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

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

RENESAS
Renesas Technology Corp.

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

SH7046 E8000S Emulator HS7046EBK81H

User's Manual

Renesas Electronics
www.renesas.com

Rev.1.0 2001.07

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
- Device control board
- Evaluation chip 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 HD64F7046 or HD64F7047 (hereafter referred to as MCU). By exchanging the device control board and evaluation chip board, this emulator product can also be used for systems using other microcomputers. 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 reserves the right to change, wholly or partially, the specifications, design, user's manual, and other documentation at any time without notice.

Target User of the Emulator Product:

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

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

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

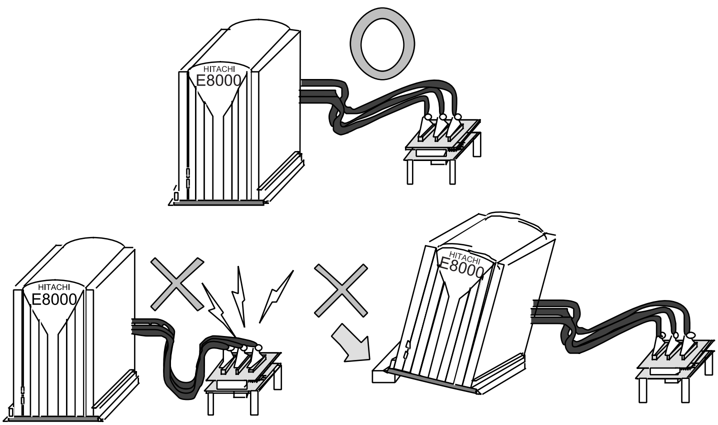
Warnings described below apply as long as you use this emulator. Be sure to read and understand the warnings below before using this emulator. Note that these are the main warnings, not the complete list.

⚠ 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.

⚠ WARNING

Place the emulator station and EV-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 SH7046.

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 SH7046 E8000S emulator (hereinafter referred to as the emulator) is an efficient software and hardware development tool for systems based on Hitachi microcomputer SH7046. By exchanging the device control board and the evaluation chip board, this emulator can also be used for systems using other microcomputers. The emulator is operated by using the Hitachi Debugging Interface (hereafter referred to as HDI). This interface program is supported by Windows® 95, Windows® 98, Windows® Me, WindowsNT® 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 E8000S Emulator is packaged with the evaluation chip board. For details, refer to section 3, Preparation before Use.

Related Manuals:

Description Notes on Using the PC Interface Board (HS6000EII01H)
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
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 WindowsNT® 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® 95 is an abbreviation for Microsoft® Windows® 95 operating system.
2. Windows® 98 is an abbreviation for Microsoft® Windows® 98 operating system.
3. Windows® Me is an abbreviation for Microsoft® Windows® Millenium Edition.
4. WindowsNT® 4.0 is an abbreviation for Microsoft® WindowsNT® 4.0 operating system.
5. Windows® 2000 is an abbreviation for Microsoft® Windows® 2000 operating system.

Contents

Figures

Tables

Section 1 Overview

1. Overview

This system is an efficient software and hardware development support tool for application systems using the HD64F7046 and HD64F7047 (hereafter referred to as SH7046) microcomputer developed by Hitachi, Ltd.

As peripheral functions for use with its high-speed CPU, the SH7046 incorporates an interrupt controller, user break controller, bus state controller, internal DMAC (direct memory access controller), data transfer controller, multi-function timer/pulse unit, motor-management timer, watchdog timer, comparison-match timer, serial communications interface (SCI), Hitachi controller area network, A/D converter, I/O ports, memory, etc.

The emulator operates in place of the SH7046 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 E8000S station, a device control board for the SH7046, an evaluation-chip board, and a user-system interface cable. The emulator is connected to the user system via the user-system interface cable.

By installing a PC interface board (available for ISA bus, PCI bus, and PCMCIA bus) on your host computer, the SH7046 E8000S Hitachi Debugging Interface (hereafter referred to as HDI) can be used for debugging. A LAN adapter allows the connection of the E8000S station to the host computer as a network. Connecting the LAN adapter also enables debugging using the HDI. For details on PC interface boards (available for ISA bus, PCI bus and PCMCIA bus specifications) and LAN adapter, refer to their description notes.

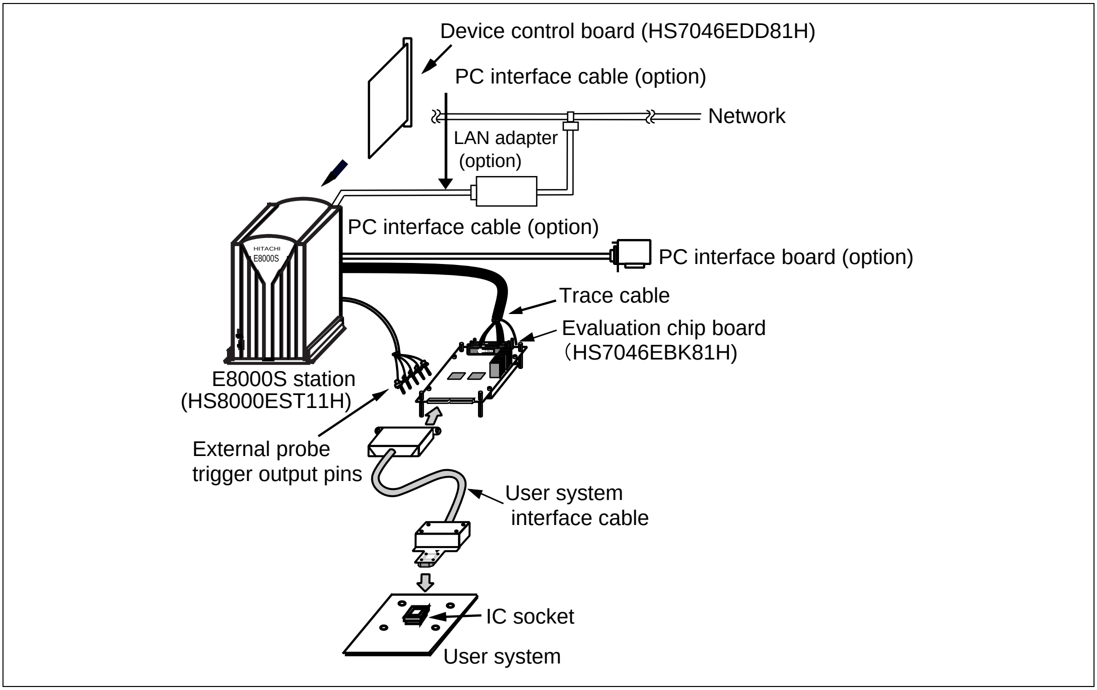


Figure 1.1 Emulator for the SH7046

The emulator provides the following features:

1. Realtime emulation of the MCU at 50 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. 4-Mbyte emulation memory for use as a substitute user-system memory installed as a standard item.
8. A LAN adapter for connecting the emulator to a host computer via a LAN interface (10BASE-T or 100BASE-TX), allowing loading of the SH7046 E8000S 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 ISA bus, PCI bus, or PCMCIA 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.6, 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:	200 VA
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® processors (300 MHz or faster is recommended)
Operating system	Windows® 95, Windows® 98, Windows® Me, WindowsNT® 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 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	ISA bus slot (not supported by Windows® Me and Windows® 2000), PCI bus slot, PC card (PCMCIA), and LAN adapter (conforming to IEEE802.3, with 10BASE-T or 100BASE-TX)
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® 95, Windows® 98, Windows® Me, WindowsNT® 4.0, and Windows® 2000
CD-ROM drive	Required for installation

1.3 Components

The emulator consists of the E8000S station, device control board, and evaluation chip board. 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 (HS8000EST11H)

Classification	Item	Quantity	Remarks
Hardware	E8000S station	1	
	Trace cable	1	CN1, CN2, CN3, with 4 cores
	AC power cable	1	
	Fuse	1	3 A, spare
Manual	Description Notes on using the HS8000EST11H	1	HS8000EST11HE

1.3.2 Device Control Board and Evaluation Chip Board for the SH7046

Tables 1.4 and 1.5 list the device control board and evaluation chip board components.

Table 1.4 Device Control Board Components for the SH7046

Classification	Item	Quantity	Remarks
Hardware	Device control board	1	One board, to be installed in the E8000S emulator station
	External probe	1	Probe input: 4 Run/break state output: 1 Trigger output: 1 GND: 2
Manual	Description Notes on using the HS7046EDD81H	1	HS7046EDD81HE

Table 1.5 Evaluation Chip Board Components for the SH7046

Item	Product No.	Quantity	Remarks
Hardware	Evaluation chip board	2	Two boards Connected to the user system by the user-system interface cable
Software	SH7046 E8000S emulator	1	CD-R HS7046EBK81SR
Additional documents	SH7046 E8000S Emulator Notes on Usage	1	HS7046EBK81HE-P(*)

Note: (*) indicates a manual revision.

1.3.3 Options

In addition to the E8000S station, the options listed in table 1.6 are also available. Refer to each option manual for details on these optional components.

Table 1.6 Optional Component Specifications

Item	Model Name	Specifications
User system interface cable for the SH7046 (FP-80Q)	HS7046ECH81H	For the SH7046 (FP-80Q)
User system interface cable for the SH7047 (FP-100M)	HS7047ECH81H	For the SH7047 (FP-100M)
PC interface board	HS6000EII01H	ISA bus
PCI interface board	HS6000EIC01H HS6000EIC02H	PCI bus
PC interface card	HS6000EIP01H	PCMCIA bus
LAN adapter	HS6000ELN01H	• TCP/IP communications protocol • 10BASE-T • 100BASE-Tx

Section 2 Components

2.1 Emulator Hardware Components

The emulator consists of an E8000S station, an SH7046 device control board, an SH7046 evaluation chip board, and a user system interface cable, as shown in figure 2.1. By installing a PC interface board (option: ISA bus, PCI bus, or PCMCIA bus specifications) on your host computer, the SH7046 E8000S Hitachi Debugging Interface (HDI) can be used for debugging. A LAN adapter can be connected to the emulator as a network, enabling debugging by the HDI. For details on the PC interface board and the LAN adapter, refer to the description notes on each product.

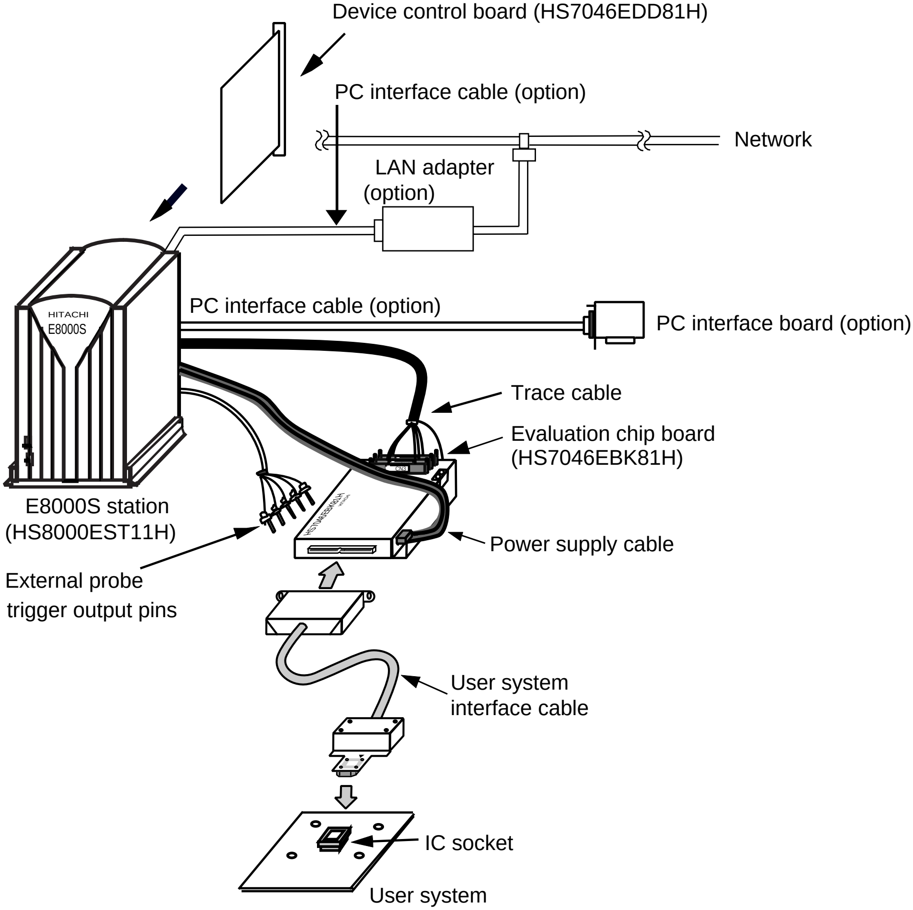


Figure 2.1 Emulator Hardware Components

2.1.1 E8000S Station Components

The names of the components on the front/rear panel of the E8000S station are listed below.

Front Panel:

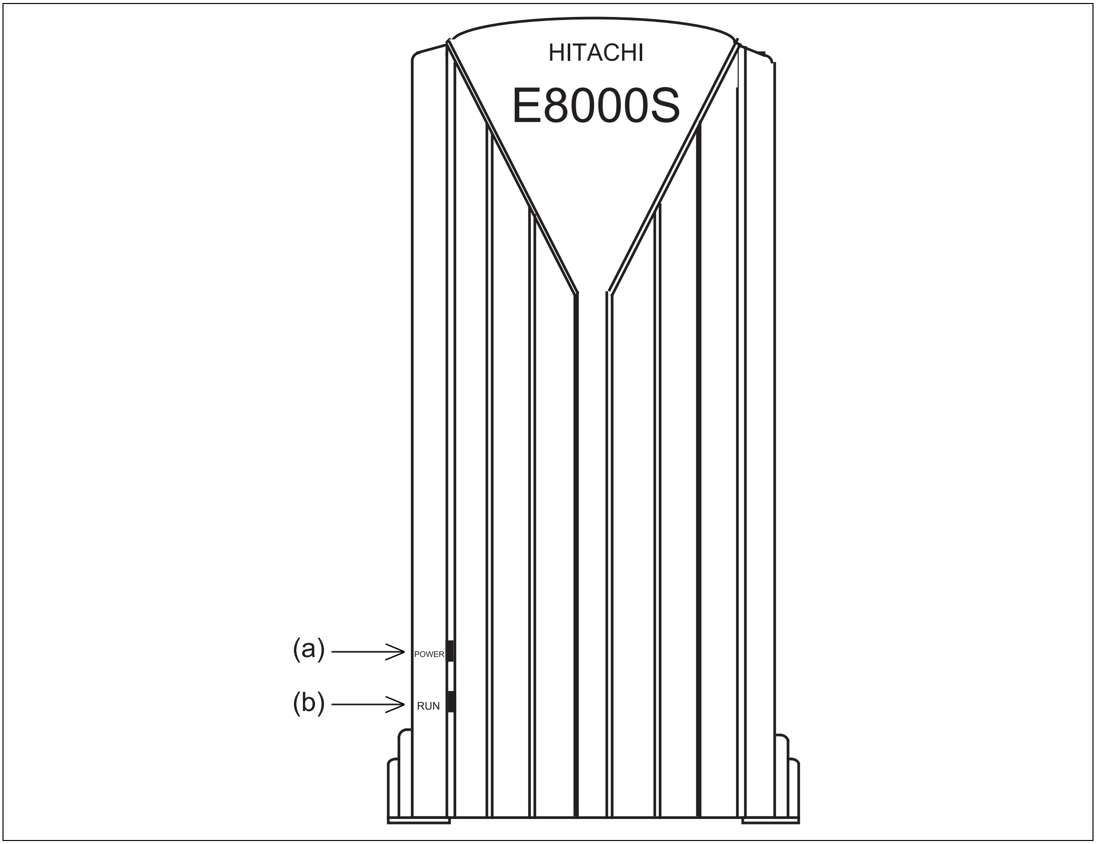


Figure 2.2 E8000S Station: Front Panel

1. POWER lamp: Is lit up while the E8000S station is supplied with power.
2. RUN lamp: Is lit up while the user program is running.

Rear Panel:

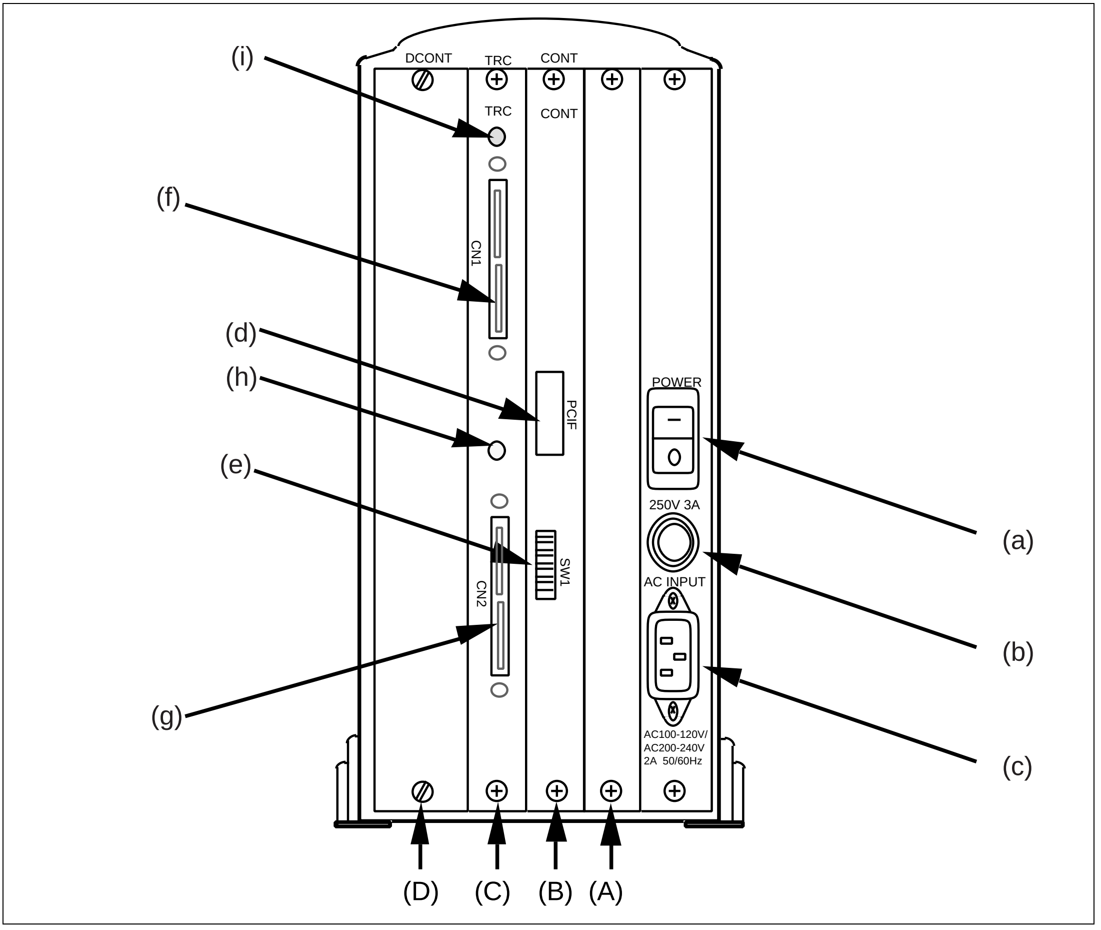


Figure 2.3 E8000S Station: Rear Panel

(A)	Optional board slot:	For installing the optional board (expansion slot).
(B)	Control board slot:	For installing the control board.
(C)	Trace board slot:	For installing the trace board.
(D)	Device control board slot:	For installing the device control board (depends on the target device).
(a)	Power switch:	Turning this switch to I (input) supplies power to the emulator (E8000S station and evaluation chip board).
(b)	Fuse box:	Contains an AC 100-V to 240-V power supply fuse (250 V, 3A).
(c)	AC power connector:	For an AC 100-V to 240-V power supply.
(d)	PC interface cable connector:	For the PC interface cable that connects the host computer to the E8000S station. A PC interface board (ISA bus, PCI bus, or PCMCIA bus) or LAN adapter can be connected. Marked PCIF.
(e)	Host interface switches:	For selecting the host interface. Do not change the settings. Marked SW1.
(f)	Station to evaluation chip board interface connector CN1:	For trace cable 1 that connects the E8000S station to the evaluation chip board.
(g)	Station to evaluation chip board interface connector CN2:	For trace cable 2 that connects the E8000S station to the evaluation chip board.
(h)	Trace cable mis-insertion inhibiting seal (CN2):	Prevents a trace cable from being inserted into the wrong place.
(i)	Trace cable mis-insertion inhibiting seal (CN1):	Prevents a trace cable from being inserted into the wrong place.

2.1.2 Device Control Board Components

The names of the components on the device control board (HS7046EDD81H) of the E8000S station are listed below.

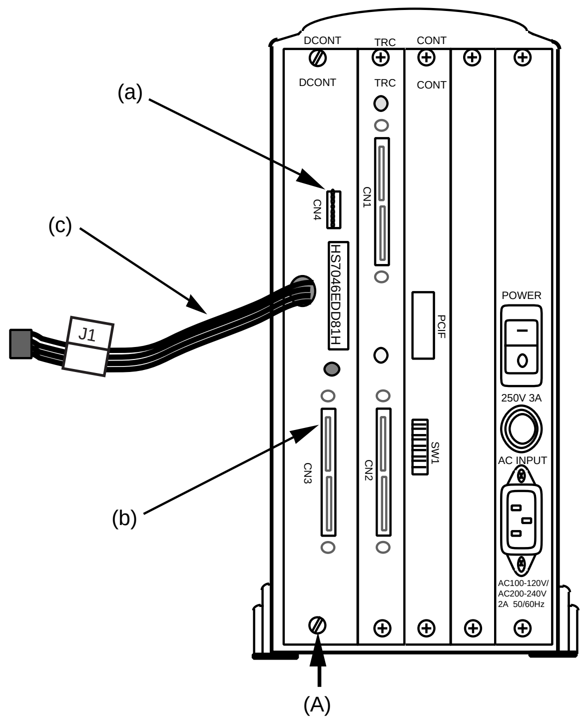


Figure 2.4 Device Control Board

(A)	Device control board slot:	For installing the device control board (depends on the target device).
(a)	External probe connector CN4:	For connecting the E8000S station to the external probe.
(b)	Station to evaluation chip board interface connector CN3:	For trace cable 3 that connects the E8000S station to the evaluation chip board.
(c)	Power supply cable J1:	For supplying power to the evaluation chip board.

2.1.3 Evaluation Chip Board Configuration

The names of the components on the evaluation chip board (HS7046EBK81H) of the emulator are listed below.

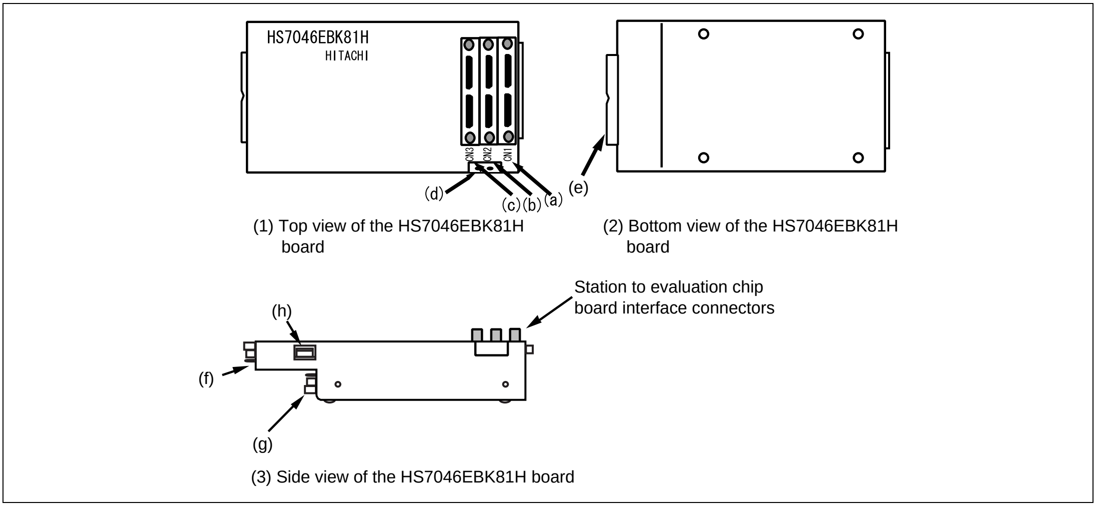


Figure 2.5 Evaluation Chip Board (HS7046EBK81H)

(a) Station to evaluation chip board interface connector CN1:	For trace cable 1 that connects the E8000S station to the evaluation chip board.
(b) Station to evaluation chip board interface connector CN2:	For trace cable 2 that connects the E8000S station to the evaluation chip board.
(c) Station to evaluation chip board interface connector CN3:	For trace cable 3 that connects the E8000S station to the evaluation chip board.
(d) Crystal oscillator terminals:	For installing a crystal oscillator to be used as an external clock source for the MCU.
(e) User system or interface cable connector:	For connecting the user system interface cable.
(f) HS7046PWB20H:	Dedicated connector to the user system interface cable is attached.
(g) HS7046PWB30H:	This is a dedicated board to the evaluation chip.
(h) Power supply connector J1	For connecting the device control board to the evaluation chip board.

Note: (a) to (h) listed above are referred to as HS7046EBK81H.

2.1.4 Configuration of User System Interface Cable

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

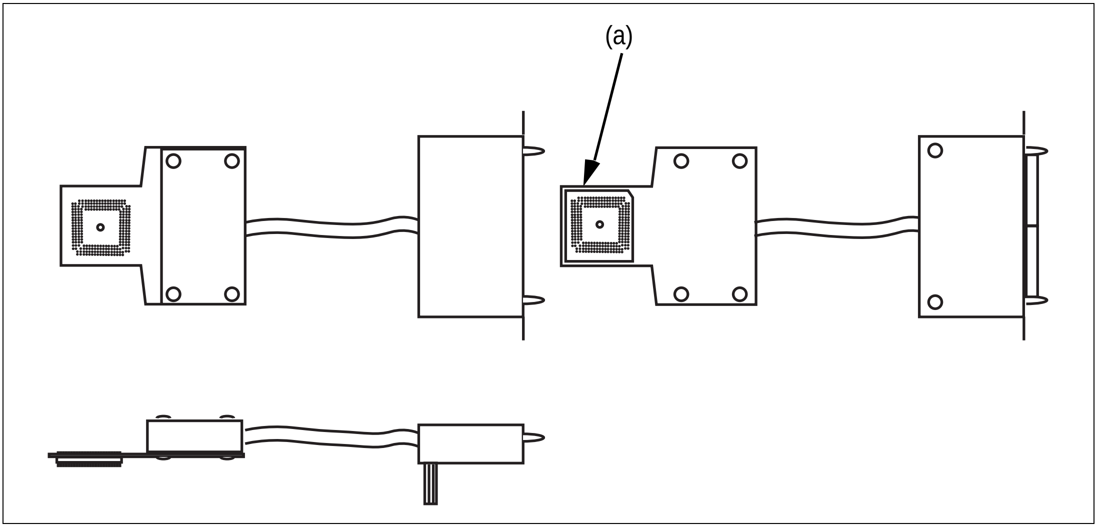


Figure 2.6 Configuration of User System Interface Cable

(a) 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 SH7046 E8000S 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 7046 in the root directory.

Table 2.1 Contents of CD-R

Directory Name	File Name	Contents	Note
	setup.exe	Installer	
\System	E8000.sys [*1]	System program for the SH7046 E8000S emulator	
\System	shcnf046.sys [*1]	Control program for the SH7046 E8000S emulator	
\System	shdct046.sys [*1]	Configuration file for the SH7046 E8000S emulator	
\System	diag.sys [*1]	Diagnostic and maintenance program	
\Drivers\Pci\95	pcihei.inf	Setup information (PCI)	For Windows® 95, Windows® 98, and Windows® Me
\Drivers\Pci\95	pcihei.vxd	Virtual driver (PCI)	For Windows® 95, Windows® 98, and Windows® Me
\Drivers\Pci\Nt	pcihei.sys	System file (PCI)	For Windows NT®
\Drivers\Pci\2000	pcihei.sys	System file (PCI)	For Windows® 2000
\Drivers\Pci\2000	pcihei2k.inf	Setup information (PCI)	For Windows® 2000
\Drivers\Isa\Nt	emulator.sys	System file (ISA)	For Windows NT®

Table 2.1 Contents of CD-R (cont)

Directory Name	File Name	Contents	Notes
\Drivers\Pcmcia\95	ulepcc.inf	Setup information (PCMCIA)	For Windows® 95 and Windows® 98
\Drivers\Pcmcia\95	ulepcc.vxd	Virtual driver (PCMCIA)	For Windows® 95 and 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 [*5]
\Manuals\Japanese	HS7046EBK81HJ.pdf [*2]	SH7046 E8000S emulator user's manual	PDF documents in Japanese [*5]
\Manuals\Japanese	HS7046TM81HJ(*)[*3].pdf [*2]	Descriptive notes on the diagnostic program for the SH7046 E8000S emulator	PDF documents in Japanese [*5]
\Manuals\English	HS6400DIIW5SE.pdf [*4]	Hitachi Debugging Interface user's manual	PDF documents in English [*5]
\Manuals\English	HS7046EBK81HE.pdf [*4]	SH7046 E8000S emulator user's manual	PDF documents in English [*5]
\Manuals\English	HS7046TM81HE(*)[*3].pdf [*4]	Descriptive notes on the diagnostic program for the SH7046 E8000S emulator	PDF documents in English [*5]
\Pdf_read\Japanese	Ar40jpn.exe	Acrobat® Reader 4.0 installer	Japanese version
\Pdf_read\English	Ar40eng.exe	Acrobat® Reader 4.0 installer	English version

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. (*) indicates a manual revision.
4. This is the English version of the manual. It cannot be installed by the Japanese-language version of the installer.
5. 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: ISA bus, PCI bus, or PCMCIA 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. For information on using the PC interface for ISA bus, PCI bus, or PCMCIA bus specification board or LAN adapter, refer to their description notes. Figure 2.7 shows the configuration of a system in which the PC interface board is used. Figure 2.8 shows the configuration of a system in which the LAN adapter is used.

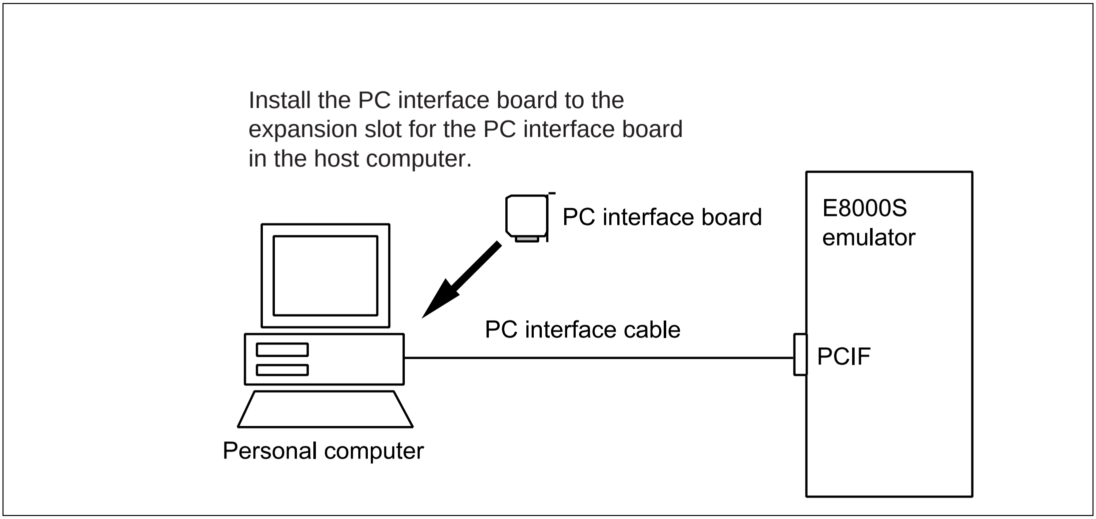


Figure 2.7 System Configuration Using a PC Interface Board

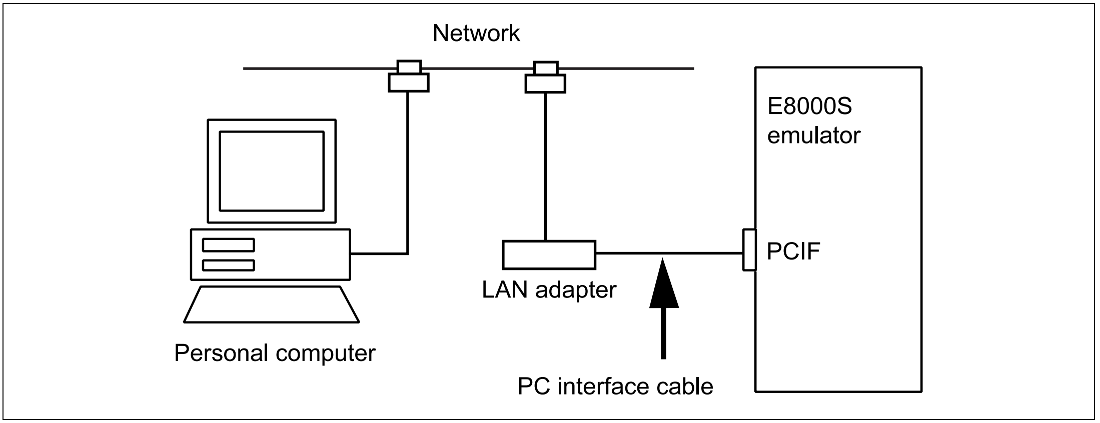


Figure 2.8 System Configuration Using a LAN Adapter

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.

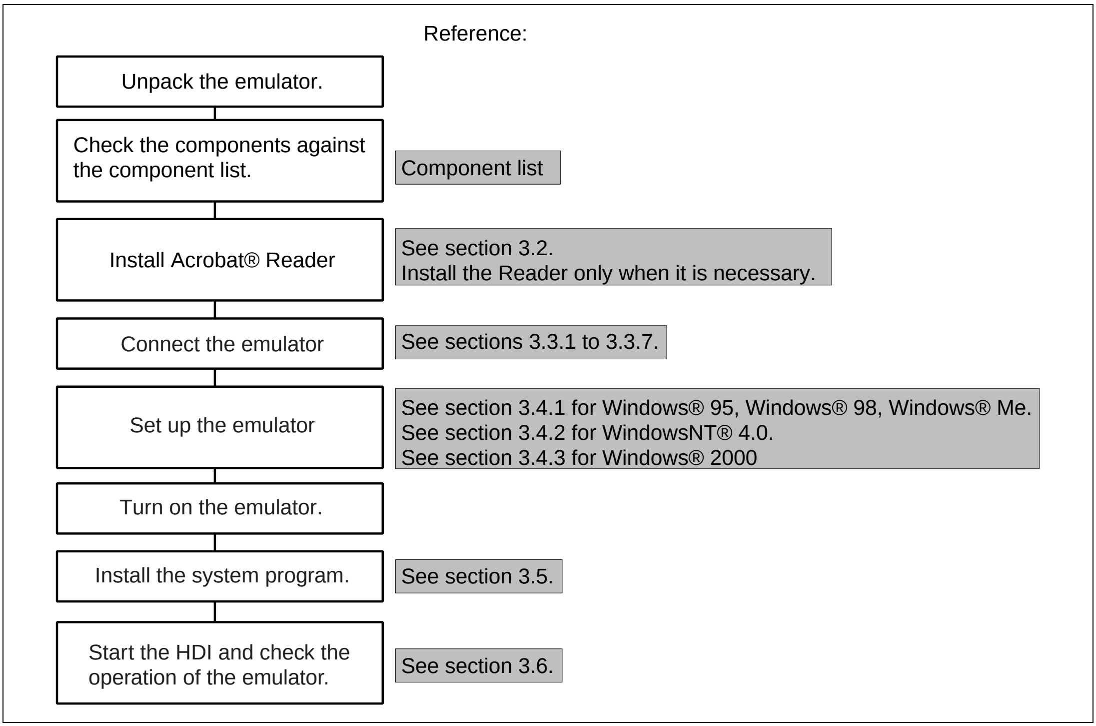


Figure 3.1 Emulator Preparation Flowchart

3.2 Installing the Acrobat® Reader

Acrobat® Reader 4.0 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 Ar40eng.exe in Pdf_Read\English directory in the [Run] dialog box (e.g. D:\7046\Pdf_Read\English\Ar40eng.exe) then click [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 Device Control Board

At shipment, the device control board is packed separately from the E8000S station. Connect the device control board to the E8000S station according to the following procedure. Also, use the following procedure to connect them after removing the device control board from the E8000S station to change the device control 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.

1. Check that the emulator power switch is turned off. Ensure that the power lamp on the left side of the E8000S station's front panel is not lit.
2. Remove the AC power cable of the E8000S station from the outlet (if the cable is connected to the outlet).
3. Remove the rear panel from the E8000S station. For the slot to which the device control board is to be connected, DCONT is marked.

4. Connect the device control board to the E8000S station. When connecting the board, prevent the upper or lower side of the board from lifting off the connector. Alternately tighten the screws on both sides of the board.

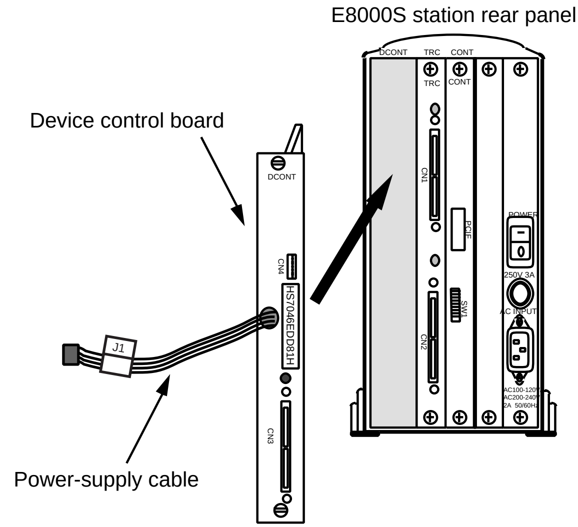


Figure 3.2 Connecting the Device Control Board

3.3.2 Connecting the Evaluation Chip Board

At shipment, the evaluation chip board is packed separately from the E8000S station. Use the following procedure to connect the evaluation chip board to the E8000S station, or to disconnect them when moving the emulator:

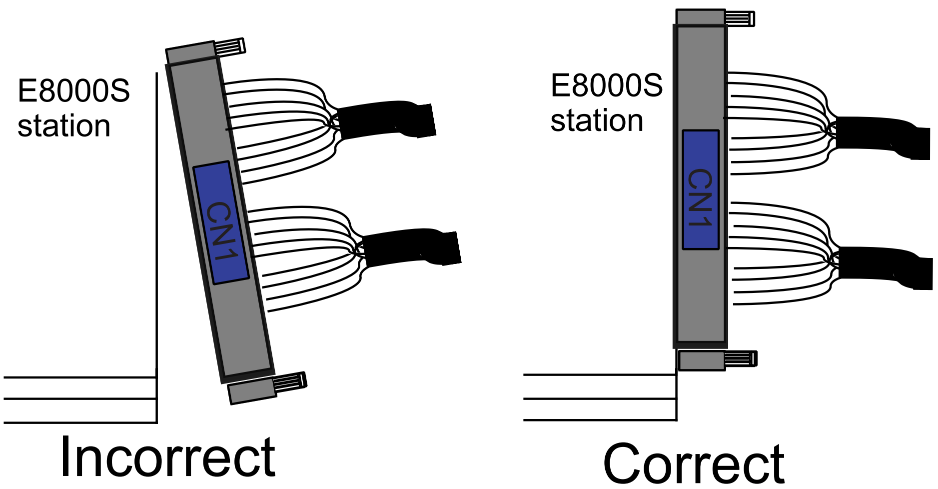
WARNING

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.

1. Check that the emulator power switch is turned off. Ensure that the power lamp on the left side of the E8000S station's front panel is not lit.
2. Remove the AC power cable of the E8000S station from the outlet (if the cable is connected to the outlet).

⚠WARNING

When connecting the cable, ensure that the upper or lower side of the cable does not lift off the connector. Alternately tighten the screws on both sides of the cable while gradually pushing the cable toward the connector. Failure to do so will result in a FIRE HAZARD, damage the user system and emulator, and will result in PERSONAL INJURY. The USER PROGRAM will be LOST.



CAUTION

At shipment, the trace cable screws are colored to prevent an insertion error (CN1: red, CN2: yellow, CN3: blue). If the connector is connected incorrectly, the connector will be damaged.

3. Connect the trace cables into the station to evaluation chip board interface connectors (CN1, CN2, and CN3) on the E8000S station's rear panel. Confirm that the shape of the trace-cable plug matches that of the station to evaluation chip board interface connector before connecting. Also note which trace cable is connected to which E8000S-station connector so that the other end of the trace cable is connected to the matching connector number on the evaluation chip board. After the connection is completed, alternately tighten the screws on both sides of the trace cable to prevent the upper or lower side of the trace cable from lifting off the connector. Figure 3.3 shows how to correctly connect the trace cables to the E8000S station connectors.

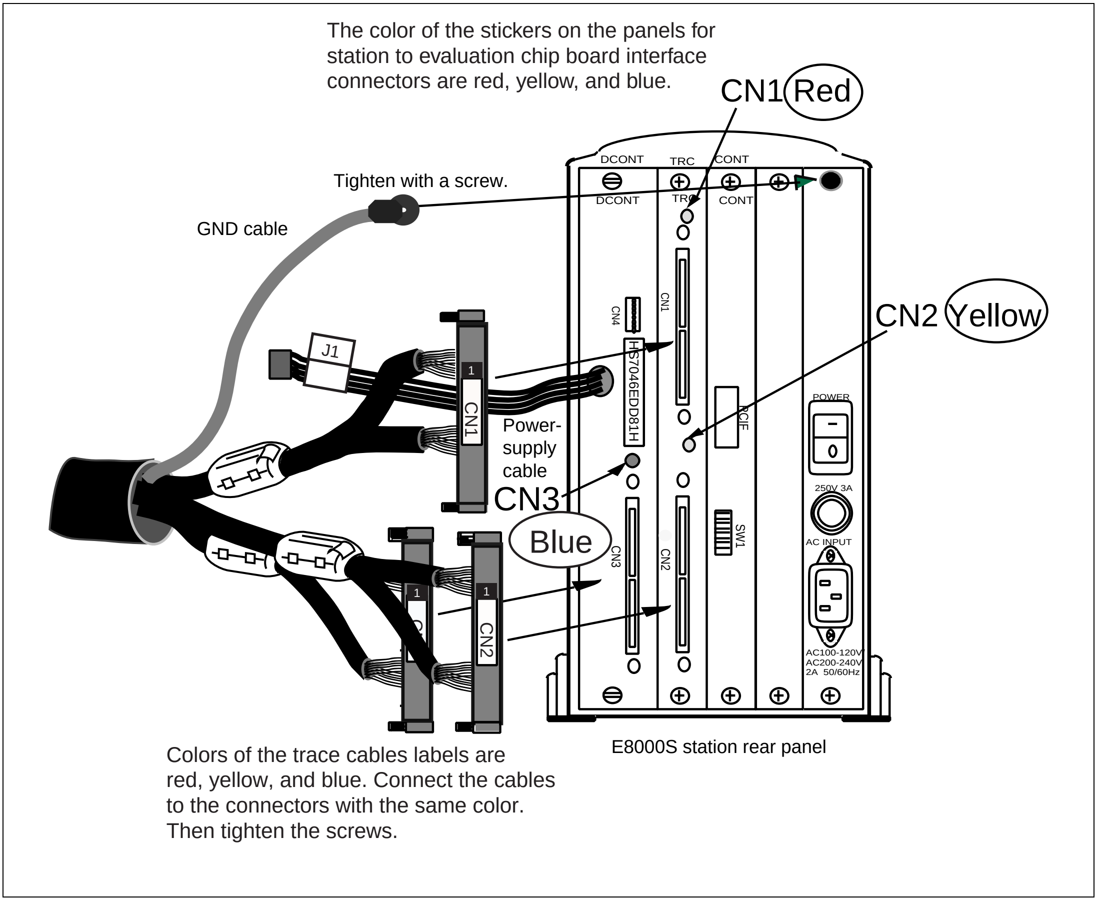


Figure 3.3 Connecting Trace Cables to the E8000S Station

⚠ WARNING

All three cables are bundled together at the other ends, where they are to be connected to the evaluation chip board. Be aware of which cables are bundled and the colors of the connectors when connecting the cables.

4. Connect the trace cables to the station to evaluation chip board interface connectors CN1, CN2, and CN3 on the evaluation chip board. Confirm that each trace cable connected to a connector on the E8000S station is also connected to its corresponding station to evaluation chip board interface connector on the evaluation chip board. Connect the trace cables using the same method as in step 3. Figure 3.4 shows how to connect the trace cables to the evaluation chip board interface connectors.

5. Connect the power-supply cables in the device control board to the evaluation chip board. The power-supply cable prevents an insertion error. Figure 3.4 shows how to connect the power-supply cables.

6. Connect the GND cable of the trace cable to the FG pin on the evaluation chip board.

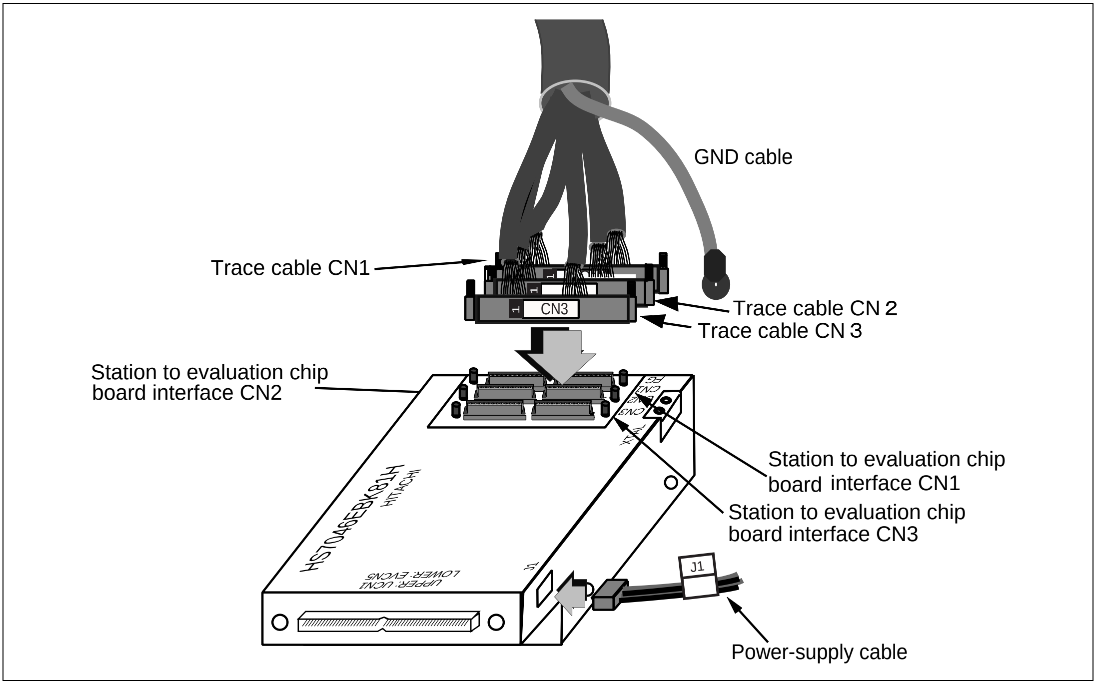


Figure 3.4 Connecting Trace Cables to the Evaluation Chip Board

3.3.3 Connecting the User System Interface Cable

⚠ 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 CABLE. Connection with the power on or incorrect connection will damage the emulator, user system interface cable, and user system, and result in a FIRE HAZARD.

For details on the method of connecting the user system interface cable, refer to the descriptions of the user system interface cables for individual SH7046-series products.

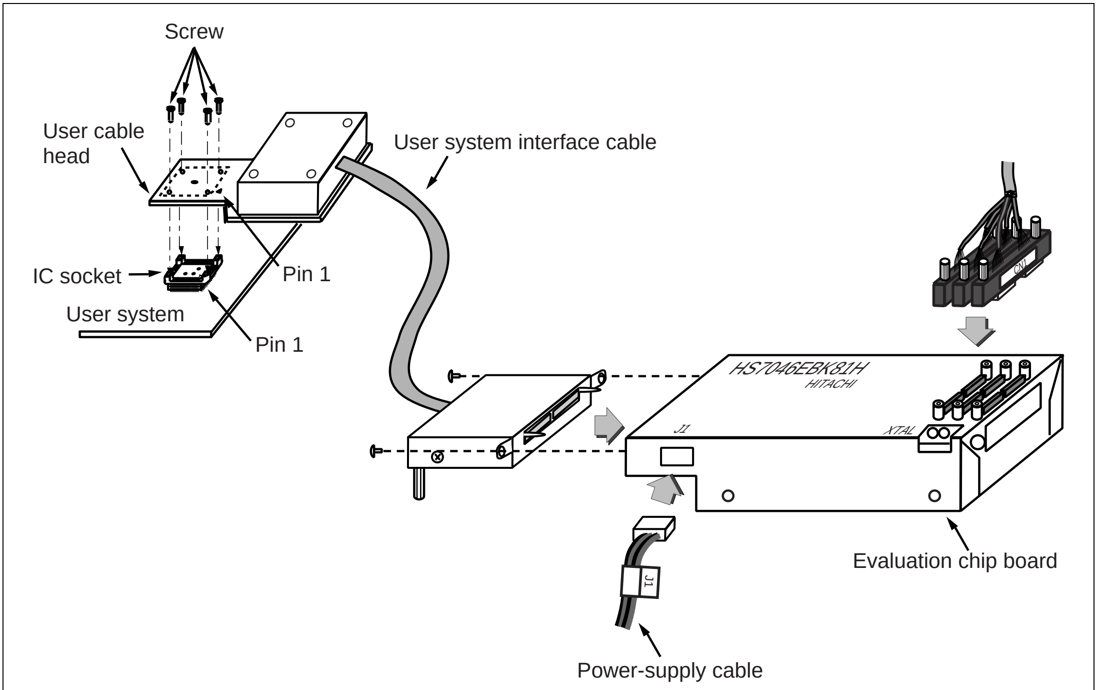
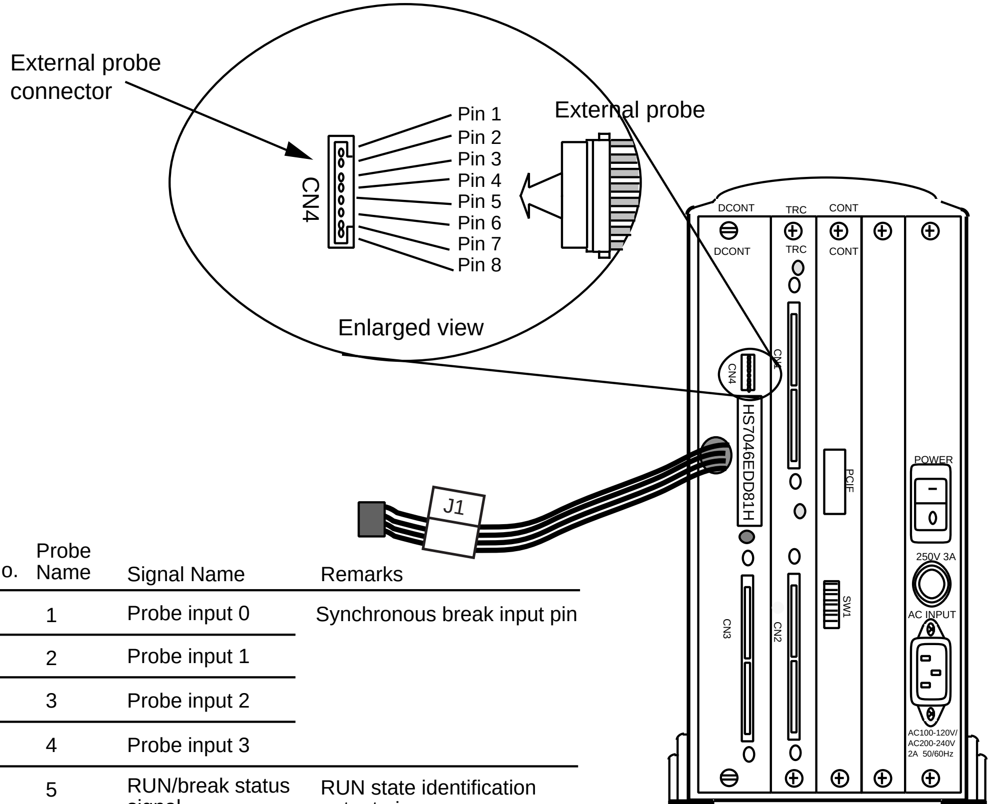


Figure 3.5 Connecting the User System Interface Cable

3.3.4 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 E8000S station's rear panel, it enables external signal tracing and multibreak detection. Figure 3.6 shows the external probe connector.



Pin No.	Probe Name	Signal Name	Remarks
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	RUN/break status signal	RUN state identification output pin
6	T	Trigger output	Trigger mode output pin
7	G	GND	GND connection pin
8	G		

Figure 3.6 External Probe Connector

3.3.5 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 50 MHz (quadruple the external clock frequency of 12.5 MHz) as the MCU clock input.

CLOCK command
- XTAL (Crystal oscillator: 5 to 12.5 MHz)
- USER (External clock: 4 to 12.5 MHz)
- 12 (Emulator internal clock: 12.5 MHz)
- 10 (Emulator internal clock: 10.0 MHz)
- 8 (Emulator internal clock: 8.0 MHz)
- 6 (Emulator internal clock: 6.0 MHz)
- 4 (Emulator internal clock: 4.0 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 to 12.5 MHz.

CAUTION

Always switch OFF the emulator and user system before connecting or disconnecting the CRYSTAL OSCILLATOR. 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.
2. Attach the crystal oscillator into the terminals on the evaluation chip board (figure 3.7).
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.

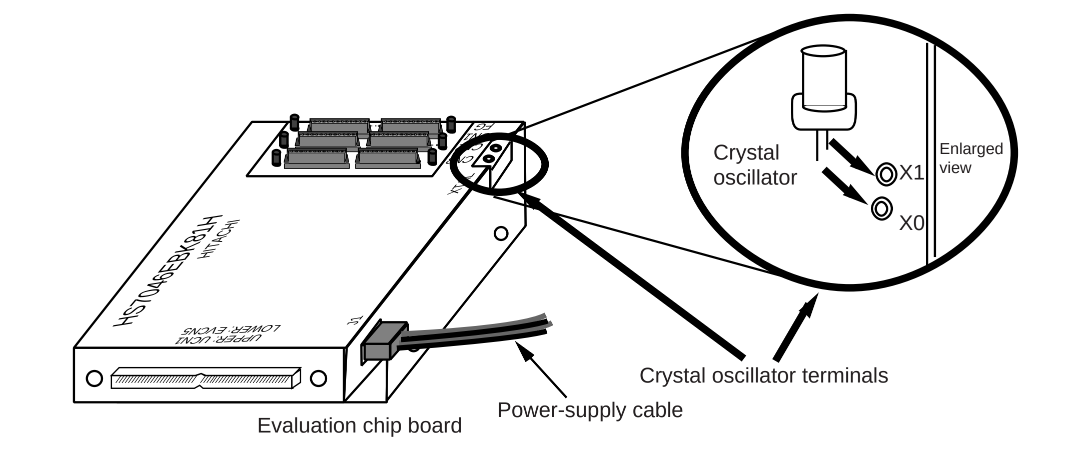


Figure 3.7 Installing the Crystal Oscillator

External Clock: Use the following procedure to select the external clock.

1. Check that the emulator power switch is turned off.
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 4 (4.0 MHz), 6 (6.0 MHz), 8 (8.0 MHz), 10 (10.0 MHz), or 12 (12.5 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

3.3.6 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 E8000S station, the signal ground and frame ground are connected (figure 3.8). 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 3.9) so that the ground potentials become even.

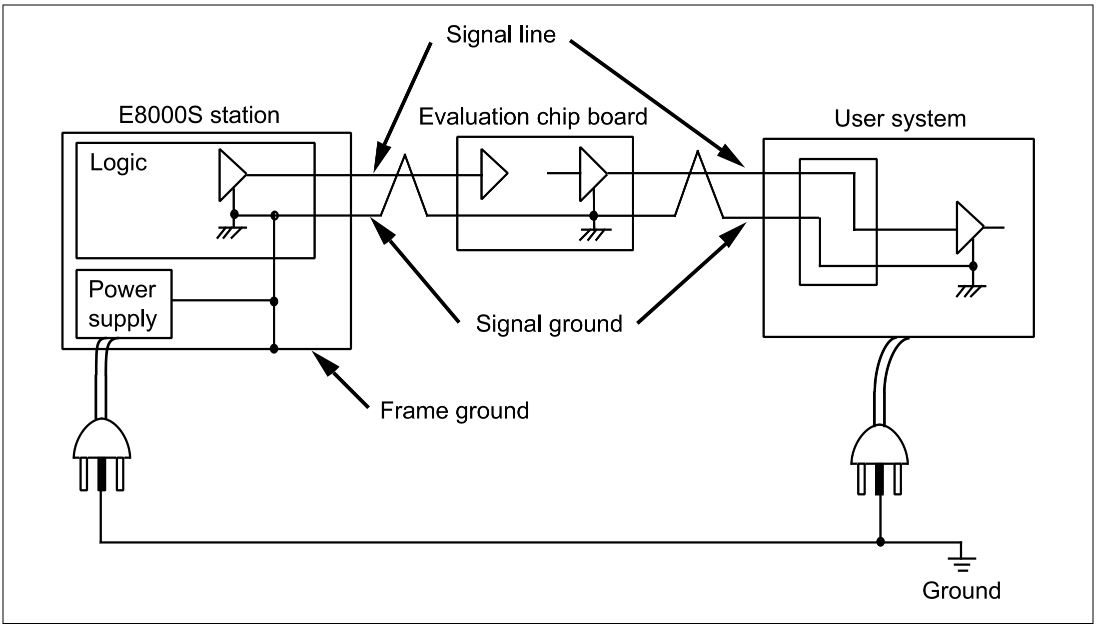


Figure 3.8 Connecting the System Ground

⚠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.

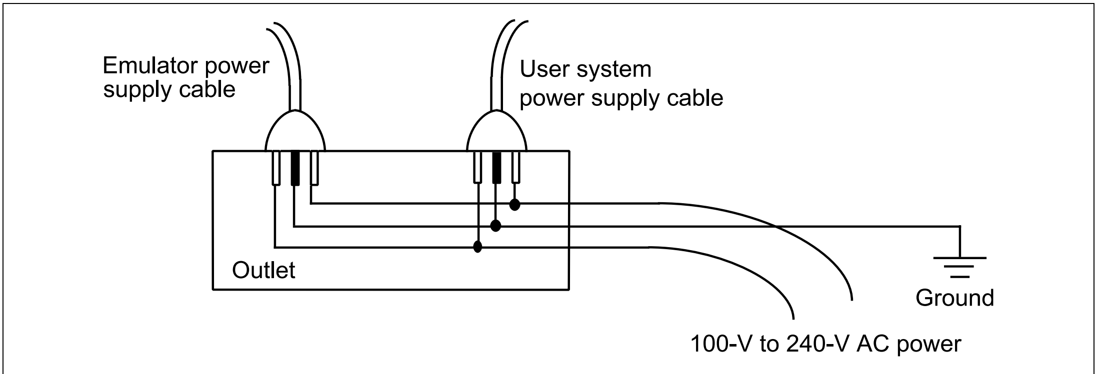


Figure 3.9 Connecting the Frame Ground

3.3.7 Setting the Host Interface Switch

> **CAUTION**
>
> **Do not change the settings of host interface switch SW1.**

The host interface consists of eight switches, as shown in figure 3.10. The switch state becomes on when the switches are pushed to the left, and the state becomes off when the switches are pushed to the right. To change the console interface settings, turn switches S1 to S6 off and switches S7 and S8 on in the console interface switch SW1.

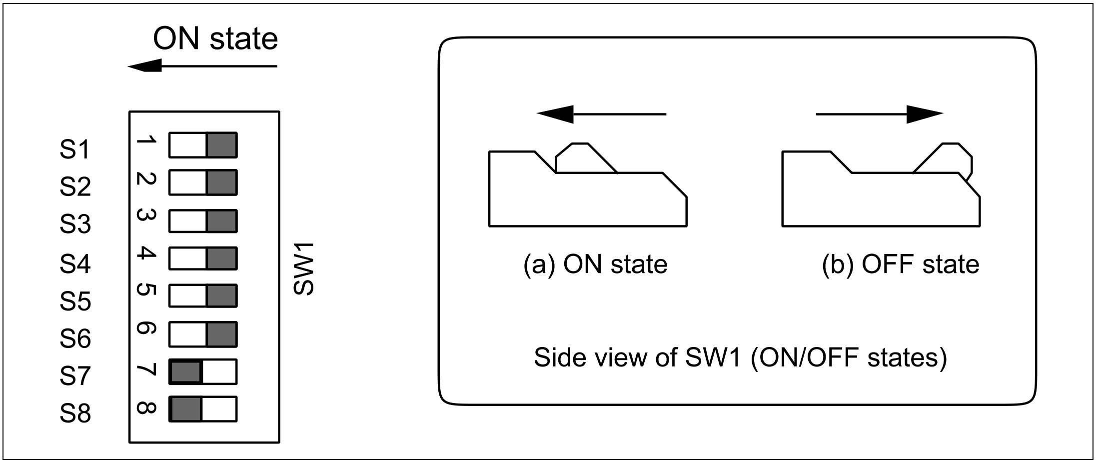


Figure 3.10 Host Interface Switches

3.4 Setting Up with Windows® Operating Systems

The following describes the setup procedure for Windows® operating systems.

Since hardware is set up during installation, the installation procedure may differ according to the operating system and interface used (ISA, PCI, PCMCIA, or LAN). Proceed with installation according to the procedures for the environment in use.

To connect the host computer to the emulator, either a PCI interface board (HS6000EIC01H and HS6000EIC02H), PC interface card (HS6000EIP01H), PC interface board (HS6000EII01H), or a LAN adapter (HS6000ELN01H) can be selected.

For details on the connection of the PC interface board (ISA bus specifications), refer to section 3.4.4, PC Interface Board Specifications (ISA Bus Specifications) or the user's manual. For details on other connections, refer to the related user's manual.

To install the provided software, insert the CD-R in the CD-ROM drive. If there are any applications running, exit from them before installing the software.

Click [Run…] from the [Start] menu. The [Run] dialog box will be displayed. Enter <Drive>:\7046\setup.exe and click the [OK] button. <Drive> is the CD-ROM drive.

Proceed with installation by following instructions provided by the install wizard.

3.4.1 Setting Up the PCI Interface Board on Windows® 95, Windows® 98, or Windows® Me

When Using the PC Interface Board:

- Install the provided software. Select [PCI Card Driver] as the component.
- Exit the operating system, shut down the host computer, and turn off the power switch.
- Install the PCI interface board into the host computer.
- Turn on the host computer. Hardware will be acknowledged and driver installation will start automatically.
- Select [Search for the best driver for your device] for driver installation, and select [Specify a location] for the location.
- Specify <Drive>:\7046\DRIVERS\PCI\95 for the location. <Drive> is the CD-ROM drive.
- Check that PCIHEI Rev 1 has been detected and complete driver installation.

When Using the PC Interface Card:

- Install the provided software. Select [PC Card Driver (PCMCIA)] as the component.
- Install the PC interface card into the host computer.
- Hardware will be acknowledged and driver installation will start automatically.
- Select [Search for the best driver for your device] for driver installation, and select [Specify a location] for the location.
- Specify <Drive>:\7046\DRIVERS\PCI\95 for the location. <Drive> is the CD-ROM drive.
- Check that E6000 PC Card has been detected and complete driver installation.

When Using the PC Interface Board:

- Refer to section 3.4.4, PC Interface Board Specifications (ISA Bus Specifications), and set the operating system environment and install the ISA bus interface board.
- Install the provided software. Select [ISA Driver] as the component.

Note: An ISA driver to run under Windows® Me is not available.

When Using the LAN Adapter:

- Install the provided software. Select [E8000 LAN Driver] as the component.
- Connect the LAN adapter to the target network and turn the LAN adapter on.
- Select [SH7046 E8000S Emulator Software]-[Tools]-[LAN Adapter Configuration] from [Start]-[Programs] to start the LAN Adapter Configuration.
- Set the LAN adapter by LAN Adapter Configuration.
- Turn the LAN adapter off and connect the LAN adapter to the target network and turn the LAN adapter on. Then LAN adapter must be turned off even when the network is not changed.

Note: When using a LAN adapter that has already been set, it is unnecessary to set it again. Use LAN Adapter Configuration to define only the host computer. For details on the usage of the LAN Adapter Configuration, refer to the online help of the LAN Adapter Configuration.

3.4.2 Setting Up the PC Interface Board on Windows NT® 4.0

When Using the PCI Interface Board:

- Exit from the operating system, shut down the host computer, and turn off the power switch.
- Install the PCI interface board into the host computer.
- Turn on the host computer. Log-on as Administrator.
- Install the provided software. Select [PCI Card Driver] as the component.
- Restart the host computer.

When Using the PC Interface Card:

- Exit from the operating system, shut down the host computer, and turn off the power switch.
- Install the PC interface card into the host computer.
- Turn on the host computer. Log-on as Administrator.
- Install the provided software. Select [PC Card Driver (PCMCIA)] as the component. Check the values specified for use by the PC interface card before the information is requested during installation. Select and start [Administrative Tools (Common)]-[Windows NT Diagnostics] from [Start]-[Programs] to check the current status of IRQ, I/O ports, and memory in the resource panels, and set values that will not cause contention with other devices. Required resources are as follows: one channel for IRQ, H'F bytes for I/O ports, H'4000 bytes for use as memory.
- Restart the host computer.

Note: Default settings start all the drivers selected as [Drivers] components after the activation of the host computer. When the host computer is activated without a given card or an inappropriate driver is installed, it is not possible to start the given driver. The service control manager informs the user of the error, but it causes no other problem.

When Using the PC Interface Board:

- Refer to section 3.4.4, PC Interface Board Specifications (ISA Bus Specifications), and set the operating system environment and install the ISA bus interface board.
- Log-on as Administrator.
- Install the provided software. Select [ISA Driver] as the component.
- Restart the host computer.

When Using the LAN Adapter:

- Log-on as Administrator.
- Install the provided software. Select [E8000 LAN Driver] as the component.
- Connect the LAN adapter to the target network and turn on the LAN adapter.
- Select [SH7046 E8000S Emulator Software]-[Tools]-[LAN Adapter Configuration] from [Start]-[Programs] and start the LAN Adapter Configuration.
- Set the LAN adapter by LAN Adapter Configuration.
- Turn off the LAN adapter and connect the LAN adapter to the target network and turn on the LAN adapter. Then LAN adapter must be turned off even when the network is not changed.

Note: When using a LAN adapter that has already been set, it is unnecessary to set it again. Use LAN Adapter Configuration to define only the host computer. For details on the usage of the LAN Adapter Configuration, refer to the online help of the LAN Adapter Configuration.

3.4.3 Setting Up the PC Interface Board on Windows® 2000

Support of the PC Interface Board:

The PC interface board is not supported in Windows® 2000.

When Using the PCI Interface Board:

- Log-on as Administrator.
- Install the provided software. Select [PCI Card Driver] as the component.
- Exit from the operating system, shut down the host computer, and turn off the power switch.
- Install the PCI bus interface board into the host computer.
- Turn on the host computer. Hardware will be acknowledged and driver installation will start automatically.
- Select [Search for a suitable driver for my device (recommended)] for driver installation, and select [Specify a location] for the location.
- Specify <Drive>:\7046\DRIVERS\PCI\2000 for the location. <Drive> is the CD-ROM drive.
- Check that E6000 PCI Card has been detected and complete driver installation.

When Using the PC Interface Card:

- Log-on as Administrator.
- Install the provided software. Select [PC Card Driver (PCMCIA)] as the component.
- Install the PC interface card into the host computer.
- Hardware will be acknowledged and driver installation will start automatically.
- Select [Search for a suitable driver for my device (recommended)] for driver installation, and select [Specify a location] for the location.
- Specify <Drive>:\7046\DRIVERS\PCMCIA\2000 for the location. <Drive> is the CD-ROM drive.
- Check that E8000/E6000 PC Card has been detected and complete driver installation.

When Using the LAN Adapter:

- Log-on as Administrator.
- Install the provided software. Select [E8000 LAN Driver] as the component.
- Connect the LAN adapter to the target network and turn on the LAN adapter.
- Select [SH7046 E8000S Emulator Software]-[Tools]-[LAN Adapter Configuration] from [Start]-[Programs] and start the LAN Adapter Configuration.
- Set the LAN adapter by LAN Adapter Configuration.
- Turn off the LAN adapter and connect the LAN adapter to the target network and turn on the LAN adapter. Then LAN adapter must be turned off even when the network is not changed.

Note: When using a LAN adapter that has already been set, it is unnecessary to set it again. Use LAN Adapter Configuration to define only the host computer. For details on the usage of the LAN Adapter Configuration, refer to the online help of the LAN Adapter Configuration.

3.4.4 PC Interface Board Specifications (ISA Bus Specifications)

PC Interface Board Specifications (ISA Bus Specifications): Table 3.1 lists the ISA-bus PC interface board specifications. For details on the PCI-bus and PCMCIA-bus interface boards, refer to their description notes.

Table 3.1 PC Interface Board Specifications

Item	Specifications
Host computer that can be used	ISA-bus specifications PC, or compatible machine
System bus	ISA-bus specifications
Memory area	16 kbytes
Memory area setting	Can be set at every 16 kbytes in the range from H'C0000 to H'EFFFF with a switch.

Switch Settings of the PC Interface Board

Memory-Area Setting: The PC interface board uses a 16-kbyte memory area on the host computer. The memory area to be used must be allocated to the memory area on the host computer with a switch on the PC interface board. Any 16 kbytes in the range of H'C0000 to H'EFFFF can be allocated (figure 3.11). Addresses to be allocated must not overlap the memory addresses of other boards. An overlap will cause incorrect operation.

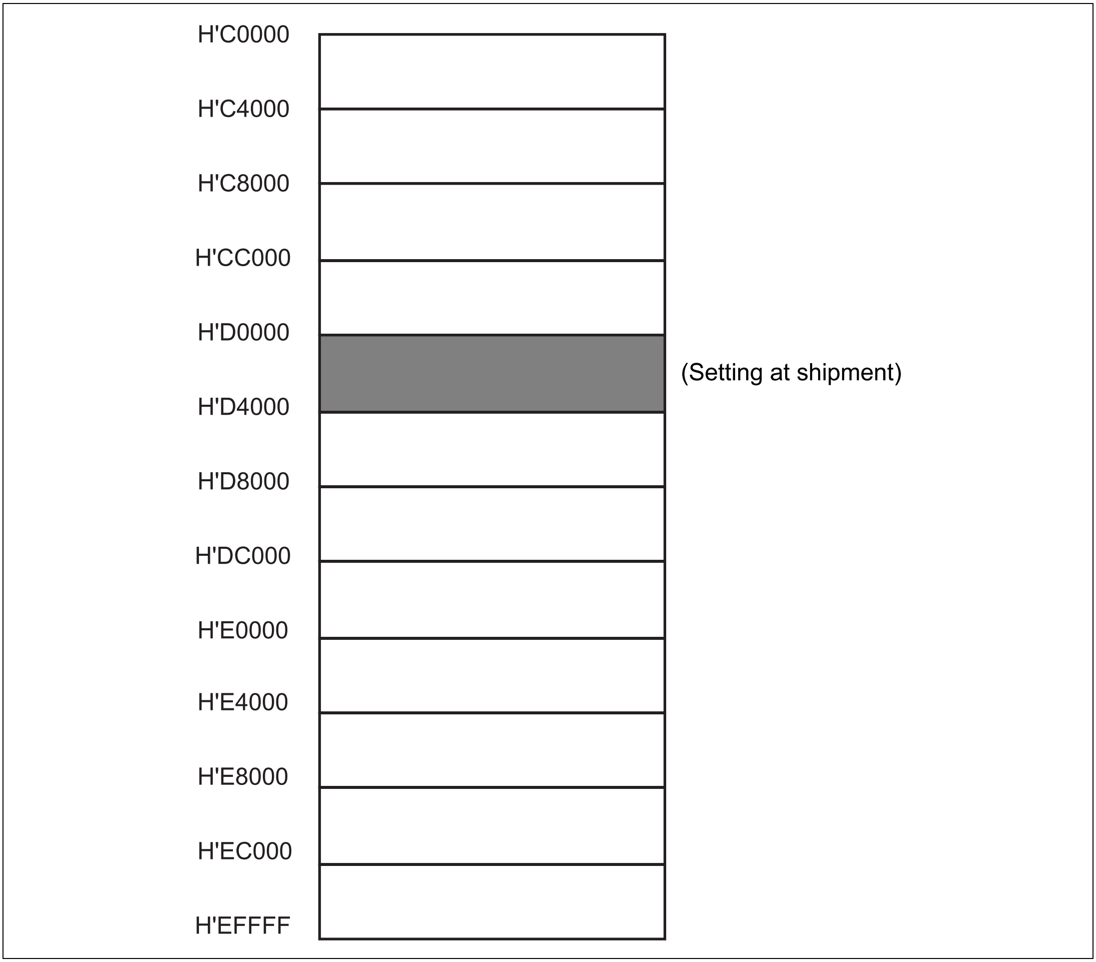


Figure 3.11 Allocatable Memory Area of PC Interface Board

Switch Setting: A rotary switch is installed on the PC interface board (figure 3.12). The switch is used to set the memory-area allocation. Table 3.2 lists the switch setting states. The switch setting at emulator shipment is No. 4 (memory area H'D0000 to H'D3FFF).

