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

User's Manual

SH7410 E8000 Debugging Interface

User's Manual

Tutorial

Renesas Electronics
www.renesas.com

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.

Contents

1 Introduction

This tutorial is a modified version of Section 3, Tutorial in the SH7410 E8000 Hitachi Debugging Interface User's Manual for the SH7410 E8000 Hitachi debugging interface (hereafter referred to as HDI). Refer to this tutorial when using the HDI.

The following describes a sample program for sorting random data in order to introduce the main functions of the HDI.

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 1 shows the configuration of the sample program.

Table 1 Configuration of Sample Program

Item No.	Item	Description
1	Tutorial file (load module in the SYSROF format)	SORT.ABS
2	Tutorial file (source file)	SORT.C

Note: In this section, the SH7410 E8000 HDI is used to simplify the explanation. When the SH7612 E8000 HDI is to be used, refer to the notes in each description.

2 Running the HDI

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

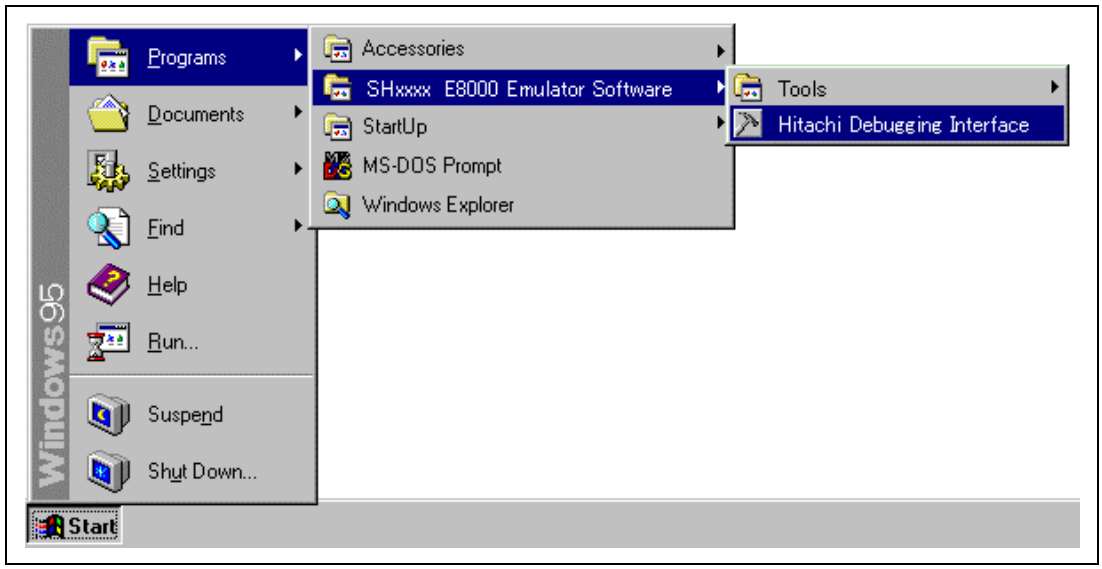

Figure 1 [Start] Menu

3 Selecting the Target Platform

The HDI supports two target platforms: the E8000 SH7410 emulator and the E8000 SH7612 emulator. When the HDI is initiated, the dialog box for selection of the platform for the current session appears.

Select the [E8000 SH7410 Emulator].

Note: When the E8000 SH7612 emulator is to be used, select the [E8000 SH7612 Emulator].

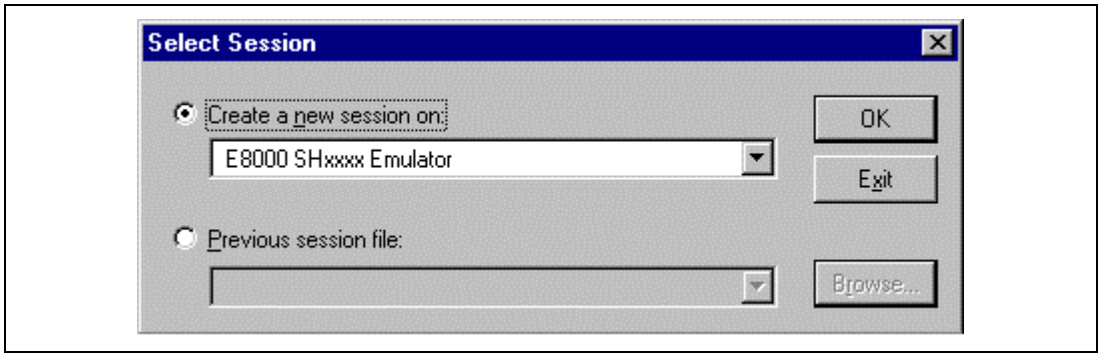

Figure 2 [Select Session] Dialog Box

To change the target platform, select [New Session...] from the [File] menu.

When the emulator is correctly set, the HDI window appears together with the `Link up` message in the status bar. The details of the HDI window are described in the following sections.

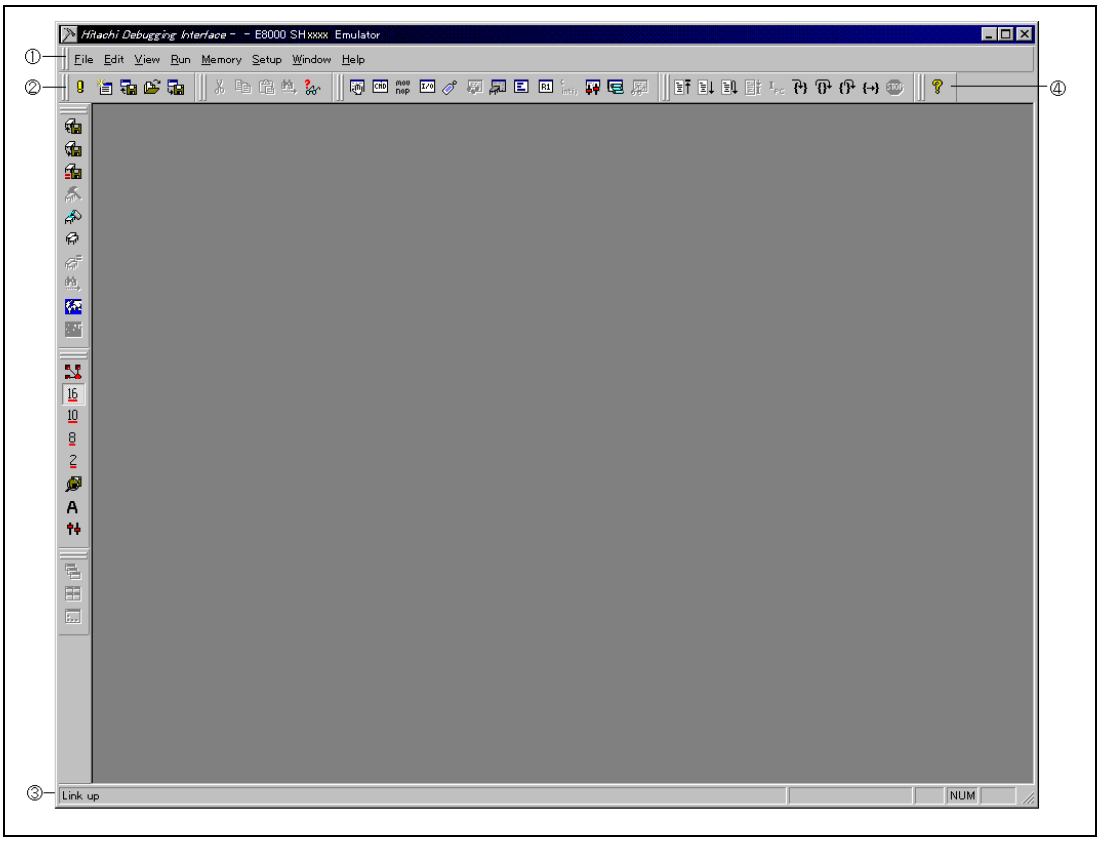

Figure 3 HDI Window

Numbers in figure 3 indicate the following:

1. Menu bar
   Indicates the HDI command menus for the use of the HDI debugger.
2. Toolbar
   Contains convenient buttons as shortcuts of menu commands most frequently used.
3. Status bar
   Indicates the state of the emulator and progress information about downloading.
4. [Help] button
   Activates the HDI on-line help.

4 Setting up the Emulator

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

- Device type
- Bus width in the CS0 area
- Clock mode
- Operating clock
- Execution operating mode
- Memory map

The following describes how to set up the emulator correctly for the tutorial programs.

4.1 Setting the [Configuration] Dialog Box

Select [Configure Platform...] from the [Setup] menu to set configuration. The following dialog box is displayed:

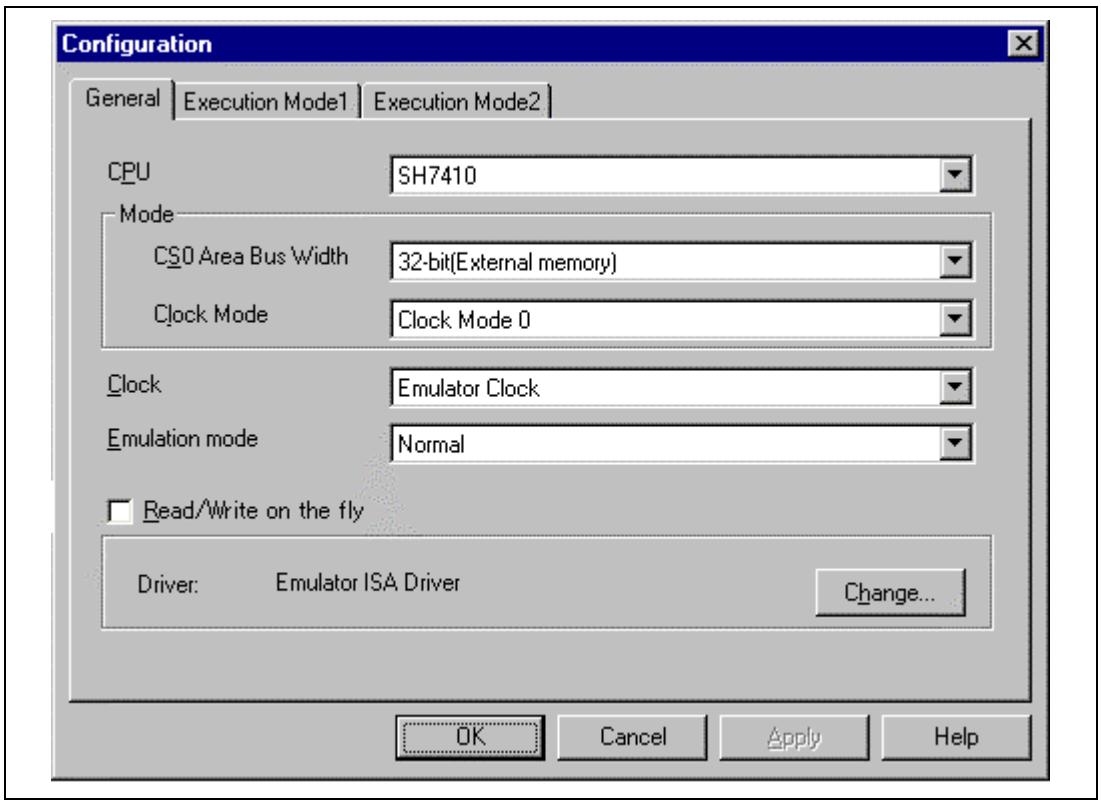

Figure 4 [Configuration] Dialog Box

Set options as follows:

Table 2 Setting the [Configuration] Dialog Box

Option	Value
Bus width in the CS0 area [CS0 Area Bus Width]	32 bits (External memory)
Clock mode [Clock Mode]	Clock Mode 0
Emulation clock [Clock]	Emulator Clock (using the emulator clock)
Emulation mode [Emulation mode]	Normal (normal execution)
Memory access enable/disable during execution [Read/Write on the fly]	Enable (no check)
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)
Bus width of the emulation memory [Emulation memory bus width] *1	32-bit bus width (default setting)
User-wait control [Enable user wait]	Invalid (default setting)
BREQ signal control [Enable the BREQ signal input]	Invalid (default setting)
Execution-count measurement-mode control of performance function [ECNT Option] *1	Upper (default setting)
Cache access trace control [Enable the Cache access trace] *2	Invalid (default setting)
Multi-break function [Enable the multi break of External probe No. 1] *2	Invalid (default setting)
UBC control [Enable UBC for user program] *2	Invalid (default setting)
Trigger output control 1 at break [TRGU Option]	Upper (default setting)
Trigger output control 2 at break [TRGB Option]	Upper (default setting)

- Click the [OK] button to set any changes in the configuration.

4.2 Setting the Memory Map

In the next step, allocate the emulation memory for the developing application.

Select [Configure Map...] from the [Memory] menu to display the current memory map. The [Memory Mapping] dialog box is displayed.

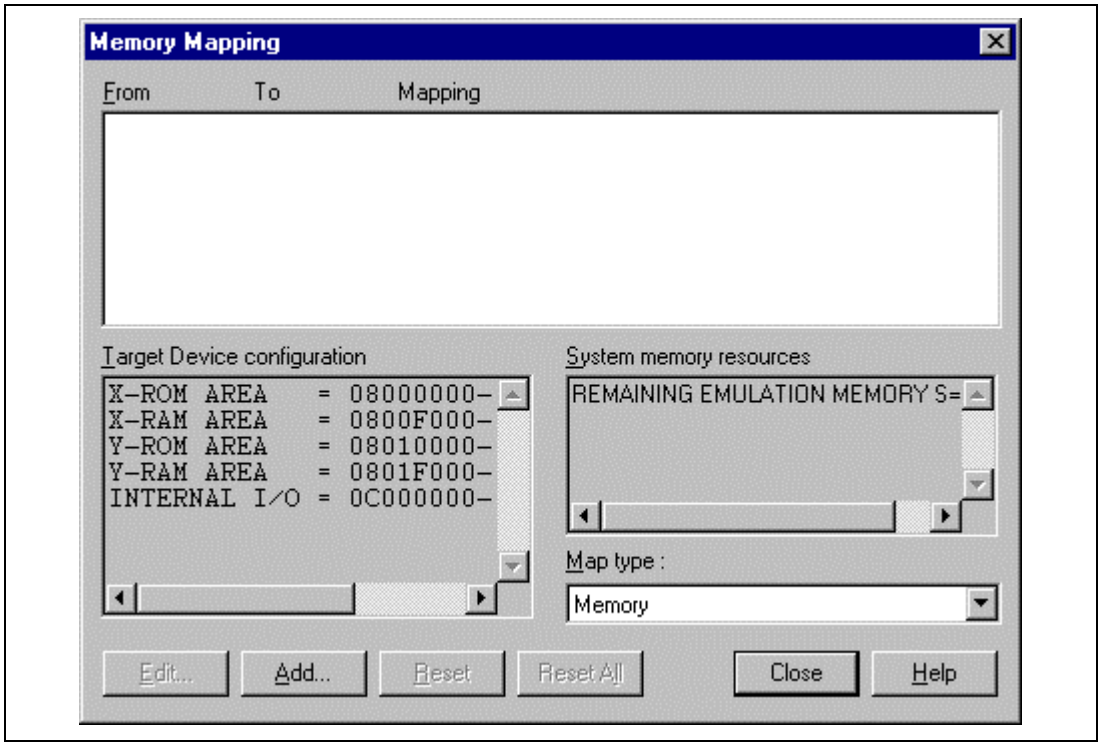

Figure 5 [Memory Mapping] Dialog Box

Note: When the SH7612 E8000 HDI is used, the information displayed in the [Target Device Configuration] is different from that shown in figure 5.

The emulator can allocate the optional memory area as one of the following two types:

Table 3 Memory Type

Memory Type	Description
USER AREA	Sets the address range of the emulation memory area.
EMULATION AREA Read-Only	Sets the emulation memory area to be write-protected.

