# SH7751R Solution Engine ™

# (MS7751RSE01)

# Overview

4th Edition

Hitachi ULSI Systems Co., Ltd.

MS7751RSE01-M

Cautions

1. Products and product specifications are subject to change without notice.
2. Hitachi ULSI Systems Co., Ltd. makes every attempt to ensure that the information of this manual is correct and reliable; however, Hitachi ULSI Systems Co., Ltd. takes no responsibility for damages or infringement of patent rights or other rights derived from the use of the information, product or circuit.
3. This manual does not authorize the use of the patent rights or other rights of third parties or Hitachi ULSI Systems Co., Ltd.
4. No one is permitted to reproduce or duplicate, in any form, the whole or part of this document without written approval from Hitachi ULSI Systems Co., Ltd.
5. The product overview described in this manual may differ slightly from the actual product.

# Notes on Using the SH Solution Engine

This section contains topics to be carefully read and considered when using the SH Solution Engine (referred to as Solution Engine) of the Hitachi ULSI systems.

**(Solution Engine Components)**

1. After opening the package, check the following items to confirm that everything is in place.
   a. Solution Engine main unit
   b. CD-ROM (A set of software, User's manual)
   c. Monitor EPROM for big endian
   d. Software license agreement

**(How to connect the Solution Engine)**

2. Before connecting the power supply, cables, development equipment and a daughterboard, the power must be turned off.
3. Before turning on the power following the connection of cables and other equipment, check all connections again to be sure that the wiring and polarity are correct.

**(Installation)**

4. Do NOT touch some parts on board during operating, because their temperature is high. Whenever you want to touch them, you must turn off the power and keep enough time to cool down.
5. The Solution Engine is developed for evaluation of products before they are put under the development phase. Do not incorporate the Solution Engine into any of the products.
6. Do not install the Solution Engine in an area subject to direct sunlight and other areas exposed to heaters or other source of high temperatures.
7. Do not choose area subject to extremely dusty condition.
8. Use care to keep the board free from contact with pieces of wire, soldering waste and other foreign matter.

**(Restriction)**

9. OS of the host system connected and checked before shipping the Solution Engine is Windows 95. In using Solution Engine by OS other than Windows 95, please use after checking enough.
10. Please connect the included power supply adapter to the Solution Engine. Although the power supply terminal stand is mounted in Solution Engine, please do not use it as much as possible.
11. When using Ethernet, be sure to connect through a hub. It may be unable to communicate, if Solution Engine is connected directly to PC via a cross cable.
12. Ethernet may encounter an abrupt drop of signal level and the line cannot be connected depending on the number of hub line connections or cable length. So that if the Solution Engine is connected to a hub, reduce the number of lines connected to a hub to a minimum to ensure reliable operation.
13. The ROM emulator controls reset and NMI while the ROM emulator is used. Do not actuate reset switch (SW2) and abort switch (SW3) during the ROM emulator is used.

14. The free warranty period of the system is one year from the delivered day. But it is limited to systems that are being used under normal condition such as environment condition, the way to operate the Solution Engine.
15. The warranty is void in the following cases.
    a. Any problems of system caused by natural disaster.
    b. Systems that are modified and repaired by user
    c. Any problems caused by improper handling
16. Do not reprovide the Solution Engine to the people who use the Solution Engine to hinder international peace and safety and do not use the Solution Engine for such aims personally and do not have third parties use the Solution Engine for such aims. For exporting the Solution Engine, follow Foreign Exchange and Foreign Trade Control Low and prescribed procedure.

# Components of the Solution Engine

Open the package and check the contents to match against the packing list. Table 0.1 lists the components of the Solution Engine. Figure 0.1 shows the contents of the Solution Engine.

Table 0.1 Solution Engine components

| No | Item | Description |
|---|---|---|
| 1 | SH7751R Solution Engine | SH7751R Solution Engine<br>Hardware |
| 2 | CD-ROM (Software, User's manual) | C compiler (Trial Version), driver software source file, various header file, User's manual |
| 3 | Monitor EPROM | Monitor EPROM for big endian Vx.xB ×2(included) |
| 4 | Software License Agreement | Condition to use software |

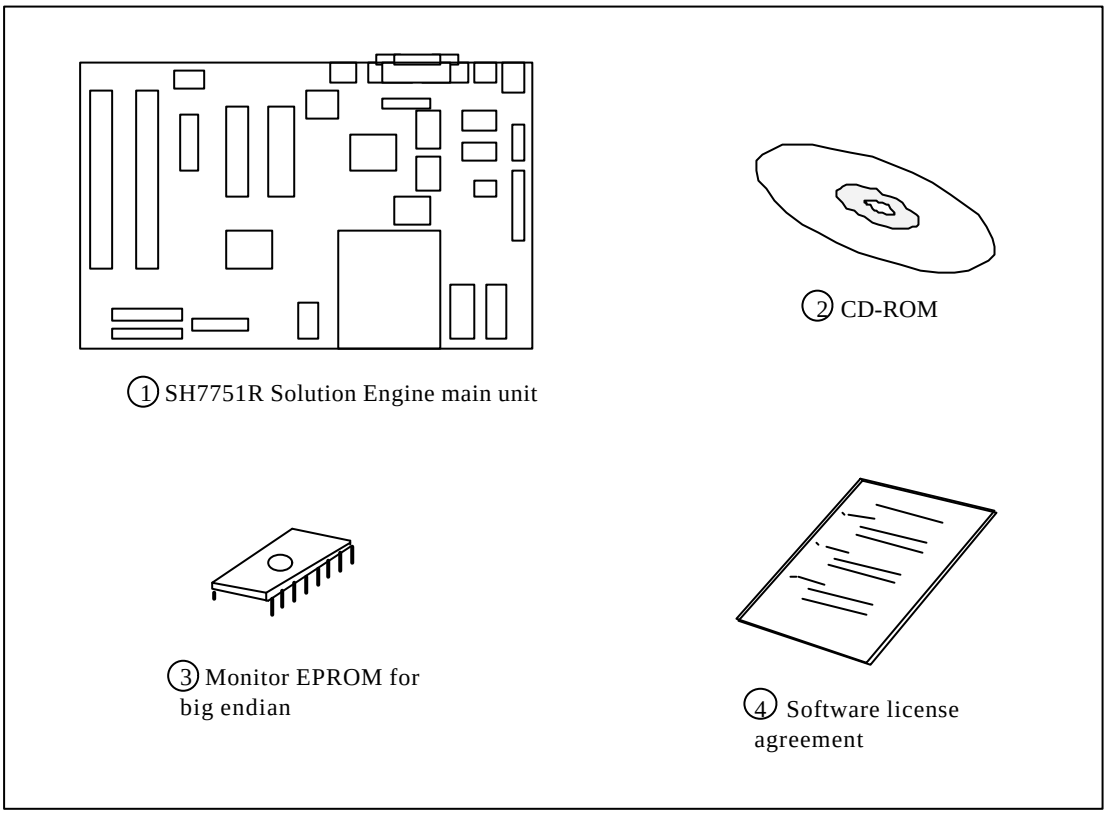



Figure0.1 Components of the Solution Engine

# Table of Contents

# 1. Features

The Solution Engine is a system capable of efficiently developing software and hardware for systems that employ the Hitachi SH7751R (SH-4) 32-bit SuperH series microcomputer.

## 1.1. Features of the Solution Engine

The Solution Engine has the following features.

1. All the information concerning this board including the circuit diagrams, various connector specifications, and the internal logic of the FPGA used with the board are contained in the manual.
2. The specification of this board is a summary of the proposals presented by Real Time OS manufacturers and many middleware development manufacturers. This is why hardware is configured so as to render OS and middleware easily portable.
3. Ethernet controller, PCMCIA controller, serial controller and other peripherals are off-the-shelf purchases.
4. Because Ethernet, PCMCIA, serial controllers and similar hardware are loaded on the board, system development is easier with these pieces of hardware applied.
5. The Solution Engine has the expansion slot outputting address bus and control signals of the SH7751R so that a user's hardware can be connected.
6. The Solution Engine has an I/O bus which carries an SH7751R port and the output of the timer output terminal.
7. The Solution Engine has CPU bus interface connector so as to trace SH7751R bus signal.

## 1.2. Debugging Function

The Solution Engine has a monitor program on the board. The monitor program has the following debugging functions.

1. Execution and pause of user programs

   The program can be executed from an optional address. When the following condition arises, the user program halts.

   a. When a break point is detected.
   b. When the Reset switch or Abort switch is pressed.

2. Display and change of register contents

   The contents of the general-purpose register can be displayed and changed as required.

3. Display and change of memory contents

   Memory contents can be expressed in mnemonic or hexadecimal numbers and changed as required.

## 1.3. System Configuration

Figure1.1 shows a system configuration of the Solution Engine. Figure1.2 shows an external overview of the Solution Engine.

Connect a host system, a modem and an Ethernet Hub to the Solution Engine according to the debugging environment and peripherals such as a modem.

User must prepare a host system, a modem and an Ethernet hub.

The host system, the modem, the Ethernet hub and power supply used to check the operation before shipping are as follows.

(Host system)
Hitachi FLORA310 and 330 (Windows 95 machine with 9-pin serial connector)
(Modem)
Microcom V.34ES II
(Ethernet Hub)
8-Port Ethernet Hub
(Power supply)
ATX power supply

[Notes]

Ethernet may encounter an abrupt drop of signal level and the line cannot be connected depending on the number of hub line connections or cable length. When connecting the Solution Engine to the hub, reduce the number of lines connected to the hub to a minimum to ensure reliable operation.

* Windows is a trademark of U.S. Microsoft Corporation.

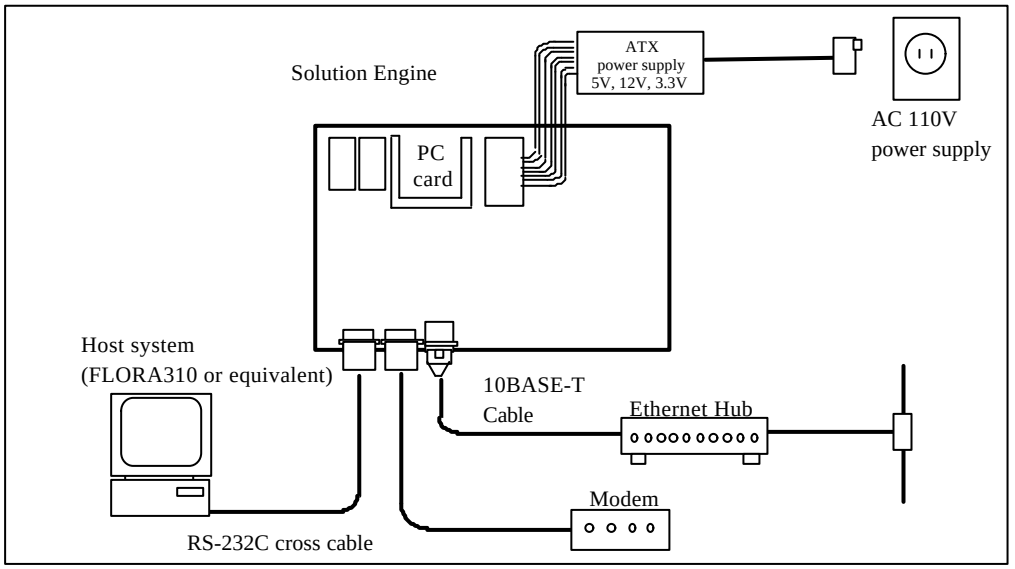



Figure1.1 System configuration of the Solution Engine

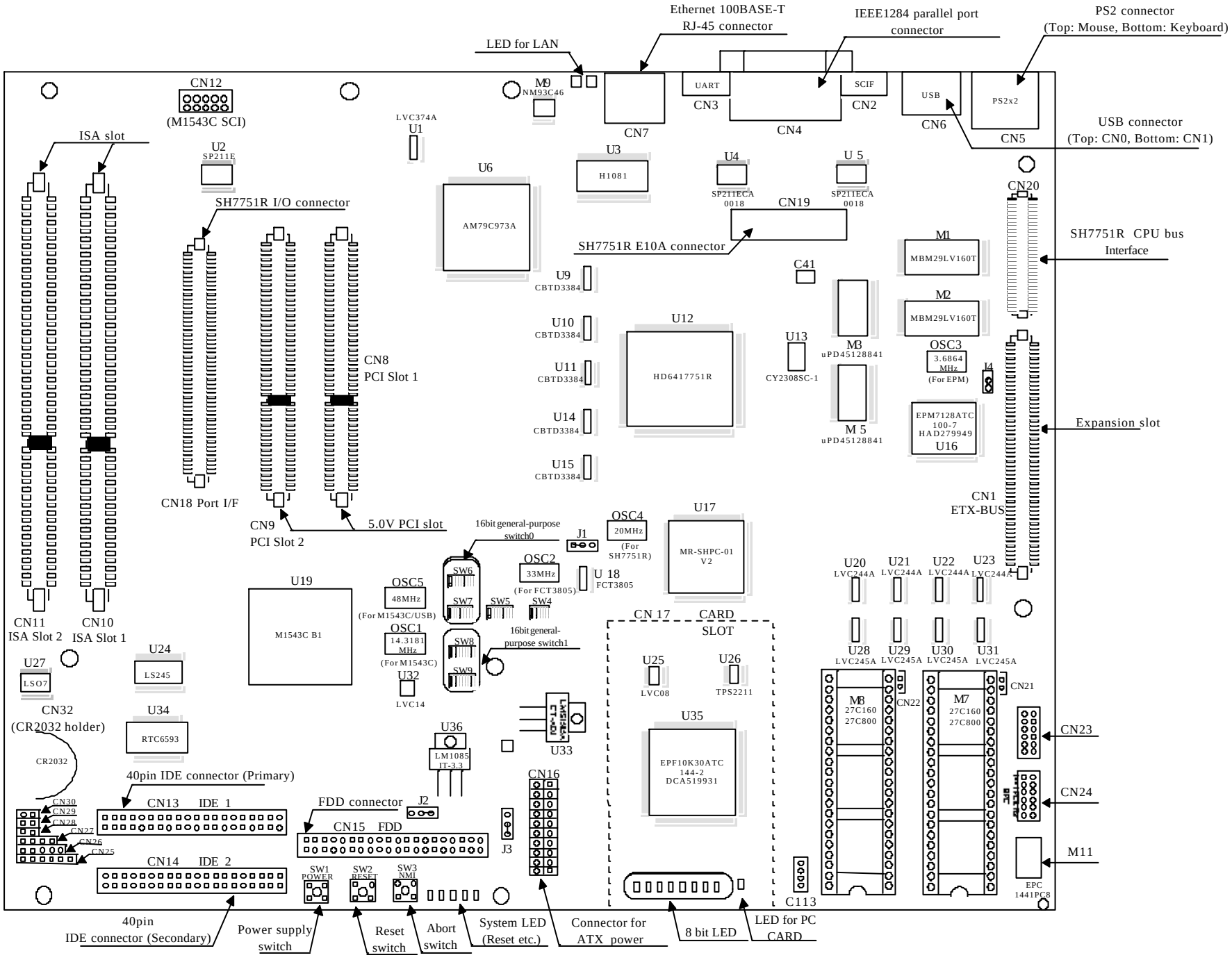



**Figure1.2 SH7751R Solution Engine External view**

## 1.4. Software Configuration

The Solution Engine has a monitor program in EPROM. The monitor program displays memory data and executes programs transferred to user memory. The user program can be executed and evaluated by connecting the host system.

For connection between host system and the Solution Engine, terminal software such as hyper terminal mounted as a standard accessory of Windows 95 can be used.

The source programs input by using various editors can be converted into machine language by using the C compiler, the assembler, the linkage editor and the object converter.

Figure1.3 shows software configuration when the Solution Engine is connected to the host system. For more details on the functions and the usage of the supplied software, refer to ReadMe.txt in the CD-ROM.

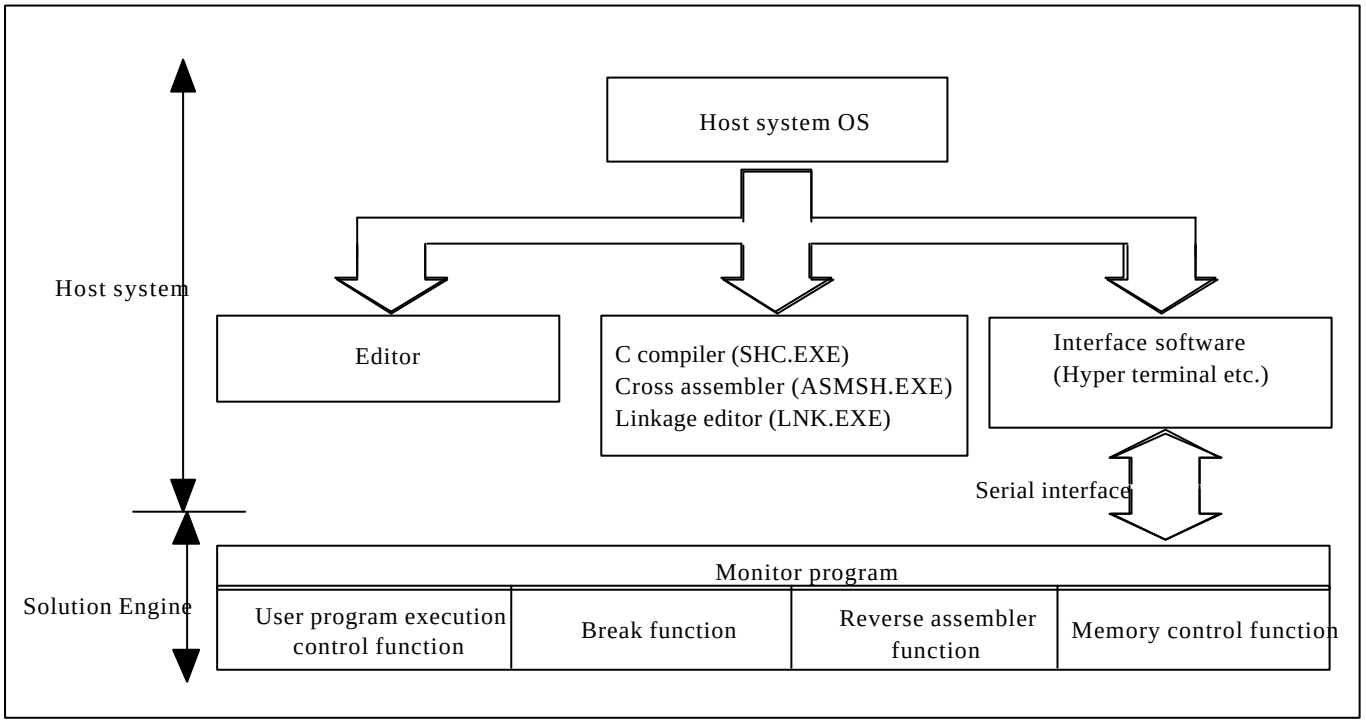



Figure 1.3 Software Configuration when connected to the Host System

## 1.5. Solution Engine Specifications

Table1.1 lists the functional specifications of the Solution Engine. Table1.2 lists the specifications for the power supply, dimensions and environmental factors.

Table 1.1 Functional Specifications of the Solution Engine

| Item | | Specifications |
|---|---|---|
| Subject device | | SH7751R(SH-4 PCI) |
| System Clock | | Operation frequency: Internal 240MHz, External: 60MHz(Maximum 81MHz*)<br>(20MHz oscillation module is equipped)<br>Oscillation module model name: SG-8002DC-20M-PTCB(SEIKO-EPSON) |
| User memory | | 64-MByte SDRAM(Cycle time: 10ns)<br>SDRAM model name: uPD45128841G5-A75-9JF (ELPIDA) |
| ROM | Flash ROM | 4-Mbyte<br>Flash ROM model name: MBM29LV160T-90PFTN (Fujitsu) |
| | EPROM | 2-Mbyte(mounted)<br>EPROM model name: MX27C8100PC-10 (MACRONIX) |
| Ethernet | | 10/100BASE-TX 1ch<br>Controller model name: Am79C973AVC(AMD) |
| PCMCIA | | 1 slot<br>Controller model name: MR-SHPC-01 V2 (Marubun) |
| Serial Interface | | 1ch<br>Controller: SH7751R on-chip SCIF |
| Super I/O<br>(Expansion Board) | | Serial 2ch, Parallel 1ch, IDE 2ch, FDC 1ch, USB 2ch, PS2(Keyboard/Mouse) 2ch mounted<br>Controller model name: M1543C B1(ALi) |
| Monitor Program | Host System | FLORA310 or equivalent (Windows95 or Windows98 is equipped) |
| | Host Interface | RS-232C interface 9 pin connector used<br>(Transfer speed: 9600, 19200, 38400, 115200bit/s) |
| | Command | 17 commands<br>ML(Memory Load), RR(Register Read), RW(Register Write),<br>RC(Register Clear), ME(Memory Edit), G(Go),<br>BS(Breakpoint Set) and etc. |
| Components | | Solution Engine, CD-ROM (User's manual) and etc. |

(Note)* MS7751RSE01 can be execute external 81MHz except for PCMCIA as maximum. 66MHz specification.

Table 1.2 Power Supply, Dimensions and Environmental Factors of the Solution Engine

| Item | Specifications |
|---|---|
| Environment | Operating conditions<br>- Temperature: 10-35 degree C<br>- Humidity: 30-85%RH (no condensation)<br>- Ambient gas: Should not have corrosive gas |
| Operating voltage | DC3.3V, 5.0V, 12V (Off-the-shelf power supply for ATX should be used) |
| Current consumption | 1A (temporary value) |
| External dimension | ATX size<br>304.8mm×243.84mm |

# 2. Setting the Solution Engine

After opening the package, set the Solution Engine as follows.

1. Choosing the debugging environment

   The Solution Engine has a monitor program stored in EPROM. Connect the host system equivalent of FLORA310 to use the monitor program.

   The monitor program is an implementation of the basic functions including reference and change of memory data and execution of programs. Use E10A emulator for trace of user programs and other debugging.

2. Connecting a daughterboard

   When using a daughterboard, connect the cable to the expansion slot (CN1) on the Solution Engine.

3. Connecting the modem

   Connect the modem to M1543C B1 COM1 connector (CN3). M1543C B1 COM1 connector outputs all signals necessary for connection of the modem.

4. Connecting the LAN

   Connect the LAN to the RJ-45 connector (CN7) via the Ethernet hub.

5. Connecting the I/O board

   Use the I/O connector (CN18) to use SH's general-purpose I/O port.

   The I/O connector outputs all general-purpose ports of the microcomputer.

6. Setting Jumper Pins and DIP SW

   Set jumper pins and the DIP SW according to the operating condition.

7. Connecting the power supply

   Connect power supply cable of ATX power to CN16. Do not connect to CN16 while ATX power supply remains connected to AC110V.

[Notes]

(1) Before connecting ATX power supply to the 110V AC power following upon completion of Steps 1 through 7 above, recheck that connection of the board and a cable, and setting of jumper pins and DIP switches are correct.

(2) When using Ethernet, be sure to connect through HUB. It may be unable to communicate, if the Solution Engine is connected directly to PC via a cross cable.

Figure 2.1 shows the procedure to install the Solution Engine.

