.47 (1)).

(b) When the level of the external probe signal changes while information is being acquired, this change will not be included in the trace information (figure 5.47 (2)).

(c) If the sampling edge and the level of an external probe signal change at the same time, the information traced will be undefined (figure 5.47 (3)). If the sampling edge and asynchronous input signal such as NMI or IRQ changes at the same time, the information traced will also be undefined.

(2) Number of clocks

Up to three clock cycles of tracing can take place in one bus cycle (A).

### 5.7.3 Trace Display

Select the trace display format on the [General] page of the [Trace Filter Properties] dialog box. Select [Filter…] from the pop-up menu of the [Trace] window to produce this dialog box.

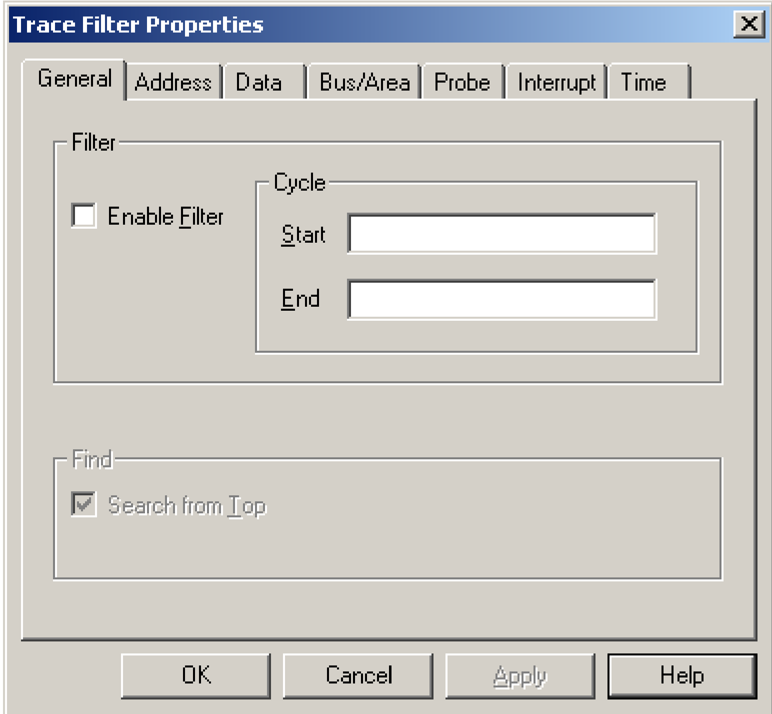



**Figure 5.48 [Trace Filter Properties] Dialog Box ([General] Page)**

**Trace Display:** The range for display can be specified by setting the start and end pointers in bus cycles (bus cycle pointers) in [Start] and [End] on the [General] page of the [Trace Filter] dialog box. The pointer is a value relative to the location at which the delay condition has been satisfied. Number of bus cycles before the delay condition has been satisfied are indicated by a minus sign (-), while numbers of cycles after the condition's satisfaction are displayed with a plus sign (+).

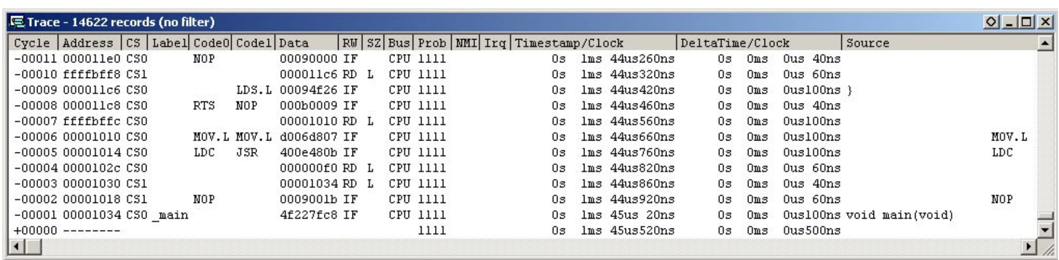



**Figure 5.49 [Trace] Window**

The items shown in table 5.39 are displayed as trace information in the [Trace] window.

**Table 5.39 Trace Information Items and Display Format in [Trace] Window**

| Item | Description and Format |
|---|---|
| Cycle | Bus cycle pointer. |
| Address | 32-bit address bus values in 4-bit units. The invalid cycle is shown as '--------'. |
| CS | Areas (CS0 to 3) in the user expansion space in the expansion mode. Nothing is displayed in the non-expansion mode. This item may be displayed before or after the bus cycle that accessed the area in the user expansion space. |
| Label | Label name related to an address (if any). |
| Code0 | Upper 16-bit instruction code being executed. Nothing is displayed here in cycles that are not instruction-execution cycles. |
| Code1 | Lower 16-bit instruction code being executed. Nothing is displayed here in cycles that are not instruction-execution cycles. |
| Data | 32-bit data bus values in 4-bit units. |
| RW | Whether the cycle was for reading, writing, or fetching.<br>RD: read cycle<br>WR: write cycle<br>IF: fetch cycle |
| SZ | Data size to be accessed.<br>L: longword<br>W: word<br>B: byte |
| Bus | Bus-cycle type conditions for trace. Cycles for CPU, DMAC, and AUM are displayed as CPU, DMAC, and AUM, respectively. |
| Prob | External probe (PRB) signal state.<br>x4x3x2x1 (xn is the state of PRBn) (0: low level; 1: high level) |
| NMI | NMI signal state. (0: low level; 1: high level) |
| IRQ | IRQ0 to IRQ7 signal state.<br>x7x6x5x4x3x2x1x0 (xn is the state of IRQn) (0: low level; 1: high level) |
| Time Stamp | Time stamp.<br>xxxsxxxmsxxxusxxxns<br>(s: second; ms: millisecond; us: microsecond; ns: nanosecond)<br>When IRQ is selected, an approximate value is displayed for Time Stamp. Time Stamp and Clock are displayed by EORed. |

**Table 5.39 Trace Information Items and Display Format in [Trace] Window (cont)**

| Item | Description and Format |
|---|---|
| Deltatime | Difference of time immediately before the cycle. |
| Clock | Number of clock cycles from the end of the previous bus cycle to the end of the current bus cycle.<br>Only one of Time Stamp or Clock can be displayed at a time. |
| Source | The corresponding source line to the cycle.<br>Clicking in the [Source] column jumps to the [Source] window. |

### 5.7.4 Trace Search Functions

The emulator has the two functions for searching for trace information that are shown in table 5.40.

**Table 5.40 Trace Search Functions**

| Search Function | Description |
|---|---|
| Trace Filter | Displays all trace information that satisfies the specified conditions. |
| Trace Find | Jumps to the next record of trace information that satisfies the specified conditions |

**Trace Filter Function:** This function displays all trace information that satisfies the conditions specified in the [Trace] window. The search conditions can be specified in the [Trace Filter Properties] dialog box.

Place the cursor in the [Trace] window then click the right-hand mouse button to display the pop-up menu. Select [Filter…] from the menu, and the [Trace Filter Properties] dialog box will appear. Select [Enable Filter] on the [General] page.

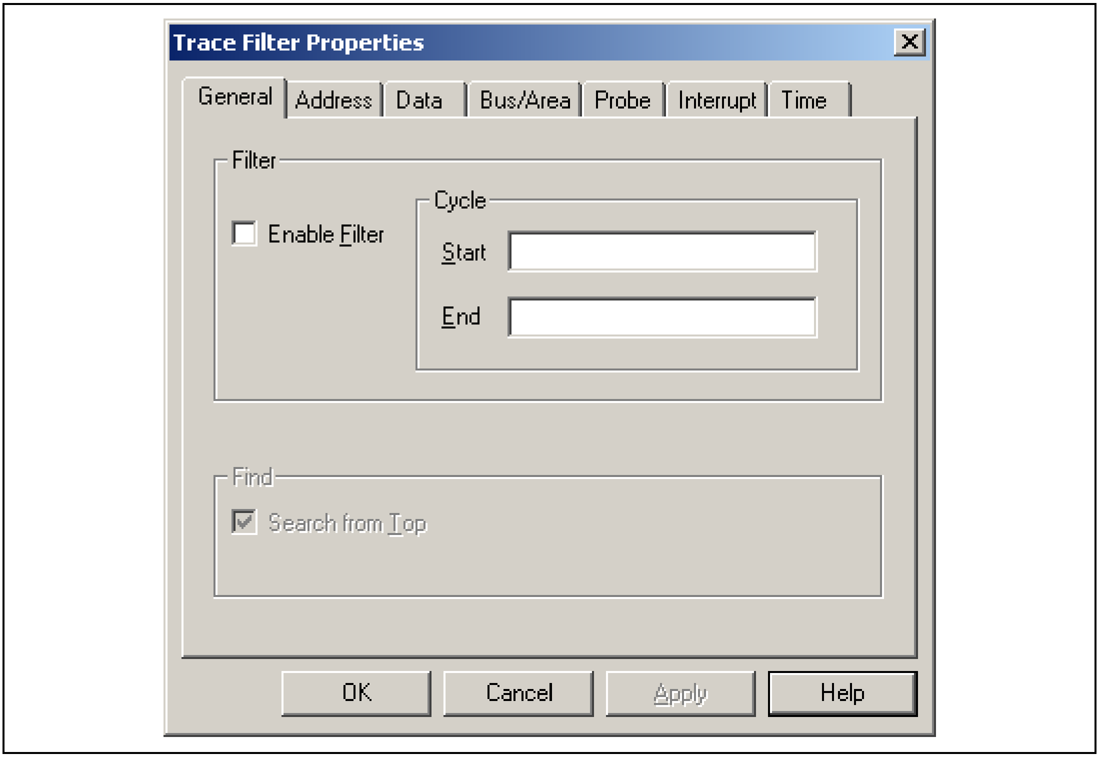



**Figure 5.50 [Trace Filter Properties] Dialog Box ([General] Page)**

If there is no trace information in updating the [Trace] window, 'No trace data' is displayed in the window (nothing is displayed when there is no trace information at opening the [Trace] window).

**Table 5.41 [General] Page Option**

| Option | Description |
|---|---|
| Filter | Sets a range to be displayed. |
| [Cycle] | Set negative values for as cycles before the delay condition was satisfied with the bus cycle on which the conditions are satisfied as the origin (0). [Start] is the pointer to the first bus cycle. [End] is the pointer to the last bus cycle. The input area must always be set. |

Specify the required conditions on the corresponding pages, then click the [OK] button. The [Trace Filter Properties] dialog box closes and the search results are displayed in the [Trace] window.

The trace search conditions that can be specified and the corresponding pages in the [Trace Filter Properties] dialog box are shown in table 5.42.

**Table 5.42 Trace Search Conditions and Pages in the [Trace Filter Properties] Dialog Box**

| Page | Condition | Description |
|---|---|---|
| [General] | — | Sets trace-search range. |
| [Address] | Address bus | Searches for records in which the value on the address bus, program counter, or memory access area matches the specified condition. |
| [Data] | Data bus | Searches for records in which the value on the data bus matches the specified condition. |
| [Bus/Area] | Bus state | Searches for records in which access type, read/write access, and BREQ signal match the specified condition. |
| [Probe] | External probe signal | Searches for records in which the external probe signal levels match the specified condition. |
| [Interrupt] | External interrupt signal | Searches for records in which the NMI, RES, MRES, or IRQ0 to IRQ7 signal levels match the specified condition. |
| [Time] | Time stamp | Searches for records in which the time stamp matches the specified condition (time or range). |

The setting of conditions for [Data] and [Probe] is the same as setting the on-emulator break condition. For details on specifying the conditions, refer to section 5.6.3, On-Emulator Break.

The descriptions given below are of [Address], [Bus State], [Interrupt], and [Time] settings.

Note: During using the Trace Filter function, do not save the trace.

(a) [Address] Page

Use the [Address] page to specify bus conditions for use in searching for address bus information.

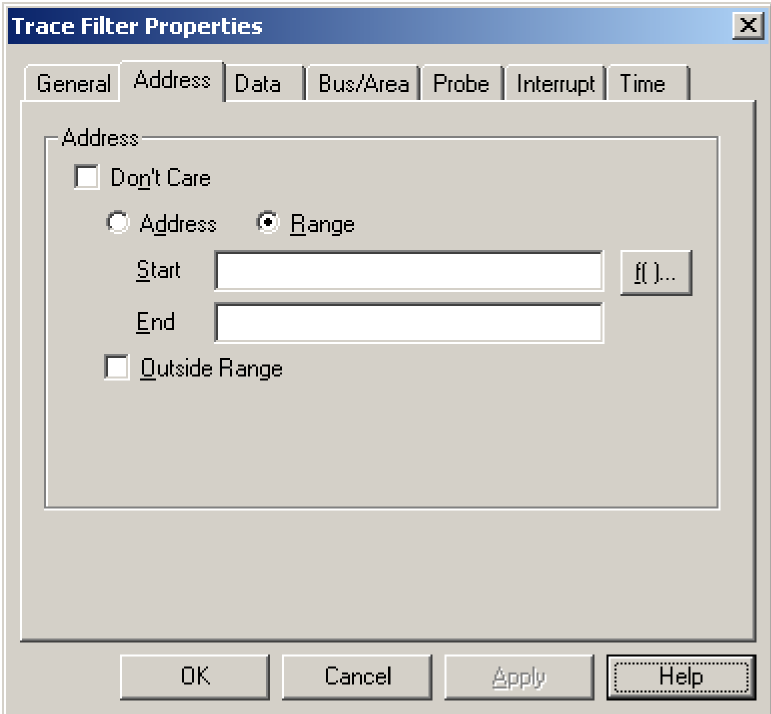



**Figure 5.51 [Trace Filter Properties] Dialog Box ([Address] Page)**

**Table 5.43 [Address] Page Options**

| Option | Description |
|---|---|
| [Don't Care] | No address bus condition is set. |
| [Address] | Selects the range set as [Start] or [Mask] as a trace-search condition. |
| [Range] | Selects the range set as [Start] to [End] as a trace-search condition. |
| [Start] | Sets the (start) address value as a numeric or a symbol. |
| [End] | When [Range] is selected, sets the (end) address value as a numeric or symbol. |
| [Outside Range] | Sets the condition that has not been set as a trace-search condition. |

(b) [Bus/Area] Page

Use the [Bus/Area] page to specify bus conditions for use in searching, i.e., access type and read or write cycle information.

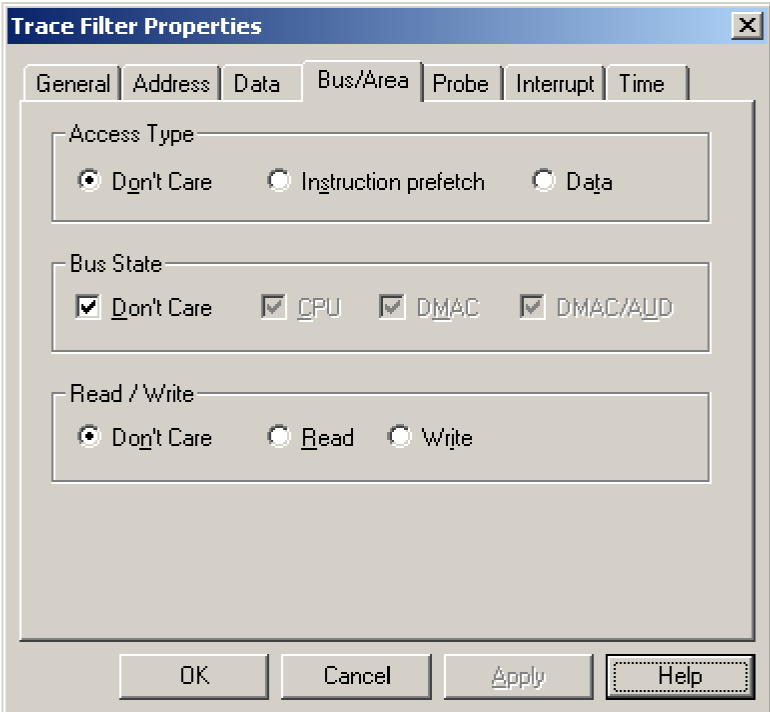



**Figure 5.52 [Trace Filter Properties] Dialog Box ([Bus/Area] Page)**

**Table 5.44 [Bus/Area] Page Options**

[Access Type] Group Box

| Option | Description |
|---|---|
| [Don't Care] | Sets all access types as the search condition. |
| [Instruction prefetch] | Sets CPU instruction fetch cycles as the search condition. |
| [Data] | Sets data access cycles as the search condition. |

[Bus State] Group Box

| Option | Description |
|---|---|
| [Don’t Care] | Sets all access types as the search condition. |
| [CPU] | Sets CPU cycles as the search condition. |
| [DMAC/AUD] | Sets DMAC/AUD cycles as the search condition. |

[Read/Write] Group Box

| Option | Description |
|---|---|
| [Don’t Care] | Sets all access types as the search condition. |
| [Read] | Sets read cycles as the search condition. |
| [Write] | Sets write cycles as the search condition. |

(c) [Interrupt] Page

Use the [Interrupt] page to specify conditions for the external interrupt signals (IRQ0 to IRQ7) and NMI signal for use in searching.

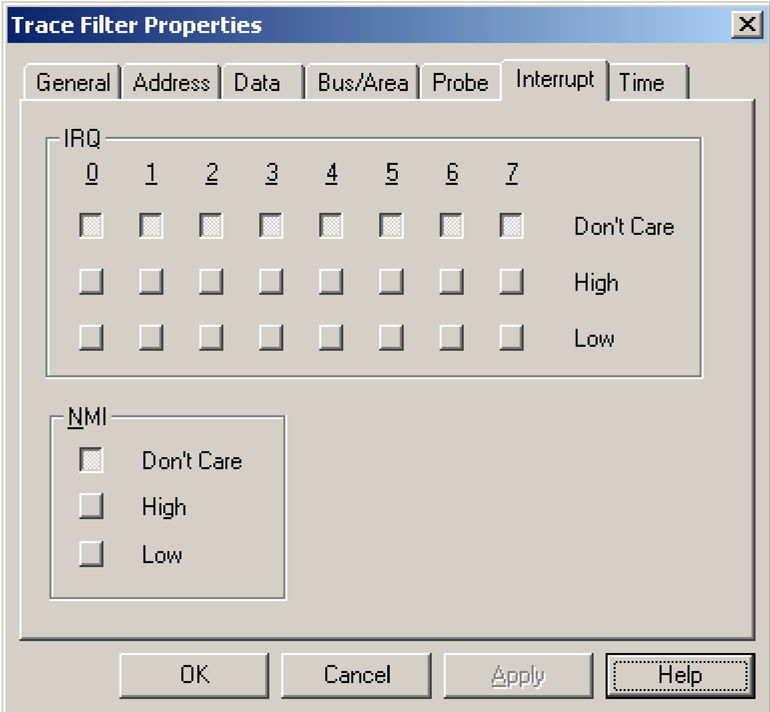



**Figure 5.53 [Trace Filter Properties] Dialog Box ([Interrupt] Page)**

**Table 5.45 [Interrupt] Page Options**

[IRQ0 to IRQ7] Group Box

| Option | Description |
|---|---|
| [Don’t Care] | The state of the external interrupt signal is not a search condition. |
| [High] | Sets the high level of the external interrupt signal as a search condition. |
| [Low] | Sets the low level of the external interrupt signal as a search condition. |

[NMI] Group Box

| Option | Description |
|---|---|
| [Don’t Care] | The state of the NMI signal is not a search condition. |
| [High] | Sets the high level of the NMI signal as a search condition. |
| [Low] | Sets the low level of the NMI signal as a search condition. |

(d) [Time] Page

Use the [Time] page to specify time-stamp conditions for use in the search for external bus trace information.

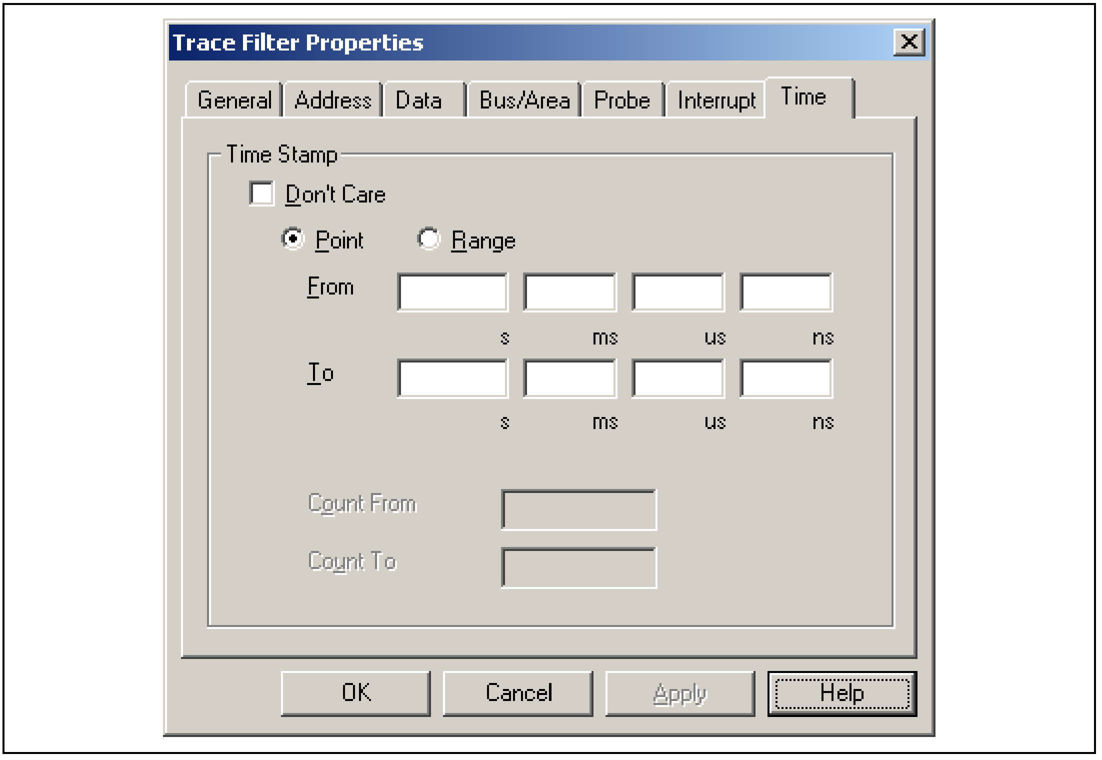



**Figure 5.54 [Trace Filter Properties] Dialog Box ([Time] Page)**

**Table 5.46 [Time] Page Options**

| Option | Description |
|---|---|
| [Don't Care] | No time stamp condition is set. |
| [Point] | Sets the trace-search condition to values above the value set as [From]. |
| [Range] | Sets the range set by [From] to [To] as a trace-search condition. |
| [From] | Sets the time-stamp value from which the valid range starts as a numeric (decimal). When any number is omitted, 0 is assumed.<br>[s]: (start) second (0 to 999999)<br>[ms]: (start) millisecond (0 to 999)<br>[us]: (start) microsecond (0 to 999)<br>[ns]: (start) nanosecond (0 to 999) |
| [To] | When [Range] is selected, sets the time-stamp value of the end of the range of valid values as a numerical value (decimal). When any number is omitted, 0 is assumed.<br>[s]: (start) second (0 to 999999)<br>[ms]: (start) millisecond (0 to 999)<br>[us]: (start) microsecond (0 to 999)<br>[ns]: (start) nanosecond (0 to 999) |

Note: Do not use this page if number of clock cycles have been selected for display.

### 5.7.5 Trace Find Functions

This function makes the display jump to the trace information item that satisfies the specified conditions in the [Trace] window. The search conditions can be specified in the [Trace Find Properties] dialog box. Search conditions are the same as those for the trace filter functions.

Place the cursor in the [Trace] window then click the right-hand mouse button to display the pop-up menu. Select [Find…] from the menu. Click [Find Next] in the pop-up menu of the [Trace] window to jump to the next trace information item that satisfies the specified conditions.

(a) [General] page

Select the [General] page, then select the [Search from top] check box to search from the start of the trace information.

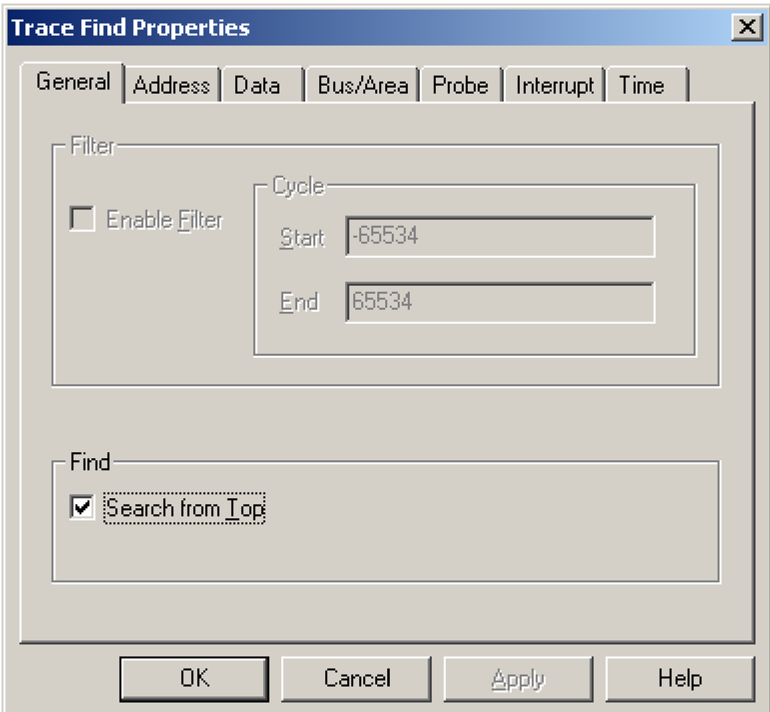



**Figure 5.55 [Trace Find Properties] Dialog Box ([General] Page)**

## 5.8 Measurement of Execution Time

### 5.8.1 Measuring Normal Execution Time

This function provides a way of measuring the total time taken to execute the user program. The user can use any of the methods shown in section 5.3.1, Execution, to start the user program. The total execution time is the total time with the user program in execution; from the start of execution until the program stops due to the satisfaction of a break condition.

The resulting measurement is displayed next to [Run Time Count] in the [Platform] page of the [System Status] window.

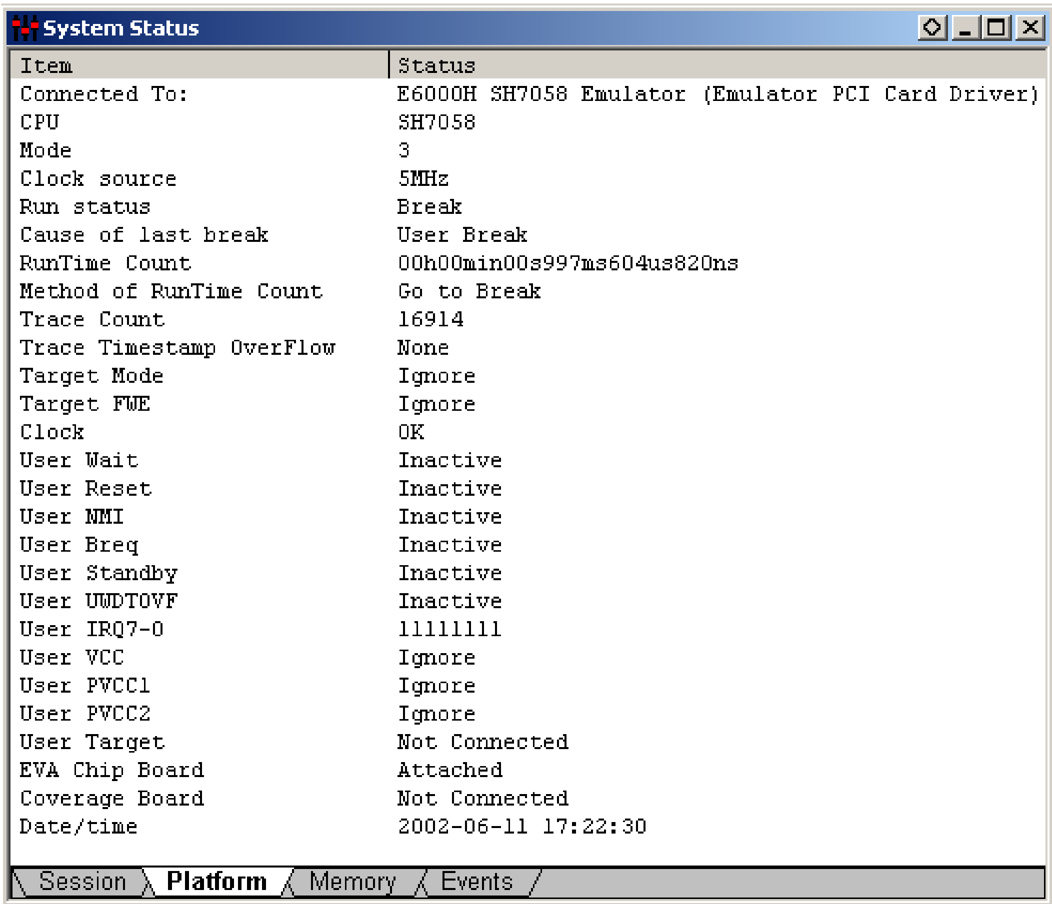



**Figure 5.56 [System Status] Window (Display of Execution Time)**

In the window, the user program execution time will be displayed as decimal numbers of hours, minutes, etc. The user can use the [Timer Resolution] on the [General] page in the [Configuration] dialog box to set a maximum period of measurement. The maximum times are approximately 15,860 hours (with a sampling interval during execution of 52 µs), 488 hours (with a sampling interval during execution of 1.6 µs), or 6 hours (with a sampling interval during execution of 20 ns).

### 5.8.2 Measuring Execution Time between Satisfaction of Specified Conditions

(1) Point To Point Time Measurement

This function is implemented by measuring the time interval of the user program with a trigger output of On Chip Break.

The execution time is measured from the satisfaction of On Chip Break 7 until the satisfaction of On Chip Break 8. After On Chip Break 7 has been satisfied, the user program stops on the satisfaction of On Chip Break 8. ‘On Chip Break’ will be displayed as the cause.