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

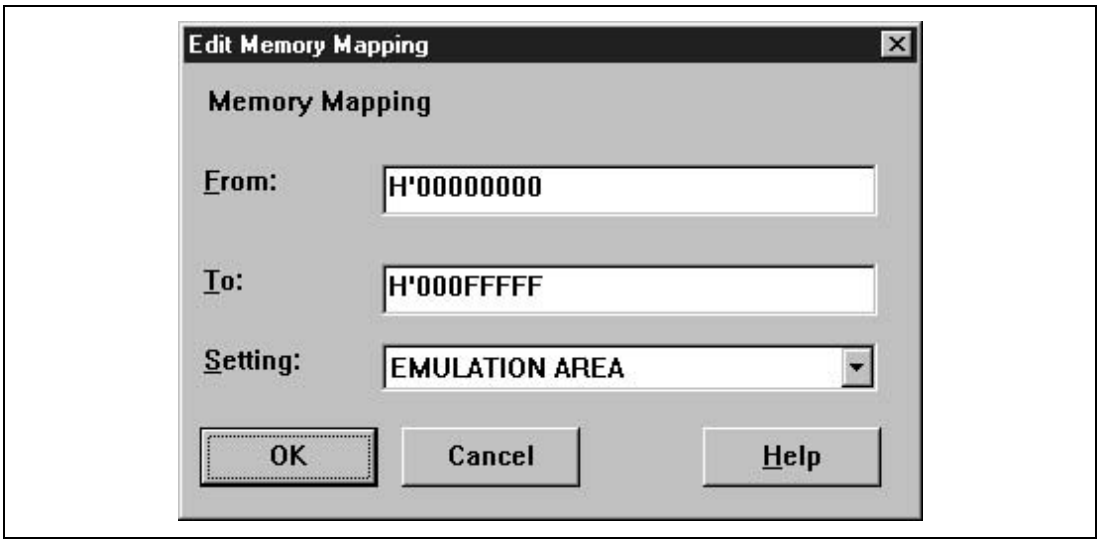

Figure 6 [Edit Memory Mapping] Dialog Box

For this tutorial, allocate the memory area of addresses ranging from H’00000000 to H’000FFFFF as an emulation memory area.

- Set the [From] and [To] edit boxes to ***H’00000000*** and ***H’000FFFFF***, 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.

- Click the close box [X] in the upper-right corner of the [Memory Mapping] dialog box to close the window.

5 Downloading the Tutorial Program

5.1 Downloading the Tutorial Program

Download the object program to be debugged.

- Select [Load Program...] from the [File] menu. The [Load Program...] dialog box is displayed.
- Click the [Browse...] button. The [Open] dialog box is displayed.
- Select the file `Sort.abs` in the `Tutorial` directory, and click the [Open] button.

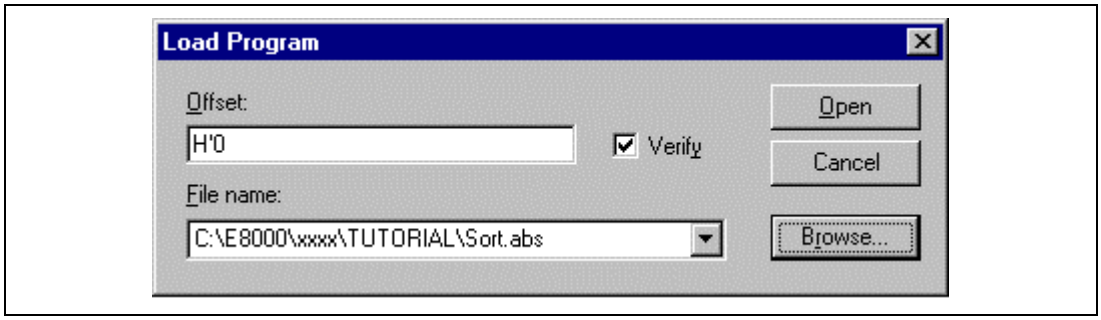

Figure 7 [Load Object File] Dialog Box

- Click the [Open] 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.

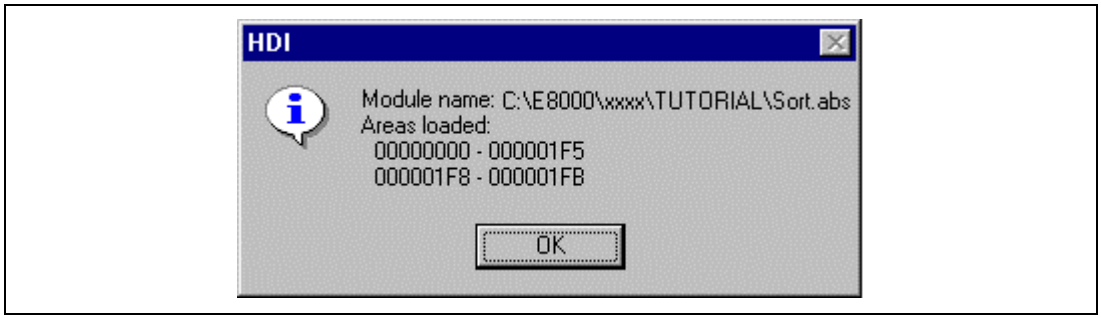

Figure 8 HDI Dialog Box

- Click the [OK] button to continue.

5.2 Displaying the Source Program

The HDI allows the user to debug a program at the source level, so that the user can see a list of the C/C++ program alongside the machine code as the user debugs. To do this, the C/C++ source file that corresponds to the object file needs to be read.

- Select [Source...] from the [View] menu. The [Open] dialog box is displayed.
- Select the file `Sort.c`, and click the [Open] button.

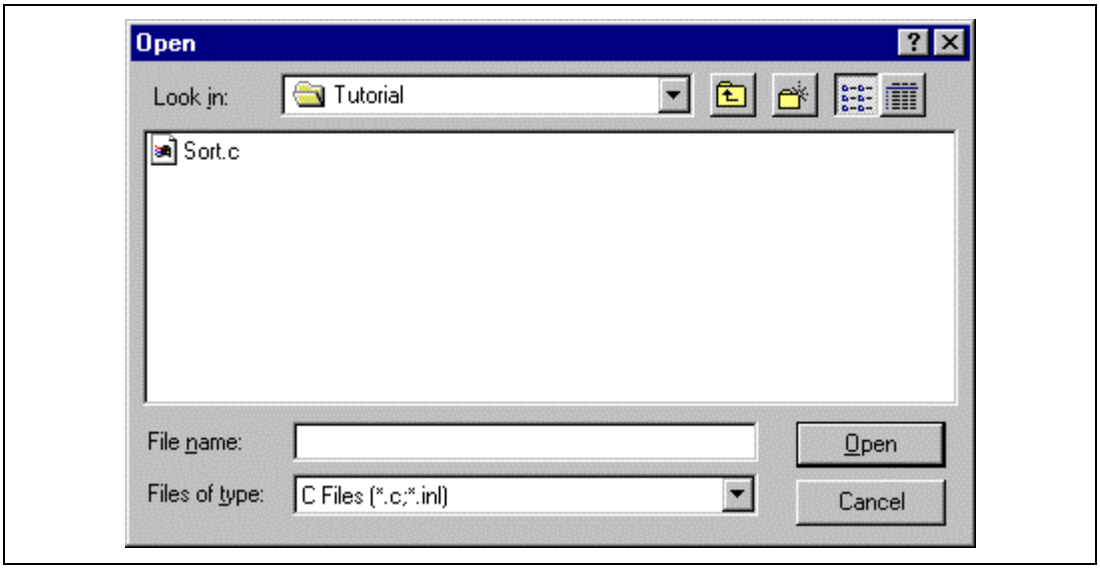

Figure 9 [Open] Dialog Box

- Select [Sort.c] and click the [OK] button. The [Source] window is displayed.

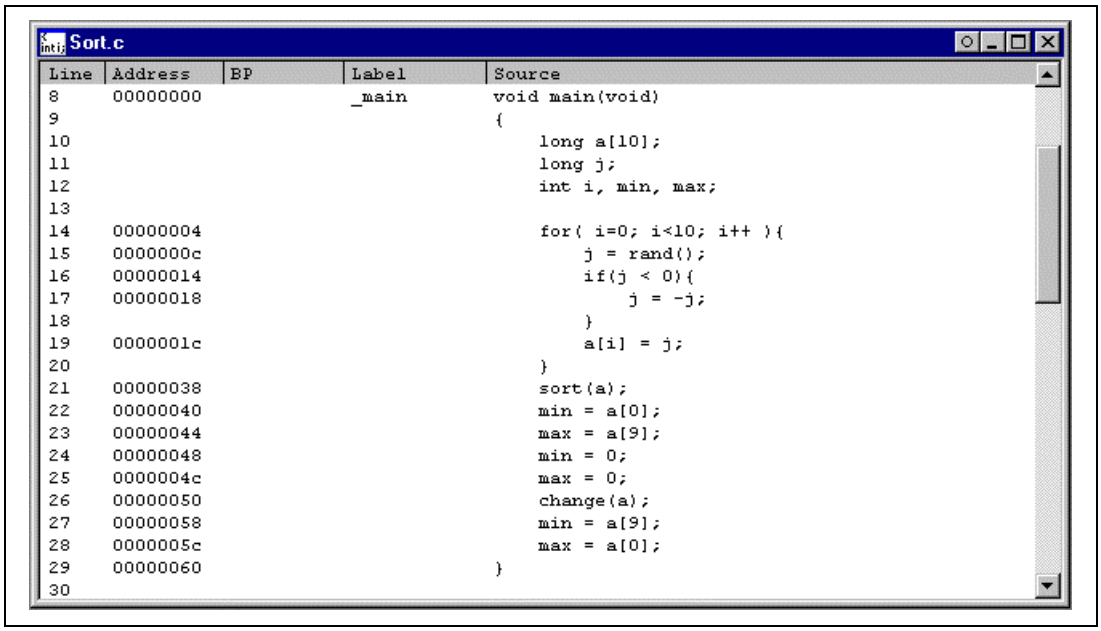

Figure 10 [Source] Window (Displaying the Source Program)

- If necessary, select the [Font] option from the [Customize] submenu on the [Setup] menu to select a font and size suitable for the host computer.

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

6 Setting the Software Breakpoint

A breakpoint is one of the easy debugging functions.

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

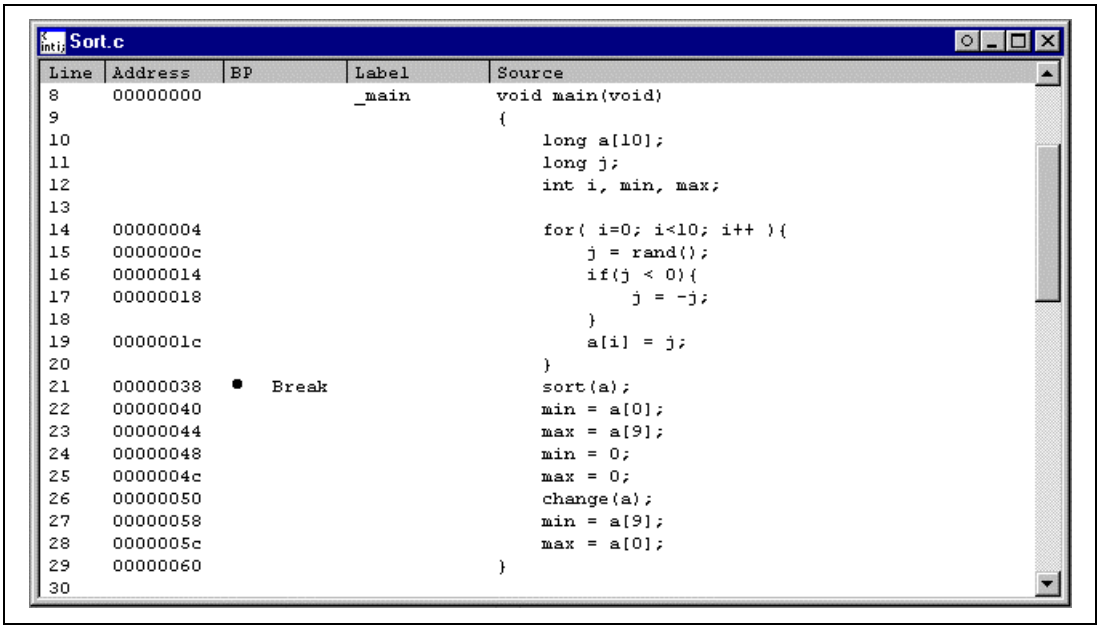

Figure 11 [Source] Window (Setting a Software Breakpoint)

Note: The software breakpoint cannot be set in the ROM area. However, in SH7410 E8000, it can be set in the internal ROM area.

7 Setting Registers

Set values of the program counter and the stack pointer before executing the program.

- Select [Registers] from the [View] menu. The [Registers] window is displayed.

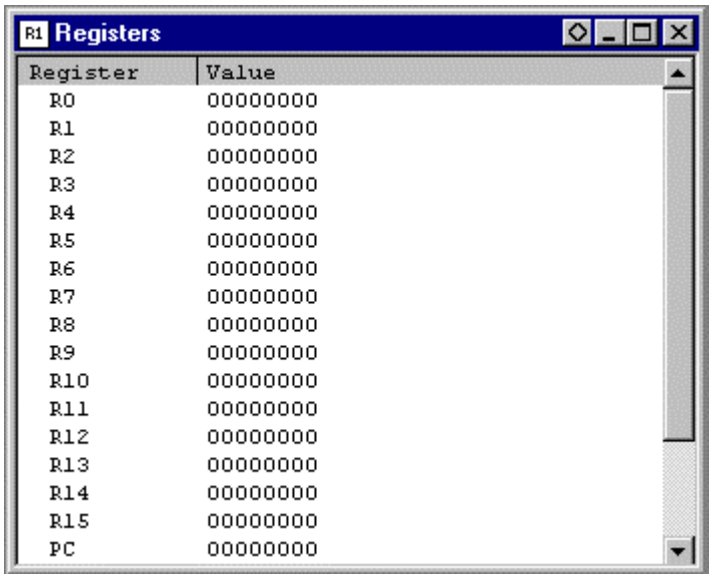

Figure 12 [Registers] Window

- Double-click [PC] in the [Registers] window to change the value of the program counter (PC).

The following dialog box enables the value to be changed.

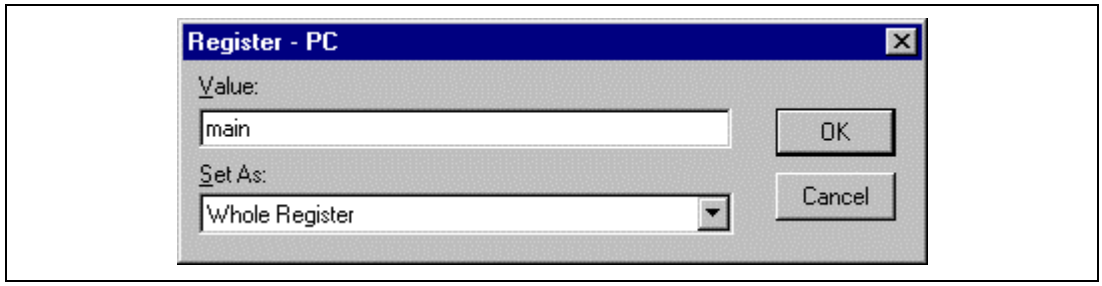

Figure 13 [Register] Dialog Box (PC)

- Set ***main*** in this sample program, and click the [OK] button.
- Double-click [R15] in the [Registers] window to change the value of the stack pointer (R15). In the same way of setting the program counter, the stack pointer can be changed by the [Register] dialog box.

