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.

User's Manual

SH7729R CPU Board HS7729RSTC01H

User's Manual

Renesas Electronics
www.renesas.com

Rev.1.0 2001.08

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 CPU board.

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

Do not attempt to use the CPU board until you fully understand its mechanism.

CPU Board:

Throughout this document, the term "CPU board" shall be defined as the following products produced only by Hitachi, Ltd. excluding all subsidiary products.

- CPU board
- Serial cable
- AC power adapter
- AC power cable

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

Purpose of the CPU Board:

This CPU board is a software and hardware development tool for systems employing the Hitachi microcomputer SH7729R. Simple debugging functions such as debugging, performance evaluation, and development of the user system including the SH7729R are enabled by connecting the CPU board to a host computer. In addition, expansion boards can be installed in the slots; therefore, memory and I/O can be expanded. However, this CPU board must not be installed in user products to be used as part of the user products; it is limited to debugging and evaluation of user systems. This CPU board must only be used for the above purpose.

Limited Applications:

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

Improvement Policy:

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

Target User of the CPU Board:

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

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

LIMITED WARRANTY

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

DISCLAIMER

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

State Law:

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

The Warranty is Void in the Following Cases:

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

All Rights Reserved:

Other Important Things to Keep in Mind:

1. Circuitry and other examples described herein are meant merely to indicate the characteristics and performance of Hitachi's semiconductor products. Hitachi assumes no responsibility for any intellectual property claims or other problems that may result from applications based on the examples described herein.
2. No license is granted by implication or otherwise under any patents or other rights of any third party or Hitachi.

Figures:

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

Limited Anticipation of Danger:

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

SAFETY PAGE

READ FIRST

• READ this user's manual before using this CPU board.

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

Do not attempt to use the CPU board until you fully understand its mechanism.

DEFINITION OF SIGNAL WORDS

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

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

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

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

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

NOTE emphasizes essential information.

⚠ WARNING

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

1. **Always satisfy the power supply conditions which are described in the manual. Ensure that there are no short circuits between VCC and GND. Do not apply voltage that is outside the guaranteed range.**
2. **Always switch OFF the CPU board and the user system before connecting or disconnecting any CABLES, CONNECTORS, or JUMPERS.**
3. **When turning on the CPU board or the user system, take care that conductive material does not touch the CPU board or the user system.**
4. **Check that the pin numbers on the connectors of the CPU board and those on the user system are correctly aligned before connecting the CPU board and the user system.**

Preface

Thank you for purchasing the CPU board for Hitachi's SH7729R microcomputer.

The CPU board is an efficient development tool for software and hardware of systems based on Hitachi's SH7729R microcomputer.

This manual describes the SH7729R CPU board (model number HS7729RSTC01H). Hereafter, this product will be referred to as "the CPU board."

This manual explains the functions and method of operation of these CPU boards.

Section 1, Overview, describes the hardware system configuration and explains environment settings to enable board use.

Section 2, Preparation before Use, explains procedures for using the CPU boards, HDI installation, various connections, and the power supply specifications.

Section 3, Tutorial, introduces the major HDI features while demonstrating methods for loading and debugging a C language program.

Section 4, Descriptions of Windows, describes each of the windows used in the HDI.

Section 5, CPU Board Specifications, explains the specifications of the CPU boards, the memory map, interfaces with external equipment, and CPU board initialization.

Section 6, Notes and Troubleshooting, explains important information regarding use and gives suggestions for troubleshooting.

Section 7, Creation of User Interrupt Handlers, explains how to create an original interrupt handler routine.

Please read this manual completely in order to gain a thorough understanding of this product's functions and performance.

The text appearing in the various windows of the HDI may differ from those appearing in this manual depending on the language of the OS being used. The figures appearing in this manual are for the English version of Microsoft® Windows® 98.

Related Manuals

- SH7729R Hardware Manual
- SH3, SH3E, SH3-DSP Programming Manual
- SH Series Cross Assembler User's Manual
- H Series Linkage Editor User's Manual
- H Series Librarian User's Manual

- SuperH RISC Engine C/C++ Compiler User’s Manual
- SH Series Simulator/Debugger User’s Manual
- Hitachi Debugging Interface User's Manual (available in the CD-ROM supplied with this CPU board)
- Hitachi Embedded Workshop User's Manual

When connecting an E10A emulator to the CPU board, the following manual should also be read.

- SH7729R E10A Emulator User’s Manual

Note: Microsoft® and Windows® are registered trademarks of Microsoft Corporation in the United States and/or other countries.

Contents

Section 1 Overview

1.1 Features

The SH7729R CPU board (hereafter, referred to as the CPU board) supports the evaluation of the functions and performance of the Hitachi SH7729R microcomputer, and the development and evaluation of systems that incorporate the SH7729R.
The features of this CPU board are as follows:

- Supports user expansion boards

  Has an expansion bus connector for I/O of signals conforming to the SH7729R external bus specifications, to which expansion boards developed by the user to increase memory and I/O can be connected and evaluated.
- Supports the maximum operating frequency

  Allows evaluation at 200 MHz (internal operating frequency), which is the maximum frequency of the SH7729R.
- Interface

  For interfacing with IBM PC compatible as the host computer, a serial interface (that conforms to RS-232C) (one channel) is provided. The Hitachi Debugging Interface (HDI) is also provided as host interface software.
- Enables user-program evaluation

  Up to 15.5 Mbytes of a user program can be loaded by serial interface into the user memory and be evaluated.
- Support for PCMCIA

  PCMCIA interface hardware is provided as standard equipment, so that user programs can be employed in PCMCIA evaluations.

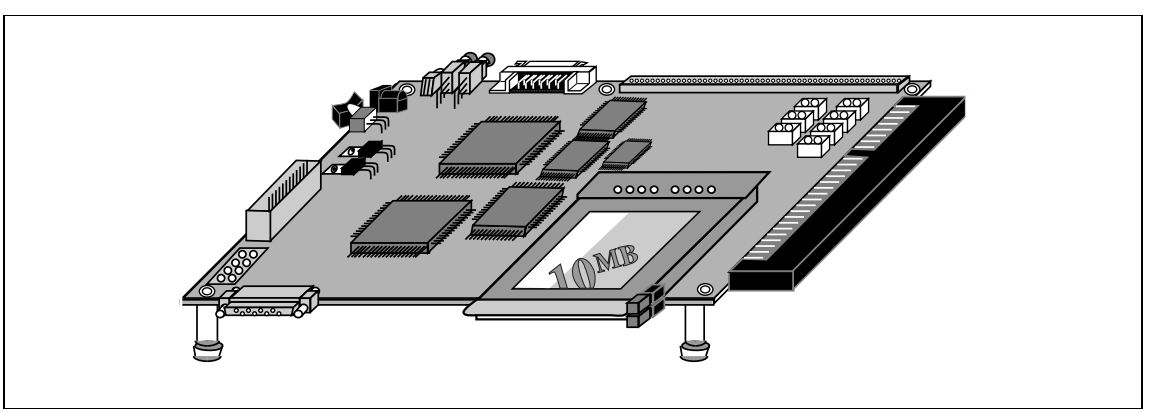


Figure 1.1 CPU Board, External View

Note: IBM PC is a registered trademark of International Business Machines Corporation in the United States.

1.2 System Configuration

The system configuration of the CPU board is shown in figure 1.2.

The following items are required to use the CPU board.

- IBM PC compatible machine: One for the monitor command input and output.
- One serial interface cable: Use the provided cable.
- One AC power supply adapter: Use the provided adapter.
- One AC power supply cable: Use the provided power cable.

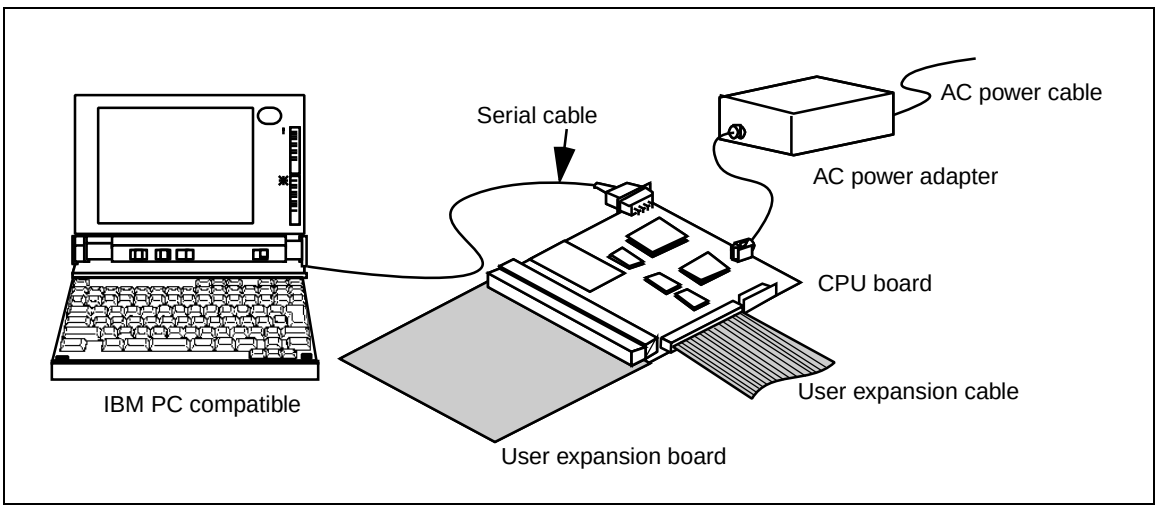


Figure 1.2 CPU Board System Configuration

1.3 Warnings

CAUTION

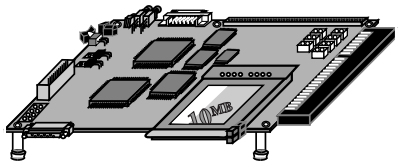
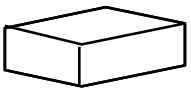
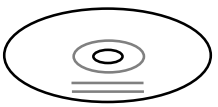
READ the following warnings before using the CPU board. Incorrect operation will damage the user system and the CPU board. The USER PROGRAM will be LOST.

1. Check all components against the component list after unpacking the CPU board.
2. Never place heavy objects on the CPU board.
3. Protect the CPU board from excessive impacts and stresses. For details, refer to section 1.6, Environmental Conditions.
4. Do not connect any cable or connector other than specified ones to the CPU board.
5. When moving the host computer or user expansion board, take care not to vibrate or damage the CPU board.
6. After connecting the cable, check that it is connected correctly. For details, refer to section 2, Preparation before Use.
7. Supply power to the connected equipment after connecting all cables. Cables must not be connected or removed while the power is on.

1.4 Components

Table 1.1 lists the components of the CPU board. Check all components after unpacking.

Table 1.1 CPU Board Component List

Item	View	Quantity	Remarks
CPU board		1	One printed circuit board
AC power supply adapter		1	
AC power supply cable		1	
Serial communication cable		1	
Jumper pin		1	
CD-R *		1	One CD-R; Model number HS7729RSTC01SR
Notes on Usage of SH7729R CPU Board		2	One Japanese version and one English version; Japanese: HS7729RSTC01HJ-P, English: HS7729RSTC01HE-P

Note: Refer to section 1.5, CD-R Contents.

1.5 CD-R Contents

The supplied CD-R includes software and user's manuals for the SH7729R CPU board as listed in table 1.2.

Table 1.2 CD-R Contents

Directory	File Name	Contents	Remarks
\setup	setup.exe	HDI installer	
\Manuals\Japanese	HS7729RSTC01HJ.pdf	SH7729R CPU Board User's Manual	PDF document in Japanese Type No.: HS7729RSTC01HJ
\Manuals\Japanese	HS6400DIIW5SJ.pdf	Hitachi Debugging Interface User's Manual	PDF document in Japanese Type No.: HS6400DIIW5SJ
\Manuals\English	HS7729RSTC01HE.pdf	SH7729R CPU Board User's Manual	PDF document in English Type No.: HS7729RSTC01HE
\Manuals\English	HS6400DIIW5SE.pdf	Hitachi Debugging Interface User's Manual	PDF document in English Type No.: HS6400DIIW5SE
\Pdf_read\Japanese	Ar40jpn.exe	Acrobat® Reader installer	Japanese version
\Pdf_read\Englsih	Ar40eng.exe	Acrobat® Reader installer	English version

Note: To read a PDF document, use the Acrobat® Reader.

1.6 Environmental Conditions

CAUTION

Observe the conditions listed in tables 1.3 and 1.4 when using the CPU board. Failure to do so will damage the user system and the CPU board. The USER PROGRAM will be LOST.

Table 1.3 Environmental Conditions

Item	Specifications
Temperature	Operating: +10°C to +35°C Storage: –10°C to +50°C
Humidity	Operating: 35% RH to 80% RH, no condensation Storage: 35% RH to 80% RH, no condensation
Vibration	Operating: 2.45 m/s^2 max. Storage: 4.9 m/s^2 max. Transportation: 14.7 m/s^2 max.
Ambient gases	There must be no corrosive gases present

Table 1.4 Operating Environments

Item	Description
Host computer	Built-in Pentium or higher-performance CPU (200 MHz or higher recommended); IBM PC or compatible machine.
OS	Windows® 95, Windows® 98, or Windows NT®
Minimum memory capacity	32 Mbytes or more (double of the load module size recommended)
Hard-disk capacity	Installation disk capacity: 5 Mbytes or more. (Prepare an area at least double the memory capacity (four-times or more recommended) as the swap area.)
CD-ROM drive	Required to install the HDI.
Pointing device such as mouse	Connectable to the host computer; compatible with Windows® 95, Windows® 98, and Windows NT®.
Power supply	Input: 100 to 240 VAC, 50/60 Hz, 0.9 A max.
(for AC power supply adapter)	Output: +5.0 VDC, 6.0 A max.

2.1 CPU Board Preparation

⚠ WARNING

READ the reference sections shaded in figure 2.1 before using the CPU board product. Incorrect operation will damage the user system and the CPU board. The USER PROGRAM will be LOST.

Unpack the CPU board and prepare it for use as shown in figure 2.1:

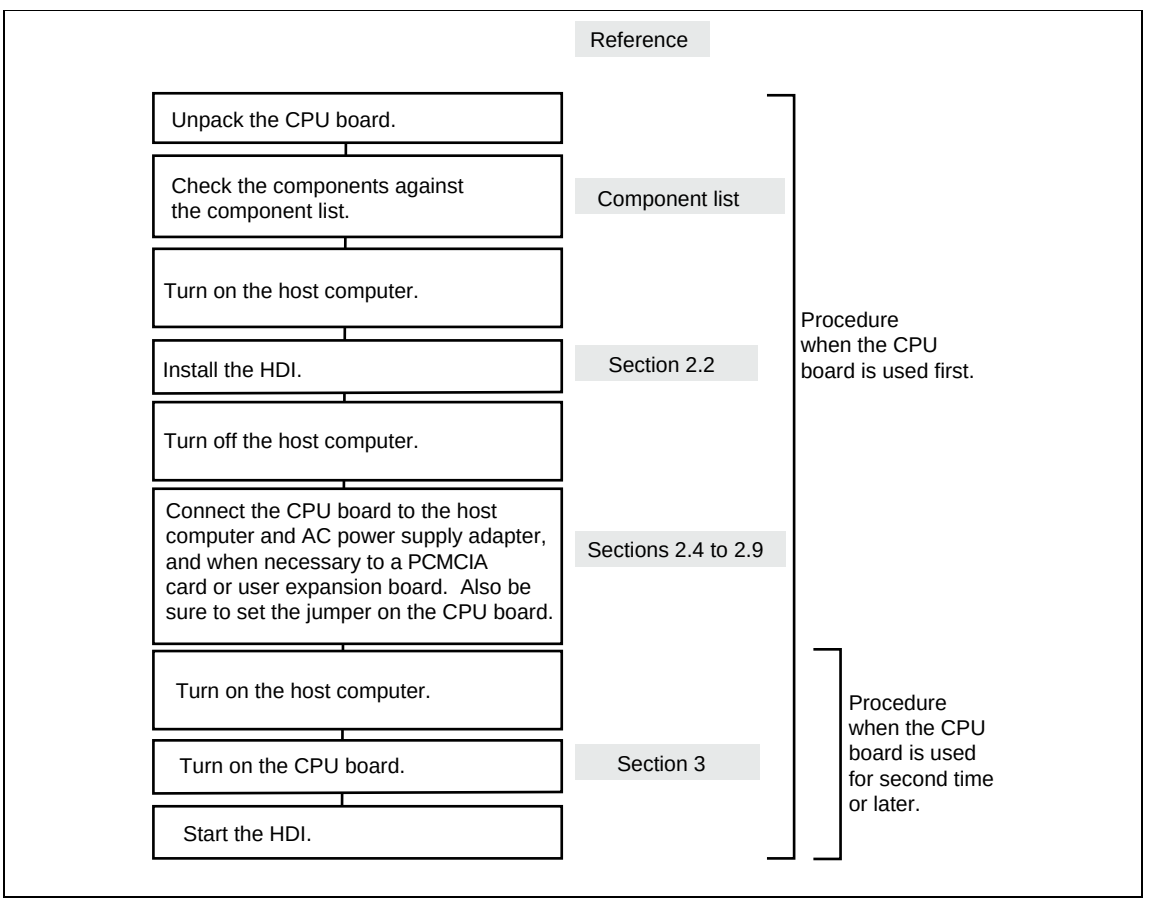


Figure 2.1 CPU Board Preparation Flow Chart

2.2 HDI Installation

An example of installing the HDI on an IBM PC compatible machine is described in this section.

Start [setup.EXE] in the \SETUP directory of the CD-R. If any other application is running, close it before starting the HDI installer.

Figure 2.2 [setup.EXE] Icon

This runs the HDI installer. A dialog box will first prompt you to select a language for the installation process. Select a language then continue according to the instructions displayed by the installer.

Note: Under Windows NT® 4.0, install the HDI in the administrator mode.

HDI Installation Directory: The default directory for installing the HDI depends on whether the Hitachi Embedded Workshop (HEW) has been installed in the host computer, as shown in table 2.1

Table 2.1 Default Installation Directory

HEW Program	Default Installation Directory
HEW has not been installed	C:\HDI5_CB\7729R
HEW has been installed (in this example, in C:\HEW)	C:\HEW\HDI5\CB\7729R

Backup File: If another version of HDI has already been installed, a message "The HDI.INI file has already existed. Can it be overwritten?" will be displayed. Clicking [Yes] will make a backup of the existing file in the Backup directory of the installation directory.

When installation is complete, [HDI for SH7729R CPU board] can be selected from the start menu.

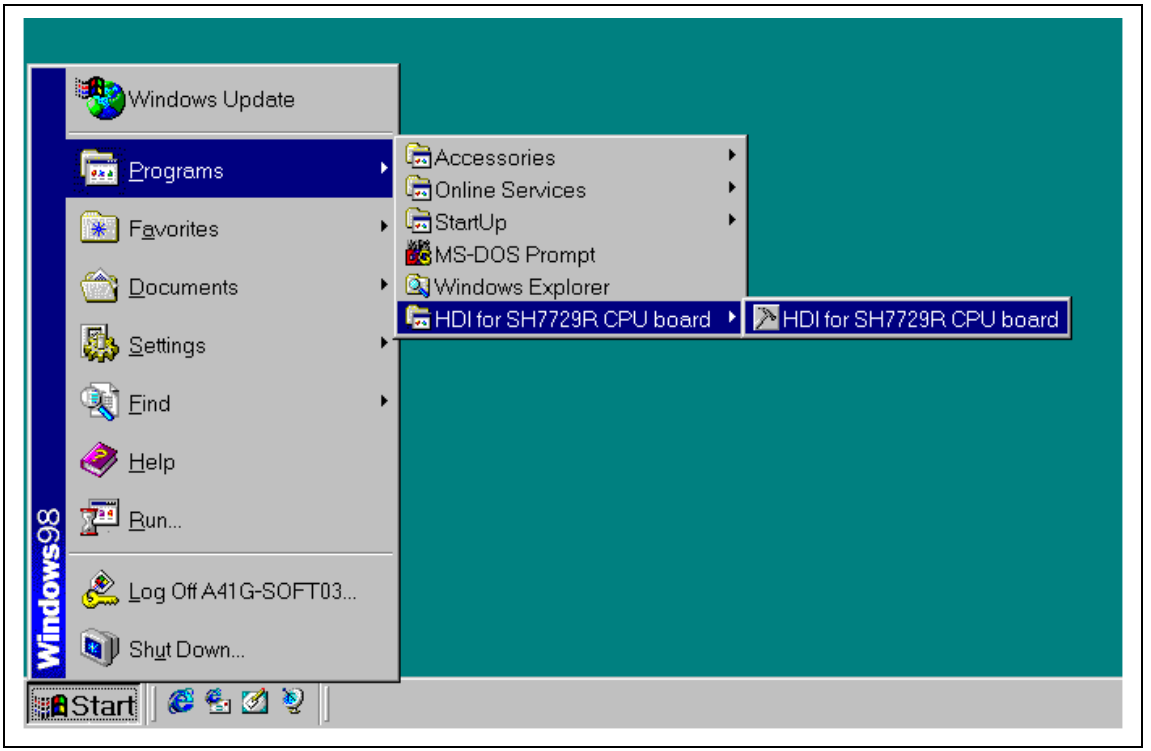


Figure 2.3 Start Menu

2.3 HDI Uninstallation

Uninstall the HDI for the SH7729R CPU board as follows:

1. Select [Settings] from the Start menu, then select [Control Panel].
2. Select [Add/Remove Programs].
3. Select [HDI for SH7729R CPU board] from the application list, then click [Add/Remove].
4. A confirmation message will be displayed, and the uninstallation procedure will start.

2.4 Connecting Cables

This section shows how to connect interface cables to the CPU board.

⚠ WARNING

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

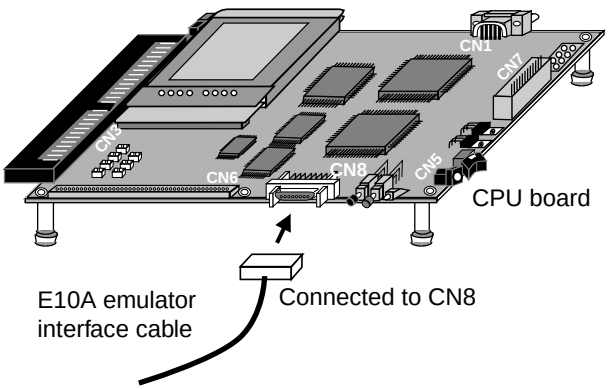


Figure 2.4 E10A Emulator Interface Cable Connection

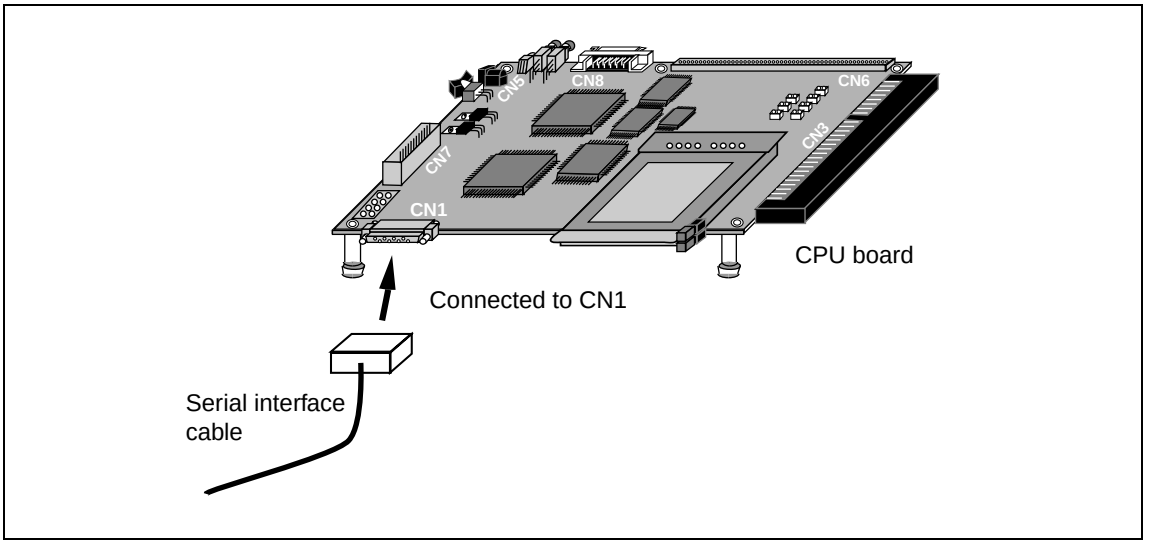


Figure 2.5 Serial Interface Cable Connection

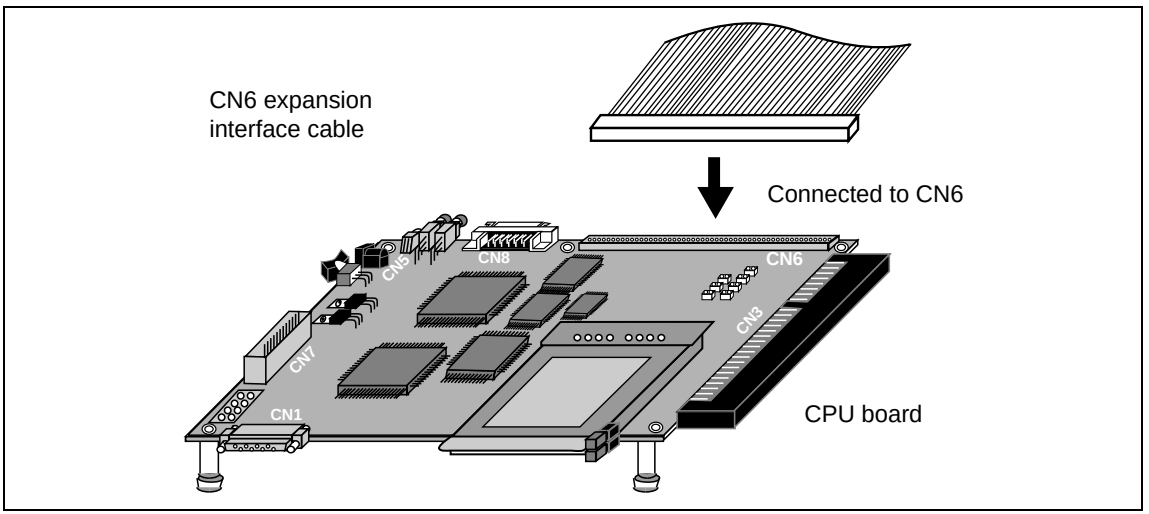


Figure 2.6 CN6 Expansion Interface Cable Connection

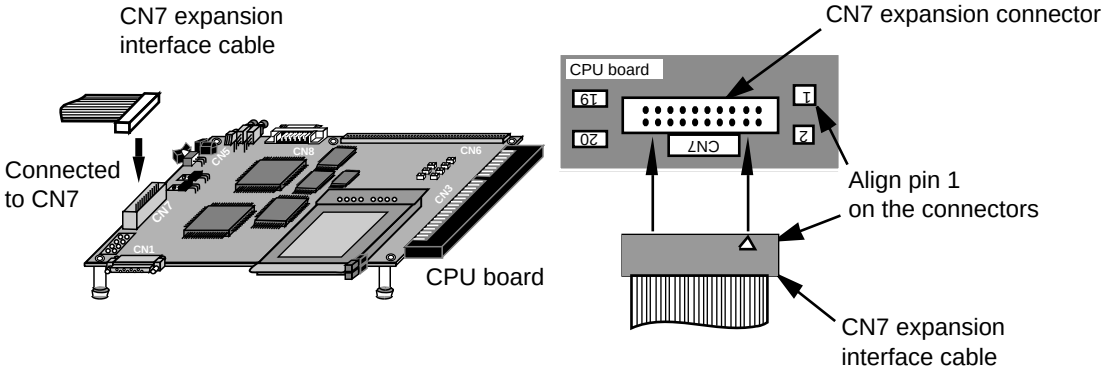


Figure 2.7 CN7 Expansion Interface Cable Connection

⚠ WARNING

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

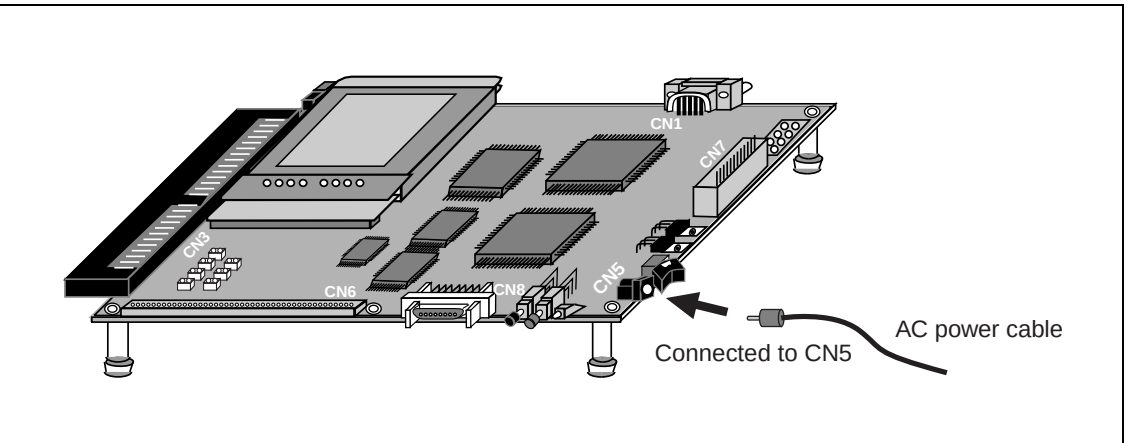


Figure 2.8 AC Power Supply Cable Connection

2.5 Connecting the User Expansion Board

Figure 2.9 shows how to connect the user expansion board.

⚠ WARNING

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

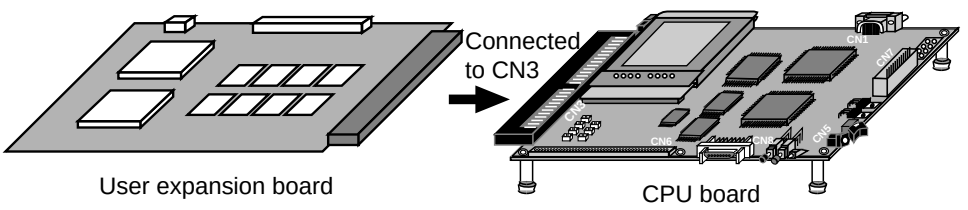


Figure 2.9 User Expansion Board Connection

2.6 Connecting the PCMCIA Card

Figure 2.10 shows how to connect the PCMCIA card.

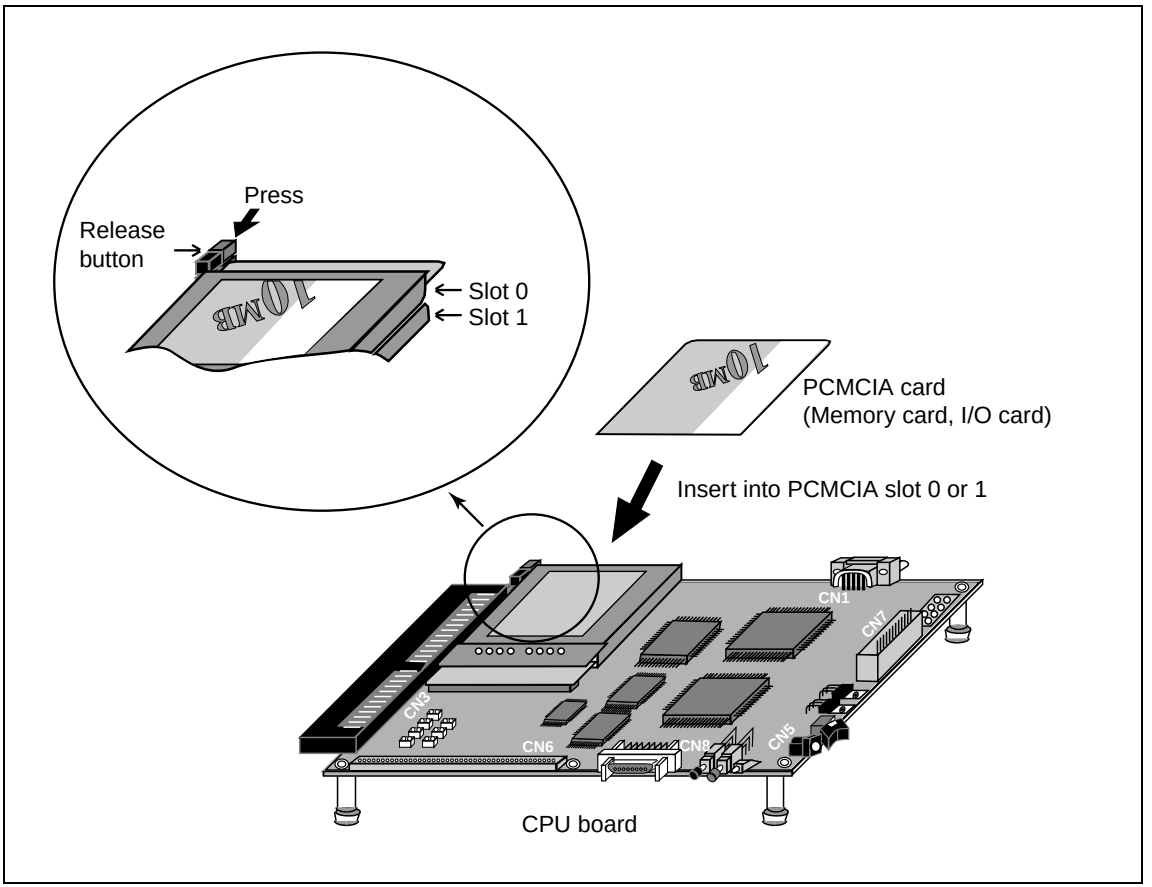


Figure 2.10 PCMCIA Card Connection

To use a PCMCIA card (memory card, I/O card), insert the PCMCIA card into slot 0 or slot 1. To remove the PCMCIA card, press the release button in the direction of the arrow.