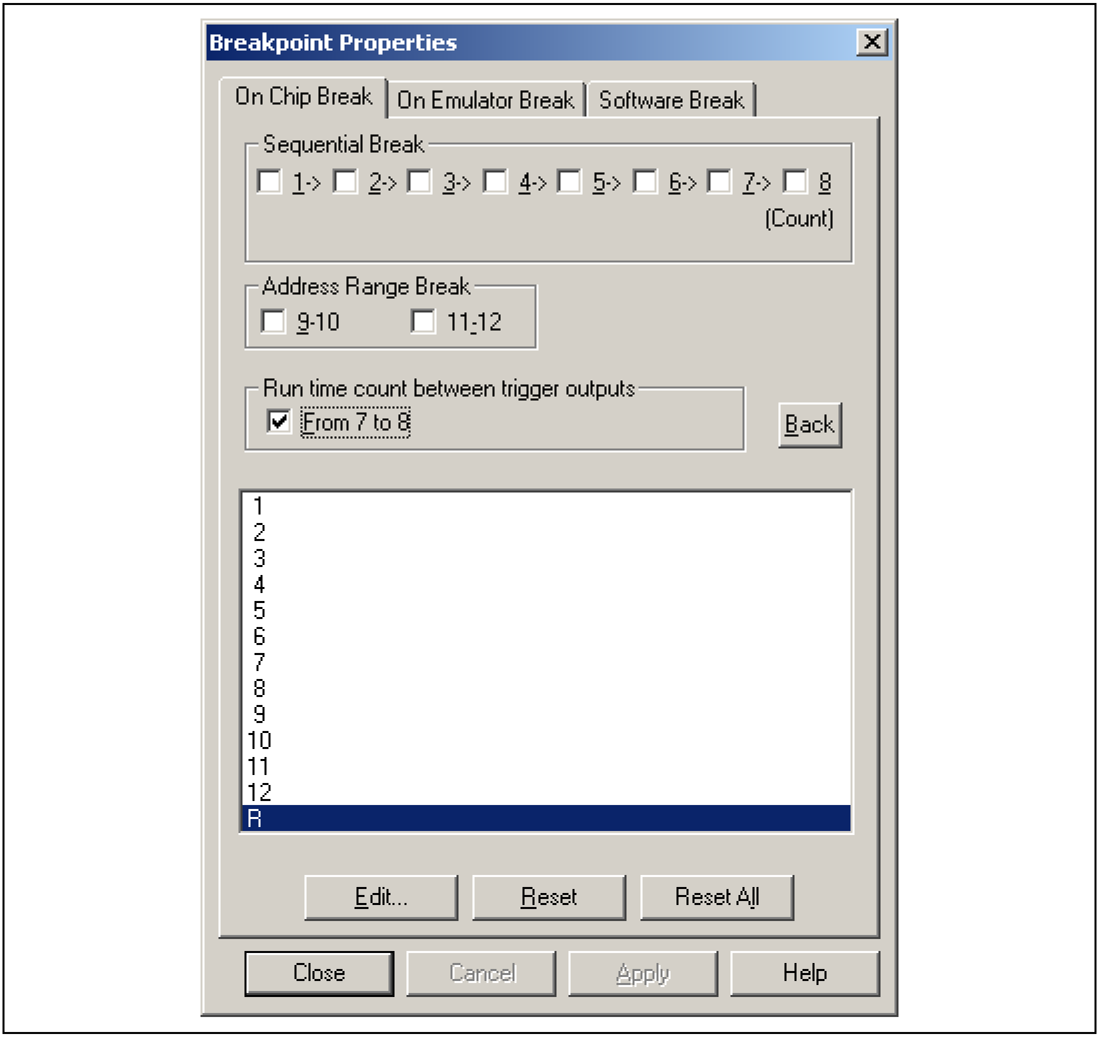



**Figure 5.57 [Breakpoint Properties] Dialog Box ([On Chip Break] Page)**

Select [From 7 to 8] in the [Point To Point Time Measurement] on the [Breakpoint Properties] dialog box.

Set the measurement start condition for break channel 7 from the [On Chip Break 7] dialog box, and the measurement end condition for break channel 8 from the [On Chip Break 8] dialog box. After setting, the execution time can be measured when the user program is executed.

The measured result of Point To Point Time Measurement is displayed on the [Platform] page in the [System Status] window.

Notes: 1. For the measurement time, the measurement counter used depending on the selected status of [Enable performance counter], which is set on the [General] page in the [Configuration] dialog box, is changed.
When [Enable performance counter] is selected and the counter for measurement has 24 bits, the user can select approximately 14 minutes (with a sampling interval during execution of 52 μs), 26 seconds (with a sampling interval during execution of 1.6 μs), or 0.33 seconds (with a sampling interval during execution of 20 ns).
When [Enable performance counter] is not selected and the counter for measurement has 40 bits, the user can select approximately 15,860 hours (with a sampling interval during execution of 52 μs), 488 hours (with a sampling interval during execution of 1.6 μs), or 6 hours (with a sampling interval during execution of 20 ns).

2. This function is implemented by using a trigger output of On Chip Break. Therefore, when this function is set, [Output Trigger] on the [Action] page of channel 7 or 8 cannot be selected by default.

## 5.9 Performance Analysis Function

Performance analysis applies functions of the emulator to the measurement of performance.

### 5.9.1 Measuring with E6000H Station Function

The emulator's performance analysis function is for measuring the efficiency of parts of a user program in terms of their execution times against the overall execution time, and for measuring the number of times a part of the user program is executed.

- Setting the Conditions for Performance Measurement

In the HDI, the user can set the execution efficiency and the execution count through a certain dialog box and display the measured results in the [Performance Analysis] window.

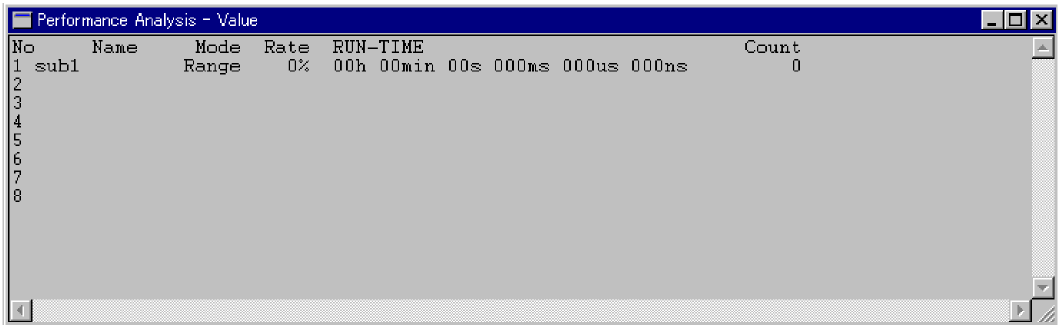



**Figure 5.58 [Performance Analysis] Window**

In the [Performance Analysis] window, you can assign any of eight independent channels. To set a condition, click [Conditions] in the pop-up menu of the [Performance Analysis] window to open the [Performance Analysis Condition] dialog box.

**Measurement Mode:** Table 5.47 shows the measurement modes.

**Table 5.47 Measurement Modes**

| Measurement Mode | Description |
|---|---|
| Time Of Specified Range Measurement | Measures the amount of time spent and the execution count during the last program run in the range between start address and end address. Start measurement when the program has been prefetched in the specified range between start address and end address. Then stop measurement when the program has been prefetched out of the specified range. Restart measurement when the program has been prefetched in the specified range. The execution count is counted whenever the program of the end address in the specified range is fetched. Measured result of the amount of time spent does not include the run time called in the specified range.<br><br>Set the conditions in the [Performance Analysis Condition 1 to 8] dialog box. |
| Start Point To End Point Measurement | Measures the amount of time spent and the execution count during the last program run in the range between start address and end address. Start measurement when the program has been prefetched in the specified range of the start address. Then stop measurement when the program of the end address has been prefetched. The execution count is counted whenever the program of the end address in the specified range is prefetched. Measured result of the amount of time spent includes the run time called in the specified range. For points 1 to 4, the maximum and minimum times in the specified range are measured.<br><br>Set the conditions in the [Performance Analysis Condition 1 to 8] dialog box. |
| Start Range To End Range Measurement | Start time measurement in the prefetch cycle of the start address range. Then stop measurement in the prefetch cycle of the end address range. The execution count is counted up whenever the end address range has passed.<br><br>Set the conditions in the [Performance Analysis Condition 1, 3, 5, 7] dialog box. |
| Access Count Of Specified Range Measurement | Measure the number of counts that accessed the address range of access area from the range specified with the start address and end address. The execution time in the range is measured by Time Of Specified Range Measurement.<br><br>Set the conditions in the [Performance Analysis Condition 1, 3, 5, 7] dialog box. |

**Table 5.47 Measurement Modes (cont)**

| Measurement Mode | Description |
|---|---|
| Called Count Of Specified Range Measurement | Measure the number of counts that called the call range from the range specified with the start address and end address. The execution time in the range is measured by using the Time Of Specified Range Measurement. Specify the start and end addresses in the specified subroutine for the call range.<br>Set the conditions in the [Performance Analysis Condition 1, 3, 5, 7] dialog box. |

Notes: 1. Execution of the subroutine is measured by using the value of address bus on the prefetch cycle. When the start or end address is set to an address of the instruction just after a branch instruction or the delay slot instruction, correct measurement is not possible.

2. In Access Count Of Specified Range Measurement, accessing by the DMA is not counted.

3. For [Time Resolution] of [Configuration Properties] selected from [Configure Platform], when the performance measurement condition is set, a value is fixed and cannot be changed.

- Example of Time Of Specified Range Measurement

  An example of Time Of Specified Range Measurement will be described below:

  Here, the start address is assumed to be H'1000 and the end address to be H'1FFF. When the user program is running, the emulator will measure the execution time of the user program in the address range from H'1000 to H'1FFF. When the user program jumps to address H'3000, measurement by the emulator will stop. When the user program returns from address H'3FFF, measurement by the emulator will start again.

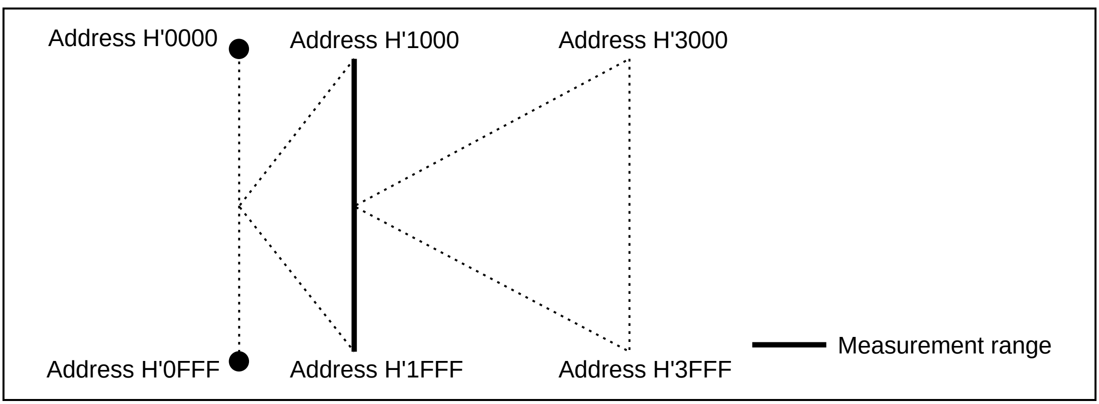



**Figure 5.59 Example of Time Of Specified Range Measurement**

- Example of Start Point To End Point Measurement

  In Start Point To End Point Measurement, the emulator starts to measure the execution time after it has passed the start address and continues to measure the time until it reaches the end address.

  An example of the use of Start Point To End Point Measurement is described below:

  Here, the start address is assumed to be H'1000 and the end address to be H'1FFF. When the user program is running, the emulator will start to measure the execution time of the user program from the start address (H'1000) until the user program reaches the end address (H'1FFF). When the emulator starts to measure the execution time, it will continue to measure until the user program reaches address H'1FFF or until user program emulation breaks. Therefore, the emulator will continue to measure the execution time of the user program after execution of the user program has jumped to address H'3000.

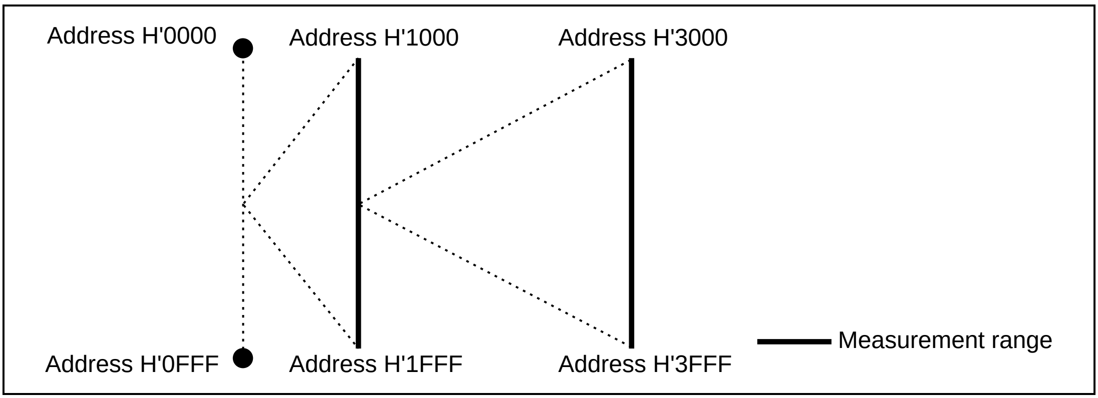



**Figure 5.60 Example of Start Point To End Point Measurement**

- Example of Start Range To End Range Measurement

  In Start Range To End Range Measurement, the emulator starts to measure the execution time of the user program in the start address range specified by the user, and continues to measure the execution time of the user program until the user program reaches the end address range.

  An example of the use of Start Range To End Range Measurement is described below:

  Here, the start address range is assumed to be from H'1000 to H'13FF and the end address range to be from H'1C00 to H'1FFF. While the user program is running, the emulator will start to measure the execution time of the user program from the start address range (H'1000 to H'13FF) until the user program reaches the end address range (H'1C00 to H'1FFF). When the emulator starts to measure the execution time of the user program, it will not stop until the user program reaches the end address range (H'1C00 to H'1FFF) or a break occurs during emulation. Therefore, the emulator will continue to measure the execution time when the user program jumps to address H'3000.

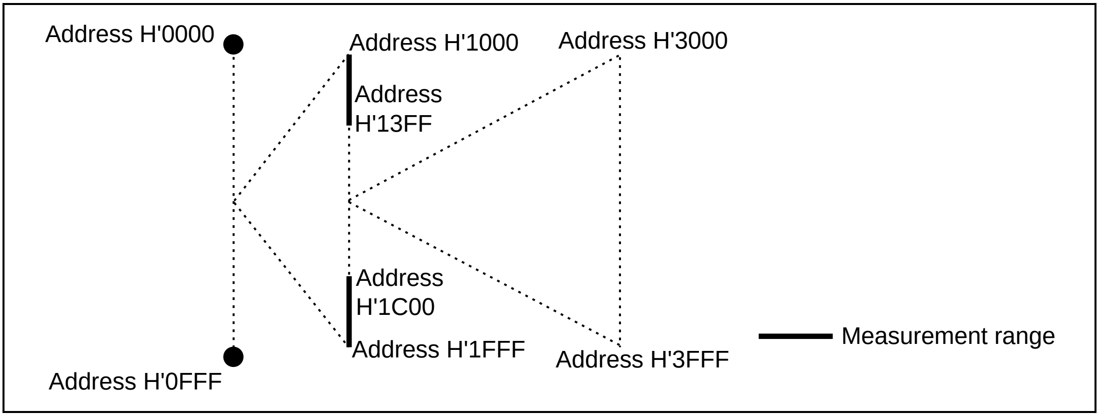



**Figure 5.61 Example of Start Range To End Range Measurement**

A list of subroutine measurement modes that can be set in the [Performance 1 to 8] dialog box is shown in table 5.48.

**Table 5.48 Modes that are Available in the [Performance 1 to 8] Dialog Boxes**

| | Channel Number for Performance Analysis | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Mode** | **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: Can be set.
×: Cannot be set.

Notes: 1. Two channels are necessary to specify the start and end address ranges for Start Range To End Range Measurement. Therefore, settings for Start Range To End Range Measurement can only be made on odd-numbered channels. In the Access Count Of Specified Range Measurement mode, two channels are required to specify the start and end addresses of the area accessed by the subroutine specified by its start and end addresses. Therefore, settings for the Access Count Of Specified Range Measurement mode can only be made on odd-numbered channels. In the Called Count Of Specified Range Measurement mode, two channels are required to specify the start and end addresses of the subroutine accessed by the first subroutine, itself specified by its start and end addresses. Therefore, settings for the Called Count Of Specified Range Measurement mode can only be made on odd-numbered channels.

2. In Start Point To End Point Measurement, the minimum and maximum time is measured only for channel numbers 1 to 4.

**Performance Measurement Time:** For the measurement time, the measurement counter used depending on the selected status of [Enable performance counter], which is set on the [General] page in the [Configuration] dialog box, is changed.
When [Enable performance counter] is selected and the counter for measurement has 40 bits, you can select approximately 15,860 hours (with a sampling interval during execution of 52 µs), 488 hours (with a sampling interval during execution of 1.6 µs), or 6 hours (with a sampling interval during execution of 20 ns).
When [Enable performance counter] is not selected and the counter for measurement has 24 bits, the user can select approximately 14 minutes (with a sampling interval during execution of 52 µs), 26 seconds (with a sampling interval during execution of 1.6 µs), or 0.33 seconds (with a sampling interval during execution of 20 ns).

Notes: 1. For measurement, the following two differences must be considered:
(1) ±1-resolution (when the resolution is 20 ns, an error is ±20 ns)
The ±1-resolution will be generated when the user program execution is started or stopped (break) and the start and end conditions are satisfied.
(2) Frequency stability of crystal oscillator for measurement: ±0.01%

2. Set the time interval from the satisfaction of end condition to the satisfaction of the next start condition as larger than one resolution in each measurement mode. If this interval is less than one resolution, the time from the satisfaction of end condition to the satisfaction of the next start condition is measured.

3. In Time Of Specified Range Measurement, the end condition is satisfied when the address is outside of the specified range. In Start Point To End Point Measurement and Start Range To End Range Measurement, the end condition is satisfied when the specified condition has been satisfied. If the same address is specified for them, the measured result of Time Of Specified Range measurement will be larger than that of Start Point To End Point Measurement and Start Range To End Range Measurement.

4. The address bus value of the prefetch cycle is used for each measurement. The measurement is not correct when the end address condition is specified for the address adjacent to the next address of the branch or delayed slot instruction. When the bus trace is displayed, check the operation after the cycle where the branch instruction has been prefetched and set the address for the prefetch cycle that is not executed at branch as the end address.

**How to Set the Conditions for Measuring the Performance:** To set the conditions for measuring the performance, use the [Performance Analysis Conditions] dialog box that can be displayed by clicking [Conditions] in the pop-up menu on the [Performance Analysis] window. An example of the use of Performance Analysis 1 is described below:

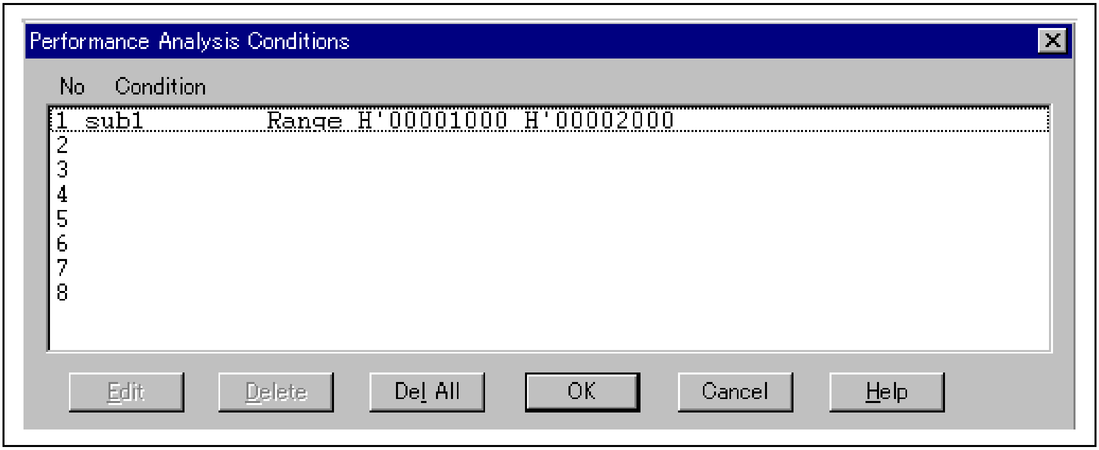



**Figure 5.62 [Performance Analysis Conditions] Dialog Box**

**Table 5.49 [Performance Analysis Conditions] Dialog Box Options**

| Option | Description |
|---|---|
| [Condition] | Displays the execution time and the settings of the subroutines which execution count is measured. Empty is displayed when nothing is set.<br>1 (Performance Analysis 1 setting)<br>2 (Performance Analysis 2 setting)<br>3 (Performance Analysis 3 setting)<br>4 (Performance Analysis 4 setting)<br>5 (Performance Analysis 5 setting)<br>6 (Performance Analysis 6 setting)<br>7 (Performance Analysis 7 setting)<br>8 (Performance Analysis 8 setting) |
| [Edit...] | Modifies the Performance Analysis settings that are set in the [Condition] list box. Clicking this button displays the [Performance *n*] dialog box. (*n*: number) |
| [Delete] | Cancels the Performance Analysis settings that are set in the [Condition] list box. |
| [Del All] | Cancels settings of all the subroutines set in the [Condition] list box. |

Select 1 in the [Condition] list then press the [Edit…] button.

The following explains the [Performance Analysis Properties] dialog box.

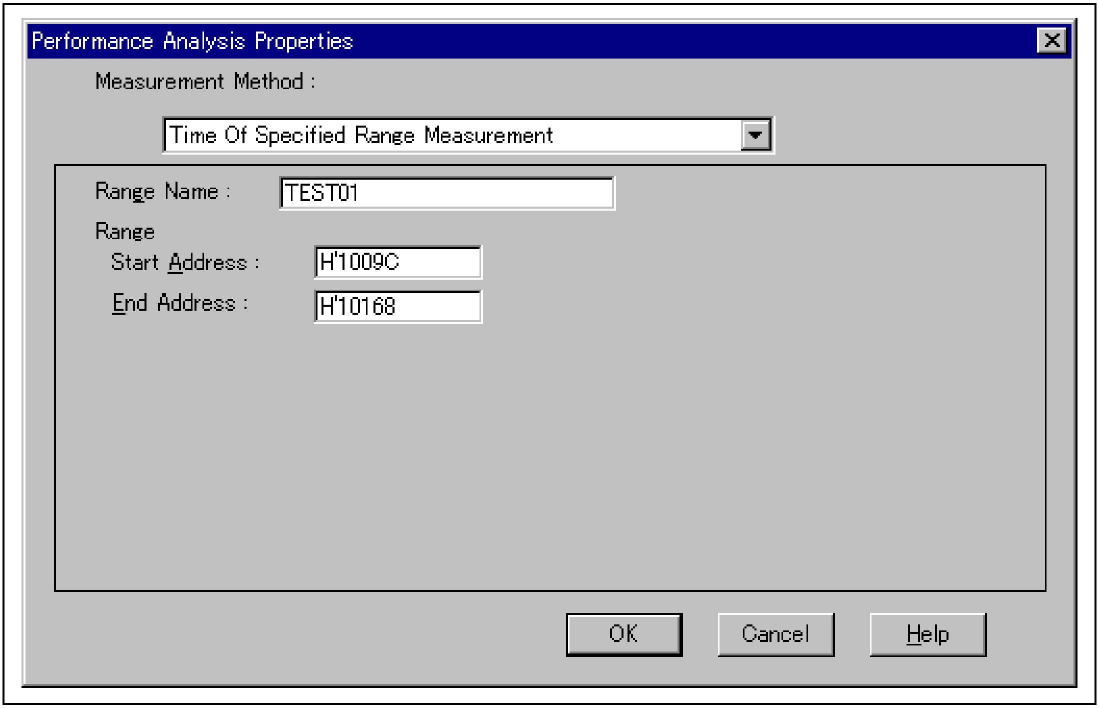



**Figure 5.63 [Performance Analysis Properties] Dialog Box**

The conditions on execution efficiency are set in this dialog box. These conditions have five modes (Measurement Methods). Set the address detection mode to the prefetch address detection mode in the Access Count Of Specified Range Measurement mode, and to the PC address detection mode in other modes.

Table 5.50 shows a list of five modes (Measurement Method).

**Table 5.50 Setting Value for Measurement Method**

| Mode | Description | | |
|---|---|---|---|
| Time Of Specified Range Measurement | Range Name | Specifies the range name. | |
| | Range | Specifies the range for Time Of Specified Range Measurement. | |
| | | Start Address | Specifies the start address. |
| | | End Address | Specifies the end address. |
| Start Point To End Point Measurement | Range Name | Specifies the range name. | |
| | Point | Specifies the range for Start Point To End Point Measurement. | |
| | | Start Address | Specifies the start address. |
| | | End Address | Specifies the end address. |
| | Time Out Action | Specifies the operation when a timeout for the execution time measurement or counting out occurs. | |
| | | Disable | Disables setting of a timeout value for the execution time measurement or a counted-out value. |
| | | Break | Stops user program when a timeout for the execution time measurement or counting out occurs. |
| | | Trace Stop | Stops trace acquisition when a timeout for the execution time measurement or counting out occurs. |
| | Time Out | To specify the minimum unit of the measurement time as 52 us, 1.6 us, and 20 ns, input h (hour), min (minute), s (second), ms (millisecond), us (microsecond), and ns (nanosecond); e.g., 1h 2min 3s 123ms 456us 789ns. A target clock is specified with six digits in hexadecimal; e.g., 123456. A break occurs when each measurement value between the start and end addresses exceeds the timeout value (not the total time). | |
| | Count | Specifies a counted-up value for measuring the execution count. | |

**Table 5.50 Setting Value for Measurement Method (cont)**

| Mode | Description | | |
|---|---|---|---|
| Start Range To End Range Measurement | Range Name | Specifies the range name. | |
| | Start Range | Specifies the start range for Start Range To End Range Measurement. | |
| | | Start Address | Specifies the start address. |
| | | End Address | Specifies the end address. |
| | End Range | Specifies the end range for Start Range To End Range Measurement. | |
| | | Start Address | Specifies the start address. |
| | | End Address | Specifies the end address. |
| Access Count Of Specified Range Measurement | Range Name | Specifies the range name. | |
| | Range | Specifies the start range for Access Count Of Specified Range Measurement. | |
| | | Start Address | Specifies the start address. |
| | | End Address | Specifies the end address. |
| | Access Area Range | Specifies the end range for Access Count Of Specified Range Measurement. | |
| | | Start Address | Specifies the start address. |
| | | End Address | Specifies the end address. |
| | Access Type | Selects the bus cycle in the access area. | |
| | | Don’t Care | All accesses |
| | | DMAC | DMAC cycle |
| | | CPU Prefetch | CPU prefetch cycle |
| | | CPU Data | CPU data cycle |
| Called Count Of Specified Range Measurement | Range Name | Specifies the range name. | |
| | Range | Specifies the start range for Called Count Of Specified Range Measurement. | |
| | | Start Address | Specifies the start address. |
| | | End Address | Specifies the end address. |
| | Call Range | Specifies the end range for Called Count Of Specified Range Measurement. | |
| | | Start Address | Specifies the start address. |
| | | End Address | Specifies the end address. |

**Displaying the Results of Performance Measurement:** The results of performance measurement are displayed in the [Performance Analysis] window. To display the results, click the right-hand mouse button with the cursor in the [Performance Analysis] window then select a menu item from the pop-up menu. The menu items are shown in table 5.51.

**Table 5.51 Menu Items for Displaying the Results**

| Menu Item | Description |
|---|---|
| [Value] | Displays the execution time and count (default). |
| [Graph] | Displays the proportions of execution time to the whole period of execution on a graph. |

**[Value]:** Select [Value] to display the executed results with a numeric as shown in figure 5.64.

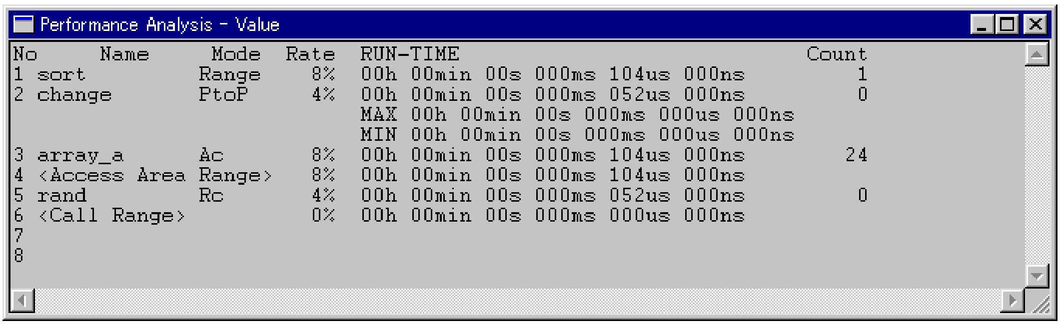



**Figure 5.64 [Performance Analysis] Window ([Value] Selected)**

**Table 5.52 Display Format of Results When [Value] is Selected**

| Item | Contents |
|---|---|
| [NO] | Channel number. |
| [NAME] | Subroutine name. |
| [MODE] | Measurement mode.<br>Range: Time Of Specified Range Measurement<br>PtoP: Start Point To End Point Measurement<br>RtoR: Start Range To End Range Measurement<br>Ac: Access Count Of Specified Range Measurement<br>Sc: Called Count Of Specified Range Measurement |
| [Rate] | Proportions of execution times. (numerical) |
| [RUN-TIME] | Total period of execution (h: hours, min: minutes, s: seconds, us: microseconds, ns: nanoseconds). |
| [Count] | Access count. |

**[Graph]:** Select [Graph] to display the proportion of total execution time of the user program, as shown in figure 5.65.

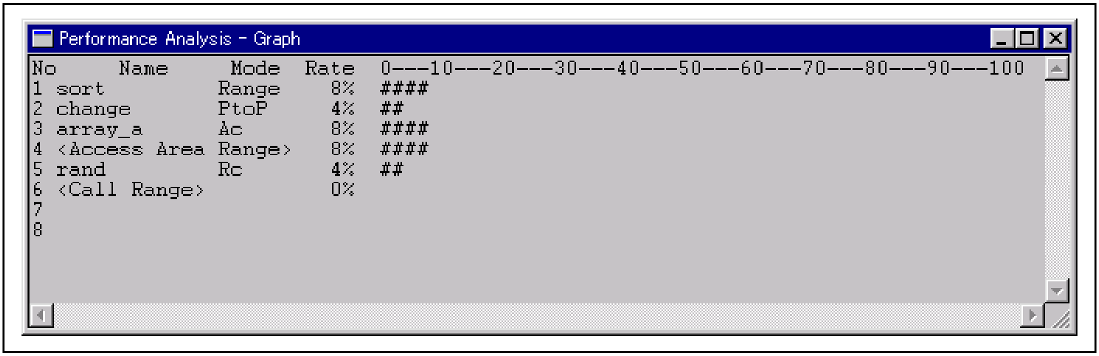



**Figure 5.65 [Performance Analysis] Window (Results of Execution Time Ratios)**

**Table 5.53 Display Format of Results When [Graph] is Selected**

| Item | Contents |
|---|---|
| [NO] | Channel number. |
| [NAME] | Subroutine name. |
| [MODE] | Measurement mode.<br>Range: Time Of Specified Range Measurement<br>PtoP: Start Point To End Point Measurement<br>RtoR: Start Range To End Range Measurement<br>Ac: Access Count Of Specified Range Measurement<br>Sc: Called Count Of Specified Range Measurement |
| [RATE] | Proportions of execution time (as numerical values and on a graph). |

**Initializing Performance Measurement Information:** To initialize performance measurement information, select [Refresh] from the pop-up menu of the [Performance Analysis] window. Before the initialization, the emulator will display a message box to ask you if you want to initialize the information. To initialize the information, press the [Yes] button.

