

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.

Precautions on Using the H8SX E6000H Emulator

Thank you for using our product. There are the following notes on use of the H8SX E6000H Emulator.

1 Precautions on Using the Emulator

1.1 Controller Area Network (RCAN-ET) (while Using HS1527REPH60H or HS1544EPH60H)

If the order of IDs in a register can be changed depending on the value of the MCR15 bit in the master control register (MCR), the [IO] window displays the both patterns of register contents; when the MCR15 value is 0 and when the MCR15 value is 1, respectively. See the register that corresponds to the MCR15 value.

Table 1 Registers in which the Order of IDs Can be Changed

Register Name Written in Hardware Manual	Register Name Displayed in the [IO] Window	
	Value of MCR15: 0	Value of MCR15: 1
MB[x].CONTROL0H	MB[x]CONTROL0H_MCR15_0	MB[x]CONTROL0H_MCR15_1
MB[x].LAFMH	MB[x]LAFMH_MCR15_0	MB[x]LAFMH_MCR15_1

x = 0 to 15

1.2 Support of Flash Memory

This emulator does not emulate the flash memory.

1.3 Clearing the IRQ Status Register (ISR)

To clear the ISR, only use a bit manipulation instruction. Do not use a MOV instruction for clearing.

1.4 Data Transfer Controller (DTC)

1.4.1 Access to BSC, MDCR, SYSCR, SCKCR, SBYCR, MSTPCR, Interrupt Controller, DMRSR or EDMRSR by Using the DTC

Do not read/write data from/to the bus controller (BSC), mode control register (MDCR), system control register (SYSCR), system clock control register (SCKCR), standby control register (SBYCR), or module stop control registers (MSTPCR) by using the DTC. Also do not write data to the interrupt controller, DMA module request select register (DMRSR), or external bus transfer DMA module request select register (EDMRSR) by using the DTC.

1.4.2 Use of the DTC Prohibited when Little Endian is Used for Access

Do not use the DTC when little endian is used for access. Otherwise, data will not be read correctly.

1.5 DTC_PARALLEL_SET Command

If the size of the RAM is changed after specifying RAM as the area being used for DTC parallel accesses, the emulator automatically sets this area in IO.

1.6 Changing the Interrupt Control Mode

To change the interrupt control mode, ensure that the trace bit (T) of the extended control register (EXR) is 0.

1.7 Tracing

1.7.1 Trace Display

Trace information immediately before (up to six bus cycles) the device enters the sleep mode, all-module-clock-stop mode, software standby mode, or reset state cannot be displayed when [Halt] from the pop-up menu is selected.

1.7.2 Satisfaction of a Trace Stop Condition

If tracing stops due to satisfaction of the condition of [Trace Stop] immediately before (up to six bus cycles) the device enters the sleep mode, all-module-clock-stop mode, software standby mode, or reset state, the emulator does not enter the trace stop state until the device exits the said mode or state.

1.7.3 Tracing at the Occurrence of a Break by the Event Function

When a break occurs after setting the event function, the cycles of break processing will be acquired as trace information. The cycles that include the addresses listed in table 1 may be acquired when a break occurs. These are not the trace cycles displayed by user program execution.

Table 1 Addresses in Cycles of Break Processing

Address
H'0FFFFE62
H'0FFFFE64
H'0FFFFE0C

1.7.4 TRACE_FILTER Command

Do not specify [data] of the [status] option.

1.7.5 Analyzing Statistical Information in Trace Information

For trace information, do not specify [Timestamp-Difference] when analyzing statistical information.

1.8 Event Function

1.8.1 On-Chip Break Conditions in [Address] and [Bus/Area]

If an address bus value is specified as the address condition and [Data] is selected as the access type condition, a break occurs as the break condition is satisfied when the instruction at the address specified as the address condition is fetched.

1.8.2 Parallel Access Function and On-Emulator Break Condition in [Bus/Area]

When a DTC cycle is selected as the break condition (i.e. selecting [Don't Care] or [DTC] in [Bus State] of the [Bus/Area] dialog box), a parallel access causes a break as the break condition is satisfied.