2.7 Switches

Table 2.2 lists the switches used in the CPU board and figure 2.11 shows where the three switches are located (on the CPU board).

Table 2.2 Switch Specifications

Switch	Symbol	Type	Function
Manual reset switch	SW1	Push button (Red)	Forcibly initializes the system. Use this switch when the system does not operate correctly, for example, when the user program goes out of control.
Abort switch	SW2	Push button (Black)	Forcibly terminates command execution. Aborts user program execution and returns the system to firmware command input wait state.
Power supply switch	SW3	Rocker switch	Turns on and off the 5 V power supplied to the CPU board.

Note If the power supply is turned on while pressing the manual reset switch, the CPU board and HDI will not be started. Please do not operate the manual reset switch while turning on the power.

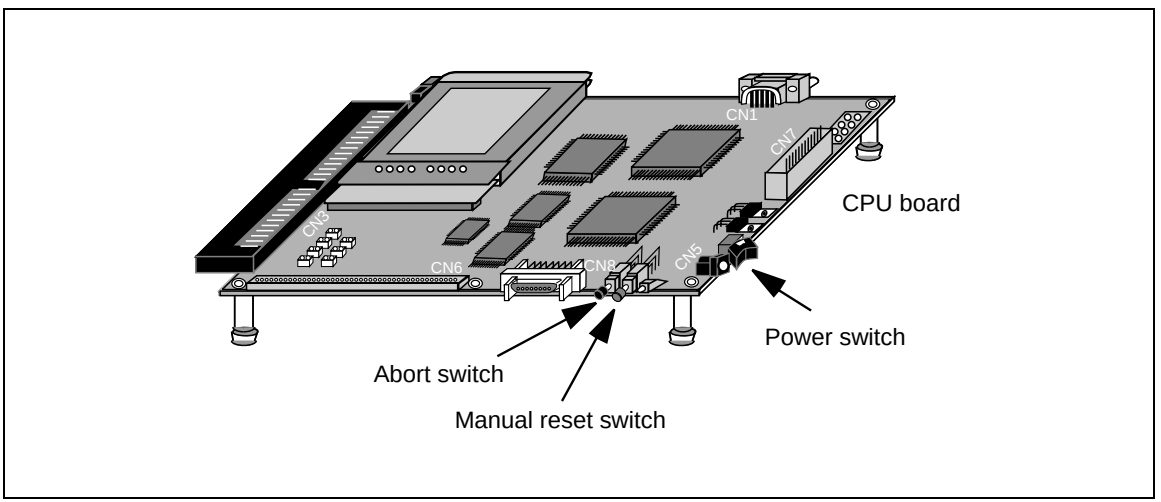


Figure 2.11 Switch Location

2.8 Jumpers

⚠ WARNING

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

Table 2.3 lists the jumpers on the CPU board and figure 2.12 shows how to insert a jumper pin.

Table 2.3 Jumper Settings

Symbol	Function	Setting		Description	
J1	Selects endian	Closed		Big endian	(default at shipment)
		Open		Little endian	
J2	For test	Open		Be sure to open this jumper.*2	
J3	Selects timer clock	1–2 closed		External clock TCLK (CN3-158 pin)	
		2–3 closed		On-board clock (1.8432 MHz) (default at shipment)	
J4	Not mounted				
J5	Not mounted				
J6	Not mounted				
J7	Selects host computer interface baud rate	Closed		115200 bit/s*3	(default at shipment)
		Open		57600 bit/s	
J8	Not mounted				
J9	Selects bus frequency*1	J9	J10		
J10		Closed	Closed	66.7 MHz	(default at shipment)
		Closed	Open	33.3 MHz	
		Open	Closed	Forbidden	
		Open	Open	Forbidden	
J11	Enables E10A emulator	Closed		For use of the emulator	
		Open		Emulator not used (default at shipment)	
J12	Not mounted				
J13	Not mounted				

Notes: 1. The same frequency is supplied to the CPU, bus clock, and expansion connectors, but the CPU internal operating clock always operates at 200 MHz.
2. If jumper J2 is closed, the monitor program will be corrupted. Be sure to open J2.
3. When the bus frequency is set to 33.3 MHz, the CPU board cannot be interfaced with the host computer at 115200 bit/s; set the baud rate to 57600 bit/s.

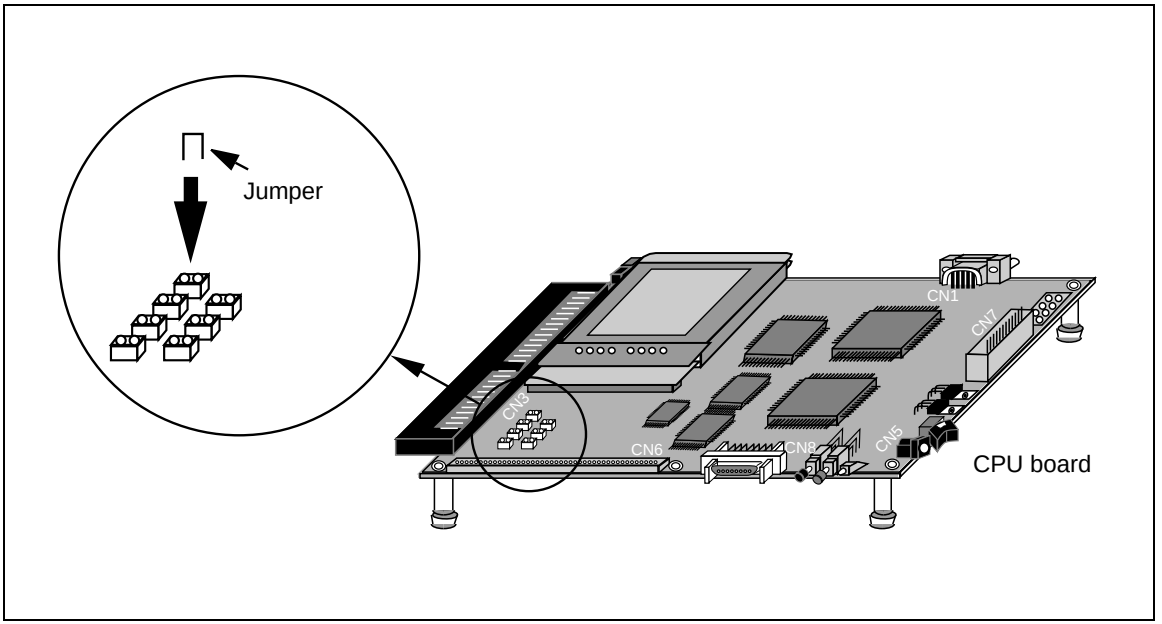


Figure 2.12 Jumper Insertion

2.9 Power Supply

2.9.1 Power-Supply Specifications

Figure 2.13 shows the power-supply specifications.

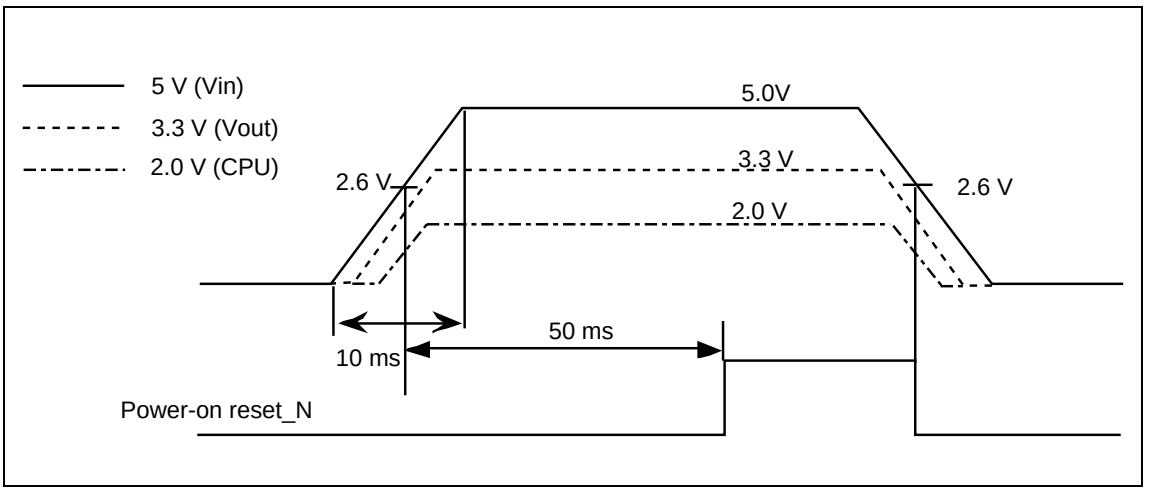


Figure 2.13 Power-Supply Specifications

2.9.2 Connecting the Power Supply Cable

Power should always be supplied to the CPU board using the provided AC power supply adapter and AC power supply cable. The method of connection is shown in figure 2.14.

⚠ WARNING

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

1. Always use the provided AC power supply adapter.

2. Always switch OFF the CPU board and the user system before connecting or disconnecting any CABLES, CONNECTORS, or JUMPERS.

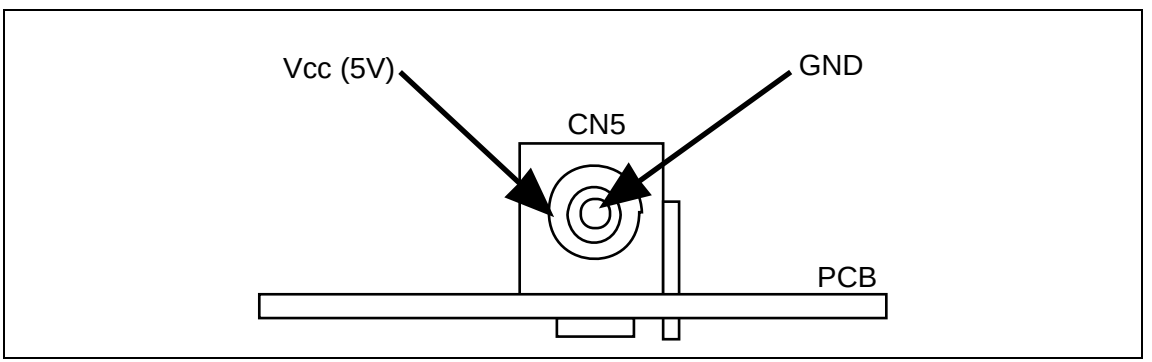


Figure 2.14 Front View of Power Supply Connector

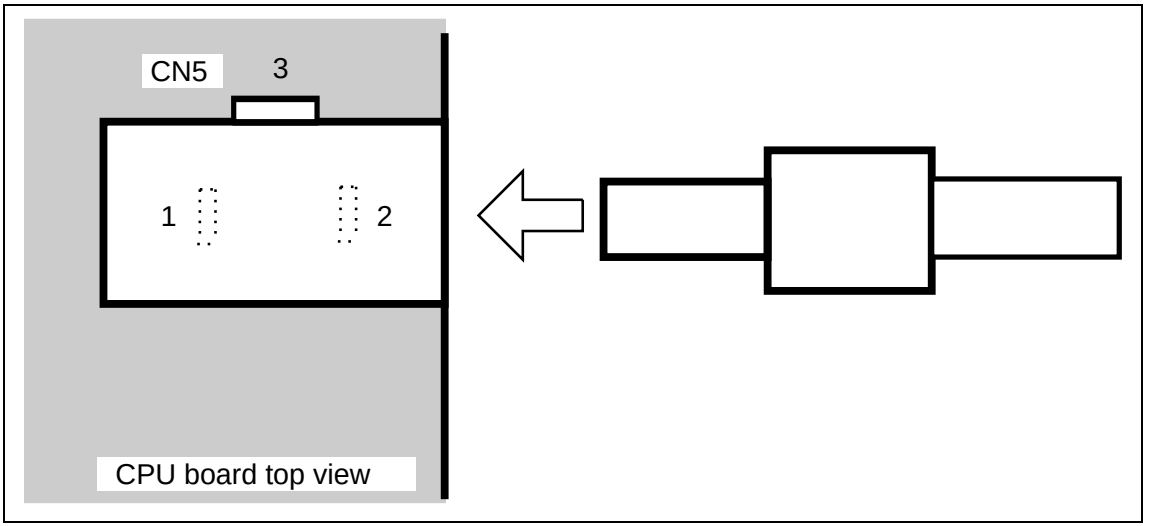


Figure 2.15 AC Power Supply Cable Connection

Section 3 Tutorial

3.1 Introduction

The following describes the main functions of the HDI by using a tutorial program.

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

The tutorial program is included in the `sort.c` file. The compiled load module is provided in the SYSROF format and is included in the `sort.abs` file.

The tutorial program is automatically installed when the HDI is installed.

Table 3.1 lists the tutorial program configuration.

Table 3.1 Tutorial Program Configuration

Item	Contents
Tutorial file (load module)	(install directory)\tutorial\sort.abs
Tutorial file (source file)	(install directory)\tutorial\sort.c

For the operating environment, use the RAM area starting from address H'AC000000. The MMU function is not used.

Notes: 1. `sort.abs` operates in big endian. `sort.abs` must be recompiled to operate in little endian.

2. The work space for this tutorial program was created using Version 1.1 (Release 4) of Hitachi Embedded Workshop (HEW).
   Hitachi SH C/C++ compiler version 5.1B
   Hitachi SH IM OptLinker version 1.1B

3.2 Running the HDI

To run the HDI, select the [HDI for SH7729R CPU board] from the [Start] menu.

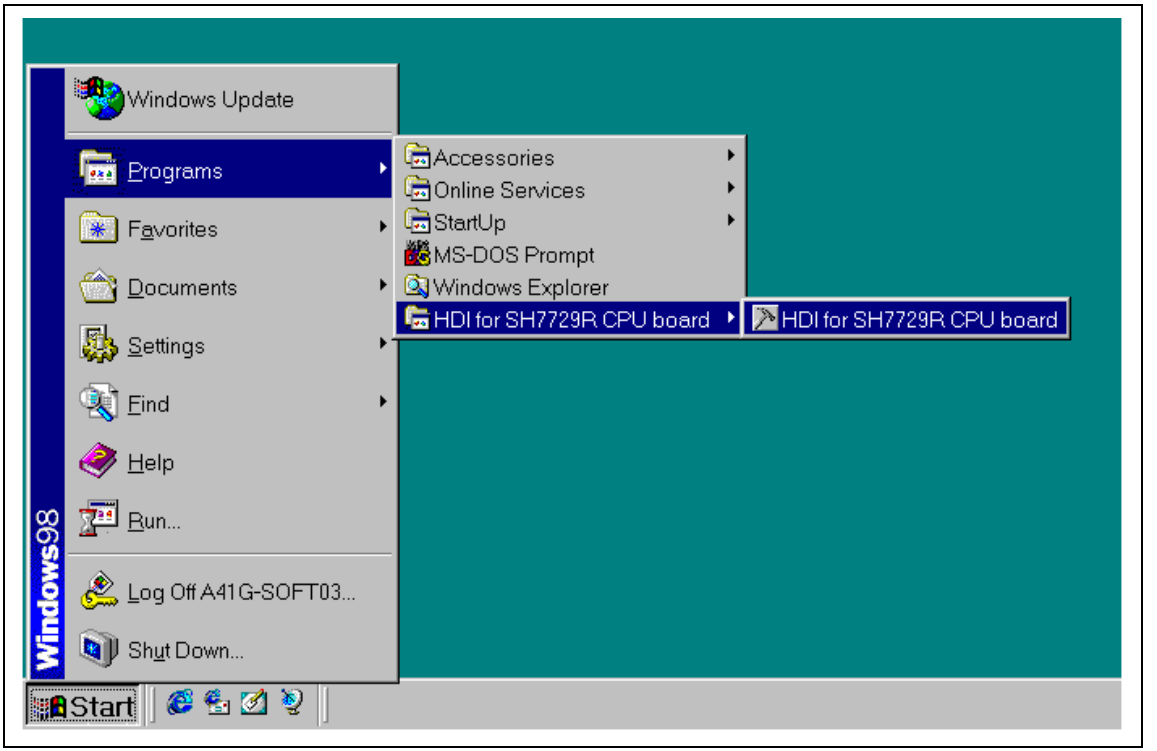


Figure 3.1 [Start] Menu

The [HDI] window will open, then the [Select Session] dialog box will appear. Check that the setting shown in figure 3.2 is complete, and click the [OK] button.

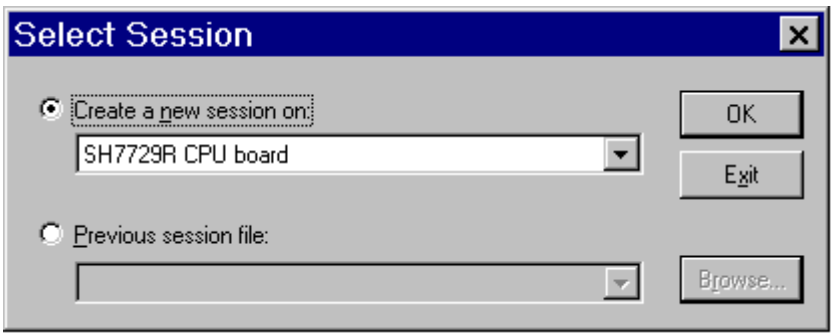


Figure 3.2 [Select Session] Dialog Box

The message box shown in figure 3.3 will appear. Check that the CPU board power is turned on, and click the [OK] button.

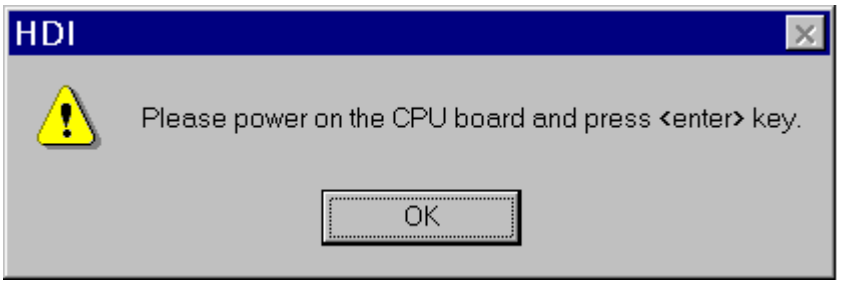


Figure 3.3 Power Supply Confirmation Message Box

When `Link up` appears on the status bar, HDI startup is completed.

If `Link up` does not appear, check the items listed in table 3.2.

Table 3.2 Check Items When HDI Cannot Be Initiated

Check Item	Reference in this Manual
Check that the power monitoring LED (LED1) on the CPU board is turned on	Section 5.5
Check that the host computer and the CPU board are correctly connected through a serial cable.	Sections 2.4 and 5.4.1
Check that the port and baud rate are set correctly in the [Monitor Setup] dialog box.	Sections 3.5 and 4.2.1
Check that the jumper pins are correctly inserted into the jumpers on the CPU board.	Section 2.8

3.3 [HDI] Window

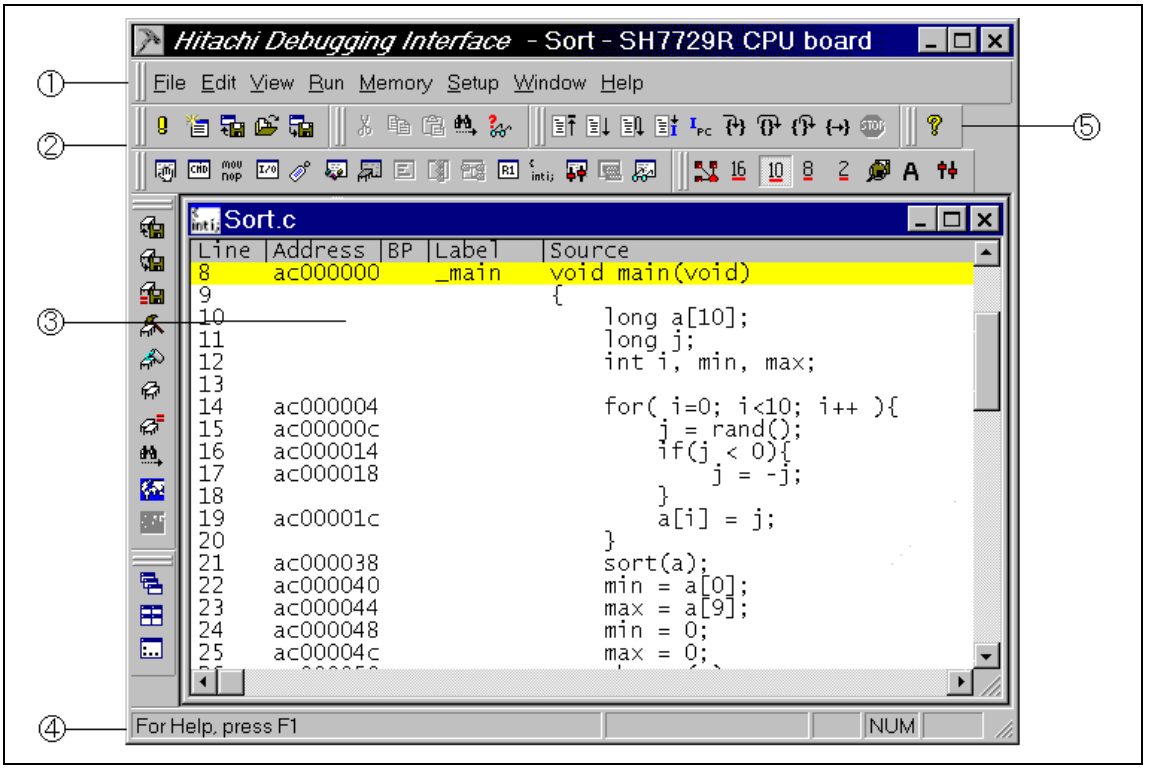


Figure 3.4 [HDI] Window

The key functions of the HDI are described in section 4, Descriptions of Windows. Numbers in figure 3.4 indicate the following:

1. Menu bar: Gives the user access to the HDI commands for using the HDI debugger.

2. Toolbar: Provides convenient buttons as shortcuts for the most frequently used menu commands.

3. Program window: Displays the source program being debugged.

4. Status bar: Displays the status of the CPU board, and progress information about downloading.

5. [Help] button: Activates on-line help about any features of the HDI user interface.

3.4 Setting up the CPU Board

The following conditions can be set up on the CPU board before downloading the program:

- Connection method
- I/O definition file
- Options on program load

The following describes how to set up the CPU board for the tutorial programs.

3.5 Setting the [Monitor Setup] Dialog Box

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

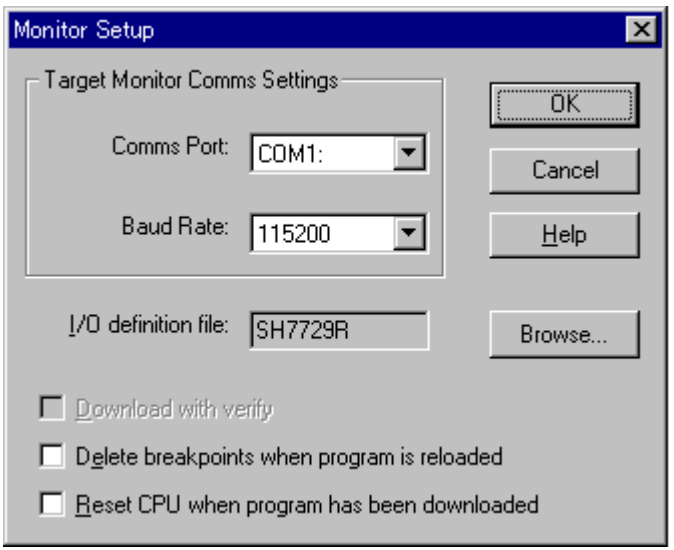


Figure 3.5 [Monitor Setup] Dialog Box

Set options as follows:

Table 3.3 Setting the [Monitor Setup] Dialog Box

Option	Default	Value
Comms Port:	COM1:	Select from among COM1, COM2, COM3, or COM4 as the host computer serial port.
Baud Rate:	115200	Sets the serial baud rate. Select either 57600 bit/s or 115200 bit/s, to match the setting of jumper J7. Connection is not possible at any other setting.
I/O definition file	SH7729R	Sets the I/O register definition file. The SH7729R definition file is set as default.. On selecting a file, the [I/O Registers] window (accessed from the [View] menu) can be used to display register information.
Download with verify	—	The CPU board does not support this function (this box cannot be selected).
Delete breakpoints when program is reloaded	Unchecked	When this box is checked, all breakpoints are deleted when a program is reloaded.
Reset CPU when program has been downloaded	Unchecked	When this box is checked, registers are initialized* when a program is loaded. No reset signal is input to the CPU board.

Note: Only the program counter and status register and VBR are initialized (PC = H'AC000000 and SR = H'600010E0 The value of VBR depends on the endian).

- Click the [OK] button.

Notes: 1. The I/O register definition file can be selected in this dialog box. Be sure to select a file within the HDI installation directory. Otherwise, the I/O register window will not operate correctly.

2. The name of the I/O register definition file can consist of up to nine characters. This number does not include the file name's extension.

3.6 Downloading the Tutorial Program

3.6.1 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. Enter the offset and file name in the [Offset] edit box and [File name] list box as shown in figure 3.6 and click the [Open] button.

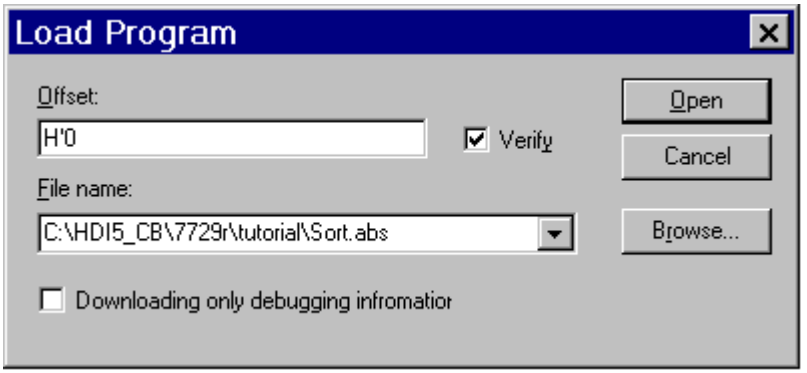


Figure 3.6 [Load Program] Dialog Box

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

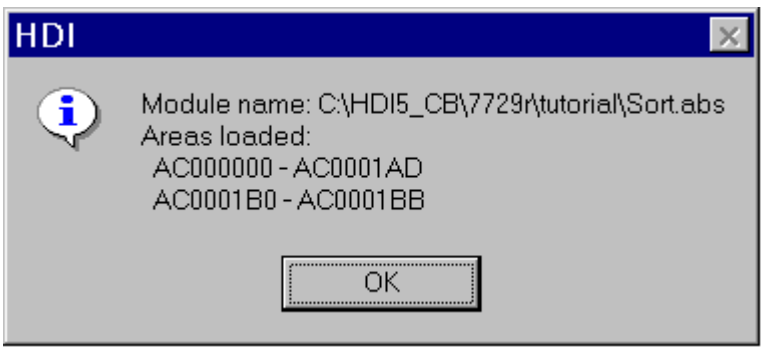


Figure 3.7 [HDI] Message Box

- Click the [OK] button to continue.

3.6.2 Displaying the Source Program

The HDI allows the user to debug a program at the source level.

- Select [Source…] from the [View] menu. The [Open] dialog box is displayed.
- Select the C source file that corresponds to the object file the user has loaded.

Figure 3.8 [Open] Dialog Box

- Select [sort.c] and click the [Open] button. The [Program] window is displayed.

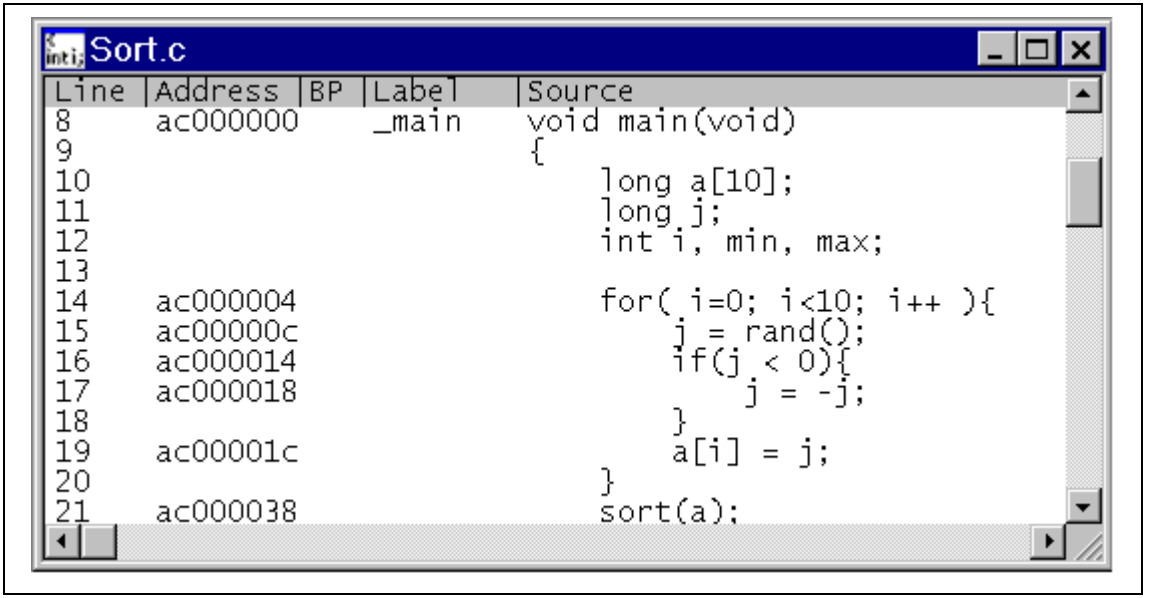


Figure 3.9 [Program] Window (Displaying the Source Program)

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

Initially the [Program] 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.

3.7 Setting the Software Breakpoint

A breakpoint is one of the easy debugging functions.

The [Program] window provides a very simple way of setting a software breakpoint in a program. For example, to set a breakpoint at the `sort` function call:

- Select by double-clicking the [BP] column on the line containing the `sort` function call.

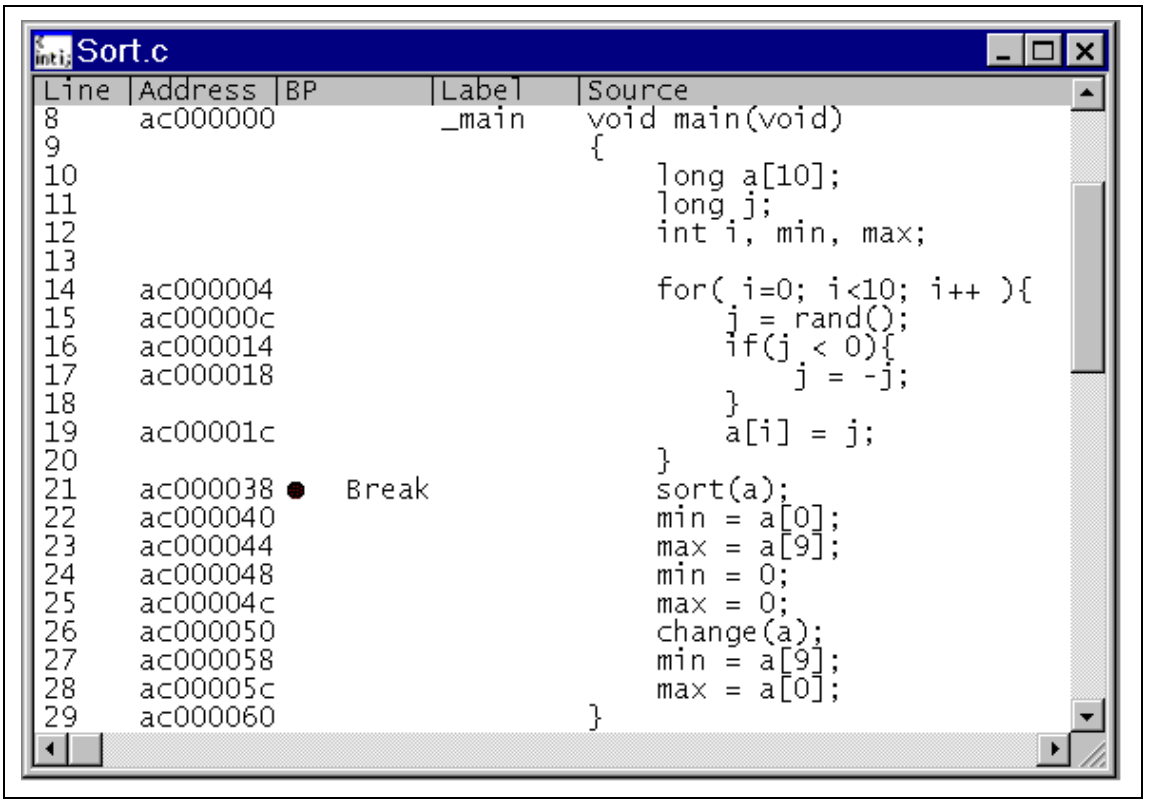


Figure 3.10 [Program] Window (Setting a Software Breakpoint)

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

Note: The software breakpoint cannot be set in the ROM area or the delay slots in the program.

3.8 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.