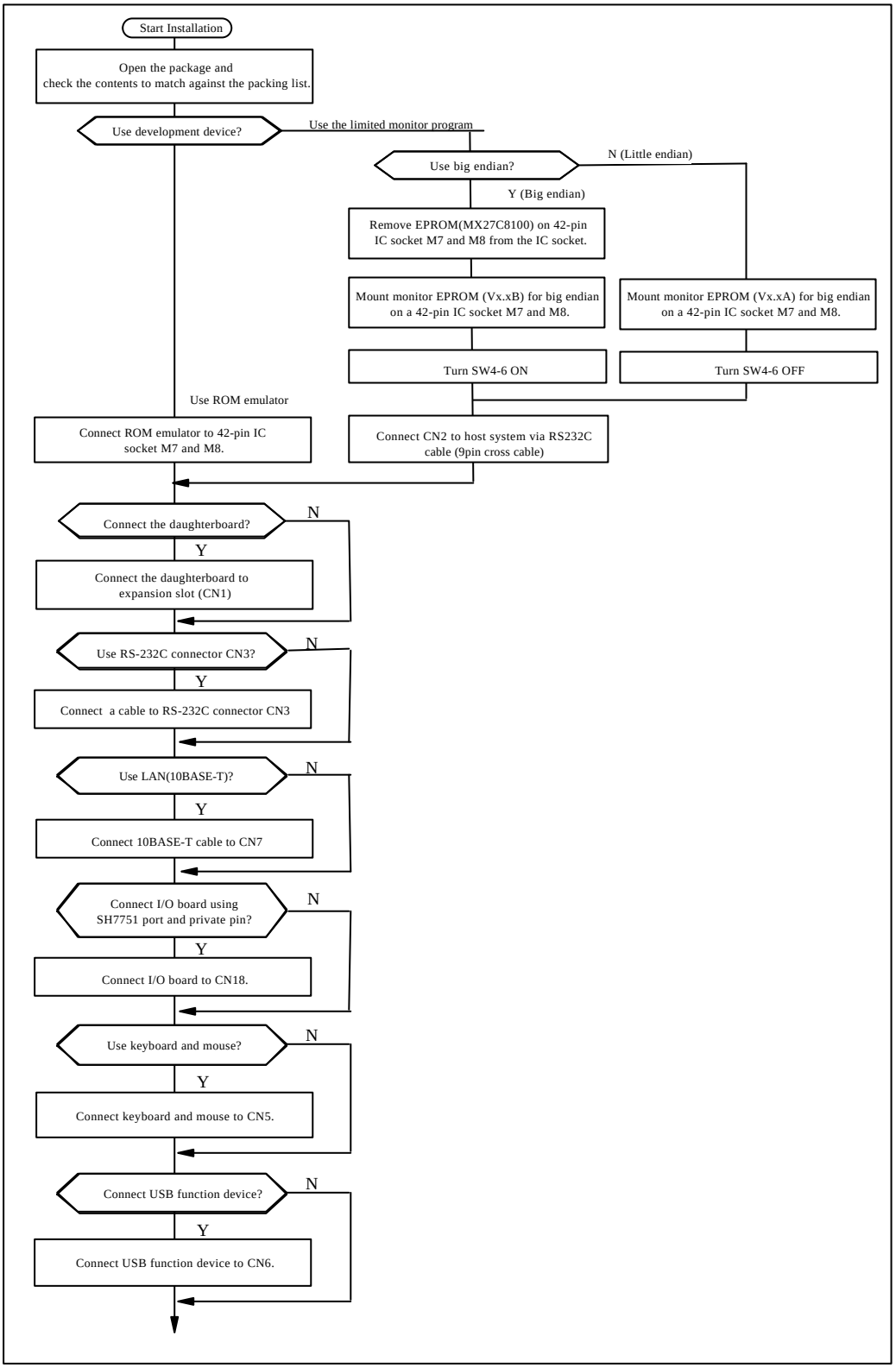



Figure 2.1 Installing the Solution Engine

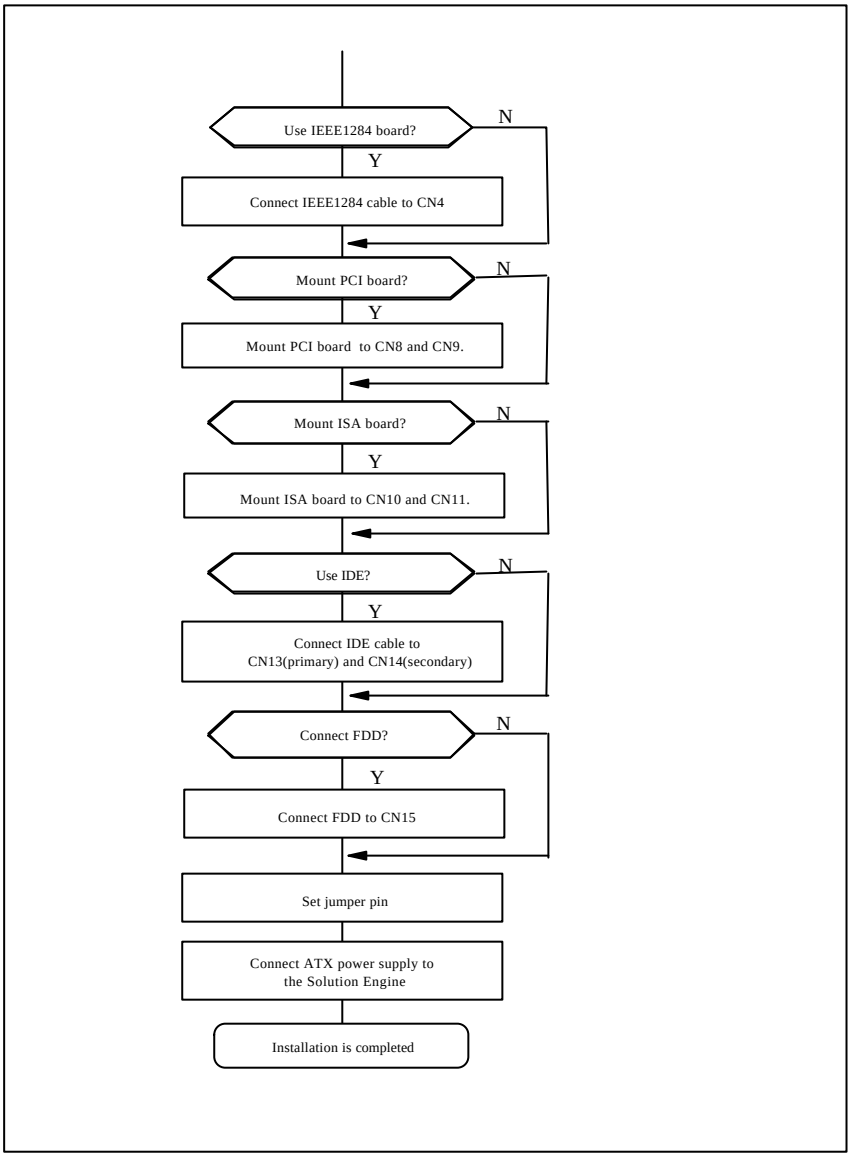



Figure2.1 Installing the Solution Engine

## 2.1. Connecting the host system

To use the monitor program, connect the host interface connector (CN2) to the host system via an interface cable. Figure2.2 shows how to connect the host system.

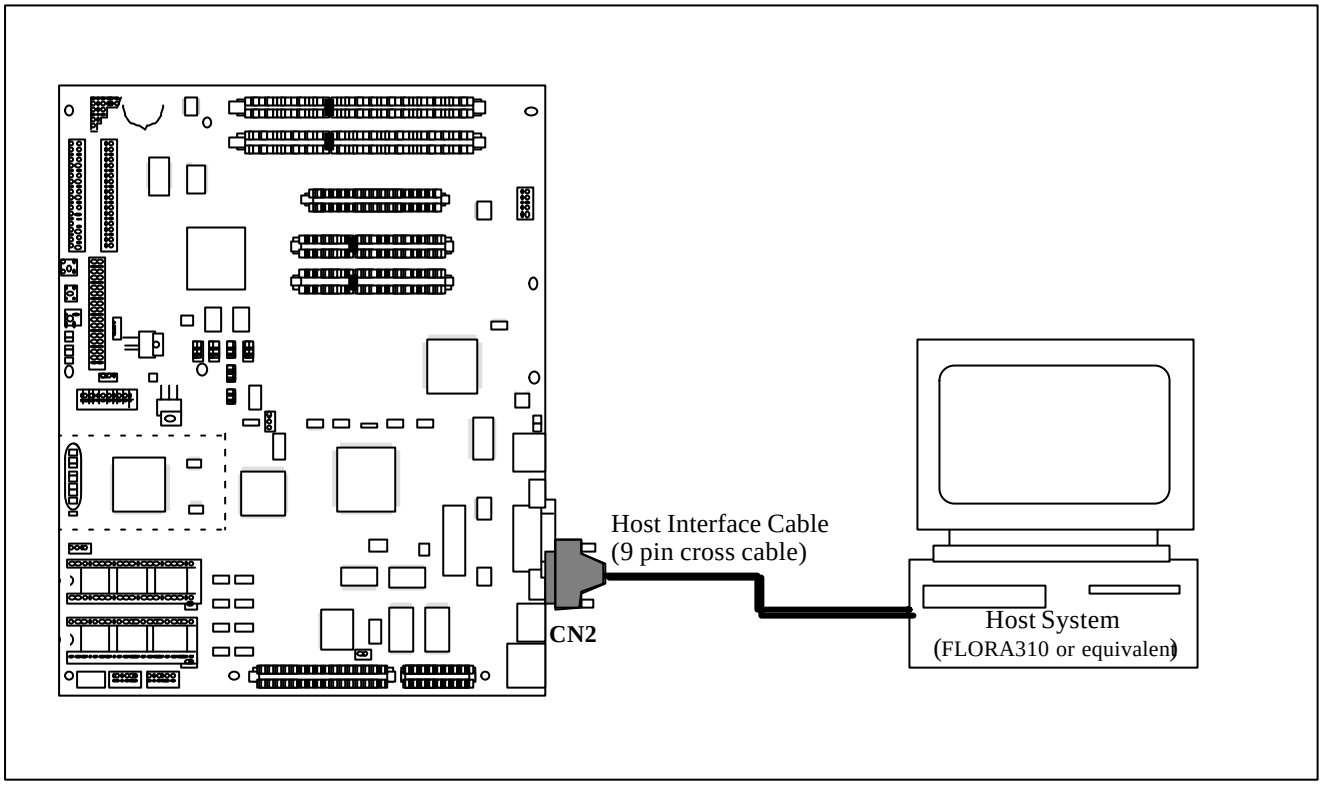



Figure 2.2 Connecting the host system

## 1. Host Interface Cable

For example, Figure 2.3 shows the wire connection when FLORA310 is connected to the Solution Engine. The Solution Engine can be connected to the host system via an off-the-shelf 9-pin cross cable.

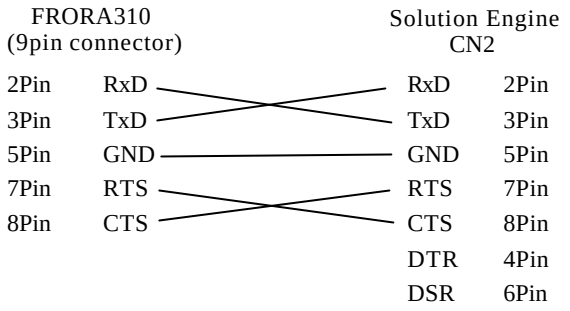


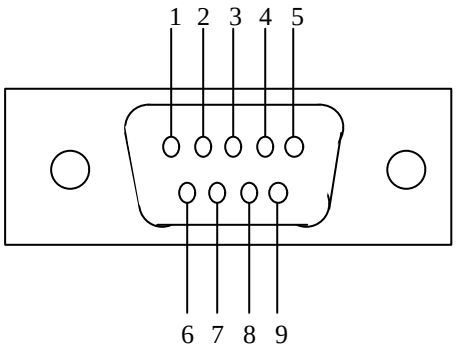

Figure 2.3 Wire connection between FLORA310 and the Solution Engine

## 2. Transfer Speed Setting

9600, 19200, 38400 and 115200 bit/s can be selected as a transfer speed with DIP switches (SW5-1, SW5-2) on the Solution Engine. Set the DIP switch according to the transfer rate. For specifications of the DIP switch, refer to Section 3.1 (4), " DIP switch(SW5) for setting baud rate".

## 3. Host Interface Connector (CN2)

Figure 2.4 shows the pin assignments of the host interface connector (CN2) and the list of signals.




| Pin | code | I/O | Name |
|---|---|---|---|
| 1 | (NC) | — | — |
| 2 | RxD | Input | Receive data |
| 3 | TxD | Output | Send data |
| 4 | DTR | Output | data terminal ready |
| 5 | GND | — | ground |
| 6 | DSR | Input | data set ready |
| 7 | RTS | Output | Request to send |
| 8 | CTS | Input | Clear to send |
| 9 | (NC) | — | — |

Figure 2.4 Pin Assignments (CN2)

## 2.2. Connecting the E10A emulator

This Solution Engine has a debugging chip(*Note) on the SH7751R, and SH7751R E10A emulator can be used. Figure2.5 shows how to connect the E10A emulator.

The PCMCIA card emulator that is main unit of SH7751R E10A emulator can be connected to the connector (CN19) via H-UDI port (Hitachi-User Debug Interface).

The E10A emulator connectable to the Solution Engine is as follows.

For more details on the connecting method and the E10A emulator setup, refer to the following manual.

Hitachi Co., Ltd. E10A emulator
HS7751RKCM02H(PCMCIA)

*Note: Debugging chip is same as actual chip.

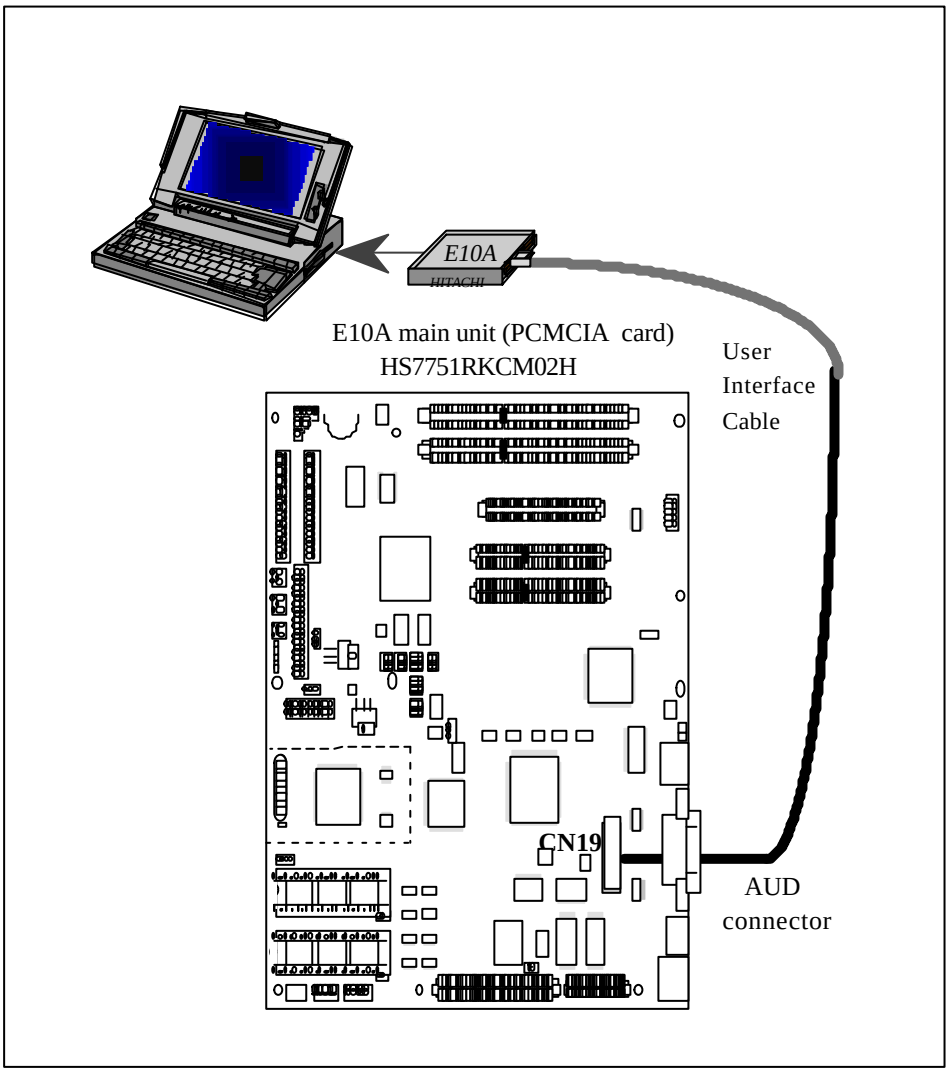



Figure 2.5 Connecting the E10A emulator

## 2.3. Connecting the power supply

### 1. Connecting ATX power supply

SH7751R Solution Engine uses ATX power supply (option product) as a power supply. Connect the power supply to AC110V as shown in figure2.6.

[Notes]

Before connecting the power adapter, recheck the board and cable are correctly connected, and check the jumper pins and DIP switch are correctly set.

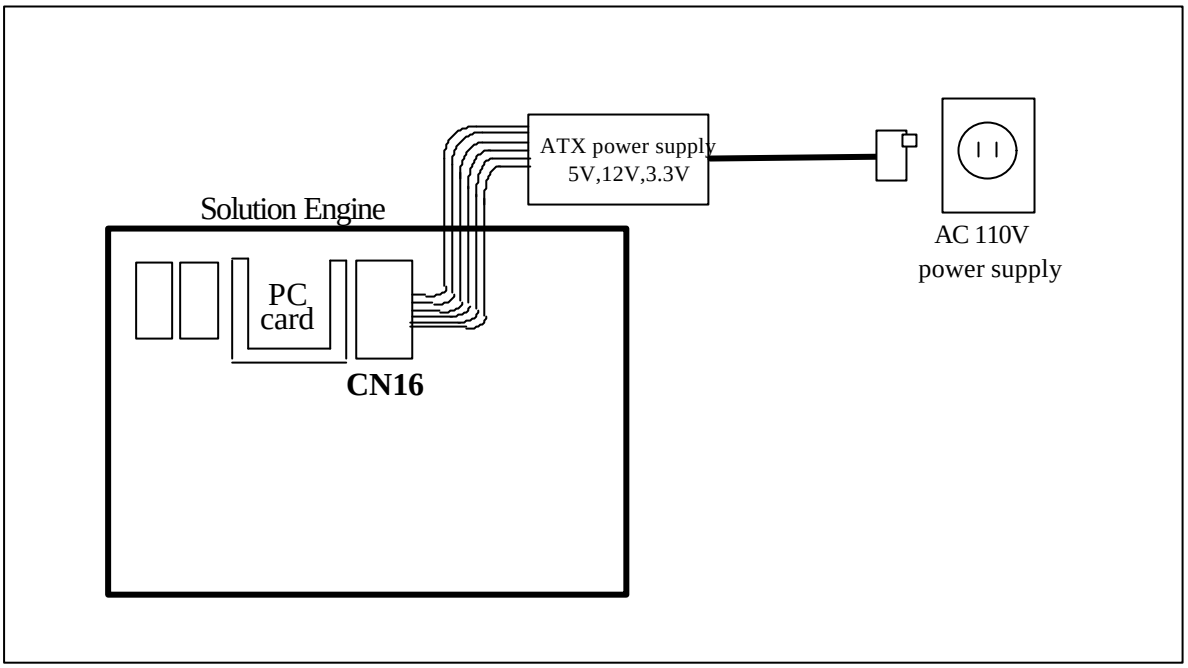



Figure 2.6 Connecting the power source

# 3. Switch Functions

## 3.1. Switch (SWn) functions

### 1. Power supply switch (SW1)

This switch is to turn ON and OFF power supply of SH7751R Solution Engine. This switch is a push button switch. Power supply is turned to ON or OFF by pushing this switch.

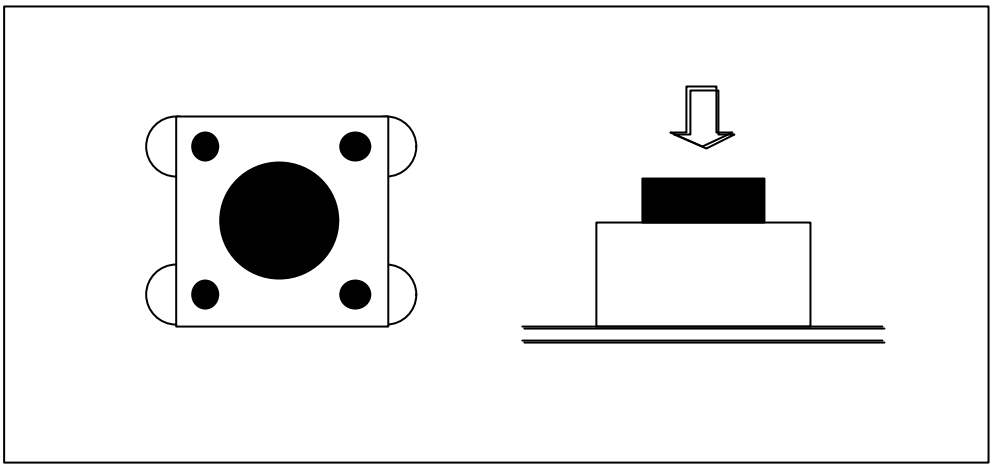


Figure 3.1 Power supply switch

### 2. Reset Switch (SW2)

This switch is to reset microcomputer. This switch is a push button switch. The microcomputer is reset by pushing this switch. Reset is cleared by releasing this switch

[Notes]

While ROM emulator is used, ROM emulator controls reset of the microcomputer. Do not operate the reset switch when connecting the ROM emulator.

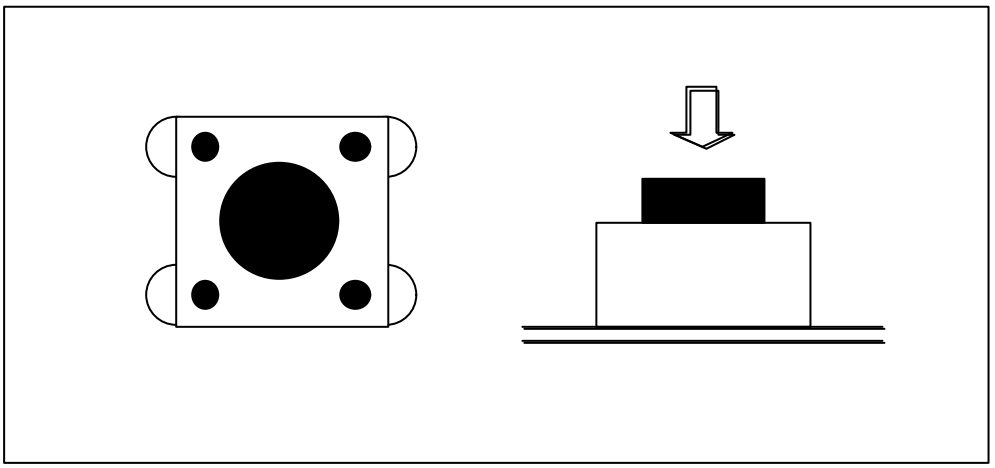


Figure 3.2 Reset switch

### 3. Abort switch(SW3)

This switch controls NMI pin of the microcomputer. This switch is a push button switch. NMI pin is turned to Low by pushing this switch. NMI pin is turned to High by releasing this switch.