1.8.3 Contention between the Parallel Access Function and a Break

When there is a contention between the parallel function and a break, "Parallel Access Error" appears on the [Debug] tab of the output window.

1.8.4 Display of a Break Source

When only the data bus condition or both of the data bus and read/write conditions is set as the on-emulator break condition, the break source will be displayed as [On Emulator Break] irrespective of the actual break source.

1.8.5 Halt Condition before Executing the Software Breakpoint and On-Chip Break

When execution is started from the address where the halt condition before executing the software breakpoint or on-chip break has been set, no break will occur even if the on-chip break or on-emulator break condition is satisfied after an instruction of the execution-start address is executed. A break will occur when the condition is satisfied again after two instructions are executed.

1.8.6 On-Chip Break Conditions in [Data] and [Bus/Area]

If only a data condition and an access type condition [Instruction prefetch] are specified, a break occurs as the break condition is satisfied when an instruction is fetched regardless of the specified data condition.

1.9 Performance Analysis

1.9.1 Access Count Of Specified Range Measurement

Do not use Access Count Of Specified Range Measurement with this product.

1.10 Custom Device Function

The on-chip ROM will not be enabled when you have selected an MCU that has no on-chip ROM area and an option other than [none] for [ROM] with the custom device function. The maximum value selectable for the on-chip ROM with the custom device function depends on the emulator in use. Do not specify a value that exceeds the maximum value selectable in the emulator.

1.11 Recovery of Software Breakpoints by the Session File

Information on software breakpoints that was saved in the session file is not available until a load module is loaded. If necessary, load a load module or set new software breakpoints.

1.12 Display of the Memory Content and Operation of Memory

1.12.1 Operating Memory during Execution of the User Program

The memory find, memory copy, memory compare, memory fill, and memory test functions are not available during execution of the user program. These command lines are not available either.

1.12.2 Memory Test

The content of the specified memory range will not be guaranteed after testing memory.

1.12.3 Updating the Memory Content during Execution of the User Program

If display of the [Memory] window is updated during user program execution under the following conditions, incorrect values will be displayed:

- Stepping (step-in, step-over, or step-out) is being performed.
- The MCU is in the all-module-clock-stop mode, software standby mode or reset.

Updating of the [Monitor] window or [Watch] window by reading values with the monitoring function is available even with those conditions.

1.12.4 Update of the [Monitor] Window

When any of the following conditions is satisfied, the [Monitor] window may display incorrect contents during user program execution.

- If the MCU enters the sleep mode or the software standby mode within six bus cycles after changing the memory contents by the user program, the contents may not be correctly displayed.
- The contents of the [Monitor] window will not be updated if [TASK Selection N] is selected in [TASK Selection] of [RTOS Support Function]. Do not use specify the [Monitor] window and [TASK Selection N] at the same time.

These notes also apply to realtime setting of the [Watch] window that uses the [Monitor] window.

1.12.5 FILE_LOAD Command

- Do not specify V for the FILE_LOAD command. Use the FILE_VERIFY command to verify memory.
- FILE_LOAD Command Options

To use the FILE_LOAD command, attach the options to replace or add symbols after the name of the file being loaded. The following table lists these options and their descriptions.

Options	Replace or append new symbol.
none (Default)	Load symbol with no debug information.
R	Replace all existing symbols with the new symbol
A	Append new symbol to symbol bank

1.12.6 FILE_SAVE Command

The content of memory cannot be saved in the Elf/Dwarf2 format.

1.12.7 Different On-Chip RAM Size in the Actual Device and the Emulator

The size of the on-chip RAM in the actual device and the emulator depends on the target MCU being selected (Table 2).