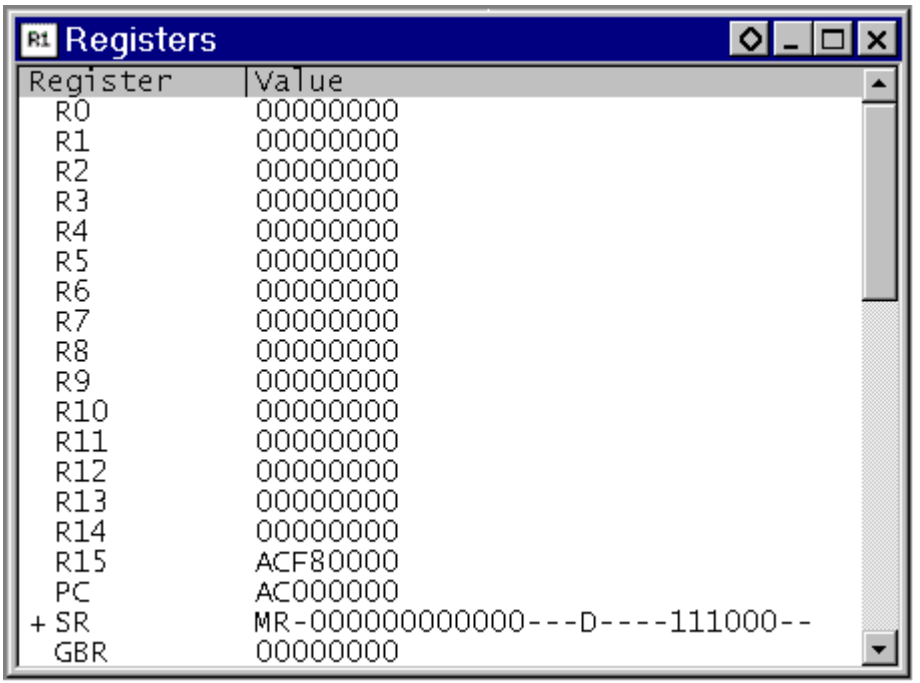


Figure 3.11 [Registers] Window

- To change the value of the program counter, double-click the value area in the [Registers] window with the mouse. The following dialog box is then displayed, and the value can be changed.

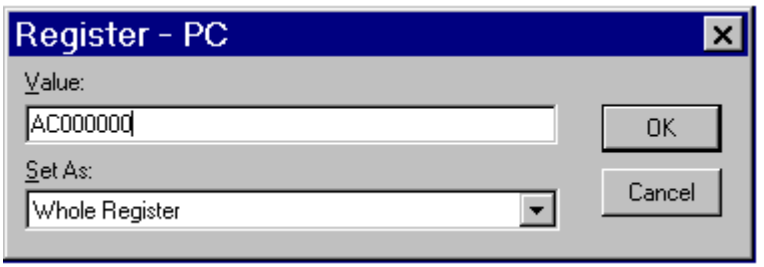


Figure 3.12 [Register] Dialog Box (PC)

- Enter `H'AC000000` in the [Value] edit box, and click the [OK] button.
- To change the value of the stack pointer, move the mouse pointer to the value area of [R15] in the [Registers] window and enter a new value directly, or double-click the value area with the mouse to open the following dialog box.

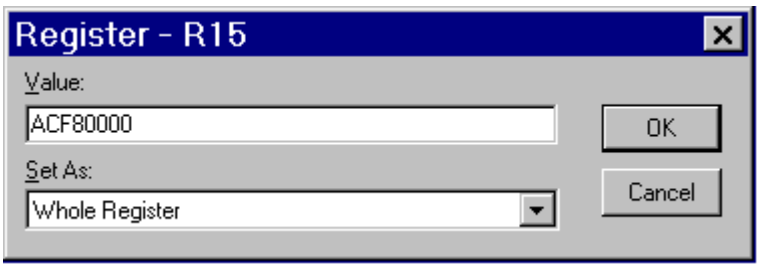


Figure 3.13 [Register] Dialog Box (SP)

- Enter `H'ACF80000` in the [Value] edit box, and click the [OK] button.

3.9 Executing the Program

Execute the program as described in the following:

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

Figure 3.14 [Go] Button

The program will be executed up to the breakpoint that has been inserted, and a statement will be highlighted in the [Program] window to show the position that the program has halted. [Break = Breakpoint] will appear on the status bar.

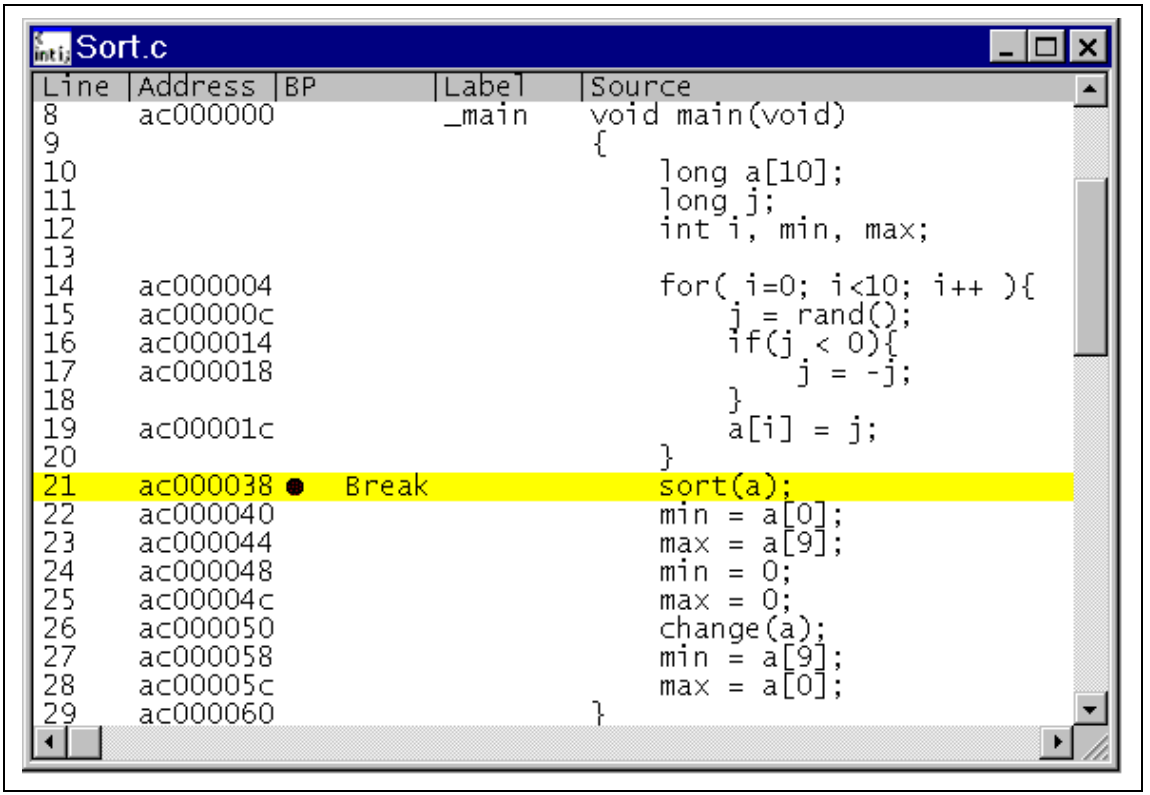


Figure 3.15 [Program] Window (Break Status)

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

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

  The [System Status] window will appear. Open the [Platform] page and check the status of `Break Cause`.

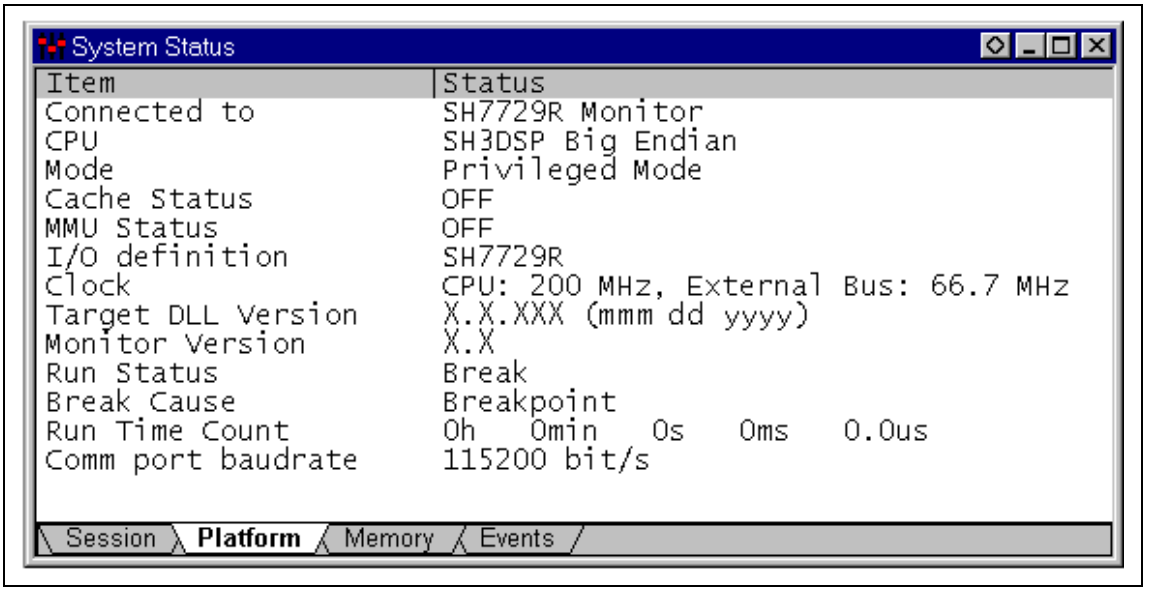


Figure 3.16 [System Status] Window

The [System Status] window displays the following items in each page.

Table 3.4 Contents of the [System Status] Window

Page	Item	Description
[Session]	Target system	Indicates whether the CPU board is connected or not.
	Session Name	Displays the session file name.
	Program Name	Displays the load module file name.
[Platform]	Connected to	Displays the name of the connected CPU board monitor program.
	CPU	Displays the target CPU and endian setting.
	Mode	Displays the CPU processor mode (privileged mode or user mode).
	Cache Status	Shows whether the cache is enabled or disabled.
	MMU Status	Shows whether the MMU is enabled or disabled.
	I/O definition	Displays the selected I/O register definition file.
	Clock	Displays the clock frequency (CPU operating frequency and external bus frequency) being used.
	Target DLL Version	Indicates the version of the target DLL for connection to the CPU board.
	Monitor Version	Displays the monitor program version.
	Run Status	Displays the user program execution status: Run: Being executed Break: Stopped
	Break Cause	Displays the cause of the program stopping at break.
	Run Time Count	Shows the time from the start of the user program to the break. When the run time count function is disabled, "0h 0min 0s 0ms 0.0us" is displayed.
	Comm port baudrate	Indicates the data baud rate for the serial interface.
[Memory]	Target Device Configuration	Not supported by this CPU board.
	System Memory Resources	Not supported by this CPU board.
	Loaded Memory Areas	Shows the area where the load module is loaded.
[Events]	Resources	Shows the number of breakpoints set.

3.10 Reviewing Breakpoints

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

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

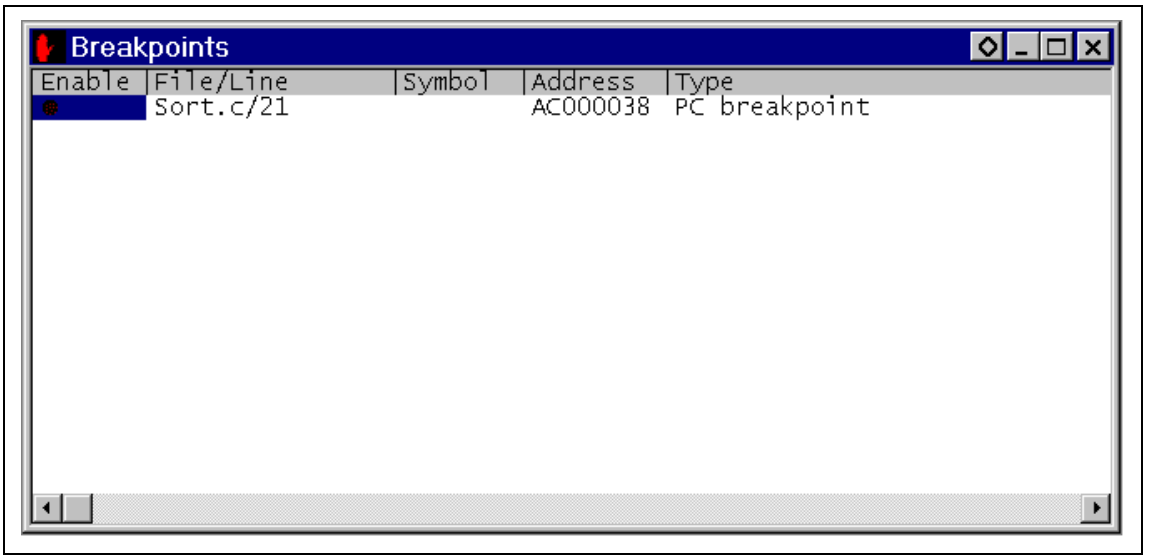


Figure 3.17 [Breakpoints] Window

Right-clicking in the [Breakpoints] window will open a pop-up menu, through which breakpoints can be set, changed, deleted, enabled, or disabled.

3.11 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` in word size:

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

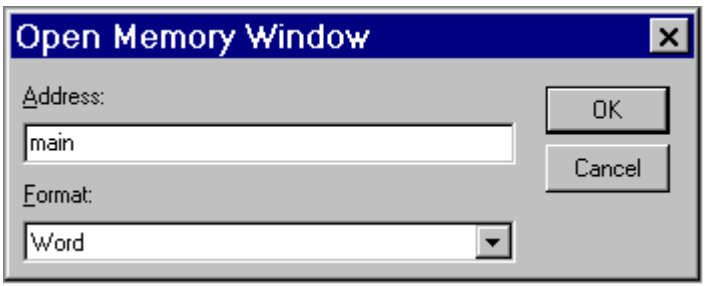


Figure 3.18 [Open Memory Window] Dialog Box

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

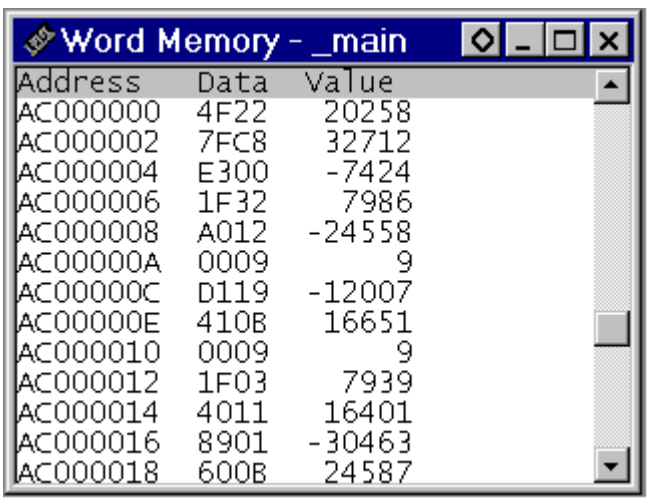


Figure 3.19 [Word Memory] Window

3.12 Watching Variables

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

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

The following dialog box will be displayed.

Figure 3.20 [Instant Watch] Dialog Box

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

Figure 3.21 [Watch Window] Window (Displaying the Array)

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

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

The following dialog box will be displayed.

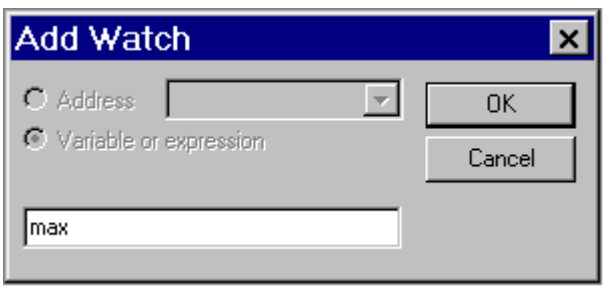


Figure 3.22 [Add Watch] Dialog Box

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

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

Figure 3.23 [Watch Window] Window (Displaying the Variable)

The user can double-click the + symbol to the left of array a to watch window the all elements in array a.

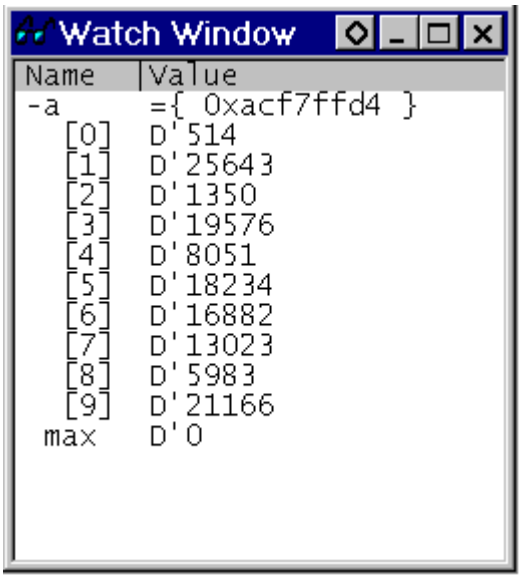


Figure 3.24 [Watch Window] Window (Displaying Array Elements)

3.13 Stepping Through a Program

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

Table 3.5 Step Option

Menu Command	Description
Step In	Executes each statement, including statements within functions.
Step Over	Executes a function call in a single step.
Step Out	Steps out of a function, and stops at the statement following the statement in the program that called the function.
Step…	Steps the specified times repeatedly at a specified rate.

Use the [Go] described in section 3.9, Executing the Program, to confirm that the program is executed up to the `sort` function statement at address H'AC000038.

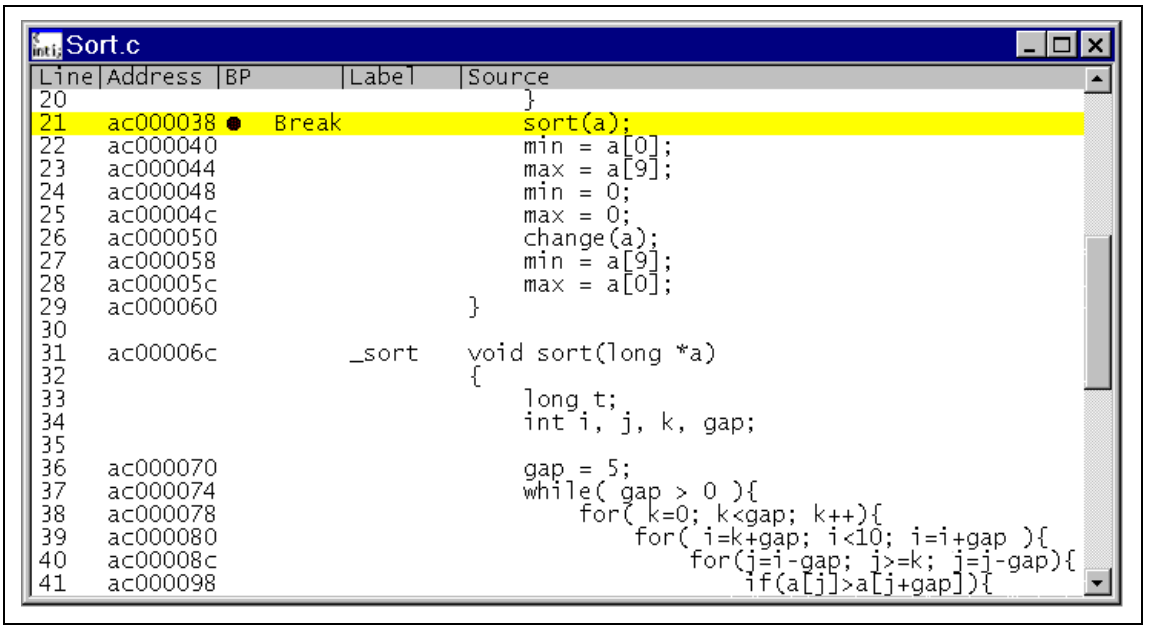


Figure 3.25 [Program] Window (Step Execution)

3.13.1 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 3.26 [Step In] Button

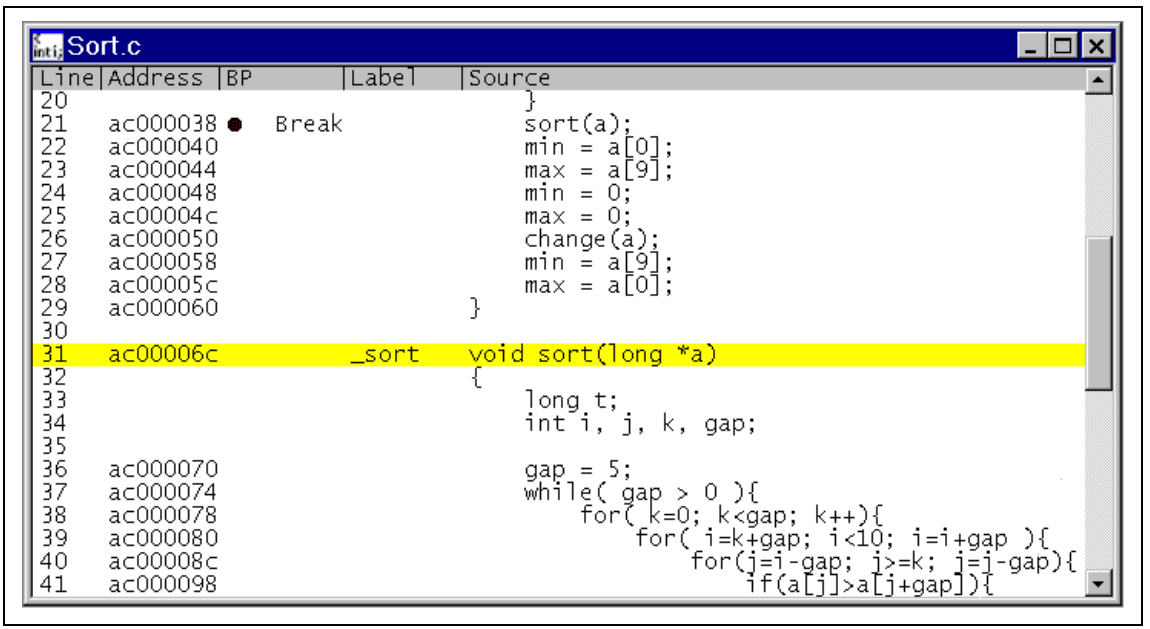


Figure 3.27 [Program] Window (Step In)

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

3.13.2 Executing [Step Out] Command

The [Step Out] steps out of the called function and stops at the next statement of the calling statement in the main function.

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

Figure 3.28 [Step Out] Button

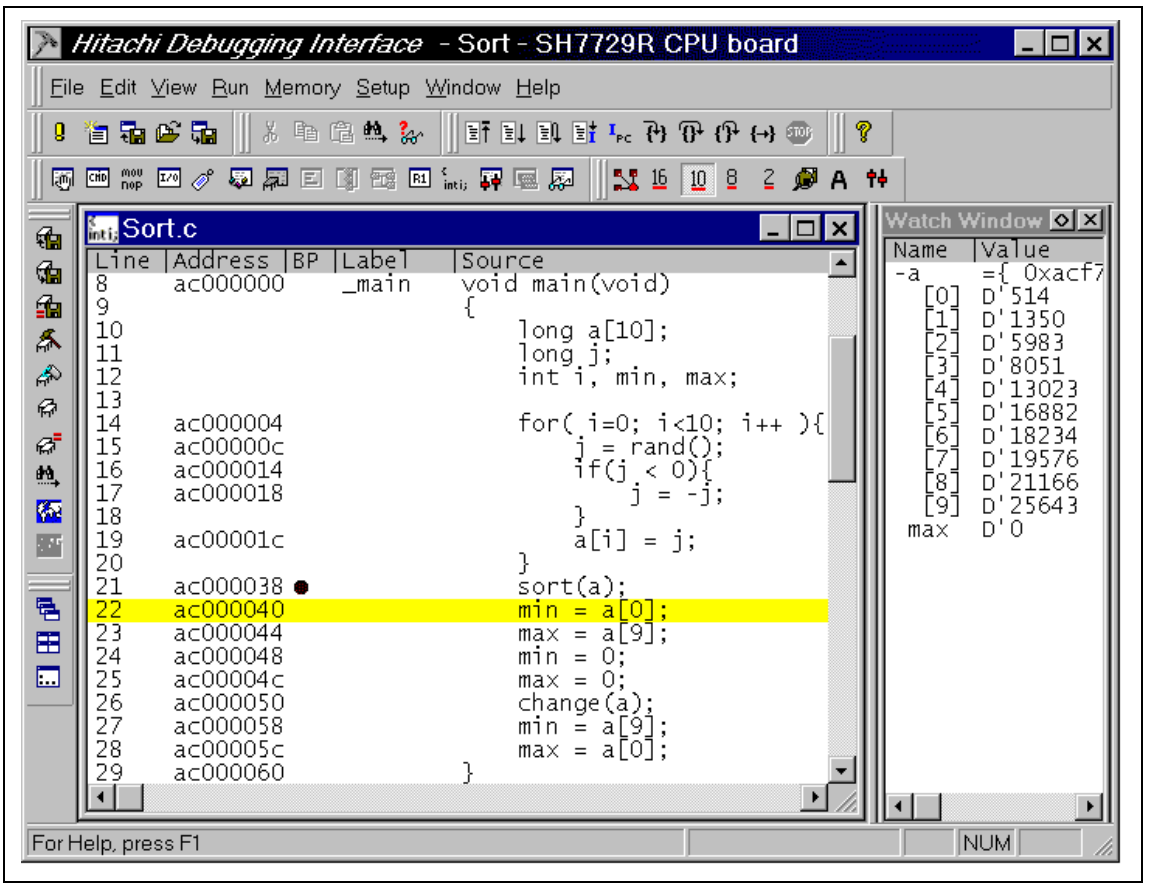


Figure 3.29 [Program] Window (Step Out)

- The values of array a is sorted in ascending order.
- To execute two steps, use [Step In] twice.

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

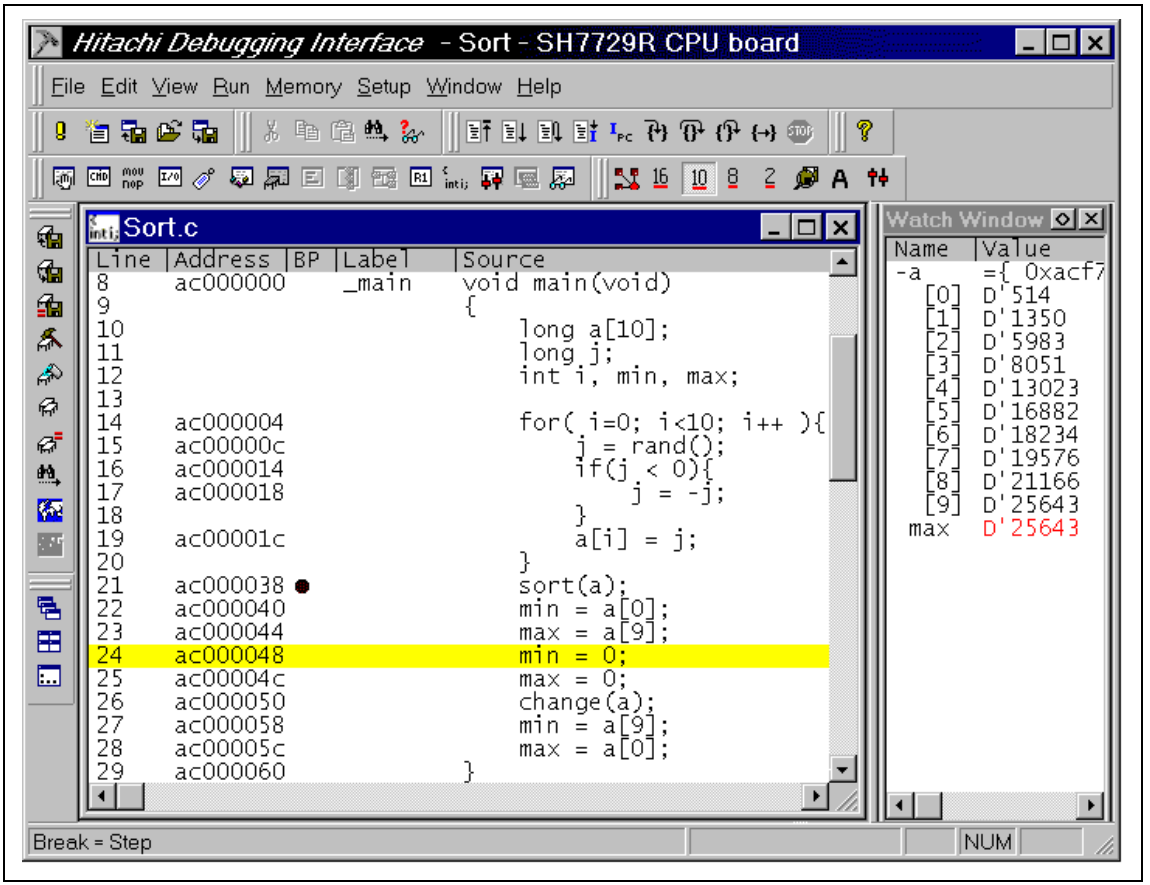


Figure 3.30 [Program] Window (Step In –> Step In)

3.13.3 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.

- Using [Step In], execute two steps to reach the change function statement.
- 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 3.31 [Step Over] Button

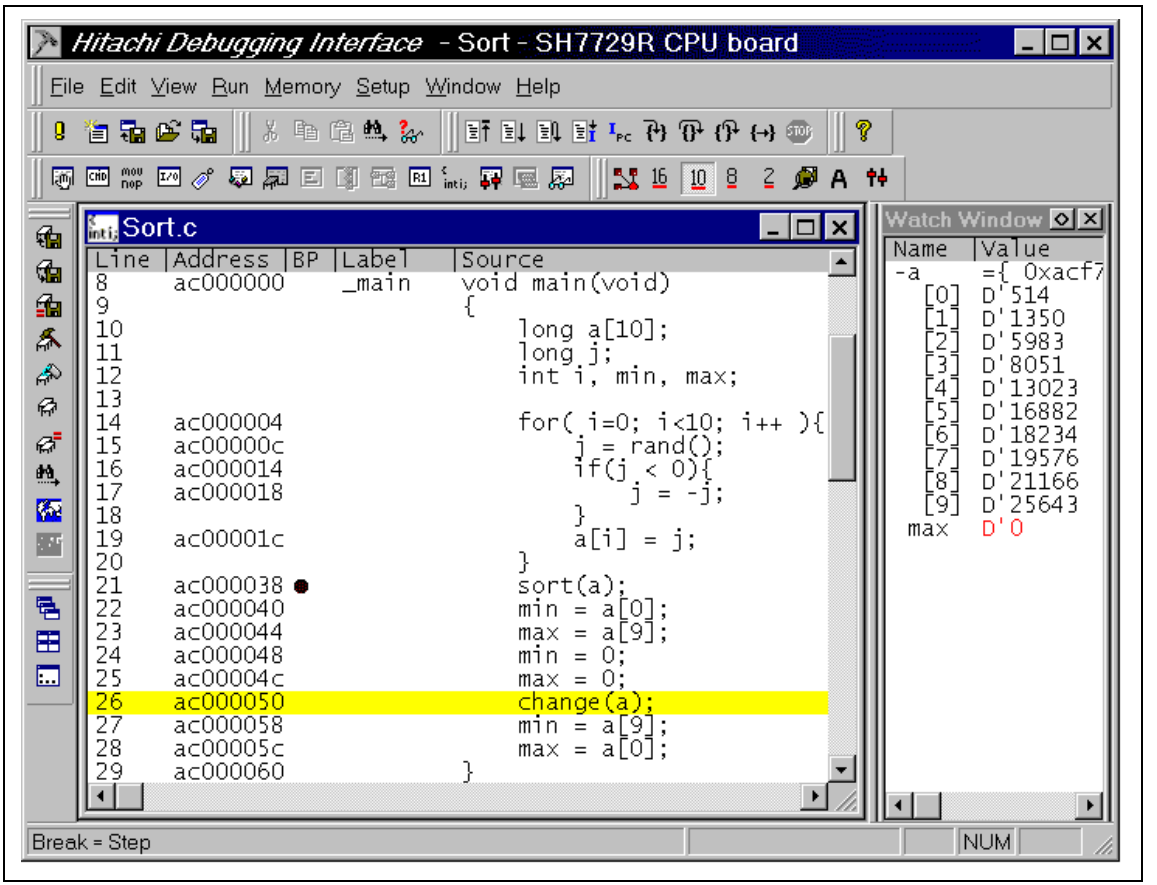


Figure 3.32 [Program] Window (Before Step Over Execution)

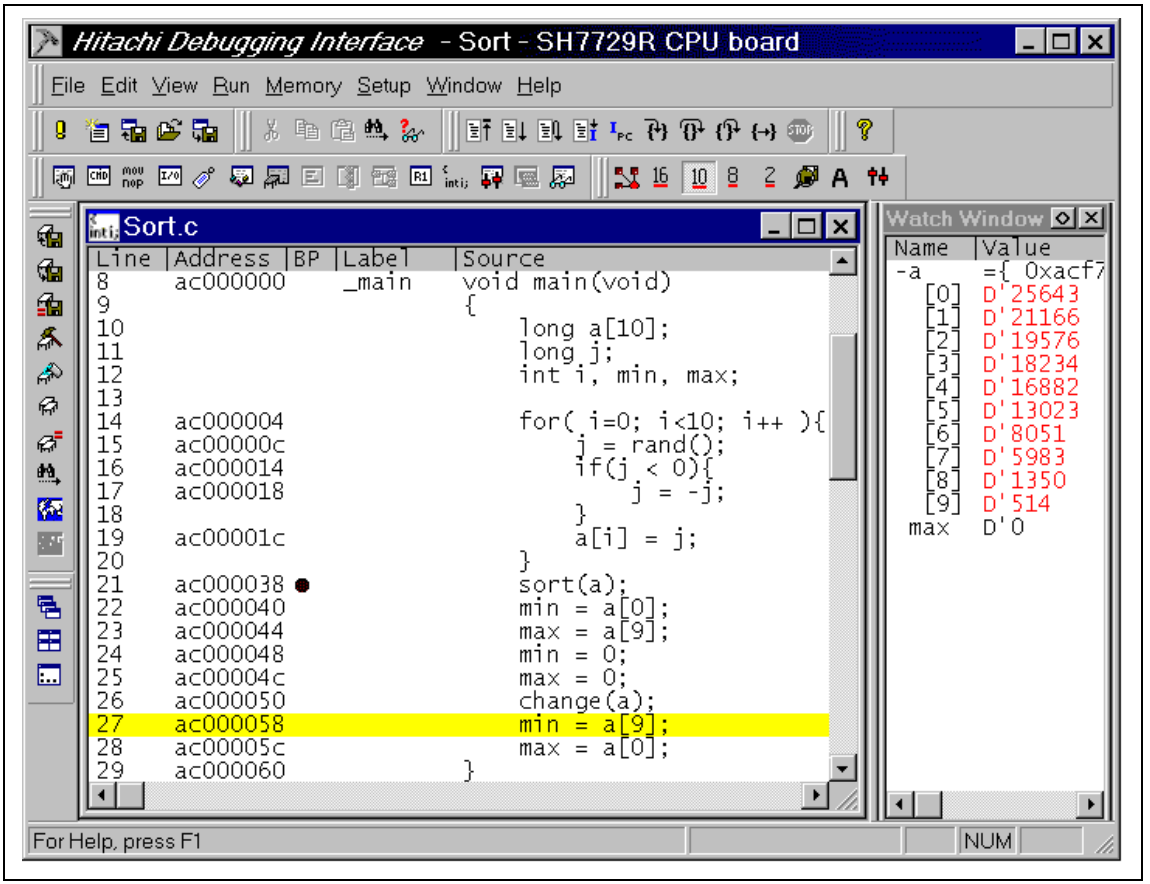


Figure 3.33 [Program] Window (Step Over)

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

3.14 Displaying Local Variables

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

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

If no local variable exists, none is displayed on the [Locals] window.