[Notes]

ROM emulator controls NMI pin while ROM emulator is used. Do not operate the abort switch when connecting ROM emulator.

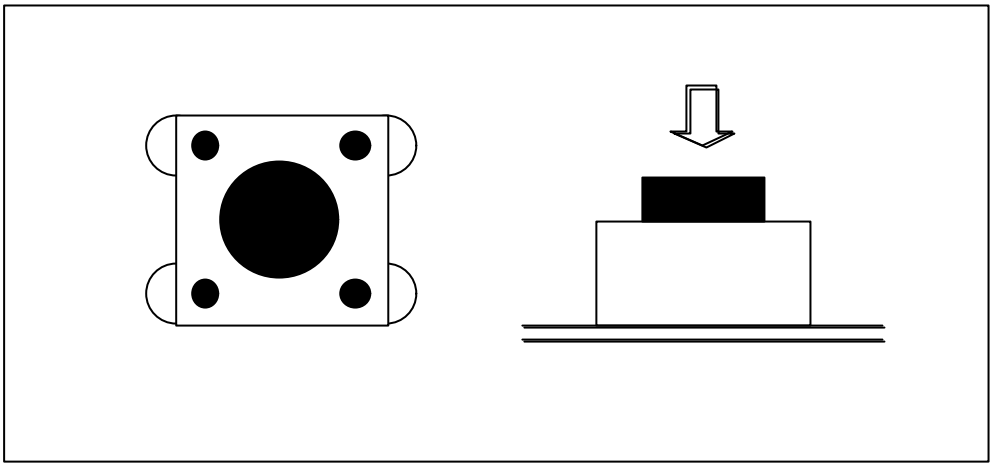


Figure 3.3 Abort switch (SW3)

## 4. DIP switch for setting microcomputer operating mode(SW4)

Figure3.4 shows the DIP switch for setting microcomputer operating mode (SW4).

Table3.1 shows switch functions. This switch is connected to mode pins (MD0-MD5) of the SH7751R. This switch can select the operating mode and endian as shown in table3.1. This switch must be switched while the power supply is in OFF state.

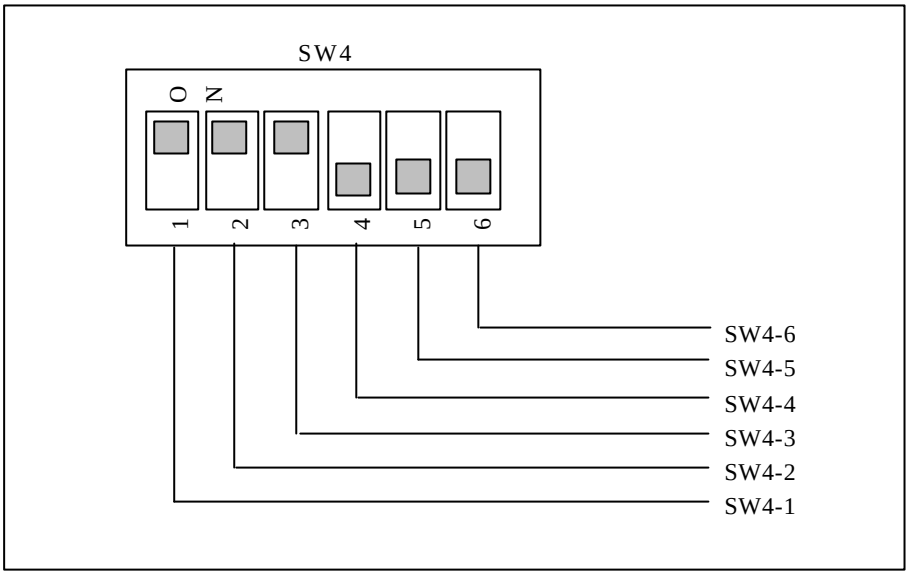



Figure3.4 DIP switch for setting microcomputer operating mode(SW4)

Table3.1 Functions of SW4

<table>
<tr><th>SW</th><th>Name</th><th>Function</th></tr>
<tr><td>SW4-1<br>SW4-2<br>SW4-3</td><td>Microcomputer operating mode setting switch</td><td>SW4-1 to SW4-3 is connected to mode pins MD0 to MD2, and this switch selects clock operating mode of the SH7751R.
<table>
<tr><th>SW4-1</th><th>SW4-2</th><th>SW4-3</th><th>MD[0:2]</th><th>Clock operating mode</th><th>remarks</th></tr>
<tr><td>ON</td><td>ON</td><td>ON</td><td>000</td><td>Mode0</td><td>At shipment</td></tr>
<tr><td>OFF</td><td>ON</td><td>ON</td><td>100</td><td>Mode1</td><td></td></tr>
<tr><td>ON</td><td>OFF</td><td>ON</td><td>010</td><td>Mode2</td><td></td></tr>
<tr><td>OFF</td><td>OFF</td><td>ON</td><td>110</td><td>Mode3</td><td></td></tr>
<tr><td>ON</td><td>ON</td><td>OFF</td><td>001</td><td>Mode4</td><td></td></tr>
<tr><td>OFF</td><td>ON</td><td>OFF</td><td>101</td><td>Mode5</td><td></td></tr>
</table></td></tr>
<tr><td>SW4-4<br>SW4-5</td><td>Area0 bus width select switch</td><td>SW4-4 and SW4-5 are connected to mode pins MD3 and MD4, and select bus width of Area0 (CS0). Bus width is fixed to 32-bit. SW4-4 and SW4-5 must be used in OFF state.</td></tr>
<tr><td>SW4-6</td><td>Endian select switch</td><td>SW4-6 is connected to mode pin MD5 of the SH7751R, and select big endian or little endian.<br>[ON]<br>Big endian<br>[OFF] (At shipment)<br>Little endian</td></tr>
</table>

## 5. DIP Switch for setting baud rate(SW5)

Figure3.5 shows DIP switch for setting the baud rate (SW5). Table3.2 lists the functions of the switch. As listed in Table3.2, this switch can select the baud rate of the SH7751R on-chip SCIF and ROM placed at area0.

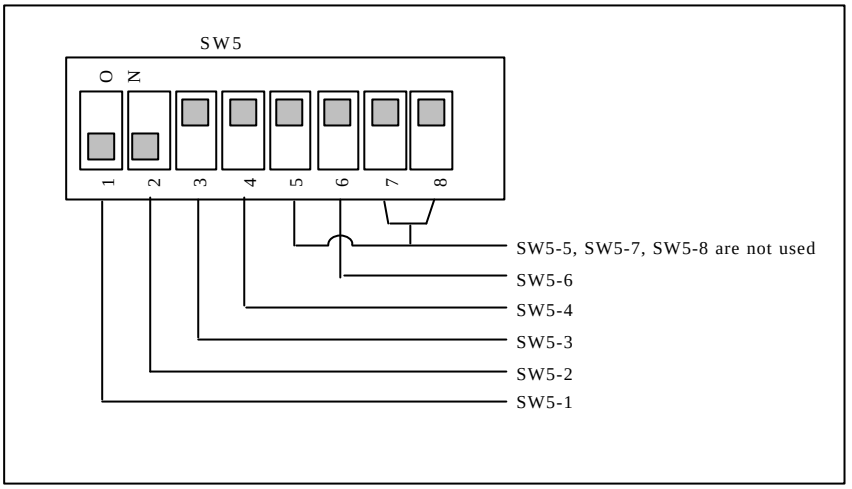



Figure 3.5 DIP Switch for setting the baud rate(SW5)

Table3.2 SW5 Functions

| SW | Name | Function |
|---|---|---|
| SW5-1<br>SW5-2 | SH7751R SCIF baud rate select switch | Select the baud rate of SH7751R on-chip SCIF2 (SCI with FIFO).<br>[SW5-1: OFF, SW5-2: OFF] (At shipment)<br>9600bit/s<br>[SW5-1: ON, SW5-2: OFF]<br>119200bit/s<br>[SW5-1: OFF, SW5-2: ON]<br>38400bit/s<br>[SW5-1: ON, SW5-2: ON]<br>115200bit/s |
| SW5-3<br>SW5-4 | ROM select switch | Select ROM placed at h'0000000-h'00FFFFFF and h'01000000-h'01FFFFFF.<br>[SW5-3: ON, SW5-4: ON](At shipment)<br>h'00000000-h'003FFFFF: EPROM<br>h'01000000-h'013FFFFF: Flash ROM<br>[SW5-3: OFF, SW5-4: ON]<br>h'00000000-h'003FFFFF: Flash ROM<br>h'01000000-h'013FFFFF: EPROM<br>[SW5-3: ON, SW5-4: OFF]<br>h'00000000-h'003FFFFF: CPU bus I/F connector(CN20)<br>h'01000000-h'013FFFFF: Flash ROM<br>[SW5-3: ON, SW5-4: OFF]<br>h'00000000-h'003FFFFF: CPU bus I/F connector(CN20)<br>h'01000000-h'013FFFFF: EPROM |
| SW5-6 | CS6 on-board resource select switch | Select whether to use peripheral LSI of CS6 area on the Solution Engine.<br>[SW5-6: ON] (At shipment)<br>Use peripheral LSI of CS6 area.<br>[SW5-6: OFF]<br>Peripheral LSI (MR-SHPC-01) placed at CS6 area is allocated at CS1.<br>All space of CS6 is allocated to expansion slot.<br>This switch can be used for evaluating a daughterboard using CS6. |
| SW5-5<br>SW5-7<br>SW5-8 | For test(Not used) | This switch is for testing.<br>[SW5-5: ON, SW5-7: ON, SW5-8: ON] (At shipment)<br>Do not change the factory-shipped setting. |

## 3.2. Jumper Pin (Jn) Functions

### 1. Test Jumper1 (J1)

This jumper is the jumper for testing SH7751R PCIC. Table 3.3 shows the function of the jumper for testing SH7751R PCIC.

This jumper must be used while pins 1-2 are closed.

When pins 2-3 are closed, microcomputer does not work.

Table3.3 Functions of the jumper for testing SH7751R PCIC (J1)

| Jumper Pin name | Jumper Pin Connected State | Function |
|---|---|---|
| J1 | J1<br>  <br>1 3<br>(1-2pin closed) | Input 33MHz to SH7751 PCICLK.(at shipment) |
| | J1<br>  <br>1 3<br>(2-3pin closed) | Power supply is not provided to SH7751 PCICLK. |

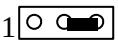
### 2. Test jumper2 (J2)

This jumper is the jumper for testing NMI pin. Table3.4 shows the function of the jumper for testing NMI pin. Use this jumper while pins 1-2 are closed.

When pins 2-3 are closed, the microcomputer does not work.

Close pins1-2 and connect NMI clip to TP3 while ROM emulator is used.

Table3.4 Function of the jumper for testing NMI pin (J2)

| Jumper Pin name | Jumper Pin Connected State | Function |
|---|---|---|
| J2 | J2<br>  <br>1 3<br>(1-2pin closed) | Abort switch can control NMI pin of the SH7751.(at shipment)<br>Close 1-2pin and connect NMI clip to TP3 when connecting ROM emulator by using ROM socket |
| | J2<br>  <br>1 3<br>(2-3pin closed) | Abort switch makes NMI pin of the SH7751 unconnected, NMI pin is connected to CPU bus interface.<br>Close 2-3pin when connecting ROM emulator by using CPU bus interface connector. |

### 3. Test Jumper3 (J3)

This jumper is the jumper for testing ATX power supply control. Table3.5 shows the function of the jumper for testing ATX power supply control. Use this jumper with 2-3pin closed.

When 1-2pin is closed, ATX power is in ON state at all times, and the power supply switch on the Solution Engine become invalid.

Table3.5 Function of the jumper for testing ATX power supply control

| Jumper Pin name | Jumper Pin Connected State | Function |
|---|---|---|
| J3 | J3<br>1 [▬ ○] 3<br>(1-2pin closed) | ATX power supply is in ON state, and power supply is provided to the Solution Engine all the time. |
| | J3<br>1 [○ ▬] 3<br>(2-3pin closed) | Power supply switch(SW1) on the Solution Engine controls ON/OFF of ATX power supply.(at shipment) |

### 4. Expansion slot 5V power supply(J4)

Table3.6 shows the function of the expansion slot 5V power supply jumper (J4). As shown table3.6, this jumper connects pins A66 and A67 of the expansion slot to 5V power supply on the Solution Engine.

When mounting LSI or IC that need analog 5V power supply, analog 5V power can be provided from pins A66 and A67 of the expansion slot by closing J1.

Signals of address bus and data bus output to the expansion slot are 3.3V.

Mount 3.3V→5V interface IC on a daughterboard if 5V interface is necessary.

Table 3.6 Function of the expansion slot 5V power supply jumper (J4)

| Jumper Pin name | Jumper Pin Connected State | Function |
|---|---|---|
| J4 | J4<br>1 [▬] 2<br>(Closed) | Pins A66 and A67 of the expansion slot are connected to A+5V of the Solution Engine.<br>In this state, A+5V power is provided to the daughterboard. |
| | J4<br>1 [○ ○] 2<br>(Open) | Power supply switch on the Solution Engine controls ON/OFF of ATX power. |

## 3.3. Description of test pin (TPn)

Table 3.7 lists the function of test pins.

Table 3.7 Test Pin Functions

| Test Pin | Function |
|---|---|
| TP1 | For test (Flash Ready/Busy) |
| TP2 | RESET-IN<br>(Connect RESET probe when using IC socket-equipped ROM emulation ) |
| TP3 | NMI-IN<br>(Connect NMI probe when using IC socket-equipped ROM emulator) |
| TP4 | For test (Super I/O) |
| TP5 | For test (Super I/O) |
| TP6 | For test (Super I/O) |
| TP7 | For test (Super I/O) |
| TP8 | For test (Super I/O) |
| TP9 | For test (Super I/O) |
| TP10 | For test (Super I/O) |
| TP11 | For test (Super I/O) |
| TP12 | For test (Super I/O) |
| TP13 | For test (Super I/O) |
| TP14 | For test (Super I/O) |
| TP15 | For test (Super I/O) |

# 4. LED Functions

**1. Power LED (LED17)**

This LED indicates that the power is supplied correctly. Function is as follows.

LED ON: Power is supplied to the Solution Engine.

LED OFF: Power is not supplied to the Solution Engine.

**2. CPU Status LED (LED9-LED12)**

This LED indicates CPU operation. Function is as follows.

LED9 ON: CPU is in RESET mode.

LED10 ON: CPU is in SLEEP mode.

LED11 ON: CPU is in STANBY mode.

LED12 ON: CPU is under operating condition

**3. PC card detection LED (LED13)**

This LED indicates that the PCMCIA controller detects the PC card normally.

LED ON: PC card is detected.

LED OFF: PC card is not detected.

**4. Ethernet line monitor LED (CN7-LED1 CN7-LED2, LED14-LED15)**

This LED indicates presence or absence of transmit signal and receive signal and connected condition of Ethernet line. For mode details on LED, refer to Section7, "Function block".

**5. HDD access LED (LED16)**

This LED indicates access condition to HDD. Function is as follows.

LED ON: Accessing to HDD

LED OFF: Not accessing to HDD.

**6. 8-bit LED (LED1-LED8)**

This LED is 8-bit LED that can turn ON and OFF LED via register allocated to memory map of the Solution Engine. For mode details on usage, refer to Section7, "Function block".

# 5. Memory map

Figure5.1 shows the memory map of the Solution Engine.

| Area No. | Space name | Bus width | Space | Device | Remarks |
|---|---|---|---|---|---|
| Area 0 | ROM Area<br><br>(Area for EPROM and Flash memory) | 32Bit<br>(5WAIT) | 16MB<br>Real capacity (4MB)<br>h'00000000-<br>h'003FFFFF | EPROM<br>MX27C8100PC-10<br>(MACRONIX)×2<br>or equivalent<br>42pin socket ×2 | (1) ROM emulator can be connected.<br>(2) SW5-3 and SW5-4 can change the place of EPROM and Flash ROM.<br>(3) 8Mbit EPROM can be used.<br>MX27C8100PC-10<br>(MACRONIX) ×2 |
| | | | 16MB<br>Real capacity (4MB)<br>h'01000000-<br>h'013FFFFF | FlashROM<br>MBM29LV160T-90PFTN<br>(FUJITSU) ×2 | |
| | | | 32MB<br>h'02000000-<br>h'03FFFFFF | | Expansion area0<br>Expansion slot/CS0 assert |
| Area1 | Expansion Area1<br><br>(On expansion connector) | Option | 64MB<br><br>h'04000000-<br>h'07FFFFFF | Expansion area1 | Expansion slot/CS1 assert |
| Area2 | Expansion Area2<br><br>(On expansion connector) | Option | 64MB<br><br>h'08000000-<br>h'0BFFFFFF | Expansion area2 | Expansion slot/CS2 assert |
| Area3 | SDRAM Area | 32Bit | 64MB<br>Real capacity (64MB)<br>h'0C000000-<br>h'0FFFFFFF | Device<br>Model name:<br>UPD45128841G5-A75-9JF<br>(ELPIDA) ×4<br><br>(128M SDRAM) | |
| Area4 | Expansion Area4 | Option | 64MB<br>h'10000000-<br>h'13FFFFFF | Expansion Area4 | Expansion Area4<br><br>Expansion slot/CS4 assert |

Figure5.1 Memory map

| Area No. | Space name | Bus width | Space | Device | Remarks |
|---|---|---|---|---|---|
| Area5 | Expansion Area5 (on expansion connector) | Option | 64MB h'14000000-h'17FFFFFF | Expansion Area5 | Expansion Area5<br><br>Expansion slot/CS5 assert |
| Area6 | 16Bit Peripheral device control register | 16Bit (3WAIT) | 16MB h'18000000 -h'18FFFFFF | Card controller LSI area manufactured by MARUBUN<br>Model name:<br>MR-SHPC-01 V2 | Memory and register must access to this area. |
| | | | 16MB h'19000000 -h'19FFFFFF | General-purpose switch area | This area reads general-purpose register ×2 |
| | | | 16MB h'1A000000 -h'1AFFFFFF | Area for debug LED | Area for debug LED<br>Single LED ×8 |
| | | | 16MB h'1B000000 -h'1BFFFFFF | Test mode area | Area for testing the Solution Engine.<br>This address is not open to users. Do not access to this area. |
| Area7 | SH7751R incorporated | - | h'1C000000 -h'1CFFFFFF | | |
| | | | H'1D000000 -h'1DFFFFFF | PCI memory space Access area | |
| | | | H'1E000000 -h'1E1F0000 | | |
| | | | h'1E200000 -h'1E2000FF | PCIC register<br>PCI configuration register area | |
| | | | h'1E20010 -h'1E200227 | PCIC register<br>PCIC local register area | |
| | | | H'1E200228 -h'1E23FFFF | | |
| | | | h'1E240000 -h'1E27FFFF | PCI I/O space Access area | |
| | | | H'E280000 -h'1FFFFFFF | | |

Figure5.1 Memory map

# 6. Hardware Configuration

Figure6.1 shows the block diagram of Solution Engine.

As figure6.1 shows, there are 3.3V bus, 5V bus and PCI bus.

**1. 3.3V Bus**

Memory including SDRAM and Flash ROM are connected to 3.3V Bus to execute user program at high-speed.

**2. 5V Bus**

EPROM is 5V bus interface. EPROM is connected to SH via 3V→5V conversion buffer.

**3. PCI Bus**

This PCI Bus used SH7751R PCIC. M1543C B1, Am79C973AVC and 2slot of PCI bus slot are connected to PCI bus.

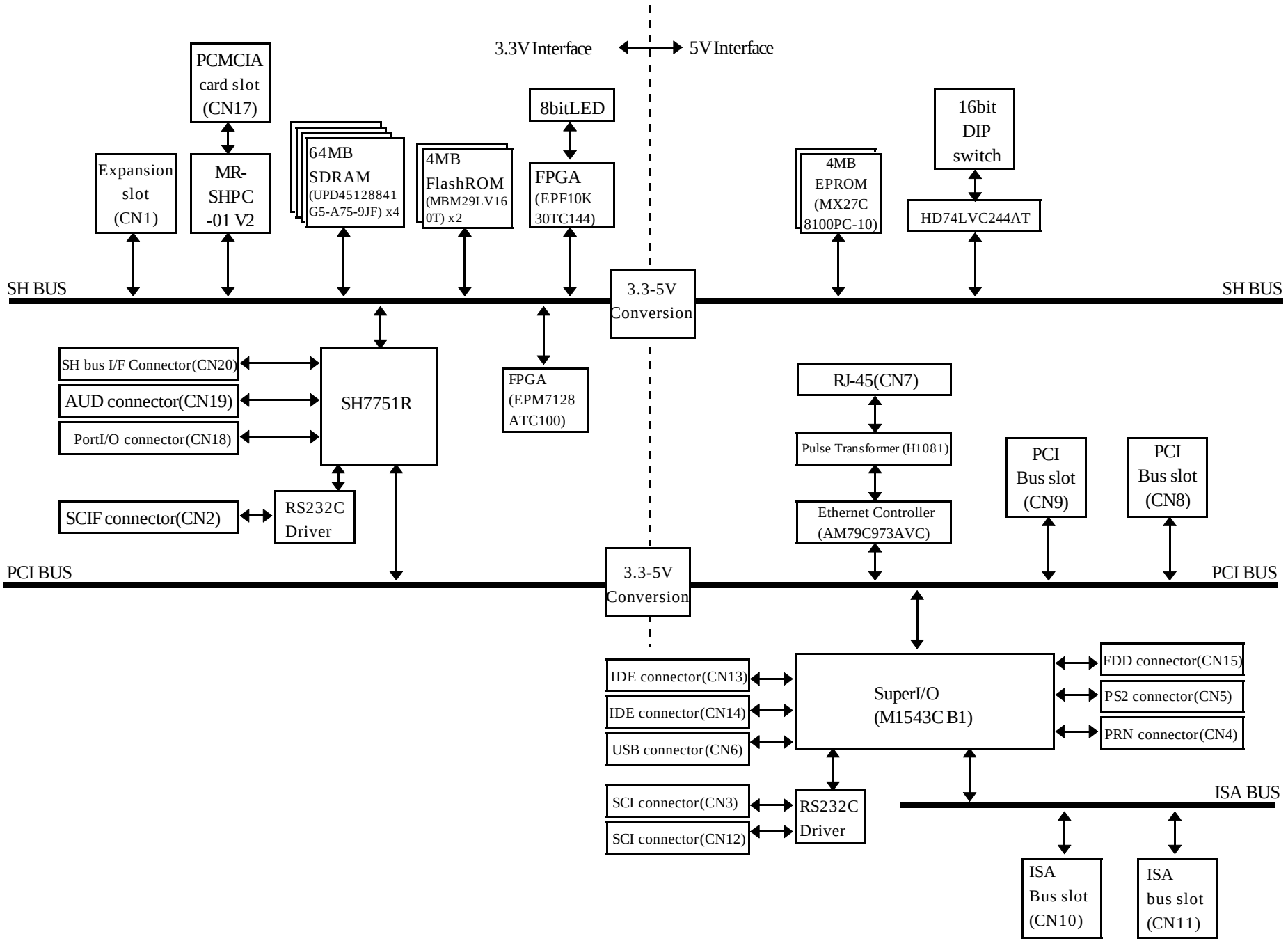



Figure6.1 Block diagramofthe Solution Engine

# 7. Function Block

## 7.1. Ethernet Control

### 1. Block description

Figure7.1 shows a block diagram of the Ethernet control block. The Ethernet control block has a controller (Am79C973A manufactured by AMD), serial EPROM (NM93C46) and a pulse transformer (H1081 manufactured by Pulse Engineering), and provides Ethernet-interface at 10BASE-T/100BASE-TX via RJ-45 connector CN7.