Table 2 Different On-Chip RAM Size in the Actual Device and the Emulator

Target MCU	Actual Device	Emulator
H8SX/1650	H'FF6000 to H'FFBFFF	H'FF6000 to H'FFBFFF
H8SX/1651	H'FF2000 to H'FFBFFF	H'FF2000 to H'FFBFFF
H8SX/1653	H'FF2000 to H'FFBFFF	H'FF2000 to H'FFBFFF
H8SX/1654	H'FF2000 to H'FFBFFF	H'FF2000 to H'FFBFFF
H8SX/1663	H'FF2000 to H'FFBFFF	H'FF2000 to H'FFBFFF
H8SX/1664	H'FF2000 to H'FFBFFF	H'FF2000 to H'FFBFFF
H8SX/1525	H'FF9000 to H'FFBFFF	H'FF8000 to H'FFBFFF
H8SX/1527	H'FF9000 to H'FFBFFF	H'FF8000 to H'FFBFFF
H8SX/1582	H'FF9000 to H'FFBFFF	H'FF8000 to H'FFBFFF
H8SX/1525R	H'FF9000 to H'FFBFFF	H'FF8000 to H'FFBFFF
H8SX/1527R	H'FF9000 to H'FFBFFF	H'FF8000 to H'FFBFFF
H8SX/1544	H'FF6000 to H'FFBFFF	H'FF6000 to H'FFBFFF
H8SX/1543	H'FF9000 to H'FFBFFF	H'FF8000 to H'FFBFFF
H8SX/1638	H'FFE000 to H'FFBFFF	H'FFE000 to H'FFBFFF
H8SX/1634	H'FF2000 to H'FFBFFF	H'FF2000 to H'FFBFFF
H8SX/1632	H'FF6000 to H'FFBFFF	H'FF6000 to H'FFBFFF
H8SX/1648	H'FFE000 to H'FFBFFF	H'FFE000 to H'FFBFFF
H8SX/1644	H'FF2000 to H'FFBFFF	H'FF2000 to H'FFBFFF
H8SX/1642	H'FF6000 to H'FFBFFF	H'FF6000 to H'FFBFFF
H8SX/1668R	H'FFE000 to H'FFBFFF	H'FFE000 to H'FFBFFF
H8SX/1664R	H'FF2000 to H'FFBFFF	H'FF2000 to H'FFBFFF
H8SX/1663R	H'FF2000 to H'FFBFFF	H'FF2000 to H'FFBFFF
H8SX/1658R	H'FFE000 to H'FFBFFF	H'FFE000 to H'FFBFFF
H8SX/1654R	H'FF2000 to H'FFBFFF	H'FF2000 to H'FFBFFF
H8SX/1653R	H'FF2000 to H'FFBFFF	H'FF2000 to H'FFBFFF
H8SX/1650C	H'FF6000 to H'FFBFFF	H'FF6000 to H'FFBFFF

1.13 Emulation

1.13.1 Temporary Breakpoints

- Run... (Run menu)

Do not set temporary breakpoints to an area wherein a software break (User Read-only) cannot be set.

1.13.2 Rate of Stepping

- To perform stepping continuously with the [Image View] or [Waveform] window open, do not select "0 seconds" as the delay.
- Do not perform stepping continuously if you have specified the realtime update of the [Image] or [Waveform] window.

1.13.3 Go to Cursor

The [Go to Cursor] function uses one unused on-chip breakpoint from Ch.1 to 3. Therefore, all on-chip breakpoints from Ch1 to 3 are used, the [Go to Cursor] function will be unavailable.

1.13.4 Note on Occurrence of Breaks

When a break occurs during execution of the user program with a dialog box open, the state of High-performance Embedded Workshop may not transit to the breaking state. In this case, close the dialog box and select [Halt Program] from the [Debug] menu or press the [Halt Program] button on the toolbar. You can check whether a break has occurred or not by the RUN lamp (LED) on the emulator station. The RUN lamp (LED) is illuminated while the user program is running.

1.13.5 Execution Status Display

- Execution Status Display on Status Bar

At memory access wait, the next fetched address is displayed, not the address where the access is halted.

1.13.6 Source-Level Execution

- Step

Even standard C libraries are executed. To return to a higher-level function, enter 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.