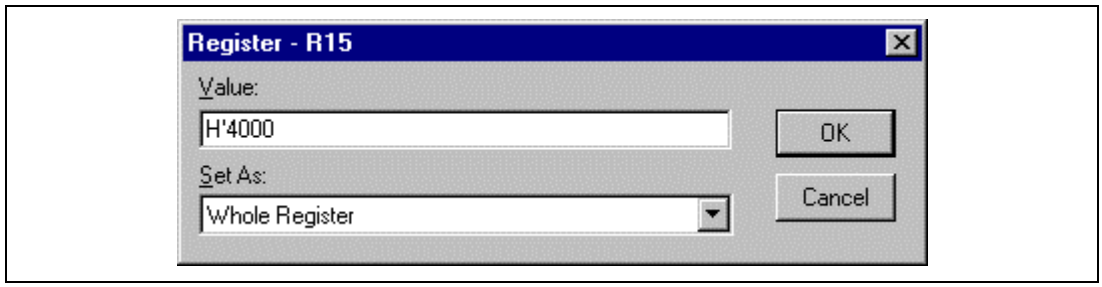

Figure 14 [Register] Dialog Box (R15)

- Set ***H'4000*** for the value of the stack pointer in this sample program, and click the [OK] button.

8 Executing the Program

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

Figure 15 [Go] Button

The program will be executed up to the breakpoint that has been inserted, and a statement will be highlighted in the [Source] window to show the position that the program has halted.

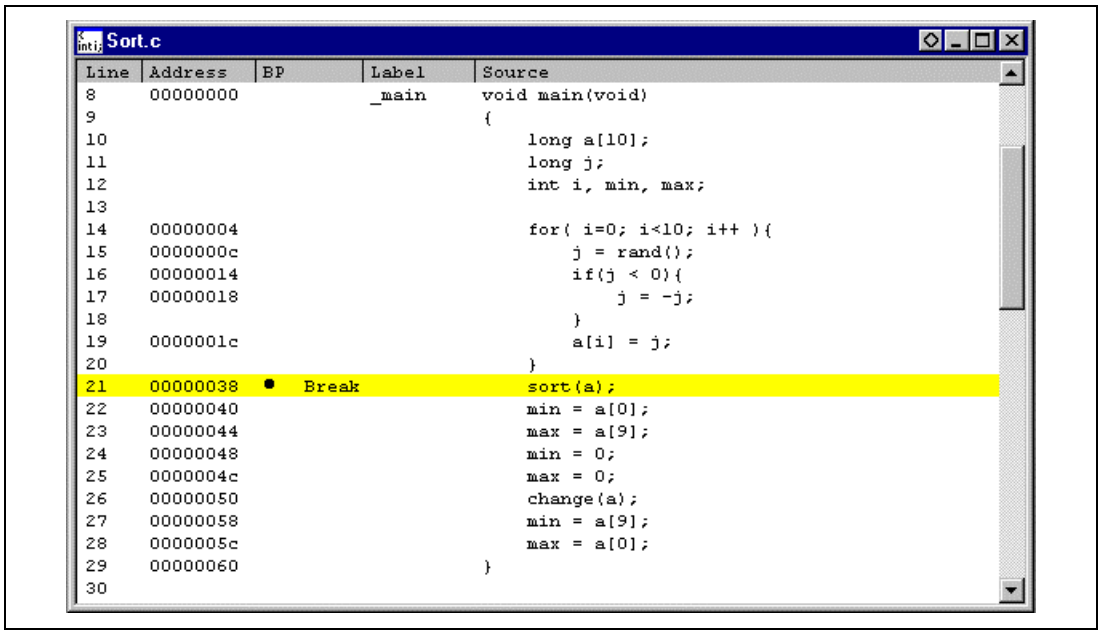

Figure 16 [Source] Window (Break State)

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.

- Select the [Platform] sheet from the [System Status] window.

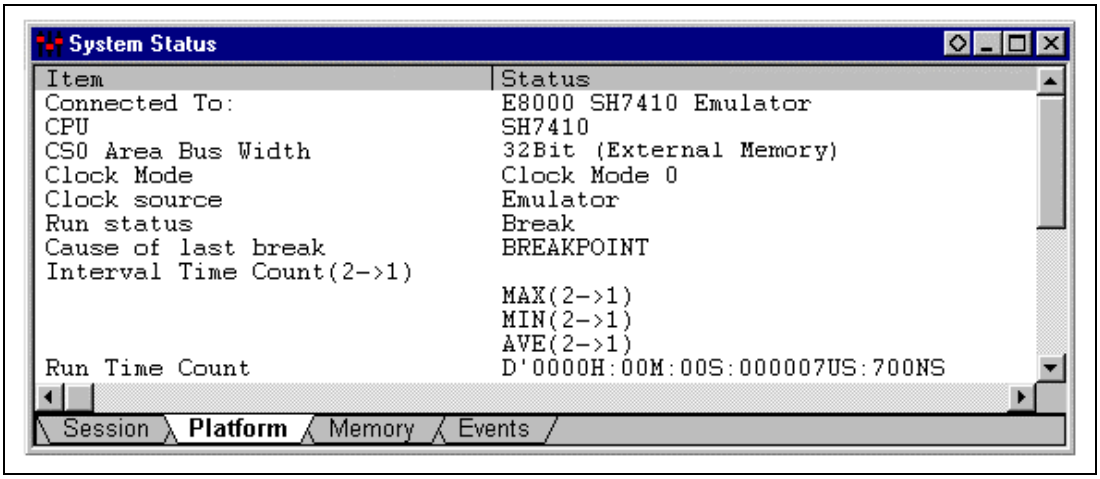

Figure 17 [System Status] Window

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

Note: When the SH7612 E8000 HDI is used, the information items displayed on the [Connected To:] and [CPU] lines are different from those of the SH7410 E8000 HDI.

9 Reviewing Breakpoints

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

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

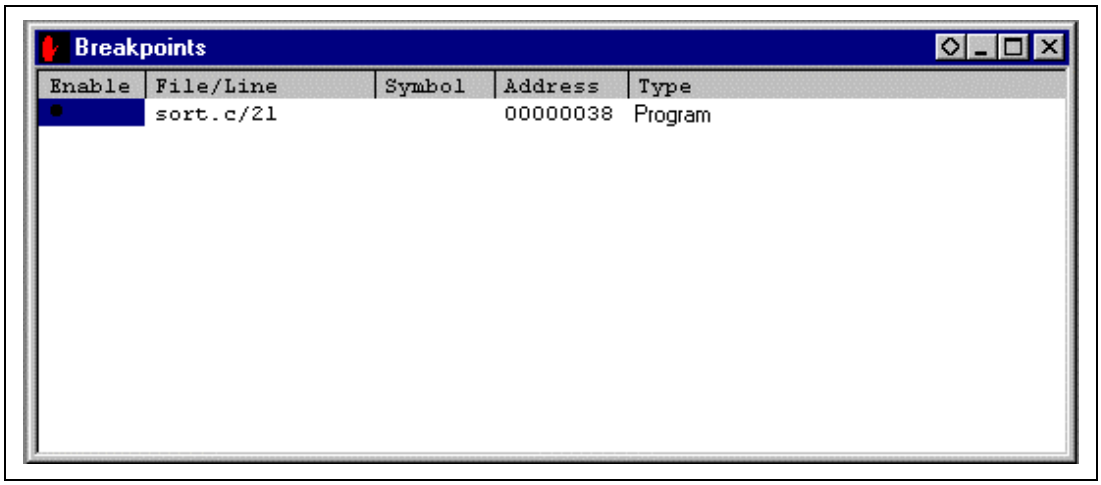

Figure 18 [Breakpoints] Window

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

- Close the [Breakpoints] window.

10 Viewing Memory

The user can view the contents of a memory block in the [Memory] window. For example, to view the memory corresponding to the `main` function in word size:

- Select [Memory...] from the [View] menu. The [Open Memory Window] dialog box is displayed.

  Input ***`main`*** in the [Address] edit box, and set the [Format] combo box as [Word].

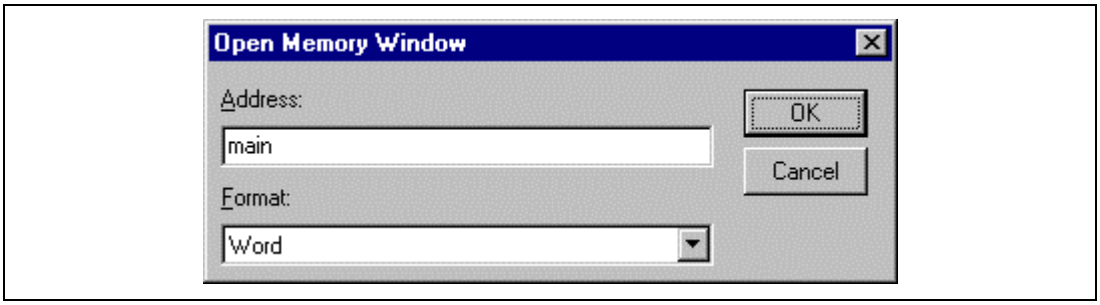

Figure 19 [Open Memory Window] Dialog Box

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

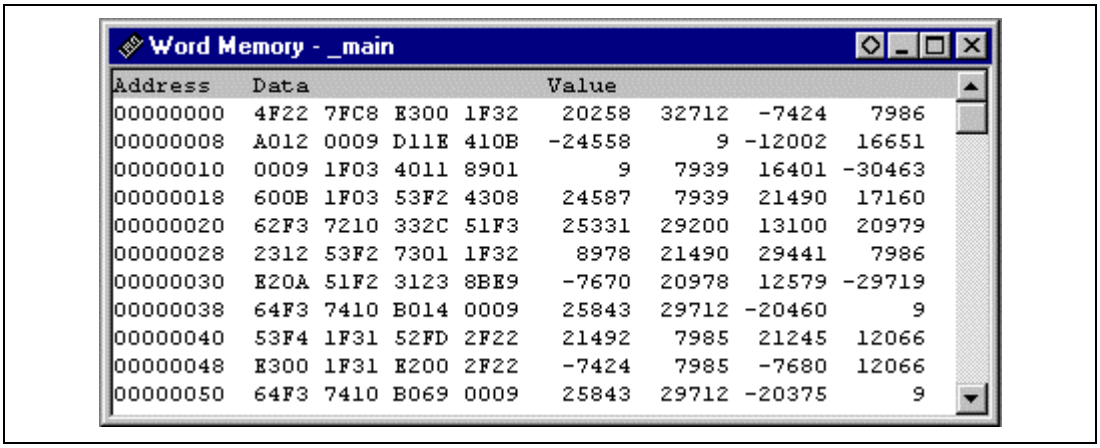

Figure 20 [Word Memory] Window

11 Watching Variables

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

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

The following dialog box will be displayed.

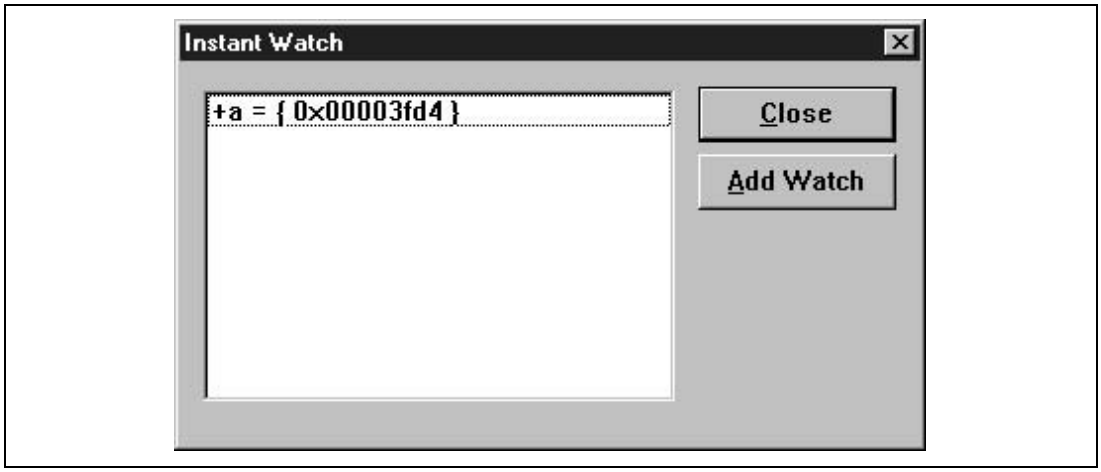

Figure 21 [Instant Watch] Dialog Box

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

Figure 22 [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 popup menu.

The following dialog box will be displayed.

Figure 23 [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 24 [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.

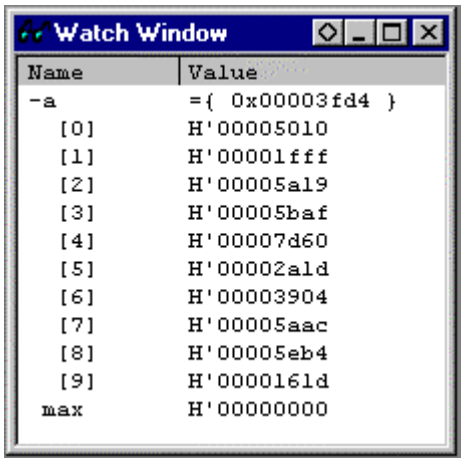

Figure 25 [Watch] Window (Displaying Array Elements)

12 Stepping Through a Program

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

Table 4 Step Option

Menu Command	Description
Step In	Executes each statement.
Step Over	Executes each statement. (A function call is executed in a single step.)
Step Out	Steps out of a function, and stops at the next statement that called the function in the program.
Step…	Steps the specified counts repeatedly at a specified rate.

To demonstrate program stepping, confirm that the `sort` function statement at address H'00000038 has been executed.

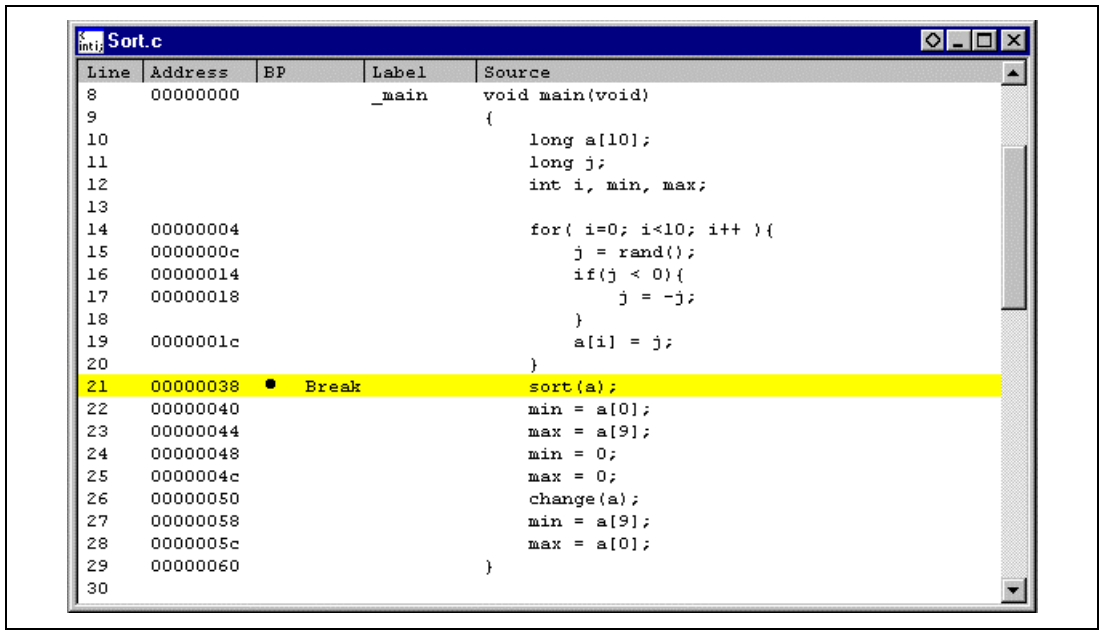

Figure 26 [Source] Window (Step Execution)

