

To our customers,

Old Company Name in Catalogs and Other Documents

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

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

April 1st, 2010
Renesas Electronics Corporation

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

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

Notice

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

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

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

"Specific": Aircraft; aerospace equipment; submersible repeaters; nuclear reactor control systems; medical equipment or systems for life support (e.g. artificial life support devices or systems), surgical implantations, or healthcare intervention (e.g. excision, etc.), and any other applications or purposes that pose a direct threat to human life.

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

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

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

To all our customers

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

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

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

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

Cautions

Keep safety first in your circuit designs!

1. Renesas Technology Corporation puts the maximum effort into making semiconductor products better and more reliable, but there is always the possibility that trouble may occur with them. Trouble with semiconductors may lead to personal injury, fire or property damage.

Remember to give due consideration to safety when making your circuit designs, with appropriate measures such as (i) placement of substitutive, auxiliary circuits, (ii) use of nonflammable material or (iii) prevention against any malfunction or mishap.

Notes regarding these materials

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

SH7060 E8000
Renesas Debugging Interface
User's Manual

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

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

DEFINITION OF SIGNAL WORDS

CAUTION indicates a hazardous situation which, **if not avoided**, may result in **minor or moderate injury** to you or other people, or may result in **damage to the machine** or **loss of the user program**. It may also be used to alert against unsafe usage.

NOTE emphasizes essential information.

Preface

The SH7060 E8000 Hitachi Debugging Interface (referred to as the HDI) is a software tool that supports program development in an environment using an E8000 emulator for the Hitachi microprocessor SH7060 (referred to as the emulator).

This manual is the Debugging Platform User's Manual, which is a separate volume to the Hitachi Debugging Interface User's Manual, and describes the HDI functions and its usage. Read this manual and the following manuals before using the HDI.

For details on the emulator,

- SH7060 E8000 Emulator User's Manual
- 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
- SH7060 E8000 Emulator Diagnostic Program Operation Manual

For details on the related software manuals,

- Hitachi Debugging Interface User's Manual
- Hitachi Embedded Workshop User's Manual
- SuperH RISC engine C/C++ Compiler User's Manual
- SuperH RISC engine Cross Assembler User's Manual
- H Series Linkage Editor, Librarian, Object Converter User's Manual
- SuperH RISC engine C/C++ Compiler, Assembler, Optimizing Linkage Editor

For details on the SH7060-series microprocessor,

- SH7065 Hardware Manual
- SH1/SH2/SH-DSP Programming Manual

Refer to section 2, Installation, for details on HDI installation.

In this user's manual, the operating environment is assumed to be the English version on the IBM PC.

- Notes:
1. Microsoft[®], Windows[®], Windows[®] 95, Windows[®] 98, Windows[®] Me, Windows NT[®], and Windows[®] 2000 are registered trademarks of Microsoft Corporation in the United States and/or in other countries.
 2. IBM PC is a registered trademark of International Business Machines Corporation in the United States.
 3. Pentium[®] is a registered trademark of Intel Corporation in the United States.
 4. Adobe[®], Acrobat, and Acrobat[®] Reader[®] are registered trademarks of Adobe Systems Incorporated.

All product names and brand names used in this user's manual are trademarks or registered trademarks of each company.

Contents

Section 1 Overview	1
1.1 Overview	1
1.1.1 Features	1
1.2 Operating Environment.....	2
1.3 Content.....	3
Section 2 Installation.....	5
2.1 Installation Procedures.....	5
2.1.1 Setting up the Emulator	5
2.2 Setting Up with Windows® Operating Systems	5
2.2.1 Setting Up the PCI Interface Board on Windows® 95, Windows® 98, or Windows® Me	6
2.2.2 Setting Up the PC Interface Board on Windows NT® 4.0.....	7
2.2.3 Setting Up the PC Interface Board on Windows® 2000	8
2.2.4 PC Interface Board Specifications (ISA Bus Specifications)	9
2.3 Installing the System Program	12
2.4 Checking the Emulator	18
2.5 Troubleshooting	21
2.5.1 Connection Error of the PC Interface Board.....	21
2.5.2 Connection Error of the Emulator.....	22
2.6 Uninstallation.....	23
2.6.1 Uninstalling Software	23
2.6.2 Uninstalling the Acrobat® Reader®	23
Section 3 Tutorial	25
3.1 Introduction	25
3.2 Running the HDI.....	26
3.2.1 Selecting a Target Platform	26
3.3 Setting up the Emulator	28
3.3.1 Setting the [Configuration] Dialog Box.....	28
3.3.2 Setting the Memory Map	30
3.4 Downloading.....	32
3.4.1 Downloading the Tutorial Program	32
3.4.2 Displaying the Source Program	34
3.5 Setting the Software Breakpoints.....	36
3.6 Executing the Program.....	37
3.7 Reviewing Breakpoints.....	39
3.8 Viewing Memory.....	40
3.9 Watching Variables.....	42

3.10	Stepping Through a Program	45
3.10.1	Executing [Step In] Command.....	47
3.10.2	Executing [Step Out] Command	48
3.10.3	Executing [Step Over] Command	50
3.11	Displaying Local Variables.....	52
3.12	Setting the Hardware Break Conditions.....	53
3.13	Setting the Sequential Break Conditions	61
3.14	Displaying and Searching for Trace Information.....	72
3.14.1	Displaying Trace Information.....	72
3.14.2	Searching for Trace Information.....	73
3.15	Setting the Trace Acquisition Conditions	78
3.16	Measuring the Subroutine Execution Time and the Execution Count	86
3.16.1	Displaying the Measurement Results of Subroutine Execution Time and Execution Count	90
3.16.2	Setting the Conditions of Subroutine Execution Time and Execution Count	91
3.16.3	Measuring the Subroutine Execution Time and Execution Count	96
3.17	Saving and Loading the Session	99
3.18	What Next?	100
 Section 4 Function		101
4.1	Windows and Dialog Boxes	101
4.2	Command Descriptions.....	104
 Section 5 Use of Diagnostic Program.....		109
5.1	Diagnostic Program Operation	109
5.1.1	Target File Setting.....	109
5.1.2	Diagnostic Program Start-Up.....	110
5.1.3	Diagnostic Program Termination.....	112
 Section 6 Error Messages.....		113
6.1	Error Messages	113

Figures

Figure 2.1	[Start] Menu (System Install Tool).....	14
Figure 2.2	[Browse for Folder] Dialog Box.....	15
Figure 2.3	[E8000 Load Files] Dialog Box (SH7060).....	16
Figure 2.4	[System Install Completed!] Dialog Box (SH7060)	17
Figure 2.5	[E8000 Load Files] Dialog Box (SH7060).....	18
Figure 2.6	[Start] Menu (Initiating the HDI)	19
Figure 2.7	[Select Session] Dialog Box.....	19
Figure 2.8	[Driver Details] Dialog Box (When ISA Bus Is Selected)	20
Figure 2.9	Status Bar during the HDI Initiation.....	20
Figure 2.10	Status Bar at the HDI Initiation Completion	21
Figure 2.11	PC Interface Board Connection Error Message Box	21
Figure 2.12	Emulator Connection Error Message Box.....	22
Figure 3.1	[Start] Menu	26
Figure 3.2	[Select Session] Dialog Box.....	26
Figure 3.3	HDI Window	27
Figure 3.4	[Configuration] Dialog Box	28
Figure 3.5	[Memory Mapping] Dialog Box (Before Setting)	30
Figure 3.6	[Add Memory Mapping] Dialog Box.....	31
Figure 3.7	[Memory Mapping] Dialog Box (at Setting).....	32
Figure 3.8	[Load Program] Dialog Box.....	33
Figure 3.9	HDI Dialog Box	34
Figure 3.10	[Open] Dialog Box	35
Figure 3.11	[Source] Window (Displaying the Source Program).....	36
Figure 3.12	[Source] Window (Setting a Software Breakpoint).....	37
Figure 3.13	[Reset Go] Button.....	37
Figure 3.14	[Source] Window (Break State)	38
Figure 3.15	[System Status] Window	39
Figure 3.16	[Breakpoints] Window	39
Figure 3.17	[Open Memory Window] Dialog Box.....	40
Figure 3.18	[Word Memory] Window.....	41
Figure 3.19	[Instant Watch] Dialog Box.....	42
Figure 3.20	[Watch] Window (Displaying the Array).....	43
Figure 3.21	[Add Watch] Dialog Box.....	43
Figure 3.22	[Watch] Window (Displaying the Variable).....	44
Figure 3.23	[Watch] Window (Displaying Array Elements)	44
Figure 3.24	[Source] Window (Step Execution).....	46
Figure 3.25	[Step In] Button	47
Figure 3.26	[Source] Window (Step In)	47
Figure 3.27	[Step Out] Button	48
Figure 3.28	[Source] Window (Step Out).....	48
Figure 3.29	[Source] Window (Step Out -> Step In).....	49

Figure 3.30	[Source] Window (Before Step Over Execution).....	50
Figure 3.31	[Step Over] Button	51
Figure 3.32	[Source] Window (Step Over).....	51
Figure 3.33	[Locals] Window	52
Figure 3.34	[Breakpoints] Window (Before Hardware Break Condition Setting).....	55
Figure 3.35	[Condition A] Page ([Break] Dialog Box).....	56
Figure 3.36	[Address] Page ([Break Condition A1] Dialog Box).....	57
Figure 3.37	[Bus State] Page ([Break Condition A1] Dialog Box)	58
Figure 3.38	[Break] Dialog Box (After Hardware Break Condition Setting).....	59
Figure 3.39	[Breakpoints] Window (at Break Condition A Setting)	60
Figure 3.40	[Configuration] Dialog Box (at Sequential Break Condition Setting).....	63
Figure 3.41	[Breakpoints] Window (Before Sequential Break Condition Setting).....	64
Figure 3.42	[Break] Dialog Box ([Break Condition UBC] Page).....	65
Figure 3.43	[Break Condition UBC2] Dialog Box (Condition 2 [Address] Page)	66
Figure 3.44	[Break Condition UBC2] Dialog Box (Condition 2 [Bus State] Page)	67
Figure 3.45	[Break] Dialog Box (After [Break Condition UBC2] Condition Setting).....	68
Figure 3.46	[Breakpoints] Window (After Sequential Break Condition Setting)	69
Figure 3.47	[Source] Window at Execution Halt (Sequential Break).....	70
Figure 3.48	Contents of [System Status] Window (Sequential Break).....	71
Figure 3.49	[Trace] Window (Free Trace Results)	72
Figure 3.50	[General] Page ([Trace Filter] Dialog Box).....	74
Figure 3.51	[Address] Page ([Trace Filter] Dialog Box)	75
Figure 3.52	[Bus & Area] Page ([Trace Filter] Dialog Box)	76
Figure 3.53	[Trace] Window (Trace Filter Results).....	77
Figure 3.54	[Trace Acquisition] Dialog Box	80
Figure 3.55	[Condition A] Page ([Trace Acquisition] Dialog Box)	81
Figure 3.56	[General] Page ([Trace Condition A1] Dialog Box)	82
Figure 3.57	[Address] Page ([Trace Condition A1] Dialog Box).....	83
Figure 3.58	[Bus State] Page ([Trace Condition A1] Dialog Box).....	84
Figure 3.59	[Condition A] Page ([Trace Acquisition] Dialog Box)	85
Figure 3.60	[Breakpoints] Window (with Break Conditions Canceled).....	89
Figure 3.61	[Performance Analysis] Window (Initial State)	90
Figure 3.62	[PA] Page ([Performance Analysis] Dialog Box).....	91
Figure 3.63	[General] Page ([Performance 1] Dialog Box).....	92
Figure 3.64	[Address] Page ([Performance 1] Dialog Box)	93
Figure 3.65	[Input Function Range] Dialog Box	93
Figure 3.66	[Address] Page (after the Address of the sort Function is Specified)	94
Figure 3.67	[PA] Page ([Performance Analysis] Dialog Box).....	95
Figure 3.68	[Performance Analysis] Window (before Program Execution).....	97
Figure 3.69	[Performance Analysis] Window (after Program Execution).....	97
Figure 3.70	[Performance Analysis] Window (Execution Count).....	98
Figure 5.1	Example of Target File Display.....	109
Figure 5.2	Diagnostic Program Start-Up Confirmation Dialog Box	110

Figure 5.3 Diagnostic Program Start-Up Window.....	111
Figure 5.4 Diagnostic Program Termination	112

Tables

Table 1.1	Operating Environment.....	2
Table 1.2	CD-R Files	3
Table 2.1	Contents of Emulator System Programs	13
Table 2.2	Types of Connection and Drivers	14
Table 3.1	Configuration of the Sample Program (at Default Installation).....	25
Table 3.2	Values to be Set in the [Configuration] Dialog Box	29
Table 3.3	Memory Type	30
Table 3.4	Step Command	45
Table 3.5	Dialog Boxes for Setting Hardware Break Conditions	53
Table 3.6	Main Break Conditions	54
Table 3.7	Dialog Boxes for Setting Sequential Break Conditions.....	61
Table 3.8	Main Sequential Break Functions.....	62
Table 3.9	Main Trace Search Conditions	73
Table 3.10	Dialog Boxes for Setting Trace Acquisition Conditions.....	78
Table 3.11	Main Trace Functions	79
Table 3.12	Dialog Boxes for Setting Subroutine Execution Time and Execution Count	87
Table 3.13	Subroutine Measurement Mode.....	88
Table 3.14	Display Formats for Measurement Results	96
Table 4.1	List of Windows and Dialog Boxes	101
Table 4.2	HDI Commands	105
Table 6.1	Error Messages	113

Section 1 Overview

1.1 Overview

The HDI promotes efficient debugging of programs written in C/C++ or assembly language in the environment using the emulator.

1.1.1 Features

The HDI has the following features:

- High-speed downloading of load module files.
- A comprehensive set of break functions, trace conditions, and functions to set or edit memory maps are enabled by the HDI windows.
- Large-capacity trace information can be displayed (a maximum of 64-k bus cycles are displayed on the window).
- Command line functions (command system for the HDI).

1.2 Operating Environment

The HDI is provided on CD-R. The following shows the acceptable operating environments:

Table 1.1 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, Windows NT® 4.0, or Windows® 2000
Minimum memory capacity for operation	32 Mbytes (more than twice the size of the load module is recommended)
Display	Resolution better than 800 × 600 (SVGA) is recommended
Empty space in a hard disk	Disk capacity required for installation: 40 Mbytes or more 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, Windows NT® 4.0, and Windows® 2000
CD-ROM drive	Required for installation of E8000 emulator software or referring to the user's manual

1.3 Content

The CD-R includes the E8000 Emulator Software and the user's manual for both SH7060 E8000 emulator. See table 1.2. The directories described in the table are under the 7060 directory, which is under the root directory.

Table 1.2 CD-R Files

Directory Name	File Name	Contents	Note
	setup.exe	Installer	
\System	E8000.sys ^[*1]	System program for the SH7060 E8000 emulator	
\System	Shcnf706.sys ^[*1]	Control program for the SH7060 E8000 emulator	
\System	Shdct706.sys ^[*1]	Configuration file for the SH7060 E8000 emulator	
\System	diag.sys ^[*1]	Diagnostic and maintenance program	
\System	Setup.cc ^[*1]	Script file for automatic installation (for IPW)	
\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\Isla\Nt	emulator.sys	System file (ISA)	For Windows NT [®]
\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

Table 1.2 CD-R Files (cont)

Directory Name	File Name	Contents	Notes
\Manuals\Japanese	HS6400DIIW5SJ.pdf [*2]	Hitachi Debugging Interface user's manual	PDF documents in Japanese [*5]
\Manuals\Japanese	HS7060EDD81HJ.pdf [*2]	SH7060 E8000 emulator user's manual	PDF documents in Japanese [*5]
\Manuals\Japanese	HS7060D8IW2SJ.pdf [*2]	SH7060 E8000 Hitachi Debugging Interface User's Manual	PDF documents in Japanese [*5]
\Manuals\Japanese	HS7060TM81HJ(*) ^[*3] .pdf [*2]	Descriptive notes on the diagnostic program for the SH7060 E8000 emulator	PDF documents in Japanese [*5]
\Manuals\Japanese	HS7060EBK81HJ(*) ^[*3] .pdf [*4]	Descriptive notes on the evaluation chip board HS7060EBK81H for the SH7060 E8000 emulator	PDF documents in Japanese [*5]
\Manuals\English	HS7060EBK81HE.pdf [*4]	SH7060 E8000 emulator user's manual	PDF documents in English [*5]
\Manuals\English	HS7060D8IW2SJ.pdf [*2]	SH7060 E8000 Hitachi Debugging Interface user's manual	PDF documents in English [*5]
\Manuals\English	HS7060D8IW2SE.pdf [*4]	SH7060 E8000 Hitachi Debugging Interface User's Manual	PDF documents in English [*5]
\Manuals\English	HS7060EBK81HE(*) ^[*3] .pdf [*4]	Descriptive notes on the evaluation chip board HS7060EBK81H for the SH7060 E8000 emulator	PDF documents in English [*5]
\Manuals\English	HS7060TM81HE(*) ^[*3] .pdf [*4]	Descriptive notes on the diagnostic program for the SH7060 E8000 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 installer.
3. (*) indicates a manual revision.
4. This is the English version of the manual. It cannot be installed by the installer.
5. Use the Acrobat® Reader to see PDF documents.

Section 2 Installation

2.1 Installation Procedures

The installation procedures for the HDI are described below.

2.1.1 Setting up the Emulator

Set up the emulator following instructions described in section 3, Preparation before Use, in the SH7060 E8000 Emulator User's Manual on the CD-R. The file in the emulator system disk (FD) described in the SH7060 E8000 Hitachi Debugging Interface User's Manual is stored in the 7060\system directory on the CD-R.

2.2 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 2.2.4, 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>:\7060\setup.exe and click the [OK] button. <Drive> is the CD-ROM drive.

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

2.2.1 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>:\7060\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>:\7060\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 2.2.4, 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 [SH7060 E8000 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.

2.2.2 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 2.2.4, 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 [SH7060 E8000 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.

2.2.3 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>:\7060\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>:\7060\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 [SH7060 E8000 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.

2.2.4 PC Interface Board Specifications (ISA Bus Specifications)

A PC interface board (HS6000EII01H, ISA bus specifications) is available for use as the interface for communications between the emulator and the HDI.

The PC interface board is a memory-mapped board; the memory area to be used by the PC interface board must be defined in the emulation memory before the PC interface board is placed in an unused expansion slot. Otherwise, programs may accidentally use the area of memory required by the PC interface hardware.

CAUTION

Be careful not to allocate the PC interface board to a memory area to which other boards have been allocated. If you fail to take care of this point, correct operation cannot be guaranteed.
When the PC interface board is shipped, the switch settings are for allocation to the address range from H'D0000 to H'D3FFF.

For details on installing the PC interface board, refer to the SH7060 E8000 Emulator User's Manual and the Description Notes on Using the PC Interface Board (HS6000EII01H).

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.

2.3 Installing the System Program

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

The E8000 system program must be transferred to flash memory in the E8000 station. The emulator cannot be used without the E8000 system program.

The system programs are stored in \SYSTEM under the HDI installation directory. Table 2.1 is a list of system programs with descriptions.

Table 2.1 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	SHDCT706.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	SHCNF706.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 E8000 station for maintenance.

The system programs are registered by using a dedicated E8000/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 E8000 station. Select [SH7060 E8000 Emulator Software]-[Tools]-[System Install Tool] from [Start]-[Programs].

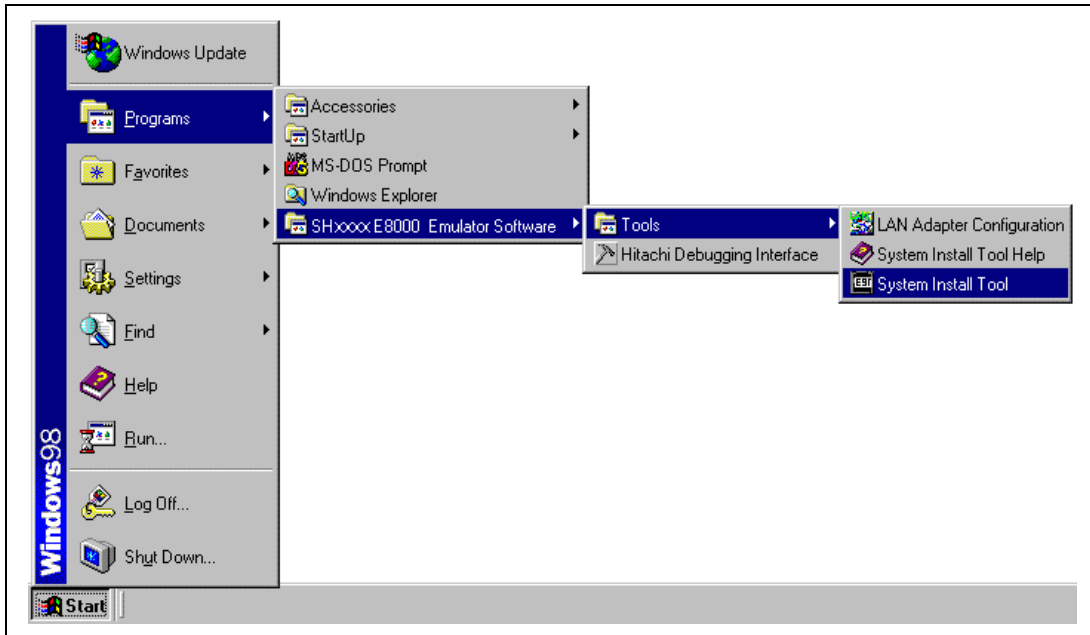


Figure 2.1 [Start] Menu (System Install Tool)

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 E8000 Emulator from [Driver]. Table 3.4 shows the types of connections and drivers.

Table 2.2 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.

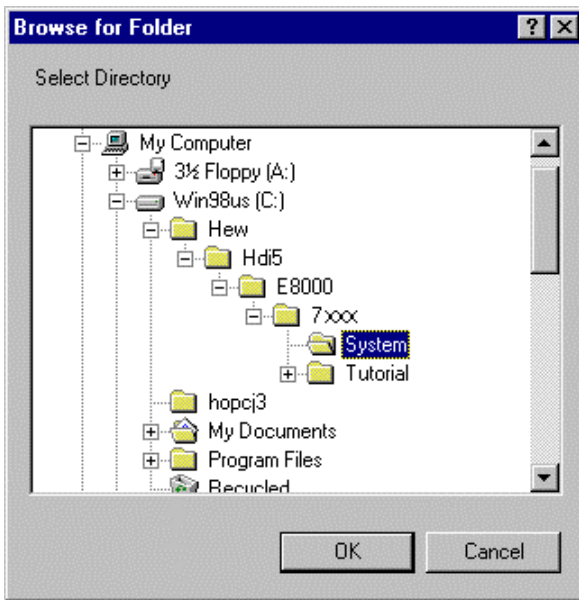


Figure 2.2 [Browse for Folder] Dialog Box

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

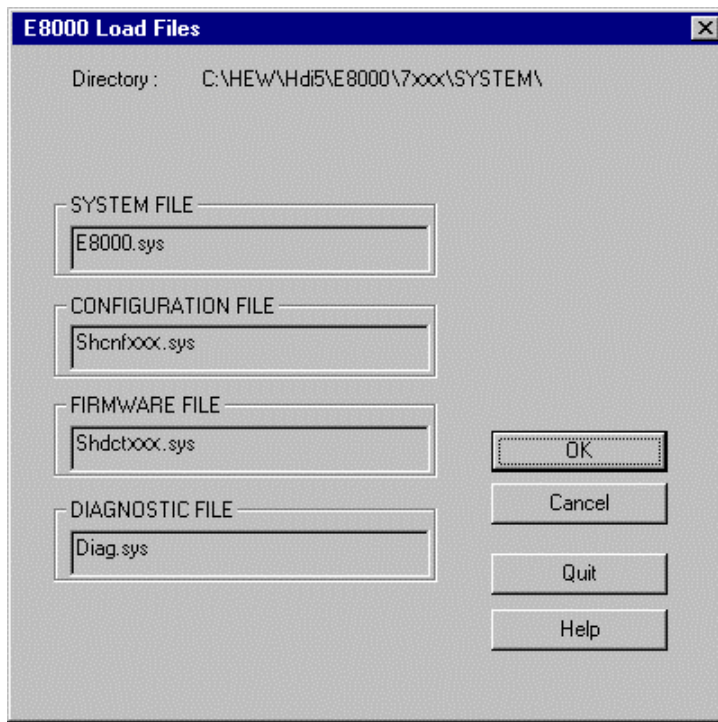


Figure 2.3 [E8000 Load Files] Dialog Box (SH7060)

Click the [OK] button. Installation of the system programs to the flash memory of the E8000 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.

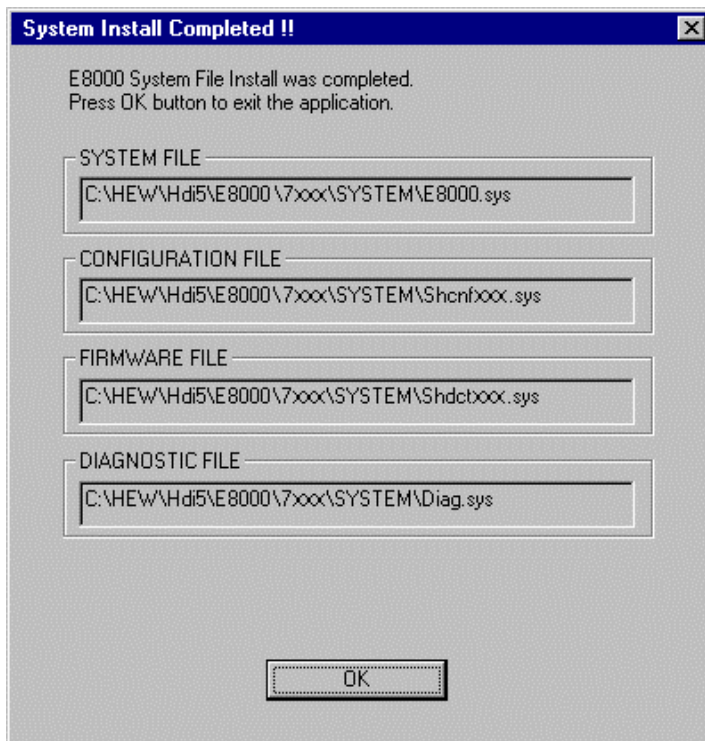


Figure 2.4 [System Install Completed!!] Dialog Box (SH7060)

Click the [OK] button. The installation of the system program to the flash memory of the E8000 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.