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

The local variables and the corresponding values are displayed on the [Locals] window.

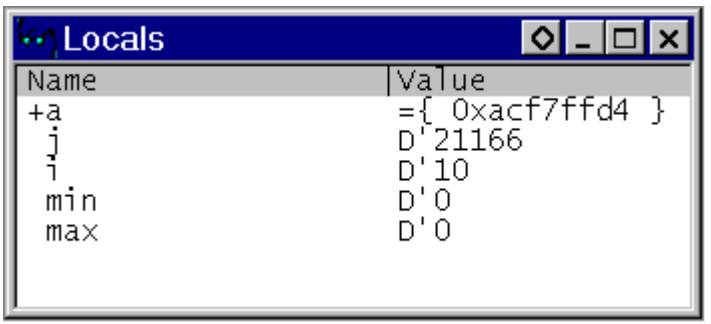


Figure 3.34 [Locals] Window

- Double-click the + symbol in front of array `a` in the [Locals] window to display the elements of array `a`.
- When the elements of the array `a` are referenced before and after executing the `sort` function in the program, the random data should be sorted in descending order. This confirms that the program is operating normally.

3.15 Software Break Function

The CPU board has software break function. With the HDI, a software breakpoint can be set using the [Breakpoints] window. The CPU board can set up to 255 software breakpoints.

Setting a software breakpoint is described below.

- Select [Breakpoints] from the [View] menu. The [Breakpoints] window is displayed.
- Right-click in the [Breakpoints] window to open a pop-up menu, and select the [Delete All] button to cancel all the breakpoints that have been set. A dialog box will prompt you to confirm the deletion of breakpoints. Click [Yes] to delete the breakpoints.

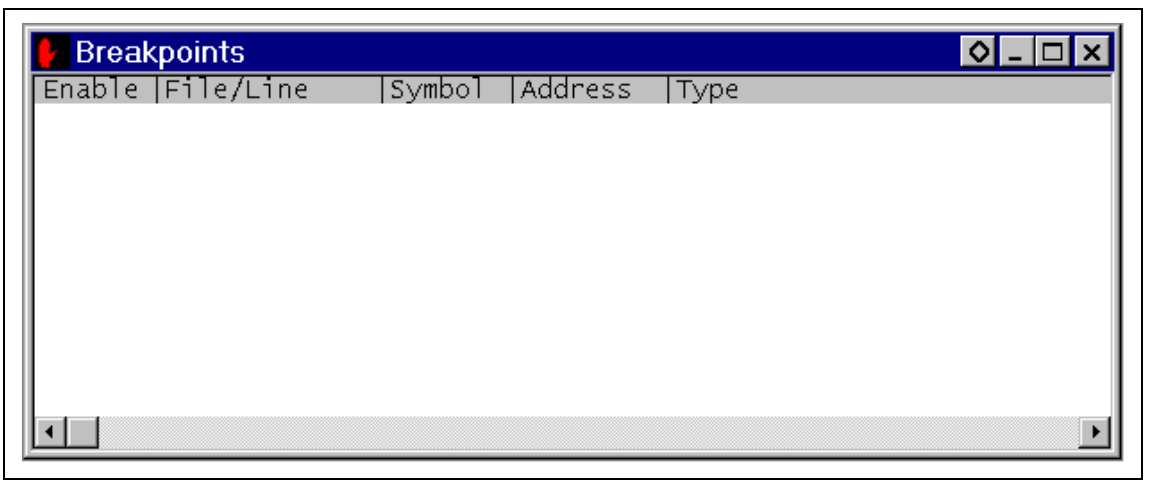


Figure 3.35 [Breakpoints] Window (Before setting software break)

Right-click in the [Breakpoints] window to open a pop-up menu, and select [Add].

The [Add/Edit Breakpoint] dialog box is displayed. Either an address or a symbol can be entered.

- Enter `change` and check the [Enable] checkbox.

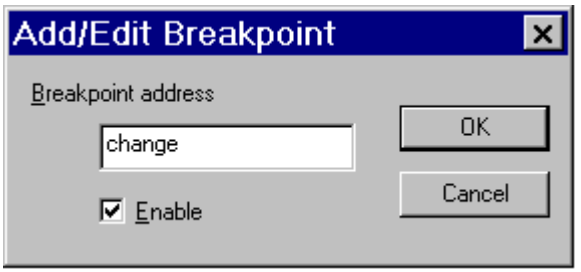


Figure 3.36 [Add/Edit Breakpoint] Dialog Box

- Click the [OK] button.

The software breakpoint that has been set is displayed in the [Breakpoints] window.

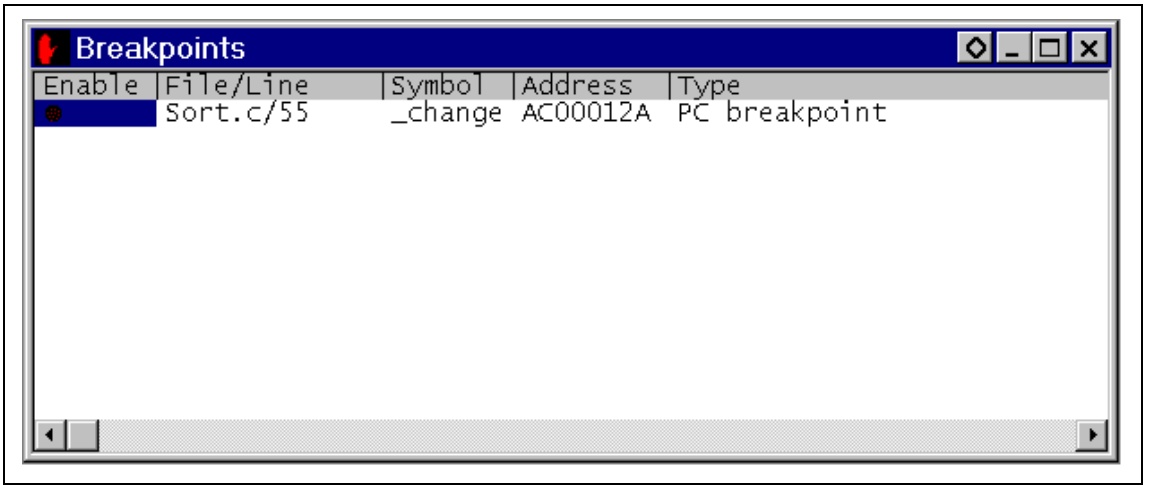


Figure 3.37 [Breakpoints] Window (Software Breakpoint Setting)

To stop the tutorial program at the breakpoint, the following procedure must be executed:

- Close the [Breakpoints] window.
- Set the program counter and stack pointer values that have been set in section 3.8, Setting Registers, (PC = H'AC000000, R15 = H'ACF80000) in the [Registers] window. Click the [Go] button.

The program runs, and stops at the set breakpoint.

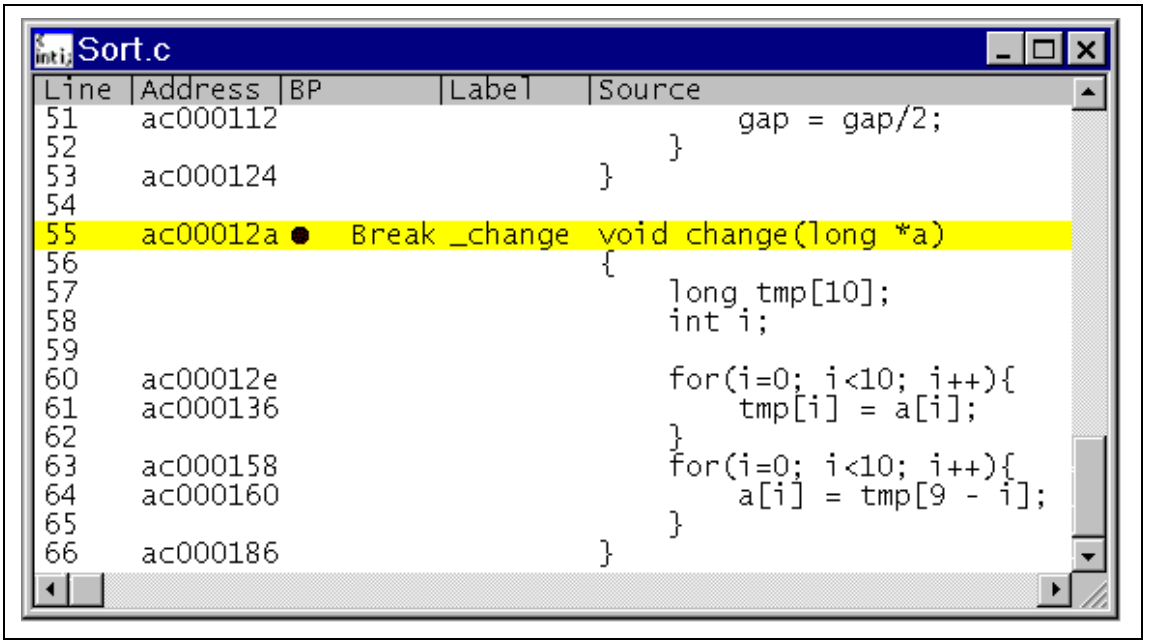


Figure 3.38 [Program] Window at Execution Stop (Software Break)

Select [Status] from the [View] menu. The [System Status] window displays the following contents. The window confirms that execution was stopped at a breakpoint.

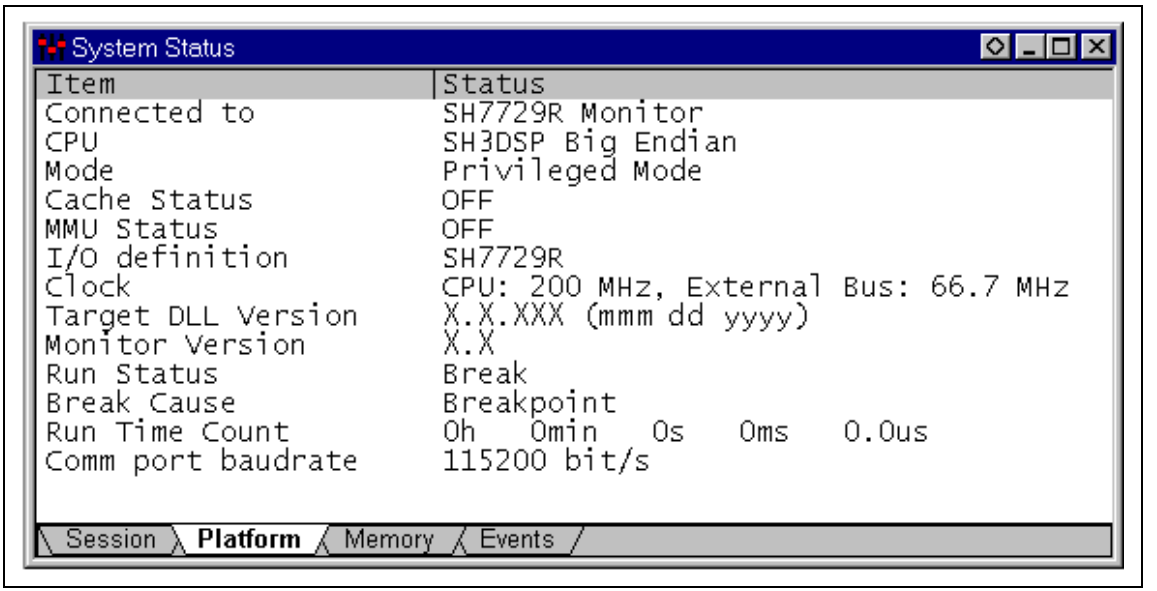


Figure 3.39 Displayed Contents of the [System Status] Window (Software Break)

3.16 Run Time Count Function

By enabling the run time count function and executing the user program, the user program run time can be measured. In the following example, the run time of the `sort` function is measured.

- Select [Delete All] from the pop-up menu in the [Breakpoints] window, and double-click the [BP] column on the 21st and 22nd lines in the [Program] window to set breakpoints.
- Set the program counter and stack pointer values that have been set in section 3.8, Setting Registers, (PC = H'AC000000, R15 = H'ACF80000) in the [Registers] window. Click the [Go] button.

The program runs and stops at the breakpoint.

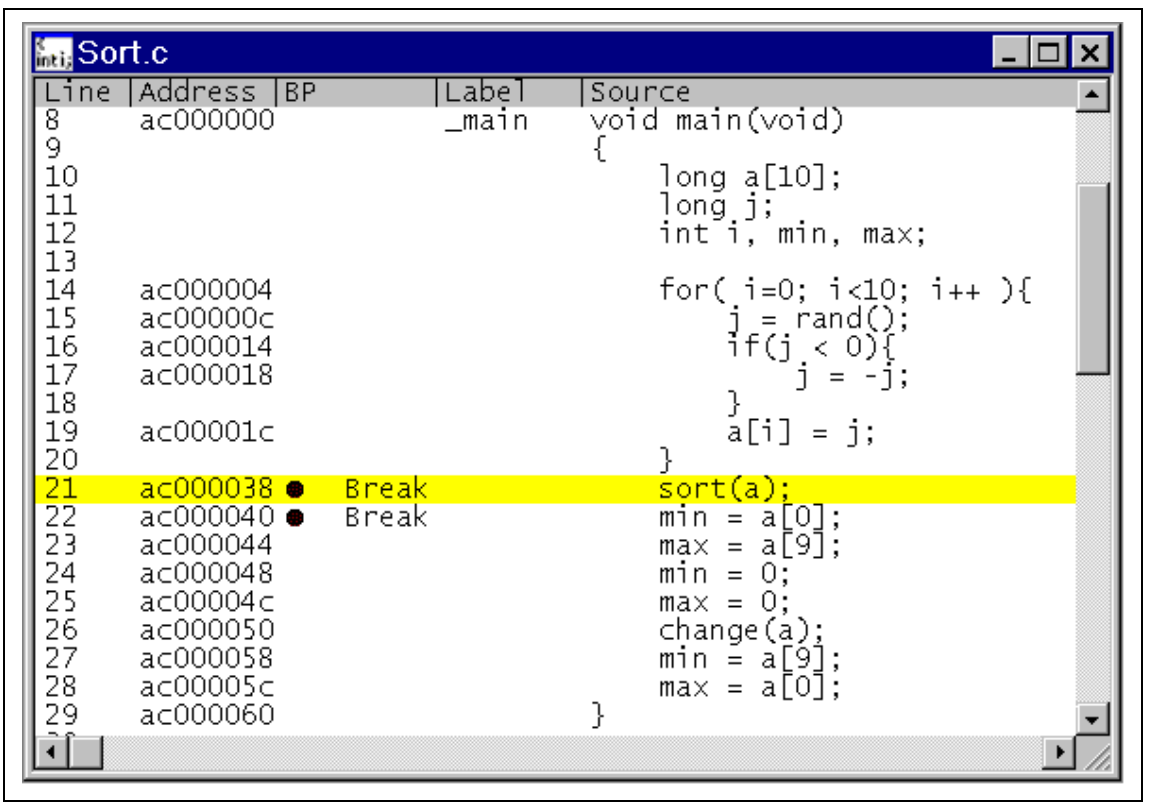


Figure 3.40 [Program] Window (Break before Run Time Count)

- Select [Status] from the [View] menu. The [System Status] window will appear.

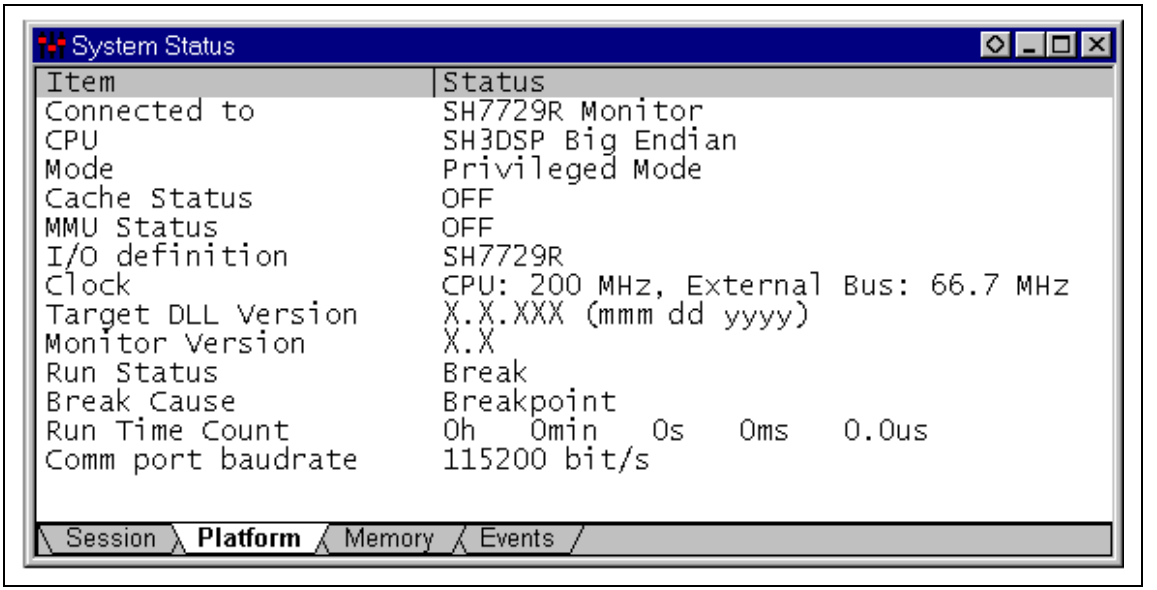


Figure 3.41 [System Status] Window (Run Time Count Disabled)

The time taken from the start of the program to the break is shown as the `Run Time Count` on the [Platform] sheet in the [System Status] window. In this example, the run time count function has not been enabled, "0h 0min 0s 0ms 0.0us" is displayed.

- Select [Run Time...] from the [View] menu. The [Run Time Count Condition] dialog box will appear.
- Check the [Enable] check button, select a measurement unit from [Measurement Mode], and click the [OK] button. In this example, 0.12us (Max 9min) is selected.

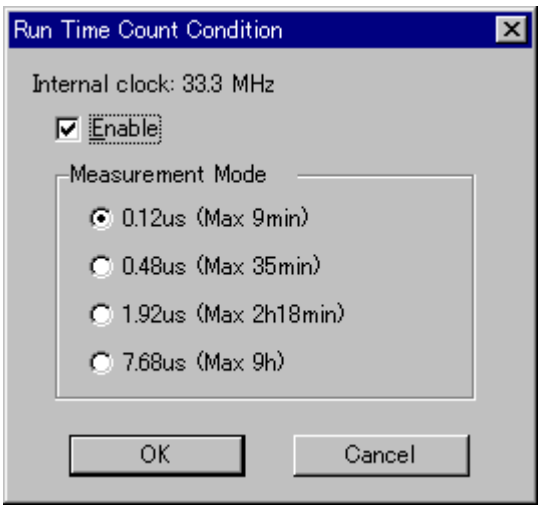


Figure 3.42 [Run Time Count Condition] Dialog Box

The items listed in table 3.6 can be set in the [Run Time Count Condition] dialog box. The run time can be checked in the [System Status] window.

Table 3.6 Items Set in [Run Time Count Condition] Dialog Box

Item	Description
Enable	Check this box to enable the run time count function. The default setting is "disable".
Measurement Mode	Select a measurement unit here.

The selectable measurement units are shown in table 3.7.

Table 3.7 Selectable Measurement Units

Measurement Unit	Maximum Measurable Time
0.12 us	Approximately 9 minutes
0.48 us	Approximately 35 minutes
1.92 us	Approximately 2 hours and 18 minutes
7.68 us	Approximately 9 hours

- Click the [OK] button to enable the run time count function.

- Click the [Go] button. Execution will stop at the line following the `sort` function, and the run time will be displayed in a message box.

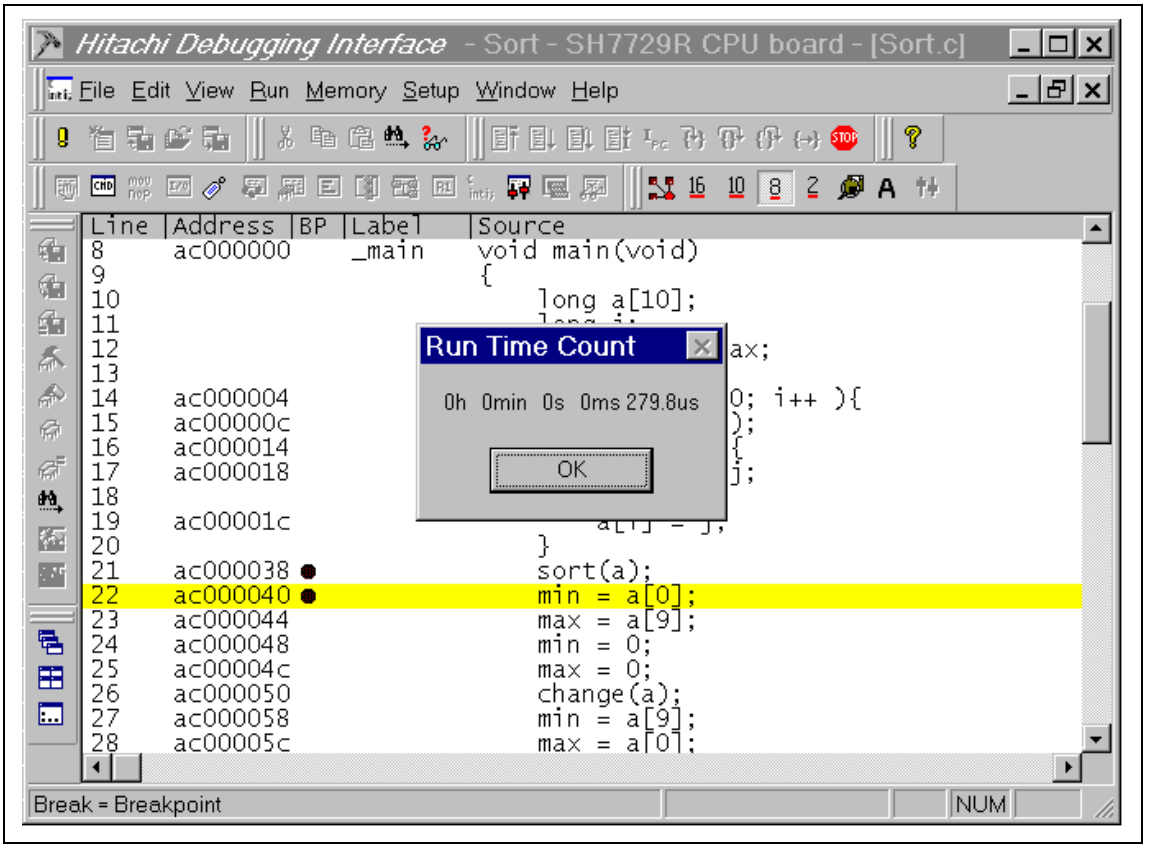


Figure 3.43 [Program] Window (Stopped at a Breakpoint after Run Time Count)

The time from the `sort` function call to the return to the caller can be checked at `Run Time Count` on the [Platform] sheet in the [System Status] window.

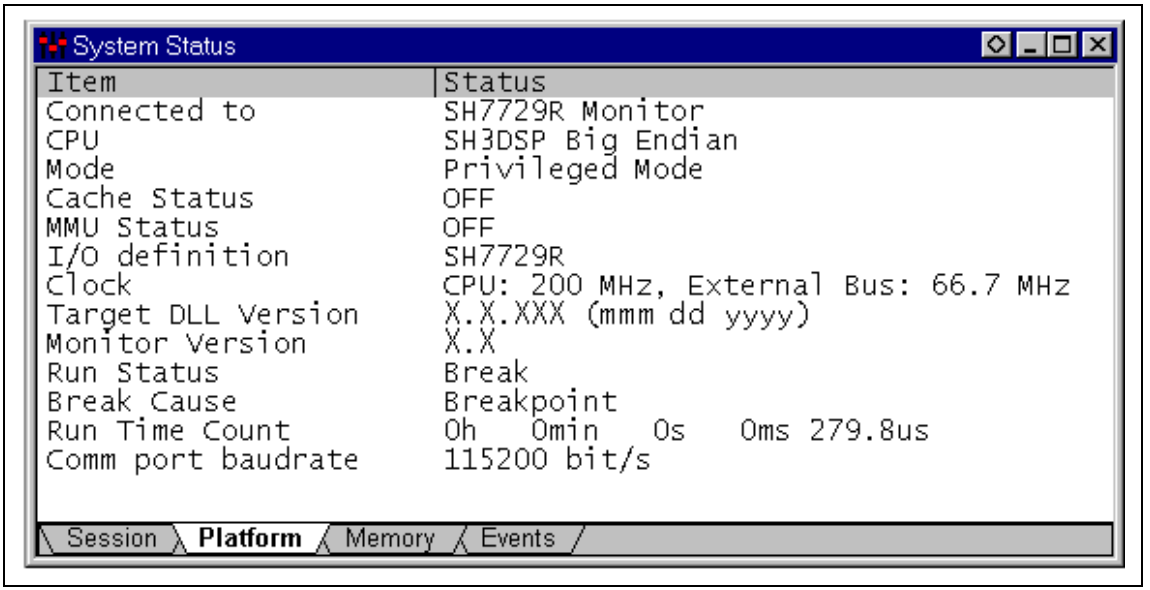


Figure 3.44 [System Status] Window (Run Time Count Result)

Notes: 1. The run time will vary depending on the execution environment.

2. The run time can be measured by executing with [Go] but cannot be measured by [Step In], [Step Out], or [Step Over].

3.17 Saving a Session

If a program has been downloaded, the corresponding source file is displayed, and numerous windows are opened, it can take some time to restore this setup the next time the program is downloaded. The HDI is able to save the current settings for retrieval the next time the program is loaded, in order to reduce setup time.

In order to save a session which has already been named, or to save the session with the same name as the current object file, select [Save Session] from the [File] menu.

To save the current settings as a session with a new name, select the [Save Session As…] command from the [File] menu. A dialog box is displayed; enter the new name for the session. Three files are saved: the HDI session file (*.hds), the target session file (*.hdt), and the watch session file (*.hdw).

The target session file stores the following information:

- Software breakpoint information
- I/O definition file information
- Run time count function information

Note: No symbol or memory information is saved in the session files. If changes are to be used again in future, they must be saved separately. For details, refer to the Hitachi Debugging Interface User's Manual.

3.18 What Next?

In this tutorial, we have introduced as examples program debugging using the CPU board and the HDI.

Further details on the use of the HDI can be found in the Hitachi Debugging Interface User's Manual available in the supplied CD-R.

Section 4 Descriptions of Windows

4.1 HDI Windows

HDI window menu bars and the corresponding pull-down menus are listed in table 4.1. A O mark and/or the relevant section number is shown in the table when menu description is included in the Hitachi Debugging Interface User's Manual or in this manual.

Menu items shown in gray on the screen are not available.

Table 4.1 HDI Window Menus and Related Manual Entries

Menu Bar	Pull-Down Menu	Hitachi Debugging Interface User's Manual	This Manual
File Menu	New Session...	O	—
	Load Session...	O	—
	Save Session	O	—
	Save Session As...	O	3.17
	Load Program...	O	3.6.1
	Initialize	O	—
	Exit	O	—
Edit Menu	Cut	O	—
	Copy	O	—
	Paste	O	—
	Find...	O	—
	Evaluate...	O	—

Table 4.1 HDI Window Menus and Related Manual Entries (cont)

Menu Bar	Pull-Down Menu	Hitachi Debugging Interface User's Manual	This Manual
View Menu	Breakpoints	O	3.10, 3.15, 4.2.2
	Command Line [*4]	O	—
	Disassembly...	O	—
	I/O Area	O	—
	Labels	O	—
	Locals	O	3.14
	Memory...	O	3.11
	Performance Analysis[*1]	O	—
	Profile-List[*1]	O	—
	Profile-Tree[*1]	O	—
	Registers	O	3.8
	Source...	O	3.6.2
	Status	O	3.9, 4.2.4
	Trace[*1]	O	—
	Watch	O	—
	Cache Control...	—	4.2.6
	Run Time...	—	3.16, 4.2.5
	Simulated I/O Window	—	4.2.7
Run Menu	Reset CPU [*5]	O	—
	Go	O	3.9
	Reset Go [*3]	O	—
	Go To Cursor	O	—
	Set PC To Cursor	O	—
	Run...	O	—
	Step In	O	3.13.1
	Step Over	O	3.13.3
	Step Out	O	3.13.2
	Step...	O	—
	Halt	O	—

Table 4.1 HDI Window Menus and Related Manual Entries (cont)

Menu Bar	Pull-Down Menu	Hitachi Debugging Interface User's Manual	This Manual
Memory Menu	Refresh	O	—
	Load...	O	—
	Save...	O	—
	Verify...	O	—
	Test...	O	—
	Fill...	O	—
	Copy...	O	—
	Compare...	O	—
	Configure Map...[*2]	O	—
	Configure Overlay...[*1]	O	—
Setup Menu	Status bar	O	—
	Options...	O	—
	Radix	O	—
	Customize	O	—
	Configure Platform...	O	3.5, 4.2.1
Window Menu	Cascade	O	—
	Tile	O	—
	Arrange Icons	O	—
	Close All	O	—
Help Menu	Index	O	—
	Using Help	O	—
	Search for Help on	O	—
	About HDI	O	—

Notes: 1. Function not supported.
2. Only CPU board ROM and RAM information display is supported.
3. User program is executed after setting PC to H'AC000000.
4. Function for test use. Correct operation cannot be guaranteed.
5. PC ,SR ,and VBR are initialized. The reset signal is not sent to the CPU.

4.2 Descriptions of Each Window

This section describes each window.

4.2.1 [Monitor Setup] Dialog Box

Function:

Specifies the setup conditions for the CPU board. This dialog can be displayed by selecting [Configure Platform…] from the [Setup] menu.

Window:

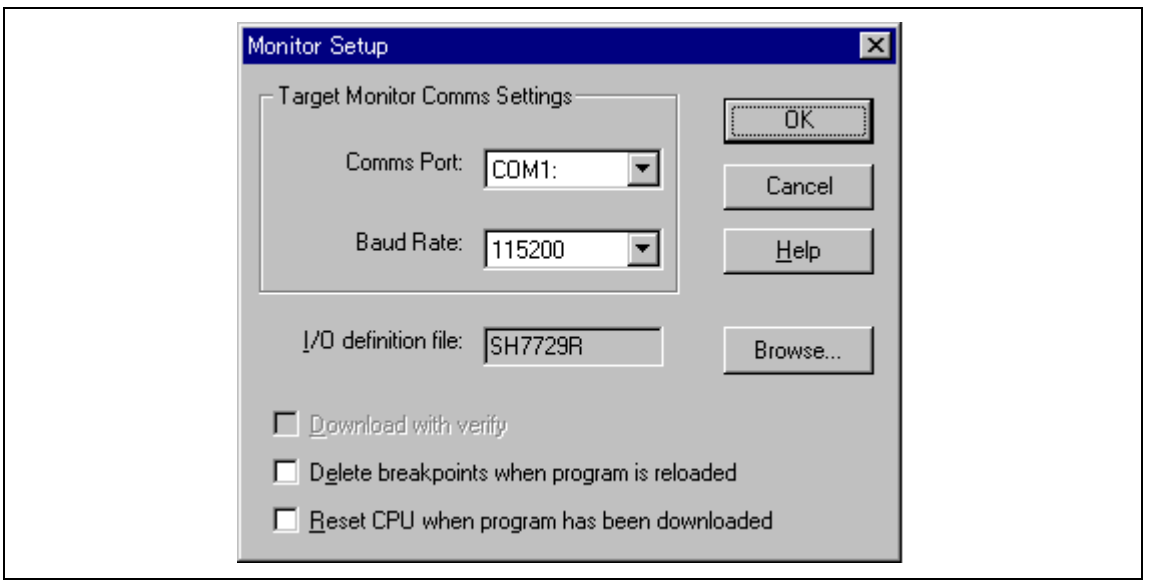


Figure 4.1 [Monitor Setup] Dialog Box

Notes: 1. The I/O register definition file can be selected in this dialog box. Be sure to select a file within the HDI installation directory. Otherwise, the I/O register window will not operate correctly.

