REJ09B0254-0500

Everywhere you imagine. RENESAS

The revision list can be viewed directly by clicking the title page.

The revision list summarizes the locations of revisions and additions. Details should always be checked by referring to the relevant text.

32

SH7727 Group Hardware Manual

Renesas 32-Bit RISC Microcomputer SuperH™ RISC engine Family/SH7700 Series

SH7727 HD6417727

Hardware Manual

Rev. 5.00
Revision Date: Dec 12, 2005

Renesas Technology
www.renesas.com

Keep safety first in your circuit designs!

1. Renesas Technology Corp. puts the maximum effort into making semiconductor products better and more reliable, but there is always the possibility that trouble may occur with them. Trouble with semiconductors may lead to personal injury, fire or property damage.
   Remember to give due consideration to safety when making your circuit designs, with appropriate measures such as (i) placement of substitutive, auxiliary circuits, (ii) use of nonflammable material or (iii) prevention against any malfunction or mishap.

Notes regarding these materials

1. These materials are intended as a reference to assist our customers in the selection of the Renesas Technology Corp. product best suited to the customer's application; they do not convey any license under any intellectual property rights, or any other rights, belonging to Renesas Technology Corp. or a third party.
2. Renesas Technology Corp. assumes no responsibility for any damage, or infringement of any third-party's rights, originating in the use of any product data, diagrams, charts, programs, algorithms, or circuit application examples contained in these materials.
3. All information contained in these materials, including product data, diagrams, charts, programs and algorithms represents information on products at the time of publication of these materials, and are subject to change by Renesas Technology Corp. without notice due to product improvements or other reasons. It is therefore recommended that customers contact Renesas Technology Corp. or an authorized Renesas Technology Corp. product distributor for the latest product information before purchasing a product listed herein.
   The information described here may contain technical inaccuracies or typographical errors.
   Renesas Technology Corp. assumes no responsibility for any damage, liability, or other loss rising from these inaccuracies or errors.
   Please also pay attention to information published by Renesas Technology Corp. by various means, including the Renesas Technology Corp. Semiconductor home page (http://www.renesas.com).
4. When using any or all of the information contained in these materials, including product data, diagrams, charts, programs, and algorithms, please be sure to evaluate all information as a total system before making a final decision on the applicability of the information and products. Renesas Technology Corp. assumes no responsibility for any damage, liability or other loss resulting from the information contained herein.
5. Renesas Technology Corp. semiconductors are not designed or manufactured for use in a device or system that is used under circumstances in which human life is potentially at stake. Please contact Renesas Technology Corp. or an authorized Renesas Technology Corp. product distributor when considering the use of a product contained herein for any specific purposes, such as apparatus or systems for transportation, vehicular, medical, aerospace, nuclear, or undersea repeater use.
6. The prior written approval of Renesas Technology Corp. is necessary to reprint or reproduce in whole or in part these materials.
7. If these products or technologies are subject to the Japanese export control restrictions, they must be exported under a license from the Japanese government and cannot be imported into a country other than the approved destination.
   Any diversion or reexport contrary to the export control laws and regulations of Japan and/or the country of destination is prohibited.
8. Please contact Renesas Technology Corp. for further details on these materials or the products contained therein.

General Precautions on the Handling of Products

1. Treatment of NC Pins

Note: Do not connect anything to the NC pins.
The NC (not connected) pins are not connected to any of the internal circuitry; they are used as test pins or to reduce noise. If something is connected to the NC pins, the operation of the LSI is not guaranteed.

2. Treatment of Unused Input Pins

Note: Fix all unused input pins to high or low level.
Generally, the input pins of CMOS products are high-impedance input pins. If unused pins are in their open states, intermediate levels are induced by noise in the vicinity, a pass-through current flows internally, and a malfunction may occur.

3. Processing before Initialization

Note: When power is first supplied, the product's state is undefined. The states of internal circuits are undefined until full power is supplied throughout the chip and a low level is input on the reset pin. During the period where the states are undefined, the register settings and the output state of each pin are also undefined. Design your system so that it does not malfunction because of processing while it is in this undefined state. For those products which have a reset function, reset the LSI immediately after the power supply has been turned on.

4. Prohibition of access to undefined or reserved address

Note: Access to undefined or reserved addresses is prohibited.
The undefined or reserved addresses may be used to expand functions, or test registers may have been be allocated to these address. Do not access these registers: the system's operation is not guaranteed if they are accessed.

Preface

The SH7727 microprocessor incorporates the 32-bit SH-3 CPU and is also equipped with peripheral functions necessary for configuring a user system.

The SH7727 is built in with a variety of peripheral functions such as cache memory, memory management unit (MMU), interrupt controller, timers, three serial communication interfaces (SCI, SCIF, SIOF), real-time clock (RTC), user break controller (UBC), bus state controller (BSC) and AFE interface. The SH7727 can be used in a variety of applications that demand a high-speed microcomputer with low power consumption.

The descriptions in this manual are based on the SH7727C. For details on using versions previous to the SH7727B please refer to Using Versions Previous to the SH7727B at the end of the manual. Note that the version is the SH7727C if "C" is engraved on the chip and the version is the SH7727B if "B" is engraved. If there is no such indication the product is a version previous to the SH7727B. (See Appendix E.)

Target Readers: This manual is designed for use by people who design application systems using the SH7727.
To use this manual, basic knowledge of electric circuits, logic circuits and microcomputers is required.

Purpose: This manual provides the information of the hardware functions and electrical characteristics of the SH7727.
The SH-3, SH-3E, SH3-DSP Programming Manual contains detailed information of executable instructions. Please read the Programming Manual together with this manual.

How to Use the Book:

- To understand general functions
  → Read the manual from the beginning.
  The manual explains the CPU, system control functions, peripheral functions and electrical characteristics in that order.
- To understanding CPU functions
  → Refer to the separate SH-3, SH-3E, SH3-DSP Programming Manual.

Explanatory Note: Bit sequence: upper bit at left, and lower bit at right

List of Related Documents: The latest documents are available on our Web site. Please make sure that you have the latest version. (http://www.renesas.com/)

- User manuals for SH7727

Name of Document	Document No.
SH7727 Hardware Manual	This manual
SH-3, SH-3E, SH3-DSP Programming Manual	ADE-602-096B

- User manuals for development tools

Name of Document	Document No.
SuperH™ RISC engine C/C++ Compiler, Assembler, Optimizing Linkage Editor Compiler Package V.9.00 User's Manual	REJ10B0152-0101
SuperH™ RISC engine High-Performance Embedded Workshop 3 User's Manual	REJ10B0025-0200H
SuperH RISC engine High-Performance Embedded Workshop 3 Tutorial	REJ10B0023-0200H

- Application Note

Name of Document	Document No.
SuperH RISC engine C/C++ Compiler Package Application Note	REJ05B0463-0200

<table>
<tr><th>Page</th><th>Previous Version</th><th>Revised Version</th></tr>
<tr><td>5 to 7</td><td>Table 1.1 SH7727 Features

<table>
<tr><th>Item</th><th>Features</th></tr>
<tr><td>Timer (TMU, CMT)</td><td>• 3-channel auto-reload-type 32-bit timer
• 1-channel 16-bit compare match timer
• Choice of six counter input clocks
• Maximum resolution: 2 MHz</td></tr>
</table>
<table>
<tr><th>Item</th><th>Features</th></tr>
<tr><td>Serial communi-cation interface (SCIF)</td><td>• 16-byte FIFO for transmission/reception
• DMA can be transferred
• Hardware flow control</td></tr>
<tr><td>Direct memory access controller (DMAC)</td><td>• 4 channels
• Burst mode and cycle-steal mode
• External request operating mode</td></tr>
</table>
<table>
<tr><th>Item</th><th>Features</th></tr>
<tr><td>LCD controller (LCDC)</td><td>• From 16 x 1 to 1024 x 1024 pixels can be supported
• 1/2/4/6/8/16 bpp (bit per pixel) with 18bit color pallet
• 1/2/4 bpp (bit per pixel) gray scale
• 8-bit Frame rate controller
• TFT/DSTN/STN
• Signal polarity setting function
• Hardware panel rotation
• Power control function
• Selectable clock source (LCLK or Bclk or Pclk)</td></tr>
<tr><td>A/D converter (ADC)</td><td>• 10 bits ± 4 LSB, 6 channels
• Conversion time: 10 µs
• Input range: 0–Vcc (max. 3.6 V)</td></tr>
</table>
Product lineup
<table>
<tr><th rowspan="2">Abb.</th><th colspan="2">Power Supply Voltage</th><th rowspan="2">Operating Frequency</th><th rowspan="2">Model Name</th><th rowspan="2">Package</th></tr>
<tr><th>I/O</th><th>Internal</th></tr>
<tr><td rowspan="4">SH7727</td><td rowspan="2">3.3 ± 0.3 V</td><td rowspan="2">1.7 to 2.05 V</td><td rowspan="2">160 MHz</td><td>HD6417727F160B</td><td>240-pin plastic HQFP (FP-240B)</td></tr>
<tr><td>HD6417727BP160B</td><td>240-pin CSP (BP-240A)</td></tr>
<tr><td rowspan="2">3.1 ± 0.5 V</td><td rowspan="2">1.6 to 2.05 V</td><td rowspan="2">100 MHz</td><td>HD6417727F100B</td><td>240-pin plastic HQFP (FP-240B)</td></tr>
<tr><td>HD6417727BP100B</td><td>240-pin CSP (BP-240A)</td></tr>
</table></td><td>

<table>
<tr><th>Item</th><th>Features</th></tr>
<tr><td>Timer (TMU)</td><td>• 3-channel auto-reload-type 32-bit timer
• Choice of six counter input clocks
• Maximum resolution: 2 MHz</td></tr>
</table>
<table>
<tr><th>Item</th><th>Features</th></tr>
<tr><td>Serial communi-cation interface (SCIF)</td><td>• 16-byte FIFO for transmission/reception
• DMA can be transferred
• On-chip modem control function</td></tr>
<tr><td>Direct memory access controller (DMAC)</td><td>• 4 channels
• Burst mode and cycle-steal mode
• External request operating mode
• 1-channel 16-bit compare match timer</td></tr>
</table>
<table>
<tr><th>Item</th><th>Features</th></tr>
<tr><td>LCD controller (LCDC)</td><td>• From 16 x 1 to 1024 x 1024 pixels can be supported
• 4/8/15/16 bpp (bit per pixel) color modes
• 1/2/4/6 bpp (bit per pixel) gray scale
• 8-bit Frame rate controller
• TFT/DSTN/STN
• Signal polarity setting function
• Hardware panel rotation
• Power control function
• Selectable clock source (LCLK, bus clock (Bø), or peripheral clock (Pø))</td></tr>
<tr><td>A/D converter (ADC)</td><td>• 10 bits ± 4 LSB, 6 channels
• Conversion time: 15 µs
• Input range: 0–Vcc (max. 3.6 V)</td></tr>
</table>
Product lineup
<table>
<tr><th rowspan="2">SH7727</th><th colspan="2">Power Supply Voltage</th><th rowspan="2">Operating Frequency</th><th rowspan="2">Model Name</th><th rowspan="2">Package</th></tr>
<tr><th>I/O</th><th>Internal</th></tr>
<tr><td rowspan="2">160 MHz products</td><td rowspan="2">3.0 V to 3.6 V</td><td rowspan="2">1.70 V to 2.05 V</td><td rowspan="2">160 MHz</td><td>HD6417727F160C</td><td>240-pin plastic HQFP (PRQP0240KC-B)</td></tr>
<tr><td>HD6417727BP160C</td><td>240-pin CSP (PLBG0240JA-A)</td></tr>
<tr><td rowspan="2">100 MHz products</td><td rowspan="2">2.6 V to 3.6 V</td><td rowspan="2">1.60 V to 2.05 V</td><td rowspan="2">100 MHz</td><td>HD6417727F100C</td><td>240-pin plastic HQFP (PRQP0240KC-B)</td></tr>
<tr><td>HD6417727BP100C</td><td>240-pin CSP (PLBG02403A-A)</td></tr>
</table></td></tr>
<tr><td>11 to 19</td><td>Table 1.2 SH7727 Pin Function

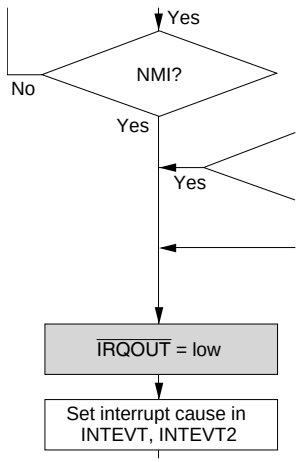
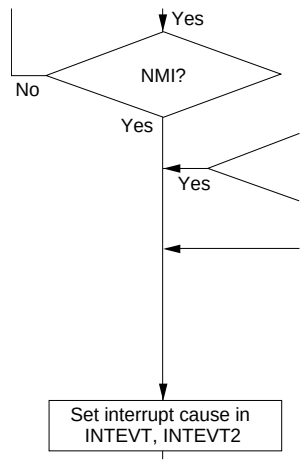
Pin No. (FP-240B) Pin No. (BP-240A)</td><td>

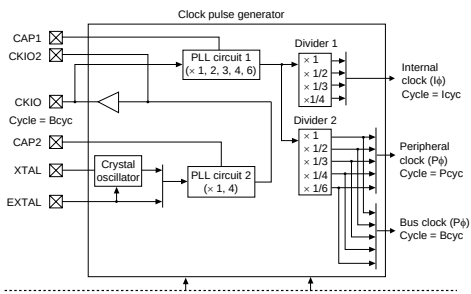
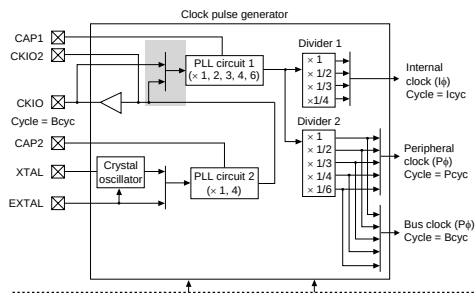
Pin No. (PRQP0240KC-B) Pin No. (PLBG0240JA-A)</td></tr>
<tr><td>167</td><td>7.1.4 Register Configuration

The INTC has 12 registers listed in table 7.2.</td><td>

The INTC has 17 registers listed in table 7.2.</td></tr>
</table>

Page	Previous Version	Revised Version
169	7.2.2 IRQ Interrupt IRQ interrupts are input by priority from pins IRQ0 to IRQ5 with a level or an edge. The priority level can be set by priority setting registers C, D (IPRC, IPRD) in a range from levels 0 to 15. When using edge-sensing for IRQ interrupts, clear the interrupt source by having software read 1 from the corresponding bit in IRR0, then write 0 to the bit.	IRQ interrupts are input by priority from pins IRQ0 to IRQ5 with a level or an edge. The priority level can be set by priority setting registers C, D (IPRC, IPRD) in a range from levels 0 to 15. When using edge sensing for IRQ interrupts, do the following to clear IR0. To clear bits IRQ5R to IRQ0R to 0, read from IRR0 before writing. After confirming that the bits to be cleared to 0 are set to 1, write 0 to them. In this case write 0 only to the bits to be cleared; write 1 to the other bits. The values of the bits to which 1 is written do not change. When level sensing is used for IRQ interrupts, bits IRQ5R to IRQ0R indicate whether or not an interrupt request has been input. They can be set and cleared by the values input to pins IRQ5R to IRQ0R alone.
175	Table 7.4 Interrupt Exception Handling Sources and Priority (IRQ Mode) **Interrupt Source** / **INTEVT Code (INTEVT2 Code)** / **Interrupt Priority (Initial Value)** / **IPR (Bit Numbers)** / **Priority within IPR Setting Unit** / **Default Priority** TMU1 TUNI1 H'420 (H'420) 0–15 (0) IPRA (11–8) — High TMU2 TUNI2 H'440 (H'440) 0–15 (0) IPRA (7–4) High TICPI2 H'460 (H'460) Low RTC ATI H'480 (H'480) 0–15 (0) IPRA (3–0) High PRI H'4A0 (H'4A0) CUI H'4C0 (H'4C0) Low SCI0 ERI H'4E0 (H'4E0) 0–15 (0) IPRB (3–0) High RXI H'500 (H'500) TXI H'520 (H'520) TEI H'540 (H'540) Low Low	**Interrupt Source** / **INTEVT Code (INTEVT2 Code)** / **Interrupt Priority (Initial Value)** / **IPR (Bit Numbers)** / **Priority within IPR Setting Unit** / **Default Priority** TMU1 TUNI1 H'420 (H'420) 0–15 (0) IPRA (11–8) — High TMU2 TUNI2 H'440 (H'440) 0–15 (0) IPRA (7–4) — RTC ATI H'480 (H'480) 0–15 (0) IPRA (3–0) High PRI H'4A0 (H'4A0) CUI H'4C0 (H'4C0) Low SCI0 ERI H'4E0 (H'4E0) 0–15 (0) IPRB (7–4) High RXI H'500 (H'500) TXI H'520 (H'520) TEI H'540 (H'540) Low Low
177	Table 7.5 Interrupt Exception Handling Sources and Priority (IRL Mode) **Interrupt Source** / **INTEVT Code (INTEVT2 Code)** / **Interrupt Priority (Initial Value)** / **IPR (Bit Numbers)** / **Priority within IPR Setting Unit** / **Default Priority** ADC ADI H'200–3C0* (H'980) 0–15 (0) IPRE (3–0) — High LCDC LCDCI H'200–3C0* (H'9A0) 0–15 (0) IPRF (11–8) — TMU1 TUNI1 H'420 (H'420) 0–15 (0) IPRA (11–8) — TMU2 TUNI2 H'440 (H'440) 0–15 (0) IPRA (7–4) High TICPI2 H'460 (H'460) Low Low	**Interrupt Source** / **INTEVT Code (INTEVT2 Code)** / **Interrupt Priority (Initial Value)** / **IPR (Bit Numbers)** / **Priority within IPR Setting Unit** / **Default Priority** ADC ADI H'200–3C0* (H'980) 0–15 (0) IPRE (3–0) — High LCDC LCDCI H'200–3C0* (H'9A0) 0–15 (0) IPRF (11–8) — TMU1 TUNI1 H'420 (H'420) 0–15 (0) IPRA (11–8) — TMU2 TUNI2 H'440 (H'440) 0–15 (0) IPRA (7–4) — Low

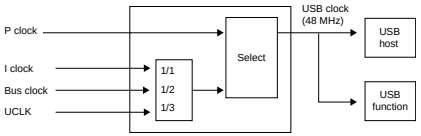
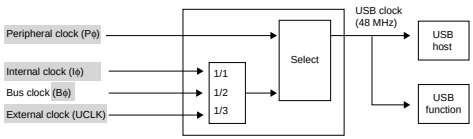
Page	Previous Version	Revised Version
188	7.3.7 Interrupt Request Register 0 (IRR0) The IRR0 is an 8-bit register that indicates interrupt requests from external input pins IRQ0 to IRQ5 and PINT0 to PINT15. This register is initialized to H'00 at power-on reset or manual reset, but is not initialized in standby mode.	When using edge sensing for IRQ interrupts, do the following to clear IR0. To clear bits IRQ5R to IRQ0R to 0, read from IRR0 before writing. After confirming that the bits to be cleared to 0 are set to 1, write 0 to them. In this case write 0 only to the bits to be cleared; write 1 to the other bits. The values of the bits to which 1 is written do not change. When level sensing is used for IRQ interrupts, bits IRQ5R to IRQ0R indicate whether or not an interrupt request has been input. They can be set and cleared by the values input to pins IRQ5R to IRQ0R alone.
198	Figure 7.3 Interrupt Operation Flowchart 	
199	7.4.2 Multiple Interrupts When these procedures are followed in order, an interrupt of higher priority than the one being handled can be accepted after clearing BL in step 4. Figure 7.3 shows a sample interrupt operation flowchart.	When these procedures are followed in order, an interrupt of higher priority than the one being handled can be accepted after clearing BL in step 4.

Page	Previous Version	Revised Version
234	Table 9.1 Power-Down Modes **Mode** / **Transition Conditions** Sleep mode: Execute SLEEP instruction with STBY bit cleared to 0 in STBCR Module standby function: Set MSTP bit of STBCR to 1	Table 9.1 Power-Down Modes **Mode** / **Transition Conditions** Sleep mode: Execute SLEEP instruction with STBY bit cleared to 0 in STBCR*7 Module standby function: Set MSTP bit of STBCR to 1*6 *Note 6, 7, added*
		Section 10.1.2 "Clock Abbreviation" deleted
259	Figure 10.1 Block Diagram of Clock Pulse Generator 	
260	10.2.1 CPG Block Diagram 1. PLL Circuit 1 PLL circuit 1 doubles, triples, quadruples, sextuples, or leaves unchanged the input clock frequency from the CKIO terminal. ...	PLL circuit 1 doubles, triples, quadruples sextuples, or leaves unchanged the input clock frequency from the CKIO pin or PLL circuit 2. …

Page	Previous Version	Revised Version
265, 266	Table 10.4 Available Combination of Clock Mode and FRQCR Values Cautions: 1. The frequency ranges of the input clock and crystal oscillator should be set within the specified frequency range based on the clock rate in table 10.4, and section 32.3, AC Characteristics. 2. The input to divider 1 becomes the output of: • PLL circuit 1 when PLL circuit 1 is on. • PLL circuit 2 when PLL circuit 1 is off and PLL circuit 2 is on. 3. The input of divider 2 becomes the output of: • PLL circuit 1 4. The frequency of the internal clock (Iϕ) becomes: • The product of the frequency of the CKIO pin, the frequency multiplication ratio of PLL circuit 1, and the division ratio of divider 1 when PLL circuit 1 is on. • Equal to the frequency of CKIO pin when PLL circuit 1 is off. • Do not set the internal clock frequency lower than the CKIO pin frequency. 5. The frequency of the peripheral clock (Pϕ) becomes: • The product of the frequency of the CKIO pin, the frequency multiplication ratio of PLL circuit 1, and the division ratio of divider 2 when the clock operating mode is 0 to 2 or 7. • The peripheral clock frequency should not be set higher than the maximum frequency specified in the AC Characteristics, higher than the frequency of the CKIO pin, higher than 40 MHz, or lower than 1/8 the internal clock (Iϕ). 6. The output frequency of PLL circuit 1 is the product of the CKIO frequency and the multiplication ratio of PLL circuit 1. This frequency should be equal to or lower than the maximum frequency specified in the AC Characteristics. 7. × 1, × 2, × 3, × 4, or × 6 can be used as the multiplication ratio of PLL circuit 1. × 1, × 1/2, × 1/3, and × 1/4 can be selected as the division ratio of divider 1. × 1, × 1/2, × 1/3, × 1/4, and × 1/6 can be selected as the division ratio of divider 2. Set the rate in the frequency control register. The on/off state of PLL circuit 2 is determined by the mode.	Cautions: 1. The frequency ranges of the input clock and crystal oscillator should be set within the specified frequency range based on the clock rate in table 10.4, and section 32.3, AC Characteristics. 2. The input to divider 1 becomes the output of PLL circuit 1 when PLL circuit 1 is on. 3. The input of divider 2 becomes the output of: • PLL circuit 1 4. The frequency of the internal clock (Iϕ) becomes: • The product of the frequency of the CKIO pin, the frequency multiplication ratio of PLL circuit 1, and the division ratio of divider 1 when PLL circuit 1 is on. • Do not set the internal clock frequency lower than the CKIO pin frequency. • Depending on the product, the clock ratio should be set to produce a frequency within one of the ranges indicated below. 100 MHz products: 24 MHz to 100 MHz 160 MHz products: 24 MHz to 160 MHz 5. Bus clock (Bϕ) frequency: • Depending on the product, the clock ratio should be set to produce a frequency within one of the ranges indicated below. 100 MHz products: 24 MHz to 50 MHz 160 MHz products: 24 MHz to 66.64 MHz 6. The frequency of the peripheral clock (Pϕ) becomes: • The product of the frequency of the CKIO pin, the frequency multiplication ratio of PLL circuit 1, and the division ratio of divider 2. • For all products, the peripheral clock frequency (Pϕ) should be set within the frequency range 6 MHz to 33.34 MHz and no higher than the frequency of the CKIO pin. • The peripheral clock frequency (Pϕ) should be set to 13 MHz or higher if the USB function module is used. 7. The output frequency of PLL circuit 1 is the product of the CKIO frequency and the multiplication ratio of PLL circuit 1. 8. ×1, ×2, ×3, ×4, or ×6 can be used as the multiplication ratio of PLL circuit 1. ×1, ×1/2, ×1/3, and ×1/4 can be selected as the division ratio of divider 1. ×1, ×1/2, ×1/3, ×1/4, and ×1/6 can be selected as the division ratio of divider 2. Set the rate in the frequency control register. The on/off state of PLL circuit 2 is determined by the mode.

<table>
<tr><th>Page</th><th>Previous Version</th><th>Revised Version</th></tr>
<tr><td>279</td><td>11.1.1 EXCPG

The extend clock pulse generator (EXCPG) generates a divided clock from the CPU clock (Iϕ), the bus clock (Bϕ), or the external clock (UCLK).</td><td>

The extend clock pulse generator (EXCPG) generates a divided clock from the internal clock (Iϕ), the bus clock (Bϕ), or the external clock (UCLK).</td></tr>
<tr><td></td><td>Figure 11.1 Block Diagram of EXCPG

</td><td>
</td></tr>
<tr><td>281</td><td>11.3.1 EXCPG Control Register (EXCPGCR)

Bits 5 to 3—Clock Select (USBCKSEL2 to USBCKSEL0):

Bits 5 to 3 | Function (Clock Selection)
110 | External clock</td><td>

Bits 5 to 3 | Function (Clock Selection)
110 | External clock (UCLK)</td></tr>
<tr><td></td><td>Bits 2 to 0 Function (Dividing Ratio Selection):

Bits 2 to 0 | Function (Dividing Ratio Selection)
1** | Iϕ, CKIO, UCLK halted
Note: To reduce power consumption, set USBDIVSEL2 to 1 and halts Iø, CKIO, or UCLK input.</td><td>

Bits 2 to 0 | Function (Dividing Ratio Selection)
1** | Internal clock (Iϕ), bus clock (Bϕ), external clock (UCLK) halted
Note: To reduce power consumption, set USBDIVSEL2 to 1 and halts internal clock (Iϕ), bus clock (Bϕ), or external clock (UCLK) input.</td></tr>
<tr><td>281</td><td></td><td>11.4 Usage Notes

Newly added</td></tr>
</table>

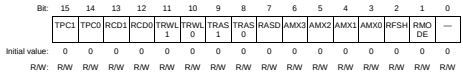
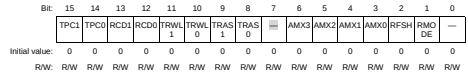
<table>
<tr><th>Page</th><th>Previous Version</th><th>Revised Version</th></tr>
<tr><td>303</td><td>12.2.5 Individual Memory Control Register (MCR)

</td><td>
</td></tr>
<tr><td></td><td>Bits 15 and 14—RAS Precharge Time (TPC1, TPC0):
… However, the number of cycles inserted immediately after the precharge all banks (PALL) command is issued when performing auto-refresh or the precharge (PRE) command is issued in bank-active mode is one fewer than the number of cycles during normal operation. Do not set TPC1 to 0 and TPC0 to 0 when in bank-active mode.</td><td>… However, the number of cycles inserted immediately after the precharge all banks (PALL) command is issued when performing auto-refresh is one fewer than the number of cycles during normal operation.</td></tr>
<tr><td></td><td>Note: * Immediately after the precharge all banks (PALL) command is issued when performing auto-refresh or the precharge (PRE) command is issued in bank-active mode.</td><td>Note: * Immediately after the precharge all banks (PALL) command is issued when performing auto-refresh.</td></tr>
<tr><td>304</td><td>Bit 7—SDRAM Bank Active (RASD): Specifies whether SDRAM is put into bank-active mode or auto-precharge mode. The auto-precharge mode should be used if both area 2 and area 3 are set in SDRAM space and the bus width is 16 bits.
<table>
<tr><th>Bit 7: RASD</th><th>Description</th><th></th></tr>
<tr><td>0</td><td>Auto-precharge mode</td><td>(Initial value)</td></tr>
<tr><td>1</td><td>Bank-active mode</td><td></td></tr>
</table></td><td>Bit 7—Reserved:
This bit is always read as 0. The write value should always be 0.

Table deleted</td></tr>
<tr><td>309</td><td>12.2.6 PCMCIA Control Register (PCR)
Bits 8, 1, and 0—Area6 $\overline{OE}$/$\overline{WE}$ Negate Address Delay (A6TEH2, A6TEH1, and A6TEH0):
<table>
<tr><th>Bit 8: A6TEH2</th><th>Bit 1: A6TEH1</th><th>Bit 0: A6TEH0</th><th>Description</th></tr>
<tr><td>1</td><td>0</td><td>0</td><td>4.5-cycle delay</td></tr>
<tr><td></td><td></td><td>1</td><td>Reserved</td></tr>
<tr><td></td><td>1</td><td>0</td><td>Reserved</td></tr>
<tr><td></td><td></td><td>1</td><td>Reserved</td></tr>
</table></td><td>
<table>
<tr><th>Bit 8: A6TEH2</th><th>Bit 1: A6TEH1</th><th>Bit 0: A6TEH0</th><th>Description</th></tr>
<tr><td>1</td><td>0</td><td>0</td><td>4.5-cycle delay</td></tr>
<tr><td></td><td></td><td>1</td><td>5.5-cycle delay</td></tr>
<tr><td></td><td>1</td><td>0</td><td>6.5-cycle delay</td></tr>
<tr><td></td><td></td><td>1</td><td>7.5-cycle delay</td></tr>
</table></td></tr>
</table>

Page	Previous Version	Revised Version
385	14.2.2 DMA Destination Address Registers 0 to 3 (DAR0 to DAR3) To transfer data in 16 bits or in 32 bits, specify the address on the 16-bit or 32-bit boundary. If any other address is specified, correct operation is not guaranteed.	To transfer data in 16 bits or in 32 bits, specify the address on the 16-bit or 32-bit boundary. When transferring data in 16-byte units, always set a value at a 16-byte boundary (16n address) as the destination address. If any other address is specified, correct operation is not guaranteed.
433	14.4.2 Register Descriptions Compare-Match Timer Control/Status Register 0 (CMCSR0) The compare-match timer control/status register 0 (CMCSR0) is a 16-bit register that indicates a compare-match occurrence, sets enable/disable of interrupts, and sets the incrementation clock. …	The compare-match timer control/status register 0 (CMCSR0) is a 16-bit register that indicates a compare-match occurrence and sets the incrementation clock. …
434	Bits 1 and 0—clock Select 1, 0 (CKS1, CKS0): These bits select the clock input to CMCNT from four internal clocks which are divided from the system clock (Pϕ). When the STR bit in CMSTR is set to 1, …	These bits select the clock input to CMCNT from four clocks which are divided from the peripheral clock (Pϕ). When the STR0 bit in CMSTR is set to 1, …
	Compare-Match Counter 0 (CMCNT0) When the internal clock is selected with the CKS1 and CKS0 bits in CMCSR0 and the STR bit in CMSTR is set to 1, …	When the clock is selected with the CKS1 and CKS0 bits in CMCSR0 and the STR0 bit in CMSTR is set to 1, …
435	14.4.3 Operation Period Count Operation When the internal clock is selected with the CKS1, CKS0 bits in CMCSR0 and the STR bit of the CMSTR is set to 1,	When the clock is selected with the CKS1 and CKS0 bits in CMCSR0 and the STR0 bit in CMSTR is set to 1, …

Page	Previous Version	Revised Version
436	CMCNT0 Count Timing One of four clocks (Pϕ/4, Pϕ/8, Pϕ/16, Pϕ/64) which are divided from the clock (Pϕ) can be selected with the CKS1 and CKS0 bits in CMCSR0. …	One of four peripheral clocks (Pϕ/4, Pϕ/8, Pϕ/16, Pϕ/64) which are divided from the clock (Pϕ) can be selected with the CKS1 and CKS0 bits in CMCSR0. …
	Figure 14.28 Count Timing CK Internal clock	Peripheral clock (Pϕ) CMT clock
437	Figure 14.29 Timing of CMF Setting CK	Peripheral clock (Pϕ)
	Figure 14.30 Timing of CMF Clear by the CPU CK	Peripheral clock (Pϕ)
442	14.6 Usage Notes	*Item 14, 15, added*
443	15.1.1 Features • Selection of six counter input clocks for each channel: On-chip RTC output clock (16 kHz), Pϕ/4, Pϕ/16, Pϕ/64, and Pϕ/256 Note: Pϕ is the internal clock for peripheral modules and can be selected as 1/4, 1/2, or the same frequency as that of the CPU operating clock ϕ.) See section 10, On-Chip Oscillation Circuits, for more information on the clock pulse generator.	• Selection of six counter input clocks for each channel: On-chip RTC output clock (16 kHz), Pϕ/4, Pϕ/16, Pϕ/64, and Pϕ/256 *Note deleted*
	• The maximum 2 MHz operating frequency for the 32-bit counter in each channel: Operate the SH7727 so that the clock input to each channel timer counter does not exceed the maximum operating frequency, by dividing the external clock and internal clock with the prescaler)	Operate the SH7727 so that the clock input to each channel timer counter does not exceed the maximum operating frequency, by dividing the external clock and peripheral clock (Pϕ) with the prescaler.

<table>
<tr><th>Page</th><th>Previous Version</th><th>Revised Version</th></tr>
<tr><td>469</td><td>16.2.15 RTC Control Register 1 (RCR1)

RCR1 is initialized to H'00 by a power-on reset. By a manual reset, bits except the CF flag are all initialized to 0, but the CF flag is undefined. When using the CF flag, it must be initialized beforehand. This register is not initialized in standby mode.

<table>
<tr><td>Bit:</td><td>7</td><td>6</td></tr>
<tr><td></td><td>CF</td><td>—</td></tr>
<tr><td>Initial value:</td><td>0</td><td>0</td></tr>
<tr><td>R/W:</td><td>R/W</td><td>R</td></tr>
</table>

Bit 7—Carry Flag (CF):
<table>
<tr><th>Bit 7: CF</th><th>Description</th></tr>
<tr><td>0</td><td>No carry in R64CNT or RSECCNT.
Clearing condition: When 0 is written to CF (Initial value)</td></tr>
</table></td><td>RCR1 is an 8-bit read/write register. The CIE, AIE, and AF bits are initialized by a power-on reset or manual reset. However, the value of the CF flag is undefined after a power-on reset or manual reset. It must therefore be initialized without fail before use. This register is not initialized in standby mode.

<table>
<tr><td>Bit:</td><td>7</td><td>6</td></tr>
<tr><td></td><td>CF</td><td>—</td></tr>
<tr><td>Initial value:</td><td>—</td><td>0</td></tr>
<tr><td>R/W:</td><td>R/W</td><td>R</td></tr>
</table>

<table>
<tr><th>Bit 7: CF</th><th>Description</th></tr>
<tr><td>0</td><td>No carry in R64CNT or RSECCNT.
Clearing condition: When 0 is written to CF</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Page</th><th>Previous Version</th><th>Revised Version</th></tr>
<tr><td>472,
473</td><td>

16.3.2 Setting the Time

Figure 16.2 shows how to set the time after stopping the clock. This procedure is available to set the entire calendar and clock function. This procedure can be programmed easily.

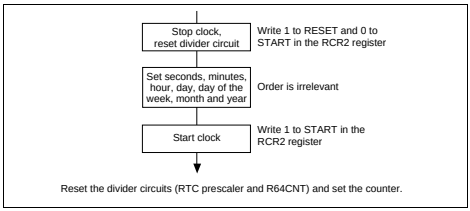


Figure 16.2 Setting the Time

</td><td>

Figures 16.2 (a) and 16.2 (b) show how to set the time after stopping the clock. This procedure can be used to set the entire calendar and clock function. It can be programmed easily.

Usage Notes

1. Initialization Timing for 64 Hz Counter (R64CNT)
   If it is necessary, after initializing the counter by means of the RESET bit in the RTC's RCR2 register, to confirm that the change has taken effect by reading the R64CNT value, wait at least 107 µs after setting the RESET bit to 1 before reading the R64CNT counter. Note that the divider circuit (RTC prescaler) is also initialized when the RESET bit is set to 1.

2. Incrementing RSECCNT by Initializing R64CNT
   Either method (a) or method (b) below may be used.
   (a) After setting the RESET bit to 1 and confirming that R64CNT has been initialized, set the START bit to 1. This process is shown in figure 16.2 (a).
   (b) Set the START bit to 1 and the RESET bit to 1 at the same time. This process is shown in figure 16.2 (b). Note that the processing indicated by the asterisk (*) in figure 16.2 (b) may be omitted if nothing is written to the RCR2 register during an interval of approximately 107 µs after the START bit is set to 1.

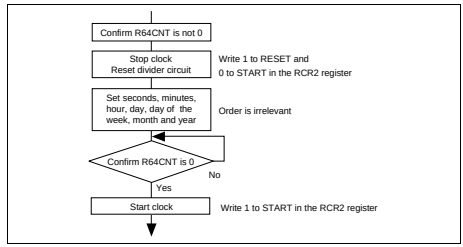


Figure 16.2(a) Setting the Time

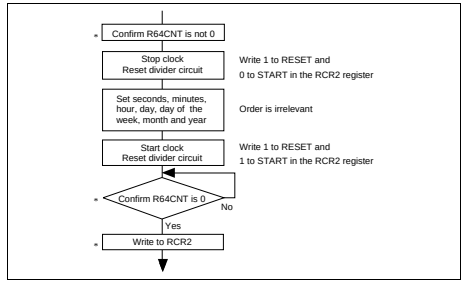


Figure 16.2(b) Setting the Time

</td></tr>
<tr><td>481</td><td>

17.1.2 Block Diagram

SCI pin I/O and data control is performed by bits 11 to 8 of SCPCR and bits 5 and 4 of SCPDR. …

</td><td>

SCI pin I/O and data control is performed by bits 3 to 0 of SCPCR and bits 1 and 0 of SCPDR. …

</td></tr>
<tr><td>484</td><td>

Table 17.2 Registers

H'A888

</td><td>

H'8008

</td></tr>
<tr><td>497</td><td>

17.2.8 Port SC Control Register (SCPCR)/Port SC Date Register (SCPDR)

SCPCR

Bit:	15	14	13	12	11	10	9	8	7	6
	SCP7 MD1	SCP7 MD0	SCP6 MD1	SCP6 MD0	SCP5 MD1	SCP5 MD0	SCP4 MD1	SCP4 MD0	SCP3 MD1	SCP3 MD0
Initial value:	1	0	1	0	1	0	0	0	1	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

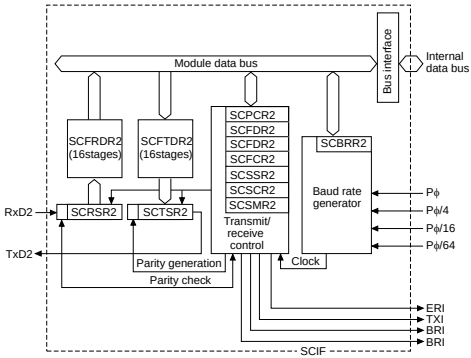
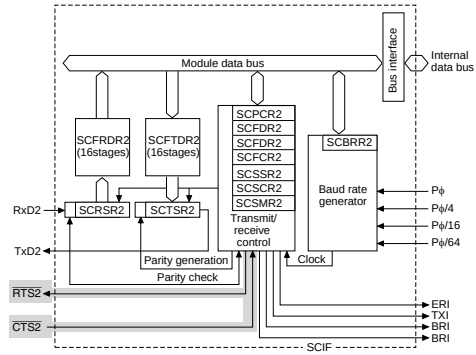
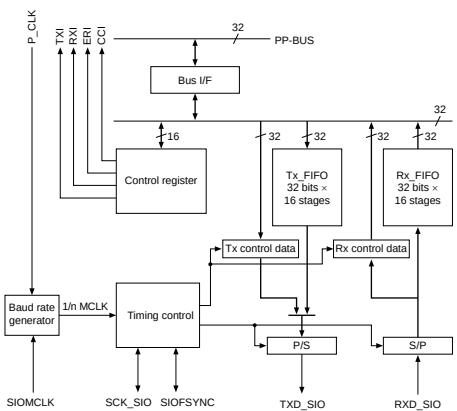
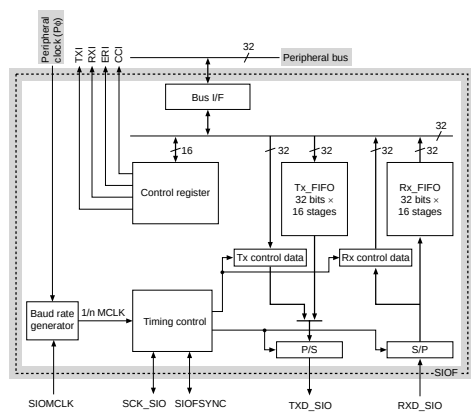
</td><td>

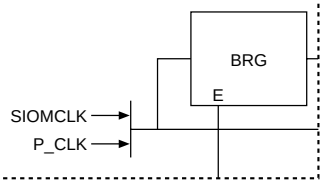
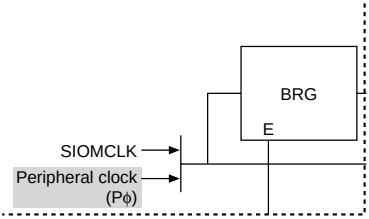
SCPCR

Bit:	15	14	13	12	11	10	9	8	7	6
	SCP7 MD1	SCP7 MD0	SCP6 MD1	SCP6 MD0	SCP5 MD1	SCP5 MD0	SCP4 MD1	SCP4 MD0	SCP3 MD1	SCP3 MD0
Initial value:	1	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

</td></tr>
</table>

RENESAS

Page	Previous Version	Revised Version
563	19.1.1 Features • On-chip modem control functions ($\overline{\text{RTS}}$ and $\overline{\text{CTS}}$)	• On-chip modem control functions ($\overline{\text{RTS2}}$ and $\overline{\text{CTS2}}$)
564	Figure 19.1 SCIF Block Diagram 	
565	19.1.2 Block Diagram Figures 19.2 and 19.3 show SCIF I/O ports. Bits 11 to 8 of SCPCR and bits 5 and 4 of SCPDR control an input/output and data of the SCIF pins. …	Figure 19.2 and 19.3 show SCIF I/O ports. Bits 15, 14, 9, 8 of SCPCR and bits 7 and 4 of SCPDR control an input/output and data of the SCIF pins. …
603	20.1.1 Features • Serial clock — External clock or internal clock (P_CLK) are able to be used as clock source.	— External clock or peripheral clock (Pϕ) are able to be used as clock source.
604	20.1.2 Block Diagram Figure 20.1 SIOF Block Diagram 	

<table>
<tr><th>Page</th><th>Previous Version</th><th>Revised Version</th></tr>
<tr><td>608</td><td>20.2.2 Clock Select Register (SISCR)
• Bit 15—Master Clock Source Choice (MSSEL):
Use PCLK as master clock</td><td>Use Peripheral clock (Pϕ) as master clock</td></tr>
<tr><td>628</td><td>Figure 20.2 Serial Clock Supply System

</td><td>
</td></tr>
<tr><td>635</td><td>20.3.5 Control Data Interface
(1) Control by Slot Positions (Master Mode 1)
Note: When using this method, PCLK should be used as the master clock (Master Clock Select (MSSEL) = 1).</td><td>Note: When using this method, Peripheral clock should be used as master clock (Master Clock Select (MSSEL) = 1).</td></tr>
<tr><td>651</td><td>20.4 Usage Notes</td><td>Item 7, 8, added</td></tr>
<tr><td>679</td><td>22.1.2 Pin Configuration
Table 22.1 Pin Configuration (Digital Transceiver Signal)
<table>
<tr><th>Name</th><th>Symbol</th><th>I/O</th><th>Description</th></tr>
<tr><td>TXDPLS pin</td><td>USB1d_TXDPLS</td><td>Output</td><td>D+ transmit output pin</td></tr>
<tr><td>TXDMNS pin</td><td>USB1d_TXDMNS</td><td>Output</td><td>D– transmit output pin</td></tr>
<tr><td>TXENL pin</td><td>USB1d_TXENL</td><td>Output</td><td>Driver output enable pin</td></tr>
</table></td><td>
<table>
<tr><th>Name</th><th>Symbol</th><th>I/O</th><th>Description</th></tr>
<tr><td>TXDPLS pin</td><td>USB1d_TXDPLS</td><td>Output</td><td>D+ transmit output pin</td></tr>
<tr><td>TXENL pin</td><td>USB1d_TXENL</td><td>Output</td><td>Driver output enable pin</td></tr>
</table></td></tr>
<tr><td>709</td><td>23.6.3 Control Transfer
Figure 23.8 Status Stage Operation (Control-In)</td><td>Replaced</td></tr>
<tr><td>720</td><td></td><td>23.9 Usage Notes
Newly added</td></tr>
<tr><td>758</td><td></td><td>24.7 Notes on Using USB Host with Versions Previous to the SH7727C
Newly added</td></tr>
</table>

Page	Previous Version	Revised Version
759	25.1.1 Features • Supports 1/2/4/8/15/16-bpp (bit per pixel) color modes • Supports 1/2/4-bpp grayscale modes	25.1.1 Features • Supports 4/8/15/16-bpp (bit per pixel) color modes • Supports 1/2/4/6-bpp grayscale modes
760	Figure 25.1 Block Diagram CKIO P clock	Figure 25.1 Block Diagram Bus clock (Bϕ) Peripheral clock (Pϕ)
762	Table 25.2 Register Configuration H'F606	Table 25.2 Register Configuration H'F60F
763	25.2.1 LCDC Input Clock Register (LDICKR) This LCDC can select CKIO (bus clock), the P clock, or the external clock as its operation clock source. …	25.2.1 LCDC Input Clock Register (LDICKR) This LCDC can select the bus clock (Bϕ), the peripheral clock (Pϕ), or the external clock as its operation clock source. …
	Bits 13, and 12—Input Clock Select (ICKSEL1 and ICKSEL0): CKIO is selected P clock is selected	Bits 13, and 12—Input Clock Select (ICKSEL1 and ICKSEL0): Bus clock (Bϕ) is selected Peripheral clock (Pϕ) is selected
766	25.2.2 LCDC Module Type Register (LDMTR) Bits 5 to 0—Module Interface Type Select (MIFTYP5 to MIFTYP0): If an STN or DSTN panel is selected, display control is performed using a 24-bit space-modulation FRC consisting of the 8-bit R, G, and B included in the LCDC, …	25.2.2 LCDC Module Type Register (LDMTR) Bits 5 to 0—Module Interface Type Select (MIFTYP5 to MIFTYP0): If an STN or DSTN panel is selected, display control is performed using a 24-bit space-modulation FRC (Frame Rate Controller) consisting of the 8-bit R, G, and B included in the LCDC, …
783	25.2.17 LCDC Power Management Mode Register (LDPMMR) Bit 4—DON Pin Enable (DONE): **Bit 4 DONE** / **Description** 0 Disabled: DON pin is masked and fixed low (Initial value) 1 Enabled: DON pin output is asserted and negated according to the power-on or power-off sequence	25.2.17 LCDC Power Management Mode Register (LDPMMR) Bit 4—DON Pin Enable (DONE): **Bit 4 DONE** / **Description** 0 Disabled: DON pin is masked and fixed low 1 Enabled: DON pin output is asserted and negated according to the power-on or power-off sequence (Initial value)
784	25.2.18 LCDC Power-Supply Sequence Period Register (LDPSPR) Bit: 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 ONA3 ONA2 ONA1 ONA0 ONB3 ONB2 ONB1 ONB0 OFFE3 OFFE2 OFFE1 OFFE0 OFFF3 OFFF2 OFFF1 OFFF0 Initial value: 0 0 0 0 0 0 0 0 0 0 1 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	25.2.18 LCDC Power-Supply Sequence Period Register (LDPSPR) Bit: 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 ONA3 ONA2 ONA1 ONA0 ONB3 ONB2 ONB1 ONB0 OFFE3 OFFE2 OFFE1 OFFE0 OFFF3 OFFF2 OFFF1 OFFF0 Initial value: 1 1 1 1 0 1 1 0 0 0 0 0 1 1 1 1 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

<table>
<tr><th>Page</th><th>Previous Version</th><th>Revised Version</th></tr>
<tr><td>786</td><td>25.3.1 LCD Module Sizes which can be Displayed in this LCDC

The overhead coefficient is 1.375 if the SDRAM in CL2 uses a 32-bit bus and 1.188 if it uses a 16-bit bus.</td><td>The overhead coefficient depends on the bus used by the SDRAM in CL2, as indicated below.

If the hardware rotation function is not used (ROT = 0), the overhead coefficient is 1.375 if a 32-bit bus is used and 1.188 if a 16-bit bus is used.

If the hardware rotation function is used (ROT = 1), the overhead coefficient is determined by the access unit select (AU) setting and the bus width, as follows.

Table added

<table>
<tr><th>Access Unit Select (AU) Setting</th><th>32-Bit Bus</th><th>16-Bit Bus</th></tr>
<tr><td>4-burst operation</td><td>2.500</td><td>1.750</td></tr>
<tr><td>8-burst operation</td><td>1.750</td><td>1.375</td></tr>
<tr><td>16-burst operation</td><td>1.375</td><td>1.188</td></tr>
<tr><td>32-burst operation</td><td>1.188</td><td>1.094</td></tr>
</table>
</td></tr>
<tr><td>787 to 793</td><td></td><td>25.3.2 Limits on the Resolution of Rotated Displays, Burst Length, and Connected Memory (SDRAM)

Replaced</td></tr>
<tr><td>816</td><td></td><td>25.5 Usage Notes

Newly added</td></tr>
<tr><td>820, 821</td><td>Table 26.1 List of Multiplexed Pins

<table>
<tr><th>Port</th><th>Port Function (Related Module)</th><th>Other Function 1 (Related Module)</th><th>Other Function 2 (Related Module)</th></tr>
<tr><td>K</td><td>PTK2 in/out (port)</td><td>CS4 out (BSC)</td><td></td></tr>
<tr><td>K</td><td>PTK1 in/out (port)</td><td>AFE_RLYCNT out (AFE)</td><td>USB1d_DMNS0 in (USB)*2</td></tr>
<tr><td>K</td><td>PTK0 in/out (port)</td><td>AFE_HC1 out (AFE)</td><td>USB1d_DPLS0 in (USB)*2</td></tr>
<tr><td>L</td><td>PTL7 in (port)</td><td>AN7 in (ADC)/DA0 out (DAC)</td><td></td></tr>
<tr><td>L</td><td>PTL2 in (port)</td><td>AN2 in (ADC)</td><td></td></tr>
<tr><td>M</td><td>PTM7 in (port)/PINT7 in (INTC)</td><td>AFE_FS in (AFE)</td><td>USB1d_RCV0 in (USB)*2</td></tr>
<tr><td>M</td><td>PTM6 in (port)/PINT6 in (INTC)</td><td>AFE_RXIN in (AFE)</td><td>USB1d_SPEED0 out (USB)*2</td></tr>
<tr><td>M</td><td>PTM5 in (port)/PINT5 in (INTC)</td><td>AFE_TXOUT out (AFE)</td><td>USB1d_TXSE0 out (USB)*2</td></tr>
<tr><td>M</td><td>PTM4 in (port)/PINT4 in (INTC)</td><td>AFE_RDET in (AFE)</td><td>USB1d_TXMNS0 out (USB)*2</td></tr>
<tr><td>M</td><td>PTM3 in (port)/ PINT10 in (INTC)</td><td>LCD15 out (LCDC)</td><td></td></tr>
</table>

<table>
<tr><th>Port</th><th>Port Function (Related Module)</th><th>Other Function 1 (Related Module)</th><th>Other Function 2 (Related Module)</th></tr>
<tr><td>SCPT</td><td>SCPT4 in (port)*1</td><td>RxD2 in (UART ch 3)</td><td></td></tr>
<tr><td></td><td>SCPT4 out (port)*1</td><td>TxD2 out (UART ch 3)</td><td></td></tr>
<tr><td>SCPT</td><td>SCPT1 in/out (port)</td><td>SCK0 in/out (UART ch 1)</td><td></td></tr>
<tr><td>SCPT</td><td>SCPT0 in (port)*1</td><td>RxD0 in (UART ch 1)</td><td></td></tr>
<tr><td></td><td>SCPT0 out (port)*1</td><td>TxD0 out (UART ch 1)</td><td></td></tr>
</table>

<table>
<tr><th>Port</th><th>Port Function (Related Module)</th><th>Other Function 1 (Related Module)</th><th>Other Function 2 (Related Module)</th></tr>
<tr><td>L</td><td>PTL2 in (port)</td><td>AN2 in (ADC)</td><td></td></tr>
<tr><td>K</td><td>PTK2 in/out (port)</td><td>CS4 out (BSC)</td><td></td></tr>
<tr><td>K</td><td>PTK1 in/out (port)</td><td>AFE_RLYCNT out (AFE)</td><td>USB1d_DMNS in (USB)*2</td></tr>
<tr><td>K</td><td>PTK0 in/out (port)</td><td>AFE_HC1 out (AFE)</td><td>USB1d_DPLS in (USB)*2</td></tr>
<tr><td>L</td><td>PTL7 in (port)</td><td>AN7 in (ADC)/DA0 out (DAC)</td><td></td></tr>
<tr><td>L</td><td>PTL2 in (port)</td><td>AN2 in (ADC)</td><td></td></tr>
<tr><td>M</td><td>PTM7 in (port)/PINT7 in (INTC)</td><td>AFE_FS in (AFE)</td><td>USB1d_RCV in (USB)*2</td></tr>
<tr><td>M</td><td>PTM6 in (port)/PINT6 in (INTC)</td><td>AFE_RXIN in (AFE)</td><td>USB1d_SPEED out (USB)*2</td></tr>
<tr><td>M</td><td>PTM5 in (port)/PINT5 in (INTC)</td><td>AFE_TXOUT out (AFE)</td><td>USB1d_TXSE0 out (USB)*2</td></tr>
<tr><td>M</td><td>PTM4 in (port)/PINT4 in (INTC)</td><td>AFE_RDET in (AFE)</td><td></td></tr>
<tr><td>M</td><td>PTM3 in (port)/ PINT10 in (INTC)</td><td>LCD15 out (LCDC)</td><td></td></tr>
</table>

<table>
<tr><th>Port</th><th>Port Function (Related Module)</th><th>Other Function 1 (Related Module)</th><th>Other Function 2 (Related Module)</th></tr>
<tr><td>SCPT</td><td>SCPT4 in (port)*1</td><td>RxD2 in (SCIF)</td><td></td></tr>
<tr><td></td><td>SCPT4 out (port)*1</td><td>TxD2 out (SCIF)</td><td></td></tr>
<tr><td>SCPT</td><td>SCPT1 in/out (port)</td><td>SCK0 in/out (SCI)</td><td></td></tr>
<tr><td>SCPT</td><td>SCPT0 in (port)*1</td><td>RxD0 in (SCI)</td><td></td></tr>
<tr><td></td><td>SCPT0 out (port)*1</td><td>TxD0 out (SCI)</td><td></td></tr>
</table>
</td></tr>
</table>

RENESAS

Page	Previous Version	Revised Version
826	26.3.4 Port D Control Register (PDCR) **Bit (2n + 1) / PDnMD1** — **Bit 2n / PDnMD0** — **Pin Function** 0 — 0 — Other function (see table 26.1) 0 — 1 — Port output (n = value other than 4 or 6), reserved (n = 4 or 6) 1 — 0 — Port input (Pullup MOS: on) 1 — 1 — Port input (Pullup MOS: off)	 **Bit (2n + 1) / PDnMD1** — **Bit 2n / PDnMD0** — **Pin Function** 0 — 0 — Other function (see table 26.1) 0 — 1 — Port output (n = value other than 4 or 6), reserved (n = 4 or 6) 1 — 0 — Port input (Pullup MOS: on) (Initial value) 1 — 1 — Port input (Pullup MOS: off)
838	26.3.13 SC Port Control Register (SCPCR) Bits 5, 4—SCP2 Mode 1, 0 (SCP2MD1, SCP2MD0): **Bit 5 / SCP2MD1** — **Bit 4 / SCP2MD0** — **Pin Function** 0 — 0 — Transmit data output 1 (TxD1) Receive data input 1 (RxD1) (Initial value) 0 — 1 — General output (SCPT[2] output pin) Receive data input 1 (RxD1) 1 — 0 — SCPT[2] input pin pullup (input pin) Transmit data output 1 (TxD1) 1 — 1 — General input (SCPT[2] input pin) Transmit data output 1 (TxD1) Note: There is no combination of simultaneous I/O of SCPT[2] because one bit (SCP2DT) is accessed using two pins of TxD1 and RxD1. When the port input is set (bit SCPnMD1 is set to 1) and when the TE bit in SCSCR is set to 1, the TxD1 pin is in the output state. When the TE bit is cleared to 0, the TxD1 pin is in the high-impedance state.	 **Bit 5 / SCP2MD1** — **Bit 4 / SCP2MD0** — **Pin Function** 0 — 0 — Transmit data output 1 (TxD_SiO) Receive data input 1 (RxD_SiO) (Initial value) 0 — 1 — General output (SCPT[2] output pin) Receive data input 1 (RxD_SiO) 1 — 0 — SCPT[2] input pin pullup (input pin) Transmit data output 1 (TxD_SiO) 1 — 1 — General input (SCPT[2] input pin) Transmit data output 1 (TxD_SiO) Note: There is no combination of simultaneous I/O of SCPT[2] because one bit (SCP2DT) is accessed using two pins of TxD_SiO and RxD_SiO. When the port input is set (bit SCPnMD1 is set to 1) and when the TE bit in SCSCR is set to 1, the TxD_SiO pin is in the output state. When the TE bit is cleared to 0, the TxD_SiO pin is in the high-impedance state.
909	30.3.3 Usage Notes Setting procedure when Using PC Card Controller: 1. Set bit 0 (A6PCM) in bus control register 1 (BCR1) in the bus state controller to 1. 2. Set bit 4 (P0USE) in the area 6 general control register in the PC card controller to 1. 3. Set the pin function controller to custom PC card pin functions (“other functions”).	 1. Drive pin $\overline{\text{ASEMD0}}$ high. 2. Set bit 0 (A6PCM) in bus control register 1 (BCR1) in the bus state controller to 1. 3. Set bit 4 (P0USE) in the area 6 general control register in the PC card controller to 1. 4. Set the pin function controller to custom PC card pin functions (“other functions”).

<table>
<tr><th>Page</th><th>Previous Version</th><th>Revised Version</th></tr>
<tr><td>918, 919</td><td>Table 31.3 Correspondence between SH7727 Pins and Boundary-Scan Register
<table><tr><th>Bit</th><th>Pin Name</th><th>I/O</th></tr><tr><td>170</td><td>PTM5/PINT5/AFE_TXOUT/ USB1d_TXSE0</td><td>Control</td></tr><tr><td>169</td><td>PTM4/PINT4/$\overline{\text{AFE_RDET}}$/ USB1d_TXDMNS</td><td>IN</td></tr><tr><td>168</td><td>Reserved/USB1d_SUSPEND0</td><td>IN</td></tr><tr><td>137</td><td>MD0</td><td>IN</td></tr><tr><td>136</td><td>PTM4/PINT4/$\overline{\text{AFE_RDET}}$/ USB1d_TXDMNS</td><td>OUT</td></tr><tr><td>135</td><td>Reserved/USB1d_SUSPEND0</td><td>OUT</td></tr><tr><td>110</td><td>PTF0/$\overline{\text{PCC0VS2}}$/Reserved</td><td>OUT</td></tr><tr><td>109</td><td>PTM4/PINT4/$\overline{\text{AFE_RDET}}$/ USB1d_TXDMNS</td><td>Control</td></tr><tr><td>108</td><td>Reserved/USB1d_SUSPEND0</td><td>Control</td></tr></table></td><td><table><tr><th>Bit</th><th>Pin Name</th><th>I/O</th></tr><tr><td>170</td><td>PTM5/PINT5/AFE_TXOUT/ USB1d_TXSE0</td><td>Control</td></tr><tr><td>169</td><td>PTM4/PINT4/$\overline{\text{AFE_RDET}}$</td><td>IN</td></tr><tr><td>168</td><td>Reserved/USB1d_SUSPEND0</td><td>IN</td></tr><tr><td>137</td><td>MD0</td><td>IN</td></tr><tr><td>136</td><td>PTM4/PINT4/$\overline{\text{AFE_RDET}}$</td><td>OUT</td></tr><tr><td>135</td><td>Reserved/USB1d_SUSPEND0</td><td>OUT</td></tr><tr><td>110</td><td>PTF0/PCC0VS2/Reserved</td><td>OUT</td></tr><tr><td>109</td><td>PTM4/PINT4/$\overline{\text{AFE_RDET}}$</td><td>Control</td></tr><tr><td>108</td><td>Reserved/USB1d_SUSPEND0</td><td>Control</td></tr></table></td></tr>
<tr><td>925</td><td>32.1 Absolute Maximum Ratings
Caution:</td><td>Item 3 added</td></tr>
<tr><td>927, 928</td><td>Table 32.2 DC Characteristics (1)
<table><tr><th>Item</th><th>Symbol</th><th>Min</th><th>Typ</th><th>Max</th><th>Unit</th><th>Measurement Conditions</th></tr><tr><td>Power supply voltage</td><td>$V_{CC}Q$</td><td>3.0</td><td>—</td><td>3.6</td><td>V</td><td>HD6417727F160, HD6417727BP160V</td></tr><tr><td></td><td></td><td>2.6</td><td>—</td><td>3.6</td><td></td><td>HD6417727F100, HD6417727BP100V
See note *4 for applied voltage when mounted.</td></tr><tr><td></td><td>V_{CC}, V_{CC}-PLL1, V_{CC}-PLL2, V_{CC}-RTC*1</td><td>1.70</td><td>—</td><td>2.05</td><td></td><td>HD6417727F160, HD6417727BP160V
See note *4 for applied voltage when mounted.</td></tr><tr><td></td><td></td><td>1.60</td><td>—</td><td>2.05</td><td></td><td>HD6417727F100, HD6417727BP100V
See note *4 for applied voltage when mounted.</td></tr><tr><td>Analog (A/D, D/A) power-supply current</td><td>During A/D conversion</td><td>AI_{CC}</td><td>—</td><td>0.8</td><td>2</td><td>mA</td><td></td></tr><tr><td></td><td>During A/D and D/A conversion</td><td></td><td>—</td><td>2.4</td><td>6</td><td>mA</td><td></td></tr><tr><td></td><td>Idle</td><td></td><td>—</td><td>0.01</td><td>5.0</td><td>mA</td><td></td></tr></table>
Notes:
*3 Current dissipation values shown are for VIHmin = VccQ – 0.5 V and VILmax = 0.5 V with 5 pF load.
*4 The voltage range that can be applied depends on the operating frequency setting. Be sure check the operating frequency range of the AC characteristics.
*5 There is no stipulation regarding the power supply in standby mode when there is no RTC clock input.</td><td><table><tr><th>Item</th><th>Symbol</th><th>Min</th><th>Typ</th><th>Max</th><th>Unit</th><th>Measurement Conditions</th></tr><tr><td>Power supply voltage</td><td>$V_{CC}Q$</td><td>3.0</td><td>—</td><td>3.6</td><td>V</td><td>160 MHz products</td></tr><tr><td></td><td></td><td>2.6</td><td>—</td><td>3.6</td><td></td><td>100 MHz products</td></tr><tr><td></td><td>V_{CC},</td><td>1.70</td><td>—</td><td>2.05</td><td></td><td>160 MHz products</td></tr><tr><td></td><td>V_{CC}-PLL1, V_{CC}-PLL2, V_{CC}-RTC*1</td><td>1.60</td><td>—</td><td>2.05</td><td></td><td>100 MHz products</td></tr><tr><td>Analog (A/D, D/A) power-supply current</td><td>During A/D conversion</td><td>AI_{CC}</td><td>—</td><td>0.8</td><td>2</td><td>mA</td><td></td></tr><tr><td></td><td>During A/D and D/A conversion</td><td></td><td>—</td><td>2.4</td><td>6</td><td>mA</td><td></td></tr><tr><td></td><td>Idle</td><td></td><td>—</td><td>0.01</td><td>5.0</td><td>mA</td><td>Ta = 25°C</td></tr></table>
3. Current dissipation values are for V_{IH} min = VccQ – 0.5 V and V_{IL} max = 0.5 V with all output pins in the no-load state.
4. There is no stipulation regarding the power supply in standby mode when there is no RTC clock input.</td></tr>
</table>

<table>
<tr><th>Page</th><th>Previous Version</th><th>Revised Version</th></tr>
<tr><td>930</td><td>32.3 AC Characteristics

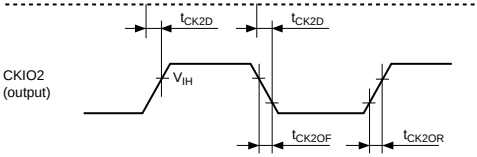
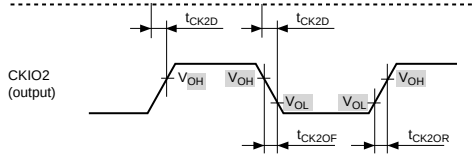
In general, inputting for this LSI should be clock synchronous. Keep the setup and hold times for each input signal unless otherwise specified. Regarding the power supply and frequency specifications of the individual products, refer to tables 32.2 and 32.4. When the measuring condition range in the timing chart is wider than that in table 32.2 or 32.4, the conditions listed in table 32.2 or 32.4 apply.

AC specifications vary depending on the product, so should be checked before the chip is used.</td><td>In general, inputting for this LSI should be clock synchronous. Keep the setup and hold times for each input signal unless otherwise specified. Regarding the power supply and frequency specifications of the individual products, refer to figure 32.2, tables 32.2 and 32.4.</td></tr>
<tr><td>931</td><td></td><td>Figure 32.2 Power Supply Voltage and Operating Frequency

Newly added</td></tr>
<tr><td>932</td><td>Table 32.4 Maximum Operating Frequencies (1)

<table>
<tr><th>Item</th><th>Symbol</th><th>Min</th><th>Max</th><th>Unit</th><th>Power Supply Voltage Conditions</th><th>Products</th></tr>
<tr><td>CPU, cache, TLB (Iφ)</td><td>f</td><td>24</td><td>100</td><td>MHz</td><td>V_{CC} = 1.60 to 2.05 V, $V_{CC}Q$ = 2.6 to 3.6 V</td><td>HD6417727F100, HD6417727BP100V</td></tr>
<tr><td>External bus (Bφ) or CKIO I/O frequency</td><td></td><td>24</td><td>33.4</td><td></td><td>V_{CC} = 1.60 to 2.05 V, $V_{CC}Q$= 2.6 to 3.6 V</td><td></td></tr>
<tr><td></td><td></td><td>24</td><td>50</td><td></td><td>V_{CC} = 1.70 to 2.05 V, $V_{CC}Q$= 3.0 to 3.6 V</td><td></td></tr>
<tr><td>Peripheral modules (Pφ)</td><td></td><td>6</td><td>33.4</td><td></td><td>V_{CC} = 1.60 to 2.05 V, $V_{CC}Q$ = 2.6 to 3.6 V</td><td></td></tr>
</table></td><td>
<table>
<tr><th>Item</th><th>Symbol</th><th>Min</th><th>Max</th><th>Unit</th><th>Power Supply Voltage Conditions</th><th>Reference</th><th>Products</th></tr>
<tr><td>CPU, cache, TLB (Iφ)</td><td>f</td><td>24</td><td>100</td><td>MHz</td><td>V_{CC} = 1.60 to 2.05 V
$V_{CC}Q$ = 2.6 to 3.6 V</td><td>—</td><td>100 MHz products</td></tr>
<tr><td>External bus (Bφ) or CKIO I/O frequency</td><td></td><td>24</td><td>33.34</td><td></td><td>V_{CC} = 1.60 to 2.05 V
$V_{CC}Q$ = 2.6 to 3.6 V</td><td>Table 32.5</td><td></td></tr>
<tr><td></td><td></td><td>24</td><td>50</td><td></td><td>V_{CC} = 1.70 to 2.05 V
$V_{CC}Q$= 3.0 to 3.6 V</td><td>Table 32.6</td><td></td></tr>
<tr><td>Peripheral modules (Pφ)</td><td></td><td>6</td><td>33.34</td><td></td><td>V_{CC} = 1.60 to 2.05 V
$V_{CC}Q$ = 2.6 to 3.6 V</td><td>—</td><td></td></tr>
</table></td></tr>
<tr><td>932</td><td>Table 32.4 Maximum Operating Frequencies (2)

<table>
<tr><th>Item</th><th>Symbol</th><th>Min</th><th>Max</th><th>Unit</th><th>Power Supply Voltage Conditions</th><th>Products</th></tr>
<tr><td>CPU, cache, TLB (Iφ)</td><td>f</td><td>24</td><td>144</td><td>MHz</td><td>V_{CC} = 1.70 to 2.05 V, $V_{CC}Q$ = 3.0 to 3.6 V</td><td>HD6417727F160, HD6417727BP160V</td></tr>
<tr><td></td><td></td><td>24</td><td>160</td><td></td><td>V_{CC} = 1.75 to 2.05 V, $V_{CC}Q$ = 3.0 to 3.6 V</td><td></td></tr>
<tr><td>External bus (Bφ) or CKIO I/O frequency</td><td></td><td>24</td><td>48</td><td></td><td>V_{CC} = 1.70 to 2.05 V, $V_{CC}Q$ = 3.0 to 3.6 V</td><td></td></tr>
<tr><td></td><td></td><td>24</td><td>66.67</td><td></td><td>V_{CC} = 1.75 to 2.05 V, $V_{CC}Q$ = 3.0 to 3.6 V</td><td></td></tr>
<tr><td>Peripheral modules (Pφ)</td><td></td><td>6</td><td>33.4</td><td></td><td>V_{CC} = 1.70 to 2.05 V, $V_{CC}Q$ = 3.0 to 3.6 V</td><td></td></tr>
</table></td><td>
<table>
<tr><th>Item</th><th>Symbol</th><th>Min</th><th>Max</th><th>Unit</th><th>Power Supply Voltage Conditions</th><th>Reference</th><th>Products</th></tr>
<tr><td>CPU, cache, TLB (Iφ)</td><td>f</td><td>24</td><td>144</td><td>MHz</td><td>V_{CC} = 1.70 to 2.05 V
$V_{CC}Q$= 3.0 to 3.6 V</td><td>—</td><td>160 MHz products</td></tr>
<tr><td></td><td></td><td>24</td><td>160</td><td></td><td>V_{CC} = 1.75 to 2.05 V
$V_{CC}Q$= 3.0 to 3.6 V</td><td>—</td><td></td></tr>
<tr><td>External bus (Bφ) or CKIO I/O frequency</td><td></td><td>24</td><td>50</td><td></td><td>V_{CC} = 1.70 to 2.05 V
$V_{CC}Q$= 3.0 to 3.6 V</td><td>Table 32.7</td><td></td></tr>
<tr><td></td><td></td><td>24</td><td>66.67</td><td></td><td>V_{CC} = 1.75 to 2.05 V
$V_{CC}Q$= 3.0 to 3.6 V</td><td>Table 32.8</td><td></td></tr>
<tr><td>Peripheral modules (Pφ)</td><td></td><td>6</td><td>33.34</td><td></td><td>V_{CC} = 1.70 to 2.05 V
$V_{CC}Q$= 3.0 to 3.6 V</td><td>—</td><td></td></tr>
</table></td></tr>
</table>

Page	Previous Version	Revised Version
933	Table 32.5 Clock Timing (1) Conditions: VccQ = 2.6 to 3.6 V, Vcc = 1.6 to 2.05 V, AVcc = 3.3 ± 0.3 V, Ta = –20 to 75°C, external bus maximum operating frequency = 33 MHz Item / Symbol / Min / Max / Unit / Figure EXTAL clock input frequency / t_{EX} / 6 / 33 / MHz / 32.1 EXTAL clock input cycle time / t_{EXcyc} / 30.3 / 167 / ns EXTAL clock input low pulse width / t_{EXL} / 7 / — / ns EXTAL clock input high pulse width / t_{EXH} / 7 / — / ns EXTAL clock input rise time / t_{EXR} / — / 6 / ns EXTAL clock input fall time / t_{EXF} / — / 6 / ns CKIO clock input frequency / t_{CKI} / 24 / 33 / MHz / 32.2 CKIO clock input cycle time / t_{CKIcyc} / 30.3 / 40 / ns CKIO clock input low pulse width / t_{CKIL} / 7 / — / ns CKIO clock input high pulse width / t_{CKIH} / 7 / — / ns CKIO clock input rise time / t_{CKIR} / — / 6 / ns CKIO clock input fall time / t_{CKIF} / — / 6 / ns CKIO clock output frequency / t_{OP} / 24 / 33 / MHz / 32.3 CKIO clock output cycle time / t_{cyc} / 30.3 / 40 / ns	Conditions: VccQ = 2.6 to 3.6 V, Vcc = 1.60 to 2.05 V, AVcc = 3.3 ± 0.3 V, Ta = –20 to 75°C, 100 MHz products Item / Symbol / Min / Max / Unit / Figure EXTAL clock input frequency / t_{EX} / 6 / 33.34 / MHz / 32.3 EXTAL clock input cycle time / t_{EXcyc} / 30 / 166.7 / ns EXTAL clock input low pulse width / t_{EXL} / 9 / — / ns EXTAL clock input high pulse width / t_{EXH} / 9 / — / ns EXTAL clock input rise time / t_{EXR} / — / 6 / ns EXTAL clock input fall time / t_{EXF} / — / 6 / ns CKIO clock input frequency / t_{CKI} / 24 / 33.34 / MHz / 32.4 CKIO clock input cycle time / t_{CKIcyc} / 30 / 41.7 / ns CKIO clock input low pulse width / t_{CKIL} / 9 / — / ns CKIO clock input high pulse width / t_{CKIH} / 9 / — / ns CKIO clock input rise time / t_{CKIR} / — / 6 / ns CKIO clock input fall time / t_{CKIF} / — / 6 / ns CKIO clock output frequency / t_{OP} / 24 / 33.34 / MHz / 32.5 CKIO clock output cycle time / t_{cyc} / 30 / 41.7 / ns
934	Table 32.5 Clock Timing (2) Conditions: VccQ = 3.0 to 3.6 V, Vcc = 1.75 to 2.05 V, AVcc = 3.3 ± 0.3 V, Ta = –20 to 75°C, external bus maximum operating frequency = 66.67 MHz Item / Symbol / Min / Max / Unit / Figure EXTAL clock input frequency / t_{EX} / 6 / 66.67 / MHz / 32.1 EXTAL clock input cycle time / t_{EXcyc} / 15.2 / 167 / ns EXTAL clock input low pulse width / t_{EXL} / 1.5 / — / ns EXTAL clock input high pulse width / t_{EXH} / 1.5 / — / ns CKIO clock input frequency / t_{CKI} / 24 / 66.67 / MHz / 32.2 CKIO clock input cycle time / t_{CKIcyc} / 15.2 / 40 / ns CKIO clock input low pulse width / t_{CKIL} / 1.5 / — / ns CKIO clock input high pulse width / t_{CKIH} / 1.5 / — / ns CKIO clock output frequency / t_{OP} / 24 / 66.67 / MHz / 32.3 CKIO clock output cycle time / t_{cyc} / 15.2 / — / ns Power-on oscillation settling time / t_{OSC1} / 10 / — / ns / 32.4	Table 32.6 Clock Timing (2) Conditions: VccQ = 3.0 to 3.6 V, Vcc = 1.70 to 2.05 V, AVcc = 3.3 ± 0.3 V, Ta = –20 to 75°C, 100 MHz products Item / Symbol / Min / Max / Unit / Figure EXTAL clock input frequency / t_{EX} / 6 / 50 / MHz / 32.3 EXTAL clock input cycle time / t_{EXcyc} / 20 / 166.7 / ns EXTAL clock input low pulse width / t_{EXL} / 4 / — / ns EXTAL clock input high pulse width / t_{EXH} / 4 / — / ns CKIO clock input frequency / t_{CKI} / 24 / 50 / MHz / 32.4 CKIO clock input cycle time / t_{CKIcyc} / 20 / 41.7 / ns CKIO clock input low pulse width / t_{CKIL} / 4 / — / ns CKIO clock input high pulse width / t_{CKIH} / 4 / — / ns CKIO clock output frequency / t_{OP} / 24 / 50 / MHz / 32.5 CKIO clock output cycle time / t_{cyc} / 20 / 41.7 / ns Power-on oscillation settling time / t_{OSC1} / 10 / — / ms / 32.6
935		Table 32.7 Clock Timing (3) *Newly added*
936		Table 32.8 Clock Timing (4) *Newly added*
938	Figure 32.3 CKIO Clock Output Timing 	Figure 32.5 CKIO Clock Output Timing 

Page	Previous Version	Revised Version
942	Table 32.6 Control Signal Timing	Table 32.9 Control Signal Timing

		33 MHz*2		66.67 MHz*3	
		Min	Max	Min	Max
$\overline{\text{RESETP}}$ pulse width	t_{RESPW}	20	—	20*4	—
$\overline{\text{RESETP}}$ setup time*1	t_{RESPS}	23	—	23	—
$\overline{\text{RESETP}}$ hold time	t_{RESPH}	2	—	2	—
$\overline{\text{RESETM}}$ pulse width	t_{RESMW}	12	—	12*5	—
$\overline{\text{RESETM}}$ setup time	t_{RESMS}	3	—	3	—
$\overline{\text{RESETM}}$ hold time	t_{RESMH}	34	—	34	—
$\overline{\text{BREQ}}$ setup time	t_{BREQS}	10	—	10	—
$\overline{\text{BREQ}}$ hold time	t_{BREQH}	3	—	3	—
NMI setup time *1	t_{NMIS}	10	—	10	—
NMI hold time	t_{NMIH}	4	—	4	—
IRQ5–IRQ0 setup time *1	t_{IRQS}	10	—	10	—
IRQ5–IRQ0 hold time	t_{IRQH}	4	—	4	—
$\overline{\text{BACK}}$ delay time	t_{BACKD}	—	10	—	10
STATUS1, STATUS0 delay time	t_{STD}	—	16	—	16
Bus tri-state delay time 1	t_{BOFF1}	0	15	0	15
Bus tri-state delay time 2	t_{BOFF2}	0	15	0	15
Bus buffer-on time 1	t_{BON1}	0	15	0	15
Bus buffer-on time 2	t_{BON2}	0	15	0	15

Notes:

*1 $\overline{\text{RESETP}}$, NMI and IRQ5 to IRQ0 are asynchronous. Changes are detected at the clock fall when the setup shown is used. When the setup cannot be used, detection can be delayed until the next clock falls. When using as IRL, please observe the setup time.

*2 When Vcc = 1.6 to 2.05 V and VccQ = 2.6 to 3.6 V, the upper limit of the external bus clock is 33 MHz.

*3 When Vcc = 1.75 to 2.05 V and VccQ = 3.0 to 3.6 V, the upper limit of the external bus clock is 66.67 MHz.

*4 In the standby mode, $t_{RESPW} = t_{OSC2}$ (10 ms). In the sleep mode, $t_{RESPW} = t_{PLL1}$ (100 µs). When the clock multiplication ratio is changed, $t_{RESPW} = t_{PLL1}$ (100 µs).

		Min	Max
$\overline{\text{RESETP}}$ pulse width	t_{RESPW}	20*2	—
$\overline{\text{RESETP}}$ setup time*1	t_{RESPS}	23	—
$\overline{\text{RESETP}}$ hold time	t_{RESPH}	2	—
$\overline{\text{RESETM}}$ pulse width	t_{RESMW}	12*3	—
$\overline{\text{RESETM}}$ setup time	t_{RESMS}	3	—
$\overline{\text{RESETM}}$ hold time	t_{RESMH}	34	—
$\overline{\text{BREQ}}$ setup time	t_{BREQS}	10	—
$\overline{\text{BREQ}}$ hold time	t_{BREQH}	3	—
NMI setup time *1	t_{NMIS}	10	—
NMI hold time	t_{NMIH}	4	—
IRQ5–IRQ0 setup time *1	t_{IRQS}	10	—
IRQ5–IRQ0 hold time	t_{IRQH}	4	—
$\overline{\text{BACK}}$ delay time	t_{BACKD}	—	10
STATUS1, STATUS0 delay time	t_{STD}	—	16
Bus tri-state delay time 1	t_{BOFF1}	0	15
Bus tri-state delay time 2	t_{BOFF2}	0	15
Bus buffer-on time 1	t_{BON1}	0	15
Bus buffer-on time 2	t_{BON2}	0	15

1. $\overline{\text{RESETP}}$, NMI and IRQ5 to IRQ0 are asynchronous. Changes are detected at the clock fall when the setup shown is used. When the setup cannot be used, detection can be delayed until the next clock falls. When using as IRL, please observe the setup time.

2. In the standby mode, $t_{RESPW} = t_{OSC2}$ (10 ms). In the sleep mode, $t_{RESPW} = t_{PLL1}$ (100 µs). When the clock multiplication ratio is changed, $t_{RESPW} = t_{PLL1}$ (100 µs).

Page	Previous Version	Revised Version
945, 946	Table 32.7 Bus Timing	Table 32.10 Bus Timing

Previous Version

Item	Symbol	33 MHz*1 Min	33 MHz*1 Max	66.67 MHz*2 Min	66.67 MHz*2 Max	Unit
Address delay time	t_{AD}	1.5	16	1.5	13	ns
Address setup time	t_{AS}	0	—	0	—	ns
Address hold time	t_{AH}*3	7	—	7	—	ns
$\overline{BS}$ delay time	t_{BSD}	—	12	—	12	ns
$\overline{CS}$ delay time 1	t_{CSD1}	1.5	12	1.5	12	ns
$\overline{CS}$ delay time 2	t_{CSD2}	1	12	1	12	ns
Read/write delay time	t_{RWD}	1.5	10	1.5	10	ns
Read/write hold time	t_{RWH}	0	—	0	—	ns
Read strobe delay time	t_{RSD}	—	10	—	10	ns
Read data setup time 1	t_{RDS1}	6	—	6	—	ns
Read data setup time 2	t_{RDS2}	7	—	7	—	ns
Read data hold time 1	t_{RDH1}*4	0	—	0	—	ns
Read data hold time 2	t_{RDH2}	2	—	2	—	ns
Write enable delay time	t_{WED}	1	10	1	10	ns
Write data delay time 1	t_{WDD1}	—	14	—	14	ns
Write data delay time 2	t_{WDD2}	—	13	—	13	ns

Item	Symbol	33 MHz*1 Min	33 MHz*1 Max	66.67 MHz*2 Min	66.67 MHz*2 Max	Unit
Write data hold time 1	t_{WDH1}	1.5	—	1.5	—	ns
Write data hold time 2	t_{WDH2}	1.5	—	1.5	—	ns
Write data hold time 3	t_{WDH3}	2	—	2	—	ns
Write data hold time 4	t_{WDH4}	2	—	2	—	ns
$\overline{WAIT}$ setup time	t_{WTS}	6	—	5	—	ns
$\overline{WAIT}$ hold time	t_{WTH}	0	—	0	—	ns
$\overline{RAS}$ delay time 2	t_{RASD2}	1.5	12	1.5	12	ns
$\overline{CAS}$ delay time 2	t_{CASD2}	1.5	12	1.5	12	ns
$\overline{DQM}$ delay time	t_{DQMD}	1.5	10	1.5	10	ns
$\overline{CKE}$ delay time	t_{CKED}	1.5	12	1.5	12	ns
$\overline{ICIORD}$ delay time	t_{ICRSD}	—	12	—	12	ns
$\overline{ICIOWR}$ delay time	t_{ICWSD}	—	12	—	12	ns
$\overline{IOIS16}$ setup time	t_{IO16S}	12	—	12	—	ns
$\overline{IOIS16}$ hold time	t_{IO16H}	4	—	4	—	ns
$\overline{DACK}$ delay time 1	t_{DAKD1}	—	10	—	10	ns

Notes:

*1 When Vcc = 1.6 to 2.05 V and VccQ =2.6 to 3.6 V, the upper limit of the external bus clock is 33 MHz.

*2 When Vcc = 1.75 to 2.05 V and VccQ = 3.0 to 3.6 V, the upper limit of the external bus clock is 66.67 MHz.

*3 t_{AH}: This is to deal with the latest negate timing of $\overline{CS}$n, RD, or $\overline{WE}$n.

*4 t_{RDH1}: This is to deal with the earliest negate timing of $\overline{CS}$n or RD.

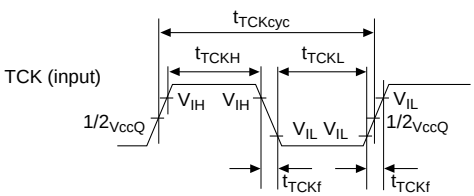
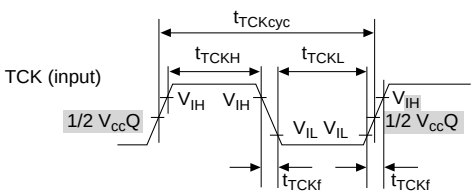
Revised Version

Item	Symbol	Min	Max	Unit	Conditions
Address delay time	t_{AD}	1.5	13	ns	Vcc = 1.70 to 2.05 V, VccQ = 3.0 to 3.6 V
		1.5	16		Other than the above
Address setup time	t_{AS}	0	—	ns	
Address hold time	t_{AH}*1	7	—	ns	
$\overline{BS}$ delay time	t_{BSD}	—	12	ns	
$\overline{CS}$ delay time 1	t_{CSD1}	1.5	12	ns	
$\overline{CS}$ delay time 2	t_{CSD2}	1	12	ns	
Read/write delay time	t_{RWD}	1.5	10	ns	
Read/write hold time	t_{RWH}	0	—	ns	
Read strobe delay time	t_{RSD}	—	10	ns	
Read data setup time 1	t_{RDS1}	6	—	ns	
Read data setup time 2	t_{RDS2}	7	—	ns	
Read data hold time 1	t_{RDH1}*2	0	—	ns	
Read data hold time 2	t_{RDH2}	2	—	ns	
Write enable delay time	t_{WED}	1	10	ns	
Write data delay time 1	t_{WDD1}	—	14	ns	
Write data delay time 2	t_{WDD2}	—	13	ns	

Item	Symbol	Min	Max	Unit	Conditions
Write data hold time 1	t_{WDH1}	1.5	—	ns	
Write data hold time 2	t_{WDH2}	1.5	—	ns	
Write data hold time 3	t_{WDH3}	2	—	ns	
Write data hold time 4	t_{WDH4}	2	—	ns	
$\overline{WAIT}$ setup time	t_{WTS}	5	—	ns	Vcc = 1.70 to 2.05 V, VccQ = 3.0 to 3.6 V
		6	—		Other than the above
$\overline{WAIT}$ hold time	t_{WTH}	0	—	ns	
$\overline{RAS}$ delay time 2	t_{RASD2}	1.5	12	ns	
$\overline{CAS}$ delay time 2	t_{CASD2}	1.5	12	ns	
$\overline{DQM}$ delay time	t_{DQMD}	1.5	10	ns	
$\overline{CKE}$ delay time	t_{CKED}	1.5	12	ns	
$\overline{ICIORD}$ delay time	t_{ICRSD}	—	12	ns	
$\overline{ICIOWR}$ delay time	t_{ICWSD}	—	12	ns	
$\overline{IOIS16}$ setup time	t_{IO16S}	12	—	ns	
$\overline{IOIS16}$ hold time	t_{IO16H}	4	—	ns	
DACK delay time 1	t_{DAKD1}	—	10	ns	

1. t_{AH}: This is to deal with the latest negate timing of $\overline{CS}$n, RD, or $\overline{WE}$n.

2. t_{RDH1}: This is to deal with the earliest negate timing of $\overline{CS}$n or RD.

Page	Previous Version	Revised Version
949	Figure 32.17 Basic Bus Cycle (External Wait, WAITSEL = 1) 	Figure 32.19 Basic Bus Cycle (External Wait, WAITSEL = 1)  *Note added* Notes: t_{RDH1}: Specified based on the earliest negate timing of $\overline{CSn}$ or $\overline{RD}$. t_{AH}: Specified based on the latest negate timing of $\overline{CSn}$, $\overline{RD}$, or $\overline{WEn}$.
971	Table 32.8 Peripheral Module Signal Timing **–66.67** **Min Max** Note: * P_{cyc} stands for “P clock cycle.”	Table 32.11 Peripheral Module Signal Timing **Min Max** Note: * P_{cyc} stands for “peripheral clock (Pϕ) cycle.”
973	Figure 32.42 I/O Port Timing PORT 7 to 0 (read) (B:P clock ratio = 1:1) PORT 7 to 0 (read) (B:P clock ratio = 1:2) PORT 7 to 0 (read) (B:P clock ratio = 1:4) PORT 7 to 0 (write)	Figure 32.44 I/O Port Timing PORT A to H, J to M, SC (read) (bus clock:peripheral clock ratio = 1:1) PORT A to H, J to M, SC (read) (bus clock:peripheral clock ratio = 1:1/2) PORT A to H, J to M, SC (read) (bus clock:peripheral clock ratio = 1:1/4) PORT A to H, J to M, SC (write)
974	Figure 32.45 TCK Input Timing  Note: When clock is input from TCK pin	Figure 32.47 TCK Input Timing 

Page	Previous Version	Revised Version
981	Table 32.12 USB Module Signal Timing Item / Symbol / Min / Max / Unit / Figure UCLK external input clock frequency (48 MHz) / t_{FREQ} / 47.9 / 48.1 / MHz / 32.56 Clock rise time / t_{R48} / — / 2 / ns Clock fall time / t_{F48} / — / 2 / ns Duty (t_{HIGH}/t_{LOW}) / t_{DUTY} / 90 / 110 / %	Table 32.15 USB Module Signal Timing Item / Symbol / Min / Max / Unit / Figure UCLK external input clock frequency (48 MHz) / t_{FREQ} / 47.9 / 48.1 / MHz / 32.58 Clock rise time / t_{R48} / — / 6 / ns Clock fall time / t_{F48} / — / 6 / ns
983	Table 32.15 AFEIF Module Signal Timing Note: t_{PCYC} is the cycle time (ns) of the peripheral clock (P clock).	Table 32.18 AFEIF Module Signal Timing Note: t_{PCYC} is the cycle time (ns) of the peripheral clock (Pϕ).
985	32.3.14 AC Characteristics Measurement Conditions • Input pulse level: Vss to 3.0 V (where $\overline{\text{RESETP}}$, $\overline{\text{RESETM}}$, $\overline{\text{ASEMD0}}$, $\overline{\text{IRL3}}$ to $\overline{\text{IRL0}}$, $\overline{\text{ADTRG}}$, PINT[15] to PINT[0], CA, $\overline{\text{NMI}}$, IRQ5 to IRQ0, CKIO, and MD5 to MD0 are within VssQ to VccQ)	• Input pulse level: VssQ to 3.0 V (where $\overline{\text{RESETP}}$, $\overline{\text{RESETM}}$, $\overline{\text{ASEMD0}}$, $\overline{\text{IRL3}}$ to $\overline{\text{IRL0}}$, $\overline{\text{ADTRG}}$, PINT[15] to PINT[0], CA, $\overline{\text{NMI}}$, IRQ5 to IRQ0, CKIO, and MD5 to MD0 are within VssQ to VccQ)
991, 992	Table A.1 Pin Functions (cont) Type / Signal Name (Initial Status: Bold) / Pin No. (HQFP) / I/O / Power-On Reset / Manual Reset / Standby / Release/ Open Bus Privileges AFE/USB digital/port related / **PTM[7]/PINT[7]/** AFE_FS/USB1d_RCV, **PTM[6]/PINT[6]/** AFE_RXIN/USB1d_SPEED, **PTM[5]/PINT[5]/** AFE_TXOUT/ USB1d_TXSE0 / 118, 119, 120 / I/I/I/O / V / I/I/I/O / Z(V)/I/Z/O / I/I/I/O **PTM[4]/PINT[4]/** AFE_RDET/ USB1d_TXDMNS / 121 / I/I/I/O / V / I/I/I/O / Z(V)/I/Z/O / I/I/I/O **USB1d_SUSPEND** / 122 / O / O / O / O / O Serial related / **SIOMCLK**/SCPT[3] / 194 / I/IO / I / Z/P / Z/K / I/P **SCK_SIO**/SCPT[5], **SIOFSYNC**/SCPT[6] / 196, 197 / IO/IO / I / Z/P / Z/K / IO/P **RxD0**/SCPT[0], **RxD2**/SCPT[4] / 198, 201 / I/I / Z / Z/I / Z/Z / I/Z	Table A.1 Pin Functions (cont) Type / Signal Name (Initial Status: Bold) / Pin No. (HQFP) / I/O / Power-On Reset / Manual Reset / Standby / Release/ Open Bus Privileges AFE/USB digital/port related / **PTM[7]/PINT[7]/** AFE_FS/USB1d_RCV, **PTM[6]/PINT[6]/** AFE_RXIN/USB1d_SPEED, **PTM[5]/PINT[5]/** AFE_TXOUT/ USB1d_TXSE0 / 118, 119, 120 / I/I/I/O / V / I/I/I/O / Z(V)/I/Z/O / I/I/I/O **PTM[4]/PINT[4]/** AFE_RDET / 121 / I/I/I / V / I/I/I / Z(V)/I/Z / I/I/I **USB1d_SUSPEND** / 122 / O / O / O / O / O Serial related / **SIOMCLK**/SCPT[3] / 194 / I/IO / I / Z/P / Z/K / I/P **SCK_SIO**/SCPT[5] / 196 / IO/IO / I / Z/P / Z/K / IO/P **SIOFSYNC**/SCPT[6] / 197 / IO/IO / I / Z/P / K/K / IO/P **RxD0**/SCPT[0], **RxD2**/SCPT[4] / 198, 201 / I/I / Z / Z/I / Z/Z / I/Z
997	Table A.2 Treatment of Unused Pins (cont) Type / Signal Name (Initial Status: Bold) / Pin No (HQFP) / Pin No (CSP) / I/O / Treatment when Not Used AFE/USB digital/port related / **PTM[7]/PINT[7]**/AFE_FS/ USB1d_RCV, **PTM[6]/PINT[6]**/AFE_RXIN/ USB1d_SPEED, **PTM[5]/PINT[5]**/AFE_TXOUT/ USB1d_TXSE0 / 118, 119, 120 / V19, T18, V18 / I/I/I/O / Open **PTM[4]/PINT[4]**/AFE_RDET/ USB1d_TXDMNS / 121 / W19 / I/I/I/O / Open **USB1d_SUSPEND** / 122 / V16 / O / Open	Table A.2 Treatment of Unused Pins (cont) Type / Signal Name (Initial Status: Bold) / Pin No (HQFP) / Pin No (CSP) / I/O / Treatment when Not Used AFE/USB digital/port related / **PTM[7]/PINT[7]**/AFE_FS/ USB1d_RCV, **PTM[6]/PINT[6]**/AFE_RXIN/ USB1d_SPEED, **PTM[5]/PINT[5]**/AFE_TXOUT/ USB1d_TXSE0 / 118, 119, 120 / V19, T18, V18 / I/I/I/O / Open **PTM[4]/PINT[4]**/AFE_RDET / 121 / W19 / I/I/I / Open **Reserved/USB1d_SUSPEND** / 122 / V16 / O / Open

<table>
<tr><th>Page</th><th>Previous Version</th><th>Revised Version</th></tr>
<tr><td>1001</td><td>Table A.3 Pin Status (Normal Memory/Little Endian) (cont)

Note: 2. Unused pins can be switched to port function, pull-up, or pull-down.</td><td>

Note: 2. Unused pins can be switched to port function, pull-up.</td></tr>
<tr><td>1003</td><td>Table A.4 Pin Status (Normal Memory/Big Endian) (cont)

Note: 2. Unused pins can be switched to port function, pull-up, or pull-down.</td><td>

Note: 2. Unused pins can be switched to port function, pull-up.</td></tr>
<tr><td>1005</td><td>Table A.5 Pin Status (Burst ROM/Little Endian) (cont)

Note: 2. Unused pins can be switched to port function, pull-up, or pull-down.</td><td>

Note: 2. Unused pins can be switched to port function, pull-up.</td></tr>
<tr><td>1007</td><td>Table A.6 Pin Status (Burst ROM/Big Endian) (cont)

Note: 2. Unused pins can be switched to port function, pull-up, or pull-down.</td><td>

Note: 2. Unused pins can be switched to port function, pull-up.</td></tr>
<tr><td>1011</td><td>Table A.9 Pin Status (PCMCIA/Little Endian) (cont)

Note: 2. Unused pins can be switched to port function, pull-up, or pull-down.</td><td>

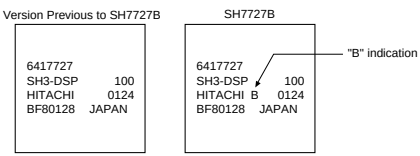
Note: 2. Unused pins can be switched to port function, pull-up.</td></tr>
<tr><td>1013</td><td>Table A.10 Pin Status (PCMCIA/Big Endian) (cont)

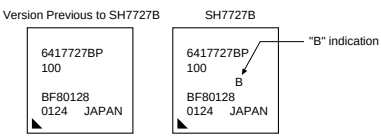
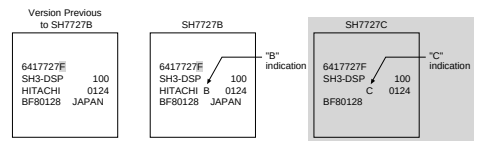
Note: 2. Unused pins can be switched to port function, pull-up, or pull-down.</td><td>

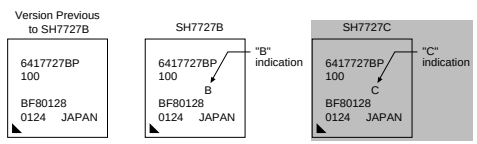
Note: 2. Unused pins can be switched to port function, pull-up.</td></tr>
<tr><td>1018, 1019, 1021</td><td>Table B.1 Memory-Mapped Control Registers (Address Map)

<table>
<tr><th>Control Register</th><th>Module*1</th><th>Bus*2</th><th>Address*4</th><th>Size (Bits)</th><th>Access Size (Bits)*3</th></tr>
<tr><td>SIRCR</td><td>SIOF</td><td>P2</td><td>H'040000EC</td><td>32</td><td>32</td></tr>
<tr><td>SITMR</td><td>SIOF</td><td>P2</td><td>H'040000FC</td><td>16</td><td>16</td></tr>
<tr><td>SIFPR</td><td>SIOF</td><td>P2</td><td>H'040000FE</td><td>16</td><td>16</td></tr>
<tr><td>PACR</td><td>PORT</td><td>P</td><td>H'04000100</td><td>16</td><td>16</td></tr>
<tr><td>PBCR</td><td>PORT</td><td>P</td><td>H'04000102</td><td>16</td><td>16</td></tr>
<tr><td>SDIR</td><td>UDI</td><td>I2</td><td>H'04000200</td><td>16</td><td>16</td></tr>
<tr><td>SDDR/SDDRH</td><td>UDI</td><td>I2</td><td>H'04000208</td><td>16/32</td><td>16/32</td></tr>
<tr><td>SDDRL</td><td>UDI</td><td>I2</td><td>H'0400020A</td><td>16</td><td>16</td></tr>
<tr><td>IPRF</td><td>PPCNT</td><td>P2</td><td>H'04000220</td><td>16</td><td>16</td></tr>
<tr><td>IPRG</td><td>PPCNT</td><td>P2</td><td>H'04000222</td><td>16</td><td>16</td></tr>
<tr><td>HcRhPortStatus2 (USBHRPS2)</td><td>USBH</td><td>P2</td><td>H'04000458</td><td>32</td><td>32</td></tr>
<tr><td>LDPR00</td><td>LCDC</td><td>P2</td><td>H'04000800</td><td>32</td><td>32</td></tr>
<tr><td>LDICKR</td><td>LCDC</td><td>P2</td><td>H'04000C00</td><td>16</td><td>16</td></tr>
</table>
</td><td>

<table>
<tr><th>Control Register</th><th>Module*1</th><th>Bus*2</th><th>Address*4</th><th>Size (Bits)</th><th>Access Size (Bits)*3</th></tr>
<tr><td>SIRCR</td><td>SIOF</td><td>P2</td><td>H'040000EC</td><td>32</td><td>32</td></tr>
<tr><td>PACR</td><td>PORT</td><td>P</td><td>H'04000100</td><td>16</td><td>16</td></tr>
<tr><td>PBCR</td><td>PORT</td><td>P</td><td>H'04000102</td><td>16</td><td>16</td></tr>
<tr><td>SDIR</td><td>H-UDI</td><td>I2</td><td>H'04000200</td><td>16</td><td>16</td></tr>
<tr><td>IPRF</td><td>PPCNT</td><td>P2</td><td>H'04000220</td><td>16</td><td>16</td></tr>
<tr><td>IPRG</td><td>PPCNT</td><td>P2</td><td>H'04000222</td><td>16</td><td>16</td></tr>
<tr><td>HcRhPortStatus2 (USBHRPS2)</td><td>USBH</td><td>P2</td><td>H'04000458</td><td>32</td><td>32</td></tr>
<tr><td>LDPR00 to LDPRFF</td><td>LCDC</td><td>P2</td><td>H'04000800 to H'04000BFC</td><td>32</td><td>32</td></tr>
<tr><td>LDICKR</td><td>LCDC</td><td>P2</td><td>H'04000C00</td><td>16</td><td>16</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Page</th><th>Previous Version</th><th>Revised Version</th></tr>
<tr><td>1024</td><td>Appendix C Product Lineup
<table><tr><th>Model Name</th><th>Package</th></tr><tr><td>HD6417727F160B</td><td>240-pin plastic HQFP (FP-240B)</td></tr><tr><td>HD6417727BP160B</td><td>240-pin CSP (BP-240A)</td></tr><tr><td>HD6417727F100B</td><td>240-pin plastic HQFP (FP-240B)</td></tr><tr><td>HD6417727BP100B</td><td>240-pin CSP (BP-240A)</td></tr></table></td><td><table><tr><th>Model Name</th><th>Package</th></tr><tr><td>HD6417727F160C</td><td>240-pin plastic HQFP (PRQP0240KC-B)</td></tr><tr><td>HD6417727BP160C</td><td>240-pin CSP (PLBG0240JA-A)</td></tr><tr><td>HD6417727F100C</td><td>240-pin plastic HQFP (PRQP0240KC-B)</td></tr><tr><td>HD6417727BP100C</td><td>240-pin CSP (PLBG0240JA-A)</td></tr></table></td></tr>
<tr><td>1025</td><td>Figure D.1 Package Dimensions (FP-240B)</td><td>Figure D.1 Package Dimensions (PRQP0240KC-B)
Replaced</td></tr>
<tr><td>1026</td><td>Figure D.2 Package Dimensions (BP-240A)</td><td>Figure D.2 Package Dimensions (PLBG0240JA-A)
Replaced</td></tr>
<tr><td>1027</td><td>E.1 Determining the Version Number Based on the Markings on the Chip
(1) HQPF-240 Package


(2) CSP-240 Package

</td><td>(1) HQFP-240 Package


Note: Once stocks bearing Hitachi markings are exhausted, chips bearing Renesas markings may begin to appear.
(2) CSP-240 Package

</td></tr>
</table>

Contents

Section 1 Overview and Pin Functions 1
1.1 Features 1
1.2 Block Diagram 8
1.3 Pin Description 9
1.3.1 Pin Arrangement 9
1.3.2 Pin Functions 11

Section 2 CPU 21
2.1 Registers 21
2.1.1 General Purpose Registers 25
2.1.2 Control Registers 27
2.1.3 System Registers 31
2.1.4 DSP Registers 31
2.2 Data Format 38
2.2.1 Data Format in Registers (Non-DSP Type) 38
2.2.2 DSP-Type Data Format 38
2.2.3 Data Format in Memory 40
2.3 Features of CPU Core Instructions 40
2.4 Instruction Formats 44
2.4.1 CPU Instruction Addressing Modes 44
2.4.2 DSP Data Addressing 48
2.4.3 CPU Instruction Formats 54
2.4.4 DSP Instruction Formats 58
2.5 Instruction Set 64
2.5.1 CPU Instruction Set 64
2.6 DSP Extended-Function Instructions 79
2.6.1 Introduction 79
2.6.2 Added CPU System Control Instructions 80
2.6.3 Single and Double Data Transfer for DSP Data Instructions 82
2.6.4 DSP Operation Instruction Set 85

Section 3 Memory Management Unit (MMU) 97
3.1 Overview 97
3.1.1 Features 97
3.1.2 Role of MMU 97
3.1.3 SH7727 MMU 99
3.1.4 Register Configuration 103
3.2 Register Description 103

3.3 TLB Functions 105
3.3.1 Configuration of the TLB 105
3.3.2 TLB Indexing 107
3.3.3 TLB Address Comparison 108
3.3.4 Page Management Information 110
3.4 MMU Functions 111
3.4.1 MMU Hardware Management 111
3.4.2 MMU Software Management 111
3.4.3 MMU Instruction (LDTLB) 112
3.4.4 Avoiding Synonym Problems 113
3.5 MMU Exceptions 116
3.5.1 TLB Miss Exception 116
3.5.2 TLB Protection Violation Exception 117
3.5.3 TLB Invalid Exception 118
3.5.4 Initial Page Write Exception 119
3.5.5 Processing Flow in Event of MMU Exception (Same Processing Flow for Address Error) 121
3.5.6 MMU Exception in Repeat Loop 123
3.6 Memory-Mapped TLB 124
3.6.1 Address Array 125
3.6.2 Data Array 125
3.6.3 Usage Examples 127
3.7 Usage Notes 128

Section 4 Exception Handling 131
4.1 Overview 131
4.1.1 Features 131
4.1.2 Register Configuration 131
4.2 Exception Handling Function 131
4.2.1 Exception Handling Flow 131
4.2.2 Exception Handling Vector Addresses 132
4.2.3 Acceptance of Exceptions 134
4.2.4 Exception Codes 136
4.2.5 Exception Request Masks 137
4.2.6 Returning from Exception Handling 138
4.3 Register Description 138
4.4 Exception Handling Operation 139
4.4.1 Reset 139
4.4.2 Interrupts 139
4.4.3 General Exceptions 140
4.5 Individual Exception Operations 140

4.5.1 Resets 140
4.5.2 General Exceptions 141
4.5.3 Interrupts 146
4.6 Usage Notes 147

Section 5 Cache 149
5.1 Overview 149
5.1.1 Features 149
5.1.2 Cache Structure 149
5.1.3 Register Configuration 151
5.2 Register Description 151
5.2.1 Cache Control Register (CCR) 151
5.2.2 Cache Control Register 2 (CCR2) 152
5.3 Cache Operation 154
5.3.1 Searching the Cache 154
5.3.2 Read Access 156
5.3.3 Prefetch Operations 156
5.3.4 Write Access 156
5.3.5 Write-Back Buffer 157
5.3.6 Coherency of Cache and External Memory 157
5.4 Memory-Mapped Cache 157
5.4.1 Address Array 157
5.4.2 Data Array 158
5.5 Usage Examples 160
5.5.1 Invalidating Specific Entries 160
5.5.2 Reading the Data of a Specific Entry 160

Section 6 X/Y Memory 161
6.1 Overview 161
6.1.1 Features 161
6.2 X/Y Memory Access from the CPU 162
6.3 X/Y Memory Access from the DSP 164
6.4 X/Y Memory Access from the DMAC 164

Section 7 Interrupt Controller (INTC) 165
7.1 Overview 165
7.1.1 Features 165
7.1.2 Block Diagram 166
7.1.3 Pin Configuration 167
7.1.4 Register Configuration 167
7.2 Interrupt Sources 169

7.2.1 NMI Interrupts 169
7.2.2 IRQ Interrupt 169
7.2.3 IRL Interrupts 170
7.2.4 PINT Interrupt 172
7.2.5 On-Chip Supporting Module Interrupts 172
7.2.6 Interrupt Exception Handling and Priority 173
7.3 INTC Registers 179
7.3.1 Interrupt Priority Registers A to G (IPRA to IPRG) 179
7.3.2 Interrupt Control Register 0 (ICR0) 180
7.3.3 Interrupt Control Register 1 (ICR1) 181
7.3.4 Interrupt Control Register 2 (ICR2) 184
7.3.5 Interrupt Control Register 3 (ICR3) 185
7.3.6 PINT Interrupt Enable Register (PINTER) 187
7.3.7 Interrupt Request Register 0 (IRR0) 187
7.3.8 Interrupt Request Register 1 (IRR1) 190
7.3.9 Interrupt Request Register 2 (IRR2) 191
7.3.10 Interrupt Request Register 3 (IRR3) 192
7.3.11 Interrupt Request Register 4 (IRR4) 195
7.4 INTC Operation 197
7.4.1 Interrupt Sequence 197
7.4.2 Multiple Interrupts 199
7.5 Interrupt Response Time 199

Section 8 User Break Controller 203
8.1 Overview 203
8.1.1 Features 203
8.1.2 Block Diagram 204
8.1.3 Register Configuration 205
8.2 Register Descriptions 206
8.2.1 Break Address Register A (BARA) 206
8.2.2 Break Address Mask Register A (BAMRA) 206
8.2.3 Break Bus Cycle Register A (BBRA) 207
8.2.4 Break Address Register B (BARB) 209
8.2.5 Break Address Mask Register B (BAMRB) 210
8.2.6 Break Data Register B (BDRB) 211
8.2.7 Break Data Mask Register B (BDMRB) 212
8.2.8 Break Bus Cycle Register B (BBRB) 213
8.2.9 Break Control Register (BRCR) 215
8.2.10 Execution Times Break Register (BETR) 218
8.2.11 Branch Source Register (BRSR) 219
8.2.12 Branch Destination Register (BRDR) 220

8.2.13 Break ASID Register A (BASRA) ... 221
8.2.14 Break ASID Register B (BASRB) ... 221
8.3 Operation Description ... 222
8.3.1 Flow of the User Break Operation ... 222
8.3.2 Break on Instruction Fetch Cycle ... 222
8.3.3 Break by Data Access Cycle ... 223
8.3.4 Break on X/Y-Memory Bus Cycle ... 224
8.3.5 Sequential Break ... 224
8.3.6 Value of Saved Program Counter ... 224
8.3.7 PC Trace ... 225
8.3.8 Usage Examples ... 227
8.3.9 Usage Notes ... 231

Section 9 Power-Down Modes and Software Reset ... 233
9.1 Overview ... 233
9.1.1 Power-Down Modes ... 233
9.1.2 Pin Configuration ... 235
9.1.3 Register Configuration ... 235
9.2 Register Description ... 236
9.2.1 Standby Control Register (STBCR) ... 236
9.2.2 Standby Control Register 2 (STBCR2) ... 237
9.2.3 Standby Control Register 3 (STBCR3) ... 239
9.2.4 Module Software Reset Register (SRSTR) ... 241
9.3 Sleep Mode ... 243
9.3.1 Transition to Sleep Mode ... 243
9.3.2 Canceling Sleep Mode ... 243
9.4 Standby Mode ... 243
9.4.1 Transition to Standby Mode ... 243
9.4.2 Canceling Standby Mode ... 245
9.4.3 Clock Pause Function ... 246
9.5 Module Standby Function ... 247
9.5.1 Transition to Module Standby Function ... 247
9.5.2 Clearing the Module Standby Function ... 249
9.6 Timing of STATUS Pin Changes ... 249
9.6.1 Timing for Resets ... 249
9.6.2 Timing for Canceling Standbys ... 250
9.6.3 Timing for Canceling Sleep Mode ... 252
9.7 Hardware Standby Mode ... 254
9.7.1 Transition to Hardware Standby Mode ... 254
9.7.2 Clearing the Hardware Standby Mode ... 254
9.7.3 Timing of Hardware Standby Mode ... 255

Section 10 On-Chip Oscillation Circuits 257
10.1 Overview 257
10.1.1 Features 257
10.2 Overview of the CPG 259
10.2.1 CPG Block Diagram 259
10.2.2 CPG Pin Configuration 261
10.2.3 CPG Register Configuration 261
10.3 Clock Operating Modes 262
10.4 Register Descriptions 267
10.4.1 Frequency Control Register (FRQCR) 267
10.4.2 CKIO2 Control Register (CKIO2CR) 269
10.5 Changing the Frequency 270
10.5.1 Changing the Multiplication Rate 270
10.5.2 Changing the Division Ratio 270
10.6 Overview of the WDT 271
10.6.1 Block Diagram of the WDT 271
10.6.2 Register Configurations 272
10.7 WDT Registers 272
10.7.1 Watchdog Timer Counter (WTCNT) 272
10.7.2 Watchdog Timer Control/Status Register (WTCSR) 273
10.7.3 Notes on Register Access 275
10.8 Using the WDT 275
10.8.1 Canceling Standby Mode 275
10.8.2 Changing the Frequency 276
10.8.3 Using Watchdog Timer Mode 276
10.8.4 Using Interval Timer Mode 277
10.9 Notes on Board Design 277

Section 11 Extend Clock Pulse Generator for USB (EXCPG) 279
11.1 Overview 279
11.1.1 EXCPG 279
11.2 Functions 279
11.2.1 Block Diagram 279
11.2.2 Pin Configuration 280
11.2.3 Register Configuration 280
11.3 Register Descriptions 280
11.3.1 EXCPG Control Register (EXCPGCR) 280
11.4 Usage Notes 281

Section 12 Bus State Controller (BSC) 283
12.1 Overview 283
12.1.1 Features 283
12.1.2 Block Diagram 285
12.1.3 Pin Configuration 286
12.1.4 Register Configuration 287
12.1.5 Area Overview 288
12.1.6 PC Card Support 292
12.2 BSC Registers 293
12.2.1 Bus Control Register 1 (BCR1) 293
12.2.2 Bus Control Register 2 (BCR2) 297
12.2.3 Wait State Control Register 1 (WCR1) 298
12.2.4 Wait State Control Register 2 (WCR2) 299
12.2.5 Individual Memory Control Register (MCR) 303
12.2.6 PCMCIA Control Register (PCR) 306
12.2.7 Synchronous DRAM Mode Register (SDMR) 310
12.2.8 Refresh Timer Control/Status Register (RTCSR) 311
12.2.9 Refresh Timer Counter (RTCNT) 313
12.2.10 Refresh Time Constant Register (RTCOR) 314
12.2.11 Refresh Count Register (RFCR) 314
12.2.12 Cautions on Accessing Refresh Control Related Registers 315
12.3 BSC Operation 316
12.3.1 Endian/Access Size and Data Alignment 316
12.3.2 Description of Areas 322
12.3.3 Basic Interface 325
12.3.4 Synchronous DRAM Interface 331
12.3.5 Burst ROM Interface 347
12.3.6 PCMCIA Interface 350
12.3.7 Waits between Access Cycles 362
12.3.8 Bus Arbitration 363
12.3.9 Bus Pull-Up 364

Section 13 Li Bus State Controller (LBSC) 367
13.1 Overview 367
13.1.1 Features 367
13.1.2 Register Configuration 367
13.1.3 Bus Control Register 1 (BCR1) 368
13.1.4 Bus Control Register 2 (BCR2) 370
13.1.5 Wait State Control Register 1 (WCR1) 371
13.1.6 Wait State Control Register 2 (WCR2) 372
13.1.7 Individual Memory Control Register (MCR) 373

13.2 LBSC Operation ... 376
13.2.1 Bus Sharing Architecture ... 376
13.2.2 Usable System Memory ... 376
13.2.3 Bus Arbitration ... 376
13.2.4 LCDC Li Bus Access ... 376
13.2.5 USBH Li Bus Access ... 377
13.2.6 Setting of DMA Transfer with Bus Arbitration of Other Module ... 377

Section 14 Direct Memory Access Controller (DMAC) ... 379
14.1 Overview ... 379
14.1.1 Features ... 379
14.1.2 Block Diagram ... 381
14.1.3 Pin Configuration ... 382
14.1.4 Register Configuration ... 382
14.2 Register Descriptions ... 384
14.2.1 DMA Source Address Registers 0 to 3 (SAR0 to SAR3) ... 384
14.2.2 DMA Destination Address Registers 0 to 3 (DAR0 to DAR3) ... 385
14.2.3 DMA Transfer Count Registers 0 to 3 (DMATCR0 to DMATCR3) ... 386
14.2.4 DMA Channel Control Registers 0 to 3 (CHCR0 to CHCR3) ... 387
14.2.5 DMA Channel Request Assign Register (CHRAR) ... 394
14.2.6 DMA Operation Register (DMAOR) ... 396
14.3 Operation ... 398
14.3.1 DMA Transfer Flow ... 398
14.3.2 DMA Transfer Requests ... 400
14.3.3 Channel Priority ... 402
14.3.4 DMA Transfer Types ... 405
14.3.5 Number of Bus Cycle States and $\overline{\text{DREQ}}$ Pin Sampling Timing ... 418
14.3.6 Source Address Reload Function ... 427
14.3.7 DMA Transfer Ending ... 429
14.4 Compare-Match Timer (CMT) ... 431
14.4.1 Overview ... 431
14.4.2 Register Descriptions ... 432
14.4.3 Operation ... 435
14.4.4 Compare-Match ... 436
14.5 Examples for Use ... 438
14.5.1 Example of DMA Transfer between A/D Converter and External Memory (Address Reload on) ... 438
14.5.2 Example of DMA Transfer between External Memory and SCIF Transmitter (Indirect Address on) ... 439
14.6 Usage Notes ... 441

Section 15 Timer (TMU) 443
15.1 Overview 443
15.1.1 Features 443
15.1.2 Block Diagram 444
15.1.3 Register Configuration 445
15.2 TMU Registers 446
15.2.1 Timer Start Register (TSTR) 446
15.2.2 Timer Control Register (TCR) 447
15.2.3 Timer Constant Register (TCOR) 448
15.2.4 Timer Counters (TCNT) 449
15.3 TMU Operation 451
15.3.1 Overview 451
15.3.2 Basic Functions 451
15.4 Interrupts 453
15.4.1 Status Flag Set Timing 453
15.4.2 Status Flag Clear Timing 454
15.4.3 Interrupt Sources and Priorities 454
15.5 Usage Notes 455
15.5.1 Writing to Registers 455
15.5.2 Reading Registers 455

Section 16 Realtime Clock (RTC) 457
16.1 Overview 457
16.1.1 Features 457
16.1.2 Block Diagram 458
16.1.3 Pin Configuration 459
16.1.4 RTC Register Configuration 460
16.2 Register Descriptions 461
16.2.1 64-Hz Counter (R64CNT) 461
16.2.2 Second Counter (RSECCNT) 461
16.2.3 Minute Counter (RMINCNT) 462
16.2.4 Hour Counter (RHRCNT) 462
16.2.5 Day of the Week Counter (RWKCNT) 463
16.2.6 Date Counter (RDAYCNT) 464
16.2.7 Month Counter (RMONCNT) 464
16.2.8 Year Counter (RYRCNT) 465
16.2.9 Second Alarm Register (RSECAR) 465
16.2.10 Minute Alarm Register (RMINAR) 466
16.2.11 Hour Alarm Register (RHRAR) 466
16.2.12 Day of the Week Alarm Register (RWKAR) 467
16.2.13 Date Alarm Register (RDAYAR) 467

16.2.14 Month Alarm Register (RMONAR) 468
16.2.15 RTC Control Register 1 (RCR1) 469
16.2.16 RTC Control Register 2 (RCR2) 470
16.3 RTC Operation 473
16.3.1 Initial Settings of Registers after Power-On 473
16.3.2 Setting the Time 473
16.3.3 Reading the Time 475
16.3.4 Alarm Function 476
16.3.5 Crystal Oscillator Circuit 477
16.4 Usage Notes 478
16.4.1 Writing Registers During RTC Count Operation 478
16.4.2 RTC Periodic Interrupts 478
16.4.3 Using the ADJ Bit in the Real Time Clock (RTC) 479

Section 17 Serial Communication Interface (SCI) 481
17.1 Overview 481
17.1.1 Features 481
17.1.2 Block Diagram 482
17.1.3 Pin Configuration 485
17.1.4 Register Configuration 486
17.2 Register Descriptions 487
17.2.1 Receive Shift Register (SCRSR) 487
17.2.2 Receive Data Register (SCRDR) 487
17.2.3 Transmit Shift Register (SCTSR) 488
17.2.4 Transmit Data Register (SCTDR) 488
17.2.5 Serial Mode Register (SCSMR) 489
17.2.6 Serial Control Register (SCSCR) 491
17.2.7 Serial Status Register (SCSSR) 495
17.2.8 Port SC Control Register (SCPCR)/Port SC Data Register (SCPDR) 499
17.2.9 Bit Rate Register (SCBRR) 500
17.3 Operation 509
17.3.1 Overview 509
17.3.2 Operation in Asynchronous Mode 511
17.3.3 Multiprocessor Communication 521
17.3.4 Clock Synchronous Operation 529
17.4 SCI Interrupt Sources 538
17.5 Usage Notes 539

Section 18 Smart Card Interface 543
18.1 Overview 543
18.1.1 Features 543

18.1.2 Block Diagram 544
18.1.3 Pin Configuration 545
18.1.4 Register Configuration 545
18.2 Register Descriptions 546
18.2.1 Smart Card Mode Register (SCSCMR) 546
18.2.2 Serial Status Register (SCSSR) 547
18.3 Operation 548
18.3.1 Overview 548
18.3.2 Pin Connections 549
18.3.3 Data Format 549
18.3.4 Register Settings 551
18.3.5 Clock 552
18.3.6 Data Transmission and Reception 555
18.4 Usage Notes 561
18.4.1 Receive Data Timing and Receive Margin in Asynchronous Mode 561
18.4.2 Retransmission (Receive and Transmit Modes) 563

Section 19 Serial Communication Interface with FIFO (SCIF) 565
19.1 Overview 565
19.1.1 Features 565
19.1.2 Block Diagram 566
19.1.3 Pin Configuration 568
19.1.4 Register Configuration 569
19.2 Register Descriptions 570
19.2.1 Receive Shift Register 2 (SCRSR2) 570
19.2.2 Receive FIFO Data Register 2 (SCFRDR2) 570
19.2.3 Transmit Shift Register 2 (SCTSR2) 571
19.2.4 Transmit FIFO Data Register 2 (SCFTDR2) 571
19.2.5 Serial Mode Register 2 (SCSMR2) 572
19.2.6 Serial Control Register 2 (SCSCR2) 574
19.2.7 Serial Status Register 2 (SCSSR2) 576
19.2.8 Bit Rate Register 2 (SCBRR2) 580
19.2.9 FIFO Control Register 2 (SCFCR2) 586
19.2.10 FIFO Data Count Set Register 2 (SCFDR2) 588
19.3 Operation 589
19.3.1 Overview 589
19.3.2 Serial Operation 590
19.4 SCIF Interrupts 600
19.5 Usage Notes 601

Section 20 Serial IO (SIOF) 605
20.1 Overview 605
20.1.1 Features 605
20.1.2 Block Diagram 606
20.1.3 Terminal 607
20.1.4 Register Configuration 607
20.2 Register Description 608
20.2.1 Mode Register (SIMDR) 608
20.2.2 Clock Select Register (SISCR) 610
20.2.3 Transmit Data Assign Register (SITDAR) 612
20.2.4 Receive Data Assign Register (SIRDAR) 613
20.2.5 Control Command Assign Register (SICDAR) 614
20.2.6 Serial Control Register (SICTR) 616
20.2.7 FIFO Control Register (SIFCTR) 618
20.2.8 Status Register (SISTR) 620
20.2.9 Interrupt Enable Register (SIIER) 624
20.2.10 Transmit Data Register (SITDR) 626
20.2.11 Receive Data Register (SIRDR) 627
20.2.12 Transmit Control Data Register (SITCR) 628
20.2.13 Receive Control Data Register (SIRCR) 629
20.3 Operation 630
20.3.1 Serial Clock 630
20.3.2 Serial Timing 631
20.3.3 Transmit Data Format 632
20.3.4 Register Assignment for Transfer Data 634
20.3.5 Control Data Interface 637
20.3.6 FIFO 639
20.3.7 Procedures for Transmit or Receive 641
20.3.8 Interrupt 646
20.3.9 Transmit or Receive Timing 648
20.4 Usage Notes 652
20.4.1 Notes on Using the SIOF with Versions Previous to the SH7727B 654

Section 21 Analog Front End Interface (AFEIF) 657
21.1 Overview 657
21.1.1 Features 657
21.1.2 Block Diagram 658
21.1.3 Pin Configuration 659
21.1.4 Register Configuration 659
21.2 Register Description 660

21.2.1 AFEIF Control Register 1 and 2 (ACTR1, ACTR2) 660
21.2.2 Make Ratio Count Register (MRCR) 663
21.2.3 Minimum Pause Count Register (MPCR) 663
21.2.4 AFEIF Status Register 1 and 2 (ASTR1, ASTR2) 664
21.2.5 Dial Pulse Number Queue (DPNQ) 669
21.2.6 Ringing Pulse Counter (RCNT) 670
21.2.7 AFE Control Data Register (ACDR) 670
21.2.8 AFE Status Data Register (ASDR) 670
21.2.9 Transmit Data FIFO Port (TDFP) 671
21.2.10 Receive Data FIFO Port (RDFP) 671
21.3 Operation 671
21.3.1 Interrupt Timing 671
21.3.2 AFE Interface 673
21.3.3 DAA Interface 675
21.3.4 Wake up Ringing Interrupt 677

Section 22 USB Pin Multiplex Controller 679
22.1 Feature 679
22.1.1 Block Diagram 680
22.1.2 Pin Configuration 681
22.1.3 Register Configuration 682
22.2 Register Description 683
22.2.1 Extra Pin Function Controller (EXPFC) 683
22.3 Examples of External Circuit 684
22.3.1 Example of the Connection between USB Function Controller and Transceiver 684
22.3.2 Example of the Connection between USB Host Controller and Transceiver 689
22.3.3 Usage Notes 690

Section 23 USB Function Controller 691
23.1 Features 691
23.2 Block Diagram 692
23.3 Pin Configuration 692
23.4 Register Configuration 693
23.5 Register Descriptions 694
23.5.1 USBEP0i Data Register (USBEPDR0I) 694
23.5.2 USBEP0o Data Register (USBEPDR0O) 694
23.5.3 USBEP0s Data Register (USBEPDR0S) 694
23.5.4 USBEP1 Data Register (USBEPDR1) 695
23.5.5 USBEP2 Data Register (USBEPDR2) 695
23.5.6 USBEP3 Data Register (USBEPDR3) 695
23.5.7 USB Interrupt Flag Register 0 (USBIFR0) 695

23.5.8 USB Interrupt Flag Register 1 (USBIFR1) ... 697
23.5.9 USB Trigger Register (USBTRG) ... 698
23.5.10 USBFIFO Clear Register (USBFCLR) ... 699
23.5.11 USBEP0o Receive Data Size Register (USBEPSZ0O) ... 699
23.5.12 USB Data Status Register (USBDASTS) ... 700
23.5.13 USB Endpoint Stall Register (USBEPSTL) ... 700
23.5.14 USB Interrupt Enable Register 0 (USBIER0) ... 701
23.5.15 USB Interrupt Enable Register 1 (USBIER1) ... 701
23.5.16 USBEP1 Receive Data Size Register (USBEPSZ1) ... 701
23.5.17 USB Interrupt Select Register 0 (USBISR0) ... 702
23.5.18 USB Interrupt Select Register 1 (USBISR1) ... 702
23.5.19 USBDMA Setting Register (USBDMAR) ... 703
23.6 Operation ... 704
23.6.1 Cable Connection ... 704
23.6.2 Cable Disconnection ... 705
23.6.3 Control Transfer ... 706
23.6.4 EP1 Bulk-Out Transfer (Dual FIFOs) ... 713
23.6.5 EP2 Bulk-In Transfer (Dual FIFOs) ... 714
23.6.6 EP3 Interrupt-In Transfer ... 716
23.7 Processing of USB Standard Commands and Class/Vendor Commands ... 717
23.7.1 Processing of Commands Transmitted by Control Transfer ... 717
23.8 Stall Operations ... 718
23.8.1 Overview ... 718
23.8.2 Forcible Stall by Application ... 718
23.8.3 Automatic Stall by USB Function Module ... 720
23.9 Usage Notes ... 722
23.9.1 Receiving Setup Data ... 722
23.9.2 Clearing the FIFO ... 722
23.9.3 Overreading and Overwriting the Data Registers ... 722
23.9.4 Assigning Interrupt Sources to EP0 ... 723
23.9.5 Clearing the FIFO when DMA Transfer Is Enabled ... 723
23.9.6 Notes on TR Interrupt ... 723
23.9.7 Peripheral Clock (Pϕ) Operation Frequency ... 724

Section 24 USB HOST Module ... 725
24.1 General Description ... 725
24.1.1 Features ... 725
24.1.2 Pin Configuration ... 726
24.1.3 Register Configuration ... 727
24.2 Register Description ... 728
24.2.1 HcRevision ... 728

24.2.2 HcControl ... 729
24.2.3 HcCommandStatus ... 732
24.2.4 HcInterruptStatus ... 735
24.2.5 HcInterruptEnable ... 737
24.2.6 HcInterruptDisable ... 739
24.2.7 HcHCCA ... 740
24.2.8 HcPeriodCurrentED ... 740
24.2.9 HcControlHeadED ... 741
24.2.10 HcControlCurrentED ... 741
24.2.11 HcBulkHeadED ... 741
24.2.12 HcBulkCurrentED ... 742
24.2.13 HcDoneHead ... 742
24.2.14 HcFmInterval ... 743
24.2.15 HcFmRemaining ... 744
24.2.16 HcFmNumber ... 745
24.2.17 HcPeriodicStart ... 746
24.2.18 HcLSThreshold ... 746
24.2.19 HcRhDescriptorA ... 747
24.2.20 HcRhDescriptorB ... 749
24.2.21 HcRhStatus ... 750
24.2.22 HcRhPortStatus[1:2] ... 751
24.3 Data Storage Format which Required by USB Host Controller ... 757
24.3.1 Storage Format of the Transferred Data ... 757
24.3.2 Storage Format of the Descriptor ... 758
24.4 Data Alignment Restriction of USB Host Controller ... 758
24.4.1 Restriction on the Line Boundary of the Synchronous DRAM ... 758
24.4.2 Restriction on the Memory Access Address ... 759
24.5 Restrictions on the Data Transfer of USB Controller ... 759
24.5.1 Restriction of the Data Size in IN Transfer ... 759
24.5.2 Restrictions on the Hub Connection on NAK/STALL Reception ... 759
24.5.3 Restrictions when a Low-Speed Device is Disconnected ... 760
24.6 Restrictions on the Software Reset and USB Reset ... 760
24.7 Notes on Using USB Host with Versions Previous to the SH7727C ... 760

Section 25 LCD Controller ... 763
25.1 Overview ... 763
25.1.1 Features ... 763
25.1.2 Block Diagram ... 764
25.1.3 Pin Configuration ... 765
25.1.4 Register Configuration ... 765
25.2 Register Descriptions ... 767

25.2.1 LCDC Input Clock Register (LDICKR) 767
25.2.2 LCDC Module Type Register (LDMTR) 768
25.2.3 LCDC Data Format Register (LDDFR) 771
25.2.4 LCDC Scan Mode Register (LDSMR) 773
25.2.5 LCDC Start Address Register for Upper Display Data Fetch (LDSARU) 774
25.2.6 LCDC Start Address Register for Lower Display Data Fetch (LDSARL) 775
25.2.7 LCDC Line Address Offset Register for Display Data Fetch (LDLAOR) 776
25.2.8 LCDC Palette Control Register (LDPALCR) 777
25.2.9 Palette Data Registers 00 to FF (LDPR00 to LDPRFF) 778
25.2.10 LCDC Horizontal Character Number Register (LDHCNR) 779
25.2.11 LCDC Horizontal Sync Signal Register (LDHSYNR) 780
25.2.12 LCDC Vertical Display Line Number Register (LDVDLNR) 781
25.2.13 LCDC Vertical Total Line Number Register (LDVTLNR) 782
25.2.14 LCDC Vertical Sync Signal Register (LDVSYNR) 783
25.2.15 LCDC AC Modulation Signal Toggle Line Number Register (LDACLNR) 784
25.2.16 LCDC Interrupt Control Register (LDINTR) 784
25.2.17 LCDC Power Management Mode Register (LDPMMR) 786
25.2.18 LCDC Power-Supply Sequence Period Register (LDPSPR) 788
25.2.19 LCDC Control Register (LDCNTR) 789
25.3 Operation 790
25.3.1 LCD Module Sizes which can be Displayed in this LCDC 790
25.3.2 Limits on the Resolution of Rotated Displays, Burst Length, and Connected Memory (SDRAM) 791
25.3.3 Color Palette Specification 797
25.3.4 Data Format 799
25.3.5 Timing Controller Register 802
25.3.6 Power Management Registers 802
25.3.7 Operation for Hardware Rotation 807
25.4 Clock and LCD Data Signal Examples 809
25.5 Usage Notes 820

Section 26 Pin Function Controller (PFC) 821
26.1 Overview 821
26.2 Register Configuration 826
26.3 Register Descriptions 827
26.3.1 Port A Control Register (PACR) 827
26.3.2 Port B Control Register (PBCR) 828
26.3.3 Port C Control Register (PCCR) 829
26.3.4 Port D Control Register (PDCR) 830
26.3.5 Port E Control Register (PECR) 831
26.3.6 Port F Control Register (PFCR) 832

26.3.7 Port G Control Register (PGCR) ... 833
26.3.8 Port H Control Register (PHCR) ... 835
26.3.9 Port J Control Register (PJCR) ... 836
26.3.10 Port K Control Register (PKCR) ... 837
26.3.11 Port L Control Register (PLCR) ... 838
26.3.12 Port M Control Register (PMCR) ... 839
26.3.13 SC Port Control Register (SCPCR) ... 840

Section 27 I/O Ports ... 845
27.1 Overview ... 845
27.2 Register Configuration ... 846
27.3 Ports A to C, E, J, K ... 847
27.3.1 Ports A to C, E, J, K Data Rgister (PADR, PBDR, PCDR, PEDR, PJDR, PKDR) ... 847
27.4 Port D ... 848
27.4.1 Port D Data Register (PDDR) ... 848
27.5 Ports F, M ... 850
27.5.1 Ports F, M Data Register (PFDR, PMDR) ... 850
27.6 Port G ... 851
27.6.1 Port G Data Register (PGDR) ... 851
27.7 Port H ... 852
27.7.1 Port H Data Register (PHDR) ... 852
27.8 Port L ... 854
27.8.1 Port L Data Register (PLDR) ... 854
27.9 SC Port ... 855
27.9.1 Port SC Data Register (SCPDR) ... 855

Section 28 A/D Converter ... 857
28.1 Overview ... 857
28.1.1 Features ... 857
28.1.2 Block Diagram ... 858
28.1.3 Input Pins ... 859
28.1.4 Register Configuration ... 860
28.2 Register Descriptions ... 861
28.2.1 A/D Data Registers A to D (ADDRA to ADDRD) ... 861
28.2.2 A/D Control/Status Register (ADCSR) ... 862
28.2.3 A/D Control Register (ADCR) ... 864
28.3 Bus Master Interface ... 865
28.4 Operation ... 867
28.4.1 Single Mode (MULTI = 0) ... 867
28.4.2 Multi Mode (MULTI = 1, SCN = 0) ... 869

28.4.3 Scan Mode (MULTI = 1, SCN = 1) ... 871
28.4.4 Input Sampling and A/D Conversion Time ... 873
28.4.5 External Trigger Input Timing ... 874
28.5 Interrupts ... 875
28.6 Definitions of A/D Conversion Accuracy ... 875
28.7 Usage Notes ... 876
28.7.1 Setting Analog Input Voltage ... 876
28.7.2 Processing of Analog Input Pins ... 876
28.7.3 Access Size and Read Data ... 877

Section 29 D/A Converter ... 879
29.1 Overview ... 879
29.1.1 Features ... 879
29.1.2 Block Diagram ... 879
29.1.3 I/O Pins ... 880
29.1.4 Register Configuration ... 880
29.2 Register Descriptions ... 881
29.2.1 D/A Data Registers 0 and 1 (DADR0/1) ... 881
29.2.2 D/A Control Register (DACR) ... 881
29.3 Operation ... 883

Section 30 PC Card Controller (PCC) ... 885
30.1 Overview ... 885
30.1.1 Features ... 885
30.1.2 Block Diagram ... 886
30.1.3 Register Configuration ... 887
30.1.4 PCMCIA Support ... 888
30.2 Register Descriptions ... 891
30.2.1 Area 6 Interface Status Register (PCC0ISR) ... 891
30.2.2 Area 6 General Control Register (PCC0GCR) ... 894
30.2.3 Area 6 Card Status Change Register (PCC0CSCR) ... 896
30.2.4 Area 6 Card Status Change Interrupt Enable Register (PCC0CSCIER) ... 900
30.3 Operation ... 903
30.3.1 PC card Connection Specification (Interface Diagram, Pin Correspondence) ... 903
30.3.2 PC Card Interface Timing ... 907
30.3.3 Usage Notes ... 912

Section 31 User-Debugging Interface (H-UDI) ... 915
31.1 Overview ... 915
31.2 User Debugging Interface (H-UDI) ... 915
31.2.1 Pin Description ... 915

31.2.2 Block Diagram 916
31.3 Register Descriptions 916
31.3.1 Bypass Register (SDBPR) 917
31.3.2 Instruction Register (SDIR) 917
31.3.3 Boundary-Scan Register (SDBSR) 918
31.4 H-UDI Operations 925
31.4.1 TAP Controller 925
31.4.2 Reset Configuration 926
31.4.3 H-UDI Reset 927
31.4.4 H-UDI Interrupt 927
31.4.5 Bypass 927
31.4.6 Using H-UDI to Recover from Sleep Mode 927
31.5 Usage Notes 928
31.6 Advanced User Debugger (AUD) 928

Section 32 Electrical Characteristics 929
32.1 Absolute Maximum Ratings 929
32.2 DC Characteristics 931
32.3 AC Characteristics 934
32.3.1 Clock Timing 937
32.3.2 Control Signal Timing 946
32.3.3 AC Bus Timing 949
32.3.4 Basic Timing 951
32.3.5 Burst ROM Timing 954
32.3.6 Synchronous DRAM Timing 957
32.3.7 PCMCIA Timing 968
32.3.8 Peripheral Module Signal Timing 975
32.3.9 H-UDI-Related Pin Timing 978
32.3.10 LCDC Timing 980
32.3.11 SIOF Module Signal Timing 982
32.3.12 USB Module Signal Timing 985
32.3.13 AFEIF Module Signal Timing 987
32.3.14 AC Characteristics Measurement Conditions 989
32.3.15 Delay Time Variation Due to Load Capacitance 990
32.4 A/D Converter Characteristics 991
32.5 D/A Converter Characteristics 991

Appendix A Pin Functions 993
A.1 Pin Functions 993
A.2 Treatment of Unused Pins 999
A.3 Pin Status when Accessing Address Spaces 1004

Appendix B Control Registers 1018
B.1 Register Address Map 1018

Appendix C Product Lineup 1028

Appendix D Package Dimensions 1029

Appendix E Using Versions Previous to the SH7727C 1031
E.1 Determining the Version Number Based on the Markings on the Chip 1031

Appendix F Using Port G Control Register (PGCR) with Versions Previous to the SH7727B 1032

Figures

Section 1 Overview and Pin Functions

Figure 1.1 Block Diagram 8
Figure 1.2 Pin Arrangement (PRQP0240KC-B) 9
Figure 1.3 Pin Arrangement (PLBG0240JA-A) 10

Section 2 CPU

Figure 2.1 Register Configuration in Each Processing Mode (1) 23
Figure 2.2 Register Configuration in Each Processing Mode (2) 24
Figure 2.3 General Purpose Register (Not in DSP Mode) 25
Figure 2.4 General Purpose Register (DSP Mode) 26
Figure 2.5 Control Registers (1) 29
Figure 2.5 Control Registers (2) 30
Figure 2.6 System Registers 31
Figure 2.7 DSP Registers 35
Figure 2.8 Connections of DSP Registers and Buses 37
Figure 2.9 Longword Operand 38
Figure 2.10 Data Format 39
Figure 2.11 Byte, Word, and Longword Alignment 40
Figure 2.12 X and Y Data Transfer Addressing 49
Figure 2.13 Single Data Transfer Addressing 50
Figure 2.14 Modulo Addressing 52
Figure 2.15 DSP Instruction Formats 58
Figure 2.16 Sample Parallel Instruction Program 87
Figure 2.17 Examples of Conditional Operations and Data Transfer Instructions 95

Section 3 Memory Management Unit (MMU)

Figure 3.1 MMU Functions 99
Figure 3.2 Logical Address Space Mapping 101
Figure 3.3 MMU Register Contents 104
Figure 3.4 Overall Configuration of the TLB 105
Figure 3.5 Logical Address and TLB Structure 106
Figure 3.6 TLB Indexing (IX = 1) 107
Figure 3.7 TLB Indexing (IX = 0) 108
Figure 3.8 Objects of Address Comparison 109
Figure 3.9 Operation of LDTLB Instruction 113
Figure 3.10 Synonym Problem 115
Figure 3.11 MMU Exception Generation Flowchart 120
Figure 3.12 MMU Exception Signals in Instruction Fetch 121

Figure 3.13 MMU Exception Signals in Data Access 122
Figure 3.14 MMU Exception in Repeat Loop 123
Figure 3.15 Specifying Address and Data for Memory-Mapped TLB Access 126

Section 4 Exception Handling
Figure 4.1 Vector Table 132
Figure 4.2 Example of Acceptance Order of General Exceptions 135
Figure 4.3 Bit Configurations of EXPEVT, INTEVT, INTEVT2, and TRA Registers 139

Section 5 Cache
Figure 5.1 Cache Structure 150
Figure 5.2 CCR Register Configuration 152
Figure 5.3 CCR2 Register Configuration 153
Figure 5.4 Cache Search Scheme 155
Figure 5.5 Write-Back Buffer Configuration 157
Figure 5.6 Specifying Address and Data for Memory-Mapped Cache Access 159

Section 6 X/Y Memory
Figure 6.1 X/Y Memory Logical Address Mapping 163
Figure 6.2 X/Y Memory Physical Address Mapping 164

Section 7 Interrupt Controller (INTC)
Figure 7.1 INTC Block Diagram 166
Figure 7.2 Example of IRL Interrupt Connection 171
Figure 7.3 Interrupt Operation Flowchart 198
Figure 7.4 Example of Pipeline Operations when IRL Interrupt is Accepted 202

Section 8 User Break Controller
Figure 8.1 Block Diagram of User Break Controller 204

Section 9 Power-Down Modes and Software Reset
Figure 9.1 Canceling Standby Mode with STBCR.STBY 246
Figure 9.2 Power-On Reset (Clock Modes 0, 1, 2, and 7) STATUS Output 249
Figure 9.3 Manual Reset STATUS Output 250
Figure 9.4 Standby to Interrupt STATUS Output 250
Figure 9.5 Standby to Power-On Reset STATUS Output 251
Figure 9.6 Standby to Manual Reset STATUS Output 251
Figure 9.7 Sleep to Interrupt STATUS Output 252
Figure 9.8 Sleep to Power-On Reset STATUS Output 252
Figure 9.9 Sleep to Manual Reset STATUS Output 253
Figure 9.10 Hardware Standby Mode Timing (CA = Low in Normal Operation) 255

Figure 9.11 Hardware Standby Mode Timing (CA = Low during WDT Operation while Standby Mode is Cleared) 256

Section 10 On-Chip Oscillation Circuits

Figure 10.1 Block Diagram of Clock Pulse Generator 259
Figure 10.2 Block Diagram of the WDT 271
Figure 10.3 Writing to WTCNT and WTCSR 275
Figure 10.4 Points for Attention when Using Crystal Resonator 277
Figure 10.5 Points for Attention when Using PLL Oscillator Circuit 278

Section 11 Extend Clock Pulse Generator for USB (EXCPG)

Figure 11.1 Block Diagram of EXCPG 279

Section 12 Bus State Controller (BSC)

Figure 12.1 Corresponding to Logical Address Space and Physical Address Space 285
Figure 12.2 Corresponding to Logical Address Space and Physical Address Space 289
Figure 12.3 Physical Space Allocation 291
Figure 12.4 Writing to RFCR, RTCSR, RTCNT, and RTCOR 315
Figure 12.5 Basic Timing of Basic Interface 326
Figure 12.6 Example of 32-Bit Data-Width Static RAM Connection 327
Figure 12.7 Example of 16-Bit Data-Width Static RAM Connection 328
Figure 12.8 Example of 8-Bit Data-Width Static RAM Connection 328
Figure 12.9 Basic Interface Wait Timing (Software Wait Only) 329
Figure 12.10 Basic Interface Wait State Timing (Wait State Insertion by $\overline{\text{WAIT}}$ Signal WAITSEL = 1) 330
Figure 12.11 Example of 64-Mbit Synchronous DRAM Connection (32-Bit Bus Width) 332
Figure 12.12 Example of 64-Mbit Synchronous DRAM (16-Bit Bus Width) 333
Figure 12.13 Basic Timing for Synchronous DRAM Burst Read 336
Figure 12.14 Synchronous DRAM Burst Read Wait Specification Timing 337
Figure 12.15 Basic Timing for Synchronous DRAM Single Read 338
Figure 12.16 Basic Timing for Synchronous DRAM Burst Write 339
Figure 12.17 Basic Timing for Synchronous DRAM Single Write 341
Figure 12.18 Auto-Refresh Operation 342
Figure 12.19 Synchronous DRAM Auto-Refresh Timing 343
Figure 12.20 Synchronous DRAM Self-Refresh Timing 344
Figure 12.21 Synchronous DRAM Mode Write Timing 346
Figure 12.22 Burst ROM Wait Access Timing 348
Figure 12.23 Burst ROM Basic Access Timing 349
Figure 12.24 Example of PCMCIA Interface (If Internal PC Card Controller is not used.) 351
Figure 12.25 Basic Timing for PCMCIA Memory Card Interface 353
Figure 12.26 Wait Timing for PCMCIA Memory Card Interface 354

Figure 12.27 Basic Timing for PCMCIA Memory Card Interface Burst Access 355
Figure 12.28 Wait Timing for PCMCIA Memory Card Interface Burst Access 356
Figure 12.29 PCMCIA Space Assignment 357
Figure 12.30 Basic Timing for PCMCIA I/O Card Interface 359
Figure 12.31 Wait Timing for PCMCIA I/O Card Interface 360
Figure 12.32 Dynamic Bus Sizing Timing for PCMCIA I/O Card Interface 361
Figure 12.33 Waits between Access Cycles 363
Figure 12.34 Pins A25 to A0 Pull-Up Timing 364
Figure 12.35 Pins D31 to D0 Pull-Up Timing (Read Cycle) 364
Figure 12.36 Pins D31 to D0 Pull-Up Timing (Write Cycle) 365

Section 13 Li Bus State Controller (LBSC)
Figure 13.1 Block Diagram of Li Bus Architecture 377

Section 14 Direct Memory Access Controller (DMAC)
Figure 14.1 DMAC Block Diagram 381
Figure 14.2 DMAC Transfer Flowchart 399
Figure 14.3 Operation in Round-Robin Mode 403
Figure 14.4 Channel Priority Order in Round-Robin Mode 404
Figure 14.5 Operation in Direct Address Mode 406
Figure 14.6 Example of DMA Transfer Timing in the Direct Address Mode (Transfer Source: Ordinary Memory, Transfer Destination: Ordinary Memory) 407
Figure 14.7 Example of DMA Transfer Timing in the Direct Address Mode (16-byte Transfer, Transfer Source: Ordinary Memory, Transfer Destination: Ordinary Memory) 408
Figure 14.8 Example of DMA Transfer Timing in the Direct Address Mode (16-byte Transfer, Transfer Source: Synchronous DRAM, Transfer Destination: Ordinary Memory) 408
Figure 14.9 Operation in Indirect Address Mode (When the External Memory Space is Set to 16-bit Width) 410
Figure 14.10 Example of Transfer Timing in Indirect Address Mode (Transfer between External Memories, External Memory with 16-bit Width) 411
Figure 14.11 Data Flow in Single Address Mode 412
Figure 14.12 Example of DMA Transfer Timing in Single Address Mode 413
Figure 14.13 Example of DMA Transfer Timing in Single Address Mode (External Memory Space (Ordinary Memory) → External Device with DACK) 414
Figure 14.14 Transfer Example in Cycle-Steal Mode 415
Figure 14.15 Example of Transfer in Burst Mode 415
Figure 14.16 Bus State in Multiple Channel Operation 417
Figure 14.17 Cycle-Steal Mode, Level Input (CPU Access: 2 Cycles) 420
Figure 14.18 Cycle-Steal Mode, Level Input (CPU Access: 3 Cycles) 421

Figure 14.19 Cycle-Steal Mode, Level input (CPU Access: 2 Cycles, DMA RD Access: 4 Cycles) ... 422
Figure 14.20 Cycle-Steal Mode, Level input (CPU Access: 2 Cycles, $\overline{\text{DREQ}}$ Input Delayed) ... 423
Figure 14.21 Cycle-Steal Mode, Edge input (CPU Access: 2 Cycles) ... 424
Figure 14.22 Burst Mode, Level Input ... 425
Figure 14.23 Burst Mode, Edge Input ... 426
Figure 14.24 Source Address Reload Function Diagram ... 427
Figure 14.25 Timing Chart of Source Address Reload Function ... 428
Figure 14.26 CMT Block Diagram ... 431
Figure 14.27 Counter Operation ... 435
Figure 14.28 Count Timing ... 436
Figure 14.29 Timing of CMF Setting ... 437
Figure 14.30 Timing of CMF Clear by the CPU ... 437

Section 15 Timer (TMU)
Figure 15.1 TMU Block Diagram ... 444
Figure 15.2 Setting the Count Operation ... 451
Figure 15.3 Auto-Reload Counter Operation ... 452
Figure 15.4 Count Timing when Internal Clock is Operating ... 452
Figure 15.5 Count Timing when On-Chip RTC Clock is Operating ... 453
Figure 15.6 UNF Set Timing ... 453
Figure 15.7 Status Flag Clear Timing ... 454

Section 16 Realtime Clock (RTC)
Figure 16.1 RTC Block Diagram ... 458
Figure 16.2(a) Setting the Time ... 474
Figure 16.2(b) Setting the Time ... 474
Figure 16.3 Reading the Time ... 475
Figure 16.4 Using the Alarm Function ... 476
Figure 16.5 Example of Crystal Oscillator Circuit Connection ... 477
Figure 16.6 Periodic Interrupt Function Setting ... 478

Section 17 Serial Communication Interface (SCI)
Figure 17.1 SCI Block Diagram ... 482
Figure 17.2 SCPT[1]/SCK0 Pin ... 483
Figure 17.3 SCPT[0]/TxD0 Pin ... 484
Figure 17.4 SCPT[0]/RxD0 Pin ... 485
Figure 17.5 Data Format in Asynchronous Communication (Example with 8-Bit Data, Parity, Two Stop Bits) ... 511
Figure 17.6 Relationship between Output Clock and Transfer Data Phase (Asynchronous Mode) ... 513

Figure 17.7 Sample SCI Initialization Flowchart 514
Figure 17.8 Sample Serial Transmission Flowchart 515
Figure 17.9 Example of Operation in Transmission in Asynchronous Mode
(Example with 8-Bit Data, Parity, One Stop Bit) 517
Figure 17.10 Sample Serial Reception Data Flowchart (1) 518
Figure 17.10 Sample Serial Reception Data Flowchart (2) 519
Figure 17.11 Example of SCI Operation in Reception
(Example with 8-Bit Data, Parity, One Stop Bit) 521
Figure 17.12 Example of Communication Using Multiprocessor Format
(Transmission of Data H'AA to Receiving Station A) 522
Figure 17.13 Sample Multiprocessor Serial Transmission Flowchart 523
Figure 17.14 Example of SCI Operation in Transmission
(Example with 8-Bit Data, Multiprocessor Bit, One Stop Bit) 525
Figure 17.15 Sample Multiprocessor Serial Reception Flowchart (1) 526
Figure 17.15 Sample Multiprocessor Serial Reception Flowchart (2) 527
Figure 17.16 Example of SCI Operation in Reception
(Example with 8-Bit Data, Multiprocessor Bit, One Stop Bit) 528
Figure 17.17 Data Format in Clock Synchronous Communication 529
Figure 17.18 Sample SCI Initialization Flowchart 531
Figure 17.19 Sample Serial Transmission Flowchart 532
Figure 17.20 Sample SCI Transmission Operation in Clocked Synchronous Mode 533
Figure 17.21 Sample Serial Reception Flowchart (1) 534
Figure 17.21 Sample Serial Reception Flowchart (2) 535
Figure 17.22 Example of SCI Operation in Reception 536
Figure 17.23 Sample Flowchart of Simultaneous Serial Transmit and Receive Operations 537
Figure 17.24 Receive Data Sampling Timing in Asynchronous Mode 540

Section 18 Smart Card Interface

Figure 18.1 Smart Card Interface Block Diagram 544
Figure 18.2 Pin Connection Diagram for the Smart Card Interface 549
Figure 18.3 Data Format for Smart Card Interface 550
Figure 18.4 Waveform of Start Character 552
Figure 18.5 Initialization Flowchart (Example) 556
Figure 18.6 Transmission Flowchart (Example) 558
Figure 18.7 Reception Flowchart (Example) 560
Figure 18.8 Receive Data Sampling Timing in Smart Card Mode 562
Figure 18.9 Retransmission in SCI Receive Mode 563
Figure 18.10 Retransmission in SCI Transmit Mode 564

Section 19 Serial Communication Interface with FIFO (SCIF)

Figure 19.1 SCIF Block Diagram 566

Figure 19.2 SCPT[4]/TxD2 Pin 567
Figure 19.3 SCPT[4]/RxD2 Pin 568
Figure 19.4 Sample SCIF Initialization Flowchart 592
Figure 19.5 Sample Serial Transmission Flowchart 593
Figure 19.6 Example of Transmit Operation (Example with 8-Bit Data, Parity, One Stop Bit) 595
Figure 19.7 Example of Operation Using Modem Control ($\overline{\text{CTS}}$) 595
Figure 19.8 Sample Serial Reception Flowchart (1) 596
Figure 19.9 Sample Serial Reception Flowchart (2) 597
Figure 19.10 Example of SCIF Receive Operation
(Example with 8-Bit Data, Parity, One Stop Bit) 599
Figure 19.11 Example of Operation Using Modem Control ($\overline{\text{RTS}}$) 599
Figure 19.12 Receive Data Sampling Timing in Asynchronous Mode 602

Section 20 Serial IO (SIOF)

Figure 20.1 SIOF Block Diagram 606
Figure 20.2 Serial Clock Supply System 630
Figure 20.3 SIOF Serial Data Synchronized Timing 631
Figure 20.4 SIOF Transmit or Receive Timing 632
Figure 20.5 Transmit or Receive Data Bit Alignment 634
Figure 20.6 Control Data Bit Alignment 636
Figure 20.7 Control Data Interface (Slot Position) 637
Figure 20.8 Control Data Interface (Secondary FS) 638
Figure 20.9 Example of Transmit Operation in Master 641
Figure 20.10 Example of Receive Operation in Master 642
Figure 20.11 Example of Transmit Operation in Slave 643
Figure 20.12 Example of Receive Operation in Slave 644
Figure 20.13 Transmit or Receive Timing (8 bits monaural—1) 648
Figure 20.14 Transmit or Receive Timing (8 bits monaural—2) 648
Figure 20.15 Transmit or Receive Timing (16 bits monaural—1) 649
Figure 20.16 Transmit or Receive Timing (16 bits stereo—1) 649
Figure 20.17 Transmit or Receive Timing (16 bits stereo—2) 650
Figure 20.18 Transmit or Receive Timing (16 bits stereo—3) 650
Figure 20.19 Transmit or Receive Timing (16 bits monaural—2) 651

Section 21 Analog Front End Interface (AFEIF)

Figure 21.1 Block Diagram of AFE Interface 658
Figure 21.2 FIFO Interrupt Timing 672
Figure 21.3 Ringing Interrupt Occurrence Timing 672
Figure 21.4 Interrupt Generator 673
Figure 21.5 AFE Serial Interface 673
Figure 21.6 AFE Control Sequence 674

Figure 21.7 DAA Block Diagram 675
Figure 21.8 Ringing Detect Sequence 676

Section 22 USB Pin Multiplex Controller
Figure 22.1 Block Diagram of USB PIN Multiplexer 680
Figure 22.2 Example 1 of Transceiver Connection for USB function Controller (On-chip transceiver is used) 684
Figure 22.3 Example 2 of Transceiver Connection for USB function Controller (On-chip transceiver is used) 685
Figure 22.4 Example 3 of Transceiver Connection for USB function Controller (On-chip transceiver is not used) 686
Figure 22.5 Example 4 of Transceiver Connection for USB function Controller (On-chip transceiver is not used) 687
Figure 22.6 Example 1 of Transceiver Connection for USB Host Controller (On-chip transceiver is used) 689
Figure 22.7 Example 2 of Transceiver Connection for USB Host Controller (On-chip transceiver is not used) 690

Section 23 USB Function Controller
Figure 23.1 Block Diagram of UBC 692
Figure 23.2 Cable Connection Operation 704
Figure 23.3 Cable Disconnection Operation 705
Figure 23.4 Transfer Stage for Control Transfer 706
Figure 23.5 Setup Stage Operation 707
Figure 23.6 Data Stage Operation (Control-In) 708
Figure 23.7 Data Stage Operation (Control-Out) 710
Figure 23.8 Status Stage Operation (Control-In) 711
Figure 23.9 Status Stage Operation (Control-Out) 712
Figure 23.10 EP1 Bulk-Out Transfer Operation 713
Figure 23.11 EP2 Bulk-In Transfer Operation 714
Figure 23.12 EP2 Interrupt-In Transfer Operation 716
Figure 23.13 Forcible Stall by Application 719
Figure 23.14 Automatic Stall by USB Function Module 721
Figure 23.15 TR Interrupt Flag Set Timing 723

Section 25 LCD Controller
Figure 25.1 Block Diagram 764
Figure 25.2 Valid Display and Retrace Period 786
Figure 25.3 Color-Palette Data Format 798
Figure 25.4 Power-Supply Control Sequence and States of the LCD Module 803
Figure 25.5 Power-Supply Control Sequence and States of the LCD Module 803

Figure 25.6 Power-Supply Control Sequence and States of the LCD Module 804
Figure 25.7 Power-Supply Control Sequence and States of the LCD Module 804
Figure 25.8 Clock and LCD Data Signal Example 809
Figure 25.9 Clock and LCD Data Signal Example 810
Figure 25.10 Clock and LCD Data Signal Example 810
Figure 25.11 Clock and LCD Data Signal Example 811
Figure 25.12 Clock and LCD Data Signal Example 811
Figure 25.13 Clock and LCD Data Signal Example 812
Figure 25.14 Clock and LCD Data Signal Example 812
Figure 25.15 Clock and LCD Data Signal Example 813
Figure 25.16 Clock and LCD Data Signal Example 813
Figure 25.17 Clock and LCD Data Signal Example 814
Figure 25.18 Clock and LCD Data Signal Example 815
Figure 25.19 Clock and LCD Data Signal Example 816
Figure 25.20 Clock and LCD Data Signal Example 817
Figure 25.21 Clock and LCD Data Signal Example 818
Figure 25.22 Clock and LCD Data Signal Example 819

Section 26 Pin Function Controller (PFC)
Figure 26.1 Overview of the Pin Selection Function 821

Section 28 A/D Converter
Figure 28.1 A/D Converter Block Diagram 858
Figure 28.2 A/D Data Register Access Operation (Reading H'AA40) 866
Figure 28.3 Example of A/D Converter Operation (Single Mode, Channel 2 Selected) 868
Figure 28.4 Example of A/D Converter Operation (Multi Mode, Channels AN4 to AN6 Selected) 870
Figure 28.5 Example of A/D Converter Operation (Scan Mode, Channels AN4 to AN6 Selected) 872
Figure 28.6 A/D Conversion Timing 873
Figure 28.7 External Trigger Input Timing 874
Figure 28.8 Definitions of A/D Conversion Accuracy 876
Figure 28.9 Example of Analog Input Protection Circuit 877
Figure 28.10 Analog Input Pin Equivalent Circuit 877

Section 29 D/A Converter
Figure 29.1 D/A Converter Block Diagram 879
Figure 29.2 Example of D/A Converter Operation 883

Section 30 PC Card Controller (PCC)
Figure 30.1 PC Card Controller Block Diagram 886

Figure 30.2 Continuous 32-MB Area Mode 889
Figure 30.3 Continuous 16-MB Area Mode (Area 6) 890
Figure 30.4 SH7727 Interface 903
Figure 30.5 PCMCIA Memory Card Interface Basic Timing 907
Figure 30.6 PCMCIA Memory Card Interface Wait Timing 908
Figure 30.7 PCMCIA I/O Card Interface Basic Timing 909
Figure 30.8 PCMCIA I/O Card Interface Wait Timing 910
Figure 30.9 Dynamic Bus Sizing Timing for PCMCIA I/O Card Interface 911

Section 31 User-Debugging Interface (H-UDI)

Figure 31.1 H-UDI Block Diagram 916
Figure 31.2 TAP Controller State Transitions 925
Figure 31.3 H-UDI Reset 927

Section 32 Electrical Characteristics

Figure 32.1 Power-On Sequence 930
Figure 32.2 Power Supply Voltage and Operating Frequency 935
Figure 32.3 EXTAL Clock Input Timing 941
Figure 32.4 CKIO Clock Input Timing 941
Figure 32.5 CKIO Clock Output Timing 942
Figure 32.6 Power-on Oscillation Settling Time 942
Figure 32.7 Oscillation Settling Time at Standby Return (Return by Reset) 943
Figure 32.8 Oscillation Settling Time at Standby Return (Return by NMI) 943
Figure 32.9 Oscillation Settling Time at Standby Return (Return by IRQ4 to IRQ0) 944
Figure 32.10 PLL Synchronization Settling Time by Reset or NMI Interrupt 944
Figure 32.11 PLL Synchronization Settling Time by IRQ/IRL and PINT0/1 Interrupt 945
Figure 32.12 PLL Sync Stabilization Time at Frequency Multiplier Factor Change 945
Figure 32.13 Reset Input Timing 947
Figure 32.14 Interrupt signal Input Timing 947
Figure 32.15 Bus Release Timing 948
Figure 32.16 Pin Drive Timing at Standby 948
Figure 32.17 Basic Bus Cycle (No Wait) 951
Figure 32.18 Basic Bus Cycle (One Wait) 952
Figure 32.19 Basic Bus Cycle (External Wait, WAITSEL = 1) 953
Figure 32.20 Burst ROM Bus Cycle (No Wait) 954
Figure 32.21 Burst ROM Bus Cycle (Two Waits) 955
Figure 32.22 Burst ROM Bus Cycle (External Wait, WAITSEL = 1) 956
Figure 32.23 Synchronous DRAM Read Bus Cycle (RCD = 0, CAS Latency = 1, TPC = 0) 957
Figure 32.24 Synchronous DRAM Read Bus Cycle (RCD = 2, CAS Latency = 2, TPC = 1) 958
Figure 32.25 Synchronous DRAM Read Bus Cycle (Burst Read (Single Read × 4),
RCD = 0, CAS Latency = 1, TPC = 1) 959

Figure 32.26 Synchronous DRAM Read Bus Cycle (Burst Read (Single Read × 4), RCD = 1, CAS Latency = 3, TPC = 0) 960
Figure 32.27 Synchronous DRAM Write Bus Cycle (RCD = 0, TPC = 0, TRWL = 0) 961
Figure 32.28 Synchronous DRAM Write Bus Cycle (RCD = 2, TPC = 1, TRWL = 1) 962
Figure 32.29 Synchronous DRAM Write Bus Cycle (Burst Mode (Single Write × 4), RCD = 0, TPC = 1, TRWL = 0) 963
Figure 32.30 Synchronous DRAM Write Bus Cycle (Burst Mode (Single Write × 4), RCD = 1, TPC = 0, TRWL = 0) 964
Figure 32.31 Synchronous DRAM Auto-Refresh Cycle (TRAS = 1, TPC = 1) 965
Figure 32.32 Synchronous DRAM Self-Refresh Cycle (TPC = 0) 966
Figure 32.33 Synchronous DRAM Mode Register Write Cycle 967
Figure 32.34 PCMCIA Memory Bus Cycle (TED = 0, TEH = 0, No Wait) 968
Figure 32.35 PCMCIA Memory Bus Cycle (TED = 2, TEH = 1, One Wait, External Wait, WAITSEL = 1) 969
Figure 32.36 PCMCIA Memory Bus Cycle (Burst Read, TED = 0, TEH = 0, No Wait) 970
Figure 32.37 PCMCIA Memory Bus Cycle (Burst Read, TED = 1, TEH = 1, Two Waits, Burst Pitch = 3, WAITSEL = 1) 971
Figure 32.38 PCMCIA I/O Bus Cycle (TED = 0, TEH = 0, No Wait) 972
Figure 32.39 PCMCIA I/O Bus Cycle (TED = 2, TEH = 1, One Wait, External Wait, WAITSEL = 1) 973
Figure 32.40 PCMCIA I/O Bus Cycle (TED = 1, TEH = 1, One Wait, Bus Sizing, WAITSEL = 1) 974
Figure 32.41 Oscillation Settling Time at RTC Crystal Oscillator Power-on 976
Figure 32.42 SCK Input Clock Timing 976
Figure 32.43 SCI I/O Timing in Clock Synchronous Mode 976
Figure 32.44 I/O Port Timing 977
Figure 32.45 $\overline{\text{DREQ}}$ Input Timing 977
Figure 32.46 DRAK Output Timing 977
Figure 32.47 TCK Input Timing 978
Figure 32.48 $\overline{\text{TRST}}$ Input Timing (Reset Hold) 979
Figure 32.49 H-UDI Data Transfer Timing 979
Figure 32.50 $\overline{\text{ASEMD0}}$ Input Timing 979
Figure 32.51 LCDC AC Specification 981
Figure 32.52 SIOMCLK Input Timing 982
Figure 32.53 SIOF Transmit/Receive Timing (Master Mode 1: Fall Sampling Time) 983
Figure 32.54 SIOF Transmit/Receive Timing (Master Mode 1: Rise Sampling Time) 983
Figure 32.55 SIOF Transmit/Receive Timing (Master Mode 2: Fall Sampling Time) 984
Figure 32.56 SIOF Transmit/Receive Timing (Master Mode 2: Rise Sampling Time) 984
Figure 32.57 SIOF Transmit/Receive Timing (Slave Mode 1 and Slave Mode 2) 985
Figure 32.58 USB Clock Timing 985
Figure 32.59 AFEIF Module AC Timing 988

Figure 32.60 Output Load Circuit 989
Figure 32.61 Load Capacitance vs. Delay Time 990

Appendix D Package Dimensions

Figure D.1 Package Dimensions (PRQP0240KC-B) 1029
Figure D.2 Package Dimensions (PLBG0240JA-A) 1030

Tables

Section 1 Overview and Pin Functions

Table 1.1 SH7727 Features 2
Table 1.2 SH7727 Pin Function 11

Section 2 CPU

Table 2.1 Initial Register Values 24
Table 2.2 Detail Behavior Under Each SH3-DSP Mode 33
Table 2.3 Destination Register of DSP Instructions 34
Table 2.4 Source Register of DSP Operations 35
Table 2.5 DSR Register Bits 36
Table 2.6 Word Data Sign Extension 41
Table 2.7 Delayed Branch Instructions 41
Table 2.8 T Bit 42
Table 2.9 Immediate Data Referencing 42
Table 2.10 Absolute Address Referencing 43
Table 2.11 Displacement Referencing 43
Table 2.12 Addressing Modes and Effective Addresses for CPU Instructions 44
Table 2.13 Overview of Data Transfer Instructions 48
Table 2.14 CPU Instruction Formats 55
Table 2.15 Double Data Transfer Instruction Formats 59
Table 2.16 Single Data Transfer Instruction Formats 60
Table 2.17 A-Field Parallel Data Transfer Instructions 61
Table 2.18 B-Field ALU Operation Instructions and Multiply Instructions 62
Table 2.19 CPU Instruction Types 65
Table 2.20 Data Transfer Instructions 69
Table 2.21 Arithmetic Operation Instructions 71
Table 2.22 Logic Operation Instructions 73
Table 2.23 Shift Instructions 74
Table 2.24 Branch Instructions 75
Table 2.25 System Control Instructions 76
Table 2.26 Added CPU System Control Instructions 81
Table 2.27 Double Data Transfer Instructions 83
Table 2.28 Single Data Transfer Instructions 84
Table 2.29 Correspondence between DSP Data Transfer Operands and Registers 85
Table 2.30 DSP Operation Instruction Formats 86
Table 2.31 Correspondence between DSP Instruction Operands and Registers 87
Table 2.32 DSP Operation Instructions 88
Table 2.33 DC Bit Update Definitions 94

Table 2.34 Examples of NOPX and NOPY Instruction Codes 96

Section 3 Memory Management Unit (MMU)
Table 3.1 Register Configuration 103
Table 3.2 Access States Designated by D, C, and PR Bits 110

Section 4 Exception Handling
Table 4.1 Register Configuration 131
Table 4.2 Exception Event Vectors 133
Table 4.3 Exception Codes 136
Table 4.4 Types of Reset 141

Section 5 Cache
Table 5.1 Cache Specifications 149
Table 5.2 LRU and Way Replacement 151
Table 5.3 Register Configuration 151
Table 5.4 LRU and Way Replacement (when W2LOCK=1) 153
Table 5.5 LRU and Way Replacement (when W3LOCK=1) 153
Table 5.6 LRU and Way Replacement (when W2LOCK=1 and W3LOCK=1) 154

Section 6 X/Y Memory
Table 6.1 X/Y Memory Specifications 161

Section 7 Interrupt Controller (INTC)
Table 7.1 Pin Configuration 167
Table 7.2 Register Configuration 168
Table 7.3 $\overline{\text{IRL3}}$ to $\overline{\text{IRL0}}$ Pins and Interrupt Levels 171
Table 7.4 Interrupt Exception Handling Sources and Priority (IRQ Mode) 174
Table 7.5 Interrupt Exception Handling Sources and Priority (IRL Mode) 176
Table 7.6 Interrupt Level and INTEVT Code 178
Table 7.7 Interrupt Request Sources and IPRA to IPRG 179
Table 7.8 Interrupt Response Time 200

Section 8 User Break Controller
Table 8.1 Register Configuration 205
Table 8.2 Data Access Cycle Addresses and Operand Size Comparison Conditions 223

Section 9 Power-Down Modes and Software Reset
Table 9.1 Power-Down Modes 234
Table 9.2 Pin Configuration 235
Table 9.3 Register Configuration 235

Table 9.4 Register States in Standby Mode ... 244

Section 10 On-Chip Oscillation Circuits
Table 10.1 Clock Pulse Generator Pins and Functions ... 261
Table 10.2 Register Configuration ... 261
Table 10.3 Clock Operating Modes ... 262
Table 10.4 Available Combination of Clock Mode and FRQCR Values ... 264
Table 10.5 Register Configuration ... 272

Section 11 Extend Clock Pulse Generator for USB (EXCPG)
Table 11.1 Pin Configuration ... 280
Table 11.2 Register Configuration ... 280

Section 12 Bus State Controller (BSC)
Table 12.1 Pin Configuration ... 286
Table 12.2 Register Configuration ... 288
Table 12.3 Physical Address Space Map ... 290
Table 12.4 Correspondence between External Pins (MD4 and MD3) and Memory bus width in area0 ... 291
Table 12.5 SH7727 and PCMCIA Pins ... 292
Table 12.6 32-Bit External Device/Big Endian Access and Data Alignment ... 316
Table 12.7 16-Bit External Device/Big Endian Access and Data Alignment ... 317
Table 12.8 8-Bit External Device/Big Endian Access and Data Alignment ... 318
Table 12.9 32-Bit External Device/Little Endian Access and Data Alignment ... 319
Table 12.10 16-Bit External Device/Little Endian Access and Data Alignment ... 320
Table 12.11 8-Bit External Device/Little Endian Access and Data Alignment ... 321
Table 12.12 Relationship between Synchronous DRAM type, bus width and AMX ... 334
Table 12.13 Relationship between LSI Address Pins and Synchronous DRAM Address Pins ... 335

Section 13 Li Bus State Controller (LBSC)
Table 13.1 Register Configuration ... 367

Section 14 Direct Memory Access Controller (DMAC)
Table 14.1 Pin Configuration ... 382
Table 14.2 DMAC Registers ... 382
Table 14.3 Selecting External Request Modes with the RS Bits ... 400
Table 14.4 Selection of On-Chip Module Request Modes Using RS3 to RS0 Bits ... 401
Table 14.5 DMA Transfers ... 405
Table 14.6 Relationship of Request Modes and Bus Modes ... 416
Table 14.7 Register Configuration ... 432

Table 14.8 Transfer Conditions and Register Settings for Transfer between On-Chip A/D Converter and External Memory 438
Table 14.9 DMAC Sate after the Fourth Transfer Ends 439
Table 14.10 Transfer Conditions and Register Settings for Transfer between External Memory and SCIF Transmitter 440

Section 15 Timer (TMU)
Table 15.1 TMU Register Configuration 445
Table 15.2 TMU Interrupt Sources 454

Section 16 Realtime Clock (RTC)
Table 16.1 RTC Pin Configuration 459
Table 16.2 RTC Registers 460
Table 16.3 Day-of-Week Codes (RWKCNT) 463
Table 16.4 Day-of-Week Codes (RWKAR) 467
Table 16.5 Recommended Oscillator Circuit Constants (Recommended Values) 477

Section 17 Serial Communication Interface (SCI)
Table 17.1 SCI Pins 485
Table 17.2 Registers 486
Table 17.3 SCSMR Settings 501
Table 17.4 Bit Rates and SCBRR Settings in Asynchronous Mode (1) 502
Table 17.4 Bit Rates and SCBRR Settings in Asynchronous Mode (2) 502
Table 17.4 Bit Rates and SCBRR Settings in Asynchronous Mode (3) 503
Table 17.4 Bit Rates and SCBRR Settings in Asynchronous Mode (4) 503
Table 17.4 Bit Rates and SCBRR Settings in Asynchronous Mode (5) 504
Table 17.4 Bit Rates and SCBRR Settings in Asynchronous Mode (6) 504
Table 17.4 Bit Rates and SCBRR Settings in Asynchronous Mode (7) 505
Table 17.5 Bit Rates and SCBRR Settings in Clock Synchronous Mode 506
Table 17.6 Maximum Bit Rates for Various Frequencies with Baud Rate Generator (Asynchronous Mode) 507
Table 17.7 Maximum Bit Rates during External Clock Input (Asynchronous Mode) 508
Table 17.8 Maximum Bit Rates during External Clock Input (Clock Synchronous Mode) 508
Table 17.9 Serial Mode Register Settings and SCI Communication Formats 510
Table 17.10 SCSMR and SCSCR Settings and SCI Clock Source Selection 510
Table 17.11 Serial Communication Formats (Asynchronous Mode) 512
Table 17.12 Receive Error Conditions and SCI Operation 520
Table 17.13 SCI Interrupt Sources 538
Table 17.14 SCSSR Status Flags and Transfer of Receive Data 539

Section 18 Smart Card Interface

Table 18.1 SCI Pins 545
Table 18.2 Registers 545
Table 18.3 Register Settings for the Smart Card Interface 551
Table 18.4 Relationship of n to CKS1 and CKS0 553
Table 18.5 Examples of Bit Rate B (Bit/s) for SCBRR Settings (n = 0) 553
Table 18.6 Examples of SCBRR Settings for Bit Rate B (Bit/s) (n = 0) 553
Table 18.7 Maximum Bit Rates for Frequencies (Smart Card Interface Mode) 554
Table 18.8 Register Set Values and SCK Pin 554
Table 18.9 Smart Card Mode Operating State and Interrupt Sources 561

Section 19 Serial Communication Interface with FIFO (SCIF)

Table 19.1 SCIF Pins 568
Table 19.2 Registers 569
Table 19.3 SCSMR2 Settings 581
Table 19.4 Bit Rates and SCBRR2 Settings 581
Table 19.5 Maximum Bit Rates for Various Frequencies with Baud Rate Generator (Asynchronous Mode) 585
Table 19.6 SCSMR2 Settings and SCIF Transmit/Receive 589
Table 19.7 Settings for SCSMR2 and SCSCR2 and Selection of Clock Source of SCIF 590
Table 19.8 Serial Transmit/Receive Formats 590
Table 19.9 SCIF Interrupt Sources 600

Section 20 Serial IO (SIOF)

Table 20.1 SIOF Pin List 607
Table 20.2 SIOF Register Configuration 607
Table 20.3 Examples of SIOF Clock Frequency 630
Table 20.4 Serial Transmit Mode 632
Table 20.5 Frame Length 633
Table 20.6 Transmit Data Sound Mode 635
Table 20.7 Receive Data Sound Mode 635
Table 20.8 Control Data Channel Number Establishment 636
Table 20.9 Transmit Request Submit Condition 639
Table 20.10 Receive Request Submit Condition 640
Table 20.11 Transmit or Receive Reset 645
Table 20.12 SIOF Interrupt Factors 646
Table 20.13 Setting Conditions for the Transmit or Receive Interrupt Flag 647

Section 21 Analog Front End Interface (AFEIF)

Table 21.1 Pins for AFE Interface 659

Table 21.2 AFEIF Registers 659
Table 21.3 Telephone Number and Data 669

Section 22 USB Pin Multiplex Controller
Table 22.1 Pin Configuration (Digital Transceiver Signal) 681
Table 22.2 Pin Configuration (Analog Transceiver Signal) 681
Table 22.3 Pin Configuration (Power Control signal) 682
Table 22.4 Register Configuration 682

Section 23 USB Function Controller
Table 23.1 Pin Configuration and Functions 692
Table 23.2 USB Function Module Registers 693
Table 23.3 Command Decoding on Application Side 717

Section 24 USB HOST Module
Table 24.1 Pin Configuration 726
Table 24.2 Register Configuration 727

Section 25 LCD Controller
Table 25.1 Pin Configuration 765
Table 25.2 Register Configuration 765
Table 25.3 Limits on the Resolution of Rotated Displays, Burst Length, and Connected Memory (32-bit Bus SDRAM) 792
Table 25.4 Limits on the Resolution of Rotated Displays, Burst Length, and Connected Memory (16-bit Bus SDRAM) 795
Table 25.5 Available Power-Supply Control-Sequence Periods at Typical Frame Rates 805
Table 25.6 LCDC Operating Modes 806
Table 25.7 LCD Module Power-Supply States 806

Section 26 Pin Function Controller (PFC)
Table 26.1 List of Multiplexed Pins 821
Table 26.2 Pin Function Controller Registers 826

Section 27 I/O Ports
Table 27.1 Pin Function Controller Registers 846
Table 27.2 Read/Write Operation of the Ports A to C, E, J, K Data Register 847
Table 27.3 Read/Write Operation of the Port D Data Register (PDDR) 849
Table 27.4 Read/Write Operation of the Ports F, M Data Register (PFDR, PMDR) 850
Table 27.5 Read/Write Operation of the Port G Data Register (PGDR) 851
Table 27.6 Read/Write Operation of the Port H Data Register (PHDR) 853
Table 27.7 Read/Write Operation of the Port L Data Register (PLDR) 854

Table 27.8 Read/Write Operation of the SC Port Data Register (SCPDR) 856

Section 28 A/D Converter
Table 28.1 A/D Converter Pins 859
Table 28.2 A/D Converter Registers 860
Table 28.3 Analog Input Channels and A/D Data Registers 861
Table 28.4 A/D Conversion Time (Single Mode) 874
Table 28.5 Analog Input Pin Ratings 878
Table 28.6 Relationship between Access Size and Read Data 878

Section 29 D/A Converter
Table 29.1 D/A Converter Pins 880
Table 29.2 D/A Converter Registers 880

Section 30 PC Card Controller (PCC)
Table 30.1 PC Card Controller Registers 887
Table 30.2 Features of the PCMCIA Interface 888
Table 30.3 PCMCIA Support Interface 904

Section 31 User-Debugging Interface (H-UDI)
Table 31.1 H-UDI Registers 917
Table 31.2 H-UDI Commands 918
Table 31.3 Correspondence between SH7727 Pins and Boundary-Scan Register 919
Table 31.4 Reset Configuration 926

Section 32 Electrical Characteristics
Table 32.1 Absolute Maximum Ratings 929
Table 32.2 DC Characteristics (1) 931
Table 32.2 DC Characteristics (2) 933
Table 32.3 Permitted Output Current Values 934
Table 32.4 Maximum Operating Frequencies (1) 936
Table 32.4 Maximum Operating Frequencies (2) 936
Table 32.5 Clock Timing (1) 937
Table 32.6 Clock Timing (2) 938
Table 32.7 Clock Timing (3) 939
Table 32.8 Clock Timing (4) 940
Table 32.9 Control Signal Timing 946
Table 32.10 Bus Timing 949
Table 32.11 Peripheral Module Signal Timing 975
Table 32.12 H-UDI-Related Pin Timing 978
Table 32.13 LCDC Timing 980

Table 32.14 SIOF Module Signal Timing 982
Table 32.15 USB Module Signal Timing 985
Table 32.16 USB Electrical Characteristics (Full-Speed) 986
Table 32.17 USB Electrical Characteristics (Low-Speed) 986
Table 32.18 AFEIF Module Signal Timing 987
Table 32.19 A/D Converter Characteristics 991
Table 32.20 D/A Converter Characteristics 991

Appendix A Pin Functions

Table A.1 Pin Functions 993
Table A.2 Treatment of Unused Pins 999
Table A.3 Pin Status (Normal Memory/Little Endian) 1004
Table A.4 Pin Status (Normal Memory/Big Endian) 1006
Table A.5 Pin Status (Burst ROM/Little Endian) 1008
Table A.6 Pin Status (Burst ROM/Big Endian) 1010
Table A.7 Pin Status (Synchronous DRAM/Little Endian) 1012
Table A.8 Pin Status (Synchronous DRAM/Big Endian) 1013
Table A.9 Pin Status (PCMCIA/Little Endian) 1014
Table A.10 Pin Status (PCMCIA/Big Endian) 1016

Appendix B Control Registers

Table B.1 Memory-Mapped Control Registers (Address Map) 1018

Section 1 Overview and Pin Functions

1.1 Features

The SH7727 is a single-chip RISC microprocessor that integrates a 32-bit RISC-type SuperH RISC engine architecture CPU with digital signal processing (DSP) extension as its core that has a cache memory, an on-chip X/Y memory, and a memory management unit (MMU) as well as peripheral functions required for system configuration. The SH7727 includes data protection, virtual memory, and other functions provided by incorporating an MMU into a SuperH Series microprocessor (SH-1 or SH-2).

The SH7727 chip has the on-chip X/Y memory with large capacitance, on-chip DSP module, and emulator support. The provision of on-chip DSP functions enables applications that previously required the use of two chips—a microprocessor and a DSP—to be implemented with a single chip.

High-speed data transfers with a direct memory access controller (DMAC) and an external memory access support function enables direct connection to each memory. The SH7727 microprocessor also supports an infrared communication function, a stereo audio recording and playback function, a USB host controller, a function controller, an LCD controller, a PCMCIA interface, an A/D converter, and a D/A converter.

The USB host controller and LCD controller have bus master functions, so that data supplied from an external memory (area 3) can be freely processed. Because the USB host controller, in particular, conforms to Open HCI standards, it is extremely easy to transfer data from the PC of a device driver or other devices. Also, low-power operation suitable for battery operation is possible because the LCD controller continues to display even in sleep mode.

An internal USB transceiver is also provided, eliminating the need for attachments.

A powerful built-in power management function keeps power consumption low, even during high-speed operation. In particular, power consumption can be significantly reduced by halting the X/Y memory. Because the LSI operates at a maximum speed eight times of the speed at which the system operates, it is ideal for electronic devices, which require both high speed and low power consumption.

The features of this LSI are listed in table 1.1. The specifications of this LSI are listed in table 1.2.

Table 1.1 SH7727 Features

Item	Features
CPU	• Original Renesas SuperH architecture • Object code level compatible with SH-1, SH-2 and SH-3 • 32-bit internal data bus • General-register — Sixteen 32-bit general registers (eight 32-bit shadow registers) — Eight 32-bit control registers — Four 32-bit system registers • RISC-type instruction set — Instruction length: 16-bit fixed length to improve code efficiency — Load-store architecture — Delayed branch instructions — Instruction set based on C language • Instruction execution time: one instruction/cycle for basic instructions • Logical address space: 4 Gbytes • Space identifier ASID: 8 bits, 256 logical address space • Five-stage pipeline

Item	Features
DSP	• Mixture of 16-bit and 32-bit instructions • 32-/40-bit internal data bus • Multiplier, ALU, barrel shifter and DSP register • 16 bits x 16 bits → 32-bit one cycle multiplier • Large DSP data register — Six 32-bit data registers — Two 40-bit data registers • Extended Harvard Architecture for DSP data bus — Two data buses — One instruction bus • Max. four parallel operations: ALU, multiply and two load or store • Two addressing units to generate addresses for two memory access • DSP data addressing modes: increment, indexing (with or without modulo addressing) • Zero overhead repeat loop control • Conditional execution instructions • User-DSP mode and privileged-DSP mode
Clock pulse generator (CPG)	• Clock mode: An input clock can be selected from the external input (EXTAL or CKIO) or crystal oscillator. • Three types of clocks generated: — CPU clock: 1–16 times the input clock — Bus clock: 1–4 times the input clock — Peripheral clock: 1/4–4 times the input clock • Power-down modes: — Sleep mode — Standby mode — Module standby mode (X/Y memory standby enabled) • One-channel watchdog timer

Item	Features
Memory management unit (MMU)	• 4 Gbytes of address space, 256 address spaces (ASID 8 bits) • Page unit sharing • Supports multiple page sizes: 1 kbytes or 4 kbytes • 128-entry, 4-way set associative TLB • Supports software selection of replacement method and random-replacement algorithms • Contents of TLB can directly be accessed according to the address mapping
Cache memory	• 16-kbyte cache, mixed instruction/data • 256 entries, 4-way set associative, 16-byte block length • Write-back, write-through, least recently used (LRU) replacement algorithm • 1-stage write-back buffer • Maximum 2 ways of the cache can be locked
X/Y memory	• User-selectable mapping mechanism — Fixed mapping for mission-critical realtime applications — Automatic mapping through TLB for easy to use • 3 independent read/write ports — 8-/16-/32-bit access from the CPU — Maximum two 16-bit accesses from the DSP — 8-/16-/32-bit access from the DMAC • 8-kbyte RAM for X and Y memory individually
Interrupt controller (INTC)	• 7 external interrupt pins (NMI, IRQ5–IRQ0) • On-chip peripheral interrupts: set priority levels for each module
User break controller (UBC)	• 2 break channels • Addresses, data values, type of access, and data size can all be set as break conditions • Supports a sequential break function

Item	Features
Bus state controller (BSC)	• Physical address space divided into six areas (area 0, areas 2 to 6), each of up to 64 Mbytes, with the following features settable for each area: — Bus size (8, 16, or 32 bits) — Number of wait cycles (hardware wait function also waited) — Direct connection of SRAM, synchronous DRAM, and burst ROM possible by designating memory to be connected to each area — Supports PCMCIA interface (2 channels) — Chip select signals (CS0, CS2–CS6) for relevant area • Synchronous DRAM refresh function — Programmable refresh interval — Supports CAS-before-RAS refresh and self-refresh modes — Supports power-down DRAM • Synchronous DRAM burst access function • Big endian or little endian can be specified
Li bus state controller (LBSC)	• Bus State Controller for LCDC or USB Host • Supports synchronous DRAM • Synchronous DRAM access function (area 3)
User debug Interface (H-UDI)	• E10A emulator support • Pin arrangement conforming to JTAG specification • Realtime branch trace
Timer (TMU)	• 3-channel auto-reload-type 32-bit timer • Choice of six counter input clocks • Maximum resolution: 2 MHz
Realtime clock (RTC)	• Built-in clock, calendar functions, and alarm functions • On-chip 32-kHz crystal oscillator circuit with a maximum resolution (cycle interrupt) of 1/256 second
Serial communi-cation interface (SCI)	• Asynchronous mode or clock synchronous mode can be selected • Full-duplex communication • Supports smart card interface

Item	Features
Serial I/O (SIOF)	• Synchronous 16 step, 8/16/32 bit word FIFO for transmission/reception • Supports 8-bit/16-bit mono/stereo sound playback or recording • DMA can be transferred • Supports frame sync signal
Serial communication interface (SCIF)	• 16-byte FIFO for transmission/reception • DMA can be transferred • On-chip modem control function
Direct memory access controller (DMAC)	• 4 channels • Burst mode and cycle-steal mode • External request operating mode • 1-channel 16-bit compare match timer
PC card controller	• Supports control signals for one slot • Interchangeable with SH7709 when not in use (2 slots)
USB Host controller (USBH)	• Conforms to OHCI Rev. 1.0 • USB Rev. 1.1 compatible • Up to 127 endpoints • Supports INT/BULK/CONTROL/ISO modes • Bus master controller (can access area 3 synchronous DRAM) • 2 ports with analog transceiver (1of 2 is common with USB function) • External clock input function
USB Function controller (USBF)	• USB Rev. 1.1 compatible • Up to 4 endpoints • Supports INT/BULK/CONTROL modes (ISO mode not supported) • 1 port with analog transceiver (common with Host), 12 Mbps only • External clock input function

Item	Features
LCD controller (LCDC)	• From 16 x 1 to 1024 x 1024 pixels can be supported • 4/8/15/16 bpp (bit per pixel) color modes • 1/2/4/6 bpp (bit per pixel) gray scale • 8-bit Frame rate controller • TFT/DSTN/STN • Signal polarity setting function • Hardware panel rotation • Power control function • Selectable clock source (LCLK, bus clock (Bϕ), or peripheral clock (Pϕ))
AFE I/F	• ST7550 direct interface • Telephone line control • 128-word FIFO for transfer • 128-word FIFO for receive
I/O port	• Thirteen 8-bit I/O ports
A/D converter (ADC)	• 10 bits ± 4 LSB, 6 channels • Conversion time: 15 µs • Input range: 0–Vcc (max. 3.6 V)
D/A converter (DAC)	• 8 bits ± 4 LSB, 2 channels • Conversion time: 10 µs • Output range: 0–Vcc (max. 3.6 V)

Product lineup

As of June 1, 2005

SH7727	Power Supply Voltage I/O	Power Supply Voltage Internal	Operating Frequency	Model Name	Package
160 MHz products	3.0 V to 3.6 V	1.70 V to 2.05 V	160 MHz	HD6417727F160C	240-pin plastic HQFP (PRQP0240KC-B)
				HD6417727BP160C	240-pin CSP (PLBG0240JA-A)
100 MHz products	2.6 V to 3.6 V	1.60 V to 2.05 V	100 MHz	HD6417727F100C	240-pin plastic HQFP (PRQP0240KC-B)
				HD6417727BP100C	240-pin CSP (PLBG0240JA-A)

1.2 Block Diagram

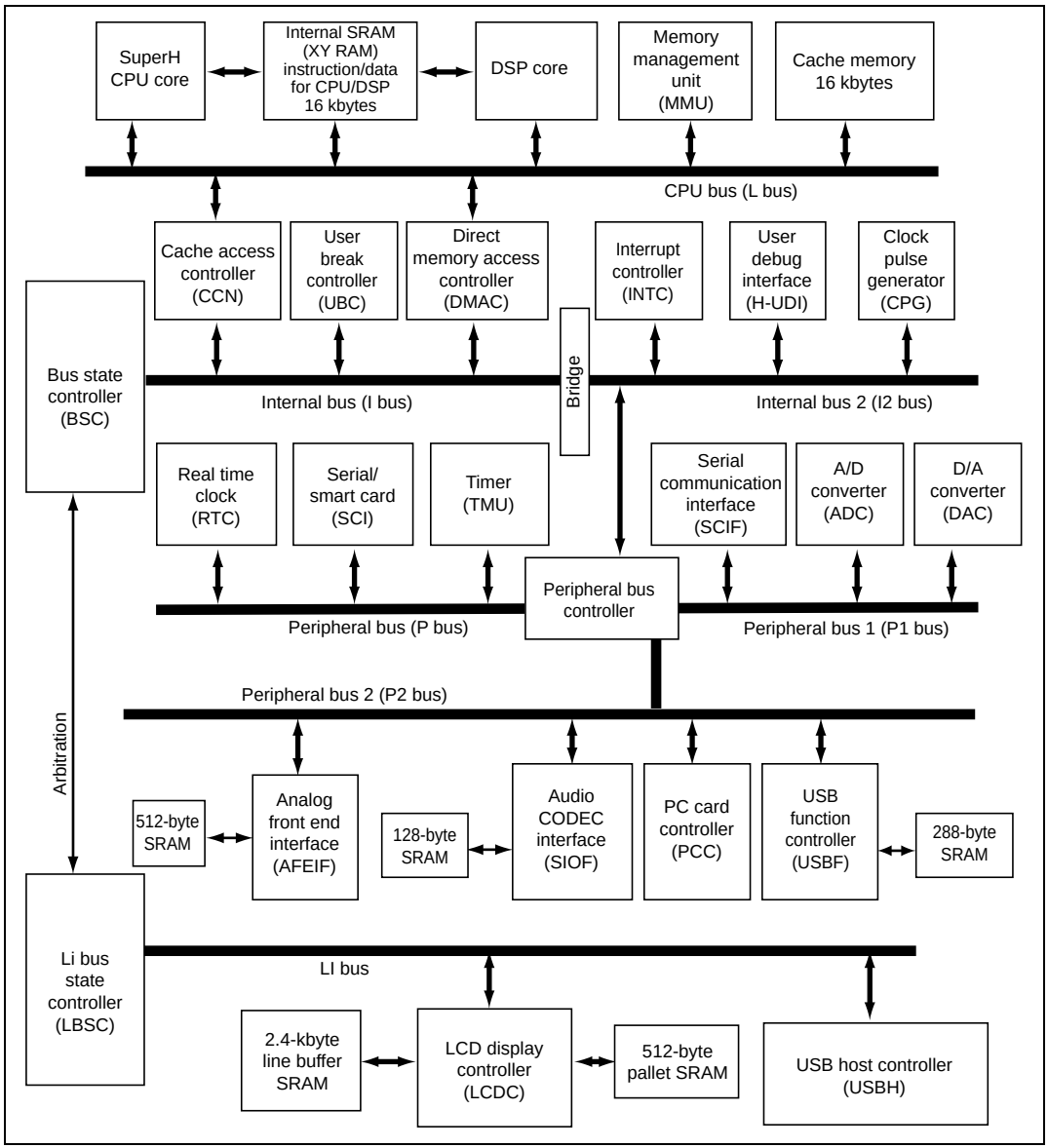


Figure 1.1 Block Diagram

1.3 Pin Description

1.3.1 Pin Arrangement

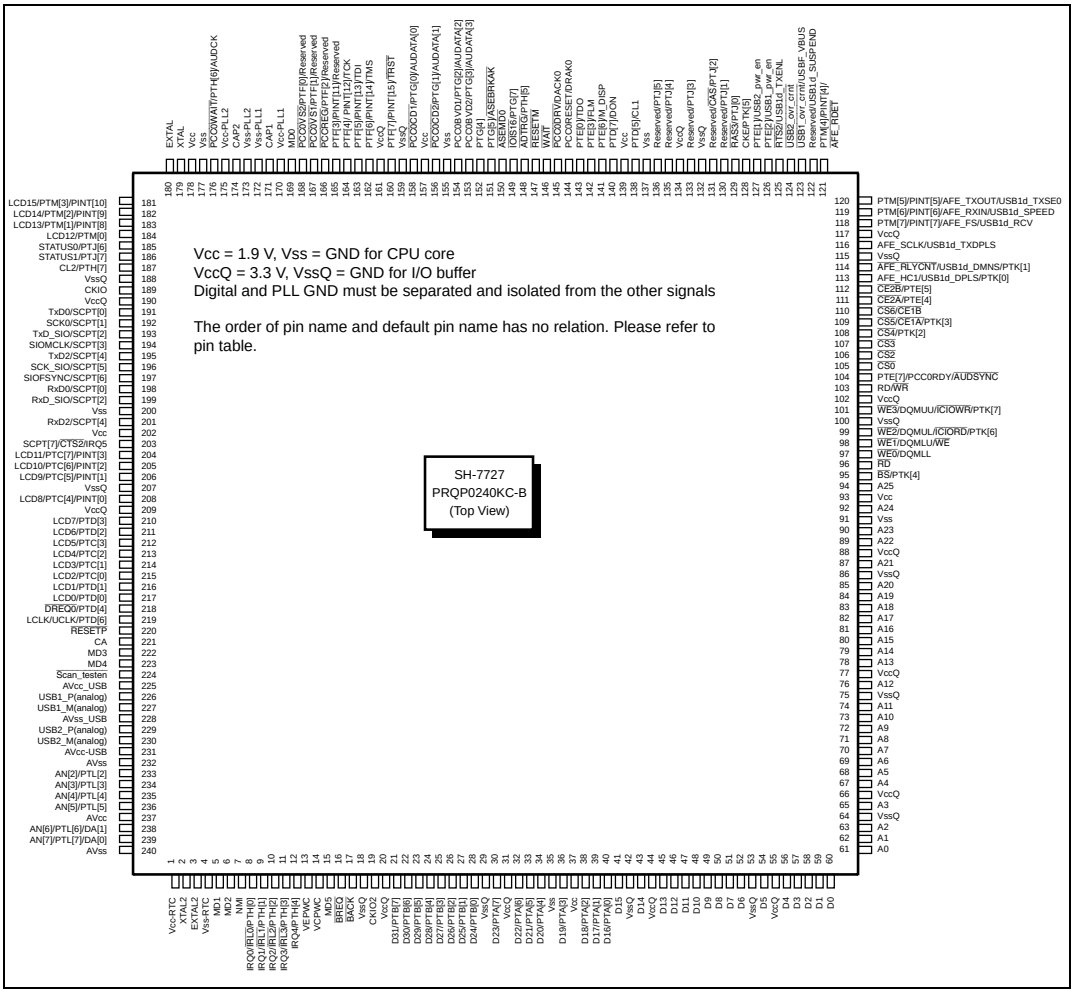
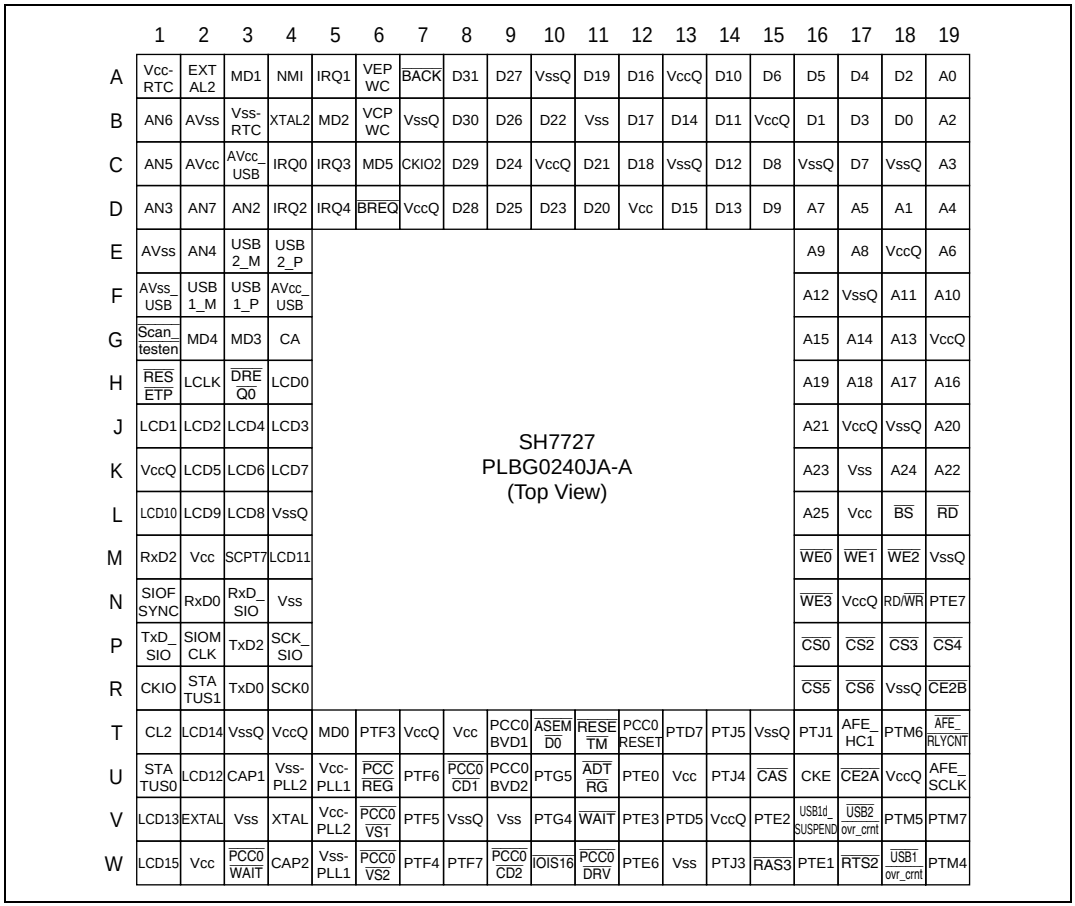


Figure 1.2 Pin Arrangement (PRQP0240KC-B)



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
A	Vcc-RTC	EXTAL2	MD1	NMI	IRQ1	VEPWC	BACK	D31	D27	VssQ	D19	D16	VccQ	D10	D6	D5	D4	D2	A0
B	AN6	AVss	Vss-RTC	XTAL2	MD2	VCPWC	VssQ	D30	D26	D22	Vss	D17	D14	D11	VccQ	D1	D3	D0	A2
C	AN5	AVcc	AVcc_USB	IRQ0	IRQ3	MD5	CKIO2	D29	D24	VccQ	D21	D18	VssQ	D12	D8	VssQ	D7	VssQ	A3
D	AN3	AN7	AN2	IRQ2	IRQ4	BREQ	VccQ	D28	D25	D23	D20	Vcc	D15	D13	D9	A7	A5	A1	A4
E	AVss	AN4	USB2_M	USB2_P												A9	A8	VccQ	A6
F	AVss_USB	USB1_M	USB1_P	AVcc_USB												A12	VssQ	A11	A10
G	Scan_testen	MD4	MD3	CA												A15	A14	A13	VccQ
H	RES ETP	LCLK	DRE Q0	LCD0												A19	A18	A17	A16
J	LCD1	LCD2	LCD4	LCD3												A21	VccQ	VssQ	A20
K	VccQ	LCD5	LCD6	LCD7												A23	Vss	A24	A22
L	LCD10	LCD9	LCD8	VssQ												A25	Vcc	BS	RD
M	RxD2	Vcc	SCPT7	LCD11												WE0	WE1	WE2	VssQ
N	SIOF SYNC	RxD0	RxD_SIO	Vss												WE3	VccQ	RD/WR	PTE7
P	TxD_SIO	SIOM CLK	TxD2	SCK_SIO												CS0	CS2	CS3	CS4
R	CKIO	STATUS1	TxD0	SCK0												CS5	CS6	VssQ	CE2B
T	CL2	LCD14	VssQ	VccQ	MD0	PTF3	VccQ	Vcc	PCC0 BVD1	ASEM D0	RESE TM	PCC0 RESET	PTD7	PTJ5	VssQ	PTJ1	AFE_HC1	PTM6	AFE_RLYCNT
U	STATUS0	LCD12	CAP1	Vss-PLL2	Vcc-PLL1	PCC REG	PTF6	PCC0 CD1	PCC0 BVD2	PTG5	ADT RG	PTE0	Vcc	PTJ4	CAS	CKE	CE2A	VccQ	AFE_SCLK
V	LCD13	EXTAL	Vss	XTAL	Vcc-PLL2	PCC0 VS1	PTF5	VssQ	Vss	PTG4	WAIT	PTE3	PTD5	VccQ	PTE2	USB1d_SUSPEND	USB2 ovr_crnt	PTM5	PTM7
W	LCD15	Vcc	PCC0 WAIT	CAP2	Vss-PLL1	PCC0 VS2	PTF4	PTF7	PCC0 CD2	IOIS16	PCC0 DRV	PTE6	Vss	PTJ3	RAS3	PTE1	RTS2	USB1 ovr_crnt	PTM4

Figure 1.3 Pin Arrangement (PLBG0240JA-A)

1.3.2 Pin Functions

Table 1.2 SH7727 Pin Function

Pin No. (PRQP0240KC-B)	Pin No. (PLBG0240JA-A)	Name	I/O	Function
1	A1	Vcc-RTC*1	—	RTC power supply (1.9 V)
2	B4	XTAL2	O	On-chip RTC crystal oscillator pin
3	A2	EXTAL2	I	On-chip RTC crystal oscillator pin
4	B3	Vss-RTC*1	—	RTC power supply (0 V)
5	A3	MD1	I	Clock mode setting
6	B5	MD2	I	Clock mode setting
7	A4	NMI	I	Nonmaskable interrupt request
8	C4	IRQ0/$\overline{\text{IRL0}}$/PTH[0]	I/I/I	External interrupt / external interrupt / input port H
9	A5	IRQ1/$\overline{\text{IRL1}}$/PTH[1]	I/I/I	External interrupt / external interrupt / input port H
10	D4	IRQ2/$\overline{\text{IRL2}}$/PTH[2]	I/I/I	External interrupt / external interrupt / input port H
11	C5	IRQ3/$\overline{\text{IRL3}}$/PTH[3]	I/I/I	External interrupt / external interrupt / input port H
12	D5	IRQ4/PTH[4]	I/I	External interrupt request / input port H
13	A6	VEPWC	O	LCD panel V_{EE} control
14	B6	VCPWC	O	LCD panel V_{CC} control
15	C6	MD5	I	Endian setting
16	D6	$\overline{\text{BREQ}}$	I	Bus request
17	A7	$\overline{\text{BACK}}$	O	Bus acknowledge
18	B7	VssQ	—	Input/output power supply (0 V)
19	C7	CKIO2	O	System clock output
20	D7	VccQ	—	Input/output power supply (3.3 V)
21	A8	D31/PTB[7]	IO/IO	Data bus / I/O port B
22	B8	D30/PTB[6]	IO/IO	Data bus / I/O port B
23	C8	D29/PTB[5]	IO/IO	Data bus / I/O port B
24	D8	D28/PTB[4]	IO/IO	Data bus / I/O port B
25	A9	D27/PTB[3]	IO/IO	Data bus / I/O port B
26	B9	D26/PTB[2]	IO/IO	Data bus / I/O port B
27	D9	D25/PTB[1]	IO/IO	Data bus / I/O port B

Pin No. (PRQP0240KC-B)	Pin No. (PLBG0240JA-A)	Name	I/O	Function
28	C9	D24/PTB[0]	IO/IO	Data bus / I/O port B
29	A10	VssQ	—	Input/output power supply (0 V)
30	D10	D23/PTA[7]	IO/IO	Data bus / I/O port A
31	C10	VccQ	—	Input/output power supply (3.3 V)
32	B10	D22/PTA[6]	IO/IO	Data bus / I/O port A
33	C11	D21/PTA[5]	IO/IO	Data bus / I/O port A
34	D11	D20/PTA[4]	IO/IO	Data bus / I/O port A
35	B11	Vss	—	Power supply (0 V)
36	A11	D19/PTA[3]	IO/IO	Data bus / I/O port A
37	D12	Vcc	—	Power supply (1.9 V)
38	C12	D18/PTA[2]	IO/IO	Data bus / I/O port A
39	B12	D17/PTA[1]	IO/IO	Data bus / I/O port A
40	A12	D16/PTA[0]	IO/IO	Data bus / I/O port A
41	D13	D15	IO	Data bus
42	C13	VssQ	—	Input/output power supply (0 V)
43	B13	D14	IO	Data bus
44	A13	VccQ	—	Input/output power supply (3.3 V)
45	D14	D13	IO	Data bus
46	C14	D12	IO	Data bus
47	B14	D11	IO	Data bus
48	A14	D10	IO	Data bus
49	D15	D9	IO	Data bus
50	C15	D8	IO	Data bus
51	C17	D7	IO	Data bus
52	A15	D6	IO	Data bus
53	C16	VssQ	—	Input/output power supply (0 V)
54	A16	D5	IO	Data bus
55	B15	VccQ	—	Input/output power supply (3.3 V)
56	A17	D4	IO	Data bus
57	B17	D3	IO	Data bus
58	A18	D2	IO	Data bus
59	B16	D1	IO	Data bus
60	B18	D0	IO	Data bus

Pin No. (PRQP0240KC-B)	Pin No. (PLBG0240JA-A)	Name	I/O	Function
61	A19	A0	O	Address bus
62	D18	A1	O	Address bus
63	B19	A2	O	Address bus
64	C18	VssQ	—	Input/output power supply (0 V)
65	C19	A3	O	Address bus
66	E18	VccQ	—	Input/output power supply (3.3 V)
67	D19	A4	O	Address bus
68	D17	A5	O	Address bus
69	E19	A6	O	Address bus
70	D16	A7	O	Address bus
71	E17	A8	O	Address bus
72	E16	A9	O	Address bus
73	F19	A10	O	Address bus
74	F18	A11	O	Address bus
75	F17	VssQ	—	Input/output power supply (0 V)
76	F16	A12	O	Address bus
77	G19	VccQ	—	Input/output power supply (3.3 V)
78	G18	A13	O	Address bus
79	G17	A14	O	Address bus
80	G16	A15	O	Address bus
81	H19	A16	O	Address bus
82	H18	A17	O	Address bus
83	H17	A18	O	Address bus
84	H16	A19	O	Address bus
85	J19	A20	O	Address bus
86	J18	VssQ	—	Input/output power supply (0 V)
87	J16	A21	O	Address bus
88	J17	VccQ	—	Input/output power supply (3.3 V)
89	K19	A22	O	Address bus
90	K16	A23	O	Address bus
91	K17	Vss	—	Power supply (0 V)
92	K18	A24	O	Address bus
93	L17	Vcc	—	Power supply (1.9 V)

Pin No. (PRQP0240KC-B)	Pin No. (PLBG0240JA-A)	Name	I/O	Function
94	L16	A25	O	Address bus
95	L18	$\overline{BS}$/PTK[4]	O/IO	Bus cycle start signal / I/O port K
96	L19	$\overline{RD}$	O	Read strobe
97	M16	$\overline{WE0}$/DQMLL	O/O	D7–D0 select signal / DQM (SDRAM)
98	M17	$\overline{WE1}$/DQMLU/$\overline{WE}$	O/O/O	D15–D8 select signal / DQM (SDRAM) / PCMCIA $\overline{WE}$
99	M18	$\overline{WE2}$/DQMUL/ $\overline{ICIORD}$/PTK[6]	O/O/ O/IO	D23–D16 select signal / DQM (SDRAM) / PCMCIA I/O read / I/O port K
100	M19	VssQ	—	Input/output power supply (0 V)
101	N16	$\overline{WE3}$/DQMUU/ $\overline{ICIOWR}$/PTK[7]	O/O/ O/IO	D31–D24 select signal / DQM (SDRAM) / PCMCIA I/O write / I/O port K
102	N17	VccQ	—	Input/output power supply (3.3 V)
103	N18	RD/$\overline{WR}$	O	Read/write
104	N19	PTE[7]/PCC0RDY/ $\overline{AUDSYNC}$	IO/I/O	I/O port E / PCMCIA0 ready / AUD synchronization
105	P16	$\overline{CS0}$	O	Chip select 0
106	P17	$\overline{CS2}$	O	Chip select 2
107	P18	$\overline{CS3}$	O	Chip select 3
108	P19	$\overline{CS4}$/PTK[2]	O/IO	Chip select 4 / I/O port K
109	R16	$\overline{CS5}$/$\overline{CE1A}$/PTK[3]	O/O/IO	Chip select 5 / CE1 (area 5 PCMCIA) / I/O port K
110	R17	$\overline{CS6}$/$\overline{CE1B}$	O/O	Chip select 6 / CE1 (area 6 PCMCIA)
111	U17	$\overline{CE2A}$/PTE[4]	O/IO	Area 5 PCMCIA card enable / I/O port E
112	R19	$\overline{CE2B}$/PTE[5]	O/IO	Area 6 PCMCIA card enable / I/O port E
113	T17	AFE_HC1/ USB1d_DPLS/ PTK[0]	O/I/IO	AFE hardware control signal / D+ signal input / I/O port K
114	T19	$\overline{AFE_RLYCNT}$/ USB1d _DMNS/ PTK[1]	O/I/IO	AFE relay control signal / D- signal input / I/O port K
115	R18	VssQ	—	Input/output power supply (0 V)
116	U19	AFE_SCLK/ USB1d _TXDPLS	I/O	AFE clock / D+ transmit output
117	U18	VccQ	—	Input/output power supply (3.3 V)

Pin No. (PRQP0240KC-B)	Pin No. (PLBG0240JA-A)	Name	I/O	Function
118	V19	PTM[7]/PINT[7]/ AFE_FS/ USB1d_RCV	I/I/I/I	Input port M / port interrupt / AFE frame synchronization / receive data input
119	T18	PTM[6]/PINT[6]/ AFE_RXIN/ USB1d_SPEED	I/I/I/O	Input port M / port interrupt / AFE receive data / transceiver speed control
120	V18	PTM[5]/PINT[5]/ AFE_TXOUT/ USB1d_TXSE0	I/I/O/O	Input port M / port interrupt / AFE transmit data / SE0 output
121	W19	PTM[4]/PINT[4]/ $\overline{\text{AFE_RDET}}$	I/I/I	Input port M / port interrupt / AFE ringing detection
122	V16	Reserved/ USB1d_SUSPEND	O/O	Reserved/Transceiver suspend state output
123	W18	$\overline{\text{USB1_ovr_crnt}}$/ USBF_VBUS	I/I	USB host 1 overcurrent detection / USB function VBUS
124	V17	$\overline{\text{USB2_ovr_crnt}}$	I	USB host 2 overcurrent detection
125	W17	$\overline{\text{RTS2}}$/ USB1d_TXENL	O/O	SCIF RTS pin / USB output enable pin
126	V15	PTE[2]/ USB1_pwr_en	IO/O	I/O port E / USB1 voltage control
127	W16	PTE[1]/ USB2_pwr_en	IO/O	I/O port E / USB2 voltage control
128	U16	CKE/PTK[5]	O/IO	CK enable (SDRAM) / I/O port K
129	W15	$\overline{\text{RAS3}}$/PTJ[0]	O/IO	RAS for SDRAM / I/O port J
130	T16	Reserved/PTJ[1]	O/IO	Reserved / I/O port J
131	U15	Reserved/$\overline{\text{CAS}}$/ PTJ[2]	O/O/IO	Reserved / CAS for SDRAM / I/O port J
132	T15	VssQ	—	Input/output power supply (0 V)
133	W14	Reserved/PTJ[3]	O/IO	Reserved / I/O port J
134	V14	VccQ	—	Input/output power supply (3.3 V)
135	U14	Reserved/PTJ[4]	O/IO	Reserved / I/O port J
136	T14	Reserved/PTJ[5]	O/IO	Reserved / I/O port J
137	W13	Vss	—	Power supply (0 V)
138	V13	PTD[5]/CL1	IO/O	I/O port D / LCD line clock
139	U13	Vcc	—	Power supply (1.9 V)
140	T13	PTD[7]/DON	IO/O	I/O port D / LCD DISPLAY on
141	W12	PTE[6]/M_DISP	IO/O	I/O port E / LCD alternating signal / DISP signal

Pin No. (PRQP0240KC-B)	Pin No. (PLBG0240JA-A)	Name	I/O	Function
142	V12	PTE[3]/FLM	IO/O	I/O port E / LCD frame line marker
143	U12	PTE[0]/TDO	IO/O	I/O port E / test data output
144	T12	PCC0RESET/ DRAK0	O/O	PCC reset / DMA request receive
145	W11	$\overline{\text{PCC0DRV}}$/DACK0	O/O	PCC buffer control / DMA acknowledge 0
146	V11	$\overline{\text{WAIT}}$	I	Hardware wait request
147	T11	$\overline{\text{RESETM}}$	I	Manual reset request
148	U11	$\overline{\text{ADTRG}}$/PTH[5]	I/I	Analog trigger / input port H
149	W10	$\overline{\text{IOIS16}}$/PTG[7]	I/I	IOIS16 (PCMCIA) / input port G
150	T10	$\overline{\text{ASEMD0}}$	I	ASE mode
151	U10	PTG[5]/$\overline{\text{ASEBRKAK}}$	I/O	Input port G / ASE break acknowledge
152	V10	PTG[4]	I	Input port G
153	U9	PCC0BVD2/PTG[3]/ AUDATA[3]	I/I/O	PCC BVD2 pin / input port G / AUD data
154	T9	PCC0BVD1/PTG[2]/ AUDATA[2]	I/I/O	PCC BVD1 pin / input port G / AUD data
155	V9	Vss	—	Power supply (0 V)
156	W9	$\overline{\text{PCC0CD2}}$/PTG[1]/ AUDATA[1]	I/I/O	PCMCIA0 CD2 pin / input port G / AUD data
157	T8	Vcc	—	Power supply (1.9 V)
158	U8	$\overline{\text{PCC0CD1}}$/PTG[0]/ AUDATA[0]	I/I/O	PCC CD1 pin / input port G / AUD data
159	V8	VssQ	—	Input/output power supply (0 V)
160	W8	PTF[7]/PINT[15]/ $\overline{\text{TRST}}$	I/I/I	Input port F / port interrupt / test reset
161	T7	VccQ	—	Input/output power supply (3.3 V)
162	U7	PTF[6]/PINT[14]/ TMS	I/I/I	Input port F / port interrupt / test mode switch
163	V7	PTF[5]/PINT[13]/ TDI	I/I/I	Input port F / port interrupt / test data input
164	W7	PTF[4]/PINT[12]/ TCK	I/I/I	Input port F / port interrupt / test clock
165	T6	PTF[3]/PINT[11]/ Reserved	I/I/O	Input port F / port interrupt / Reserved
166	U6	$\overline{\text{PCCREG}}$/PTF[2]/ Reserved	O/I/O	PCC REG pin / input port F/ Reserved

Pin No. (PRQP0240KC-B)	Pin No. (PLBG0240JA-A)	Name	I/O	Function
167	V6	$\overline{\text{PCC0VS1}}$/PTF[1]/ Reserved	I/I/O	PCC VS1 pin / input port F/ Reserved
168	W6	$\overline{\text{PCC0VS2}}$/PTF[0]/ Reserved	I/I/IO	PCC VS2 pin / input port F/ Reserved
169	T5	MD0	I	Clock mode setting
170	U5	Vcc-PLL1*2	—	PLL1 power supply (1.9 V)
171	U3	CAP1	—	PLL1 external capacitance pin
172	W5	Vss-PLL1*2	—	PLL1 power supply (0 V)
173	U4	Vss-PLL2*2	—	PLL2 power supply (0 V)
174	W4	CAP2	—	PLL2 external capacitance pin
175	V5	Vcc-PLL2*2	—	PLL2 power supply (1.9V)
176	W3	$\overline{\text{PCC0WAIT}}$/PTH[6]/ AUDCK	I/I/I	PCC hardware wait request / input port H / AUD clock
177	V3	Vss	—	Power supply (0 V)
178	W2	Vcc	—	Power supply (1.9 V)
179	V4	XTAL	O	Clock oscillator
180	V2	EXTAL	I	External clock / crystal oscillator
181	W1	LCD15/PTM[3]/ PINT[10]	O/I/I	LCD data output / input port M / port interrupt
182	T2	LCD14/PTM[2]/ PINT[9]	O/I/I	LCD data output / input port M / port interrupt
183	V1	LCD13/PTM[1]/ PINT[8]	O/I/I	LCD data output / input port M / port interrupt
184	U2	LCD12/PTM[0]	O/I	LCD data output / input port M
185	U1	STATUS0/PTJ[6]	O/IO	Processor status / I/O port J
186	R2	STATUS1/PTJ[7]	O/IO	Processor status / I/O port J
187	T1	CL2/PTH[7]	O/IO	LCD clock output / I/O port H
188	T3	VssQ	—	Input/output power supply (0 V)
189	R1	CKIO	IO	System clock input/output
190	T4	VccQ	—	Input/output power supply (3.3 V)
191	R3	TxD0/SCPT[0]	O/O	Transmit data 0 / SCI output port
192	R4	SCK0/SCPT[1]	IO/IO	Serial clock 0 / SCI I/O port
193	P1	TxD_SIO/SCPT[2]	O/O	SIOF transmit data / SCI output port
194	P2	SIOMCLK/SCPT[3]	I/IO	SIOF clock input / SCI I/O port
195	P3	TxD2/SCPT[4]	O/O	Transmit data 2 / SCI output port

Pin No. (PRQP0240KC-B)	Pin No. (PLBG0240JA-A)	Name	I/O	Function
196	P4	SCK_SIO/SCPT[5]	IO/IO	SIOF communication clock / SCI I/O port
197	N1	SIOFSYNC/SCPT[6]	IO/IO	SIOF frame synch. / SCI I/O port
198	N2	RxD0/SCPT[0]	I/I	Receive data 0 / SCI input port
199	N3	RxD_SIO/SCPT[2]	I/I	SIOF receive data / SCI input port
200	N4	Vss	—	Power supply (0 V)
201	M1	RxD2/SCPT[4]	I/I	Receive data 2 / SCI input port
202	M2	Vcc	—	Power supply (1.9 V)
203	M3	SCPT[7]/$\overline{\text{CTS2}}$/IRQ5	I/I/I	SCI input port / SCIF clear to send / external interrupt request
204	M4	LCD11/PTC[7]/ PINT[3]	O/IO/I	LCD data out / I/O port C / port interrupt
205	L1	LCD10/PTC[6]/ PINT[2]	O/IO/I	LCD data out / I/O port C / port interrupt
206	L2	LCD9/PTC[5]/ PINT[1]	O/IO/I	LCD data out / I/O port C / port interrupt
207	L4	VssQ	—	Input/output power supply (0 V)
208	L3	LCD8/PTC[4]/ PINT[0]	O/IO/I	LCD data out / I/O port C / port interrupt
209	K1	VccQ	—	Input/output power supply (3.3 V)
210	K4	LCD7/PTD[3]	O/IO	LCD data out / I/O port D
211	K3	LCD6/PTD[2]	O/IO	LCD data out / I/O port D
212	K2	LCD5/PTC[3]	O/IO	LCD data out / I/O port C
213	J3	LCD4/PTC[2]	O/IO	LCD data out / I/O port C
214	J4	LCD3/PTC[1]	O/IO	LCD data out / I/O port C
215	J2	LCD2/PTC[0]	O/IO	LCD data out / I/O port C
216	J1	LCD1/PTD[1]	O/IO	LCD data out / I/O port D
217	H4	LCD0/PTD[0]	O/IO	LCD data out / I/O port D
218	H3	$\overline{\text{DREQ0}}$/PTD[4]	I/I	DMA request / input port D
219	H2	LCLK/UCLK/PTD[6]	I/I/I	LCD clock / USB clock / input port D
220	H1	$\overline{\text{RESETP}}$	I	Power-on reset request
221	G4	CA	I	Hardware standby request
222	G3	MD3	I	Area 0 bus width setting
223	G2	MD4	I	Area 0 bus width setting
224	G1	$\overline{\text{Scan_testen}}$	I	Test pin (fixed to 3.3 V)
225	F4	AVcc_USB	—	USB analog power supply (3.3 V)

Pin No. (PRQP0240KC-B)	Pin No. (PLBG0240JA-A)	Name	I/O	Function
226	F3	USB1_P(analog)	IO	USB1 data I/O (plus)
227	F2	USB1_M(analog)	IO	USB1 data I/O (minus)
228	F1	AVss_USB	—	USB analog power supply (0 V)
229	E4	USB2_P(analog)	IO	USB2 data I/O (plus)
230	E3	USB2_M(analog)	IO	USB2 data I/O (minus)
231	C3	AVcc_USB	—	USB analog power supply (3.3 V)
232	E1	AVss	—	Analog power supply (0 V)
233	D3	AN[2]/PTL[2]	I/I	A/D converter input / input port L
234	D1	AN[3]/PTL[3]	I/I	A/D converter input / input port L
235	E2	AN[4]/PTL[4]	I/I	A/D converter input / input port L
236	C1	AN[5]/PTL[5]	I/I	A/D converter input / input port L
237	C2	AVcc	—	Analog power supply (3.3 V)
238	B1	AN[6]/PTL[6]/DA[1]	I/I/O	A/D converter input / input port L / D/A converter output
239	D2	AN[7]/PTL[7]/DA[0]	I/I/O	A/D converter input / input port L / D/A converter output
240	B2	AVss	—	Analog power supply (0 V)

Notes: All Vcc/Vss should be connected to the all system power supply (so that power is supplied at all times).

1. Always supply power to the Vcc-RTC, even if RTC is not being used.
2. Always supply power to the Vcc-PLL, even if the internal PLL is not being used.
3. Drive high when using the user system alone, and not using an emulator or the H-UDI. When this pin is low or open, $\overline{\text{RESETP}}$ may be masked.

Section 2 CPU

2.1 Registers

The SH7727 has the same registers as in SH-3. In addition, the SH7727 also support the same DSP-related registers seen in SH-DSP. The basic software-accessible registers are divided into four distinct groups:

- General-purpose registers
- Control registers
- System registers
- DSP registers

With the exception of a number of DSP registers, all of these registers are 32-bit width. The general-purpose registers are accessible from the user mode, with R0 to R7 banked to provide each processor mode access to a separate set of the R0 to R7 registers (i.e. R0 to R7_BANK0, and R0 to R7_BANK1). In the privileged mode, the register bank (RB) bit in the status register (SR) defines which set of banked registers (R0 to R7_BANK0 or R0 to R7_BANK1) is accessed as general-purpose registers, and which are accessed only by the LDC/STC instructions.

The control registers can be accessed by the LDC/STC instructions. The GBR, RS, RE, and MOD registers can also be accessed in user mode. Control registers are:

- SR: Status register
- SSR: Saved status register
- SPC: Saved program counter
- GBR: Global base register
- VBR: Vector base register
- RS: Repeat start register (DSP mode only)
- RE: Repeat end register (DSP mode only)
- MOD: Modulo register (DSP mode only)

The system registers are accessed by the LDS/STS instructions (the PC cannot be accessed by software, but is included here because its contents are saved in, and restored from, SPC). The system registers are:

- MACH: Multiply and accumulate high register
- MACL: Multiply and accumulate low register
- PR: Procedure register

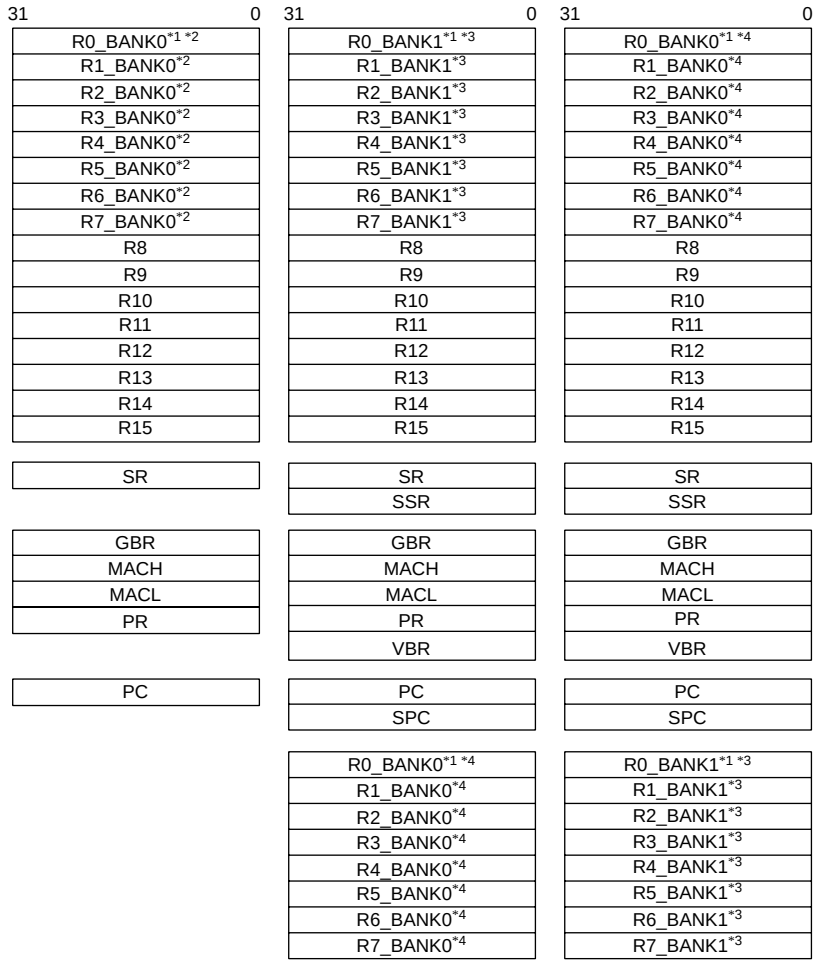
- PC: Program counter

This section explains the usage of these registers in different modes.

Figures 2.1 and 2.2 show the register configuration in each processing mode.

Switching between user mode and privileged mode is carried out by means of the processing operation mode bit (MD) in the status register.

The DSP mode is switched by means of the DSP bit in the status register.



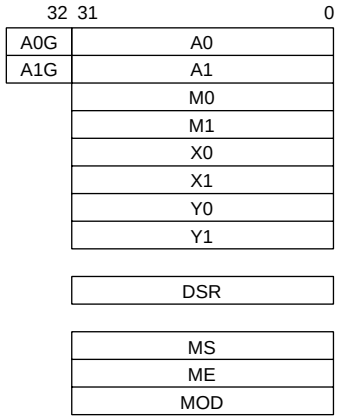
(a) User mode register configuration

(b) Privileged mode register configuration (RB = 1)

(c) Privileged mode register configuration (RB = 0)

Notes: 1. The R0 register is used as an index register in indexed register indirect addressing mode and indexed GBR indirect addressing mode.
2. Bank register
3. Bank register
Accessed as a general register when the RB bit is set to 1 in the SR register.
Accessed only by LDC/STC instructions when the RB bit is cleared to 0.
4. Bank register
Accessed as a general register when the RB bit is cleared to 0 in the SR register.
Accessed only by LDC/STC instructions when the RB bit is set to 1.

Figure 2.1 Register Configuration in Each Processing Mode (1)



(d) DSP mode register configuration (DSP = 1)

Figure 2.2 Register Configuration in Each Processing Mode (2)

Register values after a reset are shown in table 2.1.

Table 2.1 Initial Register Values

Type	Registers	Initial Value*
General registers	R0 to R15	Undefined
Control registers	SR	MD bit = 1, RB bit = 1, BL bit = 1, I3 to I0 = 1111 (H'F), reserved bits = 0, others undefined
	GBR, SSR, SPC	Undefined
	VBR	H'00000000
	RS, RE	Undefined
	MOD	Undefined
System registers	MACH, MACL, PR	Undefined
	PC	H'A0000000
DSP registers	A0, A0G, A1, A1G, M0, M1, X0, X1, Y0, Y1	Undefined
	DSR	H'00000000

Note: * Initialized by a power-on or manual reset.

2.1.1 General Purpose Registers

There are sixteen 32-bit general registers (Rn), designated R0 to R15. The general registers are used for data processing and address calculation.

With SuperH microcomputer type instructions, R0 is used as an index register. With a number of instructions, R0 is the only register that can be used. R15 is used as the stack pointer (SP). In exception handling, R15 is used to reference the stack when saving and restoring the status register (SR) and program counter (PC).

With DSP type instructions, eight of the sixteen general registers are used for addressing of X and Y data memory and data memory (single data) that uses the L-bus.

To access X memory, R4 and R5 are used as X address register [Ax] and R8 is used as X index register [Ix]. To access Y memory, R6 and R7 are used as Y address register [Ay] and R9 is used as Y index register [Iy]. To access single data that uses the L-bus, R2, R3, R4, and R5 are used as single data address register [As] and R8 is used as single data index register [Is].

Figure 2.3 shows the general purpose registers, which are identical to SH-3's, when DSP extension is disabled.

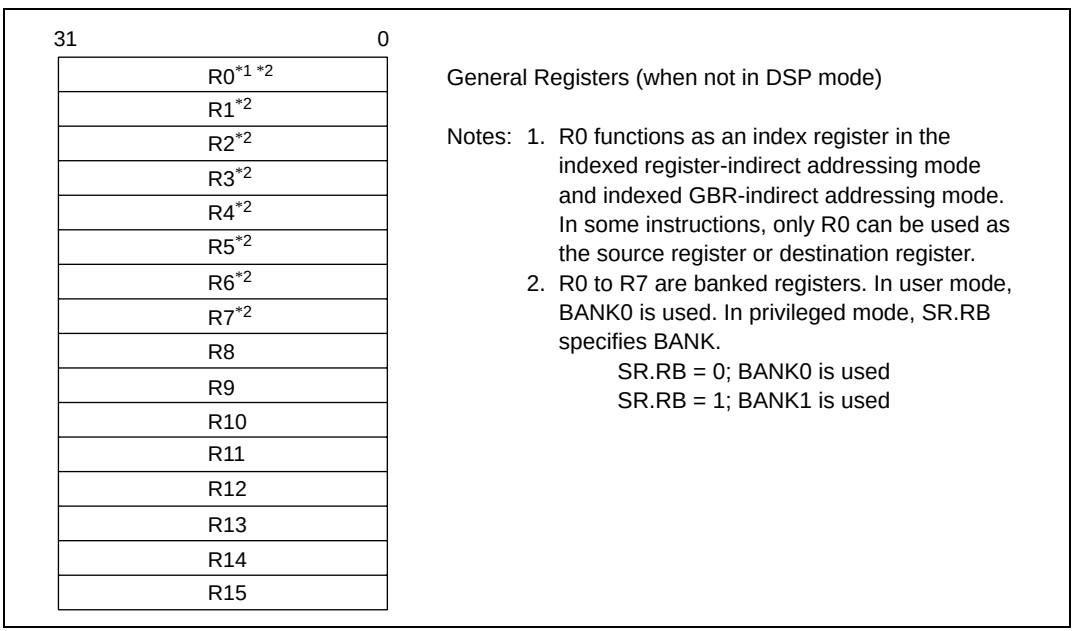


Figure 2.3 General Purpose Register (Not in DSP Mode)

On the other hand, R2 to R9 registers are also used for the DSP data address calculations, see figure 2.4, when DSP extension is enabled. Another symbol that represents the purpose of the registers in DSP type instruction is [ ].

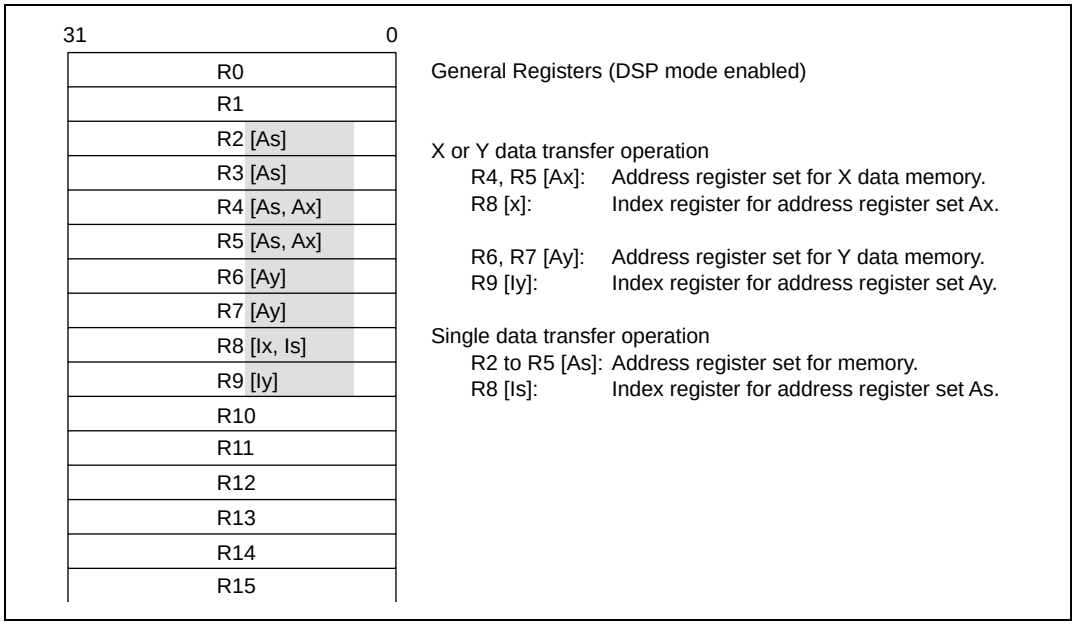


Figure 2.4 General Purpose Register (DSP Mode)

DSP type instructions can access X and Y data memory simultaneously. To specify addresses for X and Y data memory, two address pointer sets are prepared. These are:

R8[Ix], R4, R5[Ax] for X memory access, and R9[Iy], R6, R7[Ay] for Y memory access.

The names (symbol) R2 to R9 are used in the Assembler, but users can use other register names that represent the purpose of the register in the DSP instruction explicitly. In the assembly program, user can use an alias for the register. The coding in assembler is as follows.

```
Ix: .REG (R8)
```

The name Ix is the alias for R8. Other aliases are as follows.

```
Ax0:  .REG   (R4)
Ax1:  .REG   (R5)
Ix:   .REG   (R8)
Ay0:  .REG   (R6)
Ay1:  .REG   (R7)
Iy:   .REG   (R9)
As0:  .REG   (R4)  ;  This is optional. If you need another alias for single data transfer.
As1:  .REG   (R5)  ;  This is optional. If you need another alias for single data transfer.
As2:  .REG   (R2)
As3:  .REG   (R3)
Is:   .REG   (R8)  ;  This is optional. If you need another alias for single data transfer.
```

2.1.2 Control Registers

SH7727 has eight control registers: SR, SSR, SPC, GBR, VBR, RS, RE, and MOD (figure 2.5). SSR, SPC, GBR and VBR are the same as the SH-3 registers. The DSP mode is activated only when SR.DSP = 1.

Repeat start register RS, repeat end register RE, and repeat counter RC (12-bit part of the SR) and repeat control bits RF0 and RF1 are new registers and control bits which are used for repeat control. Modulo register MOD and modulo control bits DMX and DMY in the SR are also new register and control bits.

In the SR, there are six additional control bits: RC[11:0], RF0, RF1, DMX, DMY and DSP. Bits DMX, DMY, RC[11:0], and RF[1:0] can be modified in supervisor mode, DSP supervisor mode, and DSP user mode. The DMX and DMY are used for modulo addressing control. If the DMX is 1 then the modulo addressing mode is effective for the X memory address pointer, Ax (R4 or R5). If the DMY is 1 then it is effective for the Y memory address pointer, Ay (R6 or R7). However, both X and Y address pointer cannot be operated under the modulo addressing mode even though both DMX and DMY bits are set. The case of DMX = DMY = 1 is reserved for future expansion. When both DMX and DMY are set simultaneously, the hardware will preliminary treat only address pointer as the modulo addressing mode. Modulo addressing is available for X and Y data transfer operation (MOVX and MOVY), but not for single data transfer operation (MOVS).

The RF1 and RF0 hold information of the number of repeat steps and they are set when a SETRC instruction is executed. When RF[1:0] shows 00, the current repeat module consists of one-step instruction. When RF[1:0] = 01, it means two-step instructions. When RF[1:0] = 11, it means

three-step instruction. When RF[1:0] = 10, it means the current repeat module consists of four or more instructions.

Although RC[11:0] and RF[1:0] can be changed by a store/load to SR, use of the dedicated manipulation instruction SETRC is recommended.

The SR also has a 12-bit repeat counter RC which is used for efficient loop control. Repeat start register (RS) and repeat end register (RE) are also introduced for the loop control. They keep the start and end addresses of a loop (the contents of the registers, RS and RE are slightly different from the actual loop start and end address).

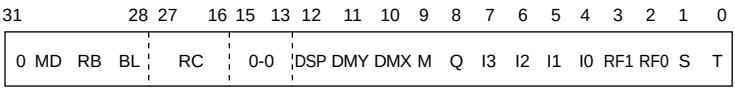
Modulo register, MOD is introduced to realize modulo addressing for circular data buffering. MOD keeps the modulo start address (MS) and the modulo end address (ME).

In order to access RS, RE and MOD, load/store (control register) instructions for them are introduced. An example for RS is as follows:

```
LDC Rm,RS;        Rm → RS
LDC.L @Rm+,RS;    (Rm) → RS, Rm+4 → Rm
STC RS,Rn;        RS → Rn
STC.L RS,@-Rn;    Rn-4 → Rn, RS → (Rn)
```

Address set instructions for the RS and RE are also prepared.

```
LDRS @(disp,PC); disp × 2 + PC → RS
LDRE @(disp,PC); disp × 2 + PC → RE
```



SR (Status register)

MD bit: Processor operation mode
MD = 1: Privileged mode
MD = 0: User mode

RB bit: Register bank bit; used to define the general registers in privileged mode.
RB = 1: R0_BANK1 to R7_BANK1 are used as general registers.
R0_BANK0 to R7_BANK0 accessed by LDC/STC instructions.
RB = 0: R0_BANK0 to R7_BANK0 are used as general registers.
R0_BANK1 to R7_BANK1 accessed by LDC/STC instructions.

BL bit: Block bit; used to mask exception in privileged mode.
BL = 1: Interrupts are masked (not accepted)
BL = 0: Interrupts are accepted

RC [11:0]: 12-bit repeat counter

DSP bit: DSP operation mode
DSP = 1: DSP instructions (LDS Rm, DSR/A0/X0/X1/Y0/Y1, LDS.L @Rm+, DSR/A0/X0/X1/Y0/Y1, STS DSR/A0/X0/X1/Y0/Y1, Rn, STS.L DSR/A0/X0/X1/Y0/Y1, @−Rn, LDC Rm, RS/RE/MOD, LDC.L @Rm+, RS/RE/MOD, STC RS/RE/MOD,Rn, STC.L RS/RE/MOD, @−Rn, LDRS, LDRE, SETRC, MOVS, MOVX, MOVY, Pxxx) are enabled.
DSP = 0: All DSP instructions are treated as illegal instructions; only SH3 instructions are supported.

DMY bit: Modulo addressing enable for Y side

DMX bit: Modulo addressing enable for X side

Q, M bit: Used by DIV0U/S and DIV1 instructions.

I [3:0]: 4-bit field indicating the interrupt request mask level.

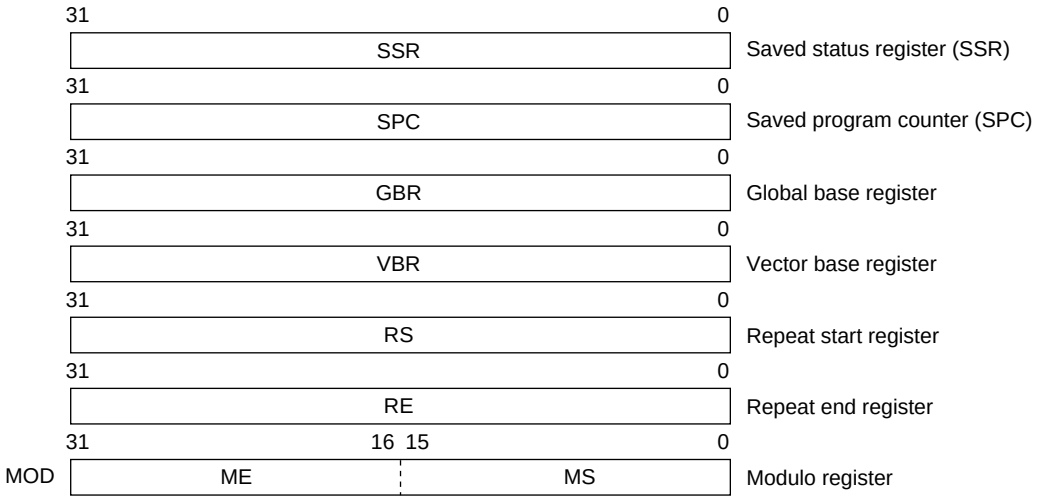
RF [1:0]: Used for repeat control

S bit: Used by the MAC instructions and DSP data.