1.14 The Load/Save of the Session File

If the target clock was chosen at the time when the session file was saved and no target clock was supplied at the time when the session file was loaded, an error occurs and the emulator clock will be selected to initiate the emulator.

1.15 Creating a New Workspace

Note the following when the High-performance Embedded Workshop is being connected to the emulator after creating a new workspace with the user system connected to the emulator.

The emulator selects the default operating mode if the value of the mode input from the MD pins of the user system (MD1 and MD0 for the HS1527KEPH60H or HS1527REPH60H, or MD2 to MD0 for the HS1650EPH60H or HS1544EPH60H) is incorrect. However, [Target] is shown in [Mode] of the [Configuration Properties] dialog box. At connection for the second time or later, the default operating mode will be shown in [Mode].

Table 3 Default Operating Mode

Debugging Platform	Default Operating Mode
H8SX/1650 E6000H Emulator CPU 1600	4
H8SX/1653 E6000H Emulator CPU 1600	7
H8SX/1663 E6000H Emulator CPU 1600	7
H8SX/1527 E6000H Emulator CPU 1600	3
H8SX/1527R E6000H Emulator CPU 1600	3
H8SX/1544 E6000H Emulator CPU 1600	7
H8SX/1638 E6000H Emulator CPU 1600	7
H8SX/1648 E6000H Emulator CPU 1600	7
H8SX/1668 E6000H Emulator CPU 1600	7
H8SX/1658 E6000H Emulator CPU 1600	7

1.16 Assembling and Disassembling

1.16.1 Line Assembly

- Input Radix

The default for line assembly input is decimal regardless of the Radix setting. Specify H' or 0x as the radix for a hexadecimal input.

1.17 Address Space Size

In absolute addressing mode, specify the size (:16, etc.)

1.18 Note on [Register] Window Operation during the User Program Execution

During user program execution, double-clicking the [Register] window displays a dialog box that allows a change of the register contents. However, the operation to change the register contents is invalid during the user program execution.

1.19 Source File

1.19.1 Note on the Source File Location after Creating a Load Module

When the source file is moved after the load module has been created, the [Open] dialog box, which specifies the source file, may be displayed during debugging of the created load module. Select the correct source file.

1.20 Watch

1.20.1 Local Variables at Optimization

Depending on the generated object code, local variables in a C source file compiled with the optimization option enabled will not be displayed correctly. Check the generated object code in the [Disassembly] window.

1.20.2 Variable Name Specification

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

1.20.3 Automatic Update

- Automatic Update will not be available when the specified variable is outside the scope. Do not specify Automatic Update for a static or auto variable.
- Do not specify Automatic Update for a pointer variable whose value changes with a break that occurs in user program execution.
- Symbols of arrays, structures, unions, and classes, and the contents (addresses) of pointer variables are not automatically updated.
- When the variable for which Auto Update has been specified is outside the display range in the [Watch] window, the emulator does not detect whether the variable is within the scope or not. Even if the variable is in the scope when the program stops, the variable outside the display range in the [Watch] window may not be displayed by scrolling the window after execution of the program.
- Display of the global variables allocated to registers will not be automatically updated.

1.20.4 Display of R Marks

(1) The color of R marks will be incorrect if the user takes the following procedure (however, the variable itself will be correctly displayed).

- (a) Set the monitoring function to change the type of a variable outside the display range in the [Watch] window to update the value by reading with the monitoring function (R mark: Blue), not by reading the normal data (R mark: Black).
- (b) Start execution without changing the display range in the [Watch] window.
- (c) If the user changes the display range in the [Watch] window during execution, the color of the R mark for the variable (which should be updated by reading data with the monitoring function) remains black.

The variable mentioned in step (c), of which the R mark is displayed in black, will be updated by reading data with the monitoring function.

(2) The color of R marks will be incorrect if the user takes the following procedure (however, the variable itself will be correctly displayed).