12.1 Executing [Step In] Command

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

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

Figure 27 [Step In] Button

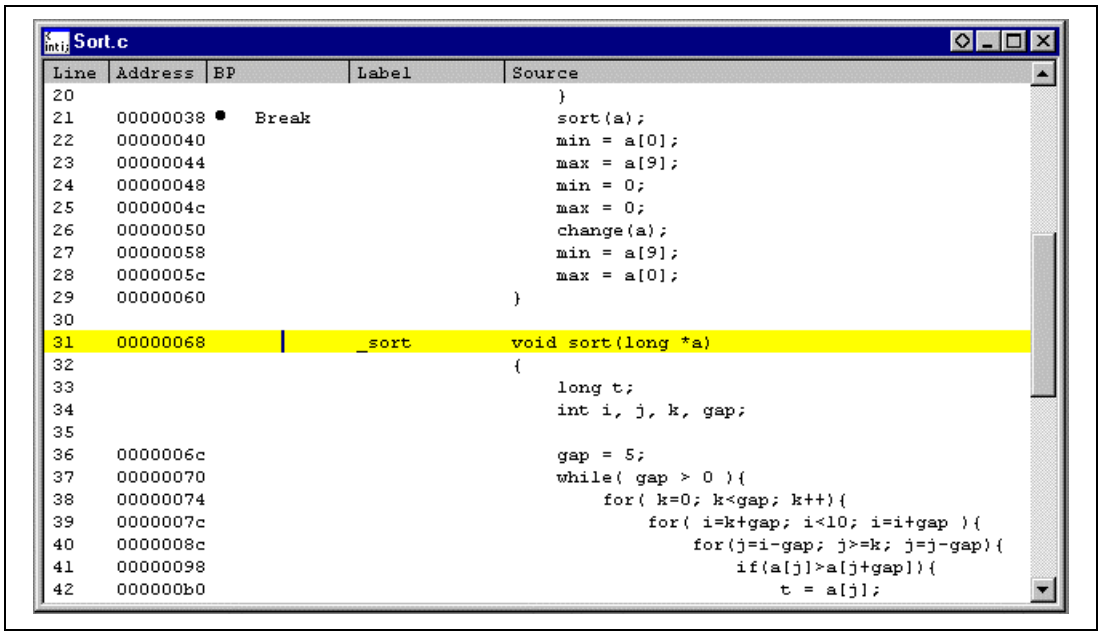

Figure 28 [Source] Window (Step In)

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

12.2 Executing [Step Out] Command

The [Step Out] command steps out of the called function and stops at the next statement 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 29 [Step Out] Button

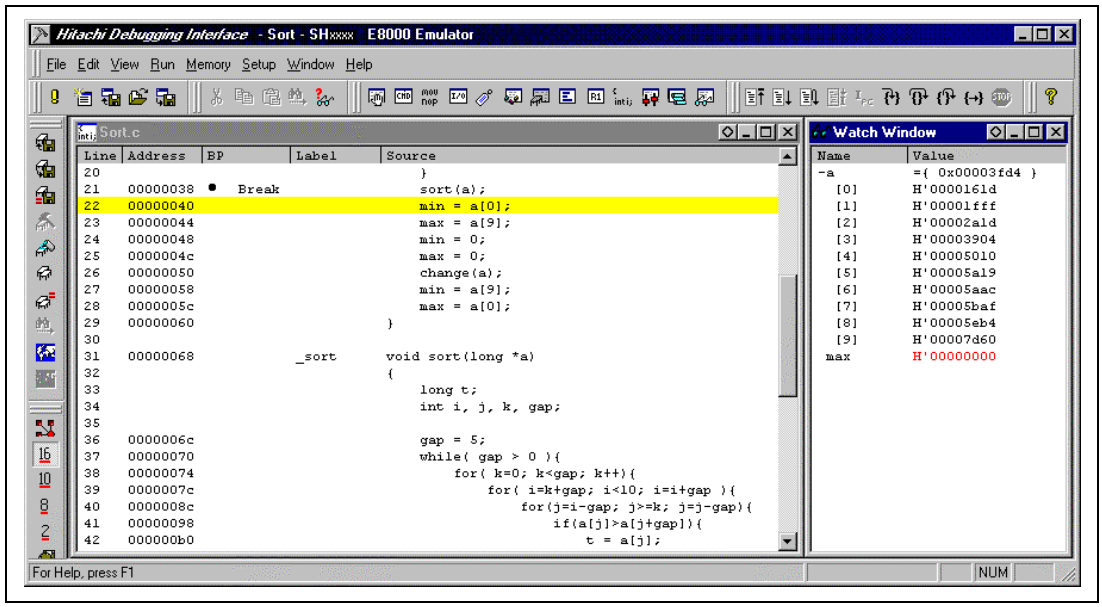

Figure 30 [Source] Window (Step Out)

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

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

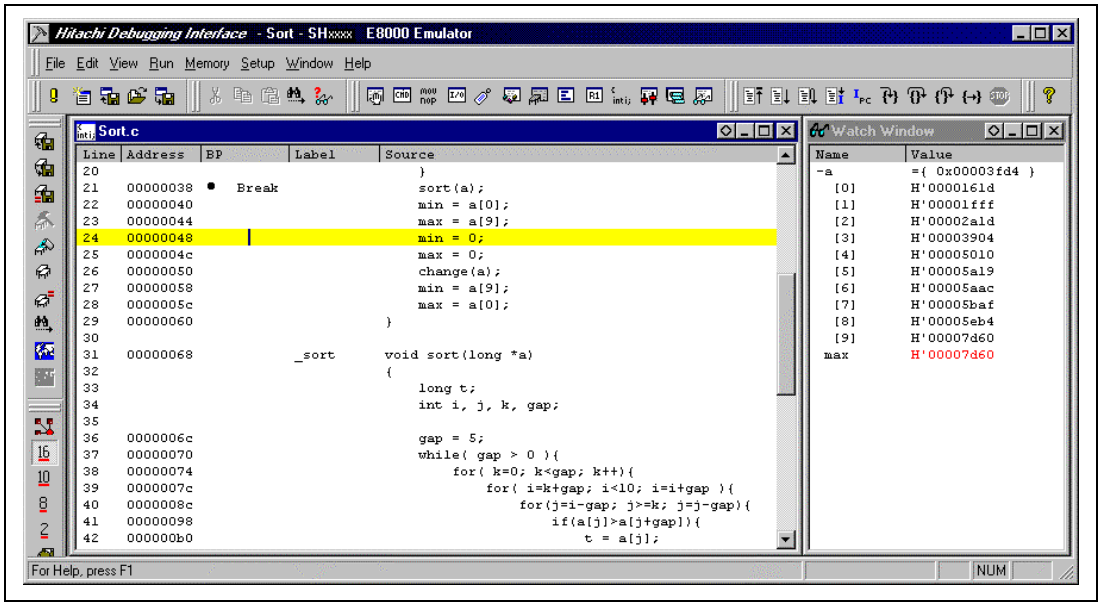

Figure 31 [Source] Window (Step Out -> Step In)

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

12.3 Executing [Step Over] Command

The [Step Over] executes a function call as a single step and stops at the next statement of the main program.

- To demonstrate the [Step Over] command, execute two steps to reach the `change` function statement.

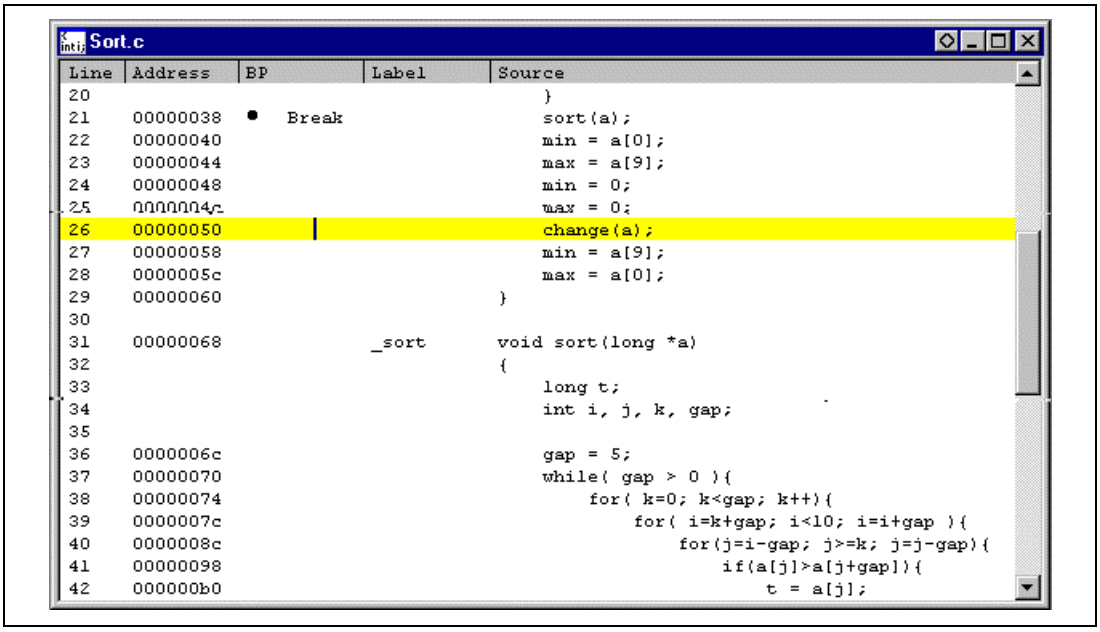

Figure 32 [Source] Window (Before Step Over Execution)

- To step through all statements in the change function at a single step, select [Step Over] from the [Run] menu, or click the [Step Over] button in the toolbar.

Figure 33 [Step Over] Button

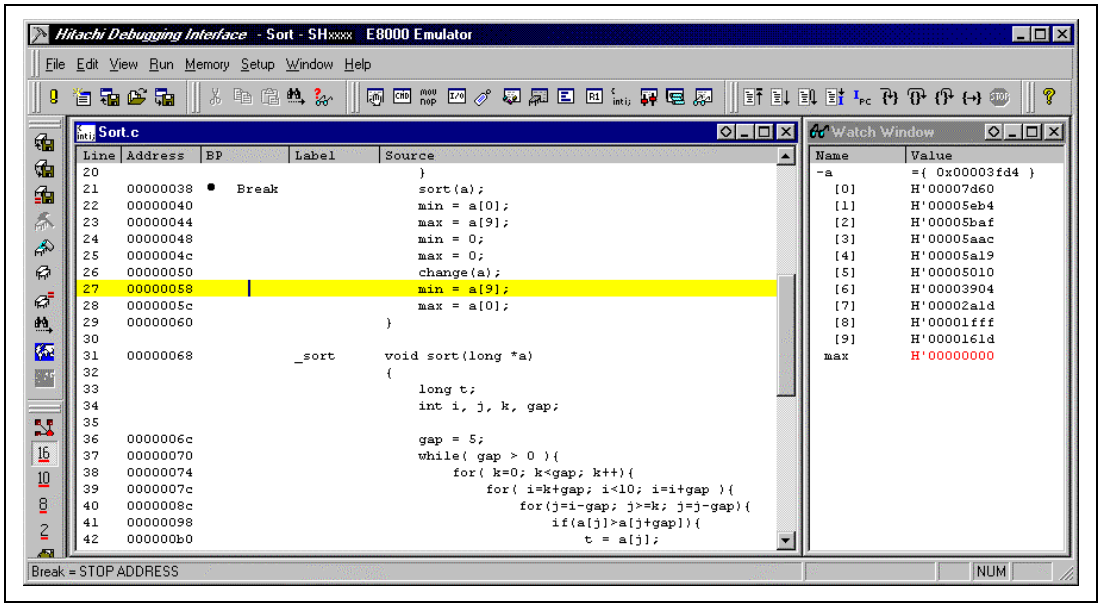

Figure 34 [Source] Window (Step Over)

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

13 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 a single step.

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

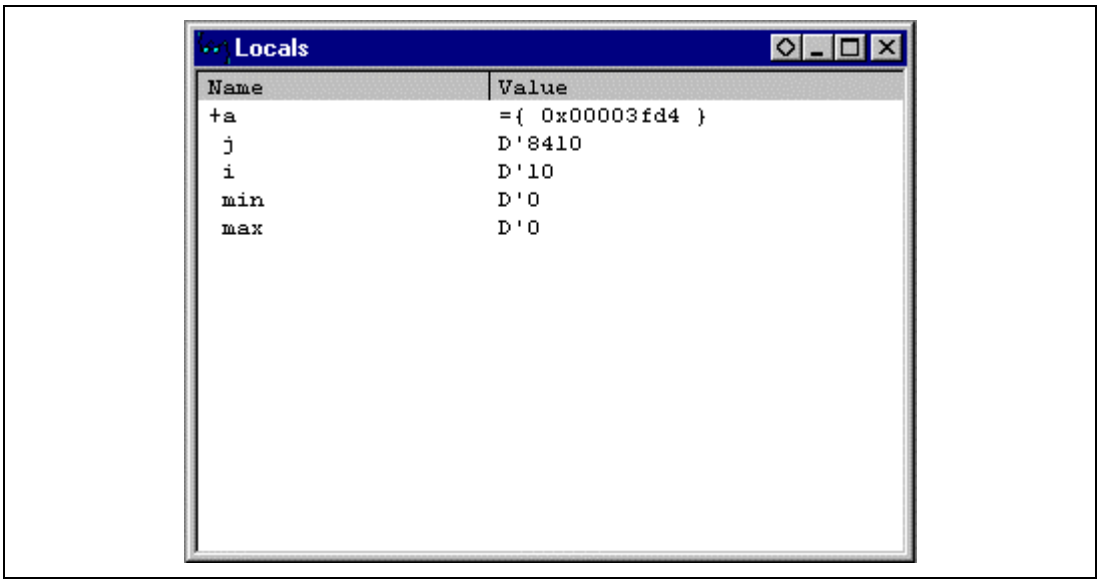

Figure 35 [Locals] Window

- 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.

14 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 5 Dialog Boxes for Setting Hardware Break Conditions

Function / Dialog Box	Address Bus Condition (Address)	Data Bus Condition (Data)	Status Condition (Status)	External Probe Condition (Probe)	Interrupt Condition (Interrupt)	Satis-faction Count (Count)	DELAY Condition (Delay)*3
[Break Condition UBC1] dialog box	O	O	O	X	X	O	X
[Break Condition UBC2] dialog box	O	X	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. Only Break Condition B7 can be set for the DELAY condition in the [Break Condition B] dialog box.

Table 6 Main Break Conditions

Break Condition	Description
Address bus condition (Address)	Breaks on a match of the CPU address bus value.
Data bus condition (Data)	Breaks on a match of the CPU data bus value. 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 CPU R/W signal level matches the specified conditions. Bus state condition: Breaks when the operating state in the CPU memory access cycle, DMA cycle, or vector fetch cycle matches 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 or an external interrupt signal (IRQ0 to IRQ3) level matches the specified conditions.
Satisfaction Count (Count)	Breaks when all the above conditions have been satisfied the number of times specified in this condition. (A maximum count of 65,536 can be specified.)
DELAY condition (Delay)	Breaks when all 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.)

Note: When the SH7612 E8000 HDI is used, the external interrupt signal [IRL] is displayed instead of [IRQ].

An example is given below in which the address bus condition and read cycles for 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 right mouse button on the [Breakpoints] window to display the popup menu.
- Select [Add...] to display the [Break] dialog box.

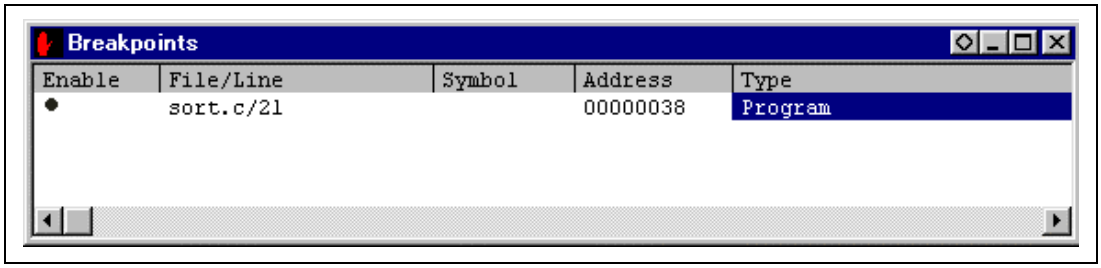

Figure 36 [Breakpoints] Window (Before Hardware Break Condition Setting)

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.