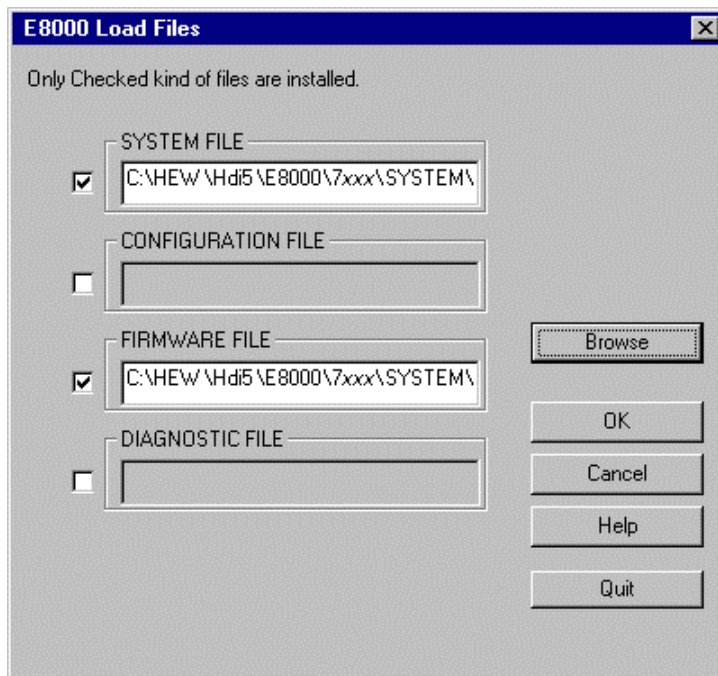


Figure 2.5 [E8000 Load Files] Dialog Box (SH7060)

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 E8000 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 [SH7060 E8000 Emulator Software]-[Tools]-[System Install Tool] from the [Start] menu.

2.4 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 [SH7060 E8000 Emulator Software]-[Hitachi Debugging Interface] from the [Start] menu.

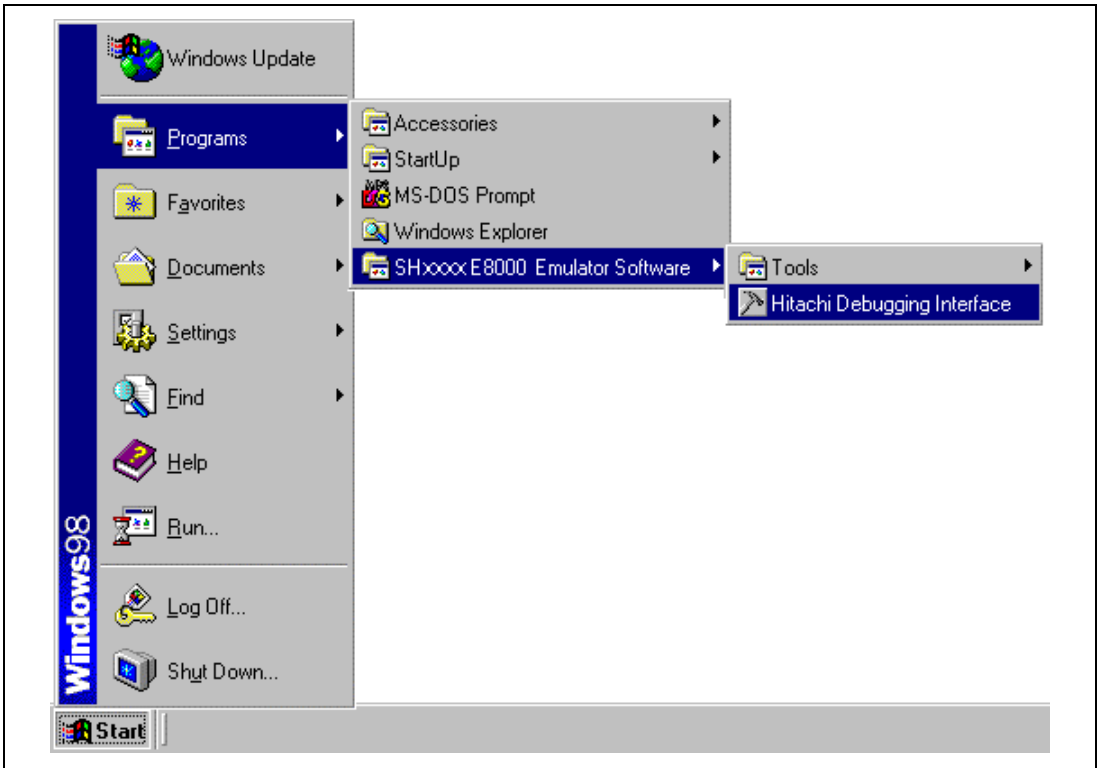


Figure 2.6 [Start] Menu (Initiating the HDI)

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

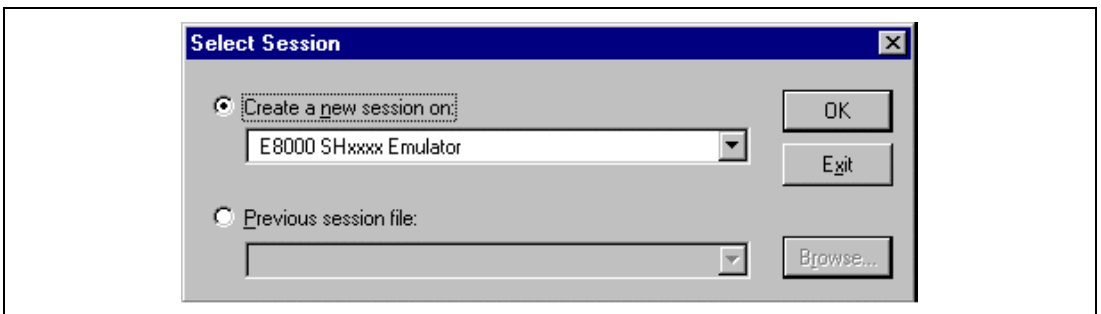


Figure 2.7 [Select Session] Dialog Box

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 2.2, Types of Connections and Drivers. The example below is when the emulator is to be connected to the PC interface via the ISA bus.

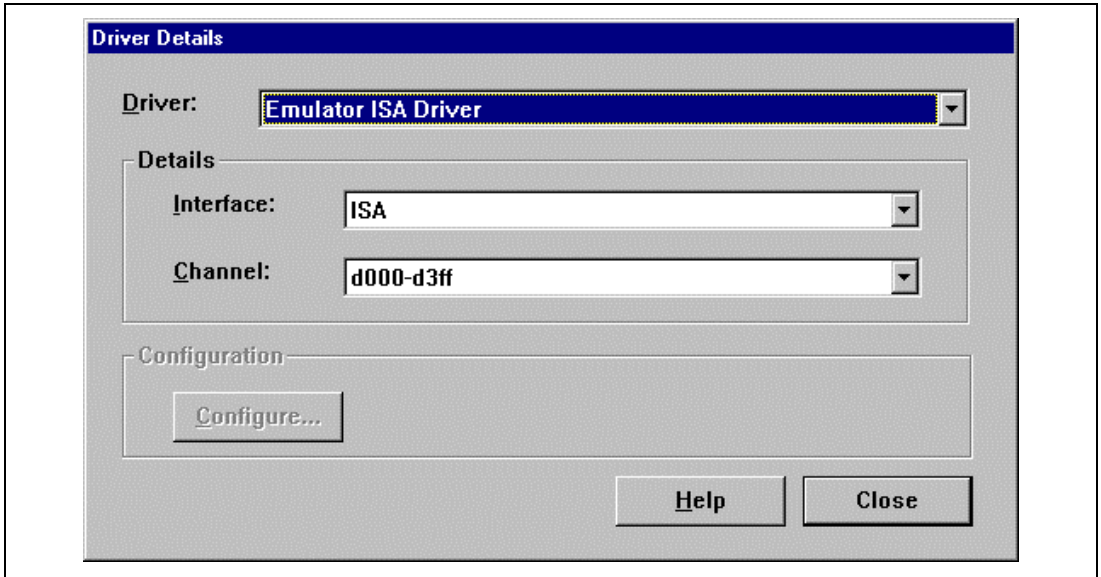


Figure 2.8 [Driver Details] Dialog Box (When ISA Bus Is Selected)

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

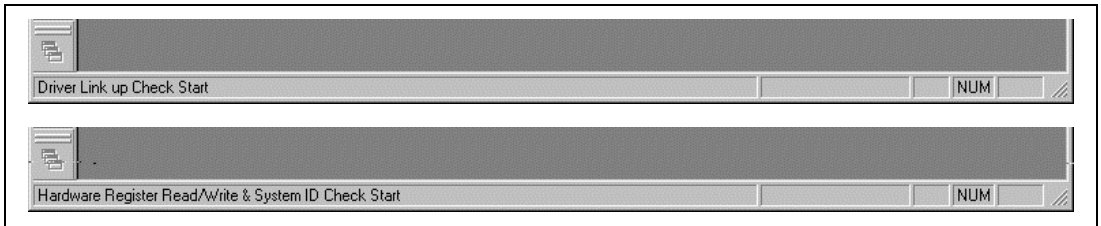


Figure 2.9 Status Bar during the HDI Initiation

A message box will be displayed to ask whether to initiate the diagnostic program (figure 5.2) 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 5, Operating Procedure for the Diagnostic Program. “Link up” appears on the status bar when the HDI and the emulator have successfully started up.

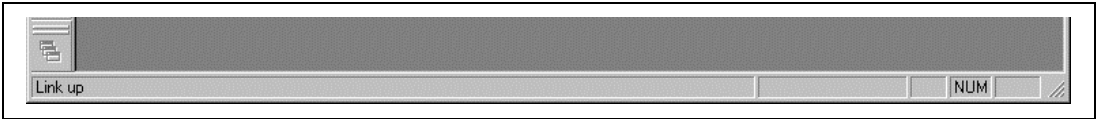


Figure 2.10 Status Bar at the HDI Initiation Completion

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.

2.5 Troubleshooting

2.5.1 Connection Error of the PC Interface Board

The HDI displays the following message box when the PC interface board for connecting to the emulator cannot be detected.



Figure 2.11 PC Interface Board Connection Error Message Box

The following causes are possible:

- The memory area reserved in one or more of the following settings differs from the setting of the rear-panel switch of the PC interface board.
 - [Computer Properties] dialog box
 - CONFIG.SYS file
 - SYSTEM.INI file
- The selected memory area has already been used for another application.

2.5.2 Connection Error of the Emulator

The HDI displays the following message box when the emulator cannot be detected.

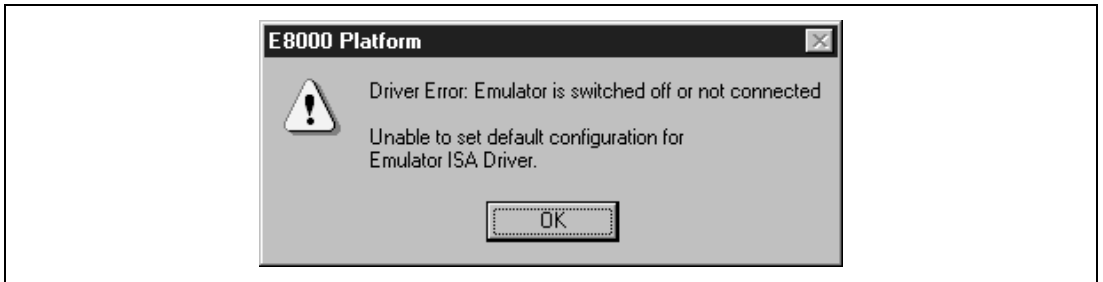


Figure 2.12 Emulator Connection Error Message Box

The following causes are possible:

- The AC power-supply cable is not connected to the emulator, or the emulator is not switched on.
- The PC interface cable is not correctly connected between the PC interface board and the emulator.

2.6 Uninstallation

2.6.1 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 [SH7060 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.

2.6.2 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 3 Tutorial

3.1 Introduction

The following describes the main functions of the HDI by using a sample program for sorting random data.

The sample program performs the following actions:

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

Table 3.1 shows the configuration of the sample program.

Table 3.1 Configuration of the Sample Program (at Default Installation)

Item No.	Description of File	File Name and Directory
1	HEW workspace file*	\HEW\HDI5\E8000\7060\TUTORIAL\TUTORIAL.HWS
2	ELF/DWARF2-type load module file	\HEW\HDI5\E8000\7060 \TUTORIAL\TUTORIAL\DEBUG\TUTORIAL.ABS
3	Source file (main program)	\HEW\HDI5\E8000\7060 \TUTORIAL\TUTORIAL\TUTORIAL.C

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

3.2 Running the HDI

To run the HDI, select the [Hitachi Debugging Interface] from the [Start] menu.

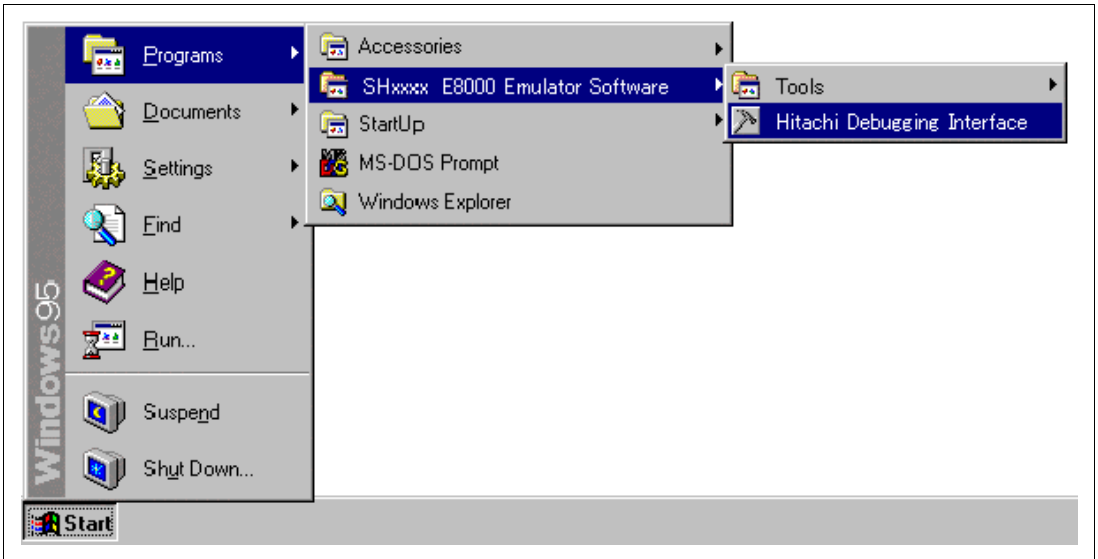


Figure 3.1 [Start] Menu

3.2.1 Selecting a Target Platform

When the HDI is activated, the HDI asks to select the current session platform.

Select the E8000 SH7060 Emulator.

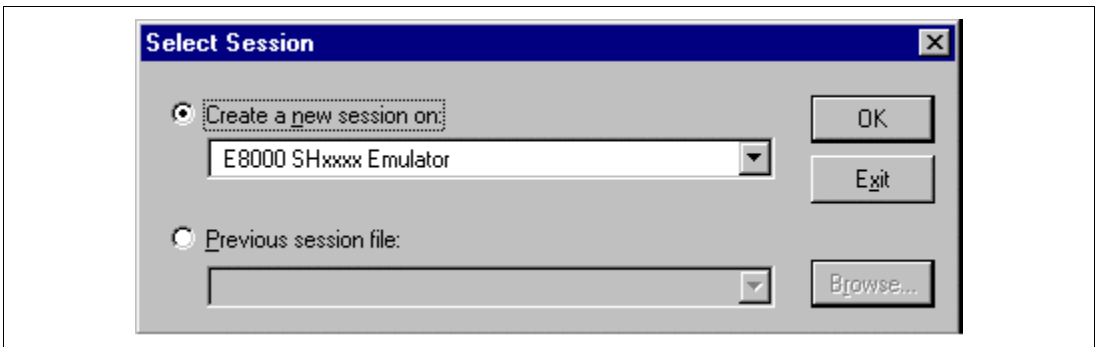


Figure 3.2 [Select Session] Dialog Box

When [New Session...] is selected from the [File] menu, a target platform can be changed at any time.

When the E8000 emulator has correctly been setup, the HDI window is displayed together with the Link Up message at the status bar. The main functions of the window are described on the next page.

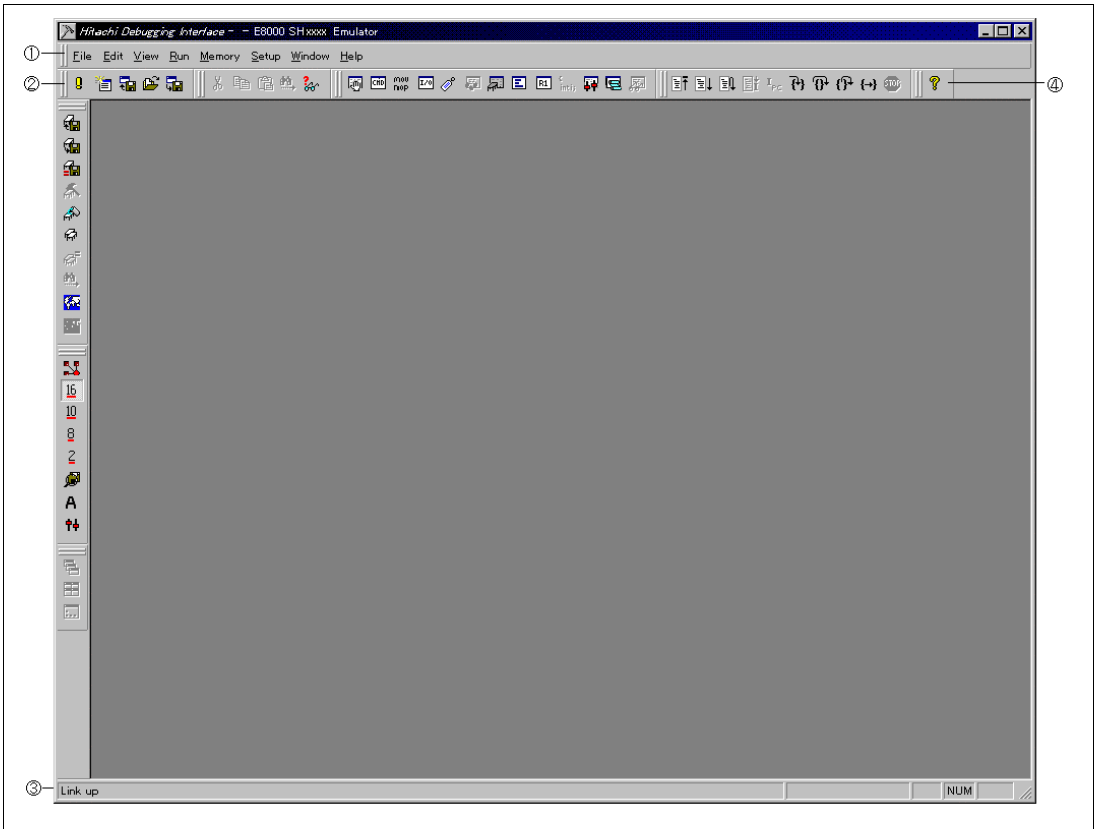


Figure 3.3 HDI Window

Numbers in figure 3 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.

3.3 Setting up the Emulator

The following CPU conditions must be set up before downloading the program:

- Device type
- CPU operating mode
- Clock mode
- Operating clock
- Execution operating mode
- Memory map

The following describes how to set up the tutorial program in the emulator.

3.3.1 Setting the [Configuration] Dialog Box

- Select [Configure Platform...] from the [Setup] menu to set configuration.

The following dialog box is displayed.

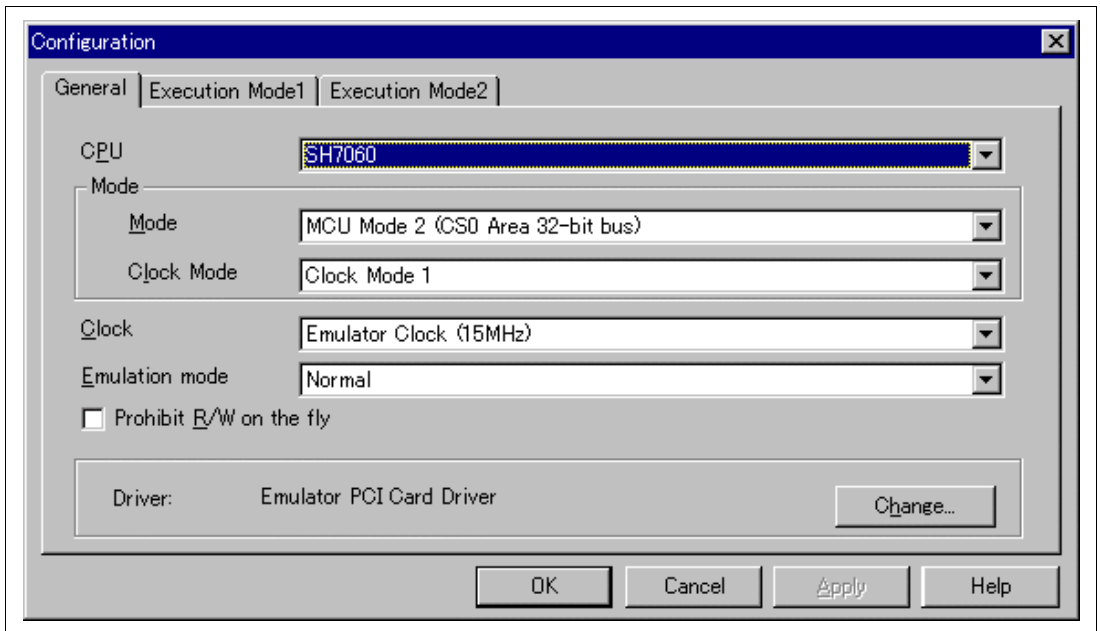


Figure 3.4 [Configuration] Dialog Box

Table 3.2 shows the options in the [Configuration] dialog box and the values to be set in the tutorial program.

Table 3.2 Values to be Set in the [Configuration] Dialog Box

Page	Option	Value
[General] page	CPU operating mode [Mode]	MCU Mode 2 (CS0 Area 32-bit bus)
	Clock mode [Clock Mode]	Clock Mode 1
	Emulation clock [Clock]	Emulator Clock (15 MHz) (depends on the connection method being used)
	Emulation mode [Emulation mode]	Normal (normal execution, default setting)
	Memory access enable/disable during execution [Prohibit R/W on the fly]	Enable (no check)
[Execution Mode1] page	Program counter display interval [Execution status display interval]	About 200 ms (default setting)
	Timer resolution [The minimum time to be measured by Go command execution]	1.6 us (default setting)
	Multi-break function [Enable the multi break of External probe No. 1]	Invalid (default setting)
	BREQ signal control [Enable the BREQ signal input]	Valid (default setting)
[Execution Mode2] page	Trigger output control 1 at break [TRGU Option]	Upper radio button (default setting)
	Trigger output control 2 at break [TRGB Option]	Upper radio button (default setting)

- Set the above values for the CPU operating mode and clock mode, and click the [OK] button.
- Start the HDI.
- After the HDI is started, display the [Configuration] dialog box. Set the emulation clock and click the [OK] button.

3.3.2 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.

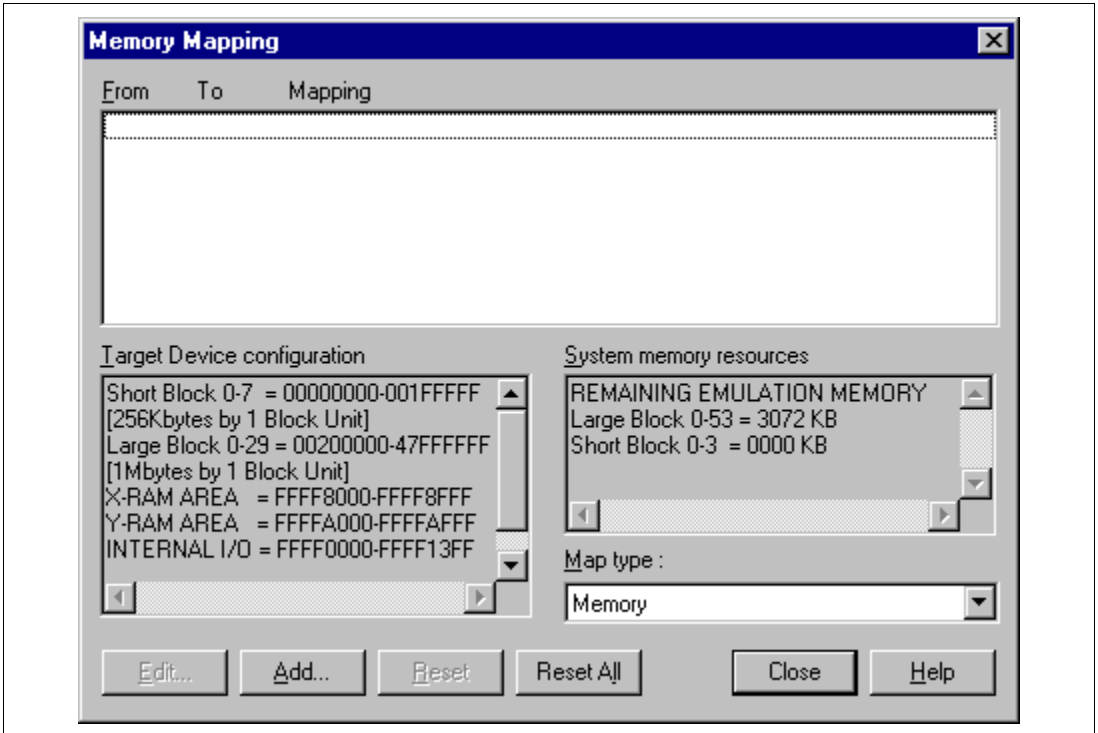


Figure 3.5 [Memory Mapping] Dialog Box (Before Setting)

The emulator can specify any memory area to the following five memory types:

Table 3.3 Memory Type

Memory Type	Description
USER AREA Read-Only	Write-protected user memory area.
USER AREA Guarded	Access-inhibited user memory area.
EMULATION AREA	Emulation memory area.
EMULATION AREA Read-Only	Write-protected emulation memory area.
EMULATION AREA Guarded	Access-inhibited emulation memory area.