2. The name of the I/O register definition file can consist of up to nine characters. This number does not include the file name's extension.

Description:

The settings of the [Monitor Setup] dialog box are indicated below.

Table 4.2 [Monitor Setup] Dialog Box Page

Option	Setting
Comms Port:	COM1, COM2, COM3, or COM4 can be selected as the host computer serial port.
Baud Rate:	Sets the serial baud rate. Select either 57600 bit/s or 115200 bit/s, to match the setting of jumper J7. Connection is not possible at any other setting.
I/O definition file:	Sets the I/O register definition file. Click the [Browse] button to select the SH7729R definition file. When selecting a file, the [I/O Registers] window (accessed from the [View] menu) can be used to display register information.
Download with verify	The CPU board does not support this function (this box cannot be selected).
Delete breakpoints when program is reloaded	When this box is checked, all breakpoints are deleted when a program is reloaded.
Reset CPU when program has been downloaded	When this box is checked, registers are initialized* when a program is loaded. No reset signal is input to the CPU board.

Note: Only the program counter ,status register and VBR are initialized (PC = H'AC000000 and SR = H'600010E0 The value of VBR depends on the endian.).

Clicking the [OK] button sets the setup conditions. If the [Cancel] button is clicked, this dialog box is closed without setting the conditions.

4.2.2 [Breakpoints] Window

Function:

This window lists all break conditions that have been set. This window can be displayed by selecting [Breakpoints] on the [View] menu.

Window:

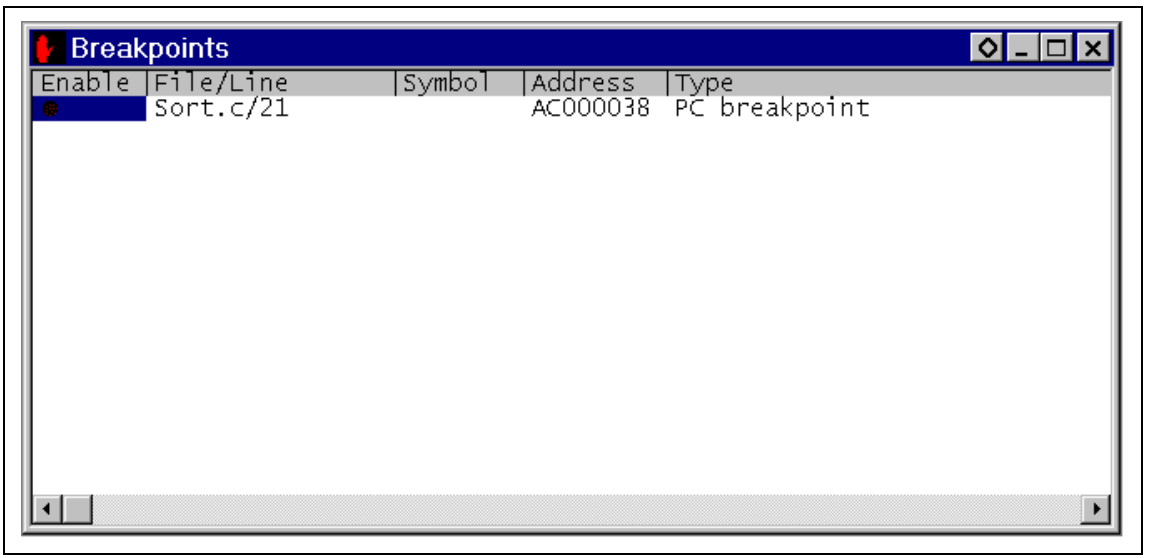


Figure 4.2 [Breakpoints] Window

Description:

The [Breakpoints] window displays breakpoint setting information. The items listed in table 4.3 are displayed.

Table 4.3 [Breakpoints] Window Display Items

Item	Description
[Enable]	Displays whether the break condition is enabled or disabled. The "•" indicates that the breakpoint is enabled.
[File/Line]	Displays the file name and line number where the breakpoint is set.
[Symbol]	Displays the symbol corresponding to the breakpoint address. If no symbol has been defined for the address, a blank is displayed.
[Address]	Displays the address where the breakpoint is set.
[Type]	Displays "PC breakpoint"

Right-clicking in the [Breakpoints] window will open a pop-up menu, through which breakpoints can be set, changed, deleted, enabled, or disabled. The pop-up menu functions are described in table 4.4.

Table 4.4 [Breakpoints] Window Pop-up Menu Operation

Name	Description
[Add]	Sets break conditions. Clicking this button will display the [Add/Edit Breakpoint] dialog box, enabling break conditions to be set.
[Edit]	Changes break conditions. Select break conditions to be changed and click this button. The [Add/Edit Breakpoint] dialog box will be displayed, enabling the break condition to be changed.
[Disable] ([Enable])	Enables or disables break conditions. Select break conditions to be enabled or disabled and click this button.
[Delete]	Clears break conditions. Select break conditions to be cleared and click this button.
[Del All]	Clears all break conditions.
[Go to Source]	Jumps to the break address in the [Source] window.

4.2.3 [Add/Edit Breakpoint] Dialog Box

Function:

Sets a breakpoint. This dialog box is displayed when the [Add] or [Edit] is selected in the pop-up menu in the [Breakpoints] window, which is displayed by selecting the [Breakpoints] item on the [View] menu.

Window:

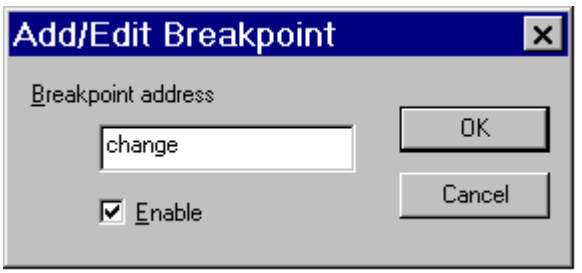


Figure 4.3 [Add/Edit Breakpoint] Dialog Box

Description:

The [Add/Edit Breakpoint] dialog box is made up of the components listed in the table below.

Table 4.5 [Add/Edit Breakpoint] Dialog Box Items

Page Name	Description
[Breakpoint address]	Enter the address or symbol for which a breakpoint is to be set.
[Enable]	The breakpoint is enabled when this box is checked.

After clicking [OK], the breakpoint is set.

4.2.4 [System Status] Window

Function:

This window lists information, such as conditions that have been set to the CPU board and execution results. It is displayed by selecting the [Status] item on the [View] menu.

Window:

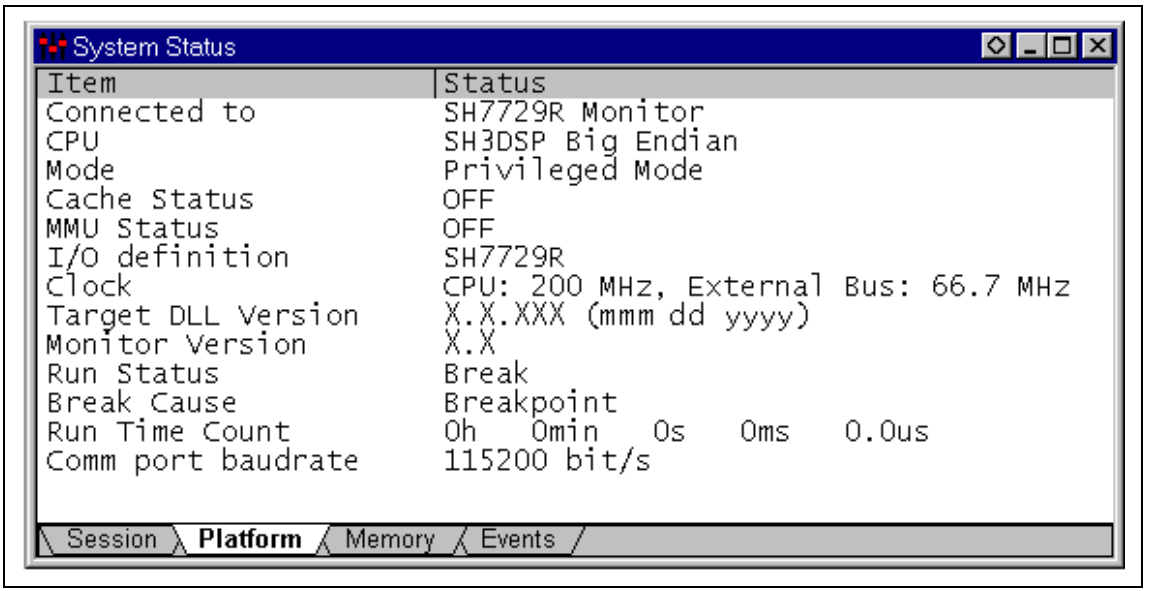


Figure 4.4 [System Status] Window

Description:

The items listed in the following table are displayed in the [System Status] window.

Table 4.6 [System Status] Window Display Items

Page	Item	Description
[Session]	Target system	Indicates whether the CPU board is connected or not.
	Session Name	Displays the session file name.
	Program Name	Displays the load module file name.
[Platform]	Connected to	Displays the name of the connected CPU board monitor program.
	CPU	Displays the target CPU and endian setting.
	Mode	Displays the CPU processor mode (privileged mode or user mode).
	Cache Status	Shows whether the cache is enabled or disabled.
	MMU Status	Shows whether the MMU is enabled or disabled.
	I/O definition	Displays the selected I/O register definition file.
	Clock	Displays the clock frequency (CPU operating frequency and external bus frequency) being used.
	Target DLL Version	Indicates the version of the target DLL for connection to the CPU board.
	Monitor Version	Displays the monitor program version.
	Run Status	Displays the user program execution status: Run: Being executed Break: Stopped
	Break Cause	Displays the cause of the program stopping at break.
	Run Time Count	Shows the time from the start of the user program to the break. When the run time count function is disabled, "0h 0min 0s 0ms 0.0us" is displayed.
	Comm port baudrate	Indicates the data baud rate for the serial interface.
[Memory]	Target Device Configuration	Not supported by this CPU board.
	System Memory Resources	Not supported by this CPU board.
	Loaded Memory Areas	Shows the area where the load module is loaded.
[Events]	Resources	Shows the number of breakpoints set.

4.2.5 [Run Time Count Condition] Dialog Box

Function:

Specifies the condition for measuring the run time. It is displayed by selecting [Run Time...] from the [View] menu.

Window:

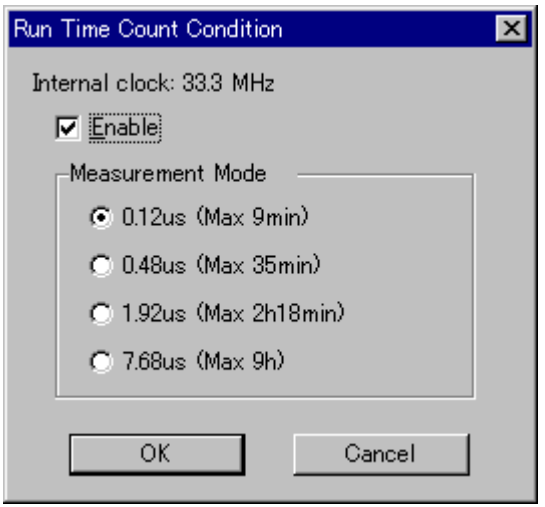


Figure 4.5 [Run Time Count Condition] Window

Description:

The items listed in table 4.7 can be set in the [Run Time Count Condition] dialog box. The run time can be checked in a message box displayed at break or in the [System Status] window.

Table 4.7 Items Set in [Run Time Count Condition] Dialog Box

Item	Description
Enable	Check this box to enable the run time count function. The default setting is "disable".
Measurement Mode	Select a measurement unit here.

The setting is stored when the [OK] button is pressed.

The selectable measurement units are shown in table 4.8.

Table 4.8 Selectable Measurement Units

Measurement Unit	Maximum Measurable Time
0.12 us	Approximately 9 minutes
0.48 us	Approximately 35 minutes
1.92 us	Approximately 2 hours and 18 minutes
7.68 us	Approximately 9 hours

Note: When the maximum measurable time shown in table 4.8 is exceeded, the measured value will be invalid.

4.2.6 [Cache Control] Dialog Box

Function:

Specifies the cache functions. This dialog is displayed on selecting [Cache Control...] from the [View] menu.

Window:

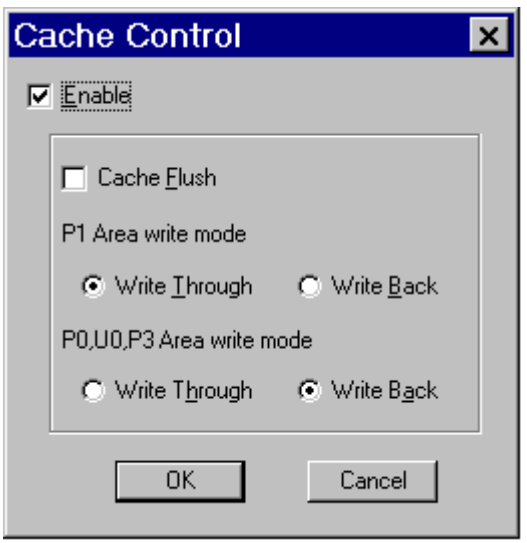


Figure 4.6 [Cache Control] Dialog Box

Description:

The items listed in table 4.9 are displayed and set in the [Cache Control] dialog box. The cache control register settings are displayed when the dialog box is opened. When the [OK] button is clicked, the settings are stored.

Table 4.9 [Cache Control] Dialog Box Items

Item	Description
Cache Flush	Check this box and click the [OK] button to flush all entries in the cache.
Enable	Check this box to enable the cache.
P1 Area write mode	Specifies the operating mode (write-through or write-back) for the P1 area.
P0,U0,P3 Area write mode	Specifies the operating mode (write-through or write-back) for the P0, U0, and P3 areas.

4.2.7 [Simulated I/O Window] Window

Function:

This window displays data input to or output from the serial line during user program execution. It is valid only during execution of a user program. Serial data output by the user program is displayed in this window.

- Data input from the keyboard of the host computer is displayed in this window in addition to being sent to the CPU board.

This window is displayed on selecting the [Simulated I/O Window] item from the [View] menu.

Right-clicking the mouse on this window displays the following pop-up menu.

[Copy] Copies the text appearing in highlighted to the Windows® clipboard.

[Paste] Pastes the contents of the Windows® clipboard to the [Simulated I/O Window], and sends the same contents to the CPU board.

[Clear Window] Clears the contents of the [Simulated I/O Window] window.

Window:

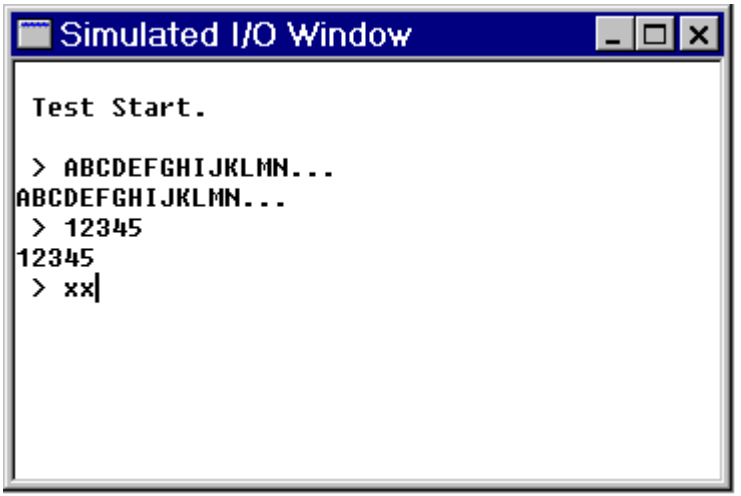


Figure 4.7 [Simulated I/O Window] Window

The above is the window displayed when the sample program supplied with this CPU board is used. For details on the sample program, refer to section 7.3, Sample Program.

Note: When using the [Simulated I/O Window] window, an interrupt handler must be prepared in the user program. For details on the interrupt handler, refer to section 7, Creation of User Interrupt Handler.

4.2.8 [Command Line] Window

The SH7729R CPU board does not guarantee the [Command Line] window operation; do not use the [Command Line] window.

5.1 Block Diagram

A block diagram of the CPU board is shown in figure 5.1.

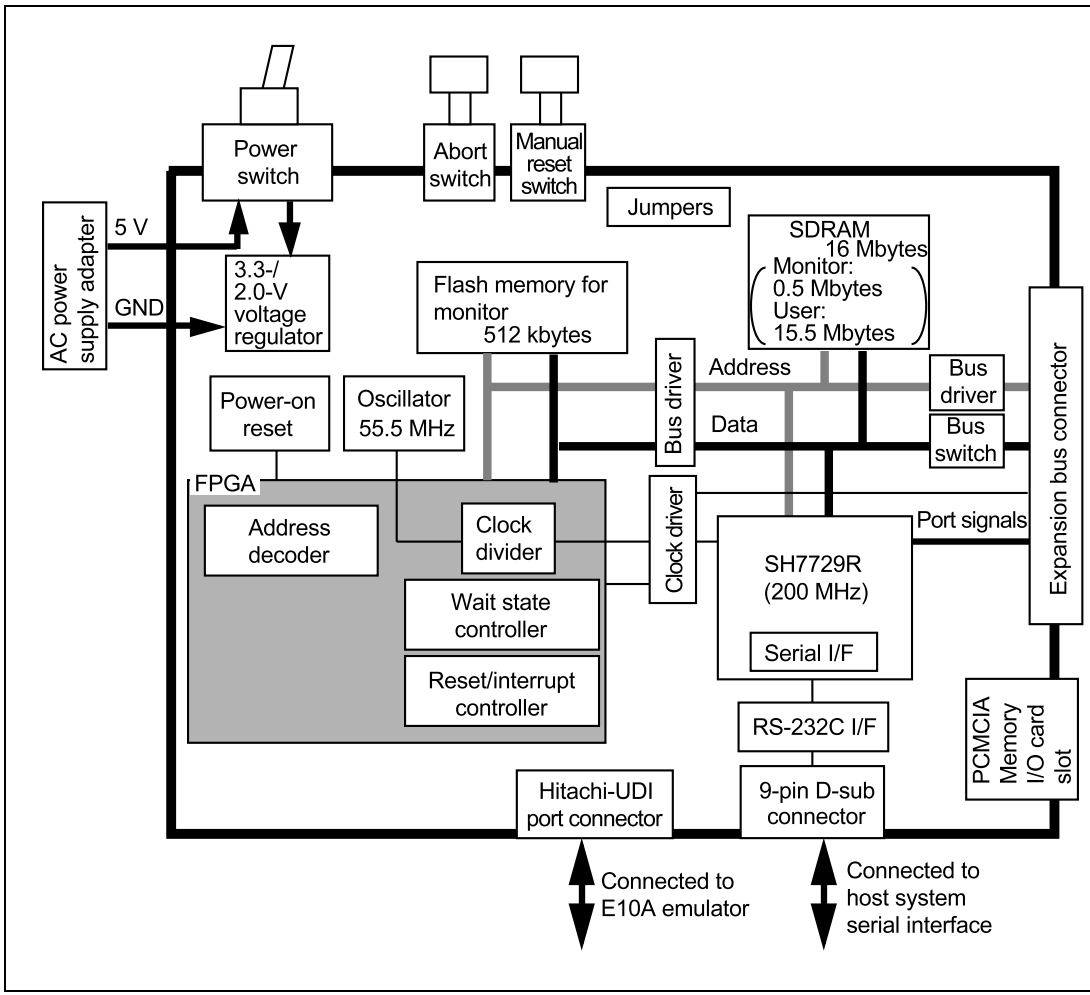


Figure 5.1 Block Diagram of the CPU board

5.2 Specifications

Table 5.1 lists the components of the CPU board.

Table 5.1 Specifications

Item		Specifications	
Microcomputer		SH7729R	Type name: HD6417729RHF200 Package: 208-pin HQFP
Operating frequency		CPU internal clock: 200 MHz (fixed) Bus clock: 66.7 or 33.3 MHz (switchover by jumpers)	
Endian		Little or big endian (switchover by jumpers)	
Memory	RAM	SDRAM	Capacity: 16 Mbytes Bus width: 32 bits Type number: HM5264165FTT-B60 x 2
	ROM (monitor program)	Flash memory	Capacity: 512 kbytes Bus width: 16 bits Type number: PA28F400B5-B60
Serial interface		One channel: Conforms to RS-232C Transfer rate: 57,600 or 115,200 bit/s (switchover by jumpers) Connector: 9-pin D-sub connector CPU board connector: DELC-J9PAF-20L9 manufactured by Japan Aviation Electronics Industry, Ltd. Maximum cable length: 3 m	

Table 5.1 Specifications (cont)

Item		Specifications
Expansion board interface		Three expansion connectors CN3 CPU board: 8817-180-170L manufactured by KEL CORPORATION User side: 8807-180-170L* manufactured by KEL CORPORATION Note: Gold-plated thickness (No mark: Flush, D: 0.25 µm or more) CN6 CPU board: 8931E-100-178S manufactured by KEL CORPORATION User side: 8925*-100-179 manufactured by KEL CORPORATION Note: Strain-relief (E: Yes, R: No) CN7 CPU board : 7620-6002SC manufactured by 3M Company. User side: 7920-7500SC manufactured by 3M Company.
PCMCIA		Memory card or I/O card x 2 slots
E10A emulator interface		CN8 CPU board type number: 10236-5202JL manufactured by 3M Company.
Switches		SW1: Manual reset switch SW2: Abort switch SW3: Power supply switch
External dimensions	Board	Width: 210 mm, Length: 150 mm
	Product	Width: 227 mm, Length: 167 mm, Height: 50 mm

5.3 Memory Map

Memory map of the CPU board is shown in figure 5.2. Each area of the CPU is allocated as follows:

Area 0: Monitor program area. Allocated to flash memory, on-board registers, and monitor input/output. Bus width is 16 bits fixed.

Area 1: CPU reserved area.

Area 2: Expansion bus area.

Area 3: 16 Mbytes of SDRAM is allocated. The first 15.5-Mbyte area is assigned to the user area and the remaining 0.5-Mbyte area is assigned to the monitor program work area. Bus width is 32 bits fixed.

Area 4: Expansion bus area.

Area 5: Expansion bus area.

Area 6: PCMCIA area.

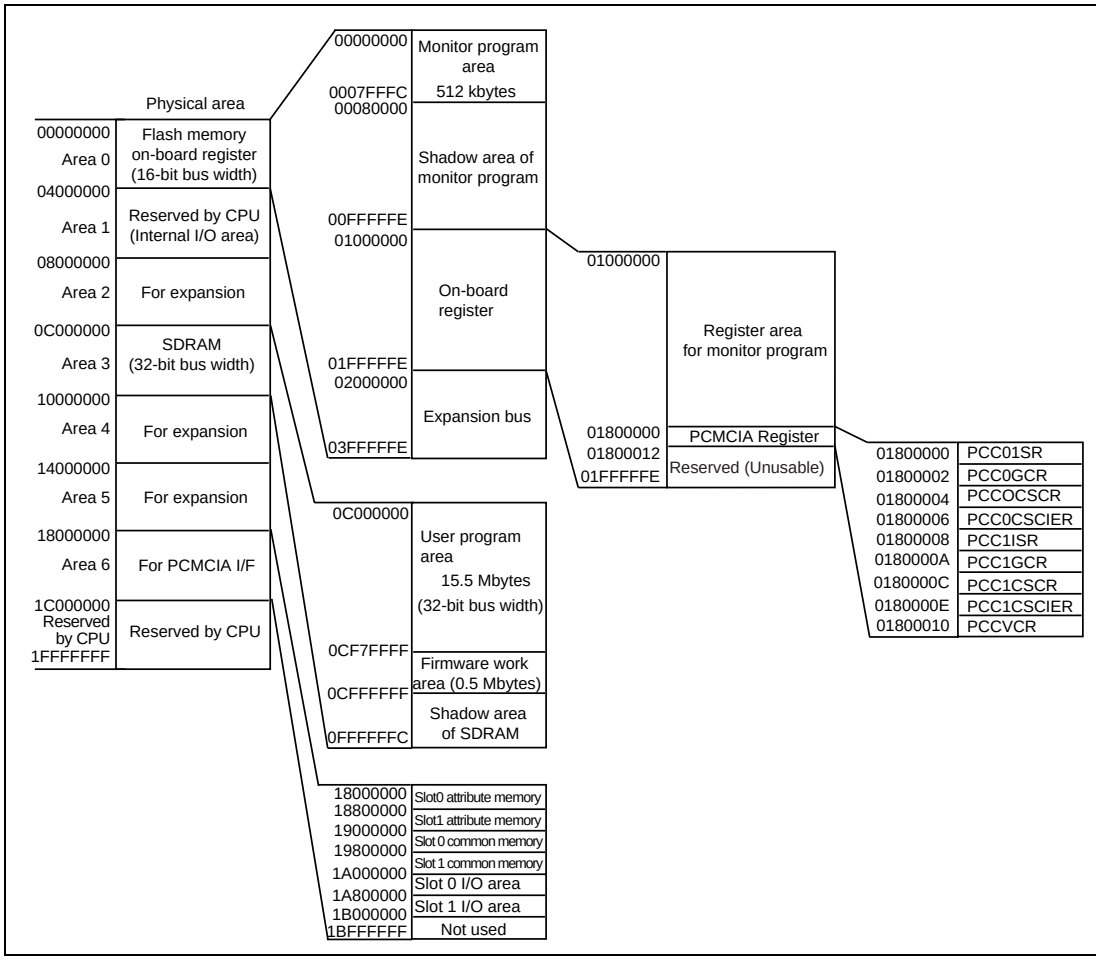


Figure 5.2 CPU Board Memory Map

5.4 External Interface

5.4.1 Serial Interface

⚠ WARNING

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

The CPU board has a serial interface (1 channel) that conforms to RS-232C and can interface with the host computer. An on-chip serial communication interface (SCI0) in the CPU is used for the interface between the host computer and the CPU board. Therefore, the users cannot use the SCI0. For the connector, a 9-pin D-sub connector is used. A baud rate of 57,600 bit/s or 115,200 bit/s can be selected. To set the baud rate, refer to section 2.8, Jumpers.

Table 5.2 lists the pin assignment for the serial interface connector. Table 5.3 shows the serial interface specifications. For details on serial interface cable connection, refer to section 2.4, Connecting Cables.

Table 5.2 Pin Assignment of the Serial Interface Connector

Pin No.	Signal Name	Description
1	Reserved	No connection
2	RXD	Received serial data
3	TXD	Transmitted serial data
4	DTR	Data terminal ready
5	SG	Signal ground
6	DSR	Data set ready
7	RTS	Request to send
8	CTS	Clear to send
9	Reserved	No connection

Table 5.3 Serial Interface Specifications

Item	Specifications
Synchronization method	Asynchronous method
Transfer rate	57,600 or 115,200 bit/s (can be switched with jumpers)
Bit configuration	Start bit: 1 bit, Stop bit: 1 bit, Parity: none, Data length: 8 bits
Controller	On-chip SCI (serial communication interface) in the SH7729R
Driver	LT1181ACSW (manufactured by LINEAR TECHNOLOGY CORP.)
Connectors	CPU board connector: DELC-J9PAF-20L9 manufactured by Japan Aviation Electronics Industry, Ltd.

Figure 5.3 shows the wiring connection between the host computer (IBM PC compatible machine) serial interface connector and the CPU board interface connector. Figure 5.4 shows the serial interface connector pin arrangement. A serial cable that matches the specifications shown in figures 5.3 and 5.4 must be used. For details on serial interface cable connection, refer to section 2.4, Connecting Cables.

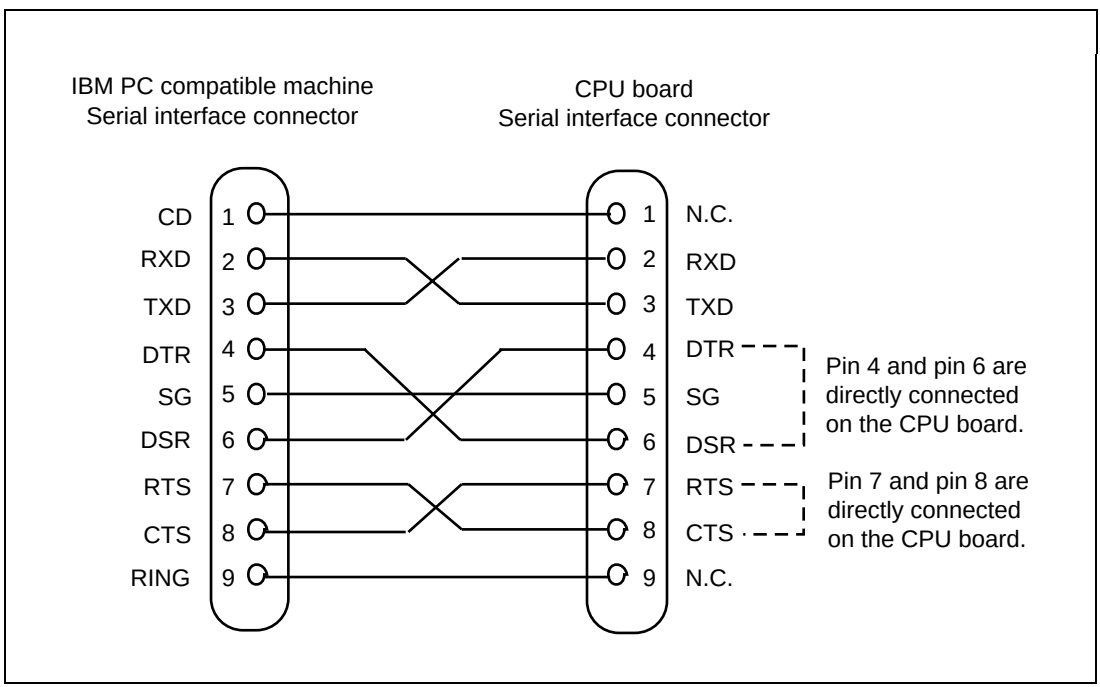


Figure 5.3 Connection to Host Computer

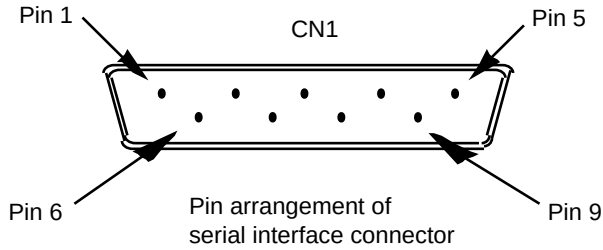


Figure 5.4 Serial Interface Connector Pin Arrangement

5.4.2 User Expansion Board Interface

The CPU board has a connector for the user expansion board interface.

⚠ WARNING

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

Figures 5.5 to 5.7 show the pin arrangement of expansion connectors CN3, CN6, and CN7. Tables 5.4 to 5.6 list the pin assignment of the expansion connectors CN3, CN6, and CN7, respectively. Refer to the CPU hardware manual for those pins that have no numbers in the pin function column. The pin signal level is 3.3 V, which is equivalent to that of the SH7729R CPU. For details on expansion board connection, refer to section 2.5, Connecting the User Expansion Board.