T bit: The MOVT, CMP/cond, TAS, TST, BT, BF, SETT, CLRT and DT instructions use the T bit to indicate true (logic one) or false (logic zero). The ADDV/C, SUBV/C, DIV0U/S, DIV1, NEGC, SHAR/L, SHLR/L, ROTR/L and ROTCR/L instructions also use the T bit to indicate a carry, borrow, overflow, or underflow.

Reserved bits: Always read as 0, and should always be written with 0 (bit 31, bits 15 to 13).

Figure 2.5 Control Registers (1)



ME: Modulo end address, MS: Modulo start address

Saved status register (SSR)
Stores current SR value at time of exception to indicate processor status when returning to instruction stream from exception handler.

Saved program counter (SPC)
Stores current PC value at time of exception to indicate return address on completion of exception handling.

Global base register (GBR)
Stores base address of GBR-indirect addressing mode. The GBR-indirect addressing mode is used for data transfer and logical operations on the on-chip peripheral module register area.

Vector base register (VBR)
Stores base address of exception vector area.

Repeat start register (RS)
Used in DSP mode only. Indicates start address of repeat loop.

Repeat end register (RE)
Used in DSP mode only. Indicates address of repeat loop end.

Modulo register (MOD)
Used in DSP mode only.
MOD[31:16]: ME: Modulo end address, MOD[15:0]: MS: Modulo start address.
In X/Y operand address generation, the CPU compares the address with ME, and if it is the same, loads MS in either the X or Y operand address register (depending on bits DMX and DMY in the SR register).

Figure 2.5 Control Registers (2)

2.1.3 System Registers

The SH7727 has four system registers, MACL, MACH, PR and PC (figure 2.6).

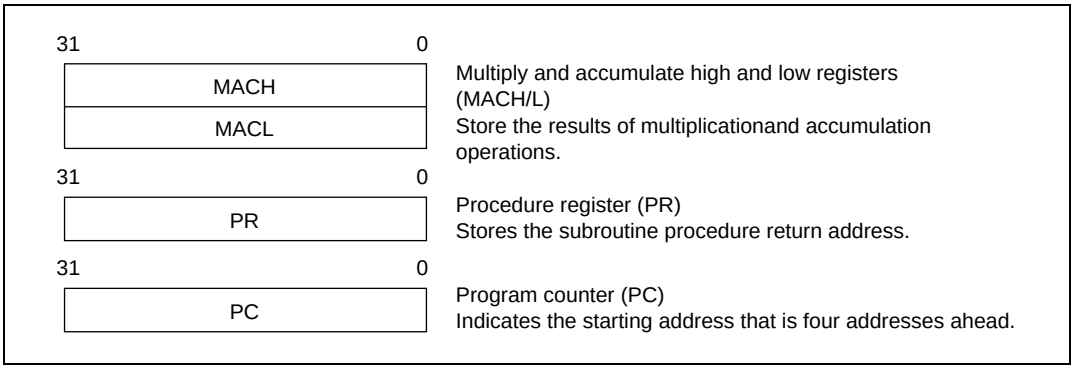


Figure 2.6 System Registers

DSR, A0, X0, X1, Y0 and Y1 registers are also treated as system registers. So, data transfer instructions between general registers and system registers are supported for them.

2.1.4 DSP Registers

The SH7727 has eight data registers and one control register (figure 2.7). The data registers are 32-bit width with the exception of registers A0 and A1. Registers A0 and A1 include 8 guard bits (fields A0G and A1G), giving them a total width of 40 bits.

Three types of operations access the DSP data registers. First one is the DSP data. When a DSP fixed-point data operation uses A0 or A1 for source register, it uses the guard bits (bits 39 to 32). When it uses A0 or A1 for destination register, bits 39 to 32 in the guard bit are valid. When a DSP fixed-point data operation uses the DSP registers other than A0 and A1 for source register, it sign-extends the source value to bits 39 to 32. When it uses them for destination register, the bits 39 to 32 of the result is discard.

Second one is X and Y data transfer operation, “MOVX.W MOVY.W”. This operation accesses the X and Y memories through 16-bit X and Y data buses (figure 2.8). Registers to be loaded or stored by this operation are always upper 16 bits (bits 31 to 16). X0 and X1 can be destination of the X memory load and Y0 and Y1 can be destination of Y memory load, but other register cannot be destination register of this operation.

When data is read into the upper 16 bits of a register (bits 31 to 16), the lower 16 bits of the register (bits 15 to 0) are automatically cleared. A0 and A1 can be stored to the X or Y memory by this operation, but other registers cannot be stored.

There are some rules to access SR by STC/LDC instruction.

1. When DSP is disabled, same as SH-3 behavior
2. When SDP supervisor mode, same as supervisor mode
3. In User DSP mode, SR can be read by STC instruction
4. In User DSP mode, LDC to SR is allowed but no DSP related bits are protected from write.

Table 2.2 shows detail behavior under each SH3-DSP mode.

Table 2.2 Detail Behavior Under Each SH3-DSP Mode

Fields	Supervisor Mode MD = 1 & DSP = 0	User Mode MD = 0 & DSP = 0	DSP Supervisor Mode MD = 1 & DSP = 1	DSP User Mode MD = 0 & DSP = 1	Access to DSP Related Bits by Dedicated Instruction	Initial Value after Reset
MD	S: OK, L: OK	S, L: illegal instruction	S: OK, L: OK	S: OK, L: NG		1
RB	S: OK, L: OK	S, L: illegal instruction	S: OK, L: OK	S: OK, L: NG		1
BL	S: OK, L: OK	S, L: illegal instruction	S: OK, L: OK	S: OK, L: NG		1
RC [11:0]	S: OK, L: OK	S, L: illegal instruction	S: OK, L: OK	S: OK, L: OK	SETRC instruction	0b000000000000
DSP	S: OK, L: OK	S, L: illegal instruction	S: OK, L: OK	S: OK, L: NG		0
DMX	S: OK, L: OK	S, L: illegal instruction	S: OK, L: OK	S: OK, L: OK		0
DMY	S: OK, L: OK	S, L: illegal instruction	S: OK, L: OK	S: OK, L: OK		0
Q	S: OK, L: OK	S, L: illegal instruction	S: OK, L: OK	S: OK, L: NG		X
M	S: OK, L: OK	S, L: illegal instruction	S: OK, L: OK	S: OK, L: NG		X
I [3:0]	S: OK, L: OK	S, L: illegal instruction	S: OK, L: OK	S: OK, L: NG		1111
RF [1:0]	S: OK, L: OK	S, L: illegal instruction	S: OK, L: OK	S: OK, L: OK	SETRC instruction	X
S	S: OK, L: OK	S, L: illegal instruction	S: OK, L: OK	S: OK, L: NG		X
T	S: OK, L: OK	S, L: illegal instruction	S: OK, L: OK	S: OK, L: NG		X

(S) STC: Store SR to Rn, SR → Rn
(L) LDC: Load Rn to SR, Rn → SR
OK: Allowed to STC/LSC operation
Illegal instruction: Treated as illegal instruction, exception should be occurred
NG: Keep previous value, nothing changed

Third one is single-data transfer instruction, “MOVS.W” and “MOVS.L”. This instruction accesses any memory location through LDB (figure 2.8). All DSP registers connect to the LDB and be able to be source and destination register of the data transfer. It has word and longword access modes. In the word mode, registers to be loaded or stored by this instruction are upper 16 bits (bits 31 to 16) for the DSP registers except A0G and A1G. When data is loaded into a register

other than A0G and A1G in the word mode, lower half of the register is cleared. When it is A0 or A1, the data is sign-extended to bits 39 to 32 and lower half of it is cleared. When A0G or A1G is a destination register in the word mode, data is loaded into 8-bit register, but A0 or A1 is not cleared. In the longword mode, when a destination register is A0 or A1, it is sign-extended to bits 39 to 32.

Tables 2.3 and 2.4 show the data type of registers used in the DSP instructions. Some instructions cannot use some registers shown in the tables because of instruction code limitation. For example, PMULS can use A1 for source registers, but cannot use A0. These tables ignore details of the register selectability.

Table 2.3 Destination Register of DSP Instructions

Registers	Instructions		Guard Bits 39 32	Register Bits 31 16	Register Bits 15 0
A0, A1	DSP	Fixed-point, PSHA, PMULS	Sign-extended	40-bit result	
		Integer, PDMSB	Sign-extended	24-bit result	Cleared
		Logical, PSHL	Cleared	16-bit result	Cleared
	Data transfer	MOVS.W	Sign-extended	16-bit data	Cleared
		MOVS.L	Sign-extended	32-bit data	
A0G, A1G	Data transfer	MOVS.W	Data	No update	
		MOVS.L	Data	No update	
X0, X1 Y0, Y1 M0, M1	DSP	Fixed-point, PSHA, PMULS		32-bit result	
		Integer, logical, PDMSB, PSHL		16-bit result	Cleared
	Data transfer	MOVX/Y.W, MOVS.W		16-bit result	Cleared
		MOVS.L		32-bit data	

Table 2.4 Source Register of DSP Operations

Registers	Instructions		Guard Bits 39 32	Register Bits 31 16	Register Bits 15 0
A0, A1	DSP	Fixed-point, PDMSB, PSHA	40-bit data		
		Integer	24-bit data		
		Logical, PSHL, PMULS		16-bit data	
	Data transfer	MOVX/Y.W, MOVS.W		16-bit data	
		MOVS.L		32-bit data	
A0G, A1G	Data transfer	MOVS.W	Data		
		MOVS.L	Data		
X0, X1 Y0, Y1 M0, M1	DSP	Fixed-point, PDMSB, PSHA	Sign*	32-bit data	
		Integer	Sign*	16-bit data	
		Logical, PSHL, PMULS		16-bit data	
	Data transfer	MOVS.W		16-bit data	
		MOVS.L		32-bit data	

Note: * Sign-extend the data and feed to the ALU

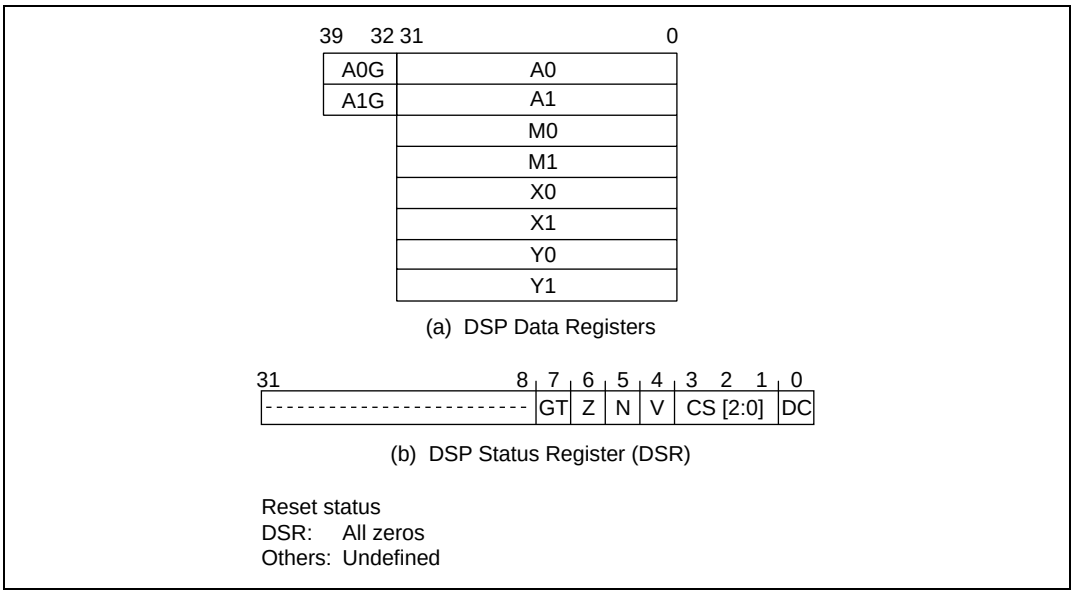


Figure 2.7 DSP Registers

Table 2.5 DSR Register Bits

Bit	Name (Abbreviation)	Function
31–8	Reserved bits	0: Always read out; always use 0 as a write value
7	Signed greater than bit (GT)	Indicates that the operation result is positive (excepting 0), or that operand 1 is greater than operand 2 1: Operation result is positive, or operand 1 is greater
6	Zero bit (Z)	Indicates that the operation result is zero (0), or that operand 1 is equal to operand 2 1: Operation result is zero (0), or equivalence
5	Negative bit (N)	Indicates that the operation result is negative, or that operand 1 is smaller than operand 2 1: Operation result is negative, or operand 1 is smaller
4	Overflow bit (V)	Indicates that the operation result has overflowed 1: Operation result has overflowed
3–1	Status selection bits (CS)	Designate the mode for selecting the operation result status set in the DC bit Do not set either 110 or 111 000: Carry/borrow mode 001: Negative value mode 010: Zero mode 011: Overflow mode 100: Signed greater mode 101: Signed above mode
0	DSP status bit (DC)	Sets the status of the operation result in the mode designated by the CS bits 0: Designated mode status not realized (unrealized) 1: Designated mode status realized

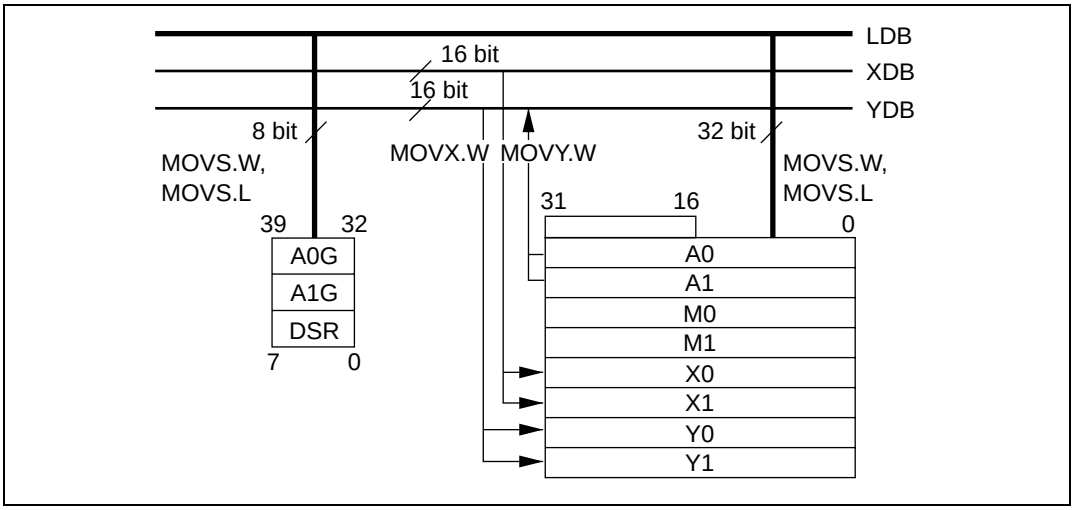


Figure 2.8 Connections of DSP Registers and Buses

The DSP unit has DSP status register (DSR). The DSR has conditions of the DSP data operation result (zero, negative, and so on) and a DC bit which is similar to the T bit in the CPU. The DC bit indicates the one of the conditional flags. A conditional DSP data processing instruction controls its execution based on the DC bit. This control affects only the operations in the DSP unit; it controls the update of DSP registers only. It cannot control operations in CPU, such as address register updating and load/store operations. The control bit CS[2:0] specifies the condition to be reflect to the DC bit.

The unconditional DSP type data operations, except PMULS, MOVX, MOVY and MOVS, update the conditional flags and DC bit, but no CPU instructions, including MAC instructions, update the DC bit. The conditional DSP type instructions do not update the DSR either.

DSR is assigned as a system register and load/store instructions are prepared as follows:

```
STS DSR,Rn;
STS.L DSR,@-Rn;
LDS Rn,DSR;
LDS.L @Rn+,DSR;
```

When DSR is read by the STS instructions, the upper bits (bit 31 to bit 8) are all 0.

2.2 Data Format

2.2.1 Data Format in Registers (Non-DSP Type)

Register operands are always longwords (32 bits) (figure 2.9). When the memory operand is only a byte (8 bits) or a word (16 bits), it is sign-extended into a longword when loaded into a register.

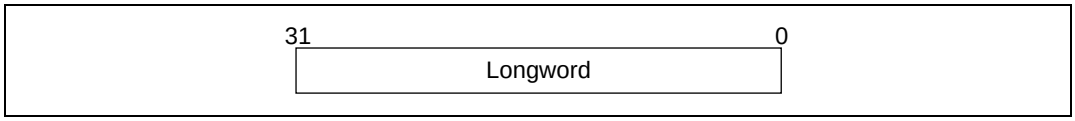


Figure 2.9 Longword Operand

2.2.2 DSP-Type Data Format

The SH7727 has several different data formats that depend on operations. This section explains the data formats for DSP type instructions.

Figure 2.10 shows three DSP-type data formats with different binary point positions. A CPU-type data format with the binary point to the right of bit 0 is also shown for reference.

The DSP-type fixed point data format has the binary point between bit 31 and bit 30. The DSP-type integer format has the binary point between bit 16 and bit 15. The DSP-type logical format does not have a binary point. The valid data lengths of the data formats depend on the operations and the DSP registers.

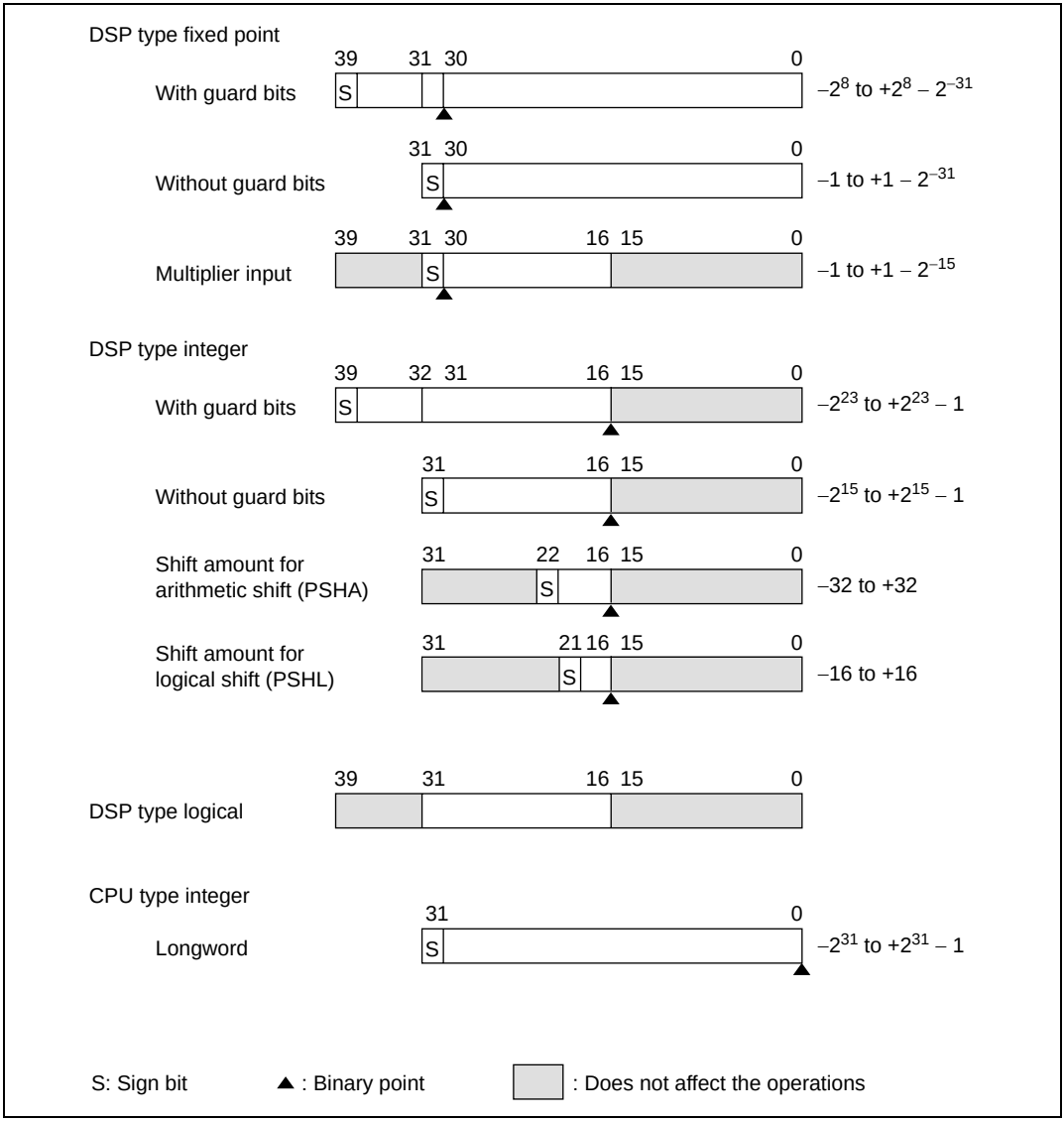


Figure 2.10 Data Format

Shift amount for arithmetic shift (PSHA) instruction has 7 bits filed that could represent –64 to +63, however –32 to +32 is the valid number for the operation. Also the shift amount for logical shift operation has 6-bits field, however –16 to +16 is the valid number for the instruction.

2.2.3 Data Format in Memory

Memory data formats are classified into bytes, words, and longwords. Byte data can be accessed from any address, but an address error will occur if the word data starting from an address other than 2n or longword data starting from an address other than 4n is accessed. In such cases, the data accessed cannot be guaranteed (figure 2.11).

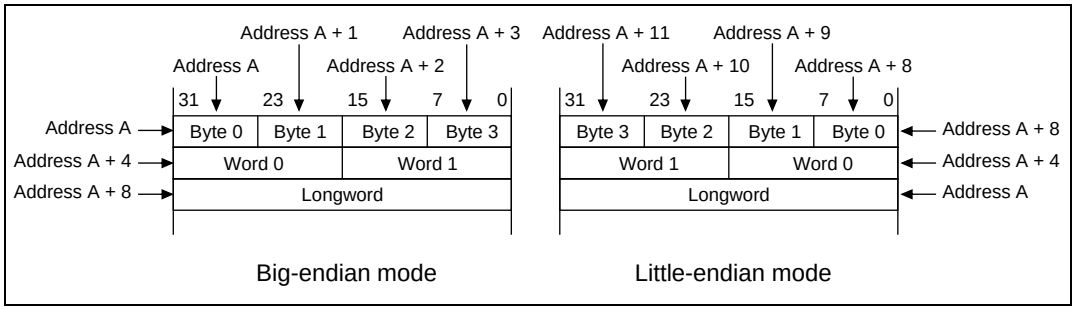


Figure 2.11 Byte, Word, and Longword Alignment

As the data format, either big endian or little endian byte order can be selected, according to the MD5 pin at reset. When MD5 is low at reset, the processor operates in big endian. When MD5 is high at reset, the processor operates in little endian.

2.3 Features of CPU Core Instructions

The CPU core instructions are RISC-type instructions with the following features:

Fixed 16-Bit Length: All instructions have a fixed length of 16 bits. This improves program code efficiency.

One Instruction per State: Pipelining is used, and basic instructions can be executed in one state. At 160-MHz operation, one state is 6.25 ns.

Data Size: The basic data size for operations is longword. Byte, word, or longword can be selected as the memory access size. Memory byte or word data is sign-extended and operated on as longword data. Immediate data is sign-extended to longword size for arithmetic operations or zero-extended to longword size for logical operations.

Table 2.6 Word Data Sign Extension

SH7727 CPU		Description	Example of Other CPU	
MOV.W	@(disp,PC),R1	Sign-extended to 32 bits, R1 becomes H'00001234, and is then operated on by the ADD instruction.	ADD.W	#H'1234,R0
ADD	R1,R0			
.....				
.DATA.W	H'1234			

Note: Immediate data is referenced by @(disp,PC).

Load/Store Architecture: Basic operations are executed between registers. In operations involving memory, data is first loaded into a register (load/store architecture). However, bit manipulation instructions such as AND are executed directly on memory.

Delayed Branching: Unconditional branch instructions, etc., are executed as delayed branches. With a delayed branch instruction, the branch is made after execution of the instruction (called the slot instruction) immediately following the delayed branch instruction. This minimizes disruption of the pipeline when a branch is made.

With a delayed branch, the actual branch operation occurs after execution of the slot instruction. However, instruction execution for register updating, etc., excluding the branch operation, is performed in delayed branch instruction → delay slot instruction order. For example, even though the contents of the register holding the branch destination address are changed in the delay slot, the branch destination address remains as the register contents prior to the change.

Table 2.7 Delayed Branch Instructions

SH7727 CPU		Description	Example of Other CPU	
BRA	TRGET	ADD is executed before branch to TRGET.	ADD.W	R1,R0
ADD	R1,R0		BRA	TRGET

Multiply/Multiply-and-Accumulate Operations: A $16 \times 16 \rightarrow 32$ multiply operation is executed in 1 to 3 states, and a $16 \times 16 + 64 \rightarrow 64$ multiply-and-accumulate operation in 2 to 3 states. A $32 \times 32 \rightarrow 64$ multiply operation and a $32 \times 32 + 64 \rightarrow 64$ multiply-and-accumulate operation are each executed in 2 to 5 states.

T Bit: The result of a comparison is indicated by the T bit in the status register (SR), and a conditional branch is performed according to whether the result is True or False. Processing speed has been improved by keeping the number of instructions that modify the T bit to a minimum.

Table 2.8 T Bit

SH7727 CPU		Description	Example of Other CPU	
CMP/GE	R1,R0	If R0 ≥ R1, the T bit is set.	CMP.W	R1,R0
BT	TRGET0	A branch is made to TRGET0 if R0 ≥ R1, or to TRGET1 if R0 < R1.	BGE	TRGET0
BF	TRGET1		BLT	TRGET1
ADD	#–1,R0	The T bit is not set by ADD.	SUB.W	#1,R0
CMP/EQ	#0,R0	If R0 = 0, the T bit is set.	BEQ	TRGET
BT	TRGET	A branch is made if R0 = 0.		

Immediate Data: Byte immediate data is placed inside the instruction code. Word and longword immediate data is not placed inside the instruction code, but in a table in memory. The table in memory is referenced with an immediate data transfer instruction (MOV) using PC-relative addressing mode with displacement.

Table 2.9 Immediate Data Referencing

Type	SH7727 CPU		Example of Other CPU	
8-bit immediate	MOV	#H'12,R0	MOV.B	#H'12,R0
16-bit immediate	MOV.W	@(disp,PC),R0	MOV.W	#H'1234,R0
		.....		
	.DATA.W	H'1234		
32-bit immediate	MOV.L	@(disp,PC),R0	MOV.L	#H'12345678,R0
		.....		
	.DATA.L	H'12345678		

Note: Immediate data is referenced by @(disp,PC).

Absolute Addresses: When data is referenced by absolute address, the absolute address value is placed in a table in memory beforehand. Using the method whereby immediate data is loaded when an instruction is executed, this value is transferred to a register and the data is referenced using register indirect addressing mode.

Table 2.10 Absolute Address Referencing

Type	SH7727 CPU	Example of Other CPU
Absolute address	MOV.L @(disp,PC),R1 MOV.B @R1,R0 ..... .DATA.L H'12345678	MOV.B @H'12345678,R0

16-Bit/32-Bit Displacement: When data is referenced with a 16- or 32-bit displacement, the displacement value is placed in a table in memory beforehand. Using the method whereby immediate data is loaded when an instruction is executed, this value is transferred to a register and the data is referenced using indexed register indirect addressing mode.

Table 2.11 Displacement Referencing

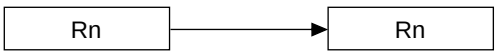
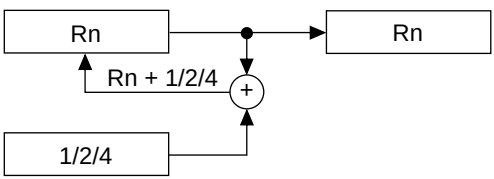
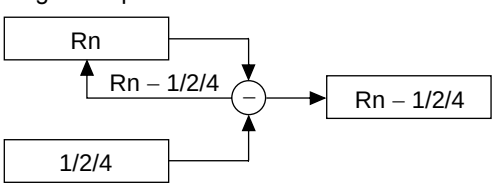
Type	SH7727 CPU	Example of Other CPU
16-bit displacement	MOV.W @(disp,PC),R0 MOV.W @(R0,R1),R2 ..... .DATA.W H'1234	MOV.W @(H'1234,R1),R2

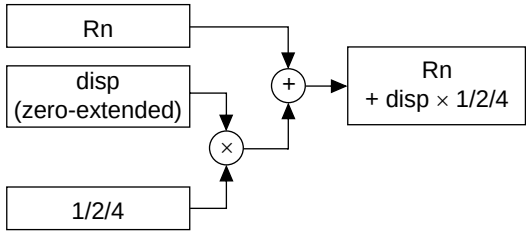
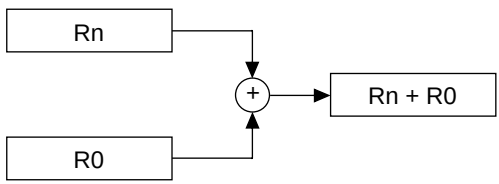
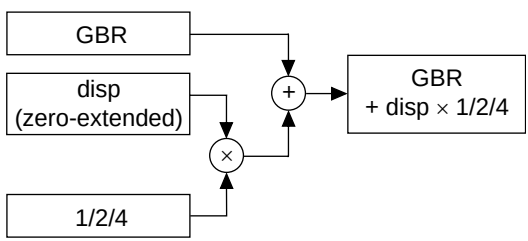
2.4 Instruction Formats

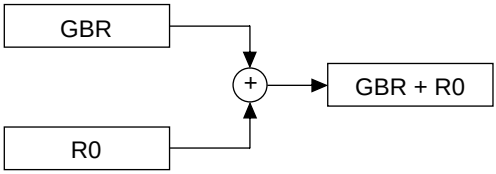
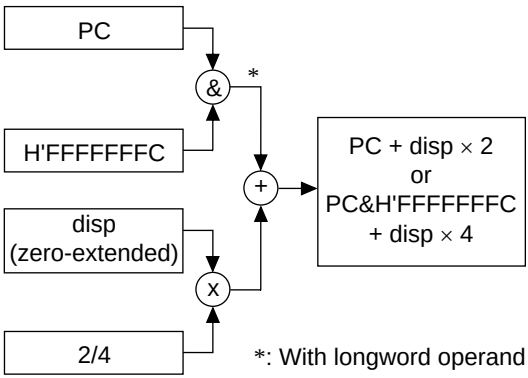
2.4.1 CPU Instruction Addressing Modes

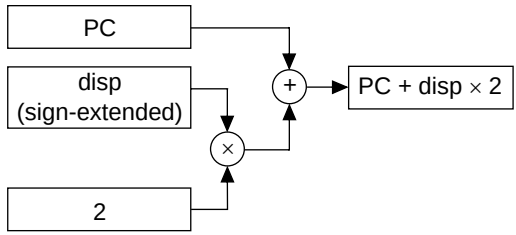
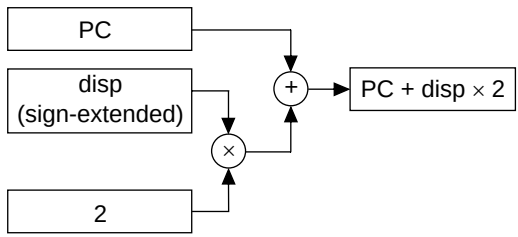
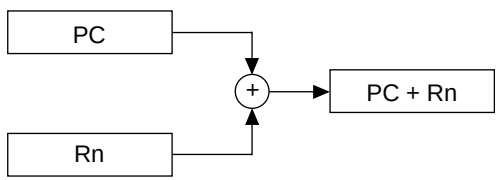
The following table shows addressing modes and effective address calculation methods for instructions executed by the CPU core.

Table 2.12 Addressing Modes and Effective Addresses for CPU Instructions

Addressing Mode	Instruction Format	Effective Address Calculation Method	Calculation Formula
Register direct	Rn	Effective address is register Rn. (Operand is register Rn contents.)	—
Register indirect	@Rn	Effective address is register Rn contents. 	Rn
Register indirect with post-increment	@Rn+	Effective address is register Rn contents. A constant is added to Rn after instruction execution: 1 for a byte operand, 2 for a word operand, 4 for a longword operand. 	Rn After instruction execution Byte: Rn + 1 → Rn Word: Rn + 2 → Rn Longword: Rn + 4 → Rn
Register indirect with pre-decrement	@–Rn	Effective address is register Rn contents, decremented by a constant beforehand: 1 for a byte operand, 2 for a word operand, 4 for a longword operand. 	Byte: Rn – 1 → Rn Word: Rn – 2 → Rn Longword: Rn – 4 → Rn (Instruction executed with Rn after calculation)

Addressing Mode	Instruction Format	Effective Address Calculation Method	Calculation Formula
Register indirect with displacement	@(disp:4, Rn)	Effective address is register Rn contents with 4-bit displacement disp added. After disp is zero-extended, it is multiplied by 1 (byte), 2 (word), or 4 (longword), according to the operand size. 	Byte: Rn + disp Word: Rn + disp × 2 Longword: Rn + disp × 4
Indexed register indirect	@(R0, Rn)	Effective address is sum of register Rn and R0 contents. 	Rn + R0
GBR indirect with displacement	@(disp:8, GBR)	Effective address is register GBR contents with 8-bit displacement disp added. After disp is zero-extended, it is multiplied by 1 (byte), 2 (word), or 4 (longword), according to the operand size. 	Byte: GBR + disp Word: GBR + disp × 2 Longword: GBR + disp × 4

Addressing Mode	Instruction Format	Effective Address Calculation Method	Calculation Formula
Indexed GBR indirect	@(R0, GBR)	Effective address is sum of register GBR and R0 contents. 	GBR + R0
PC-relative with displacement	@(disp:8, PC)	Effective address is PC with 8-bit displacement disp added. After disp is zero-extended, it is multiplied by 2 (word) or 4 (longword), according to the operand size. With a longword operand, the lower 2 bits of PC are masked. 	Word: PC + disp × 2 Longword: PC&H'FFFFFFFC + disp × 4

Addressing Mode	Instruction Format	Effective Address Calculation Method	Calculation Formula
PC-relative	disp:8	Effective address is PC with 8-bit displacement disp added after being sign-extended and multiplied by 2. 	PC + disp × 2
	disp:12	Effective address is PC with 12-bit displacement disp added after being sign-extended and multiplied by 2. 	PC + disp × 2
	Rn	Effective address is sum of PC and Rn. 	PC + Rn
Immediate	#imm:8	8-bit immediate data imm of TST, AND, OR, or XOR instruction is zero-extended.	—
	#imm:8	8-bit immediate data imm of MOV, ADD, or CMP/EQ instruction is sign-extended.	—
	#imm:8	8-bit immediate data imm of TRAPA instruction is zero-extended and multiplied by 4.	—

2.4.2 DSP Data Addressing

Two different memory accesses are made with DSP instructions. The two kinds of instructions are X and Y data transfer instructions (MOVX.W, MOVY.W) and single data transfer instructions (MOVS.W, MOVSL). The data addressing is different for these two kinds of instruction. An overview of the data transfer instructions is given in table 2.13.

Table 2.13 Overview of Data Transfer Instructions

	X/Y Data Transfer Processing (MOVX.W, MOVY.W)	Single Data Transfer Processing (MOVS.W, MOVS.L)
Address register	Ax: R4, R5, Ay: R6, R7	As: R2, R3, R4, R5
Index register	Ix: R8, Iy: R9	Is: R8
Addressing	Nop/Inc (+2)/index addition: post-increment	Nop/Inc (+2, +4)/index addition: post-increment
	—	Dec (–2, –4): pre-decrement
Modulo addressing	Possible	Not possible
Data bus	XDB, YDB	LDB
Data length	16 bits (word)	16/32 bits (word/longword)
Bus contention	No	Yes
Memory	X/Y data memory	Entire memory space
Source register	Dx, Dy: A0, A1	Ds: A0/A1, M0/M1, X0/X1, Y0/Y1, A0G, A1G
Destination register	Dx: X0/X1, Dy: Y0/Y1	Ds: A0/A1, M0/M1, X0/X1, Y0/Y1, A0G, A1G

X/Y Data Addressing: With DSP instructions, the X and Y data memory can be accessed simultaneously using the MOVX.W and MOVY.W instructions. Two address pointers are provided for DSP instructions to enable simultaneous access to X and Y data memory. Only pointer addressing can be used with DSP instructions; immediate addressing is not available. Address registers are divided into two, with register R4 or R5 functioning as the X memory address register (Ax), and register R6 or R7 as the Y memory address register (Ay). The following three kinds of addressing can be used with X and Y data transfer instructions.

1. Non-update address register addressing:
   The Ax and Ay registers are address pointers. They are not updated.

2. Addition index register addressing:

   The Ax and Ay registers are address pointers. After a data transfer, the value of the Ix or Iy register is added to each (post-increment).

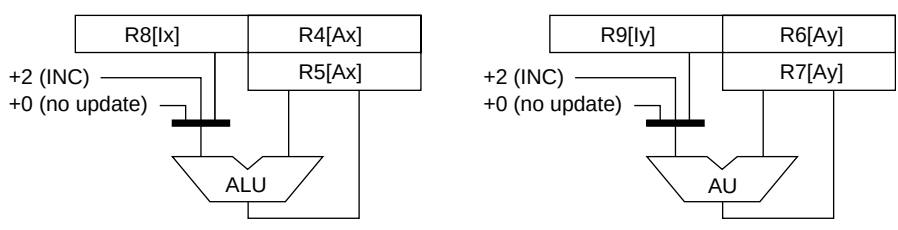
3. Increment address register addressing:

   The Ax and Ay registers are address pointers. After a data transfer, they are each incremented by 2 (post- increment).

There is an index register for each address pointer. The R8 register is the index register (Ix) for the X memory address register (Ax), and the R9 register is the index register (Iy) for the Y memory address register (Ay).

The X and Y data transfer instructions perform word-length processing, and use 16-bit access to the X/Y data memory. A value of 2 is therefore added to the address register in the increment processing. To perform decrementing, –2 is set in the index register and addition index register addressing is specified. In X/Y data addressing, only bits 1 to 15 of the address pointer are valid. When using X/Y data addressing, 0 must always be written to bit 0 of the address pointer and index register.

X/Y data transfer addressing is shown in figure 2.12. When accessing X and Y memory using the X and Y buses, the upper word of Ax (R4 or R5) and Ay (R6 or R7) is ignored. The result of @AY+ or @Ay + Iy is stored in the lower word of Ay, while the upper word retains its original value.



AU: Adder provided for DSP addressing

Note: Three address processing methods:

1. Increment
2. Index register addition (Ix/Iy)
3. No increment

Post-updating is used in all cases.
The address pointer can be decremented by setting –2/–4 in the index register.

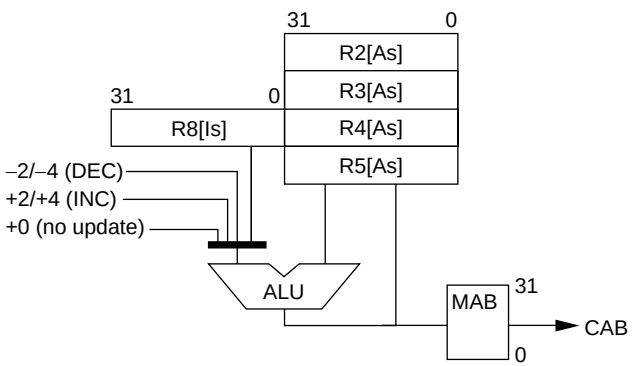
Figure 2.12 X and Y Data Transfer Addressing

Single Data Addressing: DSP instructions include two single data transfer instructions (MOVS.W, MOVS.L) that load data into, or store data from, a DSP register. With these instructions, one of registers R2 to R5 is used as the single data transfer address register (As).

The following four kinds of addressing can be used with single data transfer instructions.

1. Non-update address register addressing:
   The As register is an address pointer. It is not updated.
2. Addition index register addressing:
   The As register is an address pointer. After a data transfer, the value of the Is register is added to the As register (post-increment).
3. Increment address register addressing:
   The As register is an address pointer. After a data transfer, the As register is incremented by 2 or 4 (post-increment).
4. Decrement address register addressing:
   The As register is an address pointer. Before a data transfer, –2 or –4 is added to the As register (i.e. 2 or 4 is subtracted) (pre-decrement).

The R8 register is the index register (Is) for the address pointer (As). Single data transfer addressing is shown in figure 2.13.



Note: Four address processing methods:

1. No update
2. Index register addition (Is) — Post-increment
3. Increment — Post-increment
4. Decrement — Pre-decrement

Figure 2.13 Single Data Transfer Addressing

Modulo Addressing: Like other DSPs, the SH7727 has a modulo addressing mode. Address registers are updated in the same way in this mode. When the address pointer value reaches the preset modulo end address, the address pointer value becomes the modulo start address.

Modulo addressing is only available for the X and Y data transfer instructions (MOVX.W, MOVY.W). Modulo addressing mode is specified for the X address register by setting the DMX bit in the SR register, and for the Y address register by setting the DMY bit. Modulo addressing is valid for either the X or the Y address register, only; it cannot be set for both at the same time. Therefore, DMX and DMY cannot both be set simultaneously (if they are, the DMY setting will be valid).

The MOD register is provided to set the start and end addresses of the modulo address area. The MOD register contains MS (Modulo Start) and ME (Modulo End). An example of the use of the MOD register (MS and ME fields) is shown below.

```
          MOV.L ModAddr,Rn;      Rn=ModEnd, ModStart
          LDC Rn,MOD;            ME=ModEnd, MS=ModStart
ModAddr:  .DATA.W                mEnd; ModEnd
          .DATA.W                mStart;     ModStart


ModStart: .DATA
            :
ModEnd:   .DATA
```

The start and end addresses are specified in MS and ME, then the DMX or DMY bit is set to 1. The address register contents are compared with ME, and if they match, start address MS is stored in the address register. The lower 16 bits of the address register are compared with ME.

The maximum modulo size is 64 kbytes. This is sufficient to access the X and Y data memory. A block diagram of modulo addressing is shown in figure 2.14.

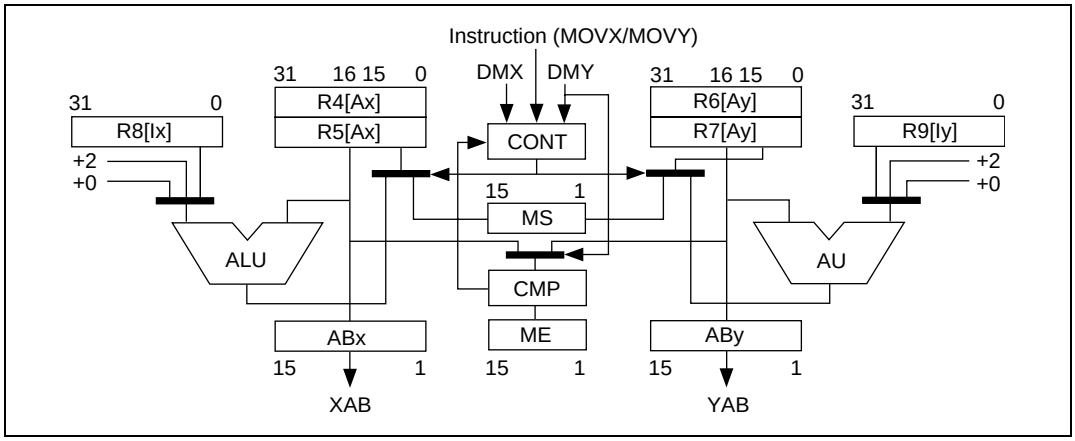


Figure 2.14 Modulo Addressing

An example of modulo addressing is given below.

```
MS = H'7008; ME=H'700C; R4=H'A5007008;
DMX = 1; DMY = 0:        (Modulo addressing setting for address register Ax (R4, R5))
```

As a result of the above settings, the R4 register changes as follows.

```
        R4: H'A5007008
Inc.    R4: H'A500700A
Inc.    R4: H'A500700C
Inc.    R4: H'A5007008  (Reaches modulo end address, so becomes modulo start address)
```

Place the data so that the upper 16 bits of the modulo start and end addresses are the same. This is because the modulo start address overwrites only the lower 15 bits of the address register, excluding bit 0.

Note: When addition indexing is used for DSP data addressing, the address pointer may exceed the ME value without actually reaching it. In this case, the address pointer will not return to the modulo start address. Not only with modulo addressing, but when X and Y data addressing is used, bit 0 is ignored. 0 must always be written to bit 0 of the address pointer, index register, MS, and ME.

DSP Addressing Operations: DSP addressing operations in the pipeline execution stage (EX), including modulo addressing, are shown below.

```
if ( Operation is MOVX.W MOVY.W ) {
   ABx=Ax; ABy=Ay;
   /* memory access cycle uses ABx and ABy. The addresses to be used
have not been updated */

   /* Ax is one of R4,R5 */
   if ( DMX==0 || DMX==1 && DMY == 1 )} Ax=Ax+(+2 or R8[Ix] or +0);
   /* Inc,Index,Not-Update */
   else if (! not-update) Ax=modulo( Ax, (+2 or R8[Ix]) );

   /* Ay is one of R6,R7 */
   if ( DMY==0 ) Ay=Ay+(+2 or R9[Iy] or +0); /* Inc,Index,Not-Update */
   else if (! not-update) Ay=modulo( Ay, (+2 or R9[Iy]) );
}
else if ( Operation is MOVS.W or MOVS.L ) {
   if ( Addressing is Nop, Inc, Add-index-reg ) {
      MAB=As;
      /* memory access cycle uses MAB. The address to be used has not
been updated */
      /* As is one of R2 to R5 */
      As=As+(+2 or +4 or R8[Is] or +0); /* Inc,Index,Not-Update */
   else { /* Decrement, Pre-update */
      /* As is one of R2 to R5 */
      As=As+(-2 or -4);
      MAB=As;
      /* memory access cycle uses MAB. The address to be used has been
updated */
}
```

```
/* The value to be added to the address register depends on addressing
operations.
For example, (+2 or R8[Ix] or +0) means that
       +2 : if operation is increment
   R8[Ix] : if operation is add-index-reg
       +0 : if operation is not-update
*/

function modulo ( AddrReg, Index ) {
   if ( AdrReg[15:0]==ME ) AdrReg[15:0]==MS;
   else AdrReg=AdrReg+Index;
   return AddrReg;
}
```

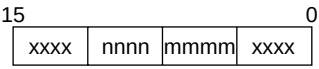
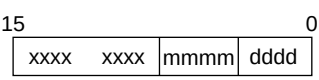
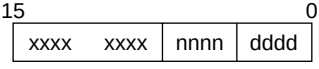
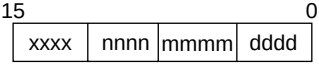
2.4.3 CPU Instruction Formats

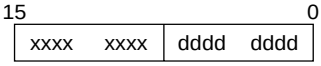
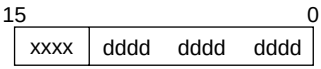
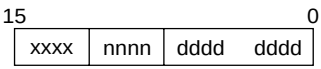
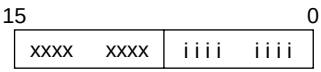
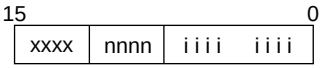
Table 2.14 shows the instruction formats, and the meaning of the source and destination operands, for instructions executed by the CPU core. The meaning of the operands depends on the instruction code. The following symbols are used in the table.

xxxx: Instruction code
mmmm: Source register
nnnn: Destination register
iiii: Immediate data
dddd: Displacement

Table 2.14 CPU Instruction Formats

Instruction Format	Source Operand	Destination Operand	Sample Instruction	
0 type 15 0 xxxx xxxx xxxx xxxx	—	—	NOP	
n type 15 0 xxxx nnnn xxxx xxxx	—	nnnn: register direct	MOVT	Rn
	Control register or system register	nnnn: register direct	STS	MACH,Rn
	Control register or system register	nnnn: pre-decrement register indirect	STC.L	SR,@-Rn
m type 15 0 xxxx mmmm xxxx xxxx	mmmm: register direct	Control register or system register	LDC	Rm,SR
	mmmm: post-increment register indirect	Control register or system register	LDC.L	@Rm+,SR
	mmmm: register indirect	—	JMP	@Rm
	PC-relative using Rm	—	BRAF	Rm

Instruction Format	Source Operand	Destination Operand	Sample Instruction
nm type 	mmmm: register direct	nnnn: register direct	ADD Rm,Rn
	mmmm: register indirect	nnnn: register indirect	MOV.L Rm,@Rn
	mmmm: post-increment register indirect (multiply-and-accumulate operation) nnnn: * post-increment register indirect (multiply-and-accumulate operation)	MACH, MACL	MAC.W @Rm+,@Rn+
	mmmm: post-increment register indirect	nnnn: register direct	MOV.L @Rm+,Rn
	mmmm: register direct	nnnn: pre-decrement register indirect	MOV.L Rm,@-Rn
	mmmm: register direct	nnnn: indexed register indirect	MOV.L Rm,@(R0,Rn)
md type 	mmmmdddd: register indirect with displacement	R0 (register direct)	MOV.B @(disp,Rm),R0
nd4 type 	R0 (register direct)	nnnndddd: register indirect with displacement	MOV.B R0,@(disp,Rn)
nmd type 	mmmm: register direct	nnnndddd: register indirect with displacement	MOV.L Rm,@(disp,Rn)
	mmmmdddd: register indirect with displacement	nnnn: register direct	MOV.L @(disp,Rm),Rn

Instruction Format	Source Operand	Destination Operand	Sample Instruction
d type 	dddddddd: GBR indirect with displacement	R0 (register direct)	MOV.L @@(disp,GBR),R0
	R0 (register direct)	dddddddd: GBR indirect with displacement	MOV.L @R0,@(disp,GBR)
	dddddddd: PC-relative with displacement	R0 (register direct)	MOVA @(disp,PC),R0
	dddddddd: PC-relative	—	BF label
d12 type 	dddddddddddd: PC-relative	—	BRA label (label=disp+PC)
nd8 type 	dddddddd: PC-relative with displacement	nnnn: register direct	MOV.L @(disp,PC),Rn
i type 	iiiiiiii: immediate	Indexed GBR indirect	AND.B #imm,@(R0,GBR)
	iiiiiiii: immediate	R0 (register direct)	AND #imm,R0
	iiiiiiii: immediate	—	TRAPA #imm
ni type 	iiiiiiii: immediate	nnnn: register direct	ADD #imm,Rn

Note: * In multiply-and-accumulate instructions, nnnn is the source register.

2.4.4 DSP Instruction Formats

The SH7727 includes new instructions for digital signal processing. The new instructions are of the following two kinds.

1. Memory and DSP register double and single data transfer instructions (16-bit length)
2. Parallel processing instructions processed by the DSP unit (32-bit length)

The instruction formats are shown in figure 2.15.

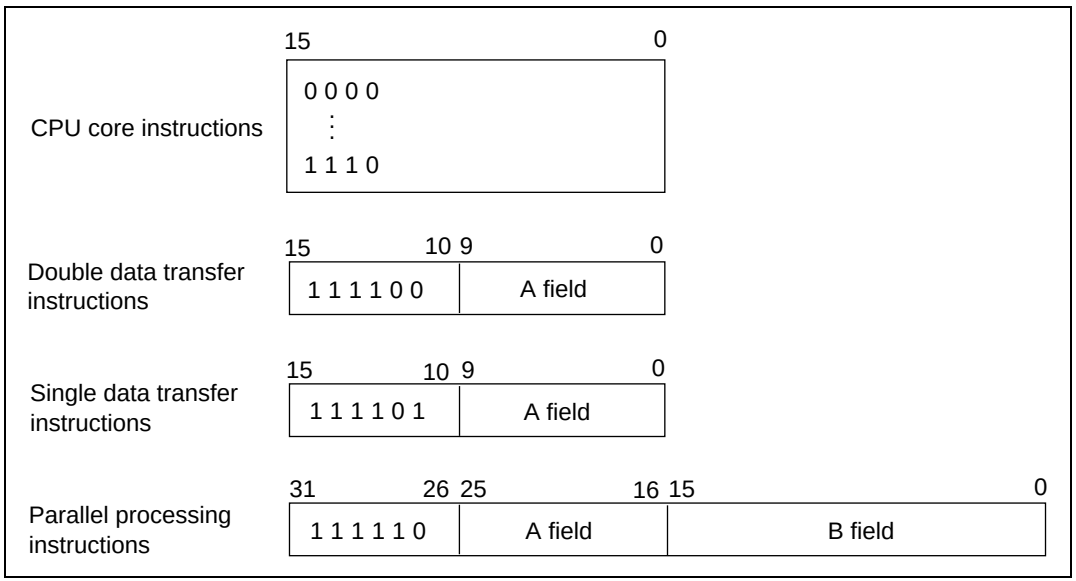


Figure 2.15 DSP Instruction Formats

Double and Single Data Transfer Instructions: The format of double data transfer instructions is shown in table 2.15, and that of single data transfer instructions in table 2.16.

Table 2.15 Double Data Transfer Instruction Formats

Type	Mnemonic	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
X memory	NOPX	1	1	1	1	0	0	0		0		0		0	0		
data	MOVX.W @Ax,Dx							Ax		Dx		0		0	1		
transfer	MOVX.W @Ax+,Dx													1	0		
	MOVX.W @Ax+Ix,Dx													1	1		
	MOVX.W Da,@Ax									Da		1		0	1		
	MOVX.W Da,@Ax+													1	0		
	MOVX.W Da,@Ax+Ix													1	1		
Y memory	NOPY	1	1	1	1	0	0		0		0		0			0	0
data	MOVY.W @Ay,Dy								Ay		Dy		0			0	1
transfer	MOVY.W @Ay+,Dy															1	0
	MOVY.W @Ay+Iy,Dy															1	1
	MOVY.W Da,@Ay										Da		1			0	1
	MOVY.W Da,@Ay+															1	0
	MOVY.W Da,@Ay+Iy															1	1

Note: Ax: 0 = R4, 1 = R5
Ay: 0 = R6, 1 = R7
Dx: 0 = X0, 1 = X1
Dy: 0 = Y0, 1 = Y1
Da: 0 = A0, 1 = A1

Table 2.16 Single Data Transfer Instruction Formats

Type	Mnemonic	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Single	MOVS.W @-As,Ds	1	1	1	1	0	1	As		Ds		0:(*)		0	0	0	0
data	MOVS.W @As,Ds							0:R4				1:(*)		0	1		
transfer	MOVS.W @As+,Ds							1:R5				2:(*)		1	0		
	MOVS.W @As+Is,Ds							2:R2				3:(*)		1	1		
	MOVS.W Ds,@-As							3:R3				4:(*)		0	0	0	1
	MOVS.W Ds,@As											5:A1		0	1		
	MOVS.W Ds,@As+											6:(*)		1	0		
	MOVS.W Ds,@As+Is											7:A0		1	1		
	MOVS.L @-As,Ds											8:X0		0	0	1	0
	MOVS.L @As,Ds											9:X1		0	1		
	MOVS.L @As+,Ds											A:Y0		1	0		
	MOVS.L @As+Is,Ds											B:Y1		1	1		
	MOVS.L Ds,@-As											C:M0		0	0	1	1
	MOVS.L Ds,@As											D:A1G		0	1		
	MOVS.L Ds,@As+											E:M1		1	0		
	MOVS.L Ds,@As+Is											F:A0G		1	1		

Note: * Codes reserved for system use.

Parallel Processing Instructions: Parallel processing instructions are provided for efficient execution of digital signal processing using the DSP unit. They are 32 bits long and allow four simultaneous processes, an ALU operation, multiplication, and two data transfers.

Parallel processing instructions are divided into an A field and a B field. The A field defines data transfer instructions and the B field an ALU operation instruction and multiply instruction. These instructions can be defined independently, and the processing is executed in parallel, independently and simultaneously. A-field parallel data transfer instructions are shown in table 2.17, and B-field ALU operation instructions and multiply instructions in table 2.18.

Table 2.17 A-Field Parallel Data Transfer Instructions

Type	Mnemonic	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
X memory data transfer	NOPX	1	1	1	1	1	0	0		0		0		0	0			B field															
	MOVX.W @Ax, Dx							Ax		Dx		0		0	1																		
	MOVX.W @Ax+, Dx													1	0																		
	MOVX.W @Ax+Ix, Dx													1	1																		
	MOVX.W Da, @Ax									Da		1		0	1																		
	MOVX.W Da, @Ax+													1	0																		
	MOVX.W Da, @Ax+Ix													1	1																		
Y memory data transfer	NOPY	1	1	1	1	1	0		0		0		0			0	0	B field															
	MOVY.W @Ay, Dy								Ay		Dy		0			0	1																
	MOVY.W @Ay+, Dy															1	0																
	MOVY.W @Ay+Iy, Dy															1	1																
	MOVY.W Da, @Ay										Da		1			0	1																
	MOVY.W Da, @Ay+															1	0																
	MOVY.W Da, @Ay+Iy															1	1																

Note: Ax: 0 = R4, 1 = R5
Ay: 0 = R6, 1 = R7
Dx: 0 = X0, 1 = X1
Dy: 0 = Y0, 1 = Y1
Da: 0 = A0, 1 = A1

Table 2.18 B-Field ALU Operation Instructions and Multiply Instructions

Type	Mnemonic	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
imm. shift	PSHL #imm, Dz	1	1	1	1	1	0	A field										0	0	0	0	0	0	−16 <= imm <= +16						Dz			
	PSHA #imm, Dz																	0	0	0	0	1	0	−32 <= imm <= +32									
	Reserved																	0	0	0		1											
																		0	0	1													
6-operand parallel instructions	PMULS Se, Sf, Dg																	0	1	0	0		Se	Sf		Sx		Sy		Dg		Du	
	Reserved																	0	1	0	1		0:X0	0:Y0		0:X0		0:Y0		0:M0		0:X0	
																							1:X1	1:Y1		1:X1		1:Y1		1:M1		1:Y0	
	PSUB Sx, Sy, Du																	0	1	1	0		2:Y0	2:X0		2:A0		2:M0		2:A0		2:A0	
	PMULS Se, Sf, Dg																						3:A1	3:A1		3:A1		3:M1		3:A1		3:A1	
	PADD Sx, Sy, Du																	0	1	1	1												
	PMULS Se, Sf, Dg																																
3-operand instructions	Reserved																	1	0	0	0	0	0	0	0					Dz			
																					0	1								0:(*1)			
	PSUBC Sx, Sy, Dz																				1	0								1:(*1)			
	PADDC Sx, Sy, Dz																				1	1								2:(*1)			
	PCMP Sx, Sy																				0	0	0	1						3:(*1)			
	Reserved																				0	1								4:(*1)			
	Reserved																				1	0								5:A1			
	Reserved																				1	1								6:(*1)			
	PABS Sx, Dz																				0	0	1	0						7:A0			
	PRND Sx, Dz																				0	1								8:X0			
	PABS Sy, Dz																				1	0								9:X1			
	PRND Sy, Dz																				1	1								A:Y0			
	Reserved																				0	0	1	1						B:Y1			
																					0	1								C:M0			
																					1	0								D:(*1)			
																					1	1								E:M1			
																														F:(*1)			

Note: 1. Codes reserved for system use.

Type	Mnemonic	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Conditional 3-operand instructions	[if cc] PSHL Sx, Sy, Dz	1	1	1	1	1	0	A field												0	0	0	0	if cc		Sx		Sy		Dz			
	[if cc] PSHA Sx, Sy, Dz																			0	1					0:X0 1:X1 2:A0 3:A1		0:Y0 1:Y1 2:M0 3:M1		0:(*1) 1:(*1) 2:(*1) 3:(*1) 4:(*1) 5:A1 6:(*1) 7:A0 8:X0 9:X1 A:Y0 B:Y1 C:M0 D:(*1) E:M1 F:(*1)			
	[if cc] PSUB Sx, Sy, Dz																			1	0												
	[if cc] PADD Sx, Sy, Dz																			1	1			01: Uncon-ditional									
	Reserved																			0	0	0	1										
	[if cc] PAND Sx, Sy, Dz																			0	1												
	[if cc] PXOR Sx, Sy, Dz																			1	0												
	[if cc] POR Sx, Sy, Dz																			1	1			10: DCT									
	[if cc] PDEC Sx, Dz																			0	0	1	0										
	[if cc] PINC Sx, Dz																			0	1												
	[if cc] PDEC Sy, Dz																			1	0												
	[if cc] PINC Sy, Dz																			1	1			11: DCF									
	[if cc] PCLR Dz																			0	0	1	1										
	[if cc] PDMSB Sx, Dz																			0	1												
	Reserved																			1	0												
	[if cc] PDMSB Sy, Dz																			1	1												
	[if cc] PNEG Sx, Dz																	1	1	0	0	1	0										
	[if cc] PCOPY Sx, Dz																			0	1												
	[if cc] PNEG Sy, Dz																			1	0												
	[if cc] PCOPY Sy, Dz																			1	1												
	Reserved																							0	0								
	[if cc] PSTS MACH, Dz																			0	0	1	1	if cc									
	[if cc] PSTS MACL, Dz																			0	1												
	[if cc] PLDS Dz, MACH																			1	0												
	[if cc] PLDS Dz, MACL																			1	1												
	(*2) Reserved																					0	0	0	0								
																						0	*										
	Reserved	1	1	1	1	1	1																										

Notes: 1. Codes reserved for system use.
2. [if cc]: DCT (DC bit True), DCF (DC bit False) or none (unconditional instruction)

2.5 Instruction Set

2.5.1 CPU Instruction Set

The SH-1/SH-2/SH-3 compatible instruction set consists of 68 basic instruction types divided into six functional groups, as shown in table 2.19. Tables 2.20 to 2.25 show the instruction notation, machine code, execution time, and function.

Table 2.19 CPU Instruction Types

Type	Kinds of Instruction	Op Code	Function	Number of Instructions
Data transfer instructions	5	MOV	Data transfer	39
			Immediate data transfer	
			Peripheral module data transfer	
			Structure data transfer	
		MOVA	Effective address transfer	
		MOVT	T bit transfer	
		SWAP	Upper/lower swap	
		XTRCT	Extraction of middle of linked registers	
Arithmetic operation instructions	21	ADD	Binary addition	33
		ADDC	Binary addition with carry	
		ADDV	Binary addition with overflow check	
		CMP/cond	Comparison	
		DIV1	Division	
		DIV0S	Signed division initialization	
		DIV0U	Unsigned division initialization	
		DMULS	Signed double-precision multiplication	
		DMULU	Unsigned double-precision multiplication	
		DT	Decrement and test	
		EXTS	Sign extension	
		EXTU	Zero extension	
		MAC	Multiply-and-accumulate, double-precision multiply-and-accumulate	
		MUL	Double-precision multiplication (32×32 bits)	
		MULS	Signed multiplication (16×16 bits)	
		MULU	Unsigned multiplication (16×16 bits)	
		NEG	Sign inversion	
		NEGC	Sign inversion with borrow	
		SUB	Binary subtraction	
		SUBC	Binary subtraction with carry	
		SUBV	Binary subtraction with underflow	

Type	Kinds of Instruction	Op Code	Function	Number of Instructions
Logic operation instructions	6	AND	Logical AND	14
		NOT	Bit inversion	
		OR	Logical OR	
		TAS	Memory test and bit setting	
		TST	Logical AND and T bit setting	
		XOR	Exclusive logical OR	
Shift instructions	12	ROTL	1-bit left shift	16
		ROTR	1-bit right shift	
		ROTCL	1-bit left shift with T bit	
		ROTCR	1-bit right shift with T bit	
		SHAL	Arithmetic 1-bit left shift	
		SHAR	Arithmetic 1-bit right shift	
		SHLL	Logical 1-bit left shift	
		SHLLn	Logical n-bit left shift	
		SHLR	Logical 1-bit right shift	
		SHLRn	Logical n-bit right shift	
		SHAD	Arithmetic dynamic shift	
		SHLD	Logical dynamic shift	
Branch instructions	9	BF	Conditional branch, delayed conditional branch (T = 0)	11
		BT	Conditional branch, delayed conditional branch (T = 1)	
		BRA	Unconditional branch	
		BRAF	Unconditional branch	
		BSR	Branch to subroutine procedure	
		BSRF	Branch to subroutine procedure	
		JMP	Unconditional branch	
		JSR	Branch to subroutine procedure	
		RTS	Return from subroutine procedure	

Type	Kinds of Instruction	Op Code	Function	Number of Instructions
System control instructions	15	CLRT	T bit clear	75
		CLRMAC	MAC register clear	
		CLRS	S bit clear	
		LDC	Load into control register	
		LDS	Load into system register	
		LDTLB	PTEH/PTEL load into TLB	
		NOP	No operation	
		PREF	Data prefetch to cache	
		RTE	Return from exception handling	
		SETS	S bit setting	
		SETT	T bit setting	
		SLEEP	Transition to power-down mode	
		STC	Store from control register	
		STS	Store from system register	
		TRAPA	Trap exception handling	
Total:	68			189

The instruction code, operation, and number of execution states of the CPU instructions are shown in the following tables, classified by instruction type, using the format shown below.

Instruction	Instruction Code	Operation	Privilege	Execution States	T Bit
Indicated by mnemonic.	Indicated in MSB ↔ LSB order.	Indicates summary of operation.	Indicates a privileged instruction.	Value when no wait states are inserted.*1	Value of T bit after instruction is executed.
Explanation of Symbols OP.Sz SRC, DEST OP: Operation code Sz: Size SRC: Source DEST: Destination Rm: Source register Rn: Destination register imm: Immediate data disp: Displacement	Explanation of Symbols mmmm: Source register nnnn: Destination register 0000: R0 0001: R1 ..... 1111: R15 iiii: Immediate data dddd: Displacement*2	Explanation of Symbols →, ←: Transfer direction (xx): Memory operand M/Q/T: Flag bits in the SR &: Logical AND of each bit |: Logical OR of each bit ^: Exclusive logical OR of each bit ~: Logical NOT of each bit <<n: n-bit left shift >>n: n-bit right shift			Explanation of Symbols —: No change

Notes: 1. The table shows the minimum number of execution states. In practice, the number of instruction execution states will be increased in cases such as the following:
(1) When there is contention between an instruction fetch and a data access
(2) When the destination register of a load instruction (memory → register) is also used by the following instruction
2. Scaled (x1, x2, or x4) according to the instruction operand size, etc.

Data Transfer Instructions

Table 2.20 Data Transfer Instructions

Instruction		Operation	Code	Privileged Mode	Cycles	T Bit
MOV	#imm,Rn	imm → Sign extension → Rn	1110nnnniiiiiiii	—	1	—
MOV.W	@(disp,PC),Rn	(disp × 2 + PC) → Sign extension → Rn	1001nnnndddddddd	—	1	—
MOV.L	@(disp,PC),Rn	(disp × 4 + PC) → Rn	1101nnnndddddddd	—	1	—
MOV	Rm,Rn	Rm → Rn	0110nnnnmmmm0011	—	1	—
MOV.B	Rm,@Rn	Rm → (Rn)	0010nnnnmmmm0000	—	1	—
MOV.W	Rm,@Rn	Rm → (Rn)	0010nnnnmmmm0001	—	1	—
MOV.L	Rm,@Rn	Rm → (Rn)	0010nnnnmmmm0010	—	1	—
MOV.B	@Rm,Rn	(Rm) → Sign extension → Rn	0110nnnnmmmm0000	—	1	—
MOV.W	@Rm,Rn	(Rm) → Sign extension → Rn	0110nnnnmmmm0001	—	1	—
MOV.L	@Rm,Rn	(Rm) → Rn	0110nnnnmmmm0010	—	1	—
MOV.B	Rm,@–Rn	Rn–1 → Rn, Rm → (Rn)	0010nnnnmmmm0100	—	1	—
MOV.W	Rm,@–Rn	Rn–2 → Rn, Rm → (Rn)	0010nnnnmmmm0101	—	1	—
MOV.L	Rm,@–Rn	Rn–4 → Rn, Rm → (Rn)	0010nnnnmmmm0110	—	1	—
MOV.B	@Rm+,Rn	(Rm) → Sign extension → Rn, Rm + 1 → Rm	0110nnnnmmmm0100	—	1	—
MOV.W	@Rm+,Rn	(Rm) → Sign extension → Rn, Rm + 2 → Rm	0110nnnnmmmm0101	—	1	—
MOV.L	@Rm+,Rn	(Rm) → Rn,Rm + 4 → Rm	0110nnnnmmmm0110	—	1	—
MOV.B	R0,@(disp,Rn)	R0 → (disp + Rn)	10000000nnnndddd	—	1	—
MOV.W	R0,@(disp,Rn)	R0 → (disp × 2 + Rn)	10000001nnnndddd	—	1	—
MOV.L	Rm,@(disp,Rn)	Rm → (disp × 4 + Rn)	0001nnnnmmmmdddd	—	1	—
MOV.B	@(disp,Rm),R0	(disp + Rm) → Sign extension → R0	10000100mmmmdddd	—	1	—
MOV.W	@(disp,Rm),R0	(disp × 2 + Rm) → Sign extension → R0	10000101mmmmdddd	—	1	—
MOV.L	@(disp,Rm),Rn	(disp × 4 + Rm) → Rn	0101nnnnmmmmdddd	—	1	—
MOV.B	Rm,@(R0,Rn)	Rm → (R0 + Rn)	0000nnnnmmmm0100	—	1	—

Instruction		Operation	Code	Privileged Mode	Cycles	T Bit
MOV.W	Rm,@(R0,Rn)	Rm → (R0 + Rn)	0000nnnnmmmm0101	—	1	—
MOV.L	Rm,@(R0,Rn)	Rm → (R0 + Rn)	0000nnnnmmmm0110	—	1	—
MOV.B	@(R0,Rm),Rn	(R0 + Rm) → Sign extension → Rn	0000nnnnmmmm1100	—	1	—
MOV.W	@(R0,Rm),Rn	(R0 + Rm) → Sign extension → Rn	0000nnnnmmmm1101	—	1	—
MOV.L	@(R0,Rm),Rn	(R0 + Rm) → Rn	0000nnnnmmmm1110	—	1	—
MOV.B	R0,@(disp,GBR)	R0 → (disp + GBR)	11000000dddddddd	—	1	—
MOV.W	R0,@(disp,GBR)	R0 → (disp × 2 + GBR)	11000001dddddddd	—	1	—
MOV.L	R0,@(disp,GBR)	R0 → (disp × 4 + GBR)	11000010dddddddd	—	1	—
MOV.B	@(disp,GBR),R0	(disp + GBR) → Sign extension → R0	11000100dddddddd	—	1	—
MOV.W	@(disp,GBR),R0	(disp × 2 + GBR) → Sign extension → R0	11000101dddddddd	—	1	—
MOV.L	@(disp,GBR),R0	(disp × 4 + GBR) → R0	11000110dddddddd	—	1	—
MOVA	@(disp,PC),R0	disp × 4 + PC → R0	11000111dddddddd	—	1	—
MOVT	Rn	T → Rn	0000nnnn00101001	—	1	—
SWAP.B	Rm,Rn	Rm → Swap the bottom two bytes → REG	0110nnnnmmmm1000	—	1	—
SWAP.W	Rm,Rn	Rm → Swap two consecutive words → Rn	0110nnnnmmmm1001	—	1	—
XTRCT	Rm,Rn	Rm: Middle 32 bits of Rn → Rn	0010nnnnmmmm1101	—	1	—

Arithmetic Operation Instructions

Table 2.21 Arithmetic Operation Instructions

Instruction		Operation	Code	Privileged Mode	Cycles	T Bit
ADD	Rm,Rn	Rn + Rm → Rn	0011nnnnmmmm1100	—	1	—
ADD	#imm,Rn	Rn + imm → Rn	0111nnnniiiiiiii	—	1	—
ADDC	Rm,Rn	Rn + Rm + T → Rn, Carry → T	0011nnnnmmmm1110	—	1	Carry
ADDV	Rm,Rn	Rn + Rm → Rn, Overflow → T	0011nnnnmmmm1111	—	1	Overflow
CMP/EQ	#imm,R0	If R0 = imm, 1 → T	10001000iiiiiiii	—	1	Comparison result
CMP/EQ	Rm,Rn	If Rn = Rm, 1 → T	0011nnnnmmmm0000	—	1	Comparison result
CMP/HS	Rm,Rn	If Rn ≥ Rm with unsigned data, 1 → T	0011nnnnmmmm0010	—	1	Comparison result
CMP/GE	Rm,Rn	If Rn ≥ Rm with signed data, 1 → T	0011nnnnmmmm0011	—	1	Comparison result
CMP/HI	Rm,Rn	If Rn > Rm with unsigned data, 1 → T	0011nnnnmmmm0110	—	1	Comparison result
CMP/GT	Rm,Rn	If Rn > Rm with signed data, 1 → T	0011nnnnmmmm0111	—	1	Comparison result
CMP/PZ	Rn	If Rn ≥ 0, 1 → T	0100nnnn00010001	—	1	Comparison result
CMP/PL	Rn	If Rn > 0, 1 → T	0100nnnn00010101	—	1	Comparison result
CMP/STR	Rm,Rn	If Rn and Rm have an equivalent byte, 1 → T	0010nnnnmmmm1100	—	1	Comparison result
DIV1	Rm,Rn	Single-step division (Rn/Rm)	0011nnnnmmmm0100	—	1	Calculation result
DIV0S	Rm,Rn	MSB of Rn → Q, MSB of Rm → M, M ^ Q → T	0010nnnnmmmm0111	—	1	Calculation result
DIV0U		0 → M/Q/T	0000000000011001	—	1	0
DMULS.L	Rm,Rn	Signed operation of Rn × Rm → MACH, MACL 32 × 32 → 64 bits	0011nnnnmmmm1101	—	2 (to 5)*1	—

Instruction		Operation	Code	Privileged Mode	Cycles	T Bit
DMULU.L	Rm,Rn	Unsigned operation of Rn × Rm → MACH, MACL 32 × 32 → 64 bits	0011nnnnmmmm0101	—	2 (to 5)*1	—
DT	Rn	Rn – 1 → Rn, if Rn = 0, 1 → T, else 0 → T	0100nnnn00010000	—	1	Comparison result
EXTS.B	Rm,Rn	A byte in Rm is sign-extended → Rn	0110nnnnmmmm1110	—	1	—
EXTS.W	Rm,Rn	A word in Rm is sign-extended → Rn	0110nnnnmmmm1111	—	1	—
EXTU.B	Rm,Rn	A byte in Rm is zero-extended → Rn	0110nnnnmmmm1100	—	1	—
EXTU.W	Rm,Rn	A word in Rm is zero-extended → Rn	0110nnnnmmmm1101	—	1	—
MAC.L	@Rm+,@Rn+	Signed operation of (Rn) × (Rm) + MAC → MAC, Rn + 4 → Rn, Rm + 4 → Rm, 32 × 32 + 64 → 64 bits	0000nnnnmmmm1111	—	2 (to 5)*1	—
MAC.W	@Rm+,@Rn+	Signed operation of (Rn) × (Rm) + MAC → MAC, Rn + 2 → Rn, Rm + 2 → Rm, 16 × 16 + 64 → 64 bits	0100nnnnmmmm1111	—	2 (to 5)*1	—
MUL.L	Rm,Rn	Rn × Rm → MACL, 32 × 32 → 32 bits	0000nnnnmmmm0111	—	2 (to 5)*1	—
MULS.W	Rm,Rn	Signed operation of Rn × Rm → MACL, 16 × 16 → 32 bits	0010nnnnmmmm1111	—	1 (to 3)*2	—
MULU.W	Rm,Rn	Unsigned operation of Rn × Rm → MACL, 16 × 16 → 32 bits	0010nnnnmmmm1110	—	1 (to 3)*2	—
NEG	Rm,Rn	0–Rm → Rn	0110nnnnmmmm1011	—	1	—
NEGC	Rm,Rn	0–Rm–T → Rn, Borrow → T	0110nnnnmmmm1010	—	1	Borrow
SUB	Rm,Rn	Rn–Rm → Rn	0011nnnnmmmm1000	—	1	—
SUBC	Rm,Rn	Rn–Rm–T → Rn, Borrow → T	0011nnnnmmmm1010	—	1	Borrow
SUBV	Rm,Rn	Rn–Rm → Rn, Underflow → T	0011nnnnmmmm1011	—	1	Underflow

Notes: 1. The normal minimum number of execution cycles is two, but five cycles are required when the operation result is read from the MAC register immediately after the instruction.
2. The normal minimum number of execution cycles is one, but three cycles are required when the operation result is read from the MAC register immediately after the MUL instruction.

Logic Operation Instructions

Table 2.22 Logic Operation Instructions

Instruction		Operation	Code	Privileged Mode	Cycles	T Bit
AND	Rm,Rn	Rn & Rm → Rn	0010nnnnmmmm1001	—	1	—
AND	#imm,R0	R0 & imm → R0	11001001iiiiiiii	—	1	—
AND.B	#imm,@(R0,GBR)	(R0 + GBR) & imm → (R0 + GBR)	11001101iiiiiiii	—	3	—
NOT	Rm,Rn	~Rm → Rn	0110nnnnmmmm0111	—	1	—
OR	Rm,Rn	Rn | Rm → Rn	0010nnnnmmmm1011	—	1	—
OR	#imm,R0	R0 | imm → R0	11001011iiiiiiii	—	1	—
OR.B	#imm,@(R0,GBR)	(R0 + GBR) | imm → (R0 + GBR)	11001111iiiiiiii	—	3	—
TAS.B	@Rn*	If (Rn) is 0, 1 → T; 1 → MSB of (Rn)	0100nnnn00011011	—	4	Test result
TST	Rm,Rn	Rn & Rm; if the result is 0, 1 → T	0010nnnnmmmm1000	—	1	Test result
TST	#imm,R0	R0 & imm; if the result is 0, 1 → T	11001000iiiiiiii	—	1	Test result
TST.B	#imm,@(R0,GBR)	(R0 + GBR) & imm; if the result is 0, 1 → T	11001100iiiiiiii	—	3	Test result
XOR	Rm,Rn	Rn ^ Rm → Rn	0010nnnnmmmm1010	—	1	—
XOR	#imm,R0	R0 ^ imm → R0	11001010iiiiiiii	—	1	—
XOR.B	#imm,@(R0,GBR)	(R0 + GBR) ^ imm → (R0 + GBR)	11001110iiiiiiii	—	3	—

Note: * An on-chip DMAC bus cycle is not inserted between a TAS instruction operand read cycle and write cycle. Also, bus release is not performed by $\overline{\text{BREQ}}$.

Shift Instructions

Table 2.23 Shift Instructions

Instruction		Operation	Code	Privileged Mode	Cycles	T Bit
ROTL	Rn	T ← Rn ← MSB	0100nnnn00000100	—	1	MSB
ROTR	Rn	LSB → Rn → T	0100nnnn00000101	—	1	LSB
ROTCL	Rn	T ← Rn ← T	0100nnnn00100100	—	1	MSB
ROTCR	Rn	T → Rn → T	0100nnnn00100101	—	1	LSB
SHAD	Rm,Rn	Rn ≥ 0: Rn << Rm → Rn Rn < 0: Rn >> Rm → [MSB → Rn]	0100nnnnmmmm1100	—	1	—
SHAL	Rn	T ← Rn ← 0	0100nnnn00100000	—	1	MSB
SHAR	Rn	MSB → Rn → T	0100nnnn00100001	—	1	LSB
SHLD	Rm,Rn	Rn ≥ 0: Rn << Rm → Rn Rn < 0: Rn >> Rm → [0 → Rn]	0100nnnnmmmm1101	—	1	—
SHLL	Rn	T ← Rn ← 0	0100nnnn00000000	—	1	MSB
SHLR	Rn	0 → Rn → T	0100nnnn00000001	—	1	LSB
SHLL2	Rn	Rn << 2 → Rn	0100nnnn00001000	—	1	—
SHLR2	Rn	Rn >> 2 → Rn	0100nnnn00001001	—	1	—
SHLL8	Rn	Rn << 8 → Rn	0100nnnn00011000	—	1	—
SHLR8	Rn	Rn >> 8 → Rn	0100nnnn00011001	—	1	—
SHLL16	Rn	Rn << 16 → Rn	0100nnnn00101000	—	1	—
SHLR16	Rn	Rn >> 16 → Rn	0100nnnn00101001	—	1	—

Branch Instructions

Table 2.24 Branch Instructions

Instruction		Operation	Code	Privileged Mode	Cycles	T Bit
BF	label	If T = 0, disp × 2 + PC → PC; if T = 1, nop (where label is disp + PC)	10001011dddddddd	—	3/1*	—
BF/S	label	Delayed branch, if T = 0, disp × 2 + PC → PC; if T = 1, nop	10001111dddddddd	—	2/1*	—
BT	label	Delayed branch, if T = 1, disp × 2 + PC → PC; if T = 0, nop	10001001dddddddd	—	3/1*	—
BT/S	label	If T = 1, disp × 2 + PC → PC; if T = 0, nop	10001101dddddddd	—	2/1*	—
BRA	label	Delayed branch, disp × 2 + PC → PC	1010dddddddddddd	—	2	—
BRAF	Rm	Delayed branch, Rm + PC → PC	0000mmmm00100011	—	2	—
BSR	label	Delayed branch, PC → PR, disp × 2 + PC → PC	1011dddddddddddd	—	2	—
BSRF	Rm	Delayed branch, PC → PR, Rm + PC → PC	0000mmmm00000011	—	2	—
JMP	@Rm	Delayed branch, Rm → PC	0100mmmm00101011	—	2	—
JSR	@Rm	Delayed branch, PC → PR, Rm → PC	0100mmmm00001011	—	2	—
RTS		Delayed branch, PR → PC	0000000000001011	—	2	—

Note: * One state when the branch is not executed.

System Control Instructions

Table 2.25 System Control Instructions

Instruction		Operation	Code	Privileged Mode	Cycles	T Bit
CLRMAC		0 → MACH, MACL	0000000000101000	—	1	—
CLRS		0 → S	0000000001001000	—	1	—
CLRT		0 → T	0000000000001000	—	1	0
LDC	Rm,SR	Rm → SR	0100mmmm00001110	√	5	LSB
LDC	Rm,GBR	Rm → GBR	0100mmmm00011110	—	3	—
LDC	Rm,VBR	Rm → VBR	0100mmmm00101110	√	3	—
LDC	Rm,SSR	Rm → SSR	0100mmmm00111110	√	3	—
LDC	Rm,SPC	Rm → SPC	0100mmmm01001110	√	3	—
LDC	Rm,R0_BANK	Rm → R0_BANK	0100mmmm10001110	√	3	—
LDC	Rm,R1_BANK	Rm → R1_BANK	0100mmmm10011110	√	3	—
LDC	Rm,R2_BANK	Rm → R2_BANK	0100mmmm10101110	√	3	—
LDC	Rm,R3_BANK	Rm → R3_BANK	0100mmmm10111110	√	3	—
LDC	Rm,R4_BANK	Rm → R4_BANK	0100mmmm11001110	√	3	—
LDC	Rm,R5_BANK	Rm → R5_BANK	0100mmmm11011110	√	3	—
LDC	Rm,R6_BANK	Rm → R6_BANK	0100mmmm11101110	√	3	—
LDC	Rm,R7_BANK	Rm → R7_BANK	0100mmmm11111110	√	3	—
LDC.L	@Rm+,SR	(Rm) → SR, Rm + 4 → Rm	0100mmmm00000111	√	7	LSB
LDC.L	@Rm+,GBR	(Rm) → GBR, Rm + 4 → Rm	0100mmmm00010111	—	5	—
LDC.L	@Rm+,VBR	(Rm) → VBR, Rm + 4 → Rm	0100mmmm00100111	√	5	—
LDC.L	@Rm+,SSR	(Rm) → SSR, Rm + 4 → Rm	0100mmmm00110111	√	5	—
LDC.L	@Rm+,SPC	(Rm) → SPC, Rm + 4 → Rm	0100mmmm01000111	√	5	—
LDC.L	@Rm+, R0_BANK	(Rm) → R0_BANK, Rm + 4 → Rm	0100mmmm10000111	√	5	—
LDC.L	@Rm+, R1_BANK	(Rm) → R1_BANK, Rm + 4 → Rm	0100mmmm10010111	√	5	—
LDC.L	@Rm+, R2_BANK	(Rm) → R2_BANK, Rm + 4 → Rm	0100mmmm10100111	√	5	—
LDC.L	@Rm+, R3_BANK	(Rm) → R3_BANK, Rm + 4 → Rm	0100mmmm10110111	√	5	—
LDC.L	@Rm+, R4_BANK	(Rm) → R4_BANK, Rm + 4 → Rm	0100mmmm11000111	√	5	—

Instruction		Operation	Code	Privileged Mode	Cycles	T Bit
LDC.L	@Rm+, R5_BANK	(Rm) → R5_BANK, Rm + 4 → Rm	0100mmmm11010111	√	5	—
LDC.L	@Rm+, R6_BANK	(Rm) → R6_BANK, Rm + 4 → Rm	0100mmmm11100111	√	5	—
LDC.L	@Rm+, R7_BANK	(Rm) → R7_BANK, Rm + 4 → Rm	0100mmmm11110111	√	5	—
LDS	Rm,MACH	Rm → MACH	0100mmmm00001010	—	1	—
LDS	Rm,MACL	Rm → MACL	0100mmmm00011010	—	1	—
LDS	Rm,PR	Rm → PR	0100mmmm00101010	—	1	—
LDS.L	@Rm+,MACH	(Rm) → MACH, Rm + 4 → Rm	0100mmmm00000110	—	1	—
LDS.L	@Rm+,MACL	(Rm) → MACL, Rm + 4 → Rm	0100mmmm00010110	—	1	—
LDS.L	@Rm+,PR	(Rm) → PR, Rm + 4 → Rm	0100mmmm00100110	—	1	—
LDTLB		PTEH/PTEL → TLB	0000000000111000	√	1	—
NOP		No operation	0000000000001001	—	1	—
PREF	@Rm	(Rm) → cache	0000mmmm10000011	—	2	—
RTE		Delayed branch, SSR/SPC → SR/PC	0000000000101011	√	4	—
SETS		1 → S	0000000001011000	—	1	—
SETT		1 → T	0000000000011000	—	1	1
SLEEP		Sleep	0000000000011011	√	4*	—
STC	SR,Rn	SR → Rn	0000nnnn00000010	√	1	—
STC	GBR,Rn	GBR → Rn	0000nnnn00010010	—	1	—
STC	VBR,Rn	VBR → Rn	0000nnnn00100010	√	1	—
STC	SSR,Rn	SSR → Rn	0000nnnn00110010	√	1	—
STC	SPC,Rn	SPC → Rn	0000nnnn01000010	√	1	—
STC	R0_BANK,Rn	R0_BANK→ Rn	0000nnnn10000010	√	1	—
STC	R1_BANK,Rn	R1_BANK→ Rn	0000nnnn10010010	√	1	—
STC	R2_BANK,Rn	R2_BANK→ Rn	0000nnnn10100010	√	1	—
STC	R3_BANK,Rn	R3_BANK→ Rn	0000nnnn10110010	√	1	—
STC	R4_BANK,Rn	R4_BANK→ Rn	0000nnnn11000010	√	1	—
STC	R5_BANK,Rn	R5_BANK→ Rn	0000nnnn11010010	√	1	—
STC	R6_BANK,Rn	R6_BANK→ Rn	0000nnnn11100010	√	1	—
STC	R7_BANK,Rn	R7_BANK→ Rn	0000nnnn11110010	√	1	—

Instruction	Operation	Code	Privileged Mode	Cycles	T Bit
STC.L SR,@–Rn	Rn–4 → Rn, SR → (Rn)	0100nnnn00000011	√	2	—
STC.L GBR,@–Rn	Rn–4 → Rn, GBR → (Rn)	0100nnnn00010011	—	2	—
STC.L VBR,@–Rn	Rn–4 → Rn, VBR → (Rn)	0100nnnn00100011	√	2	—
STC.L SSR,@–Rn	Rn–4 → Rn, SSR → (Rn)	0100nnnn00110011	√	2	—
STC.L SPC,@–Rn	Rn–4 → Rn, SPC → (Rn)	0100nnnn01000011	√	2	—
STC.L R0_BANK, @–Rn	Rn–4 → Rn, R0_BANK → (Rn)	0100nnnn10000011	√	2	—
STC.L R1_BANK, @–Rn	Rn–4 → Rn, R1_BANK → (Rn)	0100nnnn10010011	√	2	—
STC.L R2_BANK, @–Rn	Rn–4 → Rn, R2_BANK → (Rn)	0100nnnn10100011	√	2	—
STC.L R3_BANK, @–Rn	Rn–4 → Rn, R3_BANK → (Rn)	0100nnnn10110011	√	2	—
STC.L R4_BANK, @–Rn	Rn–4 → Rn, R4_BANK → (Rn)	0100nnnn11000011	√	2	—
STC.L R5_BANK, @–Rn	Rn–4 → Rn, R5_BANK → (Rn)	0100nnnn11010011	√	2	—
STC.L R6_BANK, @–Rn	Rn–4 → Rn, R6_BANK → (Rn)	0100nnnn11100011	√	2	—
STC.L R7_BANK, @–Rn	Rn–4 → Rn, R7_BANK → (Rn)	0100nnnn11110011	√	2	—
STS MACH,Rn	MACH → Rn	0000nnnn00001010	—	1	—
STS MACL,Rn	MACL → Rn	0000nnnn00011010	—	1	—
STS PR,Rn	PR → Rn	0000nnnn00101010	—	1	—
STS.L MACH,@–Rn	Rn–4 → Rn, MACH → (Rn)	0100nnnn00000010	—	1	—
STS.L MACL,@–Rn	Rn–4 → Rn, MACL → (Rn)	0100nnnn00010010	—	1	—
STS.L PR,@–Rn	Rn–4 → Rn, PR → (Rn)	0100nnnn00100010	—	1	—
TRAPA #imm	PC → SPC, SR → SSR, imm << 2 → TRA, VBR + H'0100 → PC	11000011iiiiiiii	—	8	—

Note: * Number of states before the chip enters the sleep state.

The table shows the minimum number of clocks required for execution. In practice, the number of execution cycles will be increased if there is contention between an instruction fetch and a data access, or if the destination register of a load instruction (memory → register) is also used by the following instruction.

2.6 DSP Extended-Function Instructions

2.6.1 Introduction

The newly added instructions are classified into the following three groups:

1. Additional system control instructions for the CPU unit
2. DSP unit memory-register single and double data transfer
3. DSP unit parallel processing

Group 1 instructions are provided to support loop control and data transfer between CPU core registers or memory and new control registers added to the CPU core. DSP operations employ a multi-level nested-loop structure. With a single-level loop, use of the decrement and test, DTRn, and conditional delayed branch BF/S instructions supported by the SH-3 is adequate. However, with nested loops, DSP performance can be improved by means of a zero-overhead loop control function.

The RS, RE, and MOD registers have been added to support loop control and modulo addressing functions. Instructions are supported for data transfer between these new control registers and general registers or memory. In addition, the LDRS and LDRE address calculation registers have been added to reduce the code size for the initial settings for zero-overhead loop control.

An independent control register, DSR, is provided for the DSP engine. This register is treated as a system register such as MACL and MACH. The A0, X0, X1, Y0, and Y1 registers are treated as system registers from the CPU side, and LDS/STS instructions are supported for the same purpose. Table 2.26 shows the instruction code map for the new system control instructions for the CPU core.

Group 2 instructions are provided to reduce DSP operation program code size. Data transfer instructions that perform no data processing are frequently executed by the DSP engine. In this case, a 32-bit instruction code is unnecessarily long, and wastes space in the program memory area. All instructions in this class have a 16-bit code length, the same as conventional SH core instructions. Single data transfer instructions have greater flexibility in terms of operands than the double data transfer instruction or parallel instruction class.

Group 3 instructions are provided for fast execution of digital signal processing operations using the DSP unit. These instructions have a 32-bit instruction code, so that a maximum of four instructions—an ALU operation, multiplication, and two data transfer instructions—can be executed in parallel.

2.6.2 Added CPU System Control Instructions

The new instructions in this class are treated as part of the CPU core functions, and therefore all the added instructions have a 16-bit code length. All the additional instructions belong to the system control instruction group. Table 2.26 summarizes the added system instructions. New control registers—RS, RE, and MOD—have been added to the CPU core to support loop control and modulo addressing functions, and LDC and STS type instructions have been provided for these registers.

The DSP engine's DSR, A0, X0, X1, Y0, and Y1 registers are treated as system registers such as MACH and MACL, and therefore STS and LDS instructions are supported for these registers. As digital signal processing operations usually employ a multi-level nested-loop structure, DSP performance can be improved by means of a zero-overhead loop control function. SETRC type instructions are provided to set the repeat count in the RC field in SR[27:16]. When an immediate operand type SETRC instruction is executed, the 8-bit immediate operand data is set in SR[23:16], and 0 is set in the remaining bits, SR[27:24]. When a register operand type SETRC instruction is executed, Rn[11:0] is set in SR[27:16]. The start address and end address of the repeat loop are set in the RS register and RE register. There are two ways of setting the addresses: by using an LDC type instruction, or by using the LDRS and LDRE instructions.

Table 2.26 Added CPU System Control Instructions

Instruction		Instruction Code	Operation	Execution States	T Bit
SETRC	#imm	10000010iiiiiiii	imm → RC (of SR)	3	—
SETRC	Rn	0100nnnn00010100	Rn[11:0] → R C (of SR)	3	—
LDRS	@(disp,PC)	10001100dddddddd	(disp × 2 + PC) → RS	3	—
LDRE	@(disp,PC)	10001110dddddddd	(disp × 2 + PC) → RE	3	—
STC	MOD,Rn	0000nnnn01010010	MOD → Rn	1	—
STC	RS,Rn	0000nnnn01100010	RS → Rn	1	—
STC	RE,Rn	0000nnnn01110010	RE → Rn	1	—
STS	DSR,Rn	0000nnnn01101010	DSR → Rn	1	—
STS	A0,Rn	0000nnnn01111010	A0 → Rn	1	—
STS	X0,Rn	0000nnnn10001010	X0 → Rn	1	—
STS	X1,Rn	0000nnnn10011010	X1 → Rn	1	—
STS	Y0,Rn	0000nnnn10101010	Y0 → Rn	1	—
STS	Y1,Rn	0000nnnn10111010	Y1 → Rn	1	—
STS.L	DSR,@-Rn	0100nnnn01100010	Rn – 4 → Rn, DSR → (Rn)	1	—
STS.L	A0,@-Rn	0100nnnn01110010	Rn – 4 → Rn, A0 → (Rn)	1	—
STS.L	X0,@-Rn	0100nnnn10000010	Rn – 4 → Rn, X0 → (Rn)	1	—
STS.L	X1,@-Rn	0100nnnn10010010	Rn – 4 → Rn, X1 → (Rn)	1	—
STS.L	Y0,@-Rn	0100nnnn10100010	Rn – 4 → Rn, Y0 → (Rn)	1	—
STS.L	Y1,@-Rn	0100nnnn10110010	Rn – 4 → Rn, Y1 → (Rn)	1	—
STC.L	MOD,@-Rn	0100nnnn01010011	Rn – 4 → Rn, MOD → (Rn)	2	—
STC.L	RS,@-Rn	0100nnnn01100011	Rn – 4 → Rn, RS → (Rn)	2	—
STC.L	RE,@-Rn	0100nnnn01110011	Rn – 4 → Rn, RE → (Rn)	2	—
LDS.L	@Rn+,DSR	0100nnnn01100110	(Rn) → DSR, Rn + 4 → Rn	1	—
LDS.L	@Rn+,A0	0100nnnn01110110	(Rn) → A0, Rn + 4 → Rn	1	—
LDS.L	@Rn+,X0	0100nnnn10000110	(Rn) → X0, Rn + 4 → Rn	1	—
LDS.L	@Rn+,X1	0100nnnn10010110	(Rn) → X1, Rn + 4 → Rn	1	—
LDS.L	@Rn+,Y0	0100nnnn10100110	(Rn) → Y0, Rn + 4 → Rn	1	—
LDS.L	@Rn+,Y1	0100nnnn10110110	(Rn) → Y1, Rn + 4 → Rn	1	—
LDC.L	@Rn+,MOD	0100nnnn01010111	(Rn) → MOD, Rn + 4 → Rn	5	—
LDC.L	@Rn+,RS	0100nnnn01100111	(Rn) → RS, Rn + 4 → Rn	5	—

Instruction		Instruction Code	Operation	Execution States	T Bit
LDC.L	@Rn+,RE	0100nnnn01110111	(Rn) → RE, Rn + 4 → Rn	5	—
LDS	Rn,DSR	0100nnnn01101010	Rn → DSR	1	—
LDS	Rn,A0	0100nnnn01111010	Rn → A0	1	—
LDS	Rn,X0	0100nnnn10001010	Rn → X0	1	—
LDS	Rn,X1	0100nnnn10011010	Rn → X1	1	—
LDS	Rn,Y0	0100nnnn10101010	Rn → Y0	1	—
LDS	Rn,Y1	0100nnnn10111010	Rn → Y1	1	—
LDC	Rn,MOD	0100nnnn01011110	Rn → MOD	3	—
LDC	Rn,RS	0100nnnn01101110	Rn → RS	3	—
LDC	Rn,RE	0100nnnn01111110	Rn → RE	3	—

2.6.3 Single and Double Data Transfer for DSP Data Instructions

The new instructions in this class are provided to reduce the program code size for DSP operations. All the new instructions in this class have a 16-bit code length. Instructions in this class are divided into two groups: single data transfer instructions and double data transfer instructions. The operand flexibility of the double data transfer instructions is the same as with the A field in parallel instruction class data transfer instructions described in section 2.6.4, DSP Operation Instruction Set. However, conditional load instructions cannot be used with these 16-bit instructions. In single transfer, the Ax pointer and two other pointers are used as the As pointer, but the Ay pointer is not used. Tables 2.27 and 2.28 list the single and double data transfer instructions.

With double data transfer group instructions, X memory and Y memory can be accessed in parallel. The Ax pointer can only be used by X memory access instructions, and the Ay pointer only by Y memory access instructions. Double data transfer instructions can only access the on-chip X and Y memory areas. Single data transfer instructions use a 16-bit instruction code, and can access any memory address space.

Rn (n = 2 to 7) registers are normally used as the Ax, Ay, and As pointers. The pointer names themselves can be changed with the assembler rename function. The following renaming scheme is recommended.

R2:As2, R3:As3, R4:Ax0 (As0), R5:Ax1 (As1), R6:Ay0, R7:Ay1, R8:Ix, R9:Iy

Table 2.27 Double Data Transfer Instructions

Instruction		Instruction Code	Operation	Execution States	DC
X memory data transfer	NOPX	1111000*0*0*00**	X memory no access	1	—
	MOVX.W @Ax,Dx	111100A*D*0*01**	(Ax) → MSW of Dx, 0 → LSW of Dx	1	—
	MOVX.W @Ax+,Dx	111100A*D*0*10**	(Ax) → MSW of Dx, 0 → LSW of Dx, Ax + 2 → Ax	1	—
	MOVX.W @Ax+Ix,Dx	111100A*D*0*11**	(Ax) → MSW of Dx, 0 → LSW of Dx, Ax + Ix → Ax	1	—
	MOVX.W Da,@Ax	111100A*D*1*01**	MSW of Da → (Ax)	1	—
	MOVX.W Da,@Ax+	111100A*D*1*10**	MSW of Da → (Ax), Ax + 2 → Ax	1	—
	MOVX.W Da,@Ax+Ix	111100A*D*1*11**	MSW of Da → (Ax), Ax + Ix → Ax	1	—
Y memory data transfer	NOPY	111100*0*0*0**00	Y memory no access	1	—
	MOVY.W @Ay,Dy	111100*A*D*0**01	(Ay) → MSW of Dy, 0 → LSW of Dy	1	—
	MOVY.W @Ay+,Dy	111100*A*D*0**10	(Ay) → MSW of Dy, 0 → LSW of Dy, Ay + 2 → Ay	1	—
	MOVY.W @Ay+Iy,Dy	111100*A*D*0**11	(Ay) → MSW of Dy, 0 → LSW of Dy, Ay + Iy → Ay	1	—
	MOVY.W Da,@Ay	111100*A*D*1**01	MSW of Da → (Ay)	1	—
	MOVY.W Da,@Ay+	111100*A*D*1**10	MSW of Da → (Ay), Ay + 2 → Ay	1	—
	MOVY.W Da,@Ay+Iy	111100*A*D*1**11	MSW of Da → (Ay), Ay + Iy → Ay	1	—

Table 2.28 Single Data Transfer Instructions

Instruction	Instruction Code	Operation	Execution States	DC
MOVS.W @-As,Ds	111101AADDDD0000	As – 2 → As, (As) → MSW of Ds, 0 → LSW of Ds	1	—
MOVS.W @As,Ds	111101AADDDD0100	(As) → MSW of Ds, 0 → LSW of Ds	1	—
MOVS.W @As+,Ds	111101AADDDD1000	(As) → MSW of Ds, 0 → LSW of Ds, As + 2 → As	1	—
MOVS.W @As+Ix,Ds	111101AADDDD1100	(Asc) → MSW of Ds, 0 → LSW of Ds, As + Ix → As	1	—
MOVS.W Ds,@-As*	111101AADDDD0001	As – 2 → As, MSW of Ds → (As)	1	—
MOVS.W Ds,@As*	111101AADDDD0101	MSW of Ds → (As)	1	—
MOVS.W Ds,@As+*	111101AADDDD1001	MSW of Ds → (As), As + 2 → As	1	—
MOVS.W Ds,@As+Ix*	111101AADDDD1101	MSW of Ds → (As), As + Ix → As	1	—
MOVS.L @-As,Ds	111101AADDDD0010	As – 4 → As, (As) → Ds	1	—
MOVS.L @As,Ds	111101AADDDD0110	(As) → Ds	1	—
MOVS.L @As+,Ds	111101AADDDD1010	(As) → Ds, As + 4 → As	1	—
MOVS.L @As+Ix,Ds	111101AADDDD1110	(As) → Ds, As + Ix → As	1	—
MOVS.L Ds,@-As	111101AADDDD0011	As – 4 → As, Ds → (As)	1	—
MOVS.L Ds,@As	111101AADDDD0111	Ds → (As)	1	—
MOVS.L Ds,@As+	111101AADDDD1011	Ds → (As), As + 4 → As	1	—
MOVS.L Ds,@As+Ix	111101AADDDD1111	Ds → (As), As + Ix → As	1	—

Note: * If guard bit registers A0G and A1G are specified in source operand Ds, the data is output to the LDB[7:0] bus and the sign bit is copied into the upper bits, [31:8].

The correspondence between DSP data transfer operands and registers is shown in table 2.29. CPU core registers are used as a pointer address that indicates a memory address.

Table 2.29 Correspondence between DSP Data Transfer Operands and Registers

Register		Ax	Ix	Dx	Ay	Iy	Dy	Da	As	Ds
CPU register	R0									
	R1									
	R2 (As2)								Yes	
	R3 (As3)								Yes	
	R4 (Ax0)	Yes							Yes	
	R5 (Ax1)	Yes							Yes	
	R6 (Ay0)				Yes					
	R7 (Ay1)				Yes					
	R8 (Ix)		Yes							
	R9 (Iy)					Yes				
DSP register	A0							Yes		Yes
	A1							Yes		Yes
	M0									Yes
	M1									Yes
	X0			Yes						Yes
	X1			Yes						Yes
	Y0						Yes			Yes
	Y1						Yes			Yes
	A0G									Yes
	A1G									Yes

2.6.4 DSP Operation Instruction Set

DSP operation instructions are instructions for digital signal processing performed by the DSP unit. These instructions have a 32-bit instruction code, and multiple instructions can be executed in parallel. The instruction code is divided into an A field and B field; a parallel data transfer instruction is specified in the A field, and a single or double data operation instruction in the B field. Instructions can be specified independently, and are also executed independently. The parallel data transfer instruction specified in the A field is exactly the same as a double data transfer instruction. The function of the A field—that is, the data transfer instruction field—is

basically the same as in the double data transfer instructions described in section 2.6.3, Single and Double Data Transfer for DSP Data Instructions, but has a special function in load instructions.

B-field data operation instructions are of three kinds: double data operation instructions, conditional single data operation instructions, and unconditional single data operation instructions. The formats of the DSP operation instructions are shown in table 2.30. The respective operands are selected independently from the DSP registers. The correspondence between DSP operation instruction operands and registers is shown in table 2.31.

Table 2.30 DSP Operation Instruction Formats

Type	Instruction Formats	
Double data operation instructions		`ALUop. Sx, Sy, Du`
		`MLTop. Se, Df, Dg`
Conditional single data operation instructions		`ALUop. Sx, Sy, Dz`
	`DCT`	`ALUop. Sx, Sy, Dz`
	`DCF`	`ALUop. Sx, Sy, Dz`
		`ALUop. Sx, Dz`
	`DCT`	`ALUop. Sx, Dz`
	`DCF`	`ALUop. Sx, Dz`
		`ALUop. Sy, Dz`
	`DCT`	`ALUop. Sy, Dz`
	`DCF`	`ALUop. Sy, Dz`
Unconditional single data operation instructions		`ALUop. Sx, Sy, Dz`
		`ALUop. Sx, Dz`
		`ALUop. Sy, Dz`
		`MLTop. Se, Sf, Dg`

Table 2.31 Correspondence between DSP Instruction Operands and Registers

Register	ALU/BPU Operations				Multiply Operations		
	Sx	Sy	Dz	Du	Se	Sf	Dg
A0	Yes		Yes	Yes			Yes
A1	Yes		Yes	Yes	Yes	Yes	Yes
M0		Yes	Yes				Yes
M1		Yes	Yes				Yes
X0	Yes		Yes	Yes	Yes	Yes	
X1	Yes		Yes		Yes		
Y0		Yes	Yes	Yes	Yes	Yes	
Y1		Yes	Yes			Yes	

When writing parallel instructions, the B-field instruction is written first, followed by the A-field instruction. A sample parallel processing program is shown in figure 2.16.

```
     PADD A0, M0, A0  PMULS X0, Y0, M0  MOVX.W @R4+, X0    MOVY.W @R6+, Y0 [;]
DCF  PINC X1, A1                        MOVX.W A0, @R5+R8  MOVY.W @R7+, Y0 [;]
     PCMP X1, M0                        MOVX.W @R4         [NOPY] [;]
```

Figure 2.16 Sample Parallel Instruction Program

Square brackets mean that the contents can be omitted.

The no operation instructions NOPX and NOPY can be omitted. Table 2.32 gives an overview of the B field in parallel operation instructions.

A semicolon is the instruction line delimiter, but this can also be omitted. If the semicolon delimiter is used, the area to the right of the semicolon can be used as a comment field. This has the same function as with conventional SH tools.

The DSR register condition code bit (DC) is always updated on the basis of the result of an unconditional ALU or shift operation instruction. Conditional instructions do not update the DC bit. Multiply instructions, also, do not update the DC bit. DC bit updating is performed by means of bits CS0 to CS2 in the DSR register. The DC bit update rules are shown in table 2.33.

Table 2.32 DSP Operation Instructions

Instruction	Instruction Code	Operation	Execution States	DC
PMULS Se,Sf,Dg	111110**********	Se * Sf → Dg (signed)	1	—
	0100eeff0000gg00			
PADD Sx,Sy,Du	111110**********	Sx + Sy → Du	1	*
PMULS Se,Sf,Dg	0111eeffxxyygguu	Se * Sf → Dg (signed)		
PSUB Sx,Sy,Du	111110**********	Sy – Sy → Du	1	*
PMULS Se,Sf,Dg	0110eeffxxyygguu	Se * Sf → Dg (signed)		
PADD Sx,Sy,Dz	111110**********	Sx + Sy → Dz	1	*
	10110001xxyyzzzz			
DCT PADD Sx,Sy,Dz	111110**********	If DC = 1, Sx + Sy → Dz	1	*
	10110010xxyyzzzz	If DC = 0, nop		
DCF PADD Sx,Sy,Dz	111110**********	If DC = 0, Sx + Sy → Dz	1	*
	10110011xxyyzzzz	If DC = 1, nop		
PSUB Sx,Sy,Dz	111110**********	Sx – Sy → Dz	1	*
	10100001xxyyzzzz			
DCT PSUB Sx,Sy,Dz	111110**********	If DC = 1, Sx – Sy → Dz	1	*
	10100010xxyyzzzz	If DC = 0, nop		
DCF PSUB Sx,Sy,Dz	111110**********	If DC = 0, Sx – Sy → Dz	1	*
	10100011xxyyzzzz	If DC = 1, nop		
PSHA Sx,Sy,Dz	111110**********	If Sy > = 0, Sx << Sy → Dz (arithmetic shift)	1	*
	1010001xxyyzzzz	If Sy<0, Sx>>Sy → Dz		
DCT PSHA Sx,Sy,Dz	111110**********	If DC = 1 & Sy > = 0, Sx << Sy → Dz (arithmetic shift)	1	*
	10010010xxyyzzzz	If DC = 1 & Sy < 0, Sx >> Sy → Dz		
		If DC = 0, nop		

Instruction	Instruction Code	Operation	Execution States	DC
DCF PSHA Sx,Sy,Dz	111110********** 10010011xxyyzzzz	If DC = 0 & Sy > = 0, Sx << Sy → Dz (arithmetic shift) If DC = 0 & Sy < 0, Sx >> Sy → Dz If DC = 1, nop	1	*
PSHL Sx,Sy,Dz	111110********** 10000001xxyyzzzz	If Sy > = 0, Sx << Sy → Dz (logical shift) If Sy < 0, Sx >> Sy → Dz	1	*
DCT PSHL Sx,Sy,Dz	111110********** 10000010xxyyzzzz	If DC = 1 & Sy > = 0, Sx << Sy → Dz (logical shift) If DC = 1 & Sy < 0, Sx >> Sy → Dz If DC = 0, nop	1	*
DCF PSHL Sx,Sy,Dz	111110********** 10000011xxyyzzzz	If DC = 0 & Sy > = 0, Sx << Sy → Dz (logical shift) If DC = 0 & Sy < 0, Sx >> Sy → Dz If DC = 1, nop	1	*
PCOPY Sx,Dz	111110********** 11011001xx00zzzz	Sx → Dz	1	*
PCOPY Sy,Dz	111110********** 1111100100yyzzzz	Sy → Dz	1	*
DCT PCOPY Sx,Dz	111110********** 11011010xx00zzzz	If DC = 1, Sx → Dz If DC = 0, nop	1	*
DCT PCOPY Sy,Dz	111110********** 1111101000yyzzzz	If DC = 1, Sy → Dz If DC = 0, nop	1	*
DCF PCOPY Sx,Dz	111110********** 11011011xx00zzzz	If DC = 0, Sx → Dz If DC = 1, nop	1	*
DCF PCOPY Sy,Dz	111110********** 1111101100yyzzzz	If DC = 0, Sy → Dz If DC = 1, nop	1	*
PDMSB Sx,Dz	111110********** 10011101xx00zzzz	Sx → Dz normalization count shift value	1	*

Instruction		Instruction Code	Operation	Execution States	DC
	PDMSB Sy,Dz	111110********** 1011110100yyzzzz	Sx → Dz normalization count shift value	1	*
DCT	PDMSB Sx,Dz	111110********** 10011110xx00zzzz	If DC = 1, normalization count shift value Sx → Dz If DC = 0, nop	1	*
DCT	PDMSB Sy,Dz	111110********** 1011111000yyzzzz	If DC = 1, normalization count shift value Sy → Dz If DC = 0, nop	1	*
DCF	PDMSB Sx,Dz	111110********** 10011111xx00zzzz	If DC = 0, normalization count shift value Sx → Dz If DC = 1, nop	1	*
DCF	PDMSB Sy,Dz	111110********** 1011111100yyzzzz	If DC = 0, normalization count shift value Sy → Dz If DC = 1, nop	1	*
	PINC Sx,Dz	111110********** 10011001xx00zzzz	MSW of Sx → Dz	1	*
	PINC Sy,Dz	111110********** 1011100100yyzzzz	MSW of Sy → Dz	1	*
DCT	PINC Sx,Dz	111110********** 10011010xx00zzzz	If DC = 1, MSW of Sx + 1 → Dz If DC = 0, nop	1	*
DCT	PINC Sy,Dz	111110********** 1011101000yyzzzz	If DC = 1, MSW of Sy + 1 → Dz If DC = 0, nop	1	*
DCF	PINC Sx,Dz	111110********** 10011011xx00zzzz	If DC = 0, MSW of Sx + 1 → Dz If DC = 1, nop	1	*
DCF	PINC Sy,Dz	111110********** 1011101100yyzzzz	If DC = 0, MSW of Sy + 1 → Dz If DC = 1, nop	1	*
	PNEG Sx,Dz	111110********** 11001001xx00zzzz	0 – Sx → Dz	1	*
	PNEG Sy,Dz	111110********** 1110100100yyzzzz	0 – Sy → Dz	1	*

Instruction	Instruction Code	Operation	Execution States	DC
DCT PNEG Sx,Dz	111110**********	If DC = 1, 0 – Sx → Dz	1	*
	11001010xx00zzzz	If DC = 0, nop		
DCT PNEG Sy,Dz	111110**********	If DC = 1, 0 – Sy → Dz	1	*
	1110101000yyzzzz	If DC = 0, nop		
DCF PNEG Sx,Dz	111110**********	If DC = 0, 0 – Sx → Dz	1	*
	11001011xx00zzzz	If DC = 1, nop		
DCF PNEG Sy,Dz	111110**********	If DC = 0, 0 – Sy → Dz	1	*
	1110101100yyzzzz	If DC = 1, nop		
POR Sx,Sy,Dz	111110**********	Sx | Sy → Dz	1	*
	10110101xxyyzzzz			
DCT POR Sx,Sy,Dz	111110**********	If DC = 1, Sx | Sy → Dz	1	*
	10110110xxyyzzzz	If DC = 0, nop		
DCF POR Sx,Sy,Dz	111110**********	If DC = 0, Sx | Sy → Dz	1	*
	10110111xxyyzzzz	If DC = 1, nop		
PAND Sx,Sy,Dz	111110**********	Sx & Sy → Dz	1	*
	10010101xxyyzzzz			
DCT PAND Sx,Sy,Dz	111110**********	If DC = 1, Sx & Sy → Dz	1	*
	10010110xxyyzzzz	If DC = 0, nop		
DCF PAND Sx,Sy,Dz	111110**********	If DC = 0, Sx & Sy → Dz	1	*
	10010111xxyyzzzz	If DC = 1, nop		
PXOR Sx,Sy,Dz	111110**********	Sx ^ Sy → Dz	1	*
	10100101xxyyzzzz			
DCT PXOR Sx,Sy,Dz	111110**********	If DC = 1, Sx ^ Sy → Dz	1	*
	10100110xxyyzzzz	If DC = 0, nop		
DCF PXOR Sx,Sy,Dz	111110**********	If DC = 1, Sx ^ Sy → Dz	1	*
	10100111xxyyzzzz	If DC = 0, nop		
PDEC Sx,Dz	111110**********	Sx [39:16] – 1 → Dz	1	*
	10001001xx00zzzz			
PDEC Sy,Dz	111110**********	Sy [31:16] – 1 → Dz	1	*
	1010100100yyzzzz			

Instruction		Instruction Code	Operation	Execution States	DC
DCT	PDEC Sx,Dz	111110********** 10001010xx00zzzz	If DC = 1, Sx [39:16] – 1 → Dz If DC = 0, nop	1	*
DCT	PDEC Sy,Dz	111110********** 1010101000yyzzzz	If DC = 1, Sy [31:16] – 1 → Dz If DC = 0, nop	1	*
DCF	PDEC Sx,Dz	111110********** 10001011xx00zzzz	If DC = 0, Sx [39:16] – 1 → Dz If DC = 1, nop	1	*
DCF	PDEC Sy,Dz	111110********** 1010101100yyzzzz	If DC = 0, Sy [31:16] – 1 → Dz If DC = 1, nop	1	*
	PCLR Dz	111110********** 100011010000zzzz	H'00000000 → Dz	1	*
DCT	PCLR Dz	111110********** 100011100000zzzz	If DC = 1, H'00000000 → Dz If DC = 0, nop	1	*
DCF	PCLR Dz	111110********** 100011110000zzzz	If DC = 0, H'00000000 → Dz If DC = 1, nop	1	*
	PSHA #imm,Dz	111110********** 00010iiiiiiizzzz	If imm > = 0, Dz << imm → Dz (arithmetic shift) If imm<0, Dz>>imm → Dz	1	*
	PSHL #imm,Dz	111110********** 00000iiiiiiizzzz	If imm > = 0, Dz << imm → Dz (logical shift) If imm < 0, Dz >> imm → Dz	1	*
	PSTS MACH,Dz	111110********** 110011010000zzzz	MACH → Dz	1	—
DCT	PSTS MACH,Dz	111110********** 110011100000zzzz	If DC = 1, MACH → Dz	1	—
DCF	PSTS MACH,Dz	111110********** 110011110000zzzz	If DC = 0, MACH → Dz	1	—
	PSTS MACL,Dz	111110********** 110111010000zzzz	MACL → Dz	1	—

Instruction		Instruction Code	Operation	Execution States	DC
DCT	PSTS MACL,Dz	111110**********	If DC = 1, MACL → Dz	1	—
		1101111100000zzzz			
DCF	PSTS MACL,Dz	111110**********	If DC = 0, MACL → Dz	1	—
		1101111110000zzzz			
	PLDS Dz,MACH	111110**********	Dz → MACH	1	—
		1110110100000zzzz			
DCT	PLDS Dz,MACH	111110**********	If DC = 1, Dz → MACH	1	—
		1110111100000zzzz			
DCF	PLDS Dz,MACH	111110**********	If DC = 0, Dz → MACH	1	—
		1110111110000zzzz			
	PLDS Dz,MACL	111110**********	Dz → MACL	1	—
		1111110100000zzzz			
DCT	PLDS Dz,MACL	111110**********	If DC = 1, Dz → MACL	1	—
		1111111100000zzzz			
DCF	PLDS Dz,MACL	111110**********	If DC = 0, Dz → MACL	1	—
		1111111110000zzzz			
	PADDC Sx,Sy,Dz	111110**********	Sx + Sy + DC → Dz	1	Carry
		10110000xxyyzzzz	Carry → DC		
	PSUBC Sx,Sy,Dz	111110**********	Sx – Sy – DC → Dz	1	Borrow
		10100000xxyyzzzz	Borrow → DC		
	PCMP Sx,Sy	111110**********	Sx – Sy → DC update*	1	*
		10000100xxyy0000			
	PABS Sx,Dz	111110**********	If Sx < 0, 0 – Sx → Dz	1	*
		10001000xx00zzzz	If Sx > = 0, nop		
	PABS Sy,Dz	111110**********	If Sy < 0, 0 – Sy → Dz	1	*
		1010100000yyzzzz	If Sx > = 0, nop		
	PRND Sx,Dz	111110**********	Sx + H'00008000 → Dz	1	*
		10011000xx00zzzz	LSW of Dz → H'0000		
	PRND Sy,Dz	111110**********	Sy + H'00008000 → Dz	1	*
		1011100000yyzzzz	LSW of Dz → H'0000		

Note: * See table 2.33.

Table 2.33 DC Bit Update Definitions

CS [2:0]			Condition Mode	Description
0	0	0	Carry or borrow mode	The DC bit is set if an ALU arithmetic operation generates a carry or borrow, and is cleared otherwise. When a PSHA or PSHL shift instruction is executed, the last bit data shifted out is copied into the DC bit. When an ALU logical operation is executed, the DC bit is always cleared.
0	0	1	Negative value mode	When an ALU or shift (PSHA) arithmetic operation is executed, the MSB of the result, including the guard bits, is copied into the DC bit. When an ALU or shift (PSHL) logical operation is executed, the MSB of the result, excluding the guard bits, is copied into the DC bit.
0	1	0	Zero value mode	The DC bit is set if the result of an ALU or shift operation is all-zeros, and is cleared otherwise.
0	1	1	Overflow mode	The DC bit is set if the result of an ALU or shift (PSHA) arithmetic operation exceeds the destination register range, excluding the guard bits, and is cleared otherwise. When an ALU or shift (PSHL) logical operation is executed, the DC bit is always cleared.
1	0	0	Signed greater-than mode	This mode is similar to signed greater-or-equal mode, but DC is cleared if the result is all-zeros. DC = ~{(negative value ^ over-range) | zero value}; In case of arithmetic operation DC = 0; In case of logical operation
1	0	1	Signed greater-or-equal mode	If the result of an ALU or shift (PSHA) arithmetic operation exceeds the destination register range, including the guard bits (“over-range”), the definition is the same as in negative value mode. If the result is not over-range, the definition is the opposite of that in negative value mode. When an ALU or shift (PSHL) logical operation is executed, the DC bit is always cleared. DC = ~(negative value ^ over-range); In case of arithmetic operation DC = 0; In case of logical operation
1	1	0	Reserved	
1	1	1	Reserved	

Conditional Operations and Data Transfer: Some instructions belonging to this class can be executed conditionally, as described earlier. The specified condition is valid only for the B field of the instruction, and is not valid for data transfer instructions for which a parallel specification is made. Examples are shown in figure 2.17.

```
DCT PADD X0,Y0,A0  MOVX.W @R4+,X0  MOVY.W A0,@R6+R9 ;

When condition is True

Before execution: X0=H'33333333, Y0=H'55555555, A0=H'123456789A,
                  R4=H'00008000, R6=H'00008233, R9=H'00000004
                  (R4)=H'1111, (R6)=H'2222
After execution:  X0=H'11110000, Y0=H'55555555, A0=H'0088888888,
                  R4=H'00008002, R6=H'00008237, R9=H'00000004
                  (R4)=H'1111, (R6)=H'3456

When condition is False

Before execution: X0=H'33333333, Y0=H'55555555, A0=H'123456789A,
                  R4=H'00008000, R6=H'00008233, R9=H'00000004
                  (R4)=H'1111, (R6)=H'2222
After execution:  X0=H'11110000, Y0=H'55555555, A0=H'123456789A,
                  R4=H'00008002, R6=H'00008237, R9=H'00000004
                  (R4)=H'1111, (R6)=H'3456
```

Figure 2.17 Examples of Conditional Operations and Data Transfer Instructions

Assignment of NOPX and NOPY Instruction Codes: When there is no data transfer instruction to be parallel-processed simultaneously with a DSP operation instruction, an NOPX or NOPY instruction can be written as the data transfer instruction, or the instruction can be omitted. The instruction code is the same whether an NOPX or NOPY instruction is written or the instruction is omitted. Examples of NOPX and NOPY instruction codes are shown in table 2.34.

Table 2.34 Examples of NOPX and NOPY Instruction Codes

Instruction			Code
PADD X0,Y0,A0	MOVX.W @R4+,X0	MOVY.W @R6+R9,Y0	1111100000001011
			1011000100000111
PADD X0,Y0,A0	NOPX	MOVY.W @R6+R9,Y0	1111100000000011
			1011000100000111
PADD X0,Y0,A0	NOPX	NOPY	1111100000000000
			1011000100000111
PADD X0,Y0,A0	NOPX		1111100000000000
			1011000100000111
PADD X0,Y0,A0			1111100000000000
			1011000100000111
	MOVX.W @R4+,X0	MOVY.W @R6+R9,Y0	1111000000001011
	MOVX.W @R4+,X0	NOPY	1111000000001000
	MOVS.W @R4+,X0		1111010010001000
	NOPX	MOVY.W @R6+R9,Y0	1111000000000011
		MOVY.W @R6+R9,Y0	1111000000000011
	NOPX	NOPY	1111000000000000
NOP			0000000000001001

Section 3 Memory Management Unit (MMU)

3.1 Overview

3.1.1 Features

The SH7727 has an on-chip memory management unit (MMU) that implements address translation. The SH7727 features a resident translation look-aside buffer (TLB) that caches information for user-created address translation tables located in external memory. It enables high-speed translation of logical addresses into physical addresses. Address translation uses the paging system and supports two page sizes (1 kbyte and 4 kbytes). The access right to logical address space can be set for privileged and user modes to provide memory protection.

3.1.2 Role of MMU

The MMU is a feature designed to make efficient use of physical memory. As shown in figure 3.1, if a process is smaller in size than the physical memory, the entire process can be mapped onto physical memory. However, if the process increases in size to the extent that it no longer fits into physical memory, it becomes necessary to partition the process and to map those parts requiring execution onto memory as occasion demands (1). Having the process itself consider this mapping onto physical memory would impose a large burden on the process. To lighten this burden, the idea of virtual memory was born as a means of performing en bloc mapping onto physical memory (2). In a virtual memory system, substantially more virtual memory than physical memory is provided, and the process is mapped onto this virtual memory. Thus a process only has to consider operation in virtual memory. Mapping from virtual memory to physical memory is handled by the MMU. The MMU is normally controlled by the operating system, switching physical memory to allow the virtual memory required by a process to be mapped onto physical memory in a smooth fashion. Switching of physical memory is carried out via secondary storage, etc.

The virtual memory system that came into being in this way is particularly effective in a time-sharing system (TSS) in which a number of processes are running simultaneously (3). If processes running in a TSS had to take mapping onto virtual memory into consideration while running, it would not be possible to increase efficiency. Virtual memory is thus used to reduce this load on the individual processes and so improve efficiency (4). In the virtual memory system, virtual memory is allocated to each process. The task of the MMU is to perform efficient mapping of these virtual memory areas onto physical memory. It also has a memory protection feature that prevents one process from inadvertently accessing another process's physical memory.

When address translation from virtual memory to physical memory is performed using the MMU, it may occur that the relevant translation information is not recorded in the MMU, with the result that one process may inadvertently access the virtual memory allocated to another process. In this case, the MMU will generate an exception, change the physical memory mapping, and record the new address translation information.

Although the functions of the MMU could also be implemented by software alone, the need for translation to be performed by software each time a process accesses physical memory would result in poor efficiency. For this reason, a buffer for address translation (translation look-aside buffer: TLB) is provided in hardware to hold frequently used address translation information. The TLB can be described as a cache for storing address translation information. Unlike cache memory, however, if address translation fails, that is, if an exception is generated, switching of address translation information is normally performed by software. This makes it possible for memory management to be performed flexibly by software.

The MMU has two methods of mapping from virtual memory to physical memory: a paging method using fixed-length address translation, and a segment method using variable-length address translation. With the paging method, the unit of translation is a fixed-size address space (usually of 1 to 64 kbytes) called a page.

In the following text, the SH7727 address space in virtual memory is referred to as logical address space, and address space in physical memory as physical memory space.

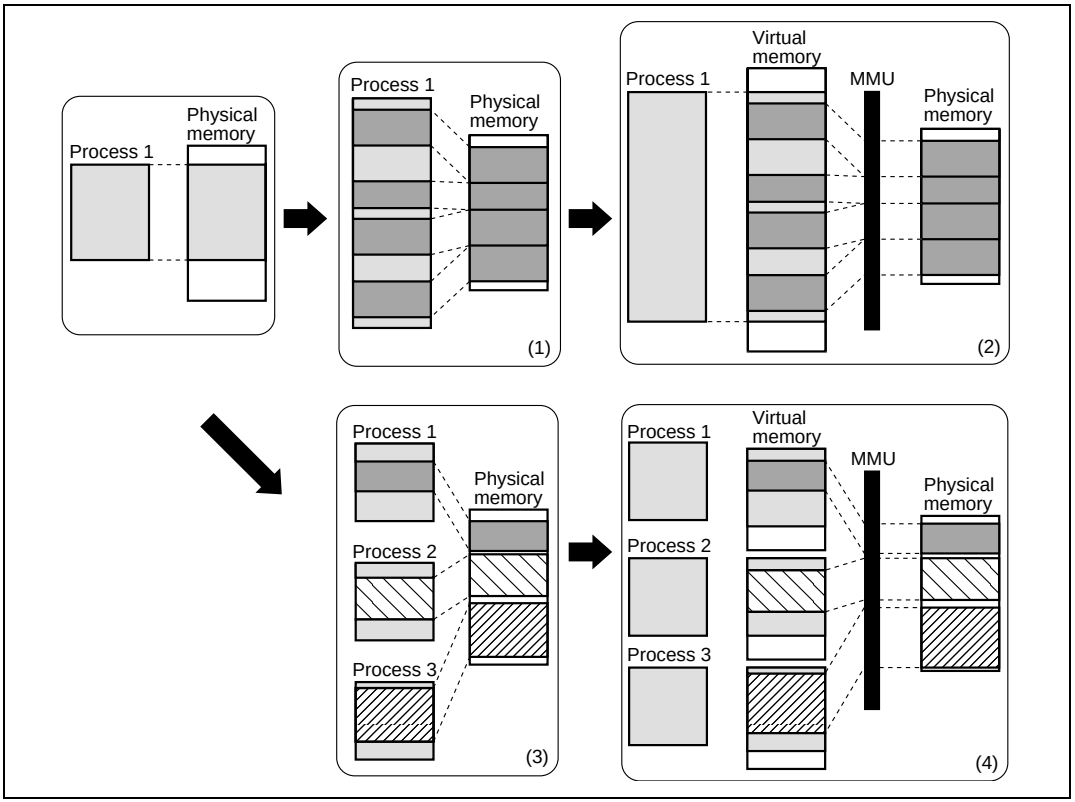


Figure 3.1 MMU Functions

3.1.3 SH7727 MMU

Logical Address Space: The SH7727 uses 32-bit logical addresses to access a 4-Gbyte logical address space that is divided into several areas. Address space mapping is shown in figure 3.2.

In the privileged mode, there are five areas, P0 to P4.

The P0 and P3 areas are mapped onto physical address space in page units, in accordance with address translation table information. Write-back or write-through can be selected for write access by means of a CCR setting.

Mapping of the P1 area is fixed in physical address space (H'00000000 to H'1FFFFFFF). In the P1 area, setting a logical address MSB (bit 31) to 0 generates the corresponding physical address. P1 area accesses can be cached, and the cache control register (CCR) is set to indicate whether to cache or not. Write-back or write-through mode can be selected.

Mapping of the P2 area is fixed to physical address space (H'00000000 to H'1FFFFFFF). In the P2 area, setting the top three logical address bits (bits 31, 30, and 29) to 0 generates the corresponding physical address. P2 area access cannot be cached.

The P1 and P2 areas are not mapped by the address translation table, so the TLB is not used and no exceptions like TLB misses occur. Initialization of MMU-related registers, exception handling, and the like are located in the P1 and P2 areas. Because the P1 area is cached, handlers that require high-speed processing are placed there.

A part of the control register in the peripheral module is allocated in area 1 of the physical address space. When the physical address space is not used for address translation, allocate that part of the control register in the P2 area. When the physical address space is used for address translation, set no caching.

The P4 area is used for mapping on-chip control register addresses.

In the user mode, 2 Gbytes of the logical address space from H'00000000 to H'7FFFFFFF (area U0) can be accessed. U0 is mapped onto physical address space in page units, in accordance with address translation table information. When SR.DSP is off, 2 Gbytes of the logical address space from H'80000000 to H'FFFFFFFF cannot be accessed in the user mode. Attempting to do so creates an address error. Write-back or write-through mode can be selected for write accesses by means of a CCR setting.

When the SR.DSP is on, a new 16-MB address space, Uxy, is defined from address H'A5000000 to H'A5FFFFFF for X/Y RAM. This Uxy space is non-cached, fixed physical address space. Any access to address space beyond U0 and Uxy creates an address error. For details on the X/Y RAM space, refer to section 6, X/Y Memory.

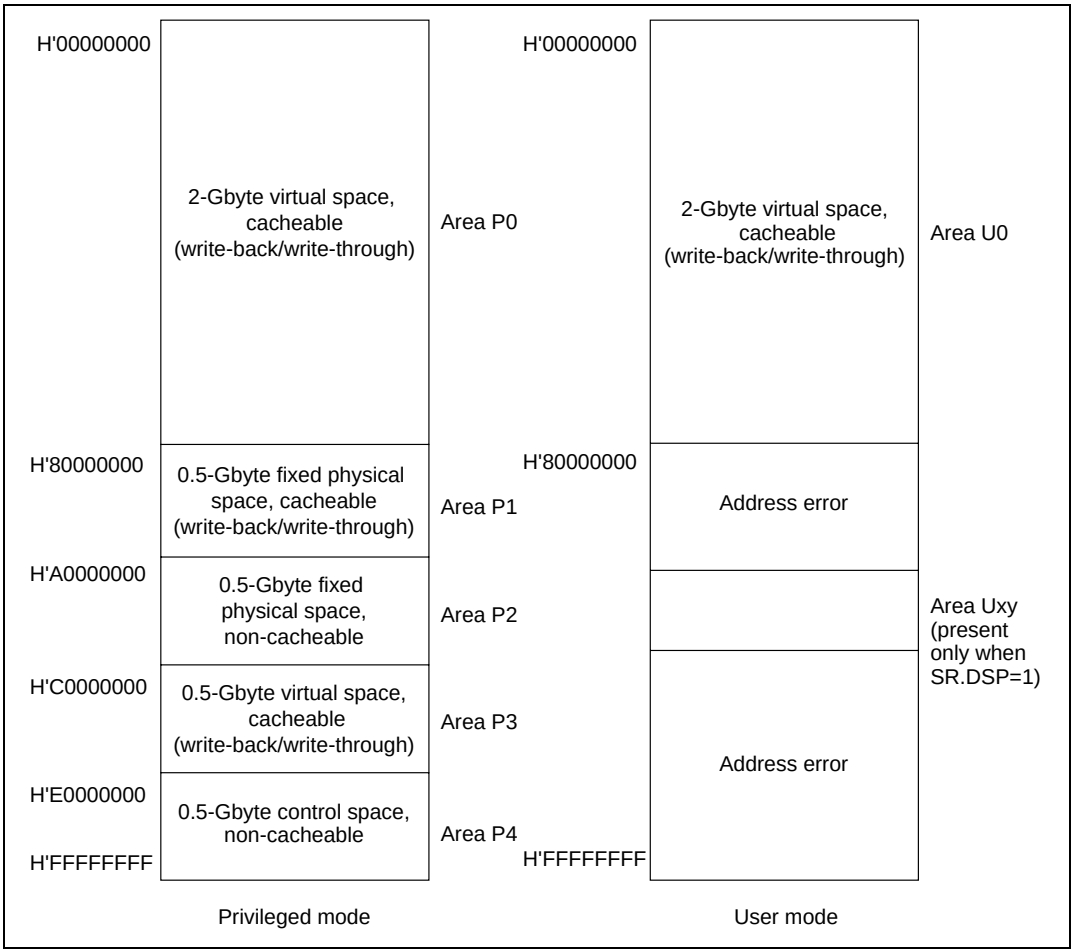


Figure 3.2 Logical Address Space Mapping

Physical Address Space: The SH7727 supports a 32-bit physical address space, but the upper 3 bits are actually ignored and treated as a shadow. See section 12, Bus State Controller (BSC), for details.

Address Translation: When the MMU is enabled, the logical address space is divided into units called pages. Physical addresses are translated in page units. Address translation tables in external memory hold information such as the physical address that corresponds to the logical address and memory protection codes. When an access to an area other than P4 occurs, if the accessed logical address belongs to area P1 or P2 there is no TLB access and the physical address is uniquely defined. If it belongs to area P0, P3 or U0, the TLB is searched by logical address and, if that logical address is registered in the TLB, the access hits the TLB. The corresponding physical

address and the page control information are read from the TLB and the physical address is determined.

If the logical address is not registered in the TLB, a TLB miss exception occurs and processing will shift to the TLB miss handler. In the TLB miss handler, the TLB address translation table in external memory is searched and the corresponding physical address and the page control information are registered in the TLB. After returning from the handler, the instruction that caused the TLB miss is re-executed. When the MMU is enabled, address translation information that results in a physical address space of H'80000000 to H'FFFFFFFF should not be registered in the TLB.

When the MMU is disabled, the logical address is used directly as the physical address. As the SH7727 supports a 29-bit address space as the physical address space, the top 3 bits of the physical address are ignored, and constitute a shadow space (see section 12, Bus State Controller (BSC)). For example, addresses H'00001000 in the P0 area, H'80001000 in the P1 area, H'A0001000 in the P2 area, and H'C0001000 in the P3 area are all mapped onto the same physical address. When access to these addresses is performed with the cache enabled, an address with the top 3 bits of the physical address masked to 0 is stored in the cache address array to ensure data congruity.

Single Virtual Memory Mode and Multiple Virtual Memory Mode: There are two virtual memory modes: single virtual memory mode and multiple virtual memory mode. In single virtual memory mode, multiple processes run in parallel using the logical address space exclusively and the physical address corresponding to a given logical address is specified uniquely. In multiple virtual memory mode, multiple processes run in parallel sharing the logical address space, so a given logical address may be translated into different physical addresses depending on the process. By the value set to the MMU control register (MMUCR), either single or multiple virtual mode is selected. In terms of operation, the only difference between single virtual memory mode and multiple virtual memory mode is in the TLB address comparison method (see section 3.3.3, TLB Address Comparison).

Address Space Identifier (ASID): In multiple virtual memory mode, the address space identifier (ASID) is used to differentiate between processes running in parallel and sharing logical address space. The ASID is 8 bits in length and can be set by software setting of the ASID of the currently running process in PTEH within the MMU. When the process is switched using the ASID, the TLB does not have to be purged.

In single virtual memory mode, the ASID is used to provide memory protection for processes running simultaneously and using the logical address space exclusively (see section 3.4.2, MMU Software Management).

3.1.4 Register Configuration

Table 3.1 shows the configuration of the MMU control registers.

Table 3.1 Register Configuration

Name	Abbreviation	R/W	Size	Initial Value*1	Address
Page table entry register high	PTEH	R/W	Longword	Undefined	H'FFFFFFF0
Page table entry register low	PTEL	R/W	Longword	Undefined	H'FFFFFFF4
Translation table base register	TTB	R/W	Longword	Undefined	H'FFFFFFF8
TLB exception address register	TEA	R/W	Longword	Undefined	H'FFFFFFFC
MMU control register	MMUCR	R/W	Longword	*2	H'FFFFFFE0

Notes: 1. Initialized by a power-on reset or manual reset.
2. SV bit = undefined
Other bits = 0

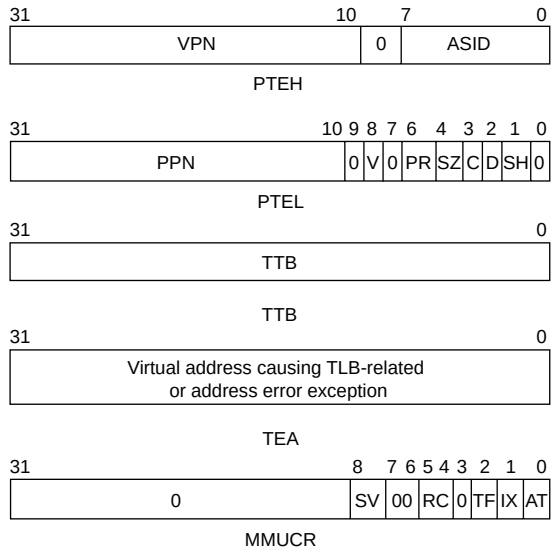
3.2 Register Description

There are five registers for MMU processing. These are all peripheral module registers, so they are located in address space area P4 and can only be accessed from privileged mode by specifying the address. These registers consist of:

1. The page table entry register high (PTEH) register residing at address H'FFFFFFF0, which consists of a virtual page number (VPN) and ASID. The VPN set is the VPN of the logical address at which the exception is generated in case of an MMU exception or address error exception. When the page size is 4 kbytes, the VPN is the upper 20 bits of the logical address, but in this case the upper 22 bits of the logical address are set. The VPN can also be modified by software. As the ASID, software sets the number of the currently executing process. The VPN and ASID are recorded in the TLB by the LDTLB instruction.
2. The page table entry register low (PTEL) register residing at address H'FFFFFFF4, and used to store the physical page number and page management information to be recorded in the TLB by the LDTLB instruction. The contents of this register are only modified in response to a software command.
3. The translation table base register (TTB) residing at address H'FFFFFFF8, which points to the base address of the current page table. The software does not set any value in TTB automatically. TTB is available to software for general purposes.

4. The TLB exception address register (TEA) residing at address H'FFFFFFFC, which stores the logical address corresponding to a TLB or address error exception. This value remains valid until the next exception or interrupt.
5. The MMU control register (MMUCR) residing at address H'FFFFFFE0, which makes the MMU settings described in figure 3.3. Any program that modifies MMUCR should reside in the P1 or P2 area.

The MMU registers are shown in figure 3.3.



0: Reserved bits. Always read as 0. Writing is ignored. However, 0 should also be specified in a write to MMUCR only.

SV: Single virtual memory mode bit. Set to 1 for the single virtual memory mode, cleared to 0 for the multiple virtual memory mode.

RC: A 2-bit random counter, automatically updated by hardware according to the following rules in the event of an MMU exception. When a TLB miss exception occurs, all TLB entry ways corresponding to the virtual address at which the exception occurred are checked, and if all ways are valid, 1 is added to RC; if there is one or more invalid way, they are set by priority from way 0, in the order: way 0, way 1, way 2, way 3. In the event of an MMU exception other than a TLB miss exception, the way which caused the exception is set in RC.

TF: TLB flush bit. Write 1 to flush the TLB (clear all valid bits of the TLB to 0). Always reads 0.

IX: Index mode bit. When 0, VPN bits 16 to 12 are used as the TLB index number. When 1, the value obtained by EX-ORing ASID bits 4 to 0 in PTEH and VPN bits 16 to 12 are used as the TLB index number.

AT: Address translation bit. Enables/disables the MMU.
0: MMU disabled
1: MMU enabled

Figure 3.3 MMU Register Contents

3.3 TLB Functions

3.3.1 Configuration of the TLB

The TLB caches address translation table information located in the external memory. The address translation table stores the physical page number translated from the virtual page number, the address space identifier, and the control information for the page, which is the unit of address translation. Figure 3.4 shows the overall TLB configuration. The TLB is 4-way set associative with 128 entries. There are 32 entries for each way. Figure 3.5 shows the configuration of logical addresses and TLB entries.

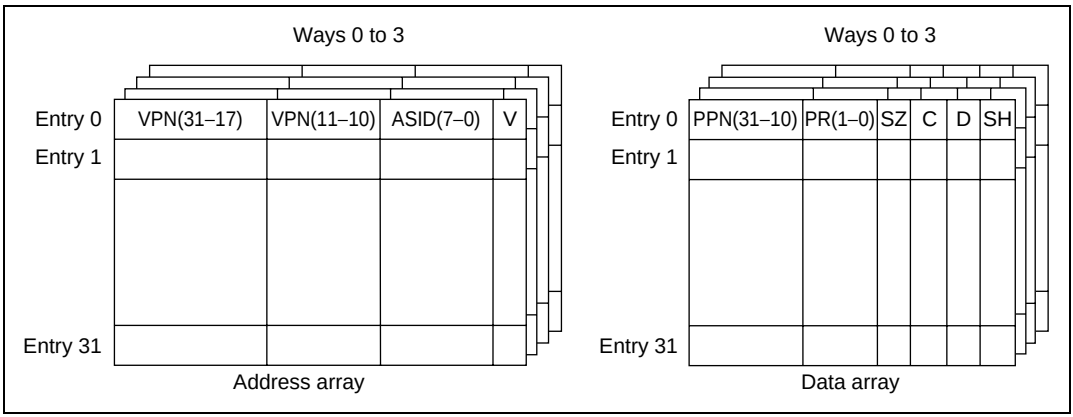
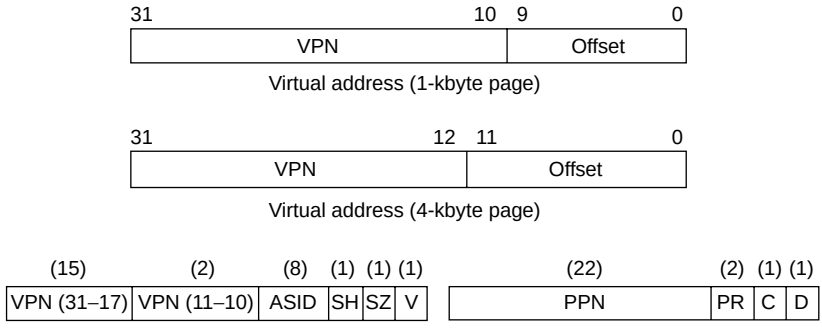


Figure 3.4 Overall Configuration of the TLB



Legend:

VPN: Virtual page number. Top 22 bits of virtual address for a 1-kbyte page, or top 20 bits of virtual address for a 4-kbyte page. Since VPN bits 16 to 12 are used as the index number, they are not stored in the TLB entry.

ASID: Address space identifier. Indicates the process that can access a virtual page. In single virtual memory mode and user mode, or in multiple virtual memory mode, if the SH bit is 0, the address is compared with the ASID in PTEH when address comparison is performed.

SH: Share status bit
0 = Page not shared between processes
1 = Page shared between processes

SZ: Page-size bit
0 = 1-kbyte page
1 = 4-kbyte page

V: Valid bit. Indicates whether entry is valid.
0 = Invalid
1 = Valid
Cleared to 0 by a power-on reset. Not affected by a manual reset.

PPN: Physical page number. Top 22 bits of physical address. PPN bits 11 and 10 are not used in case of a 4-kbyte page. Attention must be paid to the synonym problem in case of a 1-kbyte page (see section 3.4.4, Avoiding Synonym Problems).

PR: Set the most significant bit to 0.
Protection key field. 2-bit field encoded to define the access rights to the page.
00: Reading only is possible in privileged mode.
01: Reading/writing is possible in privileged mode.
10: Reading only is possible in privileged/user mode.
11: Reading/writing is possible in privileged/user mode.

C: Cacheable bit. Indicates whether the page is cacheable.
0 = Non-cacheable
1 = Cacheable

D: Dirty bit. Indicates whether the page has been written to.
0 = Not written to
1 = Written to

Figure 3.5 Logical Address and TLB Structure

3.3.2 TLB Indexing

The TLB uses a 4-way set associative scheme, so entries must be selected by index. VPN bits 16 to 12 and ASID bits 4 to 0 in PTEH are used as the index number. The index number can be generated in two different ways depending on the setting of the IX bit in MMUCR.

1. When IX = 0, VPN bits 16 to 12 alone are used as the index number
2. When IX = 1, VPN bits 16 to 12 are EX-ORed with ASID bits 4 to 0 to generate the index number

The method 1 is used to prevent lowered TLB efficiency that results when multiple processes run simultaneously in the same logical address space (multiple virtual memory) and a specific entry is selected by indexing of each process. Figures 3.6 and 3.7 show the indexing schemes.

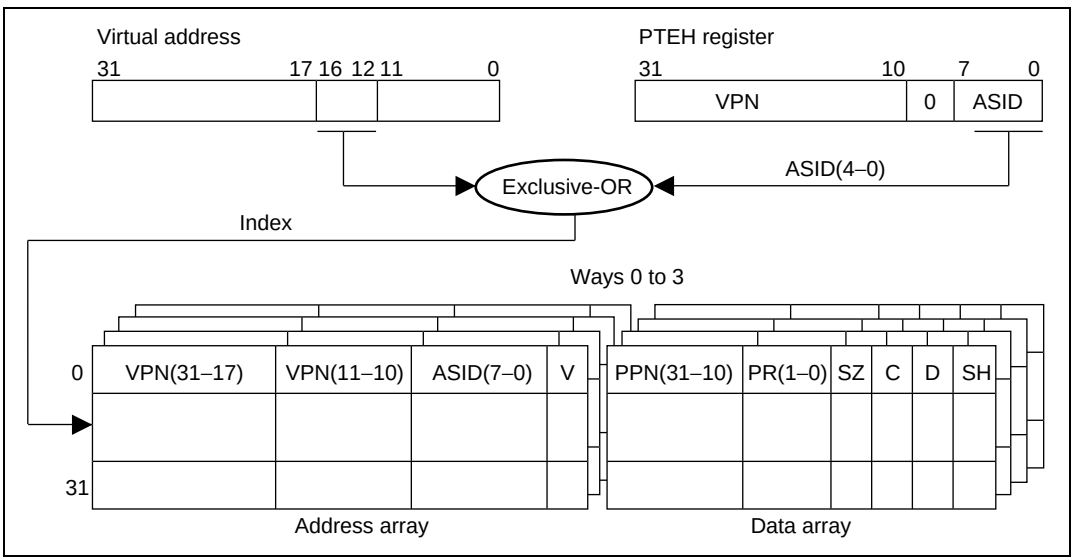


Figure 3.6 TLB Indexing (IX = 1)

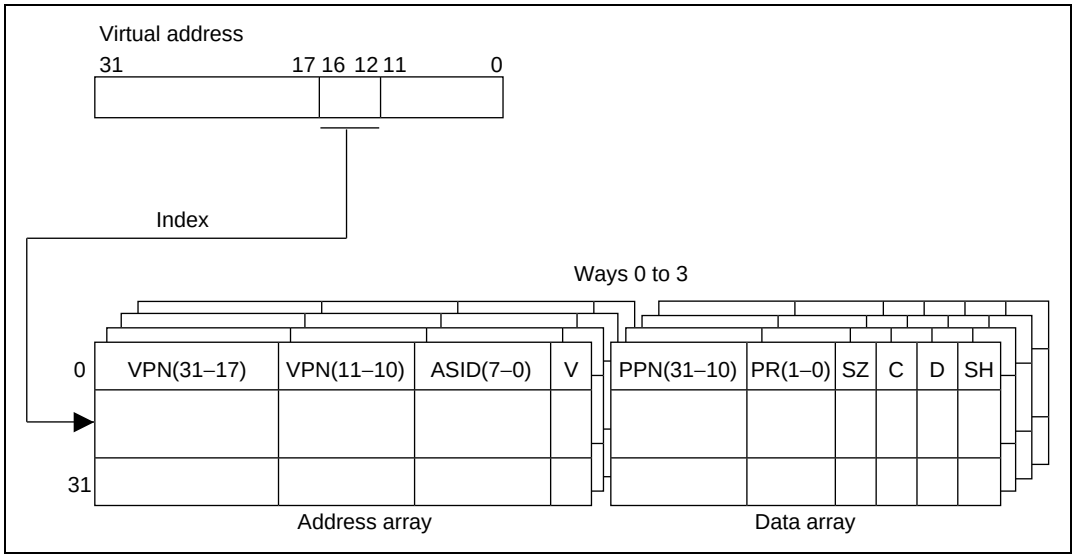


Figure 3.7 TLB Indexing (IX = 0)

3.3.3 TLB Address Comparison

A TLB address comparison is performed when an instruction is fetched from a program in external memory or data in external memory is referenced. The items used in the comparison are VPN and ASID. The VPN of the logical address that accesses external memory is compared to the VPN of the TLB entry selected with the index number. The ASID within the PTEH is compared to the ASID of the indexed TLB entry. All four ways are searched simultaneously. If the compared values match, and the indexed TLB entry is valid (V bit = 1), the hit is registered.

It is necessary to have software ensure that TLB hits do not occur simultaneously in more than one way, as hardware operation is not guaranteed if this occurs. For example, if there are two identical TLB entries with the same VPN and a setting is made such that a TLB hit is made only by a process with ASID = H'FF when one is in the shared state (SH = 1) and the other in the non-shared state (SH = 0), then if the ASID in PTEH is set to H'FF, there is a possibility of simultaneous TLB hits in both these ways. It is therefore necessary to ensure that this kind of setting is not made by software.

The object compared varies depending on the page management information (SZ, SH) in the TLB entry. It also varies depending on whether the system supports multiple virtual memory or single virtual memory.

The page-size information determines whether VPN (11, 10) is compared. VPN (11, 10) is compared for 1-kbyte pages (SZ = 0) but not for 4-kbyte pages (SZ = 1).

The sharing information (SH) determines whether the PTEH.ASID and the ASID in the TLB entry are compared. ASIDs are compared when there is no sharing between processes (SH = 0) but not when there is sharing (SH = 1).

When single virtual memory is supported (MMUCR.SV = 1) and privileged mode is engaged (SR.MD = 1), all process resources can be accessed. This means that ASIDs are not compared when single virtual memory is supported and privileged mode is engaged. The objects of address comparison are shown in figure 3.8.

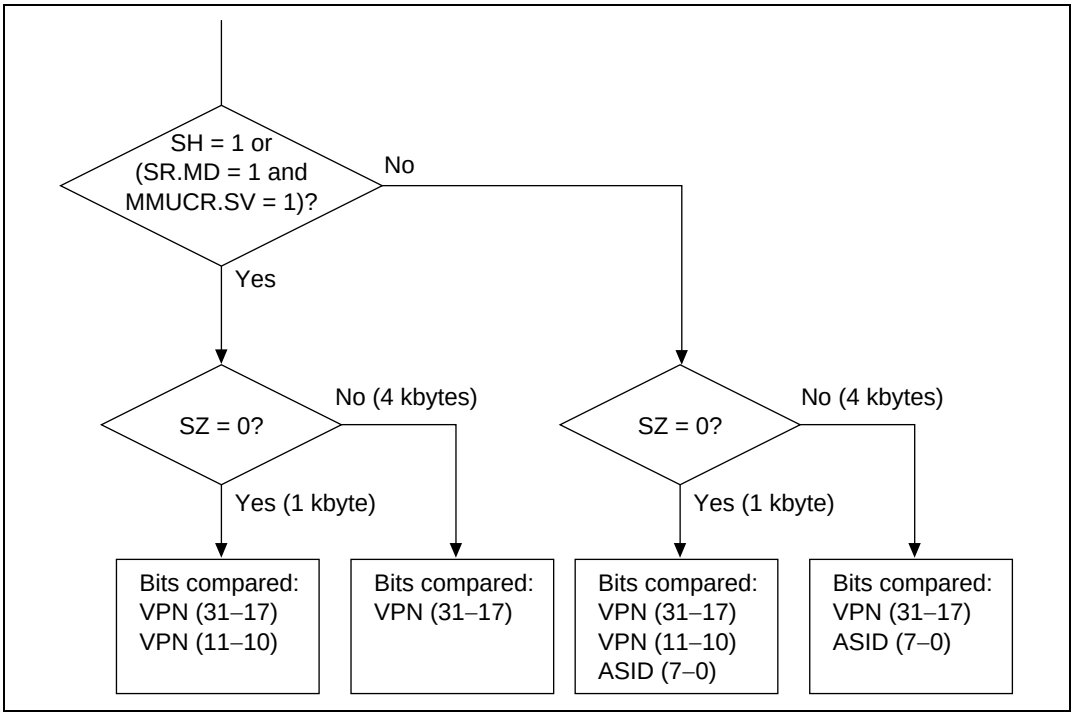


Figure 3.8 Objects of Address Comparison

3.3.4 Page Management Information

In addition to the SH and SZ bits, the page management information of TLB entries also includes D, C, and PR bits.

The D bit of a TLB entry indicates whether the page is dirty (i.e., has been written to). If the D bit is 0, an attempt to write to the page results in an initial page write exception. For physical page swapping between secondary memory and main memory, for example, pages are controlled so that a dirty page is paged out of main memory only after that page is written back to secondary memory. To record that there has been a write to a given page in the address translation table in memory, an initial page write exception is used.

The C bit in the entry indicates whether the referenced page resides in a cacheable or non-cacheable area of memory. When the control register in area 1 is mapped, set the C bit to 0.

The PR field specifies the access rights for the page in privileged and user modes and is used to protect memory. Attempts at nonpermitted accesses result in TLB protection violation exceptions.

Access states designated by the D, C, and PR bits are shown in table 3.2.

Table 3.2 Access States Designated by D, C, and PR Bits

		Privileged Mode		User Mode	
		Reading	**Writing**	**Reading**	**Writing**
D bit	0	Permitted	Initial page write exception	Permitted	Initial page write exception
	1	Permitted	Permitted	Permitted	Permitted
C bit	0	Permitted (no caching)	Permitted (no caching)	Permitted (no caching)	Permitted (no caching)
	1	Permitted (with caching)	Permitted (with caching)	Permitted (with caching)	Permitted (with caching)
PR bit	00	Permitted	TLB protection violation exception	TLB protection violation exception	TLB protection violation exception
	01	Permitted	Permitted	TLB protection violation exception	TLB protection violation exception
	10	Permitted	TLB protection violation exception	Permitted	TLB protection violation exception
	11	Permitted	Permitted	Permitted	Permitted

3.4 MMU Functions

3.4.1 MMU Hardware Management

There are two kinds of MMU hardware management as follows:

1. The MMU decodes the logical address accessed by a process and performs address translation by controlling the TLB in accordance with the MMUCR settings.
2. In address translation, the MMU receives page management information and bit information from the TLB, and determines the MMU exception and whether the cache is to be accessed (using the C bit). For details of the determination method and the hardware processing, see section 3.5, MMU Exceptions.

3.4.2 MMU Software Management

There are three kinds of MMU software management, as follows.

1. MMU register setting. MMUCR setting, in particular, should be performed in areas P1 and P2 for which address translation is not performed. Also, since SV and IX bit changes constitute address translation system changes, in this case, TLB flushing should be performed by simultaneously writing 1 to the TF bit also. Since MMU exceptions are not generated in the MMU disabled state with the AT bit cleared to 0, use in the disabled state must be avoided with software that does not use the MMU.
2. TLB entry recording, deletion, and reading. TLB entry recording can be done in two ways by using the LDTLB instruction, or by writing directly to the memory-mapped TLB. For TLB entry deletion and reading, the memory allocation TLB can be accessed. See section 3.4.3, MMU Instruction (LDTLB), for details of the LDTLB instruction, and section 3.6, Memory-Mapped TLB, for details of the memory-mapped TLB.
3. MMU exception handling. When an MMU exception is generated, it is handled on the basis of information set from the hardware side. See section 3.5, MMU Exceptions, for details.

When single virtual memory mode is used, it is possible to create a state in which physical memory access is enabled in the privileged mode only by clearing the share status bit (SH) to 0 to specify recording of all TLB entries. This strengthens inter-process memory protection, and enables special access levels to be created in the privileged mode only.

Recording a 1-kbyte page TLB entry may result in a synonym problem. See section 3.4.4, Avoiding Synonym Problems.

3.4.3 MMU Instruction (LDTLB)

The load TLB instruction (LDTLB) is used to record TLB entries. When the IX bit in MMUCR is 0, the LDTLB instruction changes the TLB entry in the way specified by the RC bit in MMUCR to the value specified by PTEH and PTEL, using VPN bits 16 to 12 specified in PTEH as the index number. When the IX bit in MMUCR is 1, the EX-OR of VPN bits 16 to 12 specified in PTEH and ASID bits 4 to 0 in PTEH are used as the index number.

Figure 3.9 shows the case where the IX bit in MMUCR is 0.

When an MMU exception occurs, the virtual page number of the logical address that caused the exception is set in PTEH by hardware. The way is set in the RC bit of MMUCR for each exception according to the rules shown in figure 3.9. Consequently, if the LDTLB instruction is issued after setting only PTEL in the MMU exception handling routine, TLB entry recording is possible. Any TLB entry can be updated by software rewriting of PTEH and the RC bits in MMUCR.

As the LDTLB instruction changes address translation information, there is a risk of destroying address translation information if this instruction is issued in the P0, U0, or P3 area. Make sure, therefore, that this instruction is issued in the P1 or P2 area. Also, an instruction associated with an access to the P0, U0, or P3 area (such as the RTE instruction) should be issued at least two instructions after the LDTLB instruction.

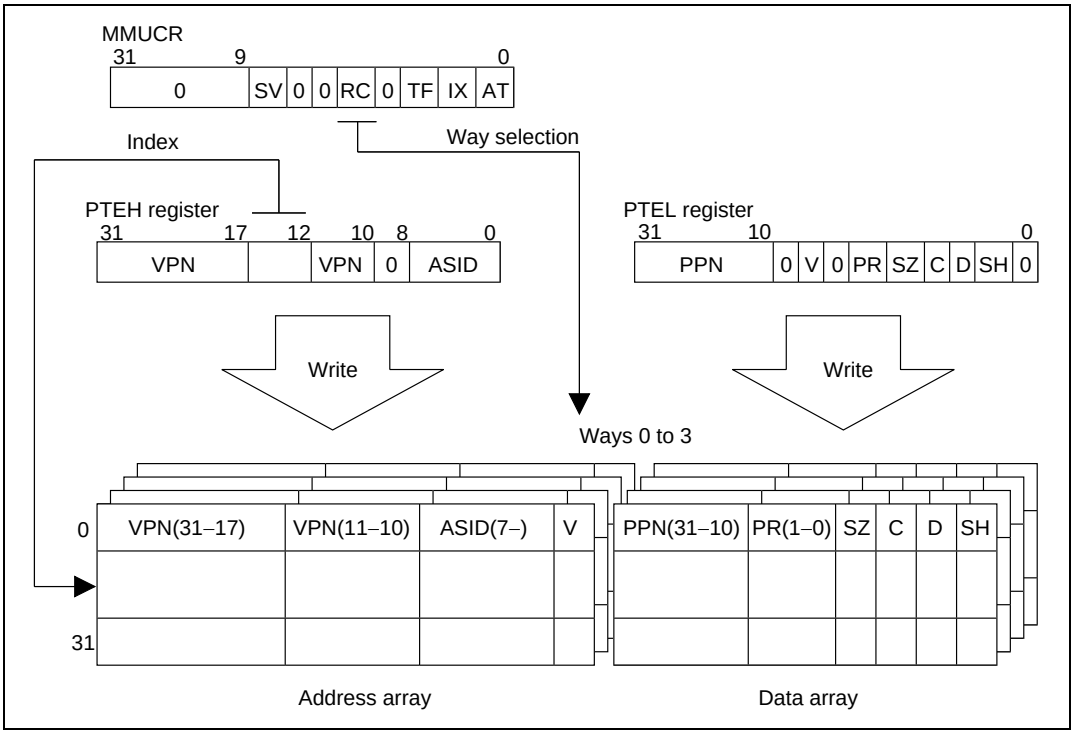


Figure 3.9 Operation of LDTLB Instruction

3.4.4 Avoiding Synonym Problems

When a 1-kbyte page is recorded in a TLB entry, a synonym problem may arise. If a number of logical addresses are mapped onto a single physical address, the same physical address data will be recorded in a number of cache entries, and it will not be possible to guarantee data congruity. The reason why this problem only occurs when using a 1-kbyte page is explained below with reference to figure 3.10.

To achieve high-speed operation of the SH7727 cache, an index number is created using logical address bits 11 to 4. When a 4-kbyte page is used, logical address bits 11 to 4 are included in the offset, and since they are not subject to address translation, they are the same as physical address bits 11 to 4. In cache-based address comparison and recording in the address array, since the cache tag address is a physical address, physical address bits 31 to 10 are recorded.

When a 1-kbyte page is used, also, a cache index number is created using logical address bits 11 to 4. However, in case of a 1-kbyte page, logical address bits 11 and 10 are subject to address translation and therefore may not be the same as physical address bits 11 and 10. Consequently,

the physical address is recorded in a different entry from that of the index number indicated by the physical address in the cache address array.

Note: When multiple address information items use the same physical memory to provide for future expansion of the SuperH RISC engine family, it is recommended that VPN[20:10] be made equal. Also, the same physical addresses should not be used with different page size address conversion information.

For example, assume that, with 1-kbyte page TLB entries, TLB entries for which the following translation has been performed are recorded in two TLBs:

Logical address 1 H'00000000 → physical address H'00000C00
Logical address 2 H'00000C00 → physical address H'00000C00

Logical address 1 is recorded in cache entry H'00, and logical address 2 in cache entry H'C0. Since two logical addresses are recorded in different cache entries despite the fact that the physical addresses are the same, memory inconsistency will occur as soon as a write is performed to either logical address.

Therefore, when recording a 1-kbyte TLB entry, if the physical address is the same as a physical address already used in another TLB entry, it should be recorded in such a way that physical address bits 11 and 10 are the same.

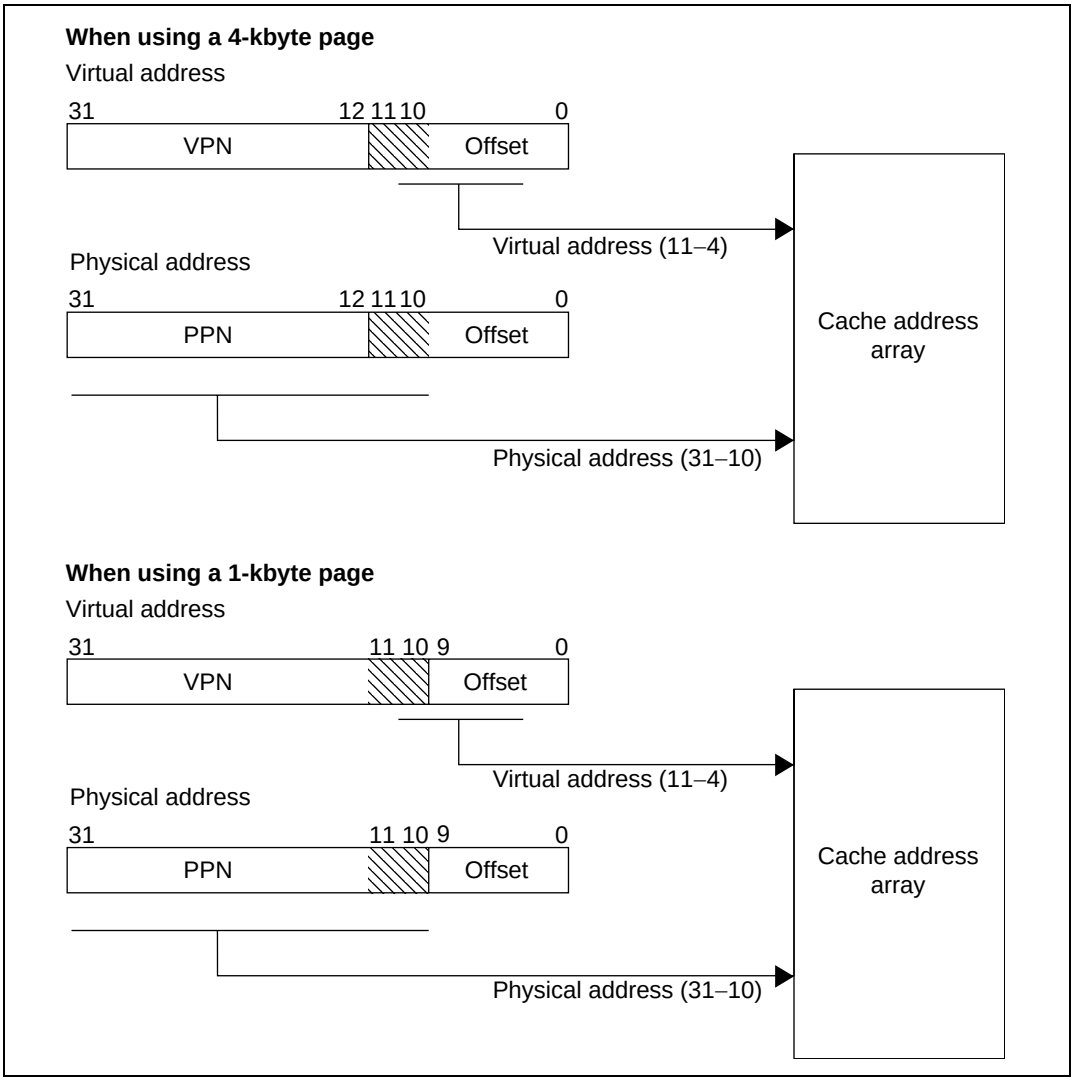


Figure 3.10 Synonym Problem

3.5 MMU Exceptions

There are four MMU exceptions: TLB miss, TLB protection violation, TLB invalid, and initial page write.

3.5.1 TLB Miss Exception

A TLB miss results when the logical address and the address array of the selected TLB entry are compared and no match is found. TLB miss exception handling includes both hardware and software operations.

Hardware Operations: In a TLB miss, the SH7727 hardware executes a set of prescribed operations, as follows:

1. The VPN field of the logical address causing the exception is written to the PTEH register.
2. The logical address causing the exception is written to the TEA register.
3. Either exception code H'040 for a load access, or H'060 for a store access, is written to the EXPEVT register.
4. The PC value indicating the address of the instruction in which the exception occurred is written to the save program counter (SPC). If the exception occurred in a delay slot, the PC value indicating the address of the related delayed branch instruction is written to the SPC.
5. The contents of the status register (SR) at the time of the exception are written to the save status register (SSR).
6. The mode (MD) bit in SR is set to 1 to place the SH7727 in the privileged mode.
7. The block (BL) bit in SR is set to 1 to mask any further exception requests.
8. The register bank (RB) bit in SR is set to 1.
9. The random counter (RC) field in the MMU control register (MMUCR) is incremented by 1 when all ways are checked for the TLB entry corresponding to the logical address at which the exception occurred, and all ways are valid. If one or more ways are invalid, those ways are set in RC in prioritized order from way 0 through way 1, way 2, and way 3.
10. Execution branches to the address obtained by adding the value of the VBR contents and H'00000400 to invoke the user-written TLB miss exception handler.

Software (TLB Miss Handler) Operations: The software searches the page tables in external memory and allocates the required page table entry. Upon retrieving the required page table entry, software must execute the following operations:

1. Write the value of the physical page number (PPN) field and the protection key (PR), page size (SZ), cacheable (C), dirty (D), share status (SH), and valid (V) bits of the page table entry

recorded in the address translation table in the external memory into the PTEL register in the SH7727.

2. If using software for way selection for entry replacement, write the desired value to the RC field in MMUCR.
3. Issue the LDTLB instruction to load the contents of PTEH and PTEL into the TLB.
4. Issue the return from exception handler (RTE) instruction to terminate the handler routine and return to the instruction stream. The RTE instruction should be issued after two LDTLB instructions.

3.5.2 TLB Protection Violation Exception

A TLB protection violation exception results when the logical address and the address array of the selected TLB entry are compared and a valid entry is found to match, but the type of access is not permitted by the access rights specified in the PR field. TLB protection violation exception handling includes both hardware and software operations.

Hardware Operations: In a TLB protection violation exception, the SH7727 hardware executes a set of prescribed operations, as follows:

1. The VPN field of the logical address causing the exception is written to the PTEH register.
2. The logical address causing the exception is written to the TEA register.
3. Either exception code H'0A0 for a load access, or H'0C0 for a store access, is written to the EXPEVT register.
4. The PC value indicating the address of the instruction in which the exception occurred is written into SPC (if the exception occurred in a delay slot, the PC value indicating the address of the related delayed branch instruction is written into SPC).
5. The contents of SR at the time of the exception are written to SSR.
6. The MD bit in SR is set to 1 to place the SH7727 in the privileged mode.
7. The BL bit in SR is set to 1 to mask any further exception requests.
8. The register bank (RB) bit in SR is set to 1.
9. The way that generated the exception is set in the RC field in MMUCR.
10. Execution branches to the address obtained by adding the value of the VBR contents and H'00000100 to invoke the TLB protection violation exception handler.

Software (TLB Protection Violation Handler) Operations: Software resolves the TLB protection violation and issues the RTE (return from exception handler) instruction to terminate the handler and return to the instruction stream. The RTE instruction should be issued after two LDTLB instructions.

3.5.3 TLB Invalid Exception

A TLB invalid exception results when the logical address is compared to a selected TLB entry address array and a match is found but the entry is not valid (the V bit is 0). TLB invalid exception handling includes both hardware and software operations.

Hardware Operations: In a TLB invalid exception, the SH7727 hardware executes a set of prescribed operations, as follows:

1. The VPN number of the logical address causing the exception is written to the PTEH register.
2. The logical address causing the exception is written to the TEA register.
3. The way number causing the exception is written to RC in MMUCR.
4. Either exception code H'040 for a load access, or H'060 for a store access, is written to the EXPEVT register.
5. The PC value indicating the address of the instruction in which the exception occurred is written to the SPC. If the exception occurred in a delay slot, the PC value indicating the address of the delayed branch instruction is written to the SPC.
6. The contents of SR at the time of the exception are written into SSR.
7. The mode (MD) bit in SR is set to 1 to place the SH7727 in the privileged mode.
8. The block (BL) bit in SR is set to 1 to mask any further exception requests.
9. The register bank (RB) bit in SR is set to 1.
10. Execution branches to the address obtained by adding the value of the VBR contents and H'00000100, and the TLB protection violation exception handler starts.

Software (TLB Invalid Exception Handler) Operations: The software searches the page tables in external memory and assigns the required page table entry. Upon retrieving the required page table entry, software must execute the following operations:

1. Write the values of the physical page number (PPN) field and the values of the protection key (PR), page size (SZ), cacheable (C), dirty (D), share status (SH), and valid (V) bits of the page table entry recorded in the external memory to the PTEL register.
2. If using software for way selection for entry replacement, write the desired value to the RC field in MMUCR.
3. Issue the LDTLB instruction to load the contents of PTEH and PTEL into the TLB.
4. Issue the RTE instruction to terminate the handler and return to the instruction stream. The RTE instruction should be issued after two LDTLB instructions.

3.5.4 Initial Page Write Exception

An initial page write exception results in a write access when the logical address and the address array of the selected TLB entry are compared and a valid entry with the appropriate access rights is found to match, but the D (dirty) bit of the entry is 0 (the page has not been written to). Initial page write exception handling includes both hardware and software operations.

Hardware Operations: In an initial page write exception, the SH7727 hardware executes a set of prescribed operations, as follows:

1. The VPN field of the logical address causing the exception is written to the PTEH register.
2. The logical address causing the exception is written to the TEA register.
3. Exception code H'080 is written to the EXPEVT register.
4. The PC value indicating the address of the instruction in which the exception occurred is written to the SPC. If the exception occurred in a delay slot, the PC value indicating the address of the related delayed branch instruction is written to the SPC.
5. The contents of SR at the time of the exception are written to SSR.
6. The MD bit in SR is set to 1 to place the SH7727 in the privileged mode.
7. The BL bit in SR is set to 1 to mask any further exception requests.
8. The register bank (RB) bit in SR is set to 1.
9. The way that caused the exception is set in the RC field in MMUCR.
10. Execution branches to the address obtained by adding the value of the VBR contents and H'00000100 to invoke the user-written initial page write exception handler.

Software (Initial Page Write Handler) Operations: The software must execute the following operations:

1. Retrieve the required page table entry from external memory.
2. Set the D bit of the page table entry in the external memory to 1.
3. Write the value of the PPN field and the PR, SZ, C, D, SH, and V bits of the page table entry in the external memory to the PTEL register.
4. If using software for way selection for entry replacement, write the desired value to the RC field in MMUCR.
5. Issue the LDTLB instruction to load the contents of PTEH and PTEL into the TLB.
6. Issue the RTE instruction to terminate the handler and return to the instruction stream. The RTE instruction should be issued after two LDTLB instructions.

Figure 3.11 shows the flowchart for MMU exceptions.

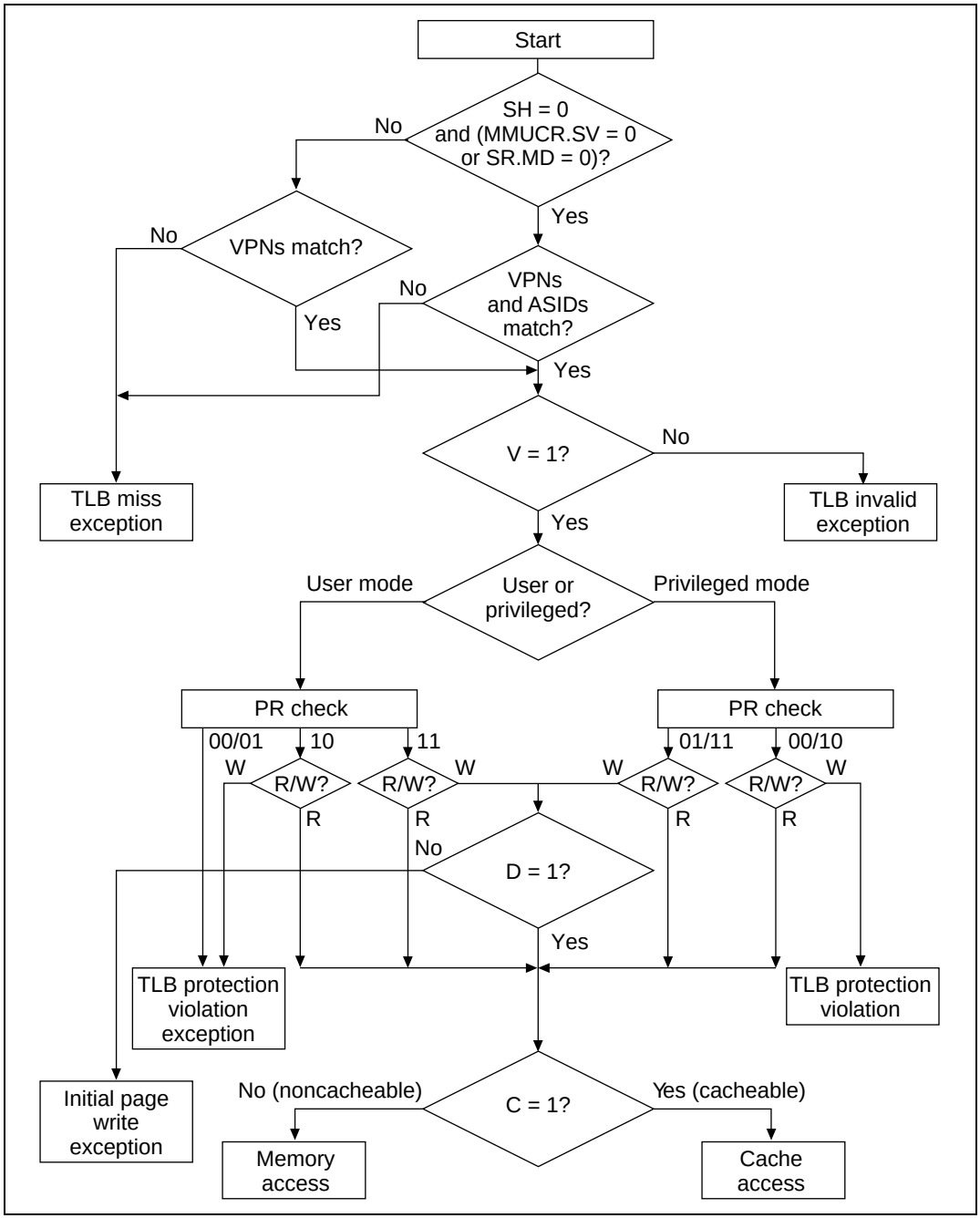


Figure 3.11 MMU Exception Generation Flowchart

3.5.5 Processing Flow in Event of MMU Exception (Same Processing Flow for Address Error)

MMU Exception in the Instruction Fetch Mode

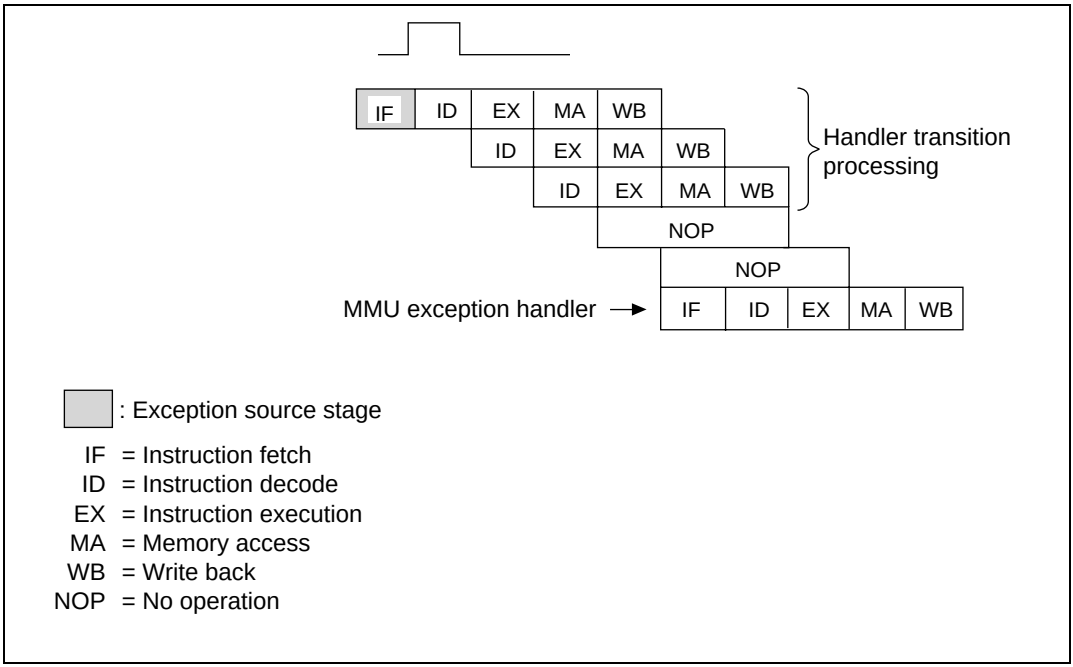


Figure 3.12 MMU Exception Signals in Instruction Fetch

MMU Exception in the Data Access Mode

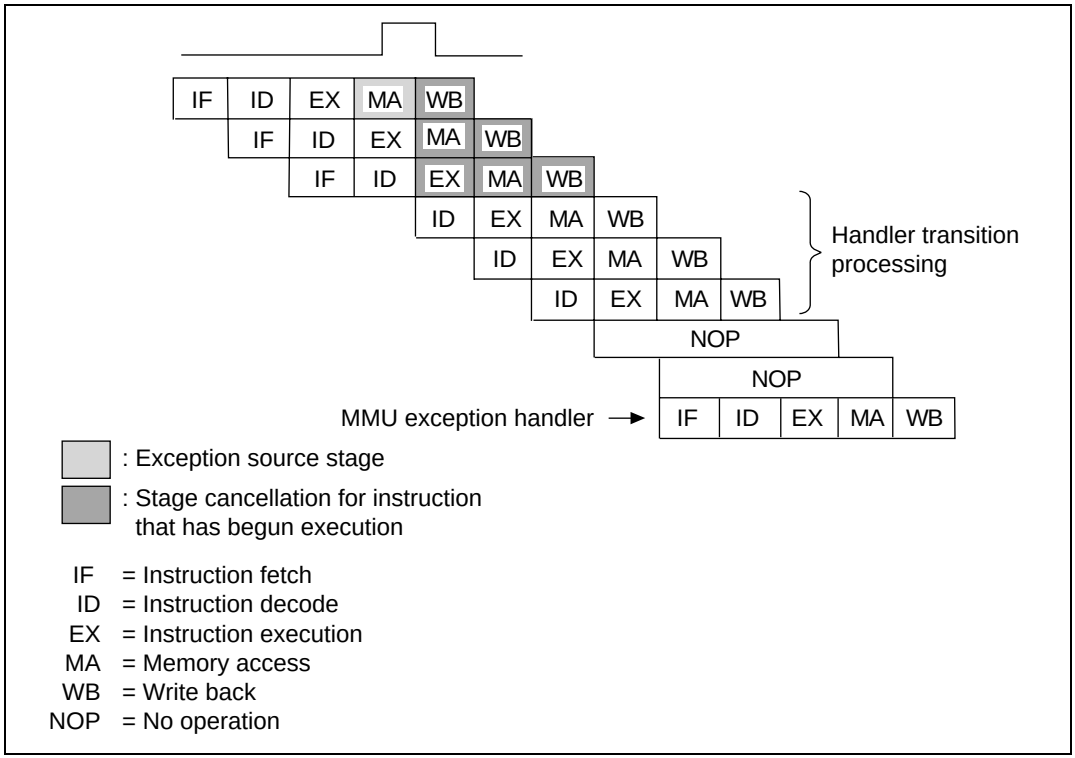


Figure 3.13 MMU Exception Signals in Data Access

3.5.6 MMU Exception in Repeat Loop

When MMU exception or CPU address error occurs immediately before or within a repeat loop, the PC of the instruction that generated the exception can not be saved in SPC correctly and repeat loop can not be restarted after returning from exception handler. EXPEVT is set to H'070 in cases of TLB miss, TLB invalid, and CPU address error. EXPEVT is set to H'0D0 in case of TLB protection violation. Figure 3.14 describes the places where this case occurs.

In a repeat loop of 4 or more instructions, only the last 4 instructions are relevant (see figure 3.14 (4)).

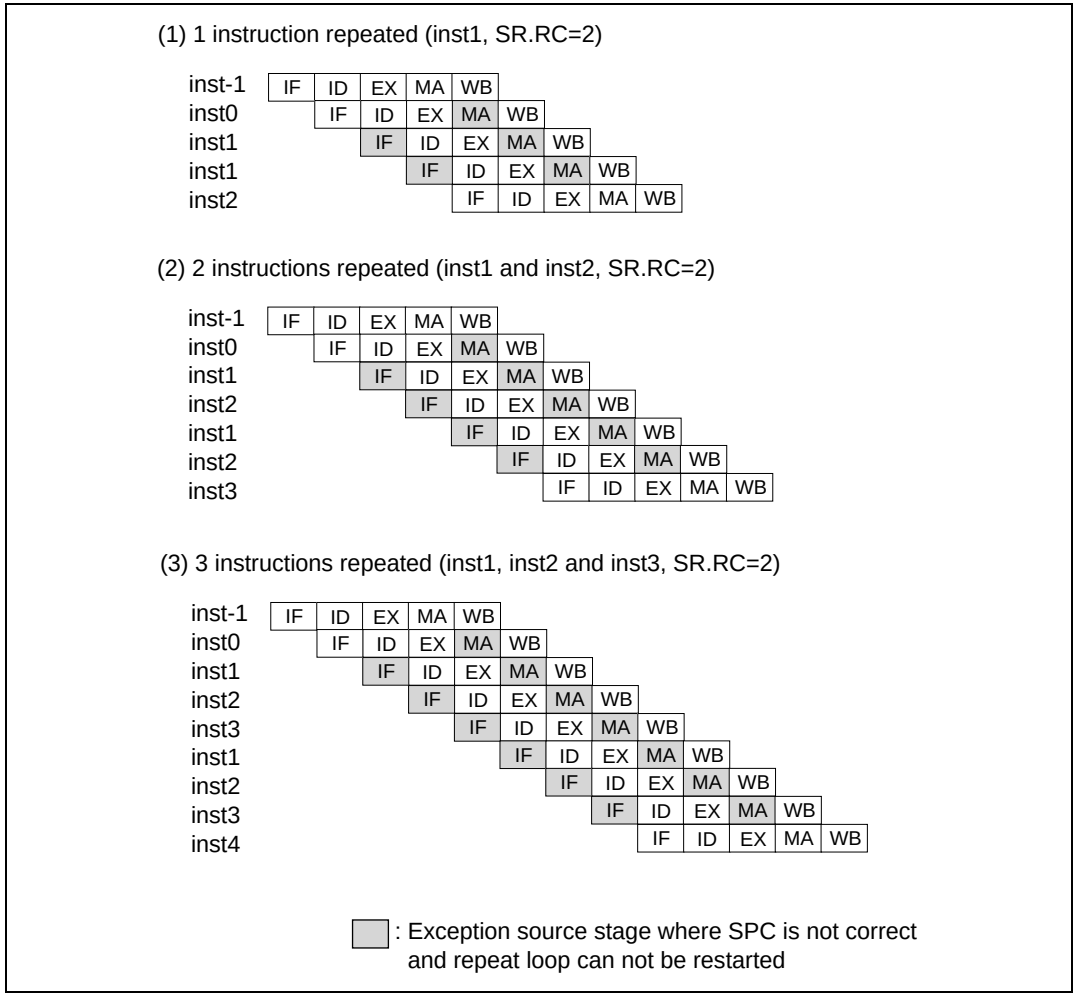


Figure 3.14 MMU Exception in Repeat Loop

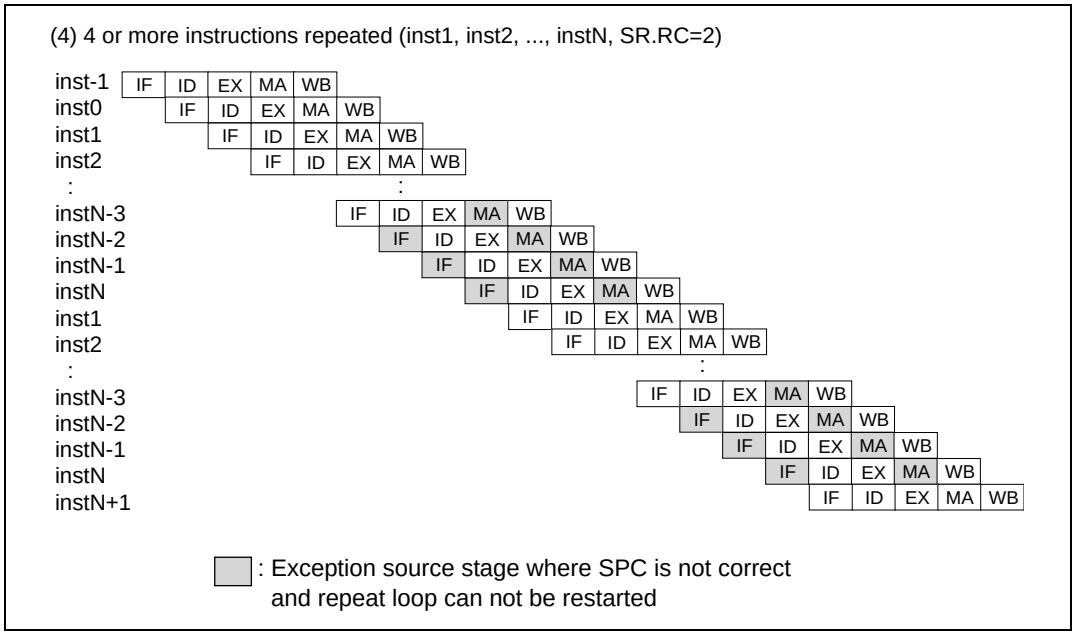


Figure 3.14 MMU Exception in Repeat Loop (cont)

3.6 Memory-Mapped TLB

In order for TLB operations to be managed by software, TLB contents can be read or written to in the privileged mode using the MOV instruction. The TLB is assigned to the P4 area in the logical address space. The TLB address array (VPN, V bit, and ASID) is assigned to H'F2000000 to H'F2FFFFFF, and the data array (PPN, PR, SZ, C, D, and SH bits) to H'F3000000 to H'F3FFFFFF. The V bit in the address array can also be accessed from the data array. Only longword access is possible for both the address array and the data array.

3.6.1 Address Array

The address array is assigned to H'F2000000 to H'F2FFFFFF. To access an address array, the 32-bit address field (for read/write operations) and 32-bit data field (for write operations) must be specified. The address field specifies information for selecting the entry to be accessed; the data field specifies the VPN, V bit and ASID to be written to the address array ((1) in figure 3.15).

In the address field, specify VPN (16 to 12) as the index address for selecting the entry (bits 16 to 12), the W bits for selecting the way (bits 9 and 8), and H'F2 to indicate address array access (bits 31 to 24). The IX bit in MMUCR indicates whether an EX-OR of VPN (16 to 12) and ASID (4 to 0) in the PTEH register is taken as the index address.

When writing, the write is performed to the entry selected with the index address and way.

When reading, the VPN, V bit, and ASID of the entry selected with the index address and way in the format of the data field in figure 3.12 without comparing addresses. 0 is written to data field bits 16 to 12.

To invalidate a specific entry, specify the entry and way, and write 0 to the corresponding V bit.

3.6.2 Data Array

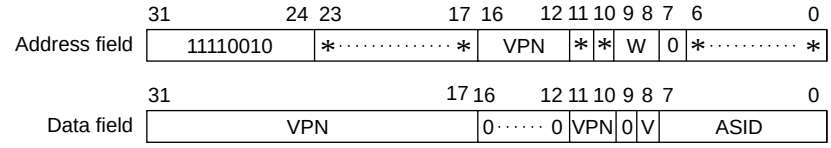
The data array is assigned to H'F3000000 to H'F3FFFFFF. To access a data array, the 32-bit address field (for read/write operations), and 32-bit data field (for write operations) must be specified. The address section specifies information for selecting the entry to be accessed; the data section specifies the longword data to be written to the data array ((2) in figure 3.15). Longword data has the same bit configuration as PTEL.

In the address field, specify VPN (16 to 12) as the index address for selecting the entry (bits 16 to 12), the W bits for selecting the way (bits 9 and 8), and H'F3 to indicate data array access (bits 31 to 24). The IX bit in MMUCR indicates whether an EX-OR of VPN (16 to 12) and ASID (4 to 0) in the PTEH register is taken as the index address.

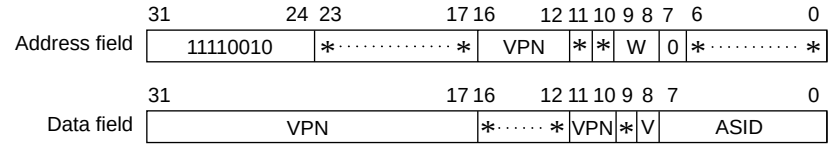
Both reading and writing use the longword of the data array specified by the entry address and way number.

(1) TLB Address Array Access

Read access



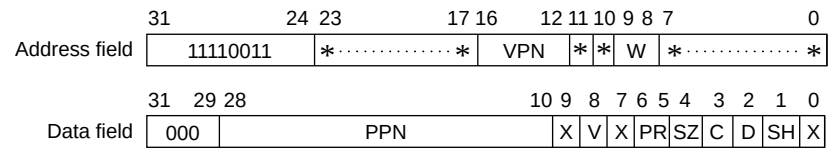
Write access



VPN: Virtual page number
V: Valid bit
W: Way (00: Way 0, 01: Way 1, 10: Way 2, 11: Way 3)
ASID: Address space identifier
*: Don't care bit

(2) TLB Data Array Access

Read/write access



PPN: Physical page number
PR: Protection key field
C: Cacheable bit
SH: Share status bit
VPN: Virtual page number
X: 0 for read, don't care bit for write
W: Way (00: Way 0, 01: Way 1, 10: Way 2, 11: Way 3)
V: Valid bit
SZ: Page-size bit
D: Dirty bit
*: Don't care bit

Figure 3.15 Specifying Address and Data for Memory-Mapped TLB Access

3.6.3 Usage Examples

Invalidating Specific Entries: Specific TLB entries can be invalidated by writing 0 to the entry's V bit. R0 specifies the write data and R1 specifies the address.

```
; R0=H'1547 381C  R1=H'F201 30
; MMUCR.IX=0
; VPN(31–17)=B'0001 0101 0100 011   VPN(11–10)=B'10   ASID=B'0001 1100
; corresponding entry association is made from the entry selected by
; the VPN(16–12)=B'1 0011 index, the V bit of the hit way is cleared to
; 0,achieving invalidation.
MOV.L  R0,@R1
```

Reading the Data of a Specific Entry: This example reads the data section of a specific TLB entry. The bit order indicated in the data field in figure 3.15 (2) is read. R0 specifies the address and the data section of a selected entry is read to R1.

```
; R1=H'F300 4300   VPN(16-12)=B'00100   Way 3
; MOV.L @R0,R1
```

3.7 Usage Notes

1. Instructions that manipulate the MD or BL bit in register SR (the LDC Rm, SR instruction, LDC @Rm+, SR instruction, and RTE instruction) and the following instruction, or the LDTLB instruction, should be used with the TLB disabled or in a fixed physical address space (the P1 or P2 space).
2. The value of the RC bit in MMUCR may be set abnormally if all of the following conditions are met:

   (1) MMU is on (AT is set to 1 in MMUCR).

   (2) Identical entries in the TLB address array reference the same VPN using multiple ways.

   (3) A TLB related exception occurs.

   The VPN is not initialized at power on reset or manual reset. Therefore, identical entries may access two or more VPNs using the same value. In such cases, certain entries in the TLB address array may end up as shown below if, for example, they are registered in way 3.
   In this case way 0 and way 3 reference the same VPN, thereby satisfying condition (2).

After reset				After registration to way 3		
WAY	VPN	V		WAY	VPN	V
0	12345	0	→	0	12345	0
3	12345	0		3	12345	1

   The above conditions can also be satisfied by TLB handling in software. For example, the situation shown below could occur if, after invalidating way 0 (by setting V from 1 to 0) for an entry in the TLB address array, the entry is registered to way 3. In this case as well, the same VPN is assigned for both way 0 and way 3, thereby satisfying condition (2) above.

After invalidation of way 0				After registration to way 3		
WAY	VPN	V		WAY	VPN	V
0	12345	0	→	0	12345	0
3	11111	0		3	12345	1

Measures to avoid the problem

The following two measures should be taken to avoid the problem described above:

a. After performing a reset and before setting AT to 1 in MMUCR, initialize to 1 the upper four bits of the VPNs for each entry in the TLB address array.

b. When invalidating an entry in the TLB address array, initialize to 1 the upper four bits of the corresponding VPN in addition to setting V to 0.

The above measures will ensure that the VPN is not in the area referenced after address conversion. This will prevent condition (3) from being satisfied and prevent the problem described above from arising.

Section 4 Exception Handling

4.1 Overview

4.1.1 Features

Exception handling is separate from normal program processing, and is performed by a routine separate from the normal program. In response to an exception handling request due to abnormal termination of the executing instruction, control is passed to a user-written exception handler. However, in response to an interrupt request, normal program execution continues until the end of the executing instruction. Here, all exceptions other than resets and interrupts will be called general exceptions. There are thus three types of exceptions: resets, general exceptions, and interrupts.

4.1.2 Register Configuration

Table 4.1 lists the registers used for exception handling. A register with an undefined initial value should be initialized by software.

Table 4.1 Register Configuration

Register	Abbr.	R/W	Size	Initial Value	Address
TRAPA exception register	TRA	R/W	Longword	Undefined	H'FFFFFFD0
Exception event register	EXPEVT	R/W	Longword	Power-on reset: H'000 Manual reset: H'020	H'FFFFFFD4
Interrupt event register	INTEVT	R/W	Longword	Undefined	H'FFFFFFD8
Interrupt event register2	INTEVT2	R	Longword	Undefined	H'04000000 (H'A4000000)*

Note: * When address translation by the MMU does not apply, the address in parentheses should be used.

4.2 Exception Handling Function

4.2.1 Exception Handling Flow

In exception handling, the contents of the program counter (PC) and status register (SR) are saved in the saved program counter (SPC) and saved status register (SSR), respectively, and execution of the exception handler is invoked from a vector address. The return from exception handler (RTE)

instruction is issued by the exception handler routine at the completion of the routine, restoring the contents of the PC and SR to return to the processor state at the point of interruption and the address where the exception occurred.

A basic exception handling sequence consists of the following operations:

1. The contents of the PC and SR are saved in the SPC and SSR, respectively.
2. The block (BL) bit in SR is set to 1, masking any subsequent exceptions.
3. The mode (MD) bit in SR is set to 1 to place the SH7727 in the privileged mode.
4. The register bank (RB) bit in SR is set to 1.
5. An exception code identifying the exception event is written to bits 11 to 0 of the exception event (EXPEVT) or interrupt event (INTEVT or INTEVT2) register.
6. Instruction execution jumps to the designated exception handling vector address to invoke the handler routine.

4.2.2 Exception Handling Vector Addresses

The reset vector address is fixed at H'A0000000. The other three events are assigned offsets from the vector base address by software. Translation look-aside buffer (TLB) miss exceptions have an offset from the vector base address of H'00000400. The vector address offset for general exception events other than TLB miss exceptions is H'00000100. The interrupt vector address offset is H'00000600. The vector base address is loaded into the vector base register (VBR) by software. The vector base address should reside in P1 or P2 fixed physical address space. Figure 4.1 shows the relationship between the vector base address, the vector offset, and the vector table.

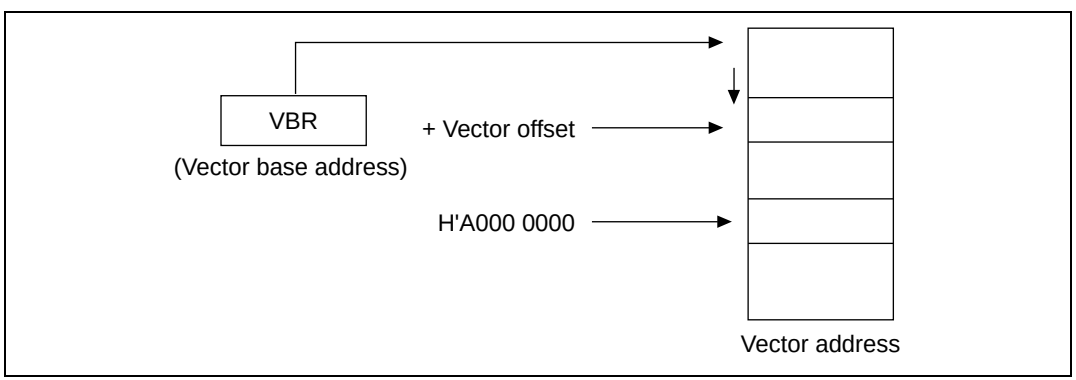


Figure 4.1 Vector Table

With regard to exceptions and their vector addresses, table 4.2 lists exception type, instruction completion state, priority, exception order, vector address, and vector offset.

Table 4.2 Exception Event Vectors

Exception Type	Current Instruction	Exception Event	Priority[*1]	Exception Order	Vector Address	Vector Offset
Reset	Aborted	Power-on reset	1	—	H'A0000000	—
		Manual reset	1	—	H'A0000000	—
		H-UDI reset	1	—	H'A0000000	—
General exception events	Aborted and retried	CPU address error (instruction access)	2	1	—	H'00000100
		TLB miss (instruction access not in repeat loop)	2	2	—	H'00000400
		TLB miss (instruction access in repeat loop)[*4]	2	2	—	H'00000100
		TLB invalid (instruction access)	2	3	—	H'00000100
		TLB protection violation (instruction access)	2	4	—	H'00000100
		General illegal instruction exception	2	5	—	H'00000100
		Illegal slot instruction exception	2	5	—	H'00000100
		CPU address error (data access)	2	6	—	H'00000100
		TLB miss (data access not in repeat loop)	2	7	—	H'00000400
		TLB miss (data access in repeat loop)[*4]	2	7	—	H'00000100
		TLB invalid (data access)	2	8	—	H'00000100
		TLB protection violation (data access)	2	9	—	H'00000100
		Initial page write	2	10	—	H'00000100
	Completed	Unconditional trap (TRAPA instruction)	2	5	—	H'00000100
		User breakpoint trap	2	n[*2]	—	H'00000100
		DMA address error	2	12	—	H'00000100

Exception Type	Current Instruction	Exception Event	Priority[*1]	Exception Order	Vector Address	Vector Offset
General interrupt requests	Completed	Nonmaskable interrupt	3	—	—	H'00000600
		External hardware interrupt	4[*3]	—	—	H'00000600
		H-UDI interrupt	4[*3]	—	—	H'00000600

Notes: 1. Priorities are indicated from high to low, 1 being highest and 4 being lowest.
2. The user defines the break point traps. 1 is a break point before instruction execution and 11 is a break point after instruction execution. For an operand break point, use 11.
3. Use software to specify relative priorities of external hardware interrupts and peripheral module interrupts (see section 7, Interrupt Controller (INTC)).
4. See section 4.5.2, General Exceptions for details.

4.2.3 Acceptance of Exceptions

Processor resets and interrupts are asynchronous events unrelated to the instruction stream. All exception events are prioritized to establish an acceptance order whenever two or more exception events occur simultaneously. When a power-on reset and a manual reset occur simultaneously, the power-on reset has priority.

All general exception events occur in a relative order in the execution sequence of an instruction (i.e., execution order), but are handled at priority level 2 in instruction-stream order (i.e., program order), where an exception detected in a preceding instruction is accepted prior to an exception detected in a subsequent instruction.

Three general exception events (general illegal instruction exception, unconditional trap exception, and illegal slot instruction exception) are detected in the decode stage (ID stage) of different instructions and are mutually exclusive events in the instruction pipeline. They have the same execution priority. Figure 4.2 shows the order of general exception acceptance.

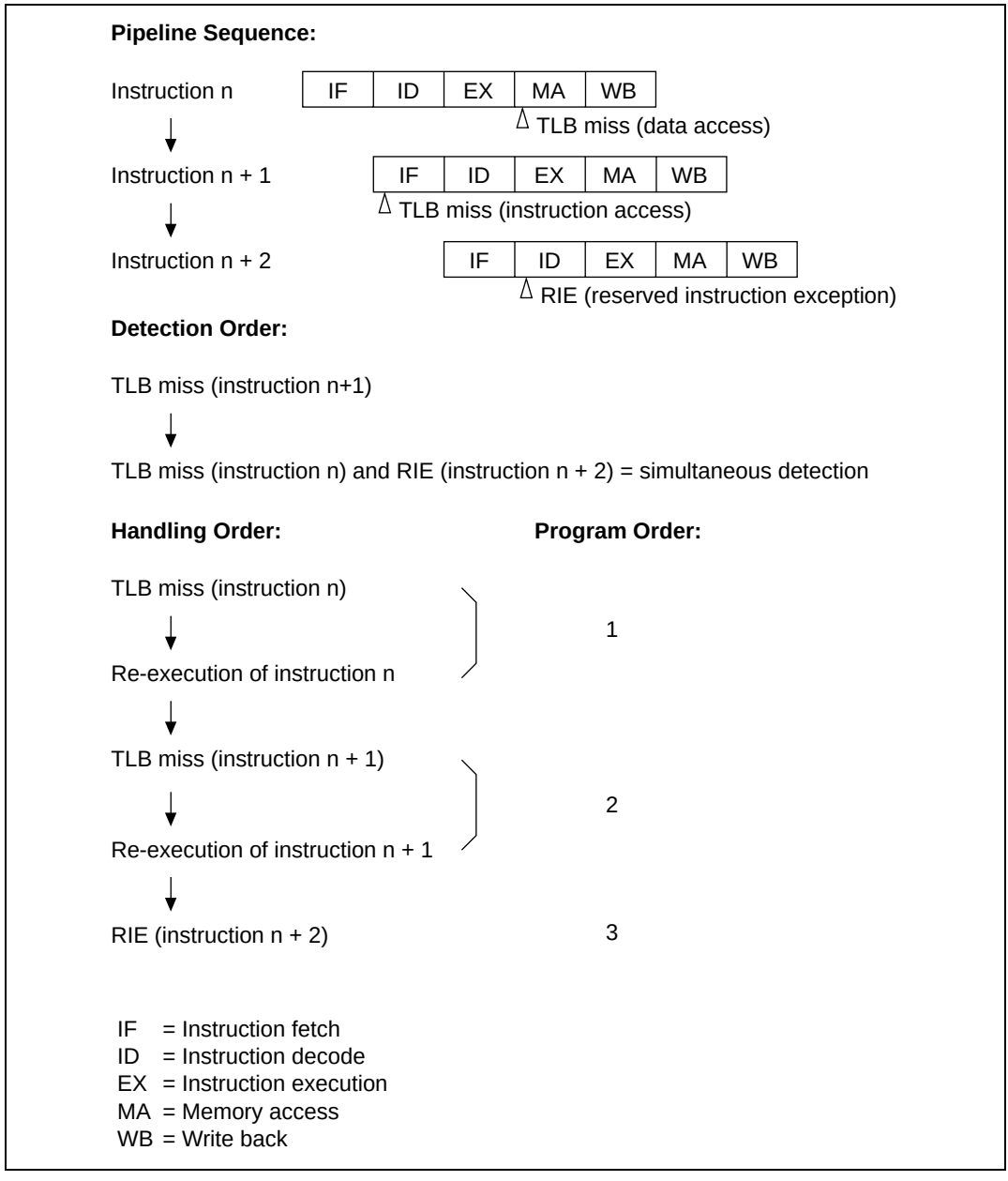


Figure 4.2 Example of Acceptance Order of General Exceptions

All exceptions other than a reset are detected in the pipeline ID stage, and accepted on instruction boundaries. However, an exception is not accepted between a delayed branch instruction and the delay slot. A re-execution type exception detected in a delay slot is accepted before execution of the delayed branch instruction. A completion type exception detected in a delayed branch instruction or delay slot is accepted after execution of the delayed branch instruction. The delay slot here refers to the next instruction after a delayed unconditional branch instruction, or the next instruction when a delayed conditional branch instruction is true.

4.2.4 Exception Codes

Table 4.3 lists the exception codes written to bits 11 to 0 of the EXPEVT register (for reset or general exceptions) or the INTEVT and INTEVT2 registers (for general interrupt requests) to identify each specific exception event. An additional exception register, the TRAPA (TRA) register, is used to hold the 8-bit immediate data in an unconditional trap (TRAPA instruction).

Table 4.3 Exception Codes

Exception Type	Exception Event	Exception Code
Reset	Power-on reset	H'000
	Manual reset	H'020
	H-UDI reset	H'000
General exception events	TLB miss/invalid (read)	H'040
	TLB miss/invalid (write)	H'060
	TLB miss/invalid/CPU Address error in repeat loop	H'070
	Initial page write	H'080
	TLB protection violation (read)	H'0A0
	TLB protection violation (write)	H'0C0
	TLB protection violation in repeat loop	H'0D0
	CPU Address error (read)	H'0E0
	CPU Address error (write)	H'100
	Unconditional trap (TRAPA instruction)	H'160
	Illegal general instruction exception	H'180
	Illegal slot instruction exception	H'1A0
	User breakpoint trap	H'1E0
	DMA address error	H'5C0

Exception Type	Exception Event	Exception Code
General interrupt requests	Nonmaskable interrupt	H'1C0
	H-UDI interrupt	H'5E0
	External hardware interrupts:	
	IRL3 to IRL0 = 0000	H'200
	IRL3 to IRL0 = 0001	H'220
	IRL3 to IRL0 = 0010	H'240
	IRL3 to IRL0 = 0011	H'260
	IRL3 to IRL0 = 0100	H'280
	IRL3 to IRL0 = 0101	H'2A0
	IRL3 to IRL0 = 0110	H'2C0
	IRL3 to IRL0 = 0111	H'2E0
	IRL3 to IRL0 = 1000	H'300
	IRL3 to IRL0 = 1001	H'320
	IRL3 to IRL0 = 1010	H'340
	IRL3 to IRL0 = 1011	H'360
	IRL3 to IRL0 = 1100	H'380
	IRL3 to IRL0 = 1101	H'3A0
	IRL3 to IRL0 = 1110	H'3C0

Note: Exception codes H'120, H'140, and H'3E0 are reserved.

4.2.5 Exception Request Masks

When the BL bit in SR is 0, exceptions and interrupts are accepted.

If a general exception event occurs when the BL bit in SR is 1, the CPU's internal registers are set to their post-reset state, other module registers retain their contents prior to the general exception, and a branch is made to the same address (H'A0000000) as for a reset.

If a general interrupt occurs when BL = 1, the request is masked (held pending) and not accepted until the BL bit is cleared to 0 by software.

For reentrant exception handling, the SPC and SSR must be saved and the BL bit in SR cleared to 0.

4.2.6 Returning from Exception Handling

The RTE instruction is used to return from exception handling. When RTE is executed, the SPC value is set in the PC, and the SSR value in SR, and the return from exception handling is performed by branching to the SPC address.

If the SPC and SSR have been saved in the external memory, set the BL bit in SR to 1, then restore the SPC and SSR, and issue an RTE instruction.

4.3 Register Description

There are four registers related to exception handling. These are peripheral module registers, and therefore reside in area P4. They can be accessed by specifying the address in the privileged mode only.

1. The exception event register (EXPEVT) resides at address H'FFFFFFD4, and contains a 12-bit exception code. The exception code set in EXPEVT is that for a reset or general exception event. The exception code is set automatically by hardware when an exception occurs. EXPEVT can also be modified by software.
2. Interrupt event register 2 (INTEVT2) resides at address H'04000000, and contains a 12-bit exception code. The exception code set in INTEVT2 is that for an interrupt request. The exception code is set automatically by hardware when an exception occurs.
3. The interrupt event register (INTEVT) resides at address H'FFFFFFD8, and contains a 12-bit interrupt exception code or a code indicating the interrupt priority. Which is set when an interrupt occurs depends on the interrupt source (see tables 7.4 and 7.5). The exception code or interrupt priority code is set automatically by hardware when an exception occurs. INTEVT can also be modified by software.
4. The TRAPA exception register (TRA) resides at address H'FFFFFFD0, and contains 8-bit immediate data (imm) for the TRAPA instruction. TRA is set automatically by hardware when a TRAPA instruction is executed. TRA can also be modified by software.

The bit configurations of the EXPEVT, INTEVT, INTEVT2, and TRA registers are shown in figure 4.3.

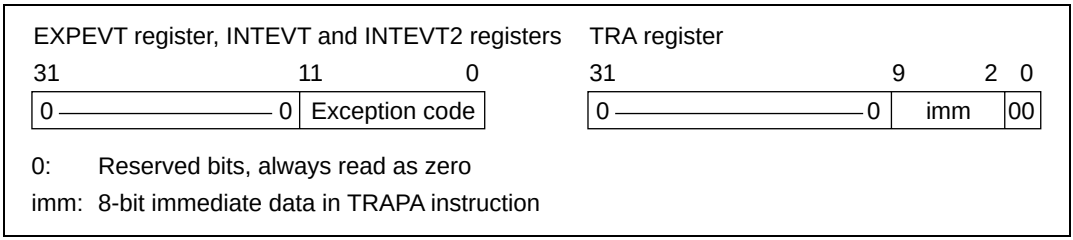


Figure 4.3 Bit Configurations of EXPEVT, INTEVT, INTEVT2, and TRA Registers

4.4 Exception Handling Operation

4.4.1 Reset

The reset sequence is used to power up or restart the SH7727 from the initialization state. The $\overline{\text{RESETP}}$ and $\overline{\text{RESETM}}$ signals are sampled every clock cycle, and in the case of a power-on reset, all processing being executed (excluding the RTC) is suspended, all unfinished events are canceled, and reset processing is executed immediately. In the case of a manual reset, however, processing to retain external memory contents is continued. The reset sequence consists of the following operations:

1. The MD bit in SR is set to 1 to place the SH7727 in privileged mode.
2. The BL bit in SR is set to 1, masking any subsequent exceptions (except NMI interrupt when BLMSK bit is 1).
3. The RB bit in SR is set to 1.
4. An encoded value of H'000 in a power-on reset or H'020 in a manual reset is written to bits 11 to 0 of the EXPEVT register to identify the exception event.
5. Instruction execution jumps to the user-written exception handler at address H'A0000000.

4.4.2 Interrupts

An interrupt processing request is accepted on completion of the current instruction. The interrupt acceptance sequence consists of the following operations:

1. The contents of the PC and SR are saved in SPC and SSR, respectively.
2. The BL bit in SR is set to 1, masking any subsequent exceptions (except NMI interrupt when BLMSK bit is 1).
3. The MD bit in SR is set to 1 to place the SH7727 in privileged mode.
4. The RB bit in SR is set to 1.

5. An encoded value identifying the exception event is written to bits 11 to 0 of the INTEVT and INTEVT2 registers.
6. Instruction execution jumps to the vector location designated by the sum of the value of the contents of the vector base register (VBR) and H'00000600 to invoke the exception handler.

4.4.3 General Exceptions

When the SH7727 encounters any exception condition other than a reset or interrupt request, it executes the following operations:

1. The contents of the PC and SR are saved in the SPC and SSR, respectively.
2. The BL bit in SR is set to 1, masking any subsequent exceptions (except NMI interrupt when BLMSK bit is 1).
3. The MD bit in SR is set to 1 to place the SH7727 in privileged mode.
4. The RB bit in SR is set to 1.
5. An encoded value identifying the exception event is written to bits 11 to 0 of the EXPEVT register.
6. Instruction execution jumps to the vector location designated by either the sum of the vector base address and offset H'00000400 in the vector table in a TLB miss trap, or by the sum of the vector base address and offset H'00000100 for exceptions other than TLB miss traps, to invoke the exception handler.

4.5 Individual Exception Operations

This section describes the conditions for specific exception handling, and the processor operations.

4.5.1 Resets

- Power-On Reset
  - Conditions: $\overline{\text{RESETP}}$ low
  - Operations: EXPEVT set to H'000, VBR and SR initialized, branch to PC = H'A0000000. Initialization sets the VBR register to H'00000000. In SR, the MD, RB and BL bits are set to 1 and the interrupt mask bits (I3 to I0) is set to 1111. The CPU and on-chip supporting modules are initialized. See the register descriptions in the relevant sections for details. A power-on reset must always be performed when powering on. A high level is output from the STATUS0 and STATUS1 pins.

- Manual Reset
  - Conditions: $\overline{\text{RESETM}}$ low
  - Operations: EXPEVT set to H'020, VBR and SR initialized, branch to PC = H'A0000000. Initialization sets the VBR register to H'00000000. In SR, the MD, RB, and BL bits are set to 1 and the interrupt mask bits (I3 to I0) is set to 1111. The CPU and on-chip supporting modules are initialized. See the register descriptions in the relevant sections for details. A high level is output from the STATUS0 and STATUS1 pins.

- H-UDI Reset
  - Conditions: H-UDI reset command input (see section 31.4.3, H-UDI Reset)
  - Operations: EXPEVT set to H'000, VBR and SR initialized, branch to PC = H'A0000000. Initialization sets the VBR register to H'0000000. In SR, the MD, RB and BL bits are set to 1 and the interrupt mask bits (I3 to I0) is set to 1111. The CPU and on-chip supporting modules are initialized. See the register descriptions in the relevant sections for details.

Table 4.4 Types of Reset

Type	Conditions for Transition to Reset State	Internal State	
		CPU	On-Chip Supporting Modules
Power-on reset	$\overline{\text{RESETP}}$ = Low	Initialized	(See register configuration in relevant sections)
Manual reset	$\overline{\text{RESETM}}$ = Low	Initialized	
H-UDI reset	H-UDI reset command input	Initialized	

4.5.2 General Exceptions

- TLB miss exception
  - Conditions: Comparison of TLB addresses shows no address match
  - Operations: The logical address (32 bits) that caused the exception is set in TEA and the corresponding virtual page number (22 bits) is set in PTEH (31 to 10). The ASID of PTEH indicates the ASID at the time the exception occurred. The RC bit in MMUCR is incremented by 1 when all ways are enabled, and if there is a disabled way, setting is prioritized starting from way 0.

  The PC and SR of the instruction that generated the exception are saved to the SPC and SSR, respectively. If the exception occurred during a read, H'040 is set in EXPEVT; if the exception occurred during a write, H'060 is set in EXPEVT. The BL, MD and RB bits in SR are set to 1 and a branch occurs to PC = VBR + H'0400.

To speed up TLB miss processing, the offset differs from other exceptions.

- TLB invalid exception
  - Conditions: Comparison of TLB addresses shows address match but V = 0.
  - Operations: The logical address (32 bits) that caused the exception is set in TEA and the corresponding virtual page number (22 bits) is set in PTEH (31 to 10). The ASID of PTEH indicates the ASID at the time the exception occurred. The way that generated the exception is set in the RC bits in MMUCR.

  The PC and SR of the instruction that generated the exception are saved in the SPC and SSR, respectively. If the exception occurred during a read, H'040 is set in EXPEVT; if the exception occurred during a write, H'060 is set in EXPEVT. The BL, MD, and RB bits in SR are set to 1 and a branch occurs to PC = VBR + H'0100.

- TLB exception/CPU address error in repeat loop
  - Conditions: TLB miss, TLB invalid or CPU address error in the last several instructions of repeat loop (see section 3.5.6, MMU Exception in Repeat Loop)
  - Operations: TEA, PTEH and RC bit in MMUCR are set in the way of the type of exception.

  The SR of the instruction that generated the exception are saved in the SSR. But the SPC is not the PC of the instruction that generated the exception. Repeat loop can not be restarted after returning from exception handler. In order to complete a repeat loop, ensure not to cause TLB exceptions or CPU address error in the last several instructions of repeat loop (see section 3.5.6, MMU Exception in Repeat Loop). If the TLB exception or CPU address error occurred in the last several instructions of repeat loop, H'070 is set in EXPEVT. The BL, MD, and RB bits in SR are set to 1 and a branch occurs to PC = VBR + H'0100.

- Initial page write exception
  - Conditions: A hit occurred to the TLB for a store access, but D = 0.
    This occurs for initial writes to the page registered by the load.
  - Operations: The logical address (32 bits) that caused the exception is set in TEA and the corresponding virtual page number (22 bits) is set in PTEH (31 to 10). The ASID of PTEH indicates the ASID at the time the exception occurred. The way that generated the exception is set in the RC bit in MMUCR.

  The PC and SR of the instruction that generated the exception are saved to the SPC and SSR, respectively. H'080 is set in EXPEVT. The BL, MD, and RB bits in SR are set to 1 and a branch occurs in PC = VBR + H'0100.

- TLB protection exception
  - Conditions: When a hit access violates the TLB protection information (PR bits) shown below:

PR	Privileged mode	User mode
00	Only read enabled	No access
01	Read/write enabled	No access
10	Only read enabled	Only read enabled
11	Read/write enabled	Read/write enabled

  - Operations: The logical address (32 bits) that caused the exception is set in TEA and the corresponding virtual page number (22 bits) is set in PTEH (31 to 10). The ASID of PTEH indicates the ASID at the time the exception occurred. The way that generated the exception is set in the RC bits in MMUCR.

  The PC and SR of the instruction that generated the exception are saved to the SPC and SSR, respectively. If the exception occurred during a read, H'0A0 is set in EXPEVT; if the exception occurred during a write, H'0C0 is set in EXPEVT. The BL, MD, and RB bits in SR are set to 1 and a branch occurs to PC = VBR + H'0100.

- TLB protection violation in repeat loop
  - Conditions: TLB protection violation in the last several instruction of repeat loop (see section 3.5.6, MMU Exception in Repeat Loop)
  - Operations: TEA, PTEH and RC bit in MMUCR are set in the way of the type of exception.

  The SR of the instruction that generated the exception are saved in the SSR. But the SPC is not the PC of the instruction that generated the exception. Repeat loop can not be restarted after returning from exception handler. In order to complete a repeat loop, ensure not to cause TLB exceptions or CPU address error in the last several instructions of repeat loop (see section 3.5.6, MMU Exception in Repeat Loop). If a TLB protection violation occurs in an instruction immediately before or during a repeat loop, H'0D0 is set in EXPEVT. The BL, MD, and RB bits in SR are set to 1 and a branch occurs to PC = VBR + H'0100.

- CPU Address error
  - Conditions:
    a. Instruction fetch from odd address (4n + 1, 4n + 3)
    b. Word data accessed from addresses other than word boundaries (4n + 1, 4n + 3)
    c. Longword accessed from addresses other than longword boundaries (4n + 1, 4n + 2, 4n + 3)
    d. Virtual space accessed in user mode in the area H'80000000 to H'FFFFFFFF.

— Operations: The logical address (32 bits) that caused the exception is set in TEA. The PC and SR of the instruction that generated the exception are saved to the SPC and SSR, respectively. If the exception occurred during a read, H'0E0 is set in EXPEVT; if the exception occurred during a write, H'100 is set in EXPEVT. The BL, MD, and RB bits in SR are set to 1 and a branch occurs to PC = VBR + H'0100. See section 3.5.5, Processing Flow in Event of MMU Exception, for more information.

- Unconditional trap
  — Conditions: TRAPA instruction executed
  — Operations: The exception is a processing-completion type, so the PC of the instruction after the TRAPA instruction is saved to the SPC. SR from the time when the TRAPA instruction was executing is saved to SSR. The 8-bit immediate value in the TRAPA instruction is quadrupled and set in TRA (9 to 0). H'160 is set in EXPEVT. The BL, MD, and RB bits in SR are set to 1 and a branch occurs to PC = VBR + H'0100.

- Illegal general instruction exception
  — Conditions:
    a. When undefined code not in a delay slot is decoded

       Delay branch instructions: JMP, JSR, BRA, BRAF, BSR, BSRF, RTS, RTE, BT/S, BF/S

       Undefined instruction: H'Fxxx

    b. When a privileged instruction not in a delay slot is decoded in user mode

       Privileged instructions: LDC, STC, RTE, LDTLB, SLEEP; instructions that access GBR with LDC/STC are not privileged instructions.

    c. When a DSP instruction not in a delay slot is decoded without DSP extension (SR.DSP=0)

       DSP instructions: LDS Rm, DSR/A0/X0/X1/Y0/Y1, LDS.L @Rm+, DSR/A0/X0/X1/Y0/Y1, STS DSR/A0/X0/X1/Y0/Y1, Rn, STS.L DSR/A0/X0/X1/Y0/Y1, @-Rn, LDC Rm, RS/RE/MOD, LDC.L @Rm+, RS/RE/MOD, STC RS/RE/MOD, Rn, STC.L RS/RE/MOD, @-Rn, LDRS, LDRE, SETRC, MOVS, MOVX, MOVY, Pxxx

    d. When an instruction that rewrites the PC/SR/RS/RE in the last three instructions of repeat loop is decoded.

       Instructions that rewrite the PC: JMP, JSR, BRA, BRAF, BSR, BSRF, RTS, RTE, BT, BF, BT/S, BF/S, TRAPA, LDC Rm, SR, LDC.L @Rm+, SR

       Instructions that rewrite the SR: LDC Rm, SR, LDC.L @Rm+, SR, SETRC

       Instructions that rewrite the RS: LDC Rm, RS, LDC.L @Rm+, RS, LDRS

       Instructions that rewrite the RE: LDC Rm, RE, LDC.L @Rm+, RE, LDRE

— Operations: The PC and SR of the instruction that generated the exception are saved to the SPC and SSR, respectively. H'180 is set in EXPEVT. The BL, MD, and RB bits in SR are set to 1 and a branch occurs to PC = VBR + H'0100. When an undefined instruction other than H'Fxxx is decoded, operation cannot be guaranteed.

- Illegal slot instruction
  — Conditions:
    a. When undefined code in a delay slot is decoded

       Delay branch instructions: JMP, JSR, BRA, BRAF, BSR, BSRF, RTS, RTE, BT/S, BF/S, Undefined instruction: H'Fxxx

    b. When an instruction that rewrites the PC in a delay slot is decoded

       Instructions that rewrite the PC: JMP, JSR, BRA, BRAF, BSR, BSRF, RTS, RTE, BT, BF, BT/S, BF/S, TRAPA, LDC Rm, SR, LDC.L @Rm+, SR

    c. When a privileged instruction in a delay slot is decoded in user mode

       Privileged instructions: LDC, STC, RTE, LDTLB, SLEEP; instructions that access GBR with LDC/STC are not privileged instructions.

    d. When a DSP instruction in a delay slot is decoded without DSP extension (SR.DSP=0)

       DSP instructions: LDS Rm, DSR/A0/X0/X1/Y0/Y1, LDS.L @Rm+, DSR/A0/X0/X1/Y0/Y1, STS DSR/A0/X0/X1/Y0/Y1, Rn, STS.L DSR/A0/X0/X1/Y0/Y1, @-Rn, LDC Rm, RS/RE/MOD, LDC.L @Rm+, RS/RE/MOD, STC RS/RE/MOD, Rn, STC.L RS/RE/MOD, @-Rn, LDRS, LDRE, SETRC, MOVS, MOVX, MOVY, Pxxx

  — Operations: The PC of the previous delay branch instruction is saved to the SPC. SR of the instruction that generated the exception is saved to SSR. H'1A0 is set in EXPEVT. The BL, MD, and RB bits in SR are set to 1 and a branch occurs to PC = VBR + H'0100. When an undefined instruction other than H'Fxxx is decoded, operation cannot be guaranteed.

- User break point trap
  — Conditions: When a break condition set in the user break controller is satisfied
  — Operations: When a post-execution break occurs, the PC of the instruction immediately after the instruction that set the break point is set in the SPC. If a pre-execution break occurs, the PC of the instruction that set the break point is set in the SPC. SR when the break occurs is set in SSR. H'1E0 is set in EXPEVT. The BL, MD, and RB bits in SR are set to 1 and a branch occurs to PC = VBR + H'0100. See section 8, User Break Controller (UBC), for more information.

- DMA Address error
  - Conditions:
    a. Word data accessed from addresses other than word boundaries (4n + 1, 4n + 3)
    b. Longword accessed from addresses other than longword boundaries (4n + 1, 4n + 2, 4n + 3)
  - Operations: The PC of the instruction immediately after the instruction executed before the exception occurs is saved to the SPC. SR when the exception occurs is saved to SSR. H'5C0 is set in EXPEVT. The BL, MD, and RB bits in SR are set to 1 and a branch occurs to PC = VBR + H'0100.

4.5.3 Interrupts

1. NMI

   Conditions: NMI pin edge detection

   Operations: The PC and SR after the instruction that receives the interrupt are saved to the SPC and SSR, respectively. H'1C0 is set to INTEVT and INTEVT2. The BL, MD, and RB bits of the SR are set to 1 and a branch occurs to PC = VBR + H'0600. This interrupt is not masked by SR.IMASK and received with top priority when the SR's BL bit in SR is 0. When the BL bit is 1, the interrupt is masked. See section 7, Interrupt Controller (INTC), for more information.

2. IRL Interrupts

   Conditions: The value of the interrupt mask bits in SR is lower than the IRL3 to IRL0 level and the BL bit in SR is 0. The interrupt is accepted at an instruction boundary.

   Operations: The PC value after the instruction at which the interrupt is accepted is saved to the SPC. SR at the time the interrupt is accepted is saved to SSR. The code corresponding to the IRL3 to IRL0 level is set in INTEVT and INTEVT2. The corresponding code is given as H'200 + [IRL3 to IRL0] × H'20. See table 7.5, for the corresponding codes. The BL, MD, and RB bits in SR are set to 1 and a branch occurs to VBR + H'0600. The received level is not set in SR.IMASK. See section 7, Interrupt Controller (INTC), for more information.

3. IRQ Pin Interrupts

   Conditions: IRQ pin is asserted and SR.IMASK is lower than the IRQ priority level and the BL bit in SR is 0. The interrupt is accepted at an instruction boundary.

   Operations: The PC value after the instruction at which the interrupt is accepted is saved to the SPC. The SR at the point the interrupt is accepted is saved to the SSR. The code corresponding to the interrupt source is set to INTEVT and INTEVT2. The BL, MD, and RB bits of the SR are set to 1 and a branch occurs to VBR + H'0600. The received level is not set to SR.IMASK. See section 7, Interrupt Controller (INTC), for more information.

4. PINT Pin Interrupts

   Conditions: The PINT pin is asserted and SR.IMASK is lower than the PINT priority level and the BL bit in SR is 0. The interrupt is accepted at an instruction boundary.

   Operations: The PC value after the instruction at which the interrupt is accepted is saved to the SPC. The SR at the point the interrupt is accepted is saved to the SSR. The code corresponding to the interrupt source is set to INTEVT and INTEVT2. The BL, MD, and RB bits of the SR are set to 1 and a branch occurs to VBR + H'0600. The received level is not set to SR.IMASK. See section 7, Interrupt Controller (INTC), for more information.

5. On-Chip Peripheral Interrupts

   Conditions: SR.IMASK is lower than the on-chip module (TMU, RTC, SCI, SIOF, SCIF, A/D, DMAC, CPG, REF, PCC, USBH, USBF, LCDC, AFEIF) interrupt level and the BL bit in SR is 0. The interrupt is accepted at an instruction boundary.

   Operations: The PC value after the instruction at which the interrupt is accepted is saved to the SPC. The SR at the point the interrupt is accepted is saved to the SSR. The code corresponding to the interrupt source is set to INTEVT and INTEVT2. The BL, MD, and RB bits of the SR are set to 1 and a branch occurs to VBR + H'0600. See section 7, Interrupt Controller (INTC), for more information.

6. H-UDI Interrupt

   Conditions: H-UDI interrupt command is input (see section 31.4.4, H-UDI Interrupt), the value of the interrupt mask bits of SR is lower than 15, and the BL bit in SR is 0, the interrupt is accepted at an instruction boundary.

   Operations: The PC after the instruction that accepts the interrupt is saved to the SPC. The SR at the point the interrupt is accepted is saved to the SSR. H'5E0 is set to INTEVT and INTEVT2. The BL, MD, and RB bits of the SR are set to 1 and a branch occurs to VBR + H'0600. See section 7, Interrupt Controller (INTC), for more information.

4.6 Usage Notes

- Return from exception handling
  - Check the BL bit in SR with software. When the SPC and SSR have been saved to external memory, set the BL bit in SR to 1 before restoring them.
  - Issue an RTE instruction. Set the SPC in the PC and SSR in SR with the RTE instruction, branch to the SPC address, and return from exception handling.

- Operation when exception or interrupt occurs while SR.BL = 1
  - Interrupt: Acceptance is suppressed until the BL bit in SR is set to 0 by software. If there is a request and the reception conditions are satisfied, the interrupt is accepted after the

execution of the instruction that sets the BL bit in SR to 0. During the sleep or standby mode, however, the interrupt will be accepted even when the BL bit in SR is 1.

NMI is accepted when BLMSK in ICR1 is 1, regardless of the setting of the BL bit.

— Exception: No user break point trap will occur even when the break conditions are met. When one of the other exceptions occurs, a branch is made to the fixed address of the reset (H'A0000000). In this case, the values of the EXPEVT, SPC, and SSR registers are undefined.

- SPC when an Exception Occurs: The PC saved to the SPC when an exception occurs is as shown below:
  — Re-executing-type exceptions: The PC of the instruction that caused the exception is set in the SPC and re-executed after return from exception handling. If the exception occurred in a delay slot, however, the PC of the immediately prior delayed branch instruction is set in the SPC. If the condition of the conditional delayed branch instruction is not satisfied, the delay slot PC is set in SPC.
  — Completed-type exceptions and interrupts: The PC of the instruction after the one that caused the exception is set in the SPC. If the exception was caused by a delayed conditional branch instruction, however, the branch destination PC is set in SPC. If the condition of the conditional delayed branch instruction is not satisfied, the delay slot PC is set in SPC.

- Initial register values after reset
  — Undefined registers
  R0_BANK0/1 to R7_BANK0/1, R8 to R15, GBR, SPC, SSR, MACH, MACL, PR
  — Initialized registers
  VBR = H'00000000
  SR.MD = 1, SR.BL = 1, SR.RB = 1, SR.I3 to SR.I0 = H'F. Other SR bits are undefined.
  PC = H'A0000000

- Ensure that an exception is not generated at an RTE instruction delay slot, as operation is not guaranteed in this case.

- When the BL bit in the SR register is set to 1, ensure that a TLB-related exception or address error does not occur at an LDC instruction that updates the SR register and the following instruction. This occurrence will be identified as multiple exceptions, and may initiate reset processing.

Section 5 Cache

5.1 Overview

5.1.1 Features

The cache specifications are listed in table 5.1.

Table 5.1 Cache Specifications

Parameter	Specification
Capacity	16 kbytes
Structure	Instruction/data mixed, 4-way set associative
Locking	Way 2 and way 3 are lockable
Line size	16 bytes
Number of entries	256 entries/way
Write system	P0, P1, P3, U0: Write-back/write-through selectable
Replacement method	Least-recently-used (LRU) algorithm

5.1.2 Cache Structure

The cache mixes data and instructions and uses a 4-way set associative system. It is composed of four ways (banks), each of which is divided into an address section and a data section. Each of the address and data sections is divided into 256 entries. The data section of the entry is called a line. Each line consists of 16 bytes (4 bytes × 4). The data capacity per way is 4 kbytes (16 bytes × 256 entries), with a total of 16 kbytes in the cache as a whole (4 ways). Figure 5.1 shows the cache structure.

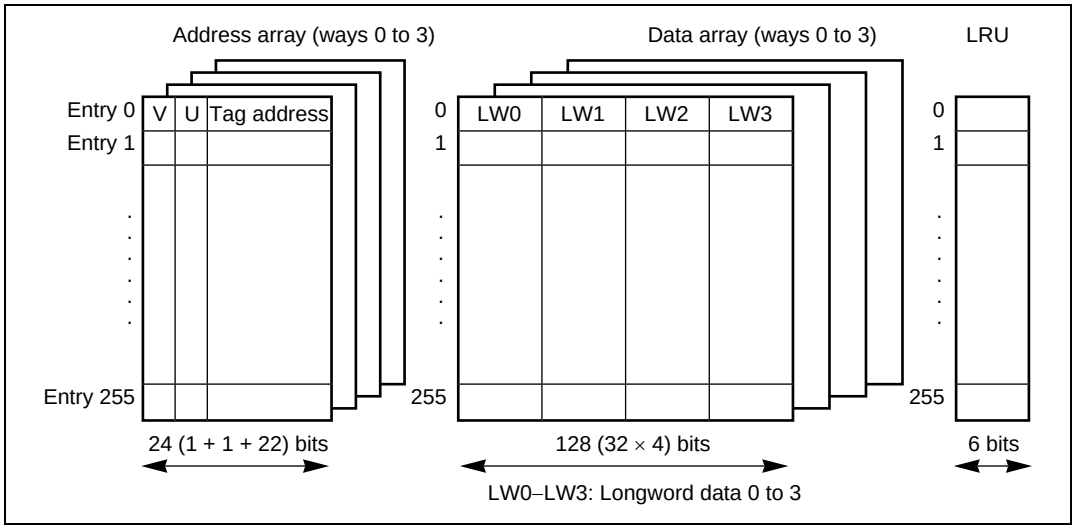


Figure 5.1 Cache Structure

Address Array: The V bit indicates whether the entry data is valid. When the V bit is 1, data is valid; when 0, data is not valid. The U bit indicates whether the entry has been written to in write-back mode. When the U bit is 1, the entry has been written to; when 0, it has not. The address tag holds the physical address used in the external memory access. It is composed of 22 bits (address bits 31 to 10) used for comparison during cache searches.

In the SH7727, the top three of 32 physical address bits are used as shadow bits (see section 12, Bus State Controller (BSC)), and therefore in a normal replace operation the top three bits of the tag address are cleared to 0.

The V and U bits are initialized to 0 by a power-on reset, but are not initialized by a manual reset. The tag address is not initialized by either a power-on or manual reset.

Data Array: Holds a 16-byte instruction or data. Entries are registered in the cache in line units (16 bytes). The data array is not initialized by a power-on or manual reset.

LRU: With the 4-way set associative system, up to four instructions or data with the same entry address (address bits 11 to 4) can be registered in the cache. When an entry is registered, the LRU shows which of the four ways it is recorded in. There are six LRU bits, controlled by hardware. A least-recently-used (LRU) algorithm is used to select the way.

In normal operation, four ways are used as cache and six LRU bits indicate the way to be replaced (table 5.2). If a bit pattern other than those listed in table 5.2 is set in the LRU bits by software, the

cache will not function correctly. When modifying the LRU bits by software, set one of the patterns listed in table 5.2.

The LRU bits are initialized to 0 by a power-on reset, but are not initialized by a manual reset.

Table 5.2 LRU and Way Replacement

LRU (5–0)	Way to be Replaced
000000, 000100, 010100, 100000, 110000, 110100	3
000001, 000011, 001011, 100001, 101001, 101011	2
000110, 000111, 001111, 010110, 011110, 011111	1
111000, 111001, 111011, 111100, 111110, 111111	0

5.1.3 Register Configuration

Table 5.3 shows details of the cache control register.

Table 5.3 Register Configuration

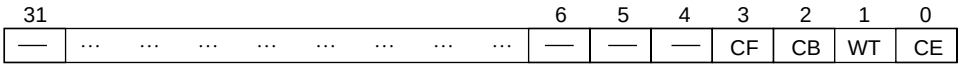
Register	Abbr.	R/W	Size	Initial Value	Address
Cache control register	CCR	R/W	Longword	H'00000000	H'FFFFFFEC
Cache control register 2	CCR2	W	Longword	H’00000000	H'040000B0 (H'A40000B0)*

Note: * When address translation by the MMU does not apply, the address in parentheses should be used.

5.2 Register Description

5.2.1 Cache Control Register (CCR)

The cache is enabled or disabled using the CE bit of the cache control register (CCR). CCR also has a CF bit (which invalidates all cache entries), and a WT and CB bits (which select either write-through mode or write-back mode). Programs that change the contents of the CCR register should be placed in address space that is not cached. Figure 5.2 shows the configuration of the CCR register.



—: Reserved bits. These bits are always read as 0. The write value should always be 0.

CF: Cache flush bit. Writing 1 flushes all cache entries (clears the V, U, and LRU bits of all cache entries to 0). Always reads 0. Write-back to external memory is not performed when the cache is flushed.

WT: Write-through bit. Indicates the cache's operating mode for areas P0, U0 and P3.
1 = write-through mode, 0 = write-back mode.

CE: Cache enable bit. Indicates whether the cache function is used.
1 = cache used, 0 = cache not used.

CB: Cache write-back bit. Indicates the cache's operating mode for area P1.
1 = write-back mode, 0 = write-through mode.

Figure 5.2 CCR Register Configuration

5.2.2 Cache Control Register 2 (CCR2)

CCR2 register is used to enable or disable cache locking mechanism during DSP mode (CPU status register bit 12) only. Executing a prefetch instruction (PREF) during DSP mode will bring in one line size of data pointed by Rn to cache, according to the setting of CCR2 [9:8] (W3LOAD, W3LOCK) and [1:0] (W2LOAD, W2LOCK):

When CCR2[9:8]=11, during DSP mode PREF @Rn will bring the data into way 3. When CCR2[9:8]=00, 01 or 10 during DSP mode, or any setting during non-DSP mode, PREF @Rn will place the data into the way pointed by LRU.

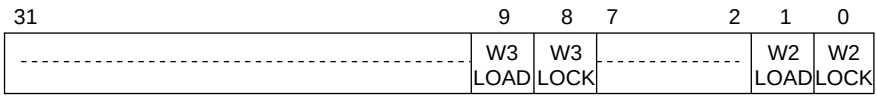
When CCR2[1:0]=11, during DSP mode PREF @Rn will bring the data into way 2. When CCR2[1:0]=00, 01 or 10 during DSP mode, or any setting during non-DSP mode, PREF @Rn will place the data into the way pointed by LRU.

CCR2 must be set before cache is enabled.

When a PREF instruction is issued and there is a cache hit, the operation is treated as NOP.

Figure 5.3 shows the configuration of the CCR2 register.

The CCR2 register is a write-only register. If read, an undefined value will be returned.



W2LOCK: Way 2 lock bit. W2LOAD: Way 2 load bit.

When W2LOCK = 1 & W2LOAD = 1 & DSP = 1, the prefetched data will always be loaded into Way2. In all other conditions the prefetched data will be loaded into the way pointed by LRU.

W3LOCK: Way 3 lock bit. W3LOAD: Way 3 load bit.

When W3LOCK = 1 & W3LOAD = 1 & DSP = 1, the prefetched data will always be loaded into Way3. In all other conditions the prefetched data will be loaded into the way pointed by LRU.

Note: W2LOAD and W3LOAD should not be set to high at the same time.

Figure 5.3 CCR2 Register Configuration

Whenever CCR2 bit 8 (W3LOCK) or bit 0 (W2LOCK) is high the cache is locked. The locked data will not be overwritten unless W3LOCK bit and W2LOCK bit are reset or the PREF condition during DSP mode matched. During cache locking mode, the LRU in table 5.2 will be replaced by tables 5.4 to 5.6.