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

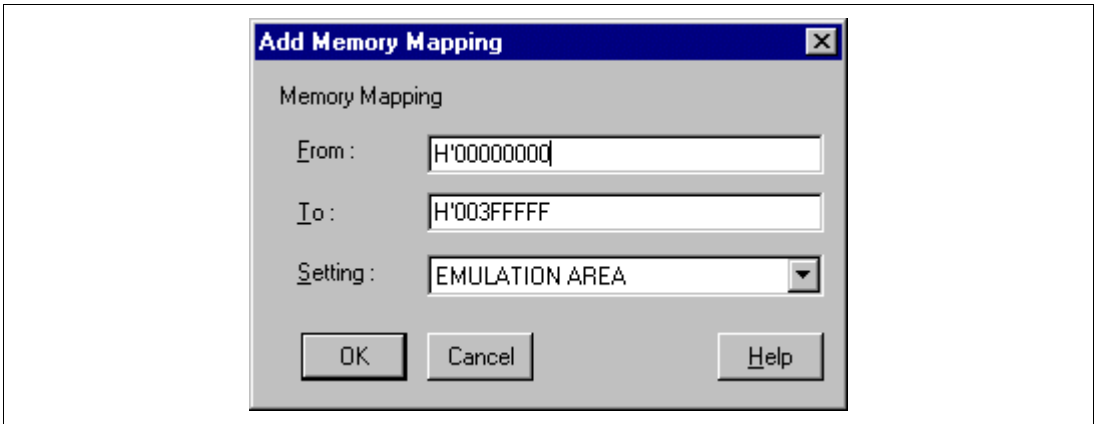


Figure 3.6 [Add Memory Mapping] Dialog Box

For this tutorial, allocate the memory area of addresses ranging from H'00000000 to H'003FFFFFF as an emulation memory area.

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

The [Memory Mapping] dialog box will now show the modified ranges.

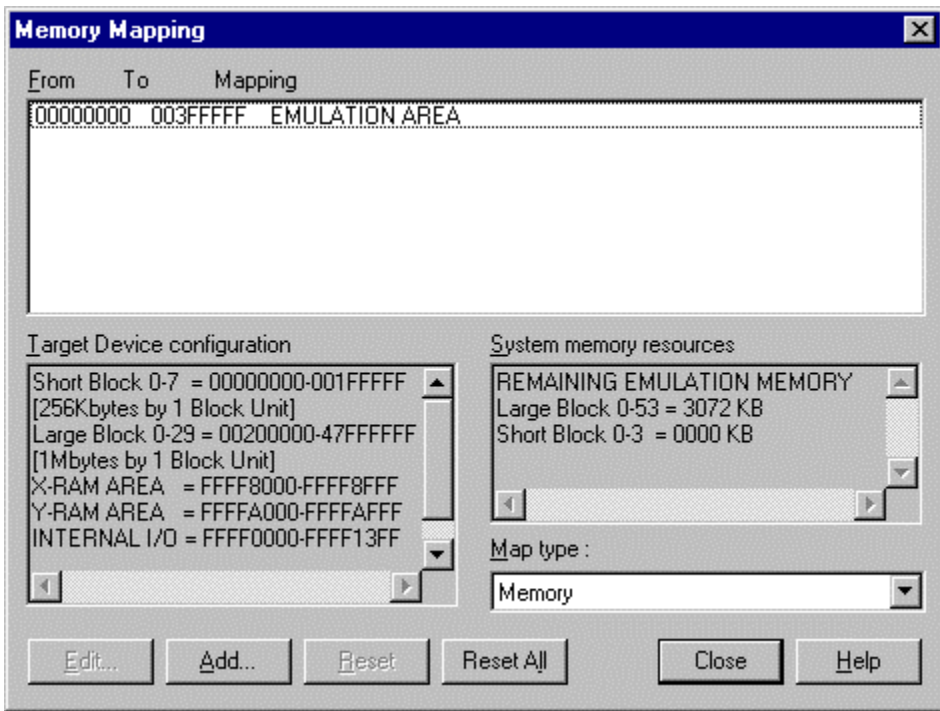


Figure 3.7 [Memory Mapping] Dialog Box (at Setting)

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

3.4 Downloading

3.4.1 Downloading the Tutorial Program

Download the object program .

- Select [Load Program...] from the [File] menu.

The [Load Object File] dialog box is displayed.

- Click the [Browse...] button.

The [Open] dialog box is displayed.

- Select the file TUTORIAL.ABS in the TUTORIAL directory, and click the [Open] button.

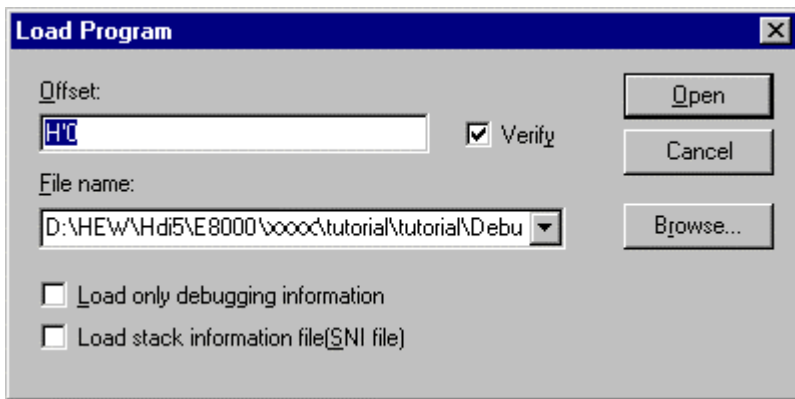


Figure 3.8 [Load Program] Dialog Box

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

When the file has been loaded, the following dialog box displays information about the memory areas that have been filled with the program code.



Figure 3.9 HDI Dialog Box

- Click the [OK] button.

3.4.2 Displaying the Source Program

The [Source] window displays 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.

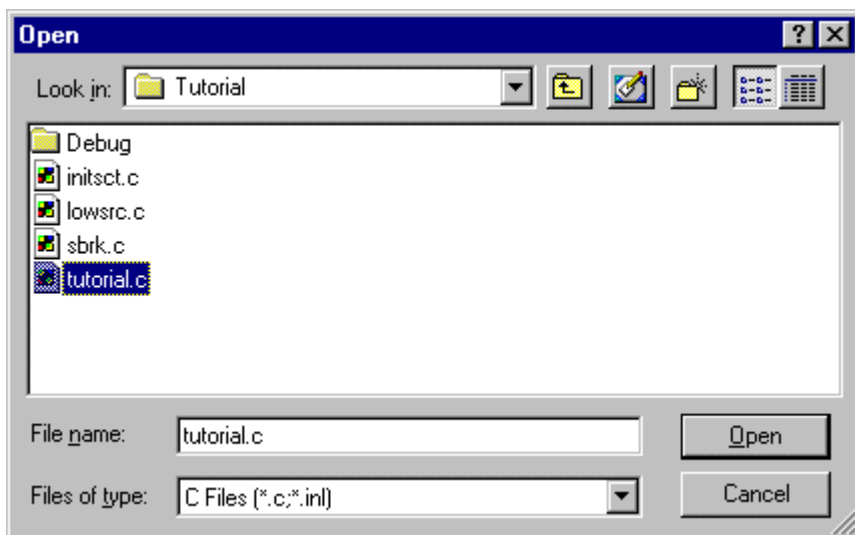


Figure 3.10 [Open] Dialog Box

- Select the file `tutorial.c` and click the [Open] button.

The [Source] window is displayed.

Line	Address	BP	Label	Source
24	00003000		_main	void main(void)
25				{
26				long a[10],min,max;
27				long j;
28				int i;
29				
30	00003002			for(i=0; i<10; i++){
31	0000300a			j = rand();
32	00003012			if(j < 0){
33	00003016			j = -j;
34				}
35	0000301a			a[i] = j;
36				}
37	00003036			sort(a);
38	0000303e			min = a[0];
39	00003042			max = a[9];
40	00003046			min = 0;
41	0000304a			max = 0;
42	0000304e			change(a);
43	00003056			min = a[9];
44	0000305a			max = a[0];
45	0000305e			while (1);
46				}
47				}
48	0000306c		_sort	void sort(long *a)
49				{

Figure 3.11 [Source] Window (Displaying the Source Program)

- If necessary, select the [Font] option from the [Customize] submenu on the [Setup] menu to select an easy-to-see font type and size.

Initially the [Source] window shows the beginning of the main program, but the user can use the scroll bar to scroll through the window to see other statements in the program.

3.5 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 `sort` function call:

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

The word “• Break” will be displayed on the line containing the `sort` function to show that a software breakpoint is set at that address.

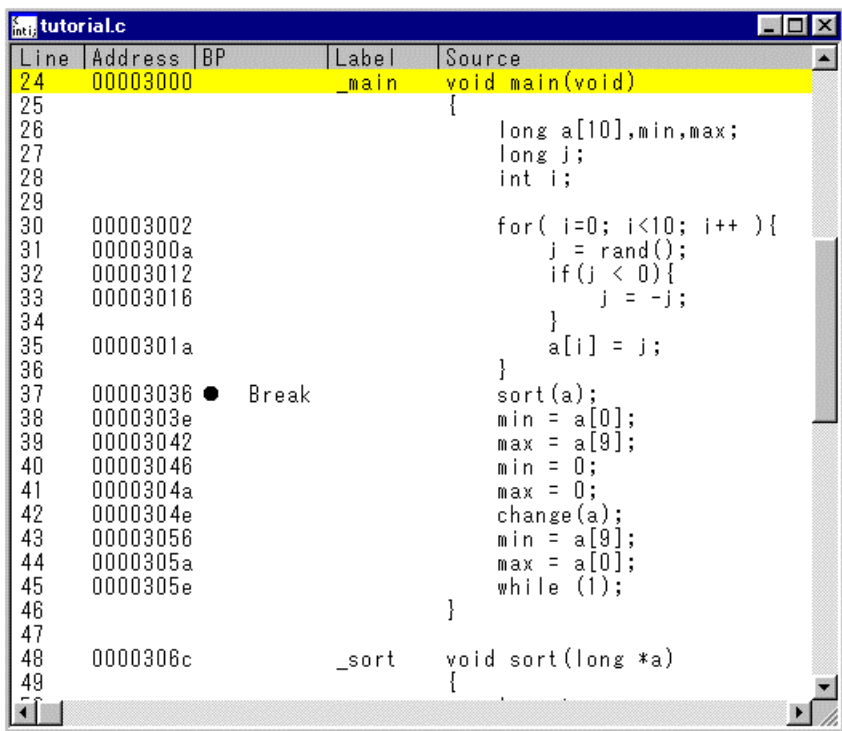


Figure 3.12 [Source] Window (Setting a Software Breakpoint)

3.6 Executing the Program

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



Figure 3.13 [Reset Go] Button

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.

Line	Address	BP	Label	Source
24	00003000		_main	void main(void)
25				{
26				long a[10],min,max;
27				long j;
28				int i;
29				
30	00003002			for(i=0; i<10; i++){
31	0000300a			j = rand();
32	00003012			if(j < 0){
33	00003016			j = -j;
34				}
35	0000301a			a[i] = j;
36				}
37	00003036	● Break		sort(a);
38	0000303e			min = a[0];
39	00003042			max = a[9];
40	00003046			min = 0;
41	0000304a			max = 0;
42	0000304e			change(a);
43	00003056			min = a[9];
44	0000305a			max = a[0];
45	0000305e			while (1);
46				}
47				
48	0000306c		_sort	void sort(long *a)
49				{

Figure 3.14 [Source] Window (Break State)

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

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

The [System Status] window is displayed.

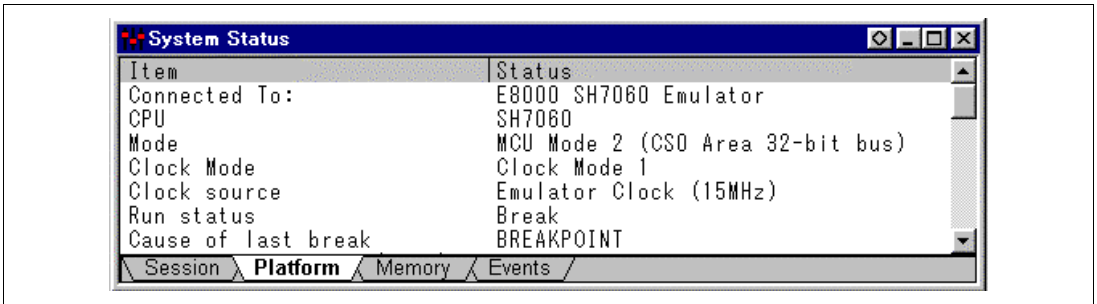


Figure 3.15 [System Status] Window

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

3.7 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.

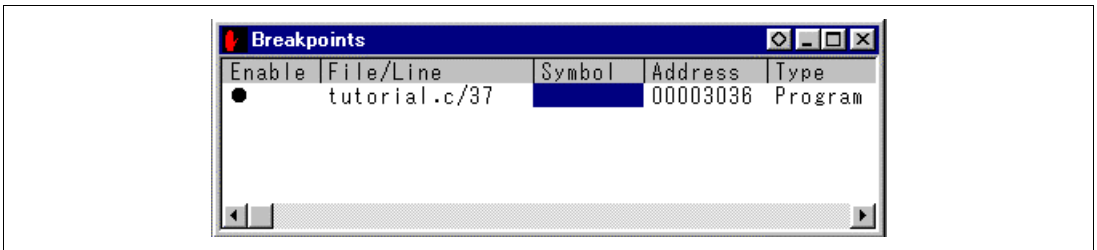


Figure 3.16 [Breakpoints] Window

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

- Close the [Breakpoints] window.

3.8 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 `main` function in word size:

- Select [Memory ...] from the [View] menu, input `main` in the [Address] edit box, and set the [Format] combo box as [Word].

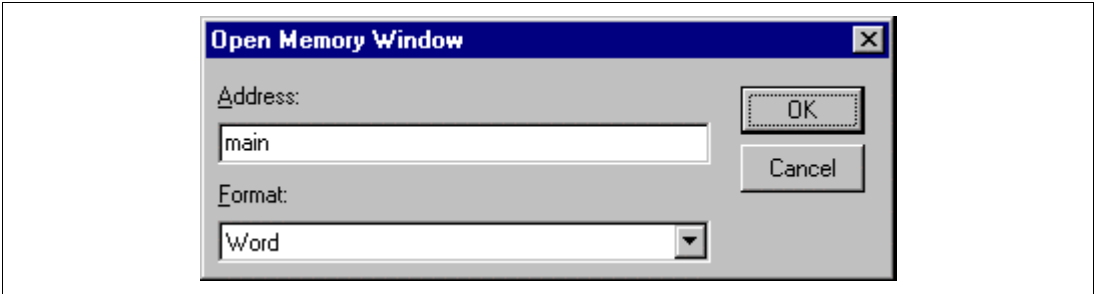
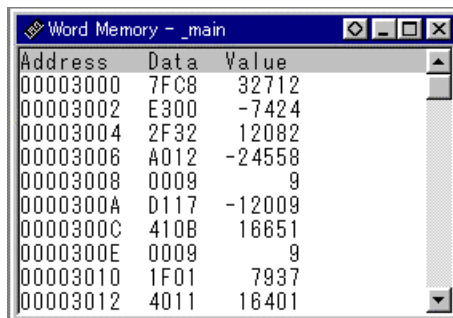


Figure 3.17 [Open Memory Window] Dialog Box

- Click the [OK] button.

The [Word Memory] window showing the specified area of memory is displayed.



The screenshot shows a window titled "Word Memory - _main" with a table of memory data. The table has three columns: "Address", "Data", and "Value". The data is as follows:

Address	Data	Value
00003000	7FC8	32712
00003002	E300	-7424
00003004	2F32	12082
00003006	A012	-24558
00003008	0009	9
0000300A	D117	-12009
0000300C	4108	16651
0000300E	0009	9
00003010	1F01	7937
00003012	4011	16401

Figure 3.18 [Word Memory] Window

3.9 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 following dialog box is displayed.

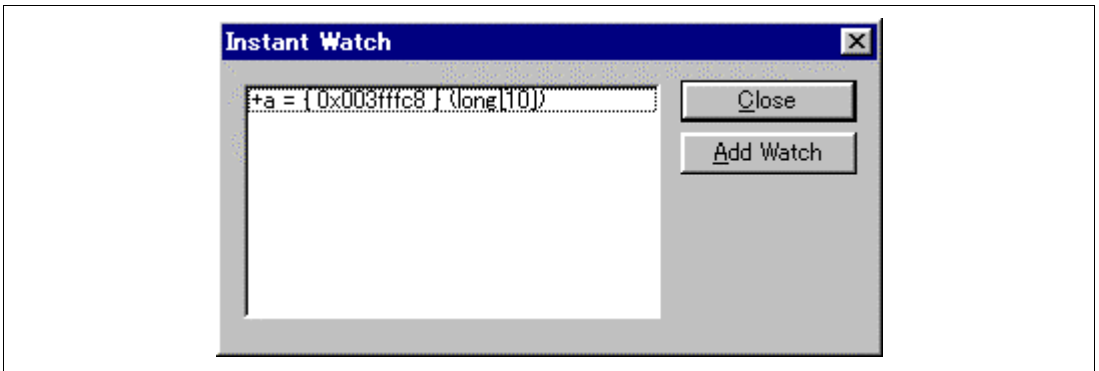


Figure 3.19 [Instant Watch] Dialog Box

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

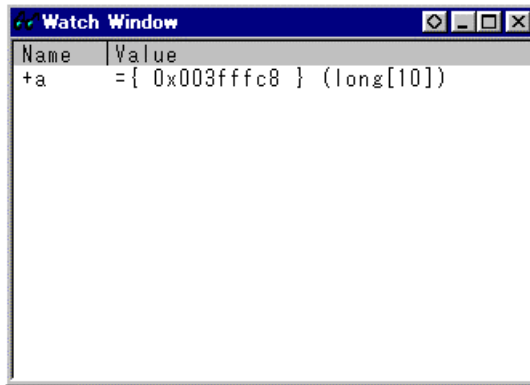


Figure 3.20 [Watch] Window (Displaying the Array)

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

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

The following dialog box is displayed.

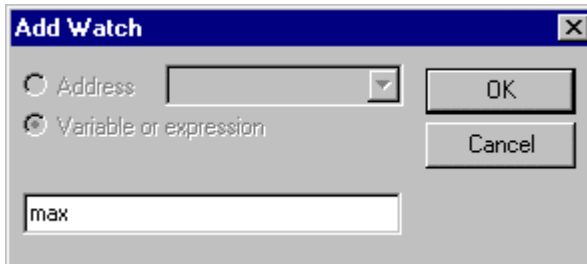


Figure 3.21 [Add Watch] Dialog Box

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

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

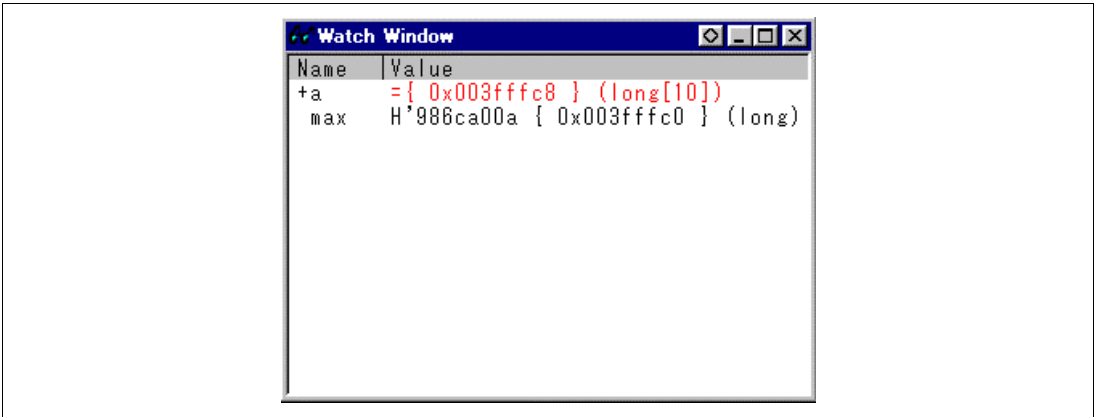


Figure 3.22 [Watch] Window (Displaying the Variable)

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

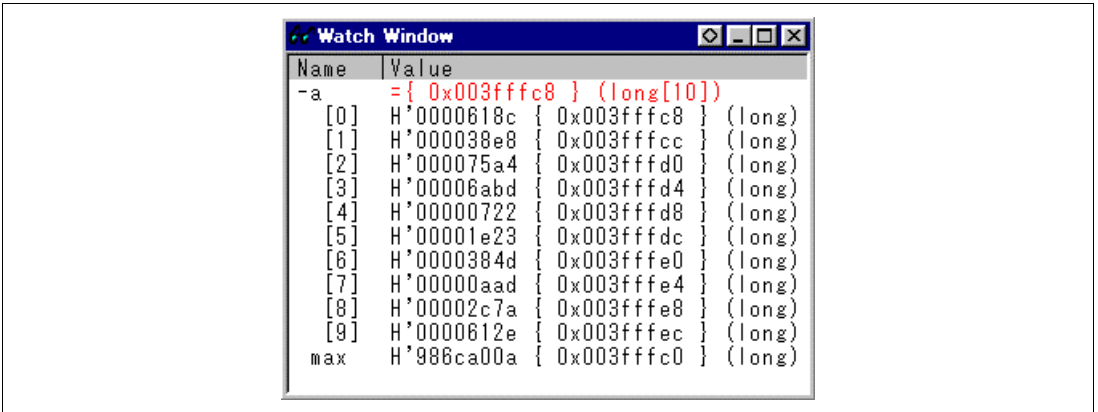


Figure 3.23 [Watch] Window (Displaying Array Elements)

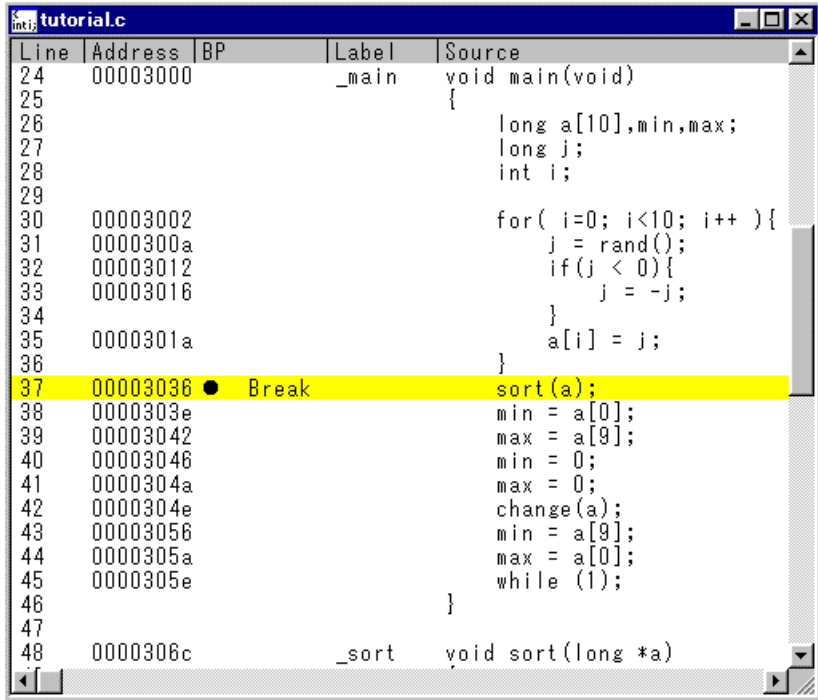
3.10 Stepping Through a Program

The HDI provides a range of step menu commands that allow efficient program debugging.

Table 3.4 Step Command

Command	Description
Step In	Steps through statements, including those in a function.
Step Over	Steps through the statements with a function call as one 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.