### 5.9.2 Profiling Function

**Profiling Data Measurement Function:** This function is used to collect and measure the elapsed time in each function. Program execution and suspension are repeated and a difference will be included.

In the HDI, three windows shown in table 5.54 are provided. For details, refer to the Hitachi Debugging Interface User's Manual on the CD-R.

**Table 5.54 Window Provided by the Profiling Data Measurement Function**

| Window | Contents |
|---|---|
| [Profile-List] | Displays the address, size, the number of calls, and execution time of a function by selecting [Profile-List] from the [View] menu. |
| [Profile-Tree] | Displays the relationship of calling functions with a tree structure. In addition, the address, size, stack size, the number of calls, and execution time of a function are displayed. The value of the stack size, the number of calls, and execution time of a function are included in the actual path of calling a function. These items are displayed by selecting [Profile-Tree] from the [View] menu. |
| [Profile-Chart] | Displays the relationship of calling functions focusing a certain function.<br><br>This window opens from the [Profile-List] or [Profile-Tree] window. When a certain function is selected and the right-hand mouse button is clicked in the [Profile-List] or [Profile-Tree] window, a pop-up menu is displayed. When [View Profile-Chart] is selected from the pop-up menu, the relationship of calling functions, selected on the [Profile-Chart] window, is displayed. The focused function is displayed at center, the calling source function at left, and calling destination function at right. The value by the side of the calling source and calling destination functions are the number of calls. |

- Measuring the Profiling Data
  - Downloading the load module

    Download the ELF/DWARF2-type load module.
  - Switching the profiling data measurement function

    To enable collecting the profiling data, select [Enable Profiler] from the pop-up menu in the window (any shown in table 5.54) and click the left of a text. To disable collecting the profiling data, select [Enable Profiler] again and remove the mark at the left of the text.
  - [Profile-List] window

    The address ([Address] cell), size ([Size] cell), the number of calls ([Times] cell), and execution time of a function are displayed. The following shows an example after the user program has been executed.

Profile-List

| Function/Variable | Address | Size | Times | Run Time |
|---|---|---|---|---|
| _main | H'00001034 | H'0000006C | 0 | 0 |
| _strpbrk | H'000013FC | H'00000078 | 0 | 0 |
| _strtok | H'00001318 | H'000000E4 | 0 | 0 |
| _srand | H'00001304 | H'00000014 | 0 | 0 |
| _rand | H'000012D0 | H'00000034 | 0 | 0 |
| _Dummy | H'000012C8 | H'00000006 | 0 | 0 |
| _INT_HUDI | H'000012C2 | H'00000006 | 0 | 0 |
| _INT_FPU | H'000012BC | H'00000006 | 0 | 0 |
| _INT_User_Break | H'000012B6 | H'00000006 | 0 | 0 |
| _INT_NMI | H'000012B0 | H'00000006 | 0 | 0 |
| _INT_DMAC_Address | H'000012AA | H'00000006 | 0 | 0 |
| _INT_CPU_Address | H'000012A4 | H'00000006 | 0 | 0 |
| _INT_Illegal_slot | H'0000129E | H'00000006 | 0 | 0 |
| _INT_Illegal_code | H'00001298 | H'00000006 | 0 | 0 |
| __INITSCT | H'0000122C | H'0000006C | 0 | 0 |
| _sbrk | H'000011EC | H'00000040 | 0 | 0 |
| __INIT_OTHERLIB | H'000011CC | H'00000020 | 0 | 0 |
| __INITLIB | H'000011C0 | H'0000000C | 0 | 0 |
| _change | H'0000115E | H'00000062 | 0 | 0 |
| _sort | H'000010A0 | H'000000BE | 0 | 0 |

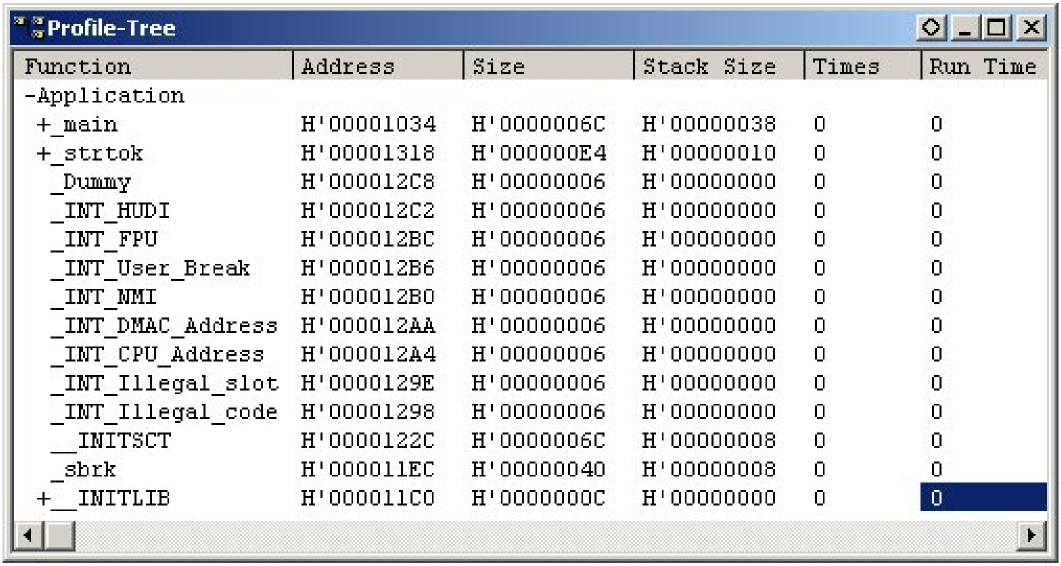
**Figure 5.66 [Profile-List] Window**

Notes: 1. If there is no information file for the amount of stack used (extension is '.sni') that is output from the optimizing linkage editor, only the function, that has been executed at measuring the profiling data, is displayed. For details, refer to the manual of the optimizing linkage editor.

2. The [Select Data] in the pop-up menu is invalid.

— [Profile-Tree] window

The address ([Address] cell), size ([Size] cell), stack size ([Stack Size] cell), the number of calls ([Times] cell), and execution time of a function are displayed. The following shows an example after the user program has been executed.


Profile-Tree

| Function | Address | Size | Stack Size | Times | Run Time |
|---|---|---|---|---|---|
| -Application | | | | | |
| +_main | H'00001034 | H'0000006C | H'00000038 | 0 | 0 |
| +_strtok | H'00001318 | H'000000E4 | H'00000010 | 0 | 0 |
| _Dummy | H'000012C8 | H'00000006 | H'00000000 | 0 | 0 |
| _INT_HUDI | H'000012C2 | H'00000006 | H'00000000 | 0 | 0 |
| _INT_FPU | H'000012BC | H'00000006 | H'00000000 | 0 | 0 |
| _INT_User_Break | H'000012B6 | H'00000006 | H'00000000 | 0 | 0 |
| _INT_NMI | H'000012B0 | H'00000006 | H'00000000 | 0 | 0 |
| _INT_DMAC_Address | H'000012AA | H'00000006 | H'00000000 | 0 | 0 |
| _INT_CPU_Address | H'000012A4 | H'00000006 | H'00000000 | 0 | 0 |
| _INT_Illegal_slot | H'0000129E | H'00000006 | H'00000000 | 0 | 0 |
| _INT_Illegal_code | H'00001298 | H'00000006 | H'00000000 | 0 | 0 |
| __INITSCT | H'0000122C | H'0000006C | H'00000008 | 0 | 0 |
| _sbrk | H'000011EC | H'00000040 | H'00000008 | 0 | 0 |
| +__INITLIB | H'000011C0 | H'0000000C | H'00000000 | 0 | 0 |

**Figure 5.67 [Profile-Tree] Window**

Note: The stack size differs from the actual value. It should be the target in the actual path of calling a function. If there is no information file for the amount of stack used (extension is '.sni') that is output from the optimizing linkage editor, the stack size is not displayed.

— [Profile-Chart] window

The relationship of calling functions, selected on the [Profile-List] or [Profile-Tree] window, is displayed. The focused function is displayed at center, the calling source function at left, and calling destination function at right. The value by the side of the calling source and calling destination functions are the number of calls.

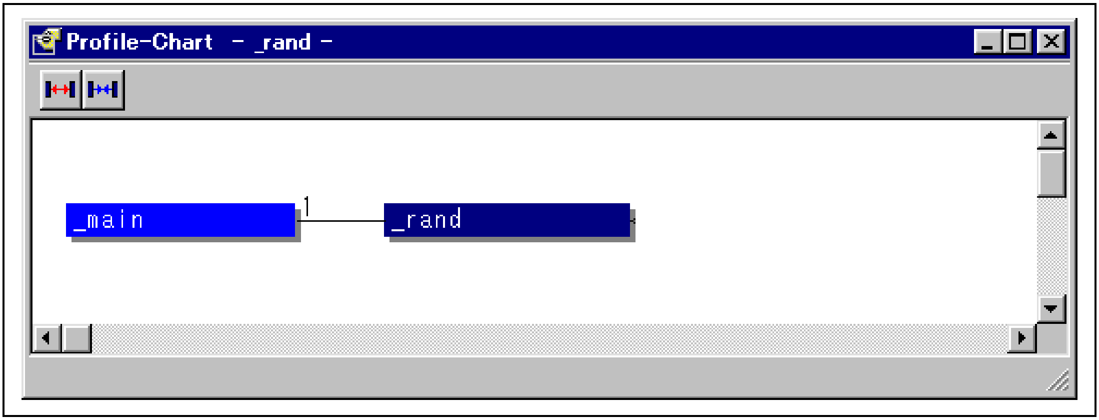



**Figure 5.68 [Profile-Chart] Window (after User Program Execution)**

— Setting the condition for starting or ending the profiling data measurement

The measurement period of the profiling data is from the start to end of the user program execution. The following shows the conditions for starting or ending the profiling data measurement.

**Table 5.55 Conditions for Starting or Ending the Profiling Data Measurement**

| Condition | Contents | Setting |
|---|---|---|
| Start | Current PC (program counter) | • [Go] from the [Run] menu ([Go] button on the toolbar)<br>• Specifies the start address in [Set PC To Cursor] from the [Run] menu |
| | Reset vector address | • [Reset Go] from the [Run] menu ([Reset Go] button on the toolbar) |
| End | On-emulator break | • Sets the [On Emulator Break] dialog box |
| | Stop address | • [Go To Cursor] from the [Run] menu ([Go To Cursor] button on the toolbar) |
| | Halt | • [Halt] from the [Run] menu ([Stop] button on the toolbar) |

Note: As the end condition, the user program is halted when the break instruction is executed.

— Limitation on the profiling function

When the profiling function is enabled, there are limitations shown in table 5.56.

**Table 5.56 Limitations on Enabled Profiling Function**

| Emulation State | Limitations |
|---|---|
| During user program execution | • Settings on software break, on-emulator break, and on-chip break are ignored.*<br>• The performance measurement condition is ignored.<br>• The trace acquisition condition is ignored.<br>• The setting of [When trace-buffer full] in the [Trace Acquisition Properties] dialog box ([Other] page) is ignored.<br>• Accessing to memory is disabled (display or modify in the [Memory] or [Disassemble] window).<br>• The displayed contents of the [Aum] window are not updated. |

Note: Do not use the profiling function for the standby mode in the user program.

- Differences

The profile function internally breaks user program execution, collects the measured data, and re-executes the user program. Since the function also counts when the measured item is generated at break or re-execution, a difference will be included in the measured profile value.

It takes a long time to start and end the user program execution. The user program execution times under the following environment are shown below. The measured value of this function should be the target.

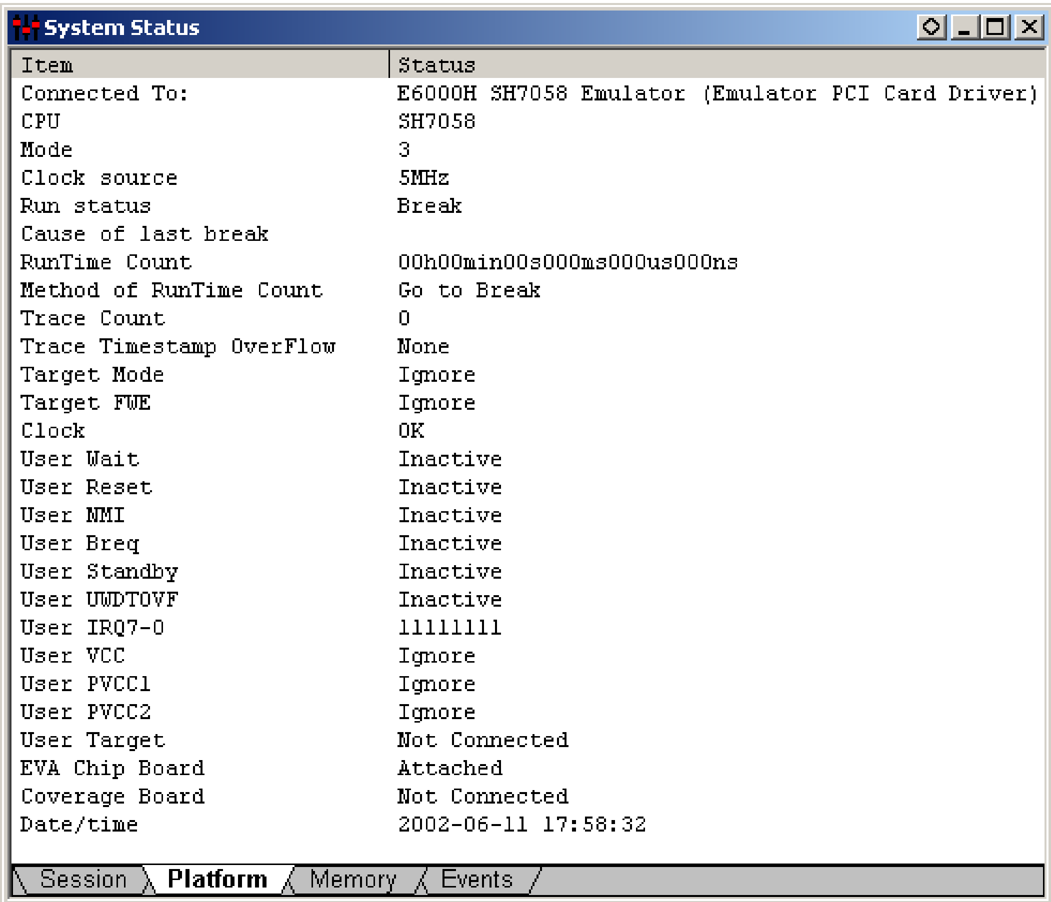
Environment:

Host computer: 400 MHz (Pentium® II)
Memory: 128 Mbytes
OS: Windows® 98
Execution program: 10,000 nested calls

- When the profile function is not used: 1 second or shorter
- When the profile function is used in the setting without including a callee function: 20 seconds
- When the profile function is used in the setting including a callee function: 400 seconds

## 5.10 Displaying Various Information

The [System Status] window displays a variety of information set by each dialog box. Open the [System Status] window by selecting [Status] from the [View] menu.


System Status

| Item | Status |
|---|---|
| Connected To: | E6000H SH7058 Emulator (Emulator PCI Card Driver) |
| CPU | SH7058 |
| Mode | 3 |
| Clock source | 5MHz |
| Run status | Break |
| Cause of last break | |
| RunTime Count | 00h00min00s000ms000us000ns |
| Method of RunTime Count | Go to Break |
| Trace Count | 0 |
| Trace Timestamp OverFlow | None |
| Target Mode | Ignore |
| Target FWE | Ignore |
| Clock | OK |
| User Wait | Inactive |
| User Reset | Inactive |
| User NMI | Inactive |
| User Breq | Inactive |
| User Standby | Inactive |
| User UWDTOVF | Inactive |
| User IRQ7-0 | 11111111 |
| User VCC | Ignore |
| User PVCC1 | Ignore |
| User PVCC2 | Ignore |
| User Target | Not Connected |
| EVA Chip Board | Attached |
| Coverage Board | Not Connected |
| Date/time | 2002-06-11 17:58:32 |



**Figure 5.69 [System Status] Window**

[System Status] window has the four sheets as shown in table 5.57.

**Table 5.57 [System Status] Window Configuration**

| Sheet Name | Description |
|---|---|
| [Session] | Contains such information on the current session as whether a debugging platform is connected and the names of loaded files. |
| [Platform] | Includes the status information on the Emulator, such as the name of the connected emulator. |
| [Memory] | Includes the information related to the current state of memory, such as memory mapping and the memory areas to be used by loaded object files. |
| [Events] | Contains information on the usage of resources on breakpoints. |

The [Session] sheet has the following contents.

**Table 5.58 [Session] Sheet Configuration**

| Item | Description |
|---|---|
| [Target System] | Indicates whether the emulator is connected. |
| [Session Name] | A session file name. |
| [Program Name] | The name of a load module that is loaded. |

The [Platform] sheet has the following contents.

**Table 5.59 [Platform] Sheet Configuration**

| Item | Description |
|---|---|
| [Connected To:] | The name of an emulator that is connected. |
| [CPU] | The type of target MCU. |
| [Mode] | The mode that is selected. |
| [Clock source] | The clock that is selected. |
| [Run status] | Whether or not the user program is being executed. |
| [Cause of last break] | The cause of termination by a break. |
| [Run Time Count] | The results of measuring the program execution time. |
| [Trace Count] | The number of all traced results. |
| [Trace Timestamp OverFlow] | Whether or not the timestamp of the trace overflows. |
| [Target Mode] | The mode input from the user system. |
| [Target FWE] | Whether or not there is the FWE signal input from the user system. |
| [Clock] | Whether or not there is the clock to be input. |
| [User Wait] | The state of the WAIT pin. |
| [User Reset] | The state of the RES pin. |
| [User NMI] | The state of the NMI pin. |
| [User Breq] | The state of the BREQ pin. |
| [User Standby] | The state of the HSTBY pin. |
| [User UWDTOVF] | The state of the WDTOVF pin. |
| [User IRQ7-0] | The state of the IRQ7, IRQ6, IRQ5, IRQ4, IRQ3, IRQ2, IRQ1, and IRQ0 pins. |
| [User Vcc] | Whether or not the user Vcc is 2.6 V or more when the target system is connected. |
| [User PVcc1] | Whether or not the user PVcc1 voltage satisfies the value specified with User Vcc Threshold in the [Configure Platform] dialog box when the target system is connected. |
| [User PVcc2] | Whether or not the user PVcc2 is 4.2 V or more when the target system is connected. |
| [User Target] | Whether or not the user target system has been connected at activation. |
| [EV Chip Board] | Whether or not the evaluation chip board has been connected at activation. |
| [Coverage Board] | Whether or not the optional coverage board has been connected at activation. |
| [Date/time] | Current date and time. |

The [Memory] sheet has the following contents.

**Table 5.60 [Memory] Sheet Configuration**

| Item | Description |
|---|---|
| [Target Device Configuration] | The memory-map status of the on-chip module (INTERNAL I/O). |
| [System Memory Resources] | The remaining capacity of the emulation memory. |
| [Load Memory Areas] | The address range of the load module that was loaded. |

The [Events] sheet has the following content.

**Table 5.61 [Events] Sheet Configuration**

| Item | Description |
|---|---|
| [Resources] | The number of effective breakpoints that have been set. |

## 5.11 Trigger Output

During execution of the user program, the emulator outputs a low-level pulse from the trigger-output probe under the following two conditions.

- Trace condition satisfaction
- Hardware break condition satisfaction

Using this pulse as an oscilloscope's trigger input signal makes it easy to adjust the user system hardware. For example, the waveform produced when the user program goes to a specified point can be viewed.

When the trigger output is specified by using of the [User Signals] option of the [General] page in the [Configuration] dialog box, a low-level pulse is output for two bus cycles from the trigger output pin after 10 bus cycles in which hardware break and hardware break conditions were satisfied during emulation. The pulse is output until the end of the subsequent bus cycle. If the conditions are satisfied over consecutive bus cycles, the trigger output remains low. When the on-chip ROM/RAM is accessed, the timing of pulse output will be as shown in figure 5.70.

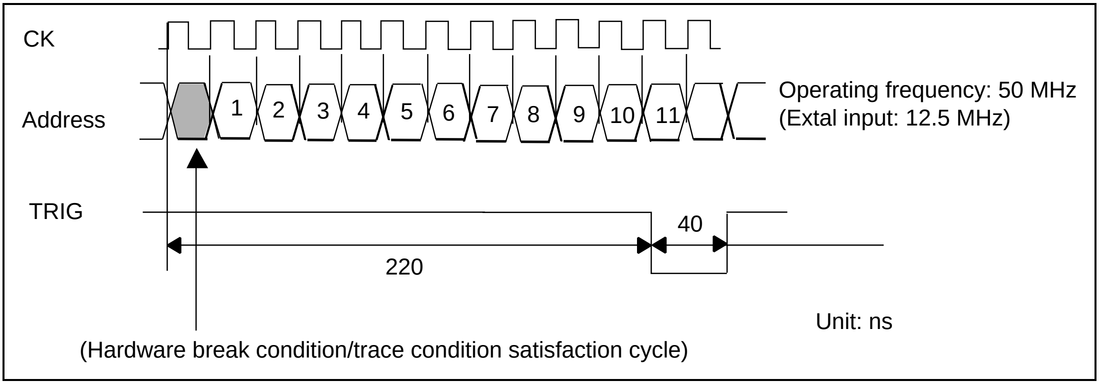
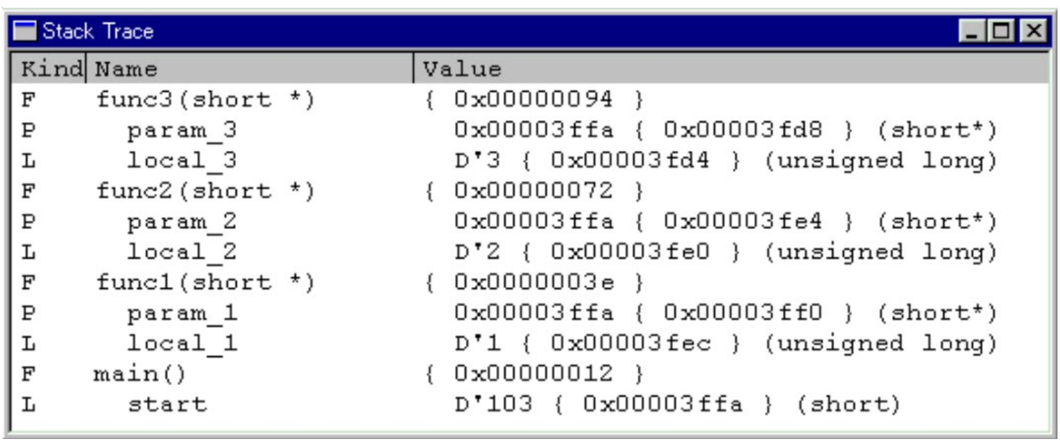



**Figure 5.70 Pulse Output Timing**

Note: No pulse is output from the trigger-output probe when a software break condition is satisfied. In addition, the timing of the pulse output and the pulse width differ according to the condition.

## 5.12 Stack Trace Function

The stack trace function displays the history of function calls. In the stack trace function, contents of memory are displayed in the [Stack Trace] window. The [Stack Trace] window can be displayed by selecting [Stack Trace] from the [View] menu.


Stack Trace

| Kind | Name | Value |
|---|---|---|
| F | func3(short *) | { 0x00000094 } |
| P | param_3 | 0x00003ffa { 0x00003fd8 } (short*) |
| L | local_3 | D'3 { 0x00003fd4 } (unsigned long) |
| F | func2(short *) | { 0x00000072 } |
| P | param_2 | 0x00003ffa { 0x00003fe4 } (short*) |
| L | local_2 | D'2 { 0x00003fe0 } (unsigned long) |
| F | func1(short *) | { 0x0000003e } |
| P | param_1 | 0x00003ffa { 0x00003ff0 } (short*) |
| L | local_1 | D'1 { 0x00003fec } (unsigned long) |
| F | main() | { 0x00000012 } |
| L | start | D'103 { 0x00003ffa } (short) |

**Figure 5.71 [Stack Trace] Window**

**Table 5.62 [Stack Trace] Window Options**

| Option | Description |
|---|---|
| [Kind] | Symbol type<br>F: Function, P: Function parameter, L: Local variable |
| [Name] | Symbol name |
| [Value] | Symbol value, address, and type |

Click the right-hand mouse button with the cursor in the [Stack Trace] window. The pop-up menu will be displayed. The menu includes the following options.

**Table 5.63 Options in the Pop-up Menu**

| Option | Description |
|---|---|
| [Copy] | Copies the highlighted text to the Windows® clipboard so that the text can be pasted to other applications. |
| [Go to Source] | Displays the source code in the program that corresponds to the selected function in the [Source] window. |
| [View Setting...] | Opens the [Stack Trace Setting] dialog box and sets the display format of the [Stack Trace] window. |

The format of the [Stack Trace Setting] dialog box is displayed in figure 5.72.

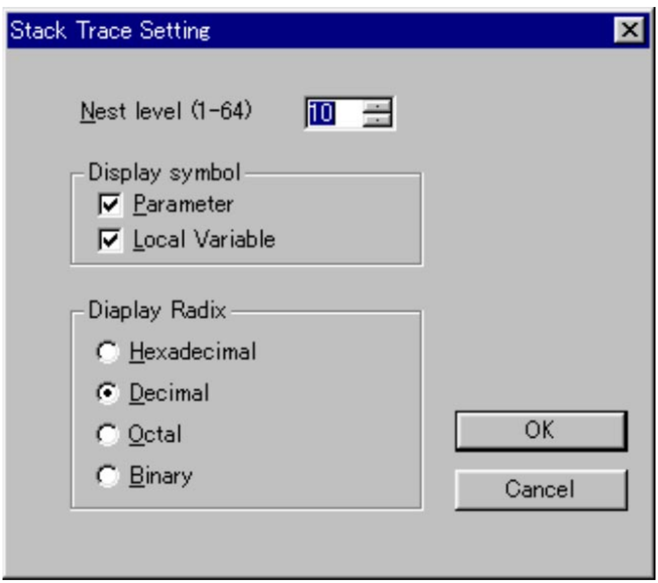



**Figure 5.72 [Stack Trace Setting] Dialog Box**

**Table 5.64 [Stack Trace Setting] Dialog Box Options**

| Option | Description |
|---|---|
| [Nest level] | Specifies the number of function call nestings for display in the [Stack Trace] window. Max: 64 |
| [Display symbol] | Specifies symbols other than functions for display. Specifies whether parameters and local variables will be displayed. |
| [Display Radix] | Specifies the radix for the display in the [Stack Trace] window. |

To refer to the online help system, press the [F1] key after moving the cursor to the [Stack Trace] window.

## 5.13 Displaying and Updating the Contents of Memory

### 5.13.1 Displaying and Updating the Contents of Memory during Execution

The emulator accesses memory in three ways to display and update the contents of memory during user program execution.

**Table 5.65 Access Types for Displaying and Modifying Contents of Memory**

| Access Type | Description | Stop Time | Display | Modification |
|---|---|---|---|---|
| Parallel access | Temporarily halts the execution of the user program | Short | Enabled | Enabled |
| Auto update Memory | Automatically updates the display of the contents of memory without stopping the execution of the user program | None | Enabled | Disabled |

Note: For details on stopped periods and other notes, refer to appendix E.7, Displaying and Modifying the Contents of Memory.

The respective access types have the following characteristics.

**Table 5.66 Characteristics of Displaying and Modifying Contents of Memory**

| Access Type | Target Window/Command Line | Target Memory Area |
|---|---|---|
| Parallel access | Windows other than the [Auto update Memory] window that display memory contents<br>Commands to display or modify memory contents | Memory, on-chip I/O area, on-chip RAM, or on-chip ROM<br>Emulation memory |
| Auto update Memory | [Auto update Memory] window<br>No target command | Specified address range |

The memory refresh function of the HDI ([Memory] or·[Refresh] menu) updates the contents of windows (other than [Auto update Memory]) that display the contents of memory. Therefore, if the memory refresh function is used during execution of the user program, parallel access will occur if the [I/O Registers] window, or the [Memory] window that displays the contents of on-chip RAM is opened.

### 5.13.2 Overview of Auto-Update Memory Function

In an auto-update of memory, the display of the contents of the specified area of memory is refreshed at intervals of 500 ms while the user program is being executed. The function has the following features:

Automatically updates, with an interval of approximately 500 ms, the display of the specified memory contents while the user program is being executed.

A color may be applied to indicate the contents that have been modified since the last update of the display.

Allows the setting of a maximum of 8 points (auto-update memory items), each running for a maximum of 32 bytes.

Allows the display of memory contents in ASCII, bit, byte, word, longword, or single-precision floating point formats (with or without a sign, and in decimal or hexadecimal).

The following dialog boxes are used to set items for the auto-update memory function.

**Table 5.67 Dialog Boxes for Setting Auto-Update Memory Items**

| Dialog Box | Description |
|---|---|
| [Auto-update Memory -Add-] | Dialog box for registering items for the auto-update memory function. Select the address, format of display, and number of bytes of memory to be displayed. |
| [AUM – Target Details] | Dialog box for registering other settings for auto-update memory items. In particular, those conditions that depend on the target MCU are set here. |

In an auto-update of memory, the contents of the specified area of memory are displayed in the [AUM] window. The [AUM] window is displayed by selecting [Auto update Memory Window] from the [View] menu and registering the Auto update Memory item in the [Auto-update Memory -Add-] dialog box.

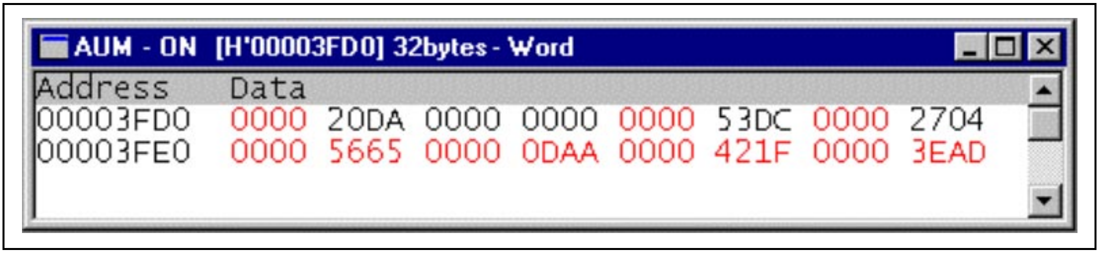



**Figure 5.73 [AUM] Window**

The [AUM] window displays the selected ranges of memory, and this display is refreshed while the user program is being executed. Up to 8 windows (8 points) can be displayed in the [AUM] window.