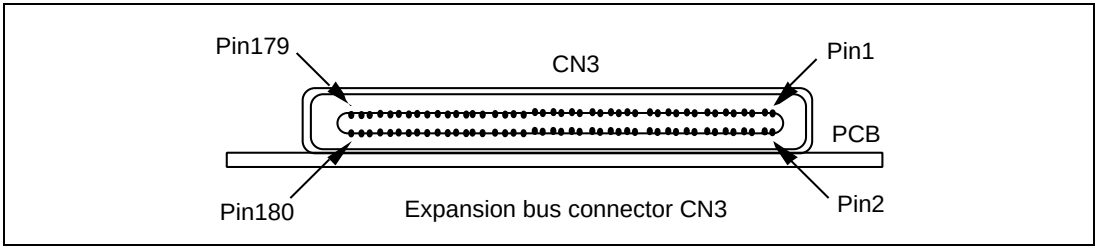


Figure 5.5 Expansion Bus CN3 Pin Arrangement

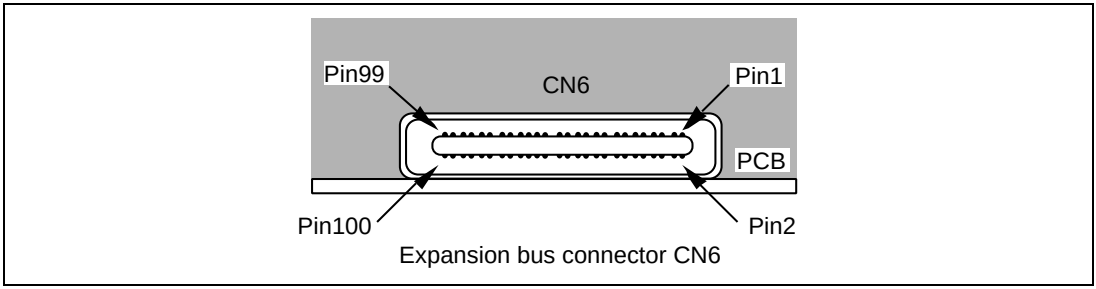


Figure 5.6 Expansion Bus CN6 Pin Arrangement

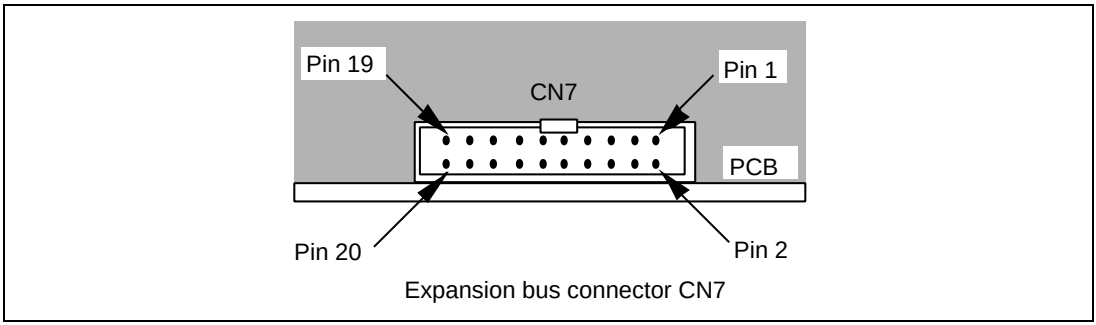


Figure 5.7 Expansion Bus CN7 Pin Arrangement

Table 5.4 Pin Assignment of Expansion Connector CN3

Pin No.	Pin Name	Pin Function	Pin No.	Pin Name	Pin Function	Pin No.	Pin Name	Pin Function
1	GND		26	D12	(1)	51	GND	
2	D0	(1)	27	GND		52	D25	(1)
3	CASL		28	D13	(1)	53	GND	
4	D1	(1)	29	GND		54	D26	(1)
5	GND		30	D14	(1)	55	GND	
6	D2	(1)	31	GND		56	D27	(1)
7	CASU/PTJ[3]	(10)	32	D15	(1)	57	GND	
8	D3	(1)	33	GND		58	D28	(1)
9	GND		34	D16	(1)	59	GND	
10	D4	(1)	35	GND		60	D29	(1)
11	PTJ[4]	(10)	36	D17	(1)	61	GND	
12	D5	(1)	37	GND		62	D30	(1)
13	GND		38	D18	(1)	63	GND	
14	D6	(1)	39	GND		64	D31	(1)
15	PTJ[5]	(10)	40	D19	(1)	65	GND	
16	D7	(1)	41	GND		66	A0	(2)
17	GND		42	D20	(1)	67	GND	
18	D8	(1)	43	GND		68	A1	(2)
19	GND		44	D21	(1)	69	GND	
20	D9	(1)	45	GND		70	A2	(2)
21	GND		46	D22	(1)	71	GND	
22	D10	(1)	47	GND		72	A3	(2)
23	GND		48	D23	(1)	73	GND	
24	D11	(1)	49	GND		74	A4	(2)
25	GND		50	D24	(1)	75	GND	

Notes: 1. Do not connect the Reserve pins.

2. Some signals are converted and input to the CPU or output to the expansion connector as follows:

(1) D[31:0]: See figure 5.11.
(2) A[25:0]: See figure 5.11.
(10) When used as a port, signals are limited to the output direction.

Table 5.4 Pin Assignment of Expansion Connector CN3 (cont)

Pin No.	Pin Name	Pin Function	Pin No.	Pin Name	Pin Function	Pin No.	Pin Name	Pin Function
76	A5	(2)	101	RAS3L		126	_RD	
77	GND		102	A18	(2)	127	GND	
78	A6	(2)	103	GND		128	Reserve	
79	GND		104	A19	(2)	129	GND	
80	A7	(2)	105	CKE		130	_CS2	
81	GND		106	A20	(2)	131	GND	
82	A8	(2)	107	Reserve		132	_CS4	
83	GND		108	A21	(2)	133	GND	
84	A9	(2)	109	Reserve		134	_CS5/_CE1A	
85	GND		110	A22	(2)	135	GND	
86	A10	(2)	111	_RSTOUT	(3)	136	_CS6/_CE1B	
87	GND		112	A23	(2)	137	GND	
88	A11	(2)	113	_RESETP		138	Reserve	
89	GND		114	A24	(2)	139	GND	
90	A12	(2)	115	GND		140	Reserve	
91	GND		116	A25	(2)	141	GND	
92	A13	(2)	117	GND		142	RD/_WR	
93	CS0		118	_WE0/DQMLL		143	GND	
94	A14	(2)	119	GND		144	_IRQOUT	
95	GND		120	_WE1/DQMLU		145	GND	
96	A15	(2)	121	GND		146	_BACK	(4)
97	_BS		122	_WE2/DQMUL		147	GND	
98	A16	(2)	123	GND		148	_CS3	
99	GND		124	_WE3/DQMUU		149	GND	
100	A17	(2)	125	GND		150	STATUS1	

Notes: 1. Do not connect the Reserve pins.

2. Some signals are converted and input to the CPU or output to the expansion connector as follows:

   (2) A[25:0]: See figure 5.11.

   (3) RSTOUT: A low-level pulse is output when the CPU is reset (at power on, when the manual reset switch is pressed, or when the reset signal is input from the expansion connector).

   (4) BACK: See figures 5.8, 5.9, 5.11 and 5.12.

Table 5.4 Pin Assignment of Expansion Connector CN3 (cont)

Pin No.	Pin Name	Pin Function	Pin No.	Pin Name	Pin Function	Pin No.	Pin Name	Pin Function
151	GND		161	GND		171	GND	
152	STATUS0		162	_WAIT	(8)	172	Reserve	
153	GND		163	GND		173	GND	
154	PTJ[1]	(10)	164	Reserve		174	Reserve	
155	GND		165	GND		175	GND	
156	_BREQ	(5)	166	Reserve		176	Reserve	
157	GND		167	GND		177	CKIO	(9)
158	TCLK	(6)	168	Reserve		178	GND	
159	GND		169	GND		179	GND	
160	_NMI	(7)	170	Reserve		180	GND	

Notes: 1. Do not connect the Reserve pins.

2. Some signals are converted and input to the CPU or output to the expansion connector as follows:

(5) BREQ: See figures 5.8, 5.9, 5.11 and 5.12.

(6) TCLK: Clock input for the timer. Jumper J3 selects either the CPU board clock or clock input from the expansion connector.

(7) NMI: Invalid and not reserved during monitor program execution.

(8) WAIT: The result of ORing the WAIT signals from the CPU board resource area 0 and area 6 (PCMCIA space) on access is input to the CPU.

(9) CKIO: Output only. Provides the same frequency as that set by jumpers J9 and J10. It is recommended that a terminal resistor is connected to the expansion board.
See figure 5.10.

(10) When used as a port, signals are limited to the output direction.

Table 5.5 Pin Assignment of Expansion Connector CN6

Pin No.	Pin Name	Pin Function	Pin No.	Pin Name	Pin Function	Pin No.	Pin Name	Pin Function
1	PTE[7]	(2)	20	_CE2A/PTE[4]		39	_DREQ1/PTD[6]	
2	GND		21	GND		40	GND	
3	_CAS2L/ PTE[6]		22	_RAS2U/PTE[1]		41	DACK0/PTD[5]	
4	GND		23	ADTRG/PTH[5]		42	GND	
5	_CE2B/PTE[5]		24	GND		43	DACK1/PTD[7]	
6	GND		25	PINT[15]/PTF[7]	(2)	44	GND	
7	_CAS2H/ PTE[3]		26	PINT[14]/PTF[6]	(2)	45	_CA	
8	_RESETM		27	PINT[13]/PTF[5]	(2)	46	GND	
9	GND		28	PINT[12]/PTF[4]	(2)	47	DRAK0/PTD[1]	
10	IOIS16/PTG[7]		29	_iRLS[3]/ PINT[11]/PTF[3]		48	GND	
11	PTG[6]		30	_iRLS[2]/ PINT[10]/PTF[2]		49	DRAK1/PTD[0]	
12	PTG[5]	(2)	31	_iRLS[1]/ PINT[9]/PTF[1]		50	GND	
13	PTG[4]		32	_iRLS[0]/ PINT[8]/PTF[0]		51	MCS[4]/PINT[OKI]/ PTC[OKI]	
14	PTG[3]	(2)	33	PTE[0]	(2)	52	GND	
15	PTG[2]	(2)	34	GND		53	MCS[5]/PINT[5]/ PTC[5]	
16	PTG[1]	(2)	35	GND		54	CS0	
17	PTG[0]	(2)	36	GND		55	MCS[6]/PINT[6]/ PTC[6]	
18	_RAS3U/ PTE[2]		37	_DREQ0/ PTD[4]		56	GND	
19	GND		38	GND		57	MCS[7]/PINT[7]/ PTC[7]	

Note: Some signals are converted and input to the CPU or output to the expansion connector as follows:

(2) With the E10A emulator connected, these signal pins cannot be used.

Table 5.5 Pin Assignment of Expansion Connector CN6 (cont)

Pin No.	Pin Name	Pin Function	Pin No.	Pin Name	Pin Function	Pin No.	Pin Name	Pin Function
58	GND		73	TXD1/SCPT[2]		88	GND	
59	MCS[0]/PINT[0]/ PTC[0]		74	GND		89	IRQ0	(1)
60	GND		75	SCK1/SCPT[3]		90	GND	
61	MCS[1]/PINT[1]/ PTC[1]		76	GND		91	IRQ1	(1)
62	GND		77	RXD2/SCPT[4]		92	GND	
63	MCS[2]/PINT[2]/ PTC[2]		78	GND		93	IRQ2	(1)
64	GND		79	TXD2/SCPT[4]		94	GND	
65	MCS[3]/PINT[3]/ PTC[3]		80	GND		95	IRQ3	(1)
66	GND		81	SCK2/SCPT[5]		96	GND	
67	_Wakeup/ PTD[3]		82	GND		97	IRQ4	(1)
68	GND		83	RTS2/SCPT[6]		98	GND	
69	RESETOUT/ PTD[2]		84	GND		99	Reserve	
70	GND		85	CTS2/SCPT[7]		100	GND	
71	RXD1/SCPT[2]		86	GND				
72	GND		87	PTH[6]				

Notes: 1. Do not connect the Reserve pin.

2. Some signals are converted and input to the CPU or output to the expansion connector as follows:

(1) IRQ[4:0]: Invalid and not reserved during monitor program execution.

Table 5.6 Pin Assignment of Expansion Connector CN7

Pin No.	Pin Name	Pin Function	Pin No.	Pin Name	Pin Function
1	Reserve		11	AN[4]/PTL[4]	
2	AVss		12	AVss	
3	AN[0]/PTL[0]		13	AN[5]/PTL[5]	
4	AVss		14	AVss	
5	AN[1]/PTL[1]		15	AN[6]/DA[1]/PTL[6]	
6	AVss		16	AVss	
7	AN[2]/PTL[2]		17	AN[7]/DA[0]/PTL[7]	
8	AVss		18	AVss	
9	AN[3]/PTL[3]		19	Reserve	
10	AVss		20	AVss	

Note: Do not connect the Reserve pins.

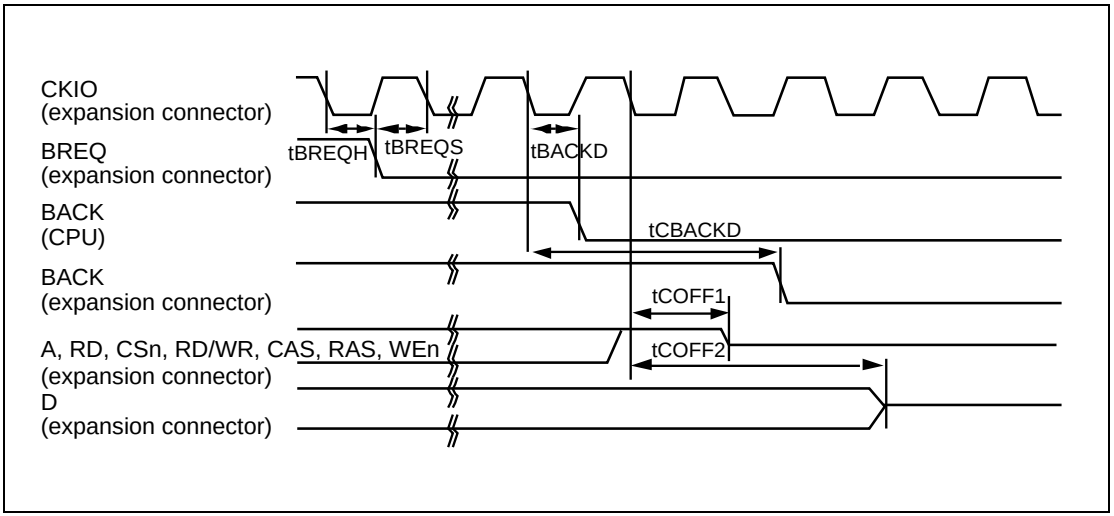


Figure 5.8 Bus Release Timing (1)

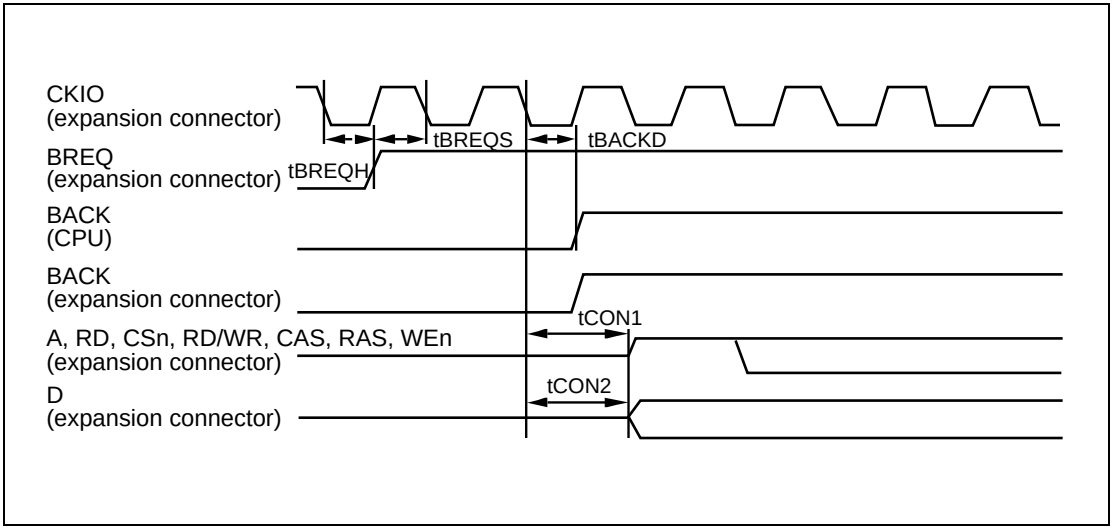


Figure 5.9 Bus Release Timing (2)

Table 5.7 AC Specifications

Parameter	Minimum	Maximum
tBREQH	=tBREQH	=tBREQH
tBREQS	=tBREQS	=tBREQS
tBACKD	=tBACKD	=tBACKD
tCBACKD	—	25.5 ns
tCOFF1	—	29.7 ns
tCOFF2	—	30.1 ns
tCON1	—	29.7 ns
tCON2	—	30.1 ns

Note: =: Equivalent to CPU AC specifications

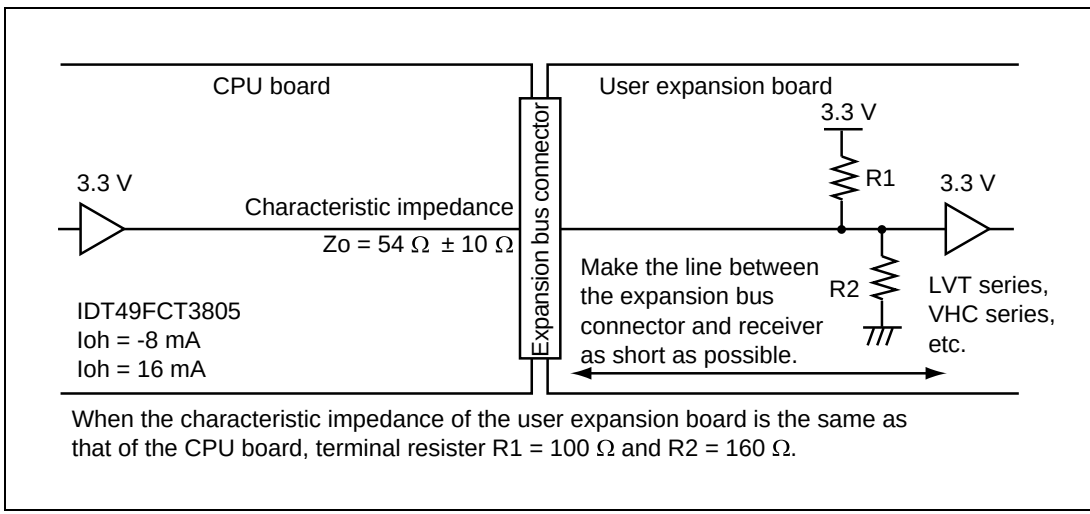


Figure 5.10 Example of CKIO Terminal Resistor Connection

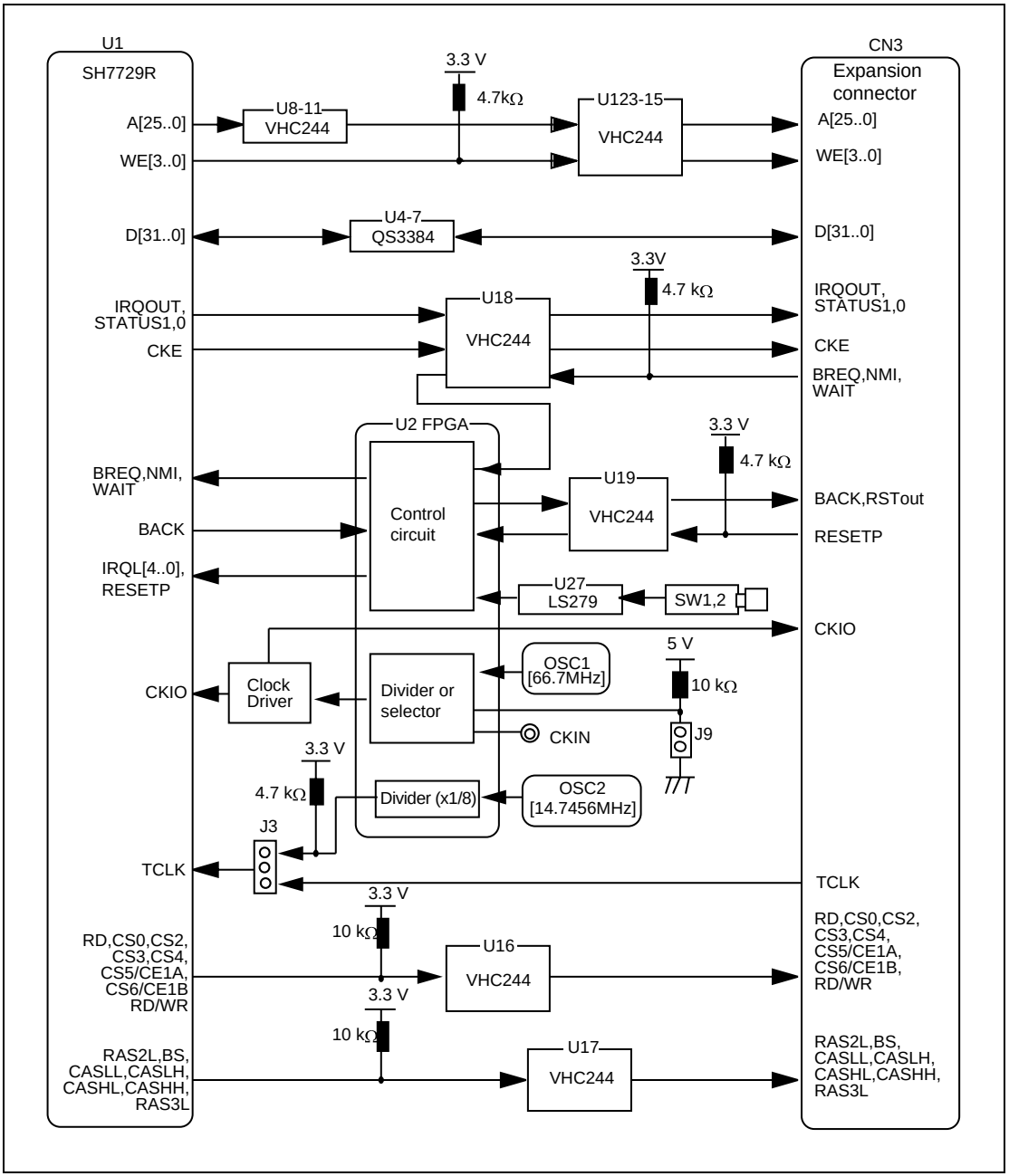


Figure 5.11 Configuration of User Expansion Board Interface Circuit (CN3)

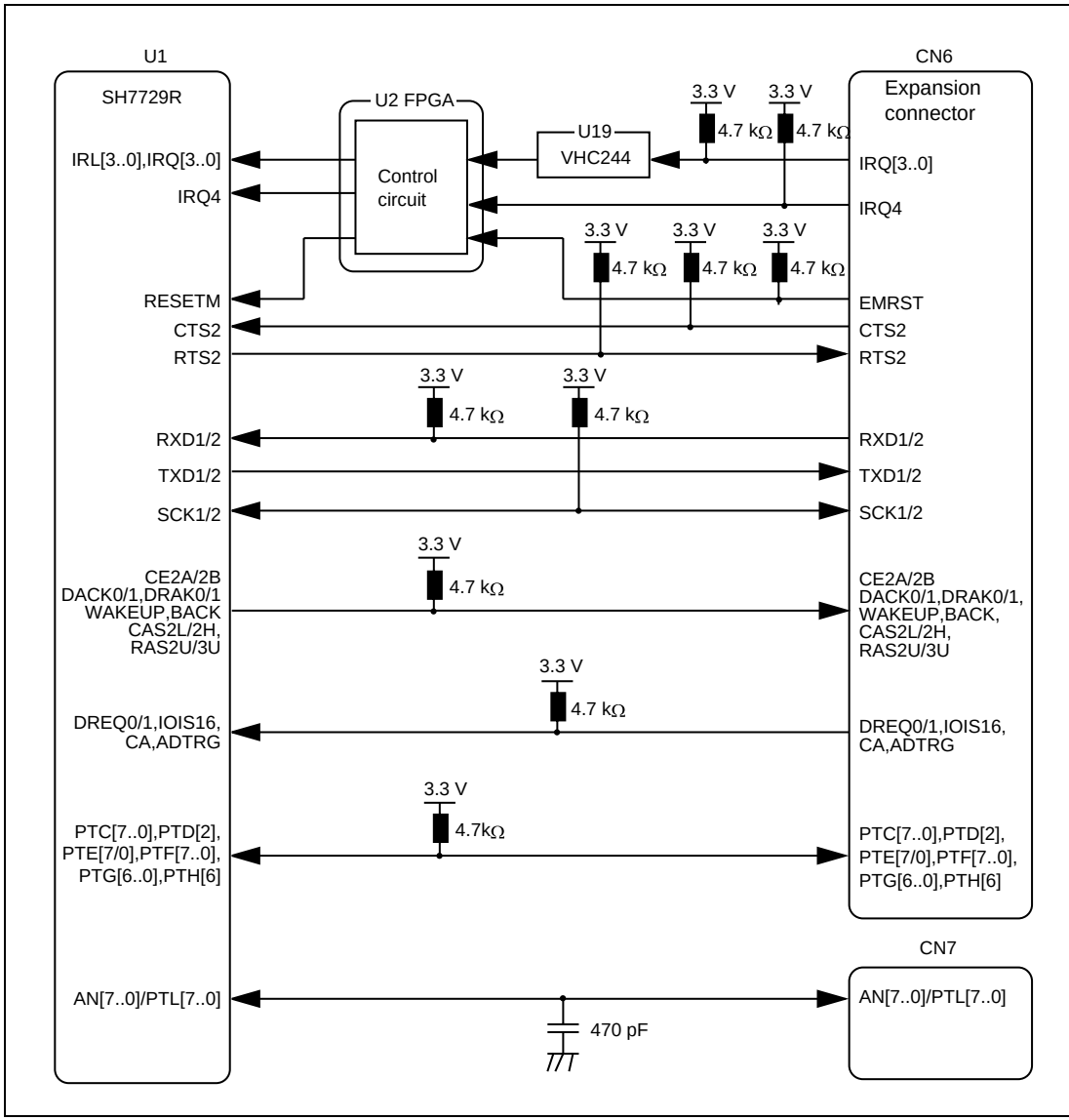


Figure 5.12 Configuration of User Expansion Board Interface Circuits (CN6, CN7)

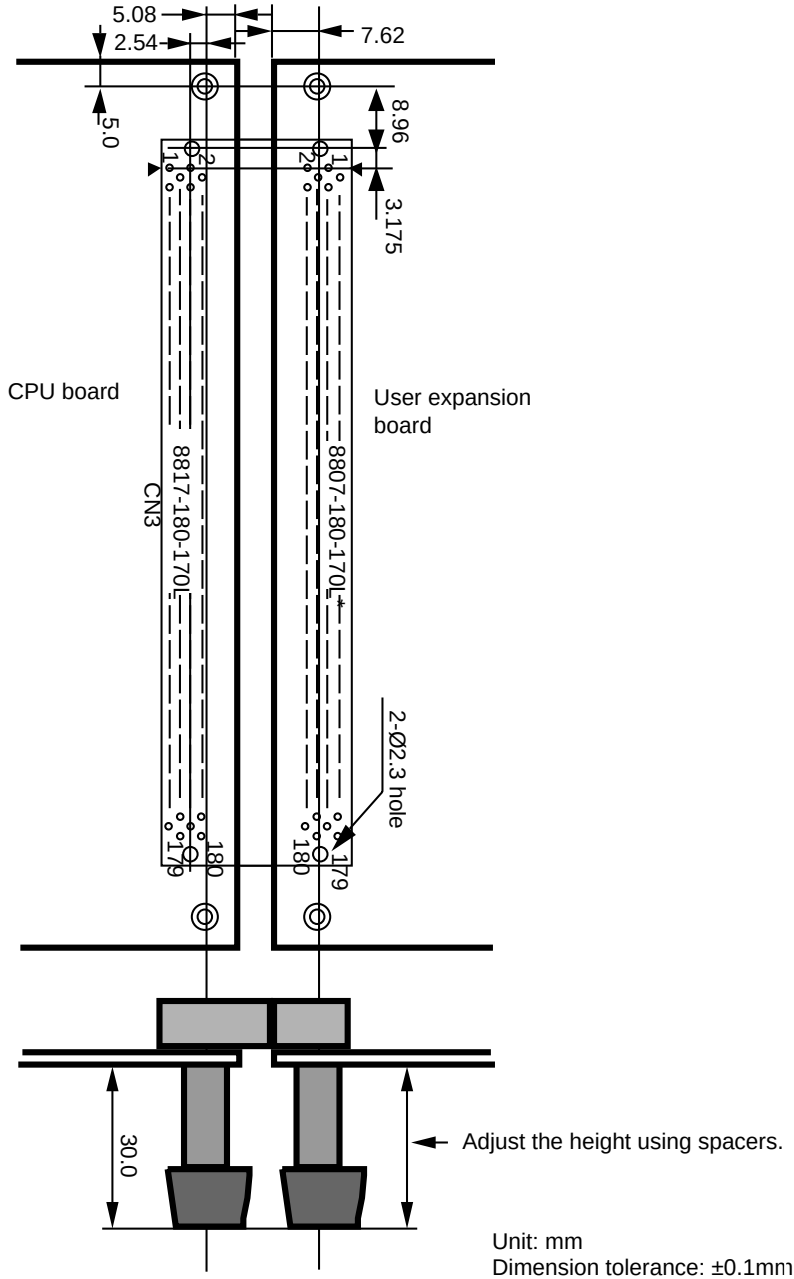


Figure 5.13 Connection-Dimension Specifications of User Expansion Board (CN3)

5.4.3 PCMCIA Interface

⚠ WARNING

Install the PCMCIA driver in the CPU board before inserting or removing a card while the power is on. Failure to do so will damage the card.

Overview:

This board is provided with two slots for PCMCIA type 1 and type 2 cards. Both slots can be used with either I/O cards or with memory cards. Card voltages of 3.3 V and 5 V are supported.

This CPU board is not provided with PCMCIA drivers. When using the PCMCIA slots, you must provide your own driver software.

Figure 5.14 shows the pin arrangements for PCMCIA slot 0 and slot 1. Pin assignments for slot 0 and slot 1 are listed in tables 5.8 and 5.9 respectively.

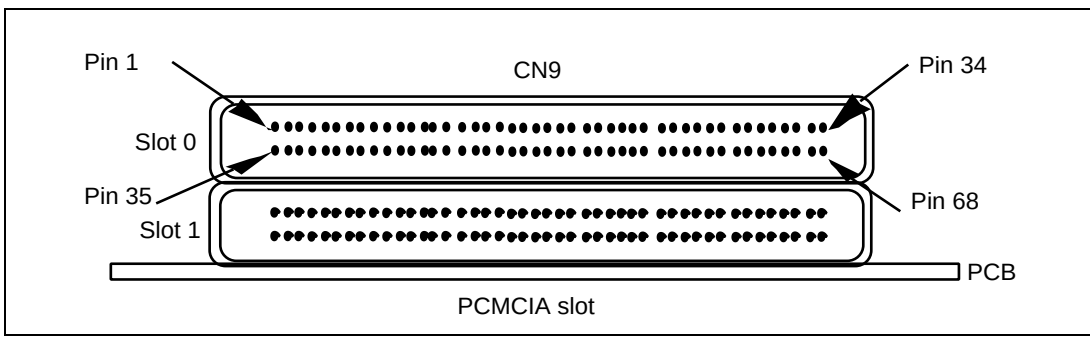


Figure 5.14 PCMCIA Slot 0 and Slot 1 Pin Arrangements

Table 5.8 PCMCIA Card Slot 0 (CN9) Pin Assignments

Pin No.	Signal Name	Pin No.	Signal Name	Pin No.	Signal Name	Pin No.	Signal Name
1	GND	18	VPPA	35	GND	52	VPPA
2	PCC0DA3	19	PCC0AD16	36	P0CD1	53	PCC0AD22
3	PCC0DA4	20	PCC0AD15	37	PCC0DA11	54	PCC0AD23
4	PCC0DA5	21	PCC0AD12	38	PCC0DA12	55	PCC0AD24
5	PCC0DA6	22	PCC0AD7	39	PCC0DA13	56	PCC0AD25
6	PCC0DA7	23	PCC0AD6	40	PCC0DA14	57	P0VS2
7	P0CE1	24	PCC0AD5	41	PCC0DA15	58	P0RESET
8	PCC0AD10	25	PCC0AD4	42	P0CE2	59	P0WAIT
9	P0RD	26	PCC0AD3	43	P0VS1	60	NC
10	PCC0AD11	27	PCC0AD2	44	P0WE2	61	P0REG
11	PCC0AD9	28	PCC0AD1	45	P0WE3	62	P0BVD2
12	PCC0AD8	29	PCC0AD0	46	PCC0AD17	63	P0BVD1
13	PCC0AD13	30	PCC0DA0	47	PCC0AD18	64	PCC0DA8
14	PCC0AD14	31	PCC0DA1	48	PCC0AD19	65	PCC0DA9
15	P0WE1	32	PCC0DA2	49	PCC0AD20	66	PCC0DA10
16	P0RDY	33	P0WP	50	PCC0AD21	67	P0CD2
17	VCCA	34	GND	51	VCCA	68	GND

Table 5.9 PCMCIA Card Slot 1 (CN9) Pin Assignments

Pin No.	Signal Name	Pin No.	Signal Name	Pin No.	Signal Name	Pin No.	Signal Name
1	GND	18	VPPB	35	GND	52	VPPB
2	PCC1DA3	19	PCC1AD16	36	P1CD1	53	PCC1AD22
3	PCC1DA4	20	PCC1AD15	37	PCC1DA11	54	PCC1AD23
4	PCC1DA5	21	PCC1AD12	38	PCC1DA12	55	PCC1AD24
5	PCC1DA6	22	PCC1AD7	39	PCC1DA13	56	PCC1AD25
6	PCC1DA7	23	PCC1AD6	40	PCC1DA14	57	P1VS2
7	P1CE1	24	PCC1AD5	41	PCC1DA15	58	P1RESET
8	PCC1AD10	25	PCC1AD4	42	P1CE2	59	P1WAIT
9	P1RD	26	PCC1AD3	43	P1VS1	60	NC
10	PCC1AD11	27	PCC1AD2	44	P1WE2	61	P1REG
11	PCC1AD9	28	PCC1AD1	45	P1WE3	62	P1BVD2
12	PCC1AD8	29	PCC1AD0	46	PCC1AD17	63	P1BVD1
13	PCC1AD13	30	PCC1DA0	47	PCC1AD18	64	PCC1DA8
14	PCC1AD14	31	PCC1DA1	48	PCC1AD19	65	PCC1DA9
15	P1WE1	32	PCC1DA2	49	PCC1AD20	66	PCC1DA10
16	P1RDY	33	P1WP	50	PCC1AD21	67	P1CD2
17	VCCB	34	GND	51	VCCB	68	GND

The CPU board PCMCIA interface has the following features.

- IRQ control: PCMCIA interrupts are input to the CPU as IRQ1.
- PCMCIA control: Two PCMCIA slots are supported. I/O cards or memory cards can be used in both slots.
- Card voltage: Both 3.3 V and 5 V are supported.
- PCMCIA area: Both slots use area 6. Because area 6 is further subdivided into eight areas, each area is allocated 8 Mbytes. PCMCIA supports up to 64 Mbytes in each area, and so the CPU board is provided with a bank-switching register corresponding to upper addresses A25 to A23.
  Access procedure is as follows.

  (1) Upper addresses A25 to A23 are written to GCR (General Control Register) bits 2 to 0 of the slot to be accessed (0/1).

  (2) An area 6 window is accessed according to the card type inserted. Allocation of the area 6 space is as shown in figure. 5.15.

