SH7707 Reference Platform

for Microsoft® Windows® CE (PFM-DS4)

Application Notes (Hardware Guide)

HITACHI

ADE-000-000
Rev.0.1
12/10/98
Hitachi Company or Division

Hitachi semiconductor

Notice

When using this document, keep the following in mind:

1. This document may, wholly or partially, be subject to change without notice.
2. All rights are reserved: No one is permitted to reproduce or duplicate, in any form, the whole or part of this document without Hitachi's permission.
3. Hitachi will not be held responsible for any damage to the user that may result from accidents or any other reasons during operation of the user's unit according to this document.
4. Circuitry and other examples described herein are meant merely to indicate the characteristics and performance of Hitachi's semiconductor products. Hitachi assumes no responsibility for any intellectual property claims or other problems that may result from applications based on the examples described herein.
5. No license is granted by implication or otherwise under any patents or other rights of any third party or Hitachi, Ltd.
6. MEDICAL APPLICATIONS: Hitachi's products are not authorized for use in MEDICAL APPLICATIONS without the written consent of the appropriate officer of Hitachi's sales company. Such use includes, but is not limited to, use in life support systems. Buyers of Hitachi's products are requested to notify the relevant Hitachi sales offices when planning to use the products in MEDICAL APPLICATIONS.

Contents

Section 1 Overview

Microsoft® Windows® CE operating system has been produced for use with mobile computing and multimedia products. The handheld PC (H/PC) is one of the applications of Windows® CE, and it attracts a great deal of attention because its infrared communication and RS-232C interface make data transfer possible between a handheld PC and a notebook or desktop computer.

The mobile computer market needs a processor incorporating the peripheral functions necessary for Windows® CE, in addition to need for high performance, low power consumption, and high integration. To meet this demand, Hitachi has developed the SH7707, a member of the SuperH RISC engine family, based on the 32-bit SuperH RISC CPU core SH-3, which has been adopted by many application systems because of its well-balanced performance and power consumption, and high code efficiency due to 16-bit fixed-length instructions. The SH7707 operates at a maximum frequency of 60 MHz and achieves a high performance of 60 MIPS. It includes an 8-kbyte cache and the interface circuits required to use Windows® CE, and allows more compact application systems to be produced easily at lower cost.

Hitachi has also developed the SH7707 reference platform PFM-DS4 for Windows® CE (hereafter called the reference platform). This reference platform provides the functions necessary for Windows® CE by making the best use of the SH7707 on-chip functions.

The reference platform includes the device driver software for Windows® CE Version 1.01 and Version 2.0 (LCD, IrDA, keyboard, touch panel, audio replay, PCMCIA, and RS-232C drivers). In addition, sample programs for hardware debugging are also supplied for use as a reference when the user makes programs.

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

Section 2 Basic Specifications

Table 2.1 shows the basic specifications of the reference platform PFM-DS4, which supports the functions necessary for the Microsoft® Windows® CE operating system Version 1.01 and Version 2.0. Table 2.2 shows the design specifications.

Table 2.1 Basic Specifications

Item		Specifications
CPU	Operating frequency	Internal operating frequency: 58.9824 MHz Internal bus frequency: 14.7456 MHz External bus frequency: 29.4912 MHz
	Bus width	32 bits
DRAM	Capacity, configuration, and access time	16 Mbytes (4 Mwords × 16 bits × 2), 60 ns
Flash memory	Capacity, configuration, and access time	8 Mbytes (128 kwords × 8 bits × 8), 80 ns
Display (LCD)	Display format	480 (W) × 240 (H) dots
	Color	Monochrome, four grayscales
Interface	RS-232C serial interface	**Baud rate:** 38400 bps max. (9600 bps when the monitor program is used) **Synchronization mode:** Asynchronous or clock synchronous mode can be selected
	PCMCIA	Conforms to PCMCIA Rev. 2.1/JEIDA Ver. 4.2*
	Touch panel	Resolution: Vertical: 84 mm/1024 dots Horizontal: 115 mm/1024 dots
	Audio	Sampling frequency: 22 kHz max.
	Keyboard	8 × 10 key matrix
	IrDA	Conforms to IIR-SIR Ver. 1.0

Note: The card slot is for 5-V operation. To use a 3.3-V power suply, change the PC card slot.

Table 2.2 Design Specifications

Item			Specifications
CPU			SH7707 (HD6417707)
Memory	DRAM		HM5165165ATT-6 (8 Mbytes × 2)
	Flash memory		HN29WT800T8 (1 Mbyte × 8)
Display	LCD	Controller	On-chip LCD controller of SH7707
		Panel	Maximum display size: 480 dots (W) × 360 dots (H)
			Displaying area: 118.6 mm (W) × 80 mm (H)
			Dot size: 0.22 mm (W) × 0.22 mm (H)
			Dot interval: 0.02 mm
Interface	SCI		On-chip serial interface of the SH7707 and RS-232C driver
	PCMCIA		On-chip PC card controller of the SH7707, HD151015T (level shifter), and a slot (5 V)
	Touch panel		On-chip 10-bit A/D converter (two channels) of SH7707
	Sound		On-chip 8-bit D/A converter (one channel) of SH7707 and current-driven amplifier
	Keyboard		Scan-type 60-key keyboard unit
	IrDA		On-chip serial interface and IrDA interface of SH7707, and light-receiving and light-emitting devices
Keyboard and touch panel timer	FPGA		EPM7160-EQC100-10 manufactured by Altera Corporation
Board	External dimensions		230 mm (W) × 120 mm (D)
	Specifications		Glass-epoxy 6-layer printed board

Section 3 System Configuration

3.1 System Configuration Diagram

Figure 3.1 shows the configuration of the reference platform.

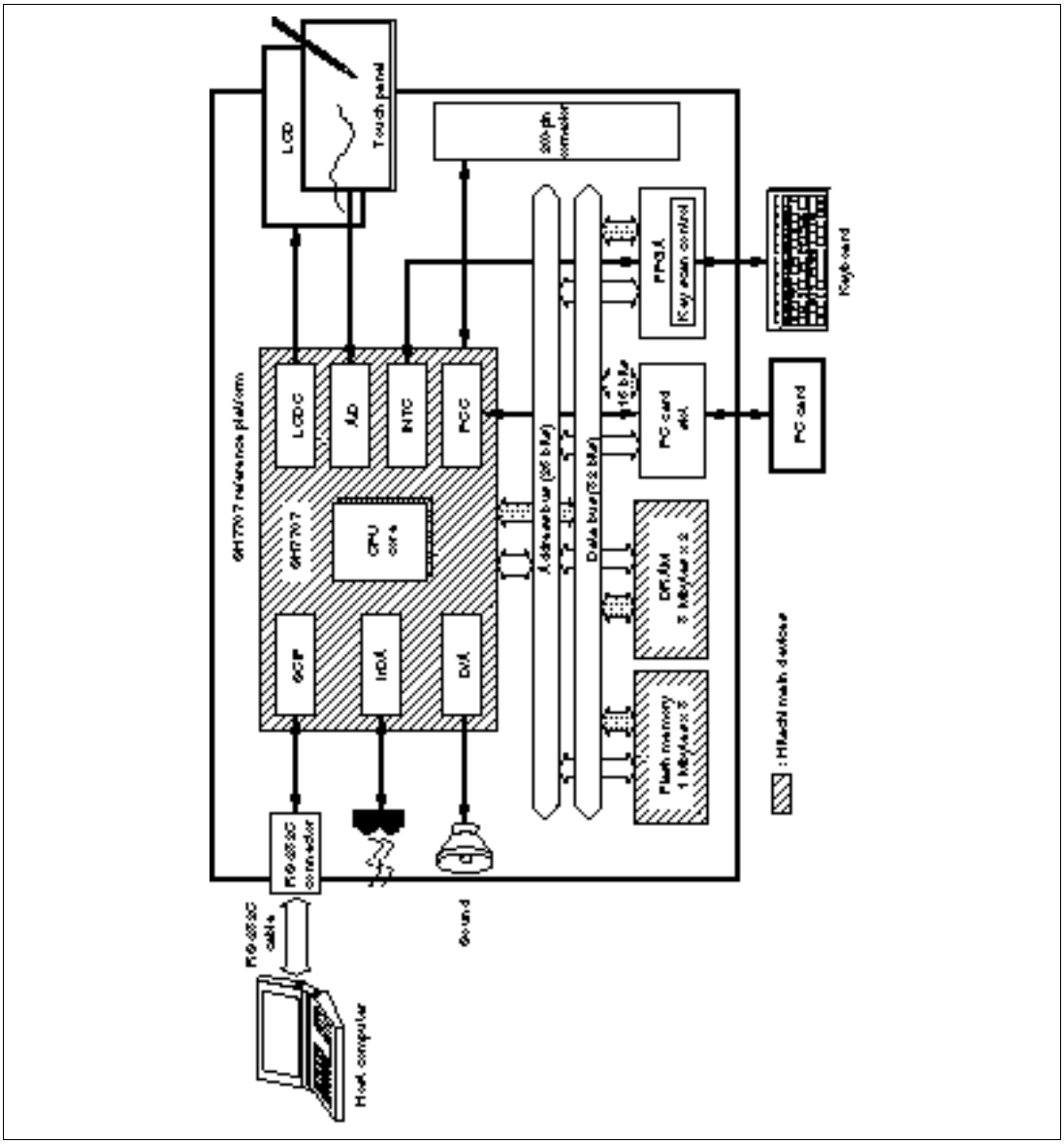


Figure 3.1 Reference Platform Configuration

3.2 Memory Map

In the reference platform, the physical space is divided into seven areas, to which memory devices and a PC card are connected, as shown in figure 3.2.

Area No.	Physical Address	Function	Size
0	H'0000 0000 H'007F FFFF H'03FF FFFF	Flash memory (8 Mbytes)	64 Mbytes
1	H'0400 0000 H'04FF FFFF H'07FF FFFF	On-chip I/O	64 Mbytes
2	H'0800 0000 H'0BFF FFFF	(Not used)	64 Mbytes
3	H'0C00 0000 H'0C FF FFFF H'0FFF FFFF	DRAM (16 Mbytes)	64 Mbytes
4	H'1000 0000 H'13FF FFFF	Companion FPGA	64 Mbytes
5	H'1400 0000 H'17FF FFFF	(Not used)	64 Mbytes
6	H'1800 0000 H'1BFF FFFF	PCMCIA interface (for memory and I/O cards)	64 Mbytes

Figure 3.2 Memory Map

3.3 Mode Setting

Specify the reference platform mode by using the jumpers on the board. Figure 3.3 shows the mode setting.

Jumper	Description	Default Setting
SW1	Reset	Open
SW2	Clock mode	Open
SW3	Clock mode	Closed
SW4	Clock mode	Closed
SW5	Bus width of area 0	Open
SW6	Bus width of area 0	Open
SW7	Endian	Open

(1) Clock Mode:

SW4	SW3	SW2	Mode
0	0	1	1

FRQCR	Clock Ratio (I : B : P)
0112	8 : 4 : 2

Note: The FRQCR is the register for specifying the ratio of internal clock (I), external bus clock (B), and internal peripheral clock (P). For other mode setting and how to set the FRQCR, refer to the SH7707 Hardware Manual.

(2) Bus Width of Area 0:

SW6	SW5	Memory Size
0	0	Reserved (illegal setting)
0	1	8 bits
1	0	16 bits
1	1	32 bits

(3) Endian

SW7	Endian
0	Big endian
1	Little endian

Note: When the endian is changed, the OS must also be changed.

: The shaded parts show the settings of the reference platform.

Figure 3.3 Reference Platform Mode and Jumper Setting

3.4 Interrupts

Table 3.1 shows the interrupts.

Table 3.1 Interrupts

Interrupt Pin	SH7707 Pin No.	Module Using the Interrupt	Description
NMI	7	Not used	—
IREQ0	8	Not used	—
IREQ1	9	FPGA	A/D conversion start for the keyboard or touch panel
IREQ2	10	SCI	CD carrier detection (when a modem is connected)
IREQ3	11	Not used	—
IREQ4	12	Touch panel	Detection of pen pressed down on the touch panel
IREQ5	176	Must not be used	This pin is used for a special purpose by the reference board and must not be used as an interrupt pin by the user.
PCC0RDY	127	PCMCIA	I/O card inserted

The NMI, IREQ0, and IREQ3 pins are not used in the reference platform; these pins can be used by the user from outside the reference platform through the expansion bus connector on the board.

3.5 SH7707 Register Initial Values in Reference Platform

Table 3.2 shows the initial values of the SH7707 registers in the reference platform.

Table 3.2 Register Initial Values

Module	Register	Size	Address	Initial Value in SH7707	Initial Value in Reference Platform
CCN	PTEH	32	H'FFFFFFF0	Undefined	←
CCN	PTEL	32	H'FFFFFFF4	Undefined	←
CCN	TTB	32	H'FFFFFFF8	Undefined	←
CCN	TEA	32	H'FFFFFFFC	Undefined	←
CCN	MMUCR	32	H'FFFFFFE0	H'00000000	←
CCN	BASRA	32	H'FFFFFFE4	Undefined	←
CCN	BASRB	32	H'FFFFFFE8	Undefined	←
CCN	CCR	32	H'FFFFFFEC	H'00000000	H'00000000
CCN	TRA	32	H'FFFFFFD0	Undefined	←
CCN	EXPEVT	32	H'FFFFFFD4	H'00000000	←
CCN	INTEVT	32	H'FFFFFFD8	Undefined	←
UBC	BARA	32	H'FFFFFFB0	Undefined	←
UBC	BAMRA	8	H'FFFFFFB4	Undefined	←
UBC	BBRA	16	H'FFFFFFB8	H'0000	←
UBC	BARB	32	H'FFFFFFA0	Undefined	←
UBC	BAMRB	8	H'FFFFFFA4	Undefined	←
UBC	BBRB	16	H'FFFFFFA8	H'0000	←
UBC	BDRB	32	H'FFFFFF90	Undefined	←
UBC	BDMRB	32	H'FFFFFF94	Undefined	←
UBC	BRCR	16	H'FFFFFF98	H'0000	←
CPG	FRQCR	16	H'FFFFFF80	H'0102	H'0112
CPG	STBCR	8	H'FFFFFF82	H'00	←
CPG	STBCR2	8	H'FFFFFF88	H'00	←
CPG	STBCR3	8	H'FFFFFF8A	H'00	←
CPG	WTCNT	8/16	H'FFFFFF84	H'00	H'A500
CPG	WTCSR	8/16	H'FFFFFF86	H'00	H'A500

Table 3.2 Register Initial Values (cont)

Module	Register	Size	Address	Initial Value in SH7707	Initial Value in Reference Platform
BCN	BCR1	16	H'FFFFFF60	H'0000	H'1011
BCN	BCR2	16	H'FFFFFF62	H'3FF0	H'2EF0
BCN	WCR1	16	H'FFFFFF64	H'3FF3	H'3F71
BCN	WCR2	16	H'FFFFFF66	H'FFFF	H'FFB9
BCN	MCR	16	H'FFFFFF68	H'0000	H'0034
BCN	DCR	16	H'FFFFFF6A	H'0000	←
BCN	PCR	16	H'FFFFFF6C	H'0000	H'FFFF
BCN	RTCSR	16	H'FFFFFF6E	H'0000	H'A518
BCN	RTCNT	16	H'FFFFFF70	H'0000	H'A500
BCN	RTCOR	16	H'FFFFFF72	H'0000	H'A507
BCN	RFCR	16	H'FFFFFF74	H'0000	←
BCN	BCR3	16	H'FFFFFF7E	H'0000	←
RTC	R64CNT	8	H'FFFFFEC0	Undefined	←
RTC	RSECCNT	8	H'FFFFFEC2	Undefined	←
RTC	RMINCNT	8	H'FFFFFEC4	Undefined	←
RTC	RHRCNT	8	H'FFFFFEC6	Undefined	←
RTC	RWKCNT	8	H'FFFFFEC8	Undefined	←
RTC	RDAYCNT	8	H'FFFFFECA	Undefined	←
RTC	RMONCNT	8	H'FFFFFECC	Undefined	←
RTC	RYRCNT	8	H'FFFFFECE	Undefined	←
RTC	RSECAR	8	H'FFFFFED0	Undefined	←
RTC	RMINAR	8	H'FFFFFED2	Undefined	←
RTC	RHRAR	8	H'FFFFFED4	Undefined	←
RTC	RWKAR	8	H'FFFFFED6	Undefined	H'00
RTC	RDAYAR	8	H'FFFFFED8	Undefined	←
RTC	RMONAR	8	H'FFFFFEDA	Undefined	←
RTC	RCR1	8	H'FFFFFEDC	H'00	←
RTC	RCR2	8	H'FFFFFEDE	H'09	←
INTC	ICR0	16	H'FFFFFEE0	H'8000/H'0000	←
INTC	IPRA	16	H'FFFFFEE2	H'0000	←
INTC	IPRB	16	H'FFFFFEE4	H'0000	←

Table 3.2 Register Initial Values (cont)

Module	Register	Size	Address	Initial Value in SH7707	Initial Value in Reference Platform
TMU	TOCR	8	H'FFFFFE90	H'00	←
TMU	TSTR	8	H'FFFFFE92	H'00	←
TMU	TCOR0	32	H'FFFFFE94	H'FFFFFFFF	←
TMU	TCNT0	32	H'FFFFFE98	H'FFFFFFFF	←
TMU	TCR0	16	H'FFFFFE9C	H'0000	←
TMU	TCOR1	32	H'FFFFFEA0	H'FFFFFFFF	←
TMU	TCNT1	32	H'FFFFFEA4	H'FFFFFFFF	←
TMU	TCR1	16	H'FFFFFEA8	H'0000	←
TMU	TCOR2	32	H'FFFFFEAC	H'FFFFFFFF	←
TMU	TCNT2	32	H'FFFFFEB0	H'FFFFFFFF	←
TMU	TCR2	16	H'FFFFFEB4	H'0000	←
TMU	TCPR2	32	H'FFFFFEB8	Undefined	←
SCI	SCSMR	8	H'FFFFFE80	H'00	←
SCI	SCBRR	8	H'FFFFFE82	H'FF	←
SCI	SCSCR	8	H'FFFFFE84	H'00	←
SCI	SCTDR	8	H'FFFFFE86	H'FF	←
SCI	SCSSR	8	H'FFFFFE88	H'84	←
SCI	SCRDR	8	H'FFFFFE8A	H'00	←
SCI	SCSPTR	8	H'FFFFFF7C	H'00	←
INT	INTEVT2	32	H'04000000	Undefined	←
INT	IRR0	8	H'04000004	H'00	←
INT	IRR1	8	H'04000006	H'00	←
INT	IRR2	8	H'04000008	H'00	←
INT	IRR3	8	H'0400000A	H'00	←
INT	IRR4	8	H'0400000C	H'00	←
INT	ICR1	16	H'04000010	H'0000	H'8000
INT	ICR2	16	H'04000012	H'0000	H'0000
INT	PINTER	16	H'04000014	H'0000	H'0000
INT	IPRC	16	H'04000016	H'0000	H'0280
INT	IPRD	16	H'04000018	H'0000	H'0017
INT	IPRE	16	H'0400001A	H'0000	H'9456
INT	IPRF	16	H'0400001C	H'0000	H'0030

Table 3.2 Register Initial Values (cont)

Module	Register	Size	Address	Initial Value in SH7707	Initial Value in Reference Platform
DMAC	SAR0	32	H'04000020	Undefined	H'AC002000
DMAC	DAR0	32	H'04000024	Undefined	H'A40000A0
DMAC	DMATCR0	32	H'04000028	Undefined	H'00000800
DMAC	CHCR0	32	H'0400002C	H'00000000	H'00001F04
DMAC	SAR1	32	H'04000030	Undefined	←
DMAC	DAR1	32	H'04000034	Undefined	←
DMAC	DMATCR1	32	H'04000038	Undefined	←
DMAC	CHCR1	32	H'0400003C	H'00000000	←
DMAC	SAR2	32	H'04000040	Undefined	←
DMAC	DAR2	32	H'04000044	Undefined	←
DMAC	DMATCR2	32	H'04000048	Undefined	←
DMAC	CHCR2	32	H'0400004C	H'00000000	←
DMAC	SAR3	32	H'04000050	Undefined	←
DMAC	DAR3	32	H'04000054	Undefined	←
DMAC	DMATCR3	32	H'04000058	Undefined	←
DMAC	CHCR3	32	H'0400005C	H'00000000	←
DMAC	DMAOR	16	H'04000060	H'0000	H'0001
A/D	ADDRAH	8	H'04000080	H'00	←
A/D	ADDRAL	8	H'04000082	H'00	←
A/D	ADDRBH	8	H'04000084	H'00	←
A/D	ADDRBL	8	H'04000086	H'00	←
A/D	ADDRCH	8	H'04000088	H'00	←
A/D	ADDRCL	8	H'0400008A	H'00	←
A/D	ADDRDH	8	H'0400008C	H'00	←
A/D	ADDRDL	8	H'0400008E	H'00	←
A/D	ADCSR	8	H'04000090	H'00	←
A/D	ADCR	8	H'04000092	H'3F	←
D/A	DADR0	8	H'040000A0	H'00	←
D/A	DADR1	8	H'040000A2	H'00	←
D/A	DACR	8	H'040000A4	H'1F	H'5F

Table 3.2 Register Initial Values (cont)

Module	Register	Size	Address	Initial Value in SH7707	Initial Value in Reference Platform
PCMCIA	PCC0ISR0	8	H'040000E0	Undefined	←
PCMCIA	PCC0GCR	8	H'040000E2	H'00	H'88
PCMCIA	PCC0CSCR	8	H'040000E4	H'00	H'00
PCMCIA	PCC0CSCIER	8	H'040000E6	H'00	H'68
PCMCIA	PCC1ISR0	8	H'040000F0	Undefined	←
PCMCIA	PCC1GSR	8	H'040000F2	H'00	←
PCMCIA	PCC1CSCR	8	H'040000F4	H'00	←
PCMCIA	PCC1CSCIER	8	H'040000F6	H'00	←
PORT	PACR	16	H'04000100	H'0000	←
PORT	PBCR	16	H'04000102	H'0000	←
PORT	PCCR	16	H'04000104	H'AAAA	H'0010
PORT	PDCR	16	H'04000106	H'AAAA	H'A801
PORT	PECR	16	H'04000108	H'AAAA	H'0205
PORT	PFCR	16	H'0400010A	H'AAAA	←
PORT	PGCR	16	H'0400010C	H'AAAA	H'0000
PORT	PHCR	16	H'0400010E	H'AAAA	H'88A2
PORT	PJCR	16	H'04000110	H'0000	H'5004
PORT	PKCR	16	H'04000112	H'0000	←
PORT	PLCR	16	H'04000114	H'0000	←
PORT	SCPCR	16	H'04000116	H'A888	H'1888
PORT	PADR	8	H'04000120	H'00	←
PORT	PBDR	8	H'04000122	H'00	←
PORT	PCDR	8	H'04000124	H'00	H'00
PORT	PDDR	8	H'04000126	H'00	H'00
PORT	PEDR	8	H'04000128	H'00	H'02
PORT	PFDR	8	H'0400012A	H'00	←
PORT	PGDR	8	H'0400012C	H'00	←
PORT	PHDR	8	H'0400012E	H'00	←
PORT	PJDR	8	H'04000130	H'00	H'42
PORT	PKDR	8	H'04000132	H'00	←
PORT	PLDR	8	H'04000134	H'00	←
PORT	SCPDR	8	H'04000136	H'00	←

Table 3.2 Register Initial Values (cont)

Module	Register	Size	Address	Initial Value in SH7707	Initial Value in Reference Platform
SCIF	SCSMR2	8	H'04000150	H'00	←
SCIF	SCBRR2	8	H'04000152	H'FF	←
SCIF	SCSCR2	8	H'04000154	H'00	←
SCIF	SCFTDR2	8	H'04000156	-	←
SCIF	SCSSR2	16	H'04000158	H'0060	←
SCIF	SCFRDR2	8	H'0400015A	-	←
SCIF	SCSFCR2	8	H'0400015C	H'00	←
SCIF	SCFDR2	16	H'0400015E	H'0000	←
IrDA	SCSMR1	8	H'04000140	H'00	H'80
IrDA	SCBRR1	8	H'04000142	H'FF	←
IrDA	SCSCR1	8	H'04000144	H'00	H'00
IrDA	SCFTDR1	8	H'04000146	-	←
IrDA	SCSSR1	16	H'04000148	H'0060	←
IrDA	SCFRDR1	8	H'0400014A	-	←
IrDA	SCSFCR1	8	H'0400014C	H'00	←
IrDA	SCFDR1	16	H'0400014E	H'0000	←
LCD	LCDAR	16	H'040000C0	H'0000	←
LCD	LCDDR	16	H'040000C2	H'0000	*[1]
LCD	LCDPR	16	H'040000C6	H'0000	*[2]
LCD	LCDDMR	16	H'040000CE	H'0000	*[3]

Notes: 1.

Register	Size	Address (LCDAR)	Address (LCDDR)	Initial Value in Reference Platform
LCDDR1	16	H'0001	H'040000C2	H'0230
LCDDR2	16	H'0002	H'040000C2	H'7779
LCDDR3	16	H'0003	H'040000C2	H'0177
LCDDR4	16	H'0004	H'040000C2	H'013F
LCDDR5	16	H'0005	H'040000C2	H'013F
LCDDR6	16	H'0006	H'040000C2	H'E000

2.

Register	Size	Address (LCDAR)	Address (LCDDR)	Initial Value in Reference Platform
LCDPR0	16	H'0000	H'040000C6	H'000F
LCDPR1	16	H'0001	H'040000C6	H'000A
LCDPR2	16	H'0002	H'040000C6	H'0005
LCDPR3	16	H'0003	H'040000C6	H'0000

3.

Register	Size	Address (LCDAR)	Address (LCDDR)	Initial Value in Reference Platform
LCDDMR0	16	H'0000	H'040000CE	H'8000
LCDDMR1	16	H'0001	H'040000CE	H'AC00
LCDDMR4	16	H'0004	H'040000CE	H'895F

The registers other than the above will not be initialized.

3.6 SH7707 Pin Assignment

Table 3.3 shows the assignment of pin functions and pin numbers on the 200-pin expansion bus connector in the reference platform.

Table 3.3 Pin Assignment

SH7707 Pin No.	SH7707 Pin Name	Function in Reference Platform	Pin No. on Expansion Bus Connector
1	MD1	Mode pin	196
2	MD2	Mode pin	146
3	V_{CC}	RTC oscillator power supply (3.3 V)	—
4	XTAL2	NC	195
5	EXTAL2	Crystal oscillator pin (on-chip RTC; 32.768 kHz)	145
6	V_{SS} (RTC)	RTC oscillator power supply (0 V)	—
7	NMI	Interrupt from FPGA (must be pulled up)	194
8	IRQ0	Interrupt from FPGA (must be pulled up)	144
9	IRQ1	Interrupt from FPGA (keyboard/touch panel/ sound timer; must be pulled up)	193
10	IRQ2	External interrupt request (SCI; must be pulled up)	143
11	IRQ3	External interrupt request (parallel interface; must be pulled up)	192
12	IRQ4	External interrupt request (touch panel; must be pulled up)	142
13	D31	Data bus 31 (must be pulled up)	191
14	D30	Data bus 30 (must be pulled up)	141
15	D29	Data bus 29 (must be pulled up)	190
16	D28	Data bus 28 (must be pulled up)	140
17	D27	Data bus 27 (must be pulled up)	189
18	D26	Data bus 26 (must be pulled up)	139
19	V_{SS}	GND	—
20	D25	Data bus 25 (must be pulled up)	188
21	V_{CC}	Power supply (3.3 V)	—
22	D24	Data bus 24 (must be pulled up)	138

Table 3.3 Pin Assignment (cont)

SH7707 Pin No.	SH7707 Pin Name	Function in Reference Platform	Pin No. on Expansion Bus Connector
23	D23	Data bus 23 (must be pulled up)	187
24	D22	Data bus 22 (must be pulled up)	137
25	D21	Data bus 21 (must be pulled up)	186
26	D20	Data bus 20 (must be pulled up)	136
27	V_{SS}	GND	—
28	D19	Data bus 19 (must be pulled up)	185
29	V_{CC}	Power supply (3.3 V)	—
30	D18	Data bus 18 (must be pulled up)	135
31	D17	Data bus 17 (must be pulled up)	184
32	D16	Data bus 16 (must be pulled up)	134
33	V_{SS}	GND	—
34	D15	Data bus 15 (must be pulled up)	183
35	V_{CC}	Power supply (3.3 V)	—
36	D14	Data bus 14 (must be pulled up)	133
37	D13	Data bus 13 (must be pulled up)	182
38	D12	Data bus 12 (must be pulled up)	132
39	D11	Data bus 11 (must be pulled up)	181
40	D10	Data bus 10 (must be pulled up)	131
41	D9	Data bus 9 (must be pulled up)	180
42	D8	Data bus 8 (must be pulled up)	130
43	D7	Data bus 7 (must be pulled up)	179
44	D6	Data bus 6 (must be pulled up)	129
45	V_{SS}	GND	—
46	D5	Data bus 5 (must be pulled up)	178
47	V_{CC}	Power supply (3.3 V)	—
48	D4	Data bus 4 (must be pulled up)	128
49	D3	Data bus 3 (must be pulled up)	177
50	D2	Data bus 2 (must be pulled up)	127
51	D1	Data bus 1 (must be pulled up)	176

Table 3.3 Pin Assignment (cont)

SH7707 Pin No.	SH7707 Pin Name	Function in Reference Platform	Pin No. on Expansion Bus Connector
52	D0	Data bus 0 (must be pulled up)	126
53	A0	Address bus 0	175
54	A1	Address bus 1	125
55	A2	Address bus 2	174
56	A3	Address bus 3	124
57	V_{SS}	GND	—
58	A4	Address bus 4	173
59	V_{CC}	Power supply (3.3 V)	—
60	A5	Address bus 5	123
61	A6	Address bus 6	172
62	A7	Address bus 7	122
63	A8	Address bus 8	171
64	A9	Address bus 9	121
65	A10	Address bus 10 (must be pulled up)	170
66	A11	Address bus 11	120
67	A12	Address bus 12	169
68	A13	Address bus 13	119
69	V_{SS}	GND	—
70	A14	Address bus 14	168
71	V_{CC}	Power supply (3.3 V)	—
72	A15	Address bus 15	118
73	A16	Address bus 16	167
74	A17	Address bus 17	117
75	A18	Address bus 18	166
76	A19	Address bus 19	116
77	A20	Address bus 20	165
78	A21	Address bus 21	115
79	V_{SS}	GND	—
80	A22	Address bus 22	164

Table 3.3 Pin Assignment (cont)

SH7707 Pin No.	SH7707 Pin Name	Function in Reference Platform	Pin No. on Expansion Bus Connector
81	V_{CC}	Power supply (3.3 V)	—
82	A23	Address bus 23	114
83	V_{SS}	GND	—
84	A24	Address bus 24	163
85	V_{CC}	Power supply (3.3 V)	—
86	A25	Address bus 25	113
87	BS	SHBS of FPGA	162
88	RD	Read strobe	112
89	WE0	D7—D0 select signal	161
90	WE1/WE	D15—D8 select signal/PCMCIA WE signal	111
91	WE2/ICIORD	D23—D16 select signal/PCMCIA IORD signal	160
92	WE3/ICIOWR	D31—D24 select signal/PCMCIA IOWR signal	110
93	RDWR	Read/write switching signal	159
94	PCCRD/WR/ PTE7	PCMCIA read/write switching signal (must be pulled up)	109
95	V_{SS}	GND	—
96	CS0	Chip selection 0 (flash memory)	158
97	V_{CC}	Power supply (3.3 V)	—
98	CS2	Chip selection 2 (FPGA)	108
99	CS3	Chip selection 3 (FPGA)	157
100	CS4	Chip selection 4 (FPGA)	107
101	CE1A	Not used (must be pulled up)	156
102	CE1B	PCMCIA CE1B	106
103	CE2A	Not used (must be pulled up)	155
104	CE2B	PCMCIA CE2B	105
105	PTK5	Flash/RDY/BSY signal (must be pulled up)	57
106	RAS	RAS for area 3	56
107	PTJ1	I/O port (touch panel; must be pulled up)	8
108	CASLL	D7—D0 selection CAS (DRAM)	58

Table 3.3 Pin Assignment (cont)

SH7707 Pin No.	SH7707 Pin Name	Function in Reference Platform	Pin No. on Expansion Bus Connector
109	V_{SS}	GND	—
110	CASLH	D15—D8 selection CAS (DRAM)	9
111	V_{CC}	Power supply (3.3 V)	—
112	CASHL	D23—D16 selection CAS (DRAM)	59
113	CASHH	D31—D24 selection CAS (DRAM)	10
114	PTD5	I/O port (FPGA)	60
115	PTD7	I/O port (FPGA)	11
116	PCCREG	PCMCIA REG	61
117	PCC0DRV	PCMCIA0 buffer control	12
118	PCC0RESET	PCMCIA0 reset output	62
119	PTE1	I/O port (PCMCIA; must be pulled up)	13
120	PTE0	I/O port (PCMCIA; must be pulled up)	63
121	BACK	Not used	14
122	BREQ	Not used (must be pulled up)	64
123	WAIT	Not used (must be pulled up)	15
124	PCC0WAIT	PCMCIA0 hardware wait request	65
125	PTH5	I/O port (FPGA; must be pulled up)	16
126	PCC0WP	PCMCIA0 WP	66
127	PCC0READY	PCMCIA0 BUSY.READ	17
128	PCC0BVD1	PCMCIA0 BVD1	67
129	PCC0BVD2	PCMCIA0 BVD2	18
130	PCC0CD1	PCMCIA0 CD1	68
131	PCC0CD2	PCMCIA0 CD2	19
132	V_{SS}	GND	—
133	PCC0VS1	PCMCIA0 VS1	69
134	V_{CC}	Power supply (3.3 V)	—
135	PCC0VS2	PCMCIA0 VS2	20
136	PTF7	Keyboard input 7 (FPGA)	70
137	PTF6	Keyboard input 6 (FPGA)	21

Table 3.3 Pin Assignment (cont)

SH7707 Pin No.	SH7707 Pin Name	Function in Reference Platform	Pin No. on Expansion Bus Connector
138	PTF5	Keyboard input 5 (FPGA)	71
139	PTF4	Keyboard input 4 (FPGA)	22
140	PTF3	Keyboard input 3 (FPGA)	72
141	PTF2	Keyboard input 2 (FPGA)	23
142	PTF1	Keyboard input 1 (FPGA)	73
143	PTF0	Keyboard input 0 (FPGA)	24
144	MD0	Mode pin	74
145	V_{CC} (PLL1)	Power supply for PLL (3.3 V)	—
146	CAP1	470 pF	—
147	V_{SS} (PLL1)	Power supply for PLL (0V)	—
148	V_{SS} (PLL2)	Power supply for PLL (0V)	—
149	CAP2	470 pF	—
150	V_{CC} (PLL2/3)	Power supply for PLL (3.3 V)	—
151	V_{SS} (PLL2/3)	Power supply for PLL (0V)	—
152	V_{SS} (PLL3)	Power supply for PLL (0V)	—
153	V_{SS}	GND	—
154	V_{CC}	Power supply (3.3 V)	—
155	XTAL	Not used	25
156	EXTAL	External clock (8 MHz is currently input)	75
157	PTJ6	I/O port (touch panel; must be pulled up)	26
158	PTJ7	I/O port (touch panel; must be pulled down)	76
159	TCLK	Not used (must be pulled up)	27
160	IRQOUT	Not used	77
161	V_{SS}	GND	—
162	CKIO	Internal clock output	28
163	V_{CC}	Power supply (3.3 V)	—
164	TxD0	Not used	78
165	SCK0	Not used (must be pulled up)	29
166	TxD1	Data output from serial port 1 (IrDA)	79

Table 3.3 Pin Assignment (cont)

SH7707 Pin No.	SH7707 Pin Name	Function in Reference Platform	Pin No. on Expansion Bus Connector
167	SCK1	Not used (must be pulled up)	30
168	TxD2	Data output from serial port 2 (SCIF)	80
169	SCK2	Not used (must be pulled up)	31
170	RTS2	Transmit request for serial port 2 (SCIF)	81
171	RxD0	Not used (must be pulled up)	32
172	RxD1	Serial port 1 data input (IrDA; must be pulled up)	82
173	V_{SS}	GND	—
174	RxD2	Data input from serial port 2 (SCIF)	33
175	V_{CC}	Power supply (3.3 V)	—
176	CT2	Transfer acknowledge for serial port 2 (SCIF)	83
177	UD3	LCD data output	34
178	UD2	LCD data output	84
179	UD1	LCD data output	35
180	UD0	LCD data output	85
181	V_{SS}	GND	—
182	CL1	LCD clock output (LCD, FPGA; must be pulled up)	36
183	V_{CC}	Power supply (3.3 V)	—
184	CL2	LCD clock output	86
185	LD3	Not used (must be pulled up)	37
186	PTC2	I/O port (PCMCIA; must be pulled up)	87
187	LD1	Not used (must be pulled up)	38
188	LD0	LCD data output (must be pulled up)	88
189	FLM	LCD control	39
190	PTD0	PCMCIA I/O port (must be pulled up)	89
191	PTD4	Interrupt of audio module (FPGA; must be pulled up)	40
192	PTD6	Interrupt of audio module (FPGA; reserved for future extension)	90

Table 3.3 Pin Assignment (cont)

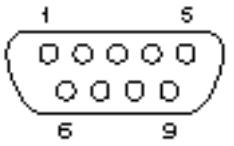
SH7707 Pin No.	SH7707 Pin Name	Function in Reference Platform	Pin No. on Expansion Bus Connector
193	RESET	Reset	41
194	V_{CC}	Power supply (3.3 V)	—
195	MD3	Mode pin	91
196	MD4	Mode pin	42
197	MD5	Mode pin	92
198	AV_{SS}	GND	—
199	AN0	Analog input (touch panel)	43
200	AN1	Analog input (touch panel)	93
201	AN2	Not used (must be pulled up)	44
202	AN3	Not used (must be pulled up)	94
203	AN4	Not used (must be pulled up)	45
204	AN5	Not used (must be pulled up)	95
205	AV_{CC}	Analog input (3.3 V)	—
206	AN6	Not used (must be pulled up)	46
207	DA0	Analog I/O (sound)	96
208	AV_{SS}	GND	—

Note: The SH7707 pins (208 pins) except for the power-supply, ground, and CAP pins are connected to the expansion bus connector.

3.7 Connector Pin Arrangement

The following describes the pin arrangement of the connectors on the reference platform.

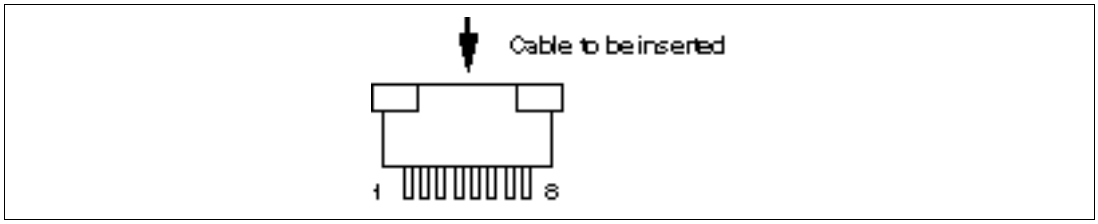
3.7.1 Serial Connector



Note: Use a straight-type serial cable.

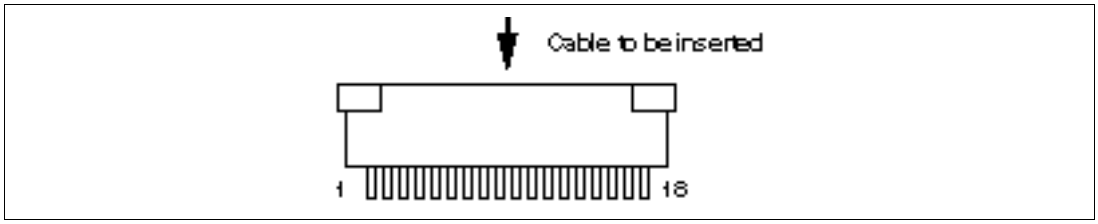
Pin No.	Signal Name
1	CD
2	RD
3	TD
4	DTR
5	GND
6	DSR
7	RTS
8	CTS
9	NC

3.7.2 Touch Panel Connector



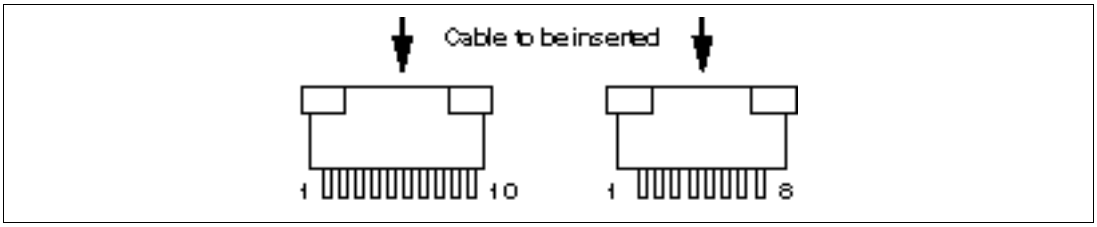
Pin No.	Signal Name
1	X2
2	NC
3	Y1
4	NC
5	X1
6	NC
7	Y2
8	NC

3.7.3 LCD Connector



Pin No.	Signal Name	Pin No.	Signal Name
1	V4	10	3.3V
2	V1	11	CP2
3	25.7V	12	V3
4	3.3V	13	V2
5	S	14	D3
6	GND	15	D2
7	CP1	16	D1
8	GND	17	D0
9	M	18	NC

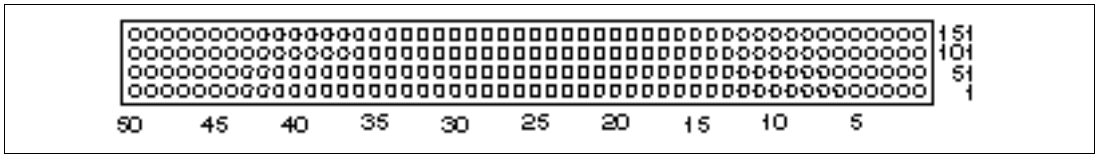
3.7.4 Keyboard Connectors



Pin No.	Signal Name
1	KEYIN0
2	KEYIN1
3	KEYIN2
4	KEYIN3
5	KEYIN4
6	KEYIN5
7	KEYIN6
8	KEYIN7
9	KEYIN8
10	KEYIN9

Pin No.	Signal Name
1	KEYOUT0
2	KEYOUT1
3	KEYOUT2
4	KEYOUT3
5	KEYOUT4
6	KEYOUT5
7	KEYOUT6
8	KEYOUT7

3.7.5 Expansion Bus Connector



Pin No.	Pin Name	SH Pin No.	Pin No.	Pin Name	SH Pin No.
1	3.3V	—	21	PTF6	137
2	GND	—	22	PTF4	139
3	25.7V	—	23	PTF2	141
4	12V	—	24	PTF0	143
5	—	—	25	XTAL (not used)	155
6	—	—	26	PTJ6	157
7	—	—	27	TCLK (not used)	159
8	PTJ1	107	28	CKIO	162
9	CASLH	110	29	SCK0 (not used)	165
10	CASHH	113	30	SCK1 (not used)	167
11	PTD7	115	31	SCK2 (not used)	169
12	PCC0DRV	117	32	RxD0 (not used)	171
13	PTE1	119	33	RxD2	174
14	BACK (not used)	121	34	UD3	177
15	WAIT (not used)	123	35	UD1	179
16	PTH5	125	36	CL1	182
17	PCC0READY	127	37	LD3 (not used)	185
18	PCC0BVD2	129	38	LD1 (not used)	187
19	PCC0CD2	131	39	FLM	189
20	PCC0VS2	135	40	PTD4	191

Pin No.	Pin Name	SH Pin No.	Pin No.	Pin Name	SH Pin No.
41	RESET	193	74	MD0	144
42	MD4	196	75	EXTAL	156
43	AN0	199	76	PTJ7	158
44	AN2 (not used)	201	77	IRQOUT (not used)	160
45	AN4 (not used)	203	78	TxD0 (not used)	164
46	AN6 (not used)	206	79	TxD1	166
47	—	—	80	TxD2	168
48	5V	—	81	RTS2	170
49	3.3V	—	82	RxD1	172
50	GND	—	83	CTS2	176
51	3.3V	—	84	UD2	178
52	GND	—	85	UD0	180
53	25.7V	—	86	CL2	184
54	12V	—	87	PTC2	186
55	—	—	88	LD0	188
56	RAS	106	89	PTD0	190
57	PTK5	105	90	PTD6	192
58	CASLL	108	91	MD3	195
59	CASHL	112	92	MD5	197
60	PTD5	114	93	AN1	200
61	PCCREG	116	94	AN3 (not used)	202
62	PCC0RESET	118	95	AN5 (not used)	204
63	PTE0	120	96	DA0	207
64	BREQ (not used)	122	97	—	—
65	PCC0WAIT	124	98	5V	—
66	PCC0WP	126	99	3.3V	—
67	PCC0BVD1	128	100	GND	—
68	PCC0CD1	130	101	3.3V	—
69	PCC0VS1	133	102	GND	—
70	PTF7	136	103	25.7V	—
71	PTF5	138	104	12V	—
72	PTF3	140	105	CE2B	104
73	PTF1	142	106	CE1B	102

Pin No.	Pin Name	SH Pin No.	Pin No.	Pin Name	SH Pin No.
107	CS4	100	140	D28	16
108	CS2	98	141	D30	14
109	PCCRD/WR	94	142	IRQ4	12
110	WE3/ICIOWR	92	143	IRQ2	10
111	WE1/WE	90	144	IRQ0	8
112	RD	88	145	EXTAL2	5
113	A25	86	146	MD2	2
114	A23	82	147	—	—
115	A21	78	148	5V	—
116	A19	76	149	3.3V	—
117	A17	74	150	GND	—
118	A15	72	151	3.3V	—
119	A13	68	152	GND	—
120	A11	66	153	25.7V	—
121	A9	64	154	12V	—
122	A7	62	155	CE2A (not used)	103
123	A5	60	156	CE1A (not used)	101
124	A3	56	157	CS3	99
125	A1	54	158	CS0	96
126	D0	52	159	RDWR	93
127	D2	50	160	WE2/ICIORD	91
128	D4	48	161	WE0	89
129	D6	44	162	PTK4	87
130	D8	42	163	A24	84
131	D10	40	164	A22	80
132	D12	38	165	A20	77
133	D14	36	166	A18	75
134	D16	32	167	A16	73
135	D18	30	168	A14	70
136	D20	26	169	A12	67
137	D22	24	170	A10	65
138	D24	22	171	A8	63
139	D26	18	172	A6	61

Pin No.	Pin Name	SH Pin No.	Pin No.	Pin Name	SH Pin No.
173	A4	58	187	D23	23
174	A2	55	188	D25	20
175	A0	53	189	D27	17
176	D1	51	190	D29	15
177	D3	49	191	D31	13
178	D5	46	192	IRQ3	11
179	D7	43	193	IRQ1	9
180	D9	41	194	NMI	7
181	D11	39	195	XTAL2	4
182	D13	37	196	MD1	1
183	D15	34	197	—	—
184	D17	31	198	5V	—
185	D19	28	199	3.3V	—
186	D21	25	200	GND	—

3.8 Reference Platform Start-Up

Figure 3.4 shows the reference platform interfaces.