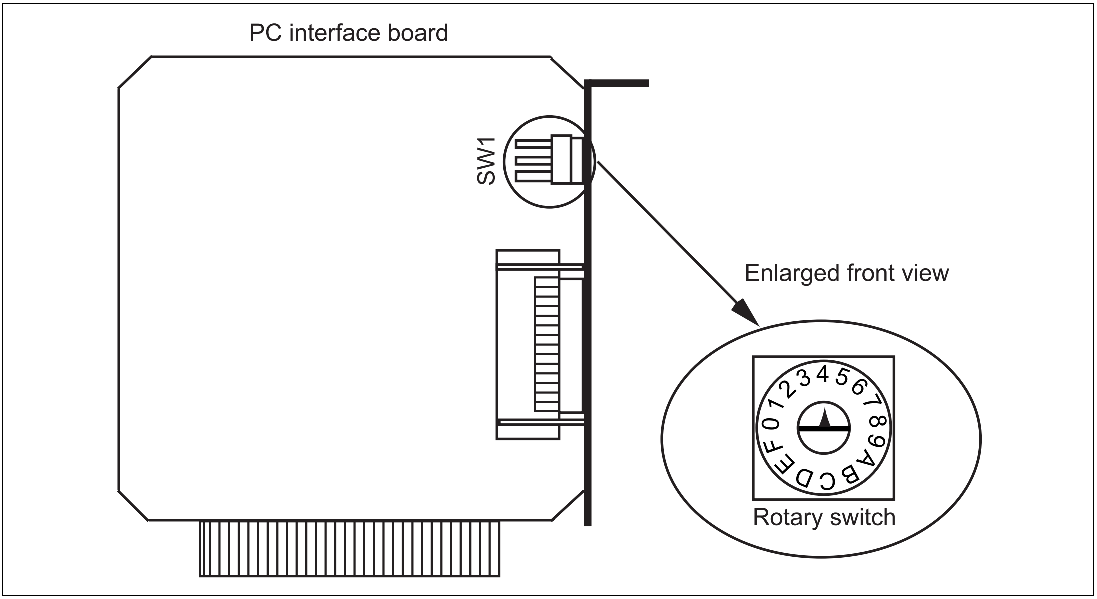


Figure 3.12 PC Interface Board Switch

Table 3.2 Switch Settings for Memory Areas

Switch Setting	Memory Area	Switch Setting	Memory Area
0	H'C0000 to H'C3FFF	8	H'E0000 to H'E3FFF
1	H'C4000 to H'C7FFF	9	H'E4000 to H'E7FFF
2	H'C8000 to H'CBFFF	A	H'E8000 to H'EBFFF
3	H'CC000 to H'CFFFF	B	H'EC000 to H'EFFFF
4 (setting at shipment)	H'D0000 to H'D3FFF	C	Not used
5	H'D4000 to H'D7FFF	D	Not used
6	H'D8000 to H'DBFFF	E	Not used
7	H'DC000 to H'DFFFF	F	Not used

Note: When C to F of the switch are set, memory areas cannot be allocated. Set one of 0 to B.

Installing the PC Interface Board:

⚠ WARNING

Always switch OFF the personal computer (host computer) and peripheral devices connected to the personal computer before installing the PC interface board. Failure to do so will result in a FIRE HAZARD and will damage the personal 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 ISA-bus specification extension slot. Tighten the screw after confirming that the PC interface cable can be connected to the board.

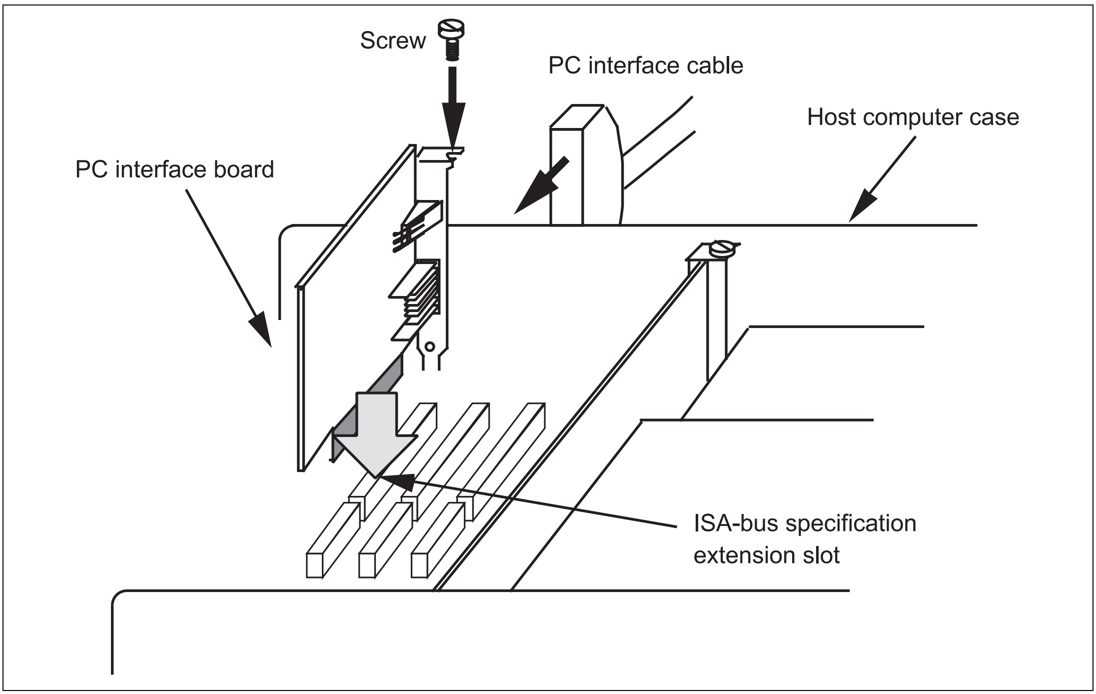


Figure 3.13 Installing the PC Interface Board

Connecting the E8000S 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 E8000S station to the PC interface board with the PC interface cable supplied, as shown in figure 3.14.

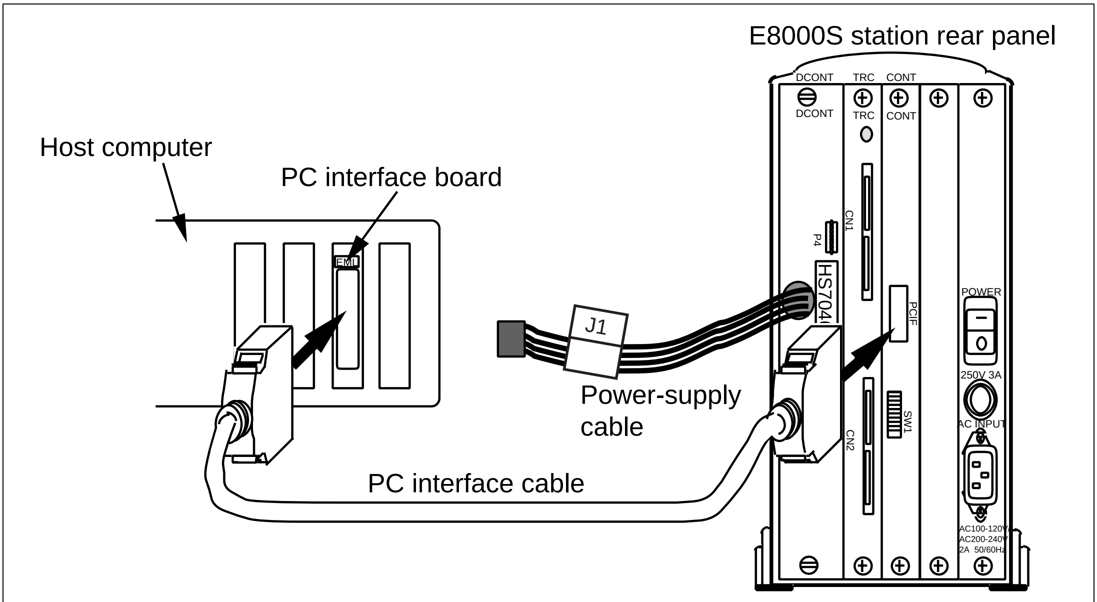


Figure 3.14 Connecting the E8000S Station to the PC Interface Board

Setting Up the PC Interface Board on Windows® 95 or Windows® 98: Description of setting up a PC interface board (HS6000EII01H) is given below, taking settings under Windows® 95 as an example.

- Start Windows® 95.
- Click the [My Computer] icon with the right mouse button and select [Properties] from the pop-up menu.

The [System Properties] dialog box will be displayed.

- Double-click the [Computer] icon in the [Device Manager] panel to open the [Computer Properties] dialog box.
- Click the [Memory] in the [View Resources] panel to display the memory resources.

Select one of the address ranges that is not listed in the [Computer Properties] dialog box. For example, if you select the range H'D8000 to H'DBFFF, the corresponding switch number will be 6.

Define the memory area so that Windows® 95 does not use the area as follows:

- Click [Memory] in the [Reserve Resources] panel and click [Add].

The [Edit Resource Setting] dialog box will be displayed.

- Enter the memory area addresses in [Start value] and [End value] and click [OK].
- Shut down the host computer (do not restart it) and turn off the power switch.
- Power on the host computer.
- Check that the area selected in the [Computer Properties] dialog box are displayed as [System Reserved] in the list.

The next step prevents the memory area for the PC interface board being accessed by another program. Modify the CONFIG.SYS file. Use the SYSEDIT program to edit the file.

- Select [Run] from the [Start] menu.
- Type SYSEDIT and click [OK]. The SYSEDIT will start.

When EMM386.EXE is used in the CONFIG.SYS file, the CONFIG.SYS file must be modified. If the CONFIG.SYS file is not used, or if EMM386.EXE is not used even when the CONFIG.SYS file is used, go to the following procedure and modify the SYSTEM.INI file.

- Locate the line in the CONFIG.SYS file that reads:

  device=C:\WINDOWS\EMM386.EXE RAM

- Change the line so that it reads as shown below.

  device=C:\WINDOWS\EMM386.EXE RAM X=*aaaa–bbbb*

  Here, *aaaa* is the upper four digits of [Start value] and *bbbb* is the upper four digits of [End value]. For example, for the switch set to 6, you would set the line to read:

  device=C:\WINDOWS\EMM386.EXE RAM X=D800–DBFF

- Save the CONFIG.SYS file.

The following is about modifying the SYSTEM.INI file.

- Add the following line to the [386Enh] section in the SYSTEM.INI file:

  EMMExclude=*aaaa-bbbb*

  Here, *aaaa* is the upper four digits of [Start value] and *bbbb* is the upper four digits of [End value]. For example, for the switch set to 6, you would set the line to read:

  EMMExclude = D800-DBFF

- Save the SYSTEM.INI file and exit the SYSEDIT.
- Restart the host computer.

Setting Up the PC Interface Board on Windows NT® 4.0: The description given below is about setting up the PC interface board on Windows NT® 4.0.

For the installation of the PC interface board to the ISA bus slot, refer to the manual of your host computer.
This section describes the general procedure for installing the PC interface board in the host computer.

Log-on to the host computer as Administrator. Check which upper memory areas have already been used.

Start Windows NT®4.0.

- Execute [Start]-[Programs]-[Administrative Tools (Common)]-[Windows NT Diagnostics].
- Click the [Memory] button in the [Resource] tab and, in the following form, make a note of the upper memory areas that have already been used.
- Exit Windows NT®4.0.
- Register the memory area for the PC interface board. Select one of the memory areas that correspond to the following PC interface board switch settings, and no other devices can access the selected memory area.

If the Intel P&P BIOS disk is supplied with the host computer, register the memory area as follows:

- Start the host computer with the Intel P&P BIOS disk.
- Check the upper memory areas that have already been used, with [View]-[System Resources].
- Add [Unlisted Card] with [Configure]-[Add Card]-[Others...]
- Click [No] in the dialog box displayed because there is no .CFG file.
- Move to the [Memory [hex]] list box in the [Configure Unlisted Card] dialog box.
- Click the [Add Memory...] button to display the [Specify Memory] dialog box.
- Enter a memory area range that is not used by any other device and that corresponds to one of the PC interface board switch settings.
- Save the file.
- Exit the current setup program.
- Shut down Windows NT® 4.0 and turn off the host computer power switch.
- Turn on the host computer.

3.5 Installing the System Program

A description of the installation of the system program is given below.

The E8000S system program must be transferred to flash memory in the E8000S station. The emulator cannot be used without the E8000S system program.

The system programs are stored in \SYSTEM under the HDI installation directory. Table 3.3 is a list of system programs with descriptions.

Table 3.3 Contents of Emulator System Programs

No.	File Name	Contents of File
1	E8000.SYS	System program for the emulator. Controls the evaluation ship board and executes various commands such as for emulation. Loaded to the memory of the emulator when the emulator system program is started up.
2	SHDCT046.SYS	MCU control program. Controls the MCU on the evaluation chip board. Loaded to the memory of the emulator when the emulator system program is started up.
3	SHCNF046.SYS	Configuration file for storing the MCU’s operating mode and map information. Loaded with the emulator system program.
4	DIAG.SYS	Diagnostic and maintenance program. Loaded to the memory of the E8000S station for maintenance.

The system programs are registered by using a dedicated E8000S system installation tool (hereafter referred to as the ESI). The procedure is described below.

To install the system programs, use Auto Install or Custom Install mode.

Auto Install mode automatically installs all system program.

Custom Install mode allows flexibility in the installation of the system programs.

First, Auto Install is described.

Turn on the E8000S station. Select [SH7046 E8000S Emulator Software]-[Tools]-[System Install Tool] from [Start]-[Programs].

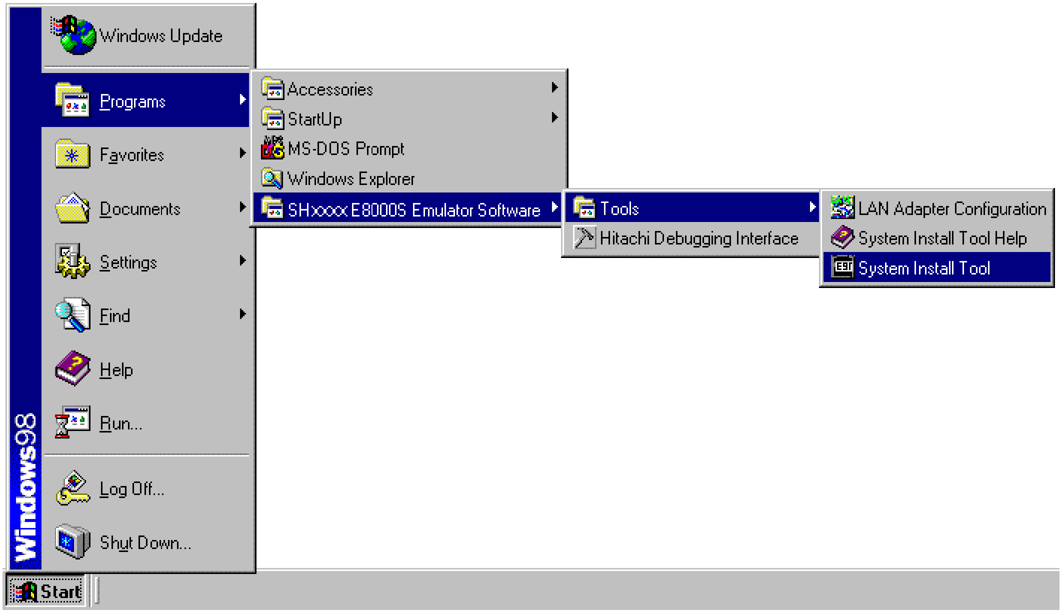


Figure 3.15 [Start] Menu (System Install Tool)

When the ESI has started up, the [Select Driver] dialog box will be displayed. Select the driver for the connection of the host computer and the E8000S Emulator from [Driver]. Table 3.4 shows the types of connections and drivers.

Table 3.4 Types of Connection and Drivers

Connection	Contents of File
When connecting the emulator to a PC interface board	Emulator ISA Driver
When connecting the emulator to a LAN adapter	E8000 LAN Driver
When connecting the emulator to a PCI interface board	Emulator PCI Card Driver
When connecting the emulator to a PC interface card	Emulator PC Card Driver

When “Link up” is displayed on the status bar, the initiation of the ESI has been completed.

Select the [Auto Install] radio button in the [Select Install] dialog box. The [Browse for Folder] dialog box will be displayed, so select the directory where the system program is stored. The default directory will be a directory under \SYSTEM in the HDI installation directory.

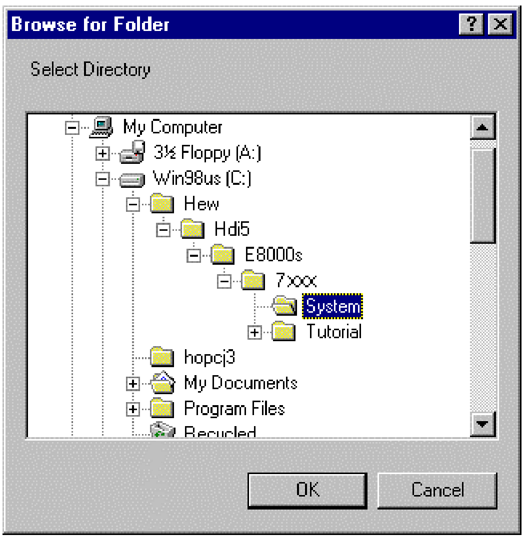


Figure 3.16 [Browse for Folder] Dialog Box

When [OK] is clicked, the [E8000 Load Files] dialog box will be displayed. Check the file names for installation.

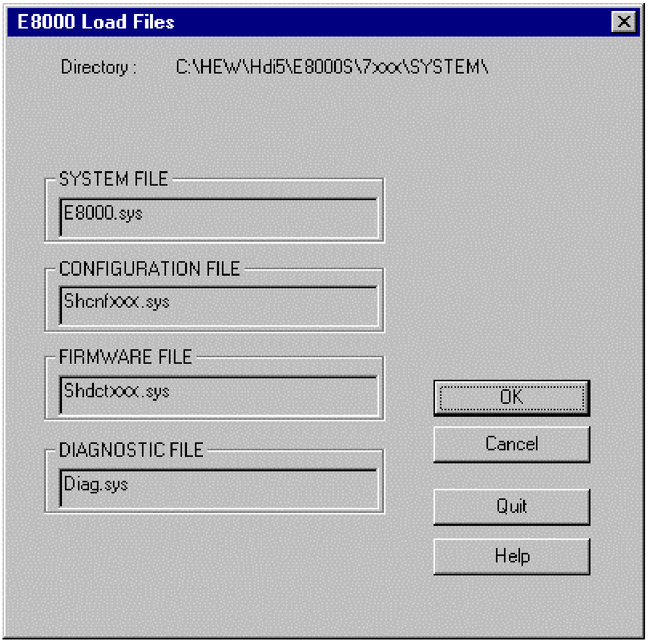


Figure 3.17 [E8000 Load Files] Dialog Box (SH7046)

Click the [OK] button. Installation of the system programs to the flash memory of the E8000S station will commence. The file name will be displayed in the status bar as each file is installed. Click [Cancel] to return to the [Select Install] dialog box.

When the registration is complete, the [System Install Completed!!] dialog box will be displayed.

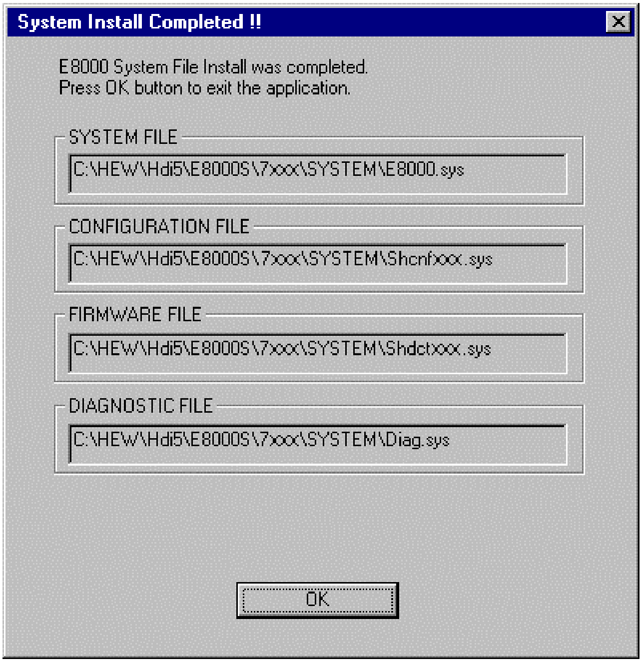


Figure 3.18 [System Install Completed!!] Dialog Box (SH7046)

Click the [OK] button. The installation of the system program to the flash memory of the E8000S station will be completed, and the ESI will terminate. Turn off the emulator's power switch.

Next, Custom Install will be described.

When the [Select Install] dialog box has been displayed, select the [Custom Install] radio button. The [E8000 Load Files] dialog box will be displayed. Add or modify files.

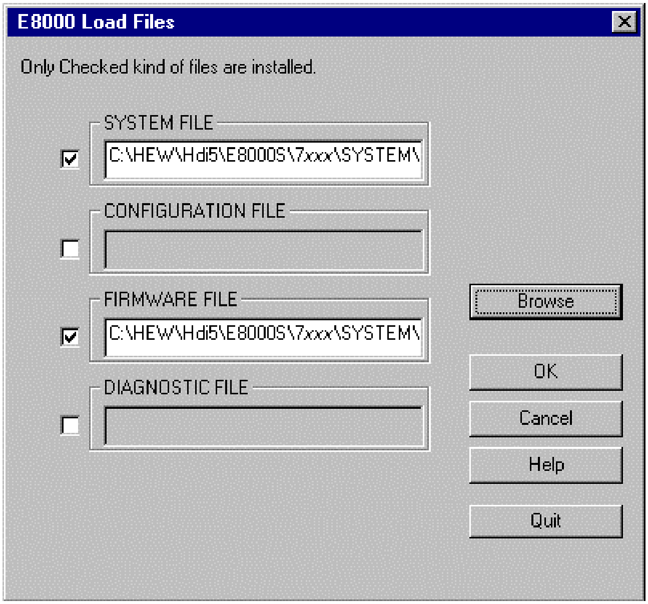


Figure 3.19 [E8000 Load Files] Dialog Box (SH7046)

Each check box corresponds to the combo box to its right. Select the check box that corresponds to the files you wish to install. Place the cursor on the combo box and click the [Browse] button. When the [Browse] button is clicked, the [File Selection] dialog box will be displayed.

Select the files to install. If you decide not to install a file, do not select the check box. When the check box is not selected, which means it is invalid, the file will be grayed-out and will not be installed.

Click the [OK] button. The system program will be installed in the flash memory of the E8000S station. The file name will be displayed in the status bar as each file is installed. From here, the description of Custom Install is the same as that of Auto Install. Click [Cancel] to return to the [Select Install] dialog box.

For details on the error messages displayed during ESI operation, refer to Troubleshooting in the ESI help file. For help on the ESI, select [SH7046 E8000S Emulator Software]-[Tools]-[System Install Tool] from the [Start] menu.

3.6 Initiating the HDI and Checking the Emulator

The next step is to check that the emulator and the HDI are initiated correctly.

Turn on the emulator after confirming that the S7 and S8 DIP switches of SW1 on the emulator are turned on.

Select [SH7046 E8000S Emulator Software]-[Hitachi Debugging Interface] from the [Start] menu.

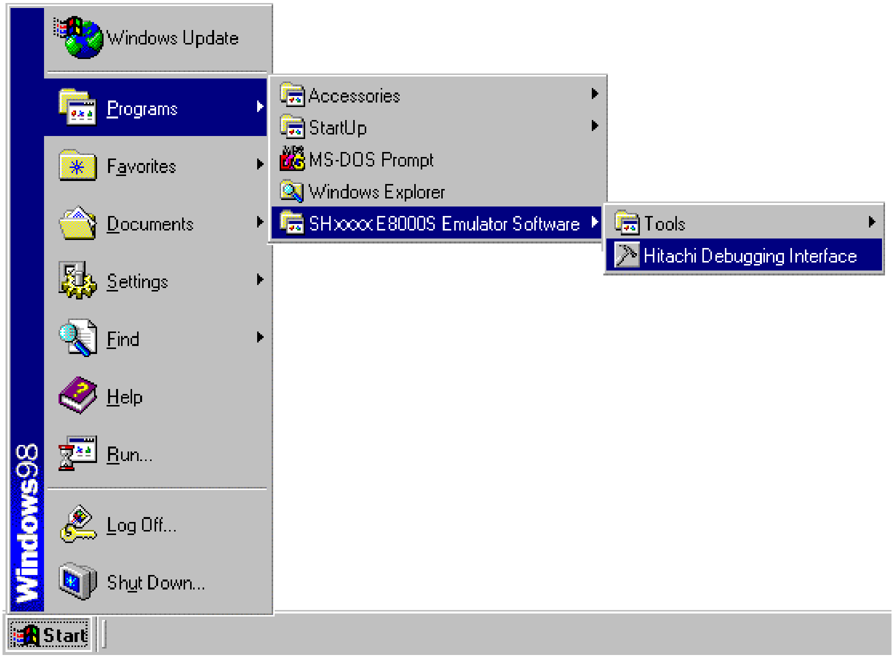


Figure 3.20 [Start] Menu (Initiating the HDI)

The [Select Session] dialog box will be displayed. Select the target emulator name in the combo box then click the [OK] button.

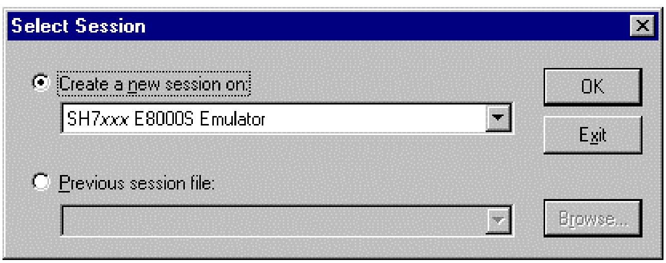


Figure 3.21 [Select Session] Dialog Box

When the HDI is started up for the first time, a [Driver Details] dialog box will be displayed. Use [Driver] to select the correct driver for the connected interface (PC interface board, PCI interface board, PC interface card, or LAN adapter) and click the [Close] button. Refer to table 3.4, Types of Connections and Drivers. The example below is when the emulator is to be connected to the PC interface via the ISA bus.

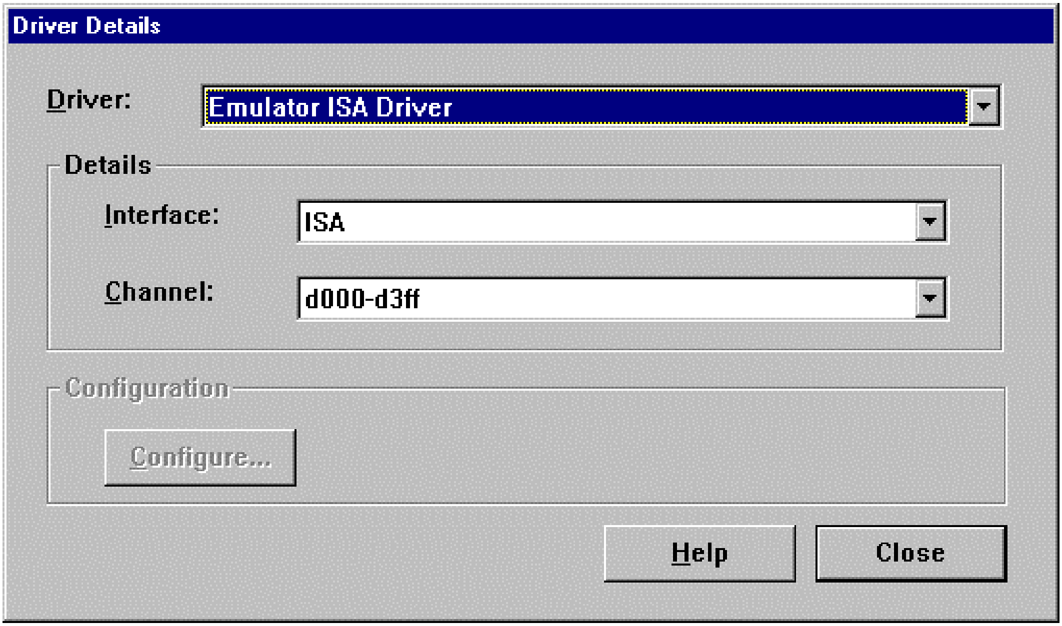


Figure 3.22 [Driver Details] Dialog Box (When ISA Bus Is Selected)

During the HDI initiation, the following messages are shown on the status bar of the HDI window.

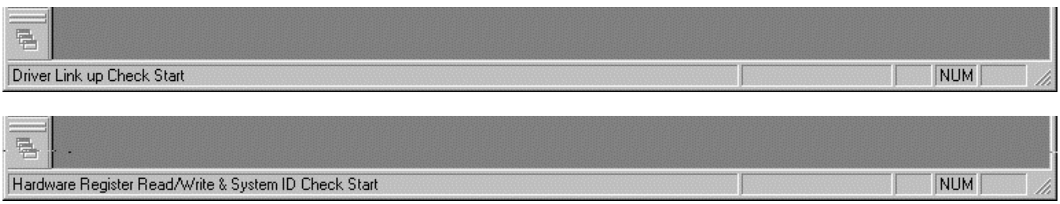


Figure 3.23 Status Bar during the HDI Initiation

A message box will be displayed to ask whether to initiate the diagnostic program (figure 3.30) when [Diagnostic Test Program] has been selected in the component selection dialog box when the HDI was installed. Click the [Yes] button to initiate the diagnostic program.
For details, refer to section 3.7.4, Operating Procedure for the Diagnostic Program. "Link up" appears on the status bar when the HDI and the emulator have successfully started up.

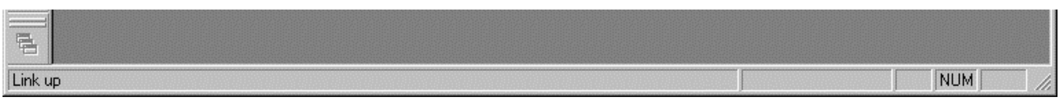


Figure 3.24 Status Bar at the HDI Initiation Completion

When HDI is started for the first time after installation, the following message box will be displayed. The same message box is not displayed again.

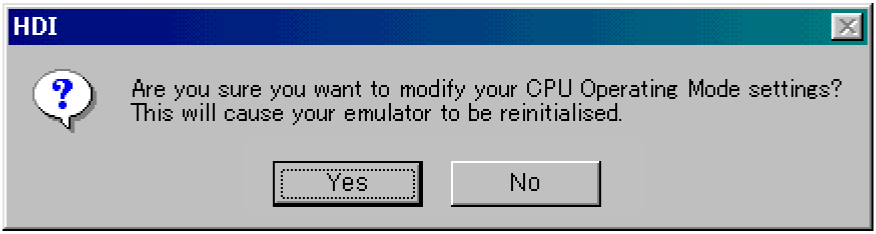


Figure 3.25 CPU Operating Mode Modification Message Mode (1)

Click [Yes] to display the [CPU Operating Mode] dialog box. In this dialog box, the operating mode of the emulator can be modified. When the emulator cannot be initiated correctly, the following message box will be displayed. Click [Yes] to display the [CPU Operating Mode] dialog box. The operating mode must be modified.

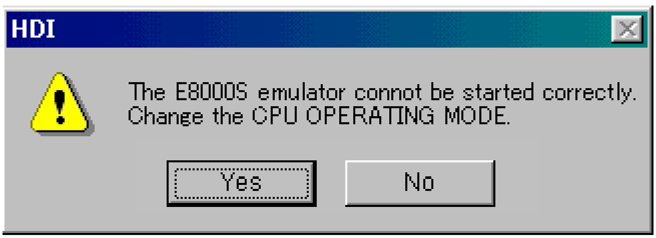


Figure 3.26 CPU Operating Mode Modification Message Mode (2)

3.7 Troubleshooting

3.7.1 Error Messages from the HDI

PC Interface Board Connection Failure: When the HDI cannot detect the PC interface board for the emulator, the HDI will display the following error message.

Figure 3.27 Error Message on PC Interface Board Connection Failure

The causes of such failures are given below:

(a) The address ranges that was set by the switch on the rear panel of the PC interface board is different from that set in the next setting.

- Settings of the [Computer Properties] dialog box
- Settings of the CONFIG.SYS file
- Settings of the SYSTEM.INI file

(b) Another application is using the selected memory area.

Emulator Connection Failure: When the HDI cannot detect the E8000S station, the HDI will display the following message box.

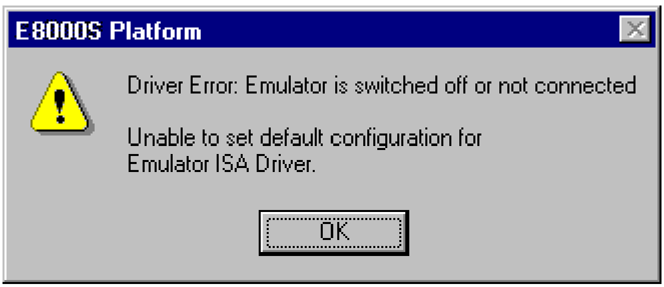


Figure 3.28 Error Message on Emulator Connection Failure

The causes of such failures are given below:

- The AC power cable is not connected to the emulator or the E8000S station power is not turned on.
- The PC interface board and the E8000S station are not correctly connected via the PC interface cable.

3.7.2 Error Messages from the Emulator

The emulator checks its internal RAM and registers from initiation of the HDI until “Link up” is displayed in the status bar. This is a diagnostic function. The emulator executes the diagnostic program when its power is turned on or when its system program is initiated. After the emulator displays any of the error messages described below, the emulator will display the [Target Link Down] dialog box and terminate the HDI session.

Errors when checking internal RAM and registers at power-on: The emulator checks its internal RAM and registers at power-on. If an error occurs, either of the following error messages is displayed during HDI operation.

Error Message	Description
** RAM ERROR ADDR=xxxxxxxx W-DATA=xxxxxxxx R-DATA=xxxxxxxx	The emulator checks its internal RAM and registers at power-on. If an error occurs, the address, write data, and read data are displayed in the following form.
*** xxxx REGISTER ERROR W-DATA=xx R-DATA=xx xxxx: Name of the emulator internal register where an error occurs.	The emulator checks the registers of the E8000S station, and displays the message when an error occurs.

Note: Operation continues if an error occurs but the error should be investigated according to section 3.7, Troubleshooting, without loading the emulator system program.

Diagnostic checks at program start-up: Next, the emulator system program starts up. The emulator system program performs diagnostic checks of the registers of the emulator. When an error is thus detected in the emulator, the emulator displays the following error message while the HDI is operating.

1. Errors that may occur when registers that control emulation are being tested

   The emulator control registers are being checked. If an error occurs, one of the following messages is displayed.

Error Message	Description
*** INVALID DCONT BOARD	Another device control board is connected. Please check the MCU type and install the correct emulator system program or exchange the device control board.
*** DEVICE CONTROL BOARD DISCONNECTION	The device control board is disconnected.
*** EVACHIP BOARD DISCONNECTION	The evaluation chip board is disconnected.
*** EVACHIP BOARD DISCONNECTION (x)	Check that the trace cable and power supply cable J1 is correctly connected. x indicates the trace cable or power supply cable J1 that is not correctly connected (x: CN1, CN2, CN3, or J1).
*** xxxxxxxx REGISTER ERROR W-DATA = xxxx R-DATA = xxxx	An error occurred in the register. xxxxxxxx: Name of the emulator internal register where an error occurs. B0TRAR,ECT,B0CNR,B0MDCNR,B0MASCR, B0CECR,B1CNR,B1MDCNR,B1MASCR, B1CECR,MAPR0,MAPR1,MAPR2,MAPR3
*** SHARED RAM ERROR ADDR = xxxxxx W-DATA = xxxxxxxx R-DATA = xxxxxxxx	An error occurred in the shared RAM.
*** BxTBM ERROR ADDR = xxxxxx W-DATA = xxxxxxxx R-DATA = xxxxxxxx	An error occurred in the trace buffer memory.
*** FIRM RAM ERROR ADDR = xxxxxx W-DATA = xxxxxxxx R-DATA = xxxxxxxx	An error occurred in the firmware RAM area.

2. Errors that may occur when the device control board is being tested

   A program operating in the device control board is being loaded and the device control board is being tested. If an error occurs, the following message is displayed.

Error Message	Description
INVALID FIRMWARE SYSTEM*	Another firmware has been installed. Reinstall the correct emulator system program. This message is displayed when the H-UDI's input clock frequency is higher than the input clock frequency for the peripheral internal module (Pϕ).
*** EMULATOR FIRMWARE NOT READY	The program operating on the device control board is not operating correctly. Please check that the evaluation chip board is connected correctly.
** FIRMWARE SYSTEM FILE NOT FOUND	A program operating in the device control board does not exits. An incorrect system program has been registered in the flash memory. Reinstall the system program and restart the emulator.
SDI BOOT FAILED*	An incorrect evaluation chip board is connected. Connect the correct evaluation chip board. . This message is displayed when the H-UDI's input clock frequency is higher than the input clock frequency for the peripheral internal module.

Note: While these error messages are displayed, the message box shown in figure 3.26 is displayed. Click the [Yes] button to display the [CPU Operating Mode] dialog box and modify the settings of the operating mode and H-UDI input clock. This may result in correct start up.

Emulator System Down: If an exceptional operation occurs during emulator monitor or emulator system program execution (HDI command wait state), the system shuts down (the HDI links down)

*** E8000 SYSTEM DOWN ***

If an error occurs, inform a Hitachi sales agency of the error.

3.7.3 Troubleshooting Procedure

When an error occurs in the system, use the troubleshooting Problem Analysis Diagram (PAD, see figure 3.29) to determine the cause of the error.

Start from START in figure 3.29 and determine the state of the system. Follow the instructions that request operator assistance or intervention.

Note that “system defect” means that the emulator is malfunctioning. Execute the diagnostic program in the way described in the Diagnostic Program Manual, and inform a Hitachi sales agency of the test results in detail, because system defects may arise for a number of reasons.

If the cause of the error is an emulator defect, execute the provided diagnostic program to collect the internal details from the emulator. Please inform us of the results of testing.

For details on the diagnostic program, refer to section 3.7.4, Operating Procedure for the Diagnostic Program.

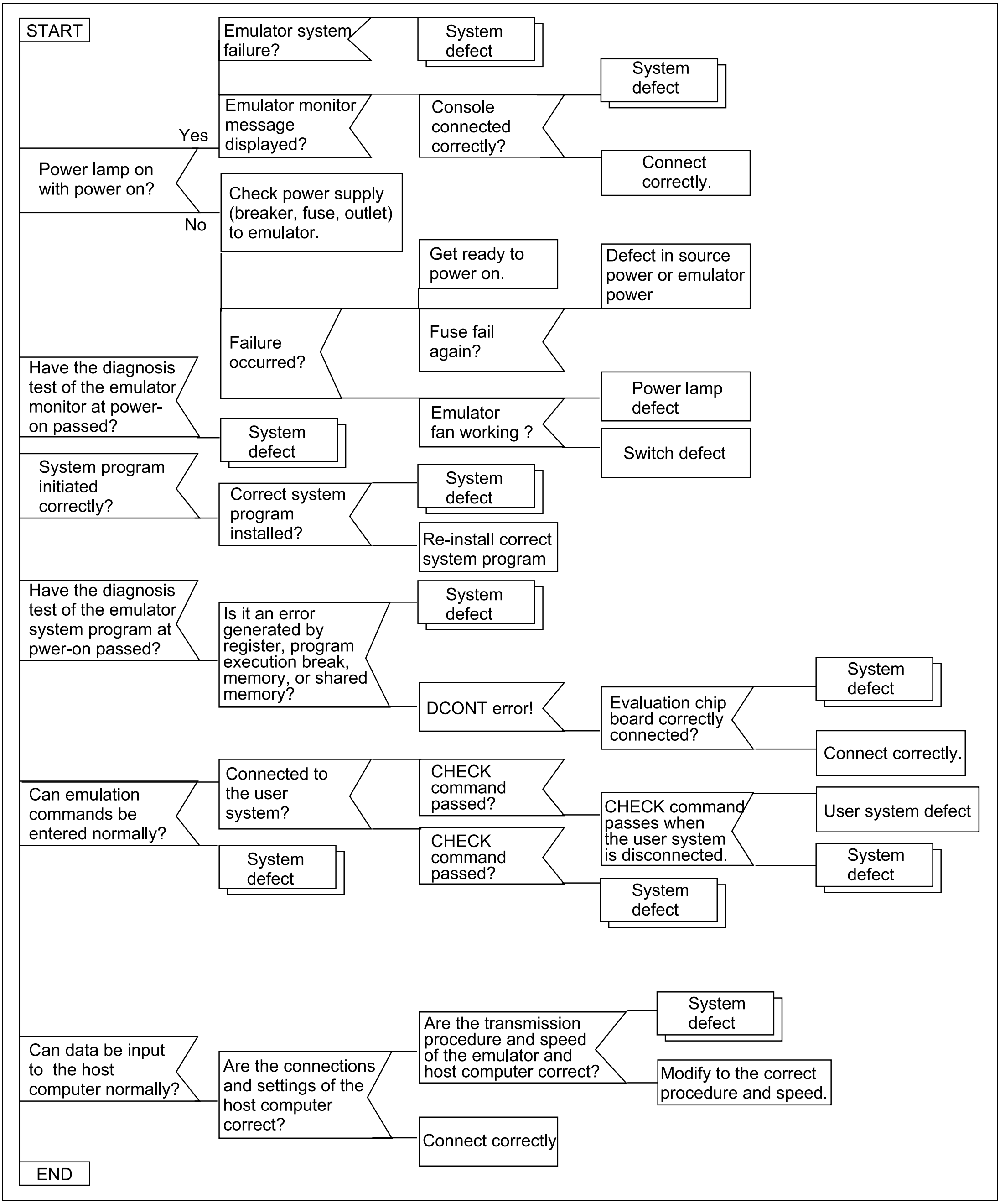


Figure 3.29 Troubleshooting PAD

3.7.4 Operating Procedure for the Diagnostic Program

A description of the diagnostic program is given below.

When the HDI is started up, the following dialog box is displayed.

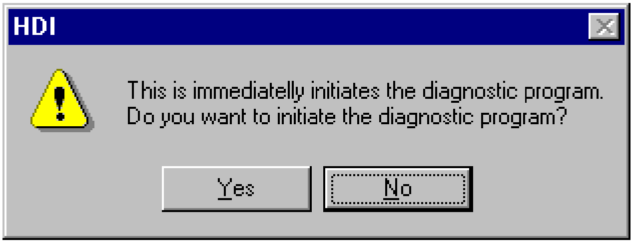


Figure 3.30 Diagnostic Program Initiation Confirmation Dialog Box

When the [Yes] button is clicked, the diagnostic program is initiated, and the following dialog box is displayed.

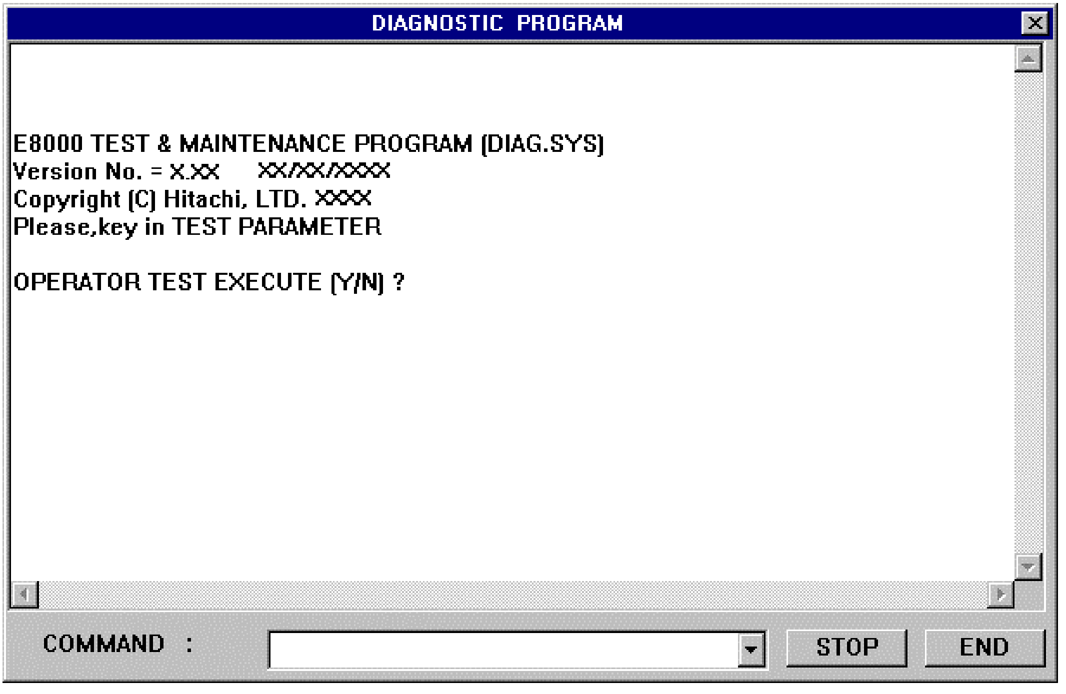


Figure 3.31 [DIAGNOSTIC PROGRAM] Dialog Box

Table 3.5 [DIAGNOSTIC PROGRAM] Dialog Box

Option	Description
Display Area	Displays the results of testing by the diagnostic program.
[COMMAND]	Accepts commands for the diagnostic program.
[STOP]	Terminates testing by the diagnostic program and enters the program's command-input mode.
[END]	Ends the diagnostic program and initiates the HDI.

Set the diagnostic program according to the Descriptive Notes on the Diagnostic Program for the SH7046 E8000S Emulator provided on the CD-R. Click the [END] button or the [STOP] button to terminate the diagnostic program. After the program has been terminated, enter Q in the [COMMAND] edit box. Close the [DIAGNOSTIC PROGRAM] dialog box and restart the HDI.

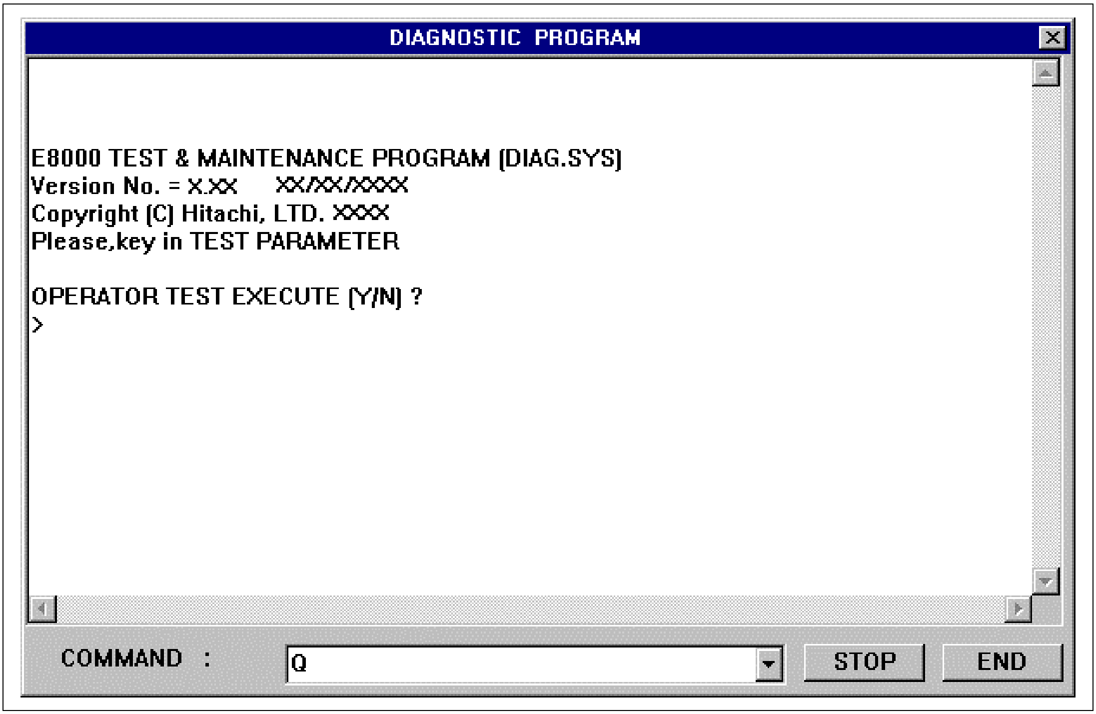


Figure 3.32 Terminating the Diagnostic Program (by Clicking the STOP Button)

A file named E87046.INI will be created in the HDI installation directory. This is a target setting file. Whether or not the diagnostic program should be initiated with the HDI is defined in the Diagnostic Program resource information of the [E8000 HDI TARGET] entry. The diagnostic program is available for installation.

```
[E8000 HDI TARGET]

Diagnostic Program = Y
```

To disable the use of the diagnostic program, modify the Diagnostic Program resource information in the way shown below. The dialog box for confirming the initiation of the diagnostic program will not be displayed on the initiation of the HDI.

Diagnostic Program = N

3.8 Uninstallation

3.8.1 Uninstalling Software

This section describes the procedure for uninstalling software. Be sure to terminate all executing applications before uninstalling software.

Select [Settings]-[Control Panel] from the [Start] menu. Double-click the [Add/Remove Programs] icon.

Select [SH7046 E8000S Emulator Software] from the list of applications installed on the tabbed page [Install/Uninstall] and click the [Add/Remove…] button. Uninstall according to the directions on the screen.

The setup program is executed again to allow the modification, correction, or removal of the installed applications. Select removal to uninstall applications.

Note: Shared files may be detected during uninstallation. Do not remove shared files if they might be used by other HDIs. When Windows NT® 4.0 is used, you might be asked whether you want to remove information registered in the driver's registry. Do not remove information registered in the registry, if it might be used by other HDIs. If other HDIs will no longer start up because of uninstallation, reinstall the other HDIs.

3.8.2 Uninstalling the Acrobat® Reader

Only uninstall the Acrobat® Reader if it is necessary. Click [Settings]-[Control Panel] from the [Start] menu. Double-click the [Add/Remove Programs] icon. Select [Adobe Acrobat Reader x.x] from the list of applications installed on the tabbed page [Install/Uninstall] and click the [Add/Remove…] button. Follow the directions on the screen.

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.

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.

Table 4.1 shows the configuration of the sample program.

Table 4.1 Configuration of the Sample Program

Item No.	Description of File	File Name and Directory
1	HEW workspace file*	\HEW\HDI5\E8000S\7046\TUTORIAL\TUTORIAL.HWS
2	ELF/DWARF2-type load module file	\HEW\HDI5\E8000S\7046 \TUTORIAL\TUTORIAL\DEBUG\TUTORIAL.ABS
3	S-type load module file	\HEW\HDI5\E8000S\7046 \TUTORIAL\TUTORIAL\DEBUG\TUTORIAL.MOT
4	Source file (main program)	\HEW\HDI5\E8000S\7046 \TUTORIAL\TUTORIAL\TUTORIAL.C

Note: This sample program is created by using the SuperH RISC engine C/C++ compiler Package (V6.0.0r1).

4.2 Running the HDI

- To run the HDI, select the [SH7046 E8000S Emulator Software]-[Hitachi Debugging Interface] from the [Start] menu.

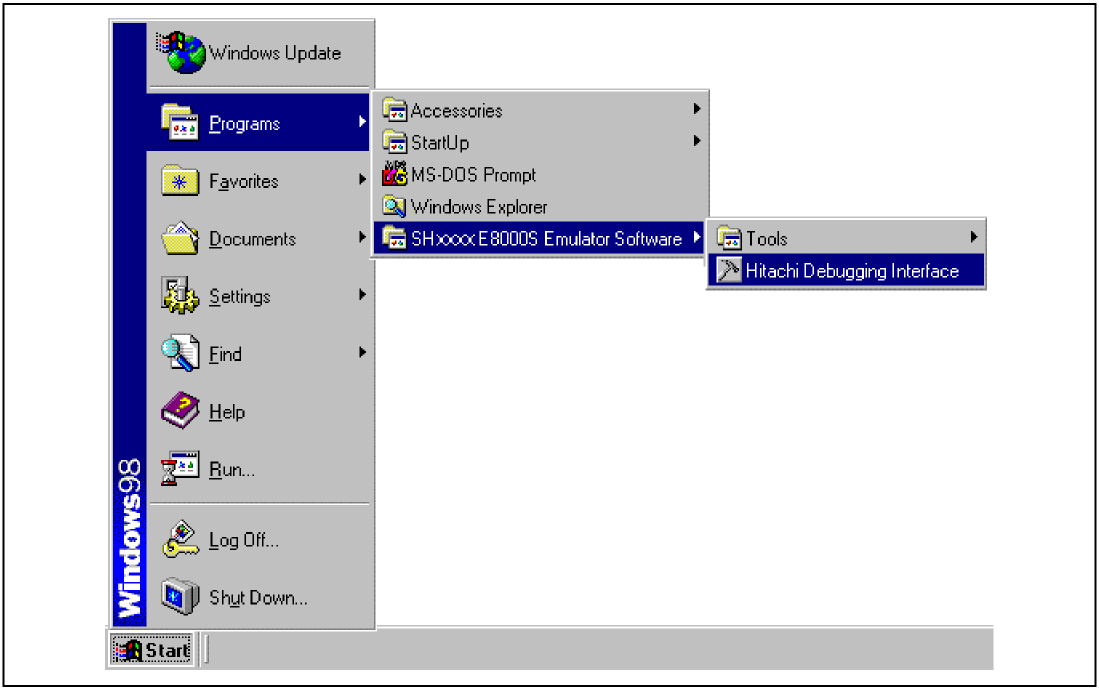


Figure 4.1 [Start] Menu

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.

If the emulator mode is not correctly set, the HDI will not operate normally. In such a case, make the settings related to the MCU, such as the MCU operating mode, in the [CPU Operating Mode] dialog box. Table 4.2 lists the setting examples of the CPU Operating Mode when running the sample program.

Table 4.2 [CPU Operation Mode] Dialog Box Setting Example

Page	Option	Setting Value
[Device (Pin Counts)]	[Operating Mode Auto select (Use Pin Counts and MD3-0 of User System)]	NORMAL
	[Device]	SH7047
	[Pin Counts]	100
[MD Pin]	[Clock Mode (MD3-2)]	Clock Mode 0
	[Operating Mode (MD1-0)]	(MCU) Mode 0
[Pin Select Registers]]	[IRQ0] to [WAIT]	HIGH
[Vcc Down Detection Level]	[CS0] to [CS6]	2.6 V
[H-UDI (JTAG) CLOCK]	[H-UDI (JTAG) CLOCK]	2.5 MHz

Restart the HDI after modifying the [CPU Operating Mode] dialog box. To display the [CPU Operating Mode] dialog box, click the [Setting...] button in the [CPU Operating Mode] page in the [Configuration] dialog box.

For details on the [CPU Operating Mode] dialog box, refer to section 5.2, Setting the Emulator Operating Conditions.

The HDI window is shown in figure 4.2.

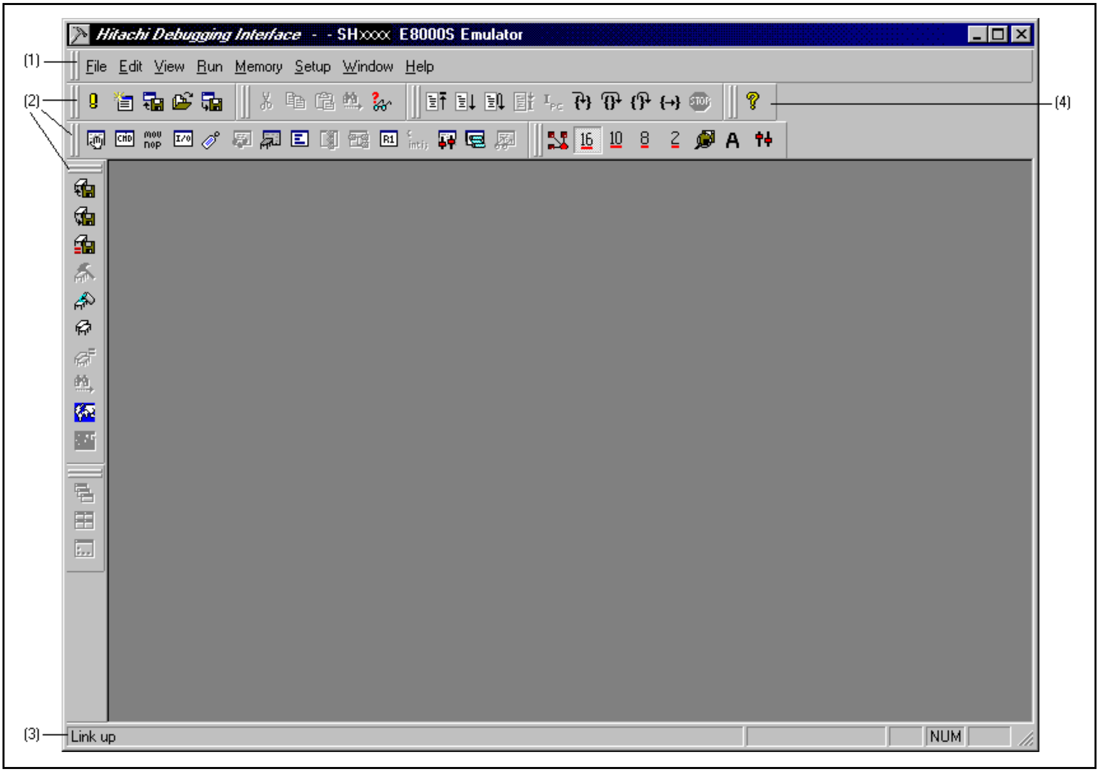


Figure 4.2 HDI Window

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 Map…] from the [Memory] menu to display the current memory map.

The [Memory Mapping] dialog box is displayed.

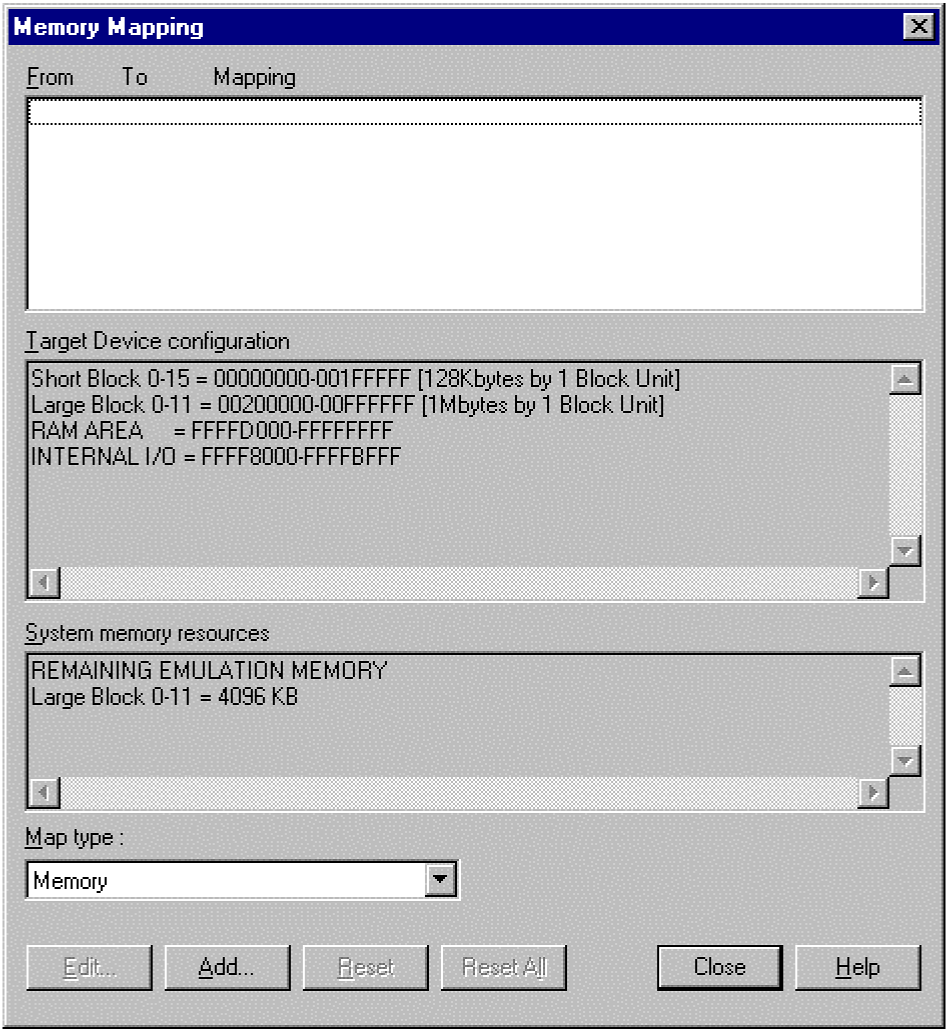


Figure 4.3 [Memory Mapping] Dialog Box (before Setting)

The emulator can allocate emulation memory to CS areas in 4-Mbyte units. The following two types of memory can be specified:

When the [Add...] button is clicked, the [Add Memory Mapping] dialog box is displayed.

In the sample program, allocate emulation memory to memory range H'00000000 to H'003FFFFF (4 Mbytes).

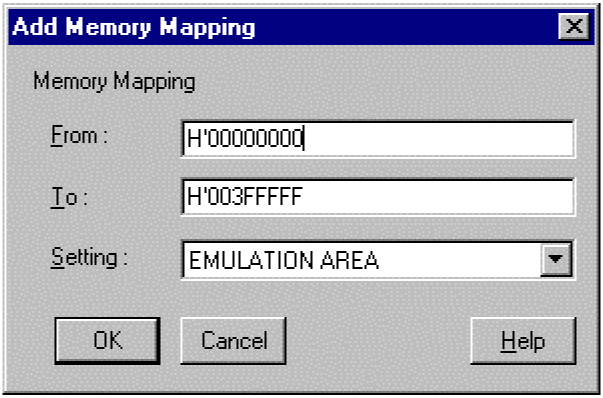


Figure 4.4 [Add Memory Mapping] Dialog Box

- Set the [From] and [To] edit boxes to H’00000000 and H’003FFFFF, respectively, set the [Setting] combo box to [EMULATION AREA], and click the [OK] button.

The [Memory Mapping] dialog box will now show the ranges to which emulation memory is allocated.

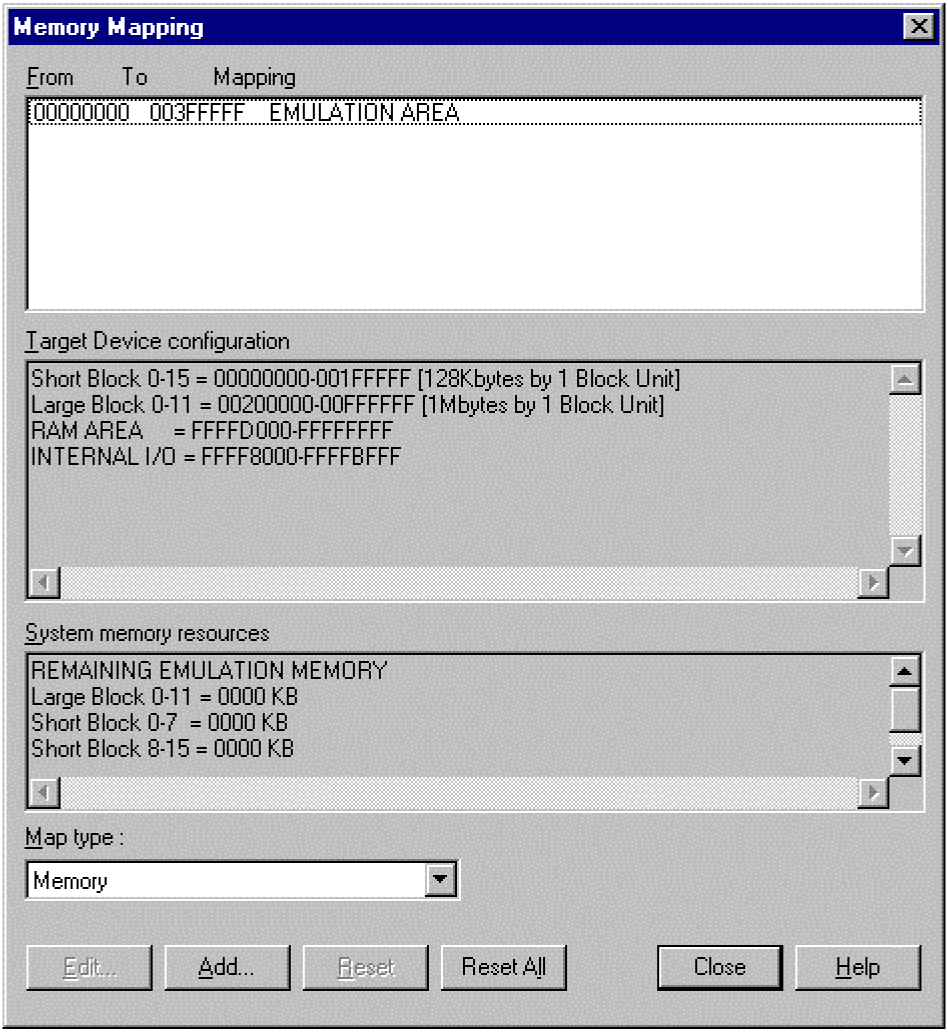


Figure 4.5 [Memory Mapping] Dialog Box (at Setting)

- Click the [Close] button of the [Memory Mapping] dialog box to close the dialog box.

For details on the allocation of emulation memory, refer to section 5.11.1, Emulation Memory Allocation Function.

An internal RAM is used as a stack in the sample program.

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.

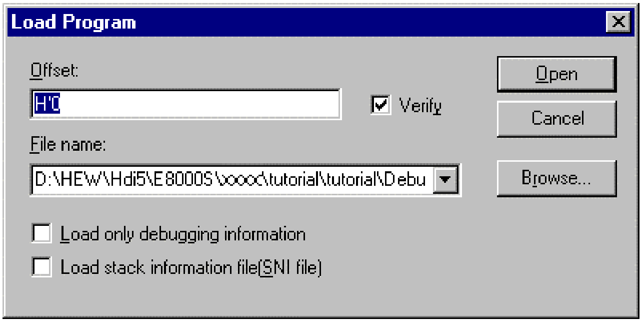


Figure 4.6 [Load Program] Dialog Box

- 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.

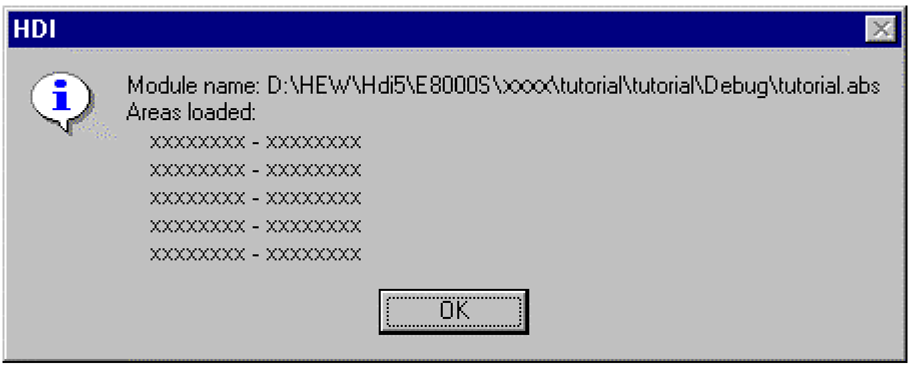


Figure 4.7 HDI Dialog Box

- 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.

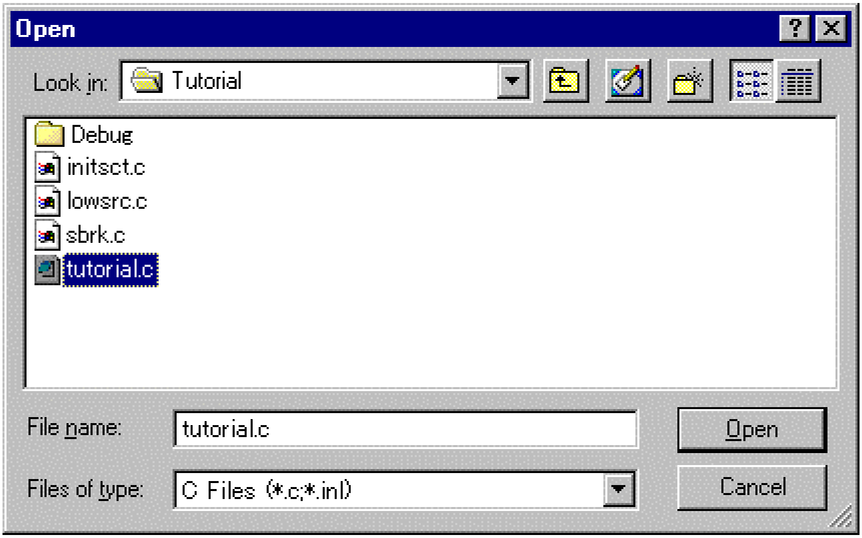


Figure 4.8 [Open] Dialog Box

- Select [Tutorial.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.

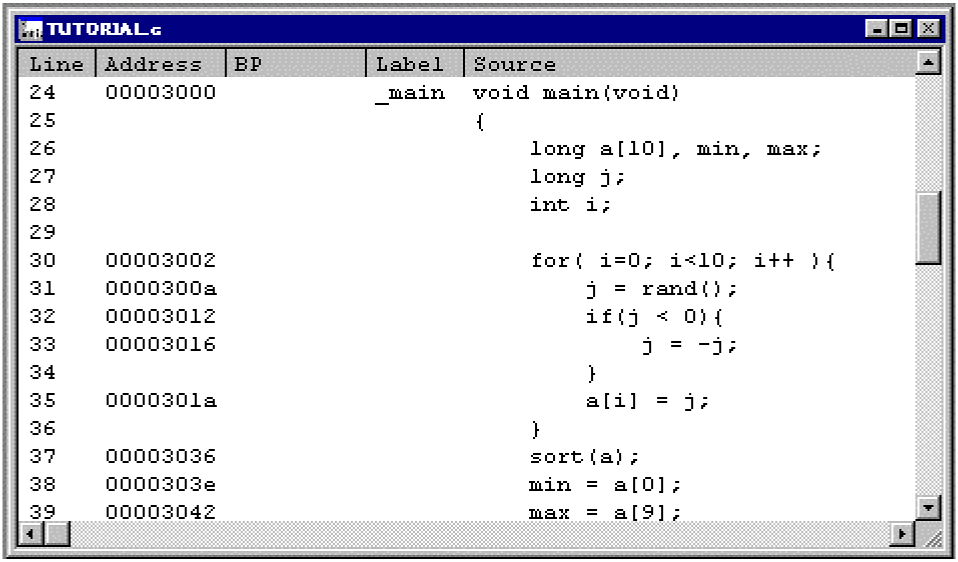


Figure 4.9 [Source] Window (Displaying the Source Program)

4.5 Setting the Software Breakpoints

A breakpoint is one of the debugging functions.

The [Source] window provides a very simple way of setting 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.

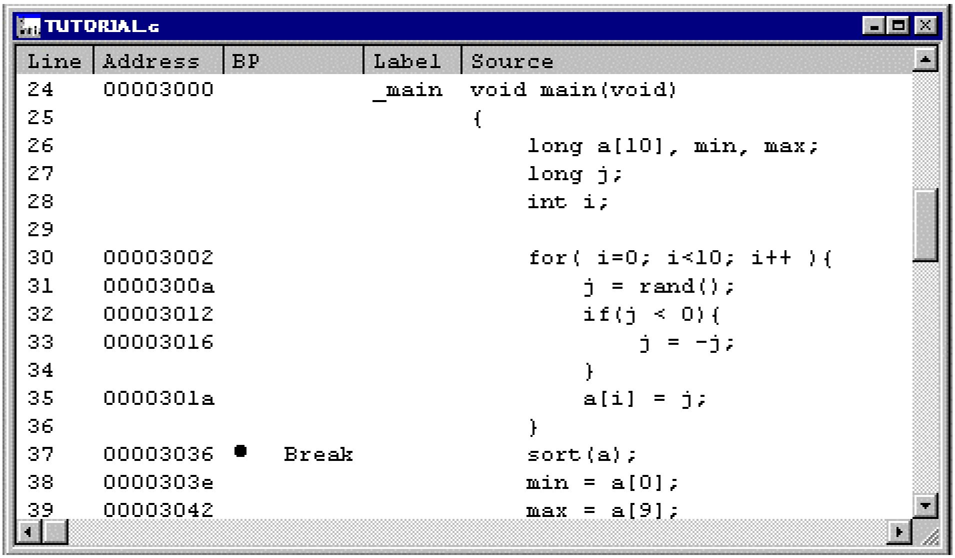


Figure 4.10 [Source] Window (Setting a Software Breakpoint)

The emulator has many break functions. For details, refer to section 5.5, 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.

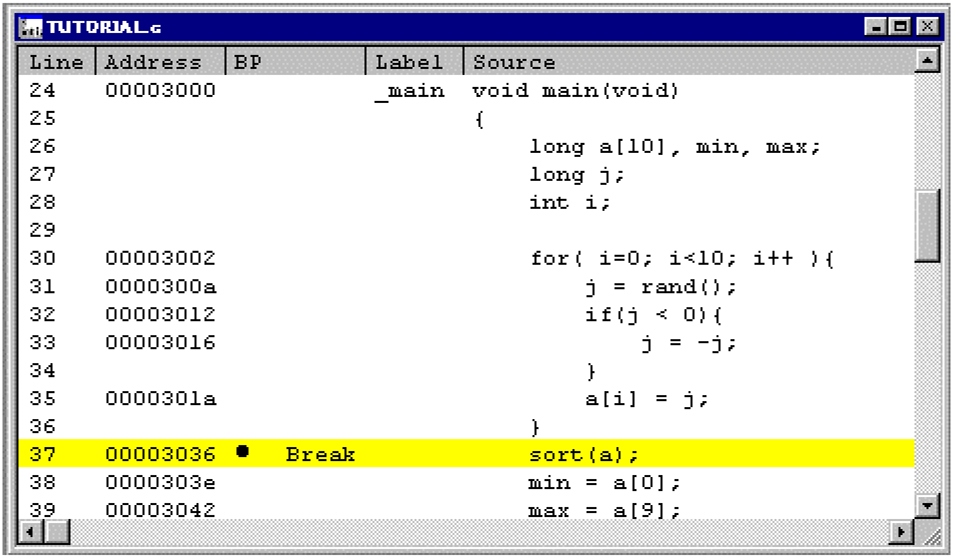


Figure 4.11 [Source] Window (Break State)

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.