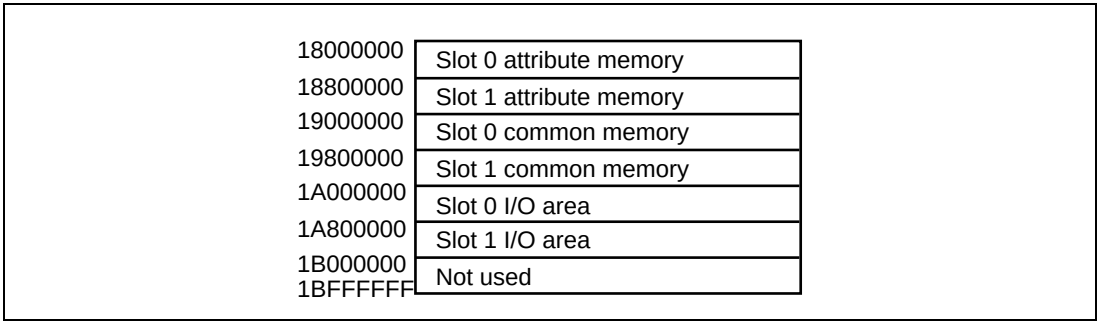


Figure 5.15 Area 6 Allocation

Explanation of Registers:

Table 5.10 lists the addresses and gives explanations of registers related to the PCMCIA interface.

Table 5.10 PCMCIA Registers and Addresses

Register Name	Formal Name	R/W	Address
PCC0ISR	PCC0 interface status register	R	H'01800000
PCC0GCR	PCC0 general control register	R/W	H'01800002
PCC0CSCR	PCC0 card status change register	R/W	H'01800004
PCC0CSCIER	PCC0 card status change interrupt enable register	R/W	H'01800006
PCC1ISR	PCC1 interface status register	R	H'01800008
PCC1GCR	PCC1 general control register	R/W	H'0180000A
PCC1CSCR	PCC1 card status change register	R/W	H'0180000C
PCC1CSCIER	PCC1 card status change interrupt enable register	R/W	H'0180000E
PCCVCR	PCC voltage control register	R/W	H'01800010

[1] PCC0 interface status register (PCC0ISR)

Bit	7	6	5	4	3	2	1	0
Name	P0RDY	P0WP	P0VS2	P0VS1	P0CD2	P0CD1	P0BVD2	P0BVD1
R/W	R	R	R	R	R	R	R	R
Initial value	—	—	—	—	—	—	—	—

This register monitors the values of each input signal in channel 0. Here 1=High and 0=Low.

[2] PCC0 general control register (PCC0GCR)

Bit	7	6	5	4	3	2	1	0
Name	P0DRVE	P0RES	P0PCCT	Not used	Not used	P0PA25	P0PA24	P0PA23
R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

This register controls the bus buffer, reset and other operations for channel 0. It also controls the uppermost three bits of the channel 0 address used to access the 64 Mbytes from the 8-Mbyte window.

Bit 7: P0DRVE (external buffer control)

0: PCC0DRV signal driven high (buffer disabled)
1: PCC0DRV signal driven low (buffer enabled)

Bit 6: P0RES (reset control)

0: PCC0RES signal driven low (normal operation)
1: PCC0RES signal driven high (reset)

Bit 5: P0PCCT (PCMCIA card type)

0: Slot 0 card handled as memory card
1: Slot 0 card handled as I/O card

Bits 4-3: Not used

Always read as 0

Bits 2-0: P0PA25-23 (used for upper address control, window switching)

0: Corresponding address line driven low
1: Corresponding address line driven high

[3] PCC0 card status change register (PCC0CSCR)

Bit	7	6	5	4	3	2	1	0
Name	P0SCDI	Not used	P0IREQ	P0SC	P0CDC	P0RC	P0BW	P0BD
R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

This register contains flags used to monitor changes in the status of channel 0. By reading this register after an interrupt (IRQ1) is received, the cause of the interrupt can be identified.

Bit 7: P0SCDI (software card detection interrupt)

When writing data via software, an interrupt is generated. This bit can be set freely, but as in the case of a card detection interrupt, the interrupt itself is masked by PCC0CSCIER bit 3 (P0CDE).

0: Interrupt not generated
1: Slot 0 card detection interrupt generated

Bit 6: Not used

Always read as 0

Bit 5: P0IREQ (IREQ interrupt)

In cases where the card in slot 0 is an I/O card, when an interrupt is generated by PCC0RDY/IREQ, this bit is set to 1 (in the case of a memory card, this bit is always 0). Conditions for setting this bit conform to the settings for bits 6 and 5 of PCC0CSCIER (IREQE1-0) (interrupt disabled, level mode, rising-edge mode, falling-edge mode). In the case of an edge mode, the interrupt can be cleared by writing 0 to this bit.

0: No IREQ interrupt request
1: IREQ interrupt request generated for the slot 0 I/O card

Bit 4: P0SC (STSCHG interrupt)

In cases where the slot 0 card is an I/O card, when an interrupt is generated by PCC0BVD1/STSCHG, this bit is set to 1 (in the case of a memory card, it is always 0). The condition for setting the bit is a transition of the STSCHG pin from high to low level. The bit is cleared by writing 0 to it. The interrupt can be masked by PCC0CSCIER bit 4.

0: No STSCHG interrupt request
1: STSCHG interrupt request by the slot 0 I/O card

Bit 3: P0CDC (card detection interrupt)

This indicates that either PCC0CD2 or PCC0CD1 has changed (a card has been inserted or removed). The condition for setting the bit is a change in signal level; it is cleared by writing 0 to it. The interrupt can be masked by PCC0CSCIER bit 3.

0: No change in PCC0CD2/PCC0CD1
1: Either PCC0CD2 or PCC0CD1 has changed

Bit 2: P0RC (ready change interrupt)

In cases where the card of slot 0 is a memory card, this bit is set to 1 when the PCC0RDY signal changes (in cases where the card is an I/O card, this bit is always 0). The condition for setting the bit is when the RDY/BSY pin changes from low to high (there is a transition from a busy state to a ready state); the bit is cleared by writing 0 to it. The interrupt can be masked by PCC0CSCIER bit 2.

0: No change in RDY/BSY
1: RDY/BSY changed from low to high level

Bit 1: P0BW (battery voltage low interrupt; data is preserved, but the battery must be replaced)

In cases where the card of slot 0 is a memory card, this bit is set to 1 when the PCC0BVD2/BVD1 signal indicates a drop in battery voltage (in the case of an I/O card, this bit is always 0). The condition for setting this bit is when PCC0BVD2=Low and in addition PCC0BVD1=High; the bit is cleared by writing 0 to it. The interrupt can be masked by PCC0CSCIER bit 1.

0: Memory card of slot 0 does not indicate low battery voltage
1: Memory card of slot 0 indicates low battery voltage

Bit 0: P0BD (battery replacement interrupt)

In cases where the card of slot 0 is a memory card, this bit is set to 1 when the PCC0BVD2/BVD1 signal indicates the battery needs to be replaced (in the case of an I/O card, this bit is always 0). The conditions for setting this bit are either that PCC0BVD2=High and PCC0BVD1=Low, or else that PCC0BVD2=Low and PCC0BVD1=Low. This bit is cleared by writing 0 to it. The interrupt can be masked by PCC0CSCIER bit 0.

0: Battery of slot 0 memory card does not need replacement
1: Battery of slot 0 memory card needs replacement; integrity of data in memory card not guaranteed

[4] PCC0 card status change interrupt enable register (PCC0CSCIER)

Bit	7	6	5	4	3	2	1	0
Name	P0CRE	P0IREQE1	P0IREQE0	P0SCE	P0CDE	P0RE	P0BWE	P0BDE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

This register masks status change interrupts for channel 0. An interrupt request is generated when the corresponding bits of PCC0CSCR and this register are both 1.

Bit 7: P0CRE (PCC0GCR reset enable)

When insertion of a card into slot 0 is detected (CD1 changes from high to low, or CD2 changes from high to low), this bit determines whether or not PCC0GCR is initialized.

0: GCR is not initialized even when card insertion is detected
1: GCR is initialized when card insertion is detected

Bits 6-5: P0IREQE1-0 (IREQ interrupt enable)

In cases where the card in slot 0 is an I/O card, these bits enable and disable IREQ interrupts and set the interrupt mode. Before changing these bits, PCC0CSCR bit 5 (P0IREQ) should first be cleared.

In the case of a memory card, operations on these bits are invalid.

Bit 6 IREQ1	Bit 5 IREQ0	Settings
0	0	Interrupt disabled. P0IREQ does not change, regardless of IREQ signal level.
0	1	Level mode interrupt. P0IREQ is set when the IREQ signal is at low level, and an interrupt request is generated.
1	0	Edge mode interrupt. P0IREQ is set at the IREQ signal falling edge, and an interrupt request is generated.
1	1	Edge mode interrupt. P0IREQ is set at the IREQ signal rising edge, and an interrupt request is generated.

Bit 4: P0SCE (STSCHG interrupt enable)

In cases where the card in slot 0 is an I/O card, this bit enables and disables PCC0BVD1/STSCHG interrupts. An interrupt request is generated when PCC0CSCR bit 4 is set and this bit is 1. When the card is a memory card, operations on this bit are invalid.

0: STSCHG interrupt disabled
1: STSCHG interrupt enabled

Bit 3: P0CDE (card detection interrupt enable)

Enables and disables PCC0CD2 and PCC0CD1 interrupts. An interrupt request is generated when PCC0CSCR bit 3 is set and this bit is 1.

0: PCC0CD2/PCC0CD1 interrupt disabled
1: PCC0CD2/PCC0CD1 interrupt enabled

Bit 2: P0RE (ready change interrupt enable)

In cases where the card in slot 0 is a memory card, this bit enables and disables PCC0RDY interrupts. An interrupt request is generated when PCC0CSCR bit 2 is set and this bit is 1. In the case of an I/O card, operations on this bit are invalid.

0: RDY/BSY interrupt disabled
1: RDY/BSY interrupt enabled

Bit 1: P0BWE (battery voltage low interrupt enable)

In cases where the card in slot 0 is a memory card, this bit enables and disables battery voltage-low interrupts. An interrupt request is generated when PCC0CSCR bit 1 is set and this bit is 1. In the case of an I/O card, operations on this bit are invalid.

0: Battery voltage-low interrupt disabled
1: Battery voltage-low interrupt enabled

Bit 0: P0BDE (battery replacement interrupt enable)

In cases where the card in slot 0 is a memory card, this bit enables and disables battery replacement interrupts. An interrupt request is generated when PCC0CSCR bit 0 is set and this bit is 1. In the case of an I/O card, operations on this bit are invalid.

0: Battery replacement interrupt disabled
1: Battery replacement interrupt enabled

[5] PCC1 interface status register (PCC1ISR)

Bit	7	6	5	4	3	2	1	0
Name	P1RDY	P1WP	P1VS2	P1VS1	P1CD2	P1CD1	P1BVD2	P1BVD1
R/W	R	R	R	R	R	R	R	R
Initial value	-	-	-	-	-	-	-	-

This register monitors the values of channel 1 input signals. Here 1=high and 0=low level.

[6] PCC1 general control register (PCC1GCR)

Bit	7	6	5	4	3	2	1	0
Name	P1DRVE	P1RES	P1PCCT	Not used	Not used	P1PA25	P1PA24	P1PA23
R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

This register controls the bus buffer, reset and other operations for channel 1. It also controls the uppermost three bits of the channel 1 address used to access the 64 Mbytes from the 8-Mbyte window.

Bit 7: P1DRVE (external buffer control)

0: PCC1DRV signal driven high (buffer disabled)
1: PCC1DRV signal driven low (buffer enabled)

Bit 6: P1RES (reset control)

0: PCC1RES signal driven low (normal operation)
1: PCC1RES signal driven high (reset)

Bit 5: P1PCCT (PCMCIA card type; interrupt processing varies depending on whether it is memory card or I/O card)

0: Slot 1 card handled as memory card
1: Slot 1 card handled as I/O card

Bits 4-3: Not used

Always read as 0

Bits 2-0: P1PA25-23 (used for upper address control, window switching)

0: Corresponding address line driven low
1: Corresponding address line driven high

[7] PCC1 card status change register (PCC1CSCR)

Bit	7	6	5	4	3	2	1	0
Name	P1SCDI	Not used	P1IREQ	P1SC	P1CDC	P1RC	P1BW	P1BD
R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

This register contains flags used to monitor changes in the status of channel 1. By reading this register after an interrupt (IRQ1) is received, the cause of the interrupt can be identified.

Bit 7: P1SCDI (software card detection interrupt)

When writing data via software, an interrupt is generated. This bit can be set freely, but as in the case of a card detection interrupt, the interrupt itself is masked by PCC1CSCIER bit 3 (P1CDE).

0: Interrupt not generated
1: Slot 1 card detection interrupt generated

Bit 6: Not used

Always read as 0

Bit 5: P1IREQ (IREQ interrupt)

In cases where the card in slot 1 is an I/O card, when an interrupt is generated by PCC1RDY/IREQ, this bit is set to 1 (in the case of a memory card, this bit is always 0). Conditions for setting this bit conform to the settings for bits 6 and 5 of PCC1CSCIER (IREQE1-0) (interrupt disabled, level mode, rising-edge mode, falling-edge mode). In the case of an edge mode, the interrupt can be cleared by writing 0 to this bit.

0: No IREQ interrupt request
1: IREQ interrupt request generated for the slot 1 I/O card

Bit 4: P1SC (STSCHG interrupt)

In cases where the slot 1 card is an I/O card, when an interrupt is generated by PCC1BVD1/STSCHG, this bit is set to 1 (in the case of a memory card, it is always 0). The condition for setting the bit is a transition of the STSCHG pin from high to low level. The bit is cleared by writing 0 to it. The interrupt can be masked by PCC1CSCIER bit 4.

0: No STSCHG interrupt request
1: STSCHG interrupt request by the slot 1 I/O card

Bit 3: P1CDC (card detection interrupt)

This indicates that either PCC1CD2 or PCC1CD1 has changed (a card has been inserted or removed). The condition for setting the bit is a change in signal level; it is cleared by writing 0 to it. The interrupt can be masked by PCC1CSCIER bit 3.

0: No change in PCC1CD2/PCC1CD1
1: Either PCC1CD2 or PCC1CD1 has changed

Bit 2: P1RC (ready change interrupt)

In cases where the card of slot 1 is a memory card, this bit is set to 1 when the PCC1RDY signal changes (in cases where the card is an I/O card, this bit is always 0). The condition for setting the bit is when the RDY/BSY pin changes from low to high (there is a transition from a busy state to a ready state); the bit is cleared by writing 0 to it. The interrupt can be masked by PCC1CSCIER bit 2.

0: No change in RDY/BSY
1: RDY/BSY changed from low to high level

Bit 1: P1BW (battery voltage low interrupt; data is preserved, but the battery must be replaced)

In cases where the card of slot 1 is a memory card, this bit is set to 1 when the PCC1BVD2/BVD1 signal indicates a drop in battery voltage (in the case of an I/O card, this bit is always 0). The condition for setting this bit is when PCC1BVD2=Low and in addition PCC1BVD1=High; the bit is cleared by writing 0 to it. The interrupt can be masked by PCC1CSCIER bit 1.

0: Memory card of slot 1 does not indicate low battery voltage
1: Memory card of slot 1 indicates low battery voltage

Bit 0: P1BD (battery replacement interrupt)

In cases where the card of slot 1 is a memory card, this bit is set to 1 when the PCC1BVD2/BVD1 signal indicates the battery needs to be replaced (in the case of an I/O card, this bit is always 0). The conditions for setting this bit are either that PCC1BVD2=High and PCC1BVD1=Low, or else that PCC1BVD2=Low and PCC1BVD1=Low. This bit is cleared by writing 0 to it. The interrupt can be masked by PCC1CSCIER bit 0.

0: Battery of slot 1 memory card does not need replacement
1: Battery of slot 1 memory card needs replacement; integrity of data in memory card not guaranteed

[8] PCC1 CSC interrupt enable register (PCC1CSCIER)

Bit	7	6	5	4	3	2	1	0
Name	P1CRE	P1IREQE1	P1IREQE0	P1SCE	P1CDE	P1RE	P1BWE	P1BDE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

This register masks status change interrupts for channel 1. An interrupt request is generated when the corresponding bits of PCC1CSCR and this register are both 1.

Bit 7: P1CRE (PCC1GCR reset enable)

When insertion of a card into slot 1 is detected (CD1 changes from high to low, or CD2 changes from high to low), this bit determines whether or not PCC1GCR is initialized.

0: GCR is not initialized even when card insertion is detected
1: GCR is initialized when card insertion is detected

Bits 6-5: P1IREQE1-0 (IREQ interrupt enable)

In cases where the card in slot 1 is an I/O card, these bits enable and disable IREQ interrupts and set the interrupt mode. Before changing these bits, PCC1CSCR bit 5 (P1IREQ) should first be cleared.

In the case of a memory card, operations on these bits are invalid.

Bit 6	Bit 5	
IREQ1	**IREQ0**	**Settings**
0	0	Interrupt disabled. P1IREQ does not change, regardless of IREQ signal level.
0	1	Level mode interrupt. P1IREQ is set when the IREQ signal is at low level, and an interrupt request is generated.
1	0	Edge mode interrupt. P1IREQ is set at the IREQ signal falling edge, and an interrupt request is generated.
1	1	Edge mode interrupt. P1IREQ is set at the IREQ signal rising edge, and an interrupt request is generated.

Bit 4: P1SCE (STSCHG interrupt enable)

In cases where the card in slot 1 is an I/O card, this bit enables and disables PCC1BVD1/STSCHG interrupts. An interrupt request is generated when PCC1CSCR bit 4 is set and this bit is 1. When the card is a memory card, operations on this bit are invalid.

0: STSCHG interrupt disabled
1: STSCHG interrupt enabled

Bit 3: P1CDE (card detection interrupt enable)

Enables and disables PCC1CD2 and PCC1CD1 interrupts. An interrupt request is generated when PCC1CSCR bit 3 is set and this bit is 1.

0: PCC1CD2/PCC1CD1 interrupt disabled
1: PCC1CD2/PCC1CD1 interrupt enabled

Bit 2: P1RE (ready change interrupt enable)

In cases where the card in slot 1 is a memory card, this bit enables and disables PCC1RDY interrupts. An interrupt request is generated when PCC1CSCR bit 2 is set and this bit is 1. In the case of an I/O card, operations on this bit are invalid.

0: RDY/BSY interrupt disabled
1: RDY/BSY interrupt enabled

Bit 1: P1BWE (battery voltage low interrupt enable)

In cases where the card in slot 1 is a memory card, this bit enables and disables battery voltage-low interrupts. An interrupt request is generated when PCC1CSCR bit 1 is set and this bit is 1. In the case of an I/O card, operations on this bit are invalid.

0: Battery voltage-low interrupt disabled
1: Battery voltage-low interrupt enabled

Bit 0: P1BDE (battery replacement interrupt enable)

In cases where the card in slot 1 is a memory card, this bit enables and disables battery replacement interrupts. An interrupt request is generated when PCC1CSCR bit 0 is set and this bit is 1. In the case of an I/O card, operations on this bit are invalid.

0: Battery replacement interrupt disabled
1: Battery replacement interrupt enabled

[9] PCC voltage control register (PCCVCR)

Bit	7	6	5	4	3	2	1	0
Name	VCC0 SEL1	VCC0 SEL0	VPP0 SEL1	VPP0 SEL0	VCC1 SEL1	VCC1 SEL0	VPP1 SEL1	VPP1 SEL0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	1	1	0	0	1	1

This register is used to control the power supply for slots 0 and 1. Four control signals for each slot are directly driven either high or low.

Bits 7-0:

0: Corresponding signal driven low
1: Corresponding signal driven high

PCC Power Supply Correspondence

	VCCXSEL1	VCCXSEL0	VCC Output	VPPXSEL1	VPPXSEL0	VPP Output
Settings	0	0	OFF	0	0	0V
	0	1	5V	0	1	VPP
	1	0	3.3V	1	0	VCC
	1	1	OFF	1	1	Hi-Z

5.4.4 E10A Emulator Interface

The CPU board is equipped with a Hitachi-UDI port connector (CN8) to which an SH7729R E10A emulator can be connected. SH7729R H-UDI and AUD signals are connected directly to this connector.

When connecting an E10A emulator, always be sure to turn on the jumper J11 on the CPU board.

Figure 5.16 shows the pin arrangement of the Hitachi-UDI port connector (CN8). Table 5.11 shows the pin assignment for the Hitachi-UDI port connector (CN8).

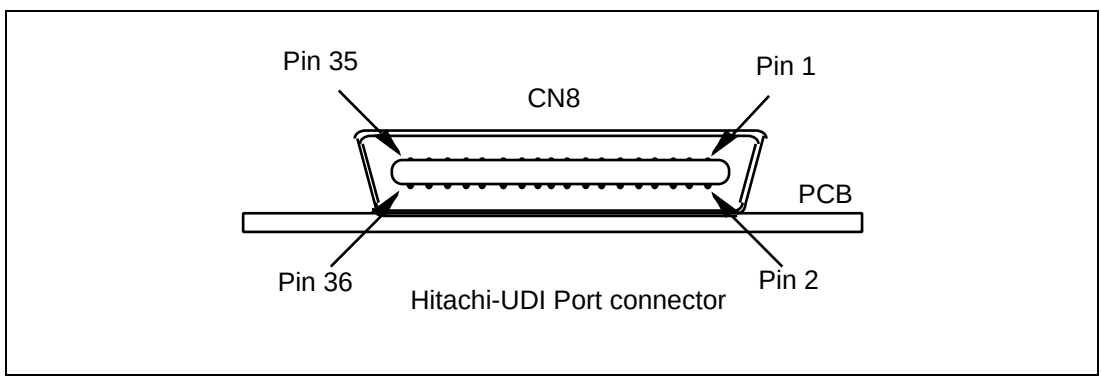


Figure 5.16 Hitachi-UDI Port Connector (CN8) Pin Arrangement

Table 5.11 Hitachi-UDI Port Connector (CN8) Pin Assignment

CN8	SH7729R		CN8	SH7729R	
Pin No.	**Signal Name**	**Input/Output**	**Pin No.**	**Signal Name**	**Input/Output**
1	NC	—	19	TMS	Input
2	GND	—	20	GND	—
3	AUDDATA[0]	Input/output	21	_TRST	Input
4	GND	—	22	GND	—
5	AUDDATA[1]	Input/output	23	TDI	Input
6	GND	—	24	GND	—
7	AUDDATA[2]	Input/output	25	TDO	Input
8	GND	—	26	GND	—
9	AUDDATA[3]	Input/output	27	_ASEBRKAK	Output
10	GND	—	28	GND	—
11	_AUDSYNC	Input/output	29	Reserve	Output
12	GND	—	30	GND	—
13	NC	—	31	_RESET	Output
14	GND	—	32	GND	—
15	NC	—	33	GND	—
16	GND	—	34	GND	—
17	TCK	Input	35	AUDCK	Input
18	GND	—	36	GND	—

5.5 Parts Layout

The parts layout of the CPU board is shown in figure 5.17. Uninstalled parts are listed in table 5.12.

Table 5.12 List of Uninstalled Parts

Part Number	Part Name	Quantity
OSC3	TCO-711S4	1
OSC4	CXO-105D	1
CN4*	FFC-10	1
J4 to J6, J12, J13	310-93-103	5
J8	410-93-202	1
TP1 to TP12	ST-1-3	12
CKIN	ST-1-3	1
TEST	ST-1-3	1
R4	MCR10EZHJ472	1
R7 to R9, R11	MCR10EZHJ103	4
R12, R149, R151	MCR10EZHJ000	3
C23	281E6801-337M	1
C24	281E1002-157M	1
C28	281E1002-107M	1

Note: A connector for testing. Should not be connected to any cables even if the connector is installed.

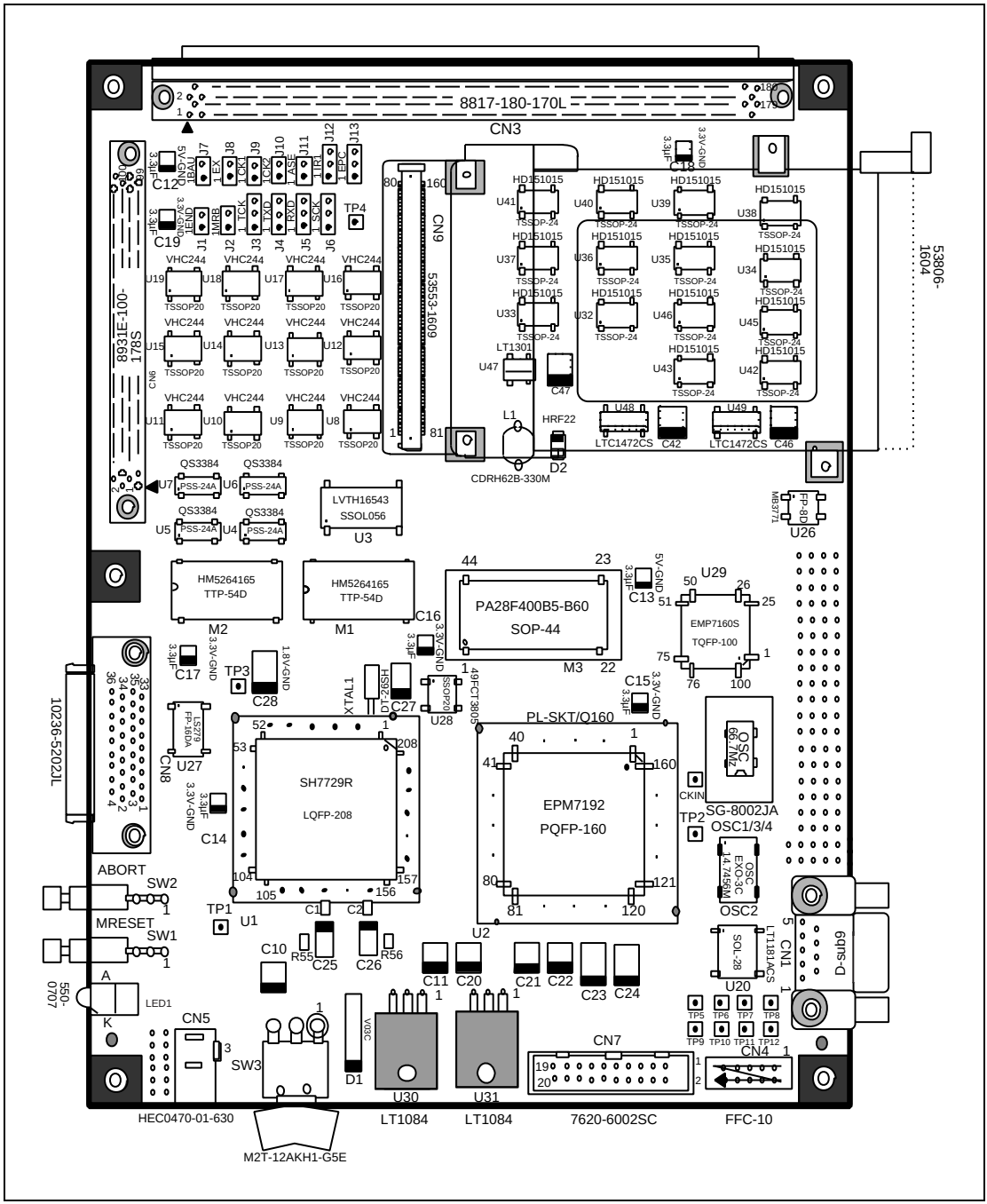


Figure 5.17 Parts Layout (Mounting Side)

5.6 Initialization

5.6.1 Initializing Resources

Table 5.13 shows which CPU board resources are initialized.

Table 5.13 Resource Initialization

		Interrupt Source Causing Initialization				
		Power-On Reset		Manual Reset Switch		
Resource		Hardware	Monitor Program	Hardware	Monitor Program	Remarks
SH7729R	CPU	O	—	O	—	
	MMU	O	—	O	—	
	CACHE	O	—	—	—	
	TLB	O	—	—	—	
	CCN	O	—	—	—	
	INTC	O	O	O	O	
	UBC	O	—	—	—	
	CPG	O	O	—	O	
	WDT	O	O	—	O	
	BSC	O	O	—	O	
	DMAC	O	—	O	—	
	TMU	O	—	O	—	
	RTC	O	—	O	—	
	SCI	O	O	O	O	
	IRDA	O	—	O	—	
	SCIF	O	—	O	—	
	I/O PORT	O	—	—	—	
	ADC	O	—	O	—	
	DAC	O	—	O	—	
	CMT	O	—	O	—	

Notes: 1. O: Initialized
—: Not initialized

2. When, during HDI and CPU board operation, a power-on reset occurs due to a power supply voltage drop or for other reasons, a “Power on reset is detected.” message box is displayed. However, in this case the CPU general-use registers and control registers are not initialized. Change settings as necessary or restart the HDI.

Table 5.13 Resource Initialization (cont)

Resource		Interrupt Source Causing Initialization: Power-On Reset: Hardware	Interrupt Source Causing Initialization: Power-On Reset: Monitor Program	Interrupt Source Causing Initialization: Manual Reset Switch: Hardware	Interrupt Source Causing Initialization: Manual Reset Switch: Monitor Program	Remarks
SH7729R	DSP	O	—	—	—	
	AUD	—	—	—	—	Initialized by TRST
	H-UDI	—	—	—	—	Initialized by TRST
	ASERAM	—	—	—	—	Initialized by TRST
	XYCNT	O	—	—	—	
	XYMEM	—	—	—	—	
Interrupt controller		O	O	O	O	
PCMCIA		O	O	—	—	
SDRAM (Monitor program work area)		—	O	—	O	
SDRAM (User program area)		—	—	—	—	

Notes: 1. O: Initialized

—: Not initialized

2. When, during HDI and CPU board operation, a power-on reset occurs due to a power supply voltage drop or for other reasons, a “Power on reset is detected.” message box is displayed. However, in this case the CPU general-use registers and control registers are not initialized. Change settings as necessary or restart the HDI.

5.6.2 Procedure for Making Initial Settings of the CPU Bus State Controller (BSC)

Figure 5.18 is a flowchart of the procedure for setting the initial settings of the bus state controller (BSC). For information on the settings of each BSC register, please refer to section 5.6.3, Initial Settings of CPU Bus State Controller (BSC).

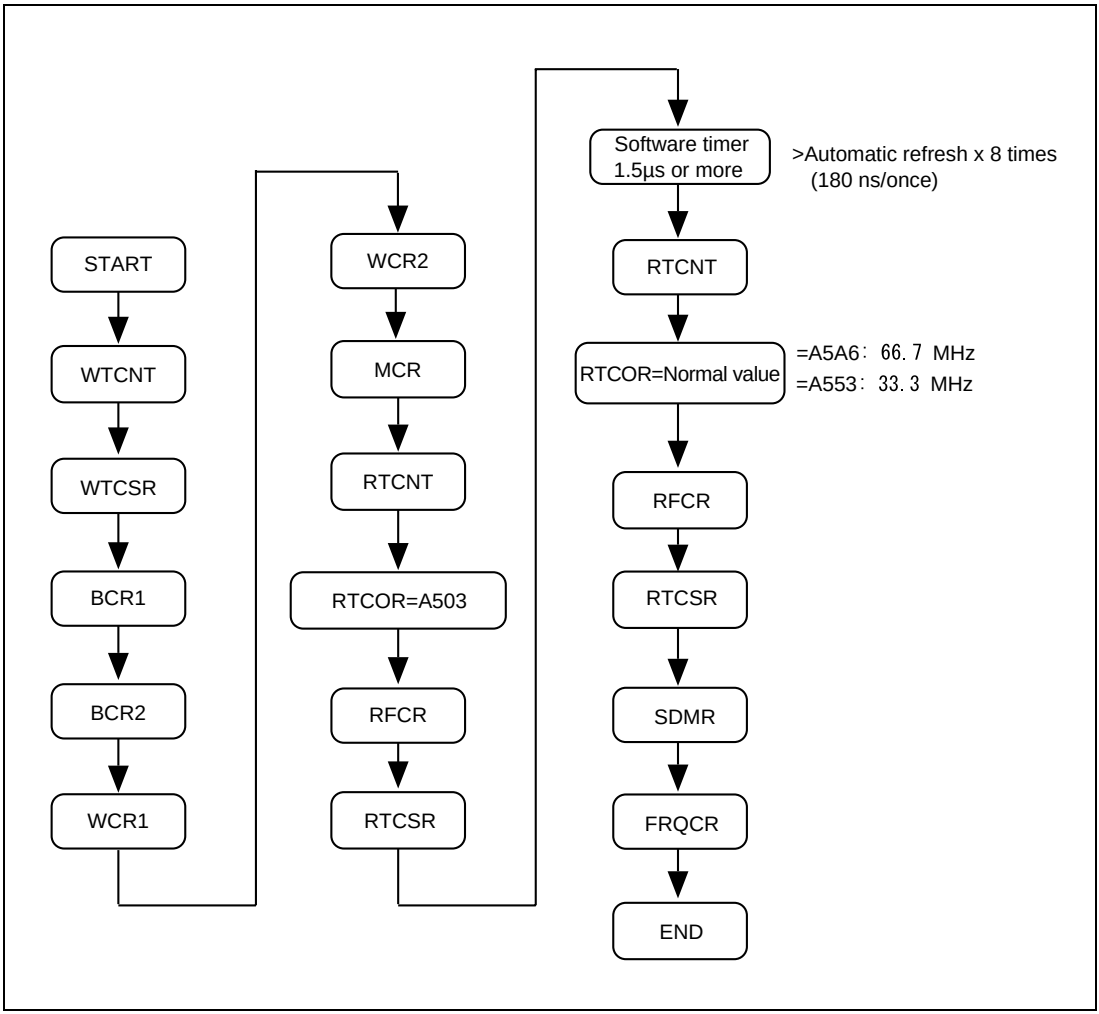


Figure 5.18 Procedure for Setting BSC Settings

5.6.3 Initial Settings of CPU Bus State Controller (BSC)

The clock mode is set to 7 in the CPU board. In the bus state controller (BSC) registers, bits corresponding to areas 0 and 3 must not be modified because these areas are assigned to resources of the CPU board. If these bits are modified, the CPU board will not operate. The following shows the initial BSC register values set by the monitor program. In the figures, the shaded bits must not be modified. Separate figures are used to show the register values that depend on the operating frequency (CKIO).