3.8.1 Starting Microsoft® Windows® CE Operating System

1. Check that the flash memory devices containing Windows® CE are installed in the correct flash memory socket on the bottom of the board. If not installed, install them according to the instructions in section 3.8.3, Installing Flash Memory.
2. Connect the power-supply (25.7 V, 5 V, and 3.3 V) and GND cables to the power-supply and GND terminals on the top of the board.
3. Connect the keyboard.
4. Check that the LCD, touch panel, keyboard, and speaker are correctly connected to the corresponding connectors.
5. Turn on the power, and Windows® CE will start.

3.8.2 Starting Monitor Program

1. Check that the flash memory devices containing the monitor program are installed in the correct flash memory sockets on the bottom of the board. If not installed, install them according to the instructions in section 3.8.3, Installing Flash Memory.
2. Connect the test module to the board.
3. Connect the power-supply (25.7 V, 5 V, and 3.3 V) and GND cables to the power-supply and GND terminals on the top of the board.
4. Connect the RS-232C interface connector on the top of the board to the RS-232C connector of the host computer with the RS-232C cable (straight cable).
5. Start the communication software on the host computer (refer to section 5, Test Specifications, for the communication settings).
6. Turn on the power, and the monitor program will start.

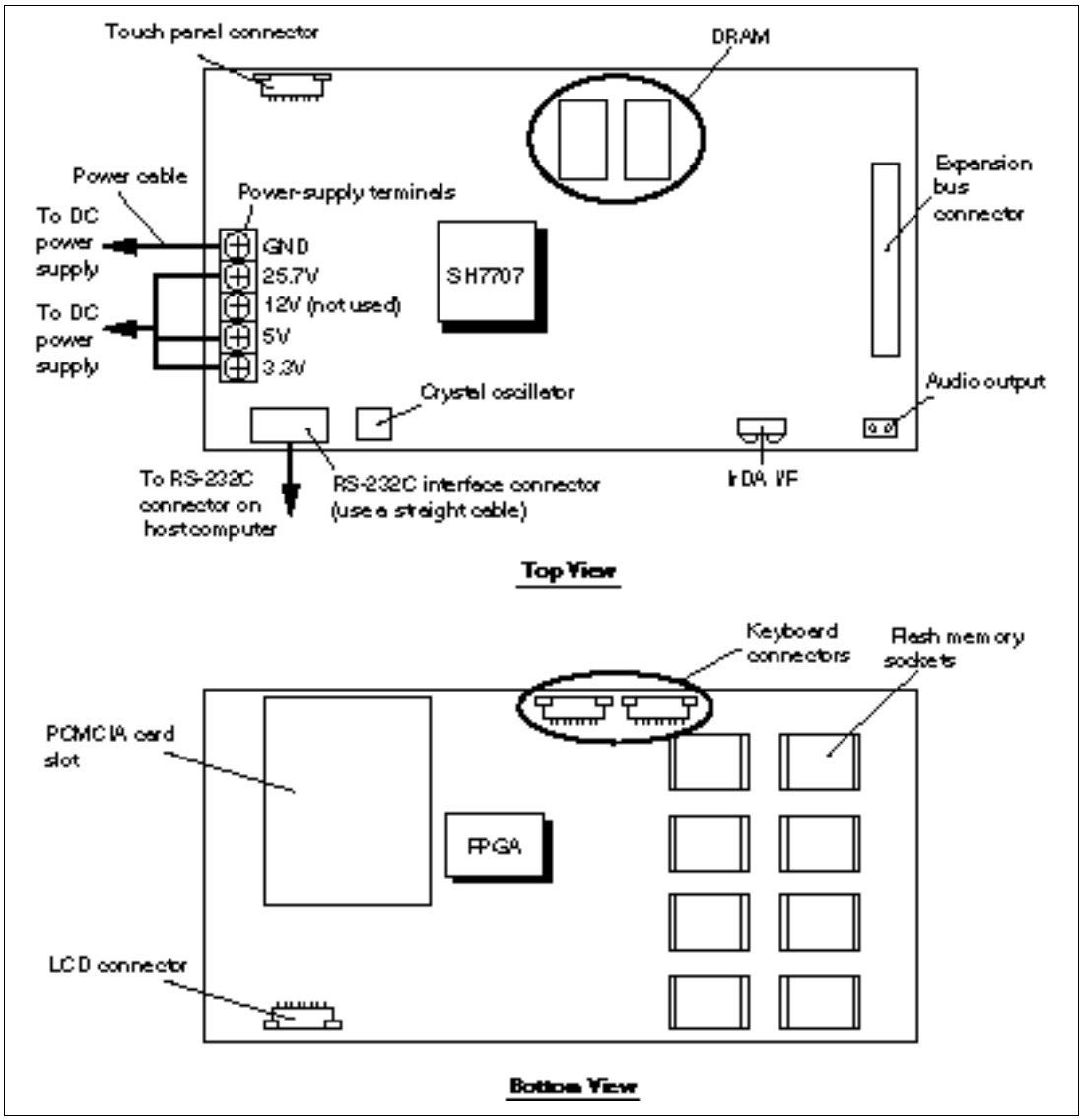


Figure 3.4 Reference Platform Interfaces

3.8.3 Installing Flash Memory

By exchanging flash memory, the debugging method for the reference platform can be changed. Figure 3.5 shows how to install flash memory.

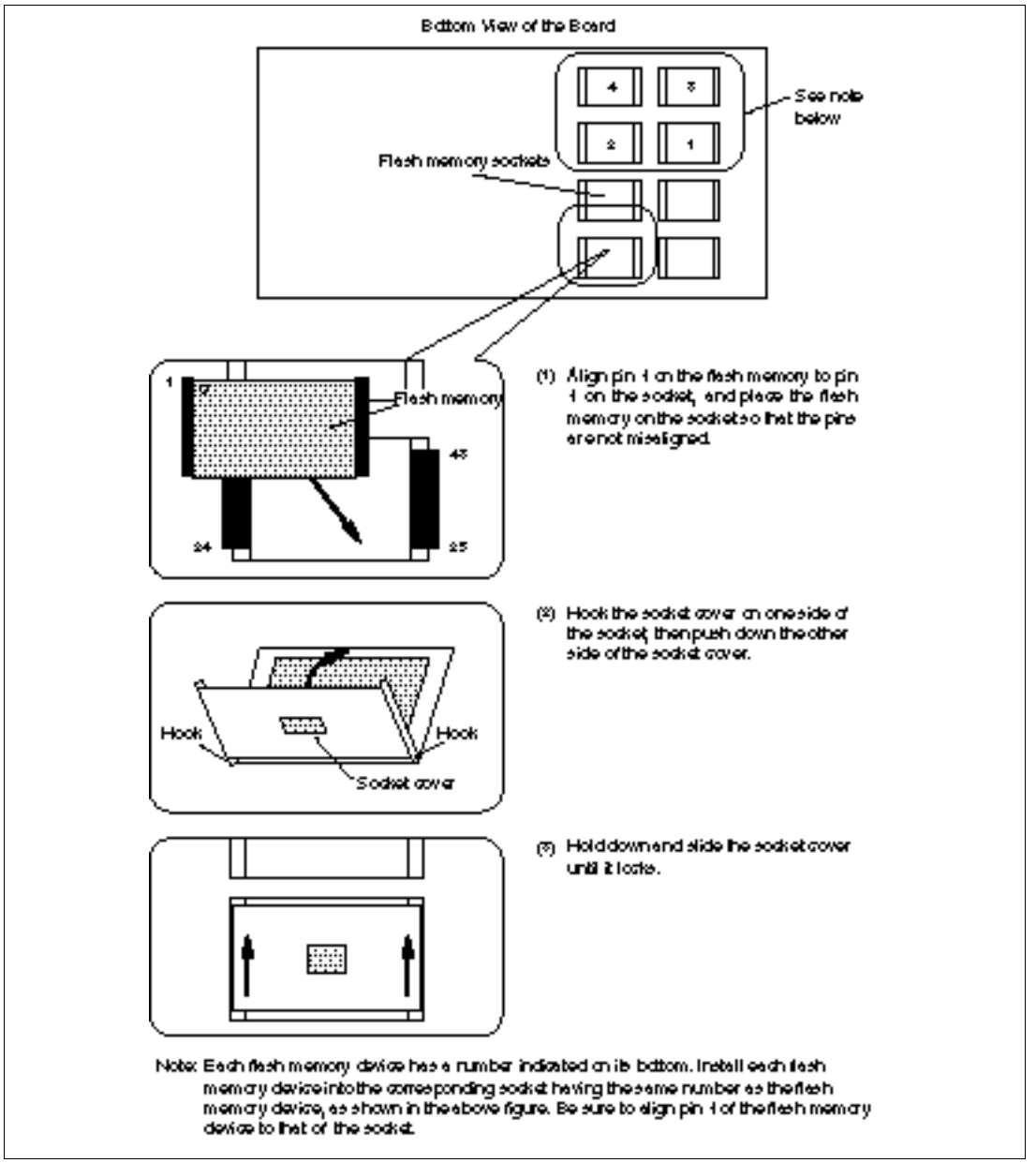


Figure 3.5 Installing Flash Memory

Section 4 Function Specifications

4.1 Memory

The reference platform includes a 16-Mbyte DRAM space (which is two 8-Mbyte DRAMs) and an 8-Mbyte flash memory space (which is eight 1-Mbyte flash memories.)

The following describe the specifications of DRAM and flash memory.

4.1.1 DRAM

The reference platform includes two Hitachi 8-Mbyte DRAMs HM5165165 (4 Mword × 16 bits). For details on access timing, refer to the SH7707 Hardware Manual and HM5164165 Series, HM5165165 Series Data Sheet.

(1) Specifications

The reference platform includes two 8-Mbyte DRAMs and provides a 16-Mbyte DRAM space. Figure 4.1 shows the DRAM connection. Area 3 in the SH7707 is allocated for the DRAM space. In the area, two DRAMs are placed in parallel, enabling data access in a 32-bit bus width. Addresses H'0C000000 to H'0CFFFFFF can be accessed.

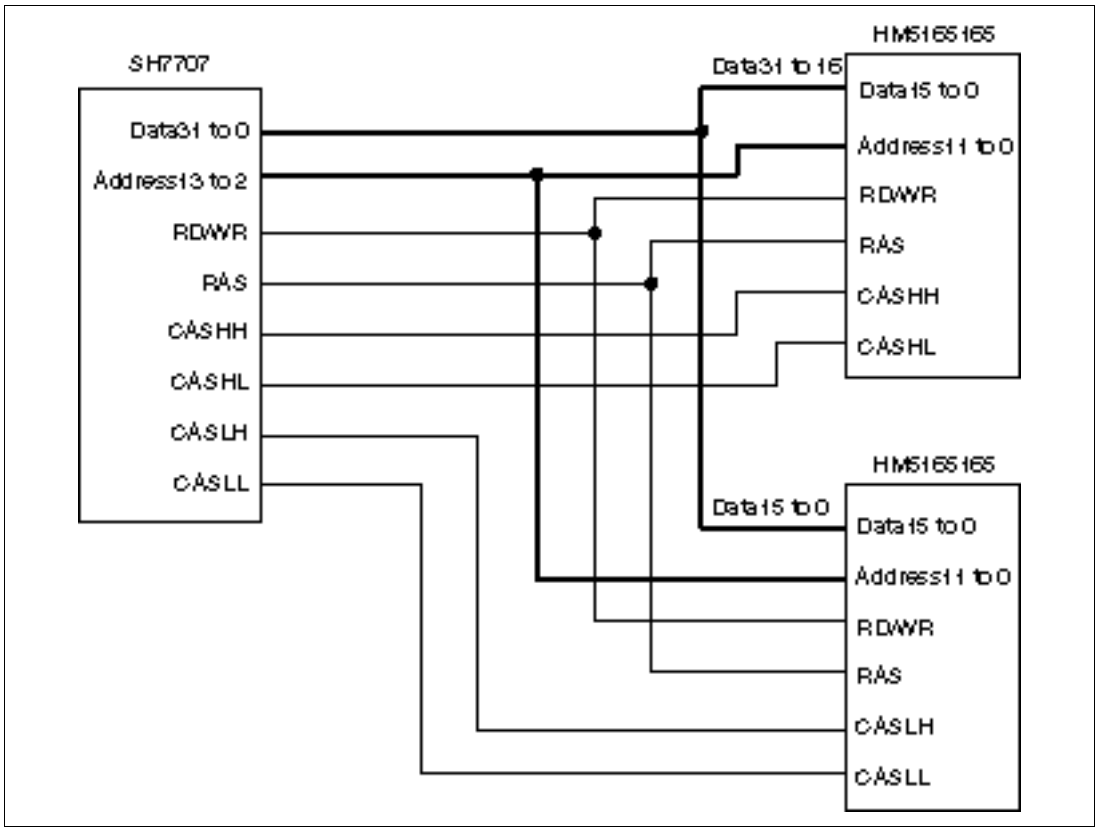


Figure 4.1 DRAM Connection

(2) Wait State Control

The following describe the number of wait states inserted into the access states.

(1) The number of wait cycles inserted to ensure RAS precharge time:
Up to four cycles can be inserted. One cycle is inserted for the reference platform.

(2) The number of wait cycles inserted between RAS assertion and CAS assertion:
Up to four cycles can be inserted. One cycle is inserted for the reference platform.

(3) The number of wait cycles inserted between CAS assertion and the completion of access:
Up to three cycles can be inserted. One cycle is inserted for the reference platform.

The settings above can be made by specifying memory control register (MCR) and wait control register 2 (WCR2).

For details, refer to the SH7707 Hardware Manual and HM5164165 Series, HM5165165 Series Data Sheet.

4.1.2 Timing Chart

(1) DRAM Basic Access Timing

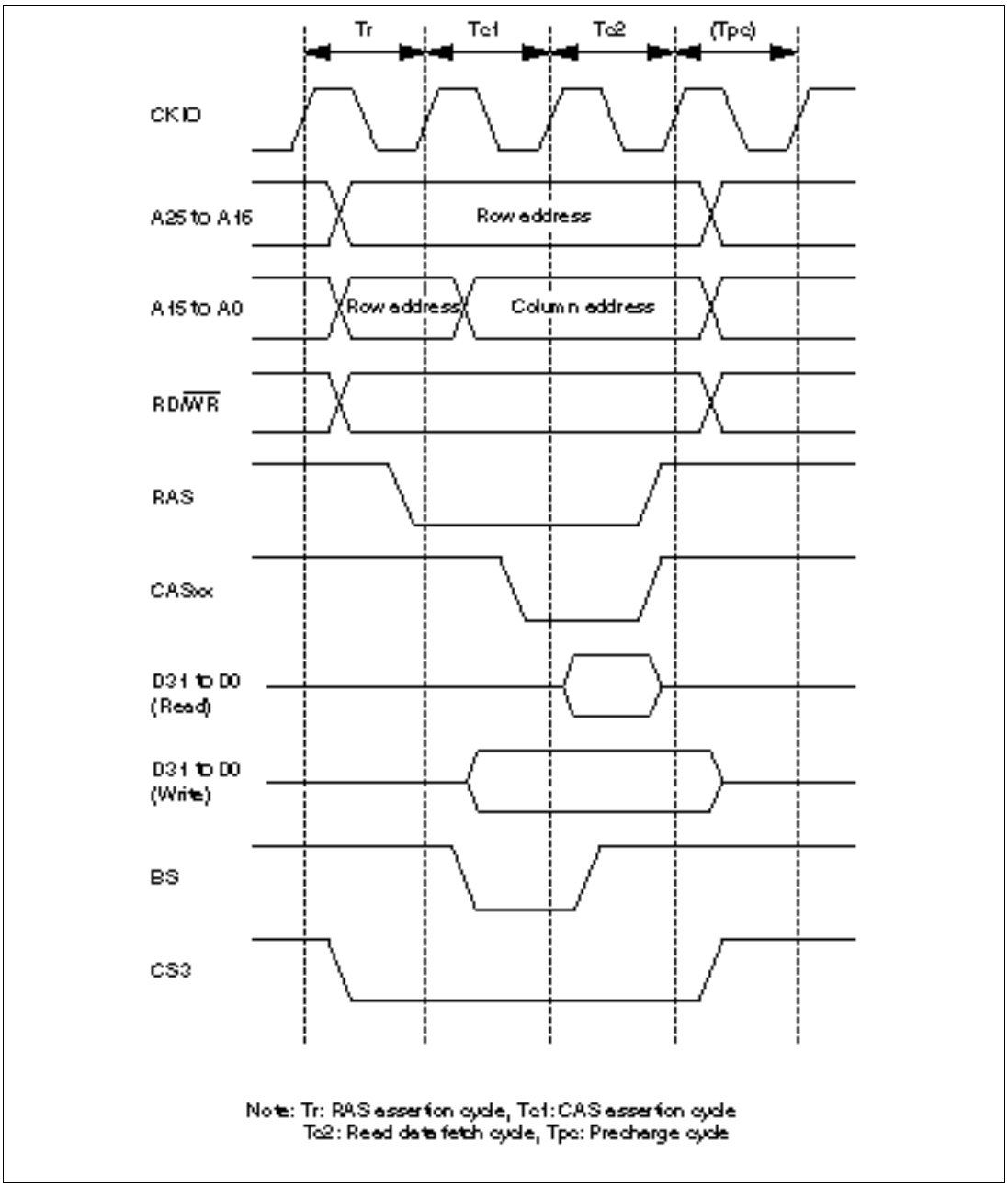


Figure 4.2 DRAM Basic Timing

(2) DRAM Wait State Insertion Timing

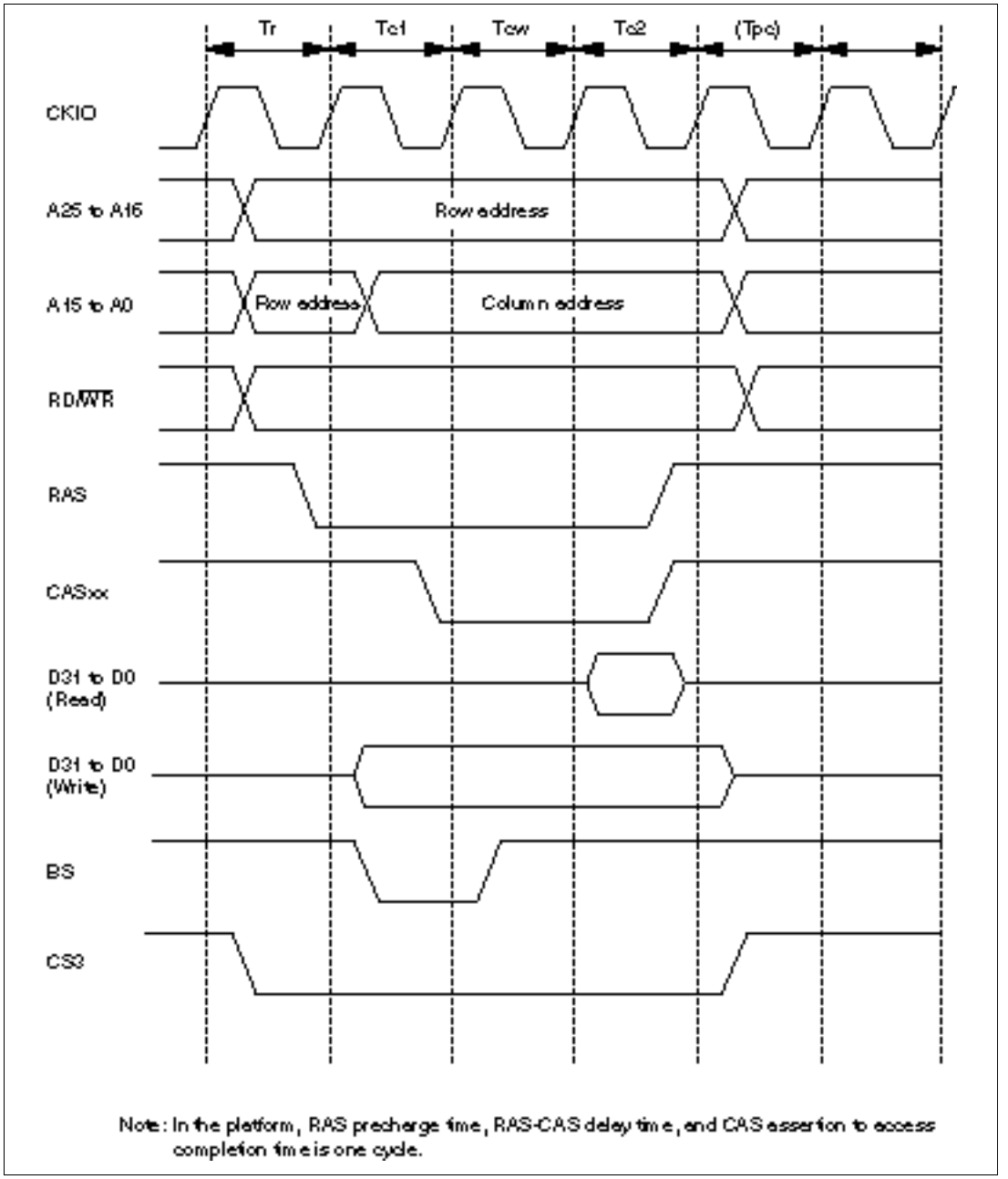


Figure 4.3 DRAM Wait Insertion Timing

4.1.3 Flash Memory

The reference platform includes Hitachi flash memory HN29WT800. This memory is a CMOS flash memory of 1 Mword × 8 bits and enables 3.3-V single power supply operation. This memory also includes automatic programming and erasing functions. Therefore, complex operation is not required for programming and erasing memory. For details, refer to the HM5164165 Series, HM5165165 Series Data Sheet.

(1) Specifications

The reference platform includes eight flash memories and provides an 8-Mbyte flash memory space. Area 0 in the SH7707 is allocated for the flash memory space. In this area, four flash memories are placed in parallel in two columns in series, enabling data access in a 32-bit bus width to a 2 Mword × 32 bit memory space.

(2) Description on Address Decoder Operation

When the SH7707 accesses flash memory, and device switching is required due to address overflow, an external address decoder does the switching. The following describes the operation of the address decoder.

Addresses H'0000000 to H'07FFFFF can be accessed for the 2 Mword × 32 bit flash memory. The address ranges for the two columns of memory in series are H'0000000 to H'03FFFFF and H'0400000 to H'07FFFFF.

Therefore, to select one of the two columns of memories in series, bit 22 in the address bus is used. Signal lines $\overline{CE}$, $\overline{WE}$, and $\overline{OE}$ need to be selected. These signal lines are all low-active. Therefore, by performing a logical OR between each of these signals and bit 22 in the address bus, the first column can be selected when bit 22 is zero, and the second column can be selected when bit 22 is 1. Refer to figure 4.4.

(3) Signal to Confirm the Completion of Programming and Erasing

In order to confirm whether data programming or erasing has been completed for the flash memory, signal RDY/$\overline{Busy}$ is output from the flash memory. This signal is connected to pin number 105 in the SH7707. Therefore, by polling pin number 105, the operation state of the flash memory can be confirmed.

When this signal is in a low-level state, the user cannot access the flash memory because the memory is being programmed or erased. When the signal is in high-level state, the user can access the flash memory.

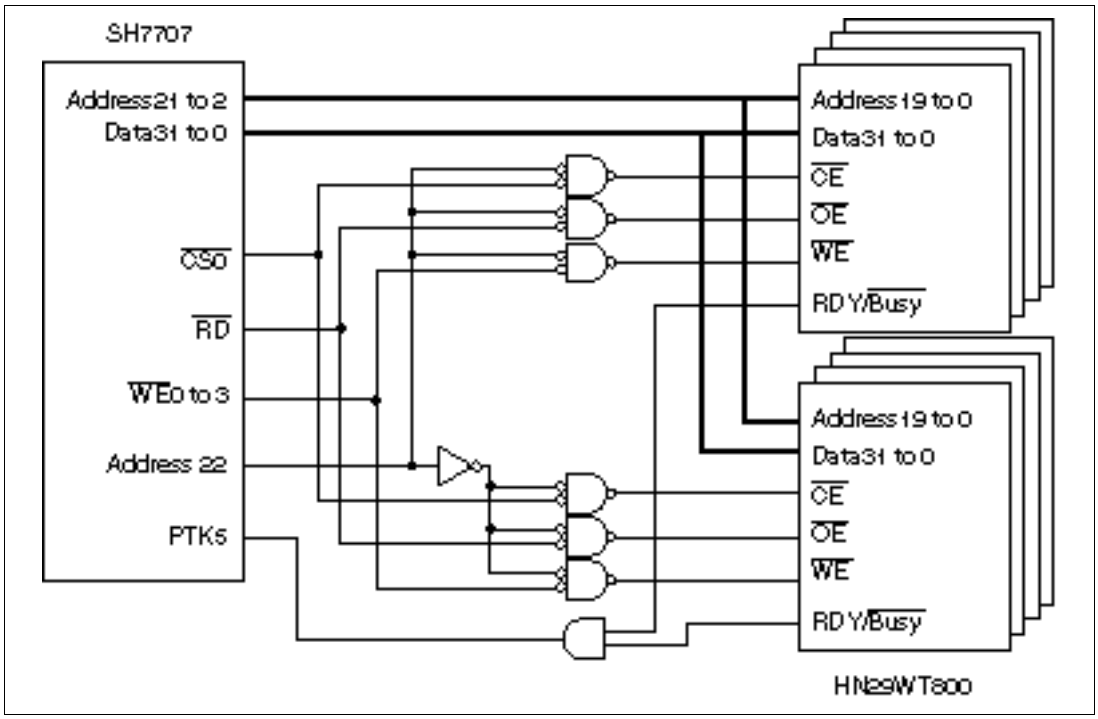


Figure 4.4 Flash Memory Connection

4.2 LCD

The SH7707 includes an LCD controller (LCDC: liquid crystal display controller). The LCDC reads display data from the system memory and displays it on the LCD. The LCDC has two channels of DMA controllers and the reference platform uses channel 0. The LCD used here is an STN type single-screen monochrome 4-level gray scale display, with a maximum resolution of 480×320 dots.

4.2.1 Interface Block Diagram

Figure 4.5 shows the interface block diagram of the SH7707/LCDC, the LCD, and the power supply controller. The LCD can be connected to the SH7707 without any external circuits. The reference platform uses LCD LM48072 manufactured by Sharp Corporation and power supply controller LA5317M manufactured by Sanyo Electric Co., Ltd.

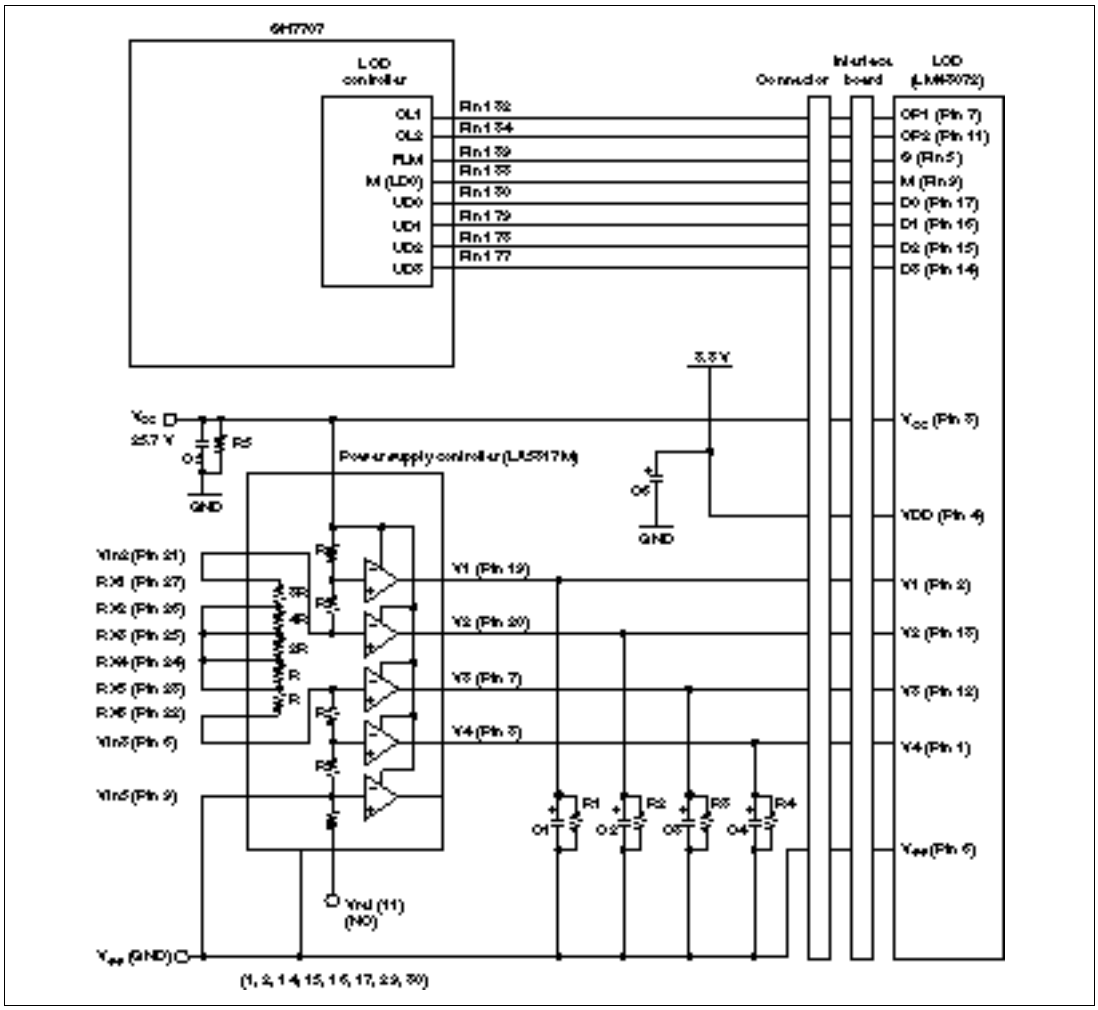


Figure 4.5 Interface Block Diagram

4.2.2 Display Mode

Table 4.1 lists the SH7707/LCDC display mode.

Table 4.1 Display Mode

Mode	Mode Name			Number of Grayscales	Bits/Dot	Resolution	LCDC Operating Frequency
A	STN monochrome	Single screen	4-bit output	4	2	480 × 320 dots	14.7456 MHz

4.2.3 SH7707/LCDC Pin Configuration

Table 4.2 lists the SH7707/LCDC pin configuration.

Table 4.2 LCDC Pin Configuration

Pin Name	Abbreviation	I/O	Function	SH7707 Pin Number
Latch clock of LCD data	CL1	Output	Outputs clock for X-driver to latch one line of display data	182
Shift clock of LCD data	CL2	Output	Outputs shift clock of display data for X-driver	184
First line marker	FLM	Output	Outputs first line mark signal for Y-driver	189
Upper data 3 to 0	UD3 to UD0	Output	Outputs display data D3 to D0.	177 to 180
LCD driving signal alternation	M	Output	Outputs signal for AC conversion of LCD driving signals	188

4.2.4 Register Configuration

Table 4.3 lists the LCDC register configuration.

Table 4.3 LCDC Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Address register	LCDAR	R/W	H'0000	H'040000C0	16 bits
Display control register	LCDDR	R/W	H'0000	H'040000C2	16 bits
Pallet register	LCDPR	R/W	H'0000	H'040000C6	16 bits
DMA control register	LCDDMR	R/W	H'0000	H'040000CE	16 bits

4.2.5 LCD Pin Configuration

Table 4.4 lists the LCD pin configuration.

Table 4.4 LCD Pin Configuration

Pin No.	Abbreviation	Signal Name	Pin No.	Abbreviation	Signal Name
1	V4	LCD voltage	10	DISP	Display control signal
2	V1	LCD voltage	11	CP2	Data shift clock
3	VEE	LCD voltage	12	V3	LCD voltage
4	VDD	Logic circuit voltage	13	V2	LCD voltage
5	S	Start signal for scanning	14	D3	Display data signal
6	VSS	Ground	15	D2	Display data signal
7	CP1	Data latch signal	16	D1	Display data signal
8	VSS	Ground	17	D0	Display data signal
9	M	LCD driving signal alternation	18	NC	No connection

4.2.6 LCDC Interface Timing

Figure 4.6 shows LCD display timing. The LCD used here has the following specifications:

- Monochrome
- Single screen
- 4-bit output (mode A)
- 480 × 320 dots
- Horizontal retrace line interval: 8 dots

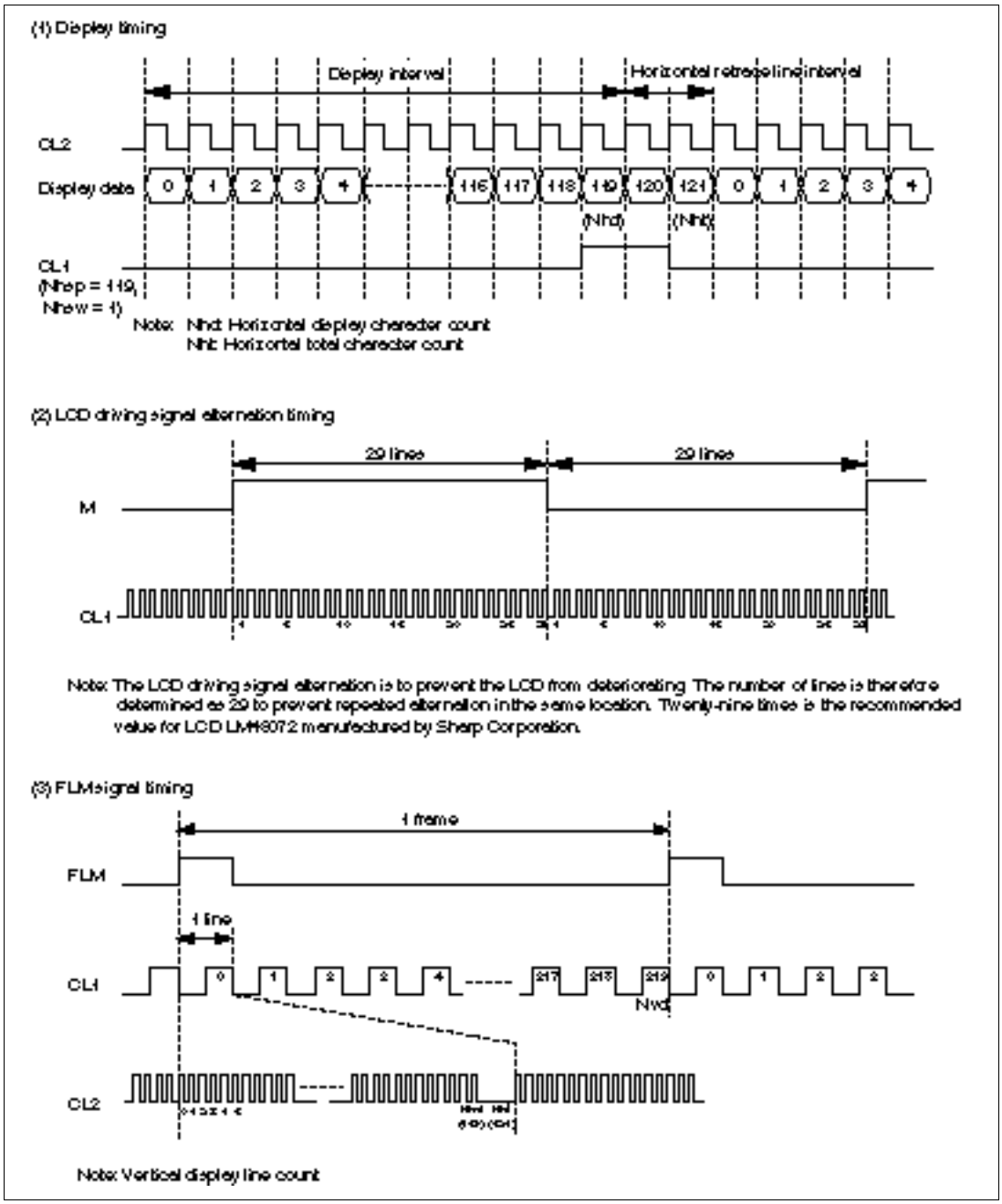


Figure 4.6 Timing Chart

4.2.7 Setting LCDC Register

The LCD used here has the following specifications:

- Monochrome
- Single screen
- 4-bit output (mode A)
- 480 × 320 dots
- Horizontal synchronization position: 8 dots

The LCDC has one address register (LCDAR) and three data registers. a display control register (LCDDR), pallet register (LCDPR), and a DMA control register (LCDDMR). Individual registers within the data registers are indirectly accessed using the address register.

The address of each register is shown below:

Register	Abbreviation	Address
Address register	LCDAR	H'040000C0
Display control register	LCDDR	H'040000C2
Pallet register	LCDPR	H'040000C6
DMA control register	LCDDMR	H'040000CE

Before setting the registers, change the function of the SH7707 pins used for the LCD. The initial state of the pin is set as an input port.

PCCR: H'A4000104 Change H'AAAA to H'0000.

PDCR: H'A4000106 Change H'AAAA to H'0000.

The following describe the register configuration and the value set for the reference platform.

(1) Address register (LCDAR)

The LCDAR is used to indirectly specify one of the LCDC's data registers (display control register, pallet register, and DMA control register). To read or write to a specific data register, the register number to be accessed is first written to the address register, then the corresponding data register is accessed. For example, to set the vertical total line count in display control register 5 (LCDDR5), 5 is first written to the address register. Once a value has been written to the address register, it remains there until a power-on reset is executed or a new value is written. Thus different data registers assigned to the same number can be accessed in succession.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LCDAR:	—	—	—	—	—	—	—	—	—	—	—					
Inital value:	—	—	—	—	—	—	—	—	—	—	—	0	0	0	0	0
R/W:	—	—	—	—	—	—	—	—	—	—	—	R/W	R/W	R/W	R/W	R/W

Bits 15 to 5—Reserved: Writing to these bits are invalid, but 0 must always be written. A read will return 0.

Bit 4—Reserved: This bit can be read or written, but 0 must always be written. Writing 1 has the same effect operationally as writing 0.

Bits 3 to 0—Data Bits: These bits can be read or written. Set a value of 0, 1, 2, 3, 4, 5, 6, or 13 to designate the display control register, a value from 0 to 15 to designate the pallet register, and a value from 0 to 4 to designate the DMA control register.

(2) Setting display control registers (LCDDR)

The LCDDR specifies various LCDC operating modes. The reference platform uses an LCD with a maximum resolution of 480 × 320 dots, and single-screen monochrome.

The registers used are LCDDR1 to LCDDR6 4-grayscale levels.

(a) LCDDR1

This register is configured as follows:

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LCDDR1:	—	—	—	—	—	DIV2	DIV1	DIV0	CC1	CC2	REN	GR1	GR0	LM2	LM1	LM0
Inital value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bits 15 to 11—Reserved

Bits 10 to 8—Clock Division Ratio (DIV2 to DIV0): These bits specify the display clock division ratio with respect to the CKIO clock.

Bit 10	Bit 9	Bit 8	
DIV2	**DIV1**	**DIV0**	**Description**
1	1	1	Illegal settings
1	1	0	
1	0	1	
1	0	0	Dot clock frequency = CKIO × 1/8
0	1	1	Dot clock frequency = CKIO × 1/4
0	1	0	Dot clock frequency = CKIO × 1/2
0	0	1	Dot clock frequency = CKIO × 1
0	0	0	Dot clock frequency = CKIO × 2 (Initial value)

The shaded part shows the setting of the reference platform.

The clock of the platform is divided by 2 (CKIO × 1/2). Therefore, (Div2, Div1, Div0) = (0, 1, 0). For clock ratio, refer to (4), Example of Setting LCD Controller Internal Operating Frequency, in section 4.2.7, Setting LCDC Register.

Bits 7 and 6—Clock Control (CC1 and CC2): These bits specify whether CL1 and CL2 clock is to be output during the retrace line interval.

Bit 7	Bit 6	
CC1	**CC2**	**Description**
1	1	Illegal setting
1	0	CL1 is not output during vertical retrace line interval CL2 is not output during vertical retrace line interval
0	1	CL1 is output during vertical retrace line interval CL2 is output during vertical retrace line interval
0	0	CL1 is output during vertical retrace line interval CL2 is not output during vertical retrace line interval (Initial value)

CL1 is always output because a retrace line interval is not provided in the vertical direction. Eight bits are provided for the horizontal retrace line interval. During the interval, CL2 is not output. Therefore, (CC1, CC0) = (0, 0).

Bit 5—DMA transfer enable (REN): DMA transfer start enable bit is used to display the data held in the frame memory of the LCD panel.

Bit 5		
REN	**Description**	
1	DMA transfer enabled	
0	DMA transfer disabled	(Initial value)

The shaded parts show the settings of the reference platform.

DMA transfer of the display data is started when REN = 1. Transfer requests are not output from the LCDC when REN = 0. We recommend enabling REN after all settings have been made.

Bits 4 and 3—Gradation/Color (GR1 and GR0): These bits specify the number of monochrome gradation or number of colors in the various display modes.

Bit 4	Bit 3		
GR1	**GR0**	**Description**	
1	1	16 gradations (colors) (4 bits/dot)	
1	0	4 gradations (colors) (2 bits/dot)	
0	1	Monochrome (2 colors) (1 bit/dot)	
0	0	Illegal setting	(Initial value)

Since 4 gradations and 2 bits/dot are selected for the reference platform, (GR1, GR0) = (1, 0)

Bits 2 to 0—Display Mode (LM2 to LM0): These bits specify the display mode.

Bit 2	Bit 1	Bit 0		
LM2	**LM1**	**LM0**	**Description**	
1	1	1	Illegal settings	—
1	1	0		
1	0	1	STN color	Mode D
1	0	0	TFT/TFD color	Mode G
0	1	1	Illegal setting	—
0	1	0	Monochrome/dual screen/4 × 2-bit output	Mode C
0	0	1	Monochrome/single screen/8-bit output	Modes B and F
0	0	0	Monochrome/single screen/4-bit output (Initial value)	Mode A and E

The shaded parts show the settings of the reference platform.

Since mode A (monochrome, single screen, 4-bit output) is selected for the reference platform, (LM2, LM1, LM0) = (0, 0, 0).

From the above, LCDDR1 is H'0310 (before transferring data).

When starting data transfer, LCDDR1 will be H'0330.

(b) LCDDR2

This register is configured as follows:

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LCDDR2:	Nhd7	Nhd6	Nhd5	Nhd4	Nhd3	Nhd2	Nhd1	Nhd0	Nht7	Nht6	Nht5	Nht4	Nht3	Nht2	Nht1	Nht0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bits 15 to 8—Horizontal Display Dot Count (Nhd7 to Nhd0): These bits specify the number of horizontal dots in the display.

Bits 7 to 0—Horizontal Total Dot Count (Nht7 to Nht0): These bits specify the total number of horizontal dots including the retrace line interval.

How to calculate Nhd and Nht in a 480 × 320-dot display

Nhd: 480/4 – 1 = 119, which is B'0111 0111, which is H'77.

Nht: (480 + 8)/4 – 1 = 121, which is B'0111 1001, which is H'79.

Therefore, LCDDR2 is H'7779.

(c) LCDDR3

This register is configured as follows:

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LCDDR3:	—	—	—	—	Nhsw3	Nhsw2	Nhsw1	Nhsw0	Nhsp7	Nhsp6	Nhsp5	Nhsp4	Nhsp3	Nhsp2	Nhsp1	Nhsp0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bits 11 to 8—Horizontal Synchronization Width (Nhsw3 to Nhsw0): These bits specify the clock width of CL1.

Bits 7 to 0—Horizontal Synchronization Position (Nhsp7 to Nhsp0): These bits specify the position to output CL1.

How to calculate Nhsw and Nhsp

Nhsw: Horizontal retrace line interval is 8 dots. If CL1 pulse width of CL1 is 8 dots, 8/4 – 1 = 1, which is B'0000 0001, which is H'01

Nhsp: 480/4 – 1 = 119, which is B'0111 0111, which is H'77.