Table 5.4 LRU and Way Replacement (when W2LOCK=1)

LRU (5–0)	Way to be Replaced
000000, 000001, 000100, 010100, 100000, 100001, 110000, 110100	3
000011, 000110, 000111, 001011, 001111, 010110, 011110, 011111	1
101001, 101011, 111000, 111001, 111011, 111100, 111110, 111111	0

Table 5.5 LRU and Way Replacement (when W3LOCK=1)

LRU (5–0)	Way to be Replaced
000000, 000001, 000011, 001011, 100000, 100001, 101001, 101011	2
000100, 000110, 000111, 001111, 010100, 010110, 011110, 011111	1
110000, 110100, 111000, 111001, 111011, 111100, 111110, 111111	0

Table 5.6 LRU and Way Replacement (when W2LOCK=1 and W3LOCK=1)

LRU (5–0)	Way to be Replaced
000000, 000001, 000011, 000100, 000110, 000111, 001011, 001111, 010100, 010110, 011110, 011111	1
100000, 100001, 101001, 101011, 110000, 110100, 111000, 111001, 111011, 111100, 111110, 111111	0

5.3 Cache Operation

5.3.1 Searching the Cache

If the cache is enabled, whenever instructions or data in memory are accessed the cache will be searched to see if the desired instruction or data is in the cache. Figure 5.4 illustrates the method by which the cache is searched. The cache is a physical cache and holds physical addresses in its address section.

Entries are selected using bits 11 to 4 of the address (virtual) of the access to memory and the address tag of that entry is read. In parallel to reading of the address tag, the logical address is translated to a physical address in the MMU. The physical address after translation and the physical address read from the address section are compared. The address comparison uses all four ways. When the comparison shows a match and the selected entry is valid (V = 1), a cache hit occurs. When the comparison does not show a match or the selected entry is not valid (V = 0), a cache miss occurs. Figure 5.4 shows a hit on way 1.

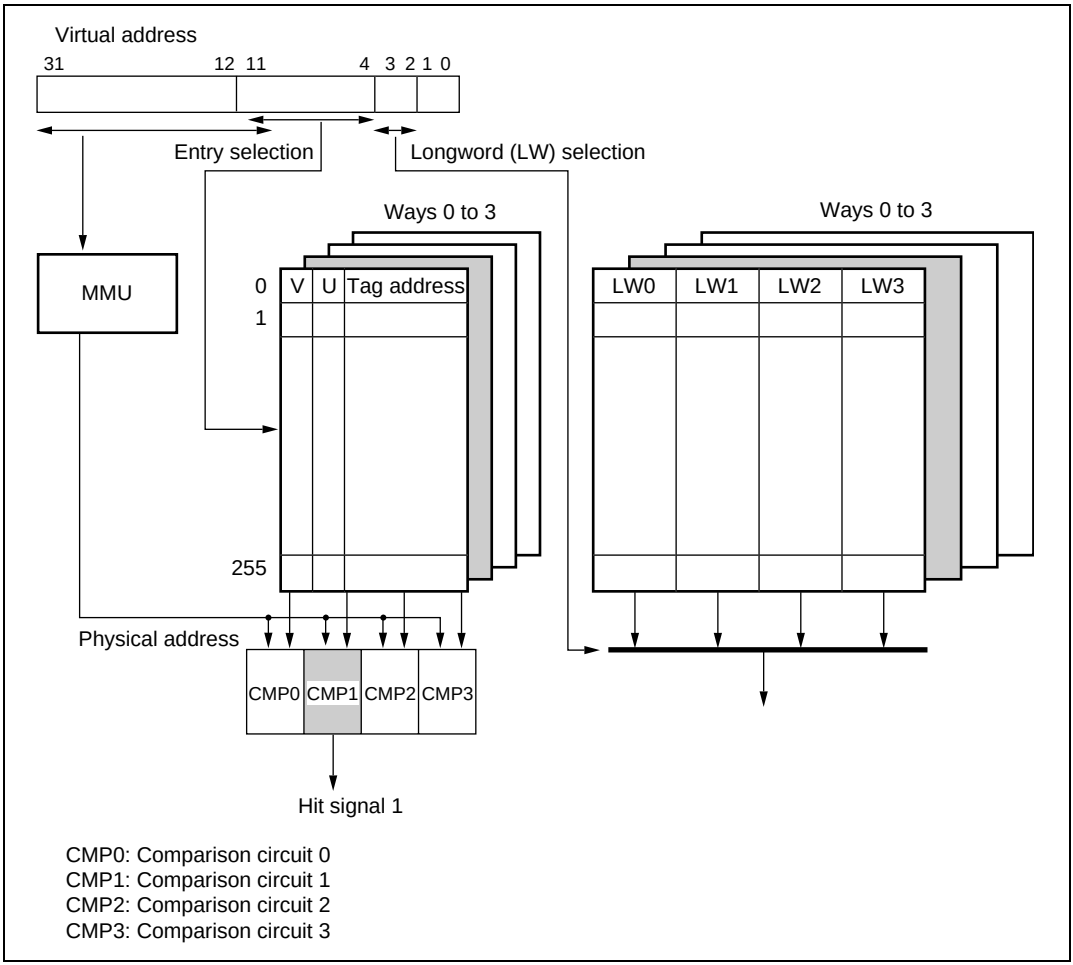


Figure 5.4 Cache Search Scheme

5.3.2 Read Access

Read Hit: In a read access, instructions and data are transferred from the cache to the CPU. The LRU is updated.

Read Miss: An external bus cycle starts and the entry is updated. The way replaced is the one least recently used. Entries are updated in 16-byte units. When the desired instruction or data that caused the miss is loaded from external memory to the cache, the instruction or data is transferred to the CPU in parallel with being loaded to the cache. When it is loaded in the cache, the U bit is cleared to 0 and the V bit is set to 1. In the write-back mode, when the U bit of the entry to be replaced is 1, the cache update cycle starts after the entry is transferred to the write-back buffer. After the cache completes its update cycle, the write-back buffer writes back the entry to the memory. The write-back unit is 16 bytes.

5.3.3 Prefetch Operations

Prefetch Hit: The LRU is updated so that the hit way becomes the most recent. Other cache contents are not updated. Instruction or data transfer to the CPU is not performed.

Prefetch Miss: Instruction or data transfer to the CPU is not performed, and the way replaced is as shown in table 5.2, table 5.4, table 5.5, and table 5.6. Other operations are the same as in the case of a read miss.

5.3.4 Write Access

Write Hit: In a write access in the write-back mode, the data is written to the cache and the U bit of the entry written is set to 1. Writing occurs only to the cache; no external memory write cycle is issued. In the write-through mode, the data is written to the cache and an external memory write cycle is issued.

Write Miss: In the write-back mode, an external write cycle starts when a write miss occurs, and the entry is updated. The way to be replaced is the one least recently used. When the U bit of the entry to be replaced is 1, the cache update cycle starts after the entry is transferred to the write-back buffer. The write-back unit is 16 bytes. Data is written to the cache and the U bit is set to 1 and the V bit is also set to 1. After the cache completes its update cycle, the write-back buffer writes back the entry to the memory. In the write-through mode, no write to cache occurs in a write miss; the write is only to the external memory.

5.3.5 Write-Back Buffer

When the U bit of the entry to be replaced in the write-back mode is 1, it must be written back to the external memory. To increase performance, the entry to be replaced is first transferred to the write-back buffer and fetching of new entries to the cache takes priority over writing back to the external memory. During the write back cycles, the cache can be accessed. The write-back buffer can hold one line of the cache data (16 bytes) and its physical address. Figure 5.5 shows the configuration of the write-back buffer.

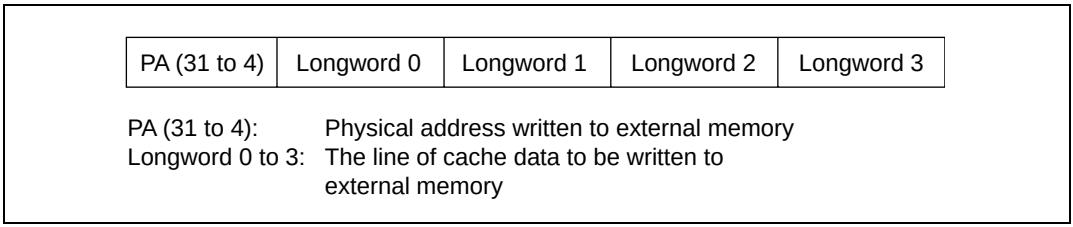


Figure 5.5 Write-Back Buffer Configuration

5.3.6 Coherency of Cache and External Memory

Use software to ensure coherency between the cache and the external memory. When memory shared by this LSI and another device is accessed, the latest data may be in a write-back mode cache, so invalidate the entry that includes the latest data in the cache, generate a write back, and update the data in memory before using it. When the caching area is updated by a device other than the SH7727, invalidate the entry that includes the updated data in the cache.

5.4 Memory-Mapped Cache

To allow software management of the cache, cache contents can be read and written by means of MOV instructions in the privileged mode. The cache is mapped onto the P4 area in logical address space. The address array is mapped onto addresses H'F0000000 to H'F0FFFFFF, and the data array onto addresses H'F1000000 to H'F1FFFFFF. Only longword can be used as the access size for the address array and data array, and instruction fetches cannot be performed.

5.4.1 Address Array

The address array is mapped onto H'F0000000 to H'F0FFFFFF. To access an address array, the 32-bit address field (for read/write accesses) and 32-bit data field (for write accesses) must be specified. The address field specifies information for selecting the entry to be accessed; the data

field specifies the address, V bit, U bit, and LRU bits to be written to the address array ((1) in figure 5.6).

In the address field, specify the entry address selecting the entry (bits 11 to 4), W for selecting the way (bits 12 and 11: in normal mode (8-kbyte cache), 00 is way 0, 01 is way 1, 10 is way 2, and 11 is way 3), and H'F0 to indicate address array access (bits 31 to 24).

When writing, specify bit 3 as the A bit. The A bit indicates whether addresses are compared during writing. When the A bit is 1, the addresses of four entries selected by the entry addresses are compared to the addresses to be written into the address array specified in the data field. Writing takes place to the way that has a hit. When a miss occurs, nothing is written to the address array and no operation occurs. The way number (W) specified in bits 12 and 11 is not used. When the A bit is 0, it is written to the entry selected with the entry address and way number without comparing addresses. The address specified by bits 31 to 10 in the data specification in figure 5.6 (1), address array access, is a logical address. When the MMU is enabled, the address is translated into a physical address, then the physical address is used in comparing addresses when the A bit is 1. The physical address is written into the address array.

When reading, the address tag, V bit, U bit, and LRU bits of the entry specified by the entry address and way number (W) are read using the data format shown in figure 5.6 without comparing addresses. To invalidate a specific entry, specify the entry by its entry address and way number, and write 0 to its V bit. To invalidate only an entry for an address to be invalidated, specify 1 for the A bit.

When an entry for which 0 is written to the V bit has a U bit set to 1, it will be written back. This allows coherency to be achieved between the external memory and cache by invalidating the entry. However, when 0 is written to the V bit, 0 must also be written to the U bit of that entry.

In the SH7727, the upper 3 bits of the 32-bit physical address are treated as a shadow field (see section 12, Bus State Controller (BSC)). Therefore, when a cache miss occurs, 0 is stored in the upper 3 bits of the address array address tag.

When using an MOV instruction to modify the address array directly, a nonzero value must not be written to the upper 3 bits of the address tag.

5.4.2 Data Array

The data array is mapped onto H'F1000000 to H'F1FFFFFF. To access a data array, the 32-bit address field (for read/write accesses) and 32-bit data field (for write accesses) must be specified. The address field specifies information for selecting the entry to be accessed; the data field specifies the longword data to be written to the data array ((2) in figure 5.6).

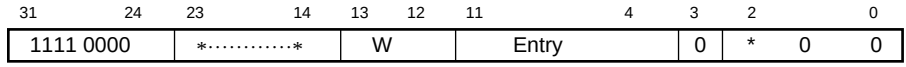
Specify the entry address for selecting the entry (bits 11 to 4), L indicating the longword position within the (16-byte) line (bits 3 and 2: 00 is longword 0, 01 is longword 1, 10 is longword 2, and 11 is longword 3), W for selecting the way (bits 12 and 11: in normal mode, 00 is way 0, 01 is way 1, 10 is way 2, and 11 is way 3), and H'F1 to indicate data array access (bits 31 to 24).

Both reading and writing use the longword of the data array specified by the entry address, way number and longword address. The access size of the data array is fixed at longword.

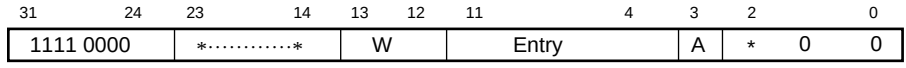
1. Address array access

Address specification

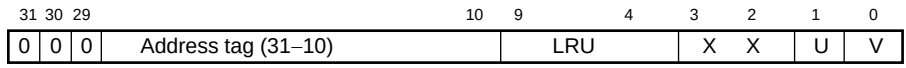
Read access



Write access

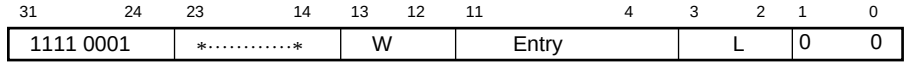


Data specification

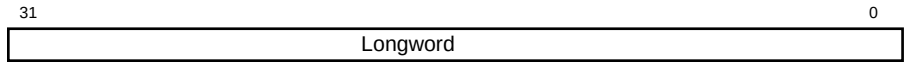


2. Data array access (both read and write accesses)

Address specification



Data specification



X: 0 for read, don't care for write
*: Don't care bit

Figure 5.6 Specifying Address and Data for Memory-Mapped Cache Access

5.5 Usage Examples

5.5.1 Invalidating Specific Entries

Specific cache entries can be invalidated by writing 0 to the entry's V bit. When the A bit is 1, the address tag specified by the write data is compared to the address tag within the cache selected by the entry address, and data is written when a match is found. If no match is found, there is no operation. R0 specifies the write data in R0 and R1 specifies the address. When the V bit of an entry in the address array is set to 0, the entry is written back if the entry's U bit is 1.

```
; R0=H'01100010; VPN=B'0000 0001 0001 0000 0000 00, U=0, V=0
; R1=H'F0000088; address array access, entry=B'00001000, A=1
;
MOV.L R0,@R1
```

5.5.2 Reading the Data of a Specific Entry

This example reads the data section of a specific cache entry. The longword indicated in the data field of the data array in figure 5.6 is read to the register. R0 specifies the address and R1 is read.

```
; R1=H'F100 004C; data array access, entry=B'00000100, Way = 0,
; longword address = 3
;
MOV.L @R0,R1 ; Longword 3 is read.
```

Section 6 X/Y Memory

6.1 Overview

The SH7727 has on-chip X-RAM and Y-RAM. It can be used by CPU, DSP and DMAC to store instructions or data.

6.1.1 Features

The X/Y Memory features are listed in table 6.1.

Table 6.1 X/Y Memory Specifications

Parameter	Features
Addressing method	User selectable mapping mechanism • Fixed mapping for mission-critical realtime applications (P2/Uxy area) • Automatic mapping through TLB for easy to use (P0/P3/U0 area)
Ports	3 independent read/write ports • 8-/16-/32-bit access from the CPU • Maximum of two simultaneous 16-bit accesses, or 16/32-bit accesses, from the DSP • 8-/16-/32-bit access from the DMAC
Size	8-kbyte RAM for X and Y memory each

6.2 X/Y Memory Access from the CPU

The X/Y memory can be located in either map-enabled area or fixed-mapped area, depending on the mode bit (MD) and DSP bit (DSP) setting in the status register (SR). Figure 6.1 shows X/Y memory logical mapping.

1. Privileged Mode

   MD = 1, DSP = 0; Any physical address in space P0 or P3 can map to X/Y memory through TLB translation. Addresses ranging from H'A500 0000 to H'A5FF FFFF in the P2 space can also fixed map to X/Y memory. Since the DSP extension is disabled, the DSP instruction set and registers are not available to the programmer.

2. User Mode

   MD = 0, DSP = 0; Any physical address in the U0 space can access X/Y memory through TLB translation. Any access to addresses beyond the U0 space will cause an address error. Since the DSP extension is disabled, the DSP instruction set and registers are not available to the programmer.

3. Privileged-DSP Mode

   MD = 1, DSP = 1; Any physical address in space P0 or P3 can map to X/Y memory through TLB translation. Addresses ranging from H'A500 0000 to H'A5FF FFFF in the P2 space can also fixed-map to X/Y memory. Since the DSP extension is enabled, the DSP instruction set and registers are available to the programmer.

4. User-DSP Mode

   MD = 0, DSP = 1; Any physical address in space U0 can map to X/Y memory through TLB translation. Addresses ranging from H'A500 0000 to H'A5FF FFFF in the Uxy spaces can also fixed map to X/Y memory. Any access to outside of U0 and Uxy space will cause an address error. Since the DSP extension is enabled, the DSP instruction set and registers are available to the programmer.

It is recommended that for the mappable area, the C (cacheable) bit in the TLB entry must be set to 0 to guarantee a two-cycle access.

Mapping through TLB translation provides a flexible X/Y memory addressing scheme but takes two cycles even when the C bit in the TLB entry is cleared to 0. Fixed mapping provides a one-cycle access for read and two-cycle access for write, which is the appropriate method for mission-critical realtime operations.

The X/Y memory resides on the second 16 MB of physical address space area 1, from H'A500 0000 to H'A5FF FFFF. These 16-MB address spaces are shadowed and maps to the same 128-kbyte X/Y ROM/RAM. Figures 6.1 and 6.2 show X/Y memory physical mapping.

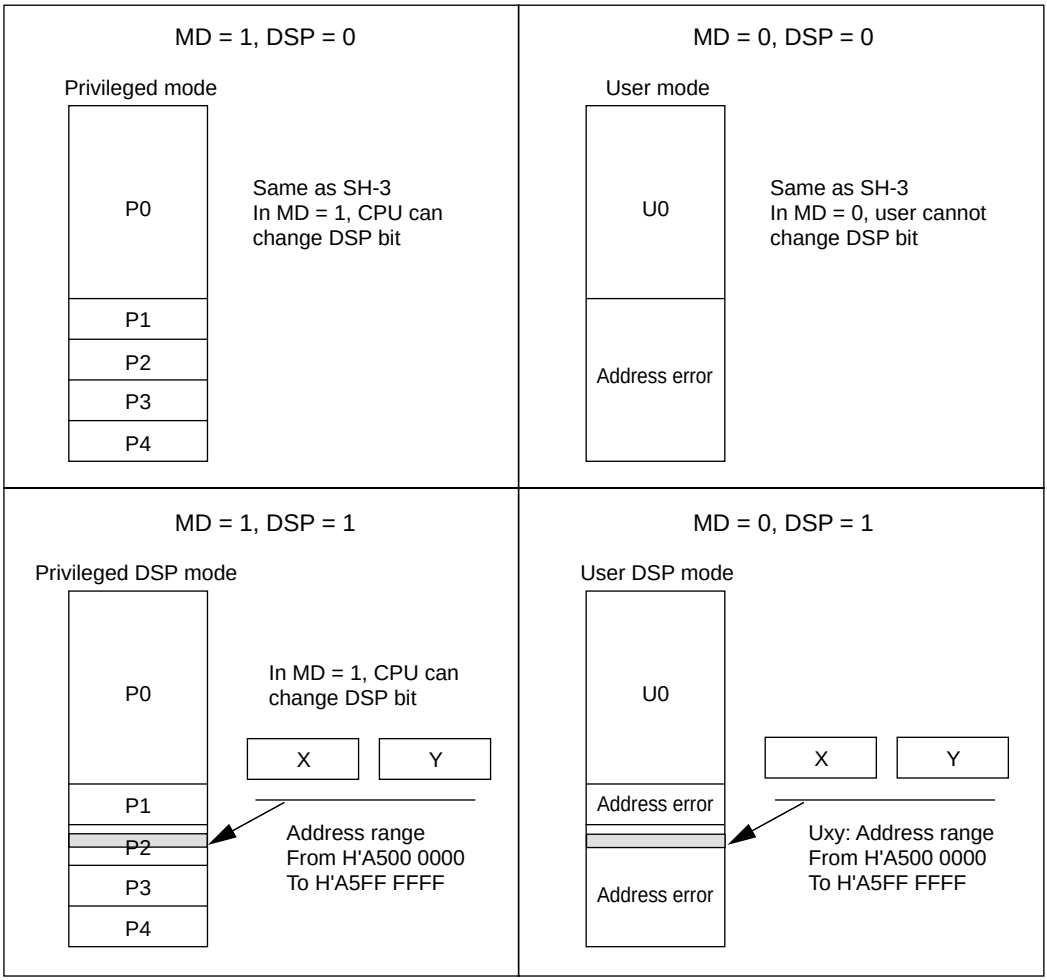


Figure 6.1 X/Y Memory Logical Address Mapping

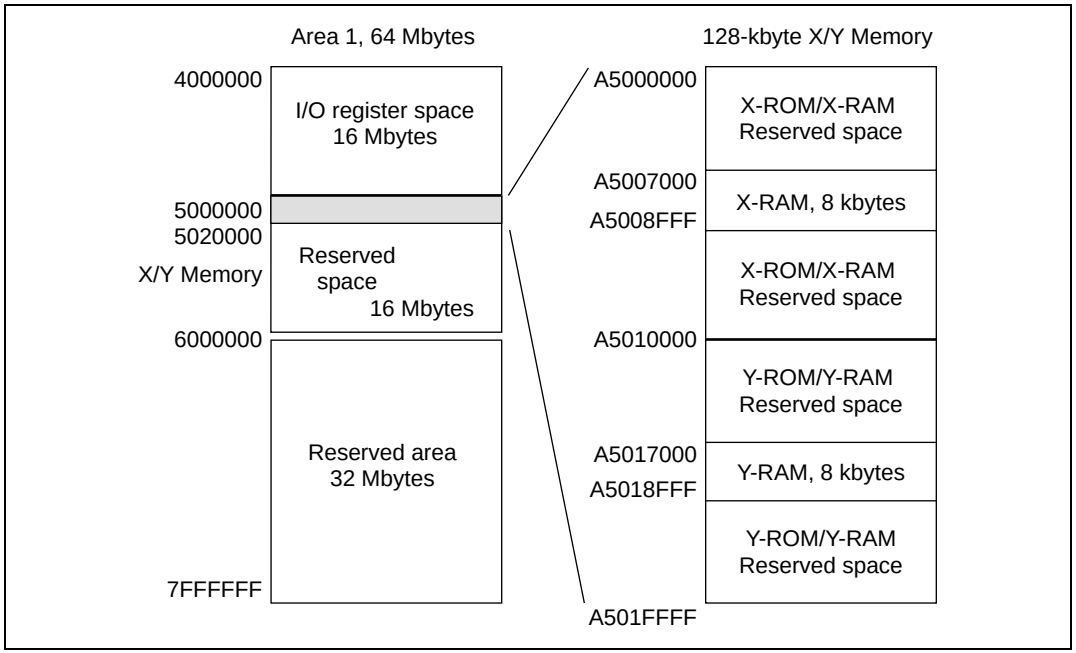


Figure 6.2 X/Y Memory Physical Address Mapping

6.3 X/Y Memory Access from the DSP

The X/Y memory can be accessed by the DSP through the X bus and Y bus. Accesses via the X bus/Y bus are always 16-bit, while accesses via the L bus are either 16-bit or 32-bit. Accesses via the X bus and Y bus cannot be specified simultaneously.

6.4 X/Y Memory Access from the DMAC

The X/Y memory also exists on the I bus and can be accessed by the DMAC. The DMAC access is 8-/16-/32-bit unit. If the I bus accesses X/Y memory simultaneously with an access from X bus/Y bus or L bus, the I bus master has a higher priority.

To access the X/Y memory by the DMAC, the physical address from H'05000000 to H'0501FFFF should be used.

Section 7 Interrupt Controller (INTC)

7.1 Overview

The interrupt controller (INTC) ascertains the priority of interrupt sources and controls interrupt requests to the CPU. The INTC registers set the order of priority of each interrupt, allowing the user to process interrupt requests according to the user-set priority.

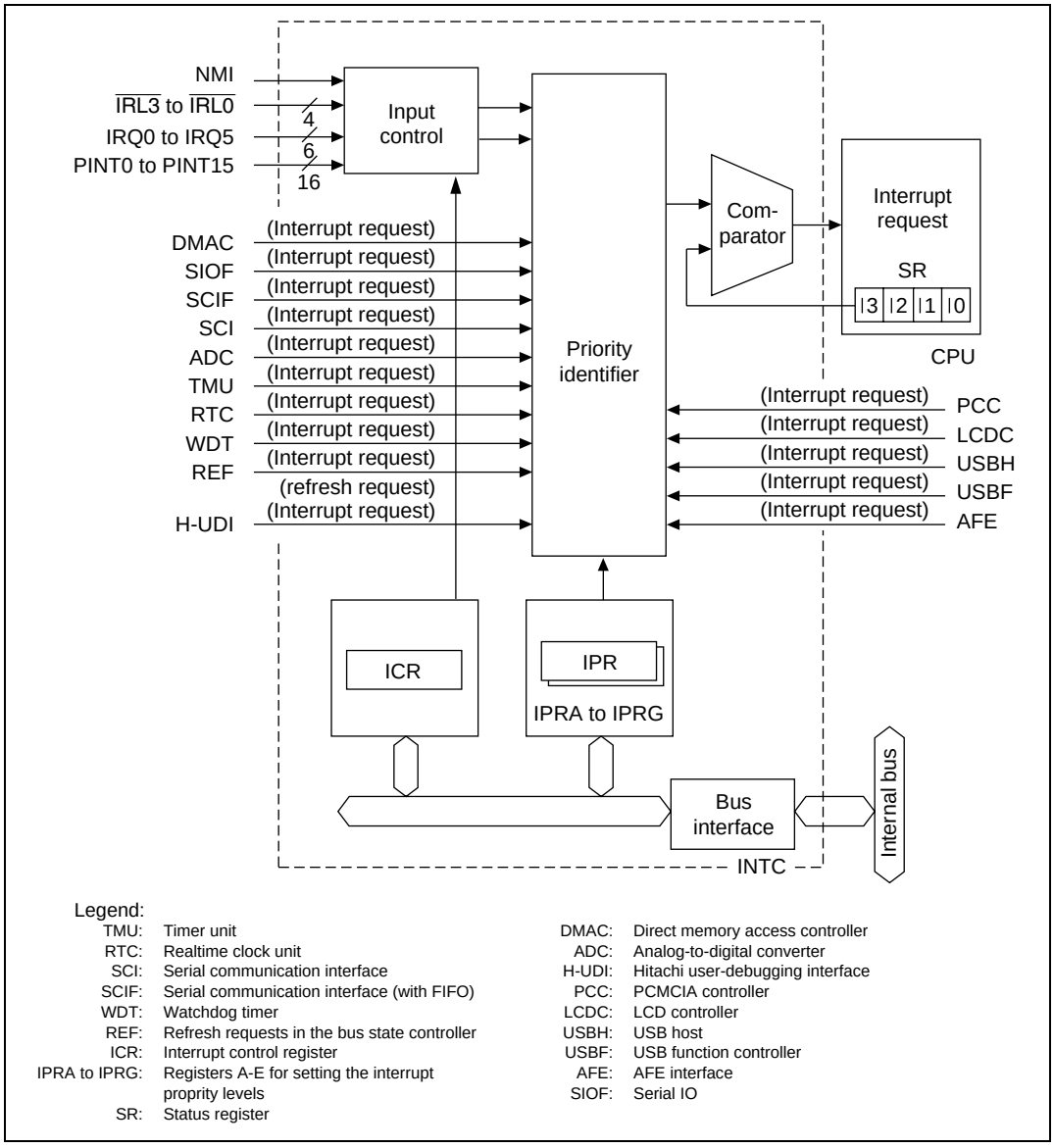
7.1.1 Features

INTC has the following features:

- 16 levels of interrupt priority can be set: By setting the five interrupt-priority registers, the priorities of on-chip supporting module, IRQ, and PINT interrupts can be selected from 16 levels for individual request sources.
- NMI noise canceller function: NMI input-level bit indicates NMI pin states. By reading this bit in the interrupt exception service routine, the pin state can be checked, enabling it to be used as a noise canceller.

7.1.2 Block Diagram

Figure 7.1 is a block diagram of the INTC.



Legend:

TMU:	Timer unit	DMAC:	Direct memory access controller
RTC:	Realtime clock unit	ADC:	Analog-to-digital converter
SCI:	Serial communication interface	H-UDI:	Hitachi user-debugging interface
SCIF:	Serial communication interface (with FIFO)	PCC:	PCMCIA controller
WDT:	Watchdog timer	LCDC:	LCD controller
REF:	Refresh requests in the bus state controller	USBH:	USB host
ICR:	Interrupt control register	USBF:	USB function controller
IPRA to IPRG:	Registers A-E for setting the interrupt proprity levels	AFE:	AFE interface
SR:	Status register	SIOF:	Serial IO

Figure 7.1 INTC Block Diagram

7.1.3 Pin Configuration

Table 7.1 lists the INTC pin configuration.

Table 7.1 Pin Configuration

Name	Abbreviation	I/O	Description
Nonmaskable interrupt input pin	NMI	I	Input of interrupt request signal, which is nonmaskable by SR.IMASK
Interrupt input pins	IRQ5 to IRQ0 IRL3 to IRL0	I	Input of interrupt request signals, which is maskable by SR.IMASK
Port interrupt input pins	PINT0 to PINT15	I	Port input of interrupt request signals, which is maskable by SR.IMASK

7.1.4 Register Configuration

The INTC has 17 registers listed in table 7.2.

Table 7.2 Register Configuration

Name	Abbr.	R/W	Initial Value*1	Address	Access Size
Interrupt control register 0	ICR0	R/W	*2	H'FFFFFEE0	16
Interrupt control register 1	ICR1	R/W	H'0000	H'04000010 (H'A4000010)*3	16
Interrupt control register 2	ICR2	R/W	H'0000	H'04000012 (H'A4000012)*3	16
Interrupt control register 3	ICR3	R/W	H'0000	H'04000228 (H'A4000228)*3	16
PINT interrupt enable register	PINTER	R/W	H'0000	H'04000014 (H'A4000014)*3	16
Interrupt priority level setting register A	IPRA	R/W	H'0000	H'FFFFFEE2	16
Interrupt priority level setting register B	IPRB	R/W	H'0000	H'FFFFFEE4	16
Interrupt priority level setting register C	IPRC	R/W	H'0000	H'04000016 (H'A4000016)*3	16
Interrupt priority level setting register D	IPRD	R/W	H'0000	H'04000018 (H'A4000018)*3	16
Interrupt priority level setting register E	IPRE	R/W	H'0000	H'0400001A (H'A400001A)*3	16
Interrupt priority level setting register F	IPRF	R/W	H'0000	H'04000220 (H'A4000220)*3	16
Interrupt priority level setting register G	IPRG	R/W	H'0000	H'04000222 (H'A4000222)*3	16
Interrupt request register 0	IRR0	R/W	H'00	H'04000004 (H'A4000004)*3	8
Interrupt request register 1	IRR1	R	H'00	H'04000006 (H'A4000006)*3	8
Interrupt request register 2	IRR2	R	H'00	H'04000008 (H'A4000008)*3	8
Interrupt request register 3	IRR3	R	H'00	H'04000224 (H'A4000224)*3	16
Interrupt request register 4	IRR4	R	H'00	H'04000226 (H'A4000226)*3	16

Notes: 1. Initialized by a power-on or manual reset.
2. H'8000 when the NMI pin is at high level. H'0000 when the NMI pin is at low level.
3. When address translation by the MMU does not apply, the address in parentheses should be used.

7.2 Interrupt Sources

There are five types of interrupt sources: NMI, IRQ, IRL, PINT, and on-chip supporting modules. Each interrupt has priority levels (0 to 16) with 0 the lowest and 16 the highest. Priority level 0 masks an interrupt.

7.2.1 NMI Interrupts

The NMI interrupt has the highest priority level of 16. When the BLMSK bit of the interrupt control register (ICR1) is 1 or the BL bit of the status register (SR) is 0, NMI interrupts are accepted when the MAI bit of the ICR1 register is 0. NMI interrupts are edge-detected. In sleep or standby mode, the interrupt is accepted regardless of the BL. The NMI edge select bit (NMIE) in the interrupt control register 0 (ICR0) is used to select either the rising or falling edge. When the NMIE bit of the ICR0 register is changed, the NMI interrupt is not detected for 20 cycles after changing the ICR0.NMIE to avoid a false detection of the NMI interrupt. NMI interrupt exception handling does not affect the interrupt mask level bits (I3 to I0) in the status register (SR).

When the BLMSK bit of the ICR1 register is set to 1 and only NMI interrupts are accepted, the SPC register and SSR register are updated by the NMI interrupt handler, making it impossible to return to the original processing from exception handling initiated prior to the NMI. Use should therefore be restricted to cases where return is not necessary.

It is possible to wake the chip up from the standby state with an NMI interrupt (except when the MAI bit of the ICR1 register is set to 1).

7.2.2 IRQ Interrupt

IRQ interrupts are input by priority from pins IRQ0 to IRQ5 with a level or an edge. The priority level can be set by priority setting registers C, D (IPRC, IPRD) in a range from levels 0 to 15.

When using edge sensing for IRQ interrupts, do the following to clear IR0.

To clear bits IRQ5R to IRQ0R to 0, read from IRR0 before writing. After confirming that the bits to be cleared to 0 are set to 1, write 0 to them. In this case write 0 only to the bits to be cleared; write 1 to the other bits. The values of the bits to which 1 is written do not change.

When level sensing is used for IRQ interrupts, bits IRQ5R to IRQ0R indicate whether or not an interrupt request has been input. They can be set and cleared by the values input to pins IRQ5R to IRQ0R alone.

When the ICR1 register is rewritten, IRQ interrupts may be mistakenly detected, depending on the pin states. To prevent this, rewrite the register while interrupts are masked, then release the mask after clearing the illegal interrupt by writing 0 to interrupt request register 0 (IRR0).

It is necessary for an edge input interrupt detection to input a pulse width more than two-cycle width by P clock basis.

With level detection, the level must be maintained until the interrupt is accepted and the CPU starts interrupt handling.

The interrupt mask bits (I3 to I0) of the status register (SR) are not affected by IRQ interrupt processing.

Interrupts IRQ5 to IRQ0 can be used to wake the chip up from the software standby mode (but only when the RTC 32 kHz oscillator is used).

In this case, the priority level of the interrupt to be used must be higher than the level of bits I3 to I0 in the SR register.

Notes: The following cautions apply when IRQ edge detection is used:

1. If an IRQ edge is input immediately before the CPU enters the standby mode (between when the CPU executes the SLEEP instruction and when STATUS0 goes high), the interrupt may not be detected properly. After this, if the IRQ edge is input again after STATUS0 goes high, the interrupt will be detected.
2. If an IRQ edge is input while the frequency is changing due to a change in the value of the STC bit in the FRQCR register (during the count by WDT), the interrupt may not be detected properly. If the IRQ edge is input again after the WDT count completes, the interrupt will be detected.

7.2.3 IRL Interrupts

IRL interrupts are input by level at pins $\overline{\text{IRL3}}$ to $\overline{\text{IRL0}}$. The priority level is the higher level indicated by pins $\overline{\text{IRL3}}$ to $\overline{\text{IRL0}}$. An $\overline{\text{IRL3}}$ to $\overline{\text{IRL0}}$ value of 0 (0000) indicates the highest-level interrupt request (interrupt priority level 15). A value of 15 (1111) indicates no interrupt request (interrupt priority level 0). Figure 7.2 shows an example of an IRL interrupt connection. Table 7.3 shows $\overline{\text{IRL}}$ pins and interrupt levels.

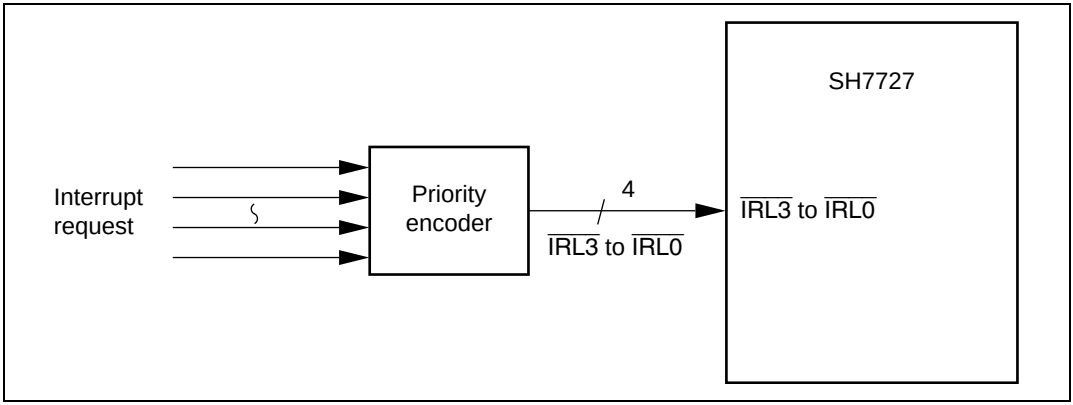


Figure 7.2 Example of IRL Interrupt Connection

Table 7.3 $\overline{\text{IRL3}}$ to $\overline{\text{IRL0}}$ Pins and Interrupt Levels

$\overline{\text{IRL3}}$	$\overline{\text{IRL2}}$	$\overline{\text{IRL1}}$	$\overline{\text{IRL0}}$	Interrupt Priority Level	Interrupt Request
0	0	0	0	15	Level 15 interrupt request
0	0	0	1	14	Level 14 interrupt request
0	0	1	0	13	Level 13 interrupt request
0	0	1	1	12	Level 12 interrupt request
0	1	0	0	11	Level 11 interrupt request
0	1	0	1	10	Level 10 interrupt request
0	1	1	0	9	Level 9 interrupt request
0	1	1	1	8	Level 8 interrupt request
1	0	0	0	7	Level 7 interrupt request
1	0	0	1	6	Level 6 interrupt request
1	0	1	0	5	Level 5 interrupt request
1	0	1	1	4	Level 4 interrupt request
1	1	0	0	3	Level 3 interrupt request
1	1	0	1	2	Level 2 interrupt request
1	1	1	0	1	Level 1 interrupt request
1	1	1	1	0	No interrupt request

A noise-cancellation feature is built in, and the IRL interrupt is not detected unless the levels sampled at every supporting module cycle remain unchanged for two consecutive cycles, so that no transient level on the $\overline{\text{IRL}}$ pin change is detected. In the standby mode, as the peripheral clock is stopped, noise cancellation is performed using the 32-kHz clock for the RTC instead. Therefore

when the RTC is not used, interruption by means of IRL interrupts cannot be performed in standby mode.

The priority level of the IRL interrupt must not be lowered unless the interrupt is accepted and the interrupt processing starts. Correct operation cannot be guaranteed if the level is not maintained. However, the priority level can be changed to a higher one.

The interrupt mask bits (I3 to I0) in the status register (SR) are not affected by IRL interrupt processing.

7.2.4 PINT Interrupt

PINT interrupts are input by priority from pins PINT0 to PINT15 with a level. The priority level can be set by priority setting registers D (IPRD) in a range from levels 0 to 15, in the unit of PINT0 to PINT7 or PINT8 to PINT15.

The PINT interrupt level should be held until the interrupt is accepted and interrupt handling is started.

The interrupt mask bits (I3 to I0) of the status register (SR) are not affected by PINT interrupt processing.

PINT interrupts can wake the chip up from the standby state when the relevant interrupt level is higher than I3 to I0 in the SR register (but only when the RTC 32-kHz oscillator is used).

7.2.5 On-Chip Supporting Module Interrupts

On-chip supporting module interrupts are generated by the following fourteen modules:

- Timer unit (TMU)
- Realtime clock (RTC)
- Serial communication interface (SCI, SCIF)
- Bus state controller (BSC)
- Watchdog timer (WDT)
- Direct memory access controller (DMAC)
- Analog-to-digital converter (ADC)
- PC Card controller (PCC)
- OHCI compliant USB HOST controller (USBH)
- USB function controller (USBF)
- AFE interface (AFEIF)

- LCD controller (LCDC)
- Hitachi user-debugging interface (H-UDI)
- Serial IO (SIOF)

Not every interrupt source is assigned a different interrupt vector. Sources are reflected on the interrupt event register (INTEVT and INTEVT2). It is easy to identify sources by using the values of the INTEVT or INTEVT2 register as branch offsets.

The priority level (from 0 to 15) can be set for each module except for H-UDI by writing to the interrupt priority setting registers A to G (IPRA to IPRG). The priority level of H-UDI interrupt is 15 (fixed).

The interrupt mask bits (I3 to I0) of the status register are not affected by the on-chip supporting module interrupt processing.

TMU and RTC interrupts can wake the chip up from the standby state when the relevant interrupt level is higher than I3 to I0 in the SR register (but only when the RTC 32-kHz oscillator is used).

7.2.6 Interrupt Exception Handling and Priority

Tables 7.4 and 7.5 list the codes for the interrupt event register (INTEVT and INTEVT2), and the order of interrupt priority. Each interrupt source is assigned unique code. The start address of the interrupt service routine is common to each interrupt source. This is why, for instance, the value of INTEVT or INTEVT2 is used as offset at the start of the interrupt service routine and branched to identify the interrupt source.

The order of priority of the on-chip supporting module, IRQ, and PINT interrupts is set within the priority levels 0 to 15 at will by using the interrupt priority level set to registers A to G (IPRA to IPRG). The order of priority of the on-chip supporting module, IRQ, and PINT interrupts is set to zero by RESET.

When the order of priorities for multiple interrupt sources are set to the same level and such interrupts are generated at the same time, they are processed according to the default order listed in tables 7.4 and 7.5.

Table 7.4 Interrupt Exception Handling Sources and Priority (IRQ Mode)

Interrupt Source		INTEVT Code (INTEVT2 Code)	Interrupt Priority (Initial Value)	IPR (Bit Numbers)	Priority within IPR Setting Unit	Default Priority
NMI		H'1C0 (H'1C0)	16	—	—	High
H-UDI		H'5E0 (H'5E0)	15	—	—	
IRQ	IRQ0	H'200–3C0* (H'600)	0–15 (0)	IPRC (3–0)	—	
	IRQ1	H'200–3C0* (H'620)	0–15 (0)	IPRC (7–4)	—	
	IRQ2	H'200–3C0* (H'640)	0–15 (0)	IPRC (11–8)	—	
	IRQ3	H'200–3C0* (H'660)	0–15 (0)	IPRC (15–12)	—	
	IRQ4	H'200–3C0* (H'680)	0–15 (0)	IPRD (3–0)	—	
	IRQ5	H'200–3C0* (H'6A0)	0–15 (0)	IPRD (7–4)	—	
PINT	PINT0–7	H'200–3C0* (H'700)	0–15 (0)	IPRD (15–12)	—	
	PINT8–15	H'200–3C0* (H'720)	0–15 (0)	IPRD (11–8)	—	
DMAC	DEI0	H'200–3C0* (H'800)	0–15 (0)	IPRE (15–12)	High	
	DEI1	H'200–3C0* (H'820)				
	DEI2	H'200–3C0* (H'840)				
	DEI3	H'200–3C0* (H'860)			Low	
SCIF	ERI2	H'200–3C0* (H'900)	0–15 (0)	IPRE (7–4)	High	
	RXI2	H'200–3C0* (H'920)				
	BRI2	H'200–3C0* (H'940)				
	TXI2	H'200–3C0* (H'960)			Low	
ADC	ADI	H'200–3C0* (H'980)	0–15 (0)	IPRE (3–0)	—	
LCDC	LCDCI	H'200–3C0* (H'9A0)	0–15 (0)	IPRF(11–8)	—	
SIOF	SIFERI	H'200–3C0* (H'B00)	0–15 (0)	IPRF(3–0)	High	
	SIFTXI	H'200–3C0* (H'B20)	0–15 (0)			
	SIFRXI	H'200–3C0* (H'B40)	0–15 (0)			
	SIFCCI	H'200–3C0* (H'B60)	0–15 (0)		Low	
USBH	USBHI	H'200–3C0* (H'A00)	0–15 (0)	IPRG(15–12)	—	
USBF	USBFI0	H'200–3C0* (H'A20)	0–15 (0)	IPRG(11–8)	High	
	USBFI1	H'200–3C0* (H'A40)	0–15 (0)	IPRG(7–4)	Low	
AFEIF	AFEIFI	H'200–3C0* (H'A 60)	0–15 (0)	IPRG(3–0)	—	Low

Interrupt Source		INTEVT Code (INTEVT2 Code)	Interrupt Priority (Initial Value)	IPR (Bit Numbers)	Priority within IPR Setting Unit	Default Priority
PCC0	PC0SWIR	H'200–3C0* (H'9C0)	0–15 (0)	IPRF(7–4)	High	High
	PC0IRIR	H'200–3C0* (H'9C0)	0–15 (0)			
	PC0SCIR	H'200–3C0* (H'9C0)	0–15 (0)			
	PC0CDIR	H'200–3C0* (H'9C0)	0–15 (0)			
	PC0RCIR	H'200–3C0* (H'9C0)	0–15 (0)			
	PC0BWIR	H'200–3C0* (H'9C0)	0–15 (0)			
	PC0BDIR	H'200–3C0* (H'9C0)	0–15 (0)		Low	
TMU0	TUNI0	H'400 (H'400)	0–15 (0)	IPRA (15–12)	—	
TMU1	TUNI1	H'420 (H'420)	0–15 (0)	IPRA (11–8)	—	
TMU2	TUNI2	H'440 (H'440)	0–15 (0)	IPRA (7–4)	—	
RTC	ATI	H'480 (H'480)	0–15 (0)	IPRA (3–0)	High	
	PRI	H'4A0 (H'4A0)				
	CUI	H'4C0 (H'4C0)			Low	
SCI0	ERI	H'4E0 (H'4E0)	0–15 (0)	IPRB (7–4)	High	
	RXI	H'500 (H'500)				
	TXI	H'520 (H'520)				
	TEI	H'540 (H'540)			Low	
WDT	ITI	H'560 (H'560)	0–15 (0)	IPRB (15–12)	—	
REF	RCMI	H'580 (H'580)	0–15 (0)	IPRB (11–8)	High	
	ROVI	H'5A0 (H'5A0)			Low	Low

Note: * The code corresponding to an interrupt level shown in table 7.6 is set.

Table 7.5 Interrupt Exception Handling Sources and Priority (IRL Mode)

Interrupt Source		INTEVT Code (INTEVT2 Code)	Interrupt Priority (Initial Value)	IPR (Bit Numbers)	Priority within IPR Setting Unit	Default Priority
NMI		H'1C0 (H'1C0)	16	—	—	High
H-UDI		H'5E0 (H'5E0)	15	—	—	↕
IRL	$\overline{IRL(3:0)}$ = 0000	H'200 (H'200)	15	—	—	
	$\overline{IRL(3:0)}$ = 0001	H'220 (H'220)	14	—	—	
	$\overline{IRL(3:0)}$ = 0010	H'240 (H'240)	13	—	—	
	$\overline{IRL(3:0)}$ = 0011	H'260 (H'260)	12	—	—	
	$\overline{IRL(3:0)}$ = 0100	H'280 (H'280)	11	—	—	
	$\overline{IRL(3:0)}$ = 0101	H'2A0 (H'2A0)	10	—	—	
	$\overline{IRL(3:0)}$ = 0110	H'2C0 (H'2C0)	9	—	—	
	$\overline{IRL(3:0)}$ = 0111	H'2E0 (H'2E0)	8	—	—	
	$\overline{IRL(3:0)}$ = 1000	H'300 (H'300)	7	—	—	
	$\overline{IRL(3:0)}$ = 1001	H'320 (H'320)	6	—	—	
	$\overline{IRL(3:0)}$ = 1010	H'340 (H'340)	5	—	—	
	$\overline{IRL(3:0)}$ = 1011	H'360 (H'360)	4	—	—	
	$\overline{IRL(3:0)}$ = 1100	H'380 (H'380)	3	—	—	
	$\overline{IRL(3:0)}$ = 1101	H'3A0 (H'3A0)	2	—	—	
	$\overline{IRL(3:0)}$ = 1110	H'3C0 (H'3C0)	1	—	—	
IRQ	IRQ4	H'200–3C0* (H'680)	0–15 (0)	IPRD (3–0)	—	
	IRQ5	H'200–3C0* (H'6A0)	0–15 (0)	IPRD (7–4)	—	
PINT	PINT0–7	H'200–3C0* (H'700)	0–15 (0)	IPRD (15–12)	—	
	PINT8–15	H'200–3C0* (H'720)	0–15 (0)	IPRD (11–8)	—	
DMAC	DEI0	H'200–3C0* (H'800)	0–15 (0)	IPRE (15–12)	High	
	DEI1	H'200–3C0* (H'820)			↕	
	DEI2	H'200–3C0* (H'840)				
	DEI3	H'200–3C0* (H'860)			Low	
SCIF	ERI2	H'200–3C0* (H'900)	0–15 (0)	IPRE (7–4)	High	
	RXI2	H'200–3C0* (H'920)			↕	
	BRI2	H'200–3C0* (H'940)				
	TXI2	H'200–3C0* (H'960)			Low	Low

Interrupt Source		INTEVT Code (INTEVT2 Code)	Interrupt Priority (Initial Value)	IPR (Bit Numbers)	Priority within IPR Setting Unit	Default Priority
ADC	ADI	H'200–3C0* (H'980)	0–15 (0)	IPRE (3–0)	—	High
LCDC	LCDCI	H'200–3C0* (H'9A0)	0–15 (0)	IPRF (11–8)	—	
SIOF	SIFERI	H'200–3C0* (H'B00)	0–15 (0)	IPRF (3–0)	High	
	SIFTXI	H'200–3C0* (H'B20)	0–15 (0)			
	SIFRXI	H'200–3C0* (H'B40)	0–15 (0)			
	SIFCCI	H'200–3C0* (H'B60)	0–15 (0)		Low	
USBH	USBHI	H'200–3C0* (H'A00)	0–15 (0)	IPRG (15–12)	—	
USBF	USBFI0	H'200–3C0* (H'A20)	0–15 (0)	IPRG (11–8)	High	
	USBFI1	H'200–3C0* (H'A40)	0–15 (0)	IPRG (7–4)	Low	
AFEIF	AFEIFI	H'200–3C0* (H'A60)	0–15 (0)	IPRG (3–0)	—	
PCC0	PC0SWIR	H'200–3C0* (H'9C0)	0–15 (0)	IPRF (7–4)	High	
	PC0IRIR	H'200–3C0* (H'9C0)	0–15 (0)			
	PC0SCIR	H'200–3C0* (H'9C0)	0–15 (0)			
	PC0CDIR	H'200–3C0* (H'9C0)	0–15 (0)			
	PC0RCIR	H'200–3C0* (H'9C0)	0–15 (0)			
	PC0BWIR	H'200–3C0* (H'9C0)	0–15 (0)			
	PC0BDIR	H'200–3C0* (H'9C0)	0–15 (0)		Low	
TMU0	TUNI0	H'400 (H'400)	0–15 (0)	IPRA (15–12)	—	
TMU1	TUNI1	H'420 (H'420)	0–15 (0)	IPRA (11–8)	—	
TMU2	TUNI2	H'440 (H'440)	0–15 (0)	IPRA (7–4)	—	
RTC	ATI	H'480 (H'480)	0–15 (0)	IPRA (3–0)	High	
	PRI	H'4A0 (H'4A0)				
	CUI	H'4C0 (H'4C0)			Low	
SCI0	ERI	H'4E0 (H'4E0)	0–15 (0)	IPRB (7–4)	High	
	RXI	H'500 (H'500)				
	TXI	H'520 (H'520)				
	TEI	H'540 (H'540)			Low	
WDT	ITI	H'560 (H'560)	0–15 (0)	IPRB (15–12)	—	
REF	RCMI	H'580 (H'580)	0–15 (0)	IPRB (11–8)	High	
	ROVI	H'5A0 (H'5A0)			Low	Low

Note: * The code corresponding to an interrupt level shown in table 7.6 is set.

Table 7.6 Interrupt Level and INTEVT Code

Interrupt level	INTEVT Code
15	H'200
14	H'220
13	H'240
12	H'260
11	H'280
10	H'2A0
9	H'2C0
8	H'2E0
7	H'300
6	H'320
5	H'340
4	H'360
3	H'380
2	H'3A0
1	H'3C0

7.3 INTC Registers

7.3.1 Interrupt Priority Registers A to G (IPRA to IPRG)

Interrupt priority registers A to G (IPRA to IPRG) are 16-bit read/write registers that set priority levels from 0 to 15 for on-chip supporting module, IRQ, and PINT interrupts. These registers are initialized to H'0000 at power-on reset, and manual reset, but are not initialized in standby mode.

Bit:	15	14	13	12	11	10	9	8
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	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

Table 7.7 lists the relationship between the interrupt sources and the IPRA to IPRG bits.

Table 7.7 Interrupt Request Sources and IPRA to IPRG

Register	Bits 15 to 12	Bits 11 to 8	Bits 7 to 4	Bits 3 to 0
IPRA	TMU0	TMU1	TMU2	RTC
IPRB	WDT	REF	SCI	Reserved*
IPRC	IRQ3	IRQ2	IRQ1	IRQ0
IPRD	PINT0 to PINT7	PINT8 to PINT15	IRQ5	IRQ4
IPRE	DMAC	Reserved*	SCIF	ADC
IPRF	Reserved*	LCDC	PCC0	SIOF
IPRG	USBH	USBF0	USBF1	AFEIF

Note: * Always read as 0. Only 0 should be written in.

As listed in table 7.7, four sets of on-chip supporting modules or IRQ or PINT interrupts are assigned to each register. 4-bit groups (bits 15 to 12, bits 11 to 8, bits 7 to 4, and bits 3 to 0) are set with values from H'0 (0000) to H'F (1111). Setting H'0 means priority level 0 (masking is requested); H'F is priority level 15 (the highest level). A reset initializes IPRA to IPRG to H'0000.

7.3.2 Interrupt Control Register 0 (ICR0)

The ICR0 is a register that sets the input signal detection mode of the external interrupt input pin NMI and indicates the input signal level to the NMI pin. This register is initialized to H'0000 or H'8000 at power-on reset or manual reset, but is not initialized in standby mode.

Bit:	15	14	13	12	11	10	9	8
	NMIL	—	—	—	—	—	—	NMIE
Initial value:	0/1*	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W

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

Note: * When NMI input is high: 1; when NMI input is low: 0.

Bit 15—NMI Input Level (NMIL): Sets the level of the signal input at the NMI pin. This bit can be read to determine the NMI pin level. This bit cannot be modified.

Bit 15: NMIL	Description
0	NMI input level is low
1	NMI input level is high

Bit 8—NMI Edge Select (NMIE): Selects whether the falling or rising edge of the interrupt request signal to the NMI is detected.

Bit 8: NMIE	Description	
0	Interrupt request is detected on the falling edge of NMI input	(Initial value)
1	Interrupt request is detected on rising edge of NMI input	

Bits 14 to 9 and 7 to 0—Reserved: These bits are always read as 0. The write value should always be 0.

7.3.3 Interrupt Control Register 1 (ICR1)

The ICR1 is a 16-bit register that specifies the detection mode to external interrupt input pins, IRQ0 to IRQ5 individually: rising edge, falling edge, or low level. This register, initialized to H'4000 at power-on reset or manual reset, is not initialized in the standby mode.

Bit:	15	14	13	12	11	10	9	8
	MAI	IRQLVL	BLMSK	—	IRQ51S	IRQ50S	IRQ41S	IRQ40S
Initial value:	0	1	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	—	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
	IRQ31S	IRQ30S	IRQ21S	IRQ20S	IRQ11S	IRQ10S	IRQ01S	IRQ00S
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 15—Mask All Interrupts (MAI): Masks NMI interrupts in standby mode when set to 1. Also selects whether or not all interrupt requests are masked when a low level is being input to the NMI pin.

Bit 15: MAI	Description
0	All interrupt requests are not masked while NMI pin is receiving low-level input (Initial value)
1	All interrupt requests are masked while NMI pin is receiving low-level input

Bit 14—Interrupt Request Level Detect (IRQLVL): Selects whether the IRQ3 to IRQ0 pins are used as four independent interrupt pins or as 15-level interrupt pins encoded as $\overline{\text{IRL3}}$ to $\overline{\text{IRL0}}$.

Bit 14: IRQLVL	Description
0	Used as four independent interrupt request pins IRQ3 to IRQ0
1	Used as encoded 15-level interrupt pins as $\overline{\text{IRL3}}$ to $\overline{\text{IRL0}}$ (Initial value)

Bit 13—BL Bit Mask (BLMSK): Specifies whether NMI interrupts are masked when the BL bit of the SR register is 1.

Bit 13: BLMSK	Description	
0	NMI interrupts are masked when the BL bit is 1	(Initial value)
1	NMI interrupts are accepted regardless of the BL bit setting	

Bit 12—Reserved: This bit is always read as 0. The write value should always be 0.

Bits 11 and 10—IRQ5 Sense Select (IRQ51S and IRQ50S): Select whether the interrupt signal to the IRQ5 pin is detected at the rising edge, at the falling edge, or at low level.

Bit 11: IRQ51S	Bit 10: IRQ50S	Description
0	0	An interrupt request is detected at IRQ5 input falling edge (Initial value)
	1	An interrupt request is detected at IRQ5 input rising edge
1	0	An interrupt request is detected at IRQ5 input low level
	1	Reserved

Bits 9 and 8—IRQ4 Sense Select (IRQ41S and IRQ40S): Select whether the interrupt signal to the IRQ4 pin is detected at the rising edge, at the falling edge, or at low level.

Bit 9: IRQ41S	Bit 8: IRQ40S	Description
0	0	An interrupt request is detected at IRQ4 input falling edge (Initial value)
	1	An interrupt request is detected at IRQ4 input rising edge
1	0	An interrupt request is detected at IRQ4 input low level
	1	Reserved

Bits 7 and 6—IRQ3 Sense Select (IRQ31S and IRQ30S): Select whether the interrupt signal to the IRQ3 pin is detected at the rising edge, at the falling edge, or at low level.

Bit 7: IRQ31S	Bit 6: IRQ30S	Description
0	0	An interrupt request is detected at IRQ3 input falling edge (Initial value)
	1	An interrupt request is detected at IRQ3 input rising edge
1	0	An interrupt request is detected at IRQ3 input low level
	1	Reserved

Bits 5 and 4—IRQ2 Sense Select (IRQ21S and IRQ20S): Select whether the interrupt signal to the IRQ2 pin is detected at the rising edge, at the falling edge, or at low level.

Bit 5: IRQ21S	Bit 4: IRQ20S	Description
0	0	An interrupt request is detected at IRQ2 input falling edge (Initial value)
	1	An interrupt request is detected at IRQ2 input rising edge
1	0	An interrupt request is detected at IRQ2 input low level
	1	Reserved

Bits 3 and 2—IRQ1 Sense Select (IRQ11S and IRQ10S): Select whether the interrupt signal to the IRQ1 pin is detected at the rising edge, at the falling edge, or at low level.

Bit 3: IRQ11S	Bit 2: IRQ10S	Description
0	0	An interrupt request is detected at IRQ1 input falling edge (Initial value)
	1	An interrupt request is detected at IRQ1 input rising edge
1	0	An interrupt request is detected at IRQ1 input low level
	1	Reserved

Bits 1 and 0—IRQ0 Sense Select (IRQ01S and IRQ00S): Select whether the interrupt signal to the IRQ0 pin is detected at the rising edge, at the falling edge, or at low level.

Bit 1: IRQ01S	Bit 0: IRQ00S	Description
0	0	An interrupt request is detected at IRQ0 input falling edge (Initial value)
	1	An interrupt request is detected at IRQ0 input rising edge
1	0	An interrupt request is detected at IRQ0 input low level
	1	Reserved

7.3.4 Interrupt Control Register 2 (ICR2)

The ICR2 is a 16-bit read/write register that sets the detection mode to external interrupt input pins PINT0 to PINT15. This register is initialized to H'0000 at power-on reset or manual reset, but is not initialized in standby mode.

Bit:	15	14	13	12	11	10	9	8
	PINT15S	PINT14S	PINT13S	PINT14S	PINT11S	PINT10S	PINT9S	PINT8S
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	6	5	4	3	2	1	0
	PINT7S	PINT6S	PINT5S	PINT4S	PINT3S	PINT2S	PINT1S	PINT0S
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 15 to 0—PINT15 to PINT0 Sense Select (PINT15S to PINT0S): Select whether interrupt request signals to PINT15 to PINT0 are detected at low levels or high levels.

Bits 15 to 0: PINT15S to PINT0S	**Description**
0	Interrupt requests are detected at low level input to the PINT pins (Initial value)
1	Interrupt requests are detected at high level input to the PINT pins

7.3.5 Interrupt Control Register 3 (ICR3)

The ICR3 is a 16-bit read/write register that sets the mask to PC Card controller. This register is initialized to H'0000 at power-on reset or manual reset, but is not initialized in standby mode.

Bit:	15	14	13	12	11	10	9	8
	—	PC0SWIM	PC0IRIM	PC0SCIM	PC0CDIM	PC0RCIM	PC0BWIM	PC0BDIM
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	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

Bit 15—Reserved: This bit is always read as 0. The write value should always be 0.

Bit 14—PC0SWIM: PC Card controller0 SWI mask.

Bit 14: PC0SWIM	Description	
0	Interrupt requests is masked	(Initial value)
1	Interrupt requests is not masked	

Bit 13—PC0IRIM: PC Card controller0 IRI mask.

Bits 13: PC0IRIM	Description	
0	Interrupt requests is masked	(Initial value)
1	Interrupt requests is not masked	

Bit 12—PC0SCIM: PC Card controller0 SCI mask.

Bit 12: PC0SCIM	Description	
0	Interrupt requests is masked	(Initial value)
1	Interrupt requests is not masked	

Bit 11—PC0CDIM: PC Card controller0 CDI mask.

Bit 11: PC0CDIM	Description	
0	Interrupt requests is masked	(Initial value)
1	Interrupt requests is not masked	

Bit 10—PC0RCIM: PC Card controller0 RCI mask.

Bit 10: PC0RCIM	Description	
0	Interrupt requests is masked	(Initial value)
1	Interrupt requests is not masked	

Bit 9—PC0BWIM: PC Card controller0 BWI mask.

Bit 9: PC0BWIM	Description	
0	Interrupt requests is masked	(Initial value)
1	Interrupt requests is not masked	

Bit 8—PC0BDIM: PC Card controller0 BDI mask.

Bit 8: PC0BDIM	Description	
0	Interrupt requests is masked	(Initial value)
1	Interrupt requests is not masked	

Bits 7 to 0— Reserved: These bits are always read as 0. The write value should always be 0.

7.3.6 PINT Interrupt Enable Register (PINTER)

The PINTER is a 16-bit read/write register that enables interrupt requests input to external interrupt input pins PINT0 to PINT15. This register is initialized to H'0000 at power-on reset or manual reset, but is not initialized in standby mode.

Bit:	15	14	13	12	11	10	9	8
	PINT15E	PINT14E	PINT13E	PINT12E	PINT11E	PINT10E	PINT9E	PINT8E
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	6	5	4	3	2	1	0
	PINT7E	PINT6E	PINT5E	PINT4E	PINT3E	PINT2E	PINT1E	PINT0E
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 15 to 0—PINT15 to PINT0 Interrupt Enable (PINT15E to PINT0E): Enable whether the interrupt requests input to the PINT15 to PINT0 pins.

Bits 15 to 0: PINT15E to PINT0E	**Description**	
0	Disables PINT input interrupt requests	(Initial value)
1	Enables PINT input interrupt requests	

When all or some of these pins, PINT0 to PINT15 are not used as an interrupt input, a bit corresponding to a pin unused as an interrupt request should be set to 0.

7.3.7 Interrupt Request Register 0 (IRR0)

The IRR0 is an 8-bit register that indicates interrupt requests from external input pins IRQ0 to IRQ5 and PINT0 to PINT15. This register is initialized to H'00 at power-on reset or manual reset, but is not initialized in standby mode.

Bit:	7	6	5	4	3	2	1	0
	PINT0R	PINT1R	IRQ5R	IRQ4R	IRQ3R	IRQ2R	IRQ1R	IRQ0R
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R/W	R/W	R/W	R/W

When using edge sensing for IRQ interrupts, do the following to clear IR0.

To clear bits IRQ5R to IRQ0R to 0, read from IRR0 before writing. After confirming that the bits to be cleared to 0 are set to 1, write 0 to them. In this case write 0 only to the bits to be cleared; write 1 to the other bits. The values of the bits to which 1 is written do not change.

When level sensing is used for IRQ interrupts, bits IRQ5R to IRQ0R indicate whether or not an interrupt request has been input. They can be set and cleared by the values input to pins IRQ5R to IRQ0R alone.

Bit 7—PINT0 to PINT7 Interrupt Request (PINT0R): Indicates whether interrupt requests are input to PINT0 to PINT7 pins.

Bit 7: PINT0R	Description	
0	Interrupt requests are not input to PINT0 to PINT7 pins	(Initial value)
1	Interrupt requests are input to PINT0 to PINT7 pins.	

Bit 6—PINT8 to PINT15 Interrupt Request (PINT1R): Indicates whether interrupt requests are input to PINT8 to PINT15 pins.

Bit 6: PINT1R	Description	
0	Interrupt requests are not input to PINT8 to PINT15 pins	(Initial value)
1	Interrupt requests are input to PINT8 to PINT15 pins.	

Bit 5—IRQ5 Interrupt Request (IRQ5R): Indicates whether an interrupt request is input to the IRQ5 pin. When edge detection mode is set for IRQ5, an interrupt request is cleared by clearing the IRQ5R bit.

Bit 5: IRQ5R	Description	
0	An interrupt request is not input to IRQ5 pin	(Initial value)
1	An interrupt request is input to IRQ5 pin	

Bit 4—IRQ4 Interrupt Request (IRQ4R): Indicates whether an interrupt request is input to the IRQ4 pin. When edge detection mode is set for IRQ4, an interrupt request is cleared by clearing the IRQ4R bit.

Bit 4: IRQ4R	Description	
0	An interrupt request is not input to IRQ4 pin	(Initial value)
1	An interrupt request is input to IRQ4 pin	

Bit 3—IRQ3 Interrupt Request (IRQ3R): Indicates whether an interrupt request is input to the IRQ3 pin. When edge detection mode is set for IRQ3, an interrupt request is cleared by clearing the IRQ3R bit.

Bit 3: IRQ3R	Description	
0	An interrupt request is not input to IRQ3 pin	(Initial value)
1	An interrupt request is input to IRQ3 pin	

Bit 2—IRQ2 Interrupt Request (IRQ2R): Indicates whether an interrupt request is input to the IRQ2 pin. When edge detection mode is set for IRQ2, an interrupt request is cleared by clearing the IRQ2R bit.

Bit 2: IRQ2R	Description	
0	An interrupt request is not input to IRQ2 pin	(Initial value)
1	An interrupt request is input to IRQ2 pin	

Bit 1—IRQ1 Interrupt Request (IRQ1R): Indicates whether an interrupt request is input to the IRQ1 pin. When edge detection mode is set for IRQ1, an interrupt request is cleared by clearing the IRQ1R bit.

Bit 1: IRQ1R	Description	
0	An interrupt request is not input to IRQ1 pin	(Initial value)
1	An interrupt request is input to IRQ1 pin	

Bit 0—IRQ0 Interrupt Request (IRQ0R): Indicates whether an interrupt request is input to the IRQ0 pin. When edge detection mode is set for IRQ0, an interrupt request is cleared by clearing the IRQ0R bit.

Bit 0: IRQ0R	Description	
0	An interrupt request is not input to IRQ0 pin	(Initial value)
1	An interrupt request is input to IRQ0 pin	

7.3.8 Interrupt Request Register 1 (IRR1)

The IRR1 is an 8-bit read-only register that indicates whether DMAC or interrupt requests are generated. This register is initialized to H'00 at power-on reset or manual reset, but is not initialized in standby mode.

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	DEI3R	DEI2R	DEI1R	DEI0R
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bits 7 to 4—Reserved: These bits are always read as 0. The write value should always be 0.

Bit 3—DEI3 Interrupt Request (DEI3R): Indicates whether a DEI3 (DMAC) interrupt request is generated.

Bit 3: DEI3R	Description	
0	A DEI3 interrupt request is not generated	(Initial value)
1	A DEI3 interrupt request is generated	

Bit 2—DEI2 Interrupt Request (DEI2R): Indicates whether a DEI2 (DMAC) interrupt request is generated.

Bit 2: DEI2R	Description	
0	A DEI2 interrupt request is not generated	(Initial value)
1	A DEI2 interrupt request is generated	

Bit 1—DEI1 Interrupt Request (DEI1R): Indicates whether a DEI1 (DMAC) interrupt request is generated.

Bit 1: DEI1R	Description	
0	A DEI1 interrupt request is not generated	(Initial value)
1	A DEI1 interrupt request is generated	

Bit 0—DEI0 Interrupt Request (DEI0R): Indicates whether a DEI0 (DMAC) interrupt request is generated.

Bit 0: DEI0R	Description	
0	A DEI0 interrupt request is not generated	(Initial value)
1	A DEI0 interrupt request is generated	

7.3.9 Interrupt Request Register 2 (IRR2)

The IRR2 is an 8-bit read-only register that indicates whether A/D converter, or SCIF interrupt requests are generated. This register is initialized to H'00 at power-on reset or manual reset, but is not initialized in standby mode.

Bit:	7	6	5	4	3	2	1	0
	—	—	—	ADIR	TXI2R	BRI2R	RXI2R	ERI2R
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bits 7 to 5—Reserved: These bits are always read as 0. The write value should always be 0.

Bit 4—ADI Interrupt Request (ADIR): Indicates whether an ADI (ADC) interrupt request is generated.

Bit 4: ADIR	Description	
0	An ADI interrupt request is not generated	(Initial value)
1	An ADI interrupt request is generated	

Bit 3—TXI2 Interrupt Request (TXI2R): Indicates whether a TXI2 (SCIF) interrupt request is generated.

Bit 3: TXI2R	Description	
0	A TXI2 interrupt request is not generated	(Initial value)
1	A TXI2 interrupt request is generated	

Bit 2—BRI2 Interrupt Request (BRI2R): Indicates whether a BRI2 (SCIF) interrupt request is generated.

Bit 2: BRI2R	Description	
0	A BRI2 interrupt request is not generated	(Initial value)
1	A BRI2 interrupt request is generated	

Bit 1—RXI2 Interrupt Request (RXI2R): Indicates whether an RXI2 (SCIF) interrupt request is generated.

Bit 1: RXI2R	Description	
0	An RXI2 interrupt request is not generated	(Initial value)
1	An RXI2 interrupt request is generated	

Bit 0—ERI2 Interrupt Request (ERI2R): Indicates whether an ERI2 (SCIF) interrupt request is generated.

Bit 0: ERI2R	Description	
0	An ERI2 interrupt request is not generated	(Initial value)
1	An ERI2 interrupt request is generated	

7.3.10 Interrupt Request Register 3 (IRR3)

The IRR3 is a 16-bit read-only register that indicates whether PC Card controller, USB Controller or LCDC interrupt requests are generated. This register is initialized to H'0000 at power-on reset or manual reset, but is not initialized in standby mode.

Bit:	15	14	13	12	11	10	9	8
	LCDCIR	PC0SWIR	PC0IRIR	PC0SCIR	PC0CDIR	PC0RCIR	PC0BWIR	PC0BDIR
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit:	7	6	5	4	3	2	1	0
	USBHIR	USBF0IR	USBF1IR	AFEIFIR	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit 15—LCDCI interrupt request (LCDCIR): Indicates whether a LCDCI (LCDC) interrupt request is generated.

Bit 15: LCDCIR	Description	
0	A LCDCI interrupt request is not generated	(Initial value)
1	A LCDCI interrupt request is generated	

Bit 14—PC0SWI Interrupt Request (PC0SWIR): Indicates whether a PC0SWI (PCC0) interrupt request is generated.

Bit 14: PC0SWIR	Description	
0	A PC0SWI interrupt request is not generated	(Initial value)
1	A PC0SWI interrupt request is generated	

Bit 13—PC0IRI Interrupt Request (PC0IRIR): Indicates whether a PC0IREQ (PCC0) interrupt request is generated.

Bit 13: PC0IRIR	Description	
0	A PC0IRI interrupt request is not generated	(Initial value)
1	A PC0IRI interrupt request is generated	

Bit 12—PC0SCI Interrupt Request (PC0SCIR): Indicates whether a PC0SCI (PCC0) interrupt request is generated.

Bit 12: PC0SCIR	Description	
0	A PC0SCI interrupt request is not generated	(Initial value)
1	A PC0SCI interrupt request is generated	

Bit 11—PC0CDI Interrupt Request (PC0CDIR): Indicates whether a PC0CDI (PCC0) interrupt request is generated.

Bit 11: PC0CDIR	Description	
0	A PC0CDI interrupt request is not generated	(Initial value)
1	A PC0CDI interrupt request is generated	

Bit 10—PC0RCI Interrupt Request (PC0RCIR): Indicates whether a PC0RCI (PCC0) interrupt request is generated.

Bit 10: PC0RCIR	Description	
0	A PC0RCI interrupt request is not generated	(Initial value)
1	A PC0RCI interrupt request is generated	

Bit 9—PC0BWI Interrupt Request (PC0BWIR): Indicates whether a PC0BWI (PCC0) interrupt request is generated.

Bit 9: PC0BWIR	Description	
0	A PC0BWI interrupt request is not generated	(Initial value)
1	A PC0BWI interrupt request is generated	

Bit 8—PC0BDI Interrupt Request (PC0BDIR): Indicates whether a PC0BDI (PCC0) interrupt request is generated.

Bit 8: PC0BDIR	Description	
0	A PC0BDI interrupt request is not generated	(Initial value)
1	A PC0BDI interrupt request is generated	

Bit 7—USBHI Interrupt Request (USBHIR): Indicates whether a USBHI (USB Host) interrupt request is generated.

Bit 7: USBHIR	Description	
0	A USBHI interrupt request is not generated	(Initial value)
1	A USBHI interrupt request is generated	

Bit 6—USBF0I Interrupt Request (USBF0IR): Indicates whether a USBF0I (USB function) interrupt request is generated.

Bit 6: USBF0IR	Description	
0	A USBF0I interrupt request is not generated	(Initial value)
1	A USBF0I interrupt request is generated	

Bit 5—USBF1I Interrupt Request (USBF1IR): Indicates whether a USBF1I (USB function) interrupt request is generated.

Bit 5: USBF1IR	Description	
0	A USBF1I interrupt request is not generated	(Initial value)
1	A USBF1I interrupt request is generated	

Bit 4—AFEIFI Interrupt Request (AFEIFIR): Indicates whether a AFEIFI (AFE I/F) interrupt request is generated.

Bit 4: AFEIFIR	Description	
0	An AFE I/F interrupt request is not generated	(Initial value)
1	An AFE I/F interrupt request is generated	

Bits 3 to 0—Reserved: These bits are always read as 0.

7.3.11 Interrupt Request Register 4 (IRR4)

The IRR4 is a 16-bit read-only register that indicates whether SIOF interrupt requests are generated. This register is initialized to H'0000 at power-on reset or manual reset, but is not initialized in standby mode.

Bit:	15	14	13	12	11	10	9	8
	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	ERI	TXI	RXI	CCI
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bits 15 to 4—Reserved: These bits are always read as 0. The write value should always be 0.

Bit 3—ERI Interrupt Request (ERI): Indicates whether a ERI (SIOF) interrupt request is generated.

Bit 3: ERI	Description	
0	ERI interrupt request is not generated	(Initial value)
1	ERI interrupt request is generated	

Bit 2—TXI Interrupt Request (TXI): Indicates whether a TXI (SIOF) interrupt request is generated.

Bit 2:TXI	Description	
0	TXI interrupt request is not generated	(Initial value)
1	TXI interrupt request is generated	

Bit 1—RXI Interrupt Request (RXI): Indicates whether a RXI (SIOF) interrupt request is generated.

Bit 1: RXI	Description	
0	RXI interrupt request is not generated	(Initial value)
1	RXI interrupt request is generated	

Bit 0—CCI Interrupt Request (CCI): Indicates whether a CCI (CCI) interrupt request is generated.

Bit 0: CCI	Description	
0	CCI interrupt request is not generated	(Initial value)
1	CCI interrupt request is generated	

7.4 INTC Operation

7.4.1 Interrupt Sequence

The sequence of interrupt operations is explained below. Figure 7.3 is a flowchart of the operations.

1. The interrupt request sources send interrupt request signals to the interrupt controller.
2. The interrupt controller selects the highest priority interrupt from the interrupt requests sent, following the priority levels set in interrupt priority registers A to G (IPRA to IPRG). Lower priority interrupts are held pending. If two of these interrupts have the same priority level or if multiple interrupts occur within a single module, the interrupt with the highest default priority or the highest priority within its IPR setting unit (as indicated in tables 7.4 and 7.5) is selected.
3. The priority level of the interrupt selected by the interrupt controller is compared with the interrupt mask bits (I3 to I0) in the status register (SR) of the CPU. If the request priority level is higher than the level in bits I3 to I0, the interrupt controller accepts the interrupt and sends an interrupt request signal to the CPU.
4. Detection timing: The INTC operates, and notifies the CPU of interrupt requests, in synchronization with the peripheral clock (Pϕ). The CPU receives an interrupt at a break in instructions.
5. The interrupt source code is set in the interrupt event registers (INTEVT and INTEVT2).
6. The status register (SR) and program counter (PC) are saved to SSR and SPC, respectively.
7. The block bit (BL), mode bit (MD), and register bank bit (RB) in SR are set to 1.
8. The CPU jumps to the start address of the interrupt handler (the sum of the value set in the vector base register (VBR) and H'00000600). This jump is not a delayed branch. The interrupt handler may branch with the INTEVT and INTEVT2 register value as its offset in order to identify the interrupt source. This enables it to branch to the processing routine for the individual interrupt source.

Notes: 1. The interrupt mask bits (I3 to I0) in the status register (SR) are not changed by acceptance of an interrupt in the SH7727.

2. The interrupt source flag should be cleared in the interrupt handler. To ensure that an interrupt request that should have been cleared is not inadvertently accepted again, read the interrupt source flag after it has been cleared, then wait for the interval shown in table 7.8 (Time for priority decision and SR mask bit comparison) before clearing the BL bit or executing an RTE instruction.

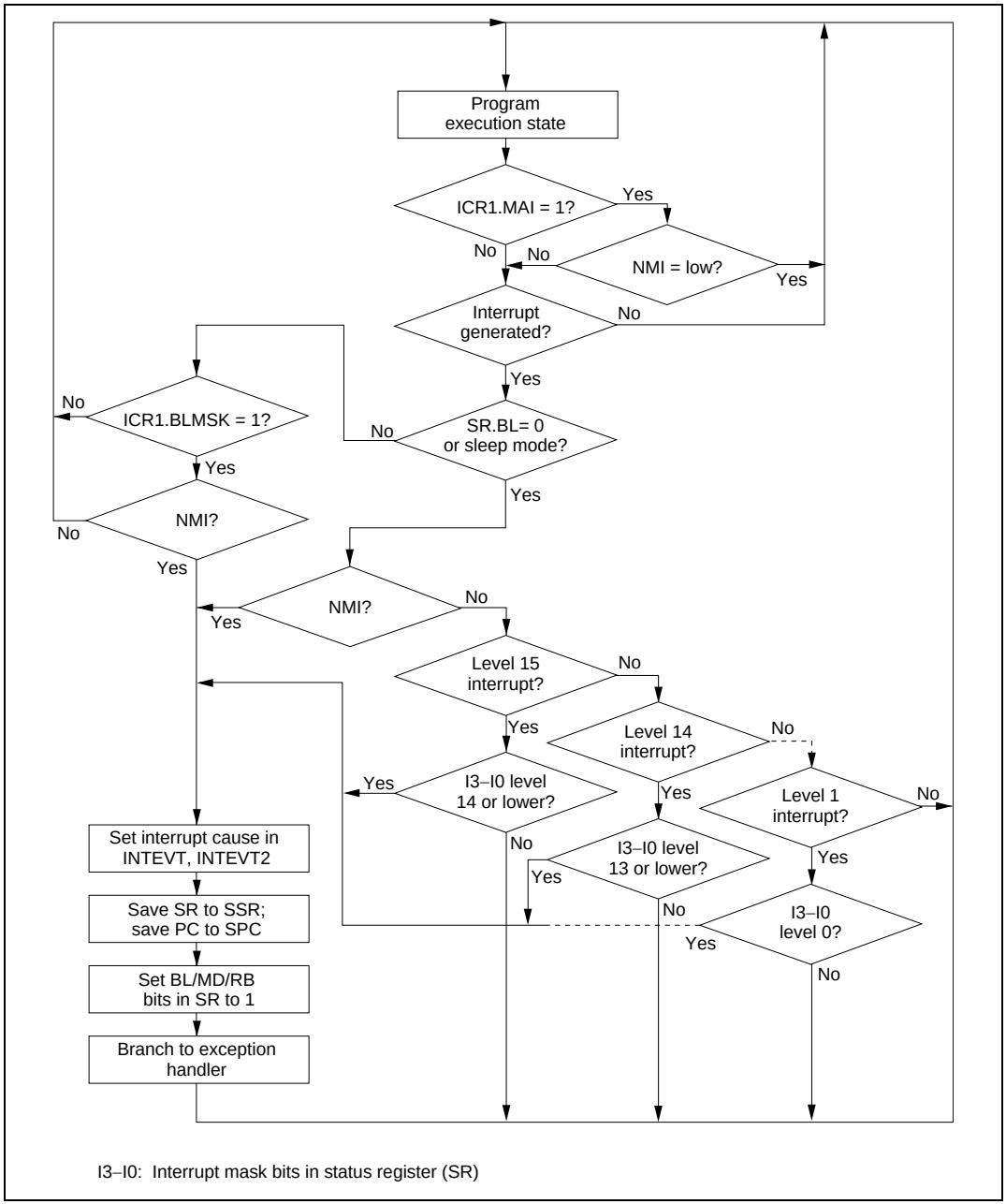


Figure 7.3 Interrupt Operation Flowchart

7.4.2 Multiple Interrupts

When processing multiple interrupts, an interrupt handler should include the following procedures:

1. Branch to a specific interrupt handler corresponding to a code set in INTEVT and INTEVT2. The code in INTEVT and INTEVT2 can be used as a branch-offset for branching to the specific handler.
2. Clear the cause of the interrupt in each specific handler.
3. Save SSR and SPC to the memory.
4. Clear the BL bit in SR, and set the accepted interrupt level in the interrupt mask bits in SR.
5. Handle the interrupt.
6. Execute the RTE instruction.

When these procedures are followed in order, an interrupt of higher priority than the one being handled can be accepted after clearing BL in step 4.

7.5 Interrupt Response Time

The time from generation of an interrupt request until interrupt exception handling is performed and fetching of the first instruction of the exception handler is started (the interrupt response time) is shown in table 7.8. Figure 7.4 shows an example of pipeline operation when an IRL interrupt is accepted. When SR.BL is 1, interrupt exception handling is masked, and is kept waiting until completion of an instruction that clears BL to 0.

Table 7.8 Interrupt Response Time

Item	Number of States: NMI	IRQ	PINT	Peripheral Modules	Notes
Time for priority decision and SR mask bit comparison	0.5 × Icyc + 0.5 × Bcyc + 0.5 × Pcyc	0.5 × Icyc + 1 × Bcyc + 4.5 × Pcyc*4	0.5 × Icyc + 3.5 × Pcyc	0.5 × Icyc + 1.5 × Pcyc*5	
				0.5 × Icyc + 3 × Pcyc*6	
Wait time until end of sequence being executed by CPU	X (≥ 0) × Icyc	X (≥ 0) × Icyc	X (≥ 0) × Icyc	X (≥ 0) × Icyc	Interrupt exception handling is kept waiting until the executing instruction ends. If the number of instruction execution states is S*1, the maximum wait time is: X = S – 1. However, if BL is set to 1 by instruction execution or by an exception, interrupt exception handling is deferred until completion of an instruction that clears BL to 0. If the following instruction masks interrupt exception handling, the processing may be further deferred.
Time from interrupt exception handling (save of SR and PC) until fetch of first instruction of exception handler is started	5 × Icyc	5 × Icyc	5 × Icyc	5 × Icyc	

Item		Number of States				Notes
		NMI	IRQ	PINT	Peripheral Modules	
Response time	Total	(5.5 + X) × Icyc + 0.5 × Bcyc + 0.5 × Pcyc	(5.5 + X) × Icyc + 1 × Bcyc + 4.5 × Pcyc[*4]	(5.5 + X) × Icyc + 3.5 × Pcyc[*5]	(5.5 + X) × Icyc + 1.5 × Pcyc[*5]	
					(5.5 + X) × Icyc + 3 × Pcyc[*6]	
	Minimum case[*2]	7.5	16.5	12.5	8.5[*5]/11.5[*6]	At 60-MHz (CKIO = 30) operation: 0.13–0.28 µs
	Maximum case[*3]	8.5 + S	26.5 + S	18.5 + S	10.5 + S[*5] 16.5 + S[*6]	At 60-MHz (CKIO = 15) operation: 0.26–0.56 µs (in case of operand cache-hit) At 60-MHz (CKIO = 15) operation: 0.29–0.59 µs (when external memory access is performed with wait = 0)

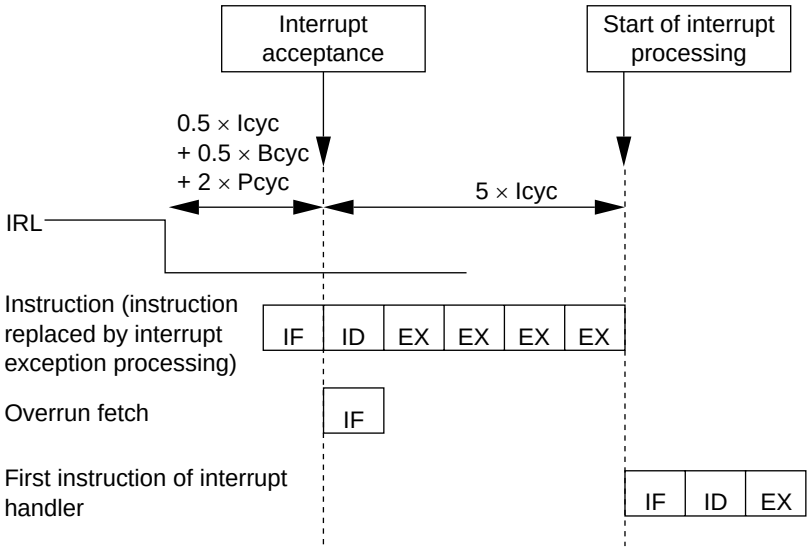
Icyc: Duration of one cycle of internal clock supplied to CPU.

Bcyc: Duration of one CKIO cycle.

Pcyc: Duration of one cycle of peripheral clock supplied to peripheral modules.

Notes: 1. S also includes the memory access wait time.
The processing requiring the maximum execution time is LDC.L @Rm+, SR. When the memory access is a cache-hit, this requires seven instruction execution cycles. When the external access is performed, the corresponding number of cycles must be added. There are also instructions that perform two external memory accesses; if the external memory access is slow, the number of instruction execution cycles will increase accordingly.

2. The internal clock: CKIO: peripheral clock ratio is 2:1:1.
3. The internal clock: CKIO: peripheral clock ratio is 4:1:1.
4. IRQ mode
5. Modules: TMU, RTC, SCI, WDT, REFC
6. Modules: DMAC, ADC, SCIF, LCDC, PCC, USB host, USB function, AFE interface



IF: Instruction fetch: Instruction is fetched from memory in which program is stored.
ID: Instruction decode: Fetched instruction is decoded.
EX: Instruction execution: Data operation and address calculation are performed in accordance with result of decoding.

Figure 7.4 Example of Pipeline Operations when IRL Interrupt is Accepted

Section 8 User Break Controller

8.1 Overview

The user break controller (UBC) provides functions that simplify program debugging. These functions make it easy to design an effective self-monitoring debugger, enabling the chip to debug programs without using an in-circuit emulator. Break conditions that can be set in the UBC are instruction fetch or data read/write access, data size, data contents, address value, and timing in the case of instruction fetch.

8.1.1 Features

The UBC has the following features:

- The following break comparison conditions can be set.
  Number of break channels: two channels (channels A and B)
  User break can be requested as either the independent or sequential condition on channels A and B (sequential break setting: channel A and, then channel B match with logical AND, but not in the same bus cycle).
  - Address (Compares 40 bits comprised of a 32-bit logical address prefixed with an ASID address
    Comparison bits are maskable in 32-bit units, user can easily program it to mask addresses at bottom 12 bits (4-k page), bottom 10 bits (1-k page), or any size of page, etc.
    The 8-bit ASID checking is from MMU control to indicate hit or not hit.)
    One of four address buses (CPU address bus (LAB), cache address bus (IAB),
    X-memory address bus (XAB) and Y-memory address bus (YAB)) can be selected.
  - Data (only on channel B, 32-bit maskable)
    One of the four data buses (CPU data bus (LDB), cache data bus (IDB), X-memory data bus (XDB) and Y-memory data bus (YDB)) can be selected.
  - Bus master: CPU cycle or DMAC cycle
  - Bus cycle: instruction fetch or data access
  - Read/write
  - Operand size: byte, word, or longword
- User break is generated upon satisfying break conditions. A user-designed user-break condition exception processing routine can be run.
- In an instruction fetch cycle, it can be selected that a break is set before or after an instruction is executed.

- Maximum repeat times for the break condition: $2^{12} - 1$ times. (It is only for channel B)
- Eight pairs of branch source/destination buffers.

8.1.2 Block Diagram

Figure 8.1 is a block diagram of the UBC.

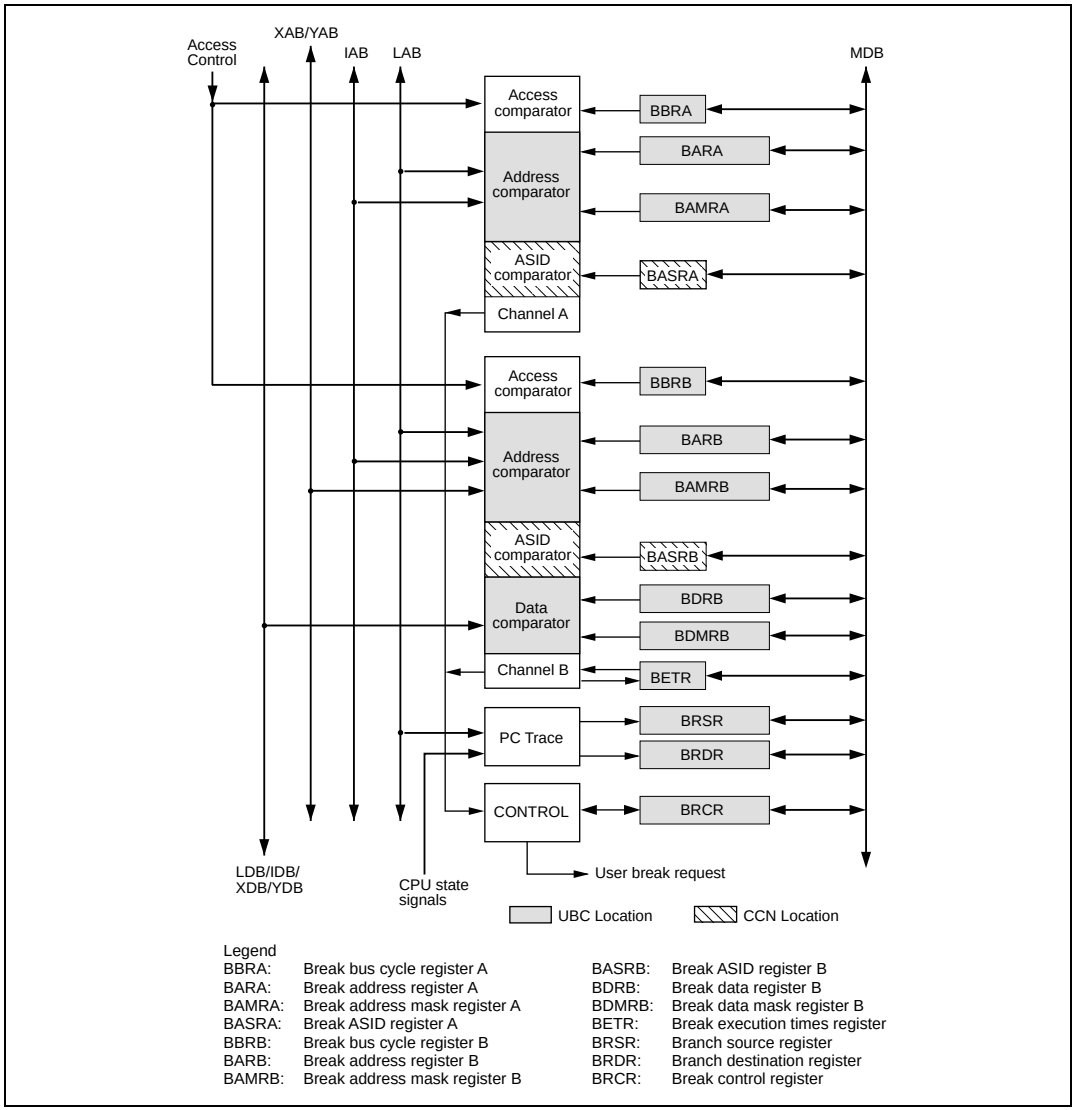


Figure 8.1 Block Diagram of User Break Controller

8.1.3 Register Configuration

Table 8.1 Register Configuration

Name	Abbr.	R/W	Initial Value*1	Address	Access Size	Location
Break address register A	BARA	R/W	H'00000000	H'FFFFFFB0	32	UBC
Break address mask register A	BAMRA	R/W	H'00000000	H'FFFFFFB4	32	UBC
Break bus cycle register A	BBRA	R/W	H'0000	H'FFFFFFB8	16	UBC
Break address register B	BARB	R/W	H'00000000	H'FFFFFFA0	32	UBC
Break address mask register B	BAMRB	R/W	H'00000000	H'FFFFFFA4	32	UBC
Break bus cycle register B	BBRB	R/W	H'0000	H'FFFFFFA8	16	UBC
Break data register B	BDRB	R/W	H'00000000	H'FFFFFF90	32	UBC
Break data mask register B	BDMRB	R/W	H'00000000	H'FFFFFF94	32	UBC
Break control register	BRCR	R/W	H'00000000	H'FFFFFF98	32	UBC
Execution count break register	BETR	R/W	H'0000	H'FFFFFF9C	16	UBC
Branch source register	BRSR	R	Undefined*2	H'FFFFFFAC	32	UBC
Branch destination register	BRDR	R	Undefined*2	H'FFFFFFBC	32	UBC
Break ASID register A	BASRA	R/W	Undefined	H'FFFFFFE4	8	CCN
Break ASID register B	BASRB	R/W	Undefined	H'FFFFFFE8	8	CCN

Notes: 1. Initialized by power-on reset. Values held in standby state and undefined by manual resets.

2. Bit 31 of BRSR and BRDR (valid flag) is initialized by power-on resets. But other bits are not initialized.

8.2 Register Descriptions

8.2.1 Break Address Register A (BARA)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	BAA 31	BAA 30	BAA 29	BAA 28	BAA 27	BAA 26	BAA 25	BAA 24	BAA 23	BAA 22	BAA 21	BAA 20	BAA 19	BAA 18	BAA 17	BAA 16
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

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BAA 15	BAA 14	BAA 13	BAA 12	BAA 11	BAA 10	BAA 9	BAA 8	BAA 7	BAA 6	BAA 5	BAA 4	BAA 3	BAA 2	BAA 1	BAA 0
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

BARA is a 32-bit read/write register. BARA specifies the address used as a break condition in channel A. A power-on reset initializes BARA to H'00000000.

Bits 31 to 0—Break Address A31 to A0 (BAA31 to BAA0): Stores the address on the LAB or IAB specifying break conditions of channel A.

8.2.2 Break Address Mask Register A (BAMRA)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	BAMA 31	BAMA 30	BAMA 29	BAMA 28	BAMA 27	BAMA 26	BAMA 25	BAMA 24	BAMA 23	BAMA 22	BAMA 21	BAMA 20	BAMA 19	BAMA 18	BAMA 17	BAMA 16
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

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BAMA 15	BAMA 14	BAMA 13	BAMA 12	BAMA 11	BAMA 10	BAMA 9	BAMA 8	BAMA 7	BAMA 6	BAMA 5	BAMA 4	BAMA 3	BAMA 2	BAMA 1	BAMA 0
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

BAMRA is a 32-bit read/write register. BAMRA specifies bits masked in the break address specified by BARA. A power-on reset initializes BAMRA to H'00000000.

Bits 31 to 0—Break Address Mask Register A31 to A0 (BAMA31 to BAMA0): Specifies bits masked in the channel A break address bits specified by BARA (BAA31 to BAA0).

Bits 31 to 0: BAMAn	Description
0	Break address bit BAAn of channel A is included in the break condition (Initial value)
1	Break address bit BAAn of channel A is masked and is not included in the break condition

n = 31 to 0

8.2.3 Break Bus Cycle Register A (BBRA)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	CDA1	CDA0	IDA1	IDA0	RWA1	RWA0	SZA1	SZA0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Break bus cycle register A (BBRA) is a 16-bit read/write register, which specifies (1) CPU cycle or DMAC cycle, (2) instruction fetch or data access, (3) read or write, and (4) operand size in the break conditions of channel A. A power-on reset initializes BBRA to H'0000.

Bits 15 to 8—Reserved: These bits are always read as 0. The write value should always be 0.

Bits 7 and 6—CPU Cycle/DMAC Cycle Select A (CDA1, CDA0): Selects the CPU cycle or DMAC cycle as the bus cycle of the channel A break condition.

Bit 7: CDA1	Bit 6: CDA0	Description	
0	0	Condition comparison is not performed	(Initial value)
*	1	The break condition is the CPU cycle	
1	0	The break condition is the DMAC cycle	

Note: * Don’t care

Bits 5 and 4—Instruction Fetch/Data Access Select A (IDA1, IDA0): Selects the instruction fetch cycle or data access cycle as the bus cycle of the channel A break condition.

Bit 5: IDA1	Bit 4: IDA0	Description
0	0	Condition comparison is not performed (Initial value)
	1	The break condition is the instruction fetch cycle
1	0	The break condition is the data access cycle
	1	The break condition is the instruction fetch cycle or data access cycle

Bits 3 and 2—Read/Write Select A (RWA1, RWA0): Selects the read cycle or write cycle as the bus cycle of the channel A break condition.

Bit 3: RWA1	Bit 2: RWA0	Description
0	0	Condition comparison is not performed (Initial value)
	1	The break condition is the read cycle
1	0	The break condition is the write cycle
	1	The break condition is the read cycle or write cycle

Bits 1 and 0—Operand Size Select A (SZA1, SZA0): Selects the operand size of the bus cycle for the channel A break condition.

Bit 1: SZA1	Bit 0: SZA0	Description
0	0	The break condition does not include operand size (Initial value)
	1	The break condition is byte access
1	0	The break condition is word access
	1	The break condition is longword access

8.2.4 Break Address Register B (BARB)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	BAB 31	BAB 30	BAB 29	BAB 28	BAB 27	BAB 26	BAB 25	BAB 24	BAB 23	BAB 22	BAB 21	BAB 20	BAB 19	BAB 18	BAB 17	BAB 16
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

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BAB 15	BAB 14	BAB 13	BAB 12	BAB 11	BAB 10	BAB 9	BAB 8	BAB 7	BAB 6	BAB 5	BAB 4	BAB 3	BAB 2	BAB 1	BAB 0
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

BARB is a 32-bit read/write register. BARB specifies the address used as a break condition in channel B. Control bits XYE and XYS in the BBRB selects an address bus for break condition B. If the XYE is 0, then BARB specifies the break address on logic or internal bus, LAB or IAB. If the XYE is 1, then the BAB31 to 16 specifies the break address on XAB (bits 15 to 1) and the BAB15 to 0 specifies the break address on YAB (bits 15 to 1). However, you have to choose one of two address buses for the break. A power-on reset initializes BARB to H'00000000.

	BAB31 to 16	**BAB15 to 0**
XYE = 0	L(I) AB31 to 16	L(I) AB15 to 0
XYE = 1	XAB15 to 1 (XYS = 0)	YAB15 to 1 (XYS = 1)

8.2.5 Break Address Mask Register B (BAMRB)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	BAMB 31	BAMB 30	BAMB 29	BAMB 28	BAMB 27	BAMB 26	BAMB 25	BAMB 24	BAMB 23	BAMB 22	BAMB 21	BAMB 20	BAMB 19	BAMB 18	BAMB 17	BAMB 16
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

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BAMB 15	BAMB 14	BAMB 13	BAMB 12	BAMB 11	BAMB 10	BAMB 9	BAMB 8	BAMB 7	BAMB 6	BAMB 5	BAMB 4	BAMB 3	BAMB 2	BAMB 1	BAMB 0
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

BAMRB is a 32-bit read/write register. BAMRB specifies bits masked in the break address specified by BARB. A power-on reset initializes BAMRB to H'00000000.

	BAMB31 to 16	**BAMB15 to 0**
XYE = 0	Mask L(I) AB31 to 16	Mask L(I) AB15 to 0
XYE = 1	Mask XAB15 to 1 (XYS = 0)	Mask YAB15 to 1 (XYS = 1)

Bits 31 to 0:

BAMBn	**Description**
0	Break address BABn of channel B is included in the break condition (Initial value)
1	Break address BABn of channel B is masked and is not included in the break condition

n = 31 to 0

8.2.6 Break Data Register B (BDRB)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	BDB 31	BDB 30	BDB 29	BDB 28	BDB 27	BDB 26	BDB 25	BDB 24	BDB 23	BDB 22	BDB 21	BDB 20	BDB 19	BDB 18	BDB 17	BDB 16
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

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BDB 15	BDB 14	BDB 13	BDB 12	BDB 11	BDB 10	BDB 9	BDB 8	BDB 7	BDB 6	BDB 5	BDB 4	BDB 3	BDB 2	BDB 1	BDB 0
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

BDRB is a 32-bit read/write register. The control bits XYE and XYS in BBRB select a data bus for break condition B. If the XYE is 0, then BDRB specifies the break data on LDB or IDB. If the XYE is 1, then BDB 31 to 16 specifies the break data on XDB (bits 15 to 0) and BDB 15 to 0 specifies the break data on YDB (bits 15 to 0). However, you have to choose one of two data buses for the break. A power-on reset initializes BDRB to H'00000000.

	BDB31 to 16	**BDB15 to 0**
XYE = 0	L(I) DB31 to 16	L(I) DB15 to 0
XYE = 1	XDB15 to 0 (XYS = 0)	YDB15 to 0 (XYS = 1)

8.2.7 Break Data Mask Register B (BDMRB)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	BDMB 31	BDMB 30	BDMB 29	BDMB 28	BDMB 27	BDMB 26	BDMB 25	BDMB 24	BDMB 23	BDMB 22	BDMB 21	BDMB 20	BDMB 19	BDMB 18	BDMB 17	BDMB 16
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

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BDMB 15	BDMB 14	BDMB 13	BDMB 12	BDMB 11	BDMB 10	BDMB 9	BDMB 8	BDMB 7	BDMB 6	BDMB 5	BDMB 4	BDMB 3	BDMB 2	BDMB 1	BDMB 0
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

BDMRB is a 32-bit read/write register. BDMRB specifies bits masked in the break data specified by BDRB. A power-on reset initializes BDMRB to H'00000000.

	BDMB31 to 16	**BDMB15 to 0**
XYE = 0	Mask L(I) DB31 to 16	Mask L(I) DB15 to 0
XYE = 1	Mask XDB15 to 0 (XYS = 0)	Mask YDB15 to 0 (XYS = 1)

Bits 31 to 0:

BDMBn	**Description**
0	Break data BDBn of channel B is included in the break condition (Initial value)
1	Break data BDBn of channel B is masked and is not included in the break condition

n = 31 to 0

Notes: 1. Specify an operand size when including the value of the data bus in the break condition.
2. When a byte size is selected as a break condition, the same break data (byte size) must be set both in bits 15 to 8 and in bits 7 to 0 in BDRB.

8.2.8 Break Bus Cycle Register B (BBRB)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	XYE	XYS	CDB1	CDB0	IDB1	IDB0	RWB1	RWB0	SZB1	SZB0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Break bus cycle register B (BBRB) is a 16-bit read/write register, which specifies (1) logic or internal bus (L or I bus), X bus, of Y bus, (2) CPU cycle or DMAC cycle, (3) instruction fetch or data access, (4) read/write, and (5) operand size in the break conditions of channel B. A power-on reset initializes BBRB to H'0000.

Bits 15 to 10—Reserved: These bits are always read as 0. The write value should always be 0.

Bit 9—X/Y Memory Bus Enable (XYE): Selects the logic bus or internal bus (L bus or I bus) or the X/Y memory bus as the bus of the channel B break condition.

Bit 9: XYE	Description
0	Select internal bus (I bus) for the channel B break condition
1	Select X/Y memory bus (X/Y bus) for the channel B break condition

Bits 8—X or Y Memory Bus Select (XYS): Selects the X bus or the Y bus as the bus of the channel B break condition.

Bit 8: XYS	Description
0	Select the X bus for the channel B break condition
1	Select the Y bus for the channel B break condition

Bits 7 and 6—CPU Cycle/DMAC Cycle Select B (CDB1 and CDB0): Select the CPU cycle or DMAC cycle as the bus cycle of the channel B break condition.

Bit 7: CDB1	Bit 6: CDB0	Description	
0	0	Condition comparison is not performed	(Initial value)
*	1	The break condition is the CPU cycle	
1	0	The break condition is the DMAC cycle	

Note: * Don’t care.

Bits 5 and 4—Instruction Fetch/Data Access Select B (IDB1 and IDB0): Select the instruction fetch cycle or data access cycle as the bus cycle of the channel B break condition.

Bit 5: IDB1	Bit 4: IDB0	Description	
0	0	Condition comparison is not performed	(Initial value)
	1	The break condition is the instruction fetch cycle	
1	0	The break condition is the data access cycle	
	1	The break condition is the instruction fetch cycle or data access cycle	

Bits 3 and 2—Read/Write Select B (RWB1 and RWB0): Select the read cycle or write cycle as the bus cycle of the channel B break condition.

Bit 3: RWB1	Bit 2: RWB0	Description	
0	0	Condition comparison is not performed	(Initial value)
	1	The break condition is the read cycle	
1	0	The break condition is the write cycle	
	1	The break condition is the read cycle or write cycle	

Bits 1 and 0—Operand Size Select B (SZB1 and SZB0): Select the operand size of the bus cycle for the channel B break condition.

Bit 1: SZB1	Bit 0: SZB0	Description
0	0	The break condition does not include operand size (Initial value)
	1	The break condition is byte access
1	0	The break condition is word access
	1	The break condition is longword access

8.2.9 Break Control Register (BRCR)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	BASMA	BASMB	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SCMFCA	SCMFCB	SCMFDA	SCMFDB	PCTE	PCBA	—	—	DBEB	PCBB	—	—	SEQ	—	—	ETBE
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	R	R/W	R/W	R	R	R/W	R	R	R/W

BRCR sets the following conditions:

1. Channels A and B are used in two independent channels condition or under the sequential condition.
2. A break is set before or after instruction execution.
3. A break is set by the number of execution times.
4. Determine whether to include data bus on channel B in comparison conditions.
5. Enable PC trace.
6. Enable the ASID check.

The break control register (BRCR) is a 32-bit read/write register that has break conditions match flags and bits for setting a variety of break conditions.

A power-on reset initializes BRCR to H'00000000.

Bits 31 to 22—Reserved: These bits are always read as 0. The write value should always be 0.

Bit 21—Break ASID Mask A (BASMA): Specifies whether the bits of the channel A break ASID7 to ASID0 (BASA7 to BASA0) set in BASRA are masked or not.

Bit 21: BASMA	Description
0	All BASRA bits are included in break condition, and ASID is checked (Initial value)
1	No BASRA bits are included in break condition, and ASID is not checked

Bit 20—Break ASID Mask B (BASMB): Specifies whether the bits of channel B break ASID7 to ASID0 (BASB7 to BASB0) set in BASRB are masked or not.

Bit 20: BASMB	Description
0	All BASRB bits are included in break condition, and ASID is checked (Initial value)
1	No BASRB bits are included in break condition, and ASID is not checked

Bits 19 to 16—Reserved: These bits are always read as 0. The write value should always be 0.

Bit 15—CPU Condition Match Flag A (SCMFCA): When the CPU bus cycle condition in the break conditions set for channel A is satisfied, this flag is set to 1 (not cleared to 0). In order to clear this flag, write 0 into this bit.

Bit 15: SCMFCA	Description	
0	The CPU cycle condition for channel A does not match	(Initial value)
1	The CPU cycle condition for channel A matches	

Bit 14—CPU Condition Match Flag B (SCMFCB): When the CPU bus cycle condition in the break conditions set for channel B is satisfied, this flag is set to 1 (not cleared to 0). In order to clear this flag, write 0 into this bit.

Bit 14: SCMFCB	Description	
0	The CPU cycle condition for channel B does not match	(Initial value)
1	The CPU cycle condition for channel B matches	

Bit 13—DMAC Condition Match Flag A (SCMFDA): When the on-chip DMAC bus cycle condition in the break conditions set for channel A is satisfied, this flag is set to 1 (not cleared to 0). In order to clear this flag, write 0 into this bit.

Bit 13: SCMFDA	Description	
0	The DMAC cycle condition for channel A does not match	(Initial value)
1	The DMAC cycle condition for channel A matches	

Bit 12—DMAC Condition Match Flag B (SCMFDB): When the on-chip DMAC bus cycle condition in the break conditions set for channel B is satisfied, this flag is set to 1 (not cleared to 0). In order to clear this flag, write 0 into this bit.

Bit 12: SCMFDB	Description	
0	The DMAC cycle condition for channel B does not match	(Initial value)
1	The DMAC cycle condition for channel B matches	

Bit 11—PC Trace Enable (PCTE): Enables PC trace.

Bit 11: PCTE	Description	
0	Disables PC trace	(Initial value)
1	Enables PC trace	

Bit 10—PC Break Select A (PCBA): Selects the break timing of the instruction fetch cycle for channel A as before or after instruction execution.

Bit 10: PCBA	Description	
0	PC break of channel A is set before instruction execution	(Initial value)
1	PC break of channel A is set after instruction execution	

Bits 9 and 8—Reserved: These bits are always read as 0. The write value should always be 0.

Bit 7—Data Break Enable B (DBEB): Selects whether or not the data bus condition is included in the break condition of channel B.

Bit 7: DBEB	Description	
0	No data bus condition is included in the condition of channel B	(Initial value)
1	The data bus condition is included in the condition of channel B	

Bit 6—PC Break Select B (PCBB): Selects the break timing of the instruction fetch cycle for channel B as before or after instruction execution.

Bit 6: PCBB	Description	
0	PC break of channel B is set before instruction execution	(Initial value)
1	PC break of channel B is set after instruction execution	

Bits 5 and 4—Reserved: These bits are always read as 0. The write value should always be 0.

Bit 3—Sequence Condition Select (SEQ): Selects two conditions of channels A and B as independent or sequential.

Bit 3: SEQ	Description	
0	Channels A and B are compared under the independent condition	(Initial value)
1	Channels A and B are compared under the sequential condition	

Bits 2 and 1—Reserved: These bits are always read as 0. The write value should always be 0.

Bit 0—The Number of Execution Times Break Enable (ETBE): Enable the execution-times break condition only on channel B. If this bit is 1 (break enable), a user break is issued when the number of break conditions matches with the number of execution times that is specified by the BETR register.

Bit 0: ETBE	Description	
0	The execution-times break condition is disabled on channel B	(Initial value)
1	The execution-times break condition is enabled on channel B	

8.2.10 Execution Times Break Register (BETR)

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

When the execution-times break condition of channel B is enabled, this register specifies the number of execution times to make the break. The maximum number is $2^{12} - 1$ times. A power-on reset initializes BETR to H'0000. When a break condition is satisfied, it decreases the BETR. A break is issued when the break condition is satisfied after the BETR becomes H'0001. Bits 15 to 12 are always read as 0 and 0 should always be written in these bits.

8.2.11 Branch Source Register (BRSR)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SVF	PID2	PID1	PID0	BSA 27	BSA 26	BSA 25	BSA 24	BSA 23	BSA 22	BSA 21	BSA 20	BSA 19	BSA 18	BSA 17	BSA 16
Initial value:	0	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BSA 15	BSA 14	BSA 13	BSA 12	BSA 11	BSA 10	BSA 9	BSA 8	BSA 7	BSA 6	BSA 5	BSA 4	BSA 3	BSA 2	BSA 1	BSA 0
Initial value:	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Note: * Undefined

BRSR is a 32-bit read register. BRSR stores the last fetched address before branch and the pointer (3 bits) which indicates the number of cycles from fetch to execution for the last executed instruction. BRSR has the flag bit that is set to 1 when branch occurs. This flag bit is cleared to 0, when BRSR is read and also initialized by power-on resets or manual resets. Other bits are not initialized by reset. Eight BRSR registers have queue structure and a stored register is shifted every branch.

Bit 31—BRSR Valid Flag (SVF): Indicates whether the address and the pointer by which the branch source address can be calculated. When a branch source address is fetched, this flag is set to 1. This flag is cleared to 0 in reading BRSR.

Bit 31: SVF	Description
0	The value of BRSR register is invalid
1	The value of BRSR register is valid

Bits 30 to 28—Instruction Decode Pointer (PID2 to PID0): PID is a 3-bit binary pointer (0 to 7). These bits indicate the instruction buffer number which stores the last executed instruction before branch.

Bits 30 to 28: PID	Description
Even	PID indicates the instruction buffer number.
Odd	PiD+2 indicates the instruction buffer number

Bits 27 to 0—Branch Source Address (BSA27 to BSA0): These bits store the last fetched address before branch.

8.2.12 Branch Destination Register (BRDR)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DVF	—	—	—	BDA 27	BDA 26	BDA 25	BDA 24	BDA 23	BDA 22	BDA 21	BDA 20	BDA 19	BDA 18	BDA 17	BDA 16
Initial value:	0	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BDA 15	BDA 14	BDA 13	BDA 12	BDA 11	BDA 10	BDA 9	BDA 8	BDA 7	BDA 6	BDA 5	BDA 4	BDA 3	BDA 2	BDA 1	BDA 0
Initial value:	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Note: * Undefined

BRDR is a 32-bit read register. BRDR stores the branch destination fetch address. BRDR has the flag bit that is set to 1 when branch occurs. This flag bit is cleared to 0, when BRDR is read and also initialized by power-on resets or manual resets. Other bits are not initialized by resets. Eight BRDR registers have queue structure and a stored register is shifted every branch.

Bit 31—BRDR Valid Flag (DVF): Indicates whether a branch destination address is stored. When a branch destination address is fetched, this flag is set to 1. This flag is cleared to 0 in reading BRDR.

Bit 31: DVF	Description
0	The value of BRDR register is invalid
1	The value of BRDR register is valid

Bits 30 to 28—Reserved: These bits are always read as 0. The write value should always be 0.

Bits 27 to 0—Branch Destination Address (BDA27 to BDA0): These bits store the first fetched address after branch.

8.2.13 Break ASID Register A (BASRA)

Bit:	7	6	5	4	3	2	1	0
	BASA7	BASA6	BASA5	BASA4	BASA3	BASA2	BASA1	BASA0
Initial value:	*	*	*	*	*	*	*	*
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: * Undefined

Break ASID register A (BASRA) is an 8-bit read/write register that specifies the ASID that serves as the break condition for channel A. It is not initialized by resets. It is located in CCN.

Bits 7 to 0—Break ASID A7 to 0 (BASA7 to BASA0): These bits store the ASID (bits 7 to 0) that is the channel A break condition.

8.2.14 Break ASID Register B (BASRB)

Bit:	7	6	5	4	3	2	1	0
	BASB7	BASB6	BASB5	BASB4	BASB3	BASB2	BASB1	BASB0
Initial value:	*	*	*	*	*	*	*	*
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: * Undefined

Break ASID register B (BASRB) is an 8-bit read/write register that specifies the ASID that serves as the break condition for channel B. It is not initialized by resets. It is located in CCN.

Bits 7 to 0: Break ASID A7 to 0 (BASB7 to BASB0): These bits store the ASID (bits 7 to 0) that is the channel B break condition.

8.3 Operation Description

8.3.1 Flow of the User Break Operation

The flow from setting of break conditions to user break exception processing is described below:

1. The break addresses and the corresponding ASIDs are loaded in the break address registers (BARA and BARB) and break ASID registers (BASRA and BASRB in CCN). The masked addresses are set in the break address mask registers (BAMRA and BAMRB). The break data is set in the break data register (BDRB). The masked data is set in the break data mask register (BDMRB). The breaking bus conditions are set in the break bus cycle registers (BBRA and BBRB). Three groups of the BBRA and BBRB (CPU cycle/DMAC cycle select, instruction fetch/data access select, and read/write select) are each set. No user break will be generated if even one of these groups is set with 00. The respective conditions are set in the bits of the BRCR.
2. When the break conditions are satisfied, the UBC sends a user break request to the interrupt controller. The break type will be sent to CPU indicating the instruction fetch, pre/post instruction break, data access break, or on-chip I/O access/LDTLB break. When conditions match up, the CPU condition match flags (SCMFCA and SCMFCB) and DMAC condition match flags (SCMFDA and SCMFDB) for the respective channels are set.
3. The appropriate condition match flags (SCMFCA, SCMFDA, SCMFCB, and SCMFDB) can be used to check if the set conditions match or not. The matching of the conditions sets flags, but they are not reset. 0 must first be written to them before they can be used again.
4. There is a chance that the data access break and its following instruction fetch break occur around the same time, there will be only one break request to the CPU, but these two break channel match flags could both be set.

8.3.2 Break on Instruction Fetch Cycle

1. When CPU/instruction fetch/read/word or longword is set in the break bus cycle registers (BBRA/BBRB), the break condition becomes the CPU instruction fetch cycle. Whether it then breaks before or after the execution of the instruction can then be selected with the PCBA/PCBB bits of the break control register (BRCR) for the appropriate channel.
2. An instruction set for a break before execution breaks when it is confirmed that the instruction has been fetched and will be executed. This means this feature cannot be used on instructions fetched by overrun (instructions fetched at a branch or during an interrupt transition, but not to be executed). When this kind of break is set for the delay slot of a delay branch instruction, the break is generated prior to execution of the instruction that then first accepts the break. Meanwhile, the break set for pre-instruction-break on delay slot instruction and post-instruction-break on SLEEP instruction are also prohibited.

3. When the condition is specified to be occurred after execution, the instruction set with the break condition is executed and then the break is generated prior to the execution of the next instruction. As with pre-execution breaks, this cannot be used with overrun fetch instructions. When this kind of break is set for a delay branch instruction, the break is generated at the instruction that then first accepts the break.
4. When an instruction fetch cycle is set for channel B, break data register B (BDRB) is ignored. There is thus no need to set break data for the break of the instruction fetch cycle.

8.3.3 Break by Data Access Cycle

1. The memory cycles in which CPU data access breaks occur are from instructions.
2. The relationship between the data access cycle address and the comparison condition for operand size are listed in table 8.2:

Table 8.2 Data Access Cycle Addresses and Operand Size Comparison Conditions

Access Size	Address Compared
Longword	Compares break address register bits 31 to 2 to address bus bits 31 to 2
Word	Compares break address register bits 31 to 1 to address bus bits 31 to 1
Byte	Compares break address register bits 31 to 0 to address bus bits 31 to 0

This means that when address H'00001003 is set without specifying the size condition, for example, the bus cycle in which the break condition is satisfied is as follows (where other conditions are met).

Longword access at H'00001000

Word access at H'00001002

Byte access at H'00001003

3. When the data value is included in the break conditions on B channel:

   When the data value is included in the break conditions, either longword, word, or byte is specified as the operand size of the break bus cycle registers (BBRA and BBRB). When data values are included in break conditions, a break is generated when the address conditions and data conditions both match. To specify byte data for this case, set the same data in two bytes at bits 15 to 8 and bits 7 to 0 of the break data register B (BDRB) and break data mask register B (BDMRB). When word or byte is set, bits 31 to 16 of BDRB and BDMRB are ignored.
4. When the DMAC data access is included in the break condition:

   When the address is included in the break condition on DMAC data access, the operand size of the break bus cycle registers (BBRA and BBRB) should be byte, word or no operand size specification. When the data value is included, select either byte or word.

8.3.4 Break on X/Y-Memory Bus Cycle

1. The break condition on X/Y-memory bus cycle is specified only in channel B. If XYE in BBRB is set to 1, break address and break data on X/Y-memory bus are selected. At this time, select X-memory bus or Y-memory bus by specifying XYS in BBRB. The Break condition cannot include both X-memory and Y-memory at the same time. The break condition is applied to X/Y-memory bus cycle by specifying CPU/data access/read or write/word or no operand size specification in the break bus cycle register B (BBRB).
2. When X-memory address is selected as the break condition, specify X-memory address in upper 16 bits in BARB and BAMRB. When Y-memory address is selected, specify Y-memory address in lower 16 bits. Specification of X/Y-memory data is the same for BDRB and BDMRB.

8.3.5 Sequential Break

1. By specifying SEQ in BRCR is set to 1, the sequential break is issued when channel B break condition matches after channel A break condition matches. A user break is ignored even if channel B break condition matches before channel A break condition matches. When channels A and B condition match at the same time, the sequential break is not issued.
2. In sequential break specification, internal/X/Y bus can be selected and the execution times break condition can be also specified. For example, when the execution times break condition is specified, the break condition is satisfied at channel B condition match with BETR = H'0001 after channel A condition match.

8.3.6 Value of Saved Program Counter

The PC when a break occurs is saved to the SPC in user breaks. The PC value saved is as follows depending on the type of break.

1. When instruction fetch (before instruction execution) is specified as a break condition:
   The value of the program counter (PC) saved is the address of the instruction that matches the break condition. The fetched instruction is not executed, and a break occurs before it.
2. When instruction fetch (after instruction execution) is specified as a break condition:
   The PC value saved is the address of the instruction to be executed following the instruction in which the break condition matches. The fetched instruction is executed, and a break occurs before the execution of the next instruction.

3. When data access (address only) is specified as a break condition:

   The PC value is the address of the instruction to be executed following the instruction that matched the break condition. The instruction that matched the condition is executed and the break occurs before the next instruction is executed.

4. When data access (address + data) is specified as a break condition:

   The PC value is the start address of the instruction that follows the instruction already executed when break processing started up. When a data value is added to the break conditions, the place where the break will occur cannot be specified exactly. The break will occur before the execution of an instruction fetched around the data access where the break occurred.

8.3.7 PC Trace

1. Setting PCTE in BRCR to 1 enables PC traces. When branch (branch instruction, repeat, and interrupt) is generated, the address from which the branch source address can be calculated and the branch destination address are stored in BRSR and BRDR, respectively. The branch address and the pointer, which corresponds to the branch, are included in BRSR.

2. The branch address before branch occurs can be calculated from the address and the pointer stored in BRSR. The expression from BSA (the address in BRSR), PID (the pointer in BRSR), and IA (the instruction address before branch occurs) is as follows: IA = BSA – 2 * PID.

   Notes are needed when an interrupt (a branch) is issued before the branch destination instruction is executed. In case of the next figure, the instruction “Exec” executed immediately before branch is calculated by IA = BSA – 2 * PID. However, when branch “branch” has delay slot and the destination address is 4n + 2 address, the address “Dest” which is specified by branch instruction is stored in BRSR (Dest = BSA). Therefore, as IA = BSA – 2 * PID is not applied to this case, this PID is invalid. The case where BSA is 4n + 2 boundary is applied only to this case and then some cases are classified as follows:

```
Exec: branch  Dest
Dest: instr       (not executed)
      interrupt
Int:  interrupt routine
```

   If the PID value is odd, instruction buffer indicates PID+2 buffer. However, these expressions in this table are accounted for it. Therefore, the true branch source address is calculated with BSA and PID values stored in BRSR.

3. The branch address before branch occurrence, IA, has different values due to some kinds of branch.

   a. Branch instruction

      The branch instruction address

   b. Repeat

      The instruction before the last instruction of a repeat loop

```
Repeat_Start: inst (1);  → BRDR
              inst (2);
                :
              inst (n-1); → the address calculated from BRSR
Repeat_End:   inst (n);
```

   c. Interrupt

      The last instruction executed before interrupt

      The top address of interrupt routine is stored in BRDR.

   In a repeat loop with instructions less than three, no instruction fetch cycle appears and branch source address is unknown. Therefore, PC trace is disabled.

4. BRSR and BRDR have eight pairs of queue structures. The top of queues is read first when the address stored in the PC trace register is read. BRSR and BRDR share the read pointer. Read BRSR and BRDR in order, the queue only shifts after BRDR is read. When reading BRDR, longword access should be used. Also, the PC trace has a trace pointer, which initially points to the bottom of the queues. The first pair of branch addresses will be stored at the bottom of the queues, then push up when next pairs come into the queues. The trace pointer will points to the next branch address to be executed, unless it got push out of the queues. When the branch address has been executed, the trace pointer will shift down to next pair of addresses, until it reaches the bottom of the queues. After switching the PCTE bit (in BRCR) off and on, the values in the queues are invalid. The read pointer stay at the position before PCTE is switched, but the trace pointer restart at the bottom of the queues.

8.3.8 Usage Examples

Break Condition Specified to a CPU Instruction Fetch Cycle

1. Register specifications
   BARA = H'00000404, BAMRA = H'00000000, BBRA = H'0054, BARB = H'00008010, BAMRB = H'00000006, BBRB = H'0054, BDRB = H'00000000, BDMRB = H'00000000, BRCR = H'00300400

   Specified conditions: Channel A/channel B independent mode
   - Channel A
     Address: H'00000404, Address mask: H'00000000
     Bus cycle: CPU/instruction fetch (after instruction execution)/read (operand size is not included in the condition)
     No ASID check is included
   - Channel B
     Address: H'00008010, Address mask: H'00000006
     Data: H'00000000, Data mask: H'00000000
     Bus cycle: CPU/instruction fetch (before instruction execution)/read (operand size is not included in the condition)
     No ASID check is included

   A user break occurs after an instruction of address H'00000404 is executed or before instructions of adresses H'00008010 to H'00008016 are executed.

2. Register specifications
   BARA = H'00037226, BAMRA = H'00000000, BBRA = H'0056, BARB = H'0003722E, BAMRB = H'00000000, BBRB = H'0056, BDRB = H'00000000, BDMRB = H'00000000, BRCR = H'00000008, BASRA = H'80, BASRB = H'70

   Specified conditions: Channel A/channel B sequence mode
   - Channel A
     Address: H'00037226, Address mask: H'00000000, ASID = H'80
     Bus cycle: CPU/instruction fetch (before instruction execution)/read/word
   - Channel B
     Address: H'0003722E, Address mask: H'00000000, ASID = H'70
     Data: H'00000000, Data mask: H'00000000
     Bus cycle: CPU/instruction fetch (before instruction execution)/read/word

An instruction with ASID = H'80 and address H'00037226 is executed, and a user break occurs before an instruction with ASID = H'70 and address H'0003722E is executed.

3. Register specifications

   BARA = H'00027128, BAMRA = H'00000000, BBRA = H'005A, BARB = H'00031415, BAMRB = H'00000000, BBRB = H'0054, BDRB = H'00000000, BDMRB = H'00000000, BRCR = H'00300000

   Specified conditions: Channel A/channel B independent mode

   - Channel A

     Address: H'00027128, Address mask: H'00000000

     Bus cycle: CPU/instruction fetch (before instruction execution)/write/word

     No ASID check is included

   - Channel B

     Address: H'00031415, Address mask: H'00000000

     Data: H'00000000, Data mask: H'00000000

     Bus cycle: CPU/instruction fetch (before instruction execution)/read (operand size is not included in the condition)

     No ASID check is included

   On channel A, no user break occurs since instruction fetch is not a write cycle. On channel B, no user break occurs since instruction fetch is performed for an even address.

4. Register specifications

   BARA = H'00037226, BAMRA = H'00000000, BBRA = H'005A, BARB = H'0003722E, BAMRB = H'00000000, BBRB = H'0056, BDRB = H'00000000, BDMRB = H'00000000, BRCR = H'00000008, BASRA = H'80, BASRB = H'70

   Specified conditions: Channel A/channel B sequence mode

   - Channel A

     Address: H'00037226, Address mask: H'00000000, ASID: H'80

     Bus cycle: CPU/instruction fetch (before instruction execution)/write/word

   - Channel B

     Address: H'0003722E, Address mask: H'00000000, ASID: H'70

     Data: H'00000000, Data mask: H'00000000

     Bus cycle: CPU/instruction fetch (before instruction execution)/read/word

   Since instruction fetch is not a write cycle on channel A, a sequence condition does not match. Therefore, no user break occurs.

5. Register specifications

   BARA = H'00000500, BAMRA = H'00000000, BBRA = H'0057, BARB = H'00001000, BAMRB = H'00000000, BBRB = H'0057, BDRB = H'00000000, BDMRB = H'00000000, BRCR = H'00300001, BETR = H'0005

   Specified conditions: Channel A/channel B independent mode

   - Channel A

     Address: H'00000500, Address mask: H'00000000

     Bus cycle: CPU/instruction fetch (before instruction execution)/read/longword
   - Channel B

     Address: H'00001000, Address mask: H'00000000

     Data: H'00000000, Data mask: H'00000000

     Bus cycle: CPU/instruction fetch (before instruction execution)/read/longword

     The number of execution-times break enable (5 times)

   On channel A, a user break occurs before an instruction of address H'00000500 is executed. On channel B, a user break occurs before the fifth instruction execution after instructions of address H'00001000 are executed four times.

6. Register specifications

   BARA = H'00008404, BAMRA = H'00000FFF, BBRA = H'0054, BARB = H'00008010, BAMRB = H'00000006, BBRB = H'0054, BDRB = H'00000000, BDMRB = H'00000000, BRCR = H'00000400, BASRA = H'80, BASRB = H'70

   Specified conditions: Channel A/channel B independent mode

   - Channel A

     Address: H'00008404, Address mask: H'00000FFF, ASID: H'80

     Bus cycle: CPU/instruction fetch (after instruction execution)/read (operand size is not included in the condition)
   - Channel B

     Address: H'00008010, Address mask: H'00000006, ASID: H'70

     Data: H'00000000, Data mask: H'00000000

     Bus cycle: CPU/instruction fetch (before instruction execution)/read (operand size is not included in the condition)

   A user break occurs after an instruction with ASID = H'80 and address H'00008000 to H'00008FFE is executed or before instructions with ASID = H'70 and addresses H'00008010 to H'00008016 are executed.

Break Condition Specified to a CPU Data Access Cycle

1. Register specifications

   BARA = H'00123456, BAMRA = H'00000000, BBRA = H'0064, BARB = H'000ABCDE, BAMRB = H'000000FF, BBRB = H'006A, BDRB = H'0000A512, BDMRB = H'00000000, BRCR = H'00000080, BASRA = H'80, BASRB = H'70

   Specified conditions: Channel A/channel B independent mode

   - Channel A

     Address: H'00123456, Address mask: H'00000000, ASID: H'80

     Bus cycle: CPU/data access/read (operand size is not included in the condition)
   - Channel B

     Address: H'000ABCDE, Address mask: H'000000FF, ASID: H'70

     Data: H'0000A512, Data mask: H'00000000

     Bus cycle: CPU/data access/write/word

   On channel A, a user break occurs with ASID = H'80 during longword read to address H'00123454, word read to address H'00123456, or byte read to address H'00123456. On channel B, a user break occurs with ASID = H'70 when word H'A512 is written in addresses H'000ABC00 to H'000ABCFE.

2. Register specifications

   BARA = H'01000000, BAMRA = H'00000000, BBRA = H'0066, BARB = H'0000F000, BAMRB = H'FFFF0000, BBRB = H'036A, BDRB = H'00004567, BDMRB = H'00000000, BRCR = H'00300080

   Specified conditions: Channel A/channel B independent mode

   - Channel A

     Address: H'01000000, Address mask: H'00000000

     Bus cycle: CPU/data access/read/word

     No ASID check is included
   - Channel B

     Y Address: H'0001F000, Address mask: H'FFFF0000

     Data: H'00004567, Data mask: H'00000000

     Bus cycle: CPU/data access/write/word

     No ASID check is included

   On channel A, a user break occurs during word read to address H'01000000 on the memory space. On channel B, a user break occurs when word H'4567 is written in address H'0001F000

on Y memory space. The X/Y-memory space is changed by a mode specification.

Break Condition Specified to a DMAC Data Access Cycle

1. Register specifications
   BARA = H'00314156, BAMRA = H'00000000, BBRA = H'0094, BARB = H'00055555, BAMRB = H'00000000, BBRB = H'00A9, BDRB = H'00000078, BDMRB = H'0000000F, BRCR = H'00000080, BASRA = H'80, BASRB = H'70

   Specified conditions: Channel A/channel B independent mode
   - Channel A
     Address: H'00314156, Address mask: H'00000000, ASID: H'80
     Bus cycle: DMAC/instruction fetch/read (operand size is not included in the condition)
   - Channel B
     Address: H'00055555, Address mask: H'00000000, ASID: H'70
     Data: H'00000078, Data mask: H'0000000F
     Bus cycle: DMAC/data access/write/byte

   On channel A, no user break occurs since instruction fetch is not performed in DMAC cycles. On channel B, a user break occurs with ASID = H'70 when the DMAC writes byte H'7* in address H'00055555.

8.3.9 Usage Notes

1. Only CPU can read/write UBC registers.

2. UBC cannot monitor CPU and DMAC access in the same channel.

3. Notes in specification of sequential break are described below:
   (1) A condition match occurs when B-channel match occurs in a bus cycle after an A-channel match occurs in another bus cycle in sequential break setting. Therefore, no condition match occurs even if a bus cycle, in which an A-channel match and a channel B match occur simultaneously, is set.
   (2) Since the CPU has a pipeline configuration, the pipeline determines the order of an instruction fetch cycle and a memory cycle. Therefore, when a channel condition matches in the order of bus cycles, a sequential condition is satisfied.
   (3) When the bus cycle condition for channel A is specified as a break before execution (PCBA = 0 in BRCR) and an instruction fetch cycle (in BBRA), the attention is as follows. A break is issued and condition match flags in BRCR are set to 1, when the bus cycle conditions both for channels A and B match simultaneously.

4. The change of a UBC register value is executed in MA (memory access) stage. Therefore, even if the break condition matches in the instruction fetch address following the instruction in which the pre-execution break is specified as the break condition, no break occurs. In order to know the timing UBC register is changed, read the last written register. Instructions after then are valid for the newly written register value.

5. Notes in specifying the instruction during repeat execution with repeat instruction as the break condition are as follows: When the instruction during repeat execution is specified as the break condition,
   (1) The break is not issued during repeat execution, which has fewer than three instructions.
   (2) When the execution times break is set, no instruction fetch from memory occurs during repeat execution under three instructions. Therefore, the execution times register BETR is not decreased.

6. The branch instruction should not be executed as soon as PC trace register BRSR and BRDR are read.

7. When PC breaks and TLB exceptions or errors occur in the same instruction. The priority is as follows:
   (1) Break and instruction fetch exceptions: Instruction fetch exception occurs first.
   (2) Break before execution and operand exception: Break before execution occurs first.
   (3) Break after execution and operand exception: Operand exception occurs first.

Section 9 Power-Down Modes and Software Reset

9.1 Overview

In the power-down modes, all CPU and some on-chip supporting module functions are halted. This lowers power consumption. In particular, the X/Y memory can be stopped to significantly reduce power consumption. Software reset function enables each module to reset itself.

9.1.1 Power-Down Modes

The SH7727 has three power-down modes:

1. Sleep mode
2. Standby mode
3. Module standby function (TMU, RTC, SCI, X/Y memory, UBC, DMAC, DAC, ADC, SCIF, LCDC, PCC, USBH, USBF, AFEIF, and SIOF on-chip supporting modules)
4. Hardware standby mode

Table 9.1 shows the transition conditions for entering any mode from the program execution state, the CPU and supporting module states in each mode, and the procedures for canceling each mode.

Table 9.1 Power-Down Modes

Mode	Transition Conditions	State							Canceling Procedure
		CPG	CPU	CPU Register	On-Chip Memory	On-Chip Supporting Modules	Pins	External Memory	
Sleep mode	Execute SLEEP instruction with STBY bit cleared to 0 in STBCR*7	Runs	Halts	Held	Held	Run	Held	Refresh	1. Interrupt 2. Reset
Standby mode	Execute SLEEP instruction with STBY bit set to 1 in STBCR*4 *5	Halts	Halts	Held	Held	Halt*1	Held	Self-refresh	1. Interrupt 2. Reset
Module standby function	Set MSTP bit of STBCR to 1*6	Runs	Runs or halts	Held	Held	Specified module halts	*2	Refresh	1. Clear MSTP bit to 0 2. Reset
Hardware standby mode	Drive CA pin low	Halts	Halts	Held	Held	Halt*3	Held	Self-refresh	Power-on reset

Notes: 1. The RTC runs if the START bit in RCR2 is set to 1 (see section 16, Realtime Clock (RTC)). TMU runs when output of the RTC is used as input to its counter (see section 15, Timer (TMU)).

2. Depends on the on-chip supporting module.
   TMU external pin: Held
   SCI external pin: Reset

3. The RTC runs if the START bit in RCR2 is set to 1. TMU does not run.

4. USB and LCDC must be stopped before entering standby mode.
   1) To stop LCDC, set 0 to DON bit.
   2) To stop the USB Host Controller, set USBRESET bit in the HcControl register.

5. For LCDC, refer to the LPS bit in LDPMMR to confirm that power-off sequence has been completed before entering standby-mode.

6. When putting the RTC into module standby mode, first access one or more of registers RTC, SCI, and TMU. Then put the RTC into module standby mode.

7. Do not cause the CPU to transition to sleep mode, or cancel sleep mode, during a transmit or receive operation in which the USB function controller or SIOF uses DMA.

9.1.2 Pin Configuration

Table 9.2 lists the pins used for the power-down modes.

Table 9.2 Pin Configuration

Pin Name	Symbol	I/O	Description
Processing state 1	STATUS1	O	Indicate operating state of the processor.
Processing state 0	STATUS0	O	HH: Reset, HL: Sleep mode, LH: Standby mode, LL: Normal operation

Note: H means high level, and L means low level.

9.1.3 Register Configuration

Table 9.3 shows the configuration of the control register for the power-down modes.

Table 9.3 Register Configuration

Name	Abbreviation	R/W	Initial Value	Address	Access Size
Standby control register	STBCR	R/W	H'00*1	H'FFFFFF82	8
Standby control register 2	STBCR2	R/W	H'00*1	H'FFFFFF88	8
Standby control register 3	STBCR3	R/W	H'00*1	H'04000230 (H'A4000230)*2	8
Software reset register	SRSTR	R/W	H'00*1	H'04000232 (H'A4000232)*2	8

Notes: 1. Initialized by power-on resets. Not initialized by manual resets but the contents are held.

2. The addresses in parentheses ( ) should be used when no address translation by the MMU is involved.

9.2 Register Description

9.2.1 Standby Control Register (STBCR)

The standby control register (STBCR) is an 8-bit readable/writable register that sets the power-down mode. STBCR is initialized to H'00 by a power-on reset.

Bit:	7	6	5	4	3	2	1	0
	STBY	—	—	STBXTL	—	MSTP2	MSTP1	MSTP0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R	R	R/W	R	R/W	R/W	R/W

Bit 7—Standby (STBY): Specifies transition to standby mode.

Bit 7: STBY	Description	
0	Executing SLEEP instruction puts the chip into sleep mode.	(Initial value)
1	Executing SLEEP instruction puts the chip into standby mode.	

Bits 6, 5, and 3—Reserved: These bits are always read as 0. The write value should always as 0.

Bit 4—Standby Crystal (STBXTL): Enables/disables crystal oscillation in standby mode.

Bit 4: STBXTL	Description	
0	Crystal oscillation in standby mode disabled	(Initial value)
1	Crystal oscillation in standby mode enabled	

Bit 2—Module Stop 2 (MSTP2): Specifies halting the clock supply to the timer unit (TMU) in the on-chip supporting module. When the MSTP2 bit is set to 1, the clock supply to the TMU is halted.

Bit 2: MSTP2	Description	
0	TMU runs.	(Initial value)
1	Clock supply to TMU is halted.	

Bit 1—Module Stop 1 (MSTP1): Specifies halting the clock supply to the realtime clock (RTC) in the on-chip supporting module. When the MSTP1 bit is set to 1, the clock supply to RTC is halted. When the clock halts, all RTC registers cannot be accessed, but the counter keeps running.

Bit 1: MSTP1	Description	
0	RTC runs.	(Initial value)
1	Clock supply to RTC is halted.	

Bit 0—Module Stop 0 (MSTP0): Specifies halting the clock supply to the serial communication interface (SCI) in the on-chip supporting module. When the MSTP0 bit is set to 1, the clock supply to the SCI is halted.

Bit 0: MSTP0	Description	
0	SCI runs.	(Initial value)
1	Clock supply to SCI is halted.	

9.2.2 Standby Control Register 2 (STBCR2)

The standby control register 2 (STBCR2) is an 8-bit readable/writable register that controls the operation of the peripheral modules in the normal mode and sleep mode. STBCR is initialized to H'00 by a power-on reset.

Bit:	7	6	5	4	3	2	1	0
	MSTP9	MDCHG	MSTP8	MSTP7	MSTP6	MSTP5	MSTP4	—
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

Bit 7— Module Stop 9 (MSTP9): Specifies halting the clock supply to the X/Y memory. When the MSTP9 bit is set to 1, the clock supply to the X/Y memory is halted. Halting of the clock supply to the X/Y memory must be controlled by software (any access is not blocked by hardware).

Bit 7: MSTP9	Description	
0	X/Y memory runs	(Initial value)
1	Clock supply to X/Y memory is halted	

Bit 6—MD5 to MD0 Pin Control (MDCHG): Specifies whether or not pins MD5 to MD0 are switched in standby mode. When this bit is set to 1, the MD5 to MD0 pin values are latched when returning from standby mode by means of a reset or interrupt.

Bit 6: MDCHG	Description	
0	Pins MD5 to MD0 are not switched in standby mode	(Initial value)
1	Pins MD5 to MD0 are switched in standby mode	

Bit 5— Module Stop 8 (MSTP8): Specifies halting the clock supply to the user break controller (UBC) in the on-chip supporting module. When the MSTP8 bit is set to 1, the clock supply to the UBC is halted.

Bit 5: MSTP8	Description	
0	UBC runs	(Initial value)
1	Clock supply to UBC is halted	

Bit 4—Module Stop 7 (MSTP7): Specifies halting of clock supply to the direct memory access controller (DMAC) in the on-chip supporting module. When the MSTP7 bit is set to 1, the clock supply to the DMAC is halted.

Bit 4: MSTP7	Description	
0	DMAC runs	(Initial value)
1	Clock supply to DMAC halted	

Bit 3—Module Stop 6 (MSTP6): Specifies halting of clock supply to the D/A converter (DAC) in the on-chip supporting module. When the MSTP6 bit is set to 1, the clock supply to the DAC is halted.

Bit 3: MSTP6	Description	
0	DAC runs	(Initial value)
1	Clock supply to DAC halted	

Bit 2—Module Stop 5 (MSTP5): Specifies halting of clock supply to the A/D converter (ADC) in the on-chip supporting module. When the MSTP5 bit is set to 1, the clock supply to the ADC is halted and all registers are initialized.

Bit 2: MSTP5	Description	
0	ADC runs	(Initial value)
1	Clock supply to ADC halted, and all registers initialized	

Bit 1—Module Stop 4 (MSTP4): Specifies halting the clock supply to the serial communication interface SCI (SCIF) with FIFO. When the MSTP4 bit is set to 1, the clock supply to the SCIF is halted

Bit 1: MSTP4	Description	
0	SCIF runs	(Initial value)
1	Clock supply to SCI2 (SCIF) halted	

Bit 0— Reserved: This bit is always read as 0. The write value should always as 0.

9.2.3 Standby Control Register 3 (STBCR3)

The standby control register 3 (STBCR3) is an 8-bit readable/writable register that controls standby operation for the on-chip supporting modules. STBCR3 is initialized to H'00 by a power-on reset.

Bit:	7	6	5	4	3	2	1	0
	MSTP17	—	MSTP15	MSTP14	MSTP13	—	MSTP11	MSTP10
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— Module Stop 17 (MSTP17): Specifies halting the clock supply to the serial IO with FIFO interface (SIOF). When the MSTP17 bit is set to 1, the clock supply to the serial IO with FIFO interface (SIOF) is halted.

Bit 7: MSTP17	Description	
0	SIOF runs	(Initial value)
1	Clock supply to SIOF halted	

Bit 6— Reserved: This bit is always read as 0. The write value should always as 0.

Bit 5— Module Stop 15 (MSTP15): Specifies halting the clock supply to the AFE interface (AFE IF). When the MSTP15 bit is set to 1, the clock supply to the AFE interface is halted.

Bit 5: MSTP15	Description	
0	AFE interface runs	(Initial value)
1	Clock supply to AFE interface halted	

Bit 4— Module Stop 14 (MSTP14): Specifies halting the clock supply to the USB function module (USBF). When the MSTP14 bit is set to 1, the clock supply to the USBF is halted.

Bit 4: MSTP14	Description	
0	USBF runs	(Initial value)
1	Clock supply to USBF halted	

Bit 3—Module Stop 13 (MSTP13): Specifies halting the clock supply to the USB host controller (USBH). When the MSTP13 bit is set to 1, the clock supply to the USBH is halted.

Bit 3: MSTP13	Description	
0	USBH runs	(Initial value)
1	Clock supply to USBH halted	

Note: This bit should not be set to 1 when MSTP14 (bit 4) is 0.

Bit 2——Reserved: This bit is always read as 0. The write value should always as 0.

Bit 1— Module Stop 11 (MSTP11): Specifies halting the clock supply LCD Controller (LCDC). When the MSTP11 bit is set to 1, the clock supply to the LCDC is halted.

Bit 1: MSTP11	Description	
0	LCDC runs	(Initial value)
1	Clock supply to LCDC halted	

Bit 0— Module Stop 10 (MSTP10): Specifies halting the clock supply to PC Card Controller (PCC). When the MSTP10 bit is set to 1, the clock supply to the PCC is halted.

Bit 0: MSTP10	Description	
0	PCC runs	(Initial value)
1	Clock supply to PCC halted	

9.2.4 Module Software Reset Register (SRSTR)

The Software Reset Register (SRSTR) is an 8-bit readable/writable register that controls module reset operation equivalent to power-on reset. SRSTR is initialized to H'00 by a power-on reset.

Bit:	7	6	5	4	3	2	1	0
	SIOFR	—	AFECR	USBFR	USBHR	LBSCR	LCDCR	PCCR
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7— SIOF Reset (SIOFR): When the SIOF bit is set to 1, the serial I/O (SIOF) is reset. 0 should be written after writing 1.

Bit 7: SIOFR	Description	
0	Not reset SIOF	(Initial value)
1	Resets SIOF	

Bit 6—Reserved: This bit is always read as 0. The write value should always be 0.

Bit 5— AFEIF Reset (AFECR): When the AFEC bit is set to 1, the AFE interface (AFEIF) is reset. 0 should be written after writing 1.

Bit 5: AFECR	Description	
0	Not reset AFEIF	(Initial value)
1	Resets AFEIF	

Bit 4— USBF Reset (USBFR): When the USBF bit is set to 1, the SUB function module (USBF) is reset. 0 should be written after writing 1.

Bit 4: USBFR	Description	
0	Not reset USBF	(Initial value)
1	Resets USBF	

Bit 3— USBH Reset (USBHR): When the USBH bit is set to 1, the USB host controller is reset. 0 should be written after writing 1.

Bit 3: USBHR	Description	
0	Not reset USBH	(Initial value)
1	Resets USBH	

Bit 2— LBSC Reset (LBSCR): When the LBSC bit is set to 1, the Li bus state controller (LBSC) is reset. 0 should be written after writing 1.

Bit 2: LBSCR	Description	
0	Not reset LBSC	(Initial value)
1	Resets LBSC	

Bit 1— LCDC Reset (LCDCR): When the LCDC bit is set to 1, the LCD controller (LCDC) is reset. 0 should be written after writing 1.

Bit 1: LCDCR	Description	
0	Not reset LCDC	(Initial value)
1	Resets LCDC	

Bit 0— PCC Reset (PCCR): When the PCC bit is set to 1, PC card controller (PCC) is reset. 0 should be written after writing 1.

Bit 0: PCCR	Description	
0	Not reset PCC	(Initial value)
1	Resets PCC	

9.3 Sleep Mode

9.3.1 Transition to Sleep Mode

Executing the SLEEP instruction when the STBY bit in STBCR is 0 causes a transition from the program execution state to sleep mode. Although the CPU halts immediately after executing the SLEEP instruction, the contents of its internal registers are retained. The on-chip supporting modules continue to run during sleep mode and the clock continues to be output to the CKIO and CKIO2 pins.

In sleep mode, the STATUS1 pin is set to high and the STATUS0 pin low.

9.3.2 Canceling Sleep Mode

Sleep mode is canceled by an interrupt (NMI, IRQ, IRL, on-chip supporting module interrupt, PINT) or reset. Interrupts are accepted during sleep mode even when the BL bit in the SR register is 1. If necessary, save SPC and SSR in the stack before executing the SLEEP instruction.

Canceling with an Interrupt: When an NMI, IRQ, IRL or on-chip supporting module interrupt occurs, sleep mode is canceled and interrupt exception handling is executed. A code corresponding to the interrupt source is set in the INTEVT and INTEVT2 registers.

Canceling with a Reset: Sleep mode is canceled by a power-on reset or a manual reset.

9.4 Standby Mode

9.4.1 Transition to Standby Mode

To enter standby mode, set the STBY bit to 1 in STBCR, then execute the SLEEP instruction. The chip moves from the program execution state to standby mode. In standby mode, not only the CPU, but the clock and on-chip supporting modules are halted. The clock output from the CKIO and CKIO2 pins also halts.

The contents of the CPU and cache register are held, but some on-chip supporting modules are initialized. Table 9.4 lists the states of registers in standby mode.

Table 9.4 Register States in Standby Mode

Module	Registers Initialized	Registers Retaining Data
Interrupt controller (INTC)	—	All registers
On-chip clock pulse generator (OSC)	—	All registers
User break controller (UBC)	—	All registers
Bus state controller (BSC)	—	All registers
Timer unit (TMU)	TSTR register	Registers other than TSTR
Realtime clock (RTC)	—	All registers
A/D converter (ADC)	All registers	—
D/A converter (DAC)	—	All registers
Li bus state controller (LBSC)	—	All registers
LCD controller (LCDC)	—	All registers
USB host controller (USBH)	—	All registers
USB function module (USBF)	—	All registers
AFE interface (AFEIF)	—	All registers
Serial IO with FIFO (SIOF)	—	All registers
PC card controller (PPC)	—	Registers other than PCC0ISR*

Note: * PCC0ISR reflects the normal status.

The procedure for moving to standby mode is as follows:

1. Clear the TME bit in the WDT's timer control register (WTCSR) to 0 to stop the WDT.
   Set the WDT's timer counter (WTCNT) to 0 and set a value to the CKS2 to CKS0 bits in the WTCSR register to secure the specified oscillation settling time.
2. After the STBY bit in the STBCR register is set to 1, the SLEEP instruction is executed.
3. When the chip enters standby mode and the clocks within the chip are halted, he STATUS1 pin output goes low and the STATUS0 pin output goes high.

9.4.2 Canceling Standby Mode

Standby mode is canceled by an interrupt (NMI, IRQ, IRL, on-chip supporting module interrupt or PINT) or a reset.

Canceling with an Interrupt: The on-chip WDT can be used for hot starts. When the chip detects an NMI, IRL, IRQ, PINT*1, or on-chip supporting module (except the interval timer)*2 interrupt, the clock will be supplied to the entire chip and standby mode canceled after the time set in the WDT's timer control/status register has elapsed. The STATUS1 and STATUS0 pins both go low. Interrupt exception handling then begins and a code corresponding to the interrupt source is set in the INTEVT and INTEVT2 registers. After branching to the interrupt processing routine occurs, clear the STBY bit in the STBCR register. The WTCNT stops automatically. If the STBY bit is not cleared, WTCNT continues operation and transits to the standby mode*3 when it reaches H'80. This function prevents the data from being destroyed due to a rising voltage under an unstable power supply. Interrupts are accepted during standby mode even when the BL bit in the SR register is 1. If necessary, save SPC and SSR in the stack before executing the SLEEP instruction.

Immediately after an interrupt is detected, the phase of the clock output of the CKIO and CKIO2 pin may be unstable, until the standby mode is canceled. The canceling condition is that the interrupt request level (IRQ, IRL, or on-chip supporting module interrupt) is higher than the mask level in the I3 to I0 bits in the SR register.

Notes: 1. When the RTC is being used, standby mode can be canceled using $\overline{\text{IRL3}}$ to $\overline{\text{IRL0}}$, IRQ4 to IRQ0 or PINT0 to PINT5.
2. Standby mode can be canceled with an RTC or TMU (only when running on the RTC clock) interrupt.
3. Use a power-on reset to cancel standby mode.
Operation is not guaranteed in the case of a manual reset or interrupt input.

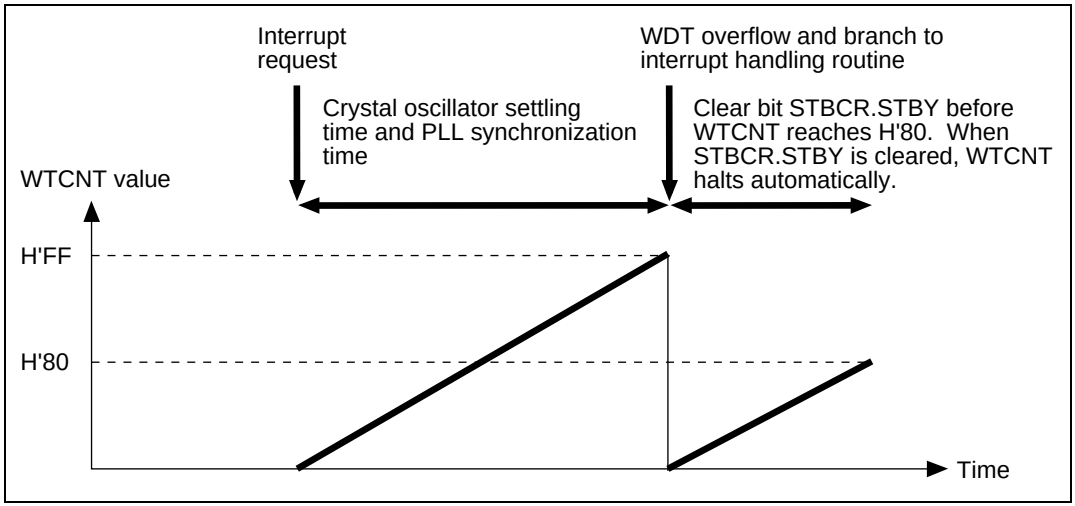


Figure 9.1 Canceling Standby Mode with STBCR.STBY

Canceling with a Reset: Standby mode can be canceled with a reset (power-on or manual).

Keep the $\overline{\text{RESET}}$ or $\overline{\text{RESETM}}$ pin low until the clock oscillation settles.

The internal clock will be output continuously to the CKIO pin.

9.4.3 Clock Pause Function

In standby mode, the clock input from the EXTAL pin or CKIO pin can be halted and the frequency can be changed. This function is used as follows:

1. Enter standby mode using the procedure for changing to standby mode.
2. When the chip enters standby mode and the clock stopped within the chip, the STATUS1 pin output is low and the STATUS0 pin output is high.
3. When the STATUS1 pin goes low and the STATUS0 pin goes high, the input clock is stopped or the frequency is changed.
4. When the frequency is changed, an NMI, IRL, IRQ, PINT or on-chip supporting module (except the internal timer) interrupt is input after changing the frequency. When the clock is stopped, the same interrupts are input after the clock is applied.
5. After the time set in the WDT has elapsed, the clock starts being applied within the chip, the STATUS1 and STATUS0 pins both go low, operation resumes from the interrupt exception handling.

9.5 Module Standby Function

9.5.1 Transition to Module Standby Function

Setting the standby control register STBCR, STBCR2, STBCR3, MSTP17, MSTP15 to MSTP13, MSTP11 to MSTP4, and MSTP 2 to MSTP0 bits to 1 halts the clock supply to the corresponding on-chip supporting modules. By using this function, the power consumption in normal mode and sleep mode can be reduced.

In the module standby function, external pins of the on-chip supporting modules are different depending on the on-chip supporting modules. TMU external pins hold their state prior to the halt. SCI external pins go to the reset state. With a few exceptions, all registers hold their values.

Bit	Value	Description
MSTP17	0	SIOF runs.
	1	Clock supply to SIOF halted.
MSTP15	0	AFEIF runs.
	1	Clock supply to AFEIF halted.
MSTP14	0	USBF runs.
	1	Clock supply to USBF halted.
MSTP13	0	USBH runs.
	1	Clock supply to USBH halted. This bit should not be set to 1 when MSTP14 (bit 4) is 0.
MSTP11	0	LCDC runs.
	1	Clock supply to LCDC halted.
MSTP10	0	PCC runs.
	1	Clock supply to PCC halted.
MSTP9	0	X/Y memory runs.
	1	Clock supply to X/Y memory halted.
MSTP8	0	UBC runs.
	1	Clock supply to UBC halted.
MSTP7	0	DMAC runs.
	1	Clock supply to DMAC halted.
MSTP6	0	DAC runs.
	1	Clock supply to DAC halted.
MSTP5	0	ADC runs.
	1	Clock supply to ADC halted, and all registers initialized.
MSTP4	0	SCIF runs.
	1	Clock supply to SCIF halted.
MSTP2	0	TMU runs.
	1	Clock supply to TMU halted.*1
MSTP1	0	RTC runs.
	1	Clock supply to RTC halted. Register access prohibited.*2
MSTP0	0	SCI runs.
	1	Clock supply to SCI halted.

Notes: 1. The initialized registers are the same as in the standby mode (see table 9.4).
2. The counter runs.

9.5.2 Clearing the Module Standby Function

The module standby function can be cleared by clearing the MSTP17, MSTP15 to MSTP13, MSTP11 to MSTP4, and MSTP2 to MSTP0 bits to 0, or by a power-on reset or manual reset.

9.6 Timing of STATUS Pin Changes

The timing of STATUS1 and STATUS0 pin changes is shown in figures 9.2 to 9.9.

The meanings of STATUS are as follows:

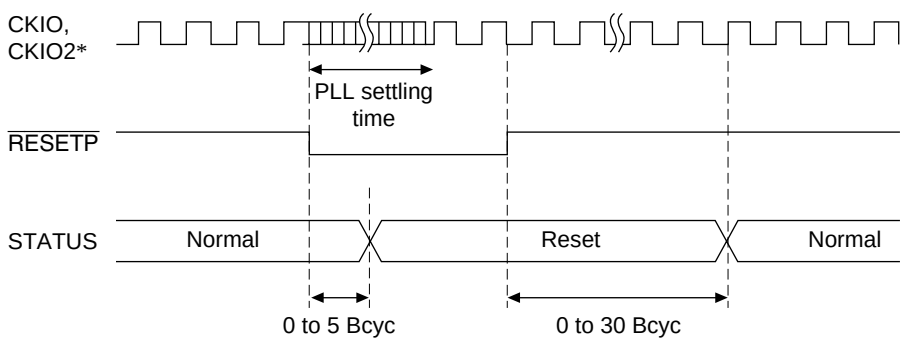
- Reset: HH (STATUS1 is high, STATUS0 is high)
- Sleep: HL (STATUS1 is high, STATUS0 is low)
- Standby: LH (STATUS1 is low, STATUS0 is high)
- Normal: LL (STATUS1 is low, STATUS0 is low)

The meanings of clock units are as follows:

- Bcyc: Bus clock cycle
- Pcyc: Peripheral clock cycle
- Rcyc: 32.768-kHz RTC clock cycle

9.6.1 Timing for Resets

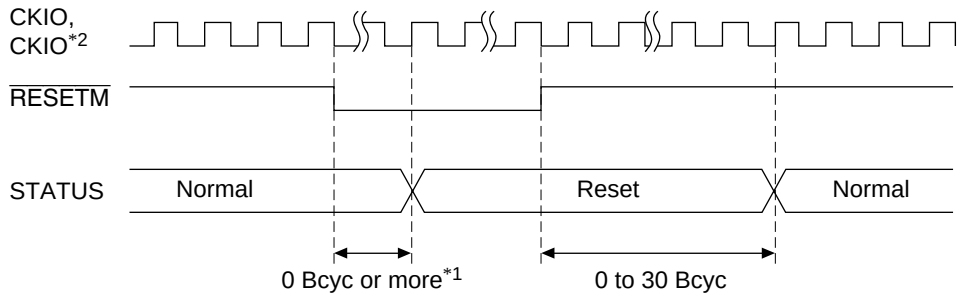
Power-On Reset



Note: * CKIO2 can be used at only clock modes 0, 1 and 2.

Figure 9.2 Power-On Reset (Clock Modes 0, 1, 2, and 7) STATUS Output

Manual Reset

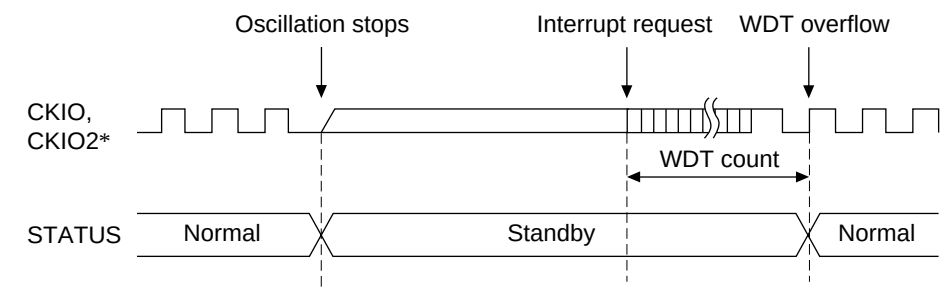


Notes: 1. During manual reset, STATUS becomes HH (reset) and the internal reset begins after waiting for the executing bus cycle to end.
2. CKIO2 can be used at only clock modes 0, 1 and 2.

Figure 9.3 Manual Reset STATUS Output

9.6.2 Timing for Canceling Standbys

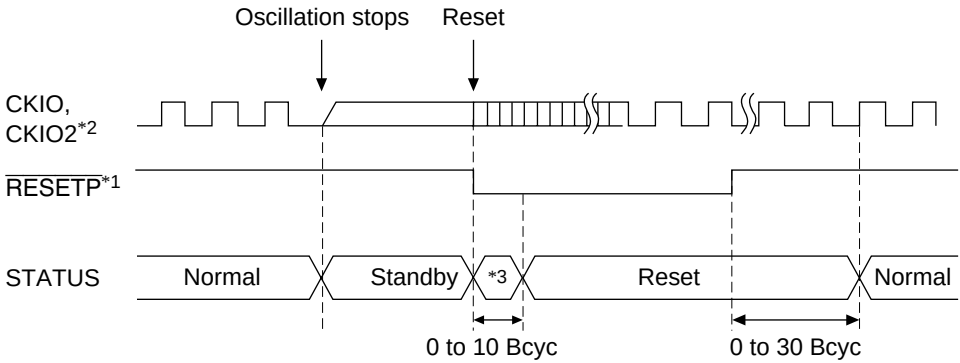
Standby to Interrupt



Note: * CKIO2 can be used at only clock modes 0, 1 and 2.

Figure 9.4 Standby to Interrupt STATUS Output

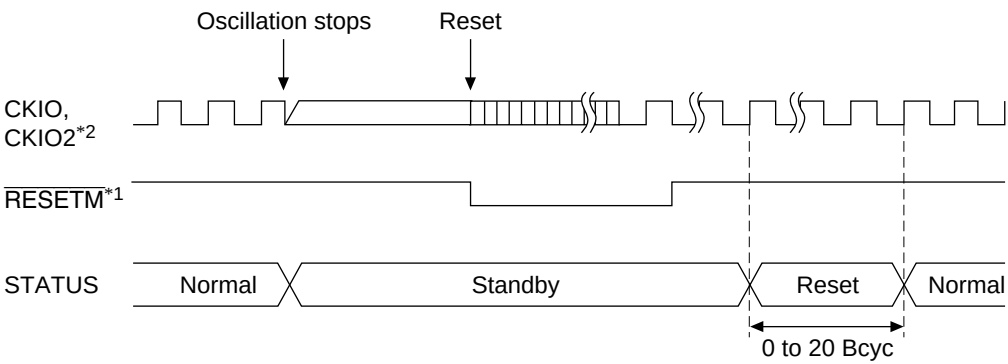
Standby to Power-On Reset



Notes: 1. When standby mode is cleared with a power-on reset, the WDT does not count. Keep $\overline{\text{RESETP}}$ low during PLL's oscillation settling time.
2. CKIO2 can be used at only clock modes 0, 1 and 2.
3. Undefined

Figure 9.5 Standby to Power-On Reset STATUS Output

Standby to Manual Reset

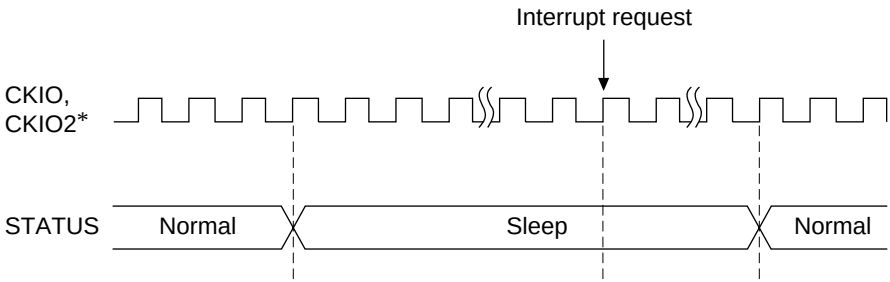


Notes: 1. When standby mode is cleared with a power-on reset, the WDT does not count. Keep $\overline{\text{RESETM}}$ low during PLL's oscillation settling time.
2. CKIO2 can be used at only clock modes 0, 1 and 2.

Figure 9.6 Standby to Manual Reset STATUS Output

9.6.3 Timing for Canceling Sleep Mode

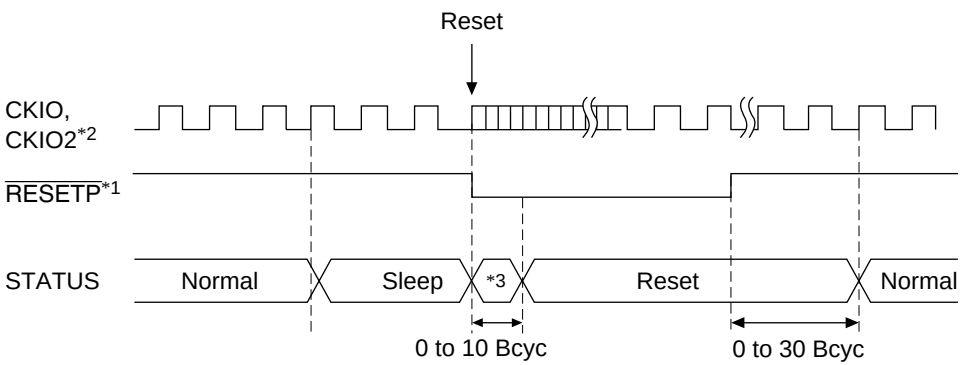
Sleep to Interrupt



Note: * CKIO2 can be used at only clock modes 0, 1 and 2.

Figure 9.7 Sleep to Interrupt STATUS Output

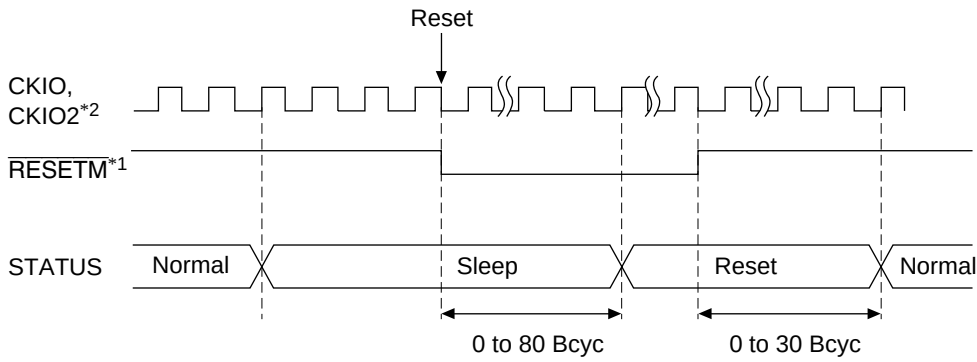
Sleep to Power-On Reset



Notes: 1. When the PLL's multiplication ratio is changed by a power-on reset, keep $\overline{\text{RESETP}}$ low during PLL's oscillation settling time.
2. CKIO2 can be used at only clock modes 0, 1 and 2.
3. Undefined

Figure 9.8 Sleep to Power-On Reset STATUS Output

Sleep to Manual Reset



Notes: 1. Keep $\overline{\text{RESETM}}$ low until the STATUS becomes reset.
2. CKIO2 can be used at only clock modes 0, 1 and 2.

Figure 9.9 Sleep to Manual Reset STATUS Output

9.7 Hardware Standby Mode

9.7.1 Transition to Hardware Standby Mode

To enter hardware standby mode, set the CA pin low. In hardware standby mode, all modules except for any modules that run with RTC clock are halted as well as in standby mode entered by sleep instruction.

Differences between hardware standby mode and standby mode are as follows.

1. Interrupts and manual reset are not accepted in hardware standby mode.
2. The TMU does not run in hardware standby mode.

Operation when the CA pin goes low depends on the CPG status.

1. In standby mode
   The chip enters hardware standby mode, clock remains halted.
   Interrupts and manual reset are not accepted and the TMU halts.

2. During WDT runs when clearing standby mode with an interrupt
   The chip enters hardware standby mode after the CPU resumes operation once standby mode is cleared.

3. In sleep mode
   The chip enters hardware standby mode after the CPU resumes operation once sleep mode is cleared.

Note that CA pin must keep low during hardware standby mode.

9.7.2 Clearing the Hardware Standby Mode

Hardware standby mode can be cleared only by power-on reset.

The clock starts oscillation by setting CA pin high while RESETP pin is low. At this time, keep the $\overline{\text{RESETP}}$ pin low until the clock oscillation settles. Then, the CPU starts power-on reset processing after setting the $\overline{\text{RESETP}}$ pin high.

The operation is not guaranteed when an interrupt or manual reset is occurred.

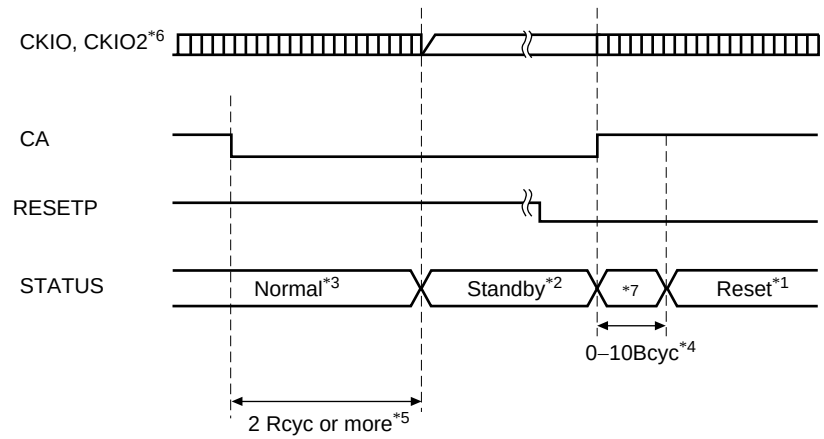
9.7.3 Timing of Hardware Standby Mode

The timings of each pin in hardware standby mode are shown in figures 9.10 and 9.11.

CA pin is sampled by EXTAL2 (32.768 kHz). Hardware standby request is detected when two continuous cycles go low in this clock.

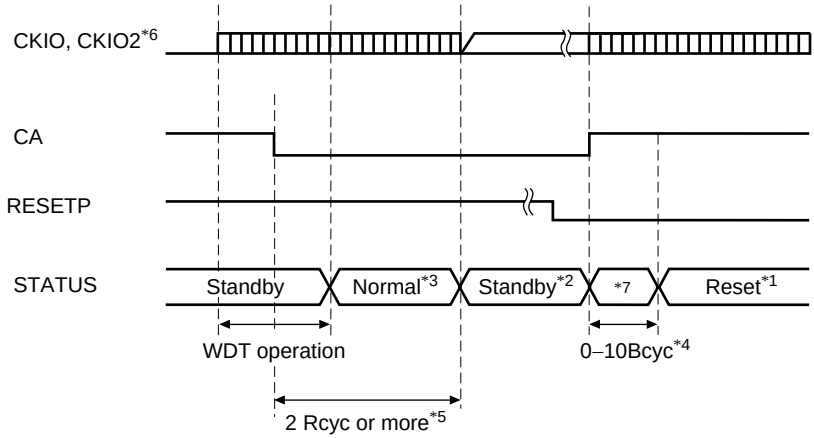
Keep CA pin low during hardware standby mode.

The clock starts oscillation when the CA pin is set high after setting the $\overline{\text{RESETP}}$ pin low.



Notes: 1. Reset: HH (STATUS1 is high, STATUS0 is high)
2. Standby: LH (STATUS1 is low, STATUS0 is high)
3. Normal: LL (STATUS1 is low, STATUS0 is low)
4. Bcyc: Bus clock cycle
5. Rcyc: EXTAL2 (32.768 kHz) clock cycle
6. CKIO2 can be used at only clock modes 0,1 and 2.
7. Undefined

Figure 9.10 Hardware Standby Mode Timing (CA = Low in Normal Operation)



Notes: 1. Reset: HH (STATUS1 is high, STATUS0 is high)
2. Standby: LH (STATUS1 is low, STATUS0 is high)
3. Normal: LL (STATUS1 is low, STATUS0 is low)
4. Bcyc: Bus clock cycle
5. Rcyc: EXTAL2 (32.768 kHz) clock cycle
6. CKIO2 can be used at only clock modes 0,1 and 2.
7. Undefined

Figure 9.11 Hardware Standby Mode Timing
(CA = Low during WDT Operation while Standby Mode is Cleared)

Section 10 On-Chip Oscillation Circuits

10.1 Overview

The on-chip oscillation circuits consist of the clock pulse generator (CPG) and watchdog timer (WDT).

The clock pulse generator (CPG) supplies all clocks to the processor and controls the power-down modes.

The watchdog timer (WDT) is a single-channel timer that counts the clock settling time and is used when clearing standby mode and temporary standby, such as frequency changes. It can also be used as an ordinary watchdog timer or interval timer.

10.1.1 Features

The CPG has the following features:

- Four clock modes: Selection of four clock modes for different frequency ranges, power consumption, direct crystal input, and external clock input.
- Three clocks generated independently: An internal clock for the CPU, cache, and TLB (Iϕ); a peripheral clock (Pϕ) for the on-chip supporting modules; and a bus clock (CKIO) for the external bus interface.
- Frequency change function: Internal and peripheral clock frequencies can be changed independently using the PLL circuit and divider circuit within the CPG. Frequencies are changed by software using frequency control register (FRQCR) settings.
- Power-down mode control: The clock can be stopped for sleep mode and standby mode and specific modules can be stopped using the module standby function.

The WDT has the following features:

- Can be used to ensure the clock settling time: Use the WDT to cancel standby mode and the temporary standby which occur when the clock frequency is changed.
- Can switch between watchdog timer mode and interval timer mode.
- Generates internal resets in watchdog timer mode: Internal resets occur after counter overflow. Selection of power-on reset or manual reset.
- Generates interrupts in interval timer mode: Internal timer interrupts occur after counter overflow.
- Selection of eight counter input clocks. Eight clocks (×1 to ×1/4096) can be obtained by dividing the peripheral clock.

10.2 Overview of the CPG

10.2.1 CPG Block Diagram

A block diagram of the on-chip clock pulse generator is shown in figure 10.1.

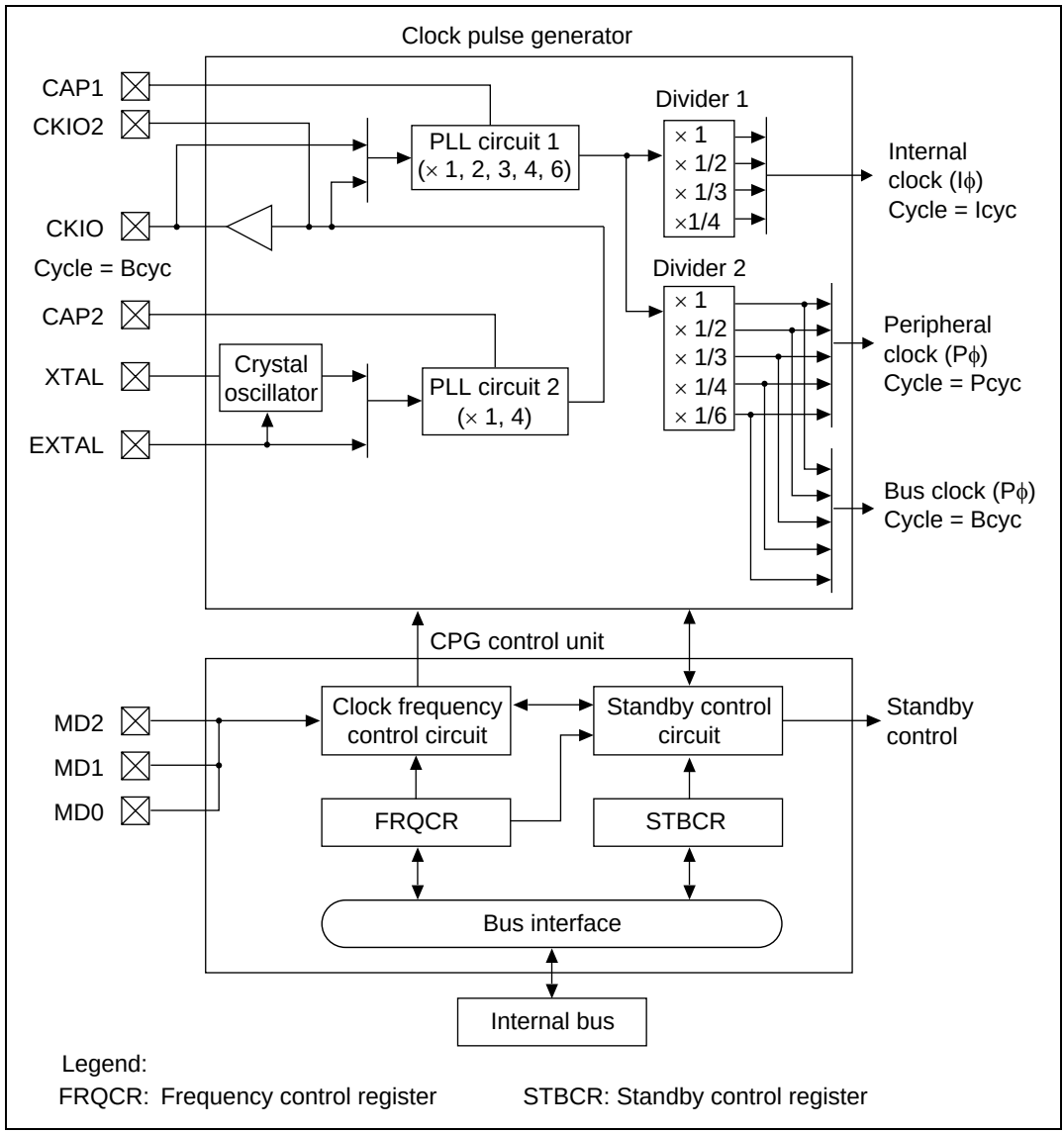


Figure 10.1 Block Diagram of Clock Pulse Generator

The clock pulse generator blocks function as follows:

1. PLL Circuit 1

   PLL circuit 1 doubles, triples, quadruples, sextuples, or leaves unchanged the input clock frequency from the CKIO pin or PLL circuit 2. The multiplication rate is set by the frequency control register. When this is done, the phase of the leading edge of the internal clock is controlled so that it will agree with the phase of the leading edge of the CKIO pin.
2. PLL Circuit 2

   PLL circuit 2 leaves quadruples the frequency of the crystal oscillator or the input clock frequency coming from the EXTAL pin. The multiplication ratio is fixed by the clock operation mode. The clock operation mode is set by pins MD0, MD1, and MD2. See table 10.3 for more information on clock operation modes.
3. Crystal Oscillator

   This oscillator is used when a crystal oscillator element is connected to the XTAL and EXTAL pins. It operates according to the clock operating mode setting.
4. Divider 1

   Divider 1 generates a clock at the operating frequency used by the internal clock. The operating frequency can be 1, 1/2, 1/3, or 1/4 times the output frequency of PLL circuit 1, as long as it stays at or above the clock frequency of the CKIO pin. The division ratio is set in the frequency control register.
5. Divider 2

   Divider 2 generates a clock at the operating frequency used by the peripheral clock. The operating frequencies can be 1, 1/2, 1/3,1/4, or 1/6 times the output frequency of PLL Circuit 1 or the clock frequency of the CKIO pin, as long as it stays at or below the clock frequency of the CKIO pin. The division ratio is set in the frequency control register.
6. Clock Frequency Control Circuit

   The clock frequency control circuit controls the clock frequency using the MD pin and the frequency control register.
7. Standby Control Circuit

   The standby control circuit controls the state of the clock pulse generator and other modules during clock switching and sleep/standby modes.
8. Frequency Control Register

   The frequency control register has control bits assigned for the following functions: clock output/non-output from the CKIO pin, PLL standby, the frequency multiplication ratio of PLL 1, and the frequency division ratio of the internal clock and the peripheral clock.
9. Standby Control Register

   The standby control register has bits for controlling the power-down modes. See section 9, Power-Down Modes and Software Reset, for more information.

10.2.2 CPG Pin Configuration

Table 10.1 lists the CPG pins and their functions.

Table 10.1 Clock Pulse Generator Pins and Functions

Pin Name	Symbol	I/O	Description
Mode control pins	MD0	I	Set the clock operating mode.
	MD1	I	
	MD2	I	
Crystal I/O pins (clock input pins)	XTAL	O	Connects a crystal oscillator.
	EXTAL	I	Connects a crystal oscillator. Also used to input an external clock.
Clock I/O pin	CKIO	I/O	Inputs or outputs an external clock.
Clock Out pin	CKIO2	O	Output external clock. Level can be fixed. Only clock modes 0, 1, 2 can be supported for this pin.
Capacitor connection pins	CAP1	I	Connects capacitor for PLL circuit 1 operation (recommended value 470 pF).
For PLL	CAP2	I	Connects capacitor for PLL circuit 2 operation (recommended value 470 pF).

10.2.3 CPG Register Configuration

Table 10.2 shows the CPG register configuration.

Table 10.2 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Frequency control register	FRQCR	R/W	H'0102	H'FFFFFF80	16 bits
CKIO2 Control Register 2	CKIO2CR	R/W	H'0000	H'0400023A (H'A400023A)*	16 bits

Note: * When address translation by the MMU does not apply, the address in parentheses should be used.

10.3 Clock Operating Modes

Table 10.3 shows the relationship between the mode control pin (MD2 to MD0) combinations and the clock operating modes. Table 10.4 shows the usable frequency ranges in the clock operating modes.

Table 10.3 Clock Operating Modes

	Pin Values			Clock I/O		PLL2	PLL1	Divider 1	Divider 2	CKIO
Mode	MD2	MD1	MD0	Source	Output	On/Off	On/Off	Input	Input	Frequency
0	0	0	0	EXTAL	CKIO	On, multi-plication ratio: 1	On	PLL1 output	PLL1	(EXTAL)
1	0	0	1	EXTAL	CKIO	On, multi-plication ratio: 4	On	PLL1 output	PLL1	(EXTAL) × 4
2	0	1	0	Crystal oscillator	CKIO	On, multi-plication ratio: 4	On	PLL1 output	PLL1	(Crystal) × 4
7	1	1	1	CKIO	—	Off	On	PLL1 output	PLL1	(CKIO)

Mode 0: An external clock is input from the EXTAL pin and undergoes waveform shaping by PLL circuit 2 before being supplied inside this LSI. PLL circuit 1 is constantly on. An input clock frequency of 24 MHz to the maximum frequency of CKIO can be used. For details on the CKIO maximum frequency, see section 32, Electrical Characteristics.

Mode 1: An external clock is input from the EXTAL pin and its frequency is multiplied by 4 by PLL circuit 2 before being supplied inside this LSI, allowing a low-frequency external clock to be used. An input clock frequency of 6 MHz to1/4 of the maximum frequency of CKIO can be used. For details on the CKIO maximum frequency, see section 32, Electrical Characteristics.

Mode 2: The on-chip crystal oscillator operates, with the oscillation frequency being multiplied by 4 by PLL circuit 2 before being supplied inside this LSI, allowing a low crystal frequency to be used. A crystal oscillation frequency of 6 MHz to 1/4 of the maximum frequency of CKIO can be used. For details on the CKIO maximum frequency, see section 32, Electrical Characteristics.

Mode 7: In this mode, the CKIO pin is an input, an external clock is input to this pin, and undergoes waveform shaping, and also frequency multiplication according to the setting, by PLL circuit 1 before being supplied to this LSI. In modes 0 to 2, the system clock is generated from the output of this LSI's CKIO pin. Consequently, if a large number of ICs are operating on the clock cycle, the CKIO pin load will be large. This mode, however, assumes a comparatively large-scale system. If a large number of ICs are operating on the clock cycle, a clock generator with a number of low-skew clock outputs can be provided, so that the ICs can operate synchronously by distributing the clocks to each one.

As PLL circuit 1 compensates for fluctuations in the CKIO pin load, this mode is suitable for connection of synchronous DRAM.

Table 10.4 Available Combination of Clock Mode and FRQCR Values

Clock Mode	FRQCR	PLL1	PLL2	Clock Rate* (I:B:P)
0	H'0100	ON (× 1)	ON (× 1)	1:1:1
	H'0101	ON (× 1)	ON (× 1)	1:1:1/2
	H'0102	ON (× 1)	ON (× 1)	1:1:1/4
	H'0111	ON (× 2)	ON (× 1)	2:1:1
	H'0112	ON (× 2)	ON (× 1)	2:1:1/2
	H'0115	ON (× 2)	ON (× 1)	1:1:1
	H'0116	ON (× 2)	ON (× 1)	1:1:1/2
	H'0122	ON (× 4)	ON (× 1)	4:1:1
	H'0126	ON (× 4)	ON (× 1)	2:1:1
	H'012A	ON (× 4)	ON (× 1)	1:1:1
	H'A100	ON (× 3)	ON (× 1)	3:1:1
	H'A101	ON (× 3)	ON (× 1)	3:1:1/2
	H'E100	ON (× 3)	ON (× 1)	1:1:1
	H'E101	ON (× 3)	ON (× 1)	1:1:1/2
	H'A111	ON (× 6)	ON (× 1)	6:1:1
1, 2	H'0100	ON (× 1)	ON (× 4)	4:4:4
	H'0101	ON (× 1)	ON (× 4)	4:4:2
	H'0102	ON (× 1)	ON (× 4)	4:4:1
	H'0111	ON (× 2)	ON (× 4)	8:4:4
	H'0112	ON (× 2)	ON (× 4)	8:4:2
	H'0115	ON (× 2)	ON (× 4)	4:4:4
	H'0116	ON (× 2)	ON (× 4)	4:4:2
	H'0122	ON (× 4)	ON (× 4)	16:4:4
	H'0126	ON (× 4)	ON (× 4)	8:4:4
	H'012A	ON (× 4)	ON (× 4)	4:4:4
	H'A100	ON (× 3)	ON (× 4)	12:4:4
	H'A101	ON (× 3)	ON (× 4)	12:4:2
	H'E100	ON (× 3)	ON (× 4)	4:4:4
	H'E101	ON (× 3)	ON (× 4)	4:4:2
	H'A111	ON (× 6)	ON (× 1)	24:4:4

Clock Mode	FRQCR	PLL1	PLL2	Clock Rate* (I:B:P)
7	H'0100	ON (× 1)	OFF	1:1:1
	H'0101	ON (× 1)	OFF	1:1:1/2
	H'0102	ON (× 1)	OFF	1:1:1/4
	H'0111	ON (× 2)	OFF	2:1:1
	H'0112	ON (× 2)	OFF	2:1:1/2
	H'0115	ON (× 2)	OFF	1:1:1
	H'0116	ON (× 2)	OFF	1:1:1/2
	H'0122	ON (× 4)	OFF	4:1:1
	H'0126	ON (× 4)	OFF	2:1:1
	H'012A	ON (× 4)	OFF	1:1:1
	H'A100	ON (× 3)	OFF	3:1:1
	H'A101	ON (× 3)	OFF	3:1:1/2
	H'E100	ON (× 3)	OFF	1:1:1
	H'E101	ON (× 3)	OFF	1:1:1/2
	H'A111	ON (× 6)	OFF	6:1:1

Note: * Taking input clock as 1

Cautions:

1. The frequency ranges of the input clock and crystal oscillator should be set within the specified frequency range based on the clock rate in table 10.4, and section 32.3, AC Characteristics.

2. The input to divider 1 becomes the output of PLL circuit 1 when PLL circuit 1 is on.

3. The input of divider 2 becomes the output of:
   - PLL circuit 1

4. The frequency of the internal clock (Iϕ) becomes:
   - The product of the frequency of the CKIO pin, the frequency multiplication ratio of PLL circuit 1, and the division ratio of divider 1 when PLL circuit 1 is on.
   - Do not set the internal clock frequency lower than the CKIO pin frequency.
   - Depending on the product, the clock ratio should be set to produce a frequency within one of the ranges indicated below.

     100 MHz products: 24 MHz to 100 MHz

     160 MHz products: 24 MHz to 160 MHz

5. Bus clock (Bϕ) frequency:
   - Depending on the product, the clock ratio should be set to produce a frequency within one of the ranges indicated below.

     100 MHz products: 24 MHz to 50 MHz

     160 MHz products: 24 MHz to 66.64 MHz

6. The frequency of the peripheral clock (Pϕ) becomes:
   - The product of the frequency of the CKIO pin, the frequency multiplication ratio of PLL circuit 1, and the division ratio of divider 2.
   - For all products, the peripheral clock frequency (Pϕ) should be set within the frequency range 6 MHz to 33.34 MHz and no higher than the frequency of the CKIO pin.
   - The peripheral clock frequency (Pϕ) should be set to 13 MHz or higher if the USB function module is used.

7. The output frequency of PLL circuit 1 is the product of the CKIO frequency and the multiplication ratio of PLL circuit 1.

8. ×1, ×2, ×3, ×4, or ×6 can be used as the multiplication ratio of PLL circuit 1. ×1, ×1/2, ×1/3, and ×1/4 can be selected as the division ratio of divider 1. ×1, ×1/2, ×1/3, ×1/4, and ×1/6 can be selected as the division ratio of divider 2. Set the rate in the frequency control register. The on/off state of PLL circuit 2 is determined by the mode.

10.4 Register Descriptions

10.4.1 Frequency Control Register (FRQCR)

The frequency control register (FRQCR) is a 16-bit read/write register that specify the frequency multiplication ratio of PLL circuit 1, and the frequency division ratio of the internal clock and the peripheral clock.

Only word access can be used on the FRQCR register. FRQCR is initialized to H'0102 by a power-on reset, but retains its value in a manual reset and in standby mode.

Bit:	15	14	13	12	11	10	9	8
	STC2	IFC2	PFC2	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	1
R/W:	R/W	R/W	R/W	R	R	R	R	R

Bit:	7	6	5	4	3	2	1	0
	—	—	STC1	STC0	IFC1	IFC0	PFC1	PFC0
Initial value:	0	0	0	0	0	0	1	0
R/W:	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bits 15, 5 and 4—Frequency Multiplication Ratio (STC2, STC1, STC0): These bits specify the frequency multiplication ratio of PLL circuit 1.

Bit 15: STC2	Bit 5: STC1	Bit 4: STC0	Description
0	0	0	× 1 (Initial value)
0	0	1	× 2
1	0	0	× 3
0	1	0	× 4
1	0	1	× 6
Values other than above			Reserved (illegal setting)

Note: Do not set the output frequency of PLL circuit 1 higher than the maximum frequency of the CPU specified in AC Characteristics.

Bits 14, 3 and 2—Internal Clock Frequency Division Ratio (IFC2, IFC1, IFC0): These bits specify the frequency division ratio of the internal clock with respect to the output frequency of PLL circuit 1.

Bit 14: IFC2	Bit 3: IFC1	Bit 2: IFC0	Description	
0	0	0	× 1	(Initial value)
0	0	1	× 1/2	
1	0	0	× 1/3	
0	1	0	× 1/4	
Values other than above			Reserved (illegal setting)	

Note: Do not set the internal clock frequency lower than the CKIO frequency.

Bits 13, 1 and 0—Peripheral Clock Frequency Division Ratio (PFC2, PFC1, PFC0): These bits specify the division ratio of the peripheral clock frequency with respect to the frequency of the output frequency of PLL circuit 1 or the frequency of the CKIO pin.

Bit 13: PFC2	Bit 1: PFC1	Bit 0: PFC0	Description	
0	0	0	× 1	
0	0	1	× 1/2	
1	0	0	× 1/3	
0	1	0	× 1/4	
1	0	1	× 1/6	(Initial value)
Values other than above			Reserved (illegal setting)	

Note: Do not set the peripheral clock frequency higher than the frequency of the CKIO pin.

Bits 12 to 9, 7 and 6—Reserved: These bits are always read as 0. The write value should always be 0.

Bit 8—Reserved: This bit is always read as 1. The write value should always be 1.

10.4.2 CKIO2 Control Register (CKIO2CR)

CKIO2CR controls CKIO2 pin output.

Upper 8 Bits:	15	14	13	12	11	10	9	8
	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Lower 8 Bits:	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	CKIO2EN
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W

Bits 15 to 1—Reserved: These bits always read 0. The write value should always be 0.

Bit 0—CKIO2 (CKIO2EN): Selects output or not output (Hi-Z) for CKIO2 clock.

Bit 0: CKIO2EN	Description	
0	Output	(Initial value)
1	Not output (Hi-Z)	

10.5 Changing the Frequency

The frequency of the internal clock and peripheral clock can be changed either by changing the multiplication rate of PLL circuit 1 or by changing the division rates of dividers 1 and 2. All of these are controlled by software through the frequency control register. The methods are described below.

10.5.1 Changing the Multiplication Rate

A PLL settling time is required when the multiplication rate of PLL circuit 1 is changed. The on-chip WDT counts the settling time.

1. In the initial state, the multiplication rate of PLL circuit 1 is 1.
2. Set a value that will become the specified oscillation settling time in the WDT and stop the WDT. The following must be set:
   WTCSR register TME bit = 0: WDT stops
   WTCSR register CKS2 to CKS0 bits: Division ratio of WDT count clock
   WTCNT counter: Initial counter value
3. Set the desired value in the STC2, STC1 and STC0 bits. The division ratio can also be set in the IFC2 to IFC0 bits and PFC2 to PFC0 bits.
4. The processor pauses internally and the WDT starts incrementing. In clock modes 0 to 2 and 7, the internal and peripheral clocks both stop.
5. Supply of the clock that has been set begins at WDT count overflow, and the processor begins operating again. The WDT stops after it overflows.

10.5.2 Changing the Division Ratio

The WDT will not count unless the multiplication rate is changed simultaneously.

1. In the initial state, IFC2 to IFC0 = 000 and PFC2 to PFC0 = 010.
2. Set the IFC2, IFC1, IFC0, PFC2, PFC1, and PFC0 bits to the new division ratio. The values that can be set are limited by the clock mode and the multiplication rate of PLL circuit 1. Note that if the wrong value is set, the processor will malfunction.
3. The clock is immediately supplied at the new division ratio.

10.6 Overview of the WDT

10.6.1 Block Diagram of the WDT

Figure 10.2 shows a block diagram of the WDT.

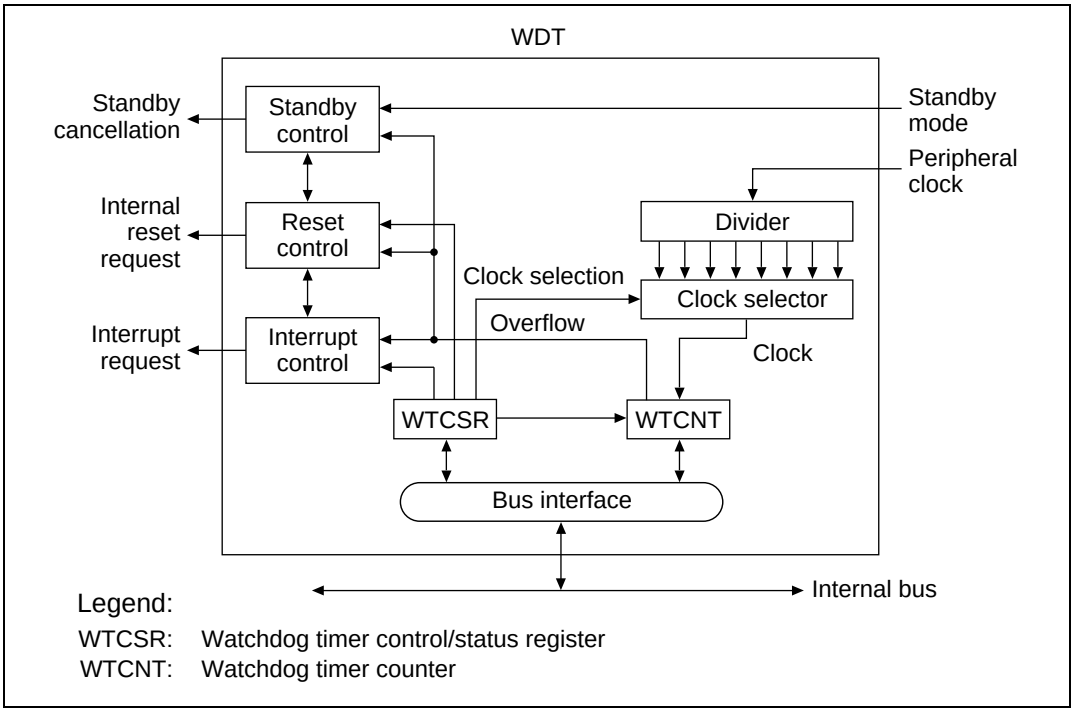


Figure 10.2 Block Diagram of the WDT

10.6.2 Register Configurations

The WDT has two registers that select the clock, switch the timer mode, and perform other functions. Table 10.5 shows the WDT register.

Table 10.5 Register Configuration

Name	Abbreviation	R/W	Initial Value	Address	Access Size
Watchdog timer counter	WTCNT	R/W*	H'00	H'FFFFFF84	R: 8; W: 16*
Watchdog timer control/ status register	WTCSR	R/W*	H'00	H'FFFFFF86	R: 8; W: 16*

Note: * Write with a word access. Write H'5A and H'A5, respectively, in the upper bytes. Byte or longword writes are not possible. Read with a byte access.

10.7 WDT Registers

10.7.1 Watchdog Timer Counter (WTCNT)

The watchdog timer counter (WTCNT) is an 8-bit read/write counter that increments on the selected clock. The WTCNT differs from other registers in that it is more difficult to write to. See section 10.7.3, Notes on Register Access, for details. When an overflow occurs, it generates a reset in watchdog timer mode and an interrupt in interval time mode. Its address is H'FFFFFF84. The WTCNT counter is initialized to H'00 by a power-on reset through the $\overline{\text{RESETP}}$ pin. Use a word access to write to the WTCNT counter, with H'5A in the upper byte. Use a byte access to read WTCNT.

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

10.7.2 Watchdog Timer Control/Status Register (WTCSR)

The watchdog timer control/status register (WTCSR) is an 8-bit read/write register composed of bits to select the clock used for the count, bits to select the timer mode, and overflow flags. The WTCSR differs from other registers in that it is more difficult to write to. See section 10.7.3, Notes on Register Access, for details. Its address is H'FFFFFF86. The WTCSR register is initialized to H'00 only by a power-on reset through the $\overline{\text{RESETP}}$ pin. When a WDT overflow causes an internal reset, the WTCSR retains its value. When used to count the clock settling time for canceling a standby, it retains its value after counter overflow. Use a word access to write to the WTCSR counter, with H'A5 in the upper byte. Use a byte access to read WTCSR.

Bit:	7	6	5	4	3	2	1	0
	TME	WT/$\overline{\text{IT}}$	RSTS	WOVF	IOVF	CKS2	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—Timer Enable (TME): Starts and stops timer operation. Clear this bit to 0 when using the WDT in standby mode or when changing the clock frequency.

Bit 7: TME	Description	
0	Timer disabled: Count-up stops and WTCNT value is retained	(Initial value)
1	Timer enabled	

Bit 6—Timer Mode Select (WT/$\overline{\text{IT}}$): Selects whether to use the WDT as a watchdog timer or an interval timer.

Bit 6: WT/$\overline{\text{IT}}$	Description	
0	Use as interval timer	(Initial value)
1	Use as watchdog timer	

Note: If WT/$\overline{\text{IT}}$ is modified when the WDT is running, the up-count may not be performed correctly.

Bit 5—Reset Select (RSTS): Selects the type of reset when the WTCNT overflows in watchdog timer mode. In interval timer mode, this setting is ignored.

Bit 5: RSTS	Description	
0	Power-on reset	(Initial value)
1	Manual reset	

Bit 4—Watchdog Timer Overflow (WOVF): Indicates that the WTCNT has overflowed in watchdog timer mode. This bit is not set in interval timer mode.

Bit 4: WOVF	Description	
0	No overflow	(Initial value)
1	WTCNT has overflowed in watchdog timer mode	

Bit 3—Interval Timer Overflow (IOVF): Indicates that the WTCNT has overflowed in interval timer mode. This bit is not set in watchdog timer mode.

Bit 3: IOVF	Description	
0	No overflow	(Initial value)
1	WTCNT has overflowed in interval timer mode	

Bits 2 to 0—Clock Select 2 to 0 (CKS2 to CKS0): These bits select the clock to be used for the WTCNT count from the eight types obtainable by dividing the peripheral clock. The overflow period in the table is the value when the peripheral clock (Pϕ) is 15 MHz.

Bit 2: CKS2	Bit 1: CKS1	Bit 0: CKS0	Clock Division Ratio		Overflow Period (when Pϕ = 15 MHz)
0	0	0	1	(Initial value)	17 µs
		1	1/4		68 µs
	1	0	1/16		273 µs
		1	1/32		546 µs
1	0	0	1/64		1.09 ms
		1	1/256		4.36 ms
	1	0	1/1024		17.48 ms
		1	1/4096		69.91 ms

Note: If bits CKS2 to CKS0 are modified when the WDT is running, the up-count may not be performed correctly. Ensure that these bits are modified only when the WDT is not running.

10.7.3 Notes on Register Access

The watchdog timer counter (WTCNT) and watchdog timer control/status register (WTCSR) are more difficult to write to than other registers. The procedure for writing to these registers is given below.

Writing to WTCNT and WTCSR: These registers must be written by a word transfer instruction. They cannot be written by a byte or longword transfer instruction. When writing to WTCNT, set the upper byte to H'5A and transfer the lower byte as the write data, as shown in figure 10.3. When writing to WTCSR, set the upper byte to H'A5 and transfer the lower byte as the write data. This transfer procedure writes the lower byte data to WTCNT or WTCSR.

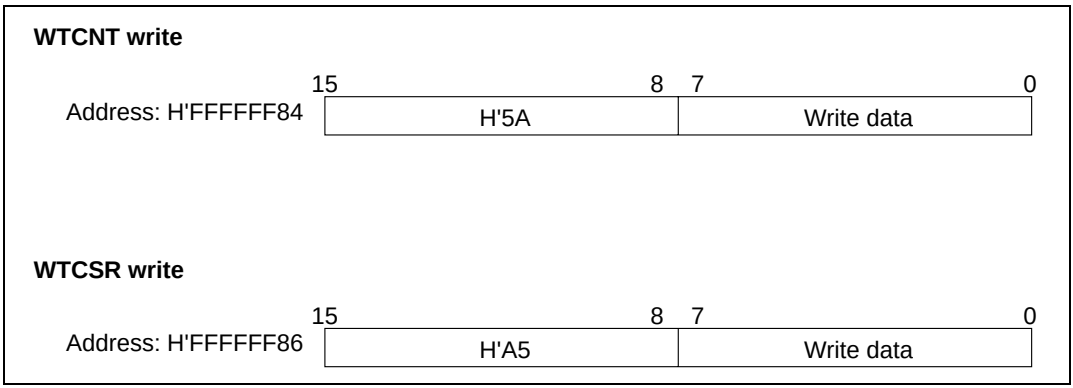


Figure 10.3 Writing to WTCNT and WTCSR

10.8 Using the WDT

10.8.1 Canceling Standby Mode

The WDT can be used to cancel standby mode with an NMI or other interrupts. The procedure is described below. (The WDT does not run when resets are used for canceling, so keep the $\overline{\text{RESETP}}$ pin low until the clock stabilizes.)

1. Before transitioning to standby mode, always clear the TME bit in WTCSR to 0. When the TME bit is 1, an erroneous reset or interval timer interrupt may be generated when the count overflows.
2. Set the type of count clock used in the CKS2 to CKS0 bits in WTCSR and the initial values for the counter in the WTCNT counter. These values should ensure that the time till count overflow is longer than the clock oscillation settling time.

3. Move to standby mode by executing a SLEEP instruction to stop the clock.
4. The WDT starts counting by detecting the edge change of the NMI signal or detecting interrupts.
5. When the WDT count overflows, the CPG starts supplying the clock and the processor resumes operation. The WOVF flag in WTCSR is not set when this happens.
6. Since the WDT continues counting from H'00, set the STBY bit in the STBCR register to 0 in the interrupt processing program and this will stop the WDT. When the STBY bit remains 1, the SH7727 again enters the standby mode when the WDT has counted up to H'80. This standby mode can be canceled by power-on resets.

10.8.2 Changing the Frequency

To change the frequency used by the PLL, use the WDT. When changing the frequency only by switching the divider, do not use the WDT.

1. Before changing the frequency, always clear the TME bit in WTCSR to 0. When the TME bit is 1, an erroneous reset or interval timer interrupt may be generated when the count overflows.
2. Set the type of count clock used in the CKS2 to CKS0 bits of WTCSR and the initial values for the counter in the WTCNT counter. These values should ensure that the time till count overflow is longer than the clock oscillation settling time.
3. When the frequency control register (FRQCR) is written, the clock stops and the processor enters standby mode temporarily. The WDT starts counting.
4. When the WDT count overflows, the CPG resumes supplying the clock and the processor resumes operation. The WOVF flag in WTCSR is not set when this happens.
5. The counter stops at the values H'00 to H'01. The stop value depends on the clock ratio.

10.8.3 Using Watchdog Timer Mode

1. Set the WT/$\overline{\text{IT}}$ bit in the WTCSR register to 1, set the reset type in the RSTS bit, set the type of count clock in the CKS2 to CKS0 bits, and set the initial value of the counter in the WTCNT counter.
2. Set the TME bit in WTCSR to 1 to start the count in watchdog timer mode.
3. While operating in watchdog timer mode, rewrite the counter periodically to H'00 to prevent the counter from overflowing.
4. When the counter overflows, the WDT sets the WOVF flag in WTCSR to 1 and generates the type of reset specified by the RSTS bit. The counter then resumes counting.

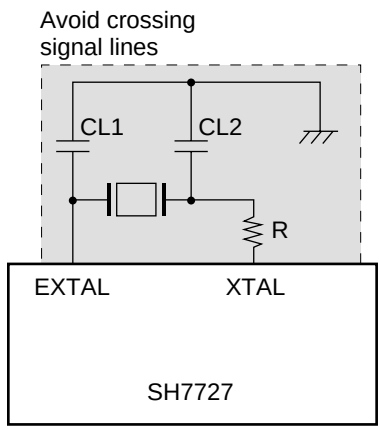
10.8.4 Using Interval Timer Mode

When operating in interval timer mode, interval timer interrupts are generated at every overflow of the counter. This enables interrupts to be generated at set periods.

1. Clear the WT/$\overline{IT}$ bit in the WTCSR register to 0, set the type of count clock in the CKS2 to CKS0 bits, and set the initial value of the counter in the WTCNT counter.
2. Set the TME bit in WTCSR to 1 to start the count in interval timer mode.
3. When the counter overflows, the WDT sets the IOVF flag in WTCSR to 1 and an interval timer interrupt request is sent to INTC. The counter then resumes counting.

10.9 Notes on Board Design

When Using an External Crystal Resonator: Place the crystal resonator, capacitors CL1 and CL2, and damping resistor R close to the EXTAL and XTAL pins. To prevent induction from interfering with correct oscillation, use a common grounding point for the capacitors connected to the resonator, and do not locate a wiring pattern near these components.



Note: The values for CL1, CL2, and the damping resistance should be determined after consultation with the crystal oscillator manufacturer.

Figure 10.4 Points for Attention when Using Crystal Resonator

Decoupling Capacitors: Insert a laminated ceramic capacitor of 0.01 to 0.1 μF as a passive capacitor for each V_{SS}/V_{CC} pair. Mount the passive capacitors to the SH3 power supply pins, and use components with a frequency characteristic suitable for the SH3 operating frequency, as well as a suitable capacitance value.

Digital system V_{SS}/V_{CC} pairs: 35-37, 91-93, 137-139, 155-157, 177-178, 200-202

Digital system V_{SS} Q/V_{CC} Q pairs: 18-20, 29-31, 42-44, 53-55, 64-66, 75-77, 86-88, 100-102, 115-117, 132-134, 159-161, 188-190, 207-209

On-chip oscillator V_{SS}/V_{CC} pairs: 1-4

When Using a PLL Oscillator Circuit: Keep the wiring from the PLL V_{CC} and V_{SS} connection pattern to the power supply pins short, and make the pattern width large, to minimize the inductance component. Ground the oscillation stabilization capacitors C1 and C2 to V_{SS} (PLL1) and V_{SS} (PLL2), respectively. Place C1 and C2 close to the CAP1 and CAP2 pins and do not locate a wiring pattern in the vicinity. In clock mode 7, connect the EXTAL pin to V_{CC} or V_{SS} and leave the XTAL pin open.

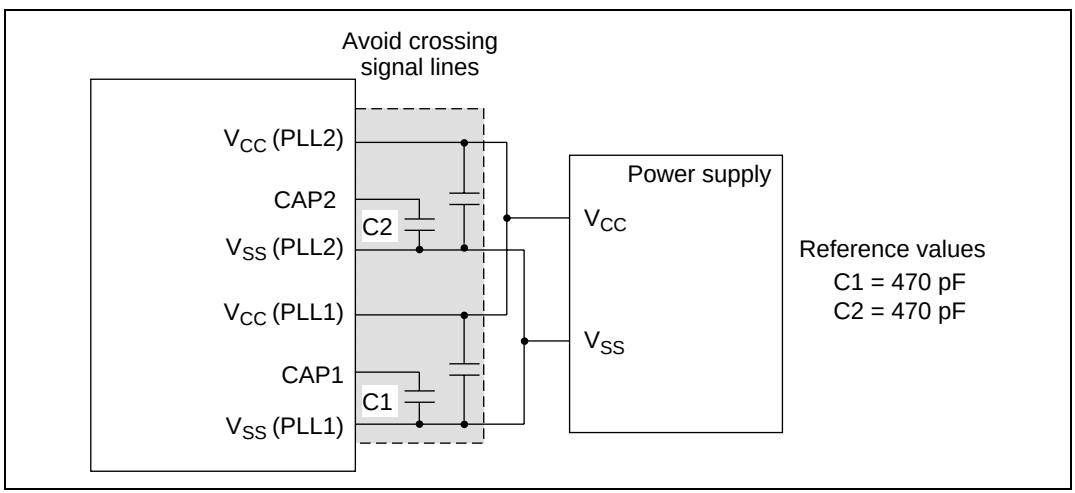


Figure 10.5 Points for Attention when Using PLL Oscillator Circuit

Notes on using pins CKIO and CKIO2 as the clock outputs: Perform board design so that the sum of pin capacitances of the CPU and socket that are connected to pins are 50 pF or less.

Section 11 Extend Clock Pulse Generator for USB (EXCPG)

11.1 Overview

11.1.1 EXCPG

The SH7727 has an on-chip USB interface (USB) which requires a fixed 48-MHz clock source.

The extend clock pulse generator (EXCPG) generates a divided clock from the internal clock (Iϕ), the bus clock (Bϕ), or the external clock (UCLK).

Because the clock sources, which can be a candidate to be used by EXCPG, vary from CPG setting or external clock source, user of SH7727 must adjust the divided clock, carefully to be 48 MHz.

11.2 Functions

11.2.1 Block Diagram

Figure 11.1 shows a block diagram of the EXCPG.

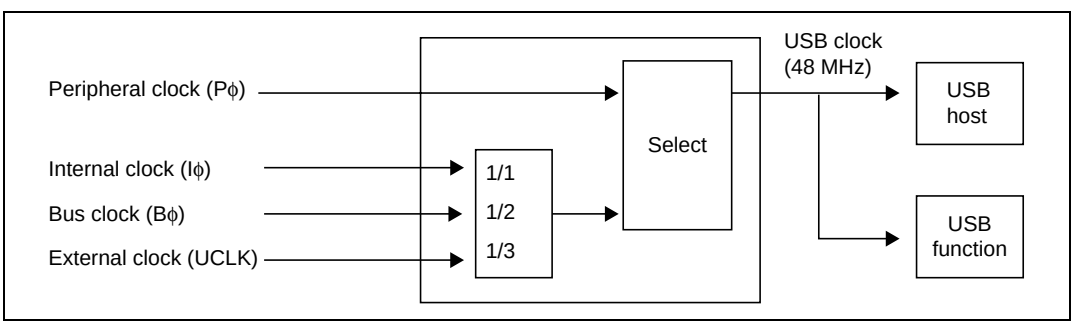


Figure 11.1 Block Diagram of EXCPG

11.2.2 Pin Configuration

Table 11.1 shows a pin configuration of the EXCPG.

Table 11.1 Pin Configuration

Pin Name	Abbreviation	I/O	Description
External clock pin	UCLK	Input	USB clock input pin (48-MHz input)

Note: UCLK is multiplexed with PTD6.

11.2.3 Register Configuration

The EXCPG has the internal registers shown in table 11.2.

Table 11.2 Register Configuration

Name	Abbreviation	R/W	Initial Value	Address	Access Size
EXCPG control register	EXCPGCR	W	H'00	H'A4000236	8

11.3 Register Descriptions

11.3.1 EXCPG Control Register (EXCPGCR)

The EXCPG control register (EXCPGCR) selects the source clock and division ratio for generation of the EXCPG clock.

EXCPGR is initialized to H'00 by a power-on reset.

Bit:	7	6	5	4	3	2	1	0
	—	—	USBCKSEL2	USBCKSEL1	USBCKSEL0	USBDIVSEL2	USBDIVSEL1	USBDIVSEL0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	W	W	W	W	W	W

Bits 7 and 6—Reserved: These bits are always read as 0. The write value should always be 0.

Bits 5 to 3—Clock Select (USBCKSEL2 to USBCKSEL0): Selects the clock source. Although initialized as peripheral clock (Pϕ) after power on reset, the value of USBCKSEL must be changed to adequate value to generate 48 MHz. To prevent malfunction, the USB Host and USB Function must be set in module standby state or module reset state when the value of USBCKSEL is changed.

Bits 5 to 3	Function (Clock Selection)	
000	Peripheral Clock (Pϕ)	(Initial value)
100	Internal Clock (Iϕ)	
101	Bus Clock (Bϕ)	
110	External clock (UCLK)	
Another value	Reserved (setting prohibited)	

Bits 2 to 0—Divider Select (USBDIVSEL2 to USBDIVSEL0): Selects the dividing ratio of clock source to generate USB clock so that the USB clock is 48 MHz.

Bits 2 to 0	Function (Dividing Ratio Selection)	
000	1/1	(Initial value)
001	1/2	
010	1/3	
1**	Internal clock (Iϕ), bus clock (Bϕ), external clock (UCLK) halted	

Note: To reduce power consumption, set USBDIVSEL2 to 1 and halts internal clock (Iϕ), bus clock (Bϕ), or external clock (UCLK) input.

11.4 Usage Notes

By selecting LCLK (LCD clock)/UCLK (USB clock) as the function of the LCLK/UCLK/PTD[6] pin, it is possible to supply the clock input to the pin to both the LCD controller and the USB function controller.

However, in this case it is necessary, using the divider select bit (USBDIVSEL[2:0]) in EXCPGCR (EXCPG control register), to set the USB clock so that the final clock frequency is 48 MHz. This means that the input clock frequency will be 48 MHz. If this frequency is not suitable as the operating clock for the LCD controller, consider selecting an internal clock for LCLK. In addition, it may be impossible to maintain the accuracy of the USB standard clock because the CPU clock (Iϕ) and bus clock (Bϕ) are generated by the internal PLL of the SH7727 by frequency multiplication. Therefore, it is recommended that a dedicated 48 MHz external clock be input to UCLK to ensure the accuracy of the USB standard clock.

Section 12 Bus State Controller (BSC)

12.1 Overview

The bus state controller (BSC) divides physical address space and output control signals for various types of memory and bus interface specifications. BSC functions enable this LSI to link directly with synchronous DRAM, SRAM, ROM, and other memory storage devices without an external circuit. The BSC also allows direct connection to PCMCIA interfaces, simplifying system design and allowing high-speed data transfers in a compact system.

12.1.1 Features

The BSC has the following features:

- Physical address space is divided into six areas
  - A maximum 64 Mbytes for each of the six areas, 0, 2 to 6
  - Area bus width can be selected by register (area 0 is set by external pin)
  - Wait states can be inserted using the $\overline{\text{WAIT}}$ pin
  - Wait state insertion can be controlled through software. Register settings can be used to specify the insertion of 1–10 cycles independently for each area (1–38 cycles for areas 5 and 6 and the PCMCIA interface only)
  - The type of memory connected can be specified for each area, and
  - Control signals are output for direct memory connection
  - Wait cycles are automatically inserted to avoid data bus conflict for continuous memory accesses to different areas or writes directly following reads of the same area
- Direct interface to synchronous DRAM (except if clock ratio Iϕ:Bϕ = 1:1)
  - Multiplexes row/column addresses according to synchronous DRAM capacity
  - Supports burst operation
  - Has both auto-refresh and self-refresh functions
  - Controls timing of synchronous DRAM direct-connection control signals according to register setting
- Burst ROM interface
  - Insertion of wait states controllable through software
  - Register setting control of burst transfers
- PCMCIA direct-connection interface*
  - Insertion of wait states controllable through software
  - Bus sizing function for I/O bus width (only in the little endian mode)

- Refresh function
  - Refresh cycles will be automatically maintained in the sleep mode even after the external bus frequency is reduced to 1/4 of its normal operating frequency
- The refresh counter can be used as an interval timer
  - Outputs an interrupt request signal using the compare-matching function
  - Outputs an interrupt request signal when the refresh counter overflows
- Automatically disables the output of clock signals to anywhere but the refresh counter, except during execution of external bus cycles

Note: * PCMCIA direct interface supported by the BSC is only signals and bus protocols shown in table 12.5. For details on other control signals, refer to section 30, PC Card Controller (PCC) (external circuit and this LSI on-chip card controller. In this BSI, both areas 5 and 6 has PCMICIA direct interface function common to the SH3 Series. The on-chip PC Card Controller supports only area 6.

12.1.2 Block Diagram

Figure 12.1 shows the functional block diagram of the BSC.

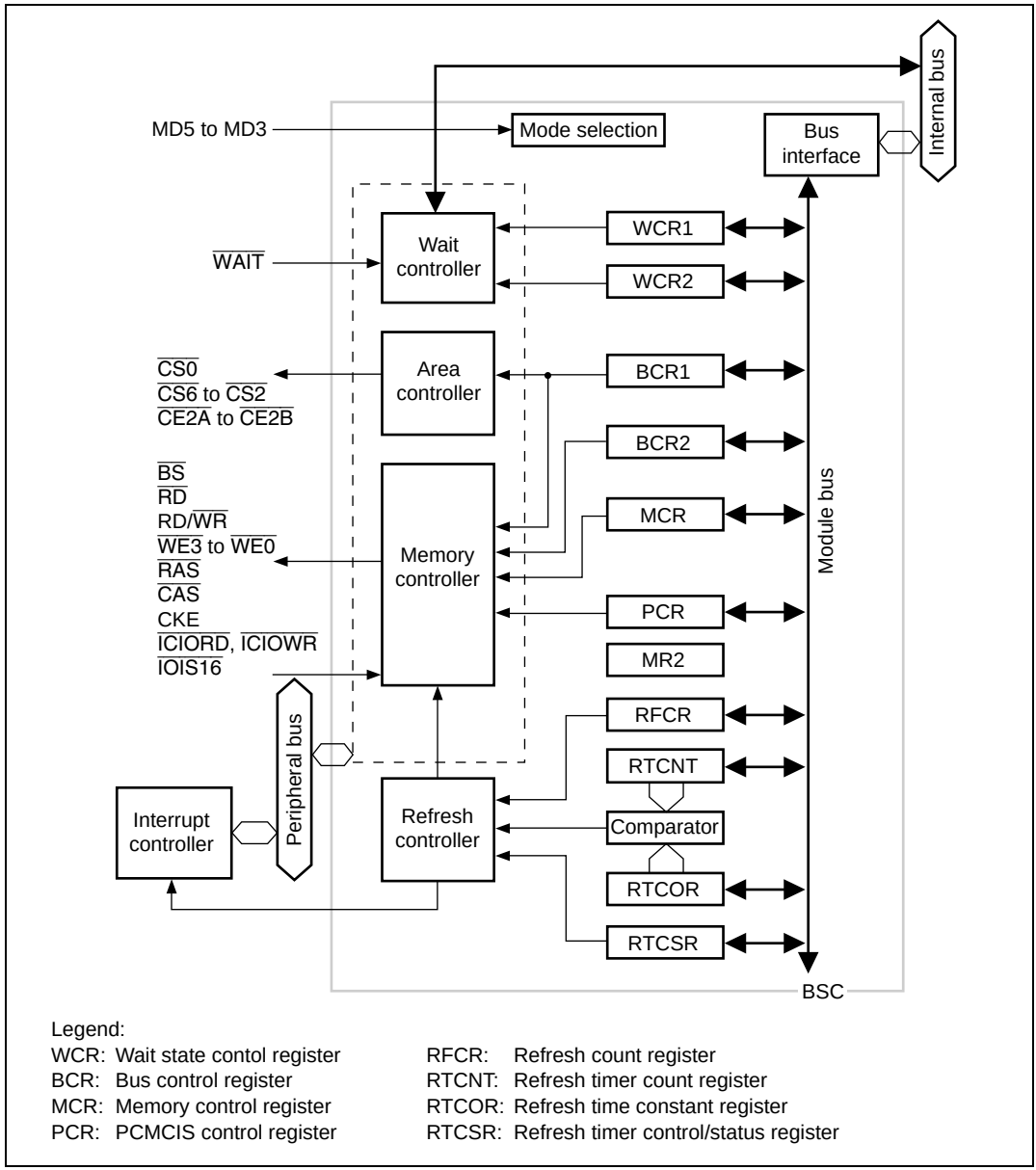


Figure 12.1 Corresponding to Logical Address Space and Physical Address Space

12.1.3 Pin Configuration

Table 12.1 lists the BSC pin configuration.

Table 12.1 Pin Configuration

Pin Name	Signal	I/O	Description
Address bus	A25–A0	Output	Address output
Data bus	D15–D0	I/O	Data I/O
	D31–D16	I/O	When 32-bit bus width, data I/O
Bus cycle start	$\overline{BS}$	Output	Shows start of bus cycle. During burst transfers, asserts every data cycle.
Chip select 0, 2–4	$\overline{CS0}$, $\overline{CS2}$–$\overline{CS4}$	Output	Chip select signal to indicate area being accessed.
Chip select 5, 6	$\overline{CS5}$/$\overline{CE1A}$, $\overline{CS6}$/$\overline{CE1B}$	Output	Chip select signal to indicate area being accessed. $\overline{CS5}$/$\overline{CE1A}$ and $\overline{CS6}$/$\overline{CE1B}$ can also be used as $\overline{CE1A}$ and $\overline{CE1B}$ of PCMCIA.
PCMCIA card select	$\overline{CE2A}$, $\overline{CE2B}$	Output	When PCMCIA is used, CE2A and CE2B
Read/write	RD/$\overline{WR}$	Output	Data bus direction indicator signal. PCMCIA write indicator signal.
Row address strobe 3	$\overline{RAS3}$	Output	When synchronous DRAM is used in area 3, RAS3 for 64-Mbyte address.
Column address strobe	$\overline{CAS}$	Output	When synchronous DRAM is used, CAS signal is used for 64Mbyte address.
Data enable 0	$\overline{WE0}$/DQMLL	Output	When memory other than synchronous DRAM is used, selects D7 to D0 write strobe signal. When synchronous DRAM is used, selects D7 to D0.
Data enable 1	$\overline{WE1}$/DQMLU/ $\overline{WE}$	Output	When memory other than synchronous DRAM and PCMCIA is used, selects D15 to D8 write strobe signal. When synchronous DRAM is used, selects D15 to D8. When PCMCIA is used, strobe signal that indicates the write cycle.
Data enable 2	$\overline{WE2}$/DQMUL/ $\overline{ICIORD}$	Output	When memory other than synchronous DRAM and PCMCIA is used, selects D23 to D16 write strobe signal. When synchronous DRAM is used, selects D23 to D16. When PCMCIA is used, strobe signal indicating I/O read.

Pin Name	Signal	I/O	Description
Data enable 3	$\overline{WE3}$/DQMUU/ $\overline{ICIOWR}$	Output	When memory other than synchronous DRAM and PCMCIA is used, selects D31 to D24 write strobe signal. When synchronous DRAM is used, selects D31 to D24. When PCMCIA is used, strobe signal indicating I/O write.
Read	$\overline{RD}$	Output	Strobe signal indicating read cycle
Wait	$\overline{WAIT}$	Input	Wait state request signal
Clock enable	CKE	Output	Clock enable control signal of synchronous DRAM
IOIS16	$\overline{IOIS16}$	Input	Signal indicating PCMCIA 16-bit I/O. Valid only in little-endian mode.
Bus release request	$\overline{BREQ}$	Input	Bus release request signal
Bus release acknowledgment	$\overline{BACK}$	Output	Bus release acknowledge signal
Mode selection	MD5 to MD3	Input	Specifies bus width and endian of area 0

12.1.4 Register Configuration

The BSC has 11 registers (table 12.2). The synchronous DRAM also has a built-in synchronous DRAM mode register. These registers control direct connection interfaces to memory, wait states, and refreshes.

Table 12.2 Register Configuration

Name		Abbr.	R/W	Initial Value*	Address	Bus Width
Bus control register 1		BCR1	R/W	H'0000	H'FFFFFF60	16
Bus control register 2		BCR2	R/W	H'3FF0	H'FFFFFF62	16
Wait state control register 1		WCR1	R/W	H'3FF3	H'FFFFFF64	16
Wait state control register 2		WCR2	R/W	H'FFFF	H'FFFFFF66	16
Individual memory control register		MCR	R/W	H'0000	H'FFFFFF68	16
PCMCIA control register		PCR	R/W	H'0000	H'FFFFFF6C	16
Refresh timer control/status register		RTCSR	R/W	H'0000	H'FFFFFF6E	16
Refresh timer counter		RTCNT	R/W	H'0000	H'FFFFFF70	16
Refresh time constant register		RTCOR	R/W	H'0000	H'FFFFFF72	16
Refresh count register		RFCR	R/W	H'0000	H'FFFFFF74	16
Synchronous DRAM mode register	For area 2	SDMR	W	—	H'FFFFD000–H'FFFFDFFF	8
	For area 3				H'FFFFE000–H'FFFFEFFF	

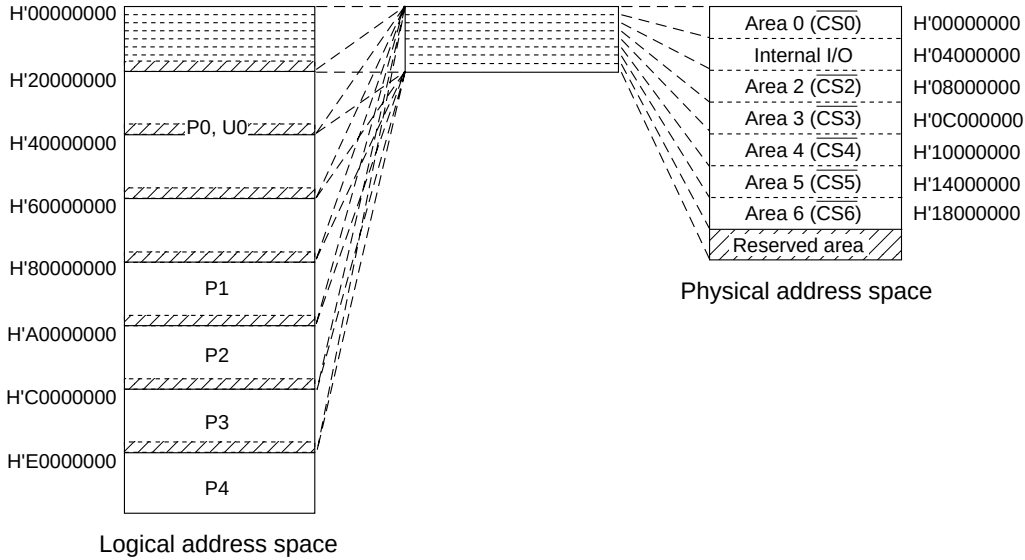
Notes: For details, see section 12.2.7, Synchronous DRAM Mode Register (SDMR).
* Initialized by power-on resets.

12.1.5 Area Overview

Space Allocation: In the architecture of this LSI, both logical spaces and physical spaces have 32-bit address spaces. The logical space is divided into five areas by the value of the upper bits of the address. The physical space is divided into eight areas.

Logical space can be allocated at physical spaces using a memory management unit (MMU). For details, refer to section 3, Memory Management Unit (MMU), which describes area allocation for physical spaces.

As listed in table 12.3, this LSI can be connected directly to six areas of memory/PC card interface, and it outputs chip select signals ($\overline{CS0}$, $\overline{CS2}$ to $\overline{CS6}$, $\overline{CE2A}$, $\overline{CE2B}$) for each of them. $\overline{CS0}$ is asserted during area 0 access; $\overline{CS6}$ is asserted during area 6 access. When PCMCIA interface is selected in area 5 or 6, in addition to $\overline{CS5}$/$\overline{CS6}$, $\overline{CE2A}$/$\overline{CE2B}$ are asserted for the corresponding bytes accessed.



Note: For logical address spaces P0 and P3, when the memory management unit (MMU) is on, it can optionally generate a physical address for the logical address. It can be applied when the MMU is off and when the MMU is on and each physical address for the logical address is equal except for upper three bits. When translating logical addresses to arbitrary physical addresses, refer to table 12.3 "Physical Address Space Map".

Figure 12.2 Corresponding to Logical Address Space and Physical Address Space

Table 12.3 Physical Address Space Map

Area	Connectable Memory	Physical Address (A28 to A0)	Capacity	Access Size
0	Ordinary memory*1, burst ROM	H'00000000 to H'03FFFFFF	64 Mbytes	8, 16, 32*2
		H'00000000 + H'2000000 x n to H'03FFFFFF + H'2000000 x n	Shadow	(n = 1 to 6)
1	Internal I/O registers*7	H'04000000 to H'07FFFFFF	64 Mbytes	8, 16, 32*3
		H'04000000 + H'2000000 x n to H'07FFFFFF + H'2000000 x n	Shadow	(n = 1 to 6)
2	Ordinary memory*1, Synchronous DRAM	H'08000000 to H'0BFFFFFF	64 Mbyte	8, 16, 32*3 *4
		H'08000000 + H'2000000 x n to H'0BFFFFFF + H'2000000 x n	Shadow	(n = 1 to 6)
3	Ordinary memory, Synchronous DRAM	H'0C000000 to H'0FFFFFFF	64 Mbytes	8, 16, 32*3 *4
		H'0C000000 + H'2000000 x n to H'0FFFFFFF + H'2000000 x n	Shadow	(n = 1 to 6)
4	Ordinary memory	H'10000000 to H'13FFFFFF	64 Mbytes	8, 16, 32*3
		H'10000000 + H'2000000 x n to H'13FFFFFF + H'2000000 x n	Shadow	(n = 1 to 6)
5	Ordinary memory, PCMCIA, burst ROM	H'14000000 to H'15FFFFFF	32 Mbytes	8, 16, 32*3 *5
		H'16000000 to H'17FFFFFF	32 Mbytes	
		H'16000000 + H'2000000 x n to H'17FFFFFF + H'2000000 x n	Shadow	(n = 1 to 6)
6	Ordinary memory, PCMCIA, bust ROM	H'18000000 to H'19FFFFFF	32 Mbytes	8, 16, 32*3 *5
		H'1A000000 to H'1BFFFFFF	32 Mbytes	
		H'1A000000 + H'2000000 x n to H'1BFFFFFF + H'2000000 x n	Shadow	(n = 1 to 6)
7*6	Reserved area	H'1C000000 + H'20000000 × n to H'1FFFFFFF + H'20000000 × n		n = 0–7

Notes: 1. Memory with interface such as SRAM or ROM.
2. Use external pin to specify memory bus width.
3. Use register to specify memory bus width.
4. With synchronous DRAM interfaces, bus width must be 16 or 32 bits.
5. With PCMCIA interface, bus width must be 8 or 16 bits.
6. The access to reserved area is prohibited.
7. When the control register in area 1 is not used for address translation by the MMU, set the top three bits of the logical address to 101 to allocate in the P2 space.

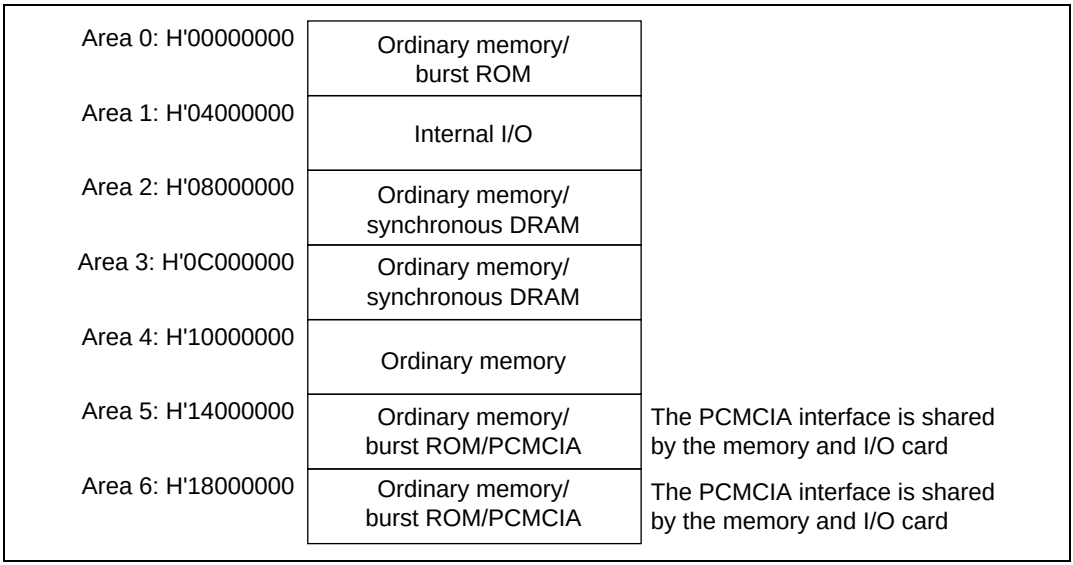


Figure 12.3 Physical Space Allocation

Memory Bus Width: The memory bus width in this LSI can be set for each area. In area 0, an external pin can be used to select byte (8 bits), word (16 bits), or longword (32 bits) on power-on reset as setting of MD4 and MD3 as below table .

Table 12.4 Correspondence between External Pins (MD4 and MD3) and Memory bus width in area0

MD4	MD3	Memory Size
0	0	Reserved (Setting prohibited)
	1	8 bits
1	0	16 bits
	1	32 bits

For areas 2 to 6, byte, word, and longword may be chosen for the bus width using bus control register 2 (BCR2) whenever ordinary memory, ROM, or burst ROM are used.

When the PCMCIA interface is used, set the bus width to byte or word. When synchronous DRAM is connected to both area 2 and area 3, set the same bus width for areas 2 and 3. When using the port function, set each of the bus widths to byte or word for all areas. For more information, see section 12.2.2, Bus Control Register 2 (BCR2).

Shadow Space: Areas 0, 2 to 6 are decoded by physical addresses A28 to A26, which correspond to areas 000 to 110. Address bits 31 to 29 are ignored. This means that the range of area 0 addresses, for example, is H'00000000 to H'03FFFFFF, and its corresponding shadow space is the address space obtained by adding to it H'20000000 × n (n = 1 to 6). The address range for area 7, which is on-chip I/O space, is H'1C000000 to H'1FFFFFFF. The address space H'1C000000 + H'20000000 × n to H'1FFFFFFF + H'20000000 × n (n = 0 to 7) corresponding to the area 7 shadow space is reserved, so do not use it.

12.1.6 PC Card Support

The Bus Controller of this LSI supports protocol signals of PCMCIA standard interface specifications in physical space areas 5 and 6 as another SH3 Series.

PC Card Bus signal (CEIA,CE2A,CE1B,CE2B,IOIS16) are supported for PC Card Bus Protocol as same as SH7708/SH7709/SH7729 series.

Dynamic bus sizing of I/O bus width is supported only in the little endian made.

Table 12.5 SH7727 and PCMCIA Pins

SH7727	PCMCIA
$\overline{\text{CE1A}}$	$\overline{\text{CE1}}$
$\overline{\text{CE1B}}$	$\overline{\text{CE1}}$
$\overline{\text{CE2A}}$	$\overline{\text{CE2}}$
$\overline{\text{CE2B}}$	$\overline{\text{CE2}}$
$\overline{\text{WE}}$	$\overline{\text{WE}}$/$\overline{\text{PGM}}$
$\overline{\text{RD}}$	$\overline{\text{OE}}$
$\overline{\text{IOIS16}}$	$\overline{\text{WP}}$/$\overline{\text{IOIS16}}$
$\overline{\text{ICIORD}}$	$\overline{\text{IORD}}$
$\overline{\text{ICIOWR}}$	$\overline{\text{IOWR}}$
A25–A0	A25–A0
D15–D0	D15–D0

12.2 BSC Registers

12.2.1 Bus Control Register 1 (BCR1)

Bus control register 1 (BCR1) is a 16-bit read/write register that sets the functions and bus cycle state for each area. It is initialized to H'0000 by a power-on reset, but is not initialized by a manual reset or by standby mode. Do not access external memory outside area 0 until BCR1 register initialization is complete.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PULA	PULD	HIZ MEM	HIZ CNT	ENDI AN	A0 BST1	A0 BST0	A5 BST1	A5 BST0	A6 BST1	A6 BST0	DRAM TP2	DRAM TP1	DRAM TP0	A5 PCM	A6 PCM
Initial value:	0	0	0	0	0/1*	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: * Samples the value of the external pin (MD5) designating endian at power-on reset.

Bit 15—Pins A25 to A0 Pull-Up (PULA): Specifies whether or not pins A25 to A0 are pulled up for 4 cycles immediately after $\overline{\text{BACK}}$ is asserted.

Bit 15: PULA	Description	
0	Not pulled up	(Initial value)
1	Pulled up	

Bit 14—Pins D31 to D0 Pull-Up (PULD): Specifies whether or not pins D31 to D0 are pulled up when not in use.

Bit 14: PULD	Description
0	Not pulled up (Initial value)
1	Pulled up

Bit 13—Hi-Z memory control (HIZMEM): Specifies the state of A25 to A0, $\overline{\text{BS}}$, $\overline{\text{CS}}$, RD/$\overline{\text{WR}}$, $\overline{\text{WE}}$/DQM, $\overline{\text{RD}}$, $\overline{\text{CE2A}}$, $\overline{\text{CE2B}}$ and DRAK0 in standby mode.

Bit 13: HIZMEM	Description	
0	High-impedance state in standby mode.	(Initial value)
1	High in standby mode.	

Bit 12—High-Z Control (HIZCNT): Specifies the state of the $\overline{RAS}$ and the $\overline{CAS}$ signals at standby and bus right release.

Bit 12: HIZCNT	Description
0	The $\overline{RAS}$ and the $\overline{CAS}$ signals are high-impedance state (High-Z) at standby and bus right release. (Initial value)
1	The $\overline{RAS}$ and the $\overline{CAS}$ signals are driven at standby and bus right release.

Bit 11—Endian Flag (ENDIAN): Samples the value of the external pin designating endian upon a power-on reset. Endian for all physical spaces is decided by this bit, which is read-only.

Bit 11: ENDIAN	Description
0	(On reset) Endian setting external pin (MD5) is low. Indicates this LSI is set as big endian.
1	(On reset) Endian setting external pin (MD5) is high. Indicates this LSI is set as little endian.

Bits 10 and 9—Area 0 Burst ROM Control (A0BST1, A0BST0): Specify whether to use burst ROM in physical space area 0. When burst ROM is used, set the number of burst transfers.

Bit 10: A0BST1	Bit 9: A0BST0	Description
0	0	Access area 0 as ordinary memory (Initial value)
	1	Access area 0 as burst ROM (4 consecutive accesses). Can be used when bus width is 8, 16, or 32.
1	0	Access area 0 as burst ROM (8 consecutive accesses). Can be used when bus width is 8 or 16.
	1	Access area 0 as burst ROM (16 consecutive accesses). Can be used only when bus width is 8.

Bits 8 and 7—Area 5 Burst Enable (A5BST1, A5BST0): Specify whether to use burst ROM and PCMCIA burst mode in physical space area 5. When burst ROM and PCMCIA burst mode are used, set the number of burst transfers.

Bit 8: A5BST1	Bit 7: A5BST0	Description
0	0	Access area 5 as ordinary memory (Initial value)
	1	Burst access of area 5 (4 consecutive accesses). Can be used when bus width is 8, 16, or 32.
1	0	Burst access of area 5 (8 consecutive accesses). Can be used when bus width is 8 or 16.
	1	Burst access of area 5 (16 consecutive accesses). Can be used only when bus width is 8.

Bits 6 and 5—Area 6 Burst Enable (A6BST1, A6BST0): Specify whether to use burst ROM and PCMCIA burst mode in physical space area 6. When burst ROM and PCMCIA burst mode are used, set the number of burst transfers.

Bit 6: A6BST1	Bit 5: A6BST0	Description
0	0	Access area 6 as ordinary memory (Initial value)
	1	Burst access of area 6 (4 consecutive accesses). Can be used when bus width is 8, 16, or 32.
1	0	Burst access of area 6 (8 consecutive accesses). Can be used when bus width is 8 or 16.
	1	Burst access of area 6 (16 consecutive accesses). Can be used only when bus width is 8.

Bits 4 to 2—Area 2, Area 3 Memory Type (DRAMTP2, DRAMTP1, DRAMTP0): Designate the types of memory connected to physical space areas 2 and 3. Ordinary memory, such as ROM, SRAM, or flash ROM, can be directly connected. Synchronous DRAM can also be directly connected.