Registers Whose Values are the Same at 66.7 MHz and 33.3 MHz:

- BCR1 (H'FFFFFF60) = H'0009

Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name	—	—	HIZ MEM	HIZ CNT	END IAN	A0 BST1	A0 BST0	A5 BST1	A5 BST0	A6 BST1	A6 BST0	DRA M TP2	DRA M TP1	DRA M TP0	A5 PCM	A6 PCM
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1

- BCR2 (H'FFFFFF62) = H'2FF0

Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name	—	—	A6 SZ1	A6 SZ0	A5 SZ1	A5 SZ0	A4 SZ1	A4 SZ0	A3 SZ1	A3 SZ0	A2 SZ1	A2 SZ0	—	—	—	—
Initial value	0	0	1	0	1	1	1	1	1	1	1	1	0	0	0	0

- WCR1 (H'FFFFFF64) = H'9F71

Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name	WAIT SEL	—	A6 IW1	A6 IW0	A5 IW1	A5 IW0	A4 IW1	A4 IW0	A3 IW1	A3 IW0	A2 IW1	A2 IW0	—	—	A0 IW1	A0 IW0
Initial value	1	0	0	1	1	1	1	1	0	1	1	1	0	0	0	1

- WTCNT (H'FFFFFF84) = H'5A00

Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name	—	—	—	—	—	—	—	—	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Initial value	0	1	0	1	1	0	1	0	0	0	0	0	0	0	0	0

- WTCSR (H'FFFFFF86) = H'A547

Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name	—	—	—	—	—	—	—	—	TME	WT/IT	RSTS	WOVF	IOVF	CSK2	CSK1	CSK0
Initial value	1	0	1	0	0	1	0	1	0	1	0	0	0	1	1	1

- MCR (H'FFFFFF68) = H'5724

Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name	TPC1	TPC0	RCD1	RCD0	TRWL1	TRWL0	TRAS1	TRAS0	—	BE	—	AMX1	AMX0	RFSH	RMODE	EDOMODE
Initial value	0	1	0	1	0	1	1	1	0	0	1	0	0	1	0	0

- RTCNT (H'FFFFFF70) = H'A500

Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	1	0	1	0	0	1	0	1	0	0	0	0	0	0	0	0

Register Values at 66.7 MHz:

- FRQCR (H'FFFFFF80) = H'A101

Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name	STC2	IFC2	PFC2	—	—	—	SLP FRQ	CKO EN	PLL EN	PST BY	STC1	STC0	IFC1	IFC0	PFC1	PFC0
Initial value	1	0	1	0	0	0	0	1	0	0	0	0	0	0	0	1

- SDMR (H'FFFFE088) = H'00

Bit No.	7	6	5	4	3	2	1	0
Bit name	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0

- RTCSR (H'FFFFFF6E) = H'A508

Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name	—	—	—	—	—	—	—	—	CMF	CMIE	CKS2	CKS1	CKS0	OVF	OVIE	LMT S
Initial value	1	0	1	0	0	1	0	1	0	0	0	0	1	0	0	0

- RTCOR (H'FFFFFF72) = H'A5A6

Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name	—	—	—	—	—	—	—	—	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Initial value	1	0	1	0	0	1	0	1	1	0	1	0	0	1	1	0

- RFCR (H'FFFFFF74) = H'A400

Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	1	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0

- WCR2 (H'FFFFFF66) = H'FFDB

Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name	A6W2	A6W1	A6W0	A5W2	A5W1	A5W0	A4W2	A4W1	A4W0	A3W1	A3W0	A2W1	A2W0	A0W2	A0W1	A0W0
Initial value	1	1	1	1	1	1	1	1	1	1	0	1	1	0	1	1

Register Values at 33.3 MHz:

- FRQCR (H'FFFFFF80) = H'A111

Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name	STC2	IFC2	PFC2	—	—	—	SLPFRQ	CKOEN	PLLEN	PSTBY	STC1	STC0	IFC1	IFC0	PFC1	PFC0
Initial value	1	0	1	0	0	0	0	1	0	0	0	1	0	0	0	1

- SDMR (H'FFFFE088) = H'00

Bit No.	7	6	5	4	3	2	1	0
Bit name	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0

- RTCSR (H'FFFFFF6E) = H'A508

Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name	—	—	—	—	—	—	—	—	CMF	CMIE	CKS2	CKS1	CKS0	OVF	OVIE	LMTS
Initial value	1	0	1	0	0	1	0	1	0	0	0	0	1	0	0	0

- RTCOR (H'FFFFFF72) = H'A553

Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name	—	—	—	—	—	—	—	—	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Initial value	1	0	1	0	0	1	0	1	0	1	0	1	0	0	1	1

- RFCR (H'FFFFFF74) = H'A400

Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	1	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0

- WCR2 (H'FFFFFF66) = H'FFDA

Bit No.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name	A6W2	A6W1	A6W0	A5W2	A5W1	A5W0	A4W2	A4W1	A4W0	A3W1	A3W0	A2W1	A2W0	A0W2	A0W1	A0W0
Initial value	1	1	1	1	1	1	1	1	1	1	0	1	1	0	1	0

Section 6 Notes and Troubleshooting

6.1 Notes

1. When executing a user program, the following interrupts cannot be used. This is because the monitor program is using the following CPU functions for debugging purposes.
   — User Break Controller (UBC)
   — Serial Communication Interface (SCI)
   — TRAPA#255 Trap instruction

2. The monitor program uses NMI and SCI interrupts, so the user can use interrupts of mask level 14 or lower. If an interrupt is set to mask level 15, correct operation cannot be guaranteed. The default mask level is set to 14.

3. Interrupt and Exception Display
   (1) The following interrupts and exceptions are displayed on the status bar during user program execution.
   - Address error
   - Illegal general instruction
   - Illegal slot instruction
   - NMI

   (2) When an exception occurs on the CPU board while the user program is not being executed, the HDI will display the EXPEVT code that corresponds to the cause of the exception, and the monitor program will enter the reset-input wait state. In this case, turn the power to the CPU board off and on, or input a reset, and start the HDI to initiate link-up processing.

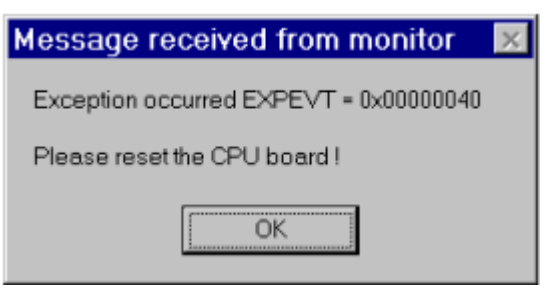


Figure 6.1 Error Message Displayed at Exception (Instruction TLB Miss Exception)

4. Breakpoints cannot be set in a delay slot of a user program. If an attempt is made to set such a breakpoint, the following message appears.

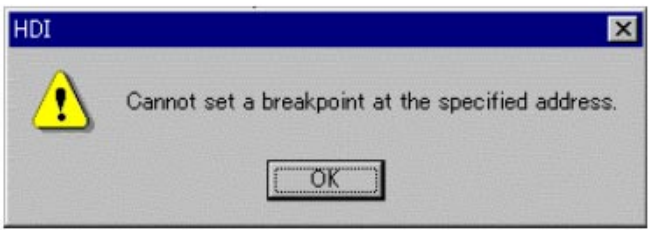


Figure 6.2 Message Box Indicating Breakpoint Cannot Be Set

For the above reason, a breakpoint may not be set if the initial data in the RAM is invalid when a user program is loaded by specifying a session file. In this case, reload the program by respecifying the session file.

5. If a breakpoint is set within an interrupt or exception handler of a user program, it will not cause a break when the program is executed.

6. When a power-on reset or manual reset is input, the message box shown in figure. 6.3 and figure. 6.4 respectively is shown.

   At this time, general-purpose registers and control registers are not reset. Either settings should be changed as necessary, or the HDI should be restarted. The run time count function must be enabled again by the Run Time Count Condition dialog box.

Figure 6.3 Power-on Reset Input Message Box

Figure 6.4 Manual Reset Input Message Box

7. If the power is turned on while pressing and holding the manual reset switch, the CPU board and HDI will not be started. When turning on power, do not operate the manual reset switch.

   In this case, a power-on reset is input to the CPU when the power is turned on, and the BSC register is initialized. However, manual reset code is set for event code flags within the CPU, and so the monitor program performs manual reset processing (registers are not initialized). Consequently initial values for monitoring are not written to the BSC in the CPU, and access to various resources is not possible.

8. In some cases I/O register values may not be correctly displayed in the memory window. This is because the HDI reads all areas in byte units. In order to display the correct I/O register values, select [I/O Registers] from the [View] menu.

9. The monitor program sets and uses some of the BSC registers. When rewriting these registers, refer to section 5.6.3, Initial Settings of CPU Bus State Controller (BSC).

10. I/O Ports

    Some of the port terminals on this board are also used as pins for other functions, and so some pins cannot be used for port functions. The pin functions that can be used are included as signal names in the pin names of the expansion connector pin assignments in tables 5.4, 5.5 and 5.6 in section 5.4, Interface. If signal name functions not included in the pin name are used, correct operation is not guaranteed.

11. CPU Operating Mode

    The operating mode of this CPU is set at clock mode 7 and area 0 bus width of 16 bits. The endian configuration can be changed using jumper J1. (See section 2.8, Jumpers, for details on settings.)

12. Expansion Bus

    (1) Areas available for use with the expansion bus

    The expansion bus can use areas 2, 4, and 5, and cannot use area 0, 1, 3, or 6. Monitor FLASH memory and monitor I/O are connected to area 0; SDRAM is connected to area 3; and two PCMCIA slots are connected to area 6.

    (2) Devices which cannot be connected to the expansion bus

    DRAM cannot be connected to the expansion bus. (This is because SDRAM on the CPU board is allocated to area 3.)

    (3) Interrupts

    The NMI, external interrupts, and port interrupts should all be processed by interrupt handlers in the user program. For details refer to section 7, Creation of User Interrupt Handlers.

    IRQ5 is assigned as a pin CTS2/SCPT[7] with other functions. For this reason, the user cannot use the IRQ5 external interrupt. Use the other external interrupts instead.

13. Refresh Timer

    The refresh timer is used as an SDRAM refresh timer, and so cannot be used as an interval timer.

14. TMU

The run time count function uses the TMU0, and so the user cannot use the TMU0. If the TMU0 register is accidentally overwritten, correct operation is not guaranteed.

15. SCI

The serial interface with the host uses the SCI0 (SCI) in the CPU. For this reason, the user cannot use the CPU's internal SCI0 except for the standard I/O processing using the [Simulated I/O Window]. For details, refer to section 7.2, User Program Using SCI. Use SCI1 (IrDA) or SCI2 (SCIF) for a purpose other than standard I/O processing. If the SCI0 interface register is accidentally overwritten, the CPU board and HDI will become inoperable.

16. PCMCIA Interface

This CPU board is not provided with a PCMCIA driver. When using a PCMCIA card, please provide your own PCMCIA driver. When using a PCMCIA card, always be sure that after detecting the card detection signal CD0/CD1, the PCMCIA driver enables PCMCIA external buffer control (the P0DRVE bit or the P1DRVE bit). If the buffer is enabled prior to card detection, correct operation is not guaranteed.

17. E10A Emulator Interface

If the monitor is used with the E10A emulator connected, correct operation is not guaranteed. Moreover, with the E10A emulator connected the port functions of the Hitachi-UDI and AUD interface signal pins cannot be used.

18. Host Interface Software

When using this CPU board, always use the included Hitachi Debugging Interface (HDI). If other host interface software is used, the operation of the CPU board and of user programs is not guaranteed.

19. User Program Execution

(1) Abort switch (SW2) is connected to the NMI on the CPU. Therefore, do not change settings about the NMI function register without preparing NMI handling routine. If the register is accidentally overwritten, correct operation cannot be guaranteed including abort switch operation.

When modifying the register value, be sure to prepare a NMI handling routine and do not use the abort switch.

(2) If a single-step execution is performed for an illegal instruction, the program counter will not increment the count; do not perform single-step executions for illegal instructions.

(3) When a multiple step execution is performed for a program that contains the SLEEP instruction (from the [Step…] menu), the execution speed ([RATE]) must be set to 6. Otherwise an error ([Command not ready] error) will occur when the SLEEP instruction is executed, and execution from then on cannot be accepted. In such case, press the abort switch and start execution again.

(4) During step execution, standard C libraries are also executed. To return to a higher-level function, use Step Out. In a for statement or a while statement, executing a single step does not move execution to the next line. To move to the next line, execute two steps.

(5) If a user program is halted by a break due to the step execution or a breakpoint, the Halt button, or the abort switch (SW2), it is handled same as the exception in the user program. If these causes are generated, 16 bytes of the user stack area will be used. Reserve the sufficient stack areas, and do not allocate the user program or data.

(6) If the Halt button is pressed in the sleep mode, the operation cannot be guaranteed. When the mode should be returned from sleep, press the abort switch on the board or enter other interrupts.

(7) If a breakpoint is set for the sleep instruction, the execution time during sleep mode cannot be measured. When the user wants to measure the execution time including the sleep instruction, do not set a breakpoint for the sleep instruction.

20. Breakpoints

(1) During single-step execution, the settings of the breakpoints are ignored.

(2) A total of 255 breakpoints (including temporary breakpoints) can be set. A temporary breakpoint cannot be set to the same address as an enabled breakpoint.

If a temporary breakpoint is set to the same address as a disabled breakpoint, the breakpoint will be deleted from the breakpoints list after program execution have been completed.

(3) After setting breakpoints, the CPU board must not be manually reset while the user program is being executed. Otherwise, illegal instructions will remain at addresses where breakpoints have been specified.

To continue debugging the user program, download the user program again.

(4) In the CPU board, the TRAPA instruction is used for breakpoint functions. When using DSP repeat loops on the CPU board, note that in some DSP repeat loops, branch instructions cannot be used, and so breakpoints cannot be set in such loops. For details, refer to the SH7729R Hardware Manual and Programming Manual.

(5) After 255 breakpoints (the maximum number) have been specified, if Add/Edit Breakpoint is selected in the Breakpoints window, an error will occur. In this case, delete any unnecessary breakpoints, and then add or edit breakpoints.

(6) When the contents of a software breakpoint address are modified during user program execution, the breakpoint is disabled. The user program will not stop at this address.

When the program stops with any other reason, the modified contents will be discarded and returned to original value (set before program execution).

21. Watch

(1) Local variables at optimization

Depending on the generated object code, local variables in a C source file that is compiled with the optimization option enabled will not be displayed correctly. Check the generated object code in the mixed display of the Program window.
If the allocation area of the specified local variable does not exist, displays as follows.

Example: The variable name is asc.

asc = ? - target error 2010 (xxxx)

(2) Variable name specification

When a name other than a variable name, such as a symbol name or function name, is specified, no data is displayed.

Example: The function name is main.

main =

(3) Array display

When the number of array elements exceeds 1000, the number exceeding 1001 will not be displayed.

22. Serial Interface

(1) When the bus frequency is set to 33.3 MHz by jumpers J9 and J10, the serial interface baud rate must not be set to 115,200 bit/s; set the baud rate to 57,600 bit/s.

(2) The CPU board does not control data flow. Therefore, when transferring a large amount of data from the CPU board to the host computer, an overrun error may occur.

If an error occurs after performing one of the following corrective actions, repeat the action. In the control panel of the host computer, set the serial port to 115,200 bit/s and FIFO to the initial state.

Table 6.1 Operations that may Generate Errors

Operation	Executing Environment	Conditions of Error Generation*	Error Message	To Recover from Error
Multiple step execution	Menu	If more than few hundred steps are specified.	Unable to fetch register (displayed in the message box)	When an error occurs, press the manual reset button and initialize the HDI. Then execute multiple steps.
Save memory	Menu	If more than few hundred kbytes are specified.	Command not ready (displayed in the message box)	Click the OK button in the message box and execute Save memory again.

Note: The Conditions of Error Generation are for reference.

If an error still occurs after the action is executed again, it may mean that a large amount of load is applied to the host computer, and is making the access speed to the serial port slower than the transfer speed. In such a case, close other applications opened on the host computer to reduce the load.

(3) If serial interface is disconnected while a program is being down loaded through the serial interface, the HDI will stop abnormally. In this case, connect the serial interface cable correctly, and restart the CPU board and HDI.

(4) The HDI does not support Motorola S-type files with only the CR code (H'0D) at the end of each record. Load Motorola S-type files with the CR and LF codes (H'0D0A) at the end of each record.

(5) The CPU board and the HDI do not limit the address range for downloading. Be sure to download program to RAM areas. Otherwise the operation cannot be guaranteed.

(6) When a Motorola S-type file is downloaded, two menus, Load Program and Load Memory, can be used, but the Load Program is recommended because it can transfer data faster.

23. HDI

(1) In this HDI, the [Command Line] menu can be selected, but operation of the command entered from the command line cannot be guaranteed. Do not use the command line.

(2) This HDI does not support software breakpoint setting in the [Select Function] dialog box (described in section 10, Selecting Functions, in the Hitachi Debugging Interface User's Manual).

(3) If the following memory contents are displayed in the [Memory] window, they will be incorrect.

- Word access from address 2n + 1
- Longword access from address 4n + 1, 4n + 2, or 4n + 3

The font size used in the memory window must be 4 or larger. In one window, up to 32768bytes can be displayed.

(4) For each Watchdog Timer register, there are two registers to be separately used for write and read.

Table 6.2 Watchdog Timer Register

Register Name	Usage	Register
WTCSR(W)	Write	Watchdog timer control/status register
WTCNT(W)	Write	Watchdog timer counter
WTCSR(R)	Read	Watchdog timer control/status register
WTCNT(R)	Read	Watchdog timer counter

(5) Since different versions of HDI cannot be used at the same time, re-install this HDI whenever another previously installed HDI is used.
If another HDI has been used, initiate this HDI with “Run” as follows, without using the session files.

```
<Directory path name in which HDI is installed>\hdi /n (RET)
```

/n initiates the HDI without loading the recently used session files.

If there is another session file of a different debug platform, the following error message is displayed:

```
invalid target system: <recently used debug platform name>
```

(6) When another HDI is uninstalled after installation of this HDI, some functions may not work correctly. In this case, re-install this HDI.

(7) For operations such as [Fill Memory] and [Test Memory], it may take a few minutes to complete operation. This depends on the size of data specified. Since only 5 seconds is specified for the timeout time in the HDI of the CPU board, the "Command not ready" error may occur. In such cases, the results of the operation will not be guaranteed; therefore, specify a smaller size and execute the operation again.

(8) The access size and target start and end addresses can be specified in the [Fill Memory] dialog box. If the access size does not match the specified start address, the HDI will treat them as follows:

- Fill size = end address - start address
- The fill size is decreased to the nearest integer multiple of the access size.
- The start address is decreased to the nearest integer multiple of the access size.
- If the fill size is smaller than the access size, the fill size is increased to the nearest integer multiple of the access size.

After the above processing, the HDI performs the memory-fill operation.

(9) The values displayed as Cache Status and MMU Status in the Status window are the values for the last break that occurred during user program execution. Values as updated in the I/O Registers window are not displayed in the Status window.

(10) The overhead due to run time measurement is about 20 us in 66.7MHz, about 35 us in 33.3MHz per execution.

24. The default I/O register definition file does not include bit-level information. To use bit-level information, create a new I/O definition file or modify the default I/O register definition file. For details on the format of I/O register definition files, refer to appendix E, I/O Register File Format, in the Hitachi Debugging Interface User’s Manual.

25. When the menu is selected from the display during user program execution, some commands are activated; however, in the CPU board, only Labels,Status,Simulated I/O Window,Halt and Help commands can be used.

6.2 Troubleshooting

1. “Illegal general instruction” appears on the HDI status bar, and program execution is halted.

   This is displayed when a general exception occurs. This message appears when the EXPEVT register value is H’180. It is caused by use of a privileged instruction in user mode, by use of an undefined instruction, or for similar reasons. For further information refer to the SH7729R Hardware Manual.

   When a privileged instruction has been used in user mode, please take the following steps.

   — In the register window, change the SR (status register) MD bit to 1 (privileged mode).

   — Or, execute the [Reset CPU] on the [Run] menu, and set the register values to the following initial values.

Table 6.3 Register Initial Value Settings

Register	Initial Value	Description
PC	H’AC000000	User program area start address
SR	H’600010E0	Privileged mode, mask level 14
R15 (SP)	H’ACF80000	Final address of user program area
VBR	H’A0008000 (big endian) H’A0048000 (little endian)	Monitor VBR (different from actual chip)

2. Step execution is slow.

   When the [Watch Window] window and [I/O Registers] window are open, the data in these windows must be rewritten each time a step is executed, and so execution speed will be reduced. Decrease the sizes of these windows to speed execution.

Section 7 Creation of User Interrupt Handlers

7.1 Creation of User Interrupt Handlers

Cases where exceptions and interrupts are not used in the user program (no user interrupt handlers are created):

Set the value of VBR to the initial value (big-endian: H'A0008000, little-endian: H'A0048000). By doing so, step execution, breaks, and other debugging functions can be used when an exception or interrupt occurs.

Cases where exceptions and interrupts are used in the user program (user interrupt handlers are created):

Set the value of VBR to the start address of the user interrupt handler. By doing so, execution will branch to the user interrupt handler when an exception or interrupt occurs. Add a routine to branch to the following addresses to the user interrupt handler; this will enable step execution, breaks and other debugging functions when an exception or interrupt occurs.

Table 7.1 lists the branch addresses for different interrupt causes.

Table 7.1 Interrupt Causes and Branch Addresses for User Interrupt Handlers

Interrupt Cause	Code	Branch Address (Big-Endian)	Branch Address (Little-Endian)
UBC trap	EXPEVT=1E0	H'A0009000	H'A0049000
Unconditional trap (FF)	EXPEVT=160	H'A0009020	H'A0049020
Reserved instruction code exception	EXPEVT=180	H'A0009040	H'A0049040
Slot illegal instruction exception	EXPEVT=1A0	H'A0009060	H'A0049060
CPU address error (load)	EXPEVT=0E0	H'A0009080	H'A0049080
CPU address error (store)	EXPEVT=100	H'A0009080	H'A0049080
DMA address error	EXPEVT=5C0	H'A00090A0	H'A00490A0
NMI	INTEVT=1C0	H'A00090C0	H'A00490C0
SCI-RXI*	INTEVT=500	H'A00090E0	H'A00490E0

Note: When the SCI is not used, a program to branch to the SCI-RXI destination address must be prepared.

Attention should be paid to the following when creating an interrupt handler.

1. When branching to a branch address from the user interrupt handler, the values of R0 and R1 (BANK1) must be saved on the stack.

   In other words, @(R15-8) = R1 at time of exception (BANK1)
   @(R15-4) = R0 at time of exception (BANK1)

   The following is an example of code which achieves this.

   ```
   MOV.L    R0, @-R15; save R0_BANK1
   MOV.L    R1, @-R15; save R1_BANK1
   ```

   By this means, register values can be displayed during debugging when an exception or interrupt occurs.

2. The values of general-purpose registers other than R0 and R1 should be saved at the time of occurrence of an exception or interrupt.

3. The BL bit of the SR register should be kept as 1 from the occurrence of an exception until branching to the branch address.

4. The values of the SSR, SPC, EXPEVT, INTEVT, INTEVT2, and TRA registers should be saved at the time of occurrence of an exception.

5. Branching should be performed with the RB and MD bits of the SR register both set to 1 (the state of occurrence of an exception or interrupt).

7.2 User Program Using SCI

The user program cannot usually access the serial communication interface with FIFO (SCIF) in the SH7729R because the CPU board uses it to communicate with the host PC. The CPU board provides the [Simulated I/O Window] window to allow the user to use the SCIF.

When the SCIF is used from the user program, the SCIF driver in the user program communicates with the [Simulated I/O Window] window on the host PC rather than with the actual SCIF directly. As processing for interrupts of the CPU must be added to the user program, a user interrupt handler must be created according to the directions in section 7.1, Creation of User Interrupt Handlers.

The HDI installer CD-R supplied with the CPU board contains a sample program for the user interrupt handler and SCIF driver. For details on the sample program, refer to section 7.3, Sample Program.

7.2.1 Creation of SCI Driver

Note the following when creating the SCI0 driver.

- To receive serial data, an interrupt must be used. Create a SCI-RXI (receive data full interrupt request) processing routine.
- When the [HALT] button is pressed during serial receiving operation, the HDI sends the HALT code (H'12) to the CPU board. When the HALT code is received, execution must branch to the HALT break processing address in the CPU board.

Table 7.2 HALT Break Destination Address

Endian	HALT Break Destination Address
Big endian	H'A00090E0
Little endian	H'A00490E0

- The branch to the HALT break processing is performed in the same interface as the branch to user interrupt handlers. For details, refer to section 7.1, Creation of User Interrupt Handlers.

Notes: 1. The SCI0 is used for communication between the CPU board and the host computer. If the user uses the SCI0 for a purpose other than the communication with the [Simulated I/O Window] window, correct operation cannot be guaranteed. For such purposes, use the SCI1 (IrDA) or SCI2 (SCIF).

2. If processing that branches execution to the HALT break when the HALT code is received is not prepared, the program cannot be stopped by clicking the [HALT] button in the HDI.

7.2.2 SCI-Related Register Settings

The initial values of the SCI-related registers are shown below. The shaded bits must not be modified.

Serial Communication Interface (SCI0):

SCSMR (H'FFFFFE80) = H'00

Bit	7	6	5	4	3	2	1	0
	C/A	CHR	PE	O/E	STOP	MP	CKS1	CKS0
Initial value	0	0	0	0	0	0	0	0

SCBRR (H'FFFFFE82) = H'FF

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—
Initial value	1	1	1	1	1	1	1	1

SCSCR (H'FFFFFE84) = H'73

Bit	7	6	5	4	3	2	1	0
	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0
Initial value	0	1	1	1	0	0	1	1

No restrictions are placed on accesses to the SCTDR, SCSSR, SCRDR, SCPDR, and SCPCR.

Interrupt Controller (INTC):

For the interrupts used by the user program, any interrupt level from 0 to 14 can be set, but interrupt level 15 must not be used except for the SCI.

ICRA (H'FFFFFEE2) = H'0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TMU0				TMU0				TMU0				RTC			
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ICRB (H'FFFFFEE4) = H'00F0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	WDT				REF				SCI				—	—	—	—
Initial value	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0

ICRC (H'A4000016) = H'0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	IRQ3				IRQ2				IRQ1				IRQ0			
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ICRD (H'A4000018) = H'0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PINT0 to PINT7				PINT8 to PINT15				IRQ5				IRQ4			
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ICRE (H'A400001A) = H'0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DMAC				IrDA				SCIF				ADC			
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ICR0 (H'FFFFFEE0) = Undefined

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	NMIL							NMIE								
Initial value	0/1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ICR1 (H'A4000010) = H'4000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MAI	IRQLVL	BLMSK	IRLSEN	IRQ51S	IRQ50S	IRQ41S	IRQ40S	IRQ31S	IRQ30S	IRQ21S	IRQ20S	IRQ11S	IRQ10S	IRQ01S	IRQ00S
Initial value	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0

No restrictions are placed on accesses to the ICF2, PINTER, IRR0, IRR1, and IRR2.

7.3 Sample Program

This section describes how to create user interrupt handlers and the SCI0 driver by using sample programs.

The sample program files were created in C language and in SH-series assembly language by the work space of the Hitachi Embedded Workshop (HEW). This program performs echo-back of the characters input from the [Simulated I/O Window] window, line by line, by using the SCI0. The sample program files, including source and object files, are automatically copied to the Sample directory under the HDI installation directory.

Load the compiled load module (Simio.abs), set the program counter and stack pointer values (PC = H'AC000000, R15 = H'ACF80000), and click the [Go] button to execute the program.

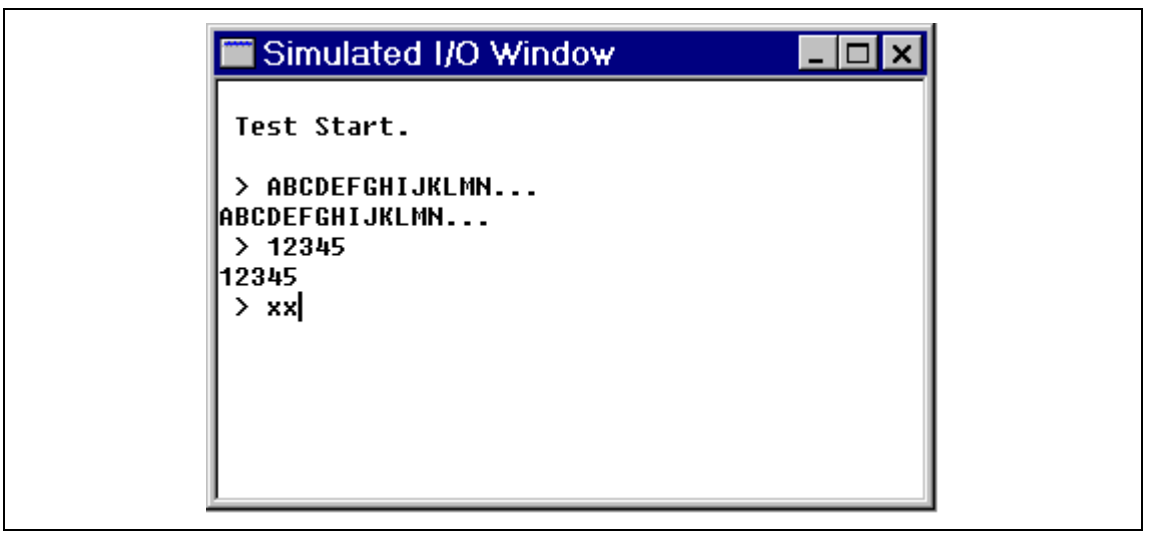


Figure 7.1 Executing the Sample Program

Note: The HEW work space supplied by this sample program was created by HEW Version 1.1 (Release 4). The program cannot be opened by a HEW version earlier than 1.1. For details, refer to the Hitachi Embedded Workshop User's Manual.

File Configuration:

Table 7.3 shows the files that compose the sample program.

Table 7.3 Sample Program Files

File Name	Description
(install directory)\Sample\Simio.hws	HEW work space file
(install directory)\Sample\Simio.hww	HEW HWW file
(install directory)\Sample\Simio\Brkaddr.inc	Branch address definition
(install directory)\Sample\Simio\Env.inc	EXPEVT/INTEVT register definition
(install directory)\Sample\Simio\Intprg.src	Interrupt processing program
(install directory)\Sample\Simio\Iodefine.h	SH7729R register definition
(install directory)\Sample\Simio\Iolevel.c	SCI driver program
(install directory)\Sample\Simio\Main.c	Main program
(install directory)\Sample\Simio\Resetprg.src	Start program
(install directory)\Sample\Simio\Serial.h	SCI relation definition
(install directory)\Sample\Simio\Stacksct.src	Global variable/stack area
(install directory)\Sample\Simio\Vect.inc	Vector definition
(install directory)\Sample\Simio\Vecttbl.src	Vector table area
(install directory)\Sample\Simio\Vhandler.src	Interrupt handler
(install directory)\Sample\Simio\Simio.hwp	HEW HWP file
(install directory)\Sample\Simio\Syntax.txt	Syntax text
(install directory)\Sample\Simio\Big\Simio.abs	ABS file for big endian
(install directory)\Sample\Simio\Little\Simio.abs	ABS file for little endian
(install directory)\Sample\Simio\Makefile\Big.mak	Make file for big endian
(install directory)\Sample\Simio\Makefile\Little.mak	Make file for little endian

Sections:

Table 7.4 shows the sections for the sample program.

Table 7.4 Sections for the Sample Program

Address	Section Name
H'0C000000 -	Start, IntPRG, P, and C
H'0CF7F000 -	Dataarea
H'0CF7FC00 - H'0CF7FFFF	Stack
H'AC010000 -	INTHandler and INTTBL

Interrupt Handlers:

The interrupt sources and their processing are shown in table 7.5. If an interrupt that is not listed in table 7.5 occurs, a sleep instruction will be executed.

Table 7.5 Interrupt Processing in the Sample Program

Interrupt Source	Code	Processing
UBC trap	EXPEVT = 1E0	Branches to the CPU board. This processing is used by step execution.
Unconditional trap (FF)	EXPEVT = 160	Branches to the CPU board. This processing is used by breakpoint function.
Reserved instruction code exception	EXPEVT = 180	Branches to the CPU board and informs the occurrence of a reserved instruction code exception.
Slot illegal instruction exception	EXPEVT = 1A0	Branches to the CPU board and informs the occurrence of a slot illegal instruction exception.
CPU address error (load)	EXPEVT = 0E0	Branches to the CPU board and informs the occurrence of a CPU address error exception.
CPU address error (store)	EXPEVT = 100	Branches to the CPU board and informs the occurrence of a CPU address error exception.
DMA address error	EXPEVT = 5C0	Branches to the CPU board and informs the occurrence of a DMA address error exception.
NMI	EXPEVT = 1C0	Branches to the CPU board. This processing is used when execution is stopped by using the abort switch.
SCI-RXI	EXPEVT = 500	Buffers the characters received through the SCI. When the HALT code (H'12) is received, execution branches to the CPU board. This processing is used when execution is stopped by the [HALT] button.