Therefore, LCDDR3 is H'0177.

(d) LCDDR4

This register is configured as follows:

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LCDDR4:	—	—	—	—	—	—	Nvt9	Nvt8	Nvt7	Nvt6	Nvt5	Nvt4	Nvt3	Nvt2	Nvt1	Nvt0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bits 9 to 0—Vertical Total Line Count (Nvt9 to Nvt0): These bits specify the total number of lines in the vertical screen direction, including the retrace line interval.

How to calculate Nvt

Since retrace line interval is not provided in the vertical direction,

Nvt: 320 – 1 = 319, which is B'0000 0001 0011 1111, which is H'013F.

Therefore, LCDDR4 is H'013F.

(e) LCDDR5

This register is configured as follows:

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LCDDR5:	—	—	—	—	—	—	Nvd9	Nvd8	Nvd7	Nvd6	Nvd5	Nvd4	Nvd3	Nvd2	Nvd1	Nvd0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bits 9 to 0—Vertical Display Line Count (Nvd9 to Nvd0): These bits specify the number of display lines in the vertical screen direction.

How to calculate Nvd

Nvd: 320 – 1 = 319, which is B'0000 0001 0011 1111, which is H'013F.

Therefore, LCDDR5 is H'013F.

(f) LCDDR6

This register is configured as follows:

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LCDDR6:	Mw4	Mw3	Mw2	Mw1	Mw0	—	Nvsp9	Nvsp8	Nvsp7	Nvsp6	Nvsp5	Nvsp4	Nvsp3	Nvsp2	Nvsp1	Nvsp0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bits 15 to 11—Interval of Output of LCD Driving Signal Alternation (Mw4 to Mw0): These bits specify the H/L interval of output of the LCD driving signal alternation.

Bits 9 to 0—FLM Clock Output Position (Nvsp9 to Nvsp0): These bits specify the FLM clock output position.

How to calculate Nvsp

Nvsp: Since the FLM clock is output on the first line, Nvsp = 0.

Therefore, Nvsp = 0, which is B'00 0000 0000.

Mw: Since the signal is output once in 29 times, the set value is (number of outputs) –1.

This means that Mw = 28. Therefore, Mw = 28, which is B'11100.

LCDDR6 is B'1110 0000 0000 0000l, which is H'E000.

Figure 4.7 shows the relationship of Nvd, Nvt, and Nht when using a 480 × 320-dot LCD with a horizontal retrace line of eight dots.

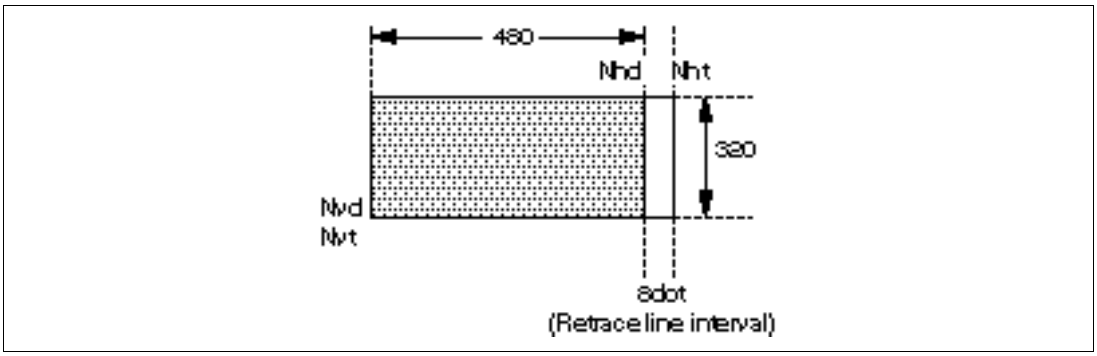


Figure 4.7 LCD Dot Display

(2) Setting Pallet Register (LCDPR)

There are 16 registers in the pallet register. The LCD used here uses only the LCDPR0 to LCDPR3 of the pallet register because the LCD is displayed in a monochrome 4 gradation mode.

LCDPR0 to LCDPR15 are configured as follows:

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LCDPR:	—	—	—	—	P11	P10	P9	P8	P7	P6	P5	P4	P3	P2	P1	P0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bits 15 to 12—Reserved

Bits 11 to 0—Data bits

The following describe how to specify colors.

- Sixteen colors, H'00 (white) to H'0F (black), can be specified in monochrome. Since only four colors can be specified here, four are chosen at random from the sixteen. In this test, we have chosen white (H'00), black (H'0F), gray (close to white) (H'05), and gray (close to black) (H'0A). The value of the four colors are then stored in LCDPR0 to LCDPR3 at random. Here, the following settings are made.
  LCDPR0: H'0000
  LCDPR1: H'0005
  LCDPR2: H'000A
  LCDPR3: H'000F

- If LCDAR is H'00, LCDPR0 is selected and white is displayed. In the same way,
  If LCDAR is H'01, LCDPR1 is selected and gray (close to white) is displayed.

If LCDAR is H'10, LCDPR2 is selected and gray (close to black) is displayed.
If LCDAR is H'11, LCDPR3 is selected and black is displayed.

Note: This setting is for LCD LM48072 manufactured by Sharp Corporation.
In DRAM, 00, 01, 10, and 11 of the LCDAR are written. For example, if 00 is written to the DRAM, it means that LCDPR0 has been written and white is displayed in the LCD. In the same way, if 01 is written to the DRAM, LCDPR1 has been written and gray close to white is displayed. Other colors will be displayed in the same way.

(3) Setting DMA Control Register (LCDDMR)

The LCDDMR specifies the start address of the LCD (start address of the memory where the LCD display data is stored) and the amount of transmit data. Since the LCD uses a single screen, only LCDDMR0 and LCDDMR1 in the upper screen display start address is used. The amount of transmit data is specified by bits 0 to 13 in LCDDMR4 , and little endian mode is specified by bit 15 in LCDDMR4. LCDDMR0, LCDDMR1, and LCDDMR4 are configured as follows:

LCDDMR0:

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LCDDMR0:	UM15	UM14	UM13	UM12	UM11	UM10	UM9	UM8	UM7	UM6	UM5	UM4	UM3	UM2	UM1	UM0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

LCDDMR1:

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LCDDMR1:	UM31	UM30	UM29	UM28	UM27	UM26	UM25	UM24	UM23	UM22	UM21	UM20	UM19	UM18	UM17	UM16
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

LCDDMR4:

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LCDDMR4:	LEM	—	TM13	TM12	TM11	TM10	TM9	TM8	TM7	TM6	TM5	TM4	TM3	TM2	TM1	TM0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

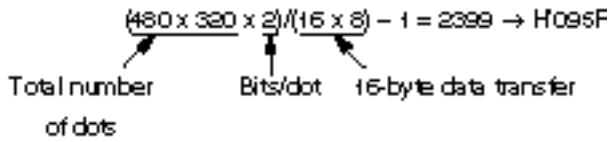
- A total of 32 bits of registers LCDDMR0 and LCDDMR1 specify the start address of the DRAM random area where display data is written to. The LCDDMR0 is the lower-side-panel display register and LCDDMR1 is the upper-side-panel display register. The reference

platform writes display data to an address range starting from H'0C010000 in the DRAM. Therefore, LCDDMR0 and LCDDMR1 will be as follows:

LCDDMR0: H'0000

LCDDMR1: H'0C01

- Specifies the amount of transmit display data. The following shows how to calculate the amount of transmit data.



- Since little endian mode is used, bit 15 in LCDDMR4 is set as 1.

From (b) and (c), LCDDMR4 is H'895F.

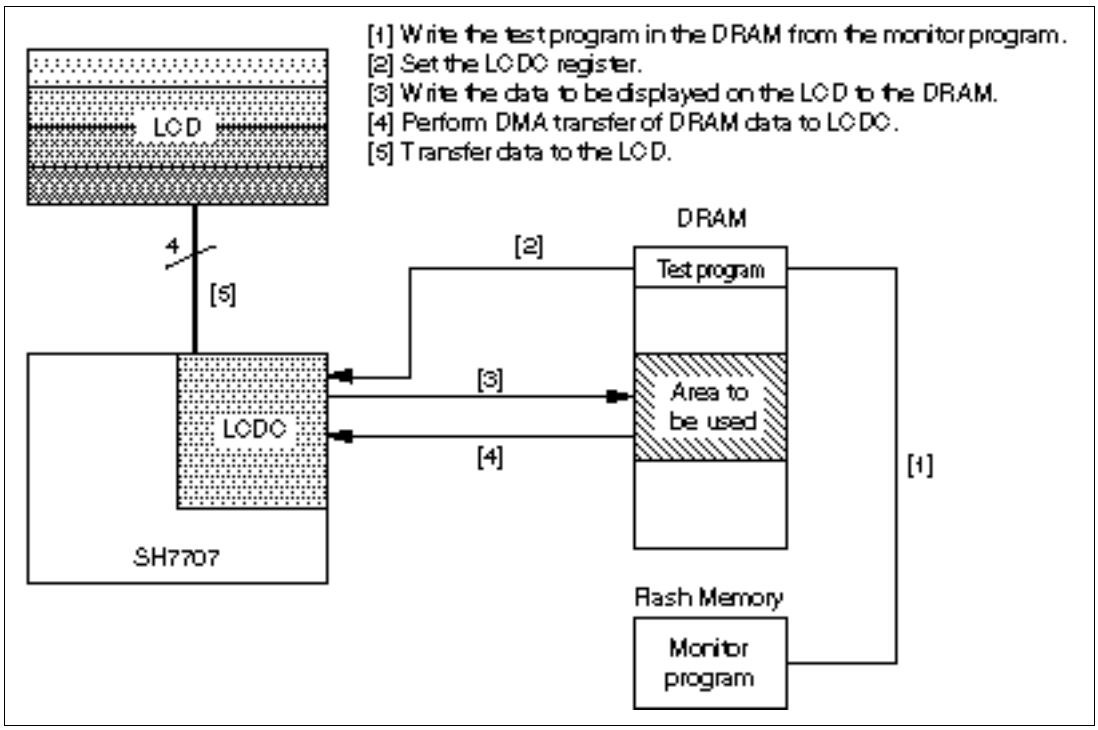


Figure 4.8 LCD Test Block Diagram

(4) Example of Setting LCD Controller Internal Operating Frequency

If the display is 480 × 320 dots and the frame frequency is 70 MHz, the LCDC operating frequency will be as follows:

480 × 320 × 70 = 10.752MHz ...(1)

(Total number of dots × frame frequency – LCDC operating frequency)

The operating frequency of the LCDC will be calculated by using the equation above. In the SH7707, since the bus clock (CKIO) is divided to determine the LCDC operating frequency (× 2, × 1, × 1/2, × 1/4, × 1/8), only specific values can be obtained. Therefore, the value obtained by dividing CKIO is assumed to be the operating frequency of LCDC. If CKIO is 32 MHz, the operating frequency of the LCDC is 64, 32, 16, 8, or 4 MHz, if CKIO is divided. If a value close to 10.752 MHz is selected, the operating frequency is 8 MHz. Therefore, we must adjust the operating frequency of the LCDC so that it becomes 8 MHz. When the size of the LCD cannot be changed, the frame frequency must be adjusted. In this case, frame frequency = 8 MHz/(480 × 320) = 52.08 Hz, and the size and frame frequency of the LCD and the operating frequency of LCDC can be determined.

LCDC operating frequency: 8MHz

Frame frequency: 52.08Hz

(5) Time for One Clock (Example When the LCDC Operating Frequency is 8 MHz)

If the LCDC operating frequency is 8 MHz, one clock is 0.125 µs (=t).

Table 4.5 Time for One Clock

Signal Name	Calculation	Time
CL1	t × number of dots (horizontal)	60 µs
CL2	t × 4 dots	500 ns
FLM	CL1 × number of dots (vertical)	19.2 ms
M	CL1 × 29 × 2	3.5 ms

4.3 PCMCIA

4.3.1 PC Card Controller (PCC)

The PC card controller (PCC) in the SH7707 has an interface pin function dedicated to the PCMCIA. The PCC can also control the level shifter and interrupts.

4.3.2 Features

(1) One PC card based on PCMCIA Rev.2.1/JEIDA Ver.4.2 is supported. *1 *2

(2) Level shifter can be controlled.

(3) A memory space of 64 Mbytes, which is the largest in the PCMCIA specifications, can be accessed.

Notes: 1. Usually, the on-chip PC card controller can control two PC card interface slots simultaneously. However, the reference platform has only one slot.

2. Both 5-V and 3.3-V operation are supported by the board specifications, however, due to the limitations of the card slot specifications, only 5 V can be used.

4.3.3 Physical Areas

(1) Since physical areas 5 and 6 are assigned for the PCMCIA physical area in the SH7707, two cards can be supported. However, since the reference platform has one slot, area 6 is used to support the memory card and I/O card. The address range of the memory card is H'18000000 to H'19FFFFFF, and H'1A000000 to H'1BFFFFFF is for the I/O card. The attribute memory space is the same as the memory card (H'18000000 to H'19FFFFFF).

(2) Two area modes can be selected for the memory space: a 32-Mbyte area mode and a 16-Mbyte area mode. To select an area mode, the Area 6 General Control Register (PCC0GCR) is used. Initial setting is the 32-Mbyte area mode. Refer to tables 4.6 and 4.7 for physical area configuration.

Table 4.6 32-Mbyte Area Mode Physical Space Configuration

SH7707 Memory Space	P0REG	P0PA25	PC Card Address Space
Attribute memory/common memory (H'18000000 to H'19FFFFFF)	0	X	Attribute memory (32 MB)
	0	1	Common memory (0:32 Mbyte)
	1	1	Common memory (32:64 MB)
I/O space (H'1A000000 to H'1BFFFFFF)	0	X	I/O space (32 MB)

Table 4.7 16-Mbyte Area Mode Physical Space Configuration

SH7707 Memory Space	P0REG	P0PA24	P0PA25	PC Card Address Space
Attribute memory (H'18000000 to H'18FFFFFF)	0	X	X	Attribute memory (16 MB)
Common memory (H'19000000 to H'19FFFFFF)	1	0	0	Common memory (0:16 MB)
	1	1	0	Common memory (16:32 MB)
	1	0	1	Common memory (32:48 MB)
	1	1	1	Common memory (48:64 MB)
I/O space (H'1A000000 to H'1BFFFFFF)	0	X	X	I/O space (16 MB)

4.3.4 PCMCIA Block Diagram

Figure 4.9 shows the connection of the SH7707 and card on the reference platform. Hitachi HD151015 is used for the level shifter for the card and SH7707.

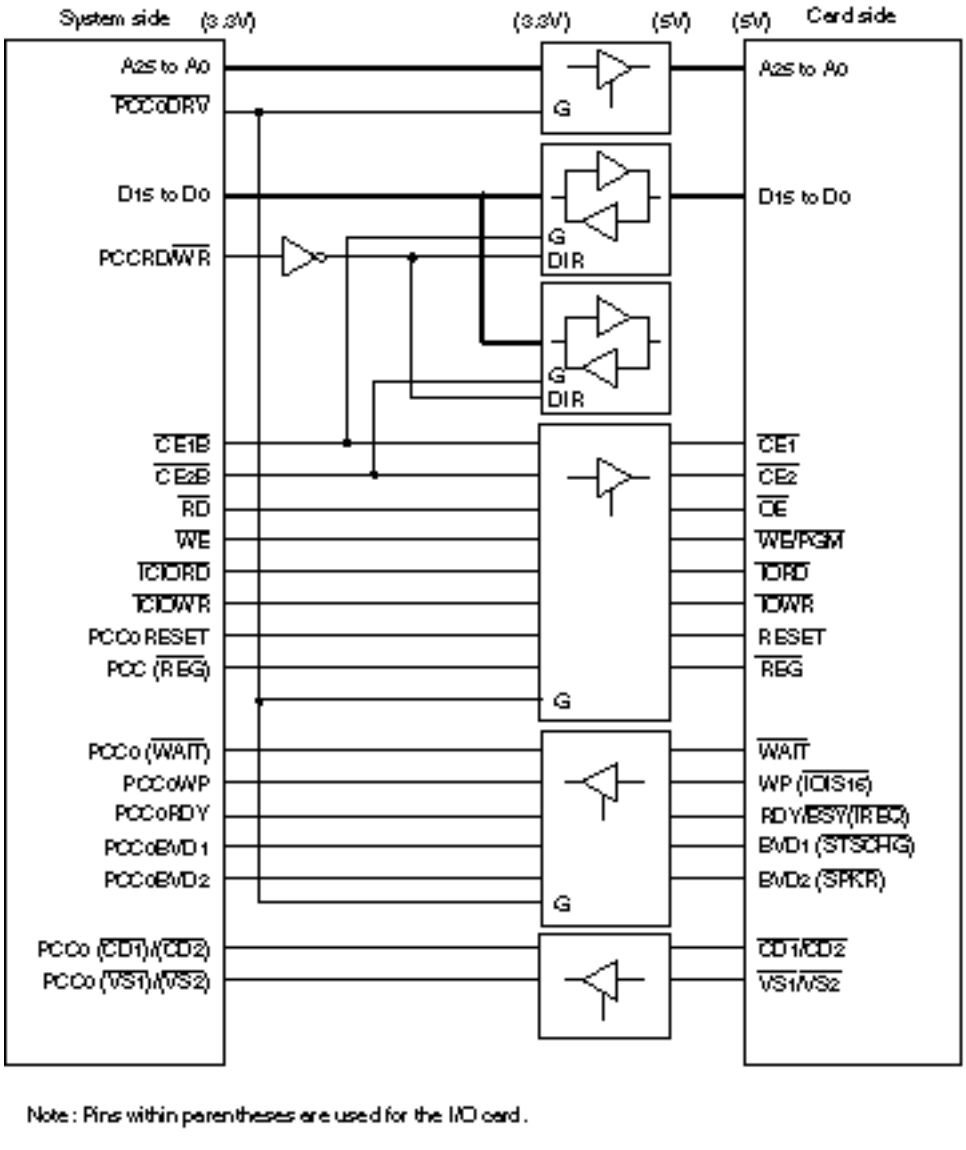


Figure 4.9 PCMCIA Block Diagram

4.3.5 PCMCIA Pin Function (Control Signal)

Table 4.8 lists the SH7707/PCC pin configuration.

Table 4.8 Pin Function

SH7707 Pin No.	Pin Name	I/O	Function (Connection to Card)
88	RD	Output	Read (Connects to OE)
90	WE1	Output	Write enable 1 (Connects to WE)
91	WE2	Output	Write enable 2 (Connects to IORD)
92	WE3	Output	Write enable 3 (Connects to IOWR)
102	CE1B	Output	Card enable 1 (Connects to CE1)
104	CE2B	Output	Card enable 2 (Connects to CE2)
94	PCC RD/(WR)	Output	Read/write (Connects to level shifter DIR)
116	PCC(REG)	Output	Attribute-memory-space selection (Connects to REG)
117	PCC0(DRV)	Output	Buffer control (Connects to level shifter G)
124	PCC0(WAIT)	Output	Wait (Connects to WAIT)
126	PCC0WP	Output	Write protection (Connects to WP)
127	PCC0READY	Output	Ready (Connects to READY)
128	PCC0BVD1	Input	Battery voltage detection 1 (Connects to BVD1)
129	PCC0BVD2	Input	Battery voltage detection 2 (Connects to BVD2)
130	PCC0(CD1)	Input	Card detection 1 (Connects to CD1)
131	PCC0(CD2)	Input	Card detection 2 (Connects to CD2)
133	PCC0(VS1)	Input	Voltage detection 1 (Connects to VS1)
135	PCC0(VS2)	Input	Voltage detection 2 (Connects to VS2)
118	PCC0RESET	Output	Reset (Connects to HD74HC32, and to the reset signal of the card)
186	V_{PP} EN0	Output	Program-power-supply control 1 (Connects to LTC1472CS)
190	V_{PP} EN1	Output	Program-power-supply control 2 (Connects to LTC1472CS)
119	V_{CC} EN0	Output	Power-supply control 1 (Connects to LTC1472CS)
120	V_{CC} EN1	Output	Power-supply control 2 (Connects to LTC1472CS)

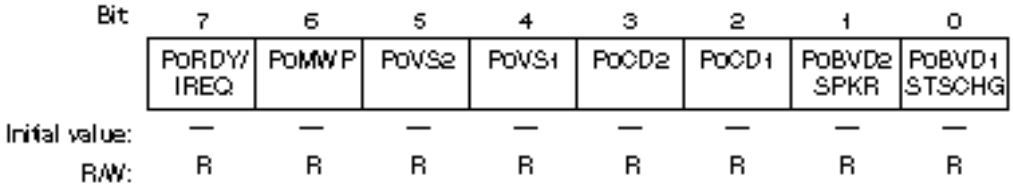
4.3.6 Register Configuration

Table 4.9 lists the register configuration. For details, refer to the SH7707 Hardware Manual.

Table 4.9 Register Configuration

Physical Space	Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Area 6	Area 6 interface status register	PCC0ISR	R	—	H'040000E0	8 bits
	Area 6 general control register	PCC0GCR	R/W	H'00	H'040000E2	8 bits
	Area 6 card status change register	PCC0CSCR	R/W	H'00	H'040000E4	8 bits
	Area 6 card status change interrupt enable register	PCC0CSCIER	R/W	H'00	H'040000E6	8 bits

(1) Area 6 Interface Status Register (PCC0ISR)



Bit 7—P0RDY/IREQ: This bit specifies the state of the card.

Bit 7

P0RDY/IREQ	Description
0	Executing instruction
1	Waiting to receive transmitted data

Bit 6—P0MWP: This bit specifies whether the card can be written to or not.

Bit 6

P0MWP	Description
0	Writing enabled
1	Writing disabled

Bits 5 and 4—P0VS2 and P0VS1: These bits specify the value of V_{CC} voltage at initial power supply and for reading CIS.

Bit 5	Bit 4	
P0VS2	**P0VS1**	**Description**
1	0	5-V for reading CIS or 3.3-V for reading CIS*
1	1	5-V for reading CIS

Note: Depends on the card slot type.

Bits 3 and 2—P0CD2 and P0CD1: These bits specify correct card insertion.

Bit 3	Bit 2	
P0CD2	**P0CD1**	**Description**
0	0	The card is inserted correctly.
0	1	The card is not inserted correctly.
1	0	
1	1	

Bits 1 and 0—P0BVD2 and P0BVD1: These bits specify the state of the card's battery voltage.

Bit 1	Bit 0	
P0BVD2	**P0BVD1**	**Description**
0	0	Data is not guaranteed. Replace the battery.
0	1	Data is maintained, but replace the battery.
1	0	Data is not guaranteed. Replace the battery.
1	1	Correct battery voltage.

(2) Area 6 General Control Register (PCC0GCR)

Bit	7	6	5	4	3	2	1	0
	P0DRVE	P0PCCR	P0PCCT	—	P0MMOD	P0PA25	P0PA24	P0REG
Inital value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	—	R/W	R/W	R/W	R/W

Bit 7—P0DRVE: This bit controls external buffer.

Bit 7		
P0DRVE	**Description**	
0	Opens the gate of external buffer	(Initial value)
1	Closes the gate of external buffer	

Bit 6—P0PCCR: This bit specifies reset state.

Bit 6		
P0PCCR	**Description**	
0	Reset wait state	(Initial value)
1	Reset	

Bit 5—P0PCCT: This bit specifies the card type.

Bit 5		
P0PCCT	**Description**	
0	IC memory card	(Initial value)
1	I/O card	

The shaded parts show the settings of the reference platform.

Bit 3—P0MMOD: This bit selects area mode.

Bit 3		
P0MMOD	**Description**	
0	32-Mbyte area mode	(Initial value)
1	16-Mbyte area mode	

Bits 2 and 1—P0PA25 and P0PA24: These bits specify the access area.

Bit 2	Bit 1	
P0PA25	**P0PA24**	**Description**
0	0	(Initial value) For details, refer to tables 4.6 and 4.7 in section 4.3.3, Physical Areas.
0	1	
1	0	
1	1	

Bit 0—P0REG: This bit specifies the memory to access.

Bit 0

P0REG	Description	
0	Attribute memory access	(Initial value)
1	Common memory access	

The shaded parts show the settings of the reference platform.

(3) Area 6 Status Change Register (PCC0CSCR)

Bit	7	6	5	4	3	2	1	0
	P0SCDI	—	IREQ	SC	P0CDC	P0RC	P0BW	P0BD
Inital value:	0	0	0	0	0	0	0	0
R/W:	R/W	—	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7—P0SCDI: This bit generates a software interrupt when there is a change in the card insertion state.

Bit 7

P0SCDI	Description	
0	An interrupt will not occur.	(Initial value)
1	When this bit is 1, an interrupt will occur if there is a change in the card insertion state.	

Bit 5—IREQ: This bit shows the state of the IREQ pin interrupt request when the I/O card is selected. This bit always reads 0 when the IC memory card is selected.

Bit 5

IREQ	Description	
0	This bit reads 0 when no interrupt signal is sent to the IREQ pin.	(Initial value)
1	This bit reads 1 when an interrupt signal is sent to the IREQ pin.	

Bit 4—SC: This bit shows the change in the STSCHG pin when the I/O card is selected. This bit always reads 0 when the IC memory card is selected.

Bit 4

SC	Description
0	This bit reads 0 when the STSCHG pin does not change from 1 to 0. (Initial value)
1	This bit reads 1 when the STSCHG pin changes from 1 to 0.

The shaded parts show the settings of the reference platform.

Bit 3—P0CDC: This bit shows the change of the CD1 and CD2 pins.

Bit 3

P0CDC	Description
0	This bit always reads 0 when the CD1 and CD2 pins do not change. (Initial value)
1	This bit reads 1 when the CD1 and CD2 pins change.

Bit 2—P0RC: This bit shows the state of the RDY/BSY pin when the IC memory card is selected. This bit always reads 0 when the I/O card is selected.

Bit 2

P0RC	Description
0	This bit reads 0 when the state of the RDY/BSY pin does not change. (Initial value)
1	This bit reads 1 when the state of the RDY/BSY pin changes.

Bit 1—P0BW: This bit shows the state of the BVD2 and BVD1 pins when the IC memory card is selected (the states of BVD2 and BVD1 are guaranteed). This bit always reads 0 when the I/O card is selected.

Bit 1

P0BW	Description
0	This bit reads 0 when (BVD1, BVD2) changes to a state other than (0,1). (Initial value)
1	This bit reads 1 when (BVD1, BVD2) changes to (0,1).

The shaded parts show the settings of the reference platform.

Bit 0—P0BD: This bit shows the state of the BVD2 and BVD1 pins when the IC memory card is selected (the states of BVD2 and BVD1 are not guaranteed). This bit always reads 0 when the I/O card is selected.

Bit 0

P0BD	Description
0	This bit reads 0 when (BVD1, BVD2) changes to a state other than (1,0) or (0,0).
1	This bit reads 1 when (BVD1, BVD2) changes to (1,0) or (0,0).

(4) Area 6 Card Status Change Interrupt Enable Register (PCC0CSCIER)

Bit:	7	6	5	4	3	2	1	0
	P0CRE	IREQE1	IREQE0	SCE	P0CDE	P0RE	P0BWE	P0BDE
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7—P0CRE: This bit initializes the PCC0GCR.

Bit 7

P0CRE	Description
0	The PCC0GCR is not initialized when this bit reads 0.
1	The PCC0GCR is initialized when the CD1 and CD2 pins are detected.

Bits 6 and 5—IREQE1 and IREQE0: These bits show the validity of the IREQ pin interrupt request when the I/O card is selected.

Bit 6	Bit 5	
IREQE1	IREQE0	Description
0	0	A request for interrupt is not accepted.
0	1	Level mode (level 0) interrupt is accepted.
1	0	Pulse mode (rising edge) interrupt is accepted.
1	1	Pulse mode (falling edge) interrupt is accepted.

The shaded parts show the settings of the reference platform.

Bit 4—SCE: This bit shows the validity of interrupts when the state of the BVD (STSCHG) pin changes and the I/O card is selected.

Bit 4

SCE	Description
0	An interrupt will not be generated even if the state of the BVD (STSCHG) pin changes.
1	An interrupt will be generated when the state of the BVD (STSCHG) pin changes from 1 to 0.

Bit 3—P0CDE: This bit shows the validity of interrupts when the states of the CD1 and CD2 pins change.

Bit 3

P0CDC	Description
0	An interrupt will not be generated even if the states of the CD1 and CD2 pins change.
1	An interrupt will be generated when the states of the CD1 and CD2 pins change.

Bit 2—P0RE: This bit shows the validity of interrupts when the state of the RDY/BSY pin changes in the IC memory card.

Bit 2

P0RC	Description
0	An interrupt will not be generated even if the state of the RDY/BSY pin changes.
1	An interrupt will be generated when the state of the RDY/BSY pin changes.

The shaded parts show the settings of the reference platform.

Bit 1—P0BWE: This bit shows the validity of interrupts when the state of the BVD2 and BVD1 pins (BVD2, BVD1) is (0,1) in the IC memory card.

Bit 1

P0BW	Description
0	An interrupt will not be generated.
1	An interrupt will be generated when (BVD1, BVD2) is (0,1).

Bit 0—P0BDE: This bit shows the validity of interrupts when the state of the BVD2 and BVD1 pins (BVD2, BVD1) is (1,0) or (0,0) in the IC memory card.

Bit 0

P0BD	Description
0	An interrupt will not be generated.
1	An interrupt will be generated when (BVD1, BVD2) changes to (1,0) or (0,0).

The shaded parts show the settings of the reference platform.

4.3.7 Pin Description

The following describe the pin function.

(1) Read (RD: Output)
This pin is connected to the OE pin of the card and is used to control the operation of reading data from the card.

(2) Write enable 1 (WE1: Output)
This pin is connected to the WE/PGM pin of the card and is used to control the operation of writing data to the card.

(3) Write enable 2 (WE2: Output)
This pin is connected to the IORD pin of the card and is used to read data from the I/O area of the card.

(4) Write enable 3 (WE3: Output)
This pin is connected to the IOWR pin of the card and is used to write data to the I/O area of the card.

(5) Card enable 1 and 2 (CE1B and CE2B: Output)
These pins are connected to the CE1 and CE2 pins of the card and control the bus state (controls even and odd addresses). Refer to section 4.3.10.

(6) Read/write (PCC RD/(WR): Output)
This pin is connected through an inverter to the DIR pin of the level shifter. When high-level signal (RD) is output, low-level signal is input to the DIR of the level shifter through the inverter. In this case, the signal is transmitted from the card side to the CPU side through the level shifter. When low-level signal (WR) is output, the signal is transmitted from the CPU side to the card side through the level shifter.

(7) Attribute memory space select (PCC(REG): Output)

This pin is connected to the REG pin of the card. When accessing common memory, a high-level signal is output from the PCC(REG). When accessing attribute memory, a low-level signal is output from the PCC(REG).

(8) Buffer control (PCC0(DRV): Output)

This pin sends signals to the G (gate) pin of the level shifter. When a high-level signal is input to the G pin, the card side and CPU side of the level shifter are separated. When a low-level signal is input to the G pin, data can be transmitted.

(9) Wait (PCC0(WAIT): Input)

This pin is connected to the WAIT pin of the card. A low-level signal is input to the PCC0WAIT from the WAIT pin (card side) to delay the termination of the memory access cycle or the I/O access cycle (inserts wait).

(10) Write protection (PCC0WP: Input)

This pin is connected to the WP pin of the card. If the card is write-protected, a high-level signal is input to the PCC0WP pin. If the card is write-enabled, a low-level signal is input to the PCC0WP pin.

(11) Ready (PCC0READY: Output)

This pin is connected to the RDY/BSY pin of the card. When an instruction is being executed, a low-level signal is input to the PCC0READY pin from the card side. When data transmission is enabled, a high-level signal is input to the PCC0READY pin from the card side.

(12) Battery voltage detection 1 and 2 (PCC0BVD1 and PCC0BVD2: Input)

These pins are connected to the BVD2 and BVD1 pins of the card. The battery voltage state is input to PCC0BVD1 and PCC0BVD2 pins. Refer to section 4.3.9.

(13) Card detection 1 and 2 (PCC0(CD1) and PCC0(CD2): Input)

These pins are connected to the CD1 and CD2 pins of the card. The card-inserted state is input to the PCC0(CD1) and PCC0(CD2) pins. Refer to section 4.3.8.

(14) Voltage detection (PCC0(VS1) and PCC0(VS2): Input)

These pins are connected to the VS1 and VS2 pins of the card. The voltage (V_{CC}) at the initial power supply and for reading the CIS is input to PCC0(VS1) and PCC0(VS2) pins. Refer to section 4.3.13.

(15) Reset (PCC0RESET: Output)

This pin is connected to the RESET pin of the card.

(16) Program-power-supply control (V_{PP}EN0 and V_{PP}EN1: Output)

These pins are connected to the power supply controller (LTC1472CS: Linear Technology Corp.). They provide a 12-V power supply when the flash card is used. Refer to section 4.3.16.

(17) Card-power-supply control (V_{CC}EN0 and V_{CC}EN1: Output)

These pins are connected to the power supply controller (LTC1472CS: Linear Technology Corp.). They provide a 5-V or 3.3-V power supply to the card. Refer to section 4.3.15.

4.3.8 Card Detection

Card insertion can be detected by the SH7707 CD1 (pin 130) and CD2 (pin 131). Since CD1 and CD2 pins are low-active, card insertion is confirmed when a low-level signal is sent to both pins on the system side. The pins are pulled-down at the card side and pulled-up at the system side. Therefore, if the card is inserted into the system correctly, the signal levels of the CD1 and CD2 pins are detected. If a low-level signal is detected for both the CD1 and CD2 pins, the card insertion is confirmed.

4.3.9 Battery Voltage Detection

Pins 128 and 129 output the state of the battery voltage listed in table 4.10.

Table 4.10 Battery Voltage Description

BVD1	BVD2	Description
1	1	Correct battery voltage.
1	0	Data is maintained. However, replace the battery.
0	1	Data is not guaranteed. Replace the battery.
0	0	Data is not guaranteed. Replace the battery.

4.3.10 Wait Cycle Insertion

(1) Wait Cycle Insertion Specified by Wait Control Register 2 (WCR2)

A usual wait cycle insertion can be specified by wait control register 2 (WCR2) of the bus state controller. Wait control register 2 is a readable/writable 16-bit register and specifies the number of wait state cycles to be inserted. The number of wait states inserted for the reference platform is ten.

Table 4.11 Wait Cycle Insertion

Area 6				
Bit 15	**Bit 14**	**Bit 13**	**Function**	
A6W2	**A6W1**	**A6W0**	**Number of states inserted**	**WAIT pin**
0	0	0	0	Ignored
0	0	1	1	Enabled
0	1	0	2	Enabled
0	1	1	3	Enabled
1	0	0	4	Enabled
1	0	1	6	Enabled
1	1	0	8	Enabled
1	1	1	10	Enabled

The shaded part shows the setting of the reference platform.

(2) Specifying the Timing of Assertion and Negation of Signals OE and WE by the PCMCIA Control Register (PCR)

The PCMCIA control register is used to specify the timing of assertion and negation of OE and WE signals of the PCMCIA interface connected to area 6. Settings are made by the wait control bit of the WCR2 register. The address-OE/WE assertion delay and OE/WE negation-address delay are 3.5 cycles in the reference platform.

Table 4.12 Area 6 Address-OE/WE Assertion Delay

Bit 5	**Bit 4**		
A6TED1	**A6TED0**	**Function**	
0	0	0.5-cycle delay	(Initial value)
0	1	1.5-cycle delay	
1	0	2.5-cycle delay	
1	1	3.5-cycle delay	

Table 4.13 Area 6 OE/WE Negation-Address Delay

Bit 1	Bit 0		
A6TEH1	**A6TEH0**	**Function**	
0	0	0.5-cycle delay	(Initial value)
0	1	1.5-cycle delay	
1	0	2.5-cycle delay	
1	1	3.5-cycle delay	

The shaded parts show the settings of the reference platform.

Note In (1) and (2) in section 4.3.10, the external bus frequency can be increased by inserting wait cycles. However, even if the maximum number of wait cycles (ten cycles) is specified, there is a limit to increasing the maximum external-bus operating frequency. Therefore, when accessing data with an operating frequency larger than the limited value, an external wait generating circuit is required.

4.3.11 Timing Chart

(1) Memory-Card Basic Timing (Read)

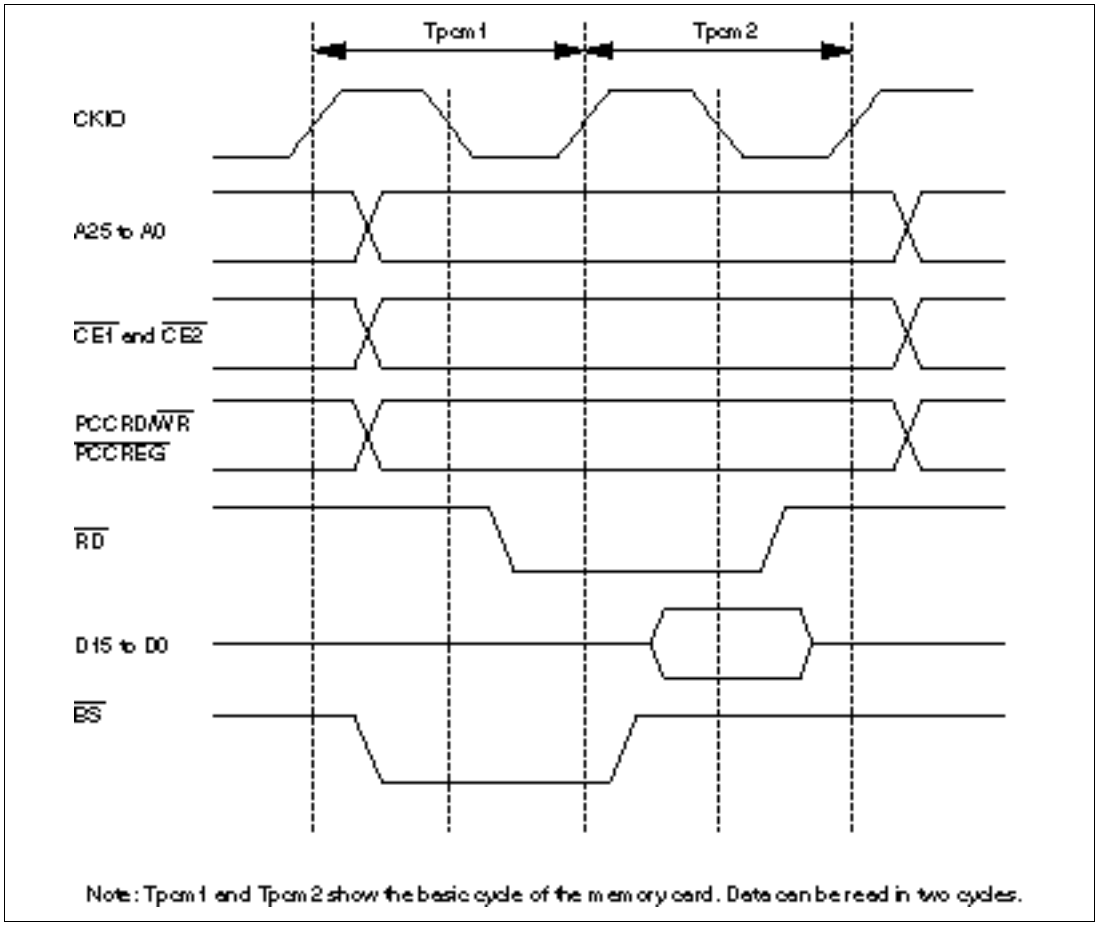


Figure 4.10 Memory Card (1)

(2) Memory-Space Read Timing (at Wait-Cycle Insertion)

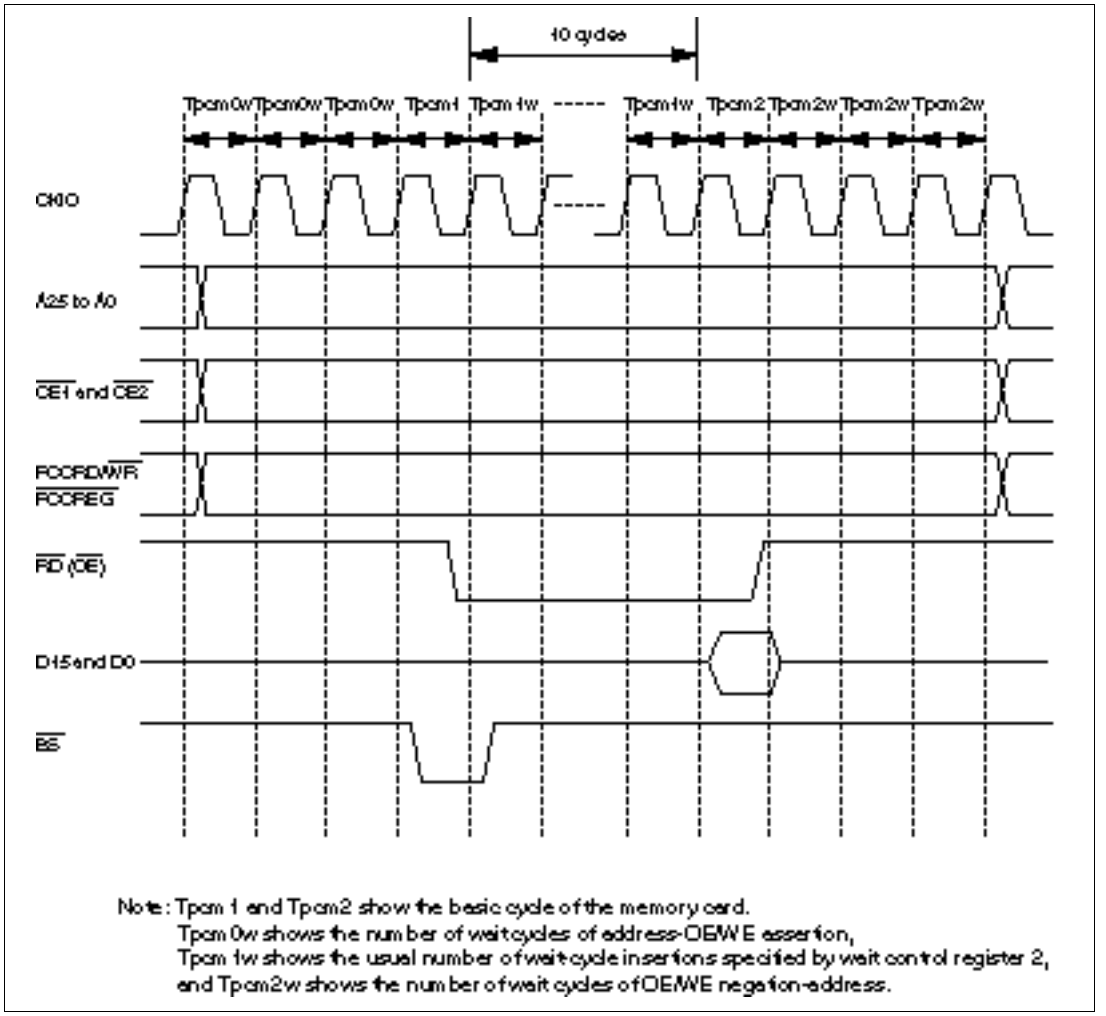


Figure 4.11 Memory Card (2)

(3) Memory-Card Basic Timing (Write)

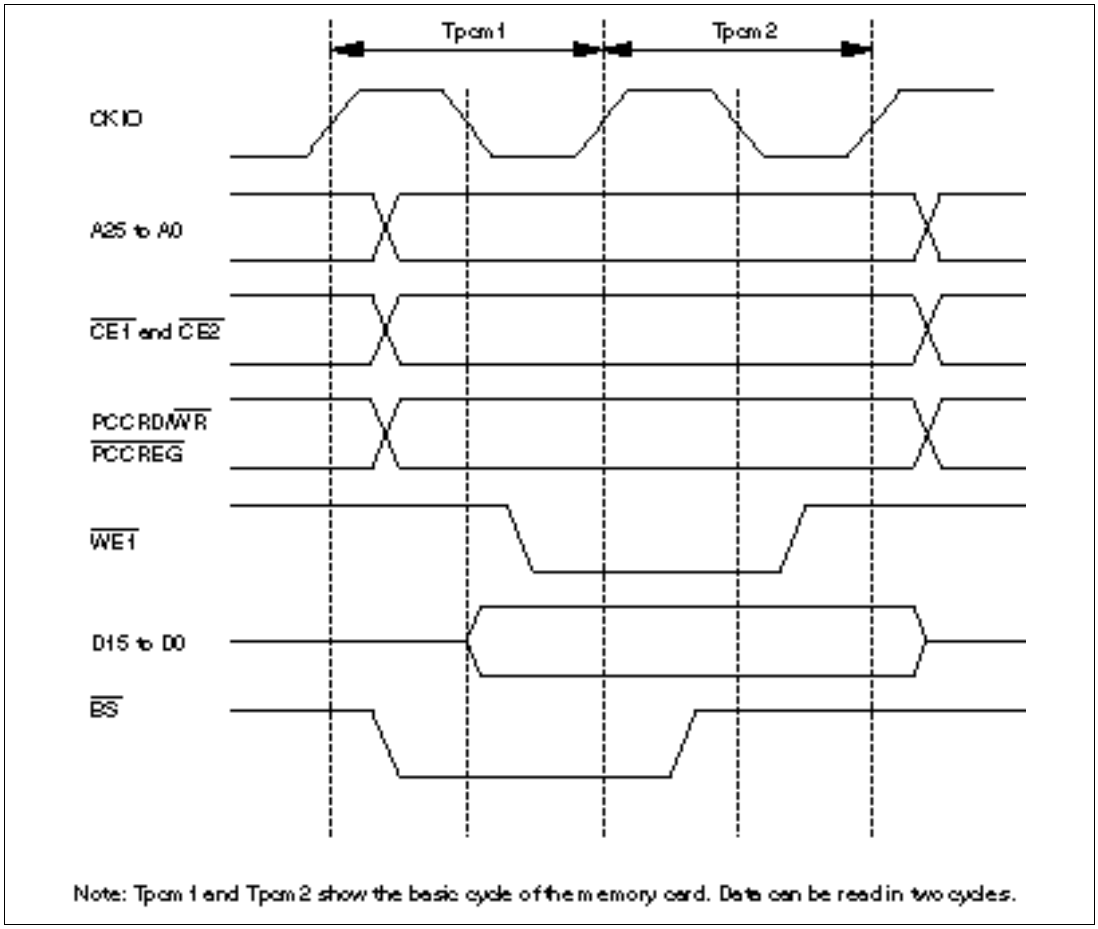


Figure 4.12 Memory Card (3)