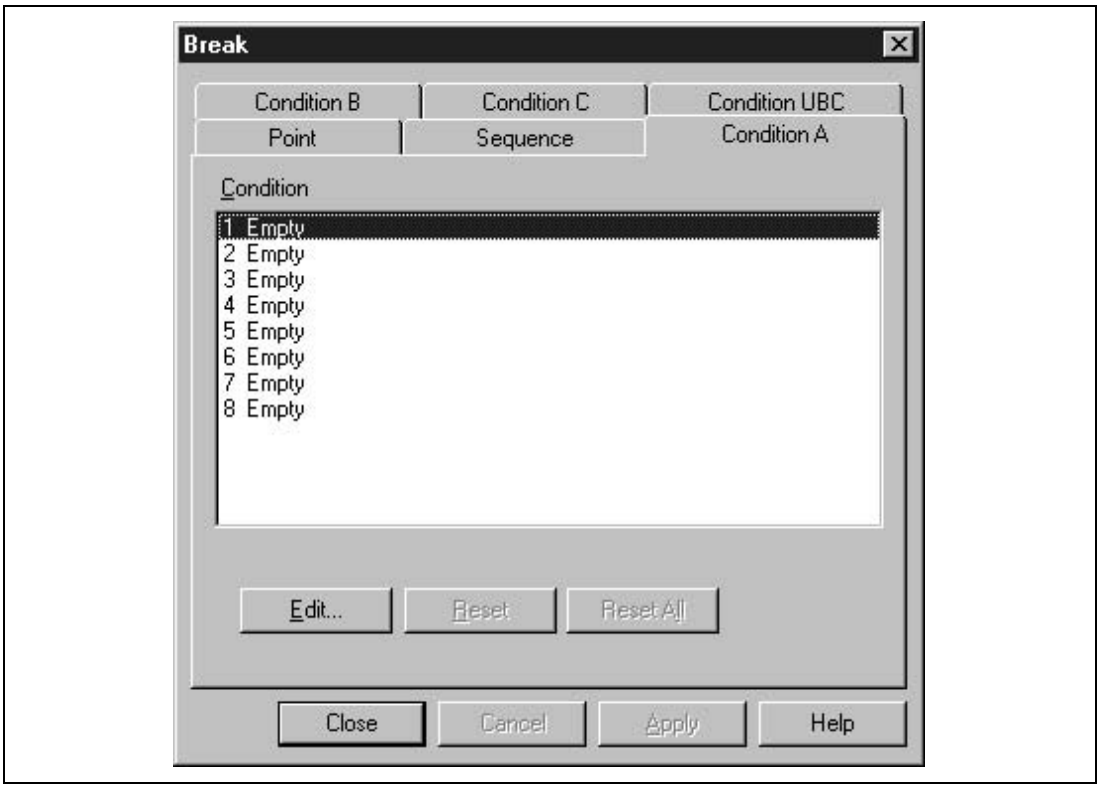

Figure 37 [Condition A] Page ([Break] Dialog Box)

Up to eight breakpoints can be set independently for 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'5A*** as the value in the [Start] edit box.

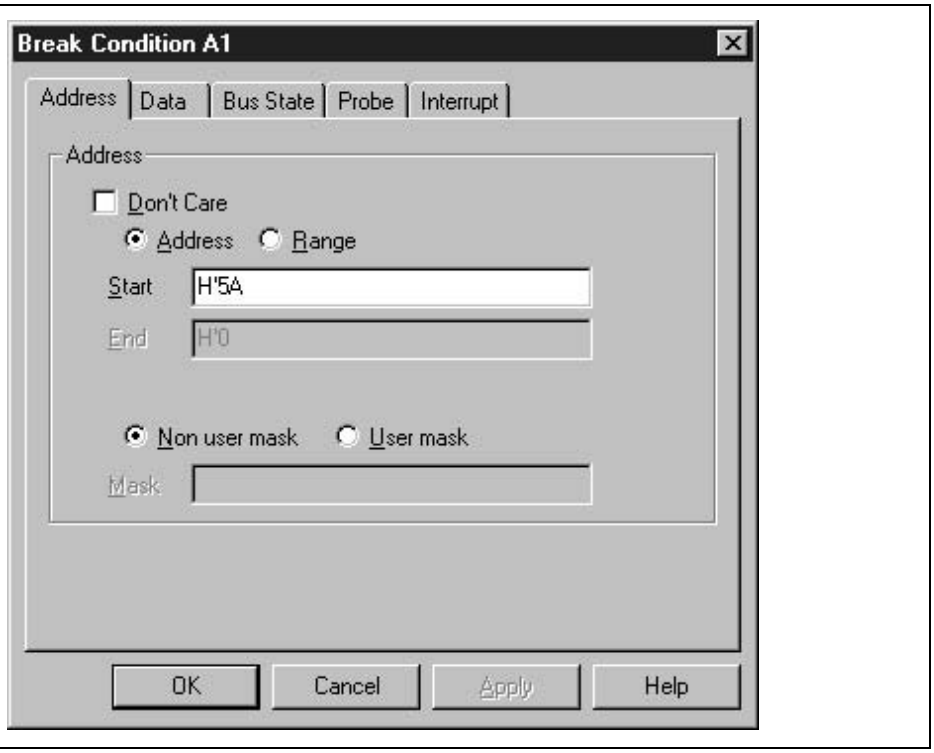

Figure 38 [Address] Page ([Break Condition A1] Dialog Box)

- Select [Bus State] to display the [Bus State] page.
- Select the [Read] radio button.

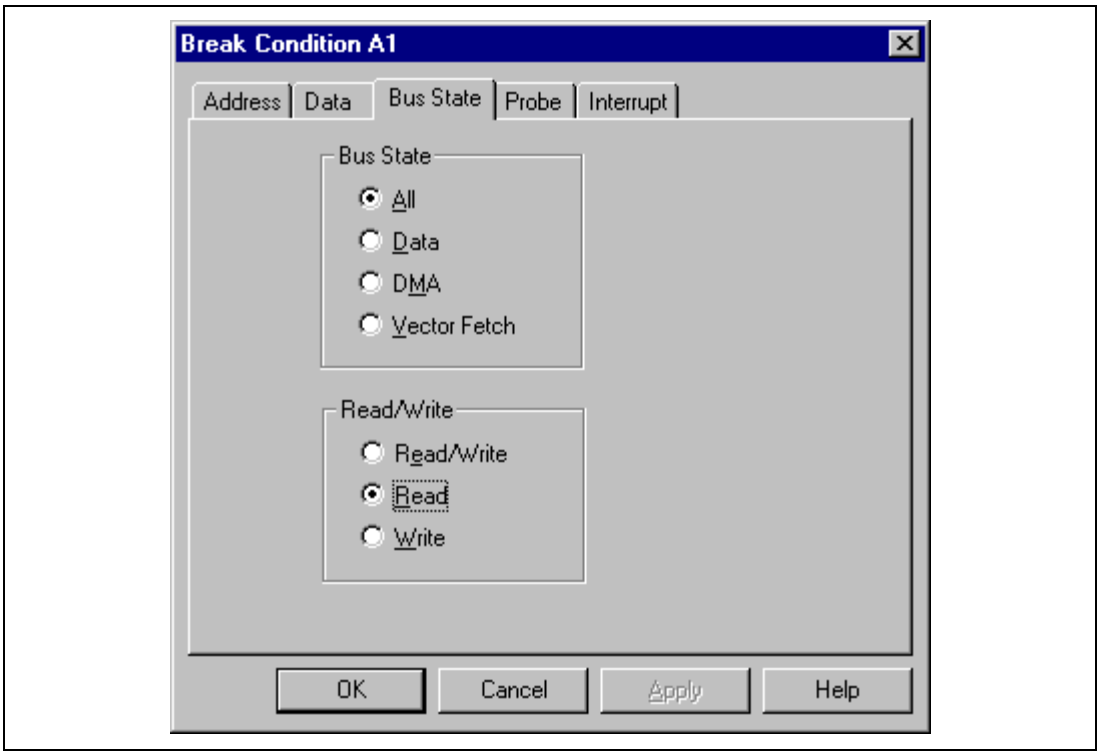

Figure 39 [Bus State] Page ([Break Condition A1] Dialog Box)

- 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`.

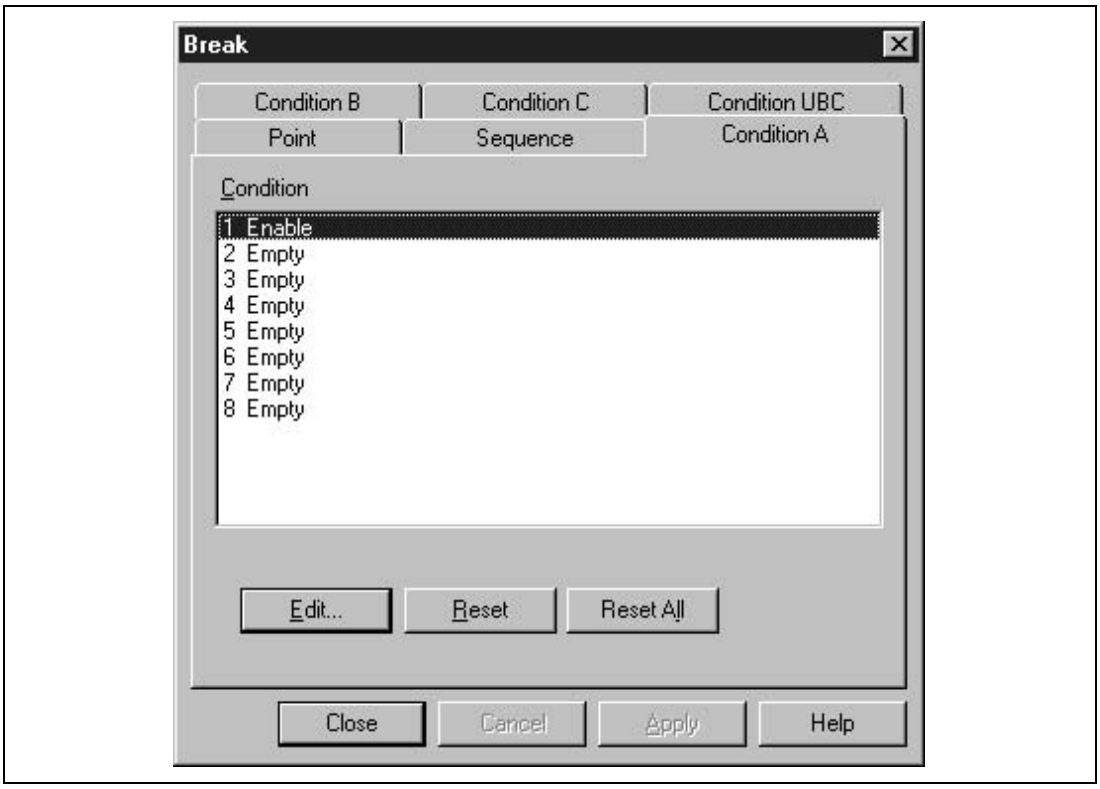

Figure 40 [Break] Dialog Box (After Hardware Break Condition Setting)

- 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. When the program is executed, a break will occur when address H'5A is accessed in a read cycle.

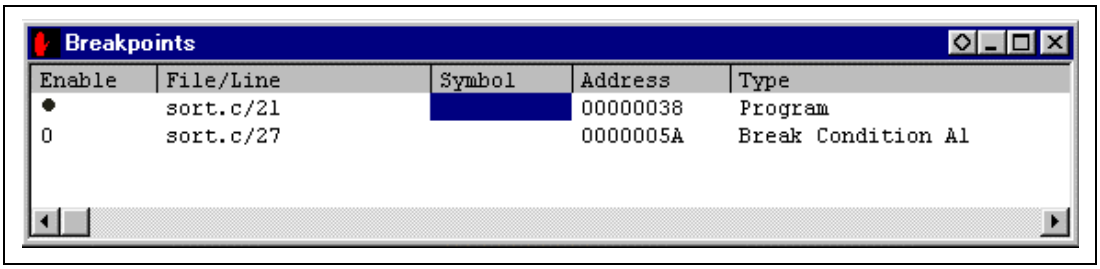

Figure 41 [Breakpoints] Window ([Break Condition A] Setting)

15 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 7 Dialog Boxes for Setting Sequential Break Conditions

Function / Dialog Box	Address Bus Condition (Address)	Data Bus Condition (Data)	Status Condition (Status)	External Probe Condition (Probe)	Interrupt 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	X	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 8 Main Sequential Break Functions

Sequential Break Function	Description
Break Sequence	A sequential break function using software breaks. Up to 7 address points and 1 reset point address can be set. When all the set points are passed in sequence, the program is stopped.
Break Condition UBC1, 2	Sequential break functions by combining satisfaction conditions of hardware break conditions, i.e., Break Condition UBC1 and 2. Program execution is halted when conditions are satisfied in the order of UBC2 and UBC1. (Sequential break mode UBC 2 -> 1)

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, and the [Configuration] dialog box will appear.
- Select [Sequential break mode UBC2 -> 1] from the [Emulation Mode] combo box.

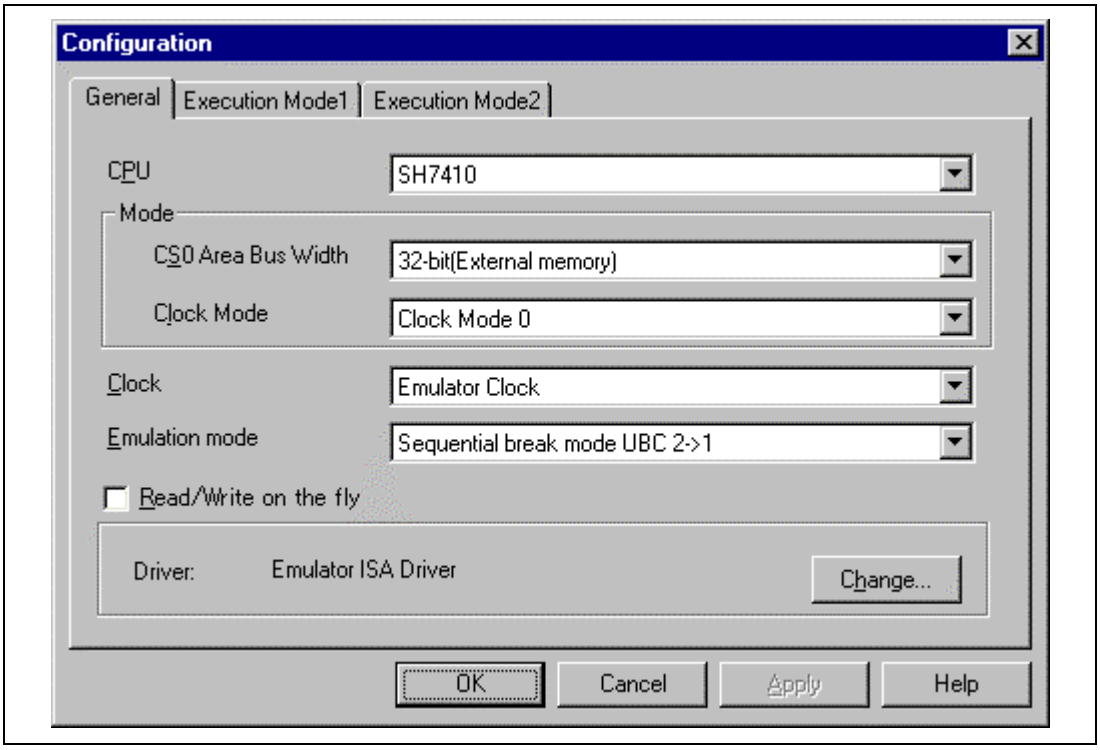

Figure 42 [Configuration] Dialog Box (When Sequential Break is Set)

- 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'58 is accessed in a read cycle. (Break Condition UBC1)

Break condition 2: A break is executed when address H'48 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 right mouse button on the [Breakpoints] window to display the popup menu. Select [Delete All] to clear all the set break points.
- Display the popup menu and select [Add...]. The [Break] dialog box is displayed.