- (a) Cancel the monitoring function to change the type of a variable outside the display range in the [Watch] window to update the value by reading the normal data (R mark: Black), not by reading with the monitoring function (R mark: Blue).
- (b) Start execution without changing the display range in the [Watch] window.
- (c) If the user changes the display range in the [Watch] window during execution, the color of the R mark for the variable (which should be updated by reading normal data) remains blue.

The variable mentioned in step (c), of which the R mark is displayed in blue, will be updated by reading normal data.

1.20.5 Synchronization with the [Monitor] window

The content of the [Watch] window will be undefined if you change the monitor settings in the [Monitor] window during execution of the user program while the [Monitor] window is used to monitor the addresses of the symbols registered in the [Watch] window.

1.21 Command Line Interface

- Overwrite File

In Command Line Interface, a file that has the same name as the output file is overwritten without asking the user.

- Log File

To change the target platform due to a change of the session while logging of the command line window, halt logging.

1.22 Monitoring Function

1.22.1 Changing Monitor Settings during User Program Execution

If you change the monitor settings in the [Monitor] window during execution of the user program, the initial values will be undefined.

1.23 Trigger Output Function

Trigger output by satisfaction of conditions is not available for the on-chip break function. To use the trigger output function, the setting must be made in [Trace Acquisition Condition Channel].

1.24 Differences between the Actual MCU and the Emulator

There are some differences in I/O registers between an actual MCU and the H8SX E6000H emulator. Note these differences when accessing the I/O registers.

I/O ports are in the input state at default. The I/O register contents indicate the emulator port state. When the user system interface cable is not connected, the read value is 'Undefined'.

In the emulator, accesses to the following registers for controlling the flash memory are invalid.

- RAM emulation register (RAMER)
- Flash code control/status register (FCCS)
- Flash program code select register (FPCS)
- Flash erase code select register (FECS)
- Flash key code register (FKEY)
- Flash transfer destination address register (FLPWCR: H'FFAC)

2 Specifications Specific to Emulator

2.1 Notes Specific to the H8SX/1650 E6000H Emulator

2.1.1 Mapping Emulator Memory and User Memory

When using the H8SX/1650 platform, emulator memory and user memory cannot be mapped in each of address area blocks listed below at the same time.

Block name	Address area	Block name	Address area
SB0	H'000000-H'00FFFF	LB0	H'100000-H'17FFFF
SB1	H'010000-H'01FFFF	LB1	H'180000-H'1FFFFF
SB2	H'020000-H'02FFFF	LB2	H'200000-H'2FFFFF
SB3	H'030000-H'03FFFF	LB3	H'300000-H'3FFFFF
SB4	H'040000-H'04FFFF	LB4	H'400000-H'7FFFFF
SB5	H'050000-H'05FFFF	LB5	H'800000-H'BFFFFF
SB6	H'060000-H'06FFFF	LB6	H'C00000-H'CFFFFF
SB7	H'070000-H'07FFFF	LB7	H'D00000-H'DFFFFF
SB8	H'080000-H'08FFFF	LB8	H'E00000-H'E7FFFF
SB9	H'090000-H'09FFFF	LB9	H'E80000-H'EFFFFF
SB10	H'0A0000-H'0AFFFF	LB10	H'F00000-H'F7FFFF
SB11	H'0B0000-H'0BFFFF	LB11	H'F80000-H'FFDFFF
SB12	H'0C0000-H'0CFFFF	LB12	H'FFE000-H'FFEFFF
SB13	H'0D0000-H'0DFFFF	LB13	H'FFF000-H'FFFEFF
SB14	H'0E0000-H'0EFFFF	LB14	H'FFFF00-H'FFFF7F
SB15	H'0F0000-H'0FFFFF	LB15	H'FFFF80-H'FFFFFF

2.1.2 Restrictions on Emulation of the H8SX/1653 or H8SX/1654

2.1.2.1 MD_CLK Pin

The MD_CLK pin is not connected to the MCU. In the [Configuration Properties] dialog box, select a device that matches the state of the MD_CLK pin.