Line	Address	BP	Label	Source
24	00003000		_main	void main(void)
25				{
26				long a[10],min,max;
27				long j;
28				int i;
29				
30	00003002			for(i=0; i<10; i++){
31	0000300a			j = rand();
32	00003012			if(j < 0){
33	00003016			j = -j;
34				}
35	0000301a			a[i] = j;
36				}
37	00003036	● Break		sort(a);
38	0000303e			min = a[0];
39	00003042			max = a[9];
40	00003046			min = 0;
41	0000304a			max = 0;
42	0000304e			change(a);
43	00003056			min = a[9];
44	0000305a			max = a[0];
45	0000305e			while (1);
46				}
47				
48	0000306c		_sort	void sort(long *a)

Figure 3.24 [Source] Window (Step Execution)

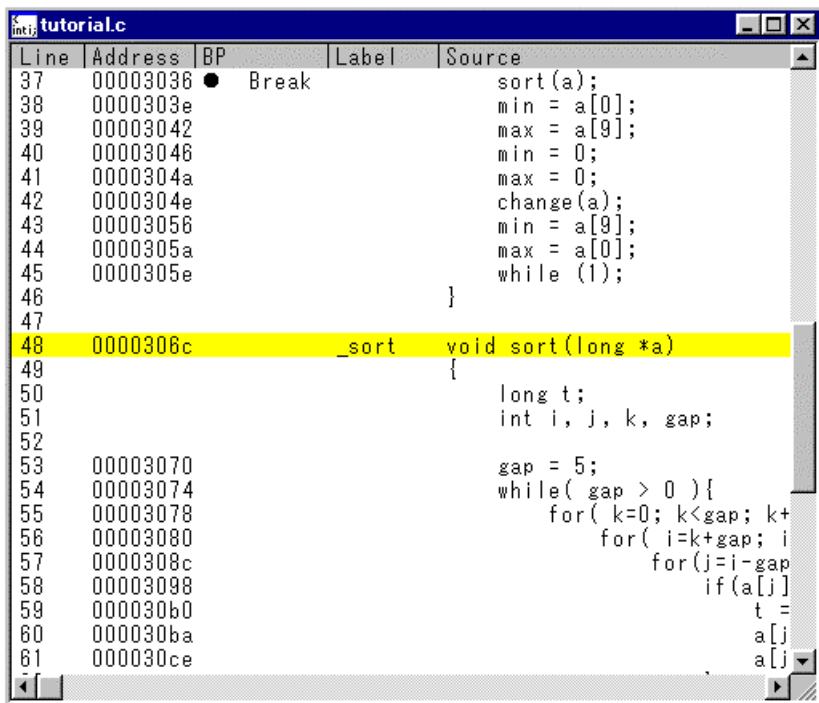
3.10.1 Executing [Step In] Command

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

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



Figure 3.25 [Step In] Button

A screenshot of a debugger's source window titled 'tutorial.c'. The window displays a table with columns for Line, Address, BP, Label, and Source. Line 48 is highlighted in yellow. The source code shows a call to 'sort(a)' at line 37, followed by initialization of 'min' and 'max' at lines 38-40, and a 'while' loop starting at line 41. The 'while' loop body contains 'change(a)' at line 42, and re-initialization of 'min' and 'max' at lines 43-44. The 'while' loop ends at line 47. Line 48 is the start of the 'void sort(long *a)' function. The function body includes a 'long t;' declaration at line 50, an 'int i, j, k, gap;' declaration at line 51, and a 'while(gap > 0)' loop starting at line 53. Inside the 'while' loop, there is a 'for(k=0; k<gap; k++)' loop starting at line 55, which contains a 'for(j=i-gap; j<i; j++)' loop starting at line 57. Inside the inner 'for' loop, there is an 'if(a[j] > a[j+1])' condition starting at line 58, followed by a swap operation at lines 59-61: 't = a[j]; a[j] = a[j+1]; a[j+1] = t;'.

Line	Address	BP	Label	Source
37	00003036	● Break		sort(a);
38	0000303e			min = a[0];
39	00003042			max = a[9];
40	00003046			min = 0;
41	0000304a			max = 0;
42	0000304e			change(a);
43	00003056			min = a[9];
44	0000305a			max = a[0];
45	0000305e			while (1);
46				}
47				}
48	0000306c		_sort	void sort(long *a)
49				{
50				long t;
51				int i, j, k, gap;
52				
53	00003070			gap = 5;
54	00003074			while(gap > 0){
55	00003078			for(k=0; k<gap; k+
56	00003080			for(i=k+gap; i
57	0000308c			for(j=i-gap
58	00003098			if(a[j]
59	000030b0			t =
60	000030ba			a[j
61	000030ce			a[j

Figure 3.26 [Source] Window (Step In)

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

3.10.2 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 3.27 [Step Out] Button

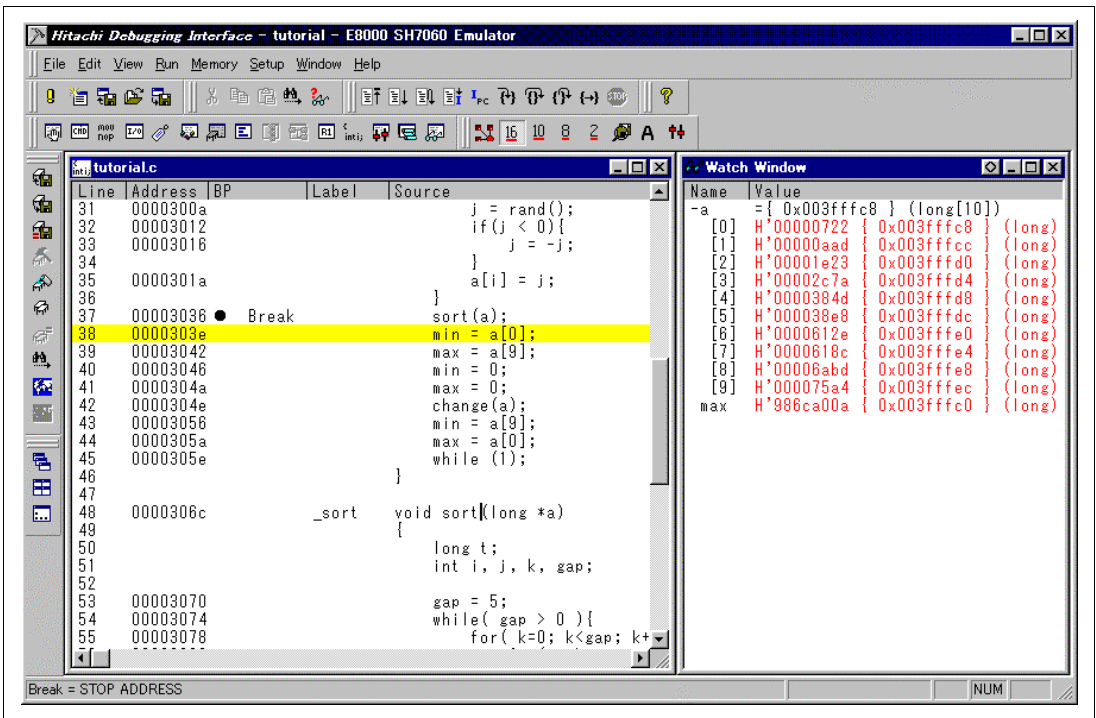


Figure 3.28 [Source] Window (Step Out)

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

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

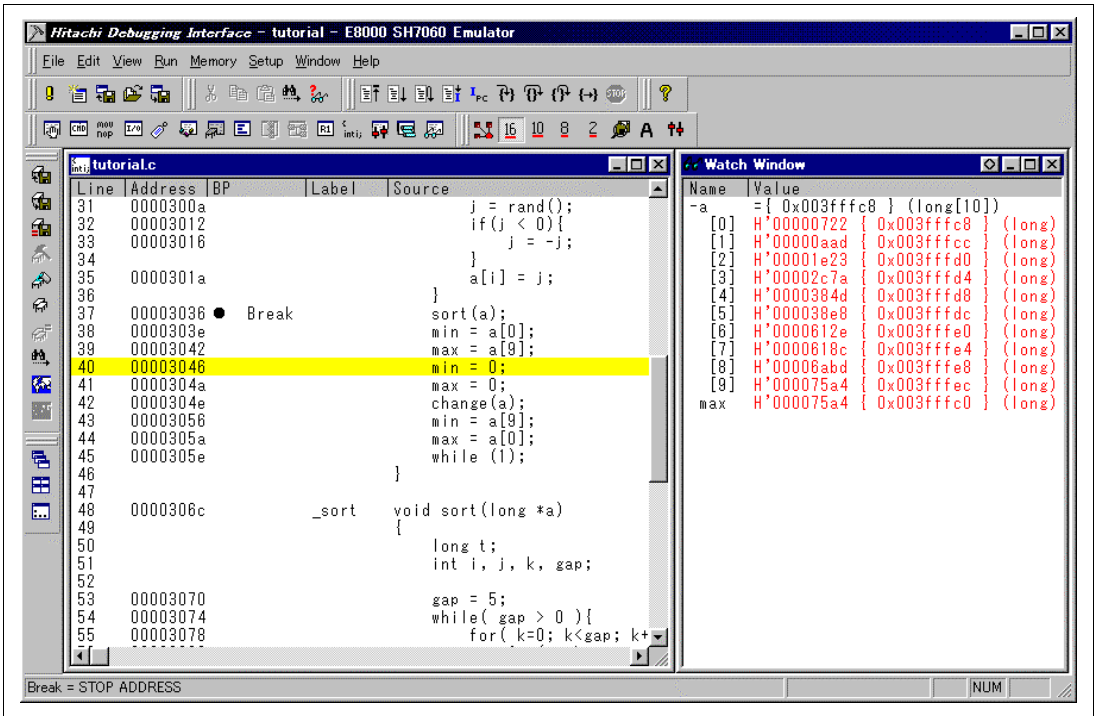


Figure 3.29 [Source] Window (Step Out -> Step In)

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

3.10.3 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.

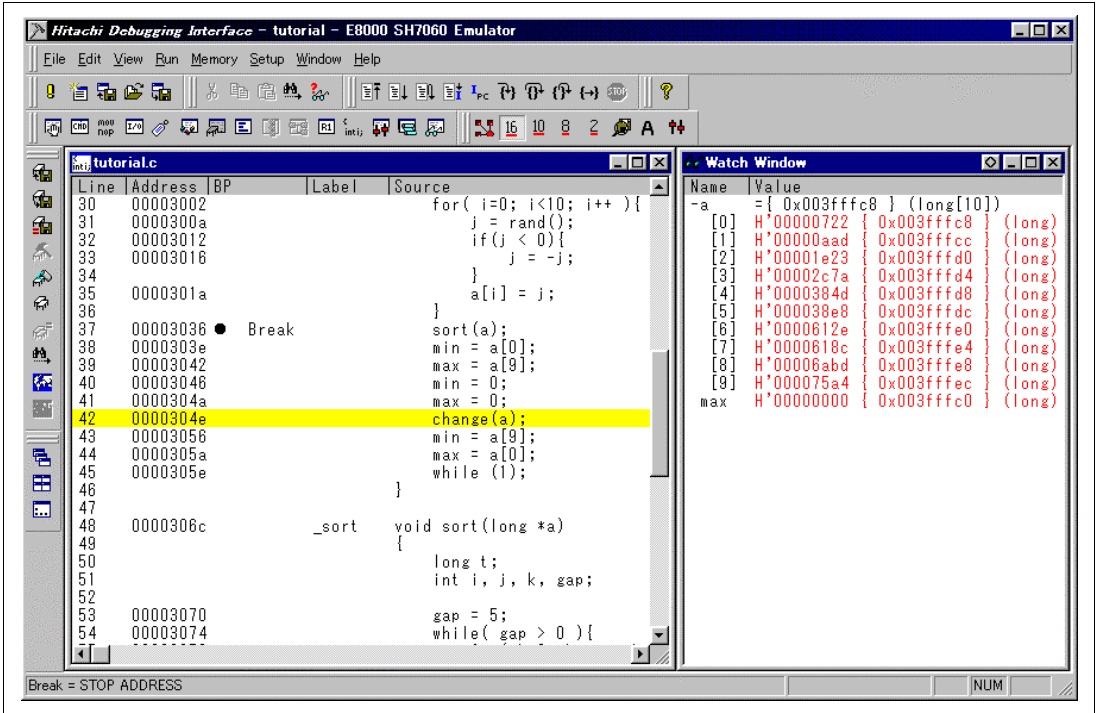


Figure 3.30 [Source] Window (Before Step Over Execution)

- To perform step execution for the statements in the change function at one time, select [Step Over] from the [Run] menu, or click the [Step Over] button in the toolbar.



Figure 3.31 [Step Over] Button

- Execute a line that calls the change function as a single step and stop execution at the next line.

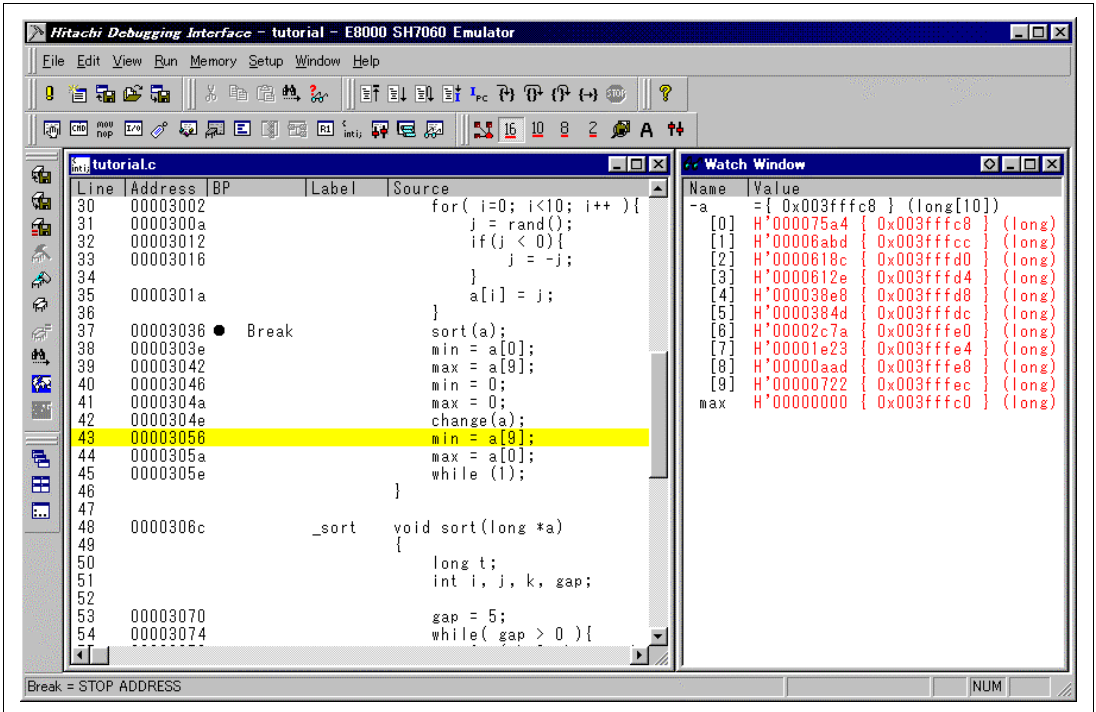


Figure 3.32 [Source] Window (Step Over)

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

3.11 Displaying Local Variables

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

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

The [Locals] window is displayed. Initially, the [Locals] window is empty because local variables have not yet been declared.

- Select [Step In] from the [Run] menu to execute one more step.

The [Locals] window will now show the local variables and their values.

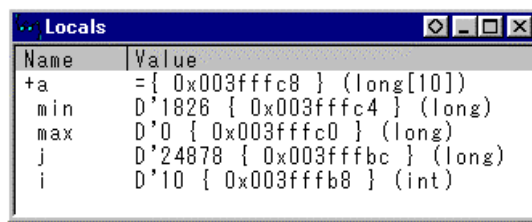


Figure 3.33 [Locals] Window

- Double-click the + symbol to the left of array `a` in the [Locals] window to display the elements of array `a`.
- Refer to the elements of array `a` before and after the execution of the `sort` function, and confirm that random data is sorted in ascending or descending order.

3.12 Setting the Hardware Break Conditions

The emulator has powerful hardware break functions. In the HDI, these hardware break conditions can be set by using dialog boxes. The dialog boxes for setting hardware break conditions, and the corresponding break conditions, are described below.

Table 3.5 Dialog Boxes for Setting Hardware Break Conditions

Dialog Box	Function						
	Address Bus Condition (Address)	Data Bus Condition (Data)	Bus State Condition (Bus State)	External Probe Signal Condition (Probe)	Interrupt Signal Condition (Interrupt)	Satisfaction Count (Count)	DELAY Condition (Delay) ^{*3}
[Break Condition UBC1] dialog box	O	O	O	X	X	O	X
[Break Condition UBC2/3/4] dialog box	O	O	O	X	X	X	X
[Break Condition A] dialog box ^{*2}	O	O	O	O	O	X	X
[Break Condition B] dialog box ^{*2}	O	O	O	O	O	O	O
[Break Condition C] dialog box ^{*2}	O	X	O	X	X	X	X

Notes: 1. O: Can be set in the dialog box.

X: Cannot be set in the dialog box.

2. Eight breakpoints can be set independently in each of the [Break Condition A,B,C] dialog boxes.
3. The DELAY condition in the [Break Condition B] dialog box can be set for only Break Condition B7.

Table 3.6 Main Break Conditions

Break Condition	Description
Address bus condition (Address)	Breaks when the CPU address bus value matches the specified conditions.
Data bus condition (Data)	Breaks when the CPU data bus value matches the specified conditions. Byte, word, or longword can be specified as the access data size.
Bus state condition (Bus State)	There are two bus state condition settings: Read/write condition: Breaks when the RD or RDWR signal level of the CPU matches the specified conditions. Bus state condition: Breaks when the conditions of CPU memory access cycle or vector fetch cycle match the specified conditions.
External probe signal condition (Probe)	Breaks when an external probe signal (PRB1 to PRB4) level matches the specified conditions.
Interrupt signal condition (Interrupt)	Breaks when the NMI signal level matches the specified conditions.
Satisfaction count (Count)	Breaks when the above conditions have been satisfied the number of times specified in this condition. (A maximum count of 65,535 can be specified.)
DELAY condition (Delay)	Breaks when the above conditions have been satisfied and the bus cycles specified in this condition have been executed. (A maximum of 32,767 bus cycles can be specified.)

An example is given below in which the address bus condition and read cycles for the bus state condition are set in Break Condition A as hardware break conditions.

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

The [Breakpoints] window is displayed.

- Click the mouse right button in the [Breakpoints] window.

The popup menu is displayed.

- Click the [Add] button.

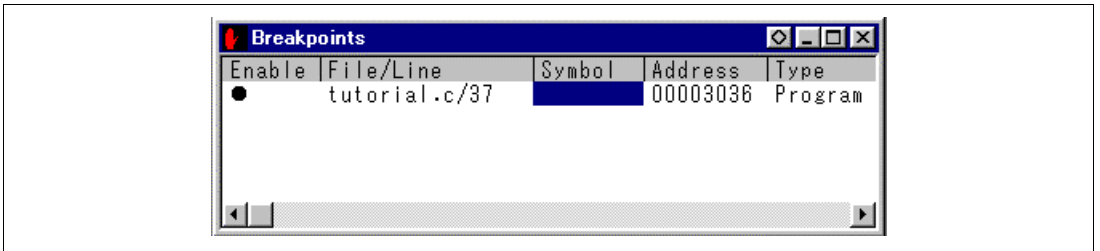


Figure 3.34 [Breakpoints] Window (Before Hardware Break Condition Setting)

The [Break] dialog box is displayed. For hardware break conditions, the [Break] dialog box pages required for the setting must be selected.

- Select [Condition A] to display the [Condition A] page.

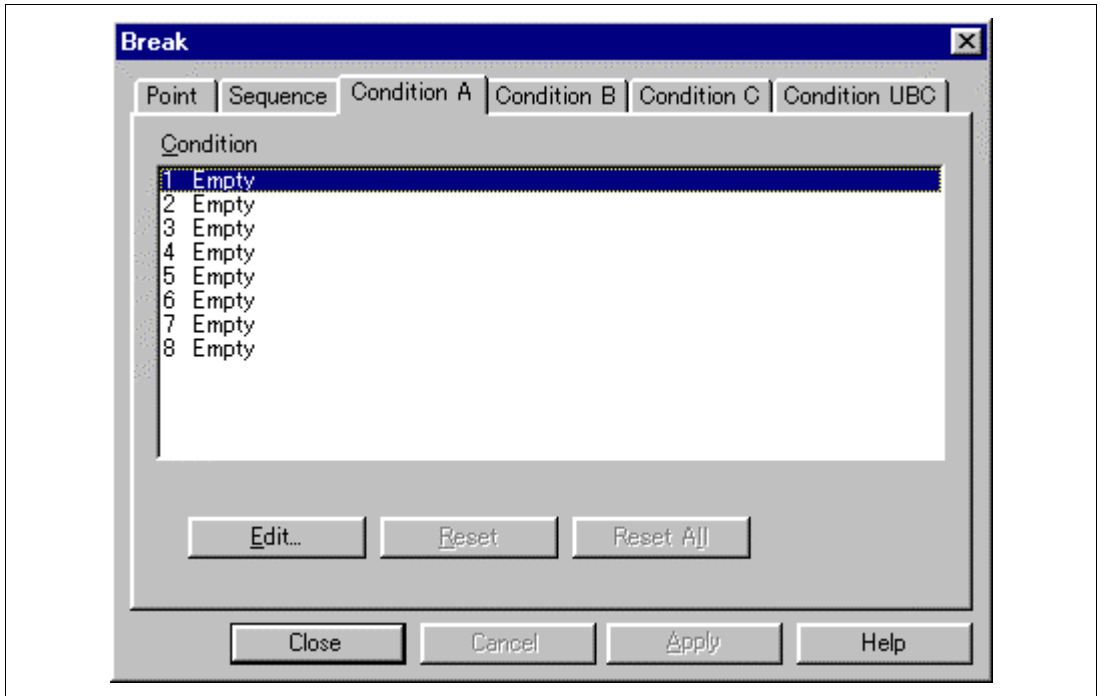


Figure 3.35 [Condition A] Page ([Break] Dialog Box)

Up to eight breakpoints can be set independently for each of the Break Condition A, B, C hardware break conditions. In the example, one point is set for the Break Condition A hardware break conditions.

- Highlight the first point in the [Condition] list box.
- Click the [Edit...] button.

The [Break Condition A1] dialog box is displayed.

- Clear the [Don't Care] check box in the [Address] page.
- Select the [Address] radio button and input address **H'3058** as the value in the [Start] edit box.

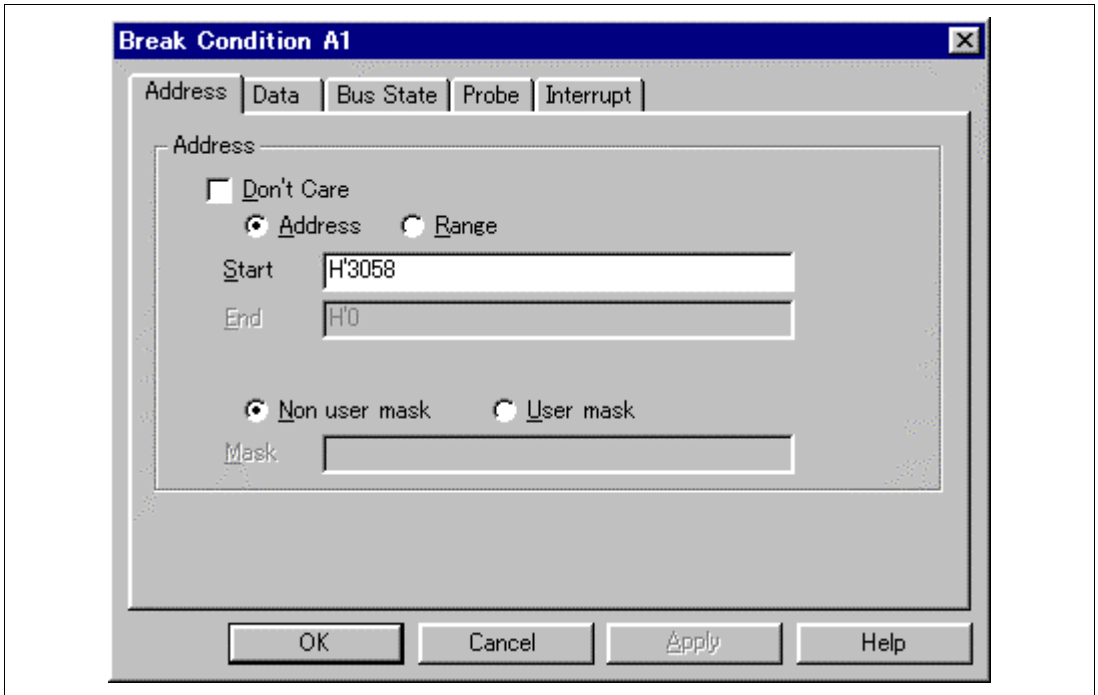


Figure 3.36 [Address] Page ([Break Condition A1] Dialog Box)

- Select [Bus State] to display the [Bus State] page.

- Select the [Read] radio button.

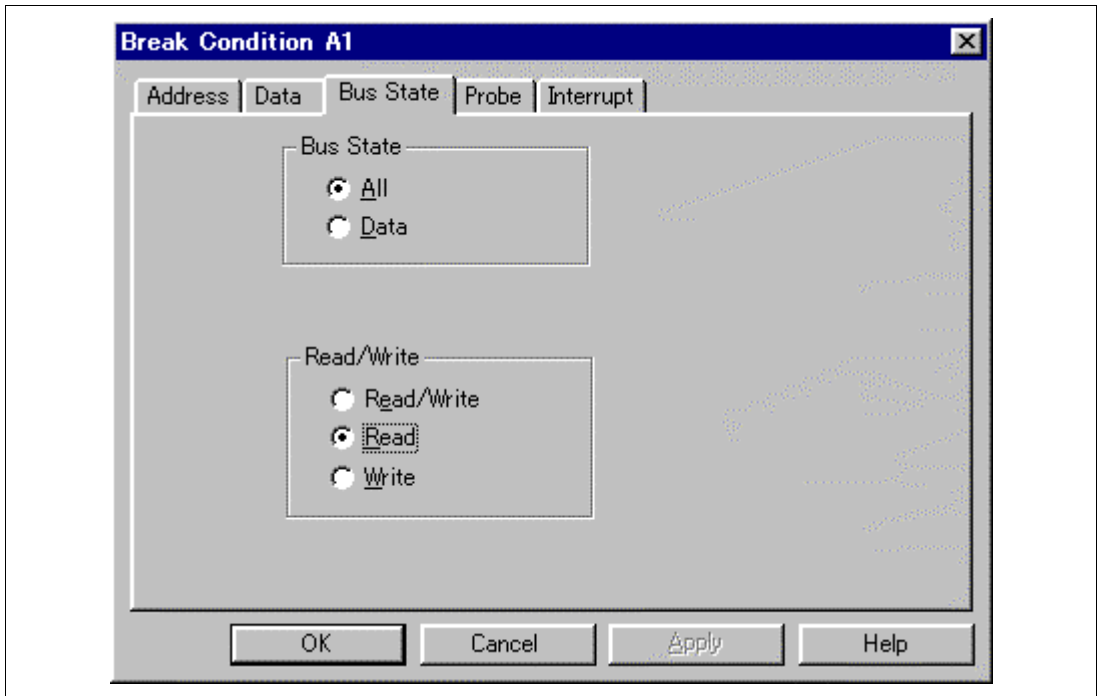


Figure 3.37 [Bus State] Page ([Break Condition A1] Dialog Box)

- Click the [OK] button.

The [Break] dialog box is displayed, and the first point display in the [Condition] list box changes from “Empty” to “Enable”.