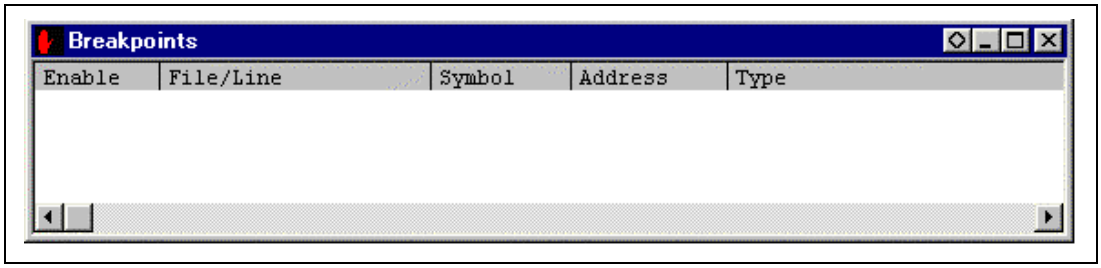

Figure 43 [Breakpoints] Window (Before Sequential Break Condition Setting)

The [Break] dialog box appears. To set sequential break conditions, select [Condition UBC] and display the [Condition UBC] page.

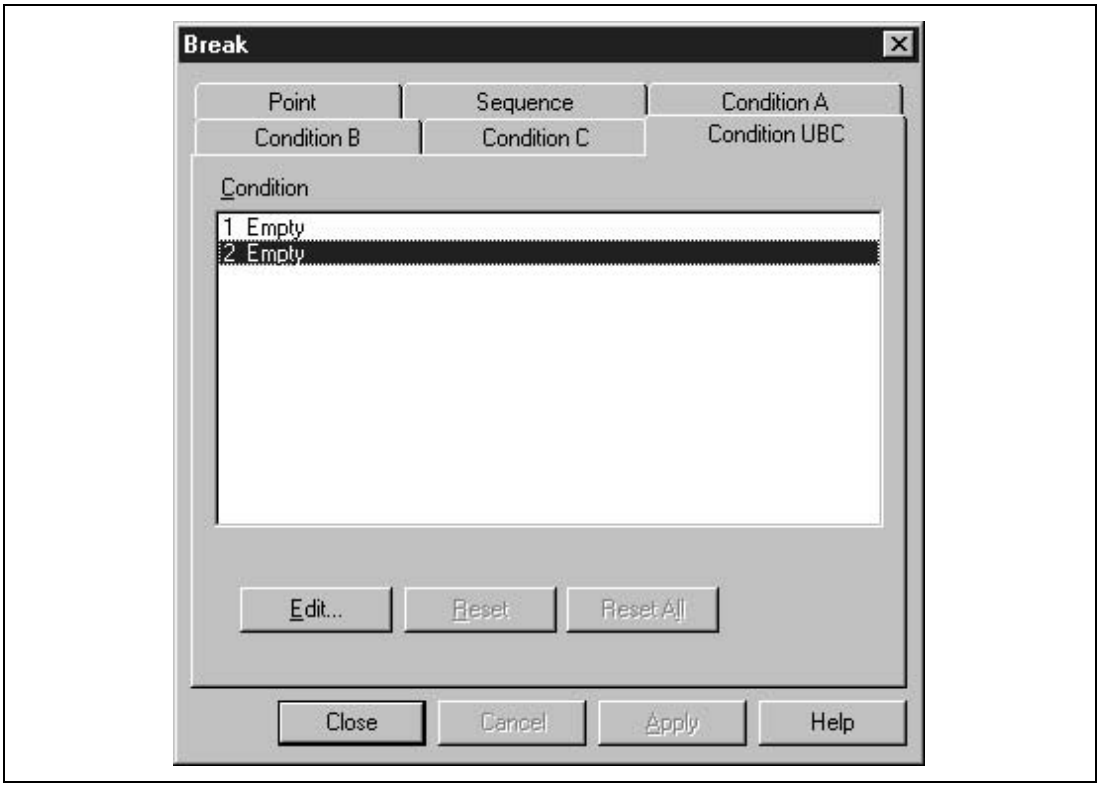

Figure 44 [Break] Dialog Box ([Break Condition UBC] Page)

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

- Highlight the second line in the [Condition] list box.
- 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'48*** as the value in the [Address] edit box.

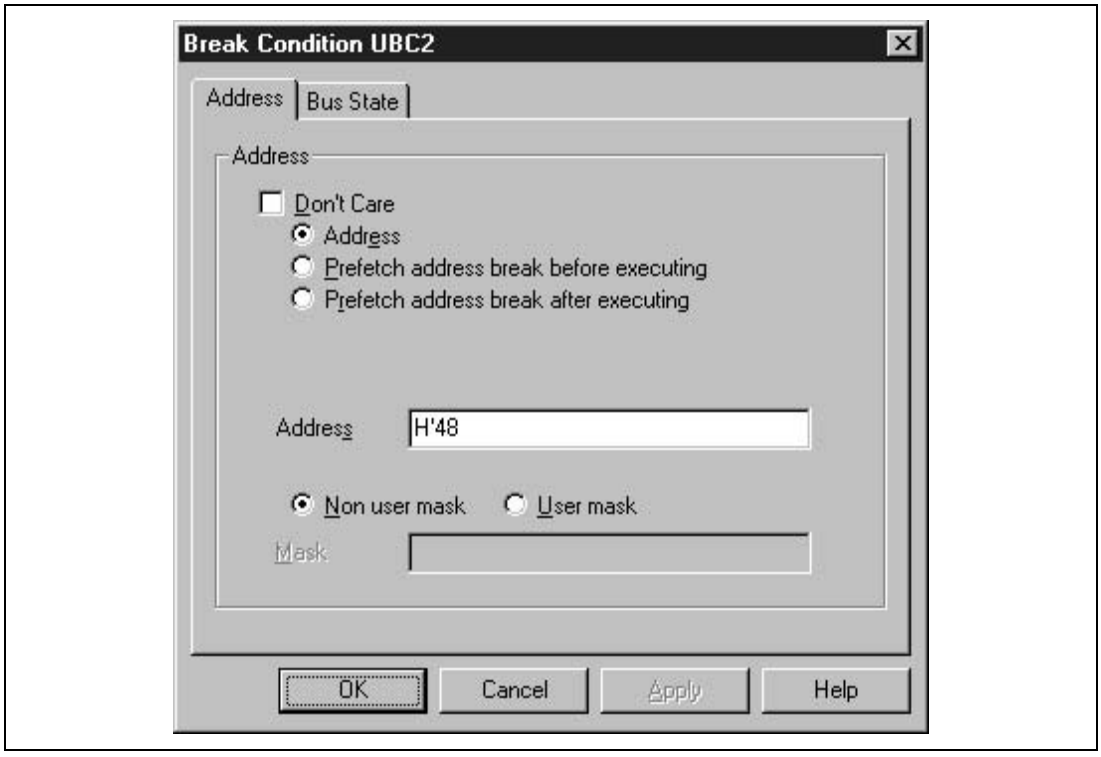

Figure 45 [Break Condition UBC2] Dialog Box (Condition 2 [Address] Page)

- Select [Bus State] to display the [Bus State] page.
- Select the [Read] radio button.

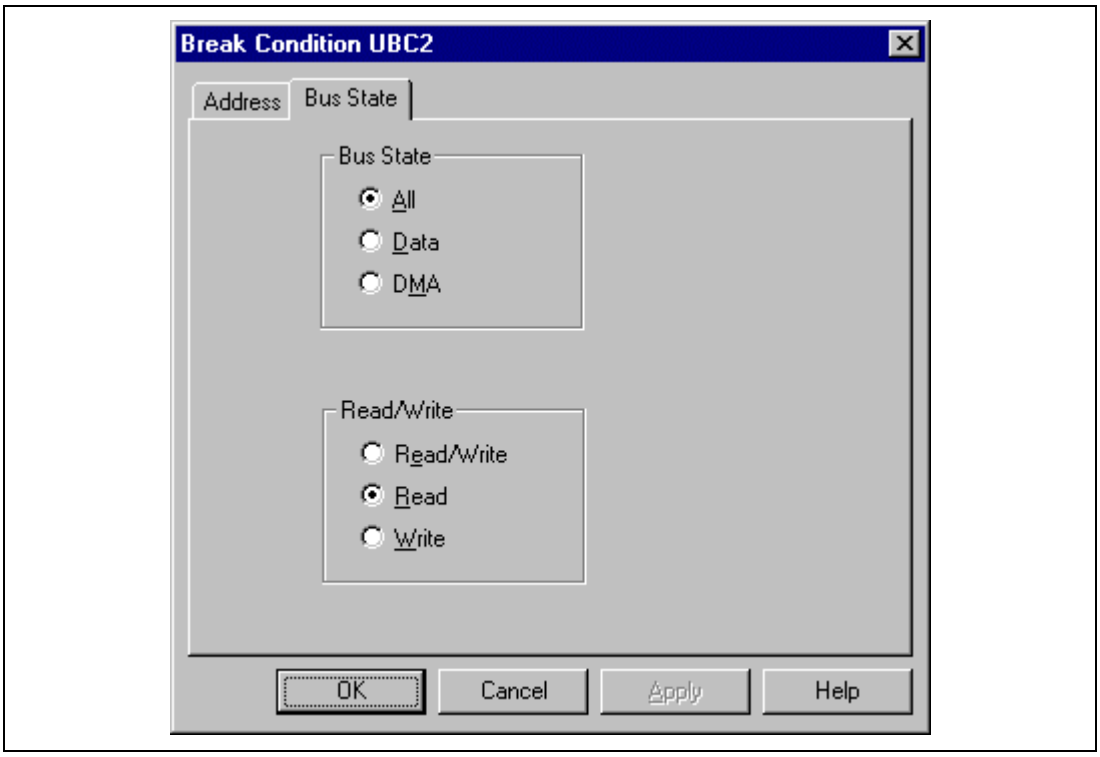

Figure 46 [Break Condition UBC2] Dialog Box (Condition 2 [Bus State] Page)

- 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`.

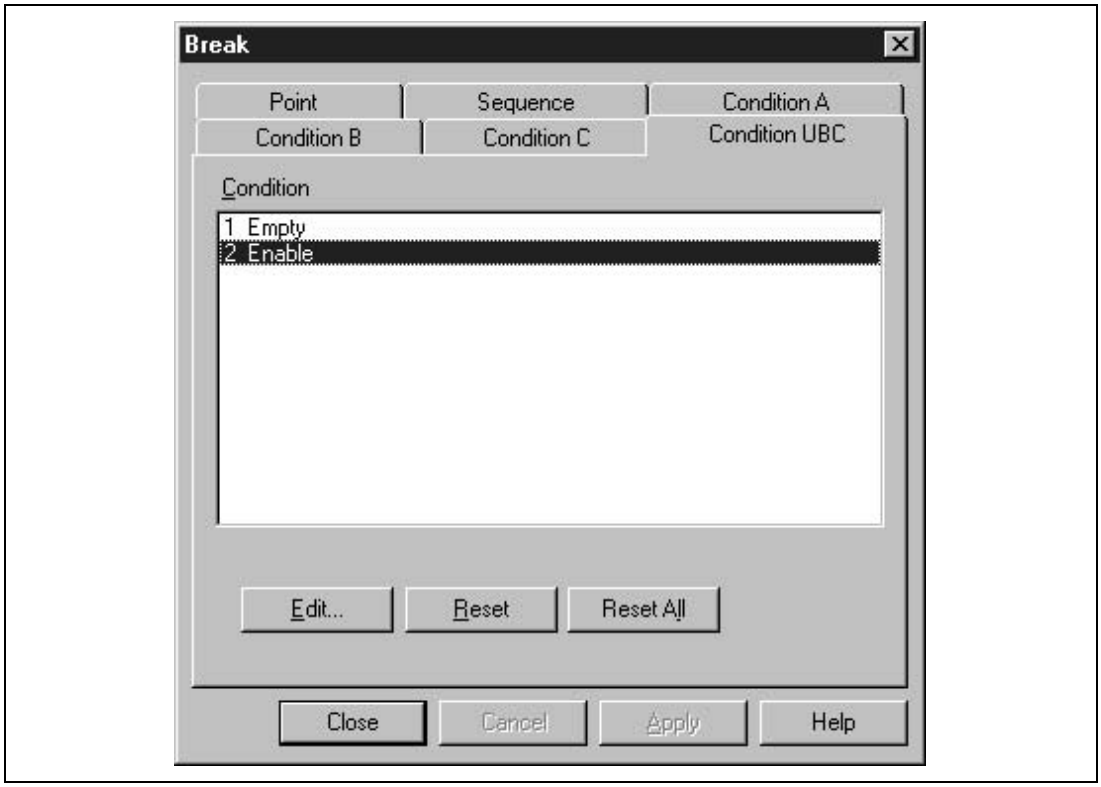

Figure 47 [Break] Dialog Box (After [Break Condition UBC2] Condition Setting)

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.

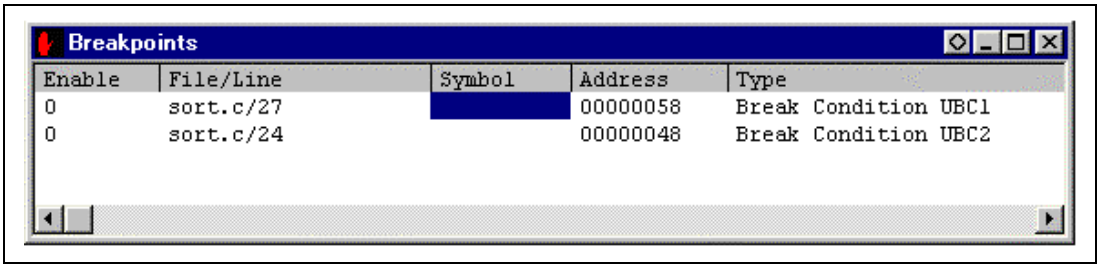

Figure 48 [Breakpoints] Window (After Sequential Break Condition Setting)

- Set the program counter and the stack pointer set in 7, Setting Registers, in the [Register] window, and click the [Go] button.

The program is executed up to the conditions of Break Condition UBC1 and comes to a halt.