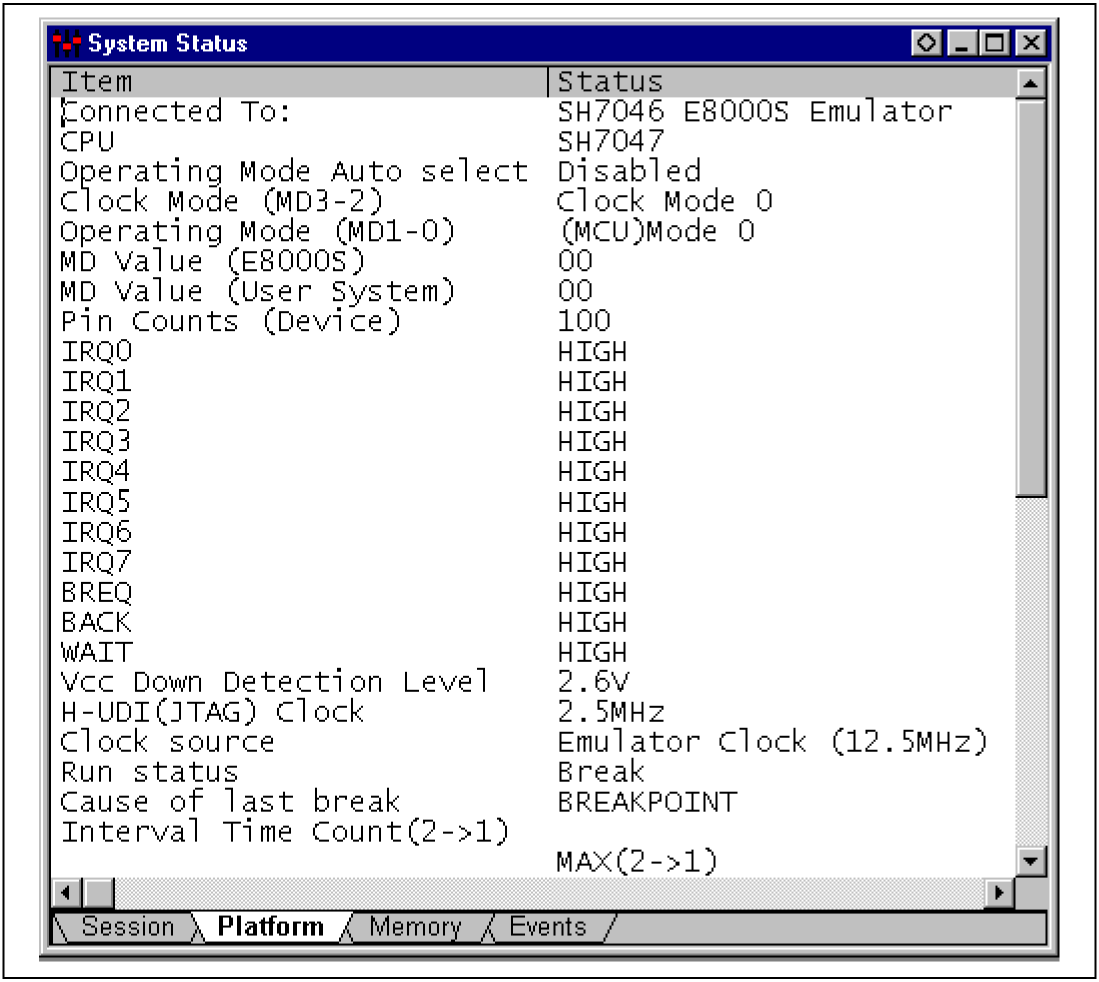


Figure 4.12 [System Status] Window

The [Cause of last break] line shows that the cause of the break is the breakpoint.

For details on program execution, refer to section 5.3, Realtime Emulation.

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.

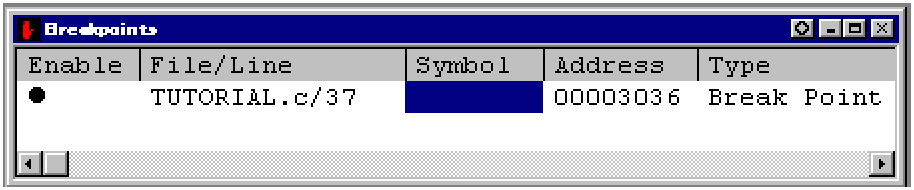


Figure 4.13 [Breakpoints] Window

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].

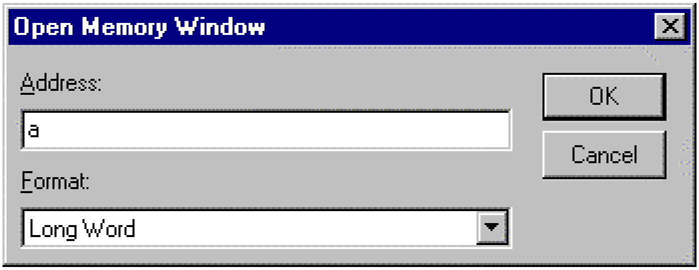


Figure 4.14 [Open Memory Window] Dialog Box

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

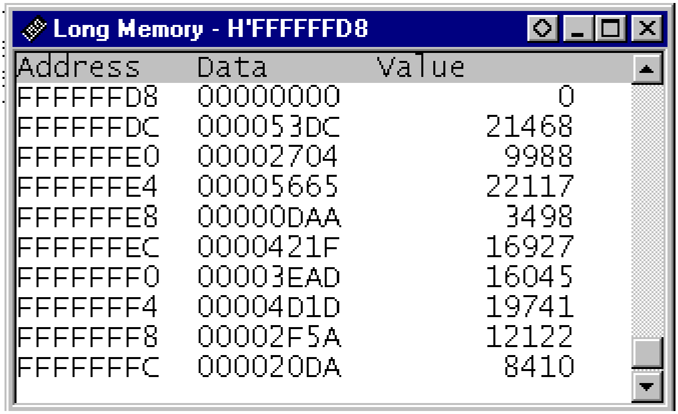


Figure 4.15 [Long Memory] Window

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.

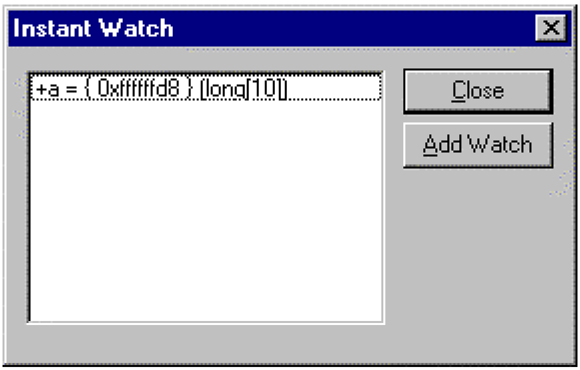


Figure 4.16 [Instant Watch] Dialog Box

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

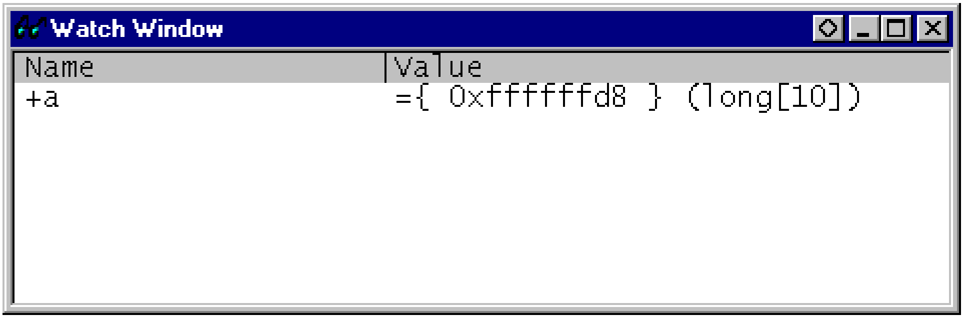


Figure 4.17 [Watch Window] Window (Displaying the Array)

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.

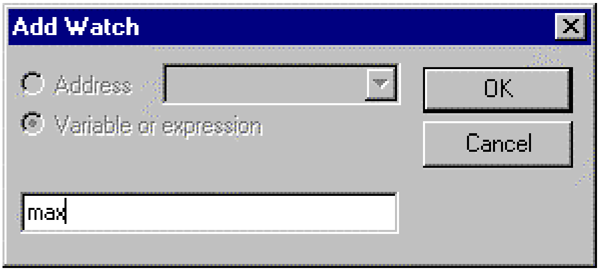


Figure 4.18 [Add Watch] Dialog Box

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

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

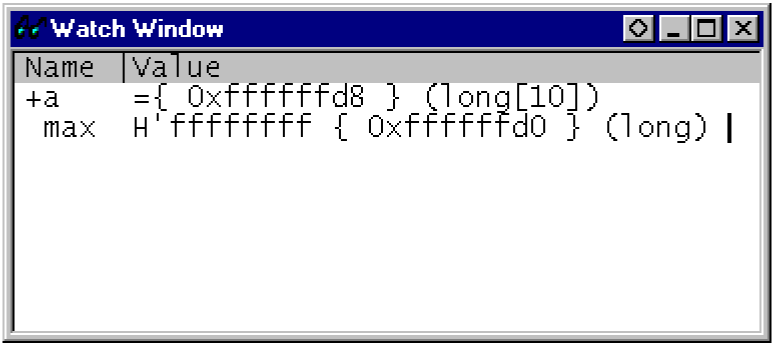


Figure 4.19 [Watch Window] Window (Displaying the Variable)

- 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.

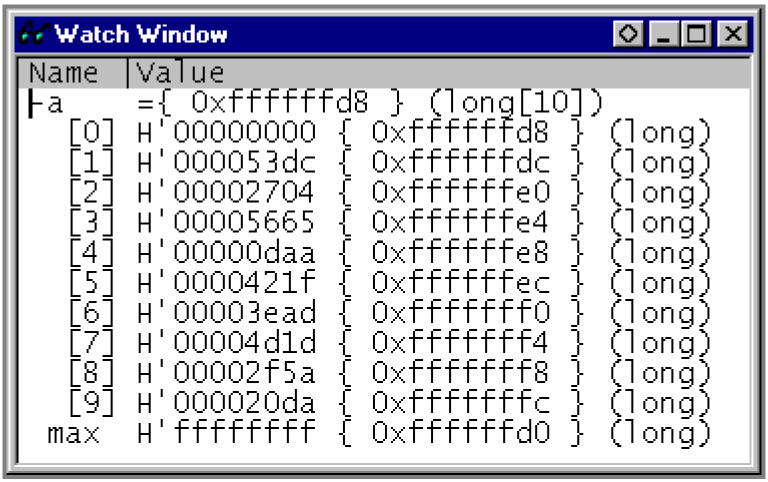


Figure 4.20 [Watch Window] Window (Displaying Array Elements)

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 5.4, Step Function.

Table 4.3 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'00003036.

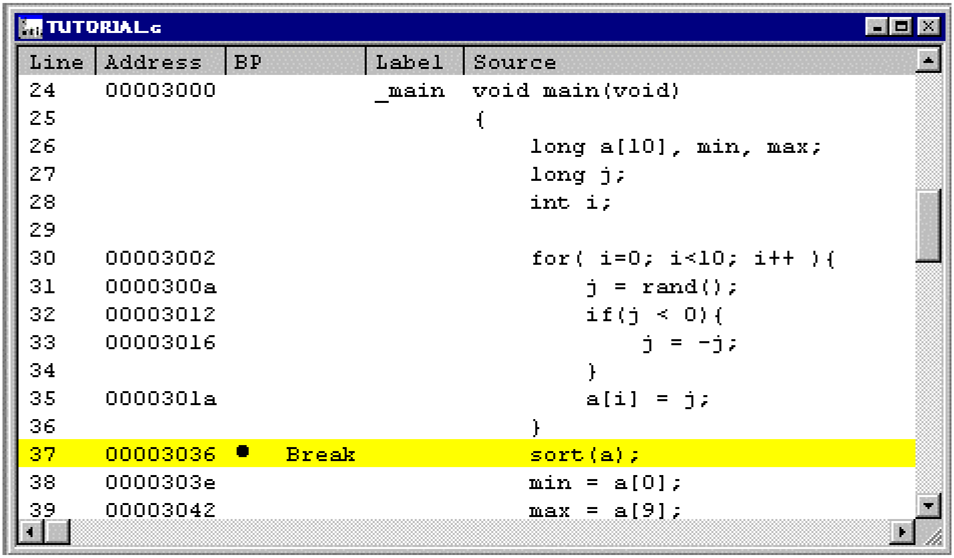


Figure 4.21 [Source] Window (Step Execution)

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.

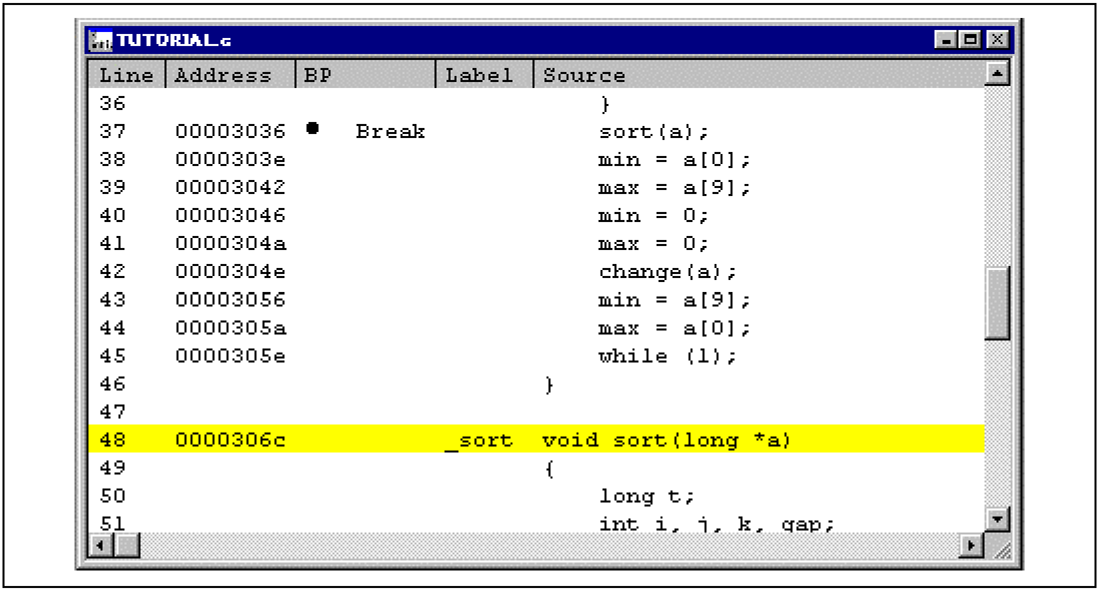


Figure 4.22 [Source] Window (Step In)

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.

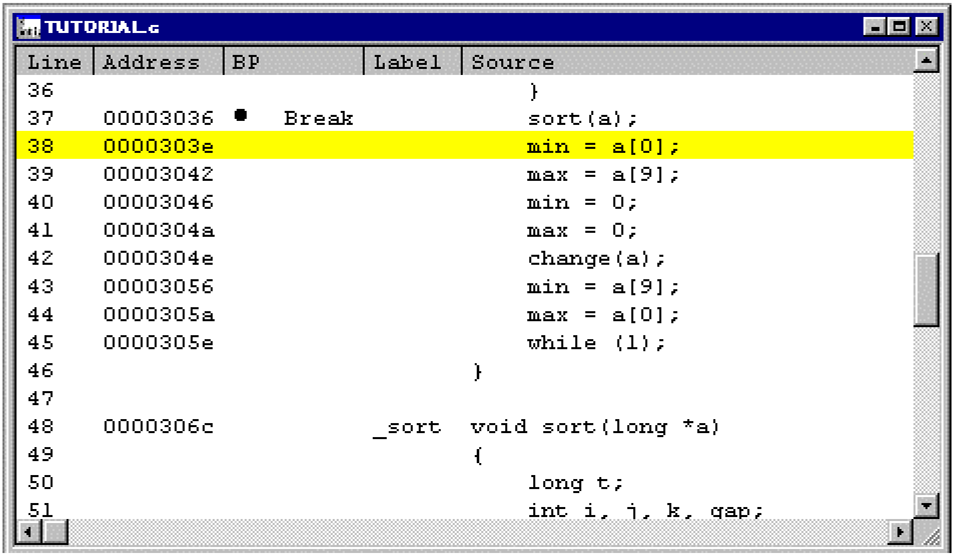


Figure 4.23 [Source] Window (Step Out)

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

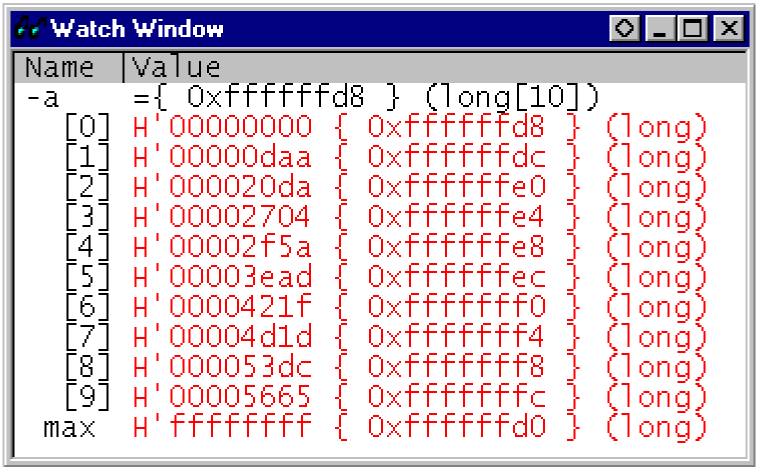


Figure 4.24 [Watch Window] Display Example (1)

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

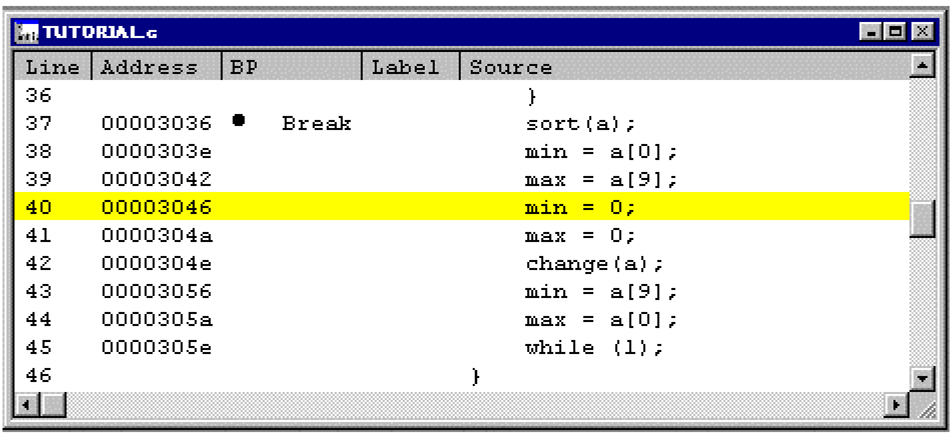


Figure 4.25 [Source] Window (Step Out → Step In)

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

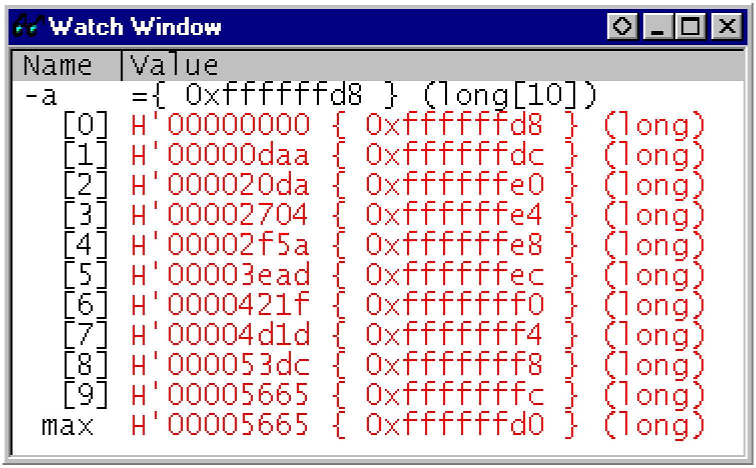


Figure 4.26 [Watch Window] Display Example (2)

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.

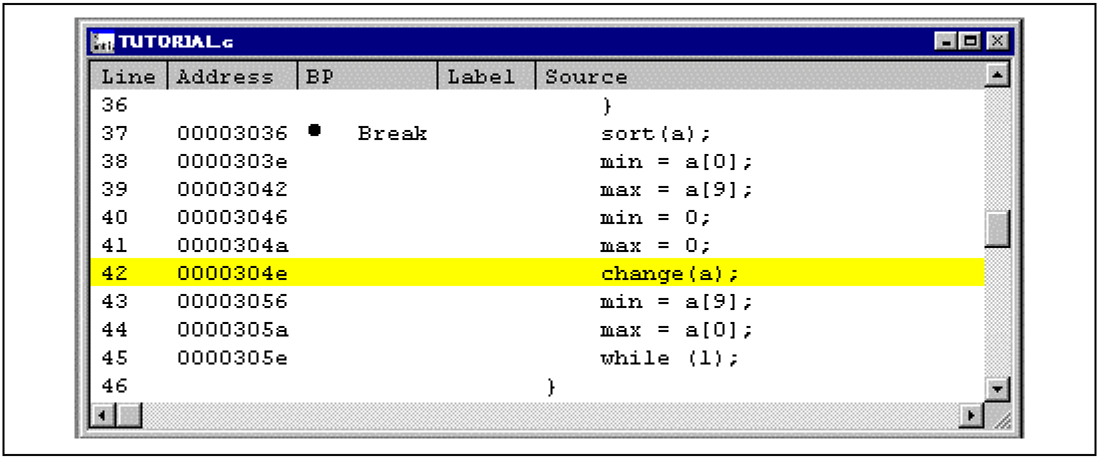


Figure 4.27 [Source] Window (Before Step Over Execution)

- 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.

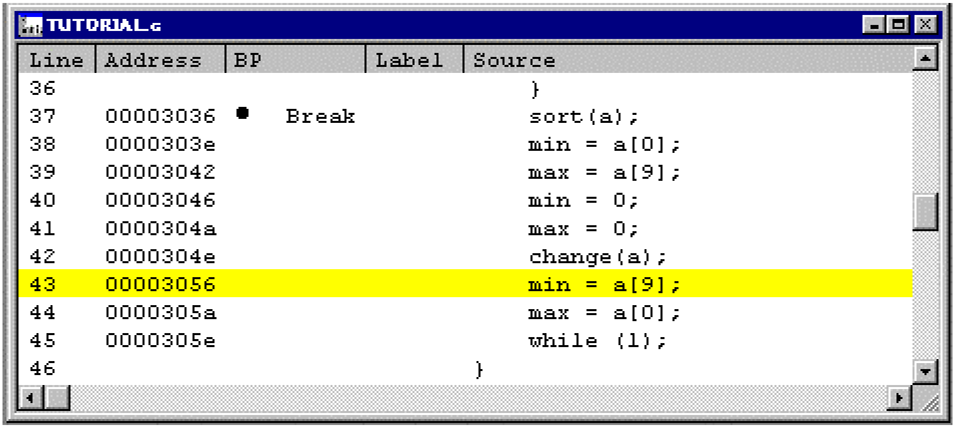


Figure 4.28 [Source] Window (Step Over)

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

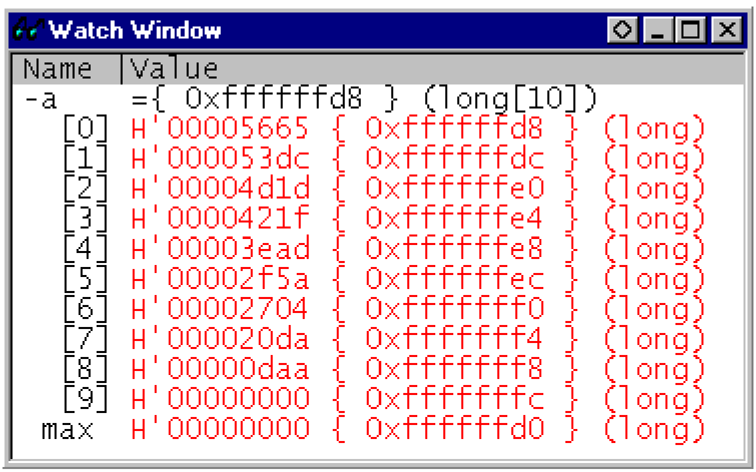


Figure 4.29 [Watch Window] Display Example (3)

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.

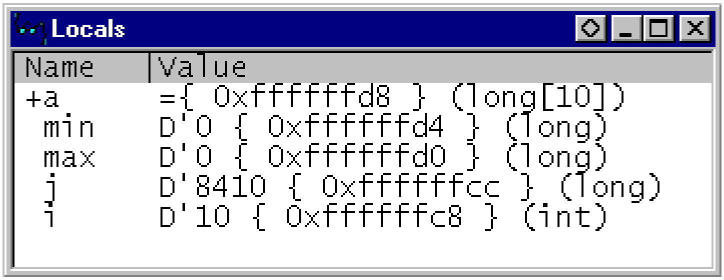


Figure 4.30 [Locals] Window

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.

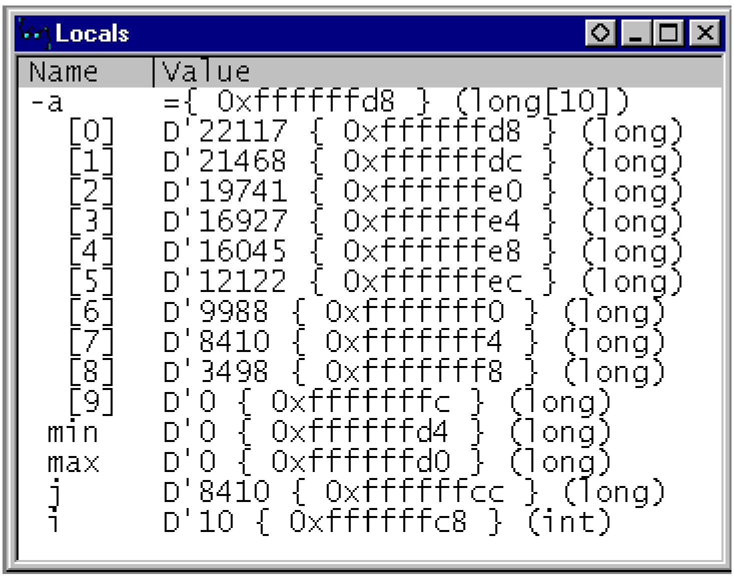


Figure 4.31 [Locals] Window (Displaying Array `a` Elements)

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	Step execution	Emulation with step execution
5.5	Break	Ten types of break function are provided
5.6	Realtime trace	Acquires, searches for, and displays tracing information
5.7	Execution time measurement	Measures the total execution time of the user program
5.8	Performance analysis	Displays measurements of the user program's efficiency of execution during execution
5.9	Informational display	Displays the various items set in each dialog box
5.10	Trigger output	Outputs a low-level pulse on the trigger-output probe
5.11	Memory space	Allocation of emulation memory
5.12	Stack trace	Displays the history of called functions
5.13	Displaying memory and auto-update memory	Displays memory in one of three ways: by parallel access, short breaks, 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	MCU control and status check	Checks the clock inputs to the emulator and the user system
5.15	Entering masks	Formats for the input of masks
5.16	Expanded [Source] window	Expanded [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 (Note: Use the [Configuration] dialog box to make these settings.)	Clock	Selects the clock supplied to drive the MCU
	Conditions for emulation	Sets conditions for emulation in terms of the following items. Operating mode during execution (emulation mode) Interrupts during step execution: Enabled or disabled Memory access during emulation: Enabled or disabled Minimum unit for the execution-time measurement counter Timeout detection period for emulation commands Entering multibreak mode: Enabled or disabled Writing to the internal ROM area: Enabled or disabled Input of control signals: Enabled or disabled Sequential conditions for Condition A and Condition B Controls conditions for trigger output during breaks: Enabled or disabled

Table 5.2 Setting the Emulator's Operating Conditions (cont)

Setting	Item	Description
MCU settings Use the [CPU Operating Mode] dialog box to make these settings.	Operating mode of the user system	Specifies whether the input of the MCU-operating mode pins of the user system is enabled
	Clock mode	Sets the mode for the MCU-driving clock
	Operating mode	Sets the operating mode
	Number of pins for the supported MCU	Sets the number of pins for the supported MCU
	Pin select register	Sets the registers that select the pins for interrupt and other signals
	Vcc Down detection level	Sets the low-Vcc detection level
	H-UDI clock setting	Sets the input clock for the H-UDI (Hitachi Debugging Interface (JTAG)) interface

5.2.1 Configuration Dialog Box

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

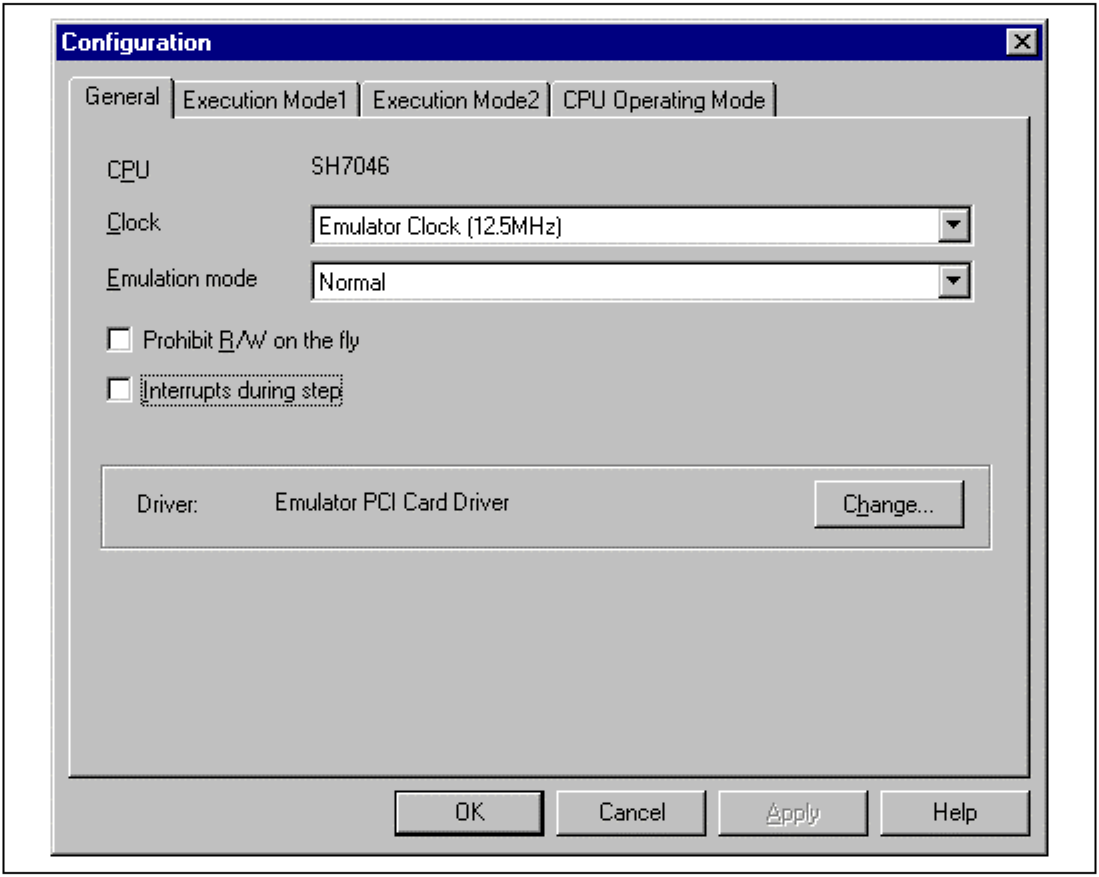


Figure 5.1 [Configuration] Dialog Box

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

Table 5.3 [Configuration] Dialog Box

Page	Description
[General]	Selects the MCU-driving clock and specifies the emulation mode, enabling/disabling of memory access during emulation, whether or not interrupts are accepted during emulation with step execution; and produces a dialog box for the setting of driver software
[Execution Mode1]	Sets the unit of time for counting by the execution-time measurement counter, timeout-detection period for emulation commands, multibreak mode, and whether or not writing to the internal ROM area and several control signals area are enabled
[Execution Mode2]	Sets the conditions for the output of a trigger when a break occurs and sequential conditions Condition A and Condition B (trace, break, or unused)
[CPU Operating Mode]	Sets and displays the MCU operating mode

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

[General] Page:

Use this page to select the MCU-driving clock and specify the emulation mode, enabling/disabling of memory access during emulation, whether or not interrupts are accepted during emulation with step execution; and produce a dialog box for the setting of driver software.

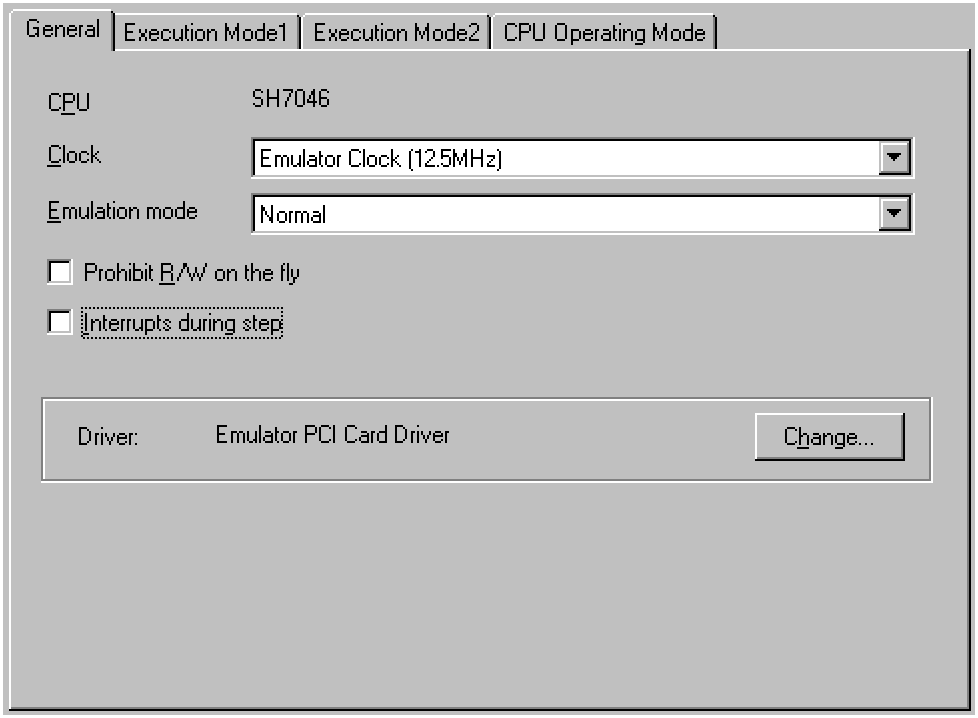


Figure 5.2 [Configuration] Dialog Box [General] Page

Table 5.4 [General] Page

Option	Item	Description
[Clock]	Selects the clock for supply to the MCU	
	Emulator clock (x MHz)	Emulator-internal clock (x: 4, 6, 8, 10, or 12.5)
	User clock	Clock signal from the user system
	X'TAL	Crystal oscillator on the evaluation chip board
[Emulation mode]	Selects the operating mode for emulation	
	Normal	Normal emulation (initial value)
	Cycle Reset x	Issues a forced RES signal to the MCU then continues emulation after a specified interval (cycle-reset mode) (x: 6.5us, 9.8us, 50us, 100us, 500us, 1ms, 5ms, 10ms, 50ms, 100ms, 500ms, or 1s)
	Break Condition U Sequential x	Uses an internal break (Break Condition U1 to U4) or reset condition to execute a sequential break (x: 2 -> 1, 3 -> 2 -> 1, or 4 -> 3 -> 2 -> 1)
	Timeout break of Performance Analysis	Emulation breaks when the Performance Analysis 1 timeout specification or maximum number of passes specification is satisfied
	Timeout trace of Performance Analysis	Trace stops when the Performance Analysis 1 timeout specification or maximum number of passes specification is satisfied
	Time interval Measurement mode x	Executes emulation in time interval measurement mode 1, 2, or 3 (x: 1, 2, or 3)
	Time interval Measurement (Condition B)	Measures the execution time over which a specific condition is satisfied by using Break Condition B or Trace Condition B
	No Break	Emulation with all break conditions disabled
[Prohibit R/W on the fly]	Selects whether or not to disable access to memory during user-program execution (initial value: Disabled)	
[Interrupts during step]	Selects whether or not to accept interrupts during step execution (initial value: Disabled)	
[Drivers]	Opens the [Driver Details] dialog box	

[Execution Mode1] Page:

Use this page to set the unit of time for counting by the execution-time measurement counter, timeout-detection period for emulation commands, multibreak mode, and whether or not writing to the internal ROM area and the input of the RES, BREQ, and WAIT signals is enabled.

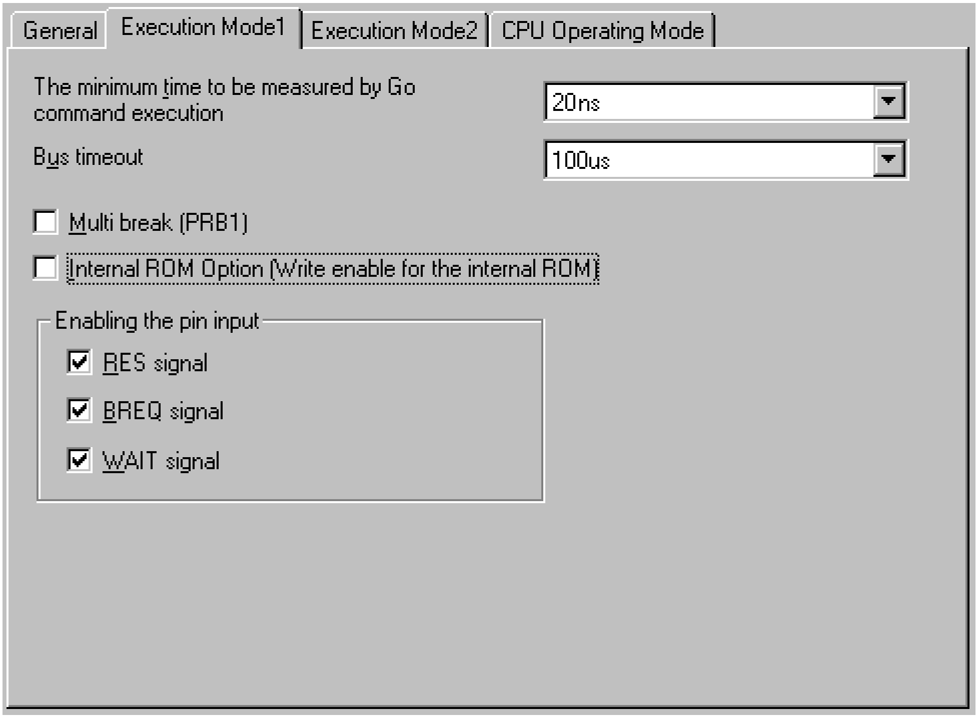


Figure 5.3 [Configuration] Dialog Box [Execution Mode1] Page

Table 5.5 [Execution Mode1] Page

Option	Description
[The minimum time to be measured by Go command execution]	Sets the unit of time to be counted by the counter for measuring execution time. Select 52 us, 1.6 us, or 20 ns. (Initial value: 20 ns)
[Bus timeout]	Sets the timeout detection period for emulation commands. Select 100 µs, 1.6 ms, 13 ms, or 210 ms. (Initial value: 100 us)
[Multi break (PRB1)]	Selects whether or not the multibreak function (uses external probe 1 to break execution by multiple emulators, one after another) is enabled during execution (Initial value: Disabled)
[Internal ROM Option (Write enable for the internal ROM)]	Selects whether or not writing to the internal ROM area is enabled. (Initial value: Disabled)
[RES signal]	Selects whether or not the input of the RES signals is enabled. (Initial value: Enabled)
[BREQ signal]	Selects whether or not the input of the BREQ signals is enabled. (Initial value: Enabled)
[WAIT signal]	Selects whether or not the input of the WAIT signals is enabled. (Initial value: Enabled)

[Execution Mode2] Page:

Use this page to set sequential conditions Condition A and Condition B (trace, break, or unused) and to set the conditions for the output of a trigger when a break occurs.

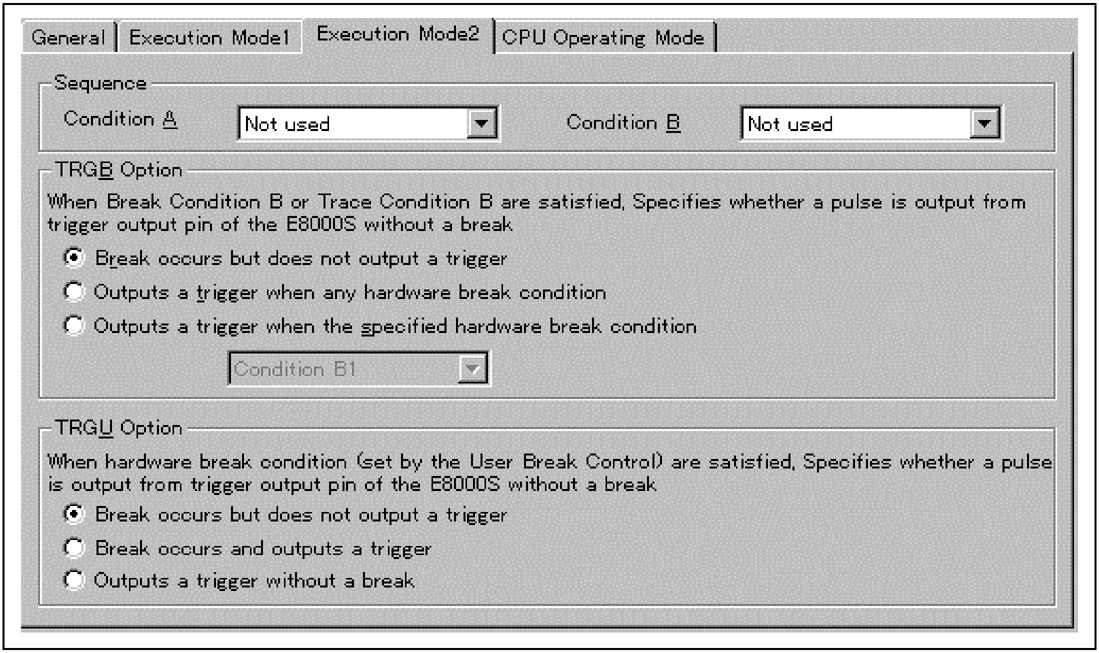


Figure 5.4 [Configuration] Dialog Box [Execution Mode2] Page

Table 5.6 [Execution Mode2] Page

Option	Description
[Sequence]	Sets a sequential break or trace. Not used: The conditions in [Condition A] are not set as sequential break or trace conditions. The conditions in [Condition B] are not set as sequential break or trace conditions (Initial value) Break: The conditions in [Condition A] are set as sequential break conditions. The conditions in [Condition B] are set as sequential break or conditions Trace: The conditions in [Condition A] are set as sequential trace conditions. The conditions in [Condition B] are not set as sequential trace conditions
[TRGB Option]	Selects a condition for the output of a pulse from the trigger output pin of the emulator when the condition set in the [Condition B] dialog box is satisfied. Condition 1: When a condition among channels 1 to 8 is satisfied, a break occurs, and no trigger is output. (Initial value) Condition 2: When a condition among channels 1 to 8 is satisfied, a trigger is output. Condition 3: When a condition of a specified channel is satisfied, a trigger is output. The channel number can be selected from the combo box.
[TRGU Option]	Selects a condition for the output of a pulse from the trigger output pin of the emulator when the condition set in the [Break Condition U1, 2, 3, 4] dialog box is satisfied. Condition 1: A break occurs, and no trigger is output. (Initial value) Condition 2: A trigger is output. Condition 3: No break occurs, and a trigger is output. Select one of the above three conditions.

Note: For details on selecting a sequential break or trace condition, refer to sections 5.5, Break Functions, and 5.6, Realtime Trace Functions.

[CPU Operating Mode] Page:

Use this page to set and display the details of the MCU’s operating mode.

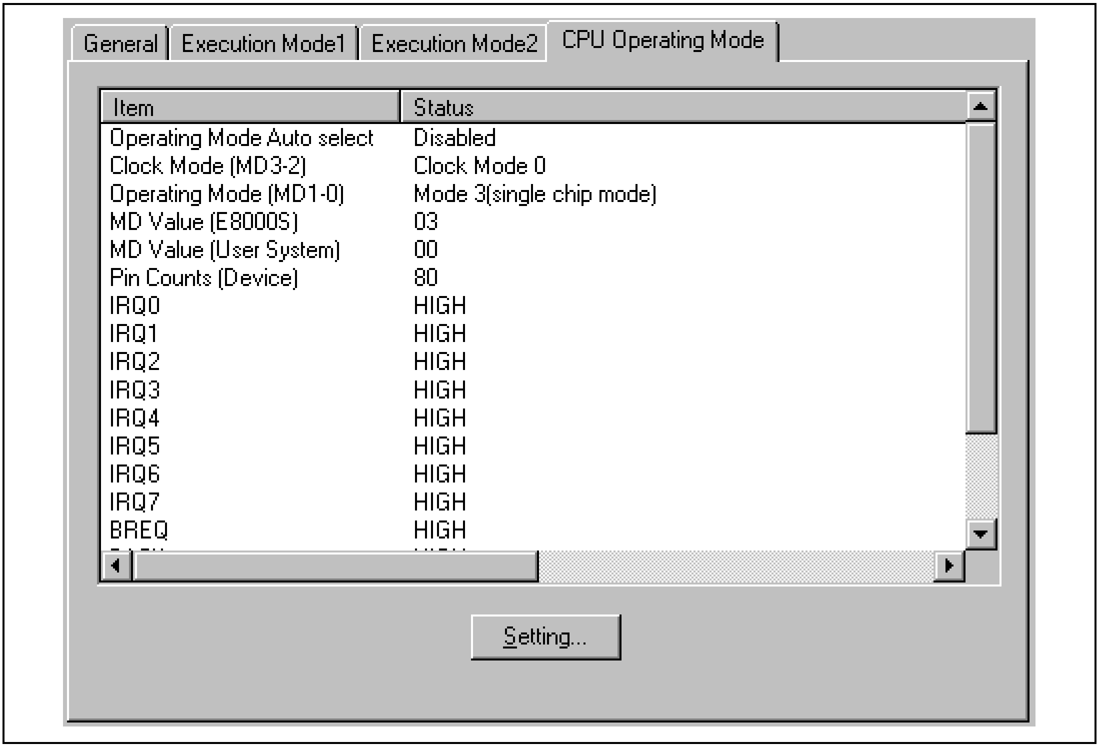


Figure 5.5 [Configuration] Dialog Box [CPU Operating Mode] Page

Table 5.7 [CPU Operating Mode] Page

Option	Description
[Item]	Displays a list of the [CPU Operating Mode] items
[Status]	Displays the current setting for the item
[Setting...]	Displays the [CPU Operating Mode] dialog box that is used to change the settings. When [Finish] is clicked in the [CPU Operating Mode] dialog box, the emulator restarts and ignores the settings made in other dialog boxes.

5.2.2 [CPU Operating Mode] Dialog Box

Use this ‘wizard’-style dialog box to set the MCU’s operating mode. This dialog box has four pages, and they are listed in table 5.8. Changing the settings in the [CPU Operating Mode] dialog box will restart the HDI.

Table 5.8 [CPU Operating Mode] Dialog Box

Page	Description
[Device (Pin Counts)]	Selects the target MCU.
[MD Pin]	Sets the MCU operating mode.
[Pin Select Registers]	Sets the registers for selecting pins.
[Vcc Down Detection Level]	Selects the detection level of Vcc Down.
[H-UDI (JTAG) clock]	Sets the input clock to the H-UDI (JTAG interface)

Note: Settings made in the [CPU Operating Mode] dialog box are saved in the emulatorand not in the session file. The initial values on each page are the values initially installed with the system files. If the settings have changed, the initial values at initiation are the changed values.

[Device (Pin Counts)] Page:

Use this page to select the target MCU. Set the conditions on each page as required, then click the [Next] button.

Figure 5.6   [CPU Operating Mode] Dialog Box [Device (Pin Counts)] Page

Table 5.9 [Device (Pin Counts)] Page

Item	Description	
[Operating Mode Auto select]	This setting is activated when the user system is connected, and allows automatic detection of the number of pins on the user system and the settings of the MD pins. When the user system is not connected, the settings under [Pin Counts] and on the [MD Pin] pages are used (initial value: Disabled).	
[Device]	Selects the target MCU. Select Custom to specify the number of pins on the supported MCU under [Pin Counts] (initial value: SH7046).	
[Pin Counts]	Specifies the number of pins of the supported MCU.	
	80	80 pins
	100	100 pins
	112	112 pins
	144	144 pins
[Next >]	Goes to the next page [MD Pin]	
[Cancel]	Cancels changes made for the settings.	

Note: Custom is for use in the future as the product range is developed and expanded. Do not select Custom except in special cases.

[MD Pin] Page:

Use this page to set the MCU operating mode. Make the settings correctly then click the [Next] button.

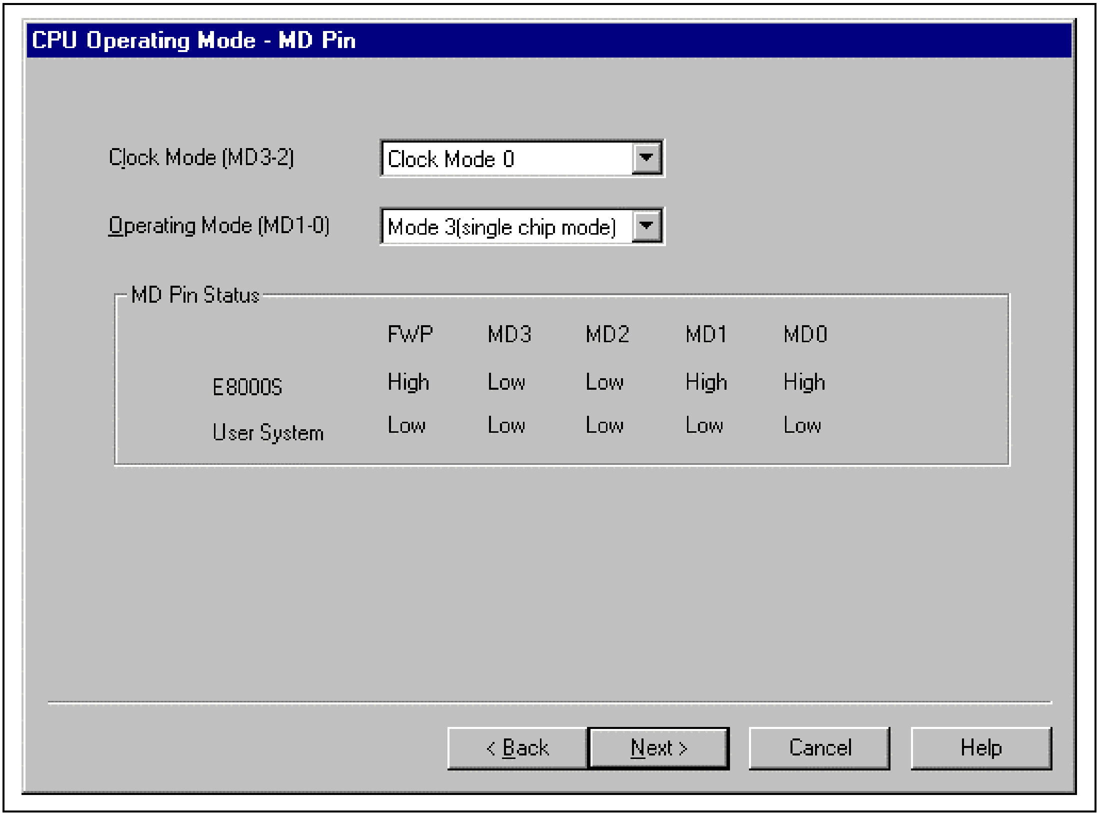


Figure 5.7 [CPU Operating Mode] Dialog Box [MD Pin] Page

The MCU operating mode specified on the [MD Pin] page may not be the MCU operating mode which is actually applied. You can check which MCU operating mode is currently being used in the [System Status] window (for details on the various modes, refer to appendix E.9, CPU Operating Modes).

Table 5.10 [MD Pin] Page

Option	Description	
[Clock Mode (MD3-2)]	Sets the clock mode.	
	Clock Mode 0	Clock mode 0 (initial value)
	Clock Mode 1	Clock mode 1
	Clock Mode 2	Clock mode 2
	Clock Mode 3	Clock mode 3
[Operating Mode (MD1-0)]	Sets the operating mode	
	(MCU) Mode 0	MCU expansion mode 0
	(MCU) Mode 1	MCU expansion mode 1
	(MCU) Mode 2	MCU expansion mode 2
	Mode 3 (single-chip mode)	Single-chip mode (initial value)
[MD Pin Status]*	E8000S	Indicates the Clock Mode [MD3-2] and Operating Mode [MD1-0] settings as High or Low
	User System	Indicates the state of the FWP and MD3 to 0 pins in the user system
[< Back]	Returns to the [Device (Pin Counts)] page	
[Next >]	Goes to the next page [Pin Select Registers]	
[Cancel]	Cancels changes made for the settings	

Note: The emulator does not support the programming mode of the flash memory. The state of the FWP pin is always high.

[Pin Select Registers] Page:

Sets the pins selected by the pin function controller (PFC). The emulator sets these pins correctly corresponding to the signals for use in the following functions.

- External interrupt signal conditions for hardware breaks and tracing
- Detecting the WAIT signal for display in the emulation-state display

Set the conditions then click the [Next] button.

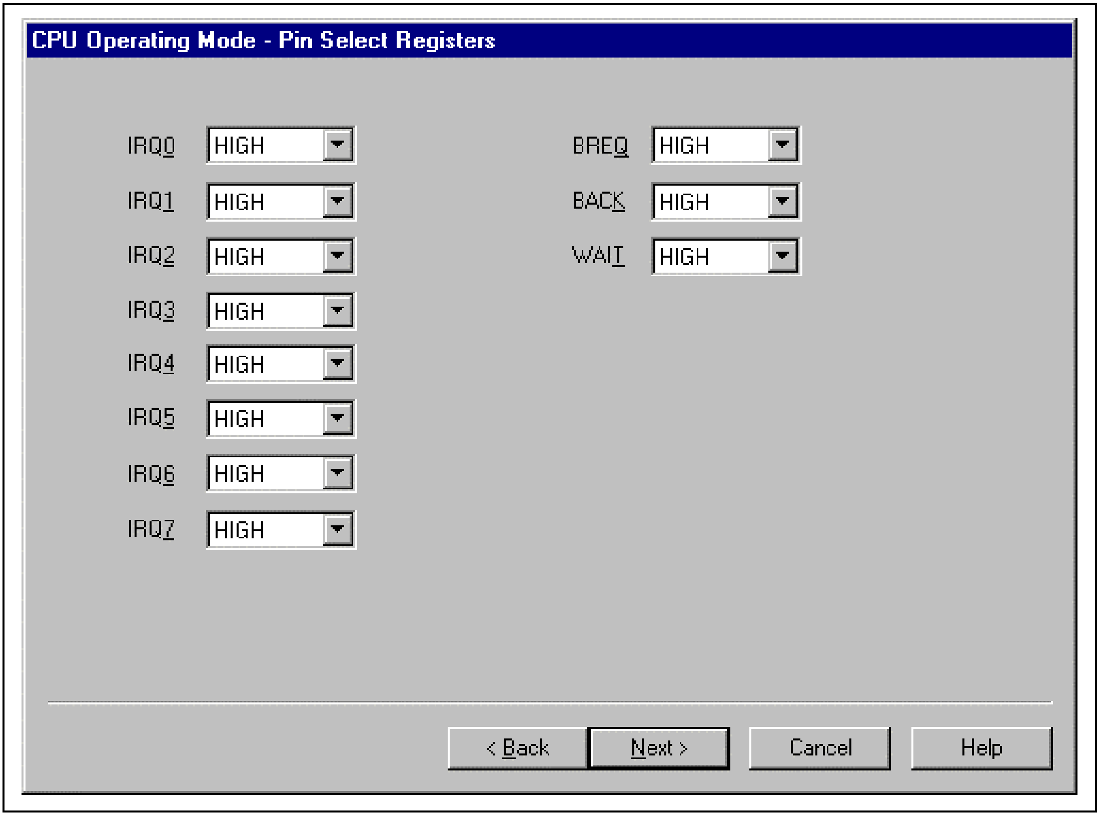


Figure 5.8 [CPU Operating Mode] Dialog Box [Pin Select Registers] Page

Table 5.11 [Pin Select Registers] Page

Option	Description
[IRQ0]	Sets the pins (PA2, PB2, PD16, HIGH) that correspond to the IRQ0 signal.
[IRQ1]	Sets the pins (PA5, PB3, PD17, HIGH) that correspond to the IRQ1 signal.
[IRQ2]	Sets the pins (PA8, PB4, PD18, HIGH) that correspond to the IRQ2 signal.
[IRQ3]	Sets the pins (PA9, PB5, PD19, HIGH) that correspond to the IRQ3 signal.
[IRQ4]	Sets the pins (PB6, PD20, HIGH) that correspond to the IRQ4 signal.
[IRQ5]	Sets the pins (PB7, PD21, HIGH) that correspond to the IRQ5 signal.
[IRQ6]	Sets the pins (PB8, PD22, HIGH) that correspond to the IRQ6 signal.
[IRQ7]	Sets the pins (PB9, PD23, HIGH) that correspond to the IRQ7 signal.
[BREQ]	Sets the pins (PA13, PB7, HIGH) that correspond to the BREQ signal.
[BACK]	Sets the pins (PA15, PB6, HIGH) that correspond to the BACK signal.
[WAIT]	Sets the pins (PA17, PA17, PB8, PE17, HIGH) that correspond to the WAIT signal.
[< Back]	Returns to the [MD Pin] page
[Next >]	Goes to the next page [Vcc Down Detection Level]
[Cancel]	Cancels changes made for the settings

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

[Vcc Down Detection Level] Page:

Sets the detection level of Vcc Down. Set the conditions then click the [Next] button.

Figure 5.9 [CPU Operating Mode] Dialog Box [Vcc Down Detection Level] Page

Table 5.12 [Vcc Down Detection Level] Page

Option	Description
[Vcc Down Detection Level]	Sets the detection level of Vcc Down. Select 2.6 V (initial value) or 4.0 V
[< Back]	Returns to the [Pin Select Registers] page
[Next >]	Goes to the next page [H-UDI (JTAG) Clock]
[Cancel]	Cancels changes made for the settings.

Note: Set the level for detecting supply of a low voltage on the Vcc pin.

[H-UDI (JTAG) Clock] Page:

Sets the H-UDI (JTAG interface) input clock. Set the conditions and click the [Finish] button to close the [CPU Operating Mode] dialog box. The HDI will be restarted.

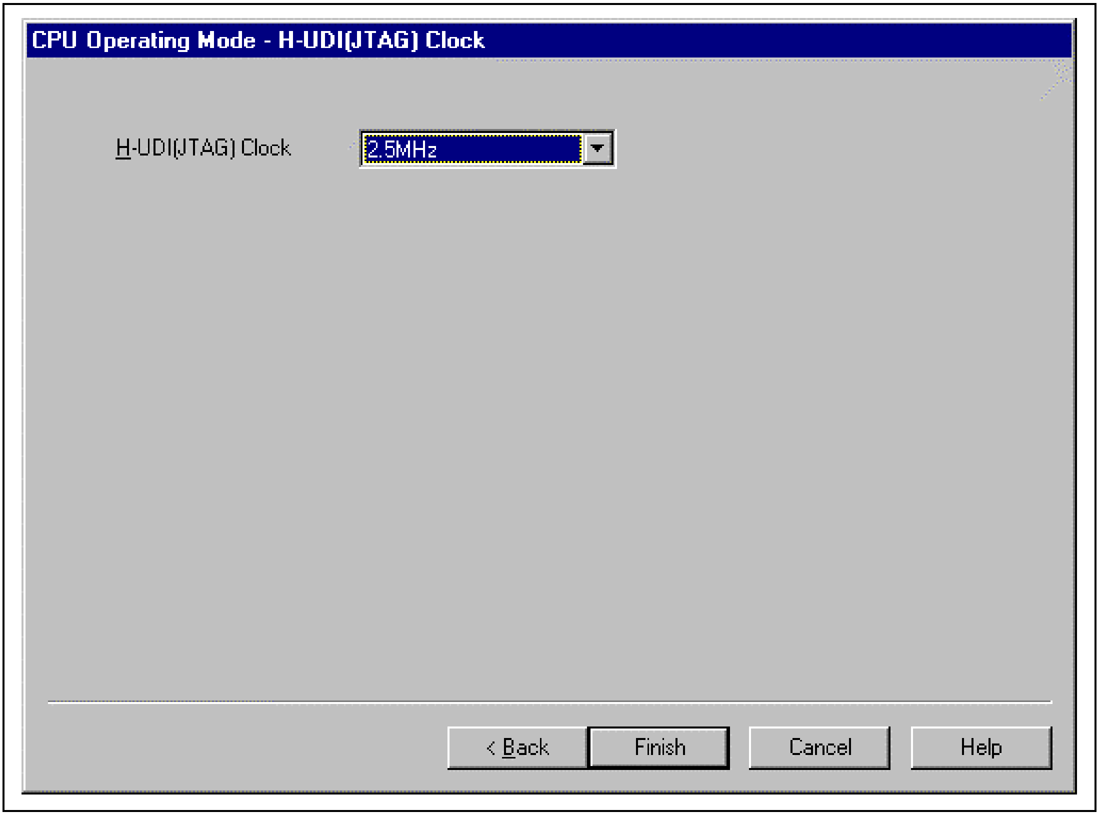


Figure 5.10 [CPU Operating Mode] Dialog Box [H-UDI (JTAG) Clock] Page

Table 5.13 [H-UDI (JTAG) Clock] Page

Option	Description
[H-UDI (JTAG) Clock]*	Sets the frequency of the clock for input to the H-UDI. Select 2.5 MHz (initial value), 5 MHz, 10 MHz, 20 MHz, CPU clock, CPU clock 1/2, CPU clock 1/4, or CPU clock 1/8. (CPU clock is a system clock (ϕ).)
[< Back]	Returns to the [Vcc Down Detection Level] page
[Finish]	Sets the emulator according to the content of each page, then re-activates the emulator
[Cancel]	Cancels changes made for the settings

Note: The input clock for the H-UDI must be set at a frequency lower than that of the clock that drives the peripheral module (Pϕ).

5.3 Realtime Emulation

5.3.1 Execution

Table 5.14 shows the main forms of realtime execution.

Table 5.14 Settings in the [Configuration] Dialog Box

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, obtains the PC and SP (stack-pointer) values from the reset vector, then execute the user program.	Click the [Reset Go] button
		Select [Reset Go] from the [Run] menu
Execution from a specified address	Executes the user program from 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
		Specify a start address for the [Program Counter] in the [Run] dialog box, then click the [Go PC] button
Execution to a specified address	Specifies the end address, and executes the user program up to that address.	Specify a start address for the [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.

Any of the various modes of emulation listed in table 5.15 can be specified by selecting [Emulation mode] from the [General] page of the [Configuration] dialog box.

Table 5.15 Emulation Modes

Emulation Mode	Description	[Emulation mode] Setting
Cycle reset mode	The emulator periodically inputs the RES signal to the MCU during realtime emulation and execution from the reset state is thus periodically repeated. When the RES signal is input to the MCU, a low-level pulse is concurrently output on the trigger-output probe. This function is useful for observing waveforms from the initial state, for example during a power-on-reset up to a specified time. The reset intervals can be selected from among 6.5 µs, 9.8 µs, 50 µs, 100 µs, 500 µs, 1 ms, 5 ms, 10 ms, 50 ms, 100 ms, 500 ms, or 1 s.	Cycle Reset x (x: times to repeat; 6.5us, 9.8us, 50us, 100us, 500us, 1ms, 5ms, 10ms, 50ms, 100ms, 500ms, or 1s)*
Internal sequential break mode	An internal sequential break can be specified by using Break Condition U1 to U4 or Reset. For details, refer to section 5.5.6, Internal Sequential Break.	Break Condition U Sequential X (X: 2 -> 1, 3 -> 2 -> 1, or 4 -> 3 -> 2 -> 1)
Timeout break mode	A break occurs when the Performance Analysis 1 [Time Out] specification or [Count] (maximum number of passes) specification is satisfied. For details, refer to section 5.5.10, Timeout Break.	Timeout break of Performance analysis
Timeout trace-stop mode	Acquisition of trace information is terminated when the Performance Analysis 1 [Time Out] specification or [Count] (maximum number of passes) specification is satisfied. For details, refer to section 5.6.1, Timeout Trace Stop.	Timeout trace of Performance analysis

Table 5.15 Emulation Modes (cont)

Emulation Mode	Description	[Emulation mode] Setting
Time interval measurement mode between satisfaction of specified conditions: For details, refer to section 5.7.2, Measuring Execution Time between Satisfaction of Specified Conditions	Time interval measurement mode 1: The execution time between satisfaction of Break Condition U2 and Break Condition U1 is measured. The user program is suspended as soon as Break Condition U1 is satisfied after Break Condition U2 has been satisfied. Time interval measurement mode 2: The total execution time between satisfaction of Break Condition U2 and Break Condition U1 is measured. Program execution is not suspended. Time interval measurement mode 3: The total execution times between satisfaction of Break Condition U2 and Break Condition U1, and satisfaction of Break Condition U4 and Break Condition U3 are measured. Program execution is not suspended.	Time interval measurement mode X (X: 1, 2,or 3)
Elapsed time of condition satisfaction mode	Time interval measurement mode 4: The execution time over which the condition specified as Condition B on the [Execution Mode1] page is satisfied.	Time interval measurement (Condition B)
Break-disabled mode	Software breaks, software sequential breaks, hardware breaks, hardware sequential breaks, and internal breaks are disabled during program execution	No break

Note: In cycle-reset mode, the RES signal is output to the MCU regardless of the MCU's operating state when the time specified by the command has elapsed. Figure 5.11 shows the timing with which the TRIG signal is output to the trigger-output probe in cycle-reset mode.

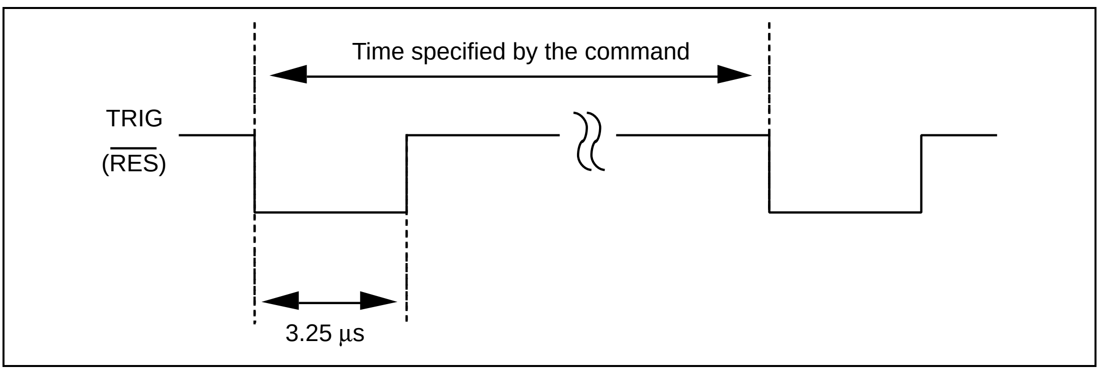


Figure 5.11 Timing for Trigger Signal Output

Restrictions on emulation modes are listed in table 5.16.

Table 5.16 Restrictions on Emulation Modes

Emulation Mode	Restrictions
Cycle-reset mode, and elapsed time of condition satisfaction mode	• Settings for software, software sequential, hardware, hardware sequential, internal, and internal sequential breaks are all ignored. • Trace-acquisition conditions are ignored. • Cannot enter trace-halt mode.
Internal sequential break mode	Settings for software and software sequential breaks are ignored.
Timeout break mode	Settings for software and software sequential breaks are ignored.
Timeout trace-stop mode	Settings for software and software sequential breaks are ignored.
Break-disabled mode	Settings for software, software sequential, hardware, hardware sequential, internal, internal sequential breaks are all ignored.

See table 5.26 for details on break conditions, and see table 5.60 for details on trace-acquisition conditions.

5.3.2 Trace-Halt Mode

Function: While in trace-halt mode, tracing is halted, and this means that trace information cannot be acquired by the trace buffer. During periods in trace-halt mode, emulation continues and is not suspended.

Entering trace-halt mode: Trace-halt mode can be entered in any of the following ways.

- Select [Halt] from the pop-up menu of the [Trace] window
- When emulation is in timeout trace-stop mode, trace-halt mode is automatically entered when the condition set in Performance Analysis 1 is satisfied (i.e., when the specified timeout period or number of passes has been exceeded).
- Trace-halt mode is automatically entered when a condition that has been specified as a trace-stop condition (Trace Conditions A, B, C, either as individual channels or as sequential breaks) is satisfied. 'TRACE STOP' will be displayed in the status bar.
- Trace-halt mode is automatically entered when a trace condition due to a trace buffer overflow is satisfied.

Returning from trace-halt mode: Any of the following actions will return the system to normal emulation.

- Enter the END command in the [Command Line] window.
- Select [Restart] from the [Trace] window's pop-up menu.

Stopping the execution of the user program: any of the followings actions will stop execution of the user program.

- Enter the HALT command in the [Command Line] window.
- Click the [Stop] button.
- Select [Halt] from the [RUN] menu.

5.3.3 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] page in the [System Status] window, and on the HDI window's status bar.

Table 5.17 is a list of the messages that indicate the various causes for termination.

Table 5.17 Causes for Termination

Display	Meaning
BREAK CONDITION A1,2,3,4,5,6,7,8	Break Condition A has been satisfied.
BREAK CONDITION B1,2,3,4,5,6,7,8	Break Condition B has been satisfied.
BREAK CONDITION C1,2,3,4,5,6,7,8	Break Condition C has been satisfied.
BREAK CONDITION U1,2,3,4	Break Condition U has been satisfied.
BREAK CONDITION SEQUENCE U	Sequential Break Condition U has been satisfied.
BREAK CONDITION SEQUENCE A	Sequential Break Condition A has been satisfied.
BREAK CONDITION SEQUENCE B	Sequential Break Condition B has been satisfied.
BREAK KEY	A forced break has been issued via the [Halt] button in the [Run] command, or the [STOP] button
BREAKPOINT	The break was triggered by a software breakpoint.
BREAK SEQUENCE	The break was triggered by a software sequential breakpoint.
ILLEGAL INSTRUCTION	A break instruction (H'0000) has been executed.
MULTI BREAK	Break triggered by the multibreak feature.
RESET BY E8000S	An error has occurred in the user system. The emulator has input a RES signal to the user system and forced the termination of execution.
STOP ADDRESS	The program has terminated at the cursor position after execution of the [Go to Cursor] menu item.
SUBROUTINE TIMEOUT	The timeout condition specified in Performance Analysis 1 has been satisfied.
SUBROUTINE COUNT OVERFLOW	The maximum number of passes condition specified in Performance Analysis 1 has been satisfied.
TRACE BUFFER OVERFLOW	The break is due to a trace-buffer overflow.
WRITE PROTECT	Execution of the program was terminated because of an attempt to write to a write-protected area.
GUARDED PROTECT	Access to an access-prohibited area was attempted.

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 5.18 is a list of the operating status messages.

Table 5.18 Operating Status Display

Display	Meaning
AB=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 AB=xxxxxxxx starts to be displayed.
Sleep	The MCU is in its sleep mode
Standby	The MCU is in its standby mode
TOUT A=xxxxxxxx	The value displayed is the value on the address bus. The bus termination period has exceeded the time specified as [Bus timeout] in the [Configuration] dialog box in the [Execution Mode1] page.
VCC DOWN	The voltage on Vcc is below the Vcc Down level specified in the [CPU Operating Mode] dialog box. The MCU is not operating correctly (this message is only displayed when the user clock has been selected).
WAIT A=xxxxxxxx	The WAIT signal is low. The value on the address bus is displayed. This is not displayed during refresh cycles.
BREQ	The BREQ signal is low.

5.4 Step Functions

5.4.1 Step Execution

Several types of step execution are available, and are shown in table 5.19.

Table 5.19 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 until a function is called. 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.	Click the [Step Over] button.
		Select [Step Over] from the [Run] menu.
Executing a specified number of steps	Executes the specified number of steps. 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: Some break conditions become disabled depending on the mode of execution in steps. For details on the relationship between the types of steps and break conditions that become invalid, see appendix E.7, Step Function.

5.4.2 Interrupts during Step Execution

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

5.5 Break Functions

The emulator provides break function shown in table 5.20. The HDI displays a list of breakpoints in the [Breakpoints] window, and the break conditions are specified in the dialog boxes for break functions.

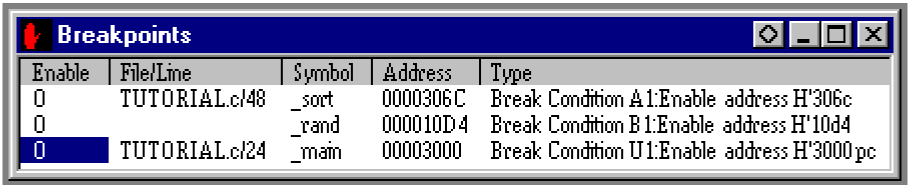


Figure 5.12 [Breakpoints] Window

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

Table 5.20 Break Functions

Type	Description
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. "Break Point" is displayed under Type in the [Breakpoints] window.
Software sequential break	An order for the satisfaction of conditions can be specified for software breakpoints. When all of the specified conditions have been satisfied in the specified order, a break occurs. Up to seven pass points (in order of satisfaction) and one reset point can be specified. "Break Sequence" is displayed under Type in the [Breakpoints] window.
Hardware break	This type of break is generated by the dedicated hardware in the emulator. Conditions can be specified as Break Condition A, B, and C, and when one of these conditions has been satisfied, a break occurs. "Break Condition Xn" (X: A, B, or C; n: number) is displayed under Type in the [Breakpoints] window.
Hardware sequential break	An order of satisfaction can be specified for hardware break conditions (Break Condition A or B). When all of the specified conditions have been satisfied in the specified order, a break occurs. "Break Condition X Sequential" (X: A or B) is displayed under Type in the [Breakpoints] window.
Internal break	This break function is implemented by the MCU. When any of the conditions specified as Break Condition U1 to U4 has been satisfied, a break occurs. "Break Condition Un" (n: number) is displayed under Type in the [Breakpoints] window. When a reset point is specified, “Break Condition U Reset” is displayed under Type in the [Breakpoints] window.
Internal sequential break	An order of satisfaction can be specified for internal break conditions. This kind of break occurs when all of the specified conditions have been satisfied in the specified order. Break Condition U1 to U4 can be used in four levels of sequential break conditions. Break Condition U Reset is used as the reset point for sequential break conditions.
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.
Forced break due to writing to a write-protected/access-forbidden area	This kind of break occurs when the current user program attempts to write to an area of the emulation memory that has the write-protected or access-forbidden attribute.
Break due to trace buffer overflow	This break occurs when the trace buffer in the emulator overflows during trace acquisition.
Timeout break	A timeout break occurs when the execution time exceeds the timeout condition specified for Performance Analysis 1.