Bit 4: DRAMTP2	Bit 3: DRAMTP1	Bit 2: DRAMTP0	Description
0	0	0	Areas 2 and 3 are ordinary memory (Initial value)
		1	Reserved (Setting disabled)
	1	0	Area 2: ordinary memory; Area 3: synchronous DRAM*1
		1	Areas 2 and 3 are synchronous DRAM*1 *2
1	0	0	Reserved (Setting disabled)
		1	Reserved (Setting disabled)
	1	0	Reserved (Setting disabled)
		1	Reserved (Setting disabled)

Notes: 1. When selecting this mode, set the same bus width for area 2 and area 3.
2. If clock rate is specified as 1φ: Bus clock = 1:1 , synchronous DRAM cannot be accessed.

Bit 1—Area 5 Bus Type (A5PCM): Designates whether to access physical space area 5 as PCMCIA space.

Bit 1: A5PCM	Description	
0	Access physical space area 5 as ordinary memory	(Initial value)
1	Access physical space area 5 as PCMCIA space	

Bit 0—Area 6 Bus Type (A6PCM): Designates whether to access physical space area 6 as PCMCIA space.

Bit 0: A6PCM	Description	
0	Access physical space area 6 as ordinary memory	(Initial value)
1	Access physical space area 6 as PCMCIA space	

12.2.2 Bus Control Register 2 (BCR2)

The bus control register 2 (BCR2) is a 16-bit read/write register that selects the bus-size width of each area. It is initialized to H'3FF0 by a power-on reset, but is not initialized by a manual reset or by standby mode. Do not access external memory outside area 0 until BCR2 register initialization is complete.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	A6 SZ1	A6 SZ0	A5 SZ1	A5 SZ0	A4 SZ1	A4 SZ0	A3 SZ1	A3 SZ0	A2 SZ1	A2 SZ0	—	—	—	—
Initial value:	0	0	1	1	1	1	1	1	1	1	1	1	0	0	0	0
R/W:	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R

Bits 15, 14, 3, 2, 1, and 0—Reserved: These bits are always read as 0. The write value should always be 0.

Bits 2n + 1, 2n—Area n (2 to 6) Bus Size Specification (AnSZ1, AnSZ0): Specify the bus sizes of physical space area n (n = 2 to 6).

Bit 2n + 1: AnSZ1	Bit 2n: AnSZ0	Port A / B	Description
0	0	Unused	Reserved (Setting disabled)
	1		Byte (8-bit) size
1	0		Word (16-bit) size
	1		Longword (32-bit) size
0	0	Used	Reserved (Setting disabled)
	1		Byte (8-bit) size
1	0		Word (16-bit) size
	1		Reserved (Setting disabled)

12.2.3 Wait State Control Register 1 (WCR1)

Wait state control register 1 (WCR1) is a 16-bit read/write register that specifies the number of idle (wait) state cycles inserted for each area. For some memories, the drive of the data bus may not be turned off quickly even when the read signal from the external device is turned off. This can result in conflicts between data buses when consecutive memory accesses are to different memories or when a write immediately follows a memory read. This LSI automatically inserts idle states equal to the number set in WCR1 in those cases.

WCR1 is initialized to H'3FF3 by a power-on reset. It is not initialized by a manual reset or by standby mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	WAIT SEL	—	A6 IW1	A6 IW0	A5 IW1	A5 IW0	A4 IW1	A4 IW0	A3 IW1	A3 IW0	A2 IW1	A2 IW0	—	—	A0 IW1	A0 IW0
Initial value:	0	0	1	1	1	1	1	1	1	1	1	1	0	0	1	1
R/W:	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W

Bit 15—WAIT Sampling Timing Select (WAITSEL): Specifies the $\overline{\text{WAIT}}$ signal sampling timing.

Bit 15: WAITSEL	Description	
0	Set to 1 when $\overline{\text{WAIT}}$ signal is used.*	(Initial value)
1	Sampled at the falling edge of CKIO.	

Note: * If low level is input to the $\overline{\text{WAIT}}$ by setting the WAITSEL bit, the LSI operation cannot be guaranteed.

Bits 14, 3, and 2 —Reserved: These bits are always read as 0. The write value should always be 0.

Bits 2n + 1, 2n—Area n (6 to 2, 0) Intercycle Idle Specification (AnIW1, AnIW0): Specify the number of idles inserted between bus cycles when switching between physical space area n (6 to 2, 0) to another space or between a read access to a write access in the same physical space.

Bit 2n + 1: AnIW1	Bit 2n: AnIW0	Description
0	0	1 idle cycle inserted
	1	1 idle cycle inserted
1	0	2 idle cycles inserted
	1	3 idle cycles inserted (Initial value)

12.2.4 Wait State Control Register 2 (WCR2)

Wait state control register 2 (WCR2) is a 16-bit read/write register that specifies the number of wait state cycles inserted for each area. It also specifies the pitch of data access for burst memory accesses. This allows direct connection of even low-speed memories without an external circuit. WCR2 is initialized to H'FFFF by a power-on reset. It is not initialized by a manual reset or by standby mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	A6 W2	A6 W1	A6 W0	A5 W2	A5 W1	A5 W0	A4 W2	A4 W1	A4 W0	A3 W1	A3 W0	A2 W1	A2 W0	A0 W2	A0 W1	A0 W0
Initial value:	1	1	1	1	1	1	1	1	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	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bits 15 to 13—Area 6 Wait Control (A6W2, A6W1, A6W0): Specify the number of wait states inserted into physical space area 6. Also specify the burst pitch for burst transfer.

			Description			
			First Cycle		Burst Cycle (Excluding First Cycle)	
Bit 15: A6W2	Bit 14: A6W1	Bit 13: A6W0	Inserted Wait States	$\overline{\text{WAIT}}$ Pin	Number of States Per Data Transfer	$\overline{\text{WAIT}}$ Pin
0	0	0	0	Disable	2	Enable
		1	1	Enable	2	Enable
	1	0	2	Enable	3	Enable
		1	3	Enable	4	Enable
1	0	0	4	Enable	4	Enable
		1	6	Enable	6	Enable
	1	0	8	Enable	8	Enable
		1	10 (Initial value)	Enable	10	Enable

Bits 12 to 10—Area 5 Wait Control (A5W2, A5W1, A5W0): Specify the number of wait states inserted into physical space area 5. Also specify the burst pitch for burst transfer.

			Description			
			First Cycle		Burst Cycle (Excluding First Cycle)	
Bit 12: A5W2	Bit 11: A5W1	Bit 10: A5W0	Inserted Wait States	$\overline{\text{WAIT}}$ Pin	Number of States Per Data Transfer	$\overline{\text{WAIT}}$ Pin
0	0	0	0	Disable	2	Enable
		1	1	Enable	2	Enable
	1	0	2	Enable	3	Enable
		1	3	Enable	4	Enable
1	0	0	4	Enable	4	Enable
		1	6	Enable	6	Enable
	1	0	8	Enable	8	Enable
		1	10 (Initial value)	Enable	10	Enable

Bits 9 to 7—Area 4 Wait Control (A4W2, A4W1, A4W0): Specify the number of wait states inserted into physical space area 4.

			Description	
Bit 9: A4W2	**Bit 8: A4W1**	**Bit 7: A4W0**	**Inserted Wait State**	**$\overline{\text{WAIT}}$ Pin**
0	0	0	0	Ignored
		1	1	Enable
	1	0	2	Enable
		1	3	Enable
1	0	0	4	Enable
		1	6	Enable
	1	0	8	Enable
		1	10	Enable (Initial value)

Bits 6 and 5—Area 3 Wait Control (A3W1, A3W0): Specify the number of wait states inserted into physical space area 3.

- For Ordinary memory

		Description		
Bit 6: A3W1	**Bit 5: A3W0**	**Inserted Wait States**	**$\overline{\text{WAIT}}$ Pin**	
0	0	0	Ignored	
	1	1	Enable	
1	0	2	Enable	
	1	3	Enable	(Initial value)

- For Synchronous SDRAM

		Description	
Bit 6: A3W1	**Bit 5: A3W0**	**Synchronous SDRAM: CAS Latency**	
0	0	1	
	1	1	
1	0	2	
	1	3	(Initial value)

Bits 4 and 3—Area 2 Wait Control (A2W1, A2W0): Specify the number of wait states inserted into physical space area 2.

- For Ordinary memory

		Description	
Bit 4: A2W0	**Bit 3: A2W0**	**Inserted Wait States**	**$\overline{\text{WAIT}}$ Pin**
0	0	0	Ignored
	1	1	Enable
1	0	2	Enable
	1	3	Enable (Initial value)

- For Synchronous SDRAM

		Description
Bit 4: A2W1	**Bit 3: A2W0**	**Synchronous DRAM: CAS Latency**
0	0	1
	1	1
1	0	2
	1	3 (Initial value)

Bits 2 to 0—Area 0 Wait Control (A0W2, A0W1, A0W0): Specify the number of wait states inserted into physical space area 0. Also specify the burst pitch for burst transfer.

			Description			
			First Cycle		Burst Cycle (Excluding First Cycle)	
Bit 2: A0W2	**Bit 1: A0W1**	**Bit 0: A0W0**	**Inserted Wait States**	**$\overline{\text{WAIT}}$ Pin**	**Number of States Per Data Transfer**	**$\overline{\text{WAIT}}$ Pin**
0	0	0	0	Ignored	2	Enable
		1	1	Enable	2	Enable
	1	0	2	Enable	3	Enable
		1	3	Enable	4	Enable
1	0	0	4	Enable	4	Enable
		1	6	Enable	6	Enable
	1	0	8	Enable	8	Enable
		1	10 (Initial value)	Enable	10	Enable

12.2.5 Individual Memory Control Register (MCR)

The individual memory control register (MCR) is a 16-bit read/write register that specifies $\overline{RAS}$ and $\overline{CAS}$ timing and burst control for synchronous DRAM (areas 2 and 3), specifies address multiplexing, and controls refresh. This enables direct connection of synchronous DRAM without external circuits.

The MCR is initialized to H'0000 by power-on resets, but is not initialized by manual resets or standby mode. The bits TPC1 to TPC0, RCD1 to RCD0, TRWL1 to TRWL0, TRAS1 to TRAS0, AMX3 to AMX0, and are written to at the initialization after a power-on reset and are not then modified again. When RFSH and RMODE are written to, write the same values to the other bits. When using synchronous DRAM, do not access areas 2 and 3 until this register is initialized.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TPC1	TPC0	RCD1	RCD0	TRWL 1	TRWL 0	TRAS 1	TRAS 0	—	AMX3	AMX2	AMX1	AMX0	RFSH	RMO DE	—
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 and 14—RAS Precharge Time (TPC1, TPC0): These bits set the minimum number of cycles until output of the next bank-active command after precharge, when the synchronous DRAM interface is selected for external memory. However, the number of cycles inserted immediately after the precharge all banks (PALL) command is issued when performing auto-refresh is one fewer than the number of cycles during normal operation.

		Description		
Bit 15: TPC1	**Bit 14: TPC0**	**Normal Operation**	**Immediately After Precharge Command***	**Immediately After Self-refresh**
0	0	1 cycle (Initial value)	0 cycle (Initial value)	2 cycles (Initial value)
	1	2 cycles	1 cycle	5 cycles
1	0	3 cycles	2 cycles	8 cycles
	1	4 cycles	3 cycles	11 cycles

Note: * Immediately after the precharge all banks (PALL) command is issued when performing auto-refresh.

Bits 13 and 12—RAS–CAS Delay (RCD1, RCD0): When synchronous DRAM interface is selected, sets the bank active read/write command delay time.

Bit 13: RCD1	Bit 12: RCD0	Description	
0	0	1 cycle	(Initial value)
	1	2 cycles	
1	0	3 cycles	
	1	4 cycles	

Bits 11 and 10—Write-Precharge Delay (TRWL1, TRWL0): The TRWL bits set the synchronous DRAM write-precharge delay time. This designates the time between the end of a write cycle and the next bank-active command. This is valid only when synchronous DRAM is connected. After the write cycle, the next bank-active command is not issued for the period TPC + TRWL.

Bit 11: TRWL1	Bit 10: TRWL0	Description	
0	0	1 cycle	(Initial value)
	1	2 cycles	
1	0	3 cycles	
	1	Reserved (Setting disabled)	

Bits 9 and 8—$\overline{\text{CAS}}$-Before-$\overline{\text{RAS}}$ Refresh $\overline{\text{RAS}}$ Assert Time (TRAS1, TRAS0): When synchronous DRAM interface is selected, no bank-active command is issues during the period TPC + TRAS after an auto-refresh command.

Bit 9: TRAS1	Bit 8: TRAS0	Description	
0	0	2 cycles	(Initial value)
	1	3 cycles	
1	0	4 cycles	
	1	5 cycles	

Bit 7—Reserved: This bit is always read as 0. The write value should always be 0.

Bits 6 to 3—Address Multiplex (AMX3 , AMX2, AMX1, AMX0): The AMX bits specify address multiplexing for synchronous DRAM. The actual address shift value differs between synchronous DRAM interface.

For Synchronous DRAM Interface: (see table 12.12)

Bit6: AMX3	Bit5: AMX2	Bit 4: AMX1	Bit 3: AMX0	Description
1	1	0	1	The row address begins with A10 when bus width is 16 bit. The row address begins with A11 when bus width is 32 bit. (The A10 value is output at A1 when the row address is output. 4M × 16-bit × 4-bank products)
		1	0	The row address begins with A11 when bus width is 16 bit. (The A11 value is output at A1 when the row address is output. 8M × 16-bit × 4-bank products)*1
0	1	0	0	The row address begins with A9 when bus width is 16 bit. The row address begins with A10 when bus width is 32 bit. (The A9 value is output at A1 when the row address is output. 1M × 16-bit × 4-bank products)
			1	The row address begins with A10 when bus width is 16 bit. The row address begins with A11 when bus width is 32 bit. (The A10 value is output at A1 when the row address is output. 2M × 16-bit × 4-bank products)
		1	0	The row address begins with A11 when bus width is 32 bit.*2 (The A11 value is output at A1 when the row address is output. 2M × 16-bit × 4-bank products)
			1	The row address begins with A9 when bus width is 16 bit. The row address begins with A10 when bus width is 32 bit. (The A9 value is output at A1 when the row address is output. 512K × 32-bit × 4-bank products)
	0	0	0	Reserved. AMX3 to AMX0 must be set to *1*** before accessing synchronous DRAM memory. (Initial value)
Values other than above				Reserved (illegal setting)

Notes: 1. Can only be set when using a 16-bit bus width.
2. Can only be set when using a 32-bit bus width.

Bit 2—Refresh Control (RFSH): The RFSH bit determines whether or not the refresh operation of the synchronous DRAM is performed. The timer for generation of the refresh request frequency can also be used as an interval timer.

Bit 2: RFSH	Description	
0	No refresh	(Initial value)
1	Refresh	

Bit 1—Refresh Mode (RMODE): The RMODE bit selects whether to perform an ordinary refresh or a self-refresh when the RFSH bit is 1. When the RFSH bit is 1 and this bit is 0, a CAS-before-RAS refresh or an auto-refresh is performed on synchronous DRAM at the period set by the refresh-related registers RTCNT, RTCOR and RTCSR. When a refresh request occurs during an external bus cycle, the bus cycle will be ended and the refresh cycle performed. When the RFSH bit is 1 and this bit is also 1, the synchronous DRAM will wait for the end of any executing external bus cycle before going into a self-refresh. All refresh requests to memory that is in the self-refresh state are ignored.

Bit 1: RMODE	Description	
0	CAS-before-RAS refresh (RFSH must be 1)	(Initial value)
1	Self-refresh (RFSH must be 1)	

Bit 0—Reserved: This bit is always read as 0. The write value should always be 0.

12.2.6 PCMCIA Control Register (PCR)

The PCMCIA control register (RCR) specifies the assert/negate timing of the $\overline{OE}$ and $\overline{WE}$ signals ($\overline{RD}$ and $\overline{WE1}$ pins of this LSI) for the PCMCIA interface connected to areas 5 and 6. Note that the assertion widths of $\overline{OE}$ and $\overline{WE}$ are set using the wait control bits of the WCR2 register.

The PCR register is a 16-bit read/write register. It is initialized at a power-on reset to H'0000. However, the register is not initialized and the contents remain unchanged at a manual reset and when in standby mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	A6 W3	A5 W3	—	—	A5 TED2	A6 TED2	A5 TEH2	A6 TEH2	A5 TED1	A5 TED0	A6 TED1	A6 TED0	A5 TEH1	A5 TEH0	A6 TEH1	A6 TEH0
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

Bit 15—Area 6 Wait Control (A6W3): The A6W3 bit specifies the number of inserted wait states for area 6 combined with bits A6W2 to A6W0 in WCR2. It also specifies the number of transfer states in burst transfer. Set this bit to 0 when area 6 is not set to PCMCIA.

				Top Cycle		Burst Cycle	
A6W3	A6W2	A6W1	A6W0	Inserted Wait State	$\overline{\text{WAIT}}$ Pin	Number of States per One-data Transfer	$\overline{\text{WAIT}}$ Pin
0	0	0	0	0	Ignored	2	Enabled
			1	1	Enabled	2	Enabled
		1	0	2	Enabled	3	Enabled
			1	3	Enabled	4	Enabled
	1	0	0	4	Enabled	5	Enabled
			1	6	Enabled	7	Enabled
		1	0	8	Enabled	9	Enabled
			1	10 (Initial value)	Enabled	11	Enabled
1	0	0	0	12	Enabled	13	Enabled
			1	14	Enabled	15	Enabled
		1	0	18	Enabled	19	Enabled
			1	22	Enabled	23	Enabled
	1	0	0	26	Enabled	27	Enabled
			1	30	Enabled	31	Enabled
		1	0	34	Enabled	35	Enabled
			1	38	Enabled	39	Enabled

Bit 14—Area 5 Wait Control (A5W3): The A5W3 bit specifies the number of inserted wait states for area 5 combined with bits A5W2 to A5W0 in WCR2. It also specifies the number of transfer states in burst transfer. Set this bit to 0 when area 5 is not set to PCMCIA.

The relationship between the setting value and the number of waits is the same as A6W3.

Bits 13 and 12—Reserved: These bits are always read as 0. The write value should always be 0.

Bits 11, 7, and 6—Area 5 Address $\overline{OE}$/$\overline{WE}$ Assert Delay (A5TED2, A5TED1, and A5TED0): The A5TED bits specify the address to $\overline{OE}$/$\overline{WE}$ assert delay time for the PCMCIA interface connected to area 5.

Bit 11: A5TED2	Bit 7: A5TED1	Bit 6: A5TED0	Description	
0	0	0	0.5-cycle delay	(Initial value)
		1	1.5-cycle delay	
	1	0	2.5-cycle delay	
		1	3.5-cycle delay	
1	0	0	4.5-cycle delay	
		1	5.5-cycle delay	
	1	0	6.5-cycle delay	
		1	7.5-cycle delay	

Bits 10, 5 and 4—Area 6 Address $\overline{OE}$/$\overline{WE}$ Assert Delay (A6TED2, A6TED1, and A6TED0): The A6TED bits specify the address to $\overline{OE}$/$\overline{WE}$ assert delay time for the PCMCIA interface connected to area 6.

Bit 10: A6TED2	Bit 5: A6TED1	Bit 4: A6TED0	Description	
0	0	0	0.5-cycle delay	(Initial value)
		1	1.5-cycle delay	
	1	0	2.5-cycle delay	
		1	3.5-cycle delay	
1	0	0	4.5-cycle delay	
		1	5.5-cycle delay	
	1	0	6.5-cycle delay	
		1	7.5-cycle delay	

Bits 9, 3, and 2—Area 5 $\overline{OE}$/$\overline{WE}$ Negate Address Delay(A5TEH2, A5TEH1, and A5TEH0): The A5TEH bits specify the $\overline{OE}$/$\overline{WE}$ negate address delay time for the PCMCIA interface connected area 5.

Bit 9: A5TEH2	Bit 3: A5TEH1	Bit 2: A5TEH0	Description	
0	0	0	0.5-cycle delay	(Initial value)
		1	1.5-cycle delay	
	1	0	2.5-cycle delay	
		1	3.5-cycle delay	
1	0	0	4.5-cycle delay	
		1	5.5-cycle delay	
	1	0	6.5-cycle delay	
		1	7.5-cycle delay	

Bits 8, 1, and 0—Area6 $\overline{OE}$/$\overline{WE}$ Negate Address Delay (A6TEH2, A6TEH1, and A6TEH0): The A6TEH bits specify the $\overline{OE}$/$\overline{WE}$ negate address delay time for the PCMCIA interface connected to area 6.

Bit 8: A6TEH2	Bit 1: A6TEH1	Bit 0: A6TEH0	Description	
0	0	0	0.5-cycle delay	(Initial value)
		1	1.5-cycle delay	
	1	0	2.5-cycle delay	
		1	3.5-cycle delay	
1	0	0	4.5-cycle delay	
		1	5.5-cycle delay	
	1	0	6.5-cycle delay	
		1	7.5-cycle delay	

12.2.7 Synchronous DRAM Mode Register (SDMR)

The synchronous DRAM mode register (SDMR) is written to via the synchronous DRAM address bus and is an 8-bit write-only register. It sets synchronous DRAM mode for areas 2 and 3. SDMR is undefined after a power-on reset. The register contents are not initialized by a manual reset or standby mode; values remain unchanged.

Bit:	31	12	11	10	9	8	7	6	5	4	3	2	1	0
	SDMR address		—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	—	—	W*	W*	W	W	W	W	W	W	W	W	—	—

Note: * Depending on the type of synchronous DRAM.

Writes to the synchronous DRAM mode register use the address bus rather than the data bus. If the value to be set is X and the SDMR address is Y, the value X is written in the synchronous DRAM mode register by writing in address X + Y. Since, with a 32-bit bus width, A0 of the synchronous DRAM is connected to A2 of the chip and A1 of the synchronous DRAM is connected to A3 of the chip, the value actually written to the synchronous DRAM is the X value shifted two bits right. With a 16-bit bus width, the value written is the X value shifted one bit right. For example, with a 32-bit bus width, when H'0230 is written to the SDMR register of area 2, random data is written to the address H'FFFFD000 (address Y) + H'08C0 (value X), or H'FFFFD8C0. As a result, H'0230 is written to the SDMR register. The range for value X is H'0000 to H'0FFC. When H'0230 is written to the SDMR register of area 3, random data is written to the address H'FFFFE000 (address Y) + H'08C0 (value X), or H'FFFFE8C0. As a result, H'0230 is written to the SDMR register. The range for value X is H'0000 to H'0FFC.

12.2.8 Refresh Timer Control/Status Register (RTCSR)

The refresh timer control/status register (RTCSR) is a 16-bit read/write register that specifies the refresh cycle, whether to generate an interrupt, and that interrupt's cycle. It is initialized to H'0000 by a power-on reset, but is not initialized by a manual reset or standby mode. Before specifying the CKS2 to CKS0 of RTCST, the RTCOR must be specified.

Note: Writing to the RTCSR differs from that to general registers to ensure the RTCSR is not rewritten incorrectly. Use the word-transfer instruction to set the upper byte as B'10100101 and the lower byte as the write data. For details, see section 12.2.12, Cautions on Accessing Refresh Control Related Registers.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	CMF	CMIE	CKS2	CKS1	CKS0	OVF	OVIE	LMTS
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bits 15 to 8—Reserved: These bits are always read as 0. The write value should always be 0.

Bit 7—Compare Match Flag (CMF): The CMF status flag indicates that the values of RTCNT and RTCOR match.

Bit 7: CMF	Description
0	The values of RTCNT and RTCOR do not match. Clear condition: When a refresh is performed After 0 has been written in CMF and RFSH = 1 and RMODE = 0 (to perform a CBR refresh). (Initial value)
1	The values of RTCNT and RTCOR match. Set condition: RTCNT = RTCOR*

Note: * Contents do not change when 1 is written to CMF.

Bit 6—Compare Match Interrupt Enable (CMIE): Enables or disables an interrupt request caused when the CMF of RTCSR is set to 1. Do not set this bit to 1 when using CAS-before-RAS refresh or auto-refresh.

Bit 6: CMIE	Description	
0	Disables an interrupt request caused by CMF	(Initial value)
1	Enables an interrupt request caused by CMF	

Bits 5 to 3—Clock Select Bits (CKS2 to CKS0): Select the clock input to RTCNT. The source clock is the external bus clock (BCLK). The RTCNT count clock is CKIO divided by the specified ratio. The specified ratios are shown below in the normal external bus clock. Before specifying the CKS2 to CKS0 of RTCST, the RTCOR must be specified.

			Description	
Bit 5: CKS2	**Bit 4: CKS1**	**Bit 3: CKS0**	**Normal external bus clock**	
0	0	0	Disables clock input	(Initial value)
		1	Bus clock (CKIO)/4	
	1	0	CKIO/16	
		1	CKIO/64	
1	0	0	CKIO/256	
		1	CKIO/1024	
	1	0	CKIO/2048	
		1	CKIO/4096	

Bit 2—Refresh Count Overflow Flag (OVF): The OVF status flag indicates when the number of refresh requests indicated in the refresh count register (RFCR) exceeds the limit set in the LMTS bit of RTCSR.

Bit 2: OVF	**Description**	
0	RFCR has not exceeded the count limit value set in LMTS Clear Conditions: When 0 is written to OVF	(Initial value)
1	RFCR has exceeded the count limit value set in LMTS Set Conditions: When the RFCR value has exceeded the count limit value set in LMTS*	

Note: * Contents don't change when 1 is written to OVF.

Bit 1—Refresh Count Overflow Interrupt Enable (OVIE): OVIE selects whether to suppress generation of interrupt requests by OVF when the OVF bit of RTCSR is set to 1.

Bit 1: OVIE	**Description**	
0	Disables interrupt requests from the OVF	(Initial value)
1	Enables interrupt requests from the OVF	

Bit 0—Refresh Count Overflow Limit Select (LMTS): Indicates the count limit value to be compared to the number of refreshes indicated in the refresh count register (RFCR). When the value RFCR overflows the value specified by LMTS, the OVF flag is set.

Bit 0: LMTS	Description	
0	Count limit value is 1024	(Initial value)
1	Count limit value is 512	

12.2.9 Refresh Timer Counter (RTCNT)

RTCNT is a 16-bit read/write register. RTCNT is an 8-bit counter that counts up with input clocks. The clock select bits (CKS2 to CKS0) of RTCSR select the input clock. When RTCNT matches RTCOR, the CMF bit of TCSR is set and RTCNT is cleared. RTCNT is initialized to H'00 by a power-on reset; it continues incrementing after a manual reset; it is not initialized by standby mode and holds its values unchanged.

Note: Writing to the RTCNT differs from that to general registers to ensure the RTCNT is not rewritten incorrectly. Use the word-transfer instruction to set the upper byte as B'10100101 and the lower byte as the write data. For details, see section 12.2.12, Cautions on Accessing Refresh Control Related Registers.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
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

12.2.10 Refresh Time Constant Register (RTCOR)

The refresh time constant register (RTCOR) is a 16-bit read/write register. The values of RTCOR and RTCNT (bottom 8 bits) are constantly compared. When the values match, the compare match flag (CMF) of RTCSR is set and RTCNT is cleared to 0. When the refresh bit (RFSH) of the individual memory control register (MCR) is set to 1 and the refresh mode is set to CAS-before-RAS refresh, a memory refresh cycle occurs when the CMF bit is set. RTCOR is initialized to H'00 by a power-on reset. It is not initialized by a manual reset or standby mode, but holds its contents. Make the RTCOR setting before setting bits CKS2 to CKS0 in RTCSR.

Note: Writing to the RTCOR differs from that to general registers to ensure the RTCOR is not rewritten incorrectly. Use the word-transfer instruction to set the upper byte as B'10100101 and the lower byte as the write data. For details, see section 12.2.12, Cautions on Accessing Refresh Control Related Registers.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
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

12.2.11 Refresh Count Register (RFCR)

The refresh count register (RFCR) is a 16-bit read/write register. It is a 10-bit counter that increments every time RTCOR and RTCNT match. When RFCR exceeds the count limit value set in the LMTS of RTCSR, RTCSR's OVF bit is set and RFCR clears. RFCR is initialized to H'0000 when a power-on reset is performed. It is not initialized by a manual reset or standby mode, but holds its contents.

Note: Writing to the RFCR differs from that to general registers to ensure the RFCR is not rewritten incorrectly. Use the word-transfer instruction to set the MSB and followed six bits of upper bytes as B'101001 and remaining bits as the write data. For details, see section 12.2.12, Cautions on Accessing Refresh Control Related Registers.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
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

12.2.12 Cautions on Accessing Refresh Control Related Registers

RFCR, RTCSR, RTCNT, and RTCOR require that a specific code be appended to the data when it is written to prevent data from being mistakenly overwritten by program overruns or other write operations (figure 12.4). Perform reads and writes using the following methods:

1. Writing to RFCR, RTCSR, RTCNT, and RTCOR

   When writing to RFCR, RTCSR, RTCNT, and RTCOR, use only word transfer instructions. You cannot write with byte transfer instructions.

   When writing to RTCNT, RTCSR, or RTCOR, place B'10100101 in the upper byte and the write data in the lower byte. When writing to RFCR, place B'101001 in the top 6 bits and the write data in the remaining bits, as shown in figure 12.4.
2. Reading from RFCR, RTCSR, RTCNT, and RTCOR

   When reading from RFCR, RTCSR, RTCNT, and RTCOR, carry out reads with 16-bit width. 0 is read out from undefined bit sections.

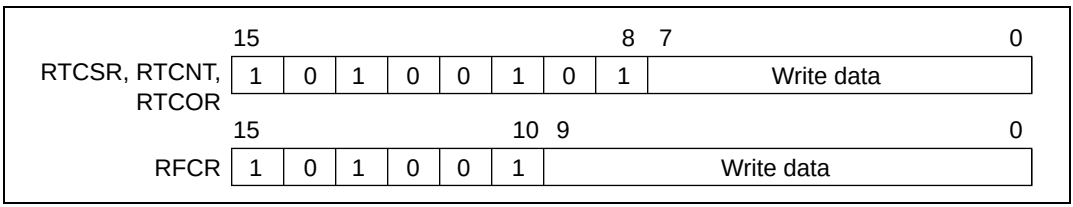


Figure 12.4 Writing to RFCR, RTCSR, RTCNT, and RTCOR

12.3 BSC Operation

12.3.1 Endian/Access Size and Data Alignment

This LSI supports both big endian, in which the 0 address is the most significant byte in the byte data, and little endian, in which the 0 address is the least significant byte. This switchover is designated by an external pin (MD5 pin) at the time of a power-on reset. After a power-on reset, big endian is engaged when MD5 is low; little endian is engaged when MD5 is high.

Three data bus widths are available for ordinary memory (byte, word, longword) and two data bus widths (word and long word) for synchronous DRAM. For the PCMCIA interface, choose from byte and word. This means data alignment is done by matching the device's data width and endian. The access unit must also be matched to the device's bus width. This also means that when longword data is read from a byte-width device, the read operation must happen 4 times. In this LSI, data alignment and conversion of data length is performed automatically between the respective interfaces.

Tables 12.6 to 12.11 show the relationship between endian, device data width, and access unit.

Table 12.6 32-Bit External Device/Big Endian Access and Data Alignment

	Data Bus				Strobe Signals			
Operation	D31 to D24	D23 to D16	D15 to D8	D7 to D0	$\overline{WE3}$, DQMUU	$\overline{WE2}$, DQMUL	$\overline{WE1}$, DQMLU	$\overline{WE0}$, DQMLL
Byte access at 0	Data 7 to 0	—	—	—	Assert			
Byte access at 1	—	Data 7 to 0	—	—		Assert		
Byte access at 2	—	—	Data 7 to 0	—			Assert	
Byte access at 3	—	—	—	Data 7 to 0				Assert
Word access at 0	Data 15 to 8	Data 7 to 0	—	—	Assert	Assert		
Word access at 2	—	—	Data 15 to 8	Data 7 to 0			Assert	Assert
Longword access at 0	Data 31 to 24	Data 23 to 16	Data 15 to 8	Data 7 to 0	Assert	Assert	Assert	Assert

Table 12.7 16-Bit External Device/Big Endian Access and Data Alignment

		Data Bus				Strobe Signals			
Operation		**D31 to D24**	**D23 to D16**	**D15 to D8**	**D7 to D0**	**$\overline{WE3}$, DQMUU**	**$\overline{WE2}$, DQMUL**	**$\overline{WE1}$, DQMLU**	**$\overline{WE0}$, DQMLL**
Byte access at 0		—	—	Data 7 to 0	—			Assert	—
Byte access at 1		—	—	—	Data 7 to 0				Assert
Byte access at 2		—	—	Data 7 to 0	—			Assert	—
Byte access at 3		—	—	—	Data 7 to 0				Assert
Word access at 0		—	—	Data 15 to 8	Data 7 to 0			Assert	Assert
Word access at 2		—	—	Data 15 to 8	Data 7 to 0			Assert	Assert
Longword access at 0	1st time at 0	—	—	Data 31 to 24	Data 23 to 16			Assert	Assert
	2nd time at 2	—	—	Data 15 to 8	Data 7 to 0			Assert	Assert

Table 12.8 8-Bit External Device/Big Endian Access and Data Alignment

		Data Bus				Strobe Signals			
Operation		**D31 to D24**	**D23 to D16**	**D15 to D8**	**D7 to D0**	**$\overline{WE3}$, DQMUU**	**$\overline{WE2}$, DQMUL**	**$\overline{WE1}$, DQMLU**	**$\overline{WE0}$, DQMLL**
Byte access at 0		—	—	—	Data 7 to 0				Assert
Byte access at 1		—	—	—	Data 7 to 0				Assert
Byte access at 2		—	—	—	Data 7 to 0				Assert
Byte access at 3		—	—	—	Data 7 to 0				Assert
Word access at 0	1st time at 0	—	—	—	Data 15 to 8				Assert
	2nd time at 1	—	—	—	Data 7 to 0				Assert
Word access at 2	1st time at 2	—	—	—	Data 15 to 8				Assert
	2nd time at 3	—	—	—	Data 7 to 0				Assert
Longword access at 0	1st time at 0	—	—	—	Data 31 to 24				Assert
	2nd time at 1	—	—	—	Data 23 to 16				Assert
	3rd time at 2	—	—	—	Data 15 to 8				Assert
	4th time at 3	—	—	—	Data 7 to 0				Assert

Table 12.9 32-Bit External Device/Little Endian Access and Data Alignment

	Data Bus				Strobe Signals			
Operation	**D31 to D24**	**D23 to D16**	**D15 to D8**	**D7 to D0**	**$\overline{WE3}$, DQMUU**	**$\overline{WE2}$, DQMUL**	**$\overline{WE1}$, DQMLU**	**$\overline{WE0}$, DQMLL**
Byte access at 0	—	—	—	Data 7 to 0				Assert
Byte access at 1	—	—	Data 7 to 0	—			Assert	
Byte access at 2	—	Data 7 to 0	—	—		Assert		
Byte access at 3	Data 7 to 0	—	—	—	Assert			
Word access at 0	—	—	Data 15 to 8	Data 7 to 0			Assert	Assert
Word access at 2	Data 15 to 8	Data 7 to 0	—	—	Assert	Assert		
Longword access at 0	Data 31 to 24	Data 23 to 16	Data 15 to 8	Data 7 to 0	Assert	Assert	Assert	Assert

Table 12.10 16-Bit External Device/Little Endian Access and Data Alignment

		Data Bus				Strobe Signals			
Operation		**D31 to D24**	**D23 to D16**	**D15 to D8**	**D7 to D0**	**$\overline{WE3}$, DQMUU**	**$\overline{WE2}$, DQMUL**	**$\overline{WE1}$, DQMLU**	**$\overline{WE0}$, DQMLL**
Byte access at 0		—	—	—	Data 7 to 0				Assert
Byte access at 1		—	—	Data 7 to 0	—			Assert	
Byte access at 2		—	—	—	Data 7 to 0				Assert
Byte access at 3		—	—	Data 7 to 0	—			Assert	
Word access at 0		—	—	Data 15 to 8	Data 7 to 0			Assert	Assert
Word access at 2		—	—	Data 15 to 8	Data 7 to 0			Assert	Assert
Longword access at 0	1st time at 0	—	—	Data 15 to 8	Data 7 to 0			Assert	Assert
	2nd time at 2	—	—	Data 31 to 24	Data 23 to 16			Assert	Assert

Table 12.11 8-Bit External Device/Little Endian Access and Data Alignment

		Data Bus				Strobe Signals			
Operation		**D31 to D24**	**D23 to D16**	**D15 to D8**	**D7 to D0**	**$\overline{WE3}$, DQMUU**	**$\overline{WE2}$, DQMUL**	**$\overline{WE1}$, DQMLU**	**$\overline{WE0}$, DQMLL**
Byte access at 0		—	—	—	Data 7 to 0				Assert
Byte access at 1		—	—	—	Data 7 to 0				Assert
Byte access at 2		—	—	—	Data 7 to 0				Assert
Byte access at 3		—	—	—	Data 7 to 0				Assert
Word access at 0	1st time at 0	—	—	—	Data 7 to 0				Assert
	2nd time at 1	—	—	—	Data 15 to 8				Assert
Word access at 2	1st time at 2	—	—	—	Data 7 to 0				Assert
	2nd time at 3	—	—	—	Data 15 to 8				Assert
Longword access at 0	1st time at 0	—	—	—	Data 7 to 0				Assert
	2nd time at 1	—	—	—	Data 15 to 8				Assert
	3rd time at 2	—	—	—	Data 23 to 16				Assert
	4th time at 3	—	—	—	Data 31 to 24				Assert

12.3.2 Description of Areas

Area 0: Area 0 physical addresses A28 to A26 are 0'0. Addresses A31 to A29 are ignored and the address range is H'00000000 + H'20000000 × n – H'03FFFFFF + H'20000000 × n (n = 0 to 6 and n = 1 to 6 are the shadow spaces).

Ordinary memories such as SRAM, ROM, and burst ROM can be connected to this space. Byte, word, or longword can be selected as the bus width using external pins MD3 and MD4. When the area 0 space is accessed, a $\overline{CS0}$ signal is asserted. An $\overline{RD}$ signal that can be used as $\overline{OE}$ and the $\overline{WE0}$ to $\overline{WE3}$ signals for write control are also asserted. The number of bus cycles is selected between 0 and 10 wait cycles using the A0W2 to A0W0 bits of WCR2. In addition, any number of waits can be inserted in each bus cycle by means of the external wait pin ($\overline{WAIT}$).
When the burst function is used, the bus cycle pitch of the burst cycle is determined within a range of 2 to 10 according to the number of waits.

Area 1: Area 1 physical addresses A28 to A26 are 001. Addresses A31 to A29 are ignored and the address range is H'04000000 + H'20000000 × n – H'07FFFFFF + H'20000000 × n (n = 0 to 6 and n = 1 to 6 are the shadow spaces).

Area 1 is the area specifically for the internal peripheral modules. The external memories cannot be connected.

Control registers of peripheral modules shown below are mapped to this area 1. Their addresses are physical address, to which logical addresses can be mapped with the MMU enabled:

> DMAC, PORT, SCIF, ADC, DAC, LCDC, PCC, SIOF, AFEIF, USBF, USBH, INTC (except INTEVT, IPRA, IPRB)

Those registers must be set not to be cached.

Area 2: Area 2 physical addresses A28 to A26 are 010. Addresses A31 to A29 are ignored and the address range is H'08000000 + H'20000000 × n – H'0BFFFFFF + H'20000000 × n (n = 0 to 6 and n = 1 to 6 are the shadow spaces).

Ordinary memories like SRAM and ROM, as well as synchronous DRAM, can be connected to this space. Byte, word, or longword can be selected as the bus width using the A2SZ1 to A2SZ0 bits of BCR2 for ordinary memory.

When the area 2 space is accessed, a $\overline{CS2}$ signal is asserted. When ordinary memories are connected, an $\overline{RD}$ signal that can be used as $\overline{OE}$ and the $\overline{WE0}$ to $\overline{WE3}$ signals for write control are also asserted and the number of bus cycles is selected between 0 and 3 wait cycles using the A2W1 to A2W0 bits of WCR2. In addition, any number of waits can be inserted in each bus cycle by means of the external wait pin ($\overline{WAIT}$) only when the ordinary memories are connected.

When synchronous DRAM is connected, the $\overline{RAS3}$ signal, $\overline{CAS}$ signal, RD/$\overline{WR}$ signal, and byte controls DQMHH, DQMHL, DQMLH, and DQMLL are all asserted and addresses multiplexed. Control of $\overline{RAS3}$, $\overline{CAS}$, data timing, and address multiplexing is set with MCR.

Area 3: Area 3 physical addresses A28 to A26 are 011. Addresses A31 to A29 are ignored and the address range is H'0C000000 + H'20000000 × n to H'0FFFFFFF + H'20000000 × n (n = 0 to 6 and n = 1 to 6 are the shadow spaces).

Ordinary memories like SRAM and ROM, as well as synchronous DRAM, can be connected to this space. Byte, word or longword can be selected as the bus width using the A3SZ1 to A3SZ0 bits of BCR2 for ordinary memory. When area 3 space is accessed, $\overline{CS3}$ is asserted.

When ordinary memories are connected, an $\overline{RD}$ signal that can be used as $\overline{OE}$ and the $\overline{WE0}$ to $\overline{WE3}$ signals for write control are asserted and the number of bus cycles is selected between 0 and 3 wait cycles using the A3W1 to A3W0 bits of WCR2. In addition, any number of waits can be inserted in each bus cycle by means of the external wait pin ($\overline{WAIT}$).only when the ordinary memories are connected.

When synchronous DRAM is connected, the $\overline{RAS3}$ signal, $\overline{CAS}$ signal, RD/$\overline{WR}$ signal, and byte controls DQMHH, DQMHL, DQMLH, and DQMLL are all asserted and addresses multiplexed.

Area 4: Area 4 physical addresses A28 to A26 are 1'0. AddressesA31 to A29 are ignored and the address range is H'10000000 + H'20000000 × n – H'13FFFFFF + H'20000000 × n (n = 0 to 6 and n = 1 to 6 are the shadow spaces).

Only ordinary memories like SRAM and ROM can be connected to this space. Byte, word, or longword can be selected as the bus width using the A4SZ1 to A4SZ0 bits of BCR2. When the area 4 space is accessed, a $\overline{CS4}$ signal is asserted. An $\overline{RD}$ signal that can be used as $\overline{OE}$ and the $\overline{WE0}$ to $\overline{WE3}$ signals for write control are also asserted. The number of bus cycles is selected between 0 and 10 wait cycles using the A4W2 to A4W0 bits of WCR2. In addition, any number of waits can be inserted in each bus cycle by means of the external wait pin ($\overline{WAIT}$).

Area 5: Area 5 physical addresses A28 to A26 are 101. Addresses A31 to A29 are ignored and the address range is the 64 Mbytes at H'14000000 + H'20000000 × n to H'17FFFFFF + H'20000000 × n (n = 0 to 6 and n = 1 to 6 are the shadow spaces).

Ordinary memories like SRAM and ROM as well as burst ROM and PCMCIA interfaces can be connected to this space. When the PCMCIA interface is used, the IC memory card interface address range-comprises the 32 Mbytes at H'14000000 + H'20000000 × n to H'15FFFFFF + H'20000000 × n (n = 0 to 6 and n = 1 to 6 are the shadow spaces), and the I/O card interface address range-comprises the 32 Mbytes at H'16000000 + H'20000000 × n to H'17FFFFFF + H'20000000 × n (n = 0 to 6 and n = 1 to 6 are the shadow spaces).

For ordinary memory and burst ROM, byte, word, or longword can be selected as the bus width using the A5SZ1 to A5SZ0 bits of BCR2. For the PCMCIA interface, byte, and word can be selected as the bus width using the A5SZ1 to A5SZ0 bits of BCR2.

When the area 5 space is accessed and ordinary memory is connected, a $\overline{CS5}$ signal is asserted. An $\overline{RD}$ signal that can be used as $\overline{OE}$ and the $\overline{WE0}$ to $\overline{WE3}$ signals for write control are also asserted. When the PCMCIA interface is used, the $\overline{CE1A}$ signal, $\overline{CE2A}$ signal, $\overline{RD}$ signal as $\overline{OE}$ signal, and $\overline{WE}$, $\overline{ICIORD}$, $\overline{ICIOWR}$ signal are asserted.

The number of bus cycles is selected between 0 and 10 wait cycles using the A5W2 to A5W0 bits of WCR2. With the PCMCIA interface, from 0 to 38 wait cycles can be selected using the A5W2 to A5W0 bits of WCR2 and the A5W3 bit of PCR. In addition, any number of waits can be inserted in each bus cycle by means of the external wait pin ($\overline{WAIT}$).

When a burst function is used, the bus cycle pitch of the burst cycle is determined within a range of 2 to 11 (2 to 39 for the PCMCIA interface) according to the number of waits. The setup and hold times of address/$\overline{CS5}$ for the read/write strobe signal can be set in the range 0.5 to 7.5 cycles using A5TED2 to A5TED0 and A5TEH2 to A5TEH0 bits of the PCR register. (Single-cycle units)

Area 6: Area 6 physical addresses A28 to A26 are 110. Address A31 to A29 are ignored and the address range is the 64 Mbytes at H'18000000 + H'20000000 × n to H'1BFFFFFF + H'20000000 × n (n = 0 to 6 and n = 1 to 6 are the shadow spaces).

Ordinary memories like SRAM and ROM as well as burst ROM and PCMCIA interfaces can be connected to this space. When the PCMCIA interface is used, the IC memory card interface address range is 32 Mbytes at H'18000000 + H'20000000 × n to H'19FFFFFF + H'2000'000 × n and 'he I/O card interface address range is 32 Mbytes at H'1A000000 + H'20000000 × n to H'1BFFFFFF + H'20000000 × n (n = 0 to 6 and n = 1 to 6 are the shadow spaces).

For ordinary memory and burst ROM, byte, word, or longword can be selected as the bus width using the A6SZ1 to A6SZ0 bits of BCR2. For the PCMCIA interface, byte, and word can be selected as the bus width using the A6SZ1 to A6SZ0 bits of BCR2.

When the area 6 space is accessed and ordinary memory is connected, a $\overline{CS6}$ signal is asserted. An $\overline{RD}$ signal that can be used as $\overline{OE}$ and the $\overline{WE0}$ to $\overline{WE3}$ signals for write control are also asserted. When the PCMCIA interface is used, the $\overline{CI1B}$ signal, $\overline{CE2B}$ signal, $\overline{RD}$ signal as $\overline{OE}$ signal, and $\overline{WE}$, $\overline{ICIORD}$, and $\overline{ICIOWR}$ signals are asserted.

The number of bus cycles is selected between 0 to 10 wait cycles using the A6W2 to A6W0 bits of WCR2. With the PCMCIA interface, from 0 to 38 wait cycles can be selected using the A6W2 to A6W0 bits of WCR2 and the A6W3 bit of PCR. In addition, any number of waits can be inserted in each bus cycle by means of the external wait pin ($\overline{WAIT}$). The bus cycle pitch of the burst cycle

is determined within a range of 2 to 11 (2 to 39 for the PCMCIA interface) according to the number of waits. The setup and hold times of address/$\overline{CS6}$ for the read/write strobe signals can be set in the range 0.5 to 7.5 cycles using A6TED2 to A6TED0 and A6TEH2 to A6TEH0 bits of the PCR register. (Single-cycle units)

12.3.3 Basic Interface

Basic Timing: The basic interface of this LSI uses strobe signal output in consideration of the fact that mainly static RAM will be directly connected. Figure 12.5 shows the basic timing of normal space accesses. A no-wait normal access is completed in two cycles. The $\overline{BS}$ signal is asserted for one cycle to indicate the start of a bus cycle. The $\overline{CSn}$ signal is negated on the T2 clock falling edge to secure the negation period. Therefore, in case of access at minimum pitch, there is a half-cycle negation period.

There is no access size specification when reading. The correct access start address is output in the least significant bit of the address, but since there is no access size specification, 32 bits are always read in case of a 32-bit device, and 16 bits in case of a 16-bit device. When writing, only the $\overline{WE}$ signal for the byte to be written is asserted. For details, see section 12.3.1, Endian/Access Size and Data Alignment.

Read/write for cache fill or write-back follows the set bus width and transfers a total of 16 bytes continuously. The bus is not released during this transfer. For cache misses that occur during byte or word operand accesses or branching to odd word boundaries, the fill is always performed by longword accesses on the chip-external interface. Write-through-area write access and non-cacheable read/write access are based on the actual address size.

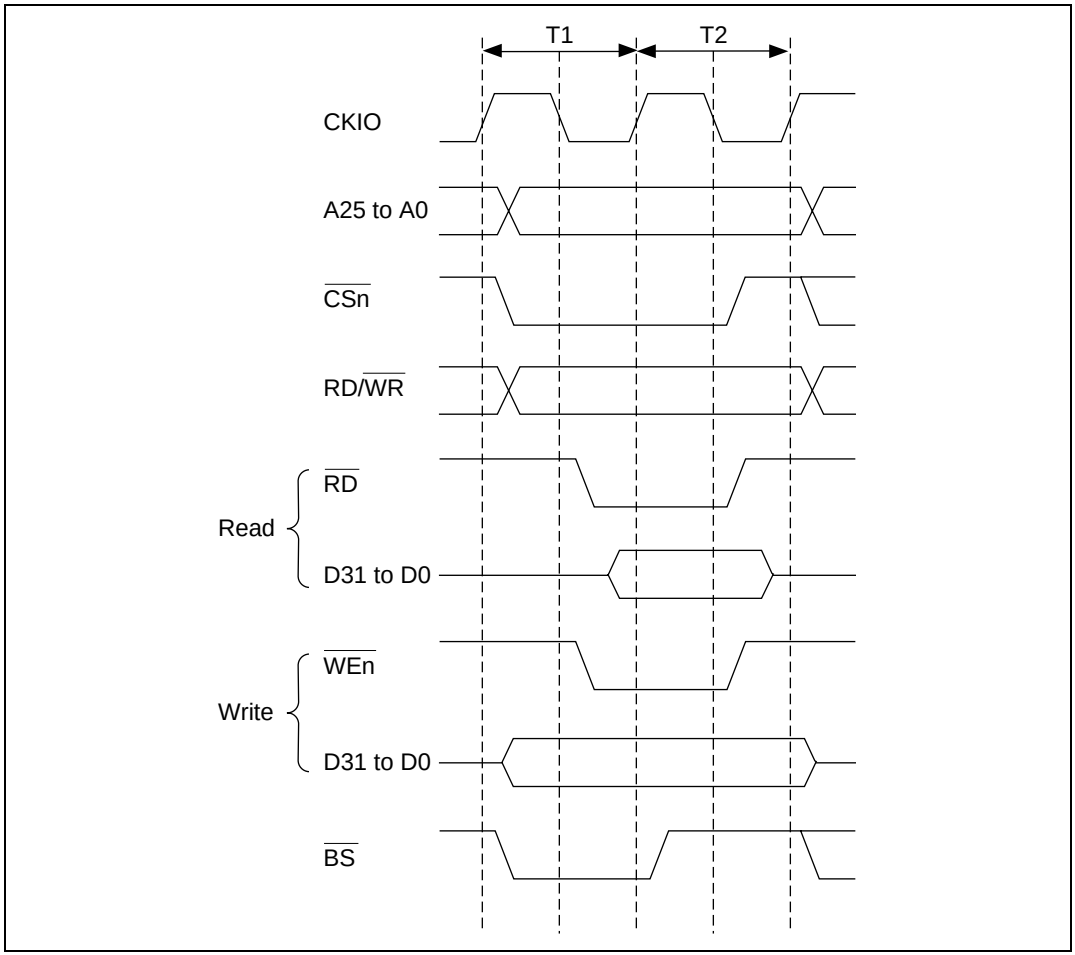


Figure 12.5 Basic Timing of Basic Interface

Figures 12.6, 12.7, and 12.8 show examples of connection to 32, 16, and 8-bit data-width static RAM, respectively.

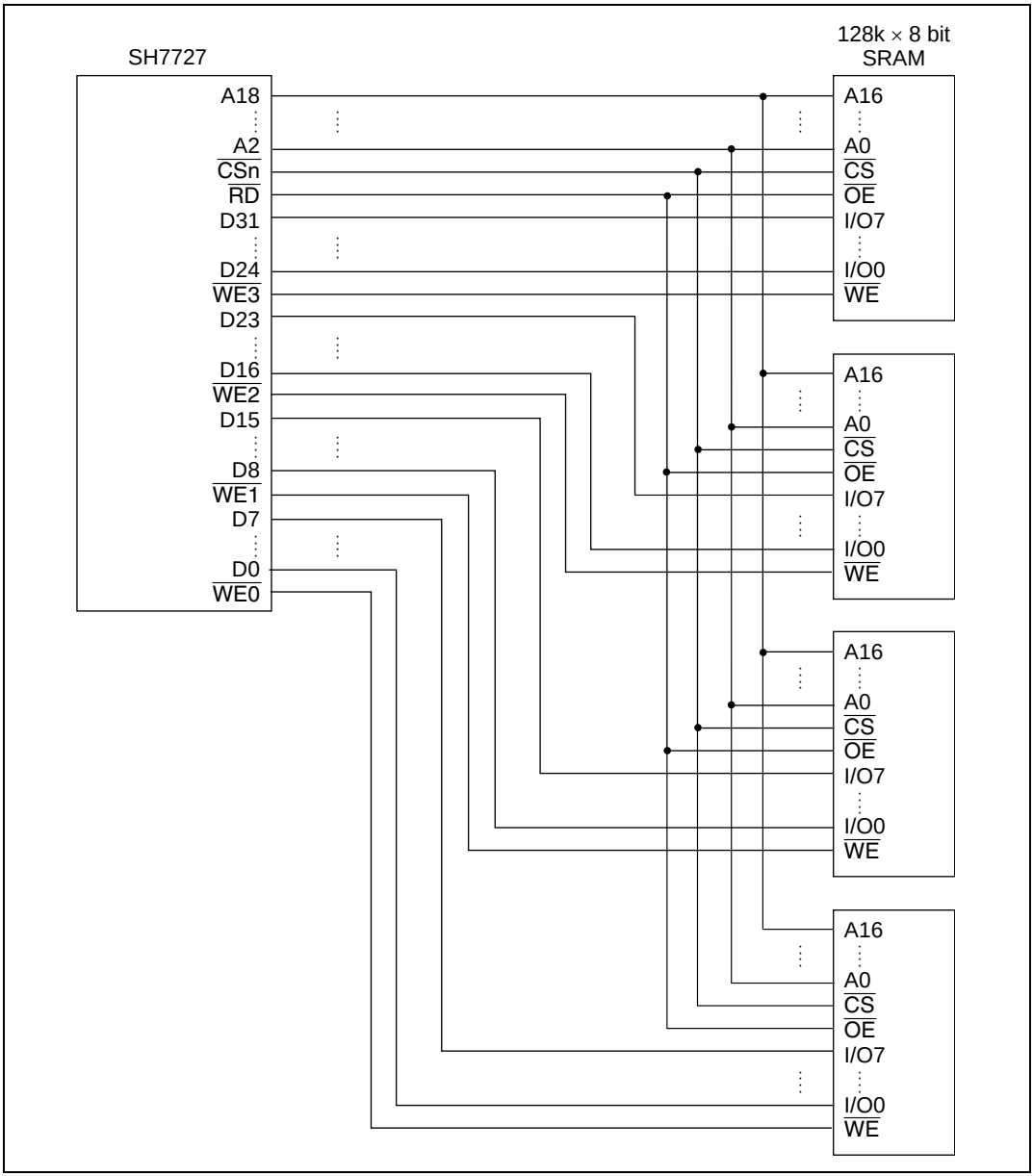


Figure 12.6 Example of 32-Bit Data-Width Static RAM Connection

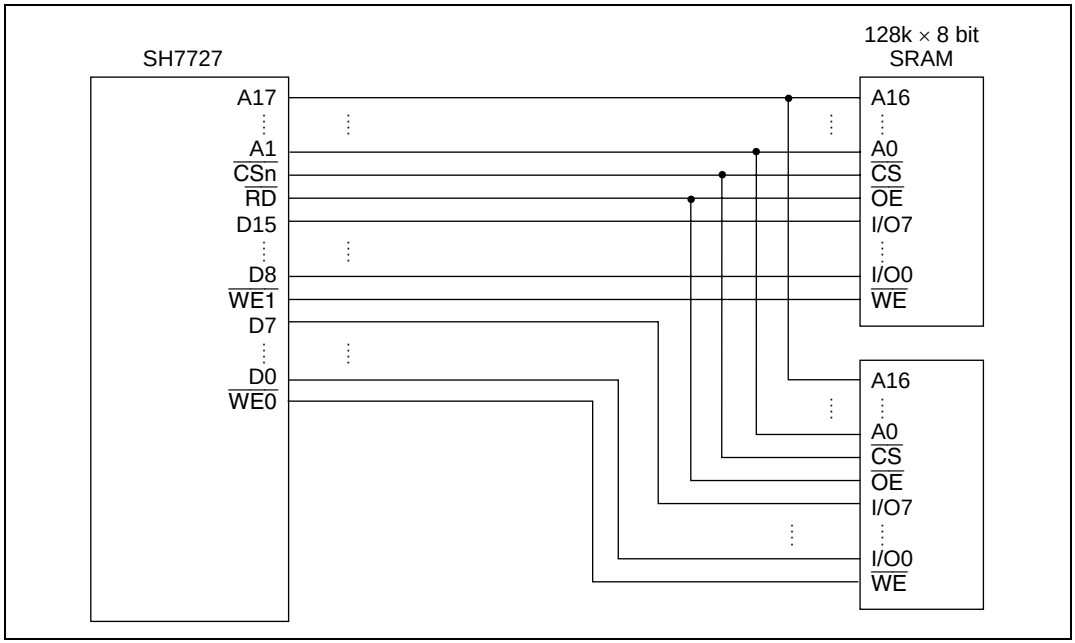


Figure 12.7 Example of 16-Bit Data-Width Static RAM Connection

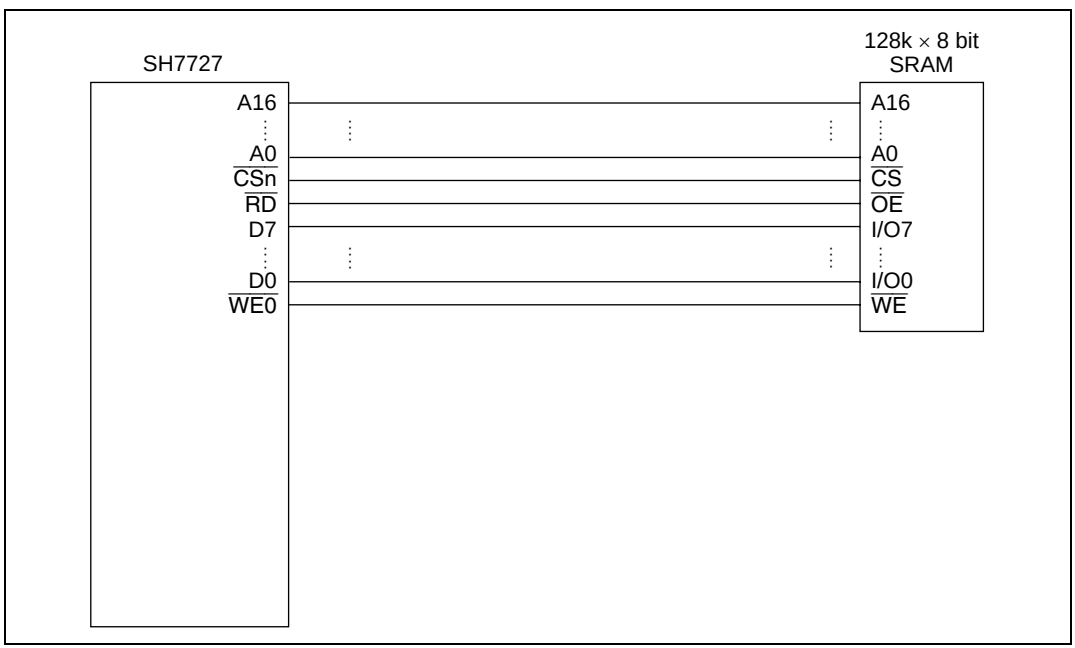


Figure 12.8 Example of 8-Bit Data-Width Static RAM Connection

Wait State Control: Wait state insertion on the basic interface can be controlled by the WCR2 settings. If the WCR2 wait specification bits corresponding to a particular area are not zero, a software wait is inserted in accordance with that specification. For details, see section 12.2.4, Wait State Control Register 2 (WCR2).

The specified number of Tw cycles are inserted as wait cycles using the basic interface wait timing shown in figure 12.9.

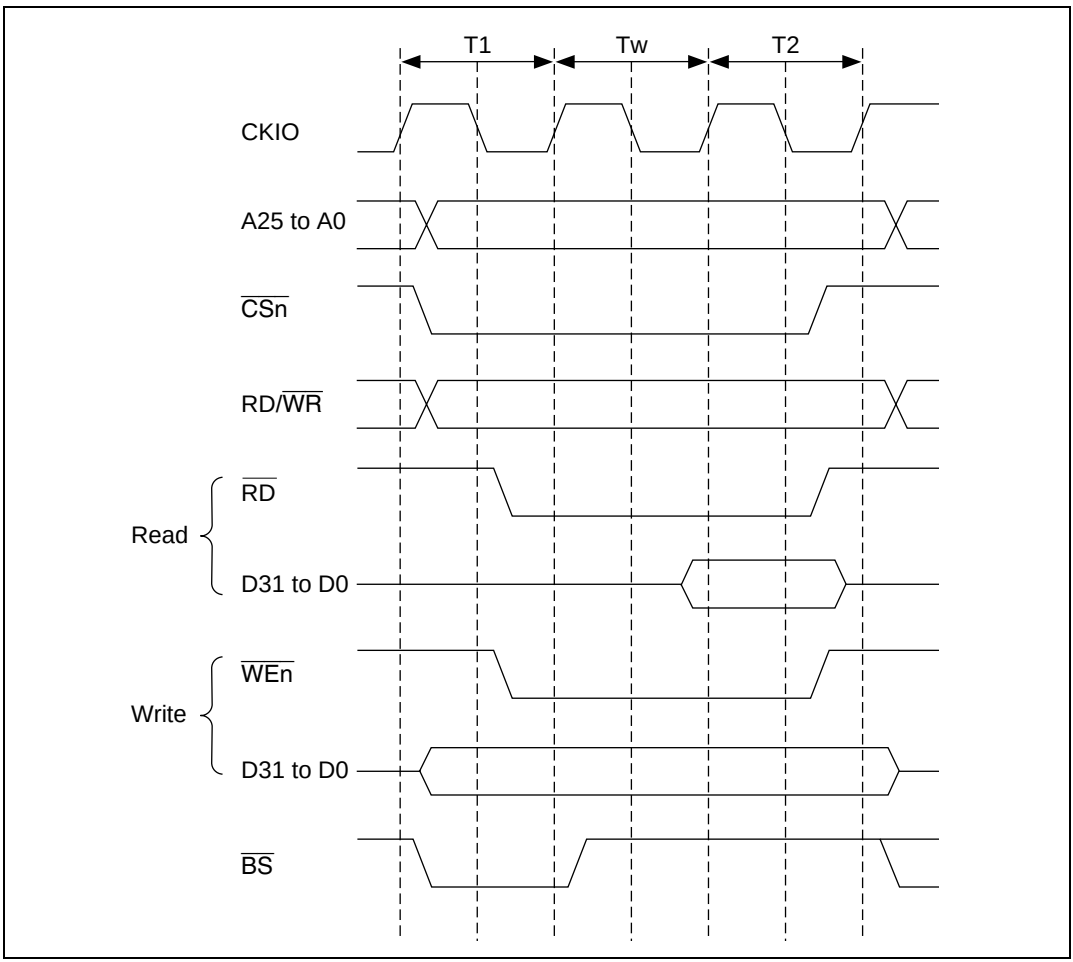


Figure 12.9 Basic Interface Wait Timing (Software Wait Only)

When software wait insertion is specified by WCR2, the external wait input WAIT signal is also sampled. TO input low level signal to WAIT, set the WAITSEL bit of the WCR1 register to 1. WAIT pin sampling is shown in figure 12.10. A 2-cycle wait is specified as a software wait.

Sampling is performed at the transition from the Tw state to the T2 state; therefore, if the $\overline{\text{WAIT}}$ signal has no effect if asserted in the T1 cycle or the first Tw cycle.

The $\overline{\text{WAIT}}$ signal is sampled at the falling edge of the clock. If the setup time and hold times with respect to the falling edge of the clock are not satisfied, the value sampled at the next falling edge is used..

However, the $\overline{\text{WAIT}}$ signal is ignored in the following three cases:

- When writing to an external address area using DMA 16-byte transfer in dual address mode
- When transferring data from a DACK-equipped external device to an external address area using DMA 16-byte transfer in dual address mode
- During cache write-back access

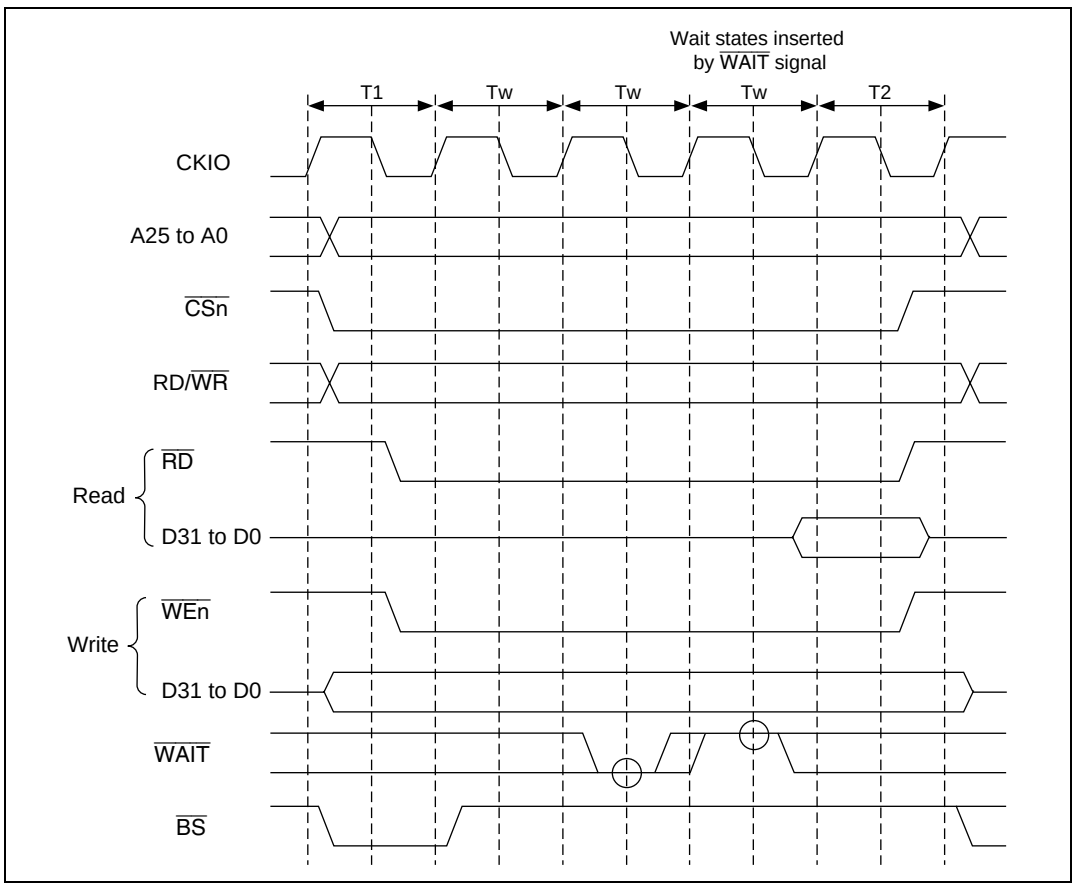


Figure 12.10 Basic Interface Wait State Timing (Wait State Insertion by $\overline{\text{WAIT}}$ Signal WAITSEL = 1)

12.3.4 Synchronous DRAM Interface

Synchronous DRAM Direct Connection: Since synchronous DRAM can be selected by the $\overline{CS}$ signal, physical space areas 2 and 3 can be connected using $\overline{RAS}$ and other control signals in common. If the memory type bits (DRAMTP2 to DRAMTP0) in BCR1 are set to 010, area 2 is ordinary memory space and area 3 is synchronous DRAM space; if set to 011, areas 2 and 3 are both synchronous DRAM space.

With this LSI, burst length 1 burst read/single write mode is supported as the synchronous DRAM operating mode. A data bus width of 16 or 32 bits can be selected. A 16-bit burst transfer is performed in a cache fill/write-back cycle, and only one access is performed in a write-through area write or a non-cacheable area read/write.

The control signals for direct connection of synchronous DRAM are $\overline{RAS3}$, $\overline{CAS}$, RD/$\overline{WR}$, $\overline{CS2}$ or $\overline{CS3}$, DQMUU, DQMUL, DQMLU, DQMLL, and CKE. All the signals other than $\overline{CS2}$ and $\overline{CS3}$ are common to all areas, and signals other than CKE are valid and fetched to the synchronous DRAM only when $\overline{CS2}$ or $\overline{CS3}$ is asserted. Synchronous DRAM can therefore be connected in parallel to a number of areas. CKE is negated (low) only when self-refreshing is performed, and is always asserted (high) at other times.

Commands for synchronous DRAM are specified by $\overline{RAS3}$, $\overline{CAS}$, RD/$\overline{WR}$, and special address signals. The commands are NOP, auto-refresh (REF), self-refresh (SELF), precharge all banks (PALL), row address strobe bank active (ACTV), read (READ), read with precharge (READA), write (WRIT), write with precharge (WRITA), and mode register write (MRS).

Byte specification is performed by DQMUU, DQMUL, DQMLU, and DQMLL. A read/write is performed for the byte for which the corresponding DQM is low. In big-endian mode, DQMUU specifies an access to address 4n, and DQMLL specifies an access to address 4n + 3. In little-endian mode, DQMUU specifies an access to address 4n + 3, and DQMLL specifies an access to address 4n.

Figures 12.11 and 12.12 show examples of the connection of two 1M × 16-bit × 4-bank synchronous DRAMs and one 1M × 16-bit × 4-bank synchronous DRAM, respectively.

CKIO and CKIO2 are the clock signals that can be input to the synchronous DRAM. When using multiple synchronous DRAMs, use either CKIO or CKIO2, but not both. Also to prevent big signal delays due to overloading, design the board so that the load capacity is 50 pF or less.

Aim to ensure wiring lengths are equal and avoid chaining the clock wiring to the synchronous DRAMs.

CKIO has higher drive capacity than CKIO2. CKIO is suitable for driving heavy load, while CKIO2 offers higher resistance to EMI noise and reflection. However, as the degree of these characteristics vary depending on applications, their usage is not specified.

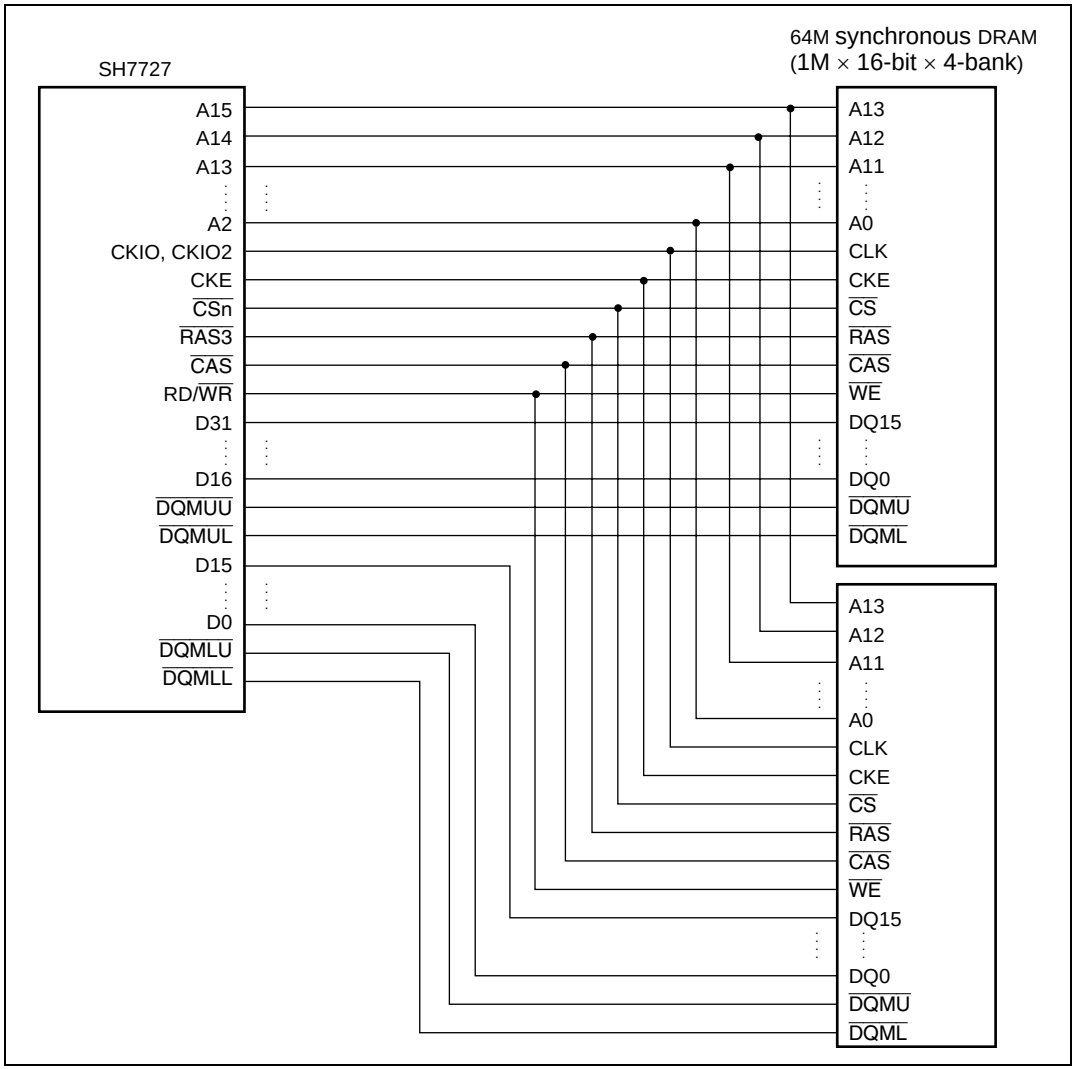


Figure 12.11 Example of 64-Mbit Synchronous DRAM Connection (32-Bit Bus Width)

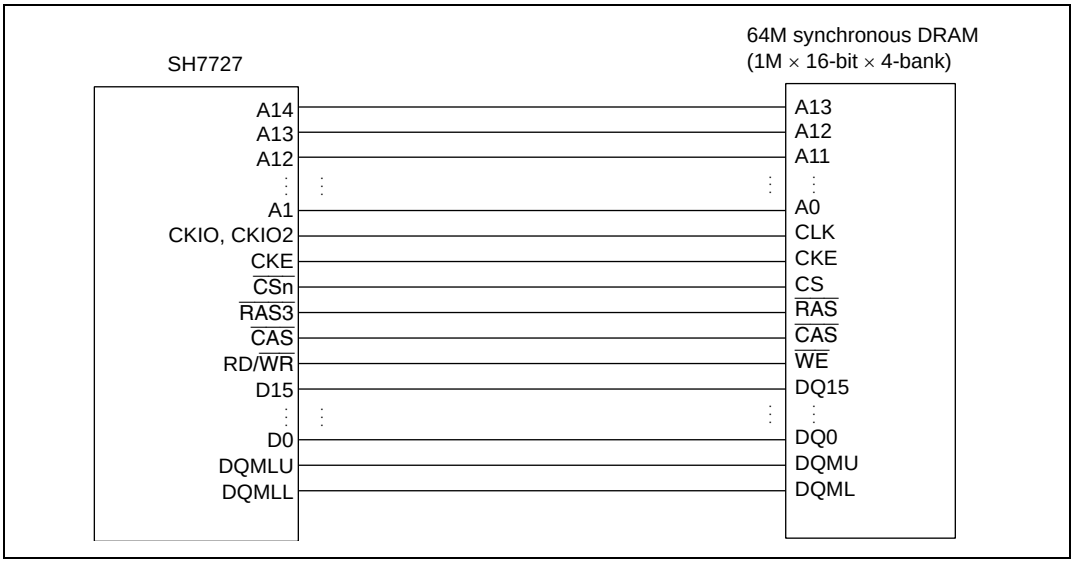


Figure 12.12 Example of 64-Mbit Synchronous DRAM (16-Bit Bus Width)

Address Multiplexing: Synchronous DRAM can be connected without external multiplexing circuitry in accordance with the address multiplex specification bits AMX3 to AMX0 in MCR. Table 12.12 shows the relationship between the address multiplex specification bits and the bits output at the address pins. Table 12.13 shows the relationship between LSI address pins and synchronous DRAM address pins.

A25 to A17 and A0 are not multiplexed; the original values are always output at these pins.

When A0, the LSB of the synchronous DRAM address, is connected to this LSI, it performs longword address specification. Connection should therefore be made in the following order: with a 32-bit bus width, connect pin A0 of the synchronous DRAM to pin A2 of this LSI, then connect pin A1 to pin A3; with a 16-bit bus width, connect pin A0 of the synchronous DRAM to pin A1 of this LSI, then connect pin A1 and pin A2.

Table 12.12 Relationship between Synchronous DRAM type, bus width and AMX

Bus Width	Memory Type		Setting				Output Timing	External Address Pin								
			AMX3	AMX2	AMX1	AMX0		A1 to A8	A9	A10	A11	A12	A13	A14	A15	A16
32 bits	256 Mbits	4M × 16-bit × 4-bank*	1	1	0	1	Column address	A1–A8	A9	A10	A11	L/H	A13	A23	A24	A25
							Row address	A10–A17	A18	A19	A20	A21	A22	A23	A24	A25
	128 Mbits	1M × 32-bit × 4-bank*	0	1	0	0	Column address	A1–A8	A9	A10	A11	L/H	A13	A22	A23	A16
							Row address	A9–A16	A17	A18	A19	A20	A21	A22	A23	A16
		2M × 16-bit × 4-bank*	0	1	0	1	Column address	A1–A8	A9	A10	A11	L/H	A13	A23	A24	A16
							Row address	A10–A17	A18	A19	A20	A21	A22	A23	A24	A16
		4M × 8-bit × 4-bank*	0	1	1	0	Column address	A1–A8	A9	A10	A11	L/H	A13	A24	A25	A16
							Row address	A11–A18	A19	A20	A21	A22	A23	A24	A25	A16
	64 Mbits	1M × 16-bit × 4-bank*	0	1	0	0	Column address	A1–A8	A9	A10	A11	L/H	A13	A22	A23	A16
							Row address	A9–A16	A17	A18	A19	A20	A21	A22	A23	A16
		2M × 8-bit × 4-bank*	0	1	0	1	Column address	A1–A8	A9	A10	A11	L/H	A13	A23	A24	A16
							Row address	A10–A17	A18	A19	A20	A21	A22	A23	A24	A16
		4M × 4-bit × 4-bank*	0	1	1	0	Column address	A1–A8	A9	A10	A11	L/H	A13	A24	A25	A16
							Row address	A11–A18	A19	A20	A21	A22	A23	A24	A25	A16
		512K × 32-bit × 4-bank	0	1	1	1	Column address	A1–A8	A9	A10	A11	L/H	A21	A22	A15	A16
							Row address	A9–A16	A17	A18	A19	A20	A21	A22	A23	A16
16 bits	512 Mbits	8M × 16-bit × 4-bank*	1	1	1	0	Column address	A1–A8	A9	A10	L/H	A12	A13	A24	A25	A16
							Row address	A11–A18	A19	A20	A21	A22	A23	A24	A25	A16
	256 Mbits	4M × 16-bit × 4-bank	1	1	0	1	Column address	A1–A8	A9	A10	L/H	A12	A22	A23	A24	A16
							Row address	A10–A17	A18	A19	A20	A21	A22	A23	A24	A16
		8M × 8-bit × 4-bank*	1	1	1	0	Column address	A1–A8	A9	A10	L/H	A12	A23	A24	A25	A16
							Row address	A11–A18	A19	A20	A21	A22	A23	A24	A25	A16
	128 Mbits	2M × 16-bit × 4-bank	0	1	0	1	Column address	A1–A8	A9	A10	L/H	A12	A22	A23	A24	A16
							Row address	A10–A17	A18	A19	A20	A21	A22	A23	A24	A16
	64 Mbits	1M × 16-bit × 4-bank	0	1	0	0	Column address	A1–A8	A9	A10	L/H	A12	A21	A22	A15	A16
							Row address	A9–A16	A17	A18	A19	A20	A21	A22	A23	A16
		2M × 8-bit × 4-bank	0	1	0	1	Column address	A1–A8	A9	A10	L/H	A12	A22	A23	A15	A16
							Row address	A10–A17	A18	A19	A20	A21	A22	A23	A24	A16

Notes: * L/H is a bit used to specify commands. It is fixed to L or H by the access mode.
: Bank address

Table 12.13 Relationship between LSI Address Pins and Synchronous DRAM Address Pins

SH7727 Address Pin	RAS Cycle	CAS Cycle	SDRAM Address Pin	Function
A16	A24	A16	A14	Address
A15	A23	A23	A13	BANK select bank address
A14	A22	A22	A12	
A13	A21	A13	A11	Address
A12	A20	L/H	A10	Address precharge specification
A11	A19	A11	A9	Address
A10	A18	A10	A8	
A9	A17	A9	A7	
A8	A16	A8	A6	
A7	A15	A7	A5	
A6	A14	A6	A4	
A5	A13	A5	A3	
A4	A12	A4	A2	
A3	A11	A3	A1	
A2	A10	A2	A0	
A1	A9	A1	Unused	
A0	A0	A0	Unused	

Burst Read: In the example in figure 12.13 it is assumed that four 2M × 8-bit synchronous DRAMs are connected and a 32-bit data width is used, and the burst length is 1. Following the Tr cycle in which ACTV command output is performed, a READ command is issued in the Tc1, Tc2, and Tc3 cycles, and a READA command in the Tc4 cycle, and the read data is accepted on the rising edge of the external command clock (CKIO) from cycle Td1 to cycle Td4. The Tpc cycle is used to wait for completion of auto-precharge based on the READA command inside the synchronous DRAM; no new access command can be issued to the same bank during this cycle, but access to synchronous DRAM for another area is possible. In this LSI, the number of Tpc cycles is determined by the TPC bit specification in MCR, and commands cannot be issued for the same synchronous DRAM during this interval.

To connect low-speed synchronous DRAM, the cycle can be extended by setting WCR2 and MCR bits. The number of cycles from the ACTV command output cycle, Tr, to the READ command output cycle, Tc1, can be specified by the RCD bit in MCR, with a values of 0 to 3 specifying 1 to 4 cycles, respectively. In case of 2 or more cycles, a Trw cycle, in which an NOP command is

issued for the synchronous DRAM, is inserted between the Tr cycle and the Tc cycle. The number of cycles from READ and READA command output cycles Tc1 to Tc4 to the first read data latch cycle, Td1, can be specified as 1 to 3 cycles independently for areas 2 and 3 by means of A2W1 and A2W0 or A3W1 and A3W0 in WCR2. This number of cycles corresponds to the number of synchronous DRAM CAS latency cycles.

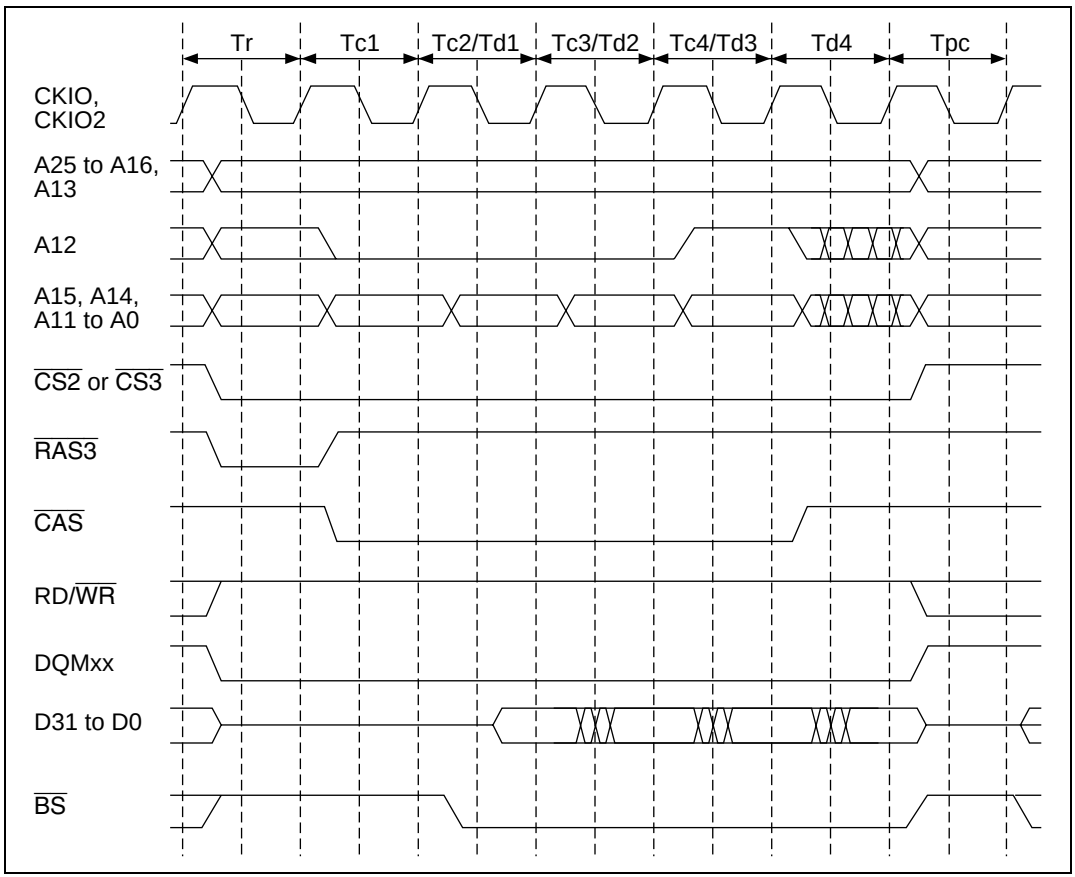


Figure 12.13 Basic Timing for Synchronous DRAM Burst Read

Figure 12.14 shows the burst read timing when RCD is set to 1, A3W1 and A3W0 are set to 10, and TPC is set to 1.

The BS cycle, which is asserted for one cycle at the start of a bus cycle for normal access space, is asserted in each of cycles Td1 to Td4 in a synchronous DRAM cycle. When a burst read is performed, the address is updated each time $\overline{CAS}$ is asserted. As the unit of burst transfer is 16 bytes, address updating is performed for A3 and A2 only. The order of access is as follows: in a

fill operation in the event of a cache miss, the missed data is read first, then 16-byte boundary data including the missed data is read in wraparound mode.

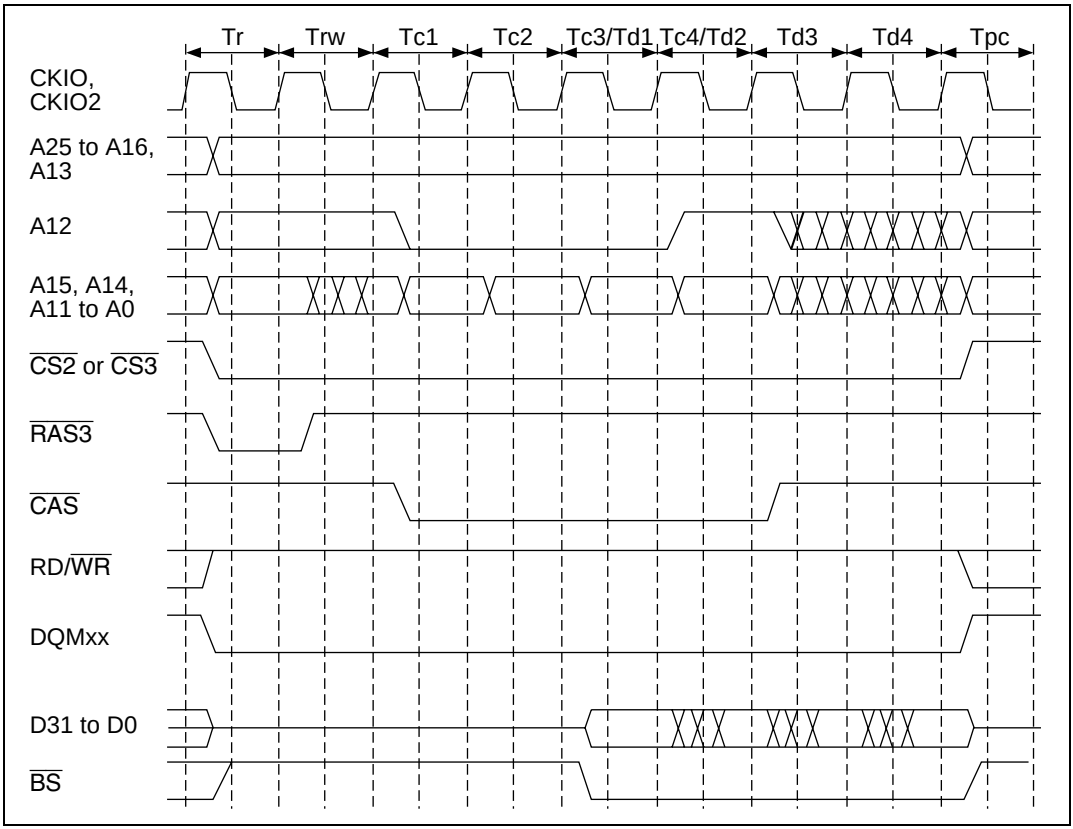


Figure 12.14 Synchronous DRAM Burst Read Wait Specification Timing

Single Read: Figure 12.15 shows the timing when a single address read is performed. As the burst length is set to 1 in synchronous DRAM burst read/single write mode, only the required data is output. Consequently, no unnecessary bus cycles are generated even when a cache-through area is accessed.

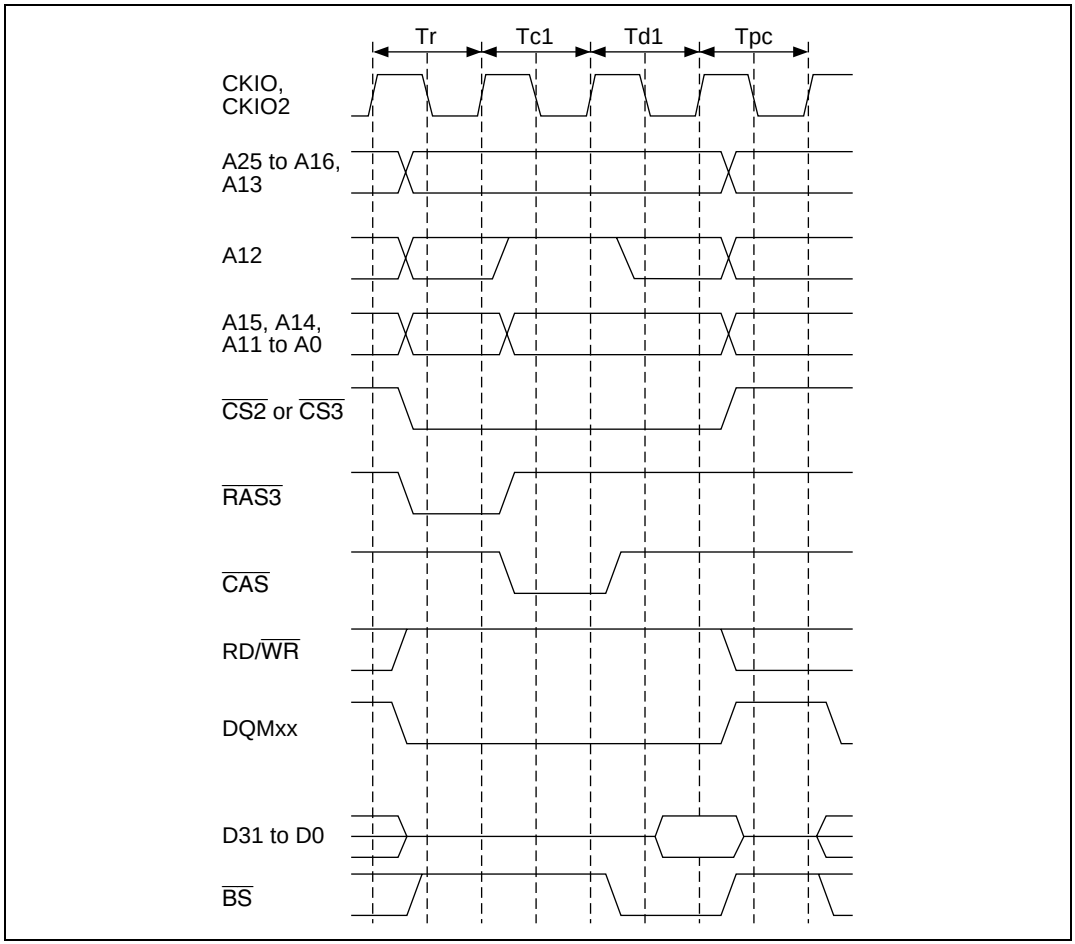


Figure 12.15 Basic Timing for Synchronous DRAM Single Read

Burst Write: The timing chart for a burst write is shown in figure 12.16. In this LSI, a burst write occurs only in the event of cache write-back. In a burst write operation, following the Tr cycle in which ACTV command output is performed, a WRIT command is issued in the Tc1, Tc2, and Tc3 cycles, and a WRITA command that performs auto-precharge is issued in the Tc4 cycle. In the write cycle, the write data is output at the same time as the write command. In case of the write with auto-precharge command, precharging of the relevant bank is performed in the synchronous DRAM after completion of the write command, and therefore no command can be issued for the same bank until precharging is completed. Consequently, in addition to the precharge wait cycle, Tpc, used in a read access, cycle Trwl is also added as a wait interval until precharging is started following the write command. Issuance of a new command for the same bank is postponed during this interval. The number of Trwl cycles can be specified by the TRWL bit in MCR.

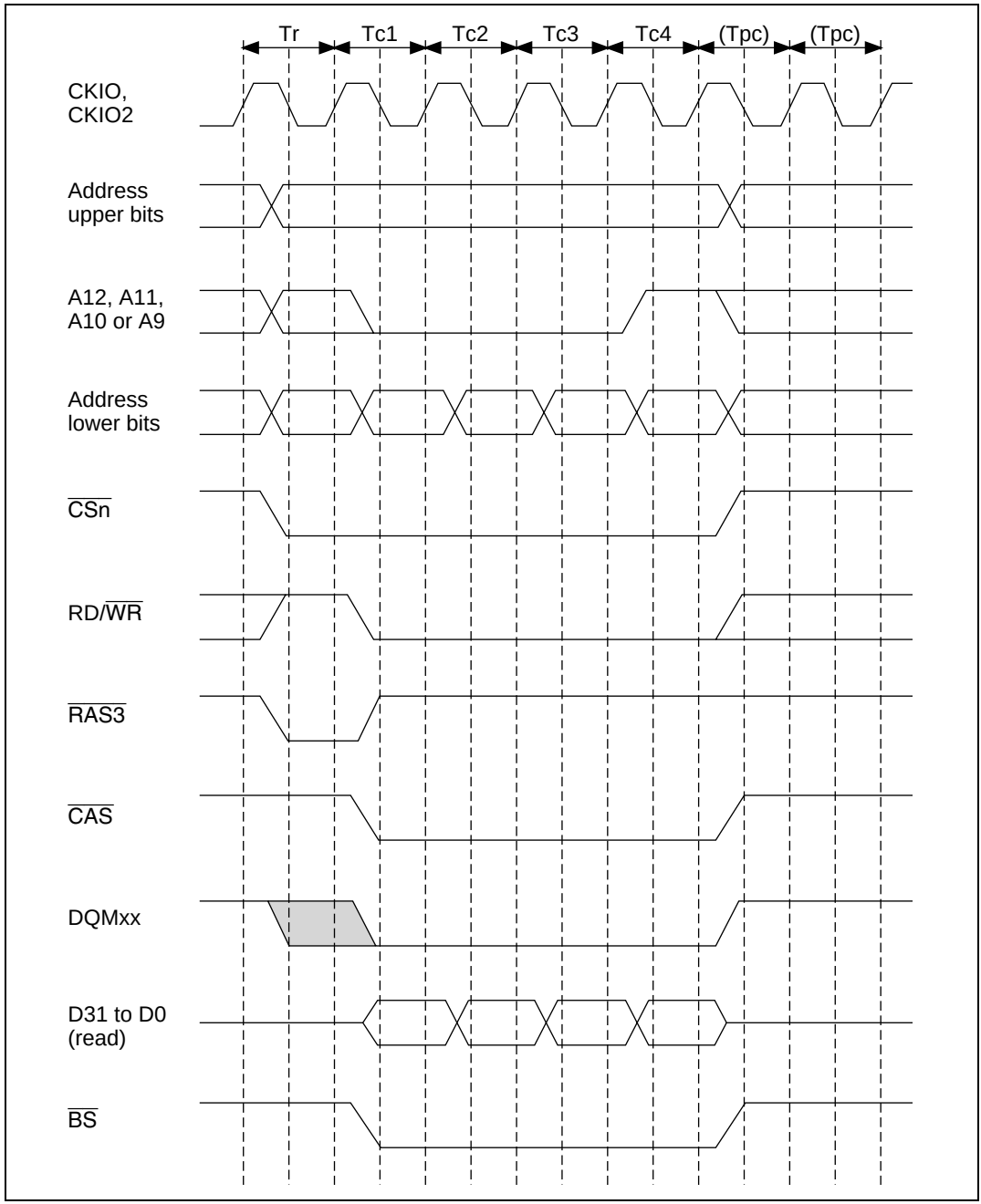


Figure 12.16 Basic Timing for Synchronous DRAM Burst Write

Single Write: The basic timing chart for write access is shown in figure 12.17. In a single write operation, following the Tr cycle in which ACTV command output is performed, a WRITA command that performs auto-precharge is issued in the Tc1 cycle. In the write cycle, the write data is output at the same time as the write command. In case of the write with auto-precharge command, precharging of the relevant bank is performed in the synchronous DRAM after completion of the write command, and therefore no command can be issued for the same bank until precharging is completed. Consequently, in addition to the precharge wait cycle, Tpc, used in a read access, cycle Trwl is also added as a wait interval until precharging is started following the write command. Issuance of a new command for the same bank is postponed during this interval. The number of Trwl cycles can be specified by the TRWL bit in MCR.

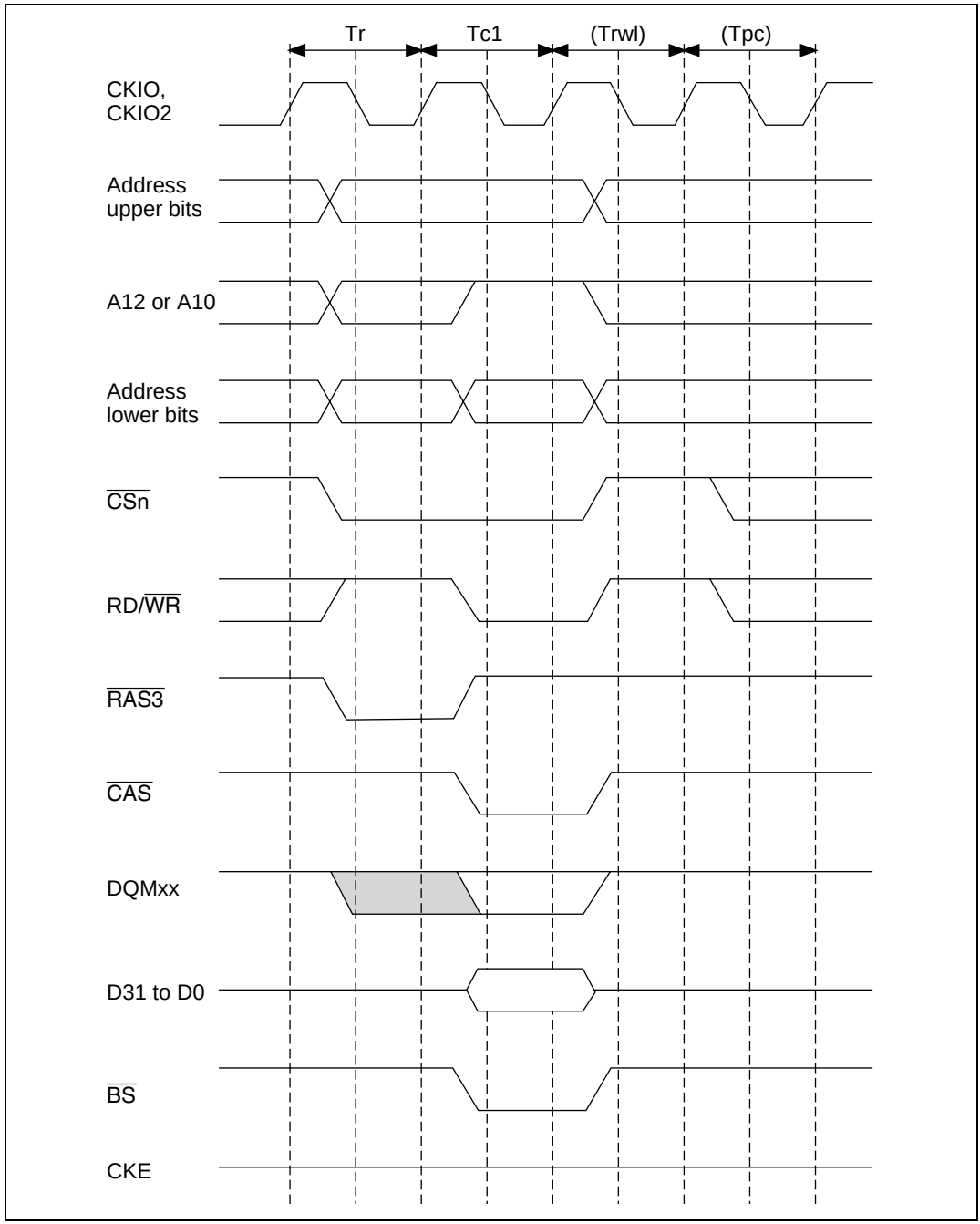


Figure 12.17 Basic Timing for Synchronous DRAM Single Write

Refreshing: The bus state controller is provided with a function for controlling synchronous DRAM refreshing. Auto-refreshing can be performed by clearing the RMODE bit to 0 and setting the RFSH bit to 1 in MCR. If synchronous DRAM is not accessed for a long period, self-refresh mode, in which the power consumption for data retain is low, can be activated by setting both the RMODE bit and the RFSH bit to 1.

1. Auto-Refreshing

Refreshing is performed at intervals determined by the input clock selected by bits CKS2 to CKS0 in RTCSR, and the value set in RTCOR. The value of bits CKS2 to CKS0 in RTCOR should be set so as to satisfy the refresh interval stipulation for the synchronous DRAM used. First make the settings for RTCOR, RTCNT, and the RMODE and RFSH bits in MCR, then make the CKS2 to CKS0 setting. When the clock is selected by CKS2 to CKS0, RTCNT starts counting up from the value at that time. The RTCNT value is constantly compared with the RTCOR value, and if the two values are the same, a refresh request is generated and an auto-refresh is performed. At the same time, RTCNT is cleared to zero and the count-up is restarted. Figure 12.18 shows the auto-refresh cycle timing.

All-bank precharging is performed in the Tp cycle, then an REF command is issued in the TRr cycle following the interval specified by the TPC bits in MCR. After the TRr cycle, new command output cannot be performed for the duration of the number of cycles specified by the TRAS bits in MCR plus the number of cycles specified by the TPC bits in MCR. The TRAS and TPC bits must be set so as to satisfy the synchronous DRAM refresh cycle time stipulation (active/active command delay time).

Auto-refreshing is performed in normal operation, in sleep mode, and in case of a manual reset.

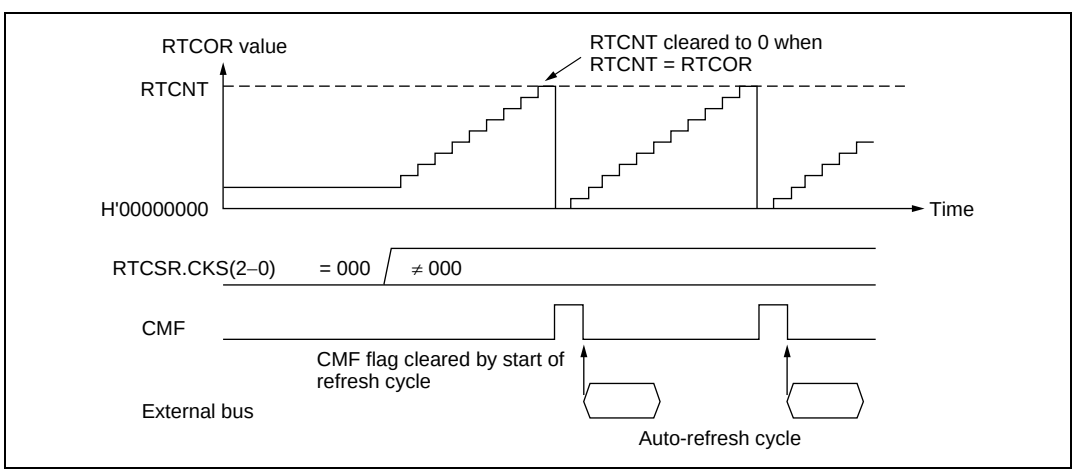


Figure 12.18 Auto-Refresh Operation

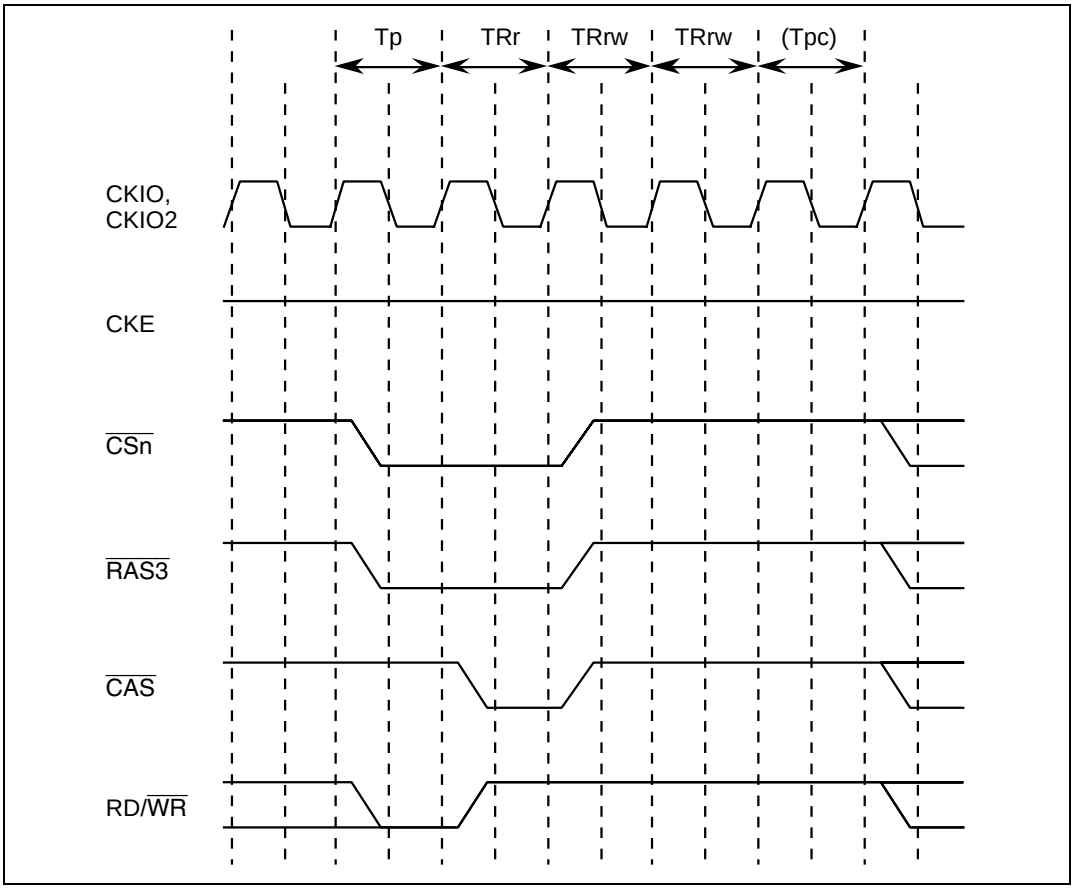


Figure 12.19 Synchronous DRAM Auto-Refresh Timing

2. Self-Refreshing

Self-refresh mode is a kind of standby mode in which the refresh timing and refresh addresses are generated within the synchronous DRAM. Self-refreshing is activated by setting both the RMODE bit and the RFSH bit to 1. The self-refresh state is maintained while the CKE signal is low. Synchronous DRAM cannot be accessed while in the self-refresh state. Self-refresh mode is cleared by clearing the RMODE bit to 0. After self-refresh mode has been cleared, command issuance is disabled for the number of cycles specified by the TPC bits in MCR. Self-refresh timing is shown in figure 12.20. Settings must be made so that self-refresh clearing and data retention are performed correctly, and auto-refreshing is performed at the correct intervals. When self-refreshing is activated from the state in which auto-refreshing is set, or when exiting standby mode other than through a power-on reset, auto-refreshing is restarted if RFSH is set to 1 and RMODE is cleared to 0 when self-refresh mode is cleared. If the transition from clearing of self-

refresh mode to the start of auto-refreshing takes time, this time should be taken into consideration when setting the initial value of RTCNT. Making the RTCNT value 1 less than the RTCOR value will enable refreshing to be started immediately.

After self-refreshing has been set, the self-refresh state continues even if the chip standby state is entered using this LSI's standby function, and is maintained even after recovery from standby mode other than through a power-on reset. In case of a power-on reset, the bus state controller's registers are initialized, and therefore the self-refresh state is cleared.

Self-refreshing is performed in normal operation, in sleep mode, in standby mode, and in case of a manual reset. In addition, halt USB and LCDC before entering standby mode.

When the synchronous DRAM is used, self-refreshing is initiated in the following procedure.

1. Clear the refresh control bit to 0.
2. Write H'00 to RTCNT
3. Set the refresh control bit and refresh mode bit to 1.

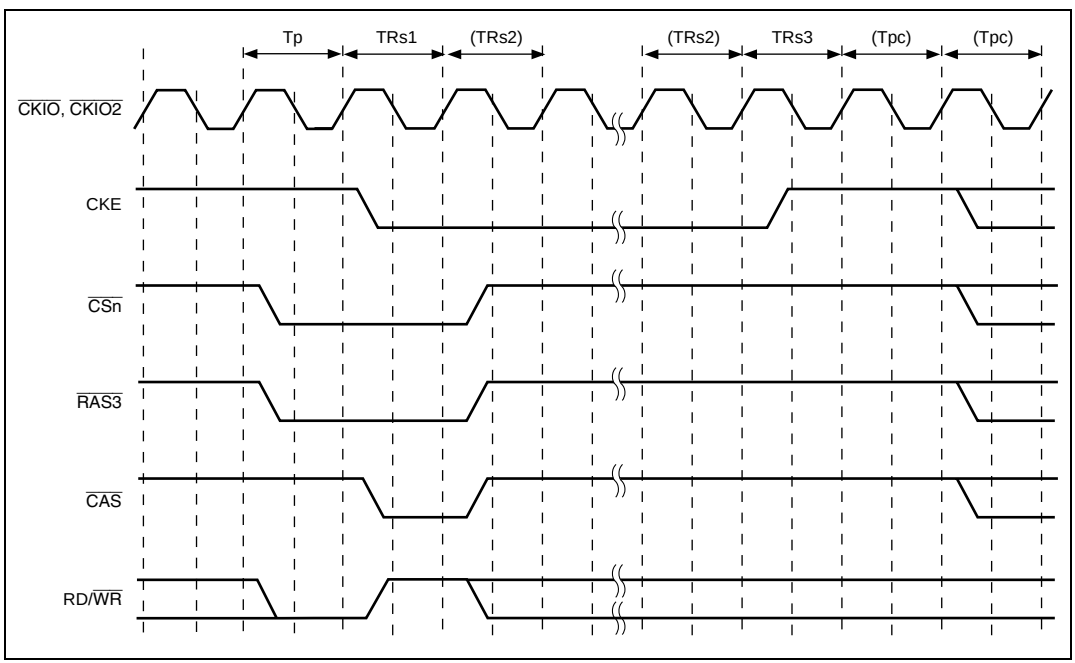


Figure 12.20 Synchronous DRAM Self-Refresh Timing

Power-On Sequence: In order to use synchronous DRAM, mode setting must first be performed after powering on. To perform synchronous DRAM initialization correctly, the bus state controller registers must first be set, followed by a write to the synchronous DRAM mode register. In synchronous DRAM mode register setting, the address signal value at that time is latched by a combination of the $\overline{RAS}$, $\overline{CAS}$, and RD/$\overline{WR}$ signals. If the value to be set is X, the bus state controller provides for value X to be written to the synchronous DRAM mode register by performing a write to address H'FFFFD000 + X for area 2 synchronous DRAM, and to address H'FFFFE000 + X for area 3 synchronous DRAM. In this operation the data is ignored, but the mode write is performed as a byte-size access. To set burst read/single write, CAS latency 1 to 3, wrap type = sequential, and burst length 1 supported by this LSI, arbitrary data is written in a byte-size access to the following addresses.

With 32-bit bus width:

	Area 2	Area 3
CAS latency 1	FFFFD840	FFFFE840
CAS latency 2	FFFFD880	FFFFE880
CAS latency 3	FFFFD8C0	FFFFE8C0

With 16-bit bus width:

	Area 2	Area 3
CAS latency 1	FFFFD420	FFFFE420
CAS latency 2	FFFFD440	FFFFE440
CAS latency 3	FFFFD460	FFFFE460

Mode register setting timing is shown in figure 12.21.

As a result of the write to address H'FFFFD000 + X or H'FFFFE000 + X, a precharge all banks (PALL) command is first issued in the TRp1 cycle, then a mode register write command is issued in the TMw1 cycle.

Address signals, when the mode-register write command is issued, are as follows:

A15 to A9 = 0000100 (burst read and single write)
A8 to A6 = CAS latency
A5 = 0 (burst type = sequential)
A4 to A2 = 000 (burst length 1)

Before mode register setting, a 100 µs idle time (depending on the memory manufacturer) must be guaranteed after powering on requested by the synchronous DRAM. If the reset signal pulse width is greater than this idle time, there is no problem in performing mode register setting immediately. The number of dummy auto-refresh cycles specified by the manufacturer (usually 8) or more must be executed. This is usually achieved automatically while various kinds of initialization are being

performed after auto-refresh setting, but a way of carrying this out more dependably is to set a short refresh request generation interval just while these dummy cycles are being executed. With simple read or write access, the address counter in the synchronous DRAM used for auto-refreshing is not initialized, and so the cycle must always be an auto-refresh cycle.

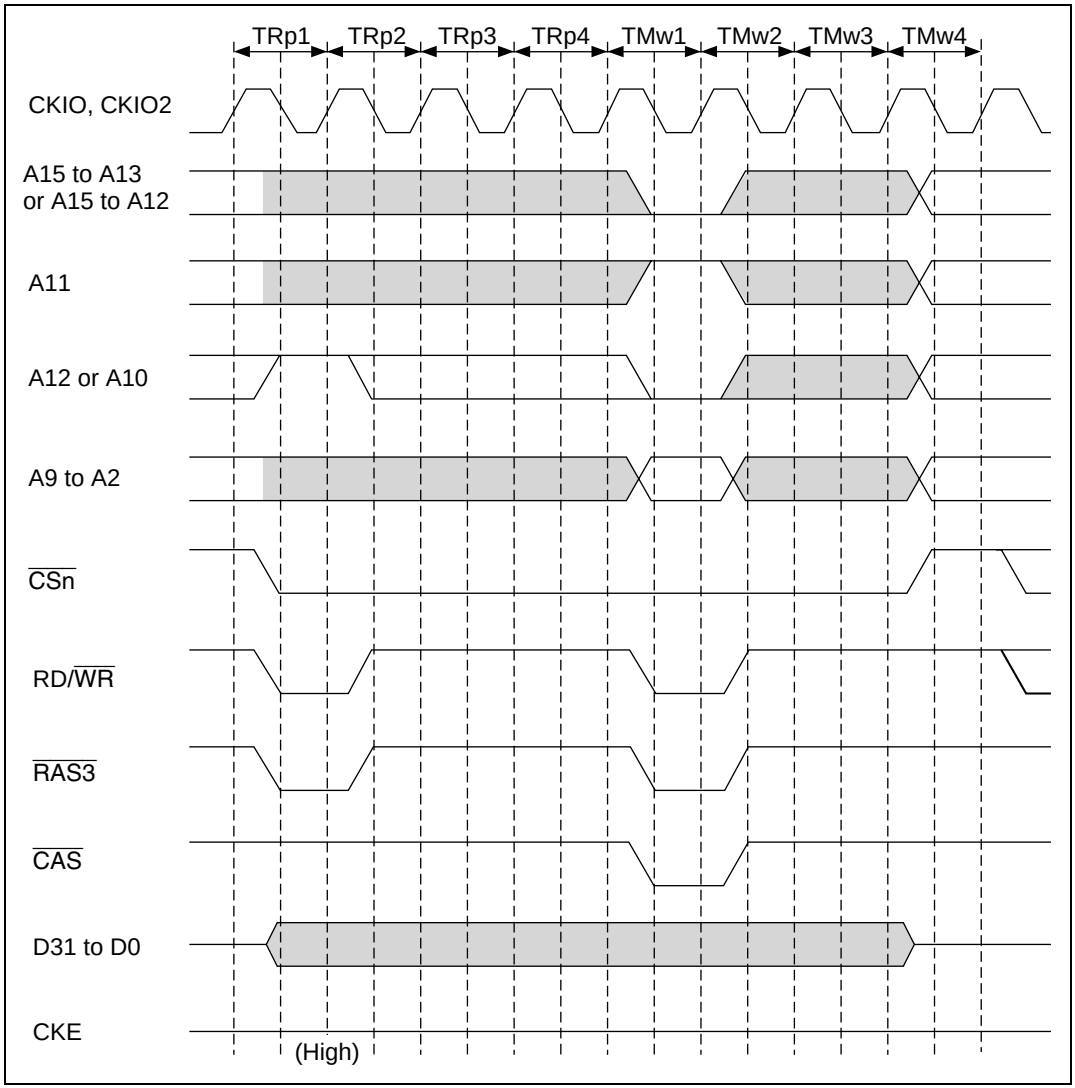


Figure 12.21 Synchronous DRAM Mode Write Timing

12.3.5 Burst ROM Interface

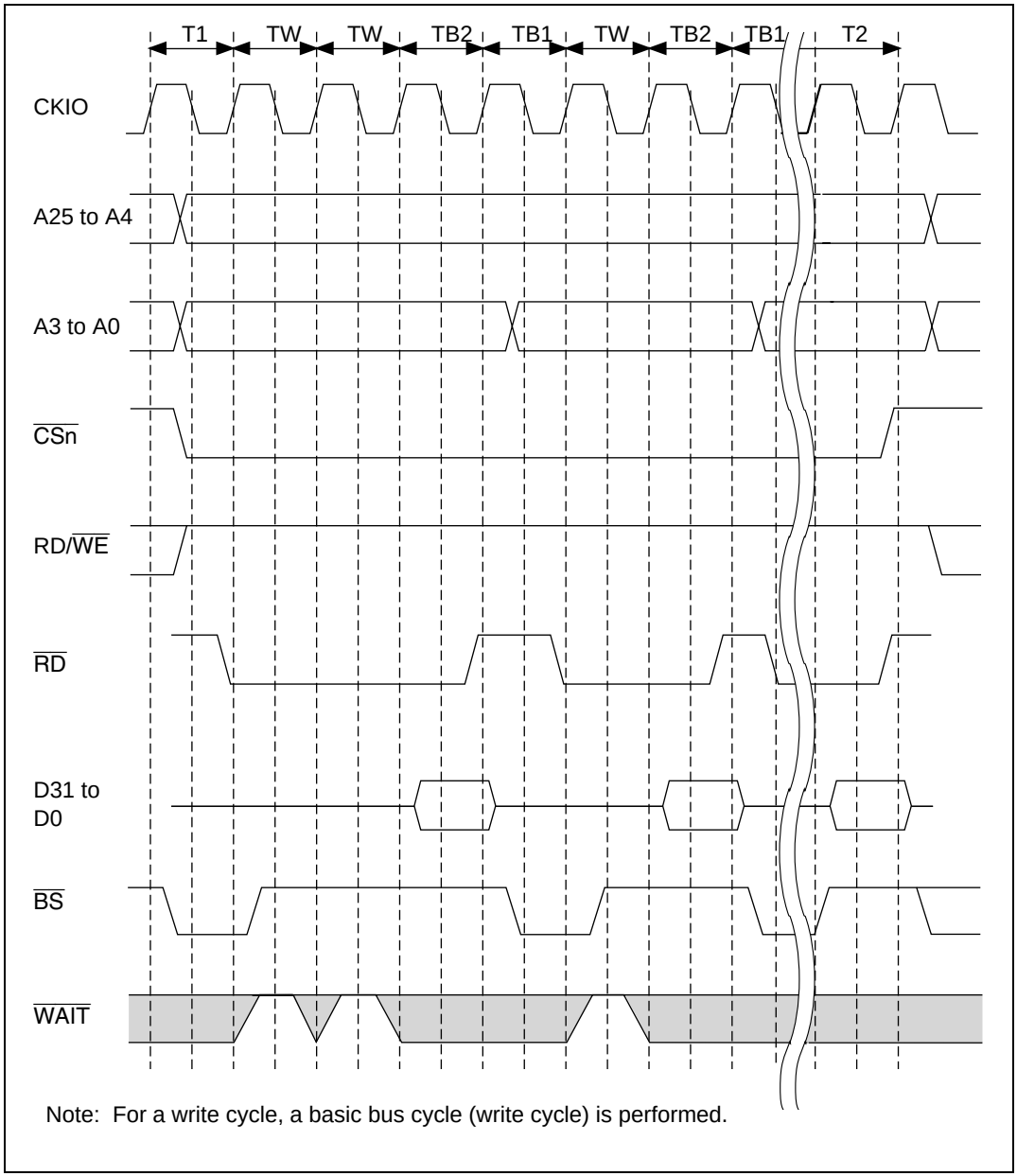
Setting bits A0BST (1, 0), A5BST (1, 0), and A6BST (1, 0) in BCR1 to a non-zero value allows burst ROM to be connected to areas 0, 5, and 6. The burst ROM interface provides high-speed access to ROM that has a nibble access function. The timing for nibble access to burst ROM is shown in figure 12.22. Two wait cycles are set. Basically, access is performed in the same way as for normal space, but when the first cycle ends the $\overline{CS0}$ signal is not negated, and only the address is changed before the next access is executed. When 8-bit ROM is connected, the number of consecutive accesses can be set as 4, 8, or 16 by bits A0BST (1, 0), A5BST (1, 0), or A6BST (1, 0). When 16-bit ROM is connected, 4 or 8 can be set in the same way. When 32-bit ROM is connected, only 4 can be set.

$\overline{WAIT}$ pin sampling is performed in the first access if one or more wait states are set, and is always performed in the second and subsequent accesses.

Even if no wait state insertion is specified in burst ROM interface settings, two wait cycles are automatically inserted in the second and subsequent accesses as shown in figure 12.23.

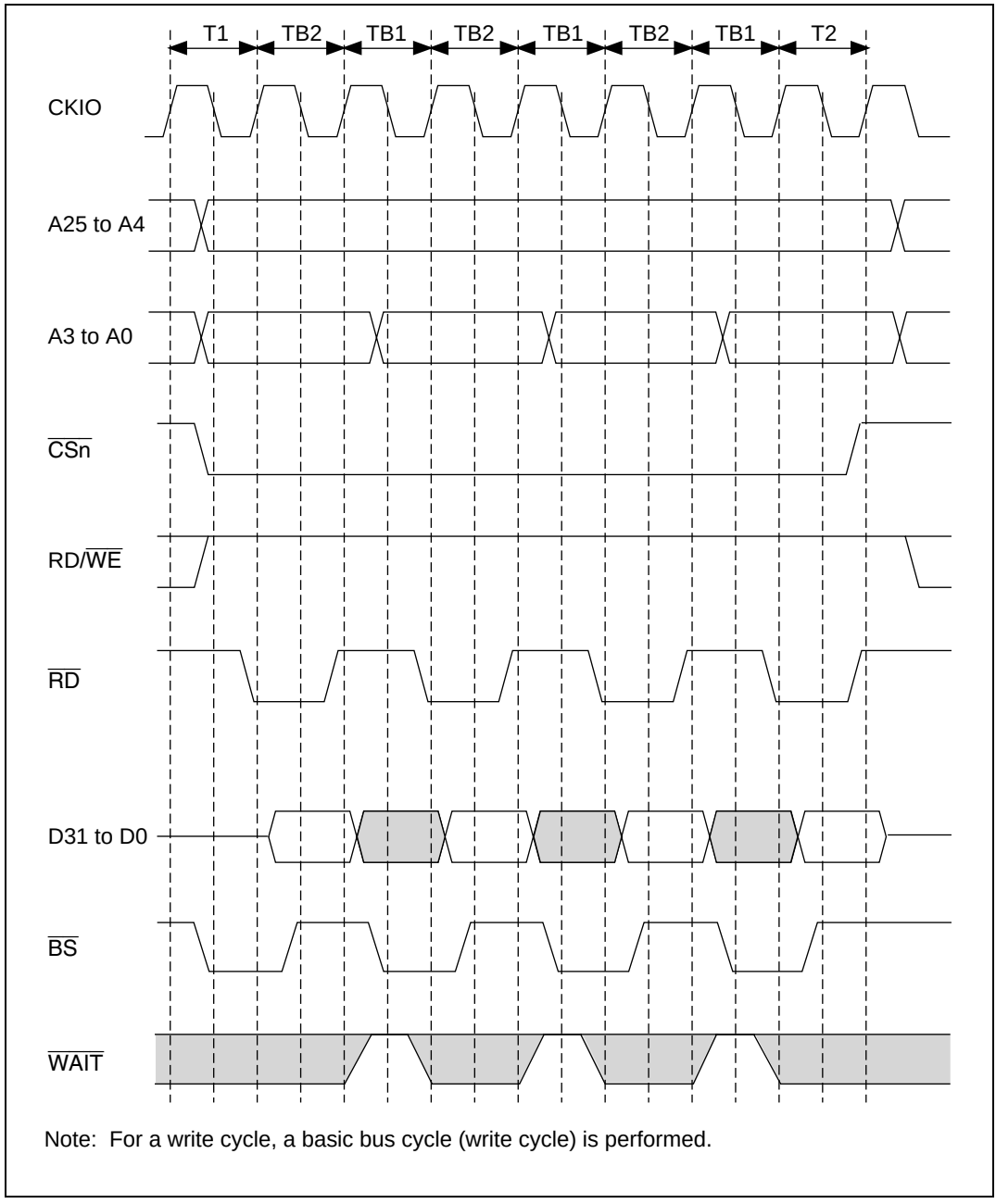
However, the $\overline{WAIT}$ signal is ignored in the following three cases:

- When writing to an external address area using DMA 16-byte transfer in dual address mode
- When transferring data from a DACK-equipped external device to an external address area using DMA 16-byte transfer in single address mode
- During cache write-back access



Note: For a write cycle, a basic bus cycle (write cycle) is performed.

Figure 12.22 Burst ROM Wait Access Timing



Note: For a write cycle, a basic bus cycle (write cycle) is performed.

Figure 12.23 Burst ROM Basic Access Timing

12.3.6 PCMCIA Interface

In this LSI, setting the A5PCM bit in BCR1 to 1 makes the bus interface for physical space area 5 an IC memory card and I/O card interface as stipulated in JEIDA version 4.2 (PCMCIA2.1). Setting the A6PCM bit to 1 makes the bus interface for physical space area 6 an IC memory card and I/O card interface as stipulated in JEIDA version 4.2.

When the PCMCIA interface is used, a bus size of 8 or 16 bits can be set by bits A5SZ1 and A5SZ0, or A6SZ1 and A6SZ0, in BCR2.

Figure 12.24 shows an example of PCMCIA card connection to this LSI. To enable active insertion of the PCMCIA cards (i.e. insertion or removal while system power is being supplied), a 3-state buffer must be connected between this LSI's bus interface and the PCMCIA cards.

As operation in big-endian mode is not explicitly stipulated in the JEIDA/PCMCIA specifications, the PCMCIA interface for this LSI in big-endian mode is stipulated independently.

However, the $\overline{\text{WAIT}}$ signal is ignored in the following three cases:

- When writing to an external address area using DMA 16-byte transfer in dual address mode
- When transferring data from a DACK-equipped external device to an external address area using DMA 16-byte transfer in single address mode
- During cache write-back access

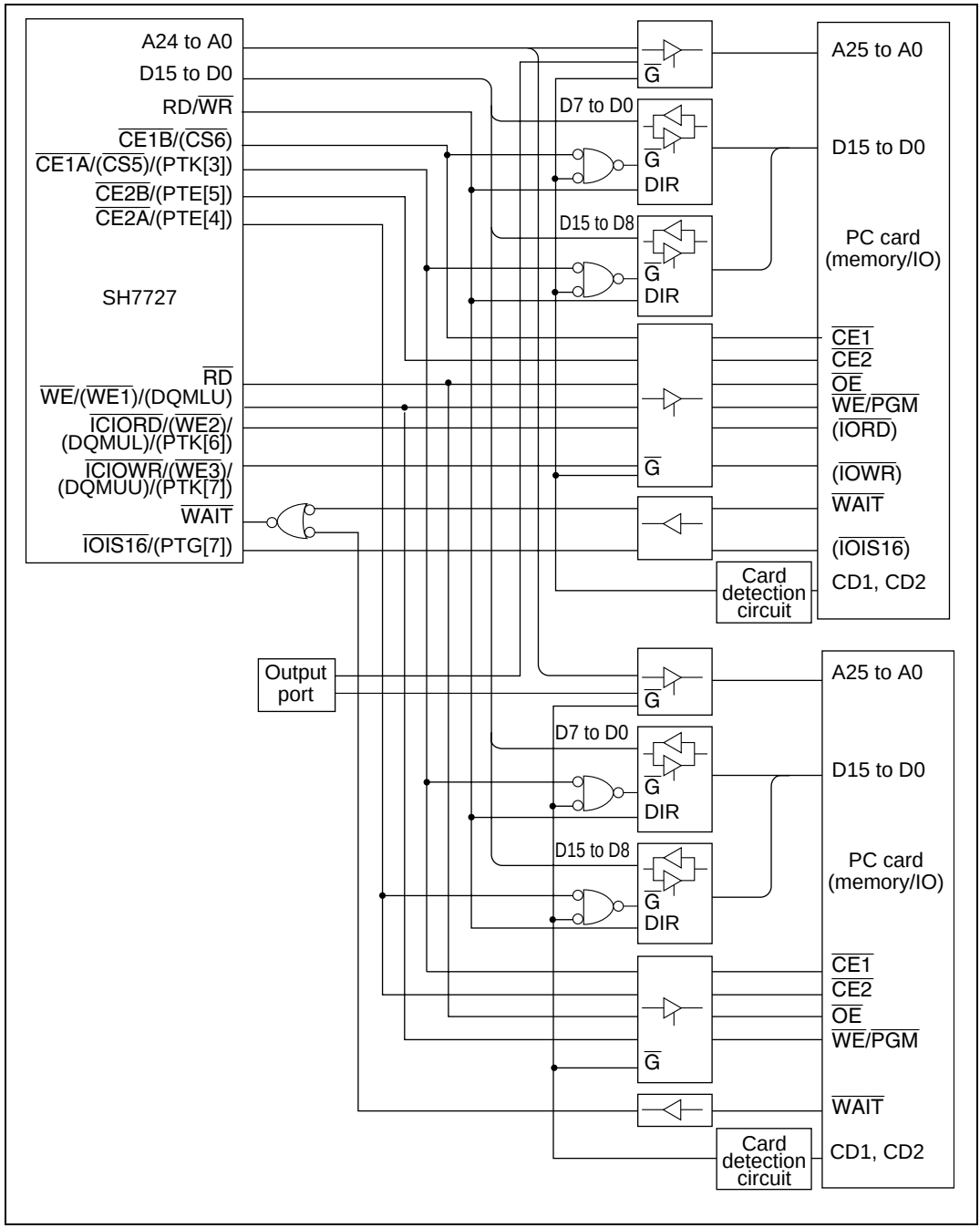


Figure 12.24 Example of PCMCIA Interface (If Internal PC Card Controller is not used.)

Memory Card Interface Basic Timing: Figure 12.25 shows the basic timing for the PCMCIA IC memory card interface. When physical space areas 5 and 6 are designated as PCMCIA interface areas, bus accesses are automatically performed as IC memory card interface accesses.

With a high external bus frequency (CKIO), the setup and hold times for the address (A24 to A0), card enable ($\overline{CS5}$, $\overline{CE2A}$, $\overline{CS6}$, $\overline{CE2B}$), and write data (D15 to D0) in a write cycle, become insufficient with respect to $\overline{RD}$ and $\overline{WR}$ ($\overline{WE}$ pin in this LSI). This LSI provides for this by enabling setup and hold times to be set for physical space areas 5 and 6 in the PCR register. Also, software waits by means of a WCR2 register setting and hardware waits by means of the $\overline{WAIT}$ pin can be inserted in the same way as for the basic interface. Figure 12.26 shows the PCMCIA memory bus wait timing.

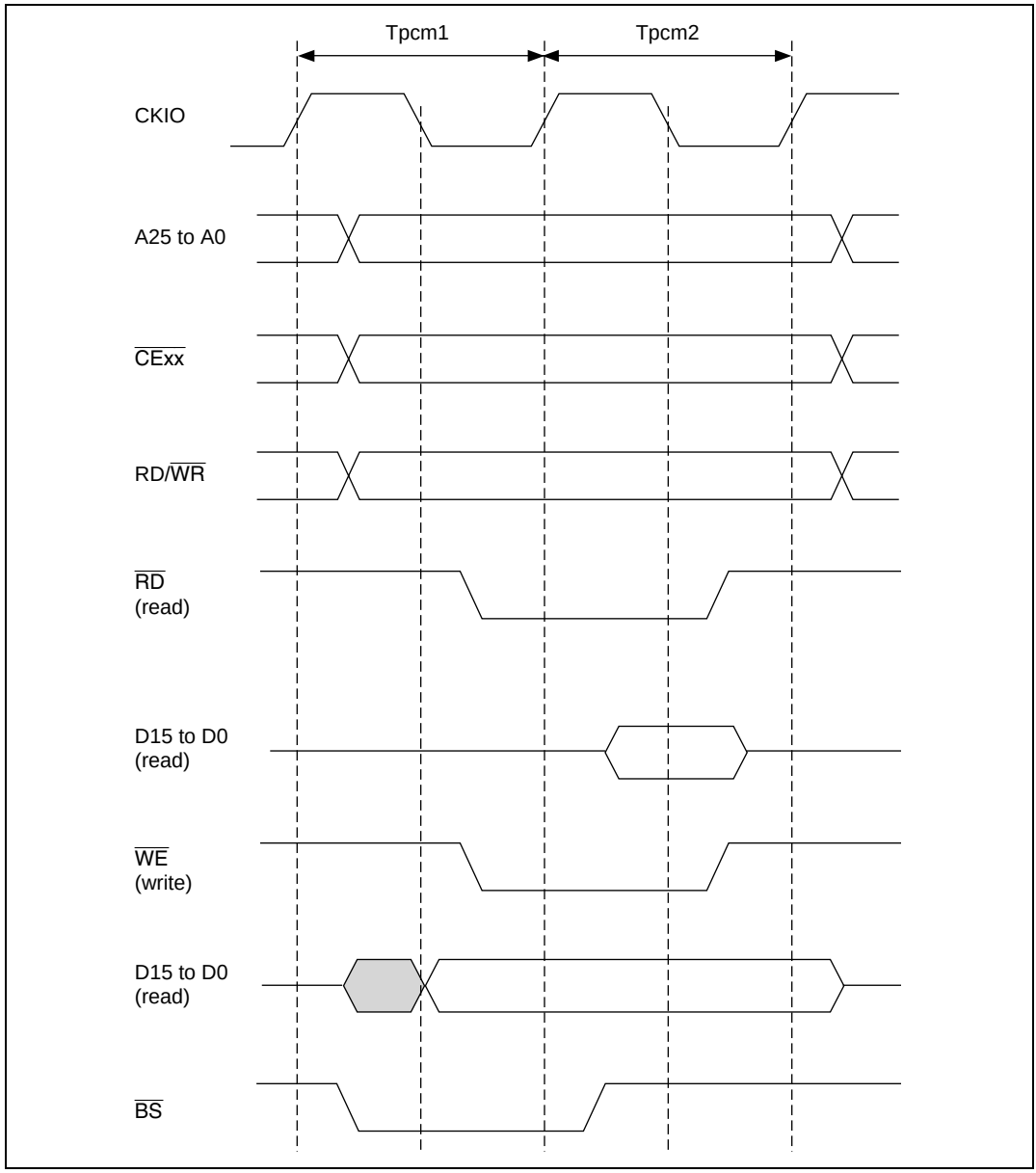


Figure 12.25 Basic Timing for PCMCIA Memory Card Interface

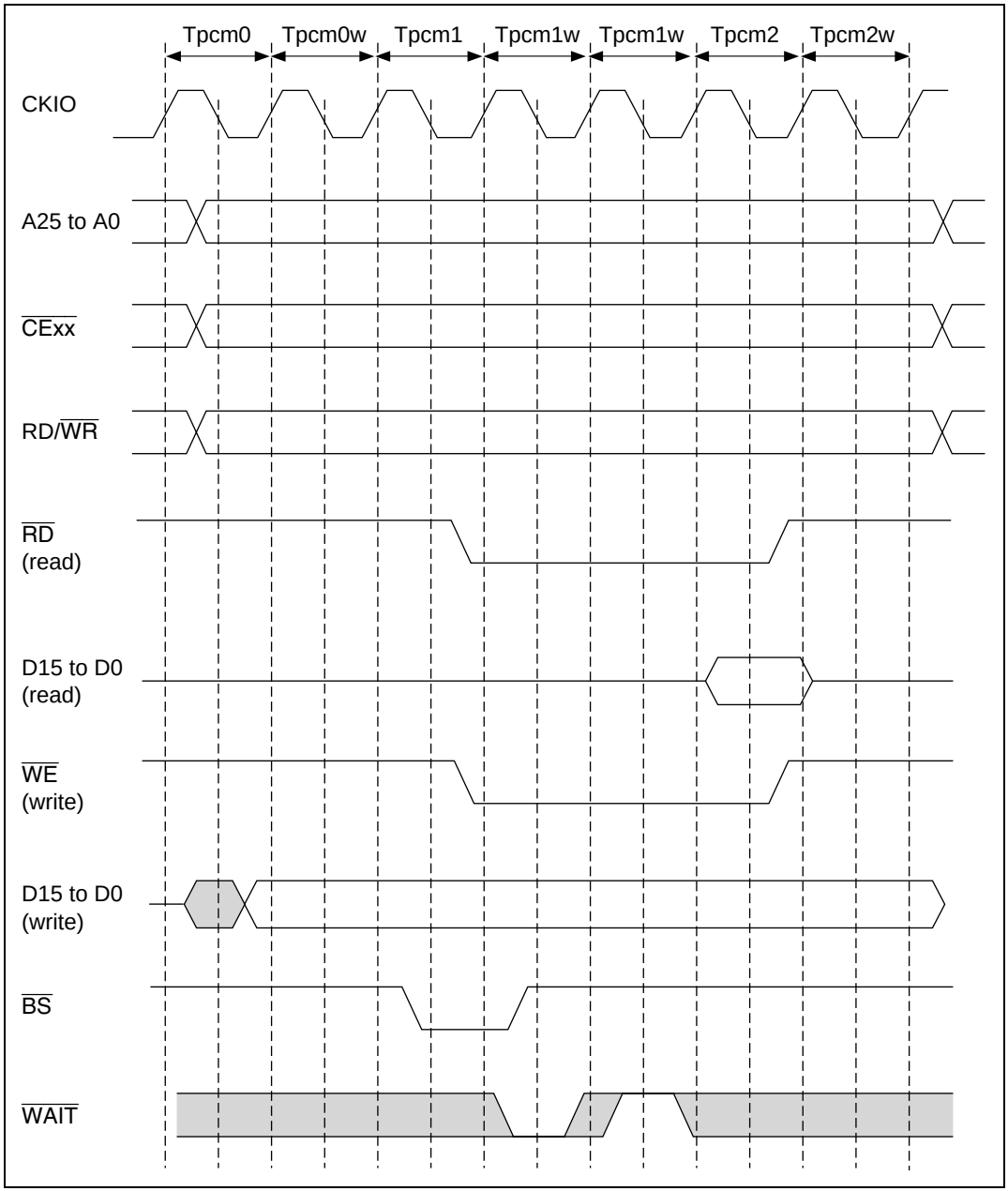


Figure 12.26 Wait Timing for PCMCIA Memory Card Interface

Memory Card Interface Burst Timing: In this LSI, when the IC memory card interface is selected, page mode burst access mode can be used, for read access only, by setting bits A5BST1 and A5BST0 in BCR1 for physical space area 5, or bits A6BST1 and A6BST0 in BCR1 for area 6. This burst access mode is not stipulated in JEIDA version 4.2 (PCMCIA2.1), but allows high-speed data access using ROM provided with a burst mode, etc.

Burst access mode timing is shown in figures 12.27 and 12.28.

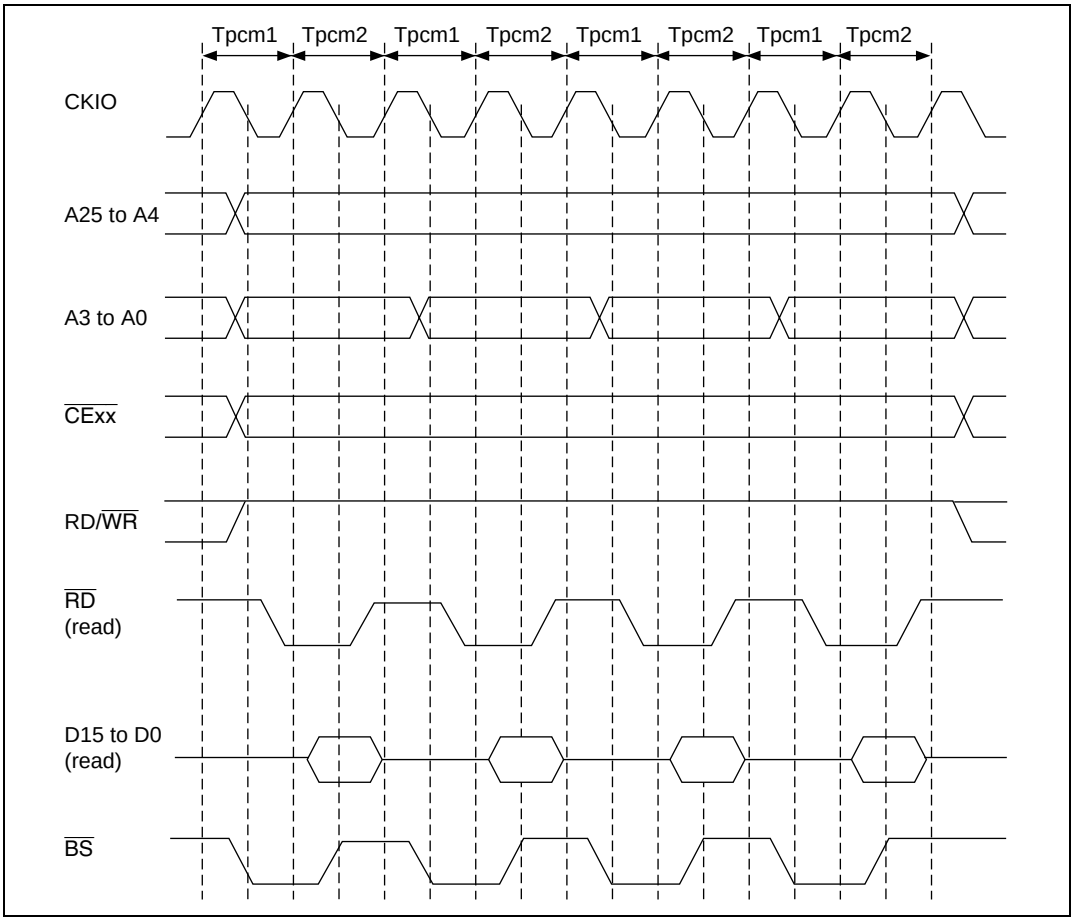


Figure 12.27 Basic Timing for PCMCIA Memory Card Interface Burst Access

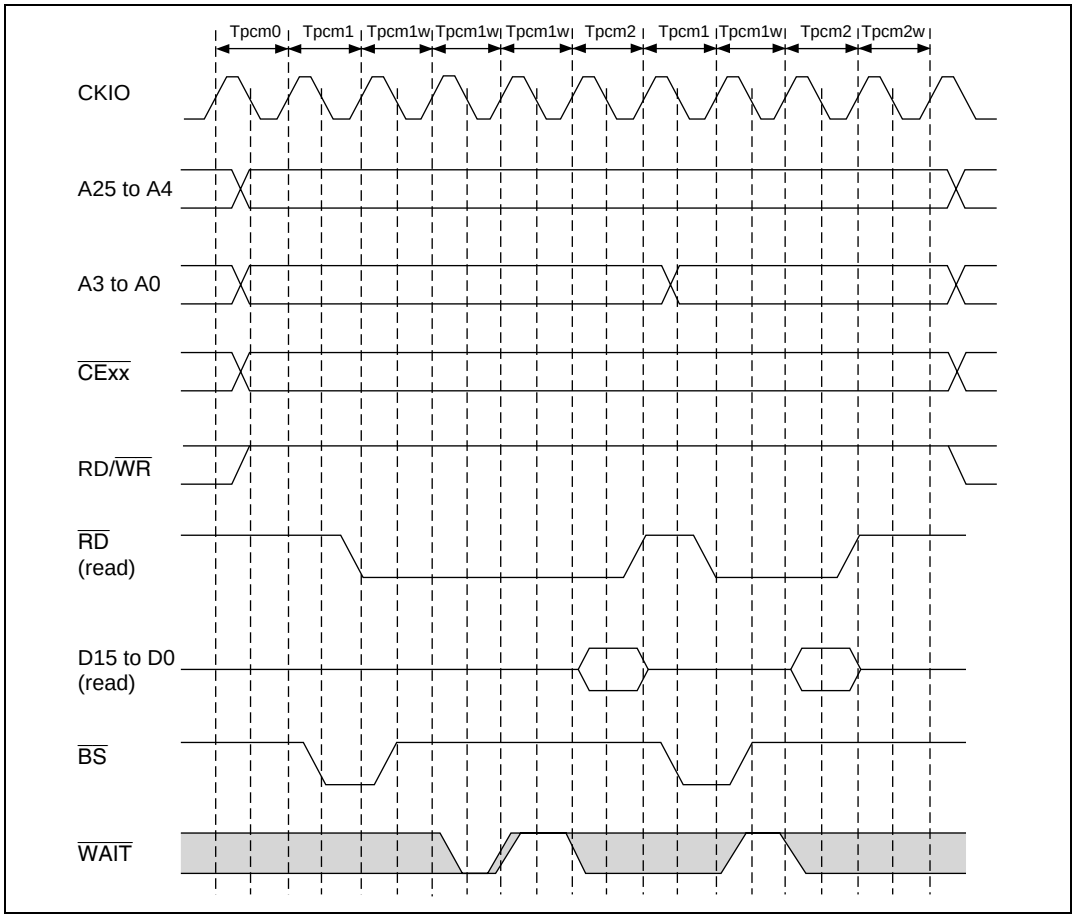


Figure 12.28 Wait Timing for PCMCIA Memory Card Interface Burst Access

When the IC memory card interface uses entire 32-Mbyte memory space, the REG signal to switch common memory and attribute memory can be generated by a port. If the IC card memory interface uses memory area of 16 Mbytes or less, 32-Mbyte memory space can be used as 16-Mbyte common memory space and 16-Mbyte attribute memory space as shown in figure 12.29. In this case, A24 pin can be used as the $\overline{\text{REG}}$ signal.

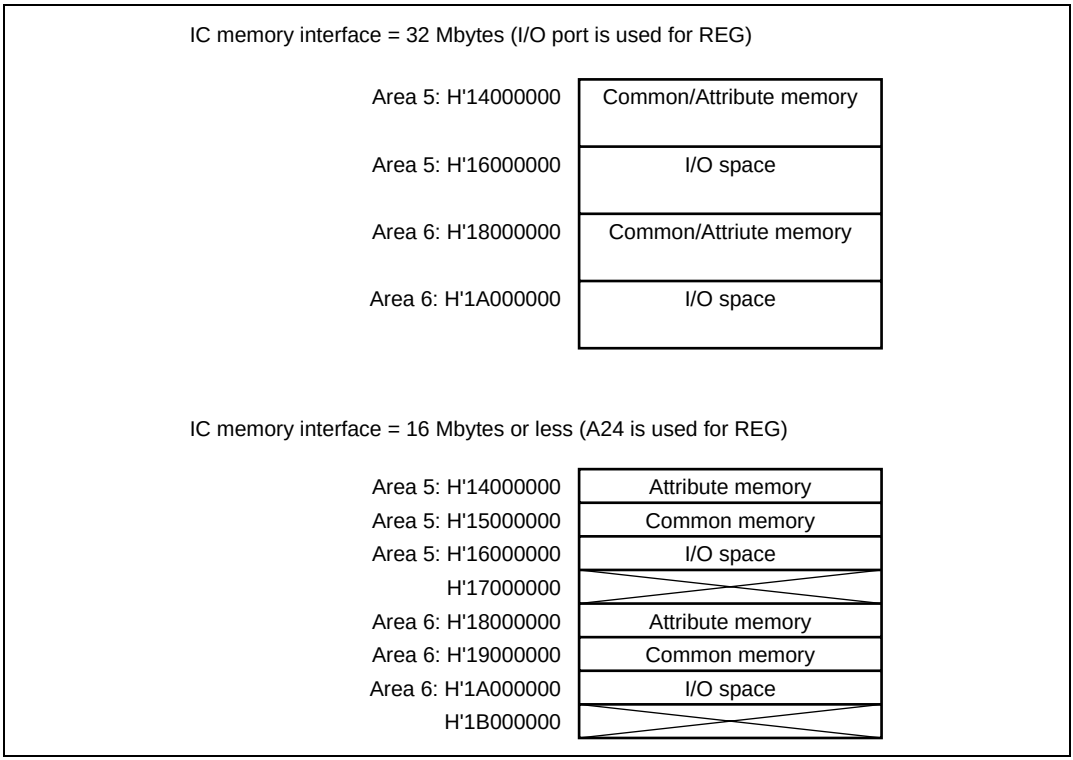


Figure 12.29 PCMCIA Space Assignment

I/O Card Interface Timing: Figures 12.30 and 12.31 show the timing for the PCMCIA I/O card interface.

Switching between the I/O card interface and the IC memory card interface is performed according to the accessed address. When PCMCIA is designed for physical space area 5, the bus access is automatically performed as an I/O card interface access when a physical address from H'16000000 to H'17FFFFFF is accessed. When PCMCIA is designated for physical space area 6, the bus access is automatically performed as an I/O card interface access when a physical address from H'1A000000 to H'1BFFFFFF is accessed.

When accessing a PCMCIA I/O card, the access should be performed using a non-cacheable area in virtual space (P2 or P3 space) or an area specified as non-cacheable by the MMU.

When an I/O card interface access is made to a PCMCIA card in little-endian mode, dynamic sizing of the I/O bus width is possible using the $\overline{\text{IOIS16}}$ pin. When a 16-bit bus width is set for area 5 or area 6, if the $\overline{\text{IOIS16}}$ signal is high during a word-size I/O bus cycle, the I/O port is recognized as being 8 bits in width. In this case, a data access for only 8 bits is performed in the I/O bus cycle being executed, followed automatically by a data access for the remaining 8 bits.

Figure 12.32 shows the basic timing for dynamic bus sizing.

In big-endian mode, the $\overline{\text{IOIS16}}$ signal is not supported.

In big-endian mode, the $\overline{\text{IOIS16}}$ signal should be fixed low.

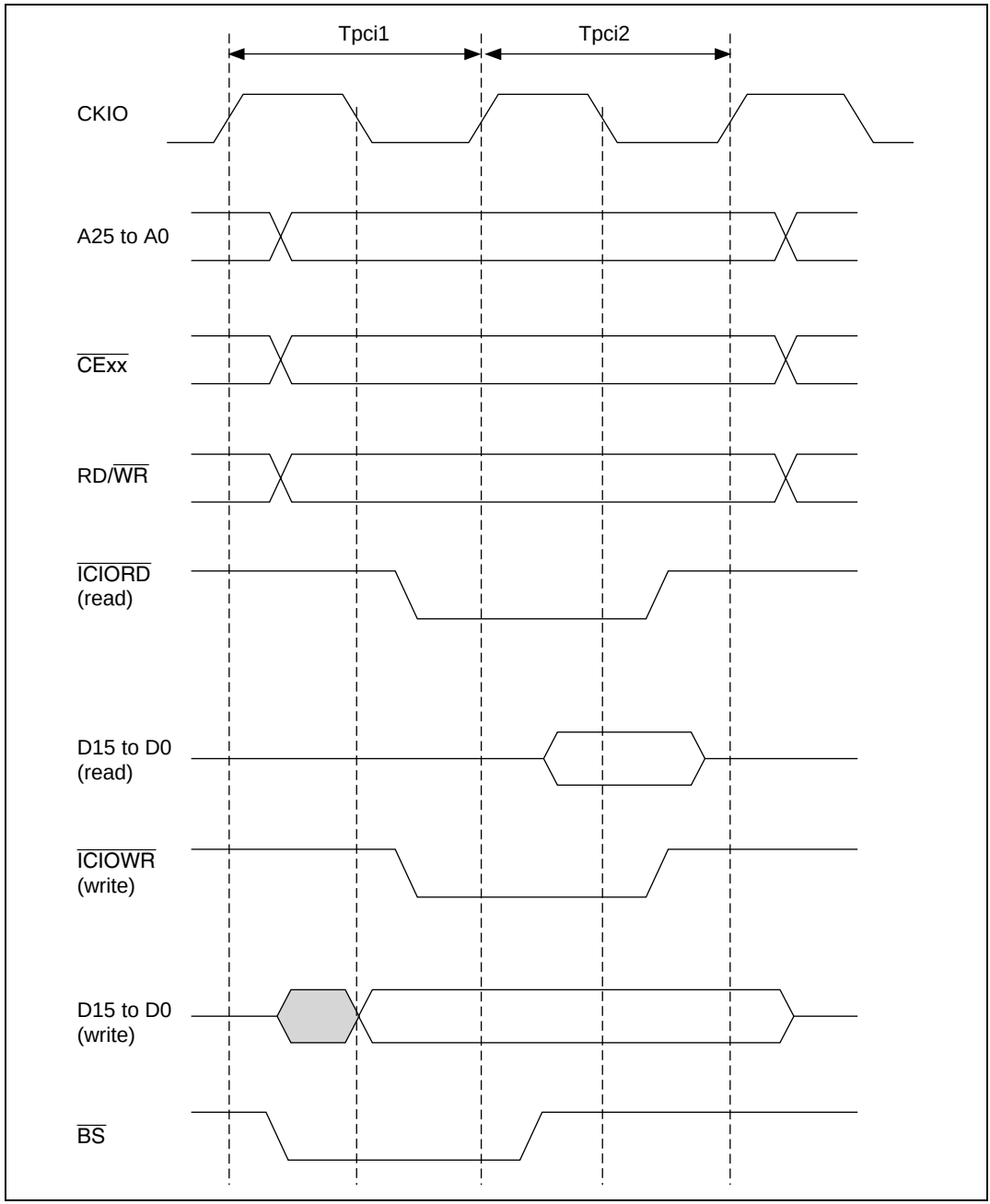


Figure 12.30 Basic Timing for PCMCIA I/O Card Interface

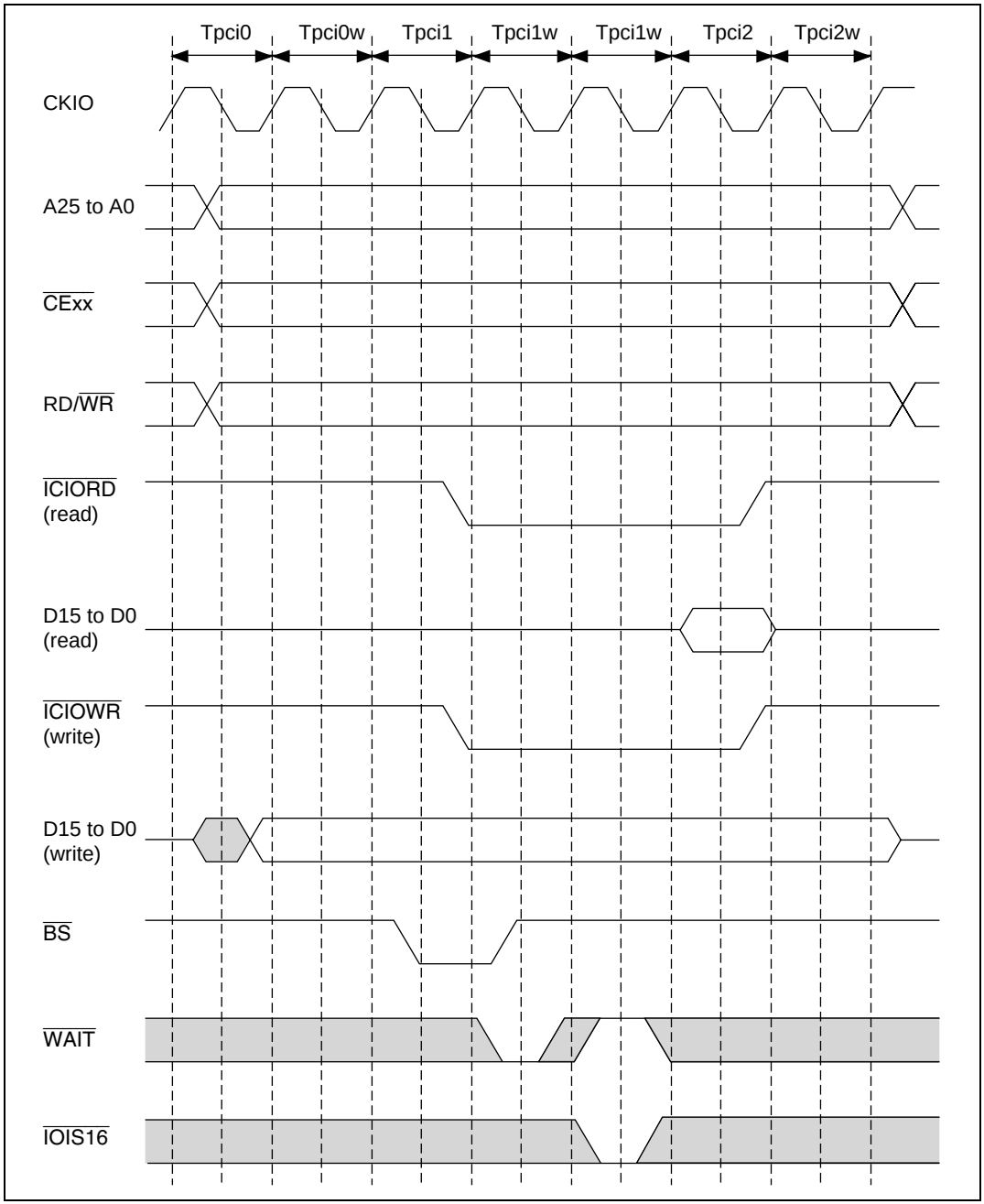


Figure 12.31 Wait Timing for PCMCIA I/O Card Interface

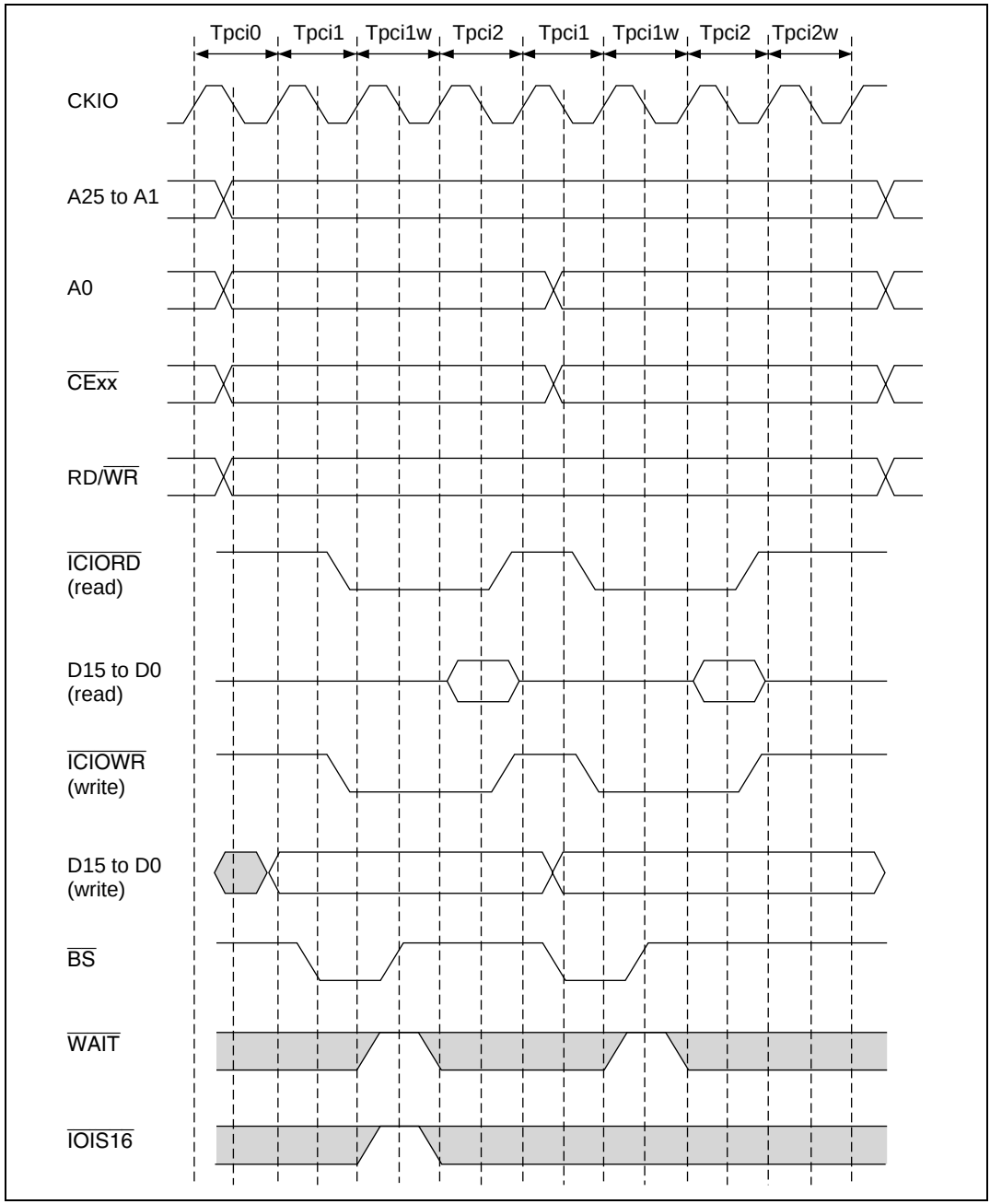


Figure 12.32 Dynamic Bus Sizing Timing for PCMCIA I/O Card Interface

12.3.7 Waits between Access Cycles

A problem associated with higher external memory bus operating frequencies is that data buffer turn-off on completion of a read from a low-speed device may be too slow, causing a collision with data in the next access. This results in lower reliability or incorrect operation. To avoid this problem, a data collision prevention feature has been provided. This memorizes the preceding access area and the kind of read/write. If there is a possibility of a bus collision when the next access is started, a wait cycle is inserted before the access cycle thus preventing a data collision. There are two cases in which a wait cycle is inserted: when an access is followed by an access to a different area, and when a read access is followed by a write access from this LSI. When this LSI performs consecutive write cycles, the data transfer direction is fixed (from this LSI to other memory) and there is no problem. With read accesses to the same area, in principle, data is output from the same data buffer, and wait cycle insertion is not performed. Bits AnIW1 and AnIW0 (n = 0, 2 to 6) in WCR1 specify the number of idle cycles to be inserted between access cycles when a physical space area access is followed by an access to another area, or when this LSI performs a write access after a read access to physical space area n. If there is originally space between accesses, the number of idle cycles inserted is the specified number of idle cycles minus the number of empty cycles.

Waits are not inserted between accesses when bus arbitration is performed, since empty cycles are inserted for arbitration purposes.

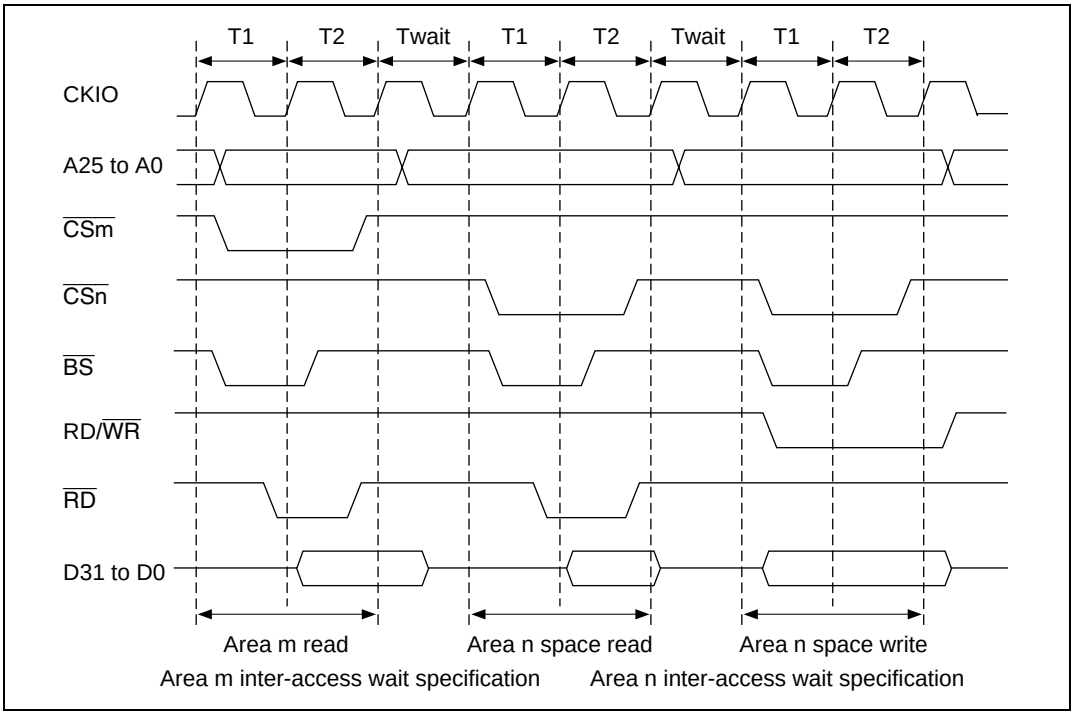


Figure 12.33 Waits between Access Cycles

12.3.8 Bus Arbitration

When a bus release request ($\overline{\text{BREQ}}$) is received from an external device, buses are released after the bus cycle being executed is completed and a bus grant signal ($\overline{\text{BACK}}$) is output. The bus is not released during burst transfers for cache fills or TAS instruction execution between the read cycle and write cycle. Bus arbitration is not executed in multiple bus cycles that are generated when the data bus width is shorter than the access size; i.e. in the bus cycles when longword access is executed for the 8-bit memory. At the negation of $\overline{\text{BREQ}}$, $\overline{\text{BACK}}$ is negated and bus use is restarted. See Appendix A.1, Pin Functions, for the pin state when the bus is released.

12.3.9 Bus Pull-Up

With this LSI, address pin pull-up can be performed when the bus is released by setting the PULA bit in BCR1 to 1. The address pins are pulled up for a 4-clock period after $\overline{\text{BACK}}$ is asserted. Figure 12.34 shows the address pin pull-up timing. Similarly, data pin pull-up can be performed by setting the PULD bit in BCR1 to 1. The data pins should be pulled up when the data bus is not in use. The data pin pull-up timing for a read cycle is shown in figure 12.35, and the timing for a write cycle in figure 12.36.

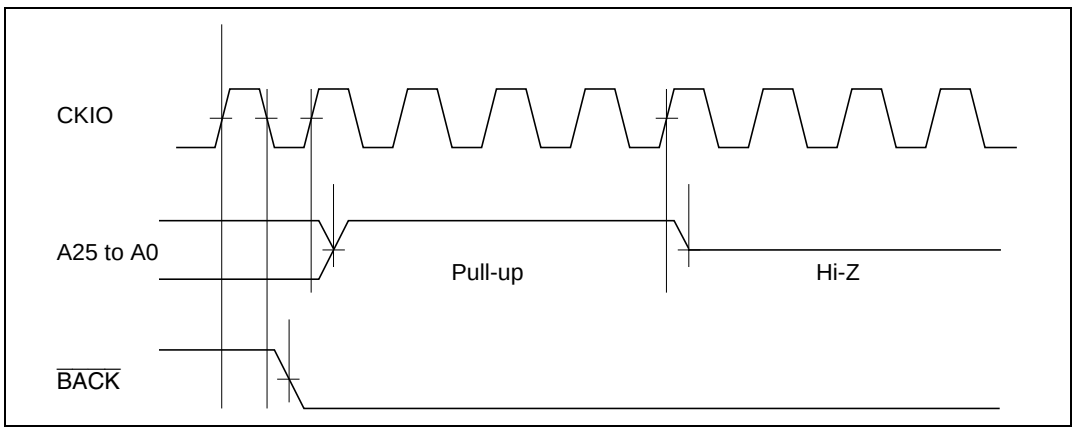


Figure 12.34 Pins A25 to A0 Pull-Up Timing

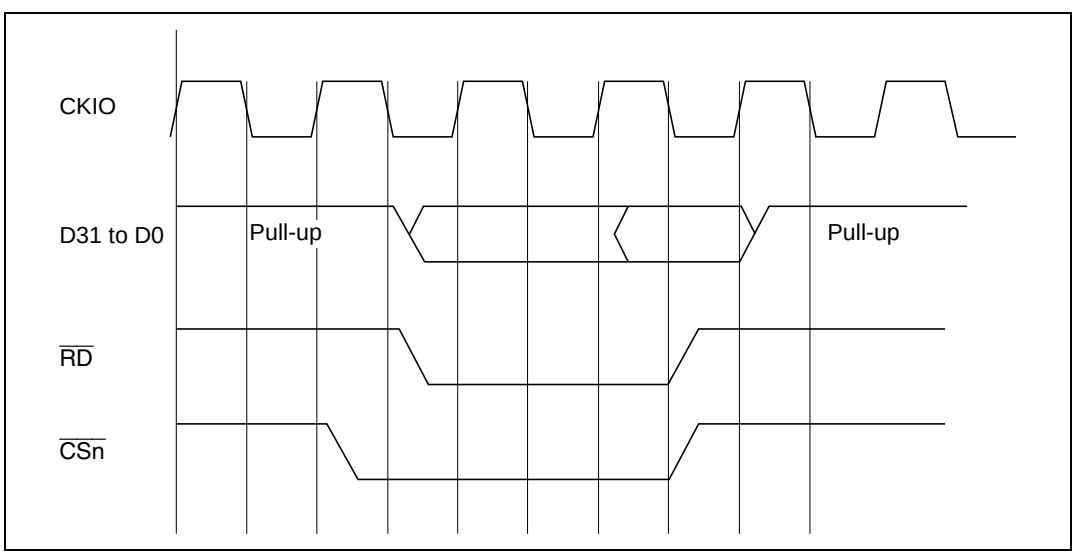


Figure 12.35 Pins D31 to D0 Pull-Up Timing (Read Cycle)

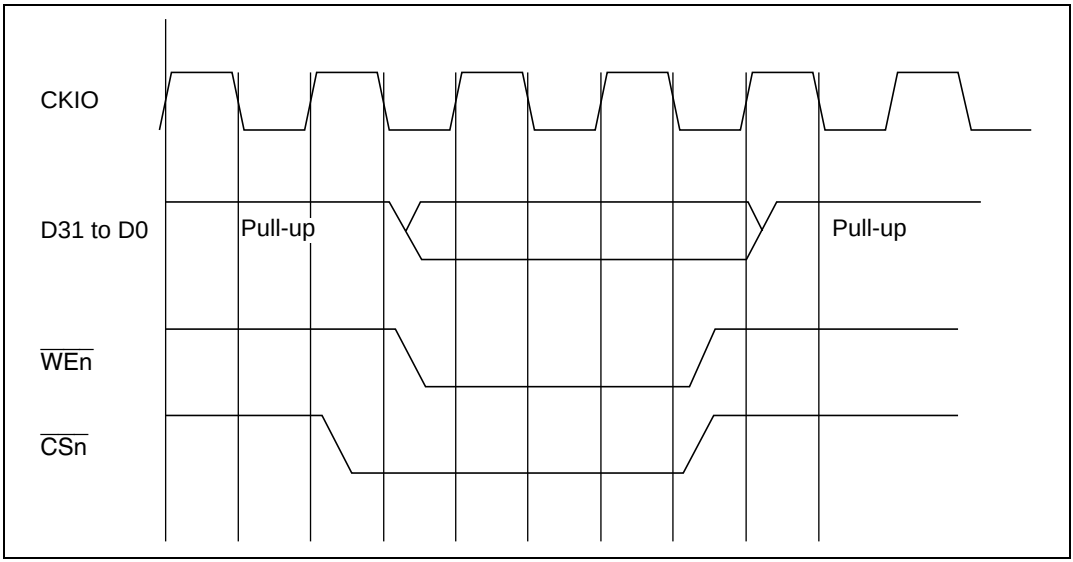


Figure 12.36 Pins D31 to D0 Pull-Up Timing (Write Cycle)

Section 13 Li Bus State Controller (LBSC)

13.1 Overview

The Li bus state controller (LBSC) functions enable LCD controller and Open HCI compliant USB Host controller to link directly with synchronous DRAM. LBSC is a slave bus state controller of BSC.

13.1.1 Features

The LBSC has the following features:

- Direct interface to synchronous DRAM
  - Physical address space is specified only to area 3
  - A maximum 64 Mbytes
  - Multiplexes row/column addresses according to synchronous DRAM capacity
  - Supports burst operation with various burst length; selectable from 1 to 32
  - Controls timing of synchronous DRAM direct-connection control signals according to register setting
  - 16-bit or 32-bit bus width according to register setting

13.1.2 Register Configuration

The LBSC does not have any register inside, but refers BSC registers shown in table 13.1.

Table 13.1 Register Configuration

Name	Abbr.	R/W	Initial Value*	Address	Bus Width
Bus control register 1	BCR1	R/W	H'0000	H'FFFFFF60	16
Bus control register 2	BCR2	R/W	H'3FF0	H'FFFFFF62	16
Wait state control register 1	WCR1	R/W	H'3FF3	H'FFFFFF64	16
Wait state control register 2	WCR2	R/W	H'FFFF	H'FFFFFF66	16
Individual memory control register	MCR	R/W	H'0000	H'FFFFFF68	16

Note: * Initialized by power-on resets.

13.1.3 Bus Control Register 1 (BCR1)

Bus control register 1 (BCR1) is a 16-bit read/write register that sets the functions and bus cycle state for each area. It is initialized to H'0000 by a power-on reset, but it is not initialized by a manual reset or in standby mode. Do not access external memory except area 0 until BCR1 register initialization is complete.

Bit:	15	14	13	12	11	10	9	8
	PULA	PULD	HIZMEM	HIZCNT	ENDIAN	A0BST1	A0BST0	A5BST1
Initial value:	0	0	0	0	0/1*	0	0	0
R/W:	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
	A5BST0	A6BST1	A6BST0	DRAM TP2	DRAM TP1	DRAM TP0	A5PCM	A6PCM
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

Note: * Samples the value of the external pin (MD5) designating endian at power-on reset.

Bits 15 to 12—Not referenced

Bit 11—Endian Flag (ENDIAN): Samples a value at the external pin which designates endian (MD5) at a power-on reset. Endian for all physical spaces is decided by this bit. This bit is read-only.

Bit 11: ENDIAN	Description
0	At a reset, the endian setting external pin (MD5) is low, which indicates that the SH7727 is set as big endian.
1	At a reset, the endian setting external pin (MD5) is high, which indicates that the SH7727 is set as little endian.

Bits 10 to 5—Not referenced

Bits 4 to 2—Area 2, Area 3 Memory Type (DRAMTP2, DRAMTP1, DRAMTP0): Specifies the type of memory connected to the physical space areas 2 and 3. Before using LCDC and USB, set area 3 to synchronous DRAM (DRAMTP2 to DRAMTP0 equal to 010 or 011).

Bit 4: DRAMTP2	Bit 3: DRAMTP1	Bit 2: DRAMTP0	Description
0	0	0	Ordinary memory for areas 2 and 3 (Initial value)
		1	Reserved (Setting disabled)
	1	0	Ordinary memory for area 2 and synchronous DRAM for area 3*1
		1	Synchronous DRAM for areas 2 and 3*1 *2
1	0	0	Reserved
		1	Reserved
	1	0	Reserved (Setting disabled)
		1	Reserved (Setting disabled)

Notes: 1. It is not possible to access synchronous DRAM if clock ratio Iϕ:bus clock = 1:1.
2. When selecting this mode, set the same bus width for area 2 and area 3.

Bits 1 and 0 —Not referenced

13.1.4 Bus Control Register 2 (BCR2)

The bus control register 2 (BCR2) is a 16-bit read/write register that sets the bus-size width of each area and selects whether an 8-bit port is used or not. It is initialized to H'3FF0 by a power-on reset, but is not initialized by a manual reset or by standby mode. Do not access external memory outside area 0 until BCR2 register initialization is complete.

Bit:	15	14	13	12	11	10	9	8
	—	—	A6SZ1	A6SZ0	A5SZ1	A5SZ0	A4SZ1	A4SZ0
Initial value:	0	0	1	1	1	1	1	1
R/W:	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
	A3SZ1	A3SZ0	A2SZ1	A2SZ0	—	—	—	—
Initial value:	1	1	1	1	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R	R	R	R

Bits 15 to 8 and 5 to 0 —Not referenced

Bits 7 and 6—Area 3 Bus Size Specification (A3SZ1, A3SZ0): Specifies the bus sizes of physical space area 3.

Bit 7: A3SZ1	Bit 6: A3SZ0	Port A/B	Description
0	0	Unused	Reserved (Setting disabled)
	1		Reserved (Setting disabled)
1	0		16-bit bus width
	1		32-bit bus width
0	0	Used	Reserved (Setting disabled)
	1		Reserved (Setting disabled)
1	0		16-bit bus width
	1		Reserved (Setting disabled)

13.1.5 Wait State Control Register 1 (WCR1)

Wait state control register 1 (WCR1) is a 16-bit read/write register that specifies the number of idle (wait) state cycles inserted for each area. For some memories, the drive of the data bus may not be turned off quickly even when the read signal from the external device is turned off. This can result in conflicts between data buses when consecutive memory accesses are to different memories or when a write immediately follows a memory read. This LSI automatically inserts idle states equal to the number set in WCR1 in those cases.

WCR1 is initialized to H'3FF3 by a power-on reset. It is not initialized by a manual reset or by standby mode.

Bit:	15	14	13	12	11	10	9	8
	WAIT SEL	—	A6IW1	A6IW0	A5IW1	A5IW0	A4IW1	A4IW0
Initial value:	0	0	1	1	1	1	1	1
R/W:	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
	A3IW1	A3IW0	A2IW1	A2IW0	—	—	A0IW1	A0IW0
Initial value:	1	1	1	1	0	0	1	1
R/W:	R/W	R/W	R/W	R/W	R	R	R/W	R/W

Bits 15 to 8 and 5 to 0 —Not referenced

Bits 7 and 6— Area 3 Idle Setting between Cycles (A3IW1, A3IW0): Specifies the number of idle state cycles to insert between bus cycles when switching from a read address in area 3 of the physical space to a write address in another space or within the same space.

Bit 7: A3IW1	Bit 6: A3IW0	Description	
0	0	1 idle state cycle inserted	
	1	1 idle state cycle inserted	
1	0	2 idle state cycle inserted	
	1	3 idle state cycle inserted	(Initial value)

13.1.6 Wait State Control Register 2 (WCR2)

Wait state control register 2 (WCR2) is a 16-bit read/write register that specifies the number of wait state cycles inserted for each area. It also specifies the pitch of data access for burst memory accesses. This allows direct connection of even low-speed memories without an external circuit.

WCR2 is initialized to H'FFFF by a power-on reset. It is not initialized by a manual reset or in standby mode.

Bit:	15	14	13	12	11	10	9	8
	A6W2	A6W1	A6W0	A5W2	A5W1	A5W0	A4W2	A4W1
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

Bit:	7	6	5	4	3	2	1	0
	A4W0	A3W1	A3W0	A2W1	A2W0	A0W2	A0W1	A0W0
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

Bits 15 to 7 and 4 to 0 —Not referenced

Bits 6 and 5— Area 3 Wait Control (A3W1, A3W0): Specifies the CAS latency for the SDRAM of area 3 of the physical space.

Bit 6: A3W1	Bit 5: A3W0	Description	
		SDRAM CAS Latency	
0	0	1	
	1	1	
1	0	2	
	1	3	(Initial value)

13.1.7 Individual Memory Control Register (MCR)

The individual memory control register (MCR) is a 16-bit read/write register that specifies $\overline{RAS}$ and $\overline{CAS}$ timing and burst control for synchronous DRAM (areas 2 and 3), specifies address multiplexing, and controls refresh. This enables direct connection of synchronous DRAM without external circuits.

The MCR is initialized to H'0000 by power-on resets, but is not initialized by manual resets or standby mode. The bits TPC1 to TPC0, RCD1 to RCD0, TRWL1 to TRWL0, TRAS1 to TRAS0, and AMX3 to AMX0 are written to at the initialization after a power-on reset and should not be modified again. When RFSH and RMODE are written to, write the same values to the other bits. When using synchronous DRAM, do not access areas 2 and 3 until this register initialization is complete.

Bit:	15	14	13	12	11	10	9	8
	TPC1	TPC0	RCD1	RCD0	TRWL1	TRWL0	TRAS1	TRAS0
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	6	5	4	3	2	1	0
	—	AMX3	AMX2	AMX1	AMX0	RFSH	RMODE	—
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 15 and 14—RAS Precharge Time (TPC1, TPC0): When synchronous DRAM interface is selected, these bits set the minimum number of cycles until output of the next bank-active command after precharge.

Bit 15: TPC1	Bit 14: TPC0	Description	
0	0	1 cycle	(Initial value)
	1	2 cycles	
1	0	3 cycles	
	1	4 cycles	

Bits 13 and 12—RAS–CAS Delay (RCD1, RCD0): When synchronous DRAM interface is selected, these bits set the bank active read/write command delay time.

Bit 13: RCD1	Bit 12: RCD0	Description	
0	0	1 cycle	(Initial value)
	1	2 cycles	
1	0	3 cycles	
	1	4 cycles	

Bits 11 and 10—Write-Precharge Delay (TRWL1, TRWL0): The TRWL bits set the synchronous DRAM write-precharge delay time. This designates the time between the end of a write cycle and the automatic precharge activation. After the write cycle, the next bank-active command is not issued for the period TPC + TRWL.

Bit 11: TRWL1	Bit 10: TRWL0	Description	
0	0	1 cycle	(Initial value)
	1	2 cycles	
1	0	3 cycles	
	1	Reserved (Setting disabled)	

Bits 9 and 8—$\overline{\text{CAS}}$-Before-$\overline{\text{RAS}}$ Refresh $\overline{\text{RAS}}$ Assert Time (TRAS1, TRAS0): When synchronous DRAM interface is selected, no bank-active command is issues during the period TPC + TRAS after an auto-refresh command.

Bit 9: TRAS1	Bit 8: TRAS0	Description	
0	0	2 cycles	(Initial value)
	1	3 cycles	
1	0	4 cycles	
	1	5 cycles	

Bit 7—Reserved: This bit is always read as 0 and should only be written with 0.

Bits 6 to 3—Address Multiplex (AMX3 , AMX2, AMX1, AMX0): The AMX bits specify address multiplexing for synchronous DRAM.

Bit6: AMX3	Bit5: AMX2	Bit 4: AMX1	Bit 3: AMX0	Description
1	1	0	1	When using a 16-bit bus width, the row address begins with A10. When using a 32-bit bus width, it begins with A11. (The A10 value is output at A1 when the row address is output. 4M × 16-bit × 4-bank products)
		1	0	When using a 16-bit bus width, the row address begins with A11. (The A11 value is output at A1 when the row address is output. 8M × 16-bit × 4-bank products)*1
0	1	0	0	When using a 16-bit bus width, the row address begins with A9. When using a 32-bit bus width, it begins with A10. (The A9 value is output at A1 when the row address is output. 1M × 16-bit × 4-bank products)
			1	When using a 16-bit bus width, the row address begins with A10. When using a 32-bit bus width, it begins with A11. (The A10 value is output at A1 when the row address is output. 2M × 8-bit products)
			0	The row address begins with A11 when bus width is 32 bit. *2 (The A11 value is output at A1 when the row address is output. 4M × 8-bit × 4-bank products)
		1	1	When using a 16-bit bus width, the row address begins with A9. When using a 32-bit bus width, it begins with A10. (The A9 value is output at A1 when the row address is output. 512K × 32-bit × 4-bank products) *2
	0	0	0	Reserved. AMX3 to AMX0 must be set to *1*** before accessing synchronous DRAM memory. (Initial value)
Other values				Reserved (Illegal setting)

Notes: 1. Can be set only when using a 16-bit bus width.
2. Can be set only when using a 32-bit bus width.

Bits 2 and 1—Not referenced

Bit 0—Reserved: This bit is always read as 0 and should only be written with 0.

13.2 LBSC Operation

13.2.1 Bus Sharing Architecture

LCDC and USB Host Controller can share the system memory with CPU and DMA Controller, so these bus masters are able to work without any independent external memory and have huge available memory space up to 64 Mbyte at area 3.

Since each LCDC, USB Host Controller, CPU, and DMA Controller can access area 3 individually. Set addresses for each controller to avoid address sharing.

13.2.2 Usable System Memory

LBSC works at below memories.

Memory area	Area3
Memory type	Synchronous DRAM
Bus width	16 or 32 bits
Burst length	1 to 4 burst (USBH)
	4 to 32 burst (LCDC) with 32-bit bus width, 8 to 64 burst (LCDC) with 16-bit bus width

13.2.3 Bus Arbitration

LBSC accepts a request that comes from LCDC or USB Host at a same time without any prioritization to each module. LBSC tries to get bus right from BSC at any time when it get a request from LCDC or USB Host. Once BSC gives LBSC a right, LCDC or BSC can access external memory directly. The arbiter of LBSC gives a bus right to LCDC or USB Host as even.

13.2.4 LCDC Li Bus Access

While displaying images, the LCDC continuously reads data from the system memory with a 32 burst length. The LCDC burst length is specified by a register in the LCDC. If the data length is shorter than 32 burst length, such as the case for the edge of LCD panel, the LCDC uses a shorter burst length.

13.2.5 USBH Li Bus Access

USB Host issues 1 to 4 burst request to LBSC as normal read or write action. Since the burst length issued by USB Host is occasionally changed as FIFO pointer rises up or falls, it is not supposed as 4 burst exactly.

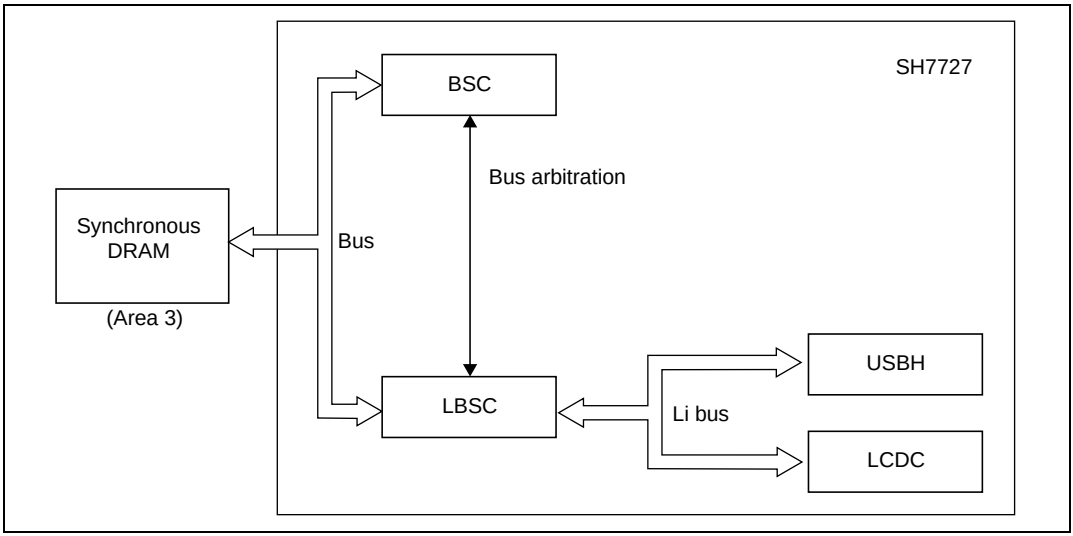


Figure 13.1 Block Diagram of Li Bus Architecture

13.2.6 Setting of DMA Transfer with Bus Arbitration of Other Module

This LSI has five types of bus master: CPU, DMAC and Refresh (BSC system), and LCDC and USBH (LBSC system). The following priority order is set for these buses.

1. The BSC and LBSC systems are the same in priority level.
2. In the BSC system, Refresh has the highest priority.
3. Between CPU and DMAC, DMAC is higher in priority when DMA burst setting is made. In cycle steal, CPU and DMAC are the same in priority level.
4. LCDC and USBH are the same priority level in the LBSC system.

In cycle steal, the priority level of DMA transfer is very low. Therefore, if the DMAC transfer speed may cause problems, it is recommended to use the level-input burst transfer setting for DMAC, especially when $\overline{\text{DREQ}}$ signals from an external device can be negated.

Section 14 Direct Memory Access Controller (DMAC)

14.1 Overview

This chip includes a four-channel direct memory access controller (DMAC). The DMAC can be used in place of the CPU to perform high-speed transfers between external devices that have DACK (transfer request acknowledge signal), external memory, memory-mapped external devices, and on-chip supporting modules (SIOF, SCIF, USB function, and A/D converter). Using the DMAC reduces the burden on the CPU and increases overall operating efficiency.

14.1.1 Features

The DMAC has the following features.

- Four channels
- 4-GB physical address space
- Selectable data transfer length: 8-bit, 16-bit, 32-bit, or 16-byte transfer (In 16-byte transfer, four 32-bit reads are executed, followed by four 32-bit writes.)
- Maximum of 16 M times of transfers (16777216 times)
- Address mode: Dual address mode and single address mode are supported. In addition, direct address transfer mode or indirect address transfer mode can be selected.
  - Dual address mode transfer: Both the transfer source and transfer destination are accessed by address. Dual address mode has direct address transfer mode and indirect address transfer mode.

    Direct address transfer mode: The values specified in the DMAC registers indicates the transfer source and transfer destination. Two bus cycles are required for one data transfer.

    Indirect address transfer mode: Data is transferred with the address stored prior to the address specified in the transfer source address in the DMAC. Other operations are the same as those of direct address transfer mode. This function is only valid in channel 3. Four bus cycles are requested for one data transfer.
  - Single address mode transfer: Either the transfer source or transfer destination peripheral device is accessed (selected) by means of the DACK signal, and the other device is accessed by address. One transfer unit of data is transferred in one bus cycle.
- Channel functions: Transfer mode that can be specified is different in each channel.
  - Channel 0: Can accept requests from peripheral modules and external requests.
  - Channel 1: Can accept requests from peripheral modules.
  - Channel 2: Can accept requests from peripheral modules. This channel has a source address reload function, which reloads a source address for each 4 transfers.

— Channel 3: Can accept requests from peripheral modules. Direct address transfer mode or indirect address transfer mode can be specified.

- Reload function: The value that was specified in the source address register can be automatically reloaded every 4 DMA transfers. This function is only valid in channel 2.
- Three types of transfer requests
  - — External requests (From two $\overline{\text{DREQ}}$ pins (channels 0 only). $\overline{\text{DREQ}}$ can be detected either by falling edge or by low level)
  - — On-chip module requests (Requests from on-chip supporting modules such as serial communications interface (SIOF, SCIF), A/D converter (A/D) and a timer (CMT) . This request can be accepted in all the channels)
  - — Auto requests (the transfer request is generated automatically within the DMAC)
- Selectable bus modes: Cycle-steal mode or burst mode
- Selectable channel priority levels:

  Fixed mode: The channel priority is fixed.

  Round-robin mode: The priority of the channel in which the execution request was accepted is made the lowest.
- Interrupt request: An interrupt request can be generated to the CPU after the specified number of times of transfers.

14.1.2 Block Diagram

Figure 14.1 is a block diagram of the DMAC.

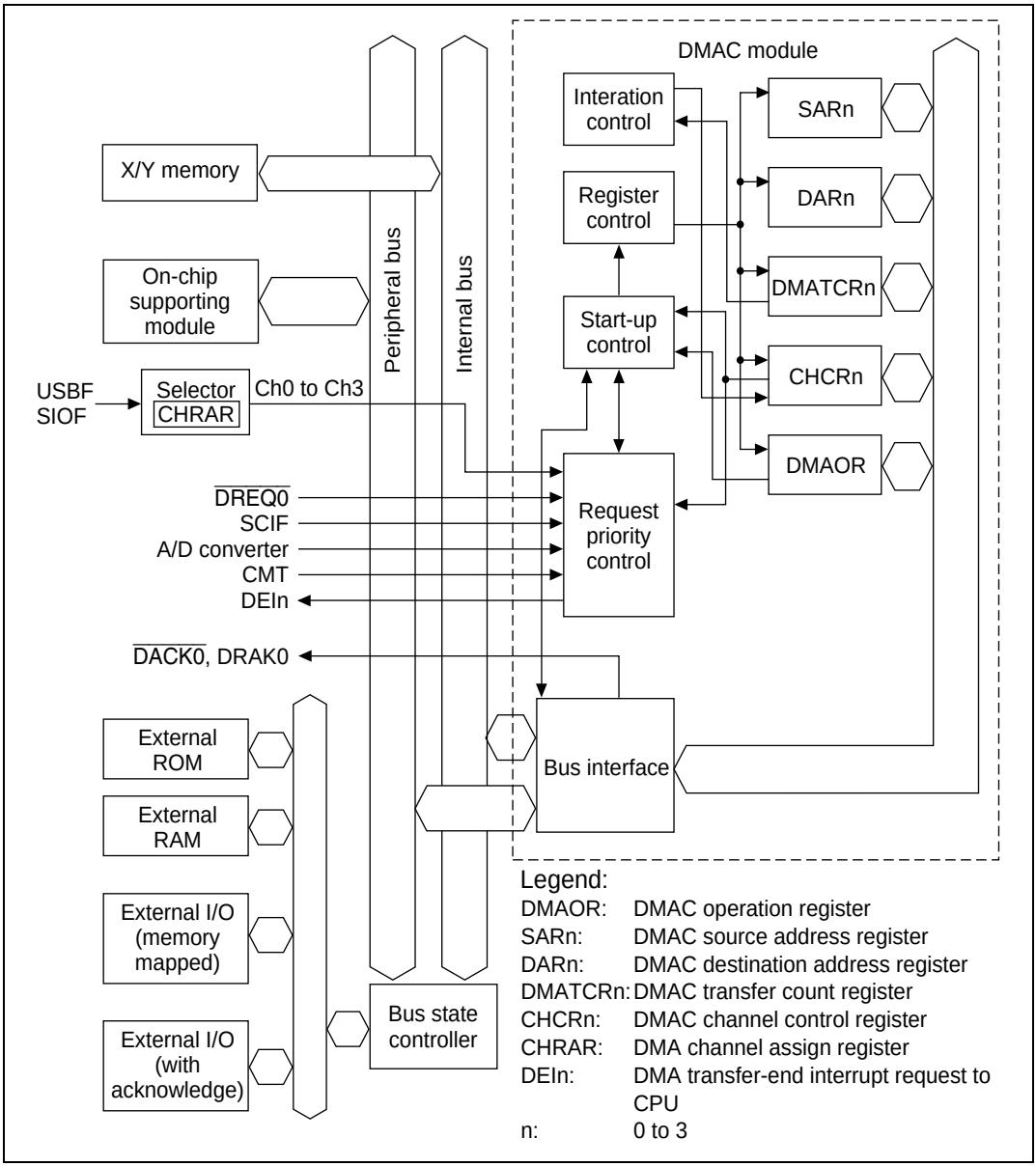


Figure 14.1 DMAC Block Diagram

14.1.3 Pin Configuration

Table 14.1 shows the DMAC pins.

Table 14.1 Pin Configuration

Channel	Name	Symbol	I/O	Function
0	DMA transfer request	$\overline{\text{DREQ0}}$	Input	DMA transfer request input from external device to channel 0
	DREQ acknowledge	DACK0	Output	Strobe output to an external I/O at DMA transfer request from external device to channel 0
	DMA request acknowledge	DRAK0	Output	Output showing that $\overline{\text{DREQ0}}$ has been accepted

14.1.4 Register Configuration

Table 14.2 summarizes the DMAC registers. DMAC has a total of 18 registers, four registers for each channel and two registers for controlling all channels.

Table 14.2 DMAC Registers

Channel	Name	Abbreviation	R/W	Initial Value	Address	Register Size	Access Size
0	DMA source address register 0	SAR0	R/W	Undefined	H'04000020 (H'A4000020)*4	32 bits	16, 32*2
	DMA destination address register 0	DAR0	R/W	Undefined	H'04000024 (H'A4000024)*4	32 bits	16, 32*2
	DMA transfer count register 0	DMATCR0	R/W	Undefined	H'04000028 (H'A4000028)*4	24 bits	16, 32*3
	DMA channel control register 0	CHCR0	R/W*1	H'00000000	H'0400002C (H'A400002C)*4	32 bits	8, 16, 32*2
1	DMA source address register 1	SAR1	R/W	Undefined	H'04000030 (H'A4000030)*4	32 bits	16, 32*2
	DMA destination address register 1	DAR1	R/W	Undefined	H'04000034 (H'A4000034)*4	32 bits	16, 32*2
	DMA transfer count register 1	DMATCR1	R/W	Undefined	H'04000038 (H'A4000038)*4	24 bits	16, 32*3
	DMA channel control register 1	CHCR1	R/W*1	H'00000000	H'0400003C (H'A400003C)*4	32 bits	8, 16, 32*2

Channel	Name	Abbrevi-ation	R/W	Initial Value	Address	Register Size	Access Size
2	DMA source address register 2	SAR2	R/W	Undefined	H'04000040 (H'A4000040)*4	32 bits	16, 32*2
	DMA destination address register 2	DAR2	R/W	Undefined	H'04000044 (H'A4000044)*4	32 bits	16, 32*2
	DMA transfer count register 2	DMATCR2	R/W	Undefined	H'04000048 (H'A4000048)*4	24 bits	16, 32*3
	DMA channel control register 2	CHCR2	R/W*1	H'00000000	H'0400004C (H'A400004C)*4	32 bits	8, 16, 32*2
3	DMA source address register 3	SAR3	R/W	Undefined	H'04000050 (H'A4000050)*4	32 bits	16, 32*2
	DMA destination address register 3	DAR3	R/W	Undefined	H'04000054 (H'A4000054)*4	32 bits	16, 32*2
	DMA transfer count register 3	DMATCR3	R/W	Undefined	H'04000058 (H'A4000058)*4	24 bits	16, 32*3
	DMA channel control register 3	CHCR3	R/W*1	H'00000000	H'0400005C (H'A400005C)*4	32 bits	8, 16, 32*2
Shared	DMA operation register	DMAOR	R/W*1	H'0000	H'04000060 (H'A4000060)*4	16 bits	8, 16*2
	DMA channel assign register	CHRAR	R/W	H'0000	H'0400022A (H'A400022A)*4	16 bits	16

Notes: These registers are located in area 1 of physical space. Therefore, when the cache is on, either access these registers from the P2 area of logical space or else make an appropriate setting using the MMU so that these registers are not cached.

1. Only a write of 0 after a read of 1 to clear a flag is enabled for bit 1 in CHCR0 to CHCR3 and bits 1 and 2 in DMAOR.
2. If SAR0 to SAR3, DAR0 to DAR3, and CHCR0 to CHCR3 are accessed in 16 bits, the 16 bit values that were not accessed are held.
3. DMATCR comprises the 24 bits from bit 0 to bit 23. The upper 8 bits, bits 24 to 31, cannot be written with 1 and are always read as 0.
4. When address translation by the MMU does not apply, the address in parentheses should be used.

14.2 Register Descriptions

14.2.1 DMA Source Address Registers 0 to 3 (SAR0 to SAR3)

Bit:	31	30	29	28	27	26	25	24
Initial value:	—	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	23	22	21	20	…	0
					…	
Initial value:	—	—	—	—	…	—
R/W:	R/W	R/W	R/W	R/W	…	R/W

The DMA source address registers 0 to 3 (SAR0 to SAR3) are 32-bit read/write registers that specify the source address of a DMA transfer. During a DMA transfer, these registers indicate the next source address.

To transfer data in 16 bits or in 32 bits, specify the address on the 16-bit or 32-bit boundary. When transferring data in 16-byte units, a 16-byte boundary (address 16n) must be set for the source address value. If any other address is specified, correct operation is not guaranteed.

Initial values are undefined after a reset. The previous values are held in standby mode.

14.2.2 DMA Destination Address Registers 0 to 3 (DAR0 to DAR3)

Bit:	31	30	29	28	27	26	25	24
Initial value:	—	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	23	22	21	20	…	0
					…	
Initial value:	—	—	—	—	…	—
R/W:	R/W	R/W	R/W	R/W	…	R/W

The DMA destination address registers 0 to 3 (DAR0 to DAR3) are 32-bit read/write registers that specify the destination address of a DMA transfer. These registers include count functions, and during a DMA transfer, these registers indicate the next destination address.

To transfer data in 16 bits or in 32 bits, specify the address on the 16-bit or 32-bit boundary. When transferring data in 16-byte units, always set a value at a 16-byte boundary (16n address) as the destination address. If any other address is specified, correct operation is not guaranteed.

Initial values are undefined after a reset. The previous value is held in standby mode.

14.2.3 DMA Transfer Count Registers 0 to 3 (DMATCR0 to DMATCR3)

Bit:	31	30	29	28	27	26	25	24
Initial value:	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R

Bit:	23	22	21	20	...	0
					...	
Initial value:	—	—	—	—	...	—
R/W:	R/W	R/W	R/W	R/W	...	R/W

The DMA transfer count registers 0 to 3 (DMATCR0 to DMATCR3) are 24-bit read/write registers that specify the DMA transfer count (bytes, words, or longwords). The number of transfers is 1 when the setting is H'000001, and 16777216 (the maximum) when H'000000 is set. During a DMA transfer, these registers indicate the remaining transfer count.

To transfer data in 16 bytes, one 16-byte transfer (128 bits) counts one.

The upper eight bits in DMATCR are always read as 0 and should only be written with 0.

Initial values are undefined after a reset. The previous value is held in standby mode.

14.2.4 DMA Channel Control Registers 0 to 3 (CHCR0 to CHCR3)

Bit:	31	...	21	20	19	18	17	16
	—	...	—	DI	RO	RL	AM	AL
Initial value:	0	...	0	0	0	0	0	0
R/W:	R	...	R	(R/W)*2	(R/W)*2	(R/W)*2	(R/W)*2	(R/W)*2

Bit:	15	14	13	12	11	10	9	8
	DM1	DM0	SM1	SM0	RS3	RS2	RS1	RS0
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	6	5	4	3	2	1	0
	—	DS	TM	TS1	TS0	IE	TE	DE
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	(R/W)*2	R/W	R/W	R/W	R/W	R/(W)*1	R/W

Notes: 1. Only a write of 0 after a read of 1 is enabled for the TE bit.
2. DI, RO, RL, AM, AL, and DS bits are not included in some channels.

The DMA channel control registers 0 to 3 (CHCR0 to CHCR3) are 32-bit read/write registers that specify operation mode, transfer method, or others in each channel. Writing to bits 31 to 21 and 7 in this register is invalid, and these bits are always read as 0.

Bit 20 is only used in CHCR3. It is not used in CHCR0 to CHCR2. Consequently, writing to this bit is invalid in CHCR0 to CHCR2, and this bit is always read as 0.

Bit 19 is only used in CHCR2. It is not used in CHCR0, CHCR1, and CHCR3. Consequently, writing to this bit is invalid in CHCR0, CHCR1, and CHCR3, and this bit is always read as 0.

Bits 6 and 16 to 18 are only used in CHCR0 and CHCR1. They are not used in CHCR2 and CHCR3. Consequently, writing to these bits is invalid in CHCR2 and CHCR3, and these bits are always read as 0.

These registers are initialized to 0 after a power-on reset. The previous values are held in standby mode.

Bits 31 to 21—Reserved: These bits are always read as 0 and should only be written with 0.

Bit 20—Direct/Indirect Selection (DI): DI selects direct address mode operation or indirect address mode operation for a channel 3 source address.

This bit is only valid in CHCR3. This bit in CHCR0 to CHCR2 is always read as 0 and should only be written with 0.

When using 16-byte transfer, direct address mode must be specified. Operation is not guaranteed if indirect address mode is specified.

Bit 20: DI	Description	
0	Direct address mode	(Initial value)
1	Indirect address mode	

Bit 19—Source Address Reload (RO): RO selects whether the source address initial value is reloaded in channel 2.

This bit is only valid in CHCR2. This bit in CHCR0, CHCR1, and CHCR3 is always read as 0 and should only be written with 0.

When using 16-byte transfer, this bit must be cleared to 0, specifying non-reloading. Operation is not guaranteed if reloading is specified.

Bit 19: RO	Description	
0	A source address is not reloaded	(Initial value)
1	A source address is reloaded	

Bit 18—Request Check Level (RL): RL specifies the DRAK (acknowledge of $\overline{\text{DREQ}}$) signal output is high active or low active.

This bit is only valid in CHCR0. This bit in CHCR1, CHCR2 and CHCR3 is always read as 0 and should only be written with 0.

Bit 18: RL	Description	
0	Low-active output of DRAK	(Initial value)
1	High-active output of DRAK	

Bit 17—Acknowledge Mode (AM): AM specifies whether DACK is output in data read cycle or in data write cycle in dual address mode.

In single address mode, DACK is always output regardless of this bit specification.

This bit is only valid in CHCR0. This bit in CHCR1, CHCR2 and CHCR3 is always read as 0 and should only be written with 0.

Bit 17: AM	Description	
0	DACK output in read cycle	(Initial value)
1	DACK output in write cycle	

Bit 16—Acknowledge Level (AL): AL specifies the $\overline{\text{DACK}}$ (acknowledge) signal output is high active or low active.

This bit is only valid in CHCR0. This bit in CHCR1, CHCR2 and CHCR3 is always read as 0 and should only be written with 0.

Bit 16: AL	Description	
0	Low-active output of DACK	(Initial value)
1	High-active output of DACK	

Bits 15 and 14—Destination Address Mode 1, 0 (DM1 and DM0): DM1 and DM0 select whether the DMA destination address is incremented, decremented, or fixed.

Bit 15: DM1	Bit 14: DM0	Description	
0	0	Fixed destination address*	(Initial value)
	1	Destination address is incremented (+1 in 8-bit transfer, +2 in 16-bit transfer, +4 in 32-bit transfer, +16 in 16-byte transfer)	
1	0	Destination address is decremented (–1 in 8-bit transfer, –2 in 16-bit transfer, –4 in 32-bit transfer; illegal setting in 16-byte transfer)	
	1	Reserved (illegal setting)	

Note: * This setting cannot be used to perform 16-byte transfers with a destination in X/Y memory.

Bits 13 and 12—Source Address Mode 1, 0 (SM1 and SM0): SM1 and SM0 select whether the DMA source address is incremented, decremented, or fixed.

Bit 13: SM1	Bit 12: SM0	Description
0	0	Fixed source address* (Initial value)
	1	Source address is incremented (+1 in 8-bit transfer, +2 in 16-bit transfer, +4 in 32-bit transfer, +16 in 16-byte transfer)
1	0	Source address is decremented (–1 in 8-bit transfer, –2 in 16-bit transfer, –4 in 32-bit transfer; illegal setting in 16-byte transfer)
	1	Reserved (illegal setting)

Note: * This setting cannot be used to perform 16-byte transfers with a destination in X/Y memory.

If the transfer source is specified in indirect address, specify the address (indirect address) where the address of data to be transferred is stored as data, in source address register 3 (SAR3).

Specification of SAR3 increment or decrement in indirect address mode depends on SM1 and SM0 settings. In this case, however, the SAR3 increment or decrement value is +4, –4, or fixed to 0 regardless of the transfer data size specified in TS1 and TS0.

Bits 11 to 8—Resource 3 to 0 (RS3 to RS0): RS3 to RS0 specify which transfer requests will be sent to the DMAC.

Bit 11: RS3	Bit 10: RS2	Bit 9: RS1	Bit 8: RS0	Description
0	0	0	0	External request*1, dual address mode (Initial value)
			1	Illegal setting
		1	0	External request*1/Single address mode External address space → external device with DACK
			1	External request*1/Single address mode External device with DACK → external address space
	1	0	0	Auto request
			1	Illegal setting
		1	0	Illegal setting
			1	Illegal setting
1	0	0	0	Select DMA request expansion*3
			1	Illegal setting
		1	0	Illegal setting
			1	Illegal setting
	1	0	0	SCIF transmission*2
			1	SCIF reception*2
		1	0	Internal A/D*2
			1	CMT*2

Notes: 1. External request specification is valid only for channels 0. None of the request sources can be selected for channels 1, 2 and 3.

2. When using 16-byte transfer, the following settings must not be made:
   1100 SCIF transmission
   1101 SCIF reception
   1110 A/D converter
   1111 CMT
   Operation is not guaranteed if these settings are made.

3. When DMA transfer is provided with the USB function controller or SIOF, set RS3 to RS0 to 1000 and select a desired module with the CHRAR register.

Bit 6—$\overline{\text{DREQ}}$ Select (DS): DS selects the sampling method of the $\overline{\text{DREQ}}$ pin that is used in external request mode is detection in low level or at the falling edge.

This bit is only valid in CHCR0. This bit in CHCR1, CHCR2 and CHCR3 is always read as 0 and should only be written with 0.

Also, it should be cleared to 0 (low-level detection) if an on-chip supporting module is specified as a transfer request source in channel 0.

Bit 6: DS	Description	
0	$\overline{\text{DREQ}}$ detected in low level	(Initial value)
1	$\overline{\text{DREQ}}$ detected at falling edge	

Bit 5—Transmit Mode (TM): TM specifies the bus mode when transferring data.

Bit 5: TM	Description	
0	Cycle steal mode	(Initial value)
1	Burst mode	

Bits 4 and 3—Transmit Size 1, 0 (TS1 and TS0): TS1 and TS0 specify the size of data to be transferred.

Bit 4: TS1	Bit 3: TS0	Description	
0	0	Byte size (8 bits)	(Initial value)
0	1	Word size (16 bits)	
1	0	Longword size (32 bits)	
1	1	16-byte unit (4 longword transfers)	

Bit 2—Interrupt Enable (IE): Setting this bit to 1 generates an interrupt request when the number of times of data transfers specified with DMATCR has completed (TE = 1).

Bit 2: IE	Description	
0	Interrupt request is not generated even when data transfer ends by the specified count	(Initial value)
1	Interrupt request is generated when data transfer ends by the specified count	

Bit 1—Transfer End (TE): TE is set to 1 when data transfer ends by the count specified in DMATCR. At this time, if the IE bit is set to 1, an interrupt request is generated.

Before this bit is set to 1, if data transfer ends due to an NMI interrupt, a DMAC address error, or clearing the DE bit or the DME bit in DMAOR, this bit is not set to 1. Even if the DE bit is set to 1 while this bit is set to 1, transfer is not enabled.

Bit 1: TE	Description
0	Data transfer does not end by the count specified in DMATCR (Initial value) Clear condition: Writing 0 after TE = 1 read at power-on reset or manual reset
1	Data transfer ends by the specified count

Bit 0—DMAC Enable (DE): DE enables channel operation.

Bit 0: DE	Description
0	Disables channel operation (Initial value)
1	Enables channel operation

If the auto request is specified in RS3 to RS0, transfer starts when this bit is set to 1. For an external request or an on-chip module request, transfer starts if a transfer request is generated after this bit is set to 1. Clearing this bit during transfer can terminate transfer.

Even if the DE bit is set, transfer is not enabled when the TE bit is 1, the DME bit in DMAOR is 0, or the NMIF bit or AE bit in DMAOR is 1.

14.2.5 DMA Channel Request Assign Register (CHRAR)

The DMA channel request assign register (CHRAR) is a 16-bit read/write registers that assign requests from USBF or SIOF to each DMA channel, to each expanded DMA. It is initialized to 0 at power-on reset, or in hardware standby mode or software standby mode. These register values are initialized to 0s after a power-on reset. The previous value is held in standby mode.

Bit:	15,14,13,12	11,10,9,8
	CH3RID3 to CH0RID0	CH2RID3 to CH2RID0
Initial value:	0	0
R/W:	R/W	R/W

Bit:	7,6,5,4	3,2,1,0
	CH1RID3 to CH1RID0	CH0RID3 to CH0RID0
Initial value:	0	0
R/W:	R/W	R/W

Bits 15 to 12 —DMA Channel 3 Request Assign 3 to 0 (CH3RID3 to CH3RID0): These bits select DMA requests from DMA channel 3.

Bits 15 to 12: CH3RID3 to CH3RID0	**Description**
0000	Unused (Initial value)
0001	USBF (USB function) reception requests to the DMA are selected from channel 3
0010	USBF (USB function) transmission requests to the DMA are selected from channel 3
1001	SIOF reception requests to the DMA are selected from channel 3
1010	SIOF transmission requests to the DMA are selected from channel 3

Bits 11 to 8 — DMA Channel 2 Request Assign 3 to 0 (CH2RID3 to CH2RID0): These bits select DMA requests from DMA channel 2.

Bits 11 to 8: CH2RID3 to CH2RID0	Description
0000	Unused (Initial value)
0001	USBF (USB function) reception requests to the DMA are selected from channel 2
0010	USBF (USB function) transmission requests to the DMA are selected from channel 2
1001	SIOF reception requests to the DMA are selected from channel 2
1010	SIOF transmission requests to the DMA are selected from channel 2

Bits 7 to 4— DMA Channel 1 Request Assign 3 to 0 (CH1RID3 to CH1RID0): These bits select DMA requests from DMA channel 1.

Bits 7 to 4: CH1RID3 to CH1RID0	Description
0000	Unused (Initial value)
0001	USBF (USB function) reception requests to the DMA are selected from channel 1
0010	USBF (USB function) transmission requests to the DMA are selected from channel 1
1001	SIOF reception requests to the DMA are selected from channel 1
1010	SIOF transmission requests to the DMA are selected from channel 1

Bits 3 to 0— DMA Channel 0 Request Assign 3 to 0 (CH0RID3 to CH0RID0): These bits select DMA requests from DMA channel 0.

Bits 3 to 0: CH0RID3 to CH0RID0	Description
0000	Unused (Initial value)
0001	USBF (USB function) reception requests to the DMA are selected from channel 0
0010	USBF (USB function) transmission requests to the DMA are selected from channel 0
1001	SIOF reception requests to the DMA are selected from channel 0
1010	SIOF transmission requests to the DMA are selected from channel 0

14.2.6 DMA Operation Register (DMAOR)

Bit:	15	14	13	12	11	10	9	8
	—	—	—	—	—	—	PR1	PR0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	—	AE	NMIF	DME
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/(W)*	R/(W)*	R/W

Note: * Only a write of 0 after a read of 1 is enabled for the AE and NMIF bits.

The DMA operation register (DMAOR) is a 16-bit read/write register that controls the DMAC transfer mode.

This register is initialized to 0 by a power-on reset. The previous values are held in standby mode.

Bits 15 to 10—Reserved: These bits are always read as 0 and should only be written with 0.

Bits 9 and 8—Priority Mode 1, 0 (PR1 and PR0): PR1 and PR0 select the priority level between channels when transfer requests are generated for multiple channels simultaneously.

Bit 9: PR1	Bit 8: PR0	Description	
0	0	CH0 > CH1 > CH2 > CH3	(Initial value)
	1	CH0 > CH2 > CH3 > CH1	
1	0	CH2 > CH0 > CH1 > CH3	
	1	Round-robin	

Bits 7 to 3—Reserved: These bits are always read as 0 and should only be written with 0.

Bit 2—Address Error Flag (AE): AE indicates that an address error occurred during DMA transfer. If this bit is set during data transfer, transfers on all channels are suspended. The CPU cannot write 1 to this bit. This bit can only be cleared by writing of 0 after reading of 1.

Bit 2: AE	Description	
0	No DMAC address error. DMA transfer is enabled.	(Initial value)
	Clear conditions: When this bit is written with 0 after it is read as 1	
	By a power-on reset	
	By a manual reset	
1	DMAC address error. DMA transfer is disabled.	
	Setting condition: When a DMAC address error occurred	

Bit 1—NMI Flag (NMIF): NMIF indicates that an NMI interrupt occurred. This bit is set both in operating state and in halt state. The CPU cannot write 1 to this bit. This bit can only be cleared by writing of 0 after reading of 1.

Bit 1: NMIF	Description	
0	No NMI input. DMA transfer is enabled.	(Initial value)
	Clear conditions: When this bit is written with 0 after it is read as 1	
	By a power-on reset	
	By a manual reset	
1	NMI input. DMA transfer is disabled.	
	Setting condition: When an NMI interrupt is generated	

Bit 0—DMA Master Enable (DME): DME enables or disables DMA transfer for all channels. If the DME bit and the DE bit corresponding to each channel in CHCR are set to 1, transfer is enabled in the corresponding channel. If this bit is cleared during transfer, transfer in all the channels can be terminated.

Even if the DME bit is set, transfer is not enabled when the TE bit is 1 or the DE bit is 0 in CHCR, or the NMIF bit is 1 in DMAOR.

Bit 0: DME	Description	
0	Disable DMA transfer for all channels	(Initial value)
1	Enable DMA transfer for all channels	

14.3 Operation

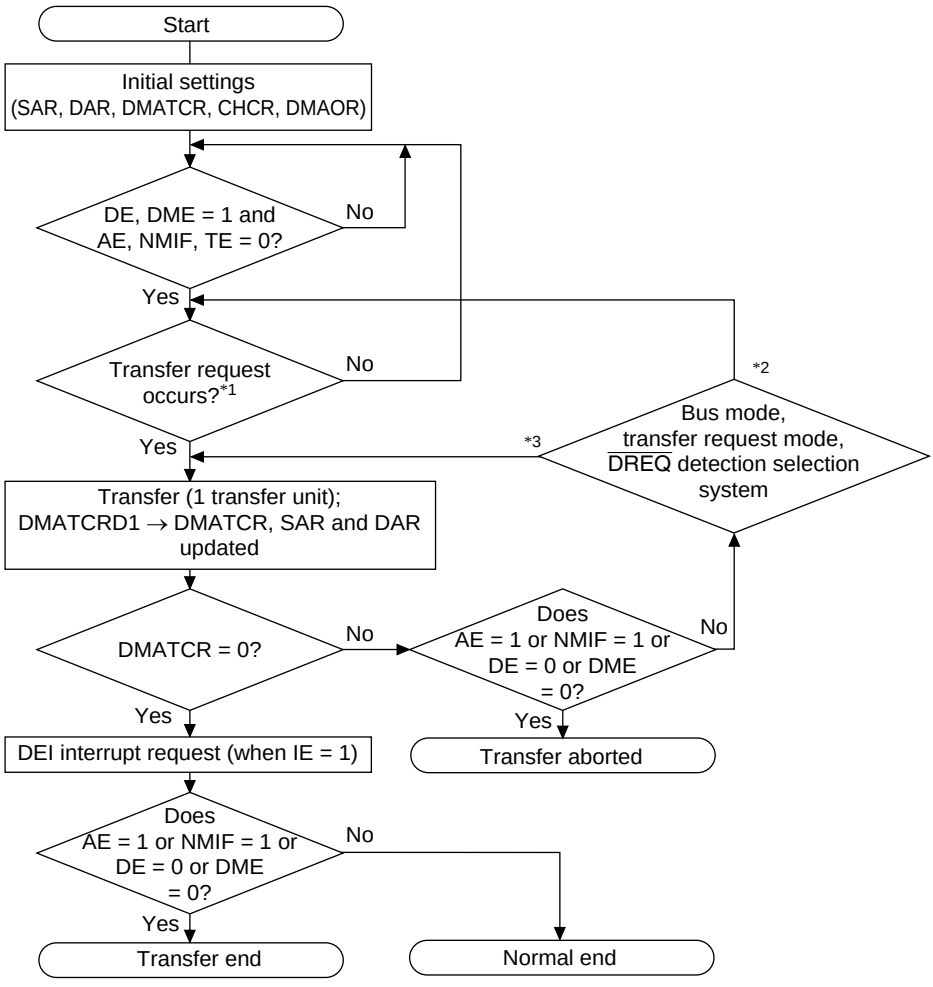
When a DMA transfer request is generated, the DMAC starts the transfer according to the predetermined channel priority order. When a transfer end condition is satisfied, it ends the transfer. Three types of transfer requests can be, auto request, external request, and on-chip module request. For the dual address mode, the direct address transfer mode and indirect address transfer mode are supported. For the bus mode, the burst mode or the cycle steal mode can be selected.

14.3.1 DMA Transfer Flow

When transfer conditions have been set to the DMA source address registers (SAR), DMA destination address registers (DAR), DMA transfer count registers (DMATCR), DMA channel control registers (CHCR), and DMA operation register (DMAOR), the DMAC transfers data according to the following procedure:

1. Checks that transfer is enabled (DE = 1, DME = 1, AE = 0, TE = 0, NMIF = 0)
2. When transfer is enabled and a transfer request is generated, the DMAC transfers 1 transfer unit of data (set with the TS0 and TS1 bits). In auto request mode, the transfer operation begins automatically when the DE bit and DME bit are set to 1. The DMATCR value will be decremented on each transfer. The actual transfer flows vary according to the address mode and bus mode.
3. When the specified number of transfers have been completed (when DMATCR reaches 0), the transfer operation ends normally. If the IE bit in CHCR is set to 1 at this time, a DEI interrupt is sent to the CPU.
4. When an NMI interrupt is generated, the transfer operation is aborted. The transfer operation is also aborted when the DE bit in CHCR or the DME bit in DMAOR is changed to 0.

Figure 14.2 shows a flowchart of this procedure.



Notes: 1. In auto-request mode, transfer begins when AE, NMIF and TE are all 0 and the DE and DME bits are set to 1.
2. $\overline{\text{DREQ}}$ = level detection in burst mode (external request) or cycle-steal mode.
3. $\overline{\text{DREQ}}$ = edge detection in burst mode (external request), or auto-request mode in burst mode.

Figure 14.2 DMAC Transfer Flowchart

14.3.2 DMA Transfer Requests

DMA transfer requests are basically generated in either the data transfer source or destination, but they can also be generated by devices and on-chip supporting modules that are neither the source nor the destination. There are three types of transfer requests, an auto request, an external request and an on-chip module request. The request mode is selected with the RS3 to RS0 bits in the DMA channel control registers 0 to 3 (CHCR0 to CHCR3).

Auto-Request Mode: When no transfer request signal is input from an external source, such as transfer between memories or between memory and an on-chip supporting module on which a transfer request is disabled, the auto-request mode allows the DMAC to automatically generate a transfer request signal internally. When the DE bits in CHCR0 to CHCR3 and the DME bit in DMAOR are set to 1, a transfer is started. At this time, the TE bits in CHCR0 to CHCR3 and the NMIF bit in DMAOR should be all 0.

External Request Mode: A transfer is started by the transfer request signal ($\overline{\text{DREQ}}$) from an external device. Choose one of the modes shown in table 14.3 according to the application system. If $\overline{\text{DREQ}}$ is input when the DMA transfer is enabled (DE = 1, DME = 1, TE = 0, NMIF = 0), a DMA transfer starts. Select whether $\overline{\text{DREQ}}$ is detected on the falling edge or low level with the DS bit in CHCR0 (level detection when DS = 0, edge detection when DS = 1).

The source of the transfer request does not have to be the data transfer source or destination.

Table 14.3 Selecting External Request Modes with the RS Bits

RS3	RS2	RS1	RS0	Address Mode	Source	Destination
0	0	0	0	Dual address mode	Arbitrary*	Arbitrary*
		1	0	Single address mode	External memory, memory-mapped external device	External device with DACK
			1		External device with DACK	External memory, memory-mapped external device

Note: * External memory, memory-mapped external device, on-chip memory, on-chip supporting module (excluding DMAC, UBC, and BSC)

On-Chip Module Request Mode: A transfer is started by the transfer request signal (interrupt request signal) from an on-chip supporting module. This mode cannot be set in case of 16-byte transfer. There are eight types of transfer request signals, a receive data full interrupt (RXI) and a transmit data empty interrupt (TXI) from the serial communication interface (SCIF), an A/D conversion end interrupt (ADI) from the A/D converter, an compare-match timer interrupt (CMI)

from CMT, a transmit request (TDREQ) and a receive request (RDREQ) from SIOR, and a transmit request (DREQN1) and a receive request (DREQN0) from USBF. TDREQ, RDREQ, DREQN1, and DREQN0 are supported for expansion.

When the DMA transfer is enabled (DE = 1, DME = 1, TE = 0, and NMIF = 0) in this mode, a transfer is started by the transfer request signal input. The source of the transfer request does not have to be the data transfer source or destination.

When RXI2 is set as a transfer request, however, the transfer source must be the SCIF's receive data register (RDR2). Likewise, when TXI2 is set as a transfer request, the transfer source must be the SCIF's transmit data register (TDR2). In addition, when the transfer requester is the A/D converter, the data transfer source must be the A/D data register (ADDR).

Table 14.4 Selection of On-Chip Module Request Modes Using RS3 to RS0 Bits

RS3	RS2	RS1	RS0	DMA Transfer Request Source		DMA Transfer Request Signal	Source	Destination	Bus Mode
1	0	0	0	Expansion	USBF receiver	DREQN[0] (DMA transfer request output)	EPDR1	Arbitrary*	Cycle steal
					USBF transmitter	DREQN[1] (DMA transfer request output)	Arbitrary*	EPDR2	Cycle steal
					SIOF receiver	RDREQ (receive-data transfer request)	SIRDR	Arbitrary*	Cycle steal
					SIOF transmitter	TDRQ (transmit-data transfer request)	Arbitrary*	SITDR	Cycle steal
	1	0	0	SCIF transmitter		TXI2 (SCIF transmit data empty interrupt)	Arbitrary*	TDR2	Cycle steal
		0	1	SCIF receiver		RXI2 (SCIF receive data full interrupt)	RDR2	Arbitrary*	Cycle steal
		1	0	A/D converter		ADI (A/D conversion end interrupt)	ADDR	Arbitrary*	Cycle steal
		1	1	CMT		CMI (Compare-match timer interrupt)	Arbitrary*	Arbitrary*	Burst/ cycle steal

Note: * External memory, memory-mapped external device, on-chip supporting module (excluding DMAC, BSC, UBC)

In order to output a transfer request from an on-chip supporting module, set the corresponding interrupt enable bit for outputting the interrupt signal.

When the interrupt request signal from an on-chip supporting module is used as a DMA transfer request signal, an interrupt is not generated to the CPU.

The DMA transfer request signals shown in table 14.4 are automatically canceled when the corresponding DMA transfer is completed. This operation is provided at the first transfer in cycle steal mode, and at the last transfer in burst mode.

14.3.3 Channel Priority

When the DMAC receives multiple transfer requests simultaneously, it provides transfer operation according to a specified priority order. The fixed mode or round-robin mode can be selected for the channel priority with the PR1 and PR0 bit in the DMA operation register (DMAOR).

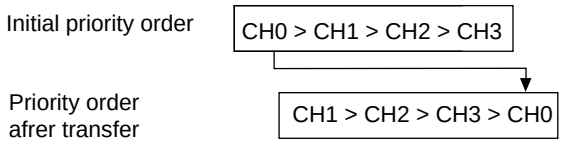
Fixed Mode: The channel priority is fixed. There are three kinds of orders as follows:

CH0 > CH1 > CH2 > CH3
CH0 > CH2 > CH3 > CH1
CH2 > CH0 > CH1 > CH3

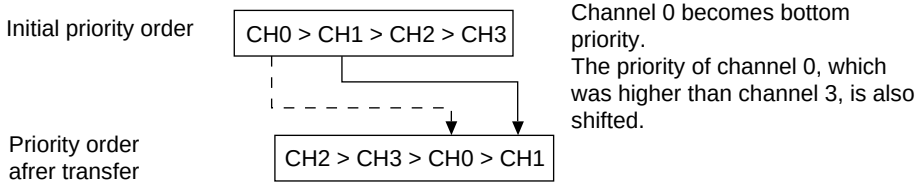
The priority is selected by the PR1 and PR0 bits in the DMA operation register (DMAOR).

Round-Robin Mode: The priority order is rotated each time one transfer unit (word, byte, or longword) of data has been transferred. The channel on which the transfer was just finished is located at the lowest in priority. The round-robin mode operation is shown in figure 14.3. The priority of the round-robin mode is CH0 > CH1 > CH2 > CH3 immediately after a reset.

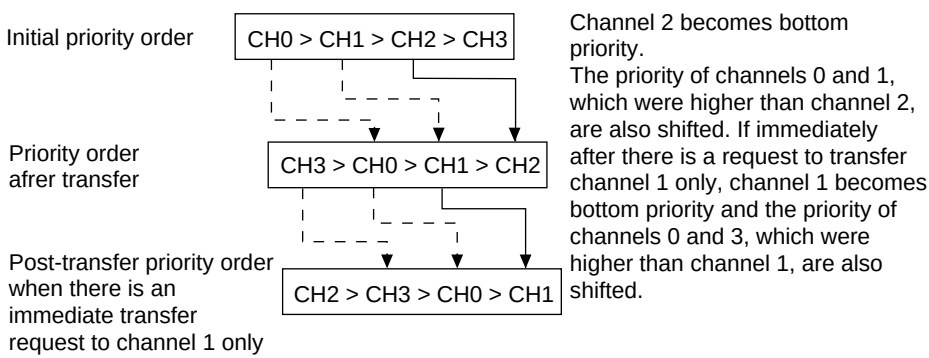
(1) When channel 0 transfers



(2) When channel 1 transfers



(3) When channel 2 transfers



(4) When channel 3 transfers

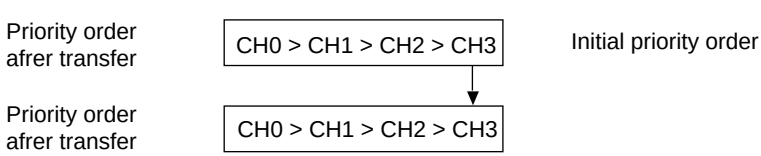


Figure 14.3 Operation in Round-Robin Mode

Figure 14.4 shows how the priority order changes when transfer requests for channel 0 and channel 3 are generated simultaneously and a transfer request for channel 1 is requested during the channel 0 transfer. The DMAC operates as follows:

1. Transfer requests are generated simultaneously for channels 0 and 3.
2. Channel 0 has the higher priority, so the channel 0 transfer begins first (channel 3 waits for transfer).
3. A channel 1 transfer request occurs during the channel 0 transfer (channels 1 and 3 are both waiting)
4. When the channel 0 transfer ends, channel 0 has the lowest priority.
5. At this time, channel 1 has the higher priority than channel 3, so the channel 1 transfer begins (channel 3 waits for transfer).
6. When the channel 1 transfer ends, channel 1 has the lowest priority.
7. The channel 3 transfer begins.
8. When the channel 3 transfer ends, channels 3 and 2 shift downward in priority order so that channel 3 has the lowest priority.

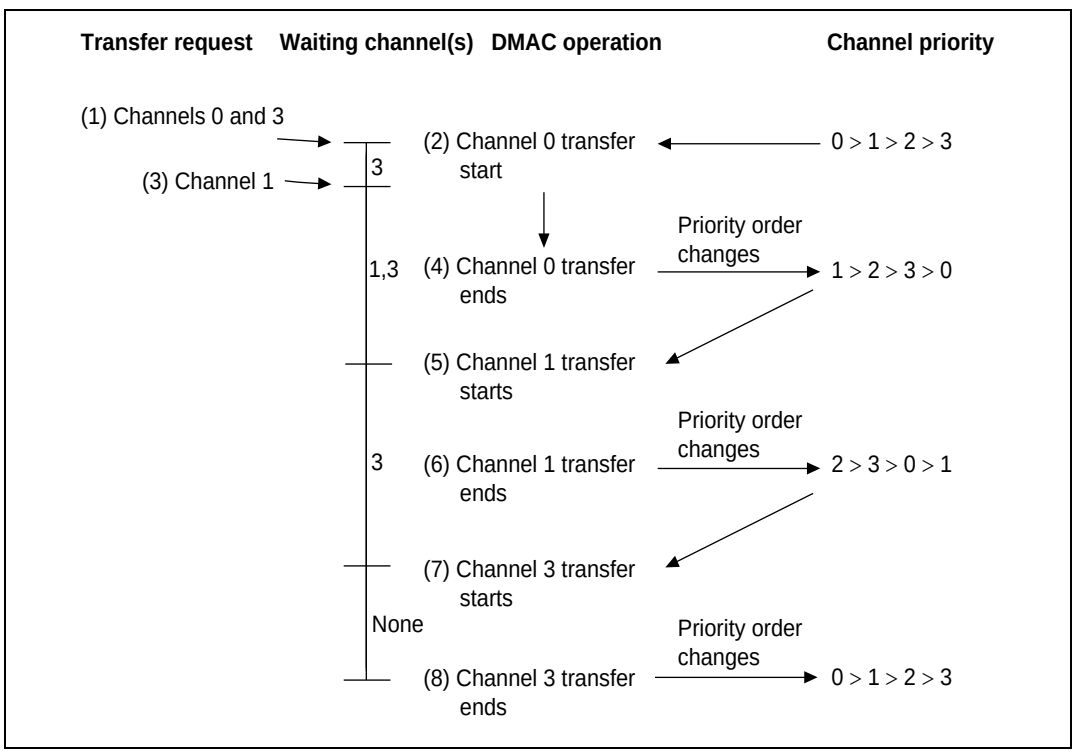


Figure 14.4 Channel Priority Order in Round-Robin Mode

14.3.4 DMA Transfer Types

The DMAC supports the transfers shown in table 14.5. The dual address mode comprises the direct address mode and indirect address mode. In the direct address mode, an output address value is the data transfer target address. In the indirect address mode, the value stored in the output address, not the output address value itself, is the data transfer target address. The data transfer timing differs depending on the bus mode. The bus mode comprises the cycle steal mode and burst mode.

Table 14.5 DMA Transfers

	Destination				
Source	**External Device with DACK**	**External Memory**	**Memory-Mapped External Device**	**On-Chip Supporting Module**	**XY Memory**
External device with DACK	Not available	Dual, single	Dual, single	Not available	Not available
External memory	Dual, single	Dual	Dual	Dual	Dual
Memory-mapped external device	Dual, single	Dual	Dual	Dual	Dual
On-chip supporting module	Not available	Dual	Dual	Dual	Dual
X/Y memory	Not available	Dual	Dual	Dual	Dual

Notes: 1. Dual: Dual address mode
2. Single: Single address mode
3. The dual address mode includes the direct address mode and the indirect address mode.
4. 16-byte transfer is not available for on-chip supporting modules.

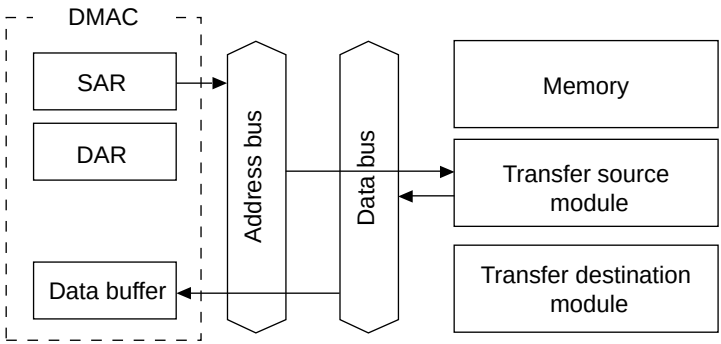
Address Mode:

- Dual Address Mode

In the dual address mode, the transfer source and destination are accessed by addresses. Both external and internal addresses can be used for the transfer source or destination. The dual address mode consists of the direct address transfer mode (1) and indirect address transfer mode (2).

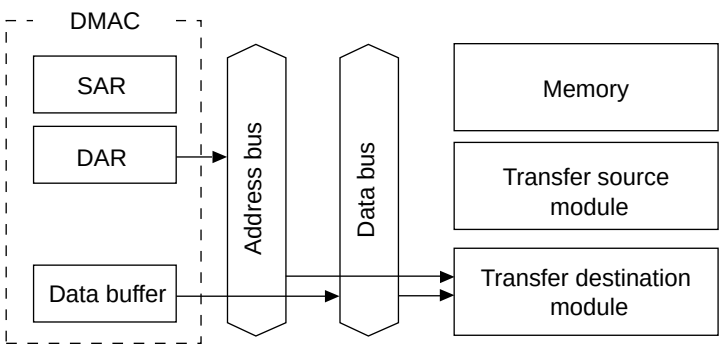
(1) Direct address transfer mode

DMA transfer requires two bus cycles because data is read from the transfer source in a data read cycle and written to the transfer destination in a data write cycle. At this time, transfer data is temporarily stored in the DMAC. In the transfer between external memories as shown in figure 14.5, data is read from one external memory to the DMAC in a data read cycle, and then that data is written to the other external memory in a write cycle. Figures 14.6 to 14.8 show examples of this operation timing



Data is read from the transfer source module using the SAR value as the address, and the read data is stored in the DMAC temporarily.

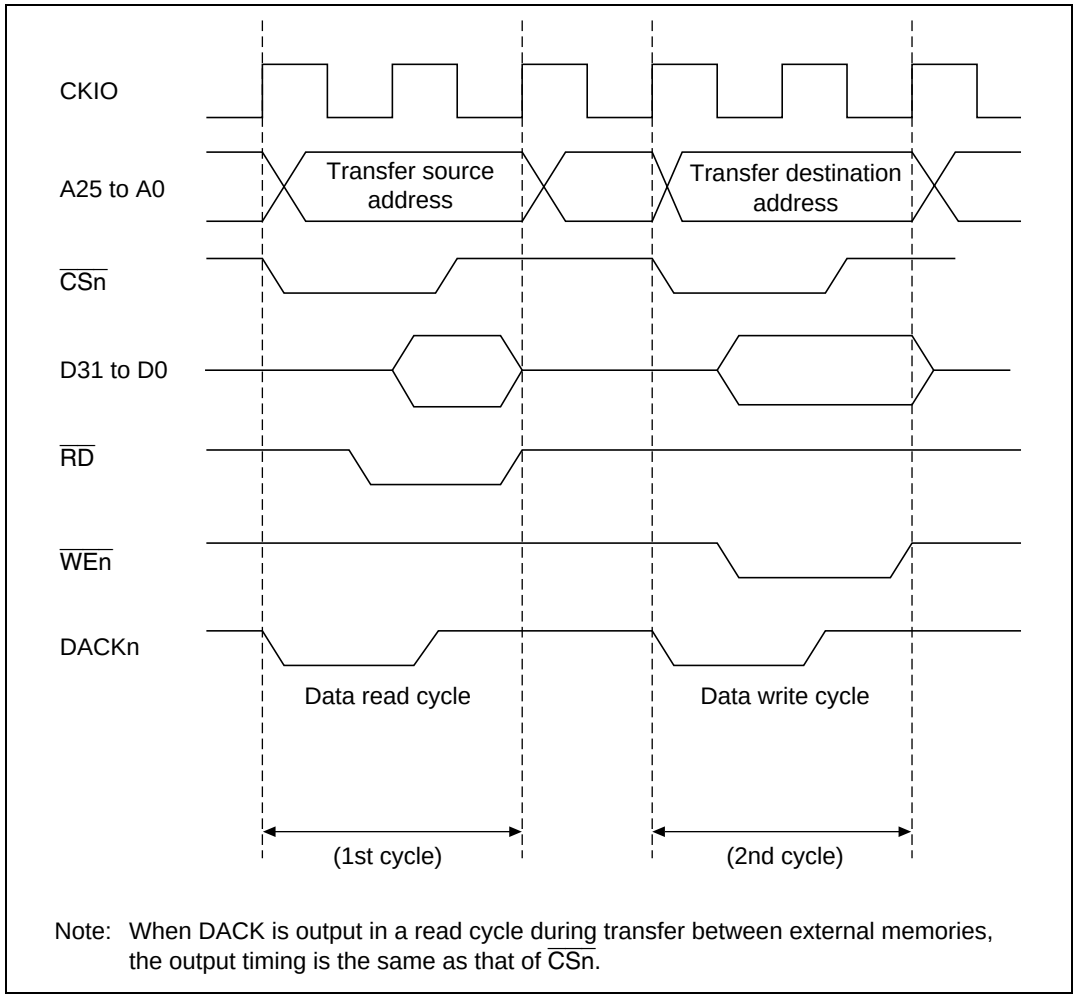
First bus cycle



The value stored in the DMAC is written to the transfer destination module using the DAR value as the address.