### 5.13.3 Setting Auto update Memory

**Points to Be Set And Byte Size to Acquire:** Up to 8 points can be set. Up to 32 bytes can be acquired from each point.

### 5.13.4 Displaying the Auto update Memory

**Display format:** The auto-update memory function allows the display of memory contents in ASCII, bit, byte, word, longword, single-precision floating point units, or double-precision floating point units (with or without signs, in decimal or hexadecimal).

Click the right-hand mouse button with the cursor in the [AUM] window. The pop-up menu will be displayed. Select [Edit…] from the menu to display the [Auto-update Memory -Edit-] dialog box and select [Format] from the [Auto-update Memory -Edit-] dialog box to change the display format, or select [Format] from the pop-up menu of the [AUM] window to change the display format.

**Display Color:** A color is applied to indicate the contents that have been modified since the last update of the display. Select [Type] from the [Color] group box in the [Auto-update Memory -Edit-] dialog box.

When [Change] is selected as [Type], modified data will be displayed in color. It is also possible to specify the foreground and background colors of the window. If [Gray] is selected, the modified data will be displayed in the color, with the unchanged data displayed in gray. When [Mayfly] is selected, the color is changed every time the contents of the window are updated.

To set the display format and display color, use the [Auto-update Memory -Edit-] dialog box. After setting the required items, click the [OK] button.

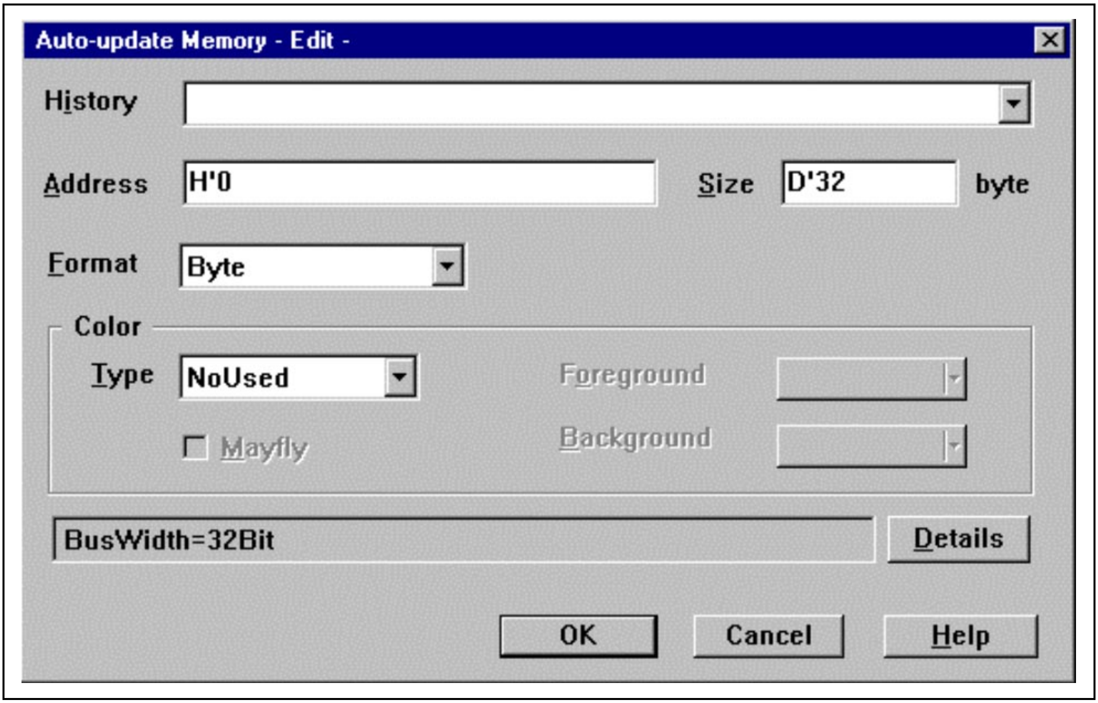



**Figure 5.74 [Auto-update Memory -Edit-] Dialog Box**

Note: In the situations listed below, the user program will not be executed in realtime.
When an auto-update memory item is set or modified while the user program is being executed.
When [Format] is modified while the user program is being executed.

## 5.14 Input Format

### 5.14.1 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 is shown below.

**Table 5.68 Address Mask Specification**

| No | Input Value | Mask Unit | Example | Masked Bits |
|---|---|---|---|---|
| 1 | Binary | 1 bit | B’00000111 | Masks bits 0 to 2 |
| 2 | Octal | 3 bits | O’000017 | Masks bits 0 to 3 |
| 3 | Hexadecimal | 4 bits | H'07FF | Masks bits 0 to 10 |

## 5.15 [Source] Window Expanded Function

### 5.15.1 Setting BP Column

In the HDI, PC (program counter) break can be set, cancelled, or displayed in the BP column of the [Source] window.

PC break can be set or cancelled by selecting a PC breakpoint with the left-hand mouse button and double-clicking in the BP column or by placing the cursor at the line where the breakpoint was set and pressing the F9 key.

This function is the same as for the BP columns in the [Disassembly] and [Labels] windows.

PC break, set on the window, firstly occupies channel 12 of On Chip Break, then subsequently uses channels 11, 10, …, 3, 2, and 1.

# Section 6 Command Line

## 6.1 List Format

This section explains the format for the command list in section 6.2. Some commands are explained on a single page. Some commands are explained over several pages.

### 6.1.1 Description

The entries are in the following format:

Command name (abbreviation)

[Command syntax and parameters]
Shows input format for each command.

[Description]
Describes the usage and function of each command.

[Examples]
Example of usage.

[Notes]
Notes on using the command. Some entries have no notes.

### 6.1.2 Format

Symbols used in the command format have the following meanings:

| | |
|---|---|
| < >: | Contents of < > are parameters. |
| [ ]: | Parameters enclosed by [ ] can be omitted. |
| < >=: | The parameter shown in the left < > can be expressed in the format in the right < >. |
| \|: | One or both can be selected, non-exclusively. |
| \| \|: | Either of two or one can be selected, exclusively. |

The parameters of each command are explained in the tables in section 6.2.

### 6.1.3 Parameter Type Input

1. Numerical parameters

   Numerical parameters must be supplied as binary, octal, decimal, or hexadecimal numbers, symbols, or expressions. A symbol can consist of up to 32 characters. Operators (e.g.: + and -) can be used to delimit expressions.
2. Keyword parameters

   The bold-faced characters in the tables for each command are the strings that are input as keyword parameters.

   Only the listed strings can be used. If a string that is not listed is input, an error will occur.
3. String parameters

   String parameters are used to input mask data, symbols, and file names. When using strings to mask data, specify H’ (hexadecimal), O’ (octal), or B’ (binary) at the head of the data as the radix. Specify 1 for the bits that correspond to the digits to be masked. When a bit is masked, it always satisfies the condition.

| No | Input Value | Mask Unit | Example | Masked Bits |
|---|---|---|---|---|
| 1 | Binary | 1 bit | B'00000111 | Masks bits 0 to 2 |
| 2 | Octal | 3 bits | O’000017 | Masks bits 0 to 3 |
| 3 | Hexadecimal | 4 bits | H'07FF | Masks bits 0 to 10 |

### 6.1.4 Examples

The examples show how to input the command. When output results, the output is also described.

## 6.2 List of Commands

The following is a list of HDI emulation commands. Sections in this manual are indicated in the second column. If there is no section number, the description is in the Hitachi Debugging Interface User’s Manual provided on the CD-R.

**Table 6.1 List of Commands**

| Command | Section | Abbreviation | Description |
|---|---|---|---|
| ! | — | — | Comment |
| ASSEMBLE | — | AS | Assembles user program. |
| ASSERT | — | — | Checks the condition. |
| ANALYSIS | 6.2.1 | AN | Displays the result of measurement for the rate of time spent. |
| ANALYSIS_RANGE*n* | 6.2.2 | AR*n* | Sets the conditions for measuring the rate of time spent. |
| ANALYSIS_RANGE_DELETE | 6.2.3 | AD*n* | Specifies or clears all the conditions for measuring the rate of time spent. |
| BREAKPOINT_ONCHIP | 6.2.4 | BC | Sets on-chip breakpoints. |
| BREAKPOINT_ONCHIP*n* | 6.2.4 | BC*n* | Sets on-chip breakpoints for each channel. |
| BREAKPOINT_ONCHIP _CLEAR | 6.2.4 | BCC | Clears on-chip breakpoints. |
| BREAKPOINT_ONCHIP _ENABLE | 6.2.4 | BCE | Enables or disables on-chip breakpoints that have been set. |
| BREAKPOINT_ONCHIP _RESET | 6.2.4 | BCR | Sets a reset point for on-chip breakpoints. |
| BREAKPOINT_ONEMULATOR | 6.2.5 | BE | Sets on-emulator breaks. |
| BREAKPOINT _ONEMULATOR*n* | 6.2.5 | BE*n* | Sets on-emulator breaks for each channel. |
| BREAKPOINT_ONEMULATOR _CLEAR | 6.2.5 | BEC | Clears the on-emulator breakpoints that have been set. |
| BREAKPOINT_ONEMULATOR _ENABLE | 6.2.5 | BEE | Enables or disables the on-emulator breakpoints that have been set. |
| BREAKPOINT_SOFTWARE | 6.2.6 | BS | Sets software breakpoints. |
| BREAKPOINT_SOFTWARE _CLEAR | 6.2.6 | BSC | Clears software breakpoints that have been set. |

**Table 6.1 List of Commands (cont)**

| Command | Section | Abbreviation | Description |
|---|---|---|---|
| BREAKPOINT_SOFTWARE _ENABLE | 6.2.6 | BSE | Enables or disables software breakpoints that have been set. |
| CLOCK | 6.2.7 | CK | Sets and displays the CLOCK signal for the MCU. |
| CONFIGURATION _PLATFORM | 6.2.8 | CP | Sets the operating conditions for the emulator. |
| DEVICE_TYPE | 6.2.9 | DE | Displays the type of a currently selected MCU. |
| DISASSEMBLE | — | DA | Disassembles user program and displays the result. |
| ERAM | 6.2.10 | EM | Sets the emulation memory or displays the contents. |
| ERASE | — | ER | Clears the contents of the Command Line window. |
| EVALUATE | — | EV | Calculates an expression. |
| FILE_LOAD | — | FL | Loads an object (program) file. |
| FILE_SAVE | — | FS | Saves the contents of memory as a file. |
| FILE_VERIFY | — | FV | Compares the contents of a file with memory. |
| FLASH_MEMORY | 6.2.11 | FM | Erases the contents of the flash memory. |
| GO | — | GO | Executes the user program. |
| GO_RESET | — | GR | Executes the user program from the reset vector. |
| GO_TILL | — | GT | Executes the user program up to a temporary breakpoint. |
| HALT | — | HA | Halts the user program. |
| HELP | — | HE | Displays help for command lines and commands. |
| INITIALISE | — | IN | Initializes a platform. |
| LOG | — | LO | Manipulates logging file. |
| MAP_DISPLAY | 6.2.12 | MA | Displays memory map information. |
| MEMORY_DISPLAY | — | MD | Displays memory contents. |
| MEMORY_EDIT | — | ME | Modifies memory contents. |

**Table 6.1 List of Commands (cont)**

| Command | Section | Abbreviation | Description |
|---|---|---|---|
| MEMORY_FILL | — | MF | Fills the memory with the specified data. |
| MODE | 6.2.13 | MO | Sets or displays the MCU mode. |
| MEMORY_MOVE | — | MV | Moves memory blocks. |
| MEMORY_TEST | — | MT | Tests memory blocks. |
| QUIT | — | QU | Terminates HDI. |
| RADIX | — | RA | Sets an input radix. |
| REFRESH | 6.2.14 | RF | Updates the memory information in HDI to reflect the latest state. |
| REGISTER_DISPLAY | — | RD | Displays MCU register values. |
| REGISTER_SET | — | RS | Sets MCU register values. |
| RESET | — | RE | Resets the MCU. |
| SLEEP | — | — | Delays command execution. |
| STEP | — | ST | Executes in steps (specified as instruction units or source line units). |
| STEP_OUT | — | SP | Executes until the function represented by the address at the program counter has finished. |
| STEP_OVER | — | SO | Steps over function. |
| STEP_RATE | — | SR | Sets and displays step execution rate. |
| SUBMIT | — | SU | Executes emulation command file. |
| SYMBOL_ADD | — | SA | Sets symbol. |
| SYMBOL_CLEAR | — | SC | Deletes symbol. |
| SYMBOL_LOAD | — | SL | Loads symbol information file. |
| SYMBOL_SAVE | — | SS | Saves symbol information file. |
| SYMBOL_VIEW | — | SV | Displays symbol. |
| TEST_EMULATOR | 6.2.15 | TE | Tests the on-chip flash memory area. |
| TIMER | 6.2.16 | TI | Displays or modifies the minimum for measurement of time. |
| TRACE_ACQUISITION | 6.2.17 | TA | Sets conditions for acquiring trace information. |
| TRACE_ACQUISITION*n* | 6.2.17 | TA*n* | Makes settings for each channel for acquiring trace information. |
| TRACE_ACQUISITION_CLEAR | 6.2.17 | TAC | Clears trace conditions that have been set. |

**Table 6.1 List of Commands (cont)**

| Command | Section | Abbreviation | Description |
|---|---|---|---|
| TRACE_COMPARE | 6.2.18 | TC | Compares the current trace data and the trace file that has been saved (refer to the trace_save section). |
| TRACE_SAVE | 6.2.19 | TV | Saves trace data as a file in the text format. |
| TRACE_SEARCH | 6.2.20 | TS | Sets conditions for displaying trace data. |
| USER_SIGNALS | 6.2.21 | US | Sets or displays user signals. |

### 6.2.1 ANALYSIS (AN)

[Command syntax and parameters]

- Display **an** [<v>]

| Parameter | Type | Description |
|---|---|---|
| <v> | Keyword | Displays the result that it has measured the rate of time spent during the last program run.<br>**Default**: Displays the rate of time spent during the last program run as percentages and histogram.<br>**v**: Displays the rate of time spent during the last program run as percentages, amount of time spent and the execution count. |

[Description]

Displays the result that it has measured of the rate of time spent during the last program run.

[Examples]

1. To display the result as percentages, amount of time spent and the execution count:

   ***an v***

2. To display the result as percentages and histogram:

   ***an***

### 6.2.2 ANALYSIS RANGE*n* (AR*n*)

Sets the conditions of the rate of time spent during the last program run, the address detection mode and the shortest measure clock. There are five measurement methods.

- The Time Of Specified Range Measurement
- The Start Point To End Point Measurement
- The Start Range To End Range Measurement
- The Access Count Of Specified Range Measurement
- The Called Count Of Specified Range Measurement

It is possible to set the condition on all points by The Time Of Specified Range Measurement / The Start Point To End Point Measurement. The condition on 1, 3, 5, 7 point can be set by the other methods.

The command syntax and parameters of each condition of the rate of time spent are listed below.

(1) The Time Of Specified Range Measurement
[Command syntax and parameters]

- Setting **AR***n* Range <Name> <StartAddress> <EndAddress>

| Parameter | Type | Description |
|---|---|---|
| *n* | Numeric | Sets a channel number from 1 to 8. |
| **Range** | Keyword | Makes settings for measurement of time in the specified range. |
| <Name> | Keyword | Specifies the name of the range. |
| <StartAddress> | Numeric | Specifies a start address. |
| <EndAddress> | Numeric | Specifies an end address. |

[Description]

This command measures time in the specified range.

[Example]

To set a condition of the rate of time spent in the specified range from the address H’100 to H’200:

***`ar1 Range RAM H’100 H’200`***

(2) The Start Point To End Point Measurement

[Command syntax and parameters]

- Setting **AR***n* **PtoP** <Name> <StartAddress> <EndAddress> [**action** <action>] [**time** <time>] [**count** <Count>]

| **Parameter** | **Type** | **Description** |
|---|---|---|
| *n* | Numeric | Sets a channel number from 1 to 8. |
| **PtoP** | Keyword | Makes settings for measurement of time between the specified addresses PtoP. |
| <Name> | Keyword | Specifies the name of the range. |
| <StartAddress> | Numeric | Specifies a start address. |
| <EndAddress> | Numeric | Specifies an end address. |
| [**action** <action>] | Keyword | Operation when a timeout or count-out occurs to finish measurement.<br>Only point 1 can be specified.<br><action> = <disable \|\| Break \|\| tracestop><br>**disable**: Does not specify operation at an occurrence of timeout or count-out.<br>**Break**: Breaks when a timeout or count-out occurs.<br>**tracestop**: Stops tracing when a timeout or count-out occurs. |
| [**time** <time>] | Numeric | The time-out value to finish measurement.<br>Specification: **time** <time><br><time> specifies the time in the following forms;<br>h: Hours<br>min: Minutes<br>s: Seconds<br>ms: Milliseconds<br>us: Microseconds<br>ns: Nanoseconds<br>(e.g. time 1h 2 min 3 s 123 ms 456 us 789 ns)<br>The target clock can be specified as hexadecimals in 10 digits (e.g. time H’123456789A). |
| [**count** <count>] | Numeric | The count-up value to measure.<br>Specification: **count** <count><br>The number can be specified in the range from H’1 to H’FFFF. |

[Description]

This command measures time between the specified addresses.

[Example]

To set a condition of the rate of time spent between the addresses H'200 and H'2FF with the timeout value as an hour, two minutes and three seconds:
***ar1 PtoP RAM 200 2FF time 1h 2min 3s 0ms 0us 0ns***

(3) The Start Range To End Range Measurement

[Command syntax and parameters]

- Setting **AR***n* RtoR <Name> <StartRange> <EndRange>

| **Parameter** | **Type** | **Description** |
|---|---|---|
| *n* | Numeric | Sets a channel number as 1, 3, 5, or 7. |
| **RtoR** | Keyword | Makes settings for measurement of time between the specified addresses RtoR. |
| <Name> | Keyword | Specifies the name of the range. |
| <StartRange> | Numeric | Specifies a start range.<br>Start address range <StartAddress>:<EndAddress><br><StartAddress>: Start address<br><EndAddress>: End address |
| <EndRange> | Numeric | Specifies an end range.<br>End address range <StartAddress>:<EndAddress><br><StartAddress>: Start address<br><EndAddress>: End address |

[Description]

This command measures time between the ranges of specified addresses.

[Example]

To set a condition of the rate of time spent between the start address range H'100 to H'1FF and the end address range H'200 to H'2FF:
***ar3 RtoR RAM H'100:H'1FF H'200:H'2FF***

(4) The Access Count Of Specified Range Measurement

[Command syntax and parameters]

- Setting **AR***n* AC <Name> <Range> <AccessRange> [bus<Type>]

| Parameter | Type | Description |
|---|---|---|
| *n* | Numeric | Sets a channel number as 1, 3, 5, or 7. |
| **AC** | Keyword | Makes settings for measurement of the number of accesses to AC areas. |
| <Name> | Keyword | Specifies the name of the range. |
| <StartRange> | Numeric | Specifies a start range.<br>Start address range <StartAddress>:<EndAddress><br><StartAddress>: Start address<br><EndAddress>: End address |
| <EndRange> | Numeric | Specifies an end range.<br>End address range <StartAddress>:<EndAddress><br><StartAddress>: Start address<br><EndAddress>: End address |
| [**bus** <Type>] | Keyword | Specifies a bus condition.<br>Specification: **bus** <Type><br><Type> = <**dmac** \|\| **prefetch** \|\| **data**><br>**dmac**: DMAC/DTC bus cycles<br>**prefetch**: CPU prefetch bus cycles<br>**data**: CPU Data bus cycles<br>**Default**: All the bus cycles |

[Description]

This command measures the number of accesses to AC areas.

[Example]

To set a condition of the rate of time spent for measuring the number of accesses in DMAC bus cycles with the address range H’100 to H’1FF and the accessed area H’200 to H’2FF:

***ar5 AC RAM H’100:H’1FF H’200:H’2FF bus dmac***

(5) The Called Count Of Specified Range Measurement

[Command syntax and parameters]

- Setting **AR***n* RC <Name> <Range> <CallRange>

| Parameter | Type | Description |
|---|---|---|
| *n* | Numeric | Sets a channel number as 1, 3, 5, or 7. |
| **RC** | Keyword | Makes settings for measurement of the number of calls in the specified range RC. |
| <Name> | Keyword | Specifies the name of the range. |
| <Range> | Numeric | Specifies the address range of a caller.<br>Address range of a caller <StartAddress>:<EndAddress><br><StartAddress>: Start address<br><EndAddress>: End address |
| <CallRange> | Numeric | Specifies the address range of a callee.<br>Address range of a callee <StartAddress>:<EndAddress><br><StartAddress>: Start address<br><EndAddress>: End address |

[Description]

This command measures the called count of the specified range.

[Example]

To set a condition of the rate of time spent for measuring the called count with the address range H’100 to H’11F and the range of addresses to be called H’120 to H’12F:

***ar7 RC RAM H’100:H’11F H’120:H’12F***

(6) Displaying the conditions for measuring the time spent

[Command syntax and parameters]

- Display **AR***n*

| Parameter | Type | Description |
|---|---|---|
| *n* | Numeric | Sets a channel number from 1 to 8.<br>At default, conditions set for all the points are displayed. |

### 6.2.3 ANALYSIS_RANGE_DELETE*n* (AD*n*)

[Command syntax and parameters]

- Cancellation **ad***n* (*n* = 1 to 8)
  **ad**

| Parameter | Type | Description |
|---|---|---|
| *n* | Keyword | Settings for cancellation by selecting points<br>**ad***n*: Cancels the condition of the point number *n.*<br>**ad**: Cancels all the conditions. |

[Description]

This command deletes the specified condition of the rate of time spent during the last program run or deletes all conditions.

### 6.2.4 On-Chip Break Commands (BC, BC*n*, BCC, BCE, BCR)

- Setting BREAKPOINT_ONCHIP (BC)
- Selection BREAKPOINT_ONCHIP*n* (BC*n*)
- Cancellation BREAKPOINT_ONCHIP_CLEAR (BCC)
- Enable or Disable
  BREAKPOINT_ONCHIP_ENABLE (BCE)
- Settings a reset point
  BREAKPOINT_ONCHIP_RESET (BCR)

[Command syntax and parameters]

- Setting **bc** [<option>…]
  <option> = <DISPLAY> | <SEQUENTIAL> | <ADDRESSRANGE> | <POINTTOPOINT>
- Selection **bc***n* (*n* = 1… 12) [<option…>]
  <option> = <ADDRESS> | <DATA> | <BUS> | <COUNT> | <ACTION>
- Cancellation **bcc** <channel>
- Enable or Disable
  **bce** <CHANNEL> <MODE>
- Setting a reset point
  **bcr** <ADDRESS>

Description of the bc command <option> (specify one or more conditions)

| Parameter | Type | Description |
|---|---|---|
| <DISPLAY> | Keyword | Displays the on-chip breakpoints that have been set.<br>Specification: **display** |
| <SEQUENTIAL> | Numeric | Selects the sequential level, <level>, from 1 to 8.<br>Specification: **sequential** <level><br><level> = <1 \| 2 \| 3 \| 4 \| 5 \| 6 \| 7 \| 8><br>When <levels> is omitted, the settings for the sequential breakpoints will be cleared. With 1 to 8 selected, sequential breakpoints can be set from the specified channel to channel 8. |
| <ADDRESSRANGE> | Numeric or characters | Selects a channel to be used for an address range condition.<br>Specification: **addressrange** <channels><br><channels> = <9-10 \| 11-12><br>When channels 9 and 10 are selected: **9-10**<br>When channels 11 and 12 are selected: **11-12**<br>When <channels> is omitted, the settings for the address range breakpoints will be cleared. |
| <POINTTOPOINT> | Keyword | Makes the settings for measuring P to P time.<br>Specification: **pointtopoint** <channels><br><channels> = 7-8<br>In settings for measurement of P to P time, use channels 7 and 8. When <channels> is omitted, the settings for measurement of P to P time will be cleared. |

Description of the bc*n* command <option> (Specify one or more conditions)

| Parameter | Type | Description |
|---|---|---|
| *n* | Numeric | Sets the channel number (*n* = 1 to 12). |

| Parameter | Description |
|---|---|
| <ADDRESS> | Specifies an address condition.<br>Specification: <ADDRESS> = **address** <address> [<address mask>]<br>Specification for a break at the specified address:<br>**address** <address><br><address>: Address value (numeric)<br>Specification for an address mask condition:<br>**address** <address> <address mask><br><address mask>: The digits to be masked for bits of <address> are set as 1 and otherwise 0. They can be displayed in hexadecimal (H’), octal (O’), or binary (B’). |
| <DATA> | Specifies a data condition.<br>Specification: <DATA> = data <data> [<data mask>] [<data size>]<br>Specification for a break with the specified data:<br>**data** <data><br><data>: Data value (numeric)<br>Specification for a data mask condition:<br>**data** <data> <data mask><br><data mask>: The digits to be masked for bits of <data> are set as 1 and otherwise 0. They can be displayed in hexadecimal (H’), octal (O’), or binary (B’).<br>Specification for a data size condition:<br>**data** <data> <data size> <location><br><data size>: Select one of the followings for <data size>;<br>For a break with byte-length data: **byte** <location><br>For a break with word-length data: **word** <location><br>For a break with longword-length data: **long**<br><location>: Specifies a location from a 4-byte boundary.<br>**4n**: Specifies data from a 4n-byte boundary.<br>**4n1**: Specifies data from a 4n-byte boundary + a byte.<br>**4n2**: Specifies data from a 4n-byte boundary + two bytes.<br>**4n3**: Specifies data from a 4n-byte boundary + three bytes.<br>When no specification is made, 4n will be selected. |

Description of the bc*n* command <option> (cont)

<table>
<tr><th>Parameter</th><th colspan="2">Description</th></tr>
<tr><td rowspan="4"><BUS></td><td colspan="2">Selects a bus condition.<br><BUS> = **bus** [cpu || dmac] [prefetch || data] [read || write]<br>When no specification is made, the condition will be cleared. A read condition must be selected when the type of an access to a bus is **prefetch**.</td></tr>
<tr><td>cpu || dmac</td><td>Sets a bus status condition.<br>For a break in execution cycles: **cpu**<br>For a break in DMA/AUD cycles: **dmac**</td></tr>
<tr><td>prefetch || data</td><td>Sets the type of an access to a bus as a break condition.<br>For a break after executing the instruction as the address value at a program fetch matches: **prefetch**<br>For a break after accessing data: **data**</td></tr>
<tr><td>read || write</td><td>Specifies a read/write condition. A read condition must be selected when the type of an access to a bus is **prefetch**.<br>For a break in read cycles: **read**<br>For a break in write cycles: **write**</td></tr>
<tr><td><COUNT></td><td colspan="2">Specifies the number of times that a condition matches.<br><COUNT> = **count** <count><br>**count**: Specifies the number of times that a condition matches.<br>For a break by the number of times that a condition matches:<br>**count** <count><br>Values H’1 to H’FFFF can be specified for **count**.<br>When <count> is omitted, the setting for the number of times that a condition matches will be cleared.</td></tr>
<tr><td><ACTION></td><td colspan="2">Specifies an operating condition after a break.<br><ACTION> = **action** [[break after || break before] | [trigger]]<br>**action**: Sets the operation at the satisfaction of a condition.<br>break [after || before]<br>For a break after the satisfaction of a condition: **action break after**<br>For a break before the satisfaction of a condition: **action break before**<br>For a trigger output at the satisfaction of a condition: **action trigger**<br>Only channel 8 can be specified when using a sequential break.</td></tr>
</table>

Description of the bcc command <option>

| Parameter | Description |
|---|---|
| <CHANNEL> | Specifies channels for which the settings are canceled.<br>Specification of channels for which the settings are canceled:<br><channels>: <1 \|\| 2 \|\| 3 \|\| 4 \|\| 5 \|\| 6 \|\| 7 \|\| 8 \|\| 9 \|\| \|\|10 \|\| 11 \|\| 12> \| <all><br>Specify a number or **all**. When **all** is selected, all the settings will be canceled. |

Description of the bce command <option> (specify one or more conditions)

| Parameter | Description |
|---|---|
| <CHANNEL> | Specifies the execution of a break for each channel.<br>Specification of **enable** and **disable** for a channel:<br><CHANNEL> = <channels> <**enable** \|\| **disable**><br><channels>: <1 \|\| 2 \|\| 3 \|\| 4 \|\| 5 \|\| 6 \|\| 7 \|\| 8 \|\| 9 \|\| 10 \|\| 11 \|\| 12><br>To enable a break at the specified address: **enable**<br>To disable a break at the specified address: **disable** |

Description of the bcr command <option>

| Parameter | Description |
|---|---|
| <ADDRESS> | Sets a reset point for the specified address.<br><ADDRESS> = **address** <address> |
| | Specifies an address condition.<br>Specification: <ADDRESS> = **address** <address><br>To set a reset point at the specified address: **address** <address><br><address>: Address value (numeric) |

[Description]

- Setting

  This command sets on-chip breakpoints. The program stops when the specified condition is satisfied. For details on conditions that can be specified, refer to the description of <option> in the Bc command section. Sequential breaks using on-chip breaks are also available. Up to eight levels, BREAKPOINT_ONCHIP_1 to BREAKPOINT_ONCHIP_8, of on-chip sequential breaks can be used.

- Selection

  This command makes settings for each channel of on-chip breaks. The program stops when the specified condition is satisfied. For details on conditions that can be specified, refer to the description of <option> in the bc*n* command section.

- Cancellation

  This command clears on-chip breakpoints that have been set.
- Enable or Disable

  This command enables or disables on-chip breakpoints that have been set.
- Setting a reset point

  This command sets a reset point for on-chip breakpoints. It also specifies an address of the reset point for on-chip breakpoints.