Other features include LEDs (CN7-LED1, CN7-LED2, LED14-LED15) used to indicate the presence of reception signals. In addition, a 25MHz crystal oscillator (X3) is mounted as the operation clock of Am70C973A.

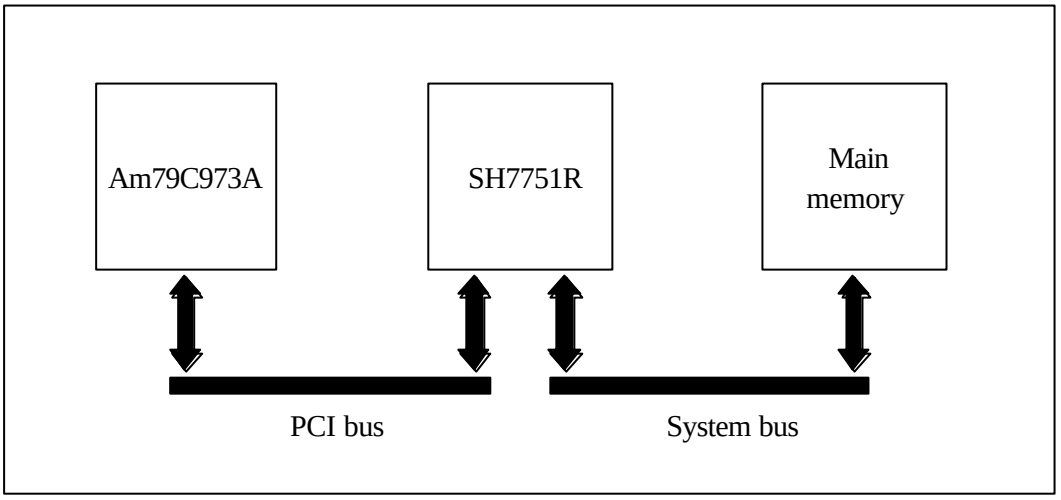



Figure 7.1 Ethernet Control Block Diagram

## 2. Memory Map

Figure 7.2 shows a memory map of the Ethernet controller.

Am79C973AVC uses 32byte(h'00-h'1F) on PCI bus.

Address of the memory map is offset address. Address on the PCI bus is decided by adding the set PCI base address.

Am79C973AVC can be assigned to both memory and I/O, because Am79C973AVC is connected to device number0 (IDSEL=AD16).

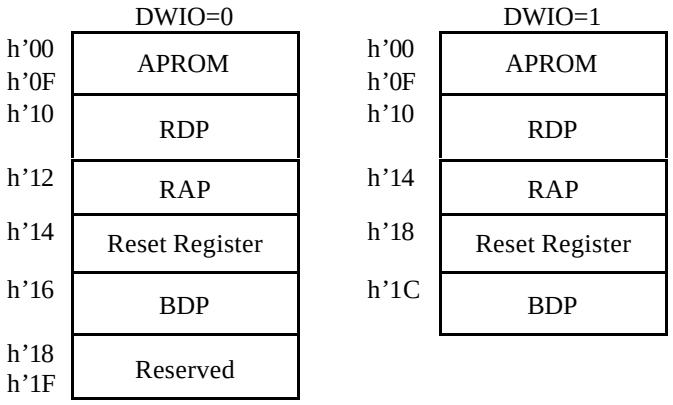



Figure 7.2. Ethernet memory map

a. Am79C973AVC PCI Configuration register

Table 7.1 shows a configuration of the PCI configuration register. The PCI configuration register is assigned to allocate Am79C973AVC on the PCI bus.

Table 7.1. Configuration of DP83902A Register

| 31 24 | 23 16 | 15 8 | 7 0 | Offset |
|---|---|---|---|---|
| Device ID | | Vendor ID | | h'00 |
| Status | | Command | | h'04 |
| Base-Class | Sub-Class | Programming IF | Revision ID | h'08 |
| Reserved | Header Type | Latency Timer | Reserved | h'0C |
| I/O Base Address | | | | h'10 |
| Memory Mapped I/O Base Address | | | | h'14 |
| Reserved | | | | h'18 |
| Reserved | | | | h'1C |
| Reserved | | | | h'20 |
| Reserved | | | | h'24 |
| Reserved | | | | h'28 |
| Subsystem ID | | Subsystem Vendor ID | | h'2C |
| Expansion ROM Base Address | | | | h'30 |
| Reserved | | | CAP-PTR | h'34 |
| Reserved | | | | h'38 |
| MAX_LAT | MIN_GNT | Interrupt Pin | Interrupt Line | h'3C |
| PMC | | NXT_ITM_PTR | CAP_ID | h'40 |
| DATA_REG | PMCSR_BSE | PMCSR | | h'44 |
| Reserved | | | | -<br>- |
| Reserved | | | | h'FC |

b. Control and Status register

Table7.2 shows a configuration of Control and Status register. CSR sets address of CSR to RAP and accesses from RDP.

Table7.2 Configuration of Control and Status register

| RAP Addr | Symbol | Default Value | Comments | Use |
|---|---|---|---|---|
| 00 | CSR0 | uuuu 0004 | Am79C973/Am79C975 Controller Status Register | R |
| 01 | CSR1 | uuuu uuuu | Lower IADR: maps to location 16 | S |
| 02 | CSR2 | uuuu uuuu | Upper IADR: maps to location 17 | S |
| 03 | CSR3 | uuuu 0000 | Interrupt Masks and Deferral Control | S |
| 04 | CSR4 | uuuu 0115 | Test and Features Control | R |
| 05 | CSR5 | uuuu 0000 | Extended Control and Interrupt 1 | R |
| 06 | CSR6 | uuuu uuuu | RXTX: RX/TX Encoded Ring Lengths | S |
| 07 | CSR7 | 0uuu 0000 | Extended Control and Interrupt 1 | R |
| 08 | CSR8 | uuuu uuuu | LADRF0: Logical Address Filter — LADRF[15:0] | S |
| 09 | CSR9 | uuuu uuuu | LADRF1: Logical Address Filter — LADRF[31:16] | S |
| 10 | CSR10 | uuuu uuuu | LADRF2: Logical Address Filter — LADRF[47:32] | S |
| 11 | CSR11 | uuuu uuuu | LADRF3: Logical Address Filter — LADRF[63:48] | S |
| 12 | CSR12 | uuuu uuuu | PADR0: Physical Address Register — PADR[15:0] | S |
| 13 | CSR13 | uuuu uuuu | PADR1: Physical Address Register — PADR[31:16] | S |
| 14 | CSR14 | uuuu uuuu | PADR2: Physical Address Register — PADR[47:32] | S |
| 15 | CSR15 | see register description | MODE: Mode Register | S |
| 16 | CSR16 | uuuu uuuu | IADRL: Base Address of INIT Block Lower (Copy) | T |
| 17 | CSR17 | uuuu uuuu | IADRH: Base Address of INIT Block Upper (Copy) | T |
| 18 | CSR18 | uuuu uuuu | CRBAL: Current RCV Buffer Address Lower | T |
| 19 | CSR19 | uuuu uuuu | CRBAU: Current RCV Buffer Address Upper | T |
| 20 | CSR20 | uuuu uuuu | CXBAL: Current XMT Buffer Address Lower | T |
| 21 | CSR21 | uuuu uuuu | CXBAU: Current XMT Buffer Address Upper | T |
| 22 | CSR22 | uuuu uuuu | NRBAL: Next RCV Buffer Address Lower | T |
| 23 | CSR23 | uuuu uuuu | NRBAU: Next RCV Buffer Address Upper | T |
| 24 | CSR24 | uuuu uuuu | BADRL: Base Address of RCV Ring Lower | S |
| 25 | CSR25 | uuuu uuuu | BADRU: Base Address of RCV Ring Upper | S |
| 26 | CSR26 | uuuu uuuu | NRDAL: Next RCV Descriptor Address Lower | T |
| 27 | CSR27 | uuuu uuuu | NRDAU: Next RCV Descriptor Address Upper | T |
| 28 | CSR28 | uuuu uuuu | CRDAL: Current RCV Descriptor Address Lower | T |
| 29 | CSR29 | uuuu uuuu | CRDAU: Current RCV Descriptor Address Upper | T |
| 30 | CSR30 | uuuu uuuu | BADXL: Base Address of XMT Ring Lower | S |
| 31 | CSR31 | uuuu uuuu | BADXU: Base Address of XMT Ring Upper | S |
| 32 | CSR32 | uuuu uuuu | NXDAL: Next XMT Descriptor Address Lower | T |
| 33 | CSR33 | uuuu uuuu | NXDAU: Next XMT Descriptor Address Upper | T |

**Note:**

u = undefined value, R = Running register, S = Setup register, T = Test register;

all default values are in hexadecimal format.

Table7.2 Configuration of Control and Status register

| RAP Addr | Symbol | Default Value | Comments | Use |
|---|---|---|---|---|
| 34 | CSR34 | uuuu uuuu | CXDAL: Current XMT Descriptor Address Lower | T |
| 35 | CSR35 | uuuu uuuu | CXDAU: Current XMT Descriptor Address Upper | T |
| 36 | CSR36 | uuuu uuuu | NNRDAL: Next Next Receive Descriptor Address Lower | T |
| 37 | CSR37 | uuuu uuuu | NNRDAU: Next Next Receive Descriptor Address Upper | T |
| 38 | CSR38 | uuuu uuuu | NNXDAL: Next Next Transmit Descriptor Address Lower | T |
| 39 | CSR39 | uuuu uuuu | NNXDAU: Next Next Transmit Descriptor Address Upper | T |
| 40 | CSR40 | uuuu uuuu | CRBC: Current Receive Byte Count | T |
| 41 | CSR41 | uuuu uuuu | CRST: Current Receive Status | T |
| 42 | CSR42 | uuuu uuuu | CXBC: Current Transmit Byte | T |
| 43 | CSR43 | uuuu uuuu | CXST: Current Transmit Status | T |
| 44 | CSR44 | uuuu uuuu | NRBC: Next RCV Byte Count | T |
| 45 | CSR45 | uuuu uuuu | NRST: Next RCV Status | T |
| 46 | CSR46 | uuuu uuuu | POLL: Poll Time Counter | T |
| 47 | CSR47 | uuuu uuuu | PI: Polling Interval | S |
| 48 | CSR48 | uuuu uuuu | Reserved | |
| 49 | CSR49 | uuuu uuuu | Reserved | |
| 50 | CSR50 | uuuu uuuu | Reserved | |
| 51 | CSR51 | uuuu uuuu | Reserved | |
| 52 | CSR52 | uuuu uuuu | Reserved | |
| 53 | CSR53 | uuuu uuuu | Reserved | |
| 54 | CSR54 | uuuu uuuu | Reserved | |
| 55 | CSR55 | uuuu uuuu | Reserved | |
| 56 | CSR56 | uuuu uuuu | Reserved | |
| 57 | CSR57 | uuuu uuuu | Reserved | |
| 58 | CSR58 | see register description | SWS: Software Style | S |
| 59 | CSR59 | uuuu uuuu | Reserved | T |
| 60 | CSR60 | uuuu uuuu | PXDAL: Previous XMT Descriptor Address Lower | T |
| 61 | CSR61 | uuuu uuuu | PXDAU: Previous XMT Descriptor Address Upper | T |
| 62 | CSR62 | uuuu uuuu | PXBC: Previous Transmit Byte Count | T |
| 63 | CSR63 | uuuu uuuu | PXST: Previous Transmit Status | T |
| 64 | CSR64 | uuuu uuuu | NXBAL: Next XMT Buffer Address Lower | T |
| 65 | CSR65 | uuuu uuuu | NXBAU: Next XMT Buffer Address Upper | T |
| 66 | CSR66 | uuuu uuuu | NXBC: Next Transmit Byte Count | T |
| 67 | CSR67 | uuuu uuuu | NXST: Next Transmit Status | T |
| 68 | CSR68 | uuuu uuuu | Reserved | |
| 69 | CSR69 | uuuu uuuu | Reserved | |
| 70 | CSR70 | uuuu uuuu | Reserved | |

Table7.2 Configuration of Control and Status register

| RAP Addr | Symbol | Default Value | Comments | Use |
|---|---|---|---|---|
| 71 | CSR71 | uuuu uuuu | Reserved | |
| 72 | CSR72 | uuuu uuuu | RCVRC: RCV Ring Counter | T |
| 73 | CSR73 | uuuu uuuu | Reserved | |
| 74 | CSR74 | uuuu uuuu | XMTRC: XMT Ring Counter | T |
| 75 | CSR75 | uuuu uuuu | Reserved | |
| 76 | CSR76 | uuuu uuuu | RCVRL: RCV Ring Length | S |
| 77 | CSR77 | uuuu uuuu | Reserved | |
| 78 | CSR78 | uuuu uuuu | XMTRL: XMT Ring Length | S |
| 79 | CSR79 | uuuu uuuu | Reserved | |
| 80 | CSR80 | uuuu 1410 | DMATCFW: DMA Transfer Counter and FIFO Threshold | S |
| 81 | CSR81 | uuuu uuuu | Reserved | |
| 82 | CSR82 | uuuu uuuu | Transmit Descriptor Pointer Address Lower | S |
| 83 | CSR83 | uuuu uuuu | Reserved | |
| 84 | CSR84 | uuuu uuuu | DMABA: Address Register Lower | T |
| 85 | CSR85 | uuuu uuuu | DMABA: Address Register Upper | T |
| 86 | CSR86 | uuuu uuuu | DMABC: Buffer Byte Counter | T |
| 87 | CSR87 | uuuu uuuu | Reserved | |
| 88 | CSR88 | 262 5003 (Am79C973) 262 7003 (Am79C975) | Chip ID Register Lower | T |
| 89 | CSR89 | uuuu 262 | Chip ID Register Upper T | |
| 90 | CSR90 | uuuu uuuu | Reserved | |
| 91 | CSR91 | uuuu uuuu | Reserved | |
| 92 | CSR92 | uuuu uuuu | RCON: Ring Length Conversion | T |
| 93 | CSR93 | uuuu uuuu | Reserved | |
| 94 | CSR94 | uuuu uuuu | Reserved | |
| 95 | CSR95 | uuuu uuuu | Reserved | |
| 96 | CSR96 | uuuu uuuu | Reserved | |
| 97 | CSR97 | uuuu uuuu | Reserved | |
| 98 | CSR98 | uuuu uuuu | Reserved | |
| 99 | CSR99 | uuuu uuuu | Reserved | |
| 100 | CSR100 | uuuu 0200 | Bus Timeout | S |
| 101 | CSR101 | uuuu uuuu | Reserved | |
| 102 | CSR102 | uuuu uuuu | Reserved | |
| 103 | CSR103 | uuuu 0105 | Reserved | |
| 104 | CSR104 | uuuu uuuu | Reserved | |
| 105 | CSR105 | uuuu uuuu | Reserved | |
| 106 | CSR106 | uuuu uuuu | Reserved | |
| 107 | CSR107 | uuuu uuuu | Reserved | |

Table7.2 Configuration of Control and Status register

| RAP Addr | Symbol | Default Value | Comments | Use |
|---|---|---|---|---|
| 108 | CSR108 | uuuu uuuu | Reserved | |
| 109 | CSR109 | uuuu uuuu | Reserved | |
| 110 | CSR110 | uuuu uuuu | Reserved | |
| 111 | CSR111 | uuuu uuuu | Reserved | |
| 112 | CSR112 | uuuu uuuu | Missed Frame Count | R |
| 113 | CSR113 | uuuu uuuu | Reserved | |
| 114 | CSR114 | uuuu uuuu | Received Collision Count | R |
| 115 | CSR115 | uuuu uuuu | Reserved | |
| 116 | CSR116 | 0000 0000 | On Now Miscellaneous S | |
| 117 | CSR117 | uuuu uuuu | Reserved | |
| 118 | CSR118 | uuuu uuuu | Reserved | |
| 119 | CSR119 | uuuu 0105 | Reserved | |
| 120 | CSR120 | uuuu uuuu | Reserved | |
| 121 | CSR121 | uuuu uuuu | Reserved | |
| 122 | CSR122 | uuuu 0000 | Receive Frame Alignment Control | S |
| 123 | CSR123 | uuuu uuuu | Reserved | |
| 124 | CSR124 | uuuu 0000 | Test Register 1 | T |
| 125 | CSR125 | 003c 0060 | MAC Enhanced Configuration Control | T |
| 126 | CSR126 | uuuu uuuu | Reserved | |
| 127 | CSR127 | uuuu uuuu | Reserved | |

c. Bus Configuration register

Table7.3 shows a configuration of the bus configuration register. BCR sets address of BCR to RAP and accesses from BDP.

Table7.3 Configuration of the bus configuration register

| RAP | Mnemonic | Default | Name | Programmability | |
|---|---|---|---|---|---|
| | | | | User | EEPROM |
| 0 | MSRDA | 0005h | Reserved | No | No |
| 1 | MSWRA | 0005h | Reserved | No | No |
| 2 | MC | 0002h | Miscellaneous Configuration | Yes | Yes |
| 3 | Reserved | N/A | Reserved | No | No |
| 4 | LED0 | 00C0h | LED0 Status | Yes | Yes |
| 5 | LED1 | 0084h | LED1 Status | Yes | Yes |
| 6 | LED2 | 0088h | LED2 Status | Yes | Yes |
| 7 | LED3 | 0090h | LED3 Status | Yes | Yes |
| 8 | Reserved | N/A | Reserved | No | No |
| 9 | FDC | 0000h | Full-Duplex Control | Yes | Yes |
| 10-15 | Reserved | N/A | Reserved | No | No |
| 16 | IOBASEL | N/A | Reserved | No | No |
| 17 | IOBASEU | N/A | Reserved | No | No |
| 18 | BSBC | 9001h | Burst and Bus Control | Yes | Yes |
| 19 | EECAS | 0002h | EEPROM Control and Status | Yes | No |
| 20 | SWS | 0000h | Software Style | Yes | No |
| 21 | INTCON | N/A | Reserved | No | No |
| 22 | PCILAT | FF06h | PCI Latency | Yes | Yes |
| 23 | PCISID | 0000h | PCI Subsystem ID | No | Yes |
| 24 | PCISVID | 0000h | PCI Subsystem Vendor ID | No | Yes |
| 25 | SRAMSIZ | 0000h | SRAM Size | Yes | Yes |
| 26 | SRAMB | 0000h | SRAM Boundary | Yes | Yes |
| 27 | SRAMIC | 0000h | SRAM Interface Control | Yes | Yes |
| 28 | EBADDRL | N/A | Expansion Bus Address Lower | Yes | No |
| 29 | EBADDRU | N/A | Expansion Bus Address Upper | Yes | No |
| 30 | EBD | N/A | Expansion Bus Data Port | Yes | No |
| 31 | STVAL | FFFFh | Software Timer Value | Yes | No |
| 32 | MIICAS | 0000h | PHY Control and Status | Yes | Yes |
| 33 | MIIADDR | 0000h | PHY Address | Yes | Yes |
| 34 | MIIMDR | N/A | PHY Management Data | Yes | No |
| 35 | PCIVID | 1022h | PCI Vendor ID | No | Yes |
| 36 | PMC_A | C811h | PCI Power Management Capabilities (PMC) Alias Register | No | Yes |
| 37 | DATA0 | 0000h | PCI DATA Register Zero Alias Register | No | Yes |
| 38 | DATA1 | 0000h | PCI DATA Register One Alias Register | No | Yes |
| 39 | DATA2 | 0000h | PCI DATA Register Two Alias Register | No | Yes |
| 40 | DATA3 | 0000h | PCI DATA Register Three Alias Register | No | Yes |

Table7.3 Configuration of the bus configuration register

| RAP | Mnemonic | Default | Name | Programmability | |
|---|---|---|---|---|---|
| | | | | User | EEPROM |
| 41 | DATA4 | 0000h | PCI DATA Register Four Alias Register | No | Yes |
| 42 | DATA5 | 0000h | PCI DATA Register Five Alias Register | No | Yes |
| 43 | DATA6 | 0000h | PCI DATA Register Six Alias Register | No | Yes |
| 44 | DATA7 | 0000h | PCI DATA Register Seven Alias Register | No | Yes |
| 45 | PMR1 | N/A | Pattern Matching Register 1 | Yes | No |
| 46 | PMR2 | N/A | Pattern Matching Register 2 | Yes | No |
| 47 | PMR3 | N/A | Pattern Matching Register 3 | Yes | No |
| 48 | Reserved | 0000h | Reserved (for Am79C975) | Yes* | Yes* |
| 49 | Reserved | 0000h | Reserved (for Am79C975) | Yes* | Yes* |
| 50 | Reserved | 0000h | Reserved (for Am79C975) | Yes* | Yes* |
| 51 | Reserved | 0000h | Reserved (for Am79C975) | Yes* | Yes* |
| 52 | Reserved | 0000h | Reserved (for Am79C975) | Yes* | Yes* |
| 53 | Reserved | 0000h | Reserved (for Am79C975) | Yes* | Yes* |
| 54 | Reserved | 0000h | Reserved (for Am79C975) | Yes* | Yes* |

## 4. Ethernet Line Monitor LED(CN7-LED1, CN7-LED2, LED14 to LED15)

LEDs (CN7-LED1, CN7-LED2, LED14 to LED15) indicate the line condition of Ethernet. Function of each LED is as follows.

(CN7-LED1)

This LED indicates that the line is normally connected.

When lit, the line is normally connected.

(CN7-LED2)

This LED indicates the state of reception of the Solution Engine.

When lit, packet is being received.

(LED14)

This LED does not light in initial state.

(LED15)

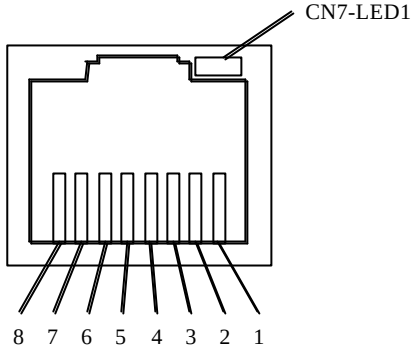
This LED indicates the state of transmission of the Solution Engine.

When lit, packet is being transmitted.

## 4. RJ-45 Connector (CN7) Pin Assignments

Figure 7.3 shows the pin assignments and functions of RJ-45 connector.