Notes: 1. In the [Enable] column of the [Breakpoints] window, O is displayed when the corresponding Break Condition A, B, C or U is enabled. Nothing is displayed when the Break Condition is disabled.
When a breakpoint is specified at an address that has also been specified as the address-bus condition for Break Condition A, B, C or U, ● is displayed instead of O.

2. During execution of the user program, [Go to Source] item of the pop-up menu in the [Breakpoints] window cannot be used to move from the display of a breakpoint to the corresponding line of source code (or address) in the [Source] (or [Disassemble]) window.

5.5.1 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. A number of passes can be specified as a break condition, and a break will then occur when the breakpoint has been passed the specified number of times.
It is possible to specify up to 255 software breakpoints. Up to 65,535 passes can be specified for each software breakpoint.

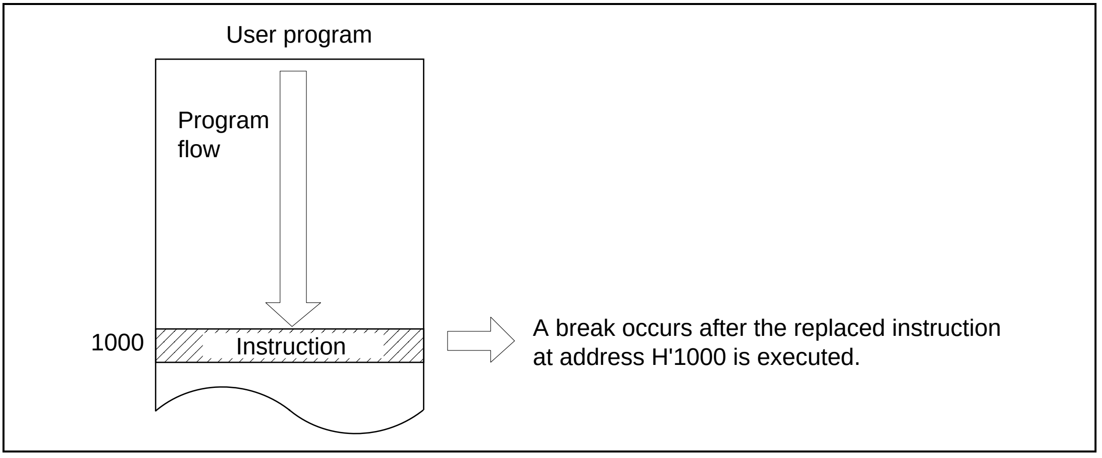


Figure 5.13 Example of a Software Break Instruction

When memory is accessed in trace-halt mode, the contents at the specified address are replaced with a break instruction.

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 [Break] dialog box will appear.

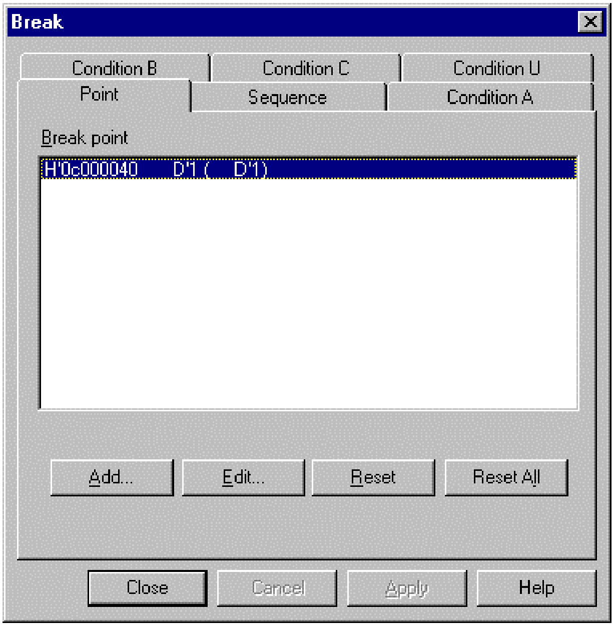


Figure 5.14 [Point] Page ([Break] Dialog Box)

Table 5.21 [Point] Page Options

Option	Description
[Break point]	Displays the specified pass point addresses. The pass point and reset point address settings are displayed as follows. <pass point address> <number of passes> <number of passes> indicates the number of times the breakpoint was passed before execution was completed. This is cleared on the next execution.
[Add…]	Sets software breakpoints. Clicking [Add…] opens the [Break Point] dialog box.
[Edit…]	Allows the user to modify the software sequential breakpoint settings selected in the [Break point] list box. Clicking [Edit...] opens the [Break Point Sequence] dialog box.
[Reset]	Clears the software sequential breakpoint settings selected in the [Break point] list box.
[Reset All]	Clears all software sequential breakpoints.

Click the [Add...] button to open the [Break Point] dialog box. Specify the breakpoint's address and the number of passes, then click the [OK] button.

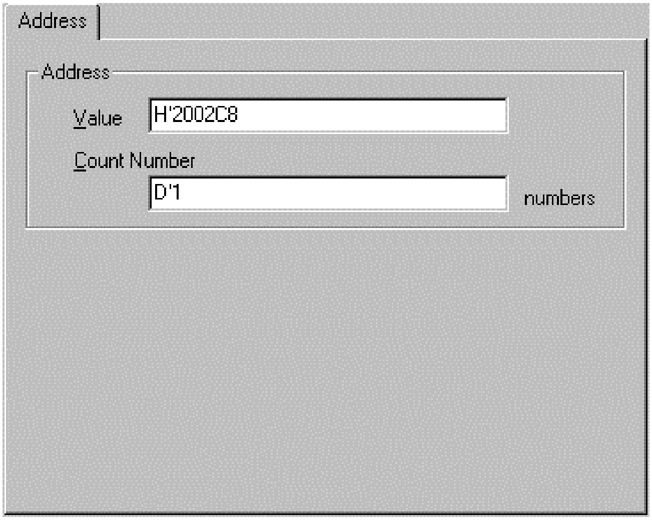


Figure 5.15 [Break Point] Dialog Box ([Address] Page)

The display returns to the [Point] page. The [Break point] list box now displays the specified address and the number of passes. Click the [OK] button to close the [Break] dialog box.

Table 5.22 [Break Point] Dialog Box Options

Option	Description
[Value]	Sets the breakpoint's address as a numeric or symbolic value.
[Count Number]	Sets a number of passes. A break occurs when the breakpoint has been passed the specified number of times. The default setting is H'1. Any value from H'1 to H'FFFF can be set here.

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 standard 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 used as part of a software sequential breakpoint
- The address is in any area other than CS area (except the internal ROM/RAM area)
- An instruction which may satisfy Break Condition U4
- The address of the delay slot for a delayed-branch instruction

Notes: 1. When the number of passes is specified, emulator firmware stops the program (for about 100 ms) every time the program passes the address set as a break condition so that it can update its count of passes. As a result, the program does not operate in realtime. When the program passes such an address, the emulator executes the instruction at the address as a single step then returns to normal program execution. Break Condition U4 becomes invalid during this single step execution.

2. 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 of the [Run...] menu is invalid. Ensure that the total number of software breakpoints and settings made by using the [Stop At] item of the [Run...] menu is 255 or less.
3. 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.
4. When the content of a software breakpoint address is modified during execution of the user program, the following message will be displayed after execution stops.

   BREAKPOINT IS DELETED A=xxxxxxxx

   When the above message is displayed, use the [Delete All] or [Disable] button in the [Breakpoints] window to cancel all software breakpoint settings.
5. 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.
6. Software breakpoints are ignored during step execution.
7. Do not set a software breakpoint immediately after the delay slot of a delayed branch 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.
8. Do not allow the user program to modify memory at a breakpoint address.
9. The contents of the specified address where a breakpoint has been set are replaced by a break instruction during emulation.
10. When execution starts at the address of a software breakpoint, counting of the number of passes starts from the next pass.

5.5.2 Software Sequential Break

Overview: A software sequential break occurs when software breakpoints are encountered in the specified order.

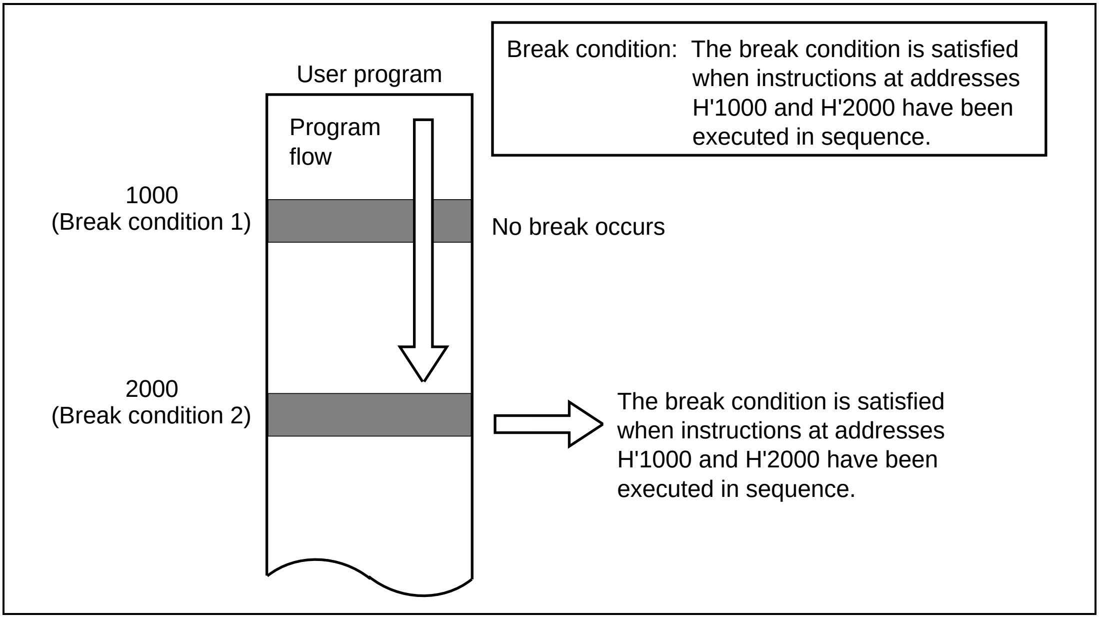


Figure 5.16 Example of a Software Sequential Break

A reset point can be specified along with the pass points. When execution passes the reset point, or if the pass points are not passed in the specified order, the execution record for the pass point up to that point is cleared. The emulator then restarts checking for satisfaction of the sequential break conditions from the first pass point. Up to seven pass points and one reset point can be specified.

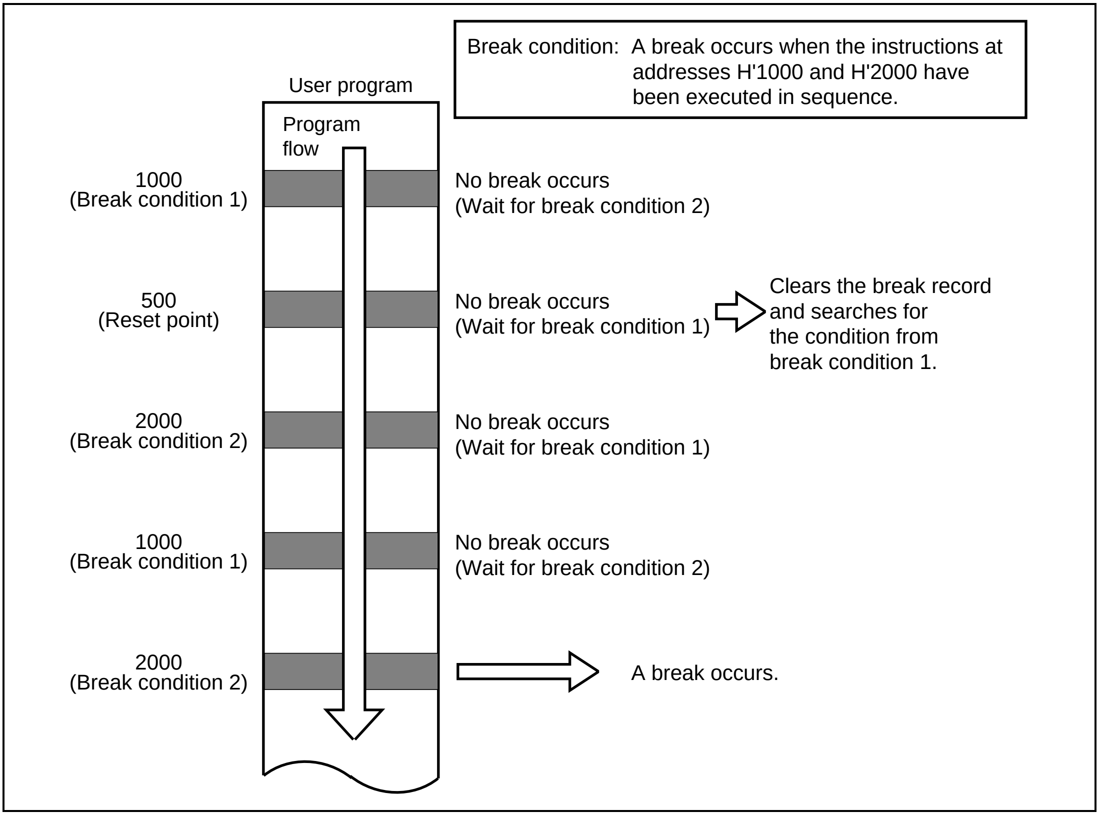


Figure 5.17 Example of a Software Sequential Break (Reset Point Specification)

Setting Software Sequential Break: Place the cursor in the [Breakpoints] window then click the right-hand mouse button to display the pop-up menu. Select [Add...] from the menu, and the [Break] dialog box will appear. Select the [Sequence] page.

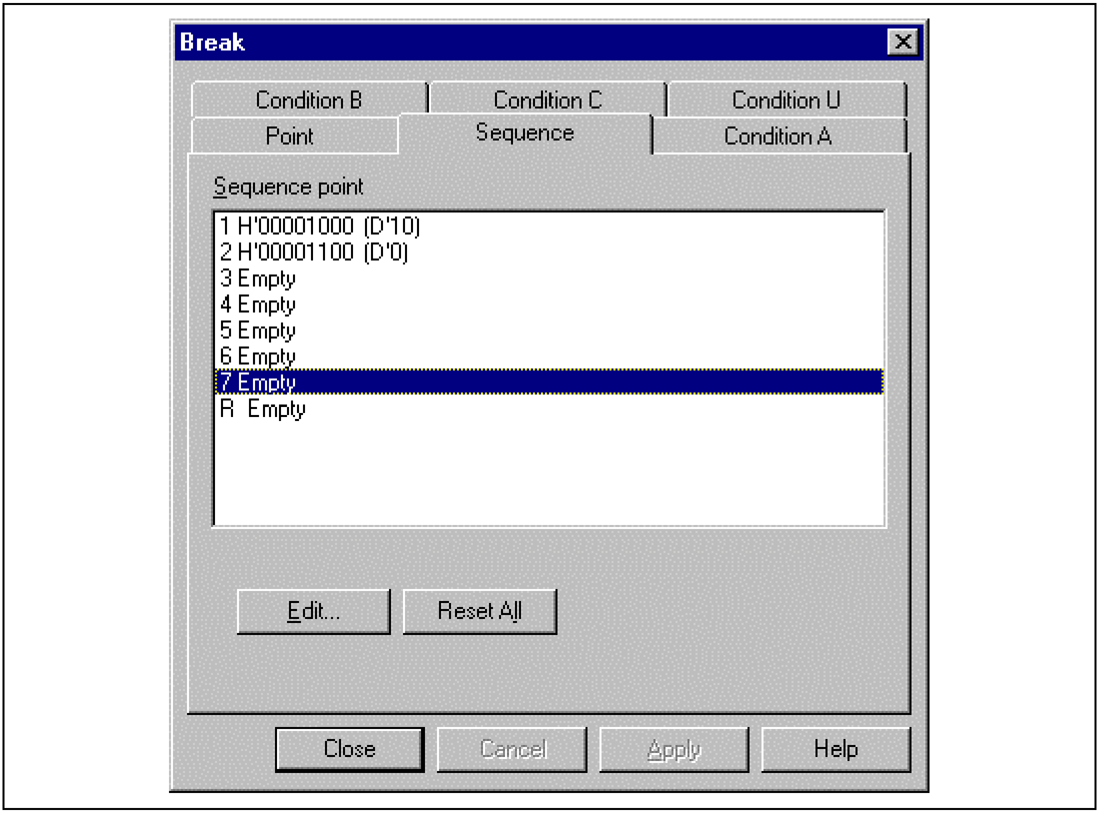


Figure 5.18 [Break] Dialog Box ([Sequence] Page)

Table 5.23 [Sequence] Page Options

Option	Description
[Sequence point]	Displays the specified pass point addresses. Empty is displayed if there are no settings. 1 to 7: Settings for pass point addresses 1 to 7 R: Setting of the reset point address The pass point and reset point address settings are displayed as follows. <pass point address> <number of passes> The number of passes indicates the number of times the pass point or reset point was passed while the program was being executed. However, counting re-starts from 0 when the number of passes exceeds D'16383. Cleared on the next execution.
[Edit...]	Allows the user to modify the software sequential breakpoint settings selected in the [Sequence point] list box. Clicking [Edit...] opens the [Break Point Sequence] dialog box.
[Reset All]	Clears all software sequential breakpoint settings in the [Sequence point] list box.

Click the [Edit…] button to open the [Break Point Sequence] dialog box. Specify pass point addresses in the order of [Address 1] to [Address 7], a reset point address in [Reset Point] (when the specified pass points are executed in order from [Address 1] to [Address 7], the break condition will be satisfied). Click the [OK] button.

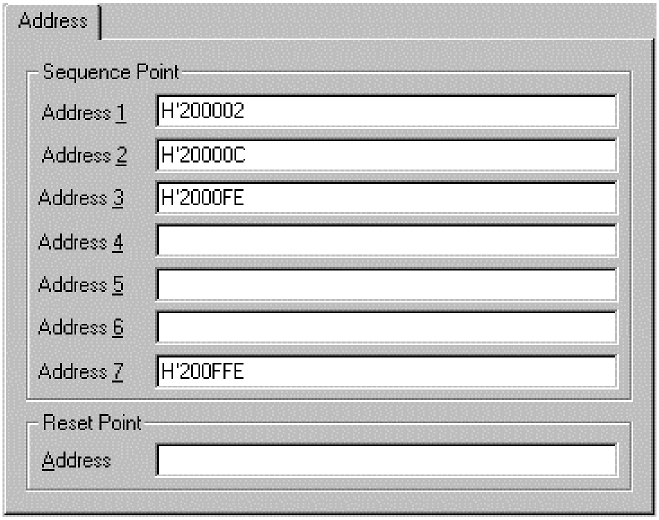


Figure 5.19 [Break Sequence] Dialog Box

The [Break Sequence] list box will now display the specified addresses. Click the [OK] button to close the [Break Sequence] dialog box.

The display returns to the [Sequence] page. The [Sequence Point] list box will display the specified software sequential break conditions. Click the [OK] button to close the [Break] dialog box.

Table 5.24 [Break Sequence] Page Options

Option		Description
[Sequence Point] group box	[Address 1] to [Address 7]	Sets a breakpoint address in the sequence of points to be passed as a numerical or symbolic value. One to seven pass points can be set. At least two points must be set.
[Reset Point] group box	[Address]	Sets the reset point as a numerical or symbolic value. A reset point need not be set.

When a software sequential break is placed, the instruction at the specified address is replaced. It is only possible to set a software breakpoint in the RAM area (including the standard emulation memory) and the internal ROM area. 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 used as part of a software sequential breakpoint
- The address is in any area other than CS area (except the internal ROM/RAM area)
- The address holds an instruction which may satisfy Break Condition U4
- The address of the delay slot for a delayed-branch instruction

Notes: 1. When the number of passes or reset point is specified, emulator firmware stops the program every time the program passes the address set as a break condition so that it can update its count of passes. As a result, the program does not operate in realtime (it will stop about 100 ms). When the program passes such an address, the emulator executes the instruction at the address as a single step then returns to normal program execution. Break Condition U4 becomes invalid during this single-step execution.

2. When a software breakpoint is set in the delay slot of a delayed branch instruction, the value in the PC will become illegal. Do not set a software breakpoint at the slot instruction after a delayed branch instruction.

3. Do not allow the user program to modify values in memory at software sequential breakpoints.

4. When execution starts from the first pass point, counting of numbers of passes starts the next time a breakpoint is passed.

5.5.3 Hardware Break

Overview: Hardware break functions are implemented by dedicated hardware in the E8000S station. The hardware break conditions shown in table 5.25 can be specified for Break Condition A, B, or C. Hardware break occurs when all of the specified conditions (an AND condition) are satisfied.

Table 5.25 Hardware Break Conditions

Break 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.
Access type	The condition is satisfied when the access type matches the specification.
Read/Write	The condition is satisfied by reading or writing as specified.
External probe	The condition is satisfied when the external probe (PRB) signal levels match a specification .
External interrupt	The condition is 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.
Delay	The break occurs the specified number of bus cycles after the above conditions have been satisfied.

Figure 5.20 shows an example of the operation of a hardware break when an address-bus condition and satisfaction-count condition have been specified.

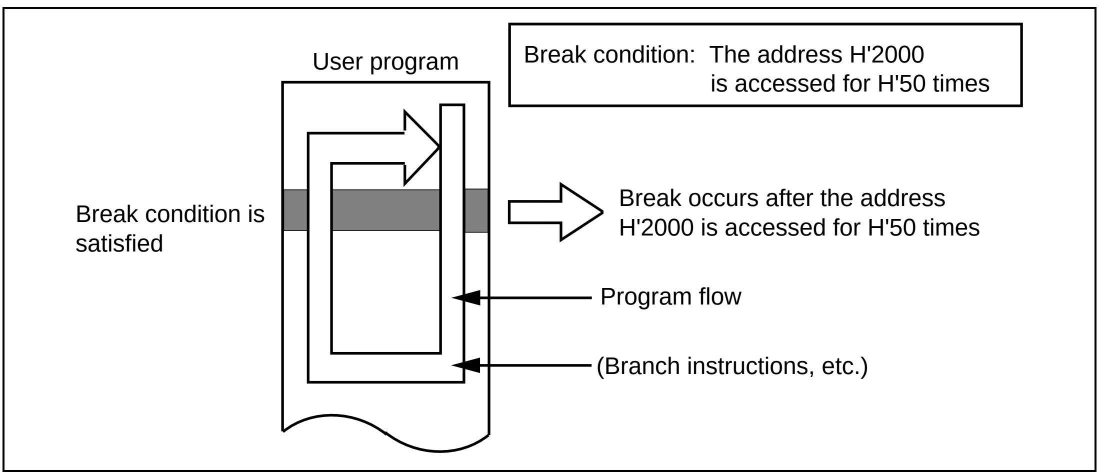


Figure 5.20 Example of a Hardware Break with a Satisfaction-Count Condition Specified

Figure 5.21 shows an example of the operation of a hardware break when an address-bus condition and delay condition have been specified.

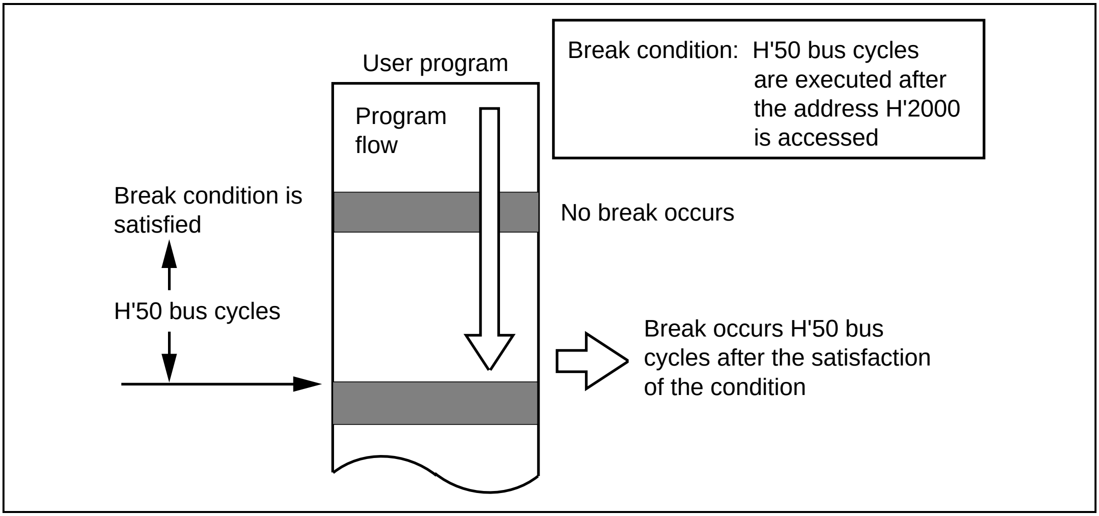


Figure 5.21 Example of a Hardware Break with Delay Condition Specified

Break Conditions A, B, and C: Eight channels for each condition (24 channels in total can be specified)

The conditions that can be specified as Break Conditions A, B, and C are shown in table 5.26.

Table 5.26 Specifiable Hardware Break Conditions

Break Condition	Break Condition A (1 to 8)	Break Condition B (1 to 8)	Break Condition C (1 to 8)
Address bus	O	O	O
Data bus	O	O	X
Access type	O	O	O
Read/Write	O	O	X
External probe	O	O	X
External interrupt	O	O	X
Satisfaction count	O	O	X
Delay	O	O	X

Note: O: Can be specified.
X: Cannot be specified.
The delay condition is only available for Break Conditions A7 and B7.

Setting a Hardware Break: The setting of Break Condition A7 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 [Add...] from the menu, and the [Break] dialog box will appear. Select the [Condition A] page.

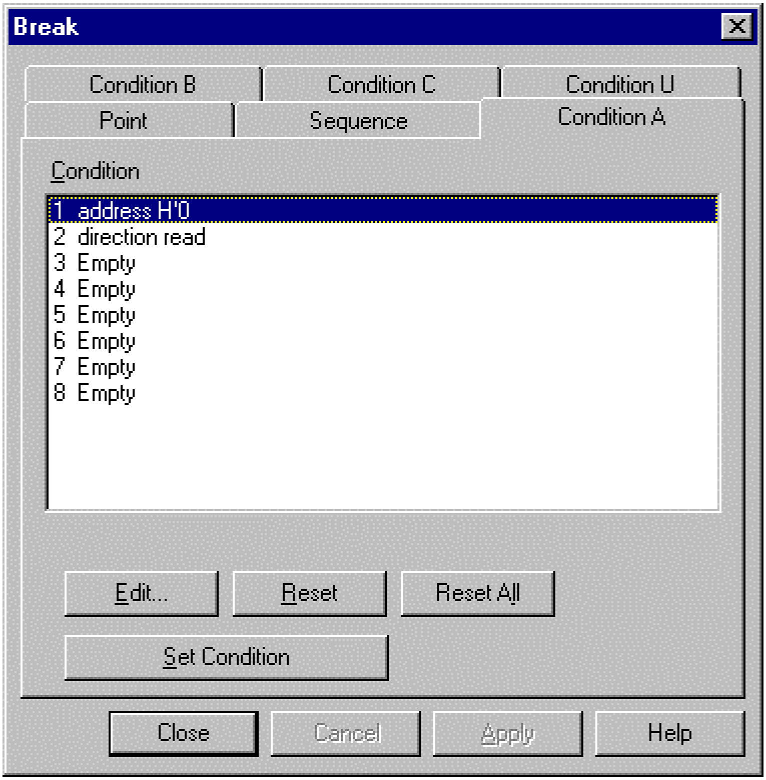


Figure 5.22 [Break] Dialog Box ([Condition A] Page)

Table 5.27 [Condition A, B, C] Page Options

Option	Description
[Condition]	Displays the current settings for Break Condition A, B, or C. 'Empty' is displayed if there are no settings. 1 to 8: Settings for Break Condition x1 to x8 (x: A, B, or C.)
[Edit...]	Modifies the Break Condition A, B, or C settings selected in the [Condition] list box. Clicking this button opens the [Break Condition Xn] dialog box. (X: A, B, or C; n: channel number.)
[Set condition]	Modifies the Break Condition A or B settings selected in the [Condition] list box. Clicking this button opens the [Condition] dialog box. (This option is not displayed on the [Condition C] page.)
[Reset]	Clears the Break Condition A, B, or C settings selected in the [Condition] list box.
[Reset All]	Clears all Break Condition A, B, or C settings in the [Condition] list box.

Click condition 7 to select it from the [Condition] list box. Click the [Edit...] button to open the [Break Condition A7] dialog box.

Hardware break conditions are specified in the [Break Condition A1 to A8, B1 to B8, C1 to C8] dialog box, which has the tabbed pages listed in table 5.28.

Table 5.28 [Break Condition A1 to A8, B1 to B8, C1 to C8] Dialog Box Pages

Dialog Box	Page	Description
[Break Condition A1 to A6 and A8], [Break Condition B1 to B6 and B8]	[Address]	Sets address bus conditions.
	[Data]	Sets data bus conditions.
	[Bus State]	Sets access type and read/write cycle conditions.
	[Probe]	Sets external probe signal conditions.
	[Interrupt]	Sets external interrupt signal conditions (NMI and IRQ0 to IRQ7).
	[Count]	Sets satisfaction count conditions.
[Break Condition A7], [Break Condition B7]	[Address]	Sets address bus conditions.
	[Data]	Sets data bus conditions.
	[Bus State]	Sets access type and read/write cycle conditions.
	[Probe]	Sets external probe signal conditions.
	[Interrupt]	Sets external interrupt signal conditions (NMI and IRQ0 to IRQ7).
	[Count]	Sets satisfaction count conditions.
	[Delay]	Sets delay conditions.
[Break Condition C1 to C8]	[Address]	Sets address bus conditions.
	[Bus State]	Sets access type conditions.

Specify the required conditions on the corresponding pages, then click the [OK] button. The [Break Condition A1] dialog box closes and the display returns to the [Condition A] page. The specified hardware break conditions will now be displayed as condition 7 in the [Condition] list box. Click the [OK] button to close the [Break] dialog box.

Hardware break conditions for the other channels are specified in the same way.

The following sections describe the pages of the [Break Condition A1 to A8, B1 to B8, C1 to C8] dialog box.

(a) [Address] Page

Use this page to specify an address bus condition.

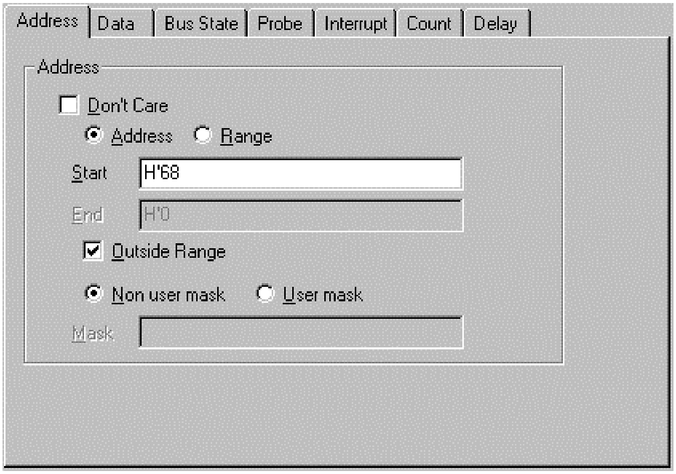


Figure 5.23 [Break Condition A7] Dialog Box ([Address] Page)

Table 5.29 [Address] Page Options

Option	Description
[Don't Care]	Selects no address condition.
[Address]	The condition is satisfied when an address in the range above [Start] or that is selected by [Mask] is accessed.
[Range]	The condition is satisfied when an address in the range set as [Start]-[End] is accessed.
[Start]	Sets the start of the range of address-bus values as a numerical or symbolic value.
[End]	When [Range] is selected, sets the end of the range of address-bus values as a numerical or symbolic value.
[Outside Range]	The condition is satisfied at any address that is not in the range above [Start] or an address selected by [Mask] or in the range set as [Start]-[End].
[Non user mask]	Sets no mask condition.
[User mask]	Sets mask conditions.
[Mask]	Sets required values for bits and selects bits to be masked after selecting [Address] and [User mask]. The [Mask] setting is disabled if [Range] is selected as the type of address condition. The [Range] condition will be satisfied or not satisfied regardless of any values set as part of the mask.

(b) [Data] Page

Take the following into account when specifying a data bus condition; the address, the data-bus width, and size of the data being accessed. For details, see appendix E.4, Hardware Break Function.

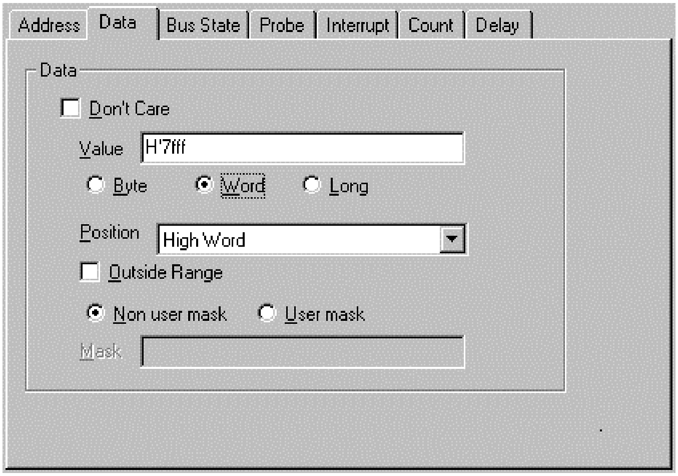


Figure 5.24 [Break Condition A7] Dialog Box ([Data] Page)

Table 5.30 [Data] Page Options

Option	Description
[Don't Care]	Selects no data bus condition.
[Value]	Sets a data bus value as a number. When a bit is masked, the bit always satisfies the condition regardless of its value.
[Byte]	Sets byte-data-access cycles.
[Word]	Sets word-data-access cycles.
[Long]	Sets longword-data-access cycles.
[Position]	Specifies the location of effective data on the bus. Select one of the following settings. High Word: High-order words (can be specified with [Word]) Low Word: Low-order words (can be specified with [Word]) Byte 3: Upper bytes of upper words (can be specified with [Byte]) Byte 2: Lower bytes of upper words (can be specified with [Byte]) Byte 1: Upper bytes of lower words (can be specified with [Byte]) Byte 0: Upper bytes of lower words (can be specified with [Byte])
[Outside Range]	Sets the condition as any value other than those specified.
[Non user mask]	Sets no mask condition.
[User mask]	Sets the mask conditions.
[Mask]	Sets the mask bits if [User mask] is selected. When a bit is masked, the bit always satisfies the condition regardless of its value.

(c) [Bus State] Page

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

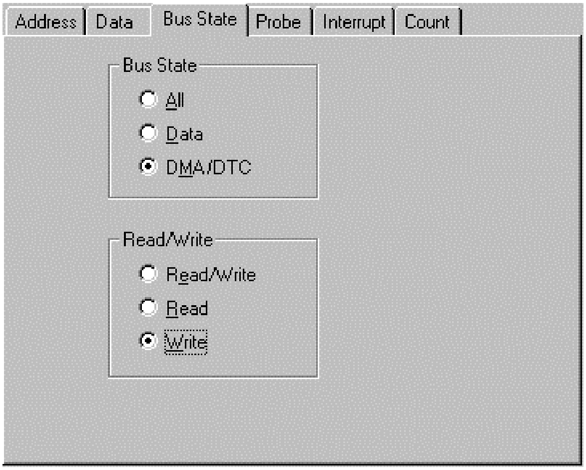


Figure 5.25 [Break Condition A7] Dialog Box ([Bus State] Page)

Table 5.31 [Bus State] Page Options

[Bus State] Group Box

Option	Description
[All]	Sets access of any type as the break condition.
[Data]	Sets memory access cycles as the break condition.
[DMA/DTC]	Sets DMA/DTC cycles as the break condition.

[Read/Write] Group Box

Option	Description
[Read/Write]	Sets either read or write cycles as the break condition.
[Read]	Sets read cycles as the break condition.
[Write]	Sets write cycles as the break condition.

(d) [Probe] Page

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

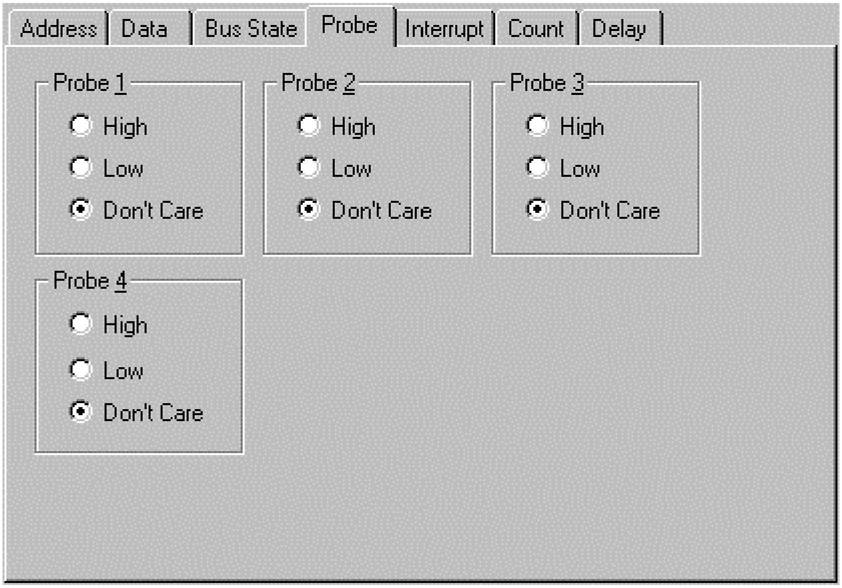


Figure 5.26 [Break Condition A7] Dialog Box ([Probe] Page)

Table 5.32 [Probe] Page Options

Option	Description
[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.
[Don't Care]	Selects no external probe signal state condition.

(e) [Interrupt] Page

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

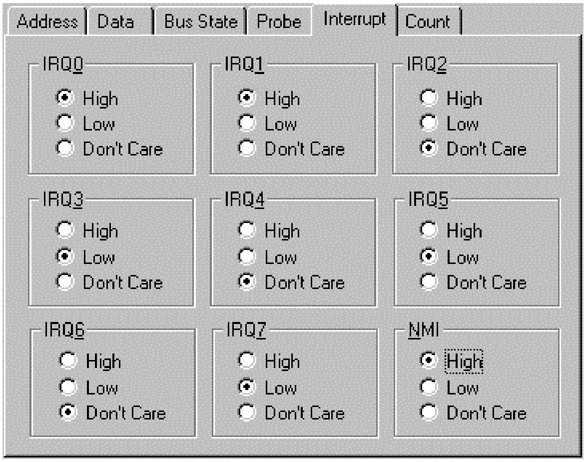


Figure 5.27 [Break Condition A7] Dialog Box ([Interrupt] Page)

Table 5.33 [Interrupt] Page Options

[IRQ0 to IRQ7] Group Box

Option	Description
[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.
[Don't Care]	Selects no external interrupt signal state condition.

[NMI] Group Box

Option	Description
[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.
[Don't Care]	Selects no NMI signal-state condition.

Note: When using signals IRQ0 to IRQ7, set the corresponding pins correctly in the [CPU Operating Mode] dialog box.

(f) [Count] Page

Use this page to specify a satisfaction count condition.

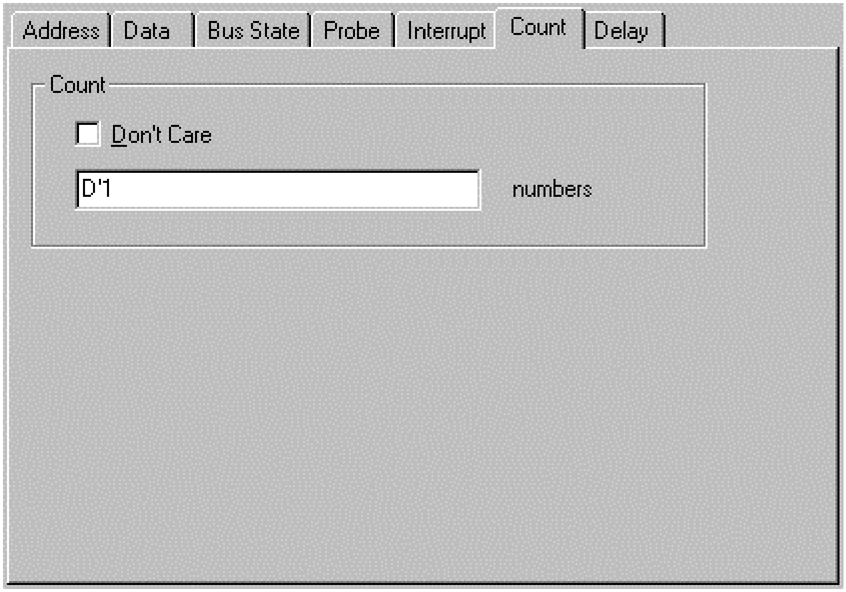


Figure 5.28 [Break Condition A7] Dialog Box ([Count] Page)

Table 5.34 [Count] Page Options

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

(g) [Delay] Page

Use this page to specify a delay condition.

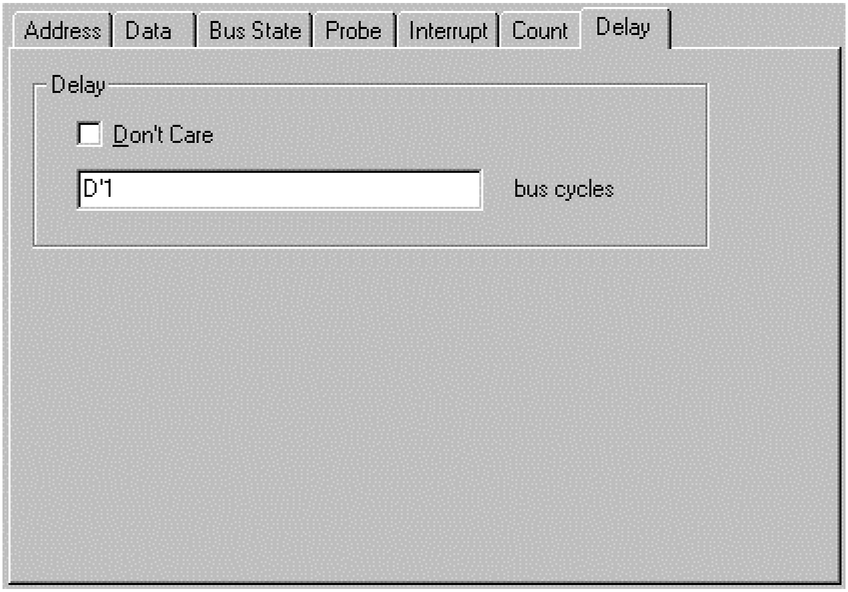


Figure 5.29 [Break Condition A7] Dialog Box ([Delay] Page)

Table 5.35 [Delay] Page Options

Option	Description
[Don't Care]	Selects no delay condition.
Input area	Sets a delay condition values as a number of bus cycles. The default is D'1. Any value in the range from D'1 to D'32767 can be set here.

Notes: 1. Break Condition A shares hardware with Trace Condition A. Therefore, when any channel of Trace Condition A has been specified, it is not possible to set or modify Break Condition A.

2. Break Condition B shares hardware with Trace Condition B. Therefore, when any channel of Trace Condition B has been specified, it is not possible to set or modify Break Condition B.

3. Break Condition C shares hardware with Trace Condition C and Performance Analysis. Therefore, when any channel of Trace Condition C or a Performance Analysis setting has been specified, it is not possible to set or modify Break Condition C.
4. When a hardware break condition has been satisfied, execution may continue for two or more instructions before it stops. Other hardware break conditions may thus be satisfied before execution stops. If this is the case, two or more causes of termination will then be displayed.

Displaying a History of Hardware Break Conditions and Creating a New Condition: A history of hardware break conditions that have been set as Break Condition A or B can be displayed in the [Condition] dialog box. New conditions can also be created by using the [Condition] dialog box.

For example, click to select a channel number from the [Condition] list box on the [Condition A] page of the [Break] dialog box. Click the [Set Condition] button to open the [Condition] dialog box.

The [Condition] dialog box has the pages shown in table 5.36.

Table 5.36 Pages of the [Condition] Dialog Box

Page	Description
[History]	Displays the history of conditions that have been set as Break Condition A or B. Conditions can be moved from the [History] page to the [Entry List] page.
[Entry List]	Creates, modifies, duplicates, and deletes new conditions for Break Condition A or B.

(a) [History] Page

Use this page to display the history of hardware break condition settings for Break Condition A or B.

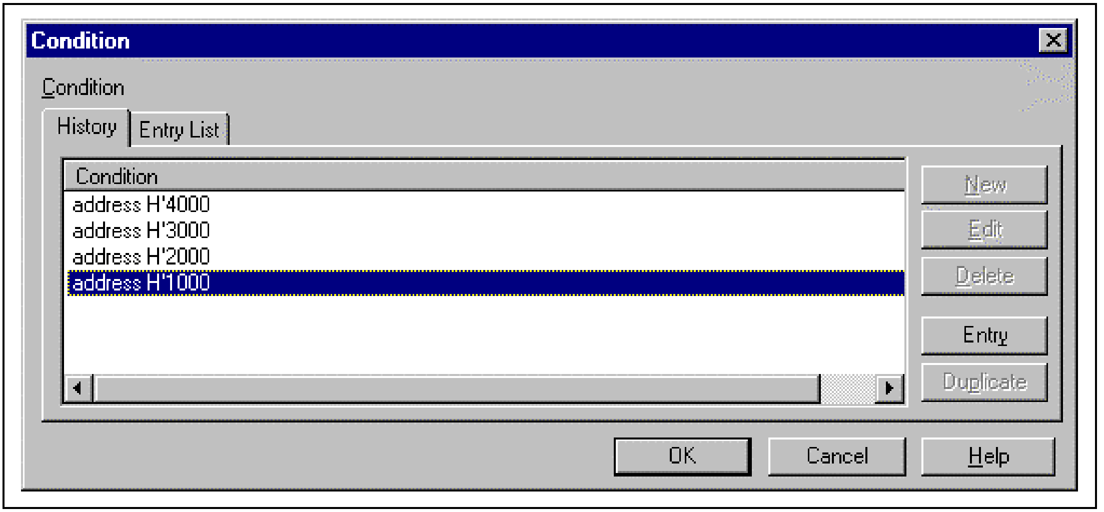


Figure 5.30 [Condition] Dialog Box ([History] Page)

Table 5.37 [History] Page Options

Option	Description
[Condition]	Up to 32 conditions are displayed in a list as a history of the conditions that have been set as Break Condition A or B.
[Entry]	Makes the condition selected in the [Condition] list available for use again as a channel of Break Condition A or B. Clicking the [Entry] button moves any condition selected in the [Condition] list box to the [Entry List] page, and deletes it from the [Condition] list on the [History] page.

Note: [New], [Edit], [Delete], and [Duplicate] are not available.

The hardware break conditions that have been specified in the [Break Condition A1 to A8, B1 to B8] dialog boxes are displayed in the [Condition] list on the [History] page.

Select a condition from the [Condition] list then click the [Entry] button to register that condition on the [Entry List] page. The condition is deleted from the [Condition] list.

(b) [Entry List] Page

Use this page to create, modify, duplicate, delete, and display conditions for Break Condition A or B.

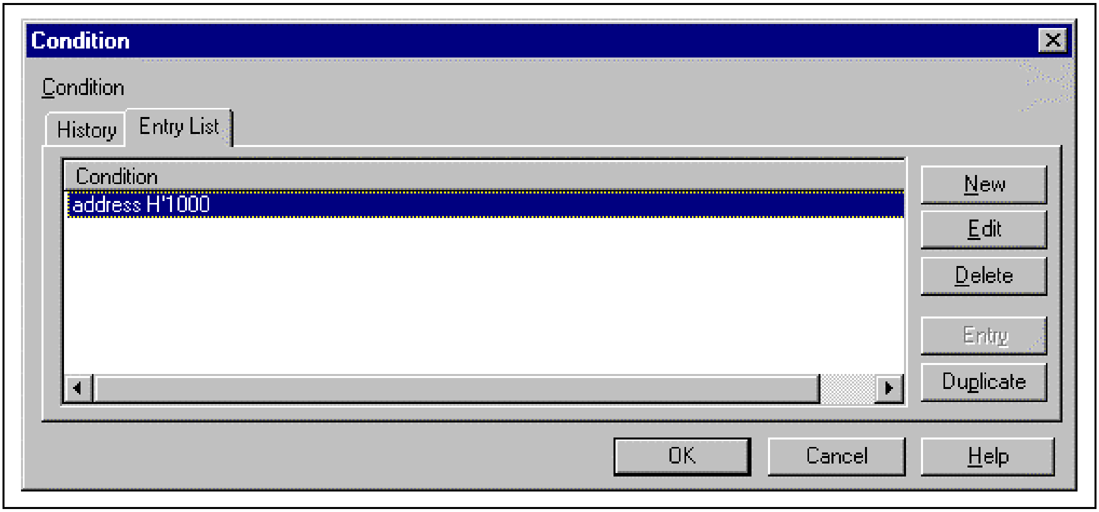


Figure 5.31 [Condition] Dialog Box ([Entry List] Page)

Table 5.38 [Entry List] Page Options

Option	Description
[Condition]	Up to 32 conditions that have previously been set can be redisplayed in a list.
[New]	Displays the [Condition A/B] dialog box so new conditions can be set.
[Edit]	Modifies a condition. Select a condition to be modified from the [Condition] list then click the [Edit] button to display the [Condition A/B] dialog box and modify the condition.
[Delete]	Deletes the condition selected in the [Condition] list.
[Duplicate]	Duplicates the condition selected in the [Condition] list.

Note: [Entry] is not available.

The [Condition] list displays a list of hardware break conditions that have been set. Click to select a condition then click the [OK] button to set that condition as a hardware break condition (here, a hardware break condition has been set as Break Condition A1).

To modify a hardware break condition after it has been created, click its entry in the [Entry List] then click the [Edit] button to open the [Condition A/B] dialog box. Modify the condition then click the [OK] button.

To set a new hardware break condition, click the [New] button to open the [Condition A/B] dialog box. Specify a condition then click the [OK] button.

Any entry in the history of hardware break condition settings displayed in the [Condition] list on the [History] page can be registered in the [Condition] list on the [Entry List] page. Click the entry to select the condition from the [Condition] list on the [History] page then click the [Entry] button. After the condition has been registered in the [Condition] list on the [Entry List] page, it is deleted from the [Condition] list on the [History] page.

To delete a hardware break condition that has been set, click the condition's entry to select it from the [Condition] list box then click the [Delete] button.

To duplicate a hardware break condition that has been set, click the condition's entry to select it from the [Condition] list box then click the [Duplicate] button.

5.5.4 Hardware Sequential Break

Overview: A hardware sequential break occurs after the selected of channels of a hardware break condition have been satisfied in a specified order.

This function uses Break Condition A or B; a sequence of up to seven break conditions along with one reset point can be specified as Break Condition A or B.

The user must consider the order of satisfaction in specifying break conditions for Break Conditions A and B; a break will only occur when the break conditions have all been satisfied in the order specified by the user.

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

Setting a Hardware Sequential Break: Select [Break] for [Condition A] under [Sequence] on the [Execution Mode2] page of the [Configuration] dialog box.

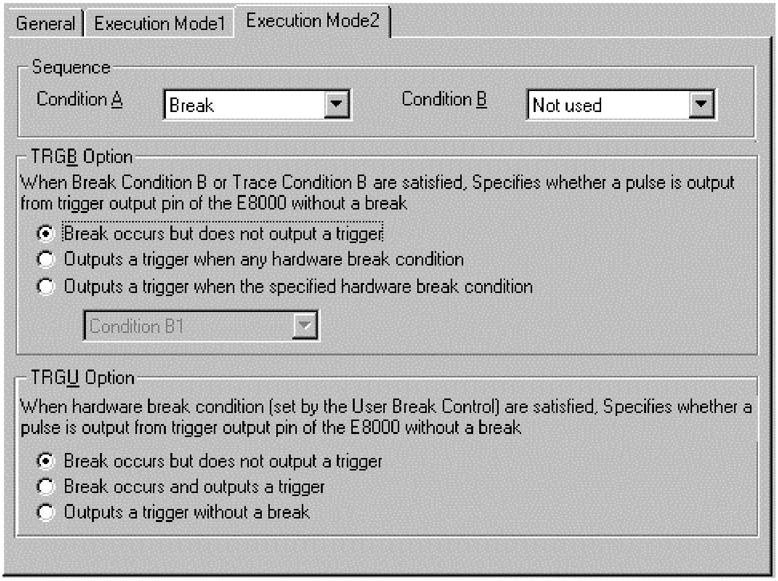


Figure 5.32 [Configuration] Dialog Box (Hardware Sequential Break)

This setting selects Break Condition A for use as a hardware sequential break. At this time, any existing condition settings for Break Condition A are cleared if there are conditions for Trace Condition A, those conditions are cleared.

Select the [Condition A] page of the [Breakpoints] window.

The same setting procedure applies to the [Condition B] page.

Figure 5.33 shows an example of the display when three break-condition points and one reset point have been specified as Break Condition A.

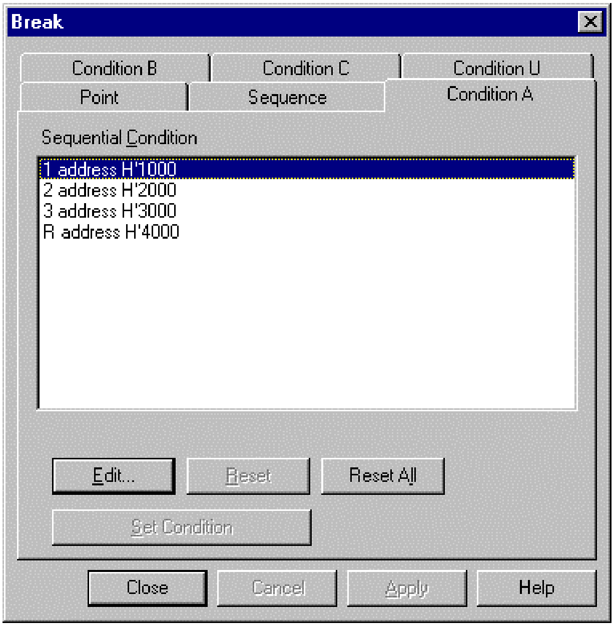


Figure 5.33 [Break] Dialog Box (after Setting a Hardware Sequential Break)

Table 5.39 [Condition A/B] Page Options (When a Hardware Sequential Break Has Been Specified)

Option	Description
[Sequential Condition]	Displays the sequence of break conditions for Break Condition A or B. Up to seven points are displayed in the order in which they must be satisfied. 1 to 7: Hardware break conditions One reset point is displayed. R: Reset point setting
[Edit...]	Modifies the Break Condition A or B setting that was selected in the [Sequential Condition] list box. Clicking this button opens the [Condition] dialog box.
[Reset All]	Clears all Break Condition A or B settings in the [Sequential Condition] list box.

Click the [Edit...] button to open the [Condition] dialog box.

The sequence of hardware break conditions is added by either of the two methods described below.

(a) Using the history of hardware break condition settings on the [History] page

Any hardware break condition displayed on the [History] page of the [Condition] dialog box can be used as a hardware sequential break condition.

Click to select a condition from the [Condition] list on the [History] page, then click the [▼] button; the condition is now added to the [Sequence] list box as No. 1. In the same way, hardware break conditions Nos. 2 to 7 can be added in sequence. The order in which the conditions must be satisfied to generate a break is the order in which they were added in the [Sequence] list box.

Click the [▼R] button to add a reset point. The reset point condition will be displayed to the right of Reset in the [Sequence] list box.

Click the [OK] button to return to the [Condition A] page, and the hardware sequential break conditions will have been set. The [Sequential Condition] list displays the hardware sequential break conditions that have been specified. Click the [OK] button to close the [Break] dialog box.

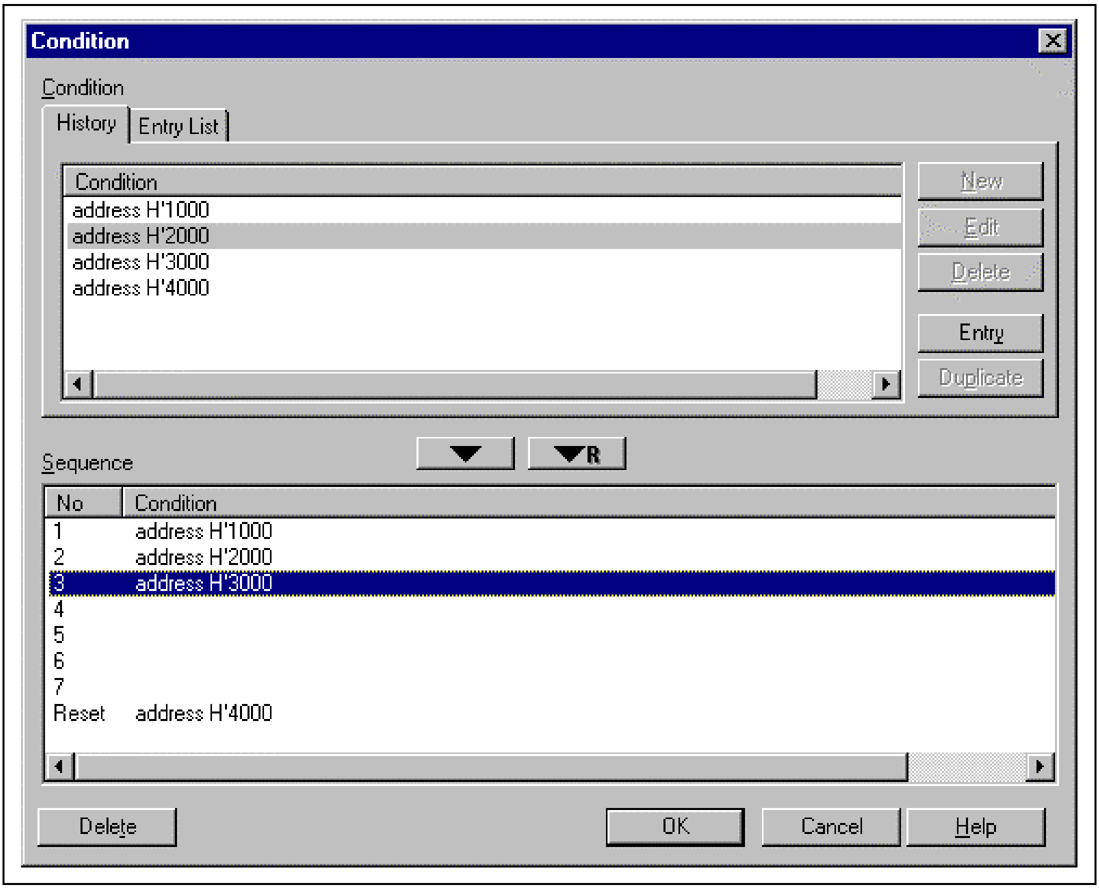


Figure 5.34 [Condition] Dialog Box ([History] Page)

Table 5.40 [History] Page Options

Option	Description
[Condition]	Up to 32 conditions can be displayed in a list as a history of the conditions that have been set as Break Condition A or B.
[Entry]	Makes the condition selected in the [Condition] list available for use again as a channel of Break Condition A or B. Clicking the [Entry] button moves any condition selected in the [Condition] list box to the [Entry List] page, and deletes it from the [Condition] list on the [History] page.
[▼]	Selecting a condition from the [Condition] list then clicking this button adds that condition to the [Sequence] list as one of Nos. 1 to 7.
[▼R]	Selecting a condition from the [Condition] list and clicking this button adds that condition as the reset condition in the [Sequence] list.
[No]	Numerical values indicate the sequential point number. Reset indicates the reset point.
[Sequence]	Displays the sequential conditions that have been specified as a list.
[Delete]	Deletes any condition selected in the [Sequence] list.

(b) Specifying hardware sequential break conditions on the [Entry List] page

A hardware break condition can be created on the [Entry List] page in the [Condition] dialog box, then added as a sequential break condition.

The [Condition] list displays the hardware break conditions that have been created. Click to select a condition, then click the [▼] button; the condition will be added to the [Sequence] list box as No. 1. In the same way, hardware break conditions 2 to 7 can be added in sequence. The order in which the conditions must be satisfied to generate a break is the order in which they were added in the [Sequence] list box.

Click the [▼R] button to add a reset point. The reset-point condition will be displayed to the right of Reset in the [Sequence] list box.

Click the [OK] button to return to the [Condition A] page, and the hardware sequential break conditions will have been set. The [Sequential Condition] list displays the hardware sequential break conditions that have been specified. Click the [OK] button to close the [Break] dialog box.

For details on the creation of hardware break conditions, refer to the description of the [Entry List] page in [Entry List] Page under Displaying a History of Hardware Break Conditions and Creating a New Condition in section 5.5.3, Hardware Break.

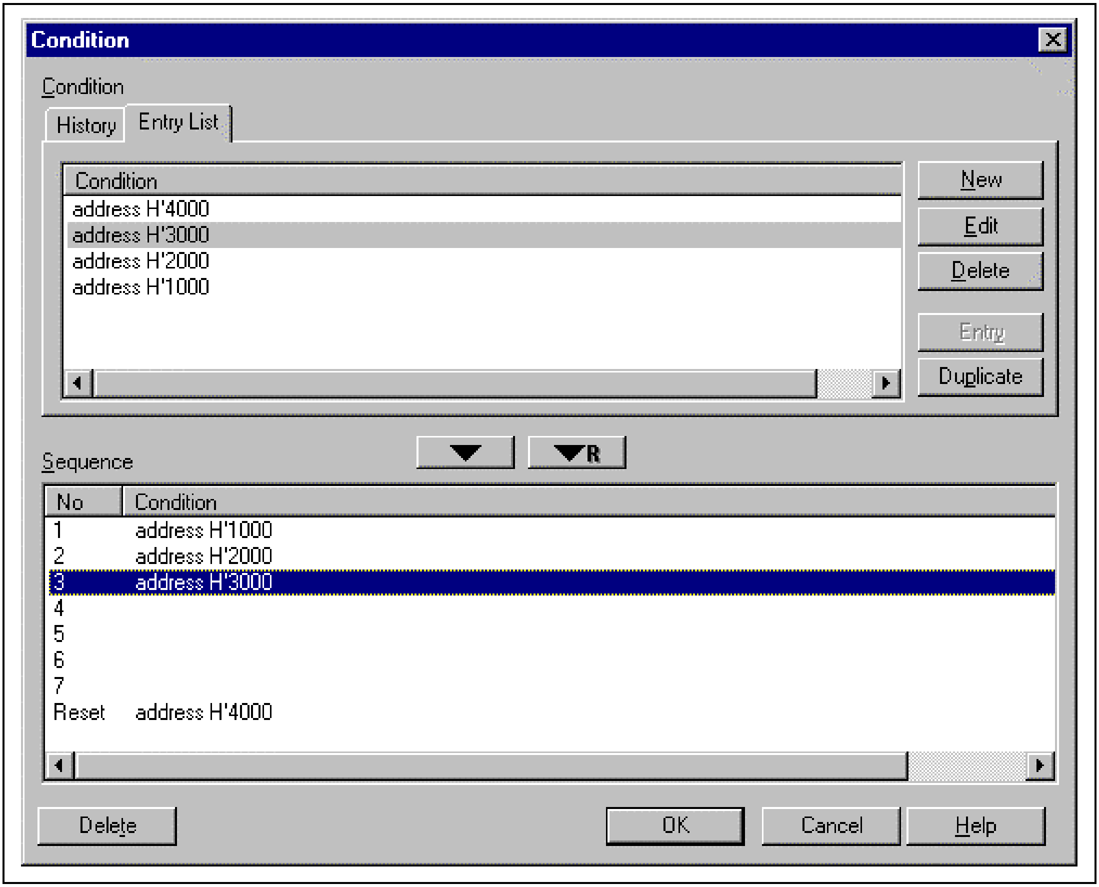


Figure 5.35 [Condition] Dialog Box ([Entry List] Page)

Table 5.41 [Entry List] Page Options

Option	Description
[Condition]	Up to 32 conditions that have previously been set are displayed in a list.
[New]	The [Condition A/B] dialog box is displayed so that new conditions can be created.
[Edit]	Selecting a condition to be modified from the [Condition] list then clicking the [Edit] button displays the [Condition A/B] dialog box so that the condition can be modified.
[Delete]	Deletes the condition selected in the [Condition] list.
[Duplicate]	Duplicates the condition selected in the [Condition] list.
[▼]	Selecting a condition from the [Condition] list then clicking this button adds that condition to the next empty spot in Nos. 1 to 7 of the [Sequence] list.
[▼R]	Selecting a condition from the [Condition] list then clicking this button places that condition next to Reset in the [Sequence] list.
[No]	Numerical values indicate the sequential point number; Reset indicates the reset point.
[Sequence]	Displays a list of sequential conditions that have been specified.
[Delete]	Deletes the condition selected in the [Sequence] list.

The status of hardware break condition satisfaction before the actual hardware break occurs will be displayed on the status bar during program execution, in the following format:

Condition A or B Seq <Number of passes/specified number>

When no condition has been specified, the status of hardware break conditions is displayed from Condition B to Condition A, rather than in the format shown above. Figure 5.36 shows an example of the display.

A hardware sequential break condition has been specified with Nos. 1 to 3 of Break Condition A, and No. 1 has been satisfied. A hardware sequential break condition has also been specified with Nos. 1 to 5 of Break Condition B and Nos. 1 to 3 have been satisfied.

Figure 5.36 Example: Display of the Status of a Hardware Sequential Break Condition

5.5.5 Internal Break (Break Condition U)

Overview: These break functions use the MCU's on-chip break function. The internal break conditions are shown in table 5.42. A given Break Condition U is satisfied when all of its specified conditions are satisfied (an AND condition).

Table 5.42 Internal Break Conditions

Break Condition	Description
Address*	Satisfied when the value on the address matches the specified value.
PC*	Satisfied when the value on the address bus for an instruction fetch matches the specified value. The break can be specified to occur either before or after execution of the fetched instruction.
Data	Satisfied when the value on the data bus matches the specified value.
Read/write	Satisfied when the read/write matches the specified condition. This condition is usually specified in combination with an address bus or data bus condition.
Access type (bus-state)	Satisfied when the bus cycle matches the specified condition. When [All] is specified here, all bus cycles, including instruction-fetch cycles, satisfy the condition. This condition is usually specified in combination with an address bus or data bus condition.
Satisfaction count	This condition is specified in combination with some other condition. The break occurs when the specified condition has been satisfied the specified number of times.
Reset point	Specifies the reset point condition for a sequential break.

Note: Either an address bus condition or a PC condition can be specified.

There are 4 channels, U1-U4. Note that Break Condition U Reset is a dedicated condition, used as the reset point for sequential break conditions.

Internal break conditions that can be specified by Break Condition U1 to U4 are shown in table 5.43.

Table 5.43 Specifiable Internal Break Conditions

Break Condition	Break Condition U1	Break Conditions U2 to U4	Break Condition U Reset*[2]
Address	O (either of the two conditions)	O (either of the two conditions)	O
PC			X
Data	O*[1]	O *[1]	X
Read/Write	O*[1]	O*[1]	X
Access type	O*[1]	O*[1]	X
Satisfaction count	O	X	X

Notes: O: Can be specified.
X: Cannot be specified.

1. A PC condition cannot be specified.
2. This condition is used as the reset point condition for internal sequential breaks.

Setting an Internal Break: The setting of Break Condition U1 is taken as an example.

Select [Add...] from the menu in the [Breakpoints] window, and the [Break] dialog box will appear. Select the [Condition U] page.

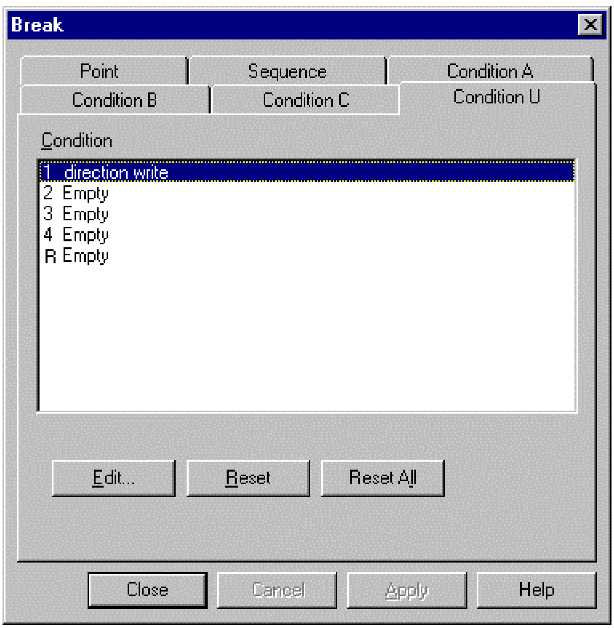


Figure 5.37 [Break] Dialog Box ([Condition U] Page)

Table 5.44 [Condition U] Page Options

Option	Description
[Condition]	Displays the current settings for Break Condition U. 'Empty' is displayed if there are no settings. 1 to 4: Settings for channels U1 to U4 R: Settings for Break Condition U Reset
[Edit...]	Modifies the Break Condition U setting selected in the [Condition] list box. Clicking this button opens the [Break Condition Un] dialog box. (n: channel number.)
[Reset]	Clears the Break Condition U settings selected in the [Condition] list box.
[Reset All]	Clears all Break Condition U settings in the [Condition] list box.

Click the list entry to select condition 1 in the [Condition] list box. Click the [Edit...] button to open the [Break Condition U1] dialog box.

The [Break Condition U1 – U4] dialog boxes have the pages shown in table 5.45.

Table 5.45 [Break Condition U1 – U4, Reset] Dialog Box Pages

Channel	Page	Description
Break Condition U1	[Address]	Sets address or PC conditions.
	[Data]	Sets data conditions.
	[Bus State]	Sets read/write cycle and access type conditions.
	[Count]	Sets the satisfaction count conditions.
Break Condition U2 to U4	[Address]	Sets address or PC conditions.
	[Data]	Sets data conditions.
	[Bus State]	Sets read/write cycle and access type conditions.
Break Condition U Reset	[Address]	Sets a reset point for the sequential break condition.

Specify the required conditions on the corresponding pages then click the [OK] button. The [Break condition U1] dialog box closes and the display returns to the [Condition U] page. The [Condition] list box will now display the specified internal break conditions as condition 1. Click the [OK] button to close the [Break] dialog box.

Internal 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.

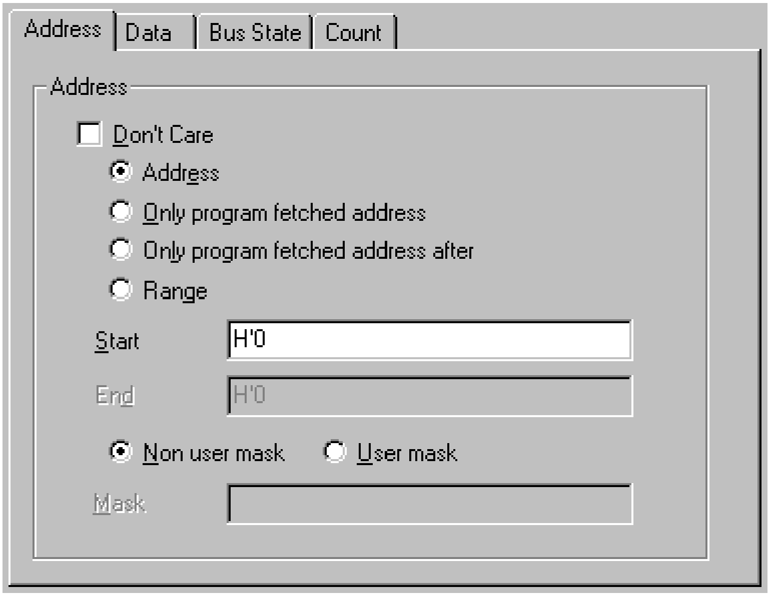


Figure 5.38 [Break Condition U1] Dialog Box ([Address] Page)

Table 5.46 [Address] Page Options

Option	Description
[Don't Care]	Selects no address condition.
[Address]	Select this button to set the address bus value specified in [Start] as the condition.
[Only program fetched address]	Select this button so that the condition is satisfied as soon as the value specified in [Start] is on the address but before the instruction is fetched from the address.
[Only program fetched address after]	Select this button so that the condition is only satisfied after an instruction fetch from the address specified in [Start].
[Range]	The condition to be satisfied is the address ranges specified in [Start] and [End] (only available with Break Condition U1 and Break Condition U3).
[Start], [End]	Sets an address-bus value as a numeric or symbolic value.
[Non user mask]	Sets no mask condition.
[User mask]	Sets mask conditions.
[Mask]	Sets the mask bits if [User mask] is selected. Masked bits on the data bus satisfy this break condition regardless of their values.

Note: When Break Condition U2 or Break Condition U4 is selected it is not possible to set [Range] for Break Condition U1 or Break Condition U3, respectively. In this case, clear the condition settings in Break Condition U2 or Break Condition U4.

(b) [Data] Page

Use this page to specify the data bus conditions.