### 6.2.5 On-Emulator Break Commands (BE, BE*n*, BEC, BEE)

- Setting BREAKPOINT_ONEMULATOR (BE)
- Selection BREAKPOINT_ONEMULATOR*n* (BE*n*)
- Cancellation BREAKPOINT_ONEMULATOR_CLEAR (BEC)
- Enable or Disable
  BREAKPOINT_ONEMULATOR_ENABLE (BEE)

[Command syntax and parameters]

- Setting **be**
- Selection **be***n* (*n* = 1…4) [<option>…]
  <option> = <ADDRESS> | <DATA> | <BUS> | <COUNT> | <PROBE>
  | <INTERRUPT>
- Cancellation **bec** <CHANNEL>
- Enable or Disable
  **bee** <CHANNEL> <mode>

Description of the be*n* command <option> (specify one or more conditions)

| Parameter | Type | Description |
|---|---|---|
| *n* | Numeric | Sets the channel number (*n* = 1 to 4). |

| Parameter | Description |
|---|---|
| <ADDRESS> | Specifies an address condition.<br><ADDRESS> = **address** <address> \|\| <start address> <end address> [**outside**]<br>When no parameter is specified, the address condition will be cleared.<br>Specification for a break at the specified address:<br>**address** <address><br><address>: Address value (numeric)<br>Specification for a break in the specified range of addresses:<br>**address** <start address> <end address> [**outside**]<br><start address>:<br>Address value at the start of the specified range (numeric)<br><end address>:<br>Address value at the end of the specified range (numeric)<br>Select [**outside**] to specify those out of the specified range as a break condition. |

Description of the be*n* command <option> (cont)

| Parameter | Description |
|---|---|
| <DATA> | Specifies a data condition.<br><br>Specification: <DATA> = **data** <data> [<data mask>]<br>[<data size> <location>] [**not**]<br><br>When no parameter is specified, the data condition will be cleared.<br><br>Specification for a break with the specified data:<br>**data** <data><br><data>: Data value (numeric)<br><br>Specification for a data mask condition:<br>**data** <data> <data mask><br><data mask>: The digits to be masked for bits of <data> are set as 1 and otherwise 0. They can be displayed in hexadecimal (H'), octal (O'), or binary (B').<br><br>Specification for a data size condition:<br>**data** <data> <data size> <location><br><data size>: Select one of the followings for <data size>;<br>For a break with byte-length data: **byte** <location><br>For a break with word-length data: **word** <location><br>For a break with longword-length data: **long**<br><location>: Specifies a location from a 4-byte boundary.<br>**4n**: Specifies data from a 4n-byte boundary.<br>**4n1**: Specifies data from a 4n-byte boundary + a byte.<br>**4n2**: Specifies data from a 4n-byte boundary + two bytes.<br>**4n3**: Specifies data from a 4n-byte boundary + three bytes.<br><br>When no specification is made, 4n will be selected.<br><br>Add [**not**] to break with a data value or in the range that have not been specified. |

Description of the be*n* command <option> (cont)

<table>
<tr><th>Parameter</th><th colspan="2">Description</th></tr>
<tr><td rowspan="4"><BUS></td><td colspan="2">Selects a bus condition.<br><br>When no specification is made for bus, the condition will be cleared. When the type of an access to a bus is prefetch, the read condition is assumed.<br><br><BUS> = bus [cpu || dmac] [prefetch || data] [read || write]</td></tr>
<tr><td>cpu || dmac</td><td>Sets a bus status condition.<br><br>For a break in execution cycles: cpu<br>For a break in DMA/DTC cycles: dmac</td></tr>
<tr><td>prefetch || data</td><td>Sets a condition to break when the bus value matches.<br><br>For a break after executing the instruction as the address value at a program fetch matches: prefetch<br>For a break after accessing data: data</td></tr>
<tr><td>read || write</td><td>Specifies a read/write condition. A read condition must be selected when the access type to a bus is prefetch.<br><br>For a break in read cycles: read<br>For a break in write cycles: write</td></tr>
<tr><td><COUNT></td><td colspan="2">Specifies the number of times that a condition matches.<br><br><COUNT> = count <count><br><br>When no specification is made for count, the condition will be cleared.<br><br>count: Specifies the number of times that a condition matches.<br><br>For a break by the number of times that a condition matches:<br>count <count><br>Values H’1 to H’FFFF can be specified for count.</td></tr>
</table>

Description of the be*n* command <option> (cont)

| Parameter | Description | |
|---|---|---|
| <PROBE> | Specifies an external probe signal condition.<br><PROBE> = **probe** <[**probe1** [**high** \|\| **low**]] \| [**probe2** [**high** \|\| **low**]] \| [**probe3** [**high** \|\| **low**]] \| [**probe4** [**high** \|\| **low**]]><br>When neither high nor low is specified, each of the probe conditions will be cleared. | |
| | **probe1** | Specifies the probe1 signal condition.<br>Specification for the probe1 signal condition: **probe1** [**high** \|\| **low**] |
| | **probe2** | Specifies the probe2 signal condition.<br>Specification for the probe2 signal condition: **probe2** [**high** \|\| **low**] |
| | **probe3** | Specifies the probe3 signal condition.<br>Specification for the probe3 signal condition: **probe3** [**high** \|\| **low**] |
| | **probe4** | Specifies the probe4 signal condition.<br>Specification for the probe4 signal condition: **probe4** [**high** \|\| **low**] |

Description of the be*n* command <option> (cont)

<table>
<tr><th>Parameter</th><th colspan="2">Description</th></tr>
<tr><td rowspan="10"><INTERRUPT></td><td colspan="2">Specifies signal conditions.<br><br><INTERRUPT> = **interrupt** <[**irq0** [**high** || **low**]] | [**irq1** [**high** || **low**]] |<br>[**irq2** [**high** || **low**]] | [**irq3** [**high** || **low**]] |<br>[**irq4** [**high** || **low**]] | [**irq5** [**high** || **low**]] |<br>[**irq6** [**high** || **low**]] | [**irq7** [**high** || **low**]] |<br>[**nmi** [**high** || **low**]]><br><br>When neither high nor low is specified, each of the interrupt conditions will be cleared.</td></tr>
<tr><td>**irq0**</td><td>Specifies the IRQ0 signal condition.<br><br>For a break by the IRQ0 signal condition:<br>**irq0** [**high** || **low**]</td></tr>
<tr><td>**irq1**</td><td>Specifies the IRQ1 signal condition.<br><br>For a break by the IRQ1 signal condition:<br>**irq1** [**high** || **low**]</td></tr>
<tr><td>**irq2**</td><td>Specifies the IRQ2 signal condition.<br><br>For a break by the IRQ2 signal condition:<br>**irq2** [**high** || **low**]</td></tr>
<tr><td>**irq3**</td><td>Specifies the IRQ3 signal condition.<br><br>For a break by the IRQ3 signal condition:<br>**irq3** [**high** || **low**]</td></tr>
<tr><td>**irq4**</td><td>Specifies the IRQ4 signal condition.<br><br>For a break by the IRQ4 signal condition:<br>**irq4** [**high** || **low**]</td></tr>
<tr><td>**irq5**</td><td>Specifies the IRQ5 signal condition.<br><br>For a break by the IRQ5 signal condition:<br>**irq5** [**high** || **low**]</td></tr>
<tr><td>**irq6**</td><td>Specifies the IRQ6 signal condition.<br><br>For a break by the IRQ6 signal condition:<br>**irq6** [**high** || **low**]</td></tr>
<tr><td>**irq7**</td><td>Specifies the IRQ7 signal condition.<br><br>For a break by the IRQ7 signal condition:<br>**irq7** [**high** || **low**]</td></tr>
<tr><td>**nmi**</td><td>Specifies the NMI signal condition.<br><br>For a break by the NMI signal condition: **nmi** [**high** || **low**]</td></tr>
</table>

Description of the bec command <option>

| Parameter | Description |
|---|---|
| <CHANNEL> | Specifies channels for which the settings are canceled.<br><br>Specification of channels for which the settings are canceled:<br><channels>: [**1** \| **2** \| **3** \| **4**] \| [**all**]<br>Specify a number or **all**. When **all** is selected, all the settings will be canceled. |

Description of the bee command <option>

| Parameter | Description |
|---|---|
| <CHANNEL> | Specifies the execution of a break for each channel.<br><br>Specification of **enable** and **disable** for a channel:<br><CHANNEL> = <channels> <**enable** \|\| **disable**><br><channels>: [**1** \| **2** \| **3** \| **4**]<br>To enable a break at the specified address: **enable**<br>To disable a break at the specified address: **disable** |

[Description]

- Display

  This command displays the settings for each channel of on-emulator breaks.
- Selection

  This command makes settings for each channel of on-emulator breaks. The program stops when the specified condition is satisfied. For details on conditions that can be specified, refer to the description of <option> in the be*n* command section.
- Cancellation

  This command clears on-emulator breakpoints that have been set.
- Enable or Disable

  This command enables or disables on-emulator breakpoints that have been set.

### 6.2.6 Software Break Commands (BS, BSC, BSE)

- Display and Setting BREAKPOINT_SOFTWARE (BS)
- Cancellation BREAKPOINT_SOFTWARE_CLEAR (BSC)
- Enable or Disable
  BREAKPOINT_SOFTWARE_ENABLE (BSE)

[Command syntax and parameters]

- Display **bs**
- Setting **bs** <ADDRESS>
- Cancellation **bsc** <ADDRESS>
- Enable or Disable
  **bse** [<ADDRESS>] <MODE>

| Parameter | Type | Description |
|---|---|---|
| <ADDRESS> | Numeric | Sets an address condition.<br><ADDRESS> = **address** <address><br>When no specification is made, the resource information will be displayed.<br><address>: Address value (numeric) |
| <MODE> | Keyword | Enables or disables breakpoints.<br>**enable**: Enabled<br>**disable**: Disabled |

[Description]

- Setting

  This command sets software breakpoints. The emulator sets a software breakpoint by replacing the contents of the specified address with a break instruction (H'0000). Up to 255 software breakpoints can be set.

  Do not set software breakpoints to the following addresses:

  — An address whose memory content is H'0000

  — Areas other than CS (except for the on-chip ROM/RAM area)

  — Slot instruction of the delay branch instruction
- Cancellation

  This command clears software breakpoints that have been set.
- Enable or Disable

  This command enables or disables software breakpoints that have been set.

### 6.2.7 CLOCK (CK)

[Command syntax and parameters]

- Setting **ck** <mode>
- Display **ck**

| Parameter | Type | Description |
|---|---|---|
| <mode> | Keyword | Selects the clock signal.<br>**5**: Clock signal (5 MHz) of the emulator<br>**10**: Clock signal (10 MHz) of the emulator<br>**T**: Target<br>**xtal**: Crystal oscillator of the head board |

[Description]

This command displays and sets clock signal used in the MCU.

The user can select whether to use the clock signal from the user system or from the emulator. When the user selects a clock, the emulator resets the MCU. Therefore, the on-chip I/O registers and control registers hold the reset values.

This command displays the type of the clock signal that has been set.

### 6.2.8 CONFIGURATION_PLATFORM (CP)

[Command syntax and parameters]

- Setting **cp** [<option>…]
  <option> = <USER_VCC> | <INTERRUPT> | <HUDI>
  | <INPUT_FREQUENCY> | <AUD> | <MULTI_BREAK>
  | <FLASH_LOAD_OPTION> | <BUS_TIMEOUT>
- Display **cp**

| Parameter | Type | Description |
|---|---|---|
| <USER_VCC> | Numeric | Displays and selects the voltage level of the user system.<br><USER_VCC> = **user_vcc** <VCC><br>Specify the value as the voltage level of the user system for <VCC>. Select from the range 0.00 to 5.10. |
| <INTERRUPT> | Keyword | Displays and selects interrupts during the execution of steps.<br><INTERRUPT> = **interrupt** <INT><br><INT> = <**enable** \|\| **disable**><br>To enable interrupts: **enable**<br>To disable interrupts: **disable** |
| <HUDI> | Keyword | Displays and selects the HUDI clock.<br><HUDI> = **hudi** <hudi><br><hudi> = <**10MHz** \|\| **15MHz** \|\| **20MHz**> |
| <INPUT_ FREQUENCY> | Keyword | Displays and selects the input frequency.<br><INPUT_FREQUENCY> = **input_frequency** <fre><br><fre> = <**5MHz** \|\| **10MHz**> |
| <AUD> | Keyword | Displays and selects the AUD function.<br><AUD> = **aud** <AUD><br><AUD> = <**emulator** \|\| **user**><br>To use by the emulator: **emulator**<br>To use by the user: **user** |
| <MULTI_BREAK> | Keyword | Specifies whether or not to enable the multibreak function.<br><MULTI_BREAK> = **multi_break** <mul><br><mul> = <**multibreak** \|\| **non**><br>To enable the multibreak function: **multibreak**<br>To disable the multibreak function: **non** |

Description of the CONFIGURATION_PLATFORM (CP) options (cont)

<table>
<tr><th>Parameter</th><th>Type</th><th>Description</th></tr>
<tr><td><FLASH_LOAD _OPTION></td><td>Keyword</td><td>Specifies a method of loading a program to the on-chip flash memory.<br><FLASH_LOAD_OPTION> = <b>flash_load_option</b> <fla><br><fla> = <<b>disable</b> || <b>update</b> || <b>erase</b> || <b>allerase</b>><br><b>disable</b>: Loading of a program to the on-chip flash memory is prohibited. A verification error occurs if loading of a program to the on-chip flash memory is attempted.<br><b>update</b>: In this mode, a program can be loaded into the on-chip flash memory. The contents of the on-chip flash memory, except for the program to be loaded, will be the same as those before loading.<br><b>erase</b>: In this mode, a program can be loaded into the on-chip flash memory. Certain blocks in the on-chip flash memory are erased before loading a program.<br><b>allerase</b>: In this mode, a program can be loaded into the on-chip flash memory. The contents of the on-chip flash memory are all erased before loading a program.</td></tr>
<tr><td><BUS_TIMEOUT></td><td>Keyword</td><td>Specifies a bus timeout detection period.<br><BUS_TIMEOUT> = <b>bus_timeout</b> <bus><br><bus> = <<b>100us</b> || <b>1.6ms</b> || <b>13ms</b> || <b>210ms</b>></td></tr>
</table>

[Description]

This command displays and makes the settings required for the operation of the emulator.

[Examples]

1. To display the current settings:

    ***cp***

2. To set the voltage level of the user system as 4.10 V:

    ***cp user_vcc 4.10***

### 6.2.9 DEVICE_TYPE (DE)

[Command syntax and parameters]

- Setting **de** <device>
- Display **de**

| Parameter | Type | Description |
|---|---|---|
| <device> | Keyword | Sets up the device type to emulate, or displays the current device type.<br>**Default**: Displays the current device type.<br>**sh7058**: Set the device to the SH7058.<br>**7058**: Set the device to the SH7058. |

[Description]

This command displays the current device type.

[Examples]

1. To display the device type:

   ***de***

2. To set the device to the SH7058:

   ***de sh7058***

3. To set the device to the SH7058:

   ***de 7058***

### 6.2.10 ERAM (EM)

[Command syntax and parameters]

- Setting **em** [<mode>]
- Display **em**

<table>
<tr><th>Parameter</th><th>Type</th><th>Description</th></tr>
<tr><td><mode></td><td>Keyword</td><td>Sets the ERAM mode.<br><mode> = <user> || <eml> || <auto><br>When the emulation RAM is used by the user program;<br><user> = user<br>When the emulation RAM is used for emulation of the on-chip ROM;<br><eml> = eml <REL> || <ROM><br>When the emulation RAM is used to set software breakpoints automatically;<br><auto> = auto <REL> || <ROM></td></tr>
<tr><td><REL> || <ROM></td><td>Keyword</td><td>Releases and allocates the emulation RAM.<br>To release the emulation RAM: <REL> = release <eram><br>To allocate the emulation RAM: <ROM> = onrom <eram> <option></td></tr>
<tr><td><eram></td><td>Keyword</td><td>Selects the emulation RAM to be set.<br><eram> = <<b>eram0</b> || <b>eram1</b> || <b>eram2</b> || <b>eram3</b> || <b>eram4</b> || <b>eram5</b> || <b>eram6</b> || <b>eram7></b></td></tr>
<tr><td><option></td><td>Keyword</td><td>Makes the other settings.<br><option> = address <add> || toflash || toeram<br>address <add>: Specifies the address to be allocated.<br><add>: Numeric<br>toflash: Writes the contents of the emulation memory into the on-chip flash memory.<br>toeram: Writes the contents of the on-chip flash memory into the emulation RAM.</td></tr>
</table>

[Description]

This command sets an emulation memory or displays its contents.

[Examples]

1. To display the settings made for the emulation memory:

   ***em***

2. To use ERAM0 for emulation as the on-chip ROM of H’1000:

   ***em eml rom eram0 1000***

3. Not to release emulation of ERAM0:

   ***em eml release eram0***

4. To use the emulation RAM as a user RAM:

   ***em user***

5. To use the emulation RAM to set software breakpoints automatically:

   ***em auto***

### 6.2.11 FLASH_MEMORY (FM)

[Command syntax and parameters]

- Setting **fm** <LOCATE>
  <LOCATE> = <**0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** | **12** | **13** | **14** | **15** | **all**>

[Description]

This command erases the user mat of the selected on-chip flash memory.

[Examples]

1. To erase the user mat area 0:

   ***`fm 0`***

2. To erase the user mat areas 1 and 2:

   ***`fm 1 2`***

### 6.2.12 MAP_DISPLAY (MA)

[Command syntax and parameters]

- Display **ma**

[Description]

This command displays the current configuration of the memory map.

[Example]

To display the current configuration of the memory map:

***ma***

### 6.2.13 MODE (MO)

[Command syntax and parameters]

- Display **mo**
- Setting **mo** <MODE>
  <MODE> = <**0** || **1** || **2** || **3** || **TARGET**>

| **Parameter** | **Type** | **Description** |
|---|---|---|
| <MODE> | Keyword | Specifies the MCU mode.<br>**0**: Selects the 8-bit bus mode.<br>**1**: Selects the 16-bit bus mode.<br>**2**: Selects the on-chip ROM enabled mode.<br>**3**: Selects the on-chip ROM disabled mode.<br>**TARGET**: Selects the mode set in the user system. |

[Description]

This command selects the MCU mode.

[Example]

To set the MCU mode to mode 1:

```
mo 1
```

### 6.2.14 REFRESH (RF)

[Command syntax and parameters]

**rf**

[Description]

This command updates the HDI memory information to the latest information.

### 6.2.15 TEST_EMULATOR (TE)

[Command syntax and parameters]

- Test **te**

[Description]

Tests the on-chip flash memory area in the following order:

(1) Erases all area (except the user boot mat) in the on-chip flash memory. If an error occurs during erasing, the processing is suspended.

(2) Writes H’AA to all areas. If an error occurs during writing, the processing is suspended.

(3) Reads all areas. If the read data is not H’AA, the processing is suspended.

(4) Erases all areas. If an error occurs during erasing, the processing is suspended.

(5) Writes H’55 to all areas. If an error occurs during writing, the processing is suspended.

(6) Reads all areas. If the read data is not H’55, the processing is suspended.

(7) Erases all areas. If an error occurs during erasing, the processing is suspended.

(8) Writes H’00 to all areas. If an error occurs during writing, the processing is suspended.

(9) Reads all areas. If the read data is not H’00, the processing is suspended.

(10) Erases all data. If an error occurs during erasing, the processing is suspended.

(11) Command processing is completed.

### 6.2.16 TIMER (TI)

[Command syntax and parameters]

- Display **ti**
- Setting **ti** <timer resolution>
  <timer resolution> = <**20ns** || **1.6us** || **52us** || **CLOCK1** || **CLOCK2** || **CLOCK4** || **CLOCK8**>

| Parameter | Type | Description |
|---|---|---|
| <timer resolution> | Keyword | Sets or displays the minimum time for time measurement.<br>**20ns**: The minimum time is set to 20 ns.<br>**1.6us**: The minimum time is set to 1.6 us.<br>**52us**: The minimum time is set to 52 us.<br>**CLOCK1**: The minimum time is set to the CPU clock.<br>**CLOCK2**: The minimum time is set to 1/2 of the CPU clock.<br>**CLOCK4**: The minimum time is set to 1/4 of the CPU clock.<br>**CLOCK8**: The minimum time is set to 1/8 of the CPU clock. |

[Description]

This command displays or sets the minimum time for time measurement. This affects the execution time and the event timing.

### 6.2.17 TRACE_ACQUISITION (TA, TA*n*)

- Setting TRACEACQUISITION (TA)
- Selection TRACEACQUISITION*n* (TA*n*)
- Cancellation TRACEACQUISITION_CLEAR (TAC)

[Command syntax and parameters]

- Setting and display **ta** <option> [<option>…]
  <option> = <DISPLAY> | <SEQUENTIAL> | <TRIGGERTIME> | <FREE> | <FULL> | <UNIT> | <SUPPRESS> | <RESET> | <SELECT>
- Selection **ta***n* (*n* = 1…12) <option> [<option>…]
  <option> = <ACTION> | <ADDRESS> | <DATA> | <BUS> | <PROBE> | <INTERRUPT> | <COUNT> | <DELAY>
- Cancellation **tac** <CHANNEL>

Description of command ta <option> (specify one or more conditions)

| Parameter | Type | Description |
|---|---|---|
| <DISPLAY> | Keyword | Displays trace points that have been set.<br>Specification: **display** |
| <SEQUENTIAL> | Numeric or keyword | Selects the sequential level.<br>Specification: **sequential** <level><br><level> = <**1** \|\| **2** \|\| **3** \|\| **4** \|\| **5** \|\| **6** \|\| **7** \|\| **8**> |
| <TRIGGERTIME> | Numeric or keyword | Specifies the channels between which the trace time is measured.<br>Specification: **triggertime** <channels><br><channels> = <**1-2** \|\| **3-4** \|\| **5-6** \|\| **7-8**> |
| <FREE> | Keyword | Specifies free trace.<br>Specification: **free [ok]**<br>**free ok**: Sets free trace.<br>**free**: Clears free trace. |
| <FULL> | Keyword | Specifies the processing when the trace buffer is full.<br>Specification: **full** <action><br><action> = <**noaction** \|\| **stoptrace** \|\| **stopexecution**><br>**noaction**: No special processing is done. Continues storing the trace results from the top of the trace buffer.<br>**stoptrace**: Stops storing the trace results.<br>**stopexecution**: Stops execution and stops storing trace results. |
| <UNIT> | Keyword | Specifies the time unit for trace measurement.<br>Specification: **unit** <timeunit><br><timeunit> = <**20ns** \|\| **1.6us** \|\| **52us**> |
| <SUPPRESS> | Keyword | Specifies the bus cycle type which trace results are not acquired.<br>Specification: **suppress** <type><br><type> = **dmac** |
| <RESET> | Keyword | Specifies the reset point.<br>Specification: **reset [ok]**<br>**reset ok**: Sets a reset point.<br>**reset**: Clears the reset point. |
| <SELECT> | Keyword | Specifies the display format of timestamp and IRQ7 to IRQ0.<br>Specification: **select** <**timestamp** \|\| **indication**><br>**timestamp**: Enables all the 32-bit timestamp counters.<br>**indication**: Displays the status of IRQ7 to IRQ0 in bit units. The lower 16 bits of the timestamp counter are fixed to 0. |

Description of command ta*n* <option> (specify one or more conditions)

| Parameter | Type | Description |
|---|---|---|
| *n* | Numeric | Sets the channel number. A value from 1 to 12 can be specified. |

| Parameter | Description |
|---|---|
| <ACTION> | Specifies the processing after the trace condition is satisfied.<br>Specification: **action** <operate><br><operate> = <**tracestop** \|\| **rangetrace** [**next**] \|\| **acquisition** \|\| **none**> \| [**trigger**]<br>**tracestop**: Stops trace.<br>**rangetrace** [**next**]: Starts range trace. When **next** is specified, the next channel is used for the trace condition setting.<br>**acquisition**: Acquires trace condition.<br>**none**: Specifies the trigger output condition for “Time measurement between trigger Outputs” of Trace Acquisition Condition.<br>**trigger**: Specifies the trigger output after the trace condition is satisfied. |

| Parameter | Description |
|---|---|
| <ADDRESS> | Specifies the address condition.<br><ADDRESS> = **address** <<address> \|\| <start address> <end address> [**outside**]><br>When the parameters are not specified, clears the address condition.<br>To specify the address condition: **address** <address><br><address>: Address value (numeric)<br>To specify the range address condition:<br>**address** <start address> <end address> [**outside**]<br><start address>: The start address of the specified range (numeric)<br><end address>: The end address of the specified range (numeric)<br>Specify **outside** to specify the address outside the specified range as the condition. |

Description of command ta*n* <option> (cont)

| Parameter | Description |
|---|---|
| <DATA> | Specifies the data condition.<br><DATA> = **data** <data> [<data mask>] [<data size> <location>] [**not**]<br>When the parameters are not specified, clears the data condition.<br><br>To specify the data condition: **data** <data><br><data>: Data value (numeric)<br><br>To specify the data mask condition:<br>d**ata** <data> <data mask><br><data mask>: The digits to be masked for bits of <data> are set as 1 and otherwise 0. They can be displayed in hexadecimal (H'), octal (O'), or binary (B').<br><br>To specify the data size condition:<br>d**ata** <data> <data size> <location><br><data size>: Select from the following:<br>**byte** <location>: When the byte data size is the condition.<br>**word** <location>: When the word data size is the condition.<br>**long**: When the longword data size is the condition.<br><br><location> specifies the position from the 4-byte boundary.<br>**4n**: Specifies data from a 4n-byte boundary.<br>**4n1**: Specifies data from a 4n-byte boundary + a byte.<br>**4n2**: Specifies data from a 4n-byte boundary + two bytes.<br>**4n3**: Specifies data from a 4n-byte boundary + three bytes.<br>When <location> is not specified, 4n is assumed.<br>When <ACTION> is set to **rangetrace**, <location> cannot be specified.<br><br>Add [**not**] to specify a value other than the specified data value or a value outside the specified range as the condition. |

Description of command ta*n* <option> (cont)

<table>
<tr><th>Parameter</th><th colspan="2">Description</th></tr>
<tr><td><BUS></td><td colspan="2">Specifies the bus condition.<br><BUS> = bus [[cpu || dmac || aud] [prefetch || data] [read || write]]<br>Clears the condition when the bus condition is not specified.<br>When the access type is prefetch, the read condition is assumed.<br>When <ACTION> is set to rangetrace, <BUS> cannot be specified.</td></tr>
<tr><td></td><td>cpu || dmac</td><td>Sets the bus status condition.<br>To set the execution cycle as the condition: cpu<br>To set the DMAC cycle as the condition: dmac</td></tr>
<tr><td></td><td>prefetch || data</td><td>Sets the bus access type as the condition.<br>To set the instruction execution cycle whose address bus matches the bus type when the program is fetched: prefetch<br>To set the data access as the bus condition: data</td></tr>
<tr><td></td><td>read || write</td><td>Sets the read or write condition. When the bus access type is prefetch, the read condition is assumed.<br>To set the read cycle as the bus condition: read<br>To set the write cycle as the bus condition: write</td></tr>
</table>

<table>
<tr><th>Parameter</th><th colspan="2">Description</th></tr>
<tr><td><PROBE></td><td colspan="2">Specifies the external probe signal condition.<br><PROBE> = <b>probe</b> <[<b>probe1</b> [<b>high</b> || <b>low</b>]] | [<b>probe2</b> [<b>high</b> || <b>low</b>]] |<br>[<b>probe3</b> [<b>high</b> || <b>low</b>]] | [<b>probe4</b> [<b>high</b> || <b>low</b>]]><br>When high or low is not specified, clears each probe condition.</td></tr>
<tr><td></td><td>probe1</td><td>To specify probe1 signal condition: probe1 [high || low]</td></tr>
<tr><td></td><td>probe2</td><td>To specify probe2 signal condition: probe2 [high || low]</td></tr>
<tr><td></td><td>probe3</td><td>To specify probe3 signal condition: probe3 [high || low]</td></tr>
<tr><td></td><td>probe4</td><td>To specify probe4 signal condition: probe4 [high || low]</td></tr>
</table>

Description of command ta*n* <option> (cont)

<table>
<tr><th>Parameter</th><th colspan="2">Description</th></tr>
<tr><td rowspan="10"><INTERRUPT></td><td colspan="2">Specifies the interrupt signal condition.<br><INTERRUPT> = <b>interrupt</b> <[<b>irq0</b> [<b>high</b> || <b>low</b>]] | [<b>irq1</b> [<b>high</b> || <b>low</b>]] |<br>[<b>irq2</b> [<b>high</b> || <b>low</b>]] | [<b>irq3</b> [<b>high</b> || <b>low</b>]]><br>[<b>irq4</b> [<b>high</b> || <b>low</b>]] | [<b>irq5</b> [<b>high</b> || <b>low</b>]]><br>[<b>irq6</b> [<b>high</b> || <b>low</b>]] | [<b>irq7</b> [<b>high</b> || <b>low</b>]]><br>[<b>nmi</b> [<b>high</b> || <b>low</b>]]><br>When high or low is not specified, clears each interrupt condition.</td></tr>
<tr><td>irq0</td><td>To specify IRQ0 signal condition: irq0 [high || low]</td></tr>
<tr><td>irq1</td><td>To specify IRQ1 signal condition: irq1 [high || low]</td></tr>
<tr><td>irq2</td><td>To specify IRQ2 signal condition: irq2 [high || low]</td></tr>
<tr><td>irq3</td><td>To specify IRQ3 signal condition: irq3 [high || low]</td></tr>
<tr><td>irq4</td><td>To specify IRQ4 signal condition: irq4 [high || low]</td></tr>
<tr><td>irq5</td><td>To specify IRQ5 signal condition: irq5 [high || low]</td></tr>
<tr><td>irq6</td><td>To specify IRQ6 signal condition: irq6 [high || low]</td></tr>
<tr><td>irq7</td><td>To specify IRQ7 signal condition: irq7 [high || low]</td></tr>
<tr><td>nmi</td><td>To specify NMI signal condition: nmi [high || low]</td></tr>
</table>

<table>
<tr><th>Parameter</th><th>Description</th></tr>
<tr><td><COUNT></td><td>Specifies the number of times the condition is matched.<br><COUNT> = count <count><br>When the parameter is not specified, clears the count condition.<br><br>To specify the number of times the condition is matched:<br>count <count><br><count>: A value from H’1 to H’FFFF</td></tr>
</table>

<table>
<tr><th>Parameter</th><th>Description</th></tr>
<tr><td><DELAY></td><td>Specifies the number of trace delay cycles as the condition.<br><DELAY> = delay <delay><br>When the parameter is not specified, clears the delay condition.<br><br>To specify the number of trace delay cycles:<br>delay <delay><br><delay>: The number of trace delay cycles in decimal.</td></tr>
</table>

Description of command tac <option> (specify one or more conditions)

| Parameter | Description |
|---|---|
| <CHANNEL> | Clears the channel that has been set.<br><CHANNEL> = <**1** \|\| **2** \|\| **3** \|\| **4** \|\| **5** \|\| **6** \|\| **7** \|\| **8** \|\| **9** \|\| **10** \|\| **11** \|\| **12**> \| <**all**><br>Select a numeric or **all**. When **all** is selected, all channel settings are cleared. |

[Description]

- Setting

  This command sets conditions for acquiring trace information. For conditions that can be set, refer to the description of command ta <option>.
- Selection

  This command select channel condition for acquiring trace information. When the specified condition is satisfied, the program execution is terminated. For conditions that can be set, refer to the description of command ta*n* <option>.
- Cancellation

  This command clears conditions for acquiring trace information.