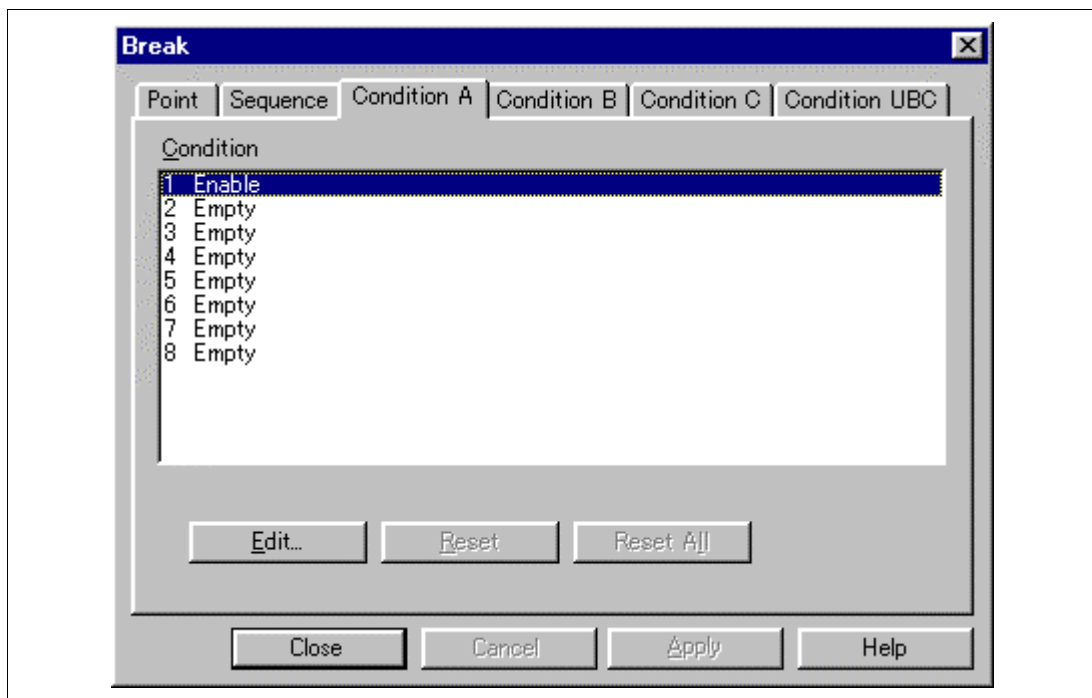


Figure 3.38 [Break] Dialog Box (After Hardware Break Condition Setting)

- Click the [Close] button.

The newly set hardware breakpoint is displayed in the [Breakpoints] window. With this setting, “Break Condition A1” is displayed in [Type] in the [Breakpoints] window.

This completes the setting of the Break Condition A1 hardware break conditions. A break will occur when address H'3058 is accessed in a read cycle during the program execution.

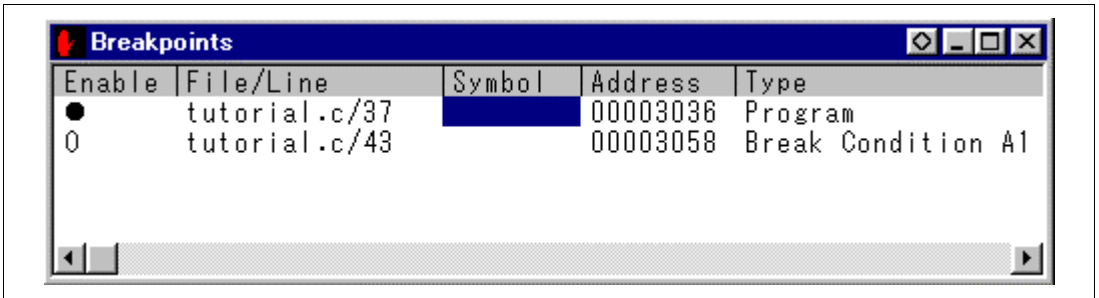


Figure 3.39 [Breakpoints] Window (at Break Condition A Setting)

3.13 Setting the Sequential Break Conditions

The emulator has powerful sequential break functions. In the HDI, these sequential break conditions can be set by using dialog boxes. The dialog boxes for setting sequential break conditions, and the corresponding sequential break functions, are described below.

Table 3.7 Dialog Boxes for Setting Sequential Break Conditions

Dialog Box	Function						
	Address Bus Condition (Address)	Data Bus Condition (Data)	Bus State Condition (Bus State)	External Probe Signal Condition (Probe)	Interrupt Signal Condition (Interrupt)	Satisfaction Count (Count)	DELAY Condition (Delay)
[Break Condition UBC1] dialog box	O	O	O	X	X	O	X
[Break Condition UBC2] dialog box	O	O	O	X	X	X	X
[Break Condition UBC3] dialog box	O	O	O	X	X	X	X
[Break Condition UBC4] dialog box	O	O	O	X	X	X	X
[Break Sequence] dialog box	O	X	X	X	X	X	X

Note: O: Can be set in the dialog box.

X: Cannot be set in the dialog box.

Table 3.8 Main Sequential Break Functions

Sequential Break Function	Description
Break Sequence	<p>A sequential break function using software breaks.</p> <p>Up to seven pass breakpoint addresses and one reset point address can be set. When all the breakpoint addresses that have been set are passed in sequence, the program is stopped.</p>
Break Condition UBC1, 2, 3, 4	<p>Sequential break functions by combining hardware break conditions, i.e., Break Condition UBC1, 2, 3, and 4.</p> <p>The following combinations of conditions can be selected:</p> <ul style="list-style-type: none">• Program execution is halted when conditions are satisfied in the order of UBC2 and UBC1. (Sequential break mode UBC 2 -> 1)• Program execution is halted when conditions are satisfied in the order of UBC3, UBC2, and UBC1. (Sequential break mode UBC 3 -> 2 -> 1)• Program execution is halted when conditions are satisfied in the order of UBC4, UBC3, UBC2, and UBC1. (Sequential break mode UBC 4 -> 3 -> 2 -> 1) <p>One reset point address can be set for each combination above.</p>

After passing the reset point addresses, these functions make sequential break conditions already passed so far invalid and resume checking break conditions from the first one.

An example is given below in which [Sequential break mode UBC2 -> 1] is set as the sequential break function.

Before executing the program, change the [Configuration] dialog box. When not changing it, the sequential break does not function.

- Select [Configure Platform...] from the [Setup] menu.

The [Configuration] dialog box will appear.

- Select [Sequential break mode UBC2 -> 1] from the [Emulation mode] combo box and click the [OK] button.

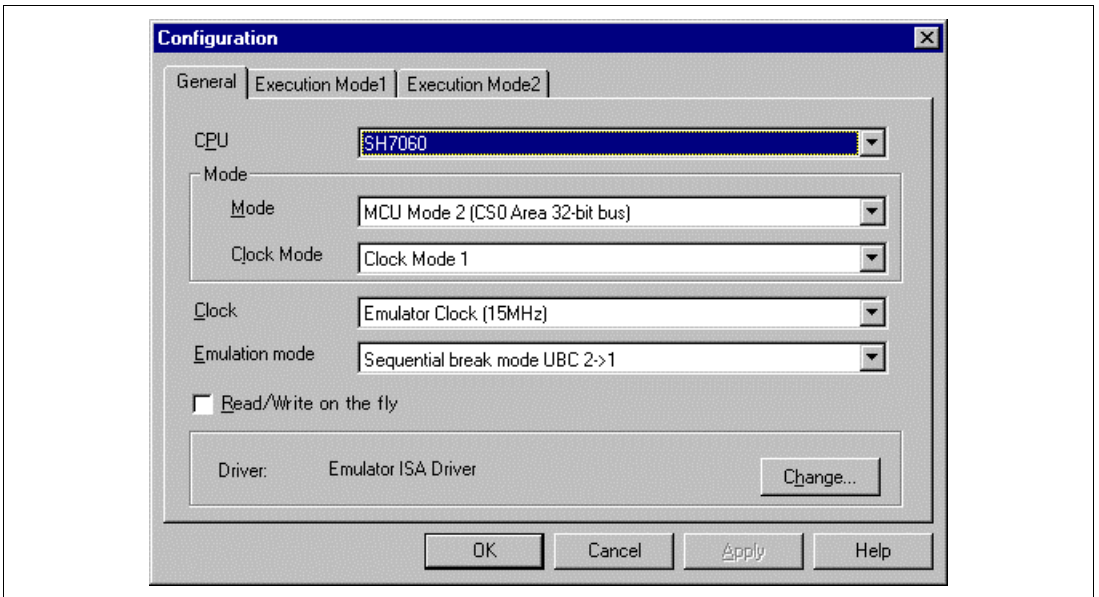


Figure 3.40 [Configuration] Dialog Box (at Sequential Break Condition Setting)

Click the [OK] button.

An example is given below in which Break Condition UBC1 and Break Condition UBC2 of the sequential break conditions are set. Set break conditions as follows:

Break condition 1: A break is executed when address H'3056 is accessed in a read cycle. (Break Condition UBC1)

Break condition 2: A break is executed when address H'3046 is accessed in a read cycle. (Break Condition UBC2)

After break condition 2 is satisfied and break condition 1 is satisfied in succession, a program being executed will stop.

Then, set the sequential break conditions.

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

The [Breakpoints] window is displayed.

- Click the mouse right button in the [Breakpoints] window.

The popup menu is displayed.

- Select [Delete All] to clear all the set break points.
- Display the popup menu again and click the [Add...] button.

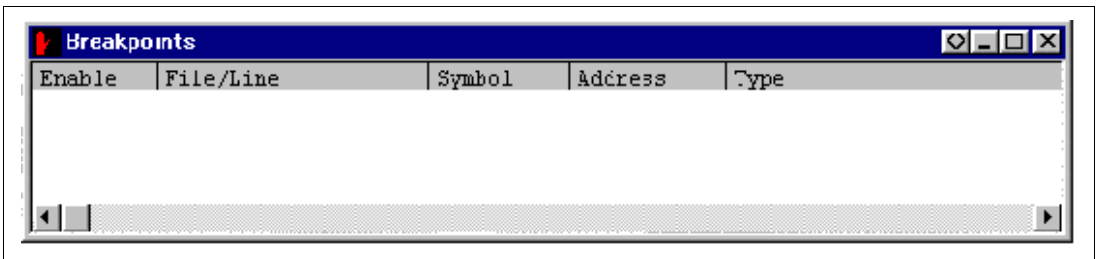


Figure 3.41 [Breakpoints] Window (Before Sequential Break Condition Setting)

The [Break] dialog box will appear.

- To set sequential break conditions, select [Condition UBC] and display the [Condition UBC] page.

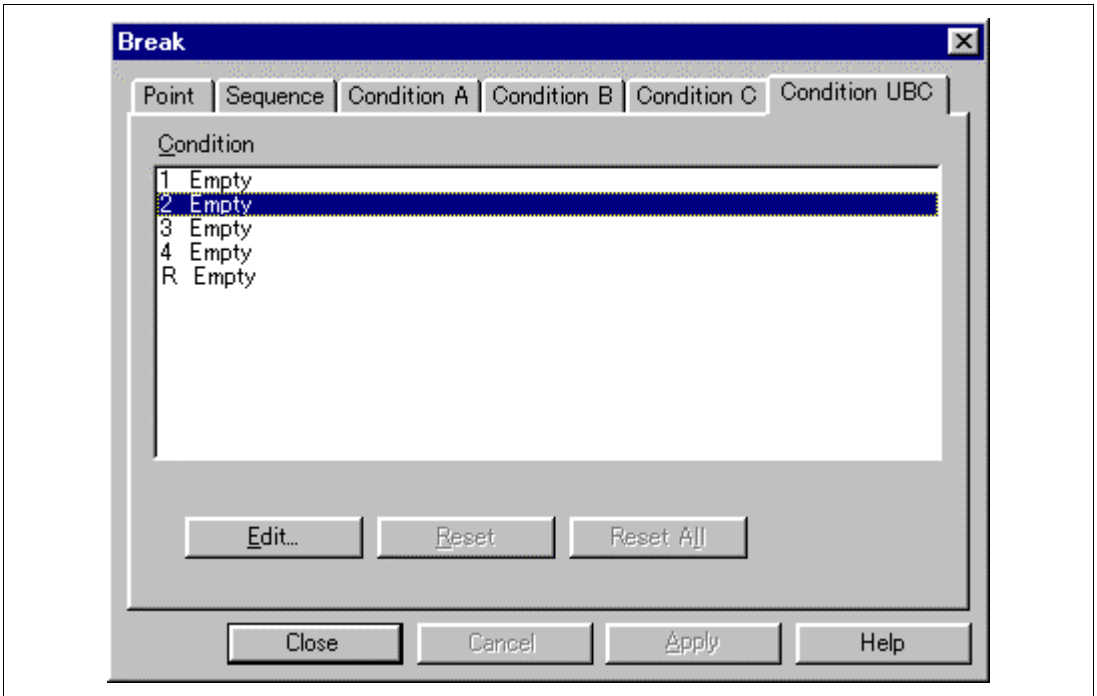


Figure 3.42 [Break] Dialog Box ([Break Condition UBC] Page)

Set break condition 2 of the sequential break conditions to Break Condition UBC2 and set break condition 1 to Break Condition UBC1.

- Click the second line in the [Condition] list box to highlight it.
- Click the [Edit...] button.

The [Break Condition UBC2] dialog box will appear.

- Make the [Don't Care] check box in the [Address] page invalid.
- Select the [Address] radio button and enter the address **H'3046** as the value in the [Address] edit box.

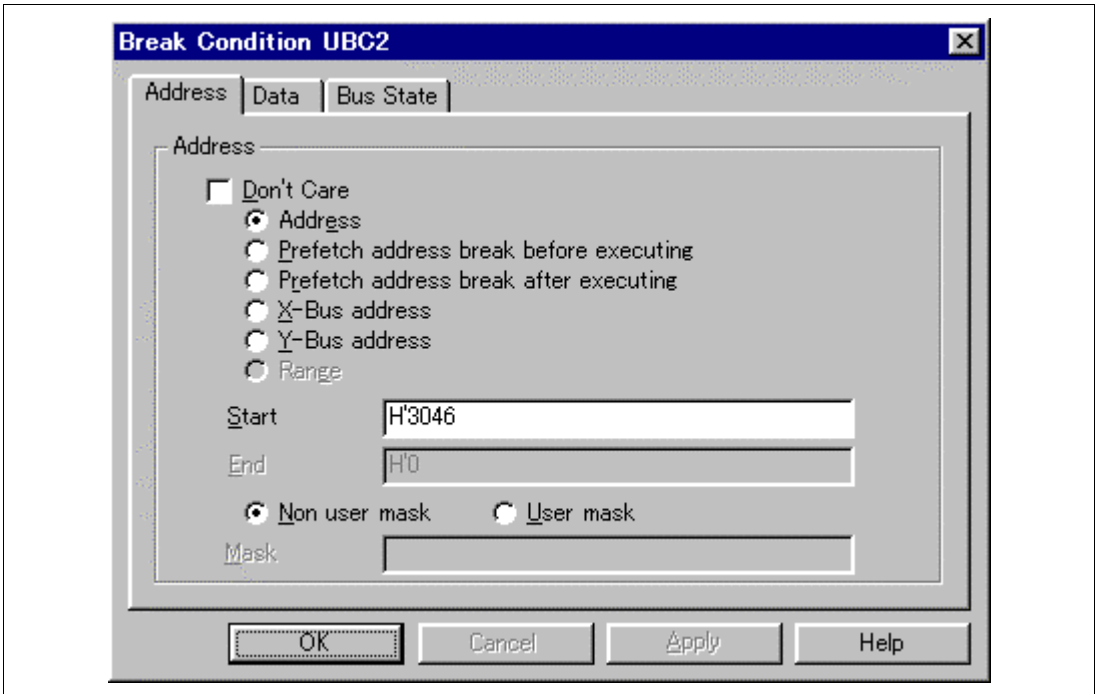


Figure 3.43 [Break Condition UBC2] Dialog Box (Condition 2 [Address] Page)

- Select [Bus State] to display the [Bus State] page.

- Select [Read] radio button.

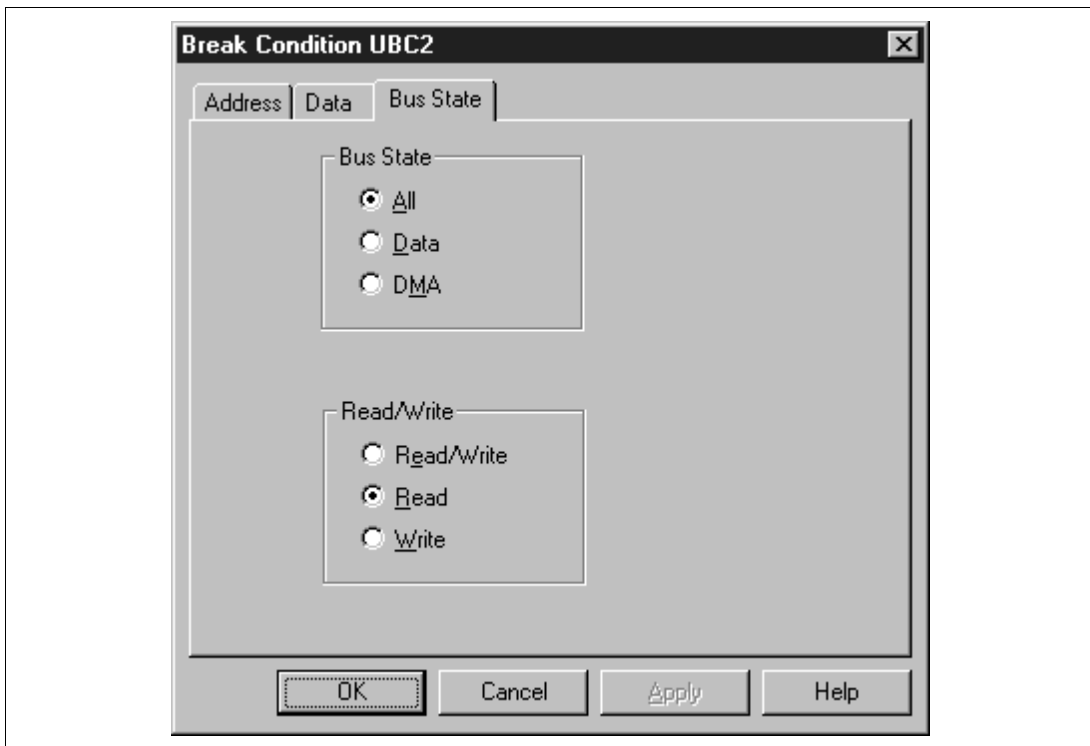


Figure 3.44 [Break Condition UBC2] Dialog Box (Condition 2 [Bus State] Page)

- Click the [OK] button.

The [Break] dialog box is displayed, and the second point display in the [Condition] list box changes from “Empty” to “Enable”.

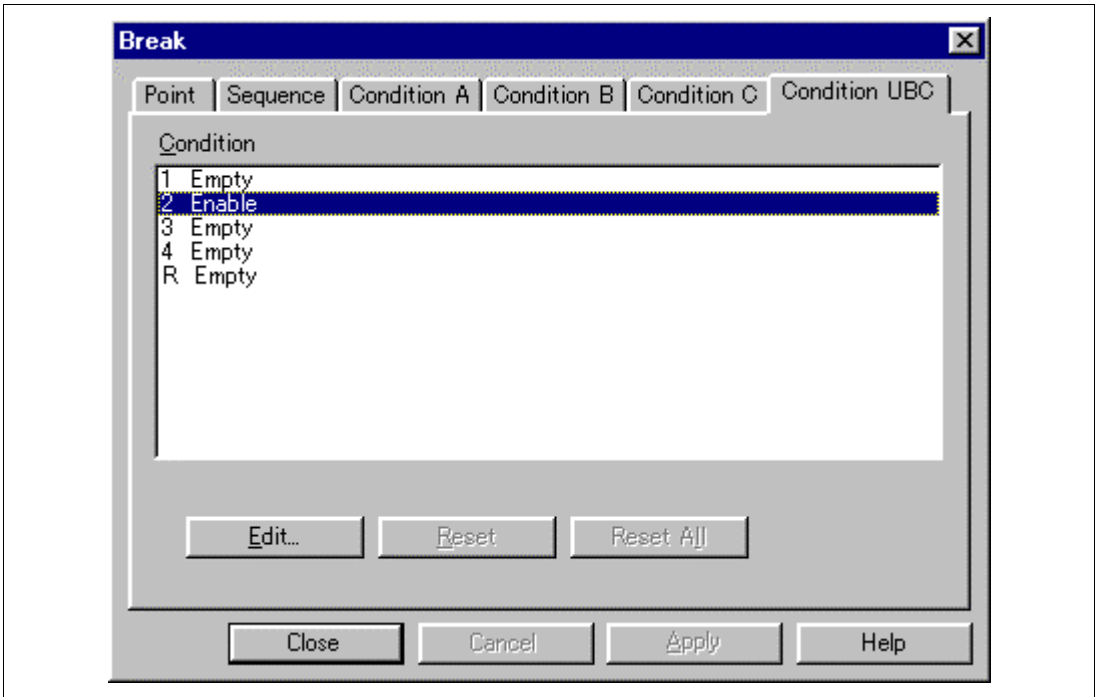


Figure 3.45 [Break] Dialog Box (After [Break Condition UBC2] Condition Setting)

This completes the setting of break condition 2. Next, Set break condition 1 as follows:

- Highlight the first point in the [Condition] list box.
- Click the [Edit...] button.

The [Break Condition UBC1] dialog box is displayed.

Break condition 1 can be set in the same way as for break condition 2.

- After setting break conditions 1 and 2, click the [Close] button.

“Break Condition UBC1” and “Break Condition UBC2” are displayed in [Type] in the [Breakpoints] window.

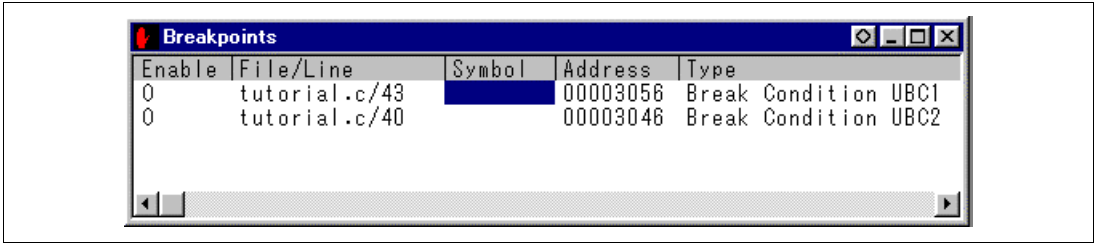
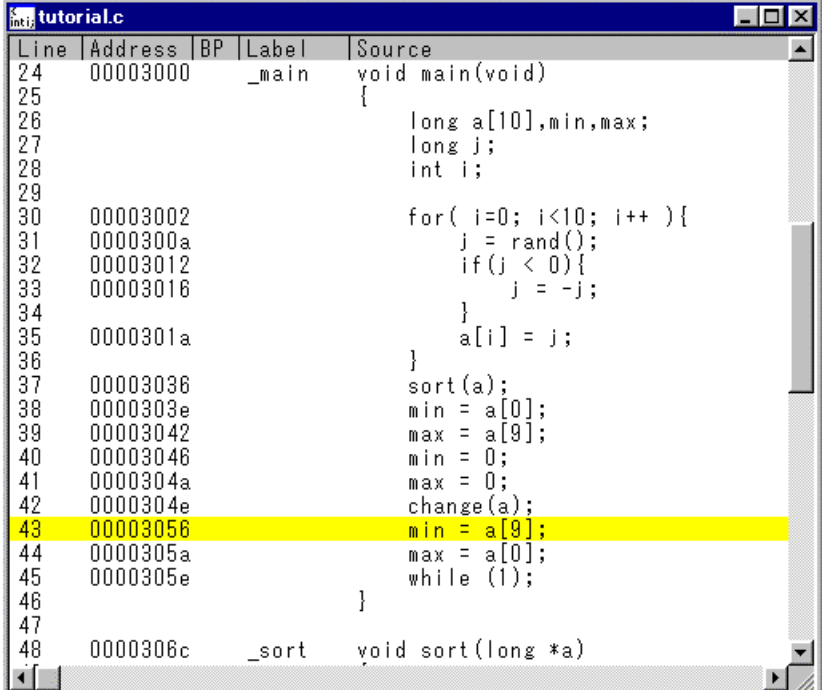


Figure 3.46 [Breakpoints] Window (After Sequential Break Condition Setting)

- Click the [Reset Go] button.

The program is executed until the conditions of Break Condition UBC1 are satisfied and then halts.

The following [Source] window is then displayed.