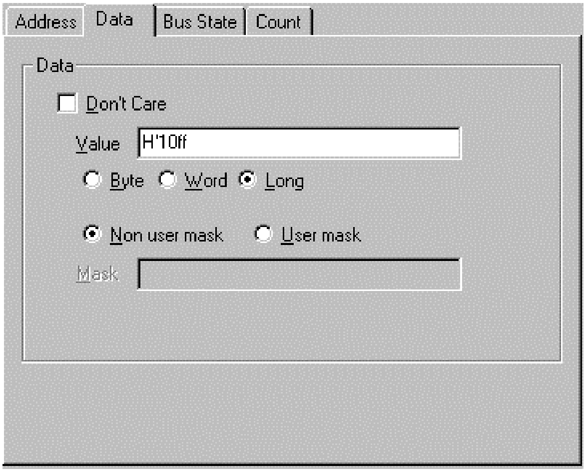


Figure 5.39 [Break Condition U1] Dialog Box ([Data] Page)

Table 5.47 [Data] Page Options

Option	Description
[Don't Care]	Selects no data condition.
[Value]	Sets a data bus value as a number.
[Byte]	Sets byte-data-access cycles.
[Word]	Sets word-data-access cycles.
[Long]	Sets longword-data-access cycles.
[Non user mask]	Sets no mask condition.
[User mask]	Sets mask conditions.
[Mask]	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.

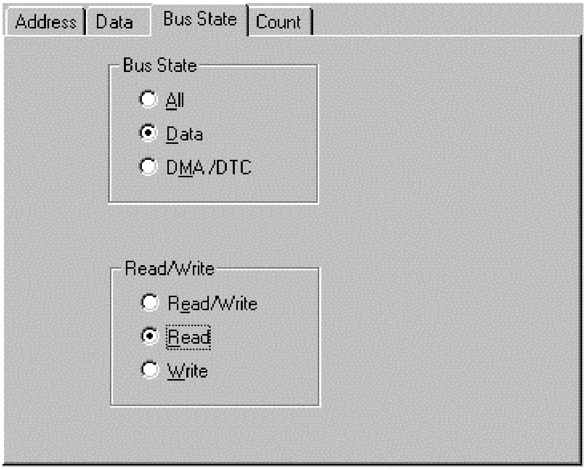


Figure 5.40 [Break Condition U1] Dialog Box ([Bus State] Page)

Table 5.48 [Bus State] Page Options

[Bus State] Group Box

Option	Description
[All]	All bus states satisfy this condition.
[Data]	Data access cycles satisfy this condition.
[DMA/DTC]	DMA/DTC cycles satisfy this condition.

[Read/Write] Group Box

Option	Description
[Read/Write]	Either read/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.

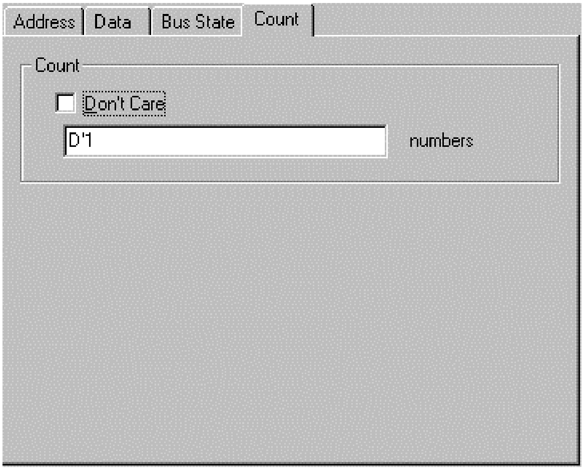


Figure 5.41 [Break Condition U1] Dialog Box ([Count] Page)

Table 5.49 [Count] Page Options

Option	Description
[Don't Care]	Selects no satisfaction count condition.
Input area	Sets the satisfaction count condition. The break occurs when the conditions specified in the [Break Condition U] dialog box have been satisfied the specified number of times. The default is D'1. Any value in the range from D'1 to D'65535 can be set here.

(e) [Break Condition U Reset] Dialog Box

Use this page to specify the reset point conditions for sequential breaks.

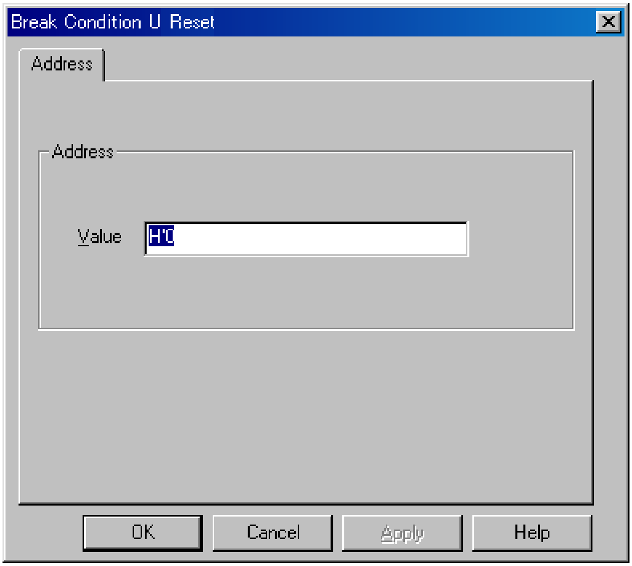


Figure 5.42 [Break Condition U Reset] Dialog Box

Table 5.50 [Break Condition U Reset] Page Options

Option	Description
[Address]	Specifies the reset point conditions when a sequential break is selected.

Note: Specification of reset point conditions is irrelevant if no internal sequential break has been set.

5.5.6 Internal Sequential Break

Overview: An internal sequential break set up with Break Condition U1 to U4 occurs when internal break conditions are satisfied in one of three specified orders.

These orders are referred to as modes and are shown in table 5.51. When a reset point condition is satisfied before satisfaction of Break Condition U1, the satisfaction record for the internal break conditions up to that point is cleared. The emulator then restarts checking for satisfaction of the internal sequential break conditions from the first condition.

Table 5.51 Internal Sequential Break Modes (Break Condition U1 to U4)

Mode	Description
1	A break occurs when internal break conditions U2 and U1 are satisfied, in that order.
2	A break occurs when internal break conditions U3, U2, and U1 are satisfied, in that order.
3	A break occurs when internal break conditions U4, U3, U2, and U1 are satisfied, in that order.

Setting an Internal Sequential Break: Specify the internal break conditions (U4–U1), then select the internal sequential break mode from [Emulation mode] on the [General] page of the [Configuration] dialog box. In the example shown in figure 5.43, internal sequential break mode 1 is selected.

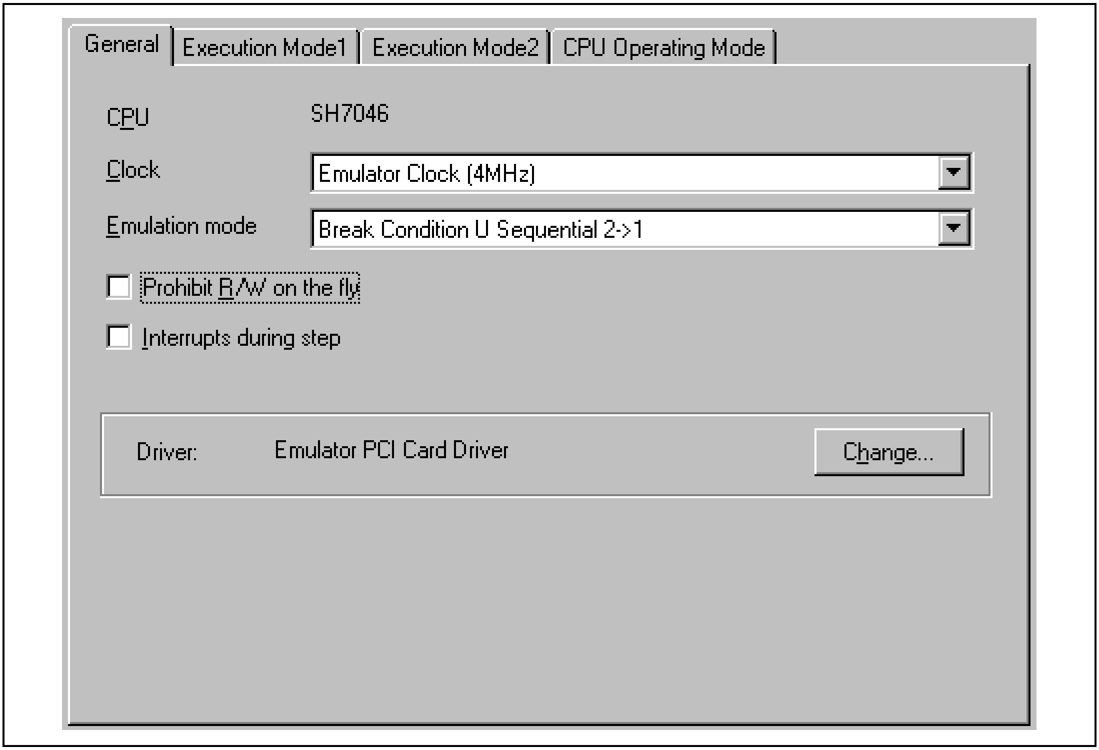


Figure 5.43 [Configuration] Dialog Box (Internal Sequential Break)

Any one of the modes shown in table 5.52 can be selected from [Emulation mode].

Table 5.52 [Emulation mode] Options (Break Condition U1 to U4)

Option	Description
Sequential break mode U2->1	Internal sequential break: mode 1
Sequential break mode U3->2->1	Internal sequential break: mode 2
Sequential break mode U4->3->2->1	Internal sequential break: mode 3

5.5.7 Forced Break

A user program can be forcibly terminated by clicking the [STOP] button or by selecting [Halt] from the [Run] menu. The system will leave trace-halt mode if trace-halt mode has been specified.

5.5.8 Forced Break on Writing to a Write-Protected Area or Access to an Access-Prohibited Area

Use the memory-allocation function to set up write-protected areas (including the internal ROM area) and access-prohibited areas. For details, refer to section 5.11, Memory Spaces.

5.5.9 Break Due to Trace-Buffer Overflow

A break occurs when the trace buffer in the E8000S station overflows during trace acquisition.

Select [Break] in [Buffer Over Flow] on the [Trace Mode] page of the [Trace Acquisition] dialog box.

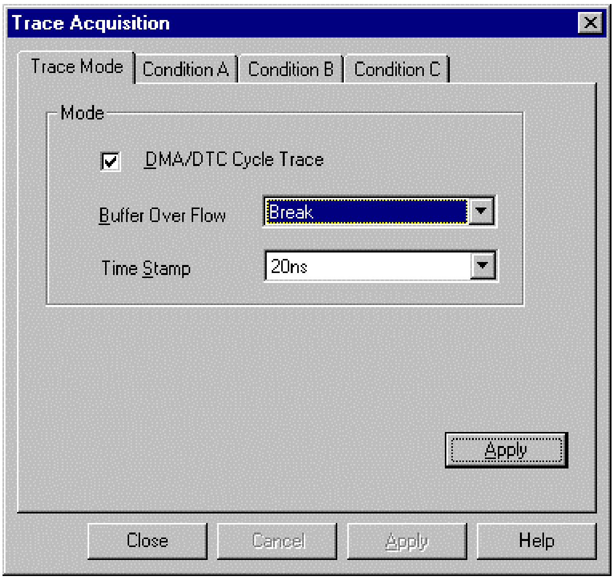


Figure 5.44 [Trace Acquisition] Dialog Box (Trace Buffer Overflow)

5.5.10 Timeout Break

A break occurs when the execution time or number of passes exceeds the conditions specified on the respective pages of Performance Analysis 1.

Select [Timeout break of Performance analysis] from [Emulation mode] in the [Configuration] dialog box.

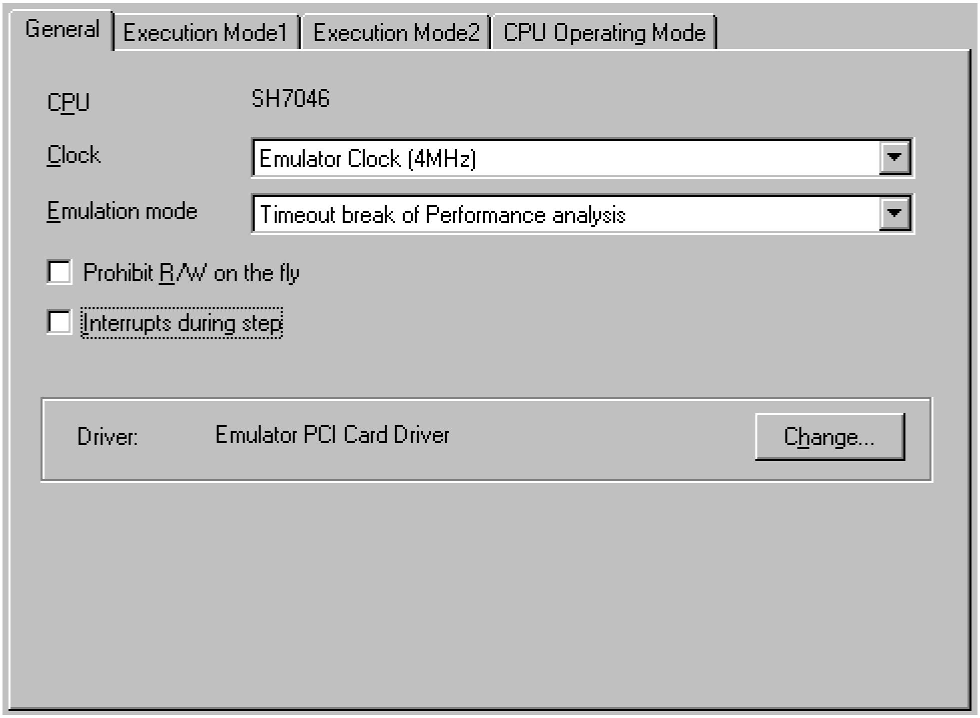


Figure 5.45 [Configuration] Dialog Box (Timeout Break)

Open the [Performance 1] dialog box from the [Performance] window, set the conditions on the [Time Out] page (timeout) and [Count] page (maximum number of executions), then execute the user program. When the specified time or number is exceeded, a break occurs.

For details on the [Performance 1] dialog box, [Time Out] page, and [Count] page, refer to section 5.8, Setting Performance Conditions.

Note: Software break conditions and software sequential break conditions are ignored.

5.6 Realtime Trace Functions

The emulator allows realtime trace acquisition of up to 131,070 cycles of the SH7046 bus information. This has no effect on the user system during emulation.

A total of up to 65,536 lines of information can be displayed in the HDI’s [Trace] window. For details on the [Trace] window, refer to the manual for the Hitachi Debugging Interface User’s Manual (on the CD-R).

5.6.1 External Bus Trace Function

Trace Conditions A, B, and C are available.

Specify a trace acquisition condition on the [Condition A, B, C] page of the [Trace Acquisition] dialog box which will be displayed by clicking the [Acquisition…] in the pop-up menu of the [Trace] window.

The trace acquisition modes for external trace are shown in table 5.53.

Table 5.53 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.
Trace stop	Trace acquisition stops when a specified condition is satisfied. In this mode, realtime emulation will not stop, but trace acquisition is stopped, and emulation enters the trace-halt mode.
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 A and B, but not for Trace Condition C.
Trace stop due to an overflow of trace buffer	Trace acquisition stops when the trace buffer in the emulator overflows.
Range trace	Trace information is only acquired during execution that satisfies the specified conditions.
Subroutine 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.
Range trace in subroutines	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.
Trigger output	A pulse is output from the trigger pin when the specified conditions are satisfied.
Timeout trace stop	Trace acquisition stops when the timeout condition specified for Performance Analysis 1 has been exceeded.

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 no trace condition is specified.

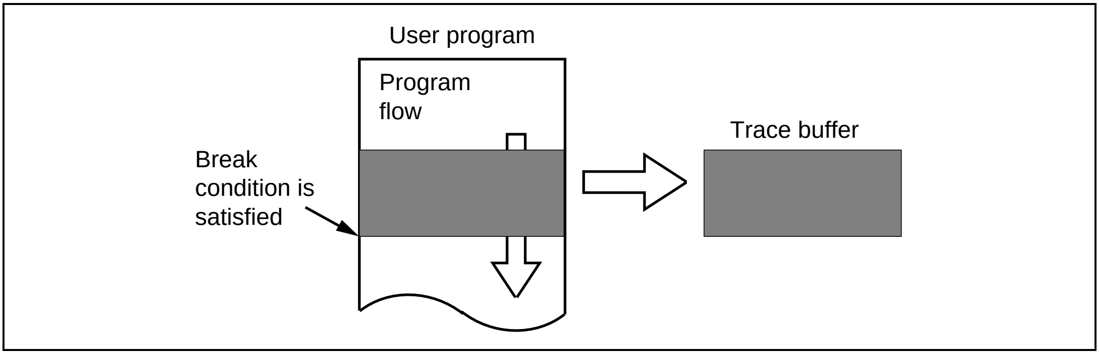


Figure 5.46 Trace Acquisition in Free Trace Mode

Trace-Stop Mode:

(a) Overview

Trace acquisition stops when the specified conditions are satisfied.

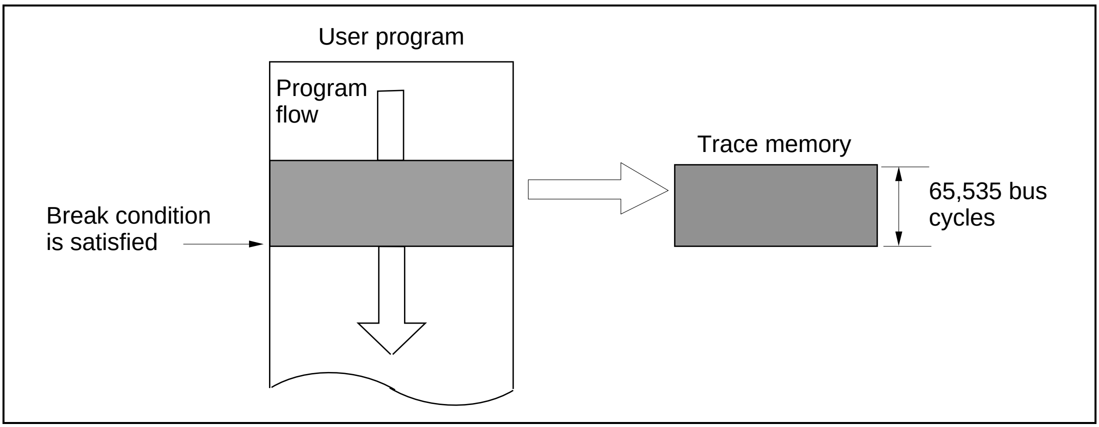


Figure 5.47 Trace Acquisition in Trace-Stop Mode

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 5.54. When all of the specified conditions (an AND condition) are satisfied, trace acquisition will stop.

Table 5.54 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.
Access type	The condition is satisfied when the access type matches the specification.
Read/Write	The condition is satisfied when the read/write condition is as specified.
External interrupt	The condition is satisfied when the external interrupt signal levels match a specification.
External probe	The condition is satisfied when the external probe (PRB) 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.

Trace-stop conditions that can be specified for each of the eight channels of Trace Conditions A, B, and C (24 channels in total) are shown in table 5.55.

Table 5.55 Specifiable Trace-Stop Conditions

Trace-Stop Condition	Trace Condition A (1 to 8)	Trace Condition B (1 to 8)	Trace Condition C (1 to 8)
Address bus	O	O	O
Data bus	O	O	X
Access type	O	O	O
Read/Write	O	O	X
External interrupt	O	O	X
External probe	O	O	X
Satisfaction count	O	O	X
Delay	O	O	X

Notes: O: Can be specified.
X: Cannot be specified.
The delay condition is only available for Trace Conditions A7 and B7.

(b) Setting Trace-Stop Conditions

Trace Condition A7 is taken as an example of setting a trace stop 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] dialog box will appear. Select the [Condition A] page.

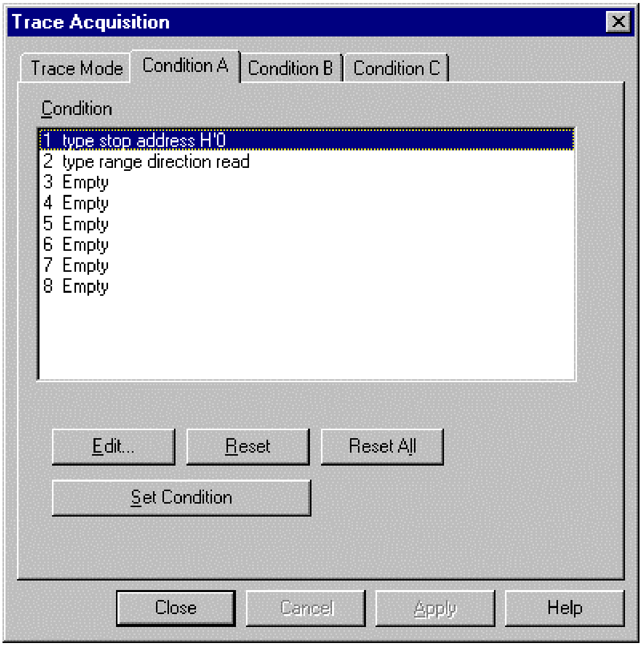


Figure 5.48 [Trace Acquisition] Dialog Box ([Condition A] Page)

Table 5.56 [Condition A, B, C] Page Options

Option	Description
[Condition]	Displays the current settings for Trace Condition A, B, or C. Empty is displayed if there are no settings. 1 to 8: Settings for Trace Condition X1 to X8 (X: A, B, or C.)
[Edit...]	Modifies the Trace Condition A, B, or C settings selected in the [Condition] list box. Clicking this button opens the [Trace Condition Xn] dialog box. (X: A, B, or C; n: channel number.)
[Set Condition]	Modifies the Trace Condition A or B settings selected in the [Condition] list box. Clicking this button opens the [Condition] dialog box. (This option is not displayed on the [Condition C] page.)
[Reset]	Clears the Trace Condition A, B, or C settings selected in the [Condition] list box.
[Reset All]	Clears all Trace Condition A, B, or C settings in the [Condition] list box.

Click condition 1 to select it from the [Condition] list box. Click the [Edit...] button to open the [Trace Condition A7] dialog box.

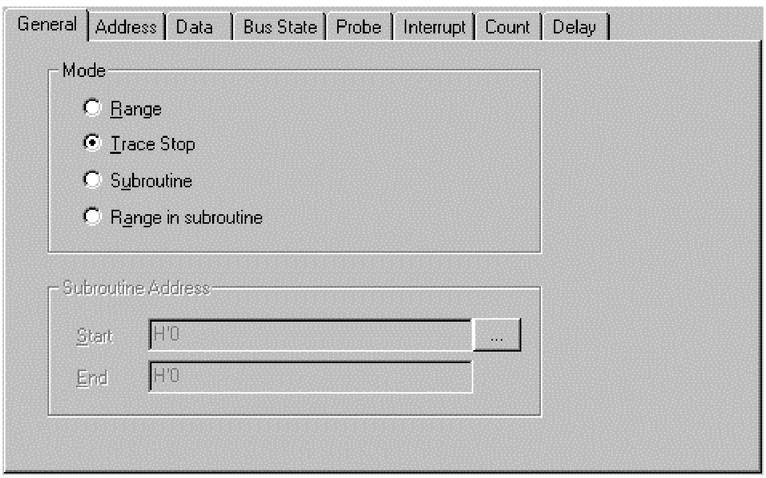


Figure 5.49 [Trace Condition A7] Dialog Box ([General] Page)

Select [Trace Stop] on the [General] page.

The [Trace Condition A1 to A8, B1 to B8, C1 to C8] dialog box has the tabbed pages listed in table 5.57.

Table 5.57 [Trace Condition A1 to A8, B1 to B8, C1 to C8] Dialog Box Pages

Dialog Box	Page	Description
[Trace Condition A1 to A6 and A8], [Trace Condition B1 to B6 and B8]	[General]	Selects the trace acquisition mode.
	[Address]	Sets address bus conditions.
	[Data]	Sets data bus conditions.
	[Bus State]	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.
[Trace Condition A7], [Trace Condition B7]	[General]	Selects the trace acquisition mode.
	[Address]	Sets address bus conditions.
	[Data]	Sets data bus conditions.
	[Bus State]	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.
[Trace Condition C1 to C8]	[General]	Selects the trace acquisition mode.
	[Address]	Sets address bus conditions.
	[Bus State]	Sets access type conditions.

Specify the required conditions on the corresponding pages, then click the [OK] button. The [Trace condition A7] dialog box closes and the display returns to the [Condition A] page. The specified trace-stop conditions will now be displayed as condition 7 in the [Condition] list box. Click the [Close] button to close the [Trace Acquisition] dialog box.

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

The options on each page are the same as those of the corresponding [Break Condition A1 to A8, B1 to B8, C1 to C8] dialog boxes, except the [General] page. For details of the options on each page, refer to section 5.5.3, Hardware Break.

When the trace conditions are satisfied during emulation, "TRACE STOP" will appear in a dedicated message box or on the status bar, and the emulator will enter the trace-halt mode.

To leave trace-halt mode and reactivate the emulation, select [Halt] from the pop-up menu in the [Trace] window, or execute the END command in the [Command Line] window. To leave the trace-halt mode and end emulation, execute the HALT command in the [Command Line] window.

Notes:
1. Trace Condition A1 to A8 share hardware with Break Condition A1 to A8. Therefore, when any channel of Break Condition A1 to A8 has been specified, it is not possible to set or modify Trace Condition A1 to A8.
2. Trace Condition B1 to B8 share hardware with Break Condition B1 to B8. Therefore, when any channel of Break Condition B1 to B8 has been specified, it is not possible to set or modify Trace Condition B1 to B8.
3. Trace Condition C1 to C8 share hardware with Break Condition C1 to C8 and Performance Analysis 1 to 8. Therefore, when any channel of Break Condition C1 to C8 or Performance Analysis 1 to 8 has been specified, it is not possible to set or modify Trace Condition C1 to C8.

(c) Displaying a History of Trace Stop Conditions and Creating a New Condition

A history of trace stop conditions that have been set as Trace Condition A or B can be displayed in the [Condition] dialog box. New conditions can also be created by using the [Condition] dialog box.

Click to select condition 1 from the [Condition] list box on the [Condition A] page of the [Trace Acquisition] dialog box. Click the [Set Condition] button to open the [Condition] dialog box.

The [Condition] dialog box has the pages shown in table 5.58.

Table 5.58 Pages of the [Condition] Dialog Box Pages

Page	Description
[History]	Displays the history of conditions that have been set as Trace Condition A or B. Conditions can be moved from the [History] page to the [Entry List] page.
[Entry List]	Creates, modifies, duplicates, and deletes conditions for Trace Condition A or B.

For details on each page, refer to the description under Displaying a History of Hardware Break Conditions and Creating a New Condition in section 5.5.3, Hardware Sequential Break.

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 Trace Condition A or B; a sequence of up to seven trace-stop conditions and one reset point can be specified as Trace Condition A or B.

The user must consider the order of satisfaction in specifying trace-stop conditions for Trace Conditions A and B; 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:

Select [Trace] for [Condition A] under [Sequence] on the [Execution Mode2] page of the [Configuration] dialog box. (The same setting procedure can be used on the [Condition B] page.)

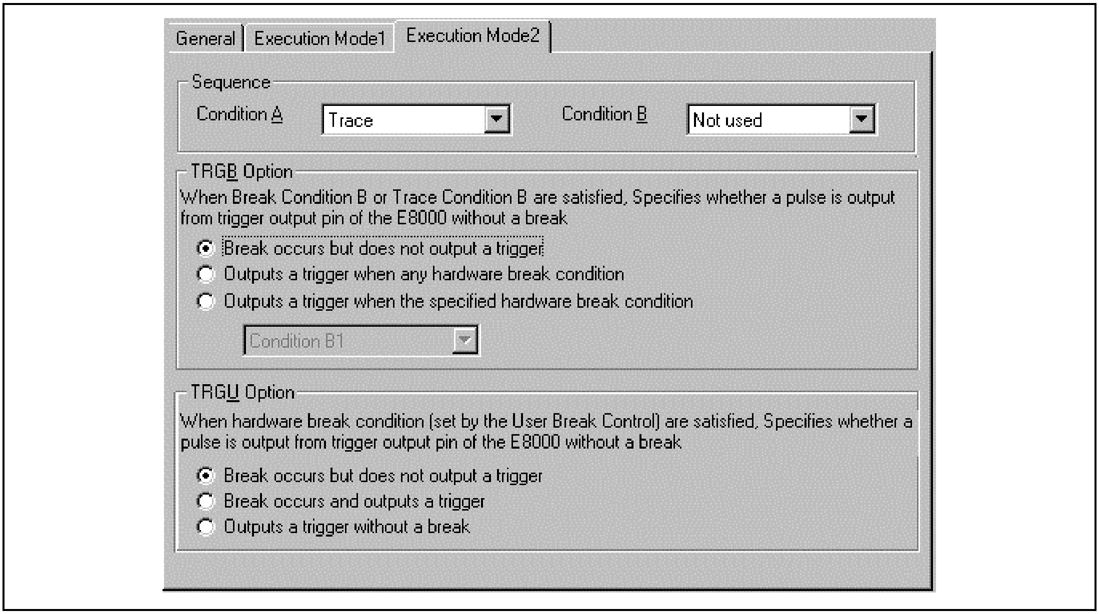


Figure 5.50 [Configuration] Dialog Box (Sequential Trace Stop)

This setting selects Trace Condition A for use as a sequential trace stop. At this time, any existing condition settings for Trace Condition A and Break Condition A are cleared. Therefore, any trace stop conditions which you wished to use must be specified again.

Select the [Condition A] page of the [Trace] window using the same procedure described when setting trace stop conditions (the same setting procedure applies to the [Condition B] page).

For details on the [Condition A] and [Condition B] pages, refer to section 5.5.4, Hardware Sequential Break.

Click the [Edit...] button to open the [Condition] dialog box.

For details on the procedure for registering sequential trace-stop conditions and the display of the number of times the conditions are satisfied on the status bar, refer to section 5.5.4, Hardware Sequential Break.

Trace Stop Due to Trace Buffer Overflow: Trace acquisition can be stopped when the trace buffer in the emulator overflows.

Select [Trace stop] under [Buffer Over Flow] on the [Trace Mode] page of the [Trace Acquisition] dialog box.

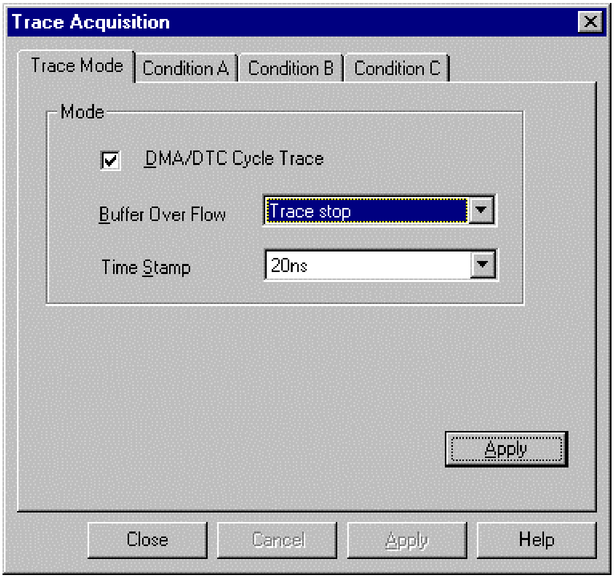


Figure 5.51 [Trace Acquisition] Dialog Box ([Trace Mode] Page)

Range Trace Mode:

Overview

Information is only acquired while the specified conditions are satisfied.

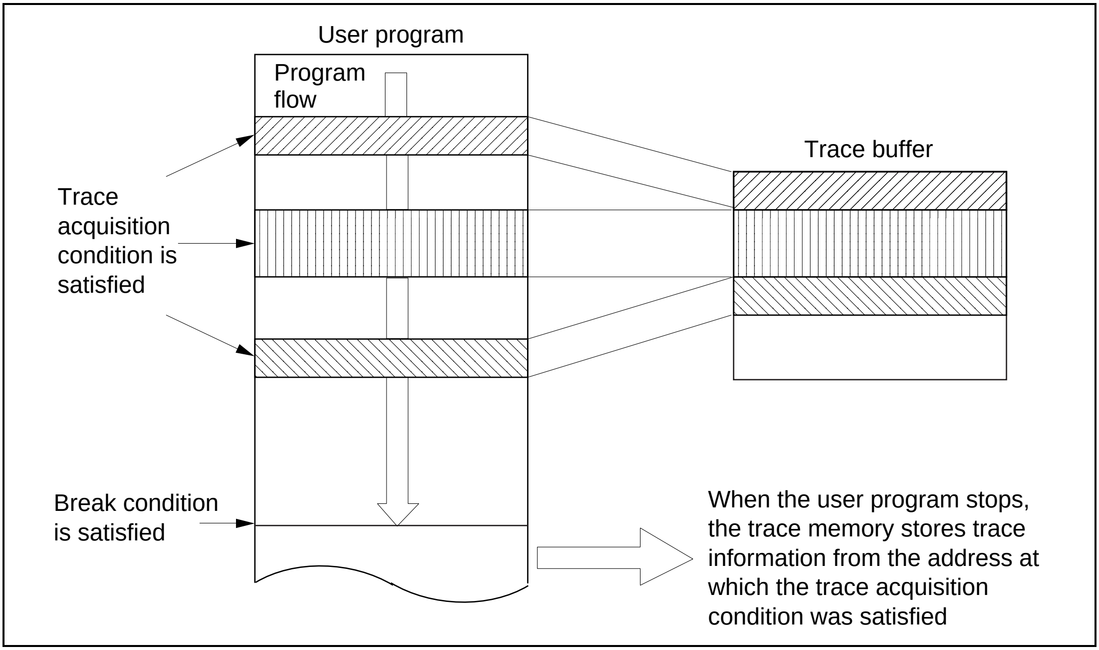


Figure 5.52 Example of Range Trace Mode

The conditions for range tracing are shown in table 5.59. Information is acquired when all of the specified conditions (an AND condition) are satisfied.

Table 5.59 Range Trace 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.
Access type	Satisfied when the bus status condition is as specified.
Read/write	Satisfied when the read/write matches the specified condition.
External interrupt	Satisfied when the external interrupt signal levels match the specified conditions.
External probe	Satisfied when the external probe (PRB) signal levels match the specified conditions.

Table 5.60 Specifiable Range Trace Conditions

Range Trace Condition	Trace Condition A (1 to 8)	Trace Condition B (1 to 8)	Trace Condition C (1 to 8)
Address bus	O	O	O
Data bus	O	O	X
Access type	O	O	X
Read/Write	O	O	O
External interrupt	O	O	X
External probe	O	O	X

Note: O: Can be specified.
X: Cannot be specified.

Trace Condition A1 is taken as an example of setting a range trace 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] dialog box will appear. Select the [Condition A] page.

Click condition 1 to select it from the [Condition] list box. Click the [Edit...] button to open the [Trace Condition A1] dialog box.

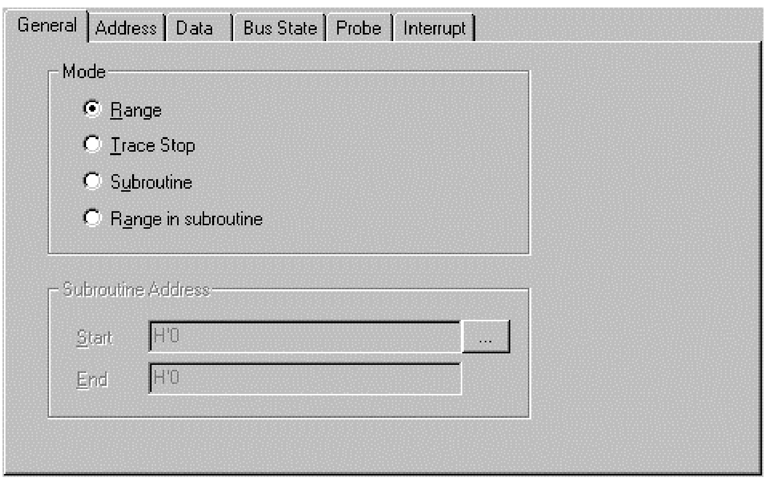


Figure 5.53 [Trace Condition A1] Dialog Box ([General] Page)

Select [Range] (range trace mode) on the [General] page.

The [Trace Condition A1 to A8, B1 to B8, C1 to C8] dialog box has the tabbed pages listed in table 5.61. However, [Trace Condition C1 to C8] dialog box has only the [General], [Address], and [Bus State] pages.

Table 5.61 [Trace Condition A1 to A8, B1 to B8, C1 to C8] Dialog Box Pages

Dialog Box	Page	Description
[Trace Condition A1 to A8] [Trace Condition B1 to B8]	[General]	Selects the range trace mode.
	[Address]	Sets address bus conditions.
	[Data]	Sets data bus conditions.
	[Bus State]	Sets read/write cycle conditions.
	[Probe]	Sets external probe signal conditions.
	[Interrupt]	Sets external interrupt signal conditions (NMI and IRQ0 to IRQ7).
[Trace Condition C1 to C8]	[General]	Selects the range trace mode.
	[Address]	Sets address bus conditions.
	[Bus State]	Sets access type conditions.

Specify the required conditions on the corresponding pages, then click the [OK] button. The [Trace condition A1] dialog box closes and the display returns to the [Condition A] page. The specified trace conditions will now be displayed as condition 1 in the [Condition] list box. Click the [Close] button to close the [Trace Acquisition] dialog box.

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

The options on each page are the same as those of the corresponding [Break Condition A1 to A8, B1 to B8, C1 to C8] dialog boxes, except the [General] page. For details of the options on each page, refer to section 5.5.3, Hardware Break.

Subroutine Tracing Mode:

Overview

Access to instructions and operands within the range between the specified start and end addresses (of the subroutine) is traced in this mode. However, if another subroutine is called by the specified subroutine, the called subroutine will not be traced.

Eight channels for each of Trace Conditions A, B, and C (24 channels in total) can be specified.

Trace Condition A1 is used as an example of setting a range trace 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] dialog box will appear. Select the [Condition A] page.

Click condition 1 to select it from the [Condition] list box. Click the [Edit...] button to open the [Trace Condition A1] dialog box.

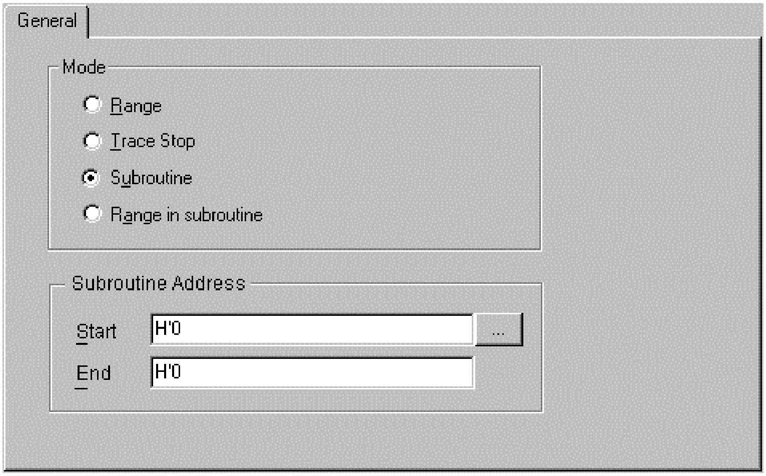


Figure 5.54 [Trace Condition A1] Dialog Box ([General] Page)

Select [Subroutine] (subroutine trace mode) on the [General] page.

Click the [OK] button to return to the [Condition A] page. The [Sequential Condition] list displays the sequential trace stop conditions that have been specified.

Specify the subroutine address range in the [Subroutine Address] group box. If […] is selected as the address range, the [Input Function Range] dialog box will be displayed. When the name of a subroutine is specified in the [Input Function Range] dialog box, the start and end addresses of the subroutine will automatically be displayed.

Click the [OK] button.

Close the [Trace Condition A1] dialog box to return to the [Condition A] page. The [Condition] list displays the conditions for trace acquisition that have been specified. Click the [Close] button to close the [Trace Acquisition] dialog box.

Subroutine trace conditions for the other channels are specified in the same way.

Subroutine Range Tracing Mode:

Overview

Trace information is acquired when instructions and operands within the range between the start and end addresses (of a subroutine) are accessed, but only on those bus cycles where the specified conditions are matched.

Table 5.62 shows the conditions for the subroutine range trace mode. When all of the specified conditions are satisfied (an AND condition), trace information will be acquired.

Table 5.62 Conditions of Trace Mode within a Subroutine Range

Break 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.
Access type condition	Satisfied when the access type matches the specified bus status condition.
Read/write cycle	Satisfied when the cycle is a read/write cycle as specified.
External interrupt signal	Satisfied when the levels of the external interrupt signals matches the specified condition.
External probe signal	Satisfied when the levels of the external probe (PRB) signals matches the specified condition.

Subroutine Range Trace conditions that can be specified for channels 1, 3, 5, and 7 of Trace Conditions A, and B are shown in table 5.63.

Table 5.63 Specifiable Range Trace Conditions

Condition	Trace Condition A (1, 3, 5, 7)	Trace Condition B (1, 3, 5, 7)
Address bus	O	O
Data bus	O	O
Access type	O	O
Read/Write	O	O
External interrupt	O	O
External probe	O	O

Note: O: Can be specified.
X: Cannot be specified.

Trace Condition A1 is used as an example of setting a subroutine range trace 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] dialog box will appear. Select the [Condition A] page.

Click condition 1 to select it from the [Condition] list box. Click the [Edit...] button to open the [Trace Condition A1] dialog box.

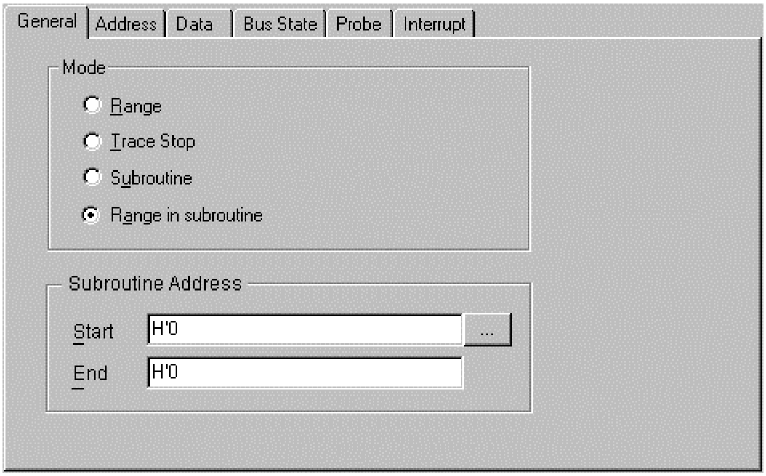


Figure 5.55 [Trace Condition A1] Dialog Box ([General] Page)

Select [Range in subroutine] (subroutine range trace mode) on the [General] page.

The [Trace Condition A1 to A8, B1 to B8, C1 to C8] dialog box has the tabbed pages listed in table 5.64.

Table 5.64 [Trace Condition A1 to A8, B1 to B8, C1 to C8] Dialog Box Pages

Dialog Box	Page	Description
[Trace Condition A1, 3, 5, 7] [Trace Condition B1, 3, 5, 7]	[General]	Selects the trace acquisition mode.
	[Address]	Sets address bus conditions.
	[Data]	Sets data bus conditions.
	[Bus State]	Sets read/write cycle conditions.
	[Probe]	Sets external probe signal conditions.
	[Interrupt]	Sets external interrupt signal conditions.

Specify the required conditions on the corresponding pages, then click the [OK] button. The [Trace condition A1] dialog box closes and the display returns to the [Condition A] page. The specified trace conditions will now be displayed as condition 1 in the [Condition] list box. Click the [Close] button to close the [Trace Acquisition] dialog box.

Range trace conditions for the other channels are specified in the same way.

The options on each page are the same as those in the corresponding [Break Condition A1 to A8, B1 to B8, C1 to C8] dialog boxes, except the [General] page. For details of the options on each page, refer to section 5.5.3, Hardware Break.

Trigger Output: A low-level pulse is output from the trigger-output probe when the conditions specified for Trace Condition B are satisfied during execution of the user program.

Timeout Trace Stop: Trace acquisition can be stopped when execution time or number of passes exceeds the respective conditions (timeout or maximum number of passes) specified in Performance Analysis 1.

To use this function, select [Timeout trace of Performance Analysis] under [Emulation mode] in the [Configuration] dialog box.

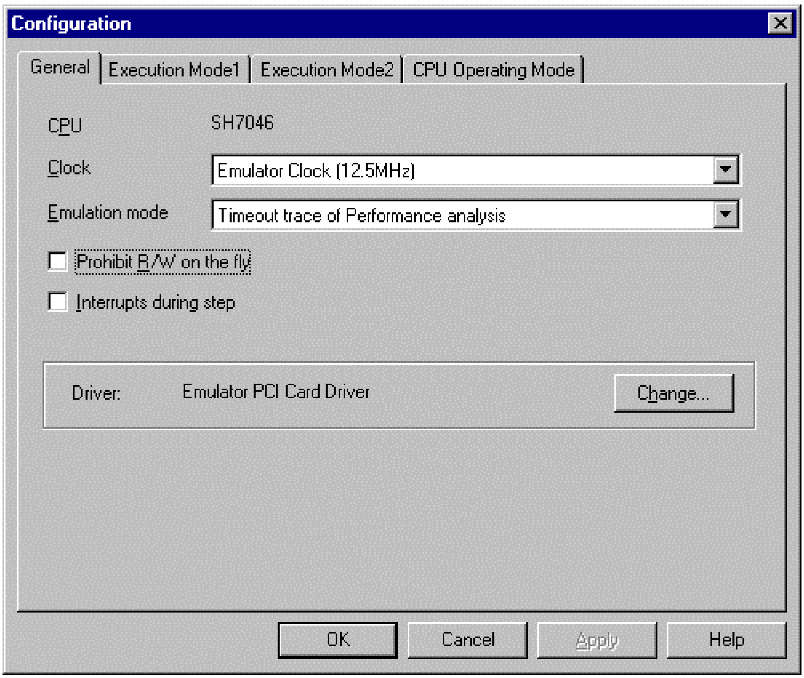


Figure 5.56 [Configuration] Dialog Box ([General] Page)

Specify the timeout on the [Time Out] page and the maximum number of passes in execution on the [Count] page of the [Performance 1] dialog box, which can be opened from the [Performance] window, then execute the user program. When either the execution time or count exceeds the specified conditions, trace acquisition will stop.

For details on the [Performance 1] dialog box, [Time Out] page, and [Count] page, refer to section 5.8, Performance Analysis Function.

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

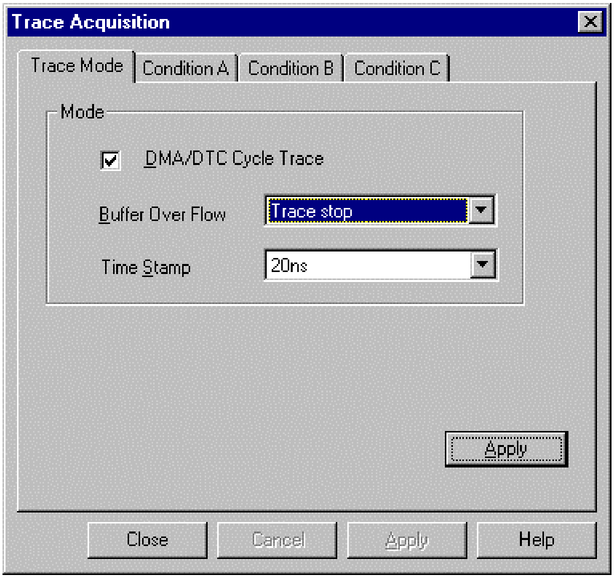


Figure 5.57 [Trace Acquisition] Dialog Box ([Trace Mode] Page)

(a) Acquisition on DMA/DTC cycles

Acquisition of trace information on DMA or DCT cycles is selected in the [DMA/DCT Cycle Trace] check box in the [Trace Mode] page of the [Trace Acquisition] dialog box.

(b) Minimum period for time stamping

The minimum period for time stamping is specified as [Time Stamp] in the [Trace Mode] page of the [Trace Acquisition] dialog box.

Table 5.65 [Time Stamp] Group Box Options

Option	Description
[Time Stamp]	Selects the minimum time for the time stamping of the measured bus trace information from among the values listed below. 20ns: Time stamping is in minimum time units of 20 ns (default). 1.6us: Time stamping is in minimum time units of 1.6 µs. 52us: Time stamping is in minimum time units of 52 µs. CPU clock: Time stamping is in terms of the number of bus-clock cycles, i.e., is synchronized with the cycles of the system clock signal. 1/2 CPU clock: Time stamping is in terms of the number of bus-clock cycles, i.e., is synchronized with 1/2 cycles of the system clock (φ) signal. 1/4 CPU clock: Time stamping is in terms of the number of bus- clock cycles, i.e., is synchronized with 1/4 cycles of the system clock (φ) signal. 1/8 CPU clock: Time stamping is in terms of the number of bus-clock cycles, i.e., is synchronized with 1/8 cycles of the system clock (φ) signal.

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

5.6.2 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 access to an area of normal SRAM is shown in figure 5.58

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.

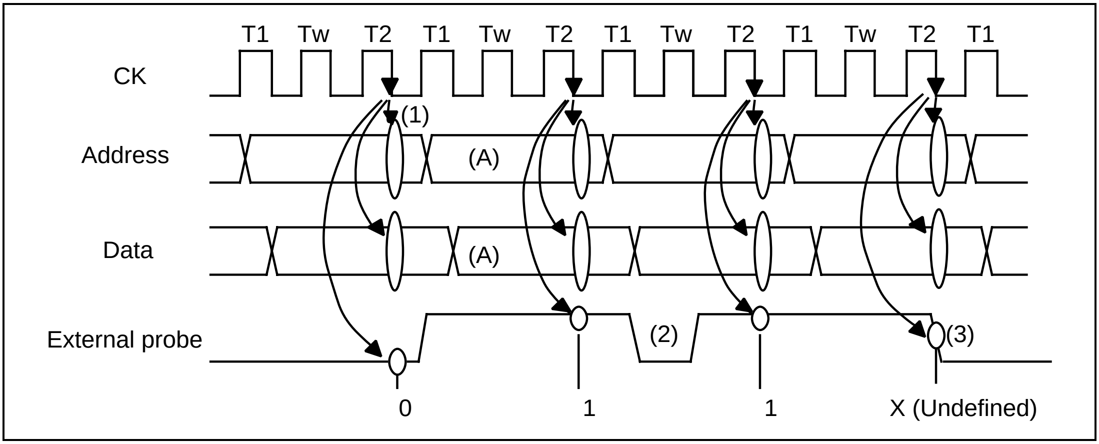


Figure 5.58 Tracing External Probe Signal

(1) External probe signal

(a) Information is traced on the falling edge of the T2 cycle of the CK signal (figure 5.58 (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 5.58 (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 5.58 (3)). If the sampling edge and any asynchronous input signal that is being traced changes at the same time, the information traced will be undefined.

(2) Number of clocks

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

5.6.3 Trace Display

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

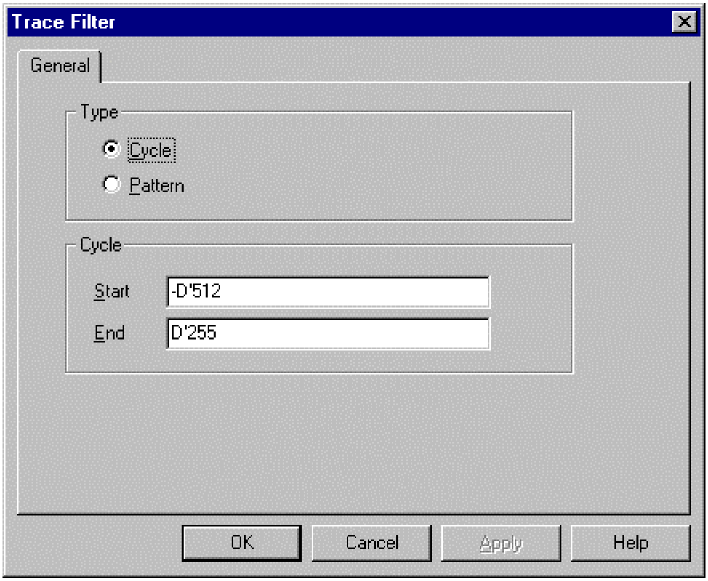


Figure 5.59 [Trace Filter] Dialog Box ([General] Page)

When there is no trace information the [Trace] window will initially display nothing. It will display “no trace record” once it has been updated.

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 (+).

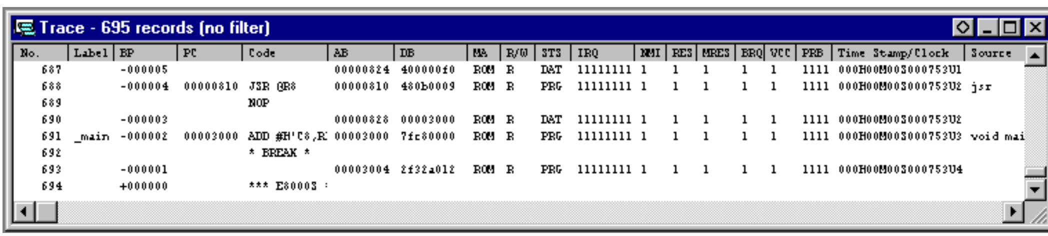


Figure 5.60 [Trace] Window

The items shown in table 5.66 are displayed as trace information in the [Trace] window.

Table 5.66 Trace Information Items and Display Format in [Trace] Window

Item	Description and Format
No	Line number in the [Trace] window.
Label	Label name.
BP	Bus cycle pointer. The location of a bus cycle relative to the bus cycle where the delay condition has been satisfied. Pointers are usually negative values (-xxxxxx), but when a delay condition has been specified as a break or trace condition, the bus cycles during the delay period are positive (+xxxxxx).
PC	32-bit program-counter values.
Code	Instruction code being executed. Nothing is displayed here in cycles that are not instruction-execution cycles. * Break *: The contents of the specified address where a software breakpoint or software sequential breakpoint has been set are replaced by a break instruction and executed. ***E8000S***: Invalid cycles
AB	32-bit address bus values.
DB	32-bit data bus values in 4-bit units.
MA	Type (area) of memory accessed by the bus master. EXT: Emulation/non-emulation memory ROM: Internal ROM IO: Internal I/O area RAM: Internal RAM NON: No access
R/W	Whether the cycle was for reading or writing. R: read cycle W: write cycle
STS	DMA: Internal DMA/DTC execution cycle AUD: AUD cycle BRL: User bus release DAT: CPU data-access cycle (except for PC-relative data access) PRG: CPU instruction-fetch cycle (includes PC-relative data access)
IRQ	IRQ0 to IRQ7 signal state. x7x6x5x4x3x2x1x0 (xn is the state of IRQn) (0: low level; 1: high level)
NMI	NMI signal state. (0: low level; 1: high level)
RES	RES signal state. (0: low level; 1: high level)
MRES	MRES signal state. (0: low level; 1: high level)
BRQ	BREQ signal state. (0: low level; 1: high level)
VCC	Voltage on VCC. (1: search level or more; 0: less than search level)

Table 5.66 Trace Information Items and Display Format in [Trace] Window (cont)

Item	Description and Format
PRB	External probe (PRB) signal state. x4x3x2x1 (xn is the state of PRBn) (0: low level; 1: high level)
Time Stamp	Time stamp. xxxHxxMxxSxxxxxxUxxxN (H: hour; M: minute; S: second; U: microsecond; N: nanosecond)
Clock	Number of clock cycles from the end of the previous bus cycle to the end of the current bus cycle. xx: Hexadecimal value. Up to 255 clock cycles can be counted. When execution has continued for more than 255 clock cycles, ** is displayed here. Only one of Time Stamp or Clock can be displayed at a time.
Source	The corresponding line of source code to the program counter. Clicking in the Source column activates the [Source] window and jumps to the corresponding line. The contents of the source column are not displayed in external bus trace.

Note: The correspondence between pins and signals must be set correctly set in the [CPU Operating Mode] dialog box to correctly trace the IRQ0 to IRQ7 and BREQ signals.
The voltage level used by the VCC voltage is specified in the [CPU Operating Mode] dialog box.

5.6.4 Trace Search Functions

The emulator has the two functions for searching for trace information that are shown in table 5.67.

Table 5.67 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] 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] dialog box will appear. Select the

[General] page of the [Trace Filter] dialog box, then select the [Pattern] radio button in the [Type] group box.

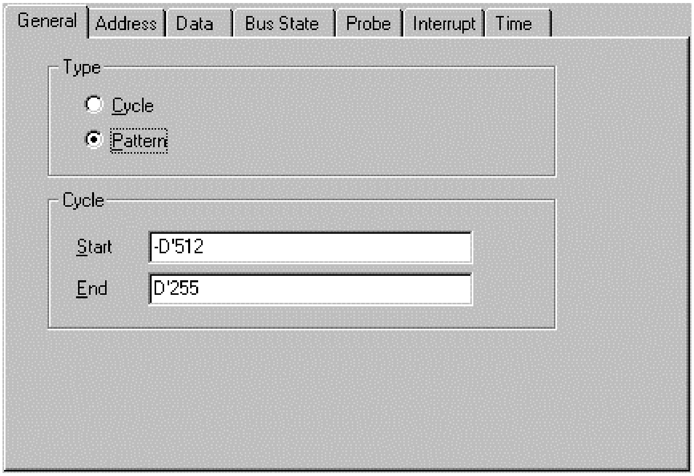


Figure 5.61 [Trace Filter] Dialog Box ([General] Page)

Table 5.68 [General] Page Options

Option	Description
[Type]	Selects a format for the display of trace information. [Cycle]: Disables any search conditions set in the [Trace Filter] dialog box. Displays the whole contents of the trace buffer in the [Trace] window. [Pattern]: Searches under the conditions set in the [Trace Filter] dialog box, and displays the results in the [Trace] window.
[Cycle]	Sets a range to be displayed. 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. The default values are -D'512 for [Start] and D'255 for [End].

Specify the required conditions on the corresponding pages, then click the [OK] button. The [Trace Filter] 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] dialog box are shown in table 5.69.

Table 5.69 Trace Search Conditions and Pages in the [Trace Filter] 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 State]	Bus state	Searches for records in which access type, read/write access, and BREQ signal match the specified condition.
[Probe]	External probe	Searches for records in which the external probe signal levels match the specified condition.
[Interrupt]	External interrupt	Searches for records in which the NMI, RES, MRES signal, 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] (except for [Outside Range] items) and [Probe] is the same as setting the corresponding Break Condition. For details on specifying the conditions, refer to section 5.5.3, Hardware Break.

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

(a) [Address] Page

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

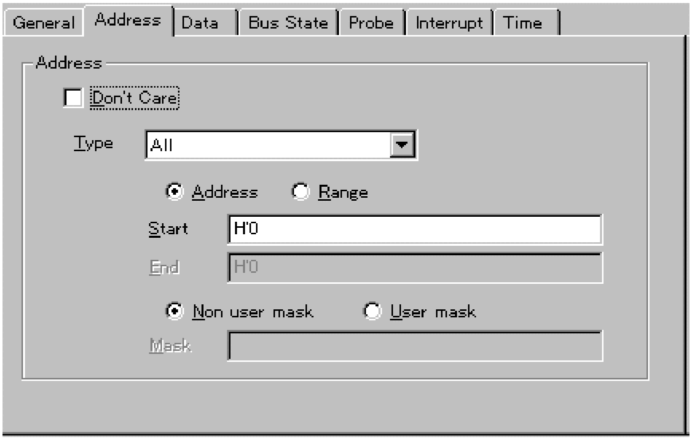


Figure 5.62 [Trace Filter] Dialog Box ([Address] Page)

Table 5.70 [Address] Page Options

Option	Description
[Don't Care]	No address bus condition is set.
[Type]	Type of address as a search condition. [Address] to [Mask] are only available when address types [All] or [PC] are specified. All: The address bus settings from [Address] to [Mask] are available. PC: The program counter settings from [Address] to [Mask] are available. External space: Emulation/non-emulation memory area Internal ROM space: Internal ROM area Internal I/O space: Internal I/O area Internal RAM space: Internal RAM area
[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 numerical value or a symbol.
[End]	When [Range] is selected, sets the (end) address value as a numerical value or symbol.
[Non user mask]	A mask condition is not specified.
[User mask]	A mask condition is specified.
[Mask]	When [Address] and [User mask] is selected, the mask is set. When [Range] is specified, the value becomes invalid.

(b) [Bus State] Page

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

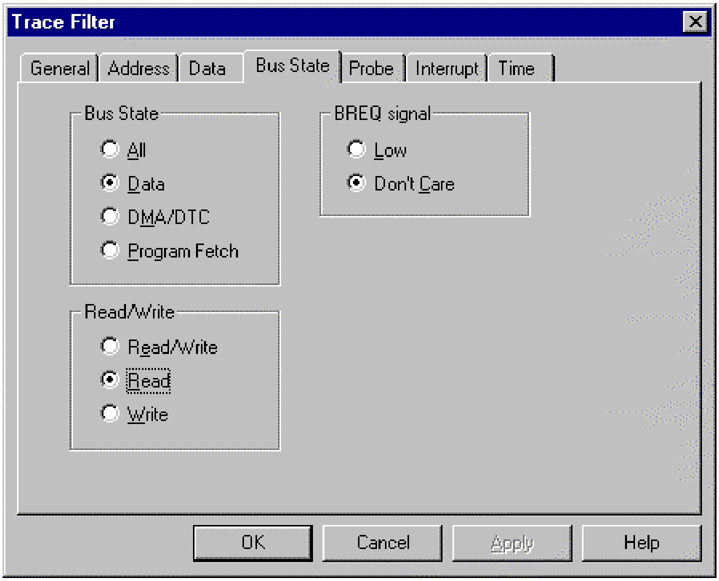


Figure 5.63 [Trace Filter] Dialog Box ([Bus State] Page)

Table 5.71 [Bus State] Page Options

[Bus State] Group Box

Option	Description
[All]	This search condition is any access type.
[Data]	This search condition instruction execution cycles only.
[DMA/DTC]	This search condition is DMA/DTC cycles only.
[Program Fetch]	This search condition is CPU instruction fetch cycles only.

[Read/Write] Group Box

Option	Description
[Read/Write]	Sets either read/write cycles as a search condition.
[Read]	Sets read cycles as the search condition.
[Write]	Sets write cycles as the search condition.

[BREQ signal] Group Box

Option	Description
[Low]	Sets the low level of the BREQ signal as a search condition.
[Don't Care]	The BREQ signal is not a search condition.

(c) [Interrupt] Page

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

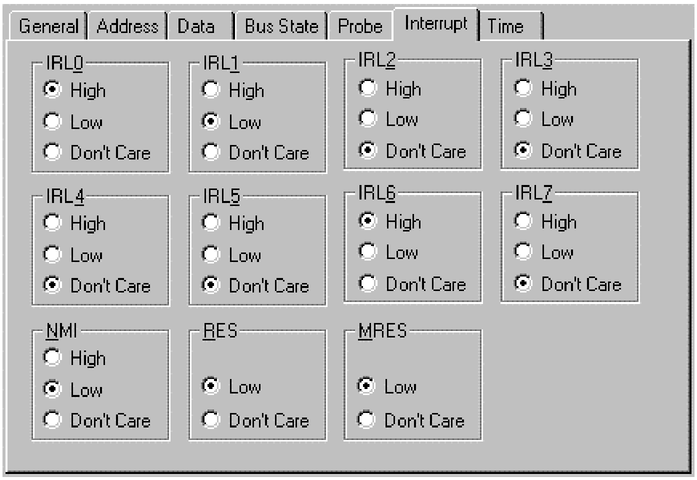


Figure 5.64 [Trace Filter] Dialog Box ([Interrupt] Page)

Table 5.72 [Interrupt] Page Options

[IRQ0 to IRQ7] Group Box

Option	Description
[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.
[Don't Care]	The state of the external interrupt signal is not a search condition.

[NMI] Group Box

Option	Description
[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.
[Don't Care]	The NMI signal is not a search condition.

[RES] Group Box

Option	Description
[Low]	Sets the low level of the RES signal as a search condition.
[Don't Care]	The RES signal is not a search condition.

[MRES] Group Box

Option	Description
[Low]	Sets the low level of the MRES signal as a search condition.
[Don't Care]	The MRES signal is not 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.

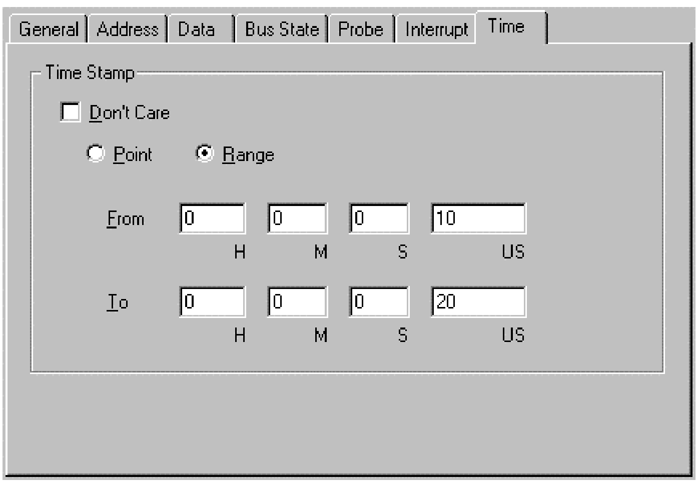


Figure 5.65 [Trace Filter] Dialog Box ([Time] Page)

Table 5.73 [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 numerical value (decimal). When any number is omitted, 0 is assumed. [H]: hour (0 to 999) [M]: minute (0 to 59) [S]: second (0 to 59) [US]: microsecond (0 to 999999)
[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. [H]: hour (0 to 999) [M]: minute (0 to 59) [S]: second (0 to 59) [US]: microsecond (0 to 999999)

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

Trace Find Function: 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] dialog box. Search conditions are the same as those for the trace filter functions, except on the [General] page.

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

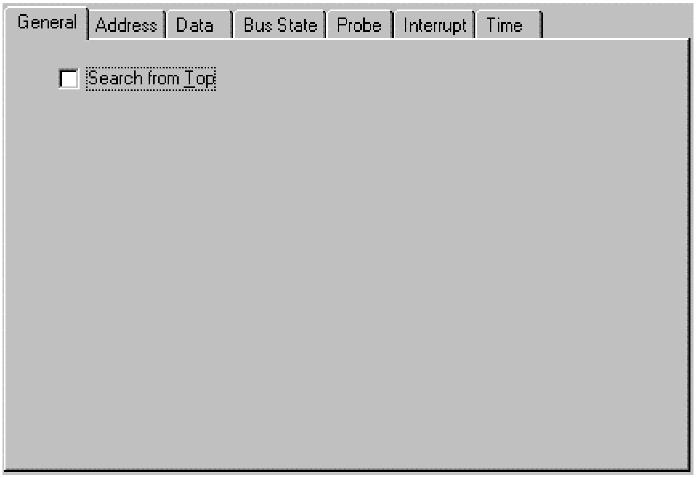


Figure 5.66 [Trace Find] Dialog Box ([General] Page)

5.7 Measurement of Execution Time

5.7.1 Measuring Execution Time to a Break or Termination

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 table 5.14 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.

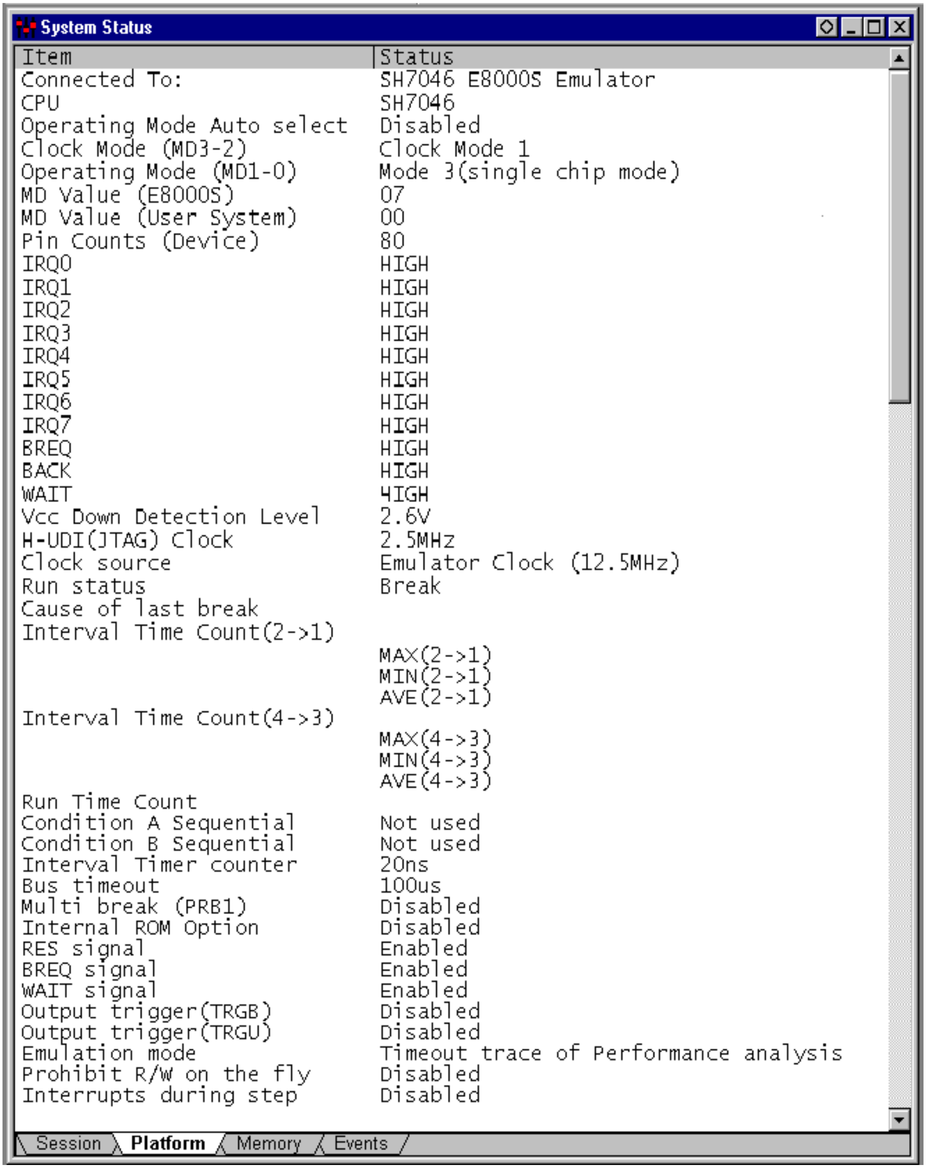


Figure 5.67 [System Status] Window (Display of Execution Time)

In the window, the user program execution time will be displayed as decimal numbers of hours, minutes, etc. The user can use the [The minimum time to be measured by Go command execution] in the [Configuration] dialog box to set a maximum period of measurement. The maximum times are approximately 9999 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). If the user specifies a time other than the available values, the emulator will display * instead of the setting.

5.7.2 Measuring Execution Time between Satisfaction of Specified Conditions

(1) Execution time interval measurement mode 1, 2, 3

Time interval measurement mode 1, 2, 3 uses Break Condition U to measure time intervals in the execution of the user program.

(a) Execution time interval measurement mode 1

Measures the execution time from the satisfaction of Break Condition U2 until the satisfaction of Break Condition U1. After Break Condition U2 has been satisfied, the user program stops on the satisfaction of Break Condition U1. 'BREAK CONDITION SEQUENTIAL U' will be displayed as the cause.

(b) Execution time interval measurement mode 2

Measures the total execution time from the satisfaction of Break Condition U2 until the satisfaction of Break Condition U1. Program execution does not stop after Break Condition U1 and U2 have been satisfied, and total of the execution time between the satisfaction of Break Condition U2 and Break Condition U1 is obtained.

(c) Execution time interval measurement mode 3

Measures the total execution time from the satisfaction of Break Condition U2 until the satisfaction of Break Condition U1, and from the satisfaction of Break Condition U4 until the satisfaction of Break Condition U3. Program execution does not stop after Break Condition U1 and U2 or Break Condition U3 and U4 have been satisfied, and the execution times between the satisfaction of Break Condition U2 and Break Condition U1 and Break Condition U4 and Break Condition U4 are recorded as a total.

The measurement results from time interval measurement mode1, 2, 3 are displayed in Interval Time Count (2->1) and Interval Time Count (4->3) on the [Platform] page of the [System Status] window.

The measurement time is set by [The minimum time to be measured by Go command execution]. The user can select between 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) as the maximum time intervals over which the emulator is to measure performance. The counter for measurement has 24 bits.

The display format is shown below.

```
Interval time Count (2->1)  (a)D'0000H:00M:00S:000000US:000NS(00.0%)E-COUNT=D'00000
                            (b)MAX=D'0000H:00M:00S:000000US:000NS
                            (c)MIN=D'0000H:00M:00S:000000US:000NS
                            (d)AVE=D'0000H:00M:00S:000000US:000NS
Interval Time Count (4->3)  (e)D'0000H:00M:00S:000000US:000NS(00.0%)E-COUNT=D'00000
                            (f)MAX=D'0000H:00M:00S:000000US:000NS
                            (g)MIN=D'0000H:00M:00S:000000US:000NS
                            (h)AVE=D'0000H:00M:00S:000000US:000NS
```

(a)	In time interval measurement mode 1, 2, 3, this is the execution time from the satisfaction of Break Condition U2 to the satisfaction of Break Condition U1. In time interval measurement mode 2, 3, the execution count from the satisfaction of Break Condition U2 to the satisfaction of Break Condition U1 (E-COUNT) is displayed here.
(b)	In time interval measurement mode 2, 3, this is the maximum execution time from the satisfaction of Break Condition U2 to the satisfaction of Break Condition U1.
(c)	In time interval measurement mode 2, 3, this is the minimum execution time from the satisfaction of Break Condition U2 to the satisfaction of Break Condition U1.
(d)	In time interval measurement mode 2, 3, this is the average execution time from the satisfaction of Break Condition U2 to the satisfaction of Break Condition U1.
(e)	In time interval measurement mode 3, this is the execution time and execution count (E-COUNT) from the satisfaction of Break Condition U4 to the satisfaction of Break Condition U3.
(f)	In time interval measurement mode 3, this is the maximum execution time from the satisfaction of Break Condition U4 to the satisfaction of Break Condition U3.
(g)	In time interval measurement mode 3, this is the minimum execution time from the satisfaction of Break Condition U4 to the satisfaction of Break Condition U3.
(h)	In time interval measurement mode 3, this is the average execution time from the satisfaction of Break Condition U4 to the satisfaction of Break Condition U3.