(4) Memory Space Write Timing (at Wait-Cycle Insertion)

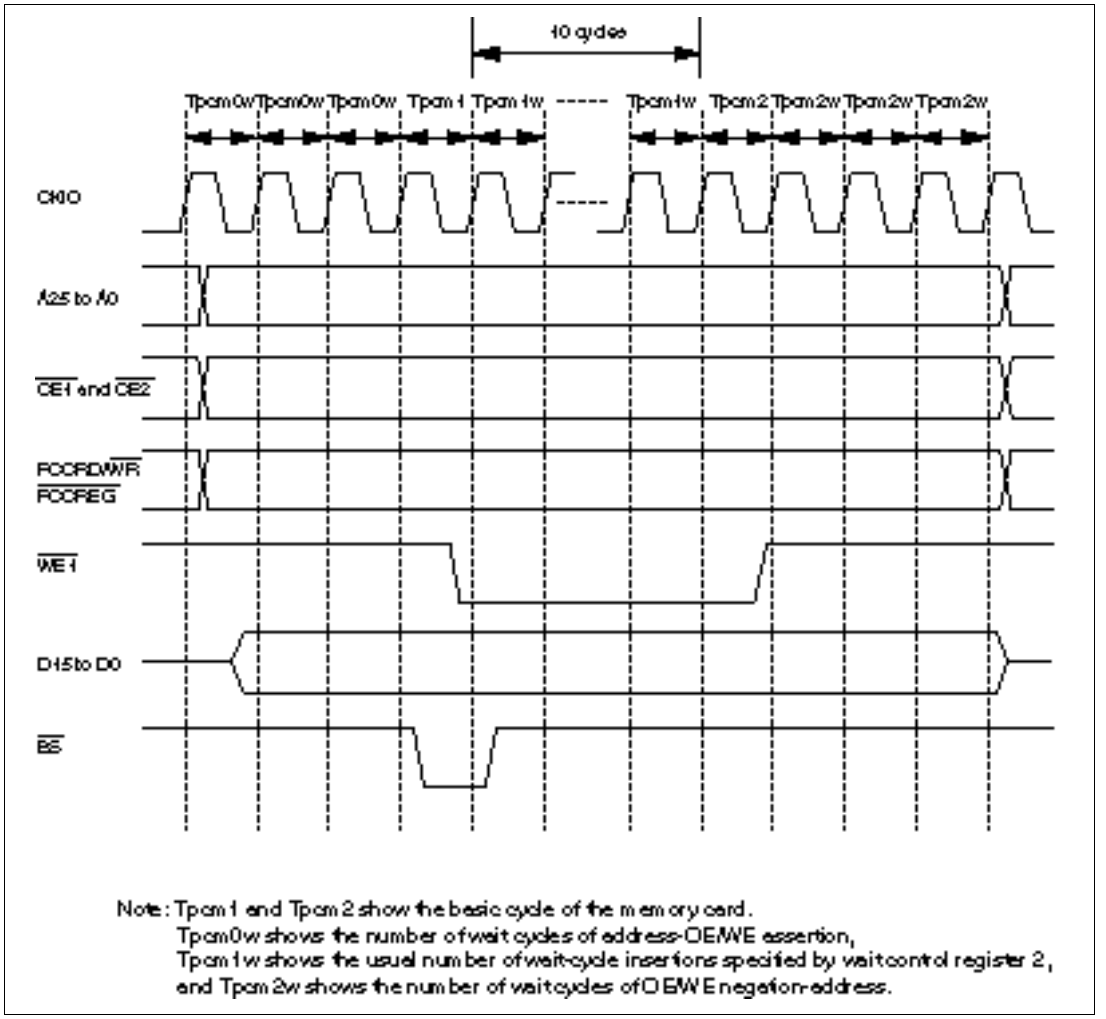


Figure 4.13 Memory Card (4)

(5) I/O-Card Basic Timing (IORD)

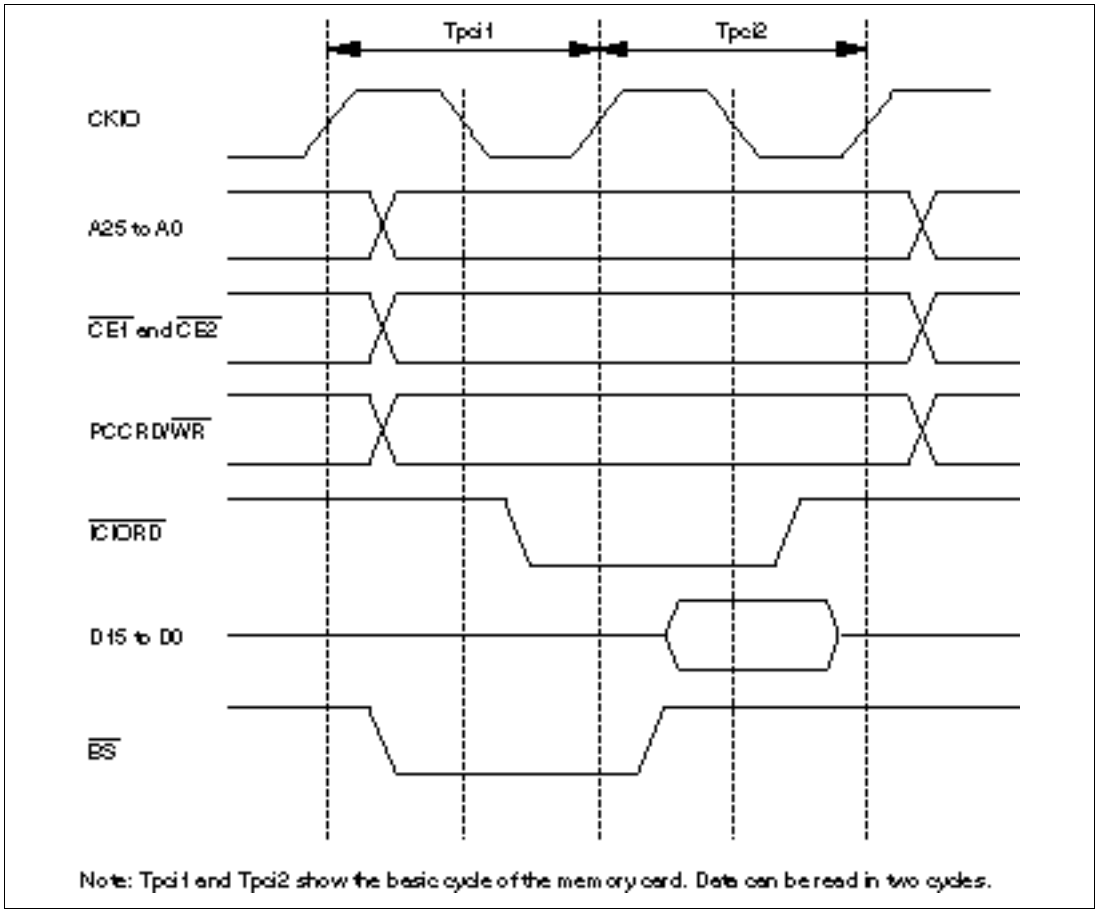


Figure 4.14 I/O Card (1)

(6) I/O Card Read (at Wait-Cycle Insertion)

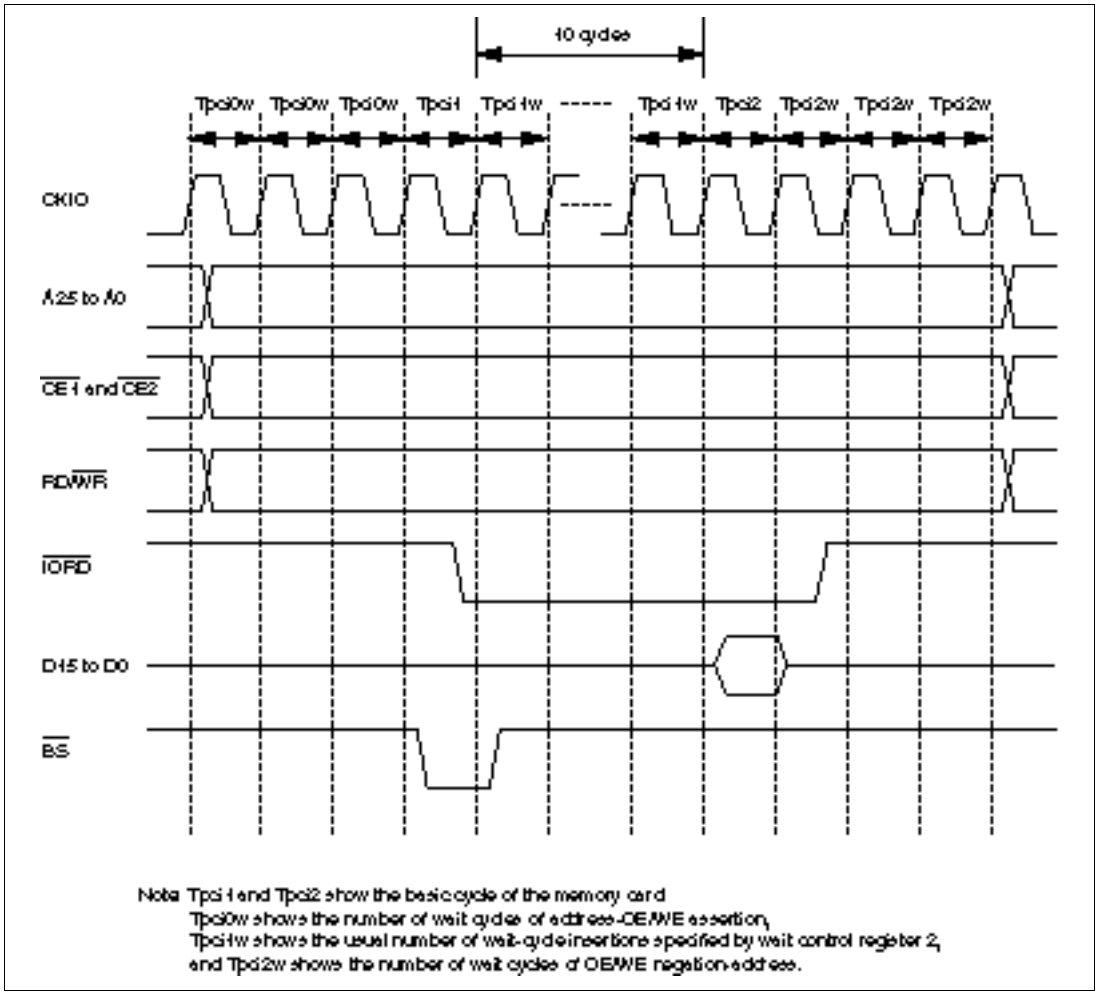


Figure 4.15 I/O Card (2)

(7) I/O-Card Basic Timing (IOWR)

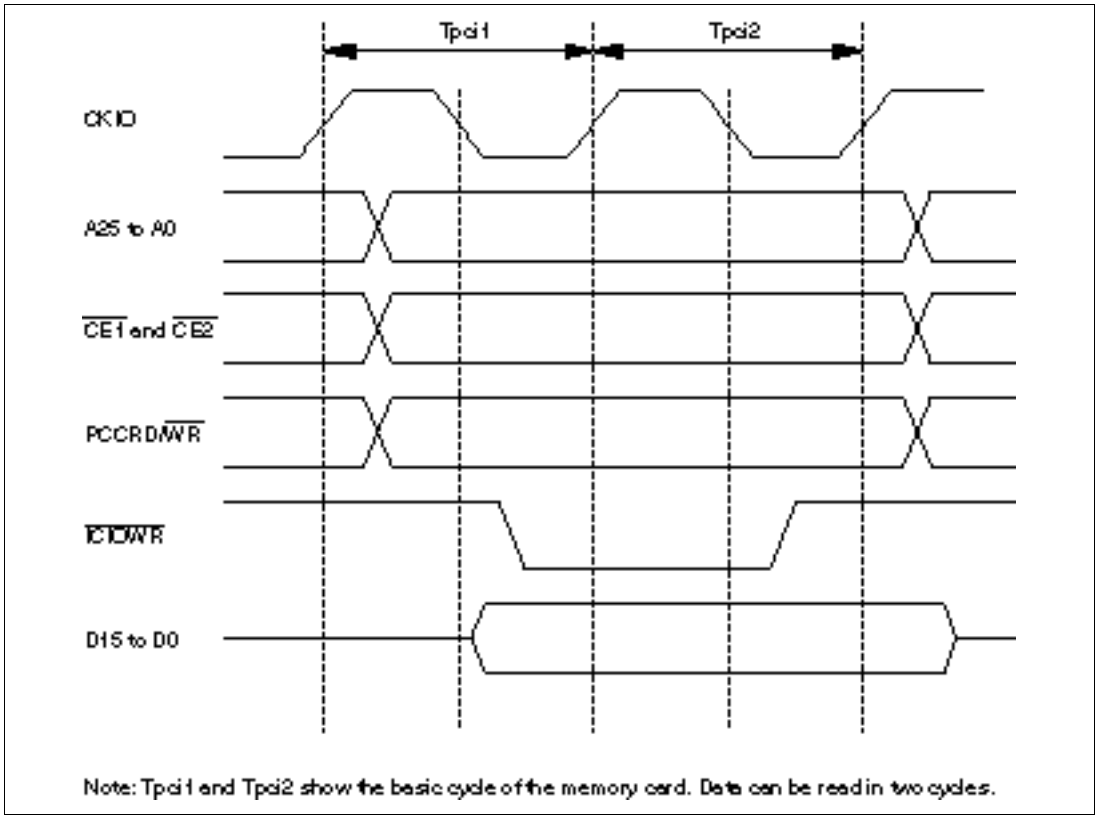


Figure 4.16 I/O Card (3)

(8) I/O-Card Write (at Wait-Cycle Insertion)

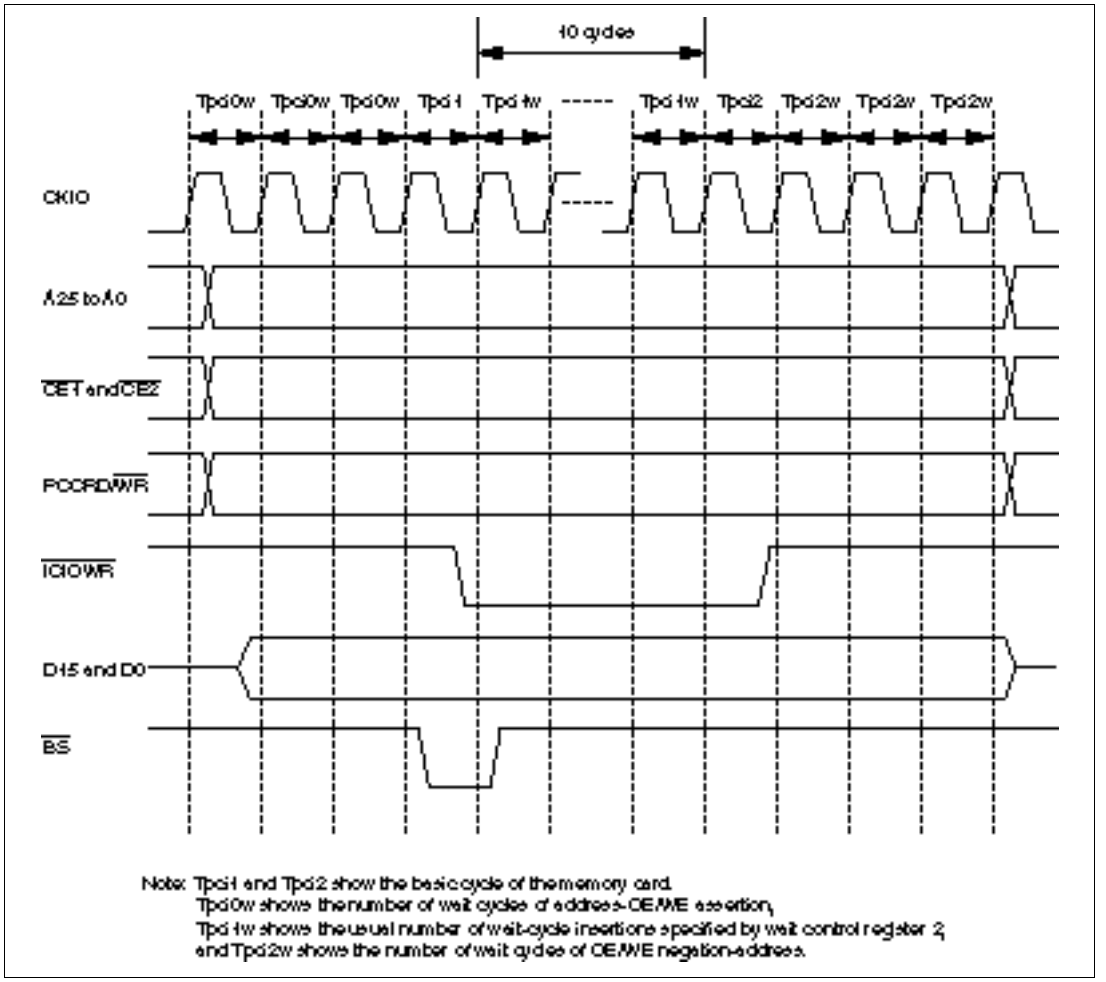


Figure 4.17 I/O Card (4)

4.3.12 Operation for Inserting the Card and Accessing the Card

The following procedures describe PC card access.

(1) Inserting the card

— 5-V card

The card can be inserted or removed before or after power-on. However, if the card is inserted before power-on, the definition of power-on and power-off given in PCMCIA Rev.2.1/JEIDA Ver.4.2 must be followed. If the card is inserted after power-on, the system must be non-activated.

— 3.3-V card

The card must be inserted or removed after power-off.

(2) Checking whether or not a card is inserted

Pins CD1 and CD2 check whether a card is inserted into the slot correctly.

Note: If a card is inserted correctly,the pins will generate an interrupt. However, if a card is not inserted, the system wait for an interrupt.

(3) Providing enough power supply for card operation

Determines the voltage that can be sent to the card by pins VS1 and VS2.

(4) Opening the gate of the level shifter (puts pin 23 to low level for HD151015)

The system is activated.

(5) Accessing attribute memory

Reads and analyzes attribute data.

(6) Reading the attribute memory determines whether the card inserted is a memory card or an I/O card

The card side and system side are first set as a memory card interface.

(7) Card access

— I/O card

(a) Selects the I/O-card interface for the card side

(b) Selects the I/O-card interface for the system side

(c) Accesses I/O card

— Memory card

Accesses memory card

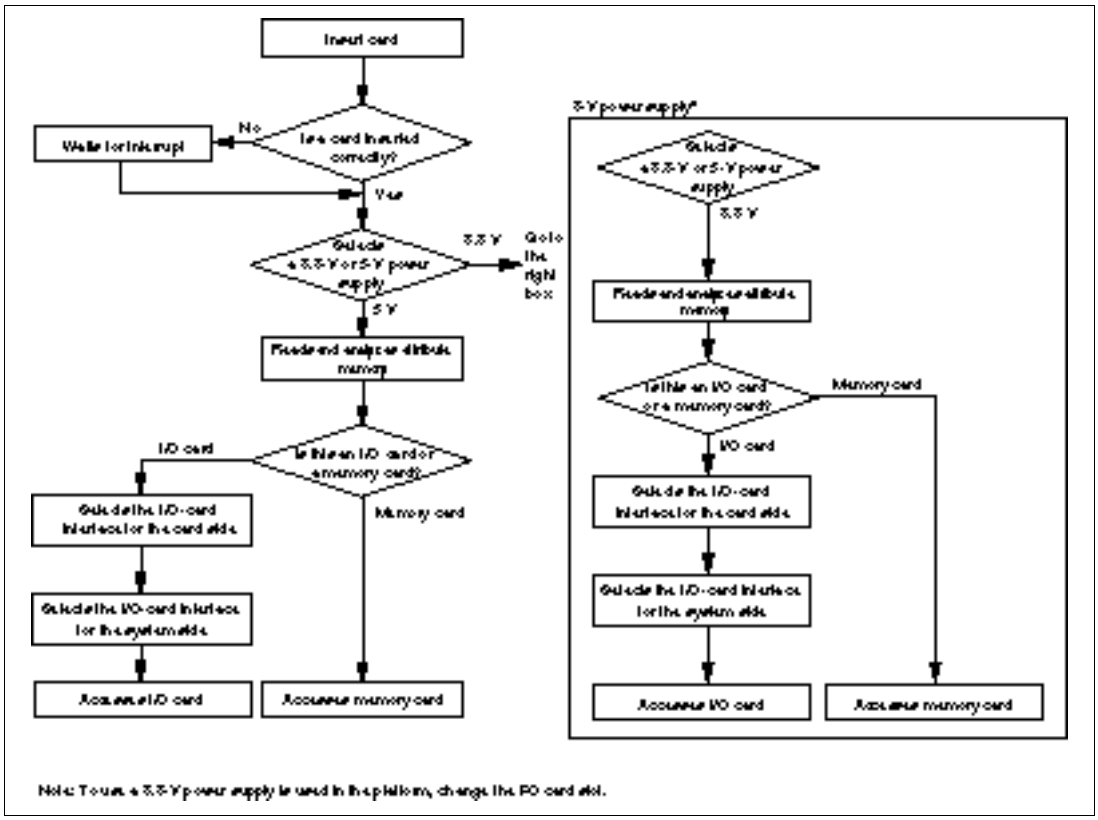


Figure 4.18 Flowchart of Card Access

4.3.13 Voltage Detection

Pins VS1 (pin 133) and VS2 (pin 135) of the SH7707 detect voltage used. The system side confirms the voltage (V_{CC}) of the card according to these two signals and provides the appropriate voltage to the card. Table 4.14 lists the voltage (V_{CC}) of the VS1 and VS2 signals.

Table 4.14 V_{CC} Voltage

Card	VS1	VS2	System (Slot Type)	Voltage (V_{CC}) at Initial Power-on
5-V card CIS: 5 V	H	H	5-V type	5V for reading CIS
5-V and 3.3-V card CIS: 5 V and 3.3 V	L	H	5-V type	5V for reading CIS
			3.3-V type	3.3V for reading CIS

4.3.14 Card Access Mode Selection

Pins CE1B (pin 102) and CE2B (pin 104) are specified for card access mode selection and control the bus.

The following actions describe PC card access.

(1) If CE1B and CE2B are both high-level signals, standby mode is entered.
(2) If CE1B is a low-level signal and CE2B is a high-level signal, byte access mode is entered. In this case, if A0 is low level, even addresses (D15 to D8) are accessed. If A0 is a high-level signal, odd addresses D7 to D0 are accessed. This attribute memory is stored in the even address area. The odd address is therefore invalid.
(3) If CE1B and CE2B are both low-level signals, standby mode is entered. When data is accessed in word size, A0 is ignored.

4.3.15 Power-Supply Control

Pins 119 and 120 control the power supply required for the card. 5 V and 3.3 V are supplied to the card. Table 4.15 lists the settings.

Table 4.15 Power-Supply-Control Pin Configuration

Pin 119 (V_{CC} EN0)	Pin 120 (V_{CC} EN1)	V_{CC} (OUT)
0	0	OFF
1	0	5 V
0	1	3.3 V
1	1	OFF

4.3.16 Power-Supply Control for Programming

Pins 186 and 190 control the power supply for the card (flash memory card, etc.), which requires 12 V for programming. Refer to table 4.16 for mode setting.

Table 4.16 Power-Supply-Control Pin Configuration for Programming

Pin 186 (V_{PP} EN0)	Pin 190 (V_{PP} EN1)	V_{PP} (OUT)
0	0	0V
0	1	V_{CC} (3V or 5V)
1	0	12V
1	1	Hi-Z

Note: V_{CC} is the same as in table 4.9.

4.4 Audio Play-Back

The audio play-back circuit outputs sound data, which was converted by the SH7707 D/A converter, to the speaker via the external audio-amplifier. The sound data is stored in the memory.

4.4.1 Audio-Play-Back Block Diagram

Figure 4.19 shows the block diagram for the audio play-back.

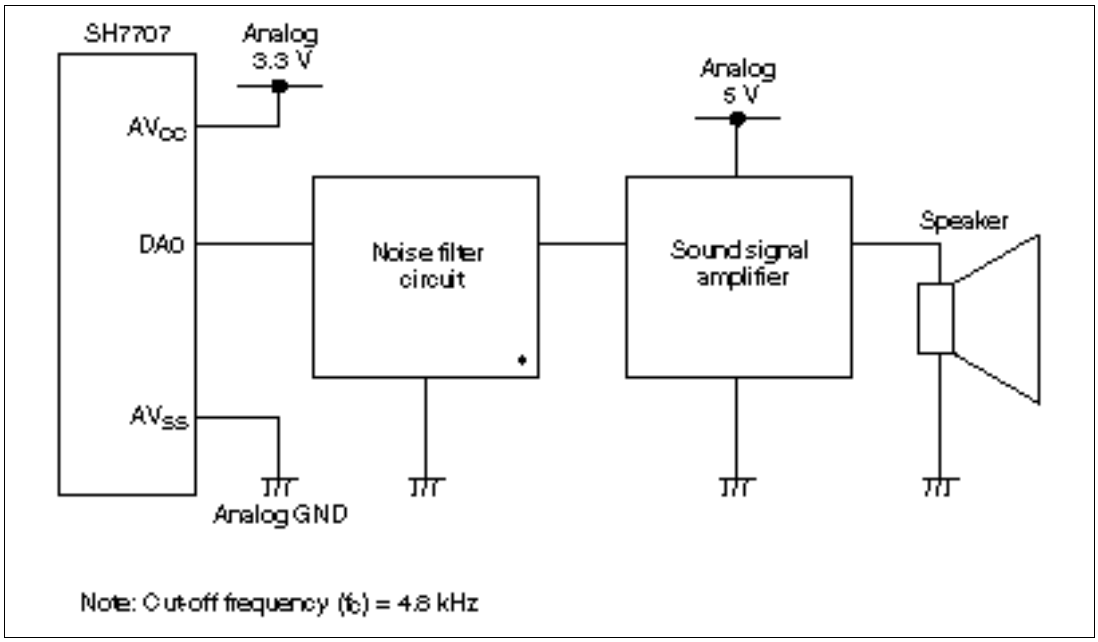


Figure 4.19 Block Diagram

4.4.2 SH7707 DAC Output Specification

Table 4.17 lists the DAC output specification.

Table 4.17 DAC Output Specification

Item	Specification
Resolution	8 bits (48 dB)
Channels used for output	Channel 1
Output level	0 to AV_{CC} (3.3 V)
D/A conversion time	Max: 10 µs

4.4.3 SH7707 Pin Specification

Table 4.18 lists the D/A converter pin specification.

Table 4.18 D/A Converter Pin Specification

Pin No.	Pin Name	Specification
205	AV_{CC}	Analog power supply (3.3 V)
206	DA1	Analog output (channel 1)
207	DA0	Analog output (channel 0)
208	AV_{SS}	Analog GND (0 V)

4.4.4 List of SH7707 D/A Converter Registers

The following figures show the D/A converter registers. Only channel 0 is used for audio output.

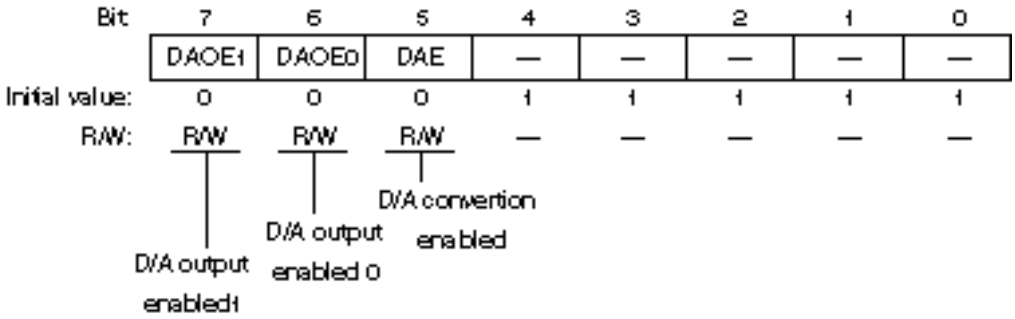
(1) DADR0/1 (D/A Data Register 0, 1)

Bit	7	6	5	4	3	2	1	0
Inital value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

The D/A data register (DADR0/1) is an 8-bit readable/writable register that stores data to be converted. When the analog output pin is enabled, the value of the D/A data register is always converted, and data is output from the analog output pin.

The D/A data register is initialized as H'00 at reset and in standby mode.

(2) DACR (D/A Control Register)



The D/A control register (DACR) enables and disables D/A conversion and channel output. Refer to tables 4.19 and 4.20 for details.

Table 4.19 Details of DACE

Bit	Name	Meaning	When 0	When 1
7	DAOE1	DA1: Enables or disables D/A conversion and channel 1 output	Disabled	Enabled
6	DAOE0	DA0: Enables or disables D/A conversion and channel 0 output	Disabled	Enabled
5	DAE	DAE: Controls DA1 and DA0	Refer to table 4.20	

Table 4.20 Details of DACR

DAOE1	DAOE0	DAE	Meaning
0	0	—	D/A conversion is disabled for both channel 0 and channel 1
0	1	0	D/A conversion is disabled for channel 1 and enabled for channel 0
0	1	1	D/A conversion is enabled for both channel 0 and channel 1
1	0	0	D/A conversion is disabled for channel 0 and enabled for channel 1
1	0	1	D/A conversion is enabled for both channel 0 and channel 1
1	1	—	D/A conversion is enabled for both channel 0 and channel 1

The shaded part shows the setting of the reference platform.

4.4.5 Circuit Diagram

Figure 4.20 shows the circuit diagram of the audio-play-back interface.

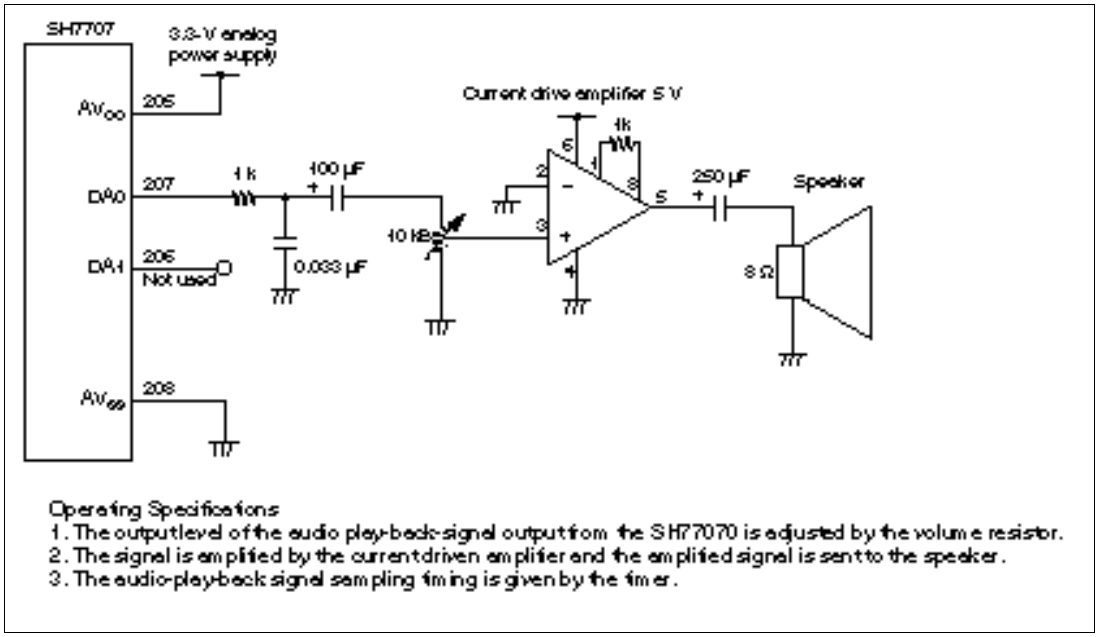


Figure 4.20 Circuit Diagram

4.5 Touch Panel Section

The SH7707 A/D converter is used for the touch panel interface to detect the position of pen input to the touch panel.

4.5.1 Features

Using the SH7707 A/D converter and I/O port interface reduces the number of external circuits. The touching of the pen is detected by the SH7707 as an interrupt.

4.5.2 Block Diagram

Figure 4.21 shows the block diagram of the touch-panel circuit.

- By changing the output of the PTJ1 and PTJ7 to high or low, the voltage is applied from X1 to X2, and Y1 to Y2. Since the pen input changes the output electric potential, the position of the XY direction is detected. (Refer to section 4.5.6, Touch Panel Mechanism).
- Three output ports are used: One for interrupt detection, and one each for X and Y direction electric potential detection switching.
- Two analog input channels are used to detect the XY direction position.
- One interrupt pin is used to detect an interrupt.

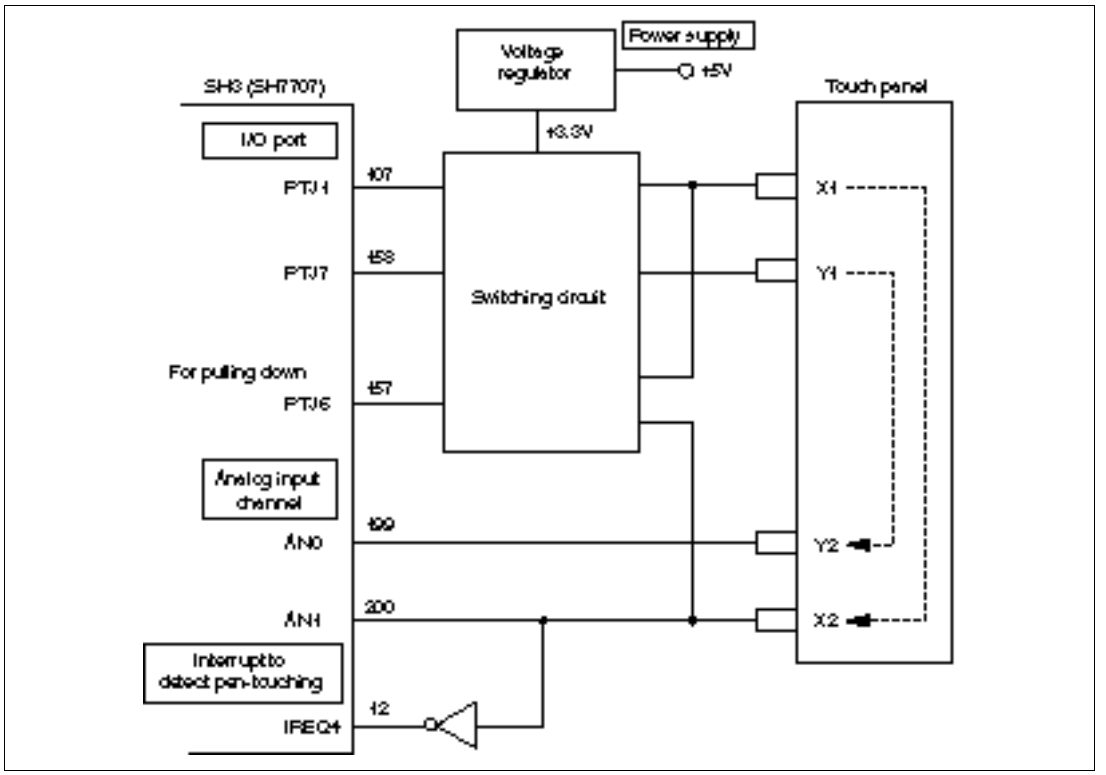


Figure 4.21 Block Diagram of the Touch Panel Circuits

4.5.3 Pin Function of the A/D Converter

The pin functions of the SH7707 A/D converter are listed in tables 4.21, 4.22, and 4.23.

(1) A/D Converter Interface Pin Function

Table 4.21 A/D Converter Interface Pin Function

Pin	SH7707 Pin No.	Input/Output	Function
AN0	199	Input	Analog input (X direction electric potential input)
AN1	200	Input	Analog input (Y direction electric potential input)
IRQ4	12	Input	Pen-input-interrupt input

(2) I/O Port Pin Function

Table 4.22 I/O Port Terminal Function

Pin	SH7707 Pin No.	Input/Output	Function
PTJ1	107	Output	Switching port output
PTJ7	158	Output	Switching port output
PTJ6	157	Output	Switching port output

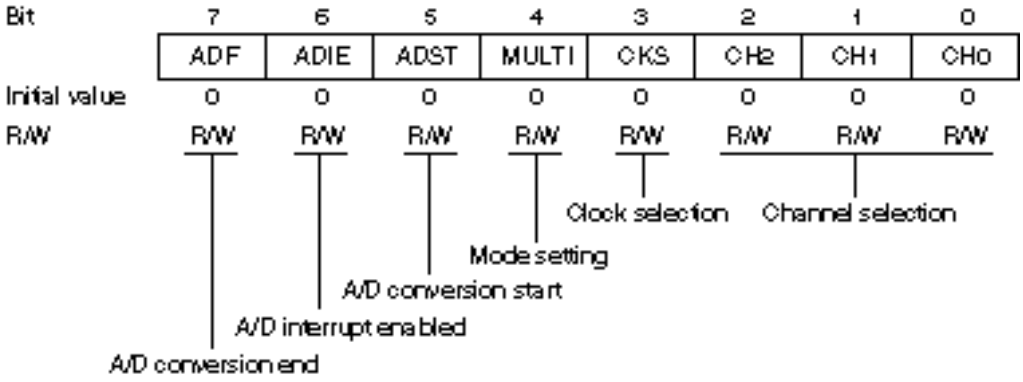
(3) Other Pin Function

Table 4.23 Other Terminal Functions

Pin	SH7707 Pin No.	Input/Output	Function
AV_{CC}	205	Input	Analog power supply voltage
AV_{SS}	198	Input	Analog ground, reference voltage

4.5.4 Registers for the A/D converter

(1) A/D Control/Status Register (ADCSR)



A/D control/status register (ADCSR) controls the start of the A/D conversion, channel selection, etc. Table 4.24 lists the details.

Table 4.24 ADCSR Description

Bit	Name	Function	Low	High	Default
7	ADF	Indicates A/D conversion end	*[1]	*[2]	Low
6	ADIE	Enables or disables A/D conversion end interrupt request	Enable	Disable	Low
5	ADST	Starts or ends A/D conversion	Stop	*[3]	Low
4	MULTI	Selects single mode or scan mode	Single mode	Scan mode	Low
3	CKS	Selects the time for A/D conversion	266 states	134 states	Low
2	CH2	Selects analog input channel			Low
1	CH1		*[4]	*[4]	Low
0	CH0				Low

is the set value of the reference platform.

Notes: 1. Clearing condition:

(1) When the ADF bit is 1, the ADF bit is read, and after that, writing 0 to the ADF bit clears the ADF bit.

(2) When the DRAM is initiated by the AD interrupt, or the ADDR is read, the ADF bit is cleared.

2. Set condition

In single mode: When the A/D conversion ends.

In scan mode: When the A/D conversion of all the selected channels ends.

3. In single mode: A/D conversion starts. After the conversion, the ADST is automatically cleared to 0.

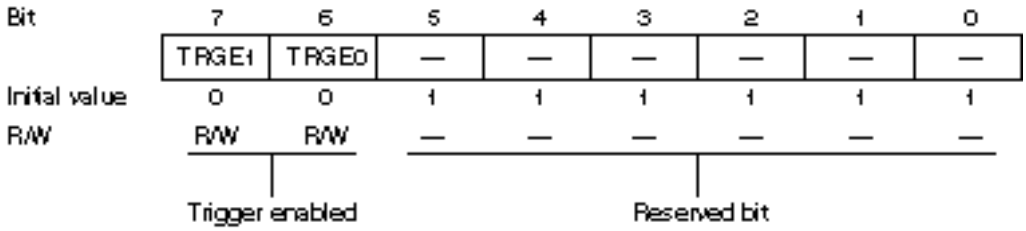
In scan mode: A/D conversion starts and continues. The A/D conversion is repeated between the selected channels until the ADST is cleared to 0 in software mode, reset mode, or standby mode.

4.

CH2	CH1	CH0	Single Mode	Scan Mode
0	0	0	AN0	AN0
0	0	1	AN1	AN0 and 1
0	1	0	AN2	AN0 to AN2
0	1	1	AN3	AN0 to AN3
1	0	0	AN4	AN4
1	0	1	AN5	AN4 and 5
1	1	0	AN6	AN4 to AN6
1	1	1	AN7	AN4 to AN7

indicates the set value of the reference platform.

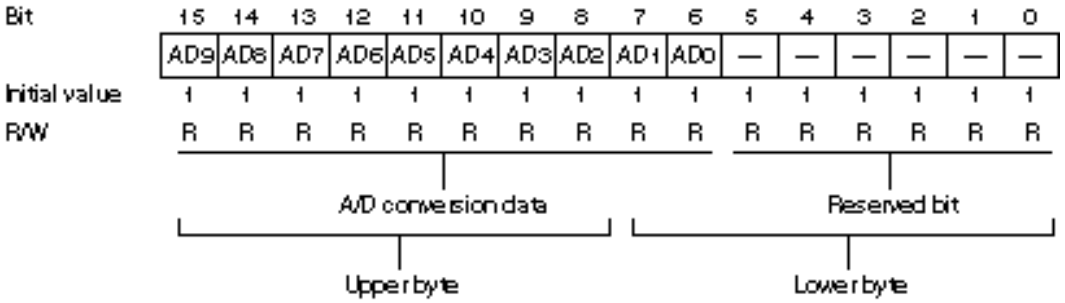
(2) ADCR (A/D Control Register)



Bit	Name	Description	Other than 11	At 11	Default
7 and 6	TRGE1, 0	A/D conversion external trigger is enabled or disabled	Disabled	*1	00

Note: Enabled: The A/D conversion starts at the falling edge of the input signal from the external trigger pin.

(3) ADDRA to ADDRD (A/D Data Register A to D)



Bit	Name	Description	Other than 11	At 11	Default
15 to 6	AD9 to 0	A/D conversion data	—	—	—

ADDR must be read in the order of upper bit then lower bit, using the word transfer instruction (such as MOV.W). Note that only the upper bit can be read, but the operation is not guaranteed when only the lower bit is read.

4.5.5 Circuit Diagram

Figure 4.22 shows the touch-panel circuit diagram. The regulator MM1180ZM is manufactured by Mitsumi Electric Co., Ltd., and touch panel HTC-A1 is manufactured by Nissha Printing Co., Ltd., and touch panel connector FH10A-8S-1SH is manufactured by Hirose Electric Co., Ltd.