Type: RJHS-5381
Maker: Amphenol




| Pin No. | Pin Name | I/O | Function |
|---|---|---|---|
| 1 | TX+ | O | Transmit Data |
| 2 | TX- | O | Transmit Data |
| 3 | RX+ | I | Receive Data |
| 4 | (NC) | - | - |
| 5 | (NC) | - | - |
| 6 | RX- | I | Receive Data |
| 7 | (NC) | - | - |
| 8 | (NC) | - | - |

Figure 7.3 RJ-45 connector

## 7.2. Super I/O control

### 7.2.1. Block Diagram

Figure7.4 shows a block diagram of the Super I/O control block.

The Super I/O control block has a controller (M1543C B1 manufactured by ALi). The Super I/O control block provides various input device-interface.

The Super I/O controller has the following functions.

(1) PCI device

- ISA bus interface (PCI to ISA Bridge)
- IDE interface (IDE Master M5229)
- USB interface (USB M5237)
- Power management unit (PMU M7101)

(2) Super I/O

- Serial interface (UART1, UART3)
- Parallel interface (Parallel Port)
- RTC (Lithium battery can be connected)
- Keyboard interface (PS2)
- Mouse interface (PS2)
- FIR interface (UART2)
- FDD interface (FDC)

The Super I/O control block has a 14.3181MHz crystal oscillator (OSC1) and 48MHz crystal oscillator (OSC5) as the operation clock.

Transfer speed (baud rate) of serial interface is generated based on 1.8462MHz.

The Super I/O control block has a 32.768KHz crystal oscillator for RTC(×2).

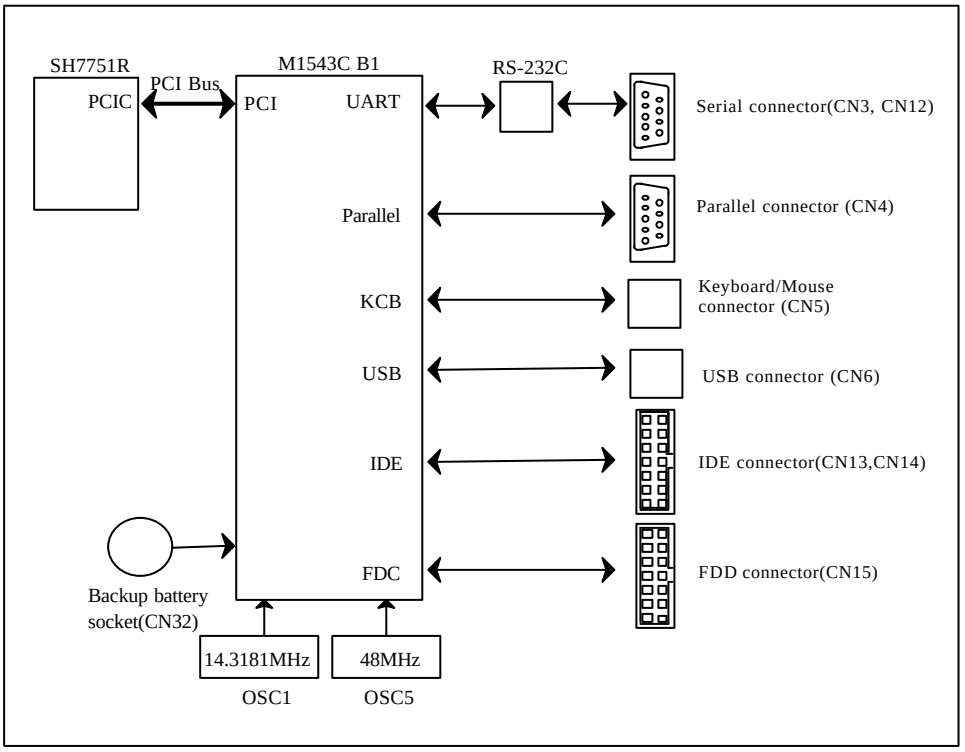



Figure7.4 Super I/O control block

### 7.2.2. Super I/O controller

To use various kinds of M1543C B1-embedded modules, it is necessary to set configuration data (base address and etc.) to M1543C B1.

Set M1543C B1 configuration data as follows.

This configuration data includes using/not using each of modules, interruption allocation and base address setting of modules of PCI device and Super I/O.

(1) Configuration of PCI device

Configuration of PCI device is performed by configuration cycle of PCI bus.

For data of configuration register of each device, refer to SuperI/O (M1543C B1) manual.

PCI device number of each device is as follows.

| Device name | Device No. | Remarks |
|---|---|---|
| ISA bus interface | H'2 | IDSEL=AD18 |
| IDE interface | H'B | IDSEL=AD27 |
| USB interface | H'F | IDSEL=AD31 |
| Power management unit | H'C | IDSEL=AD28 |

(2) Configuration of SuperI/O

Perform configuration of SuperI/O as follows.

Address described below is address of PCI I/O area.

1. Write 0x51 and 0x23 to CONFIG PORT(0x000003F0) twice.
   By this, FDC37C935A enters into configuration data setting mode.
2. Set INDEX to INDEX PORT(0x000003F0) and set configuration data from DATA PORT(0x000003F1).
3. After setting configuration data, go out of configuration data setting mode by writing 0xBB to CONFIG PORT.

Refer to manual of super I/O (M1543C B1) for details on configuration data.

### 7.2.3. Serial Controller

#### 1. Register Map

Table 7.4 lists the memory map of M1543C B1 super I/O serial controller.

Base address initial value of serial controller register is UART1: h'03F8, UART3: h'02F8.

Set the configuration data as shown in section 7.2.2, "SuperI/O Controller".

Table 7.4 M1543C B1 Super I/O serial controller register map

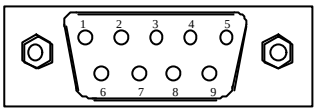
| Channel | Address | R/W | DLAB | Register name |
|---|---|---|---|---|
| UART1 *(CN3) | h'0(h'000003F8) | R | 0 | RBR(Receiver Buffer Register) |
| | h'0(h'000003F8) | W | 0 | THR(Transmitter Holding Register) |
| | h'0(h'000003F8) | W | 1 | DLL(Divisor Latch LSB) |
| | h'1(h'000003F9) | W | 1 | DLM(Divisor Latch MSB) |
| | h'1(h'000003F9) | R/W | 0 | IER(Interrupt Enable Register) |
| | h'2(h'000003FA) | R | X | IIR(Interrupt Identification Register) |
| | h'2(h'000003FA) | W | X | FCR(FIFO Control Register) |
| | h'3(h'000003FB) | R/W | X | LCR(Line Control Register) |
| | h'4(h'000003FC) | R/W | X | MCR(Modem Control Register) |
| | h'5(h'000003FD) | R/W | X | LSR(Line Status Register) |
| | h'6(h'000003FE) | R/W | X | MSR(Modem Status Register) |
| | h'7(h'000003FF) | R/W | X | SCR(Scratch Register) |
| UART3 *(CN12) | h'0(h'000002F8) | R | 0 | RBR(Receiver Buffer Register) |
| | h'0(h'000002F8) | W | 0 | THR(Transmitter Holding Register) |
| | h'0(h'000002F8) | W | 1 | DLL(Divisor Latch LSB) |
| | h'1(h'000002F9) | W | 1 | DLM(Divisor Latch MSB) |
| | h'1(h'000002F9) | R/W | 0 | IER(Interrupt Enable Register) |
| | h'2(h'000002FA) | R | X | IIR(Interrupt Identification Register) |
| | h'2(h'000002FA) | W | X | FCR(FIFO Control Register) |
| | h'3(h'000002FB) | R/W | X | LCR(Line Control Register) |
| | h'4(h'000002FC) | R/W | X | MCR(Modem Control Register) |
| | h'5(h'000002FD) | R/W | X | LSR(Line Status Register) |
| | h'6(h'000002FE) | R/W | X | MSR(Modem Status Register) |
| | h'7(h'000002FF) | R/W | X | SCR(Scratch Register) |

*DLAB is bit7 of "LCR". Don't care X.

* ( ): Serial interface connector section

**2. 9-pin D-sub connector (CN3) pin assignment**

Connector Model Name: DM11351-Z3

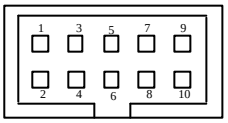



| Pin No. | Pin Name | I/O | Function |
|---|---|---|---|
| 1 | CD | I | Carrier Detect |
| 2 | RxD | I | Receive Data |
| 3 | TxD | O | Transmit Data |
| 4 | DTR | O | Data Terminal Ready |
| 5 | GND | - | Ground |
| 6 | DSR | I | Data Set Ready |
| 7 | RTS | O | Request To Send |
| 8 | CTS | I | Clear To Send |
| 9 | RI | I | Ring Indicator |

Figure7.5 9-pin D-SUB connector(CN3) pin assignment

**2. 10-pin connector (CN12) pin assignment**

Connector Model Name: HIF3C-10PA-2.54DSA




| Pin No. | Pin Name | I/O | Function |
|---|---|---|---|
| 1 | CD | I | Carrier Detect |
| 2 | RxD | I | Receive Data |
| 3 | TxD | O | Transmit Data |
| 4 | DTR | O | Data Terminal Ready |
| 5 | GND | - | Ground |
| 6 | DSR | I | Data Set Ready |
| 7 | RTS | O | Request To Send |
| 8 | CTS | I | Clear To Send |
| 9 | RI | I | Ring Indicator |
| 10 | NC | - | No Connect |

Figure7.6 10-pin connector (CN12) pin assignment

### 7.2.4. Parallel controller

### 1. Register map

Table7.5 (1) and (2) list a memory map of IEEE 1284 parallel controller register of M1543C B1 super I/O controller.

Base address initial value of parallel controller register is h'378. Set the configuration data as shown in section 7.2.2, "SuperI/O controller" to change base address.

Table 7.5 (1) Parallel port register

| Address | Compatible Parallel Port | | Enhanced Parallel Port(EPP) | |
|---|---|---|---|---|
| | Register Name | Description | Register Name | Description |
| h'0(h'00000378) | DTP | Data Port | DTP | Data Port |
| h'1(h'00000379) | STP | Status Port | STP | Status Port |
| h'2(h'0000037A) | CTP | Control Port | CTP | Control Port |
| h'3(h'0000037B) | | | ADDR | EPP Address Port |
| h'4(h'0000037C) | | | DATA0 | EPP Data Port0 |
| h'5(h'0000037D) | | | DATA1 | EPP Data Port1 |
| h'6(h'0000037E) | | | DATA2 | EPP Data Port2 |
| h'7(h'0000037F) | | | DATA3 | EPP Data Port3 |

Table7.5(2) Parallel port register

| Address | Extended Capacities Parallel Port(EPC) | | |
|---|---|---|---|
| | Register Name | Description | EPC MODES |
| h'000(h'00000378) | data<br>ecpAFifo | Data Register<br>ECP FIFO(Address) | 000-001<br>011 |
| h'001(h'00000379) | dsr | Status Register | All |
| h'002(h'0000037A) | dcr | Control Register | All |
| h'400(h'00000778) | cFifo<br>ecpFio<br>tFifo<br>cofigA | Parallel Port Data FIFO<br>ECP FIFO(Data)<br>Test FIFO<br>Configuration RegisterA | 010<br>011<br>110<br>111 |
| h'401(h'00000779) | cnfigB | Configurationj RegisterB | 111 |
| h'402(h'0000077A) | ecr | Extended Control Register | All |

## 2. 25 pin D-sub parallel connector (CN4) pin assignment

Table7.6 lists pin assignments and functions of 25-pin D-sub connector (CN4).

Table7.6 25-pin D-sub parallel connector (CN4) pin assignment

| Pin No | Signal name | I/O | Remarks |
|---|---|---|---|
| 1 | /STROBE | I/O | /STB, /WRITE |
| 2 | D0 | I/O | PD0 |
| 3 | D1 | I/O | PD1 |
| 4 | D2 | I/O | PD2 |
| 5 | D3 | I/O | PD3 |
| 6 | D4 | I/O | PD4 |
| 7 | D5 | I/O | PD5 |
| 8 | D6 | I/O | PD6 |
| 9 | D7 | I/O | PD7 |
| 10 | /ACK | I | /ACK |
| 11 | BUSY | I | Busy |
| 12 | PF | I | PE |
| 13 | SLCT | I | SLCT |
| 14 | /AUTOFD | I/O | /AFD, /DSTRB |
| 15 | /ERR | I | /ERR |
| 16 | /INIT | I/O | /INIT |
| 17 | /SLCTin | I/O | /SLIN, /ASTRB |
| 18 | GND | | |
| 19 | GND | | |
| 20 | GND | | |
| 21 | GND | | |
| 22 | GND | | |
| 23 | GND | | |
| 24 | GND | | |
| 25 | GND | | |

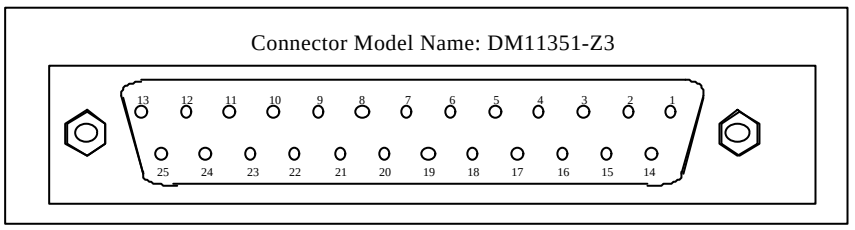



Figure 7.7 25 pin D-sub parallel connector(CN4)

### 7.2.5. Keyboard /Mouse controller

#### 1. keyboard(KBC) register map

Table7.7 lists a register map of a keyboard controller(KBC).

Base address initial value of the keyboard controller register is h'60.

Set the configuration data as section 7.2.2, "SuperI/O controller".

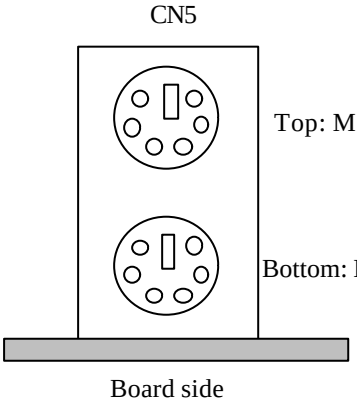
Table 7.7 keyboard controller (KBC) register map

| Address | Read | | Write | |
|---|---|---|---|---|
| | Register Name | Description | Register Name | Description |
| h'00000060 | DBBOUT | DBBOUT | DTP | F1 Clear(Data) |
| h'00000064 | STATUS | STATUS | STP | F1 Set(Command) |

#### 2. Keyboard/Mouse interface connector (CN5) pin assignment

Figure7.8 shows pin assignments and functions of a keyboard/mouse interface connector (CN5).

Connecter Model Name: MH11061-D2




Mouse connector pin assignment

| Pin No. | Signal | I/O | Remark |
|---|---|---|---|
| 1 | MDAT | I/O | |
| 2 | N.C | - | Reserved |
| 3 | GND | - | |
| 4 | +5V | - | |
| 5 | MCLK | I/O | |
| 6 | N.C | - | Reserved |

Keyboard connector pin assignment

| Pin No. | Signal | I/O | Remark |
|---|---|---|---|
| 1 | KBDAT | I/O | |
| 2 | N.C | - | Reserved |
| 3 | GND | - | |
| 4 | +5V | - | |
| 5 | KBCLK | I/O | |
| 6 | N.C | - | Reserved |

Figure7.8 Keyboard/mouse interface connector(CN5)

### 7.2.6. RTC controller

#### 1. Register map

Table7.8 lists the register map of RTC controller.

Base address initial value of RTC register address is h'70.

Base address initial value of RTC register address cannot be changed.

Table7.8 Register map of RTC controller

| Address | Register Name |
|---|---|
| h'00000070 | Address Register |
| h'00000071 | Data Register |

Table 7.8 Register map of RTC controller

| Address | Register Set | |
|---|---|---|
| | Register Name | Description |
| h'00 | SEC | Seconds |
| h'01 | SEC ALM | Seconds Alarm |
| h'02 | MIN | Minutes |
| h'03 | MIN ALM | Minutes Alarm |
| h'04 | HR | Hours |
| h'05 | HR ALM | Hours Alarm |
| h'06 | DOW | Day of Week |
| h'07 | DOM | Date of Month |
| h'08 | MON | Month |
| h'09 | YEAR | Year |
| h'0A | CRA | Control Register A |
| h'0B | CRB | Control Register B |
| h'0C | CRC | Control Register C |
| h'0D | CRD | Control Register D |
| h'0E - h'FF | - | General-purpose RAM |

### 7.2.7. Floppy disk controller

#### 1. Register map

Table7.10 lists the register map of a super I/O floppy disk controller(FDC).

Base address initial value of the SuperI/O floppy disk controller is h'3F0.

Set configuration data shown in section 7.2.2, "SuperI/O controller" to change base address.

Table 7.10 Register map of the floppy disk controller(FDC)

| Address | Read | | Write | |
|---|---|---|---|---|
| | Register Name | Description | Register Name | Description |
| h'000003F0 | SRA | Status RegisterA | - | |
| h'000003F1 | SRB | Status RegisterB | - | |
| h'000003F2 | DOR | Digital Output Register | DOR | Digital Output Register |
| h'000003F3 | TDR | Tape Driver Register | TDR | Tape Driver Register |
| h'000003F4 | MSR | Main Status Register | DSR | Data Rate Select Register |
| h'000003F5 | FIFO | Data Register | FIFO | Data Register |
| h'000003F6 | - | Reserved | - | Reserved |
| h'000003F7 | DIR | Digital Input Register | CCR | Configuration Control Register |

## 2. Pin assignment of a floppy disk interface connector(CN15)

Table 7.11 lists pin assignments of the floppy disk interface connector pin (CN15).

Table 7.11 Pin assignment of the floppy disk interface connector pin(CN15)

| Pin No. | Signal name | I/O | Remarks | Pin No. | Signal name | I/O | Remarks |
|---|---|---|---|---|---|---|---|
| 1 | GND | | | 18 | DIRECTION SELECT | O | /DIR |
| 2 | MODE SELECT | O | DENSEL | 19 | GND | | |
| 3 | GND | | | 20 | STEP | O | /STEP |
| 4 | OPEN | | | 21 | GND | | |
| 5 | GND | | | 22 | WRITE DATA | O | /WDATA |
| 6 | DRATE0 | O | DRATE0 | 23 | GND | | |
| 7 | GND | | | 24 | WRITE GATE | O | /WGATE |
| 8 | INDEX | I | /INDEX | 25 | GND | | |
| 9 | GND | | | 26 | TRACK 00 | I | /TRK0 |
| 10 | MOTOR ON 0 | O | /MTR0 | 27 | GND | | |
| 11 | GND | | | 28 | WRITE PROTECT | I | /WP |
| 12 | DRIVE SELECT1 | O | /DR1 | 29 | GND | | |
| 13 | GND | | | 30 | READ DATA | I | /RDATA |
| 14 | DRIVE SELECT0 | O | /DR0 | 31 | GND | | |
| 15 | GND | | | 32 | SIDE ONE SELECT | O | /HDSEL |
| 16 | MOTOR ON 1 | O | /MTR1 | 33 | GND | | |
| 17 | GND | | | 34 | DISK CHANGE | I | /DSKCHG |

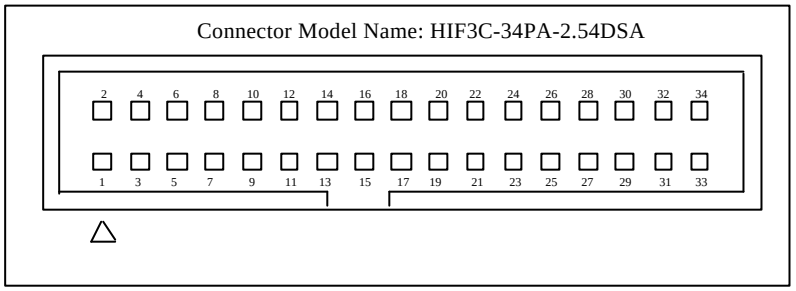



Figure7.9 floppy disk interface pin connector(CN15)

### 7.2.8. IDE controller

#### 1. Register map

Table7.12 lists the IDE controller register map of super I/O interface.

Base address initial value of IDE controller register address is h'170(secondary) and h'1F0(primary).

Set configuration data shown in section 7.2.2, "SuperI/O controller" to change base address.

Table 7.12 IDE controller register map

| Address | Bank1 Register Set | |
|---|---|---|
| | Register Name | Description |
| h'170 | - | Task File Register(Secondary) |
| h'1F0 | - | Task File Register(Primary) |
| h'374 | - | MISC. AT Register(Secondary) |
| h'3F4 | - | MISC. AT Register(Primary) |

#### 2. 40-pin connector (CN13, Cn14) pin assignment

Table7.13 shows pin assignments of 40-pin connector(CN13,CN14).

Table7.13 Pin assignment of 40-pin connector (CN13, CN14)

| Pin No. | Signal name | I/O | Pin No. | Signal name | I/O |
|---|---|---|---|---|---|
| 1 | RESET | O | 21 | DREQ | I |
| 2 | GND | | 22 | GND | |
| 3 | D7 | I/O | 23 | /DIOW | O |
| 4 | D8 | I/O | 24 | GND | |
| 5 | D6 | I/O | 25 | /DIOR | O |
| 6 | D9 | I/O | 26 | GND | |
| 7 | D5 | I/O | 27 | IORDY | I |
| 8 | D10 | I/O | 28 | CSEL | O |
| 9 | D4 | I/O | 29 | /DMAACK | O |
| 10 | D11 | I/O | 30 | GND | |
| 11 | D3 | I/O | 31 | IRQ | I |
| 12 | D12 | I/O | 32 | /IOCS16 | I |
| 13 | IRQ | I/O | 33 | A1 | O |
| 14 | D13 | I/O | 34 | /PDIAG | I |
| 15 | D1 | I/O | 35 | A0 | O |
| 16 | D14 | I/O | 36 | A2 | O |
| 17 | D0 | I/O | 37 | /CS0 | O |
| 18 | D15 | I/O | 38 | /CS1 | O |
| 19 | GND | I/O | 39 | /DASP | I/O |
| 20 | KEY | | 40 | GND | |

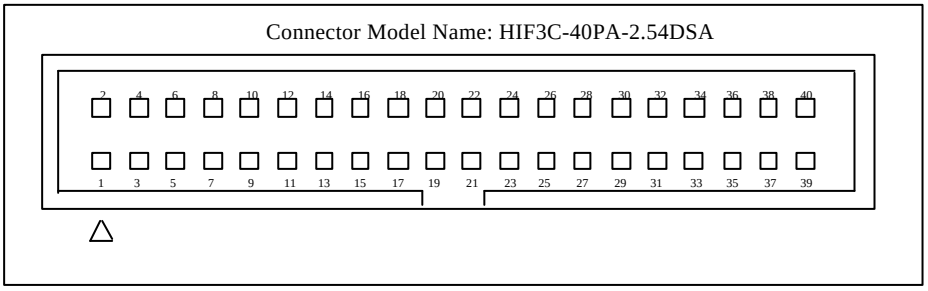



Figure7.10 40-pin connector(CN6)

### 7.2.9. USB control

#### 1. Register map

Table7.14(1) and (2) show the USB controller register map of SuperI/O interface.

Base address initial value of USB control register address is h'00000000.