Device	MD_CLK Pin
H8SX/1653_MD_CLK0	LOW
H8SX/1653_MD_CLK1	HIGH
H8SX/1654_MD_CLK0	LOW
H8SX/1654_MD_CLK1	HIGH

At initiation of the High-performance Embedded Workshop, H8SX/1653_MD_CLK0 is selected (i.e. the state of the MD_CLK pin is low).

2.1.3 Restrictions on Emulation of the H8SX/1663 or H8SX/1664

2.1.3.1 MD_CLK Pin

The MD_CLK pin is not connected to the MCU. In the [Configuration Properties] dialog box, select a device that matches the state of the MD_CLK pin.

Device	MD_CLK Pin
H8SX/1663_MD_CLK0	LOW
H8SX/1663_MD_CLK1	HIGH
H8SX/1664_MD_CLK0	LOW
H8SX/1664_MD_CLK1	HIGH

At initiation of the High-performance Embedded Workshop, H8SX/1663_MD_CLK0 is selected (i.e. the state of the MD_CLK pin is low).

2.1.4 Restrictions on Emulation of the H8SX/1668R, H8SX/1664R or H8SX/1663R

2.1.4.1 MD_CLK Pin

The MD_CLK pin is not connected to the MCU. In the [Configuration Properties] dialog box, select a device that matches the state of the MD_CLK pin.

Device	MD_CLK Pin
H8SX/1668R_MD_CLK0	LOW
H8SX/1668R_MD_CLK1	HIGH
H8SX/1664R_MD_CLK0	LOW
H8SX/1664R_MD_CLK1	HIGH
H8SX/1663R_MD_CLK0	LOW
H8SX/1663R_MD_CLK1	HIGH

At initiation of the High-performance Embedded Workshop, H8SX/1668R_MD_CLK0 is selected (i.e. the state of the MD_CLK pin is low).

2.1.5 Restrictions on Emulation of the H8SX/1658R, H8SX/1654R or H8SX/1653R

2.1.5.1 MD_CLK Pin

The MD_CLK pin is not connected to the MCU. In the [Configuration Properties] dialog box, select a device that matches the state of the MD_CLK pin.

Device	MD_CLK Pin
H8SX/1658R_MD_CLK0	LOW
H8SX/1658R_MD_CLK1	HIGH
H8SX/1654R_MD_CLK0	LOW
H8SX/1654R_MD_CLK1	HIGH
H8SX/1653R_MD_CLK0	LOW
H8SX/1653R_MD_CLK1	HIGH

At initiation of the High-performance Embedded Workshop, H8SX/1658R_MD_CLK0 is selected (i.e. the state of the MD_CLK pin is low).

2.2 Others

The circuits that interface the H8SX/1688 evaluation chip in the emulator to the user system include level-shift buffers and resistors. When connecting the emulator to a user system, adjust the user system hardware compensating for FANIN, FANOUT, and propagation delays. For details, see the user's manual of the emulator.

3 Commands Specific to High-performance Embedded Workshop

The following commands are specific to High-performance Embedded Workshop:

No.	Command	Function
(1)	hdi cmd	Executes an HDI command.
(2)	hdi expr	Evaluates an expression.
(3)	hdi memread	Reads data from the target system memory.
(4)	hdi memwrite	Writes a list of data items to the target system memory.
(5)	hdi progress	Sets the text and percentage shown on the progress bar.
(6)	hdi doevents	Allows background processing and other windows programs to execute while a TCL script is running.
(7)	Radix	Sets the default radix used when parsing numbers.
(8)	hdi msgbox	Displays a Windows message box.
(9)	hdi assemble	Assembles text to a list of bytes.
(10)	hdi getbrkcause	Returns a platform specific string giving the reason for the last user program break.
(11)	hdi islinkup	Returns true (non-zero) if the link to the debugging platform (emulator) is up.
(12)	hdi status	Returns a platform specific status line.

(1) hdi cmd

Execute a High-performance Embedded Workshop command

SYNOPSIS

```
hdi cmd command  
command
```

This command allows any of the High-performance Embedded Workshop commands to be executed from within a Tcl script. The second form is provided for convenience so that the programmer can freely mix High-performance Embedded Workshop and Tcl commands in the script without having to know which are High-performance Embedded Workshop commands.