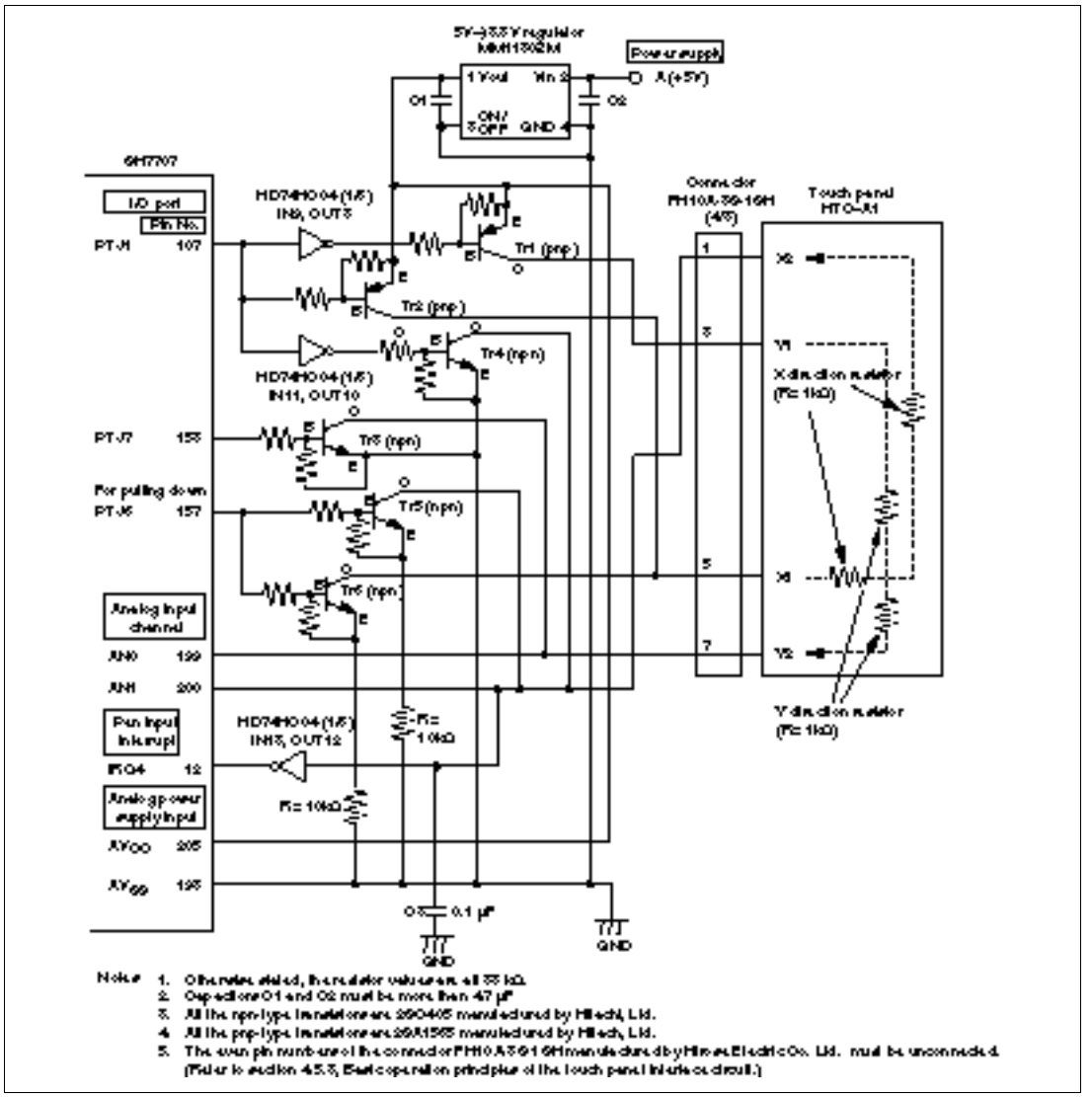


Figure 4.22 Circuit Diagram of the Touch Panel

4.5.6 Touch Panel Mechanism

In figure 4.23, the touch panel current usually flows in the direction from A to B, so the output Eo is 0 V. When a pen input occurs at point P, the touch panel current is divided in two routes; one current flows from point P to direction B and the other current flows from point P to direction C. The applied voltage is divided between the A-P route, and the P-B route. Because the partial voltage value is changed by the input point P, the input position P on the X axis can be detected by measuring the partial voltage (Eo). (The voltage drop between P and C can be ignored.) The position on the Y axis can also be detected by applying the voltage in the Y axis direction.

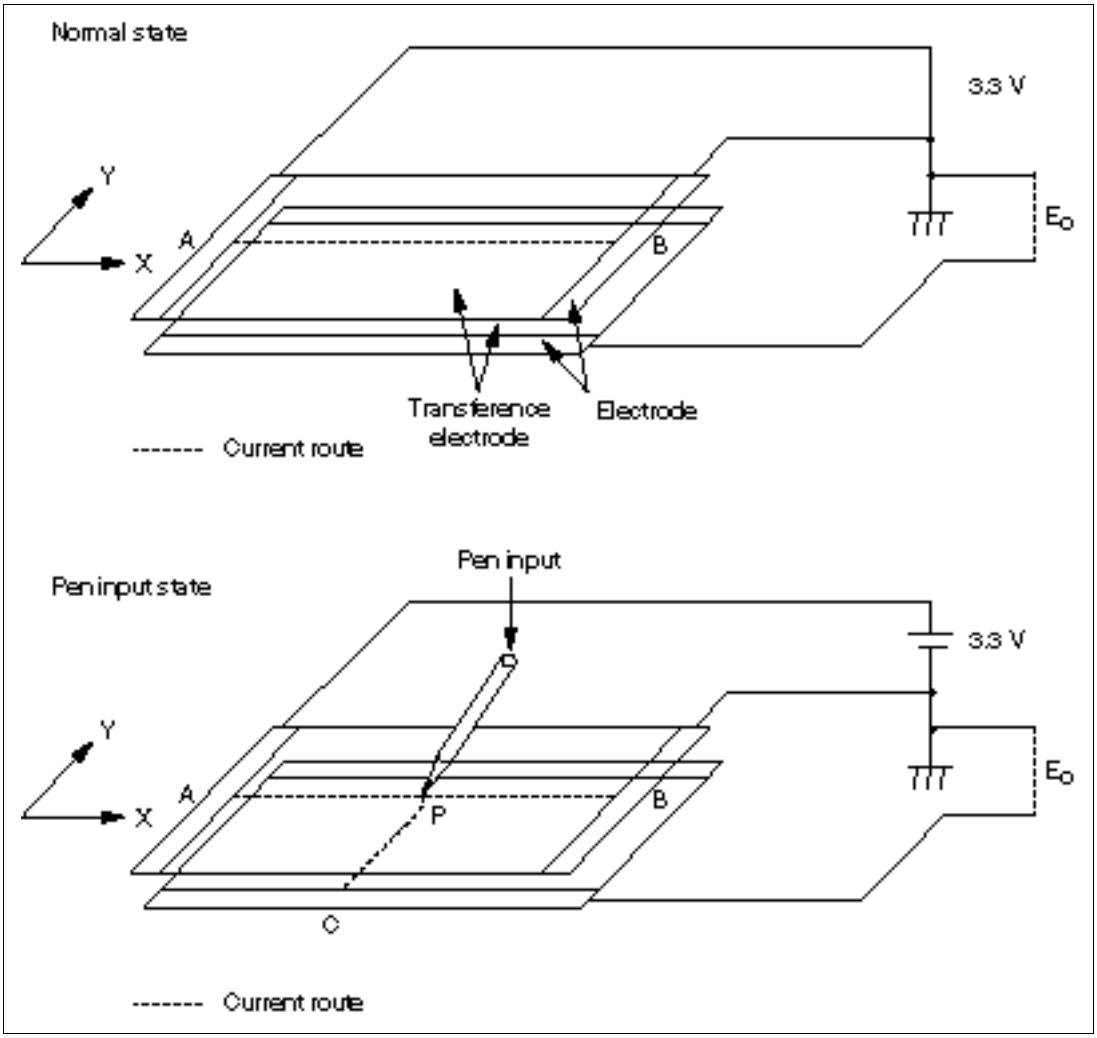


Figure 4.23 Mechanism of a Touch Panel

4.5.7 Operation of the A/D Converter

Figure 4.24 shows the A/D converter operation timing chart when single mode and channel 1 are set in the A/D control/status register (ADCSR).

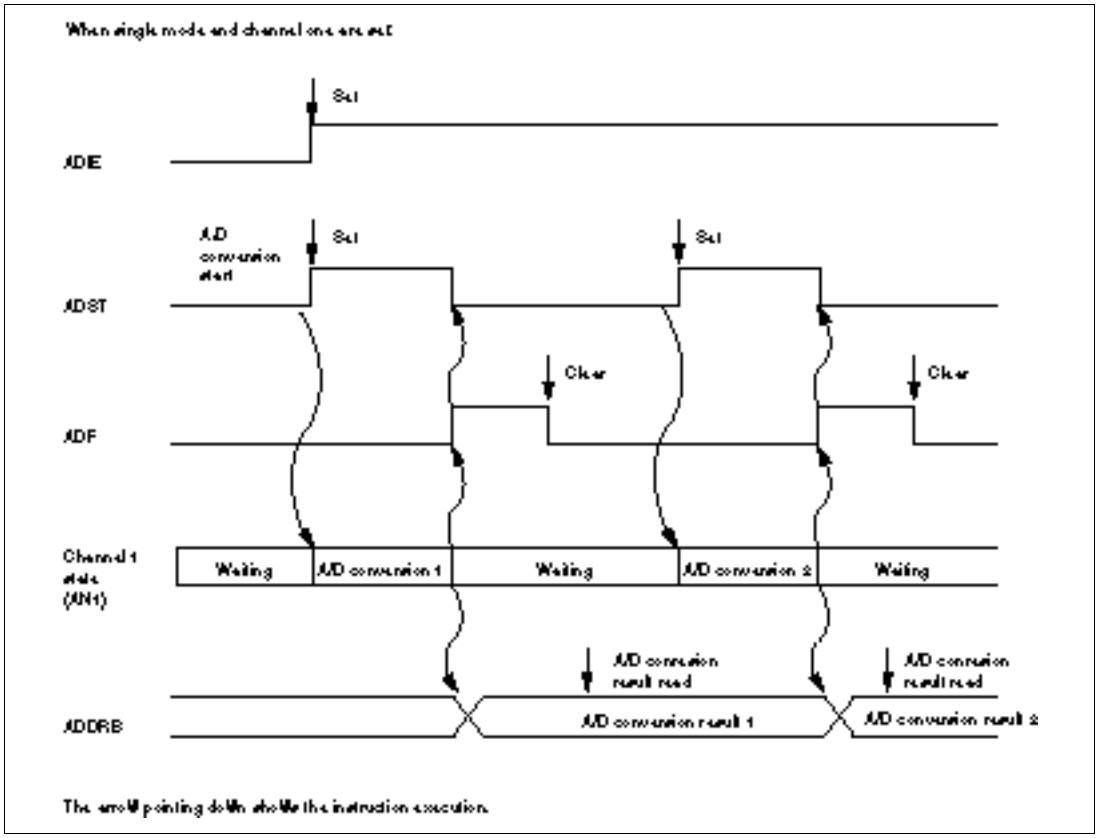


Figure 4.24 Operation Timing Chart

4.5.8 The Basic Operation Principle of the Touch Panel Interface Circuits

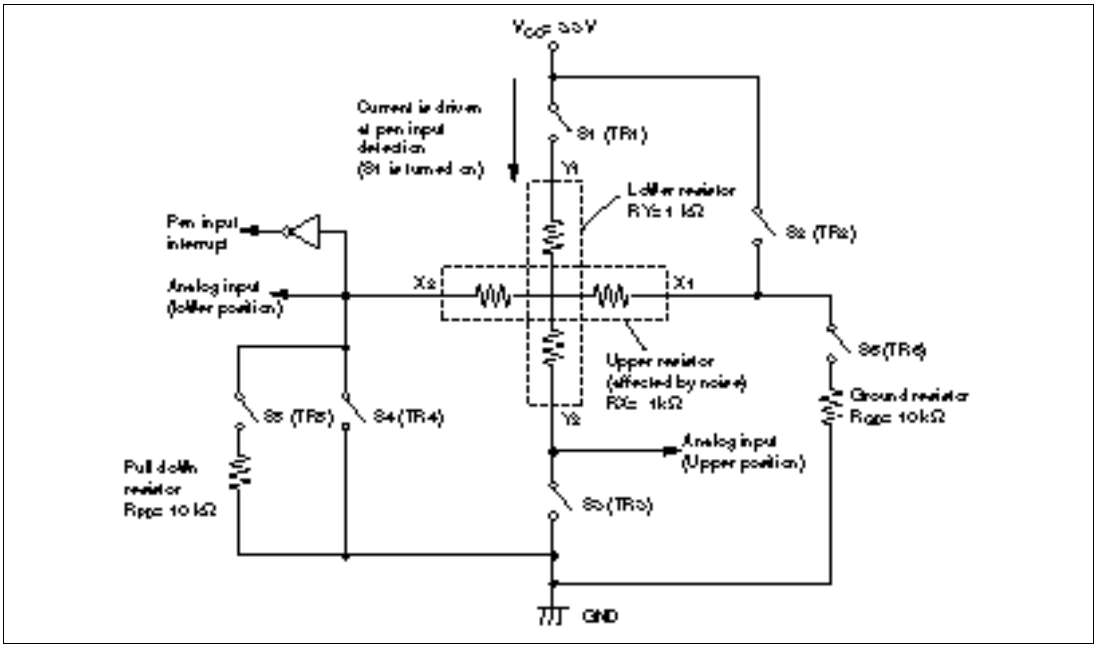


Figure 4.25 Basic Operation Principle of the Touch Panel Interface Circuits

The initial state of the touch panel can detect pen input as shown in figure 4.25. Since S1 is on, the voltage V_{CC} is applied on the lower-side resistor. When a pen is input, the upper resistor connects with the lower resistor. Therefore, electric potential appears at X2 in the figure through S5 and S6. The electric potential of X2 is the same value as the V_{CC} because the resistor value (RPD and RGD) between the S5, S6, and ground are large compared with the touch-panel resistor value. When a low pulse is input to the pen-input interrupt pin by the inverter in front of the pin, pen input is detected (low active).

When the pen input is detected, the operation moves to the measurement mode of the actual pen point position. The X direction position is detected when S2 and S4 in the figure are on. At this time, the voltage drops between X1 and X2, and voltage is divided at the pen point position. Using the partial voltage value, the pen point position is measured. The same procedure can be applied to the Y direction position. When S1 and S3 are on, the pen point position can be measured.

Table 4.25 lists the relations between the circuit state (on/off of each switch) and the port output.

Table 4.25 Relations between Circuit State (On/Off of Each Switch) and the Port Output

	PTJ1	PTJ7	PTJ6	S1	S2	S3	S4	S5	S6
Pen input detection (initial state)	H	L	H	ON	OFF	OFF	OFF	ON	ON
X direction position detection	L	L	L	OFF	ON	OFF	ON	OFF	OFF
Y direction position detection	H	H	L	ON	OFF	ON	OFF	OFF	OFF

4.5.9 The Relation between the Touch Panel Axes and the Output Voltage

Figure 4.26 shows the relation between the touch panel axes value and the output voltage when the touch panel axes value is determined as shown in figure 4.27.

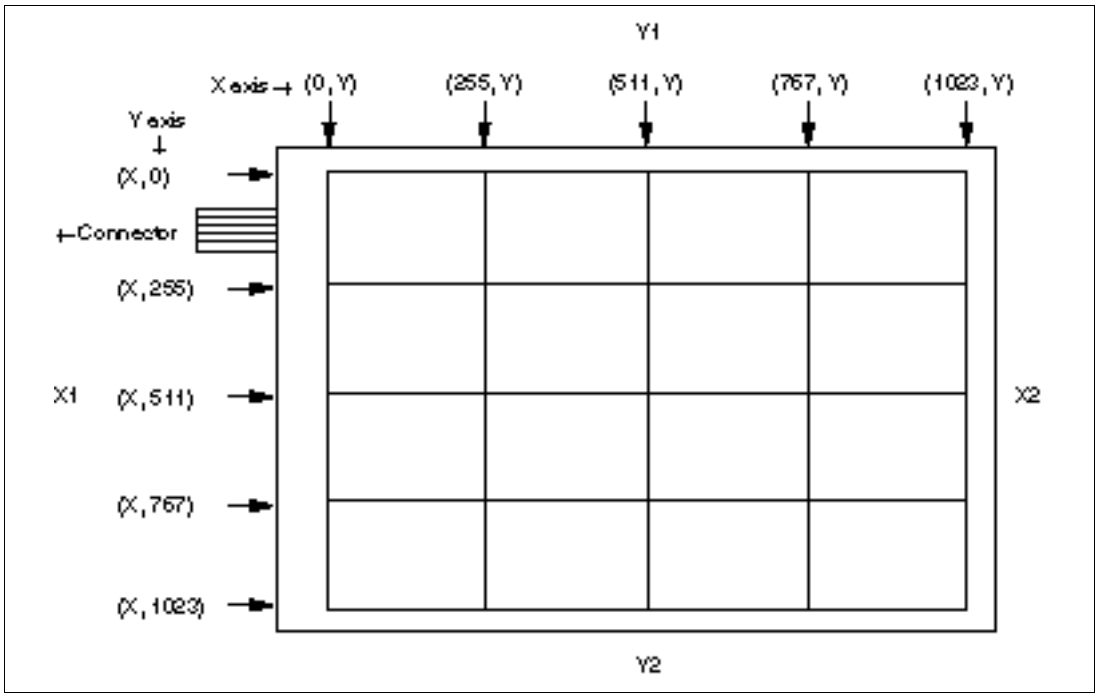


Figure 4.26 Axis Values on the Touch Panel

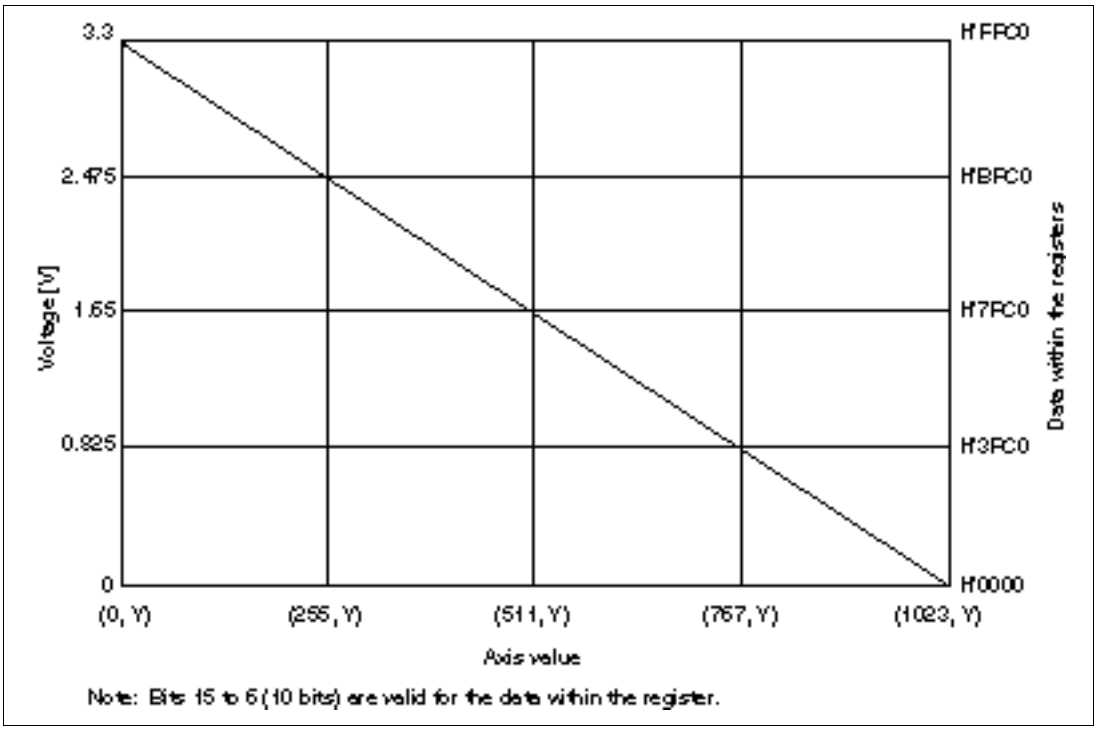


Figure 4.27 Relation between the Touch-Panel Axis Values and the Output Voltage

4.5.10 Canceling the Noise Using the LCD M Signal

Figure 4.28 shows how the noise caused by the LCD alternating current inversion signal (called M signal hereafter) is canceled. The SH7707 on-chip LCD controller periodically generates M signals. When the M signal is switched to and from high and low, the high frequency noise occurs, which affects the touch panel sampling data. Therefore, noise must be cancelled, in the following procedures.

First, the divider circuit in the FPGA described below is used to generate the delay signal of the M signal (called M_D signal hereafter). Then, the M signal and M_D signal are exclusively OR-ed. A/D interrupt is enabled while the resultant signal is low, and A/D interrupt is disabled while the resultant signal is high. In the period where the noise occurs, the A/D conversion is terminated. The M_D signal delay time can be changed by the FPGA divider circuit setting.

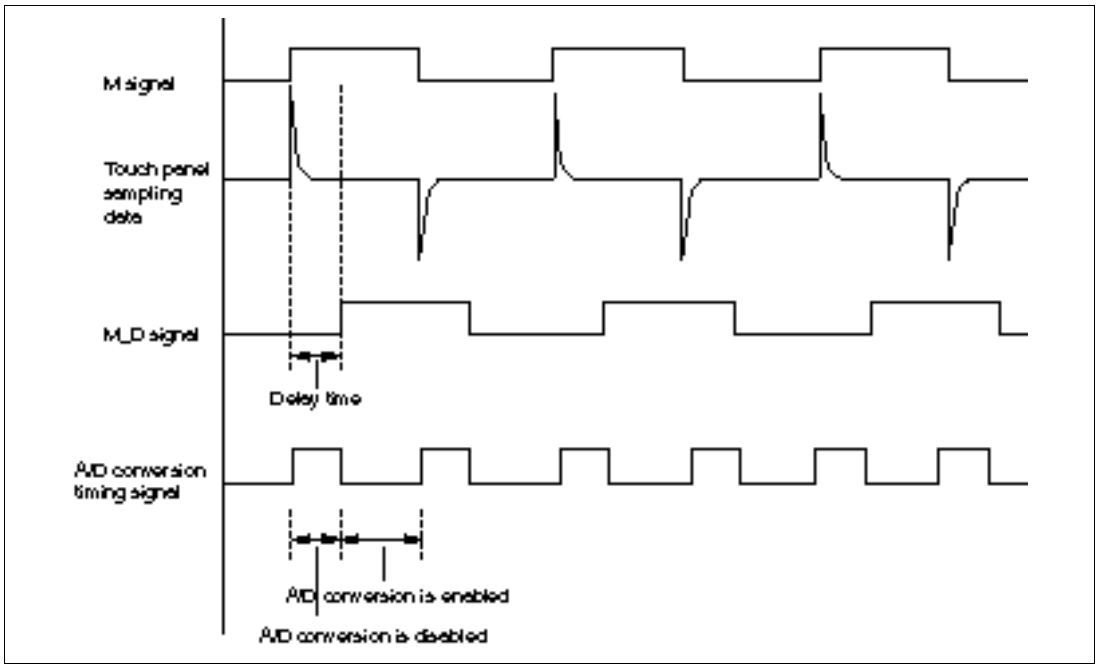


Figure 4.28 Creating the A/D Conversion Timing Signal

4.6 SCI Section

SH7707 has the serial communication interface (SCIF) with FIFO. SCIF enables serial communication by two methods: asynchronous communication and synchronous communication. In the reference platform, asynchronous communication is adopted. For the RS-232C driver/receiver, LT1330CG manufactured by Linear Technology Corporation is used.

4.6.1 Interface Block Diagram

Figure 4.29 shows the SCIF block diagram.

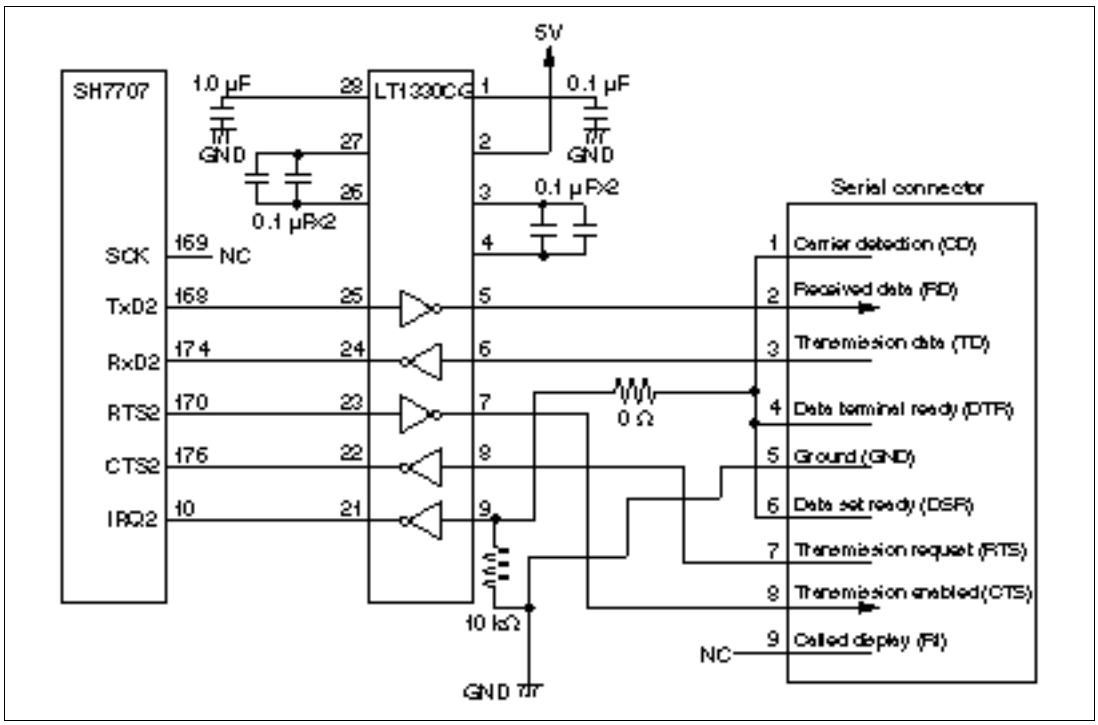


Figure 4.29 SCIF Block Diagram

4.6.2 Pin Configuration

Table 4.26 lists the SCIF pin configuration.

Table 4.26 SCI Pin Function

Pin	Signal	Input/Output	Function	SH Pin No.
Receive data pin	RxD2	Input	Inputs receive data	174
Transmit data pin	TxD2	Output	Outputs send data	168
Transmit request pin	PTS2	Output	Requests transmit	170
Transmit enable pin	CTS2	Input	Enables transmit	176
Interrupt pin	IREQ2	Input	Receives CD signal	10

4.6.3 Register Configuration

This interface has the SCIF mode register, the input/output data register, and the control register as the internal register. Table 4.27 lists the register configuration.

Table 4.27 Register Configuration

Register	Abbreviation	R/W	Initial Value	Address	Access Size
Serial mode register	SCSMR2	R/W	H'00	H'A4000150	8 bits
Bit rate register	SCBRR2	R/W	H'FF	H'A4000152	8 bits
Serial control register	SCSCR2	R/W	H'00	H'A4000154	8 bits
Transmit data register	SCFTDR2	W	—	H'A4000156	8 bits
Serial status register	SCSSR2	R/(W)	H'0060	H'A4000158	16 bits
Receive data register	SCFRDR2	R	Undefined	H'A400015A	8 bits
FIFO control register	SCFCR2	R/W	H'00	H'A400015C	8 bits
FIFO number data register	SCFDR2	R	H'0000	H'A400015E	16 bits

(1) Serial Mode Register (SCSMR2)

Sets the SCIF serial communication format and selects the clock source of the baud rate generator.

Bit	7	6	5	4	3	2	1	0
	0	CHR	PE	O/E	STOP	0	CKS1	CKS0
Initial value	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R/W	R	R/W	R/W

Bit 6—Character Length (CHR): Selects either 7 or 8 bits as the asynchronous mode data length.

Bit 6

CHR	Description
0	Eight-bit data
1	Seven-bit data (MSB of SCTDR2 is not transmitted.)

is the set value of the Reference platform.

Bit 5—Parity Enable (PE): Determines whether to add the parity bit at transmission and checks parity bit at receiving.

Bit 5

PE	Description
0	Does not add parity bit
1	Adds and checks parity bit

Bit 4—Parity Mode (O/E): Selects even or odd parity. Valid only when the PE bit is 1.

Bit 4

O/E	Description
0	Odd parity
1	Even parity

Bit 3—Stop Bit Length (STOP): Selects the length of stop bit.

Bit 3	
STOP	**Description**
0	One stop bit
1	Two stop bits

is the set value of the Reference platform.

Bits 1 and 0—Clock Select (CKS1, CKS0): Selects internal baud rate generator clock source. By the CKS1 and CKS0 bit settings, one of the P ø, P ø/4, P ø/16, or P ø/64 clock source can be selected.

Bit 1	Bit 0	
CKS1	**CKS0**	**Description**
0	0	P ø clock
0	1	P ø/4 clock
1	0	P ø/16 clock
1	1	P ø/64 clock

P ø: Peripheral clock

is the set value of the reference platform.

(2) Bit Rate Register (SCBRR2)

The bit rate register sets the serial transmission/receiving bit rate together with the operation clock that is selected by the CKS1 and CKS0 bits of the serial mode register (SCSMR2). The bit rate differs in accordance with the source clock even when the value is the same. (For the relationship between the source clock and the bit rate, refer to the SH7707 hardware manual.)

Bit	7	6	5	4	3	2	1	0
Initial value	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

(3) Serial Control Register (SCSCR2)

The serial control register operates SCI transmission/receiving, enables/disables interrupt request, and selects transmit and receive clock source.

Bit	7	6	5	4	3	2	1	0
	TIE	RIE	TE	RE	0	0	CKE1	CKE0
Initial value	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7—Transmit Interrupt Enable (TIE): Enables/disables the generation of the transmit data empty interrupt (TXI) request when the serial transmit data is sent from the transmit data register (SCTDR2) to the transmit shift register (SCTSR) and TDRE flag of the serial status register (SCSSR2) is set to 1.

Bit 7

TIE	Description	
0	Disables transmit data empty interrupt (TXI) request	(Initial value)
1	Enables transmit data empty interrupt (TXI) request	

Bit 6—Receive Interrupt Enable (RIE): Enables/disables receive data full interrupt (RXI) request and receive error interrupt (ERI) request when the data is transferred from receive shift register (SCRSR) to receive data register (SCRDR) and the SCSSR RDRF flag is set to 1.

Bit 6

RIE	Description
0	Disables receive data full interrupt (RXI) request and receive error interrupt (ERI).*
1	Disables receive data full interrupt (RXI) request and receive error interrupt (ERI).*

Note: RXI and ERI interrupt requests can be cleared by reading the RDRF flag or error flag (FER, PER, or ORER) after it has been set to 1, then clearing the flag to 0, or by clearing RIE to 0.

Bit 5—Transmit Enable (TE): Enables/disables start of the SCIF serial transmit operation.

Bit 5

TE	Description
0	Disables transmission.
1	Enables transmission.

is the set value of the reference platform.

Bit 4—Receive Enable (RE): Enables/disables start of the SCIF serial receive operation.

Bit 4	
RE	**Description**
0	Disables receiving
1	Enables receiving

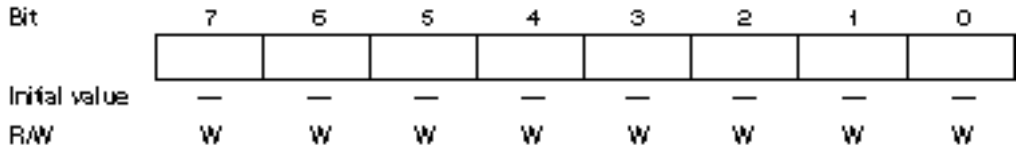
Bit 1 and 0—Clock Enable (CKE1 and CKE0): Selects SCI clock source, and enables/disables clock output from the SCK pin. Whether to use the SCK pin for the serial clock output pin or the serial clock input pin depends on the combination of the CKE1 bit and the CKE0 bit. Note that the CKE0 bit is valid only when the internal clock (CKE1 = 0) is operating.

Bit 1	Bit 0		
CKE1	**CKE0**	**Description**	
0	0	Asynchronous mode	Internal clock/SCK pin is the input pin (input signals are ignored) (Initial value)
0	1	Asynchronous mode	Internal clock/SCK pin is the clock output (Same frequency as the bit rate)
1	0	Asynchronous mode	External clock/SCK pin is the clock input (clock of 16 times the frequency of the bit rate is the input)
1	1	Asynchronous mode	External clock/SCK pin clock input (clock of 16 times the frequency of the bit rate is input)

is the set value of the reference platform.

(4) Transmit Data Register (SCFTDR2)

The transmit data register is an 8-bit register that stores data to be serially transmitted.



(5) Serial Status Register (SCSSR2)

Serial status register shows the operating status of the SCIF.

Bit	7	6	5	4	3	2	1	0
	ER	TEND	TDFE	BRK	FER	PER	RDF	DR
Initial value	0	1	1	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W

Bit 7—Parity Error (ER): Indicates that the parity error occurred at receiving when the parity was added and the operation has terminated abnormally.

Bit 7

ER	Description
0	Indicates that it is currently receiving, or that the receiving was successful (Initial value)
1	Indicates that parity error occurred at receiving

Bit 6—Transmit End (TEND): Indicates that transmission ended without a valid data in SCFTDR2 when transmitting the transmission character end bit. This bit is for read-only. It cannot be written to.

Bit 6

TEND	Description	
0	Currently transmitting. When 0 is written to the TDFE bit, it becomes 0.	
1	Transmission completed.	(Initial value)

is the set value of the reference platform.

Bit 5—Transmit FIFO Data Empty (TDFE): Data is transferred from the transmit FIFO data register (SCFTDR2) to the transmit shift register (SCTSR), and the next serial transmit data can be written to the SCFTDR2.

Bit 5

TDFE	Description	
0	Valid transmission data is written to SCFTDR2.	
1	Indicates that there are no valid transmission data in SCFTDR2	(Initial value)

Bit 4—Break Detection (BRK): The BRK bit detects the break signal.

Bit 4

BRK	Description	
0	No break signal is detected	(Initial value)
1	Break signal is detected	

Bit 3—Framing Error (FER): The FER indicates that a framing error occurred at receiving and the operation was abnormally terminated.

Bit 3

FER	Description
0	Indicates that it is currently receiving, or the receiving is successfully completed.
1	Indicates that the framing error occurred at receiving

Bit 2—Parity Error (PER): The PER indicates that the parity error occurred when the parity is added at receiving, and the receiving is terminated abnormally.

Bit 2

PER	Description
0	Indicates that it is currently receiving, or the receiving is successfully completed (Initial value)
1	Indicates that the parity error occurred during receiving

is the set value of the reference platform.

Bit 1—Receive FIFO Data Full (RDF): The RDF indicates that the received data is stored in the receive FIFO data register (SCFRDR2).

Bit 1

RDF	Description	
0	No valid data is stored in SCFRDR2	(Initial value)
1	A valid data is stored in SCFRDR2.	

Bit 0—Receive Data Ready (DR): The DR indicates that the received data is stored in the receive FIFO data register (SFCRDR2)

Bit 0

DR	Description
0	During receiving, or after normal receiving, there is no data left.
1	The next data is not received.

is the set value of the reference platform.

(6) Receive Data Register (SCFRDR2)

The receive data register (SCFRDR2) stores the received serial data. Continuous receive operation is enabled because it is a 16-byte FIFO register. SCFRDR2 is a read-only register and therefore cannot be written to from the CPU.

Bit	7	6	5	4	3	2	1	0
Initial value	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

4.7 IrDA Section

The SH7707 has the IrDA interface functions conforming to the IR-SIR version 1.0, and can transmit data at 115.2 kbps. Its features are described below.

4.7.1 Features

(1) Conforms to IR-SIR version 1.0
(2) Asynchronous serial communication
— Data length: 8 bits
— Stop bit: 1 bit
— Parity bit: None
(3) Includes transmit/receive FIFO buffer
(4) Includes baud rate generator

4.7.2 IrDA Interface Block Diagram

Figure 4.30 shows the IrDA interface block diagram.

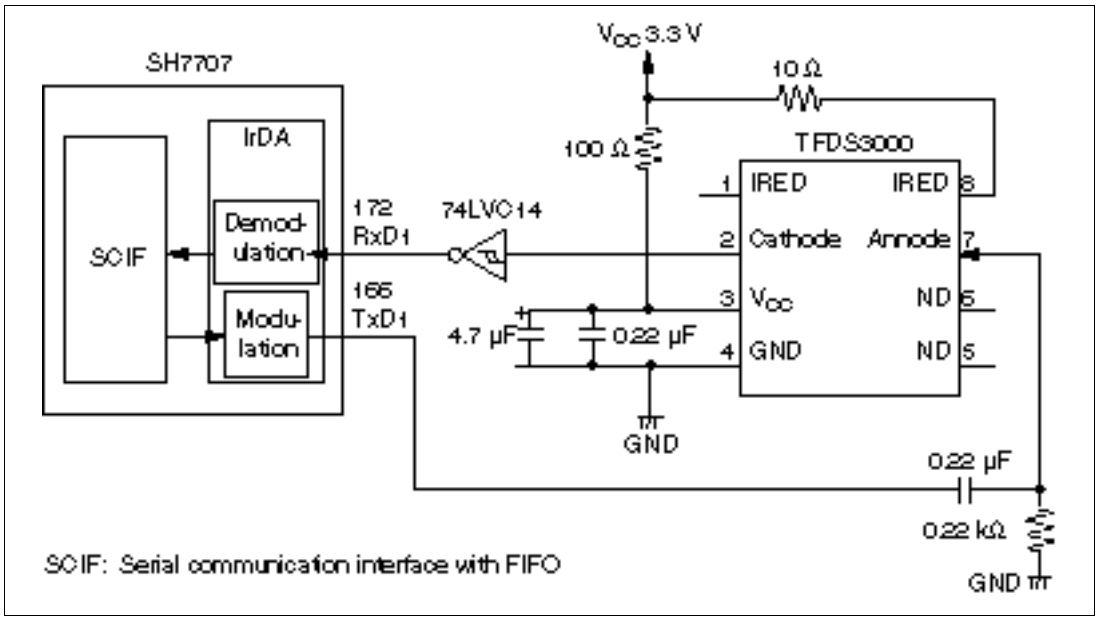


Figure 4.30 IrDA Interface Block Diagram

4.7.3 Pin Configuration

Table 4.28 lists the SH7707 pin configuration, and table 4.29 lists the TDFS3000 pin configuration.

Table 4.28 SH7707 Pin Configuration

Pin Name	Signal Name	Function
Receive data pin	RxD1	Receive data input
Transmit data pin	TxD1	Transmit data output

Table 4.29 TFDS3000 Pin Configuration

	Pin Name	Function
1	IRED cathode	Connected to IRED cathode and driver transistor
2	Rx	Receive data
3	V_{CC}	Power supply of 3.3 V
4	GND	Ground
5	NC	Not connected
6	NC	Not connected
7	Tx	Data transmission
8	IRED anode	IRED anode

4.7.4 Register Configuration

The interface has the SCIF mode register, the input/output data register, and the control register. Table 4.30 lists the register configuration.

Table 4.30 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Serial mode register 1	SCSMR1	R/W	H'00	H'A4000140	8 bits
Bit rate register 1	SCBRR1	R/W	H'FF	H'A4000142	8 bits
Serial control register 1	SCSCR1	R/W	H'00	H'A4000144	8 bits
FIFO transmit data register 1	SCFTDR1	W	—	H'A4000146	8 bits
Serial status register 1	SCSSR1	R/(W)	H'0060	H'A4000148	16 bits
FIFO receive data register 1	SCFRDR1	R	Undefined	H'A400014A	8 bits
FIFO control register 1	SCFCR1	R/W	H'00	H'A400014C	8 bits
FIFO data count register 1	SCFDR1	R	H'0000	H'A400014E	16 bits

4.7.5 Description of Each Register

The register configuration is the same as the SCIF register configuration except for the serial mode register (SCSMR1). For registers other than the serial mode register, refer to the SCIF register.

Serial Mode Register (SCSMR1)

Bit	7	6	5	4	3	2	1	0
	IRMOD	ICK3	ICK2	ICK1	ICK0	PSEL	CKS1	CKS0
Initial value	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7—IrDA Mode (IRMOD)

Bit 7

IRMOD	Function
0	SCIF mode
1	IrDA mode

Bits 6 to 3—IrDA Clock Selection Bit (ICK3 to ICK0)

Bit 2—Pulse Output Selection Bit (PSEL)

Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Function
ICK3	ICK2	ICK1	ICK0	PSEL	Pulse width: 3/16 of 115 kbps bit length
ICK3	ICK2	ICK1	ICK0	1	
X	X	X	X	0	Pulse width: 3/16 of bit length

Bits 1 to 0—Clock Selection 1 to 0 (CKS1 to CKS0): The internal baud rate generator clock can be set from PΦ, PΦ/4, PΦ/16, or PΦ/64.

Bit1	Bit 0	
CKS1	CKS0	Function
0	0	PΦ clock (Initial value)
0	1	PΦ/4 clock
1	0	PΦ/16 clock
1	1	PΦ/64 clock

is the set value of the reference platform.

4.7.6 Operation Description

Using the IrDA module, the IR communication is enabled with the light emission and the light receiving unit that conforms to the IR-SIR version 1.0. The serial communication interface unit has a 16-stage FIFO buffer for transmission and receiving. DMA transmission is supported.

(1) Transmitter

Serial output data (UART frame) is output from the SCIF. This waveform is modified to be the IR frame serial output data by the IrDA module. Figure 4.31 shows the transmit/receive data frame structure.

(2) Receiver

The 3/16 bit-width waveform of the received IR frame serial output data is converted to the serial output data (UART frame) after demodulation. When demodulated to 0, the waveform is output, and when modulated to 1, the waveform is not output.

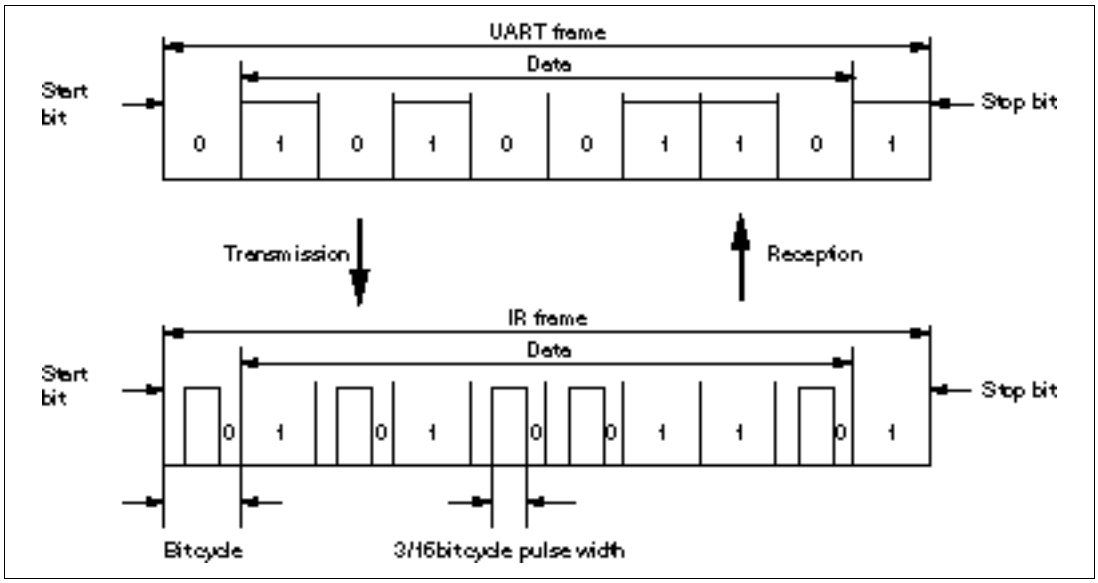


Figure 4.31 Transmit/Receive Data Frame Structure

4.8 FPGA Section

SH7707 includes the peripheral functions necessary for the Windows® CE. Note, however, that the reference platform uses the external bus as the 32-bit bus width, and the general I/O ports necessary for key scanning by the keyboard section are installed in the external FPGA. This FPGA is not necessary for the users who are using the external bus width of 16 bits.

The reference platform uses one FPGA: EPM7160EQC100-10 manufactured by Altera Corporation. The FPGA has the 8-bit timer (2 channels), the key scan control, and a touch-panel sampling timer. Table 4.31 shows the FPGA specifications.

Table 4.31 FPGA Specifications

Product Name	EPM7160EQC100-10
Manufacturer	MAX7000 series manufactured by ALTERA
Package	PQFP (plastic quad flat package) of 100 pins, –10 speed grade
Power voltage	5 V or 3.3 V
Usable number of gates	3200 gates (About 2540 gates are used)
Number of macro cells	160 cells (127 cells are used)
Number of usable I/O	84

4.8.1 Features

(1) CPU interface
- — Can be directly connected to the SH7707 bus interface
- — 8-bit data bus

(2) 8-bit reload timer
- — Includes 8-bit down counter (counts down clocks)
- — Includes 8-bit reload register
- — Can start and stop timers
- — Can output interrupt requests
- — Outputs toggle-type underflow

(3) Key scan control
- — Maximum of 80 keys with eight output ports and 10 input ports can be scanned
- — Supports key input interrupt
- — After key input interrupt, key scanning is provided by software

(4) Touch-panel sampling timer
- — Outputs interrupt request that is delayed LCD alternating current inversion signal
- — Can also be used as a 4-bit reload timer

4.8.2 Register Address Map and IRQ Allocation

Table 4.32 lists the register address map.

Table 4.32 Register Address Map

Module	Address	Register	Abbreviation
Timer 0	H'10000000	Timer 0 control counter	TMCR0
	H'10000004	Timer 0 count down register	TMDCR0
	H'10000008	Timer 0 reload data register	TMRCR0
Timer 1	H'10000010	Timer 1 control register	TMCR1
	H'10000014	Timer 1 count down register	TMDCR1
	H'10000018	Timer 1 reload data register	TMRDR1
Key scan control	H'10000020	Key scan control/input high register	KSCIHR
	H'10000024	Key scan input register	KSILR
	H'10000028	Key scan output register	KSOR
Touch-panel sampling timer	H'10000030	Sampling timer control register	STMCR0
	H'10000034	Sampling timer count down register	STMDCR0
	H'10000038	Sampling timer reload data register	STMRDR0

Table 4.33 shows the configuration of interrupts that are output from FPGA. The FPGA integrates the following four types of interrupt source to one IRQ signal, and outputs to the SH7707. Each source register must be analyzed to determine the interrupt source.

Table 4.33 IRQ Allocation

Module	IRQ
Timer 0	SHIRQ
Timer 1	
Key scan control	
Sampling timer control	

4.8.3 Timer

The FPGA has the 2-channel 8-bit reload timer. Timer 0 is used as the key scan timer after a key is input to the keyboard section. Timer 1 is used as the DMA transfer trigger for sound.

(1) Features

- Clock can be selected (/16, /64, /256, /1024)
- Includes an 8-bit down counter
- Includes an 8-bit reload data register
- Timer can be started or stopped
- Interrupt request can be output
- Outputs a toggle-type underflow

(2) Register Configuration

Table 4.34 shows the register configuration.