Set configuration data shown in section 7.2.2, "SuperI/O controller" to change base address.

Table 7.14 (1) Register map of USB controller

| Address | Register Name | R/W | Default Value |
|---|---|---|---|
| h'00 | HcRevision | R | h'00000110 |
| h'04 | HcControl | R/W | h'00000000 |
| h'08 | HcCommandStatus | R/W | h'00000000 |
| h'0C | HcInterruptStatus | R/W | h'00000000 |
| h'10 | HcInterruptEnable | R/W | h'00000000 |
| h'14 | HcInterruptDisable | R/W | h'00000000 |
| h'18 | HcHCCA | R/W | h'00000000 |
| h'1C | HcPeriodCurrentED | R/W | h'00000000 |
| h'20 | HcControlHeadED | R/W | h'00000000 |
| h'24 | HcControlCurrentED | R/W | h'00000000 |
| h'28 | HcBulkHeadED | R/W | h'00000000 |
| h'2C | HcBulkCurrentED | R/W | h'00000000 |
| h'30 | HcDoneHead | R/W | h'00000000 |
| h'34 | HcFmInterval | R/W | h'00002EDF |
| h'38 | HcFrameRemaining | R/W | h'00000000 |
| h'3C | HcFmNumber | R/W | h'00000000 |
| h'40 | HcPeriodicStart | R/W | h'00000000 |
| h'44 | HcLSThreshold | R/W | h'00000000 |
| h'48 | HcRhDescriptorA | R/W | h'01000002<br>h'01000003 |
| h'4C | HcRhDescriptorB | R/W | h'00000000 |
| h'50 | HcRhStatus | R/W | h'00000000 |
| h'54 | HcRhPortStatus0 | R/W | h'00000000 |
| h'58 | HcRhPortStatus1 | R/W | h'00000000 |
| h'5C | HcRhPortStatus2 | R/W | h'00000000 |

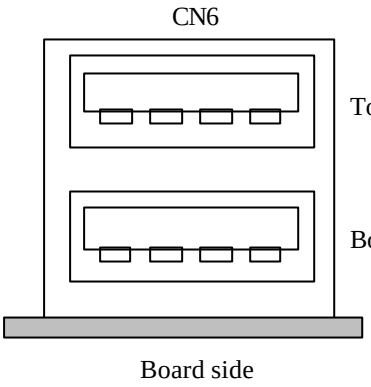
Table7.14 (2) Register map of USB controller

| Address | Register Name | R/W | Default Value |
|---|---|---|---|
| h'100 | HceControl Register | R/W | h'00000000 |
| h'104 | HceInput Register | R/W | h'000000xx |
| h'108 | HceOutput Register | R/W | h'000000xx |
| h'10C | HceStatus Register | R/W | h'00000000 |

**2. Pin assignment of USB interface connector(CN6)**

Figure7.11 shows pin assignments and functions of USB interface connector(CN6).

Connecter Model Name: USB1112C-D1




Pin assignment of USB connector

| Pin No. | Signal | I/O | Remark |
|---|---|---|---|
| 1 | Vcc | - | |
| 2 | DATA+ | I/O | Reserved |
| 3 | DATA+ | I/O | |
| 4 | GND | - | |

Figure7.11 USB interface connector(CN6)

## 7.3. PCMCIA Control

### 1. Block description

Figure7.12 shows a PCMCIA control block. As shown in figure7.13, the PCMCIA control block has a controller (Marubun-supplied MR-SHPC-01 V2), a 68-pin IC card connector (molex-supplied 53409-6810) and a power control IC(TI-supplied TPS2211IDB). The PCMCIA control block provides ATA card-interface and I/O card-interface via a 68-pin IC card connector.

This controller provides system-interface with ATA card based on PC card standard 97 and I/O card. This controller has following features.

- Support 68-pin card slot based on PC card standard97
- 2 memory windows and one I/O window incorporated
- Card access timing adjustment function incorporated
- Read /Write buffer incorporated
- Endian control on-chip circuit
- Support 5.0V/3.3V card
- External buffer is not necessary
- Interrupt steering function incorporated
- Power-down function incorporated
- Suspend function incorporated

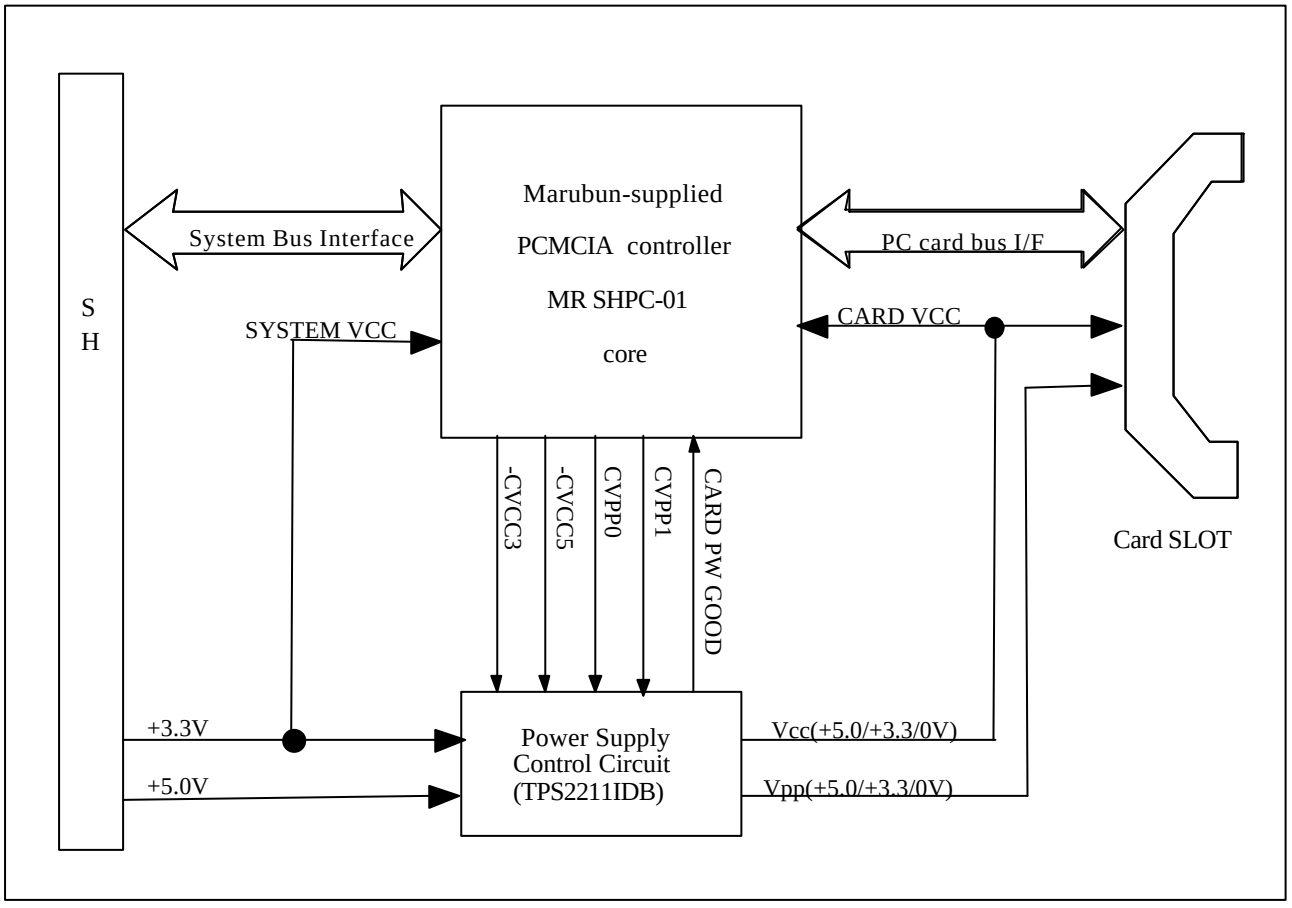



Figure7.12 PCMCIA I/F control block

## 2. 68-pin IC Card Connector (CN17) Pin Assignments

Table 7.15 lists the pin assignments of the 68-pin IC card connector (CN17).

Table 7.15 Pin Assignments of 68-pin IC Connector (CN17)

| Pin No. | Pin Name | Pin No. | Pin Name | Pin No. | Pin Name | Pin No. | Pin Name |
|---|---|---|---|---|---|---|---|
| 1 | GND | 18 | CARD_Vpp | 35 | GND | 52 | CARD_Vpp |
| 2 | PD3 | 19 | PA16 | 36 | /P_CD1 | 53 | PA22 |
| 3 | PD4 | 20 | PA15 | 37 | PD11 | 54 | PA23 |
| 4 | PD5 | 21 | PA12 | 38 | PD12 | 55 | PA24 |
| 5 | PD6 | 22 | PA7 | 39 | PD13 | 56 | PA25 |
| 6 | PD7 | 23 | PA6 | 40 | PD14 | 57 | /P_VS2 |
| 7 | /P_CE1 | 24 | PA5 | 41 | PD15 | 58 | /P_RESET |
| 8 | PA10 | 25 | PA4 | 42 | /P_CE2 | 59 | /P_WAIT |
| 9 | /P_OE | 26 | PA3 | 43 | /P_VS1 | 60 | /P_INPACK |
| 10 | PA11 | 27 | PA2 | 44 | /P_IORD | 61 | /P_REG |
| 11 | PA9 | 28 | PA1 | 45 | /P_IOWR | 62 | P_BVD2 |
| 12 | PA8 | 29 | PA0 | 46 | PA17 | 63 | P_BVD1 |
| 13 | PA13 | 30 | PD0 | 47 | PA18 | 64 | PD8 |
| 14 | PA14 | 31 | PD1 | 48 | PA19 | 65 | PD9 |
| 15 | /P_WE | 32 | PD2 | 49 | PA20 | 66 | PD10 |
| 16 | /P_RDY | 33 | /P_IOIS16 | 50 | PA21 | 67 | /P_CD2 |
| 17 | CARD_Vcc | 34 | GND | 51 | CARD_Vcc | 68 | GND |

## 3. Register Map

Table 7.16 shows a memory map of PCMCIA control register. All registers should be accessed in word size.

Table7.16 PCMCIA Control Register

| Address | Initial value | Register name | Function |
|---|---|---|---|
| h'B83FFFE4 (h'183FFFE4*1*2) | H'0000 | Mode register | Set operating mode of PCIC |
| h'B83FFFE6 (h'183FFFE6*1*2) | H'000C | Option register | Control option function |
| h'B83FFFE8 (h'183FFFE8*1*2) | H'03BF | Card status register | Monitor input signal from card |
| h'B83FFFEA (h'183FFFEA*1*2) | H'0000 | Interrupt source register | Show interrupt occurrence source |
| h'B83FFFEC (h'183FFFEC*1*2) | H'0000 | Interrupt control register | Control interrupt occurrence condition |
| h'B83FFFEE (h'183FFFEE*1*2) | H'0000 | Card power control register | Control card power and low power consumption |
| h'B83FFFF0 (h'183FFFF0*1*2) | H'7FC0 | Memory window 0 Control register 1 | Control system address area for memory access |
| h'B83FFFF2 (h'183FFFF2*1*2) | H'7FC0 | Memory window 1 Control register 1 | Control system address area for memory access |
| h'B83FFFF4 (h'183FFFF4*1*2) | H'7FC0 | I/O window Control register 1 | Control system address area for I/O access |
| h'B83FFFF6 (h'183FFFF6*1*2) | H'0000 | Memory window 0 Control register 2 | Control access condition to card |
| h'B83FFFF8 (h'183FFFF8*1*2) | H'0000 | Memory window 1 Control register 2 | Control access condition to card |
| h'B83FFFFA (h'183FFFFA*1*2) | H'0000 | I/O window Control register 2 | Control access condition to card |
| h'B83FFFFC (h'183FFFFC*1*2) | H'0000 | Card control register | Control card mode |
| h'B83FFFFE (h'183FFFFE*1*2) | H'5333 | Chip information register | Chip Revision |

*1 Physical address when MMU is used

*2 When MMU is used, do not cache at the time of TLB entry (TLB entry C-bit=0).

## 7.4. Memory Block

EPROM and FlashROM are placed at area 0, and SDRAM is placed at area3.

Figure7.13 shows a memory map of area 0. As shown in figure7.13, the assignment of EPROM (M7, M8) and FlashROM (M1, M2) can be changed depending on the state of SW5-5, SW5-4 and SW5-3.

EPROM and FlashROM are placed at area 0 which bus width is 32-bit.

Two EPROM and two FlashROM with 16-bit bus width are used to connect to 32-bit bus. Assignment of EPROM and FlashROM is as follows.

High order 16 bits: M7, M1

Low order 16 bits: M8, M2

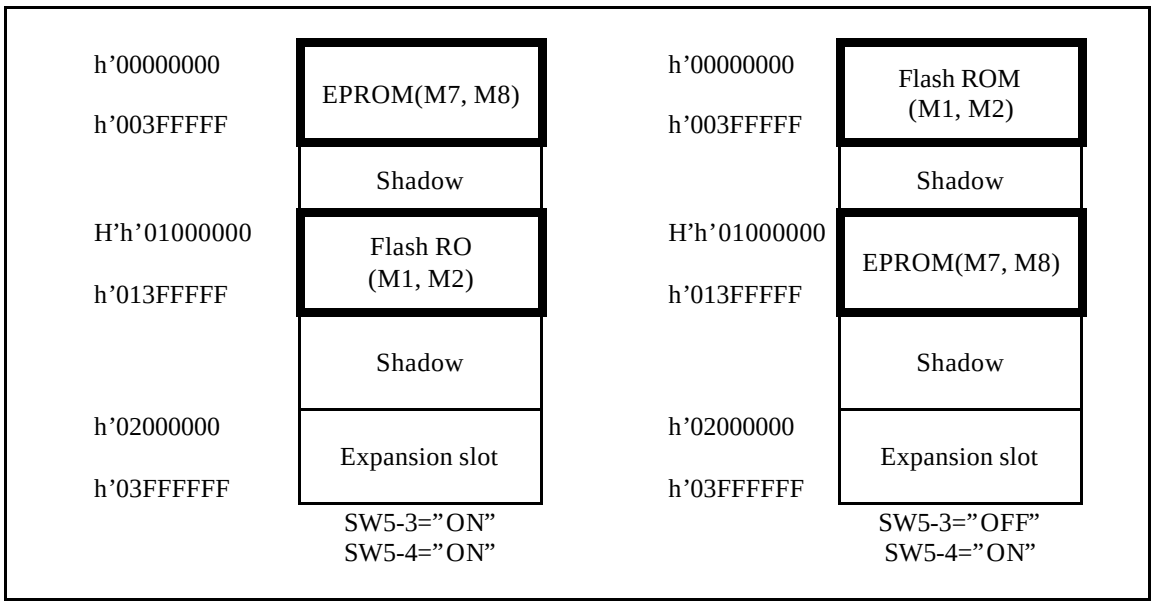



Figure7.13 Area 0 memory map

## 7.5. General-purpose Switch

Figure7.14 shows a configuration of general-purpose switches. SW6 to SW9 can detect ON or OFF state from the registers (h'B9000000(h'19000000 *1*2) and h'B9000002(h'19000002 *1 *2)) allocated on the memory map. This switch is useful for setting IP addresses.

Read addresses h'B9000000 (h'19000000 *1*2) and h'B9000002 (h'19000002 *1*2) by 16bit access. This register is a read only register.

*1 Physical address when MMU is used

*2 When MMU is used, do not cache at the time of TLB entry(TLB entry C-bit=0).

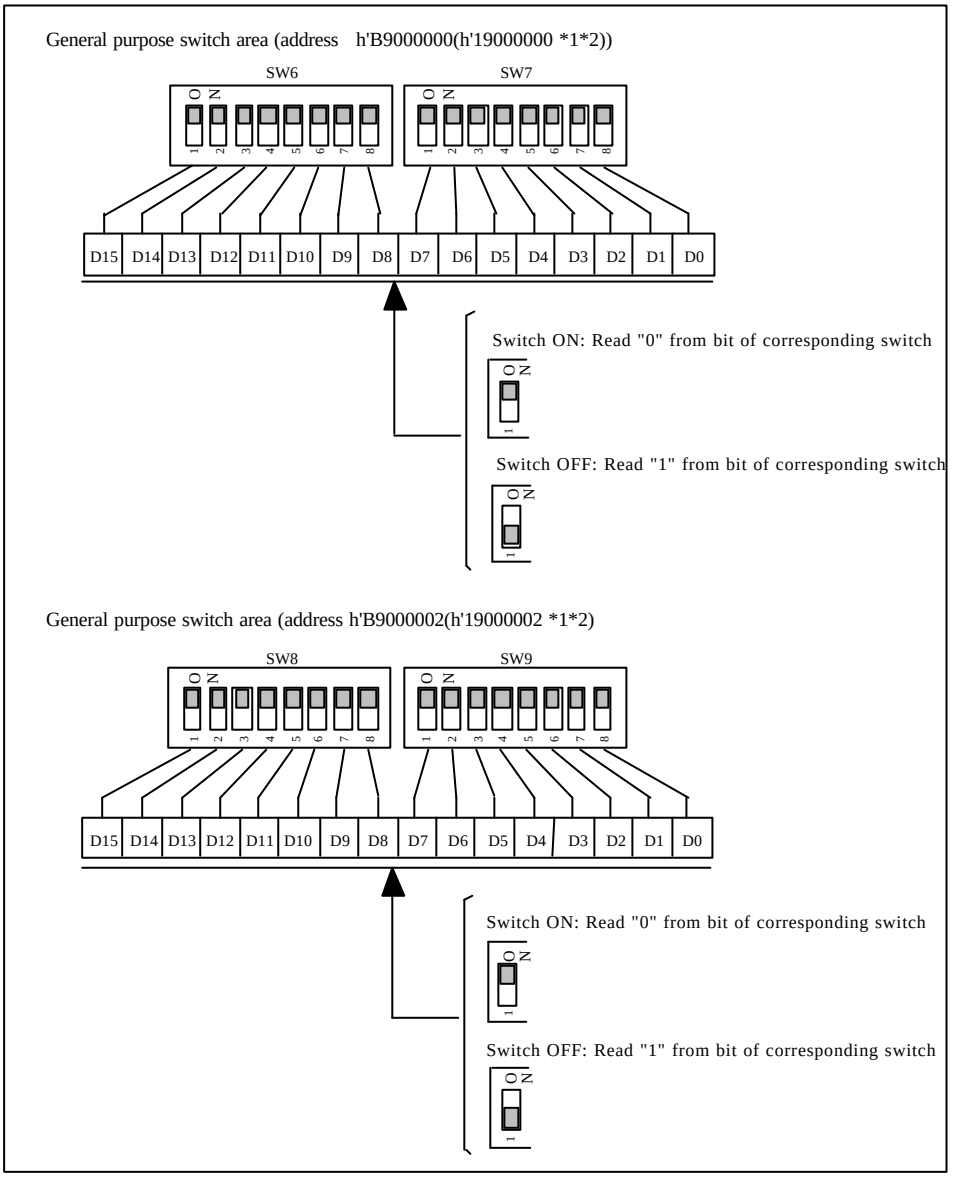



Figure7.14 Configuration of General-purpose Switches

## 7.6. 8-bit LED

Figure7.15 shows the configuration of the 8-bit LED. LED1 to LED8 are capable of controlling LED ON/OFF by writing data to the register (h'BA000000(h'1A000000*1*2)) allocated on the memory map. These LEDs are useful for checking the operation of programs.

Read and write to address h'BA000000(h'1A000000 *1 *2) in 16-bit width. 1and 0 written to each register become invalid because nothing is connected to D7-D0.

*1 Physical address when MMU is used

*2 When MMU is used, do not cache at the time of TLB entry(TLB entry C-bit=0).

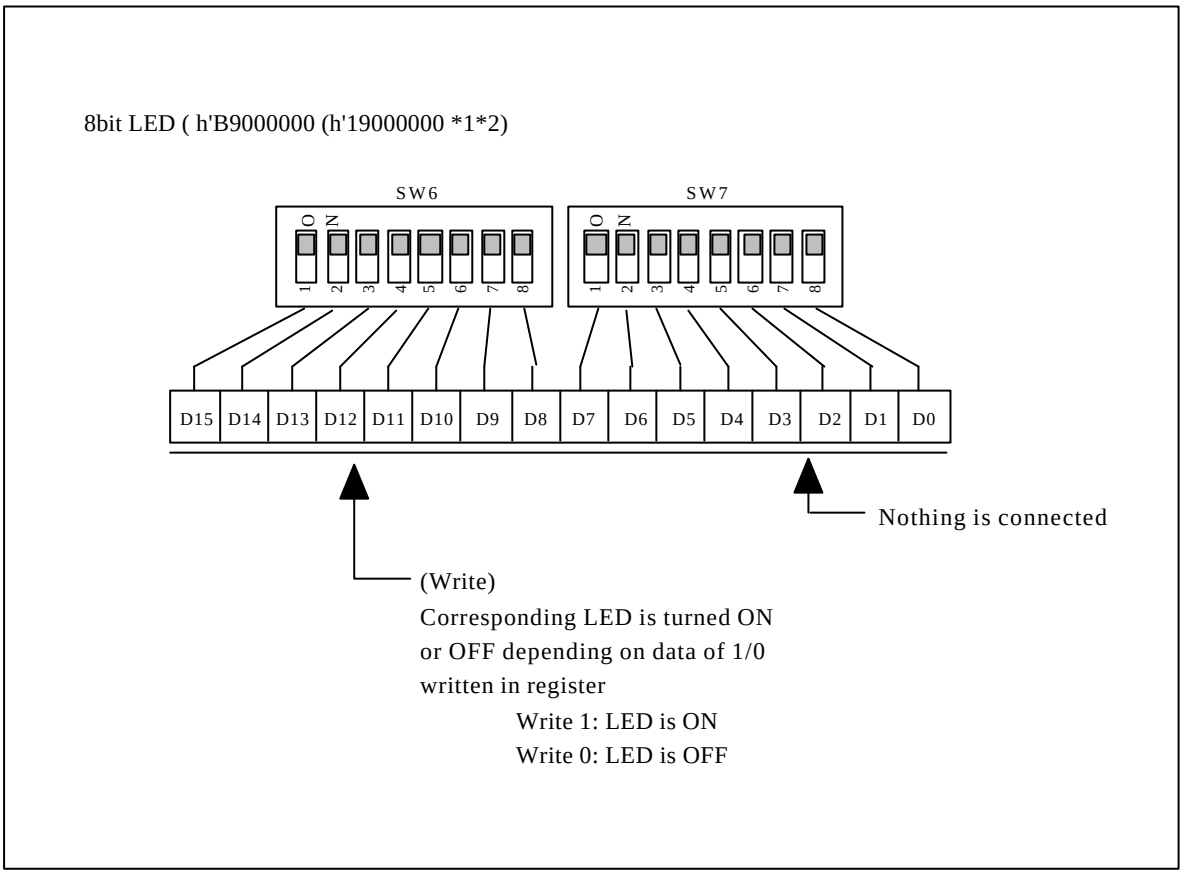



Figure7.15 Configuration of 8-bit LED

# 8. Interrupt Controller

The SolutionEngine has the interrupt controller FPGA1 (U17) that determines the priority of interrupts output from each device.