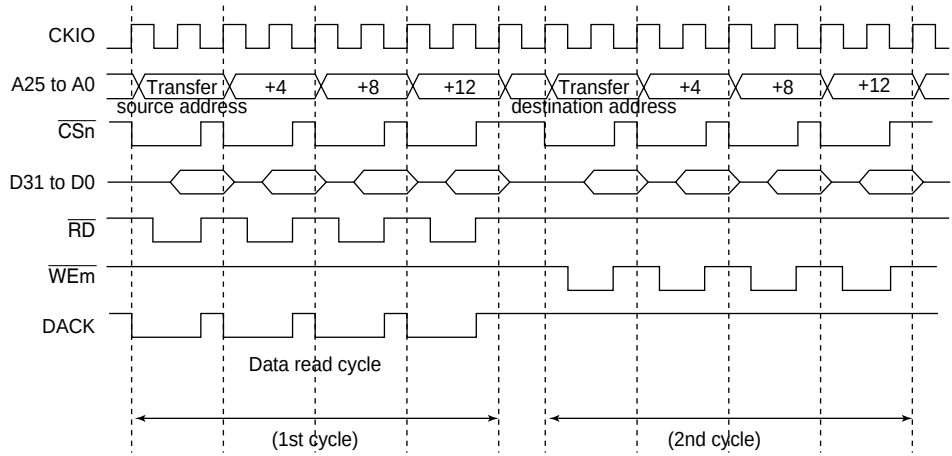
Second bus cycle

Figure 14.5 Operation in Direct Address Mode



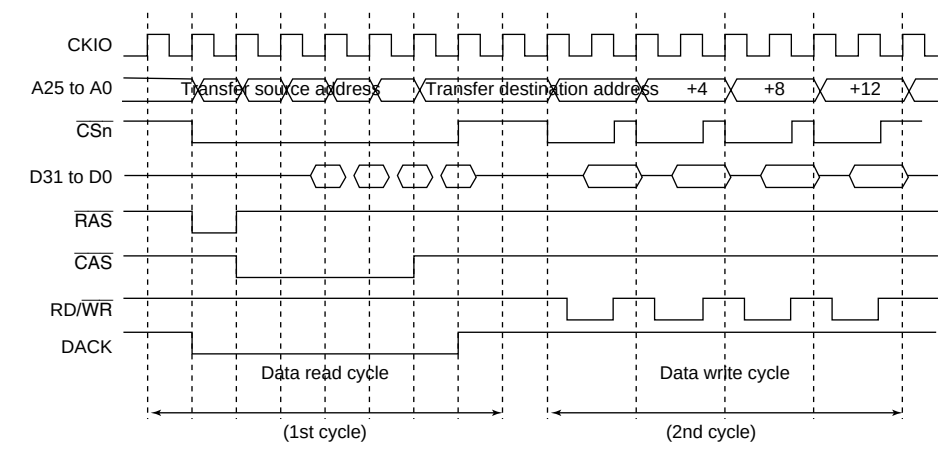
Note: When DACK is output in a read cycle during transfer between external memories, the output timing is the same as that of $\overline{CSn}$.

Figure 14.6 Example of DMA Transfer Timing in the Direct Address Mode (Transfer Source: Ordinary Memory, Transfer Destination: Ordinary Memory)



Note: When DACK is output in a read cycle during transfer between external memories, the output timing is the same as that of CSn.

Figure 14.7 Example of DMA Transfer Timing in the Direct Address Mode (16-byte Transfer, Transfer Source: Ordinary Memory, Transfer Destination: Ordinary Memory)



Note: When DACK is output in a read cycle during transfer between external memories, the output timing is the same as that of CSn.

Figure 14.8 Example of DMA Transfer Timing in the Direct Address Mode (16-byte Transfer, Transfer Source: Synchronous DRAM, Transfer Destination: Ordinary Memory)

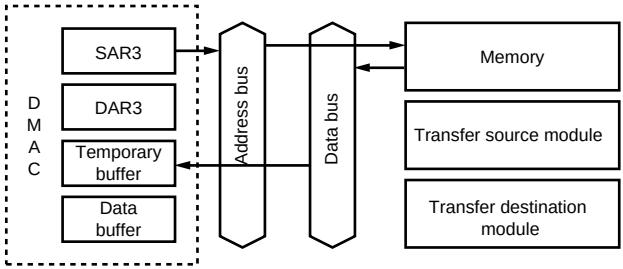
(2) In the indirect address transfer mode

The address of the memory in which data to be transferred is stored is specified in the transfer source address register (SAR3) in the DMAC. 16-byte transfer is not provided in this mode. The address value specified in the transfer source address register in the DMAC is read first, and this value is temporarily stored in the DMAC. Next, the read value is output as an address, and data on that address is stored in in the DMAC again. Then, the value read afterwards is written to the address specified by the transfer destination address register; thus one DMA transfer is completed.

Figure 14.9 shows an example of this operation. In this example, the transfer destination, transfer source, and storage destination of the indirect address are all in external memories, and the transfer data size is 16 or 8 bits. Figure 14.10 shows an example of the transfer timing.

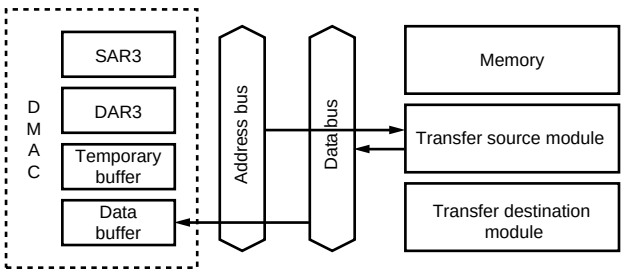
In this mode, one NOP cycle (CK1 cycle shown in figure 14.10) is required to output data which was read as an indirect address to an address bus.

For a 32-bit data transfer, third and fourth bus cycles shown in figure 14.10 are required twice for each; a total of six bus cycles and one NOP cycle are required.



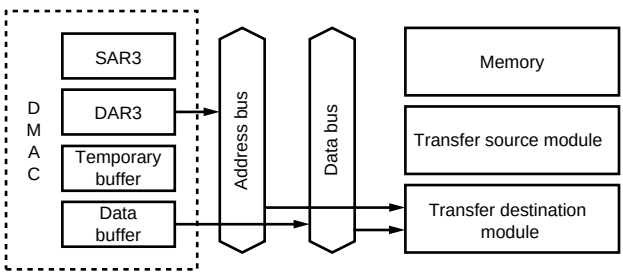
Data is read from memory using the SAR3 value as the data address, and the read data is stored in the temporary buffer. The read data must be a 32-bit value because it is used as an address.
Two bus cycles are required if a 16-bit data bus is used to connect to an external device.

First and second bus cycles



Data is read from the source module using the temporary buffer value as the address, and the read data is transferred to the data buffer.

Third bus cycle

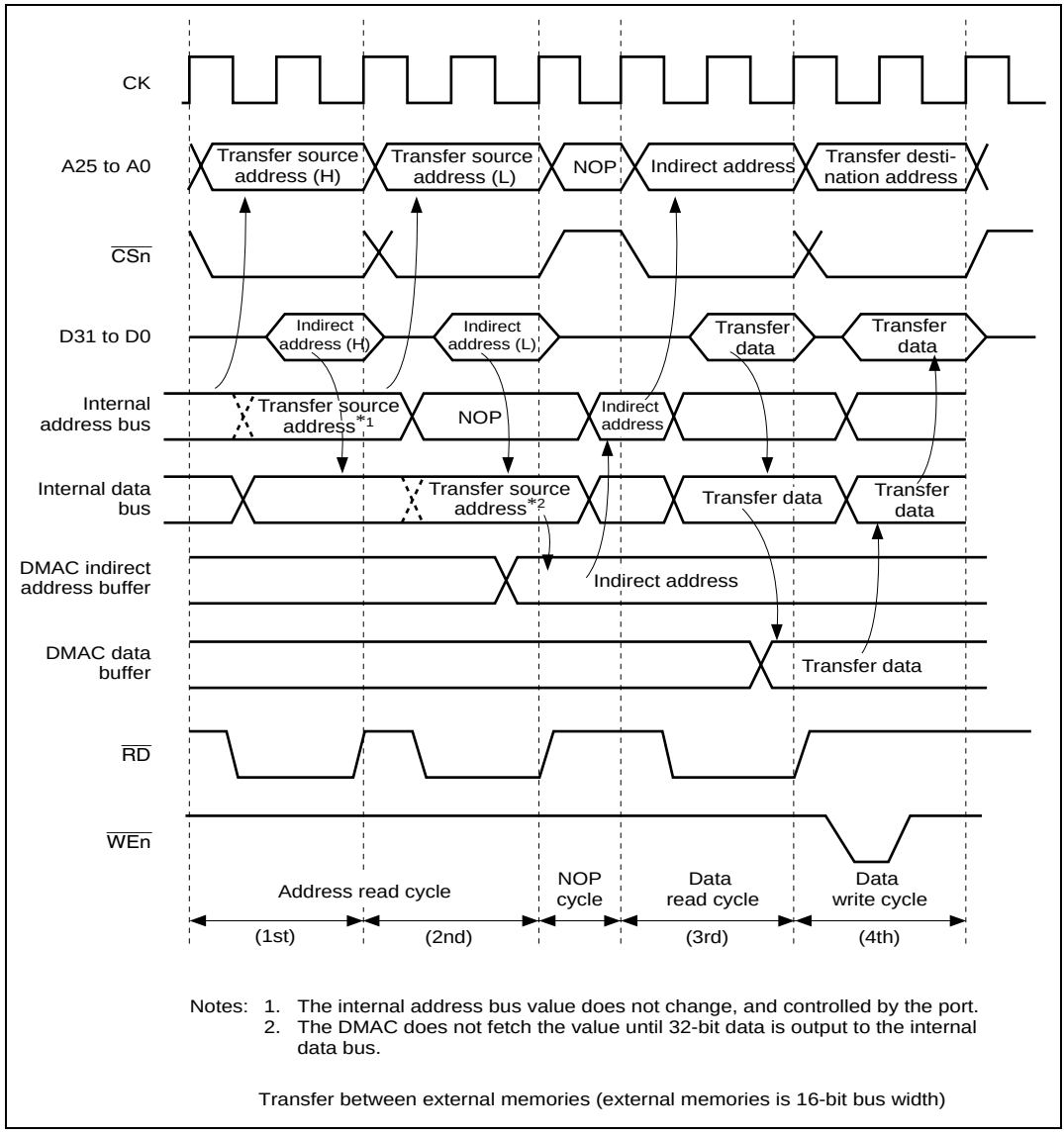


The data buffer value is written to the destination module using the DAR3 value as the destination address.

Fourth bus cycle

Note: The above description uses the memory, transfer source module, or transfer destination module; in practice, any module can be connected in the addressing space.

Figure 14.9 Operation in Indirect Address Mode (When the External Memory Space is Set to 16-bit Width)



Notes: 1. The internal address bus value does not change, and controlled by the port.
2. The DMAC does not fetch the value until 32-bit data is output to the internal data bus.

Transfer between external memories (external memories is 16-bit bus width)

Figure 14.10 Example of Transfer Timing in Indirect Address Mode (Transfer between External Memories, External Memory with 16-bit Width)

• Single Address Mode

The single address mode is used when transfer is performed between external devices including external memories, one of which is accessed (selected) by the DACK signal and the other of which is accessed by address. In this mode, the DMAC outputs the transfer request acknowledge signal DACK to one external device, and simultaneously outputs an address to the other device; thus DMA transfer is performed in one bus cycle. An example of transfer between an external memory and an external device with DACK is shown in figure 14.11. The external device outputs data to a data bus and the data is written to the external memory in a single bus cycle.

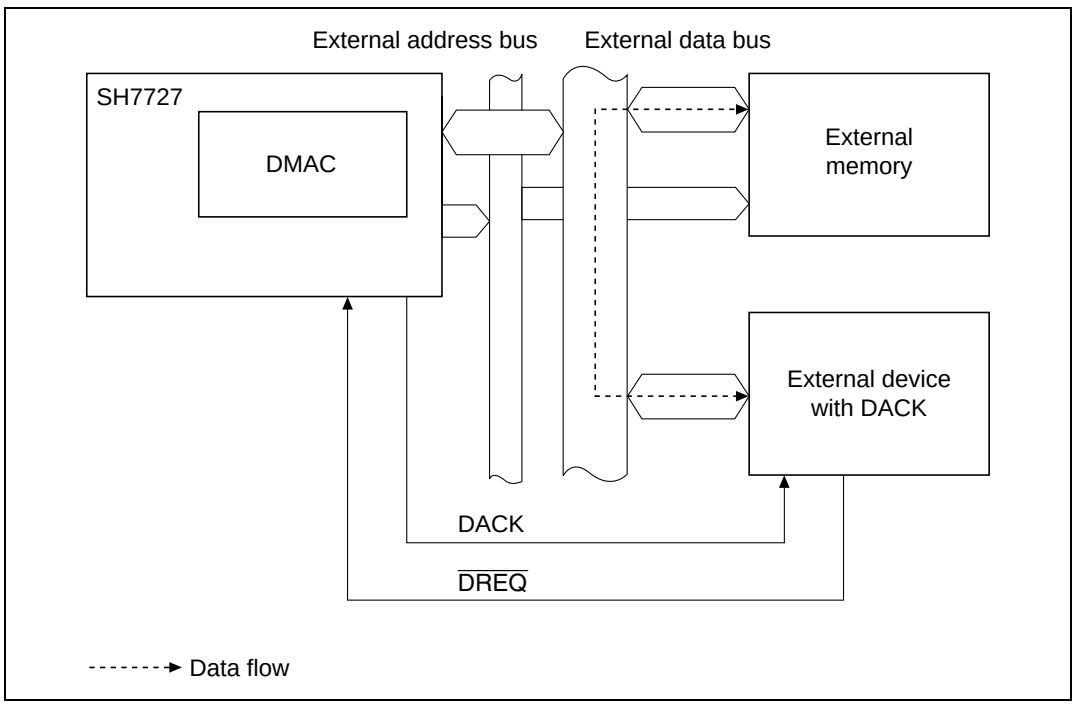


Figure 14.11 Data Flow in Single Address Mode

Two kinds of transfer are possible in single address mode: (1) transfer between an external device with DACK and a memory-mapped external device, and (2) transfer between an external device with DACK and an external memory. In both cases, only the external request signal ($\overline{\text{DREQ}}$) is used as a transfer request.

Figures 14.12 and 14.13 show examples of the DMA transfer timing in single address mode.

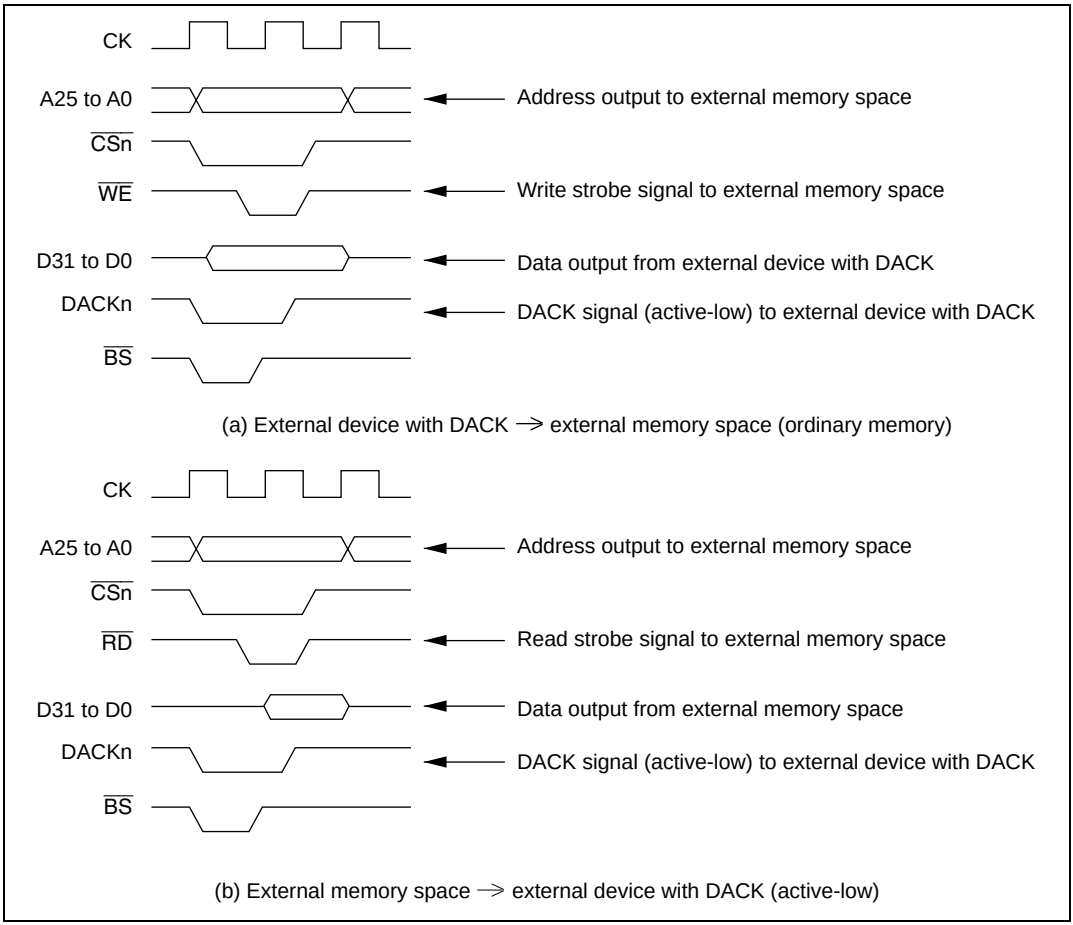


Figure 14.12 Example of DMA Transfer Timing in Single Address Mode

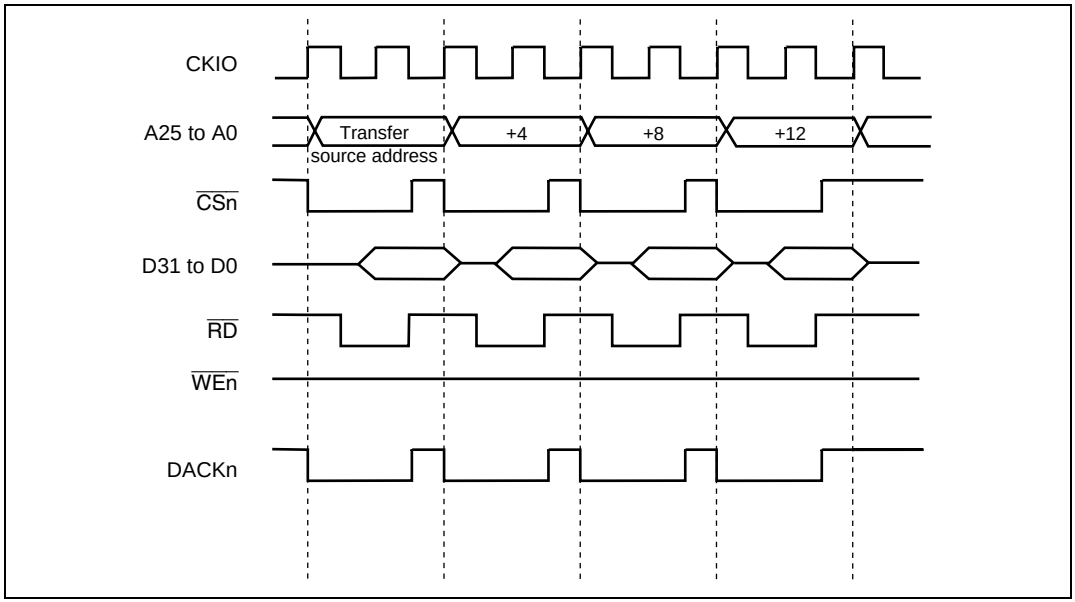


Figure 14.13 Example of DMA Transfer Timing in Single Address Mode (External Memory Space (Ordinary Memory) → External Device with DACK)

Bus Modes: There are two types of bus modes, cycle steal mode and burst mode. Select the mode in the TM bits in CHCR0 to CHCR3.

• Cycle-Steal Mode

In the cycle-steal mode, the bus right is moved to another bus master after one transfer unit (byte, word, longword, or 16-byte unit) of DMA transfer. If another transfer request occurs after the bus right moving, the bus right are re-moved to the DMAC. Then, the DMAC performs transfer for one transfer unit and releases the bus right again. This operation is repeated until the transfer end condition is satisfied.

In the cycle-steal mode, transfer areas are not affected by settings of the transfer request source, transfer source, and transfer destination. Figure 14.14 shows an example of the DMA transfer timing in the cycle steal mode. In this example, the following conditions are set:

- Dual address mode
- $\overline{\text{DREQ}}$ level detection

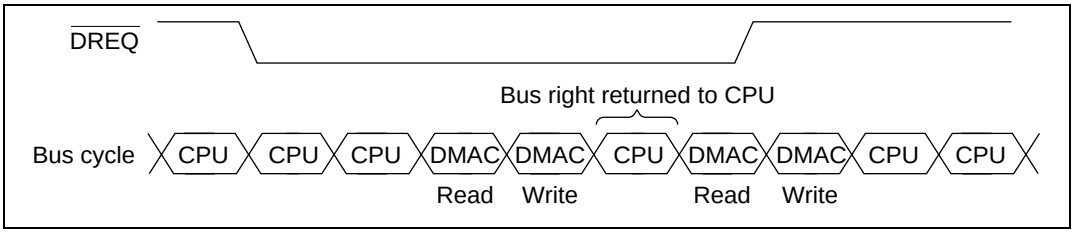


Figure 14.14 Transfer Example in Cycle-Steal Mode

- Burst Mode

Once the DMAC obtains the bus right, the transfer is continued until the transfer end condition is satisfied. However, when the $\overline{\text{DREQ}}$ pin is driven high in the external request mode with low level detection of the $\overline{\text{DREQ}}$ pin, the bus right is passed to the other bus master after the DMA transfer request that has already been accepted ends, even if the transfer end condition has not been satisfied.

The burst mode cannot be used when the transfer request source is set to the serial communications interface with FIFO (SCIF). Figure 14.15 shows a timing of the DMA transfer operation in the burst mode.

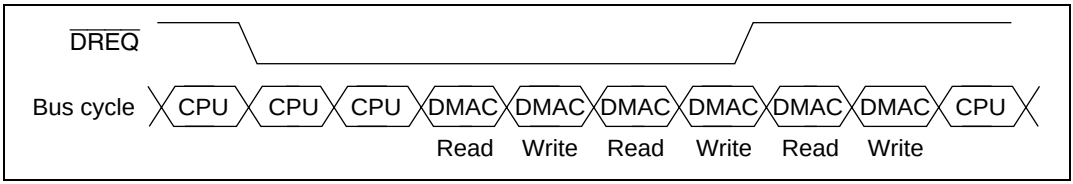


Figure 14.15 Example of Transfer in Burst Mode

Relationship between Request Mode and Bus Mode: Table 14.6 shows the relationship between request mode and bus mode for each combination of DMA transfer areas.

Table 14.6 Relationship of Request Modes and Bus Modes

Address Mode	Transfer Areas	Request Mode	Bus Mode	Transfer Size (bits)	Usable Channels
Dual	External device with DACK and external memory	External	B/C	8/16/32/128	0
	External device with DACK and memory-mapped external device	External	B/C	8/16/32/128	0
	External memory and external memory	All*1	B/C	8/16/32/128	0–3*5
	External memory and memory-mapped external device	All*1	B/C	8/16/32/128	0–3*5
	Memory-mapped external device and memory-mapped external device	All*1	B/C	8/16/32/128	0–3*5
	External memory and on-chip supporting module	All*2	B/C*3	8/16/32*4	0–3*5
	Memory-mapped external device and on-chip supporting module	All*2	B/C*3	8/16/32*4	0–3*5
	On-chip supporting module and on-chip supporting module	All*2	B/C*3	8/16/32*4	0–3*5
	X/Y memory and X/Y memory	All	B/C	8/16/32/128	0–3
	X/Y memory and memory-mapped external device	All*1	B/C	8/16/32/128	0–3
	X/Y memory and on-chip supporting module	All*2	B/C*3	8/16/32	0–3
	X/Y memory and external memory	All	B/C	8/16/32/128	0–3
Single	External device with DACK and external memory	External	B/C	8/16/32/128	0
	External device with DACK and memory-mapped external device	External	B/C	8/16/32/128	0

B: Burst

C: Cycle steal

Notes: 1. External requests, auto requests and on-chip supporting module (CMT) requests are all available.

2. External requests, auto requests and on-chip supporting module requests are all available. When the SIOF, USBF, SCIF, or A/D converter is the transfer request source, the transfer destination or transfer source must be also the SIOF, USBF, SCIF, or A/D converter, respectively.

3. The SIOF, USBF, SCIF, or A/D converter can be specified for the transfer request source in the cycle-steal mode only.

4. The access size permitted when the transfer destination or source is an on-chip supporting module register.

5. If the transfer request is an external request, only channel 0 is available.

Bus Mode and Channel Priority Order: For example, when channel 1 provides transfer operation in burst mode and then a transfer request to channel 0 with the higher priority is generated, the transfer of channel 0 will begin immediately.

At this time, if the priority is set in the fixed mode (CH0 > CH1), the channel 1 transfer will be continued after the channel 0 transfer has completely finished, even if channel 0 is set to the cycle steal mode or burst mode.

If the round-robin mode is selected, channel 1 will begin operating again after channel 0 completes the transfer of one transfer unit, even if channel 0 is set to the cycle steal mode or burst mode. The bus is moved between the two in the order channel 1, channel 0, channel 1, channel 0.

Even if the fixed mode or in the round-robin mode is selected, the bus is not passed to the CPU since channel 1 is in the burst mode. Figure 14.16 shows an example of operation in the round-robin mode.

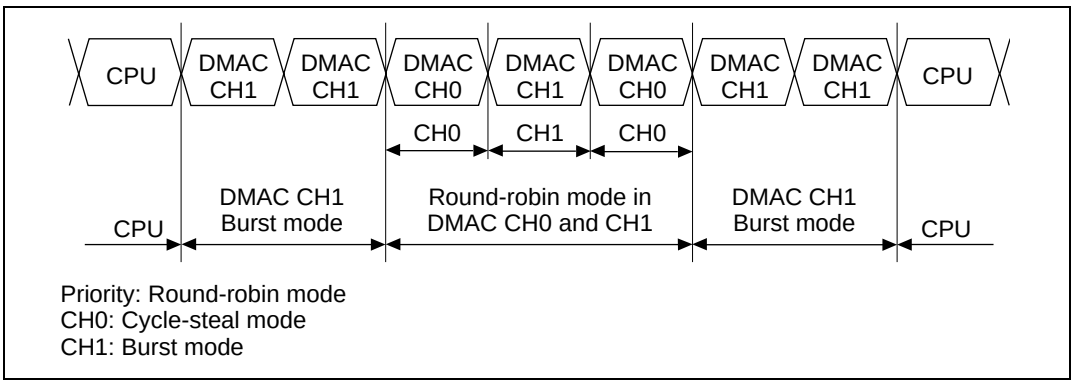


Figure 14.16 Bus State in Multiple Channel Operation

14.3.5 Number of Bus Cycle States and $\overline{\text{DREQ}}$ Pin Sampling Timing

Number of Bus Cycle States: When the DMAC is the bus master, the number of bus cycle states is controlled by the bus state controller (BSC) in the same way as when the CPU is the bus master. For details, see section 12, Bus State Controller (BSC).

$\overline{\text{DREQ}}$ Pin Sampling Timing: In external request mode, the $\overline{\text{DREQ}}$ pin is sampled with the clock pulse (CKIO) falling edge detection or low level detection. When a $\overline{\text{DREQ}}$ input is detected, a DMAC bus cycle is generated and DMA transfer starts three or more states later.

The second and subsequent $\overline{\text{DREQ}}$ sampling operations are started two cycles after the first sample.

Operation

- Cycle-Steal Mode

  In cycle-steal mode, the $\overline{\text{DREQ}}$ sampling timing does not change according to the $\overline{\text{DREQ}}$ detection method, the level detection or edge detection.

  For example, as shown in figure 14.17 (cycle-steal mode, level detection), DMA transfer begins, at the earliest, three cycles after the first sampling is performed. The second sampling is started two cycles after the first. If $\overline{\text{DREQ}}$ is not detected at this time, sampling is performed in each subsequent cycle.

  Thus, $\overline{\text{DREQ}}$ sampling is performed one step in advance. The third sampling operation is not performed until the idle cycle following the end of the first DMA transfer.

  The above operation is performed continuously for the desired CPU transfer cycles or DMA transfer cycles, as shown in figures 14.18 and 14.19.

  Figures 14.17 and 14.18 show examples in which $\overline{\text{DACK}}$ is output in a read and in a write, respectively. In both cases, $\overline{\text{DACK}}$ is output for the same period as $\overline{\text{CSn}}$.

  Figure 14.20 shows an example in which sampling is executed in all subsequent cycles when $\overline{\text{DREQ}}$ cannot be detected. Figure 14.21 shows an example of operation in cycle steal mode with the edge detection.

- Burst Mode, Level Detection

  In the case of burst mode with level detection, the $\overline{\text{DREQ}}$ sampling timing is the same as in the cycle-steal mode.

  For example, as shown in figure 14.22, DMAC transfer begins, at the earliest, three cycles after the first sampling is performed. The second sampling is started two cycles after the first. Subsequent sampling operations are performed in the idle cycle following the end of the DMA transfer cycle.

  In the burst mode, also, the $\overline{\text{DACK}}$ output period is the same as that in the cycle-steal mode.

- Burst Mode, Edge Detection

  In the case of burst mode with edge detection, $\overline{\text{DREQ}}$ sampling is performed only once.

  For example, as shown in figure 14.23, DMAC transfer begins, at the earliest, three cycles after the first sampling is performed. After this, DMAC transfer is executed continuously until the number of data transfers set in the DMATCR register have been completed. $\overline{\text{DREQ}}$ is not sampled during this operation.

  To restart DMA transfer after it has been suspended by an NMI, first clear NMIF, then input an edge request again.

  In the burst mode, also, the $\overline{\text{DACK}}$ output period is the same as that in the cycle-steal mode.

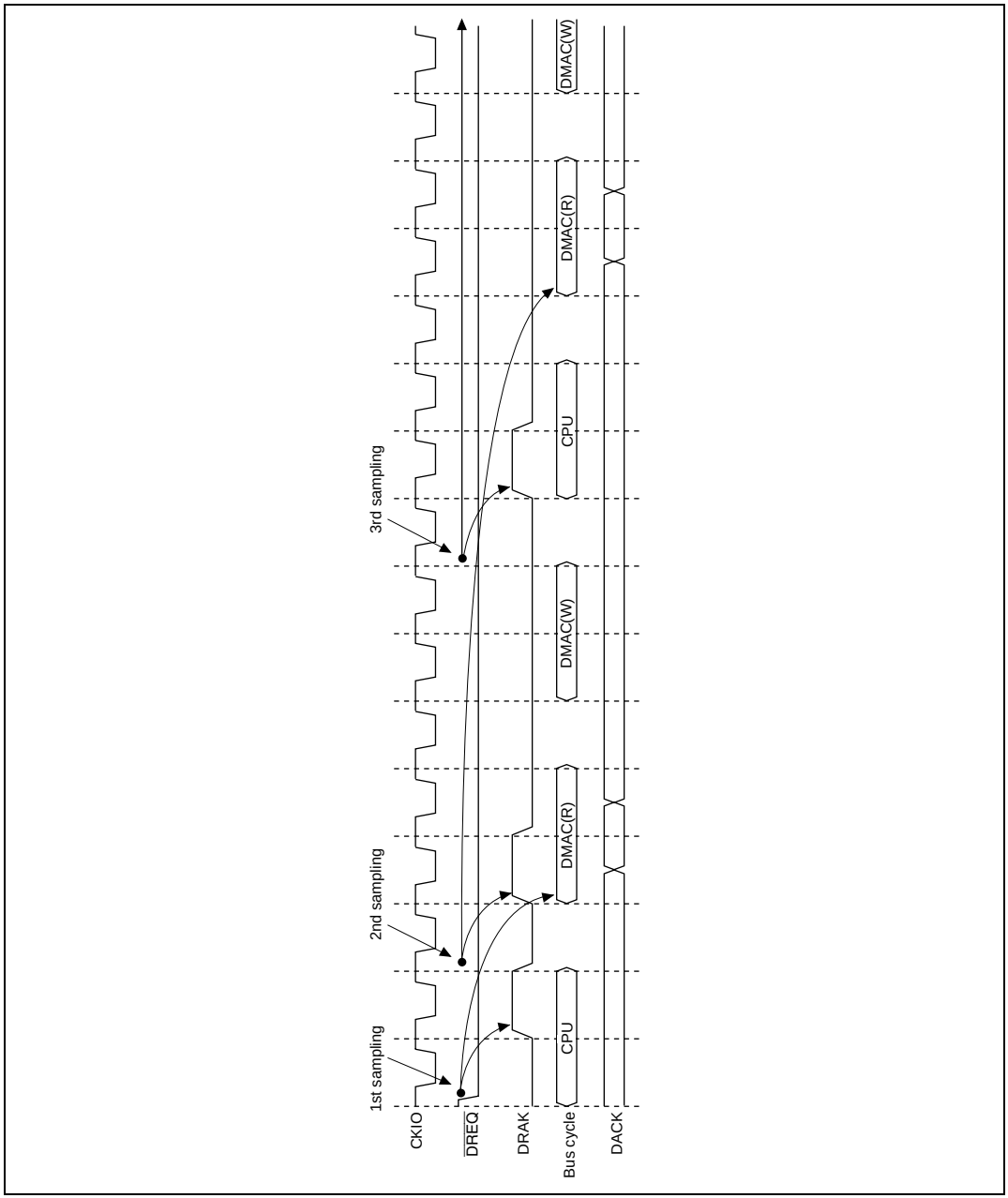


Figure 14.17 Cycle-Steal Mode, Level Input (CPU Access: 2 Cycles)

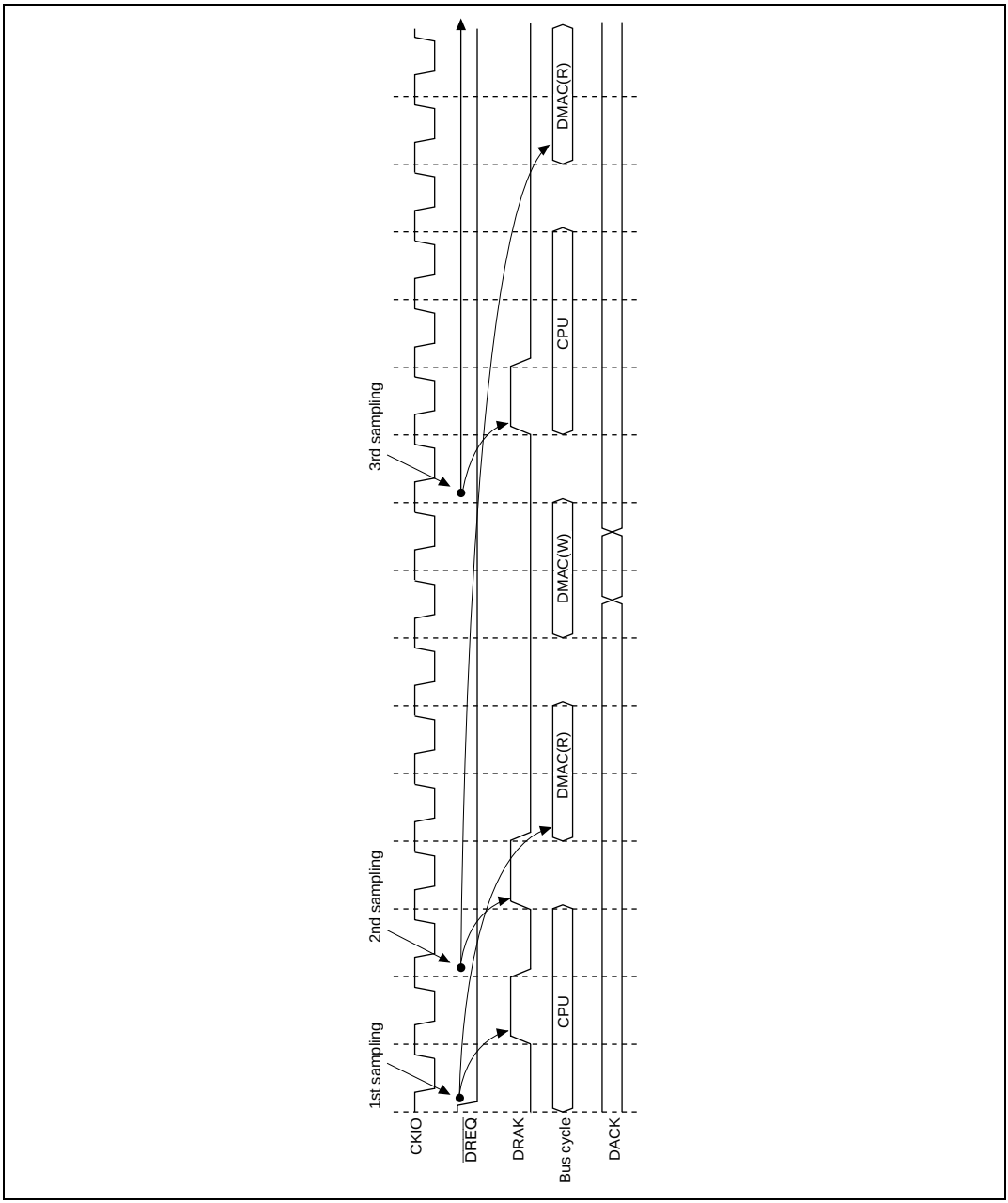


Figure 14.18 Cycle-Steal Mode, Level Input (CPU Access: 3 Cycles)

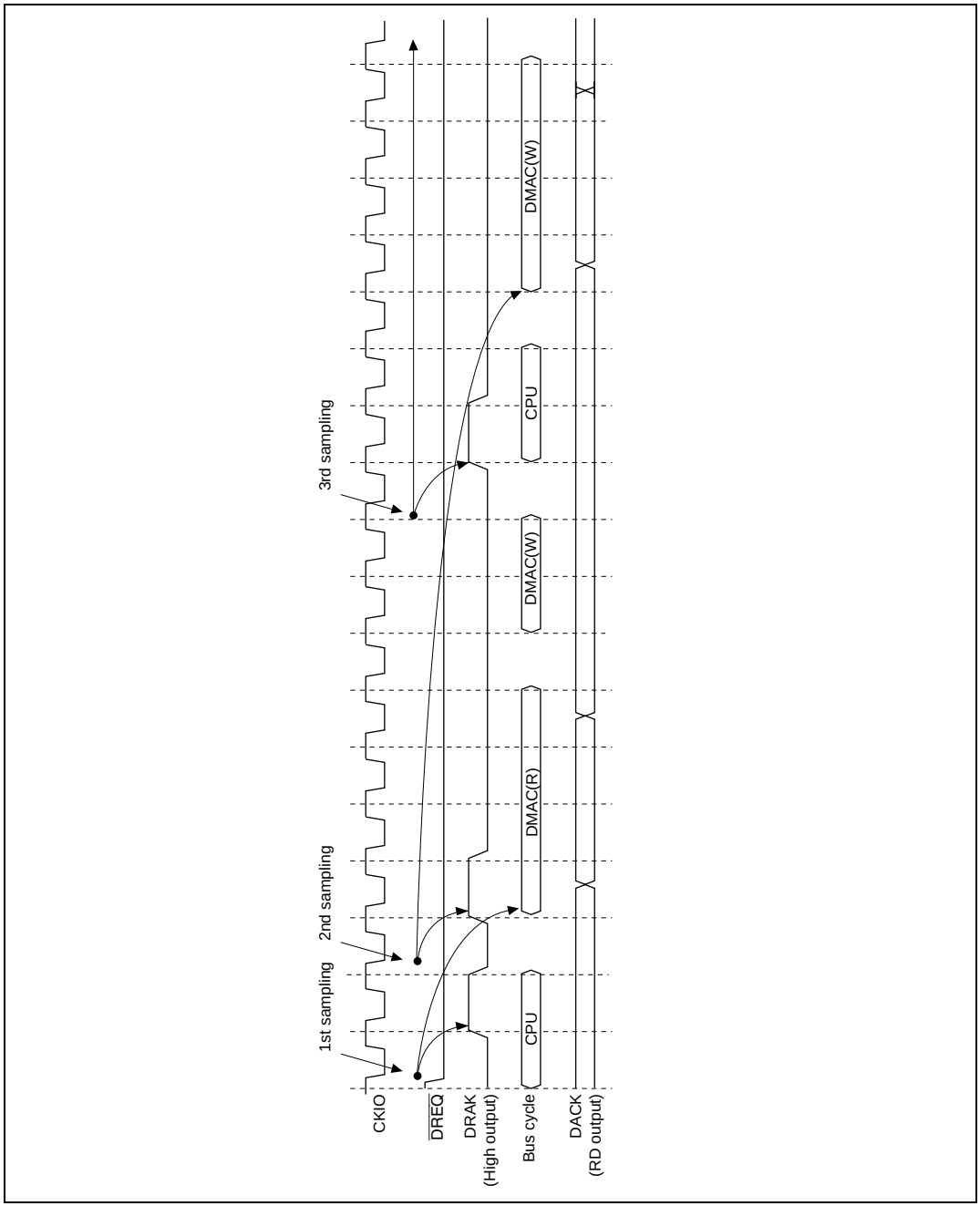


Figure 14.19 Cycle-Steal Mode, Level input (CPU Access: 2 Cycles, DMA RD Access: 4 Cycles)

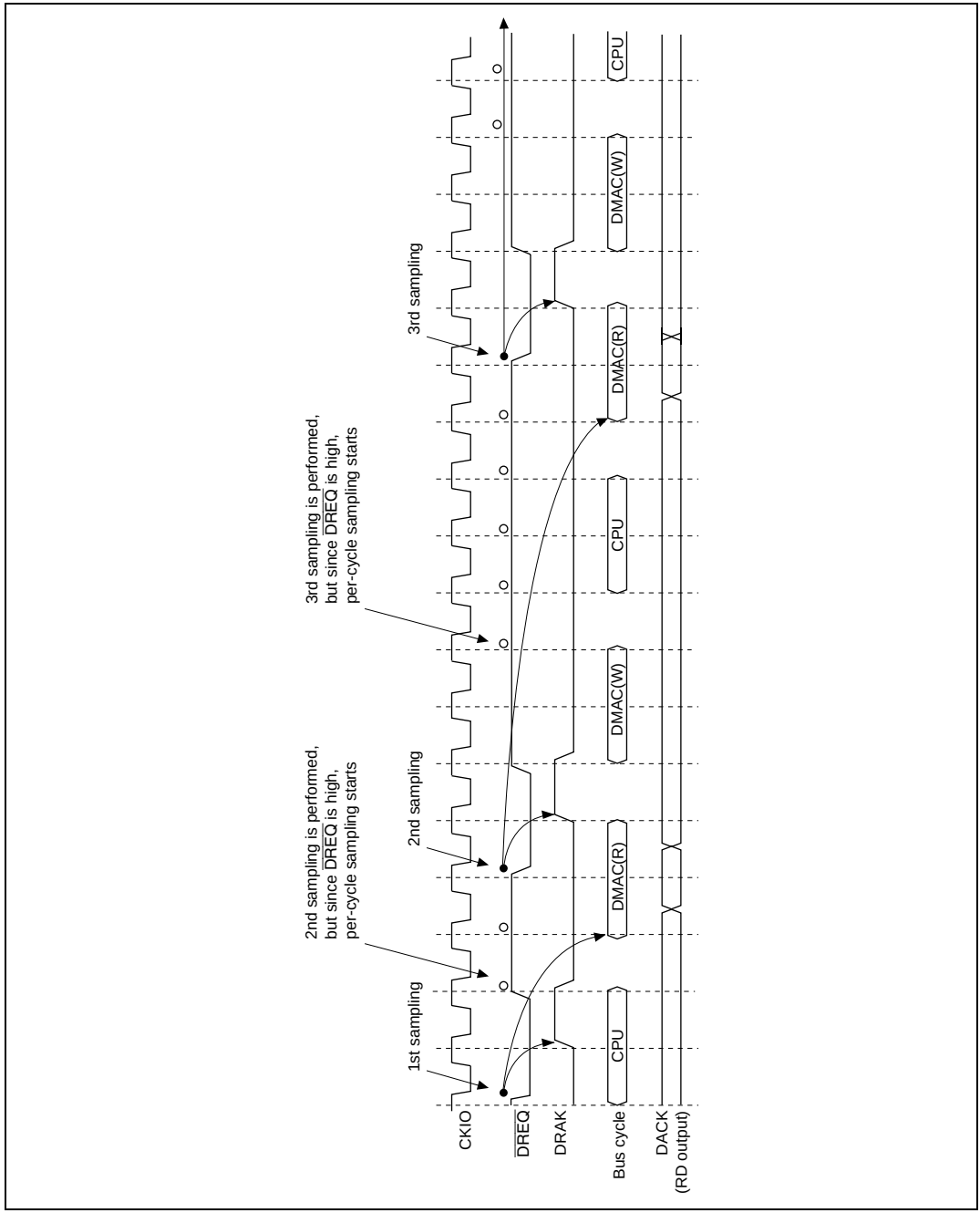


Figure 14.20 Cycle-Steal Mode, Level input (CPU Access: 2 Cycles, $\overline{\text{DREQ}}$ Input Delayed)

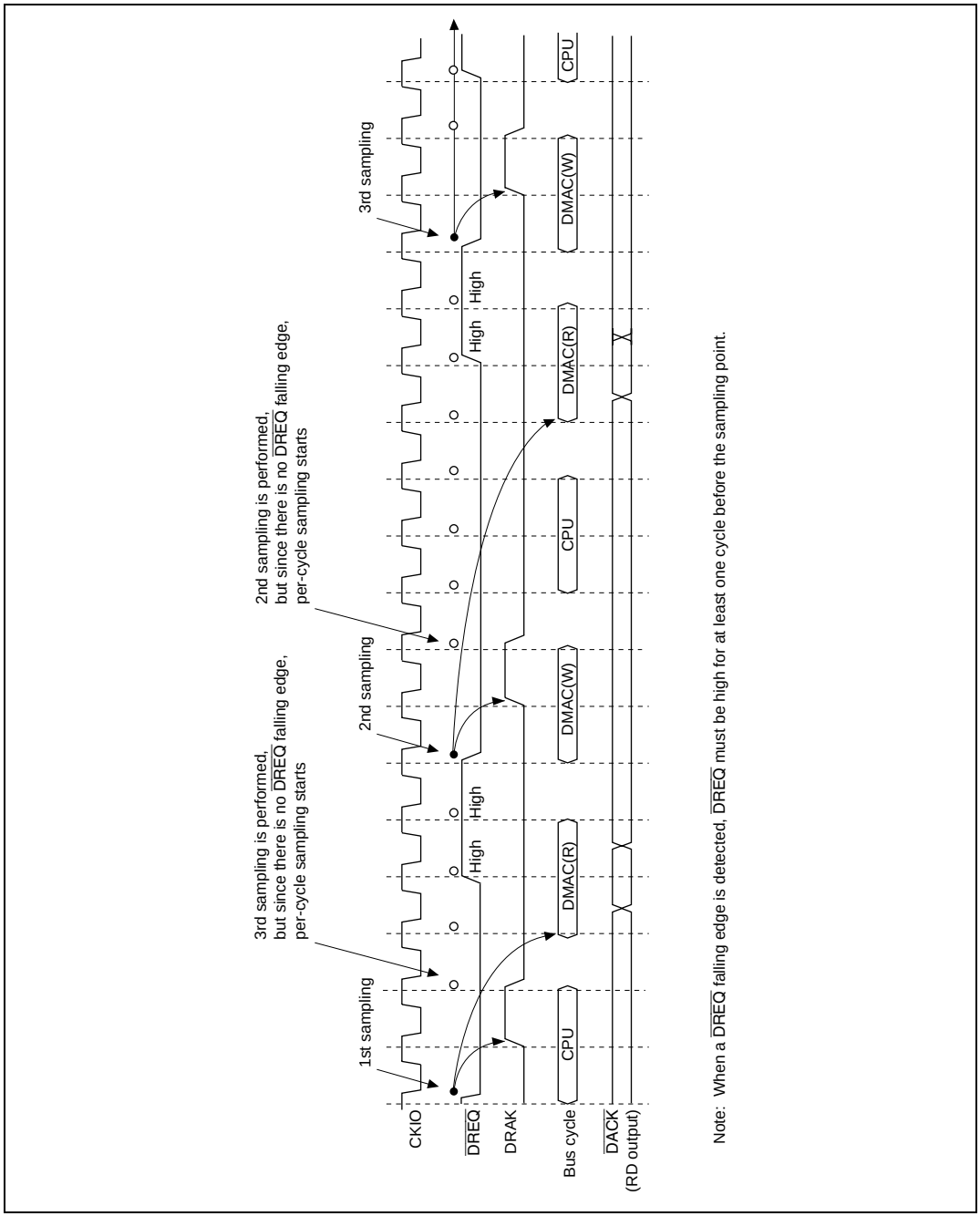


Figure 14.21 Cycle-Steal Mode, Edge input (CPU Access: 2 Cycles)

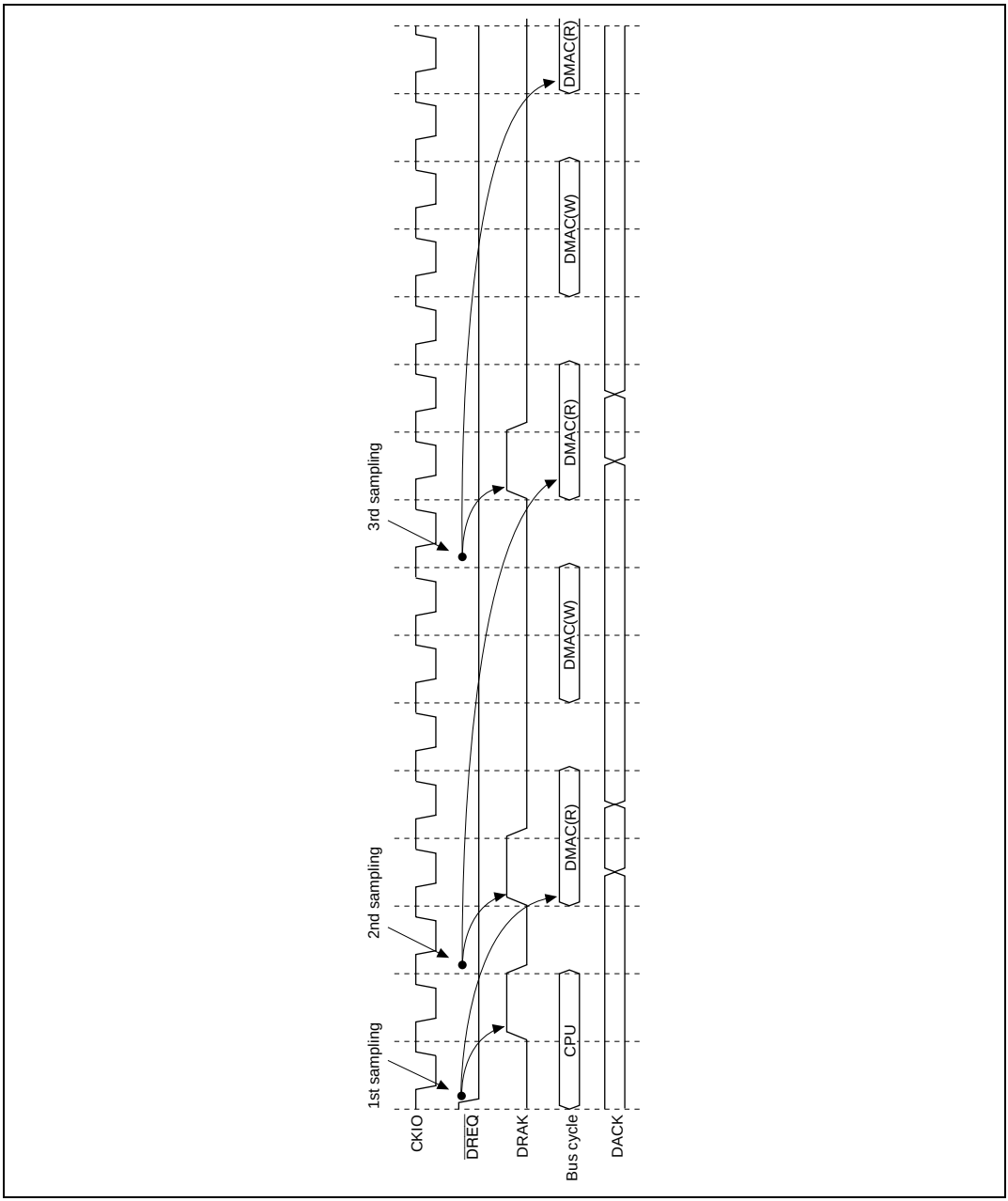


Figure 14.22 Burst Mode, Level Input

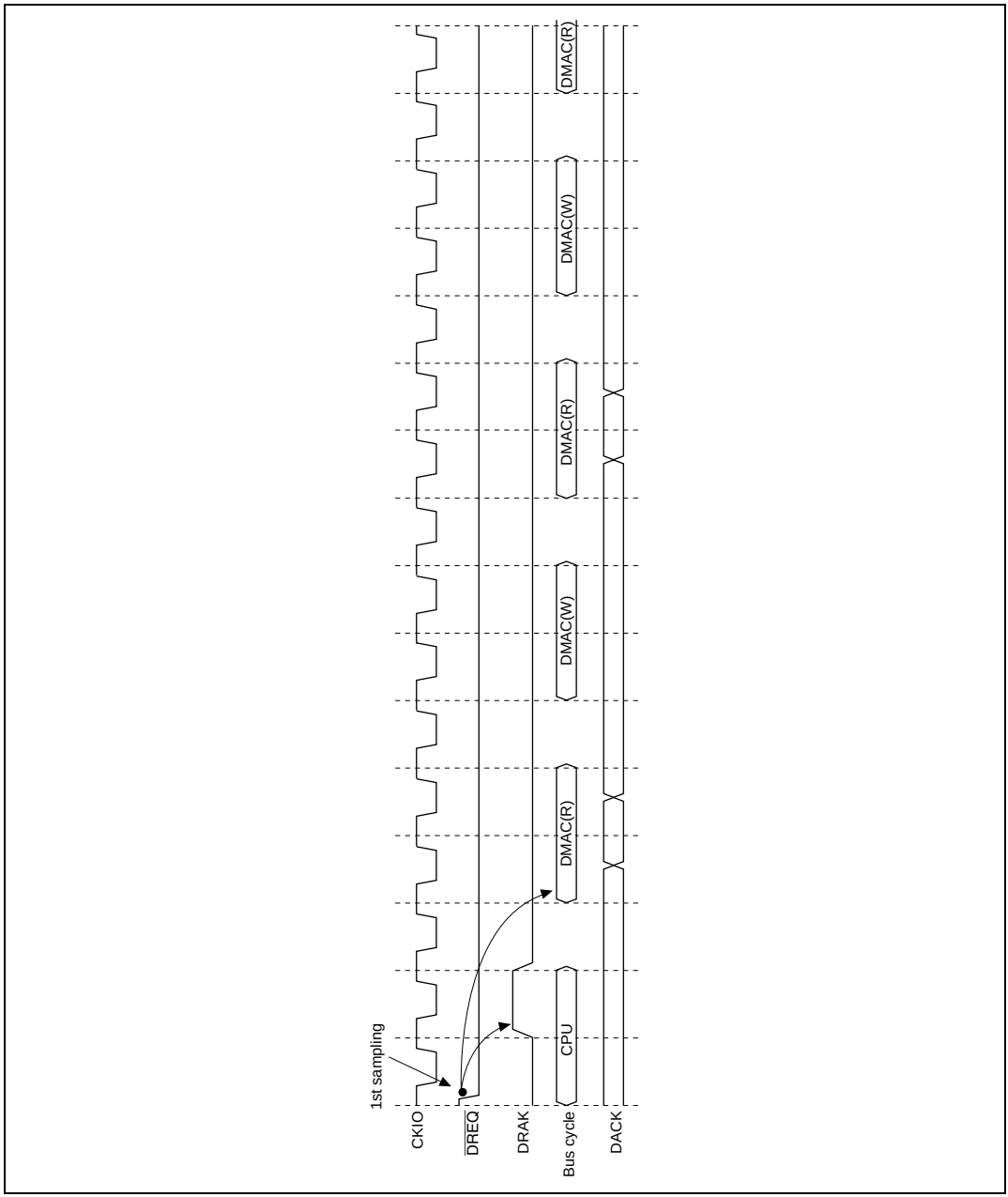


Figure 14.23 Burst Mode, Edge Input

14.3.6 Source Address Reload Function

Channel 2 includes a reload function, in which the value set in the source address register (SAR2) is restored every four transfers when the RO bit in CHCR2 is set to 1. This function cannot be used with the 16-byte transfer. Figure 14.24 shows this operation. Figure 14.25 shows a timing chart of the source address reload function with the following conditions: burst mode, auto request, 16-bit transfer data size, SAR2 count-up, DAR2 fixed, reload function on, and usage of only channel 2.

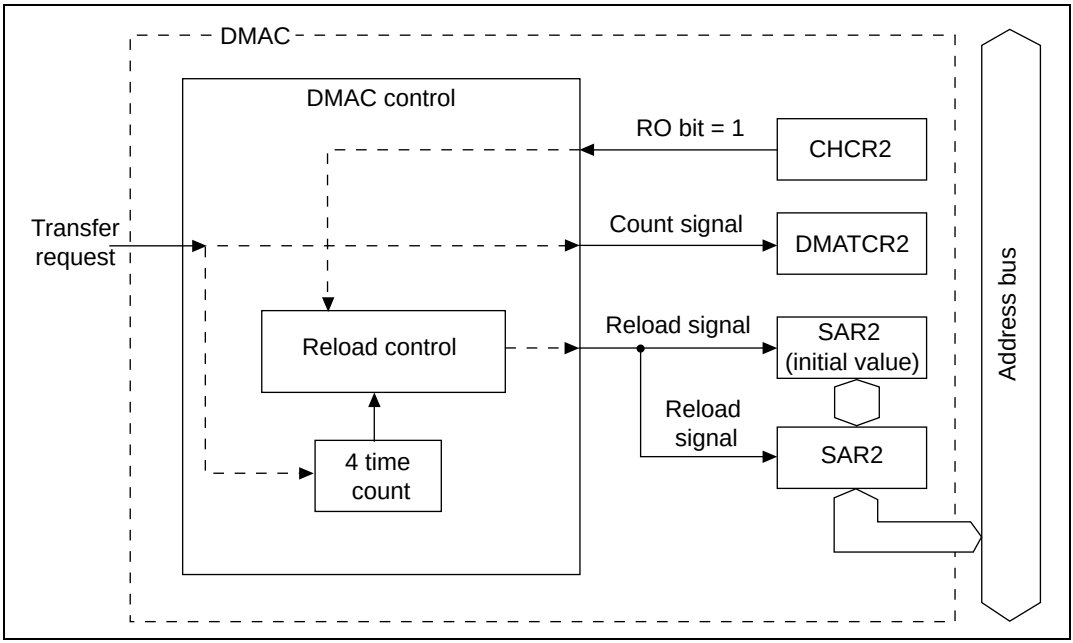


Figure 14.24 Source Address Reload Function Diagram

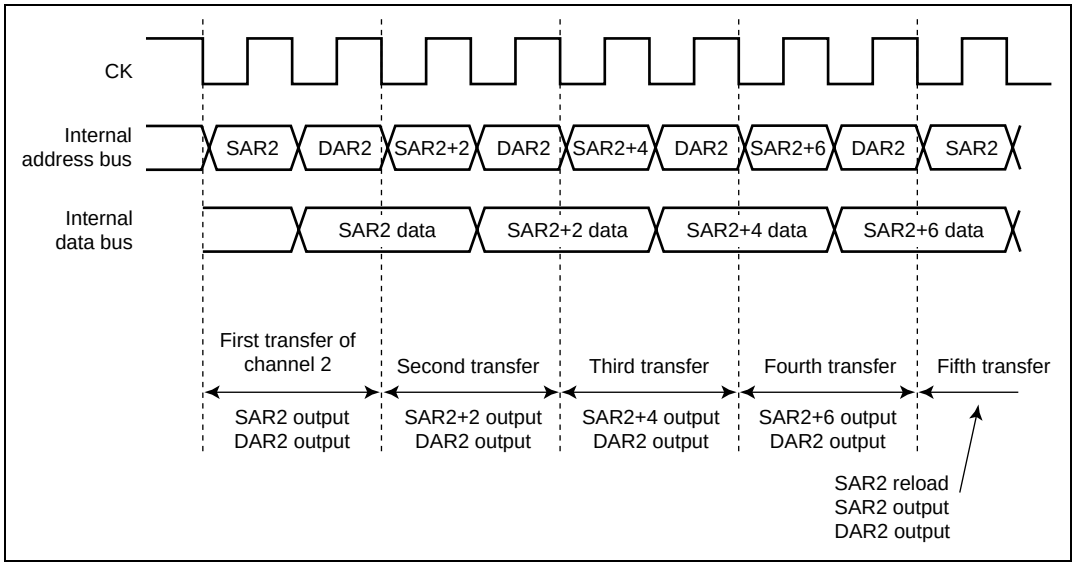


Figure 14.25 Timing Chart of Source Address Reload Function

The reload function can be used for the 8-, 16- and 32-bit data transfer.

DMATCR2, which specifies a transfer count, is incremented by 1 each time a transfer ends regardless of the reload function setting. Consequently, be sure to specify the value multiple of four in DMATCR2 when the reload function is on. If other values are specified, correct operation is not guaranteed.

The counters that count transfers of four times for the reload function are reset by clearing the DME bit in DMAOR or the DE bit in CHCR2, by setting the transfer end flag (TE bit in CHCR2), by inputting an NMI, besides by a reset or in standby mode. However, the SAR2, DAR2, DMATCR2 registers are not reset. Therefore, the above reset source is generated, some counters are initialized but some are not in the DMAC, which may cause a malfunction when the DMAC is restarted. To avoid this problem, if a reset source except the TE bit setting is generated when the reload function is used, set SAR2, DAR2, and DMATCR2 again.

14.3.7 DMA Transfer Ending

DMA transfer ending conditions to terminate transfer differ according to the ending types, individual channel ending and all channel ending. At a transfer end, the following conditions are applied except the case when the DMA transfer count register (DMATCR) value reaches 0.

(a) Cycle-steal mode (external request, internal request, and auto request)

When a transfer ending condition is satisfied, DMAC transfer request acceptance is suspended. The DMAC stops operation after completing the number of transfers that has accepted before the ending conditions are satisfied.

In the cycle-steal mode, the same operation is provided regardless of the transfer request detection method; the level detection or the edge detection.

(b) Burst mode, edge detection (external request, internal request, and auto request)

The timing of DMAC operation ending after an ending condition is satisfied differs from that in cycle steal mode. In the edge detection in the burst mode, though only one transfer request is generated at the DMAC start-up, a stop request sampling is performed in the same timing as a transfer request sampling in the cycle-steal mode. As a result, the period when a stop request is not sampled is regarded as the period when a transfer request is generated, and after performing the DMA transfer for this period, the DMAC stops operation.

(c) Burst mode, level detection (external request)

Same as described in (a).

(d) Bus timing when transfers are suspended

Transfer is suspended when one transfer ends. Even if a transfer ending condition is satisfied during a read with the direct address transfer in the dual address mode, the subsequent write process is executed, and after the transfer in (a) to (c) above has been executed, DMAC operation suspends.

Conditions for Individual-Channel Ending: When one of the following conditions is satisfied, transfer in the corresponding channel ends.

When the value in the DMA transfer count register (DMATCR) is 0

When the DE bit in CHCR is cleared to 0.

- When DMATCR is 0: When the DMATCR value reaches 0, the DMA transfer in the corresponding channel ends and the transfer end flag bit (TE) in CHCR is set. If the IE (interrupt enable) bit is set at this time, a DMAC interrupt (DEI) is requested to the CPU. The conditions described in (a) to (d) above are not applied for this transfer ending.
- When DE in CHCR is 0: When the DE bit in CHCR is cleared, the DMA transfer in the corresponding channel stops. The conditions described in (a) to (d) above are not applied for this transfer ending.

Conditions for All-Channel Ending: When one of the following conditions is satisfied, transfer in all channels end simultaneously.

When the NMIF (NMI flag) bit in DMAOR is set to 1

When the DME bit in DMAOR is cleared to 0.

- When the NMIF bit in DMAOR is set to 1: When an NMI interrupt occurs, the NMIF bit in DMAOR is set to 1 and all channels stop their transfers according to the conditions in (a) to (d) described above, and pass the bus right to another bus master. Consequently, even if the NMI bit is set to 1 during transfer, the SAR, DAR, DMATCR are updated. Then the TE bit is not set. To resume the transfer after the NMI interrupt exception handling, clear the NMIF bit to 0. At this time, for the channels that should not be restarted, clear the corresponding DE bit in CHCR.
- When DME is cleared to 0 in DMAOR: Clearing the DME bit to 0 in the DMAOR forcibly aborts the transfers on all channels. Then the TE bit is not set. All channels abort their transfers according to the conditions (a) to (d) described in section 14.3.7, DMAC Transfer Ending, in the same way as that at the generation of an address error by the DMAC or NMI interrupt generation. In this case, the values in SAR, DAR, and DMATCR are also updated.

14.4 Compare-Match Timer (CMT)

14.4.1 Overview

The DMAC has an on-chip compare-match timer (CMT) to generate DMA transfer requests. The CMT is 16-bit counter.

Features

The CMT has the following features:

- Four types of counter input clocks can be selected
  — One of four internal clocks (Pϕ/4, Pϕ/8, Pϕ/16, and Pϕ/64) can be selected.
- Generates a DMA transfer request when a compare-match occurs.

Block Diagram

Figure 14.26 shows a CMT block diagram.

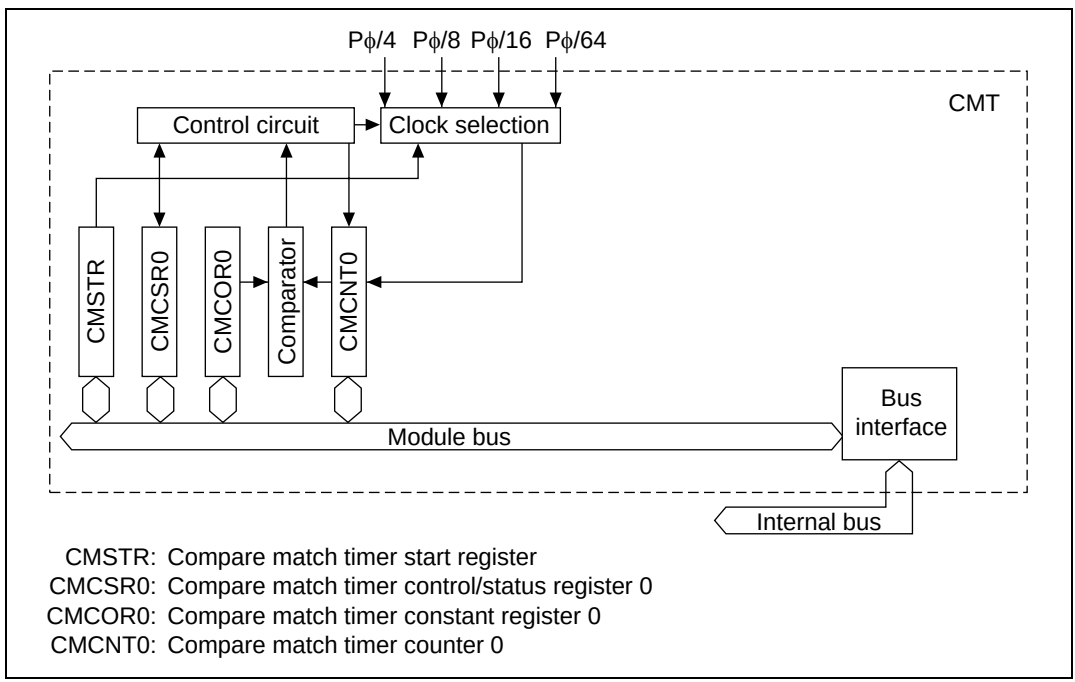


Figure 14.26 CMT Block Diagram

Register Configuration

Table 14.7 summarizes the CMT register configuration.

Table 14.7 Register Configuration

Name	Abbreviation	R/W	Initial Value	Address	Access Size (Bits)
Compare-match timer start register	CMSTR	R/(W)	H'0000	H'04000070 (H'A4000070)*2	8, 16, 32
Compare-match timer control/status register 0	CMCSR0	R/(W)*1	H'0000	H'04000072 (H'A4000072)*2	8, 16, 32
Compare-match counter 0	CMCNT0	R/W	H'0000	H'04000074 (H'A4000074)*2	8, 16, 32
Compare-match constant register	CMCOR0	R/W	H'FFFF	H'04000076 (H'A4000076)*2	8, 16, 32

Notes: 1. Only a 0 can be written to CMF bits in CMCSR0, to clear the flag.
2. When the address conversion by the MMU is not provided, use the address in parentheses.

14.4.2 Register Descriptions

Compare-Match Timer Start Register (CMSTR)

The compare-match timer start register (CMSTR) is a 16-bit register that selects whether compare-match counter 0 (CMCNT0) is operated or halted. CMSTR is initialized to H'0000 by a reset, but it retains its previous values in standby mode.

Bit:	15	14	13	12	11	10	9	8
	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	STR0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W

Bits 15 to 2—Reserved: These bits are always read as 0 and should only be written with 0.

Bit 1—Reserved: This is a readable/writable bit, but the write value should be always be 0.

Bit 0—Count start 0 (STR0): Selects whether the compare-match timer counter 0 is operated or halted.

Bit 0: STR0	Description	
0	CMCNT0 count operation is halted	(Initial value)
1	CMCNT0 count operation is provided	

Compare-Match Timer Control/Status Register 0 (CMCSR0)

The compare-match timer control/status register 0 (CMCSR0) is a 16-bit register that indicates a compare-match occurrence and sets the incrementation clock. CMCSR0 is initialized to H'0000 by a reset, but it retains its previous values in standby mode.

Bit:	15	14	13	12	11	10	9	8
	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit:	7	6	5	4	3	2	1	0
	CMF	—	—	—	—	—	CKS1	CKS0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/(W)*	R/W	R	R	R	R	R/W	R/W

Note: * Only a 0 can be written, to clear the flag.

Bits 15 to 8 and 5 to 2—Reserved: These bits are always read as 0and should only be written with 0.

Bit 7—Compare-Match Flag (CMF): This flag indicates that a compare-match of the compare-match timer counter 0 (CMCNT0) and compare-match constant register 0 (CMCOR0) occurred.

Bit 7: CMF	Description	
0	CMCNT0 and CMCOR0 have not matched Clear condition: Write 0 to CMF after reading CMF = 1	(Initial value)
1	A compare-match of CMCNT0 and CMCOR0 occurred	

Bit 6—Reserved: This is a readable/writable bit, but the write value should be always be 0.

Bits 1 and 0—Clock Select 1, 0 (CKS1, CKS0): These bits select the clock input to CMCNT from four clocks which are divided from the peripheral clock (Pϕ). When the STR0 bit in CMSTR is set to 1, the CMCNT0 starts incrementation with the clock selected by CKS1 and CKS0.

Bit 1: CKS1	Bit 0: CKS0	Description
0	0	Pϕ/4 (Initial value)
	1	Pϕ/8
1	0	Pϕ/16
	1	Pϕ/64

Compare-Match Counter 0 (CMCNT0)

The compare-match counter 0 (CMCNT0) is a 16-bit register that is used as an up-counter.

When the clock is selected with the CKS1 and CKS0 bits in CMCSR0 and the STR0 bit in CMSTR is set to 1, CMCNT0 starts incrementation with the selected clock. When the CMCNT0 value matches that in the compare-match constant register 0 (CMCOR0), the CMCNT0 is cleared to H'0000 and the CMF flag in CMCSR0 is set to 1.

CMCNT0 is initialized to H'0000 by a reset, but it retains its previous values in standby mode.

Bit:	15	14	13	12	11	10	9	8
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	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

Compare-Match Constant Register 0 (CMCOR0)

The compare-match constant register 0 (CMCOR0) is a 16-bit register that sets the period until a compare-match of CMCNT0 and CMCOR0 occurs.

The CMCOR0 is initialized to H'FFFF by a reset, but it retains its previous values in standby mode.

Bit:	15	14	13	12	11	10	9	8
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

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

14.4.3 Operation

Period Count Operation

When the clock is selected with the CKS1 and CKS0 bits in CMCSR0 and the STR0 bit in CMSTR is set to 1, the CMCNT0 starts incrementation with the selected clock. When the CMCNT value matches that in CMCOR0, CMCNT0 is cleared to H'0000 and the CMF flag in CMCSR0 is set to 1. The CMCNT0 counter starts incrementation again from H'0000.

Figure 14.27 shows the compare-match counter operation.

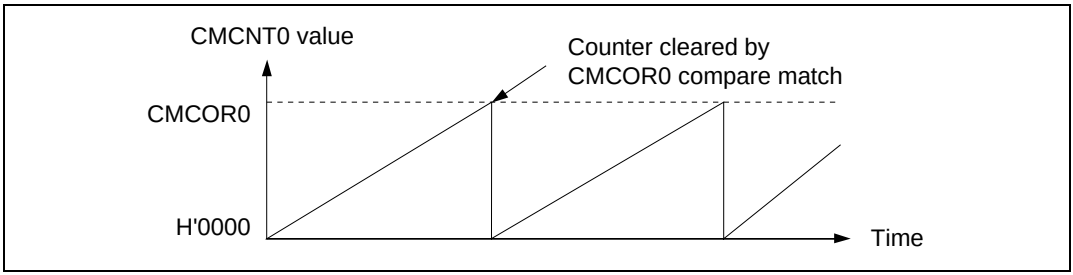


Figure 14.27 Counter Operation

CMCNT0 Count Timing

One of four peripheral clocks (Pϕ/4, Pϕ/8, Pϕ/16, Pϕ/64) which are divided from the clock (Pϕ) can be selected with the CKS1 and CKS0 bits in CMCSR0. Figure 14.28 shows the timing.

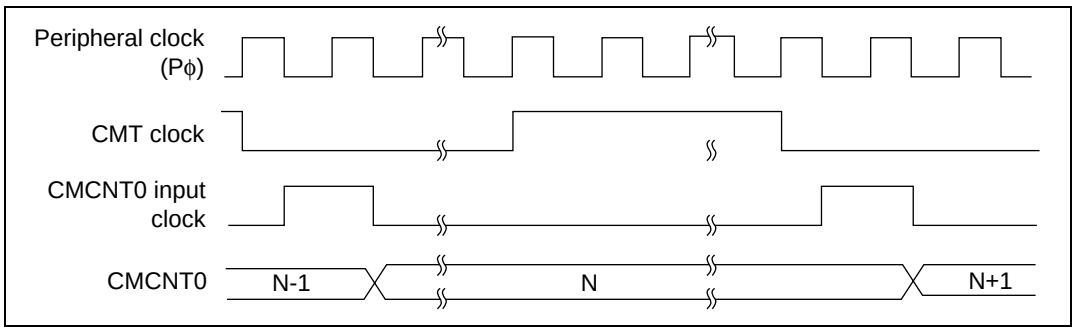


Figure 14.28 Count Timing

14.4.4 Compare-Match

Compare-Match Flag Set Timing

When the CMCOR0 register and the CMCNT0 counter match, a compare-match signal is generated and the CMF bit in the CMCSR0 register is set to 1. The compare-match signal is generated upon the final state of the match (timing at which the CMCNT0 counter value is updated). Consequently, after the CMCOR0 register and the CMCNT0 counter match, a compare-match signal will not be generated until a CMCNT0 counter input clock occurs. Figure 14.29 shows a timing of the CMF bit setting.

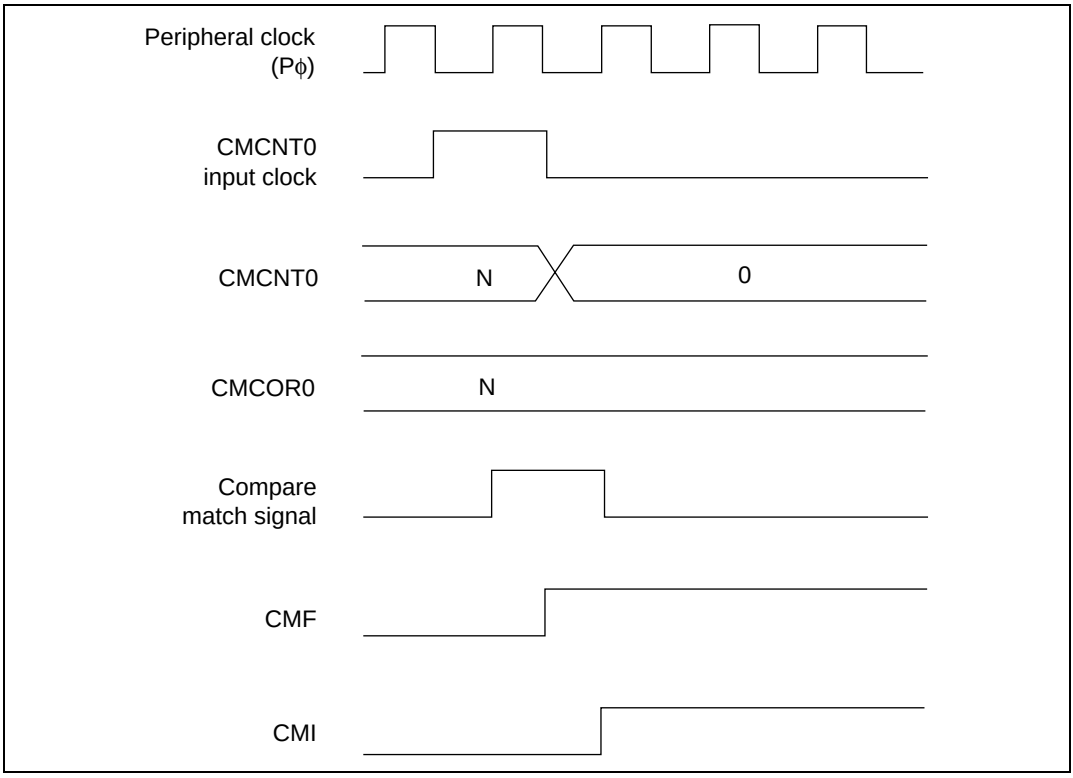


Figure 14.29 Timing of CMF Setting

Compare-Match Flag Clear Timing

The CMF bit in the CMCSR0 register is cleared by writing 0 to the bit after reading 1. Figure 14.30 shows the timing when the CMF bit is cleared by the CPU.

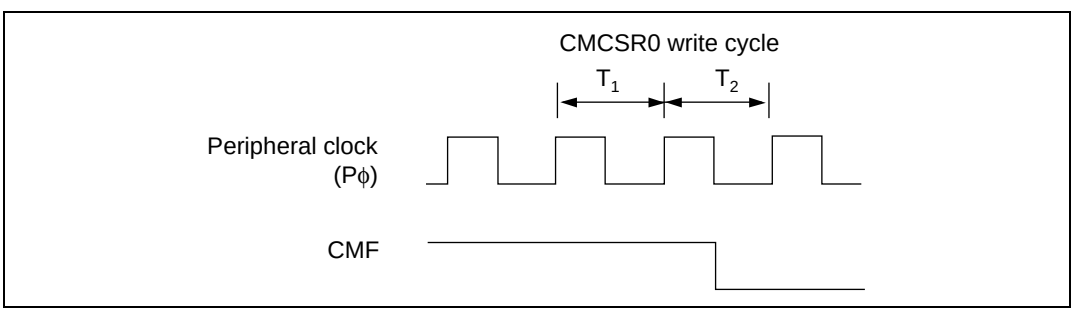


Figure 14.30 Timing of CMF Clear by the CPU

14.5 Examples for Use

14.5.1 Example of DMA Transfer between A/D Converter and External Memory (Address Reload on)

In this example, DMA transfer is performed between the on-chip A/D converter (transfer source) and the external memory (transfer destination) with the address reload function on. Table 14.8 shows the transfer conditions and register settings.

Table 14.8 Transfer Conditions and Register Settings for Transfer between On-Chip A/D Converter and External Memory

Transfer Conditions	Register	Setting
Transfer source: on-chip A/D converter	SAR2	H'04000080
Transfer destination: external memory	DAR2	H'00400000
Number of transfers: 128 (reloading 32 times)	DMATCR2	H'00000080
Transfer source address: incremented	CHCR2	H'00089E35
Transfer destination address: decremented		
Transfer request source: A/D converter		
Bus mode: burst		
Transfer unit: long word		
Interrupt request generated at end of transfer		
Channel priority order: 0 > 2 > 3 > 1	DMAOR	H'0101

When the address reload function is turned on, the value set in SAR returns to the initially set value at each four transfers. In this example, when an interrupt request is generated from the AD converter, longword data is read from the register in address H'04000080 of the A/D converter, and the data is written to external memory address H'00400000. Since longword data has been transferred, the values in SAR and DAR are H'04000084 and H'003FFFFC, respectively. The bus right is retained and data transfers are successively performed because this transfer is in the burst mode.

After four transfers end, fifth and sixth transfers are performed when the address reload function is turned off, and the value in SAR is incremented by 4, such as H'0400008C, H'04000090, H'04000094,.... When the address reload function is on, the DMA transfer stops after the fourth transfer ends and the bus request signal to the CPU is cleared. At this time, the value stored in SAR is not incremented from H'0400008C to H'04000090, but returns to the initially set value H'04000080. The value in DAR continues being incremented regardless of the address reload function setting.

The state in the DMAC differ depending on the address reload function setting as shown in table 14.9.

Table 14.9 DMAC Sate after the Fourth Transfer Ends

Items	Address Reload On	Address Reload Off
SAR	H'04000080	H'04000090
DAR	H'003FFFFC	H'003FFFFC
DMATCR	H'0000007C	H'0000007C
Bus right	Released	Held
DMAC operation	Stops	Continues operating
Interrupt	Not generated	Not generated
Transfer request source flag clear	Executed	Not executed

Notes: 1. When the value in DMATCR reaches 0 and the IE bit in CHCR has been set to 1, interrupts are generated regardless of the address reload function setting.
2. When the value in DMATCR reaches 0, the transfer request source flag is cleared regardless of the address reload function setting.
3. Specify the burst mode when using the address reload function. This function may not be correctly executed in the cycle steal mode.
4. Set the DMATCR value to a multiple of four when using the address reload function. This function may not be correctly executed if other values are specified.

14.5.2 Example of DMA Transfer between External Memory and SCIF Transmitter (Indirect Address on)

In this example, DMA transfer is performed between the external memory specified with the indirect address (transfer source) and the SCIF transmitter (transfer destination) using DMAC channel 3. Table 14.10 shows the transfer conditions and register settings. In addition, it is recommendable that the trigger for the number of transmit FIFO data is set to 1 (TTRG1 = TTRG0 = 1 in SCFCR).

Table 14.10 Transfer Conditions and Register Settings for Transfer between External Memory and SCIF Transmitter

Transfer Conditions	Register	Setting
Transfer source: external memory	SAR3	H'00400000
Value stored in address H'00400000	—	H'00450000
Value stored in address H'04500000	—	H'55
Transfer destination: On-chip SCIF TDR2	DAR3	H'04000156
Number of transfers: 10	DMATCR3	H'0000000A
Transfer source address: incremented	CHCR3	H'00011C01
Transfer destination address: fixed		
Transfer request source: SCIF (TXI2)		
Bus mode: cycle steal		
Transfer unit: byte		
No interrupt request generated at end of transfer		
Channel priority order: 0 > 1 > 2 > 3	DMAOR	H'0001

When the indirect address is on, data stored in the address set in SAR is not used as transfer source data. In the indirect address, after the value stored in the address set in SAR is read, the read value is used as an address again, and the value stored in the address is read and stored in the address set in DAR.

In the example shown in table 14.10, when an SCIF transfer request is generated, the DMAC reads the value in address H'00400000 that is set in SAR3. Since the value H'00450000 is stored in the address, the DMAC reads the value H'00450000. Next, the DMAC uses the read value as an address again, and reads the value H'55 stored in that address. Then, the DMAC writes the value H'55 to address H'04000156 that is set in DAR3; thus one indirect address transfer has completed.

In the indirect address, when data is read first from the address set in SAR3, the data transfer size is always longword regardless of the settings of the TS0 and the TS1 bits that specify the transfer data size. However, whether the transfer source address is fixed, incremented, or decremented is specified with the SM0 and SM1 bits. Therefore, in this example, though the transfer data size is specified as byte, the value in SAR3 is H'00400004 when one transfer ends. The write operation is the same as that in the normal dual address transfer.

14.6 Usage Notes

1. The DMA channel control registers (CHCR0 to CHCR3) can be accessed with any data size. The DMA operation register (DMAOR) must be accessed in byte (8 bits) or word (16 bits) units; other registers must be accessed in word (16 bits) or longword (32 bits) units.
2. When modifying the RS0 to RS3 bits in CHCR0 to CHCR3, first clear the DE bit to 0 (when modifying CHCR with a byte address, be sure to set the DE bit to 0 in advance).
3. If an NMI interrupt is input when the DMAC is not operating, the NMIF bit of the DMAOR is set.
4. A transition to standby mode should be made after the DME bit in DMAOR is cleared to 0 and the transfers that has been accepted by the DMAC end.
5. The on-chip supporting modules that the DMAC can access are, SIOF, SCIF, USB function, A/D converter, and I/O ports. Do not access the other on-chip supporting modules by the DMAC.
6. When starting up the DMAC, set CHCR or DMAOR last. Specifying other registers last does not guarantee normal operation.
7. When the DMA transfer ends normally and subsequently the maximum number of transfers is performed in the same channel, write 0 to DMATCR. Otherwise, normal DMA transfer may not be performed.
8. When using the address reload function, specify the burst mode for the transfer mode. In the cycle-steal mode, normal DMA transfer may not be performed.
9. When using the address reload function, set a multiple of four to DMATCR. Specifying other values does not guarantee normal operation.
10. When detecting an external request at the falling edge, keep the external request pin high when setting the DMAC.
11. Do not access the space from H'4000062 to H'400006F, which is not used by the DMAC. Accessing that space may cause malfunctions.
12. The $\overline{\text{WAIT}}$ signal is ignored when writing to an external address area using DMA 16-byte transfer in dual address mode, and also when transferring data from a DACK-equipped external device to an external address area using DMA 16-byte transfer in single address mode.
13. Big-endian access is used when transferring data from XY memory using the DMAC if all of the following conditions are met:

    Conditions:

    (1) Transfer source address in XY memory

    (2) Indirect addressing mode

    (3) Byte size data

    (4) Little-endian data transfer

Measures to avoid the problem:

The problem described above occurs only when all of the above conditions are met. It does not arise if even one of the conditions is not met. One of the methods listed below should therefore be employed when using the DMAC to transfer data from XY memory:

(1) Use the direct address mode

(2) Use long word size or word size data

(3) Use big-endian data transfer

14. Do not use the DMAC when in sleep mode. Alternately, set the clock ratio to Iϕ:Bϕ = 1:1 when using sleep mode. Normal operation cannot be guaranteed otherwise.
15. Do not use the DMAC when only the IFC[2:0] bits in the frequency control register (FRQCR) are modified and the clock ratio is set to other than Iϕ:Bϕ = 1:1. Normal operation cannot be guaranteed otherwise. However, there is no problem if the STC[2:0] bits are modified simultaneously with the IFC[2:0] bits in the frequency control register (FRQCR).

Section 15 Timer (TMU)

15.1 Overview

This LSI has an on-chip 32-bit timer unit (TMU) comprised of three 32-bit timer channels (channels 0 to 2).

15.1.1 Features

The TMU has the following features:

- Auto-reload 32-bit down-counters for each channel
- Auto-reload 32-bit constant registers and 32-bit down counters that can be read or written to at any time for each channel
- Interrupt request generation at the counter underflow:
  Interrupt requests can be generated when the 32-bit down counter underflows (H'00000000 → H'FFFFFFFF) in each channel.
- Selection of six counter input clocks for each channel:
  On-chip RTC output clock (16 kHz), Pϕ/4, Pϕ/16, Pϕ/64, and Pϕ/256
- All channels can operate when the SH7727 is in standby mode:
  When the RTC output clock is used as the counter input clock, the count operation is normally performed in standby mode.
- Synchronized read:
  TCNT is a 32-bit register that is successively modified. Since the internal bus for the SH7727 on-chip supporting modules is 16 bits wide, a time lag can occur between the time when the upper 16 bits and lower 16 bits are read. To correct the discrepancy in the counter read value caused by this time lag, a synchronization circuit is built in the TCNT so that the entire 32-bit data in the TCNT can be read at once.
- The maximum 2 MHz operating frequency for the 32-bit counter in each channel:
  Operate the SH7727 so that the clock input to each channel timer counter does not exceed the maximum operating frequency, by dividing the external clock and peripheral clock (Pϕ) with the prescaler.

15.1.2 Block Diagram

Figure 15.1 shows a block diagram of the TMU.

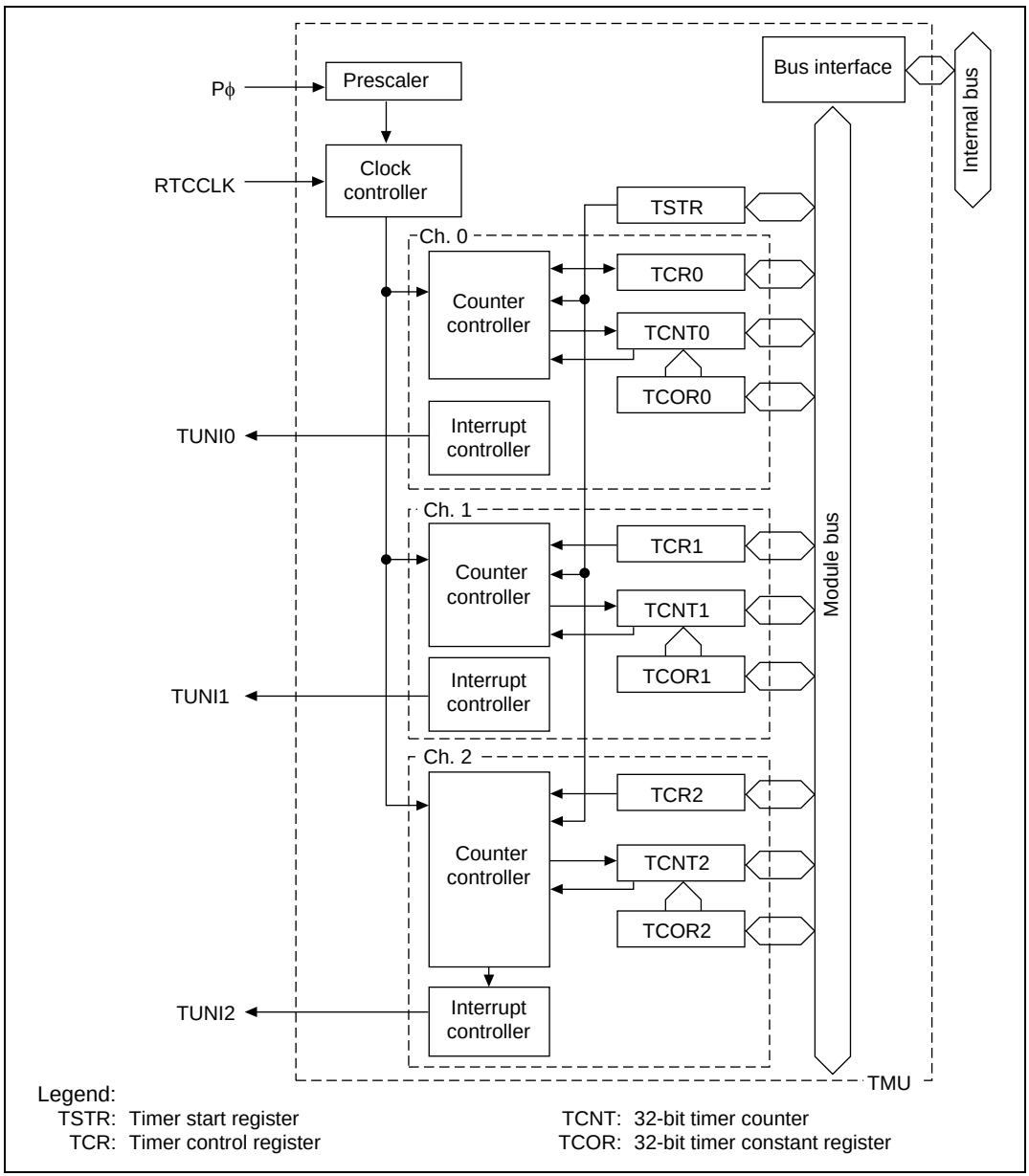


Figure 15.1 TMU Block Diagram

15.1.3 Register Configuration

Table 15.1 shows the TMU register configuration.

Table 15.1 TMU Register Configuration

Channel	Register	Abbre-viation	R/W	Initial Value*	Address	Access Size
Common	Timer start register	TSTR	R/W	H'00	H'FFFFFE92	8
0	Timer constant register 0	TCOR0	R/W	H'FFFFFFFF	H'FFFFFE94	32
	Timer counter 0	TCNT0	R/W	H'FFFFFFFF	H'FFFFFE98	32
	Timer control register 0	TCR0	R/W	H'0000	H'FFFFFE9C	16
1	Timer constant register 1	TCOR1	R/W	H'FFFFFFFF	H'FFFFFEA0	32
	Timer counter 1	TCNT1	R/W	H'FFFFFFFF	H'FFFFFEA4	32
	Timer control register 1	TCR1	R/W	H'0000	H'FFFFFEA8	16
2	Timer constant register 2	TCOR2	R/W	H'FFFFFFFF	H'FFFFFEAC	32
	Timer counter 2	TCNT2	R/W	H'FFFFFFFF	H'FFFFFEB0	32
	Timer control register 2	TCR2	R/W	H'0000	H'FFFFFEB4	16

Note: * Initialized by a power-on reset or manual reset.

15.2 TMU Registers

15.2.1 Timer Start Register (TSTR)

TSTR is an 8-bit read/write register that selects starting or stopping of the timer counters (TCNT) for channels 0 to 2. TSTR is initialized to H'00 by a power-on reset or manual reset. TSTR is not initialized in standby mode when the on-chip RTC clock (RTCCLK) is selected as the input clock for the channel. However, only if the peripheral clock (Pϕ) is selected for the channels, it is initialized in standby mode when the multiplying ratio of PLL circuit 1 is modified and when the MSTP2 bit in STBCR is set to 1.

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	—	STR2	STR1	STR0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W

Bits 7 to 3—Reserved: These bits are always read as 0 and should be written with 0.

Bit 2—Counter Start 2 (STR2): Selects starting or stopping of the timer counter 2 (TCNT2).

Bit 2: STR2	Description	
0	Halts TCNT2 operation	(Initial value)
1	Starts TCNT2 operation	

Bit 1—Counter Start 1 (STR1): starting or stopping of the timer counter 1 (TCNT1).

Bit 1: STR1	Description	
0	Halts TCNT1 operation	(Initial value)
1	Starts TCNT1 operation	

Bit 0—Counter Start 0 (STR0): Selects starting or stopping of the timer counter 0 (TCNT0).

Bit 0: STR0	Description	
0	Halts TCNT0 operation	(Initial value)
1	Starts TCNT0 operation	

15.2.2 Timer Control Register (TCR)

The timer control registers (TCR) are 16-bit read/write registers that control the timer counters (TCNT) and interrupts. The TMU has a total of three TCR registers, one for each channel.

The TCR registers control the interrupt generated when the flag that indicates the timer counter (TCNT) underflow has been set to 1, and select the counter clock. When an external clock has been selected, the clock edge can also be selected.

TCR is initialized to H'0000 by a power-on reset or manual reset. In standby mode, it is not initialized and retains the value.

Bit:	15	14	13	12	11	10	9	8
	—	—	—	—	—	—	—	UNF
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W

Bit:	7	6	5	4	3	2	1	0
	—	—	UNIE	—	—	TPSC2	TPSC1	TPSC0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R	R	R/W	R/W	R/W

Bits 15 to 9, 7, 6, 4, and 3—Reserved: These bits are always read as 0 and should only be written with 0.

Bit 8—Underflow Flag (UNF): This is a status flag that indicates that TCNT underflowed.

Bit 8: UNF	Description
0	TCNT has not underflowed. Clear condition: When 0 is written to UNF (Initial value)
1	TCNT has underflowed (H'00000000 → H'FFFFFFFF). Setting condition: When TCNT underflows*

Note: * When a write of 1 is provided to UNF, it is not modified and the previous value is retained.

Bit 5—Underflow Interrupt Control (UNIE): Controls enabling or disabling of interrupt generation when the status flag (UNF) that indicates TCNT underflow has been set to 1.

Bit 5: UNIE	Description	
0	Interrupt due to UNF (TUNI) is disabled.	(Initial value)
1	Interrupt due to UNF (TUNI) is enabled.	

Bits 2 to 0—Timer Prescalers 2 to 0 (TPSC2 to TPSC0): These bits select the TCNT count clock.

Bit 2: TPSC2	Bit 1: TPSC1	Bit 0: TPSC0	Description	
0	0	0	Counts on peripheral clock Pϕ/4	(Initial value)
		1	Counts on peripheral clock Pϕ/16	
	1	0	Counts on peripheral clock Pϕ/64	
		1	Counts on peripheral clock Pϕ/256	
1	0	0	Counts on on-chip RTC clock outputs (RTCCLK)	
		1	Reserved (Setting disabled)	
	1	0	Reserved (Setting disabled)	
		1	Reserved (Setting disabled)	

15.2.3 Timer Constant Register (TCOR)

The TMU has a total of three TCOR registers, one for each channel. The TCOR registers are 32-bit read/write registers that specify a value to be set to the TCNT counter after a TCNT counter underflow occurred.

TCOR is initialized to H'FFFFFFFF by a power-on reset or manual reset. In standby mode, it is not initialized and retains the value.

Bit:	31	30	29	28	27	26	25	24
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

Bit:	23	22	21	20	19	18	17	16
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

Bit:	15	14	13	12	11	10	9	8
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

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

15.2.4 Timer Counters (TCNT)

The TMU has a total of three timer counters (TCNT), one for each channel. The TCNT counters are 32-bit read/write registers that are decremented according to the input clock. The input clock can be selected with the TPSC2 to TPSC0 bits in the timer control register (TCR).

When a TCNT decrementation results in an underflow (H'00000000 → H'FFFFFFFF), the underflow flag (UNF) in the timer control register (TCR) of the relevant channel is set. The TCOR value is simultaneously set in TCNT itself and the decrementation continues from that value.

The TCNT counter is a 32-bit readable/writable register. Because the internal bus for the SH7727 on-chip peripheral modules is 16 bits wide, a time lag occurs when reading data from 32-bit registers because the upper 16 bits and lower 16 bits are read separately. Since TCNT counts sequentially, this time lag can create discrepancies between the data in the upper and lower halves. To prevent this, a buffer register is connected to TCNT so that upper and lower halves are not read separately. Thus all 32 bits in TCNT can thus be read at once and no timing discrepancies occur when reading data.

TCNT is initialized to H'FFFFFFFF by a power-on reset or manual reset. In standby mode, it is not initialized and retains the value.

Bit:	31	30	29	28	27	26	25	24
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

Bit:	23	22	21	20	19	18	17	16
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

Bit:	15	14	13	12	11	10	9	8
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

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

15.3 TMU Operation

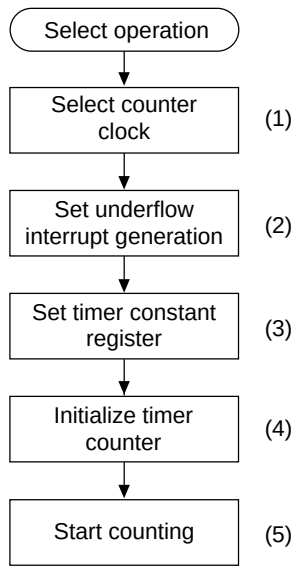
15.3.1 Overview

Each channel has a 32-bit timer counter (TCNT) and a 32-bit timer constant register (TCOR). The TCNT is a down-counter. The auto-reload function can be used to enable synchronized counting and counting by external events.

15.3.2 Basic Functions

Counter Operation: When the STR0 to STR2 bits in the timer start register (TSTR) are set, the corresponding timer counters (TCNT) start decrementation. When TCNT underflows, the UNF flag in the corresponding timer control register (TCR) is set. At this time, if the UNIE bit in TCR is 1, an interrupt request is sent to the CPU. Also at this time, the value is copied from TCOR to TCNT and the decrementation is continued.

The decrementation is set as follows (figure 15.2):



(1) Select the counter clock with the TPSC2 to TPSC0 bits in the timer control register (TCR).

(2) Set whether or not an interrupt is generated when TCNT underflows, with the UNIE bit in TCR.

(3) Set a value in the timer constant register (TCOR) (the cycle is the set value plus 1).

(4) Set the initial value in the timer counter (TCNT).

(5) Set the STR bit in the timer start register (TSTR) to 1 to start operation.

Note: When an interrupt has been generated, clear the flag in the interrupt handler that caused it.
If interrupts are enabled without clearing the flag, another interrupt will be generated.

Figure 15.2 Setting the Count Operation

Auto-Reload Count Operation: Figure 15.3 shows the TCNT auto-reload operation.

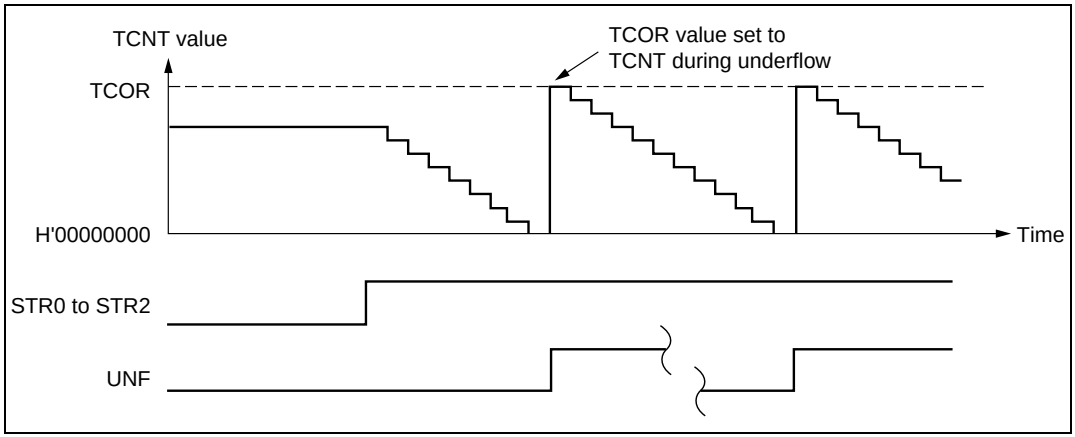


Figure 15.3 Auto-Reload Counter Operation

TCNT Count Timing:

- Internal Clock Operation

Select one of the four internal clocks (Pϕ/4, Pϕ/16, Pϕ/64, Pϕ/256), which are divided from the peripheral clock Pϕ, with the TPSC2 to TPSC0 bits in TCR. Figure 15.4 shows the timing.

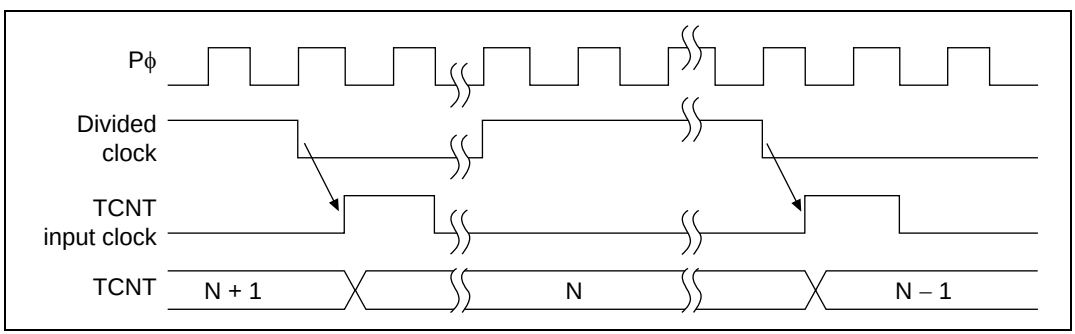


Figure 15.4 Count Timing when Internal Clock is Operating

• On-Chip RTC Clock Operation

Select the on-chip RTC clock as the timer clock with the TPSC2 to TPSC0 bits in TCR.

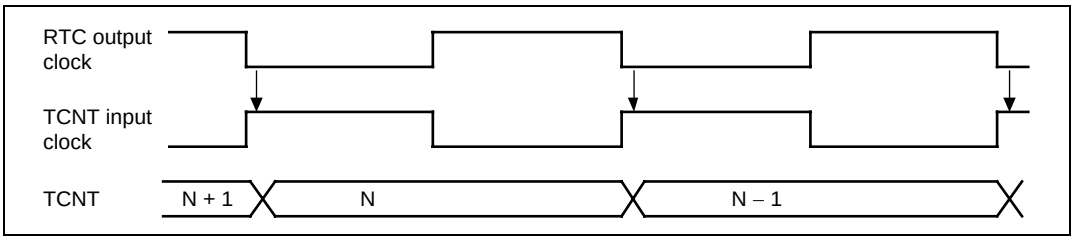


Figure 15.5 Count Timing when On-Chip RTC Clock is Operating

15.4 Interrupts

There is only one source for TMU interrupts: underflow interrupts (TUNI).

15.4.1 Status Flag Set Timing

The UNF bit is set to 1 when TCNT underflows. Figure 15.6 shows the timing.

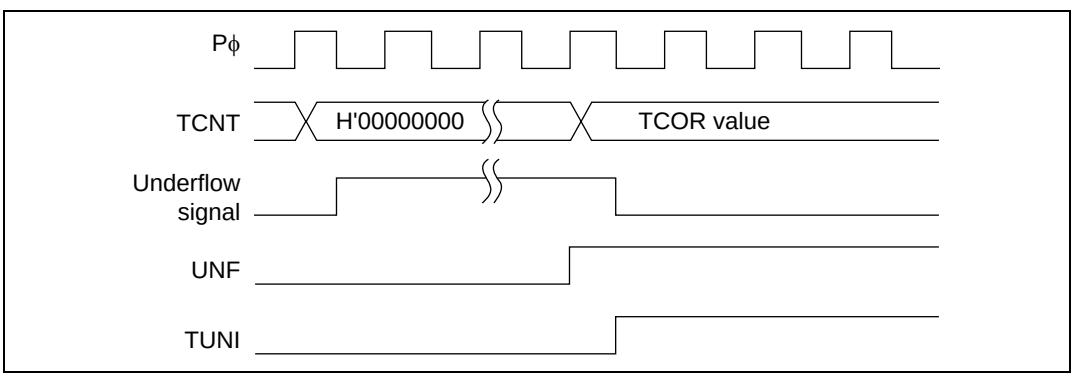


Figure 15.6 UNF Set Timing

15.4.2 Status Flag Clear Timing

The status flag is cleared when 0 is written by the CPU. Figure 15.7 shows the timing.

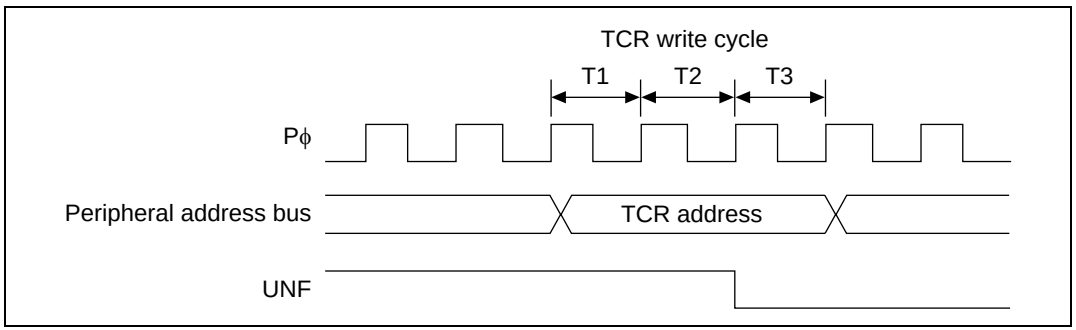


Figure 15.7 Status Flag Clear Timing

15.4.3 Interrupt Sources and Priorities

The TMU generates underflow interrupts for each channel. When the interrupt request flag and interrupt enable bit are both set to 1, the corresponding interrupt is requested. When an interrupt is generated, codes are set in the interrupt event register (INTEVT, INTEVT2). Provide the appropriate interrupt handling according to the codes.

The channel priority can be changed using the interrupt controller (see section 4, Exception Handling, and section 7, Interrupt Controller (INTC)). Table 15.2 lists TMU interrupt sources.

Table 15.2 TMU Interrupt Sources

Channel	Interrupt Source	Description	Priority
0	TUNI0	Underflow interrupt 0	High
1	TUNI1	Underflow interrupt 1	↕
2	TUNI2	Underflow interrupt 2	Low

15.5 Usage Notes

15.5.1 Writing to Registers

Synchronous processing is not performed for timer count operation during register writes. When writing to registers, always clear the start bits (STR2 to STR0) for the desired channel in the timer start register (TSTR) to halt timer counting.

15.5.2 Reading Registers

Synchronous processing is performed for timer count operation during register reads. When timer counting and register read processing are performed simultaneously, the register value prior to the TCNT decrementation is read with the synchronous processing.

Section 16 Realtime Clock (RTC)

16.1 Overview

This LSI has a realtime clock (RTC) with its own 32.768-kHz crystal oscillation circuit.

16.1.1 Features

The RTC has the following features:

- Clock and calendar functions (BCD display): Seconds, minutes, hours, date, day of the week, month, and year
- 1 to 64-Hz timer (binary display)
- Start/stop function
- 30-second adjustment
- Alarm interrupt:
  Frame comparisons of seconds, minutes, hours, date, day of the week, and month can be selected for the alarm interrupt condition
- Periodic interrupts:
  The interrupt cycle can be selected from 1/256 second, 1/64 second, 1/16 second, 1/4 second, 1/2 second, 1 second, or 2 seconds
- Carry interrupt: A carry interrupt indicates when a carry occurs during a counter read
- Automatic leap year correction

16.1.2 Block Diagram

Figure 16.1 shows a block diagram of the RTC.

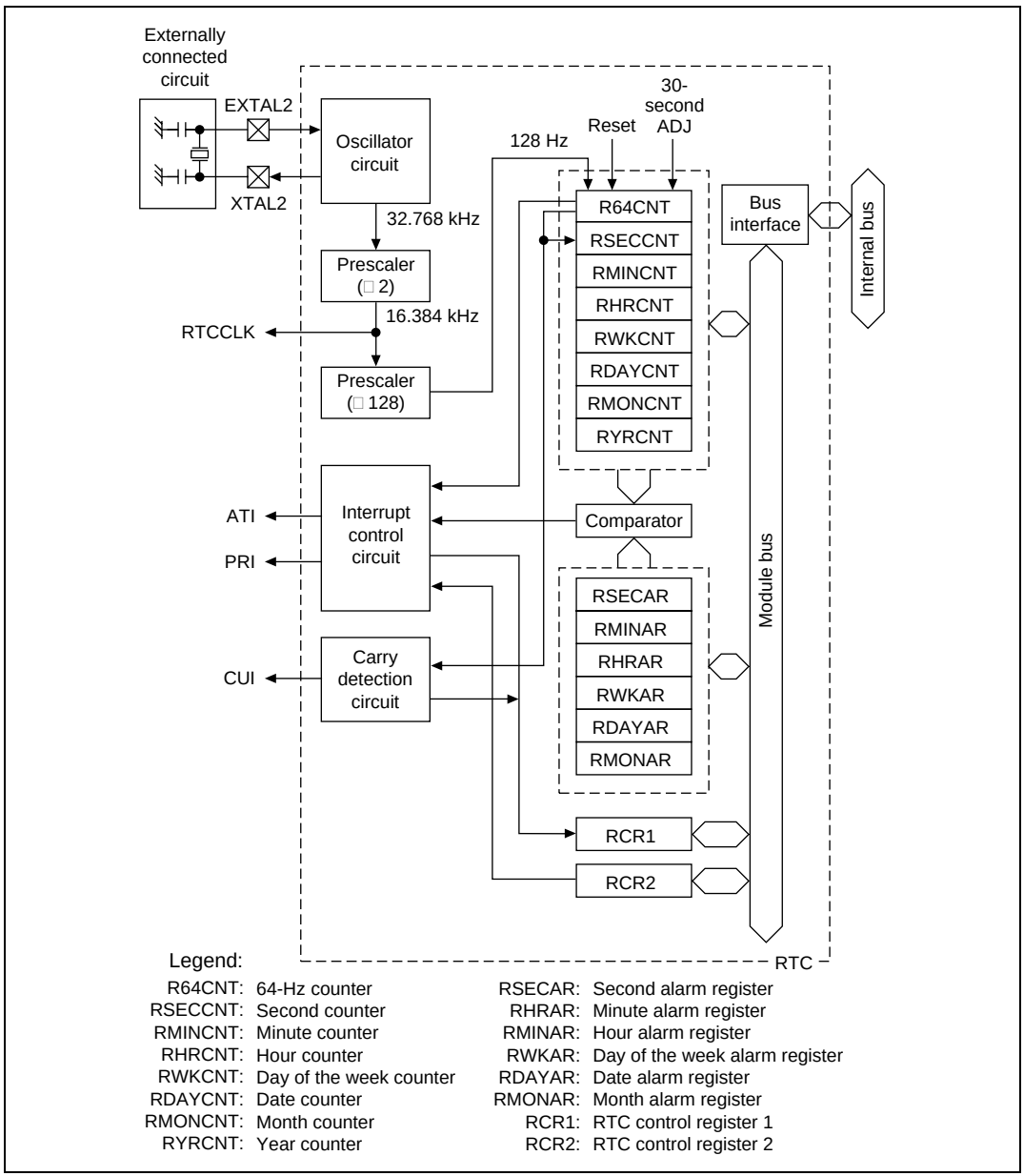


Figure 16.1 RTC Block Diagram

16.1.3 Pin Configuration

Table 16.1 shows the RTC pin configuration.

Table 16.1 RTC Pin Configuration

Pin	Abbreviation	I/O	Description
RTC oscillator crystal pin	EXTAL2	I	Connects crystal to RTC oscillator*1
RTC oscillator crystal pin	XTAL2	O	Connects crystal to RTC oscillator*1
Power-supply pin dedicated for RTC	Vcc-RTC	—	Power-supply pin for RTC oscillator*1
GND pin dedicated for RTC	Vss-RTC	—	GND pin for RTC oscillator*2

Notes: 1. When the RTC is not used, set EXTAL2 to pull-up (to Vcc) and make no connection for XTAL2.

2. Input of external noise via the Vss-RTC pin can cause the device to malfunction. To prevent external noise input via the Vss-RTC, the system and circuitry should include a noise elimination circuit.

16.1.4 RTC Register Configuration

Table 16.2 shows the RTC register configuration.

Table 16.2 RTC Registers

Name	Abbreviation	R/W	Initial Value	Address	Access Size
64-Hz counter	R64CNT	R	Undefined	H'FFFFFEC0	8
Second counter	RSECCNT	R/W	Undefined	H'FFFFFEC2	8
Minute counter	RMINCNT	R/W	Undefined	H'FFFFFEC4	8
Hour counter	RHRCNT	R/W	Undefined	H'FFFFFEC6	8
Day of week counter	RWKCNT	R/W	Undefined	H'FFFFFEC8	8
Date counter	RDAYCNT	R/W	Undefined	H'FFFFFECA	8
Month counter	RMONCNT	R/W	Undefined	H'FFFFFECC	8
Year counter	RYRCNT	R/W	Undefined	H'FFFFFECE	8
Second alarm register	RSECAR	R/W	Undefined*	H'FFFFFED0	8
Minute alarm register	RMINAR	R/W	Undefined*	H'FFFFFED2	8
Hour alarm register	RHRAR	R/W	Undefined*	H'FFFFFED4	8
Day of week alarm register	RWKAR	R/W	Undefined*	H'FFFFFED6	8
Date alarm register	RDAYAR	R/W	Undefined*	H'FFFFFED8	8
Month alarm register	RMONAR	R/W	Undefined*	H'FFFFFEDA	8
RTC control register 1	RCR1	R/W	H'00	H'FFFFFEDC	8
RTC control register 2	RCR2	R/W	H'09	H'FFFFFEDE	8

Note: * Only the ENB bit in each register is initialized.

16.2 Register Descriptions

16.2.1 64-Hz Counter (R64CNT)

The 64-Hz counter (R64CNT) is an 8-bit read-only register that indicates the state of the RTC divider circuits (RTC prescaler or R64CNT) between 64 Hz and 1 Hz.

R64CNT is reset to H'00 when the RESET bit in RTC control register 2 (RCR2) or the ADJ bit in RCR2 is set to 1.

R64CNT is not initialized by a power-on reset or manual reset, or in standby mode, and the operation is continued.

Bit 7 is always read as 0.

Bit:	7	6	5	4	3	2	1	0
	—	1Hz	2Hz	4Hz	8Hz	16Hz	32Hz	64Hz
Initial value:	0	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R

16.2.2 Second Counter (RSECCNT)

The second counter (RSECCNT) is an 8-bit read/write register that is used for setting/counting in the BCD-coded second section of the RTC. The count operation is performed by a carry for each second of the 64-Hz counter.

The settable range is 00 to 59 in decimal. If other values are set, correct operation is not provided. When modifying RSECCNT, check that the count operation has been halted with the START bit in RCR2.

RSECCNT is not initialized by a power-on reset or manual reset, or in standby mode, and the operation is continued.

Bit:	7	6	5	4	3	2	1	0
	—	10 seconds			1 second			
Initial value:	0	—	—	—	—	—	—	—
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

16.2.3 Minute Counter (RMINCNT)

The minute counter (RMINCNT) is an 8-bit read/write register used for setting/counting in the BCD-coded minute section of the RTC. The count operation is performed by a carry for each minute of the second counter.

The settable range is 00 to 59 in decimal. If other values are set, correct operation is not provided. When modifying RMINCNT, check that the count operation has been halted with the START bit in RCR2.

RMINCNT is not initialized by a power-on reset or manual reset, or in standby mode, and the operation is continued.

Bit:	7	6	5	4	3	2	1	0
	—	10 minutes			1 minute			
Initial value:	0	—	—	—	—	—	—	—
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

16.2.4 Hour Counter (RHRCNT)

The hour counter (RHRCNT) is an 8-bit read/write register used for setting/counting in the BCD-coded hour section of the RTC. The count operation is performed by a carry for each 1 hour of the minute counter.

The settable range is 00 to 23 in decimal. If other values are set, correct operation is not provided. When modifying RHRCNT, check that the count operation has been halted with the START bit in RCR2.

RHRCNT is not initialized by a power-on reset or manual reset, or in standby mode, and the operation is continued.

Bit:	7	6	5	4	3	2	1	0
	—	—	10 hours		1 hour			
Initial value:	0	0	—	—	—	—	—	—
R/W:	R	R	R/W	R/W	R/W	R/W	R/W	R/W

16.2.5 Day of the Week Counter (RWKCNT)

The day of the week counter (RWKCNT) is an 8-bit read/write register used for setting/counting in the BCD-coded day of week section of the RTC. The count operation is performed by a carry for each day of the date counter.

The settable range is 0 to 6 in decimal. If other values are set, correct operation is not provided. When modifying RWKCNT, check that the count operation has been halted with the START bit in RCR2.

RWKCNT is not initialized by a power-on reset or manual reset, or in standby mode, and the operation is continued.

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	—	Day of week		
Initial value:	0	0	0	0	0	—	—	—
R/W:	R	R	R	R	R	R/W	R/W	R/W

Days of the week are coded as shown in table 16.3.

Table 16.3 Day-of-Week Codes (RWKCNT)

Day of Week	Code
Sunday	0
Monday	1
Tuesday	2
Wednesday	3
Thursday	4
Friday	5
Saturday	6

16.2.6 Date Counter (RDAYCNT)

The date counter (RDAYCNT) is an 8-bit read/write register used for setting/counting in the BCD-coded date section of the RTC. The count operation is performed by a carry for each day of the hour counter.

The settable range is 01 to 31 in decimal. If other values are set, correct operation is not provided. When modifying RDAYCNT, check that the count operation has been halted with the START bit in RCR2.

RDAYCNT is not initialized by a power-on reset or manual reset, or in standby mode, and the operation is continued.

The settable RDAYCNT range differs according to the month and leap year. Please confirm the correct setting.

Bit:	7	6	5	4	3	2	1	0
	—	—	10 days		1 day			
Initial value:	0	0	—	—	—	—	—	—
R/W:	R	R	R/W	R/W	R/W	R/W	R/W	R/W

16.2.7 Month Counter (RMONCNT)

The month counter (RMONCNT) is an 8-bit read/write register used for setting/counting in the BCD-coded month section of the RTC. The count operation is performed by a carry for each month of the date counter.

The settable range is 01 to 12 in decimal. If other values are set, correct operation is not provided. When modifying RMONCNT, check that the count operation has been halted with the START bit in RCR2.

RMONCNT is not initialized by a power-on reset or manual reset, or in standby mode, and the operation is continued.

Bit:	7	6	5	4	3	2	1	0
	—	—	—	10 months	1 month			
Initial value:	0	0	0	—	—	—	—	—
R/W:	R	R	R	R/W	R/W	R/W	R/W	R/W

16.2.8 Year Counter (RYRCNT)

The year counter (RYRCNT) is an 8-bit read/write register used for setting/counting in the BCD-coded year section of the RTC. The least significant 2 digits of the western calendar year are displayed. The count operation is performed by a carry for each year of the month counter.

The settable range is 00 to 99 in decimal. If other values are set, correct operation is not provided. When modifying RYRCNT, check that the count operation is halted with the START bit in RCR2.

RYRCNT is not initialized by a power-on reset or manual reset, or in standby mode, and the operation is continued.

Leap years are recognized by dividing the year counter value by 4 and obtaining a fractional result of 0. Note that a counter value of 00 is treated as a leap year.

Bit:	7	6	5	4	3	2	1	0
	10 years				1 year			
Initial value:	—	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

16.2.9 Second Alarm Register (RSECAR)

The second alarm register (RSECAR) is an 8-bit read/write alarm register that corresponds to the BCD-coded second section counter RSECCNT of the RTC. When the ENB bit is set to 1in RSECAR, the RSECAR value and RSECCNT value are compared. In this way, the RSECAR, RMINAR, RHRAR, RWKAR, RDAYAR, and RMONAR registers are checked, and when the ENB bit is set to 1, the alarm register and the corresponding counter are compared. If all values in the specified alarm registers and the corresponding counters match, an RTC alarm interrupt is generated.

The settable range is “00 to 59 in decimal + ENB bit”. If other values are set, correct operation is not provided.

Only the ENB bit in RSECAR is initialized to 0 by a power-on reset, and the other bits are not initialized. The RSECAR contents are retained after a manual reset or in standby mode.

Bit:	7	6	5	4	3	2	1	0
	ENB	10 seconds			1 second			
Initial value:	0	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

16.2.10 Minute Alarm Register (RMINAR)

The minute alarm register (RMINAR) is an 8-bit read/write alarm register that corresponds to the BCD-coded minute section counter RMINCNT of the RTC. When the ENB bit is set to 1in RMINAR, the RMINAR value and RMINCNT value are compared. In this way, the RSECAR, RMINAR, RHRAR, RWKAR, RDAYAR, and RMONAR registers are checked, and when the ENB bit is set to 1, the alarm register and the corresponding counter are compared. If all values in the specified alarm registers and the corresponding counters match, an RTC alarm interrupt is generated.

The settable range is “00 to 59 in decimal + ENB bit”. If other values are set, correct operation is not provided.

Only the ENB bit in RMINAR is initialized to 0 by a power-on reset, and the other bits are not initialized. The RMINAR contents are retained after a manual reset or in standby mode.

Bit:	7	6	5	4	3	2	1	0
	ENB	10 minutes			1 minute			
Initial value:	0	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

16.2.11 Hour Alarm Register (RHRAR)

The hour alarm register (RHRAR) is an 8-bit read/write alarm register that corresponds to the BCD-coded hour section counter RHRCNT of the RTC. When the ENB bit is set to 1in RHRAR, the RHRAR value and RHRCNT value are compared. In this way, the RSECAR, RMINAR, RHRAR, RWKAR, RDAYAR, and RMONAR registers are checked, and when the ENB bit is set to 1, the alarm register and the corresponding counter are compared. If all values in the specified alarm registers and the corresponding counters match, an RTC alarm interrupt is generated.

The settable range is “00 to 23 in decimal + ENB bit”. If other values are set, correct operation is not provided.

Only the ENB bit in RHRAR is initialized to 0 by a power-on reset, and the other bits are not initialized. The RHRAR contents are retained after a manual reset or in standby mode.

Bit:	7	6	5	4	3	2	1	0
	ENB	—	10 hours		1 hour			
Initial value:	0	0	—	—	—	—	—	—
R/W:	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W

16.2.12 Day of the Week Alarm Register (RWKAR)

The day of the week alarm register (RWKAR) is an 8-bit read/write alarm register that corresponds to the BCD-coded day of week section counter RWKCNT of the RTC. When the ENB bit is set to 1in RWKAR, the RWKAR value and RWKCNT value are compared. In this way, the RSECAR, RMINAR, RHRAR, RWKAR, RDAYAR, and RMONAR registers are checked, and when the ENB bit is set to 1, the alarm register and the corresponding counter are compared. If all values in the specified alarm registers and the corresponding counters match, an RTC alarm interrupt is generated.

The settable range is “0 to 6 in decimal + ENB bit”. If other values are set, correct operation is not provided.

Only the ENB bit in RWKAR is initialized to 0 by a power-on reset, and the other bits are not initialized. The RWKAR contents are retained after a manual reset or in standby mode.

Bit:	7	6	5	4	3	2	1	0
	ENB	—	—	—	—	Day of week		
Initial value:	0	0	0	0	0	—	—	—
R/W:	R/W	R	R	R	R	R/W	R/W	R/W

Days of the week are coded as shown in table 16.4.

Table 16.4 Day-of-Week Codes (RWKAR)

Day of Week	Code
Sunday	0
Monday	1
Tuesday	2
Wednesday	3
Thursday	4
Friday	5
Saturday	6

16.2.13 Date Alarm Register (RDAYAR)

The date alarm register (RDAYAR) is an 8-bit read/write alarm register that corresponds to the BCD-coded date section counter RDAYCNT of the RTC. When the ENB bit is set to 1in RDAYAR, the RDAYAR value and RDAYCNT value are compared. In this way, the RSECAR,

RMINAR, RHRAR, RWKAR, RDAYAR, and RMONAR registers are checked, and when the ENB bit is set to 1, the alarm register and the corresponding counter are compared. If all values in the specified alarm registers and the corresponding counters match, an RTC alarm interrupt is generated.

The settable range is “01 to 31 in decimal + ENB bit”. If other values are set, correct operation is not provided. The settable RDAYCNT range differs according to the month and leap year. Please confirm the correct setting.

Only the ENB bit in RDAYAR is initialized to 0 by a power-on reset, and the other bits are not initialized. The RDAYAR contents are retained after a manual reset or in standby mode.

Bit:	7	6	5	4	3	2	1	0
	ENB	—	10 days		1 day			
Initial value:	0	0	—	—	—	—	—	—
R/W:	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W

16.2.14 Month Alarm Register (RMONAR)

The month alarm register (RMONAR) is an 8-bit read/write alarm register that corresponds to the BCD-coded month section counter RMONCNT of the RTC. When the ENB bit is set to 1in RMONAR, the RMONAR value and RMONCNT value are compared. In this way, the RSECAR, RMINAR, RHRAR, RWKAR, RDAYAR, and RMONAR registers are checked, and when the ENB bit is set to 1, the alarm register and the corresponding counter are compared. If all values in the specified alarm registers and the corresponding counters match, an RTC alarm interrupt is generated.

The settable range is “01 to 12 in decimal + ENB bit”. If other values are set, correct operation is not provided.

Only the ENB bit in RMONAR is initialized to 0 by a power-on reset, and the other bits are not initialized. The RMONAR contents are retained after a manual reset or in standby mode.

Bit:	7	6	5	4	3	2	1	0
	ENB	—	—	10 months	1 month			
Initial value:	0	0	0	—	—	—	—	—
R/W:	R/W	R	R	R/W	R/W	R/W	R/W	R/W

16.2.15 RTC Control Register 1 (RCR1)

The RTC control register 1 (RCR1) is an 8-bit read/write register that contains carry flags and alarm flags. It also selects whether to generate interrupts for each flag. Avoid the use of read-modify-write processing for this register because flags are sometimes set after an operand read.

RCR1 is an 8-bit read/write register. The CIE, AIE, and AF bits are initialized by a power-on reset or manual reset. However, the value of the CF flag is undefined after a power-on reset or manual reset. It must therefore be initialized without fail before use. This register is not initialized in standby mode.

Bit:	7	6	5	4	3	2	1	0
	CF	—	—	CIE	AIE	—	—	AF
Initial value:	—	0	0	0	0	0	0	0
R/W:	R/W	R	R	R/W	R/W	R	R	R/W

Bit 7—Carry Flag (CF): Status flag that indicates that a carry has occurred. CF is set to 1 when a carry occurs in R64CNT or RSECCNT. If the count register is read at this time, the value is not guaranteed; therefore, another read is required.

Bit 7: CF	Description
0	No carry in R64CNT or RSECCNT. Clearing condition: When 0 is written to CF
1	Setting condition: Carry occurred in RSECCNT Read of R64CNT at carry occurrence When 1 is written to CF

Bits 6, 5, 2, and 1—Reserved: These bits are always read as 0 and should only be written with 0.

Bit 4—Carry Interrupt Enable Flag (CIE): Enables or disables interrupt generation when the carry flag (CF) is set to 1.

Bit 4: CIE	Description	
0	A carry interrupt is not generated when the CF flag is set to 1	(Initial value)
1	A carry interrupt is generated when the CF flag is set to 1	

Bit 3—Alarm Interrupt Enable Flag (AIE): Enables or disables interrupt generation when the alarm flag (AF) is set to 1.

Bit 3: AIE	Description
0	An alarm interrupt is not generated when the AF flag is set to 1 (Initial value)
1	An alarm interrupt is generated when the AF flag is set to 1

Bit 0—Alarm Flag (AF): The AF flag is set to 1 when the alarm time set in alarm registers (only for the registers with ENB bit set to 1) match the clock and calendar time. This flag is cleared to 0 when 0 is written, but the previous value is retained when 1 is to be written.

Bit 0: AF	Description
0	Clock/calendar and alarm register have not matched since last reset to 0. Clearing condition: When 0 is written to AF (Initial value)
1	Setting condition: Clock/calendar and alarm register have matched (only for the registers with ENB set to 1)*

Note: * The value is not modified when 1 is written to AF.

16.2.16 RTC Control Register 2 (RCR2)

The RTC control register 2 (RCR2) is an 8-bit read/write register that controls periodic interrupts, 30-second adjustment ADJ, divider circuits RESET, and starting and stopping of the RTC count. It is initialized to H'09 by a power-on reset. By a manual reset, bits except RTCEN and START are initialized. RCR2 is not initialized and retains its contents in standby mode.

Bit:	7	6	5	4	3	2	1	0
	PEF	PES2	PES1	PES0	RTCEN	ADJ	RESET	START
Initial value:	0	0	0	0	1	0	0	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7—Periodic Interrupt Flag (PEF): Indicates that interrupts are generated with the period designated by the PES bits. When this bit is set to 1, periodic interrupts are generated.

Bit 7: PEF	Description
0	Interrupts not generated with the period designated by the PES bits. Clearing condition: When 0 is written to PEF (Initial value)
1	Setting condition: When interrupts are generated with the period designated by the PES bits When 1 is written to PEF

Bits 6 to 4—Periodic Interrupt Flags (PES2 to PES0): These bits specify the periodic interrupt.

Bit 6: PES2	Bit 5: PES1	Bit 4: PES0	Description
0	0	0	No periodic interrupts generated (Initial value)
		1	Periodic interrupt generated every 1/256 second
	1	0	Periodic interrupt generated every 1/64 second
		1	Periodic interrupt generated every 1/16 second
1	0	0	Periodic interrupt generated every 1/4 second
		1	Periodic interrupt generated every 1/2 second
	1	0	Periodic interrupt generated every 1 second
		1	Periodic interrupt generated every 2 seconds

Bit 3—RTCEN: Controls the operation of the crystal oscillator for the RTC.

Bit 3: RTCEN	Description
0	Halts the crystal oscillator for the RTC. *
1	Runs the crystal oscillator for the RTC. * (Initial value)

Note: * RTCEN should be set to 0 when the RTC is not used.

Bit 2—30 Second Adjustment (ADJ): When the ADJ bit is written with 1, the time of 29 seconds or less will be rounded to 00 seconds and the time of 30 seconds or more to 1 minute. The divider circuits (RTC prescaler and R64CNT) will be simultaneously reset. This bit is always read as 0.

Bit 2: ADJ	Description	
0	Normal operation	(Initial value)
1 (write)	30-second adjustment.	

Bit 1—Reset (RESET): When 1 is written to the RESET bit, the divider circuits (RTC prescaler and R64CNT) are initialized. This bit is always read as 0.

Bit 1: RESET	Description	
0	Runs normally.	(Initial value)
1 (Write)	Divider circuits are reset.	

Bit 0—Start Bit (START): Halts and restarts the counter (clock).

Bit 0: START	Description
0	Second, minute, hour, day, week, month, and year counters are halted.*
1	Second, minute, hour, day, week, month, and year counters operate normally.* (Initial value)

Note: * The 64-Hz counter operates normally until it is stopped with the RTCEN bit.

16.3 RTC Operation

16.3.1 Initial Settings of Registers after Power-On

All RTC registers should be set initially after the power is turned on.

16.3.2 Setting the Time

Figures 16.2 (a) and 16.2 (b) show how to set the time after stopping the clock. This procedure can be used to set the entire calendar and clock function. It can be programmed easily.

Usage Notes

1. Initialization Timing for 64 Hz Counter (R64CNT)
   If it is necessary, after initializing the counter by means of the RESET bit in the RTC's RCR2 register, to confirm that the change has taken effect by reading the R64CNT value, wait at least 107 µs after setting the RESET bit to 1 before reading the R64CNT counter. Note that the divider circuit (RTC prescaler) is also initialized when the RESET bit is set to 1.

2. Incrementing RSECCNT by Initializing R64CNT
   Either method (a) or method (b) below may be used.
   (a) After setting the RESET bit to 1 and confirming that R64CNT has been initialized, set the START bit to 1. This process is shown in figure 16.2 (a).
   (b) Set the START bit to 1 and the RESET bit to 1 at the same time. This process is shown in figure 16.2 (b). Note that the processing indicated by the asterisk (*) in figure 16.2 (b) may be omitted if nothing is written to the RCR2 register during an interval of approximately 107 µs after the START bit is set to 1.

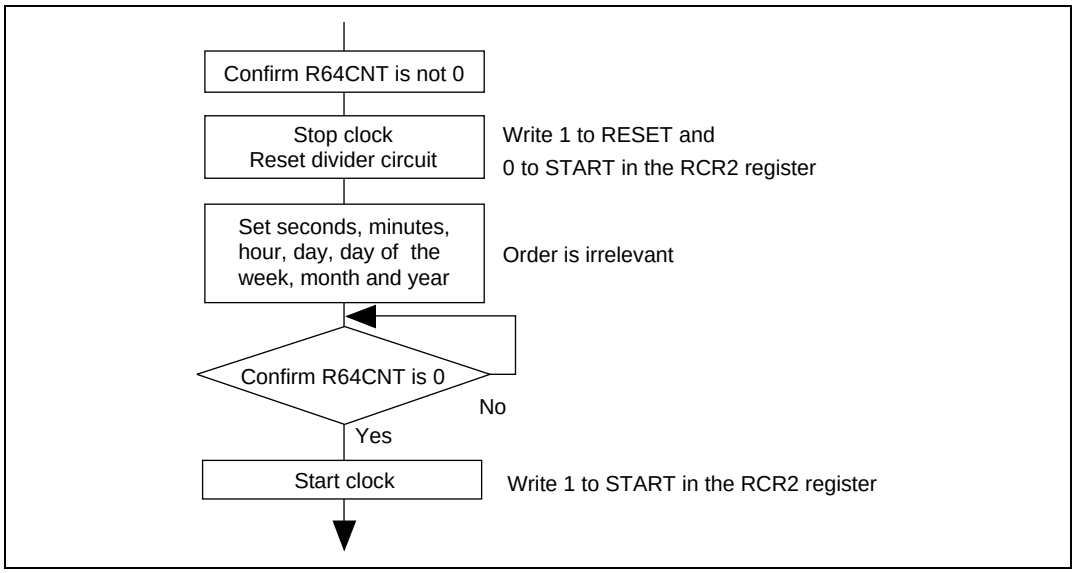


Figure 16.2(a) Setting the Time

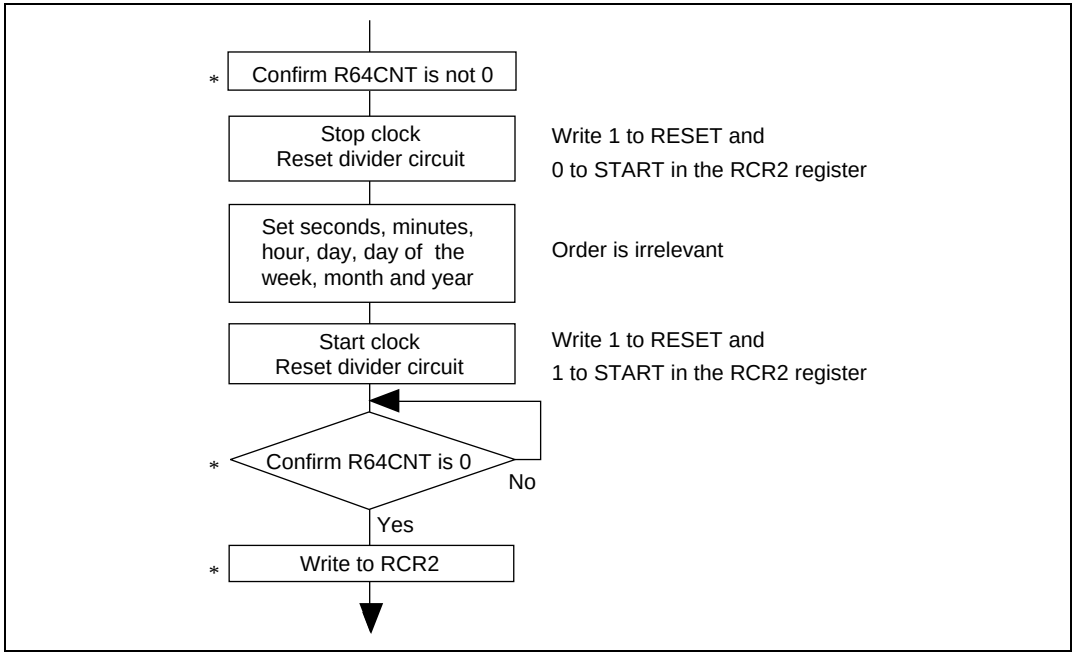


Figure 16.2(b) Setting the Time

16.3.3 Reading the Time

Figure 16.3 shows how to read the time. If a carry occurs while reading the time, the correct time will not be obtained, so it must be read again. The method of reading the time without using interrupts is shown at (a) in figure 16.3, and the method using carry interrupts is shown at (b). To keep the program simple, method (a) is used normally.

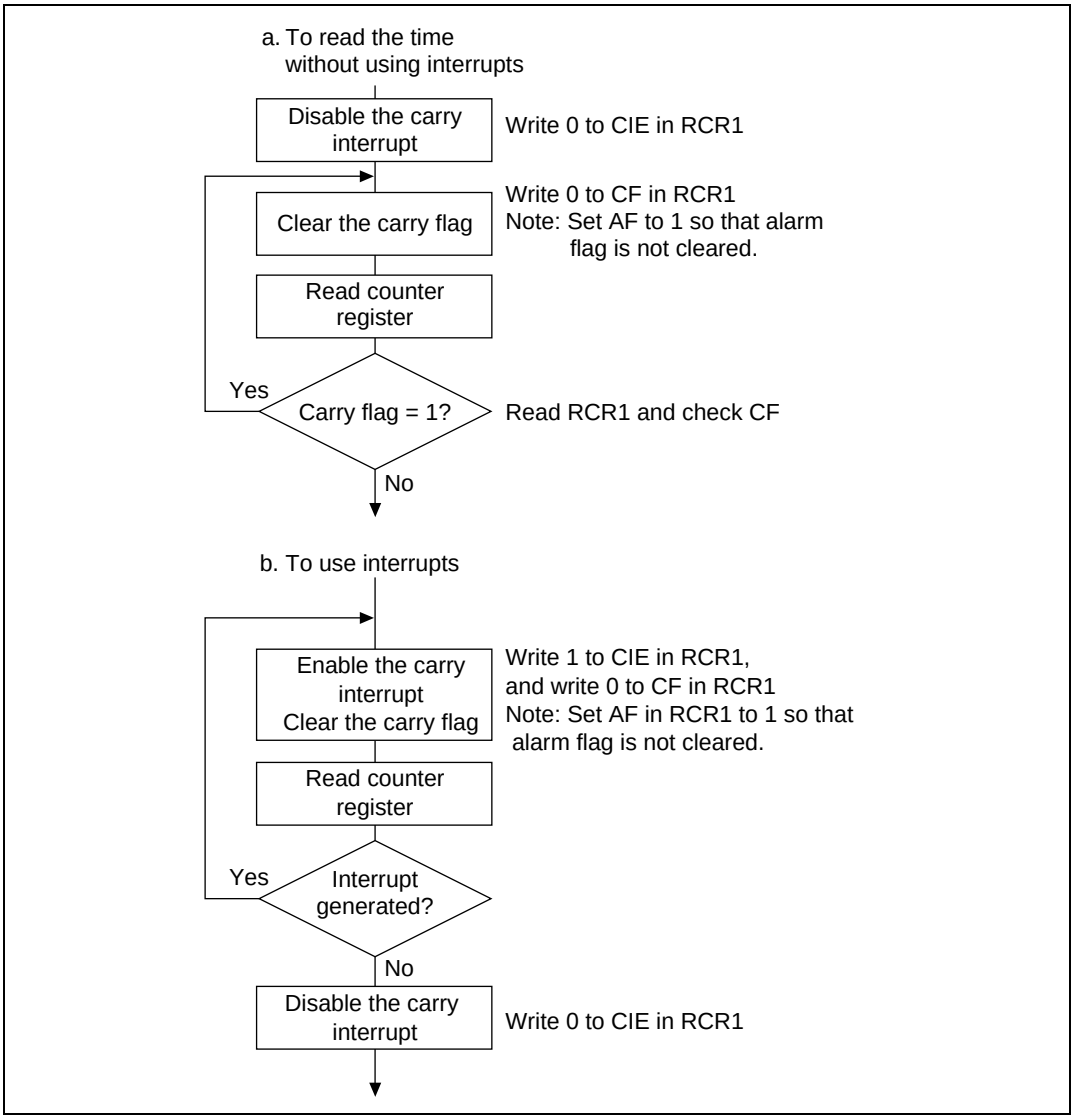


Figure 16.3 Reading the Time

16.3.4 Alarm Function

Figure 16.4 shows how to use the alarm function.

Alarm can be generated using seconds, minutes, hours, day of the week, date, month, or any combination of these. Set the ENB bits (bit 7) in the desired alarm registers to 1, and then set the alarm time in the lower bits. Clear the ENB bits in the registers which are not used for the alarm to 0.

When the clock and alarm time match, the AF bit (bit 0) in RCR1 is set to 1. The alarm detection can be checked by reading this bit, but normally it is checked by the interrupt generation. If the AIE bit (bit 3) in RCR1 is written with 1, an interrupt is generated when an alarm occurs.

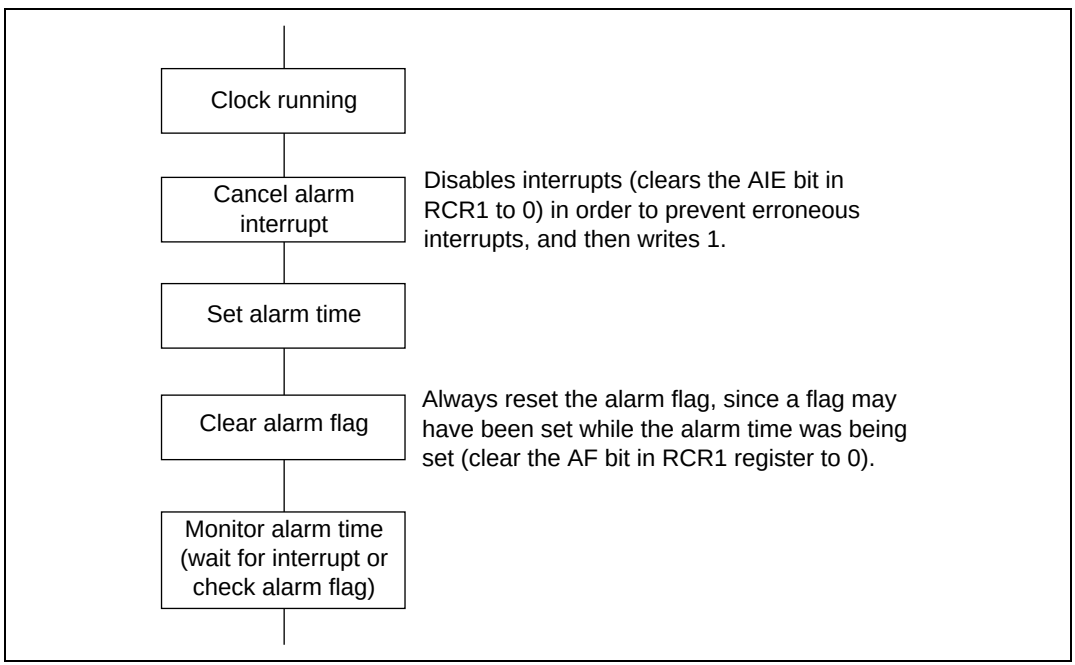


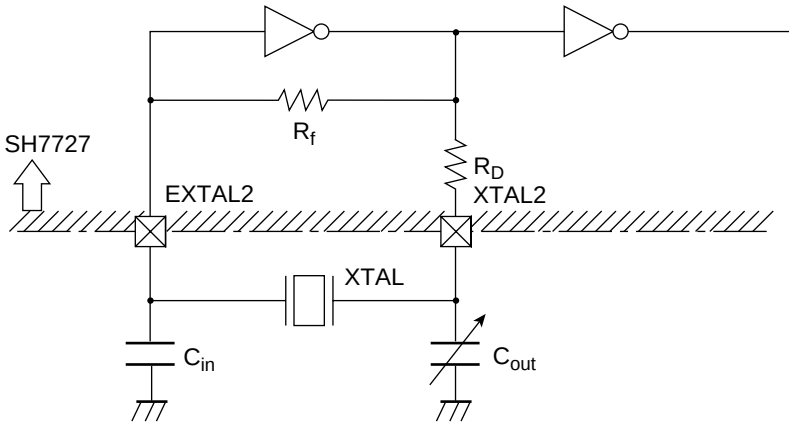
Figure 16.4 Using the Alarm Function

16.3.5 Crystal Oscillator Circuit

Crystal oscillator circuit constants (recommended values) are shown in table 16.5, and the RTC crystal oscillator circuit in figure 16.5.

Table 16.5 Recommended Oscillator Circuit Constants (Recommended Values)

f_{osc}	C_{in}	C_{out}
32.768 kHz	10 to 22 pF	10 to 22 pF



Notes:
1. Select either the C_{in} or C_{out} side for frequency adjustment variable capacitor according to requirements such as frequency range, degree of stability, etc.
2. Built-in resistance value R_f (Typ value) = 10 MΩ, R_D (Typ value) = 400 kΩ
3. C_{in} and C_{out} values include floating capacitance due to the wiring. Take care when using a ground plane.
4. The crystal oscillation settling time depends on the mounted circuit constants, floating capacitance, etc., and should be decided after consultation with the crystal resonator manufacturer.
5. Place the crystal resonator and load capacitors C_{in} and C_{out} as close as possible to the chip.
   (Correct oscillation may not be possible if there is externally induced noise in the EXTAL2 and XTAL2 pins.)
6. Ensure that the crystal resonator connection pin (EXTAL2, XTAL2) wiring is routed as far away as possible from other power lines (except GND) and signal lines.

Figure 16.5 Example of Crystal Oscillator Circuit Connection

16.4 Usage Notes

16.4.1 Writing Registers During RTC Count Operation

During the RTC count operation (RCR2 bits 0 = 1), the following registers cannot be written.

RSECCNT, RMICNT, RHRCNT, RDAYCNT, RWKCNT, RMONCNT, and RYRCNT

To write these registers, the RTC count operation should be stopped.

16.4.2 RTC Periodic Interrupts

Figure 16.6 shows the periodic interrupt function setting flow.

Periodic interrupts can be generated with the period specified by the periodic interrupt enable flag (PES) in the RTC control register (RCR2). When the time period specified by PES passed, the periodic interrupt flag (PEF) is set to 1.

PEF is cleared to 0 when PES is set and a periodic interrupt is generated. The periodic interrupt generation can be checked by reading this bit, but is usually checked by the interrupt function.

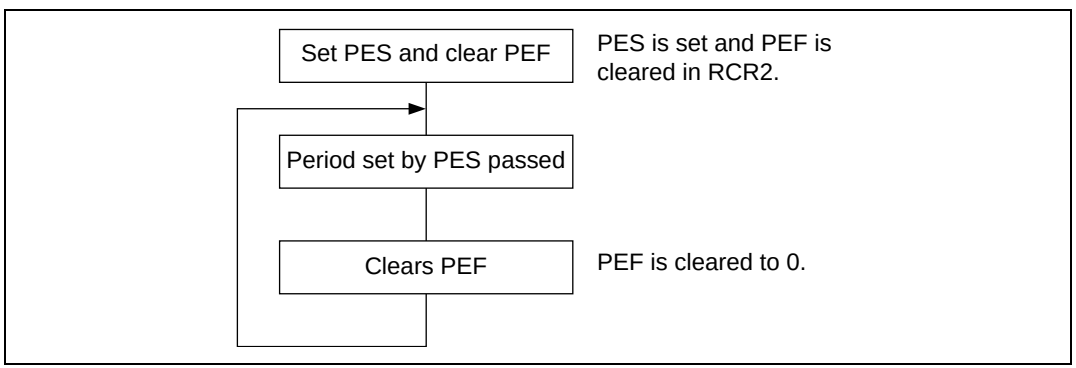


Figure 16.6 Periodic Interrupt Function Setting

16.4.3 Using the ADJ Bit in the Real Time Clock (RTC)

(1) Description

The maximum amount of time from when the ADJ bit in RCR2 of the RTC is set to 1 and when the value read from the second counter (RSECCNT) is reflected is approximately 91.6 μs (the time required for pin of the EXTAL2 to connect to the 32.768 kHz oscillator). Note that the second counter itself performs a 30-second adjustment when the ADJ bit is set to 1, so the above delay causes no problems with the functioning of the RTC.

(2) Precautions

If it is necessary to ensure that the 30-second adjustment triggered by the ADJ bit in RCR2 of the RTC is properly read and its value reflected, the second counter should not be read until a minimum of approximately 91.6 μs has passed following the setting of the ADJ bit.

Section 17 Serial Communication Interface (SCI)

17.1 Overview

This LSI has an on-chip serial communication interface (SCI) that supports both asynchronous and clock synchronous serial communication. It also has a multiprocessor communication function for serial communication among two or more processors. The SCI supports a smart card interface, which is a serial communications feature for IC card interfaces that conforms to the ISO/IEC standard 7816-3 for identification cards data transmission protocol type T = 0. See section 18, Smart Card Interface, for more information.

17.1.1 Features

Select asynchronous or clock synchronous as the serial communications mode.

- Asynchronous mode:
  - Serial data communications are synched by start-stop in character units. The SCI can communicate with a universal asynchronous receiver/transmitter (UART), an asynchronous communication interface adapter (ACIA), or any other communications chip that employs a standard asynchronous serial system. It can also communicate with two or more other processors using the multiprocessor communication function. There are 12 selectable serial data communication formats.
  - Data length: Seven or eight bits
  - Stop bit length: One or two bits
  - Parity: Even, odd, or none
  - Multiprocessor bit: 1 or 0
  - Receive error detection: Parity, overrun, and framing errors
  - Break detection: By reading the RxD level directly from the port SC data register (SCSPTR) when a framing error occurs
- Clock synchronous mode:
  - Serial data communication is synchronized with a clock signal. The SCI can communicate with other chips having a clock synchronous communication function. There is one serial data communication format.
  - Data length: Eight bits
  - Receive error detection: Overrun errors
- Full duplex communication: The transmitting and receiving sections are independent, so the SCI can transmit and receive simultaneously. Both sections use double buffering, so continuous data transfer is possible in both the transmit and receive directions.

- On-chip baud rate generator with selectable bit rates
- Internal or external transmit/receive clock source: From either baud rate generator (internal) or SCK pin (external)
- Four types of interrupts: Transmit-data-empty, transmit-end, receive-data-full, and receive-error interrupts are requested independently.
- When the SCI is not in use, it can be stopped by halting the clock supplied to it, saving power.

17.1.2 Block Diagram

Figure 17.1 shows an SCI block diagram.

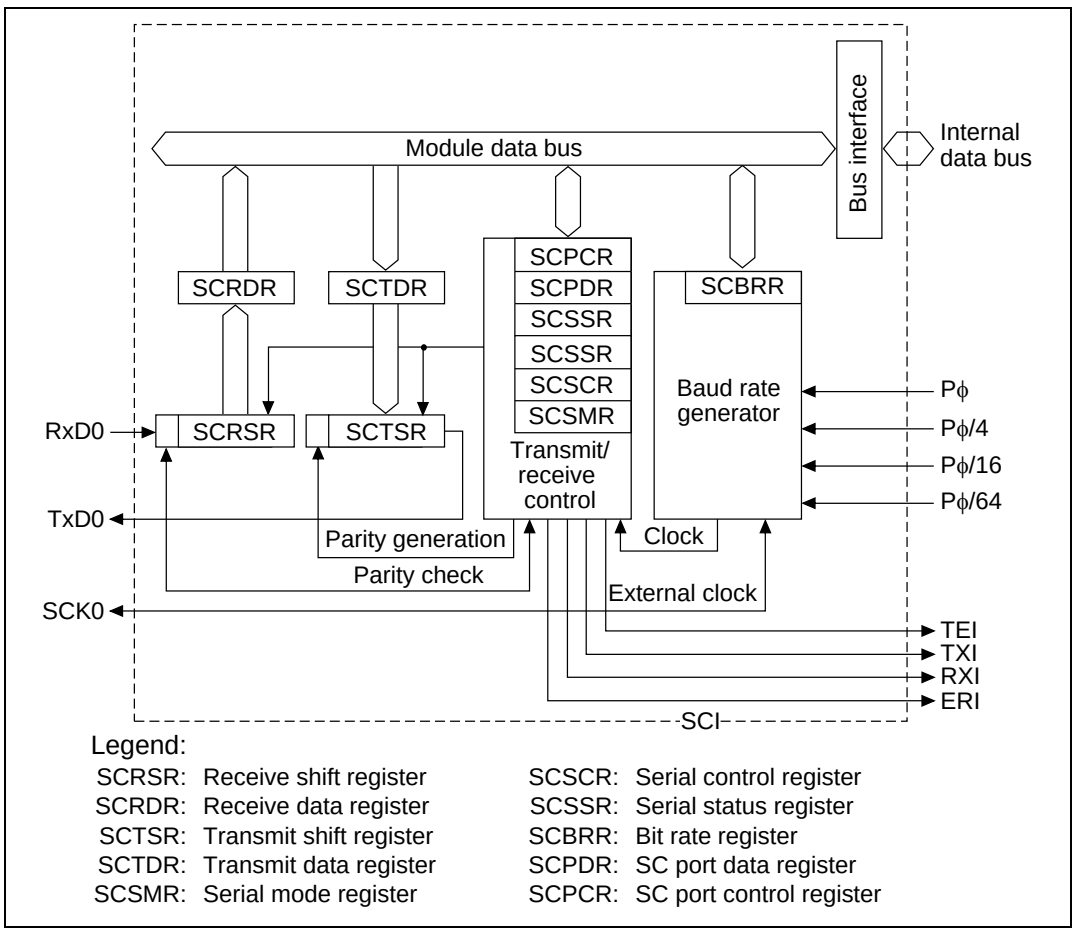


Figure 17.1 SCI Block Diagram

Figures 17.2 to 17.4 show the block diagrams of the SCI I/O port.

SCI pin I/O and data control is performed by bits 3 to 0 of SCPCR and bits 1 and 0 of SCPDR. For details, see section 17.2.8, Port SC Control Register (SCPCR)/Port SC Data Register (SCPDR).

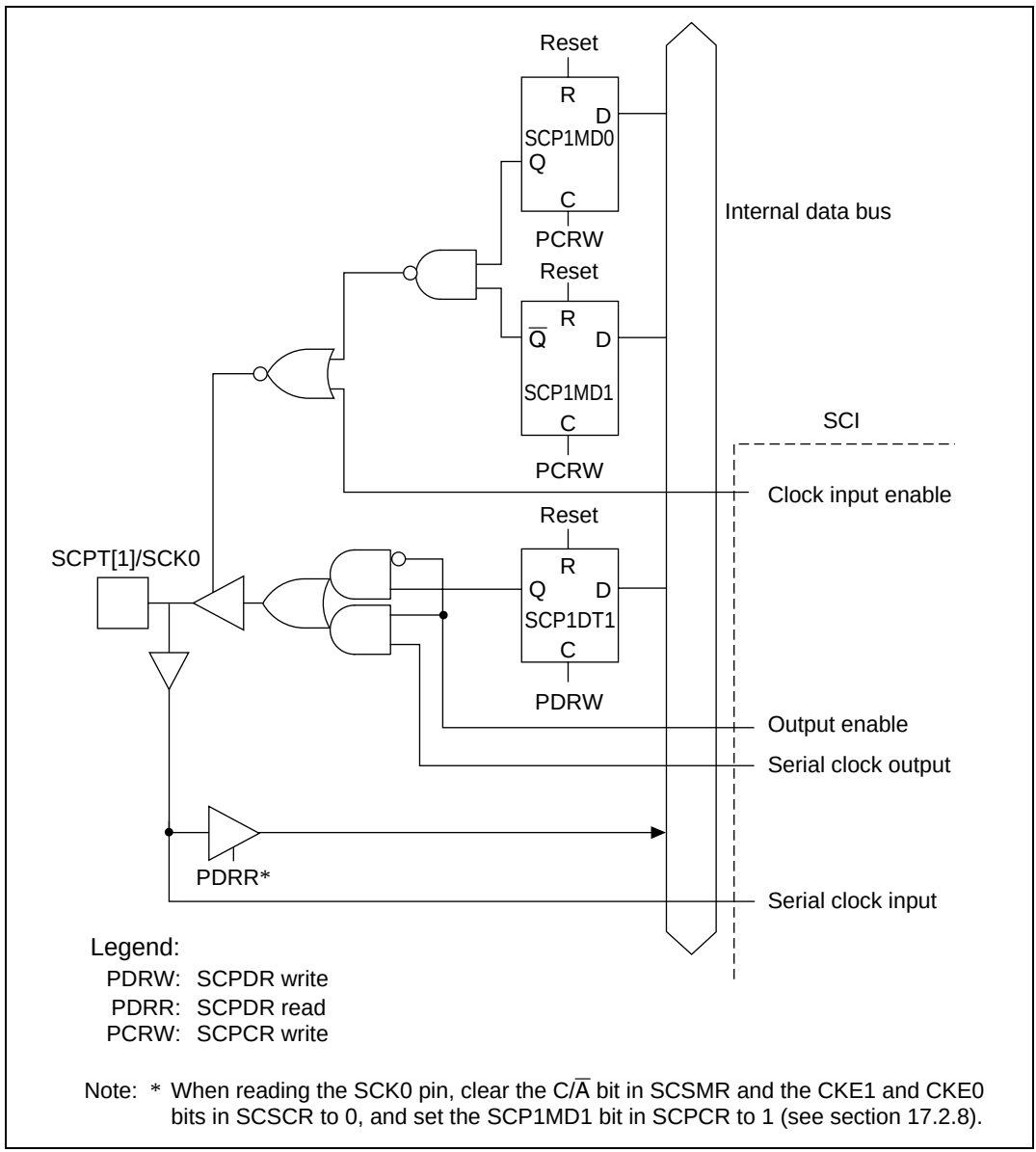


Figure 17.2 SCPT[1]/SCK0 Pin

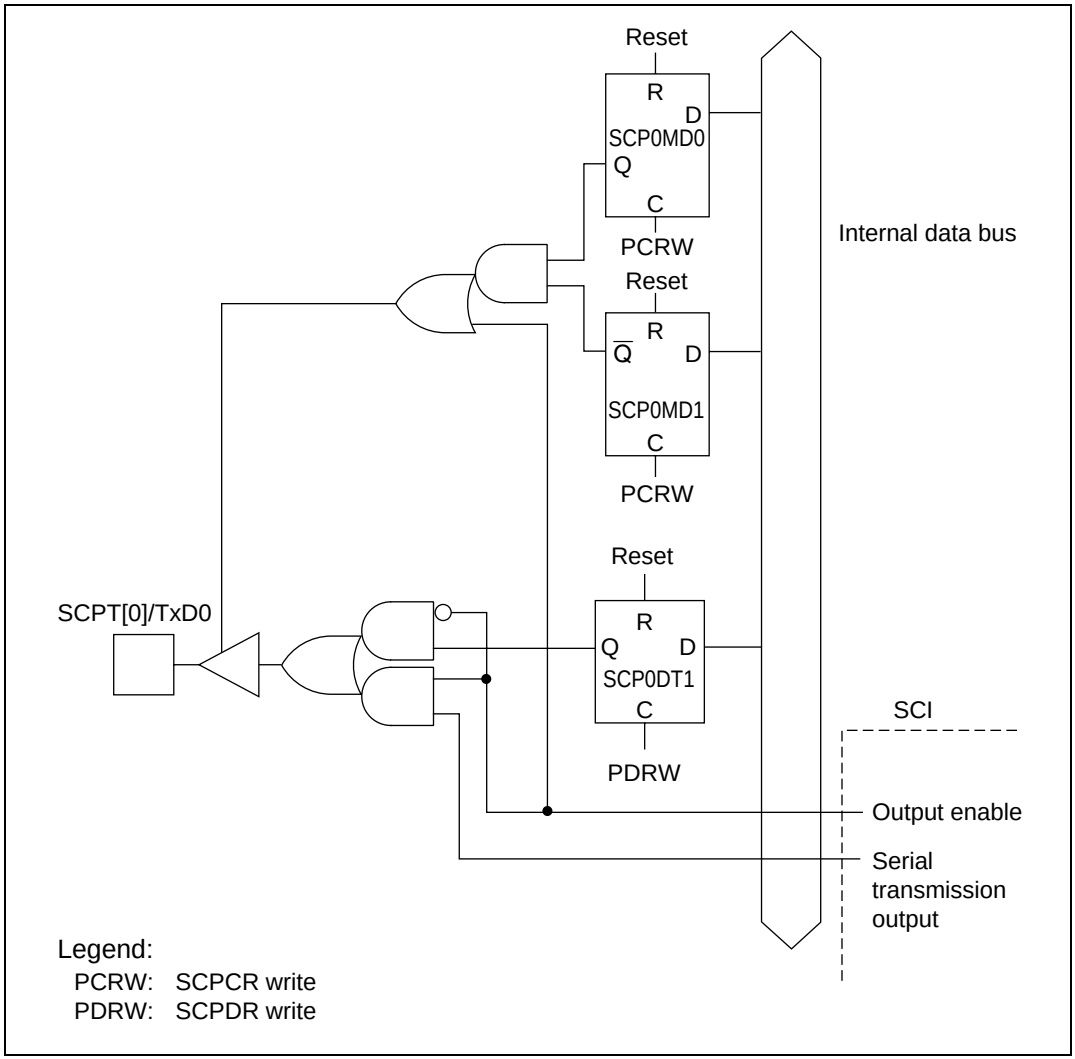


Figure 17.3 SCPT[0]/TxD0 Pin

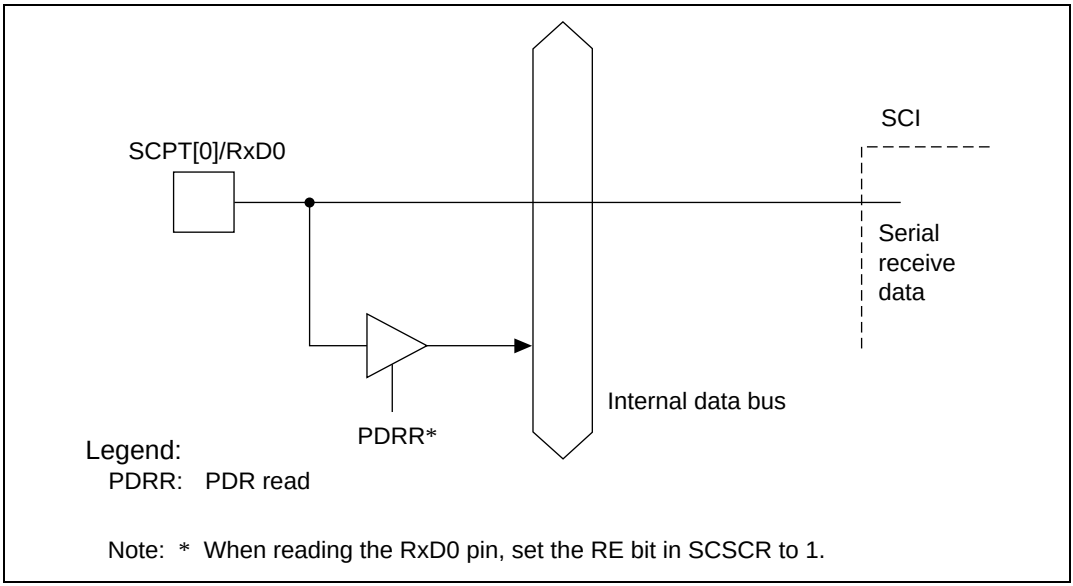


Figure 17.4 SCPT[0]/RxD0 Pin

17.1.3 Pin Configuration

The SCI has the serial pins summarized in table 17.1.

Table 17.1 SCI Pins

Pin Name	Abbreviation	I/O	Function
Serial clock pin	SCK0	I/O	Clock I/O
Receive data pin	RxD0	Input	Receive data input
Transmit data pin	TxD0	Output	Transmit data output

Note: They are made to function as serial pins by performing SCI operation settings with the TE, RE, CKEI, and CKEO bits in SCSCR and the C/$\overline{A}$ bit in SCSMR. Break state transmission and detection can be performed by means of the SCI's SCSPTR register.

17.1.4 Register Configuration

Table 17.2 summarizes the SCI internal registers. These registers select the communication mode (asynchronous or clock synchronous), specify the data format and bit rate, and control the transmitter and receiver sections.

Table 17.2 Registers

Name	Abbreviation	R/W	Initial Value	Address	Access size
Serial mode register	SCSMR	R/W	H'00	H'FFFFFE80	8
Bit rate register	SCBRR	R/W	H'FF	H'FFFFFE82	8
Serial control register	SCSCR	R/W	H'00	H'FFFFFE84	8
Transmit data register	SCTDR	R/W	H'FF	H'FFFFFE86	8
Serial status register	SCSSR	R/(W)*1	H'84	H'FFFFFE88	8
Receive data register	SCRDR	R	H'00	H'FFFFFE8A	8
Port SC data register	SCPDR	R/W	H'00	H'04000136 (H'A4000136)*2	8
Port SC control register	SCPCR	R/W	H'8008	H'04000116 (H'A4000116)*2	16

Notes: Registers with addresses beginning at H'04 are located in area 1 of physical space. Consequently, when the cache is on, either access these registers from the P2 area of logical space or else make an appropriate setting using the MMU so that these registers are not cached.

1. The only value that can be written is 0 to clear the flags.
2. When address translation by the MMU is not executed, the address in parentheses should be used.

17.2 Register Descriptions

17.2.1 Receive Shift Register (SCRSR)

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

The receive shift register (SCRSR) receives serial data.

Data input at the RxD0 pin is loaded into the SCRSR in the order received, LSB (bit 0) first, converting the data to parallel form. When one byte has been received, it is automatically transferred to the SCRDR.

The CPU cannot read or write the SCRSR directly.

17.2.2 Receive Data Register (SCRDR)

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

The receive data register (SCRDR) stores serial receive data.

The SCI completes the reception of one byte of serial data by moving the received data from the receive shift register (SCRSR) into the SCRDR for storage. The SCRSR is then ready to receive the next data.

This double buffering allows the SCI to receive data continuously.

The CPU can read but not write to SCRDR. SCRDR is initialized to H'00 by a reset and in standby or module standby mode.

17.2.3 Transmit Shift Register (SCTSR)

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

The transmit shift register (SCTSR) transmits serial data.

The SCI loads transmit data from the transmit data register (SCTDR) into the SCTSR, then transmits the data serially from the TxD0 pin, LSB (bit 0) first.

After transmitting one-byte data, the SCI automatically loads the next transmit data from the SCTDR into the SCTSR and starts transmitting again. If the TDRE bit of the SCSSR is 1, however, the SCI does not load the SCTDR contents into the SCTSR.

The CPU cannot read or write the SCTSR directly.

17.2.4 Transmit Data Register (SCTDR)

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

The transmit data register (SCTDR) is an eight-bit register that stores data for serial transmission.

When the SCI detects that the transmit shift register (SCTSR) is empty, it moves transmit data written in the SCTDR into the SCTSR and starts serial transmission. Continuous serial transmission is possible by writing the next transmit data in the SCTDR during serial transmission from the SCTSR.

The CPU can always read and write the SCTDR. The SCTDR is initialized to H'FF by a reset or in standby and module standby modes.

17.2.5 Serial Mode Register (SCSMR)

Bit:	7	6	5	4	3	2	1	0
	C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	MP	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

The serial mode register (SCSMR) is an eight-bit register that specifies the SCI serial communication format and selects the clock source for the baud rate generator.

The CPU can always read and write the SCSMR.

The SCSMR is initialized to H'00 by a reset or in standby and module standby modes.

Bit 7—Communication Mode (C/$\overline{A}$): Selects whether the SCI operates in the asynchronous or clock synchronous mode.

Bit 7: C/$\overline{A}$	Description	
0	Asynchronous mode	(Initial value)
1	Clock synchronous mode	

Bit 6—Character Length (CHR): Selects seven-bit or eight-bit data in the asynchronous mode. In the clock synchronous mode, the data length is always eight bits, regardless of the CHR setting.

Bit 6: CHR	Description	
0	Eight-bit data	(Initial value)
1	Seven-bit data*	

Note: * When seven-bit data is selected, the MSB (bit 7) of the transmit data register is not transmitted.

Bit 5—Parity Enable (PE): Selects whether to add a parity bit to transmit data and to check the parity of receive data, in the asynchronous mode. In the clock synchronous mode, a parity bit is neither added nor checked, regardless of the PE setting.

Bit 5: PE	Description	
0	Parity bit not added or checked	(Initial value)
1	Parity bit added and checked*	

Note: * When PE is set to 1, an even or odd parity bit is added to transmit data, depending on the parity mode (O/$\overline{E}$) setting. Receive data parity is checked according to the even/odd (O/$\overline{E}$) mode setting.

Bit 4—Parity Mode (O/$\overline{E}$): Selects even or odd parity when parity bits are added and checked. The O/$\overline{E}$ setting is used only in asynchronous mode and only when the parity enable bit (PE) is set to 1 to enable parity addition and check. The O/$\overline{E}$ setting is ignored in the clock synchronous mode, or in the asynchronous mode when parity addition and check is disabled.

Bit 4: O/$\overline{E}$	Description	
0	Even parity*1	(Initial value)
1	Odd parity*2	

Notes: 1. If even parity is selected, the parity bit is added to transmit data to make an even number of 1s in the transmitted character and parity bit combined. Receive data is checked to see if it has an even number of 1s in the received character and parity bit combined.
2. If odd parity is selected, the parity bit is added to transmit data to make an odd number of 1s in the transmitted character and parity bit combined. Receive data is checked to see if it has an odd number of 1s in the received character and parity bit combined.

Bit 3—Stop Bit Length (STOP): Selects one or two bits as the stop bit length in the asynchronous mode. This setting is used only in the asynchronous mode. It is ignored in the clock synchronous mode because no stop bits are added.

In receiving, only the first stop bit is checked, regardless of the STOP bit setting. If the second stop bit is 1, it is treated as a stop bit, but if the second stop bit is 0, it is treated as the start bit of the next incoming character.

Bit 3: STOP	Description	
0	One stop bit *1	(Initial value)
1	Two stop bits*2	

Notes: 1. In transmitting, a single bit of 1 is added at the end of each transmitted character.
2. In transmitting, two bits of 1 are added at the end of each transmitted character.

Bit 2—Multiprocessor Mode (MP): Selects multiprocessor format. When multiprocessor format is selected, settings of the parity enable (PE) and parity mode (O/$\overline{E}$) bits are ignored. The MP bit setting is used only in the asynchronous mode; it is ignored in the clock synchronous mode. For the multiprocessor communication function, see section 17.3.3, Multiprocessor Communication.

Bit 2: MP	Description	
0	Multiprocessor function disabled	(Initial value)
1	Multiprocessor format selected	

Bits 1 and 0—Clock Select 1 and 0 (CKS1 and CKS0): These bits select the internal clock source of the on-chip baud rate generator. Four clock sources are available. Pϕ, Pϕ/4, Pϕ/16 and Pϕ/64. For further information on the clock source, bit rate register settings, and baud rate, see section 17.2.9, Bit Rate Register (SCBRR).

Bit 1: CKS1	Bit 0: CKS0	Description	
0	0	Pϕ	(Initial value)
	1	Pϕ/4	
1	0	Pϕ/16	
	1	Pϕ/64	

Note: Pϕ: Peripheral clock

17.2.6 Serial Control Register (SCSCR)

Bit:	7	6	5	4	3	2	1	0
	TIE	RIE	TE	RE	MPIE	TEIE	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

The serial control register (SCSCR) operates the SCI transmitter/receiver, selects the serial clock output in the asynchronous mode, enables/disables interrupt requests, and selects the transmit/receive clock source. The CPU can always read and write the SCSCR. The SCSCR is initialized to H'00 by a reset or in standby and module standby modes.

Bit 7—Transmit Interrupt Enable (TIE): Enables or disables the transmit-data-empty interrupt (TXI) requested when the transmit data register empty bit (TDRE) in the serial status register (SCSSR) is set to 1 due to transfer of serial transmit data from the SCTDR to the SCTSR.

Bit 7: TIE	Description	
0	Transmit-data-empty interrupt request (TXI) is disabled*	(Initial value)
1	Transmit-data-empty interrupt request (TXI) is enabled	

Note: * The TXI interrupt request can be cleared by reading TDRE after it has been set to 1, then clearing TDRE to 0, or by clearing TIE to 0.

Bit 6—Receive Interrupt Enable (RIE): Enables or disables the receive-data-full interrupt (RXI) requested when the receive data register full bit (RDRF) in the serial status register (SCSSR) is set to 1 due to transfer of serial receive data from the SCRSR to the SCRDR. It also enables or disables receive-error interrupt (ERI) requests.

Bit 6: RIE	Description	
0	Receive-data-full interrupt (RXI) and receive-error interrupt (ERI) requests are disabled*	(Initial value)
1	Receive-data-full interrupt (RXI) and receive-error interrupt (ERI) requests are enabled	

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 or disables the SCI serial transmitter.

Bit 5: TE	Description	
0	Transmitter disabled*1	(Initial value).
1	Transmitter enabled*2	

Notes: 1. The transmit data register empty bit (TDRE) in the serial status register (SCSSR) is fixed to 1.
2. Serial transmission starts when the transmit data register empty (TDRE) bit in the serial status register (SCSSR) is cleared to 0 after writing of transmit data into the SCTDR. Select the transmit format in the SCSMR before setting TE to 1.

Bit 4—Receive Enable (RE): Enables or disables the SCI serial receiver.

Bit 4: RE	Description	
0	Receiver disabled*1	(Initial value)
1	Receiver enabled*2	

Notes: 1. Clearing RE to 0 does not affect the receive flags (RDRF, FER, PER, ORER). These flags retain their previous values.
2. Serial reception starts when a start bit is detected in the asynchronous mode, or synchronous clock input is detected in the clock synchronous mode. Select the receive format in the SCSMR before setting RE to 1.

Bit 3—Multiprocessor Interrupt Enable (MPIE): Enables or disables multiprocessor interrupts. The MPIE setting is used only in the asynchronous mode, and only if the multiprocessor mode bit (MP) in the serial mode register (SCSMR) is set to 1 during reception. The MPIE setting is ignored in the clock synchronous mode or when the MP bit is cleared to 0.

Bit 3: MPIE	Description	
0	Multiprocessor interrupts are disabled (normal receive operation) [Clear conditions] 1. When MPIE is cleared to 0 2. When the multiprocessor bit (MPB) is set to 1 in receive data	(Initial value)
1	Multiprocessor interrupts are enabled* Receive-data-full interrupt requests (RXI), receive-error interrupt requests (ERI), and setting of the RDRF, FER, and ORER status flags in the serial status register (SCSSR) are disabled until data with a multiprocessor bit of 1 is received.	

Note: * The SCI does not transfer receive data from the SCRSR to the SCRDR, does not detect receive errors, and does not set the RDRF, FER, and ORER flags in the serial status register (SCSSR). When it receives data that includes MPB = 1, the SCSSR's MPB flag is set to 1, and the SCI automatically clears MPIE to 0, generates RXI and ERI interrupts (if the TIE and RIE bits in the SCSCR are set to 1), and allows the FER and ORER bits to be set.

Bit 2—Transmit-End Interrupt Enable (TEIE): Enables or disables the transmit-end interrupt (TEI) requested if SCTDR does not contain new transmit data when the MSB is transmitted.

Bit 2: TEIE	Description	
0	Transmit-end interrupt (TEI) requests are disabled*	(Initial value)
1	Transmit-end interrupt (TEI) requests are enabled*	

Note: * The TEI request can be cleared by reading the TDRE bit in the serial status register (SCSSR) after it has been set to 1, then clearing TDRE to 0 and clearing the transmit end (TEND) bit to 0, or by clearing the TEIE bit to 0.

Bits 1 and 0—Clock Enable 1 and 0 (CKE1 and CKE0): These bits select the SCI clock source and enable or disable clock output from the SCK0 pin. Depending on the combination of CKE1 and CKE0, the SCK0 pin can be used for serial clock output or serial clock input.

The CKE0 setting is valid only when the asynchronous mode and the internal clock are selected (CKE1 = 0). The CKE0 setting is ignored in the clock synchronous mode, or when an external clock source is selected (CKE1 = 1). Before selecting the SCI operating mode in the serial mode register (SCSMR), set CKE1 and CKE0. For further details on selection of the SCI clock source, see table 17.10 in section 17.3, Operation.

Bit 1: CKE1	Bit 0: CKE0	Description	
0	0	Asynchronous mode	Internal clock, SCK pin used for input pin (input signal is ignored)*1
		Clock synchronous mode	Internal clock, SCK pin used for synchronous clock output*1
	1	Asynchronous mode	Internal clock, SCK pin used for clock output*2
		Clock synchronous mode	Internal clock, SCK pin used for synchronous clock output
1	0	Asynchronous mode	External clock, SCK pin used for clock input*3
		Clock synchronous mode	External clock, SCK pin used for synchronous clock input
	1	Asynchronous mode	External clock, SCK pin used for clock input*3
		Clock synchronous mode	External clock, SCK pin used for synchronous clock input

Notes: 1. Initial value
2. The output clock frequency is the same as the bit rate.
3. The input clock frequency is 16 times the bit rate.

17.2.7 Serial Status Register (SCSSR)

Bit:	7	6	5	4	3	2	1	0
	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT
Initial value:	1	0	0	0	0	1	0	0
R/W:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R	R	R/W

Note: * The only value that can be written is a 0 to clear the flag.

The serial status register (SCSSR) is an 8-bit register containing multiprocessor bit values, and status flags that indicate SCI operating state.

The CPU can always read and write the SCSSR, but cannot write 1 in the status flags (TDRE, RDRF, ORER, PER, and FER). These flags can be cleared to 0 only if they have first been read (after being set to 1). Bits 2 (TEND) and 1 (MPB) are read-only bits that cannot be written.

The SCSSR is initialized to H'84 by a reset or in standby and module standby modes.

Bit 7—Transmit Data Register Empty (TDRE): Indicates that the SCI has loaded transmit data from the SCTDR into the SCTSR and new serial transmit data can be written in the SCTDR.

Bit 7: TDRE	Description
0	SCTDR contains valid transmit data [Clear condition] When software reads TDRE after it has been set to 1, then writes 0 in TDRE or data is written in SCTDR.
1	SCTDR does not contain valid transmit data (Initial value) [Setting conditions] 1. When the chip is reset or enters standby mode 2. When the TE bit in the serial control register (SCSCR) is cleared to 0 3. When SCTDR contents are loaded into SCTSR, so new data can be written in SCTDR.

Bit 6—Receive Data Register Full (RDRF): Indicates that SCRDR contains received data.

Bit 6: RDRF	Description
0	SCRDR does not contain valid received data (Initial value) [Clear conditions] 1. When the chip is reset or enters standby mode 2. When software reads RDRF after it has been set to 1, then writes 0 in RDRF.
1	SCRDR contains valid received data [Setting condition] When serial data is received normally and transferred from SCRSR to SCRDR.

Note: The SCRDR and RDRF are not affected by detection of receive errors or by clearing of the RE bit to 0 in the serial control register. They retain their previous contents.
If RDRF is still set to 1 when reception of the next data ends, an overrun error (ORER) occurs and the received data is lost.

Bit 5—Overrun Error (ORER): Indicates that data reception aborted due to an overrun error.

Bit 5: ORER	Description
0	Receiving is in progress or has ended normally*1 (Initial value) [Clear conditions] 1. When the chip is reset or enters standby mode 2. When ORER=1 is read and then 0 is written to ORER.
1	A receive overrun error occurred*2 [Setting condition] When reception of the next serial data ends when RDRF is set to 1.

Notes: 1. Clearing the RE bit to 0 in the serial control register does not affect the ORER bit, which retains its previous value.
2. SCRDR continues to hold the data received before the overrun error, so subsequent receive data is lost. Serial receiving cannot continue while ORER is set to 1. In the clock synchronous mode, serial transmitting is also disabled.

Bit 4—Framing Error (FER): Indicates that data reception aborted due to a framing error in the asynchronous mode.

Bit 4: FER	Description
0	Receiving is in progress or has ended normally*1 (Initial value) [Clear conditions] 1. When the chip is reset or enters standby mode 2. When FER=1 is read and then 0 is written to FER.
1	A receive framing error occurred FER is set to 1 if the stop bit at the end of receive data is checked and found to be 0.*2

Notes: 1. Clearing the RE bit to 0 in the serial control register does not affect the FER bit, which retains its previous value.
2. When the stop bit length is two bits, only the first bit is checked. The second stop bit is not checked. When a framing error occurs, the SCI transfers the receive data into the SCRDR but does not set RDRF. Serial receiving cannot continue while FER is set to 1. In the clock synchronous mode, serial transmitting is also disabled.

Bit 3—Parity Error (PER): Indicates that data reception (with parity) aborted due to a parity error in the asynchronous mode.

Bit 3: PER	Description
0	Receiving is in progress or has ended normally*1 (Initial value) [Clear conditions] 1. When the chip is reset or enters standby mode 2. When PER=1 is read and then 0 is written to PER.
1	A receive parity error occurred*2 [Setting condition] When the number of 1s in receive data, including the parity bit, does not match the even or odd parity setting of the parity mode bit (O/$\overline{E}$) in the serial mode register (SCSMR).

Notes: 1. Clearing the RE bit to 0 in the serial control register does not affect the PER bit, which retains its previous value.
2. When a parity error occurs, the SCI transfers the receive data into the SCRDR but does not set RDRF. Serial receiving cannot continue while PER is set to 1. In the clock synchronous mode, serial transmitting is also disabled.

Bit 2—Transmit End (TEND): Indicates that when the last bit of a serial character was transmitted, the SCTDR did not contain valid data, so transmission has ended. TEND is a read-only bit and cannot be written.

Bit 2: TEND	Description	
0	Transmission is in progress [Clear condition] When TDRE=1 is read and then 0 is written to TDRE.	
1	End of transmission [Setting conditions] 1. When the chip is reset or enters standby mode 2. When TE is cleared to 0 in the serial control register (SCSCR) 3. If TDRE is 1 when the last bit of a one-byte serial character is transmitted.	(Initial value)

Bit 1—Multiprocessor Bit (MPB): Stores the value of the multiprocessor bit in receive data when a multiprocessor format is selected for receiving in the asynchronous mode. The MPB is a read-only bit and cannot be written.

Bit 1: MPB	Description	
0	Multiprocessor bit value in receive data is 0*	(Initial value)
1	Multiprocessor bit value in receive data is 1	

Note: * If RE is cleared to 0 when a multiprocessor format is selected, the MPB retains its previous value.

Bit 0—Multiprocessor Bit Transfer (MPBT): Stores the value of the multiprocessor bit added to transmit data when a multiprocessor format is selected for transmitting in the asynchronous mode. The MPBT setting is ignored in the clock synchronous mode, when a multiprocessor format is not selected, or when the SCI is not transmitting.

Bit 0: MPBT	Description	
0	Multiprocessor bit value in transmit data is 0	(Initial value)
1	Multiprocessor bit value in transmit data is 1	

17.2.8 Port SC Control Register (SCPCR)/Port SC Data Register (SCPDR)

The port SC control register (SCPCR) and port SC data register (SCPDR) control I/O and data for the port multiplexed with the serial communication interface (SCI) pins.

SCPCR settings are used to perform I/O control, to enable data written in SCPDR to be output to the TxD0 pin, and input data to be read from the RxD0 pin, and to control serial transmission/reception breaks.

It is also possible to read data on the SCK0 pin, and write output data.

SCPCR

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SCP7 MD1	SCP7 MD0	SCP6 MD1	SCP6 MD0	SCP5 MD1	SCP5 MD0	SCP4 MD1	SCP4 MD0	SCP3 MD1	SCP3 MD0	SCP2 MD1	SCP2 MD0	SCP1 MD1	SCP1 MD0	SCP0 MD1	SCP0 MD0
Initial value:	1	0	0	0	0	0	0	0	0	0	0	0	1	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

SCPDR

Bit:	7	6	5	4	3	2	1	0
	SCP7DT	SCP6DT	SCP5DT	SCP4DT	SCP3DT	SCP2DT	SCP1DT	SCP0DT
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

SCI pin I/O and data control are performed by bits 3 to 0 of SCPCR and bits 1 and 0 of SCPDR.

SCPCR Bits 3 and 2—Serial Clock Port I/O (SCP1MD1, SCP1MD0): These bits specify serial port SCK0 pin I/O. When the SCK0 pin is actually used as a port I/O pin, clear the C/$\overline{A}$ bit of SCSMR and bits CKE1 and CKE0 of SCSCR to 0.

Bit 3: SCP1MD1	Bit 2: SCP1MD0	Description
0	0	SCP1DT bit value is not output to SCK0 pin
	1	SCP1DT bit value is output to SCK0 pin
1	0	SCK0 pin value is read from SCP1DT bit
	1	(Initial values: 1 and 0)

SCPDR Bit 1—Serial Clock Port Data (SCP1DT): Specifies the serial port SCK0 pin I/O data. Input or output is specified by the SCP1MD0 and SCP1MD1 bits. In output mode, the value of the SCP1DT bit is output from the SCK0 pin.

Bit 1: SCP1DT	Description	
0	I/O data is low	(Initial value)
1	I/O data is high	

SCPCR Bits 1 and 0—Serial Port Break I/O (SCP0MD1, SCP0MD0): These bits specify the serial port TxD0 pin output condition. When the TxD0 pin is actually used as a port output pin and outputs the value set with the SCP0DT bit, clear the TE bit of SCSCR to 0.

Bit 1: SCP0MD1	Bit 0: SCP0MD0	Description	
0	0	SCP0DT bit value is not output to TxD0 pin	(Initial value)
0	1	SCP0DT bit value is output to TxD0 pin	

SCPDR Bit 0—Serial Port Break Data (SCP0DT): Specifies the serial port RxD0 pin input data and TxD0 pin output data. The TxD0 pin output condition is specified by the SCP0MD0 and SCP0MD1 bits. When the TxD0 pin is set to output mode, the value of the SCP0DT bit is output to the TxD0 pin. The RxD0 pin value is read from the SCP0DT bit regardless of the values of the SCP0MD0 and SCP0MD1 bits, if RE in the SCSCR is set to 1. The initial value of this bit after a power-on reset is undefined.

Bit 0: SCP0DT	Description	
0	I/O data is low	(Initial value)
1	I/O data is high	

Block diagrams of the SCI I/O ports are shown in figures 17.2 to 17.4.

17.2.9 Bit Rate Register (SCBRR)

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

The bit rate register (SCBRR) is an eight-bit register that, together with the baud rate generator clock source selected by the CKS1 and CKS0 bits in the serial mode register (SCSMR), determines the serial transmit/receive bit rate.

The CPU can always read and write the SCBRR.

The SCBRR is initialized to H'FF by a reset or in module standby or standby mode. Each channel has independent baud rate generator control, so different values can be set in two channels.

The SCBRR setting is calculated as follows:

$$\text{Asynchronous mode: } N = \frac{P\phi}{64 \times 2^{2n-1} \times B} \times 10^6 - 1$$

$$\text{Clock synchronous mode: } N = \frac{P\phi}{8 \times 2^{2n-1} \times B} \times 10^6 - 1$$

B: Bit rate (bit/s)
N: SCBRR setting for baud rate generator ($0 \le N \le 255$)
Pϕ: Operating frequency for peripheral modules (MHz)
n: Baud rate generator clock source (n = 0, 1, 2, 3) (for the clock sources and values of n, see table 17.3.)

Table 17.3 SCSMR Settings

		SCSMR Settings	
n	**Clock Source**	**CKS1**	**CKS0**
0	Pϕ	0	0
1	Pϕ/4	0	1
2	Pϕ/16	1	0
3	Pϕ/64	1	1

Note: Find the bit rate error for the asynchronous mode by the following formula:

$$\text{Error } (\%) = \left\{ \frac{P\phi \times 10^6}{(N + 1) \times B \times 64 \times 2^{2n-1}} - 1 \right\} \times 100$$

Table 17.4 lists examples of SCBRR settings in the asynchronous mode; table 17.5 lists examples of SCBRR settings in the clock synchronous mode.

Table 17.4 Bit Rates and SCBRR Settings in Asynchronous Mode (1)

Bit Rate (bits/s)	Pϕ (MHz)								
	2			2.097152			2.4576		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	1	141	0.03	1	148	–0.04	1	174	–0.26
150	1	103	0.16	1	108	0.21	1	127	0.00
300	0	207	0.16	0	217	0.21	0	255	0.00
600	0	103	0.16	0	108	0.21	0	127	0.00
1200	0	51	0.16	0	54	–0.70	0	63	0.00
2400	0	25	0.16	0	26	1.14	0	31	0.00
4800	0	12	0.16	0	13	–2.48	0	15	0.00
9600	0	6	–6.99	0	6	–2.48	0	7	0.00
19200	0	2	8.51	0	2	13.78	0	3	0.00
31250	0	1	0.00	0	1	4.86	0	1	22.88
38400	0	1	–18.62	0	1	–14.67	0	1	0.00

Table 17.4 Bit Rates and SCBRR Settings in Asynchronous Mode (2)

Bit Rate (bits/s)	Pϕ (MHz)								
	3			3.6864			4		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	1	212	0.03	2	64	0.70	2	70	0.03
150	1	155	0.16	1	191	0.00	1	207	0.16
300	1	77	0.16	1	95	0.00	1	103	0.16
600	0	155	0.16	0	191	0.00	0	207	0.16
1200	0	77	0.16	0	95	0.00	0	103	0.16
2400	0	38	0.16	0	47	0.00	0	51	0.16
4800	0	19	–2.34	0	23	0.00	0	25	0.16
9600	0	9	–2.34	0	11	0.00	0	12	0.16
19200	0	4	–2.34	0	5	0.00	0	6	–6.99
31250	0	2	0.00	—	—	—	0	3	0.00
38400	—	—	—	0	2	0.00	0	2	8.51

Table 17.4 Bit Rates and SCBRR Settings in Asynchronous Mode (3)

Bit Rate (bits/s)	Pϕ (MHz)								
	4.9152			5			6		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	86	0.31	2	88	–0.25	2	106	–0.44
150	1	255	0.00	2	64	0.16	2	77	0.16
300	1	127	0.00	1	129	0.16	1	155	0.16
600	0	255	0.00	1	64	0.16	1	77	0.16
1200	0	127	0.00	0	129	0.16	0	155	0.16
2400	0	63	0.00	0	64	0.16	0	77	0.16
4800	0	31	0.00	0	32	–1.36	0	38	0.16
9600	0	15	0.00	0	15	1.73	0	19	–2.34
19200	0	7	0.00	0	7	1.73	0	9	–2.34
31250	0	4	–1.70	0	4	0.00	0	5	0.00
38400	0	3	0.00	0	3	1.73	0	4	–2.34

Table 17.4 Bit Rates and SCBRR Settings in Asynchronous Mode (4)

Bit Rate (bits/s)	Pϕ (MHz)								
	6.144			7.3728			8		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	108	0.08	2	130	–0.07	2	141	0.03
150	2	79	0.00	2	95	0.00	2	103	0.16
300	1	159	0.00	1	191	0.00	1	207	0.16
600	1	79	0.00	1	95	0.00	1	103	0.16
1200	0	159	0.00	0	191	0.00	0	207	0.16
2400	0	79	0.00	0	95	0.00	0	103	0.16
4800	0	39	0.00	0	47	0.00	0	51	0.16
9600	0	19	0.00	0	23	0.00	0	25	0.16
19200	0	9	0.00	0	11	0.00	0	12	0.16
31250	0	5	2.40	0	6	5.33	0	7	0.00
38400	0	4	0.00	0	5	0.00	0	6	–6.99

Table 17.4 Bit Rates and SCBRR Settings in Asynchronous Mode (5)

	Pϕ (MHz)											
	9.8304			10			12			12.288		
Bit Rate (bits/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	174	–0.26	2	177	–0.25	2	212	0.03	2	217	0.08
150	2	127	0.00	2	129	0.16	2	155	0.16	2	159	0.00
300	1	255	0.00	2	64	0.16	2	77	0.16	2	79	0.00
600	1	127	0.00	1	129	0.16	1	155	0.16	1	159	0.00
1200	0	255	0.00	1	64	0.16	1	77	0.16	1	79	0.00
2400	0	127	0.00	0	129	0.16	0	155	0.16	0	159	0.00
4800	0	63	0.00	0	64	0.16	0	77	0.16	0	79	0.00
9600	0	31	0.00	0	32	–1.36	0	38	0.16	0	39	0.00
19200	0	15	0.00	0	15	1.73	0	19	0.16	0	19	0.00
31250	0	9	–1.70	0	9	0.00	0	11	0.00	0	11	2.40
38400	0	7	0.00	0	7	1.73	0	9	–2.34	0	9	0.00

Table 17.4 Bit Rates and SCBRR Settings in Asynchronous Mode (6)

	Pϕ (MHz)											
	14.7456			16			19.6608			20		
Bit Rate (bits/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	3	64	0.70	3	70	0.03	3	86	0.31	3	88	–0.25
150	2	191	0.00	2	207	0.16	2	255	0.00	3	64	0.16
300	2	95	0.00	2	103	0.16	2	127	0.00	2	129	0.16
600	1	191	0.00	1	207	0.16	1	255	0.00	2	64	0.16
1200	1	95	0.00	1	103	0.16	1	127	0.00	1	129	0.16
2400	0	191	0.00	0	207	0.16	0	255	0.00	1	64	0.16
4800	0	95	0.00	0	103	0.16	0	127	0.00	0	129	0.16
9600	0	47	0.00	0	51	0.16	0	63	0.00	0	64	0.16
19200	0	23	0.00	0	25	0.16	0	31	0.00	0	32	–1.36
31250	0	14	–1.70	0	15	0.00	0	19	–1.70	0	19	0.00
38400	0	11	0.00	0	12	0.16	0	15	0.00	0	15	1.73

Table 17.4 Bit Rates and SCBRR Settings in Asynchronous Mode (7)

	Pϕ (MHz)											
	24			24.576			28.7			30		
Bit Rate (bits/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	3	106	–0.44	3	108	0.08	3	126	0.31	3	132	0.13
150	3	77	0.16	3	79	0.00	3	92	0.46	3	97	–0.35
300	2	155	0.16	2	159	0.00	2	186	–0.08	2	194	0.16
600	2	77	0.16	2	79	0.00	2	92	0.46	2	97	–0.35
1200	1	155	0.16	1	159	0.00	1	186	–0.08	1	194	0.16
2400	1	77	0.16	1	79	0.00	1	92	0.46	1	97	–0.35
4800	0	155	0.16	0	159	0.00	0	186	–0.08	0	194	–1.36
9600	0	77	0.16	0	79	0.00	0	92	0.46	0	97	–0.35
19200	0	38	0.16	0	39	0.00	0	46	–0.61	0	48	–0.35
31250	0	23	0.00	0	24	–1.70	0	28	–1.03	0	29	0.00
38400	0	19	–2.34	0	19	0.00	0	22	1.55	0	23	1.73

Table 17.5 Bit Rates and SCBRR Settings in Clock Synchronous Mode

Bit Rate (bits/s)	Pϕ (MHz)									
	4		8		16		28.7		30	
	n	N	n	N	n	N	n	N	n	N
110	—	—	—	—	—	—	—	—	—	—
250	2	249	3	124	3	249	—	—	—	—
500	2	124	2	249	3	124	3	223	3	233
1k	1	249	2	124	2	249	3	111	3	116
2.5k	1	99	1	199	2	99	2	178	2	187
5k	0	199	1	99	1	199	2	89	2	93
10k	0	99	0	199	1	99	1	178	1	187
25k	0	39	0	79	0	159	1	71	1	74
50k	0	19	0	39	0	79	0	143	0	149
100k	0	9	0	19	0	39	0	71	0	74
250k	0	3	0	7	0	15	—	—	0	29
500k	0	1	0	3	0	7	—	—	0	14
1M	0	0*	0	1	0	3	—	—	—	—
2M			0	0*	0	1	—	—	—	—

Notes: Settings with an error of 1% or less are recommended.

Blank: No setting possible

—: Setting possible, but error occurs. (Refer to section 17.2.9, Bit Rate Register (SCBRR))

*: Continuous transmit/receive not possible as transfer capability to the buffer becomes insufficient.

Table 17.6 indicates the maximum bit rates in the asynchronous mode when the baud rate generator is being used. Tables 17.7 and 17.8 list the maximum rates for external clock input.

Table 17.6 Maximum Bit Rates for Various Frequencies with Baud Rate Generator (Asynchronous Mode)

		Settings	
Pϕ (MHz)	**Maximum Bit Rate (bits/s)**	**n**	**N**
2	62500	0	0
2.097152	65536	0	0
2.4576	76800	0	0
3	93750	0	0
3.6864	115200	0	0
4	125000	0	0
4.9152	153600	0	0
8	250000	0	0
9.8304	307200	0	0
12	375000	0	0
14.7456	460800	0	0
16	500000	0	0
19.6608	614400	0	0
20	625000	0	0
24	750000	0	0
24.576	768000	0	0
28.7	896875	0	0
30	937500	0	0

Table 17.7 Maximum Bit Rates during External Clock Input (Asynchronous Mode)

Pϕ (MHz)	External Input Clock (MHz)	Maximum Bit Rate (bits/s)
2	0.5000	31250
2.097152	0.5243	32768
2.4576	0.6144	38400
3	0.7500	46875
3.6864	0.9216	57600
4	1.0000	62500
4.9152	1.2288	76800
8	2.0000	125000
9.8304	2.4576	153600
12	3.0000	187500
14.7456	3.6864	230400
16	4.0000	250000
19.6608	4.9152	307200
20	5.0000	312500
24	6.0000	375000
24.576	6.1440	384000
28.7	7.1750	448436
30	7.5000	468750

Table 17.8 Maximum Bit Rates during External Clock Input (Clock Synchronous Mode)

Pϕ (MHz)	External Input Clock (MHz)	Maximum Bit Rate (bits/s)
8	1.3333	1333333.3
16	2.6667	2666666.7
24	4.0000	4000000.0
28.7	4.7833	4783333.3
30	5.0000	5000000.0

17.3 Operation

17.3.1 Overview

For serial communication, the SCI has an asynchronous mode in which characters are synchronized individually, and a clock synchronous mode in which communication is synchronized with clock pulses.

Asynchronous/clock synchronous mode and the transmission format are selected in the serial mode register (SCSMR), as listed in table 17.9. The SCI clock source is selected by the combination of the C/$\overline{A}$ bit in the serial mode register (SCSMR) and the CKE1 and CKE0 bits in the serial control register (SCSCR), as listed in table 17.10.

Asynchronous Mode:

- Data length is selectable: seven or eight bits.
- Parity and multiprocessor bits are selectable. So is the stop bit length (one or two bits). The combination of the preceding selections constitutes the communication format and character length.
- In receiving, it is possible to detect framing errors, parity errors, overrun errors and breaks.
- An internal or external clock can be selected as the SCI clock source.
  - When an internal clock is selected, the SCI operates using the on-chip baud rate generator, and can output a serial clock signal with a frequency matching the bit rate.
  - When an external clock is selected, the external clock input must have a frequency 16 times the bit rate. (The on-chip baud rate generator is not used.)

Clock Synchronous Mode:

- The transmission/reception format has a fixed eight-bit data length.
- In receiving, it is possible to detect overrun errors.
- An internal or external clock can be selected as the SCI clock source.
  - When an internal clock is selected, the SCI operates using the on-chip baud rate generator, and outputs a synchronous clock signal to external devices.
  - When an external clock is selected, the SCI operates on the input synchronous clock. The on-chip baud rate generator is not used.

Table 17.9 Serial Mode Register Settings and SCI Communication Formats

SCSMR Settings						SCI Communication Format			
Bit 7 C/$\overline{A}$	Bit 6 CHR	Bit 5 PE	Bit 2 MP	Bit 3 STOP	Mode	Data Length	Parity Bit	Multipro-cessor Bit	Stop Bit Length
0	0	0	0	0	Asynchronous	8-bit	Not set	Not set	1 bit
				1					2 bits
		1		0			Set		1 bit
				1					2 bits
	1	0		0		7-bit	Not set		1 bit
				1					2 bits
		1		0			Set		1 bit
				1					2 bits
	0	*	1	0	Asynchronous (multiprocessor format)	8-bit	Not set	Set	1 bit
		*		1					2 bits
	1	*		0		7-bit			1 bit
		*		1					2 bits
1	*	*	*	*	Clock synchronous	8-bit		Not set	None

Note: * Don't care.

Table 17.10 SCSMR and SCSCR Settings and SCI Clock Source Selection

SCSMR	SCSCR Settings			SCI Transmit/Receive Clock	
Bit 7 C/$\overline{A}$	Bit 1 CKE1	Bit 0 CKE0	Mode	Clock Source	SCK0 Pin Function
0	0	0	Asynchronous mode	Internal	SCI does not use the SCK0 pin
		1			Outputs a clock with frequency matching the bit rate
	1	0		External	Inputs a clock with frequency 16 times the bit rate
		1			
1	0	0	Clock synchronous mode	Internal	Outputs the synchronous clock
		1			
	1	0		External	Inputs the synchronous clock
		1			

17.3.2 Operation in Asynchronous Mode

In the asynchronous mode, each transmitted or received character begins with a start bit and ends with a stop bit. Serial communication is synchronized one character at a time.

The transmitting and receiving sections of the SCI are independent, so full duplex communication is possible. The transmitter and receiver are both double buffered, so data can be written and read while transmitting and receiving are in progress, enabling continuous transmitting and receiving.

Figure 17.5 shows the general format of asynchronous serial communication. In asynchronous serial communication, the communication line is normally held in the mark (high) state. The SCI monitors the line and starts serial communication when the line goes to the space (low) state, indicating a start bit. One serial character consists of a start bit (low), data (LSB first), parity bit (high or low), and stop bit (high), in that order.

When receiving in the asynchronous mode, the SCI synchronizes on the falling edge of the start bit. The SCI samples each data bit on the eighth pulse of a clock with a frequency 16 times the bit rate. Receive data is latched at the center of each bit.

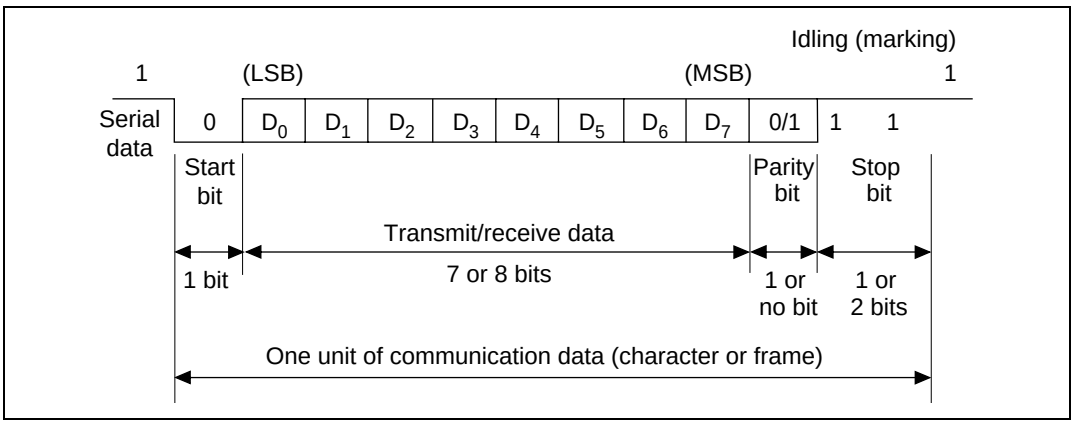


Figure 17.5 Data Format in Asynchronous Communication (Example with 8-Bit Data, Parity, Two Stop Bits)

Transmit/Receive Formats

Table 17.11 lists the 12 communication formats that can be selected in the asynchronous mode. The format is selected by settings in the serial mode register (SCSMR).

Table 17.11 Serial Communication Formats (Asynchronous Mode)

SCSMR Bits				Serial Transmit/Receive Format and Frame Length											
CHR	PE	MP	STOP	1	2	3	4	5	6	7	8	9	10	11	12
0	0	0	0	START	8-bit data								STOP		
0	0	0	1	START	8-bit data								STOP	STOP	
0	1	0	0	START	8-bit data								P	STOP	
0	1	0	1	START	8-bit data								P	STOP	STOP
1	0	0	0	START	7-bit data							STOP			
1	0	0	1	START	7-bit data							STOP	STOP		
1	1	0	0	START	7-bit data							P	STOP		
1	1	0	1	START	7-bit data							P	STOP	STOP	
0	—	1	0	START	8-bit data								MPB	STOP	
0	—	1	1	START	8-bit data								MPB	STOP	STOP
1	—	1	0	START	7-bit data							MPB	STOP		
1	—	1	1	START	7-bit data							MPB	STOP	STOP	

Notes: —: Don’t care bits
START: Start bit
STOP: Stop bit
P: Parity bit
MPB: Multiprocessor bit

Clock

An internal clock generated by the on-chip baud rate generator or an external clock input from the SCK0 pin can be selected as the SCI transmit/receive clock. The clock source is selected by the C/$\overline{A}$ bit in the serial mode register (SCSMR) and bits CKE1 and CKE0 in the serial control register (SCSCR) (table 17.10).

When an external clock is input at the SCK0 pin, it must have a frequency equal to 16 times the desired bit rate.

When the SCI operates on an internal clock, it can output a clock signal at the SCK0 pin. The frequency of this output clock is equal to the bit rate. The phase is aligned as in figure 17.6 so that the rising edge of the clock occurs at the center of each transmit data bit.

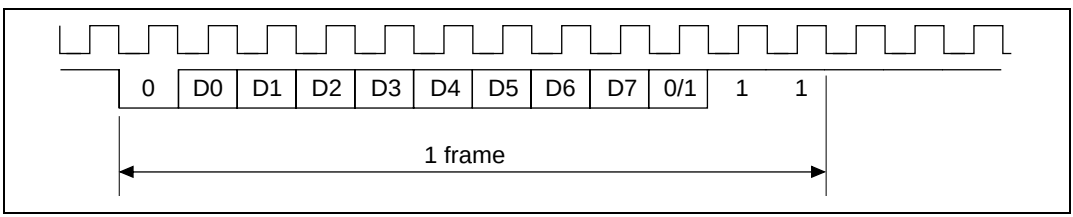


Figure 17.6 Relationship between Output Clock and Transfer Data Phase (Asynchronous Mode)

Transmitting and Receiving Data

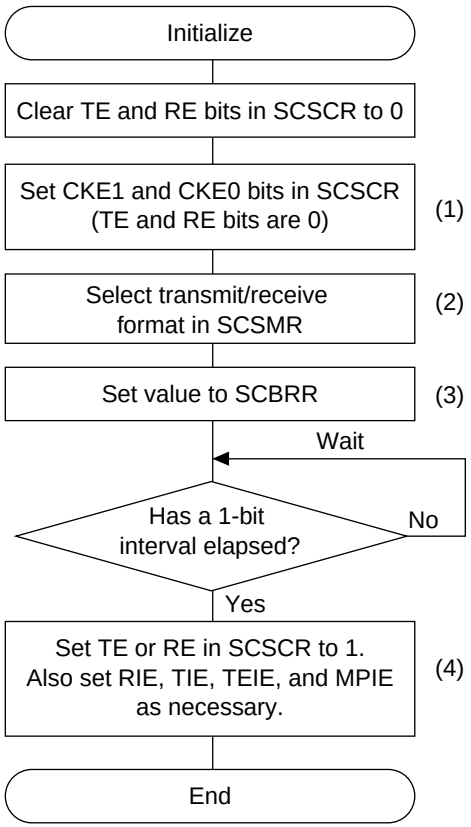
SCI Initialization (Asynchronous Mode):

Before transmitting or receiving, clear the TE and RE bits to 0 in the serial control register (SCSCR), then initialize the SCI as follows.

When changing the operation mode or communication format, always clear the TE and RE bits to 0 before following the procedure given below. Clearing TE to 0 sets TDRE to 1 and initializes the transmit shift register (SCTSR). Clearing RE to 0, however, does not initialize the RDRF, PER, FER, and ORER flags or receive data register (SCRDR), which retain their previous contents.

When an external clock is used, the clock should not be stopped during initialization or subsequent operation. SCI operation becomes unreliable if the clock is stopped.

Figure 17.7 is a sample flowchart for initializing the SCI. The procedure for initializing the SCI is:



(1) Select the clock source in the serial control register (SCSCR). Leave RIE, TIE, TEIE, MPIE, TE, and RE cleared to 0. If clock output is selected in asynchronous mode, clock output starts immediately after the setting is made to SCSCR.

(2) Select the communication format in the serial mode register (SCSMR).

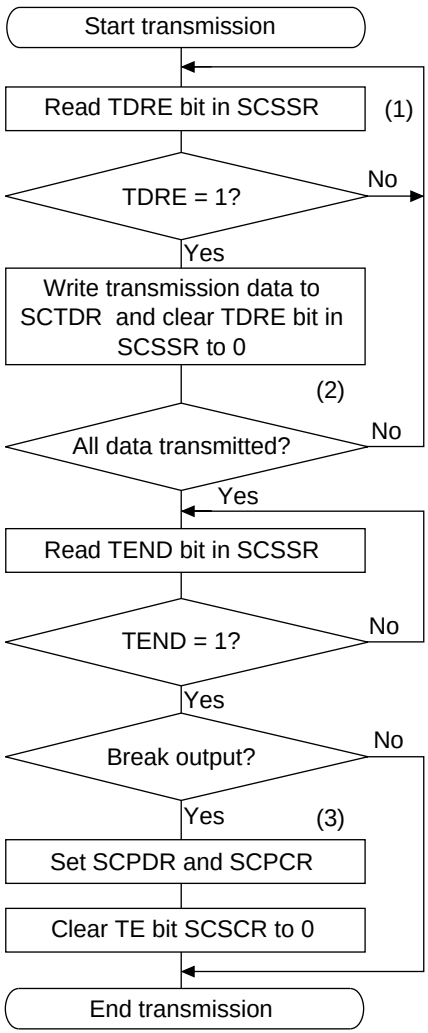
(3) Write the value corresponding to the bit rate in the bit rate register (SCBRR) unless an external clock is used.

(4) Wait at least one bit interval, then set TE or RE in the serial control register (SCSCR) to 1. Also set RIE, TIE, TEIE, and MPIE as necessary. Setting TE and RE enables the TxD and RxD pins to be used. The initial states are the mark transmit state, and the idle receive state (waiting for a start bit).

Figure 17.7 Sample SCI Initialization Flowchart

Serial Data Transmission (Asynchronous Mode):

Figure 17.8 shows a sample flow chart for serial data transmission. After enabling the SCI transmission, transmit serial data following the procedure shown below:



(1) SCI status check and transmit data write:
Read the serial status register (SCSSR), check that the TDRE bit is 1, then write transmit data in the transmit data register (SCTDR) and clear TDRE to 0.

(2) To continue transmitting serial data:
Read the TDRE bit to check whether it is safe to write (if it reads 1); if so, write data in SCTDR, then clear TDRE to 0.

(3) To output a break at the end of serial transmission:
Set the SC port data register (SCPDR) and SC port control register (SCPCR), then clear the TE bit to 0 in the serial control register (SCSCR). For SCPCR and SCPDR settings, see section 17.2.8, Port SC Control Register (SCPCR)/Port SC Data Register (SCPDR).

Figure 17.8 Sample Serial Transmission Flowchart

In transmitting serial data, the SCI operates as follows:

1. The SCI monitors the TDRE bit in the SCSSR. When TDRE is cleared to 0, the SCI recognizes that the transmit data register (SCTDR) contains new data, and loads this data from the SCTDR into the transmit shift register (SCTSR).

2. After loading the data from the SCTDR into the SCTSR, the SCI sets the TDRE bit to 1 and starts transmitting. If the transmit-data-empty interrupt enable bit (TIE) is set to 1 in the SCSCR, the SCI requests a transmit-data-empty interrupt (TXI) at this time.

   Serial transmit data is transmitted in the following order from the TxD pin:

   a. Start bit: One 0 bit is output.
   b. Transmit data: Seven or eight bits of data are output, LSB first.
   c. Parity bit or multiprocessor bit: One parity bit (even or odd parity) or one multiprocessor bit is output. Formats in which neither a parity bit nor a multiprocessor bit is output can also be selected.
   d. Stop bit: One or two 1 bits (stop bits) are output.
   e. Marking: Output of 1 bits continues until the start bit of the next transmit data.

3. The SCI checks the TDRE bit when it outputs the stop bit.

   If TDRE is 0, the SCI loads new data from the SCTDR into the SCTSR, outputs the stop bit, then begins serial transmission of the next frame.

   If TDRE is 1, the SCI sets the TEND bit to 1 in the SCSSR, outputs the stop bit, then continues output of 1 bits (marking). If the transmit-end interrupt enable bit (TEIE) in the SCSCR is set to 1, a transmit-end interrupt (TEI) is requested.

Figure 17.9 shows an example of SCI transmit operation in the asynchronous mode.

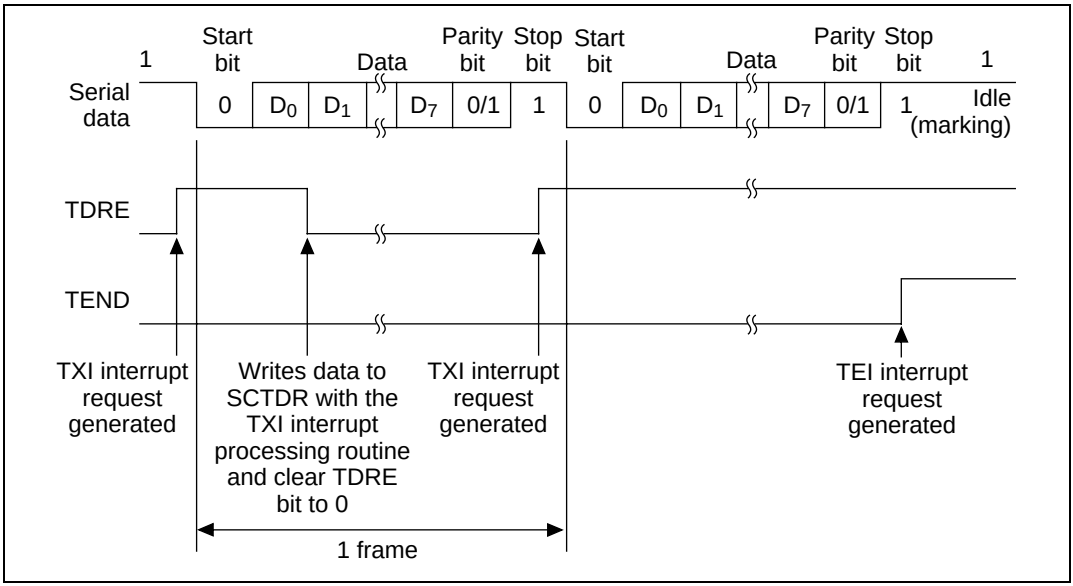


Figure 17.9 Example of Operation in Transmission in Asynchronous Mode (Example with 8-Bit Data, Parity, One Stop Bit)

Serial Data Reception (Asynchronous Mode):

Figure 17.10 shows a sample flow chart for serial data reception. After enabling the SCI reception, receive serial data following the procedure shown below:

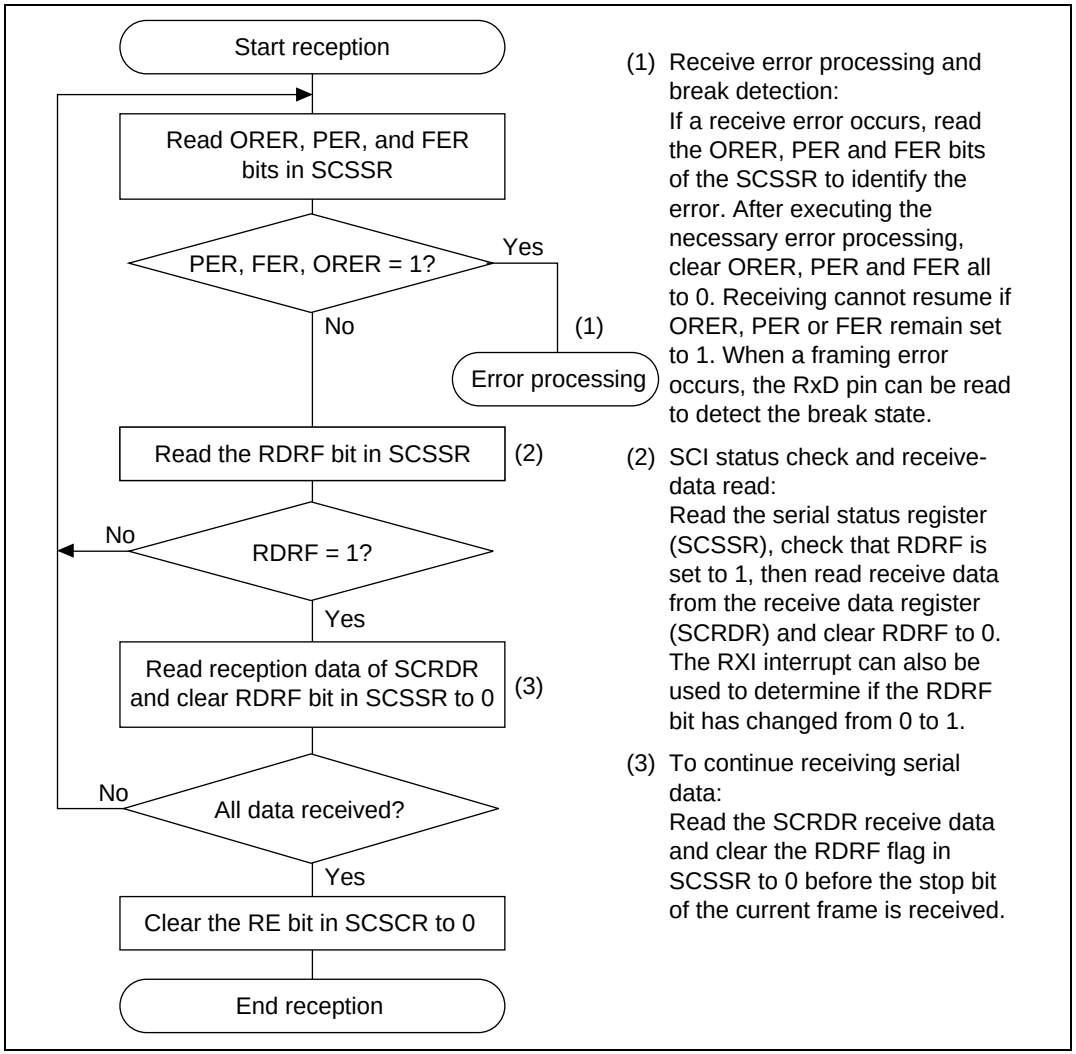


Figure 17.10 Sample Serial Reception Data Flowchart (1)

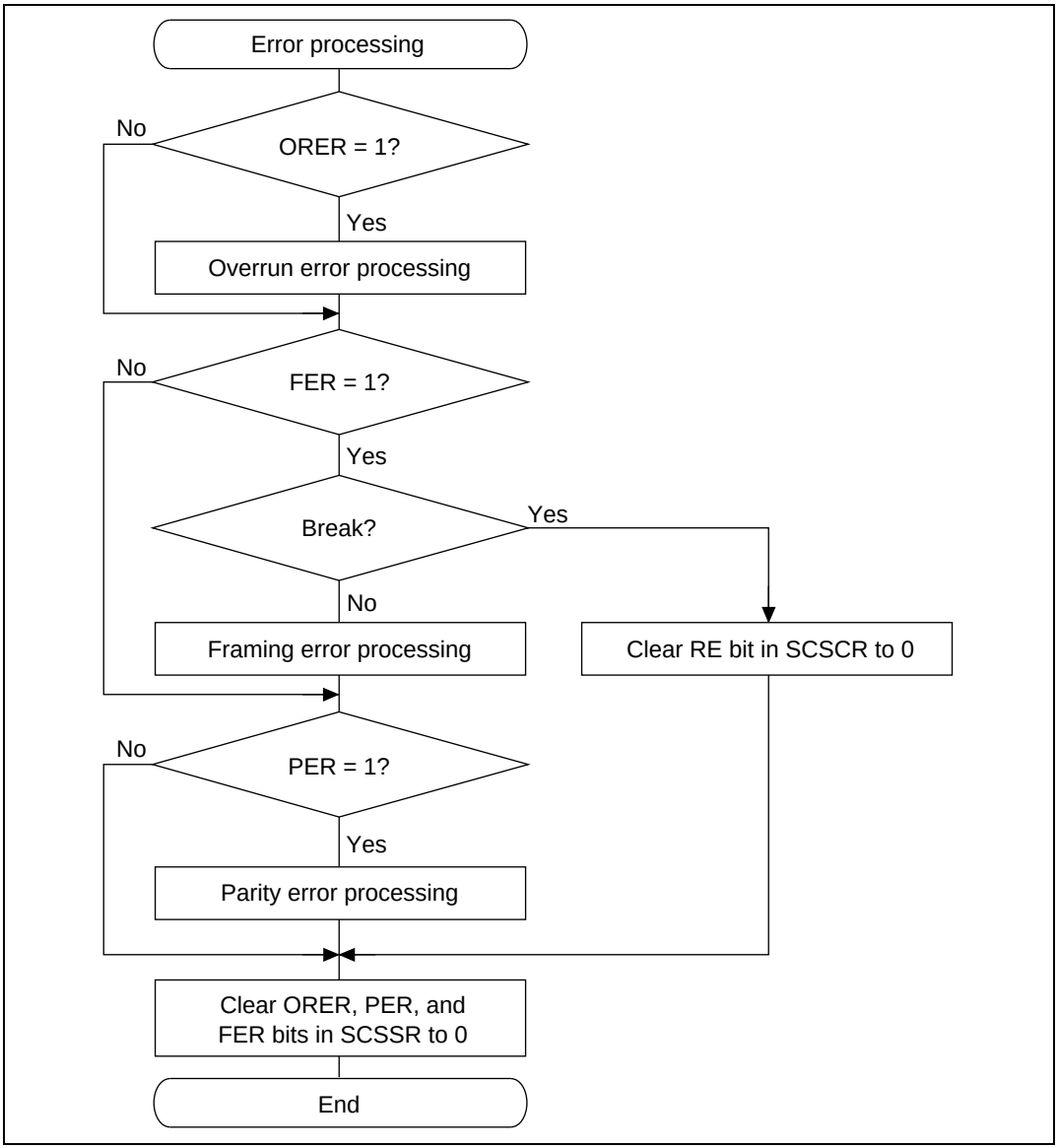


Figure 17.10 Sample Serial Reception Data Flowchart (2)

In receiving, the SCI operates as follows:

1. The SCI monitors the communication line. When it detects a start bit (0), the SCI synchronizes internally and starts receiving.

2. Receive data is shifted into the SCRSR in order from the LSB to the MSB.

3. The parity bit and stop bit are received. After receiving these bits, the SCI makes the following checks:

   a. Parity check: The number of 1s in the receive data must match the even or odd parity setting of the O/$\overline{E}$ bit in the SCSMR.
   b. Stop bit check: The stop bit value must be 1. If there are two stop bits, only the first stop bit is checked.
   c. Status check: RDRF must be 0 so that receive data can be loaded from the SCRSR into the SCRDR.

   If these checks all pass, the SCI sets RDRF to 1 and stores the received data in the SCRDR. If one of the checks fails (receive error), the SCI operates as indicated in table 17.12.

Note: When a receive error flag is set, further receiving is disabled. The RDRF bit is not set to 1. Be sure to clear the error flags.

4. After setting RDRF to 1, if the receive-data-full interrupt enable bit (RIE) is set to 1 in the SCSCR, the SCI requests a receive-data-full interrupt (RXI). If one of the error flags (ORER, PER, or FER) is set to 1 and the receive-data-full interrupt enable bit (RIE) in the SCSCR is also set to 1, the SCI requests a receive-error interrupt (ERI).

Table 17.12 Receive Error Conditions and SCI Operation

Receive Error	Abbreviation	Condition	Data Transfer
Overrun error	ORER	Receiving of next data ends while RDRF is still set to 1 in SCSSR	Receive data not loaded from SCRSR into SCRDR
Framing error	FER	Stop bit is 0	Receive data loaded from SCRSR into SCRDR
Parity error	PER	Parity of receive data differs from even/odd parity setting in SCSMR	Receive data loaded from SCRSR into SCRDR

Figure 17.11 shows an example of SCI receive operation in the asynchronous mode.

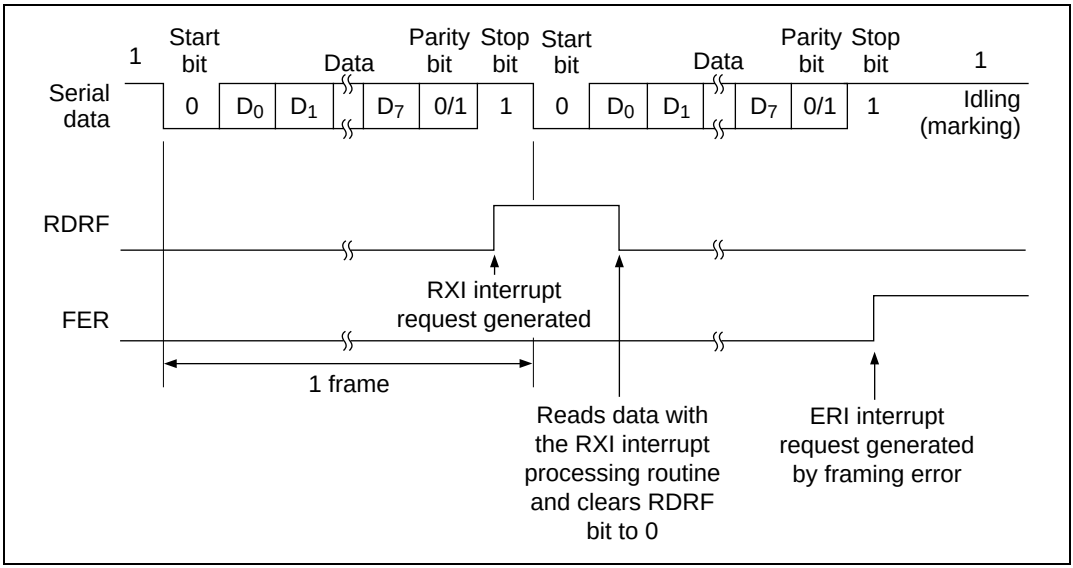


Figure 17.11 Example of SCI Operation in Reception (Example with 8-Bit Data, Parity, One Stop Bit)

17.3.3 Multiprocessor Communication

The multiprocessor communication function enables several processors to share a single serial communication line. The processors communicate in the asynchronous mode using a format with an additional multiprocessor bit (multiprocessor format).

In multiprocessor communication, each receiving processor is addressed by a unique ID.

A serial communication cycle consists of an ID-sending cycle that identifies the receiving processor, and a data-sending cycle. The multiprocessor bit distinguishes ID-sending cycles from data-sending cycles.

The transmitting processor starts by sending the ID of the receiving processor with which it wants to communicate as data with the multiprocessor bit set to 1. Next the transmitting processor sends transmit data with the multiprocessor bit cleared to 0.

Receiving processors skip incoming data until they receive data with the multiprocessor bit set to 1.

When they receive data with the multiprocessor bit set to 1, receiving processors compare the data with their IDs. The receiving processor with a matching ID continues to receive further incoming data. Processors with IDs not matching the received data skip further incoming data until they

again receive data with the multiprocessor bit set to 1. Multiple processors can send and receive data in this way.

Figure 17.12 shows an example of communication among processors using the multiprocessor format.

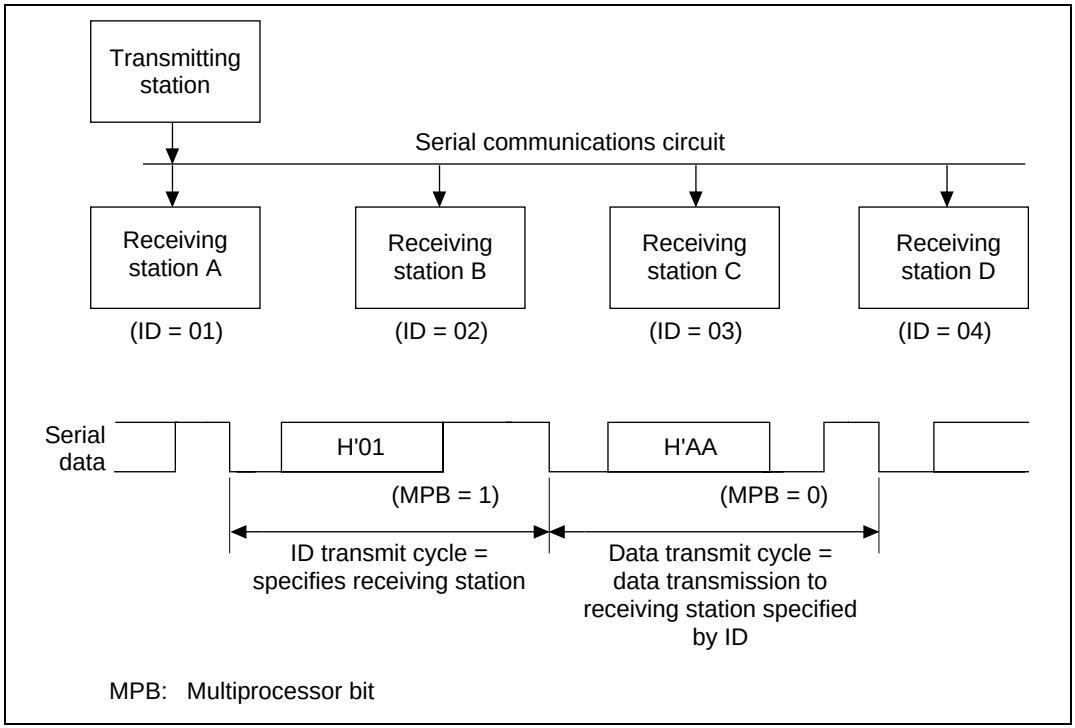


Figure 17.12 Example of Communication Using Multiprocessor Format (Transmission of Data H'AA to Receiving Station A)

Communication Formats

Four formats are available. Parity-bit settings are ignored when the multiprocessor format is selected. For details see table 17.11.

Clock

See the description in the asynchronous mode section.

Transmitting and Receiving Data

Multiprocessor Serial Data Transmission:

Figure 17.13 shows a sample flow chart for multiprocessor serial data transmission. After enabling the SCI transmission, transmit multiprocessor serial data following the procedure shown below:

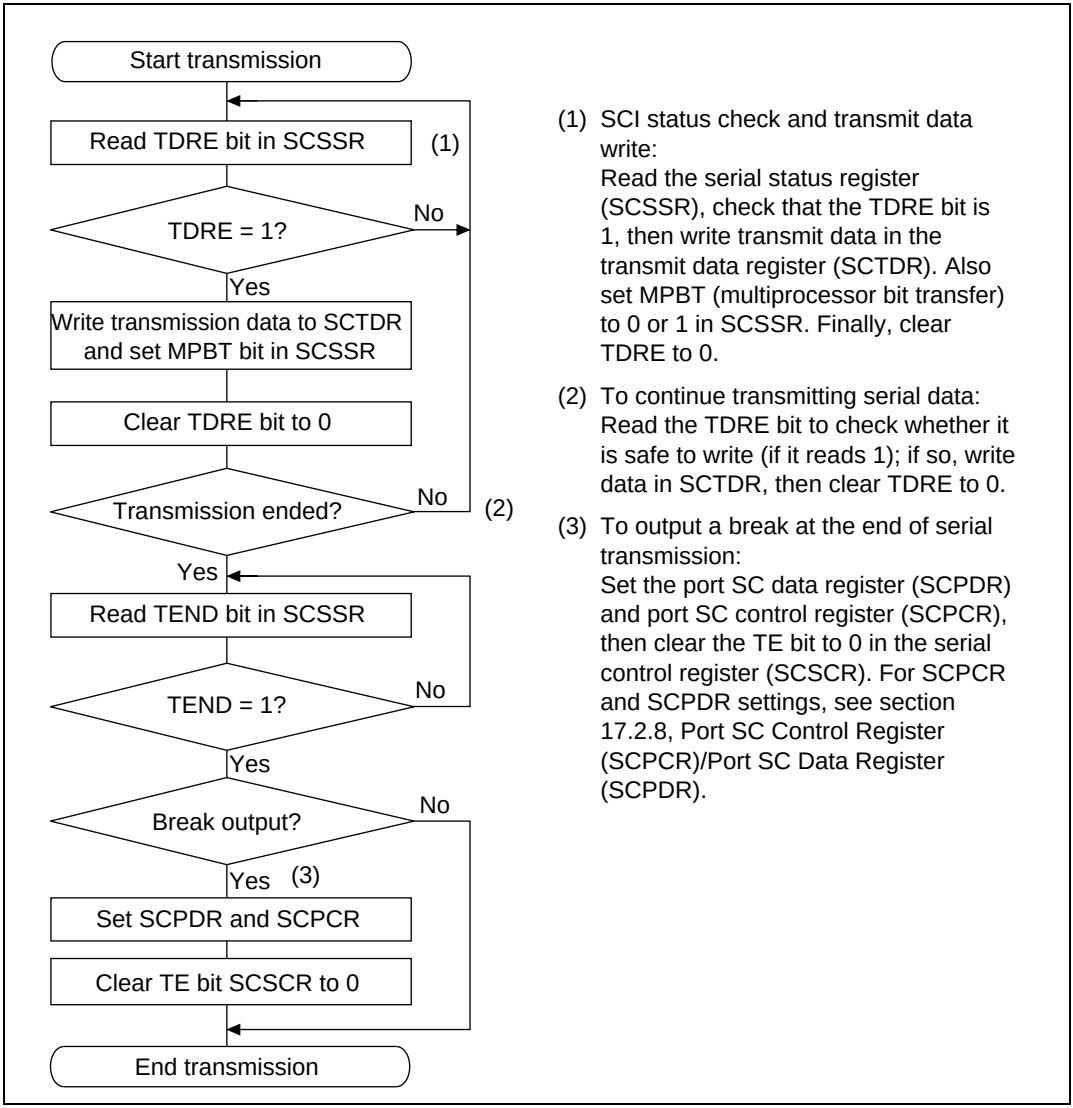


Figure 17.13 Sample Multiprocessor Serial Transmission Flowchart

In transmitting serial data, the SCI operates as follows:

1. The SCI monitors the TDRE bit in the SCSSR. When TDRE is cleared to 0 the SCI recognizes that the transmit data register (SCTDR) contains new data, and loads this data from the SCTDR into the transmit shift register (SCTSR).

2. After loading the data from the SCTDR into the SCTSR, the SCI sets the TDRE bit to 1 and starts transmitting. If the transmit-data-empty interrupt enable bit (TIE) in the SCSCR is set to 1, the SCI requests a transmit-data-empty interrupt (TXI) at this time. Serial transmit data is transmitted in the following order from the TxD pin:

   a. Start bit: One 0 bit is output.
   b. Transmit data: Seven or eight bits are output, LSB first.
   c. Multiprocessor bit: One multiprocessor bit (MPBT value) is output.
   d. Stop bit: One or two 1 bits (stop bits) are output.
   e. Marking: Output of 1 bits continues until the start bit of the next transmit data.

3. The SCI checks the TDRE bit when it outputs the stop bit. If TDRE is 0, the SCI loads data from the SCTDR into the SCTSR, outputs the stop bit, then begins serial transmission of the next frame. If TDRE is 1, the SCI sets the TEND bit in the SCSSR to 1, outputs the stop bit, then continues output of 1 bits in the marking state. If the transmit-end interrupt enable bit (TEIE) in the SCSCR is set to 1, a transmit-end interrupt (TEI) is requested at this time.

Figure 17.14 shows SCI transmission in the multiprocessor format.

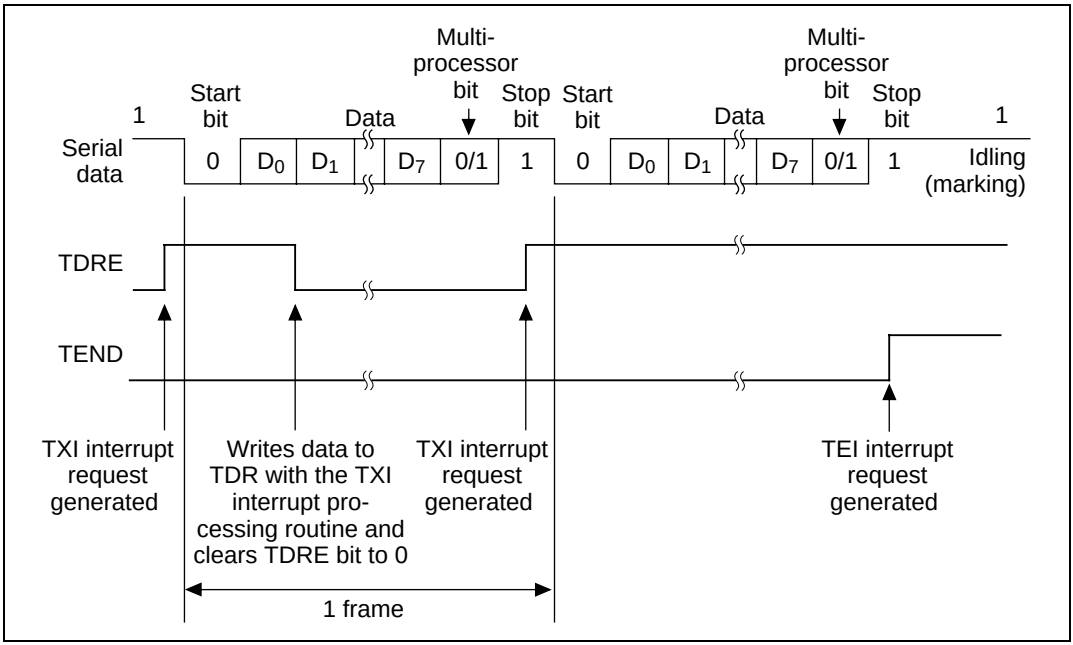


Figure 17.14 Example of SCI Operation in Transmission (Example with 8-Bit Data, Multiprocessor Bit, One Stop Bit)

Multiprocessor Serial Data Reception:

Figure 17.15 shows a sample flow chart for multiprocessor serial data reception. After enabling the SCI reception, receive multiprocessor serial data following the procedure shown below:

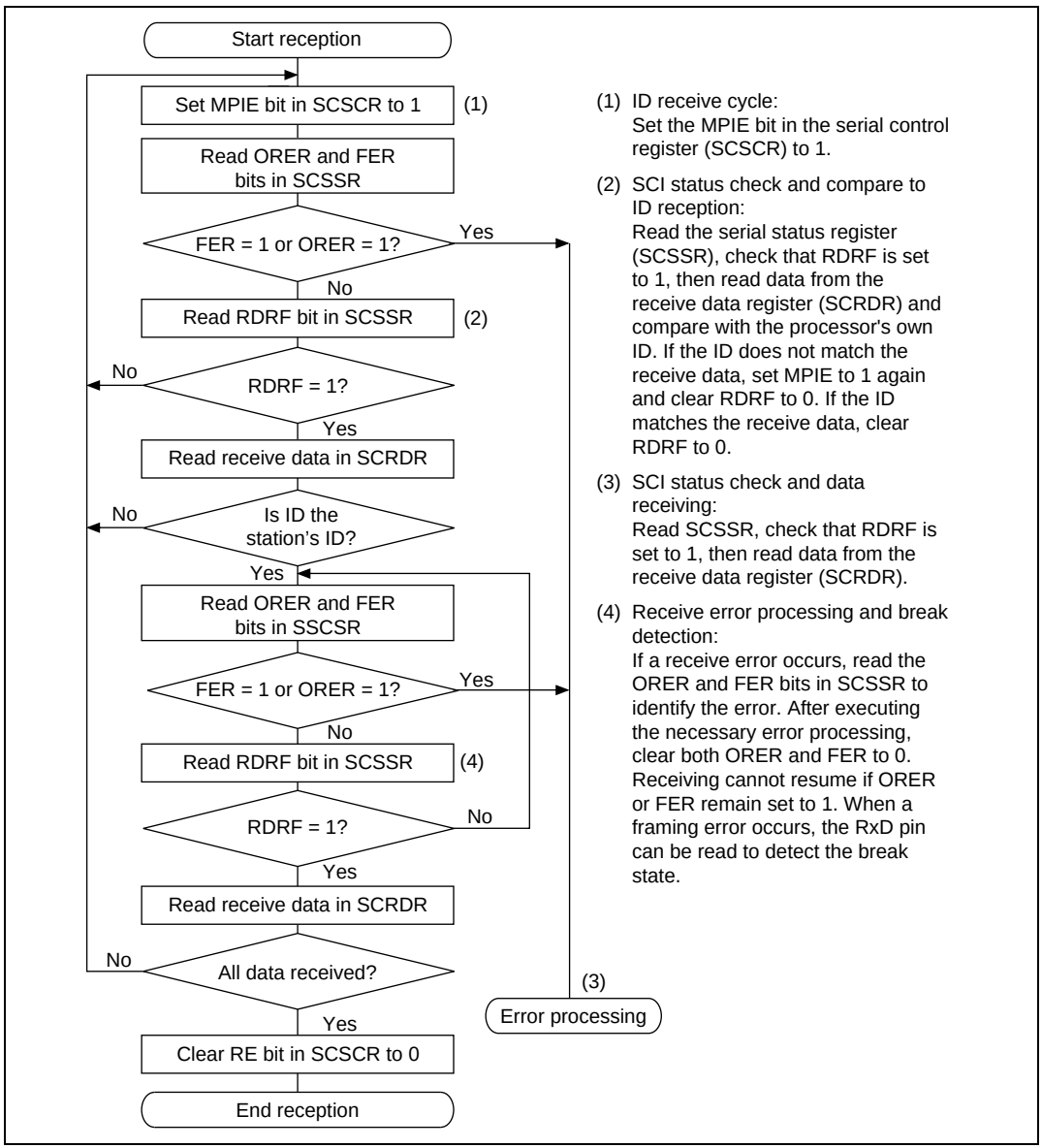


Figure 17.15 Sample Multiprocessor Serial Reception Flowchart (1)

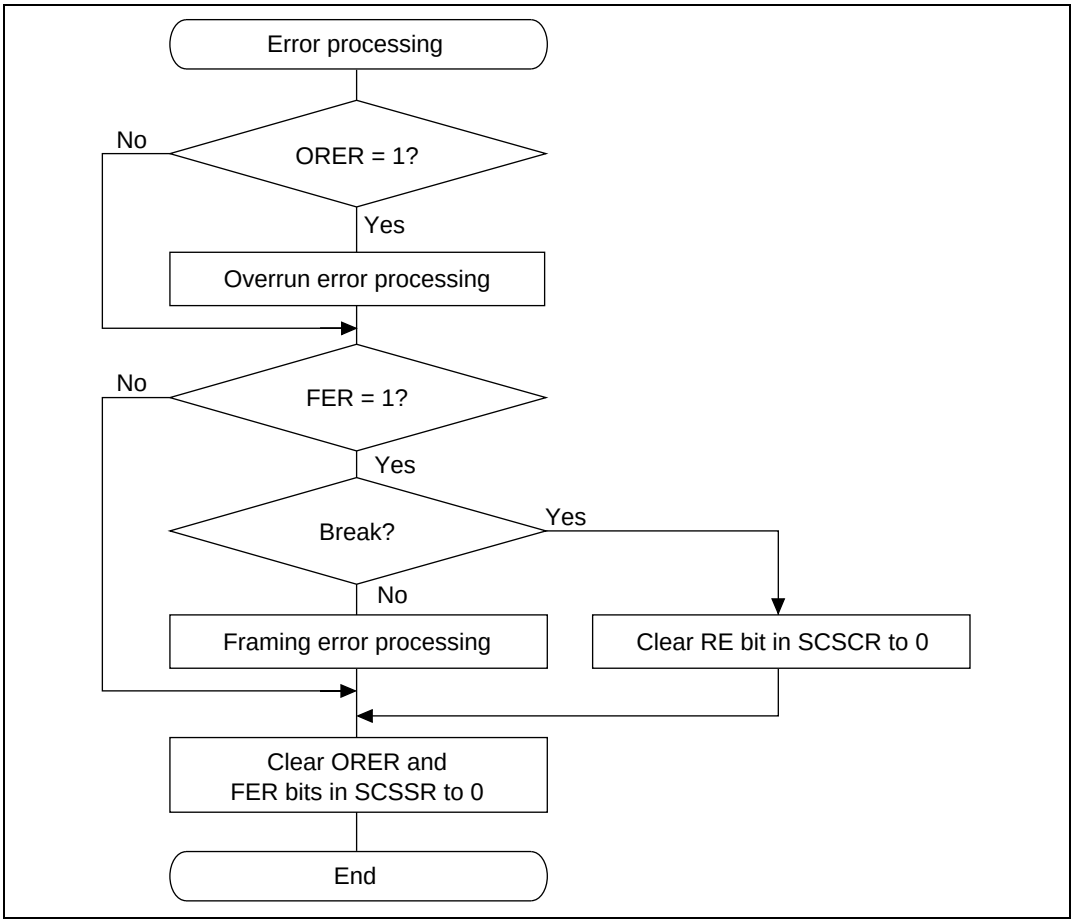


Figure 17.15 Sample Multiprocessor Serial Reception Flowchart (2)

Figure 17.16 shows an example of SCI receive operation using a multiprocessor format.