### 6.2.18 TRACE_ALL (TL)

[Command syntax and parameters]

- Setting, display **tl** <option>
  <option> = <<record> [counter] || **old** [counter] >> || **all**

[Description]

Displays the trace records from the data in the cycle specified by [record]. The number of records is specified by [counter].

<record> must be –131071 or more and 65534 or less.

[counter] must be 0 or more and 131071 or less.

[Examples]

1. To display 10 records of the latest trace data that was executed most recently:

   ***tl***

2. To display 10 records of trace data from –30 cycle:

   ***tl -30***

3. To display 20 records of trace data from –50 cycle:

   ***tl –50 20***

4. To display 10 records of the oldest data in the trace buffer:

   ***tl OLD***

5. To display 5 records of the oldest data in the trace buffer:

   ***tl old 5***

6. To display all trace data in the trace buffer:

   ***tl all***

[Notes]

1. When all parameters are omitted, 10 records of trace data which was executed most recently are displayed.
2. When old is specified, the oldest data in the trace buffer becomes the display start trace record.
3. When all is specified, all data in the trace buffer are output.
4. When <count> is omitted, 10 is assumed.
5. When a cycle prior to the cycle in the trace buffer is specified, the number of trace records specified by <count> from the oldest data in the trace buffer are displayed.
6. When a cycle later than the cycles in the trace buffer is specified, the following message is displayed:

   Trace record out of range

7. When the number of records in the trace buffer is large, and all is specified, it may take some time to output all the data and in the meantime, the HDI cannot be operated.

#### 6.2.19 TRACE_COMPARE (TC)

[Command syntax and parameters]

**tc** <filename>
<filename>: Name of the trace file to be compared.

[Description]

Compares the trace data with the trace file which is saved (refer to section 6.2.20, TRACE_SAVE).

### 6.2.20 TRACE_SAVE (TV)

[Command syntax and parameters]

**tv** <filename>
<filename>: Name of the trace file to be saved.

[Description]

Saves the trace data in a file.

### 6.2.21 TRACE_SEARCH (TS)

[Command syntax and parameters]

**ts [**<option>…]

<option> = [<ADDRESS> | <DATA> | <BUS> | <PROBE> | <INTERRUPT> | <TIME> | <FROM>]

Description of command ts <option> (specify one or more conditions)

| Parameter | Description |
|---|---|
| <ADDRESS> | Specifies the address condition to be displayed.<br>Specification: <ADDRESS> = **address** <address> [<address mask>[**not**]]<br>When the parameter is not specified, clears the address condition.<br>To specify the address condition: **address** <address> [**not**]<br><address>: Address value (numeric)<br>Add **not** to specify a value that is not specified as the condition.<br>To specify the address mask condition:<br>**address** <address> <address mask> [**not**]<br><address mask>: The digits to be masked for bits of <address> are set as 1 and otherwise 0. They can be displayed in hexadecimal (H’), octal (O’), or binary (B’).<br>Add [**not**] to specify a value other than the specified address value or a value outside the specified range as the condition. |

| Parameter | Description |
|---|---|
| <DATA> | Specifies the data condition.<br><DATA> = **data** <data> [<data mask>] [<data size> <location>] [**not**]<br>When the parameters are not specified, clears the data condition.<br><br>To specify the data condition: **data** <data><br><data>: Data value (numeric)<br><br>To specify the data mask condition:<br>**data** <data> <data mask><br><data mask>: The digits to be masked for bits of <data> are set as 1 and otherwise 0. They can be displayed in hexadecimal (H’), octal (O’), or binary (B’).<br><br>To specify the data size condition:<br>**data** <data> <data size> <location><br><data size>: Select from the following:<br>**byte** <location>: When the byte data size is the condition.<br>**word** <location>: When the word data size is the condition.<br>**long**: When the longword data size is the condition.<br><br><location> specifies the position from the 4-byte boundary.<br>**4n**: Specifies data from a 4n-byte boundary.<br>**4n1**: Specifies data from a 4n-byte boundary + a byte.<br>**4n2**: Specifies data from a 4n-byte boundary + two bytes.<br>**4n3**: Specifies data from a 4n-byte boundary + three bytes.<br>When <location> is not specified, 4n is assumed.<br><br>Add [**not**] to specify a value other than the specified data value or a value outside the specified range as the condition. |

Description of command ts <option> (cont)

<table>
<tr><th>Parameter</th><th colspan="2">Description</th></tr>
<tr><td rowspan="4"><BUS></td><td colspan="2">Specifies the bus condition.<br><BUS> = bus [cpu || dmac || aud] [prefetch || data] [read || write]<br>Clears the condition when the bus condition is not specified.<br>When the access type is prefetch, the read condition is assumed.</td></tr>
<tr><td>cpu || dmac</td><td>Sets the bus status condition.<br>To set the execution cycle as the condition: cpu<br>To set the DMAC cycle as the condition: dmac</td></tr>
<tr><td>prefetch || data</td><td>Sets the bus access type as the condition.<br>To set the instruction execution cycle whose address bus matches the bus type when the program is fetched: prefetch<br>To set the data access as the bus condition: data</td></tr>
<tr><td>read || write</td><td>Sets the read or write condition. When the bus access type is prefetch, the read condition is assumed.<br>To set the read cycle as the bus condition: read<br>To set the write cycle as the bus condition: write</td></tr>
</table>

<table>
<tr><th>Parameter</th><th colspan="2">Description</th></tr>
<tr><td rowspan="5"><PROBE></td><td colspan="2">Specifies the external probe signal condition.<br><PROBE> = <b>probe</b> <[<b>probe1</b> [<b>high</b> || <b>low</b>]] | [<b>probe2</b> [<b>high</b> || <b>low</b>]] |<br>[<b>probe3</b> [<b>high</b> || <b>low</b>]] | [<b>probe4</b> [<b>high</b> || <b>low</b>]]><br>When high or low is not specified, clears each probe condition.</td></tr>
<tr><td>probe1</td><td>To specify probe1 signal condition: probe1 [high || low]</td></tr>
<tr><td>probe2</td><td>To specify probe2 signal condition: probe2 [high || low]</td></tr>
<tr><td>probe3</td><td>To specify probe3 signal condition: probe3 [high || low]</td></tr>
<tr><td>probe4</td><td>To specify probe4 signal condition: probe4 [high || low]</td></tr>
</table>

Description of command ts <option> (cont)

<table>
<tr><th>Parameter</th><th colspan="2">Description</th></tr>
<tr><td rowspan="10"><INTERRUPT></td><td colspan="2">Specifies the interrupt signal condition.<br><INTERRUPT> = <b>interrupt <</b>[<b>irq0</b> [<b>high</b> || <b>low</b>]] | [<b>irq1</b> [<b>high</b> || <b>low</b>]] |<br>[<b>irq2</b> [<b>high</b> || <b>low</b>]] | [<b>irq3</b> [<b>high</b> || <b>low</b>]] |<br>[<b>irq4</b> [<b>high</b> || <b>low</b>]] | [<b>irq5</b> [<b>high</b> || <b>low</b>]] |<br>[<b>irq6</b> [<b>high</b> || <b>low</b>]] | [<b>irq7</b> [<b>high</b> || <b>low</b>]] |<br>[<b>nmi</b> [<b>high</b> || <b>low</b>]]><br>When high or low is not specified, clears the timestamp condition.</td></tr>
<tr><td>irq0</td><td>To specify IRQ0 signal condition: irq0 [high || low]</td></tr>
<tr><td>irq1</td><td>To specify IRQ1 signal condition: irq1 [high || low]</td></tr>
<tr><td>irq2</td><td>To specify IRQ2 signal condition: irq2 [high || low]</td></tr>
<tr><td>irq3</td><td>To specify IRQ3 signal condition: irq3 [high || low]</td></tr>
<tr><td>irq4</td><td>To specify IRQ4 signal condition: irq4 [high || low]</td></tr>
<tr><td>irq5</td><td>To specify IRQ5 signal condition: irq5 [high || low]</td></tr>
<tr><td>irq6</td><td>To specify IRQ6 signal condition: irq6 [high || low]</td></tr>
<tr><td>irq7</td><td>To specify IRQ7 signal condition: irq7 [high || low]</td></tr>
<tr><td>nmi</td><td>To specify NMI signal condition: nmi [high || low]</td></tr>
</table>

Description of command ts <option> (cont)

| **Parameter** | **Description** |
|---|---|
| <TIME> | Specifies the timestamp condition.<br><TIME> = **time point** \|\| **range from** <time1> [**to** <time2>]<br>When the parameter is not specified, the timestamp is cleared.<br><br>**time**<br><br>To specify the timestamp condition:<br>**time point** \|\| **range from** <time1> [**to** <time2>]<br><br>To search with the specified time:<br>**time point from** <time1><br><br>To search with the specified range of time:<br>**time range from** <time1> **to** <time2><br><br>The specification of the search time <time1> to and <time2>:<br>hhh[:mm[:ss[:uuuuuu]]]<br>hhh: Hours (0 to 999 are valid)<br>mm: Minutes (0 to 59 are valid)<br>ss: Seconds (0 to 59 are valid)<br>uuuuuu: Microseconds (0 to 999999 are valid) |

| **Parameter** | **Description** |
|---|---|
| <FROM> | Specifies the trace start location. When the location is not specified, the start of 64k is assumed. Search cycle range is from –65534 to 65534.<br><FROM> = **from** <cycle><br><cycle>: A numeric from –65534 to 65534 |

[Description]

Specifies the conditions for displaying trace data. For each condition, see the description of command ts <option>.

### 6.2.22 USER_SIGNALS (US)

[Command syntax and parameters]

- Display **us**
- Enabled/disabled **us** <**enable** [**nmi** | **reset** | **busrq** | **wait**] **||**
  **disable** [**nmi** | **reset** | **busrq** | **wait**]>

[Description]

Sets whether or not the user signals (NMI, reset, standby, bus request, and wait) are enabled or disabled. When the parameter is not set, displays the enabled or disabled status of these signals.

[Examples]

1. To display the status of the user signals:

   ***us***

2. To enable the NMI signal:

   ***us enable nmi***

3. To disable the BREQ signal:

   ***us disable busrq***

# Section 7 Error Messages

## 7.1 Error Messages of the Emulator

### 7.1.1 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 HDI is used.
Table 7.1 lists error messages at emulator initiation.

**Table 7.1 Error Messages at Initiation**

| Error Message | Description and Solution |
|---|---|
| Can't find config file | The configuration file that is required to initiate the HDI cannot be found. Re-install the HDI. |
| Invalid config file | The configuration file that is required to initiate the HDI is invalid. Re-install the HDI. |
| Main Board not Support(XX XX XX)<br>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 HDI, re-connect the user system interface cable, turn on the power of the emulator, and re-initiate the HDI. If the problem is not solved, contact the Hitachi sales office where you purchased the HDI, and describe the error occurrence in detail. |
| Evaluation Board not Support(XX XX XX)<br>EVA Chip Board not Support(XX XX XX) | Exit the HDI, re-connect the user system interface cable, turn on the power of the emulator, and re-initiate the HDI. If the problem is not solved, contact the Hitachi sales office where you purchased the HDI, and describe the error occurrence in detail. |
| Can't initialize G/A registers | An error occurred during the initialization of the emulator. Exit the HDI, re-connect the user system interface cable, turn on the power of the emulator, and re-initiate the HDI. If the problem is not solved, contact the Hitachi sales office where you purchased the HDI, and describe the error occurrence in detail. |
| There is no effective clock source | A valid clock source cannot be found. Exit the HDI, re-connect the user system interface cable, connect a valid clock source by referring to the manual, turn on the power of the emulator, and re-initiate the HDI. If the problem is not solved, contact the Hitachi sales office where you purchased the HDI, and describe the error occurrence in detail. |

**Table 7.1 Error Messages at Initiation (cont)**

| Error Message | Description and Solution |
|---|---|
| Firmware Name Error<br>Can't find firmware file<br>Firmware open Error<br>Firmware Download Error | There is an error in the file that is required at emulator initiation. Exit the HDI, and re-install the HDI. Reconnect the user system interface cable, turn on the power of the emulator, and re-initiate the HDI. If the problem is not solved, contact the Hitachi sales office where you purchased the HDI, and describe the error occurrence in detail. |
| JTAG Timeout Srval Error<br>JTAG Packet Receive Error | Exit the HDI, reconnect the user system interface cable, turn on the power of the emulator, and re-initiate the HDI. If the problem is not solved, contact the Hitachi sales office where you purchased the HDI, and describe the error occurrence in detail. |
| TARGET ERROR | An error other than above has occurred. Exit the HDI, reconnect the user system interface cable, turn on the power of the emulator, and re-initiate the HDI. If the problem is not solved, contact the Hitachi sales office where you purchased the HDI, and describe the error occurrence in detail. |

### 7.1.2 Error messages during Emulation

The emulator displays error messages if an error occurs during emulation in the dedicated message dialog box when the HDI is used, and on the status bar. Table 7.2 lists error messages during emulation.

**Table 7.2 Error Messages during Emulation**

| Error Message | Description and Solution |
|---|---|
| No Clock occurred | Clock oscillation has stopped. Exit the HDI, reconnect the user system interface cable, connect a valid clock source by referring to the manual, turn on the power of the emulator, and re-initiate the HDI. If the problem is not solved, contact the Hitachi sales office where you purchased the HDI, and describe the error occurrence in detail. |

## 7.2 HDI Error Messages Related to the Emulator

Error messages that occur in HDI processing are also displayed in error-message dialog boxes. Tables 7.3 lists these error messages, descriptions of the errors, and solutions to the errors.

**Table 7.3 HDI Error Messages**

| Error Message | Description and Solution |
|---|---|
| Command not ready | The function that was attempted to be executed is not available now. |
| NOT FOUND | The specified data or information does not exist. Check the specified data. |
| VERIFY ERROR<br>ADDRESS H'xxxxxxxx WRITE: H'xx<br>Read: H'xx | Verification error occurred. (xx: numerical value). |
| Invalid address | The specified address is invalid. |
| Invalid version number in target configuration | The HDI version when the session file was created is different from the current HDI. Do not use the session file of the old version. |
| function not available<br>Currently not available<br>Command currently not available | The used function is not available. |
| Not support | The function is not supported. |
| Invalid parameter | The parameter of the command is incorrect. |

## 7.3 Error Messages for the LAN Driver

When the LAN adapter (HS6000ELN01H) is used, error messages of the following type may be output due to a network problem.

WSxxxxxxxx: <error message>

| | |
|---|---|
| WSxxxxxxxx | : Error code (xxxxxxxx is an alphanumeric string) |
| <error message> | : Error message that corresponds to the error code |

If such an error message is output, consult your system manager.

# Appendix A User System Interface Circuit

## A.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.

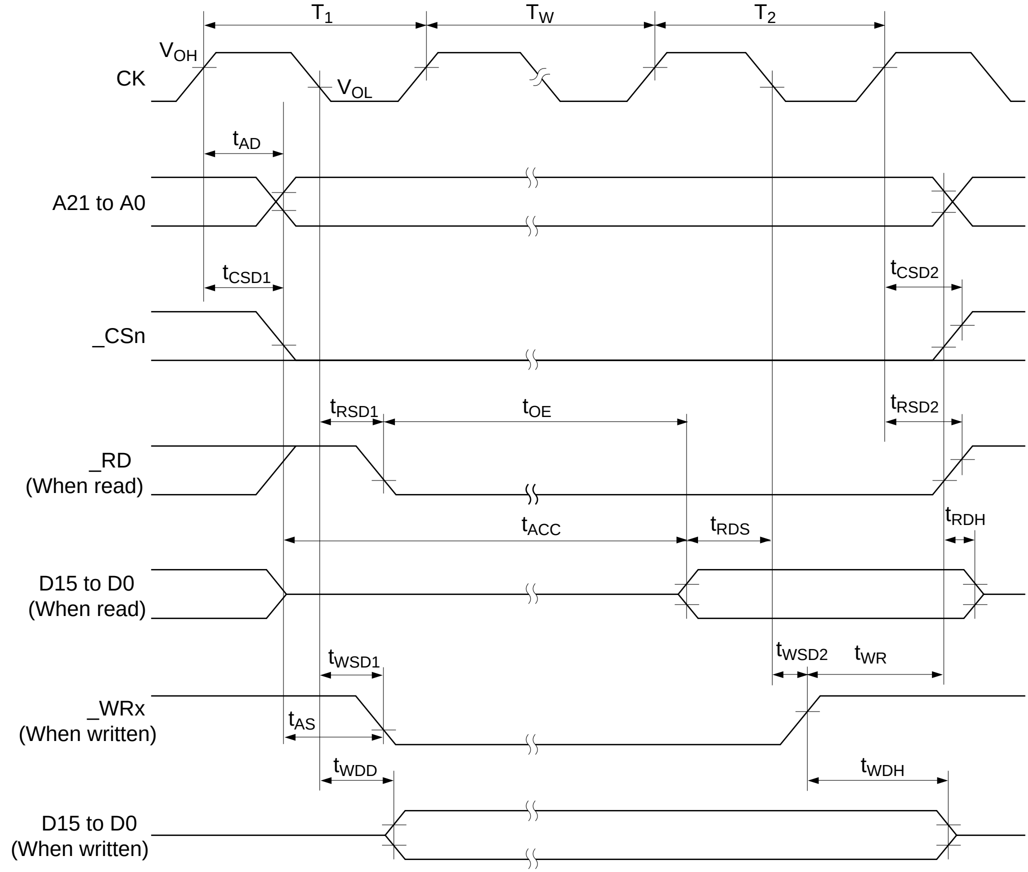
The AC timing values when using the emulator are shown in table A.1.

Note: The values with the emulator connected, in table A.1, are measurements for reference and are not guaranteed values.

**Table A.1 Bus Timing when Using the Emulator (Bus Clock: 20.0 MHz)**

| | MCU Specifications (ns) | | Values with Emulator Connected (ns) | |
|---|---|---|---|---|
| Item | Min | Max | Min | Max |
| tRDS | 15 | — | 18 | — |
| tACC | tcyc x (n + 1.5) – 39 (n is the number of waits) | — | tcyc x (n + 1.5) – 45 (n is the number of waits) | — |

The basic bus cycle (software wait) is shown in figure A.1. The user system interface circuits connected to the user system are shown in figure A.2.




Note: tRDH: Specified from the earliest negation timing from A21 to A0, _CSn, or _RD.