Table8.1 lists the outputs of IRL3-IRL0 signals of SH controlled by the interrupt controller.

Table8.1 Interrupt Level Cross Reference Table

| No. | Interrupt request source | Signal name | Interrupt level | SH7751R pin state IRL[3:0] | Other |
|---|---|---|---|---|---|
| 1 | Abort switch/ ROM emulator | NMI | - | - | |
| 2 | Expansion slot | ~SLOT_IRQ8 | 15 | 0000 | Expansion slot~IRQ8 signal |
| 3 | Expansion slot | ~SLOT_IRQ7 | 14 | 0001 | Expansion slot~IRQ7 signal |
| 4 | MR-SHPC-01-IRQ2 | ~PCIC_SIRQ2 | 13 | 0010 | MR-SHPC-01 register setting |
| 5 | Super I/O | INTR | 12 | 0011 | Super I/O |
| 6 | Expansion slot | ~SLOT_IRQ6 | 11 | 0100 | Expansion slot~IRQ6 signal |
| 7 | Expansion slot | ~SLOT_IRQ5 | 10 | 0101 | Expansion slot~IRQ5 signal |
| 8 | MR-SHPC-01-IRQ1 | ~PCIC_SIRQ1 | 9 | 0110 | MR-SHPC-01 register setting |
| 9 | Not defined | - | 8 | 0111 | |
| 10 | Expansion slot | ~SLOT_IRQ4 | 7 | 1000 | Expansion slot~IRQ4 signal |
| 11 | Expansion slot | ~SLOT_IRQ3 | 6 | 1001 | Expansion slot~IRQ3 signal |
| 12 | Not defined | - | 5 | 1010 | |
| 13 | Expansion slot | ~SLOT_IRQ2 | 4 | 1011 | Expansion slot~IRQ2 signal |
| 14 | Not defined | - | 3 | 1100 | |
| 15 | Expansion slot | ~SLOT_IRQ1 | 2 | 1101 | Expansion slot~IRQ1 signal |
| 16 | MR-SHPC-01-IRQ0 | ~PCIC_SIRQ0 | 1 | 1110 | MR-SHPC-01 register setting |

# 9. Expansion Slot (CN1)

## 9.1. Expansion Slot Pin Assignments

Table 9.1 lists the pin assignments of the expansion slot.

SH bus signals (data bus, address bus and control signals) are connected to the expansion slot via buffers (74ALVCH16244T, 245T). Electrical level is 3.3V. When LSI or the IC that need 5V interface are mounted, mount the IC with 3V→5V interface on the daughter board.

The symbols listed in Table 9.1 have the following meanings.

OUT: Output, IN: Input, BO: Buffer output, BI: Buffer input, P-UP: Pull up

Table 9.1 Expansion Slot Pin Assignments (Column A)

| No. | Pin Assignment | Type | I/O | Remarks | No. | Pin Assignment | Type | I/O | Remarks |
|---|---|---|---|---|---|---|---|---|---|
| A1 | GND | - | - | | A36 | A16 | BO | OUT | |
| A2 | CKIO | BO | OUT | | A37 | A18 | BO | OUT | |
| A3 | GND | - | - | | A38 | A20 | BO | OUT | |
| A4 | D0 | T | I/O | | A39 | A22 | BO | OUT | |
| A5 | D2 | T | I/O | | A40 | A24 | BO | OUT | |
| A6 | D4 | T | I/O | | A41 | GND | - | - | |
| A7 | D6 | T | I/O | | A42 | /DACK0 | BO | OUT | |
| A8 | GND | - | - | | A43 | /DREQ0 | BI,P-UP | IN | |
| A9 | D8 | T | I/O | | A44 | GND | - | - | |
| A10 | D10 | T | I/O | | A45 | /CS0 | BO | OUT | |
| A11 | D12 | T | I/O | | A46 | /CS2 | BO | OUT | |
| A12 | D14 | T | I/O | | A47 | /CS4 | BO | OUT | |
| A13 | GND | - | - | | A48 | /CS6 | BO | OUT | |
| A14 | D16 | T | | | A49 | GND | - | - | |
| A15 | D18 | T | I/O | | A50 | /RD | BO | OUT | |
| A16 | D20 | T | I/O | | A51 | GND | - | - | |
| A17 | D22 | T | I/O | | A52 | /WE0 | BO | OUT | |
| A18 | GND | - | - | | A53 | /WE2 | BO | OUT | |
| A19 | D24 | T | I/O | | A54 | GND | - | - | |
| A20 | D26 | T | I/O | | A55 | /WAIT0 | BI,P-UP | IN | |
| A21 | D28 | T | I/O | | A56 | /WAIT2 | BI,P-UP | IN | |
| A22 | D30 | T | I/O | | A57 | GND | - | - | |
| A23 | 3.3V | - | - | | A58 | /IRQ1 | BI,P-UP | IN | |
| A24 | 3.3V | - | - | | A59 | /IRQ3 | BI,P-UP | IN | |
| A25 | NC0 | Option | Option | Spare pin | A60 | /IRQ5 | BI,P-UP | IN | |
| A26 | A0 | BO | OUT | | A61 | /IRQ7 | BI,P-UP | IN | |
| A27 | A2 | BO | OUT | | A62 | +5V | - | - | |
| A28 | A4 | BO | OUT | | A63 | +5V | - | - | |
| A29 | A6 | BO | OUT | | A64 | NC1 | Option | Option | Reserve |
| A30 | GND | - | - | | A65 | /RES | BO | OUT | |
| A31 | A8 | BO | OUT | | A66 | A+5V | - | - | |
| A32 | A10 | BO | OUT | | A67 | A+5V | - | - | |
| A33 | A12 | BO | OUT | | A68 | NC3 | Option | Option | Spare pin |
| A34 | A14 | BO | OUT | | A69 | NC5 | Option | Option | Spare pin |
| A35 | GND | - | - | | A70 | NC7 | Option | Option | Spare pin |

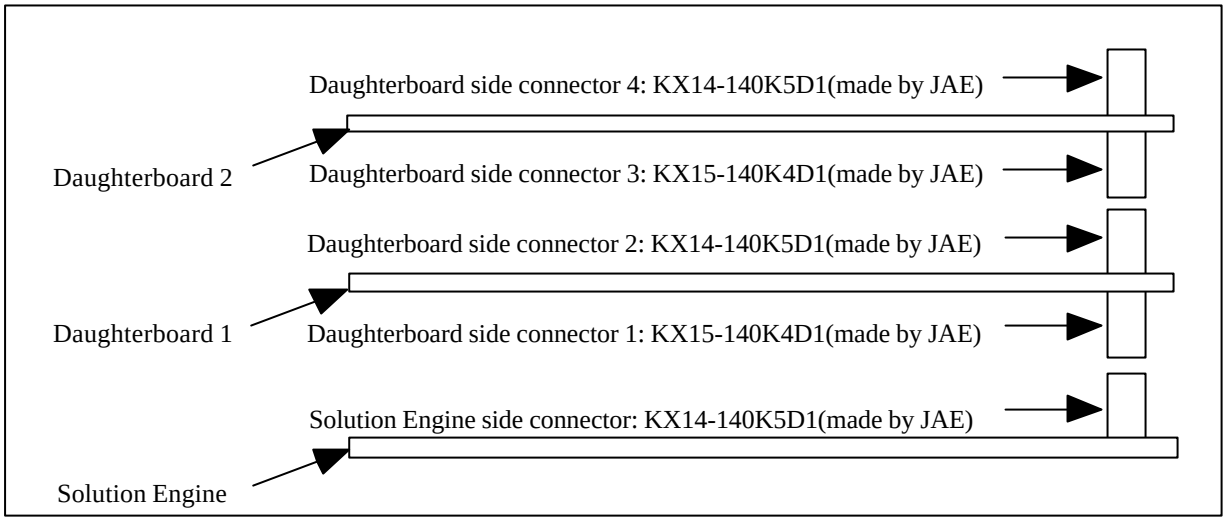
Table 9.2 Expansion Slot Pin Assignments (Column B)

| No. | Pin assignment | Type | I/O | Remarks | No. | Pin assignment | Type | I/O | Remarks |
|---|---|---|---|---|---|---|---|---|---|
| B1 | GND | - | - | | B36 | A17 | BO | OUT | |
| B2 | GND | - | - | | B37 | A19 | BO | OUT | |
| B3 | GND | - | - | | B38 | A21 | BO | OUT | |
| B4 | D1 | T | I/O | | B39 | A23 | BO | OUT | |
| B5 | D3 | T | I/O | | B40 | A25 | BO | OUT | |
| B6 | D5 | T | I/O | | B41 | GND | - | - | |
| B7 | D7 | T | I/O | | B42 | /DACK1 | BO | OUT | |
| B8 | GND | - | - | | B43 | /DREQ1 | BI,P-UP | IN | |
| B9 | D9 | T | I/O | | B44 | GND | - | - | |
| B10 | D11 | T | I/O | | B45 | /CS1 | BO | OUT | |
| B11 | D13 | T | I/O | | B46 | /CS3 | BO | OUT | |
| B12 | D15 | T | I/O | | B47 | /CS5 | BO | OUT | |
| B13 | GND | - | - | | B48 | R/W | BO | OUT | |
| B14 | D17 | T | I/O | | B49 | GND | - | - | |
| B15 | D19 | T | I/O | | B50 | /BS | BO | OUT | |
| B16 | D21 | T | I/O | | B51 | GND | - | - | |
| B17 | D23 | T | I/O | | B52 | /WE1 | BO | OUT | |
| B18 | GND | - | - | | B53 | /WE3 | BO | OUT | |
| B19 | D25 | T | I/O | | B54 | GND | - | - | |
| B20 | D27 | T | I/O | | B55 | /WAIT1 | BI,P-UP | IN | |
| B21 | D29 | T | I/O | | B56 | /WAIT3 | BI,P-UP | IN | |
| B22 | D31 | T | I/O | | B57 | GND | - | - | |
| B23 | 3.3V | - | - | | B58 | /IRQ2 | BI,P-UP | IN | |
| B24 | 3.3V | - | - | | B59 | /IRQ4 | BI,P-UP | IN | |
| B25 | 3.3V | - | - | | B60 | /IRQ6 | BI,P-UP | IN | |
| B26 | A1 | BO | OUT | | B61 | /IRQ8 | BI,P-UP | IN | |
| B27 | A3 | BO | OUT | | B62 | +5V | - | - | |
| B28 | A5 | BO | OUT | | B63 | +5V | - | - | |
| B29 | A7 | BO | OUT | | B64 | +5V | - | - | |
| B30 | GND | - | - | | B65 | +5V | - | - | |
| B31 | A9 | BO | OUT | | B66 | +5V | - | - | |
| B32 | A11 | BO | OUT | | B67 | NC2 | Option | Option | Spare pin |
| B33 | A13 | BO | OUT | | B68 | NC4 | Option | Option | Spare pin |
| B34 | A15 | BO | OUT | | B69 | NC6 | Option | Option | Spare pin |
| B35 | GND | - | - | | B70 | NC8 | Option | Option | Spare pin |

## 9.2. Expansion Slot Connector Configuration

Figure9.1 shows a connector configuration of the expansion slot. As shown in Figure9.1, an additional daughter board can be connected on top of the daughter board by mounting connectors on both sides of the daughter board. Maximum 4 daughter boards can be connected by using the expansion slot.

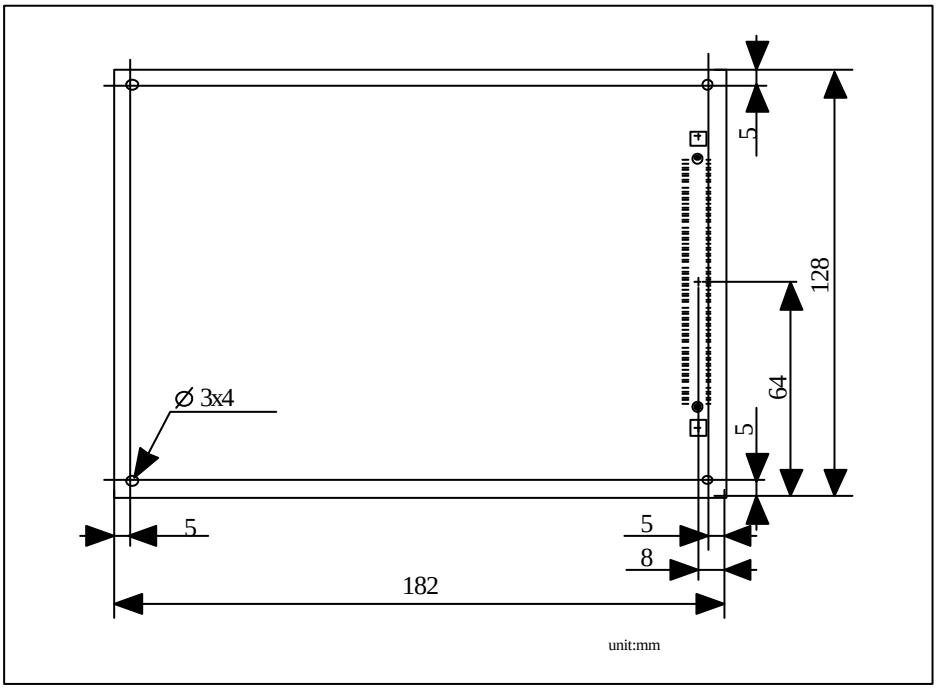
Figure9.1 Expansion Slot Connector Configuration




## 9.3. Daughter Board Dimensions

Figure9.2 shows the dimensions of the daughter board to be mounted on the Solution Engine. When a user design a daughter board originally, design the board with dimensions shown in figure9.2.

Figure9.2 Daughterboard Dimensions



# 10. I/O Connector (CN18)

Table10.1 lists the functions of I/O connector (CN18). I/O port, timer output pin and SCI signals are connected to the I/O connector. Use this connector to control by using the I/O port.

Solution Engine-side connector: 8800-080-170S(KEL)

I/O board-side connector: 8810-080-170L (right angle) (KEL)

/ 8810-080-170S (straight) (KEL)

Table 10.1 I/O Connector Functions

| Pin No. | Signal Name | Pin No. | Signal Name | Pin No. | Signal Name | Pin No. | Signal Name |
|---|---|---|---|---|---|---|---|
| 1 | +3.3V | 21 | NC | 41 | +5V | 61 | NC |
| 2 | +3.3V | 22 | NC | 42 | +5V | 62 | NC |
| 3 | +3.3V | 23 | NC | 43 | NC | 63 | NC |
| 4 | +3.3V | 24 | NC | 44 | NC | 64 | NC |
| 5 | RXD0/SCPT0 | 25 | NC | 45 | NC | 65 | GND |
| 6 | TXD0/SCPT0 | 26 | NC | 46 | NC | 66 | GND |
| 7 | SCK0/SCPT1 | 27 | GND | 47 | GND | 67 | NC |
| 8 | NC | 28 | GND | 48 | GND | 68 | NC |
| 9 | NC | 29 | NC | 49 | NC | 69 | NC |
| 10 | NC | 30 | NC | 50 | NC | 70 | NC |
| 11 | GND | 31 | NC | 51 | NC | 71 | GND |
| 12 | GND | 32 | NC | 52 | NC | 72 | GND |
| 13 | NC | 33 | NC | 53 | NC | 73 | +3.3V |
| 14 | NC | 34 | NC | 54 | NC | 74 | +3.3V |
| 15 | NC | 35 | NC | 55 | NC | 75 | NC |
| 16 | NC | 36 | NC | 56 | NC | 76 | NC |
| 17 | GND | 37 | GND | 57 | GND | 77 | NC |
| 18 | GND | 38 | GND | 58 | GND | 78 | NC |
| 19 | NC | 39 | +5V | 59 | GND | 79 | NC |
| 20 | NC | 40 | +5V | 60 | GND | 80 | NC |

# 11. Bus Controller Setting

SH7751R bus controller has 21 registers. Set the set values to each of 21 registers by using user programs when using various kinds of Solution Engine hardware. For the bus controller setting, refer to monitor program source (START.SRC) of sample software on the attached CD-ROM.

# 12. SH 7751R CPU Bus Interface

Table12.1 lists the pin assignment of the SH7751R CPU bus interface (CN20). SH7751R bus signal (data bus, address bus, control signal) is output to CPU bus interface connector directly. These signals can be used for tracing SH7751R signal to the emulator.

Connector Model Name: WR-120PB-VF-1(JAE)

Table12.1 Pin assignment of SH7751R CPU bus interface (CN20)

| Pin No. | Signal name | Pin No. | Signal name | Pin No. | Signal name | Pin No. | Signal name |
|---|---|---|---|---|---|---|---|
| 1 | Vcc | 31 | A22 | 61 | Vcc | 91 | GND |
| 2 | Vcc | 32 | A23 | 62 | Vcc | 92 | GND |
| 3 | NC | 33 | A24 | 63 | D12 | 93 | /WE2 |
| 4 | /BS1 | 34 | A25 | 64 | D13 | 94 | /WE3 |
| 5 | A2 | 35 | A0 | 65 | D14 | 95 | DQM0 |
| 6 | A3 | 36 | A1 | 66 | D15 | 96 | DQM1 |
| 7 | A4 | 37 | /CS5 | 67 | D16 | 97 | DQM2 |
| 8 | A5 | 38 | /CS6 | 68 | D17 | 98 | DQM3 |
| 9 | GND | 39 | GND | 69 | D18 | 99 | /RAS |
| 10 | GND | 40 | GND | 70 | D19 | 100 | NC |
| 11 | A6 | 41 | /CS1 | 71 | GND | 101 | GND |
| 12 | A7 | 42 | /CS4 | 72 | GND | 102 | GND |
| 13 | A8 | 43 | /CS0 | 73 | D20 | 103 | NC |
| 14 | A9 | 44 | NC | 74 | D21 | 104 | NC |
| 15 | A10 | 45 | D0 | 75 | D22 | 105 | /CS2 |
| 16 | A11 | 46 | D1 | 76 | D23 | 106 | /CS3 |
| 17 | A12 | 47 | D2 | 77 | D24 | 107 | TADPCS |
| 18 | A13 | 48 | D3 | 78 | D25 | 108 | NMIN |
| 19 | GND | 49 | GND | 79 | D26 | 109 | /RSTOUT |
| 20 | GND | 50 | GND | 80 | D27 | 110 | NMIOUT |
| 21 | A14 | 51 | D4 | 81 | GND | 111 | GND |
| 22 | A15 | 52 | D5 | 82 | GND | 112 | GND |
| 23 | A16 | 53 | D6 | 83 | D28 | 113 | NC |
| 24 | A17 | 54 | D7 | 84 | D29 | 114 | CKIO |
| 25 | A18 | 55 | D8 | 85 | D30 | 115 | NC |
| 26 | A19 | 56 | D9 | 86 | D31 | 116 | NC |
| 27 | A20 | 57 | D10 | 87 | RDWR | 117 | NC |
| 28 | A21 | 58 | D11 | 88 | /RD | 118 | NC |
| 29 | GND | 59 | Vcc | 89 | /WEO | 119 | Vcc |
| 30 | GND | 60 | Vcc | 90 | /WE1 | 120 | Vcc |

# 13. Usage of Monitor Program

## 13.1. Usage of Monitor Program

### 1. How to Connect the Host System

Connect the serial port of the host system to CN2 of the Solution Engine via a RS-232C cross cable.

After completion of serial connection, start communications software. Any communication software for personal computer communications can be used (Hyper terminal, Windows terminal and etc.). Set communication software as listed in the table13.1. The transfer rate can be selected with the DIP switch (SW5-1, 5-2) on the Solution Engine. For details, refer to Section 3.1 (4), "DIP Switch for setting baud rate(SW5)". This monitor program outputs CR+LF as a line feed code.

Table 13.1 Communication specifications

| | |
|---|---|
| Data communication | 8 bit |
| Parity | None |
| Stop bit | 1 bit |
| Control flow | Xon/Xoff |
| Data communication speed | 9600, 19200, 38400, 115200 bit/s |

### 2. Monitor Program Specifications

Figure13.1 shows the address map of the monitor program. Do not write at the area used by the monitor program (H'0DF00000-H'0DFFFFFF). For more details of each memory area, refer to Section 5, "Memory Map".

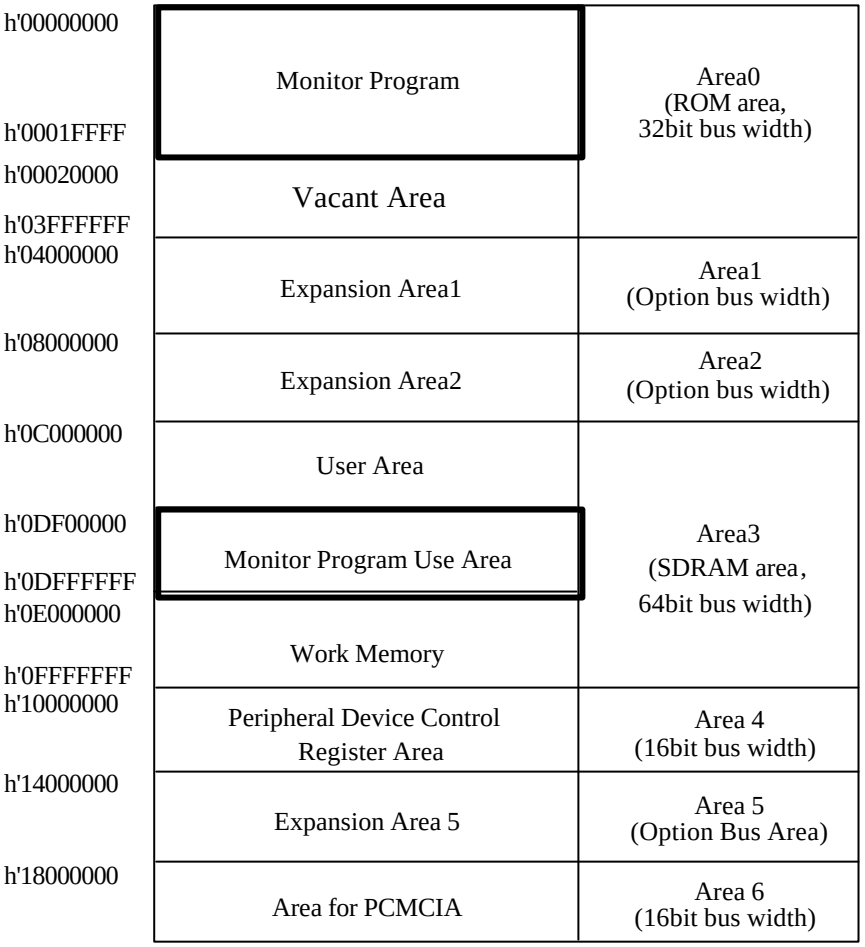



Figure 13.1 Memory Map (Real Memory Space)

### 3. Starting Monitor Program

The following starting message is displayed on the host system screen after connecting the Solution Engine to the host system via a RS232C cross cable and the monitor program is started.

```
==========================================================
                Self Debugger Ver x.x n
----------------------------------------------------------
  (C) Copyright 1999-2005.  Hitachi.Ltd. All rights reserved.
==========================================================
                                        H[elp] for help messages...
 Ready>
```

x.x means monitor program version. n is changed depending on endian.