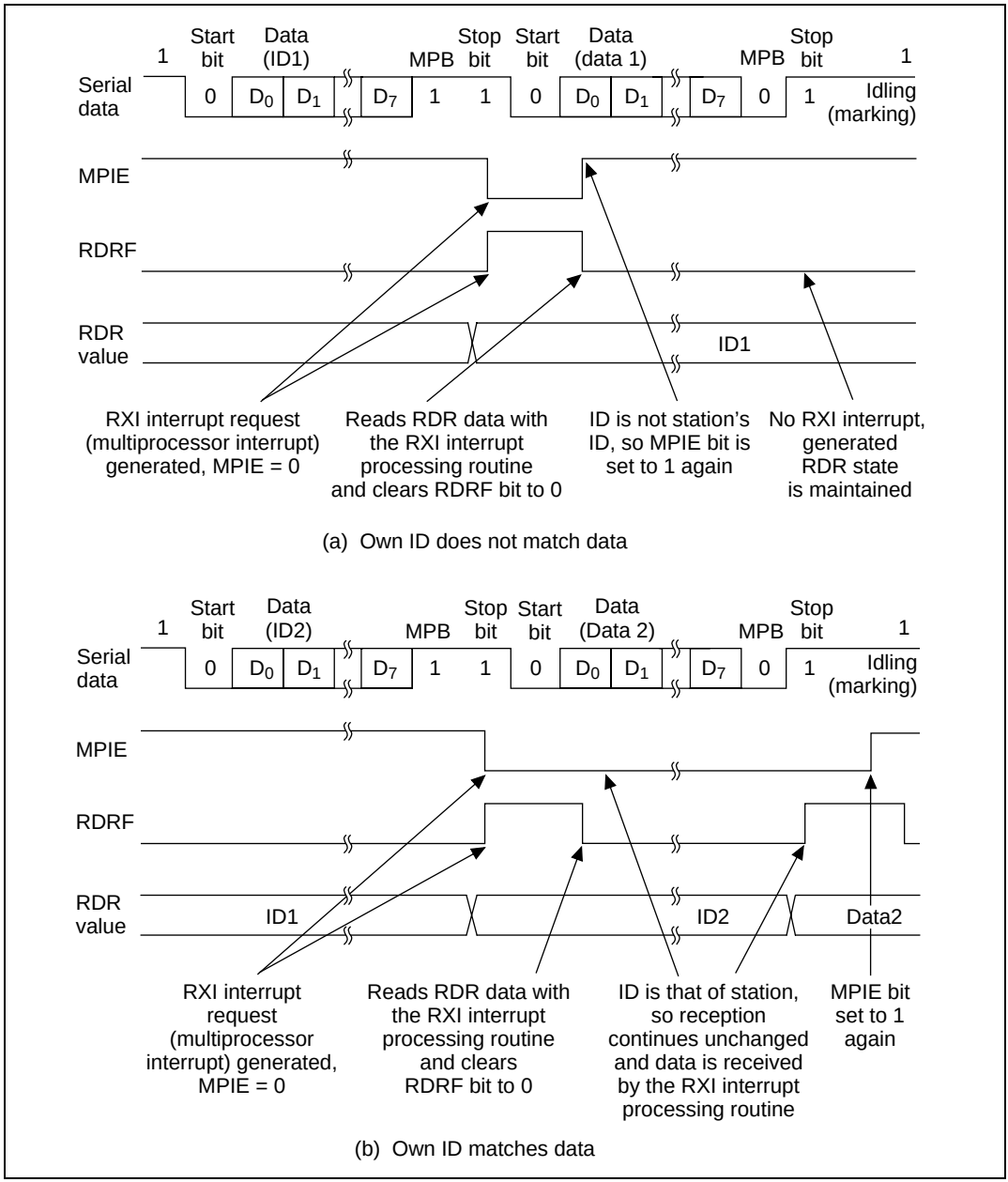


Figure 17.16 Example of SCI Operation in Reception (Example with 8-Bit Data, Multiprocessor Bit, One Stop Bit)

17.3.4 Clock Synchronous Operation

In the clock synchronous mode, the SCI transmits and receives data in synchronization with clock pulses. This mode is suitable for high-speed serial communication.

The SCI transmitter and receiver are independent, so full-duplex communication is possible while sharing the same clock. The transmitter and receiver are also double buffered, so continuous transmitting or receiving is possible by reading or writing data while transmitting or receiving is in progress.

Figure 17.17 shows the general format in clock synchronous serial communication.

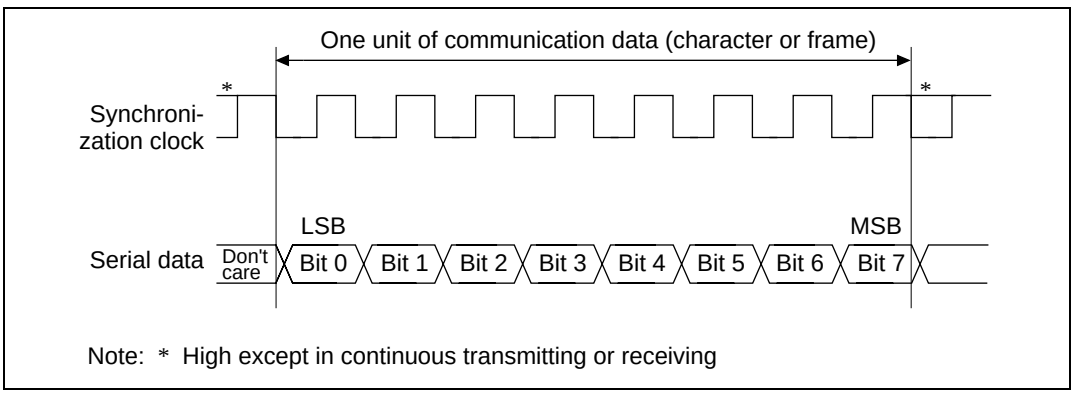


Figure 17.17 Data Format in Clock Synchronous Communication

In clock synchronous serial communication, each data bit is output on the communication line from one falling edge of the serial clock to the next. Data are guaranteed valid at the rising edge of the serial clock.

In each character, the serial data bits are transmitted in order from the LSB (first) to the MSB (last). After output of the MSB, the communication line remains in the state of the MSB.

In the clock synchronous mode, the SCI transmits or receives data by synchronizing with the rising edge of the serial clock.

Communication Format

The data length is fixed at eight bits. No parity bit or multiprocessor bit can be added.

Clock

An internal clock generated by the on-chip baud rate generator or an external clock input from the SCK pin can be selected as the SCI transmit/receive clock. The clock source is selected by the C/$\overline{A}$ bit in the serial mode register (SCSMR) and bits CKE1 and CKE0 in the serial control register (SCSCR). See table 17.10.

When the SCI operates on an internal clock, it outputs the clock signal at the SCK pin. Eight clock pulses are output per transmitted or received character. When the SCI is not transmitting or receiving, the clock signal remains in the high state. When only receiving, the SCI receives in 2-character units, so a 16 pulse synchronization clock is output. To receive in 1-character units, select an external clock source.

Transmitting and Receiving Data

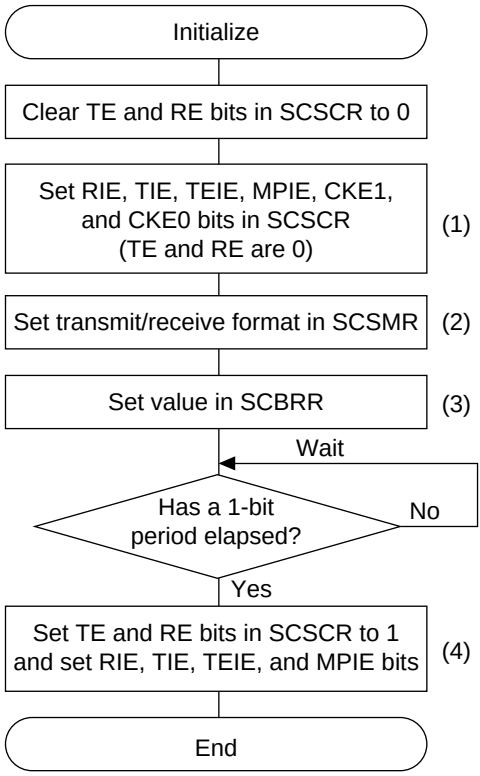
SCI Initialization (clock synchronous mode)

Before transmitting and receiving data, the TE and RE bits in SCSCR should be cleared to 0, then the SCI should be initialized as described in a sample flowchart in figure 17.18.

When the operating mode, or transfer format, is changed for example, the TE and RE bits must be cleared to 0 before making the change using the following procedure. When the TE bit is cleared to 0, the TDRE flag is set to 1 and the transmit shift register (SCTSR) is initialized.

Note that clearing the RE bit to 0 does not change the contents of the RDRF, PER, FER, and ORER flags, or the contents of SCRDR.

Figure 17.18 is a sample flowchart for initializing the SCI.



(1) Select the clock source in the serial control register (SCSCR). Leave RIE, TIE, TEIE, MPIE, TE and RE cleared to 0.

(2) Select the communication format in the serial mode register (SCSMR).

(3) Write the value corresponding to the bit rate in the bit rate register (SCBRR) unless an external clock is used.

(4) Wait for at least the interval required to transmit or receive one bit, then set TE or RE in the serial control register (SCSCR) to 1. Also set RIE, TIE, TEIE and MPIE. Setting TE and RE allows use of the TxD and RxD pins.

Figure 17.18 Sample SCI Initialization Flowchart

Serial Data Transmission (Clock Synchronous Mode):

Figure 17.19 shows a sample flow chart for serial data transmission. After enabling the SCI transmission, transmit serial data following the procedure shown below:

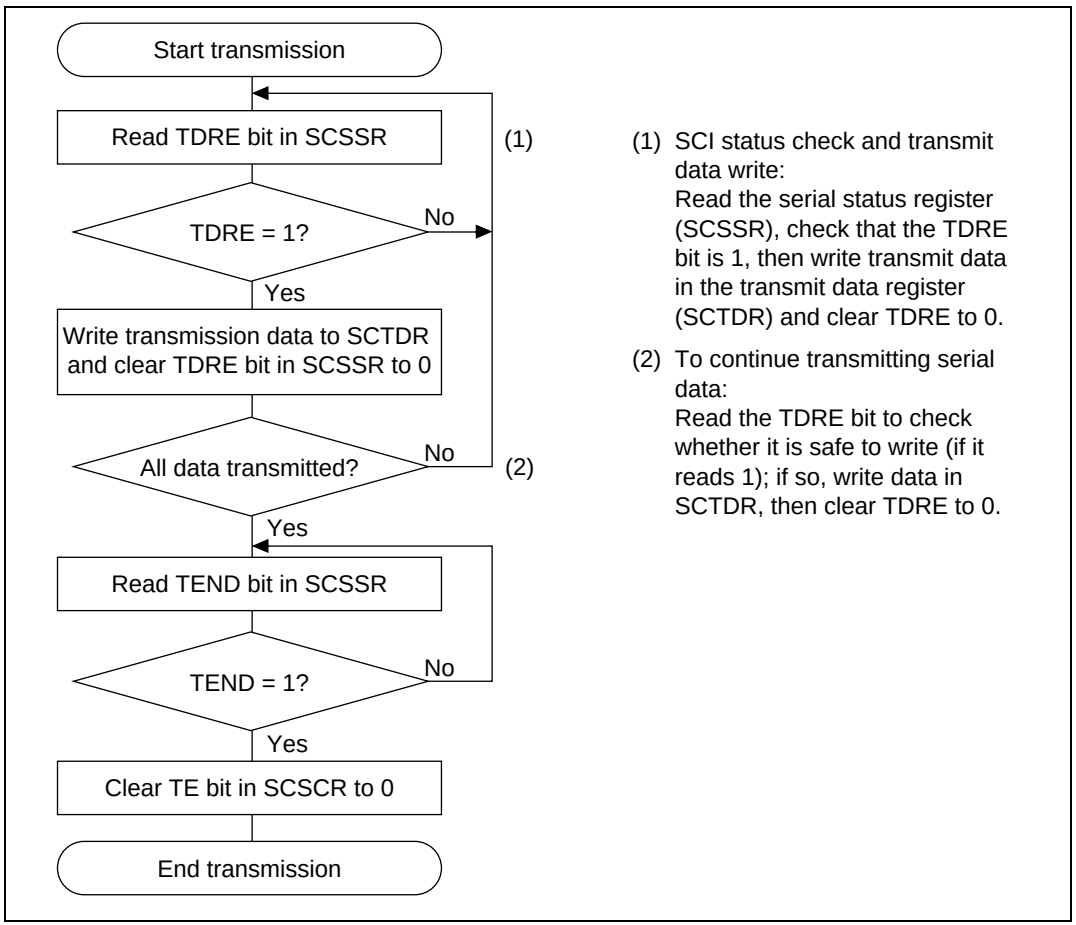


Figure 17.19 Sample Serial Transmission Flowchart

In transmitting serial data, the SCI operates as follows:

1. The SCI monitors the TDRE bit in the SCSSR. When TDRE is cleared to 0 the SCI recognizes that the transmit data register (SCTDR) contains new data and loads this data from the SCTDR into the transmit shift register (SCTSR).

2. After loading the data from the SCTDR into the SCTSR, the SCI sets the TDRE bit to 1 and starts transmitting. If the transmit-data-empty interrupt enable bit (TIE) in the SCSCR is set to 1, the SCI requests a transmit-data-empty interrupt (TXI) at this time.

   If clock output mode is selected, the SCI outputs eight synchronous clock pulses. If an external clock source is selected, the SCI outputs data in synchronization with the input clock. Data are output from the TxD0 pin in order from the LSB (bit 0) to the MSB (bit 7).

3. The SCI checks the TDRE bit when it outputs the MSB (bit 7). If TDRE is 0, the SCI loads data from the SCTDR into the SCTSR, then begins serial transmission of the next frame. If TDRE is 1, the SCI sets the TEND bit in the SCSSR to 1, transmits the MSB, then holds the transmit data pin (TxD0) in the MSB state. If the transmit-end interrupt enable bit (TEIE) in the SCSCR is set to 1, a transmit-end interrupt (TEI) is requested at this time.

4. After the end of serial transmission, the SCK0 pin is held in the high state.

Figure 17.20 shows an example of SCI transmit operation.

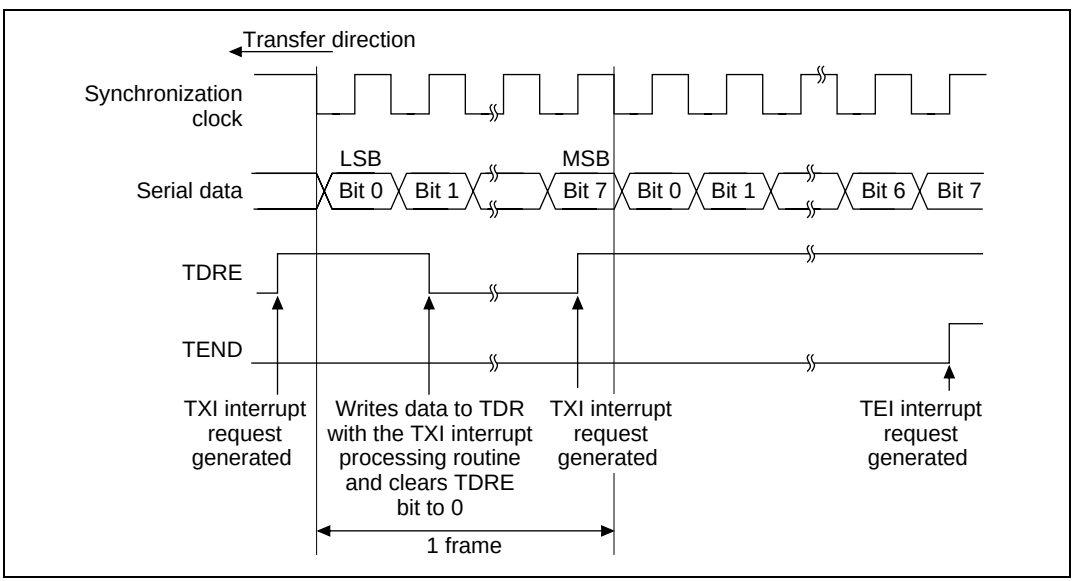


Figure 17.20 Sample SCI Transmission Operation in Clocked Synchronous Mode

Serial Data Reception (Clock Synchronous Mode):

Figure 17.21 shows a sample flow chart for serial data reception. After enabling the SCI transmission, transmit serial data following the procedure shown below:

When switching from the asynchronous mode to the clock synchronous mode, make sure that ORER, PER, and FER are cleared to 0. If PER or FER is set to 1, the RDRF bit will not be set and both transmitting and receiving will be disabled.

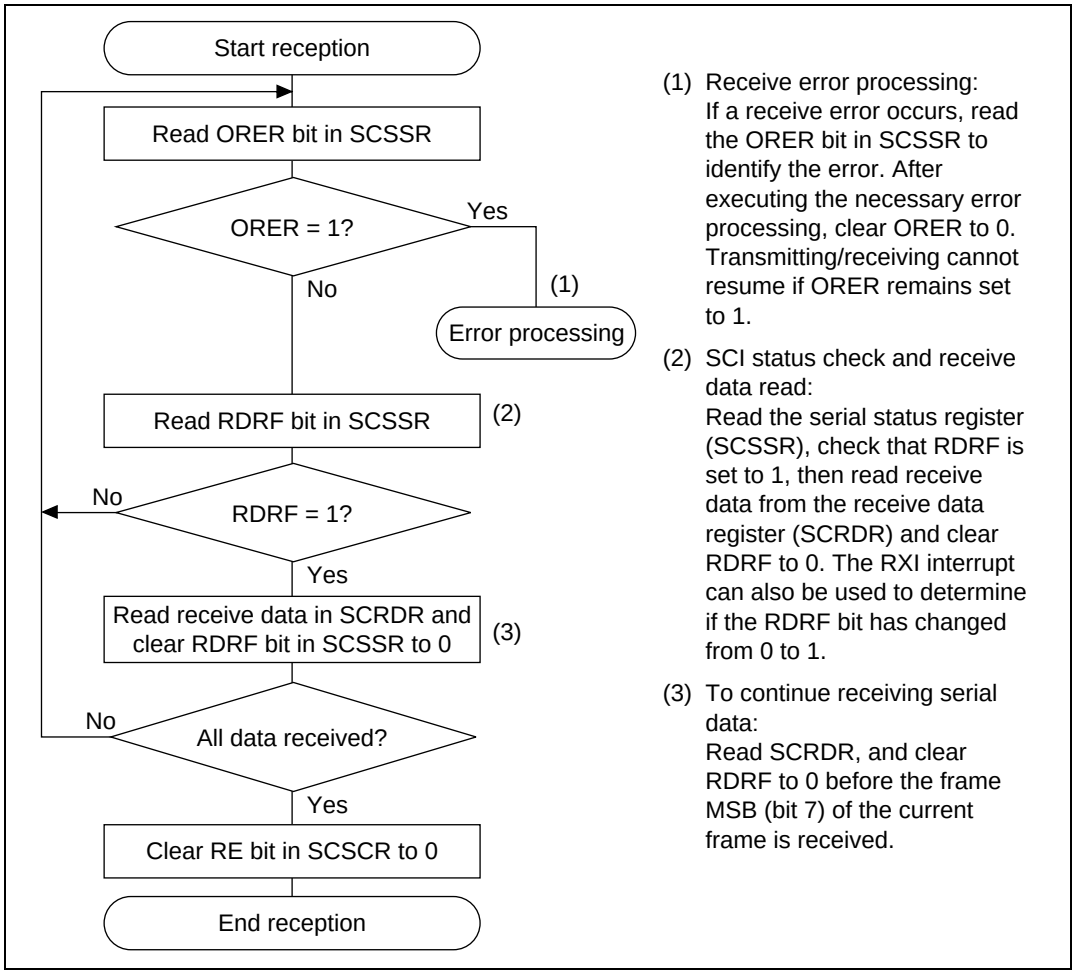


Figure 17.21 Sample Serial Reception Flowchart (1)

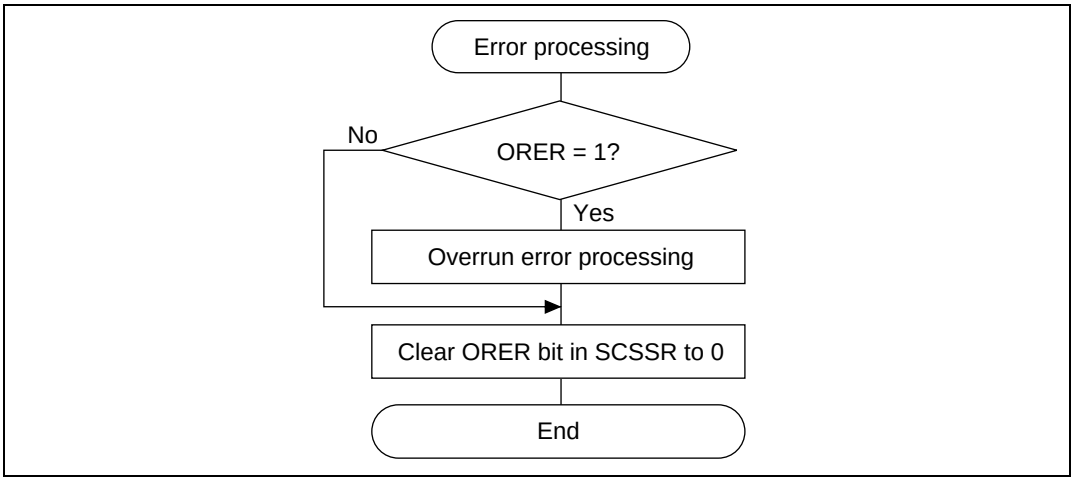


Figure 17.21 Sample Serial Reception Flowchart (2)

In receiving, the SCI operates as follows:

1. The SCI synchronizes with serial clock input or output and initializes internally.

2. Receive data is shifted into the SCRSR in order from the LSB to the MSB. After receiving the data, the SCI checks that RDRF is 0 so that receive data can be loaded from the SCRSR into the SCRDR. If this check is passed, the SCI sets RDRF to 1 and stores the received data in the SCRDR. If the check is not passed (receive error), the SCI operates as indicated in table 17.12. This state prevents further transmission or reception. While receiving, the RDRF bit is not set to 1. Be sure to clear the error flag.

3. After setting RDRF to 1, if the receive-data-full interrupt enable bit (RIE) is set to 1 in the SCSCR, the SCI requests a receive-data-full interrupt (RXI). If the ORER bit is set to 1 and the receive-data-full interrupt enable bit (RIE) in the SCSCR is also set to 1, the SCI requests a receive-error interrupt (ERI).

Figure 17.22 shows an example of the SCI receive operation.

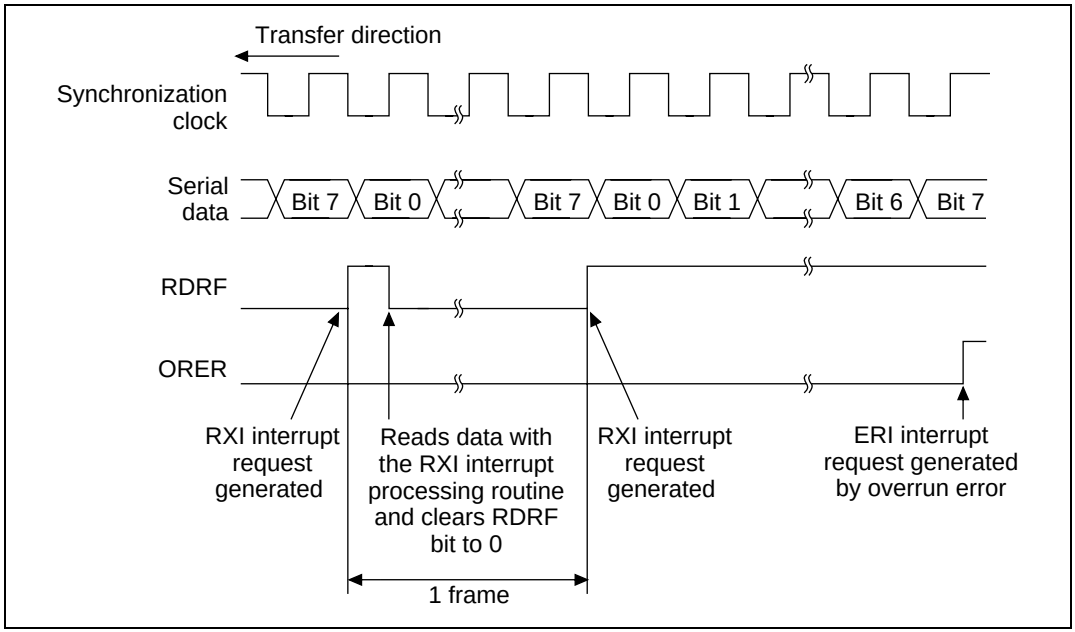
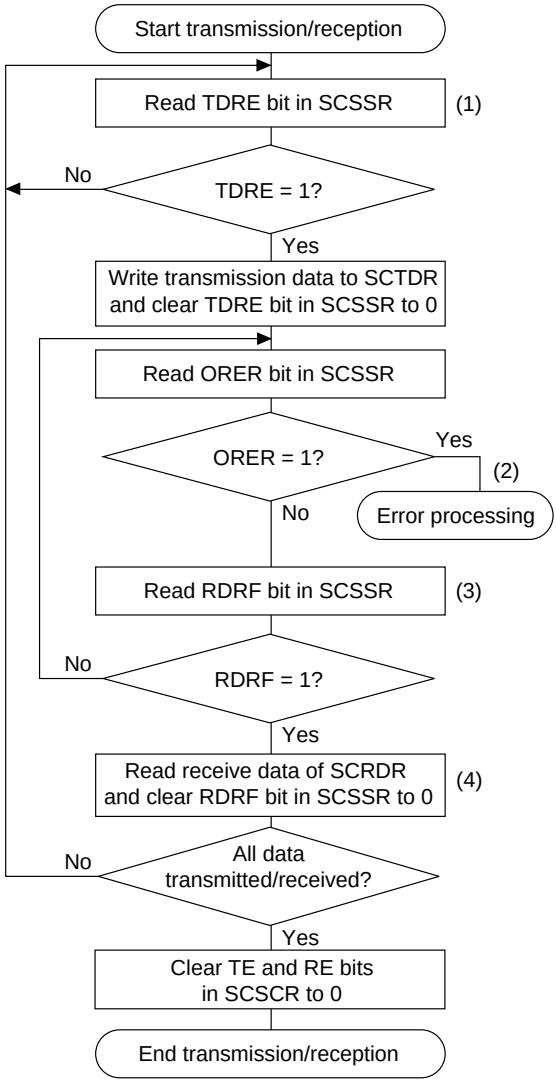


Figure 17.22 Example of SCI Operation in Reception

Transmitting and Receiving Serial Data Simultaneously (Clock Synchronous Mode)

Figure 17.23 shows a sample flowchart for simultaneous serial transmit and receive operations. After enabling the SCI transmission/reception, provide simultaneous serial transmit and receive operations following the procedure shown below:



(1) SCI status check and transmit data write:
Read the serial status register (SCSSR), check that the TDRE bit is 1, then write transmit data in the transmit data register (SCTDR) and clear TDRE to 0. The TXI interrupt can also be used to determine if the TDRE bit has changed from 0 to 1.

(2) Receive error processing:
If a receive error occurs, read the ORER bit in SCSSR to identify the error. After executing the necessary error processing, clear ORER to 0. Transmitting/receiving cannot resume if ORER remains set to 1.

(3) SCI status check and receive data read:
Read the serial status register (SCSSR), check that RDRF is set to 1, then read receive data from the receive data register (SCRDR) and clear RDRF to 0. The RXI interrupt can also be used to determine if the RDRF bit has changed from 0 to 1.

(4) To continue transmitting and receiving serial data:
Read the RDRF bit and SCRDR, and clear RDRF to 0 before the frame MSB (bit 7) of the current frame is received. Also read the TDRE bit to check whether it is safe to write (if it reads 1); if so, write data in SCTDR, then clear TDRE to 0 before the MSB (bit 7) of the current frame is transmitted.

Note: When switching transmit or receive operation to simultaneous serial transmit and receive operations, first clear the TE bit and RE bit to 0, and then set both these bits to 1 simultaneously.

Figure 17.23 Sample Flowchart of Simultaneous Serial Transmit and Receive Operations

17.4 SCI Interrupt Sources

The SCI has four interrupt sources in each channel: Transmit-end (TEI), receive-error (ERI), receive-data-full (RXI), and transmit-data-empty (TXI). Table 17.13 lists the interrupt sources and indicates their priority. These interrupts can be enabled and disabled by the TIE, RIE, and TEIE bits in the serial control register (SCSCR). Each interrupt request is sent separately to the interrupt controller.

TXI is requested when the TDRE bit in the SCSSR is set to 1.

RXI is requested when the RDRF bit in the SCSSR is set to 1.

ERI is requested when the ORER, PER, or FER bit in the SCSSR is set to 1.

TEI is requested when the TEND bit in the SCSSR is set to 1. Where the TXI interrupt indicates that transmit data writing is enabled, the TEI interrupt indicates that the transmit operation is complete.

Table 17.13 SCI Interrupt Sources

Interrupt Source	Description	Priority When Reset Is Cleared
ERI	Receive error (ORER, PER, or FER)	High
RXI	Receive data full (RDRF)	↕
TXI	Transmit data empty (TDRE)	
TEI	Transmit end (TEND)	Low

See section 4, Exception Handling, for information on the priority order and relationship to non-SCI interrupts.

17.5 Usage Notes

Note the following points when using the SCI.

SCTDR Write and TDRE Flags: The TDRE bit in the serial status register (SCSSR) is a status flag indicating loading of transmit data from the SCTDR into the SCTSR. The SCI sets TDRE to 1 when it transfers data from the SCTDR to the SCTSR. Data can be written to the SCTDR regardless of the TDRE bit state. If new data is written in the SCTDR when TDRE is 0, however, the old data stored in the SCTDR will be lost because the data has not yet been transferred to the SCTSR. Before writing transmit data to the SCTDR, be sure to check that TDRE is set to 1.

Simultaneous Multiple Receive Errors: Table 17.14 indicates the state of the SCSSR status flags when multiple receive errors occur simultaneously. When an overrun error occurs, the SCRSR contents cannot be transferred to the SCRDR, so receive data is lost.

Table 17.14 SCSSR Status Flags and Transfer of Receive Data

	SCSSR Status Flags				Receive Data Transfer
Receive Error Status	**RDRF**	**ORER**	**FER**	**PER**	**SCRSR → SCRDR**
Overrun error	1	1	0	0	X
Framing error	0	0	1	0	O
Parity error	0	0	0	1	O
Overrun error + framing error	1	1	1	0	X
Overrun error + parity error	1	1	0	1	X
Framing error + parity error	0	0	1	1	O
Overrun error + framing error + parity error	1	1	1	1	X

X: Receive data is not transferred from SCRSR to SCRDR.

O: Receive data is transferred from SCRSR to SCRDR.

Break Detection and Processing: Break signals can be detected by reading the RxD0 pin directly when a framing error (FER) is detected. In the break state, the input from the RxD0 pin consists of all 0s, so FER is set and the parity error flag (PER) may also be set. In the break state, the SCI receiver continues to operate, so if the FER bit is cleared to 0, it will be set to 1 again.

Sending a Break Signal: The TxD0 pin I/O condition and level can be determined by means of the SCP0DT bit of the port SC data register (SCPDR) and bits SCP0MD0 and SCP0MD1 of the port SC control register (SCPCR). These bits can be used to send breaks. To send a break during serial transmission, clear the SCP0DT bit to 0 (designating low level), then clear the TE bit to 0

(halting transmission). When the TE bit is cleared to 0, the transmitter is initialized regardless of the current transmission state, and 0 is output from the TxD0 pin.

TEND Flag and TE Bit Processing: The TEND flag is set to 1 during transmission of the stop bit of the last data. Consequently, if the TE bit is cleared to 0 immediately after setting of the TEND flag has been confirmed, the stop bit will be in the process of transmission and will not be transmitted normally. Therefore, the TE bit should not be cleared to 0 for at least 0.5 serial clock cycles (or 1.5 cycles if two stop bits are used) after setting of the TEND flag setting is confirmed.

Receive Error Flags and Transmitter Operation (Clock Synchronous Mode Only): When a receive error flag (ORER, PER, or FER) is set to 1, the SCI will not start transmitting even if TDRE is set to 1. Be sure to clear the receive error flags to 0 before starting to transmit. Note that clearing RE to 0 does not clear the receive error flags.

Receive Data Sampling Timing and Reception Margin in the Asynchronous Mode: In the asynchronous mode, the SCI operates on a base clock of 16 times the transfer rate frequency. In receiving, the SCI synchronizes internally with the falling edge of the start bit, which it samples on the base clock. Receive data is latched on the rising edge of the eighth base clock pulse (figure 17.24).

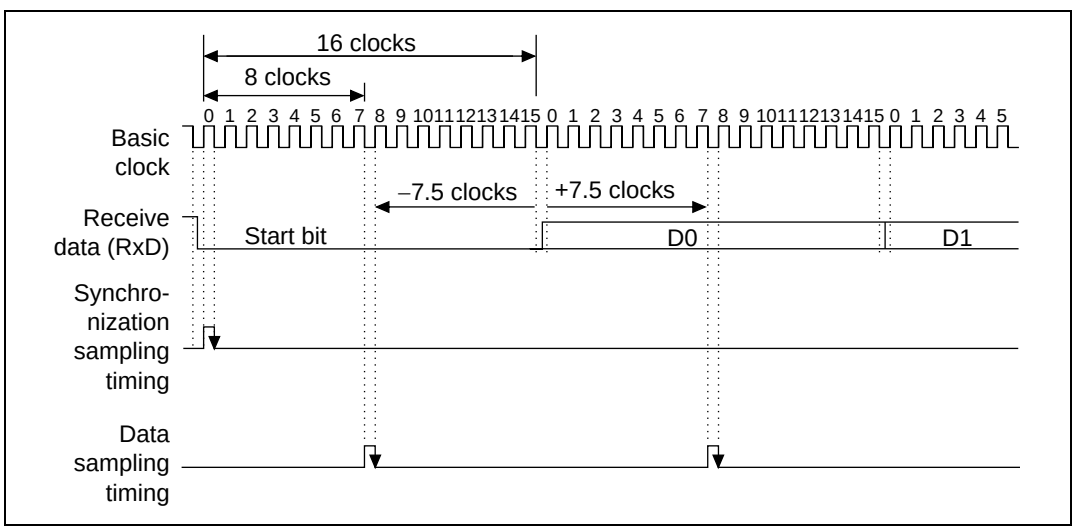


Figure 17.24 Receive Data Sampling Timing in Asynchronous Mode

The reception margin in the asynchronous mode is given by formula 1.

Formula 1:

$$M = \left| \left(0.5 - \frac{1}{2N}\right) - (L - 0.5)F - \frac{|D - 0.5|}{N}(1 + F) \right| \times 100\%$$

Where: M = Reception margin (%)
N = Ratio of clock frequency to bit rate (N = 16)
D = Clock duty cycle (D = 0–1.0)
L = Frame length (L = 9–12)
F = Absolute deviation of clock frequency

Assuming values of F = 0, D = 0.5 and N = 372 in formula (1), the reception margin is given by formula 2.

Formula 2:

$$M = (0.5 - 1/(2 \times 16)) \times 100\%$$
$$= 46.875\%$$

This is a theoretical value. A reasonable margin to allow in system designs is 20 to 30%.

Cautions for Clock Synchronous External Clock Mode:

- Set TE = RE = 1 only when the external clock SCK is 1.
- Do not set TE = RE = 1 until at least four clocks after the external clock SCK has changed from 0 to 1.
- When receiving, RDRF is 1 when RE is set to zero 2.5 to 3.5 clocks after the rising edge of the SCK input of the D7 bit in RxD, but it cannot be copied to SCRDR.

Caution for Clock Synchronous Internal Clock Mode: When receiving, RDRF is 1 when RE is set to zero 1.5 clocks after the rising edge of the SCK output of the D7 bit in RxD, but it cannot be copied to SCRDR.

Section 18 Smart Card Interface

18.1 Overview

As an additional serial communications interface function (SCI), an IC card (smart card) interface that is compatible to the ISO/IEC standard 7816-3 for identification of cards is supported. Register settings are used to switch between the ordinary serial communication interface and the smart card interface.

18.1.1 Features

The smart card interface has the following features:

- Asynchronous mode
  - Data length: Eight bits
  - Parity bit generation and check
  - Receive mode error signal detection (parity error)
  - Transmit mode error signal detection and automatic re-transmission of data
  - Supports both direct convention and inverse convention
- Bit rate can be selected using on-chip baud rate generator.
- Three types of interrupts: Transmit-data-empty, receive-data-full, and communication-error interrupts are requested independently.

18.1.2 Block Diagram

Figure 18.1 shows a block diagram of the smart card interface.

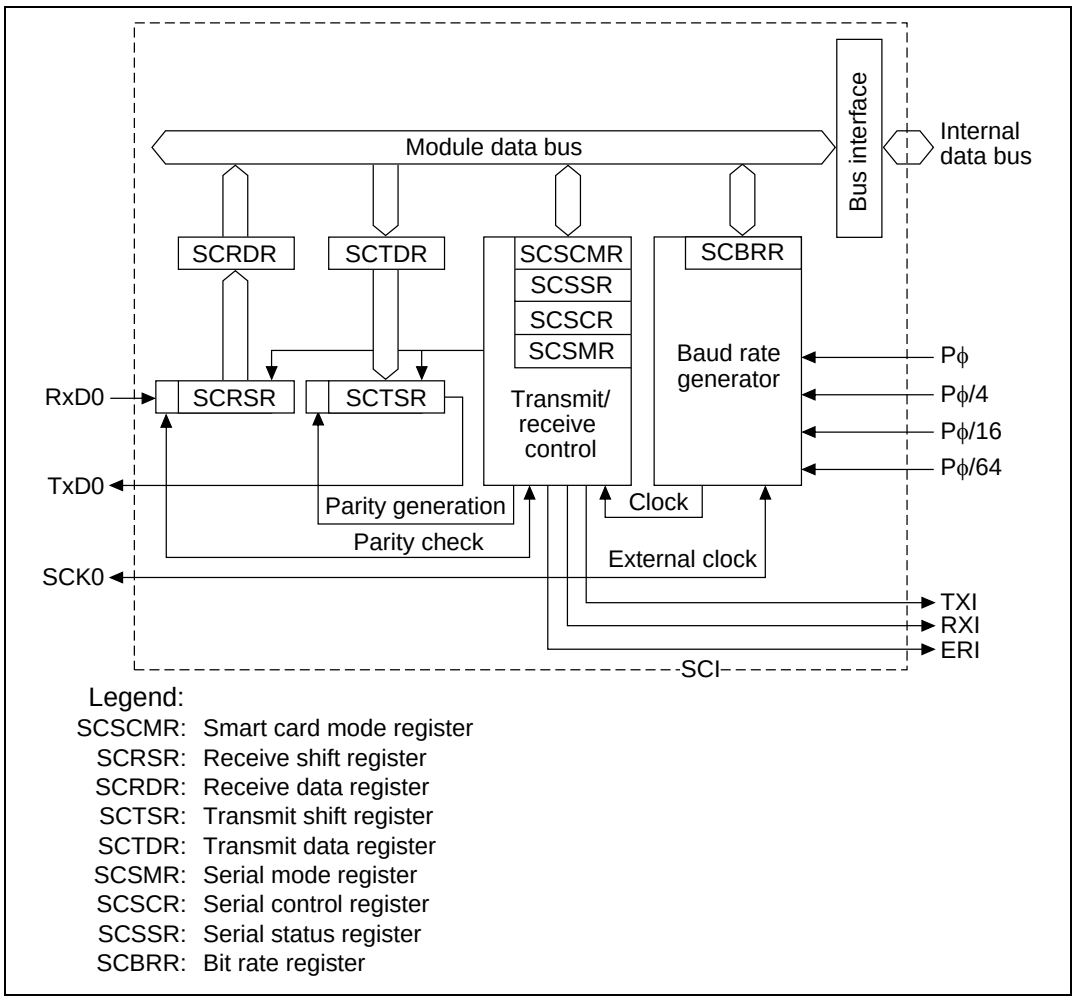


Figure 18.1 Smart Card Interface Block Diagram

18.1.3 Pin Configuration

Table 18.1 summarizes the smart card interface pins.

Table 18.1 SCI Pins

Pin Name	Abbreviation	I/O	Function
Serial clock pin	SCK0	Output	Clock output
Receive data pin	RxD0	Input	Receive data input
Transmit data pin	TxD0	Output	Transmit data output

18.1.4 Register Configuration

Table 18.2 summarizes the registers used by the smart card interface. The SCSMR, SCBRR, SCSCR, SCTDR, and SCRDR registers are the same as in the ordinary SCI function. They are described in section 17, Serial Communication Interface (SCI).

Table 18.2 Registers

Name	Abbreviation	R/W	Initial Value*3	Address	Access Size
Serial mode register	SCSMR	R/W	H'00	H'FFFFFE80	8
Bit rate register	SCBRR	R/W	H'FF	H'FFFFFE82	8
Serial control register	SCSCR	R/W	H'00	H'FFFFFE84	8
Transmit data register	SCTDR	R/W	H'FF	H'FFFFFE86	8
Serial status register	SCSSR	R/(W)*1	H'84	H'FFFFFE88	8
Receive data register	SCRDR	R	H'00	H'FFFFFE8A	8
Smart card mode register	SCSCMR	R/W	*2	H'FFFFFE8C	8

Notes: 1. Only 0 can be written, to clear the flags.
2. Bits 0, 2, and 3 are cleared. The value of the other bits is undefined.
3. Initialized by a power-on or manual reset.

18.2 Register Descriptions

This section describes the registers added for the smart card interface and the bits whose functions are changed.

18.2.1 Smart Card Mode Register (SCSCMR)

The smart card mode register (SCSCMR) is an 8-bit read/write register that selects smart card interface functions. SCSCMR bits 0, 2, and 3 are initialized to H'00 by a reset and in standby mode.

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	SDIR	SINV	—	SMIF
Initial value:	—	—	—	—	0	0	—	0
R/W:	R	R	R	R	R/W	R/W	R	R/W

Bits 7 to 4 and 1—Reserved: These bits are always read as 0. The write value should always be 0.

Bit 3—Smart Card Data Transfer Direction (SDIR): Selects the serial/parallel conversion format.

Bit 3: SDIR	Description
0	Contents of SCTDR are transferred as LSB first, receive data is stored in SCRDR as LSB first. (Initial value)
1	Contents of SCTDR are transferred as MSB first, receive data is stored in SCRDR as MSB first.

Bit 2—Smart Card Data Inversion (SINV): Specifies whether to invert the logic level of the data. This function is used in combination with bit 3 for transmitting and receiving with an inverse convention card. SINV does not affect the logic level of the parity bit. See section 18.3.4, Register Settings, for information on how parity is set.

Bit 2: SINV	Description
0	Contents of SCTDR are transferred unchanged, receive data is stored in SCRDR unchanged. (Initial value)
1	Contents of SCTDR are inverted before transfer, receive data is inverted before storage in SCRDR.

Bit 0—Smart Card Interface Mode Select (SMIF): Enables the smart card interface function.

Bit 0: SMIF	Description	
0	Smart card interface function disabled	(Initial value)
1	Smart card interface function enabled	

18.2.2 Serial Status Register (SCSSR)

In the smart card interface mode, the function of SCSSR bit 4 is changed. The setting conditions for bit 2, the TEND bit, are also changed.

Bit:	7	6	5	4	3	2	1	0
	TDRE	RDRF	ORER	FER/ERS	PER	TEND	MPB	MPBT
Initial value:	1	0	0	0	0	1	0	0
R/W:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R	R	R/W

Note: * Only 0 can be written, to clear the flag.

Bits 7 to 5—These bits have the same function as in the ordinary SCI. See section 17, Serial Communication Interface (SCI), for more information.

Bit 4—Error Signal Status (ERS): In the smart card interface mode, bit 4 indicates the state of the error signal returned from the receiving side during transmission. The smart card interface cannot detect framing errors.

Bit 4: ERS	Description	
0	Receiving ended normally with no error signal. ERS is cleared to 0 when the chip is reset or enters standby mode, or when software reads ERS after it has been set to 1, then writes 0 in ERS.	(Initial value)
1	An error signal indicating a parity error was transmitted from the receiving side. ERS is set to 1 if the error signal sampled is low.	

Note: The ERS flag maintains its state even when the TE bit in SCSCR is cleared to 0.

Bits 3 to 0—These bits have the same function as in the ordinary SCI. See section 17, Serial Communication Interface (SCI), for more information. The setting conditions for bit 2, the transmit end bit (TEND), are changed as follows.

Bit 2: TEND	Description
0	Transmission is in progress. TEND is cleared to 0 when software reads TDRE after it has been set to 1, then writes 0 in TDRE.
1	End of transmission. (Initial value) TEND is set to 1 when: • the chip is reset or enters standby mode, • the TE bit in SCSCR is 0 and the FER/ERS bit is also 0, • the C/$\overline{A}$ bit in SCSMR is 0, and TDRE = 1 and FER/ERS = 0 (normal transmission) 2.5 etu after a one-byte serial character is transmitted, or • the C/$\overline{A}$ bit in SCSMR is 1, and TDRE = 1 and FER/ERS = 0 (normal transmission) 1.0 etu after a one-byte serial character is transmitted.

Note: etu: Elementary Time Unit (time for transfer of 1 bit)

18.3 Operation

18.3.1 Overview

The primary functions of the smart card interface are described below.

1. Each frame consists of 8-bit data and 1 parity bit.
2. During transmission, the card leaves a guard time of at least 2 etu (elementary time units: time for transfer of 1 bit) from the end of the parity bit to the start of the next frame.
3. During reception, the card outputs an error signal low level for 1 etu after 10.5 etu has elapsed from the start bit if a parity error was detected.
4. During transmission, it automatically transmits the same data after allowing at least 2 etu from the time the error signal is sampled.
5. Only start-stop type asynchronous communication functions are supported; no synchronous communication functions are available.

18.3.2 Pin Connections

Figure 18.2 shows the pin connection diagram for the smart card interface. During communication with an IC card, transmission and reception are both carried out over the same data transfer line, so connect the TxD0 and RxD0 pins on the chip. Pull up the data transfer line to the power supply V_{CC} side with a resistor.

When using the clock generated by the smart card interface on an IC card, input the SCK pin output to the IC card's CLK pin. This connection is not necessary when the internal clock is used on the IC card.

Use the chip's port output as the reset signal. Apart from these pins, the power and ground pin connections are usually also required.

Note: When the IC card is not connected and both RE and TE are set to 1, closed communication is possible and auto-diagnosis can be performed.

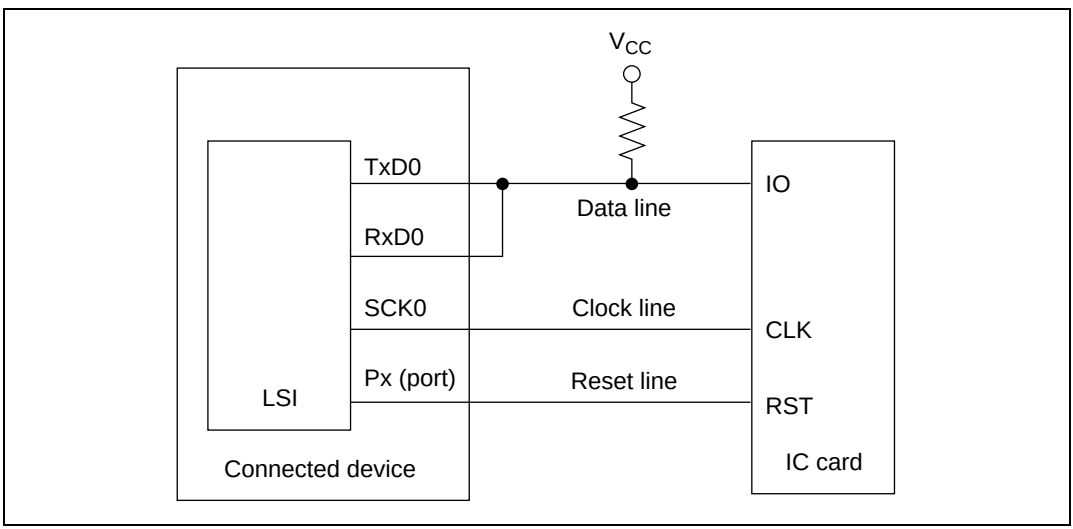


Figure 18.2 Pin Connection Diagram for the Smart Card Interface

18.3.3 Data Format

Figure 18.3 shows the data format for the smart card interface. In this mode, parity is checked every frame while receiving and error signals sent to the transmitting side whenever an error is detected so that data can be re-transmitted. During transmission, error signals are sampled and data re-transmitted whenever an error signal is detected.

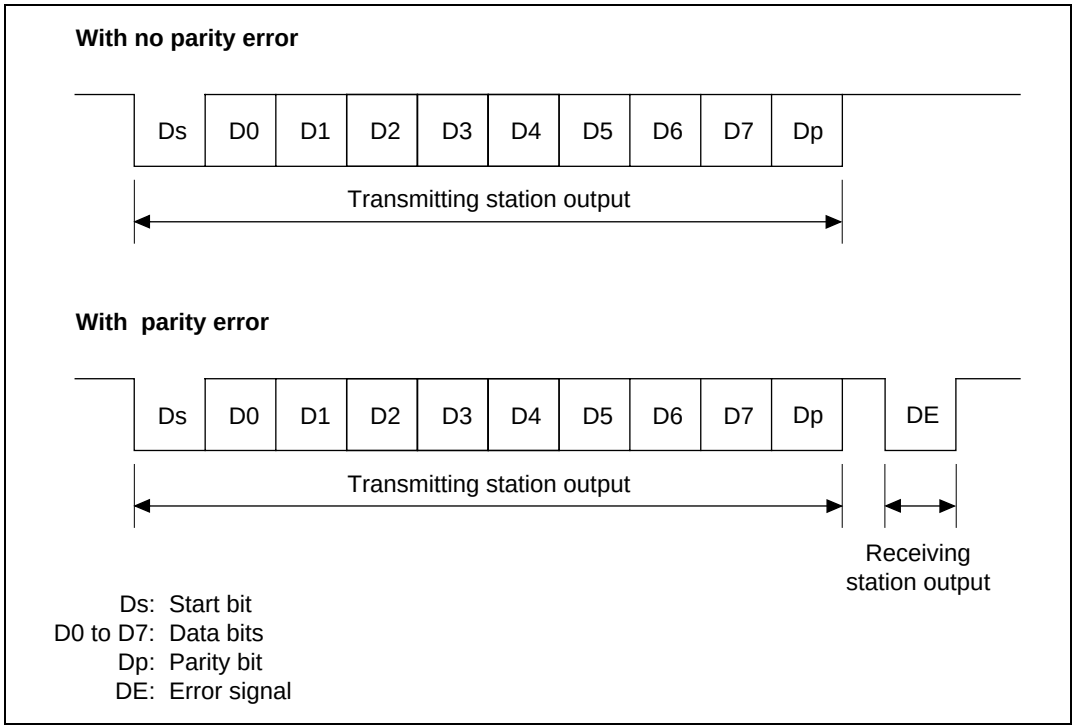


Figure 18.3 Data Format for Smart Card Interface

The operating sequence is:

1. The data line is high impedance when not in use and is fixed high with a pull-up resistor.
2. The transmitting side starts one frame of data transmission. The data frame starts with a start bit (Ds, low level). The start bit is followed by eight data bits (D0 to D7) and a parity bit (Dp).
3. On the smart card interface, the data line returns to high impedance after this. The data line is pulled high with a pull-up resistor.
4. The receiving side checks parity. When the data is received normally with no parity errors, the receiving side then waits to receive the next data. When a parity error occurs, the receiving side outputs an error signal (DE, low level) and requests re-transfer of data. The receiving station returns the signal line to high impedance after outputting the error signal for a specified period. The signal line is pulled high with a pull-up resistor.
5. The transmitting side transmits the next frame of data unless it receives an error signal. If it does receive an error signal, it returns to step 2 to re-transmit the erroneous data.

18.3.4 Register Settings

Table 18.3 shows the bit map of the registers that the smart card interface uses. Bits shown as 1 or 0 must be set to the indicated value. The settings for the other bits are described below.

Table 18.3 Register Settings for the Smart Card Interface

Register	Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SCSMR	H'FFFFFE80	C/$\overline{A}$	0	1	O/$\overline{E}$	1	0	CKS1	CKS0
SCBRR	H'FFFFFE82	BRR7	BRR6	BRR5	BRR4	BRR3	BRR2	BRR1	BRR0
SCSCR	H'FFFFFE84	TIE	RIE	TE	RE	0	0	CKE1	CKE0
SCTDR	H'FFFFFE86	TDR7	TDR6	TDR5	TDR4	TDR3	TDR2	TDR1	TDR0
SCSSR	H'FFFFFE88	TDRE	RDRF	ORER	FER/ ERS	PER	TEND	0	0
SCRDR	H'FFFFFE8A	RDR7	RDR6	RDR5	RDR4	RDR3	RDR2	RDR1	RDR0
SCSCMR	H'FFFFFE8C	—	—	—	—	SDIR	SINV	—	SMIF

Note: Dashes indicate unused bits.

1. Setting the serial mode register (SCSMR): The C/$\overline{A}$ bit selects the set timing of the TEND flag, and selects the clock output state with the combination of bits CKE1 and CKE0 in the serial control register (SCSCR). Set the O/$\overline{E}$ bit to 0 when the IC card uses the direct convention or to 1 when it uses the inverse convention. Select the on-chip baud rate generator clock source with the CKS1 and CKS0 bits (see section 18.3.5, Clock).
2. Setting the bit rate register (SCBRR): Set the bit rate. See section 18.3.5, Clock, to see how to calculate the set value.
3. Setting the serial control register (SCSCR): The TIE, RIE, TE and RE bits function as they do for the ordinary SCI0. See section 17, Serial Communication Interface (SCI), for more information. The CKE0 bit specifies the clock output. When no clock is output, set 0; when a clock is output, set 1.
4. Setting the smart card mode register (SCSCMR): The SDIR and SINV bits are both set to 0 for IC cards that use the direct convention and both to 1 when the inverse convention is used. The SMIF bit is set to 1 for the smart card interface.

   Figure 18.4 shows sample waveforms for register settings of the two types of IC cards (direct convention and inverse convention) and their start characters.

   In the direct convention type, the logical 1 level is state Z, the logical 0 level is state A, and communication is LSB first. The start character data is H'3B. The parity bit is even (from the smart card standards), and thus 1.

In the inverse convention type, the logical 1 level is state A, the logical 0 level is state Z, and communication is MSB first. The start character data is H'3F. The parity bit is even (from the smart card standards), and thus 0, which corresponds to state Z.

Only data bits D7 to D0 are inverted by the SINV bit. To invert the parity bit, set the O/$\overline{E}$ bit in SCSMR to odd parity mode. This applies to both transmission and reception.

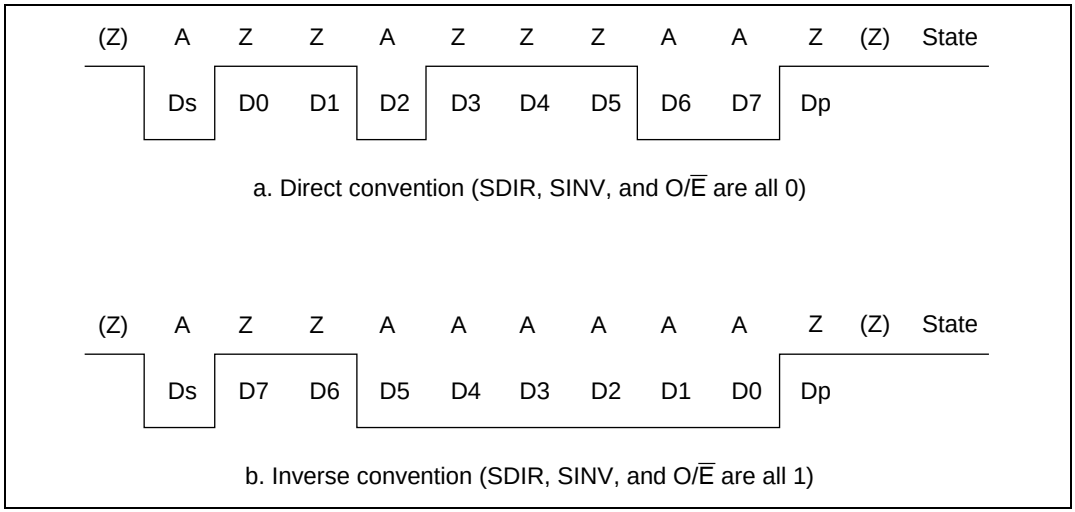


Figure 18.4 Waveform of Start Character

18.3.5 Clock

Only the internal clock generated by the on-chip baud rate generator can be used as the communication clock in the smart card interface. The bit rate for the clock is set by the bit rate register (SCBRR) and the CKS1 and CKS0 bits in the serial mode register (SCSMR), and is calculated using the equation below. Table 18.5 shows sample bit rates. If clock output is then selected by setting CKE0 to 1, a clock with a frequency 372 times the bit rate is output from the SCK0 pin.

$$B = \frac{P\phi}{1488 \times 2^{2n-1} \times (N + 1)} \times 10^6$$

Where: N = Value set in SCBRR ($0 \leq N \leq 255$)
B = Bit rate (bit/s)
Pϕ = Peripheral module operating frequency (MHz)
n = 0 to 3 (table 18.4)

Table 18.4 Relationship of n to CKS1 and CKS0

n	CKS1	CKS0
0	0	0
1	0	1
2	1	0
3	1	1

Table 18.5 Examples of Bit Rate B (Bit/s) for SCBRR Settings (n = 0)

	Pϕ (MHz)						
N	**7.1424**	**10.00**	**10.7136**	**13.00**	**14.2848**	**16.00**	**18.00**
0	9600.0	13440.9	14400.0	17473.1	19200.0	21505.4	24193.5
1	4800.0	6720.4	7200.0	8736.6	9600.0	10752.7	12096.8
2	3200.0	4480.3	4800.0	5824.4	6400.0	7168.5	8064.5

Note: The bit rate is rounded to two decimal places.

Calculate the value to be set in the bit rate register (SCBRR) from the operating frequency and the bit rate. N is an integer in the range $0 \le N \le 255$, specifying a smallish error.

$$N = \frac{P\phi}{1488 \times 2^{2n-1} \times B} \times 10^6 - 1$$

Table 18.6 Examples of SCBRR Settings for Bit Rate B (Bit/s) (n = 0)

	ϕ (MHz) (9600 Bits/s)											
7.1424		**10.00**		**10.7136**		**13.00**		**14.2848**		**16.00**		**18.00**	
N	**Error**	**N**	**Error**	**N**	**Error**	**N**	**Error**	**N**	**Error**	**N**	**Error**	**N**	**Error**
0	0.00	1	30.00	1	25.00	1	8.99	1	0.00	1	12.01	2	15.99

Table 18.7 Maximum Bit Rates for Frequencies (Smart Card Interface Mode)

Pϕ (MHz)	Maximum Bit Rate (Bit/s)	N	n
7.1424	9600	0	0
10.00	13441	0	0
10.7136	14400	0	0
13.00	17473	0	0
14.2848	19200	0	0
16.00	21505	0	0
18.00	24194	0	0

The bit rate error is found as follows:

$$\text{Error (\%)} = \left(\frac{P\phi}{1488 \times 2^{2n-1} \times B \times (N + 1)} \times 10^6 - 1\right) \times 100$$

Table 18.8 shows the relationship between transmit/receive clock register set values and output states on the smart card interface.

Table 18.8 Register Set Values and SCK Pin

	Register Value				SCK Pin	
Setting	**SMIF**	**C/$\overline{A}$**	**CKE1**	**CKE0**	**Output**	**State**
1*1	1	0	0	0	Port	Determined by setting of port register SCP1MD1 and SCP1MD0 bits
	1	0	0	1	⎍⎍⎍	SCK (serial clock) output state
2*2	1	1	0	0	Low output	Low output state
	1	1	0	1	⎍⎍⎍	SCK (serial clock) output state
3*2	1	1	1	0	High output	High output state
	1	1	1	1	⎍⎍⎍	SCK (serial clock) output state

Notes: 1. The SCK output state changes as soon as the CKE0 bit is modified. The CKE1 bit should be cleared to 0.
2. The clock duty remains constant despite stopping and starting of the clock by modification of the CKE0 bit.

18.3.6 Data Transmission and Reception

Initialization: Initialize the SCI0 using the following procedure before sending or receiving data. Initialization is also required for switching from transmit mode to receive mode or from receive mode to transmit mode. Figure 18.5 shows a flowchart of the initialization process (example).

1. Clear TE and RE in the serial control register (SCSCR) to 0.
2. Clear error flags FER/ERS, PER, and ORER to 0 in the serial status register (SCSSR).
3. Set the C/$\overline{A}$ bit, parity bit (O/$\overline{E}$ bit), and baud rate generator select bits (CKS1 and CKS0 bits) in the serial mode register (SCSMR). At this time also clear the CHR and MP bits to 0 and set the STOP and PE bits to 1.
4. Set the SMIF, SDIR, and SINV bits in the smart card mode register (SCSCMR). When the SMIF bit is set to 1, the TxD and RxD pins both switch from ports to SCI0 pins and become high impedance.
5. Set the value corresponding to the bit rate in the bit rate register (SCBRR).
6. Set the clock source select bits (CKE1 and CKE0 bits) in the serial control register (SCSCR). Clear the TIE, RIE, TE, RE, MPIE, and TEIE bits to 0. When the CKE0 bit is set to 1, a clock is output from the SCK pin.
7. After waiting at least 1 bit, set the TIE, RIE, TE, and RE bits in SCSCR. Do not set the TE and RE bits simultaneously unless performing auto-diagnosis.

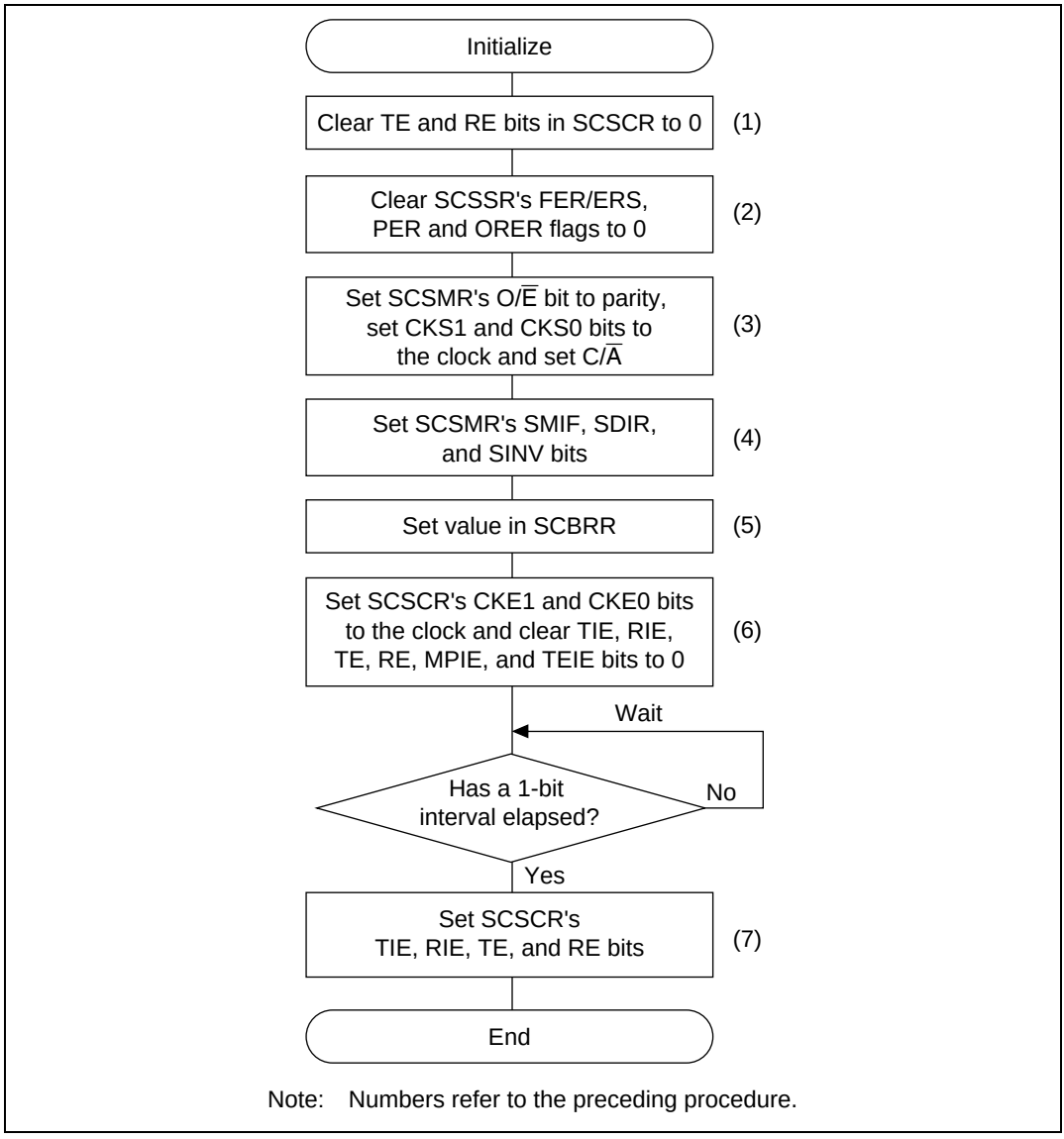


Figure 18.5 Initialization Flowchart (Example)

Serial Data Transmission: The processing procedures in the smart card mode differ from ordinary SCI processing because data is retransmitted when an error signal is sampled during a data transmission. This results in the transmission processing flowchart shown in figure 18.6 (example).

1. Initialize the smart card interface mode as described in initialization above.
2. Check that the FER/ERS bit in SCSSR is cleared to 0.
3. Repeat steps 2 and 3 until the TEND flag in SCSSR is set to 1.
4. Write the transmit data into SCTDR, clear the TDRE flag to 0 and start transmitting. The TEND flag will be cleared to 0.
5. To transmit more data, return to step 2.
6. To end transmission, clear the TE bit to 0.

This processing can be interrupted. When the TIE bit is set to 1 and interrupt requests are enabled, a transmit-data-empty interrupt (TXI) will be requested when the TEND flag is set to 1 at the end of the transmission. When the RIE bit is set to 1 and interrupt requests are enabled, a communication error interrupt (ERI) will be requested when the ERS flag is set to 1 when an error occurs in transmission. See Interrupt Operation below for more information.

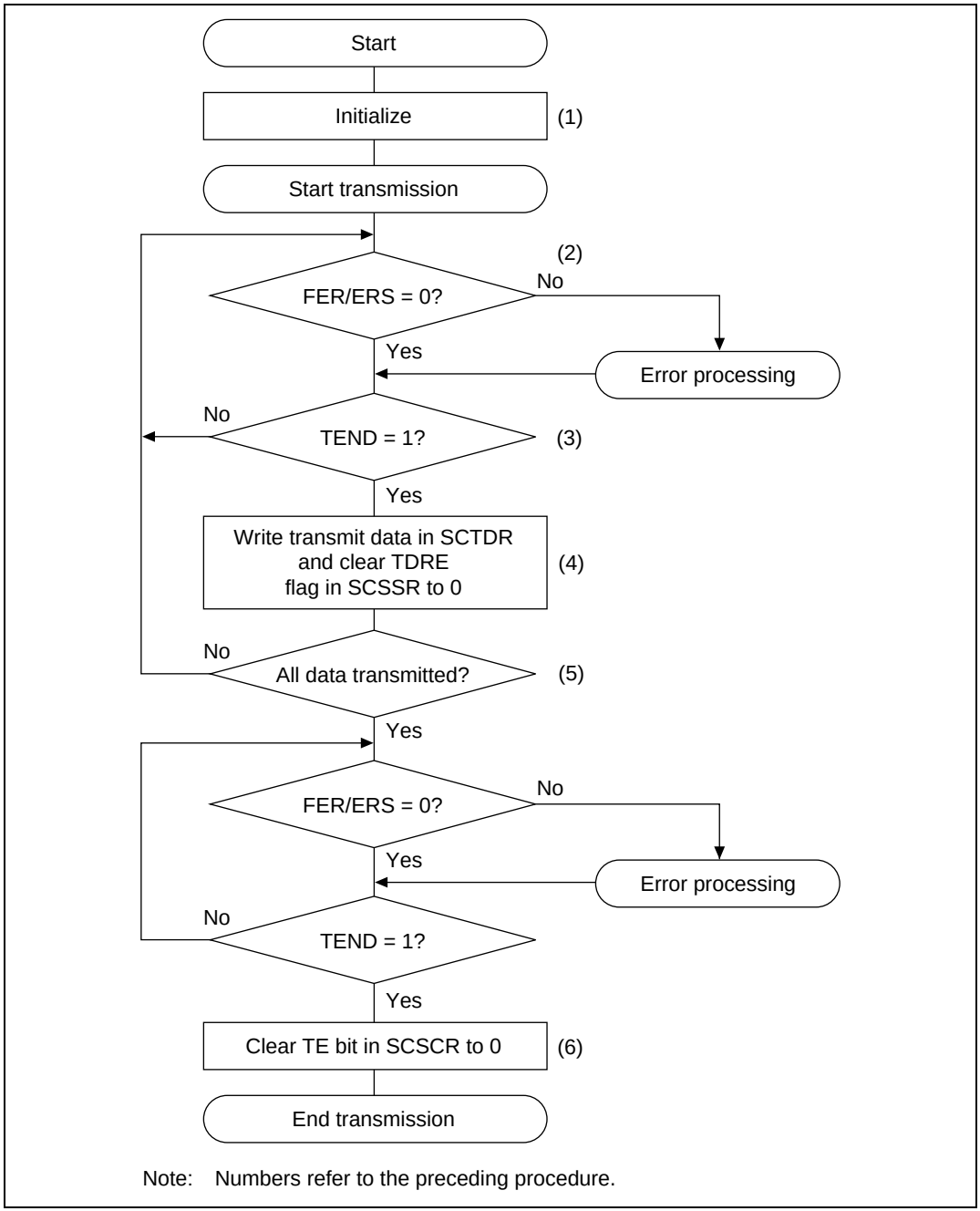


Figure 18.6 Transmission Flowchart (Example)

Serial Data Reception: The processing procedures in the smart card mode are the same as in ordinary SCI processing. The reception processing flowchart is shown in figure 18.7 (example).

1. Initialize the smart card interface mode as described above in Initialization and in figure 18.5.
2. Check that the ORER and PER flags in SCSSR are cleared to 0. If either flag is set, clear both to 0 after performing the appropriate error processing procedures.
3. Repeat steps 2 and 3 until the RDRF flag is set to 1.
4. Read the receive data from SCRDR.
5. To receive more data, clear the RDRF flag to 0 and return to step 2.
6. To end reception, clear the RE bit to 0.

This processing can be interrupted. When the RIE bit is set to 1 and interrupt requests are enabled, a receive-data-full interrupt (RXI) will be requested when the RDRF flag is set to 1 at the end of the reception. When an error occurs during reception and either the ORER or PER flag is set to 1, a communication error interrupt (ERI) will be requested. See Interrupt Operation below for more information.

The received data will be transferred to SCRDR even when a parity error occurs during reception and PER is set to 1, so this data can still be read.

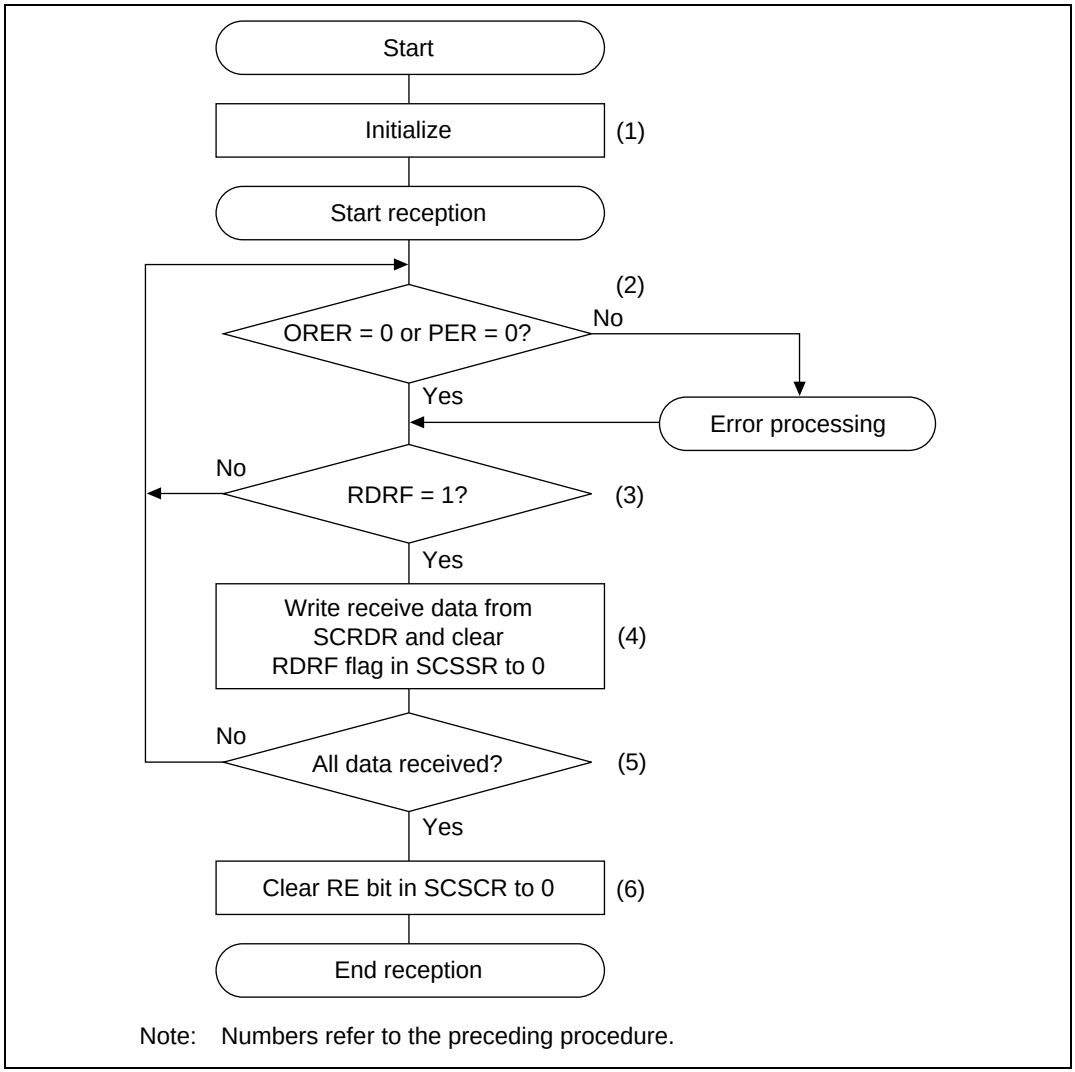


Figure 18.7 Reception Flowchart (Example)

Switching Modes: When switching from receive mode to transmit mode, check that the receive operation is completed before starting initialization and setting RE to 0 and TE to 1. The RDRF, PER, and ORER flags can be used to check if reception is completed. When switching from transmit mode to receive mode, check that the transmit operation is completed before starting initialization and setting TE to 0 and RE to 1. The TEND flag can be used to check if transmission is completed.

Interrupt Operation: In the smart card interface mode, there are three types of interrupts: transmit-data-empty (TXI), communication error (ERI) and receive-data-full (RXI). In this mode, the transmit-end interrupt (TEI) cannot be requested.

Set the TEND flag in SCSSR to 1 to request a TXI interrupt. Set the RDRF flag in SCSSR to 1 to request an RXI interrupt. Set the ORER, PER, or FER/ERS flag in SCSSR to 1 to request an ERI interrupt (table 18.9).

Table 18.9 Smart Card Mode Operating State and Interrupt Sources

Mode	State	Flag	Mask Bit	Interrupt Source
Transmit mode	Normal	TEND	TIE	TXI
	Error	FER/ERS	RIE	ERI
Receive mode	Normal	RDRF	RIE	RXI
	Error	PER, ORER	RIE	ERI

18.4 Usage Notes

When the SCI is used as a smart card interface, be sure that all criteria in sections 18.4.1, Receive Data Timing and Receive Margin in Asynchronous Mode and 18.4.2, Retransmission (Receive and Transmit Modes) are applied.

18.4.1 Receive Data Timing and Receive Margin in Asynchronous Mode

In asynchronous mode, the SCI runs on a basic clock with a frequency of 372 times the transfer rate. During reception, the SCI0 samples the falling of the start bit using the base clock to achieve internal synchronization. Receive data is latched internally on the rising edge of the 186th basic clock cycle (figure 18.8).

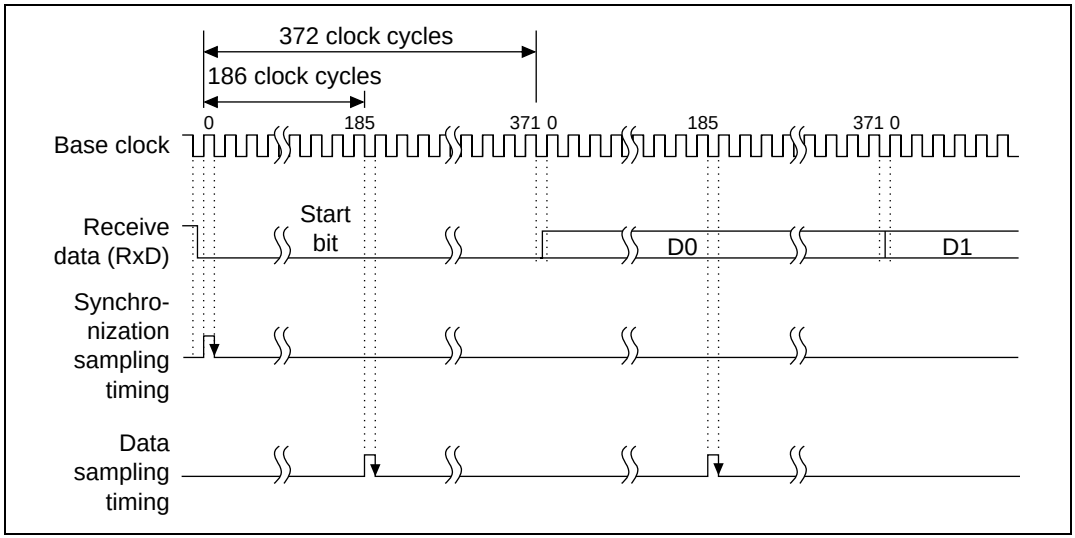


Figure 18.8 Receive Data Sampling Timing in Smart Card Mode

The receive margin is found from the following equation:

For smart card mode:

$$M = \left| (0.5 - \frac{1}{2N}) - (L - 0.5)F - \frac{|D - 0.5|}{N}(1 + F) \right| \times 100\%$$

Where: M = Receive margin (%)
N = Ratio of bit rate to clock (N = 372)
D = Clock duty (D = 0 to 1.0)
L = Frame length (L = 10)
F = Absolute value of clock frequency deviation

Using this equation, the receive margin when F = 0 and D = 0.5 is as follows:

$$M = (0.5 - 1/2 \times 372) \times 100\% = 49.866\%$$

18.4.2 Retransmission (Receive and Transmit Modes)

Retransmission by the SCI in Receive Mode: Figure 18.9 shows the retransmission operation in the SCI receive mode.

1. When the received parity bit is checked and an error is found, the PER bit in SCSSR is automatically set to 1. If the RIE bit in SCSCR is enabled at this time, an ERI interrupt is requested. Be sure to clear the PER bit before the next parity bit is sampled.
2. The RDRF bit in SCSSR is not set in the frame that caused the error.
3. When the received parity bit is checked and no error is found, the PER bit in SCSSR is not set.
4. When the received parity bit is checked and no error is found, reception is considered to have been completed normally and the RDRF bit in SCSSR is automatically set to 1. If the RIE bit in SCSCR is enabled at this time, an RXI interrupt is requested.
5. When a normal frame is received, the pin maintains a three-state state when it transmits the error signal.

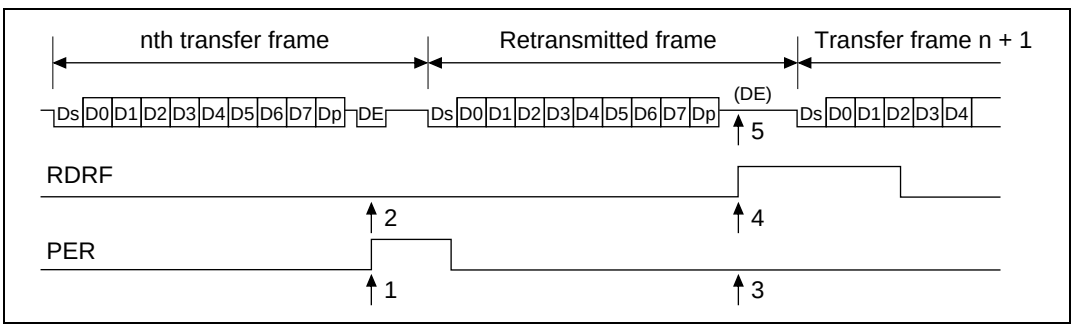


Figure 18.9 Retransmission in SCI Receive Mode

Retransmission by the SCI in Transmit Mode: Figure 18.10 shows the retransmission operation in the SCI transmit mode.

1. After transmission of one frame is completed, the FER/ERS bit in SCSSR is set to 1 when a error signal is returned from the receiving side. If the RIE bit in SCSCR is enabled at this time, an ERI interrupt is requested. Be sure to clear the FER/ERS bit before the next parity bit is sampled.
2. The TEND bit in SCSSR is not set in the frame that received the error signal that indicated the error.
3. The FER/ERS bit in SCSSR is not set when no error signal is returned from the receiving side.
4. When no error signal is returned from the receiving side, the TEND bit in SCSSR is set to 1 when the transmission of the frame that includes the retransmission is considered completed. If the TIE bit in SCSCR is enabled at this time, a TXI interrupt will be requested.

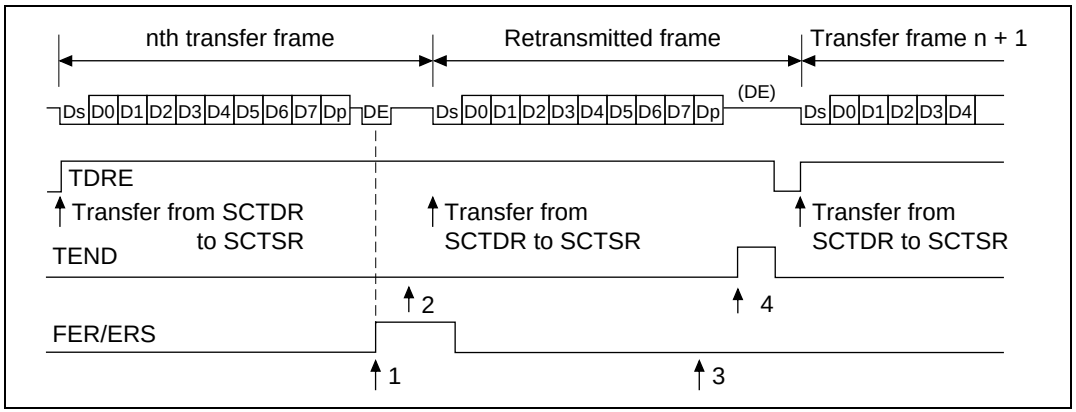


Figure 18.10 Retransmission in SCI Transmit Mode

Section 19 Serial Communication Interface with FIFO (SCIF)

19.1 Overview

This LSI has one-channel serial communication interface with FIFO (SCIF) that supports asynchronous serial communication. It also has 16-stage FIFO registers for both transfer and receive that enables this LSI efficient high-speed continuous communication.

19.1.1 Features

- Asynchronous serial communication:
  - Serial data communications are performed by start-stop in character units. The SCI can communicate with a universal asynchronous receiver/transmitter (UART), an asynchronous communication interface adapter (ACIA), or any other communications chip that employs a standard asynchronous serial system. There are eight selectable serial data communication formats.
  - Data length: Seven or eight bits
  - Stop bit length: One or two bits
  - Parity: Even, odd, or none
  - Receive error detection: Parity and framing errors
  - Break detection: Break is detected when the receive data next the generated framing error is the space 0 level and has the framing error. It is also detected by reading the RxD level directly from the port SC data register (SCPDR) when a framing error occurs
- Full duplex communication: The transmitting and receiving sections are independent, so the SCI can transmit and receive simultaneously. Both sections use 16-stage FIFO buffering, so high-speed continuous data transfer is possible in both the transmit and receive directions.
- On-chip baud rate generator with selectable bit rates
- Internal transmit/receive clock source
- Four types of interrupts: Transmit-FIFO-data-empty, break, receive-FIFO-data-full, and receive-error interrupts are requested independently. The direct memory access controller (DMAC) can be activated to execute a data transfer by a transmit-FIFO-data-empty or receive-FIFO-data-full interrupt.
- When the SCIF is not in use, it can be stopped by halting the clock supplied to it, saving power.
- On-chip modem control functions ($\overline{\text{RTS2}}$ and $\overline{\text{CTS2}}$)

- The quantity of data in the transmit and receive FIFO registers and the number of receive errors of the receive data in the receive FIFO register can be known.
- The time-out error (DR) can be detected in receiving.

19.1.2 Block Diagram

Figure 19.1 shows a block diagram of the SCIF.

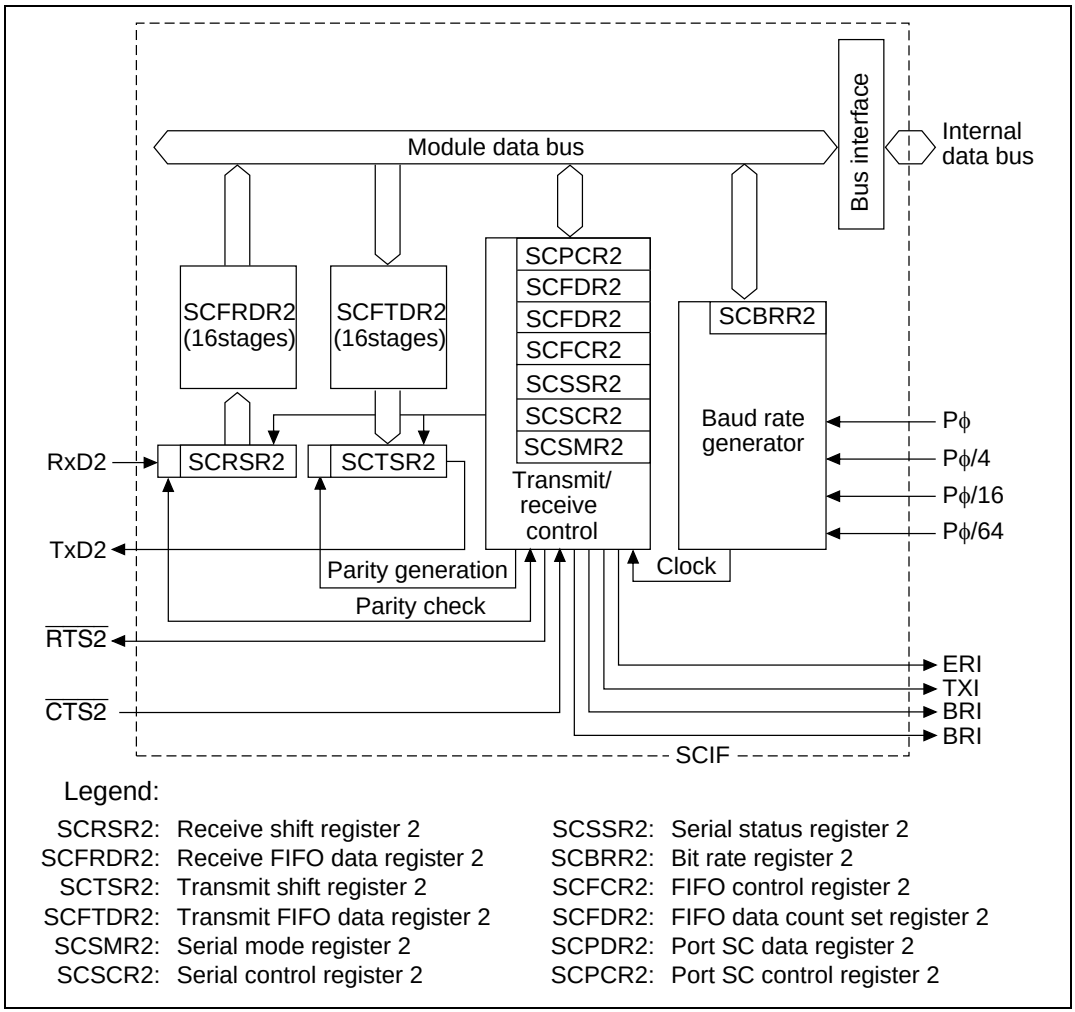


Figure 19.1 SCIF Block Diagram

Figures 19.2 and 19.3 show SCIF I/O ports. Bits 15, 14, 9, 8 of SCPCR and bits 7 and 4 of SCPDR control an input/output and data of the SCIF pins. See section 26.3.13, SC Port Control Register (SCPCR) for more details.

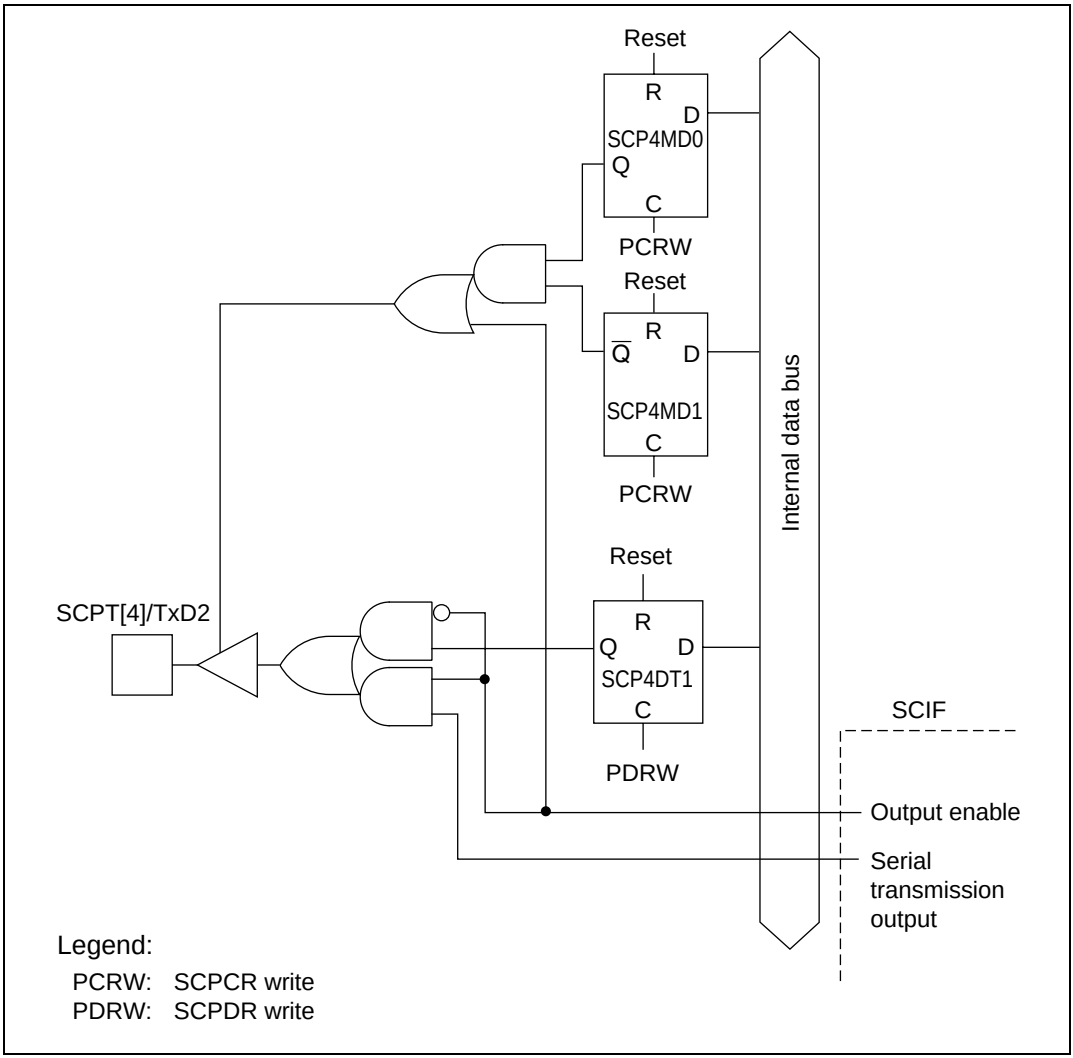


Figure 19.2 SCPT[4]/TxD2 Pin

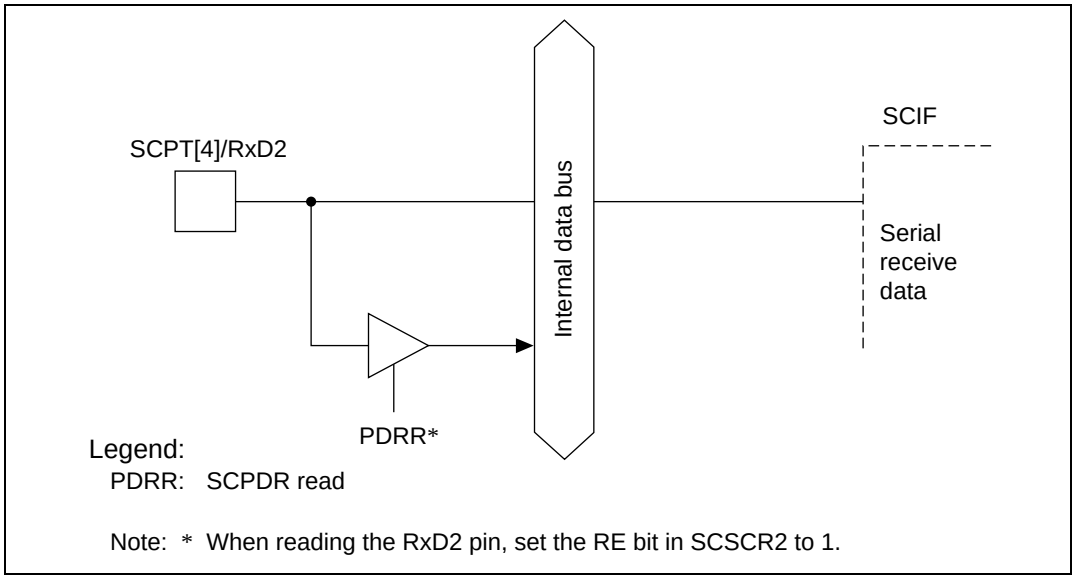


Figure 19.3 SCPT[4]/RxD2 Pin

19.1.3 Pin Configuration

The SCIF has the serial pins summarized in table 19.1.

Table 19.1 SCIF Pins

Pin Name	Abbreviation	I/O	Function
Receive data pin	RxD2	Input	Receive data input
Transmit data pin	TxD2	Output	Transmit data output
Request to send pin	$\overline{\text{RTS2}}$	Output	Request to send
Clear to send pin	$\overline{\text{CTS2}}$	Input	Clear to send

19.1.4 Register Configuration

Table 19.2 summarizes the SCIF internal registers. These registers specify the data format and bit rate, and control the transmitter and receiver sections.

Table 19.2 Registers

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Serial mode register 2	SCSMR2	R/W	H'00	H'04000150 (H'A4000150)*2	8 bits
Bit rate register 2	SCBRR2	R/W	H'FF	H'04000152 (H'A4000152)*2	8 bits
Serial control register 2	SCSCR2	R/W	H'00	H'04000154 (H'A4000154)*2	8 bits
Transmit FIFO data register 2	SCFTDR2	W	—	H'04000156 (H'A4000156)*2	8 bits
Serial status register 2	SCSSR2	R/(W)*1	H'0060	H'04000158 (H'A4000158)*2	16 bits
Receive FIFO data register 2	SCFRDR2	R	Undefined	H'0400015A (H'A400015A)*2	8 bits
FIFO control register 2	SCFCR2	R/W	H'00	H'0400015C (H'A400015C)*2	8 bits
FIFO data count set register 2	SCFDR2	R	H'0000	H'0400015E (H'A400015E)*2	16 bits

Notes: These registers are located in area 1 of physical space. Therefore, when the cache is on, either access these registers from the P2 area of logical space or else make an appropriate setting using the MMU so that these registers are not cached.

1. Only 0 can be written to clear the flag.
2. When address translation by the MMU does not apply, the address in parentheses should be used.

19.2 Register Descriptions

19.2.1 Receive Shift Register 2 (SCRSR2)

The receive shift register 2 (SCRSR2) receives serial data. Data input at the RxD2 pin is loaded into the SCRSR2 in the order received, LSB (bit 0) first, converting the data to parallel form. When one byte has been received, it is automatically transferred to the SCFRDR2, which is a receive FIFO data register 2. The CPU cannot read or write the SCRSR2 directly.

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

19.2.2 Receive FIFO Data Register 2 (SCFRDR2)

The 16-byte receive FIFO data register 2 (SCFRDR2) stores serial receive data. The SCIF completes the reception of one byte of serial data by moving the received data from the receive shift register 2 (SCRSR2) into the SCFRDR2 for storage. Continuous receive is enabled until 16 bytes are stored.

The CPU can read but not write the SCFRDR2. When data is read without received data in the SCFRDR2, the value is undefined. When the received data in this register becomes full, the subsequent serial data is lost.

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

19.2.3 Transmit Shift Register 2 (SCTSR2)

The transmit shift register 2 (SCTSR2) transmits serial data. The SCI loads transmit data from the transmit FIFO data register 2 (SCFTDR2) into the SCTSR2, then transmits the data serially from the TxD2 pin, LSB (bit 0) first. After transmitting one data byte, the SCI automatically loads the next transmit data from the SCFTDR2 into the SCTSR2 and starts transmitting again. The CPU cannot read or write the SCTSR2 directly.

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

19.2.4 Transmit FIFO Data Register 2 (SCFTDR2)

The transmit FIFO data register 2 (SCFTDR2) is a 16-byte 8-bit-length FIFO register that stores data for serial transmission. When the SCIF detects that the transmit shift register (SCTSR2) is empty, it moves transmit data written in the SCFTDR2 into the SCTSR2 and starts serial transmission. Continuous serial transmission is performed until the transmit data in the SCFTDR2 becomes empty. The CPU can always write to the SCFTDR2.

When the transmit data in the SCFTDR2 is full (16 bytes), next data cannot be written. If attempted to write, the data is ignored.

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

19.2.5 Serial Mode Register 2 (SCSMR2)

The serial mode register 2 (SCSMR2) is an eight-bit register that specifies the SCIF serial communication format and selects the clock source for the baud rate generator.

The CPU can always read and write the SCSMR2. The SCSMR2 is initialized to H'00 by a reset or in standby and module standby modes.

Bit:	7	6	5	4	3	2	1	0
	—	CHR	PE	O/$\overline{E}$	STOP	—	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 7—Reserved: This bit always read 0. The write value should always be 0.

Bit 6—Character Length (CHR): Selects seven-bit or eight-bit data in the asynchronous mode.

Bit 6: CHR	Description	
0	Eight-bit data.	(Initial value)
1	Seven-bit data. *	

Note: * When seven-bit data is selected, the MSB (bit 7) of the transmit FIFO data register 2 is not transmitted.

Bit 5—Parity Enable (PE): Selects whether or not to add a parity bit to transmit data and to check the parity of receive data.

Bit 5: PE	Description	
0	Parity bit not added or checked.	(Initial value)
1	Parity bit added and checked. When PE is set to 1, an even or odd parity bit is added to transmit data, depending on the parity mode (O/$\overline{E}$) setting. Receive data parity is checked according to the even/odd (O/$\overline{E}$) mode setting.	

Bit 4—Parity Mode (O/$\overline{E}$): Selects even or odd parity when parity bits are added and checked. The O/$\overline{E}$ setting is used only when the parity enable bit (PE) is set to 1 to enable parity addition and check. The O/$\overline{E}$ setting is ignored when parity addition and check is disabled.

Bit 4: O/$\overline{E}$	Description
0	Even parity. (Initial value) If even parity is selected, the parity bit is added to transmit data to make an even number of 1s in the transmitted character and parity bit combined. Receive data is checked to see if it has an even number of 1s in the received character and parity bit combined.
1	Odd parity. If odd parity is selected, the parity bit is added to transmit data to make an odd number of 1s in the transmitted character and parity bit combined. Receive data is checked to see if it has an odd number of 1s in the received character and parity bit combined.

Bit 3—Stop Bit Length (STOP): Selects one or two bits as the stop bit length.

In receiving, only the first stop bit is checked, regardless of the STOP bit setting. If the second stop bit is 1, it is treated as a stop bit, but if the second stop bit is 0, it is treated as the start bit of the next incoming character.

Bit 3: STOP	Description
0	One stop bit. (Initial value) In transmitting, a single bit of 1 is added at the end of each transmitted character.
1	Two stop bits. In transmitting, two bits of 1 are added at the end of each transmitted character.

Bit 2—Reserved: This bit is always read as 0. The write value should always be 0.

Bits 1 and 0—Clock Select 1 and 0 (CKS1 and CKS0): These bits select the internal clock source of the on-chip baud rate generator. Four clock sources are available. Pϕ, Pϕ/4, Pϕ/16 and Pϕ/64. For further information on the clock source, bit rate register 2 settings, and baud rate, see section 19.2.8, Bit Rate Register 2 (SCBRR2).

Bit 1: CKS1	Bit 0: CKS0	Description
0	0	Pϕ clock (Initial value)
	1	Pϕ/4 clock
1	0	Pϕ/16 clock
	1	Pϕ/64 clock

Note: Pϕ: Peripheral clock

19.2.6 Serial Control Register 2 (SCSCR2)

The serial control register 2 (SCSCR2) operates the SCI transmitter/receiver, selects the serial clock output in the asynchronous mode, enables/disables interrupt requests, and selects the transmit/receive clock source. The CPU can always read and write the SCSCR2. The SCSCR2 is initialized to H'00 by a reset or in standby and module standby modes.

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

Bit 7—Transmit Interrupt Enable (TIE): Enables or disables the transmit-FIFO-data-empty interrupt (TXI) requested when serial transmit data is transferred from transmit FIFO data register 2 (SCFTDR2) to transmit shift register 2 (SCTSR2), when the quantity of data in transmit FIFO register 2 becomes less than the specified number of transmission triggers, and when the TDFE flag in serial status register 2 (SCSSR2) is set to1.

Bit 7: TIE	Description
0	Transmit-FIFO-data-empty interrupt request (TXI) is disabled.* (Initial value)
1	Transmit-FIFO-data-empty interrupt request (TXI) is enabled

Note: * The TXI interrupt request can be cleared by writing the greater quantity of transmit data than the specified number of transmission triggers to SCFTDR2 and by clearing TDFE to 0 after reading 1 from TDFE, or can be cleared by clearing TIE to 0.

Bit 6—Receive Interrupt Enable (RIE): Enables or disables the receive-data-full (RXI) and receive-error (ERI) interrupts requested when serial receive data is transferred from receive shift register 2 (SCRSR2) to receive FIFO data register 2 (SCFRDR2), when the quantity of data in receive FIFO register 2 becomes more than the specified number of receive triggers, and when the RDRF flag in SCSSR2 is set to1.

Bit 6: RIE	Description
0	Receive-data-full interrupt (RXI), receive-error interrupt (ERI), and receive break interrupt (BRI) requests are disabled.* (Initial value)
1	Receive-data-full interrupt (RXI) and receive-error interrupt (ERI) requests are enabled.

Note: * RXI and ERI interrupt requests can be cleared by reading the DR, ER, or RDF flag after it has been set to 1, then clearing the flag to 0, or by clearing RIE to 0. At RDF, read 1 from the RDF flag and clear it to 0, after reading the received data from SCFRDR2 until the quantity of received data becomes less than the specified number of the receive triggers.

Bit 5—Transmit Enable (TE): Enables or disables the SCIF serial transmitter.

Bit 5: TE	Description	
0	Transmitter disabled.	(Initial value)
1	Transmitter enabled. *	

Note: * Serial transmission starts after writing of transmit data into the SCFTDR2. Select the transmit format in the SCSMR2 and SCFCR2 and reset the SCFTDR2 before setting TE to 1.

Bit 4—Receive Enable (RE): Enables or disables the SCIF serial receiver.

Bit 4: RE	Description	
0	Receiver disabled.*1	(Initial value)
1	Receiver enabled.*2	

Notes: 1. Clearing RE to 0 does not affect the receive flags (DR, ER, BRK, FER, and PER). These flags retain their previous values.
2. Serial reception starts when a start bit is detected. Select the receive format in the SCSMR2 before setting RE to 1.

Bits 3 and 2—Reserved: These bits are always read as 0. The write value should always be 0.

Bits 1 and 0—Clock Enable 1 and 0 (CKE1 and CKE0): These bits should always be set to 00.

19.2.7 Serial Status Register 2 (SCSSR2)

Serial status register 2 (SCSSR2) is a 16-bit register. The upper 8 bits indicate the number of receive errors in the data of receive FIFO data register 2, and the lower 8 bits indicate SCIF operating state.

The CPU can always read and write the SCSSR2, but cannot write 1 in the status flags (ER, TEND, TDFE, BRK, RDF, and DR). These flags can be cleared to 0 only if they have first been read (after being set to 1). Bits 3 (FER) and 2 (PER) are read-only bits that cannot be written. The SCSSR2 is initialized to H'0060 by a reset or in standby and module standby modes.

Lower 8 bits:	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)*

Note: * The only value that can be written is 0 to clear the flag.

Bit 7—Receive Error (ER): Indicates that a parity error has occurred when received data includes a framing error or a parity.

Bit 7: ER	Description
0	Receive is in progress, or receive is normally completed.*1 (Initial value) ER is cleared to 0 when the chip is reset or enters standby mode, or when 0 is written after 1 is read from ER.
1	A framing error or a parity error has occurred. ER is set to 1 when the stop bit is 0 after checking whether or not the last stop bit of the received data is 1 at the end of one-data receive*2, or when the total number of 1's in the received data and in the parity bit does not match the even/odd parity specification specified by the O/$\overline{E}$ bit of the SCSMR2.

Notes: 1. Clearing the RE bit to 0 in SCSCR2 does not affect the ER bit, which retains its previous value. Even if a receive error occurs, the received data is transferred to SCFRDR2 and the receive operation is continued. Whether or not the data read from SCFRDR2 includes a receive error can be detected by the FER and PER bits of SCSSR2.

2. In the stop mode, only the first stop bit is checked; the second stop bit is not checked.

Bit 6—Transmit End (TEND): Indicates that when the last bit of a serial character was transmitted, the SCFTDR2 did not contain valid data, so transmission has ended.

Bit 6: TEND	Description
0	Transmission is in progress. TEND is cleared to 0 when data is written in SCFTDR2.
1	End of transmission. (Initial value) TEND is set to 1 when the chip is reset or enters standby mode, TE is cleared to 0 in the serial control register (SCSCR2), or when SCFTDR2 does not contain received data when the last bit of a one-byte serial character is transmitted.

Bit 5—Transmit FIFO Data Empty (TDFE): Indicates that data is transferred from transmit FIFO data register 2 (SCFTDR2) to transmit shift register (SCTSR), the quantity of data in SCFTDR2 becomes less than the number of transmission triggers specified by the TTRG1 and TTRG0 bits in FIFO control register 2 (SCFCR2), and writing the transmit data to SCFTDR2 is enabled.

Bit 5: TDFE	Description
0	The quantity of transmit data written to SCFTDR2 is greater than the specified number of transmission triggers. (Initial value) TDFE is cleared to 0 when data exceeding the specified transmission trigger number is written to SCFTDR2, and software reads 1 from TDFE and then writes 0 to TDFE.
1	The quantity of transmit data in SCFTDR2 is less than the specified number of transmission triggers.* TDFE is set to 1 at reset or at standby mode, or when the quantity of transmission data in SCFTDR2 becomes less than the specified number of transmission triggers as a result of transmission.

Note: * Since SCFTDR2 is a 16-byte FIFO register, the maximum quantity of data which can be written when TDFE is 1 is “16 minus the specified number of transmission triggers.” If an attempt is made to write additional data, the data is ignored. The quantity of data in SCFTDR2 is indicated by the upper 8 bits of SCFTDR2.

Bit 4—Break Detection (BRK): Indicates that a break signal is detected in received data.

Bit 4: BRK	Description
0	No break signal is being received. (Initial value) BRK is cleared to 0 when the chip is reset or enters standby mode, or software reads BRK after it has been set to 1, then writes 0 in BRK.
1	The break signal is received.* BRK is set to 1 when data including a framing error is received and a framing error occurs with space 0 in the subsequent received data.

Note: * When a break is detected, transfer of the received data (H'00) to SCFRDR2 stops after detection. When the break ends and the receive signal becomes mark 1, the transfer of the received data resumes. The received data of a frame in which a break signal is detected is transferred to SCFRDR2. After this, however, no received data is transferred until a break ends with the received signal being mark 1 and the next data is received.

Bit 3—Framing Error (FER): Indicates a framing error in the data read from the receive FIFO data register 2 (SCFRDR2).

Bit 3: FER	Description
0	No receive framing error occurred in the data read from SCFRDR2. (Initial value) FER is cleared to 0 when the chip is power-on reset or enters standby mode, or when no framing error is present in the data read from SCFRDR2.
1	A receive framing error occurred in the data read from SCFRDR2. FER is set to 1 when a framing error is present in the data read from SCFRDR2.

Bit 2—Parity Error (PER): Indicates a parity error in the data read from the receive FIFO data register 2 (SCFRDR2).

Bit 2: PER	Description
0	No receive parity error occurred in the data read from SCFRDR2. (Initial value) PER is cleared to 0 when the chip is power-on reset or enters standby mode, or when no parity error is present in the data read from SCFRDR2.
1	A receive parity error occurred in the data read from SCFRDR2. PER is set to 1 when a parity error is present in the data read from SCFRDR2.

Bit 1—Receive FIFO Data Full (RDF): Indicates that received data is transferred to the receive FIFO data register 2 (SCFRDR2), the quantity of data in SCFRDR2 becomes more than the number of receive triggers specified by the RTRG1 and RTRG0 bits in FIFO control register 2 (SCFCR2).

Bit 1: RDF	Description
0	The quantity of transmit data written to SCFRDR2 is less than the specified number of receive triggers. (Initial value) RDF is cleared to 0 at power-on reset or in standby mode, or when SCFRDR2 is read until the quantity of receive data in SCFRDR2 is less than the specified receive trigger number, and software reads 1 from RDF and then writes 0 to RDF.
1	The quantity of receive data in SCFRDR2 is more than the specified number of receive triggers. RDF is set to 1 when the quantity of receive data which is greater than the specified number of receive triggers is stored in SCFRDR2.*

Note: * Since SCFTDR2 is a 16-byte FIFO register, the maximum quantity of data which can be read when RDF is 1 is the specified number of receive triggers. If attempted to read after all data in the SCFRDR2 have been read, the data is undefined. The quantity of receive data in SCFRDR2 is indicated by the lower 8 bits of SCFTDR2.

Bit 0—Receive Data Ready (DR): Indicates that the receive FIFO data register 2 (SCFRDR2) stores the data which is less than the specified number of receive triggers, and that next data is not yet received after 15 ETU has elapsed from the last stop bit.

Bit 0: DR	Description
0	Receive is in progress, or no received data remains in SCFRDR2 after completing receive normally. (Initial value) DR is cleared to 0 when the chip is power-on reset or enters standby mode, or software reads DR after it has been set to 1, then writes 0 in DR.
1	Next receive data is not received. DR is set to 1 when SCFRDR2 stores the data which is less than the specified number of receive triggers, and that next data is not yet received after 15 ETU has elapsed from the last stop bit.*

Note: * This is equivalent to 1.5 frames with the 8-bit 1-stop-bit format. (ETU: Element Time Unit)

Upper 8 bits:	15	14	13	12	11	10	9	8
	PER3	PER2	PER1	PER0	FER3	FER2	FER1	FER0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bits 15 to 12—Number of Parity Errors 3 to 0 (PER3 to PER0): Indicates the quantity of data including a parity error in the received data stored in the receive FIFO data register 2 (SCFRDR2). The value indicated by the bits 15 to 12 represents the number of parity errors in SCFRDR2.

Bits 11 to 8—Number of Framing Errors 3 to 0 (FER3 to FER0): Indicates the quantity of data including a framing error in the received data stored in SCFRDR2. The value indicated by bits 11 to 8 represents the number of framing errors in SCFRDR2.

19.2.8 Bit Rate Register 2 (SCBRR2)

The bit rate register 2 (SCBRR2) is an eight-bit register that, together with the baud rate generator clock source selected by the CKS1 and CKS0 bits in the serial mode register 2 (SCSMR2), determines the serial transmit/receive bit rate.

The CPU can always read and write the SCBRR2. The SCBRR2 is initialized to H'FF by a reset or in module standby or standby mode. Each channel has independent baud rate generator control, so different values can be set in two channels.

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

The SCBRR2 setting is calculated as follows:

Asynchronous mode:

$$N = \frac{P\phi}{64 \times 2^{2n-1} \times B} \times 10^6 - 1$$

B: Bit rate (bit/s)
N: SCBRR2 setting for baud rate generator ($0 \leq N \leq 255$)
Pϕ: Operating frequency for peripheral modules (MHz)
n: Baud rate generator clock source (n = 0, 1, 2, 3) (for the clock sources and values of n, see table 19.3.)

Table 19.3 SCSMR2 Settings

		SCSMR2 Settings	
n	**Clock Source**	**CKS1**	**CKS0**
0	Pϕ	0	0
1	Pϕ/4	0	1
2	Pϕ/16	1	0
3	Pϕ/64	1	1

Note: Find the bit rate error by the following formula:

$$\text{Error (\%)} = \left\{ \frac{P\phi}{(N+1) \times 64 \times 2^{2n-1} \times B} \times 10^6 - 1 \right\} \times 100$$

Table 19.4 lists examples of SCBRR2 settings.

Table 19.4 Bit Rates and SCBRR2 Settings

	Pϕ (MHz)								
	2			2.097152			2.4576		
Bit Rate (bits/s)	**n**	**N**	**Error (%)**	**n**	**N**	**Error (%)**	**n**	**N**	**Error (%)**
110	1	141	0.03	1	148	–0.04	1	174	–0.26
150	1	103	0.16	1	108	0.21	1	127	0.00
300	0	207	0.16	0	217	0.21	0	255	0.00
600	0	103	0.16	0	108	0.21	0	127	0.00
1200	0	51	0.16	0	54	–0.70	0	63	0.00
2400	0	25	0.16	0	26	1.14	0	31	0.00
4800	0	12	0.16	0	13	–2.48	0	15	0.00
9600	0	6	–6.99	0	6	–2.48	0	7	0.00
19200	0	2	8.51	0	2	13.78	0	3	0.00
31250	0	1	0.00	0	1	4.86	0	1	22.88
38400	0	1	–18.62	0	0	–14.67	0	1	0.00

	Pϕ (MHz)								
	3			3.6864			4		
Bit Rate (bits/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	1	212	0.03	2	64	0.70	2	70	0.03
150	1	155	0.16	1	191	0.00	1	207	0.16
300	1	77	0.16	1	95	0.00	1	103	0.16
600	0	155	0.16	0	191	0.00	0	207	0.16
1200	0	77	0.16	0	95	0.00	0	103	0.16
2400	0	38	0.16	0	47	0.00	0	51	0.16
4800	0	19	–2.34	0	23	0.00	0	25	0.16
9600	0	9	–2.34	0	11	0.00	0	12	0.16
19200	0	4	–2.34	0	5	0.00	0	6	–6.99
31250	0	2	0.00	0	3	–7.84	0	3	0.00
38400	—	—	—	0	2	0.00	0	2	8.51

	Pϕ (MHz)								
	4.9152			5			6		
Bit Rate (bits/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	86	0.31	2	88	–0.25	2	106	–0.44
150	1	255	0.00	2	64	0.16	2	77	0.16
300	1	127	0.00	1	129	0.16	1	155	0.16
600	0	255	0.00	1	64	0.16	1	77	0.16
1200	0	127	0.00	0	129	0.16	0	155	0.16
2400	0	63	0.00	0	64	0.16	0	77	0.16
4800	0	31	0.00	0	32	–1.36	0	38	0.16
9600	0	15	0.00	0	15	1.73	0	19	–2.34
19200	0	7	0.00	0	7	1.73	0	9	–2.34
31250	0	4	–1.70	0	4	0.00	0	5	0.00
38400	0	3	0.00	0	3	1.73	0	4	–2.34

	Pϕ (MHz)								
	6.144			7.3728			8		
Bit Rate (bits/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	108	0.08	2	130	–0.07	2	141	0.03
150	2	79	0.00	2	95	0.00	2	103	0.16
300	1	159	0.00	1	191	0.00	1	207	0.16
600	1	79	0.00	1	95	0.00	1	103	0.16
1200	0	159	0.00	0	191	0.00	0	207	0.16
2400	0	79	0.00	0	95	0.00	0	103	0.16
4800	0	39	0.00	0	47	0.00	0	51	0.16
9600	0	19	0.00	0	23	0.00	0	25	0.16
19200	0	9	0.00	0	11	0.00	0	12	0.16
31250	0	5	2.40	0	6	5.33	0	7	0.00
38400	0	4	0.00	0	5	0.00	0	6	–6.99

	Pϕ (MHz)											
	9.8304			10			12			12.288		
Bit Rate (bits/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	1	174	–0.26	2	177	–0.25	1	212	0.03	2	217	0.08
150	1	127	0.00	2	129	0.16	1	155	0.16	2	159	0.00
300	0	255	0.00	2	64	0.16	1	77	0.16	2	79	0.00
600	0	127	0.00	1	129	0.16	0	155	0.16	1	159	0.00
1200	0	255	0.00	1	64	0.16	0	77	0.16	1	79	0.00
2400	0	127	0.00	0	129	0.16	0	38	0.16	0	159	0.00
4800	0	63	0.00	0	64	0.16	0	19	0.16	0	79	0.00
9600	0	31	0.00	0	32	–1.36	0	9	0.16	0	39	0.00
19200	0	15	0.00	0	15	1.73	0	4	0.16	0	19	0.00
31250	0	9	–1.70	0	9	0.00	0	2	0.00	0	11	2.40
38400	0	1	0.00	0	7	1.73	0	9	–2.34	0	9	0.00

	Pϕ (MHz)											
	14.7456			16			19.6608			20		
Bit Rate (bits/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	3	64	0.70	3	70	0.03	3	86	0.31	3	88	–0.25
150	2	191	0.00	2	207	0.16	2	255	0.00	2	64	0.16
300	2	95	0.00	2	103	0.16	2	127	0.00	2	129	0.16
600	1	191	0.00	1	207	0.16	1	255	0.00	1	64	0.16
1200	1	95	0.00	1	103	0.16	1	127	0.00	1	129	0.16
2400	0	191	0.00	0	207	0.16	0	255	0.00	0	64	0.16
4800	0	95	0.00	0	103	0.16	0	127	0.00	0	129	0.16
9600	0	47	0.00	0	51	0.16	0	63	0.00	0	64	0.16
19200	0	23	0.00	0	25	0.16	0	31	0.00	0	32	–1.36
31250	0	14	–1.70	0	15	0.00	0	19	–1.70	0	19	0.00
38400	0	11	0.00	0	12	0.16	0	15	0.00	0	15	1.73
115200	0	3	0.00	0	3	8.51	0	4	6.67	0	4	8.51
500000	0	0	–7.84	0	0	0.00	0	0	22.9	0	0	25.0

	Pϕ (MHz)											
	24			24.576			28.7			30		
Bit Rate (bits/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	3	106	–0.44	3	108	0.08	3	126	0.31	3	132	0.13
150	3	77	0.16	3	79	0.00	3	92	0.46	3	97	–0.35
300	2	155	0.16	2	159	0.00	2	186	–0.08	2	194	0.16
600	2	77	0.16	2	79	0.00	2	92	0.46	2	97	–0.35
1200	1	155	0.16	1	159	0.00	1	186	–0.08	1	194	0.16
2400	1	77	0.16	1	79	0.00	1	92	0.46	1	97	–0.35
4800	0	155	0.16	0	159	0.00	0	186	–0.08	0	194	–1.36
9600	0	77	0.16	0	79	0.00	0	92	0.46	0	97	–0.35
19200	0	38	0.16	0	39	0.00	0	46	–0.61	0	48	–0.35
31250	0	23	0.00	0	24	–1.70	0	28	–1.03	0	29	0.00
38400	0	19	–2.34	0	19	0.00	0	22	1.55	0	23	1.73
115200	0	6	–6.99	0	6	–4.76	0	7	–2.68	0	7	1.73
500000	0	1	–25.0	0	1	–23.2	0	1	–10.3	0	1	–6.25

Table 19.5 indicates the maximum bit rates in the asynchronous mode when the baud rate generator is being used.

Table 19.5 Maximum Bit Rates for Various Frequencies with Baud Rate Generator (Asynchronous Mode)

		Settings	
Pϕ (MHz)	**Maximum Bit Rate (bits/s)**	**n**	**N**
2	62500	0	0
2.097152	65536	0	0
2.4576	76800	0	0
3	93750	0	0
3.6864	115200	0	0
4	125000	0	0
4.9152	153600	0	0
8	250000	0	0
9.8304	307200	0	0
12	375000	0	0
14.7456	460800	0	0
16	500000	0	0
19.6608	614400	0	0
20	625000	0	0
24	750000	0	0
24.576	768000	0	0
28.7	896875	0	0
30	937500	0	0

19.2.9 FIFO Control Register 2 (SCFCR2)

Bit:	7	6	5	4	3	2	1	0
	RTRG1	RTRG0	TTRG1	TTRG0	MCE	TFRST	RFRST	LOOP
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

The FIFO control register 2 (SCFCR2) resets the number of data in the transmit and receive FIFO register 2, sets the number of trigger data, and contains the permit bit for the loop back test. The SCFCR2 is always read and written by the CPU. It is initialized to H'00 by the reset, the module standby function, or in the standby mode.

Bits 7 and 6—Trigger of the Number of Receive FIFO Data (RTRG1 and RTRG0): Set the number of receive data which sets the receive data full (RDF) flag in the serial status register 2 (SCSSR2). These bits set the RDF flag when the number of receive data stored in the receive FIFO register 2 (SCFRDR2) is increased more than the number of setting triggers listed below.

Bit 7: RTRG1	Bit 6: RTRG0	Number of Received Triggers
0	0	1 (Initial value)
	1	4
1	0	8
	1	14

Bits 5 and 4—Trigger of the Number of Transmit FIFO Data (TTRG1 and TTRG0): Set the number of remaining transmit data which sets the transmit FIFO data register empty (TDFE) flag in the serial status register 2 (SCSSR2). These bits set the TDFE flag when the number of transmit data in the transmit FIFO data register 2 (SCFTDR2) is decreased less than the number of setting triggers listed below.

Bit 5: TTRG1	Bit 4: TTRG0	Number of Transmitted Triggers
0	0	8 (8)*
	1	4 (12)
1	0	2 (14)
	1	1 (15)

Note: * Initial value. Values in brackets mean the number of empty SCFTDR2 when a flag occurs.

Bit 3—Modem Control Enable (MCE): Enables the modem control signals $\overline{\text{CTS}}$ and $\overline{\text{RTS}}$.

Bit 3: MCE	Description	
0	Disables the modem signal*	(Initial value)
1	Enables the modem signal	

Note: * The CTS is fixed to active 0 regardless of the input value, and the RTS is also fixed to 0.

Bit 2—Transmit FIFO Data Register Reset (TFRST): Disables the transmit data in the transmit FIFO data register 2 and resets the data to the empty state.

Bit 2: TFRST	Description	
0	Disables reset operation*	(Initial value)
1	Enables reset operation	

Note: * Reset is operated in resets or the standby mode.

Bit 1—Receive FIFO Data Register Reset (RFRST): Disables the receive data in the receive FIFO data register 2 and resets the data to the empty state.

Bit 1: RFRST	Description	
0	Disables reset operation*	(Initial value)
1	Enables reset operation	

Note: * Reset is operated in resets or the standby mode.

Bit 0—Loop Back Test (LOOP): Internally connects the transmit output pin (TXD2) and receive input pin (RXD2) and enables the loop back test.

Bit 0: LOOP	Description	
0	Disables the loop back test	(Initial value)
1	Enables the loop back test	

19.2.10 FIFO Data Count Set Register 2 (SCFDR2)

The SCFDR2 is a 16-bit register which indicates the number of data stored in the transmit FIFO data register 2 (SCFTDR2) and the receive FIFO data register 2 (SCFRDR2). It indicates the number of transmit data in the SCFTDR2 with the upper eight bits, and the number of receive data in the SCFRDR2 with the lower eight bits. The SCFDR2 is always read from the CPU.

Upper 8 Bits:	15	14	13	12	11	10	9	8
	—	—	—	T4	T3	T2	T1	T0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

The SCFDR2 indicates the number of non-transmitted data stored in the SCFTDR2. The H'00 means no transmit data, and the H'10 means that the full of transmit data are stored in the SCFTDR2.

Lower 8 Bits:	7	6	5	4	3	2	1	0
	—	—	—	R4	R3	R2	R1	R0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

The SCFDR2 indicates the number of receive data stored in the SCFRDR2. The H'00 means no receive data, and the H'10 means that the full of receive data are stored in the SCFRDR2.

19.3 Operation

19.3.1 Overview

For serial communication, the SCIF has an asynchronous mode in which characters are synchronized individually. Refer to section 17.3.2, Operation in Asynchronous Mode. The SCIF has the 16-byte FIFO buffer for both transmit and receive, reduces an overhead of the CPU, and enables continuous high-speed communication. Moreover, it has the $\overline{\text{RTS}}$ and $\overline{\text{CTS}}$ signals as the modem control signals. The transmission format is selected in the serial mode register 2 (SCSMR2), as listed in table 19.6. The clock source of SCIF is determined by the combination of CKE1 and CKE0 bits in the serial control register 2 (SCSCR2) as shown in table 19.7.

- Data length is selectable from seven or eight bits.
- Parity and multiprocessor bits are selectable. So is the stop bit length (one or two bits). The combination of the preceding selections constitutes the communication format and character length.
- In receiving, it is possible to detect framing errors (FER), parity errors (PER), receive FIFO data full, receive data ready, and breaks.
- In transmitting, it is possible to detect transmit FIFO data empty.
- The number of stored data for both the transmit and receive FIFO registers is displayed.
- SCIF clock source
  - The SCIF operates using the on-chip baud rate generator, and can output a serial clock signal with a frequency 16 times the bit rate.

Table 19.6 SCSMR2 Settings and SCIF Transmit/Receive

	SCSMR2 Settings				SCIF Transmit/Receive	
Mode	Bit 6 CHR	Bit 5 PE	Bit 3 STOP	Data Length	Parity Bit	Stop Bit Length
Asynchronous	0	0	0	8-bit	Not set	1 bit
			1			2 bits
		1	0		Set	1 bit
			1			2 bits
	1	0	0	7-bit	Not set	1 bit
			1			2 bits
		1	0		Set	1 bit
			1			2 bits

Table 19.7 Settings for SCSMR2 and SCSCR2 and Selection of Clock Source of SCIF

	SCSCR2 Settings		SCIF Transmit/Receive Clock	
Mode	Bit 1 CKE1	Bit 0 CKE0	Clock Source	Functions of SCK2 Pins
Asynchronous	0	0	Internal	SCIF does not use SCK2 pins.
		1		Inhibited
	1	0	External	Inhibited
		1		

19.3.2 Serial Operation

Transmit/Receive Formats: Table 19.8 lists eight communication formats that can be selected. The format is selected by settings in the serial mode register (SCSMR2).

Table 19.8 Serial Transmit/Receive Formats

SCSMR2 Bits			Serial Transmit/Receive Format and Frame Length											
CHR	PE	STOP	1	2	3	4	5	6	7	8	9	10	11	12
0	0	0	START	8-Bit data								STOP		
0	0	1	START	8-Bit data								STOP	STOP	
0	1	0	START	8-Bit data								P	STOP	
0	1	1	START	8-Bit data								P	STOP	STOP
1	0	0	START	7-Bit data							STOP			
1	0	1	START	7-Bit data							STOP	STOP		
1	1	0	START	7-Bit data							P	STOP		
1	1	1	START	7-Bit data							P	STOP	STOP	

Notes: START: Start bit
STOP: Stop bit
P: Parity bit

Transmitting and Receiving Data (SCIF Initialization): Before transmitting or receiving, clear the TE and RE bits to 0 in the serial control register 2 (SCSCR2), then initialize the SCIF as follows.

When changing the communication format, always clear the TE and RE bits to 0 before following the procedure given below. Clearing TE to 0 initializes the transmit shift register 2 (SCTSR2). Clearing TE and RE to 0, however, does not initialize the serial status register 2 (SCSSR2), transmit FIFO data register 2 (SCFTDR2), or receive FIFO data register 2 (SCFRDR2), which retain their previous contents.

Clear TE to 0 after all transmit data are transmitted and the TEND flag in the SCSSR2 is set. The transmitting data enters the high impedance state after clearing to 0 although the bit can be cleared to 0 in transmitting. Set the TFRST bit in the SCFCR2 to 1 and reset the SCFTDR2 before TE is set again to start transmission.

When an external clock is used, the clock should not be stopped during initialization or subsequent operation. SCIF operation becomes unreliable if the clock is stopped.

Figure 19.4 is a sample flowchart for initializing the SCIF. The procedure for initializing the SCIF is:

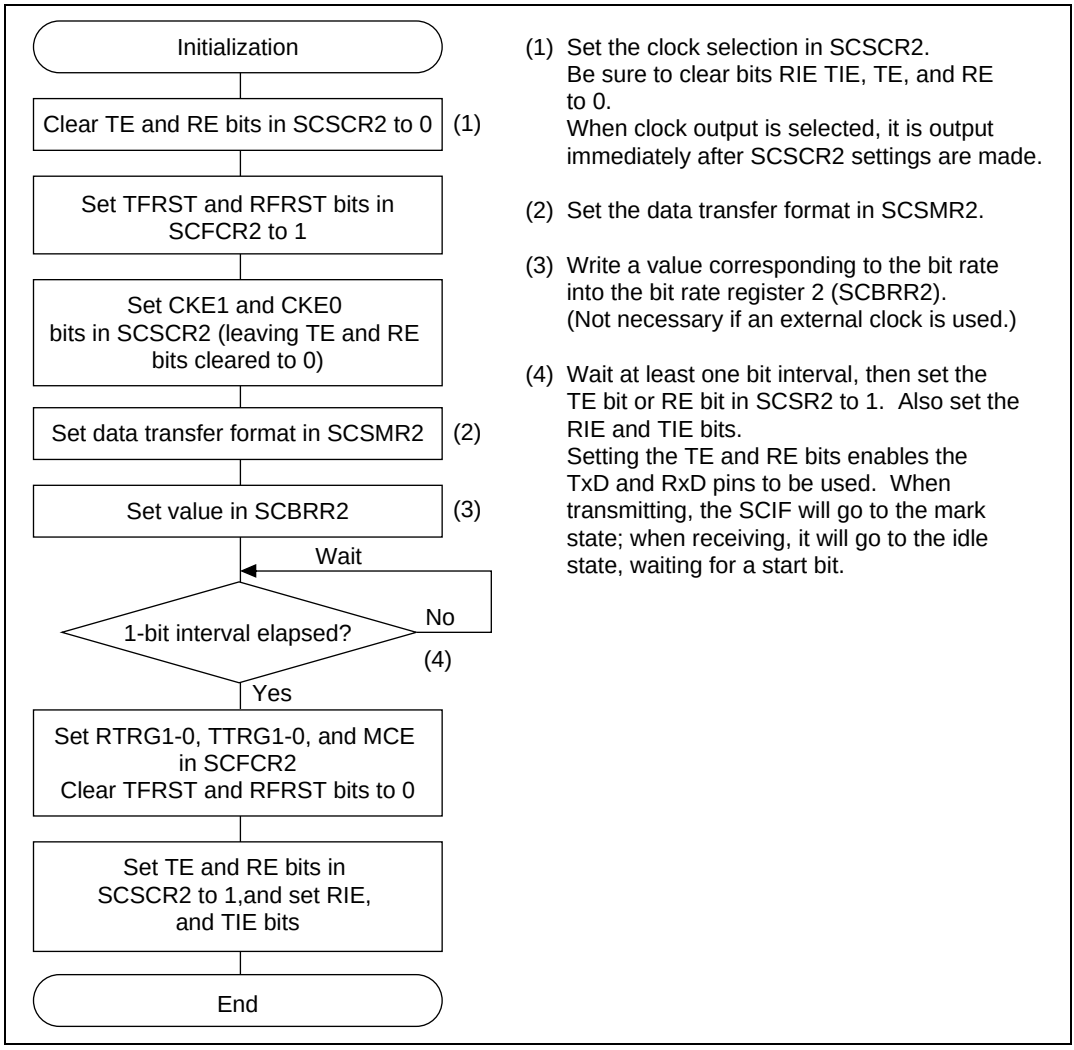
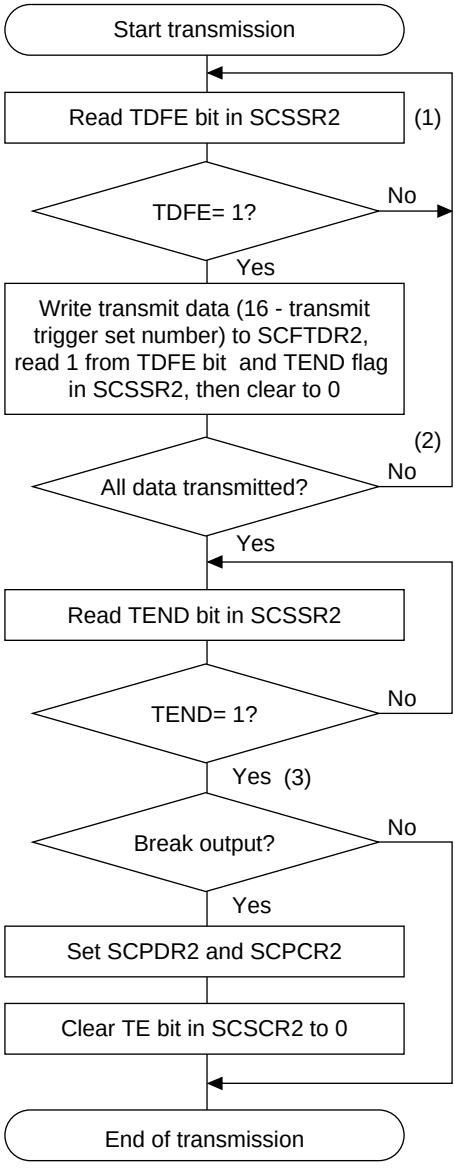


Figure 19.4 Sample SCIF Initialization Flowchart

- Serial data transmission

Figure 19.5 shows a sample serial transmission flowchart. After SCIF transmission is enabled, use the following procedure to perform serial data transmission.



(1) SCIF status check and transmit data write:
Read serial status register 2 (SCSSR2) and check that the TDFE flag is set to 1, then write transmit data to the transmit FIFO data register 2 (SCFTDR2), read 1 from the TDFE and TEND flags, then clear these flags to 0.
The number of transmit data bytes that can be written is 16 - (transmit trigger set number).

(2) Serial transmission continuation procedure:
To continue serial transmission, read 1 from the TDFE flag to confirm that writing is possible, then write data to SCFTDR2, and then clear the TDFE flag to 0.

(3) Break output at the end of serial transmission:
To output a break in serial transmission, set the port SC data register (SCPDR) and port SC control register (SCPCR), then clear the TE bit to 0 in the serial control register 2 (SCSCR2). For information on SCPDR2 and SCPCR2, see section 17.2.8.
In steps 1 and 2, it is possible to ascertain the number of data bytes that can be written from the number of transmit data bytes in SCFTDR indicated by the upper 8 bits of the FIFO data count set register 2 (SCFDR2).

Figure 19.5 Sample Serial Transmission Flowchart

In serial transmission, the SCIF operates as described below.

1. When data is written into the transmit FIFO data register 2 (SCFTDR2), the SCIF transfers the data from SCFTDR2 to the transmit shift register 2 (SCTSR2) and starts transmitting. Confirm that the TDFE flag in the serial status register 2 (SCSSR2) is set to 1 before writing transmit data to SCFTDR2. The number of data bytes that can be written is (16 – transmit trigger setting).

2. When data is transferred from SCFTDR2 to SCTSR2 and transmission is started, consecutive transmit operations are performed until there is no transmit data left in SCFTDR2. When the number of transmit data bytes in SCFTDR2 falls below the transmit trigger number set in the FIFO control register 2 (SCFCR2), the TDFE flag is set. If the TIE bit in the serial control register (SCSR2) is set to 1 at this time, a transmit-FIFO-data-empty interrupt (TXI) request is generated.

   The serial transmit data is sent from the TxD pin in the following order.

   a. Start bit: One-bit 0 is output.
   b. Transmit data: 8-bit or 7-bit data is output in LSB-first order.
   c. Parity bit: One parity bit (even or odd parity) is output. (A format in which a parity bit is not output can also be selected.)
   d. Stop bit(s): One- or two-bit 1s (stop bits) are output.
   e. Mark state: 1 is output continuously until the start bit that starts the next transmission is sent.

3. The SCIF checks the SCFTDR2 transmit data at the timing for sending the stop bit. If data is present, the data is transferred from SCFTDR2 to SCTSR2, the stop bit is sent, and then serial transmission of the next frame is started.

   If there is no transmit data, the TEND flag in SCSSR2 is set to 1, the stop bit is sent, and then the line goes to the mark state in which 1 is output continuously.

   Figure 19.6 shows an example of the operation for transmission.

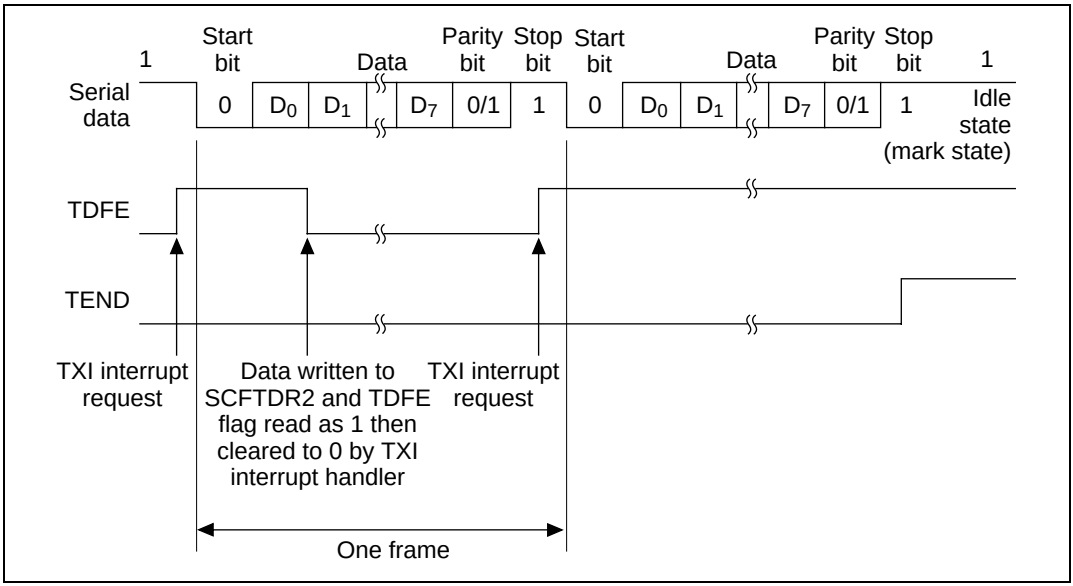


Figure 19.6 Example of Transmit Operation (Example with 8-Bit Data, Parity, One Stop Bit)

4. When modem control is enabled, transmission can be stopped and restarted in accordance with the $\overline{\text{CTS}}$ input value. When $\overline{\text{CTS}}$ is set to 1, if transmission is in progress, the line goes to the mark state after transmission of one frame. When $\overline{\text{CTS}}$ is set to 0, the next transmit data is output starting from the start bit.

   Figure 19.7 shows an example of the operation when modem control is used.

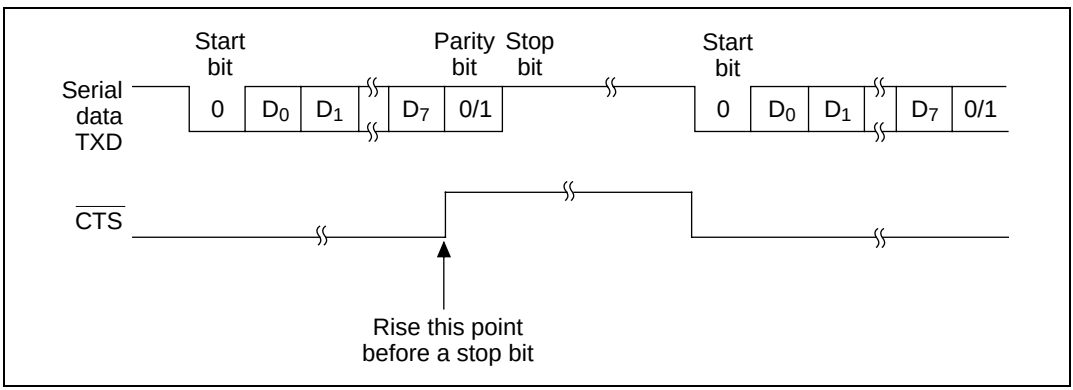
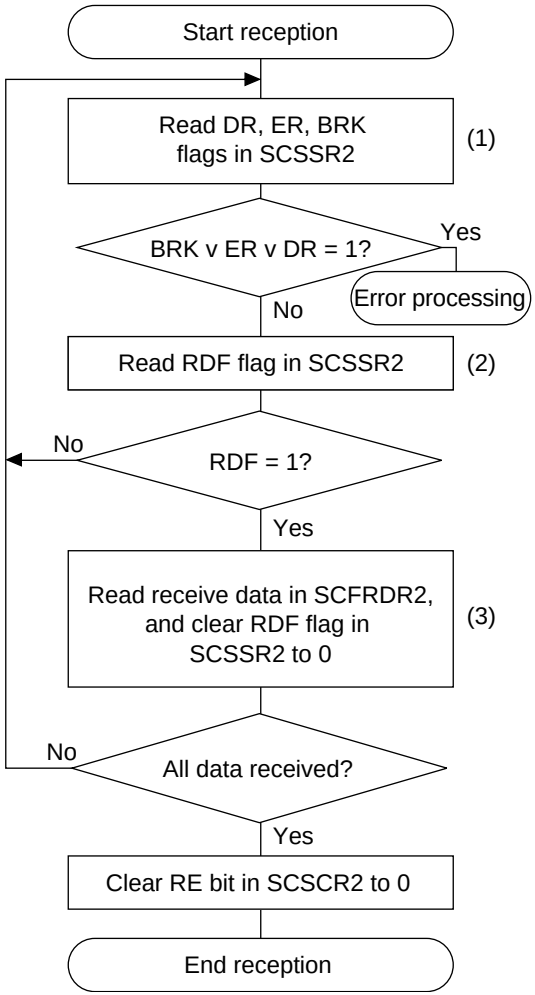


Figure 19.7 Example of Operation Using Modem Control ($\overline{\text{CTS}}$)

- Serial data reception

Figures 19.8 and 19.9 show sample serial reception flowcharts. After SCIF reception is enabled, use the following procedure to perform serial data reception.

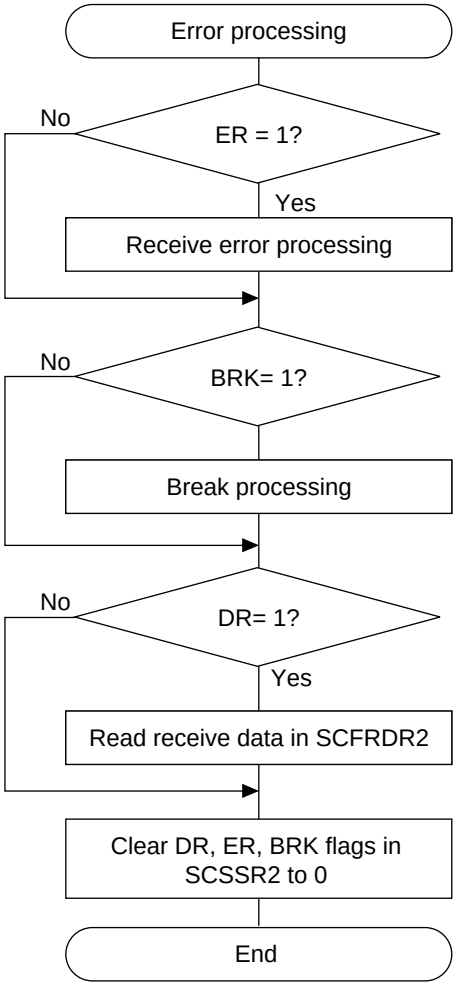


(1) Receive error handling and break detection:
Read the DR, ER, and BRK flags in SCSSR2 to identify any error, perform the appropriate error handling, then clear the DR, ER, and BRK flags to 0. In the case of a framing error, a break can also be detected by reading the value of the RxD2 pin.

(2) SCIF status check and receive data read :
Read the serial status register 2 (SCSSR2) and check that RDF = 1, then read the receive data in the receive FIFO data register 2 (SCFRDR2), read 1 from the RDF flag, and then clear the RDF flag to 0. The transition of the RDF flag from 0 to 1 can be identified by an RXI interrupt.

(3) Serial reception continuation procedure:
To continue serial reception, read at least the receive trigger set number of receive data bytes from SCFRDR2, read 1 from the RDF flag, then clear the RDF flag to 0. The number of receive data bytes in SCFRDR2 can be ascertained by reading the lower bits of SCFDR2.

Figure 19.8 Sample Serial Reception Flowchart (1)



1. Whether a framing error or parity error has occurred in the receive data read from SCFRDR2 can be ascertained from the FER and PER bits in SCSSR2.

2. When a break signal is received, receive data is not transferred to SCFRDR2 while the BRK flag is set. However, note that the last data in SCFRDR2 is H'00 and the break data in which a framing error occurred is stored.

Figure 19.9 Sample Serial Reception Flowchart (2)

In serial reception, the SCIF operates as described below.

1. The SCIF monitors the transmission line, and if a 0 start bit is detected, performs internal synchronization and starts reception.

2. The received data is stored in SCRSR2 in LSB-to-MSB order.

3. The parity bit and stop bit are received.

   After receiving these bits, the SCIF carries out the following checks.

   a. Stop bit check: The SCIF checks whether the stop bit is 1. If there are two stop bits, only the first is checked.

   b. The SCIF checks whether receive data can be transferred from the receive shift register 2 (SCRSR2) to SCFRDR2.

   c. Break check: The SCIF checks that the BRK flag is 0, indicating that the break state is not set.

      If all the above checks are passed, the receive data is stored in SCFRDR2.

Note: Reception becomes in possible after a receive error occurred.

4. If the RIE bit in SCSCR2 is set to 1 when the RDF or DR flag changes to 1, a receive-FIFO-data-full interrupt (RXI) request is generated.
   If the RIE bit in SCSCR2 is set to 1 when the ER flag changes to 1, a receive-error interrupt (ERI) request is generated.

   If the RIE bit in SCSCR2 is set to 1 when the BRK flag changes to 1, a break reception interrupt (BRI) request is generated.

   Figure 19.10 shows an example of the operation for reception.

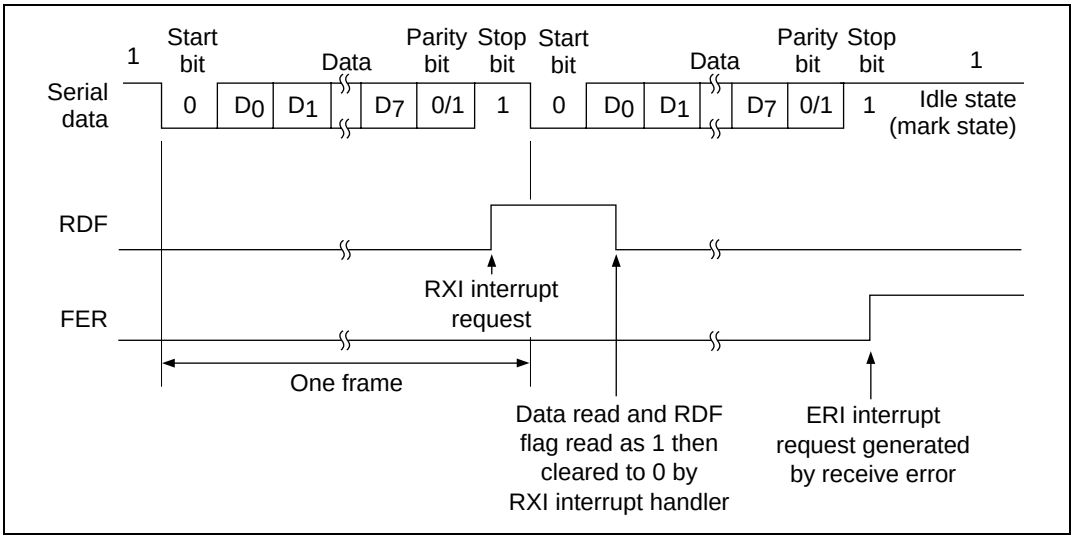


Figure 19.10 Example of SCIF Receive Operation (Example with 8-Bit Data, Parity, One Stop Bit)

5. When modem control is enabled, the $\overline{\text{RTS}}$ signal is output when SCFRDR2 is empty. When $\overline{\text{RTS}}$ is 0, reception is possible. When $\overline{\text{RTS}}$ is 1, this indicates that SCFRDR2 is full and reception is not possible.

   Figure 19.11 shows an example of the operation when modem control is used.

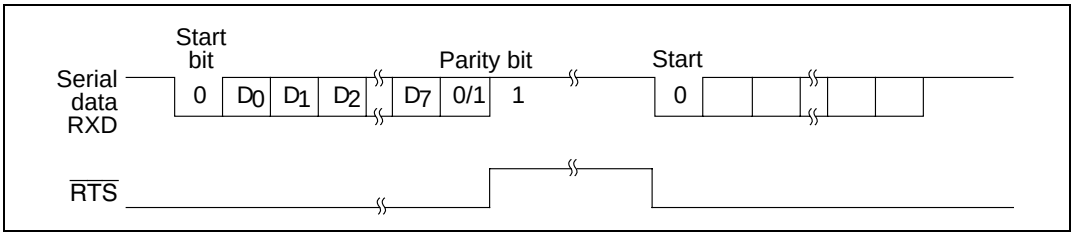


Figure 19.11 Example of Operation Using Modem Control ($\overline{\text{RTS}}$)

19.4 SCIF Interrupts

The SCIF has four interrupt sources: transmit-FIFO-data-empty (TXI), receive-error (ERI), receive-data-full (RXI), and break (BRI).

Table 19.9 shows the interrupt sources and their order of priority. The interrupt sources are enabled or disabled by means of the TIE and RIE bits in SCSCR2. A separate interrupt request is sent to the interrupt controller for each of these interrupt sources.

When the TDFE flag in the serial status register 2 (SCSSR2) is set to 1, a TXI interrupt request is generated. The DMAC can be activated and data transfer performed when this interrupt is generated. The TDFE flag is cleared when data exceeding the transmit trigger number is written to transmit FIFO data register 2 (SCFTDR2) by the DMAC, 1 is read from TDFE, and then 0 is written to TDFE.

When the RDF flag in SCSSR2 is set to 1, an RXI interrupt request is generated. The DMAC can be activated and data transfer performed when the RDF flag in SCSSR is set to 1. The RDF flag is cleared when receive data is read from receive FIFO data register 2 (SCFRDR2) by the DMAC until the quantity of receive data in SCFRDR2 is less than the receive trigger number, 1 is read from RDF, and then 0 is written to RDF.

When the ER flag in SCSSR2 is set to 1, an ERI interrupt request is generated.

When the BRK flag in SCSSR2 is set to 1, a BRI interrupt request is generated.

The TXI interrupt indicates that transmit data can be written, and the RXI interrupt indicates that there is receive data in SCFRDR2.

Table 19.9 SCIF Interrupt Sources

Interrupt Source	Description	DMAC Activation	Priority on Reset Release
ERI	Interrupt initiated by receive error flag (ER)	Impossible	High
RXI	Interrupt initiated by receive data FIFO full flag (RDF) or data ready flag (DR)	Possible (RDF only)	↑
BRI	Interrupt initiated by break flag (BRK)	Impossible	
TXI	Interrupt initiated by transmit FIFO data empty flag (TDFE)	Possible	↓ Low

See section 4, Exception Processing, for priorities and the relationship with non-SCIF interrupts.

19.5 Usage Notes

Note the following when using the SCIF.

1. SCFTDR2 Writing and the TDFE Flag: The TDFE flag in the serial status register 2 (SCSSR2) is set when the number of transmit data bytes written in the transmit FIFO data register 2 (SCFTDR2) has fallen below the transmit trigger number set by bits TTRG1 and TTRG0 in the FIFO control register 2 (SCFCR2). After TDFE is set, transmit data up to the number of empty bytes in SCFTDR2 can be written, allowing efficient continuous transmission. However, if the number of data bytes written in SCFTDR2 is less than or equal to the transmit trigger number, the TDFE flag will be set to 1 again after being cleared to 0. The TDFE flag should therefore be cleared to 0 after a number of data bytes exceeding the transmit trigger number has been written to SCFTDR2.
   The number of transmit data bytes in SCFTDR2 can be found from the upper 8 bits of the FIFO data count set register 2 (SCFDR2).

2. SCFRDR2 Reading and the RDF Flag: The RDF flag in the serial status register 2 (SCSSR2) is set when the number of receive data bytes in the receive FIFO data register 2 (SCFRDR2) has become equal to or greater than the receive trigger number set by bits RTRG1 and RTRG0 in the FIFO control register 2 (SCFCR2). After RDF is set, receive data equivalent to the trigger number can be read from SCFRDR2, allowing efficient continuous reception. However, if the number of data bytes in SCFRDR2 exceeds the trigger number, the RDF flag will be set to 1 again after being cleared to 0. The RDF flag should therefore be cleared to 0 when 1 has been written to RDF after all receive data has been read.
   The number of receive data bytes in SCFRDR2 can be found from the lower 8 bits of the FIFO data count set register 2 (SCFDR2).

3. Break Detection and Processing: Break signals can be detected by reading the RxD pin directly when a framing error (FER) is detected. In the break state the input from the RxD pin consists of all 0s, so the FER flag is set and the parity error flag (PER) may also be set. Note that, although transfer of receive data to SCFRDR2 is halted in the break state, the SCIF receiver continues to operate, so if the BRK flag is cleared to 0 it will be set to 1 again.

4. Sending a Break Signal: The I/O condition and level of the TxD pin are determined by the SCP4DT bit in the port SC data register 2 (SCPDR2) and bits SCP4MD0 and SCP4MD1 in the port SC control register 2 (SCPCR2). This feature can be used to send a break signal.
   To send a break signal during serial transmission, clear the CP4DT bit to 0 (designating low level), then set the SCP4MD0 and SCP4MD1 bits to 0 and 1, respectively, and finally clear the TE bit to 0 (halting transmission). When the TE bit is cleared to 0, the transmitter is initialized regardless of the current transmission state, and 0 is output from the TxD pin.

5. TEND Flag and TE Bit Processing: The TEND flag is set to 1 during transmission of the stop bit of the last data. Consequently, if the TE bit is cleared to 0 immediately after setting of the TEND flag has been confirmed, the stop bit will be in the process of transmission and will not be transmitted normally. Therefore, the TE bit should not be cleared to 0 for at least 0.5 serial clock cycles (or 1.5 cycles if two stop bits are used) after setting of the TEND flag setting is confirmed.

6. Receive Data Sampling Timing and Receive Margin: The SCIF operates on a base clock with a frequency of 16 times the transfer rate. In reception, the SCIF synchronizes internally with the fall of the start bit, which it samples on the base clock. Receive data is latched at the rising edge of the eighth base clock pulse. The timing is shown in figure 19.12.

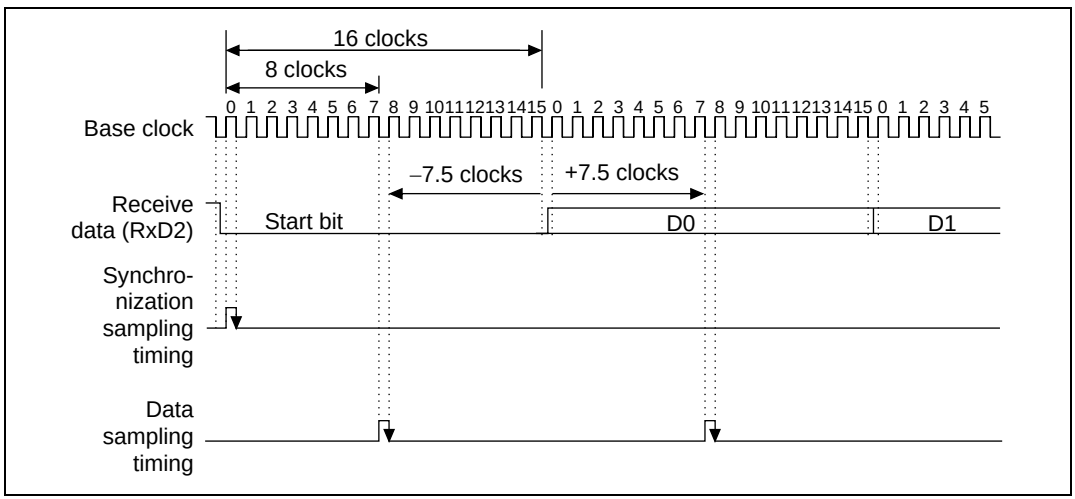


Figure 19.12 Receive Data Sampling Timing in Asynchronous Mode

The receive margin in asynchronous mode can therefore be expressed as shown in equation (1).

Equation 1:

$$M = \left| \left(0.5 - \frac{1}{2N}\right) - (L - 0.5)\,F - \frac{|D - 0.5|}{N}(1 + F) \right| \times 100\% \quad \cdots\cdots\cdots\cdots\cdots\cdots (1)$$

M: Receive margin (%)
N: Ratio of clock frequency to bit rate (N = 16)
D: Clock duty cycle (D = 0 to 1.0)
L: Frame length (L = 9 to 12)
F: Absolute deviation of clock frequency

From equation (1), if F = 0 and D = 0.5, the receive margin is 46.875%, as given by equation (2).

When D = 0.5 and F = 0:

$$M = (0.5 - 1/(2 \times 16)) \times 100\%$$
$$= 46.875\% \quad \text{(2)}$$

This is a theoretical value. A reasonable margin to allow in system designs is 20% to 30%.

Section 20 Serial IO (SIOF)

20.1 Overview

SIOF is a clock-synchronized serial I/O module that can be directly connected to the audio CODEC.

20.1.1 Features

Feature of SIOF are listed below.

- Serial transmitting
  - 32 bit × 16 step sized FIFO for send or receive communication
  - 8 bit/16 bit/recording or playback function for 16 bit stereo sound
  - Both big endian and little endian are supported in data transmission and reception
  - Variable sampling rate up to 48 kHz
  - Synchronization corresponds to frame sync pulse and switching right/left channels
  - Support CODEC control function through data line
  - Support linear/audio A-Law and μ-Law CODEC chip
  - Support both master and receive communication mode
- Serial clock
  - External clock or peripheral clock (Pϕ) may be used as clock source.
- Interrupt

  It is possible to require independently the following 4 interruptions.
  - Transmit interrupt
  - Receive interrupt
  - Error interrupt
  - Control interrupt
- DMA transfer
  - Supports transmit/receive operations using DMA transfer in response to a transmit/receive transfer request

20.1.2 Block Diagram

Figure 20.1 shows SIOF block diagram.

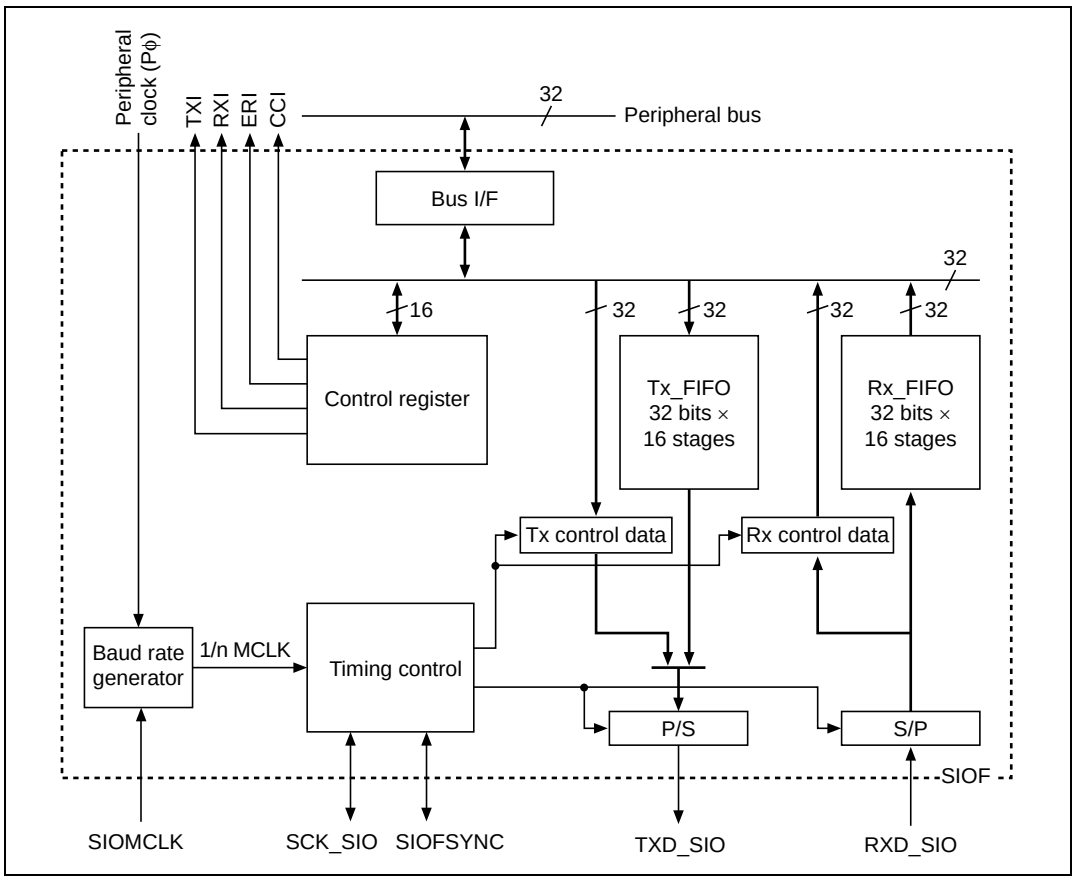


Figure 20.1 SIOF Block Diagram

20.1.3 Terminal

Table 20.1 shows pin list of SIOF.

Table 20.1 SIOF Pin List

Name	Symbol	I/O	Function
Clock input pin	SIOMCLK	I	Master clock input
Communication clock pin	SCK_SIO	I/O	Serial clock (common for transmitting/ receiving)
Frame sync pin	SIOFSYNC	I/O	Frame synchronized signal (common for transmitting/receiving)
Transmit data pin	TXD_SIO	O	Transmit data
Receive data pin	RXD_SIO	I	Receive data

20.1.4 Register Configuration

Table 20.2 lists the internal registers of SIOF.

Table 20.2 SIOF Register Configuration

Name	Symbol	R/W	Initial Value	Address	Access Size
Serial mode register	SIMDR	R/W	H'0000	H'040000C0 (H'A40000C0)*	16
Clock select register	SISCR	R/W	H'0000	H'040000C2 (H'A40000C2)*	16
Transmit data assign register	SITDAR	R/W	H'0000	H'040000C4 (H'A40000C4)*	16
Receive data assign register	SIRDAR	R/W	H'0000	H'040000C6 (H'A40000C6)*	16
Control data assign register	SICDAR	R/W	H'0000	H'040000C8 (H'A40000C8)*	16
Serial control register	SICTR	R/W	H'0000	H'040000CC (H'A40000CC)*	16
FIFO control register	SIFCTR	R/W	H'1000	H'040000D0 (H'A40000D0)*	16
Status register	SISTR	R/W	H'0000	H'040000D4 (H'A40000D4)*	16
Interruption enabling register	SIIER	R/W	H'0000	H'040000D6 (H'A40000D6)*	16
Transmit data register	SITDR	W	H'0000	H'040000E0 (H'A40000E0)*	32
Receive data register	SIRDR	R	H'0000	H'040000E4 (H'A40000E4)*	32
Transmit control register	SITCR	R/W	H'0000	H'040000E8 (H'A40000E8)*	32
Receive control register	SIRCR	R	H'0000	H'040000EC (H'A40000EC)*	32

Note: * Use the address surrounded by parenthesis when the address translation process by MMU is not applied. Refer to section 20.3.5, Control Data Interface for more details of the control data.

20.2 Register Description

20.2.1 Mode Register (SIMDR)

This register sets the operating mode of SIOF.

This register is initialized by the power on reset or manual reset.

Bit:	15	14	13	12	11	10	9	8
	TRMD1	TRMD0	—	REDG	FL3	FL2	FL1	FL0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R*	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
	TXDIZ	LSBF	RCIM	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R*	R*	R*	R*	R*

Note: * 0 must be written into these bits. The operation of this LSI is unpredictable when setting the value other than 0.

Bits 13 and 4 to 0—Reserved

Bits 15 and 14—Transmit Mode Setting (TRMD1 and TRMD0)

Bit 15: TRMD1	Bit 14: TRMD0	Description	
0	0	Slave mode 1	(Initial value)
	1	Slave mode 2	
1	0	Master mode 1	
	1	Master mode 2	

Note: Refer to section 20.3.3, Transmit Data Format for more details of the functions of each mode.

Bit 12—Receive with Sampling Edge (REDG): TXD_SIO is output at the opposite edge from the sampling time of RXD_SIO. (see figure 20.4)

Bit 12: REDG	Description	
0	Sample RXD_SIO by falling edge of SCK_SIO	(Initial value)
1	Sample RXD_SIO by rising edge of SCK_SIO	

Note: This mode is effective in master mode 1 or master mode 2.

Bits 11 to 8—Frame Length (FL3 to FL0)

Bit 11: FL3	Bit 10: FL2	Bit 9: FL1	Bit 8: FL0	Description	
0	0	0/1*	0/1*	Slot size is 8 bit, frame length is 8 bit	(Initial value)
	1	0	0	Slot size is 8 bit, frame length is 16 bit	
			1	Slot size is 8 bit, frame length is 32 bit	
		1	0	Slot size is 8 bit, frame length is 64 bit	
			1	Slot size is 8 bit, frame length is 128 bit	
1	0	0/1*	0/1*	Slot size is 16 bit, frame length is 16 bit	
	1	0	0	Slot size is 16 bit, frame length is 32 bit	
			1	Slot size is 16 bit, frame length is 64 bit	
		1	0	Slot size is 16 bit, frame length is 128 bit	
			1	Slot size is 16 bit, frame length is 256 bit	

Notes: 1. When 8 bit slot size is chosen, control data is not able to be transmitted or received.
2. When LSB first is chosen, control data is not able to be transmitted or received.
* Same setting in both 1 and 0. (Don't care)

Bit 7—Hi-Z Output Control under Ineffective Case for Transmission (TXDIZ): Ineffective case is the case of transmit data or command are not assigned or transmit operation is disabled.

Bit 7: TXDIZ	Description	
0	1 output ineffective transmission	(Initial value)
1	Hi-Z output ineffective transmission	

Bit 6—LSB First Transmission and Reception (LSBF): Selects MSB first or LSB first for the transmission or reception frame.

Bit 6: LSBF	Description	
0	MSB First	(Initial value)
1	LSB First	

Bit 5—Transmit or Receive Control Command Interrupt Mode (RCIM)

Bit 5: RCIM	Description
0	Set RCRDY bit of SISTR register when the contents of SIRCR register is changed. (Initial value)
1	Set RCRDY bit of SISTR register when every control commands are received and set to SIRCR register

20.2.2 Clock Select Register (SISCR)

This register sets the operate of baud rate generator. To set up this register, TRMD bit of SIMDR register must be set 10 or 11.

This register is initialized in power on reset or software reset.

Bit:	15	14	13	12	11	10	9	8
	MSSEL	MSIMM	—	BRPS4	BRPS3	BRPS2	BRPS1	BRPS0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R*	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	—	BRDV2	BRDV1	BRDV0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R*	R*	R*	R*	R*	R/W	R/W	R/W

Note: * 0 must be written into this bit. The operation of this LSI is unpredictable when setting the value other than 0.

Bit 15—Master Clock Source Choice (MSSEL): Master clock means the clock that is input to the baud rate generator.

Bit 15: MSSEL	Description
0	Use external clock source SIOMCLK input signal as master clock (Initial value)
1	Use peripheral clock (Pϕ) as master clock

Bit 14—Master Clock Select (MSIMM)

Bit 14: MSIMM	Description	
0	Use baud rate generator output clock as clock source	(Initial value)
1	Use master clock as clock source	

Bits 13 and 7 to 3—Reserved

Bits 12 to 8—Setting of Prescaler (BRPS4 to BRPS0): The dividing ratio of master clock (BRPS) is set within the range of 00001 (1/1 times), 00010 (1/2 times), to 11111 (1/31 times) and 00000 (1/32 times: initial value).

Bits 2 to 0—Setting of Dividing Ratio (BRDV2 to BRDV0): Set the dividing ratio of frequency of output stage (BRDV). Finally, baud rate is decided by the value of BRPS * BRDV (maximum 1/1024).

Bit 2: BRDV2	Bit 1: BRDV1	Bit 0: BRDV0	Description	
0	0	0	1/2 times of prescaler output	(Initial value)
		1	1/4 times of prescaler output	
	1	0	1/8 times of prescaler output	
		1	1/16 times of prescaler output	
1	0	0	1/32 times of prescaler output	
Settings other than the above			(Reserved)	

20.2.3 Transmit Data Assign Register (SITDAR)

This register specifies the data assignment of transmit data in each transmit frame. This register is initialized in power on reset or software reset.

Bit:	15	14	13	12	11	10	9	8
	TDLE	—	—	—	TDLA3	TDLA2	TDLA1	TDLA0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R	R	R	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
	TDRE	TLREP	—	—	TDRA3	TDRA2	TDRA1	TDRA0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R	R	R/W	R/W	R/W	R/W

Bits 14 to 12, 5, and 4—Reserved

Bit 15—Transmit Data for Left Channel Enable (TDLE)

Bit 15: TDLE	Description	
0	Disable data transmitting of left channel data	(Initial value)
1	Enable data transmitting of left channel data	

Bits 11 to 8—Transmit Data for Left Channel Assignment (TDLA3 to TDLA0): The slot assignment of transmit data for Left channel in transmit frame is specified from 0000(0: initial value) to 1110(14) by this register. The transmit data for left channel is set in bits SITDL 15 to SITDR register.

Note: The operation of this LSI is unpredictable when setting 1111 in bits TDLA3 to TDLA0.

Bit 7—Transmit Data for Right Channel Enable (TDRE)

Bit 7: TDRE	Description	
0	Disable data transmitting of right channel data	(Initial value)
1	Enable data transmitting of right channel data	

Bit 6—Transmit Left Channel Data Repeatedly (TLREP): Setting of this bit is effective when TDRE bit is 1. When 1 is set to this bit, setting of bits 15 to 0 in SITDR register is ignored.

Bit 6: TLREP	Description
0	The data in SITDR bit of SITDR register is transmitted as right channel data. (Initial value)
1	The data in SITDL bit of SITDL register is transmitted as right channel data.

Bits 3 to 0—Transmit Data for Right Channel Slot Assignment (TDRA3 to TDRA0): The slot assignment of transmit data for Right channel in transmit frame is specified from 0000(0: initial value) to 1110(14) by this register. The transmit data for right channel is set in SITDR bits 15 to 0 in SITDR register.

Note: The operation of this LSI is unpredictable when setting 1111 in bits TDRA3 to TDRA0.

20.2.4 Receive Data Assign Register (SIRDAR)

This register specifies the data assignment of received data in each received frame. This register is initialized at power on reset or software reset.

Bit:	15	14	13	12	11	10	9	8
	RDLE	—	—	—	RDLA3	RDLA2	RDLA1	RDLA0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R	R	R	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
	RDRE	—	—	—	RDRA3	RDRA2	RDRA1	RDRA0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R	R	R	R/W	R/W	R/W	R/W

Bits 14 to 12, and 6 to 4—Reserved

Bit 15—Receive Data for Left Channel Enable (RDLE)

Bit 15: RDLE	Description
0	Disable receiving of left channel data (Initial value)
1	Enable receiving of left channel data

Bits 11 to 8—Receive Data for Left Channel Slot Assignment (RDLA3 to RDLA0): The slot assignment of received data for left channel in received frame is specified from 0000(0: initial value) to 1110(14) by this register. The receive data for left channel is stored in bits 15 to 0 in SIRDL of SIRDR register.

Note: The operation of this LSI is unpredictable when setting 1111 in bits RDLA3 to RDLA0.

Bit 7—Receive Data for Right Channel Enable (RDRE)

Bit 7: RDRE	Description	
0	Disable receiving of right channel data	(Initial value)
1	Enable receiving of right channel data	

Bits 3 to 0—Receive Data for Right Channel Slot Assignment (RDRA3 to RDRA0): The slot assignment of received data for right channel in received frame is specified from 0000(0: initial value) to 1110(14) by this register. The receive data for right channel is stored in bits 15 to 0 in SIRDR of SIRDR register.

Note: The operation of this LSI is unpredictable when setting 1111 in bits RDRA3 to RDRA0.

20.2.5 Control Command Assign Register (SICDAR)

This register specifies the position of control command in each frame. The setting to this register is enabled when 1*** is set to bits FL3 to FL0 of SIMDR register. This register is initialized at power on reset or software reset.

Bit:	15	14	13	12	11	10	9	8
	CD0E	—	—	—	CD0A3	R/W2	R/W1	R/W0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R	R	R	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
	CD1E	—	—	—	CD1A3	CD1A2	CD1A1	CD1A0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R	R	R	R/W	R/W	R/W	R/W

Bits 14 to 12, and 6 to 4—Reserved

Bit 15—Control Command Data Channel 0 Enable (CD0E)

Bit 15: CD0E	Description
0	Disable transmitting or receiving of control command of channel 0. (Initial value)
1	Enable transmitting or receiving of control command of channel 0.

Bits 11 to 8—Control Command Data Assignment for Channel 0 (CD0A3 to CD0A0): The slot assignment for control channel 0 in transmit and receive frames is specified from 0000(0: initial value) to 1110(14) by this register. The receive data for control channel 0 is set in bits SITC05 to SITC00 of SIRCR register. The receive data for control channel 0 is stored in bits SIRC015 to SIRC00 in SIRCR register.

Note: The operation of this LSI is unpredictable when setting 1111 in bits CD0A3 to CD0A0.

Bit 7—Control Command Data Channel 1 Enable (CD1E)

Bit 7: CD1E	Description
0	Disable transmitting or receiving of control command of channel 1. (Initial value)
1	Enable transmitting or receiving of control command of channel 1.

Bits 3 to 0—Control Command Data Assignment for Channel 1 (CD1A3 to CD1A0): The slot assignment for control channel 1 in transmit and receive frames is specified from 0000(0: initial value) to 1110(14) by this register. The receive data for control channel 1 is set in bits SIRC115 to SIRC10 of SIRCR register.

Note: The operation of this LSI is unpredictable when setting 1111 in bits CD1A3 to CD1A0.

20.2.6 Serial Control Register (SICTR)

This register sets the operating states of SIOF.

This register is initialized at power on reset or software reset.

Bit:	15	14	13	12	11	10	9	8
	SCKE	FSE	—	—	—	—	TXE	RXE
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R	R	R	R	R/W	R/W

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

Bits 13 to 10, and 7 to 2—Reserved

Bit 15—Serial Clock Output Enable (SCKE): This bit is effective in master mode. When 1 is set to this bit, SIOF initializes baud rata generator, then starts, operation, and outputs the clock that is generated by baud rate generator to SCK_SIO.

Bit 15: SCKE	Description	
0	Disable output of SCK_SIO (outputs 0)	(Initial value)
1	Enable output of SCK_SIO	

Bit 14—Frame Synchronize Signal Output Enable (FSE): This bit is effective at master mode. When 1 is set to this bit, SIOF initializes the frame counter, then starts operation.

Bit 14: FSE	Description	
0	Disable output of SIOFSYNC (outputs 0)	(Initial value)
1	Enable output of SIOFSYNC	

Bit 9—Transmit Enable (TXE): Setting of this bit becomes effective when next frame starts (at the rising edge of frame synchronize signal)and data are stored in transmit FIFO. After the setting "1" to this bit becomes effective, SIOF submit the transmit request according to the TFWM bit of SIFCTR register. When data is sets to transmit FIFO, transmit data is transfer from TXD_SIO. This bit is initialized at transmit reset.

Bit 9: TXE	Description	
0	Disable to transmit data from TXD_SIO (outputs 1)	(Initial value)
1	Enable to transmit data from TXD_SIO	

Bit 8—Receive Enable (RXE): Setting of this bit is effective when next frame starts (at the rising edge of frame synchronizing signal). After the setting "1" to this bit becomes effective, SIOF begins to receive the data from RXD_SIO. When data is sets to the receive FIFO, SIOF submits the request to transfer according to RFWM bit of SIFCTR register. This bit is initialized at receive reset.

Bit 8: RXE	Description	
0	Disable to receive data from RXD_SIO	(Initial value)
1	Enable to receive data from RXD_SIO	

Bit 1—Transmitting Operation Reset (TXRST): Setting to this bit becomes effective immediately. After the setting 1 to this bit becomes effective, SIOF change transmit data from TXD_SIO to 1 and initializes the following registers.

1. SITDR register
2. Transmit FIFO write pointer and read pointer
3. TCRDY, TFEMP, and TDREQ bits of SISTR register
4. TXE bit

SIOF is cleared automatically when this bit completes the reset, so 0 is always read from this bit.

Bit 1: TXRST	Description	
0	Transmitting operation is not reset	(Initial value)
1	Transmitting operation is reset	

Bit 0—Receiving Operation Reset (RXRST): Setting to this bit becomes effective immediately. After the setting 1 to this bit becomes effective, SIOF initializes the following registers and stop receiving from SIORXD.

1. SIRDR register
2. Receiving FIFO write pointer and read pointer
3. RCRDY, RFFUL, and RDREQ bits of SISTR register
4. RXE bit

SIOF is cleared automatically when this bit completes the reset, so 0 is always read from this bit.

Bit 0: RXRST	Description	
0	Receiving operation is not reset	(Initial value)
1	Receiving operation is reset	

20.2.7 FIFO Control Register (SIFCTR)

This register set trigger point and show current available area of Transmit and Receive FIFO. This register is initialized at power on reset or software reset.

Bit:	15	14	13	12	11	10	9	8
	TFWM2	TFWM1	TFWM0	TFUA4	TFUA3	TFUA2	TFUA1	TFUA0
Initial value:	0	0	0	1	0	0	0	0
R/W:	R/W	R/W	R/W	R	R	R	R	R

Bit:	7	6	5	4	3	2	1	0
	RFWM2	RFWM1	RFWM0	RFUA4	RFUA3	RFUA2	RFUA1	RFUA0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R	R	R	R	R

Bits 15 to 13—Transmit FIFO Water Mark (TFWM2 to TFWM0): The transfer request of the transmit FIFO is controlled by TDREQ bit of SISTR register. FIFO depth is always 16 steps, nevertheless the setting to these bits.

Bit 15: TFWM2	Bit 14: TFWM1	Bit 13: TFWM0	Description
0	0	0	The transfer request is submitted when the size of empty area in transmit FIFO is 16 steps (Initial value)
1		0	The transfer request is submitted when the size of empty area in transmit FIFO is larger than 12 steps
		1	The transfer request is submitted when the size of empty area in transmit FIFO is larger than 8 steps
	1	0	The transfer request is submitted when the size of empty area in transmit FIFO is larger than 4 steps
		1	The transfer request is submitted when the size of empty area in transmit FIFO is larger than 1 step

Bits 7 to 5—Receive FIFO Water Mark (RFWM2 to RFWM0): The transfer request of the receive FIFO is controlled by TDREQ bit of SISTR register. FIFO depth is always 16 steps, nevertheless the setting to these bits.

Bit 7: RFWM2	Bit 6: RFWM1	Bit 5: RFWM0	Description
0	0	0	The transfer request is submitted when the size of empty area of receive FIFO is larger than 1 step (Initial value)
1		0	The transfer request is submitted when the size of empty area of receive FIFO is larger than 4 steps
		1	The transfer request is submitted when the size of empty area of receive FIFO is larger than 8 steps
	1	0	The transfer request is submitted when the size of empty area of receive FIFO is larger than 12 steps
		1	The transfer request is submitted when the size of empty area of receive FIFO is at 16 steps

Bits 12 to 8—Transmit FIFO Usable Area (TFUA4 to TFUA0): TFUA shows usable number of words for CPU or DMAC to transfer from 00000 to 10000 (initial value).

Bits 4 to 0—Receive FIFO Usable Area (RFUA4 to RFUA0): RFUA shows usable number of words for CPU or DMAC to transfer from 00000 (initial value) to 10000.

20.2.8 Status Register (SISTR)

This register shows states of SIOF. Each bit of this register becomes interrupt source when 1 is set to corresponding register of SIIER register.

This register is initialized at power on reset or software reset.

Bit:	15	14	13	12	11	10	9	8
	—	TCRDY	TFEMP	TDREQ	—	RCRDY	RFFUL	RDREQ
Initial value:	0	0	0	0	0	0	0	0
R/W:	R*	R*	R*	R*	R*	R*	R*	R*

Bit:	7	6	5	4	3	2	1	0
	—	—	—	FSERR	TFOVR	TFUDR	RFUDR	RFOVR
Initial value:	0	0	0	0	0	0	0	0
R/W:	R*	R*	R*	R/W	R/W	R/W	R/W	R/W

Note: * 0 should be written into these bits. Otherwise the operation is unpredictable.

Bits 15, 11, and 7 to 5—Reserved

Bit 14—Transmit Control Data Ready (TCRDY): This bit displays condition of SITCR register. SIOF clears when any value is written to SITCR register. This bit becomes effective when 1 is written to TXE bit of SICTR register. SIOF issues control interrupt if interrupt issuing is allowed for this bit. Once any data are written to SICTR register with 0 of TCRDY bit, new data is overwritten to original data and original data of TXD_SIO will be lost.

Note: When using this bit, refer to note 2 in section 20.4, Notes on Use.

Bit 14: TCRDY	Description	
0	Disable writing into SITCR register	(Initial value)
1	Enable writing into SITCR register	

Bit 13—Transmit FIFO Empty (TFEMP): This bit is showing condition, SIOF clear by writing to SITDR register. This bit becomes effective when 1 is written to the TXE bit of SICTR register. SIOF issues control interrupt if interrupt issuing is allowed by this bit.

Bit 13: TFEMP	Description	
0	Transmit FIFO is not empty	(Initial value)
1	Transmit FIFO is empty	

Bit 12—Transmit Data Transfer Request (TDREQ): The transmit data transfer request is issued when empty area of transmit FIFO exceed the setting of TFWM bit of SIFCTR register. This bit is effective when 1 is written to TXE bit of SICTR register. This bit shows condition of transmit FIFO. SIOF clears this bit if empty area of transmit FIFO is smaller than the set value of TFWM bit of SIMDR register. SIOF issues a transmit interrupt if the interrupt issuing is allowed for this bit.

Bit 12: TDREQ	Description	
0	No transmit request exists.	(Initial value)
1	Transmit request exists.	

Bit 10—Receive Control Data Ready (RCRDY): This bit shows condition of SIRCR register. SIOF clears SIOF register when SIRCR register is read.

New received data will be overwritten to SIRCR register if valid data is received and written to SIRCR register while this bit shows 1. This bit is effective when 1 is written to RXE bit of SICTR register. SIOF issues a control interrupt if the interrupt issuing is allowed to bit.

Bit 10: RCRDY	Description	
0	Effective data is not stored in SIRCR register	(Initial value)
1	Effective data is stored in SIRCR register	

Bit 9—Receive FIFO Full (RFFUL): This bit shows condition of Receive FIFO. SIOF clears when SIRDR register is read. This bit is effective when 1 is written to RXE bit of SICTR register. SIOF issues a control interrupt when the interrupt issuing is allowed.

Bit 9: RFFUL	Description	
0	Receive FIFO is not full	(Initial value)
1	Receive FIFO is full	

Bit 8—Receive Data Transfer Request (RDREQ): The receive data transfer request is issued when effective received data in receive FIFO exceed the setting of RFWM bit of SIMDR register.

This bit is effective when 1 is written to RXE bit of SICTR register. This bit shows condition of receive FIFO. SIOF clears this bit if effective received data area in FIFO is smaller than the set

value of RFWM bit of SIMDR register. SIOF issues a receive interrupt if the interrupt issuing is allowed for this bit.

Bit 8: RDREQ	Description
0	Effective data in receive FIFO does not exceed setting of RFWM bit of SIMDR register (Initial value)
1	Effective data in receive FIFO exceeds setting of RFWM bit of SIMDR register

Bit 4—Frame Synchronization Error (FSERR): Frame synchronization error shows that next frame synchronize timing has come before data or control command are transferred. When frame synchronization error has occurred , SIOF transmits or receives data to the slots that are enable to transmit or receive data.

This bit becomes effective when 1 is written to TXE bit or RXE bit of SICTR register. This bit is cleared when 1 is written to this bit. SIOF issues the transmit interrupt when the interrupt issuing is allowed to this bit.

Bit 4: FSERR	Description
0	Frame synchronization error does not occur (Initial value)
1	Frame synchronization error occurs

Bit 3—Transmit FIFO Over Run (TFOVR): Transmit FIFO overrun shows that data are written to SITDR register when transmit FIFO is full. Written data is ignored when Transmit FIFO over run happens.

This bit is effective when 1 is written to TXE bit of SICTR register. This bit is cleared when 1 is written to this bit. SIOF issues the transmit interrupt when the interrupt issuing is allowed to this bit.

Bit 3: TFOVR	Description
0	Transmit FIFO over run does not occur (Initial value)
1	Transmit FIFO over run occurs

Bit 2—Transmit FIFO Under Run (TFUDR): Transmit FIFO under run shows that the load by data transfer from FIFO has occurred when transmit FIFO is empty.

SIOF repeats to send the data that was sent before when this under run has occurred.

This bit is effective when 1 is written to RXE bit of SICTR register. This bit is cleared when 1 is written to this bit. SIOF issues the transmit interrupt when the interrupt issuing is allowed to this bit.

Bit 2: TFUDR	Description	
0	Transmit FIFO under run does not occur	(Initial value)
1	Transmit FIFO under run occurs	

Bit 1—Receive FIFO Under Run (RFUDR): Receive FIFO under run shows that SIRDR register is read when receive FIFO is empty.

The data that has been read out from SIRDR is not guaranteed when this under run has occurred.

This bit is effective when 1 is written to RXE bit of SICTR register. This bit is cleared when 1 is written to this bit. SIOF issues the transmit interrupt when the interrupt issuing is allowed to this bit.

Bit 1: RFUDR	Description	
0	Receive FIFO under run does not occur	(Initial value)
1	Receive FIFO under run occurs	

Bit 0—Receive FIFO Over Run (RFOVR): Receive FIFO over shows write action has occurred to receive FIFO by SIOF, when receive FIFO is full. The received data disappears when receive FIFO overrun occurs.

This bit is effective when 1 is written to TXE bit or RXE bit of SICTR register. This bit is cleared when 1 is written to this bit. SIOF issues the transmit interrupt when the interrupt issuing is allowed to this bit.

Bit 0: RFOVR	Description	
0	Transmit over run does not generate	(Initial value)
1	Transmit over run generate	

20.2.9 Interrupt Enable Register (SIIER)

This register allows SIOF interrupt resources to issue interrupt to CPU. When 1 is written to each bit, corresponding interrupt is issued by SIOF. This register is initialized at power on reset, or software reset.

Bit:	15	14	13	12	11	10	9	8
	—	TCRDYE	TFEMPE	TDREQE	—	RCRDYE	RFFULE	RDREQE
Initial value:	0	0	0	0	0	0	0	0
R/W:	R*	R/W	R/W	R/W	R*	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
	—	—	—	FSERRE	TFOVRE	TFUDRE	RFUDRE	RFOVRE
Initial value:	0	0	0	0	0	0	0	0
R/W:	R*	R*	R*	R/W	R/W	R/W	R/W	R/W

Note: * 0 should be written into these bits. Otherwise the operation is unpredictable.

Bits 15, 11, and 7 to 5—Reserved

Bit 14—Transmit Control Data Ready Enable (TCRDYE)

Bit 14: TCRDYE	Description	
0	Disable interrupt of transmit control data ready	(Initial value)
1	Enable interrupt of transmit control data ready (control interrupt)	

Bit 13—Transmit FIFO Empty Enable (TFEMPE)

Bit 13: TFEMPE	Description	
0	Disable interrupt of transmit FIFO empty	(Initial value)
1	Enable interrupt of transmit FIFO empty (control interrupt)	

Bit 12—Transmit Data Transfer Request Enable (TDREQE)

Bit 12: TDREQE	Description	
0	Disable interrupt of transmit data transfer request enable	(Initial value)
1	Enable interrupt of transmit data transfer request enable (transmit interrupt)	

Bit10—Receive Control Data Ready Enable (RCRDYE)

Bit10: RCRDYE	Description	
0	Disable interrupt of receive control data ready	(Initial value)
1	Enable interrupt of receive control data ready (control interrupt)	

Bit 9—Receive FIFO Full Enable (RFFULE)

Bit 9: RFFULE	Description	
0	Disable interrupt of receive FIFO full	(Initial value)
1	Enable interrupt of receive FIFO full (control interrupt)	

Bit 8—Receive Data Transfer Request Enable (RDREQE)

Bit 8: RDREQE	Description	
0	Disable interrupt of receive data transfer request	(Initial value)
1	Enable interrupt of receive data transfer request (receive interrupt)	

Bit 4—Frame Synchronization Error Enable (FSERRE)

Bit 4: FSERRE	Description	
0	Disable interrupt of frame synchronization error	(Initial value)
1	Enable interrupt of frame synchronized error (error interrupt)	

Bit 3—Transmit FIFO Over Run Enable (TFOVRE)

Bit 3: TFOVRE	Description	
0	Disable interrupt of transmit FIFO over run	(Initial value)
1	Enable interrupt of transmit FIFO over run (error interrupt)	

Bit 2—Transmit FIFO Under Run Enable (TFUDRE)

Bit 2: TFUDRE	Description	
0	Disable interrupt of transmit FIFO under run	(Initial value)
1	Enable interrupt of transmit FIFO under run (error interrupt)	

Bit 1—Receive FIFO Under Run Enable (RFUDRE)

Bit 1: RFUDRE	Description
0	Disable interrupt of receive FIFO under run
1	Enable interrupt of receive FIFO under run (error interrupt)

Bit 0—Receive FIFO Over Run Enable (RFOVRE)

Bit 0: RFOVRE	Description
0	Disable interrupt of receive FIFO over run
1	Enable interrupt of receive FIFO over run (error interrupt)

20.2.10 Transmit Data Register (SITDR)

This register sets transmit data to SIOF. The data that has been set to this register is stored in transmit FIFO. This register is initialized at power on reset, software reset, or transmit reset.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SITDL 15	SITDL 14	SITDL 13	SITDL 12	SITDL 11	SITDL 10	SITDL 9	SITDL 8	SITDL 7	SITDL 6	SITDL 5	SITDL 4	SITDL 3	SITDL 2	SITDL 1	SITDL 0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

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

Bits 31 to 16—Transmit Data for Left Channel (SITDL15 to SITDL0): These bits set data transmitted from TXD_SIO as left channel data. The position for left channel side data are assigned as TDLA bit of SITDA register.

This bit becomes effective when 1 is set to TDLE bit of SITDAR register.

Bits 15 to 0—Transmit Data for Right Channel (SITDR15 to SITDR0): These bits set data transmitted from TXD_SIO as right channel data. The position for left channel side data are assigned as TDRA bit of SITDA register.

This bit becomes effective when 1 is set to TDRE bit of SITDAR register, and 0 is set to TLREP bit of SITDAR register.

20.2.11 Receive Data Register (SIRDR)

This register reads receive data of SIOF. The data from receive FIFO is stored in this register. This register is initialized at power on reset, software reset, or transmit reset.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SIRDL 15	SIRDL 14	SIRDL 13	SIRDL 12	SIRDL 11	SIRDL 10	SIRDL 9	SIRDL 8	SIRDL 7	SIRDL 6	SIRDL 5	SIRDL 4	SIRDL 3	SIRDL 2	SIRDL 1	SIRDL 0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

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

Bits 31 to 16—Receive Data for Left Channel (SIRDL15 to SIRDL0): These bits stores received data from RXD_SIO as left channel data. The position of left channel side data are assumed as the position what defined by RDLA bit of SIRDAR register.

These bits are effective when 1 is written to RDLE bit of SIRDAR register.

Bits 15 to 0—Receive Data for Right Channel (SIRDR15 to SIRDR0): These bits stores received data from RXD_SIO as right channel data. The position of left channel side data are assumed as the position what defined by RDRA bit of SIRDAR register.

These bits are effective when 1 is written to RDRE bit of SIRDAR register.

20.2.12 Transmit Control Data Register (SITCR)

This register sets the transmit control data for SIOF. Setting to this register is effective when 1*** is set to FL bit of SIMDR register. This register is initialized at power on reset, software reset, or transmit reset.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SITC0 15	SITC0 14	SITC0 13	SITC0 12	SITC0 11	SITC0 10	SITC0 9	SITC0 8	SITC0 7	SITC0 6	SITC0 5	SITC0 4	SITC0 3	SITC0 2	SITC0 1	SITC0 0
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

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SITC1 15	SITC1 14	SITC1 13	SITC1 12	SITC1 11	SITC1 10	SITC1 9	SITC1 8	SITC1 7	SITC1 6	SITC1 5	SITC1 4	SITC1 3	SITC1 2	SITC1 1	SITC1 0
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 31 to 16—Transmit Control Data for Channel 0 (SITC015 to SITC00): These bits stores data to be transfer as transmit control channel 0 data from TXD_SIO. The position of control data for channel 0 is determined by the setting of CD0A bit of SICDAR register.

This bit is effective when 1 is set to CD0E bit of SICDAR register.

Bits 15 to 0—SIOF Transmit Control Data for Channel 1 (SITC115 to SITC10): These bits stores data to be transfer as transmit control channel 1 command from TXD_SIO. The position of control data for channel 1 is determined by the setting of CD1A bit of SICDAR register.

This bit is effective when 1 is set to CD1E bit of SICDAR register.

20.2.13 Receive Control Data Register (SIRCR)

This register stores the received control data for SIOF. Setting to this register is effective when 1*** is set to FL bit of SIMDR register. This register is initialized at power on reset, software reset, or receive reset.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SIRC0 15	SIRC0 14	SIRC0 13	SIRC0 12	SIRC0 11	SIRC0 10	SIRC0 9	SIRC0 8	SIRC0 7	SIRC0 6	SIRC0 5	SIRC0 4	SIRC0 3	SIRC0 2	SIRC0 1	SIRC0 0
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

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SIRC1 15	SIRC1 14	SIRC1 13	SIRC1 12	SIRC1 11	SIRC1 10	SIRC1 9	SIRC1 8	SIRC1 7	SIRC1 6	SIRC1 5	SIRC1 4	SIRC1 3	SIRC1 2	SIRC1 1	SIRC1 0
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 31 to 16—Receive Data for Channel 0 (SIRC015 to SIRC00): Received data from RXD_SIO as control channel 0 data is stored to these bits. The position of control channel 0 data is determined by the setting of CD0A bit of SICDAR register.

This bit is effective when 1 is set to CD0E bit of SICDAR register.

Bits 15 to 0—Receive Data for Channel 1 (SIRC115 to SIRC10): Received data from RXD_SIO as control channel 1 data is stored to these bits. The position of control channel 1 data is determined by the setting of CD1A bit of SICDAR register.

This bit is effective when 1 is set to CD1E bit of SICDAR register.

20.3 Operation

20.3.1 Serial Clock

(1) Master/Slave

There are two modes as serial clock listed as below.

- Slave mode: SCK_SIO and SIOFSYNC is input.
- Master mode: SCK_SIO and SIOFSYNC is output.

(2) Baud Rate Generator (BRG)

At the SIOF master mode, serial clock is generated using the baud rate generator (BRG). The baud rate can be selected from 1/2 to 1/1024.

Figure 20.2 shows the serial clock supply system.

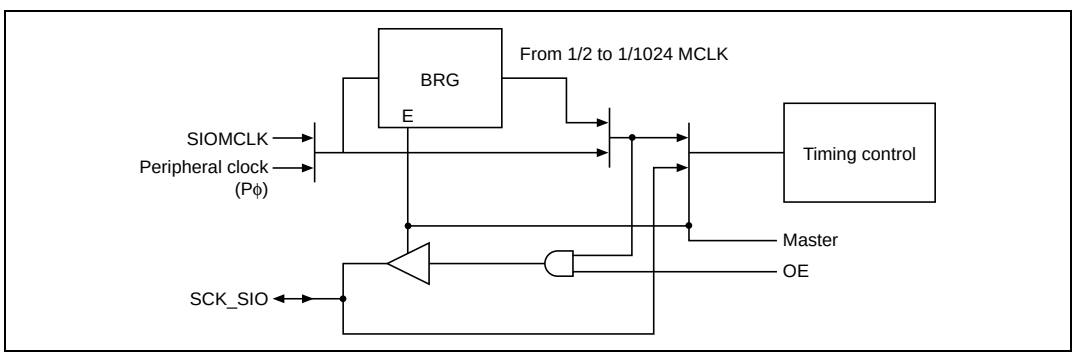


Figure 20.2 Serial Clock Supply System

Table 20.3 shows examples about serial clock frequency.

Table 20.3 Examples of SIOF Clock Frequency

	Sampling Rate		
Frame Length	**8 kHz**	**44.1 kHz**	**48 kHz**
32 bits	256 kHz	1.4112 MHz	1.536 MHz
64 bits	512 kHz	2.8224 MHz	3.072 MHz
128 bits	1.024 MHz	5.6448 MHz	6.144 MHz
256 bits	2.048 MHz	11.2896 MHz	12.288 MHz

Note: In Master mode, SCK_SIO continues to be output regardless of whether there is any data.

20.3.2 Serial Timing

(1) SIOFSYNC

SIOFSYNC is the frame sync signal, and supports the two modes shown below.

- The pulse with 1 bit width which shows the first of the sync pulse frame.
- The pulse with 1/2 frame width, which shows left channel in L/R stereo data as high and right channel as low.

Figure 20.3 shows synchronized timing as SIOFSYNC. Figure 20.3(a) shows the case for master mode 1, slave mode 1, and slave mode 2. Figure 20.3(b) shows the case for master mode 2.

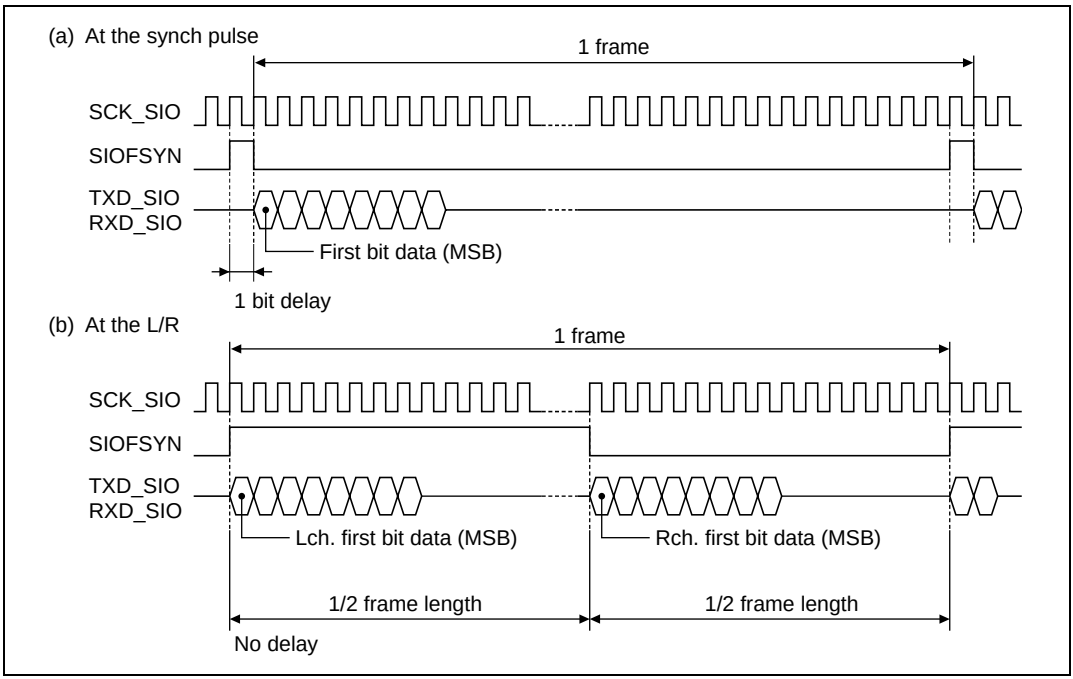


Figure 20.3 SIOF Serial Data Synchronized Timing

(2) Transmit or Receive Timing

Timing to SCK_SIO for transmitting TXD_SIO or receiving RXD_SIO can be chosen from the following two cases. Timing for transmitting or receiving is set into REDG bit in SIMDR register. In slave mode 1 or slave mode 2, only the sample at falling is valid.

- Sample at falling
- Sample at rising

Figure 20.4 shows the timing for transmitting or receiving.

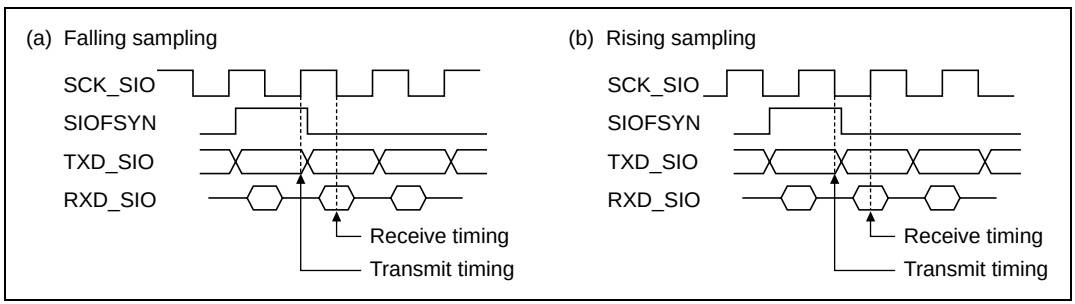


Figure 20.4 SIOF Transmit or Receive Timing

20.3.3 Transmit Data Format

SIOF transmit two kind of data shown below.

- Transmit or receive data: Transmit data of 8 bit/16 bit/16 bit stereo
- Control data: 16 bit length (interface by using the dedicated register)

(1) Transmit Mode

SIOF has four modes as transmit mode shown in table 20.4. Transmit mode is set to bits TRMD1 to TRMD0 in SIMDR register.

Table 20.4 Serial Transmit Mode

Transmit Mode	SIOFSYNC	Bit Delay	Control Data
Slave mode 1	Sync pulse	1 bit	Slot position
Slave mode 2	Sync pulse	1 bit	Secondary FS
Master mode 1	Sync pulse	1 bit	Slot Position
Master mode 2	L/R	Nothing	No support

(2) Frame Length

Frame length of SIOF transmitting is set by bits FL3 bit to FL0 in SIMDR register.

Table 20.5 shows relations between value and frame length.

Table 20.5 Frame Length

FL3, FL2, FL1, FL0	Slot Length	Bit/Frame	Support Transmit Data
00 - -	8	8	8 bit monaural
0100	8	16	8 bit monaural
0101	8	32	8 bit monaural
0110	8	64	8 bit monaural
0111	8	128	8 bit monaural
10 - -	16	16	16 bit monaural/stereo
1100	16	32	16 bit monaural/stereo
1101	16	64	16 bit monaural/stereo
1110	16	128	16 bit monaural/stereo
1111	16	256	16 bit monaural/stereo

(3) Slot Position

SIOF can set the position of transmit data, receive data, and control data (common in transmit/receive) in 1 frame independently by slot number. The following register are used for this setting.

- Transmit data: SITDAR register
- Receive data: SIRDAR register
- Control data: SICDAR register

Control data is effective when the slot length is 16 bit. The control data for transmission and reception is always assigned to the same slot.

20.3.4 Register Assignment for Transfer Data

(1) Transmit or Receive Data

Writing into/reading out of transmit/receive data is done for the following registers.

- Writing transmit data: SITDR register (32 bit access)
- Reading receive data: SIRDR register (32 bit access)

Figure 20.5 shows bit alignment of transmit or receive data and SITDR and SIRDR registers.

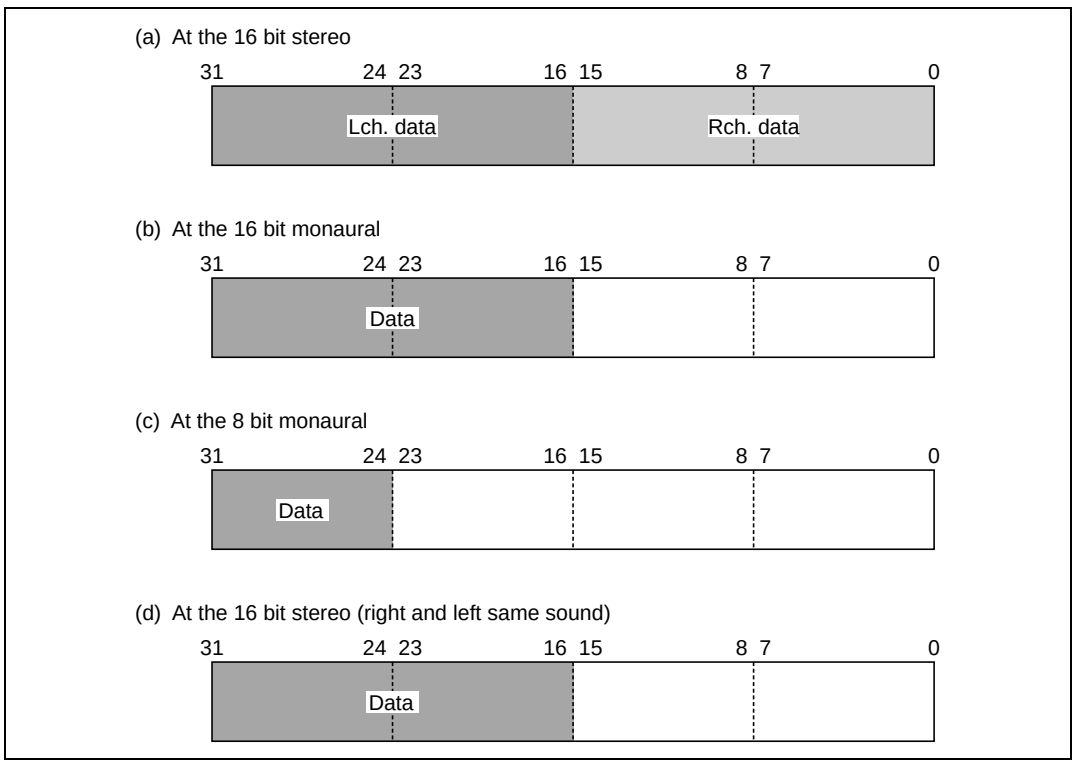


Figure 20.5 Transmit or Receive Data Bit Alignment

Note: In figure 20.5, only data portions that are shown by the oblique lines are transmitted or received as effective data.
Thus, it is necessary to transmit in byte for 8 bit data and in word 16 bit data.
The areas without the oblique lines are not the object to transmit or receive.

Monaural or stereo of transmit data is set with TDLE bit and TDRE bit of SILTDAR register. To choose monaural or stereo for receive, RDLE bit and RDRE bit of SIRDAR register must be set. The same sound for right and left in transmit data is set with TLREP bit of SITDAR register.

Table 20.6 shows establishment of sound mode for transmit data, table 20.7 shows establishment of sound mode for receive data, use only the left channel in case of monaural 8 bits transfer.

Table 20.6 Transmit Data Sound Mode

	Bit		
Mode	**TDLE**	**TDRE**	**TDREP**
Monaural	1	0	*
Stereo	1	1	0
Same sound for right and left	1	1	1

*: Don't care

Table 20.7 Receive Data Sound Mode

	Bit	
Mode	**RDLE**	**RDRE**
Monaural	1	0
Stereo	1	1

Note: Same mode for right and left don't exist in receive data.

(2) Control Data

Control data is read out/written into the following registers.

- Writing transmit control data: SITCR register (32 bit access)
- Reading receive control data: SIRCR register (32 bit access)

Figure 20.6 shows bit alignments of transmit or receive data and SITCR and SIRCR registers.

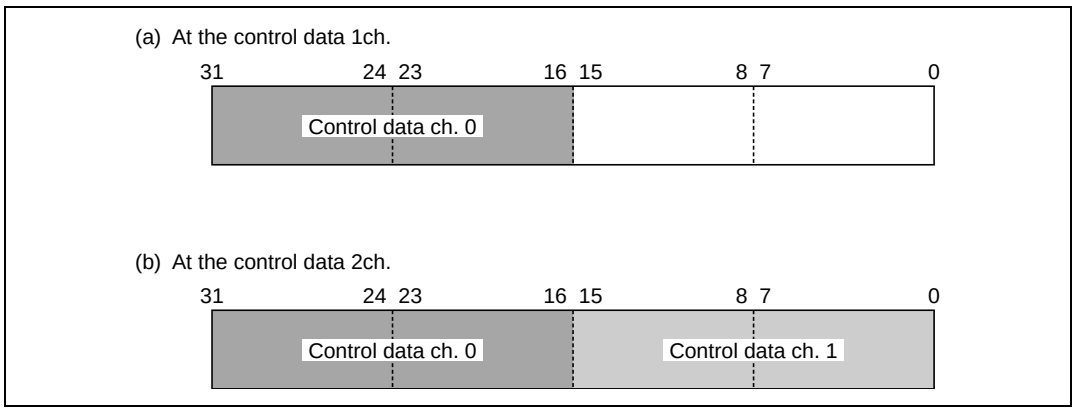


Figure 20.6 Control Data Bit Alignment

The channel number of control data is set with CD0E bit CD1E bit of SICDAR register.

Table 20.8 establishment of ch. number for control data. Use channel 0 when using only data one channel as control data.

Table 20.8 Control Data Channel Number Establishment

	Bit	
Ch. Number	**CD0E**	**CD1E**
1	1	0
2	1	1

20.3.5 Control Data Interface

Control data outputs the control command to CODEC and receive the state of CODEC. SIOF support the following two operations as an interface operation of control data.

- Control by the slot positions
- Control by secondary FS

Control data is effective when selecting 16 bit as data length and MSB first receive mode.

(1) Control by Slot Positions (Master Mode 1)

This is the method that dedicates the slot passion of control data in a frame to transmit or receive the control data.

Figure 20.7 shows a sample of control data interface timing by slot position.

Note: When using this method, peripheral clock (Pϕ) should be used as the master clock (Master Clock Select (MSSEL) = 1).

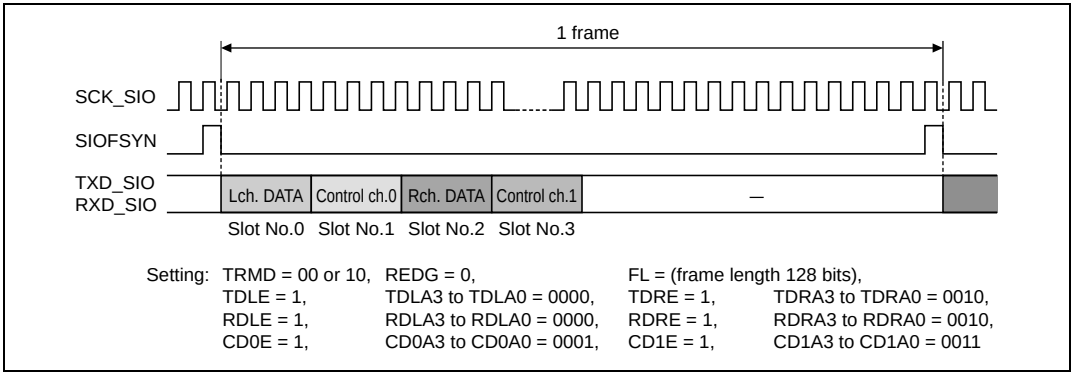


Figure 20.7 Control Data Interface (Slot Position)

(2) Control by Secondary FS

This is the method that CODEC, which outputs SIOFSYNC as a sync. pulse (FS), transmit or receive the control data by outputting the secondary FS used for transmit or receive for only control data after the period of 1/2 frame, which is different from the original FS output position.

Order of the control data interface as secondary FS are listed below.

- Normal data are sent as LSB=0 (compulsory is 0 by SIOF)
- Transmit data of LSB=1 at transmitting the control data
  (For 1 by SIOF reading to SITCR register)
- CODEC transmits secondary FS
- SIOF synchronizes secondary FS and transmit or receive (storing into SIRCR register) the control data (setting data in SITCR register)

Figure 20.8 shows timing of control data interface by secondary FS.

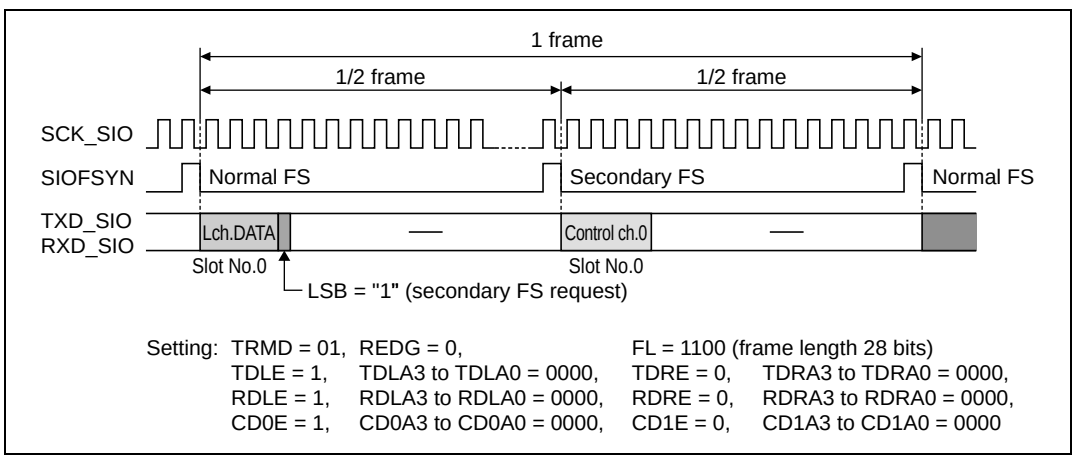


Figure 20.8 Control Data Interface (Secondary FS)

20.3.6 FIFO

(1) Outline

Features of SIOF transmit or receive FIFO are listed as below.

- Capacity of 32 bits × 16 stages for each of transmission and reception
- Pointer is updated by a read/write cycle for all of the access sizes of CPU and DMAC
  (It is impossible to separate one stage access to multiple times)
- Access cycle number is always 2 cycles (P bus cycle) for all of the access sizes.

(2) Transmit Request

Transmit request of FIFO is displayed in the following two bits of SISTR register.

- Transmit request: TDREQ (transmit interrupt factor)
- Receive request: RDREQ (receive interrupt factor)

It is possible to set independently the condition for each of submitting transmit request of transmit or receive FIFO. Condition of transmit request are set to bits TFWM2 to TFWM0 in SIFCTR register and transfer request of receive FIFO are set to bits RFWM2 to RFWM0.

Table 20.9 shows transmit request submit condition, and table 20.10 shows receive request submit condition.

Table 20.9 Transmit Request Submit Condition

TFWM2 to TFWM0	Request Stage Number	Transmit Request Submit	Used Area
000	1	16 stages empty area	Small
100	4	Over 12 stages empty area	↕
101	8	Over 8 stages empty area	
110	12	Over 4 stages empty area	
111	16	Over 1 stage empty area	Large

Table 20.10 Receive Request Submit Condition

RFWM2 to RFWM0	Request Stage Number	Receive Request Submit	Used Area
000	1	Over 1 stage effective area	Small
100	4	Over 4 stages effective area	
101	8	Over 8 stages effective area	
110	12	Over 12 stages effective area	
111	16	16 stages effective area	Large

When the data area or empty area exceed the above stage number, FIFO capacity always can be used 16 stages. Therefore, over flow or under flow error are submitted when the data area, or empty area excesses 16 stages.

Even if FIFO is not empty or full, the transmit request is cancelled when the above conditions become not to be satisfied.

(3) Showing of Stage Number

The state of using transmit or receive FIFO is displayed in the following registers.

- Transmit FIFO: Shows stage number of empty area to bits TFUA4 to TFUA0 in SIFCTR register
- Receive FIFO: Shows stage number of effective data to bits RFUA4 to RFUA0 of SIFCTR register

The above contents show the number of data which CPU or DMAC can transfer.

20.3.7 Procedures for Transmit or Receive

(1) Transmitting in Master

Figure 20.9 shows an example of setting and operation of transmitting in master.

No.	Time chart	Setting content of SIOF	SIOF operation
1	Start Settting of SIMDR register, SIMCR register, SITDAR register, SIRDAR register, SICDAR register, SIFCTR register	Setting of operation mode, serial clock, slot position of transmit or receive data, slot position of control data and limit of FIFO request	
2	"1" is set to SCKE bit of SICTR register	Set the beggin to operation of baud rate generator	
3	SCK_SIO begin to transmit		Transmit serial clock
4	"1" is set to FSE bit of SICTR register	Set the begin to transmit of frame synchronized signal	Transmit frame synchronized signal
5	"1" is set to TXE bit of SICTR register	Set the transmit enable	Submit the transmit request
6	TDREQ = 1? N Y		
7	Setting of SITDR register	Set the transmit data	
8	Synchronized to SIOFSYNC, content of SITDR is setn from TXD_SIO		Transmit
9	Finish to transmit? N Y "0" is set to TXE bit of SICTR register End	Set to transmit disable	Finish to transmit

Figure 20.9 Example of Transmit Operation in Master

(2) Receiving in Master

Figure 20.10 shows an example of receiving and operation in master.

No.	Time chart	Setting content of SIOF	SIOF operation
1	Start Settting of SIMDR register, SIMCR register, SITDAR register, SIRDAR register, SICDAR register, SIFCTR register	Setting of operation mode, serial clock, slot position of transmit or receive data, slot position of control data and limit of FIFO request	
2	"1" is set to SCKE bit of SICTR register	Set the beggin to operation of baud rate generator	
3	SCK_SIO begin to transmit		Transmit serial clock
4	“1” is set to FSE bit of SICTR register	Set the begin to transmit of frame synchronized signal	Transmit frame synchronized signal
5	"1" is set to RXE bit of SICTR register	Set the transmit enable	
6	Synchronized to SIOFSYNC receive data from RXD_SIOR is stored to SIRDR		Receive request is submitted by limit of reveive FIFO
7	RDREQ = 1? N Y		Receive
8	Setting of SIRDR register	Reading of receive data	
9	Finish to transmit? N Y "0" is set to RXE bit of SICTR register End	Set to transmit disable	Finish to receive

Figure 20.10 Example of Receive Operation in Master

(3) Transmitting in Slave

Figure 20.11 shows an example of transmitting and operation in slave.

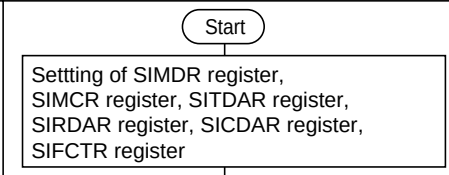
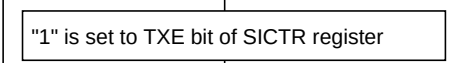
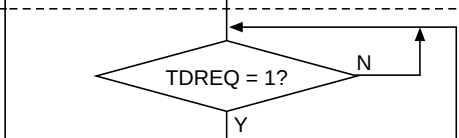
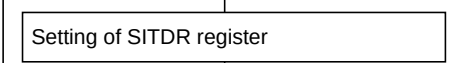
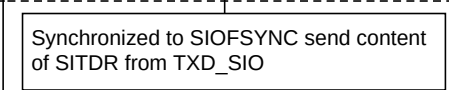
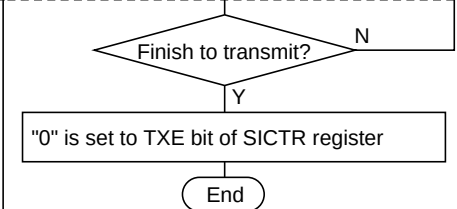
No.	Time chart	Setting content of SIOF	SIOF operation
1		Setting of operation mode, serial clock, slot position of transmit or receive data, slot position of control data and limit of FIFO request	
2		Set transmit enable	Disable to transmit when frame synchronized transmit submit the transmit request
3			
4		Set the transmit data	
5			Transmit
6		Set to transit disable	Finish to transmit

Figure 20.11 Example of Transmit Operation in Slave

(4) Receiving in Slave

Figure 20.12 shows an example of receiving and operation in slave.

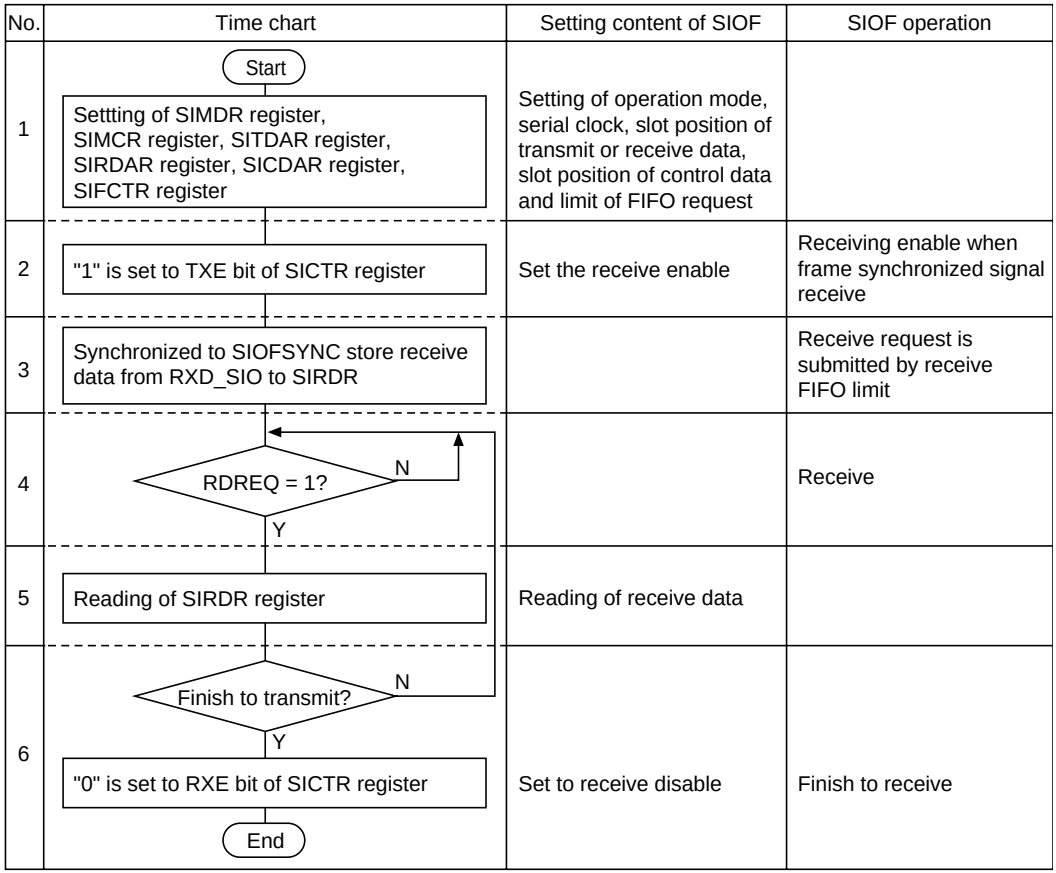


Figure 20.12 Example of Receive Operation in Slave

(5) Transmit or Receive Reset

SIOF can reset independently the transmit and receive portions by setting 1 in the following bits.

- Transmit reset: (TXRST bit of SICTR register)
- Receive reset: (RXRST bit of SICTR register)

Table 20.11 shows the initialized contents by transmit or receive reset.

Table 20.11 Transmit or Receive Reset

Reset Type	Initialized Register or Bits
Transmit reset	SITDR register
	Transmit FIFO write pointer
	Transmit FIFO read pointer
	TCRDY, TFEMP, and TDREQ bits in SISTR register
	TXE bit in SICTR register
Receive reset	SIRDR register
	Receive FIF0 write pointer
	Receive FIF0 read pointer
	RCRDY, RFFUL, and RDREQ bits in SISTR register
	RXE bit in SICTR register

(6) Module Stop

SIOF stops transmit or receive operation with holding the contents of all registers at module stop. Issue the transmit or receive reset, when the transmit or receive operation is not executed directly after the module stop.

20.3.8 Interrupt

SIOF has the following four types of interrupt. These types are reflected to IRR4 register in interrupt controller (INTC).

- Transmit interrupt (TXI)
- Receive interrupt (RXI)
- Control interrupt (CCI)
- Error interrupt (ERI)

(1) Interrupt Factor

Each interrupt is submitted by its multiple factors. Each factor is shown in SIOF register as SIOF states. Table 20.12 shows SIOF interrupt factors.

Table 20.12 SIOF Interrupt Factors

No.	Type	Bit	Function	Explanation of Interrupt
1	Transmit (TXI)	TDREQ	Transmit FIFO send request	Data of over setting is stared to transmit FIFO
2	Receive (RXI)	RDREQ	Receive FIFO send request	Data of over setting is stared to receive FIFO
3	Control (CCI)	TCRDY	Transmit control data ready	Enable writing into transmit control data register
4		RCRDY	Receive control data ready	Valid data is stored in receive control data register
5		TFEMP	Transmit FIFO empty	Transmit FIFO is empty
6		RFFUL	Receive FIFO full	Receive FIFO is full
7	Error (ERI)	TFUDF	Transmit FIFO under flow	The serial data send timing comes when transmit FIFO is empty
8		TFOVF	Transmit FIFO over flow	Write to transit FIFO when transmit FIFO is full
9		RFOVF	Receive FIFO over flow	Receive serial data when receive FIFO is full
10		RFUDF	Receive FIFO under flow	Read receive FIFO when receive FIFO is empty
11		FSERR	F3 error	Serial signal is received before setting bit number (at slave)

It depends on setting of SIIER register whether or not an interrupt corresponding to each interrupt factor is submitted or not depend. SIOF submits each interrupt when interrupt factor that 1 is set to corresponding bit of SIIER register is set to 1.

(2) Transmit/Receive Interrupt Flag

Transmit or receive interrupt requests INTC or DMAC to accept the interruption through the interrupt flag, which is generated from the value of TDREQ bit and RDREQ bit in SISTR register.

Table 20.13 shows the setting conditions for the transmit or receive interrupt flag.

Table 20.13 Setting Conditions for the Transmit or Receive Interrupt Flag

	Setting Conditions	Resetting Conditions
Transmit interrupt flag	TDREQ in SISTR register = 1	• TDREQ in SISTR register = 0 • Acknowledge from DMAC
Receive interrupt flag	RDREQ in SISTR register = 1	• RDREQ in SISTR register = 0 • Acknowledge from DMAC

(3) Operations in Case of Error

SIOF executes the following operations for the errors which are shown in SISTR as status.

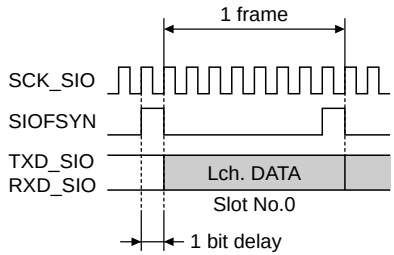
- Transmit FIFO under run (TFUDR): The data that was transmitted directly before is sent again.
- Transmit FIFO over run (TFOVR): The contents of transmit FIFO is protected, the written data that became to over flow is ignored.
- Receive FIFO over run (RFOVR): Data that became to over flow is disposed and vanished.
- Receive FIFO under run (RFUDR): Data that is read as final data is output on bus. (indefinite in specification)
- FS error (FSERR): Internal counter is reset according to the sync. signal that became to error.

20.3.9 Transmit or Receive Timing

Figures 20.13 to 20.19 show examples of serial transmit or receive of SIOF.

(1) A Case of 8 bits Monaural (No.1)

Sync pulse method, falling edge sampling, transmit data and receive data are assigned to slot No. 0, frame length is 8 bits.

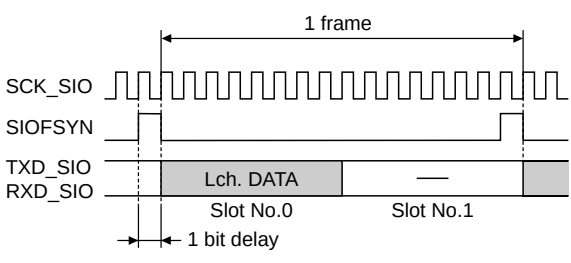


Setting:	TRMD = 00 or 10,	REDG = 0,	FL = 0000 (frame length 16 bits),	
	TDLE = 1,	TDLA3 to TDLA0 = 0000,	TDRE = 0,	TDRA3 to TDRA0 = 0000,
	RDLE = 1,	RDLA3 to RDLA0 = 0000,	RDRE = 0,	RDRA3 to RDRA0 = 0000,
	CD0E = 1,	CD0A3 to CD0A0 = 0000,	CD1E = 0,	CD1A3 to CD1A0 = 0000

Figure 20.13 Transmit or Receive Timing (8 bits monaural—1)

(2) A Case of 8 bits Monaural (No.2)

Sync pulse method, falling edge sampling, transmit data and receive data are assigned to slot No.0, frame length is 16 bits.



Setting:	TRMD = 00 or 10,	REDG = 0,	FL = 0100 (frame length 16 bits),	
	TDLE = 1,	TDLA3 to TDLA0 = 0000,	TDRE = 0,	TDRA3 to TDRA0 = 0000,
	RDLE = 1,	RDLA3 to RDLA0 = 0000,	RDRE = 0,	RDRA3 to RDRA0 = 0000,
	CD0E = 0,	CD0A3 to CD0A0 = 0000,	CD1E = 0,	CD1A3 to CD1A0 = 0000

Figure 20.14 Transmit or Receive Timing (8 bits monaural—2)

(3) A Case of 16 bits Monaural (No.1)

Sync pulse method, falling edge sampling, transmit data and receive data are assigned to slot No. 0, frame length is 64 bits.

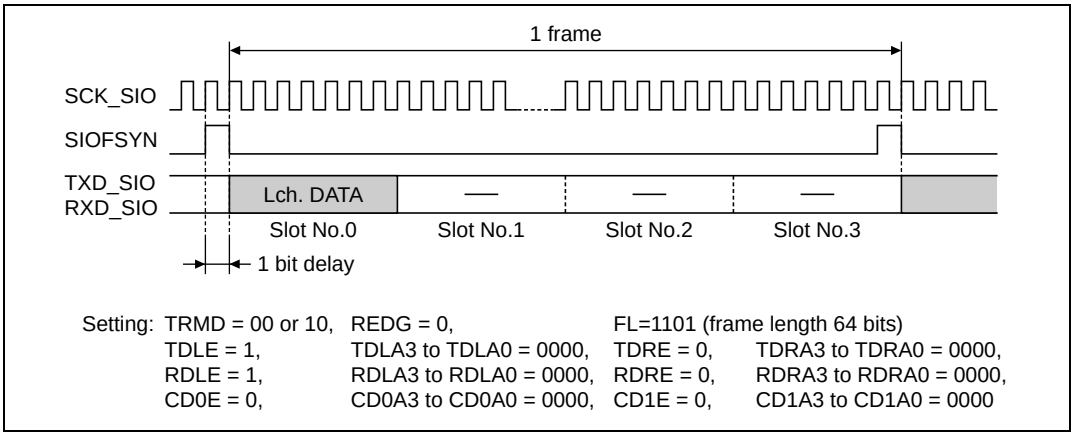


Figure 20.15 Transmit or Receive Timing (16 bits monaural—1)

(4) A Case of 16 bits Stereo (No.1)

L/R method, rising edge sampling and Lch. data are assigned to slot No. 0, Rch.data is assigned to slot No. 1, and frame length is 32 bits.

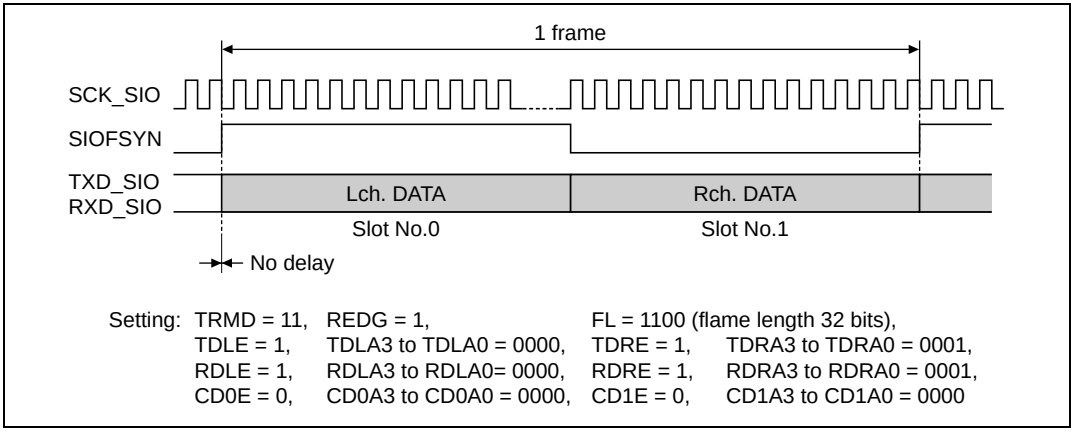


Figure 20.16 Transmit or Receive Timing (16 bits stereo—1)

(5) A Case of 16 bits Stereo (No.2)

L/R method, rising edge sampling and Lch.transmit data are assigned to slot No. 0, Lch. receive data are assigned to slot No.1, Lch. receive data are assigned to slot No. 2, Rch. receive data is assigned to slot No. 3, and frame length is 64 bits.

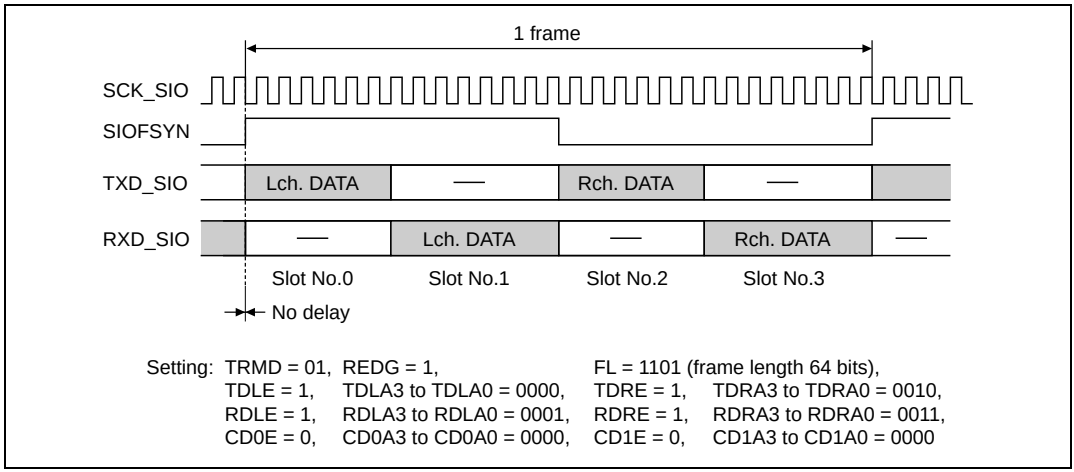


Figure 20.17 Transmit or Receive Timing (16 bits stereo—2)

(6) A Case of 16 bits Stereo (No. 3)

Sync pulse method, falling edge sampling and Lch. data are assigned to slot No. 0, Rch. data is assigned to slot No. 2, control ch. data 0 is assigned to slot No. 1, control ch. data 0 is assigned to slot No. 3, and frame length is 128 bits.

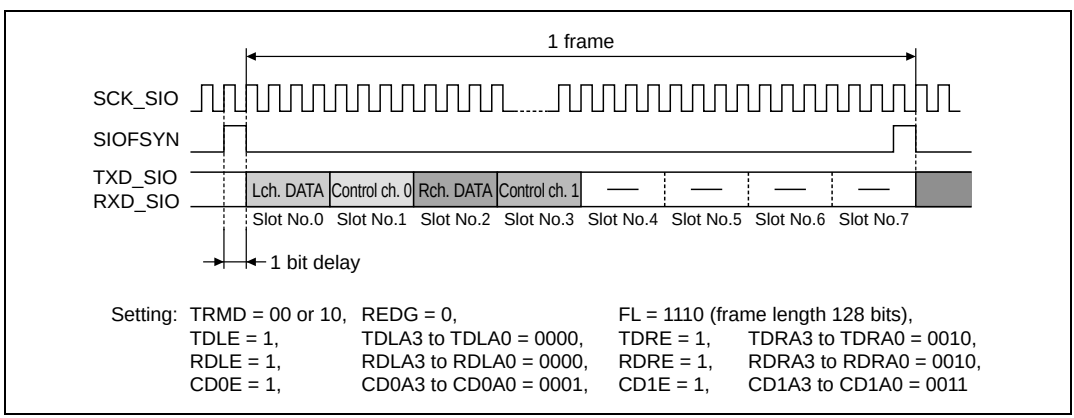


Figure 20.18 Transmit or Receive Timing (16 bits stereo—3)

(7) A Case of bits Monaural (No. 2)

Sync pulse method, falling edge sampling and secondary FS are requested, Lch. data is assigned to slot No. 0, control ch. data 0 are assigned to slot No. 0, and frame length is 128 bits.