If the execution time exceeds the maximum measurement time, the emulator will display *. When the maximum, minimum, or average times exceed the maximum measuring time, the results will be undefined.

The following items are the limitations of this function.

- Settings of software, software sequential, hardware breaks, and hardware sequential breaks are ignored.
- Trace acquisition conditions will be ignored.
- Trace-halt mode cannot be entered.

- Memory cannot be written to or read from during execution.

(d) Time interval measurement mode 4

This mode measure the execution time over which a specified condition is satisfied, using the conditions of Break Condition B or Trace Condition B.

Measurement starts when a condition set in [Break Condition B1 to B8] or [Trace Condition B1 to B8] dialog boxes are satisfied, and ends when the next condition is satisfied. Each time a condition is satisfied, the starting and stopping of measurement is repeated.

Example:

When conditions are satisfied in order from condition 1 (set by Break Condition 1 or Trace Condition 1), condition 2 (set by Break Condition 2 or Trace Condition 2), condition 1, and condition 2, measurement is done in the following way.

Condition 1 (measurement starts) → Condition 2 (measurement ends) → Condition 1 (measurement starts) → Condition 2 (measurement ends)

In this way, total execution time between conditions 1 and 2 can be measured.

Descriptions on the usage of the function of execution time measurement by specifying conditions are given below.

Consider the conditions for starting and stopping of measurement, then set them in [Break Condition B1 to B8] and [Trace Condition B1 to B8].

Select [Time interval measurement (Condition B)] in [Emulation mode] (on the [General] page of the [Configuration] dialog box).

Execute the user program. The results of measurement will be displayed in [Run Time Count] in the [System Status] window. The maximum measurement time will change with the time set as [The minimum time to be measured by Go command execution] in the [Configuration] dialog box. The maximum times are approximately 9999 hours (with a sampling interval during execution of 52 µs), 488 hours (with a sampling interval during execution of 1.6 µs), and 6 hours (with a sampling interval during execution of 20 ns). If the execution time exceeds the measurement time, the emulator will display *.

When conditions are not set for Break Condition B or Trace Condition B, execution time measurement will not take place when [Time interval measurement (Condition B)] is selected in [Emulation mode] (on the [General] page of the [Configuration] dialog box).

The following limitations apply when this function is used.

- All settings of software, software sequential, and hardware sequential breaks will be ignored.

- Trace acquisition conditions will be ignored.
- Trace-halt mode cannot be entered.
- Memory cannot be written to or read from during execution.

5.8 Performance Analysis Function

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

5.8.1 Measuring with E8000S 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.

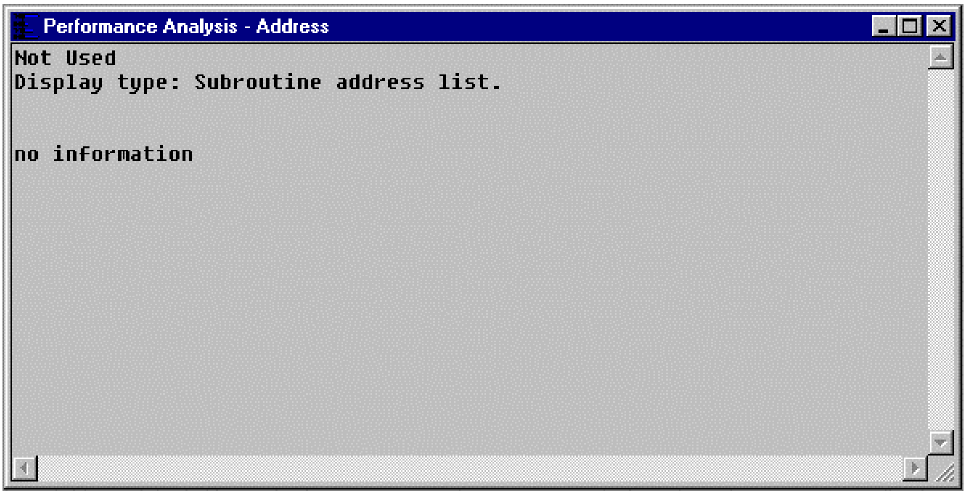


Figure 5.68 [Performance Analysis] Window

In the [Performance Analysis] window, the user can assign any of eight independent channels. To set a condition, click [Edit...] in the pop-up menu of the [Performance Analysis] window to produce the [Performance 1 to 8] dialog box.

Measurement Mode: Measurement modes are explained in table 5.74.

Table 5.74 Measurement Modes

Measurement Mode	Description
Subroutine Time Measurement Mode 1	Measures the execution time and number of passes through the subroutine over the specified address range. Measurement starts with the execution of the subroutine within the address range, is suspended when execution moves outside the specified range, and is restarted when execution returns to the specified address range of the subroutine. The execution count (count of the number of passes) is incremented every time the <end address> of the specified address range is passed. The execution times of instructions at addresses outside the range that are called from within the specified range are not included in the results of measurement. Set the conditions in the [Performance 1 to 8] dialog box. Set the timeout value and maximum number of passes as the conditions for Performance Analysis 1.
Subroutine Time Measurement Mode 2	Measurement is executed in the same way as in mode 1 above, except that the execution times of instructions at addresses outside the range that are called from within the specified range are included in the results of measurement . The emulator will also measure the maximum/minimum execution times over the specified range. Set the timeout value and maximum number of passes as the conditions for Performance Analysis 1.
Subroutine Time Measurement Mode 3	Measurement starts at the <start address range> and halts when it reaches the <end address range>. The execution count is incremented every time the specified <end address range> is passed.
Access Area Count Measurement Mode	Measures the number of times the area set in <call subroutine address range> is accessed by the subroutine set as <start address> and <end address>. The subroutine execution time is measured in subroutine time measurement mode 1.
Subroutine Call Count Measurement Mode	Measures the number of times the subroutine set in <call subroutine address range> is called by the subroutine set as <start address> and <end address>. The subroutine execution time is measured in subroutine time measurement mode 1.

Note: Subroutine execution times are 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 after a branch instruction or the slot after a delayed branch instruction, correct measurement is not possible. In the access area count measurement mode, accesses by the DTC/DMA are not counted.

- Example of subroutine time measurement mode 1

  An example of subroutine time measurement mode 1 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.

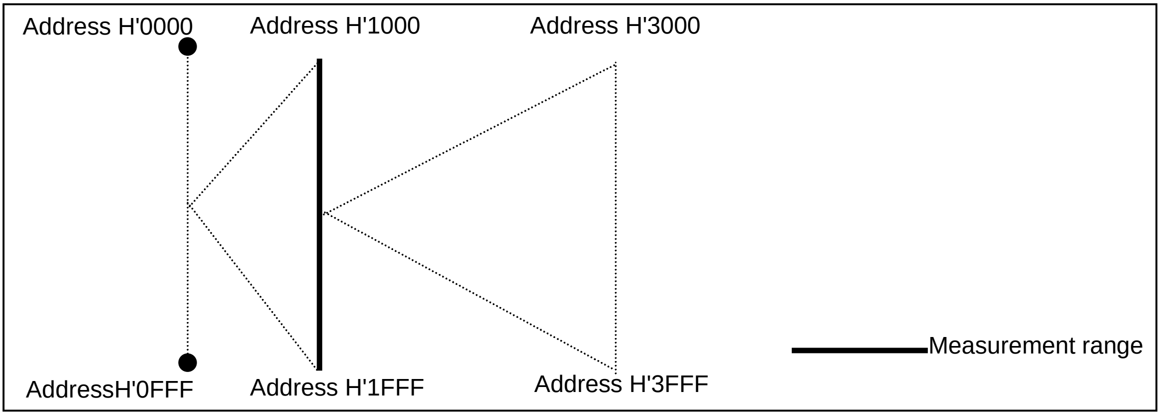


Figure 5.69 Example of Subroutine Time Measurement Mode 1

- Example of subroutine measurement mode 2

  In subroutine measurement mode 2, 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 subroutine time measurement mode 2 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.

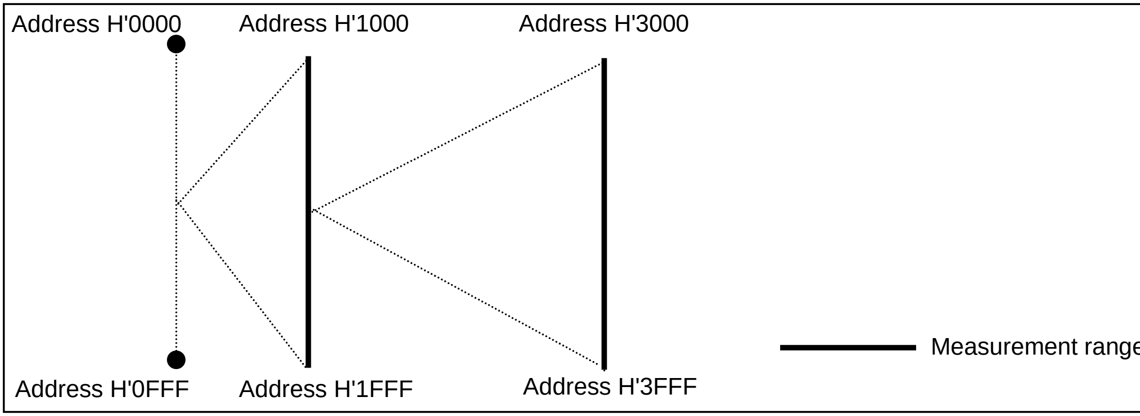


Figure 5.70 Example of Subroutine Time Measurement Mode 2

- Example of subroutine time measurement mode 3

  In subroutine time measurement mode 3, 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 subroutine time measurement mode 3 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 5.71 Example of Subroutine Time Measurement Mode 3

A list of subroutine measurement modes that can be set in the [Performance 1 to 8] dialog box is shown in table 5.75.

Table 5.75 Modes that are Available in the [Performance1 to 8] Dialog Boxes

	Channel Number for Performance Analysis							
Mode	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Subroutine time measurement mode 1	o	o	o	o	o	o	o	o
Subroutine time measurement mode 2	o	o	o	o	o	o	o	o
Subroutine time measurement mode 3*2	o	×	o	×	o	×	o	×
Access area count measurement mode	o	×	o	×	o	×	o	×
Subroutine call count measurement mode	o	×	o	×	o	×	o	×

Note: o: Can be set.
×: Cannot be set

Note: Two channels are necessary to specify the start and end address ranges for subroutine time measurement mode 3. Therefore, settings for subroutine time measurement mode 3 can only be made on odd-numbered channels. In the access area count 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 access area count measurement mode can only be made on odd-numbered channels. In the subroutine call count 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 subroutine call count measurement mode can only be made on odd-numbered channels.

Performance Measurement Time: To set the time over which the emulator to measure the performance, use the [The minimum time to be measured by Go command execution] of the [Execution Mode1] page in the [Configuration] dialog box. 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) as the maximum time interval over which the emulator is to measure performance. The counter for measurement has 24 bits.

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

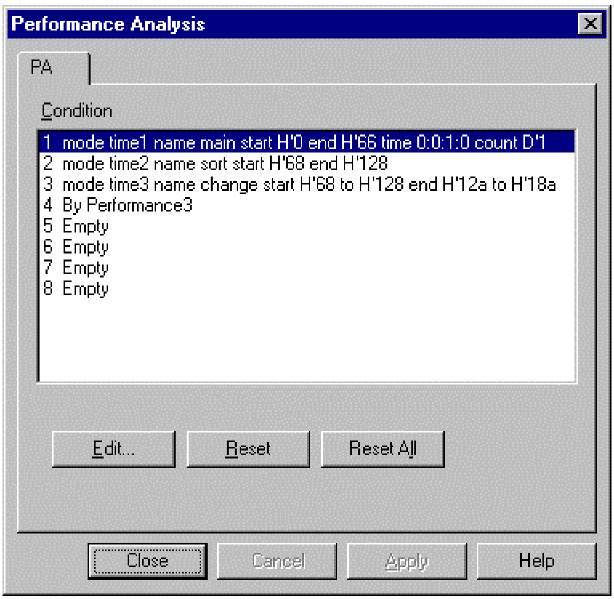


Figure 5.72 [Performance Analysis] Dialog Box

Table 5.76 [Performance Analysis] Dialog Box ([PA] Page)

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. 1 (Performance Analysis 1 setting) 2 (Performance Analysis 2 setting) 3 (Performance Analysis 3 setting) 4 (Performance Analysis 4 setting) 5 (Performance Analysis 5 setting) 6 (Performance Analysis 6 setting) 7 (Performance Analysis 7 setting) 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)
[Reset]	Cancels the Performance Analysis settings that are set in the [Condition] list box.
[Reset All]	Cancels settings of all the subroutines set in the [Condition] list box.

Select 1 from the [Condition] list then click the [Edit…] button. The [Performance 1] dialog box will be displayed. Set the conditions for measuring performance then click the [OK] button.

The individual pages of the [Performance 1 to 8] dialog box are explained in table 5.76.

Table 5.77 Pages of the [Performance 1 to 8] Dialog Box

Page	Description
[General]	Sets the measurement mode.
[Address]	Sets conditions for the address range.
[Time Out]*	Sets a timeout value. When the user program is running in the address range specified by the user, the emulator will compare the measured time with the timeout time specified by the user. When (timeout value setting) < (measured time), the user program emulation will break (the timeout break function). While (timeout value setting) > (measured time), the emulator will measure the user program's execution time.
[Count]*	Sets a maximum number of passes.

Note: These settings are only available in Performance Analysis 1.

Each page is described in detail below.

- [General] page

  Use this page to set the measurement mode.

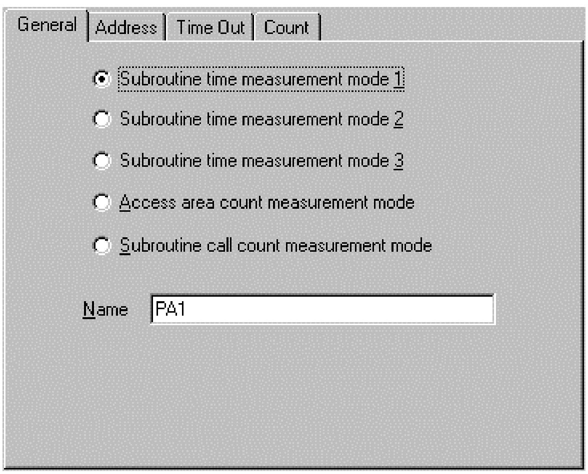


Figure 5.73 [Performance 1] Dialog Box ([General] Page)

Table 5.78 Options on the [General] page

Option	Description
[Subroutine time measurement mode 1]	Selects subroutine measurement mode 1.
[Subroutine time measurement mode 2]	Selects subroutine measurement mode 2.
[Subroutine time measurement mode 3]	Selects subroutine measurement mode 3.
[Access area count measurement mode]	Selects access area count measurement mode.
[Subroutine call count measurement mode]	Selects subroutine call count measurement mode.
[Name]	Sets a name for the address range to be measured.

- [Address] Page

  Set conditions for the address range on this page.

  The format varies according to the subroutine measurement mode selected on the [General] page. The format of the [Address] page is shown below.

  — When [Subroutine time measurement mode 1,2] is selected

  Set the start and end addresses.

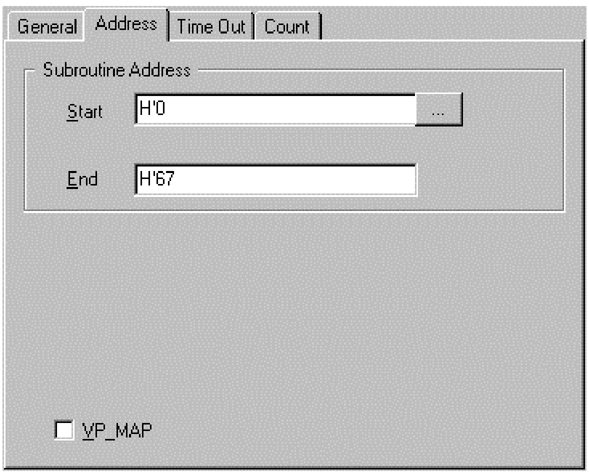


Figure 5.74 [Performance 1] Dialog Box ([Address] Page)

Table 5.79 Options on the [Address] page (When [Subroutine time measurement mode 1,2] is Selected)

Option	Description
[Start]	Sets a start address as a numerical or symbolic value.
[End]	Sets an end address as a numerical or symbolic value.
[...]	Displays the [Input Function Range] dialog box, in which a start address is entered as a numerical or symbolic value. When the setting is made in this dialog box, the corresponding start and end addresses are automatically displayed.

— When [Subroutine time measurement mode 3] is selected

Set the start and end address ranges.

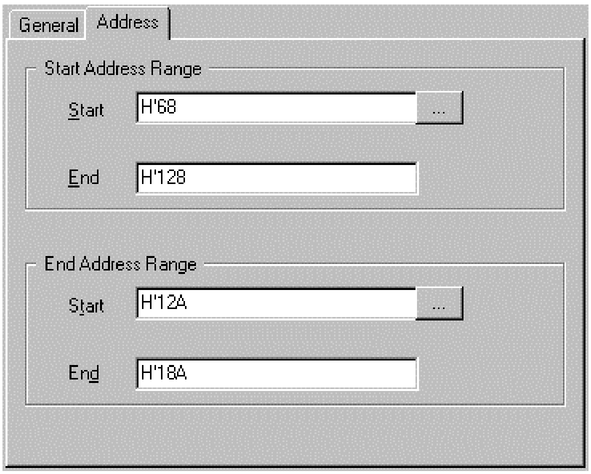


Figure 5.75 [Performance 1] Dialog Box ([Address] Page)

Table 5.80 Options on the [Address] Page (When [Subroutine time measurement mode 3] is Selected)

[Start Address Range]

Option	Description
[Start]	Sets the start address in the start-address range as a numerical or symbolic value.
[End]	Sets the end address in the start-address range as a numerical or symbolic value.
[...]	Displays the [Input Function Range] dialog box to which a start address in the start address range is input as a numerical or symbolic value. When the setting is made in this dialog box, the start address and end address in the start address range are automatically displayed.

[End Address Range]

Option	Description
[Start]	Sets the start address in the end-address range as a numerical or symbolic value.
[End]	Sets the end address in of the end-address range as a numerical or symbolic value.
[...]	Displays the [Input Function Range] dialog box to which a start address in the end address range is input as a numerical or symbolic value. When the setting is made in this dialog box, the start address and end address in the end address range are automatically displayed.

— When [Access area count measurement mode] is selected

Set the start and end addresses and the address range accessed by the subroutine.

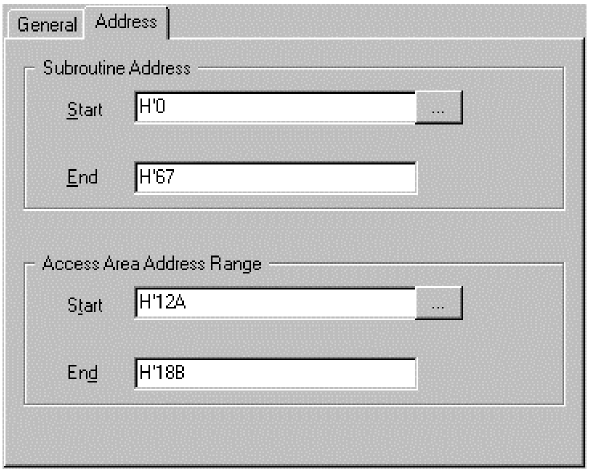


Figure 5.76 [Performance 1] Dialog Box ([Address] Page)

Table 5.81 Options on the [Address] Page (When [Access area count measurement mode] is Selected)

[Subroutine Address]

Option	Description
[Start]	Sets the start address of a subroutine as a numerical or symbolic value.
[End]	Sets the end address of a subroutine as a numerical or symbolic value.
[...]	Displays the [Input Function Range] dialog box in which the start address of a subroutine is input as a numerical or symbolic value. When a setting is made in this dialog box, the start and end addresses of the subroutine are automatically displayed.

[Access Area Address Range]

Option	Description
[Start]	Sets the start address of the address range for access, as a numerical or symbolic value.
[End]	Sets the end address of an address range for access, as a numerical or symbolic value.
[...]	Displays the [Input Function Range] dialog box in which the start address of the address range is input as a numerical or symbolic value. When the setting is made in this dialog box, the start and end addresses of the address range are automatically displayed.

— When [Subroutine call count measurement mode] is selected

Set the start and end addresses and the called subroutine's address range.

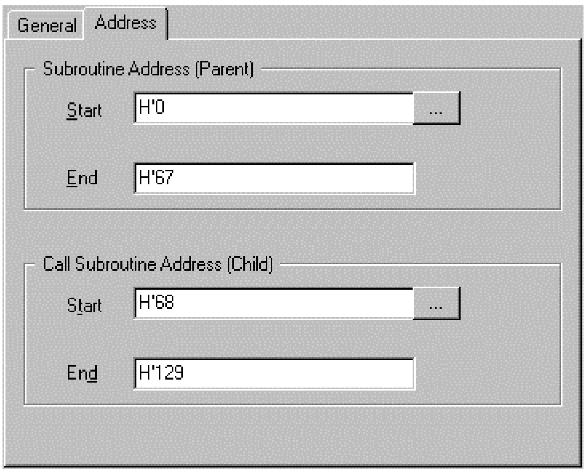


Figure 5.77 [Performance 1] Dialog Box ([Address] Page)

Table 5.82 Options on the [Address] Page (When [Subroutine call count measurement mode] is Selected)

[Subroutine Address (Parent)]

Option	Description
[Start]	Sets the start address of a subroutine as a numerical or symbolic value.
[End]	Sets the end address of a subroutine as a numerical or symbolic value.
[...]	Displays the [Input Function Range] dialog box in which the start address of a subroutine is input as a numerical or symbolic value. When a setting is made in this dialog box, the start and end addresses of the subroutine are automatically displayed.

[Call Subroutine Address (Child)]

Option	Description
[Start]	Sets the start address of the called subroutine's address range as a numerical or symbolic value.
[End]	Sets the end address of the called subroutine's address range as a numerical or symbolic value.
[...]	Displays the [Input Function Range] dialog box in which the start address of the called subroutine's address range is input as a numerical or symbolic value. When the setting is made in this dialog box, the start and end addresses of the end address range are automatically displayed.

- [Time Out] page

  Use this page to set the timeout value. This dialog box is only displayed to allow setting of the conditions for Performance Analysis 1.

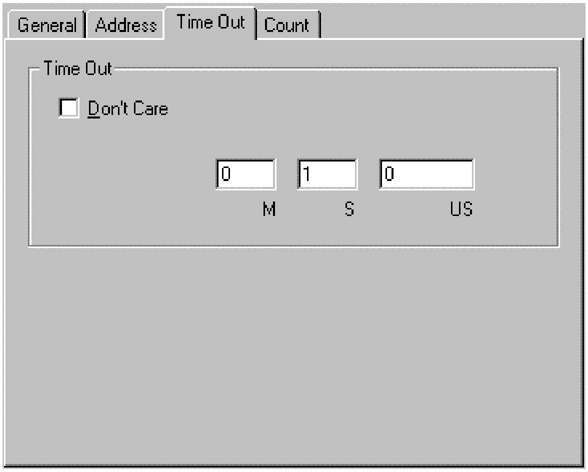


Figure 5.78 [Performance 1] Dialog Box ([Time Out] Page)

Table 5.83 Options of the [Time Out] Page

Option	Description
[Don't Care]	Selects no timeout value.
Input area	Sets a timeout value as a numerical value. When this is omitted, the setting is 0. [M]: minutes, in the range from 0 to 59* [S]: seconds, in the range from 0 to 59 [US]: microseconds, in the range from 0 to 999999

Note: The maximum measurement time is approximately 14 minutes. If the value set is greater than 14 minutes it will be ignored.
When a time-out occurs while the display of the [Performance Analysis] window is being updated, the message `RUN-TIME OVERFLOW` will be displayed.

- [Count] page

  Set the value for the maximum number of passes. This dialog box is only displayed to allow setting of the conditions for Performance Analysis 1.

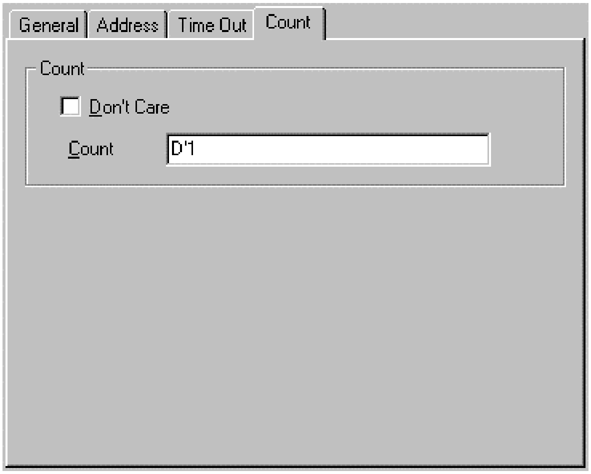


Figure 5.79 [Performance 1] Dialog Box ([Count] Page)

Table 5.84 Options on the [Count] Page

Option	Description
[Don't Care]	Selects no maximum number of passes.
Input area	Sets a maximum number of passes as a numerical value. A break occurs when the conditions set in the [Performance1] dialog box are satisfied the specified number of times. The default value is D'1. Any value from D'1 to D'65,535 can be set here.

Note: This condition is detected as an overflow. The test takes place when the user program passes through the end address. Therefore, the execution time and execution count displayed after break due to this setting will represent the number of specified passes plus one. When a time-out occurs while the display of the [Performance Analysis] window is being updated, the message `RUN-TIME OVERFLOW` will be displayed.

Entering a function name in [Subroutine Address] of the [Input Function Range] dialog box will automatically set the address range of that function in the area for the user input of addresses to the [Address] page.

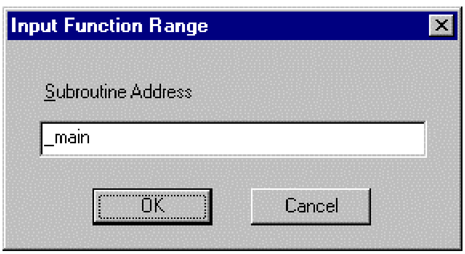


Figure 5.80 [Input Function Range] Dialog Box

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 5.85.

Table 5.85 Menu Items for Displaying the Results

Menu Item	Description
[Address]	Displays the list of executed addresses (default).
[Count]	Displays the number and period of execution.
[Graph]	Displays the proportions of execution times to the whole period of execution on a graph.

[Address]: Select [Address] to produce a list of addresses as shown in figure 5.81.

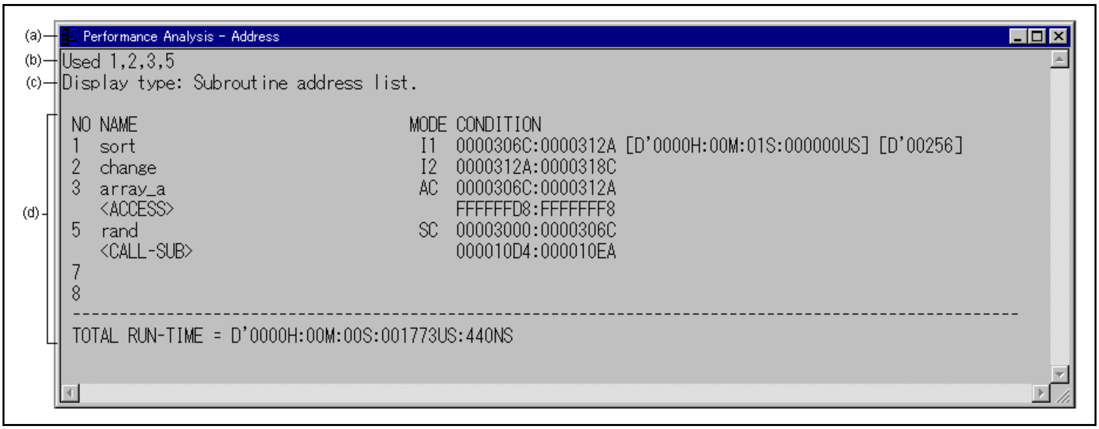


Figure 5.81 [Performance Analysis] Window (Measurement of Executed Addresses)

(a) Window name and display format

(b) Channel numbers of [Performance Analysis] that were used

(c) Display format

(d) Results

Table 5.86 Display Format of Results When [Address] is Selected

Item	Contents
[NO]	Channel number.
[NAME]	Subroutine name.
[MODE]	Subroutine measurement mode. I1: Subroutine time measurement mode 1 I2: Subroutine time measurement mode 2 I3: Subroutine time measurement mode 3 AC: Access area count measurement mode SC: Subroutine call count measurement mode
[ADDRESS]	Subroutine start address and end address (for I1, I2). Timeout value and count-setting value (only when the respective conditions are set in I1 or I2). Start-address range and end-address range (for I3). Access area address range. Call subroutine address range (only for SC).
[TOTAL RUN-TIME]	Total period of execution (H: hours, M: minutes, S: seconds, US: microseconds, NS: nanoseconds).

[Count]: Select [Count] to produce the information on the run times and number of passes in the user program in numerical form shown in figure 5.82.

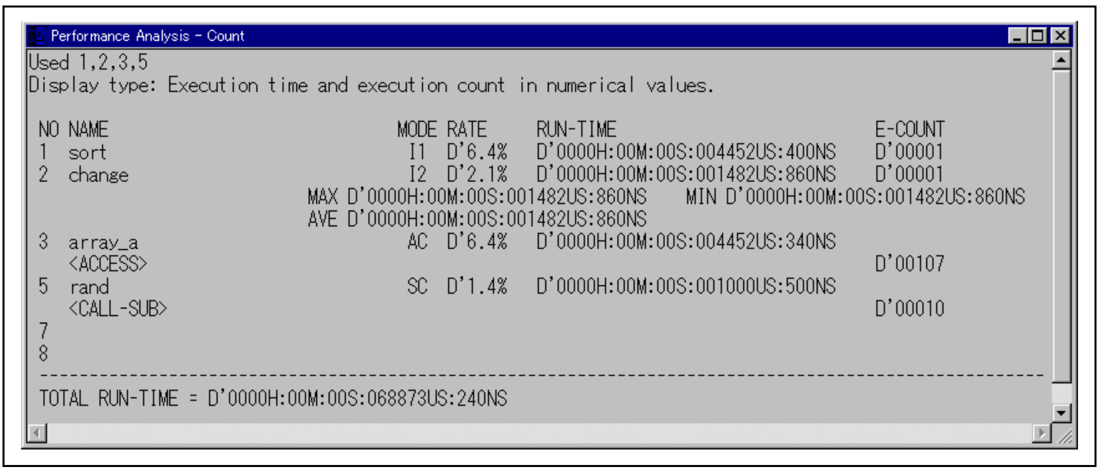


Figure 5.82 [Performance Analysis] Window (Run Time and Execution Count)

Table 5.87 Display Format of Results When [Count] is Selected

Item	Contents
[NO]	Channel number.
[NAME]	Subroutine name.
[MODE]	Subroutine measurement mode. I1: Subroutine time measurement mode 1. I2: Subroutine time measurement mode 2. I3: Subroutine time measurement mode 3. AC: Access area count measurement mode SC: Subroutine call count measurement mode
[RATE]	Proportion of total execution time (numerical value).
[RUN-TIME]	Execution time.
[E-COUNT]	Number of calls of this subroutine.
[TOTAL RUN-TIME]	Total execution time (H: hours, M: minutes, S: seconds, US: microseconds, NS: nanoseconds).

[Graph]: Select [Graph] to produce a display of the proportion of total execution time of the user program, as shown in figure 5.83.

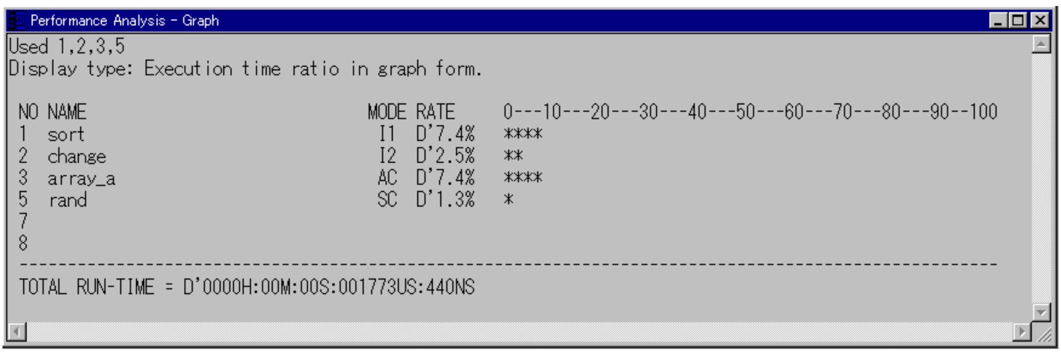


Figure 5.83 [Performance Analysis] Window (Execution Time Ratios)

Table 5.88 Display Format of Results When [Graph] is Selected

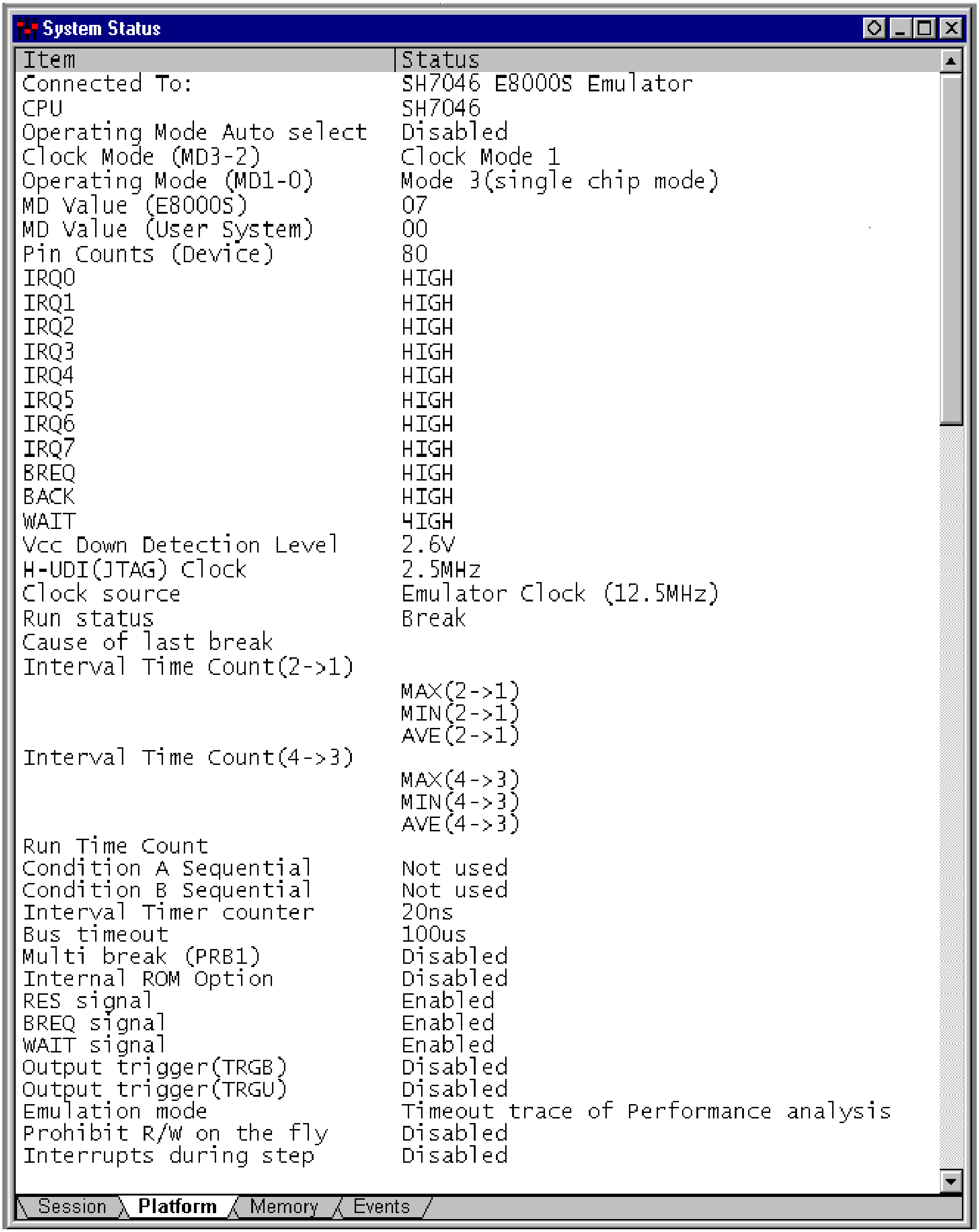
Item	Contents
[NO]	Channel number.
[NAME]	Subroutine name.
[MODE]	Subroutine measurement mode. I1: Subroutine time measurement mode 1. I2: Subroutine time measurement mode 2. I3: Subroutine time measurement mode 3. AC: Access area count measurement mode SC: Subroutine call count measurement mode
[RATE]	Proportions of execution time (as numerical values and on a graph).
[TOTAL RUN-TIME]	Total execution time (H: Hours, M: Minutes, S: Seconds, US: Microseconds, NS: Nanoseconds).

Initializing Performance Measurement Information: To initialize performance measurement information, select [Initialize] 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, click the [OK] button.

Note: When using the [The minimum time to be measured by Go command execution] in the [Configuration] dialog box to modify the unit for counting by the execution-time measurement counter, be sure to reset the settings for performance analysis before altering the minimum time interval. Otherwise, the emulator will continue to use the minimum time interval from before the alteration in measuring the program's performance.

5.9 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:	SH7046 E8000S Emulator
CPU	SH7046
Operating Mode Auto select	Disabled
Clock Mode (MD3-2)	Clock Mode 1
Operating Mode (MD1-0)	Mode 3(single chip mode)
MD Value (E8000S)	07
MD Value (User System)	00
Pin Counts (Device)	80
IRQ0	HIGH
IRQ1	HIGH
IRQ2	HIGH
IRQ3	HIGH
IRQ4	HIGH
IRQ5	HIGH
IRQ6	HIGH
IRQ7	HIGH
BREQ	HIGH
BACK	HIGH
WAIT	HIGH
Vcc Down Detection Level	2.6V
H-UDI(JTAG) Clock	2.5MHz
Clock source	Emulator Clock (12.5MHz)
Run status	Break
Cause of last break	
Interval Time Count(2->1)	
	MAX(2->1)
	MIN(2->1)
	AVE(2->1)
Interval Time Count(4->3)	
	MAX(4->3)
	MIN(4->3)
	AVE(4->3)
Run Time Count	
Condition A Sequential	Not used
Condition B Sequential	Not used
Interval Timer counter	20ns
Bus timeout	100us
Multi break (PRB1)	Disabled
Internal ROM Option	Disabled
RES signal	Enabled
BREQ signal	Enabled
WAIT signal	Enabled
Output trigger(TRGB)	Disabled
Output trigger(TRGU)	Disabled
Emulation mode	Timeout trace of Performance analysis
Prohibit R/W on the fly	Disabled
Interrupts during step	Disabled

Session | Platform | Memory | Events

Figure 5.84 [System Status] Window

[System Status] window has the four sheets as shown in table 5.89.

Table 5.89 [System Status] Window Configuration

Sheet Name	Description
[Session]	Contains such information on the current session as the 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 5.90 [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 5.91 [Platform] Sheet Configuration

Item	Description
[Connected To:]	The name of an emulator that is connected.
[CPU]	The type of target MCU.
[Operating Mode Auto Select]	Whether or not automatic detection of the number of pins on the user system and the inputs on the MD pins was enabled when the system was last initiated. Enable: Enabled, Disable: Disabled[*]
[Clock Mode (MD3-2)]	The current clock mode.[*]
[Operating Mode (MD1-0)]	The current operating mode.[*]
[MD Value (E8000S)]	The values set for the mode pins (E8000S) which are currently set.
[MD Value (User System)]	The values set for the mode pins which are input from the user system.
[Pin Counts (Device)]	The number of pins of the supported MCU set.[*]
[IRQ0] to [H-UDI (JTAG) Clock]	Settings of the [CPU Operating mode] dialog box.
[Clock source]	The clock that is selected.
[Run status]	Whether or not the user program program is being executed. 'Running' is displayed during execution and 'Break' is displayed during stop.
[Cause of last break]	The cause of termination by a break. xxxx:H xxM:xxS:xxxxxxUS:xxxNS (H: Hour, M: Minute, S:Second, US: Microsecond, NS: Nanosecond, x: Number)
[Interval Time Count (2->1)] [MAX (2->1)] [MIN (2->1)] [AVE (2->1)] [Interval Time Count (4->3)] [MAX (4->3)] [MIN (4->3)] [AVE (4->3)]	Results of execution time interval measurement mode 1, 2, 3
[Run Time Count]	The results of measuring the program execution time (results of measuring Go to Break) or of measurement in the execution time interval measurement mode 4.
[Condition A Sequential]	The Condition A sequential points that have been passed.
[Condition B Sequential]	The Condition B sequential points that have been passed.
[Interval Timer counter]	The timer resolution used to measure execution times (setting in the [The minimum time to be measured by Go command execution] combo box of the [Configuration] dialog box).
[Bus timeout]	Timeout detection time (settings in the [Bus timeout] combo box of the [Configuration] dialog box).

Table 5.91 [Platform] Sheet Configuration (cont)

Item	Description
[Multi break (PRB1)]	Multi-break function (selection of the [Enable the multi-break for External probe No.1] check box of the [Configuration] dialog box).
[Internal ROM Option]	Writing to internal ROM area is enabled or disabled (selection of the [Internal ROM Option (Write enable for the internal ROM)] check box of the [Configuration] dialog box).
[RES signal]	Selection of whether or not to permit the input of the RES signal (selection of the [RES signal] check box of the [Configuration] dialog box).
[BREQ signal]	Selection of whether or not to permit the input of the BREQ signal (selection of the [BREQ signal] check box of the [Configuration] dialog box).
[WAIT signal]	Selection of whether or not to permit the input of the WAIT signal (selection of the [WAIT signal] check box of the [Configuration] dialog box).
[Output trigger (TRGB)]	The condition for the output of a pulse from the trigger output pin (the setting in the [TRGB Option] group box of the [Configuration] dialog box).
[Output trigger (TRGU)]	The condition for the output of a pulse from the trigger output pin (the setting in the [TRGU Option] group box of the [Configuration] dialog box).
[Emulation mode]	The emulator's operating mode (the setting in the [Emulation mode] combo box of the [Configuration] dialog box).
[Prohibit R/W on the fly]	Whether or not memory access is inhibited while the user program is being executed (the setting in the [Prohibit R/W on the fly] check box of the [Configuration] dialog box).
[Interrupts during step]	Whether or not a user interrupt is permitted during single-step execution (the setting information in the [Interrupts during step] check box of the [Configuration] dialog box).

Note: The settings may be different from those in the [CPU Operating Mode] dialog box. For details, refer to appendix E.9, CPU Operating Mode.

The [Memory] sheet has the following contents.

Table 5.92 [Memory] Sheet Configuration

Item	Description
[Target Device Configuration]	The memory-map status of the internal 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 5.93 [Events] Sheet Configuration

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

5.10 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 [Execution Mode2] option of the [Configuration] dialog box, a low-level pulse is output for 2 bus cycles from the trigger output pin 10 cycles after bus cycles in which hardware break and hardware break conditions were satisfied during emulation. The trigger signal 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 internal ROM/RAM is accessed, the timing of trigger output will be as shown in figure 5.85.

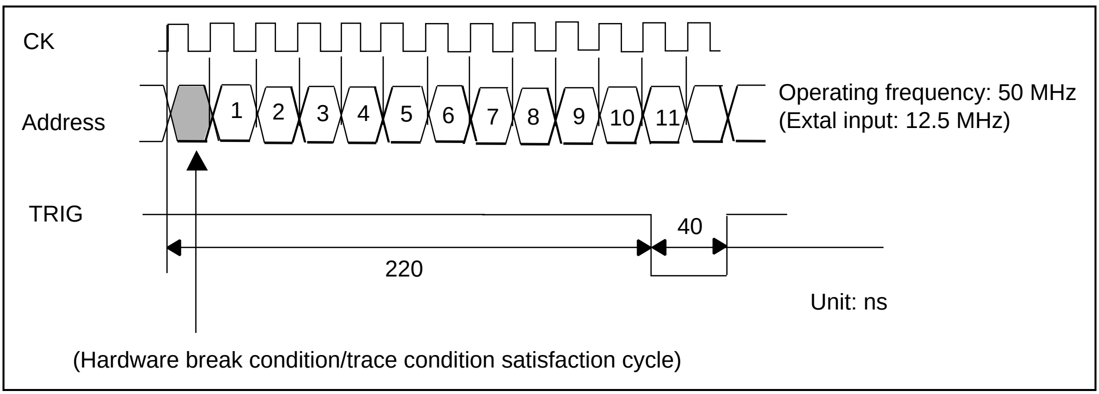


Figure 5.85 Pulse Output Timing

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.

5.11 Memory Areas

5.11.1 The Allocation of Emulation Memory

Use the [Memory Mapping] dialog box to set up this emulation memory. Select [Memory] from [Configure Map...]. The [Memory Mapping] dialog box will be displayed.

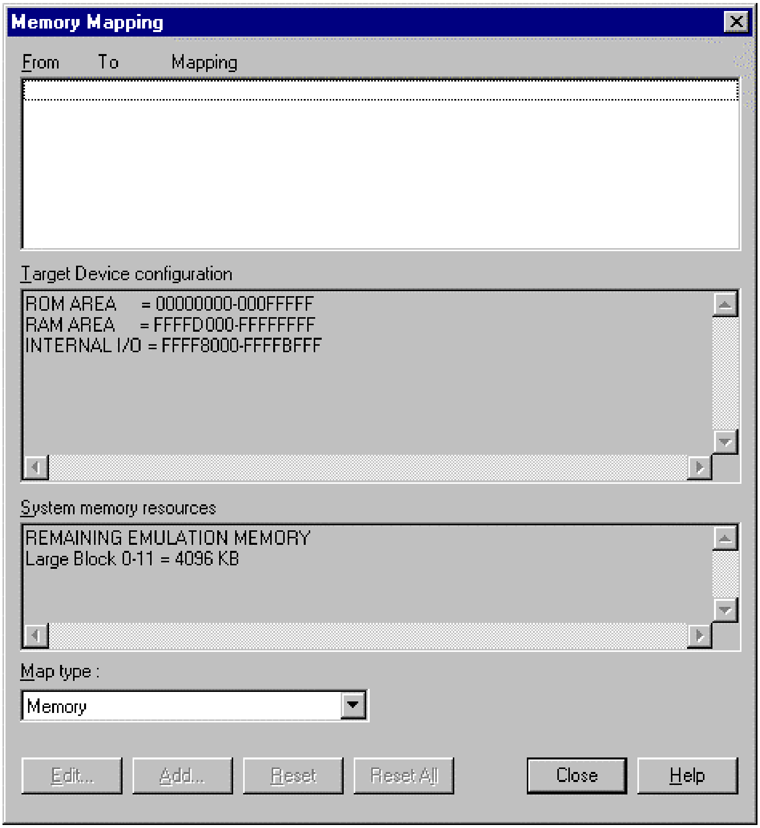


Figure 5.86 [Memory Mapping] Dialog Box

Table 5.94 Configuration Items of the [Memory Mapping] Dialog Box

Option	Description
[From To Mapping]	Displays memory address ranges and memory type settings.
[Target Device configuration]	Displays the memory mapping of the emulation memory block (Short Block 0-15, Large Block 0-11), internal ROM area (ROM AREA), internal RAM area (RAM AREA), and internal I/O area (INTERNAL I/O).
[System memory resources]	Displays the remaining capacities of the emulation memory.
[Map type]	Selects the map type (cannot be specified).
[Edit...]	Allows modification of the memory allocation information selected by [From To Mapping] by displaying the [Edit Memory Mapping] dialog box.
[Add...]	Allows the allocation of new memory by displaying the [Add Memory Mapping] dialog box.
[Reset]	Clears the memory allocation information selected under [From To Mapping].
[Reset All]	Clears all memory allocation information under [From To Mapping].

Since the Emulator operating modes set different emulation memory areas, the [Target Device Configuration] list box will display different memory areas according to the operating mode.

Table 5.95 Items Displayed in the [System Memory resources] List Box

Display	Description
Short Block 0-15 = (address range) [128Kbytes By 1 Block Unit]	Emulation memory can be allocated in 128-kbyte units in the displayed address range
Large Block 0-11 = (address range) [1Mbytes By 1 Block Unit]	Emulation memory can be allocated in 1-Mbyte units in the displayed address range
ROM AREA = (address range)	Internal ROM area
RAM AREA = (address range)	Internal RAM area
INTERNAL I/O = (address range)	Internal I/O area

Notes: 1. Emulation memory cannot be allocated in single-chip mode.

2. Refer to appendix E.2 Emulation Memory, for details on memory mapping.

The [System Memory resources] list box displays the remaining emulation memory in the following format (xxxx, yyyy, and zzzz are displayed in hexadecimal). When no memory is allocated as Short Block, the information below will not be displayed.

REMAINING EMULATION MEMORY

The remaining emulation memory area is displayed. Here, the emulation memory can be allocated in 1-Mbyte units.

Large Block 0-11 = xxxxKB

The remaining emulation memory area is displayed. Here, the emulation memory can be allocated in 128-kbyte units.

Short Block 0-7 = yyyyKB
Short Block 8-15 = zzzzKB

Click the [Close] button to close the dialog box. Click [Add…] button to open the [Add Memory Mapping] dialog box to allocate emulation memory. To modify the allocation of emulation memory, select the memory contents for modification from [From To Mapping] then click the [Edit…] button. The [Edit Memory Mapping] dialog box will be displayed.

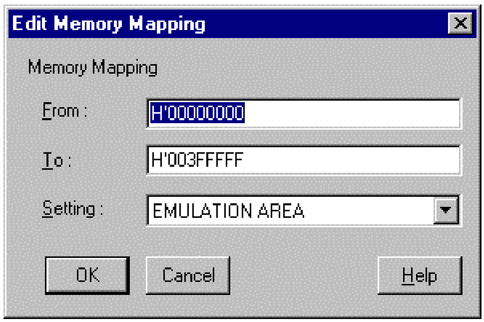


Figure 5.87   [Edit Memory Mapping] Dialog Box

Table 5.96   Configuration Items of the [Edit /Add Memory Mapping] Dialog Box

Option	Description
[From]	Sets the start address of an emulation memory block.
[To]	Sets the end address of an emulation memory block.
[Setting]	Selects the memory type.

When the [OK] button is clicked, the conditions are set and the dialog box is closed.
When the [Cancel] button is clicked, the dialog box is closed and the conditions are not set.
The following shows the memory types that can be set in the [Setting] combo box.

Table 5.97 Settings for Memory Type Available in the [Setting] Combo Box

Memory Type	Description
USER AREA Read-Only	Sets the address range as a write-protected area in the user memory area.
USER AREA Guarded	Sets the address range as a access-prohibited area in the user memory area.
EMULATION AREA	Sets the address range in the emulation memory area.
EMULATION AREA Read Only	Sets the address range as a write-protected area in the emulation memory area.
EMULATION AREA Guarded	Sets the address range as a access-prohibited area in the emulation memory area.

The attribute settings listed above are only for external memory, and cannot be applied to the internal ROM area, internal RAM area, and internal I/O area.

Notes: 1. Do not allocate emulation memory to memory area other than CS.

2. Refer to appendix E.2 Emulation Memory, for details on emulation memory and other limitations.

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.

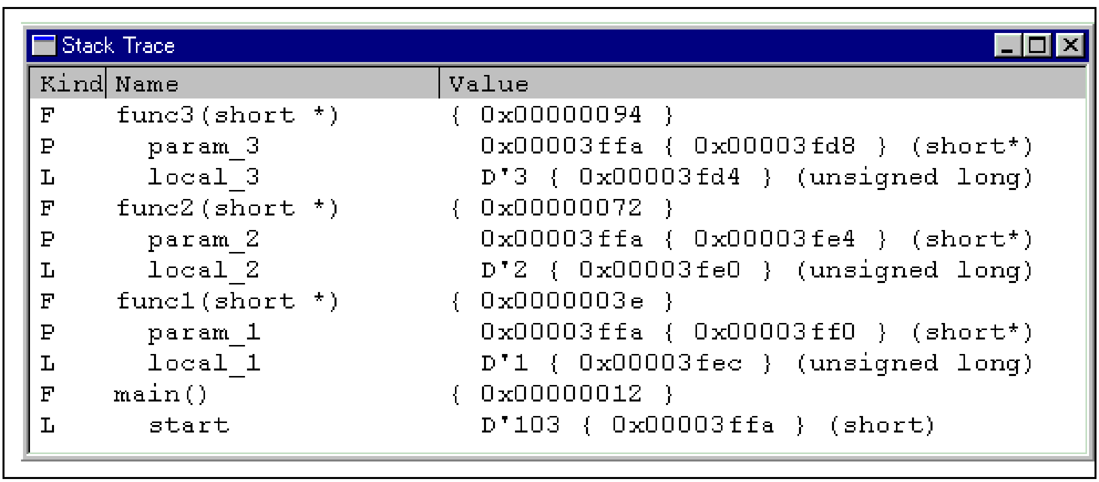


Figure 5.88 [Stack Trace] Window

Table 5.98 [Stack Trace] Window Options

Option	Description
[Kind]	Symbol type 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 5.99 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 5.89.

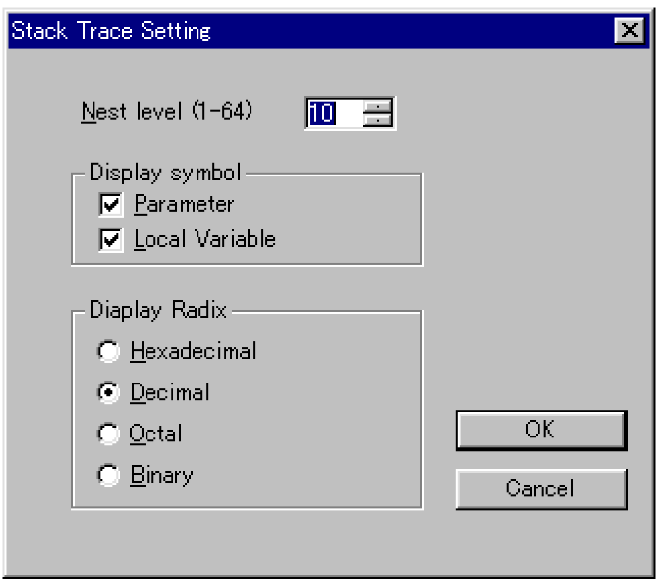


Figure 5.89 [Stack Trace Setting] Dialog Box

Table 5.100 [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 5.101 Access Types for Displaying and Modifying Contents of Memory

Access Type	Description	Stop Time	Display	Modification
Short break	Temporarily suspends the execution of the user program	Large	Enabled	Enabled
Parallel access	Temporarily halts the execution of the user program	Small	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.8, Displaying and Modifying the Contents of Memory.

The respective access types have the following characteristics.

Table 5.102 Characteristics of Displaying and Modifying Contents of Memory

Access Type	Target Window/Command Line	Target Memory Area
Short break	Windows other than the [Auto update Memory] window that display memory contents Commands to display or modify memory contents	Areas other than the target types for parallel access (memory in the user system, in the internal I/O area, and in internal RAM)
Parallel access	Windows other than the [Auto update Memory] window that display memory contents Commands to display or modify memory contents	Internal ROM Emulation memory
Auto update Memory	[Auto update Memory] window No target command	Specified address range

Note: For details on stopped periods and other notes, refer to appendix E.8, Displaying and Modifying the Contents of Memory.

The memory refresh function of the HDI ([Memory]·[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, short breaks will occur if the [I/O Registers] window, or the [Memory] window that displays the contents of internal RAM is opened.

When parallel access is used to display memory while the user program is running, times in debugging will be shortened by the required suspensions of the user program. Take care on this point.

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 5.103 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 in the [Auto-update Memory -Add-] dialog box.

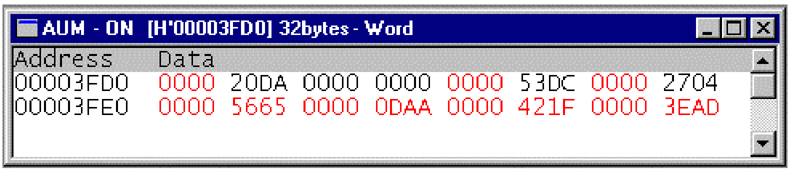


Figure 5.90 [AUM] Window

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.

Acquisition Mode: Auto-update memory has two acquisition modes.

- Realtime parallel monitoring mode

  In this mode, the contents of memory are directly acquired and displayed. Therefore, it does not affect the emulation. Select the memory access bus width through the [AUM – Target Details] dialog box. Memory access bus widths can be selected from 8, 16, or 32 (default) bits. After setting, click the [OK] button.

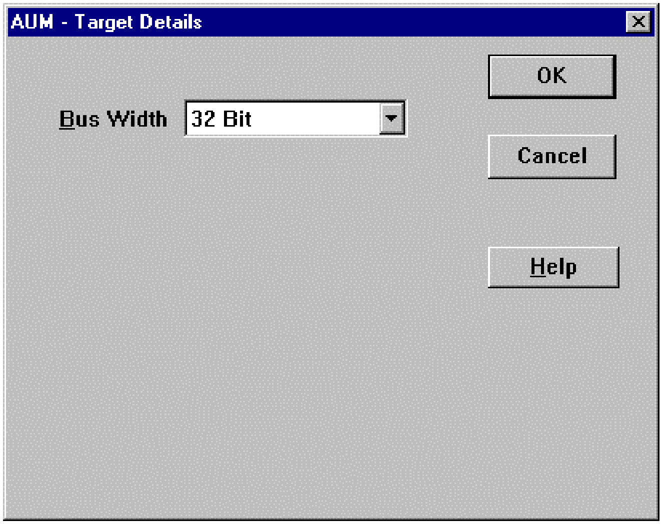


Figure 5.91 [AUM – Target Details] Dialog Box

5.13.4 Displaying the Memory

Display format: The auto-update memory function allows the display of memory contents in ASCII, bit, byte, word, longword, or single-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. 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.

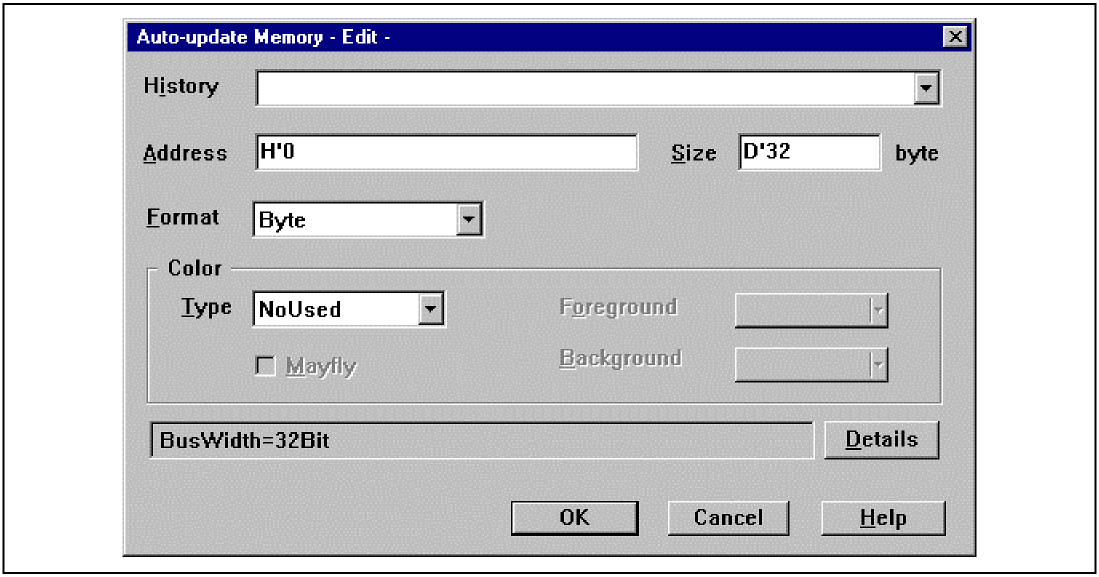


Figure 5.92 [Auto-update Memory -Edit-] Dialog Box

Notes: 1. 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.

2. When [Halt] is selected from the pop-up menu of the [Trace] window, or a trace stop condition has been satisfied, Auto update Memory items cannot be added until [Restart] is selected from the pop-up menu of the [Trace] window.
3. When user program is executed in cycle reset mode and execution-time measurement mode, the auto-update memory function is not available.
4. When the [AUM] window is opened in cycle reset mode or execution-time measurement mode, and a user program is executed while the profile function is valid, the automatic updating of Auto update Memory items will be halted.
5. When a software-break count has been specified, the user program will be halted temporarily. When the auto-update memory display is updated while the user program is halted, the [AUM] window may not be displayed correctly.
6. When a software-break condition has been specified, error message ‘EMULATOR BUSY’ may be displayed.
7. The [Auto Update Memory] window does not support the Double float type. If this is specified, an error will occur.

5.14 Controlling and Checking the State of MCU

The emulator can select the clock to input to the MCU, check the operation, power supply, and clock state of the user system. The functions are useful when debugging the user system hardware.

5.14.1 Selecting Clock for the MCU

This emulator supports three types of clock for the MCU: external clock input from the user system, a crystal oscillator attached on the evaluation chip board, and the emulator external clock (12.5 MHz). For details on selecting clock, refer to
section 6.2.7, CLOCK and to section 3.3.5, Selecting the Clock.

When selecting a clock, refer to the following:

When a clock is selected, the emulator resets the MCU. Note that this initializes the registers.

To select the external clock, the external clock must be input. Otherwise, an error message will be displayed and the emulator internal clock will be selected.

When the emulator system program is started, the emulator inputs a clock to the MCU in the following order:

- When external clock is input from the user system, the user system clock is selected.
- When a crystal oscillator is attached on the evaluation chip board, crystal oscillation clock will be selected.
- Emulator internal clock (12.5 MHz)
  The frequency of the emulator internal clock selected is 12.5 MHz.

5.14.2 Checking the I/O Signals

The emulator checks whether the connection with the user system is correct when the emulator is initiated. Through this check, abnormal operations such as short-circuits of the user system interface signals can be detected.

The same operation as above can be done with the CHECK command. For details on the CHECK command, refer to section 6.2.6, CHECK command.

5.14.3 Checking the Power Supply and Clock State of the User System

The emulator monitors the power supply and the clock state of the user system.

When the MCU clock is selected by setting USER in the CLOCK command, the next operation will be carried out when the user system power is turned off or the clock is stopped, according to the emulator state.

Notes: 1. When the user system power is turned off, the turn off of the user system power will be detected before the clock will stop (the Vcc level used to detect switching off of the power can be set in the [CPU Operating Mode] dialog box).

2. The meaning of clock will stop above is the stop of the clock while the user system power is still turned on.

During User Program Execution:

- When user system power is turned off

  'VCC Down' is displayed in the status bar. When power is turned on emulation will resume and the PC of the currently executing user program will be displayed.
- When clock is stopped

  'User system not ready' is displayed and is linked down. When using the emulator continuously, link up again.

During Break:

- When user system power is turned off and clock is stopped

  'User system not ready' is displayed and is linked down. When using the emulator continuously, link up again.

5.15 Input Format

5.15.1 Entering Masks

Address bus conditions and data bus conditions can be input with masks. Addresses can be masked in bits or in 4-bit units. When a bit is masked, it always satisfies the condition.

To specify a mask for an address condition, specify the mask value in the [Mask] area.

The mask for data conditions is specified in the [Data] area.

There is a separate [Mask] list on the [Data] page of the [Break Condition U1] dialog box. To specify any further mask, specify * for the digits to be ignored. Examples of mask specification is shown below.

Table 5.104 Address Mask Specification

No	Input Value	Mask Unit	Example	Masked Bits
1	Binary	1 bit	B'01101***	Masks bits 0 to 2
2	Hexadecimal	4 bits	H'F50***	Masks bits 0 to 11

5.16 [Source] Window Expanded Function

5.16.1 Setting BP Column

In the HDI, software breakpoints can be set, cancelled, or displayed, and software sequential breakpoints can be displayed or cancelled in the BP column of the [Source] window.

Software breakpoints can be set or cancelled by selecting a program (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 press the F9 key.

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

The contents of the items displayed in the BP columns are shown in table 5.105.

Table 5.105 BP Column Display Item

Displayed Item	Contents
No display	Nothing is set.
Break	A software break is set.
PASS	A pass point for a software sequential break is set.
RESET	A reset point for a software sequential break is set.

Notes: 1. When a software breakpoint is set in the BP column, the satisfaction count is 1.
2. Software sequential breakpoints (PASS or RESET) can be displayed and cancelled in the BP column, but cannot be set in the BP column. To set a software sequential breakpoint, use the [Break] window or the BSS command.

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 and file names. When using strings to mask data, specify H' (hexadecimal) or B' (binary) at the head of the data as the radix, and specify "*" for the digits to be masked.

No	Input Value	Mask Unit	Example	Masked Bits
1	Binary	1 bit	B'01110***	Masks bits 0 to 2
2	Hexadecimal	4 bits	H'000F50**	Masks bits 0 to 7

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 condition.
BREAKCONDITION_CLEAR	6.2.1	BCC	Clears hardware break conditions.
BREAKCONDITION_DISPLAY	6.2.1	BCD	Displays hardware break conditions.
BREAKCONDITION_ENABLE	6.2.1	BCE	Enables or disables hardware break conditions.
BREAKCONDITION_SET	6.2.1	BCS	Sets hardware break conditions.
BREAKCONDITION_U_CLEAR	6.2.2	BCUC	Clears internal breakpoints that have been set.
BREAKCONDITION_U _DISPLAY	6.2.2	BCUD	Displays internal breakpoints that have been set.
BREAKCONDITION_U _ENABLE	6.2.2	BCUE	Enables or disables internal breakpoints that have been set.
BREAKCONDITION_U _SET	6.2.2	BCUS	Sets an internal breakpoint.
BREAKPOINT	6.2.4	BP	Sets software breakpoints.
BREAKPOINT_CLEAR	6.2.4	BC	Clears software breakpoints that have been set.
BREAKPOINT_DISPLAY	6.2.4	BD	Displays software breakpoints that have been set.
BREAKPOINT_ENABLE	6.2.4	BE	Enables or disables software breakpoints that have been set.
BREAKSEQUENCE_CLEAR	6.2.5	BSC	Clears software sequential breakpoints that have been set.
BREAKSEQUENCE_DISPLAY	6.2.5	BSD	Displays software sequential breakpoints that have been set.
BREAKSEQUENCE_ENABLE	6.2.5	BSE	Enables or disables software sequential breakpoints that have been set.
BREAKSEQUENCE_SET	6.2.5	BSS	Sets software sequential breakpoints.
CHECK	6.2.6	—	Checks the state of each pin for the MCU.
CLOCK	6.2.7	CK	Sets and displays the CLOCK signal for the MCU.
CONDITION_SEQUENCE	6.2.3	CSQ	Sets hardware sequential break.

Table 6.1 List of Commands (cont)

Command	Section	Abbreviation	Description
COVERAGE_DISPLAY	6.2.8	CVD	Displays coverage information.
COVERAGE_CLEAR	6.2.8	CVC	Initializes coverage information.
COVERAGE_SET	6.2.8	CVS	Sets coverage information and modifies display format.
DEVICE_TYPE	6.2.9	DE	Displays the type of a currently selected MCU.
DISASSEMBLE	—	DA	Disassembles user program and displays the result.
END	6.2.10	END	Returns to user program execution when the emulator enters the trace halt state with trace conditions satisfied.
ERASE	—	ER	Clears the contents of the Command Line window.
EVALUATE	—	EV	Calculates expression.
EXECUTION_MODE	6.2.11	EM	Sets and displays debugging conditions for user program execution.
FILE_LOAD	—	FL	Loads 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.
GO	—	GO	Executes user program.
GO_OPTION	6.2.12	GP	Sets and displays the emulation mode for user program execution.
GO_RESET	—	GR	Executes user program from reset vector.
GO_TILL	—	GT	Executes user program up to temporary breakpoint.
HALT	—	HA	Halts user program.
HELP	—	HE	Displays help for command line and commands.
ID	6.2.13	ID	Displays emulator type and version number.
INITIALISE	—	IN	Initializes a platform.

Table 6.1 List of Commands (cont)

Command	Section	Abbreviation	Description
LOG	—	LO	Manipulates logging file.
MAP_DISPLAY	—	MA	Displays memory map information.
MAP_SET	6.2.14	MS	Sets emulator memory-map.
MEMORY_DISPLAY	—	MD	Displays memory contents.
MEMORY_EDIT	—	ME	Modifies memory contents.
MEMORY_FILL	—	MF	Fills the memory with the specified data.
MEMORY_MOVE	—	MV	Moves memory block.
MEMORY_TEST	—	MT	Tests memory block.
QUIT	—	QU	Terminates HDI.
PERFORMANCE_ANALYSIS	6.2.15	PA	Displays measurements of performance by emulator.
PERFORMANCE_CLEAR	6.2.15	PC	Clears performance conditions that have been set for the emulator.
PERFORMANCE_SET	6.2.15	PS	Sets performance conditions.
RADIX	—	RA	Sets input radix.
REFRESH	6.2.16	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 MCU.
SLEEP	—	—	Delays command execution.
STATUS	6.2.17	STS	Displays emulator state information.
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.

Table 6.1 List of Commands (cont)

Command	Section	Abbreviation	Description
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.
TRACE_DISPLAY	6.2.19	TD	Displays the acquired trace information.
TRACE_MODE	6.2.20	TM	Sets and displays the trace mode.
TRACE_SEARCH	6.2.21	TS	Searches for trace information.
TRACEACQUISITION_CLEAR	6.2.18	TAC	Clears trace conditions that have been set.
TRACEACQUISITION_ DISPLAY	6.2.18	TAD	Displays trace conditions that have been set.
TRACEACQUISITION_SET	6.2.18	TAS	Sets conditions for acquiring trace information.

6.2.1 Hardware Break Commands (BCS, BCC, BCD, BCE)

- Setting BREAKCONDITION_SET (BCS)
- Cancellation BREAKCONDITION_CLEAR (BCC)
- Display BREAKCONDITION_DISPLAY (BCD)
- Enable or Disable
  BREAKCONDITION_ENABLE (BCE)

[Command syntax and parameters]

- Setting **bcs** <type> **channel** <channel_number> <option> [<option>...]
  <option> = <addropt> | <dataopt> | <r/wopt> | <accessopt> | <prbopt> |
  <nmiopt> | <irqopt> | <countopt> | <delayopt>
- Cancellation **bcc** <type> [**channel** <channel_number>]
- Display **bcd** <type> [**channel** <channel_number>]
- Enable or Disable
  bce <type> [**channel** <channel_number>] <mode>

Parameter	Type	Description
<type>	Keyword	Sets a hardware break condition type. **a/b/c**: Break Condition A/B/C
<channel_number>	Numeric or string	Sets 1 to 8, **p**, or **r**. When setting a break condition channel number, specify 1, 2, 3, 4, 5, 6, 7, or 8. When setting a hardware sequential break, specify **p** or **r**. **p**: Sequential point **r**: Reset point
<mode>	Keyword	Sets whether to enable or disable break conditions. **enable**: Enables break conditions. **disable**: Disables break conditions.

Description of the bcs command <option> (Specify one or more conditions.)

Parameter	Description
<addropt>	Specifies address condition. To specify an address break condition: **address** <address> [**not**] To specify an address range break condition: **address** <address1> **to** <address2> [**not**] To mask address: **address mask** <maskdata> [**not**] <address>: Address (numeric) <address1>: Start address (numeric) <address2>: End address (numeric) <maskdata>: Mask value Add **not** to specify an address or range outside which the user program should break. However, do not add **not** when specifying **c** as <type>.

Parameter	Description
<dataopt>	Specifies data condition. This parameter can be used when <type> is **a** or **b**. To specify an 8-bit data break condition: **data** <data> **byte** <position> [**not**] To specify a 16-bit data break condition: **data** <data> **word** <position> [**not**] To specify a 32-bit data break condition: **data** <data> **long** [**not**] <data>: Specify data value (numeric) <position>: Specify valid position of the data bus value **high**: upper word **low**: lower word **byte3**: upper byte of the upper word **byte2**: lower byte of the upper word **byte1**: upper byte of the lower word **byte0**: lower byte of the lower word When **byte** is set for <datawidth>, **byte3**, **byte2**, **byte1**, or **byte0** is valid. When **word** is set for <datawidth>, **high** or **low** is valid. When **long** is set, <position> cannot be selected. Mask specification is as follows: To specify an 8-bit data break condition (mask specification): **data mask** <maskdata> **byte** <position> [**not**] To specify a 16-bit data break condition (mask specification): **data mask** <maskdata> **word** <position> [**not**] To specify a 32-bit data break condition (mask specification): **data mask** <maskdata> **long** [**not**] <maskdata>: Specifies the mask data. Add **not** to the specification to break the user program when the data bus holds a value other than the specified value.

<table>
<tr><th>Parameter</th><th>Description</th></tr>
<tr><td><r/wopt></td><td>Specifies read or write condition. (This parameter can be used when <type> is a or b.)

To break the user program with the read cycle: direction read

To break the user program with the write cycle: direction write</td></tr>
<tr><td><accessopt></td><td>Specifies bus status condition.

To break the user program with the data access cycle: access dat

To break the user program with the DMA/DTC cycle: access dma</td></tr>
<tr><td><prbopt></td><td>Specifies external probe signal condition. (This parameter can be used when <type> is a or b.)

To specify external probe signal condition: prb <bit specification>

Specify each bit as follows:

PRB1 to PRB4 signal bit specification

3 2 1 0 : Bit location
x x x x : Value to be specified (Specify 0 (low level) or 1 (high level)
| | | | for x.)
4 3 2 1 : PRB number

When * is specified, the condition is specified as not including the state of the external probe signal at that bit location.</td></tr>
<tr><td><nmiopt></td><td>Specifies NMI signal condition (This parameter can be used when <type> is a or b.)