Returns either an empty string if the command completed successfully, or an error message.

(2) hdi expr

Evaluate a expression

SYNOPSIS

```
hdi expr expression
```

This allows symbol names and register names to be used in the expression.

Returns the resulting integer value in hex, with a leading d'.

(3) hdi memread

Read data from the target system memory

SYNOPSIS

```
hdi memread space address count size
```

This command allows the Tcl script to read data from the target memory. The arguments are as follows:

space Specifies which address space is to be used. This argument is only of significance for processors which support multiple address spaces. For processors which only support a single address space, this argument should be zero.

address The address of the first data item to be read.

count The number of data items to read.

size The size of the data items to be read; 1 = byte, 2 = word, 4 = long word.

Returns a list containing the data items.

(4) hdi memwrite
Writes a list of data items to the target system memory.

SYNOPSIS

hdi memwrite space address size data

This command allows the Tcl script to write data to the target memory. The arguments are as follows:

space Specifies which address space is to be used. This argument is only of significance for processors which support multiple address spaces. For processors which only support a single address space, this argument should be zero.
address The address of the first data item to be written.
size The size of the data items to be read; 1 = byte, 2 = word, 4 = longword.
data A list of data items.

Returns an empty string on success, or an error string on failure.

(5) hdi progress
Set the text and percentage shown on the progress bar.

SYNOPSIS

hdi progress text percentage

Sets the status bar to show the text and percentage specified.

(6) hdi doevents
Allows background processing and other windows programs to execute while a TCL script is running. If you have a long loop it is a good idea to include this in the loop body..

SYNOPSIS

hdi doevents

(7) Radix
Sets the default radix used when parsing numbers.

SYNOPSIS

radix
 radix base

If the first form is used then the current default radix is returned.

(8) hdi msgbox
Display a Windows message box..

SYNOPSIS

hdi msgbox text [options]

Displays a message box containing "text". Options is a number representing the standard windows message box options:

- 0x0000 Display OK button
 - 0x0001 Display OK and Cancel buttons
 - 0x0002 Display Abort Retry and Ignore buttons.
 - 0x0003 Display Yes No and Cancel buttons.
 - 0x0004 Display Yes and No buttons.
 - 0x0005 Display Retry and cancel buttons.
 - 0x0010 Display a Hand icon.
 - 0x0020 Display a Question mark icon.
 - 0x0030 Display an Exclamation mark icon.
 - 0x0040 Display an Asterisk icon.
- These values may be Ored together. The default value is 0 (OK button).

The following values are returned:

- 1 OK button
- 2 Cancel button
- 3 Abort button
- 4 Retry button
- 5 Ignore button
- 6 Yes button
- 7 No button

(9) hdi assemble
Assemble text to a list of bytes.

SYNOPSIS

hdi assemble address instruction

Assembles the instruction for the current CPU at the specified address and returns a list of bytes. Does not load the code into memory.

(10) hdi getbrkcause
Returns a platform specific string giving the reason for the last user program break (stop)..

SYNOPSIS

hdi getbrkcause

(11) hdi islinkup
Returns true (non-zero) if the link to the debugging platform (emulator) is up.

SYNOPSIS

hdi islinkup

(12) hdi status
Returns a platform specific status line.

SYNOPSIS

hdi status linenumber

Returns one of the platform specific status lines as a string. The status lines are as displayed in the status window.

4 High-Speed Downloading of Files

The debugging function in the E6000H emulator is realized by serial communication with the target MCU. The H-UDI clock is used as the input clock in this serial communication. Set the H-UDI clock frequency as high as possible to improve performance in downloading. Note that, however, the H-UDI clock frequency must be equal to or lower than that of the peripheral module clock ($P\phi$) of the target MCU.

The following sections describe how to change the H-UDI frequency and procedure for high-speed downloading of files.