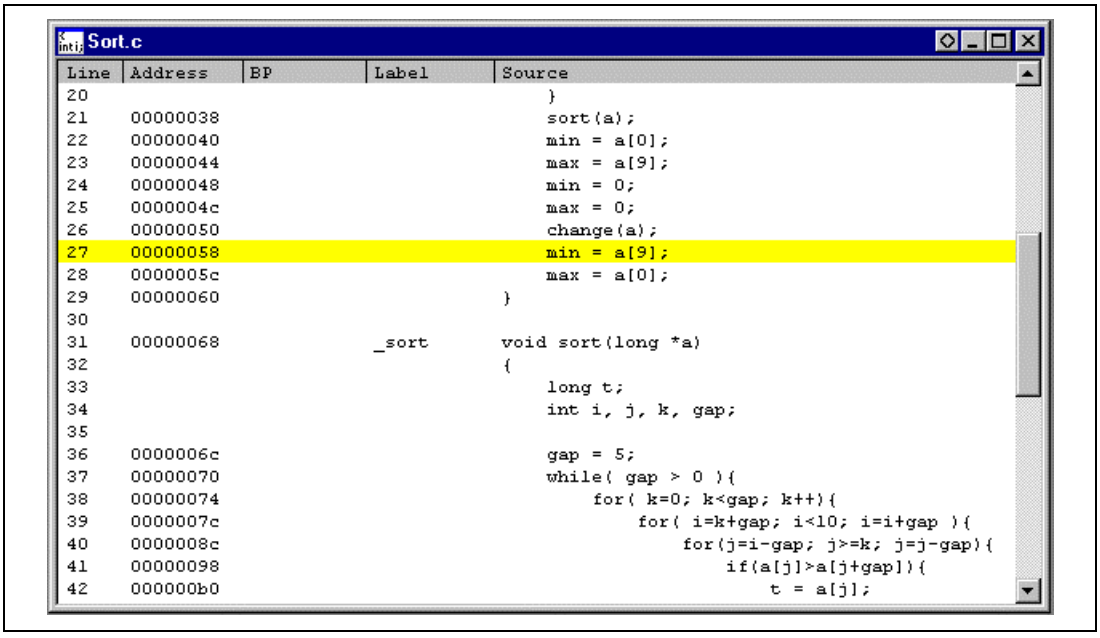

Figure 49 [Source] Window at Execution Halt (Sequential Break)

The contents of the [System Status] window are as follows:

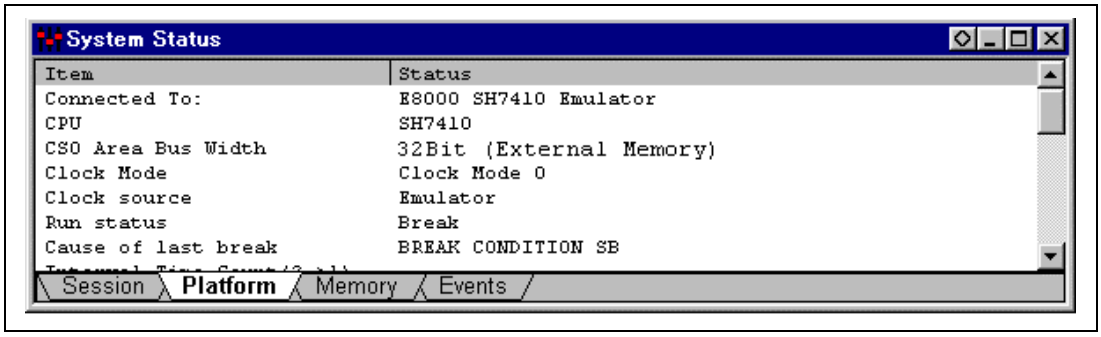

Figure 50 Contents of [System Status] Window (Sequential Break)

Note: When the SH7612 E8000 HDI is used, the information items displayed on the [Connected To:] and [CPU] lines are different from those of the SH7410 E8000 HDI.

16 Using the Trace Buffer

16.1 Displaying the Trace Buffer

The contents of the trace buffer can be displayed in the [Trace] window.

- Select [Trace] from the [View] menu to open the [Trace] window.
- If necessary, adjust the column width by dragging the column divider beside the label immediately below the title bar.

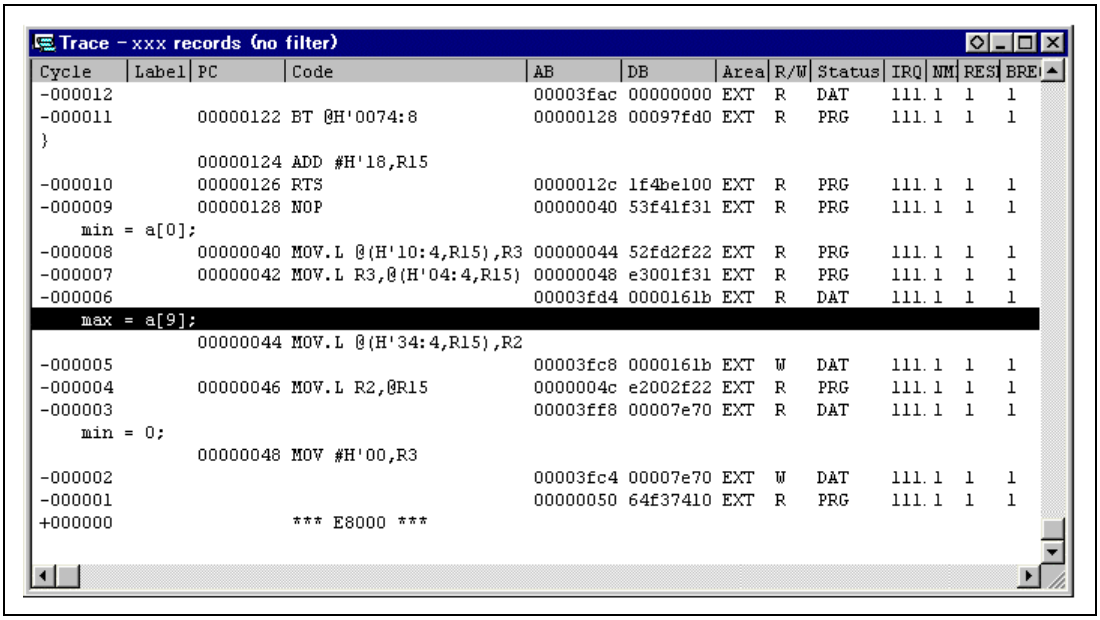

Figure 51 [Trace] Window (Free Trace Results)

Note: When the SH7612 E8000 HDI is used, [IRL] is displayed instead of [IRQ].

16.2 Setting the Trace Filter

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

Table 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 status condition (Bus & Area)	There are three bus state condition settings: Read/write condition: Searches for an item for which the CPU R/W signal level matches the specified conditions. Bus state condition: Searches for an item for which the conditions of CPU memory access cycle, DMA 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 levels of the NMI signal, external interrupt signals (IRQ0 to IRQ3), and 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.

Note: When the SH7612 E8000 HDI is used, the operation state during a refresh cycle can be checked. In addition, the external interrupt signal is displayed as [IRL], instead of [IRQ].

Click the right mouse button on the [Trace] window to display the popup menu.

For the trace search conditions, click the [Filter] button.

The [Trace Filter] dialog box then appears.

The filter conditions that limit the cycles to be displayed in the trace buffer can then be set.

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

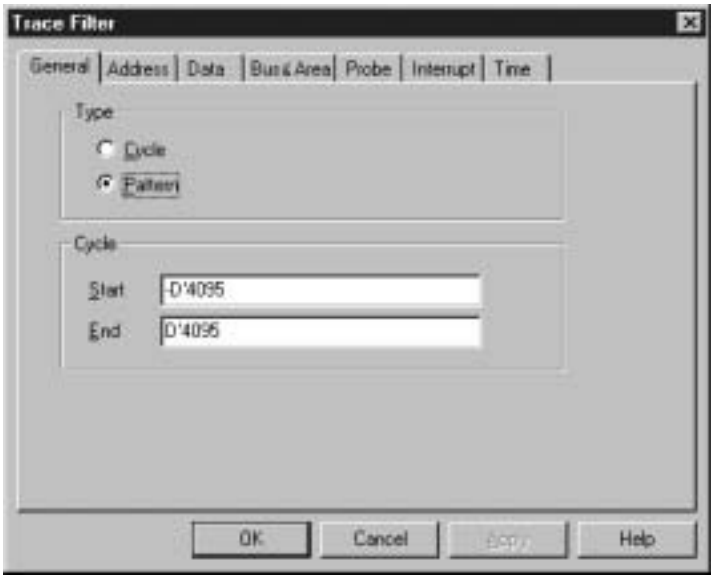

Figure 52 [General] Page ([Trace Filter] Dialog Box)

- Select [Address] to display the [Address] page.
- Clear the [Don't Care] check box in the [Address] page.
- Select [Address] and input address ***H'48*** as the value in the [Start] edit box.

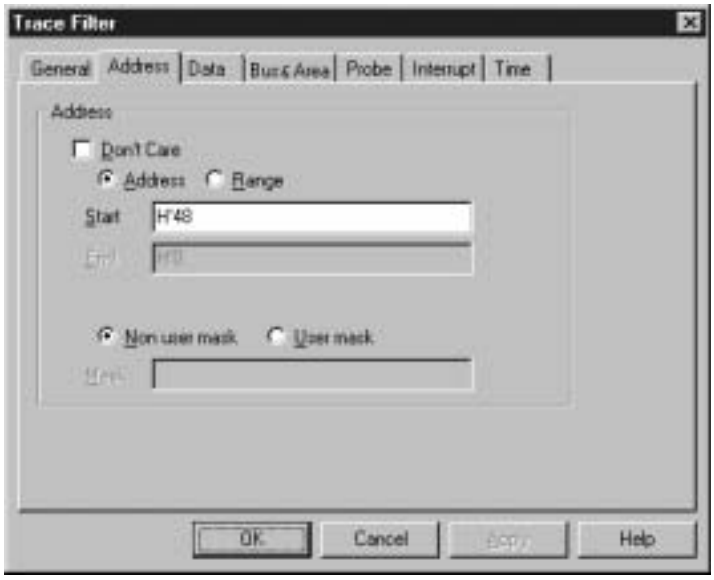

Figure 53 [Address] Page ([Trace Filter] Dialog Box)

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

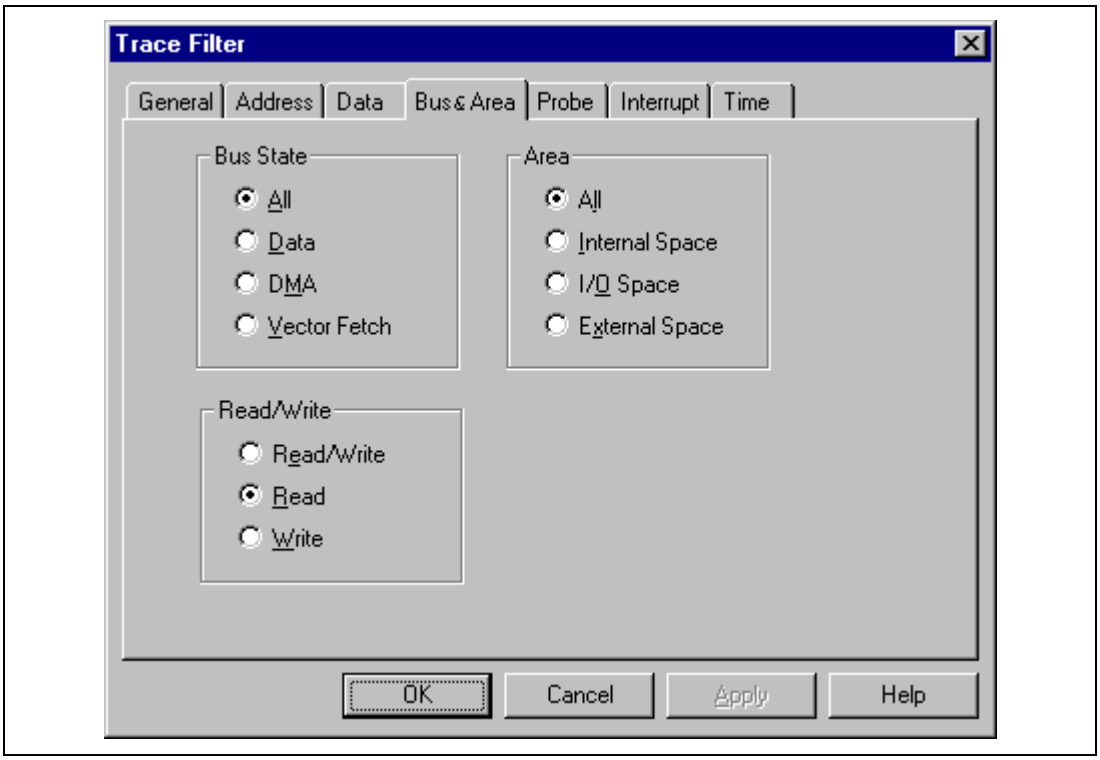

Figure 54 [Bus & Area] Page ([Trace Filter] Dialog Box)

- Click the [OK] button to save the trace filter.

Note: When the SH7612 E8000 HDI is used, the [Ref] radio button and [Cache Hit] button are displayed.

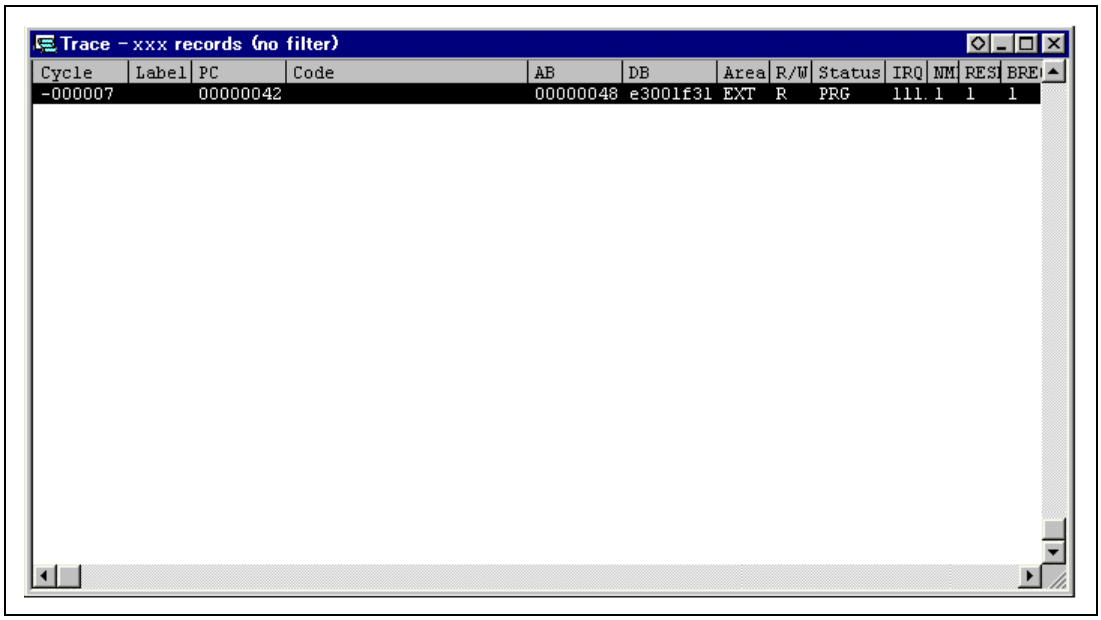

Figure 55 [Trace] Window (Trace Filter Results)

Note: When the SH7612 E8000 HDI is used, [IRL] is displayed instead of [IRQ].

17 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 10 Dialog Boxes for Setting Trace Acquisition Conditions

Function / Dialog Box	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 11 Main Trace Functions

Trace Function	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 end address of the specific subroutine with Trace Condition B,C.
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 status condition (DMA cycle, execution cycle, and vector fetch 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 status condition (DMA cycle, execution cycle, vector fetch cycle) System control signal (BREQ) External probe condition DELAY condition
Subroutine range trace	Performs trace acquisition only for places where a subroutine instruction and an operand that have been specified by Trace Condition B are accessed, and that the conditions are satisfied.

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 right mouse button on the [Trace] window to display the popup menu.

- Select [Acquisition...] to display the [Trace Acquisition] dialog box.