Table 4.34 Register Configuration

Module Name	Address	Register Name	Abbreviation
Timer 0	H'10000000	Timer 0 control register	TMCR0
	H'10000004	Timer 0 count down register	TMDCR0
	H'10000008	Timer 0 reload data register	TMRCR0
Timer 1	H'10000010	Timer 1 control register	TMCR1
	H'10000014	Timer 1 count down register	TMDCR1
	H'10000018	Timer 1 reload data register	TMRDR1

(3) Register Description

(a) Timer Control Register

Timer control register (TMCR) controls counter, timer output, and interrupts.

Bit	7	6	5	4	3	2	1	0
	UNDIE	UNDF	—	—	TMST	TMOE	TMPS1	TMPS0
Initial value	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7—Underflow Interrupt Enable (UNDIE): Controls whether to enable interrupt occurrence when the flag indicating the TMDCR underflow is set.

Bit 7

UNDIE	Description	
0	Disables interrupt by the UNDIE	(Initial value)
1	Enables interrupt by the UNDIE	

Bit 6—Underflow Flag (UNDF): UNDF is a flag that indicates that the TMDCR underflow occurred.

Bit 6

UNDF	Description	
0	TMDCR indicates no underflow occurrence. Clearing condition: When 0 is written to UNDF.	(Initial value)
1	Enables interrupt by UNDF. Setting condition: When TMDCR indicates the underflow occurrence.	

Bit 3—Timer Start/Stop (TMST): Selects whether or not to operate timer down counter.

Bit 3

TMST	Description	
0	Terminates the TMDCR count operation.	(Initial value)
1	Starts the TMDCR count operation.	

Bit 2—Timer Output Enable (TMOE): Selects whether to output timer signal.

Bit 2

TMST	Description	
0	Does not output timer signal. Low level is output for timer output	(initial value)
1	Outputs timer signal. Output is toggled for each TMDCR underflow occurrence.	

Bit 1 and 0—Timer Prescaler: Selects TMDCR clock to be counted.

Bit 1	Bit 0	
TMPS1	**TMPS0**	**Description**
0	0	Counts at CLK/16 (default)
0	1	Counts at CLK/64
0	0	Counts at CLK/256
1	1	Counts at CLK/1024

(b) Timer count down register

Timer count down register (TMDCR) is an 8-bit counter, and counts backwards by the input clock. If an underflow occurs due to counting backwards, the UNDF is set.

Bit	7	6	5	4	3	2	1	0
Initial value	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

(c) Timer reload data register

Timer reload data register (TMRDR) specifies the value to be set at TMDCR when the TMDCR underflow occurs. At reset, all bits become 0.

Bit	7	6	5	4	3	2	1	0
Initial value	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

4.8.4 Key Scan Control

The FPGA has the input/output port and the interrupt control circuits for key scanning in the keyboard section.

(1) Features

- Maximum of 80 keys with 8 output ports and 10 input ports can be scanned
- Supports key input interrupt
- Key scan after key input interrupt is supported by software

(2) Register Configuration

Table 4.35 shows the register configuration.

Table 4.35 Register Configuration

Module	Address	Register Name	Abbreviation
Key scan control	H'10000020	Key scan control/input high register	KSCIHR
	H'10000024	Key scan input register	KSILR
	H'10000028	Key scan output register	KSOR

(3) Register Description

(a) Key scan input high/control register

Indicates the input level of the key scan input port. Interrupt output is controlled at key input detection.

Bit	7	6	5	4	3	2	1	0
	KSIE	KSIF	—	—	—	—	KSID9	KSID8
Initial value	0	0	—	—	—	—	—	—
R/W	R/W	R/W	—	—	—	—	R	R

Bit 7—Key Scan Interrupt Enable (KSIE): Selects whether to operate the timer down counter (TMDCR).

Bit 7

KSIE	Description	
0	Disables key scan interrupt	(Initial value)
1	Enables key scan interrupt	

Bit 6—Key Scan Interrupt Request (KSIF): Selects whether to operate the timer down counter (TMDCR)

Bit 6

KSIF	Description
0	No key input is detected. Clearing condition: When 0 is written to KSIF. (Initial value)
1	Key input is detected. Setting condition: When a key input is detected.

(b) Key scan low input register (KSILR)

Indicates the input level of the key scan input port. Controls interrupt output of key input detection.

Bit	7	6	5	4	3	2	1	0
	KSID7	KSID6	KSID5	KSID4	KSID3	KSID2	KSID1	KSID0
Initial value	—	—	—	—	—	—	—	—
R/W	R	R	R	R	R	R	R	R

Bits 7 to 0—Key Scan Input Data (KSID)

Bits 7 to 0

KSID7 to 0	Description
0	Input port is low level
1	Input port is high level

(c) Key scan output data (KSODR)

Specifies the output level of the key scan output port.

Bit	7	6	5	4	3	2	1	0
	KSOD7	KSOD6	KSOD5	KSOD4	KSOD3	KSOD2	KSOD1	KSOD0
Initial value	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bits 7 to 0—Key Scan Output Data (KSOD)

Bits 7 to 0

KS0D7 to 0	Description
0	Output port is low level
1	Output port is high level

4.8.5 Touch-Panel Sampling Timer

The FPGA has the timer to avoid noise that is caused by the LCD alternating current inversion signal. The noise affects the touch panel.

(1) Features

- Enables interrupt output of the signal in which the LCD alternating inversion
- Can be used as a 4-bit reload timer.
- Delay time of up to 512 ms can be set. (at CLK = 32 MHz)

(2) Register Configuration

Table 4.36 shows the register configuration.

Table 4.36 Register Configuration

Module	Address	Register Name	Abbreviation
Digitizer sampling timer	H'10000030	Sampling timer control register	STMCR0
	H'10000034	Sampling timer count down register	STMDCR0
	H'10000038	Sampling timer reload data register	STMRDR0

(3) Register Description

(a) Sampling timer control register (STMCR)

Sampling timer control register controls the counter, the timer output, and the interrupts.

Bit	7	6	5	4	3	2	1	0
	UNDIE	UNDF	—	STMD	STMST	STMOE	STMPS1	STMPS0
Inital value	0	0	—	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7—Underflow Interrupt Enable (UNDIE): Controls whether to enable the interrupt occurrence when flag UNDF is set indicating the TMDCR underflow occurrence.

Bit 7

UNDIE	Description	
0	Disables interrupt by the UNDF	(Initial value)
1	Enables interrupt by the UNDF	

Bit 6—Underflow Flag (UNDF): Indicates the STMDCR underflow occurrence.

Bit 6

UNDF	Description	
0	Underflow has not occured in the STMDCR.	(Initial value)
	Clearing condition: When 0 is written to UNDF	
1	Setting condition: When underflow occurs in STMDCR	

Bit 4—Timer Mode (STMD): Selects timer mode.

Bit 4

STMD	Description	
0	Delay timer	(Initial value)
1	Reload timer	

Bit 3—Timer Start/Stop (STMST): Determines whether to operate or stop timer down counter (TMDCR).

Bit 3

STMST	Description	
0	Terminates STMDCR count operation.	(Initial value)
1	Operates STMDCR count.	

Bit 2—Timer Output Enable (STMOE): Selects whether to perform output timer.

Bit 2

STMOE	Description	
0	Outputs no timer output.	(Initial value)
	Timer output is low.	
1	Outputs timer output. In delay timer mode, outputs the delayed M signal, and in reload timer mode, outputs toggle output.	

Bits 1 and 0—Timer Prescaler 1 and 0: Selects STMDCR count clock.

Bit 1	Bit 0		
STMPS1	**STMPS0**	**Description**	
0	0	Counts by CLK/16	(Initial value)
0	1	Counts by CLK/64	
1	0	Counts by CLK/256	
1	1	Counts by CLK/1024	

(b) Sampling timer count down register (STMDCR)

The sampling timer count down register is a 4-bit register and counts backwards by the input clock. If underflow occurs due to the count, UNDF is set.

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

(c) Sampling timer reload data register (STMRDR)

Sampling timer reload data register specifies the value set to STMDCR. At reset, all bits become 0.

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

(4) Sampling Timer Operation Description

(a) Delay timer mode

When delay timer is specified, the LCD alternating current inversion signal (called M signal hereafter) is delayed for a specified time and is output to the timer output. Interrupt request is enabled. Figure 4.32 shows the relationship between the M signal and timer output.

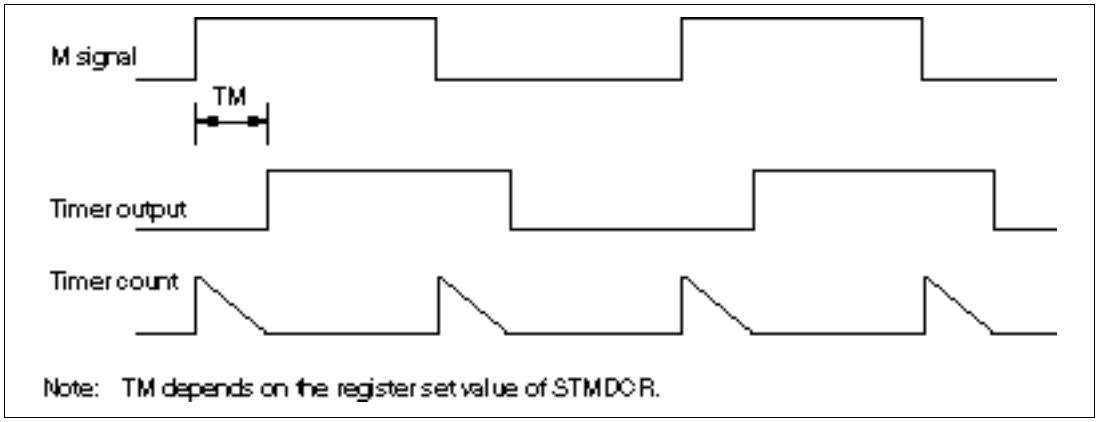


Figure 4.32 Relationship between M signal and Timer Output

(b) Operation in reload timer mode

Figure 4.33 shows the timer output in reload timer mode.

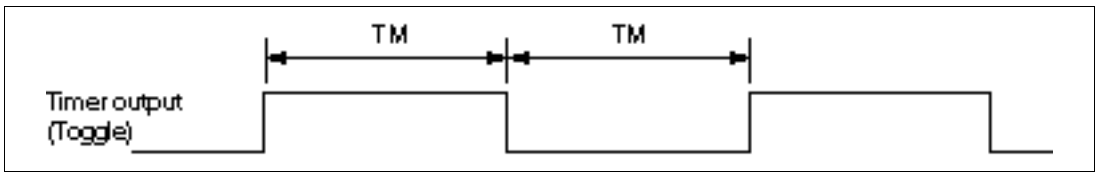


Figure 4.33 Timer Output in Reload Timer Mode

4.9 Keyboard Section

Key scanning of the keyboard in the reference platform is supported by the SH7707 and the FPGA. Based on the 8 × 10 key matrix, the specified key from 60 keys is detected.

4.9.1 Features

(1) SH7707

— Interrupt signal from FPGA can be detected by using only IRQ1.

(2) EPM7160 (FPGA)

— 8-bit reload timer

- Includes an 8-bit down counter and a reload register
- Can start and stop the timer arbitrarily

— Key scan control

- 60 keys with 8 output ports and 10 input ports can be scanned.
- Supports key input interrupt

(3) Keyboard

— 8 × 10 key-scanning keyboard is used.

4.9.2 Key Scan Block Diagram

Figure 4.34 shows the key scan block diagram.

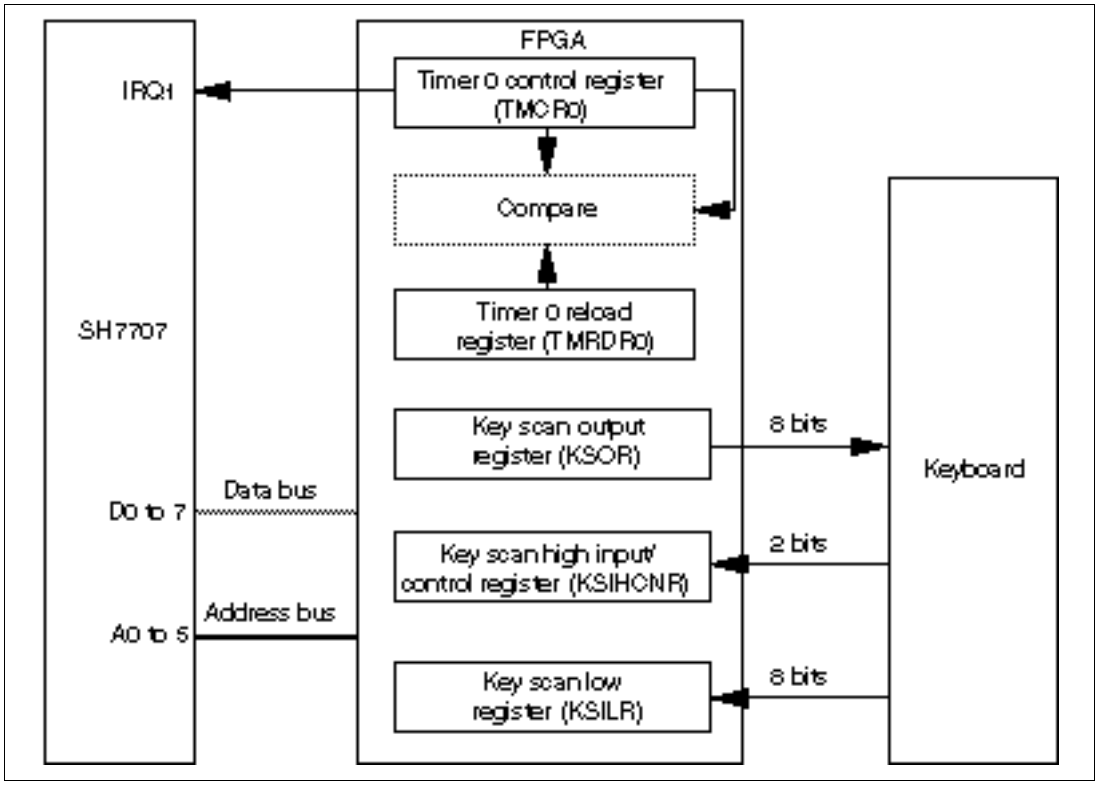


Figure 4.34 Key Scan Block Diagram

4.9.3 Pin Configuration

Table 4.37 shows the SH7707 pin configuration.

Table 4.37 SH7707 Pin Configuration

Pin No.	Pin Name	Input/Output	Function
9	IRQ1	Input	Interrupt request
88	RD_L	Output	Strobe read
89	WE0_L	Output	D7 to D0 selection signal
100	CS4_L	Output	Chip selection
162	CKIO	Input/output	System clock
193	RES_L	Input	Reset request

4.9.4 Used Internal Register Description

Table 4.38 lists the FPGA and SH7707 internal register configuration and functions that are used for key scanning.

Table 4.38 Used Internal Registers

Register Name	Function
KSIHCNR (FPGA)	Key scan input high/control register. Indicates the input level of the key scan input port and controls interrupt output at key input detection.
KSOR (FPGA)	Key scan output register. Specifies the output level of the key scan output port.
TMCR01 (FPGA)	Timer control register. Controls counter, timer output, and interrupts.
TMRDR0 (FPGA)	Timer reload data register. If underflow occurs after counting backwards, a value is set at underflow flag UNDF. At reset, all bits become 0.
SSR (SH7707)	Save status register. When an exception occurs, it saves the SR register contents, and when the exception processing ends, it indicates the return address. After reset, the value is undefined.
SR (SH7707)	Status register. When the BL bit is 1, it suppresses the exception and interrupt occurrence. When the BL bit is 0, it accepts the exception and interrupts.
VBR (SH770)	Stores the base address of the exception processing vector address. At reset, the value is initialized to 00000000.
ICR1 (SH7707)	Interrupt control register. Sets input-signal detection mode of the external interrupt input pin NMI, and indicates the input signal level to the NMI pin.
IPRC (SH7707)	Interrupt priority level setting register. Sets the interrupt level of the 8-bit timer.

4.9.5 FPGA Key Scan Control

Table 4.39 lists set values corresponding to the key scan input/output register at key input.

Table 4.39 Set Value Corresponding to Key Scan Input/Output Register

Key No./Register Name	KSOR (output)	KSIHCNR (input)	KSILR (input)
Alt	FE (KR0)	3 (KC0)	FE (KC0)
Space	FE (KR0)	3 (KC4)	EF (KC4)
~	FE (KR0)	2 (KC8)	FF (KC8)
Shift*[1]	FD (KR1)	3 (KC0)	FE (KC0)
A	FD (KR1)	3 (KC1)	FD (KC1)
ESC	FD (KR1)	3 (KC2)	FB (KC2)
Tab	FD (KR1)	3 (KC3)	F7 (KC3)
Delete	FD (KR1)	3 (KC7)	7F (KC7)
I	FD (KR1)	2 (KC8)	FF (KC8)
Control	FB (KR2)	3 (KC0)	FE (KC0)
S	FB (KR2)	3 (KC1)	FD (KC1)
1	FB (KR2)	3 (KC2)	FB (KC2)
Q	FB (KR2)	3 (KC3)	F7 (KC3)
Z	FB (KR2)	3 (KC4)	EF (KC4)
}	FB (KR2)	3 (KC7)	7F (KC7)
+	FB (KR2)	2 (KC8)	FF (KC8)
Shift*[2]	F7 (KR3)	3 (KC0)	FE (KC0)
D	F7 (KR3)	3 (KC1)	FD (KC1)
2	F7 (KR3)	3 (KC2)	FB (KC2)
W	F7 (KR3)	3 (KC3)	F7 (KC3)
X	F7 (KR3)	3 (KC4)	EF (KC4)
Return	F7 (KR3)	3 (KC6)	BF (KC6)
{	F7 (KR3)	3 (KC7)	7F (KC7)
–	F7 (KR3)	2 (KC8)	FF (KC8)
F	EF (KR4)	3 (KC1)	FD (KC1)
3	EF (KR4)	3 (KC2)	FB (KC2)
E	EF (KR4)	3 (KC3)	F7 (KC3)
C	EF (KR4)	3 (KC4)	EF (KC4)
?	EF (KR4)	3 (KC5)	DF (KC5)
”	EF (KR4)	3 (KC6)	BF (KC6)
P	EF (KR4)	3 (KC7)	7F (KC7)
0	EF (KR4)	2 (KC8)	FF (KC8)

Table 4.39 Set Value Corresponding to Key Scan Input/Output Register (cont)

Key No./Register Name	KSOR (output)	KSIHCNR (input)	KSILR (input)
Fn	EF (KR4)	1 (KC9)	FF (KC9)
Alt	DF (KR5)	3 (KC0)	FE (KC0)
G	DF (KR5)	3 (KC1)	FD (KC1)
4	DF (KR5)	3 (KC2)	FB (KC2)
R	DF (KR5)	3 (KC3)	F7vKC3)
V	DF (KR5)	3 (KC4)	EF (KC4)
>	DF (KR5)	3 (KC5)	DF (KC5)
:	DF (KR5)	3vKC6)	BF (KC6)
O	DF (KR5)	3 (KC7)	7F (KC7)
9	DF (KR5)	2 (KC8)	FF (KC8)
H	BF (KR6)	3 (KC1)	FD (KC1)
5	BF (KR6)	3 (KC2)	FB (KC2)
T	BF (KR6)	3 (KC3)	F7 (KC3)
B	BF (KR6)	3 (KC4)	EF (KC4)
<	BF (KR6)	3 (KC5)	DF (KC5)
L	BF (KR6)	3 (KC6)	BF (KC6)
I	BF (KR6)	3 (KC7)	7F (KC7)
8	BF (KR6)	2 (KC8)	FF (KC8)
STOP	BF (KR6)	1 (KC9)	FF (KC9)
J	7F (KR7)	3 (KC1)	FD (KC1)
6	7F (KR7)	3 (KC2)	FB (KC2)
Y	7F (KR7)	3 (KC3)	F7 (KC3)
N	7F (KR7)	3 (KC4)	EF (KC4)
M	7F (KR7)	3 (KC5)	DF (KC5)
K	7F (KR7)	3 (KC6)	BF (KC6)
U	7F (KR7)	3 (KC7)	7FvKC7)
7	7F (KR7)	2 (KC8)	FF (KC8)
◆	7F (KR7)	1 (KC9)	FF (KC9)

Notes
1. ( ) indicates the key input/output row or column (see figure 4.35, Key Matrix.)
2. Each time a key is pressed, the input/output row or column register value changes from 1 (high) to 0 (low).
3. The output values are set to KSOR (8 bits), and the upper 2 bits of the 10-bit input values are set to KSIHCNR, and the lower 8 bits are set to KSILR.

4.9.6 Key Matrix

Figure 4.35 shows the 8 × 10 key matrix.

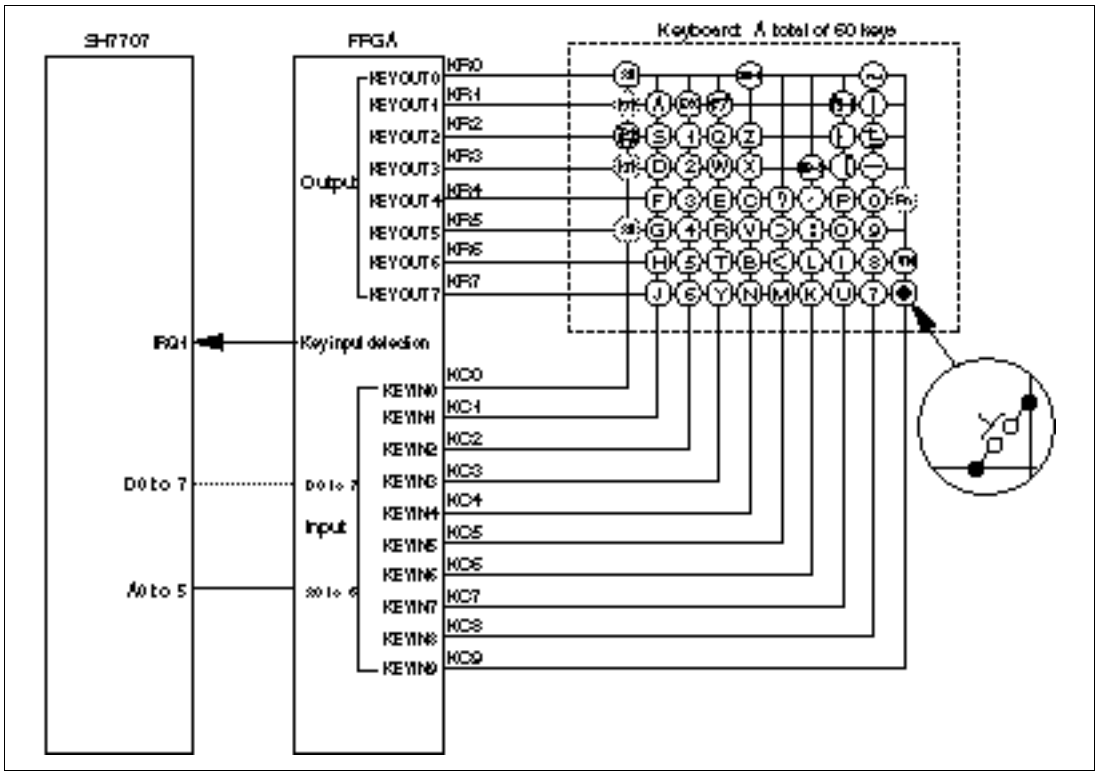


Figure 4.35 Key Scanning of 8 × 10 Key Matrix

4.9.7 Key Scan Procedures

Table 4.40 describes the key scan procedures.

Table 4.40 Key Scan Procedures

No.	Hardware Processing	Software Processing	Event
1	Pull up key scan input		Initialization
2	Output NAND of key scan input to IRQ3		
3		Key scan; all output are 0; set all of FPGA [KEYOUT1 to KEYOUT7] to 0	
4		Enable key scan interrupt IRQ1	
5	IRQ1 occurs		A key input
6		Initiate key scan timer	IRQ3 interrupt processing
7		Key scan; all output are high; set all of FPGA [KEYOUT1 to KEYOUT7] to 1	
8		Disable key scan interrupt IRQ1	
9	IRQ1 occurs		The specified time elapsed
10		Set key scan output 1 port to low, and other ports to high, and check input port.	IRQ0 interrupt processing
11		Detect the input key, and perform chattering. Then, repeat nos. 4 and 5.	
12		If the input key cannot be detected, determine input end, stop key scan timer, set all key scan output to low, enable key scan interrupt, and return to no. 2.	

4.9.8 Key Scan Operation

Figure 4.36 shows the timing chart in the key scan operation.

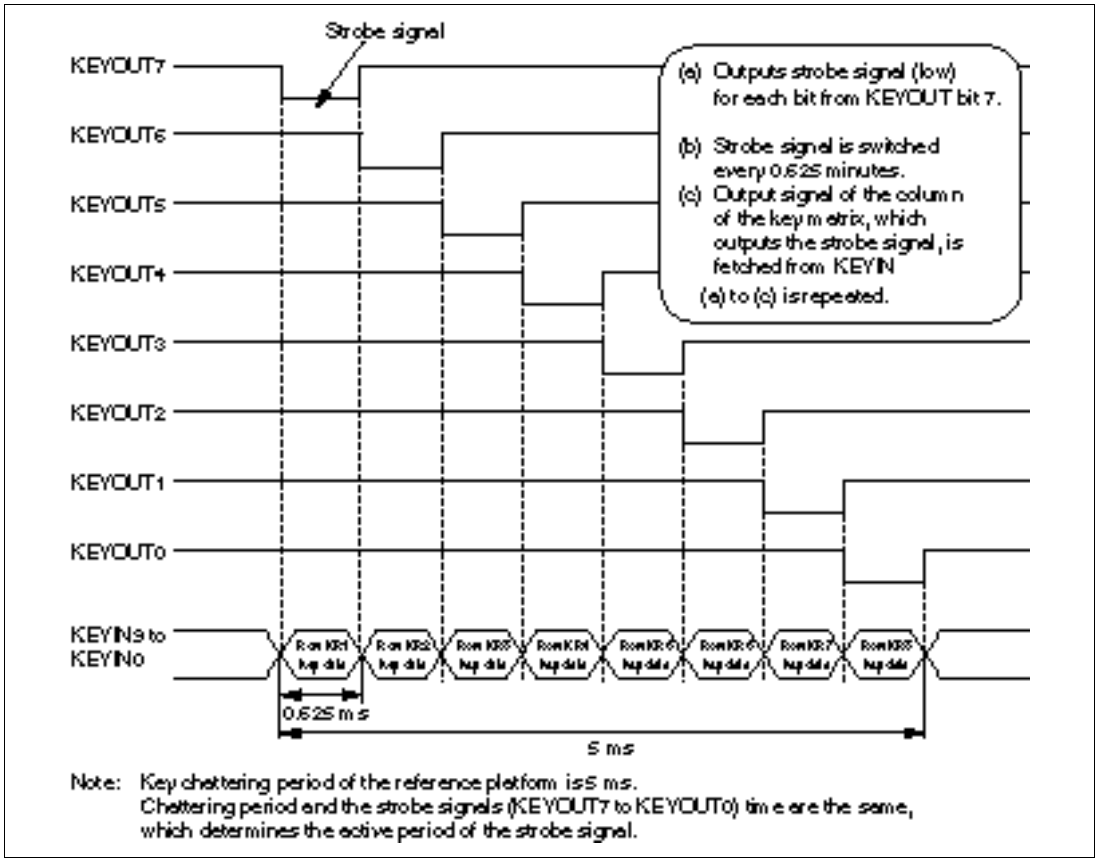


Figure 4.36 Key Scan Operation Timing Chart

Appendix A Test Specifications

A.1 Monitor Program

The monitor program enables the user to read or write to the system memory space and CPU registers; the state of peripheral modules such as the AUDIO, LCD, or DRAM, in addition to the CPU registers can be checked. The monitor program can also modify the register contents; this function is useful for hardware and software debugging.

(1) Starting Monitor Program

The reference platform operates in the little endian mode. Before starting the monitor program, follow the procedure below:

1. Insert jumper pins into SW3 and SW4, and leave SW2, SW5, SW6, and SW7 open.

2. Replace the flash memory devices storing Microsoft® Windows® CE operating system with the flash memory devices storing the monitor program, which are supplied together with the reference platform board.

3. Each flash memory device has one of the numbers 1 to 4 on its bottom (M1 to M4 for the flash memory devices storing the monitor program, and W1 to W4 for those storing Windows® CE).
   Install the devices as follows:
   U7: Flash memory 1
   U11: Flash memory 2
   U8: Flash memory 3
   U12: Flash memory 4
   Be sure to align pin 1 of the flash memory device to pin 1 of the corresponding socket.

After the above procedure, start up the program by the following procedure:

4. Start up the communication software on the host computer. (The host computer and the reference platform must be connected through a 9-pin straight serial cable.)

5. Set the communication parameters as follows:
   Baud rate: 9600 bps
   Data length: 8 bits
   Stop bit: 1 bit
   Parity: None
   Flow control: None

6. After setting the above, turn on the reference platform. The following message will be displayed, and the command input wait state is entered.

```
****************************************
     C1 monitor (9.6 Kbps)  Ver. 1.0
                      / APL 1997.3.16
****************************************
>
```

(After the above message appears, commands can be entered.)

(2) Monitor Program Commands

In the following description of each monitor program command format, the parameter in { } can be omitted, but that in [ ] must be specified.

(a) H Command (Help)

Format: H

The H command displays the list of command formats and functions as follows:

```
>h
     MINI monitor command
 D   {sadr {eadr}}          memory dump (DM sets dump size)
 F   [sadr] [eadr] [data]   fill memory
 FL  [sadr] [eadr] [data]   fill memory (LONG)
 G   {start_adr}            go program
 H                          help
 M   [adr]                  set memory (BYTE)
 MW  [adr]                  set memory (WORD)
 ML  [adr]                  set memory (LONG)
 X   {reg {data}}           change register
 R                          display registers
 L                          load program
 DM  {B|W|L}                set&disp dump mode
 MV  [sadr] [dadr] [len]    move memory
```

(b) D Command (Dump)

Format: D {start address {end address}}

The D command displays 256-byte memory contents within the specified range.

To display the contents in word or longword units, enter the DM command before D command as follows:

DM L (4-byte units)

DM W (2-byte units)

DM B (1-byte units)

Examples:

```
>d ffffff00
FFFFFF00  16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16  ................
FFFFFF10  16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16  ................
FFFFFF20  16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16  ................
FFFFFF30  16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16  ................
FFFFFF40  11 00 F0 00 F3 00 EF 00 34 00 00 00 00 00 15 00  ........4.......
FFFFFF50  02 00 D0 00 95 00 00 00 00 00 00 00 00 00 00 00  ....-...........
FFFFFF60  11 00 F0 00 F3 00 EF 00 34 00 00 00 00 00 15 00  ........4.......
FFFFFF70  89 00 D0 00 92 00 00 00 00 00 00 00 00 00 00 00  >...*...........
FFFFFF80  16 16 00 00 00 00 00 00 00 00 00 00 16 16 16 16  ................
FFFFFF90  16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16  ................
FFFFFFA0  16 16 16 16 02 02 16 16 16 16 16 16 16 16 16 16  ................
FFFFFFB0  16 16 16 16 00 00 16 16 16 16 16 16 16 16 16 16  ................
FFFFFFC0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
FFFFFFD0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
FFFFFFE0  00 00 00 00 08 08 08 08 40 40 40 40 00 00 00 00  ........@@@@....
FFFFFFF0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................

>dmL
DMmode = long>
>d
00000000  6002D003 89078800 402BD002 00000009  ...`......+@....
00000010  FFFFFFD4 A0001000 402BD001 00000009  .........+@....
00000020  A0001000 FFFFFFFF FFFFFFFF FFFFFFFF  ................
00000030  FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF  ................
00000040  FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF  ................
00000050  FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF  ................
00000060  FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF  ................
```

```
00000070  FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF  ................
00000080  FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF  ................
00000090  FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF  ................
000000A0  FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF  ................
000000B0  FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF  ................
000000C0  FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF  ................
000000D0  FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF  ................
000000E0  FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF  ................
000000F0  FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF  ................
>
```

(c) F and FL Commands (Fill)

Format: F [start address] [end address] [data] (data must be one byte long)
FL [start address] [end address] [data] (data must be four bytes long)

The F and FL commands modify the memory contents of the specified range.

The F command writes data in byte (8-bit) units, and the FL command writes in longword (32-bit) units.

Examples:

```
>f 0c000000 0c000010 34
```

34 (hexadecimal) is written to addresses 0c000000 to 0c000010 in byte units.

```
>fl 0c000000 0c000020 12345678
```

12345678 (hexadecimal) is written to addresses 0c000000 to 0c000020 in longword units.

(d) G Command

Format: G {start address}

The G command starts program execution from the specified address.

Example:

```
>G 0c000000
```

The program is started from address 0C000000.

(e) M, MW, and ML Commands

Format: M [address] (modifies in byte units)
MW [address] (modifies in word (16-bit) units)
ML [address] (modifies in longword (32-bit) units)

The M, MW, and ML commands display and modify the specified address contents. After the address contents are modified, the next address contents will be displayed and can be modified.

The M command writes data in byte units, the MW command writes in word units, and the ML command writes in longword units.

While an address and its contents are displayed, entering the ^ key displays the previous address, the return key displays the next address, and the . key makes the display return to the command input wait state.

Examples:

```
> m 10000000
```

The memory contents are displayed and modified in byte units, starting from address 10000000.

```
> mw 10000000
```

The memory contents are displayed and modified in word units, starting from address 10000000.

(f) X Command

Format: X {register {data}}

The X command modifies the SH7707 register contents.

Example:

```
>X R0 12345678
```

The contents of the R0 register in the SH7707 are modified to 12345678 (hexadecimal).

(g) R Command

Format: R

The R command displays the current SH7707 register contents.

Example:

```
>R
PC  AC000000 SR  600000F0 GBR  00000000 VBR  00000000
PR  00000000 MACH 00000000 MACL 00000000 SSR  600000F0 SPC  AC000000
R0-7  12345678 00000000 00000000 00000000 00000000 00000000 00000000 00000000
R8-15 00000000 00000000 00000000 00000000 00000000 00000000 00000000 ACFAFFF0
>
```

(h) L Command

Format: L

The L command loads the Motorola S-type program file to the memory on the reference platform.

Example:

```
>L
please send !('.' & CR stop load)
```

When the above message is displayed, send the Motorola S-type program file from the host computer. After loading is completed, the display returns to the command input wait state.

(i) MV Command

Format: MV [source address] [destination address] [number of bytes]

The MV command copies the contents of the specified number of bytes from the source address to the destination address.

Example:

```
>mv 0c000000 0c0000f0 6
```

The contents in the 6-byte area starting from address 0C000000 are copied to the 6-byte area starting from address 0C0000F0.

(3) Test Program Execution Example

The following shows an example of test program execution after the monitor program has been started up (the following message has been displayed). In this example, the LCD test program is executed.

To execute a test program, the communication software must be used on the host computer, and the host computer and the reference platform must be connected through a 9-pin straight serial cable.

```
****************************************
     C1 monitor(9.6 Kbps)  Ver. 1.0
                     / APL 1997.3.16
****************************************
>
```

The following shows the procedure for test program execution:

1. After the above message, enter L [return], and the following message will be displayed:

   ```
   Please send!('.'&CR stop load)
   ```

2. The transfer file must have extension ".MOT". In this example, send the LCD_L.MOT file using the communication software on the host computer.

3. After the transfer is completed and the command input wait state is entered, enter the following command:

```
>G 0c000000 [return]
```

The test program will run (0c000000 is the start address of the test program).

The following table lists the test programs.

Test Name	File Name	Description
DRAM	DRAM_L.MOT	Reads from and writes to DRAM in byte, word, and longword units.
LCD	LCD_L.MOT	Displays 4-grayscale-color stripes, then moves vertical lines from left to right, and horizontal lines from top to bottom.
SOUND	AUDIO_L.MOT	Outputs a 200-Hz sine wave.
SCI	SEND.MOT	Outputs characters in sequence from the reference platform to the host computer.
PCMCIA	PCMCIA_L.MOT	Reads and writes through the PCMCIA in byte and word units.
IrDA (transmit)	L_SEND.MOT	Outputs a character from the reference platform to the host computer.
IrDA (receive)	L_RECV.MOT	Outputs a character from the host computer to the reference platform.

A.2 DRAM

The following shows the DRAM test method and the DRAM test program.

A.2.1 DRAM Test Method

(1) Test Items

The DRAM is tested by using the test program and the monitor program. The test program performs the following three test items:

(a) Writes and reads data to and from each address in byte units, and compares the written data with the read data.

(b) Writes and reads data to and from each even address in word units, and compares the written data with the read data.

(c) Writes and reads data to and from each 4-byte boundary address, and compares the written data with the read data.

To each address, a data item and its reversed data item are written, that is, each address is written to twice. When the test is failed at an address, the test ends there.

(2) Test Results

When the test ends, the test program writes the test results to the specified DRAM addresses. When all test items have been passed, the data indicating that the test have been passed is written to the specified address. When the test has been failed, the address and item in which the test has been failed are written to the DRAM. Table A.1 shows the data to be written as test results and the DRAM addresses to store the test results.

Table A.1 Test Result Data and DRAM Addresses

Test Result	DRAM Address	Data to Be Written
All test items passed	H'0CFFF004	H'12345678
Byte test failed	H'0CFFF004	H'00424154
	H'0CFFF008	Failed address
Word test failed	H'0CFFF004	H'00574154
	H'0CFFF008	Failed address
Longword test failed	H'0CFFF004	H'4C574154
	H'0CFFF008	Failed address

(3) Monitor Program Operation

Operate the monitor by the following procedure when testing the DRAM:

(a) Start up the communication software on the host computer.

(b) Enter the L command to load the DRAM test program.

(c) Enter the G command with the start address (H'0C000000) specified, to start the test program.

(d) Leave the test program executing for two or three minutes, then reset the reference platform to stop the test program.

(e) Read the test result from the DRAM address (H'0CFFF004) to check if all test items have been passed.

A.2.2 DRAM Test Program

The following test program reads and writes to the DRAM. The program must be loaded and executed from the monitor program described in appendix A.1.

```
;=============================================================
;
;                    DRAM test program for Windows-CE
;        <<copyright (c) Hitachi,LTD.,1996-1997 All rights reserved.>>
;
;
      .org    h'A0000000

     mov.l   $$DRAM_TEST_PREPRO,r0       ;
     jmp     @r0                         ;
     nop                                 ;
      .align 4
$$DRAM_TEST_PREPRO:.data.l DRAM_TEST_PREPRO

;=============================================================
;
;
;                           Initialize BSC
;
      BCR1:          .equ    h'00
      BCR2:          .equ    h'02
      WCR1:          .equ    h'04
      WCR2:          .equ    h'06
      MCR:           .equ    h'08
      DCR:           .equ    h'0A
      PCR:           .equ    h'0C
      RTCSR:         .equ    h'0E
      RTCNT:         .equ    h'10
      RTCOR:         .equ    h'12
      RFCR:          .equ    h'14
      BCR3:          .equ    h'1E
```

Figure A.1 DRAM Test Program

```
;=====================================================================
      .org    h'0800
DRAM_TEST_PREPRO:
      mov.l MAINSP,sp                    ;
;
;=====================================================================
INIT_BSC:
      mov.l  BSC_GBR,r0                  ; Stores the BSC base address
      ldc    r0,gbr                      ; into the GBR.
      mov.l  BCR1_DAT,r0                 ;
      mov.w  r0,@(BCR1,gbr)              ;
      mov.l  BCR2_DAT,r0                 ;
      mov.w  r0,@(BCR2,gbr)              ;
      mov.l  BCR3_DAT,r0                 ;
      mov.w  r0,@(BCR3,gbr)              ;
      mov.l  WCR1_DAT,r0                 ;
      mov.w  r0,@(WCR1,gbr)              ;
      mov.l  WCR2_DAT,r0                 ;
      mov.w  r0,@(WCR2,gbr)              ;
      mov.l  MCR_DAT,r0                  ;
      mov.w  r0,@(MCR,gbr)               ;
      mov.l  MCR_DAT,r0                  ;
      mov.w  r0,@(MCR,gbr)               ;
      mov.l  DCR_DAT,r0                  ;
      mov.w  r0,@(DCR,gbr)               ;
      mov.l  PCR_DAT,r0                  ;
      mov.w  r0,@(DCR,gbr)               ;
      mov.l  RTCSR_DAT,r0                ;
      mov.w  r0,@(RTCSR,gbr)             ;
      mov.l  RTCNT_DAT,r0                ;
      mov.w  r0,@(RTCNT,gbr)             ;
      mov.l  RTCOR_DAT,r0                ;
      mov.w  r0,@(RTCOR,gbr)             ;
      mov.l  RFCR_DAT,r0                 ;
      mov.w  r0,@(RFCR,gbr)              ;
      bra    SET_DRAM_REF                ;
      nop                                ;
      .align 4
```

Figure A.1 DRAM Test Program (cont)

```
BSC_GER:        .data.l  h'FFFFFF60
BCR1_DAT:       .data.l  h'00001011
BCR2_DAT:       .data.l  h'00003FFC
BCR3_DAT:       .data.l  h'00000000
WCR1_DAT:       .data.l  h'00003FF3
WCR2_DAT:       .data.l  h'0000FFEF
MCR_DAT:        .data.l  h'0000002C
DCR_DAT:        .data.l  h'00000000
PCR_DAT:        .data.l  h'00000000
RTCSR_DAT:      .data.l  h'0000A500
RTCNT_DAT:      .data.l  h'0000A500
RTCOR_DAT:      .data.l  h'0000A500
RFCR_DAT:       .data.l  h'0000A400

;=====================================================================

SET_DRAM_REF

DRAM_TEST
        mov.l   DRAM_TOP_ADR,r1         ;
        mov.l   DRAM_BOTTOM_ADR,r1      ;
Byte_Access_Test                        ; Tests in longword units.
        sts.l   pr,@-s                  ;
        bsr     CHK_BOTTOM              ;
        nop                             ;
        lds.l   @sp+,pr                 ;
        bt      BAT_SUCCESS             ;

        mov.b   r1,@(r0,r10)            ;
        mov.b   @(r0,r10),r3            ;
        extu.b  r1,r1                   ;
        extu.b  r3,r3                   ;
        cmp/eq  r1,r3                   ;
        bf      BAT_FAIL                ;
        not     r1,r2                   ;
        mov.b   r2,@(r0,r10)            ;
        mov.b   @(r0,r10),r3            ;
```

Figure A.1 DRAM Test Program (cont)

```
        extu.b  r2,r2                   ;
        extu.b  r3,r3                   ;
        cmp/eq  r2,r3                   ;
        bf      BAT_FAIL                ;
        add     #1,r0                   ;
        bra     Byte_Access_Test        ;
        add     #1,r1                   ;
BAT_SUCCESS:
        xor     r0,r0                   ;
        xor     r1,r1                   ;
Word_Access_Test:
        sts.l   pr,@-sp                 ;
        bsr     CHK_BOTTOM              ;
        nop                             ;
        lds.l   @sp+,pr                 ;
        bt      WAT_SUCCESS             ;
        mov.w   r1,@(r0,r10)            ;
        mov.w   @(r0,r10),r3            ;
        extu.w  r1,r1                   ;
        extu.w  r3,r3                   ;
        cmp/eq  r1,r3                   ;
        bf      WAT_FAIL                ;
        not     r1,r2                   ;
        mov.w   r2,@(r0,r10)            ;
        mov.w   @(r0,r10),r3            ;
        extu.w  r2,r2                   ;
        extu.w  r3,r3                   ;
        cmp/eq  r2,r3                   ;
        bf      WAT_FAIL                ;
        add     #2,r0                   ;
        bra     Word_Access_Test        ;
        add     #1,r1                   ;
WAT_SUCCESS:
        xor     r0,r0                   ;
        xor     r1,r1                   ;
Long_Word_Access_Test
        sts.l   pr,@-sp                 ;
```

Figure A.1 DRAM Test Program (cont)

```
        bsr     CHK_BOTTOM              ;
        nop                             ;
        lds.l   @sp+,pr                 ;
        bt      LWRT_SUCCESS            ;

        mov.l   r1,@(R0,r10)            ;
        mov.l   @(r0,r10),r2            ;
        cmp/eq  r1,r2                   ;
        bf      LWRT_FAIL               ;
        not     r1,r2                   ;
        mov.l   r2,@(R0,r10)            ;
        mov.l   @(r0,r10),r2            ;
        cmp/eq  r2,r2                   ;
        bf      LWRT_FAIL               ;
        add     #4,r0                   ;
        bra     Long_Word_Access_Test   ;
        add     #1,r1                   ;
LWRT_SUCCESS:
        mov.l   RESULT_ADR,r1           ;
        mov.l   SUCCESS_DAT,r0          ;
        bra     LWRT_SUCCESS            ;
        mov.l   r0,@r1                  ;
BAT_FAIL:
        mov.l   RESULT_ADR,r1           ;
        mov.l   BAT_FAIL_DAT,r2         ;
        mov.l   r2,@r1                  ;
        add     r10,r0                  ;
        bra     BAT_FAIL                ;
        mov.l   r0,@(4,r1)              ;
WRT_FAIL:
        mov.l   RESULT_ADR,r1           ;
        mov.l   WRT_FAIL_DAT,r2         ;
        mov.l   r2,@r1                  ;
                r10,r0                  ;
                WRT_FAIL                ;
        mov.l   r0,@(4,r1)              ;
```