Line	Address	BP	Label	Source
24	00003000		_main	void main(void)
25				{
26				long a[10],min,max;
27				long j;
28				int i;
29				
30	00003002			for(i=0; i<10; i++){
31	0000300a			j = rand();
32	00003012			if(j < 0){
33	00003016			j = -j;
34				}
35	0000301a			a[i] = j;
36				}
37	00003036			sort(a);
38	0000303e			min = a[0];
39	00003042			max = a[9];
40	00003046			min = 0;
41	0000304a			max = 0;
42	0000304e			change(a);
43	00003056			min = a[9];
44	0000305a			max = a[0];
45	0000305e			while (1);
46				}
47				
48	0000306c		_sort	void sort(long *a)

Figure 3.47 [Source] Window at Execution Halt (Sequential Break)

The user can see the cause of the last break in the [System Status] window. The contents of the [System Status] window are as follows:

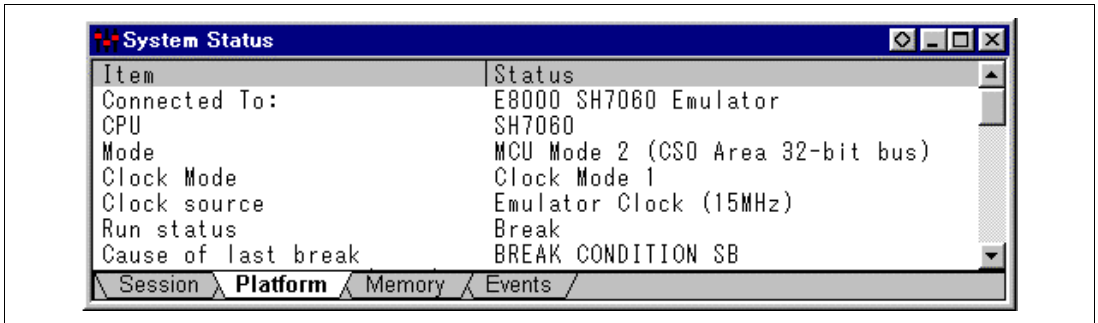


Figure 3.48 Contents of [System Status] Window (Sequential Break)

3.14 Displaying and Searching for Trace Information

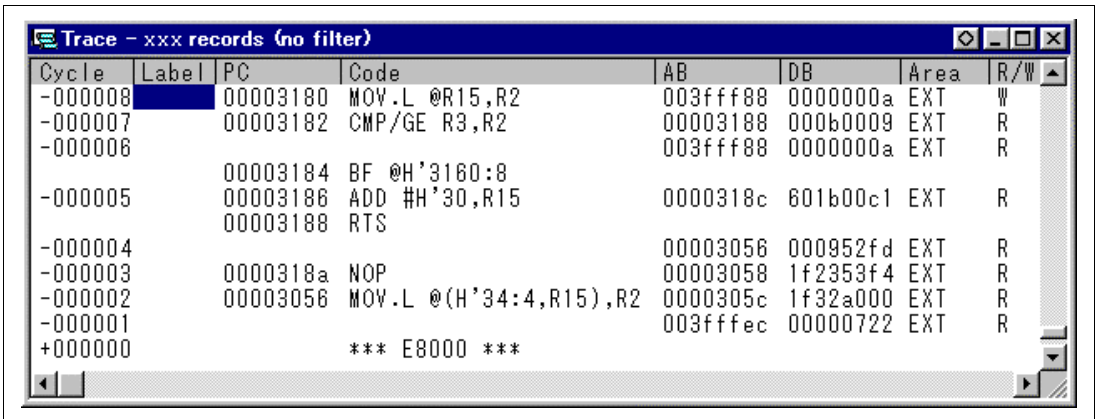
3.14.1 Displaying Trace Information

Trace information can be displayed in the [Trace] window.

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

The [Trace] window will be displayed.

- If necessary, adjust the column width by dragging the column divider beside the label immediately below the title bar.



The screenshot shows a window titled "Trace - xxx records (no filter)". The window contains a table with the following columns: Cycle, Label, PC, Code, AB, DB, Area, and R/W. The data is as follows:

Cycle	Label	PC	Code	AB	DB	Area	R/W
-000008		00003180	MOV.L @R15,R2	003fff88	0000000a	EXT	W
-000007		00003182	CMP/GE R3,R2	00003188	000b0009	EXT	R
-000006				003fff88	0000000a	EXT	R
		00003184	BF @H'3160:8				
-000005		00003186	ADD #H'30,R15	0000318c	601b00c1	EXT	R
		00003188	RTS				
-000004				00003056	000952fd	EXT	R
-000003		0000318a	NOP	00003058	1f2353f4	EXT	R
-000002		00003056	MOV.L @(H'34:4,R15),R2	0000305c	1f32a000	EXT	R
-000001				003fffec	00000722	EXT	R
+000000			*** E8000 ***				

Figure 3.49 [Trace] Window (Free Trace Results)

3.14.2 Searching for Trace Information

By setting specific search conditions, it is possible to display only the trace information that matches the search conditions in the [Trace] window.

Table 3.9 Main Trace Search Conditions

Break Condition	Description
Address bus condition (Address)	Searches for an item that matches the CPU address bus value.
Data bus condition (Data)	Searches for an item that matches the CPU data bus value. Access data size (byte, word, or longword) can be specified.
Bus state condition (Bus & Area)	There are three bus state condition settings: Read/write condition: Searches for an item for which the RD or RDWR signal level of the CPU matches the specified conditions. Bus state condition: Searches for an item for which the conditions of CPU memory access cycle or vector fetch cycle match the specified conditions. Area condition: Searches for an item for which the memory space accessed in an CPU bus cycle matches the specified conditions.
External probe signal condition (Probe)	Searches for an item for which an external probe signal (PRB1 to PRB4) level matches the specified conditions.
Interrupt signal condition (Interrupt)	Searches for an item for which the level of the NMI signal or the RESET signal matches the specified conditions.
Time condition (Time)	Searches for an item for which the time stamp value or range matches the specified conditions.

- Click the mouse right button in the [Trace] window.

The popup menu is displayed.

- Click the [Filter...] button.

The [Trace Filter] dialog box then appears.

The filter conditions that limit the cycles to be displayed in the [Trace] window can be set.

- Select the [Pattern] radio button in [Type].

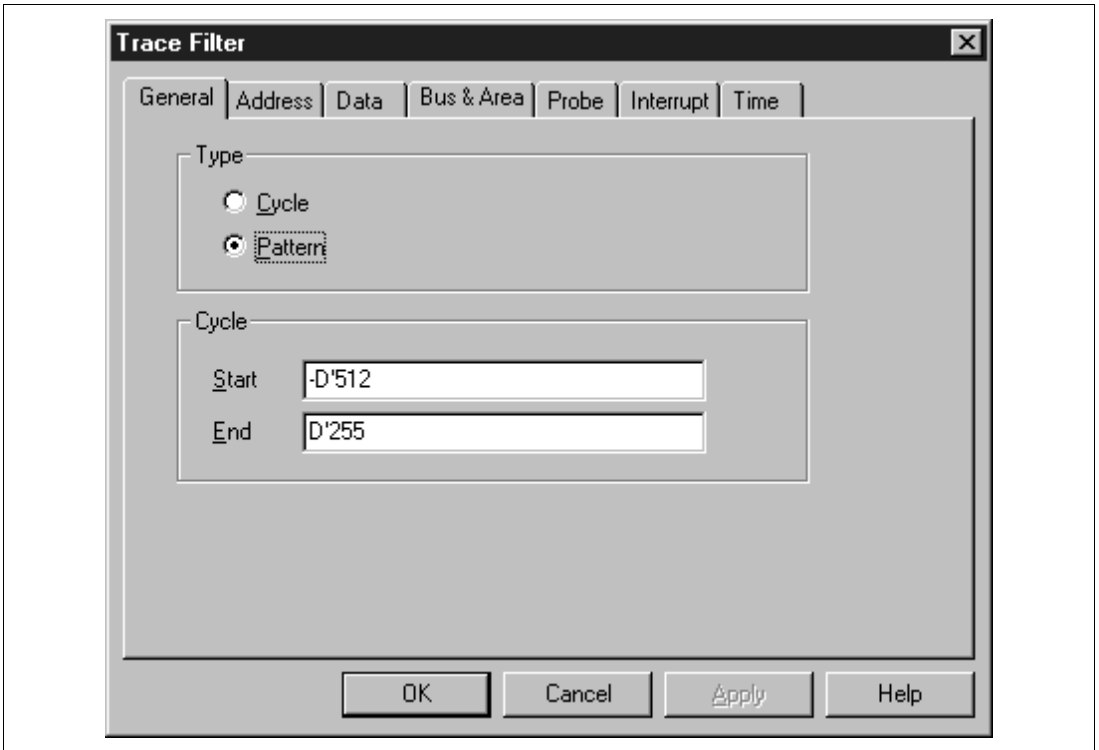


Figure 3.50 [General] Page ([Trace Filter] Dialog Box)

- Select [Address] to display the [Address] page.

- Clear the [Don't Care] check box in the [Address] page.
- Select [Address] and input address *H'3048* as the value in the [Start] edit box.

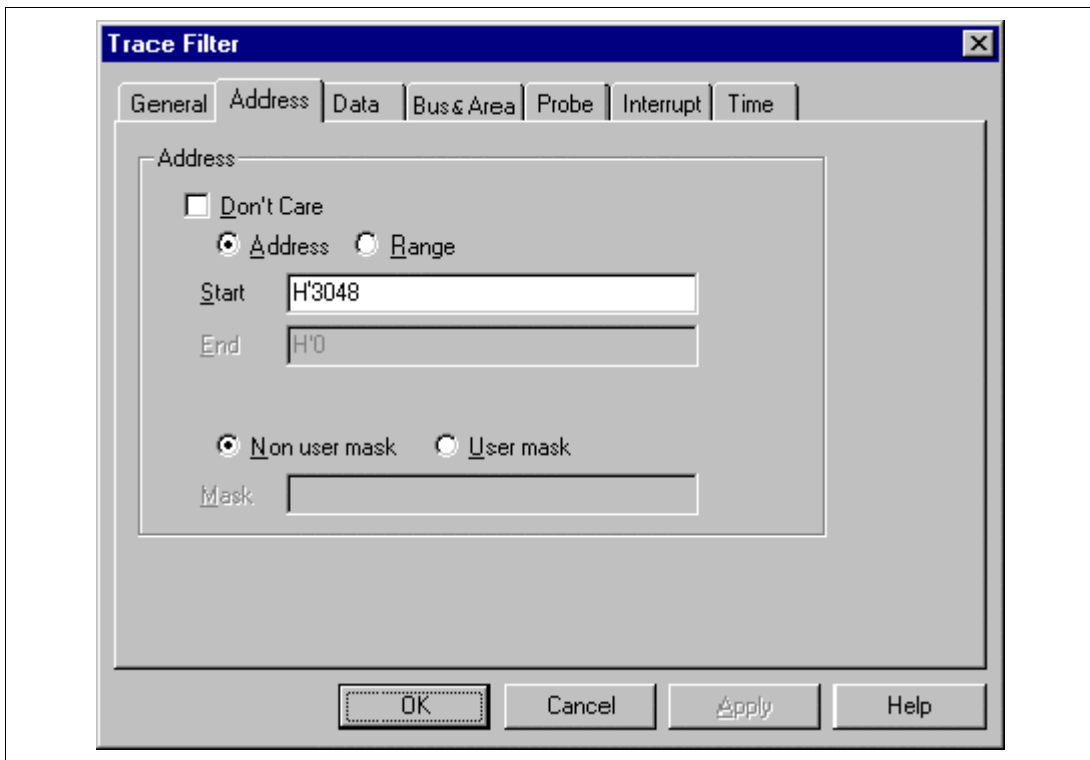


Figure 3.51 [Address] Page ([Trace Filter] Dialog Box)

- Select [Bus & Area] to display the [Bus & Area] page.

- Select [Read] radio button.

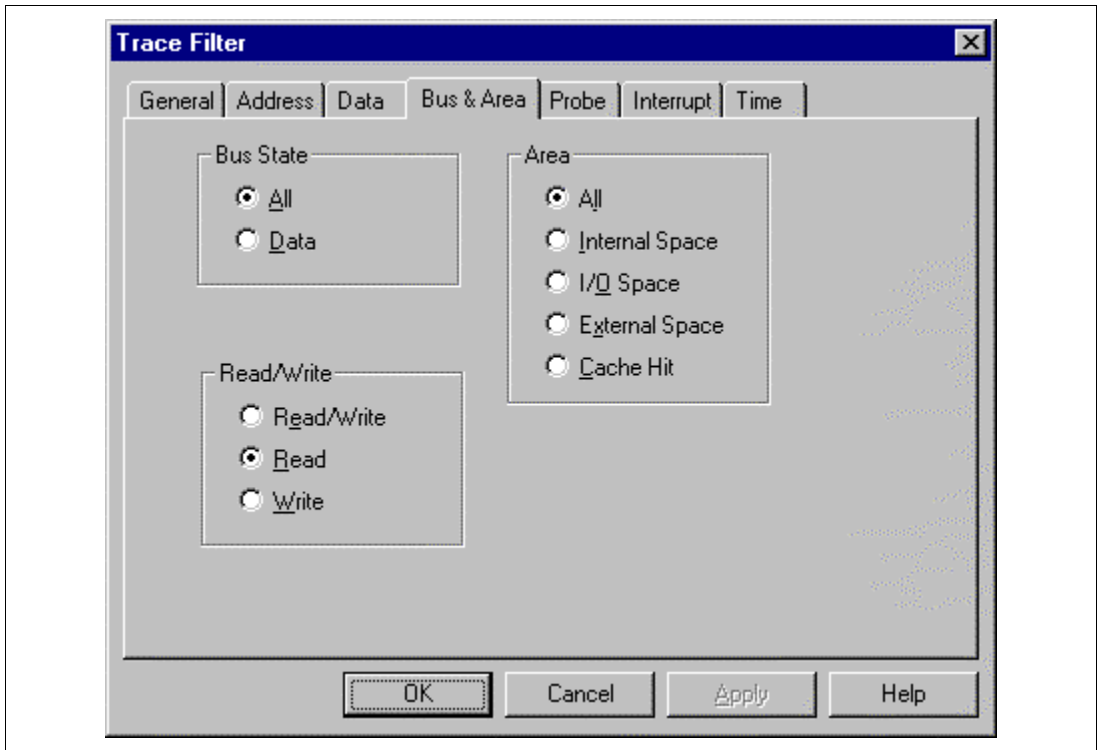
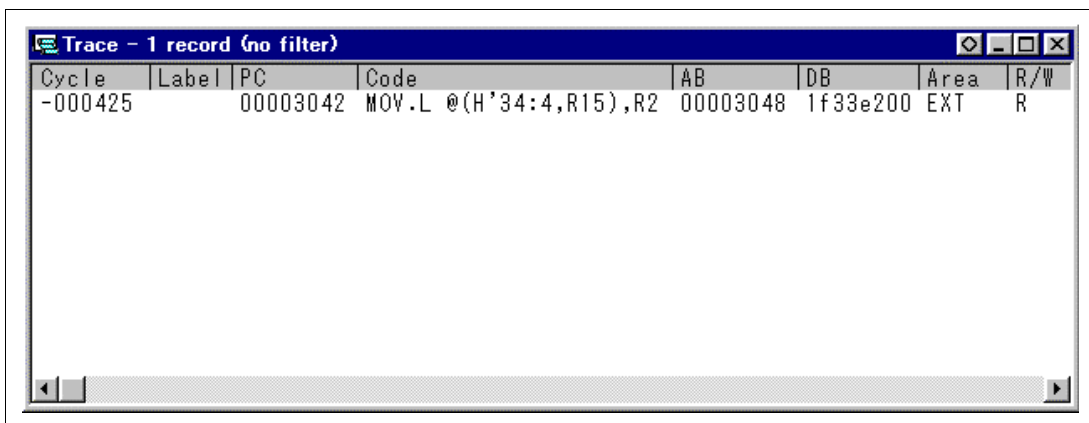


Figure 3.52 [Bus & Area] Page ([Trace Filter] Dialog Box)

- Click the [OK] button.

If trace information that matches the trace-information search conditions is found, the contents are displayed in the [Trace] window.



The screenshot shows a window titled "Trace - 1 record (no filter)". The window contains a table with the following data:

Cycle	Label	PC	Code	AB	DB	Area	R/W
-000425		00003042	MOV.L @(H'34:4,R15),R2	00003048	1f33e200	EXT	R

Figure 3.53 [Trace] Window (Trace Filter Results)

3.15 Setting the Trace Acquisition Conditions

The emulator has powerful realtime trace functions. Trace information for up to 131,070 bus cycles can be acquired. In the HDI, trace acquisition conditions can be set by using dialog boxes. The dialog boxes for setting trace acquisition conditions and the corresponding trace functions are described below.

Table 3.10 Dialog Boxes for Setting Trace Acquisition Conditions

Dialog Box	Function			
	Subroutine Trace	Range Trace	Trace Stop	Subroutine Range Trace
[Trace Condition A] dialog box	X	O	O	X
[Trace Condition B] dialog box	O	O	O	O
[Trace Condition C] dialog box	O	O	O	X

Note: O: Can be set in the dialog box.

X: Cannot be set in the dialog box.

Table 3.11 Main Trace Functions

Trace Functions	Description
Free trace	Acquires trace information continuously from the start of execution of the user program until the program breaks. If Trace Condition A,B,C is not set, this mode is entered.
Subroutine trace	Performs trace acquisition of instructions or operand accesses between the start address and the end address of the subroutine specified by Trace Condition B,C. If the specified subroutine calls other subroutines, trace acquisition is not performed for the called subroutines.
Range trace	Performs trace acquisition only for places where the conditions specified by Trace Condition A,B,C are satisfied. Specifiable conditions are: Address bus condition Data bus condition Read/write condition Bus state condition (execution cycle) System control signal (BREQ) External probe condition
Trace stop	Stops trace acquisition when the conditions specified by Trace Condition A,B,C are satisfied. Specifiable conditions are: Address bus condition Data bus condition Read/write condition Bus state condition (execution cycle) System control signal (BREQ) External probe condition Satisfaction count DELAY condition
Subroutine range trace	Performs trace acquisition only for places that have been accessed within a subroutine specified by Trace Condition B, and when the specified conditions have been satisfied at those places (specifiable conditions are the same as those for range trace).

An example is given below in which trace stop mode (in which address bus condition and read cycles for bus state condition are set) is selected for Trace Condition A as the trace function.

- Click the mouse right button in the [Trace] window.

The popup menu is displayed.

- Click the [Acquisition] button to display the [Trace Acquisition] dialog box.

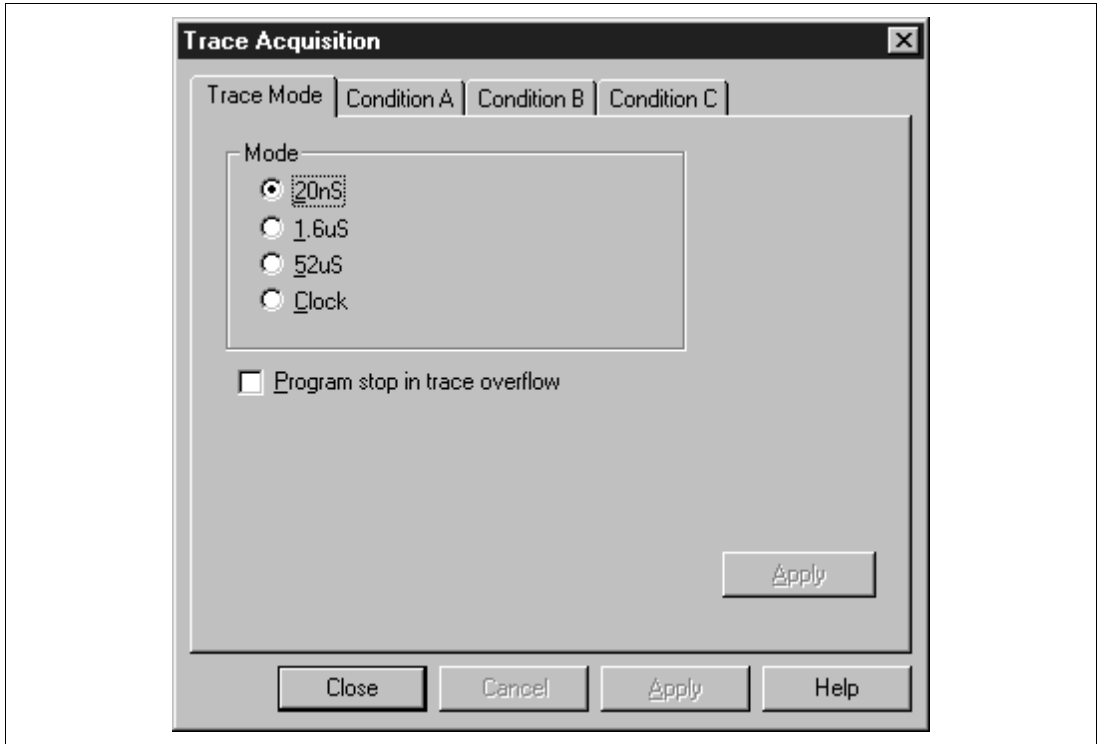


Figure 3.54 [Trace Acquisition] Dialog Box

For trace acquisition conditions, the [Trace Acquisition] dialog box pages required for the setting must be selected.

- Select [Condition A] to display the [Condition A] page.

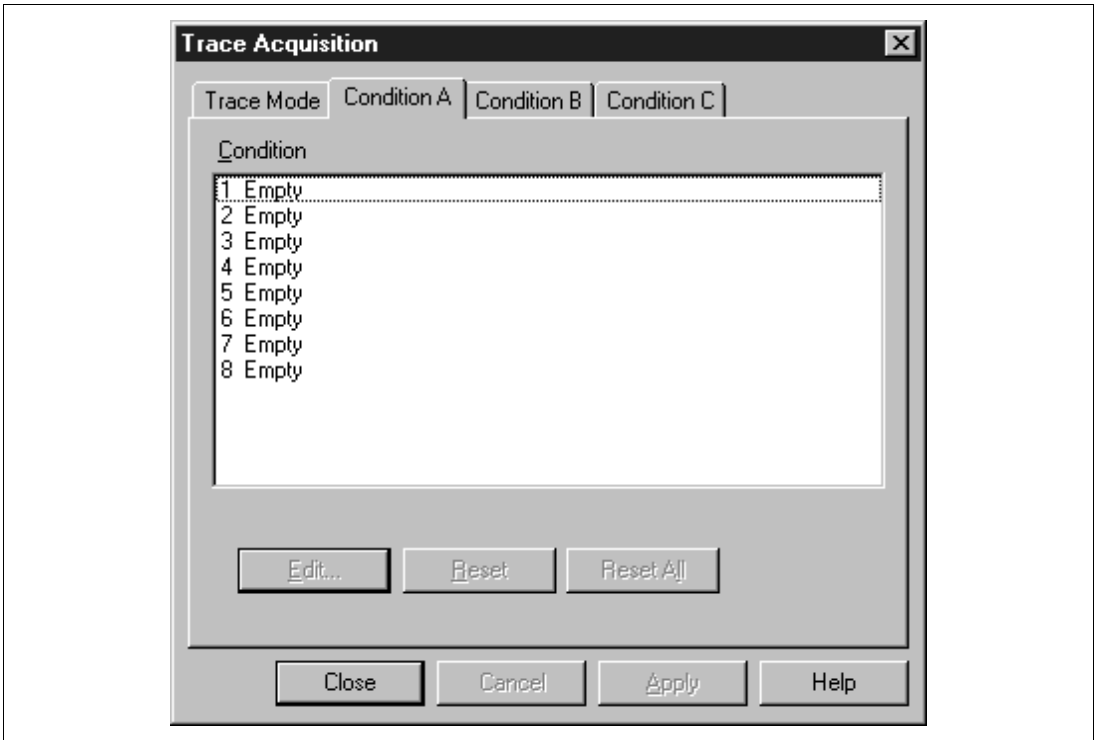


Figure 3.55 [Condition A] Page ([Trace Acquisition] Dialog Box)

- Highlight the first point in the [Condition] list box by clicking.
- Click the [Edit...] button.

The [Trace Condition A1] dialog box is displayed.

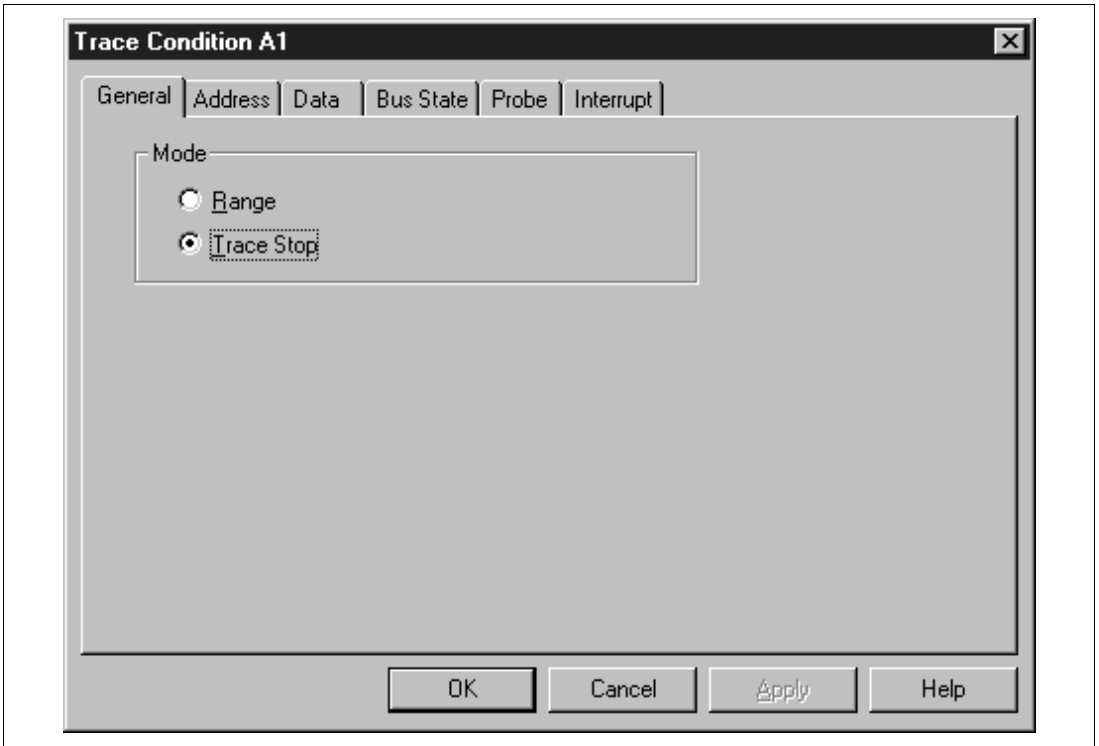


Figure 3.56 [General] Page ([Trace Condition A1] Dialog Box)

- Select [Trace Stop] radio button as [Mode] in the [General] page.
- Select [Address] to display the [Address] page.

- Clear the [Don't Care] check box in the [Address] page.
- Select [Address] and input **H'3044** as the value in the [Start] edit box.