**Figure A.1 Basic Bus Cycle (Software Wait)**

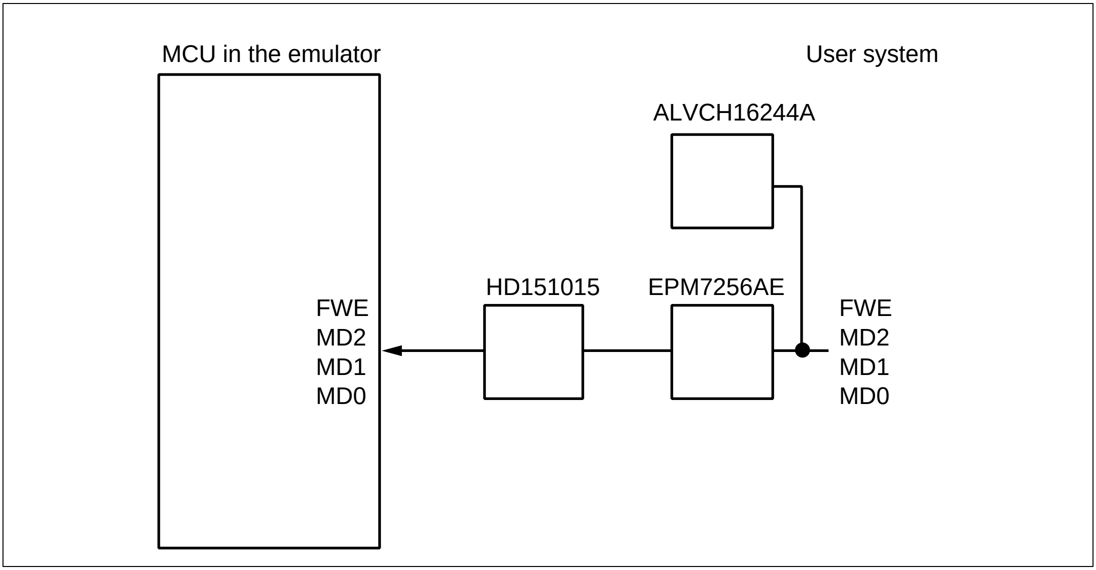



**Figure A.2 User System Interface Circuits (1)**

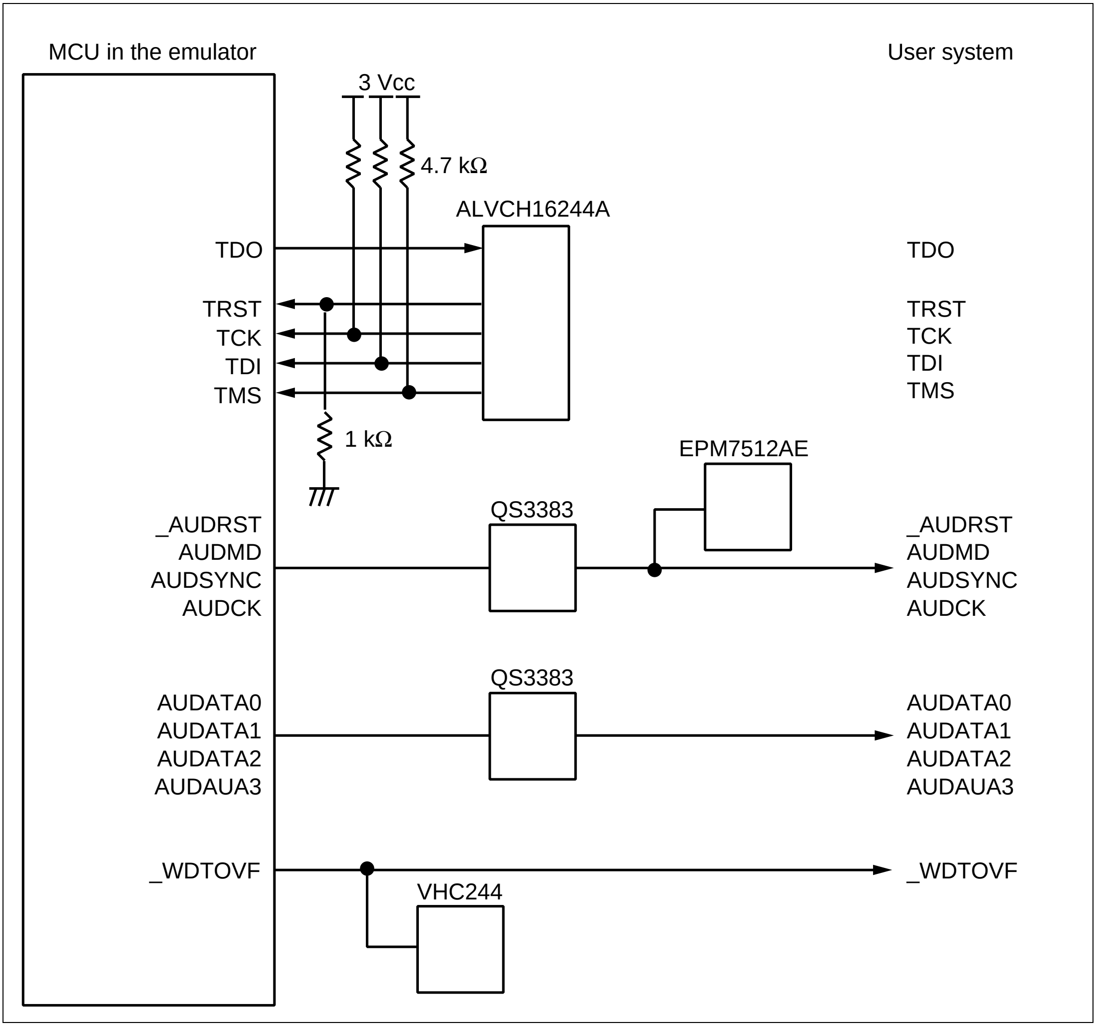



**Figure A.2 User System Interface Circuits (2)**

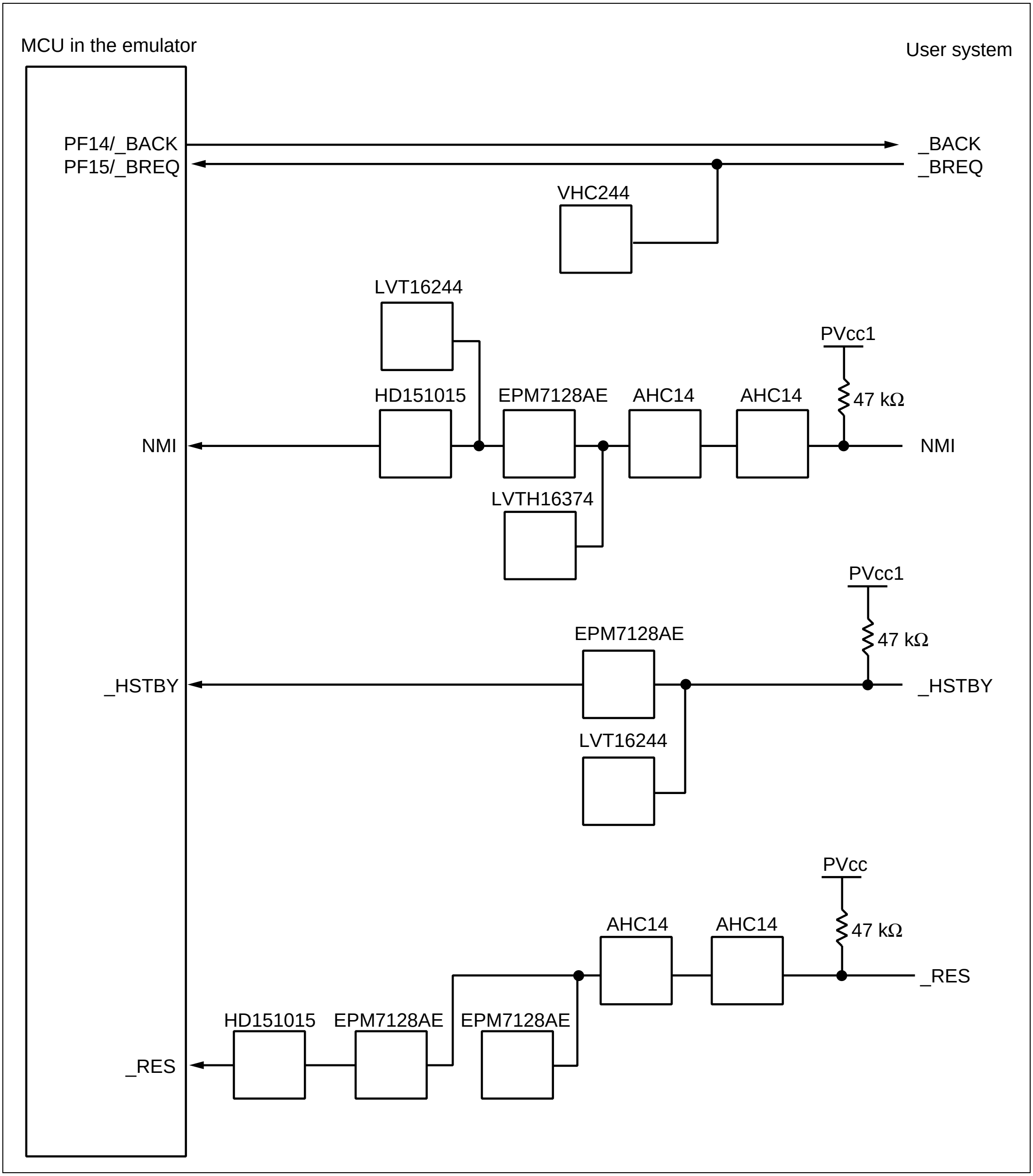



**Figure A.2 User System Interface Circuits (3)**

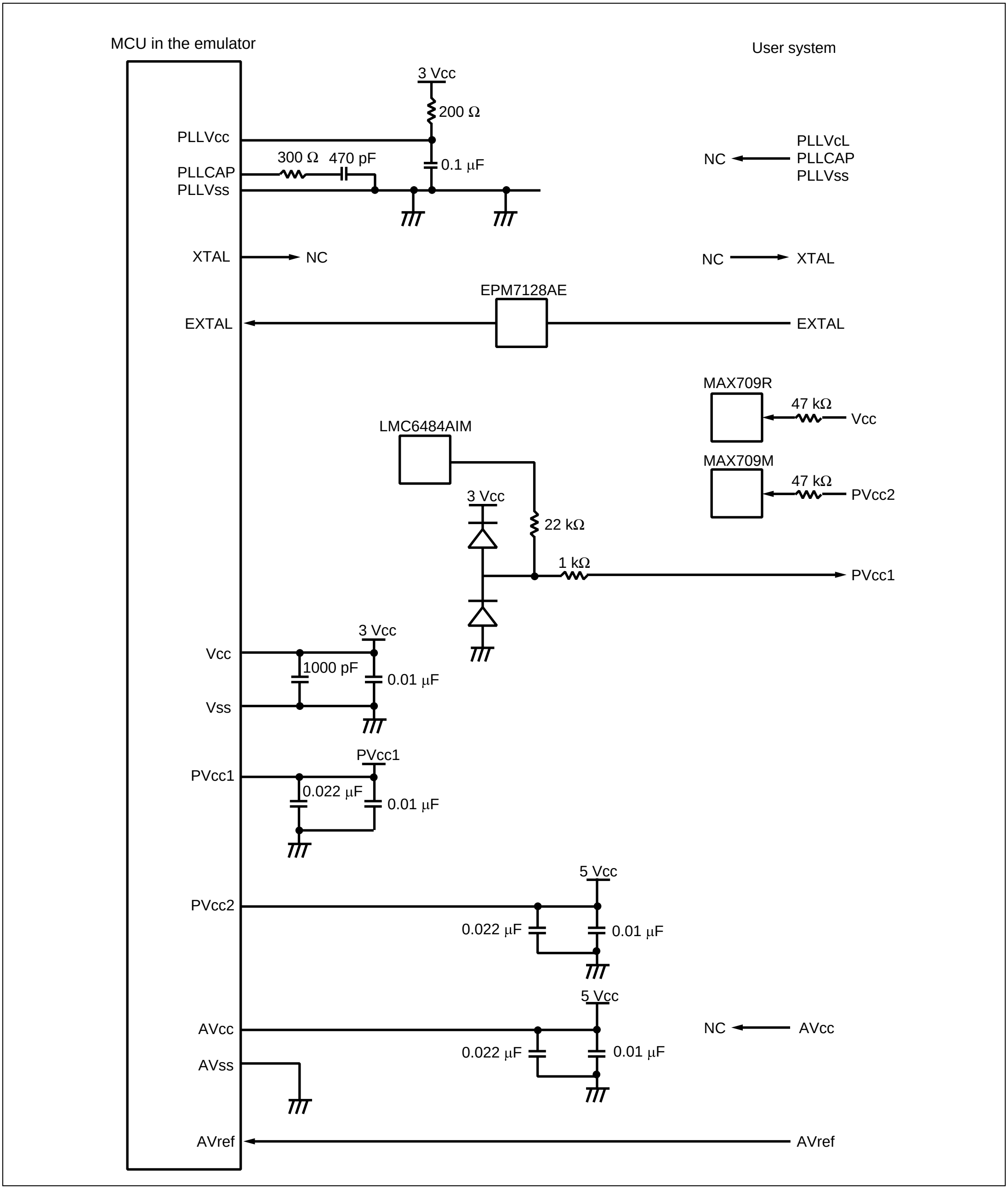



**Figure A.2 User System Interface Circuits (4)**

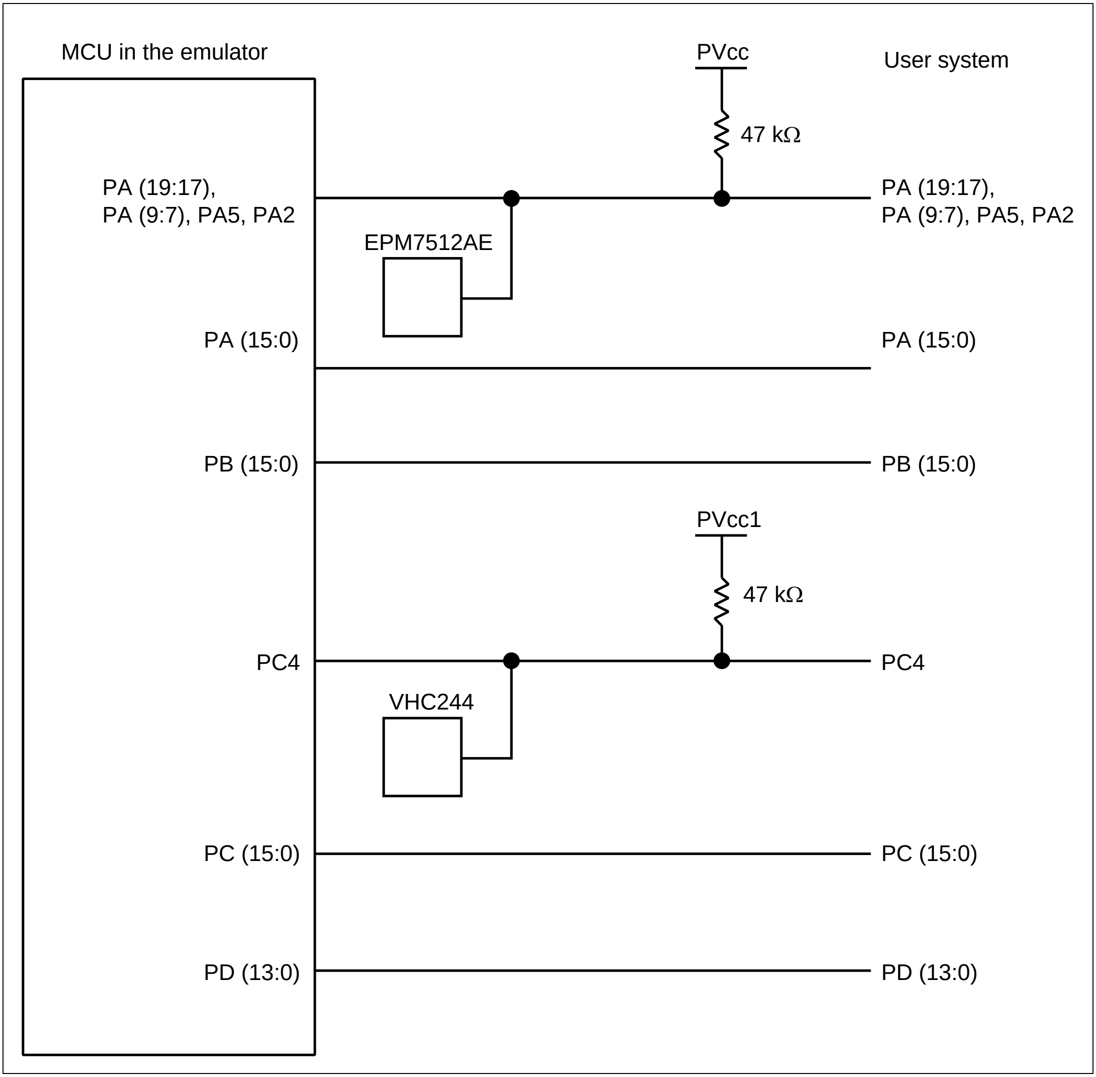



**Figure A.2 User System Interface Circuits (5)**

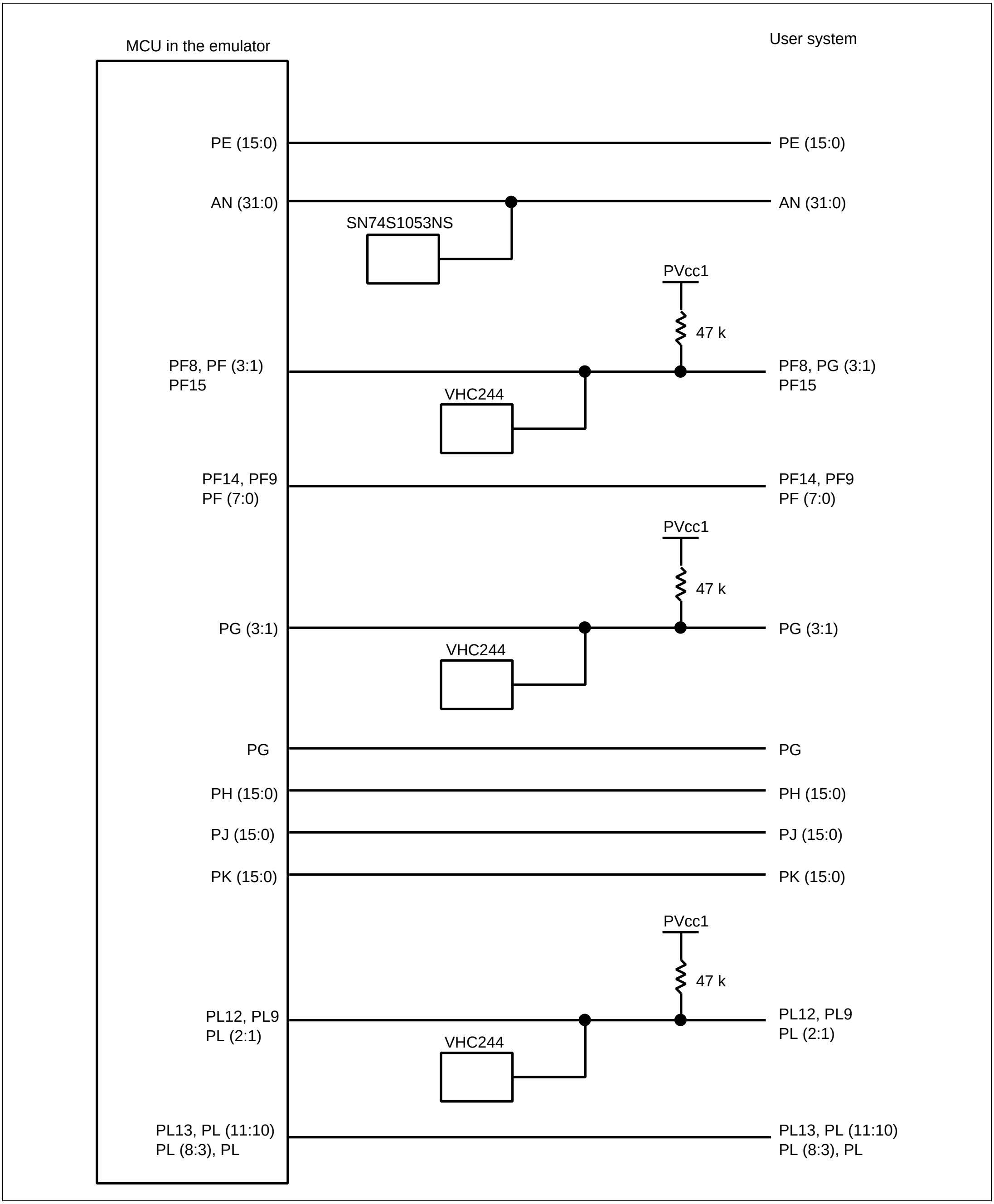



**Figure A.2 User System Interface Circuits (6)**

# Appendix B Emulator External Dimensions and Mass

Figures B.1 shows the external dimensions and mass of the E6000H emulator.

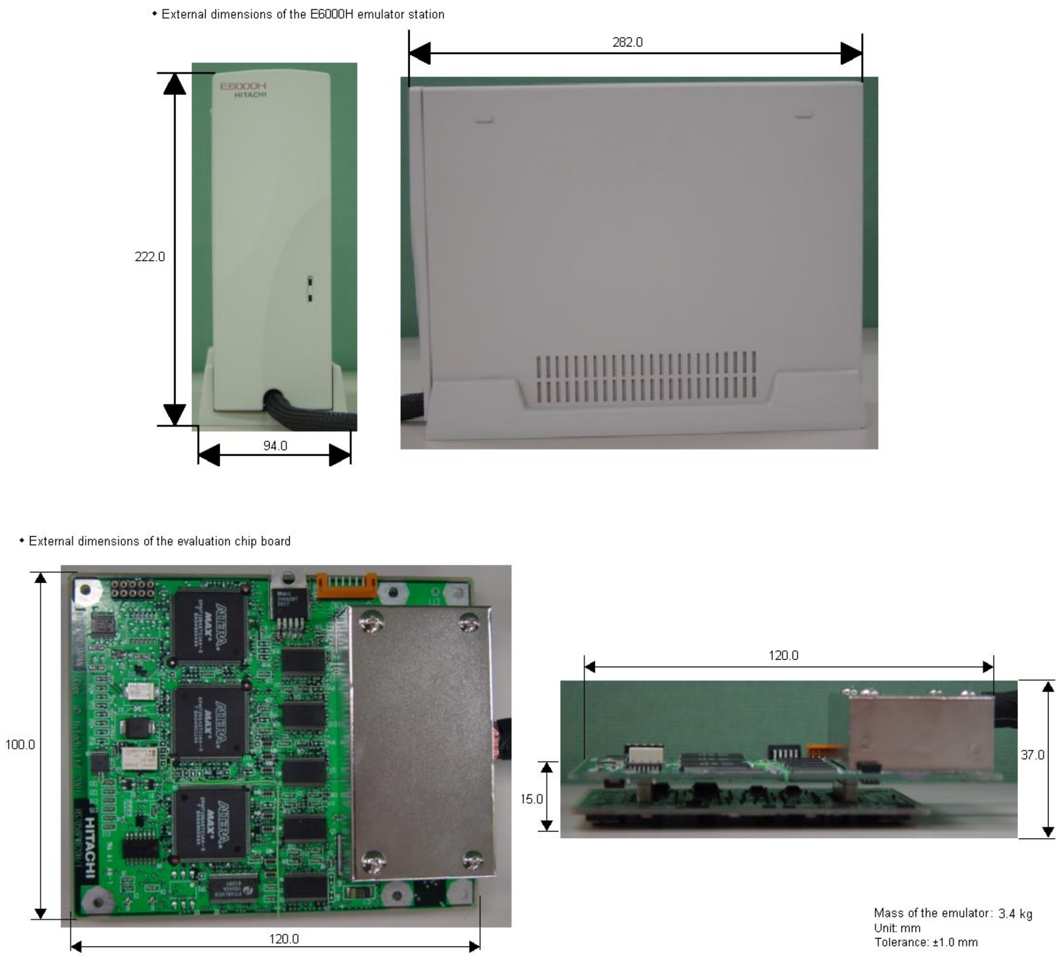



**Figure B.1 External Dimensions and Mass of the Emulator**

# Appendix C Connecting the Emulator to the User System

## C.1 Connecting to 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 or emulator or result in PERSONAL INJURY. Also, the USER PROGRAM will be LOST.**

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

**Table C.1 User System Interface Board and User Interfaces**

| User System Interface Board | User Interface |
|---|---|
| HS7058ECF61H | FP-256H (TQPACK256RD)* |

Note: The TQPACK series is manufactured by Tokyo Eletech Corporation.

### C.1.1 Installing IC Socket

1. Installing IC Socket

Install the IC socket for an FP-256H 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.)

### C.1.2 Connection Using the HS7058ECF61H

> **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 HS7058ECF61H, 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 (TQPACK256RD).

Install the FP-256H pin socket (TQPACK256RD 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 C.1 shows the connection of the HS7058ECF61H, figure C.2 shows the size restrictions for the installed components of the HS7058ECF61H, and figure C.3 shows the recommended mount pad dimensions of the user system IC socket.

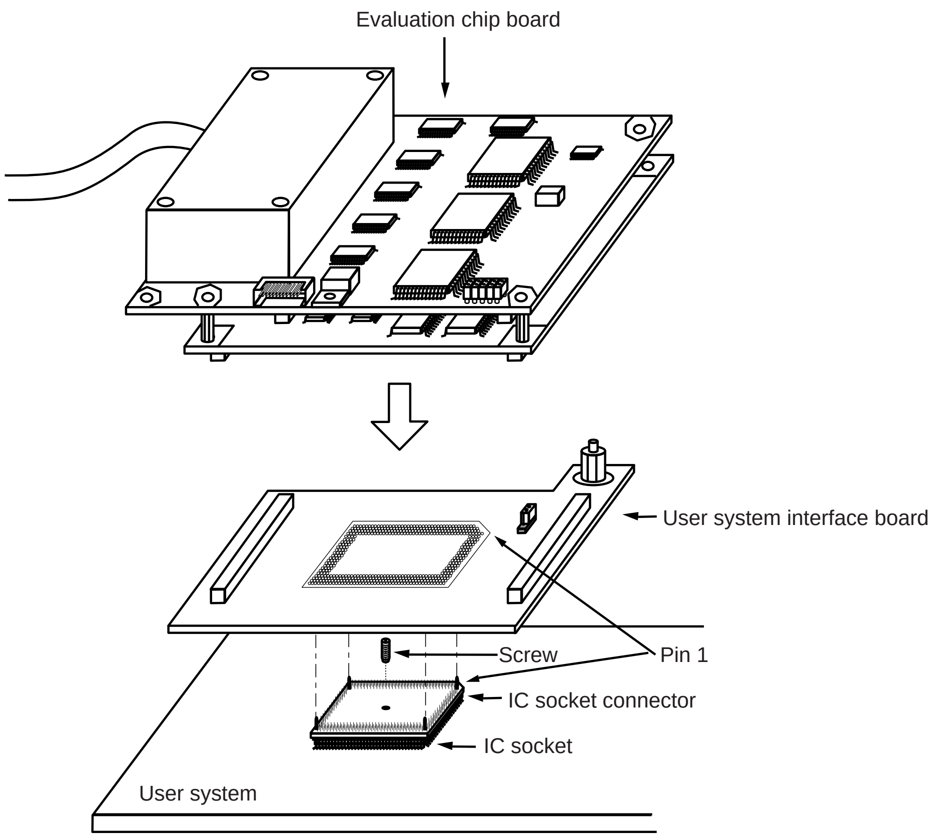



**Figure C.1 Connection Using the HS7058ECF61H**

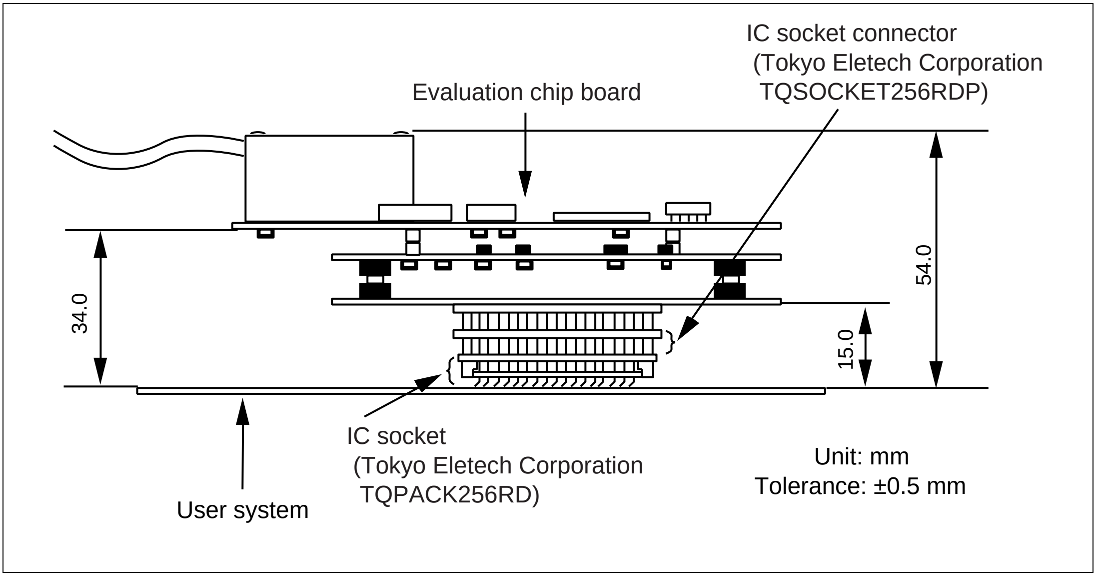



**Figure C.2 Restrictions on Component Installation**

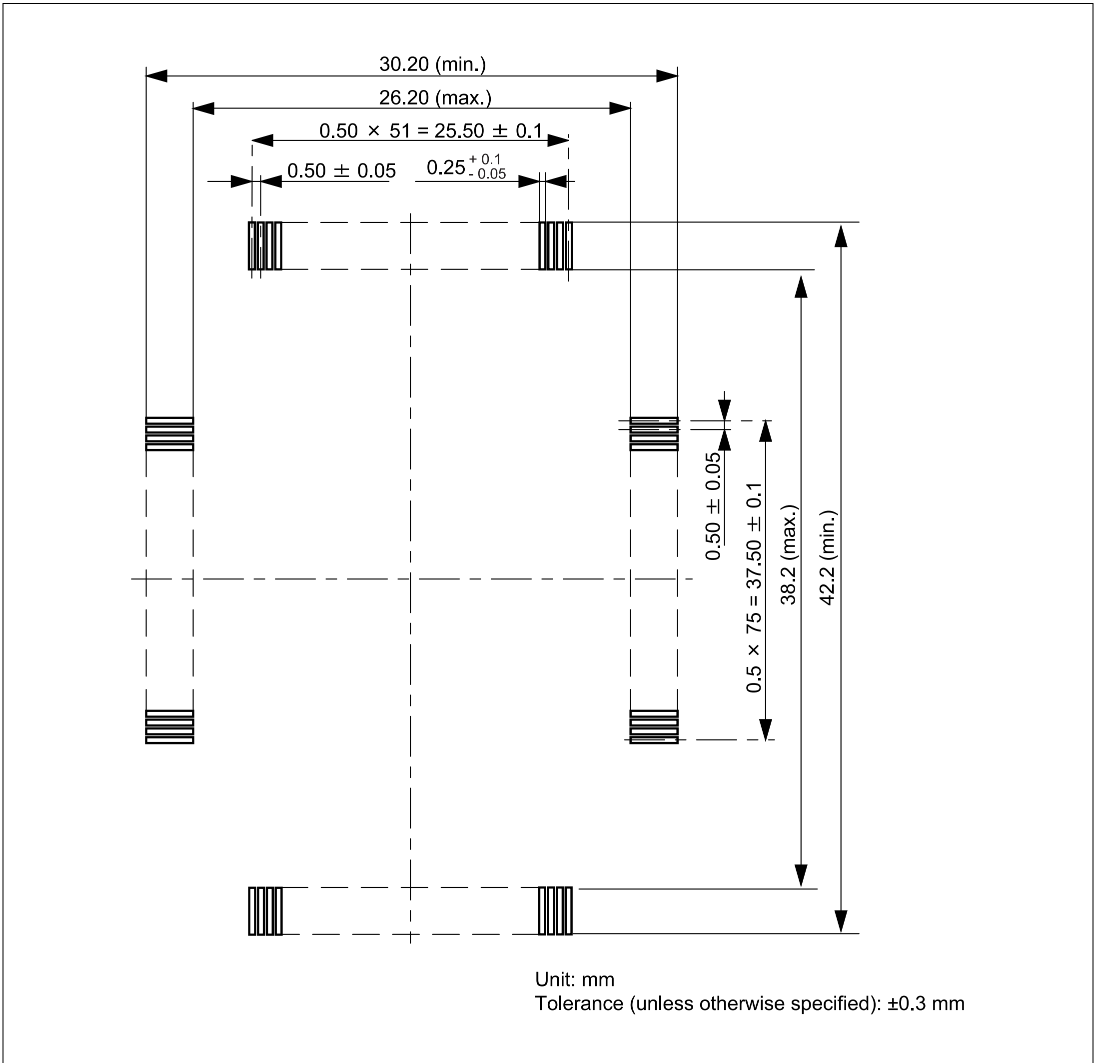



**Figure C.3 Recommended Mount Pad Dimensions of the User System IC Socket**

### C.1.3 Connection Using the Dedicated Connector

| **WARNING** |
| --- |
| **Always switch OFF the emulator and user system before connecting or disconnecting any CABLES.**<br>**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.**<br>**For correct connection, check the location of pin 1.** |

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 C.4 to C.6 show connection using the dedicated connector, size restrictions for the installed components, and the location for mounting the connector in the user system, respectively.

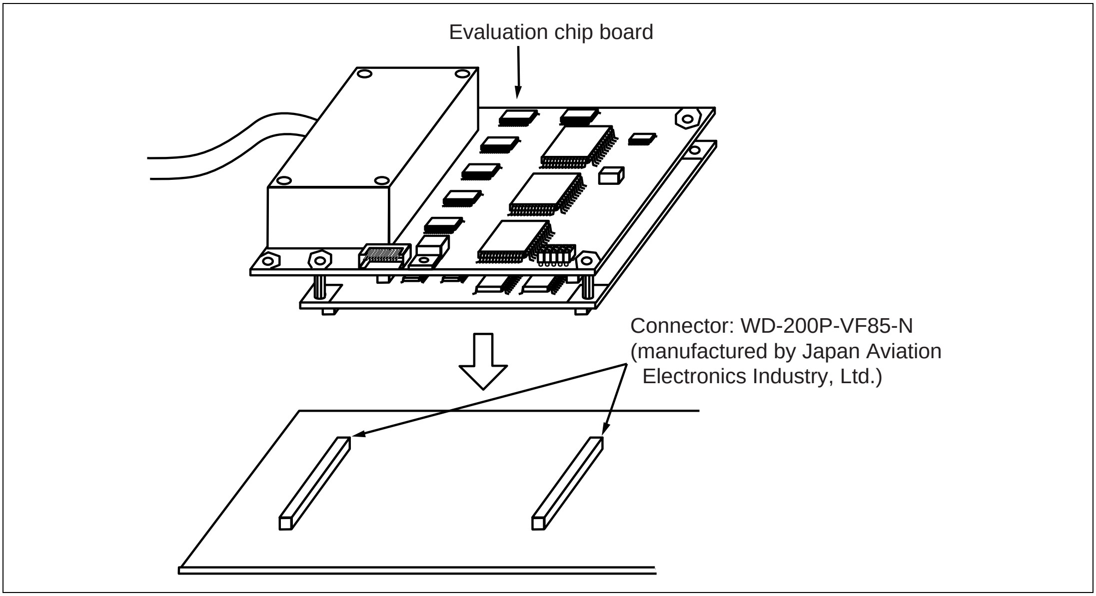



**Figure C.4 Connection Using the Dedicated Connector**

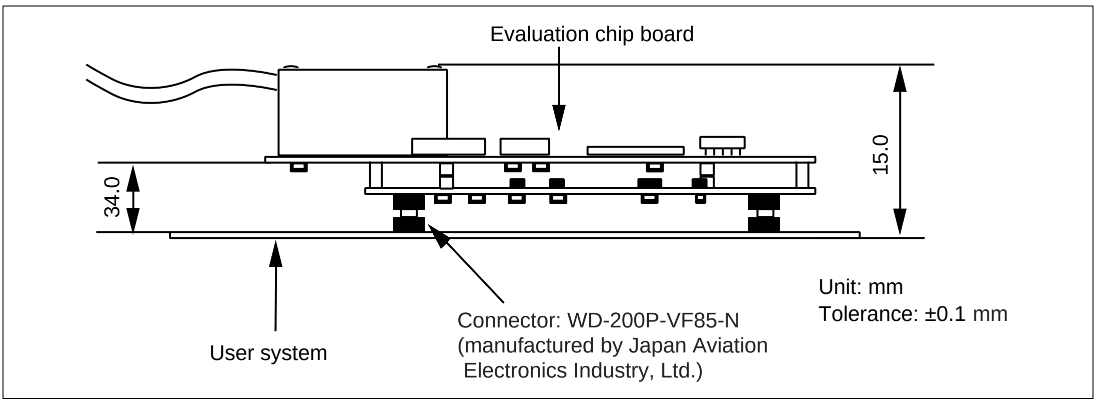



**Figure C.5 Size Restrictions for the Installed Components**

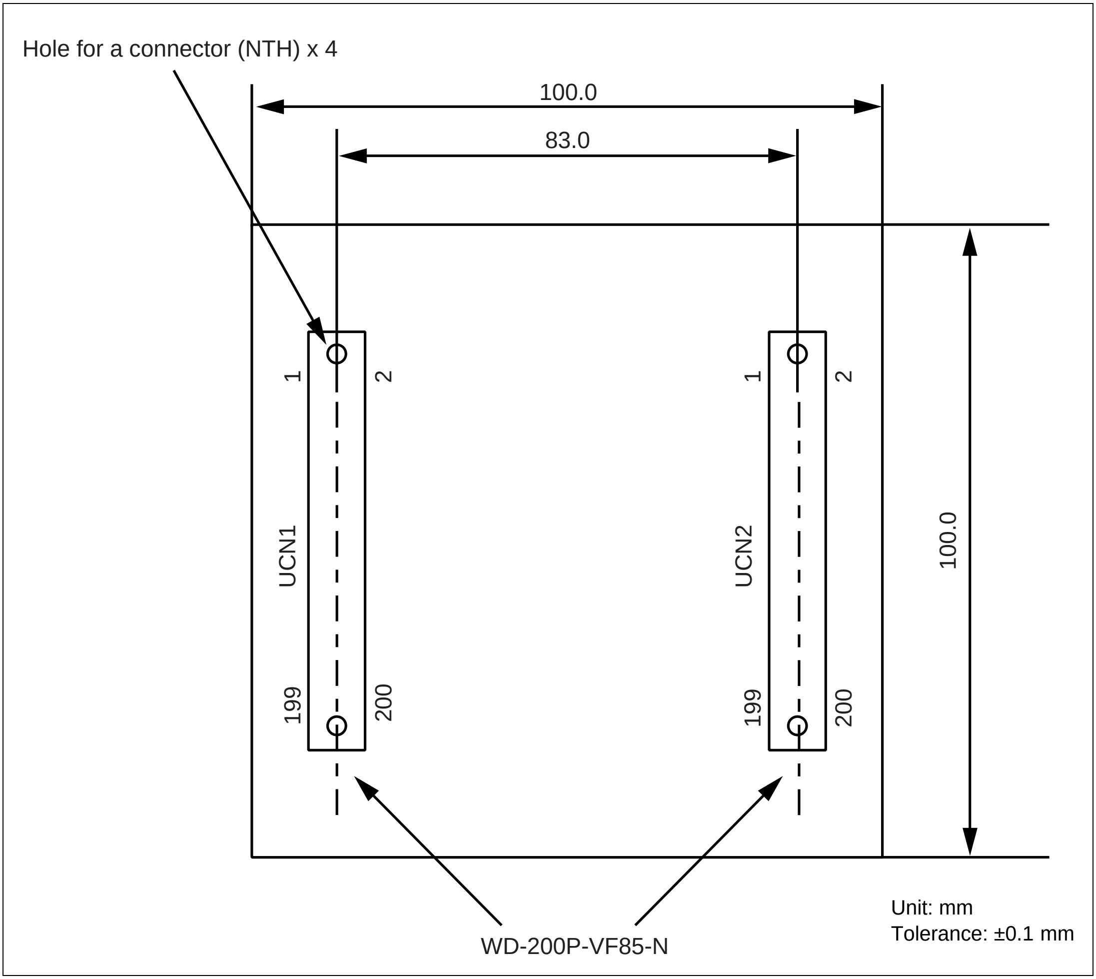



**Figure C.6 Location for Mounting the Connector in the User System**

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

## C.2 Pin Arrangement on the User System Interface Connector

Table C.2 lists the pin arrangement on the user system interface connector of HS7058EPH60H.