Figure A.1 DRAM Test Program (cont)

```
LWAT_FAIL:
      mov.l   RESULT_ADR,r1           ;
      mov.l   LWAT_FAIL_DAT,r2        ;
      mov.l   r2,@r1                  ;
      add     r10,r0                  ;
      bra     LWAT_FAIL               ;
      mov.l   r0,@(4,r1)              ;
CHK_BOTTOM:
      mov     r10,r5                  ; Local common routine
      add     r0,r5                   ;
      cmp/eq  r5,r11                  ; Returns with the T bit value
      rts                             ; retained.
      nop                             ;
      .align 4
DRAM_TOP_ADR:          .data.l h'0C000000
DRAM_BOTTOM_ADR:       .data.l h'0CFFF000
RESULT_ADR:            .data.l h'0CFFF004
SUCCESS_DAT:           .data.l h'12345678
BAT_FAIL_DAT:          .data.l h'00424154
WAT_FAIL_DAT:          .data.l h'00574154
LWAT_FAIL_DAT:         .data.l h'4C574154
MAINSP:                .data.l h'0CFFFFFC

       .end
```

Figure A.1 DRAM Test Program (cont)

A.3 LCD

A.3.1 LCD Test Program

The following test program displays 4-grayscale-color stripes, then moves vertical lines from left to right and horizontal lines from top to bottom. The program must be loaded and executed from the monitor program described in appendix A.1.

```
;=======================================================================
;
;                       LCD test program for Windows-CE
;                              For Little-Endian
;            ccopyright (c) Hitachi,Ltd.,1996-1997 All rights reserved.cc
;
;=======================================================================
            .org    h'0C000000

            bra     SET_LCD_TEST_PORT       ; Initialize the SH7707 to prepare
            nop                             ; for the LCD test
LCD_TEST:

            bsr     Four_color              ;
            nop                             ;
            mov.l   Wait_Timer_Value,r0     ;
            bsr     Wait_Timer              ;
            nop                             ;

;=======================================================================
;
;                       Horizontal Scroll Test
;
;
;=======================================================================
Horizontal_Test:
            mov.l   HPADR,r0                ; The HPADR initial value is LCD_START.
            mov.l   LCD_START,r1            ;
            mov.l   r1,@r0                  ; Stores H'0C000000 in address H'0C000840.

            xor     r0,r0                   ; The HPCNT initial value is 0.
            mov.l   HPCNT,r1                ;
            mov.l   r0,@r1                  ; Stores H'00000000 in address H'0C000840.

            bsr     Save_color              ; Paints the screen white or black
            xor     r0,r0                   ; before the horizontal scroll test.
```

Figure A.2 LCD Test Program

```
Horizontal_Test_loop:
            mov.l   #HPCNT,r1               ; Determines whether to stop the horizontal test.
            mov.l   #0x1,r0                 ;
            mov.l   #HSIZE,r2               ; Stops the test when the horizontal pointer
            cmp/hw  r2,r0                   ; value becomes 120.
            bt      Horizontal_Test_End     ;  (480 dots * 2 bits) = 120 bytes
            mov     r0,r14                  ; Copies HPCNT.
            mov     #5,r9                   ;

Horizontal_test_exec:
            mov     r14,r15                 ;
            Add     r9,r15                  ;
            mov.l   #HPCNT,r1               ;
            mov.l   r15,@r1                 ;
            wtw.l   pr,@-wp                 ;
            bwr     Horizontal_exec         ;
            nop                             ;
            ldw.l   @wp+,pr                 ;
            twt     r9,r9                   ;
            bt      Horizontal_exec_loop    ;
            Add     #(-1),r9                ;
            bra     Horizontal_test_exec    ;
            nop                             ;

Horizontal_exec_loop:
            Add     #4,r14                  ;
            mov.l   #HPCNT,r1               ; Increments the horizontal pointer value.
            bra     Horizontal_Test_loop    ;
            mov.l   r14,@r1                 ;

Horizontal_exec:
            wtw.l   pr,@-wp                 ;
            bwr     Paint_Pattern           ;
            mov     #h'00,r0                ;
            bwr     Wait_Timer              ;
            xor     r0,r0                   ;
            bwr     Paint_Pattern           ;
            mov     #h'50,r0                ;
            bwr     Wait_Timer              ;
            xor     r0,r0                   ;
            bwr     Paint_Pattern           ;
            mov     #h'0C,r0                ;
            bwr     Wait_Timer              ;
            xor     r0,r0                   ;
            bwr     Paint_Pattern           ;
            mov     #h'05,r0                ;
            bwr     Wait_Timer              ;
            xor     r0,r0                   ;
            bwr     Paint_Pattern           ;
            xor     r0,r0                   ;
            ldw.l   @wp+,pr                 ;
            rtw                             ;
            nop                             ;
Horizontal_Test_End:
```

Figure A.2 LCD Test Program (cont)

```
;=============================================================================
;
;                          Vertical Scroll Test
;
;=============================================================================
Vertical_Test:
          mov.l    VRAIR,r0                ; The VP1ADR initial value is LCD_START.
          mov.l    LCD_START,r1            ;
          mov.l    r1,@r0                  ;
          mov.l    VR2AIR,r0               ; The VP2ADR initial value is LCD_START.
          mov.l    LCD_START,r1            ;
          mov.l    r1,@r0                  ;

          xor      r0,r0                   ; The VPCNT initial value is zero.
          mov.l    VRCNT,r1                ;
          mov.l    r0,@r1                  ;

          bsr      Same_color              ; Paints the screen white or black
          xor      r0,r0                   ; before the vertical scroll test.

Vertical_Test_loop:
          mov.l    VRCNT,r1                ;
          mov.l    @r1,r0                  ;
          mov.l    VRSIZE,r2               ;
          cmp/eq   r0,r2                   ;
          bt       Vertical_Test_End       ;

          mov.l    pr,@-sp                 ;
          mov.l    VRAIR,r5                ;
          bsr      Paint_VpAttern          ;
          mov      #h'FF,r0                ;
          bsr      Wait_Timer              ;
          xor      r0,r0                   ;
          mov.l    VR2AIR,r5               ;
          bsr      Paint_VpAttern          ;
          xor      r0,r0                   ;

          mov.l    VRCNT,r1                ;
          mov.l    @r1,r0                  ;
          add      #1,r0                   ;
          bra      Vertical_Test_loop      ;
          mov.l    r0,@r1                  ;

Vertical_Test_End:

          nop                              ;
          nop                              ;
          bra      LCD_TEST                ;
          nop                              ;

;=============================================================================
;
;          Function for Painting Vertical Lines in Same Color
;
;
;=============================================================================
```

Figure A.2 LCD Test Program (cont)

```
Paint_MpAttern:                                 ; The parameter is passed through R0.
            mov.l   MRADR,r1                    ;
            mov.l   @r1,r2                      ;

            mov.l   MRCNT,r5                    ; The offset value for the horizontal address
            mov.l   @r5,r5                      ;
            Add     r5,r2                       ;
Paint_MpAttern_loop:
            mov.l   LCD_END,r5                  ;
            cmp/hw  r5,r2                       ;
            bt      END_OF_PAINTING             ;

            mov.b   r0,@r2                      ;
            Add     #h'78,r2                    ;
            bra     Paint_MpAttern_loop         ;
            mov.l   r2,@r1                      ;
END_OF_PAINTING:
            mov.l   LCD_START,r2                ;
            rtw                                 ;
            mov.l   r2,@r1                      ;
;==========================================================================
;
;           Function for Painting Horizontal Lines in Same Color
;
;
;==========================================================================
Paint_VpAttern:                                 ; The parameters are passed through R0 and R5.
        xor         r5,r5                       ;
Paint_VpAttern_loop:
            mov.l   @r5,r2                      ;
            mov.l   r0,@r2                      ;
            Add     #4,r2                       ;
            mov.l   r2,@r5                      ;

            mov.l   Vertical_END,r4             ;
            cmp/eq  r5,r4                       ;
            Add     #1,r5                       ;
            bf      Paint_VpAttern_loop         ;
            rtw                                 ;
            xor     r5,r5                       ;

            .Align 4
Wait_Timer_Value: .data.l           h'00500000
MRADR:            .data.l           h'00009510
VRADR:            .data.l           h'00009514
VR2ADR:           .data.l           h'00009518
MRCNT:            .data.l           h'0000951C
VRCNT:            .data.l           h'00009520
MRSIZE:           .data.l           h'00000078
VRSIZE:           .data.l           h'00000140
LCD_START:        .data.l           h'00010000
LCD_END:          .data.l           h'00012600
Vertical_END:     .data.l           h'0000001D
```

Figure A.2 LCD Test Program (cont)

```
;=============================================================================
;
;                       Function for Wait Timer
;
;
;       The parameter is passed through R0. When the R0 value is zero,
;       the default timer value is used; otherwise, the R0 value is used
;       as the timer value.
;
;=============================================================================
Wait_Timer:
          tst     r0,r0                   ;
          bf      Wait_Timer_loop         ;
          mov.l   Timer_cnt,r0            ;
Wait_Timer_loop:
          tst     r0,r0                   ;
          bt      Wait_Timer_END          ;
          bra     Wait_Timer_loop         ;
          add     #-1,r0                  ;
Wait_Timer_END:
          rts                             ;
          nop                             ;
          .align 4
Timer_cnt:.data.l  h'00011200
;=============================================================================
;
;                   Function for Painting Screen in Same Color
;
;
;=============================================================================
Same_color:
          tst     r0,r0                   ;
          bt      PALETTE_00              ;
          cmp/eq  #h'05,r0                ;
          bt      PALETTE_55              ;
          cmp/eq  #h'0A,r0                ;
          bt      PALETTE_AA              ;
PALETTE_FF:
          bra     LCD_FM_CLR_START        ;
          mov     #h'FF,r0                ;
PALETTE_00:
          bra     LCD_FM_CLR_START        ;
          xor     r0,r0                   ;
PALETTE_55:
          mov.l   PALETTE_0x55,r0         ;
          bra     LCD_FM_CLR_START        ;
          nop                             ;
PALETTE_AA:
          mov.l   PALETTE_0xAA,r0         ;
          bra     LCD_FM_CLR_START        ;
          nop                             ;

LCD_FM_CLR_START:
          mov.l   LCD_FM_TOP_ADR,r1       ; Specifies the frame memory area
          mov.l   LCD_FM_BOTTOM_ADR,r2    ; for the LCD.
```

Figure A.2 Test Program (cont)

```
LCD_FH_CLR_LOOP:
          cmp/eq   r1,r2                    ; Paints the screen area specified
          bt       LCD_FH_CLR_END           ; by the frame memory area in the same color.
          mov.l    r0,@r1                   ;
          bra      LCD_FH_CLR_LOOP          ;
          add      #4,r1                    ;
LCD_FH_CLR_END:
          rts                               ;
          nop                               ;

;=======================================================================================
;
;
;
;
;=======================================================================================
Four_color:
          mov.l    pr,@-sp                  ;
          xor      r0,r0                    ;
          mov.l    &$TOP1,r1                ;
          mov.l    &$END1,r2                ;
          bsr      Four_color_exec          ;
          nop                               ;
          mov.l    PALETTE_0x55,r0          ;
          mov.l    &$TOP2,r1                ;
          mov.l    &$END2,r2                ;
          bsr      Four_color_exec          ;
          nop                               ;
          mov.l    PALETTE_0xAA,r0          ;
          mov.l    &$TOP3,r1                ;
          mov.l    &$END3,r2                ;
          bsr      Four_color_exec          ;
          nop                               ;
          mov      #h'FF,r0                 ;
          mov.l    &$TOP4,r1                ;
          mov.l    &$END4,r2                ;
          bsr      Four_color_exec          ;
          nop                               ;
          lds.l    @sp+,pr                  ;
          rts                               ;
          nop                               ;
Four_color_exec:
          cmp/eq   r1,r2                    ;
          bt       Four_color_end           ;
          mov.l    r0,@r1                   ;
          bra      Four_color_exec          ;
          add      #4,r1                    ;
Four_color_end:
          rts                               ;
          nop                               ;

          .align 4
PALETTE_0x55:          .data.l  h'55555555
PALETTE_0xAA:          .data.l  h'AAAAAAAA
```

Figure A.2 LCD Test Program (cont)

```
;=============================================================================
;
;                          LCD Initialization Routine

        LCDAR:          .equ    h'C0
        LCDDR:          .equ    h'C2
        LCDDR:          .equ    h'C5
        LCDDtR:         .equ    h'C7
;
;=============================================================================
SET_LCD_TEST_PORT:                          ; Initializes the ports to prepare for the LCD test.

        xor     r1,r1                       ;
        mov.l   PCCR,r0                     ; PortC
        mov.w   r1,@r0                      ;
        mov.l   PDCR,r0                     ; PortD
        mov.w   r1,@r0                      ;

        mov.l   LCD_GER,r0                  ; Initializes the LCD registers.
        ldc     r0,gbr                      ;

        xor     r0,r0                       ; Selects bank R0.
        mov.w   r0,@(LCDAR,gbr)             ;
        mov.w   r0,@(LCDDR,gbr)             ;
        mov.w   r0,@(LCDDtR,gbr)            ;

        mov     #1,r0                       ; Selects bank R1.
        mov.w   r0,@(LCDAR,gbr)             ;
        mov.l   INIT_LCDDR1,r0              ;
        mov.w   r0,@(LCDDR,gbr)             ;
        mov.l   INIT_LCDDR1,r0              ;
        mov.w   r0,@(LCDDR,gbr)             ;
        mov.l   INIT_LCDDtR1,r0             ;
        mov.w   r0,@(LCDDtR,gbr)            ;

        mov     #2,r0                       ; Selects bank R2.
        mov.w   r0,@(LCDAR,gbr)             ;
        mov.l   INIT_LCDDR2,r0              ;
        mov.w   r0,@(LCDDR,gbr)             ;
        mov.l   INIT_LCDDR2,r0              ;
        mov.w   r0,@(LCDDR,gbr)             ;

        mov     #3,r0                       ; Selects bank R3.
        mov.w   r0,@(LCDAR,gbr)             ;
        mov.l   INIT_LCDDR3,r0              ;
        mov.w   r0,@(LCDDR,gbr)             ;
        mov.l   INIT_LCDDR3,r0              ;
        mov.w   r0,@(LCDDR,gbr)             ;

        mov     #4,r0                       ; Selects bank R4.
        mov.w   r0,@(LCDAR,gbr)             ;
        mov.l   INIT_LCDDR4,r0              ;
        mov.w   r0,@(LCDDR,gbr)             ;
        mov.l   INIT_LCDDtR4,r0             ;
        mov.w   r0,@(LCDDtR,gbr)            ;
```

Figure A.2 LCD Test Program (cont)

```
mov     #5,r0                   ; Selects bank R5.
mov.w   r0,@(LCDAR,gbr)         ;
mov.l   INIT_LCDDR5,r0          ;
mov.w   r0,@(LCDDR,gbr)         ;

mov     #6,r0                   ; Selects bank R6.
mov.w   r0,@(LCDAR,gbr)         ;
mov.l   INIT_LCDDR6,r0          ;
mov.w   r0,@(LCDDR,gbr)         ;

mov     #1,r0                   ; Selects bank R1.
mov.w   r0,@(LCDAR,gbr)         ;
mov.l   #INIT_LCDDR1,r0         ;
mov.w   r0,@(LCDDR,gbr)         ;

bra     LCD_TEST                ;
nop                             ;
        .align 4
$$TOP1:                         .data.l   h'00010000
$$END1:                         .data.l   h'00012580
$$TOP2:                         .data.l   h'00012580
$$END2:                         .data.l   h'00014B00
$$TOP3:                         .data.l   h'00014B00
$$END3:                         .data.l   h'00017080
$$TOP4:                         .data.l   h'00017080
$$END4:                         .data.l   h'00019600
LCD_FH_TOP_ADR:                 .data.l   h'00010000
LCD_FH_BOTTOM_ADR:              .data.l   h'00019600
LCD_GFR:                        .data.l   h'A4000000

INIT_LCDDR1:                    .data.l   h'00000005
INIT_LCDDR2:                    .data.l   h'0000000A
INIT_LCDDR3:                    .data.l   h'0000000F

INIT_LCDMDR1:                   .data.l   h'00000001
INIT_LCDMDR4:                   .data.l   h'0000093F

INIT_LCDDR4:                    .data.l   h'00000540
#INIT_LCDDR4:                   .data.l   h'00000550
INIT_LCDDR2:                    .data.l   h'00007779
INIT_LCDDR3:                    .data.l   h'00000177
INIT_LCDDR4:                    .data.l   h'0000013F
INIT_LCDDR5:                    .data.l   h'0000013F
INIT_LCDDR6:                    .data.l   h'0000E300

PCCR:                           .data.l   h'A4000104
PICR:                           .data.l   h'A4000106

        .end
```

Figure A.2 LCD Test Program (cont)

A.4 PCMCIA

A.4.1 PCMCIA Test Method 1 (Using SRAM Card)

The following describes how to test the PCMCIA by the command-line operation from the monitor program. This test uses a 2-Mbyte SRAM card. Before the PCMCIA test, the SH7707 registers must be initialized.

The following two items are tested:

- Whether the attribute memory can be read
- Whether the common memory can be read or written to

(1) SRAM Card

Use the ML-2M-TB4N 2-Mbyte SRAM card manufactured by Hitachi Maxell, Ltd. Figure A.3 shows the memory area allocation in the SRAM card.

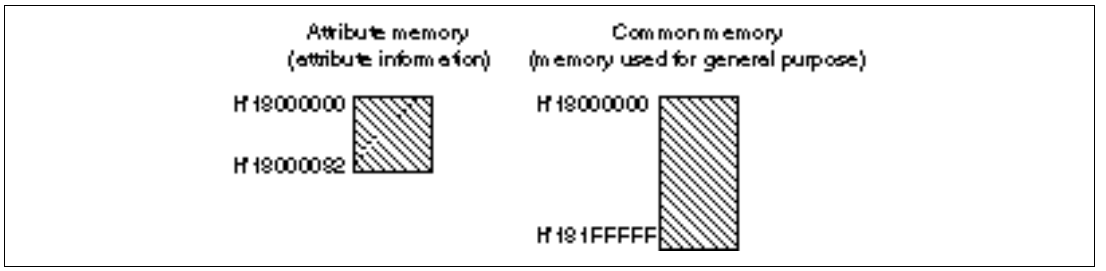


Figure A.3 Memory Area Allocation in 2-Mbyte SRAM Card

(2) Test Procedure

(a) The PCMCIA operates in the little endian mode. Check the jumper settings.

(b) Insert the card into the card slot.

(c) Start up the monitor program (see section A.1).

(d) Enter MW FFFFFF62 2FFF0.

(e) Initialize the port to be used.

(i) Enter MW 04000108 0005 to initialize the PECR.

(ii) Enter M 04000128 02 to initialize the I/O setting.

(This port is used to control power supply. By these settings, 5-V power supply will be used.)

(f) Initialize the register to be used to read the attribute memory.

Enter M 040000E2 80.

(By this setting, the attribute memory is selected and the level shifter gate is opened.)

Here, enter M 040000E0 to check the current card status, such as whether the card is detected, or the battery status.

(g) Read the attribute memory.
The attribute memory is allocated to addresses H'18000000 to H'18000082, and the memory contents can be accessed at even addresses.

(h) Initialize the register to be used to read and write to the common memory.
Enter M 040000E2 81.

(i) Write to some addresses at random in the common memory, then read the contents of the addresses to check whether the data have been correctly written to the common memory.
(The common memory of the 2-Mbyte SRAM card is allocated to addresses H'18000000 to H'181FFFFF. Therefore, read and write within these addresses.)

A.4.2 PCMCIA Test Method 2 (Using SRAM Card and Test Program)

The following test program writed and reads data to and from the common memory, which is used for general purposes, in the 2-Mbyte SRAM card. The program must be loaded and executed from the monitor program described in appendix A.1.

After the test, the result is displayed as

(1) OK when the SRAM card has been correctly read and written to in byte and word units
(2) BAT when the test has been failed or no card has been inserted
(3) NG when the card is write-protected.

```
;=====================================================================
;
;       PCMCIA(SRAM CARD) test program for Windows-CE
;       <<copyright(c) Hitachi,Ltd.,1996-1997 All rights reserved.>>
;
;=====================================================================
      .org    h'0c000000

BSC_SET:
      mov.l   BSC_ADR,r1          ;
      mov.w   #h'2ff0,r2          ;
      mov.w   r2,@r1              ;

      mov.l   #h'0000ffff,r4      ;
wait0:
      dt      r4
      bf      wait0               ;

PORT1_SET:
      mov.l   PORT1_ADR,r1        ;
      mov.w   #h'0005,r2          ;
      mov.w   r2,@r1              ;

      mov.l   #h'0000ffff,r4      ;
wait1:
      dt      r4                  ;
      bf      wait1               ;

PORT2_SET:
      mov.l   PORT2_ADR,r1        ;
      mov.b   #h'02,r2            ;
      mov.b   r2,@r1              ;

      mov.l   #h'0000ffff,r4      ;
wait2:
      dt      r4                  ;
      bf      wait2               ;

ATR_SET:
      mov.l   ATR_ADR,r1          ;
      mov.b   #h'80,r2            ;
      mov.b   r2,@r1              ;
```

Figure A.4 PCMCIA Test Program

```
      mov.l   #h'0000ffff,r4              ;
wait3:
      dt      r4                          ;
      bf      wait3                       ;

      bra     SCI                         ;
      nop                                 ;

      .align 4

BSC_ADR:      .data.l  h'ffffff62
PORT1_ADR:    .data.l  h'04000108
PORT2_ADR:    .data.l  h'04000128
ATR_ADR:      .data.l  h'040000e2

;--------------------------------------
;            SRAM CARD TEST
;--------------------------------------
SCI:
      bra     SCI_INIT                    ;
      nop                                 ;

SRAM_CARD_STATUS1:
      mov.l   CARD_STATUS_ADR,r9          ;
      mov.b   #h'b3,r2                    ;
      mov.b   @r9,r3                      ;
      cmp/eq  r3,r2                       ;
      bt      SRAM_CARD_TEST              ;
      bra     SRAM_CARD_STATUS2           ;
      nop                                 ;

SRAM_CARD_STATUS2:
      mov.l   CARD_STATUS_ADR,r9          ;
      mov.b   #h'f3,r2                    ;
      mov.b   @r9,r3                      ;
      cmp/eq  r3,r2                       ;
      bt      SEND_NG1_START_0            ;

      bra     SEND_START2                 ;
      nop                                 ;

SEND_NG1_START_0:
      bra     SEND_NG1_START              ;
      nop                                 ;
```

Figure A.4 PCMCIA Test Program (cont)

```
SRAM_CARD_TEST:
      mov.l   COMMON_ADR,r1              ;
      mov.b   #h'81,r2                   ;
      mov.b   r2,@r1                     ;
      mov.l #h'0000ffff,r4               ;
wait4:
      dt      r4                         ;
      bf      wait4                      ;

      mov.l  SRAM_CARD_TOP_ADR,r10       ;SRAM CARD START ADDRESS
      mov.l  SRAM_CARD_END_ADR,r11       ;SRAM CARD END ADDRESS
BYTE_ACCESS_TEST_1:
      mov.b  #h'55,r1                    ;
      mov.b  r1,@r10                     ;
      mov.b  @r10,r0                     ;
      cmp/eq #h'55,r0                    ;
      bf     SEND_START2                 ;
      cmp/eq r10,r11                     ;
      bt     BYTE_ACCESS_TEST_2          ;
      add    #1,r10                      ;
      bra    BYTE_ACCESS_TEST_1          ;
      nop                                ;

BYTE_ACCESS_TEST_2:
      mov.l  SRAM_CARD_TOP_ADR,r10       ;SRAM CARD START ADDRESS
BYTE_ACCESS_TEST_2_1:
      mov.b  #h'aa,r1                    ;
      mov.b  r1,@r10                     ;
      mov.b  @r10,r0                     ;
      cmp/eq #h'aa,r0                    ;
      bf     SEND_START2                 ;
      cmp/eq r10,r11                     ;
      bt     WORD_ACCESS_TEST_1          ;
      add    #1,r10                      ;
      bra    BYTE_ACCESS_TEST_2_1        ;
      nop                                ;

WORD_ACCESS_TEST_1:
      mov.l  SRAM_CARD_TOP_ADR,r10       ;SRAM CARD START ADDRESS
WORD_ACCESS_TEST_1_1:
      mov.w  #h'5555,r1                  ;
      mov.w  r1,@r10                     ;
      mov.w  @r10,r0                     ;
```

Figure A.4 PCMCIA Test Program (cont)

```
        cmp/eq r1,r0                            ;
        bf     SEND_START2                      ;
        cmp/eq r10,r11                          ;
        bt     WORD_ACCESS_TEST_2               ;
        add    #2,r10                           ;
        bra    WORD_ACCESS_TEST_1_1             ;
        nop                                     ;
WORD_ACCESS_TEST_2:
        mov.l  SRAM_CARD_TOP_ADR,r10            ;SRAM CARD START ADDRESS
WORD_ACCESS_TEST_2_1:
        mov.w  #h'aaaa,r1                       ;
        mov.w  r1,@r10                          ;
        mov.w  @r10,r0                          ;
        cmp/eq r1,r0                            ;
        bf     SEND_START2                      ;
        cmp/eq r10,r11                          ;
        bt     SEND_START1                      ;
        add    #2,r10                           ;
        bra    WORD_ACCESS_TEST_2_1             ;
        nop                                     ;

        .align 4

SRAM_CARD_TOP_ADR:      .data.l    h'18000000   ;
SRAM_CARD_END_ADR:      .data.l    h'181ffffe   ;
CARD_STATUS_ADR:        .data.l    h'040000e0   ;
COMMON_ADR:             .data.l    h'040000e2   ;

;--------------------------------------
;       SCI INITIALIZE
;--------------------------------------
SCI_INIT:
        mov.l  SCI_SCSCR2,r8                    ;
        mov.b  @r8,r0                           ;
        and    #h'cf,r0                         ;
        mov.b  r0,@r8                           ;
        and    #h'fc,r0                         ;
        mov.b  r0,@r8                           ;

;serial_mode_registe
        mov.l  SCI_SCSMR2,r8                    ;
        mov.b  #h'00,r0                         ;
        mov.b  r0,@r8                           ;
```

Figure A.4 PCMCIA Test Program (cont)

```
;P(MHz)=8 9600bps #h'19
;BIT_RATE_REGISTE
       mov.l   SCI_SCSMR2,r8
       mov.b   #26,r0                          ;
       mov.b   r0,@r8                          ;

;1BIT_WAIT
       nop                                     ;
       nop                                     ;

;TE RE 1 SET RIT TIE TEIE SET
       mov.l   SCI_SCSCR2,r8                   ;
       mov.b   @r8,r0                          ;
       or      #h'30,r0                        ;
       mov.b   r0,@r8                          ;
       or      #h'cc,r0                        ;
       mov.b   r0,@r8                          ;

       bra     SRAM_CARD_STATUS1               ;
       nop                                     ;

;-------------------------------------
;SCI initialize finish
;-------------------------------------
;-------------------------------------
;SEND START OK
;-------------------------------------
SEND_START1:
       mov.b   #h'4f,r3                        ;O
       mov.l   SCI_SCSSR2,r8                   ;
       mov.w   @r8,r0                          ;
       and     #h'0020,r0                      ;
       mov.l   #h'00000020,r1                  ;
       cmp/eq  r0,r1                           ;
       bf      SEND_START1                     ;
       nop                                     ;

;Write the send data1
       mov.l   SCI_SCFTDR2,r8                  ;
       mov     r3,r0                           ;
       mov.b   r0,@r8                          ;
       mov.l   SCI_SCSSR2,r8                   ;
       mov.w   @r8,r0                          ;
       and     #h'00df,r0                      ;
```

Figure A.4 PCMCIA Test Program (cont)

```
SEND_START1_2:
        mov.b   #h'4b,r3                        ;K
        mov.l   @r8,r0                          ;
        and     #h'0020,r0                      ;
        mov.l   #h'00000020,r1                  ;
        cmp/eq  r0,r1                           ;
        bf      SEND_START1_2                   ;
        nop                                     ;

;Write the send data1_1
        mov.l   SCI_SCFTDR2,r8                  ;
        mov     r3,r0                           ;
        mov.b   r0,@r8                          ;
        mov.l   SCI_SCSSR2,r8                   ;
        mov.w   @r8,r0                          ;
        and     #h'00df,r0                      ;

OK:
       bra     OK
       nop

;-------------------------------------
;SEND START BAT
;-------------------------------------
SEND_START2:
       mov.b   #h'42,r3                         ;B
       mov.l   SCI_SCSSR2,r8                    ;
       mov.w   @r8,r0                           ;
       and     #h'0020,r0                       ;
       mov.l   #h'00000020,r1                   ;
       cmp/eq  r0,r1                            ;
       bf      SEND_START2                      ;
       nop                                      ;

;Write the send data2
       mov.l   SCI_SCFTDR2,r8                   ;
       mov     r3,r0                            ;
       mov.b   r0,@r8                           ;
       mov.l   SCI_SCSSR2,r8                    ;
       mov.w   @r8,r0                           ;
       and     #h'00df,r0                       ;

SEND_START2_2:
       mov.b   #h'41,r3                         ;A
```

Figure A.4 PCMCIA Test Program (cont)

```
        mov.w   @r8,r0                  ;
        and     #h'0020,r0              ;
        mov.l   #h'00000020,r1          ;
        cmp/eq  r0,r1                   ;
        bf      SEND_START2_2           ;
        nop                             ;
;Write the send dara2_2:
        mov.l   SCI_SCFTDR2,r8          ;
        mov     r3,r0                   ;
        mov.b   r0,@r8                  ;
        mov.l   SCI_SCSSR2,r8           ;
        mov.w   @r8,r0                  ;
        and     #h'00df,r0              ;
SEND_START2_3:
        mov.b   #h'54,r3                ;T
        mov.w   @r8,r0                  ;
        and     #h'0020,r0              ;
        mov.l   #h'00000020,r1          ;
        cmp/eq  r0,r1                   ;
        bf      SEND_START2_3           ;
        nop                             ;
;Write the send dara2_3
        mov.l   SCI_SCFTDR2,r8          ;
        mov     r3,r0                   ;
        mov.b   r0,@r8                  ;
        mov.l   SCI_SCSSR2,r8           ;
        mov.w   @r8,r0                  ;
        and     #h'00df,r0              ;
BAT:
        bra     BAT                     ;
        nop                             ;
SEND_NG1_START:
        mov.b   #h'4e,r3                ;N
        mov.l   SCI_SCSSR2,r8           ;
        mov.w   @r8,r0                  ;
        and     #h'0020,r0              ;
        mov.l   #h'00000020,r1          ;
        cmp/eq  r0,r1                   ;
        bf      SEND_NG1_START          ;
        nop                             ;
```

Figure A.4 PCMCIA Test Program (cont)

```
;Write the NG data1
      mov.l   SCI_SCFTDR2,r8              ;
      mov     r3,r0                       ;
      mov.b   r0,@r8                      ;
      mov.l   SCI_SCSSR2,r8               ;
      mov.w   @r8,r0                      ;
      and     #h'00df,r0                  ;
                                          ;
SEND_NG1_2:
      mov.b   #h'47,r3                    ;G
      mov.l   @r8,r0                      ;
      and     #h'0020,r0                  ;
      mov.l   #h'00000020,r1              ;
      cmp/eq r0,r1                        ;
      bf      SEND_NG1_2                  ;
      nop                                 ;

;Write the NG data1_1
      mov.l   SCI_SCFTDR2,r8              ;
      mov     r3,r0                       ;
      mov.b   r0,@r8                      ;
      mov.l   SCI_SCSSR2,r8               ;
      mov.w   @r8,r0                      ;
      and     #h'00df,r0                  ;

NG:
      bra    NG
      nop

;--------------------------------------
;SEND FINISH
;--------------------------------------
      .align 4

SCI_SCSMR2:         .data.l     h'a4000150  ;
SCI_SCBRR2:         .data.l     h'a4000152  ;
SCI_SCSCR2:         .data.l     h'a4000154  ;
SCI_SCFTDR2:        .data.l     h'a4000156  ;
SCI_SCSSR2:         .data.l     h'a4000158  ;
SCI_SCRDR2:         .data.l     h'a400015a  ;
SCI_SCSPTR:         .data.l     h'a400015c  ;

      .end
```

Figure A.4 PCMCIA Test Program (cont)

A.5 Audio Play-Back

A.5.1 Audio Test Program

The following test program outputs a 1-kHz sine wave continuously from the speaker. The program must be loaded and executed from the monitor program described in appendix A.1.

During the test program execution, check that the sine wave is output from the speaker, by using an oscilloscope. If the wave is not correct, adjust it with the volume resistor.

```
;=====================================================================
;
;                AUDIO test program for Windows-CE
;
;          <<copyright(c) Hitachi,Ltd.,1996-1997 All rights reserved>>
;=====================================================================
        .SECTION P,CODE,LOCATE=h'0C000000

AUDIO_TEST:
        bsr     SET_AUDIO_TEST_PORT     ;
        nop                             ;

SET_AUDIO_DATA_ADR:

        mov.l   TRNS_MK_ADR,r0          ;
        mov.l   $$AUDIO_DATA_START,r1   ; Initializes the buffer.
        mov.l   r1,@r0                  ;

AUDIO_DATA_TRNS_EXE:

        bsr     START_TIMER             ;
        nop                             ;

        mov.l   $$AUDIO_DATA_END,r1     ;
        mov.l   TRNS_MK_ADR,r2          ; Determines whether to terminate
        mov.l   @r2,r0                  ; the routine.
        cmp/eq  r0,r1                   ; When the routine is terminated,
        bt      SET_AUDIO_DATA_ADR      ; jumps to the SET_AUDIO_DATA
                                        ; _ADR (returns to the beginning)
        mov.l   TRNS_TARGET_ADR,r10     ;
        mov.l   TRNS_MK_ADR,r1          ;
        mov.l   @r1,r0                  ; Acquires transfer data.
        mov.w   @r0,r2                  ;
        swap.b  r2,r2                   ;
        mov.b   r2,@r10                 ;
        add     #2,r0                   ;
```

Figure A.5 Audio Test Program

```
        bra     AUDIO_DATA_TRNS_EXE     ;
        mov.l   r0,@r1                  ;

;=============================================================================
SET_AUDIO_TEST_PORT:
        mov.l   PTL,r1                  ;
        xor     r0,r0                   ;
        mov.w   r0,@r1                  ;

        mov.l   DACR,r1                 ;
        mov     #h'FF,r0                ;
        rts                             ;
        mov.b   r0,@r1                  ;

        .align 4
DACR:           .data.l h'040000a4
PTL:            .data.l h'04000114
;=============================================================================
START_TIMER:
        mov.l   TIMER_CNT,r0            ;
TIMER_LOOP:
        tst     r0,r0                   ; Down counter
        bt      TIMER_STOP              ;
        bra     TIMER_LOOP              ;
        add     #-1,r0                  ;
TIMER_STOP:
        rts                             ;
        nop                             ;
;=============================================================================

        .align 4
TIMER_CNT:              .data.l h'00000085
TRNS_NX_ADR:            .data.l h'0C000200
TRNS_TARGET_ADR:        .data.l h'040000a0
$$AUDIO_DATA_START:     .data.l AUDIO_DATA_START
$$AUDIO_DATA_END:       .data.l AUDIO_DATA_END

AUDIO_DATA_START:
;-----------------------------------------------------;
;       fs = 10kHz
;       f   =   200Hz
;-----------------------------------------------------;
        .data.w h'8000
        .data.w h'900a
        .data.w h'9fd5
        .data.w h'af1e
        .data.w h'bdaa
        .data.w h'cb3c
        .data.w h'd79f
        .data.w h'e2a0
        .data.w h'ec12
```

Figure A.5 Audio Test Program (cont)

```
        .data.w h'f3d1
        .data.w h'f9bc
        .data.w h'fdbb
        .data.w h'ffbf
        .data.w h'ffbf
        .data.w h'fdbb
        .data.w h'f9bc
        .data.w h'f3d1
        .data.w h'ec12
        .data.w h'e2a0
        .data.w h'd79f
        .data.w h'cb3c
        .data.w h'bdaa
        .data.w h'af1e
        .data.w h'9fd5
        .data.w h'900a
        .data.w h'8000
        .data.w h'6ff6
        .data.w h'602b
        .data.w h'50e2
        .data.w h'4256
        .data.w h'34c4
        .data.w h'2861
        .data.w h'1d60
        .data.w h'13ee
        .data.w h'0c2f
        .data.w h'0644
        .data.w h'0245
        .data.w h'0041
        .data.w h'0041
        .data.w h'0245
        .data.w h'0644
        .data.w h'0c2f
        .data.w h'13ee
        .data.w h'1d60
        .data.w h'2861
        .data.w h'34c4
        .data.w h'4256
        .data.w h'50e2
        .data.w h'602b
        .data.w h'6ff6

AUDIO_DATA_END:
        .data.w h'0000

        .end
```

Figure A.5 Audio Test Program (cont)

A.6 Touch Panel

A.6.1 Touch Panel Test Method

Test the touch panel by operating on the command line of the monitor program as follows:

Initializing Registers

1. Set the port register values.

[Register name]	[Register address]		[Initial value]		[Set value]
PHCR	H'0400010E	=	H'AAAA	→	H'4400
PJCR	H'04000110	=	H'0000	→	H'5004

2. Set the A/D register values.

ADCSR	H'04000090	Check that the value is H'00.
ADCR	H'04000092	Check that the value is H'3F.

Enabling Pen Input Interrupt Detection

1. Set the port output value.

PJDR	H'04000130	=	H'00	→	H'42

2. Check the interrupt detection status.

   Use a digital oscilloscope and check that the voltage on pin 12 of the HD74HC04 (U34) changes from high (3.3 V) to low (0 V) when pressing down the pen on the touch panel.

Enabling X-Direction Location Detection

1. Set the port output value.

PJDR	H'04000130	=	H'42	→	H'00

2. Check the X-direction output voltage.

   Check that the Tr3 collector voltage varies between 0 V and 3.3 V while moving the pen from one side to the other side of the touch panel in the X direction.

3. Set the A/D register value.

ADCSR	H'04000090	=	H'00	→	H'40

4. Preparing for A/D conversion.

   Press down the pen on a point on the panel (hold the pen at the point until the A/D conversion is completed).

5. Start A/D conversion.

ADCSR	H'04000090	=	H'40	→	H'60

   When the return key is pressed after the above setting, A/D conversion starts and stops automatically.

Check that the A/D conversion has been successfully completed (the ADCSR value has changed as follows):

ADCSR	H'04000090	=	H'60	→	H'C0

6. Check the A/D conversion result.

ADDRAH	H'04000080	=	H'XXXX (Upper 10 bits show the converted data)

Enabling Y-Direction Location Detection

1. Set the port output value.

PJDR	H'04000130	=	H'00	→	H'82

2. Check the Y-direction output voltage.

   Check that the Tr5 collector voltage varies between 0 V and 3.3 V while moving the pen from one side to the other side of the touch panel in the Y direction.

3. Set the A/D register value.

ADCSR	H'04000090	=	H'00	→	H'41

4. Preparing for A/D conversion.

   Press down the pen on a point on the panel (hold the pen at the point until the A/D conversion is completed).

5. Start A/D conversion.

ADCSR	H'04000090	=	H'40	→	H'61

   When the return key is pressed after the above setting, A/D conversion starts and stops automatically.

   Check that the A/D conversion has been successfully completed (the ADCSR value has changed as follows):

ADCSR	H'04000090	=	H'61	→	H'C1

6. Check the A/D conversion result.

ADDRAH	H'04000084	=	H'XXXX (Upper 10 bits show the converted data)

When testing the touch panel, be sure to initialize the registers (step A) first. Steps B, C, and D can be performed in any order.

A.7 SCI

A.7.1 SCIF Initialization Procedure

Figure A.6 shows an example of SCIF initialization procedure.

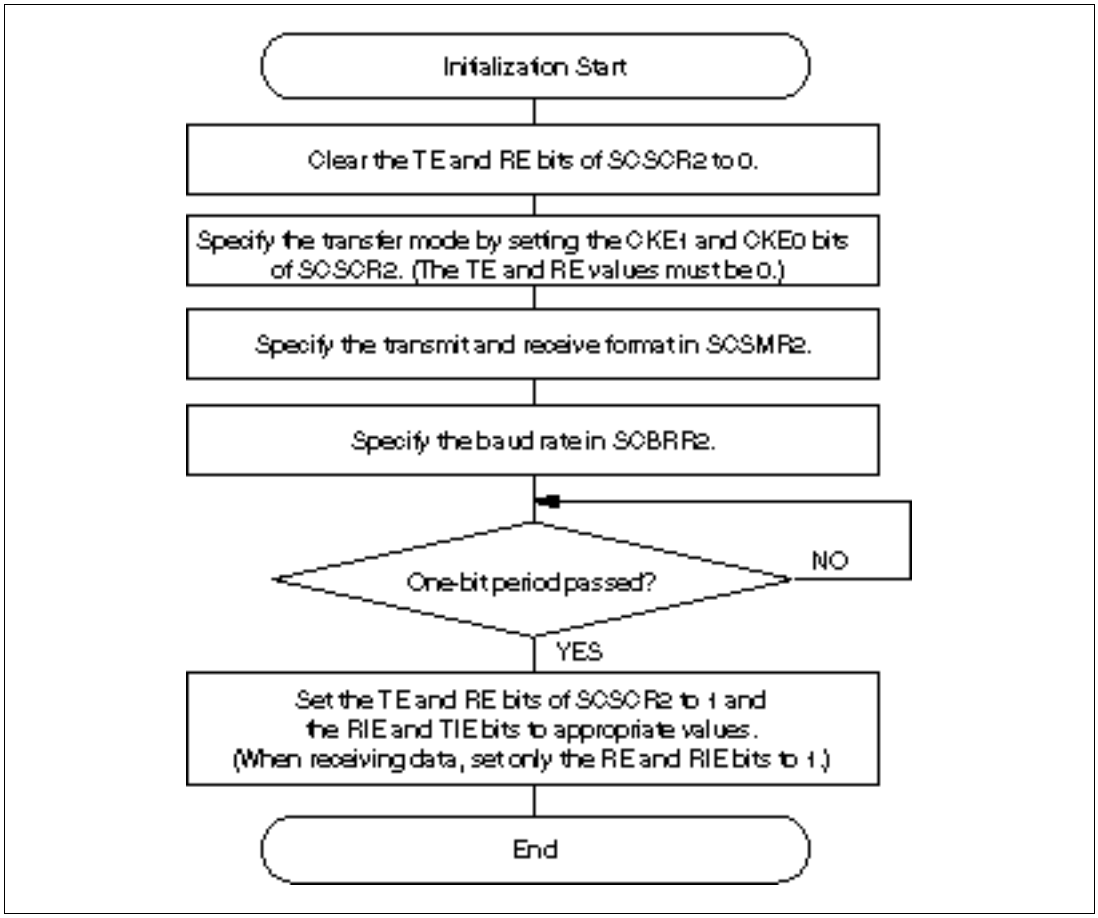


Figure A.6 SCIF Initialization Flowchart

A.7.2 SCIF Transmission Procedure

Figure A.7 shows an example of SCIF transmission procedure.

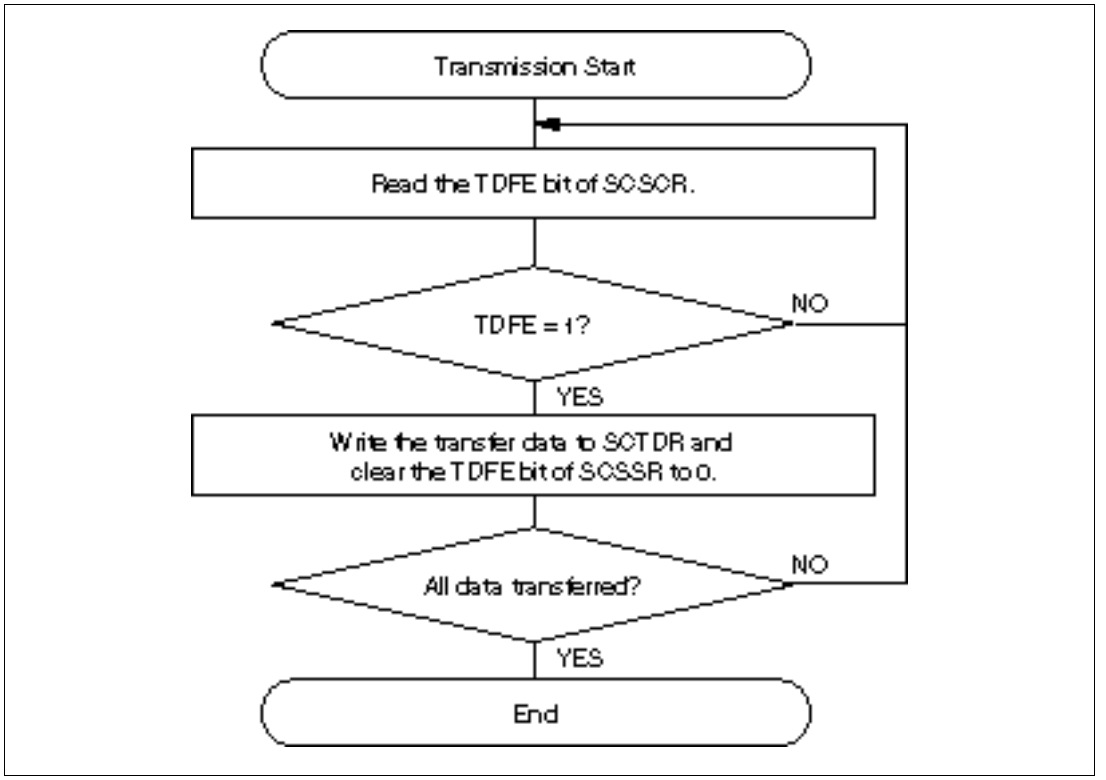


Figure A.7 SCIF Transmission Flowchart

A.7.3 SCIF Receiving Flowchart

Figure A.8 shows the SCIF receiving procedure.