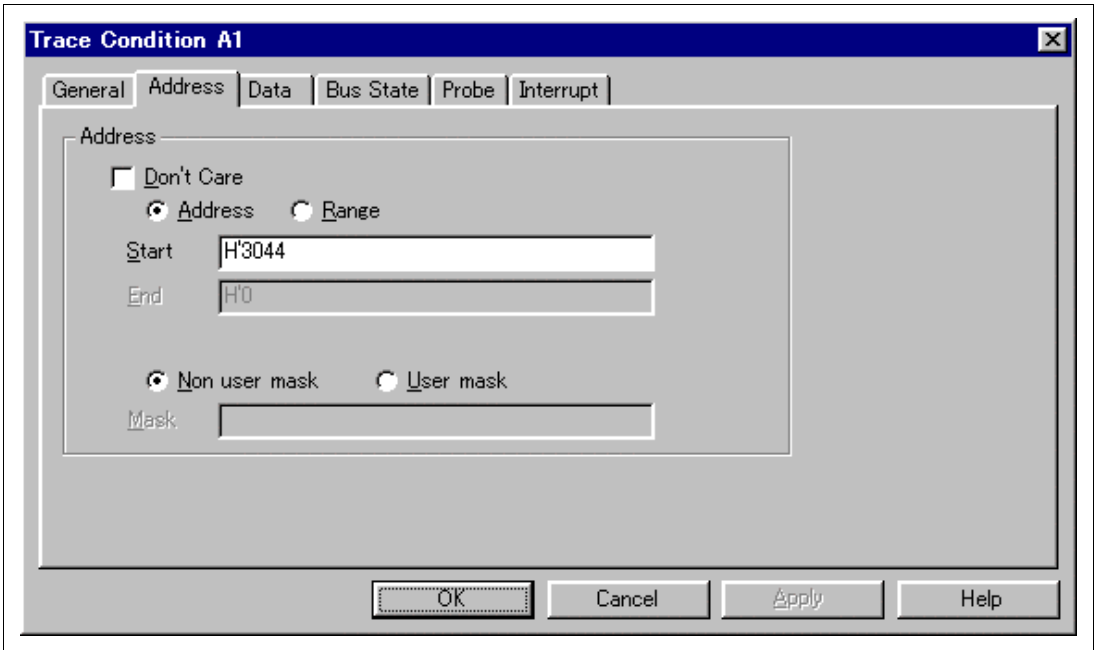


Figure 3.57 [Address] Page ([Trace Condition A1] Dialog Box)

- Select [Bus State] to display the [Bus State] page.

- Select [Read] radio button.

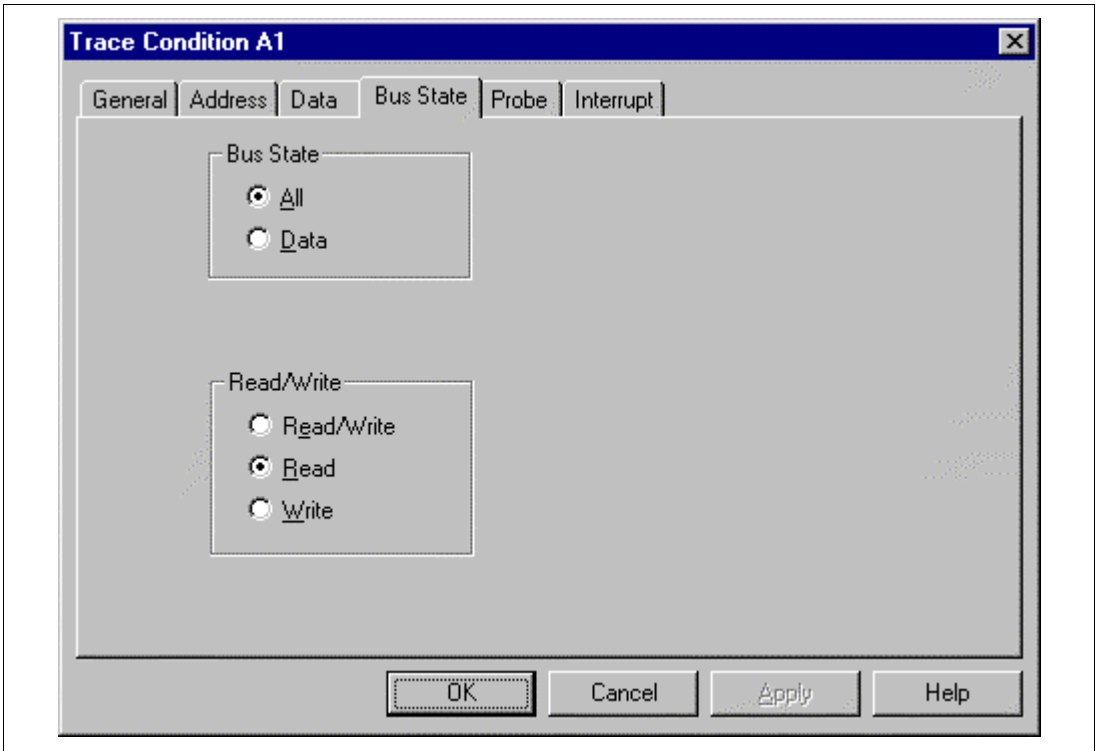


Figure 3.58 [Bus State] Page ([Trace Condition A1] Dialog Box)

- Click the [OK] button.

The [Trace Acquisition] dialog box is displayed, and the first point display in the [Condition] list box changes from “Empty” to “Enable”.

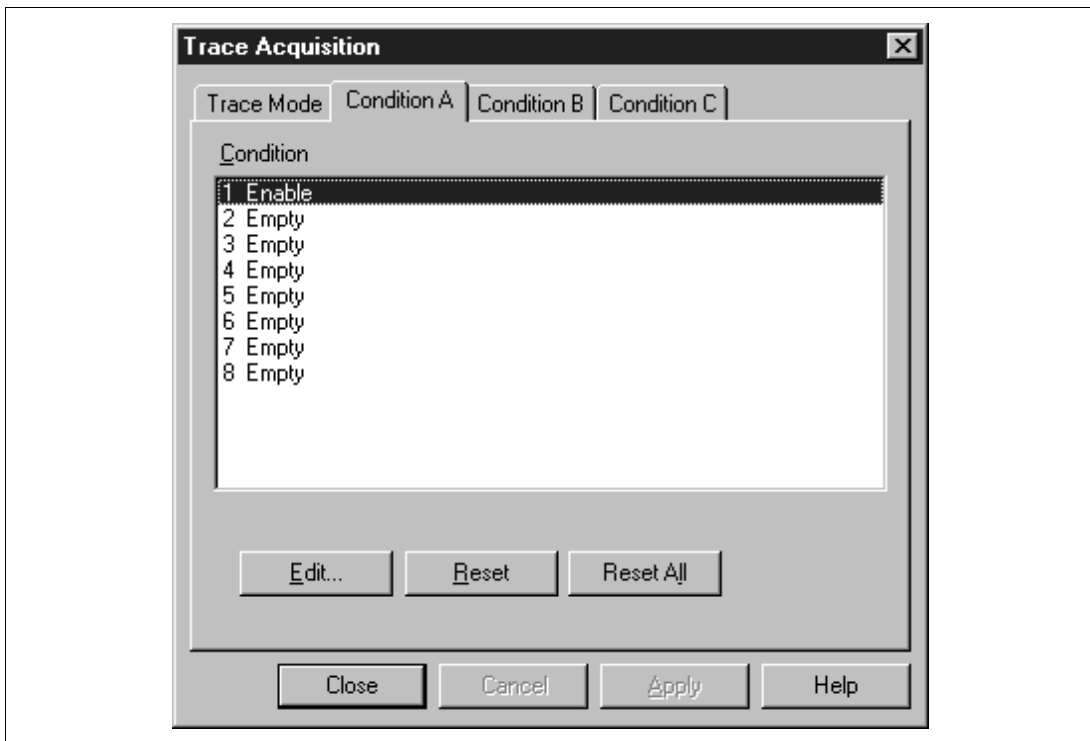


Figure 3.59 [Condition A] Page ([Trace Acquisition] Dialog Box)

This completes the setting of the Trace Condition A1 trace acquisition conditions. When the program is executed, trace acquisition will stop when address H'3044 is accessed in a read cycle.

3.16 Measuring the Subroutine Execution Time and the Execution Count

The E8000 emulator has a function for measuring the ratio of the subroutine execution time to the total program execution time and a subroutine execution count.

In the HDI, the conditions of the subroutine measurement time and execution count can be set in a dialog box, and the measurement results can be displayed in the window.

The dialog boxes and the subroutine measurement modes are described in the following:

Table 3.12 Dialog Boxes for Setting Subroutine Execution Time and Execution Count

Dialog Box	Subroutine measurement mode					Remarks
	(1) Subroutine Time Measure- ment Mode 1	(2) Subroutine Time Measure- ment Mode 2	(3) Subroutine Time Measure- ment Mode 3	(4) Access Area Count Measurement Mode	(5) Subroutine Call Count Measure- ment Mode	
[Performance Analysis 1] dialog box (PA1)	O	O	O	O	O	If PA2 is set, (3), (4), and (5) cannot be set by PA1.
[Performance Analysis 2] dialog box (PA2)	O	O	X	X	X	
[Performance Analysis 3] dialog box (PA3)	O	O	O	O	O	If PA4 is set, (3), (4), and (5) cannot be set by PA3.
[Performance Analysis 4] dialog box (PA4)	O	O	X	X	X	
[Performance Analysis 5] dialog box (PA5)	O	O	O	O	O	If PA6 is set, (3), (4), and (5) cannot be set by PA5.
[Performance Analysis 6] dialog box (PA6)	O	O	X	X	X	
[Performance Analysis 7] dialog box (PA7)	O	O	O	O	O	If PA8 is set, (3), (4), and (5) cannot be set by PA7.
[Performance Analysis 8] dialog box (PA8)	O	O	X	X	X	

Note: O: Can be set in the dialog box.

X: Cannot be set in the dialog box.

Table 3.13 Subroutine Measurement Mode

Subroutine Measurement Mode	Description
Subroutine time measurement mode 1	<p>Measures the execution time and count of the subroutine defined by the address condition according to the setting of Performance Analysis 1 to 8. Measurement starts when an address within the specified range is prefetched, halts when an address outside the specified range is prefetched, and restarts when an address within the specified range is prefetched again. The subroutine execution count is incremented every time the subroutine end address is fetched. The execution time of subroutines called from the specified subroutine is not included in the measurement results.</p> <p>For Performance Analysis 1 only, a timeout value and a limit count value can be specified in addition to the address condition.</p>
Subroutine time measurement mode 2	<p>Measures the execution time and count of the subroutine defined by the address condition according to the setting of Performance Analysis 1 to 8. Measurement starts when the start address is prefetched and halts when the end address is prefetched. The subroutine execution count is incremented every time the subroutine end address is fetched. The execution time of subroutines called from the specified subroutine is included in the measurement results.</p> <p>For Performance Analysis 1 only, a timeout value and a limit count value can be specified in addition to the address condition.</p>
Subroutine time measurement mode 3	<p>Starts measurement when an address in the start address range is prefetched and halts when an address in the end address range is prefetched, according to the setting of Performance Analysis 1/3/5/7. The subroutine execution count is incremented every time <end address range> is passed.</p>
Access area count measurement mode	<p>Counts the number of times the subroutine defined by <start address> and <end address> accesses the range specified by <accessed area address range>, according to the setting of Performance Analysis 1/3/5/7. The subroutine execution time is measured using subroutine execution time measurement mode 1.</p>
Subroutine call count measurement mode	<p>Counts the number of times the subroutine defined by <subroutine name>, <start address>, and <end address> calls the subroutine specified by <called subroutine address range> according to the setting of Performance Analysis 1/3/5/7. The subroutine execution time is measured using subroutine execution time measurement mode 1.</p>

An example is described where subroutine time measurement mode 1 is set to Performance Analysis 1 and Performance Analysis 2.

Set the conditions as follows to compare the execution time and execution count between the `sort` function and the `change` function in the `main` function:

Performance Analysis 1: Measures the execution time and execution count for the `sort` function

Performance Analysis 2: Measures the execution time and execution count for the `change` function

Before executing the program, cancel the conditions that have been set for the sequential break function according to the following steps:

- Change the setting of the [Emulation mode] combo box in the [Configuration] dialog box to [Normal]. (Refer to section 4.1, Setting the [Configuration] Dialog Box.)
- Select the [Breakpoints] from the [View] menu.

The [Breakpoints] window then appears.

- Click the mouse right button in the [Breakpoints] window.

The popup menu is displayed.

- Cancel all the breakpoints that have been set by selecting [Delete All].

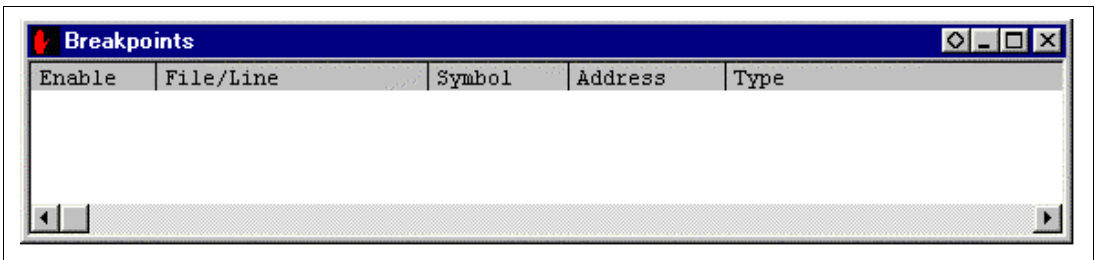


Figure 3.60 [Breakpoints] Window (with Break Conditions Canceled)

3.16.1 Displaying the Measurement Results of Subroutine Execution Time and Execution Count

The [Performance Analysis] window displays the measurement results of the subroutine execution time and execution count. To display the measurement results, set conditions for the subroutine execution time and execution count using the corresponding dialog box and execute a program using the Reset Go command.

- Select the [Performance Analysis Window] from the [View] menu.

The [Performance Analysis] window then appears.



Figure 3.61 [Performance Analysis] Window (Initial State)

3.16.2 Setting the Conditions of Subroutine Execution Time and Execution Count

The conditions of subroutine execution time and execution count can be set using the dialog box selected from the pop-up menu in the [Performance Analysis] window.

- Click the [Performance Analysis] window with the right button of the mouse and select [Edit...] from the pop-up menu.

The [Performance Analysis] dialog box appears.

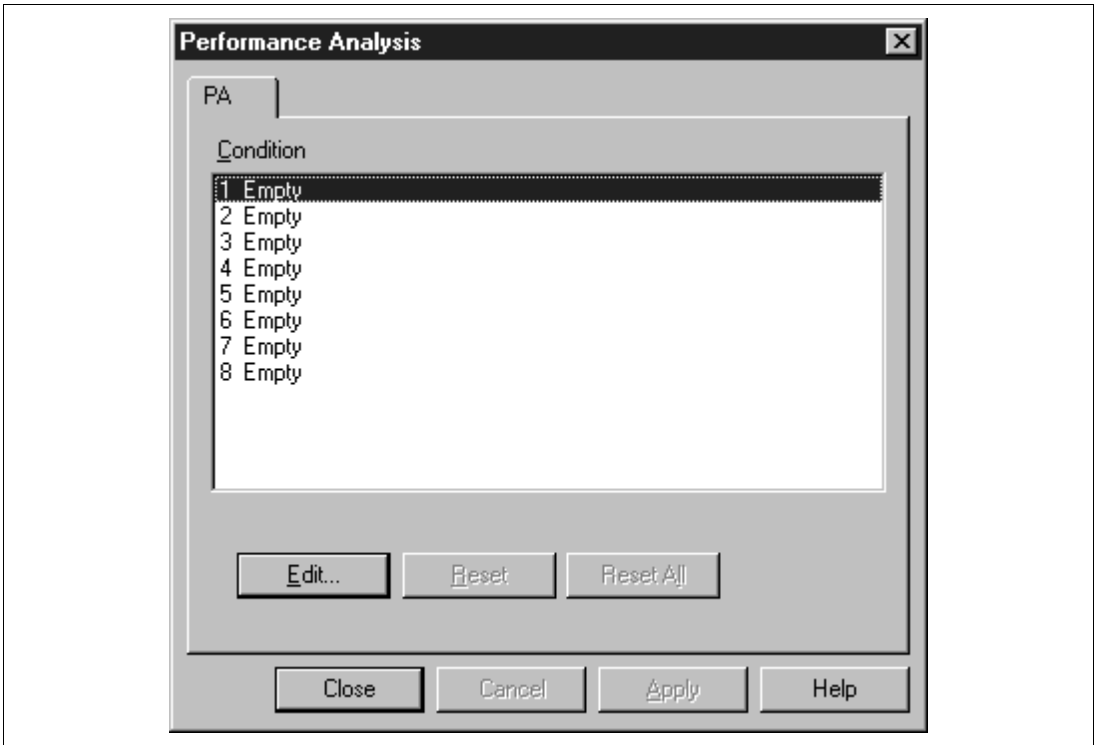


Figure 3.62 [PA] Page ([Performance Analysis] Dialog Box)

- Highlight the first point of the [Condition] list box by clicking.
- Click the [Edit...] button.

The [Performance 1] dialog box is then displayed.

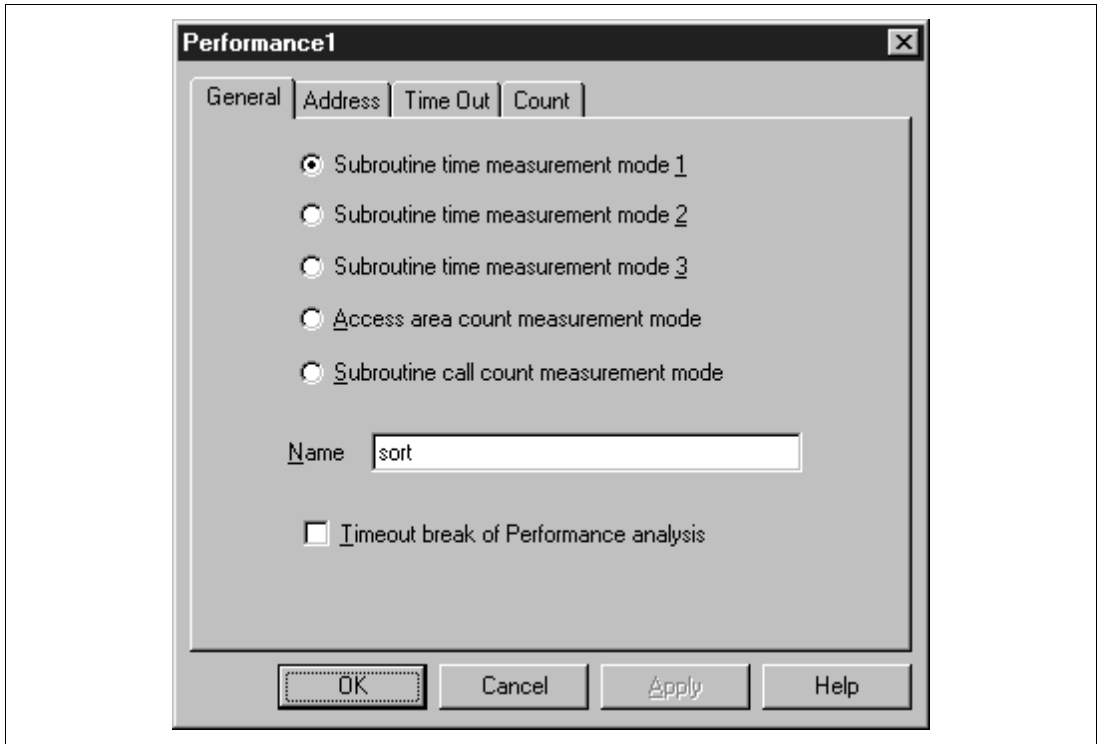


Figure 3.63 [General] Page ([Performance 1] Dialog Box)

- Select the [Subroutine time measurement mode 1] radio button in the [General] page and enter the function name *sort* in the [Name] edit box.
- Select [Address] to display the [Address] page.

- Click the [...] button in the [Subroutine Address].

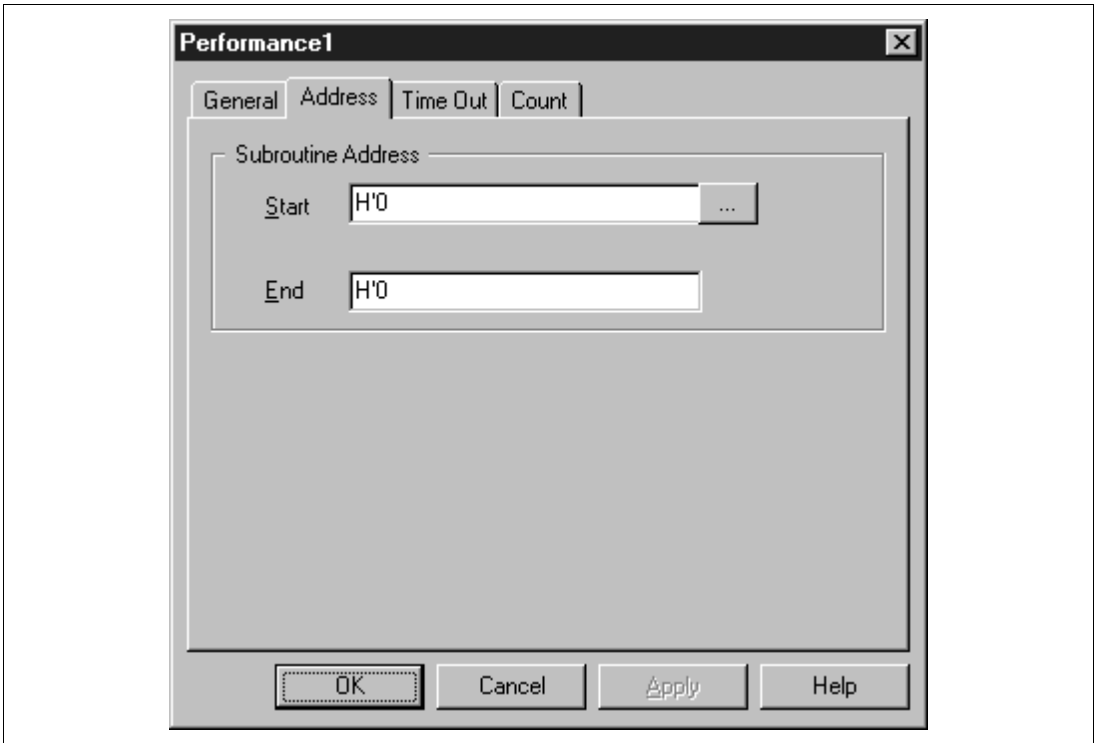


Figure 3.64 [Address] Page ([Performance 1] Dialog Box)

The [Input Function Range] dialog box is displayed.

- Enter the function name *sort* in the [Subroutine Address] edit box.
- Click the [OK] button.

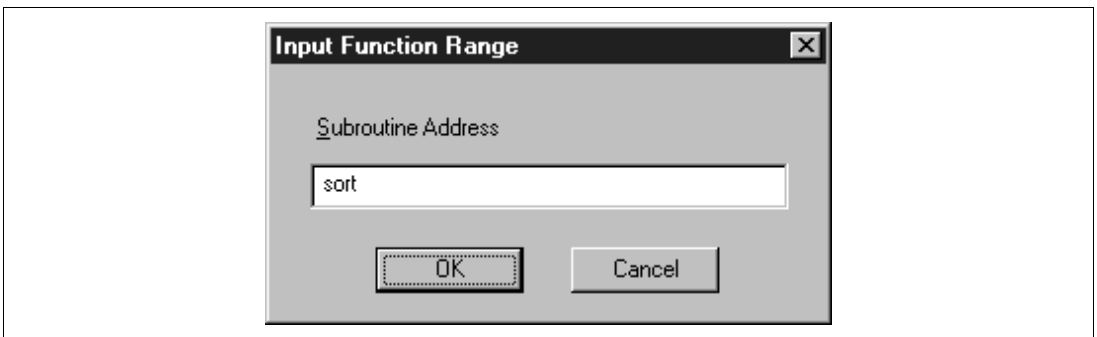


Figure 3.65 [Input Function Range] Dialog Box

Address values are then displayed in the [Start] and [End] edit boxes in the [Address] page.

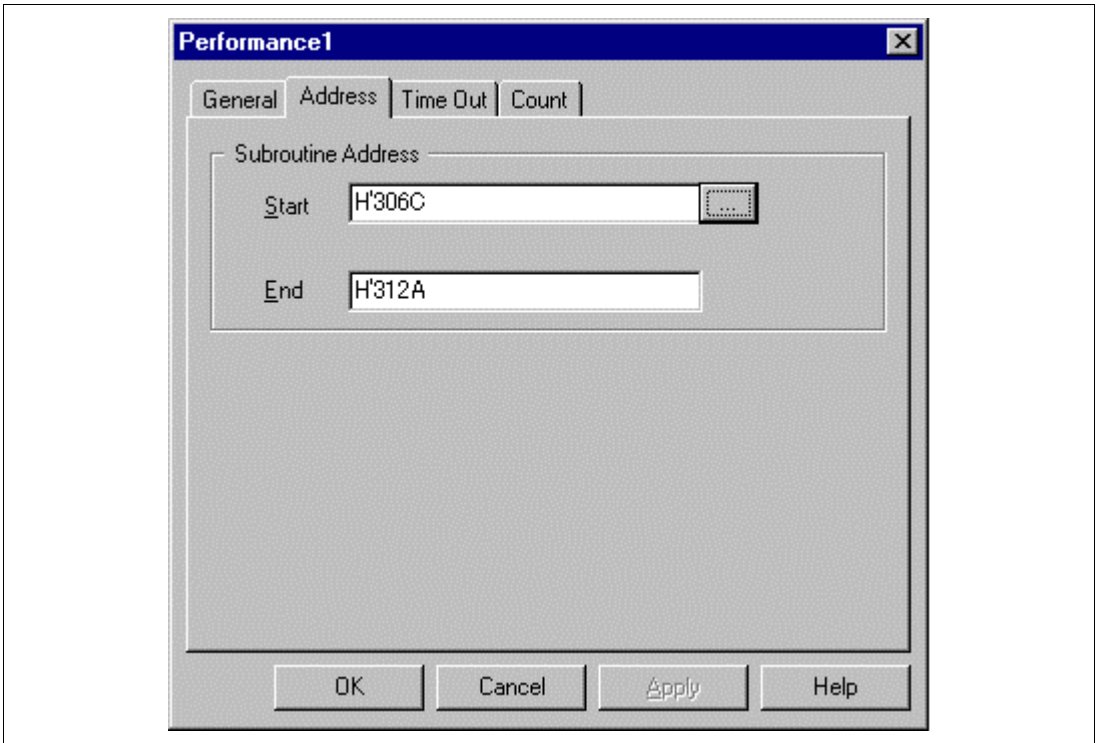


Figure 3.66 [Address] Page (after the Address of the sort Function is Specified)

- Click the [OK] button.