To break the user program when the NMI signal is high: nmi hi

To break the user program when the NMI signal is low: nmi low</td></tr>
<tr><td><irqopt></td><td>Specifies IRQ0 to IRQ7 signal conditions. (This parameter can be used when <type> is a or b.)

To specify IRQ0 to IRQ7 signal conditions: irq <bit specification>

Specify each bit as follows:

To specify bits IRQ0 to IRQ7

7 6 5 4 3 2 1 0 : Bit location
x x x x x x x x : Value to be specified (Specify 0 (low level) or
| | | | | | | | 1 (high level) for x.)
7 6 5 4 3 2 1 0: IRQ number

When * is specified, the condition can be specified not to include the IRQ condition for bit location.</td></tr>
</table>

Parameter	Description
<countopt>	Specifies the pass count that satisfies the break condition. (This parameter can be used when <type> is **a** or **b**.) To specify pass count: **count** <value> Any value from 1 to H'FFFF can be specified as <value>.
<delayopt>	Specifies the number of bus cycles to be executed after the break condition is satisfied. (This parameter can be used when <type> is **a** or **b** and <channel_number> is 7.) To specify the number of bus cycles to be executed: **delay** <value> Any value from 1 to H'7FFF can be specified as <value>.

[Description]

- Setting

  This command specifies hardware break conditions (Break Condition A/B/C). The emulator stops program execution when the specified conditions are satisfied. For details on conditions, refer to the description of <option> in the BCS command. When the user specifies a hardware sequential break, sequential points and reset points can be set.

- Cancellation

  This command clears hardware break conditions (Break Condition A/B/C), sequential points, and reset points (Break Condition A/B).

  When the user does not specify a channel number, all of the specified hardware break conditions are cleared. If the user specifies a hardware sequential break, the user cannot specify a channel number.

- Display

  This command displays break conditions (Break Condition A/B/C) that have been set. When the user does not specify a channel number, the emulator displays all of the specified hardware break conditions.

      Break Condition Xn: <Enable/Disable> < Contents specified>

      X: A/B/C

      n: Number (from 1 to 8)

  The following will be shown when a hardware sequential break is specified.

      Break Condition X Sequential:<Enable/Disable>

      n <Contents specified>

      X: A/B

      n: Sequential number (start from 1 and condition is satisfied at maximum of 7.
      R indicates a reset point.)

- Enable or Disable

  This command enables or disables break conditions (Break Condition A/B/C), sequential points, and reset points (Break Condition A/B) that have been set. When the user does not specify the channel number, the emulator enables or disables all of the specified break conditions. When the user specifies a hardware sequential break, the user cannot specify a channel number.

[Examples]

1. To set the following conditions for channel 2 of Break Condition A:
   Address condition: An address bus value from H'10027C to H'100304,
   Read/write cycle condition: Write cycle only.

   BCS A CHANNEL 2 ADDRESS H'10027C TO H'100304 DIRECTION WRITE

2. To set the following conditions for channel 4 of Break Condition B:
Data condition: Data bus value of H'4750,
Read/write cycle condition: Read cycle only.

BCS B CHANNEL 4 DATA MASK H'*4750 DIRECTION READ***

3. To set the following conditions for channel 5 of Break Condition A:
Address condition: An address bus value other than H'10027C,
External probe condition: PROBE4 = low, PROBE3 = high, PROBE2 = low, PROBE1 = high.

BCS A CHANNEL 5 ADDRESS H'10027C NOT PRB 0101

4. To set the following conditions for channel 7 of Break Condition B:
NMI signal condition: NMI = low.

BCS B CHANNEL 7 NMI LOW

5. To set the following conditions for channel 1 of Break Condition B:
Address condition: Mask specification at address bus value = H'1000***.

BCS B CHANNEL 1 ADDRESS MASK H'1000*

6. To set the following conditions for channel 6 of Break Condition B:
Address bus value: Except the range from H'100000 to H'1001C0.

BCS B CHANNEL 6 ADDRESS H'100000 TO H'1001C0 NOT

7. To display the condition for channel 6 of Break Condition B:

BCD B CHANNEL 6

The display format is as follows:

```
> BCD B CHANNEL 6
Break Condition B6:Enable address H'100000 to H'1001c0 not
```

8. To display all of the conditions set to Break Condition A:

BCD A

The display format is as follows:

```
>BCD A
Break Condition A1:Disable
Break Condition A2:Enable address H'10027c to H'100304 direction write
Break Condition A3:Disable
Break Condition A4:Disable
Break Condition A5:Enable address H'10027c not prb 0101
Break Condition A6:Disable
Break Condition A7:Disable
Break Condition A8:Disable
```

9. To enable the conditions set to channel 3 of Break Condition B:

BCE B CHANNEL 3 ENABLE

10. To clear all of the conditions set to Break Condition A.

BCE A DISABLE

11. To clear the condition set to channel 3 of Break Condition B.

    BCC B CHANNEL 3

12. To clear all of the conditions set to Break Condition A.

    BCC A

13. To set a sequential point when a hardware sequential break is set to Break Condition A.
    Address bus value: H'100000

    BCS A CHANNEL P ADDRESS H'100000

14. To set a reset point when a hardware sequential break is set to Break Condition A.
    Address bus value: H'300000

    BCS A CHANNEL R ADDRESS H'300000

15. To display the display format when a hardware sequential break is set to Break Condition A.

    BCD A

    The display format is as follows:

    ```
    >BCD A
    Break Condition A Sequential:Enable
    1 address H'100000
    2 address H'200000
    R address H'300000
    ```

16. To disable the display format when a hardware sequential break is set to Break Condition A.

    BCE A DISABLE

17. To clear the display format when a hardware sequential break is set to Break Condition A.

    BCC A

[Notes]

- When a Break Condition is satisfied, emulation may stop after two or more instructions have been executed.
- It is not possible to use numbers to specify the points of a hardware sequential break. It sets the condition in the order opposite to the sequence of conditions. To set the sequential conditions again, clear all of the conditions.

6.2.2 Internal Break Commands (BCUS, BCUC, BCUD, BCUE)

- Setting BREAKCONDITION_U_SET (BCUS)
- Cancellation BREAKCONDITION_U_CLEAR (BCUC)
- Display BREAKCONDITION_U_DISPLAY (BCUD)
- Enable or Disable BREAKCONDITION_U_ENABLE (BCUE)

[Command syntax and parameters]

- Setting **bcus channel** <channel_number> <option> [<option>…] <option> = <addropt> | <dataopt> | <r/wopt> | <accessopt> | <countopt>
  To specify a reset point: **bcus reset** <reset>
- Cancellation **bcuc** [**channel** <channel_number>]
  To specify a reset point: **bcuc reset**
- Display **bcud** [**channel** <channel_number>]
  To specify a reset point: **bcud reset**
- Enable or Disable
  bcue [**channel** <channel_number>] <mode>
  To specify a reset point: **bcue reset** <mode>

Parameter	Type	Description
<channel_number>	Numeric	Sets an internal break (Break Condition U) channel number from 1 to 4. For BCUS commands, items that can be specified as <option> depend on the channel number. For the <option> parameter, see the description of the BCUS command <option> parameters. 1: <addropt>, <dataopt>, <r/wopt>, <accessopt>, or <countopt> 2 to 4: <addropt>, <dataopt>, <r/wopt>, or <accessopt>
<mode>	Keyword	Enables or disables internal break. **enable**: Enabled **disable**: Disabled
<reset>	Keyword	Specifies a reset point of the internal sequential break. To specify a reset point (only valid when **reset** is specified with the BCUS command): **address** <address>

Description of the bcus command <option> (Specify one or more conditions.)

Parameter	Description
<addropt>	Specifies an address condition. To specify an address as a break condition: **address** <address> To specify the address as prefetched and to generate a break before the corresponding instruction is executed: **address** <address> **pc** To specify the address as prefetched and to generate a break after the corresponding instruction is executed: **address** <address> **pcafter** To specify the address range: **address** <address> **to** <address> <address>: Address (numeric) When masking the address value is specified, the access condition for the address bus can be specified. **address mask** <maskdata> **address mask** <maskdata> **pc** **address mask** <maskdata> **pcafter** <maskdata>: Specifies the mask data.
<dataopt>	Specifies a data condition. To break on a specified 8-bit value: **data** <data> **byte** To break on a specified 16-bit value: **data** <data> **word** To break on a specified 32-bit value: **data** <data> **long** <data>: Address value (numeric) Specifies a mask value: To break on a specified 8-bit value (mask): **data mask** <maskdata> **byte** To break on a specified 16-bit value (mask): **data mask** <maskdata> **word** To break on a specified 32-bit value (mask): **data mask** <maskdata> **long** <maskdata>: Specifies the mask data. To generate a break on data of size smaller than a byte, specify a mask. Instruction fetch cycles are not a target of break specifications.
<r/wopt>	Specifies a read/write condition. **direction read**: Searches for a read cycle, **direction write**: Searches for a write cycle

Parameter	Description
<accessopt>	Specifies a bus status condition. To break the user program with the execution cycle: **access dat** To break the user program with the DMA/DTC cycle: **access dma**
<countopt>	Specifies the pass count for the breakpoint. (This parameter can be used when <channel_number> is 1.) Any value from H'1 to H'FFFF can be specified.

[Description]

- Setting

  Sets an internal break condition. When the specified condition is satisfied, the user program execution breaks. For conditions that can be set, see the description of BCUS command <option>. A sequential break can be specified by using an internal break. When sb1, sb2, or sb3 is specified with the GO_OPTION command, up to four levels of internal sequential breaks of Break Condition U1 to U4 can be used.
- Cancellation

  Cancels the current internal breakpoints. When the channel number is omitted, cancels all internal breakpoints.
- Display

  Displays set internal breakpoints. When the channel number is omitted, the emulator displays all internal breakpoints. The display format is as follows:

  Break Condition Un: <Enable/Disable> <Settings>

  n: Number
- Enable or Disable

  Enables or disables current internal breakpoints. When the channel number is omitted, enables or disables all current internal breakpoints.

[Examples]

1. To set an address bus value of H'1000000 (<addropt>), byte data with the lowest bit zero D0 bit (<dataopt>), and the write cycle (<r/wopt>) for internal breakpoint channel 1:

   BCUS CHANNEL 1 ADDRESS H'1000000 DATA MASK B'****0 BYTE DIRECTION WRITE***
2. To set internal breakpoint channel 2 to break on the address bus value of H'1000000 before it is executed on the instruction fetch cycle (<addropt>):

   BCUS CHANNEL 2 ADDRESS H'1000000 PC
3. To set internal breakpoint channel 3 to break on the address bus value of H'1000000 before it is executed on the instruction fetch cycle (<addropt>):

   BCUS CHANNEL 3 ADDRESS H'1000000 PCAFTER

4. To set <accessopt> as the execution cycle and to set <r/wopt> as the read cycle on internal breakpoint channel 4:
   BCUS CHANNEL 4 ACCESS DAT DIRECTION READ

5. To specify address bus value H'1000000 as a reset point of the internal sequential break:
   BCUS RESET ADDRESS H'1000000

6. To display the internal breakpoint channel 1 settings:
   BCUD CHANNEL 1

Display:

```
>BCUD CHANNEL 1
Break Condition U1:Enable address H'1000000 data mask B'*******0 byte direction write
```

7. To display all internal breakpoints:
   BCUD

Display:

```
>BCUD
Break Condition U1:Enable address H'1000000 data mask B'*******0 byte direction write
Break Condition U2:Enable address H'1000000 pc
Break Condition U3:Enable address H'1000000 pcafter
Break Condition U4:Enable access dat direction read
Break Condition U Reset:Enable address H'1000000
```

8. To disable internal breakpoint channel 1 conditions:
   BCUE CHANNEL 1 DISABLE

9. To enable all internal breakpoint conditions:
   BCUE ENABLE

10. To cancel internal breakpoint channel 2 conditions:
    BCUC CHANNEL 2

11. To cancel all internal breakpoint conditions:
    BCUC

6.2.3 Hardware Sequential Break Specification Command (CSQ)

- Setting CONDITION_SEQUENCE (CSQ)
- Display CONDITION_SEQUENCE (CSQ)

[Command syntax and parameters]

- Setting **csq** <type> <action>
- Display **csq** <type>

Parameter	Type	Description
<type>	Keyword	Sets hardware sequential break type. **a/b**: Specifies Condition A/B
<action>	Keyword	Enables or disables hardware sequential function. **off**: No sequential specification **break**: Sequential break **trace**: Sequential trace stop

[Description]

- Setting

  This command specifies hardware sequential break or sequential trace stop, which are the conditions set by the user in hardware break condition (Break Condition A/B) or trace condition (Trace Condition A/B). The user must set the conditions by using the BCS or TAS command.
- Display

  This command displays the conditions currently set.

[Examples]

1. To specify Break Condition A as sequential break:

   CSQ A BREAK
2. To specify Trace Condition B as sequential trace stop:

   CSQ B TRACE
3. To cancel trace stop specification of CONDITION B:

   CSQ B OFF
4. To specify the contents set for CONDITION A:

   CSQ A

   Display:

   ```
   >CSQ A
   Break Condition A Sequential
   ```

[Notes]

- The emulator clears all conditions set to Break Condition A/B and Trace Condition A/B when hardware break condition (Break Condition A/B) and trace condition (Trace Condition A/B) are set, and when the user specifies **break** or **trace** with the CSQ command. Therefore, the user must set Break Condition A/B or Trace Condition A/B for hardware sequential break or trace sequential stop.
- The user cannot use this command to enable or disable sequential hardware breaks during user program execution.

6.2.4 Software Break Commands (BP, BC, BD, BE)

- Setting BREAKPOINT (BP)
- Cancellation BREAKPOINT_CLEAR (BC)
- Display BREAKPOINT_DISPLAY (BD)
- Enable or Disable
  BREAKPOINT_ENABLE (BE)

[Command syntax and parameters]

- Setting **bp** <address> [**count** <count>]
- Cancellation **bc** [<address>]
- Display **bd**
- Enable or Disable
  be [<address>] <mode>

Parameter	Type	Description
<address>	Numeric	Sets breakpoint address. When the user sets an odd address, the emulator will round it down to an even address.
<count>	Numeric	Sets the breakpoint pass count within the range from H'1 to H'FFFF. When the user does not specify the pass count, the setting will be H'1.
<mode>	Keyword	Enables or disables breakpoints. **enable** : Enables breakpoint setting. **disable** : Disables breakpoint setting.

[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. After emulation passes the specified number of breakpoints, the emulator stops the execution of the user program.

  Do not set software breakpoints to the following addresses:

  — An address whose memory content is H'0000
  — The address where BREAKSEQUENCE_SET was set
  — Areas other than CS (except for the internal ROM/RAM area)
  — Instructions that satisfy Break Condition U4
  — Slot instruction of the delay branch instruction

- Cancellation

  This command clears software breakpoints that have been set. When <address> is omitted, all of the set breakpoints will be cleared.

- Display

  This command displays software breakpoints that have been set.

  The following shows the display format:

  ```
  >BD
  H'00001000 D'1 (D'1) Enable
  H'00002000 D'1 (D'0) Disable
  H'00003000 D'1 (D'0) Enable
      (a)        (b)   (c)    (d)
  ```

  (a) Breakpoint address

  (b) Number of times specified

  (c) Pass count (cannot be displayed during user program execution)

  (d) Enable/Disable

- Enable or Disable

  This command enables or disables software breakpoints that have been set. When the user does not specify parameter <address>, all of the specified software breakpoints are enabled or disabled.

[Examples]

1. To set a software breakpoint at address H'00001000:

   BP H'00001000

2. To disable a software breakpoint set at address H'00001000:

   BE H'00001000 DISABLE

3. To enable a software breakpoint set at address H'00001000:

   BE H'00001000 ENABLE

4. To enable all software breakpoints that have been set:

   BE ENABLE

5. To clear a software breakpoint set at address H'00001000:

   BC H'00001000

6. To clear all software breakpoints:

   BC

6.2.5 Software Sequential Break Commands (BSS, BSC, BSD, BSE)

- Setting BREAKSEQUENCE_SET (BSS)
- Cancellation BREAKSEQUENCE_CLEAR (BSC)
- Display BREAKSEQUENCE_DISPLAY (BSD)
- Enable or Disable
  BREAKSEQUENCE_ENABLE (BSE)

[Command syntax and parameters]

- Setting **bss** <address1> <address2> [<address3...>...] [[**reset** <address8>]]
- Cancellation **bsc**
- Display **bsd**
- Enable or Disable
  bse <mode>

Parameter	**Type**	**Description**
<address1>	Numeric	Sets the address of the first pass point. When the user sets an odd address, it is rounded down to an even address.
<address2>	Numeric	Sets the address of the second pass point.
<address3...>	Numeric	Sets the addresses of the third pass point and subsequent points. The user can set up to seven pass points.
<address8>	Numeric	Sets the address of the reset point.
<mode>	Keyword	Enables or disables software sequential breakpoints. **enable** : Enables breakpoint setting. **disable** : Disables breakpoint setting.

[Description]

- Setting

  This command sets software sequential breakpoints. When the emulator executes the user program from the first pass point address, emulation will pass the software sequential breakpoints and stop at the last pass point. If emulation does not pass the pass points in the specified order, the emulator will start analyzing from the first pass point again. The user can specify up to seven breakpoints and a reset point. When emulation passes a reset point, the emulator starts analyzing from the first pass point.

  The user cannot set software sequential breakpoints to the following addresses:

  — An address whose memory content is H’0000

  — The address where BREAKPOINT command was set.

  — Areas other than CS (except for the internal RAM/ROM area)

  — Instructions that satisfy Break Condition U4

  — Slot instructions of the delay branch instruction

- Cancellation

  This command clears software sequential breakpoints that have been set. It also clears reset points.

- Display

  This command displays software sequential breakpoints that have been set.

  The following shows the display format:

```
>BSD
Enable : H'1000 H'2000 H'3000reset H'4000
  (a)          (b)                 (c)
1   H’00001000 (D’0)   (d)
2   H’00002000 (D’0)
3   H’00003000 (D’0)
4   Empty
5   Empty
6   Empty
7   Empty
R   H’00004000 (D’0)
```

  (a) Enable/Disable

  (b) Software sequential breakpoint address

  (c) Reset point address

  (d) Pass count of each pass point or reset point at the end of execution (cannot be displayed during execution)

- Enable or Disable

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

[Examples]

1. To set a software sequential breakpoint at which user program execution stops when the user program has passed the pass points in the order of H’C010000 and H'C020000, and the analysis for the pass sequence is reset when the user program has passed H'C030000:

   BSS H'C010000 H'C020000 RESET H'C030000
2. To set a software sequential breakpoint at which user program execution stops when the user program has passed the pass points in the order of H'C01000, H'C02000, and H'C03000:

   BSS H'C01000 H'C02000 H'C03000
3. To enable software sequential breakpoints that have been set.

   BSE ENABLE
4. To disable software sequential breakpoints that have been set.

   BSE DISABLE
5. To clear software sequential breakpoints that have been set.

   BSC

6.2.6 CHECK (CHECK)

[Command syntax and parameters]

check

[Description]

This command tests the MCU pin state. When the emulator generates an error, the following is displayed:

<Pin name> HIGH or LOW

The following shows the pins to test the pin state.

Pin Name	Error State
IRQ0	Displays high or low IRQ0 pin state.
IRQ1	Displays high or low IRQ1 pin state.
IRQ2	Displays high or low IRQ2 pin state.
IRQ3	Displays high or low IRQ3 pin state.
IRQ4	Displays high or low IRQ4 pin state.
IRQ5	Displays high or low IRQ5 pin state.
IRQ6	Displays high or low IRQ6 pin state.
IRQ7	Displays high or low IRQ7 pin state.
WAIT	Displays high or low WAIT pin state.
BREQ	Displays high or low BREQ pin state.
RES	Displays high or low RES pin state.
MRES	Displays high or low MRES pin state.
NMI	Displays high or low NMI pin state.
HSTBY	Displays high or low HSTBY pin state.
PVCC (5V)	Displays NG when PVcc is less than 4.0 V, and OK when 4.0 V or more.
PVCC (3.3V)	Displays NG when PVcc is less than 2.6 V, and OK when 2.6 V or more.

[Example]

To test the MCU pin state:

CHECK

The display format is as follows:

```
>CHECK
IRQ0          HIGH
IRQ1          HIGH
IRQ2          HIGH
IRQ3          HIGH
IRQ4          HIGH
IRQ5          HIGH
IRQ6          HIGH
IRQ7          HIGH
WAIT          HIGH
BREQ          HIGH
RES           HIGH
MRES          HIGH
NMI           HIGH
HSTBY         HIGH
PVCC(5V)      NG
PVCC(3.3V)    NG
```

6.2.7 CLOCK (CK)

[Command syntax and parameters]

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

Parameter	Type	Description
<mode>	Keyword	Selects the clock signal. **user**: Clock signal of user system **4**: Clock signal (4 MHz) of the emulator **6**: Clock signal (6 MHz) of the emulator **8**: Clock signal (8 MHz) of the emulator **10**: Clock signal (10 MHz) of the emulator **12**: Clock signal (12.5 MHz) of the emulator **xtal**: Crystal oscillator of the evaluation chip 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 internal I/O registers and control registers hold the reset values. (For the reset values, refer to appendix E.6, Differences between the Emulator and the MCU.)

This command displays the type of the clock signal that has been set. When the user selects **user** for the clock signal, but the user system clock is not input from the user system, the emulator will generate an error and select the clock in the emulator. When the user turns on the emulator, the emulator will select the user system clock (**user**), evaluation chip board crystal oscillator (**xtal**), and emulator internal clock signal (**12**), and set the correct clock signal.

[Examples]

1. To select the user system clock signal:
   CK USER
2. To display the selected clock signal:
   CK
   The display format is as follows:
   ```
   >CK
   Clock=Emulator Clock (12.5MHz)
   ```

[Note]

When the user system is selected and the following abnormal condition holds, the emulator system program will not run correctly and the emulator will display error message, USER SYSTEM NOT READY, when the user inputs commands or starts the HDI. In this case, the user must exit the HDI and start it again.

- If **user** has been selected and the user system clock has been used, but the user system clock is cut off (V_{CC} is supplied correctly.)

6.2.8 Coverage Commands (CVS, CVC, CVD)

- Setting COVERAGE_SET (CVS)
- Cancellation COVERAGE_CLEAR (CVC)
- Display COVERAGE_DISPLAY (CVD)

[Command syntax and parameters]

- Setting **cvs** <mode>
- Cancellation **cvc** [<addropt>]
- Display **cvd** <addropt> [<dispopt>]
  Displays the result of a coverage trace.
  cvs
  Displays the settings.

Parameter	Description
<mode>	Enables or disables the coverage tracing function. When **off** is specified, neither **cvc** nor **cvd** command is available. **on**: Enabled **off**: Disabled (default)
<addropt>	Sets the address range in which the result of coverage tracing will be placed for display. Parameter **rom** can be specified only in the internal ROM enabled mode. **start** <address> **end** <address> <address>: Address (numeric) To specify one address range of internal RAM (H'FFFF0000 to H'FFFF3FFF): **ram1** To specify the other address range of internal RAM (H'FFFFC000 to H'FFFFFFFF): **ram2** To specify the address range of internal ROM (H'00000000 to H'000FFFFF): **rom**
<dispopt>	Sets the display format of the result of coverage trace. When omitted, the executed address range is displayed. To display the data as a list of the addresses of instructions that have been executed: **address** To display the data as a list of the addresses of instructions that have not been executed: **address not** To display the data as a list of the addresses of instructions that have been executed or not with the dump format in byte units: **dump** To display the rate of executed or non-executed address: **rate**

[Description]

- Setting

  This command sets whether or not the coverage trace function is used.
- Cancellation

  This command initializes the result of a coverage trace.
- Display

  This command displays the result of coverage trace.

  - When the **dump** option is set and the address range is specified, the start address and end address are adjusted to be a multiple of 16 and (multiple of 16) – 1, respectively.

    xxxxxxxx yy yy yy yy yy yy yy yy   yy yy yy yy yy yy yy yy

    xxxxxxxx: Address
    yy: Displays information on executed addresses as a hexadecimal (00-FF) number. One bit is one address, and 1 is set in bits that correspond to executed addresses.

    (Example)
    0001000   8F 00 ....

    The data 8F at the head of the list indicates that addresses 1000, 1008, 100A, 100C, and 100E all have instructions that have been executed. The addresses, until they are reinitialized, display the coverage trace information on the program that has been executed.

  - When the **rate** option is set, the ratio of addresses that have executed instructions to all the addresses is displayed.

    <S-ADDR> <E-ADDR> <RATE>
    xxxxxxxx – xxxxxxxx   rr.rr% (e/a)

    xxxxxxxx – xxxxxxxx: Coverage address range
    rr.rr: Rate
    e: Number of instructions executed
    a: All addresses

[Examples]

1. To use the coverage trace function:

   CVS ON
2. To display the settings of the coverage trace function:

   CVS

   The display format is as follows:

   ```
   >CVS
   CVS ON
   ```
3. To reinitialize the result of a coverage trace from H’400 to H’40F:

   CVC START 400 END 40F
4. To reinitialize the result of a coverage trace:

CVC

5. To display the data as a list of the addresses of instructions that have been executed from H’400 to H’40F:

   CVD START 400 END 40F ADDRESS

   The display format is as follows:

```
>CVD START 400 END 40F ADDRESS
<S-ADDR> <E-ADDR>
00000400-0000040B
```

6. To display the data as a list of the addresses of instructions that have not been executed from H’400 to H’40F:

   CVD START 400 END 40F ADDRESS NOT

   The display format is as follows:

```
>CVD START 400 END 40F ADDRESS NOT
<S-ADDR> <E-ADDR>
0000040C-0000040F
```

7. To display the ratio of executed address to all the addresses from H’400 to H’40F:

   CVD START 400 END 40F RATE

   The display format is as follows:

```
>CVD START 400 END 40F RATE
<S-ADDR> <E-ADDR> <RATE>
00000400-0000040F 75.00%(6/8)
```

8. To display the rate of the data as a list of the addresses of instructions that have been executed or not from H’400 to H’40F with the dump format:

   CVD START 400 END 40F DUMP

   The display format is as follows:

```
>CVD START 400 END 40F DUMP
00000400  FF FF FF FF FF FF FF FF   FF FF FF FF 00 00 00 00
```

6.2.9 DEVICE_TYPE (DE)

[Command syntax and parameters]

de

[Description]

This command displays the selected target MCU.

[Examples]

To display the selected MCU:

DE

The display format is as follows:

```
>DE

Current device = SH7046
```

6.2.10 END (END)

[Command syntax and parameters]

end

[Description]

This command returns control to user program emulation when the trace halt state is entered due to the satisfaction of trace conditions. This command clears the trace information and the emulator starts to acquire new trace information.

[Example]

To return the emulator state from parallel mode to user program execution mode:

END

6.2.11 EXECUTION_MODE (EM)

[Command syntax and parameters]

- Setting **em** [<time_count>] | [<timeout>] | [<multi_break>] | [<internal_rom>] | [<reset_signal>] | [<busreqeust>] | [<wait_signal>] | [<trigger_bcb>] | [<trigger_bcu>]
- Display **em**

Parameter	Type	Description
<time_count>	Keyword	Specifies the execution-time measurement unit. **time 52us**: 52 µs unit. **time 1.6us**: 1.6 µs unit. **time 20ns**: 20 ns unit (default).
<timeout>	Keyword	Sets the timeout detection time. **tout 100us**: 100 µs unit (default) **tout 1.6ms**: 1.6 ms unit **tout 13ms**: 13 ms unit **tout 210ms**: 210 ms unit
<multi_break>	Keyword	Enables or disables the multibreak function (the emulator can simultaneously stop the execution of user programs in other emulators by using external probe 1). **mb enable**: Enables multibreak. **mb disable**: Disables multibreak (default).
<internal_rom>	Keyword	Enables or disables the write to the internal ROM. **rom enable**: Enabled. **rom disable**: Disabled (default).
<reset_signal>	Keyword	Enables or disables the input of RES (power-on reset) signal. **res enable**: Enabled (default). **res disable**: Disabled.
<busrequest>	Keyword	Enables or disables the input of the bus-mastership request signal. **breq enable**: Enabled (default). **breq disable**: Disabled.
<wait_signal>	Keyword	Enables or disables the input of WAIT signal. **wait enable**: Enabled (default). **wait disable**: Disabled.
<trigger_bcb>	Keyword	Specifies the pulse output mode when the emulator satisfies Break Condition B or Trace Condition B. **trgb 1, trgb 2, trgb 3, trgb 4, trgb 5, trgb 6, trgb 7, or trgb 8**: The emulator will output a pulse when the emulator satisfies a channel condition set to Break Condition B or Trace Condition B. **trgb all**: The emulator will output a pulse when the emulator satisfies a channel condition set to Break Condition B or Trace Condition B. **trgb disable**: The emulator stops the execution of user program but does not output a pulse. (default)

Parameter	Type	Description
<trigger_bcu>	Keyword	Specifies the pulse output mode when the emulator satisfies hardware break (BCU). **trgu enable**: The emulator stops the execution of user program and outputs a pulse. **trgu stop**: The emulator does not stop the execution of user program but outputs a pulse. **trgu disable**: The emulator stops the execution of user program but does not output any pulse. (default)

[Description]

This command displays and sets debugging conditions while the emulator executes user program.

[Examples]

1. To display current debugging conditions set during user program execution:

   EM

   The following shows the display format:

```
>EM
Execution Mode
Condition A Sequential      Not used
Condition B Sequential      Not used
Interval Timer counter      20ns
Bus timeout                 100us
Multi break (PRB1)          Disabled
Internal ROM Option         Disabled
RES signal                  Enabled
BREQ signal                 Enabled
WAIT signal                 Enabled
Output trigger (TRGB)       Disabled
Output trigger (TRGU)       Disabled
```

2. To enable the input of the bus-mastership request signal for the debugging conditions set during user program execution:

   EM BREQ ENABLE

3. To output a trigger pulse when the hardware break conditions are satisfied for any of the channels set by Break Condition B or Trace Condition B, and to set the execution-time measurement unit to 20 ns for the debugging conditions set during user program execution:

   EM TIME 20NS TRGB ALL

6.2.12 GO_OPTION (GP)

[Command syntax and parameters]

- Setting **gp eml_mode** <eml_mode>
- Display **gp**

Parameter	Type	Description
<eml_mode>	Keyword	Sets the emulation mode. Refer to the next table for settings.

Pin Mode	Description
normal	Executes the user program normally.
6.5us	Executes the user program by inputting the RES signal to the MCU at intervals of 6.5 µs.
9.8us	Executes the user program by inputting the RES signal to the MCU at intervals of 9.8 µs.
50us	Executes the user program by inputting the RES signal to the MCU at intervals of 50 µs.
100us	Executes the user program by inputting the RES signal to the MCU at intervals of 100 µs.
500us	Executes the user program by inputting the RES signal to the MCU at intervals of 500 µs.
1ms	Executes the user program by inputting the RES signal to the MCU at intervals of 1 ms.
5ms	Executes the user program by inputting the RES signal to the MCU at intervals of 5 ms.
10ms	Executes the user program by inputting the RES signal to the MCU at intervals of 10 ms.
50ms	Executes the user program by inputting the RES signal to the MCU at intervals of 50 ms.
100ms	Executes the user program by inputting the RES signal to the MCU at intervals of 100 ms.
500ms	Executes the user program by inputting the RES signal to the MCU at intervals of 500 ms.
1s	Executes the user program by inputting the RES signal to the MCU at intervals of 1 s.
pabreak	A break occurs under the timeout condition set by the [Performance 1] dialog box or the PERFORMANCE_SET command (set by channel 1).
patrace	A trace acquisition under the timeout condition set by the [Performance 1] dialog box or the PERFORMANCE_SET command (set by channel 1 and execution continues).
sb1	Internal sequential break mode 1 (A break occurs when break conditions set by Break Condition U2,1 are satisfied in the sequence of 2, 1.)
sb2	Internal sequential break mode 2 (A break occurs when break conditions set by Break Condition U3,2,1 are satisfied in the sequence of 3, 2, 1.)
sb3	Internal sequential break mode 3 (A break occurs when break conditions set by Break Condition U4,3,2,1 are satisfied in the sequence of 4, 3, 2, 1.)
measurement1	Sets time interval measurement mode 1.
measurement2	Sets time interval measurement mode 2.
measurement3	Sets time interval measurement mode 3.

Pin Mode	Description
timcb	Measures the execution time with the execution time measurement function by specifying the condition.
no_break	Temporarily disables the software and hardware break conditions.

[Description]

This command displays and sets emulation mode during user program execution.

[Examples]

1. To set the emulation mode so that the user program is executed by inputting the RES signal to the MCU at intervals of 100 ms:
   GP EML_MODE 100MS
2. To display the current emulation mode during user program execution:
   GP
   The display format is as follows:
   ```
   >GP
   Emulator emulation mode=Cyclic reset(100ms)
   ```

[Notes]

- When the user selects 6.5us, 9.8us, 50us, 100us, 500us, 1ms, 5ms, 10ms, 50ms, 100ms, 500ms, or 1s for the emulation mode, the emulator will disable all trace and break conditions. Also, the emulator cannot halt tracing when the user selects the [Halt] button in the [Trace] window.
- When the user selects measurement1, measurement2, measurement3, or timcb for the emulation mode, the emulator cannot halt tracing when the user selects the [Halt] button in the [Trace] window.
- When the user selects sb1, sb2, or sb3, the user must set Break Condition U.
- When the user selects measurement1, measurement2, measurement3, or timcb, the user must set Break Condition U.
- When the user selects sb1, sb2, sb3, pabreak, patrace, measurement1, measurement2, or measurement3 for the emulation mode, the emulator will disable the software break conditions.

6.2.13 ID (ID)

[Command syntax and parameters]

id

[Description]

This command displays the emulator system program version.

[Examples]

To display the emulator system program version:

ID

The display format is as follows:

```
>ID

SH7046 E8000S Emulator system file Vm.n

Copyright (C) Hitachi, Ltd. xxxx

Licensed Material of Hitachi, Ltd.
```

6.2.14 MAP_SET (MS)

[Command syntax and parameters]

ms <start> <end> <mode>

Parameter	Type	Description
<start>	Numeric	Sets the start address.
<end>	Numeric	Sets the end address.
<mode>	Keyword	Sets the memory map mode. **user**: Uses user memory (does not use the emulation memory). **user-read-only**: Uses user memory with write protection. **user-guarded**: Uses user memory with access prohibited. **emulator**: Uses the emulation memory area. **read-only**: Protects the emulation memory area from being written. **guarded**: Prohibits writes to the emulation memory area.

[Description]

This command sets the emulator's emulation memory.

[Example]

To set the address range from H'4000000 to H'4FFFFF as the emulation memory area:

MS H'4000000 H'4fffff EMULATOR

[Notes]

- The user must not allocate the emulation memory to an area other than CS.
- The user cannot set the memory map over the reserved area.
- The standard emulation memory cannot be allocated in the single-chip mode.

6.2.15 Performance Commands (PS, PC, PA)

- Setting PERFORMANCE_SET (PS)
- Cancellation PERFORMANCE_CLEAR (PC)
- Display PERFORMANCE_ANALYSYS (PA)

[Command syntax and parameters]

- Setting **ps channel** <channel_number> <modeopt> <nameopt> <startopt> <endopt>
  [[[<timeopt>] [<countopt>]] | [<accessopt>] | [<subroutineopt>]]
- Cancellation **pc** [**channel** <channel_number>]
- Display **pa** [<display_mode>]

Parameter	Type	Description
<channel_number>	Numeric	Sets the performance channel number from 1 to 8.
<display_mode>	Keyword	Sets the display mode of program execution state. **address**: Displays subroutine address list. **count**: Displays execution time and execution count in numeric. **graph**: Displays an execution time ratio in graph form (default). **init**: Initializes measurement information.

Parameter	Description
<modeopt>	Sets the conditions to acquire data relating to performance analysis. There are three ways to set the conditions. • Time measurement mode 1 The emulator measures the execution time of the subroutine between <startopt> and <endopt> and counts the number of times it is executed. The emulator starts to measure on prefetching of instruction in the range of <startopt> and <endopt> and stops measuring on prefetching of instruction outside the specified range. The emulator restarts measurement when another instruction is prefetched from the specified range. The emulator increments the number of passes when it passes through the <startopt> and fetches the instruction from <endopt>. The emulator does not measure the performance of subroutines that were called from the target subroutine. Specification: **mode time1** • Time measurement mode 2 The emulator measures the execution time of the subroutine between <startopt> and <endopt> and counts the number of times it is executed. The emulator starts to measure on prefetching of instruction at <startopt> and stops measuring on prefetching of instruction at <endopt>. The emulator increments the number of passes (<countopt>) whenever it passes through the <startopt> and fetches the instruction from <endopt>. The emulator also measures the performance of subroutines that were called from the target subroutine. Specification: **mode time2** • Time measurement mode 3 The emulator starts to measure on prefetching of instruction at <start address range> and stops measuring at <end address range>. The emulator increments the number of passes (<countopt>) whenever it passes through <start address range> and fetches the instruction from <end address range>. Specification: **mode time3** The user can use channels 1, 3, 5, and 7 in time measurement mode 3. • Area access count mode The emulator counts the number of times the subroutine between <startopt> and <endopt> has accessed the area specified by <accessopt>. The subroutine execution time is measured by using subroutine time measurement mode 1. Specification: **mode access** The user can use channels 1, 3, 5, and 7 in area-access count mode.

Parameter	Description
<modeopt> (cont)	• Subroutine call count measurement mode The emulator measures counts that the subroutine specified by <startopt> and <endopt> has called the subroutine specified by <subroutineopt>. The subroutine execution time is measured by using subroutine time measurement mode 1. Specification: **mode subroutine** The user can use channels 1, 3, 5, and 7 in subroutine call count measurement mode.
<nameopt>	Specifies the subroutine to measure the performance. Specification: **name** <subroutine name>
<startopt>	Specifies the subroutine start address in time measurement modes 1 and 2, area-access count mode, and subroutine call count measurement mode. Specification: **start** <address> <address>: Address value To specify the start address range in time measurement mode 3: Specification: **start** <address1> **to** <address2> <address1>: Start address (numeric) <address2>: End address (numeric)
<endopt>	Specifies the subroutine end address in time measurement modes 1 and 2, area-access count mode, and subroutine call count measurement mode. Specification: **end** <address> <address>: Address (numeric) To specify the end address range in time measurement mode 3: Specification: **end** <address1> **to** <address2> <address1>: Start address (numeric) <address2>: End address (numeric)
<timeopt>	Specifies the timeout period. (The user can use parameter <timeopt> in channel number 1 and time measurement modes 1 and 2.) When the user sets a condition to PERFORMANCE ANALYSIS1, the emulator stops the execution of the user program after the time set to <timeopt> has passed. In this case, the user must specify **pabreak** in the GO_OPTION command. Specification: **time** <time> Parameter <time> specifies the time value in the following format: [:mm[:ss[:uuuuuu]]] mm : minute (0 to 59) ss : second (0 to 59) uuuuuu : microsecond (0 to 999999)

Parameter	Description
<countopt>	Specifies the pass count. (The user can use parameter <countopt> in channel number 1 and subroutine time measurement modes 1 and 2.) Specification: **count** <count> H'1 to H'FFFF can be specified for count.
<accessopt>	Specifies the address range accessed from the subroutine in the area-access count mode. (The user can use parameter <accessopt> in **mode access**.) Specification: **access** <address1> **to** <address2> <address1>: Start address (numeric) <address2>: End address (numeric)
<subroutineopt>	Specifies the address range of a child subroutine accessed from the parent subroutine in the subroutine call count measurement mode. (The user can use parameter <subroutineopt> in **mode subroutine**.) Specification: **subroutine** <address1> **to** <address2> <address1>: Start address (numeric) <address2>: End address (numeric)

[Description]

- Setting

  This command sets the condition to measure the subroutine performance. Up to eight subroutines can be specified in measurement modes 1 and 2, and up to four subroutines can be specified in measurement mode 3, area-access count mode, and subroutine call count measurement mode.

- Cancellation

  This command clears the condition to measure the subroutine performance. When the user specifies a channel number, the emulator clears the specified channel number. When the user does not specify any channel number, all of the performance conditions are cleared.

- Display

  This command displays the result of measuring the subroutine performance.

  The following shows the display format:

  1. To display an execution time ratio for the program execution state:

     ```
     >PA GRAPH
     NO    NAME   MODE   RATE      0---10---20---30---40---50---60---70---80---90---100
     1     SUBA   I1     D'10.0%   *****
     2     SUBB   I2     D'20.0%   **********
     3     SUBC   I3     D'30.0%   ***************
     5     SUBD   AC     D'15.0%   ********
     7     SUBE   SC     D'30.0%   ***************
      (a)  (b)    (c)    (d)                          (e)
     ---------------------------------------------------------------------------
     TOTAL RUN-TIME = D'0000H:00M:10S:000020US:250NS    (f)
     ```

     (a) Channel number

     (b) Subroutine name (up to 32 characters can be used)

     (c) Time measurement mode (I1: Time measurement mode 1, I2: Time measurement mode 2, I3: Time measurement mode 3, AC: Area-access count mode, SC: Subroutine call count measurement mode)

     (d) Displays the execution time ratio numerically.

     (e) Displays the execution time ratio as a graph.

     (f) Total execution time (Hours (H), minutes (M), seconds (S), microseconds (US), and nanoseconds (NS))

2. To display subroutine address list:

>PA ADDRESS

```
NO      NAME        MODE    ADDRESS
1       SUBA        I1      00000100:00001FF0 TIME=xxxH:xxM:xxS:xxxxxxUS COUNT=nnnnnnnn
(a)     (b)         (c)     (d)       (e)          (f)                         (g)
2       SUBB        I2      00005000:00007FF0
3       SUBC        I3      00010000:0001008F      (h)
                            00020000:00020098      (i)
5       SUBD        AC      00002030:0000207F
        <ACCESS>            FFFFFF00:FFFFFF7F
                             (j)
7       SUBE        SC      00020100:0002FFFF
        <CALL-SUB>          00030000:00030060  (k)
---------------------------------------------------------------------------
TOTAL RUN-TIME = D'0000H:10M:00S:000020US:250NS  (l)
```

(a) Channel number

(b) Subroutine name (up to 32 characters can be used)

(c) Time measurement mode (I1: Time measurement mode 1, I2: Time measurement mode 2, I3: Time measurement mode 3, AC: Area-access count mode, SC: Subroutine call count measurement mode)

(d) Subroutine start address

(e) Subroutine end address

(f) Timeout value (Timeout values can be displayed in time measurement modes 1 and 2 when the timeout condition is specified.)

(g) Counter value (Counter values can be displayed in time measurement modes 1 and 2 when count condition is specified.)

(h) Start address range (Time measurement mode 3)

(i) End address range (Time measurement mode 3)

(j) Area-access address range in the area-access count mode

(k) Call-subroutine address range in the subroutine call count measurement mode

(l) Total execution time

3. To display an program execution time and execution count numerically:

   >PA COUNT

```
NO     NAME    MODE   RATE     RUN-TIME                                   E-COUNT
1      SUBA    I1     D'10.0%  D'0000H:00M:10S:010305US:500NS             D'00005
(a)    (b)     (c)    (d)                (e)                              (f)
       MAX D'0000H:00M:05S:001000US:250NS MIN D'0000H:00M:05S:001000US:250NS
       (g)                                (h)
       AVE D'0000H:00M:05S:001000US:250NS
                      (i)
2      SUBB    I2     D'20.0%  D'0000H:00M:10S:010305US:500NS             D'00010
       AVE D'0000H:00M:05S:001000US:250NS
3      SUBC    I3     D'30.0%  D'0000H:00M:10S:010305US:500NS             D'00010
       AVE D'0000H:00M:05S:001000US:250NS
5      SUBD    AC     D'10.0%  D'0000H:00M:05S:001000US:250NS
       <ACCESS>                                                           D'00005
7      SUBE    SC     D'20.0%  D'0000H:00M:10S:010305US:500NS
       <CALL-SUB>                                                         D'00010
--------------------------------------------------------------------------
TOTAL RUN-TIME = D'0000H:00M:08S:029397US:600NS      (j)
```

   (a) Channel number

   (b) Subroutine name (up to 32 characters can be used)

   (c) Time measurement mode (I1: Time measurement mode 1, I2: Time measurement mode 2, I3: Time measurement mode 3, AC: Area-access count mode, SC: Subroutine call count measurement mode)

   (d) Displays the execution time ratio numerically.

   (e) Execution time

   (f) Area-access count mode: Area access count, subroutine call count measurement mode: subroutine call count

   (g) Maximum subroutine execution time (Time measurement mode 2)

   (h) Minimum subroutine execution time (Time measurement mode 2)

   (i) Average subroutine execution time (Time measurement mode 2)

   (j) Total execution time

[Examples]

1. To set the following conditions to acquire on channel 2:

   Subroutine time measurement mode: 1,

   Subroutine name: SORT1,

   Start address: H'10002E,

End address: H'10015C.

PS CHANNEL 2 MODE TIME1 NAME SORT1 START H'10002E END H'10015C

2. To set the following conditions to acquire on channel 5:

   Subroutine time measurement mode: 3,

   Subroutine name: TEST1,

   Start address range: H'100000 to H'10002E,

   End address range: H'100030 to H'10015C.

   PS CHANNEL 5 MODE TIME3 NAME TEST1 START H'100000 TO H'10002E END H'100030 TO H'10015C

3. To clear the performance condition set to channel 2:

   PC CHANNEL 2

4. To clear all performance conditions:

   PC

[Notes]

- The emulator measures the performance analysis of a subroutine by using the address bus value of the prefetch cycle. Therefore, if the user sets a subroutine end address to an address near to an address next to a branch or delay slot instruction, the emulator will not be able to measure the performance analysis correctly. Therefore, before setting the end address, the user must check how the MCU operates after the cycle that the branch instruction are prefetched and make sure not to set the end address to address in the prefetch cycle that will not be executed due to a branch instruction.
- The emulator can continuously measure performance analysis up to 14 minutes (when specified as 52 µs), 26 seconds (when specified as 1.6 µs), or 0.33 second (when specified as 20 ns) by setting the TIME option in the EXECUTION_MODE command.
- When the user sets break or trace condition to Break Condition C or Trace Condition C, no condition can be set to PERFORMANCE_ANALYSIS that has the same number. In other words, when the user sets break or trace condition to Break Condition C1 or Trace Condition C1, no condition can be set to PERFORMANCE ANALYSIS1. To set a condition to PERFORMANCE_ANALYSIS, the user must cancel the settings of Break Condition C or Trace Condition C.
- The emulator increments the pass count when it passes through a subroutine end address. Therefore, the emulator will display one more than the specified pass count, the subroutine execution time, and number of times the subroutine was executed.
- The emulator cannot measure the step execution time.

6.2.16 REFRESH (RF)

[Command syntax and parameters]

rf

[Description]

Updates the HDI memory information.

[Examples]

To update the HDI memory information:
RF

6.2.17 STATUS (STS)

[Command syntax and parameters]

sts

[Description]

Displays status information for the emulator. For details, refer to the contents of the [System Status] window described in section 5.9, Displaying Various Information.

[Examples]

To display status information for the emulator:
STS

Display:

```
>STS
Emulator Status
Connected To:                  SH7046 E8000S Emulator
CPU                            SH7046
Operating Mode Auto select     Disabled
Clock Mode (MD3-2)             Clock Mode 0
Operating Mode (MD1-0)         Mode 3 (single chip mode)
MD Value (E8000S)              03
MD Value (User System)         00
Pin Counts (Device)            80
IRQ0                           HIGH
IRQ1                           HIGH
IRQ2                           HIGH
IRQ3                           HIGH
IRQ4                           HIGH
IRQ5                           HIGH
IRQ6                           HIGH
IRQ7                           HIGH
BREQ                           HIGH
BACK                           HIGH
WAIT                           HIGH
```

```
Vcc Down Detection Level      2.6V
H-UDI (JTAG) Clock            2.5MHz
Clock source                  Emulator Clock (12.5MHz)
Run status Break
Cause of last break
Interval Time Count (2->1)
     MAX (2->1)
     MIN (2->1)
     AVE (2->1)
Interval Time Count (4->3)
     MAX (4->3)
     MIN (4->3)
     AVE (4->3)
Run Time Count
Condition A Sequential        Not used
Condition B Sequential        Not used
Interval Timer counter        20ns
Bus timeout                   100us
Multi break (PRB1)            Disabled
Internal ROM Option           Disabled
RES signal                    Enabled
BREQ signal                   Enabled
WAIT signal                   Enabled
Output trigger (TRGB)         Disabled
Output trigger (TRGU)         Disabled
Emulation mode                Normal
Prohibit R/W on the fly       Disabled
Interrupts during step        Disabled
```

6.2.18 Trace Condition Commands (TAS, TAC, TAD)

- Setting TRACEACQUISITION_SET (TAS)
- Cancellation TRACEACQUISITION_CLEAR (TAC)
- Display TRACEACQUISITION_DISPLAY (TAD)

[Command syntax and parameters]

- Setting **tas** <type> **channel** <channel_number> <tracetype> <option> [<option>…]
  <option> = <addropt> | <dataopt> | <accessopt> | <r/wopt> | <prbopt> |
  <nmiopt> | <irqopt> | <countopt> | <delayopt>
- Cancellation **tac** <type> [**channel** <channel_number>]
- Display **tad** <type> [**channel** <channel_number>]

Parameter	Type	Description
<type>	Keyword	Selects the trace condition type. **a/b/c**: Sets Trace Condition A/B/C
<channel_number>	Numeric or keyword	Sets a channel number from 1 to 8. When hardware sequential condition has been specified, specify **p** or **r**. **p**: Sequential point **r**: Reset point
<tracetype>	Keyword	Sets conditions for trace information acquisition. (This parameter cannot be specified when the sequential trace stop is specified.) **type range**: Range trace **type stop**: Trace stop **type subroutine** <subroutine>: Subroutine trace (<option> cannot be specified) **type subrange** <subroutine>: Range trace in the subroutine (only valid when <channel_number> is 1, 3, 5, or 7)
<subroutine>	Numeric	Specifies the start and end addresses of the subroutine. To specify a pass count: <address1> **to** <address2> <address1>: Start address (numeric) <address2>: End address (numeric)

Description of the tas command <option> (Specify one or more conditions.)

Parameter	Description
<addropt>	Specifies an address condition. To specify an address as a trace condition: **address** <address> [**not**] To specify an address range as a trace condition: **address** <address1> **to** <address2> [**not**] To mask address data: **address mask** <maskdata> [**not**] <address>: Address (numeric) <address1>: Start address (numeric) <address2>: End address (numeric) <maskdata>: Mask data Add **not** to specify an address or range outside which the user program should be traced. However, do not add **not** when <type> is **c**.

Parameter	Description
<dataopt>	Specifies a data condition. Valid when <type> is **a** or **b**. To specify an 8-bit data trace condition: **data** <data> **byte** <position> [**not**] To specify a 16-bit data trace condition: **data** <data> **word** <position> [**not**] To specify a 32-bit data trace condition: **data** <data> **long** [**not**] <data>: Specify data value (numeric) <position>: Specify valid position of the data bus value **high**: upper word **low**: lower word **byte3**: upper byte of the upper word **byte2**: lower byte of the upper word **byte1**: upper byte of the lower word **byte0**: lower byte of the lower word When **byte** is set for <datawidth>, **byte3**, **byte2**, **byte1**, or **byte0** is valid. When **word** is set for <datawidth>, **high** or **low** is valid. When **long** is set, <position> cannot be selected. Mask specification is as follows: To specify an 8-bit data trace condition (mask specification): **data mask** <maskdata> **byte** <position> [**not**] To specify a 16-bit data trace condition (mask specification): **data mask** <maskdata> **word** <position> [**not**] To specify a 32-bit data trace condition (mask specification): **data mask** <maskdata> **long** [**not**] <maskdata>: Specifies the mask data. To trace the user program at values smaller than the data size, specify the mask data. Add **not** to the specification to trace the data other than the specified one.
<accessopt>	Specifies bus status condition. To break the user program with the data access cycle: **access dat** To break the user program with the DMA/DTC cycle: **access dma**

<table>
<tr><th>Parameter</th><th>Description</th></tr>
<tr><td><r/wopt></td><td>Specifies a read or write condition. (Valid when <type> is a or b.)
To trace the user program during the read cycle: direction read
To trace the user program during the write cycle: direction write</td></tr>
<tr><td><prbopt></td><td>Specifies an external probe signal condition. (Valid when <type> is a or b.)
To specify an external probe signal condition: prb <bit specification>

A bit is specified as follows:
PRB1 to PRB4 signal bit specification
3 2 1 0 : Bit location
x x x x : Value to be specified (Specify 0 (low level) or 1 (high level) for x.)
| | | |
4 3 2 1 : PRB number
When * is specified, the condition can be specified not to include the external probe signal condition for bit location.</td></tr>
<tr><td><nmiopt></td><td>Specifies an NMI signal. (Valid when <type> is a or b.)
To trace the user program when the NMI signal is high: nmi hi
To trace the user program when the NMI signal is low: nmi low</td></tr>
<tr><td><irqopt></td><td>Specifies IRQ0 to IRQ7 signal conditions. (Valid when <type> is a or b.)
To specify IRQ0 to IRQ7 signal conditions: irq <bit specification>

A bit is specified as follows:
Specify each bit as follows:
To specify bits IRQ0 to IRQ7
7 6 5 4 3 2 1 0 : Bit location
x x x x x x x x : Value to be specified (Specify 0 (low level) or
| | | | | | | | 1 (high level) for x.)
7 6 5 4 3 2 1 0: IRQ number
When * is specified, the condition can be specified not to include the IRQ condition for bit location.</td></tr>
<tr><td><countopt></td><td>Specifies the pass count until trace acquisition condition is satisfied. (Valid when <tracetype> is stop, and <type> is a or b.)
To specify a pass count: count <value>
Any value from H'1 to H'FFFF can be specified as <value>.</td></tr>
</table>

Parameter	Description
<delayopt>	Specifies the number of bus cycles to be executed after the trace acquisition condition is satisfied. (Valid when <tracetype> is **stop**, <type> is **a** or **b**, and <channel_number> is 7.) To specify the number of bus cycles to be executed: **delay** <value> Any value from H'1 to H'7FFF can be specified as <value>.

[Description]

- Setting

  Specifies a trace acquisition condition (Trace Condition A/B/C).

  — Free trace

  Acquires trace information in all bus cycles when no conditions are set.

  — Range trace

  Acquires trace information in the bus cycles within which the specified conditions are matched. For a description of the conditions that can be specified, see the description of <option> for the tas command.

  — Trace stop

  When the set condition is satisfied, the emulator stops acquiring trace information and the system enters the trace halt state. For conditions that can be set, see the description of <option> for the tas command. Although the user program is still in emulation, trace information cannot be acquired in the trace halt state. When the trace stop condition is satisfied, the following message is displayed on the status bar and a message box is displayed.

  ** TRACE STOP **

  — Subroutine trace

  Traces access to instruction and operands within the range (subroutine) set as the start and end addresses.

  — Subroutine range trace

  Acquires trace information in the bus cycles that match the specified conditions when instructions and operands of the subroutine set by its start and end addresses are accessed. For conditions that can be set, see the description of <option> for the tas command.

  When more than one condition is set for range tracing, the trace information is acquired when the OR of the conditions is satisfied. When the trace stop condition is set, trace information is acquired until the trace stop condition is satisfied. When the trace stop condition is satisfied, acquisition of trace information halts, and the system enters the trace halt state. To resume acquisition, exit from the trace halt state by using the END command. When a subroutine trace has been specified and the specified subroutine has called other subroutines, execution in the called subroutines is not traced. When the sequential trace stop has been specified, the sequential point and a reset point must be set.

- Cancellation

  Cancels the trace conditions (Trace Condition A/B/C), sequential points, or reset points (Trace Condition A/B) setting. When the channel number is omitted, all trace conditions are cancelled. A channel number cannot be set when a hardware sequential condition is in place.

- Display

  Displays the set trace conditions (Trace Condition A/B/C). When the channel number is omitted, all trace conditions that have been set are displayed.

  Display format is as follows:

    Trace Condition Xn: <settings>

    X: A/B/C

    n: Channel number (from 1 to 8)

  The following will be displayed when a hardware sequential condition has been specified.

    Trace Condition X Sequential:<Enable/Disable>

    n <Settings>

    X: A/B

    n: Sequential number (starts from 1 and condition is satisfied at the maximum of 7, or R for a reset point)

[Examples]

1. To set address bus values from H'10027C to H'100304 as the address condition, and the write cycle as the read/write cycle condition of Trace Condition A on channel 2 (range trace):

   TAS A CHANNEL 2 TYPE RANGE ADDRESS H'10027C TO H'100304 DIRECTION WRITE

2. To set a value of H'4750 as the condition for the data bus, and the read cycle as the read/write cycle condition of Trace Condition B on channel 4 (trace stop).

   TAS B CHANNEL 4 TYPE STOP MASK H'*4750 LONG DIRECTION READ***

3. To set a value of H'11111111 as a data condition, and external probe 4 low, external probe 3 high, external probe 2 low, and external probe 1 high, as the external probe conditions, of Trace Condition A on channel 5 (trace stop):

   TAS A CHANNEL 5 TYPE STOP DATA H'11111111 LONG PRB 0101

4. To set the low NMI as the NMI signal condition for Trace Condition B on channel 7 (trace stop):

   TAS B CHANNEL 7 TYPE STOP NMI LOW

5. To set bus value H'1000*** as the address condition for Trace Condition B on channel 1 (trace stop):

   TAS B CHANNEL 1 TYPE STOP ADDRESS MASK H'1000*

6. To set address bus values outside the range from H’100000 to H'1001C0 as the address condition for Trace Condition B on channel 3 (trace stop):

   TAS B CHANNEL 3 TYPE STOP ADDRESS H'100000 TO H'1001C0 NOT

7. To display settings for Trace Condition B on channel 3:

   TAD B CHANNEL 3

```
>TAD B CHANNEL 3
Trace Condition B3: Enable type range H'100000 to H'1001c0 not
```

8. To display all settings for Trace Condition:

   TAD A

```
>TAD A
Trace Condition A1:Disable
Trace Condition A2:Enable  type range address H'10027c to H’100304 direction write
Trace Condition A3:Disable
Trace Condition A4:Disable
Trace Condition A5:Enable  type stop data H’11111111 long prb 0101
Trace Condition A6:Disable
Trace Condition A7:Disable
Trace Condition A8:Disable
```

9. To cancel conditions set as Trace Condition B on channel 2:

   TAC B CHANNEL 2

10. To cancel all settings for Trace Condition A:

    TAC A

11. To set Trace Condition B as a sequential point at an address bus value of H’100000 when a sequential trace stop condition has already been set:

    TAS B CHANNEL P ADDRESS H'100000

12. To set Trace Condition B as a reset point at an address bus value of H’300000 when a sequential trace stop condition has already been set:

    TAS B CHANNEL R ADDRESS H'300000

13. To display the settings when the sequential trace stop condition has been set for Trace Condition B:

    TAD B

    ```
    >TAD B
    Trace Condition B Sequential:Enable
    1 address H'100000
    2 address H'200000
    R address H'300000
    ```

14. To clear the settings when the sequential trace stop condition has been set for Trace Condition B:

    TAC B

[Note]

It is not possible to use numbers to specify the points of a hardware sequential trace. If this is attempted, it sets the condition at the opposite position in the sequence of conditions. To set the sequential conditions again, clear all of the conditions.

6.2.19 TRACE_DISPLAY (TD)

[Command syntax and parameters]

td range <startcycle> **to** <endcycle>

[Description]

Displays trace information.

Parameter	Type	Description
<startcycle>	Numeric	Specifies the first cycle value of the range of trace information to be displayed.
<endcycle>	Numeric	Specifies the last cycle value of the range of trace information to be displayed.

[Example]

To display acquired trace information within the range from –D’1000 to D'0:
TD RANGE –D’1000 to 0

6.2.20 TRACE_MODE (TM)

[Command syntax and parameters]

- Setting **tm** [<dmaopt>] [<tracemode>] [<timeopt>]
- Display **tm**

Parameter	Type	Description
<dmaopt>	Keyword	Specifies whether or not trace information is acquired on DMA/DTC cycles. **dma enable**: Acquired (default). **dma disable**: Not acquired.
<tracemode>	Keyword	Specifies whether a break is generated or tracing is stopped when the trace buffer overflows. **ofoff**: A break is not generated (default). **ofbreak**: A break is generated. **oftrace**: Tracing is stopped.
<timeopt>	Keyword	Specifies the minimum time interval for time stamping of trace information. **time 20ns**: Display is in 20-ns units (default). **time 1.6us**: Display is in 1.6-µs units. **time 52us**: Display is in 52-µs units. **time clk**: Trace information is acquired in units of clock cycles. **time clk2**: Trace information is acquired in units of 1/2 clock cycles. **time clk4**: Trace information is acquired in units of 1/4 clock cycles. **time clk8**: Trace information is acquired in units of 1/8 clock cycles.

[Description]

Specifies trace information acquisition mode and displays the mode settings.

[Examples]

1. To display the set trace information acquisition mode:
   TM

   Results:
   ```
   >TM
   trace_mode dma enable ofoff time 20ns
   ```
2. To stop the acquisition of trace information and break when the trace buffer overflows:
   TM OFBREAK

6.2.21 TRACE_SEARCH (TS)

[Command syntax and parameters]

ts range <startcycle> **to** <endcycle> <option> [<option>…]

<option> = <addropt> | <dataopt> | <accessopt> | <r/wopt> | <prbopt> | <irqopt> | <nmiopt> | <resetopt> | <resetmopt> | <breqopt> | <timeopt>

Parameter	Type	Description
<startcycle>	Numeric	Specifies the start cycle value of the search range.
<endcycle>	Numeric	Specifies the end cycle value of the search range.

Description of the ts command <option>

Parameter	Description
<addropt>	Specifies address condition. To specify an address condition with no special conditions: **type all** <address> To specify an address condition for the program counter: **type pc** <address> To specify an address condition in the emulation area: **type ext** To specify an address condition in the internal ROM area: **type rom** To specify an address condition in the internal I/O area: **type io** To specify an address condition in the internal RAM area: **type ram** Parameter <address> is specified as follows: To specify an address condition as a single address: **address** <value> To specify an address condition as a range of addresses: **address** <value1> **to** <value2> To specify an address value with masking: **address mask** <maskdata> <value>: Address (numeric) <value1>: Start address (numeric) <value2>: End address (numeric) <maskdata>: Specifies mask data.

Parameter	Description
<dataopt>	Specifies data condition. To specify an 8-bit data trace search condition: **data** <data> **byte** To specify a 16-bit data trace search condition: **data** <data> **word** To specify a 32-bit data trace search condition: **data** <data> **long** <data>: Specify data value (numeric) Mask specification is as follows: To specify an 8-bit data trace search condition (mask specification): **data mask** <maskdata> **byte** To specify a 16-bit data trace search condition (mask specification): **data mask** <maskdata> **word** To specify a 32-bit data trace search condition (mask specification): **data mask** <maskdata> **long** <maskdata>: Specifies the mask data. To search for trace information on data of size smaller than a byte, specify a mask.
<accessopt>	Specifies a bus status condition. To search tracing with the DMA/DTC cycle: **access dma** To search tracing with the CPU data access cycle: **access dat** To search tracing with the CPU instruction fetch cycle: **access prg**

Parameter	Description
<r/wopt>	Specifies read or write condition. To search through the trace information for read cycles: **direction read** To search through the trace information for write cycles: **direction write**
<prbopt>	Specifies an external probe signal condition. To specify an external probe signal condition: **prb** <bit specification> A bit is specified as follows: PRB1 to PRB4 signal bit specification 3 2 1 0 : Bit location x x x x : Value to be specified (Specify 0 (low level) or 1 (high level) for x.) | | | | 4 3 2 1 :PRB number When * is specified, the condition can be specified not to include the external probe signal condition for bit location.
<irqopt>	Specifies IRQ0 to IRQ7 signal conditions. To specify IRQ0 to IRQ7 signal conditions: **irq** <bit specification> A bit is specified as follows: To specify bits IRQ0 to IRQ7 7 6 5 4 3 2 1 0 : Bit location x x x x x x x x : Value to be specified (Specify 0 (low level) or | | | | | | | | 1 (high level) for x.) 7 6 5 4 3 2 1 0: IRQ number When * is specified, the condition can be specified not to include the IRQ condition for bit location.
<nmiopt>	Specifies an NMI signal condition. To search through the trace information when the NMI signal is high: **nmi hi** To search through the trace information when the NMI signal is low: **nmi low**
<resetopt>	Specifies a RES signal condition. To search through the trace information when the RES signal is low: **res low**

Parameter	Description
<resetmopt>	Specifies a MRES signal condition. To search through the trace information when the MRES signal is low: **mres low**
<breqopt>	Specifies a BREQ signal condition. To search through the trace information when the BREQ signal is low: **breq low**
<timeopt>	Searches through the trace information for the specified period of time. To check the trace information at a specific time: **time** <time1> To search through the trace information for a specified range of time: **time** <time1> **to** <time2> Specify the start of the period as <time1>, and the end of the period as <time2>. Specify the search time as follows: hhh[:mm[:ss[:uuuuuu]]] hhh: Hours (numeric, from 0 to 999) mm: Minutes (numeric, from 0 to 59) ss: Seconds (numeric, from 0 to 59) uuuuuu: Microseconds (numeric, from 0 to 999999)

[Description]

Searches for trace information that satisfies the specified conditions and displays the information for bus cycles on which it was acquired. The search is in the range specified by <startcycle> and <endcycle>. For conditions that can be specified, see the description of <option>.

[Examples]

1. To display trace information with the address bus in the range from H'10027C to H'100304, or the write cycle for the last five instructions of acquired trace information:
   TS RANGE –D'5 TO 0 TYPE ALL ADDRESS H'10027C TO H'100304 DIRECTION WRITE
2. To display trace information with the data bus value at H'4750, or the read cycle, for the last five instructions of acquired trace information:
   TS RANGE –D'5 TO 0 DATA MASK H'*4750 LONG DIRECTION READ***

Section 7 Error Messages

7.1 Emulator Error Messages of the Emulator

The emulator system program displays error messages in the format below if an error occurs during emulation command execution. Table 7.1 lists error messages.

Table 7.1 Error Messages

Error Message	Description and Solution
INVALID COMMAND	The specified command is invalid, or this command cannot be executed in trace halt mode. Correctly enter the command.
INSUFFICIENT MEMORY	The size of emulation memory to be allocated with the MAP_SET command is not available. Emulation memory has been allocated within the available memory size instead.
CANNOT USE THIS MODE	The GO command cannot be executed because settings for the execution mode are invalid. Correctly specify the settings necessary for the specified execution mode.
SET POINT IS NOT IN RAM	A write-inhibited address is specified by the BREAKPOINT or BREAKSEQUENCE_SET command. Specify a correct address.
CANNOT RECOVER A = xxxxxxxx	The break instruction at the address (xxxxxxxx) where a breakpoint is specified with the BREAK or BREAKSEQUENCE_SET command could not be recovered after GO command execution is terminated. Accordingly, a break instruction remains at the breakpoint address. A hardware error might have occurred. Correct the error, and reload and re-execute the program.
NOT FOUND	The specified data or information was not found. Correctly specify data.
INTERNAL I/O AREA	The internal I/O area was accessed. To display or modify the internal I/O register, use the [I/O Registers] window.
ILLEGAL INSTRUCTION ADDRESS	The memory contents of the address specified with the BREAK or BREAKSEQUENCE_SET command is a break instruction (H'003B). A breakpoint cannot be specified at this address.

Table 7.1 Error Messages (cont)

Error Message	Description and Solution
OUT OF CS AREA ADDRESS	An attempt has been made to allocate emulation memory to an area other than CS0 to CS6. The emulation memory has been allocated within the available area.
MAPPING BOUND MUST BE IN 128KB UNITS	Memory has been allocated in 128-kbyte unit with the MAP command. For details, refer to the MAP_SET command.
MAPPING BOUND MUST BE IN 1MB UNITS	Memory has been allocated in 1-Mbyte unit with the MAP command. For details, refer to the MAP_SET command.
BREAK POINT IS DELETED A = xxxxxxxx	A software breakpoint specified at the displayed address has been canceled because the contents of the software breakpoint are modified with the user program.
CANNOT SET A = xxxxxxxx	A breakpoint cannot be specified at the displayed address by the BREAKPOINT or BREAKSEQUENCE_SET command before GO command execution. A hardware error might have occurred or the contents of the memory address might be a break instruction. Correct the error, and reload and re-execute the program.
EMULATOR BUSY	The emulator has been processing a command of trace halt mode simultaneously with continuous processing of a software breakpoint, so another command cannot be executed. Re-enter the command.
TRACE CONDITION RESET	Satisfied trace conditions are all reset when trace halt mode is entered. When trace halt mode is terminated, the trace conditions are rechecked from the beginning.
VERIFY ERROR ADDRESS H'xxxxxxxx WRITE:H'xx READ:H'xx	A verify error occurred (xx: numeric).
RUN-TIME OVERFLOW	Indicated with the result of performance analysis when the timeout or excessive number specification is satisfied in Performance Analysis 1 mode.
DOUBLE DEFINITION	Software break or software sequential break is specified for the same address.
INVALID ADDRESS	An illegal address was specified.

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 7.2 lists these error messages, descriptions of the errors, and solutions to the errors.

Table 7.2 HDI Error Messages

Error Message	Description and Solution
Addresses are few	Too few points have been specified for a software sequential break. Specify at least two addresses.
Can not set the minimum time.	Since the performance information has already been acquired, the minimum time interval is impossible to change. Change the performance information after reinitialization.
Command not ready	The function that was entered for execution is not available now.
Internal I/O Area	Loading of program to the I/O area was attempted.
Invalid address	The specified address is invalid.
Invalid mask value	The specified mask value is invalid.
Invalid value Invalid command syntax Invalid value or symbol:xx	The specified value or symbol is invalid.
No condition data	No condition has been specified. Specify one or more valid conditions.
Program counter is odd	Since the program counter holds an odd value, execution of the program is not possible.
Stack pointer is odd	Since the stack pointer holds an odd value, execution of the program is not possible.
The E8000S emulator cannot be started correctly. Change the CPU Operating Mode.	The emulator was illegally started. Change the CPU's operating mode.
Verify error at H'x wrote H'x, read H'x	Verification failed at H'x.
Cannot use command when user program executing	A command line was input while it was not possible to issue commands to the emulator. Wait for the completion of processing.
Command timeout	The emulator did not respond after a command was issued from the HDI, so a timeout occurred in the HDI. Exit from the HDI, turn on the power supply of the emulator, and restart the HDI.

Table 7.2 HDI Error Messages (cont)

Error Message	Description and Solution
Emulator Command send/receive check Error	Illegal communication between the HDI and the emulator during HDI start up. Exit from the HDI, turn on the power supply of the emulator, and restart the HDI. If this does not solve the problem, contact Hitachi's sales department or the agent through whom you purchased the HDI.
Emulator firmware not ready.	A message EMULATOR FIRMWARE NOT READY has been received from the emulator. Exit from the HDI and check that the emulator is operating correctly.
Emulator Timeout.	A timeout message has been detected from the emulator. Exit from the HDI and check that the emulator is operating correctly.
Failed to find matching trace record	The searching for the trace information has failed because no information is currently displayed in the [Trace] window. This error message is also output when there is no trace information.
Hardware register read/write check error	When the HDI was started up, an error was detected during checking of the hardware registers of the emulator. Exit from the HDI and check that the emulator is operating correctly.
Invalid version number in target configuration	The HDI version when the session file was created and the current HDI version differ. Do not use a session file for an earlier version.
System ID error	An emulator different from that selected in the [Select Platform] dialog box is currently connected. Check whether the correct emulator is in use.
Target internal error	A command cannot be issued to the emulator. Wait for the completion of processing.
User system not ready	The message USER SYSTEM NOT READY was received from the emulator.
Function information not found	The function information that was entered in the [Input Function Range] dialog box matches no function. Input a correct function name.
function not available Currently not available Command currently not available	The requested function is not available now.
Not support	The requested function is not supported.

Table 7.2 HDI Error Messages (cont)

Error Message	Description and Solution
Can’t add this item because there is not enough Auto-update memory resource.	The number of Auto-update Memory settings reached the maximum (eight points) and the new item cannot be added. Change or cancel items that have already been set.
The AUM setting range is in the 32 byte boundary.	The address for an Auto-update Memory item was set so that the range runs the 32-byte boundary. An address range cannot run across a 32-byte boundary.
trace mode data no change	The contents of the [Trace Mode] page have not been changed. Set a condition and click the [Apply] button.
Invalid parameter	An illegal command parameter was input.
Types of the user cable and the device (the number of pins of the package) selected in Cpu Operating Mode are not matched.	The type of connected user interface cable and the number of MCU pins set in the [Cpu Operating Mode] dialog box do not match. Check the setting in the [Cpu Operating Mode] dialog box.
The type of the user cable is illegally used with the MD pin. Start the E8000S emulator in clock mode 0 or single chip mode.	The combination of the type of connected user interface cable and the settings on the MD pins of the user system is illegal. Start the emulator in clock mode 0 or single chip mode. Check that the MD pin values of the user system are set correctly.
A session file cannot be loaded because the value of the MCU operating mode when a session file was saved and the value of the current MCU operating mode are different.	To load a session file correctly, the value of the MCU operating mode when a session file was saved and the value of the current MCU operating mode must be matched.

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

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: 40.0 MHz)

	MCU Specifications (ns)		Values with Emulator Connected (ns)	
Item	**Min**	**Max**	**Min**	**Max**
tRDS	15	—	30	—
tACC	tcyc x (n + 2) – 39 (n is the number of waits)	—	tcyc x (n + 2) – 40 (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.