**Table C.2 Pin Arrangement on HS7058EPH60H**

| User I/F 1 | Pin No. | Signal Name | User I/F 1 | Pin No. | Signal Name |
|---|---|---|---|---|---|
| UCN1 | 1 | GND | UCN1 | 25 | PF8/_WAIT |
| | 2 | GND | | 26 | GND |
| | 3 | GND | | 27 | GND |
| | 4 | GND | | 28 | PE7/A7 |
| | 5 | GND | | 29 | PF7/_WRH |
| | 6 | GND | | 30 | PE6/A6 |
| | 7 | GND (TGBON1) | | 31 | PF6/_WRL |
| | 8 | PE15/A15 | | 32 | PE5/A5 |
| | 9 | PF15/_BREQ | | 33 | PF5/A21/_POD |
| | 10 | PE14/A14 | | 34 | PE4/A4 |
| | 11 | PF14/_BACK | | 35 | PF4/A20 |
| | 12 | PE13/A13 | | 36 | GND |
| | 13 | PF13/_CS3 | | 37 | GND |
| | 14 | PE12/A12 | | 38 | PE3/A3 |
| | 15 | PF12/_CS2 | | 39 | PF3/A19 |
| | 16 | GND | | 40 | PE2/A2 |
| | 17 | GND | | 41 | PF2/A18 |
| | 18 | PE11/A11 | | 42 | PE1/A1 |
| | 19 | PF11/_CS1 | | 43 | PF1/A17 |
| | 20 | PE10/A10 | | 44 | PE0/A0 |
| | 21 | PF10/_CS0 | | 45 | PF0/A16 |
| | 22 | PE9/A9 | | 46 | GND |
| | 23 | PF9/_RD | | 47 | GND |
| | 24 | PE8/A8 | | 48 | PD13/PULS6/<br>HRxD0/HTxD1 |

Table C.2 Pin Arrangement on HS7058EPH60H (cont)

| User I/F 1 | Pin No. | Signal Name | User I/F 1 | Pin No. | Signal Name |
|---|---|---|---|---|---|
| UCN1 | 49 | GND | UCN1 | 75 | GND |
| | 50 | PD12/PULS4 | | 76 | PD2/TIO1C |
| | 51 | GND | | 77 | GND |
| | 52 | GND | | 78 | PD1/TIO1B |
| | 53 | GND | | 79 | GND |
| | 54 | PD11/PULS3 | | 80 | PD0/TIO1A |
| | 55 | GND | | 81 | AUDCK |
| | 56 | PD10/PULS2 | | 82 | GND |
| | 57 | GND | | 83 | AUDMD |
| | 58 | PD9/PULS1 | | 84 | PL13/_IRQOUT |
| | 59 | GND | | 85 | _AUDSYNC |
| | 60 | PD8/PULS0 | | 86 | PL12/_IRQ4 |
| | 61 | GND | | 87 | _AUDRST |
| | 62 | GND | | 88 | GND |
| | 63 | GND | | 89 | GND |
| | 64 | PD7/TIO1H | | 90 | PL11/HRxD/HRxD1/ HRxD0&HRxD1 |
| | 65 | GND | | 91 | AUDATA3 |
| | 66 | PD6/TIO1G | | 92 | PL10/HTxD0/ HTxD1/HTxD0&HTxD 1 |
| | 67 | GND | | 93 | AUDATA2 |
| | 68 | PD5/TIO1F | | 94 | PL9/SCK4/_IRQ5 |
| | 69 | GND | | 95 | AUDATA1 |
| | 70 | PD4/TIO1E | | 96 | PL8/SCK3 |
| | 71 | GND | | 97 | AUDATA0 |
| | 72 | GND | | 98 | GND |
| | 73 | GND | | 99 | N.C. |
| | 74 | PD3/TIO1D | | 100 | PL7/SCK2 |

**Table C.2 Pin Arrangement on HS7058EPH60H (cont)**

| User I/F 1 | Pin No. | Signal Name | User I/F 1 | Pin No. | Signal Name |
|---|---|---|---|---|---|
| UCN1 | 101 | 3Vcc | UCN1 | 127 | GND |
| | 102 | PL6/ADEND | | 128 | GND |
| | 103 | 3Vcc | | 129 | GND |
| | 104 | PL5/_ADTRG1 | | 130 | PK11/TO8L |
| | 105 | 3Vcc | | 131 | GND |
| | 106 | PL4/_ADTRG0 | | 132 | PK10/TO8K |
| | 107 | 3Vcc | | 133 | GND |
| | 108 | GND | | 134 | PK9/TO8J |
| | 109 | 3Vcc | | 135 | GND |
| | 110 | PL3/TCLKB | | 136 | PK8/TO8I |
| | 111 | 5Vcc | | 137 | N.C. |
| | 112 | PL2/TIO11B/_IRQ7 | | 138 | GND |
| | 113 | 5Vcc | | 139 | N.C. |
| | 114 | PL1/TIO11A/_IRQ6 | | 140 | PK7/TO8H |
| | 115 | 5Vcc | | 141 | N.C. |
| | 116 | PL0/TI10 | | 142 | PK6/TO8G |
| | 117 | 5Vcc | | 143 | N.C. |
| | 118 | GND | | 144 | PK5/TO8F |
| | 119 | 5Vcc | | 145 | N.C. |
| | 120 | PK15/TO8P | | 146 | PK4/TO8E |
| | 121 | 5Vcc | | 147 | N.C. |
| | 122 | PK14/TO8O | | 148 | GND |
| | 123 | 5Vcc | | 149 | N.C. |
| | 124 | PK13/TO8N | | 150 | PK3/TO8D |
| | 125 | GND | | 151 | N.C. |
| | 126 | PK12/TO8M | | 152 | PK2/TO8C |

**Table C.2 Pin Arrangement on HS7058EPH60H (cont)**

| User I/F 1 | Pin No. | Signal Name | User I/F 1 | Pin No. | Signal Name |
|---|---|---|---|---|---|
| UCN1 | 153 | N.C. | UCN1 | 177 | GND |
| | 154 | PK1/TO8B | | 178 | GND |
| | 155 | GND | | 179 | GND |
| | 156 | PK0/TO8A | | 180 | PJ7/TIO2H |
| | 157 | GND | | 181 | GND |
| | 158 | GND | | 182 | PJ6/TIO2G |
| | 159 | GND | | 183 | GND |
| | 160 | PJ15/TI9F | | 184 | PJ5/TIO2F |
| | 161 | GND | | 185 | GND |
| | 162 | PJ14/TI9E | | 186 | PJ4/TIO2E |
| | 163 | GND | | 187 | PG3/_IRQ3/ _ADTRG0 |
| | 164 | PJ13/TI9D | | 188 | GND |
| | 165 | GND | | 189 | PG2/_IRQ2/ADEND |
| | 166 | PJ12/TI9C | | 190 | PJ3/TIO2D |
| | 167 | GND | | 191 | PG1/_IRQ1 |
| | 168 | GND | | 192 | PJ2/TIO2C |
| | 169 | GND | | 193 | PG0/PULS7/HRxD0/ HRxD1 |
| | 170 | PJ11/TI9B | | 194 | PJ1/TIO2B |
| | 171 | GND | | 195 | GND |
| | 172 | PJ10/TI9A | | 196 | PJ0/TIO2A |
| | 173 | GND | | 197 | GND |
| | 174 | PJ9/TIO5D | | 198 | GND |
| | 175 | GND | | 199 | GND |
| | 176 | PJ8TIO5C | | 200 | GND |

**Table C.2 Pin Arrangement on HS7058EPH60H (cont)**

| User I/F 2 | Pin No. | Signal Name | User I/F 2 | Pin No. | Signal Name |
|---|---|---|---|---|---|
| UCN2 | 1 | GND | UCN2 | 31 | PH9/D9 |
| | 2 | GND | | 32 | CK |
| | 3 | GND | | 33 | PH10/D10 |
| | 4 | GND | | 34 | GND |
| | 5 | GND | | 35 | PH11/D11 |
| | 6 | GND | | 36 | GND |
| | 7 | GND | | 37 | GND |
| | 8 | GND | | 38 | GND |
| | 9 | PH0/D0 | | 39 | PH12/D12 |
| | 10 | FWE | | 40 | GND |
| | 11 | PH1/D1 | | 41 | PH13/D13 |
| | 12 | MD2 | | 42 | GND |
| | 13 | PH2/D2 | | 43 | PH14/D14 |
| | 14 | MD1 | | 44 | GND |
| | 15 | PH3/D3 | | 45 | PH15/D15 |
| | 16 | MD0 | | 46 | GND |
| | 17 | GND | | 47 | GND |
| | 18 | GND | | 48 | Vcc |
| | 19 | PH4/D4 | | 49 | GND |
| | 20 | EXTAL | | 50 | Vcc |
| | 21 | PH5/D5 | | 51 | GND |
| | 22 | GND | | 52 | Vcc |
| | 23 | PH6/D6 | | 53 | GND |
| | 24 | _RES | | 54 | PVcc1 |
| | 25 | PH7/D7 | | 55 | GND |
| | 26 | GND | | 56 | PVcc1 |
| | 27 | GND | | 57 | GND |
| | 28 | _HSTBY | | 58 | PVcc1 |
| | 29 | PH8/D8 | | 59 | GND |
| | 30 | GND | | 60 | PVcc2 |

**Table C.2 Pin Arrangement on HS7058EPH60H (cont)**

| User I/F 2 | Pin No. | Signal Name | User I/F 2 | Pin No. | Signal Name |
|---|---|---|---|---|---|
| UCN2 | 61 | GND | UCN2 | 90 | N.C. |
| | 62 | PVcc2 | | 91 | GND |
| | 63 | AN0 | | 92 | N.C. |
| | 64 | PVcc2 | | 93 | AN12 |
| | 65 | AN1 | | 94 | N.C. |
| | 66 | GND | | 95 | AN13 |
| | 67 | AN2 | | 96 | N.C. |
| | 68 | GND | | 97 | AN14 |
| | 69 | AN3 | | 98 | N.C. |
| | 70 | GND | | 99 | AN15 |
| | 71 | GND | | 100 | N.C. |
| | 72 | GND | | 101 | GND |
| | 73 | AN4 | | 102 | AVcc |
| | 74 | GND | | 103 | AN16 |
| | 75 | AN5 | | 104 | AVcc |
| | 76 | GND | | 105 | AN17 |
| | 77 | AN6 | | 106 | AVcc |
| | 78 | N.C. | | 107 | AN18 |
| | 79 | AN7 | | 108 | AVcc |
| | 80 | NMI | | 109 | AN19 |
| | 81 | GND | | 110 | AVref |
| | 82 | N.C. | | 111 | GND |
| | 83 | AN8 | | 112 | AVref |
| | 84 | N.C. | | 113 | AN20 |
| | 85 | AN9 | | 114 | AVss |
| | 86 | N.C. | | 115 | AN21 |
| | 87 | AN10 | | 116 | AVss |
| | 88 | N.C. | | 117 | AN22 |
| | 89 | AN11 | | 118 | AVss |

### Table C.2 Pin Arrangement on HS7058EPH60H (cont)

| User I/F 2 | Pin No. | Signal Name | User I/F 2 | Pin No. | Signal Name |
|---|---|---|---|---|---|
| UCN2 | 119 | AN23 | UCN2 | 146 | GND |
| | 120 | AVss | | 147 | PA2/TI0C |
| | 121 | GND | | 148 | GND |
| | 122 | GND | | 149 | PA3/TI0D |
| | 123 | AN24 | | 150 | GND |
| | 124 | GND | | 151 | GND |
| | 125 | AN25 | | 152 | GND |
| | 126 | _WDTOVF | | 153 | PA4/TIO3A |
| | 127 | AN26 | | 154 | GND |
| | 128 | GND | | 155 | PA5/TIO3B |
| | 129 | AN27 | | 156 | PB0/TO6A |
| | 130 | GND | | 157 | PA6/TIO3C |
| | 131 | GND | | 158 | PB1/TO6B |
| | 132 | GND | | 159 | PA7/TIO3D |
| | 133 | AN28 | | 160 | PB2/TO6C |
| | 134 | GND | | 161 | GND |
| | 135 | AN29 | | 162 | PB3/TO6D |
| | 136 | GND | | 163 | PA8/TIO4A |
| | 137 | AN30 | | 164 | GND |
| | 138 | GND | | 165 | PA9/TIO4B |
| | 139 | AN31 | | 166 | PB4/TO7A/ TO8A |
| | 140 | GND | | 167 | PA10/TIO4C |
| | 141 | GND | | 168 | PB5/TO7B/ TO8B |
| | 142 | GND | | 169 | PA11/TIO4D |
| | 143 | PA0/TI0A | | 170 | PB6/TO7C/ TO8C |
| | 144 | GND | | 171 | GND |
| | 145 | PA1/TI0B | | 172 | PB7/TO7D/ TO8D |

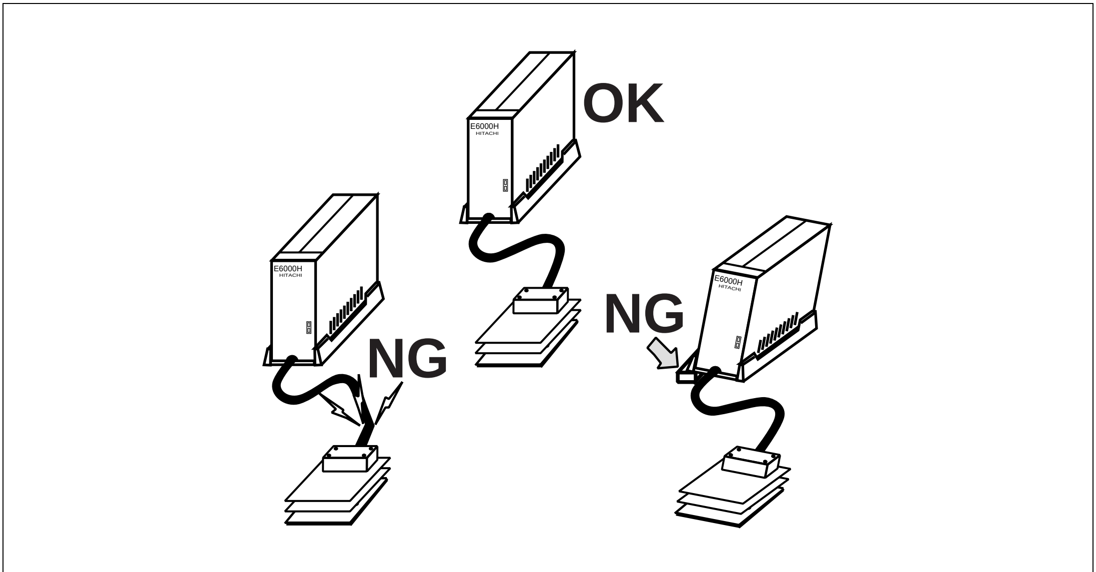
**Table C.2 Pin Arrangement on HS7058EPH60H (cont)**

| User I/F 2 | Pin No. | Signal Name | User I/F 2 | Pin No. | Signal Name |
|---|---|---|---|---|---|
| UCN2 | 173 | PA12/TIO5A | UCN2 | 187 | PC2/TxD2 |
| | 174 | GND | | 188 | PB13/SCK0 |
| | 175 | PA13/TIO5B | | 189 | PC3/RxD2 |
| | 176 | PB8/TxD3/ TO8E | | 190 | PB14/SCK1/ TCLKB/TI10 |
| | 177 | PA14/TxD0 | | 191 | PC4/_IRQ0 |
| | 178 | PB9/RxD3/ TO8F | | 192 | PB15/PULS5/ SCK2 |
| | 179 | PA15/RxD0 | | 193 | GND |
| | 180 | PB10/TxD4/ HTxD0/TO8G | | 194 | GND (TGBON2) |
| | 181 | GND | | 195 | GND |
| | 182 | PB11/RxD4/ HRxD0/TO8H | | 196 | GND |
| | 183 | PC0/TxD1 | | 197 | GND |
| | 184 | GND | | 198 | GND |
| | 185 | PC1/RxD1 | | 199 | GND |
| | 186 | PB12/TCLKA/ _UBCTRG | | 200 | GND |

## C.3 Precautions on Connecting the User System

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

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 Vcc, PVcc1, and PVcc2 to the user system power.
   The emulator monitors and determines whether the user system is turned on or off by the following Vcc pins:
   (a) Connecting the dedicated connector for HS7058EPH60H
   Vcc: Pins UCN2-48, UCN2-50, and UCN2-52
   PVcc1: Pins UCN2-54, UCN2-56, and UCN2-58
   PVcc2: Pins UCN2-60, UCN2-62, and UCN2-64
   (b) Connecting the user system interface board
   Vcc: Pins 11, 49, 52, 75, 139, 187, 203, and 237

PVcc1: Pins 20, 39, 70, and 83
PVcc2: Pins 128, 148, 172, 194, 212, and 247
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.

# Appendix D MCU Internal Module Support

## D.1 Memory Space

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

### D.1.1 Internal Flash Memory Area

- Access to the internal flash memory area

The emulator includes internal flash memory for the MCU. The internal flash memory is accessed in the mode where the internal flash memory exists. The internal flash memory area access differs between user program execution and the emulator commands.

Only read access is enabled during execution of the user program. A break does not occur if the internal flash memory area is written to.

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

The internal ROM area is accessed in one state.

### D.1.2 Internal I/O Area

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

### D.1.3 External Memory Area

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

### D.1.4 Emulation RAM Area

The emulator includes the emulation RAM (4 kbytes x 8 blocks) for RAM emulation of the internal flash memory. This emulation RAM (ERAM) can be used by overlapping with the internal flash memory address.

If an attempt is made to write the address that overlaps ERAM to the internal flash memory area, the address is written by ERAM and a break does not occur.

## D.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.

### D.2.1 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.

### D.2.2 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 E.7, Displaying and Modifying the Contents of Memory.

## D.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.

## D.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). The _BREQ and _WAIT signals are valid during emulation with the display and modification of memory contents, execution, and step execution. While emulation is being halted (break), the input of the _RES, _BREQ, or _WAIT signal to the MCU by the user system is not possible.

The input of the _RES, _BREQ, or _WAIT signal during execution or step execution can be disabled by a setting in the [Configuration] dialog box.

## D.5 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.

## D.6 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 MCU. At final debugging of the user system using the A/D converter, use the actual SH7058-series F-ZTAT microcomputer chip.

## D.7 Emulator Status and Internal Modules

Some internal modules do not operate when the emulator is in break mode. Table D.1 shows the relation between the emulator's state and operation of the internal modules.

**Table D.1 Emulator State and Operation of Internal Modules**

| Internal Module | Operation During Emulation Halted (Break) | Operation During Emulation (Execution or Step Execution) |
|---|---|---|
| UBC (user break controller) | No | Yes |
| DMAC (direct memory access controller) | Yes | Yes |
| ATU-II (advanced timer unit) | Yes | Yes |
| APC (advanced pulse controller) | Yes | Yes |
| WDT (watchdog timer) | No | Yes |
| CMT (compare-match timer) | Yes | Yes |
| SCI (serial communication interface) | Yes | Yes |
| HCAN2 (Hitachi controller area network 2) | Yes | Yes |
| A/D converter | Yes | Yes |
| AUD (advanced user debugger) | Yes*1 | Yes*1 |
| I/O port | Yes | Yes |
| H-UDI (Hitachi user debugging interface) | Not available*2 | Not available*2 |

Notes: 1. The AUD can be used by the user or the emulator function. Therefore, when the AUD is used by the emulator, it is not available for the user.

2. The user cannot use the H-UDI.

# Appendix E Notes on Debugging

## E.1 Notes on HDI

### E.1.1 Memory Compare Function

The emulator does not support the memory compare function, which is used by selecting [Compare] from the [Memory] menu.

### E.1.2 Source-Level Execution

- Step

  Even standard C libraries are executed. To return to a caller function, use the step out function. In a for or while statement, executing a single step does not move execution to the next line. To move to the next line, execute a step again.

### E.1.3 Watch

- Local variables at optimization

  Depending on the generated object code, local variables in a C source file that is compiled with the optimization option enabled will not be displayed correctly. Check the generated object code by displaying the [Disassemble] window.

- Variable name specification

  When a name other than a variable name, such as a symbol name or function name, is specified, no data is displayed.

  Example: The function name is main.

  main =

- Array display

  When array elements exceed 1000, elements from after 1000 will not be displayed.

### E.1.4 Symbol Description for Expression

When a symbol of the following condition is described in an expression, enclose the symbol name with '{' and '}'.

- When a symbol is defined by a load module and the symbol name includes a space.
  Example: {func( short, long)}

- When a symbol is registered in the [Labels] window and the symbol name includes a character other than alphanumeric characters.
  Example: {ASM_DATA1[10]}

### E.1.5 Register Function

The default input radix in the [Register] dialog box is hexadecimal irrespective of the radix setting. When a radix other than a hexadecimal is input, specify the prefix code such as B'.

### E.1.6 Session File Function

When the [Load last session on startup] check box in the [HDI Options] dialog box is valid and the HDI is activated, loading the session file can be stopped by clicking the [Stop] button during HDI activation.

### E.1.7 Command Line Window

- Command file
  To display the message “Not currently available” while executing a command file, enter the sleep command. Adjust the sleep time length which differs according to the operating environment.
  Example: To display “Not currently available” during MEMORY_FILL command execution:
  ```
  sleep d’3000
  memory_fill 0 ffff 0
  ```

- Overwriting a file
  A file having the same name as the output file is overwritten without asking the user.

- File specification by commands

  The current directory may be altered by file specifications in commands.  Absolute paths are recommended to be used to specify the files in a command file so that the current directory alteration is not affected.

  Example:    FILE_LOAD C:\\HEW\\ E6000H\\SH7058\\TUTORIAL\\TUTORIAL.ABS

### E.1.8 [I/O Registers] Window

- Watchdog timer

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

**Table E.1 Watchdog Timer Registers**

| Abbreviation | Read/Write | Register |
|---|---|---|
| TCNT_W | Write | Watchdog timer counter |
| TCNT_R | Read | Watchdog timer counter |
| TCSR_W | Write | Watchdog timer control/status register |
| TCSR_R | Read | Watchdog timer control/status register |
| RSTCSR_W | Write | Reset control/status register |
| RSTCSR_R | Read | Reset control/status register |

### E.1.9 Bit Field

The internal I/O-register definition file (SH7058.IO) provided with the emulator does not define bit fields.  The bit unit cannot be changed in the [I/O Registers] window.  When a bit-field definition is added to the I/O register file, the bit unit can be changed.  For details, refer to appendix E of the Hitachi Debugging Interface User's Manual provided on the CD-R.

### E.1.10 Line Assembly

Regardless of the radix setting, the default for line assembly input is decimal.  Specify H’ or 0x as the radix for a hexadecimal input.

### E.1.11 Usage with Another HDI

- Automatic load of session files

  If automatic load of session files is active in an environment with another HDI installed, the following error message is displayed when initiating this HDI and the program is not linked up.

  ```
  invalid target system: <recently used debugging platform name>
  ```

  In this case, change the target session file from [File] - [New Session…] or the [Load Session…] menu.

- Uninstallation of another HDI

  If another version of HDI is uninstalled after this HDI has been installed, the [Auto update Memory] and stack trace functions can not be used. In this case, reinstall this HDI.

### E.1.12 Operation During Accessing Files

Do not perform other operations during saving in the [Load Program], [Verify Memory], [Save Memory], or [Trace] window because this will not allow correct saving to be performed.

### E.1.13 Moving Source File Position after Creating Load Module

When the source file is moved after creating the load module, the [Open] dialog box may be displayed to specify the source file during the debugging of the created load module. Select the corresponding source file and click the [Open] button.

## E.2 User System Interface

The delay time is generated on the timing of the $\overline{\text{RES}}$ signal when it is input to the MCU from the user system, as shown in table E.2, because this connection for this signal is via logic circuit on the evaluation chip board.

**Table E.2 Delay Time for Signal Connected via the Evaluation Chip Board**

| Signal Name | Delay Time (ns) |
|---|---|
| $\overline{\text{RES}}$ | 15.0 |

## E.3 On-Emulator Break Functions

### E.3.1 Setting Address Bus and Data Bus Conditions

The address bus and data bus conditions are satisfied on 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 single bus cycles. A data condition is only valid for longword access when specified as longword. An address condition is only valid for a longword-access cycle when specified as a multiple of four.

- Word access

  Word data is read and written in single bus cycles. A data condition is only valid for word access when specified as word. Any multiple of two is a valid address condition.

- Byte access

  Byte data is read and written in single bus cycles. A data condition is only valid for byte access when specified as byte. Any address condition, whether an even or odd number, is valid.

## E.4 Sequential Trace Function

1. Six or more external bus cycles are required between each address condition that is specified for each channel.
2. When a user program is executed with the address of a PC address condition specified, a sequential break or a sequential trace stop may not operate correctly. Specify the address of the instruction following the address condition as the PC for the start of the execution, then execute the user program.

## E.5 Differences between the Emulator and the MCU

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

**Table E.3 Initial Values of Registers in the MCU and the Emulator**

| Register Name | Emulator: Power On | Emulator: Reset (Reset CPU) | MCU (Power-On Reset) |
|---|---|---|---|
| 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.

2. There is a delay on these signals since the user system interface circuit in the emulator includes pull-up resistors and buffers. Due to the pull-up resistors, the signals can be at a high level even when they are also in their high-impedance state. Take these points into account when preparing hardware for the user system. For details on the user system interface and the delays on signals, see appendix A, User System Interface Circuit.

## E.6 Step Function

During step execution, all break conditions are disabled.

## E.7 Displaying and Modifying the Contents of Memory

### E.7.1 Suspension of Program Execution: Reference Periods

Table E.4 shows the reference value for displaying and modifying the memory contents during user program execution.

**Table E.4 Suspension of Program Execution: Reference Values**

| Method of Access | Condition | | Period Suspended |
|---|---|---|---|
| Parallel access | Read | Reading a longword from the internal RAM | 320 ns |
| | Write | Writing a longword to the internal RAM | 320 ns |
| Auto update Memory | Execution of the user program does not stop | | |

Table E.5 shows the environment in which these measurements were obtained.

**Table E.5 Measurement Environment**

| Item | Setting |
|---|---|
| System clock ($\phi$) | 10.0 MHz |
| Clock mode | Clock mode 3 |
| H-UDI clock | 5.0 MHz |

### E.7.2 Parallel Access

The display of memory cannot be updated or modified by parallel access during standby mode.

### E.7.3 External Memory Area Access in Single-Chip Mode

In single-chip mode, do not access the external memory area (display or modify the memory contents) within the range of H’00200000 to H’7FFFFFFF in the memory window. An address error will occur.

# Appendix F Diagnostic Test Procedure

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

## F.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 (HS7058EPH60H)
- 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)

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® 98 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.

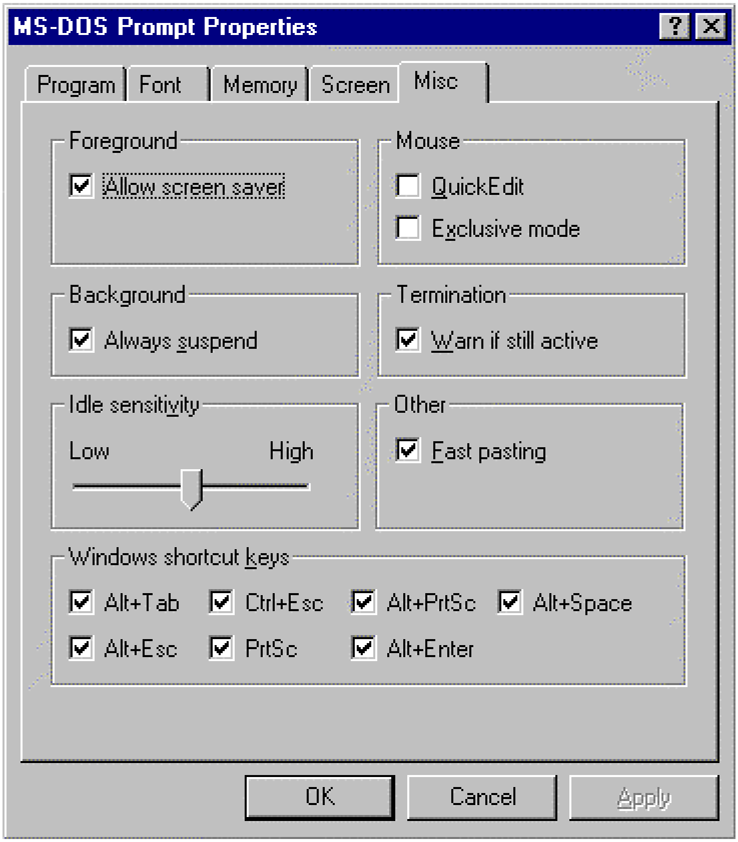



**Figure F.1 [Misc] Page**

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 HDI before executing the diagnostic program.

## F.2 Test Item of the Diagnostic Program

Table F.1 shows the test items of this 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 |

## F.3 Diagnostic Test Procedure Using the Diagnostic Program

Insert the CD-R (HS7058EPH60SR supplied with the E6000H) into the CD-ROM drive of the host computer by pressing the Shift key, 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)
   > TM7058 –PCI (RET)

2. PC card interface (HS6000EIP01H)
   > TM7058 –PCCD (RET)

3. LAN adapter (HS6000ELN01H)
   > TM7058 –ELN (RET)

4. USB adapter (HS6000EIU01H)
   > TM7058 –USB (RET)

The HDI 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\TM7058 –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 > TM7058 –PCI –S (RET), steps 1 to 13 will be repeatedly executed.  To stop the execution, enter Q.

Notes:
1. When the CD-R is inserted into the CD-ROM drive without pressing the Shift key, the HDI installation wizard is automatically started.
   In such a case, exit the HDI installation wizard.
2. <Drive> is a drive name for the CD-ROM drive.
3. Do not remove the CD-R from the CD-ROM drive during test program execution.

The following messages are displayed during the test. There are 13 steps in this test (when 166-MHz host computer and Windows® 98 PCI interface board are used, the time for the test will be about three minutes).

| Message | Description |
|---|---|
| `E6000H SH7058 Emulator Tests Vx.x`<br>`Copyright (c) 2002 Hitachi Ltd.` | 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:`<br>`Main board ID: H'0 Emulation board ID: H'001` | Shows the ID number of the E6000H emulator. |
| `Normal started at Tue Jul 16 20:47:41 2002` | 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) AUD Interface Registers Initial Value Check ............OK
     06) AUD Interface Registers Write/Verify ...................OK
```

```
    07) AUD Interface Registers Address Decode Test ............OK
 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
 4. Basic Function
    01) GO to BREAK ............................................OK
    02) RESET GO ...............................................OK
    03) STEP ...................................................OK
    04) KEYBREAK ...............................................OK
    05) BRKCONT ................................................OK
 5. GO to BREAK Time Measurement
    01) Counter Test Mode ......................................OK
    02) EMU 5MHz MPU 20MHz Sampling 20ns (Default) .............OK
    03) EMU 5MHz MPU 20MHz Sampling 1.6u .......................OK
    04) EMU 5MHz MPU 20MHz 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  5MHz MPU 20MHz Sampling 20ns ......................OK
    09) EMU 10MHz MPU 80MHz Sampling 20ns ......................OK
 6. Emulation Monitor
    01) AUDRES .................................................OK
    02) TRES ...................................................OK
    03) ASEST1 - 0 .............................................OK
    04) VCC3VNG ................................................OK
    05) VCC2-5VNG ..............................................OK
    06) VCC1-NG ................................................OK
 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 (ASEBM1 ASEAS2-0) ..............OK
    05) Control Signal Condition (ASEAS2-1) ....................OK
    06) Control Signal Condition (ASEIF-N) .....................OK
    07) Control Signal Condition (ASEERAM-N) ...................OK
    08) Control Signal Condition (/DMA_AUD ASEBM0) .............OK
    09) Control Signal Condition (ASEST1-0) ....................OK
 8. G/A Performance Analysis Function
    01) Time Measurement (20ns Sampling) .......................OK
 9. G/A Monitor Function
    01) STEP/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
11. G/A Trace Function
    01) Free Trace .............................................OK
    02) Trace Stop .............................................OK
    03) Time Stamp .............................................OK
    04) Trace Suppress .........................................OK
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) AUD Write (INROM BYTE) ..................................OK
     02) AUD Write (INROM WORD) ..................................OK
     03) AUD Write (INROM LONGWORD) ..............................OK
     04) AUD Write (INRAM BYTE) ..................................OK
     05) AUD Write (INRAM WORD) ..................................OK
     06) AUD Write (INRAM LONGWORD) ..............................OK
     07) AUD Sample (INROM BYTE) .................................OK
     08) AUD Sample (INROM WORD) .................................OK
     09) AUD Sample (INROM LONGWORD) .............................OK
     10) AUD Sample (INRAM BYTE) .................................OK
     11) AUD Sample (INRAM WORD) .................................OK
     12) AUD Sample (INRAM LONGWORD) .............................OK
     13) AUD Sample 256point .....................................OK
```

`Normal stopped at Tue Jul 16 20:50:18 2002`

Shows the time when the diagnostic program has ended.

`Tests run for 0H:2M:37S`

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)
```

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