The [Performance Analysis] dialog box is then displayed. The indication of the first point in the [Condition] list box changes from "Empty" to "Enable sort."

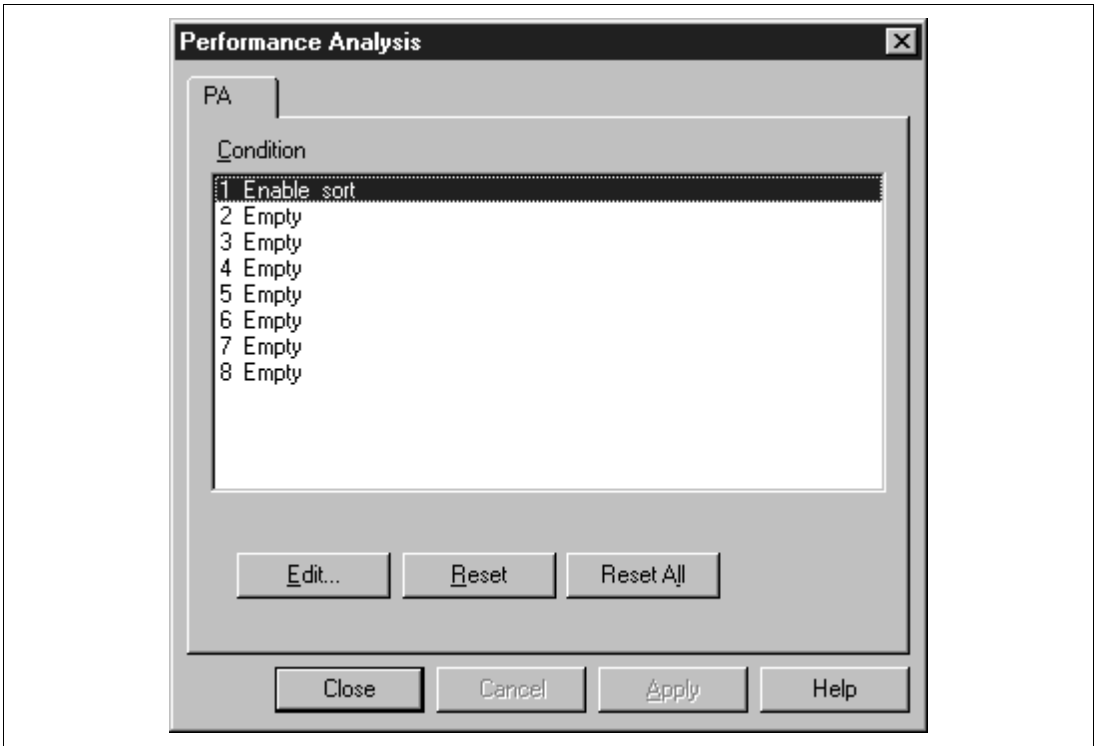


Figure 3.67 [PA] Page ([Performance Analysis] Dialog Box)

This completes the setting for the subroutine call count measurement mode in Performance Analysis 1. Next, Performance Analysis 2 is set.

- Highlight the second point of the [Condition] list box by clicking.
- Click the [Edit...] button.

The [Performance Analysis 2] dialog box is then displayed. Set Performance Analysis 2 condition in the same way as that for Performance Analysis 1.

- Click the [Close] button when the setting for Performance Analysis 1,2 is completed.

3.16.3 Measuring the Subroutine Execution Time and Execution Count

The subroutine execution time and execution count are compared between the `sort` and `change` functions after the program is executed from the first step to the last step of the `main` function.

- Set a software breakpoint in the [Source] window by double clicking the [Break] column at address H'305e.

The format for the measurement results displayed in the [Performance Analysis] window is selected to compare the subroutine execution time and execution count.

- Click the [Performance Analysis] window with the right button of the mouse to display the pop-up menu.

The following display formats can be selected.

Table 3.14 Display Formats for Measurement Results

Measurement Result Display Format	Description
Address	Displays the list of subroutine addresses that have been set (default).
Count	Displays with numerical values the execution time and execution count of a subroutine that has been set.
Graph	Displays with a graph the ratio of the subroutine execution time to the total execution time for the specified subroutine.

- Select [Graph] to display the execution performance.

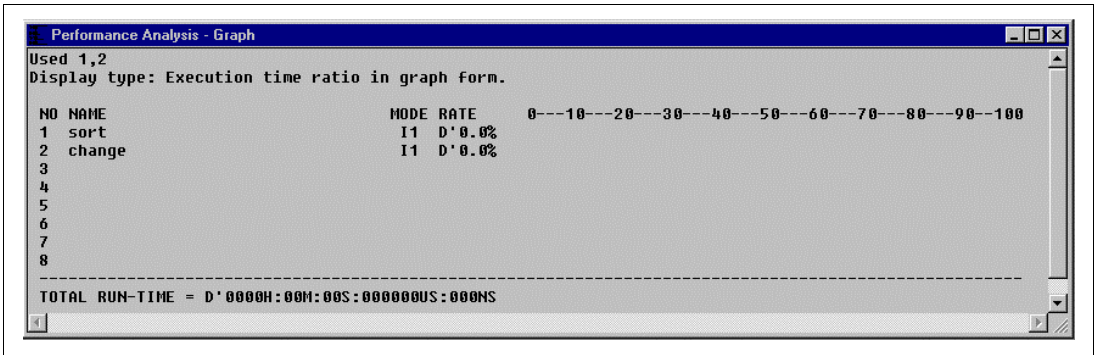


Figure 3.68 [Performance Analysis] Window (before Program Execution)

The program is executed until address H'305e, then it stops. In this case, the measurement results concerning the execution performance are displayed in the [Performance Analysis] window.

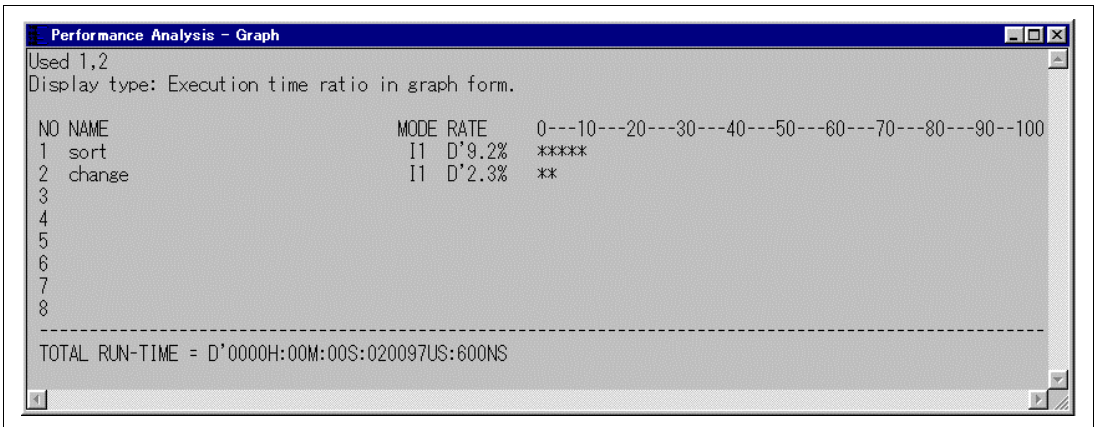


Figure 3.69 [Performance Analysis] Window (after Program Execution)

When [Count] is selected in the pop-up menu, the measurement result of the execution count is displayed.

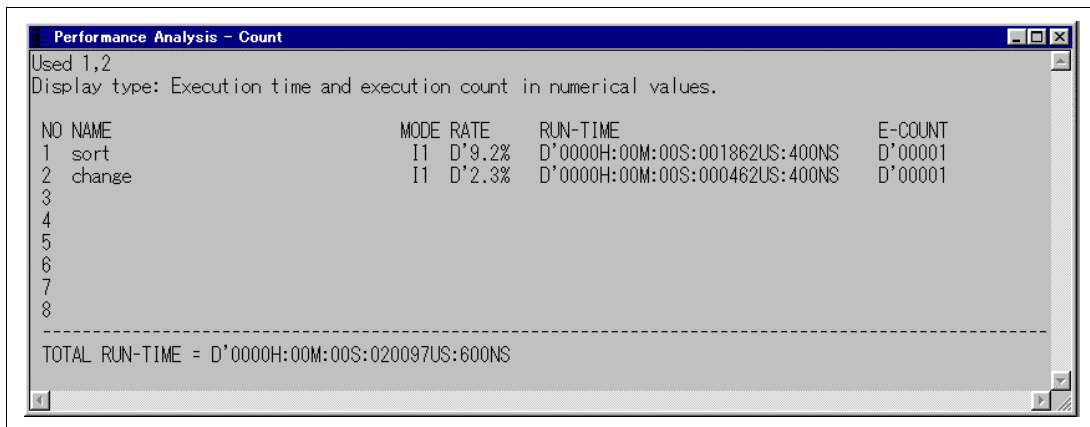


Figure 3.70 [Performance Analysis] Window (Execution Count)

3.17 Saving and Loading the Session

The settings for the HDI window and dialog box at a given point can be saved as a session file. By loading this session file when the HDI is started, debugging can be resumed from the same point as last time.

Select [Save Session As...] from the [File] menu to save a session file. The [Save Session As...] dialog box is displayed. Input a session file name and click the [Save] button. Select [Load Session...] from the [File] menu to load a session file.

Automatic saving and loading of a session file can be set in the [HDI Options] dialog box (select [Options...] in the [Setup] menu).

Select the [Save session automatically] radio button on the [Session] page to specify the automatic saving of a session file. The file to save at the end of HDI sessions is displayed in the dialog box. When a file name is specified, session information is automatically saved in this file after each HDI session is ended.

Make the [Load last session on startup] check box on the [Session] page valid for automatic loading of session files. Session information is loaded automatically from the specified file when the HDI session is ended.

Refer to the Hitachi Debugging Interface User's Manual, issued separately, for a detailed description of session settings, and of the way to set them.

3.18 What Next?

This tutorial has described the major features of the E8000 emulator and the use of the HDI.

Advanced debugging can be performed by combining the powerful emulation function of the E8000 emulator with the HDI. This provides for an effective investigation of hardware and software problems by accurately isolating the conditions under which such problems arise.

For details on the use of the HDI, refer to the Hitachi Debugging Interface User's Manual, issued separately.

Section 4 Function

4.1 Windows and Dialog Boxes

Table 4.1 shows the SH7060 E8000 HDI-specific windows and dialog boxes. For more information, see the online-help. To see the online-help, select [Index] in the [Help] menu, or click [Help] button in each dialog box. Also the related commands in the SH7060 E8000 Emulator User's Manual are described in the online-help.

Table 4.1 List of Windows and Dialog Boxes

Category	Name of Window and Dialog box	Description	Related E8000 commands
Emulation condition	[System Status] window	Displays the set-up conditions and the program execution time of the emulation-associated functions in the emulator.	CLOCK, EXECUTION_MODE, GO, MODE
	[Configuration] dialog box	Sets the emulation conditions in the emulator.	CLOCK, EXECUTION_MODE, GO, MODE
Break condition	[Breakpoints] window	Lists the all break conditions having been set up.	BREAK, BREAK_SEQUENCE, BREAK_CONDITION_A, BREAK_CONDITION_B, BREAK_CONDITION_C, BREAK_CONDITION_UBC
	[Break] dialog box	Displays the set-up state of each break condition.	BREAK, BREAK_SEQUENCE, BREAK_CONDITION_A, BREAK_CONDITION_B, BREAK_CONDITION_C, BREAK_CONDITION_UBC
	[Breakpoint] dialog box	Sets up to 255 software breakpoints.	BREAK

Table 4.1 List of Windows and Dialog Boxes (cont)

Category	Name of Window and Dialog box	Description	Related E8000 commands
Break condition	[Break Sequence] dialog box	Sets the software sequential break with up to seven passing points and one reset point.	BREAK_SEQUENCE
	[Break Condition A] dialog box [Break Condition B] dialog box [Break Condition C] dialog box	Each sets up to eight hardware break conditions.	BREAK_CONDITION_A, BREAK_CONDITION_B, BREAK_CONDITION_C
	[Break Condition UBC] dialog box	Sets up to four UBC hardware break conditions.	BREAK_CONDITION_UBC
	[Break Condition UBC Reset] dialog box	Sets reset points for UBC sequential break conditions.	BREAK_CONDITION_UBC
	Trace condition	[Trace] window	Displays the contents of the trace buffer.
[Trace Acquisition] dialog box		Sets trace acquisition conditions and displays the setting state.	TRACE_CONDITION_A, TRACE_CONDITION_B, TRACE_CONDITION_C
[Trace Condition A] dialog box [Trace Condition B] dialog box [Trace Condition C] dialog box		Each sets up to eight acquisition conditions of the trace information.	TRACE_CONDITION_A, TRACE_CONDITION_B, TRACE_CONDITION_C
[Trace Filter] dialog box		Only displays the trace result that matches the conditions having been set up, from among the results displayed in the [Trace] window.	TRACE_SEARCH
[Trace Find] dialog box		Searches the trace result which matches the conditions having been set up, from among the trace results displayed in the [Trace] window.	TRACE_SEARCH

Table 4.1 List of Windows and Dialog Boxes (cont)

Category	Name of Window and Dialog box	Description	Related E8000 commands
Memory map	[Memory Mapping] window	Displays and edits the information about the emulation memory allocation.	MAP
	[Edit Memory Mapping] dialog box [Add Memory Mapping] dialog box	Sets and modifies the information about the emulation memory allocation.	MAP
Performance condition	[Performance Analysis] window	Displays the program execution state.	PERFORMANCE_ANALYSIS 1-8
	[Performance Analysis] dialog box	Sets eight performance conditions.	PERFORMANCE_ANALYSIS 1-8
	[Input Function Range] dialog box	Sets the address range of the function.	None

4.2 Command Descriptions

The SH7060 E8000 HDI-specific commands are shown in table 4.2. For more information, see the online-help. To see the on-line help, select [Index] in the [Help] menu, or execute the HELP command from the [Command Line] window.

Example

To open the on-line help related to the BREAKPOINT command:

```
help BREAKPOINT (RET)
```

Table 4.2 HDI Commands

No	Command	Abb.	Function	Related E8000 Commands
1	BREAKCONDITION_CLEAR	BCC	Clears hardware breakpoints (break conditions) that have been set.	BREAK_CONDITION_A,B,C
2	BREAKCONDITION_DISPLAY	BCD	Displays hardware breakpoints (break conditions) that have been set.	BREAK_CONDITION_A,B,C
3	BREAKCONDITION_ENABLE	BCE	Enables or disables hardware breakpoints (break conditions) that have been set.	BREAK_CONDITION_A,B,C
4	BREAKCONDITION_SET	BCS	Sets hardware breakpoints (break conditions) that have been set.	BREAK_CONDITION_A,B,C
5	BREAKSEQUENCE_CLEAR	BSC	Clears software sequential breakpoints that have been set.	BREAK_SEQUENCE
6	BREAKSEQUENCE_DISPLAY	BSD	Displays software sequential breakpoints that have been set.	BREAK_SEQUENCE
7	BREAKSEQUENCE_ENABLE	BSE	Enables or disables software sequential breakpoints that have been set.	BREAK_SEQUENCE
8	BREAKSEQUENCE_SET	BSS	Sets software sequential breakpoints.	BREAK_SEQUENCE
9	BREAKPOINT	BP	Sets software breakpoints.	BREAK
10	BREAKPOINT_CLEAR	BC	Clears software breakpoints that have been set.	BREAK
11	BREAKPOINT_DISPLAY	BD	Displays software breakpoints that have been set.	BREAK
12	BREAKPOINT_ENABLE	BE	Enables or disables software breakpoints that have been set.	BREAK
13	CLOCK	CK	Selects a CLOCK signal used by the SH7060.	CLOCK

Table 4.2 HDI Commands (cont)

No	Command	Abb.	Function	Related E8000 Commands
14	DEVICE_TYPE	DE	Displays the CPU type currently selected.	None
15	END	END	Returns to a user program execution state when the emulator enters the parallel mode due to trace condition satisfaction.	END
16	EXECUTION_MODE	EM	Sets debugging conditions during user program execution.	EXECUTION_MODE
17	GO_OPTION	GP	Sets the emulation mode during user program execution.	GO
18	ID	ID	Displays an emulator type and a version number.	ID
19	MAP_SET	MS	Sets emulator memory map.	MAP
20	MODE	MO	Selects emulator mode.	MODE
21	MOVE_TO_RAM	MR	Stores a user program on ROM into RAM.	MOVE_TO_RAM
22	PERFORMANCE_ANALYSIS	PA	Displays program execution state.	PERFORMANCE_ANALYSIS1 to 8
23	PERFORMANCE_CLEAR	PC	Clears performance conditions that have been set.	PERFORMANCE_ANALYSIS1 to 8
24	PERFORMANCE_SET	PS	Sets performance conditions.	PERFORMANCE_ANALYSIS1 to 8
25	STATUS	STS	Displays emulator state information.	None
26	TRACEACQUISITION_CLEAR	TAC	Clears trace conditions that have been set.	TRACE_CONDITION_A,B,C

Table 4.2 HDI Commands (cont)

No	Command	Abb.	Function	Related E8000 Commands
27	TRACEACQUISITION_DISPLAY	TAD	Displays trace conditions that have been set.	TRACE_CONDITION_A,B,C
28	TRACEACQUISITION_SET	TAS	Sets trace conditions that acquire trace information.	TRACE_CONDITION_A,B,C
29	TRACE_DISPLAY	TD	Display the acquisition of trace information.	TRACE
30	TRACE_MODE	TM	Sets the trace information acquisition mode.	TRACE_MODE
31	TRACE_SEARCH	TS	Searches for information corresponding to acquired trace information.	TRACE_SEARCH
32	UBC_CLEAR	UBC	Clears UBC breakpoints that have been set.	BREAK_CONDITION_UBC
33	UBC_DISPLAY	UBD	Displays UBC breakpoints that have been set.	BREAK_CONDITION_UBC
34	UBC_ENABLE	UBE	Enables or disables UBC breakpoints that have been set.	BREAK_CONDITION_UBC
35	UBC_SET	UBS	Sets UBC breakpoints.	BREAK_CONDITION_UBC

Section 5 Use of Diagnostic Program

5.1 Diagnostic Program Operation

An E8000 emulator diagnostic program can be used on the HDI. The installation procedure and operating details of diagnostic program are given in the SH7060 E8000 Emulator Diagnostic Program Operation Manual. This section describes how to run the diagnostic program on the HDI.

5.1.1 Target File Setting

In HDI installation, target file E8706032.INI is created in the C:\WINDOWS directory. This target file contains resource information indicating whether or not the diagnostic program is to be started when the HDI is initiated.

```
[E8000 HDI TARGET]
Diagnostic Program=Y
```

Figure 5.1 Example of Target File Display

Diagnostic Program=Y

Diagnostic Program=Y is set in the default target file.

If the diagnostic program is not to be run, change the setting as follows:

Diagnostic Program=N

5.1.2 Diagnostic Program Start-Up

When the HDI is initiated, the following dialog box is displayed.

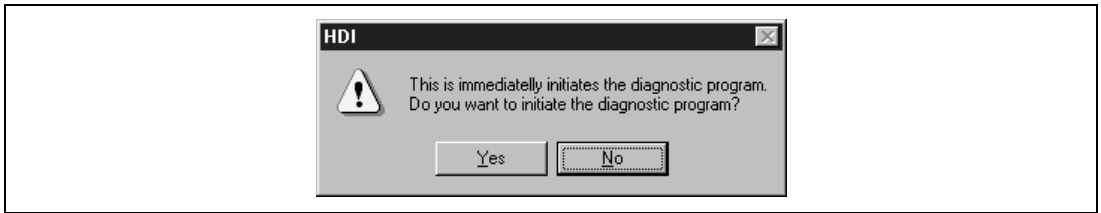


Figure 5.2 Diagnostic Program Start-Up Confirmation Dialog Box

When the [Yes] button is clicked, the diagnostic program is started and the following window is displayed.

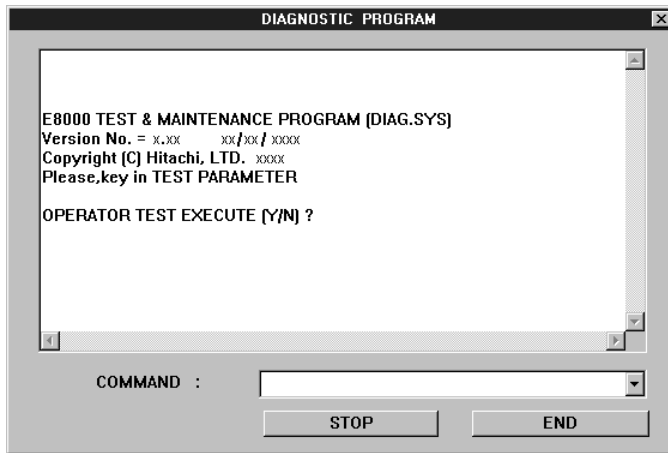


Figure 5.3 Diagnostic Program Start-Up Window

- Display area
Displays the diagnostic program test results.
- [COMMAND] edit box
For input of a diagnostic program operation command.
- [STOP] button
Stops the diagnostic program test and switches to the diagnostic program operation command input mode.
- [END] button
Terminates the diagnostic program and initiates the HDI.

5.1.3 Diagnostic Program Termination

To terminate the diagnostic program, either click the [END] button or else click the [STOP] button to stop the test and then enter Q in the [COMMAND] edit box.

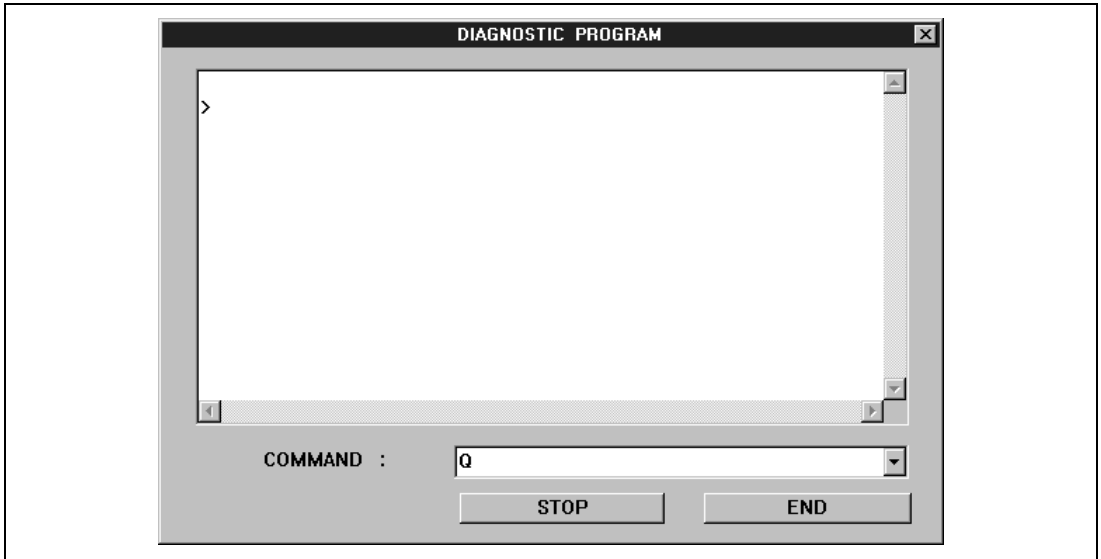


Figure 5.4 Diagnostic Program Termination

Section 6 Error Messages

6.1 Error Messages

The HDI outputs error messages to notify the user of operating errors. The error messages output by the HDI are shown in table 6.1.

Table 6.1 Error Messages

Message	Cause and Countermeasure
Can not set target configuration (Clock mode option)	A mode in which the emulator is not supported has been selected in [Clock] in the [Configuration] window.
Can not set target configuration (CPU mode option)	A mode in which the emulator is not supported has been selected in [CPU mode] in the [Configuration] window.
Can not set target configuration (Execution mode option)	A mode in which the emulator is not supported has been selected in [Execution mode] in the [Configuration] window.
Cannot use command when user program executing	Command line input has been performed when a command cannot be issued to the emulator. Wait for processing to finish.
Command timeout	The HDI has timed out because no response has been received from the emulator after the HDI issued a command. Terminate the HDI, power on the emulator, and restart the HDI.
Emulator command send/receive check error	Communication with the emulator was not performed correctly on HDI activation. Terminate the HDI, power on the emulator, and restart the HDI. If the illegal communication is not corrected, inform a Hitachi sales representative or agency of the situation.
Emulator firmware not ready	“EMULATOR FIRMWARE NOT READY” is output from the emulator. Terminate the HDI and check whether the emulator is functioning normally.
Emulator timeout	A timeout message has been detected from the emulator. Terminate the HDI and check whether the emulator is functioning normally.
Failed to find matching trace record	The HDI has failed to find trace information because there was no matching trace information to be displayed in the [Trace] window. This message is also output when the HDI has no trace information to be displayed.

Table 6.1 Error Messages (cont)

Message	Cause and Countermeasure
Hardware register read/write check error	An error was detected when the emulator hardware registers were tested. Check whether the emulator is functioning normally.
Invalid version number in target configuration	The current HDI version number is different from that when the session file was created. Invalidate the lower version number.
System ID error	An emulator different from that selected using the [Select Platform] dialog box is connected. Check the emulator type.
Target internal error	Commands cannot be issued to the emulator. Wait for processing to finish.
User system not ready	“No clock” has been detected. The HDI is terminated.
Function information not found	The function information input to the [Input Function Range] dialog box was not found. Input the correct function name.
When specifying byte access for the data condition, also specify an address condition.	When the byte access condition is set for the data bus condition, also set the address bus condition. In this case, the address range, mask, and Outside Range cannot be set for the address bus condition.