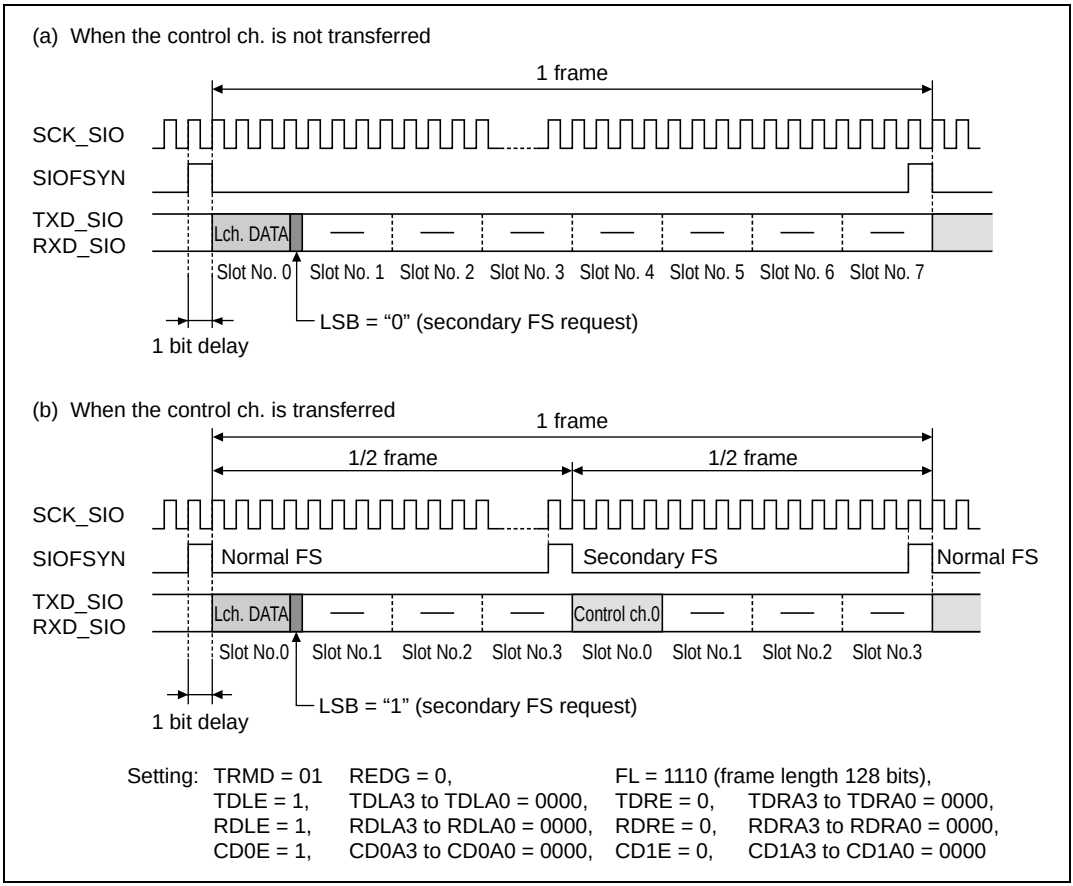


Figure 20.19 Transmit or Receive Timing (16 bits monaural—2)

20.4 Usage Notes

Note the following when using the SIOF.

For details on using versions previous to the SH7727B please refer to 20.4.1, Notes on Using the SIOF with Versions Previous to the SH7727B, in addition to the notes below.

1. Using the transmit function in sleep mode

   If transmission is enabled when data has already been written to the transmit FIFO, one or two of the initial data bytes may be lost.

   Therefore, data should not be written to the transmit FIFO before enabling transmission.

2. Using control data transmission/reception consecutively on control data interface (secondary FS position)

   The TCRDY value may become 1 before transmit control data is sent, and if the next control data is written to the control data register at this point, the control data waiting to be sent will be overwritten and erased.

   At this time, also, the control sequence is disrupted and the SIOF switches around the primary FS and secondary FS, with the result that transmission/reception of data and control data can no longer be performed normally.

   The control data register should therefore be written to after transmit control data has been sent.

   Example:

   Reference RCRDY, and write to the control data register when RCRDY is 1.

   After transmit control data has been written, it is essential to read the receive control register (SIRCR) and clear RCRDY.

3. DMA transfer

   Do not use 16-byte DMA transfer. (See section 14.3.4, DMA Transfer Types.)

4. Access from the CPU

   When performing access from the CPU, do not access the SIOF's transmit/receive FIFO consecutively, but instead insert an access to somewhere else between SIOF transmit/receive FIFO accesses.

5. Transmit/receive FIFO underflow

   If the transmit/receive FIFO underflows during a transmit/receive operation, control of the SIOF's transmit/receive FIFO may fail and data may be lost.

   To prevent this, either set a watermark so that underflow does not occur, or execute a transmit reset (TXRST) or receive reset (RXRST) when an empty interrupt is generated.

6. Transmit/receive reset execution

   When using the SIOF again after a transmit/receive operation ends, or after erroneous operation occurs, first execute a transmit reset (TXRST) or receive reset (RXRST).

7. Using the control data interface (slot position)

   Loss of transmit or control data may occur if transmit and control data are mixed within a single frame during a transmit operation in any of the following modes:

   (1) Master mode with an external clock (SIOMCLK) used as the master clock

   (2) Slave mode

   Therefore, the control data interface (slot position) should be used under the following conditions:

   (1) Master mode

   (2) Master clock: Peripheral clock (PCLK)

8. Serial IO (SIOF) reception operation

   During serial IO (SIOF) operation, if reception is performed with a slot length of 8 bits and LSB first, unwanted data is added at the start of the reception data, FIFO storage is delayed one byte at a time, and the final portion of the data remains in the shift register. To prevent this, use one of the three methods described below.

   (1) Perform reception using a slot length of 8 bits and LSB first, and read and discard the unwanted data.

   Read and discard the unwanted data at the start of the reception data. Then, input a dummy FS after the final portion of data so that the real final portion of data is stored in the FIFO. This will ensure that reception operates correctly when a slot length of 8 bits and LSB first is used.

   (2) Perform reception using a slot length of 8 bits and MSB first, then use software processing to convert the data to LSB-first format.

   Data reception operates correctly when a slot length of 8 bits and MSB first is used. After receiving the data in MSB-first format, use software processing to convert the data read from the FIFO from MSB-first to LSB-first format. The result can then be used as 8-bit slot length LSB-first data.

   (3) Perform reception using a slot length of 16 bits and LSB first, and read only the required data.

   Data reception operates correctly when a slot length of 16 bits and LSB first is used. Then one of the following two methods can be used to obtain data that can be used as 8-bit slot length LSB-first data.

   (a) Make settings on the transmitting side so that only the upper 8 bits are used for actual data. Then, after the 16-bit data is received by the SH7727, extract the upper 8 bits and use it as 8-bit slot length LSB-first data.

(b) Transmit two 8-bit units of data at once. Then, after the 16-bit data is received by the SH7727, separate the upper and lower 8-bit portions and treat them as two 8-bit units of data. They can then be used as 8-bit slot length LSB-first data.

20.4.1 Notes on Using the SIOF with Versions Previous to the SH7727B

Notes

When using SIOF, the following phenomenon may occur.

(1) During SIOF transmit with DMA transferring, SIOF may suddenly stop internal DMA transfer request, then underflow error ocurs and transmit operation stops.
(2) During SIOF transmit and transmit FIFO empty, underflow or overflow, some data transmit may fail, depending on the timing relationship between transmit FIFO write in and read out.
(3) During SIOF receive operation with DMA internal peripheral module request mode, some data receive may fail, due to unexpected overflow errors caused on the manner that only one data transfer request exceeding watermark of receive FIFO.
(4) During receive operation, some receive data may fail when a write occurs at reading from receive FIFO.
(5) During SIOF receive operation and receive FIFO empty, underflow or overflow, some data receive may fail, depending on the timing relationship between receive FIFO write in and read out. In this case, the statuses of full, underflow, and overflow may not be reflected to flags.

Countermeasures

Countermeasures to deal with software using SIOF.

(1) Notes (1) and (2)

With referring to transmit FIFO transfer request interrupt (SIFTXI) caused by under watermark of transmit FIFO, write the exactly same number of data with that of transmit FIFO empty slots with DMA auto request.

At that time, make sure to set the watermark value so as not to occur transmit FIFO empty nor underflow when transmit operation.

Example: When 12 empty slots are set to transmit FIFO, write 12 data to transmit FIFO with DMA auto request by transmit FIFO transfer interrupt (SIFTXI).

(2) Notes (3), (4), and (5)

With referring to receive FIFO transfer request interrupt (SIFRXI) caused by over watermark of receive FIFO, read the (valid number – 2) of data from receive FIFO with DMA auto request.

Read operation shall be done before next receive data write.

At that time, make sure to set the watermark value so as not to receive FIFO full nor overflow when receive operation.

Example: When 12 empty slots are set to receive FIFO, read 12 data from receive FIFO with DMA auto request by receive FIFO transfer interrupt (SIFRXI).

Etc.

(1) Not to use DMA 16 bytes transfer. (See section 14.3.4, DMA Transfer Types.)

(2) Recommend DMA auto request for SIOF access. When from CPU, not to use continuous access.

(3) When newly use SIOF after transmit/receive operation, proceed transfer operation after transmit reset (TXRST) or receive reset (RXRST).

Section 21 Analog Front End Interface (AFEIF)

21.1 Overview

This LSI has an AFE interface that supports softwaremodem. This AFE interface can efficiently execute the modem processing, because it includes 128 stages of FIFO for each of transmission and reception. This AFE interface also includes the interface to data access arrangement (DAA) such as dial pulse generator circuit and ringing detection. Therefore, it is possible to establish a modem system with a minimum of hardware.

21.1.1 Features

- Serial interface with FIFO
- Clock synchronized serial interface
- Transmit/receive FIFO size is 16 bits (maximum) × 128 words
- Transmit/receive interrupt threshold size is programmable
- Dial pulse generator circuit is included
- Ringing detection (calling signal) function is included

21.1.2 Block Diagram

Figure 21.1 shows a block diagram of AFEIF.

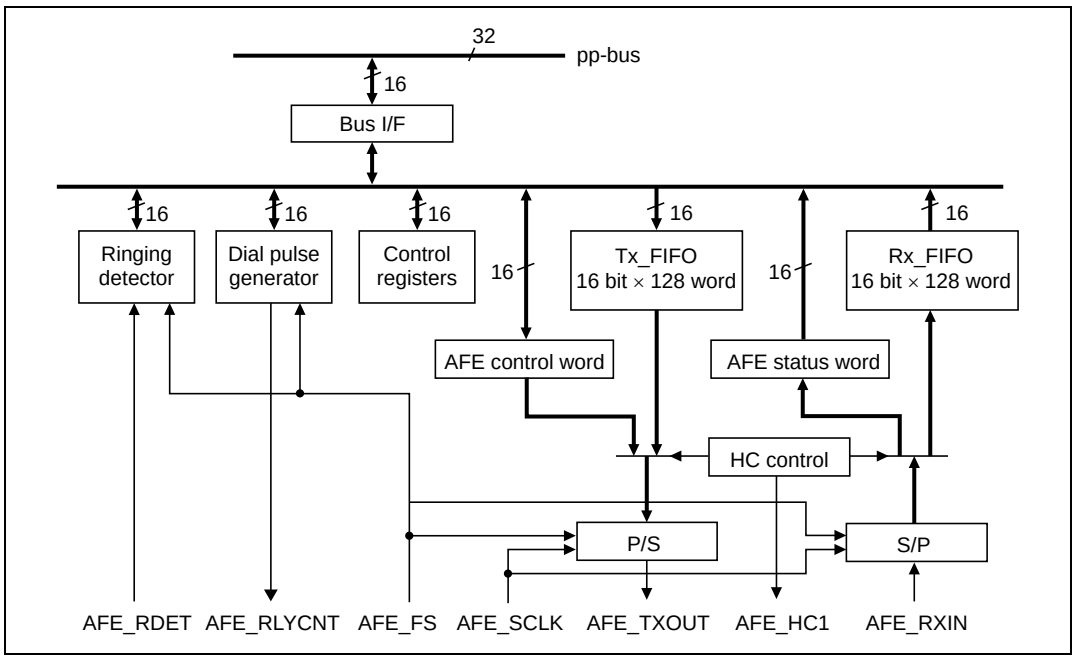


Figure 21.1 Block Diagram of AFE Interface

21.1.3 Pin Configuration

Table 21.1 shows the pins for AFE interface.

Table 21.1 Pins for AFE Interface

Pin No.	Name	I/O	Function
121	AFE_RDET	I	Ringing signal input
114	AFE_RLYCNT	O	On-hook control signal
116	AFE_SCLK	I	Shift clock
118	AFE_FS	I	Frame synchronization signal
119	AFE_RXIN	I	Serial receive data
113	AFE_HC1	O	AFE hardware control signal
120	AFE_TXOUT	O	Serial transmit data

21.1.4 Register Configuration

Table 21.2 shows registers for AFEIF. Byte access registers to these is inhibited.

Table 21.2 AFEIF Registers

Register Name	Abbre-viation	R/W	Initial Value	Address	Access Size
AFEIF control register 1	ACTR1	R/W	H'0000	H'04000180	16
AFEIF control register 2	ACTR2	R/W	H'0000	H'04000182	16
AFEIF status register 1	ASTR1	R/W	H'0F0A	H'04000184	16
AFEIF status register 2	ASTR2	R/W	H'0300	H'04000186	16
Make ratio count register	MRCR	R/W	H'0000	H'04000188	16
Minimum pose count register	MPCR	R/W	H'0000	H'0400018A	16
Dial number queue	DPNQ	R/W	H'0000	H'0400018C	16
Ringing pulse counter	RCNT	R	H'0000	H'0400018E	16
AFE control data register	ACDR	R/W	H'0000	H'04000190	16
AFE status data register	ASDR	R	H'0000	H'04000192	16
Transmit data FIFO port	TDFP	W	Undetermined	H'04000194	32 (16)
Receive data FIFO port	RDFP	R	Undetermined	H'04000198	32 (16)

21.2 Register Description

21.2.1 AFEIF Control Register 1 and 2 (ACTR1, ACTR2)

ACTR is the control register for AFEIF and is composed of ACTR1 and ACTR2. ACTR1 is mainly used for FIFO control commands. ACTR2 is used for AFE control commands and DAA control commands.

(1) AFEIF Control Register 1 (ACTR1)

Bit:	15	14	13	12	11	10	9	8
	HC	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R	R	R	R	R	R	R

Bit:	7	6	5	4	3	2	1	0
	DLB	—	—	FFSZ2	FFSZ1	FFSZ0	TE	RE
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R	R	R/W	R/W	R/W	R/W	R/W

Bits 14 to 8, 6, and 5—Reserved

Bit 15—AFE Hardware Control Bit (HC): This bit controls AFE. AFE_HC1 signal is made to high directly often the next serial transmit data transfer, when this bit is written to 1. Then ACDR data (AFE control word) is transferred by founding the second AFE.FS. AFEIF module automatically makes AFE_HC1 signal to low and HC bit to 0, directly after transferring the AFE control word. See section 21.3.2, AFE Interface for more detail about AFE control sequences.

Bit 7—FIFO Digital Loop Back (DLB)

Bit 7: DLB	Description
0	Normal operation (Initial value)
1	Digital loop back between Tx FIFO and Rx FIFO is performed. In this time the transmit data is output to AFE_TXOUT, too.

Bits 4 to 2—FIFO Interrupt Size Set 2 to 0 (FFSZ2 to FFSZ0): Specifies the size of FIFO. FIFO size to generate interrupt (TFE, RFF, THE, and RHF) is assigned as follows:

Bit 4: FFSZ2	Bit 3: FFSZ1	Bit 2: FFSZ0	Description			
			FIFO Size	TFE/RFF	THE/RHF	
0	0	0	128	128 empty/full	64 empty/full	(Initial value)
		1	64	64 empty/full	32 empty/full	
	1	0	32	32 empty/full	16 empty/full	
		1	16	16 empty/full	8 empty/full	
1	0	0	8	8 empty/full	4 empty/full	
		1	4	4 empty/full	2 empty/full	
	1	0	2	2 empty/full	1 empty/full	
		1	96	96 empty/full	48 empty/full	

Bit 1—Tx Enable (TE)

Bit 1: TE	Description
0	Transmit operation is disabled. The READ pointer of FIFO is stacked to the first address. WRITE pointer is reset when 0 is written to this bit. TFEM and THEM bits in ASTR1 is set to 1 at that time. (Initial value)
1	Transmit operation is enabled.

Bit 0—Rx Enable (RE)

Bit 0: RE	Description
0	Receive operation is disabled. The WRITE/READ pointer is fixed to the first address. RFFM and RHFM bits in ASTR1 is set to 1 at that time.
1	Receive operation is enabled

(2) AFEIF Control Register 2 (ACTR2)

Bit:	15	14	13	12	11	10	9	8
	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit:	7	6	5	4	3	2	1	0
	—	—	—	DPST	PPS	RCEN	—	RLYC
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R/W	R/W	R	R/W

Bits 15 to 5, and 1—Reserved

Bit 4—Dial Pulse Start (DPST): Start bit of dial pulse. Dial number within the DPNQ register is output to AFE_RLYCNT as specified by PPS, MRCR and MPCR. After all dial number is output , DPE interrupt is generated to modify the DPST bit to 0. See section 21.3.3, DAA Interface for more detail about dial pulse output sequence. Take care that AFE_RLYCNT must be “H” to enable dial pulse generating circuit

Bit 3—Dial Pulse Duration Set (PPS)

Bit 3: PPS	Description	
0	10PPS	(Initial value)
1	20PPS	

Bit 2—Ringing Counter Enable (RCEN)

Bit 2: RCEN	Description	
0	Stop Ringing Counter	(Initial value)
1	Start Ringing Counter	

Note: See section 21.3.3, DAA Interface for more detail about how to count.

Bit 0—Relay Control (RLYC): The signal controls Hook Relay.

Bit 0: RLYC	Description	
0	On hook state. AFE_RLYCNT goes Low Level.	(Initial value)
1	Off hook state. AFE_RLYCNT goes High Level.	

21.2.2 Make Ratio Count Register (MRCR)

MRCR is the counter that specifies make ratio of dial pulse. Make interval is specified with AFE_FS as base clock of 9,600 Hz.

Pulse signal is not output when an invalid data (a data that is greater than 1E0H in case of PPS = 1 (20 pps), or a data that is greater than 3C0H in case of PPS = 0 (10 pps)) was input.

Bit:	15	14	13	12	11	10	9 to 0
	—	—	—	—	—	—	MRCR
Initial value:	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W

21.2.3 Minimum Pause Count Register (MPCR)

MPCR is a counter that sets the dial number interval of the dial pulse. The interval is specified with AFE_FS as base clock of 9600 Hz.

Bit:	15 to 0
	MPCR
Initial value:	0
R/W:	R/W

21.2.4 AFEIF Status Register 1 and 2 (ASTR1, ASTR2)

ASTR is the control register for AFEIF, and composed of ASTR1 and ASTR2. ASTR1 is mainly used for transmit/receive FIFO interrupt control commands. ASTR2 is used for DAA interrupt control commands. See section 21.3.1, Interrupt Timing for more detail about interrupt handling.

(1) AFEIF Status Register 1 (ASTR1)

Bit:	15	14	13	12	11	10	9	8
	—	—	—	—	TFEM	RFFM	THEM	RHFM
Initial value:	0	0	0	0	1	1	1	1
R/W:	R/W	R	R	R	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	TFE	RFF	THE	RHF
Initial value:	0	0	0	0	1	0	1	0
R/W:	R	R	R	R	R	R	R	R

ASTR1 is composed by interrupt status flags (4 bits) relating transmit/receive FIFO and mask flags (4 bits) for transmit/receive FIFO interrupt signal. Status flag displays full/empty interrupt status of transmit/receive FIFO and half size interrupt status for FIFO. FIFO empty (TFE) and FIFO half size interrupt(THE) shows “1” as initial value, because transmit FIFO is empty after power on reset. These interrupt flags are to be cleared with the data write / read action to FIFO from CPU.

Each interrupt mask flag is able to prohibit interrupt generation of each interrupt that indicated in interrupt status flag. Every mask bits are automatically set when TE or RE bit are modified to 1. TFEM and THEM are 1 when TE = 0. RFFM and RHFM are “1” when RE = “0”. Each mask bit are reset as 1.

Bits 15 to 12 and 7 to 4—Reserved

Bit 11—Tx FIFO Empty Interrupt Mask (TFEM)

Bit 11: TFEM	Description	
0	TFE Interrupt enable	(Initial value)
1	TFE interrupt masked	

Bit 10—Rx FIFO Full Interrupt Mask (RFFM)

Bit 10: RFFM	Description	
0	RFF Interrupt enable	(Initial value)
1	RFF Interrupt masked	

Bit 9—Threshold of Tx FIFO Empty Interrupt Mask (THEM)

Bit 9: THEM	Description	
0	THE Interrupt enable	(Initial value)
1	THE Interrupt masked	

Bit 8—Threshold of Rx FIFO Full Interrupt Mask (RHFM)

Bit 8: RHFM	Description	
0	RHF Interrupt enable	(Initial value)
1	RHF Interrupt masked	

Bit 3—Tx FIFO Empty Interrupt (TFE)

Bit 3: TFE	Description	
0	Normal state	(Initial value)
1	TFIFO empty interrupt	

Set condition:

1. Reset
2. No effective data in area of FIFO
3. TE bit (ACTR1) is set to 0
   (TFEM is automatically masked in case 3.)

Clear condition:

1. Data are written into FIFO

Bit 2—Rx FIFO Full Interrupt (RFF)

Bit 2: RFF	Description	
0	No interrupt	(Initial value)
1	Rx FIFO full interrupt	

Set condition:

1. Specified size with FFSZ(ACTR1) of receive data is accumulated into FIFO.

Clear condition:

1. Reset
2. Number of data in FIFO becomes smaller than the size that is indicated with FFSZ (ACTR1).
3. RE bit (ACTR1) is set to 0.

Bit 1—TX FIFO Half Size Empty (THE)

Bit 1: THE	Description	
0	Normal state	(Initial value)
1	Tx FIFO Half Size Interrupt	

Set condition:

1. Reset
2. Number of valid data in FIFO becomes smaller than the half of the size that is indicated with FFSZ.
3. TE bit (ACTR1) is set to 0
   (THEM is automatically masked in case 3.)

Clear condition:

1. Number of valid data in FIFO becomes greater than the half of the size that is indicated by FFSZ.

Bit 0—RX FIFO Half Size Full (RHF)

Bit 0: RHF	Description	
0	Normal state	(Initial value)
1	Rx FIFO half size interrupt	

Set condition:

1. The half of specified size with FFSZ (ACTR1) of receive data is accumulated into FIFO.

Clear condition:

1. Reset
2. Number of data in FIFO becomes smaller than the half of the size that is indicated by FFSZ (ACTR1).
3. RE bit (ACTR1) is set to 0

(2) AFEIF Status Register 2 (ASTR2)

Bit:	15	14	13	12	11	10	9	8
	—	—	—	—	—	—	DPEM	RDETM
Initial value:	0	0	0	0	0	0	1	1
R/W:	R	R	R	R	R	R	R/W	R/W

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

ASTR2 is the register that is composed of interrupt status flag (2 bits) relating DAA control and mask flag (2 bits) of interrupt signals for DAA control. Status flags shows statuses of ringing detect interrupt, end of dial pulse output interrupt. Interrupt flags are cleared by 0 write after read action of this register. Each Interrupt signal are able to be masked by each interrupt masks.

Bit 9—Dial Pulse End Interrupt Mask (DPEM)

Bit 9: DPEM	Description	
0	Interrupt enable	(Initial value)
1	Interrupt mask	

Bit 8—Ringing Detect Mask (RDETM)

Bit 8: RDETM	Description	
0	Ringing interrupt enable	(Initial value)
1	Ringing interrupt mask	

Bit 1—Dial Pulse End (DPE)

Bit 1: DPE	Description	
0	Normal state	(Initial value)
1	Dial pulse end interrupt	

Set condition:

1. Output of all of dial pulse sequences completed or end command 0H detected
2. Illegal end (unspecified dial number and DPST set when RLYC bit (ACTR2) is low level)

Clear condition:

1. Reset
2. Interrupt status 1 is read and then 0 is written to this bit.

Bit 0—Ringing Detect (RDET)

Bit 0: RDET	Description	
0	Normal state	(Initial value)
1	Ringing waveform detect	

Set condition:

1. Ringing waveform is input to AFE_RDET pin (Latched at rising edge)

Clear condition:

1. Reset
2. Interrupt status 1 is read and then 0 is written to this bit.

21.2.5 Dial Pulse Number Queue (DPNQ)

This is the dial pulse number queue up to 4 digits which has 4-bits registers. This queue generates dial pulse according to the following table in the order of dial pulse number.

A dial-pulse-end interrupt is sent out after DN3 is output or if 0H or a value other than the corresponding data is detected.

Bit:	15 to 12	11 to 8	7 to 4	3 to 0
	DN0	DN1	DN2	DN3
Initial value:	0000	0000	0000	0000
R/W:	R/W	R/W	R/W	R/W

Table 21.3 Telephone Number and Data

TEL No.	Corresponding Data
0	AH
1	1H
2	2H
3	3H
4	4H
5	5H
6	6H
7	7H
8	8H
9	9H
Pause	FH
End	0H

21.2.6 Ringing Pulse Counter (RCNT)

The result of counting 1 cycle of ringing wave form with AFE_FS is shown here.

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

Bits 15 to 0—Ringing Counter Value (RCNTV): The result of counting 1 cycle of input ringing wave form with AFE_FS (output of AFE). See section 21.3.3, DAA Interface for more detail about the ringing detect sequence.

21.2.7 AFE Control Data Register (ACDR)

ACDR is the register to store the AFE control word. After 1 is written to HC bit (ACTR1), data is transferred to AFE at the timing of 3rd FS.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ACDR															
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

21.2.8 AFE Status Data Register (ASDR)

ASDR is the register to store the AFE status word. After 1 is written to HC bit (ACTR2), data is transferred to ASDR from AFE at the timing of 3rd FS.

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

21.2.9 Transmit Data FIFO Port (TDFP)

TDFP is the write only port for transmit FIFO. Transmit FIFO has 128 stages (maximum), and can generate interrupt of the data empty as well as of the threshold size specified by FFSZ (ACTR1). Directly after the reset and when TE (ACTR1) bit is 0, the pointer of FIFO is set to the first address and data becomes empty. The interrupt will occur when the TE bit (ACTR1) is written to 1 at that state. In normal case, TE bit should be changed after writing data into transmit FIFO.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TDFP															
Initial value:	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
R/W:	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

21.2.10 Receive Data FIFO Port (RDFP)

RDFP is the read only register for receive FIFO. Receive FIFO has 128 stages (maximum), and can generate interrupt of the data full as well as of the threshold size specified by FFSZ (ACTR1). Directly after the reset and when RE bit (ACTR1) is 0, the pointer of FIFO is fixed at the first address and data from RDFP becomes undetermined.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RDFP															
Initial value:	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

21.3 Operation

21.3.1 Interrupt Timing

AFE interface module generates 3 types of interrupt: FIFO data transfer, ringing detect, and dial pulse transmit end. The timing of each interruption is described below.

(1) FIFO Interrupt Timing

Figure 21.2 shows interrupt timing of data transfer FIFO. Transmit FIFO generates the TFE and THE interrupts after the last data is transfer red shift register. Receive FIFO generates the RFF and RHF interrupt after the last data or specified word is transferred from shift register to FIFO.

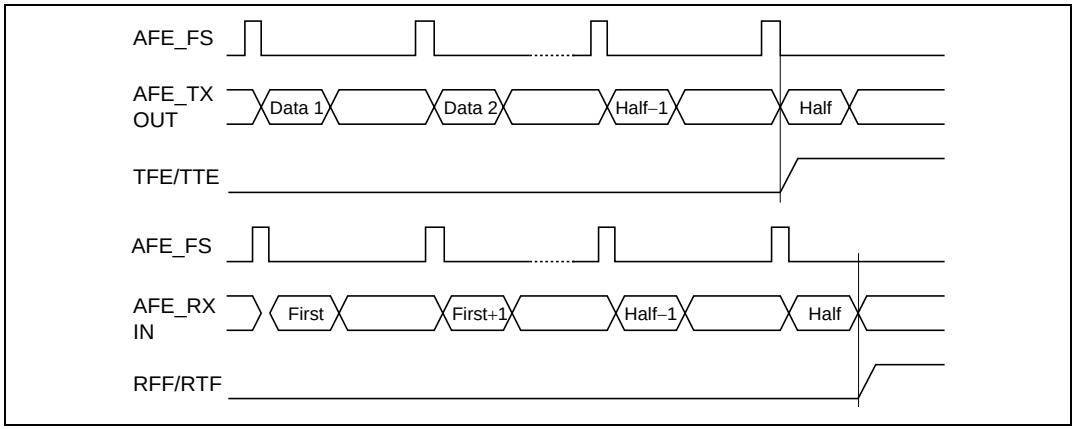


Figure 21.2 FIFO Interrupt Timing

(2) Ringing Interrupt Timing

As the figure 21.3 shows, the ringing signal from the line is transformed to rectangular wave and then input to AFEIF. The interrupt is generated at the rising edge of input wave in AFEIF module.

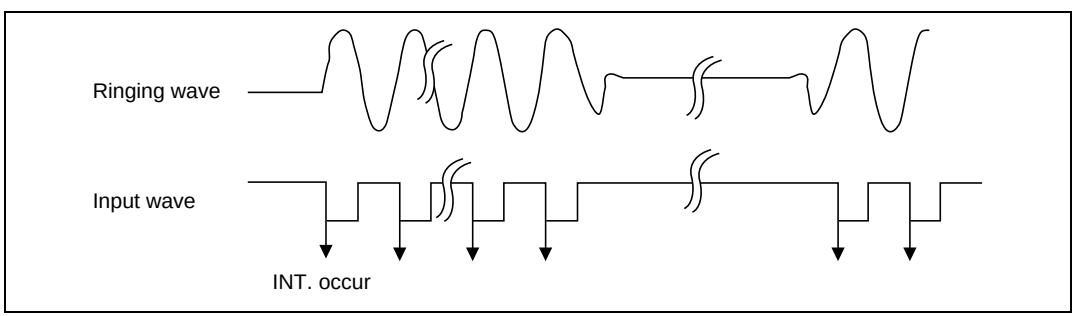


Figure 21.3 Ringing Interrupt Occurrence Timing

(3) Dial Pulse Interrupt Timing

Dial pulse interrupt is generated in the dial pulse transmit sequence when AFEIF reads 0H (end) data from DPNQ register or all of 4 digits are output. Refer to section 21.3.3, DAA Interface about dial pulse sequence.

(4) Interrupt Generator Circuit

Interrupt is generated as is shown in figure 21.4. That is, AFEIFI signal is generated by performing OR operation on the four signals from ASTR1 in FIFO interrupt control and the two signals from ASTR2 in DAA interrupt control, and then sent out to INTC as one interrupt signal.

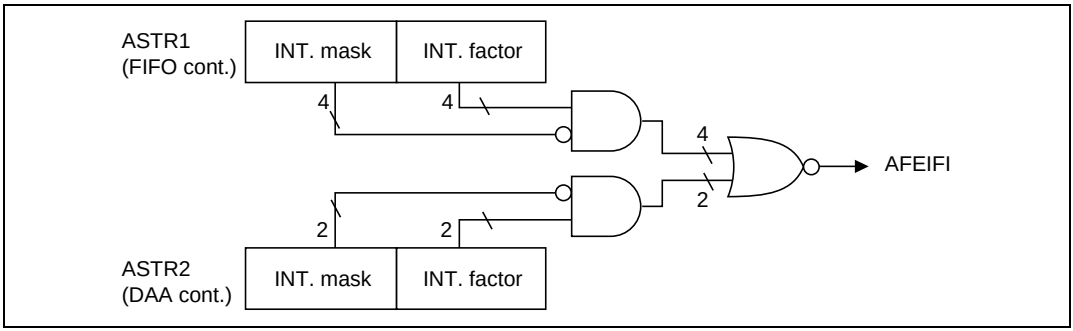


Figure 21.4 Interrupt Generator

21.3.2 AFE Interface

(1) Serial Data Transfer Specification

The specification for serial data transfer is base on that of STLC7550, which is an AFE manufactured by ST microelectronics. STLC7550 has a self-oscillation mode, and flame synchronous signal AFE_FS used for serial transfer and serial bit clock AFE_SCLK are supplied by AFE. Figure 21.5 shows the serial transfer interface. After outputting the valid data, AFE_TXOUT holds the value of LSB.

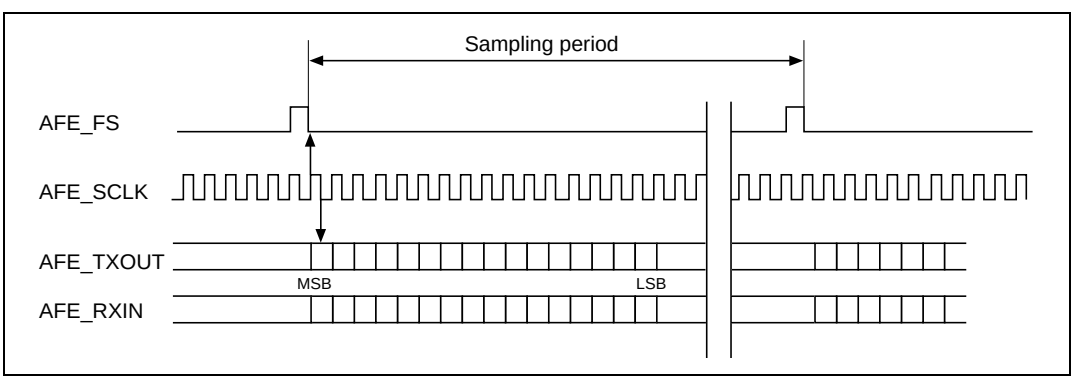
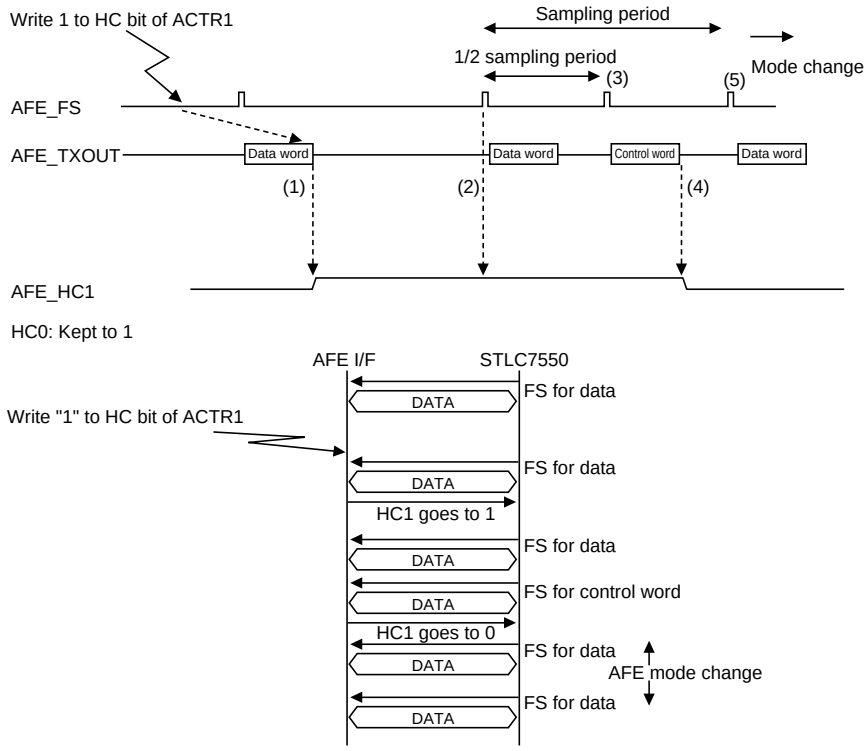


Figure 21.5 AFE Serial Interface

(2) HC Control Sequence

AFEIF module supports hardware control STLC7550 that is an AFE manufactured by ST microelectronics. Figure 21.6 shows the AFE control sequence.



1. If the CPU write "1" to the HC bit of ACTR1, the AEFIF drives AFE_HC1 to "H" right after transmit next data.
2. AFE fetches the HC1's status of "H" at rising edgge of next AFE_FS.
3. AFE output the FS at the next 1/2 sampling period and then AFEIF transfer the AFE control word.
4. AFEIF keeps AFE_HC1 to "H" for 2nd FS and return to "L" after transmit the control word.
5. AFE fetches the AFE_HC1's status of "L" and changes the mode of itself.

Figure 21.6 AFE Control Sequence

21.3.3 DAA Interface

Figure 21.7 shows the blocks diagram of DAA circuit. Ringing detect and dial pulse sending sequence are described below.

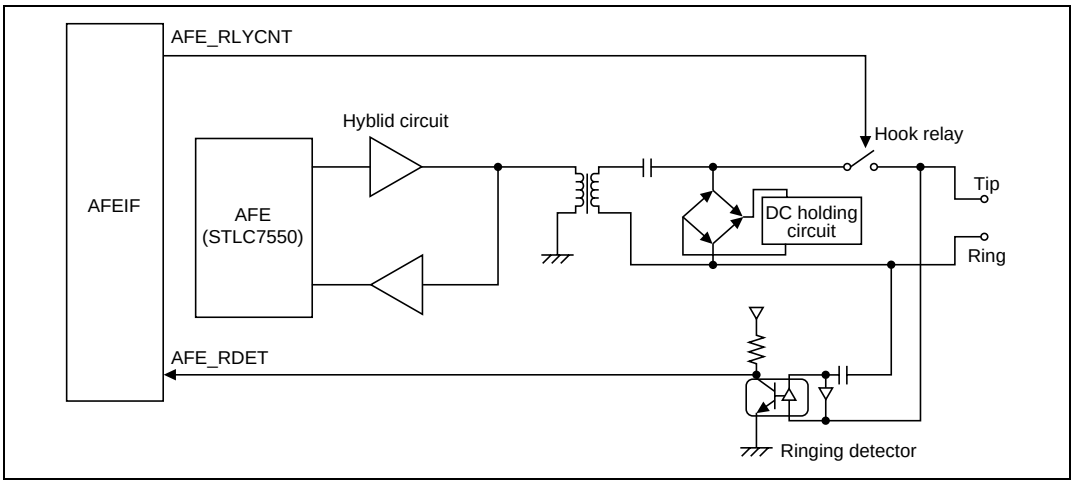


Figure 21.7 DAA Block Diagram

(1) Ringing Detect Sequence

After the first ringing interrupt occurs, counting starts with writing 1 into RCEN bit of CTR2. AFE must be operating before counting, because periodic counter counts AFE_FS from falling edge to next falling edge.

The value of RCNTV register is effective only after 2nd interrupt generation, because the value of RCNTV register is transferred from counter with a trigger of ending of 1st period cycle.

RCNTV will be 258H (600 in decimal) if ringing cycle is 16Hz and counted by 9600Hz which is default value of AFE_FS. Figure 21.8 shows detecting sequence of ringing.

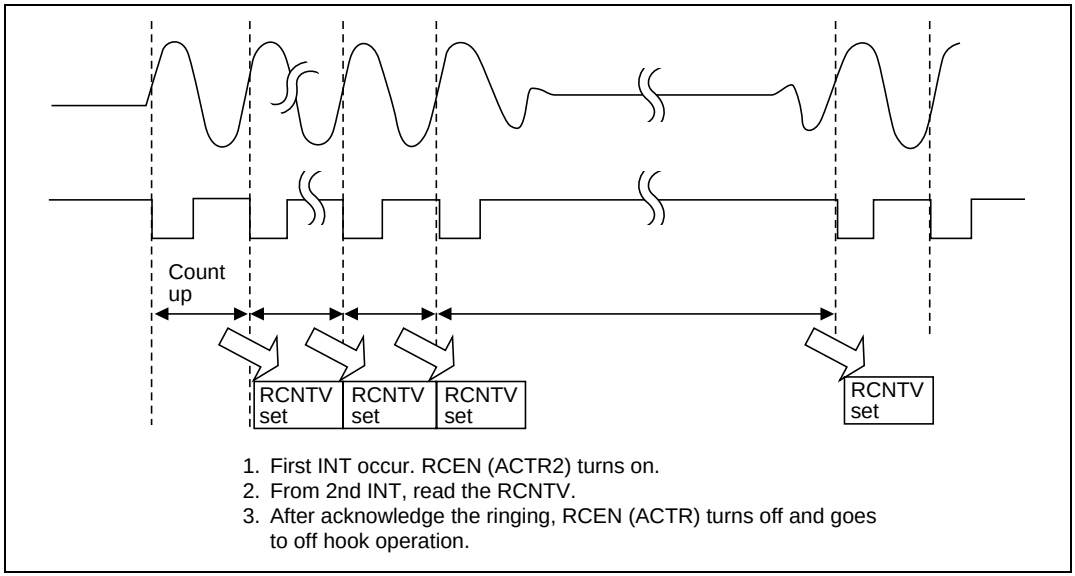


Figure 21.8 Ringing Detect Sequence

(2) Dial Pulse Sending Sequence

A dial pulse is generated according to the conditions that are specified in ACTR2, and is sent out to AFE_RLYCNT.

As the basic clock for generating the dial pulse is AFE_FS that is input from AFE, it is necessary to make AFE in operating state. An example of control sequence for dial pulse sending is shown below.

Note that this sequence cannot be operated when RLYC bit (ACTR2) is low.

[Conditions]
Make ratio: 33%
Pulse interval: 20 PPS
Minimum pause: 600 ms
Dial number: 0,1234567 (“,” means pause)

[Control sequence]
1. Set PPS (ACTR2) → “1”, MKR → “9EH1”, MNRPCNT → “1680H”
2. Set DPNQ → “AF12H”.
3. Set RLYC → “H”. (Off Hook)

4. Detect dial tone or wait specific period. (Controlled by software)
5. Write “1” to DPST (ACTR2). (Start sending dial pulse)
6. After 4 digit of dial pulses are sent, interrupt is generated. (DPST is reset to “0”)
7. Set DPNQ1 → “3456H”.
8. Write “1” to DPST (ACTR2).
9. After 4 digit of dial pulses are sent, interrupt is generated. (DPST is reset to “0”)
10. Set DPNQ2 → “70XXH”.
11. Write “1” to DPST (ACTR2).
12. After 1 digit of dial pulse is sent, interrupt is generated. (DPST is reset to “0”, finish sending)

21.3.4 Wake up Ringing Interrupt

System wake up function by the ringing signal from telephone line is realized by inputting AFE_RDET signal, that is an input signal for ringing, to PINT pin.

Section 22 USB Pin Multiplex Controller

22.1 Feature

The USB multiplex controller controls the data path to USB transceiver from USB host controller port 1 or USB function controller.

Both USB host port 1 and USB function controller are connected to USB transceiver 1 via multiplexer that is controlled by EXPFC register. The USB host controller port 2 and USB transceiver 2 are connected one-to-one. USB transceiver 1 can be connected to USB host controller or USB function controller, while USB transceiver 2 can only be connected to the USB host controller. Because these ports and transceivers are controlled individually, USB transceiver 2 can be connected to either the USB host controller or the USB function controller regardless its status. The signals to USB transceiver are used as external pins USB1d _**** which are multiplexed with pins 113 to 122.

22.1.1 Block Diagram

Figure 22.1 shows the connections between the on-chip USB host controller of the SH7727, the USB function controller, and the on-chip 2-port USB transceiver.

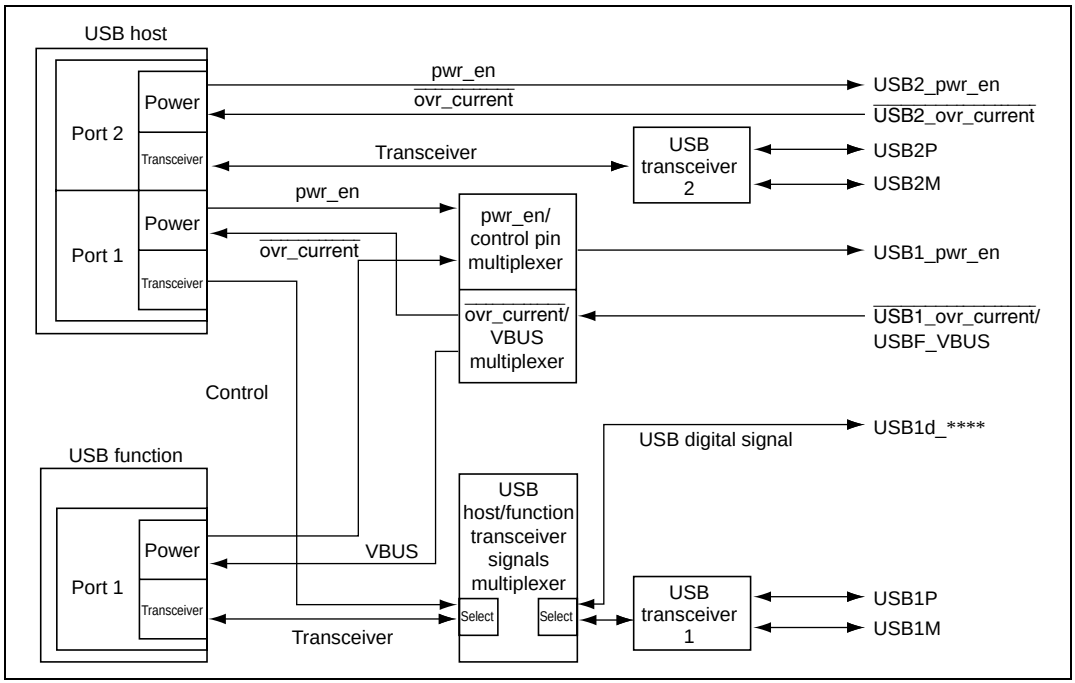


Figure 22.1 Block Diagram of USB PIN Multiplexer

22.1.2 Pin Configuration

USB pin multiplexer controller has pins that are shown in tables 22.1, 22.2, and 22.3

Table 22.1 Pin Configuration (Digital Transceiver Signal)

Name	Symbol	I/O	Description
RCV pin	USB1d_RCV	Input	Input pin for receive data from differential receiver
DPLS pin	USB1d_DPLS	Input	Input pin for D+ signal from receiver
DMNS pin	USB1d_DMNS	Input	Input pin for D– signal from receiver
TXDPLS pin	USB1d_TXDPLS	Output	D+ transmit output pin
TXENL pin	USB1d_TXENL	Output	Driver output enable pin
SUSPEND pin	USB1d_SUSPEND	Output	Transceiver suspend state output pin
SPEED pin	USB1d_SPEED	Output	Transceiver speed control pin
TXSE0 pin	USB1d_TXSE0	Output	SE0 state output pin

Note: The pins shown in table 22.1 are used for connecting an external USB transceiver, and cannot be used when the on-chip USB transceiver is connected.

Table 22.2 Pin Configuration (Analog Transceiver Signal)

Name	Symbol	I/O	Description
1P pin	USB1P	I/O	D+ port1 transceiver pin
1M pin	USB1M	I/O	D– port1 transceiver pin
2P pin	USB2P	I/O	D+ port2 transceiver pin
2M pin	USB2M	I/O	D– port2 transceiver pin

Note: The pins shown in table 22.2 can be used as two ports USB host controller pins, or one port USB host controller pins and one port USB function controller pins. make these pins open, when they are not used.

Table 22.3 Pin Configuration (Power Control signal)

Name	Symbol	I/O	Description
Power enable pin 1	USB1_pwr_en	Output	USB port 1 power enable control
Power enable pin 2	USB2_pwr_en	Output	USB port 2 power enable control
One current pin 1/ VBUS pin	$\overline{\text{USB1_ovr_current}}$/ USBF_VBUS	Input	USB port 1 over-current detect/ USB cable connection monitor pin
Over current pin 2	$\overline{\text{USB2_ovr_current}}$	Input	USB port2 Over-current detect

Note: The pins shown in table 22.3 can be used for power control of USB. Pins for port 1 have the functions that are multiplexed functions of USB controller and USB function controller.

22.1.3 Register Configuration

Table 22.4 shows the registers for USB pin multiplexer controller.

Table 22.4 Register Configuration

Name	Abbreviation	R/W	Initial Value	Address	Access Size
Extra pin function controller	EXPFC	R/W	H'0000	H'A4000234	16

22.2 Register Description

22.2.1 Extra Pin Function Controller (EXPFC)

Bit:	15	14	13	12	11	10	9	8
	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	USB_ TRANS	USB_SEL
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R/W*	R/W*	R/W*	R/W*	W	W

Note: * 0 must be set in reading or writing.

Bits 15 to 2—Reserved: Write 0 to these bits. When 1 is written to this bits, the operation is unpredictable.

Bit 1—USB Port 1 Transceiver (USB_TRANS)

Bit 1	Function	
0	USB transceiver is enabled	(Initial value)
1	USB digital signals output is enabled	

Bit 0—USB Port 1 Signal Source Selection (USB_SEL)

Bit 0	Function (signal source selection)	
0	USB HOST is used	(Initial value)
1	USB Function is used	

Note: USB port 2 is for dedicated use by the USB host controller.

22.3 Examples of External Circuit

22.3.1 Example of the Connection between USB Function Controller and Transceiver

Figures 22.2 to 22.5 show example connections of USB function controller and transceiver. Figures 22.2 and 22.3 show connections when using the built-in USB transceiver. Figures 22.4 and 22.5 show connections when not using the built-in USB transceiver. When using the USB function controller, the signals must be input to the cable connection monitor pin UJBF_VBUS. The USBF_VBUS pin is multiplexed with the $\overline{\text{USB1_ovr_current}}$ pin, and writing 1 to bit 0 of the EXPFC register selects the USBF_VBUS pin functions. According to the status of the USBF_VBUS pin, the USB function controller recognizes whether the cable is connected/disconnected. Also, pin D+ must be pulled up in order to notify the USB host/hub that the connection is established. The sample circuits in figures 22.2 to 22.5 use the USB1_pwr_en pin for pull-up control.

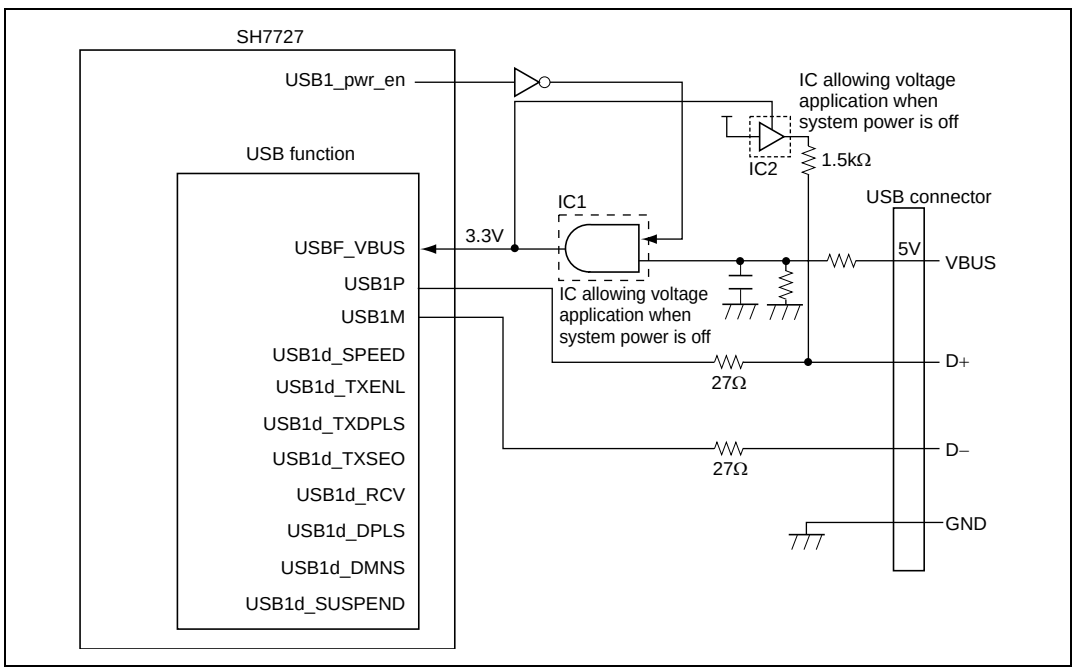


Figure 22.2 Example 1 of Transceiver Connection for USB function Controller (On-chip transceiver is used)

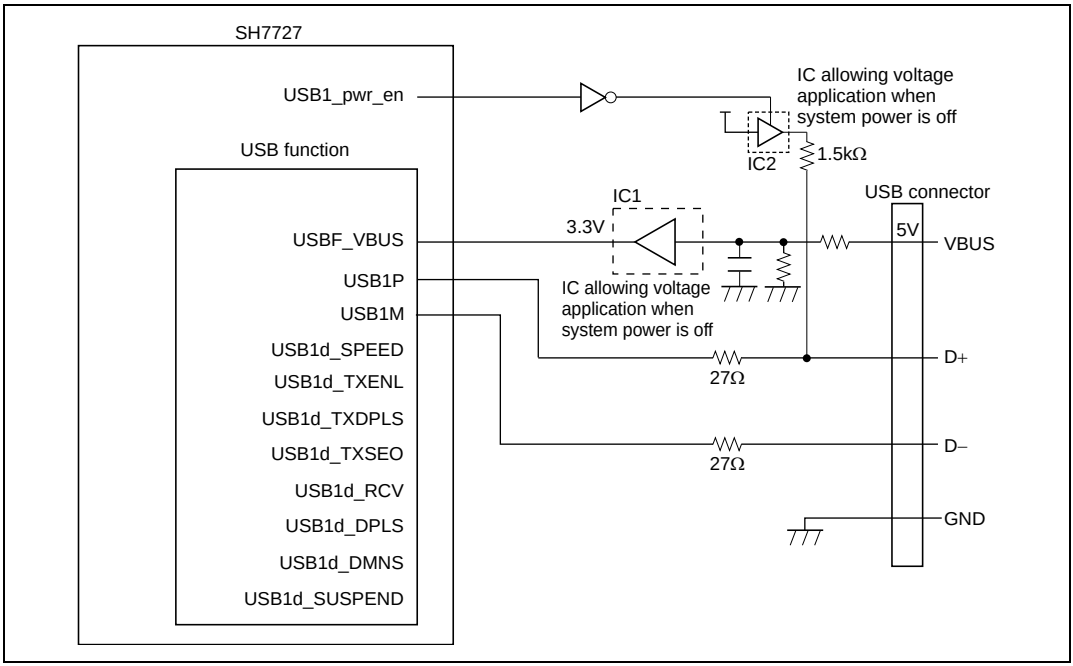


Figure 22.3 Example 2 of Transceiver Connection for USB function Controller (On-chip transceiver is used)

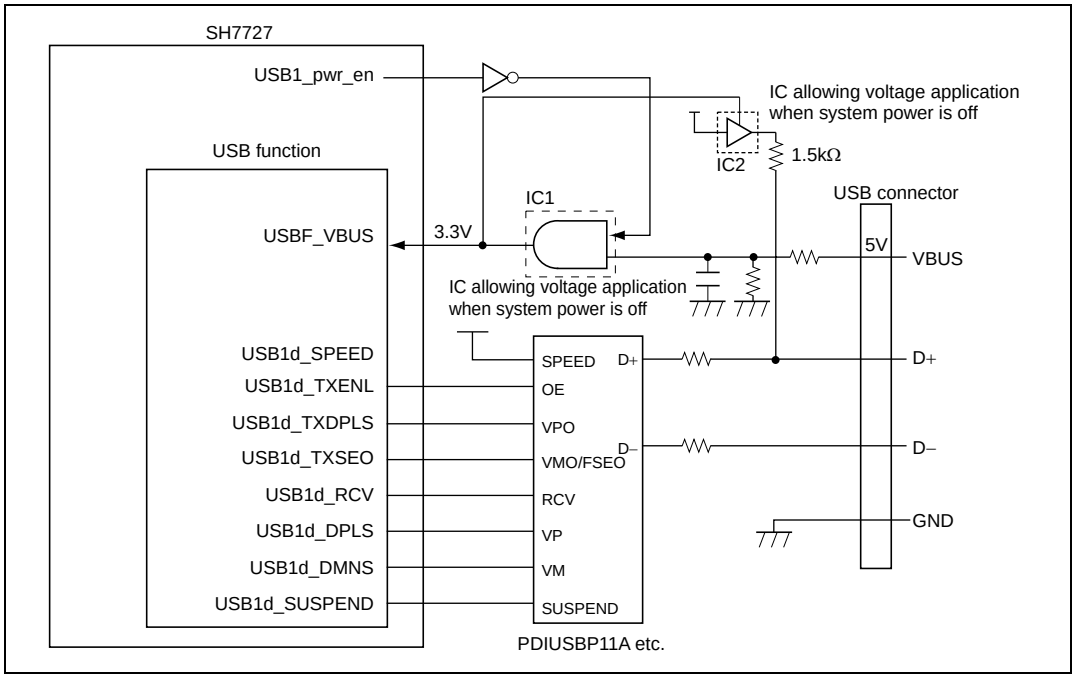


Figure 22.4 Example 3 of Transceiver Connection for USB function Controller (On-chip transceiver is not used)

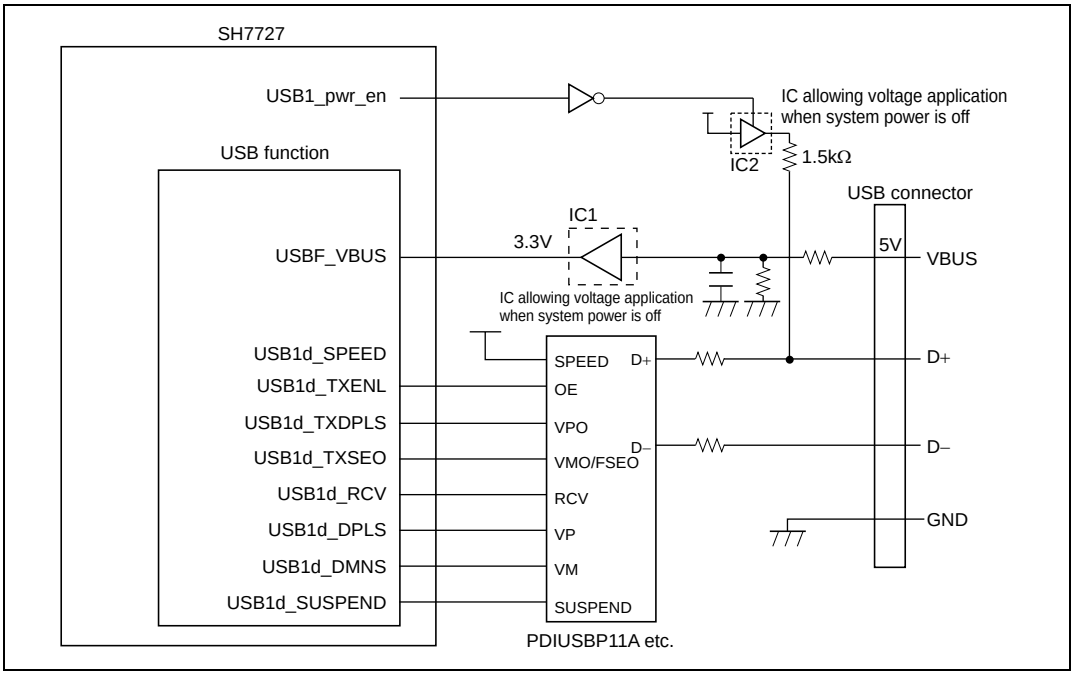


Figure 22.5 Example 4 of Transceiver Connection for USB function Controller (On-chip transceiver is not used)

D+ Pull-up Control

Control D+ pull-up by using USB1_pwr_en pin in the system when the connection—notification (D+ pull-up) to USB host or hub is wished to be inhibited (i.e., during high-priority processing or initialization processing).

The D+ pull-up control signal and USBF_VBUS pin input signal should be controlled by using the USB1_pwr_en pin and the USB cable VBUS (AND circuit) as is shown in examples of figures 22.2 and 22.4

D+ pull-up is inhibited when the USB1_pwr_en pin is high in examples of figures 22.2 to 22.5. (The initial setting of the USB1_pwr_en pin is High.)

Pull-up D+ after confirming that USBF_VBUS pin became high, when the pull-up control is performed directly by USB1_pwr_en pin as is shown in examples of figures 22.3 and 22.5.

D+ pull-up is inhibited when the USB1_pwr_en pin is low in examples of figures 22.3 and 22.5. Use an IC such that allows voltage application when system power is off (for example, HD74LV1G126A) for the pull-up control IC (IC2 in figures 22.2 to 22.5).

(The UDC core in the SH7727 holds the powered state when USBF_VBUS pin is low, regardless of the D+/D− state.)

Detection of USB Cable Connection/Disconnection

As USB function controller in the SH7727 manages the state by hardware, USB_VBUS signal is necessary to recognize connection or disconnection of the USB cable. The power supply signal (VBUS) in the USB cable is used for USBF_VBUS. However, if the cable is connected to the USB host or hub when the power of USB function controller (SH7727—installed system) is off, a voltage of 5 V will be applied from the USB host or hub.

Therefore, use an IC such that allows voltage application when system power is off (for example, HD74LV1G08A) for the IC1 in figures 22.2 to 22.5.

22.3.2 Example of the Connection between USB Host Controller and Transceiver

Figures 22.6 and 22.7 show example connections of the USB host controller and transceiver. Figure 22.6 shows an example connection using the built-in transceiver 1. By using the $\overline{\text{USB2_ovr_current}}$, USB2_pwr_en, USB2P, and USB2M pins in an external circuit similar to that in figure 22.6, you can also use built-in USB transceiver 2. Figure 22.7 shows an example connection when not using the built-in USB transceiver. When using the USB host controller, a separate LSI must be used for USB power bus control (equivalent to the USB power control LSIs in figures 22.6 and 22.7). Make sure the LSI has the power supply capacity to satisfy the USB standard, and select one that has an overcurrent protection function. Configure the system so that the input to the $\overline{\text{USB1_ovr_current}}$ pin is Low on detection of an overcurrent.

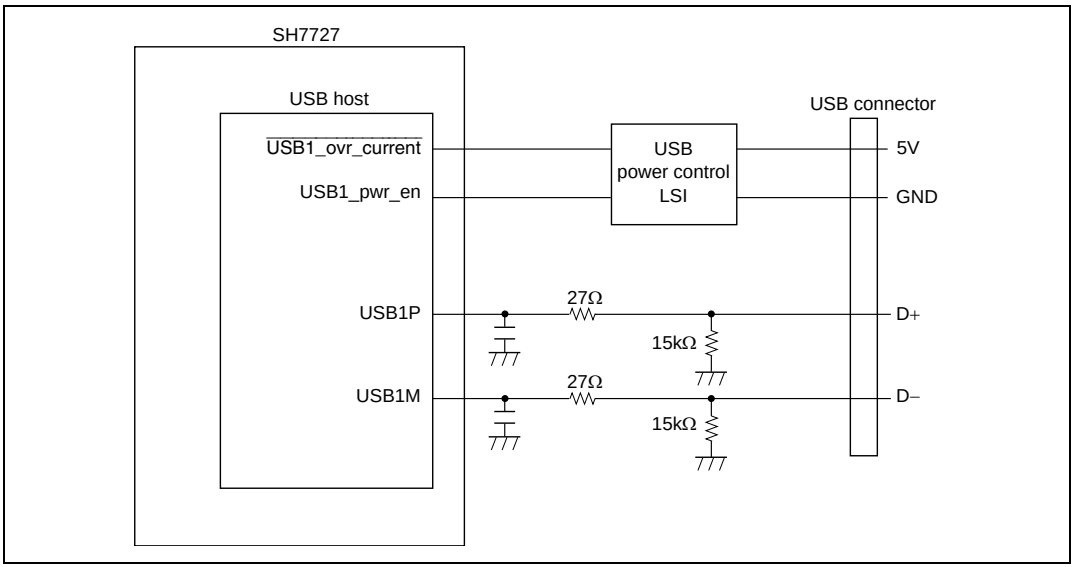


Figure 22.6 Example 1 of Transceiver Connection for USB Host Controller (On-chip transceiver is used)

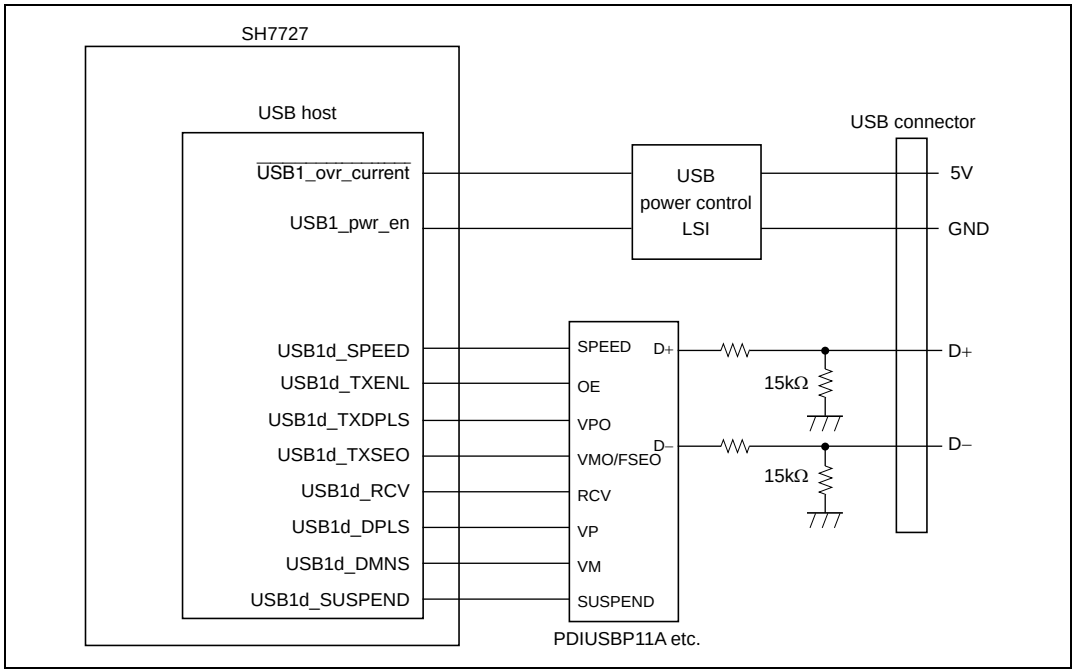


Figure 22.7 Example 2 of Transceiver Connection for USB Host Controller (On-chip transceiver is not used)

22.3.3 Usage Notes

About the USB Transceiver

USB transceiver is included in the SH7727. it is also possible to connect an external transceiver according to the setting in EXPFC register (see figures 22.4, 22.5, and 22.7). In this case, ask the manufacturer of the transceiver about the recommended circuit that is used between the USB transceiver and USB connectors.

About the Examples of External Circuit

These examples of transceiver connection in this chapter are for reference only, therefore proper operation is not guaranteed with these circuit examples. If system countermeasures are required for external surges and ESD noise, use a protective diode, etc.

Section 23 USB Function Controller

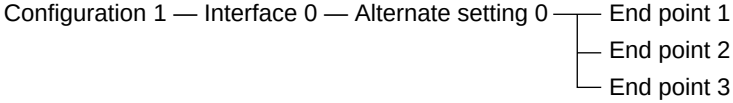
23.1 Features

- Incorporates UDC (USB device controller) conforming to USB1.1

  Automatic processing of USB protocol

  Automatic processing of USB standard commands for endpoint 0 (some commands and class/vendor commands require decoding and processing by firmware)
- Transfer speed: Full-speed
- Endpoint configuration

Endpoint Name	Abbreviation	Transfer Type	Maximum Packet Size	FIFO Buffer Capacity	DMA Transfer
Endpoint 0	EP0s	Setup	8	8	—
	EP0i	Control-in	8	8	—
	EP0o	Control-out	8	8	—
Endpoint 1	EP1	Bulk-out	64	128	Possible
Endpoint 2	EP2	Bulk-in	64	128	Possible
Endpoint 3	EP3	Interrupt	8	8	—



- Interrupt requests: generates various interrupt signals necessary for USB transmission/reception
- Clock: Selection by means of EXCPG
  For details, see section 11, Extend Clock Pulse Generator for USB (EXCPG).
- Power-down mode

  Power consumption can be reduced by stopping internal clock when UDC cable is disconnected

  Automatic transition to/recovery from suspend state
- Can be connected to a Philips PDIUSBP11 Series transceiver or compatible product (when using a compatible product, carry out evaluation and investigation with the manufacturer supplying the transceiver beforehand)
- This USB function controller is a self-power device. It cannot operate by power supplied from the USB cable.

23.2 Block Diagram

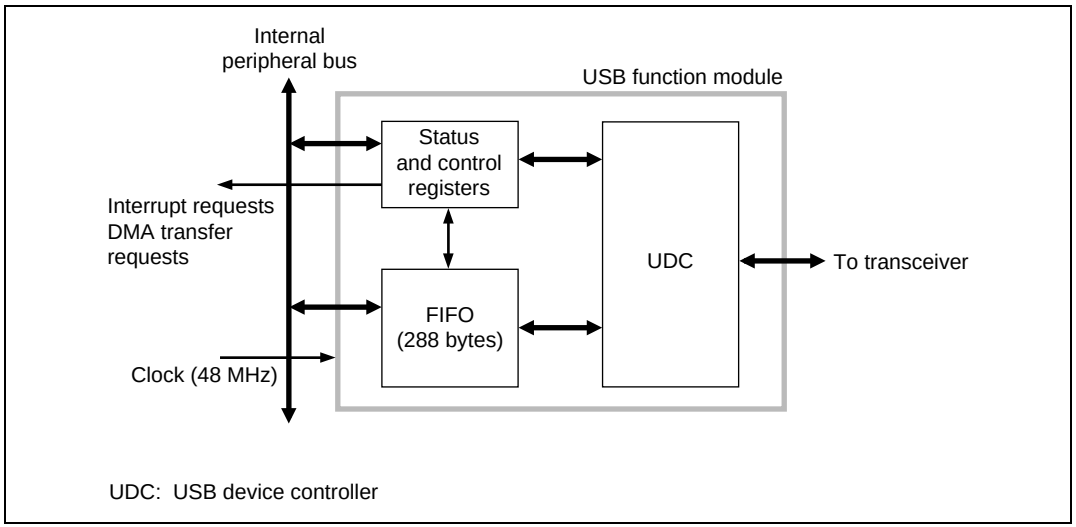


Figure 23.1 Block Diagram of UBC

23.3 Pin Configuration

Table 23.1 Pin Configuration and Functions

Pin Name	I/O	Function
USBF_VBUS	Input	USB cable connection monitor pin
USB1_pwr_en	Output	USB1 power control pin

Can be connected to a Philips PDIUSBP11 series transceiver or compatible product (when using a compatible product, carry out evaluation and investigation with the manufacturer supplying the transceiver beforehand). See section 22, USB Pin Multiplex Controller, for connection to the USB transceiver.

23.4 Register Configuration

Table 23.2 USB Function Module Registers

Name	Abbreviation	R/W	Initial Value	Address	Access Size
USBEP0i data register	USBEPDR0I	W	—	H'04000242 (H'A4000242)*	8
USBEP0o data register	USBEPDR0O	R	—	H'04000243 (H'A4000243)*	8
USBEP0s data register	USBEPDR0S	R	—	H'04000247 (H'A400247)*	8
USBEP1 data register	USBEPDR1	R	—	H'0400024E (H'A400024E)*	8
USBEP2 data register	USBEPDR2	W	—	H'04000249 (H'A4000249)*	8
USBEP3 data register	USBEPDR3	W	—	H'04000252 (H'A4000252)*	8
Interrupt flag register 0	USBIFR0	R/W	H'10	H'04000240 (H'A4000240)*	8
Interrupt flag register 1	USBIFR1	R/W	H'00	H'04000241 (H'A4000241)*	8
Trigger register	USBTRG	W	—	H'04000244 (H'A4000244)*	8
FIFO clear register	USBFCLR	W	—	H'04000245 (H'A4000245)*	8
USBEP0o receive data size register	USBEPSZ0O	R	H'00	H'04000246 (H'A4000246)*	8
Data status register	USBDASTS	R	H'00	H'04000248 (H'A4000248)*	8
Endpoint stall register	USBEPSTL	R/W	H'00	H'0400024B (H'A400024B)*	8
Interrupt enable register 0	USBIER0	R/W	H'00	H'0400024C (H'A400024C)*	8
Interrupt enable register 1	USBIER1	R/W	H'00	H'0400024D (H'A400024D)*	8
USBEP1 receive data size register	USBEPSZ1	R	H'00	H'0400024F (H'A400024F)*	8
USBDMA setting register	USBDMA	R/W	H'00	H'04000251 (H'A4000251)*	8

Name	Abbreviation	R/W	Initial Value	Address	Access Size
Interrupt select register 0	USBISR0	R/W	H'00	H'0400024A (H'A400024A)*	8
Interrupt select register 1	USBISR1	R/W	H'07	H'04000250 (H'A4000250)*	8

Note: * If the MMU does not convert addresses, use addresses in parentheses.

23.5 Register Descriptions

23.5.1 USBEP0i Data Register (USBEPDR0I)

USBEPDR0I is an 8-byte FIFO buffer for endpoint 0, holding one packet of transmit data for control-in. Transmit data is fixed by writing one packet of data and setting bit 0 in the USB trigger register. When an ACK handshake is returned from the host after the data has been transmitted, EP0i TS in USB interrupt flag register 0 is set. This FIFO buffer can be initialized by means of EP0i CLR in the USBFIFO clear register.

23.5.2 USBEP0o Data Register (USBEPDR0O)

USBEPDR0O is an 8-byte receive FIFO buffer for endpoint 0. USBEPDR0O holds endpoint 0 receive data other than setup commands. When data is received normally, EP0o TS in USB interrupt flag register 0 is set, and the number of receive bytes is indicated in the EP0o receive data size register. After the data has been read, setting EP0o RDFN in the USB trigger register enables the next packet to be received. This FIFO buffer can be initialized by means of EP0o CLR in the USBFIFO clear register.

23.5.3 USBEP0s Data Register (USBEPDR0S)

USBEPDR0S is an 8-byte FIFO buffer specifically for endpoint 0 setup command reception. USBEPDR0S receives only setup commands requiring processing on the application side. When command data is received normally, SETUP TS in USB interrupt flag register 0 is set. As a setup command must be received without fail, if data is left in this buffer, it will be overwritten with new data. If reception of the next command is started while the current command is being read, command reception has priority, the read by the application is forcibly terminated, and the read data is invalid.

23.5.4 USBEP1 Data Register (USBEPDR1)

USBEPDR1 is a 128-byte receive FIFO buffer for endpoint 1. USBEPDR1 has a dual-FIFO configuration, and has a capacity of twice the maximum packet size. When one packet of data is received normally from the host, EP1 FULL in USB interrupt flag register 0 is set. The number of receive bytes is indicated in the USBEP1 receive data size register. After the data has been read, the buffer that was read is enabled to receive again by writing 1 to EP1 RDFN in the USB trigger register. The receive data in this FIFO buffer can be transferred by DMA (see section 23.5.19, USBDMA Setting Register (USBDMAR)). This FIFO buffer can be initialized by means of EP1 CLR in the USBFIFO clear register.

23.5.5 USBEP2 Data Register (USBEPDR2)

USBEPDR2 is a 128-byte transmit FIFO buffer for endpoint 2. USBEPDR2 has a dual-buffer configuration, and has a capacity of twice the maximum packet size. When transmit data is written to this FIFO buffer and EP2 PKTE in the USB trigger register is set, one packet of transmit data is fixed, and the dual-FIFO buffer is switched over. Transmit data for this FIFO buffer can be transferred by DMA (see section 23.5.19, USBDMA Setting Register (USBDMAR)). This FIFO buffer can be initialized by means of EP2 CLR in the USBFIFO clear register.

23.5.6 USBEP3 Data Register (USBEPDR3)

USBEPDR3 is an 8-byte transmit FIFO buffer for endpoint 3, holding one packet of transmit data in endpoint 3 interrupt transfer. Transmit data is fixed by writing one packet of data and setting EP3 PKTE in the USB trigger register. When an ACK handshake is received from the host after one packet of data has been transmitted normally, EP3 TS in the USB interrupt flag register 1 is set. This FIFO buffer can be initialized by means of EP3 CLR in the USB FIFO clear register.

23.5.7 USB Interrupt Flag Register 0 (USBIFR0)

Together with USB interrupt flag register 1, USBIFR0 indicates interrupt status information required by the application. When an interrupt source occurs, the corresponding bit is set to 1 and an interrupt request is sent to the CPU according to the combination with USB interrupt enable register 0. Clearing is performed by writing 0 to the bit to be cleared, and 1 to the other bits. However, EP1 FULL and EP2 EMPTY are status bits, and cannot be cleared.

Bit:	7	6	5	4	3	2	1	0
	BRST	EP1 FULL	EP2 TR	EP2 EMPTY	SETUP TS	EP0o TS	EP0i TR	EP0i TS
Initial value:	0	0	0	1	0	0	0	0
R/W:	R/W	R	R/W	R	R/W	R/W	R/W	R/W

Bit 7—Bus Reset (BRST): Set to 1 when the bus reset signal is detected on the USB bus.

Bit 6—EP1 FIFO Full (EP1 FULL): This bit is set when endpoint 1 receives one packet of data normally from the host, and holds a value of 1 as long as there is valid data in the FIFO buffer. EP1 FULL is a status bit, and cannot be cleared.

Bit 5—EP2 Transfer Request (EP2 TR): This bit is set if there is no valid transmit data in the FIFO buffer when an IN token for endpoint 2 is received from the host. A NACK handshake is returned to the host until data is written to the FIFO buffer and packet transmission is enabled.

Bit 4—EP2 FIFO Empty (EP2 EMPTY): This bit is set when at least one of the dual endpoint 2 transmit FIFO buffers is ready for transmit data to be written. EP2 EMPTY is a status bit, and cannot be cleared.

Bit 3—Setup Command Receive Complete (SETUP TS): This bit is set to 1 when endpoint 0 receives normally a setup command requiring decoding on the application side, and returns an ACK handshake to the host.

Bit 2—EP0o Receive Complete (EP0o TS): This bit is set to 1 when endpoint 0 receives data from the host normally, stores the data in the FIFO buffer, and returns an ACK handshake to the host.

Bit 1—EP0i Transfer Request (EP0i TR): This bit is set if there is no valid transmit data in the FIFO buffer when an IN token for endpoint 0 is received from the host. A NACK handshake is returned to the host until data is written to the FIFO buffer and packet transmission is enabled.

Bit 0—EP0i Transmit Complete (EP0i TS): This bit is set when data is transmitted to the host from endpoint 0 and an ACK handshake is returned.

23.5.8 USB Interrupt Flag Register 1 (USBIFR1)

Together with USB interrupt flag register 0, USBIFR1 indicates interrupt status information required by the application. When an interrupt source occurs, the corresponding bit is set to 1 and an interrupt request is sent to the CPU according to the combination with USB interrupt enable register 1. Clearing is performed by writing 0 to the bit to be cleared, and 1 to the other bits.

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	VBUSMN	EP3 TR	EP3 TS	VBUSF
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W

Bits 7 to 4—Reserved: These bits are always read as 0. The write value should always be 0.

Bit 3—USB Connect Status (VBUSMN): This bit is a status bit for monitoring the state of the USBF_VBUS pin. It reflects the state of the USBF_VBUS pin.

Bit 2—EP3 Transfer Request (EP3 TR): This bit is set if there is no valid transmit data in the FIFO buffer when an IN token for endpoint 3 is received from the host. A NACK handshake is returned to the host until data is written to the FIFO buffer and packet transmission is enabled.

Bit 1—EP3 Transmit Complete (EP3 TS): This bit is set when data is transmitted to the host from endpoint 3 and an ACK handshake is returned.

Bit 0—USB Bus Connect (VBUSF): This bit is set to 1 when connecting to or disconnecting from the USB bus. The USBF_VBUS pin is used to detect connection/disconnection.

The USBF_VBUS pin must be connected, as it is needed inside the module.

23.5.9 USB Trigger Register (USBTRG)

USBTRG generates one-shot triggers to control the transmit/receive sequence for each endpoint.

Bit:	7	6	5	4	3	2	1	0
	—	EP3 PKTE	EP1 RDFN	EP2 PKTE	—	EP0s RDFN	EP0o RDFN	EP0i PKTE
R/W:	W	W	W	W	W	W	W	W

Bit 7—Reserved

Bit 6—EP3 Packet Enable (EP3 PKTE): After one packet of data has been written to the endpoint 3 transmit FIFO buffer, the transmit data is fixed by writing 1 to this bit.

Bit 5—EP1 Read Complete (EP1 RDFN): Write 1 to this bit after one packet of data has been read from the endpoint 1 FIFO buffer. The endpoint 1 receive FIFO buffer has a dual-FIFO configuration. Writing 1 to this bit initializes the FIFO that was read, enabling the next packet to be received.

Bit 4—Endpoint 2 Packet Enable (EP2 PKTE): After one packet of data has been written to the endpoint 2 FIFO buffer, the transmit data is fixed by writing 1 to this bit.

Bit 3—Reserved

Bit 2—EP0s Read Complete (EP0s RDFN): Write 1 to this bit after EP0s command FIFO data has been read. Writing 1 to this bit enables transmission/reception of data in the following data stage. A NACK handshake is returned in response to transmit/receive requests from the host in the data stage until 1 is written to this bit.

Bit 1—EP0o Read Complete (EP0o RDFN): Writing 1 to this bit after one packet of data has been read from the endpoint 0 transmit FIFO buffer initializes the FIFO buffer, enabling the next packet to be received.

Bit 0—EP0i Packet Enable (EP0i PKTE): After one packet of data has been written to the endpoint 0 transmit FIFO buffer, the transmit data is fixed by writing 1 to this bit.

23.5.10 USBFIFO Clear Register (USBFCLR)

USBFCLR is provided to initialize the FIFO buffers for each endpoint. Writing 1 to a bit clears all the data in the corresponding FIFO buffer. The corresponding interrupt flag is not cleared. Do not clear a FIFO buffer during transmission/reception.

Bit:	7	6	5	4	3	2	1	0
	—	EP3 CLR	EP1 CLR	EP2 CLR	—	—	EP0o CLR	EP0i CLR
R/W:	W	W	W	W	W	W	W	W

Bit 7—Reserved

Bit 6—EP3 Clear (EP3 CLR): When 1 is written to this bit, the endpoint 3 transmit FIFO buffer is initialized.

Bit 5—EP1 Clear (EP1 CLR): When 1 is written to this bit, both FIFOs in the endpoint 1 receive FIFO buffer are initialized.

Bit 4—EP2 Clear (EP2 CLR): When 1 is written to this bit, both FIFOs in the endpoint 2 transmit FIFO buffer are initialized.

Bits 3 and 2—Reserved

Bit 1—EP0o Clear (EP0o CLR): When 1 is written to this bit, the endpoint 0 receive FIFO buffer is initialized.

Bit 0—EP0i Clear (EP0i CLR): When 1 is written to this bit, the endpoint 0 transmit FIFO buffer is initialized.

23.5.11 USBEP0o Receive Data Size Register (USBEPSZ0O)

USBEPSZ0O indicates, in bytes, the amount of data received from the host by endpoint 0.

23.5.12 USB Data Status Register (USBDASTS)

USBDASTS indicates whether the transmit FIFO buffers contain valid data. A bit is set when data is written to the corresponding FIFO buffer and the packet enable state is set, and cleared when all data has been transmitted to the host.

Bit:	7	6	5	4	3	2	1	0
	—	—	EP3 DE	EP2 DE	—	—	—	EP0i DE
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bits 7 and 6—Reserved: These bits are always read as 0. The write value should always be 0.

Bit 5—EP3 Data Present (EP3 DE): This bit is set when the endpoint 3 FIFO buffer contains valid data.

Bit 4—EP2 Data Present (EP2 DE): This bit is set when the endpoint 2 FIFO buffer contains valid data.

Bits 3 to 1—Reserved: These bits are always read as 0. The write value should always be 0.

Bit 0—EP0i Data Present (EP0i DE): This bit is set when the endpoint 0 FIFO buffer contains valid data.

23.5.13 USB Endpoint Stall Register (USBEPSTL)

The bits in USBEPSTL are used to forcibly stall the endpoints on the application side. While a bit is set to 1, the corresponding endpoint returns a stall handshake to the host. The stall bit for endpoint 0 (EP0 STL) is cleared automatically on reception of 8-bit command data for which decoding is performed by the function. When the SETUPTS flag in USB interrupt flag register 0 is set, a write of 1 to the EP0 STL bit is ignored. For details see section 23.8, Stall Operations.

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	EP3 STL	EP2 STL	EP1 STL	EP0 STL
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W	R/W

Bits 7 to 4—Reserved: These bits are always read as 0. The write value should always be 0.

Bit 3—EP3 Stall (EP3 STL): When this bit is set to 1, endpoint 3 is placed in the stall state.

Bit 2—EP2 Stall (EP2 STL): When this bit is set to 1, endpoint 2 is placed in the stall state.

Bit 1—EP1 Stall (EP1 STL): When this bit is set to 1, endpoint 1 is placed in the stall state.

Bit 0—EP0 Stall (EP0 STL): When this bit is set to 1, endpoint 0 is placed in the stall state.

23.5.14 USB Interrupt Enable Register 0 (USBIER0)

USBIER0 enables the interrupt requests indicated in USB interrupt flag register 0 (USBIFR0). When an interrupt flag is set while the corresponding bit in USBIER0 is set to 1, an interrupt request is sent to the CPU. The contents of the interrupt event register (INTEVT2) are determined by the contents of USB interrupt select register 0 (USBISR0).

Bit:	7	6	5	4	3	2	1	0
	BRST	EP1 FULL	EP2 TR	EP2 EMPTY	SETUP TS	EP0o TS	EP0i TR	EP0i TS
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

23.5.15 USB Interrupt Enable Register 1 (USBIER1)

USBIER1 enables the interrupt requests indicated in USB interrupt flag register 1 (USBIFR1). When an interrupt flag is set while the corresponding bit in USBIER1 is set to 1, an interrupt request is sent to the CPU. The contents of the interrupt event register (INTEVT2) are determined by the contents of USB interrupt select register 1 (USBISR1).

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	—	EP3 TR	EP3 TS	VBUSF
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W

23.5.16 USBEP1 Receive Data Size Register (USBEPSZ1)

USBEPSZ1 is the endpoint 1 receive data size register, indicating the amount of data received from the host. The endpoint 1 FIFO buffer has a dual-FIFO configuration; the receive data size indicated by this register refers to the currently selected FIFO.

23.5.17 USB Interrupt Select Register 0 (USBISR0)

USBISR0 selects the interrupt event register (INTEVT2) codes of the interrupt requests indicated in USB interrupt flag register 0. If the USB issues an interrupt request to the INTC when the corresponding bit in USBISR0 is cleared to 0, the interrupt will be USBFI0 (USB function interrupt 0), with an interrupt event register (INTEVT2) code of H'A20. If the USB issues an interrupt request to the INTC when the corresponding bit in USBISR0 is set to 1, the interrupt will be USBFI1 (USB function interrupt 1), with an interrupt event register (INTEVT2) code of H'A40. The initial value designates an interrupt event register (INTEVT2) code of H'A20. If interrupts occur simultaneously, USBFI0 has priority by default. For details on the interrupt event register (INTEVT2), refer to section 4, Exception Handling, and section 7, Interrupt Controller (INTC).

Bit:	7	6	5	4	3	2	1	0
	BSRT	EP1 FULL	EP2 TR	EP2 EMPTY	SETUP TS	EP0o TS	EP0i TR	EP0i TS
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

23.5.18 USB Interrupt Select Register 1 (USBISR1)

USBISR1 selects the interrupt event register (INTEVT2) codes of the interrupt requests indicated in USB interrupt flag register 1. If the USB issues an interrupt request to the INTC when the corresponding bit in USBISR1 is cleared to 0, the interrupt will be USBFI0 (USB function interrupt 0), with an interrupt event register (INTEVT2) code of H'A20. If the USB issues an interrupt request to the INTC when the corresponding bit in USBISR1 is set to 1, the interrupt will be USBFI1 (USB function interrupt 1), with an interrupt event register (INTEVT2) code of H'A40. The initial value designates an interrupt event register (INTEVT2) code of H'A20. If interrupts occur simultaneously, USBFI0 has priority by default. For details on the interrupt event register (INTEVT2), refer to section 4, Exception Handling, and section 7, Interrupt Controller (INTC).

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	—	EP3 TR	EP3 TS	VBUS
Initial value:	0	0	0	0	0	1	1	1
R/W:	R	R	R	R	R	R/W	R/W	R/W

23.5.19 USBDMA Setting Register (USBDMAR)

DMA transfer can be carried out between the endpoint 1 and endpoint 2 data registers by means of the on-chip DMA controller. Dual address transfer is performed, using byte transfer units. In order to start DMA transfer, DMA control settings must be made in addition to the settings in this register.

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	—	PULLUP_E	EP2 DMAE	EP1 DMAE
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W

Bits 7 to 3—Reserved: These bits are always read as 0. The write value should always be 0.

Bit 2—Pull-up Enable (PULLUP_E): This bit is for controlling connection notification (D + pull-up) to the USB host/hub. This bit enables the level of the USB1_pwr_en pin to be controlled.

Writing 1 outputs the High level and 0 outputs the Low level. For more information on the D + pull-up control, see section 22, USB Pin Multiplex Controller.

Bit 1—Endpoint 2 DMA Transfer Enable (EP2 DMAE): When this bit is set, DMA transfer is enabled from memory to the endpoint 2 transmit FIFO buffer. If there is at least one byte of space in the FIFO buffer, the transfer request signal to the DMA controller is asserted. When 64 bytes are written to the FIFO buffer in DMA transfer, EP2 packet enabling is set automatically, and 64-byte data can be transferred. If there is a space in another FIFO, a transfer request is asserted for the DMA controller again. However, since EP2 packet enable is not set automatically if data packet size for transfer is less than 64 bytes, set EP2 packet enabling by the CPU with a DMA transfer end interrupt.

Since EP2-related interrupt requests to the CPU are not masked automatically, interrupt requests must also be masked as necessary in the interrupt enable register.

Bit 0—Endpoint 1 DMA Transfer Enable (EP1 DMAE): When this bit is set, DMA transfer can be performed from the endpoint 1 receive FIFO buffer to memory. If there is at least one byte of space in the FIFO buffer, the transfer request signal to the DMA controller is asserted. When all received data is read in DMA transfer, the EP1 read-end trigger is performed automatically.

EP1-related interrupt requests to the CPU are not masked automatically.

23.6 Operation

23.6.1 Cable Connection

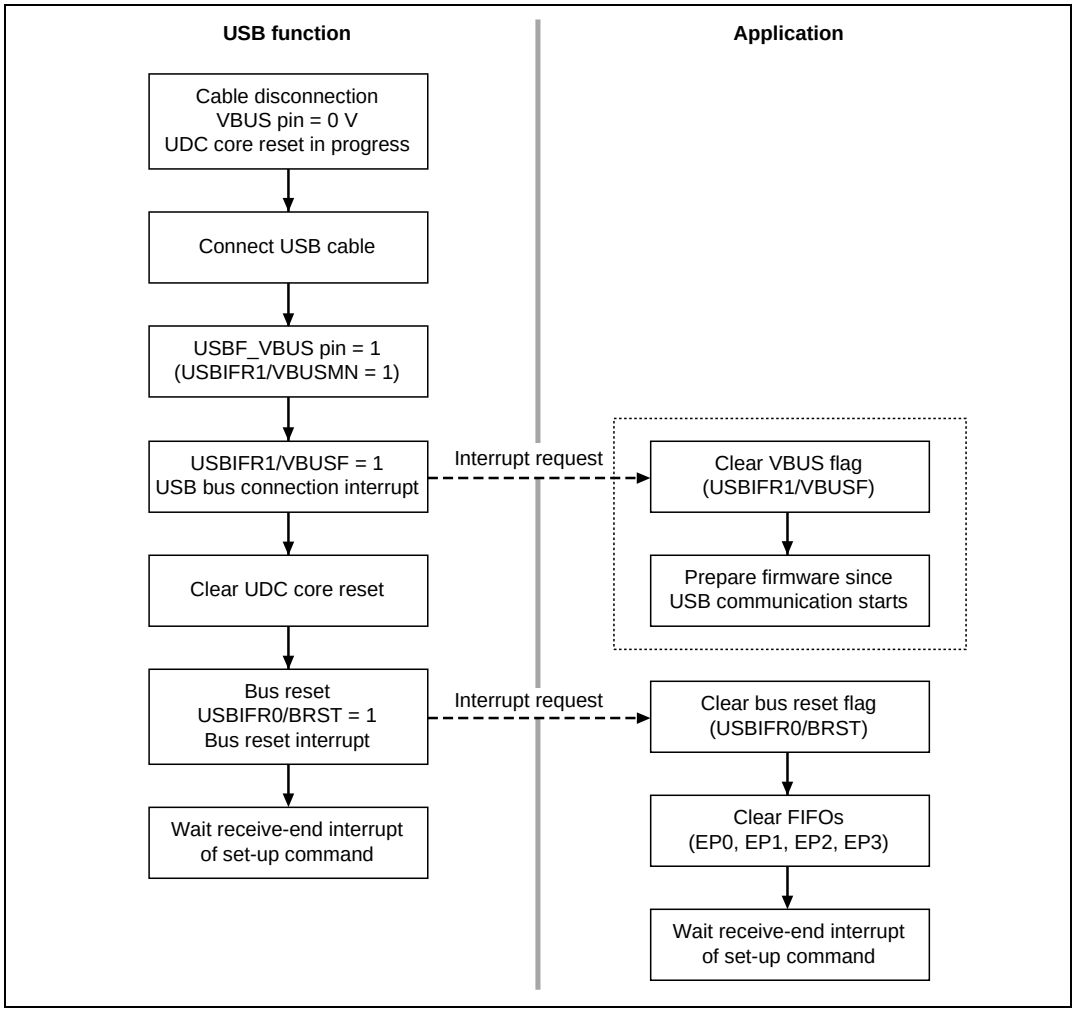


Figure 23.2 Cable Connection Operation

The above flowchart shows the operation in cable connection. When a USB connection interrupt occurs, connection can be detected by confirming the status of the USBF_VBUS pin.

Processing by a USB bus connection interrupt is unnecessary for applications in which USB cable connection is not required to be detected. Prepare processing by a bus reset interrupt.

23.6.2 Cable Disconnection

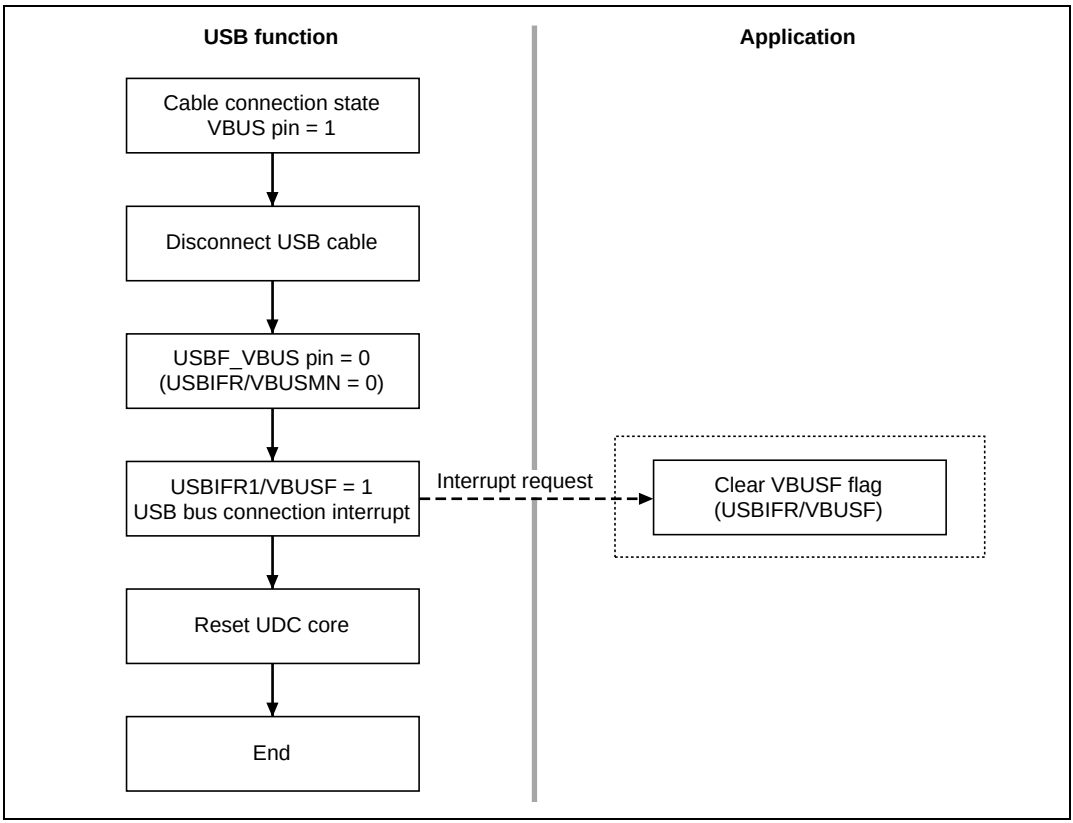


Figure 23.3 Cable Disconnection Operation

The above flowchart shows the operation in cable disconnection. When a USB bus connection interrupt occurs, disconnection can be detected by confirming the status of the USBF_VBUS pin.

23.6.3 Control Transfer

Control transfer consists of three stages of setup, data (that may not be included), and status (figure 23.4). The data stage consists of multiple bus transactions. The operating flowchart of each stage is shown below.

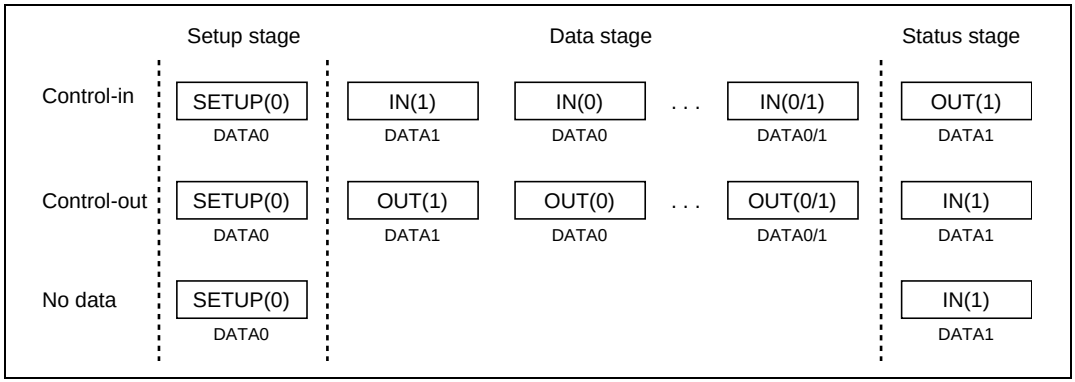
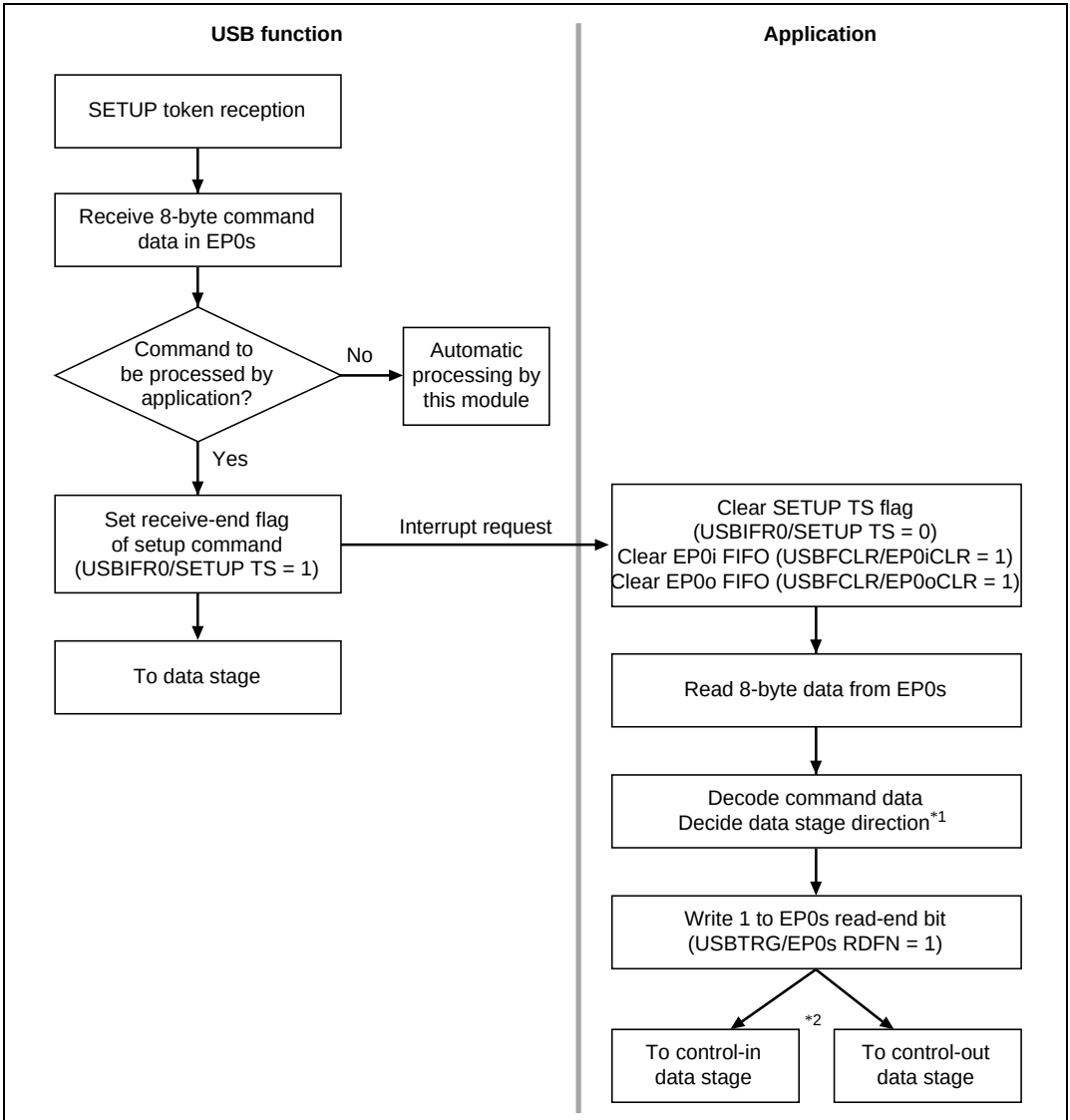


Figure 23.4 Transfer Stage for Control Transfer

(1) Setup Stage



Notes: 1. The application analyzes command data from the host that must be processed by the application in the setup stage, and decides the following processing methods (for instance, data stage direction).
2. When the transfer direction is control-out, enable an EP0i transfer request interrupt that is required in the status stage. When the transfer direction is control-in, disable the interrupt since it is not used.

Figure 23.5 Setup Stage Operation

(2) Data Stage (Control-In)

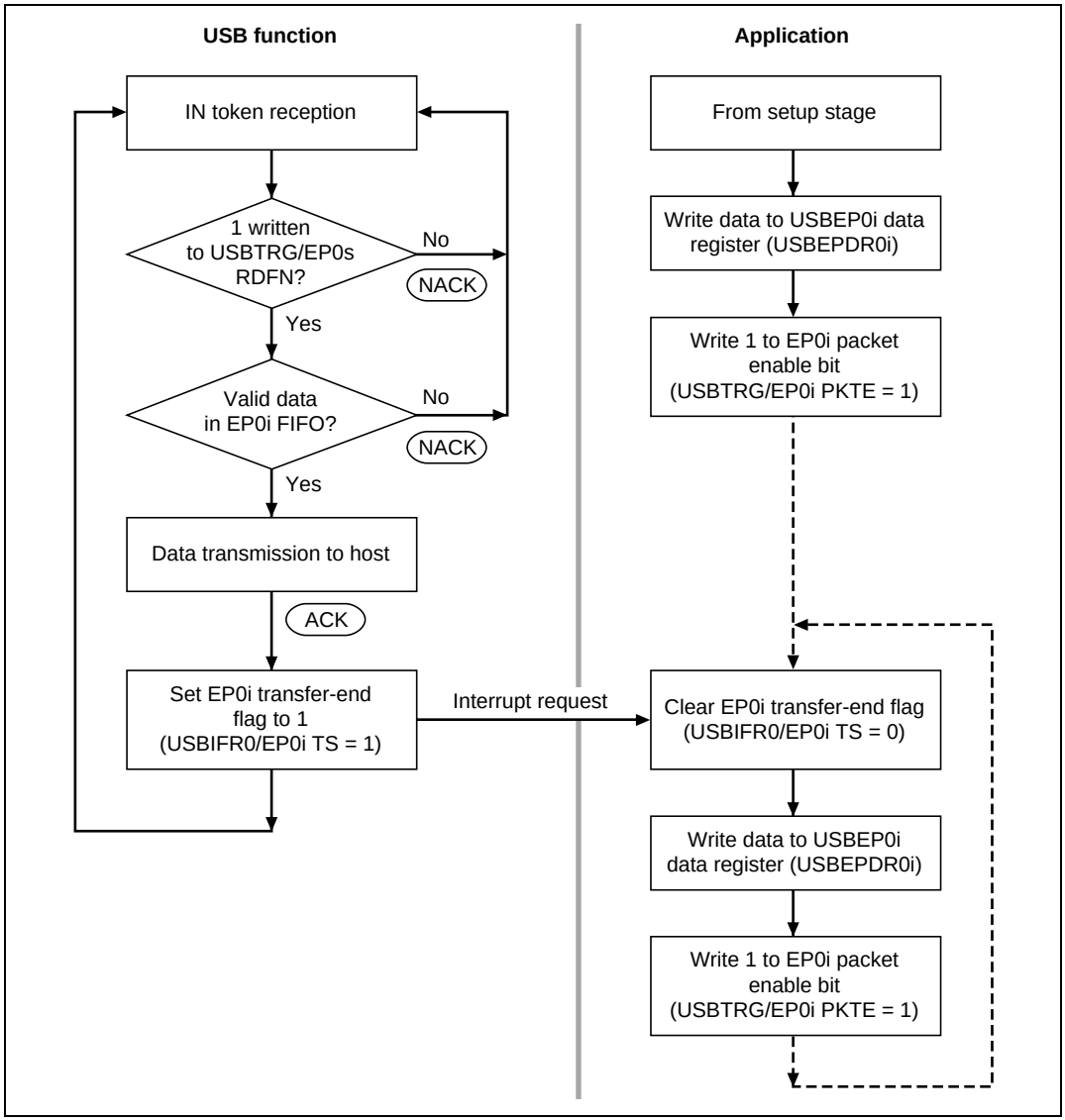


Figure 23.6 Data Stage Operation (Control-In)

The application analyzes command data from the host in the setup stage and decides the following data stage direction. As a result of command data analysis, when the data stage is in-transfer, one-packet data to be sent to the host is written to FIFO. If there is more data to be sent to the host, after data written first is sent to the host (USBIFR0/EP0i TS = 1), data is written to FIFO.

The end of the data stage is decided by transferring out-token by the host and entering the status stage.

Note: When the size of data transferred from the function is smaller than the size of data requested from the host, the function indicates data stage end by returning a packet smaller than the maximum packet to the host. When the size of data transferred from the function is integer times larger than the maximum packet size, a 0-length packet is transferred and the data stage end is indicated.

(3) Data Stage (Control-Out)

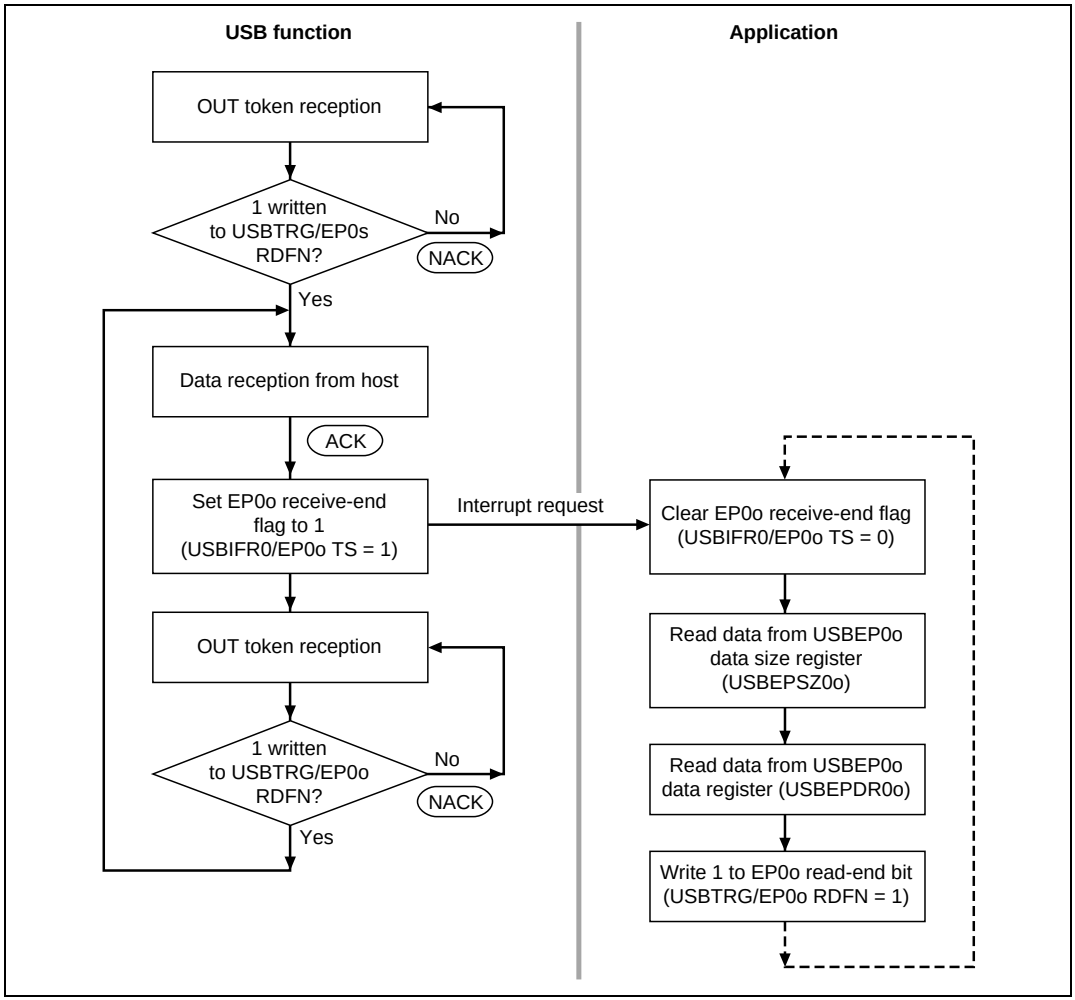


Figure 23.7 Data Stage Operation (Control-Out)

The application analyzes command data from the host in the setup stage and decides the following data stage direction. As a result of command data analysis, when the data stage is out-transfer, data from the host is waited. After data reception (USBIFR0/EP0o TS = 1), data is read from FIFO. Applications then write 1 to the EP0o read-end bit, make reception FIFO empty, and wait for the next data reception.

The end of the data stage is decided by transferring in-token by the host and entering the status stage.

(4) Status Stage (Control-In)

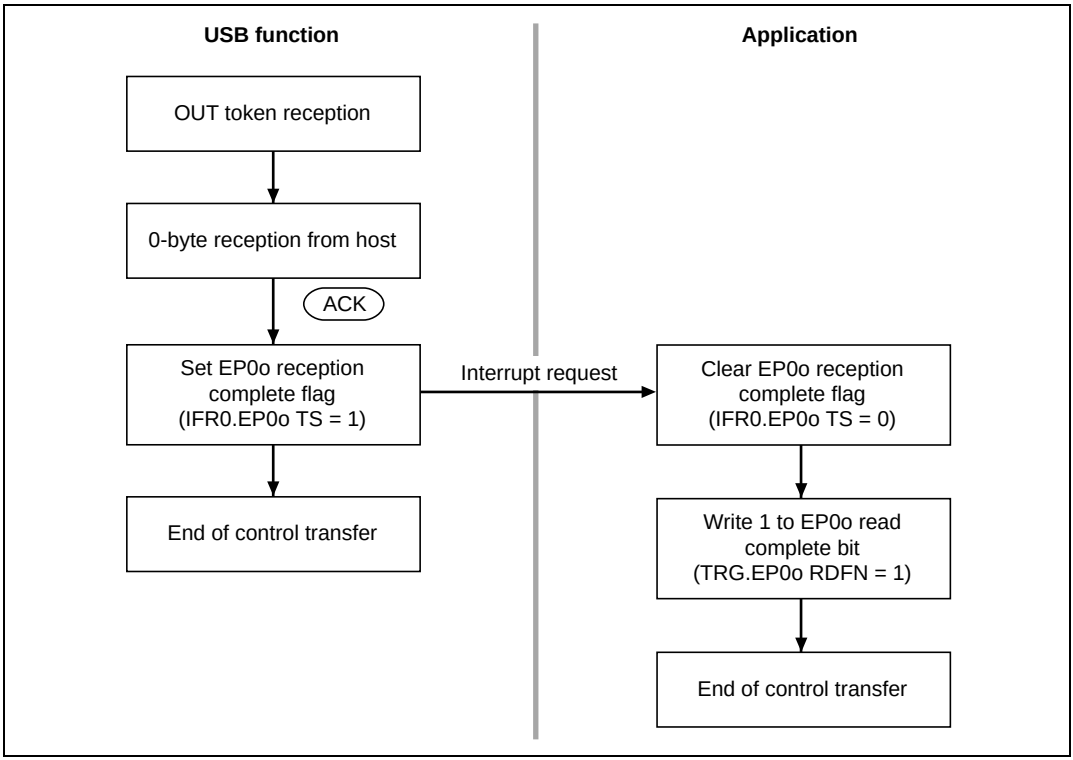


Figure 23.8 Status Stage Operation (Control-In)

The status stage in control-in starts with out-token from the host. The application receives 0-byte data from the host and completes control transfer.

(5) Status Stage (Control-Out)

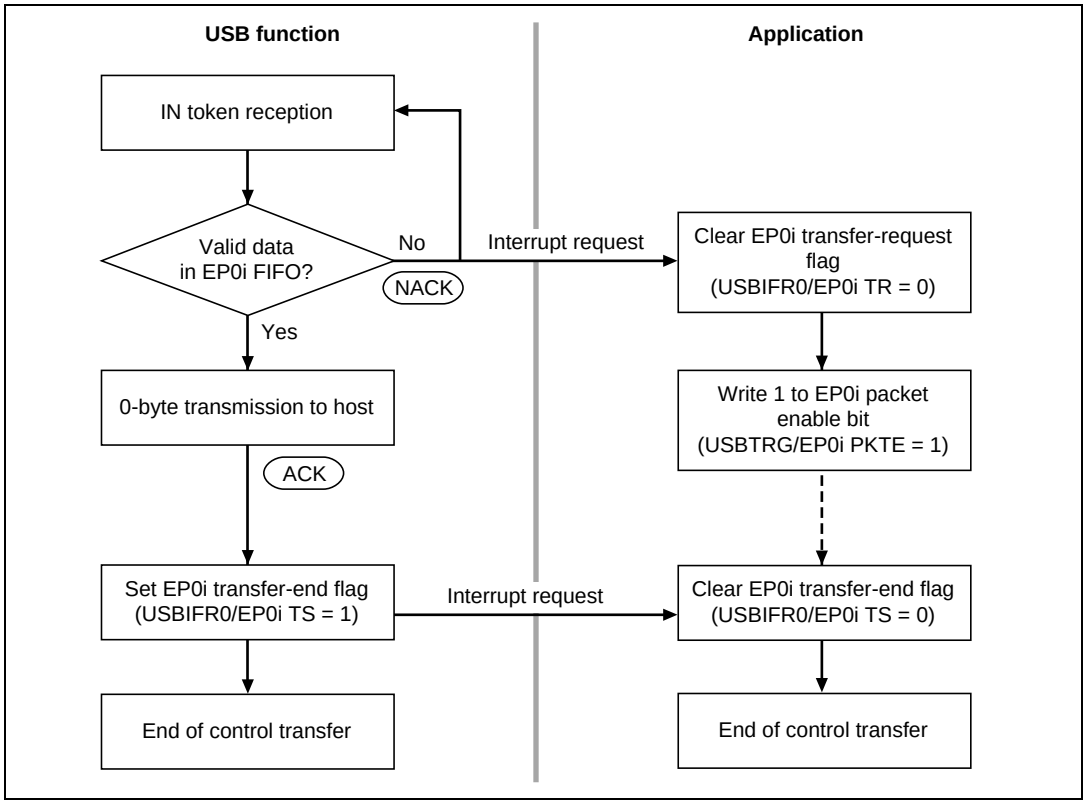


Figure 23.9 Status Stage Operation (Control-Out)

The status stage in control-out starts with in-token from the host. In in-token reception at the start of status stage, an EP0o transfer-request interrupt occurs since no data is in EP0i FIFO. The application acknowledges that the status stage has started by the interrupt. To transfer 0-byte data to the host, no data is written to the EP0i FIFO, and 1 is written to the EP0i packet-enable bit. Therefore, 0-byte data is transferred to the host in the next in-token, and control transfer is completed.

However, after the application completes all processing related to the data stage, write 1 to the EP0i packet-enable bit.

23.6.4 EP1 Bulk-Out Transfer (Dual FIFOs)

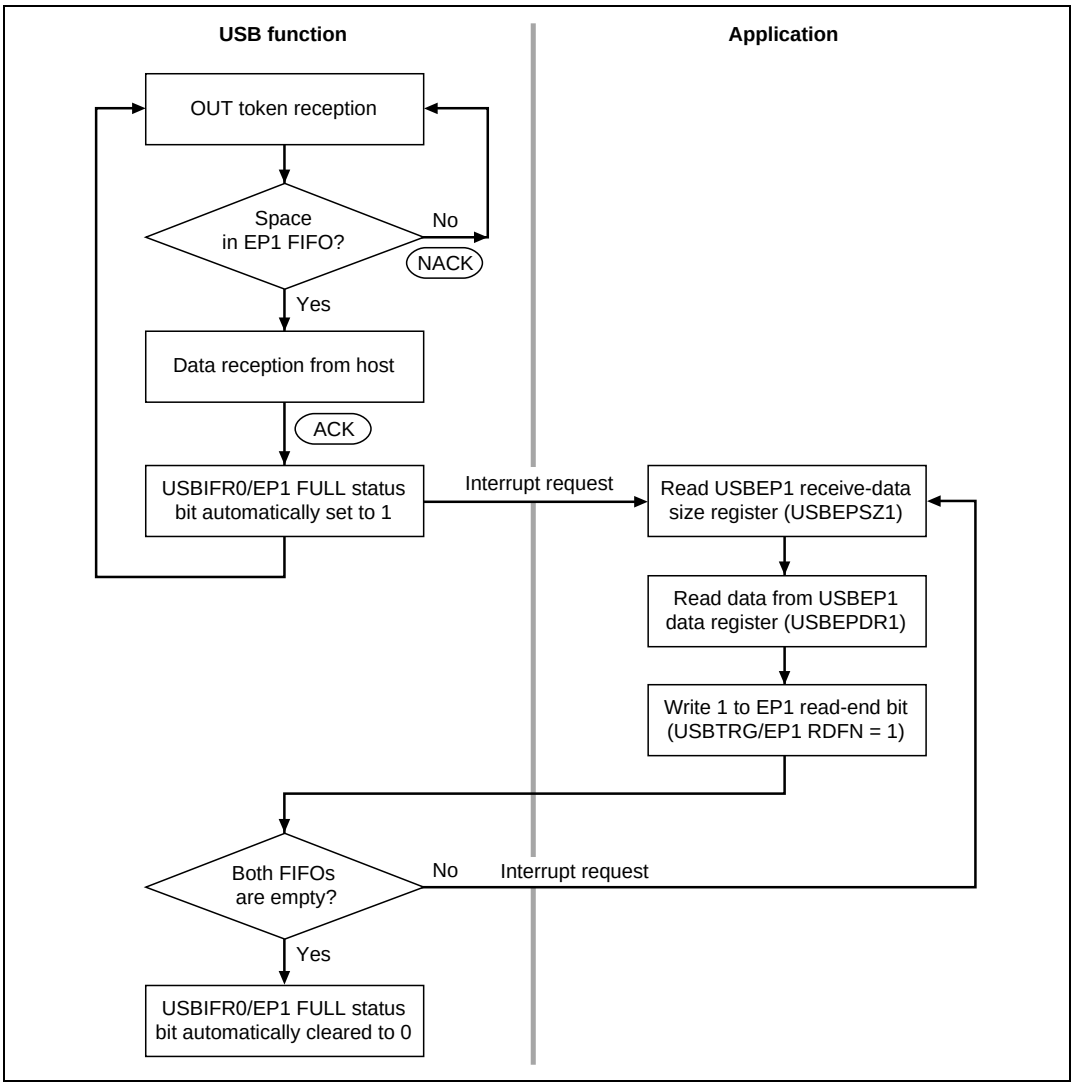


Figure 23.10 EP1 Bulk-Out Transfer Operation

EP1 has two 64-byte FIFOs, but the user can perform data reception and receive-data reads without being aware of this dual-FIFO configuration.

When one FIFO is full after reception is completed, the USBIFR0/EP1 FULL bit is set. After the first receive operation into one of the FIFOs when both FIFOs are empty, the other FIFO is empty,

and so the next packet can be received immediately. When both FIFOs are full, NACK is returned to the host automatically. When reading of the receive data is completed following data reception, 1 is written to the USBTRG/EP1 RDFN bit. This operation empties the FIFO that has just been read, and makes it ready to receive the next packet.

23.6.5 EP2 Bulk-In Transfer (Dual FIFOs)

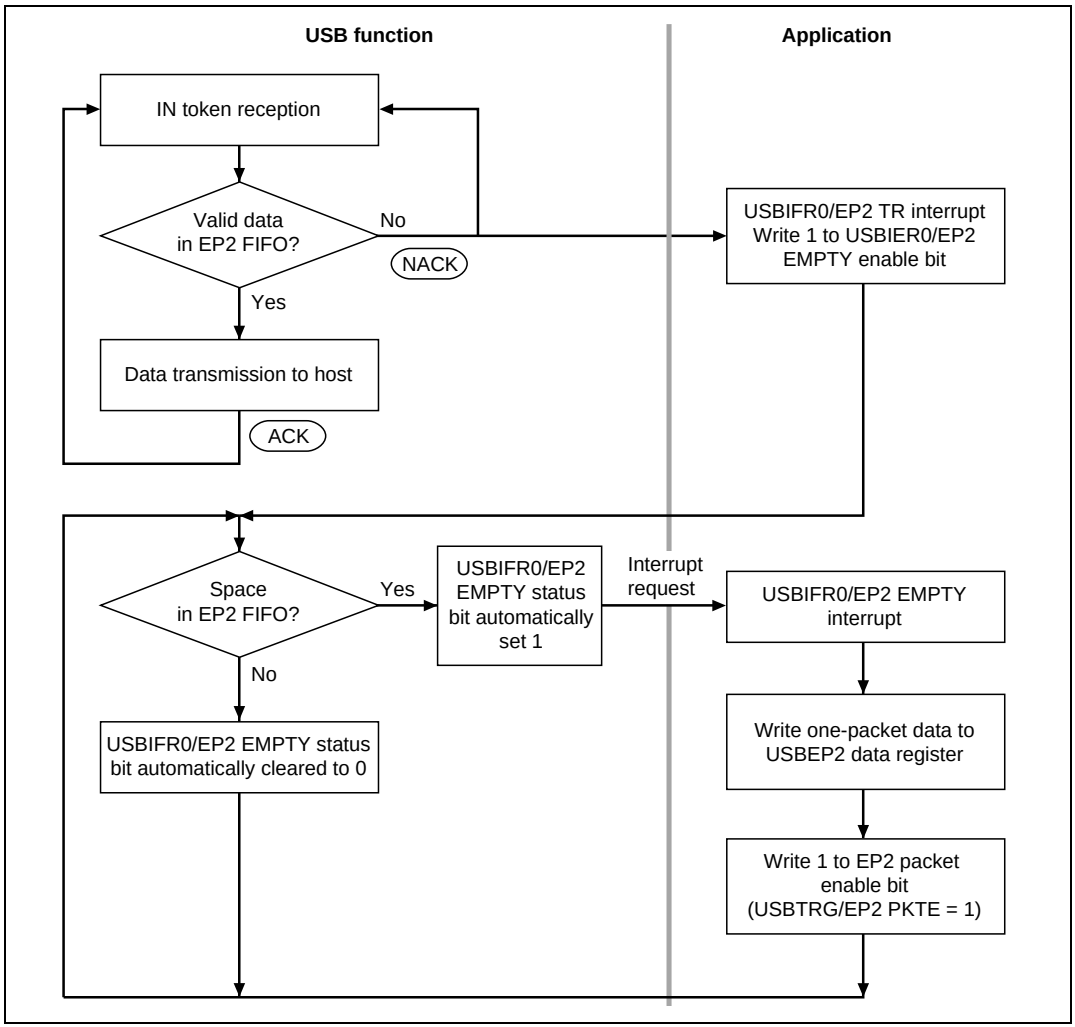


Figure 23.11 EP2 Bulk-In Transfer Operation

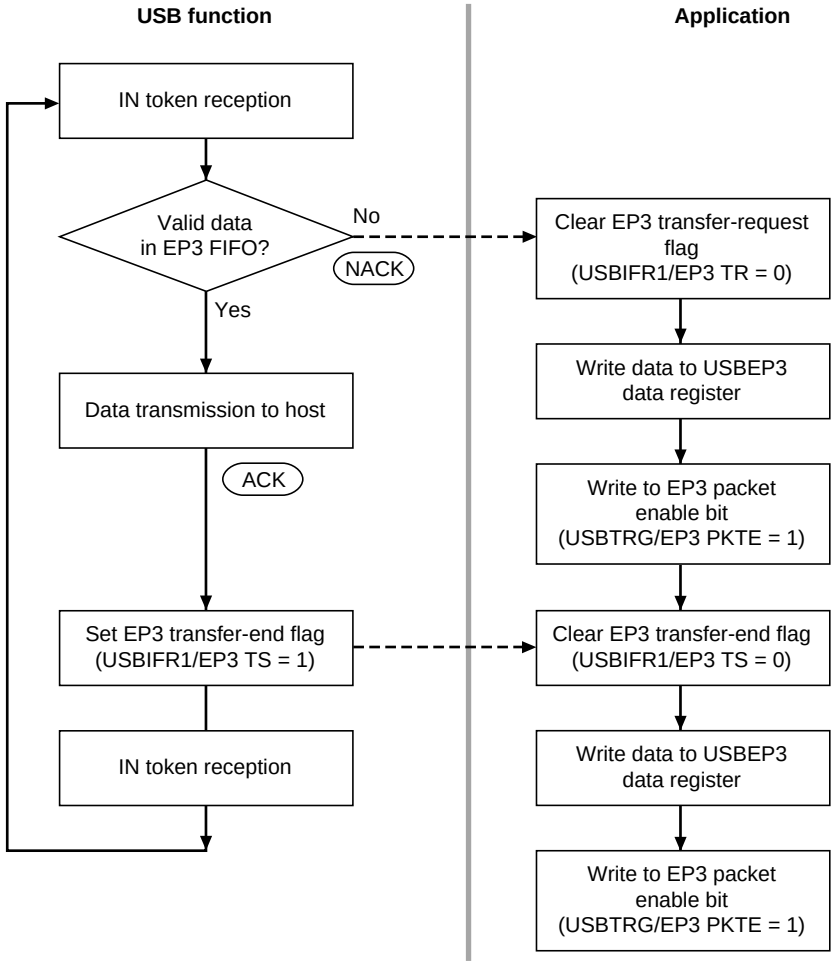
EP2 has two 64-byte FIFOs, but the user can perform data transmission and transmit-data writes without being aware of this dual-FIFO configuration. Write data to one FIFO at one time. For instance, even if two FIFOs are empty, EP2/PKTE cannot be performed after writing 128-byte data continuously. Perform EP2/PKTE in every 64-byte write.

To perform bulk-in transfer, since there is no valid data in FIFO in the first in-token, a USBIFR0/EP2 TR interrupt is requested. By the interrupt, write 1 to the USBIER0/EP2 EMPTY bit and enable the EP2 FIFO EMPTY interrupt. Since the two EP2 FIFOs are empty first, the EP2 FIFO EMPTY interrupt is generated immediately.

The data to be transmitted is written to the data register using this interrupt. After the first transmit data write, the other FIFO is empty, and so the next transmit data can be written immediately. When both FIFOs are full, EP2 EMPTY is cleared to 0. If at least one FIFO is empty, USBIFR0/EP2 EMPTY is set to 1. When ACK is returned from the host after data transmission is completed, the FIFO used in the data transmission becomes empty. If the other FIFO contains valid transmit data at this time, transmission is continued.

When transmission of all data has been completed, write 0 to USBIFR0/EP2 EMPTY and disable interrupt requests.

23.6.6 EP3 Interrupt-In Transfer



Note: This flow is an example of interrupt transfer processing. When there is data to be transferred, the following flow can also be considered.
"Confirm that FIFO is empty by referring to the EP3 DE bit of the USB data status register and write data to FIFO."

Figure 23.12 EP2 Interrupt-In Transfer Operation

23.7 Processing of USB Standard Commands and Class/Vendor Commands

23.7.1 Processing of Commands Transmitted by Control Transfer

A command transmitted from the host by control transfer may require decoding and execution of command processing on the application side. Whether command decoding is required on the application side is indicated in table 23.3 below.

Table 23.3 Command Decoding on Application Side

Decoding not Necessary on Application Side	Decoding Necessary on Application Side
Clear feature	Get descriptor
Get configuration	Sync frame
Get interface	Set descriptor
Get status	Class/Vendor command
Set address	
Set configuration	
Set feature	
Set interface	

If decoding is not necessary on the application side, command decoding and data stage and status stage processing are performed automatically. No processing is necessary by the user. An interrupt is not generated in this case.

If decoding is necessary on the application side, the USB function module stores the command in the EP0s FIFO. After normal reception is completed, the USBIER0/SETUP TS flag is set and an interrupt request is generated. In the interrupt routine, 8 bytes of data must be read from the EP0s data register (USBEPDR0S) and decoded by firmware. The necessary data stage and status stage processing should then be carried out according to the result of the decoding operation.

23.8 Stall Operations

23.8.1 Overview

This section describes stall operations in the USB function module. There are two cases in which the USB function module stall function is used:

- When the application forcibly stalls an endpoint for some reason
- When a stall is performed automatically within the USB function module due to a USB specification violation

The USB function module has internal status bits that hold the status (stall or non-stall) of each endpoint. When a transaction is sent from the host, the module references these internal status bits and determines whether to return a stall to the host. These bits cannot be cleared by the application; they must be cleared with a Clear Feature command from the host.

23.8.2 Forcible Stall by Application

The application uses the USBEPSTL register to issue a stall request for the USB function module. When the application wishes to stall a specific endpoint, it sets the corresponding bit in USBEPSTL (1-1 in figure 23.13). The internal status bits are not changed. When a transaction is sent from the host for the endpoint for which the USBEPSTL bit was set, the USB function module references the internal status bit, and if this is not set, references the corresponding bit in USBEPSTL (1-2 in figure 23.13). If the corresponding bit in USBEPSTL is set, the USB function module sets the internal status bit and returns a stall handshake to the host (1-3 in figure 23.13). If the corresponding bit in USBEPSTL is not set, the internal status bit is not changed and the transaction is accepted.

Once an internal status bit is set, it remains set until cleared by a Clear Feature command from the host, without regard to the USBEPSTL register. Even after a bit is cleared by the Clear Feature command (3-1 in figure 23.13), the USB function module continues to return a stall handshake while the bit in USBEPSTL is set, since the internal status bit is set each time a transaction is executed for the corresponding endpoint (1-2 in figure 23.13). To clear a stall, therefore, it is necessary for the corresponding bit in USBEPSTL to be cleared by the application, and also for the internal status bit to be cleared with a Clear Feature command (2-1, 2-2, and 2-3 in figure 23.13).

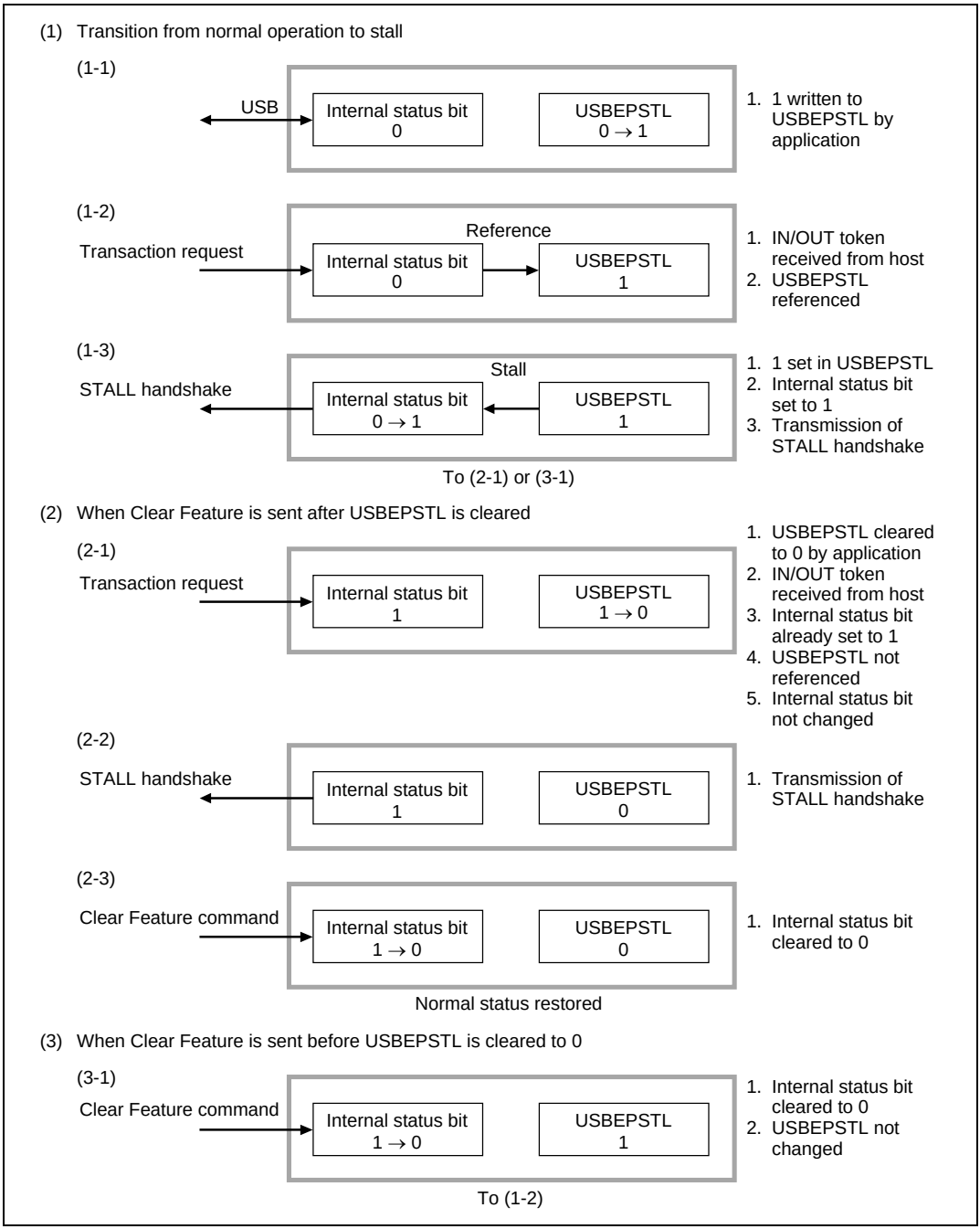


Figure 23.13 Forcible Stall by Application

23.8.3 Automatic Stall by USB Function Module

When a stall setting is made with the Set Feature command, or in the event of a USB specification violation, the USB function module automatically sets the internal status bit for the relevant endpoint without regard to the USBEPSTL register, and returns a stall handshake (1-1 in figure 23.14).

Once an internal status bit is set, it remains set until cleared by a Clear Feature command from the host, without regard to the USBEPSTL register. After a bit is cleared by the Clear Feature command, USBEPSTL is referenced (3-1 in figure 23.14). The USB function module continues to return a stall handshake while the internal status bit is set, since the internal status bit is set even if a transaction is executed for the corresponding endpoint (2-1 and 2-2 in figure 23.14). To clear a stall, therefore, the internal status bit must be cleared with a Clear Feature command (3-1 in figure 23.14). If set by the application, USBEPSTL should also be cleared (2-1 in figure 23.14).

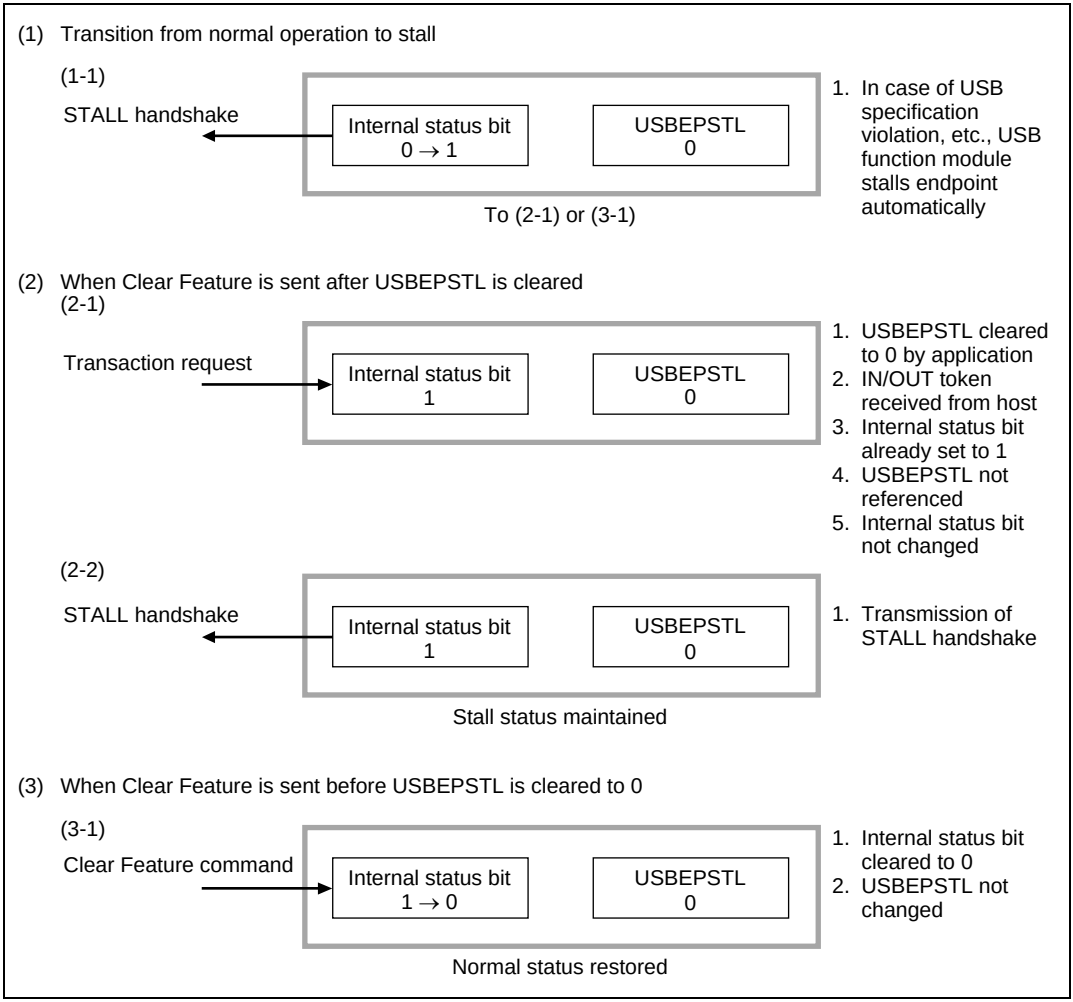


Figure 23.14 Automatic Stall by USB Function Module

23.9 Usage Notes

23.9.1 Receiving Setup Data

Note the following for EPDR0s that receives 8-byte setup data:

1. As a latest setup command must be received in high priority, the write from the USB bus takes priority over the read from the CPU. If the next setup command reception is started while the CPU is reading data after the data is received, the read from the CPU is forcibly terminated. Therefore, the data read after reception is started becomes invalid.
2. EPDR0s must always be read in 8-byte units. If the read is terminated at a midpoint, the data received at the next setup cannot be read correctly.

23.9.2 Clearing the FIFO

If a USB cable is disconnected during data transfer, the data being received or transmitted may remain in the FIFO. When disconnecting a USB cable, clear the FIFO.

While a FIFO is transferring data, it must not be cleared.

23.9.3 Overreading and Overwriting the Data Registers

Note the following when reading or writing to a data register of this module.

(1) Receive data registers

The receive data registers must not be read exceeding the valid amount of receive data, that is, the number of bytes indicated by the receive data size register. Even for EPDR1 which has double FIFO buffers, the maximum data to be read at one time is 64 bytes. After the data is read from the current valid FIFO buffer, be sure to write 1 to EP1RDFN in TRG, which switches the valid buffer, updates the receive data size to the new number of bytes, and enables the next data to be received.

(2) Transmit data registers

The transmit data registers must not be written to exceeding the maximum packet size. Even for EPDR2 which has double FIFO buffers, write data within the maximum packet size at one time. After the data is written, write 1 to PKTE in TRG to switch the valid buffer and enable the next data to be written. Data must not be continuously written to the two FIFO buffers.

23.9.4 Assigning Interrupt Sources to EP0

The EP0-related interrupt sources indicated by the interrupt source bits (bits 0 to 3) in IFR0 must be assigned to the same interrupt signal with ISR0. The other interrupt sources have no limitations.

23.9.5 Clearing the FIFO when DMA Transfer Is Enabled

The endpoint 1 data register (EPDR1) cannot be cleared when DMA transfer for endpoint 1 is enabled (EP1 DMAE in USBDMA = 1). Cancel DMA transfer before clearing the register.

23.9.6 Notes on TR Interrupt

Note the following when using the transfer request interrupt (TR interrupt) for IN transfer to EP0i, EP2, or EP3.

The TR interrupt flag is set if the FIFO for the target EP has no data when the IN token is sent from the USB host. However, at the timing shown in figure 23.15, multiple TR interrupts occur successively. Take appropriate measures against malfunction in such a case.

Note: This module determines whether to return NAK if the FIFO of the target EP has no data when receiving the IN token, but the TR interrupt flag is set only after a NAK handshake is sent. If the next IN token is sent before PKTE of TRG is written to, the TR interrupt flag is set again.

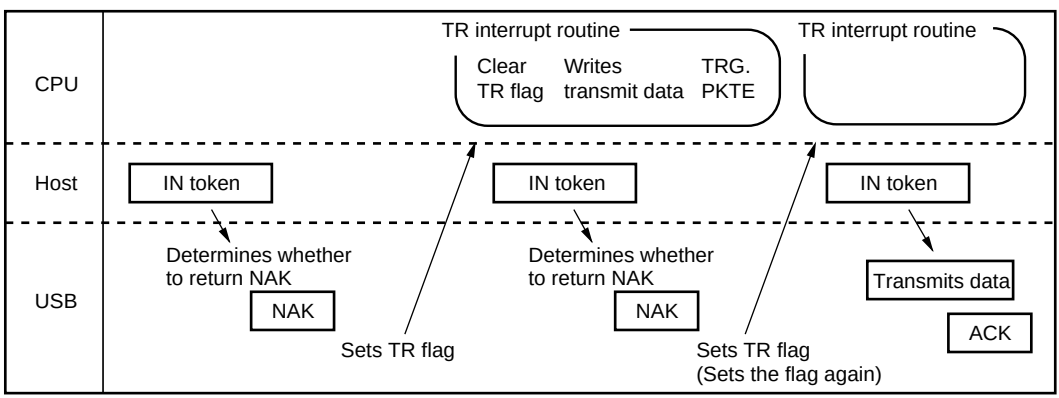


Figure 23.15 TR Interrupt Flag Set Timing

23.9.7 Peripheral Clock (Pϕ) Operation Frequency

The peripheral clock frequency (Pϕ) should be 13 MHz or above when using this module.

Section 24 USB HOST Module

24.1 General Description

The USB Host Controller module incorporated in SH7727 supports Open Host Controller (Open HCI) Specification for the Universal Serial Bus (USB) as well as the Universal Serial Bus specification ver.1.1.

The Open HCI Specification for the USB is a register-level description of Host Controller for the USB which in turn is described by the USB specification.

It is necessary to refer Open HCI specification to develop drivers for this USB Host Controller and hardware.

24.1.1 Features

- Support open HCI standard ver.1.0 register set
- Support Universal Serial Bus standard ver.1.1
- Root Hub function
- Support Full speed (12Mbps) mode and Low speed (1.5Mbps) mode
- Support Overcurrent detection
- Support 127 endpoints control in maximum
- The whole area of the synchronous DRAM in area 3 connected to the CPU can be used for transfer data and descriptor.

24.1.2 Pin Configuration

Pin configuration of the USB host controller is shown in table 24.1.

For the detailed method for setting each pin, see section 22, USB Pin Multiplex Controller.

Table 24.1 Pin Configuration

Pin name	Symbol	I/O	Function
Power enable pin 1	USB1_pwr_en	Output	USB port 1 power enable control
Power enable pin 2	USB2_pwr_en	Output	USB port 2 power enable control
Over-current pin 1/ VBUS pin	$\overline{\text{USB1_ovr_current}}$/ USBF_VBUS	Input	USB port 1 over-current detect/ USB cable connection monitor pin
Over-current pin 2	$\overline{\text{USB2_ovr_current}}$	Input	USB port 2 over-current detect
1P pin	USB1P	I/O	D+ port 1 transceiver pin
1M pin	USB1M	I/O	D– port 1 transceiver pin
2P pin	USB2P	I/O	D+ port 2 transceiver pin
2M pin	USB2M	I/O	D– port 2 transceiver pin

24.1.3 Register Configuration

Table 24.2 Register Configuration

Register Name	Symbol	R/W	Initial Value	Address	Access Size
HcRevision	—	R	H'00000010	H'04000400	32
HcControl	—	R/W	H'00000000	H'04000404	32
HcCommandStatus	—	R/W	H'00000000	H'04000408	32
HcInterruptStatus	—	R/W	H'00000000	H'0400040C	32
HcInterruptEnable	—	R/W	H'00000000	H'04000410	32
HcInterruptDisable	—	R/W	H'00000000	H'04000414	32
HcHCCA	—	R/W	H'00000000	H'04000418	32
HcPeriodCurrentE	—	R/W	H'00000000	H'0400041C	32
HcControlHeadED	—	R/W	H'00000000	H'04000420	32
HcControlCurrentED	—	R/W	H'00000000	H'04000424	32
HcBulkHeadED	—	R/W	H'00000000	H'04000428	32
HcBulkCurrentED	—	R/W	H'00000000	H'0400042C	32
HcDonrHeadED	—	R/W	H'00000000	H'04000430	32
HcFmInterval	—	R/W	H'00002EDF	H'04000434	32
HcFmRemaining	—	R	H'00000000	H'04000438	32
HcFmNumber	—	R	H'00000000	H'0400043C	32
HcPeriodicStart	—	R/W	H'00000000	H'04000440	32
HcLSThreshold	—	R/W	H'00000628	H'04000444	32
HcRhDescriptorA	—	R/W	H'02001202	H'04000448	32
HcRhDescriptorB	—	R/W	H'00000000	H'0400044C	32
HcRhStatus	—	R/W	H'00000000	H'04000450	32
HcRhPortStatus1	—	R/W	H'00000100	H'04000454	32
HcRhPortStatus2	—	R/W	H'00000100	H'04000458	32

24.2 Register Description

24.2.1 HcRevision

HcRevision Register (H'04000400)

Register: *HcRevision*			**Offset: 00–03**
Bits	**Reset**	**R/W**	**Description**
31–8	0h	—	Reserved.
7–0	10h	R	**Revision (Rev)** These read only bits include the BCD expression of the HCI specification version implemented for the host controller. The value H'10 corresponds to version 1.0. All HCI implementation complying to this specification have the value of H'10.

24.2.2 HcControl

HcControl Register (H'04000404)

The HcControl register defines the operation mode for the host controller. Most of bits of this register are amended only by the host controller driver other than HostController Function State and Remote Wakeup Command.

Register: *HcControl*			**Offset: 04–07**
Bits	**Reset**	**R/W**	**Description**
31–11	0h	—	Reserved. Read/Write 0's
10	0b	R/W	**RemoteWakeupEnable (RWE)** This bit is used by HCD to enable/disable the remote wakeup function at the same time as the detection of an upstream resume signal. This function is not supported. Be sure to write 0.
9	0b	R/W	**RemoteWakeupConnected (RWC)** This bit indicates whether the host controller supports a remote wakeup signal or not. When the remote wakeup is supported and used in the system, the host controller must set this bit between POST in the system firmware. The host controller clears the bit at the same time of the hardware reset, however, does not change at the same time as the software reset. The remote wakeup signal to the system of the host is specific for the host bus, so it is not described in this specification. 0: Remote wakeup signal is not supported. (initial value) 1: Remote wakeup signal is supported.
8	0b	R/W	**InterruptRouting (IR)** This bit determines the routing of interrupts generated by the event registered in HcInterruptStatus. HCD clears this bit at the same time as the hardware reset, however, does not clear at the same time as the software reset. HCD uses this bit as a tag to indicate the ownership of the host controller. 0: All interrupts are routed to normal bus interrupt mechanism. (initial value) 1: Interrupts are routed to SMI.

Register: *HcControl* **Offset: 04–07**

Bits	Reset	R/W	Description
7, 6	00b	R/W	**HostControllerFunctionalState (HCFS)** HCD determines whether the host controller has started to route SOF after having read the StartofFrame bit of HcInterruptStatus. This bit can be changed by the host controller only in the UsbSuspend state. The host controller can move from the UsbSuspend state to the UsbResume state after having detected the resume signal from the downstream port. In the host controller, UsbSuspend is entered after the software reset so that UsbReset is entered after the hardware reset. The former resets the route hub. 00: USB reset 01: USB resume 10: USB operation 11: USB suspend
5	0b	R/W	**BulkListEnable (BLE)** This bit is set to enable the processing of the bulk list in the next frame. The host controller checks this bit when the processing of the list has been determined. When disabling, HCD can correct the list. When HcBulkCurrentED indicates ED to be deleted, HCD should hasten the pointer by updating HcBulkCurrentED before re-enabling the list processing. 0: Bulk list processing is not carried out. (initial value) 1: Bulk list processing is carried out.
4	0b	R/W	**ControlListEnable (CLE)** This bit is set to enable the processing of the control list in the next frame. If cleared by HCD, the processing of the control list is not carried out after next SLF. The host controller must check this bit whenever the list will be processed. When disabling, HCD can correct the list. When HcControlCurrentED indicates ED to be deleted, HCD should hasten the pointer by updating HcBulk before re-enabling the list processing. 0: Control list processing is not carried out. (initial value) 1: Control list processing is carried out.

Register: *HcControl*			Offset: 04–07
Bits	**Reset**	**R/W**	**Description**
3	0b	R/W	**IsochronousEnable (IE)** This bit is used by HCD to enable/disable the processing of isochronous ED. While processing the periodic list, HC will check the status of this bit when it finds an isochronous ED (F =1). If set (enabled), the host controller continues to process ED. If cleared (disabled), the host controller stops the processing of the periodic list (currently includes only isochronous ED) and starts to process the bulk/control list. Setting this bit is guaranteed to be valid in the next frame (not in the current frame). 0: Processes ED. (initial value) 1: Processes the bulk/control list.
2	0b	R/W	**PeriodicListEnable (PLE)** This bit is set to enable the processing of the periodic list. If cleared by HCD, no periodic list processing is carried out after next SOF. HC must check this bit before HC starts to process the list. 0: The periodic list processing is not carried out after next SOF. (initial value) 1: The periodic list processing is carried out after next SOF.
1, 0	00b	R/W	**ControlBulkServiceRatio (CBSR)** This bit specifies the service ration of the control and bulk ED. The host controller must compare the ratio specified by the internal calculation whether it has processed several non-vacant control ED in determining whether another control ED is continued to be supplied or switched to bulk ED before any a periodic list is processed. In case of reset, HCD is responsible for restoring this value. 00: 1:1 (initial value) 01: 2:1 10: 3:1 11: 4:1

24.2.3 HcCommandStatus

HcCommandStatus Register (H'04000408)

The HcCommandStatus register is used by the host controller not only for reflecting the current status of the host controller, but also for receiving a command issued by the host controller driver. A write is for setting the host controller driver. The host controller must guarantee that the bit to which 1 is written to is set and the bit to which 0 is written to is unchanged. The host controller driver must distribute multiple clear commands to the host controller by a previously issued command. The host controller driver can read all bits normally.

The SchedulingOverrunCount bit indicates the number of the frame that has detected the Scheduling Overrun error by the host controller. This occurs when the periodic list has not completed before EOF. When the Scheduling Overrun error is detected, the host controller increments the counter and sets SchedulingOverrun in the HcInterruptStatus register.

Register: *HcCommandStatus* **Offset: 08–0B**

Bits	Reset	R/W	Description
31–18	0h	—	Reserved.
17–16	00b	R	**SchedulingOverrunCount (SOC)** These bits are incremented in each SchedulingOverrun error. These bits are initially set to B'00 and returned to B'11. These bits are incremented when SchedulingOverrun is detected even though the SchedulingOverrun bit in HcInterruptStatus is set. These bits are used by HCD to monitor any continuous scheduling problem.
15–4	0h	—	Reserved.
3	0b	R/W	**OwnershipChangeRequest (OCR)** This bit is set by OS HCD to request the change of the control of the host controller. When this bit is set, the host controller sets the OwnershipChange bit in the HcInterruptStatus register. After a change, this bit is cleared and remains until the next request from OS HCD. 0: After a change, this bit is cleared and remains until the next request from OS HCD. (initial value) 1: Set the OwnershipChange bit in the HcInterruptStatus register.

Register: *HcCommandStatus* **Offset: 08–0B**

Bits	Reset	R/W	Description
2	0b	R/W	**BulkListFilled (BLF)** This bit is used to indicate that there are some TDs in the list. This bit is set by HCD to the list when TD is added to ED. When the host controller starts to process the head of the list, it checks BF. As long as BulkListFilled is 0, the host controller does not start to process the list. When BulklistFilled is 1, the host controller starts to process the list to set BF to 0. When the host controller finds TD in the list, the host controller sets BulkListFilled to 1. When TD is never found in the list and HCD does not set BulkListFilled, the host controller completes the processing of the list. BulklistFilled is still 0 when the size list processing is stopped. 0: The list is not processed (initial value) 1: The list is processed.
1	0b	R/W	**ControlListFilled (CLF)** This bit is used to indicate that there are some TDs in the control list. This bit is set by HCD when TD is added to ED in the control list. When the host controller starts to process the head of the control list, it checks CLK. As long as ControlListFilled is 0, the host controller does not start to process the control list. If CF is 1, the host controller starts to process the control list and ControlListFilled is set to 0. When the host controller finds TD in the list, the host controller sets ControlListFilled to 1. When TD is never found in the control list and HCD does not set ControlListFilled, the host controller completes the processing of the control list. ControllListFilled is still 0 when the control list processing is stopped. 0: The list is not processed. (initial value) 1: The list is processed.

Register: *HcCommandStatus* **Offset: 08–0B**

Bits	Reset	R/W	Description
0	0b	R/W	**HostControllerReset (HCR)** This bit is set by HCD to initiate the software reset of the host controller. The system is moved to the UsbSuspend state in which most of the operational registers are reset except for the next state regardless of the functional state of the host controller. For example, an access to the InterrupRouting field in HcControl and without host bus are allowed. This bit is cleared by the host controller upon completion of the reset operation. This bit does not cause any reset to the route hub and the next reset signal is not issued to the downstream port. 0: Cleared by the host controller at the completion of the reset control 1: UsbSuspend state

24.2.4 HcInterruptStatus

HcInterruptStatus Register (H'0400040C)

This register indicates the status in various events that cause hardware interrupts. When an event occurs, the host controller sets the corresponding bit in this register. When the bit is set to 1, a hardware interrupt is generated while an interrupt is enabled and the MsterInterrupEnable bit is set in the HcInterruptEnable register (see section 24.2.5, HcInterruptEnable). The host control driver clears a specified bit in this register by writing 1 in the bit position to be cleared. The host controller driver cannot set any bit of these bits. The host controller never clears bits.

Register: *HcInterruptStatus* **Offset: 0C–0F**

Bits	Reset	R/W	Description
31	0h	—	Reserved.
30	0b	R/W	**OwnershipChange (OC)** This bit is set by the host controller when the OwnershipChangeRequest bit in the HcCommandStatus reigster is set. This event generates a system management interrupt at once when not masked. When there is no SMI pin, this bit is set to 0. 0: The OCR bit in the HcCommandStatus register is not set. (initial value) 1: The OCR bit in the HcCommandStatus register is set.
29–7	0h	—	Reserved.
6	0b	R/W	**RootHubStatusChange (RHSC)** This bit is set when the content of HcRhStatus or the content of any HcRhPortStatus 1, 2 register [Number of Downstream Port] has changed. 0: The content of the HcRhStatus register or HcRhPortStatus register is not changed. (initial value) 1: The content of the HcRhStatus register or HcRhPortStatus register is changed.
5	0b	R/W	**FrameNumberOverflow (FNO)** This bit is set when MSB (bit 15) in the HcFumnumber register changes value from 0 to 1 or from 1 to 0 or the HccaFrameNumber bit is updated. 0: MSB or the HccaFrameNumber bit in the HcFmNumber register is not updated. (initial value) 1: MSB or the HccaFrameNumber bit in the HcFmNumber register is updated.

Register: *HcInterruptStatus* **Offset: 0C–0F**

Bits	Reset	R/W	Description
4	0b	R/W	**UnrecoverableError (UE)** This bit is set when the host controller detects a system error that is not related to USB. HCD clears this bit after the host controler is reset. 0: System error has not generated yet. (initial value) 1: System error is detected.
3	0b	R/W	**ResumeDetected (RD)** This bit is set when the host controller detects that a device of USB issues a resume signal. This bit is not set when HCD sets UsbResume state. 0: The resume signal is not detected. (initial value) 1: The resume signal is detected.
2	0b	R/W	**StartofFrame (SF)** This bit is set by the host controller when each frame starts and after the HccaFrameNumber is updated. The host controller simultaneously generates the SOF token. 0: Each frame has not initiated or HccaFrame Number is not updated (initial value) 1: Initiation of each frame and updating of HccaFrameNumber
1	0b	R/W	**WritebackDoneHead (WDH)** This bit is set immediately after the host controller has written HcDoneHead to HccaDoneHead. HccaDoneHead is not updated until this bit is cleared. HCD should clear this bit only after the content of HccaDoneHead has been stored. 0: When cleared after set to 1. (initial value) 1: When HcDoneHead is written to HccaDonehead.
0	0b	R/W	**SchedulingOverrun (SO)** This bit is set when the USB schedule has overrun after HccaFrameNumber has updated. SchedulingOverrun also increments the SchedulingOverrunCount bit in HcCommandStatus. 0: The USB schedule has not overrun. (initial value) 1: The USB schedule has overrun.

24.2.5 HcInterruptEnable

HcInterrutpEnable Register (H'04000410)

Each enable bit in the HcInterruptEnable register corresponds to the related interrupt bit in the HcInterruptStatus register. The HcInterruptEnable register is used to control an event to generate a hardware interrupt. A hardware interrupt is requested in the host bus when a bit in the HcInterruptEnable register is set, a corresponding bit in the HcInteruptEnable register is set, and the MasterInterrupEnable bit is set. As a result, the USBHI bit in Interrupt Request Register 3 (IRR3) of Interrupt Controller INTC is set (the USBHI bit is used in common regardless of the content of the interrupt generation event). Therefore, the USBHI bit can be used when an interrupt generation is detected by HCD.

Writing 1 in this register sets the corresponding bit, while writing 0 leaves the bit. When read, the current value of this register is returned.

Register: *HcInterruptEnable* **Offset: 10–13**

Bits	Reset	R/W	Description
31	0b	R/W	**MasterInterruptEnable (MIE)** Setting of this bit to 0 is ignored by the host controller. When this bit is set to 1, an interrupt generation by the event specified in another bit in this register is enabled. This is used by HDC that the master interrupt is enabled. When an interrupt is detected by HCD, use the USBIH bit of Interrupt Controller INTC. 0: Ignore (initial value) 1: Enable interrupt generation due to the specified event.
30	0b	R/W	**OwnershipChangeEnable (OC)** 0: Ignore (initial value) 1: Enable interrupt generation due to Ownership Change.
29–7	0h	—	Reserved.
6	0b	R/W	**RootHubStatusChangeEnable (RHSC)** 0: Ignore (initial value) 1: Enable interrupt generation due to Root Hub Status Change.
5	0b	R/W	**FrameNumberOverflowEnable (FNO)** 0: Ignore (initial value) 1: Enable interrupt generation due to Frame Number Overflow.
4	0b	R/W	**UnrecoverableErrorEnable(UE)** 0: Ignore (initial value) 1: Enable interrupt generation due to unrecoverable error.
3	0b	R/W	**ResumeDetectedEnable (RD)** 0: Ignore (initial value) 1: Enable interrupt generation due to Resume Detected.
2	0b	R/W	**StartOfFrameEnable (SF)** 0: Ignore (initial value) 1: Enable interrupt generation due to Start of Frame.
1	0b	R/W	**WritebackDoneHeadEnable (WDH)** 0: Ignore (initial value) 1: Enable interrupt generation due to Writeback Done Head.
0	0b	R/W	**SchedulingOverrunEnable (SO)** 0: Ignore (initial value) 1: Enable interrupt generation due to Scheduling Overrun.

24.2.6 HcInterruptDisable

HcInterruptDisable Register (H'04000414)

Each disable bit in the HcInterruptDisable register corresponds to the related interrupt bit in the HcInterruptStatus register. The HcInterruptDisable register is related to the HcInterruptEnable register. Therefore, writing a 1 to a bit in this register clears the corresponding bit in HcInterruptEnable register, while writing a 0 to a bit leaves the corresponding bit in the HcInterruptEnable register. When read, the current value of the HcInterruptEnable register is returned.

Register: *HcInterruptDisable* **Offset: 14–17**

Bits	Reset	R/W	Description
31	0b	R/W	**MasterInterruptEnable (MIE)** 0: Ignore 1: Disable interrupt generation due to the specified event.
30	0b	R/W	**OwnershipChangeEnable (OC)** 0: Ignore 1: Disable interrupt generation due to Ownership Change.
29–7	0h	—	Reserved. Read/Write 0's
6	0b	R/W	**RootHubStatusChangeEnable (RHSC)** 0: Ignore 1: Disable interrupt generation due to Root Hub Status Change.
5	0b	R/W	**FrameNumberOverflowEnable (FNO)** 0: Ignore 1: Disable interrupt generation due to Frame Number Overflow.
4	0b	R/W	**UnrecoverableErrorEnable (UE)** 0: Ignore 1: Disable interrupt generation due to unrecoverable error.
3	0b	R/W	**ResumeDetectedEnable (RD)** 0: Ignore 1: Disable interrupt generation due to Resume Detected.
2	0b	R/W	**StartofFrameEnable (SF)** 0: Ignore 1: Disable interrupt generation due to Start of Frame.
1	0b	R/W	**WritebackDoneHeadEnable (WDH)** 0: Ignore 1: Disable interrupt generation due to Writeback Done Head.

Register: *HcInterruptDisable*			Offset: 14–17
Bits	**Reset**	**R/W**	**Description**
0	0b	R/W	**SchedulingOverrunEnable (SO)** 0: Ignore 1: Disable interrupt generation due to Scheduling Overrun.

24.2.7 HcHCCA

HCCA Register (H'04000418)

The HcHCCA register includes physical addresses of the host controller communication area. The host controller driver determines the alignment limitation by writing 1 to all bits in the HcHCCA register and by reading the content of the HcHCCA register. Alignment is evaluated by checking the number of 0 in the lower bits. The minimum alignment is 256 bytes. Consequently, bits 0 to 7 must be always returned to 0 when they are read. This area is used to retain the control structure and interrupt table that are accessed by the host controller and host controller driver.

Register: *HcHCCA*			Offset: 18–1B
Bits	**Reset**	**R/W**	**Description**
31–8	0h	R/W	HCCA
7–0	0h	—	Reserved.

24.2.8 HcPeriodCurrentED

HcPeriodCurrentED Register(H'0400041C)

The HcPeriodCurrentED register includes a physical address of current Isochronous ED or Interrupt ED.

Register: *HcPeriodCurrentED*			Offset: 1C–1F
Bits	**Reset**	**R/W**	**Description**
31–4	0h	R/W	**PeriodCurrentED (PCED)** Pointer to the current Periodic List ED. (Within memory area3)
3–0	0h	—	Reserved.

24.2.9 HcControlHeadED

HcControlHeadED (H'04000420)

The HcControlHeadED register includes a physical address of first ED.

Register: *HcControlHeadED* **Offset: 20–23**

Bits	Reset	R/W	Description
31–4	0h	R/W	**ControlHeadED (CHED)** Pointer to the Control List Head ED. (Within SRAM memory space)
3–0	0h	—	Reserved.

24.2.10 HcControlCurrentED

HcControlCurrentED Regsiter (H'04000424)

The HcControlCurrentED register includes a physical address of current ED.

Register: *HcControlCurrentED* **Offset: 24–27**

Bits	Reset	R/W	Description
31–4	0h	R/W	**ControlCurrentED (CCED)** Pointer to the current Control List ED. (Within memory area3)
3–0	0h	—	Reserved.

24.2.11 HcBulkHeadED

HcBulkHeadED Register (H'04000428)

The HcBulkHeadEDregister includes a physical address of first ED in Bulk List.

Register: *HcBulkHeadED* **Offset: 28–2B**

Bits	Reset	R/W	Description
31–4	0h	R/W	**BulkHeadED (BHED)** Pointer to the Bulk List Head ED. (Within memory area3)
3–0	0h	—	Reserved.

24.2.12 HcBulkCurrentED

HcBulkCurrentED Register(H'0400042C)

The HcBulkCurrentED register includes a physical address of current ED in the Bulk List. When the bulk list is supplied by the round robin method, endpoints are ordered to the list according to these insertions.

Register: *HcBulkCurrentED*			**Offset: 2C–2F**
Bits	**Reset**	**R/W**	**Description**
31–4	0h	R/W	**BulkCurrentED (BCED)** Pointer to the current Bulk List ED. (Within memory area3)
3–0	0h	—	Reserved.

24.2.13 HcDoneHead

HcDoneHead Register (H'04000430)

The HcDoneHead register includes a physical address of finally completed TD added to Done queue. The host controller needs not read this register so that the content is written to HCCA periodically in normal operation.

Register: *HcDoneHead*			**Offset: 30–33**
Bits	**Reset**	**R/W**	**Description**
31–4	0h	R/W	**DoneHead (DH)** Pointer to the current Done List Head ED. (Within memory area 3)
3–0	0h	—	Reserved.

24.2.14 HcFmInterval

HcFmInterval Register(H'04000434)

The HcFmInterval register includes a 14-bit value indicating the bit time interval of the frame (i.e., between two serial SOFs) and a 15-bit value indicating the maximum packet size at a full speed that is transmitted and received by the host controller without causing scheduling overrun. The host controller driver adjusts the frame interval minutely by writing a new value over the current value in each SOF. This supplies required programming ability to the host controller to synchronize with an external clock source and to synchronize with offset of an unknown local clock.

Register: *HcFmInterval* **Offset: 34–37**

Bits	Reset	R/W	Description
31	0b	R/W	**FrameIntervalToggle (FIT)** This bit is toggled by HCD whenever it loads a new value into **FrameInterval**.
30–16	0h	R/W	**FSLargestDataPacket (FSMPS)** This field specifies a value which is loaded into the Largest Data Packet Counter at the beginning of each frame. The counter value expresses the largest data amount of the bit that can be transmitted and received in one transaction by the host controller at any given time without causing scheduling overrun. The field value is calculated by HCD.
15–14	0h	—	Reserved.
13–0	2EDFh	R/W	**FrameInterval (FI)** These bits specifies the interval between two serial SOFs with bit times. The nominal value is set to 11999. HCD must store the current value of this field before resetting the host controller. With this procedure, this bit is reset to its nominal value by the host controller by setting the HostControllerReset bit in the HcCommandStatus register. HCD can select to restore the stored value at the completion of the reset sequence.

24.2.15 HcFmRemaining

HcFrameRemaining Register (H'04000438)

The HcFmRemaining register is a 14-bit down counter indicating the bit time remaining in the current frame.

Register: *HcFrameRemaining* **Offset**: **38–3B**

Bits	Reset	R/W	Description
31	0b	R	**FrameRemainingToggle (FRT)** This bit is always loaded from the FrameIntervalToggle bit in HcFminterval when FrameReamining reaches 0. This bit is used by HCD for the synchronization between FrameInterval and FrameReamining.
30–14	0h	—	Reserved.
13–0	0b	R	**FrameRemaining (FR)** This counter is decremented at each bit time. When this counter reaches 0, this counter is reset by loading the value of the FramInterval bit specified in the HcFminterval register at the next bit time boundary. When the host controller transits to the UsbOperational state, it read the FrameInterval bit in the HcFminterval register again and use the updated value from the next SOF.

24.2.16 HcFmNumber

HcFmNumberb Register (H'040004BC)

The HcFmNumber register is a 16-bit counter. It indicates the reference of timing between events occurring in the host controller and host controller driver. The host controller driver uses a 16-bit value specified in this register and generates a 32-bit frame number without necessity for a frequent access to the register.

Register: *HcFmNumber* **Offset: 3C–3F**

Bits	Reset	R/W	Description
31–16	0h	—	Reserved.
15–0	0b	R	**FrameNumber (FN)** These bits are incremented when HcFmRemaining is reloaded. The count will return to H'0 after H'FFFF. When the host controller transits to the UsbOperational state, these bits are automatically incremented. After the host controller increments the FramNumber bit and sends SOF in each frame boundary, the content is written to HCCA before the host controller reads first ED in the frame. After writing to HCCA, the host controller sets the StartofFrame bit in the HcInterruptStatus register.

24.2.17 HcPeriodicStart

HcPeriodicStart Register(H'04000440)

The HcPeriodicStart register has a 14-bit programmable value, which determines the earliest time when the host controller should start to process the periodic list.

Register: *HcPeriodicStart* **Offset: 40–43**

Bits	Reset	R/W	Description
31–14	0h	—	Reserved.
13–0	0b	R/W	**PeriodicStart (PS)** This field is cleared after the hardware has reset. Then this field is set by HCD while the host controller performs initial settings. The value is roughly calculated as the value of the HcFmInterval register minus 10%. When the HcFm Remaining register reaches the specified value, the processing of the periodic list has a higher priority than the control/bulk processing. Consequently, the host controller starts to process the interrupt list after the completion of the current control/bulk transaction.

24.2.18 HcLSThreshold

HcLSThreshold Register (H'04000444)

The HcLSIThreshold register includes an 11-bit value that is used by the host controller to determine whether or not to authorize the transfer of the LS packed 8 bytes in maximum before EOF. The host controller and host controller driver cannot change this value.

Register: *HcLSThreshold* **Offset: 44–47**

Bits	Reset	R/W	Description
31–12	0h	—	Reserved.
11–0	628h	R/W	**LSThreshold (LST)** This field contains a value to be compared with the FrameRemaining bit prior to the beginning of low-speed transaction. The transaction is started only when the Frame Remaining bit value is beyond the value of the list. The value is calculated by HCD considering the transmission and set-up overhead.

24.2.19 HcRhDescriptorA

HcRhDescriptorA Register (H'04000448)

The HcRhDescriptorA register is the first register of two registers describing the features of the root hub. The reset value is implementation specific. The descriptor length (11), descriptor type (TBD), the hub controller current bit (0) of Class Descriptor of the hub are emulated by HCD. All other bits are placed in the HcRhDescriptorA register and HcRhDescriptorB register.

Register: *HcRhDescriptorA* **Offset: 48–4B**

Bits	Reset	R/W	Description
31–24	02h	R/W	**PowerOnToPowerGoodTime (POTPGT)** These bits specify the time required for waiting before accessing the power-on port of the root hub. These bits are implementation specific. The unit of time is 2 ms. The time is calculated as POTPGT × 2 ms.
23–13	0h	—	Reserved.
12	1	R/W	**NoOverCurrentProtection (NOCP)** This bit selects how the over-current status of the root hub is reported. When this bit is cleared, the OverCureentProtectionMode bit specifies global report or report at each port. 0: Over-current status is collectively reported for all downstream ports. 1: Over-current protection is not supported. (initial value)
11	0	R/W	**OverCurrentProtectionMode (OCPM)** This bit selects how the over-current status in the root-hub port is reported. At reset, this bit reflects the same mode of PowerSwitchingMode. When the NoOverCureentProtection bit is cleared, this bit is valid. 0: Over-current status is collectively reported for all downstream ports. (initial value) 1: Over-current protection is not supported.
10	0	R	**DeviceType (DT)** USB Host Controller is not a compound device. Always set to 0.

Register: *HcRhDescriptorA* **Offset: 48–4B**

Bits	Reset	R/W	Description
9	1	R/W	**NoPowerSwitching (NPS)** This bit selects whether the power switching is supported or ports are always power-supplied. This bit is implementation specific. When this bit is cleared, Power-SwitchingMode specifies the global/port switching. 0: Ports can be power-switched. 1: Ports are always powered on when the host controller is powered on. (initial value) Note: Because the initial value is 1, first clear this bit (write 0 with the HCD) to enable power switching of the port.
8	0	R/W	**PowerSwitchingMode (PSM)** This bit specifies how the power switching of the root-hub port is controlled. This bit is implementation specific. This bit is valid only when the NoPowerSwitching bit is cleared. 0: All ports are simultaneously power-supplied. (initial value) 1: Each port is power-supplied individually. In this mode, the port power is controlled with either of global/port switching. When the PortPowerControlMask bit is set, the port is reacted only to the port-power command (set/clear port power). When the port mask is cleared, the port is controlled only by the global power-switch (set/clear global power).
7–0	02h	R	**NumberDownstreamPorts (NDP)** These bits specify the number of downstream ports supported by the root hub. These bits are implementation specific. In the SH7727 their value is H'2.

24.2.20 HcRhDescriptorB

HcRhDescriptorB Register (H'0400044C)

The HcRhDescriptorB register is the second register of two registers describing the features of the root hub. These bits are written during the initial setting so as to correspond to the system implementation. The reset value is implementation specific.

Register: *HcRhDescriptorB* **Offset: 4C–4F**

Bits	Reset	R/W	Description
31–16	0h	R/W	**PortPowerControlMask (PPCM)** This bit indicates that the port is influenced by the global power-control command when the PowerSwitchingMode bit is set. When this bit is set, the power state of the port is affected by the power control at each port (set/clear port power). When this bit is cleared, the port is controlled by the global power switch (set/clear global power). If the device is placed in the global switching mode (PowerSwitchingMode = 0), this bit is not valid. Bit 15: Assured Bit 16: Port#1 power mask Bit 17: Port#2 power mask … Bit 31: Port#15 power mask Note: Clear the No Power Switching of the RhDescriptorA register so that the power to all ports is OFF (Port Power Status = 0), then set this bit.
15–0	0h	R/W	**DeviceRemoveable (DR)** These bits are dedicated to the ports of the root hub. When these bits are cleared, the set device becomes removable. When these bits are set, do not remove the set device. Bit 0: Assured Bit 1: Device affixed to Port#1 Bit 2: Device affixed to Port#2 … Bit 15: Device affixed to Port#15

24.2.21 HcRhStatus

HcRhStatus Register (H'04000450)

The HcRhStatus register is divided into two parts. The lower word of a long word indicates the hub status bits and the upper word indicates the hub status change bit. Reserved bits should be set to 0.

Register: *HcRhStatus* **Offset: 50–53**

Bits	Reset	R/W	Description
31	0	W	**ClearRemoteWakeupEnable (CRWE)** Writing a 1 to this bit clears **DeviceRemoteWakeupEnable**. Writing a 0 has no effect.
30–18	0h	—	Reserved. Read/Write 0's
17	0	R/W	**OverCurrentIndicatorChange (OCIC)** This bit is set when **OverCurrentIndicator** changes. Writing a 1 clears this bit. Writing a 0 has no effect.
16	0	R/W	(read) **LocalPowerStatusChange (LPSC)** The root hub does not support the local power status function. Therefore, this bit is always read 0. (write) **SetGlobalPower** This bit is written to 1 to power on (clears the PortPowerStatus bit) all ports in global power mode (PowerSwitchingMode = 0). This bit sets the PortPowerStatus bit only to the port in which the PortPowerControlMask bit is not set in power mode at each port. When 0 is written to, this bit is not cleared.
15	0	R/W	(read) **DeviceRemoteWakeupEnable (DRWE)** This bit enables the ConnectStatusChange bit as a resume event and generates the state transition from UsbSuspend to UsbResume and ResumeDetected interrupt. 0: ConnectStatusChange is not the remote wakeup event (initial value) 1: ConnectStatusChange is the remote wakeup event. (write) **SetRemoteWakeupEnable** Writing a 1 sets **DeviceRemoteWakeupEnable**. Writing a 0 has no effect.
14–2	0h	—	Reserved.

Register: *HcRhStatus*			Offset: 50–53
Bits	**Reset**	**R/W**	**Description**
1	0	R	**OverCurrentIndicator (OCI)** This bit reports the over-current condition. When this bit is set, an over-current condition exists. When this bit is cleared, all power operations are normal. This bit is always 0 when the over-current protection at each port is carried out. 0: All power operations are normal. (initial value) 1: An over-current condition exists.
0	0	R/W	(read) **LocalPowerStatus (LPS)** The root hub does not support the local power status function. Therefore, the bit is always read 0. (write) **ClearGlobalPower** This bit is written to 1 to power on (leaves the PortPowerStatus bit) all ports in global power mode (Power Switching Mode = 0). This bit clears the PortPowerStatus bit only to the port in which the PortPowerControlMask bit is not set. Writing a "0" has no effect. In the power mode at each port, the PortpowerStatus bit is cleared to the port in which the PortPowerControlmask bit is not set. Writing 0, has no effect.

24.2.22 HcRhPortStatus[1:2]

HcRhPortStatus Register ([1]:04000454 [2]:04000458)

This register is reset by the USBRESET state.

HcRhPortStatus 1, 2 registers are used for base-controlling each port and to report the port event. The lower word is used to reflect the port status while the upper word reflects the status change. Some status bits have special writing (see below). If an attempt to write to a bit indicating a change in port status occurs when a transaction in which a token is passed via a handshake is in progress, the writing to the bit is delayed until the transaction is completed. Always write reserved bits to 0.

Register: *HcRhPortStatus[1:2]* **Offset: 54–57, 58–5B**

Bits	Reset	R/W	Description
31–21	0h	—	Reserved.
20	0	R/W	**PortResetStatusChange (PRSC)** This bit is set when the 10 ms port reset signal has completed. Writing a 1 clears this bit writing a 0 has no effect. 0 = Port reset is not complete. 1 = Port reset is complete.
19	0	R/W	**PortOverCurrentIndicatorChange (OCIC)** This bit is valid when an over-current condition is reported on the base of each port. This bit is set when the root hub changes the PortOverCurrentIndicator bit. Writing a 1 clears this bit. Writing a 0 has no effect. 0: PortOverCurrentIndicator has not changed. (initial value) 1: PortoverCurrentIndicator has changed.
18	0	R/W	**PortSuspendStatusChange (PSSC)** This bit is set when all resume sequences have completed. These sequences include 20 ms resume pulse, LS EOP, and 3 ms resychronization delay. Writing a 1 clears this bit. Writing a 0 has no effect. This bit is cleared also when ResetStatusChange is set. 0: Port resume has not completed. (initial value) 1: Port resume has completed.
17	0	R/W	**PortEnableStatusChange (PESC)** This bit is set when the PortEnableStatus bit is cleared due to a hardware event . This bit is not set by the change of writing of HCD. Writing a 1 clears this bit. Writing a 0 has no effect. 0: PortEnableStatus has not changed (initial value) 1: PortEnableStatus has changed

Register: *HcRhPortStatus[1:2]* **Offset: 54–57, 58–5B**

Bits	Reset	R/W	Description
16	0	R/W	**ConnectStatusChange (CSC)** This bit is set whenever the connection or disconnection event occurs. Writing a 1 clears this bit. Writing a 0 has no effect. If CurrentConnectStatus is cleared when SetPortReset, SetPortEnable, or SetPortSuspend is written to, writing when the power supply of the port is disconnected does not occur, so this bit is set to enforce the driver to re-evaluate the connection status. 0: CurrentConnectionStatus has not changed (initial value) 1: CurrentConnectionStatus has changed Note: If the Device Removable bit is set, this bit is set only after the root hub reset to inform that the system that a device can be attached.
15–10	0h	—	Reserved.
9	0	R/W	(read) **LowSpeedDeviceAttached (LSDA)** This bit indicates the speed of the device attached to this port. When this bit is set, a low-seed device is attached to this port. When this bit is cleared, a full-speed device is attached to this port. This bit is valid only when the CurrentConnectStatus bit is set. 0: A full-speed device is set. (initial value) 1: A low-speed device is set. (write) **ClearPortPower** Writing a 1 clears **PortPowerStatus**. Writing a 0 has no effect

Register: *HcRhPortStatus[1:2]* **Offset: 54–57, 58–5B**

Bits	Reset	R/W	Description
8	1	R/W	(read) **PortPowerStatus (PPS)** This bit reflects the power state of the port regardless of the power switching mode to be executed. However, because the initial value of the No Power Switching bit of the HcRhDescriptorA register is 1, this bit is first fixed to 1. No Power Switching bit must first be cleared before the power is switched, as shown below. When an over-current condition is detected, this bit is cleared. Writing SetPortPower or SetGlovalPower sets this bit. Writing ClearPortPower or ClearGlobalPower clears this bit. PowerSwitchingMode and PortPowerControlMask determine which power control switch can be used. Only Set/ClearGlobalPower controls this bit in global switching mode (PowerSwitchingMode= 0). If the PortPowerControlMask bit of that port is set in power switching mode (PowerSwitchingMode = 1). Only the Set/ClearGlobalPower command is enabled if the mask is not set. When the port power is disabled, CurrentConnectStatus, PortEnableStatus, PortSuspendStatus, and portResetStatus are reset. 0 = Port power is off. 1 = Port power is on. (initial value) Note: If **NoPowerSwitching** is set, this bit is always read as 1. (write) **SetPortPower** Writing a 1 sets **PortPowerStatus**. Writing a 0 has no effect.
7–5	0h	—	Reserved.
4	0	R/W	(read) **PortResetStatus (PRS)** When this bit is set by writing to SetPortReset, the port reset signal is output. This bit is cleared when PortResetStatusChange is set upon completion of a reset. When CurrentConnectStatus is cleared, this bit is not set. 0 = Port reset signal is not active. (initial value) 1 = Port reset signal is active. (write) **SetPortReset** Writing a 1 sets PortReset signal. Writing a 0 has no effect. When CurrentConnectStatus is cleared, this write does not set PortResetStatus, instead, sets ConnectStatusChange. This reports a reset of the power disconnection port to the driver.

Register: *HcRhPortStatus[1:2]* **Offset: 54–57, 58–5B**

Bits	Reset	R/W	Description
3	0	R/W	(read) **PortOverCurrentIndicator (POCI)** This bit is valid only when a root hub is placed in such a way that an over-current condition is reported on the base of each port. If the over-current report at each port is not supported, this bit is set to 0. If this bit is cleared, all power controls are normal in this port. If this bit is set, an over-current status exists in this port. This bit always reflects an over-current input signal. 0 = No over-current condition (initial value) 1 = Over-current condition is detected (write) **ClearSuspendStatus** Writing a 1 initiates a resume. Writing a 0 has no effect. If PortSuspendStatus is set, a resume is initiated.
2	0	R/W	(read) **PortSuspendStatus (PSS)** This bit indicates that the port is suspended or during the resume sequence. Writing SetSuspendState sets this bit and setting PortSuspendStatusChange clears this bit at the end of the resume interval. If CurrentConnectStatus is cleared, this bit cannot be set. When portResetStatusChange is set upon completion of the port reset or HC is placed in the UsbResume state, this bit is cleared. If an upstream resume is in progress, it is transmitted to the host controller. 0 = Port is not suspended 1 = Port is selectively suspended (write) **SetPortSuspend** Writing a 1 sets **PortSuspendStatus**. Writing a 0 has no effect. In addition, when CurrentConnectStatus is cleared, PortSuspendStatus is not set by this writing. Instead, ConnectStatusChange is set. This reports the suspended state of the power disconnection to the driver.

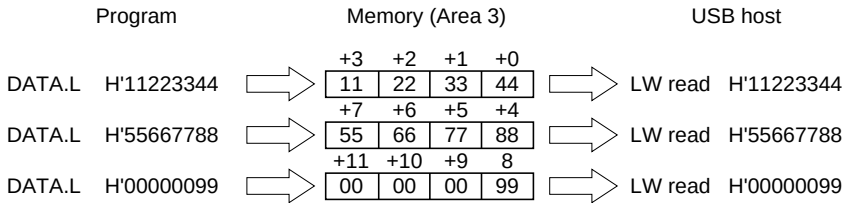
Register: *HcRhPortStatus[1:2]* **Offset: 54–57, 58–5B**

Bits	Reset	R/W	Description
1	0	R/W	(read) **PortEnableStatus (PES)** This bit indicates whether the port is enabled or disabled. The root hub clears this bit when the over-current condition and a operational bus error such as disconnect event, power-off switch, or babble is detected. PortEnabledStatusChange is set by this change. This bit is set by writing SetPortEnable and cleared by writing ClearPortEnable. This bit cannot be set when CurrentConnectStatus is cleared. In addition, this bit is set upon completion of the port reset by which ResetStatusChange is set, or uponcompletion of the port suspend by which SuspendStatusChange is set. 0 = Port disabled. (initial value) 1 = Port enabled. (write) **SetPortEnable** Writing a 1 sets **PortEnableStatus**. Writing a 0 has no effect. If CurrentConnectStatus is cleared, this writing does not set PortEnableStatus, instead, sets ConnectStatusChange. This reports the driver that the power disconnection port has been tried to be enabled.
0	0	R/W	(read) **CurrentConnectStatus (CCS)** This bit indicates the status of the downstream port. 0 = No device connected. 1 = Device connected. Note: If **DeviceRemoveable** is set (not removable) this bit is always 1. (write) **ClearPortEnable** Writing a 1 clears **PortEnableStatus**. Writing a 0 has no effect. CurrentConnectStatus is not affected by any writing.

24.3 Data Storage Format which Required by USB Host Controller

24.3.1 Storage Format of the Transferred Data

USB Host Controller expects that data are compiled from lower address to upper address regardless endian setting of the CPU. Below figure shows data read operation which is done by USB Host Controller.



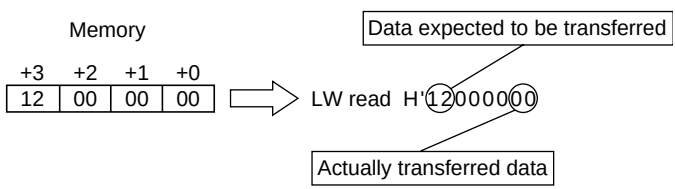
The correspondence between data in memory and data read by USB Host Controller must be equal. When USB Host Controller reads data from external memory, USB Host Controller reads data by long word read operation every time regardless that the read data are written in byte ,word or long word. USB Host Controller uses data in byte from lower address in long word which it reads regardless the endian mode. Even endian mode is set as big or little, set the data from down addresses.

Below program flow is the example of failure.

1st In program , set transfer address A to register R0 at big endian

2nd In program , "MOV.B #H'12,@R0"

3rd In program , set transfer start address A to USB Host Controller , and set 1byte as transfer size.



This example show that above operation transfers expected data #H'12.

Data is filled from the lower bits of the memory in writing so that the data is read/written in bi-direction consistently regardless of the endian type. That is, the data is always aligned with the little endian specification.

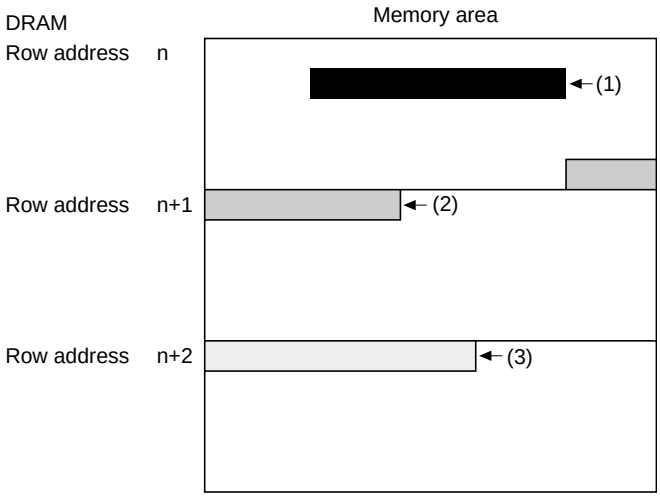
24.3.2 Storage Format of the Descriptor

ED (endpoint descriptor) and TD (transfer descriptor) that define each transfer transaction of USB Host Controller must be aligned so that each Dword corresponds to the long-word boundary (addresses 4n to 4n + 3) of the memory.

24.4 Data Alignment Restriction of USB Host Controller

24.4.1 Restriction on the Line Boundary of the Synchronous DRAM

The transferred data is stored in shared system memory with CPU. The data alignment in system memory are restricted depends on synchronous DRAM specification which is used as system memory.



In above figure , transfer data 1 and 3 are able to be read or written by USB Host Controller. But transfer data 2 are possibly unable to be read or written by USB Host controller. Any data which have possibility to be accessed by USB Host Controller must be aligned in synchronous DRAM not to cross row address alignment.

24.4.2 Restriction on the Memory Access Address

MPS in ED, CBP in General TD, and BP0 and OFFSET0 to 7 in Ischoronous TD must be set in multiples of 4 (4n). In the OpenHCI standard, 1 packet is transferred by ITD in General TD and 1 packet by 1 offset in Ischronous TD. In addition, when the amount of the data specified by TD during OUT transfer exceeds MAXPACKETSIZE (MPS), a packet transmission is carried out in MAXPACKETSIZE. Therefore, the setting value can be made as above. This restriction is due to the difference between the specifications of the HCI interface which is the standard of the IP bus interface of USB and of the bus interface of SH7727. Data might be correctly written to if data is transferred from addresses other than 4n address. For example, when a two-byte transfer is carried out from the address that terminates at 1, a long-word transfer is carried out and an unexpected data is written to starting address 0.

24.5 Restrictions on the Data Transfer of USB Controller

24.5.1 Restriction of the Data Size in IN Transfer

When a data packet shorter than MAXPACKETSIZE (short packet) is transferred in the IN data transfer other than the isochronous transfer, following usages are restricted.

1. Usage when a dribble bit is added
   When HUB are connected in multiple steps, a dribble bit may be added at the end of the packet.
2. When receiving the data with final 6 bits in CRC are all 1 (in this case, bit stuffing occurs)
   In this case, this USB controller may write IN data in the memory with one byte additionally. Therefore, usage of 1 is prohibited. In usage of 2, the software must be written to so that no problem occurs even additional data of 1 byte is written to. In concrete, in the usage to connect the received short packets are connected in the memory, 1 byte of unnecessary data might be inserted. Be sure to transfer MAXPACKETSIZE mainly and the processed data will be used so that the end of the data or head can be recognized when a short packet is sent.

24.5.2 Restrictions on the Hub Connection on NAK/STALL Reception

When NAK or STALL is received as a handshake from the USB function module, the following usage is restricted.

1. Usage where a dribble bit is added
   When HUBs are connected in multiple steps, a dribble bit may be added at the end of the packet. In this case, this USB controller might not correctly recognize the NAK/STALL handshake. Be sure not to connect HUB to decrease steps so that no dribble bit occurs.

24.5.3 Restrictions when a Low-Speed Device is Disconnected

When a low-speed device connected to the root port is disconnected while the data is transmitted to the host, this USB controller might hung. Therefore, be sure not to disconnect the low-speed device while the data is transmitted. If that disconnection is detected, reset USB (write 00 to the HCFS bit in the HcControl register).

24.6 Restrictions on the Software Reset and USB Reset

The operation of the controller might become stuck if a software reset (1 written to the HCR bit in the HcCommandStatus register) or USB reset (00 written to the HCFS bit in the HcControl register) is performed while the USB host controller is carrying out a master memory write. Master memory write operations are performed at the start of frame (SOF) and during data transfer. To prevent the host controller from becoming stuck, use one of the following methods:

1. Write the program so that the SDF timing is derived from the HcFmRemaining value and a software reset or USB reset is applied around the middle of a frame in which all list processing is disabled.
2. If you issue a reset for the USBH module after applying a software reset or USB reset, no stack will occur. (Set the module software reset register (SRSTR).)

24.7 Notes on Using USB Host with Versions Previous to the SH7727C

The USB host controller of the SH7727 and SH7727B may erroneously recognize an SE0 state during data transfer. As a result, the controller incorrectly detects a device not connected state (erroneous device not connected detection) even though the device actually is connected. The following methods may be used to work around this problem.

1. Use the low speed (1.5 Mbps) mode. Erroneous device not connected detection does not occur during low speed data transfer.
2. When using the full speed (12 Mbps) mode, use method (a) or (b) below to avoid erroneous device not connected detection.
   a. Ensure that the D+, D– crossover voltage output by the USB function device exceeds SH7727 AVcc_USB × 0.55 (V).
   b. Ensure that received data strings do not contain more than 28 consecutive 0 bits.

      Note: The prohibition on consecutive 0 bits applies to all bits of SYNC + PID + DATA + CRC16 + EOP. EOP is counted as 2 bits.

Explanation of Terms

- SE0: USB data transfer uses a D+/D– differential signal. Consequently, D+/D– is normally a reciprocal signal. However, in some special cases D+ and D– are both defined as low. This state is referred to as SE0.
- Device not connected detection: Detection of a USB device not connected state when SE0 continues beyond the device not connected detection duration.

Section 25 LCD Controller

25.1 Overview

A unified memory architecture is adopted for the LCD controller (LCDC) so that the image data for display is stored in system memory. The LCDC module reads data from system memory, uses the pallet memory to determine the colors, then puts the display on the LCD panel. It is possible to connect the LCDC to the LCD module of most types other than microcomputer bus interface types and NTSC/PAL types and those that apply the LVDS interface*.

Note: * An LVDS-conversion LSI can be connected to the LCDC to allow connection to an LVDS interface.

25.1.1 Features

The LCDC has the following features.

- Panel interface
  - Serial interface method
  - Supports data formats for STN/dual-STN/TFT panels (8/12/16/18-bit bus width)
- Supports 4/8/15/16-bpp (bit per pixel) color modes
- Supports 1/2/4/6-bpp grayscale modes
- Supports LCD-panel sizes from 16 × 1 to 1024 × 1024
- 24-bit color palette memory (16 of the 24 bits are valid; R:5/G:6/B:5)
- STN/DSTN panels are prone to flicker and shadowing. The controller applies 65536-color control by 24-bit space-modulation FRC with 8-bit RGB values for reduced flicker.
- Dedicated display memory is unnecessary because the controller uses synchronous DRAM which is connected to area 3 of the CPU's memory map
- The display is stable because of the large 2.4-kbyte line buffer
- Supports the inversion of the output signal to suit the LCD panel's signal polarity
- Supports variation of the burst length in reading from the synchronous DRAM, to realize high-speeds in the reading of data
- Supports the selection of data formats (the endian setting for bytes, backed pixel method) by register settings
- A hardware-rotation mode is included to support the use of landscape-format LCD panels as portrait-format LCD panels (the horizontal width of the panel before rotation must be within 320 pixels—see table 25.3).

Note: When connecting the LCDC to a TFT panel with an unwired 18-bit bus, the lower-order bit lines should be connected to GND or to the lowest bit from which data is output.

25.1.2 Block Diagram

Figure 25.1 shows a block diagram of LCDC.

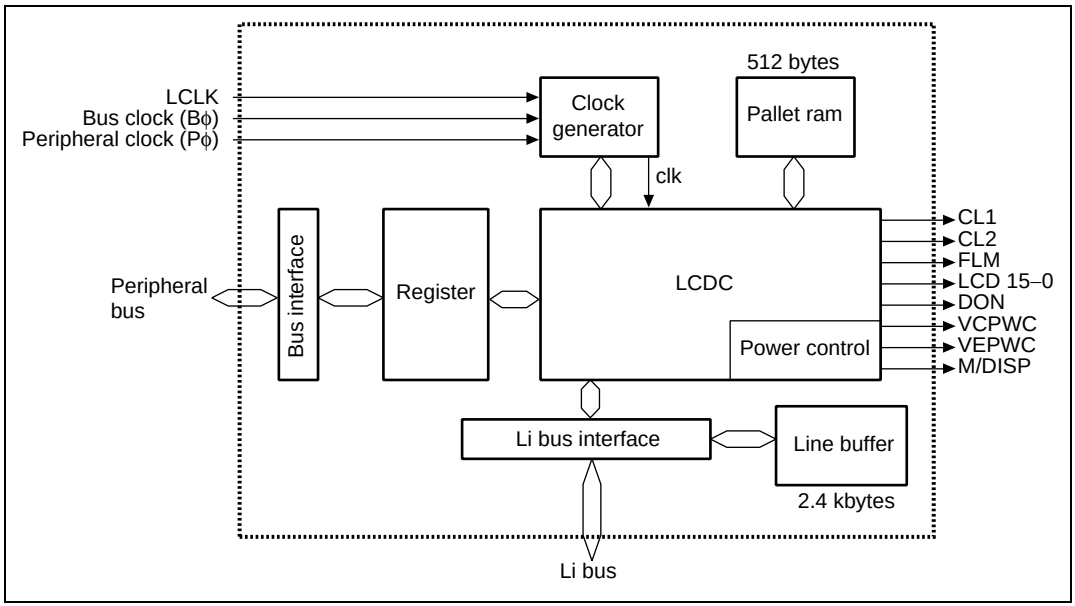


Figure 25.1 Block Diagram

25.1.3 Pin Configuration

Table 25.1 summarizes the LCDC's pin configuration.

Table 25.1 Pin Configuration

Name	I/O	Function
LCD 15–0	O	Data for LCD panel
DON	O	Display-on signal (DON)
CL1	O	Shift-clock 1 (STN/DSTN)/Horizontal sync signal (HSYNC) (TFT)
CL2	O	Shift-clock 2 (STN/DSTN)/dot clock (DOTCLOCK) (TFT)
M/DISP	O	LCD current-alternation signal/(STN/DSTN), Display enable $\overline{\text{BLANK}}$ (TFT)/DISP signal
FLM	O	First line marker/Vertical sync signal (VSYNC) (TFT)
VCPWC	O	LCD-module power control (Vcc)
VEPWC	O	LCD-module power control (V_{EE})
LCLK	I	LCD clock-source input

Note: Check the LCD module specifications carefully in section 25.4, Clock and LCD Data Signal Examples, before deciding on the wiring specifications for the LCD module.

25.1.4 Register Configuration

Table 25.2 summarizes the configuration of the LCDC's registers.

Table 25.2 Register Configuration

Register Name	Abbreviation	Initial Value	Address	Access Size
LCDC input clock register	LDICKR	H'0101	H'04000C00 (H'A4000C00)*	16
LCDC module type register	LDMTR	H'0109	H'04000C02 (H'A4000C02)*	16
LCDC data format register	LDDFR	H'000C	H'04000C04 (H'A4000C04)*	16
LCDC scan mode register	LDSMR	H'0000	H'04000C06 (H'A4000C06)*	16
LCDC data fetch start address register for upper portion of display panel	LDSARU	H'0C000000	H'04000C08 (H'A4000C08)*	32
LCDC data fetch start address register for lower portion of display panel	LDSARL	H'0C000000	H'04000C0C (H'A4000C0C)*	32

Register Name	Abbreviation	Initial Value	Address	Access Size
LCDC fetch data line address offset register for display panel	LDLAOR	H'0280	H'04000C10 (H'A4000C10)*	16
LCDC palette control register	LDPALCR	H'0000	H'04000C12 (H'A4000C12)*	16
LCDC palette data registers 00 to FF	LDPR00–FF	Undefined	H'04000800 to H'04000BFC (H'A4000800 to H'A4000BFC)*	32
LCDC horizontal character number register	LDHCNR	H'4F52	H'04000C14 (H'A4000C14)*	16
LCDC horizontal synchronization signal register	LDHSYNR	H'0050	H'04000C16 (H'A000C16)*	16
LCDC vertical displayed line number register	LDVDLNR	H'01DF	H'04000C18 (H'A4000C18)*	16
LCDC vertical total line number register	LDVTLNR	H'01DF	H'04000C1A (H'A4000C1A)*	16
LCDC vertical synchronization signal register	LDVSYNR	H'01DF	H'04000C1C (H'A4000C1C)*	16
LCDC ac modulation signal toggle line number register	LDACLNR	H'000C	H'04000C1E (H'A4000C1E)*	16
LCDC interrupt control register	LDINTR	H'0000	H'04000C20 (H'A4000C20)*	16
LCDC power management mode register	LDPMMR	H'0010	H'04000C24 (H'A4000C24)*	16
LCDC power supply sequence period register	LDPSPR	H'F60F	H'04000C26 (H'A4000C26)*	16
LCDC control register	LDCNTR	H'0000	H'04000C28 (H'A4000C28)*	16

Note: * When the LCDC is not a target for address conversion by the MMU, use the addresses in parentheses.

25.2 Register Descriptions

25.2.1 LCDC Input Clock Register (LDICKR)

This LCDC can select the bus clock (Bϕ), the peripheral clock (Pϕ), or the external clock as its operation clock source. The selected clock source can be divided using an internal divider into a clock of 1/1 to 1/16 and be used as the LCDC operating clock (DOTCLOCK). The clock output from the LCDC is used to generate the synchronous clock output (CL2) for the LCD panel from the operating clock selected in this register. The average frequency of CL2 can be calculated using the formula below. The actual frequency, however, will differ depending on the type of LCD panel and the bus width of the data output to the LCD panel. See section 25.4, Clock and LCD Data Signal Examples, for details.

TFT panel
CL2 = DOTCLOCK

STN or DSTN panel
Monochrome: CL2 = (DOTCLOCK/data bus width of output to LCD panel)
Color: CL2 = 3 × (DOTCLOCK/data bus width of output to LCD panel)

Set this register so that the clock input to the LCDC is 50 MHz or less regardless of CL2.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	ICKSEL1	ICKSEL0	—	—	—	—	—	—	—	DCDR4	DCDR3	DCDR2	DCDR1	DCDR0
Initial value:	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1
R/W:	R	R	R/W	R/W	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W

Bits 15, 14, and 11 to 5—Reserved

Bits 13 and 12—Input Clock Select (ICKSEL1 and ICKSEL0): Set the clock source for DOTCLOCK.

Bit 13 ICKSEL1	Bit 12 ICKSEL0	Description	
0	0	Bus clock (Bϕ) is selected	(Initial value)
	1	Peripheral clock (Pϕ) is selected	
1	0	External clock (LCLK) is selected	
	1	Reserved (setting prohibited)	

Bits 4 to 0—Denominator of Clock Division Ratio (DCDR4 to DCDR0): Set denominator of the input clock division ratio.

		I/O Clock Frequency (MHz)	
DCDR[4:0]	**Clock Division Ratio**	**50.000**	
00001	1/1	50.000	(Initial value)
00010	1/2	25.000	
00100	1/4	12.500	
01000	1/8	6.250	
10000	1/16	3.125	

Any setting other than above is handled as a clock division ratio of 1/1 (initial value).

Note: The access size indicates the size the CPU uses to access (read from or write to) the register. When accessing this register in a size other than the displayed one, LCDC operation is not guaranteed.

Only 0 can be written to a reserved bit.

When a setting not allowed is made, e.g. a reserved bit is written to, though the LCDC operates with its initial values, normal operation is not guaranteed.

This is the common rule to all registers in this LCDC.

25.2.2 LCDC Module Type Register (LDMTR)

LDMTR sets the control signals output from this LCDC and the polarity of the data signals, according to the polarity of the signals for the LCD module connected to the LCDC.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FLM POL	CL1 POL	DISP POL	DPOL	—	MCNT	CL1 CNT	CL2 CNT	—	—	MIF TYP5	MIF TYP4	MIF TYP3	MIF TYP2	MIF TYP1	MIF TYP0
Initial value:	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	1
R/W:	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bits 11, 7, and 6—Reserved

Bit 15—FLM (Vertical Sync Signal) Polarity Select (FLMPOL): Selects the polarity of the FLM (vertical sync signal, first line marker) for the LCD module.

Bit 15 FLMPOL	**Description**	
0	FLM pulse is high active	(Initial value)
1	FLM pulse is low active	

Bit 14—CL1 (Horizontal Sync Signal) Polarity Select (CL1POL): Selects the polarity of the CL1 (horizontal sync signal) for the LCD module.

Bit 14 CL1POL	Description	
0	CL1 pulse is high active	(Initial value)
1	CL1 pulse is low active	

Bit 13—DISP (Display Enable) Polarity Select (DISPPOL): Selects the polarity of the DISP (display enable) for the LCD module. Valid for TFT panels only.

Bit 13 DISPPOL	Description	
0	DISP is high active	(Initial value)
1	DISP is low active	

Bit 12—Display Data Polarity Select (DPOL): Selects the polarity of the LCDD (display data) for the LCD module. This bit supports inversion of the LCD module.

Bit 12 DPOL	Description	
0	LCDD is high active, transparent-type LCD panel	(Initial value)
1	LCDD is low active, can be used for reflective-type LCD panels	

Bit 10—M Signal Control (MCNT): Sets whether or not to output the LCD's current-alternating signal of the LCD module. Valid for STN/DSTN.

Bit 10 MCNT	Description	
0	M (AC line modulation) signal is output	(Initial value)
1	M signal is fixed low	

Bit 9—CL1 (Horizontal Sync Signal) Control (CL1CNT): Sets whether or not to enable CL1 output during the vertical retrace period.

Bit 9 CL1CNT	Description	
0	CL1 is output during vertical retrace period	(Initial value)
1	CL1 is inactive during vertical retrace period	

Bit 8—CL2 (Data Latch Clock of LCD Module) Control (CL2CNT): Sets whether or not to enable CL2 output during the vertical retrace period.

Bit 8 CL2CNT	Description	
0	CL2 is output during vertical retrace period	(Initial value)
1	CL2 is inactive during vertical and horizontal retrace periods	

Bits 5 to 0—Module Interface Type Select (MIFTYP5 to MIFTYP0): Set the LCD panel type and data bus width for output to the LCD panel. There are three LCD panel types: STN, DSTN, and TFT. There are four data bus widths for output to the LCD panel: 4, 8, 12, and 16 bits. When the required data bus width for a TFT panel is 16 bits or less, connect the LCDC and LCD panel according to the data bus size of the LCD panel. Unlike in a TFT panel, in an STN or DSTN panel, the data bus width setting does not have a 1:1 correspondence with the number of display colors and display resolution, e.g., an 8-bit data bus can be used for 16 bpp, and a 12-bit data bus can be used for 4 bpp. This is because the number of display colors in an STN or DSTN panel is determined by how data is placed on the bus, and not by the bus width. For data specifications for an STN or DSTN panel, see the specifications of the LCD panel used. The output data bus width should be set according to the mechanical interface specifications of the LCD panel.

If an STN or DSTN panel is selected, display control is performed using a 24-bit space-modulation FRC (Frame Rate Controller) consisting of the 8-bit R, G, and B included in the LCDC, regardless of the color and gradation settings. Accordingly, the color and gradation specified by DSPCOLOR is selected from 16 million colors in an STN or DSTN panel. If a palette is used, the color specified in the palette is displayed.

Bit 5 MIFTYP5	Bit 4 MIFTYP4	Bit 3 MIFTYP3	Bit 2 MIFTYP2	Bit 1 MIFTYP1	Bit 0 MIFTYP0	Description
0	0	0	0	0	0	STN monochrome 4-bit data bus module
					1	STN monochrome 8-bit data bus module
		1	0	0	0	STN color 4-bit data bus module
					1	STN color 8-bit data bus module (Initial value)
				1	0	STN color 12-bit data bus module
					1	STN color 16-bit data bus module
	1	0	0	0	1	DSTN monochrome 8-bit data bus module
				1	1	DSTN monochrome 16-bit data bus module
		1	0	0	1	DSTN color 8-bit data bus module
				1	0	DSTN color 12-bit data bus module
					1	DSTN color 16-bit data bus module
1	0	1	0	1	1	TFT color 16-bit data bus module
Other than above						Reserved

25.2.3 LCDC Data Format Register (LDDFR)

LDDFR sets the bit alignment for pixel data in one byte and selects the data type and number of colors used for display so as to match the display driver software specifications.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	PABD	—	DSP COLOR 6	DSP COLOR 5	DSP COLOR 4	DSP COLOR 3	DSP COLOR 2	DSP COLOR 1	DSP COLOR 0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0
R/W:	R	R	R	R	R	R	R	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bits 15 to 9 and 7—Reserved

Bit 8—Byte Data Pixel Alignment (PABD): Sets the pixel data alignment type in one byte of data. The contents of aligned data per pixel are the same regardless of this bit's setting. For example, data H'05 should be expressed as 0101 (binary) which is the normal style handled by a MOV instruction of the SH7727 CPU, and should not be selected between 0101 (binary) and 1010 (binary).

Bit 8 PABD	Description	
0	Big endian for byte data	(Initial value)
1	Little endian for byte data	

Bits 6 to 0—Display Color Select (DSPCOLOR6 to DSPCOLOR0): Set the number of display colors for the display (0 is written to upper bits of for unpacked 4, 5, and 6 bpp). For display colors to which the description (via palette) is added below, the color set by the color palette is actually selected by the display data and displayed.

Bit 6 DSP COLOR6	Bit 5 DSP COLOR5	Bit 4 DSP COLOR4	Bit 3 DSP COLOR3	Bit 2 DSP COLOR2	Bit 1 DSP COLOR1	Bit 0 DSP COLOR0	Description
0	0	0	0	0	0	0	Monochrome, 2 grayscales, 1 bpp (via palette)
						1	Monochrome, 4 grayscales, 2 bpp (via palette)
				0	1	0	Monochrome, 16 grayscales, 4 bpp (via palette)
				1	0	0	Monochrome, 64 grayscales, 6 bpp (via palette)
			1	0	1	0	Color, 16 colors, 4 bpp (via palette)
				1	0	0	Color, 256 colors, 8 bpp (via palette)
		1	1	1	0	1	Color, 32k colors (RGB: 555), 15 bpp
	1	0	1	1	0	1	Color, 64k colors (RGB: 565), 16 bpp
Other than above							Reserved

Note: The number of colors that can be selected in rotation mode is restricted by the display resolution. For details, see table 25.3, in section 25.3, Operation.

25.2.4 LCDC Scan Mode Register (LDSMR)

LDSMR selects whether or not to enable the hardware rotation function that is used to rotate the LCD panel, and sets the burst length for the system memory (VRAM) obtained for display.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	ROT	—	—	—	AU1	AU0	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R	R	R	R/W	R/W	R	R	R	R	R	R	R	R

Bits 15, 14, 12 to 10, and 7 to 0—Reserved

Bit 13—Rotation Module Select (ROT): Selects whether or not to rotate the display by hardware. Note that the following restrictions are applied to rotation.

- An STN or TFT panel must be used. A DSTN panel is not allowed.
- The maximum horizontal (internal scan direction of the LCD panel) width of the LCD panel is 320.
- Set a binary exponential that exceeds the display size in LDLAOR. (For example, select 256 when a 320 × 240 panel is rotated to be used as a 240 × 320 panel and the horizontal width of the image is 240 bytes.)

Bit 13 ROT	Description	
0	Not rotated	(Initial value)
1	Rotated 90 degrees rightwards (left side of image is displayed on the upper side of the LCD module)	

Bits 9 and 8—Access Unit Select (AU1 and AU0): Select the unit for accessing VRAM. This bit is valid only when the ROT bit is set to 1 (rotation is performed). When the ROT bit is cleared to 0, 16-burst operation is performed regardless of the AU bits.

Bit 9 AU1	Bit 8 AU0	Description	
0	0	4-burst operation	(Initial value)
	1	8-burst operation	
1	0	16-burst operation	
	1	32-burst operation	

25.2.5 LCDC Start Address Register for Upper Display Data Fetch (LDSARU)

LDSARU sets the start address from which data is fetched by the LCDC for upper display of the LCDC panel. When a DSTN panel is used, this register specifies the fetch start address for the upper side of the panel.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	SAU25	SAU24	SAU23	SAU22	SAU21	SAU20	SAU19	SAU18	SAU17	SAU16
Initial value:	0	0	0	0	1	1	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

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SAU15	SAU14	SAU13	SAU12	SAU11	SAU10	SAU9	SAU8	SAU7	SAU6	SAU5	SAU4	SAU3	SAU2	SAU1	SAU0
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 31 to 26—Reserved

Bits 25 to 0—Start Address for Upper Display Data Fetch (SAU31 to SAU0): The start address for data fetch of the display data must be set within the synchronous DRAM area of area 3.

Notes: 1. When using the hardware rotation function (ROT = 1), set the LDSARU value so that the upper-left address of the image is aligned with the 512-byte boundary.

2. When the hardware rotation function is used (ROT = 1), set the upper-left address of the image which can be calculated from the display image size in this register. The equation below shows how to calculate the LDSARU value when the image size is 240 × 340 and LDLAOR = 256. The LDSARU value is obtained not from the panel size but from the memory size of the image to be displayed. Note that LDLAOR must be a binary exponential at least as large as the horizontal width of the image. Calculate backwards using the LDSARU value (LDSARU − 256 (LDLAOR value) × (320 − 1)) to ensure that the upper-left address of the image is aligned with the 512-byte boundary.

   LDSARU = (upper-left address of image) + 256 (LDLAOR value) × 319 (line)

25.2.6 LCDC Start Address Register for Lower Display Data Fetch (LDSARL)

LDSARL sets the start address from which data is fetched by the LCDC for lower display of the LCD panel. When a DSTN panel is used, this register specifies the fetch start address for the lower side of the panel.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	SAL25	SAL24	SAL23	SAL22	SAL21	SAL20	SAL19	SAL18	SAL17	SAL16
Initial value:	0	0	0	0	1	1	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

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SAL15	SAL14	SAL13	SAL12	SAL11	SAL10	SAL9	SAL8	SAL7	SAL6	SAL5	SAL4	SAL3	SAL2	SAL1	SAL0
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 31 to 26—Reserved

Bits 25 to 0—Start Address for Lower Panel Display Data Fetch (SAL31 to SAL0): The start address for data fetch of the display data must be set within the synchronous DRAM area of area 3.

STN and TFT: Cannot be used

DSTN: Start address for fetching display data corresponding to the lower panel

Note: The minimum alignment unit of LDSARU and LDSARL is four bytes. Because the LCDC handles these values as longword data, the values written to the lower two bits of each register are always treated as 0. The lower two bits of each register are always read as 0. For 1 or 2 bpp, set the registers so that the start of each line is aligned with the longword boundary (32 bits). (Data at the start of each line is always valid.) Data that exceeds the longword boundary at the end of each line (1, 2, or 3 bytes) will be discarded. For 4, 8, 15, 16, or 32 bpp, set the registers so that the start of each line is aligned with the longword boundary (32 bits).

25.2.7 LCDC Line Address Offset Register for Display Data Fetch (LDLAOR)

LDLAOR sets the address width of the Y-coordinates increment used for LCDC to read the image recognized by the graphics driver. This register specifies how many bytes the address from which data is to be read should be moved when the Y coordinates (vertical direction) have been incremented by 1. This register does not have to be equal to the horizontal width of the LCD panel. When the memory address of a point (X, Y) in the two-dimensional image is calculated by Ax + By+ C, this register becomes equal to B in this equation.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	LAO15	LAO14	LAO13	LAO12	LAO11	LAO10	LAO9	LAO8	LAO7	LAO6	LAO5	LAO4	LAO3	LAO2	LAO1	LAO0
Initial value:	0	0	0	0	0	0	1	0	1	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 0—Line Address Offset (LDLAOR)

Notes: 1. The minimum alignment unit of LDLAOR is four bytes. Because the LCDC handles these values as longword data, the values written to the lower two bits of the register are always treated as 0. The lower two bits of the register are always read as 0. The initial values (× resolution = 640) will continuously and accurately place the VGA (640 × 480 dots) display data without skipping an address between lines. For details, see table 25.3, in section 25.3, Operation.

2. A binary exponential at least as large as the horizontal width of the image is recommended for the LDLAOR value, taking into consideration the software operation speed. When the hardware rotation function is used (ROT = 1), the LDLAOR value should be a binary exponential (in this example, 256) at least as large as the horizontal width of the image (after rotation, it becomes 240 in a 240 × 320 panel) instead of the horizontal width of the LCD panel (320 in a 320 × 240 panel).

25.2.8 LCDC Palette Control Register (LDPALCR)

LDPALCR selects whether the CPU or LCDC accesses the palette memory. When the palette memory is being used for display operation, display mode should be selected. When the palette memory is being written to, CPU access mode should be selected.

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

Bits 15 to 5 and 3 to 1—Reserved

Bit 4—Palette State (PALS): Indicates the access right state of the palette.

Bit 4 PALS	Description	
0	Display mode: LCDC uses the palette	(Initial value)
1	CPU access mode: The host (CPU) uses the palette	

Bit 0—Palette Read/Write Enable (PALEN): Controls CPU accesses to the palette.

Bit 0 PALEN	Description	
0	Display mode: LCDC uses the palette	(Initial value)
1	CPU access mode: The host (CPU) uses the palette	

25.2.9 Palette Data Registers 00 to FF (LDPR00 to LDPRFF)

These registers are for accessing palette data directly allocated (4 bytes × 256 addresses) to the memory space. To access the palette memory, access the corresponding register among this register group (LDPR00 to LDPRFF). Each palette register is a 32-bit register including three 8-bit areas for R, G, and B. For details on the color palette specifications, see section 25.3.3, Color Palette Specification.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	PAL D00-FF	PAL D00-FF	PAL D00-FF	PAL D00-FF	PAL D00-FF	PAL D00-FF	PAL D00-FF	PAL D00-FF
Initial value:	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PAL D00-FF	PAL D00-FF	PAL D00-FF	PAL D00-FF	PAL D00-FF	PAL D00-FF	PAL D00-FF	PAL D00-FF	PAL D00-FF	PAL D00-FF	PAL D00-FF	PAL D00-FF	PAL D00-FF	PAL D00-FF	PAL D00-FF	PAL D00-FF
Initial value:	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
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

*: Undefined

Bits 31 to 24—Reserved

Bits 18 to 16, 9, 8, and 2 to 0—Reserved Bits in Palettes R, G, and B (Though they cannot be changed, they are extended according to the upper bits.)

Bits 23 to 19, 15 to 10, 7 to 3—Palette Data (PALD00 to PALDFF):

PALD00: H'10008000
PALDnn: (H'10008000+4×nn)
PALDFF: H'100083FC

25.2.10 LCDC Horizontal Character Number Register (LDHCNR)

LDHCNR specifies the LCD module's horizontal size (in the scan direction) and the entire scan width including the horizontal retrace period.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	HDCN7	HDCN6	HDCN5	HDCN4	HDCN3	HDCN2	HDCN1	HDCN0	HTCN7	HTCN6	HTCN5	HTCN4	HTCN3	HTCN2	HTCN1	HTCN0
Initial value:	0	0	1	0	1	1	1	1	0	1	0	1	0	0	1	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 Character Number (HDCN): Set the number of horizontal display characters (unit: character = 8 dots).

Subtract 1 from the setting (0 to 252 (H'FC)).

Example: For an LCD module with a width of 640 pixels
HDCN = (640/8) – 1 = 79 = H'4F

Bits 7 to 0—Horizontal Total Character Number (HTCN): Set the number of total horizontal characters (unit: character = 8 dots). Subtract 1 from the setting (3 to 255 (H'FF)).

The minimum horizontal retrace period is three characters (24 dots).
The values set in HDCN and HTCN must satisfy the relationship of HTCN ≥ HDCN + 3.
HTCN should be set to an odd number value when using a color STN 16-bit I/F module.

Example: For an LCD module with a width of 640 pixels
HTCN = [(640/8) – 1] + 3 = 82 = H'52

25.2.11 LCDC Horizontal Sync Signal Register (LDHSYNR)

LDHSYNR specifies the timing of the generation of the horizontal (scan direction) sync signals (CL1/Hsync) for the LCD module.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	HSYNW3	HSYNW2	HSYNW1	HSYNW0	—	—	—	—	HSYNP7	HSYNP6	HSYNP5	HSYNP4	HSYNP3	HSYNP2	HSYNP1	HSYNP0
Initial value:	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bits 15 to 12—Horizontal Sync Signal Width (HSYNW): Sets the width in characters of the horizontal sync signals (CL1 and Hsync).

Subtract 1 from the setting (0 to 15 (H'F)).

Example: For a horizontal sync signal width of 8 dots
HSYNW = (8 dots/8 dots/character) – 1 = 0 = H'0

Bits 7 to 0—Horizontal Sync Signal Output Position (HSYNP): Sets the output position in characters of the horizontal sync signals.

Subtract 1 from the setting (0 to 255 (H'FF)).

The following conditions must be satisfied: HTCN >= HSYNP + HSYNW + 1
HSYNP >= HDCN + 1

Example: For an LCD module with a width of 640 pixels
HSYNP = [(640/8) + 1] – 1 = 80 = H'50
In this case, the horizontal sync signal is active from the 648th through the 655th dot.

25.2.12 LCDC Vertical Display Line Number Register (LDVDLNR)

LDVDLNR specifies the LCD module's vertical size (for both scan direction and vertical direction). For a DSTN panel, specify an even number at least as large as the LCD panel's vertical size regardless of the size of the up and down panels, e.g. 480 for a 640 × 480 panel.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	VDLN 10	VDLN 9	VDLN 8	VDLN 7	VDLN 6	VDLN 5	VDLN 4	VDLN 3	VDLN 2	VDLN 1	VDLN 0
Initial value:	0	0	0	0	0	0	0	1	1	1	0	1	1	1	1	1
R/W:	R	R	R	R	R	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 0—Vertical Display Line Number (VDLN): Set the number of vertical display lines (unit: line).

Subtract 1 from the setting (0 to 2047 (H'7FF)).

Example: For an 480-line LCD module
VDLN = 480 – 1 = H'1DF

25.2.13 LCDC Vertical Total Line Number Register (LDVTLNR)

LDVTLNR specifies the LCD panel's entire vertical size including the vertical retrace period.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	VTLN 10	VTLN 9	VTLN 8	VTLN 7	VTLN 6	VTLN 5	VTLN 4	VTLN 3	VTLN 2	VTLN 1	VTLN 0
Initial value:	0	0	0	0	0	0	0	1	1	1	0	1	1	1	1	1
R/W:	R	R	R	R	R	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 0—Vertical Total Line Number (VTLN): Set the total number of vertical display lines (unit: line).

Subtract 1 from the setting (1 to 2047 (H'7FF)).

The minimum for the total number of vertical lines is 2 lines. The following conditions must be satisfied: VTLN >= VDLN, VTLN >= 1

Example: For an 480-line LCD module and a vertical retrace period of 0 lines
VTLN = (480 + 0) – 1 = 479 = H'1DF

25.2.14 LCDC Vertical Sync Signal Register (LDVSYNR)

LDVSYNR specifies the timing of the generation of the vertical (scan direction and vertical direction) sync signals (FLM/Vsync) for the LCD module.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	VSYNW3	VSYNW2	VSYNW1	VSYNW0	—	VSYNP10	VSYNP9	VSYNP8	VSYNP7	VSYNP6	VSYNP5	VSYNP4	VSYNP3	VSYNP2	VSYNP1	VSYNP0
Initial value:	0	0	0	0	0	0	0	1	1	1	0	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 11—Reserved

Bits 15 to 12—Vertical Sync Signal Width (VSYNW): Set the width of the vertical sync signals (FLM and Vsync) (unit: line).

Subtract 2 from the setting (0 to 15 (H'F)).

Example: For a vertical sync signal width of 1 line
VSYNW = (1 – 1) = 0 = H'0

Bits 10 to 0— Vertical Sync Signal Output Position (VSYNP): Set the output position of the vertical sync signals (FLM and Vsync) (unit: line).

Subtract 2 from the setting (0 to 2046 (H'7FE)).

DSTN should be set to an odd number value. It is handled as (setting value + 1)/ 2.

Example: For an 480-line LCD module and a vertical retrace period of 0 lines (in other words, VTLN = 479 and the vertical sync signal is active for the first line):

- Single display
  VSYNP=[(1-1)+VTLN] mod (VTLN+1) = [(1-1)+479] mod (479+1)
  = 479 mod 480 = 479
  = H'1DF
- Dual displays
  VSYNP=[(1-1)×2+VTLN] mod (VTLN+1) = [(1-1)×2+479] mod (479+1)
  = 479 mod 480 = 479
  = H'1DF

25.2.15 LCDC AC Modulation Signal Toggle Line Number Register (LDACLNR)

LDACLNR specifies the timing to toggle the AC modulation signal (LCD current-alternating signal) of the LCD module.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	ACLN4	ACLN3	ACLN2	ACLN1	ACLN0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W

Bits 15 to 5—Reserved

Bits 4 to 0—AC Line Number (ACLN): Set the line number where the LCD current-alternating signal of the LCD module is toggled (unit: line).

Subtract 1 from the setting (0 to 31 (H'1F)).

Note: When the total line number of the LCD panel is even, set an even number so that toggling is performed at an odd line.

Example: For toggling every 13 lines
ACLN = 13 – 1 = 12 = H'0C

25.2.16 LCDC Interrupt Control Register (LDINTR)

LDINTR specifies where to start the Vsync interrupt (LCDCI).

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	VINT SEL	—	—	—	VINTE	—	—	—	—	—	—	—	VINTS
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R	R	R	R/W	R	R	R	R	R	R	R	R/W

Bits 15 to 13, 11 to 9, and 7 to 1—Reserved

Bit 12—Vsync Interrupt Select (VINTSEL): Sets the starting point of the LCDC's Vsync interrupt.

Bit 12 VINTSEL	Description
0	Vsync interrupt is generated at starting point of vertical retrace period for memory access (Initial value)
1	Vsync interrupt is generated at starting point of vertical retrace period for LCD display

Bit 8—Vsync Interrupt Enable (VINTE): Sets whether or not to enable LCDC's Vsync interrupts.

Bit 8 VINTE	Description
0	Vsync interrupts are disabled (Initial value)
1	Vsync interrupts are enabled

Bit 0—Vsync Interrupt State (VINTS): Indicates the LCDC's Vsync interrupt handling state. This bit is set to 1 at the time a Vsync interrupt is generated. During the Vsync interrupt handling routine, this bit should be cleared by writing 0 to it.

Bit 0 VINTS	Description
0	LCDC did not generate a Vsync interrupt or has been informed that the generated Vsync interrupt has completed (Initial value)
1	LCDC has generated a Vsync interrupt and has not yet been informed that the generated Vsync interrupt has completed

Notes: • Interrupt Handling Flow:

1. An interrupt signal is input to the CPU.
2. The CPU reads from VINTS.
3. If VINTS is set to 1, a Vsync interrupt has occurred, and the Vsync interrupt handling is carried out.
4. If VINTS is cleared to 0, no Vsync interrupt has occurred and another processing is carried out.

• When Vsync interrupts are enabled, the VINTE bit must be set to 1 before the DON bit is set to 1, and the VINTE bit must not be cleared to 0.

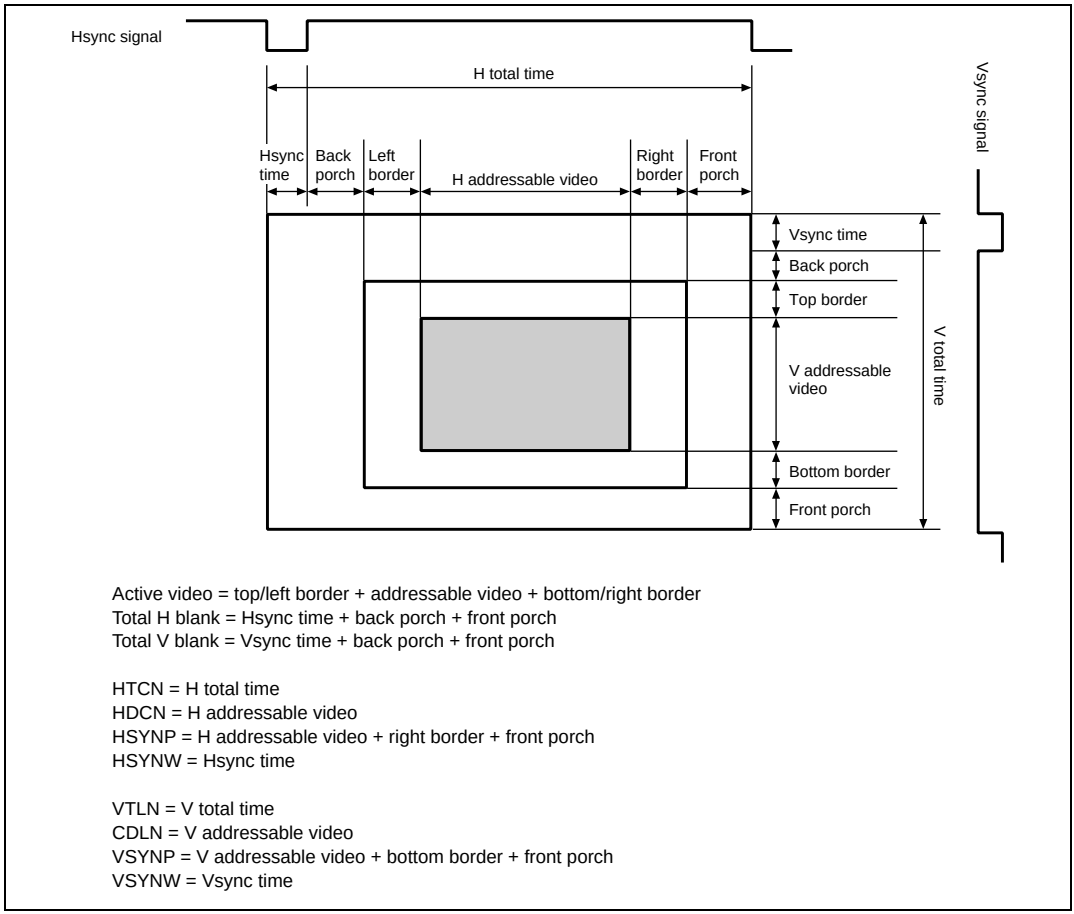


Figure 25.2 Valid Display and Retrace Period

25.2.17 LCDC Power Management Mode Register (LDPMMR)

LDPMMR controls the power supply circuit that provides power to the LCD module. The usage of two types of power-supply control pins, VCPWC and VEPWC, and turning on or off the power supply are selected.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ONC3	ONC2	ONC1	ONC0	OFFD3	OFFD2	OFFD1	OFFD0	—	VCPE	VEPE	DONE	—	—	LPS1	LPS0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R	R	R

Bits 7, 3, and 2—Reserved

Bits 15 to 12—LCDC Power-On Sequence Period (ONC): Set the period from VEPWC assertion to DON assertion in the power-on sequence of the LCD module in frame units.

This period is the (c) period in figures 25.4 to 25.7. For details on setting this register, see table 25.4. (The setting method is common for ONC, ONA, ONB, OFFD, OFFE, and OFFF.) 1 is to be subtracted from the setting.

Bits 11 to 8—LCDC Power-Off Sequence Period (OFFD): Set the period from DON negation to VEPWC negation in the power-off sequence of the LCD module in frame units.

This period is the (d) period in figures 25.4 to 25.7. 1 is to be subtracted from the setting.

Bit 6—VCPWC Pin Enable (VCPE): Sets whether or not to enable a power-supply control sequence using the VCPWC pin.

Bit 6 VCPE	Description
0	Disabled: VCPWC pin is masked and fixed low (Initial value)
1	Enabled: VCPWC pin output is asserted and negated according to the power-on or power-off sequence

Bit 5—VEPWC Pin Enable (VEPE): Sets whether or not to enable a power-supply control sequence using the VEPWC pin.

Bit 5 VEPE	Description
0	Disabled: VEPWC pin is masked and fixed low (Initial value)
1	Enabled: VEPWC pin output is asserted and negated according to the power-on or power-off sequence

Bit 4—DON Pin Enable (DONE): Sets whether or not to enable a power-supply control sequence using the DON pin.

Bit 4 DONE	Description
0	Disabled: DON pin is masked and fixed low
1	Enabled: DON pin output is asserted and negated according to the power-on or power-off sequence (Initial value)

Bits 1 and 0—LCD Module Power-Supply Input State (LPS1 and LPS0): Indicate the power-supply input state of the LCD module when using the power-supply control function.

Bit 1 LPS1	Bit 0 LPS0	Description	
0	0	LCD module power off	(Initial value)
1	1	LCD module power on	

25.2.18 LCDC Power-Supply Sequence Period Register (LDPSPR)

LDPSPR controls the power supply circuit that provides power to the LCD module. The timing to start outputting the timing signals to the VEPWC and VCPWC pins is specified.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ONA3	ONA2	ONA1	ONA0	ONB3	ONB2	ONB1	ONB0	OFFE3	OFFE2	OFFE1	OFFE0	OFFF3	OFFF2	OFFF1	OFFF0
Initial value:	1	1	1	1	0	1	1	0	0	0	0	0	1	1	1	1
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—LCDC Power-On Sequence Period (ONA): Set the period from VCPWC assertion to starting output of the display data (LCDD) and timing signals (FLM, CL1, CL2, and M/DISP) in the power-on sequence of the LCD module in frame units.

This period is the (a) period in figures 25.4 to 25.7. 1 is to be subtracted from the setting.

Bits 11 to 8—LCDC Power-On Sequence Period (ONB): Set the period from starting output of the display data (LCDD) and timing signals (FLM, CL1, CL2, and DISP/M) to the VEPWC assertion in the power-on sequence of the LCD module in frame units.

This period is the (b) period in figures 25.4 to 25.7. 1 is to be subtracted from the setting.

Bits 7 to 4—LCDC Power-Off Sequence Period (OFFE): Set the period from VEPWC negation to stopping output of the display data (LCDD) and timing signals (FLM, CL1, CL2, and DISP/M) in the power-off sequence of the LCD module in frame units.

This period is the (e) period in figures 25.4 to 25.7. 1 is to be subtracted from the setting.

Bits 3 to 0—LCDC Power-Off Sequence Period (OFFF): Set the period from stopping output of the display data (LCDD) and timing signals (FLM, CL1, CL2, and DISP/M) to VCPWC negation to in the power-off sequence of the LCD module in frame units.

This period is the (f) period in figures 25.4 to 25.7. 1 is to be subtracted from the setting.

25.2.19 LCDC Control Register (LDCNTR)

LDCNTR specifies start and stop of display by the LDCD.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	DON2	—	—	—	DON
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R	R	R/W

Bits 15 to 1—Reserved

Bit 4— Display start auxiliary bit (DON2): Specifies the start of display operation using LCDC. LCDC operation cannot be guaranteed if 0 is written to this bit at any time other than the start of display operation. Note that a 1 written to this bit is automatically cleared to 0, so it is not necessary to write 0 to it in order to clear it.

Bit 0—Display On (DON): Specifies the start and stop of the LCDC display operation.

The control sequence state can be checked by referencing the LPS value in bit 0 of the LCDC power management mode register (LDPMMR).

Bit 4 DON2	Bit 0 DON	Description	
0	0	Display-off mode: LCDC is stopped	(Initial value)
1	1	Display-on mode: LCDC operates	

Starting LCDC Display Operation (DON2 and DON bits change from B'00 to B'11):

1. Start LCDC operation.
2. Turn on the LCD module following the sequence set in the LCDC power management mode register (LDPMMR) and LCDC control register (LDCNTR).
   The sequence ends when the LPS value changes from B'00 to B'11.
   Do not make any action to the DON bit until the sequence ends.

Stopping LCDC Display Operation (DON2 and DON bits change from B'01 to B'00):

1. Turn off the LCD module following the sequence set in the LCDC power management mode register (LDPMMR) and LCDC control register (LDCNTR).
2. Stop LCDC operation.
   The sequence ends when the LPS value changes from B'11 to B'00.
   Do not make any action to the DON bit until the sequence ends.

25.3 Operation

25.3.1 LCD Module Sizes which can be Displayed in this LCDC

This LCDC is capable of controlling displays with up to 1024 × 1024 dots and 16 bpp (bits per pixel). The image data for display is stored in system memory, which is shared with the CPU. This LCDC should read the data from system memory between display periods.

The SH7727 has a maximum 32-burst memory read operation and a 2.4-kbyte line buffer, so although a complete breakdown of the display is unlikely, there may be some problems with the display depending on the combination.

The bus-occupancy rate described below should not, as a rule, exceed 40%.

$$\text{Bus-occupancy rate (\%)} = \frac{\text{Overhead coefficient} \times \text{total number of display pixels } ((\text{HDCN} + 1) \times 8 \times (\text{VDLN} + 1)) \times \text{frame rate (Hz)} \times \text{number of colors (bpp)} \times 100}{\text{CKIO (Hz)} \times \text{bus width (bits)}}$$

The overhead coefficient depends on the bus used by the SDRAM in CL2, as indicated below.

If the hardware rotation function is not used (ROT = 0), the overhead coefficient is 1.375 if a 32-bit bus is used and 1.188 if a 16-bit bus is used.

If the hardware rotation function is used (ROT = 1), the overhead coefficient is determined by the access unit select (AU) setting and the bus width, as follows.

Access Unit Select (AU) Setting	32-Bit Bus	16-Bit Bus
4-burst operation	2.500	1.750
8-burst operation	1.750	1.375
16-burst operation	1.375	1.188
32-burst operation	1.188	1.094

25.3.2 Limits on the Resolution of Rotated Displays, Burst Length, and Connected Memory (SDRAM)

This LCDC is capable of displaying a landscape-format image on a LCD module by rotating a portrait format image for display by 90 degrees. Only the numbers of colors for each resolution are supported as shown in tables 25.3 and 25.4. The size of the SDRAM (the number of column address bits) and its burst length are limited to read the SDRAM continuously.

The number of colors for display, SDRAM column addresses, and LCDC burst length are shown table 25.3 and 25.4.

A monochromatic LCD module is necessary for the display of images in the above monochromatic formats. A color LCD module is necessary for the display of images in the above color formats.

Table 25.3 Limits on the Resolution of Rotated Displays, Burst Length, and Connected Memory (32-bit Bus SDRAM)

Image for Display in Memory (X-Resolution × Y-Resolution)	LCD Module (X-Resolution × Y-Resolution)	Number of Colors for Display		Number of Column Address Bits of SDRAM	Burst Length of LCDC (LDSMR*)
240 × 320	320 × 240	Monochrome	4 bpp (packed)	8 bits	Not more than 8 bursts
				9 bits	Not more than 16 bursts
				10 bits	—
			4 bpp (unpacked)	8 bits	4 bursts
				9 bits	Not more than 8 bursts
				10 bits	Not more than 16 bursts
			6 bpp	8 bits	4 bursts
				9 bits	Not more than 8 bursts
				10 bits	Not more than 16 bursts
		Color	8 bpp	8 bits	4 bursts
				9 bits	Not more than 8 bursts
				10 bits	Not more than 16 bursts
			16 bpp	8 bits	Unusable
				9 bits	4 bursts
				10 bits	Not more than 8 bursts
234 × 320	320 × 234	Monochrome	6 bpp	8 bits	4 bursts
				9 bits	Not more than 8 bursts
				10 bits	Not more than 16 bursts
		Color	16 bpp	8 bits	Unusable
				9 bits	4 bursts
				10 bits	Not more than 8 bursts

Image for Display in Memory (X-Resolution × Y-Resolution)	LCD Module (X-Resolution × Y-Resolution)	Number of Colors for Display		Number of Column Address Bits of SDRAM	Burst Length of LCDC (LDSMR*)
80 × 160	160 × 80	Monochrome	2 bpp	8 bits	—
				9 bits	—
				10 bits	—
			4 bpp (packed)	8 bits	Not more than 16 bursts
				9 bits	—
				10 bits	—
			4 bpp (unpacked)	8 bits	Not more than 8 bursts
				9 bits	Not more than 16 bursts
				10 bits	—
			6 bpp	8 bits	Not more than 8 bursts
				9 bits	Not more than 16 bursts
				10 bits	—
		Color	4 bpp (packed)	8 bits	Not more than 16 bursts
				9 bits	—
				10 bits	—
			4 bpp (unpacked)	8 bits	Not more than 8 bursts
				9 bits	Not more than 16 bursts
				10 bits	—
			8 bpp	8 bits	Not more than 8 bursts
				9 bits	Not more than 16 bursts
				10 bits	—
			16 bpp	8 bits	4 bursts
				9 bits	Not more than 8 bursts
				10 bits	Not more than 16 bursts

Image for Display in Memory (X-Resolution × Y-Resolution)	LCD Module (X-Resolution × Y-Resolution)	Number of Colors for Display		Number of Column Address Bits of SDRAM	Burst Length of LCDC (LDSMR*)
64 × 128	128 × 64	Monochrome	1 bpp	8 bits	—
				9 bits	—
				10 bits	—
			2 bpp	8 bits	—
				9 bits	—
				10 bits	—
			4 bpp (packed)	8 bits	—
				9 bits	—
				10 bits	—
			4 bpp (unpacked)	8 bits	Not more than 16 bursts
				9 bits	—
				10 bits	—
			6 bpp	8 bits	Not more than 16 bursts
				9 bits	—
				10 bits	—
		Color	4 bpp (packed)	8 bits	—
				9 bits	—
				10 bits	—
			4 bpp (unpacked)	8 bits	Not more than 16 bursts
				9 bits	—
				10 bits	—
			8 bpp	8 bits	Not more than 16 bursts
				9 bits	—
				10 bits	—

Note: * Specify the data of the number of line specified as burst length can be stored in address of SDRAM same as that of ROW.

Table 25.4 Limits on the Resolution of Rotated Displays, Burst Length, and Connected Memory (16-bit Bus SDRAM)

Image for Display in Memory (X-Resolution × Y-Resolution)	LCD Module (X-Resolution × Y-Resolution)	Number of Colors for Display		Number of Column Address Bits of SDRAM	Burst Length of LCDC (LDSMR*)
240 × 320	320 × 240	Monochrome	4 bpp (packed)	8 bits	Not more than 4 bursts
				9 bits	Not more than 8 bursts
				10 bits	Not more than 16 bursts
			4 bpp (unpacked)	8 bits	Unusable
				9 bits	4 bursts
				10 bits	Not more than 8 bursts
			6 bpp	8 bits	Unusable
				9 bits	4 bursts
				10 bits	Not more than 8 bursts
		Color	8 bpp	8 bits	Unusable
				9 bits	4 bursts
				10 bits	Not more than 8 bursts
			16 bpp	8 bits	Unusable
				9 bits	Unusable
				10 bits	4 bursts
234 × 320	320 × 234	Monochrome	6 bpp	8 bits	Unusable
				9 bits	4 bursts
				10 bits	Not more than 8 bursts
		Color	16 bpp	8 bits	Unusable
				9 bits	Unusable
				10 bits	4 bursts

Image for Display in Memory (X-Resolution × Y-Resolution)	LCD Module (X-Resolution × Y-Resolution)	Number of Colors for Display		Number of Column Address Bits of SDRAM	Burst Length of LCDC (LDSMR*)
80 × 160	160 × 80	Monochrome	2 bpp	8 bits	Not more than 16 bursts
				9 bits	—
				10 bits	—
			4 bpp (packed)	8 bits	Not more than 8 bursts
				9 bits	Not more than 16 bursts
				10 bits	—
			4 bpp (unpacked)	8 bits	4 bursts
				9 bits	Not more than 8 bursts
				10 bits	Not more than 16 bursts
			6 bpp	8 bits	4 bursts
				9 bits	Not more than 8 bursts
				10 bits	Not more than 16 bursts
		Color	4 bpp (packed)	8 bits	Not more than 8 bursts
				9 bits	Not more than 16 bursts
				10 bits	—
			4 bpp (unpacked)	8 bits	4 bursts
				9 bits	Not more than 8 bursts
				10 bits	Not more than 16 bursts
			8 bpp	8 bits	4 bursts
				9 bits	Not more than 8 bursts
				10 bits	Not more than 16 bursts
			16 bpp	8 bits	Unusable
				9 bits	4 bursts
				10 bits	Not more than 8 bursts

Image for Display in Memory (X-Resolution × Y-Resolution)	LCD Module (X-Resolution × Y-Resolution)	Number of Colors for Display		Number of Column Address Bits of SDRAM	Burst Length of LCDC (LDSMR*)
64 × 128	128 × 64	Monochrome	1 bpp	8 bits	—
				9 bits	—
				10 bits	—
			2 bpp	8 bits	—
				9 bits	—
				10 bits	—
			4 bpp (packed)	8 bits	Not more than 16 bursts
				9 bits	—
				10 bits	—
			4 bpp (unpacked)	8 bits	Not more than 8 bursts
				9 bits	Not more than 16 bursts
				10 bits	—
			6 bpp	8 bits	Not more than 8 bursts
				9 bits	Not more than 16 bursts
				10 bits	—
		Color	4 bpp (packed)	8 bits	Not more than 16 bursts
				9 bits	—
				10 bits	—
			4 bpp (unpacked)	8 bits	Not more than 8 bursts
				9 bits	Not more than 16 bursts
				10 bits	—
			8 bpp	8 bits	Not more than 8 bursts
				9 bits	Not more than 16 bursts
				10 bits	—

Note: * Specify the data of the number of line specified as burst length can be stored in address of SDRAM same as that of ROW.