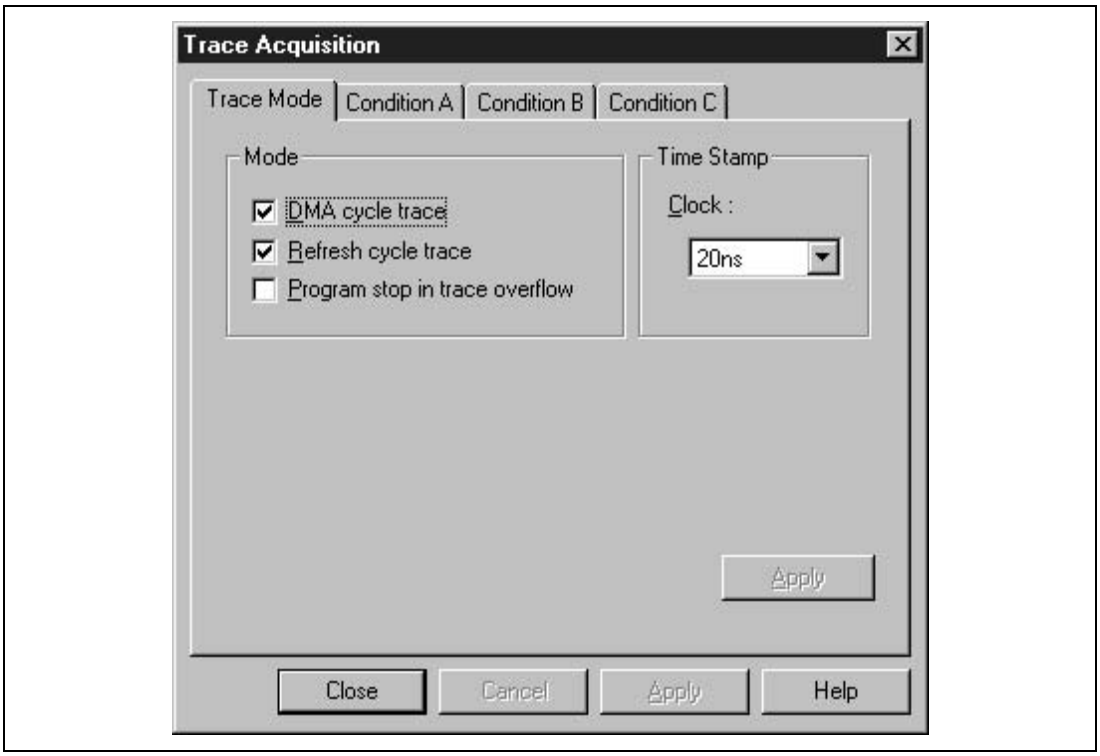

Figure 56 [Trace Acquisition] Dialog Box

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.

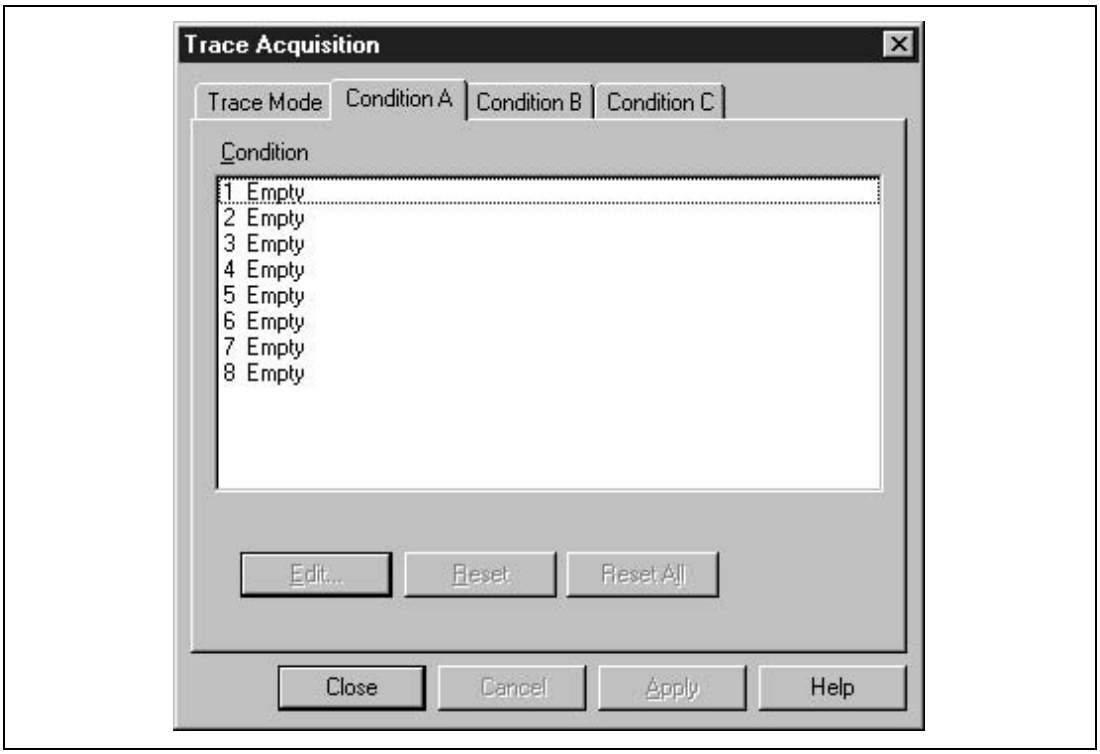

Figure 57 [Condition A] Page ([Trace Acquisition] Dialog Box)

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

- The [Trace Condition A1] dialog box is displayed.

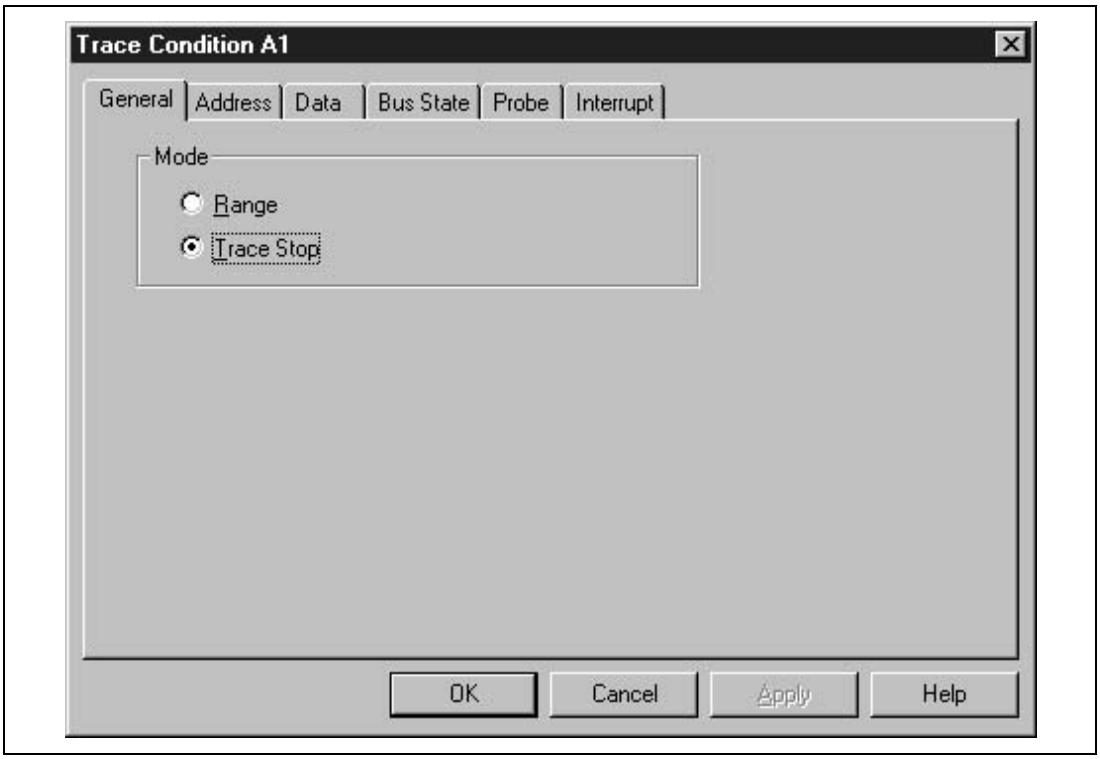

Figure 58 [General] Page ([Trace Condition A1] Dialog Box)

- 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'48*** as the value in the [Start] edit box.

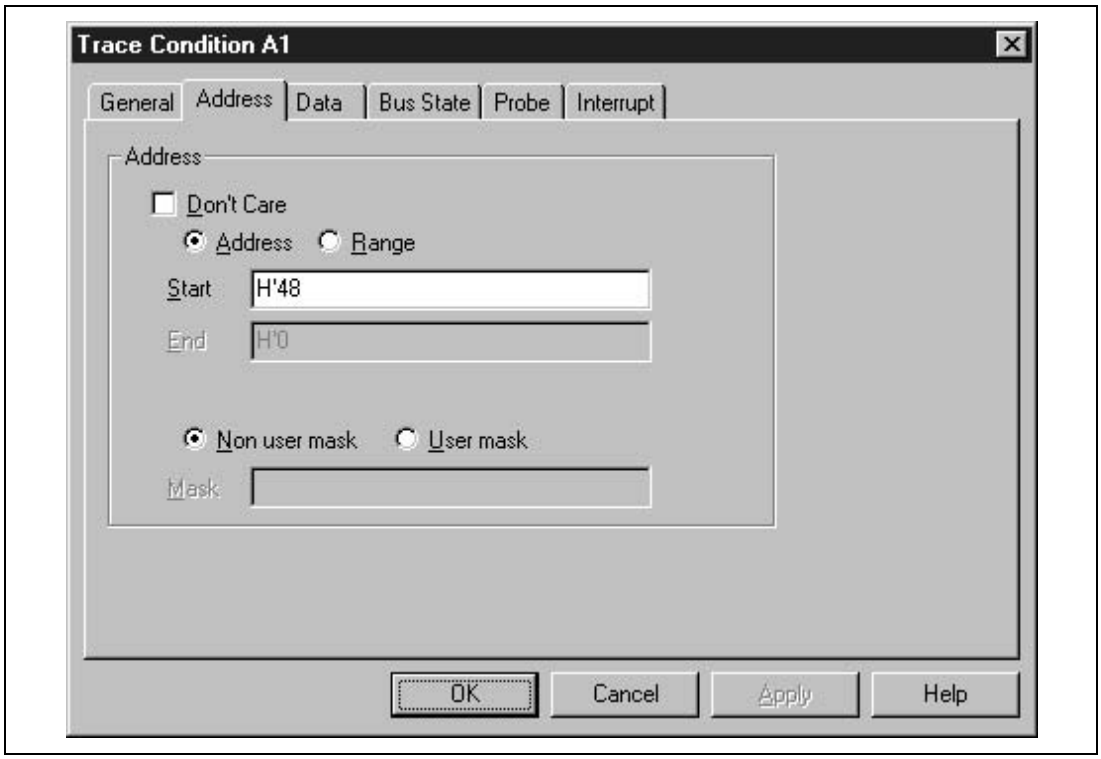

Figure 59 [Address] Page ([Trace Condition A1] Dialog Box)

- Select [Bus State] to display the [Bus State] page.
- Select [Read] radio button.

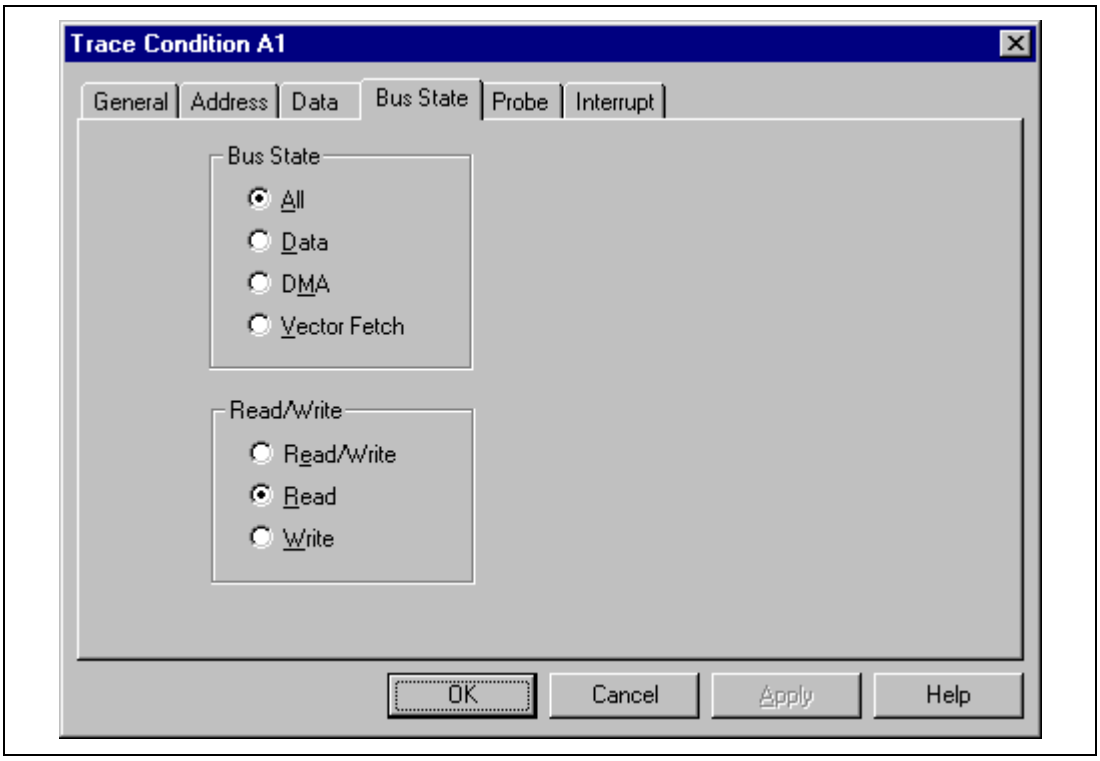

Figure 60 [Bus State] Page ([Trace Condition A1] Dialog Box)

- 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`.

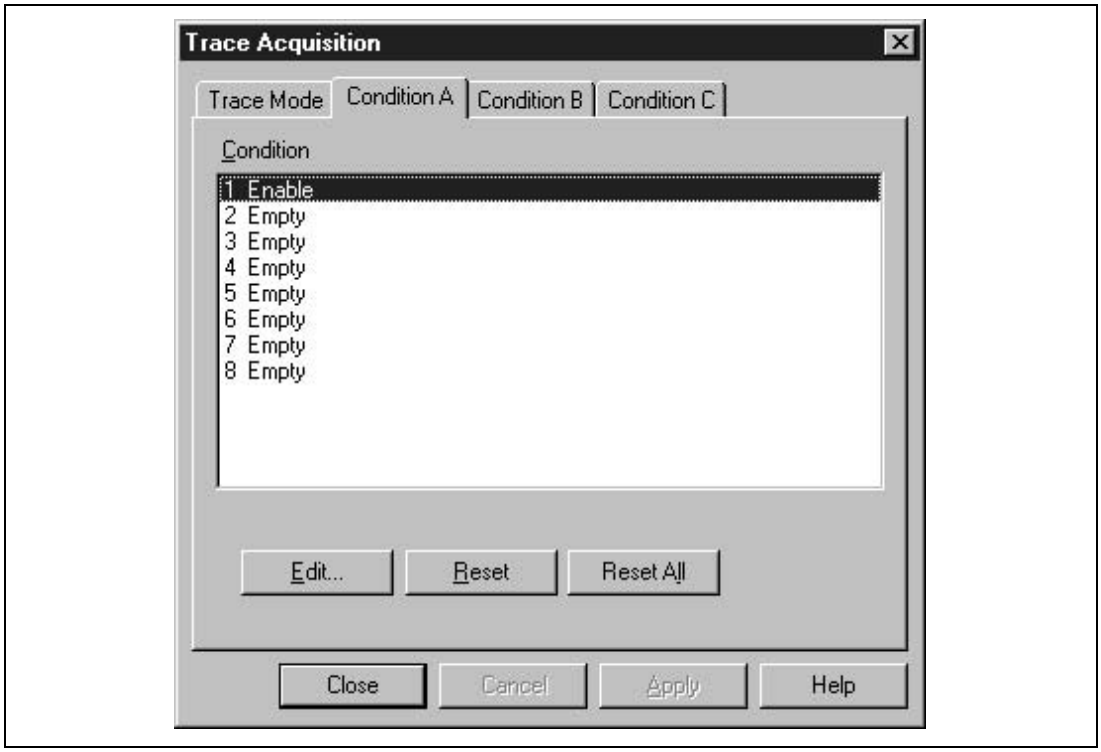

Figure 61 [Condition A] Page ([Trace Acquisition] Dialog Box)

This completes the setting of the Trace Condition A1 trace acquisition conditions. When the program is executed, trace acquisition will stop when address H'48 is accessed in a read cycle.

18 Saving 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.