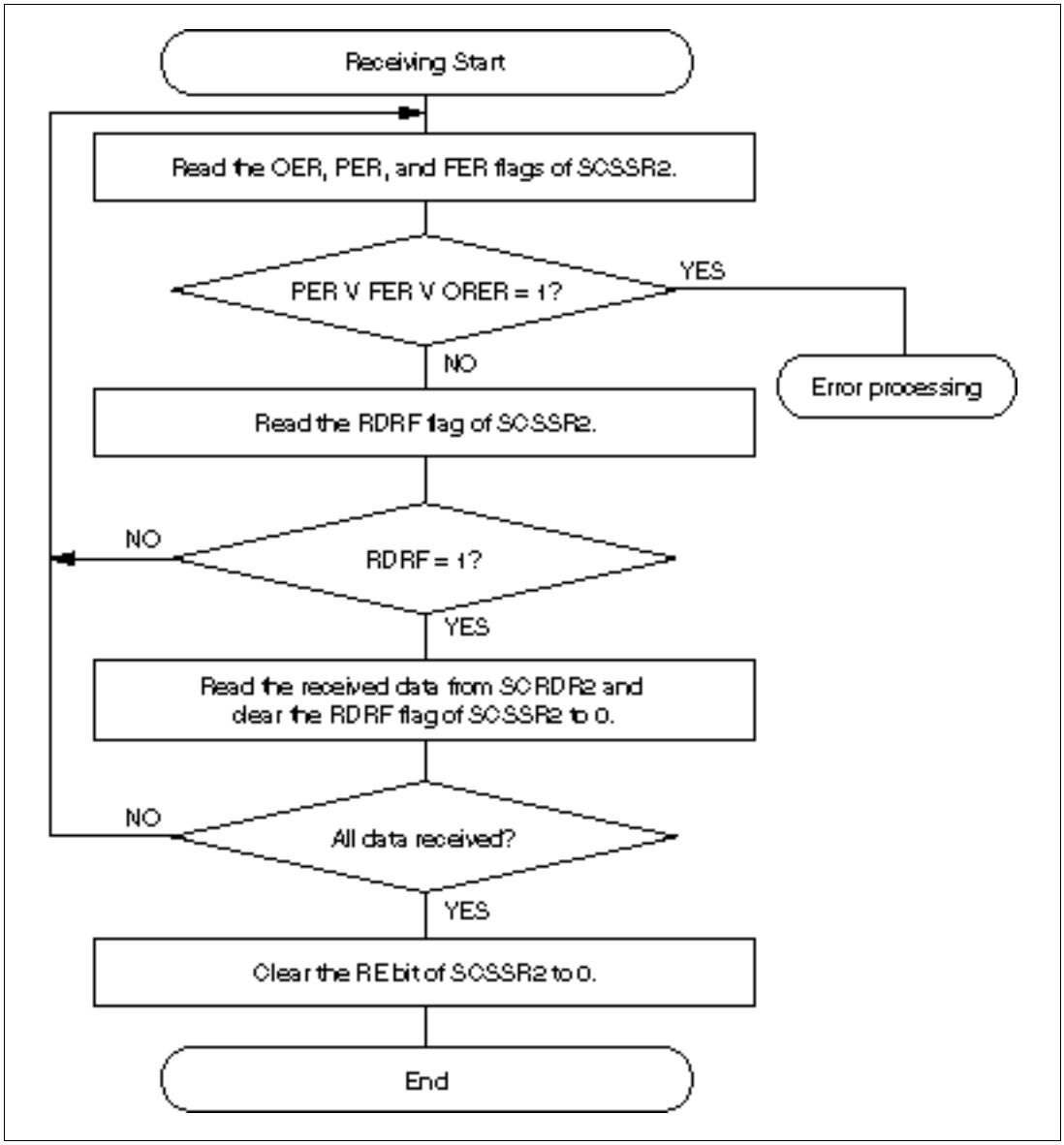


Figure A.8 SCIF Receiving Flowchart

A.7.4 Sample Program for SCIF Transmission

The following sample program outputs “N” sequentially from the SCIF port. The program must be loaded and executed from the monitor program described in appendix A.1.

```
;---------------------------------------------------------------------------
;
;                       SCIF test program for Hitachi-CE
;       <<copyright (c) Hitachi, Ltd., 1996-1997 All rights reserved.>>
;
;-------------------------------------------
        .org    h'00000000
        mov.l   HADR5F,sp       ;

;-------------------------------------------
;SCI Initialization
;-------------------------------------------
SCI_INIT:
     mov.l   SCI_SCSCR2,r5
     mov.b   @r5,r0          ;Clears the TE and RE bits
     And     #H'CF,r0        ;of SCSCR
     mov.b   r0,@r5
     And     #H'FC,r0        ; Specifies the asynchronous mode and internal clock
     mov.b   r0,@r5          ; by setting the CKE1 and CKE0 bits.

;Serial mode register
     mov.l   SCI_SCSMR2,r5
     mov.b   #H'00,r0        ; Specifies the transmission and receiving format
     mov.b   r0,@r5

;Bit rate register
     mov.l   SCI_SCBRR2,r5
     mov.b   #H'19,r0
     mov.b   r0,@r5
     nop

;Wait for 1-bit period
     nop
     nop
     mov.l   SCI_SCSCR2,r5
     mov.b   @r5,r0
     OR      #H'30,r0        ;Sets TE and RE to 1
     mov.b   r0,@r5
     OR      #H'CC,r0        ;Sets RIE, TIE, TEIE, and MPIE to 1
     mov.b   r0,@r5
     nop
;-------------------------------------------
;SCI Initialization End
;-------------------------------------------
```

Figure A.9 Sample Program for SCIF Transmission

```
;---------------------------------------------
;Transmission Processing Start (Transmits the R5 contents)
;---------------------------------------------
SEND_START1:
     mov.b   #'4E,r5          ; Outputs 4E (= 'N').
     mov.l   SCI_SCSSR2,r8
     mov.w   @r8,r0
     and     #'0020,r0        ; Extracts the TDFE bit from SCSSR2.
     mov.l   #'00000020,r1
     cmp/eq  r0,r1            ; Transmission ready?
     bf      SEND_START1      ; Waits until transmission is ready.
     nop

;Writes transmission data to SCFTDR2
     mov.l   SCI_SCFTDR2,r8
     mov     r5,r0            ;
     mov.b   r0,@r8           ; Outputs the (R5).
     mov.l   SCI_SCSSR2,r8
     mov.w   @r8,r0
     and     #'00BF,r0
     mov.w   r0,@r8           ; Clears the TDFE bit to 0.
     bra     SEND_START1      ; Goes to the beginning of the transmission processing.
     nop

;---------------------------------------------
;Transmission Processing End
;---------------------------------------------
     .align 4

RAMEND:      .data.l          h'0CFFFFFC      ;DPRAM last address

SCI_SCSMR2:  .data.l          h'A4000150      ;Serial mode register
SCI_SCBRR2:  .data.l          h'A4000152      ;Bit rate register
SCI_SCSCR2:  .data.l          h'A4000154      ;Serial control register
SCI_SCFTDR2: .data.l          h'A4000156      ;Transmission data register
SCI_SCSSR2:  .data.l          h'A4000158      ;Serial status register
SCI_SCRDR2:  .data.l          h'A400015A      ;Received data register
SCI_SCSPTR2: .data.l          h'A400015C      ;Serial port register

        .end
```

Figure A.9 Sample Program for SCIF Transmission (cont)

A.7.5 Sample Program for SCIF Receiving

The following sample program receives data through the SCIF port, then outputs the received data through the SCIF port. The program must be loaded and executed from the monitor program described in appendix A.1.

```
;---------------------------------------------------------------------------
;                       SCIF test program for Hitachi-CE
;       <<copyright (c) Hitachi, Ltd., 1996-1997 All rights reserved.>>
        .org    h'00000000
;---------------------------------------------------------------------------
;-------------------------------------------
;SCI Initialization
;-------------------------------------------
SCI_INIT:
     mov.l   SCI_SCSCR2,r5
     mov.b   @r5,r0              ;Clears the TE and RE bits
     And     #H'CF,r0            ;of SCSCR
     mov.b   r0,@r5
     And     #H'FC,r0            ;Specifies the asynchronous mode and internal clock
     mov.b   r0,@r0              ;by setting the CKE1 and CKE0 bits

;Serial mode register
     mov.l   SCI_SCSMR2,r5
     mov.b   #H'00,r0            ;Specifies the transmission and receiving format
     mov.b   r0,@r5

;Bit rate register
     mov.l   SCI_SCBRR2,r5
     mov.b   #H'1F,r0
     mov.b   r0,@r5
     nop

;Wait for 1-bit period
     nop
     nop
     mov.l   SCI_SCSCR2,r5
     mov.b   @r5,r0
     OR      #H'30,r0            ;Sets TE and RE to 1
     mov.b   r0,@r5
     OR      #H'0C,r0            ;Sets RIE, TIE, TEIE, and MPIE to 1
     mov.b   r0,@r5
     nop
;-------------------------------------------
;SCI Initialization End
;-------------------------------------------
```

Figure A.10 Sample Program for SCIF Receiving

```
;-------------------------------------------
;Receiving Processing Start
;-------------------------------------------
RECV_START1:
     mov.l   SCI_SCSSR2,r8
     mov.w   @r8,r0          ;Reads the ER, BRK, and DR flags of SCSSR2.
     tst     #H'0091,r0
     bf      RECV_ERROR      ;Goes to error processingif one of the above flags is set to 1.
     nop
     mov.w   @r8,r0          ;Reads the RDF flag of SCSSR2.
     tst     #H'0002,r0      ;Received data stored?
     bt      RECV_START1     ;Returns to the beginning of the routineif data has not been stored.
     nop
     mov.l   SCI_SCFRDR2,r8
     mov.b   @r8,r0          ;Reads the received data.
     mov     r0,r5           ;Saves the received data in R5.
     mov.l   SCI_SCSSR2,r8
     mov.w   @r8,r0          ;Reads the RDRF flag of SCSSR2.
     and     #H'00FD,r0      ;
     mov.w   r0,@r8          ;Clears the RDRF flag to 0.
     bra     SEND_START1
     nop

RECV_ERROR:
     mov.l   SCI_SCSSR2,r
     mov.w   @r8,r0          ;Reads the ORER, PER, and FER flags of SCSSR2.
     and     #H'006Z,r0
     mov.w   r0,@r8          ;Clears the ORER, PER, and FER flags to 0.
     bra     RECV_START1     ;Returns to the beginning of the routine
     nop                     ;to wait for next data receiving
;-------------------------------------------
;Receiving Processing End
;-------------------------------------------
;-------------------------------------------
;Transmission Processing Start (Transmits the R5 contents)
;-------------------------------------------
SEND_START1:
     mov.l   SCI_SCSSR2,r8
     mov.w   @r8,r0
     and     #H'0020,r0      ;Extracts the TDFE bit from SCSSR2.
     mov.l   #H'00000020,r1
     cmp/eq  r0,r1           ;Transmission ready?
     bf      SEND_START1     ;Waits until transmission is ready.
     nop

     mov.l   SCI_SCFTDR2,r8
     mov     r5,r0           ;
     mov.b   r0,@r8          ;Outputs the (R5).
     mov.l   SCI_SCSSR2,r8
     mov.w   @r8,r0
     and     #H'00BF,r0
     mov.w   r0,@r8          ;Clears the TDFE bit to 0.
     bra     RECV_START1     ;Goes to receiving processing.
     nop
;-------------------------------------------
;Transmission Processing End
;-------------------------------------------
```

Figure A.10 Sample Program for SCIF Receiving (cont)

```
        .align 4

MAINSP:         .data.l     h'OCFFFFFC      ;DRAM last address

SCI_SCSMR2:     .data.l     h'A4000150      ;Serial mode register
SCI_SCBRR2:     .data.l     h'A4000152      ;Bit rate register
SCI_SCSCR2:     .data.l     h'A4000154      ;Serial control register
SCI_SCFTDR2:    .data.l     h'A4000156      ;Thereremission data register
SCI_SCSSR2:     .data.l     h'A4000158      ;Serial status register
SCI_SCFRDR2:    .data.l     h'A400015A      ;Received data register
SCI_SCSPTR2:    .data.l     h'A400015C      ;Serial port register

        .end
```

Figure A.10 Sample Program for SCIF Receiving (cont)

A.8 IrDA

A.8.1 Transmission Flowchart

Figure A.11 shows the transmission procedure.

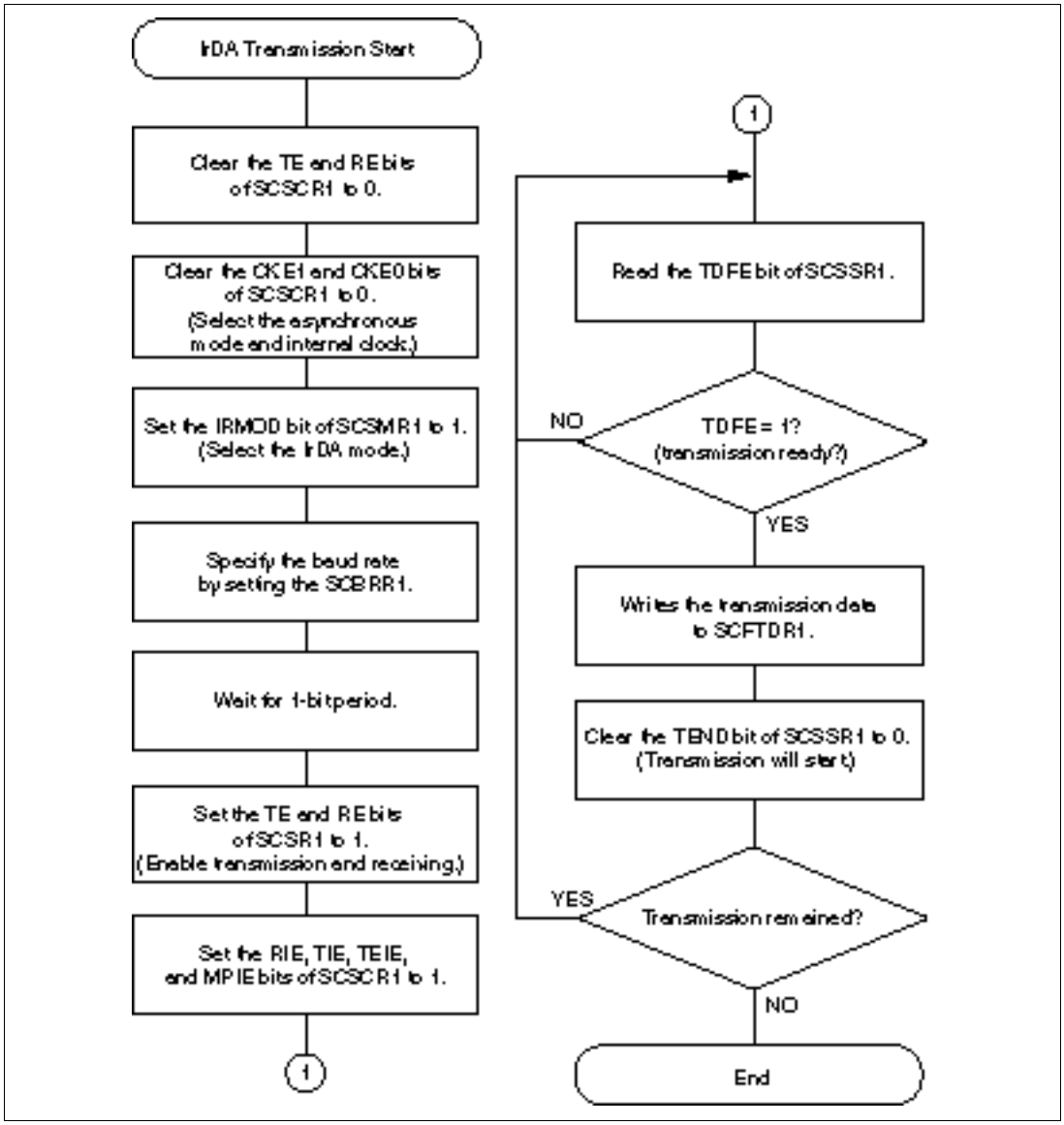


Figure A.11 Transmission Flowchart

A.8.2 Sample Program for Transmission

The following sample program outputs “C” in serial from the IrDA port. The program must be loaded and executed from the monitor program described in appendix A.1.

```
;=====================================================
;               IrDA send test program for Hitachi SH
;       <<copyright (c) Hitachi, Ltd., 1996-1997 All rights reserved.>>
;=====================================================
     .org     h'00000000

    mov.l    #MAINSP,sp                ;
;-----------------------------------------------
;IrDA Initialization
;-----------------------------------------------
    mov.l    SCI_SCSCR1,r5
    mov.b    @r5,r0                    ;Clears the TE and RE bits
    and      #H'CF,r0                  ;of SCSCR
    mov.b    r0,@r5
    and      #H'FC,r0                  ;Specifies the asynchronous mode and internal clock
    mov.b    r0,@r5                    ;by setting the CKE1 and CKE0 bits

;Serial mode register
    mov.l    SCI_SCSMR1,r5
    mov.b    #H'80,r0                  ;Specifies the transmission and receiving format
    mov.b    r0,@r5

;Bit rate register
    mov.l    SCI_SCBRR1,r5
    mov.b    #H'19,r0
    mov.b    r0,@r5
    nop
;Wait for 1-bit period
    nop
    nop
;TE,  RE,  RIE,  TIE,  and TEIE set to 1
    mov.l    SCI_SCSCR1,r5
    mov.b    @r5,r0
    OR       #H'30,r0                  ;Sets TE and RE to 1
    mov.b    r0,@r5
    OR       #H'0C,r0                  ;Sets RIE, TIE, TEIE, and MPIE to 1
    mov.b    r0,@r5
    nop
;-----------------------------------------------
;IrDA Initialization End
;-----------------------------------------------
```

Figure A.12 Sample Program for Transmission

```
;Character string transmission
SEND_DATA:
    mov.b   #H'43,r5                ;Outputs 'C'
    bsr     IRDA_SEND_START1
    nop
    bra     SEND_DATA
    nop

;-----------------------------------------
;Transmission Processing Start (Transmits the R5 contents)
;-----------------------------------------
IRDA_SEND_START1:
    mov.l   SCI_SCSSR1,r6
    mov.w   @r6,r0
    and     #H'0020,r0              ;Extracts the TDFE bit from SCSSR1.
    mov.l   #H'00000020,r1
    cmp/eq  r0,r1                   ;Transmission ready?
    bf      IRDA_SEND_START1        ;Waits until transmission is ready.
    nop
;Data transmission to SCFTDR1.
    mov.l   SCI_SCFTDR1,r6
    mov     r5,r0                   ;
    mov.b   r0,@r6                  ;Outputs the (R5).
    mov.l   SCI_SCSSR1,r6
    mov.w   @r6,r0
    and     #H'00ff,r0
    mov.w   r0,@r6                  ;Clears the TDFE bit to 0.
;bra IRDA_SEND_START1               ;Goes to the beginning of the transmission processing.
    nop

    rts
    nop
;-----------------------------------------
;IrDA Transmission Processing End
;-----------------------------------------
    .align 4
MAINSP:         .data.l     h'0CFFFFFC  ;DRAM last address

;IrDA SCI
SCI_SCSMR1:     .data.l     h'A4000140  ;Serial mode register
SCI_SCBRR1:     .data.l     h'A4000142  ;Bit rate register
SCI_SCSCR1:     .data.l     h'A4000144  ;Serial control register
SCI_SCFTDR1:    .data.l     h'A4000146  ;Transmission data register
SCI_SCSSR1:     .data.l     h'A4000148  ;Serial status register
SCI_SCRDR1:     .data.l     h'A400014A  ;Received data register
SCI_SCSPTR1:    .data.l     h'A400014C  ;Serial port register

    .end
```

Figure A.12 Sample Program for Transmission (cont)

A.8.3 Receiving Flowchart

Figure A.13 shows the receiving procedure.

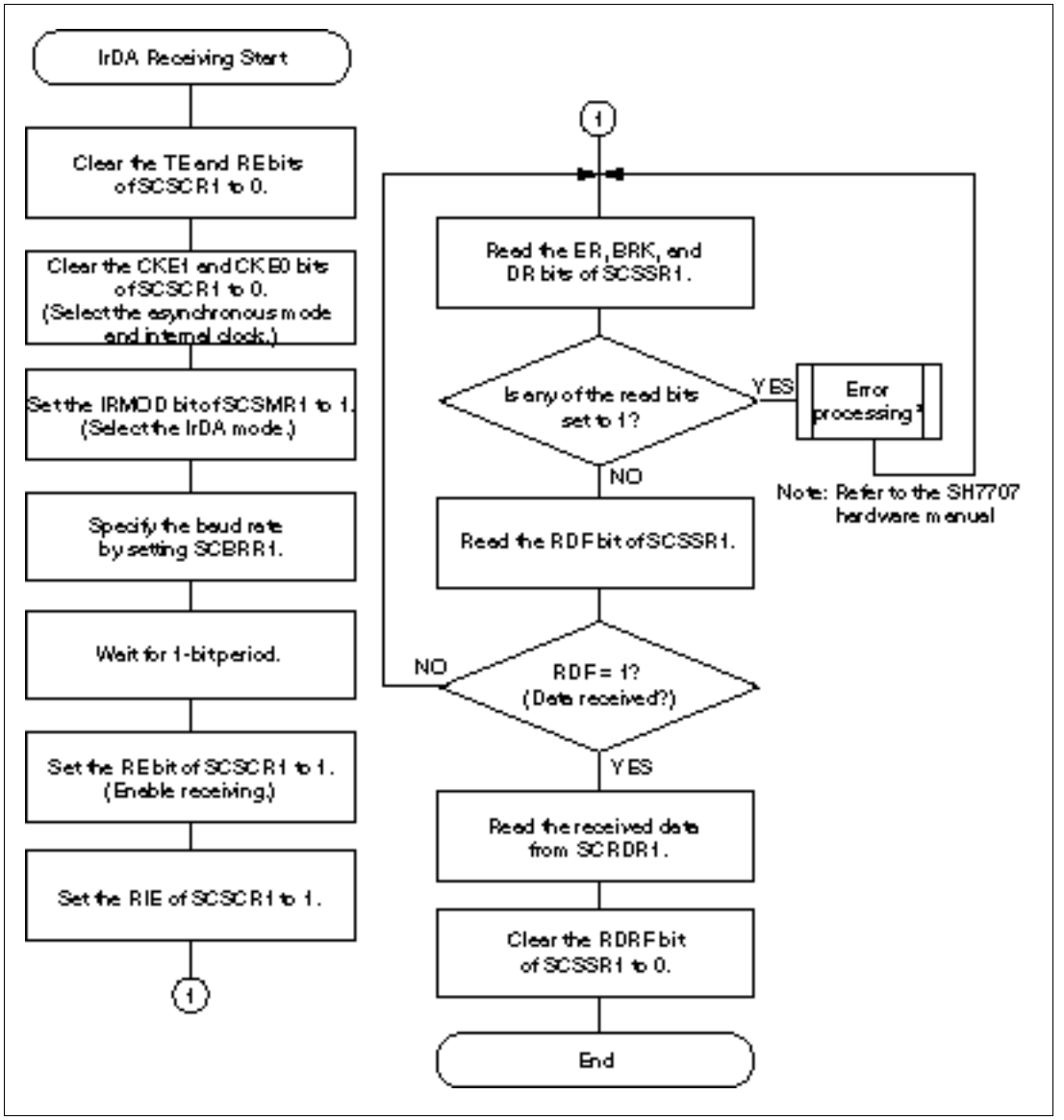


Figure A.13 Receiving Flowchart

A.8.4 Sample Program for Receiving

The following sample program receives character string data through the IrDA port, then outputs the received data through the SCI port. The program must be loaded and executed from the monitor program described in appendix A.1.

```
;---------------------------------------------
;       IrDA receive test program for Hitachi H8
;   <<copyright (c) Hitachi, Ltd., 1996-1997 All rights reserved.>>
;---------------------------------------------
    .org    h'00000000

    mov.l   #MAINSP,sp              ;
;---------------------------------------------
;IrDA Initialization
;---------------------------------------------
SCI_INIT:
    mov.l   #SCI_SCSCR1,r5
    mov.b   @r5,r0                  ;Clears the TE and RE bits
    and     #h'CF,r0                ;of SCSCR1.
    mov.b   r0,@r5
    and     #h'FC,r0                ;Specifies the asynchronous mode and
    mov.b   r0,@r5                  ;internal clock by setting the CKE1 and CKE0 bits.

;Serial mode register
    mov.l   #SCI_SCSMR1,r5
    mov.b   #h'50,r0                ;Sets SCSMR1 register
    mov.b   r0,@r5

;Bit rate register
    mov.l   #SCI_SCBRR1,r5
    mov.b   #h'12,r0
    mov.b   r0,@r5
    nop

;Wait for 1-bit period
    nop
    nop

    mov.l   #SCI_SCSCR1,r5
    mov.b   @r5,r0
    OR      #h'10,r0                ;Sets RE to 1.
    and     #h'DF,r0                ;Sets TE to 0.
    mov.b   r0,@r5
    OR      #h'40,r0                ;Sets RIE to 1.
    mov.b   r0,@r5
    nop
;---------------------------------------------
;IrDA Initialization End
;---------------------------------------------
```

Figure A.14 Sample Program for Receiving

```
;---------------------------------------
;IrDA Receiving Processing Start
;---------------------------------------
RECV_START1:
    mov.l   SCI_SCSSR1,r5
    mov.w   @r5,r0                  ;Reads the ER, BRK, and DR flags of SCSSR1.
    tst     #H'0091,r0
    bf      RECV_ERROR              ;Goes to error processing
    nop                             ;if one of the above flags is set to 1.

RECV_START2:
    nop
    mov.l   SCI_SCSSR1,r5
    mov.w   @r5,r0                  ;Reads the RDF flag of SCSSR1.

    tst     #H'0002,r0              ;Received data stored?
    bt      RECV_START1             ;Returns to the beginning of the routine
    nop                             ;if data has not been stored.

RECV_START3:
    mov.l   SCI_SCRDR1,r5
    mov.b   @r5,r0                  ;Reads the received data.
    mov     r0,r3                   ;Saves the received data in R3.
    mov.l   SCI_SCSSR1,r5
    mov.w   @r5,r0                  ;Reads the RDRF flag of SCSSR1.
    and     #H'00FD,r0              ;
    mov.w   r0,@r5                  ;Clears the RDRF flag to 0.
    bra     SEND_START2
    nop

RECV_ERROR:
    mov.l   SCI_SCSSR1,r5
    mov.w   @r5,r0                  ;Reads the ORER, FER, and PER flags of SCSSR1.
    and     #H'006F,r0
    mov.w   r0,@r5                  ;Clears the ORER, FER, and PER flags to 0.
    bra     RECV_START1             ;Returns to the beginning of the routine
    nop                             ;to wait for next data receiving.
;---------------------------------------
;IrDA Receiving Processing End
;---------------------------------------
;---------------------------------------
;SCI Transmission Processing Start (Transmits the R3 contents)
;---------------------------------------
SEND_START2:
    mov.l   SCI_SCSSR2,r5
    mov.w   @r5,r0
    and     #H'0020,r0              ;Extracts the TDFE bit from SCSSR2.
    mov.l   #H'00000020,r1
    cmp/eq  r0,r1                   ;Transmission ready?
    bf      SEND_START2             ;Waits until transmission is ready.
    nop

    mov.l   SCI_SCFTDR2,r5
    mov     r3,r0                   ;
    mov.b   r0,@r5                  ;Outputs the (R3).
    mov.l   SCI_SCSSR2,r5
    mov.w   @r5,r0
```

Figure A.14 Sample Program for Receiving (cont)

```
    and     #H'00DF,r0
    mov.b   r0,@r5                  ;Clears the TDFE bit to 0.
    bra     RECV_START1             ;Goes to receiving processing
    nop
    nop
;---------------------------------------
;SCI Transmission Processing End
;---------------------------------------
    .align 4

MAINSP:         .data.l     H'0CFFFFFC  ;DPRAM last address

;IrDA register
SCI_SCSMR1:     .data.l     H'A4000140  ;Serial mode register
SCI_SCBRR1:     .data.l     H'A4000142  ;Bit rate register
SCI_SCSCR1:     .data.l     H'A4000144  ;Serial control register
SCI_SCFTDR1:    .data.l     H'A4000146  ;Transmission data register
SCI_SCSSR1:     .data.l     H'A4000148  ;Serial status register
SCI_SCRDR1:     .data.l     H'A400014A  ;Received data register
SCI_SCSPTR1:    .data.l     H'A400014C  ;Serial port register

;SCI register
SCI_SCSMR2:     .data.l     H'A4000150  ;Serial mode register
SCI_SCBRR2:     .data.l     H'A4000152  ;Bit rate register
SCI_SCSCR2:     .data.l     H'A4000154  ;Serial control register
SCI_SCFTDR2:    .data.l     H'A4000156  ;Transmission data register
SCI_SCSSR2:     .data.l     H'A4000158  ;Serial status register
SCI_SCFRDR2:    .data.l     H'A400015A  ;Received data register
SCI_SCSPTR2:    .data.l     H'A400015C  ;Serial port register

    .end
```

Figure A.14 Sample Program for Receiving (cont)

A.9 FPGA

A.9.1 FPGA On-Chip Function Test Procedure

(1) Install the FPGA on the reference platform and turn on the platform. (The LCD is not necessary. 3-V and 5-V power must be supplied.)

(2) Check that the interrupt pins connected to the FPGA are in the following states:

SHIRQ0_	MR6-3	High
SHIRQ_	MR6-4	High
SHIRQ3_	MR6-6	High

Note: In this test, only the SHIRQ_ is used. The SHIRQ0_ and SHIRQ3_ are pulled up to the power-supply voltage on the platform.

(3) Start the monitor program.

(4) Initialize the LCDC on the command line of the monitor program as follows:

```
>MW 04000104 0
>MW 04000106 0
>MW A40000C0 0
>MW A40000C2 0
>MW A40000C0 1
>MW A40000C2 0310
>MW A40000C0 2
>MW A40000C2 7779
>MW A40000C0 3
>MW A40000C2 0177
>MW A40000C0 4
>MW A40000C2 013F
>MW A40000C0 5
>MW A40000C2 013F
>MW A40000C0 6
>MW A40000C2 E800
```

Note: This test assumes that the CKIO = 32 MHz (8-MHz crystal oscillator used, (I : B : P) = (4 : 4 : 1)).

(5) Tests the M signal output.

Check that the M signal is output from the SH7707, by connecting a probe to MR4-2, which is connected to the SH7707 LD0/PTC0 pin (pin 188), and checking with an oscilloscope. The M signal frequency can be calculated as follows

1/52.08 = 19.2 (ms) $19.2 \times 29/320 = 1.74$ (ms) $1.74 \times 2 = 3.48$ (ms)

Note: This test assumes that the frame frequency is 52.08 Hz, which is used in the example in section 4.2, LCD

(6) Tests the operation of the touch panel sampling timer.

The switching noise caused by the LC driving voltage inversion has an amplitude of 2.5 V and a cycle of 8.5 µs on the reference platform. Therefore, the M signal must be delayed for about 10 µs. This test checks whether the M signal can be delayed for 10 µs by using the FPGA functions.

When CKIO is 32 MHz and timer prescaler value is CLK/64, the timer takes 2 µs to count one. Therefore, 10-µs delay corresponds to five counts in the timer.

The following shows the settings of the sampling timer reload data register and control register.

Address: H'10000038
Name: Sampling timer reload data register (STMRDR0)

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—				
Settings:	—	—	—	—	0	1	0	1

A value from H'0 to H'E can be set in bits 3 to 0 in this data register. The reload counter counts down the input clock cycles, and when the counter value changes from H'0 to H'F, the reload data is loaded from the data register to the counter. Therefore, H'F must not be set to the data register; if attempted, the counter repeats reloading and stops counting.

In this test, enter as follows on the command line to set the data register to 5 (five counts):

```
>M 10000038 5
```

Address: H'10000030
Name: Sampling Timer Control Register (STMCR0)

Bit:	7	6	5	4	3	2	1	0
	UNDIE	UNDF	—	STMD	STMST	STMOE	STMPS1	STMPS0
Setting:	1	0	0	0	1	1	0	1

Set bit 7 to 1 to enable the UNDF interrupt check. Set bit 3 to 1 to output the delayed M signal, and compare the delayed signal with the M signal. To select CLK/64 as the timer prescaler value, set the STMPS1 and STMPS0 bits to 0 and 1, respectively. These setting can be entered on the command line as follows:

```
>M 10000030 8D
```

After entering the above command, compare the M signal (MR4-2) output from the SH7707 with the delayed M signal (MR23-6) by using an oscilloscope to check that the delay is correct (10 µs).

(7) Check the key scan control.

To drive all the key scan output signals low and output strobe signals to all keys, set the key scan output register and the key scan input high/control register as follows:

Address: H'10000028
Name: Key scan output register (KSOR)

Bit:	7	6	5	4	3	2	1	0
	KSOD7	KSOD6	KSOD5	KSOD4	KSOD3	KSOD2	KSOD1	KSOD0
Settings:	0	0	0	0	0	0	0	0

Enter as follows on the command line of the monitor program:

```
>M 10000028 0
```

Address: H'10000020
Name: Key scan input high/control register (KSIHCNR)

Bit:	7	6	5	4	3	2	1	0
	KSIE	KSIF	—	—	—	—	KSID9	KSID8
Settings:	1	0	0	0	0	0	1	1

Enter as follows on the command line of the monitor program:

```
>M 10000020 83
```

The key scan input signals are pulled up to power-supply voltage by hardware, and therefore, the signals are high when no key is pressed. After setting the above, press a key, and a key input interrupt will occur. Check interrupt line SHIRQ1 (MR6-4) by using an oscilloscope. Hold down a key, read the key scan input register on the monitor program, and check, with referring to the key matrix, that the correct line is selected.

(8) Check the 8-bit reload timer.

When the CKIO is 32 MHz and the timer prescaler value is CLK/16, the timer takes 500 ns to count one. This test counts 256 cycles, generates an interrupt at underflow, and checks the interrupt status. This test is performed for timers 0 and 1. The following shows the settings of the timer 0 reload data register and timer 0 control register.

Address: H'10000008
Name: Timer 0 reload data register (TMRDR0)

Bit:	7	6	5	4	3	2	1	0
Settings:	1	1	1	1	1	1	1	0

A value from H'0 to H'FE can be set in this data register. The reload counter counts down the clock, and when the counter value changes from H'0 to H'FF, the reload data is loaded from the data register to the counter. Therefore, H'FF must not be set to the data register; if attempted, the counter repeats reloading and stops counting.

In this test, enter as follows on the command line to set the data register to 256:

```
>M 10000008 FE
```

Address: H'10000000
Name: Timer 0 control register (TMCR0)

Bit:	7	6	5	4	3	2	1	0
	UNDIE	UNDF			TMST	TMOE	TMPS1	TMPS0
Settings:	1	0	0	0	1	1	0	0

Set bit 7 to 1 to enable the UNDF interrupt check. Set bit 3 to 1 to output the delayed M signal, and compare the delayed signal with the M signal. To select CLK/16 as the timer prescaler value, set both the STMPS1 and STMPS0 bits to 0. These setting can be entered on the command line as follows:

```
>M 10000000 8C
```

After entering the above command, check the timer output signal (MR23-7) by using an oscilloscope to check that the timer output interval is correct (256 µs).

A.10 Keyboard

Test the keyboard by using the monitor program. For each one of 60 keys, press the key and check the input and output register values (hexadecimal) on the monitor program, by the following procedure:

(1) Connect the keyboard.

(2) Start up the monitor program.

(3) Start execution from address H'10000000.

(4) Check on the monitor display that the value of the FPGA output register (address: H'10000028) is FF (1111 1111) and the value of the input registers (addresses: H'10000020 and H'10000024) are 3 (11) and FF (1111 1111). (This indicates that no key is pressed.)

(5) Press a key.

The bits corresponding to the key in the input and output registers are changed from 1 (high) to 0 (low). For example, when the key marked *f* in figure A.15 is pressed, the output register value should be displayed as FB (1111 1011) and the input register values should be displayed as 3 (11) and FD (1111 1101) on the monitor display. Check that the displayed values are correct, reset the reference platform, then test other keys by the same procedure.

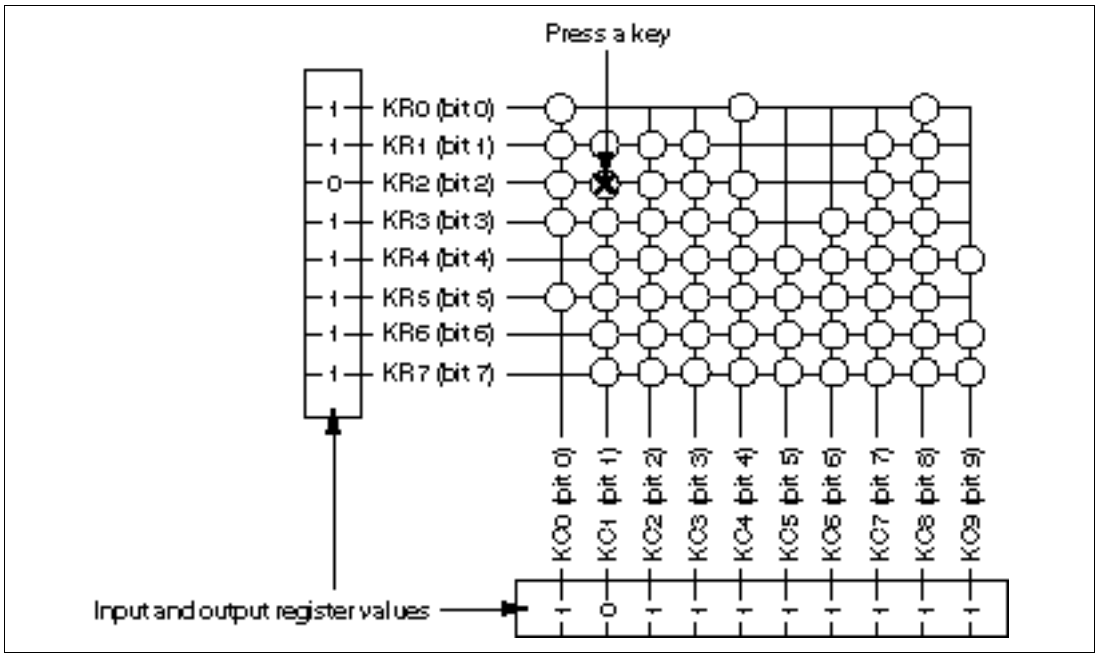


Figure A.15 8 × 10 Keyboard Matrix

Appendix B Circuit Diagrams

B.1 CPU Section

B.2 FPGA and Peripheral Sections

B.3 Flash Memory Section

B.4 DRAM, Touch Panel, and Audio Sections

B.5 Serial, LCD, and IrDA Sections

B.6 PCMCIA Section

B.7 200-Pin Connector

Appendix C Component List

The following table lists the components of the reference platform.

No.	Component	Manufacturer	Quantity	Description
1	SH7707 (HD6417707)	Hitachi, Ltd.	1	CPU
2	EPM7160-EQC100-10	Altera Corporation	1	FPGA
3	MXO-51A 7.37280 MHz	Mita Dempa Co., Ltd.	1	Crystal oscillator
4	LM480172	Sharp Corporation	1	LCD
5	ESA-2P112A	Matsushita Electric Industrial Co., Ltd.	1	Micro-speaker
6	HTC-A1	Nissha Printing Co., Ltd.	1	Touch panel
7	Keyboard for PFM-DS4	—	1	Keyboard
8	HM5165165ATT-6	Hitachi, Ltd.	2	DRAM
9	LT1330CG	Linear Technology Corporation	1	RS-232C driver
10	NJM386M	New Japan Radio Co., Ltd.	1	Audio power amplifier
11	LA5317M	SANYO Electric Co., Ltd.	1	Power supply for LCD
12	MM1180ZM	Mitsumi Electric Co., Ltd.	1	5 V-to-3 V regulator
13	HD151015T	Hitachi, Ltd.	8	Level shifter
14	LTC1472CS	Linear Technology Corporation	1	Power supply controller for PCMCIA
15	TFDS3000	TEMIC Semiconductors	1	IrDA (light-receiving and emitting device)
16	HD74LVC04FP	Hitachi, Ltd.	1	Logic IC
17	HD74LVC08FP	Hitachi, Ltd.	2	Logic IC
18	HD74LVC14FP	Hitachi, Ltd.	1	Logic IC
19	HD74LVC32FP	Hitachi, Ltd.	4	Logic IC
20	C-002RX 32.768 kHz	Seiko Epson Corporation	1	Crystal oscillator
21	IC7A-68PD-1.27SFL-EJL	Hirose Electric Co., Ltd.	1	PCMCIA slot (for 5-V operation)
22	FOLC-150-01-S-Q	SAMTEC	1	200-pin connector (female type)
23	FH10A-10S-1SHB	Hirose Electric Co., Ltd.	1	Keyboard connector (10-pin type)

No.	Component	Manufacturer	Quantity	Description
24	FH10A-8S-1SHB	Hirose Electric Co., Ltd.	4	Keyboard and touch panel connector (8-pin type)
25	FH10A-18S-1SHB	Hirose Electric Co., Ltd.	3	LCD connector (18-pin type)
26	2SC4050	Hitachi, Ltd.	4	Transistor
27	2SA1566	Hitachi, Ltd.	2	Transistor
28	IC197-4807-2000	Yamaichi Electronics Co., Ltd.	1	Flash memory socket
29	PLSKTQ100	Altera Corporation	8	FPGA socket
30	DELC-J9PAF-13L9	Japan Aviation Electronics Industry, Ltd.	1	Dsub connector
31	SMCD-8*200-BD*6-BL-P1.0	Sumitomo Electric Industries, Ltd.	1	Flat cable for keyboard (8-pin type)
32	SMCD-10*200-BD*6-BL-P1.0	Sumitomo Electric Industries, Ltd.	1	Flat cable for keyboard (10-pin type)
33	SMCD-8*100-BD*6-BL-P1.0	Sumitomo Electric Industries, Ltd.	1	Flat cable for touch panel (8-pin type)
34	SMCD-18*100-BD*6-BL-P1.0	Sumitomo Electric Industries, Ltd.	1	Flat cable for LCD (18-pin type)
35	Interface board	—	1	LCD-touch panel interface board
36	AB-15AB	Nihon Kaiheiki Ind. Co., Ltd.	1	SW1
37	DIC152-2P	—	6	Jumper
38	OTB-136-B-5P	—	1	Power-supply terminal
39	1S2074H	Hitachi, Ltd.	1	Diode
40	MRGC09X 4.7 kW	—	36	Resistor
41	MCR10 4.7 kW	Rohm Co., Ltd.	12	Resistor
42	MCR10 600 W	Rohm Co., Ltd.	1	Resistor
43	MCR10 20 kW	Rohm Co., Ltd.	2	Resistor
44	MCR10 33 kW	Rohm Co., Ltd.	12	Resistor
45	MCR10 1 kW	Rohm Co., Ltd.	2	Resistor
46	MCR10 10 kW	Rohm Co., Ltd.	9	Resistor
47	MCR18 2.2 kW	Rohm Co., Ltd.	1	Resistor
48	MCR18 100 W	Rohm Co., Ltd.	1	Resistor
49	MCR18 10 W	Rohm Co., Ltd.	1	Resistor

No.	Component	Manufacturer	Quantity	Description
50	MCR18 33 kW	Rohm Co., Ltd.	1	Resistor
51	MCR18 1 MW	Rohm Co., Ltd.	4	Resistor
52	3296W 10 kW	—	1	Resistor
53	MCH212F104Z	Rohm Co., Ltd.	68	Capacitor
54	MCH212F471Z	Rohm Co., Ltd.	3	Capacitor
55	MCH212F224Z	Rohm Co., Ltd.	2	Capacitor
56	MCH212F105Z	Rohm Co., Ltd.	2	Capacitor
57	SME25VB22M	Nippon Chemi-Con Corporation	1	Capacitor
58	SME25VB4R7M	Nippon Chemi-Con Corporation	3	Capacitor
59	SME25VB220M	Nippon Chemi-Con Corporation	1	Capacitor
60	SME25VB100M	Nippon Chemi-Con Corporation	1	Capacitor
61	SME50VB10M	Nippon Chemi-Con Corporation	6	Capacitor
62	SME25VB33M	Nippon Chemi-Con Corporation	4	Capacitor
63	MCR10 0 W	Rohm Co., Ltd.	2	Resistor
64	HN29WT800T8	Hitachi, Ltd.	8	Flash memory
65	IC26-0803-GS4	Yamaichi Electronics Co., Ltd.	1	Crystal oscillator socket
66	MRGC09X 10 kW	—	1	Resistor
67	MCH212F333Z	Rohm Co., Ltd.	1	Capacitor
68	15pF (chip ceramic capacitor)	—	2	Capacitor

Others: (1) Acrylic case

Appendix D Pull-Up Resistors

The SH7707 pins are pulled up to the power-supply voltage by resistors, except for power-supply, GND, address bus, and output pins. Pull-up resistors are generally used to prevent the pins from entering the high-impedance state, but on the reference platform, these resistors are also used to prevent malfunctions during development and debugging or to protect the signals from noises generated by measuring tools. Therefore, some of these pull-up resistors can be removed from the user products only after the possibility of and actions against adverse effects are well examined.