4.1 Changing the H-UDI Clock Frequency

Specify the CONFIGURE_PLATFORM (CPF) command of the command line.

Syntax:

```
CPF [HUDI (3.125MHz | 6.25MHz | 12.5MHz | 25MHz)]
```

HUDI: Keyword

3.125MHz | 6.25MHz | 12.5MHz | 25MHz: H-UDI clock frequency

4.2 Procedure for High-Speed Downloading of Files

Reset the target MCU.

1. If necessary, set the bus controller depending on the type of memory selected as the destination.
2. Set SCKCR to decide the frequencies of the system clock ($I\phi$) from the user system, peripheral module clock ($P\phi$), and external bus clock ($B\phi$).
3. Set the H-UDI clock frequency to be equal to or lower than the peripheral module clock ($P\phi$).
4. Download a file.
5. Set the H-UDI clock frequency as 3.125 MHz*.

Note: The initial value of the emulator's H-UDI clock frequency is 3.125 MHz. Be sure to set the H-UDI clock frequency back to 3.125 MHz when you execute the user program or perform debugging (except for downloading of a file). Failure to do so will cause unexpected operation.

Example: Command Input for High-Speed Downloading of Files

In this example, a file is loaded to emulation memory. High-speed loading is realized here by temporarily setting the peripheral module clock ($P\phi$) as 32 MHz and the H-UDI clock as 25 MHz, respectively.

Clock input by the user:	16-MHz emulator clock
System clock ($I\phi$):	32 MHz (16 MHz \times 2)
Peripheral module ($P\phi$):	32 MHz (16 MHz \times 2)
External bus clock ($B\phi$):	32 MHz (16 MHz \times 2)
Type of memory selected as the destination:	Emulation memory
Setting of the bus controller:	Not required

```

>RESET ...(1)
>MEMORY_EDIT H'FFFDC4 WORD
00FFFDC4 0222
>0111
00FFFDC6 4F00
>
>CPF HUDI 25MHz ...(3)
>FILE_LOAD ELF/DWARF2 C:\Sample\Sample.abs ...(4)
Module name 'C:\Sample\ Sample.abs'
Area loaded:
H'00000000 – H'00000003
H'00000400 – H'00000425
H'00001000 – H'000015EF
H'00002000 – H'00002139
>CPF HUDI 3.125MHz ...(5)

```

Time required for loading can be reduced by making this command line as a batch file.

5 Installation and Uninstallation

5.1 Uninstallation from [Tools Administration]

Note that a message dialog box appears for confirmation if the folder to be removed includes a file other than those installed. Even though this dialog box is displayed, press the [Ignore] button to continue uninstallation.

5.2 [Add/Remove Programs Properties] of the Control Panel

Note the followings:

- Uninstalling from [Add/Remove Programs Properties] of the control panel only removes folders and files of the product most recently installed if several products that conform to High-performance Embedded Workshop have been installed. Use the explorer to remove the folders and files of the rest of the products.
- Once uninstallation is attempted, even the products that were not removed may not operate correctly. If you want to use these products, re-install them.

6 Usage with Other Products

To install the SuperH™ RISC engine C/C++ compiler package or H8S, H8/300 series C/C++ compiler package after installing the E6000H emulator software for High-performance Embedded Workshop, select [High-performance Embedded Workshop] as the component to be installed in [Select Components] during installation of the compiler package.

7 Connecting the User System Interface Board

Before connecting the evaluation chip board and the user system interface board, check that pin 1 of each of the connectors is correctly aligned.

8 Reducing EMI Noise

To prevent EMI noise, enclose the evaluation chip board in a case at usage, as shown in the figure below. In the case of HS1527KEPH60H, HS1527REPH60H or HS1544EPH60H, enclose the User System Interface Cable head in a case.

The recommended material of the case is iron plated with nickel. For EMI compliance, we have used a case made of iron plated with nickel during testing.

The case must be large enough to include the evaluation chip board and the user system.

The opening in the case (on the top) should be large enough to let the emulator trace cable go through.

Note: EMI stands for Electrical Magnetic Interference.