A: Little endian

B: Big endian

## 4. Download user program

Use the ml command to transfer the user program to user RAM. Input "ml" in response to a command prompt as follows.

```
Ready>ml
```

After inputting the command, the following transfer request message is output from the monitor program, and the message is displayed on the host system screen.

```
Please Send A S-format Record
```

When the message is displayed, send the S-format object file by using the file transfer function of the communication software.

Address information is also added to the S-format object file. Allocate the object program according to this address information.

For it is a relocatable file that does not have specified address in object file, specify the offset address with the "ml" command as follows.

The specified address should be within the user area shown in Figure13.1.

Upon completion of loading into memory, the following message is displayed on the host system screen. (In this example, the program is loaded from address H'AC100000 of area 3.)

```
Start Addrs = AC100000
End Addrs = AC1000BC

Transfer complete
```

## 5. Display and change register contents

Before running the program, set the stack pointer for the program loaded into memory to R15. Because h'CF00000 has already been set to R15, change the setting as follows to set a stack pointer at different location.

```
Ready >rw r15 CEF0000
```

After completion of register setting, the information about all registers is displayed as follows and enter into command prompt status.

```
---General Registers-----------------------------------------------
      R0  =00000000       R1  =00000000       R2  =00000000       R3  =00000000
      R4  =00000000       R5  =00000000       R6  =00000000       R7  =00000000
      R8  =00000000       R9  =00000000       R10=00000000        R11=00000000
      R12=00000000        R13=00000000        R14=00000000        R15=0CEF0000
 R0_BANK=00000000 R1_BANK=00000000 R2_BANK=00000000 R3_BANK=00000000
 R4_BANK=00000000 R5_BANK=00000000 R6_BANK=00000000 R7_BANK=00000000
---Control Registers-----------------------------------------------
      SSR=600000E0        SPC=00000000        GBR=00000000        VBR=00000000
Ready >
```

## 6. Dump memory contents

Confirm the command transferred to user memory by using the md command. Input the md command as follows.

```
Ready >md ac100000
```

When the md command is executed, the data of the area (address H'AC100000 - H'AC1000FF in this example) of 256 bytes is dumped from the address input on the command line.

```
AC100000 03 61 21 41 13 62 21 42 23 63 21 43 33 64 21 44
AC100010 43 65 21 45 53 66 21 46 63 67 21 47 73 68 21 48
AC100020 83 69 21 49 93 6A 21 4A A3 6B 21 4B B3 6C 21 4C
AC100030 C3 6D 21 4D D3 6E 21 4E FF C3 1C D0 1C D1 01 21
AC100040 1C D0 1D D2 01 22 58 00 1F 42 33 4F 43 4F 83 4F
AC100050 93 4F A3 4F B3 4F C3 4F D3 4F E3 4F F3 4F 32 00
AC100060 3E 40 42 00 4E 40 82 08 8E 49 92 09 9E 48 A2 08
AC100070 AE 49 B2 09 BE 48 C2 08 CE 49 D2 09 DE 48 E2 08
AC100080 EE 49 F2 09 FE 48 F7 4F E7 4F D7 4F C7 4F B7 4F
AC100090 A7 4F 97 4F 87 4F 47 4F 37 4F 83 0F 0D 31 25 33
AC1000A0 4C 45 6D 47 09 00 FD AF 09 00 00 00 02 00 00 00
AC1000B0 00 00 11 0C FF FF 00 00 10 00 11 0C 36 9F EA BB
AC1000C0 20 50 0A 04 CC 18 41 10 04 CF 28 47 F1 FC 1F AF
AC1000D0 20 1E 04 43 95 45 D3 A8 79 10 88 C5 97 47 D1 2D
AC1000E0 82 86 80 70 B3 A2 6A 02 B7 FA 81 72 7D 22 1B B9
AC1000F0 D0 00 0A 00 98 04 97 AC EA 2F 9C 40 83 18 13 BB
```

## 7. Execute user program

Execute the program transferred to user memory with the g command. Input the g command as follows.

```
Ready >g ac100000
```

When the g command is input, h'AC100000 is set to the program counter (PC), and the program is executed from address h'AC100000. When either Ctrl+C key or the Abort switch (SW2) is pressed, the information about all registers is displayed as follows and the user program execution is suspended.

```
---General Registers-----------------------------------------------
      R0 =00000000      R1 =00000000      R2 =00000000      R3 =00000000
      R4 =00000000      R5 =00000000      R6 =00000000      R7 =00000000
      R8 =00000000      R9 =00000000      R10=00000000      R11=00000000
      R12=00000000      R13=00000000      R14=00000000      R15=0CEF0000
R0_BANK=00000000 R1_BANK=00000000 R2_BANK=00000000 R3_BANK=00000000
R4_BANK=00000000 R5_BANK=00000000 R6_BANK=00000000 R7_BANK=00000000
---Control Registers-----------------------------------------------
      SSR=600000E0      SPC=AC10003A      GBR=00000000      VBR=00000000
     SR =700000E1            MD RB BL M Q I S T =1 1 1 0 0 E 0 1
---System  Registers-----------------------------------------------
    MACH=00000000     MACL=00000000     PR =00000000     PC =AC10003A
```

## 8. Step user program

Step the program transferred to user memory with the s command.

Input the s command as follows.

```
Ready >s ac100000
```

When the s command is executed, instruction of executed address is displayed.

```
Ready >s ac100000
AC100000 0009 NOP

Ready >
```

## 9. Set Breakpoint

Set a breakpoint with the bs command. Input the bs command as follows. The breakpoint is set at address h'AC100010 by inputting the bs command. When the program is executed under this condition, a break is occurred at address h'AC100010 and user program is aborted.

This break is generated by replacing the instruction of the said address with an illegal instruction. It is impossible to break read only memory.

```
Ready >bs ac100010
```

Use the bi command to disable the breakpoint set previously.

```
Ready >bi
================
< IGNORE BREAK >
================
PC Break Address
[0000] : AC100010
================
```

Using the be command enables the breakpoint disabled by the bi command

```
Ready >be
================
< ENABLE BREAK >
================
PC Break Address
[0000] : AC100010
================
```

## 10. Change memory contents

Use the me command to change memory data. Input the me command as follows. If characters other than hexadecimal number are input, the program come out of the me command, and goes into command wait status.

```
Ready >me ac100000

AC100000 03-
AC100001 61-.
```

## 11. Writing to the Flash ROM

Use the “fl” command to transfer the user program to user RAM and to write to Flash ROM.

Input the “fl” in command standby mode as follows.

To write to Flash ROM, erase Flash ROM and start writing to Flash ROM.

Figure13.2 shows the procedure to execute user programs from Flash ROM after completion of writing user programs to Flash ROM.

Offset is necessary. When offset is not used, set offset to 0.

```
Ready >fl    offset
```

After input, following transfer request message is output from the monitor program and the message is displayed on the host system screen.

```
Flash ROM data copy to RAM
Please Send A S-format Record
```

After the message is displayed, send the S-format object file by using the file transfer function of communication software.

S-format object file has address information. Place object program according to this address information.

If it is relocatable file which address is not specified in object file, specify offset address with the fl command.

Specify the address within the user area shown in figure13.1. Following message is displayed on the host system screen after completion of loading to memory.

(In this example, program is loaded from h’A0000000 address of area3)

```
Start Addrs = A0000000
End Addrs = A00044BE

Transfer complete
```

Erasion of Flash ROM is started and the following message is displayed.

```
Flash chip erase:
```

After completion of erasing Flash ROM, the following message is displayed and writing is started.

```
Flash chip erase: complete
Program :
```

After completion of writing, command prompt is displayed on the screen.

```
Program :complete
Flash write complete
Ready
```

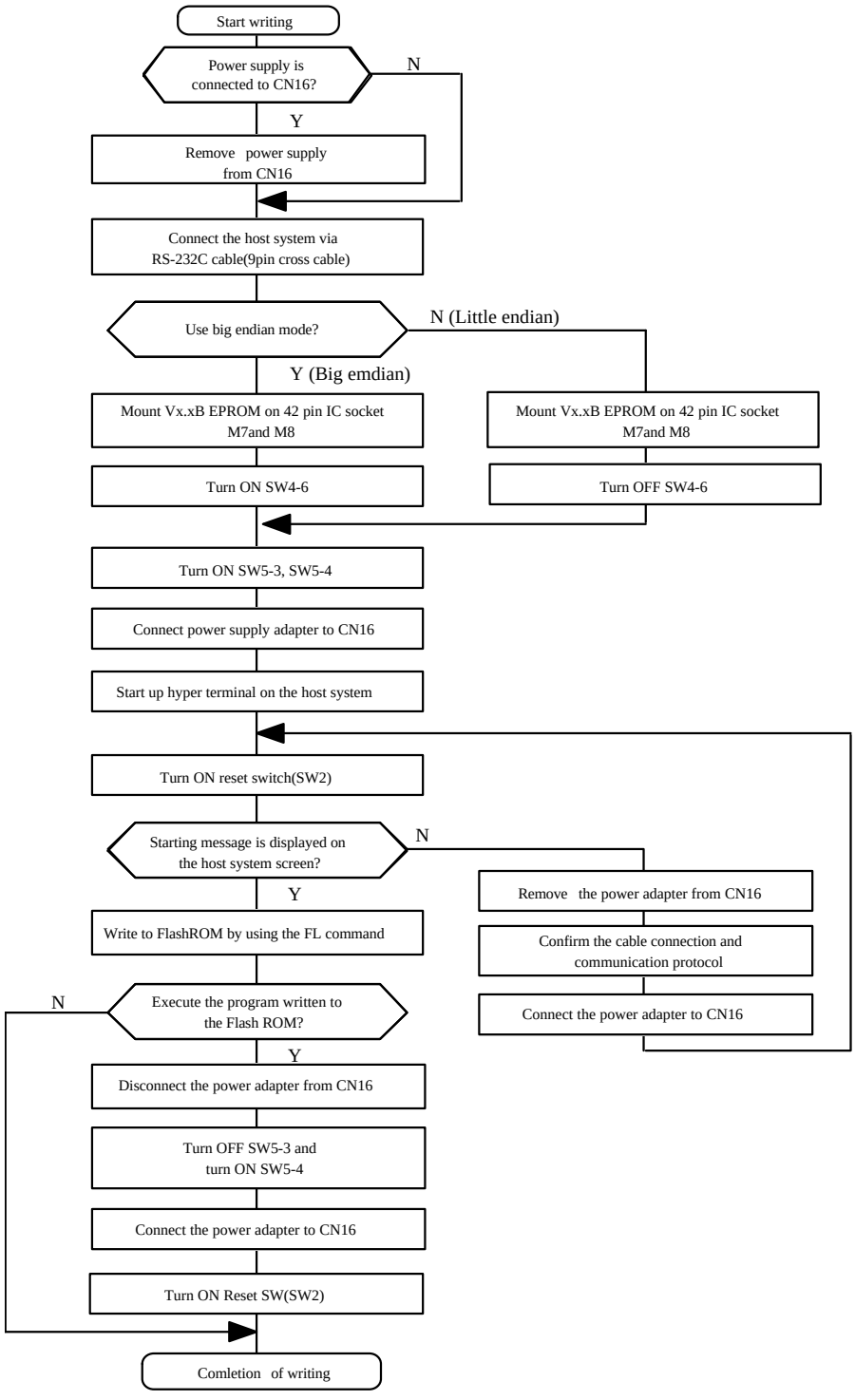



Figure13.2 Procedure to write to Flash ROM

## 13.2. Monitor Program Function List

Table 13.2 lists the commands of the monitor program.

Table 13.2 List of Monitor Functions

| Classification | Command | Description |
|---|---|---|
| Host PC Interface | ML(Memory Load) | Download object from the host |
| Write to Flash ROM | FL(Flash Load) | Write to Flash |
| Register display | RR(Register Read) | Read all register of SH |
| | RW(Register Write) | Write to the specific register of SH |
| | RC(Register Clear) | Clear all register of SH |
| Memory | ME(Memory Edit) | Edit memory |
| | MD(Memory Dump) | Dump memory |
| | MF(Memory Fill) | Fill memory |
| | DA(Disassemble) | Disassemble |
| Execute Program | G(Go) | Execute program |
| | S(Step) | Step program |
| | BS(Breakpoint Set) | Set breakpoints |
| | BD(Breakpoint Delete) | Delete a breakpoint |
| | BC(Breakpoint Clear) | Delete all breakpoints |
| | BE(Breakpoint Enable) | Break at breakpoint |
| | BI(Break Ignore) | Ignore breakpoint |
| Other | H(Help) | Describe command format of commands |

# 14. Command

| Command | Function |
|---|---|
| **ML (Memory Load)** | Load objects from the host. |
| Option<br>None | |

Format

ML ( offset address )

Example: Ready >ML

Ready >ML AC000000

(Note) Program can be loaded by specifying offset addressing only when loaded program does not use absolute addressing (only when loaded program is relocatable).

The operation is not guaranteed when loading a program that executes jumps by absolute addressing, by using offset addressing. Therefore, do not use offset addressing but load to linking address.

| Command | Function |
|---|---|
| FL(Flash Load) | Write data and programs to Flash ROM |
| Option | |
| offset | |

Format

FL(offset)

Example: Ready>FL

Write FL command to Flash ROM as follows.

(1) Make Flash ROM image on SDRAM

Make Flash ROM image on SDRAM by copying Flash ROM data to the first 4-Mbyte area of the SDRAM address

(2) Download S format object file

Transfer MOTOROLA S format object file on PC to SDRAM. MOTOROLA S format object files should be transferred to the following SDRAM address.

SDRAM address =MOTOROLA S format address + (SDRAM top address(H'0c000000)-offset)

Upper 4 bits of MOTOROLA S format address are ignored.

(3) Delete FlashROM data

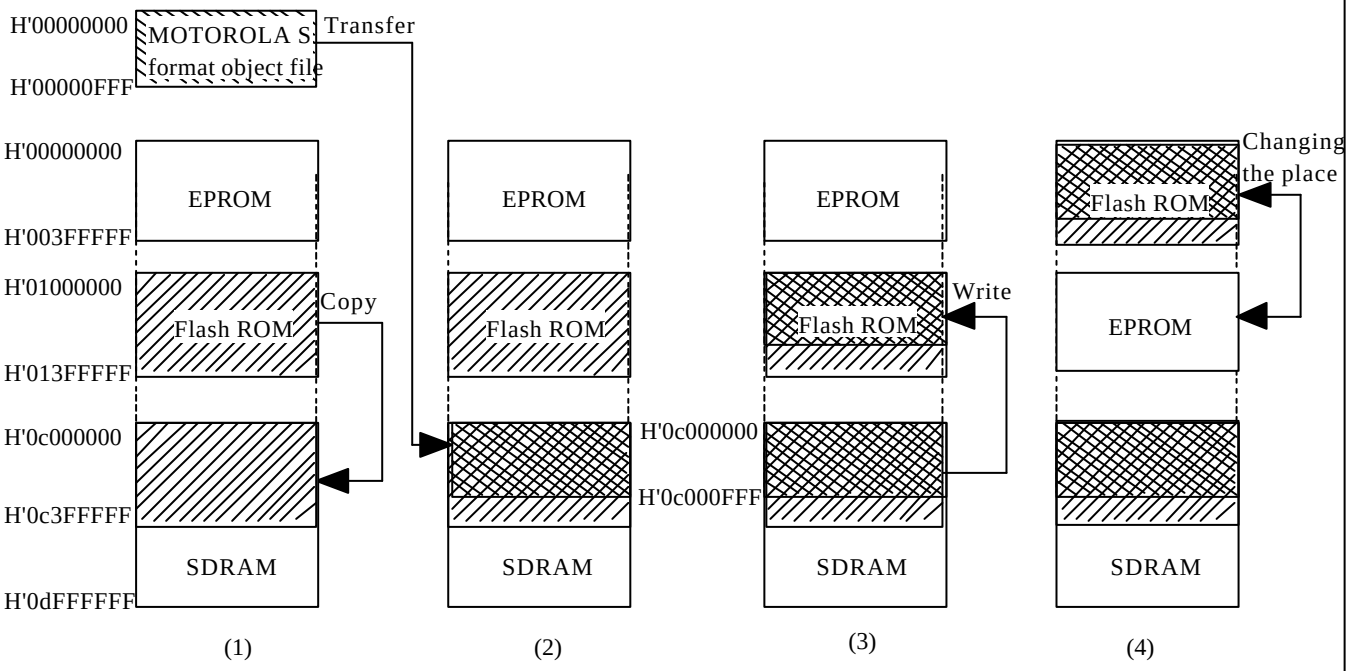
Delete all Flash ROM data after the transfer.

(4) Writing

Write first 4-Mbyte data of the SDRAM address to Flash ROM.

(5) Changing place between Flash ROM and EPROM

Change the place between EPROM and Flash ROM by DIP switch setting after writing. Programs written to Flash ROM can b by changing the place.



**(offset)**

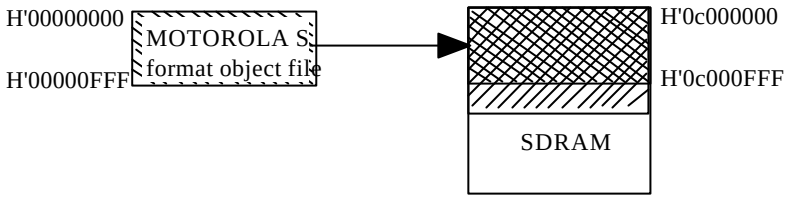
The transfer address for downloading MOTOROLA S format object files to SDRAM can be adjusted by specifying the off on command input. The final Flash ROM address can be specified by adjusting the transfer address.

1. When running the program written to Flash ROM right after power on reset

(1) Place the program at area 0 of SH microcomputer to run the program on Flash ROM. Link the program to place the program at area 0 on MOTOROLA S format object file generation.

(2) It is necessary to set the offset to 0 to download the object generated in process (1). Ttansfer the object to top address of SDRAM by setting the offset to 0.

Example: Ready>FL 0




2. When writing the data of address H'1000 to address H'0.

(1) MOTOROLA S format object file should be transferred to the following SDRAM address.

SDRAM address = MOTOROLA S format address +(SDRAM top address(H'0c000000)-offset)

Specify the offset to write the object of address H'1000 to address H'0. Obtain the offset as follows.

offset = MOTOROLA S format address+SDRAM top address (H'0c000000)-SDRAM address

=H'1000+H'c000000-H'c000000

=H'1000

Example: Ready>FL 1000

[NOTE]

(1) Be sure to specify the offset. It is impossible to write normally without specifying the offset.

(2) The object file data of address from H'0000 to H'0FFF is not transferred to user memory when specifying the offset 1000. Flash ROM data is not changed when specifying the offset 1000 and writing the object file with address from H'0000 to H'0FFF to Flash ROM.

| Command | Function |
|---|---|
| **RR (Register Read)** | Read all registers. |
| Option | |
| None | |
| Format<br>RR | |
| Example: Ready >RR | |

| Command | Function |
|---|---|
| **RW (Register Write)** | Writes to the corresponding register. |
| Option | |
| None | |
| Format<br>RW <regname> <data> | |
| Example: Ready >RW R0 12AB | |

| Command | Function |
|---|---|
| **RC (Register Clear)** | Clears all registers to 0. |
| Option | |
| None | |
| Format<br>RC | |
| Example: Ready >RC | |

| Command | Function |
|---|---|
| **ME (Memory Edit)** | Edit memory. |
| Option | |
| -W,-L | Word access, Long word access |

Format

ME <address> ( option )

Example: Ready >ME AC000000

Ready >ME AC000000 -W

Ready >ME AC000000 -L

| Command | Function |
|---|---|
| **MD (Memory Dump)** | Dumps memory. |
| Option | |
| -A | Displays in ASCII code. |

Format

MD ( start address ) ( end address )

Example: Ready >MD

Ready >MD 0

Ready >MD 0 200 -A

| Command | Function |
|---|---|
| **MF (Memory Fill)** | Fills memory. |
| Option | |
| - ( data ) | Fills by specified data. |

Format

MF ( start address ) ( end address ) ( option )

Example: Ready >MF

Ready >MF AC000000 AC000200

Ready >MF AC000000 AC000200 -55

| Command | Function |
|---|---|
| **DA (Disassemble)** | Disassembles from the specified address. |
| Option<br>None | |
| Format<br>DA ( start address ) | |
| Example: Ready >DA AC000000 | |

| Command | Function |
|---|---|
| **G (Go)** | Executes from the specified address. |
| Option<br>None | |
| Format<br>G ( start address ) | |
| Example: Ready >G AC000000 | |

| Command | Function |
|---|---|
| **S (Step)** | Step from specified address. |
| Option<br>None | |
| Format<br>S (start address) | |
| Example: Ready >S AC000000 | |

| Command | Function |
|---|---|
| **BS (Breakpoint Set)** | Set breakpoint |
| Option<br>None | |
| Format<br>BS ( address) | |
| Example: Ready >BS AC000000 | |

| Command | Function |
|---|---|
| **BD (Break Delete)** | Deletes breakpoints |
| Option<br>None | |
| Format<br>BD <address> | |
| Example: Ready >BD 45C | |

| Command | Function |
|---|---|
| **BC (Break Clear)** | Deletes all breakpoints. |
| Option<br>None | |
| Format<br>BC | |
| Example: Ready >BC | |

| Command | Function |
|---|---|
| **BE (Break Enable)** | Breaks at breakpoints. |
| Option<br>None | |
| Format<br>BE | |
| Example: Ready >BE | |

| Command | Function |
|---|---|
| **BI (Break Ignore)** | Ignores breakpoints. |
| Option | |
| None | |
| Format | |
| BI | |
| Example: Ready >BI | |

| Command | Function |
|---|---|
| **H (Help)** | Describes the monitor system commands. |
| Option | |
| ( number ) | Describes concerned items in detail. |
| Format | |
| H ( number ) | |

```
Ready >h        <----Display help menu
-------------------------------------------------------------------------------
Debugger Help:    Address or data must be specified by hex(need not H')
-------------------------------------------------------------------------------
[1] General          ---H
[2] Register         ---RC    RR      RW
[3] Break Point      ---BS    BR      BD      BC      BI      BE
[4] Memory           ---ML    ME      MD      MF      FL
[5] Disassemble      ---DA
[6] Start User Program        --- G     S
.....H[elp] number(or class), for more information.
-------------------------------------------------------------------------------
Ready>H4        <----Display detailed help of command related to [4] Memory
-------------------------------------------------------------------------------
Debugger Help: [4] Memory
-------------------------------------------------------------------------------
Memory Load      : M[em] L[oad]
Memory Edit      : M[em] E[dit] startAdrs [size(-W, -L)]
Memory Dump      : M[em] D[ump] [startAdrs] [endAdrs] [ASCIIcode(-A)]
Memory Fill      : M[em] F[ill] [startAdrs] [endAdrs] [Data(-Data)]
Flash Load       : F[lash] L[oad] [offsetAdrs]
-------------------------------------------------------------------------------
Ready >
```

SH7751R Solution Engine (MS7751RSE01) Overview

Publication Date: 4th Edition, Dec. 2002

Published by: Platform Design Dept.
Hitachi ULSI Systems Co., Ltd.

Edit by: Platform Design Dept.
Hitachi ULSI Systems Co., Ltd.