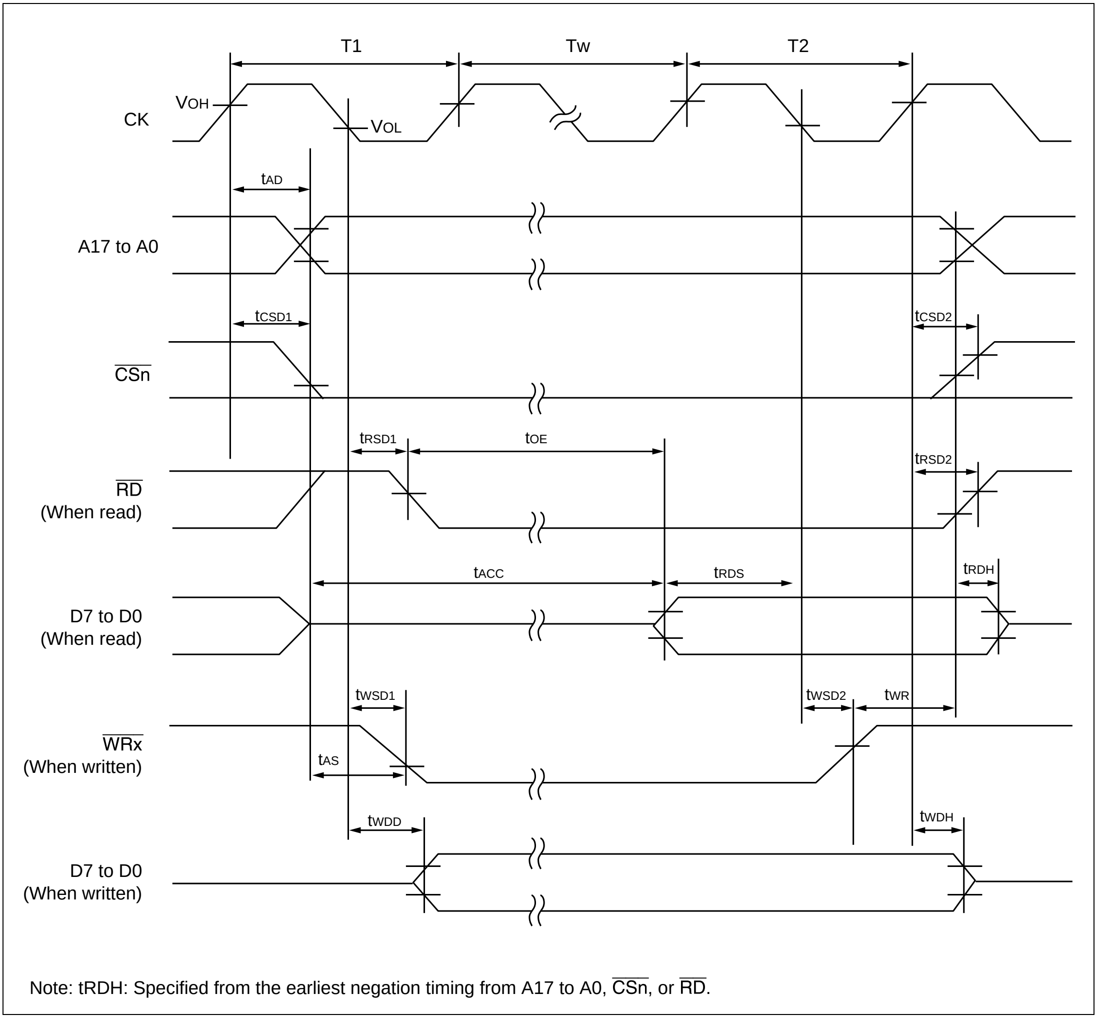


Figure A.1 Basic Bus Cycle (Software Wait)

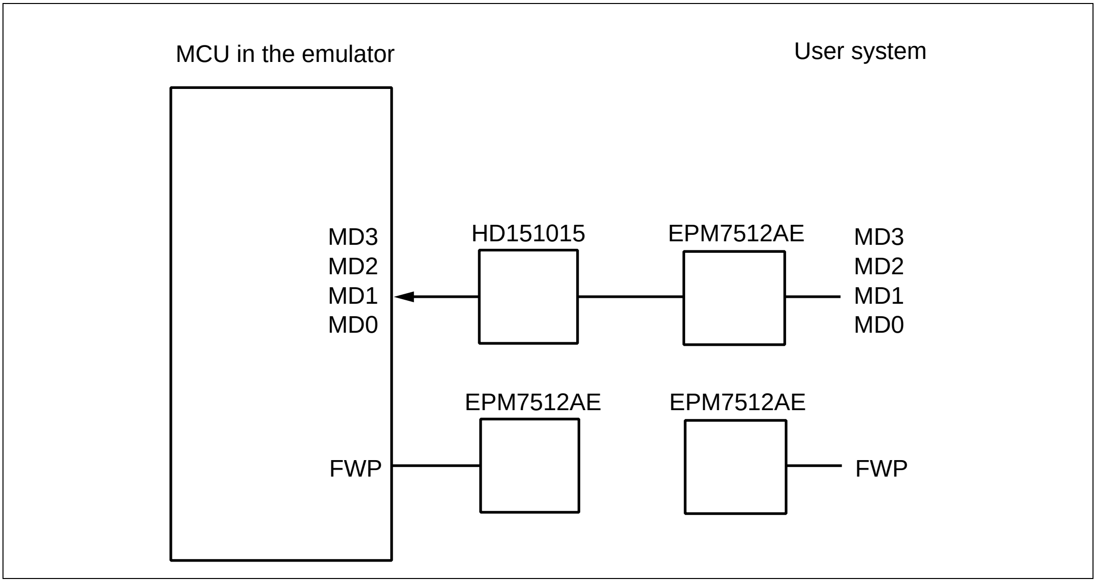


Figure A.2 User System Interface Circuits (1)

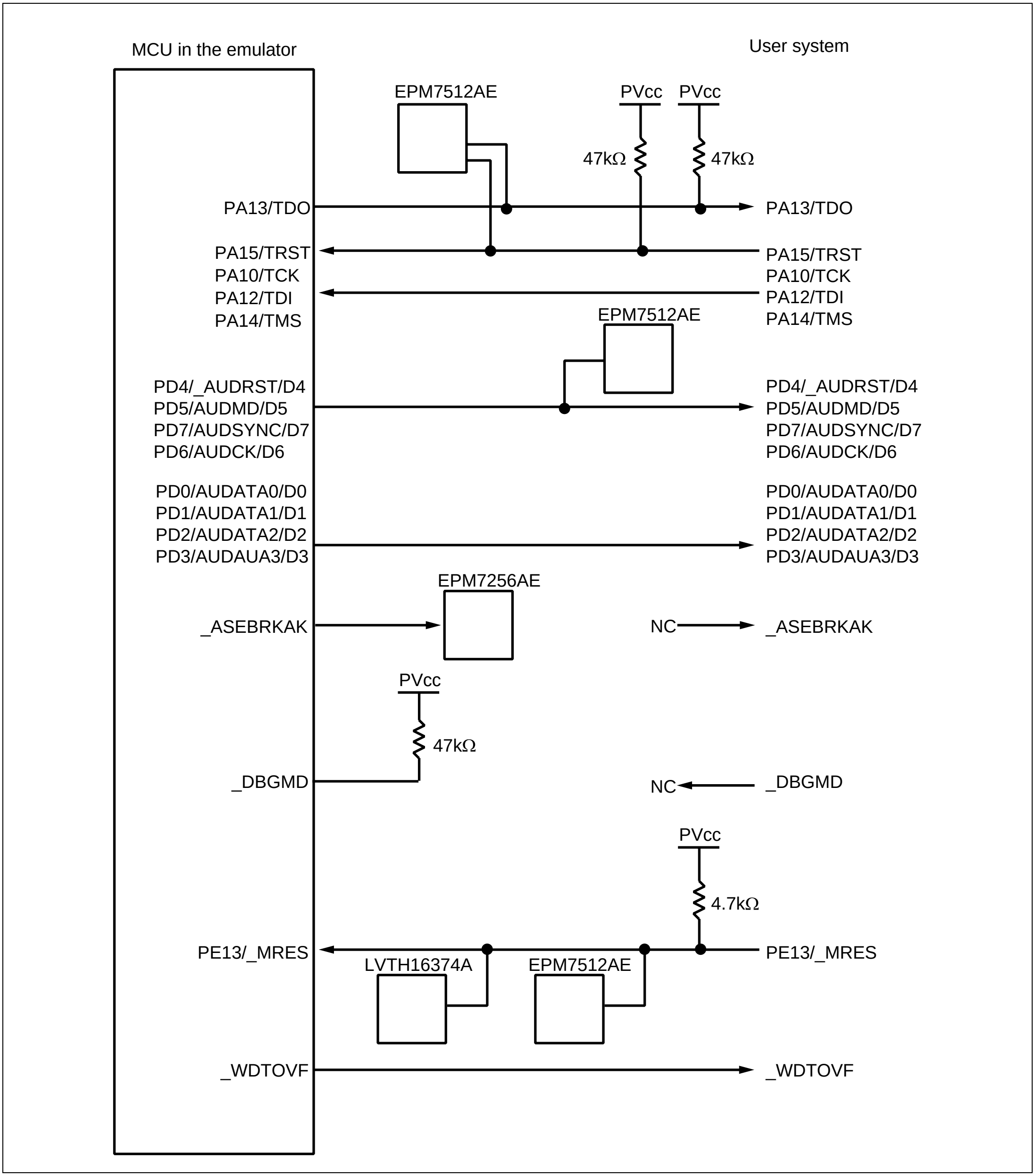


Figure A.2 User System Interface Circuits (2)

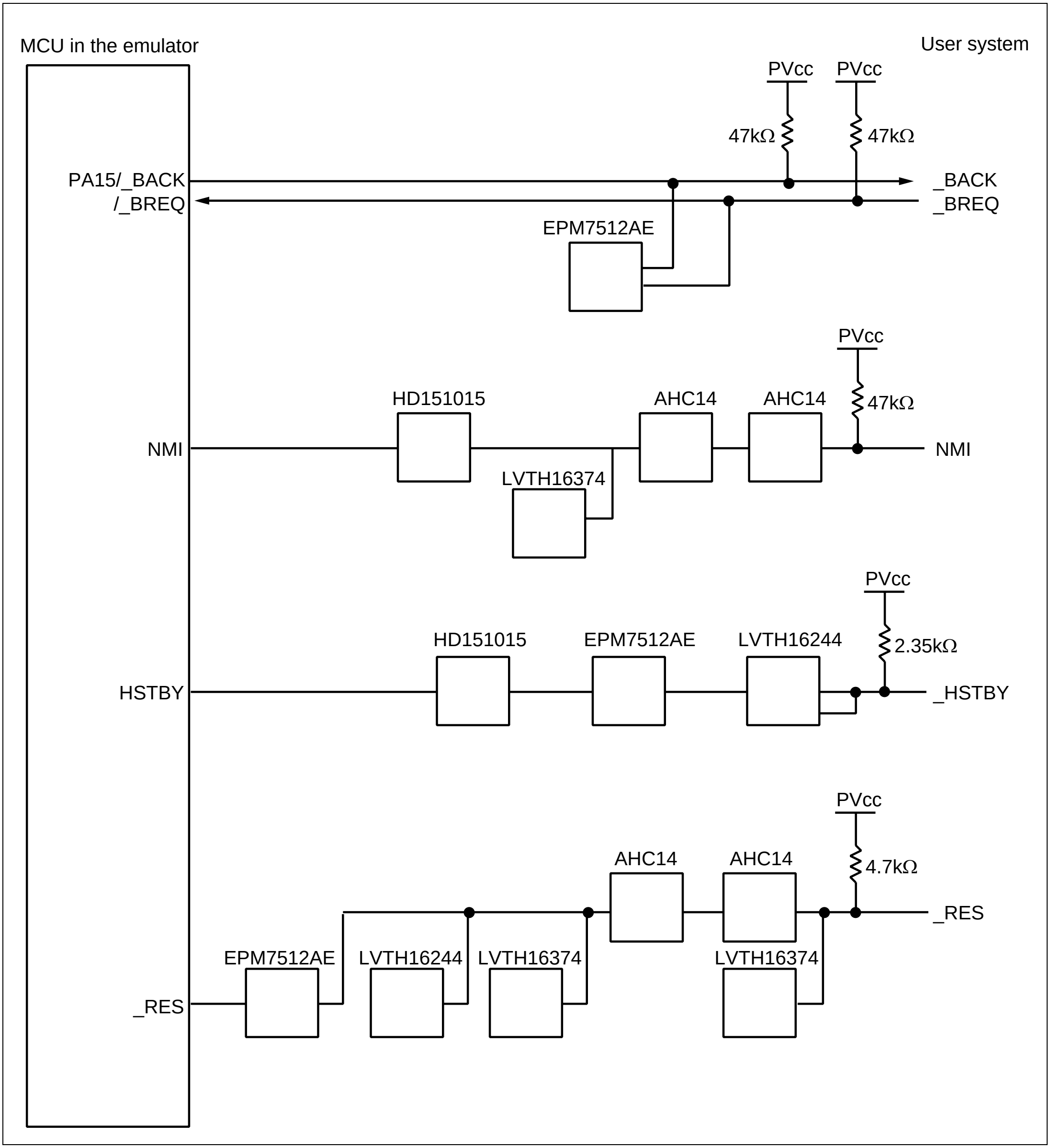


Figure A.2 User System Interface Circuits (3)

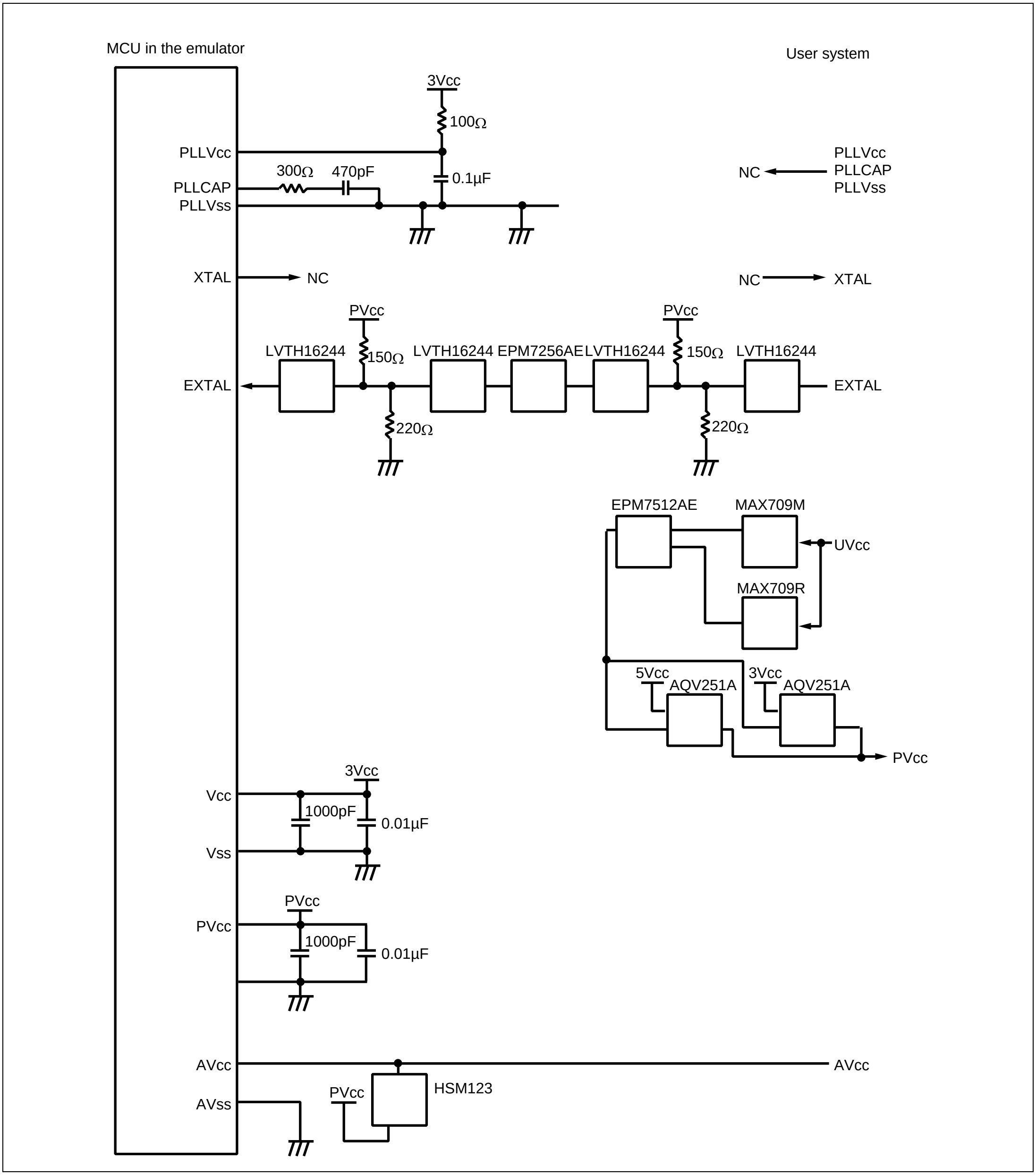


Figure A.2 User System Interface Circuits (4)

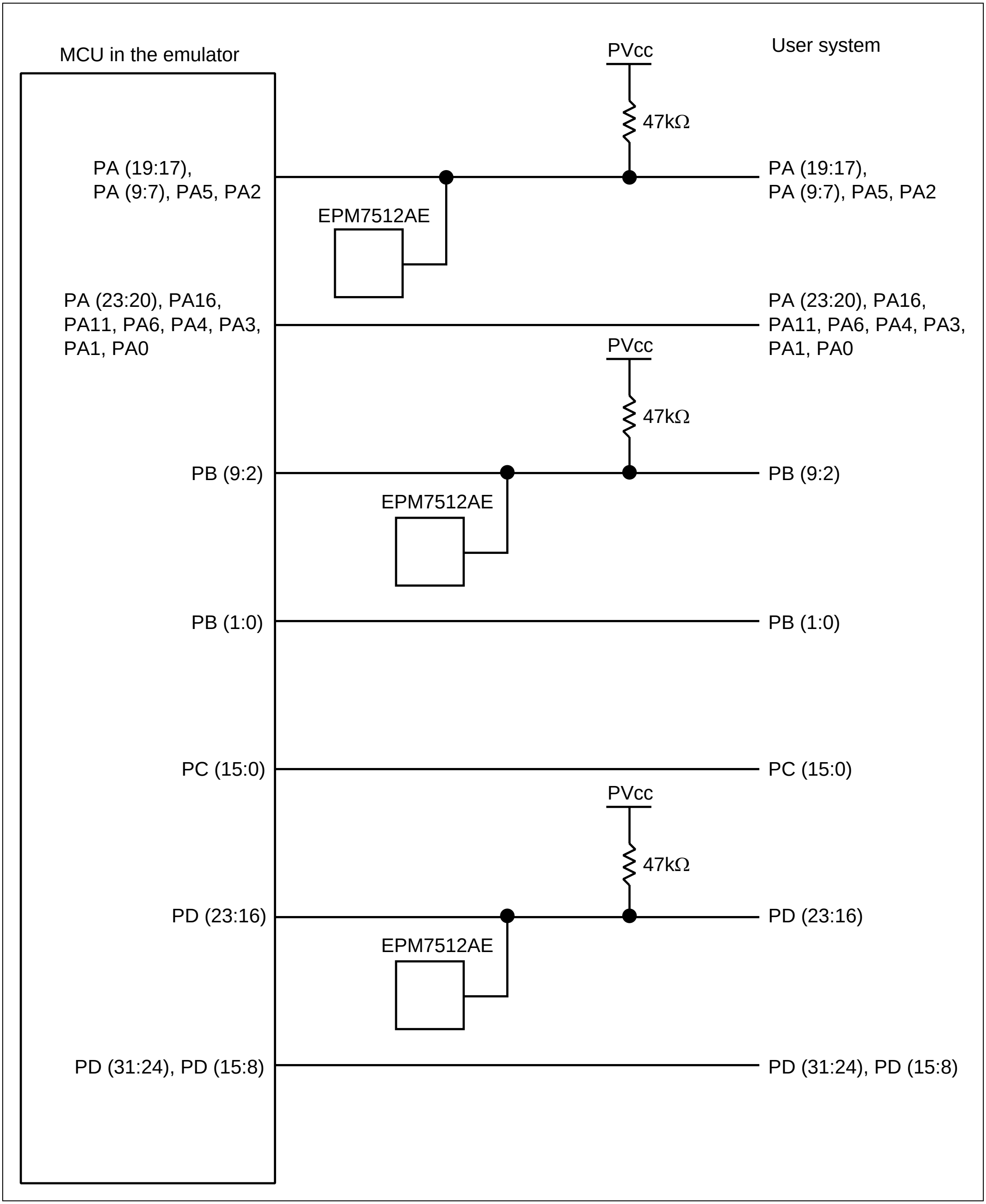


Figure A.2 User System Interface Circuits (5)

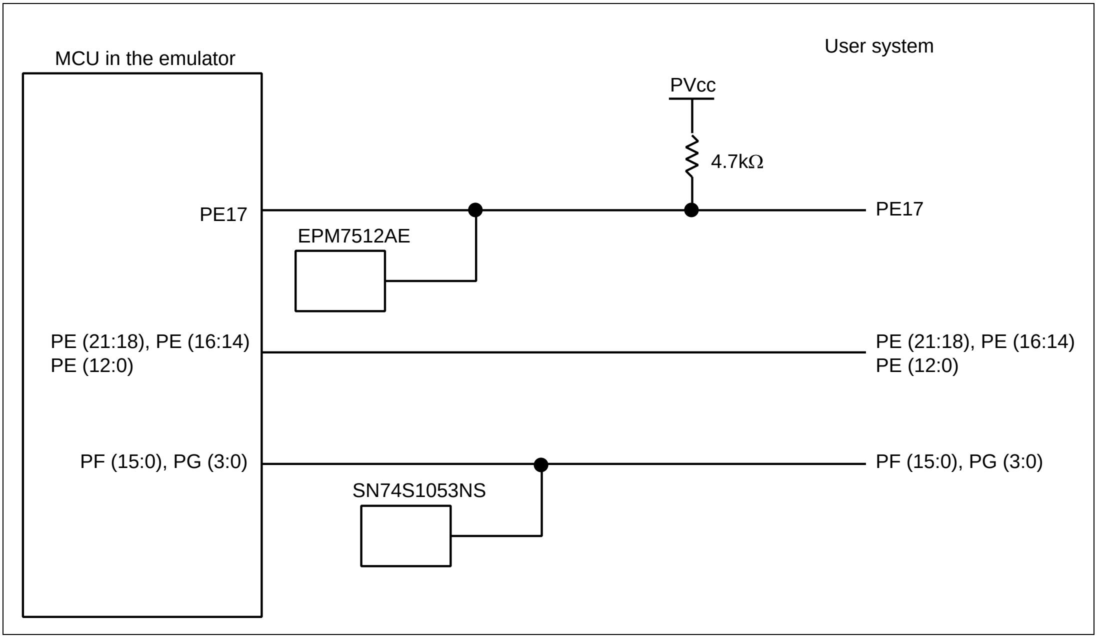


Figure A.2 User System Interface Circuits (6)

Appendix B Emulator External Dimensions and Mass

Figures B.1 and B.2 show the external dimensions and mass of the E8000S station and evaluation chip board, respectively.

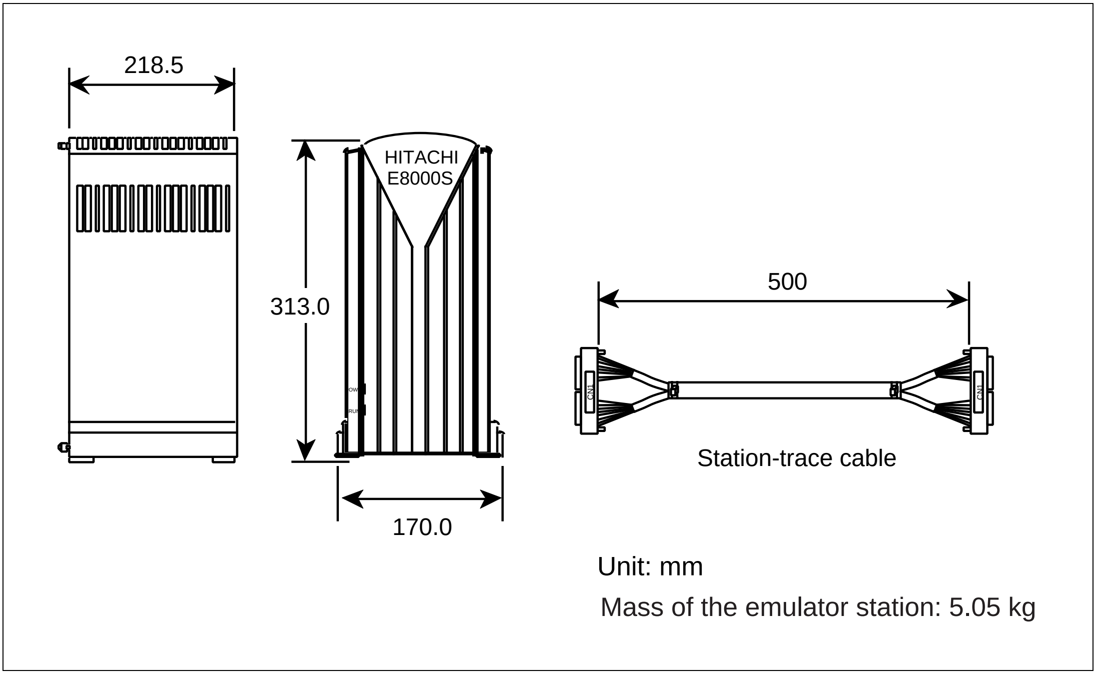
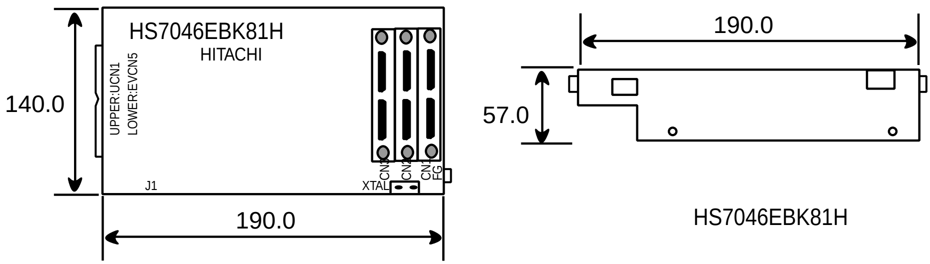


Figure B.1 External Dimensions and Mass of the Emulator



Unit: mm

Mass of the evaluation chip board: HS7046EBK81H: 0.950 kg

Figure B.2 External Dimensions and Mass of the Evaluation Chip Board

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 cable.

Table C.1 User System Interface Cable and User Interfaces

User System Interface Cable	User Interface
HS7046ECH81H	FP-80Q (IC149-080-017-B5)
HS7047ECH81H	FP-100M (IC149-100-054-B51)

Note: The IC149 series is manufactured by YAMAICHI ELECTRONICS Co., Ltd.

C.1.1 Installing IC Socket

1. Installing IC Socket

Install the IC socket for an FP-80Q 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 HS7046ECH81H

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 HS7046ECH81H, refer to the user's manual supplied with the evaluation chip board.

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

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

Figure C.1 shows the connection of the HS7046ECH81H, figure C.2 shows the size restrictions for the installed components of the HS7046ECH81H, and figure C.3 shows the recommended mount pad dimensions of the user system IC socket.

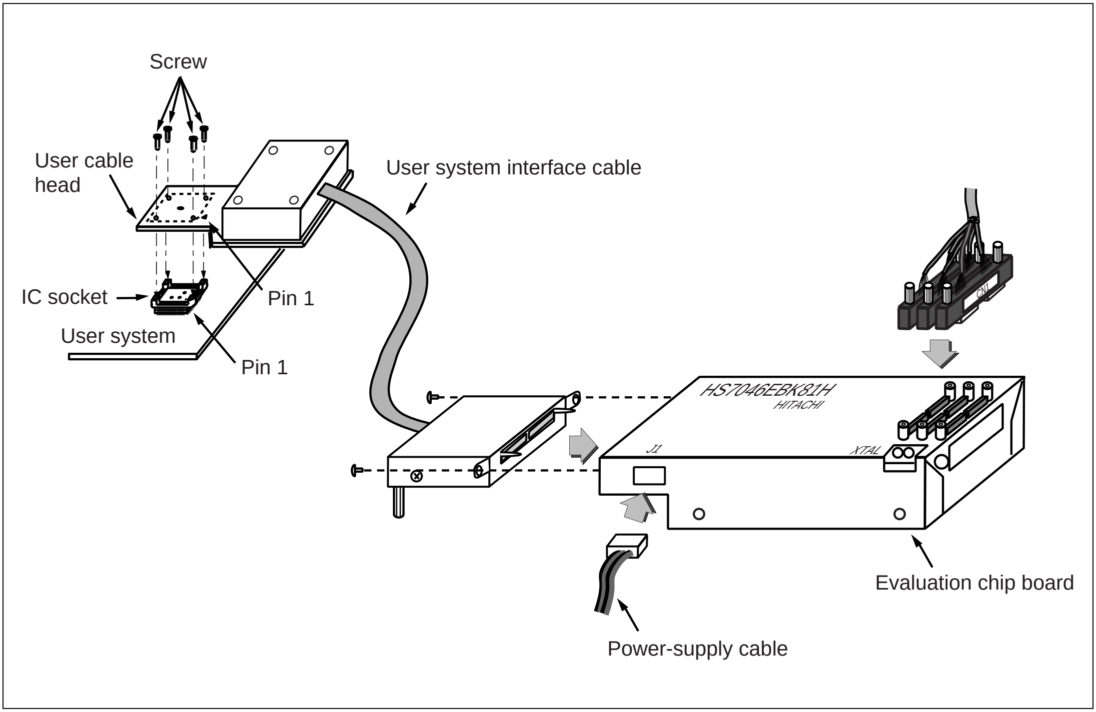


Figure C.1 Connection Using the HS7046ECH81H

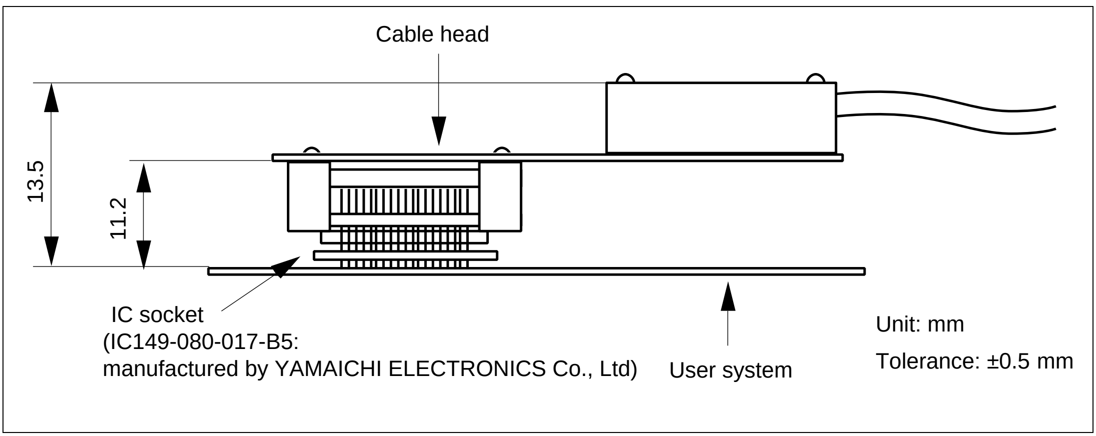


Figure C.2 Restrictions on Component Installation

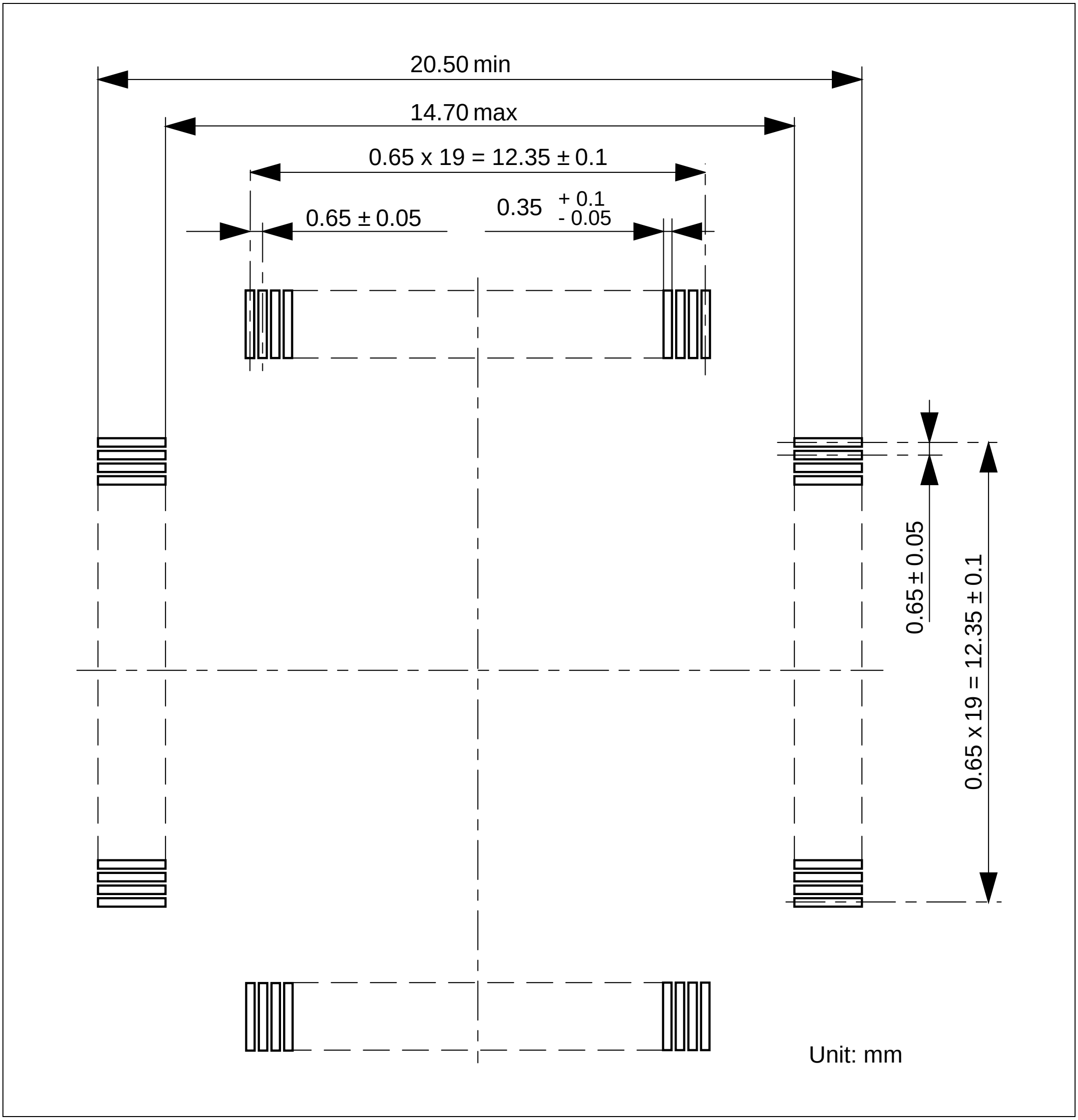


Figure C.3 Recommended Mount Pad Dimensions of the User System IC Socket

C.1.3 Connection Using the HS7047ECH81H

⚠ 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 HS7047ECH81H, refer to the user's manual supplied with the evaluation chip board.
2. This evaluation chip board can only be used in combination with the specified QFP socket (IC149-100-054-B51).

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

Figure C.1 shows the connection of the HS7047ECH81H, figure C.2 shows the size restrictions for the installed components of the HS7047ECH81H, and figure C.3 shows the recommended mount pad dimensions of the user system IC socket.

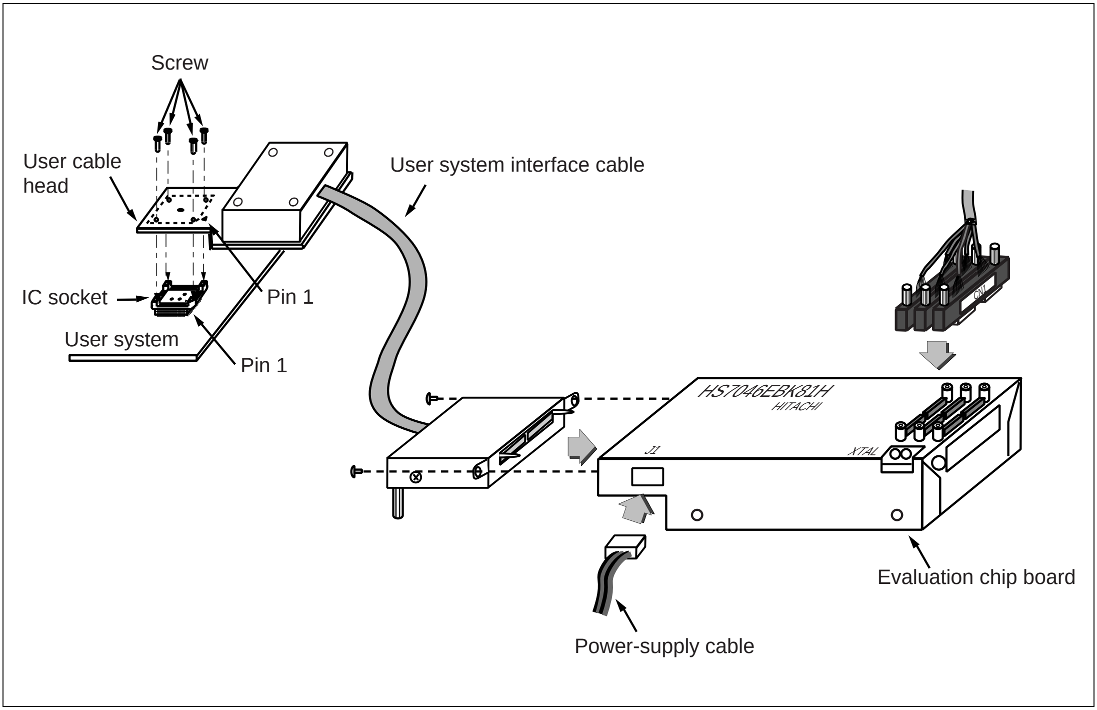


Figure C.4 Connection Using the HS7047ECH81H

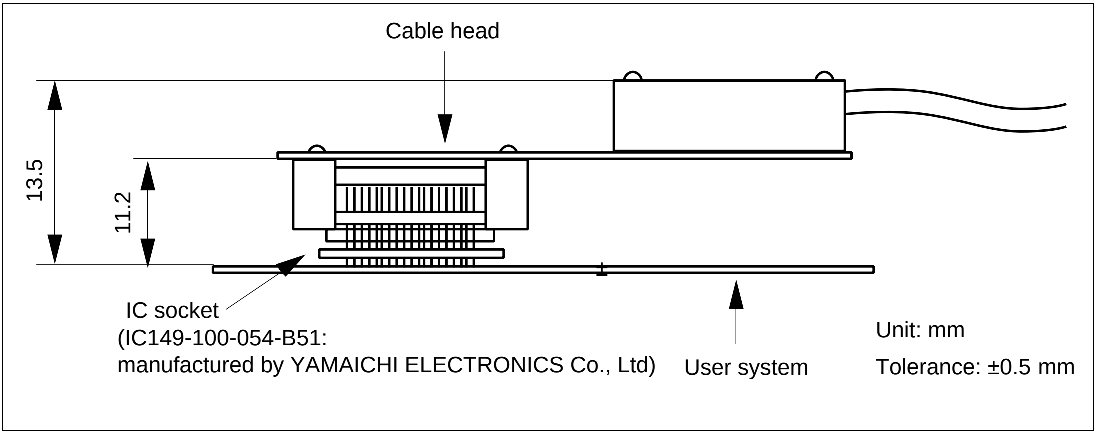


Figure C.5 Restrictions on Component Installation

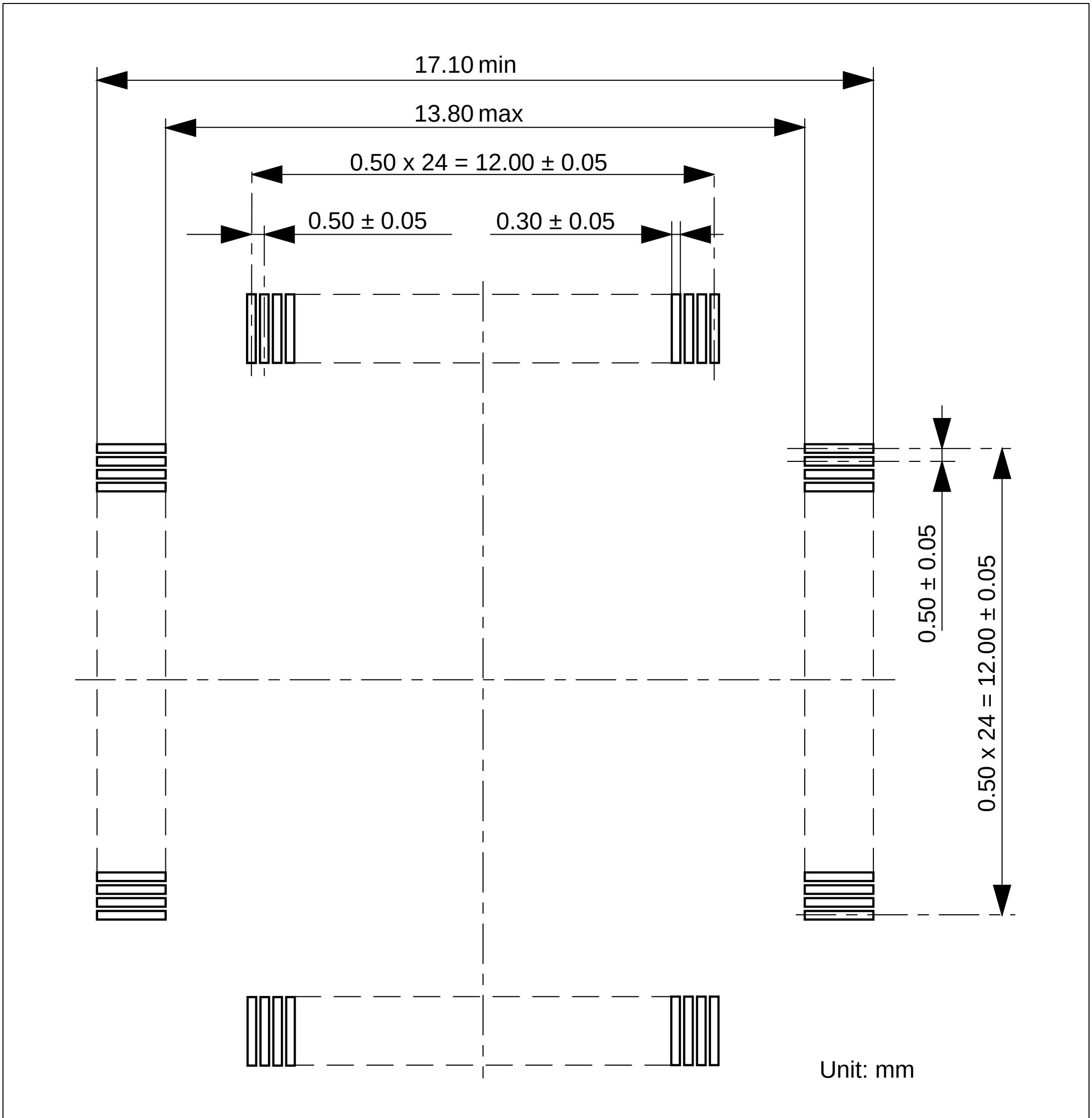
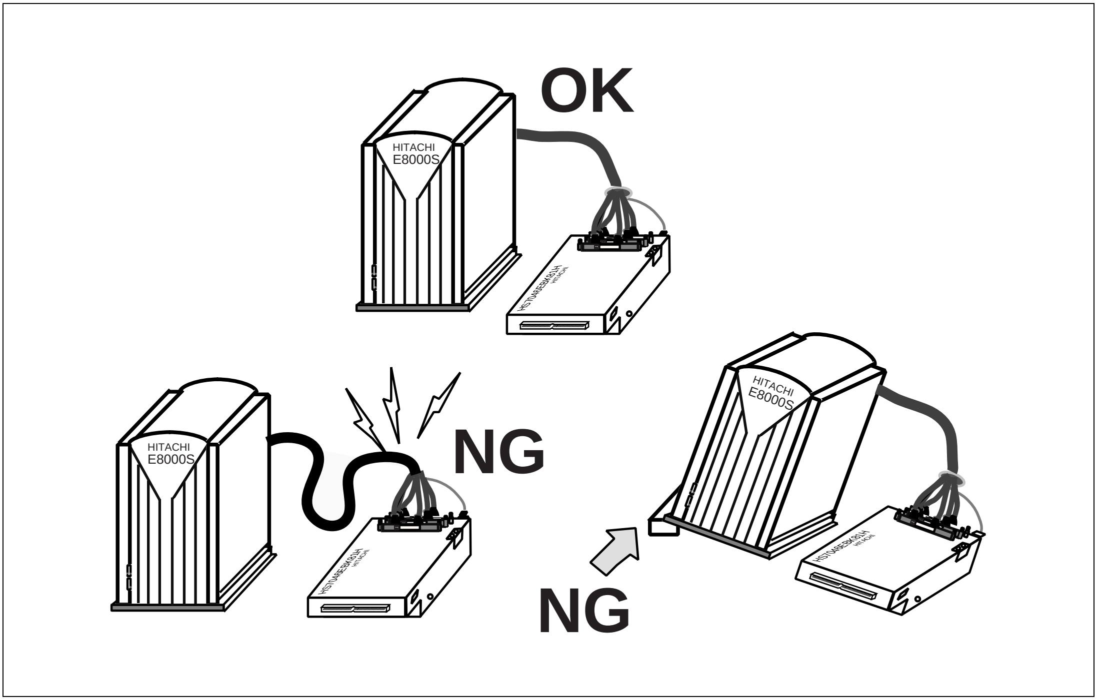


Figure C.6 Recommended Mount Pad Dimensions of the User System IC Socket

C.2 Precautions for User System Connection

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

1. Secure the E8000S station location.
   Place the E8000S 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 E8000S station is placed in a secure position so that it does not move and impose stress on the user interface during use.



2. Make sure the power supply is off.
   Before connecting the evaluation chip board to the user system, check that the emulator and the user system are turned off.
3. Connect the UVcc to the user system power.
   The emulator monitors and determines whether the user system is turned on or off by the UVcc pin (pin 167 or 175 on UCN1 for HS7046EBK81H). Accordingly, after connecting the user system to the emulator, be sure to supply power to the UVcc pin. Otherwise, the emulator assumes that the user system is not connected.

C.3 Connecting the Cables for Tracing

WARNING

Always switch the emulator and user system OFF 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. Also, the USER PROGRAM being debugged will be LOST.

C.3.1 Connecting the cables for tracing (trace cables) to the emulator

Connect the trace cables to the emulator, before connecting it to the HS7046EBK81H board.

CAUTION

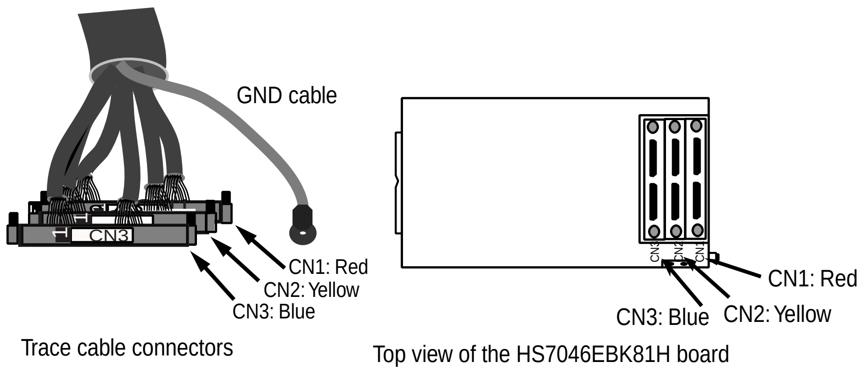
The ends of the trace cables, CN1, CN2, and CN3, which are to be connected to the emulator are bundled together at the time of shipment so that the cables are not inserted into the wrong places. The cables are also bundled together at their other ends, which are to be connected to the evaluation chip board. For details on the connection of cables to the emulator, see 3.3.2, Connecting the Evaluation Chip Board.

C.3.2 Connecting trace cables to the evaluation chip board

Connect the cables to the evaluation-chip board’s connectors. Make sure that the names on the trace cables, emulator, and E8000S station to evaluation chip board interface connectors (CN1, CN2, and CN3) all match. The trace cables and evaluation-chip board’s interface connectors are color-coded (red for CN1, yellow for CN2, and blue for CN3) to prevent incorrect connection.

CAUTION

The screws for fixing the trace cables and evaluation chip interface connectors in place are also color-coded as shown below to reduce the risk of their being inserted into the wrong places.



Connect the cables to the connectors correctly by holding the HS7046EBK81H board by hand so that the connector is flush with the board, and then screw the cables firmly in place.

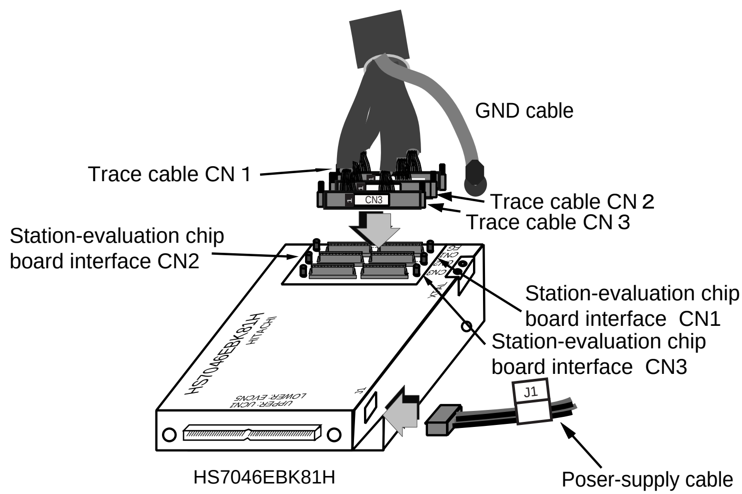


Figure C.7 Connecting the Trace Cables to the Evaluation Chip Board

WARNING

Before connecting the trace cables to the evaluation-chip board's interface connectors, make sure that the numbers match and that they are correctly aligned.

Connecting the cables and connectors with different numbers will result in a FIRE HAZARD.

C.4 Installing the MCU on the User System

Install the MCU in the IC socket after checking the location of pin 1, as shown in figures C.8 and C.9, then use four screws (FP-80Q: M2.0 × 6 mm, FP-100M: M2.6 × 6 mm) to fix the cover to the IC socket. Hold the soldered part of the IC socket in place by hand to prevent rotational pressure due to screwing.

CAUTION

1. **Use the dedicated driver which is attached.**
2. **The torque for screwing must be 0.299 N•m or less.**
3. **If a controlled-torque is not possible, stop screwing as soon as the pressure required changes abruptly. Excessive pressure on the screws will damage the threads of the sockets or cause contact failures by cracking the solder on the IC socket.**
4. **Crack in the soldered connection of the IC socket will result in the emulator malfunction. In that case, check for the conduction by using, for example, a tester, and solder that part again if necessary.**

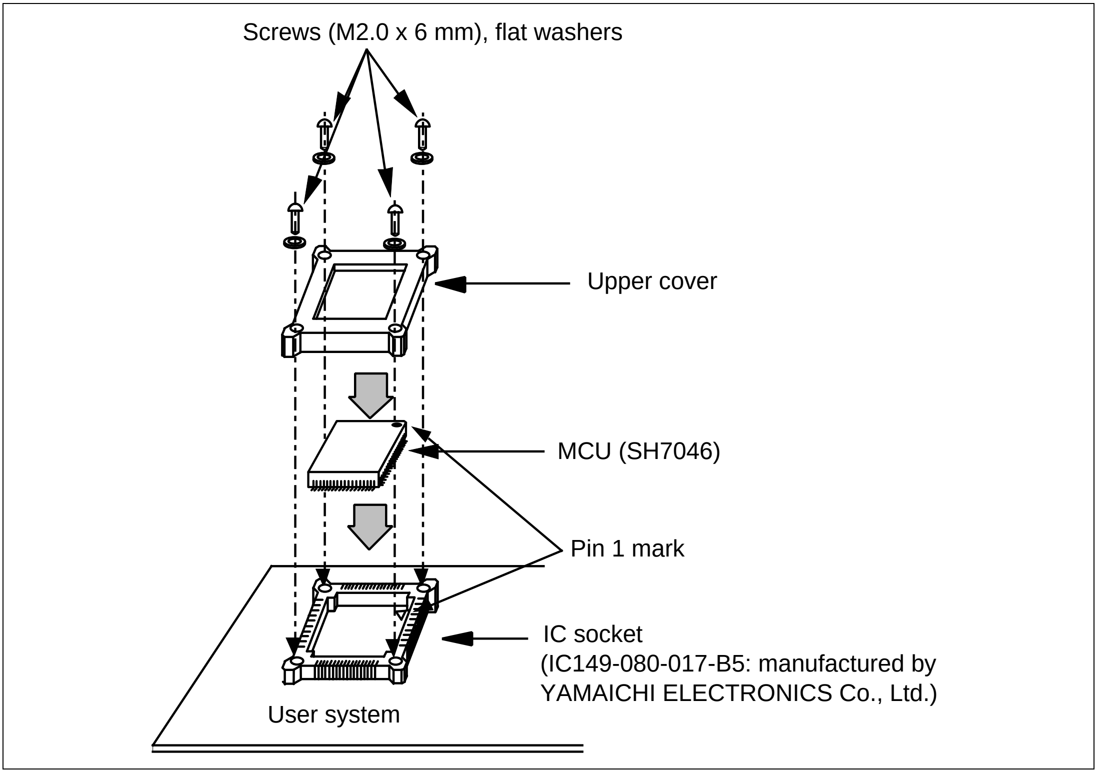


Figure C.8 Installing the MCU (FP-80Q)

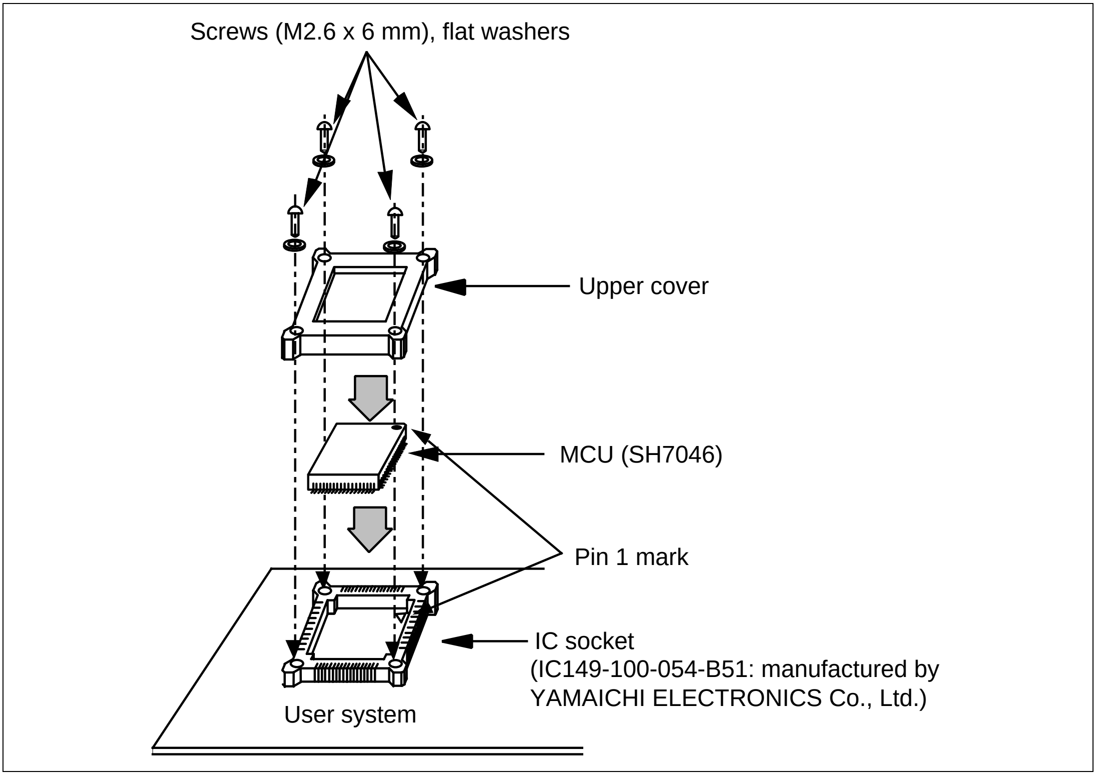


Figure C.9 Installing the MCU (FP-100M)

Appendix D MCU Internal Module Support

D.1 Memory Space

The MCU has a 4-Gbyte memory space in its architecture. Standard emulation memory (4 Mbytes) can be set in 1-Mbyte or 128-kbyte units to the memory space. For details, refer to section E.2, Emulation Memory.

D.1.1 Internal ROM Area

- Access to the internal ROM area

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

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

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

The internal ROM area is accessed in one state.

- Flash memory

The emulator does not support the flash memory.

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.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.8, Displaying and Modifying the Contents of Memory.

D.3 Interrupts

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

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 RES, BREQ, or WAIT signals 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 Bus State Controller

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

D.6 A/D Converter

The A/D converter has AVcc, AVss, and ADTRG pins as well as the analog input pins. Because 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. Because 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 SH7046-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)
WDT (watchdog timer)	No	Yes
MTU (multifunction timer-pulse unit)	Yes	Yes
MMT (motor management unit)	Yes	Yes
CMT (compare-match timer)	Yes	Yes
SCI (serial communication interface)	Yes	Yes
DTC (data transfer controller)	Yes[*1]	Yes
HCAN2 (Hitachi controller area network 2)	Yes	Yes
UBC (user break controller)	No	Yes
AUD (advanced user debugger)	Yes	Yes
I/O port	Yes	Yes
A/D converter	Yes	Yes
H-UDI (Hitachi user debugging interface)	Not available[*2]	Not available[*2]

Notes: 1. If a break occurs during a DTC cycle (vector read, read/write of transferred information, or data read/write), the DTC continues operation until the DTC cycle is complete. The DTC resumes operation after it returns to emulation.

2. The user cannot use the H-UDI.

Appendix E Notes on Debugging

E.1 Notes on HDI

E.1.1 Memory Test Function

This product does not support the memory test function, which is used by selecting [Test] from the [Memory] menu.

E.1.2 Source-level Execution

- Step

  Even standard C libraries are executed. To return to a higher-level function, use the step out function. In a for statement or a while statement, executing a single step does not move execution to the next line. To move to the next line, execute two steps.

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 characters 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 display. 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
  ```

- Overwrite 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\\HDI5\\E8000S\\TUTORIAL\\SORT.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.

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 files (SH7046.IO, SH7047.IO, and CUSTOM.IO) provided with this product do 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 Profiler Function

This product does not support the profiler function.

E.1.12 Usage with Another Version of 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 version of HDI

  If another version of HDI is uninstalled after this HDI has been installed, the Auto update Memory and stack trace functions will not be used. In this case, reinstall this HDI.

E.1.13 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.1.14 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 Emulation Memory

1. The emulator manages areas in the memory blocks shown in figures E.1 and E.2. Emulation memory and user memory cannot coexist in a single block.
2. When emulation memory is used in 50-MHz operation, one or more cycles are required as wait-state cycles for access to those areas to which emulation memory is allocated. When emulation memory is used, refer to the table below and set the number of waits for the bus-state controller.

External Operating Frequency	Number of Required Waits
50 MHz	One or more
40 MHz or lower	0

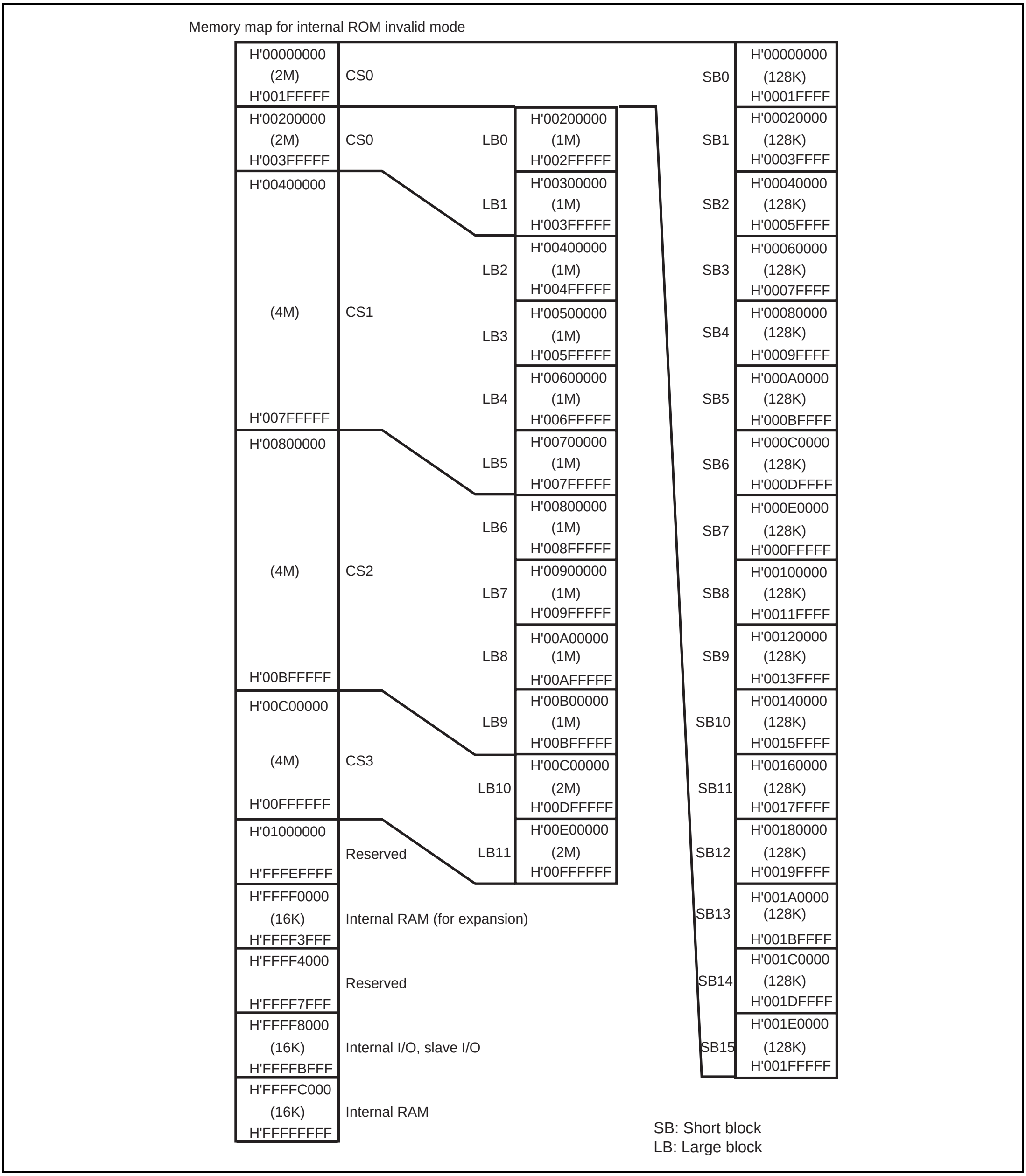


Figure E.1 Memory Map in the Invalid Internal ROM Mode

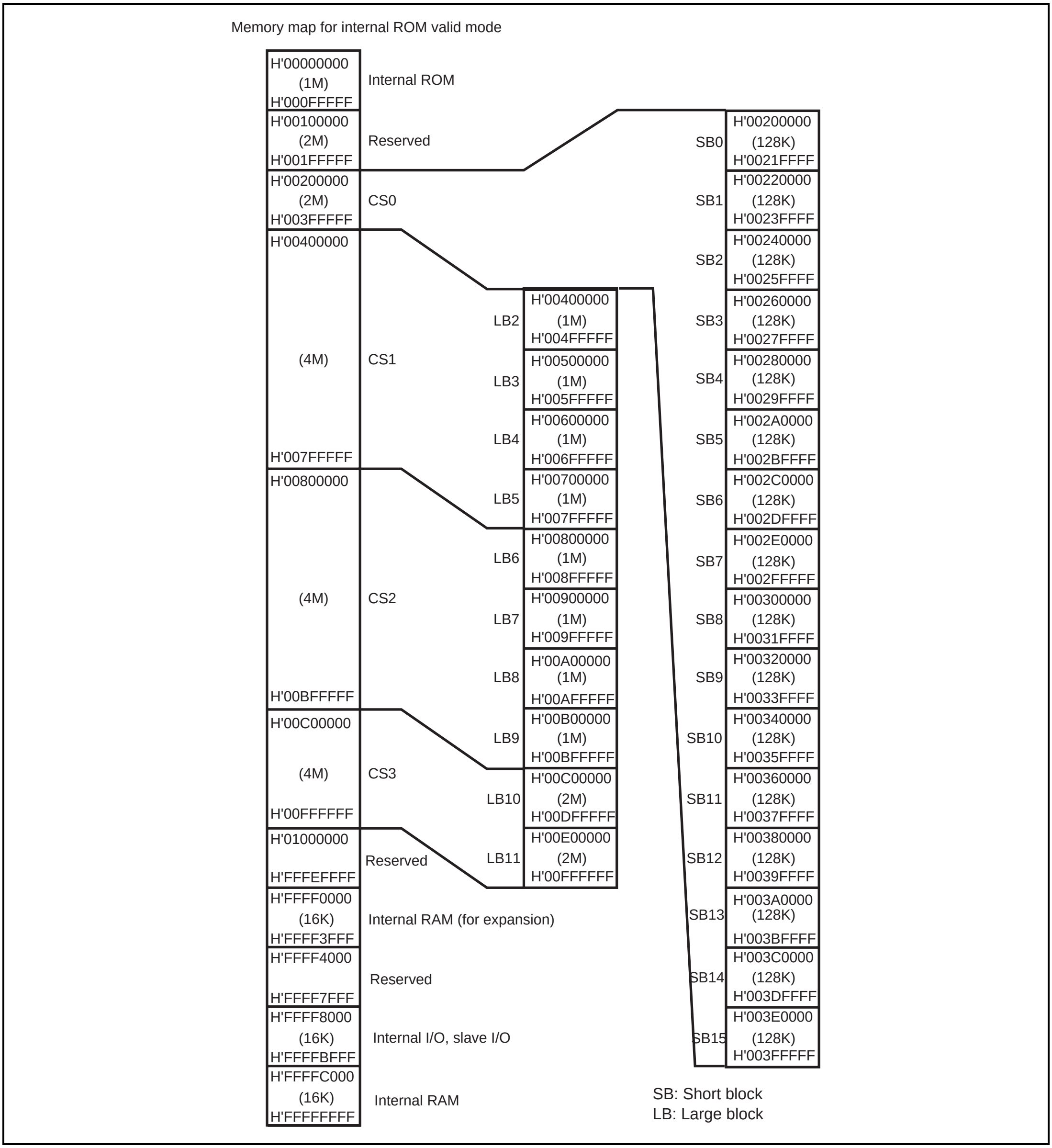


Figure E.2 Memory Map in the Valid Internal ROM Mode

E.3 User System Interface

1. The pins listed in table E.1 are occupied solely by the emulator, and are not connected to the user system.

Table E.1 Pins Occupied Solely by the Emulator

FP-80Q Pin No.	FP-100M Pin No.	Signal Name
74	97	XTAL
—	85	$\overline{\text{HSTBY}}$
—	16	$\overline{\text{DBGMD}}$
—	11	$\overline{\text{ASEBRKAK}}$

2. The H-UDI function is occupied solely by the emulator, and does not operate as the user function.
3. 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.4 Hardware Break Function

E.4.1 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.

1. 32-bit bus area

- 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.

2. 16-bit bus area

- Longword access

  Longword data is read and written in two-word cycles. A data condition is only valid for word access when specified as longword. An address condition is only valid for a longword-access cycle when specified as a multiple of two.

- 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. An address condition is only valid for a longword-access cycle when specified as a multiple of two.

- Byte access

  Byte data is read and written in single bus cycles. A data condition is only valid for word access when specified as word. Any address condition, whether an even or odd number, is valid.

3. 8-bit bus area

   This area is accessed by byte (four byte accesses for a longword and two byte accesses for a word). Either an odd or even number is a valid address specification, but the data specification must be for a byte.

E.5 Hardware Sequential Break/Trace Function

1. Six or more external bus cycles are required between the satisfaction of each condition that is specified.
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, then execute the user program.

E.6 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

	Emulator		
Register Name	**Power On**	**Reset (Reset CPU)**	**MCU (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.7 Step Function

Some types of break will be disabled according to the type of step function to be executed. Table E.4 shows the relations between the type of step function and the enabling or disabling of break conditions.

Table E.4 Relations between the Type of Step Function and Available Break Conditions

	Step In	**Step Over**	**Step Out**
Hardware break (BREAK CONDITION A/B/C)	O	O	O
Internal break (BREAK CONDITION U), internal sequential break	X	X	X
Hardware sequential break (BREAK CONDITION SEQUENTIAL A/B)	O	X	X
Software break	X	X	X
Software sequential break	X	X	X
Timeout break	X	X	X
Break due to trace buffer overflow	X	X	X

Note: O: Break conditions are enabled.
X: Break conditions are disabled.

E.8 Displaying and Modifying the Contents of Memory

E.8.1 Suspension of Program Execution: Reference Periods

Table E.5 shows the reference value for displaying and modifying the memory contents during user program execution.

Table E.5 Suspension of Program Execution: Reference Values

Method of Access	Condition		Period Suspended
Short breaks	Read	Reading a 256-byte longword from the internal RAM	4.2 ms
	Write	Writing a longword to the internal RAM	1.2 ms
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.6 shows the environment in which these measurements were obtained.

Table E.6 Measurement Environment

Item	Setting
System clock (ϕ)	12.5 MHz
Clock mode	Clock mode 0
H-UDI clock	2.5 MHz

E.8.2 Short Break

When memory is displayed or modified by a short break during sleep mode or standby mode, the system leaves that mode and execution resumes from the instruction following the SLEEP instruction.

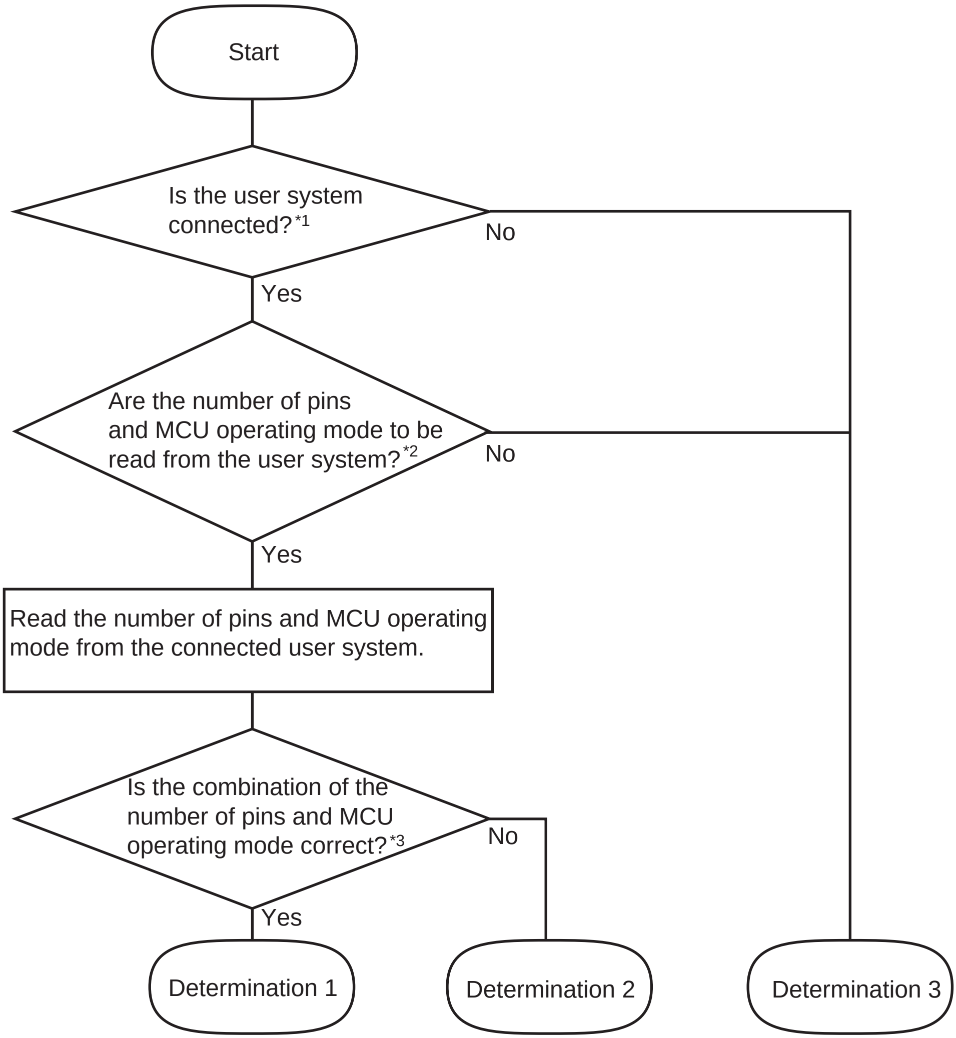
E.8.3 Parallel Access

The display of memory cannot be updated or modified by parallel access during standby mode.

E.9 CPU Operating Mode

1. Number of pins on the supported MCU and the MCU's operating mode

   When the emulator starts up, it determines the number of pins on the supported MCU and the MCU's operating mode according to the following procedure.



Notes:
1. If the user system is not connected to the user system interface cable, the emulator will detect a low voltage on Vcc (Vcc Down) from the user system and be linked down.
2. Set on the [MD Pin (Device)] page of the [CPU Operating Mode] dialog box.
3. Only the single-chip mode is available with the SH7046 (80 pins). Expanded MCU mode is not available with the SH7047 (100 pins).

Figure E.3 Procedure of Determination

Determination 1:	Uses the number of pins and MCU operating mode read from the connected user system.
Determination 2:	Uses the number of pins read from the connected user system. MCU operating mode: Clock mode 0 and single-chip mode
Determination 3:	Uses the number of pins and MCU operating mode set in the [CPU Operating Mode] dialog box.

2. I/O register file

The I/O register file is switched by selection of an MCU in [Device] on the [MD Pin (Device)] page. The I/O register file that is used with the SH7046 and SH7047 is Custom.io, when [Custom] is selected as [Device].