25.3.3 Color Palette Specification

Color Palette Register: This LCDC has a color palette which outputs 24 bits of data per entry and is able to simultaneously hold 256 entries. The color palette thus allows the simultaneous display of 256 colors chosen from among 16-M colors.

The below procedure may be used to set up color palettes at any time.

1. The PALEN bit in the LCDC color palette register is 0 (initial value); normal display operation
2. Access LDPALCR and set the PALEN bit to 1; enter color-palette setting mode
3. Access LDPALCR and confirm that the PALS bit is 1.
4. Access LDPR00 to LDPRFF and write the required values to the PALD00 to PALDFF bits.
5. Access LDPALCR and clear the PALEN bit to 0; return to normal display mode

0 is output on the LCDC display data output (LCDD) while the controller is in color palette setting mode.

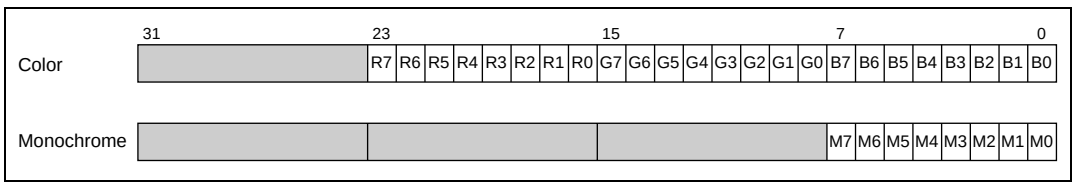


Figure 25.3 Color-Palette Data Format

PALDnn color and gradation data should be set as above, using 256-gradation values for R, G, B, and M.

For a color display, PALDnn [23:16], PALDnn [15:8], and PALDnn [7:0] respectively hold the R, G, and B data. Although the bits PALDnn [18:16], PALDnn [9:8], and PALDnn [2:0] exist, no memory is associated with these bits. PALDnn [18:16], PALDnn [9:8], and PALDnn [2:0] are thus not available for storing palette data. The numbers of valid bits are thus R: 5, G: 6, and B: 5.24-bit (R: 8 bits, G: 8 bits, and B: 8 bits) data should, however, be written to the palette-data registers. When the values for PALDnn [23:19], PALDnn [15:10], or PALDnn [7:3] are not 0, 1s should be written to PALDnn [18:16], PALDnn [9:8], or PALDnn [2:0], respectively. When the values of PALDnn [23:19], PALDnn [15:10], or PALDnn [7:3] are 0, 0 should be written to PALDnn [18:16], PALDnn [9:8], or PALDnn [2:0], respectively. Then 24 bits are extended.

Grayscale data for a monochromatic display should be set in PALDnn [7:3]. PALDnn [23:8] are all ‘don’t care’. When the value in PALDnn [7:3] is not 0, 1s should be written to PALDnn [2:0]. When the value in PALDnn [7:3] is 0, 0s should be written to PALDnn [2:0]. Then 8 bits are extended.

25.3.4 Data Format

1. Packed 1bpp (Pixel Alignment in Byte is Big Endian) [Windows CE Recommended Format]

	MSB							LSB	
Address	7	6	5	4	3	2	1	0	[Bit]
+00	P00	P01	P02	P03	P04	P05	P06	P07	(Byte0)
+01	P08								(Byte1)
+02									
+03									
…				…					
+LAO+00	P10	P11	P12	P13	P14	P15	P16	P17	
+LAO+01	P18								
+LAO+02				…					
+LAO+03									
…				Display Memory					

↓Top Left Pixel

P00	P01	P02	P03	P04	P05	P06	P07	…
P10	P11	P12	P13	P14	P15	P16	P17	…
…								
…								

Display

Pn: Put 1bit data

LAO: Line Address Offset

—Unused bits should be 0

2. Packed 2bpp (Pixel Alignment in Byte is Big Endian) [Windows CE Recommended Format]

	MSB							LSB	
Address	7	6	5	4	3	2	1	0	[Bit]
+00	P00		P01		P02		P03		(Byte0)
+01	P04		P05		P06		P07		(Byte1)
+02									
+03									
…				…					
+LAO+00	P10		P11		P12		P13		
+LAO+01	P14		P15		P16		P17		
+LAO+02				…					
+LAO+03									
…				Display Memory					

↓Top Left Pixel

P00	P01	P02	P03	P04	P05	P06	P07	…
P10	P11	P12	P13	P14	P15	P16	P17	…
…								
…								

Display

Pn=Pn[1:0]: Put 2bit data

LAO: Line Address Offset

—Unused bits should be 0

3. Packed 4bpp (Pixel Alignment in Byte is Big Endian) [Windows CE Recommended Format]

	MSB							LSB	
Address	7	6	5	4	3	2	1	0	[Bit]
+00	P00				P01				(Byte0)
+01	P02				P03				(Byte1)
+02	P04				P05				(Byte2)
+03									
…				…					
+LAO+00	P00				P01				
+LAO+01	P02				P03				
+LAO+02	P04				P05				
+LAO+03				…					
…				Display Memory					

↓Top Left Pixel

P00	P01	P02	P03	P04	P05	P06	P07	…
P10	P11	P12	P13	P14	P15	P16	P17	…
…								
…								

Display

Pn=Pn[3:0]: Put 4bit data

LAO: Line Address Offset

—Unused bits should be 0

4. Packed 1bpp (Pixel Alignment in Byte is Little Endian)

	MSB							LSB	
Address	7	6	5	4	3	2	1	0	[Bit]
+00	P07	P06	P05	P04	P03	P02	P01	P00	(Byte0)
+01								P08	(Byte1)
+02									
+03									
…				…					
+LAO+00	P17	P16	P15	P14	P13	P12	P11	P10	
+LAO+01								P18	
+LAO+02				…					
+LAO+03									
…				Display Memory					

↓Top Left Pixel

P00	P01	P02	P03	P04	P05	P06	P07	…
P10	P11	P12	P13	P14	P15	P16	P17	…
…								
…								

Display

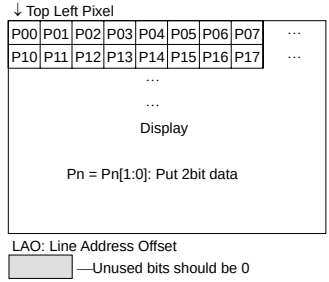
Pn: Put 1bit data

LAO: Line Address Offset

—Unused bits should be 0

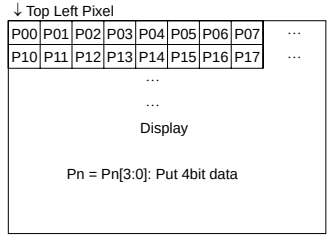
5. Packed 2bpp (Pixel Alignment in Byte is Little Endian)

Address	MSB 7	6	5	4	3	2	1	0 LSB	[Bit]
+00	P03		P02		P01		P00		(Byte0)
+01	P07		P06		P05		P04		(Byte1)
+02									
+03									
…			…						
+LAO+00	P13		P12		P11		P10		
+LAO+01	P17		P16		P15		P14		
+LAO+02			…						
+LAO+03									
…	Display Memory								



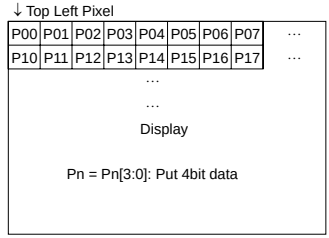
6. Packed 4bpp (Pixel Alignment in Byte is Little Endian)

Address	MSB 7	6	5	4	3	2	1	0 LSB	[Bit]
+00	P01				P00				(Byte0)
+01	P04				P03				(Byte1)
+02	P06				P05				(Byte2)
+03									
…			…						
+LAO+00	P11				P10				
+LAO+01	P13				P12				
+LAO+02	P15				P14				
+LAO+03			…						
…	Display Memory								



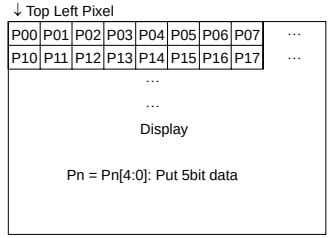
7. Unpacked 4bpp [Windows CE Recommended Format]

Address	MSB 7	6	5	4	3	2	1	0 LSB	[Bit]
+00					P00				(Byte0)
+01					P01				(Byte1)
+02					P02				(Byte2)
+03									
…			…						
+LAO+00					P10				
+LAO+01					P11				
+LAO+02					P12				
+LAO+03			…						
…	Display Memory								



8. Unpacked 5bpp [Windows CE Recommended Format]

Address	MSB 7	6	5	4	3	2	1	0 LSB	[Bit]
+00				P00					(Byte0)
+01				P01					(Byte1)
+02				P02					(Byte2)
+03									
…			…						
+LAO+00				P10					
+LAO+01				P11					
+LAO+02				P12					
+LAO+03			…						
…	Display Memory								



9. Unpacked 6bpp [Windows CE Recommended Format]

Address	MSB 7	6	5	4	3	2	1	LSB 0	[Bit]
+00			P00						(Byte0)
+01			P01						(Byte1)
+02			P02						(Byte2)
+03									
…			…						
+LAO+00			P10						
+LAO+01			P11						
+LAO+02			P12						
+LAO+03			…						
…			Display Memory						

↓ Top Left Pixel

P00	P01	P02	P03	P04	P05	P06	P07	…
P10	P11	P12	P13	P14	P15	P16	P17	…

…

…

Display

Pn = Pn[5:0]: Put 6bit data

LAO: Line Address Offset

—Unused bits should be 09

10. Packed 8bpp [Windows CE Recommended Format]

Address	MSB 7	6	5	4	3	2	1	LSB 0	[Bit]
+00				P00					(Byte0)
+01				P01					(Byte1)
+02				P02					(Byte2)
+03									
…				…					
+LAO+00				P10					
+LAO+01				P11					
+LAO+02				P12					
+LAO+03				…					
…				Display Memory					

↓ Top Left Pixel

P00	P01	P02	P03	P04	P05	P06	P07	…
P10	P11	P12	P13	P14	P15	P16	P17	…

…

…

Display

Pn = Pn[7:0]: Put 8bit data

LAO: Line Address Offset

—Unused bits should be 0

11. Unpacked color 15bpp (RGB 555) [Windows CE Recommended Format]

Address	MSB 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	LSB 0	[Bit]
+00		P00R					P00G					P00B					(Word0)
+02		P01R					P01G					P01B					(Word2)
+04		P02R					P02G					P02B					(Word4)
+06																	
…							…										
+LAO		P10R					P10G					P10B					
+LAO+02		P11R					P11G					P11B					
+LAO+04		P12R					P12G					P12B					
+LAO+06							…										
…							Display Memory										

↓ Top Left Pixel

P00	P01	P02	P03	P04	P05	P06	P07	…
P10	P11	P12	P13	P14	P15	P16	P17	…

…

Display

Pr = (PrR, PrG, PrB). Pr 15bit data

PrR = PrR[4.0]. Pr 5bit RED data

PrG = PrG[4.0]. Pr 5bit GREEN data

PrB = PrB[4.0]. Pr 5bit BLUE data

LAO: Line Address Offset

—Unused bits should be 0

12. Packed color 16bpp (RGB 565) [Windows CE Recommended Format]

Address	MSB 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	LSB 0	[Bit]
+00		P00R					P00G					P00B					(Word0)
+02		P01R					P01G					P01B					(Word2)
+04		P02R					P02G					P02B					(Word4)
+06																	
…							…										
+LAO		P10R					P10G					P10B					
+LAO+02		P11R					P11G					P11B					
+LAO+04		P12R					P12G					P12B					
+LAO+06							…										
…							Display Memory										

↓ Top Left Pixel

P00	P01	P02	P03	P04	P05	P06	P07	…
P10	P11	P12	P13	P14	P15	P16	P17	…

…

Display

Pr = (PrR, PrG, PrB). Pr 16bit data

PrR = PrR[4.0]. Pr 5bit RED data

PrG = PrG[5.0]. Pr 6bit GREEN data

PrB = PrB[4.0]. Pr 5bit BLUE data

LAO: Line Address Offset

—Unused bits should be 0

25.3.5 Timing Controller Register

The timing controller register is used to run the controller in a way that matches the display resolution of the LCD module. The display resolution is set up in the LCDC horizontal number character number register, LCDC horizontal synchronization signal register, LCDC vertical line displayed number register, LCDC vertical total line number register, and LCDC vertical synchronization signal register. The LCD current-alternating period for an STN or DSTN display is set by using the LCDC ac modulation signal toggle line number register. The initial values in these registers are typical settings for VGA (640 × 480 dots) on an STN or DSTN display.

The clock to be used is set with the LCD input clock register. The LCD module frame rate is determined by the display interval + retrace line interval (non-display interval) for one screen set in a size related register and the frequency of the clock used.

This LCDC has a V_{sync} interrupt function so that it is possible to issue an interrupt at the beginning of each vertical retrace line period (to be exact, at the beginning of the line after the last line of the display). This function is set up by using the LCDC interrupt control register.

25.3.6 Power Management Registers

An LCD module normally requires a specific sequence for processing to do with the cutoff of the input power supply. Settings in the LCDC power management mode register, LCDC power supply sequence period register, and LCDC control register, in conjunction with the LCD power-supply control pins (VCPWC, VEPWC, and DON), are used to provide processing of power-supply control sequences that suits the requirements of the LCD module.

Figures 25.4 to 25.7 are summary timing charts for power-supply control sequences and table 25.5 is a summary of available power-supply control sequence periods.

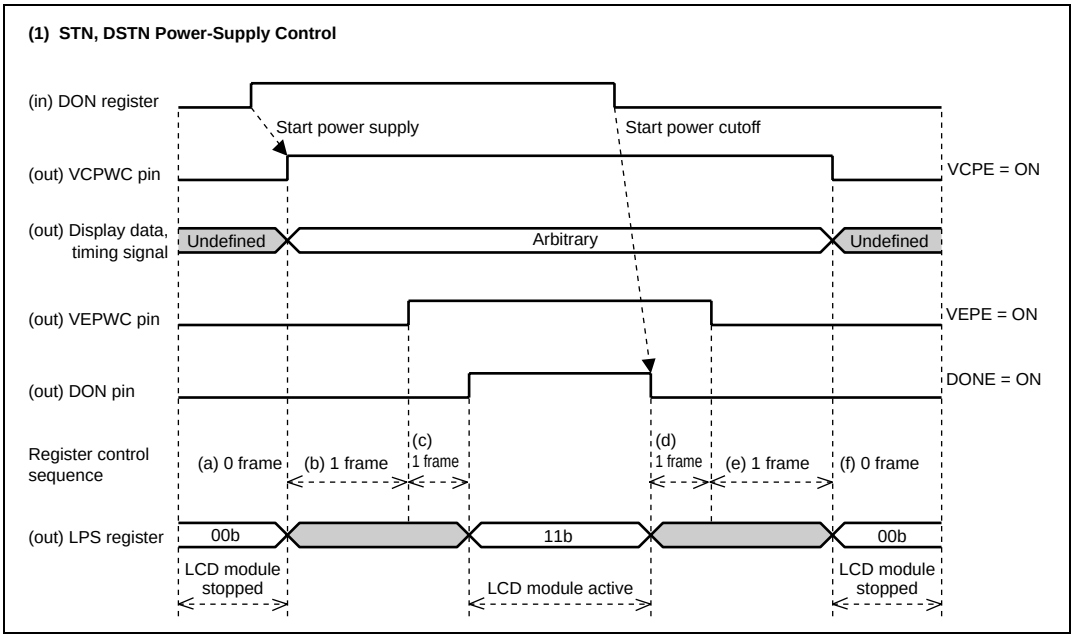


Figure 25.4 Power-Supply Control Sequence and States of the LCD Module

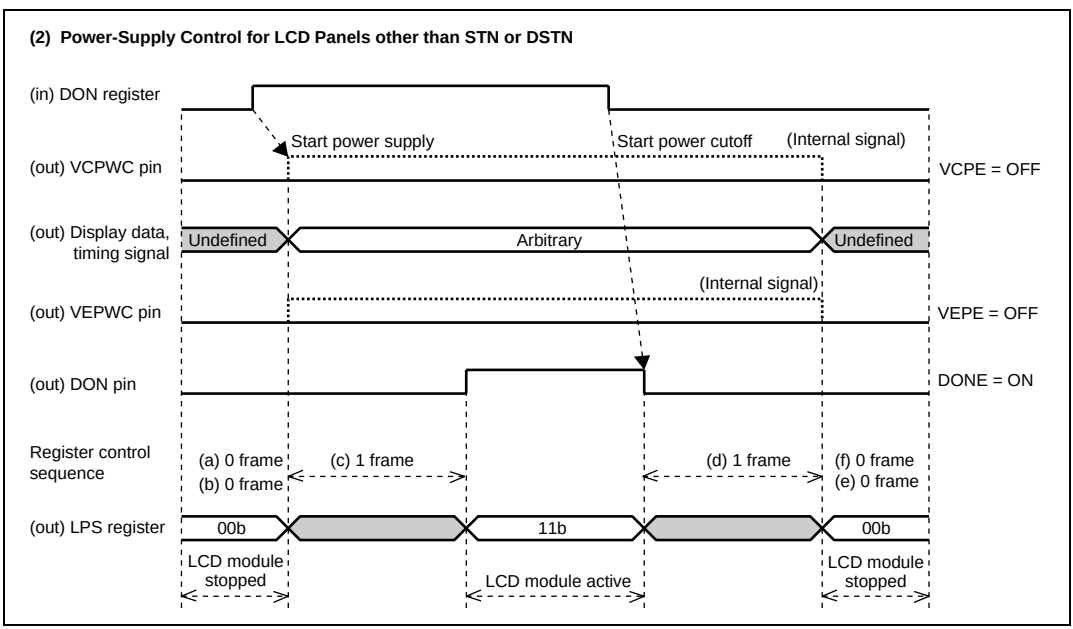


Figure 25.5 Power-Supply Control Sequence and States of the LCD Module

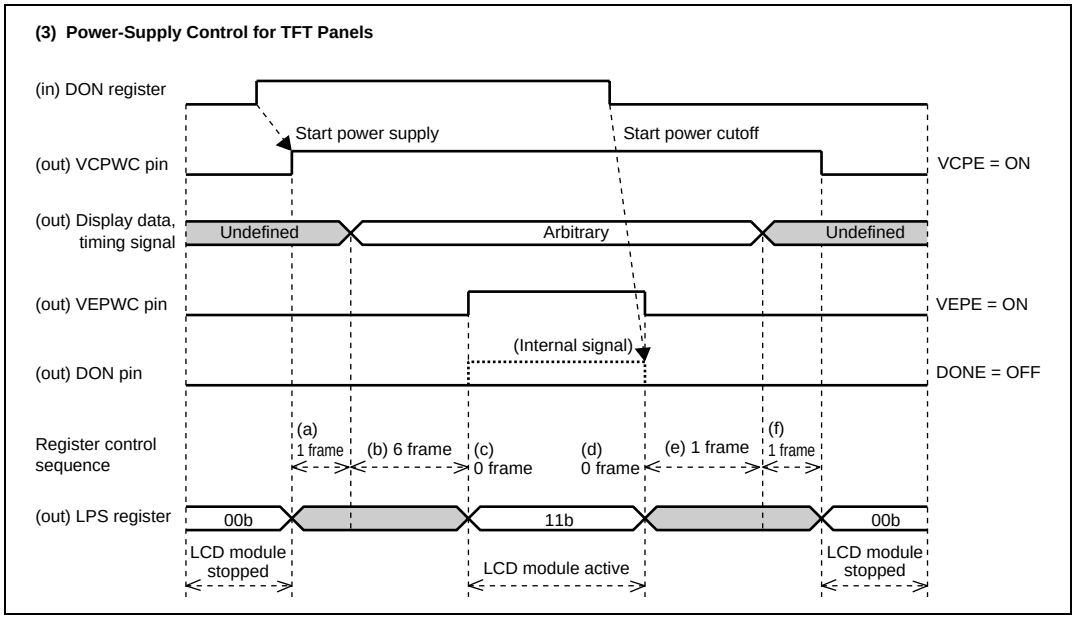


Figure 25.6 Power-Supply Control Sequence and States of the LCD Module

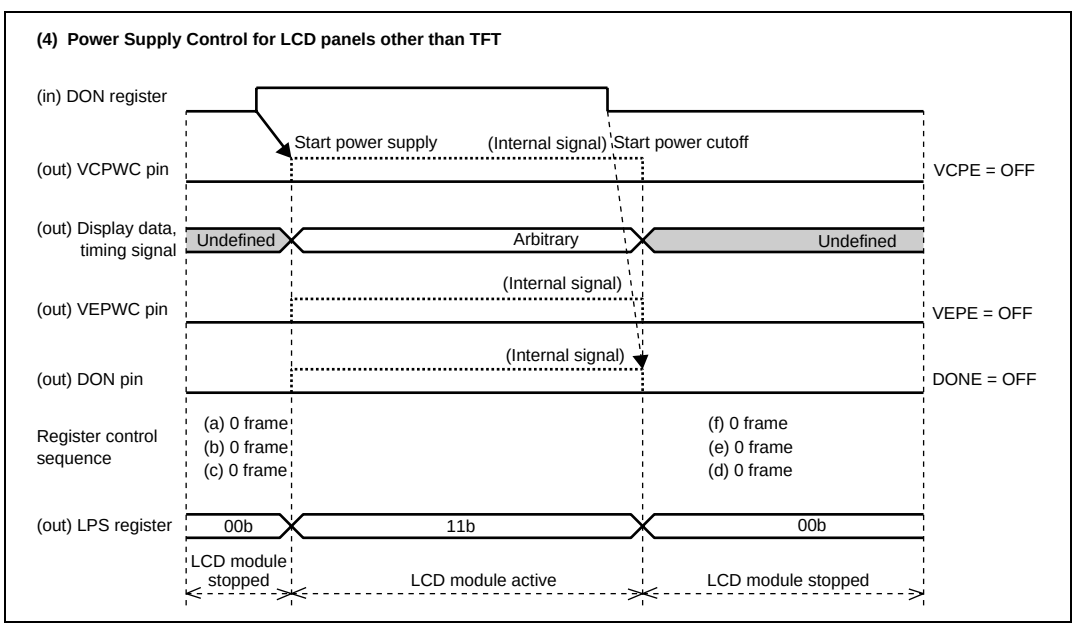


Figure 25.7 Power-Supply Control Sequence and States of the LCD Module

Table 25.5 Available Power-Supply Control-Sequence Periods at Typical Frame Rates

ONX, OFFX Register Value	Frame Rate	
	120 Hz	60 Hz
H'F	(–1+1)/120 = 0.00 (ms)	(–1+1)/60 = 0.00 (ms)
H'0	(0+1)/120 = 8.33 (ms)	(0+1)/60 = 16.67 (ms)
H'1	(1+1)/120 = 16.67 (ms)	(1+1)/60 = 33.33 (ms)
H'2	(2+1)/120 = 25.00 (ms)	(2+1)/60 = 50.00 (ms)
H'3	(3+1)/120 = 33.33 (ms)	(3+1)/60 = 66.67 (ms)
H'4	(4+1)/120 = 41.67 (ms)	(4+1)/60 = 83.33 (ms)
H'5	(5+1)/120 = 50.00 (ms)	(5+1)/60 = 100.00 (ms)
H'6	(6+1)/120 = 58.33 (ms)	(6+1)/60 = 116.67 (ms)
H'7	(7+1)/120 = 66.67 (ms)	(7+1)/60 = 133.33 (ms)
H'8	(8+1)/120 = 75.00 (ms)	(8+1)/60 = 150.00 (ms)
H'9	(9+1)/120 = 83.33 (ms)	(9+1)/60 = 166.67 (ms)
H'A	(10+1)/120 = 91.67 (ms)	(10+1)/60 = 183.33 (ms)
H'B	(11+1)/120 = 100.00 (ms)	(11+1)/60 = 200.00 (ms)
H'C	(12+1)/120 = 108.33 (ms)	(12+1)/60 = 216.67 (ms)
H'D	(13+1)/120 = 116.67 (ms)	(13+1)/60 = 233.33 (ms)
H'E	(14+1)/120 = 125.00 (ms)	(14+1)/60 = 250.00 (ms)

ONA, ONB, ONC, OFFD, OFFE, and OFFF are used to set the power-supply control-sequence periods, in units of frames, from 0 to 15. 1 is subtracted from each register. H'0 to H'E settings select from 1 to15 frames. The setting H'F selects 0 frames.

Actual sequence periods depend on the register values and the frame frequency of the display. The following table gives power-supply control-sequence periods for display frame frequencies used by typical LCD modules.

When ONB is Set to 6h and Display's Frame Frequency is 120 Hz: The display's frame frequency is 120 Hz. 1 frame period is thus 8.33 (ms) = 1/120 (sec).

The power-supply input sequence period is 7 frames because ONB setting is subtracted by 1.

As a result, the sequence period is 58.33 (ms) = 8.33 (ms) × 7.

Table 25.6 LCDC Operating Modes

Mode		Function
Display on (LCDC active)	Register setting: DON = 1 DON = 2	Fixed resolution, the format of the data for display is determined by the number of colors, timing signals are output to the LCD module.
Display off (LCDC stopped)	Register setting: DON = 0 DON2 = 0	Register access is enabled. Fixed resolution, the format of the data for display is determined by the number of colors, timing signals are not output to the LCD module.

Table 25.7 LCD Module Power-Supply States

(STN, DSTN module)

State	Power Supply for Logic	Display Data, Timing Signal	Power Supply for High-Voltage Systems	DON Signal
Control Pin	VCPWC	CL2, CL1, FLM, M/DISP, LCD	VEPWC	DON
Operating State	Supply	Supply	Supply	Supply
(Transitional State)	Supply	Supply	Supply	
	Supply	Supply		
	Supply			
Stopped State				

(TFT module)

State	Power Supply for Logic	Display Data, Timing Signal	Power Supply for High-Voltage Systems
Control Pin	VCPWC	CL2, CL1, FLM, M/DISP, LCD	VEPWC
Operating State	Supply	Supply	Supply
(Transitional State)	Supply	Supply	
	Supply		
Stopped State			

The above table shows the states of the power supply, display data, and timing signals for the typical LCD module in its active and stopped states. Some of the supply voltages described may not be necessary, because some modules internally generate the power supply required for high-voltage systems from the logic-level power-supply voltage.

Notes on display-off mode (LCDC stopped):

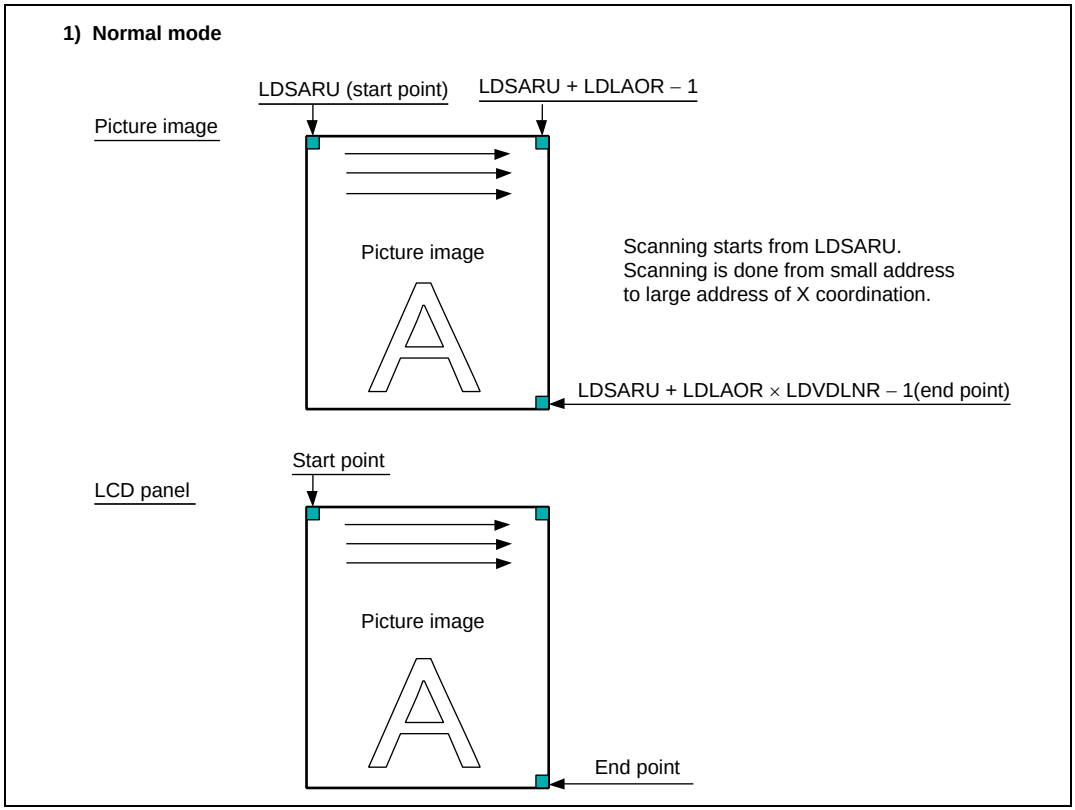
If LCD-module power-supply control-sequence processing is in use by the LCDC or the supply of power is cut off while the LCDC is in its display-on mode, normal operation is not guaranteed. In the worst case, the connected LCD module may be damaged.

25.3.7 Operation for Hardware Rotation

Operation in hardware-rotation mode is described below. Hardware-rotation mode can be thought of as using a landscape-format LCD panel instead of a portrait-format LCD panel by placing the landscape-format LCD panel as if it were a portrait-format panel. Whether the panel is intended for use in landscape or portrait format is thus no problem. The panel must, however, be within 320 pixels wide.

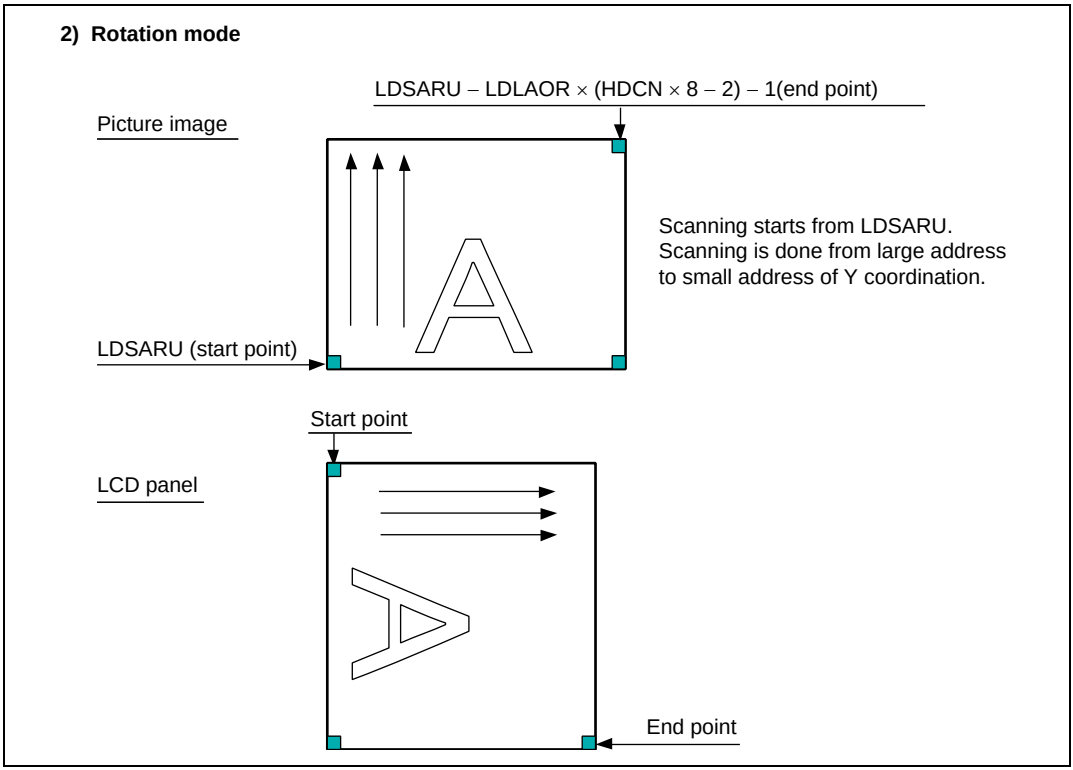
When making settings for hardware rotation, the following five differences from the setting for no hardware rotation must be noted. (The following example is for a display at 8 bpp. At 16 bpp, the amount of memory per dot will be doubled. The image size and register values used for rotation will thus be different.)

1. The image data must be prepared for display in the rotated panel. (If 240 × 320 pixels will be required after rotation, 240 × 320 pixel image data must be prepared.)
2. The register settings for the address of the image data must be changed (LDSARU and LDLAOR).
3. LDLAOR should be power of 2 (when the horizontal width after rotation is 240 pixels, LDLAOR should be set to 256).
4. Graphics software should be set up for the number 3 setting.
5. LDSARU should be changed to represent the address of the data for the lower-left pixel of the image rather than of the data for the upper-left pixel of the image.



For example, the registers have been set up for the display of image data in landscape format (320 × 240), which starts from LDSARU = 0x0c001000, on a 320 × 240 LCD panel. The graphics driver software is complete. Some changes are required to apply hardware rotation and use the panel as a 240 × 320 display. If LDLAOR is 512, the graphics driver software uses this power of 2 as the offset for the calculation of the addresses of Y coordinates in the image data. Before setting ROT to 1, the image data must be redrawn to suit the 240 × 320 LCD panel. LDLAOR will then be 256 because the size has changed and the graphics driver software must be altered accordingly. The point that corresponds to LDSARU moves from the upper left to the lower left of the display, so LDSARU should be changed to 0x0c001000 + 256 * 319.

Note: Hardware rotation allows the use of an LCD panel that has been rotated by 90 degrees. The settings in relation to the LCD panel should match the settings for the LCD panel before rotation. Rotation is possible regardless of the drawing processing carried out by the graphics driver software. However, the sizes in the image data and address offset values which are managed by the graphics driver software must be altered.



25.4 Clock and LCD Data Signal Examples

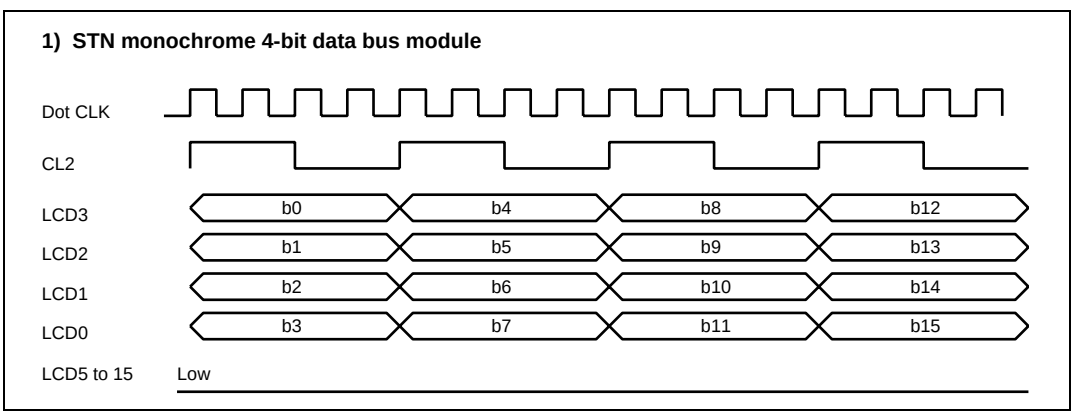


Figure 25.8 Clock and LCD Data Signal Example

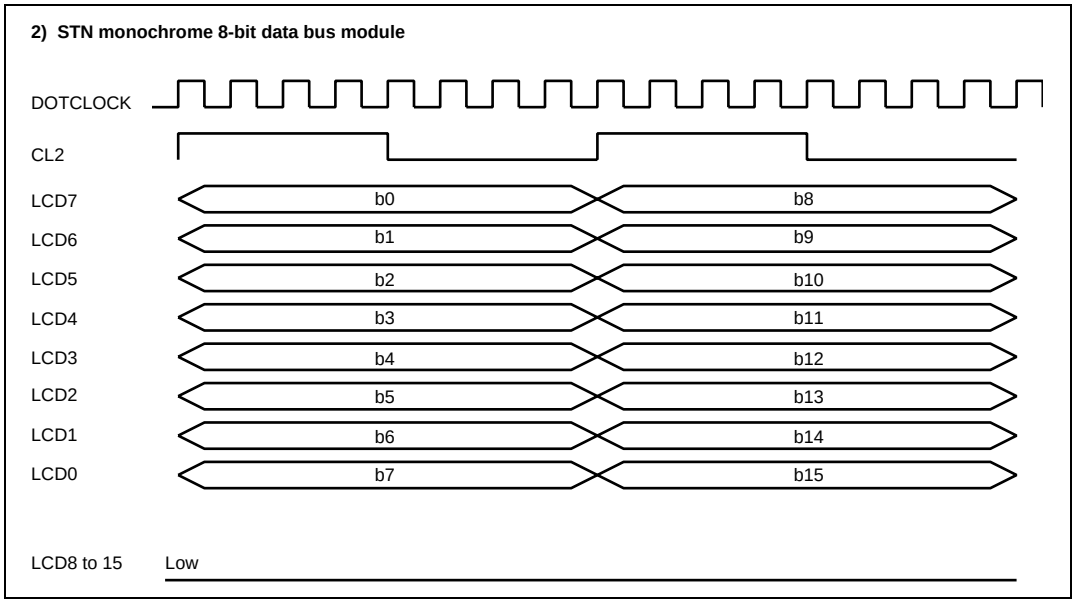


Figure 25.9 Clock and LCD Data Signal Example

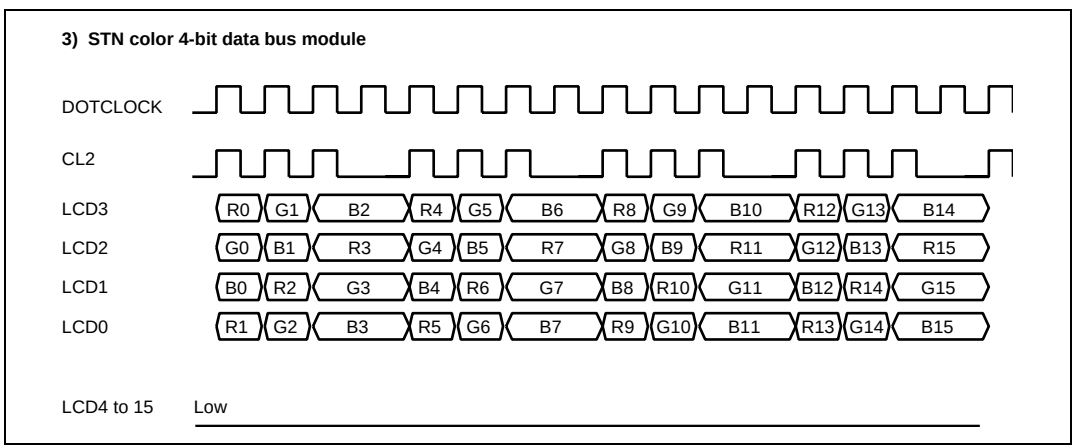


Figure 25.10 Clock and LCD Data Signal Example

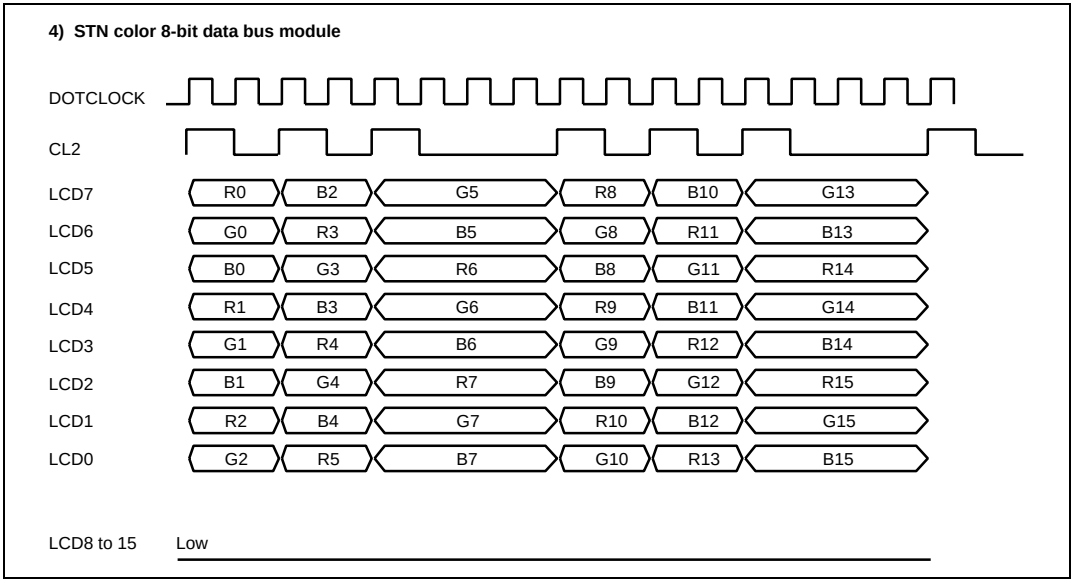


Figure 25.11 Clock and LCD Data Signal Example

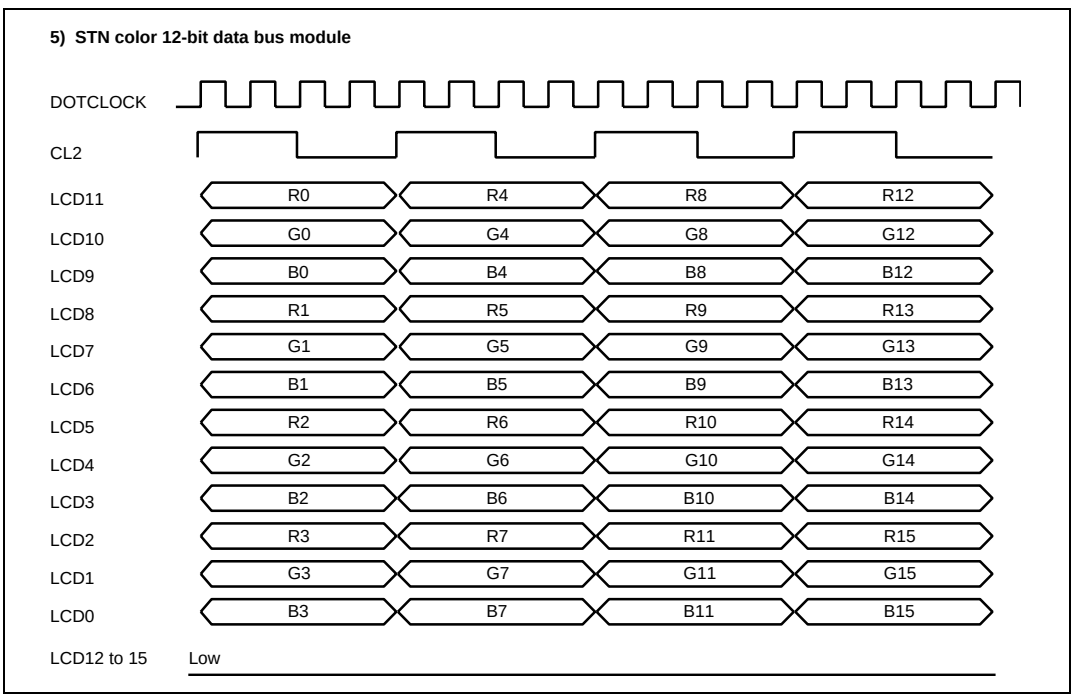


Figure 25.12 Clock and LCD Data Signal Example

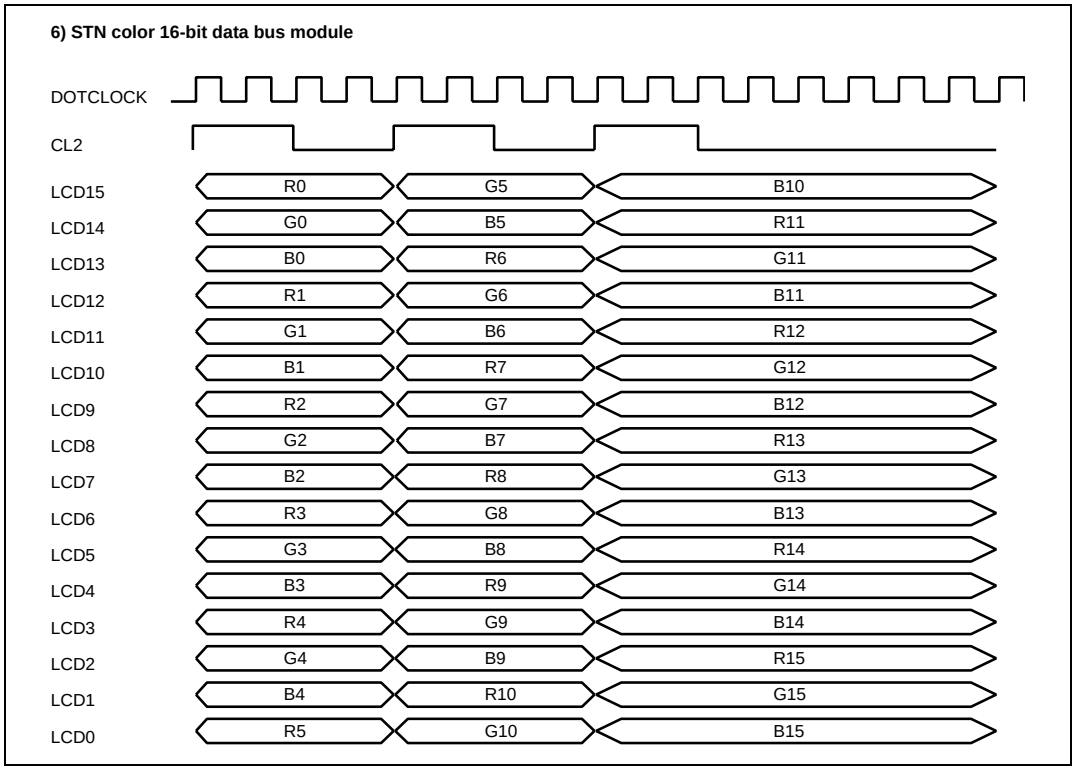


Figure 25.13 Clock and LCD Data Signal Example

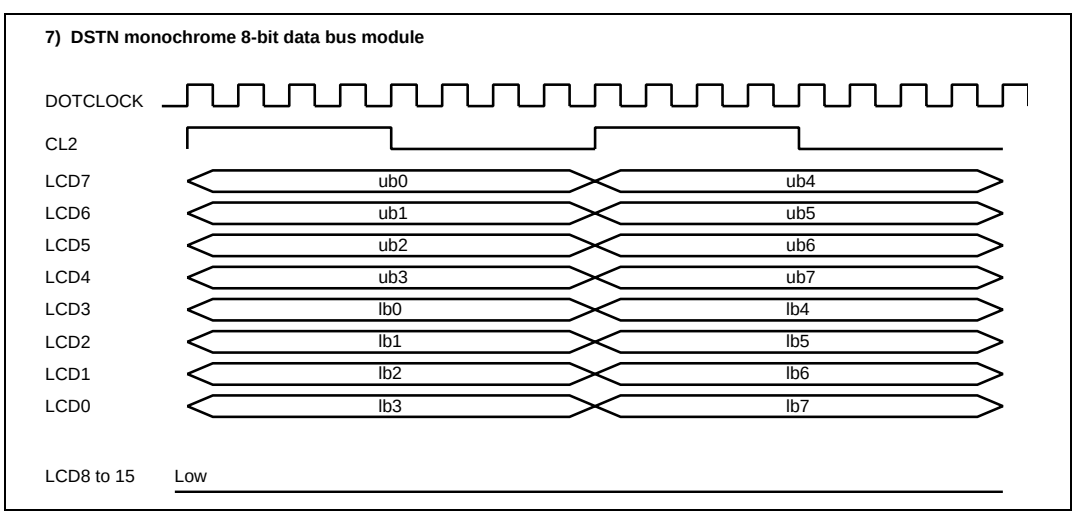


Figure 25.14 Clock and LCD Data Signal Example

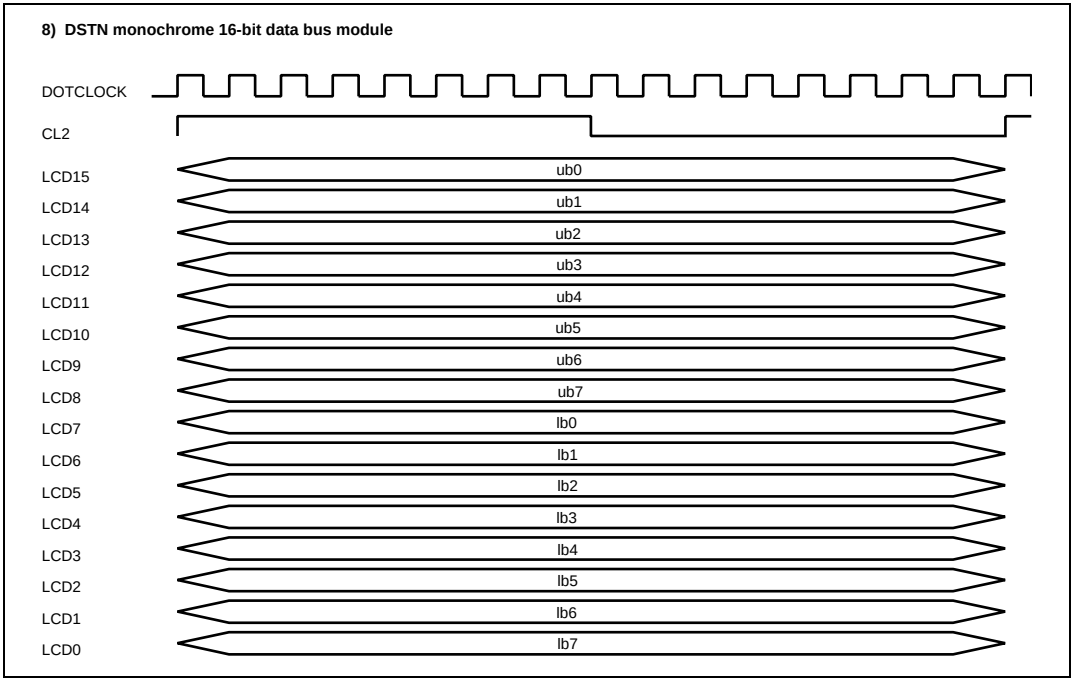


Figure 25.15 Clock and LCD Data Signal Example

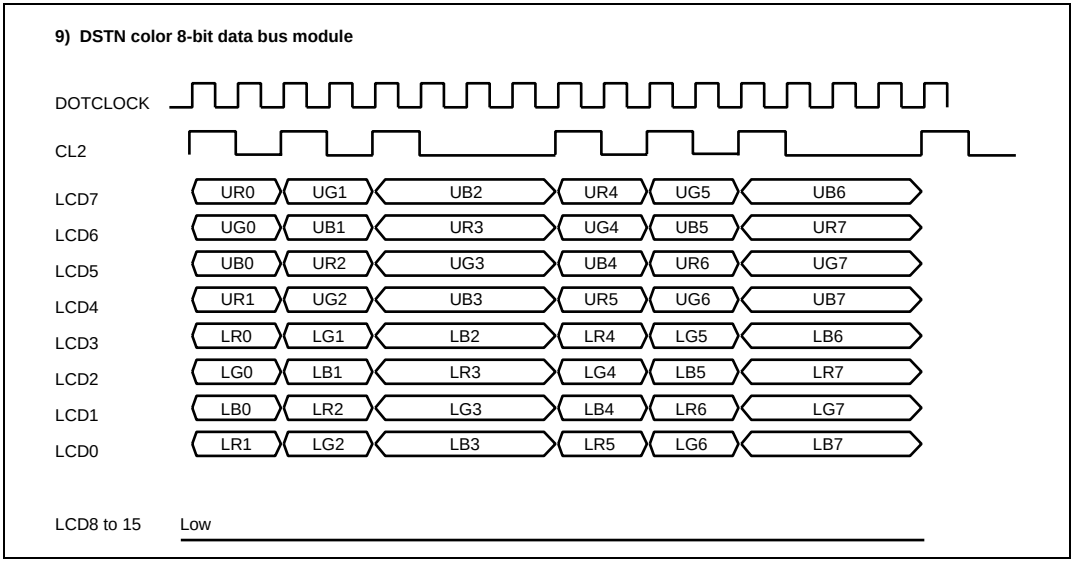


Figure 25.16 Clock and LCD Data Signal Example

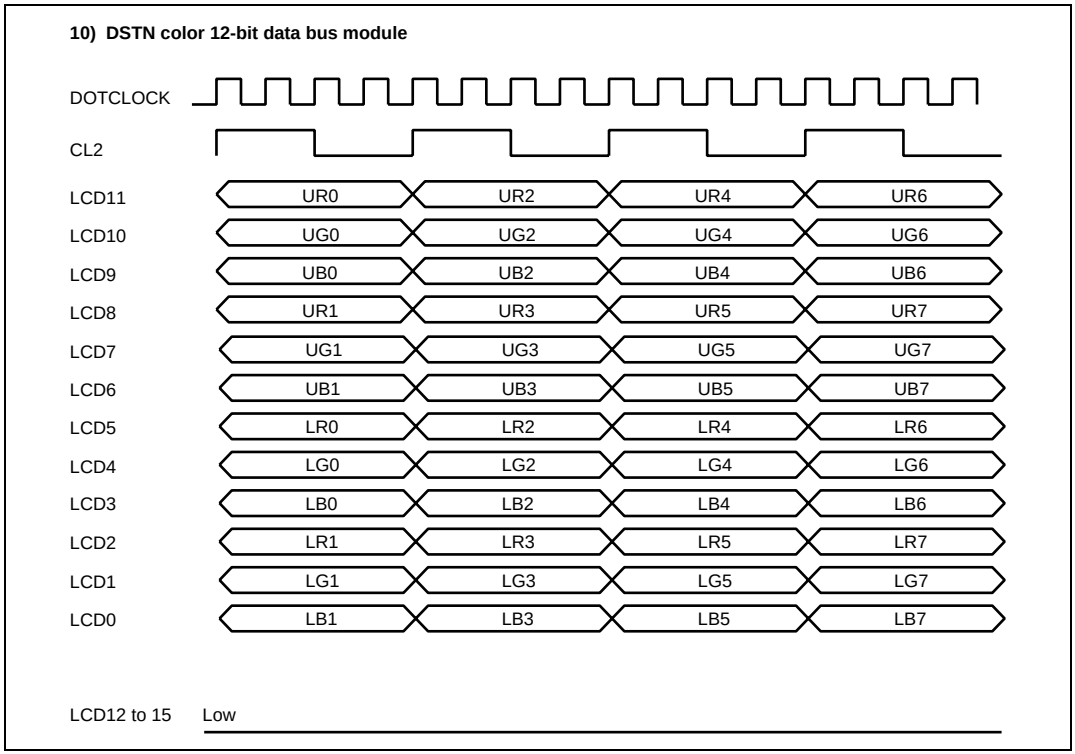


Figure 25.17 Clock and LCD Data Signal Example

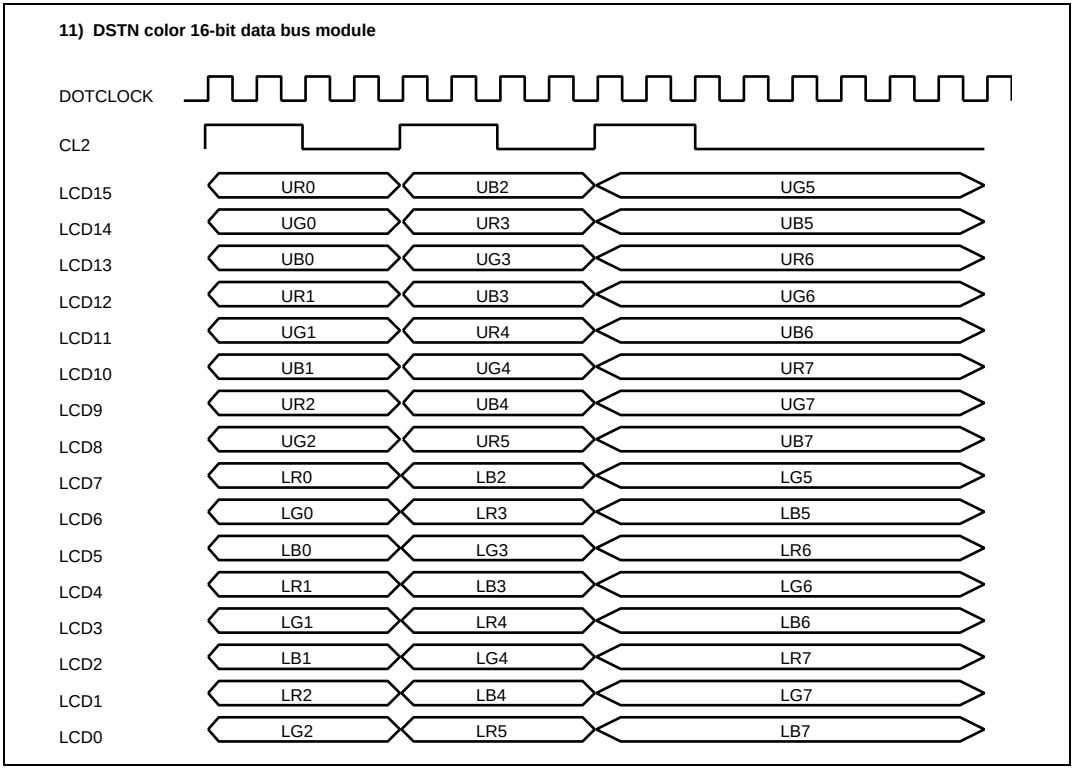


Figure 25.18 Clock and LCD Data Signal Example

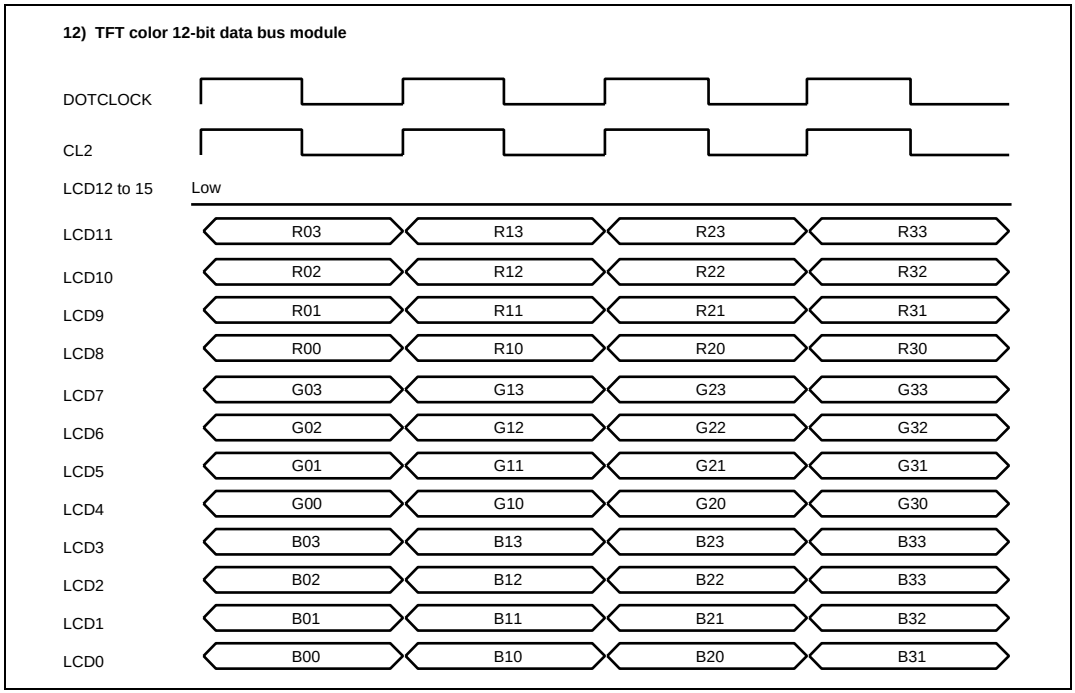


Figure 25.19 Clock and LCD Data Signal Example

13) TFT color 16-bit data bus module

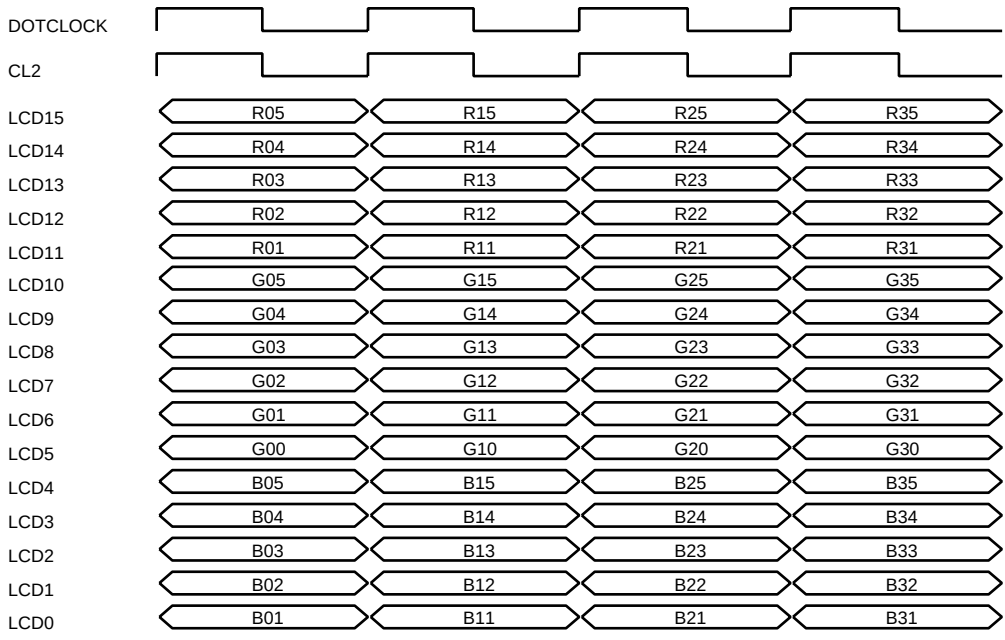


Figure 25.20 Clock and LCD Data Signal Example

14) 8-bit I/F color 640 × 840
STN-LCD

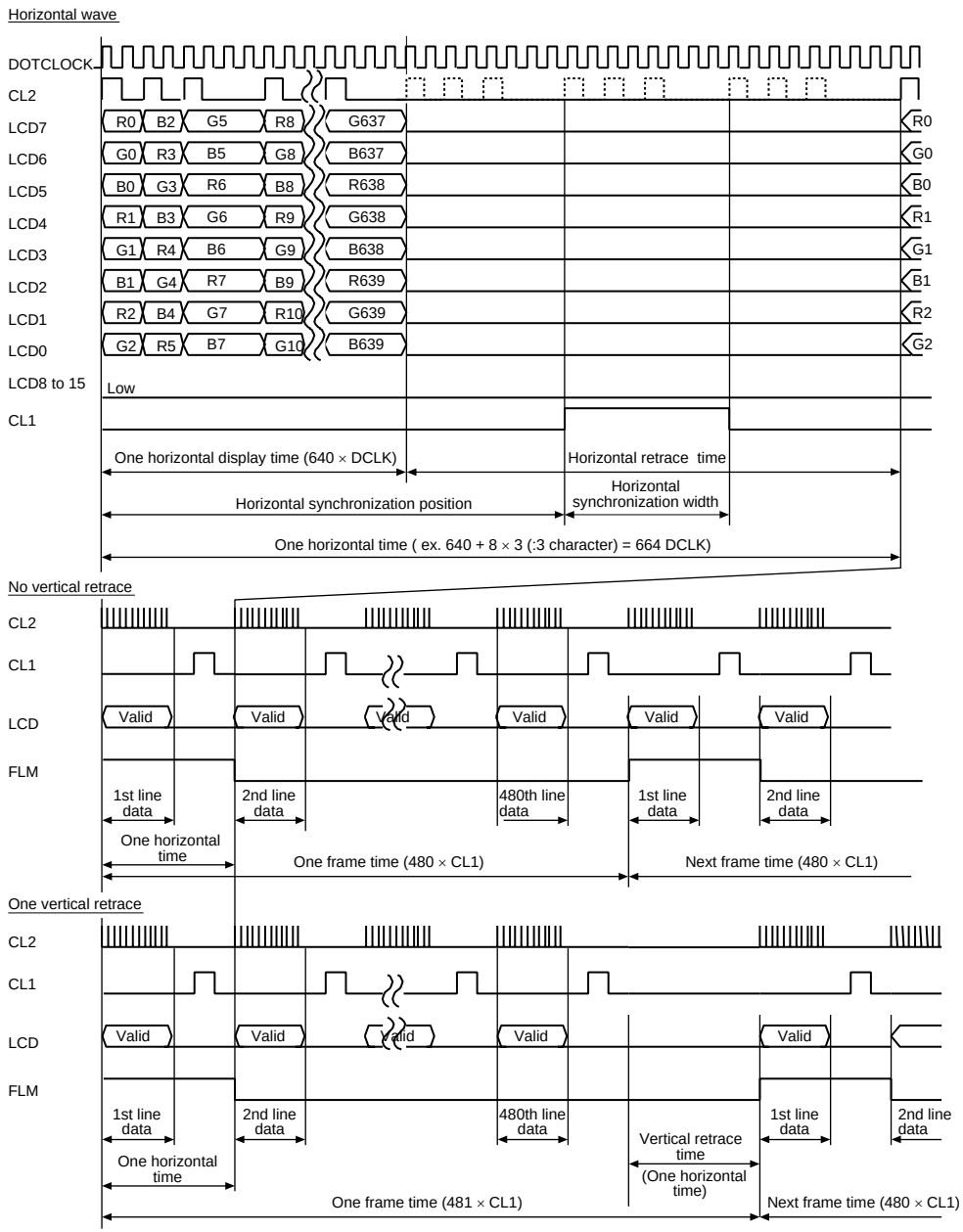


Figure 25.21 Clock and LCD Data Signal Example

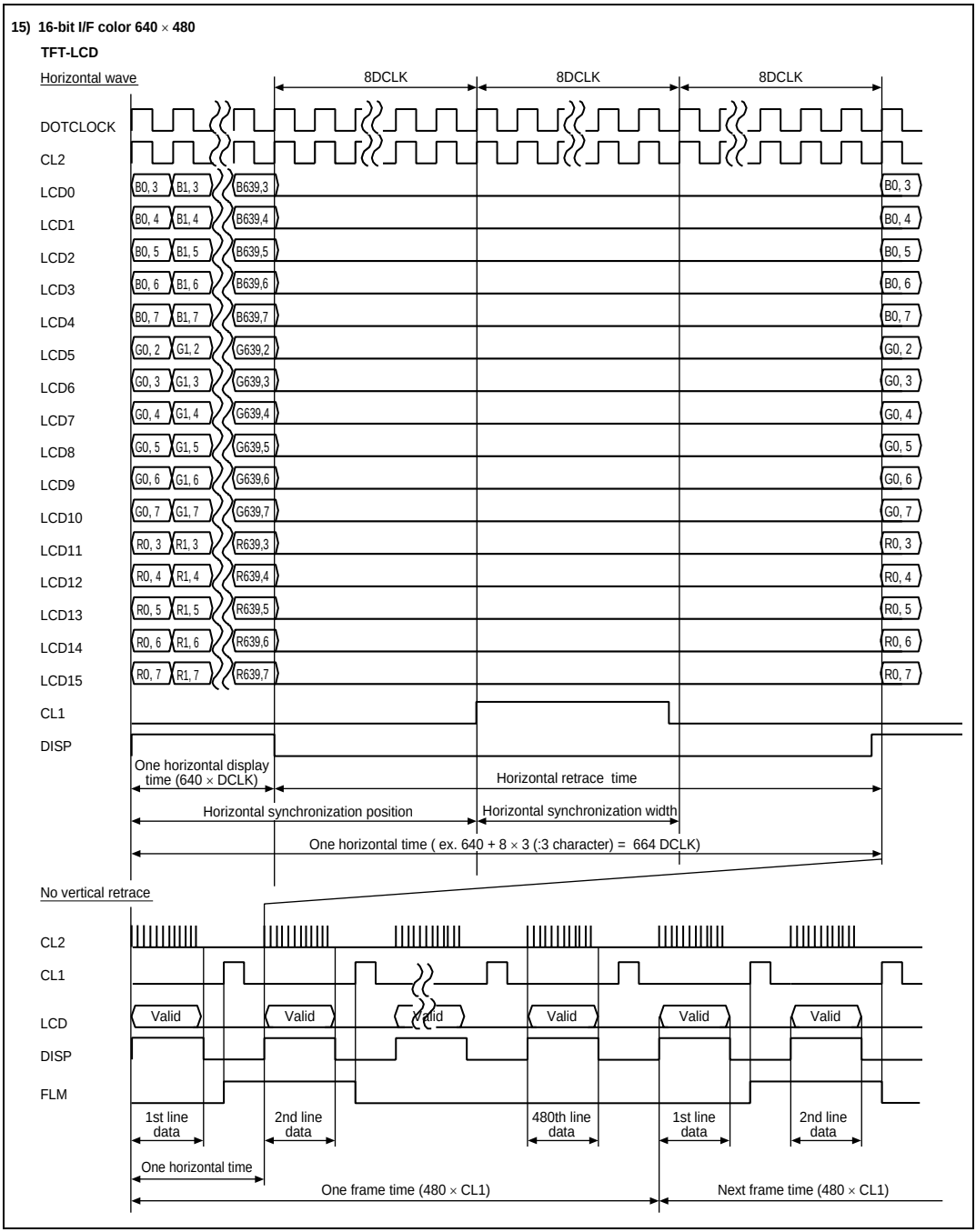


Figure 25.22 Clock and LCD Data Signal Example

25.5 Usage Notes

Note the following points when using the LCDC.

The following steps should be performed to prohibit access to the system memory used for LCDC module display (synchronous DRAM in area 3).

(1) Confirm that bits LPS1 and LPS0 in the LDPMMR register are set to 1.
(2) Clear the DON bit in LDCNTR to 0 (display off mode).
(3) Confirm that bits LPS1 and LPS0 in LDPMMR are cleared to 0.
(4) Wait the display duration of one frame.

The above steps to prohibit access are necessary before entering standby mode or using the LCDC module's standby function.

Section 26 Pin Function Controller (PFC)

26.1 Overview

The pin function of the SH7727 can be changed by the pin function controller (switch of the I/O port function), PC card controller (described in section 30), USB pin multiplexed controller (described in section 22), and user debugging interface (described in section 31). Figure 26.1 shows the overview of the pin selection function.

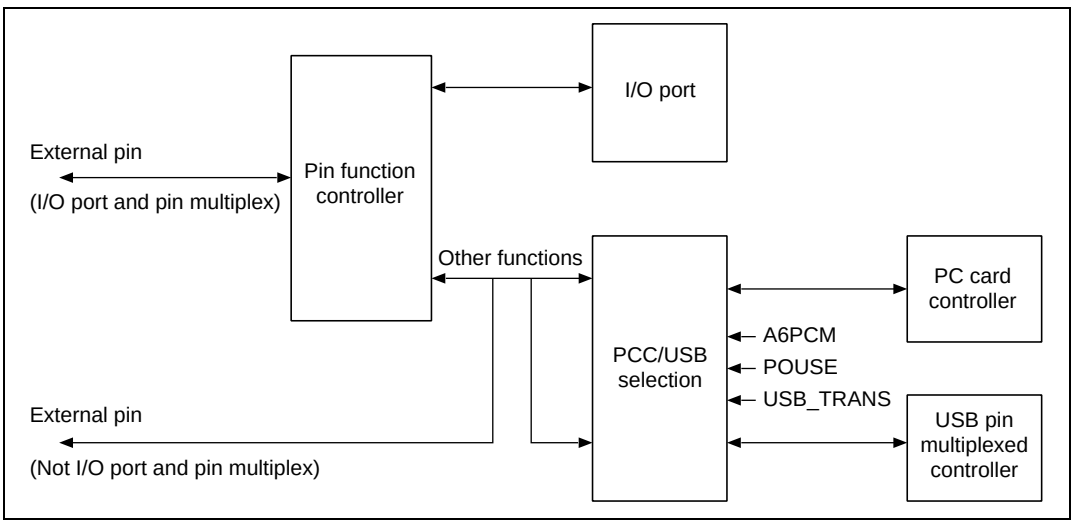


Figure 26.1 Overview of the Pin Selection Function

Table 26.1 List of Multiplexed Pins

Port	Port Function (Related Module)	Other Function 1 (Related Module)	Other Function 2 (Related Module)
A	PTA7 in/out (port)	D23 in/out (data bus)	
A	PTA6 in/out (port)	D22 in/out (data bus)	
A	PTA5 in/out (port)	D21 in/out (data bus)	
A	PTA4 in/out (port)	D20 in/out (data bus)	
A	PTA3 in/out (port)	D19 in/out (data bus)	
A	PTA2 in/out (port)	D18 in/out (data bus)	
A	PTA1 in/out (port)	D17 in/out (data bus)	
A	PTA0 in/out (port)	D16 in/out (data bus)	

Port	Port Function (Related Module)	Other Function 1 (Related Module)	Other Function 2 (Related Module)
B	PTB7 in/out (port)	D31 in/out (data bus)	
B	PTB6 in/out (port)	D30 in/out (data bus)	
B	PTB5 in/out (port)	D29 in/out (data bus)	
B	PTB4 in/out (port)	D28 in/out (data bus)	
B	PTB3 in/out (port)	D27 in/out (data bus)	
B	PTB2 in/out (port)	D26 in/out (data bus)	
B	PTB1 in/out (port)	D25 in/out (data bus)	
B	PTB0 in/out (port)	D24 in/out (data bus)	
C	PTC7 in/out (port)/PINT3 in (INTC)	LCD11 out (LCDC)	
C	PTC6 in/out (port)/PINT2 in (INTC)	LCD10 out (LCDC)	
C	PTC5 in/out (port)/PINT1 in (INTC)	LCD9 out (LCDC)	
C	PTC4 in/out (port)/PINT0 in (INTC)	LCD8 out (LCDC)	
C	PTC3 in/out (port)	LCD5 out (LCDC)	
C	PTC2 in/out (port)	LCD4 out (LCDC)	
C	PTC1 in/out (port)	LCD3 out (LCDC)	
C	PTC0 in/out (port)	LCD2 out (LCDC)	
D	PTD7 in/out (port)	DON out (LCDC)	
D	PTD6 in (port)	LCLK in (LCDC)/UCLK (USB)	
D	PTD5 in/out (port)	CL1 out (LCDC)	
D	PTD4 in (port)	$\overline{\text{DREQ0}}$ in (DMAC)	
D	PTD3 in/out (port)	LCD7 out (LCDC)	
D	PTD2 in/out (port)	LCD6 out (LCDC)	
D	PTD1 in/out (port)	LCD1 out (LCDC)	
D	PTD0 in/out (port)	LCD0 out (LCDC)	

Port	Port Function (Related Module)	Other Function 1 (Related Module)	Other Function 2 (Related Module)
E	PTE7 in/out (port)	$\overline{\text{AUDSYNC}}$ out (AUD)*3	PCC0RDY in (PCC)*2
E	PTE6 in/out (port)	M/DISP out (LCDC)	
E	PTE5 in/out (port)	$\overline{\text{CE2B}}$ out (PCMCIA)	
E	PTE4 in/out (port)	$\overline{\text{CE2A}}$ out (PCMCIA)	
E	PTE3 in/out (port)	FLM out (LCDC)	
E	PTE2 in/out (port)	USB1_pwr_en out (USB)	
E	PTE1 in/out (port)	USB2_pwr_en out (USB)	
E	PTE0 in/out (port)	TDO out (H-UDI)*2	
F	PTF7 in (port)/PINT15 in (INTC)	$\overline{\text{TRST}}$ in (AUD, H-UDI)*3	
F	PTF6 in (port)/PINT14 in (INTC)	TMS in (H-UDI)*3	
F	PTF5 in (port)/PINT13 in (INTC)	TD1 in (H-UDI)*3	
F	PTF4 in (port)/PINT12 in (INTC)	TCK in (H-UDI)*3	
F	PTF3 in (port)/PINT11 in (INTC)	Reserved	
F	PTF2 in (port)	Reserved	PCCOREG out (PCC)*2
F	PTF1 in (port)	Reserved	PCCOVS1 in (PCC)*2
F	PTF0 in (port)	Reserved	PCCOVS2 in (PCC)*2
G	PTG7 in (port)	$\overline{\text{IOIS16}}$ in (PCMCIA)	
G	PTG5 in (port)	$\overline{\text{ASEBRKAK}}$ out (AUD)*3	
G	PTG4 in (port)	—	
G	PTG3 in (port)	AUDATA3 out (AUD)*3	PCC0BVD2 (PCC)*2
G	PTG2 in (port)	AUDATA2 out (AUD)*3	PCC0BVD1 (PCC)*2
G	PTG1 in (port)	AUDATA1 out (AUD)*3	$\overline{\text{PCC0CD2}}$ (PCC)*2
G	PTG0 in (port)	AUDATA0 out (AUD)*3	$\overline{\text{PCC0CD1}}$ (PCC)*2
H	PTH7 in/out (port)	CL2 out (LCDC)	
H	PTH6 in (port)	AUDCK in (AUD)*1	$\overline{\text{PCC0WAIT}}$ (PCC)
H	PTH5 in (port)	$\overline{\text{ADTRG}}$ in (ADC)	
H	PTH4 in (port)/IRQ4 in (INTC)	IRQ4 in (INTC)	
H	PTH3 in (port)/IRQ3 in $\overline{\text{IRL3}}$ in (INTC)	IRQ3 in $\overline{\text{IRL3}}$ in (INTC)	
H	PTH2 in (port)/IRQ2 in $\overline{\text{IRL2}}$ in (INTC)	IRQ2 in $\overline{\text{IRL2}}$ in (INTC)	
H	PTH1 in (port)/IRQ1 in $\overline{\text{IRL1}}$ in (INTC)	IRQ1 in $\overline{\text{IRL1}}$ in (INTC)	
H	PTH0 in (port)/IRQ0 in $\overline{\text{IRL0}}$ in (INTC)	IRQ0 in $\overline{\text{IRL0}}$ in (INTC)	

Port	Port Function (Related Module)	Other Function 1 (Related Module)	Other Function 2 (Related Module)
J	PTJ7 in/out (port)	STATUS1 out (CPG)	
J	PTJ6 in/out (port)	STATUS0 out (CPG)	
J	PTJ5 in/out (port)	Reserved	
J	PTJ4 in/out (port)	Reserved	
J	PTJ3 in/out (port)	Reserved	
J	PTJ2 in/out (port)	$\overline{\text{CAS}}$ out (BSC)	Reserved
J	PTJ1 in/out (port)	Reserved	
J	PTJ0 in/out (port)	$\overline{\text{RAS3}}$ out (BSC)	
K	PTK7 in/out (port)	$\overline{\text{WE3}}$ out (BSC)/DQMUU out (BSC)/$\overline{\text{ICIOWR}}$ out (BSC)	
K	PTK6 in/out (port)	$\overline{\text{WE2}}$ out (BSC)/DQMUL out (BSC)/$\overline{\text{ICIORD}}$ out (BSC)	
K	PTK5 in/out (port)	CKE out (BSC)	
K	PTK4 in/out (port)	$\overline{\text{BS}}$ out (BSC)	
K	PTK3 in/out (port)	$\overline{\text{CS5}}$ out (BSC)/$\overline{\text{CE1A}}$ out (BSC)	
K	PTK2 in/out (port)	$\overline{\text{CS4}}$ out (BSC)	
K	PTK1 in/out (port)	$\overline{\text{AFE_RLYCNT}}$ out (AFE)	USB1d_DMNS in (USB)*2
K	PTK0 in/out (port)	AFE_HC1 out (AFE)	USB1d_DPLS in (USB)*2
L	PTL7 in (port)	AN7 in (ADC)/DA0 out (DAC)	
L	PTL6 in (port)	AN6 in (ADC)/DA1 out (DAC)	
L	PTL5 in (port)	AN5 in (ADC)	
L	PTL4 in (port)	AN4 in (ADC)	
L	PTL3 in (port)	AN3 in (ADC)	
L	PTL2 in (port)	AN2 in (ADC)	
M	PTM7 in (port)/PINT7 in (INTC)	AFE_FS in (AFE)	USB1d_RCV in (USB)*2
M	PTM6 in (port)/PINT6 in (INTC)	AFE_RXIN in (AFE)	USB1d_SPEED out (USB)*2
M	PTM5 in (port)/PINT5 in (INTC)	AFE_TXOUT out (AFE)	USB1d_TXSE0 out (USB)*2
M	PTM4 in (port)/PINT4 in (INTC)	$\overline{\text{AFE_RDET}}$ in (AFE)	
M	PTM3 in (port)/ PINT10 in (INTC)	LCD15 out (LCDC)	
M	PTM2 in (port)/PINT9 in (INTC)	LCD14 out (LCDC)	
M	PTM1 in (port)/PINT8 in (INTC)	LCD13 out (LCDC)	
M	PTM0 in (port)	LCD12 out (LCDC)	

Port	Port Function (Related Module)	Other Function 1 (Related Module)	Other Function 2 (Related Module)
SCPT	SCPT7 in (port)/IRQ5 in (INTC)	$\overline{\text{CTS2}}$ in (SCIF)/IRQ5 in (INTC)	
SCPT	SCPT6 in/out (port)	SIOFSYNC in/out (SIOF)	
SCPT	SCPT5 in/out (port)	SCK_SIO in/out (SIOF)	
SCPT	SCPT4 in (port)*1	RxD2 in (SCIF)	
	SCPT4 out (port)*1	TxD2 out (SCIF)	
SCPT	SCPT3 in/out (port)	SIOMCLK in (SIOF)	
SCPT	SCPT2 in (port)*1	RxD_SIO in (SIOF)	
	SCPT2 out (port)*1	TxD_SIO out (SIOF)	
SCPT	SCPT1 in/out (port)	SCK0 in/out (SCI)	
SCPT	SCPT0 in (port)*1	RxD0 in (SCI)	
	SCPT0 out (port)*1	TxD0 out (SCI)	
—	—	AFE_SCLK in (AFE)	USB1d_TXDPLS out (USB)
—	—	Reserved	USB1d_SUSPEND out (USB)
—	—	$\overline{\text{RTS2}}$ out (SCIF)	USB1d_TXENL out (USB)
—	—	DRAK0 out (DMAC)	PCC0RESET out (PCC)
—	—	DACK0 out (DMAC)	$\overline{\text{PCC0DRV}}$ out (PCC)

Notes: 1. SCPT0, SCPT2, and SCPT4 are different input pins and output pins, but the accessed data register is the same.

2. For pins with which PCC or USB pin multiplex controller related pins are multiplexed, other functions (in normal operation) and other functions (with special settings) are switched according to the setting of the P0USE bit in PCC, the A6PCM bit in BSC, and a USB pin multiplex controller register setting.

   To enable a PCC pin, set A6PCM and P0USE to use of a PC card. Switching to the PCC pin is performed automatically according to the value of the two bits.

   For I/O ports with which PCC or USB pin multiplex controller related pins are multiplexed, do not change the P0USE bit in PCC, the A6PCM bit in BSC, or the USB pin multiplex controller register, after switching to the other function.

3. Enabled when $\overline{\text{ASEMD0}}$ is low.

26.2 Register Configuration

Table 26.2 summarizes the registers of the pin function controller (PFC).

Table 26.2 Pin Function Controller Registers

Name	Abbreviation	R/W	Initial Value	Address	Access Size
Port A control register	PACR	R/W	H'0000	H'04000100 (H'A4000100)*	16
Port B control register	PBCR	R/W	H'0000	H'04000102 (H'A4000102)*	16
Port C control register	PCCR	R/W	H'AAAA	H'04000104 (H'A4000104)*	16
Port D control register	PDCR	R/W	H'AAAA	H'04000106 (H'A4000106)*	16
Port E control register	PECR	R/W	H'AAAA/H'2AA8	H'04000108 (H'A4000108)*	16
Port F control register	PFCR	R/W	H'AAAA/H'00AA	H'0400010A (H'A400010A)*	16
Port G control register	PGCR	R/W	H'AAAA/H'A200	H'0400010C (H'A400010C)*	16
Port H control register	PHCR	R/W	H'AAAA/H'8AAA	H'0400010E (H'A400010E)*	16
Port J control register	PJCR	R/W	H'0000	H'04000110 (H'A4000110)*	16
Port K control register	PKCR	R/W	H'0000	H'04000112 (H'A4000112)*	16
Port L control register	PLCR	R/W	H'0000	H'04000114 (H'A4000114)*	16
SC port control register	SCPCR	R/W	H'8008	H'04000116 (H'A4000116)*	16
Port M control register	PMCR	R/W	H'AAAA	H'04000118 (H'A4000118)*	16

Notes: These registers are located in area 1 of physical space. Therefore, when the cache is on, either access these registers from the P2 area of logical space or else make an appropriate setting using the MMU so that these registers are not cached.

If a low level is input at the $\overline{\text{ASEMD0}}$ pin while the $\overline{\text{RESETP}}$ pin is asserted, ASE mode is entered; if a high level is input, normal mode is entered. See section 31, User-Debugging Interface (H-UDI), for more information on the H-UDI.

The initial value of the port E, F, G, and H control registers depends on the state of the $\overline{\text{ASEMD0}}$ pin.

* When address translation by the MMU does not apply, the address in parentheses should be used.

26.3 Register Descriptions

26.3.1 Port A Control Register (PACR)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PA7 MD1	PA7 MD0	PA6 MD1	PA6 MD0	PA5 MD1	PA5 MD0	PA4 MD1	PA4 MD0	PA3 MD1	PA3 MD0	PA2 MD1	PA2 MD0	PA1 MD1	PA1 MD0	PA0 MD1	PA0 MD0
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

Port A Control Register (PACR) is a 16-bit read/write register that selects the pin functions. PACR is initialized to H'0000 by power-on resets; however, it is not initialized by manual resets, in standby mode, or in sleep mode.

Bits 15, 14: PA7 Mode 1, 0 (PA7MD1, PA7MD0)
Bits 13, 12: PA6 Mode 1, 0 (PA6MD1, PA6MD0)
Bits 11, 10: PA5 Mode 1, 0 (PA5MD1, PA5MD0)
Bits 9, 8: PA4 Mode 1, 0 (PA4MD1, PA4MD0)
Bits 7, 6: PA3 Mode 1, 0 (PA3MD1, PA3MD0)
Bits 5, 4: PA2 Mode 1, 0 (PA2MD1, PA2MD0)
Bits 3, 2: PA1 Mode 1, 0 (PA1MD1, PA1MD0)
Bits 1, 0: PA0 Mode 1, 0 (PA0MD1, PA0MD0)
These bits select the pin functions and the input pullup MOS control.

Bit (2n + 1)	Bit 2n		
PAnMD1	**PAnMD0**	**Pin Function**	
0	0	Other function (see table 26.1)	(Initial value)
0	1	Port output	
1	0	Port input (Pullup MOS: on)	
1	1	Port input (Pullup MOS: off)	

(n = 0 to 7)

26.3.2 Port B Control Register (PBCR)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PB7 MD1	PB7 MD0	PB6 MD1	PB6 MD0	PB5 MD1	PB5 MD0	PB4 MD1	PB4 MD0	PB3 MD1	PB3 MD0	PB2 MD1	PB2 MD0	PB1 MD1	PB1 MD0	PB0 MD1	PB0 MD0
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

Port B Control Register (PBCR) is a 16-bit read/write register that selects the pin functions. PBCR is initialized to H'0000 by power-on resets; however, it is not initialized by manual resets, in standby mode, or in sleep mode.

Bits 15, 14: PB7 Mode 1, 0 (PB7MD1, PB7MD0)
Bits 13, 12: PB6 Mode 1, 0 (PB6MD1, PB6MD0)
Bits 11, 10: PB5 Mode 1, 0 (PB5MD1, PB5MD0)
Bits 9, 8: PB4 Mode 1, 0 (PB4MD1, PB4MD0)
Bits 7, 6: PB3 Mode 1, 0 (PB3MD1, PB3MD0)
Bits 5, 4: PB2 Mode 1, 0 (PB2MD1, PB2MD0)
Bits 3, 2: PB1 Mode 1, 0 (PB1MD1, PB1MD0)
Bits 1, 0: PB0 Mode 1, 0 (PB0MD1, PB0MD0)
These bits select the pin functions and the input pullup MOS control.

Bit (2n + 1)	Bit 2n		
PBnMD1	**PBnMD0**	**Pin Function**	
0	0	Other function (see table 26.1)	(Initial value)
0	1	Port output	
1	0	Port input (Pullup MOS: on)	
1	1	Port input (Pullup MOS: off)	

(n = 0 to 7)

26.3.3 Port C Control Register (PCCR)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PC7 MD1	PC7 MD0	PC6 MD1	PC6 MD0	PC5 MD1	PC5 MD0	PC4 MD1	PC4 MD0	PC3 MD1	PC3 MD0	PC2 MD1	PC2 MD0	PC1 MD1	PC1 MD0	PC0 MD1	PC0 MD0
Initial value:	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	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

Port C Control Register (PCCR) is a 16-bit read/write register that selects the pin functions. PCCR is initialized to H'AAAA by power-on resets; however, it is not initialized by manual resets, in standby mode, or in sleep mode.

Bits 15, 14: PC7 Mode 1, 0 (PC7MD1, PC7MD0)
Bits 13, 12: PB6 Mode 1, 0 (PC6MD1, PC6MD0)
Bits 11, 10: PC5 Mode 1, 0 (PC5MD1, PC5MD0)
Bits 9, 8: PC4 Mode 1, 0 (PC4MD1, PC4MD0)
Bits 7, 6: PC3 Mode 1, 0 (PC3MD1, PC3MD0)
Bits 5, 4: PC2 Mode 1, 0 (PC2MD1, PC2MD0)
Bits 3, 2: PC1 Mode 1, 0 (PC1MD1, PC1MD0)
Bits 1, 0: PC0 Mode 1, 0 (PC0MD1, PC0MD0)
These bits select the pin functions and the input pullup MOS control.

Bit (2n + 1)	Bit 2n		
PCnMD1	**PCnMD0**	**Pin Function**	
0	0	Other function (see table 26.1)	
0	1	Port output	
1	0	Port input (Pullup MOS: on)	(Initial value)
1	1	Port input (Pullup MOS: off)	

(n = 0 to 7)

26.3.4 Port D Control Register (PDCR)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PD7 MD1	PD7 MD0	PD6 MD1	PD6 MD0	PD5 MD1	PD5 MD0	PD4 MD1	PD4 MD0	PD3 MD1	PD3 MD0	PD2 MD1	PD2 MD0	PD1 MD1	PD1 MD0	PD0 MD1	PD0 MD0
Initial value:	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	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

Port D Control Register (PDCR) is a 16-bit read/write register that selects the pin functions. PDCR is initialized to H'AAAA by power-on resets; however, it is not initialized by manual resets, in standby mode, or in sleep mode.

Bits 15, 14: PD7 Mode 1, 0 (PD7MD1, PD7MD0)
Bits 13, 12: PD6 Mode 1, 0 (PD6MD1, PD6MD0)
Bits 11, 10: PD5 Mode 1, 0 (PD5MD1, PD5MD0)
Bits 9, 8: PD4 Mode 1, 0 (PD4MD1, PD4MD0)
Bits 7, 6: PD3 Mode 1, 0 (PD3MD1, PD3MD0)
Bits 5, 4: PD2 Mode 1, 0 (PD2MD1, PD2MD0)
Bits 3, 2: PD1 Mode 1, 0 (PD1MD1, PD1MD0)
Bits 1, 0: PD0 Mode 1, 0 (PD0MD1, PD0MD0)
These bits select the pin functions and the input pullup MOS control.

Bit (2n + 1)	Bit 2n	
PDnMD1	**PDnMD0**	**Pin Function**
0	0	Other function (see table 26.1)
0	1	Port output (n = value other than 4 or 6), reserved (n = 4 or 6)
1	0	Port input (Pullup MOS: on) (Initial value)
1	1	Port input (Pullup MOS: off)

(n = 0 to 7)

26.3.5 Port E Control Register (PECR)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PE7 MD1	PE7 MD0	PE6 MD1	PE6 MD0	PE5 MD1	PE5 MD0	PE4 MD1	PE4 MD0	PE3 MD1	PE3 MD0	PE2 MD1	PE2 MD0	PE1 MD1	PE1 MD0	PE0 MD1	PE0 MD0
Initial value:	1/0	0	1	0	1	0	1	0	1	0	1	0	1	0	1/0	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

Port E Control Register (PECR) is a 16-bit read/write register that selects the pin functions. PECR is initialized to H'AAAA ($\overline{\text{ASEMD0}}$ = 1) or H'2AA8 ($\overline{\text{ASEMD0}}$ = 0) by power-on resets; however, it is not initialized by manual resets, in software standby mode, or in sleep mode.

Bits 15, 14: PE7 Mode 1, 0 (PE7MD1, PE7MD0)
Bits 13, 12: PE6 Mode 1, 0 (PE6MD1, PE6MD0)
Bits 11, 10: PE5 Mode 1, 0 (PE5MD1, PE5MD0)
Bits 9, 8: PE4 Mode 1, 0 (PE4MD1, PE4MD0)
Bits 7, 6: PE3 Mode 1, 0 (PE3MD1, PE3MD0)
Bits 5, 4: PE2 Mode 1, 0 (PE2MD1, PE2MD0)
Bits 3, 2: PE1 Mode 1, 0 (PE1MD1, PE1MD0)
Bits 1, 0: PE0 Mode 1, 0 (PE0MD1, PE0MD0)
These bits select the pin functions and the input pullup MOS control.

Bit (2n + 1)	Bit 2n		
PEnMD1	**PEnMD0**	**Pin Function**	
0	0	Reserved (n = 0, 7) (see table 26.1)	(Initial value) $\overline{\text{ASEMD0}}$ = 0
0	1	Port output	
1	0	Port input (Pullup MOS: on)	(Initial value) $\overline{\text{ASEMD0}}$ = 1
1	1	Port input (Pullup MOS: off)	

If $\overline{\text{ASEMD0}}$ = 0, port function cannot be selected with any PE7MD1, PE7MD0 value. (n = 0, 7)

Bit (2n + 1)	Bit 2n		
PEnMD1	**PEnMD0**	**Pin Function**	
0	0	Other function (see table 26.1)	
0	1	Port output	
1	0	Port input (Pullup MOS: on)	(Initial value)
1	1	Port input (Pullup MOS: off)	

(n = 1 to 6)

26.3.6 Port F Control Register (PFCR)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PF7 MD1	PF7 MD0	PF6 MD1	PF6 MD0	PF5 MD1	PF5 MD0	PF4 MD1	PF4 MD0	PF3 MD1	PF3 MD0	PF2 MD1	PF2 MD0	PF1 MD1	PF1 MD0	PF0 MD1	PF0 MD0
Initial value:	1/0	0	1/0	0	1/0	0	1/0	0	1	0	1	0	1	0	1	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

Port F Control Register (PFCR) is a 16-bit read/write register that selects the pin functions. PFCR is initialized to H'AAAA ($\overline{\text{ASEMD0}}$ = 1) or H'00AA ($\overline{\text{ASEMD0}}$ = 0) by power-on resets; however, it is not initialized by manual resets, in standby mode, or in sleep mode.

Bits 15, 14: PF7 Mode 1, 0 (PF7MD1, PF7MD0)
Bits 13, 12: PF6 Mode 1, 0 (PF6MD1, PF6MD0)
Bits 11, 10: PF5 Mode 1, 0 (PF5MD1, PF5MD0)
Bits 9, 8: PF4 Mode 1, 0 (PF4MD1, PF4MD0)
Bits 7, 6: PF3 Mode 1, 0 (PF3MD1, PF3MD0)
Bits 5, 4: PF2 Mode 1, 0 (PF2MD1, PF2MD0)
Bits 3, 2: PF1 Mode 1, 0 (PF1MD1, PF1MD0)
Bits 1, 0: PF0 Mode 1, 0 (PF0MD1, PF0MD0)
These bits select the pin functions and the input pullup MOS control.

Bit (2n + 1)	Bit 2n		
PFnMD1	**PFnMD0**	**Pin Function**	
0	0	Reserved (see table 26.1)	(Initial value) $\overline{\text{ASEMD0}}$ = 0
0	1	Reserved	
1	0	Port input (Pullup MOS: on)	(Initial value) $\overline{\text{ASEMD0}}$ = 1
1	1	Port input (Pullup MOS: off)	

(n = 4 to7)

Bit (2n + 1)	Bit 2n		
PFnMD1	**PFnMD0**	**Pin Function**	
0	0	Other function (see table 26.1)	
0	1	Reserved	
1	0	Port input (Pullup MOS: on)	(Initial value)
1	1	Port input (Pullup MOS: off)	

(n = 0 to 3)

26.3.7 Port G Control Register (PGCR)

For details on using versions previous to the SH7727B please refer to appendix F, Specifications for Using Port G Control Register (PGCR) with Versions Previous to the SH7727B.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PG7 MD1	PG7 MD0	—	—	PG5 MD1	PG5 MD0	PG4 MD1	PG4 MD0	PG3 MD1	PG3 MD0	PG2 MD1	PG2 MD0	PG1 MD1	PG1 MD0	PG0 MD1	PG0 MD0
Initial value:	1	0	1	0	1/0	0	1	0	1/0	0	1/0	0	1/0	0	1/0	0
R/W:	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Port G Control Register (PGCR) is a 16-bit read/write register that selects the pin functions. PGCR is initialized to H'AAAA ($\overline{\text{ASEMD0}}$ = 1) or H'A200 ($\overline{\text{ASEMD0}}$ = 0) by power-on resets; however, it is not initialized by manual resets, in standby mode, or in sleep mode.

Bits 15, 14: PG7 Mode 1, 0 (PG7MD1, PG7MD0)
Bits 13, 12: Reserved
Bits 11, 10: PG5 Mode 1, 0 (PG5MD1, PG5MD0)
Bits 9, 8: PG4 Mode 1, 0 (PG4MD1, PG4MD0)
Bits 7, 6: PG3 Mode 1, 0 (PG3MD1, PG3MD0)
Bits 5, 4: PG2 Mode 1, 0 (PG2MD1, PG2MD0)
Bits 3, 2: PG1 Mode 1, 0 (PG1MD1, PG1MD0)
Bits 1, 0: PG0 Mode 1, 0 (PG0MD1, PG0MD0)

These bits select the pin functions and the input pullup MOS control.

Bit (2n + 1)	Bit 2n	
PGnMD1	**PGnMD0**	**Pin Function**
0	0	Other function (n = 1, 2, 3, 5) (see table 26.1) (Initial value) $\overline{\text{ASEMD0}}$ = 0
0	1	Reserved
1	0	Port input (Pullup MOS: on) (Initial value) $\overline{\text{ASEMD0}}$ = 1
1	1	Port input (Pullup MOS: off)

(n = 0 to 3, 5)

Bit (2n + 1)	Bit 2n	
PGnMD1	**PGnMD0**	**Pin Function**
0	0	Other function (n = 7) (see table 26.1), Reserved (n = 4)
0	1	Reserved
1	0	Port input (Pullup MOS: on) (Initial value)
1	1	Port input (Pullup MOS: off)

(n = 4, 7)

26.3.8 Port H Control Register (PHCR)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PH7 MD1	PH7 MD0	PH6 MD1	PH6 MD0	PH5 MD1	PH5 MD0	PH4 MD1	PH4 MD0	PH3 MD1	PH3 MD0	PH2 MD1	PH2 MD0	PH1 MD1	PH1 MD0	PH0 MD1	PH0 MD0
Initial value:	1	0	1/0	0	1	0	1	0	1	0	1	0	1	0	1	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

Port H Control Register (PHCR) is a 16-bit read/write register that selects the pin functions. PHCR is initialized to H'AAAAA($\overline{\text{ASEMD0}}$ = 1) or H'8AAA ($\overline{\text{ASEMD0}}$ = 0) by power-on resets; however, it is not initialized by manual resets, in standby mode, or in sleep mode.

Bits 15, 14: PH7 Mode 1, 0 (PH7MD1, PH7MD0)
Bits 13, 12: PH6 Mode 1, 0 (PH6MD1, PH6MD0)
Bits 11, 10: PH5 Mode 1, 0 (PH5MD1, PH5MD0)
Bits 9, 8: PH4 Mode 1, 0 (PH4MD1, PH4MD0)
Bits 7, 6: PH3 Mode 1, 0 (PH3MD1, PH3MD0)
Bits 5, 4: PH2 Mode 1, 0 (PH2MD1, PH2MD0)
Bits 3, 2: PH1 Mode 1, 0 (PH1MD1, PH1MD0)
Bits 1, 0: PH0 Mode 1, 0 (PH0MD1, PH0MD0)
These bits select the pin functions and the input pullup MOS control.

Bit 15	Bit 14		
PH7MD1	**PH7MD0**	**Pin Function**	
0	0	Other function (see table 26.1)	
0	1	Port output	
1	0	Port input (Pullup MOS: on)	(Initial value)
1	1	Port input (Pullup MOS: off)	

Bit 13	Bit 12		
PH6MD1	**PH6MD0**	**Pin Function**	
0	0	Other function (see table 26.1)	(Initial value) $\overline{\text{ASEMD0}}$ = 0
0	1	Reserved	
1	0	Port input (Pullup MOS: on)	(Initial value) $\overline{\text{ASEMD0}}$ = 1
1	1	Port input (Pullup MOS: off)	

Bit (2n + 1)	Bit 2n		
PHnMD1	**PHnMD0**	**Pin Function**	
0	0	Other function (see table 26.1)	
0	1	Reserved	
1	0	Port input (Pullup MOS: on)	(Initial value)
1	1	Port input (Pullup MOS: off)	

(n = 0 to 5)

26.3.9 Port J Control Register (PJCR)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PJ7 MD1	PJ7 MD0	PJ6 MD1	PJ6 MD0	PJ5 MD1	PJ5 MD0	PJ4 MD1	PJ4 MD0	PJ3 MD1	PJ3 MD0	PJ2 MD1	PJ2 MD0	PJ1 MD1	PJ1 MD0	PJ0 MD1	PJ0 MD0
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

Port J Control Register (PJCR) is a 16-bit read/write register that selects the pin functions. PJCR is initialized to H'0000 by power-on resets; however, it is not initialized by manual resets, in standby mode, or in sleep mode.

Bits 15, 14: PJ7 Mode 1, 0 (PJ7MD1, PJ7MD0)
Bits 13, 12: PJ6 Mode 1, 0 (PJ6MD1, PJ6MD0)
Bits 11, 10: PJ5 Mode 1, 0 (PJ5MD1, PJ5MD0)
Bits 9, 8: PJ4 Mode 1, 0 (PJ4MD1, PJ4MD0)
Bits 7, 6: PJ3 Mode 1, 0 (PJ3MD1, PJ3MD0)
Bits 5, 4: PJ2 Mode 1, 0 (PJ2MD1, PJ2MD0)
Bits 3, 2: PJ1 Mode 1, 0 (PJ1MD1, PJ1MD0)
Bits 1, 0: PJ0 Mode 1, 0 (PJ0MD1, PJ0MD0)
These bits select the pin functions and the input pullup MOS control.

Bit (2n + 1)	Bit 2n		
PJnMD1	**PJnMD0**	**Pin Function**	
0	0	Other function (see table 26.1)	(Initial value)
0	1	Port output	
1	0	Port input (Pullup MOS: on)	
1	1	Port input (Pullup MOS: off)	

(n = 0 to 7)

26.3.10 Port K Control Register (PKCR)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PK7 MD1	PK7 MD0	PK6 MD1	PK6 MD0	PK5 MD1	PK5 MD0	PK4 MD1	PK4 MD0	PK3 MD1	PK3 MD0	PK2 MD1	PK2 MD0	PK1 MD1	PK1 MD0	PK0 MD1	PK0 MD0
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

Port K Control Register (PKCR) is a 16-bit read/write register that selects the pin functions. PKCR is initialized to H'0000 by power-on resets; however, it is not initialized by manual resets, in standby mode, or in sleep mode.

Bits 15, 14: PK7 Mode 1, 0 (PK7MD1, PK7MD0)
Bits 13, 12: PK6 Mode 1, 0 (PK6MD1, PK6MD0)
Bits 11, 10: PK5 Mode 1, 0 (PK5MD1, PK5MD0)
Bits 9, 8: PK4 Mode 1, 0 (PK4MD1, PK4MD0)
Bits 7, 6: PK3 Mode 1, 0 (PK3MD1, PK3MD0)
Bits 5, 4: PK2 Mode 1, 0 (PK2MD1, PK2MD0)
Bits 3, 2: PK1 Mode 1, 0 (PK1MD1, PK1MD0)
Bits 1, 0: PK0 Mode 1, 0 (PK0MD1, PK0MD0)
These bits select the pin functions and the input pullup MOS control.

Bit (2n + 1)	Bit 2n		
PKnMD1	**PKnMD0**	**Pin Function**	
0	0	Other function (see table 26.1)	(Initial value)
0	1	Port output	
1	0	Port input (Pullup MOS: on)	
1	1	Port input (Pullup MOS: off)	

(n = 0 to 7)

26.3.11 Port L Control Register (PLCR)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PL7 MD1	PL7 MD0	PL6 MD1	PL6 MD0	PL5 MD1	PL5 MD0	PL4 MD1	PL4 MD0	PL3 MD1	PL3 MD0	PL2 MD1	PL2 MD0	—	—	—	—
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	R	R	R

Port L Control Register (PLCR) is a 16-bit read/write register that selects the pin functions. PLCR is initialized to H'0000 by power-on resets; however, it is not initialized by manual resets, in standby mode, or in sleep mode.

Bits 15, 14: PL7 Mode 1, 0 (PL7MD1, PL7MD0)
Bits 13, 12: PL6 Mode 1, 0 (PL6MD1, PL6MD0)
Bits 11, 10: PL5 Mode 1, 0 (PL5MD1, PL5MD0)
Bits 9, 8: PL4 Mode 1, 0 (PL4MD1, PL4MD0)
Bits 7, 6: PL3 Mode 1, 0 (PL3MD1, PL3MD0)
Bits 5, 4: PL2 Mode 1, 0 (PL2MD1, PL2MD0)
Bits 3 to 0: Reserved
These bits select the pin functions and the input pullup MOS control.

Bit (2n + 1)	Bit 2n		
PLnMD1	**PLnMD0**	**Pin Function**	
0	0	Other function (see table 26.1)	(Initial value)
0	1	Reserved	
1	*	Port input	

(n = 2 to 7)

When the DA0 and DA1 pins are used as the D/A converter outputs or when PTL7 and PTL6 are used as the other function states, PLCR should remain at its initial value.

26.3.12 Port M Control Register (PMCR)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PM7 MD1	PM7 MD0	PM6 MD1	PM6 MD0	PM5 MD1	PM5 MD0	PM4 MD1	PM4 MD0	PM3 MD1	PM3 MD0	PM2 MD1	PM2 MD0	PM1 MD1	PM1 MD0	PM0 MD1	P0 MD0
Initial value:	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	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

Port M Control Register (PMCR) is a 16-bit read/write register that selects the pin functions. PMCR is initialized to H'AAAA by power-on resets; however, it is not initialized by manual resets, in standby mode, or in sleep mode.

Bits 15, 14: PM7 Mode 1, 0 (PM7MD1, PM7MD0)
Bits 13, 12: PM6 Mode 1, 0 (PM6MD1, PM6MD0)
Bits 11, 10: PM5 Mode 1, 0 (PM5MD1, PM5MD0)
Bits 9, 8: PM4 Mode 1, 0 (PM4MD1, PM4MD0)
Bits 7, 6: PM3 Mode 1, 0 (PM3MD1, PM3MD0)
Bits 5, 4: PM2 Mode 1, 0 (PM2MD1, PM2MD0)
Bits 3, 2: PM1 Mode 1, 0 (PM1MD1, PM1MD0)
Bits 1, 0: PM0 Mode 1, 0 (PM0MD1, PM0MD0)
These bits select the pin functions and the input pullup MOS control.

Bit (2n + 1)	Bit 2n		
PMnMD1	**PMnMD0**	**Pin Function**	
0	0	Other function (see table 26.1)	
0	1	Reserved	
1	0	Port input (Pullup MOS: on)	(Initial value)
1	1	Port input (Pullup MOS: off)	

(n = 0 to 7)

26.3.13 SC Port Control Register (SCPCR)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SCP7MD1	SCP7MD0	SCP6MD1	SCP6MD0	SCP5MD1	SCP5MD0	SCP4MD1	SCP4MD0	SCP3MD1	SCP3MD0	SCP2MD1	SCP2MD0	SCP1MD1	SCP1MD0	SCP0MD1	SCP0MD0
Initial value:	1	0	0	0	0	0	0	0	0	0	0	0	1	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

SC Port Control Register (SCPCR) is a 16-bit read/write register that selects the pin functions. The setting of SCPCR is valid only when the transmit/receive operation is disabled in the setting of the SCSCR register. SCPCR is initialized to H'8008 by power-on resets; however, it is not initialized by manual resets, in standby mode, or in sleep mode. When the TE bit in SCSCR is set to 1, the other function output state has a higher priority than the SCPCR setting of the TxD[2:0] pin. When the RE bit in SCSCR is set to 1, the input state has a higher priority than the SCPCR setting of the RxD[2:0] pin.

Bits 15, 14—SCP7 Mode 1, 0 (SCP7MD1, SCP7MD0): These bits select the pin functions and the input pullup MOS control.

Bit 15	Bit 14		
SCP7MD1	**SCP7MD0**	**Pin Function**	
0	0	Other function (see table 26.1)	
0	1	Reserved	
1	0	Port input (Pullup MOS: on)	(Initial value)
1	1	Port input (Pullup MOS: off)	

Bits 13, 12—SCP6 Mode 1, 0 (SCP6MD1, SCP6MD0): These bits select the pin functions and the input pullup MOS control.

Bit 13	Bit 12		
SCP6MD1	**SCP6MD0**	**Pin Function**	
0	0	Other function (see table 26.1)	(Initial value)
0	1	Port output	
1	0	Port input (Pullup MOS: on)	
1	1	Port input (Pullup MOS: off)	

Bits 11, 10—SCP5 Mode 1, 0 (SCP5MD1, SCP5MD0): These bits select the pin functions and the input pullup MOS control.

Bit 11	Bit 10		
SCP5MD1	**SCP5MD0**	**Pin Function**	
0	0	Other function (see table 26.1)	(Initial value)
0	1	Port output	
1	0	Port input (Pullup MOS: on)	
1	1	Port input (Pullup MOS: off)	

Bits 9, 8—SCP4 Mode 1, 0 (SCP4MD1, SCP4MD0): These bits select the pin functions and the input pullup MOS control.

Bit 9	Bit 8		
SCP4MD1	**SCP4MD0**	**Pin Function**	
0	0	Transmit data output 2 (TxD2) Receive data input 2 (RxD2)	(Initial value)
0	1	General output (SCPT[4] output pin) Receive data input 2 (RxD2)	
1	0	SCPT[4] input pin pullup (input pin) Transmit data output 2 (TxD2)	
1	1	General input (SCPT[4] input pin) Transmit data output 2 (TxD2)	

Note: There is no combination of simultaneous I/O of SCPT[4] because one bit (SCP4DT) is accessed using two pins of TxD2 and RxD2.

When the port input is set (bit SCPnMD1 is set to 1) and when the TE bit in SCSCR is set to 1, the TxD2 pin is in the output state. When the TE bit is cleared to 0, the TxD2 pin is in the high-impedance state.

Bits 7, 6—SCP3 Mode 1, 0 (SCP3MD1, SCP3MD0): These bits select the pin functions and the input pullup MOS control.

Bit 7	Bit 6		
SCP3MD1	**SCP3MD0**	**Pin Function**	
0	0	Other function (see table 26.1)	(Initial value)
0	1	Port output	
1	0	Port input (Pullup MOS: on)	
1	1	Port input (Pullup MOS: off)	

Bits 5, 4—SCP2 Mode 1, 0 (SCP2MD1, SCP2MD0): These bits select the pin functions and the input pullup MOS control.

Bit 5	Bit 4		
SCP2MD1	**SCP2MD0**	**Pin Function**	
0	0	Transmit data output 1 (TxD_SIO) Receive data input 1 (RxD_SIO)	(Initial value)
0	1	General output (SCPT[2] output pin) Receive data input 1 (RxD_SIO)	
1	0	SCPT[2] input pin pullup (input pin) Transmit data output 1 (TxD_SIO)	
1	1	General input (SCPT[2] input pin) Transmit data output 1 (TxD_SIO)	

Note: There is no combination of simultaneous I/O of SCPT[2] because one bit (SCP2DT) is accessed using two pins of TxD_SIO and RxD_SIO.

When the port input is set (bit SCPnMD1 is set to 1) and when the TE bit in SCSCR is set to 1, the TxD_SIO pin is in the output state. When the TE bit is cleared to 0, the TxD_SIO pin is in the high-impedance state.

Bits 3, 2—SCP1 Mode 1, 0 (SCP1MD1, SCP1MD0): These bits select the pin functions and the input pullup MOS control.

Bit 3	Bit 2		
SCP1MD1	**SCP1MD0**	**Pin Function**	
0	0	Other function (see table 26.1)	
0	1	Port output	
1	0	Port input (Pullup MOS: on)	(Initial value)
1	1	Port input (Pullup MOS: off)	

Bits 1, 0—SCP0 Mode 1, 0 (SCP0MD1, SCP0MD0): These bits select the pin functions and the input pullup MOS control.

Bit 1	Bit 0		
SCP0MD1	**SCP0MD0**	**Pin Function**	
0	0	Transmit data output 0 (TxD0) Receive data input 0 (RxD0)	(Initial value)
0	1	General output (SCPT[0] output pin) Receive data input 0 (RxD0)	
1	0	SCPT[0] input pin pullup (input pin) Transmit data output 0 (TxD0)	
1	1	General input (SCPT[0] input pin) Transmit data output 0 (TxD0)	

Note: There is no combination of simultaneous I/O of SCPT[0] because one bit (SCP0DT) is accessed using two pins of TxD0 and RxD0.

When the port input is set (bit SCPnMD1 is set to 1) and when the TE bit in SCSCR is set to 1, the TxD0 pin is in the output state. When the TE bit is cleared to 0, the TxD0 pin is in the high-impedance state.

Section 27 I/O Ports

27.1 Overview

This LSI has thirteen 8-bit ports (ports A to M and SC). All port pins are multiplexed with other pin functions (Pin Function Controller (PFC) selects the pin functions and pullup MOS control). Each port has a data register which stores data for the pins.

27.2 Register Configuration

Table 27.1 summarizes the registers of the pin function controller.

Table 27.1 Pin Function Controller Registers

Name	Abbreviation	R/W	Initial Value	Address	Access Size
Port A data register	PADR	R/W	H'00	H'04000120 (H'A4000120)*1	8
Port B data register	PBDR	R/W	H'00	H'04000122 (H'A4000122)*1	8
Port C data register	PCDR	R/W	H'00	H'04000124 (H'A4000124)*1	8
Port D data register	PDDR	R/W or R	B'0*0*0000	H'04000126 (H'A4000126)*1	8
Port E data register	PEDR	R/W	H'00	H'04000128 (H'A4000128)*1	8
Port F data register	PFDR	R	H'**	H'0400012A (H'A400012A)*1	8
Port G data register	PGDR	R	H'**	H'0400012C (H'A400012C)*1	8
Port H data register	PHDR	R/W or R	B'0*******	H'0400012E (H'A400012E)*1	8
Port J data register	PJDR	R/W	H'00	H'04000130 (H'A4000130)*1	8
Port K data register	PKDR	R/W	H'00	H'04000132 (H'A4000132)*	8
Port L data register	PLDR	R	H'**	H'04000134 (H'A4000134)*1	8
SC port data register	SCPDR	R/W or R	B'*0000000	H'04000136 (H'A4000136)*1	8
Port M data register	PMDR	R	B'********	H'04000138 (H'A4000138)*1	8

Notes: These registers are located in area 1 of physical space. Therefore, when the cache is on, either access these registers from the P2 area of logical space or else make an appropriate setting using the MMU so that these registers are not cached.

* Means no value.

1. When address translation by the MMU does not apply, the address in parentheses should be used.

27.3 Ports A to C, E, J, K

Each pin has an input pullup MOS, which is controlled by Port A to C, E, J, K Control Register in PFC.

27.3.1 Ports A to C, E, J, K Data Rgister (PADR, PBDR, PCDR, PEDR, PJDR, PKDR)

Bit:	7	6	5	4	3	2	1	0
	Px7DT	Px6DT	Px5DT	Px4DT	Px3DT	Px2DT	Px1DT	Px0DT
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

Ports A to C, E, J, K Data Register (PADR, PBDR, PCDR, PEDR, PJDR, PKDR) is an 8-bit read/write register that stores data for pins PTx7 to PTx0. Px7DT to Px0DT bit corresponds to PTx7 to PTx0 pin. When the pin function is general output port, if the port is read, the value of the corresponding PADR, PBDR, PCDR, PEDR, PJDR and PKDR bit is returned directly. When the function is general input port, if the port is read, the corresponding pin level is read. Table 27.2 shows the function of PADR, PBDR, PCDR, PEDR, PJDR, PKDR.

PADR, PBDR, PCDR, PEDR, PJDR, PKDR is initialized to H'00 by a power-on reset. When $\overline{\text{ASEMD0}}$ is equal to 1, after PDCR and PEDR are initialized to H'00, the general input port function (pullup MOS: on) is set as the initial pin function, and the corresponding pin levels are fetched. It retains its previous value in standby mode and sleep mode, and by a manual reset.

Table 27.2 Read/Write Operation of the Ports A to C, E, J, K Data Register

PxnMD1	PxnMD0	Pin State	Read	Write
0	0	Other function	PxDR value	Value is written to PxDR, but does not affect pin state.
	1	Output	PxDR value	Write value is output from pin.
1	0	Input (Pullup MOS: on)	Pin state	Value is written to PxDR, but does not affect pin state.
	1	Input (Pullup MOS: off)	Pin state	Value is written to PxDR, but does not affect pin state.

(n = 0 to 7)

(x = A to C, E, J, K)

27.4 Port D

Each pin has an input pullup MOS, which is controlled by Port D Control Register (PDCR) in PFC.

27.4.1 Port D Data Register (PDDR)

Bit:	7	6	5	4	3	2	1	0
	PD7DT	PD6DT	PD5DT	PD4DT	PD3DT	PD2DT	PD1DT	PD0DT
Initial value:	0	*	0	*	0	0	0	0
R/W:	R/W	R	R/W	R	R/W	R/W	R/W	R/W

Note: * Undefined

Port D Data Register (PDDR) is a 6-bit read/write and 2-bit read register that stores data for pins PTD7 to PTD0. PD7DT to PD0DT bit corresponds to PTD7 to PTD0 pin. When the pin function is general output port, if the port is read, the value of the corresponding PDDR bit is returned directly. When the function is general input port, if the port is read, the corresponding pin level is read. Table 27.3 shows the function of PDDR.

PDDR is initialized to B'0*0*0000 by a power-on reset. After initialization, the general input port function (pullup MOS: on) is set as the initial pin function, and the corresponding pin levels are fetched. It retains its previous value in standby mode and sleep mode, and by a manual reset.

Table 27.3 Read/Write Operation of the Port D Data Register (PDDR)

PDnMD1	PDnMD0	Pin State	Read	Write
0	0	Other function	PDDR value	Value is written to PDDR, but does not affect pin state.
	1	Output	PDDR value	Write value is output from pin.
1	0	Input (Pullup MOS: on)	Pin state	Value is written to PDDR, but does not affect pin state.
	1	Input (Pullup MOS: off)	Pin state	Value is written to PDDR, but does not affect pin state.

(n = 0 to 3, 5, 7)

PDnMD1	PDnMD0	Pin State	Read	Write
0	0	Other function	H'00	Ignored (no affect on pin state)
	1	Reserved*	—	—
1	0	Input (Pullup MOS: on)	Pin state	Ignored (no affect on pin state)
	1	Input (Pullup MOS: off)	Pin state	Ignored (no affect on pin state)

Note: * Operation cannot be guaranteed when this bit it set to “reserved.” (n = 4, 6)

27.5 Ports F, M

Each pin has an input pullup MOS, which is controlled by Ports F, M Control Register (PFDR, PMDR) in PFC.

27.5.1 Ports F, M Data Register (PFDR, PMDR)

Bit:	7	6	5	4	3	2	1	0
	Px7DT	Px6DT	Px5DT	Px4DT	Px3DT	Px2DT	Px1DT	Px0DT
Initial value:	*	*	*	*	*	*	*	*
R/W:	R	R	R	R	R	R	R	R

Note: * Undefined

Ports F, M Data Register (PFDR, PMDR) is an 8-bit read register that stores data for pins PTx7 to PTx0. Px7DT to Px0DT bit corresponds to PTx7 to PTx0 pin. When the pin function is general input port, if the port is read, the corresponding pin level is read. Table 27.4 shows the function of PFDR and PMDR.

PFDR and PMDR are initialized by a power-on reset. After initialization, the general input port function (pullup MOS: on) is set as the initial pin function, and the corresponding pin levels are read.

Table 27.4 Read/Write Operation of the Ports F, M Data Register (PFDR, PMDR)

PxnMD1	PxnMD0	Pin State	Read	Write
0	0	Other function	H'00	Ignored (no affect on pin state)
	1	Reserved*	—	—
1	0	Input (Pullup MOS on)	Pin state	Ignored (no affect on pin state)
	1	Input (Pullup MOS off)	Pin state	Ignored (no affect on pin state)

Note: * Operation cannot be guaranteed when this bit it set to "reserved." (n = 0 to 7)
(x = F, M)

27.6 Port G

Each pin has an input pullup MOS, which is controlled by Port G Control Register (PGCR) in PFC.

27.6.1 Port G Data Register (PGDR)

Bit:	7	6	5	4	3	2	1	0
	PG7DT	—	PG5DT	PG4DT	PG3DT	PG2DT	PG1DT	PG0DT
Initial value:	*	*	*	*	*	*	*	*
R/W:	R	R	R	R	R	R	R	R

Note: * Undefined

Port G Data Register (PGDR) is an 8-bit read register that stores data for pins PTG7 and PTG5 to PTG0. PG7DT and PTG5DT to PG0DT bit corresponds to PTG7 and PTG5 to PTG0 pin. When the pin function is general input port, if the port is read, the corresponding pin level is read. Table 27.5 shows the function of PGDR.

When $\overline{\text{ASEMD0}}$ is equal to 1, after PGDR is initialized by a power-on reset, the general input port function (pullup MOS: on) is set as the initial pin function, and the corresponding pin levels are fetched.

Table 27.5 Read/Write Operation of the Port G Data Register (PGDR)

PGnMD1	PGnMD0	Pin State	Read	Write
0	0	Other function	H'00	Ignored (no affect on pin state)
	1	Reserved*	—	—
1	0	Input (Pullup MOS: on)	Pin state	Ignored (no affect on pin state)
	1	Input (Pullup MOS: off)	Pin state	Ignored (no affect on pin state)

Note: * Operation cannot be guaranteed when this bit it set to "reserved." (n = 0 to 5, 7)

27.7 Port H

Each pin has an input pullup MOS, which is controlled by Port H Control Register (PHCR) in PFC.

27.7.1 Port H Data Register (PHDR)

Bit:	7	6	5	4	3	2	1	0
	PH7DT	PH6DT	PH5DT	PH4DT	PH3DT	PH2DT	PH1DT	PH0DT
Initial value:	0	*	*	*	*	*	*	*
R/W:	R/W	R	R	R	R	R	R	R

Note: * Undefined

Port H Data Register (PHDR) is a 1-bit read/write and 7-bit read register that stores data for pins PTH7 to PTH0. PH7DT to PH0DT bit corresponds to PTH7 to PTH0 pin. When the pin function is general output port, if the port is read, the value of the corresponding PHDR bit is returned directly. When the function is general input port, if the port is read, the corresponding pin level is read. Table 27.6 shows the function of PHDR.

When $\overline{\text{ASEMD0}}$ is equal to 1, after PHDR is initialized to B'0******* by a power-on reset, the general input port function (pullup MOS: on) is set as the initial pin function, and the corresponding pin levels are fetched. It retains its previous value in standby mode and sleep mode, and by a manual reset.

Table 27.6 Read/Write Operation of the Port H Data Register (PHDR)

PHnMD1	PHnMD0	Pin State	Read	Write
0	0	Other function	PHDR value	Value is written to PHDR, but does not affect pin state.
	1	Output	PHDR value	Write value is output from pin.
1	0	Input (Pullup MOS: on)	Pin state	Value is written to PHDR, but does not affect pin state.
	1	Input (Pullup MOS: off)	Pin state	Value is written to PHDR, but does not affect pin state.

(n = 7)

PHnMD1	PHnMD0	Pin State	Read	Write
0	0	Other function	H'00	Ignored (no affect on pin state)
	1	Reserved*	—	—
1	0	Input (Pullup MOS: on)	Pin state	Ignored (no affect on pin state)
	1	Input (Pullup MOS: off)	Pin state	Ignored (no affect on pin state)

Note: * Operation cannot be guaranteed when this bit it set to “reserved.” (n = 0 to 6)

27.8 Port L

27.8.1 Port L Data Register (PLDR)

Bit:	7	6	5	4	3	2	1	0
	PL7DT	PL6DT	PL5DT	PL4DT	PL3DT	PL2DT	—	—
Initial value:	*	*	*	*	*	*	*	*
R/W:	R	R	R	R	R	R	R	R

Note: * Undefined

Port L Data Register (PLDR) is an 8-bit read register that stores data for pins PTL7 to PTL2. PL7DT to PL2DT bit corresponds to PTL7 to PTL2 pin. When the pin function is general input port, if the port is read, the corresponding pin level is read. Table 27.7 shows the function of PLDR.

PLDR is initialized to a power-on reset. It retains its previous value in software standby mode and sleep mode, and by a manual reset.

Table 27.7 Read/Write Operation of the Port L Data Register (PLDR)

PLnMD1	PLnMD0	Pin State	Read	Write
0	0	Other function	H'00	Ignored (no affect on pin state)
	1	Reserved*1	—	—
1	*	Input	Pin state	Ignored (no affect on pin state)

Notes: * Undefined (n = 2 to 7)

1. Operation cannot be guaranteed when this bit it set to “reserved.”

27.9 SC Port

Each pin has an input pullup MOS, which is controlled by SC port control register (SCPCR) in PFC.

27.9.1 Port SC Data Register (SCPDR)

Bit:	7	6	5	4	3	2	1	0
	SCP7DT	SCP6DT	SP5DT	SCP4DT	SCP3DT	SCP2DT	SCP1DT	SCP0DT
Initial value:	*	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: * Undefined

Port SC Data Register (SCPDR) is a 7-bit read/write and 1-bit read register that stores data for pins SCPT7 to SCPT0. SCP7DT to SCP0DT bit corresponds to SCPT7 to SCPT0 pin. When the pin function is general output port, if the port is read, the value of the corresponding SCPDR bit is returned directly. When the function is general input port, if the port is read, the corresponding pin level is read. Table 27.8 shows the function of SCPDR.

SCPDR is initialized to B'*0000000 by a power-on reset. After initialization, the general input port function (pullup MOS: on) is set as the initial pin function, and the corresponding pin levels are read from bits SCP7DT to SCP5DT, SCP3DT, and SCP1DT. It retains its previous value in standby mode and sleep mode, and by a manual reset.

Note that the low level is read if bit 7 is read except in general-purpose input.

Set the RE bit in SCSCR to 1, when reading RxD2 to RxD0 pin states of the SCP4DT, SCP2DT, and SCP0DT bits in SDPDR while the TE or RE bit in SCSCR is not cleared to 0. When the RE bit is set to 1, the RxD pins function as input pins and their states are read in preference to the SCPCR setting.

Table 27.8 Read/Write Operation of the SC Port Data Register (SCPDR)

SCPnMD1	SCPnMD0	Pin State	Read	Write
0	0	Other function	SCPDR value	Value is written to SCPDR, but does not affect pin state.
	1	Output	SCPDR value	Write value is output from pin.
1	0	Input (Pullup MOS: on)	Pin state	Value is written to SCPDR, but does not affect pin state.
	1	Input (Pullup MOS: off)	Pin state	Value is written to SCPDR, but does not affect pin state.

(n = 0 to 6)

SCPnMD1	SCPnMD0	Pin State	Read	Write
0	0	Other function	H'00	Ignored (no affect on pin state)
	1	Reserved*	—	—
1	0	Input (Pullup MOS: on)	Pin state	Ignored (no affect on pin state)
	1	Input (Pullup MOS: off)	Pin state	Ignored (no affect on pin state)

Note: * Operation cannot be guaranteed when this bit it set to “reserved.” (n = 7)

Section 28 A/D Converter

28.1 Overview

This LSI includes a 10-bit successive-approximation A/D converter with a selection of up to six analog input channels.

28.1.1 Features

A/D converter features are listed below.

- 10-bit resolution
- Six input channels
- High-speed conversion
  - Conversion time: maximum 15 μs per channel (with 33-MHz peripheral clock)
- Three conversion modes
  - Single mode: A/D conversion of one channel
  - Multi mode: A/D conversion on one to four channels
  - Scan mode: Continuous A/D conversion on one to four channels
- Four 16-bit data registers
  - A/D conversion results are transferred for storage into data registers corresponding to the channels.
- Sample-and-hold function
- A/D conversion can be externally triggered
- A/D interrupt requested at the end of conversion
  - At the end of A/D conversion, an A/D end interrupt (ADI) can be requested.

28.1.2 Block Diagram

Figure 28.1 shows a block diagram of the A/D converter.

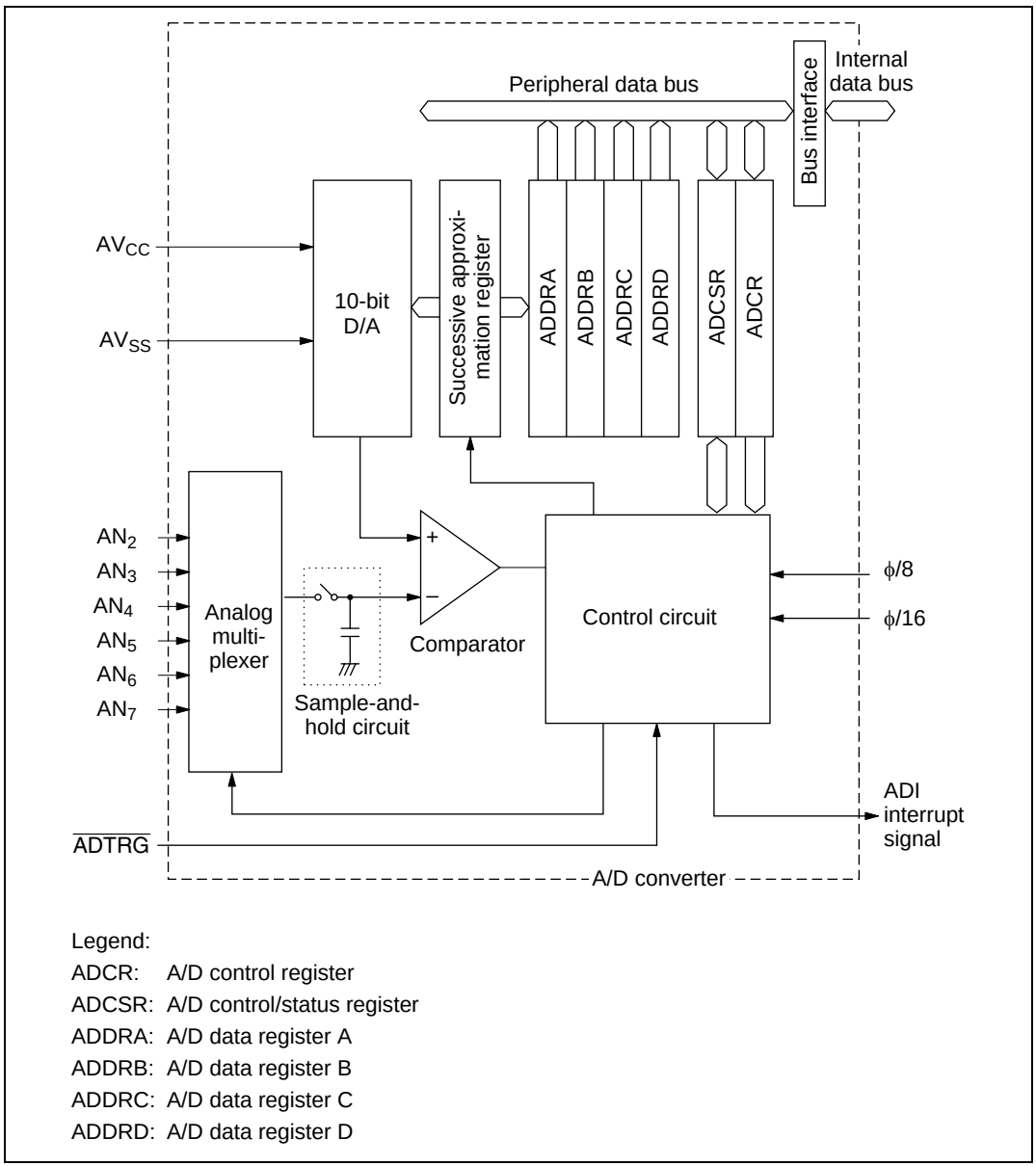


Figure 28.1 A/D Converter Block Diagram

28.1.3 Input Pins

Table 28.1 summarizes the A/D converter's input pins. The six analog input pins are divided into two groups: group 0 (AN2 , AN3), and group 1 (AN4 to AN7). AV_{CC} and AV_{SS} are the power supply for the analog circuits in the A/D converter. AVcc also functions as the A/D converter reference voltage.

Table 28.1 A/D Converter Pins

Pin Name	Abbreviation	I/O	Function
Analog power-supply pin	AVcc	Input	Analog power supply and A/D converter standard voltage
Analog ground pin	AVss	Input	Analog ground and reference voltage
Analog input pin 2	AN2	Input	Group 0 analog inputs
Analog input pin 3	AN3	Input	
Analog input pin 4	AN4	Input	Group 1 analog inputs
Analog input pin 5	AN5	Input	
Analog input pin 6	AN6	Input	
Analog input pin 7	AN7	Input	
A/D external trigger input pin	$\overline{\text{ADTRG}}$	Input	External trigger input for starting A/D conversion

28.1.4 Register Configuration

Table 28.2 summarizes the A/D converter's registers.

Table 28.2 A/D Converter Registers

Name	Abbreviation	R/W	Initial Value	Address	Access size
A/D data register A (high)	ADDRAH	R	H'00	H'04000080 (H'A4000080)*2	16, 8
A/D data register A (low)	ADDRAL	R	H'00	H'04000082 (H'A4000082)*2	8
A/D data register B (high)	ADDRBH	R	H'00	H'04000084 (H'A4000084)*2	16, 8
A/D data register B (low)	ADDRBL	R	H'00	H'04000086 (H'A4000086)*2	8
A/D data register C (high)	ADDRCH	R	H'00	H'04000088 (H'A4000088)*2	16, 8
A/D data register C (low)	ADDRCL	R	H'00	H'0400008A (H'A400008A)*2	8
A/D data register D (high)	ADDRDH	R	H'00	H'0400008C (H'A400008C)*2	16, 8
A/D data register D (low)	ADDRDL	R	H'00	H'0400008E (H'A400008E)*2	8
A/D control/status register	ADCSR	R/(W)*1	H'00	H'04000090 (H'A4000090)*2	8
A/D control register	ADCR	R/W	H'07	H'04000092 (H'A4000092)*2	8

Notes: These registers are located in area 1 of physical space. Therefore, when the cache is on, either access these registers from the P2 area of logical space or else make an appropriate setting using the MMU so that these registers are not cached.

1. Only 0 can be written to bit 7, to clear the flag.
2. When address translation by the MMU does not apply, the address in parentheses should be used.

28.2 Register Descriptions

28.2.1 A/D Data Registers A to D (ADDRA to ADDRD)

Upper register: H

Bit:	15	14	13	12	11	10	9	8
	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Lower register: L

Bit:	7	6	5	4	3	2	1	0
	AD1	AD0	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

The four A/D data registers (ADDRA to ADDRD) are 16-bit read-only registers that store the results of A/D conversion.

An A/D conversion produces 10-bit data, which is transferred for storage into the A/D data register corresponding to the selected channel. The upper 8 bits of the converted data are transferred to upper register H (bits 7 to 0) of the A/D data register, and the lower 2 bits are transferred to lower register L (bits 7 and 6), for storage. Lower register L (bits 5 to 0) is always read as 0. Table 28.3 indicates the pairings of analog input channels and A/D data registers.

The A/D data registers are initialized to H'0000 by a reset and in standby mode.

Table 28.3 Analog Input Channels and A/D Data Registers

Analog Input Channel		
Group 0	**Group 1**	**A/D Data Register**
reserved	AN4	ADDRA
reserved	AN5	ADDRB
AN2	AN6	ADDRC
AN3	AN7	ADDRD

28.2.2 A/D Control/Status Register (ADCSR)

Bit:	7	6	5	4	3	2	1	0
	ADF	ADIE	ADST	MULTI	CKS	CH2	CH1	CH0
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

Note: * Write 0 to clear the flag.

ADCSR is an 8-bit read/write register that controls the A/D converter and indicates the status. ADCSR is initialized to H'00 by a reset and in standby mode.

Bit 7—A/D End Flag (ADF): Indicates the end of A/D conversion.

Bit 7: ADF	Description	
0	[Clear condition]	(Initial value)
	(1) Cleared by reading ADF while ADF = 1, then writing 0 in ADF	
	(2) Cleared when DMAC is activated by ADI interrupt and ADDR is read	
1	[Set conditions]	
	Single mode: A/D conversion ends	
	Multi mode and scan mode: A/D conversion ends in all selected channels	

Bit 6—A/D Interrupt Enable (ADIE): Enables or disables the interrupt (ADI) requested at the end of A/D conversion.

Bit 6: ADIE	Description	
0	A/D end interrupt request (ADI) is disabled	(Initial value)
1	A/D end interrupt request (ADI) is enabled	

Bit 5—A/D Start (ADST): Starts or stops A/D conversion. The ADST bit remains set to 1 during A/D conversion. It can also be set to 1 by external trigger input at the $\overline{\text{ADTRG}}$ pin.

Bit 5: ADST	Description	
0	A/D conversion is stopped	(Initial value)
1	Single mode: A/D conversion starts; ADST is automatically cleared to 0 when conversion ends.	
	Multi mode: A/D conversion starts, cycling among the selected channels. After the cycling has been completed, ADST is cleared to 0.	
	Scan mode: A/D conversion starts and continues, until ADST is cleared to 0 by software, by a reset, or by a transition to standby mode.	

Bit 4—Multi Mode (MULTI): Selects single mode, multi mode or scan mode. For further information on operation in these modes, see section 28.4, Operation.

Bit 4: MULTI	ADCR: Bit 5: SCN	Description	
0	0	Single mode	(Initial value)
	1		
1	0	Multi mode	
	1	scan mode	

Bit 3—Clock Select (CKS): Selects the A/D conversion time. Clear the ADST bit to 0 before switching the conversion time.

Bit 3:CKS	Description	
0	Conversion time = 536 states (maximum)	(Initial value)
1	Conversion time = 266 states (maximum)*	

Note: * The CKS value should be set so that the A/D conversion time is 16 µs (minimum).

Bits 2 to 0—Channel Select 2 to 0 (CH2 to CH0): These bits and the MULTI bit select the analog input channels. Clear the ADST bit to 0 before changing the channel selection.

Channel Selection			Description	
CH2	**CH1**	**CH0**	**Single Mode (MULTI = 0)**	**Multi Mode (MULTI = 1)**
0	0	0	reserved	reserved
		1	reserved	reserved
	1	0	AN2	AN2
		1	AN3	AN2, AN3
1	0	0	AN4	AN4
		1	AN5	AN4, AN5
	1	0	AN6	AN4 to AN6
		1	AN7	AN4 to AN7

28.2.3 A/D Control Register (ADCR)

Bit:	7	6	5	4	3	2	1	0
	TRGE1	TRGE0	SCN	RESVD1	RESVD2	—	—	—
Initial value:	0	0	0	0	0	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R	R	R

ADCR is an 8-bit read/write register that enables or disables external triggering of A/D conversion. ADCR is initialized to H'07 by a reset and in standby mode.

Bits 7 and 6—Trigger Enable (TRGE1, TRGE0): Enables or disables external triggering of A/D conversion.

Bit 7: TRGE1	Bit 6: TRGE0	Description
0	0	When an external trigger is input, the A/D conversion does not start (Initial value)
0	1	
1	0	
1	1	The A/D conversion starts at the falling edge of an input signal from the external trigger pin ($\overline{\text{ADTRG}}$).

Bit 5—Scan Mode (SCN): Selects multi mode or scan mode when the MULTI bit is set to 1. See the description of bit 4 in section 28.2.2, A/D Control/Status Register (ADCSR).

Bits 4 and 3—Reserved (RESVD1, RESVD2): These bits always read 0. The write value should always be 0.

Bits 2 to 0—Resrved: These bits are always read as 0. The write value should always be 0.

28.3 Bus Master Interface

ADDRA to ADDRD are 16-bit registers, but they are connected to the bus master by the upper 8 bits of the 16-bit peripheral data bus. Therefore, although the upper byte can be accessed directly by the bus master, the lower byte is read through an 8-bit temporary register (TEMP).

An A/D data register is read as follows. When the upper byte is read, the upper-byte value is transferred directly to the bus master and the lower-byte value is transferred into TEMP. Next, when the lower byte is read, the TEMP contents are transferred to the bus master.

When reading an A/D data register, always read the upper byte before the lower byte. It is possible to read only the upper byte, but if only the lower byte is read, the read value is not guaranteed.

Figure 28.2 shows the data flow for access to an A/D data register.

See section 28.7.3, Access Size and Read Data.

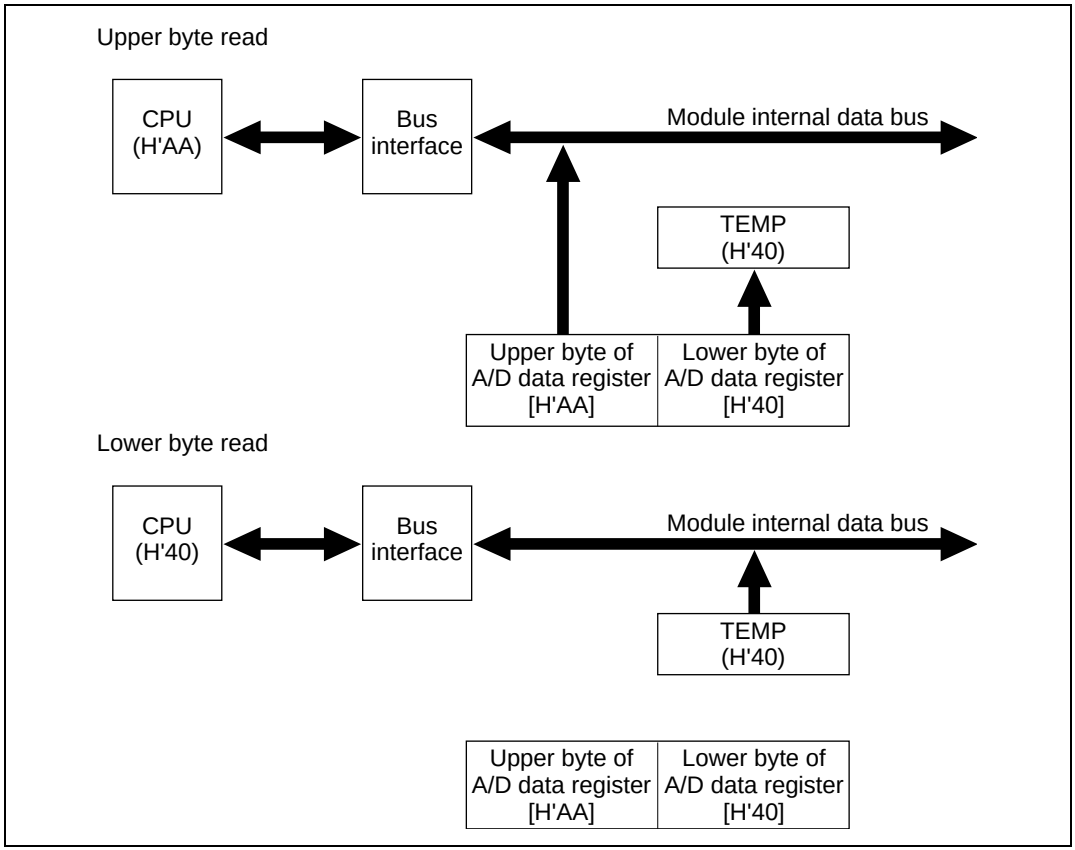


Figure 28.2 A/D Data Register Access Operation (Reading H'AA40)

28.4 Operation

The A/D converter operates by successive approximations with 10-bit resolution. It has three operating modes: single mode, multi mode, and scan mode.

28.4.1 Single Mode (MULTI = 0)

Single mode should be selected when only one A/D conversion on one channel is required. A/D conversion starts when the ADST bit is set to 1 by software, or by external trigger input. The ADST bit remains set to 1 during A/D conversion and is automatically cleared to 0 when conversion ends.

When conversion ends the ADF bit is set to 1. If the ADIE bit is also set to 1, an ADI interrupt is requested at this time. To clear the ADF flag to 0, first read ADF, then write 0 in ADF.

When the mode or analog input channel must be switched during A/D conversion, to prevent incorrect operation, first clear the ADST bit to 0 in ADCSR to halt A/D conversion. After making the necessary changes, set the ADST bit to 1 to start A/D conversion again. The ADST bit can be set at the same time as the mode or channel is changed.

Typical operations when channel 1 (AN2) is selected in single mode are described next.

Figure 28.3 shows a timing diagram for this example.

1. Single mode is selected (MULTI = 0), input channel AN2 is selected (CH2 = CH0 = 0, CH1 = 1), the A/D interrupt is enabled (ADIE = 1), and A/D conversion is started (ADST = 1).
2. When A/D conversion is completed, the result is transferred into ADDRC. At the same time the ADF flag is set to 1, the ADST bit is cleared to 0, and the A/D converter becomes idle.
3. Since ADF = 1 and ADIE = 1, an ADI interrupt is requested.
4. The A/D interrupt processing routine starts.
5. The routine reads ADF, then writes 0 in the ADF flag.
6. The routine reads and processes the conversion result (ADDRC = 0).
7. Execution of the A/D interrupt processing routine ends. Then, when the ADST bit is set to 1, A/D conversion starts to execute 2 to 7 above.

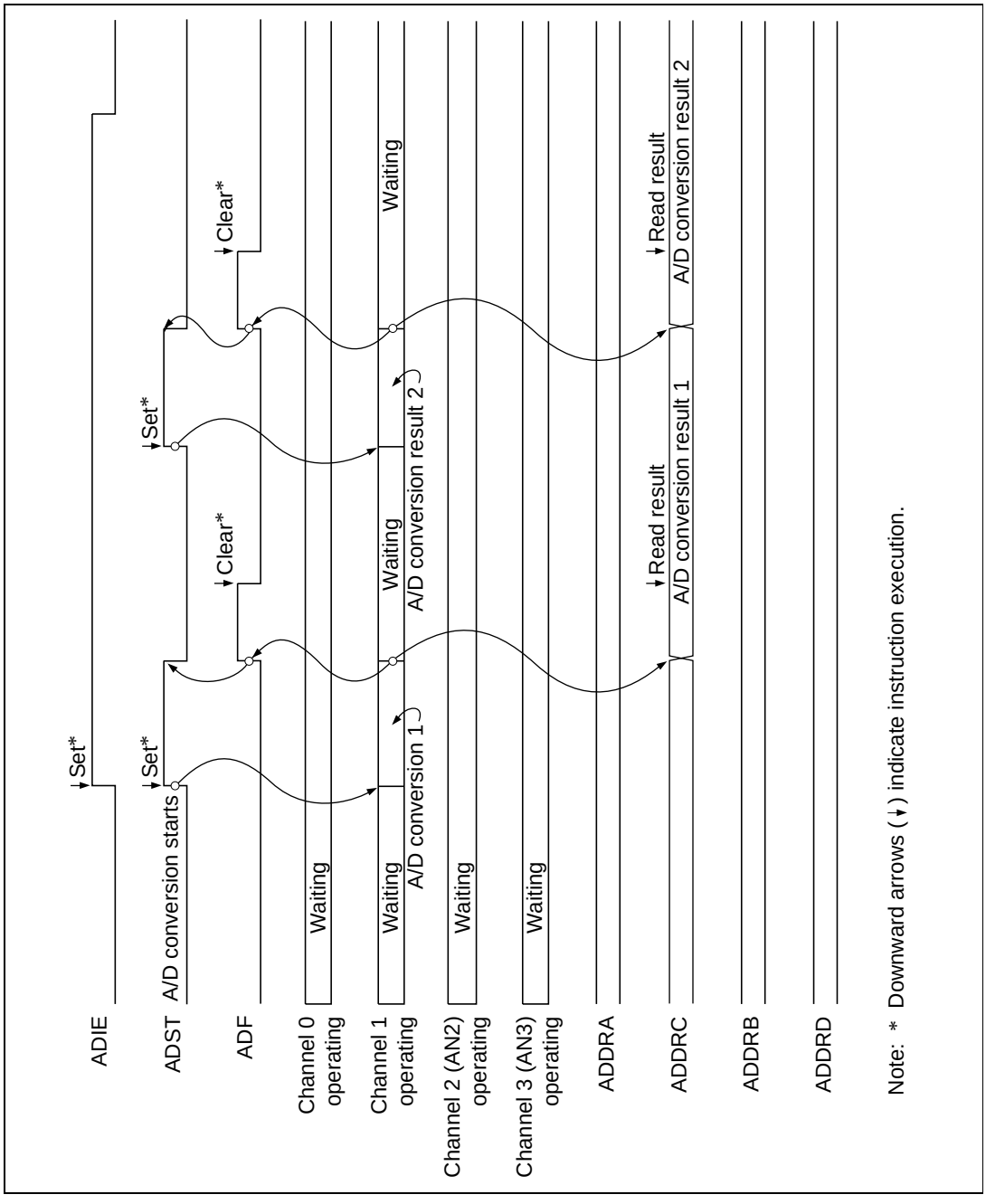


Figure 28.3 Example of A/D Converter Operation (Single Mode, Channel 2 Selected)

28.4.2 Multi Mode (MULTI = 1, SCN = 0)

Multi mode should be selected when performing multi channel A/D conversions on one or more channels including channel 1. When the ADST bit is set to 1 by software or external trigger input, A/D conversion starts on the first channel in the group (AN2 when CH2 = 0, and AN4 when CH2 = 1). When two or more channels are selected, after conversion of the first channel (AN2 or AN4) ends, conversion of the second channel (AN3 or AN5) starts immediately. Finally, all of the specified channels are converted in a loop. The conversion results are transferred for storage into the A/D data registers corresponding to the channels.

When the mode or analog input channel selection must be changed during A/D conversion, to prevent incorrect operation, first clear the ADST bit to 0 in ADCSR to halt A/D conversion. After making the necessary changes, set the ADST bit to 1. A/D conversion will start again from the first channel in the group. The ADST bit can be set at the same time as the mode or channel selection is changed.

Typical operations when three channels in group 1 (AN4 to AN6) are selected in multi mode are described next. Figure 28.4 shows a timing diagram for this example.

1. Multi mode is selected (MULTI = 1), channel group 1 is selected (CH2 = 1), analog input channels AN4 to AN6 are selected (CH1 = 1, CH0 = 0), and A/D conversion is started (ADST = 1).
2. When A/D conversion of the first channel (AN4) is completed, the result is transferred into ADDRA. Next, conversion of the second channel (AN5) starts automatically.
3. Conversion proceeds in the same way through the third channel (AN6).
4. When conversion of all selected channels (AN4 to AN6) is completed, the ADF flag is set to 1 and ADST bit is cleared to 0. If the ADIE bit is set to 1, an ADI interrupt is requested after A/D conversion.

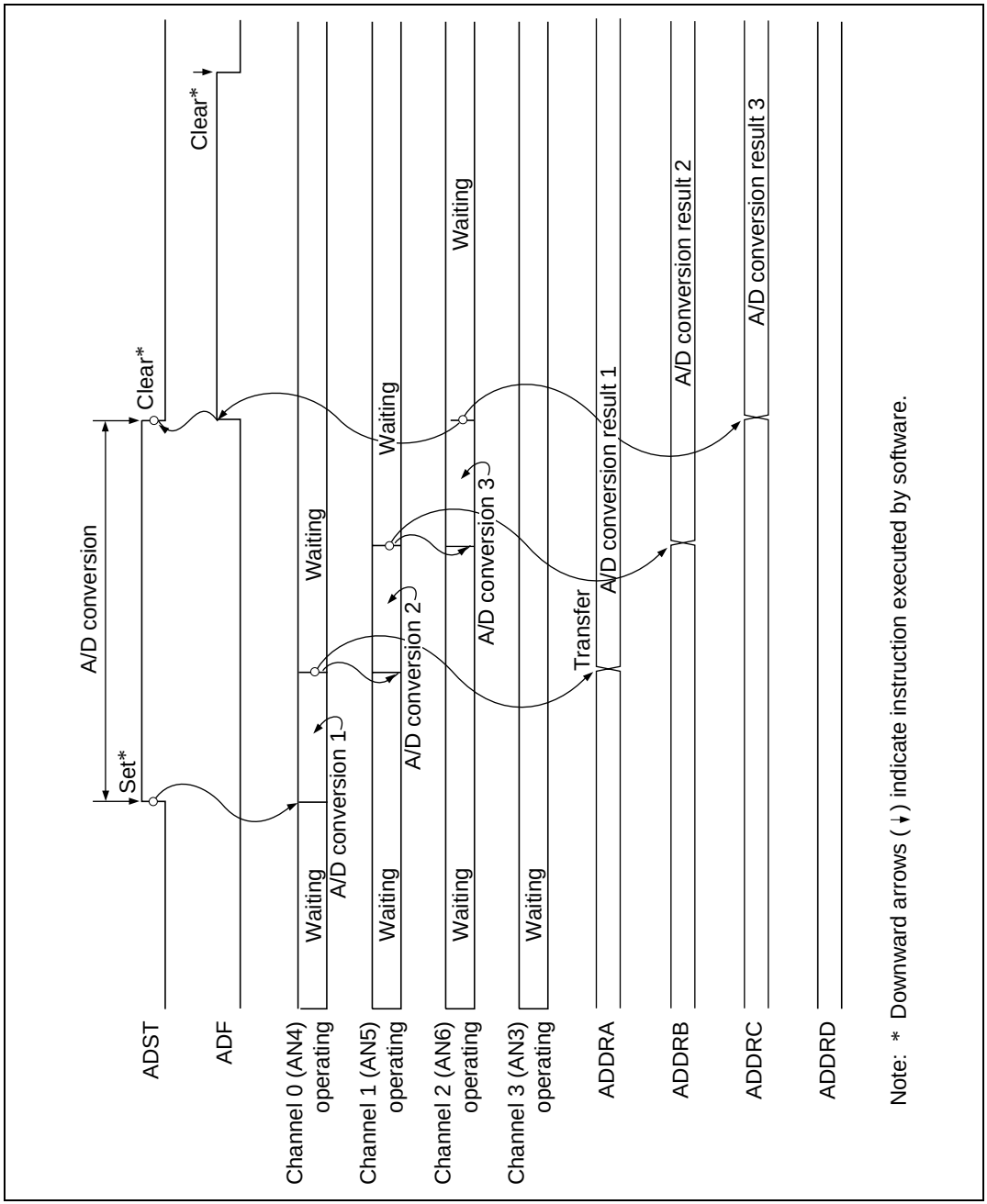


Figure 28.4 Example of A/D Converter Operation (Multi Mode, Channels AN4 to AN6 Selected)

28.4.3 Scan Mode (MULTI = 1, SCN = 1)

Scan mode is useful for monitoring analog inputs in a group of one or more channels including channel 1. When the ADST bit in the A/D control/status register (ADCSR) is set to 1 by software or external trigger input, A/D conversion starts on the first channel in the group (AN2 when CH2 = 0, AN4 when CH2 = 1). When two or more channels are selected, after conversion of the first channel ends, conversion of the second channel (AN3 or AN5) starts immediately. A/D conversion is repeated continuously on the selected channels until the ADST bit is cleared to 0.

The conversion results are transferred for storage into the A/D data registers corresponding to the channels.

When the mode or analog input channel must be changed during analog conversion, to prevent incorrect operation, first clear the ADST bit to 0 to halt A/D conversion. After making the necessary changes, set the ADST bit to 1. A/D conversion will start again from the first channel in the group. The ADST bit can be set at the same time as the mode or channel selection is changed.

Typical operations when three channels (AN4 to AN6) are selected in scan mode are described next. Figure 28.5 shows a timing diagram for this example.

1. Scan mode is selected (MULTI = 1, SCN = 1), channel group 1 is selected (CH2 = 1), analog input channels AN4 to AN6 are selected (CH1 = 1, CH0 = 0), and A/D conversion is started (ADST = 1).
2. When A/D conversion of the first channel (AN4) is completed, the result is transferred into ADDRA. Next, conversion of the second channel (AN5) starts automatically.
3. Conversion proceeds in the same way through the third channel (AN6).
4. When conversion of all the selected channels (AN4 to AN6) is completed, the ADF flag is set to 1 and conversion of the first channel (AN4) starts again. If the ADIE bit is set to 1, an ADI interrupt is requested at this time.
5. Steps 2 to 4 are repeated as long as the ADST bit remains set to 1. When the ADST bit is cleared to 0, A/D conversion stops. After that, if the ADST bit is set to 1, A/D conversion starts again from the first channel (AN4).

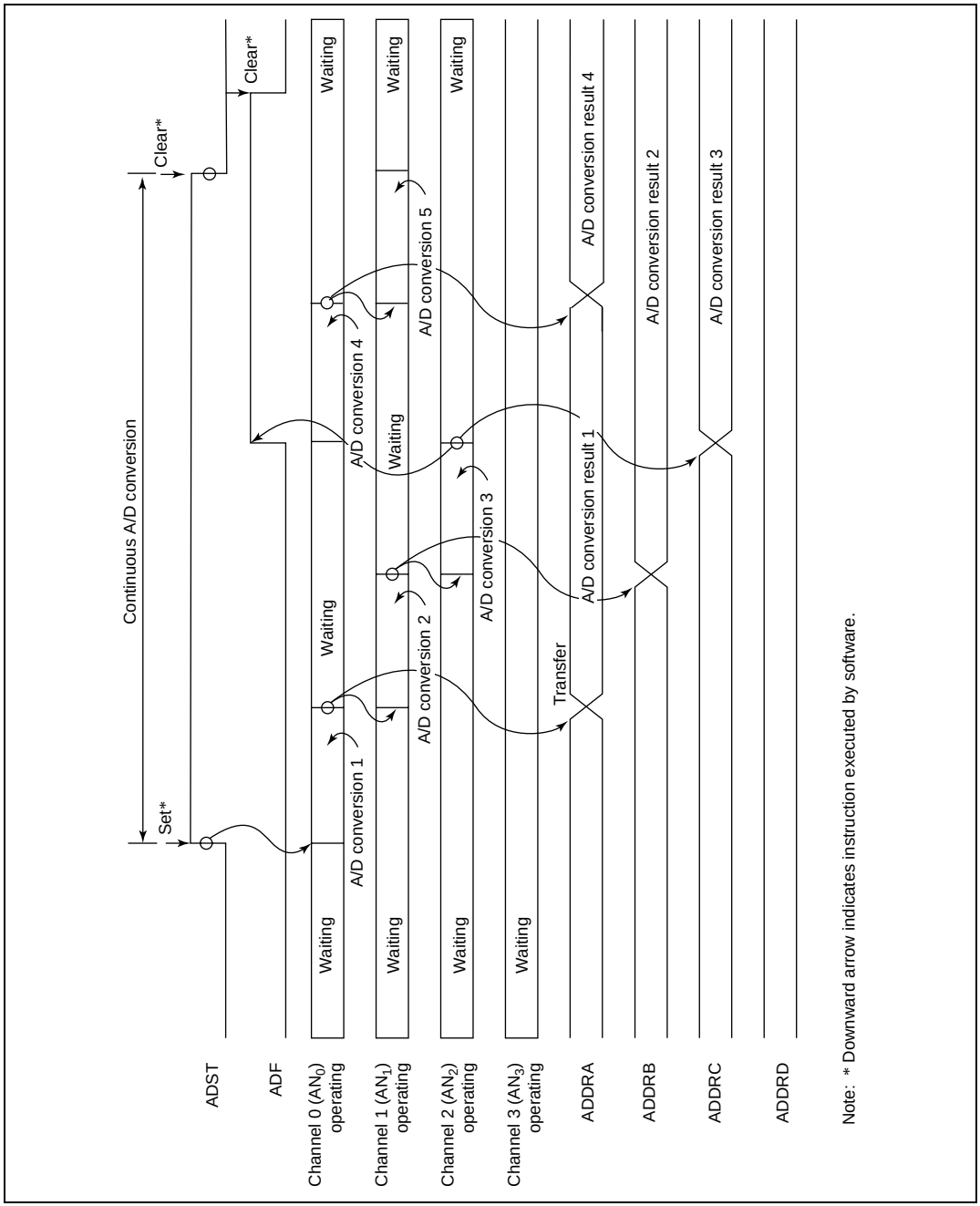


Figure 28.5 Example of A/D Converter Operation (Scan Mode, Channels AN4 to AN6 Selected)

28.4.4 Input Sampling and A/D Conversion Time

The A/D converter has a built-in sample-and-hold circuit. The A/D converter samples the analog input at a time t_D after the ADST bit of ADCSR is set to 1, then starts conversion. Figure 28.6 shows the A/D conversion timing. Table 28.4 indicates the A/D conversion time.

As indicated in figure 28.6, the A/D conversion time includes t_D and the input sampling time. The length of t_D varies depending on the timing of the write access to ADCSR. The total conversion time therefore varies within the ranges indicated in table 28.4.

In multi mode and scan mode, the values given in table 28.4 apply to the first conversion. In the second and subsequent conversions the conversion time is fixed at 536 states when CKS = 0 or 266 states when CKS = 1.

In all cases, the CKS bit in ADCSR should be set according to the frequency of Pϕ so that the conversion time is within the range shown in table 32.16 in section 32, Electrical Characteristics.

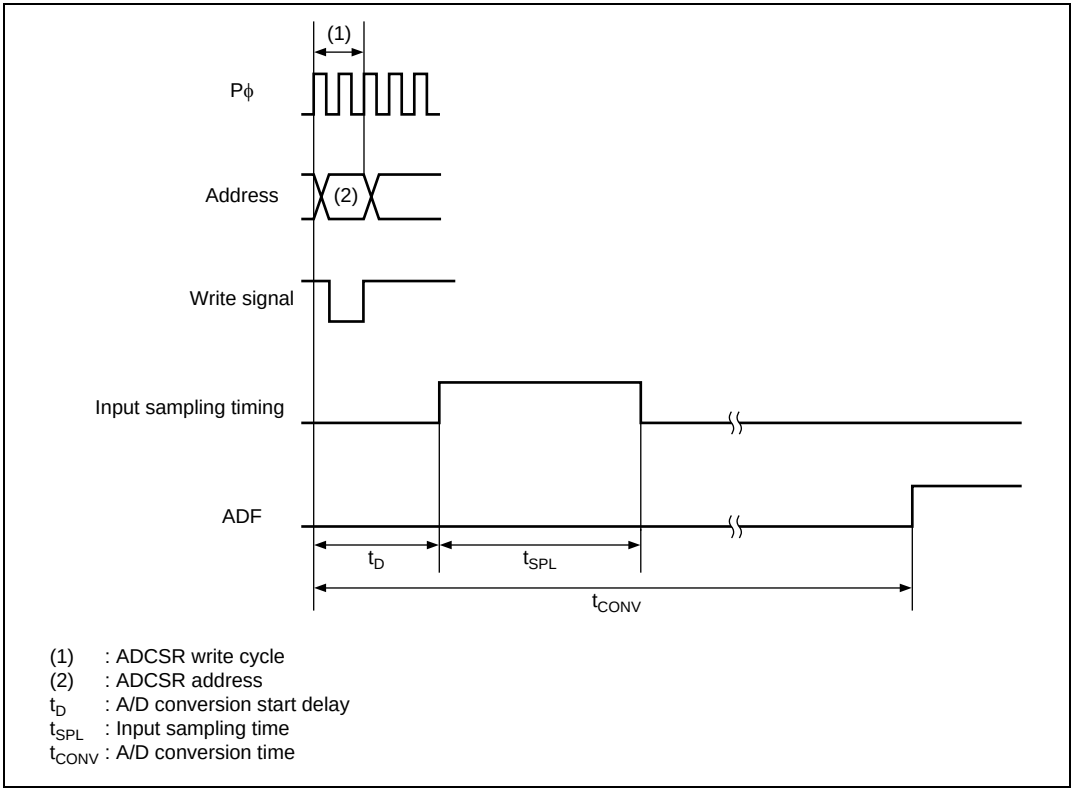


Figure 28.6 A/D Conversion Timing

Table 28.4 A/D Conversion Time (Single Mode)

		CKS = 0			CKS = 1		
	Symbol	**Min**	**Typ**	**Max**	**Min**	**Typ**	**Max**
A/D conversion start delay	t_D	17	—	28	10	—	17
Input sampling time	t_{SPL}	—	129	—	—	65	—
A/D conversion time	t_{CONV}	514	—	525	259	—	266

Note: Values in the table are numbers of states (t_{cyc}).

28.4.5 External Trigger Input Timing

A/D conversion can be externally triggered. When the TRGE1, TRGE0 bits are set to 1 in ADCR, external trigger input is enabled at the $\overline{\text{ADTRG}}$ pin. A high-to-low transition at the $\overline{\text{ADTRG}}$ pin sets the ADST bit to 1 in ADCSR, starting A/D conversion. Other operations, regardless of the conversion mode, are the same as if the ADST bit had been set to 1 by software. Figure 28.7 shows the timing.

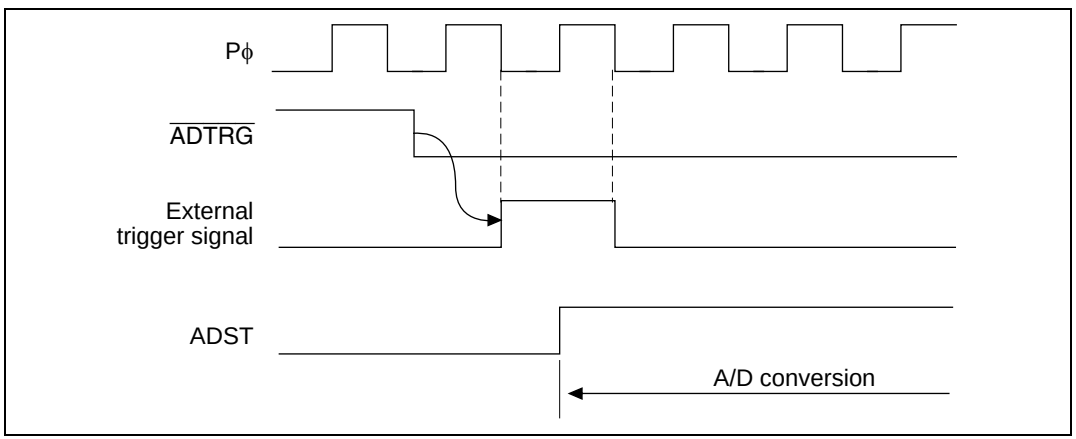


Figure 28.7 External Trigger Input Timing

28.5 Interrupts

The A/D converter generates an interrupt (ADI) at the end of A/D conversion. The ADI interrupt request can be enabled or disabled by the ADIE bit in ADCSR.

28.6 Definitions of A/D Conversion Accuracy

The A/D converter compares an analog value input from an analog input channel to its analog reference value and converts it into 10-bit digital data. The absolute accuracy of this A/D conversion is the deviation between the input analog value and the output digital value. It includes the following errors:

- Offset error
- Full-scale error
- Quantization error
- Nonlinearity error

These four error quantities are explained below using figure 28.8. In the figure, the 10 bits of the A/D converter have been simplified to 3 bits.

Offset error is the deviation between actual and ideal A/D conversion characteristics when the digital output value changes from the minimum (zero voltage) 0000000000 (000 in the figure) to 000000001 (001 in the figure)(figure 28.8, item (1)). Full-scale error is the deviation between actual and ideal A/D conversion characteristics when the digital output value changes from the 1111111110 (110 in the figure) to the maximum 1111111111 (111 in the figure)(figure 28.8, item (2)). Quantization error is the intrinsic error of the A/D converter and is expressed as 1/2 LSB (figure 28.8, item (3)). Nonlinearity error is the deviation between actual and ideal A/D conversion characteristics between zero voltage and full-scale voltage (figure 28.8, item (4)). Note that it does not include offset, full-scale or quantization error.

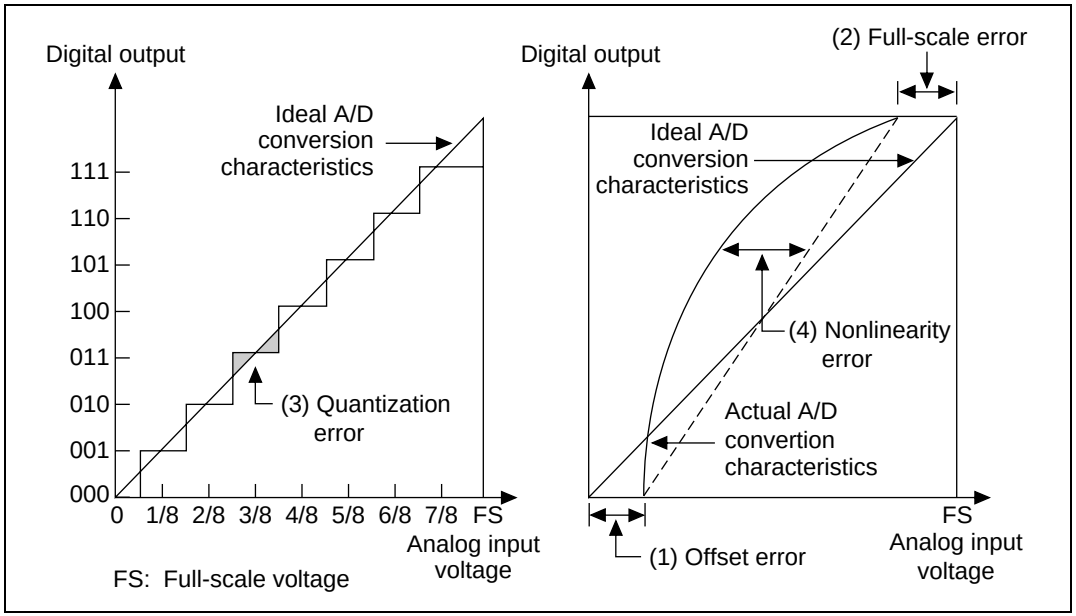


Figure 28.8 Definitions of A/D Conversion Accuracy

28.7 Usage Notes

When using the A/D converter, note the points listed in section 28.7.1 below.

28.7.1 Setting Analog Input Voltage

- Analog Input Voltage Range: During A/D conversion, the voltages input to the analog input pins ANn should be in the range $AV_{SS} \leq ANn \leq AV_{CC}$ (n = 2 to 7).
- Set the AV_{CC} and AV_{SS} input voltages as defined in section 32, Electrical Characteristics.

28.7.2 Processing of Analog Input Pins

To prevent damage from voltage surges at the analog input pins (AN2 to AN7), connect an input protection circuit like the one shown in figure 28.9. The circuit shown also includes an RC filter to suppress noise. This circuit is shown as an example; The circuit constants should be selected according to actual application conditions. Table 28.5 lists the analog input pin specifications and figure 28.10 shows an equivalent circuit diagram of the analog input ports.

28.7.3 Access Size and Read Data

Table 28.6 shows the relationship between access size and read data. Note the read data obtained with different access sizes, bus widths, and endian modes.

The case is shown here in which H'3FF is obtained when AV_{CC} is input as an analog input. FF is the data containing the upper 8 bits of the conversion result, and C0 is the data containing the lower 2 bits.

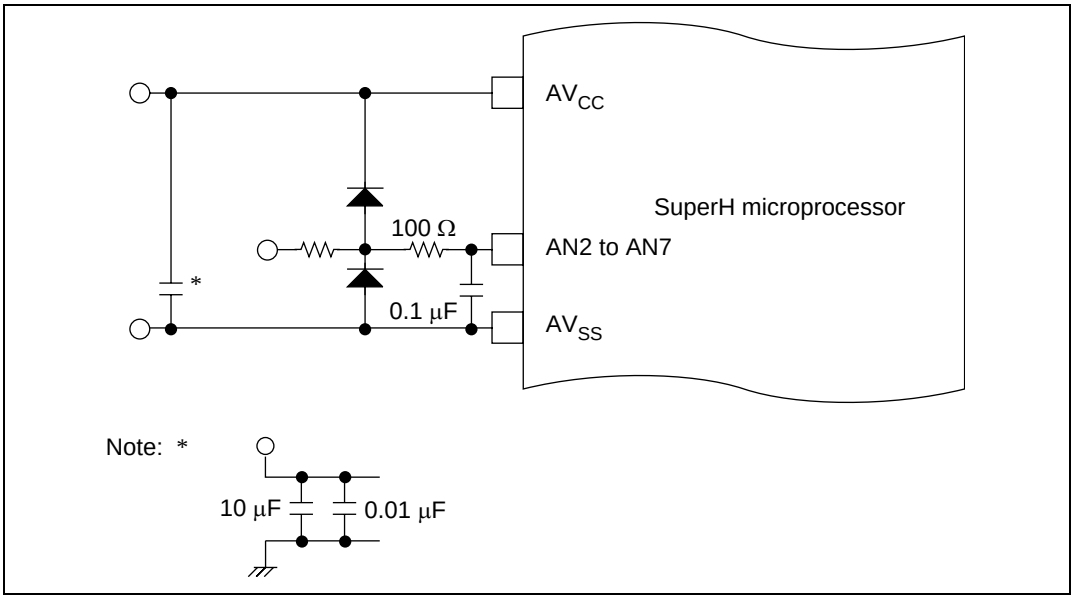


Figure 28.9 Example of Analog Input Protection Circuit

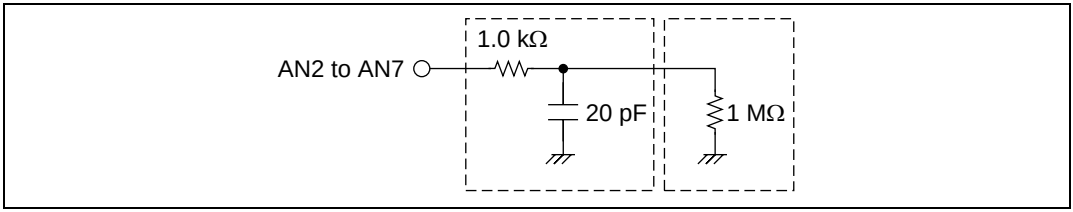


Figure 28.10 Analog Input Pin Equivalent Circuit

Table 28.5 Analog Input Pin Ratings

Item	Min	Max	Unit
Analog input capacitance	—	20	pF
Allowable signal-source impedance	—	5	kΩ

Table 28.6 Relationship between Access Size and Read Data

Access Size	Command	Bus Width / Endian	32 Bits (D31—D0) Big	32 Bits (D31—D0) Little	16 Bits (D15—D0) Big	16 Bits (D15—D0) Little	8 Bits (D7—D0) Big	8 Bits (D7—D0) Little
Byte access	MOV.L	#ADDRAH,R9						
	MOV.B	@R9,R8	FFFFFFFF	FFFFFFFF	FFFF	FFFF	FF	FF
	MOV.L	#ADDRAL,R9						
	MOV.B	@R9,R8	C0C0C0C0	C0C0C0C0	C0C0	C0C0	C0	C0
Word access	MOV.L	#ADDRAH,R9						
	MOV.W	@R9,R8	FFxxFFxx	FFxxFFxx	FFxx	FFxx	FF	xx
							xx	FF
	MOV.L	#ADDRAL,R9						
	MOV.W	@R9,R8	C0xxC0xx	C0xxC0xx	C0xx	C0xx	C0	xx
							xx	C0
Longword access	MOV.L	#ADDRAH,R9						
	MOV.L	@R9,R8	FFxxC0xx	FFxxC0xx	FFxx	C0xx	FF	xx
					C0xx	FFxx	xx	C0
							C0	xx
							xx	FF

In this table:
```
#ADDRAH    .EQU    H'04000080
#ADDRAL    .EQU    H'04000082
```
Values are shown in hexadecimal for the case where read data is output to an external device via R8.

Section 29 D/A Converter

29.1 Overview

This LSI includes a D/A converter with two channels.

29.1.1 Features

D/A converter features are listed below.

- Eight-bit resolution
- Two output channels
- Conversion time: maximum 10 µs (with 20-pF capacitive load)
- Output voltage: 0 V to AVcc

29.1.2 Block Diagram

Figure 29.1 shows a block diagram of the D/A converter.

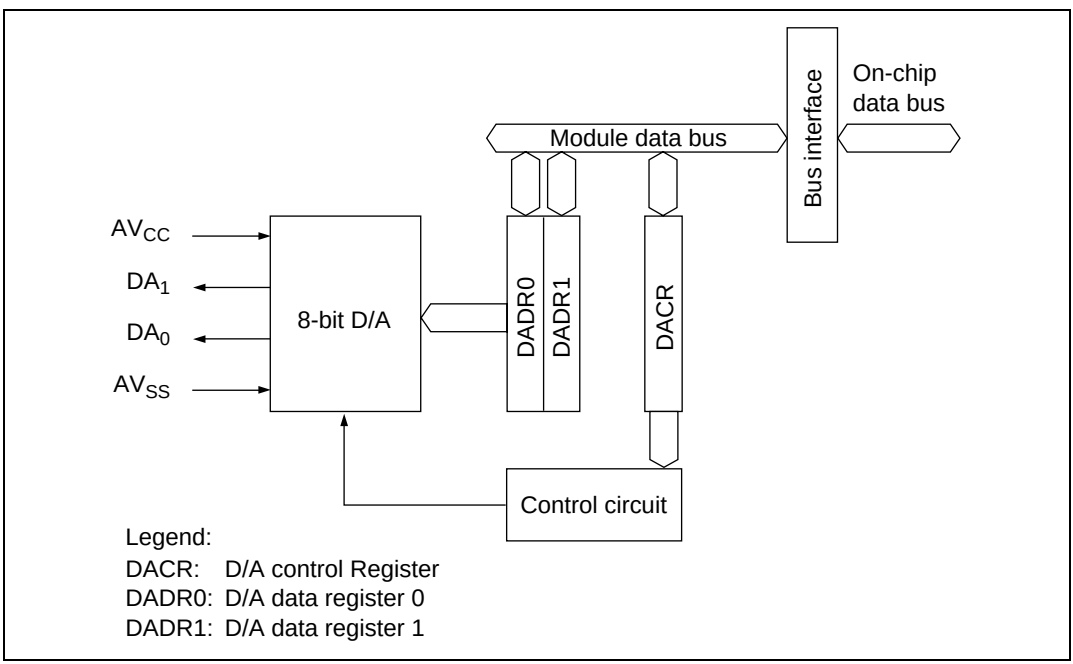


Figure 29.1 D/A Converter Block Diagram

29.1.3 I/O Pins

Table 29.1 summarizes the D/A converter's input and output pins.

Table 29.1 D/A Converter Pins

Pin Name	Abbreviation	I/O	Function
Analog power-supply pin	AVcc	Input	Analog power supply and D/A converter standard voltage
Analog ground pin	AVss	Input	Analog ground
Analog output pin 0	DA0	Output	Analog output, channel 0
Analog output pin 1	DA1	Output	Analog output, channel 1

29.1.4 Register Configuration

Table 29.2 summarizes the D/A converter's registers.

Table 29.2 D/A Converter Registers

Name	Abbreviation	R/W	Initial Value	Address*1	Access size
D/A data register 0	DADR0	R/W	H'00	H'040000A0 (H'A40000A0)*2	8
D/A data register 1	DADR1	R/W	H'00	H'040000A2 (H'A40000A2)*2	8
D/A control register	DACR	R/W	H'1F	H'040000A4 (H'A40000A4)*2	8

Notes: These registers are located in area 1 of physical space. Therefore, when the cache is on, either access these registers from the P2 area of logical space or else make an appropriate setting using the MMU so that these registers are not cached.

1. Lower 16 bits of the address
2. When address translation by the MMU does not apply, the address in parentheses should be used.

29.2 Register Descriptions

29.2.1 D/A Data Registers 0 and 1 (DADR0/1)

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

The D/A data registers (DADR0 and DADR1) are 8-bit read/write registers that store the data to be converted. When analog output is enabled, the D/A data register values are constantly converted and output at the analog output pins.

The D/A data registers are initialized to H'00 by a reset.

29.2.2 D/A Control Register (DACR)

Bit:	7	6	5	4	3	2	1	0
	DAOE1	DAOE0	DAE	—	—	—	—	—
Initial value:	0	0	0	1	1	1	1	1
R/W:	R/W	R/W	R/W	R	R	R	R	R

DACR is an 8-bit read/write register that controls the operation of the D/A converter. DACR is initialized to H'1F by a reset.

Bit 7—D/A Output Enable 1 (DAOE1): Controls D/A conversion and analog output.

Bit 7: DAOE1	Description	
0	DA1 analog output is disabled	(Initial value)
1	Channel-1 D/A conversion and DA1 analog output are enabled	

Bit 6—D/A Output Enable 0 (DAOE0): Controls D/A conversion and analog output.

Bit 6: DAOE0	Description	
0	DA0 analog output is disabled	(Initial value)
1	Channel-0 D/A conversion and DA0 analog output are enabled	

Bit 5—D/A Enable (DAE): Controls D/A conversion, together with bits DAOE0 and DAOE1. When the DAE bit is cleared to 0, D/A conversion is controlled independently in channels 0 and 1. When this LSI enters standby mode while D/A conversion is enabled, the D/A output is held and the analog power-supply current is equivalent to that during D/A conversion. To reduce the analog power-supply current in standby mode, clear the DAOE0 and DAOE1 bits and disable the D/A output.

Bit 7: DAOE1	Bit 6: DAOE0	Bit 5: DAE	Description
0	0	—	D/A conversion is disabled in channels 0 and 1 (Initial value)
0	1	0	D/A conversion is enabled in channel 0 D/A conversion is disabled in channel 1
0	1	1	D/A conversion is enabled in channels 0 and 1
1	0	0	D/A conversion is disabled in channel 0 D/A conversion is enabled in channel 1
1	0	1	D/A conversion is enabled in channels 0 and 1
1	1	—	D/A conversion is enabled in channels 0 and 1

When the DAE bit is set to 1, even if bits DAOE0 and DAOE1 in DACR and the ADST bit in ADCSR are cleared to 0, the same current is drawn from the analog power supply as during A/D and D/A conversion.

Bits 4 to 0—Reserved: Read-only bits, always read as 1.

29.3 Operation

The D/A converter has two built-in D/A conversion circuits that can perform conversion independently.

D/A conversion is performed constantly while enabled in DACR. If the DADR0 or DADR1 value is modified, conversion of the new data begins immediately. The conversion results are output when bits DAOE0 and DAOE1 are set to 1.

An example of D/A conversion on channel 0 is given next. Timing is indicated in figure 29.2.

1. Data to be converted is written in DADR0.
2. Bit DAOE0 is set to 1 in DACR. D/A conversion starts and DA0 becomes an output pin. The converted result is output after the conversion time. The output value is (DADR0 contents/256) × AVcc. Output of this conversion result continues until the value in DADR0 is modified or the DAOE0 bit is cleared to 0.
3. If the DADR0 value is modified, conversion starts immediately, and the result is output after the conversion time.
4. When the DAOE0 bit is cleared to 0, DA0 becomes an input pin.

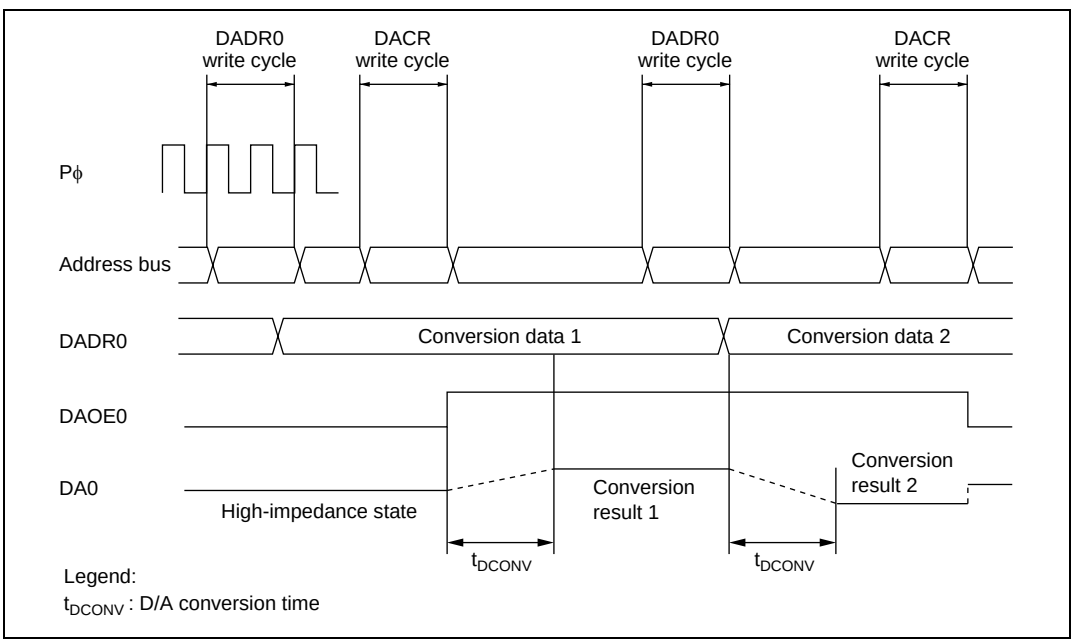


Figure 29.2 Example of D/A Converter Operation

Section 30 PC Card Controller (PCC)

30.1 Overview

The PC card controller (PCC) controls the external buffer, interrupts, and exclusive ports of the PC card interface to be connected to the SH7727. Using the PCC enables two slots of PC cards that conform to the PCMCIA Rev. 2.1/JEIDA Ver. 4.2 standard to be easily connected to the SH7727.

30.1.1 Features

The PCC has the following features:

- As a PC card interface to be connected to physical area 6, an IC memory card interface and an I/O card interface are supported.
- Outputs control signals for the external buffer ($\overline{\text{PCC0DRV}}$).
- Supports a preemptive operating system by switching attribute memory, common memory, and I/O space by using addresses.
- Provides a segment bit (an address bit for the PC card) for common memory, enabling access to a 64-MB space fully conforming to PCMCIA specifications.
- Disables the PCC operation and supports only a bus interface of a PC card interface which is same as the SH7709/SH7729 Series (by using the P0USE bit of PCC0GCR).

30.1.2 Block Diagram

Figure 30.1 shows a block diagram of the PC card controller.

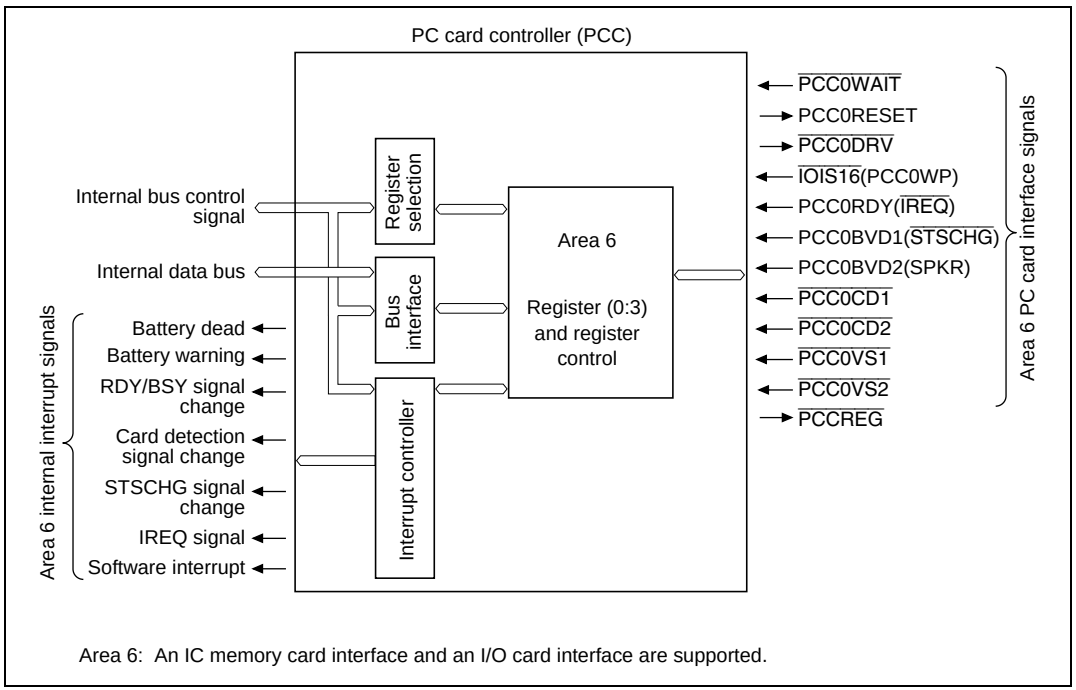


Figure 30.1 PC Card Controller Block Diagram

30.1.3 Register Configuration

Table 30.1 lists the PC card controller registers.

Table 30.1 PC Card Controller Registers

Physical Area	Register Name	Symbol	Read/ Write	Initial Value	Address	Access Size
Physical area 6 (PCC0)	Area 6 interface status register	PCC0ISR	R	*1	H'04000160 (H'A4000160)*2	8 bits
	Area 6 general control register	PCC0GCR	R/W	H'00	H'04000162 (H'A4000162)*2	8 bits
	Area 6 card status change register	PCC0CSCR	R/W	H'00	H'04000164 (H'A4000164)*2	8 bits
	Area 6 card status change interrupt enable register	PCC0CSCIER	R/W	H'00	H'04000166 (H'A4000166)*2	8 bits

Notes: 1. Depends on the PC card status.
2. When address translation by the MMU does not apply, the address in parentheses should be used.

30.1.4 PCMCIA Support

The SH7727 supports an interface based on PCMCIA specifications for physical areas 6. Interfaces supported are the IC memory card interface and I/O card interface defined in the PCMCIA Rev. 2.1/JEIDA Ver. 4.2 standard. Both the IC memory card interface and I/O card interface are supported in area 6.

Table 30.2 Features of the PCMCIA Interface

Item	Feature
Access	Random access
Data bus	8/16 bits
Memory type	Mask ROM, OTPROM, EPROM, EEPROM, flash memory, SRAM
Common memory capacity	Maximum 64 Mbytes (Supports full PCMCIA specifications by using a segment bit (an address bit for the PC card))
Attribute memory capacity	Maximum 32 Mbytes
I/O space capacity	Maximum 32 Mbytes
Others	Dynamic bus sizing for I/O bus width* The PCMCIA interface can be accessed from the address-conversion region and non-address-conversion region.

Note: * Dynamic bus sizing for the I/O bus width is supported only in little-endian mode.

The SH7727 can directly access 32- and 64-MB physical areas in a 64-MB memory space and an I/O space of the PC card (continuous 32/16-MB area mode). The SH7727 provides a segment bit (an address bit for the PC card) in the general control register for area 6 to support a common memory space with full PCMCIA specifications (64 MB).

Continuous 32-MB Area Mode: Setting 0 (initial value) in bit 3 (P0MMOD) of the general control register enables the continuous 32-MB area mode. In this mode, the attribute memory space and I/O memory space are 32 MB and the common memory space is 64 MB. In the common memory space, set 1 in bit 2 (P0PA25) of the general control register to access an address of more than 32 MB. By this operation, 1 is output to A25 pin, enabling an address space of more than 32 MB to be accessed. When an address of 32 MB or less is accessed, no setting is required (initial value: 0). This bit does not affect access to attribute memory space or I/O memory space.

Figure 30.2 shows the relationship between the memory space of the SH7727 and the memory and I/O spaces of the PC card in the continuous 32-MB area mode. Although memory space and I/O space are supported in area 6.

In area 6, set 1 in bit 0 (P0REG) of the general control register to access the common memory space of the PC card, and set 0 in bit 0 to access the attribute memory space (initial value: 0). By this operation, the set value is output to $\overline{\text{PCCREG}}$ pin, enabling any space to be accessed. When the I/O space is accessed in area 6, the output of $\overline{\text{PCCREG}}$ pin is always 0 regardless of the value of bit 0 (P0REG).

See the register descriptions in section 30.2, Register Descriptions for details of register settings.

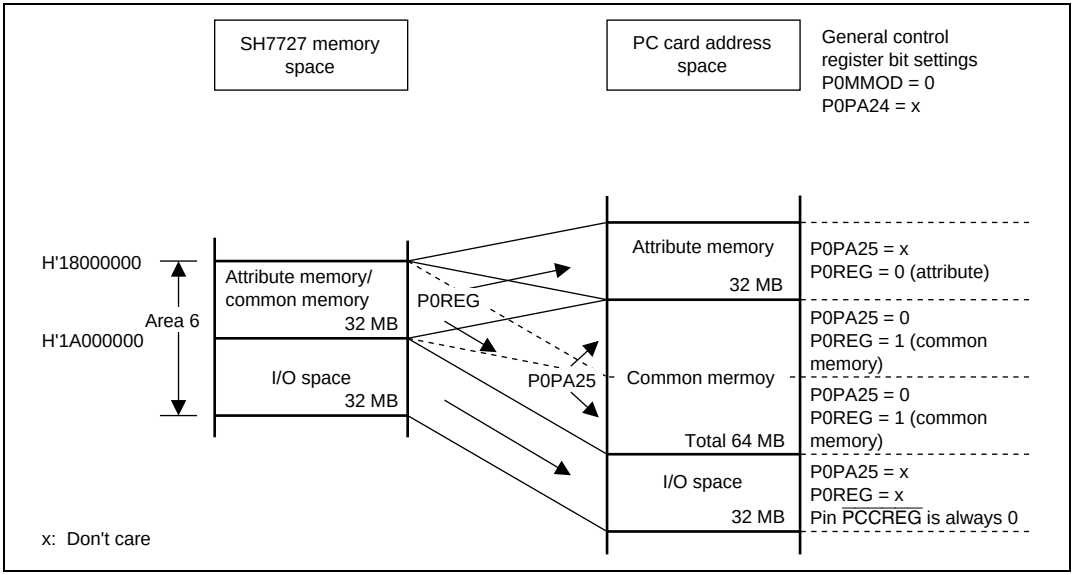


Figure 30.2 Continuous 32-MB Area Mode

Continuous 16-MB Area Mode: Setting 1 in bit 3 (P0MMOD) of the general control register enables the continuous 16-MB area mode. In this mode, the attribute memory space and I/O memory space are 16 MB, and the common memory space is 64 MB. In the common memory space, set the PC card address in bit 2 (P0PA25) and bit 1 (P0PA24) of the general control register to access an address of more than 16 MB. By this operation, values are output to A25 and A24 pins, enabling an address space of more than 16 MB to be accessed (initial value: 0 for P0PA25 and P0PA24). When an address of 16 MB or less is accessed, no settings are required. This bit does not affect access to attribute memory space or I/O memory space.

Figures 30.3 and 30.4 show the relationship between the memory space of the SH7727 and the memory and I/O spaces of the PC card in the continuous 16-MB area mode. Although memory space and I/O space are supported in area 6.

The attribute memory space, common memory space, and I/O space of the PC card are provided as 16-MB physical spaces in this mode. Therefore, the SH7727 automatically controls $\overline{\text{PCCREG}}$ pin (the value of bit 0 (P0REG) in the general control register is ignored). In area 6, the output of $\overline{\text{PCCREG}}$ pin is 0 when the attribute memory space or I/O space is accessed, and 1 when the common memory space is accessed.

See the register descriptions in section 30.2, Register Descriptions for details of register settings.

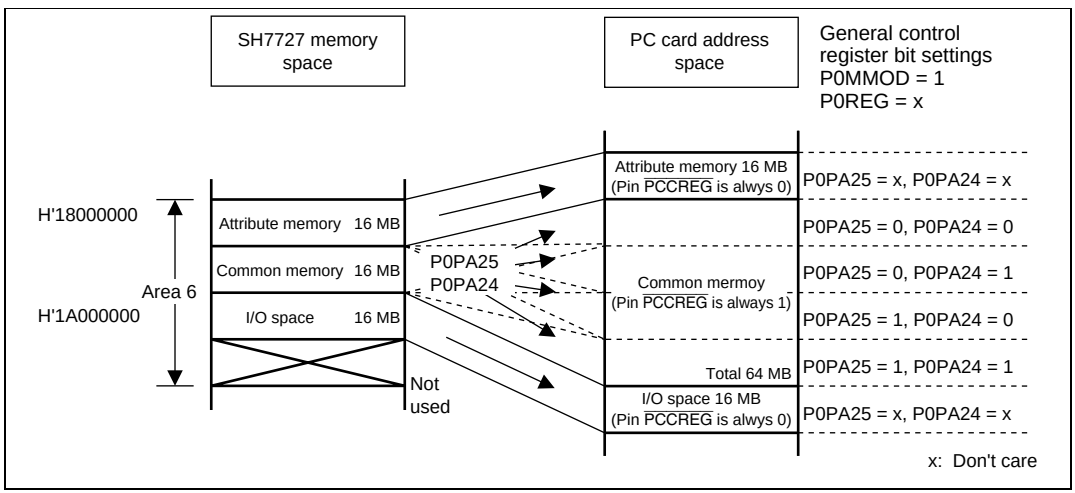


Figure 30.3 Continuous 16-MB Area Mode (Area 6)

30.2 Register Descriptions

30.2.1 Area 6 Interface Status Register (PCC0ISR)

Bit:	7	6	5	4	3	2	1	0
Bit name:	P0RDY/ IREQ	P0MWP	P0VS2	P0VS1	P0CD2	P0CD1	P0BVD2 SPKR	P0BVD1 STSCHG
Initial value:	*	*	*	*	*	*	*	*
R/W:	R	R	R	R	R	R	R	R

Note: * Depends on the PC card status.

The area 6 interface status register (PCC0ISR) is an 8-bit read-only register which is used to read the status of the PC card connected to area 6. The initial value of PCC0ISR depends on the PC card status.

Bit 7—PCC0 Ready (P0RDY/IREQ): The value of RDY/$\overline{\text{BSY}}$ pin of the PC card connected to area 6 is read when the IC memory card interface is connected. The value of $\overline{\text{IREQ}}$ pin of the PC card connected to area 6 is read when the I/O card interface is connected. This bit cannot be written to.

Bit 7: P0RDY/IREQ	Description
0	Indicates that the value of RDY/$\overline{\text{BSY}}$ pin is 0 when the PC card connected to area 6 is the IC memory card interface type. Indicates that the value of $\overline{\text{IREQ}}$ pin is 0 when the PC card connected to area 6 is the I/O card interface type
1	Indicates that the value of RDY/$\overline{\text{BSY}}$ pin is 1 when the PC card connected to area 6 is the IC memory card interface type. Indicates that the value of $\overline{\text{IREQ}}$ pin is 1 when the PC card connected to area 6 is the I/O card interface type

Bit 6—PCC0 Write Protect (P0MWP): The value of WP pin of the PC card connected to area 6 is read when the IC memory card interface is connected. 0 is read when the I/O card interface is connected. This bit cannot be written to.

Bit 6: P0MWP	Description
0	Indicates that the value of WP pin is 0 when the PC card connected to area 6 uses the IC memory card interface. The value of bit 6 is always 0 when the PC card connected to area 6 is the I/O card interface type
1	Indicates that the value of WP pin is 1 when the PC card connected to area 6 is the IC memory card interface type

Bit 5—PCC0 Voltage Sense 2 (P0VS2): The value of $\overline{VS2}$ pin of the PC card connected to area 6 is read. This bit cannot be written to.

Bit 5: P0VS2	Description
0	The value of $\overline{VS2}$ pin of the PC card connected to area 6 is 0
1	The value of $\overline{VS2}$ pin of the PC card connected to area 6 is 1

Bit 4—PCC0 Voltage Sense 1 (P0VS1): The value of $\overline{VS1}$ pin of the PC card connected to area 6 is read. This bit cannot be written to.

Bit 4: P0VS1	Description
0	The value of $\overline{VS1}$ pin of the PC card connected to area 6 is 0
1	The value of $\overline{VS1}$ pin of the PC card connected to area 6 is 1

Bit 3—PCC0 Card Detect 2 (P0CD2): The value of $\overline{CD2}$ pin of the PC card connected to area 6 is read. This bit cannot be written to.

Bit 3: P0CD2	Description
0	The value of $\overline{CD2}$ pin of the PC card connected to area 6 is 0
1	The value of $\overline{CD2}$ pin of the PC card connected to area 6 is 1

Bit 2—PCC0 Card Detect 1 (P0CD1): The value of $\overline{\text{CD1}}$ pin of the PC card connected to area 6 is read. This bit cannot be written to.

Bit 2: P0CD1	Description
0	The value of $\overline{\text{CD1}}$ pin of the PC card connected to area 6 is 0
1	The value of $\overline{\text{CD1}}$ pin of the PC card connected to area 6 is 1

Bits 1 and 0—PCC0 Battery Voltage Detect 2 and 1 (P0BVD2, P0BVD1): The values of BVD2 and BVD1 pins of the PC card connected to area 6 are read when the IC memory card interface is connected. The values of $\overline{\text{SPKR}}$ and $\overline{\text{STSCHG}}$ pins of the PC card connected to area 6 are read when the I/O card interface is connected. These bits cannot be written to.

- IC Memory Interface

Bit 1: P0BVD2	Bit 0: P0BVD1	Description
1	1	The battery voltage of the PC card connected to area 6 is normal (Battery Good)
0	1	The battery must be changed although data is guaranteed for the PC card connected to area 6 (Battery Warning)
1	0	The battery voltage is abnormal and data is not guaranteed for the PC card connected to area 6 (Battery Dead)
0	0	The battery voltage is abnormal and data is not guaranteed for the PC card connected to area 6 (Battery Dead)

- I/O Card Interface

Bit 1: P0SPKR	Description
0	The value of $\overline{\text{SPKR}}$ pin of the PC card connected to area 6 is 0
1	The value of $\overline{\text{SPKR}}$ pin of the PC card connected to area 6 is 1

Bit 0: P0STSCHG	Description
0	The value of $\overline{\text{STSCHG}}$ pin of the PC card connected to area 6 is 0
1	The value of $\overline{\text{STSCHG}}$ pin of the PC card connected to area 6 is 1

30.2.2 Area 6 General Control Register (PCC0GCR)

Bit:	7	6	5	4	3	2	1	0
Bit name:	P0DRVE	P0PCCR	P0PCCT	P0USE	P0MMOD	P0PA25	P0PA24	P0REG
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

The area 6 general control register (PCC0GCR) is an 8-bit readable/writable register which controls the external buffer, resets, address A25 and A24 pins, and $\overline{\text{REG}}$ pin, and sets the PC card type for the PC card connected to area 6. PCC0GCR is initialized by a power-on reset but retains its value in a manual reset and in software standby mode.

Bit 7—PCC0 Buffer Control (P0DRVE): Controls the external buffer for the PC card connected to area 6.

Bit 7: P0DRVE	Description
0	High-level setting for control $\overline{\text{PCC0DRV}}$ pin of the external buffer for the PC card connected to area 6 (Initial value)
1	Low-level setting for control $\overline{\text{PCC0DRV}}$ pin of the external buffer for the PC card connected to area 6

Bit 6—PCC0 Card Reset (P0PCCR): Controls resets for the PC card connected to area 6.

Bit 6: P0PCCR	Description
0	Low-level setting for reset PCC0RESET pin for the PC card connected to area 6 (Initial value)
1	High-level setting for reset PCC0RESET pin for the PC card connected to area 6

Bit 5—PCC0 Card Type (P0PCCT): Specifies the type of the PC card connected to area 6. Cleared to 0 when the PC card is the IC memory card interface type; set to 1 when the PC card is the I/O card interface type.

Bit 5: P0PCCT	Description
0	The PC card connected to area 6 is handled as the IC memory card interface type (Initial value)
1	The PC card connected to area 6 is handled as the I/O card interface type

Bit 4—PCC0 USE/NOT USE (P0USE): Specifies that the PC Card Controller to be work or not work.

Bit 4: P0USE	Description
0	PC Card Controller doesn't work.
1	PC Card Controller works.

Bit 3—PCC0 Mode (P0MMOD): Controls $\overline{\text{PCCREG}}$ and A24 pins for the PC card connected to area 6. Specifies either A24 of the address to be accessed or bit P0REG for outputting to $\overline{\text{PCCREG}}$ pin. When the common memory space is accessed, specifies either A24 of the address to be accessed or bit P0PA24 for outputting to A24 pin. By this operation, continuous 32 or 16 Mbytes can be selected for the address area of the common memory space of the PC card.

Bit 3: P0MMOD	Description
0	Bit P0REG is output to $\overline{\text{PCCREG}}$ pin, and A24 of address to be accessed is output to A24 pin (continuous 32-MB area mode) (Initial value)
1	A24 of address to be accessed is output to $\overline{\text{PCCREG}}$ pin. When the common memory space is accessed, P0PA24 is output to A24 pin (continuous 16-MB area mode)

Bit 2—PC Card Address (P0PA25): Controls A25 pin for the PC card connected to area 6. When the common memory space is accessed for the PC card connected to area 6, this bit is output to A25 pin. When the attribute memory space or I/O space is accessed, this bit is meaningless.

Bit 2: P0PA25	Description
0	When the common memory space is accessed for the PC card connected to area 6, 0 is output to A25 pin (Initial value)
1	When the common memory space is accessed for the PC card connected to area 6, 1 is output to A25 pin

Bit 1—PC Card Address (P0PA24): Controls A24 pin for the PC card connected to area 6. When bit P0MMOD is 1 and the common memory space is accessed for the PC card connected to area 6, this bit is output to A24 pin. When bit P0MMOD is 0 or the attribute memory space or I/O space is accessed, this bit is meaningless.

Bit 1: P0PA24	Description
0	When bit P0MMOD is 1 and the common memory space is accessed for the PC card connected to area 6, 0 is output to A24 pin (Initial value)
1	When bit P0MMOD is 1 and the common memory space is accessed for the PC card connected to area 6, 1 is output to A24 pin

Bit 0—PCC0REG Space Indication (P0REG): Controls $\overline{\text{PCCREG}}$ pin for the PC card connected to area 6. When bit P0MMOD is 0, this bit is output to $\overline{\text{PCCREG}}$ pin for the PC card connected to area 6. When bit P0MMOD is 1 or the I/O card interface is accessed, this bit is meaningless.

Bit 0: P0REG	Description
0	When bit P0MMOD is 0 and the PC card connected to area 6 is accessed, 0 is output to $\overline{\text{PCCREG}}$ pin (Initial value)
1	When bit P0MMOD is 0 and the PC card connected to area 6 is accessed, 1 is output to $\overline{\text{PCCREG}}$ pin

30.2.3 Area 6 Card Status Change Register (PCC0CSCR)

Bit:	7	6	5	4	3	2	1	0
Bit name:	P0SCDI	—	P0IREQ	P0SC	P0CDC	P0RC	P0BW	P0BD
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

The area 6 card status change register (PCC0CSCR) is an 8-bit readable/writable able register. PCC0CSCR bits are set to 1 by interrupt sources of the PC card connected to area 6 (only bit 7 can be set to 1 as required). PCC0CSCR is initialized by a power-on reset but retains its value in a manual reset and in software standby mode.

Bit 7—PCC0 Software Card Detect Change Interrupt (P0SCDI): A PCC0 software card detect change interrupt can be generated by writing 1 to this bit. When this bit is set to 1, the same interrupt as the PCC0 card detect change interrupt (bit 3 set status) occurs if bit 3 (PCC0 card detect change enable) in the area 6 card status change interrupt enable register (PCC0CSCIER) is set to 1. If bit 3 is cleared to 0, no interrupt occurs.

Bit 7: P0SCDI	Description
0	No software card detect change interrupt occurs for the PC card connected to area 6 (Initial value)
1	Software card detect change interrupt occurs for the PC card connected to area 6

Bit 6—Reserved: Always reads 0. The write value should always be 0.

Bit 5—PCC0IREQ Request (P0IREQ): Indicates the interrupt request for the $\overline{\text{IREQ}}$ pin of the PC card when the PC card connected to area 6 is the I/O card interface type. The P0IREQ bit is set to 1 when an interrupt request signal in pulse mode or level mode is input to the $\overline{\text{IREQ}}$ pin. The mode is selected by bits 5 and 6 (PCC0IREQ interrupt enable bits) in the area 6 card status change interrupt enable register (PCC0CSCIER). This bit can be cleared to 0 only in pulse mode. Write 0 to bit 5 to clear the bit to 0. This bit is not changed if 1 is written. In level mode, bit 5 is a read-only bit which reflects the $\overline{\text{IREQ}}$ pin state (if the $\overline{\text{IREQ}}$ pin is low, 1 is read). This bit always reads 0 on the IC memory card interface.

Bit 5: IREQ	Description
0	No interrupt request on the $\overline{\text{IREQ}}$ pin of the PC card when the PC card is on the I/O card interface (Initial value)
1	An interrupt request on the $\overline{\text{IREQ}}$ pin of the PC card has occurred when the PC card is on the I/O card interface

Bit 4—PCC0 Status Change (P0SC): Indicates a change in the value of the $\overline{\text{STSCHG}}$ pin of the PC card when the PC card connected to area 6 is the I/O card interface type. When the $\overline{\text{STSCHG}}$ pin is changed from 1 to 0, the SC bit is set to 1. When $\overline{\text{STSCHG}}$ pin is not changed, the P0SC bit remains at 0. Write 0 to bit 4 when this bit is set to 1 in order to clear this bit to 0. This bit is not changed if 1 is written. This bit always reads 0 on the IC memory card interface.

Bit 4: P0SC	Description
0	$\overline{\text{STSCHG}}$ pin of the PC card is not changed when the PC card is on the I/O card interface (Initial value)
1	$\overline{\text{STSCHG}}$ pin of the PC card is changed from 1 to 0 when the PC card is on the I/O card interface

Bit 3—PCC0 Card Detect Change (P0CDC): Indicates a change in the value of the $\overline{\text{CD1}}$ and $\overline{\text{CD2}}$ pins in the PC card connected to area 6. When the $\overline{\text{CD1}}$ and $\overline{\text{CD2}}$ values are changed, the P0CDC bit is set to 1. When the values are not changed, the P0CDC bit remains at 0. Write 0 to bit 3 in order to clear this bit to 0. This bit is not changed if 1 is written.

Bit 3: P0CDC	Description
0	$\overline{\text{CD1}}$ and $\overline{\text{CD2}}$ pins in the PC card are not changed (Initial value)
1	$\overline{\text{CD1}}$ and $\overline{\text{CD2}}$ pins in the PC card are changed

Bit 2—PCC0 Ready Change (P0RC): Indicates a change in the value of the RDY/$\overline{\text{BSY}}$ pin of the PC card when the PC card connected to area 6 is the IC memory card interface type. When the RDY/$\overline{\text{BSY}}$ pin is changed from 0 to 1, the P0RC bit is set to 1. When the RDY/$\overline{\text{BSY}}$ pin is not changed, the P0RC bit remains at 0. Write 0 to bit 2 in order to clear this bit to 0. This bit is not changed if 1 is written. This bit always reads 0 on the I/O card interface.

Bit 2: P0RC	Description
0	RDY/$\overline{\text{BSY}}$ pin in the PC card is not changed when the PC card is on the IC memory card interface (Initial value)
1	RDY/$\overline{\text{BSY}}$ pin in the PC card is changed from 0 to 1 when the PC card is on the IC memory card interface

Bit 1—PCC0 Battery Warning (P0BW): Indicates whether the BVD2 and BVD1 pins of the PC card are in the state in which "the battery must be changed although the data is guaranteed" when the PC card connected to area 6 is on the IC memory card interface. When the BVD2 and BVD1 pins are 0 and 1, respectively, the P0BW bit is set to 1; in other cases, the P0BW bit remains at 0. This bit is updated when the BVD2 and BVD1 pins are changed. Write 0 to bit 1 in order to clear this bit to 0. This bit is not changed if 1 is written. This bit always reads 0 on the I/O card interface.

Bit 1: P0BW	Description
0	BVD2 and BVD1 of the PC card are not in the battery warning state when the PC card is in the IC memory card interface (Initial value)
1	BVD2 and BVD1 of the PC card are in the battery warning state and "the battery must be changed although the data is guaranteed" when the PC card is on the IC memory card interface

Bit 0—PCC0 Battery Dead (P0BD): Indicates whether the BVD2 and BVD1 pins of the PC card are in the state in which "the battery must be changed since the data is not guaranteed" when the PC card connected to area 6 is on the IC memory card interface. When the BVD2 and BVD1 pins are 1 and 0 or 0 and 0, the P0BD bit is set to 1; in other cases, the P0BD bit remains at 0. This bit is updated when the BVD2 and BVD1 pins are changed. Write 0 to bit 0 in order to clear this bit to 0. This bit is not changed if 1 is written. This bit always reads 0 on the I/O card interface.

Bit 0: P0BD	Description
0	BVD2 and BVD1 of the PC card are not in the state in which "the battery must be changed since the data is not guaranteed" when the PC card is on the IC memory card interface (Initial value)
1	BVD2 and BVD1 of the PC card are in the state in which "the battery must be changed since the data is not guaranteed" when the PC card is on the IC memory card interface

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

Bit:	7	6	5	4	3	2	1	0
Bit name:	P0CRE	IREQE1	IREQE0	P0SCE	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

The area 6 card status change interrupt enable register (PCC0CSCIER) is an 8-bit readable/writable register. PCC0CSCIER enables or disables interrupt requests for interrupt sources for the PC card connected to area 6. When a PCC0CSCIER is set to 1, the corresponding interrupt is enabled, and when the bit is cleared to 0, the interrupt is disabled. PCC0CSCIER is initialized by a power-on reset but retains its value in a manual reset and in software standby mode.

Bit 7—PCC0 Card Reset Enable (P0CRE): When this bit is set to 1, and when the $\overline{CD1}$ and $\overline{CD2}$ pins detect that a PC card is connected to area 6, the area 6 general control register (PCC0GCR) is initialized.

Bit 7: P0CRE	Description
0	The area 6 general control register (PCC0GCR) is not initialized even if a PC card is detected in area 6 (Initial value)
1	The area 6 general control register (PCC0GCR) is initialized when a PC card is detected connected to area 6

Bits 6 and 5—PCC0IREQ Request Enable (IREQE1, IREQE0): These bits enable or disable $\overline{\text{IREQ}}$ pin interrupt requests and select the interrupt mode when the PC card connected to area 6 is the I/O card interface type. Note that bit 5 (P0IREQ) in the area 6 card status change register (PCC0CSCR) is cleared if the values in bits 6 and 5 in this register are changed. These bits have no meaning on the IC memory card interface.

Bit 6: IREQE1	Bit 5: IREQE0	Description
0	0	IREQ requests are not accepted for the PC card connected to area 6. Bit 5 in the status change register (PCC0CSCR) functions as a read-only bit that indicates the inverse of the $\overline{\text{IREQ}}$ pin signal. (Initial value)
0	1	The level-mode IREQ interrupt request signal is accepted for the PC card connected to area 6. In level mode, an interrupt occurs when level 0 of the signal input from the $\overline{\text{IREQ}}$ pin is detected.
1	0	The pulse-mode IREQ interrupt request signal is accepted for the PC card connected to area 6. In pulse mode, an interrupt occurs when a falling edge from 1 to 0 of the signal input from the $\overline{\text{IREQ}}$ pin is detected.
1	1	The pulse-mode IREQ interrupt request signal is accepted for the PC card connected to area 6. In pulse mode, an interrupt occurs when a rising edge from 0 to 1 of the signal input from the $\overline{\text{IREQ}}$ pin is detected.

Bit 4—PCC0 Status Change Enable (P0SCE): When the PC card connected to area 6 is on the I/O card interface, bit 4 enables or disables the interrupt request when the value of the BVD1 pin ($\overline{\text{STSCHG}}$ pin) is changed. This bit has no meaning in the IC memory card interface.

Bit 4: P0SCE	Description
0	No interrupt occurs for the PC card connected to area 6 regardless of the value of the BVD1 pin ($\overline{\text{STSCHG}}$ pin) (Initial value)
1	An interrupt occurs for the PC card connected to area 6 when the value of the BVD1 pin ($\overline{\text{STSCHG}}$ pin) is changed from 1 to 0

Bit 3—PCC0 Card Detect Change Enable (P0CDE): Bit 3 enables or disables the interrupt request when the values of the $\overline{\text{CD1}}$ and $\overline{\text{CD2}}$ pins are changed.

Bit 3: P0CDE	Description
0	No interrupt occurs for the PC card connected to area 6 regardless of the values of the $\overline{\text{CD1}}$ and $\overline{\text{CD2}}$ pins (Initial value)
1	An interrupt occurs for the PC card connected to area 6 when the values of the $\overline{\text{CD1}}$ and $\overline{\text{CD2}}$ pins are changed

Bit 2—PCC0 Ready Change Enable (P0RE): When the PC card connected to area 6 is on the IC memory card interface, bit 2 enables or disables the interrupt request when the value of the RDY/$\overline{\text{BSY}}$ pin is changed. This bit has no meaning on the I/O card interface.

Bit 2: P0RE	Description
0	No interrupt occurs for the PC card connected to area 6 regardless of the value of the RDY/$\overline{\text{BSY}}$ pin (Initial value)
1	An interrupt occurs for the PC card connected to area 6 when the value of the RDY/$\overline{\text{BSY}}$ pin is changed from 0 to 1

Bit 1—PCC0 Battery Warning Enable (P0BWE): When the PC card connected to area 6 is on the IC memory card interface, bit 1 enables or disables the interrupt request when the BVD2 and BVD1 pins are in the state in which "the battery must be changed although the data is guaranteed". This bit has no meaning on the I/O card interface.

Bit 1: P0BWE	Description
0	No interrupt occurs when the BVD2 and BVD1 pins are in the state in which "the battery must be changed although the data is guaranteed" (Initial value)
1	An interrupt occurs when the BVD2 and BVD1 pins are in the state in which "the battery must be changed although the data is guaranteed"

Bit 0—PCC0 Battery Dead Enable (P0BDE): When the PC card connected to area 6 is on the IC memory card interface, bit 0 enables or disables the interrupt request when the BVD2 and BVD1 pins are in the state in which "the battery must be changed since the data is not guaranteed". This bit has no meaning on the I/O card interface.

Bit 0: P0BDE	Description
0	No interrupt occurs when the BVD2 and BVD1 pins are in the state in which "the battery must be changed since the data is not guaranteed" (Initial value)
1	An interrupt occurs when the BVD2 and BVD1 pins are in the state in which "the battery must be changed since the data is not guaranteed"

30.3 Operation

30.3.1 PC card Connection Specification (Interface Diagram, Pin Correspondence)

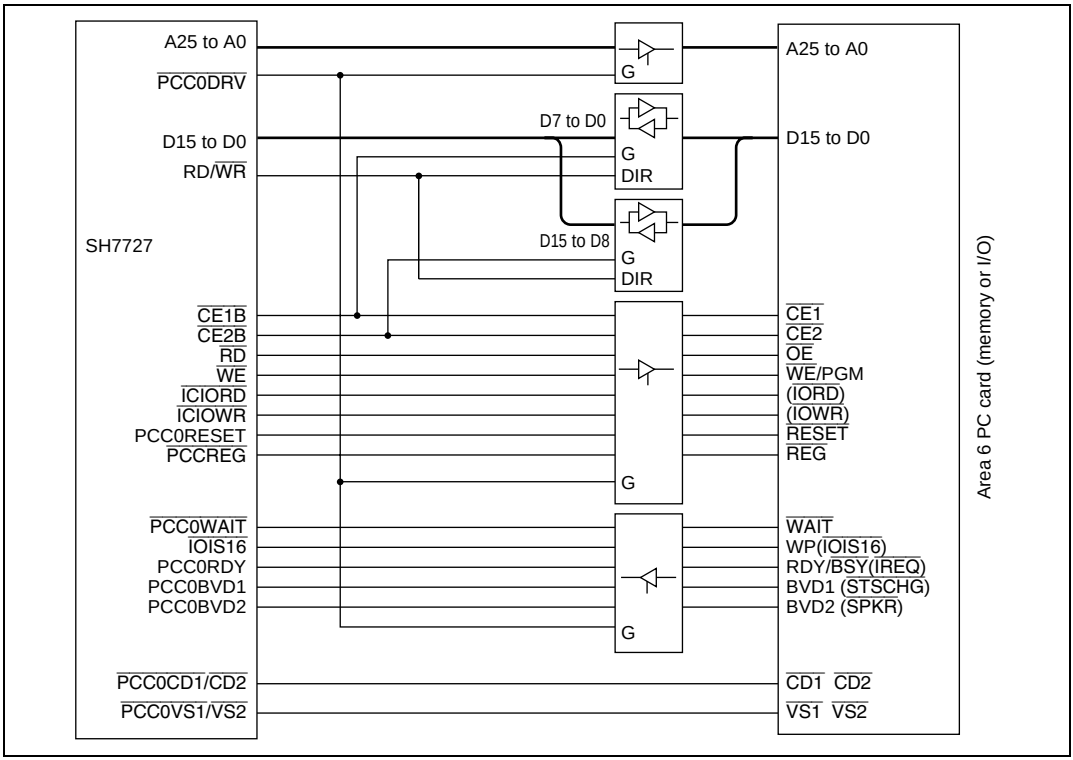


Figure 30.4 SH7727 Interface

Table 30.3 PCMCIA Support Interface

	IC Memory Card Interface			I/O Card Interface			SH7727 Corresponding Pin
Pin	**Signal Name**	**I/O**	**Function**	**Signal Name**	**I/O**	**Function**	
1	GND		Ground	GND		Ground	—
2	D3	I/O	Data	D3	I/O	Data	D3
3	D4	I/O	Data	D4	I/O	Data	D4
4	D5	I/O	Data	D5	I/O	Data	D5
5	D6	I/O	Data	D6	I/O	Data	D6
6	D7	I/O	Data	D7	I/O	Data	D7
7	$\overline{CE1}$	I	Card enable	$\overline{CE1}$	I	Card enable	CE1B
8	A10	I	Address	A10	I	Address	A10
9	$\overline{OE}$	I	Output enable	$\overline{OE}$	I	Output enable	RD
10	A11	I	Address	A11	I	Address	A11
11	A9	I	Address	A9	I	Address	A9
12	A8	I	Address	A8	I	Address	A8
13	A13	I	Address	A13	I	Address	A13
14	A14	I	Address	A14	I	Address	A14
15	$\overline{WE}$/$\overline{PGM}$	I	Write enable	$\overline{WE}$/$\overline{PGM}$	I	Write enable	$\overline{WE}$
16	RDY/$\overline{BSY}$	O	Ready/busy	$\overline{IREQ}$	O	Interrupt request	PCC0RDY
17	VCC		Power supply	VCC		Power supply	—
18	VPP1		Programming power supply	VPP1		Programming and peripheral power supply	—
19	A16	I	Address	A16	I	Address	A16
20	A15	I	Address	A15	I	Address	A15
21	A12	I	Address	A12	I	Address	A12
22	A7	I	Address	A7	I	Address	A7
23	A6	I	Address	A6	I	Address	A6
24	A5	I	Address	A5	I	Address	A5
25	A4	I	Address	A4	I	Address	A4
26	A3	I	Address	A3	I	Address	A3
27	A2	I	Address	A2	I	Address	A2
28	A1	I	Address	A1	I	Address	A1

	IC Memory Card Interface			I/O Card Interface			SH7727
Pin	Signal Name	I/O	Function	Signal Name	I/O	Function	Corresponding Pin
29	A0	I	Address	A0	I	Address	A0
30	D0	I/O	Data	D0	I/O	Data	D0
31	D1	I/O	Data	D1	I/O	Data	D1
32	D2	I/O	Data	D2	I/O	Data	D2
33	WP	O	Write protect	$\overline{\text{IOIS16}}$	O	16-bit I/O port	$\overline{\text{IOIS16}}$
34	GND		Ground	GND		Ground	—
35	GND		Ground	GND		Ground	—
36	$\overline{\text{CD1}}$	O	Card detection	$\overline{\text{CD1}}$	O	Card detection	$\overline{\text{PCC0CD1}}$
37	D11	I/O	Data	D11	I/O	Data	D11
38	D12	I/O	Data	D12	I/O	Data	D12
39	D13	I/O	Data	D13	I/O	Data	D13
40	D14	I/O	Data	D14	I/O	Data	D14
41	D15	I/O	Data	D15	I/O	Data	D15
42	$\overline{\text{CE2}}$	I	Card enable	$\overline{\text{CE2}}$	I	Card enable	CE2B
43	$\overline{\text{VS1}}$	O	Voltage sense	$\overline{\text{VS1}}$	O	Voltage sense	$\overline{\text{PCC0VS1}}$
44	RFU		Reserved	$\overline{\text{IORD}}$	I	I/O read	ICIORD
45	RFU		Reserved	$\overline{\text{IOWR}}$	I	I/O write	ICIOWR
46	A17	I	Address	A17	I	Address	A17
47	A18	I	Address	A18	I	Address	A18
48	A19	I	Address	A19	I	Address	A19
49	A20	I	Address	A20	I	Address	A20
50	A21	I	Address	A21	I	Address	A21
51	VCC		Power supply	VCC		Power supply	—
52	VPP2		Programming power supply	VPP2		Programming and peripheral power supply	—
53	A22	I	Address	A22	I	Address	A22
54	A23	I	Address	A23	I	Address	A23
55	A24	I	Address	A24	I	Address	A24
56	A25	I	Address	A25	I	Address	A25
57	$\overline{\text{VS2}}$	O	Voltage sense	$\overline{\text{VS2}}$	O	Voltage sense	$\overline{\text{PCC0VS2}}$

	IC Memory Card Interface			I/O Card Interface			SH7727 Corresponding Pin
Pin	**Signal Name**	**I/O**	**Function**	**Signal Name**	**I/O**	**Function**	
58	RESET	I	Reset	RESET	I	Reset	PCC0RESET
59	$\overline{\text{WAIT}}$	O	Wait request	$\overline{\text{WAIT}}$	O	Wait request	$\overline{\text{PCC0WAIT}}$
60	RFU		Reserved	$\overline{\text{INPACK}}$	O	Input acknowledge	—
61	$\overline{\text{REG}}$	I	Attribute memory space select	$\overline{\text{REG}}$	I	Attribute memory space select	$\overline{\text{PCCREG}}$
62	BVD2	O	Battery voltage detection	$\overline{\text{SPKR}}$	O	Digital sound signal	PCC0BVD2
63	BVD1	O	Battery voltage detection	$\overline{\text{STSCHG}}$	O	Card status change	PCC0BVD1
64	D8	I/O	Data	D8	I/O	Data	D8
65	D9	I/O	Data	D9	I/O	Data	D9
66	D10	I/O	Data	D10	I/O	Data	D10
67	$\overline{\text{CD2}}$	O	Card detection	$\overline{\text{CD2}}$	O	Card detection	$\overline{\text{PCC0CD2}}$
68	GND		Ground	GND		Ground	—

30.3.2 PC Card Interface Timing

(1) Memory card interface timing

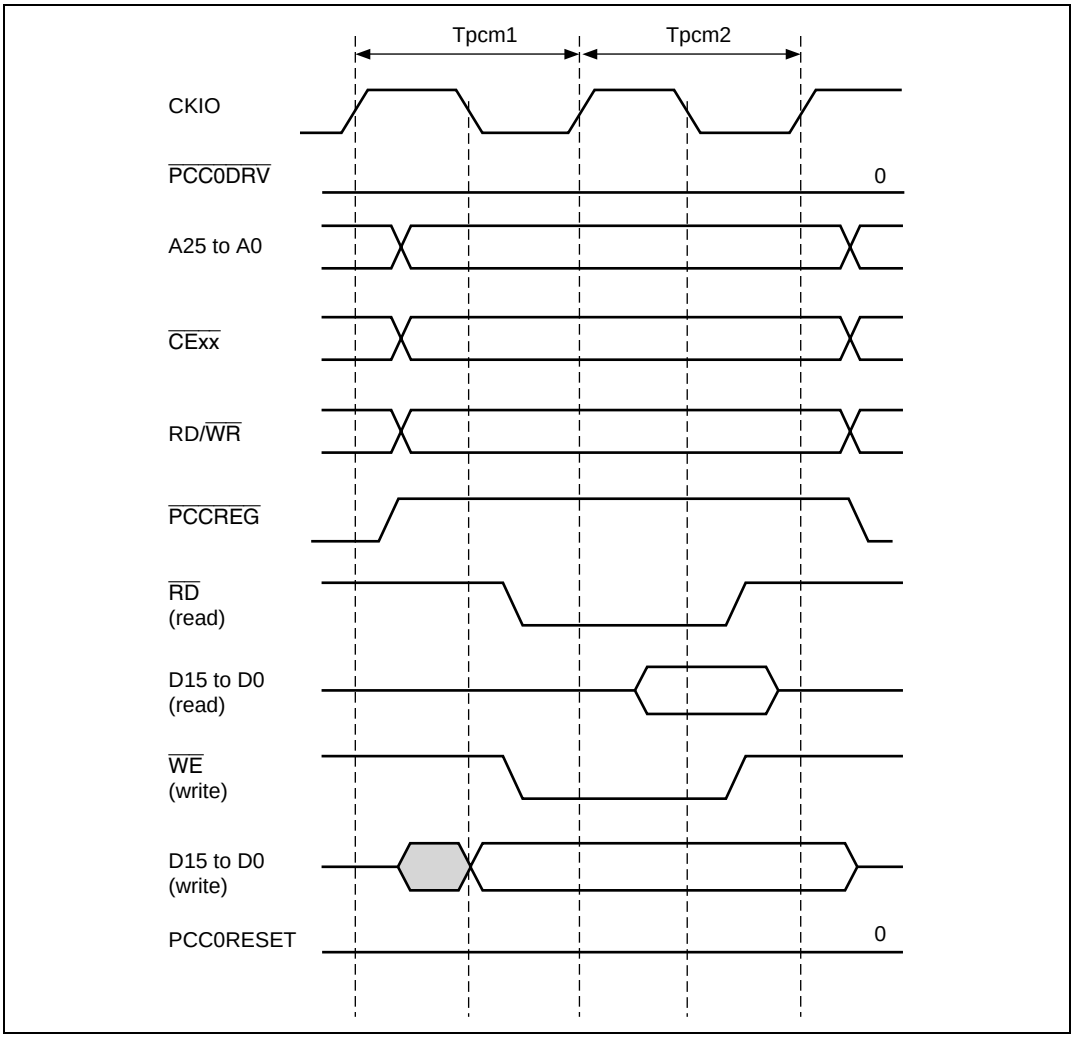


Figure 30.5 PCMCIA Memory Card Interface Basic Timing

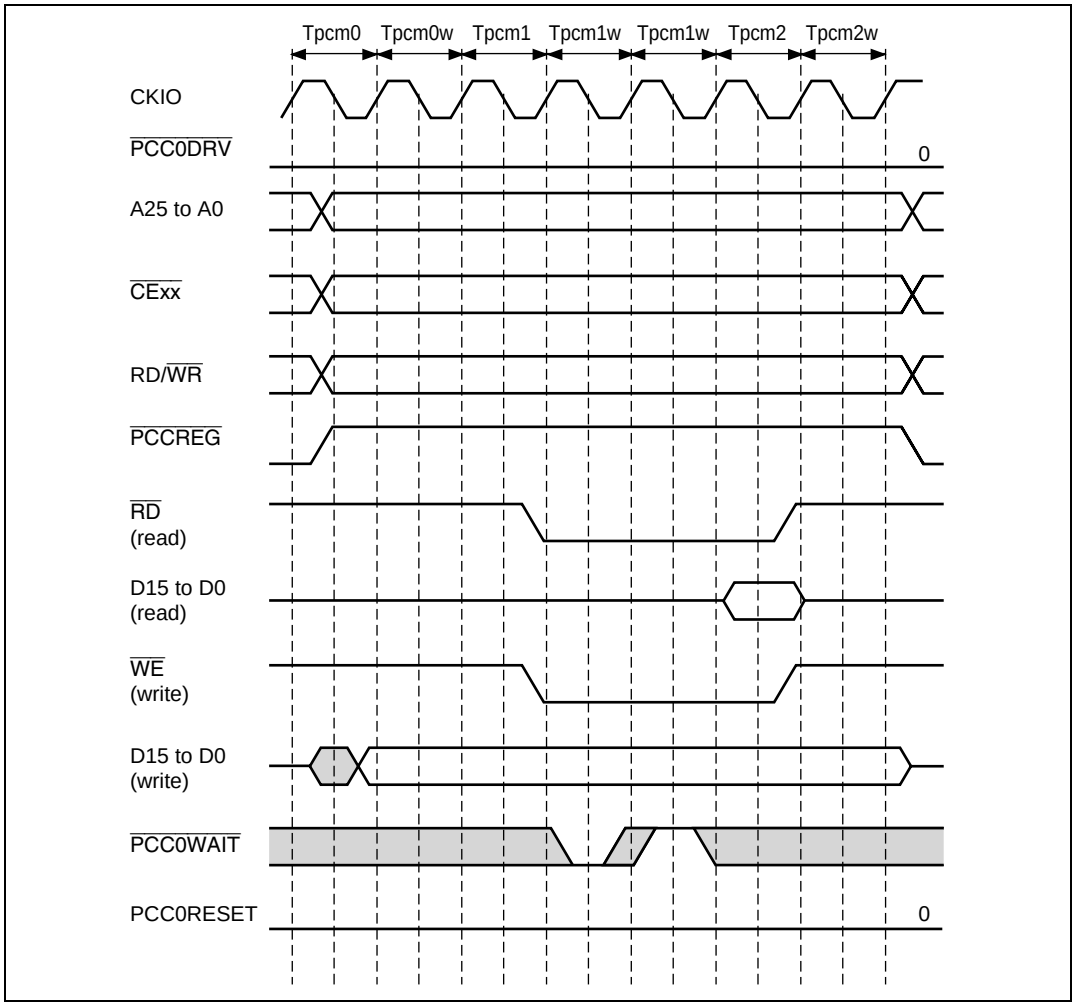


Figure 30.6 PCMCIA Memory Card Interface Wait Timing

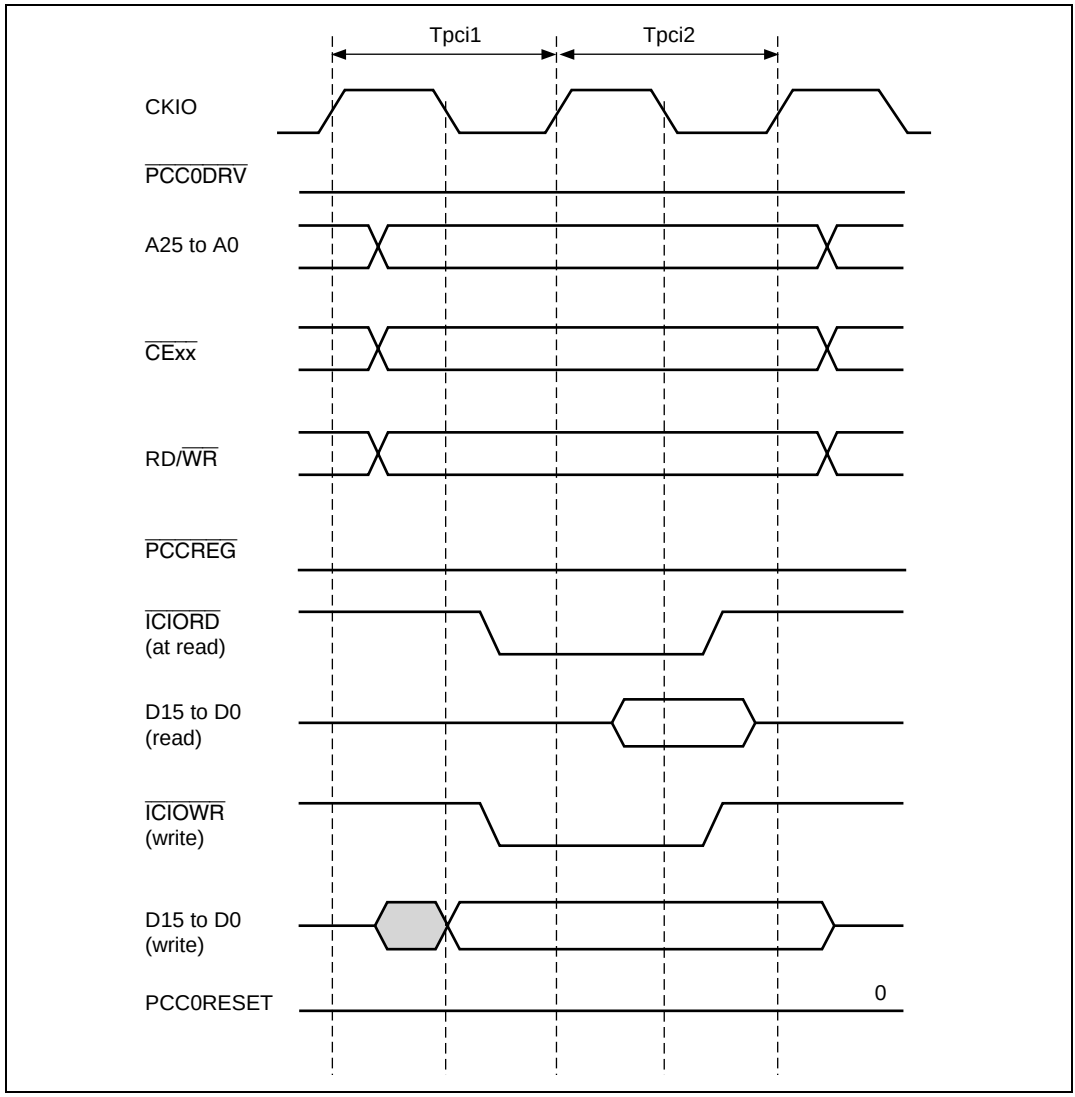


Figure 30.7 PCMCIA I/O Card Interface Basic Timing

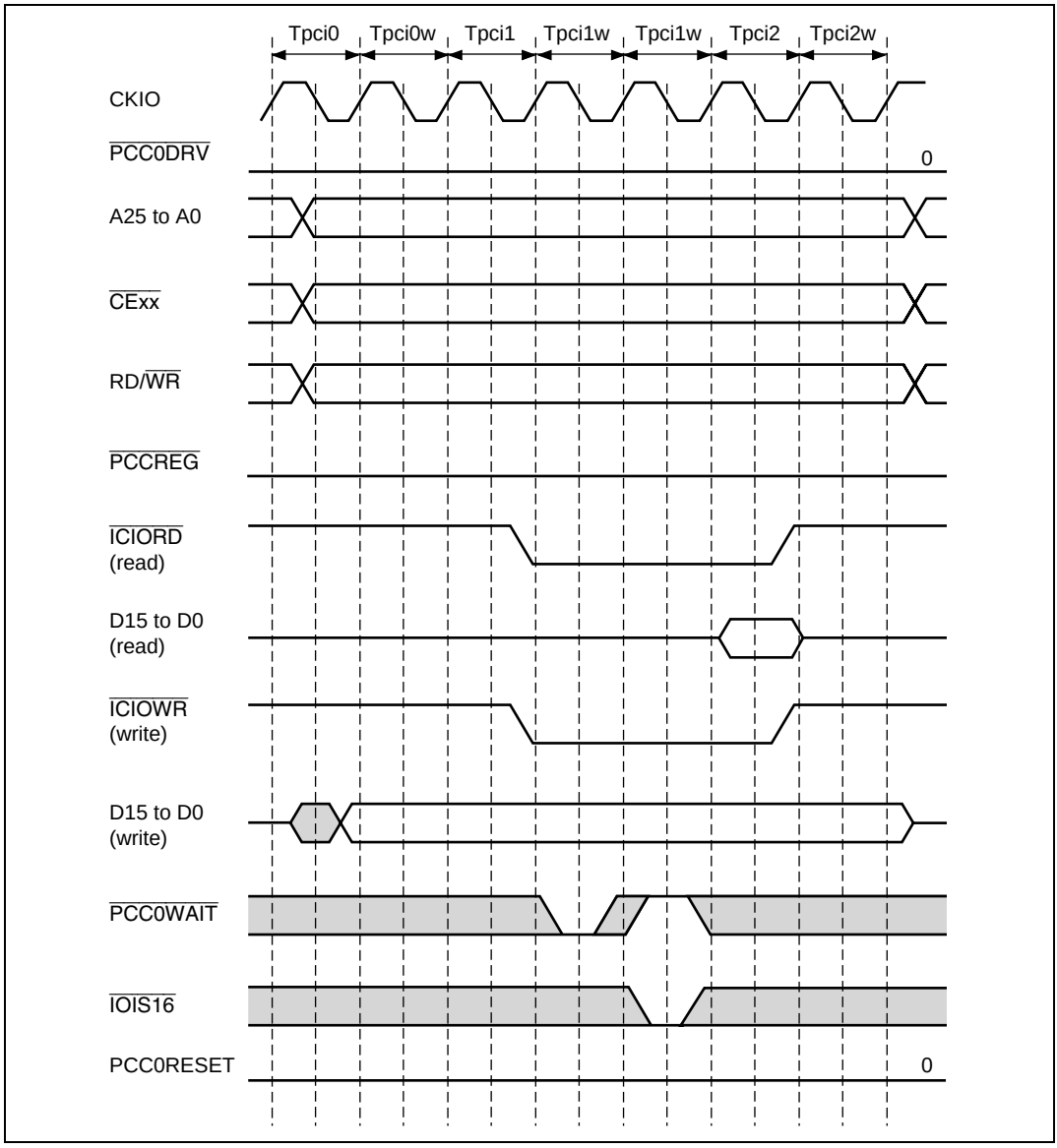


Figure 30.8 PCMCIA I/O Card Interface Wait Timing

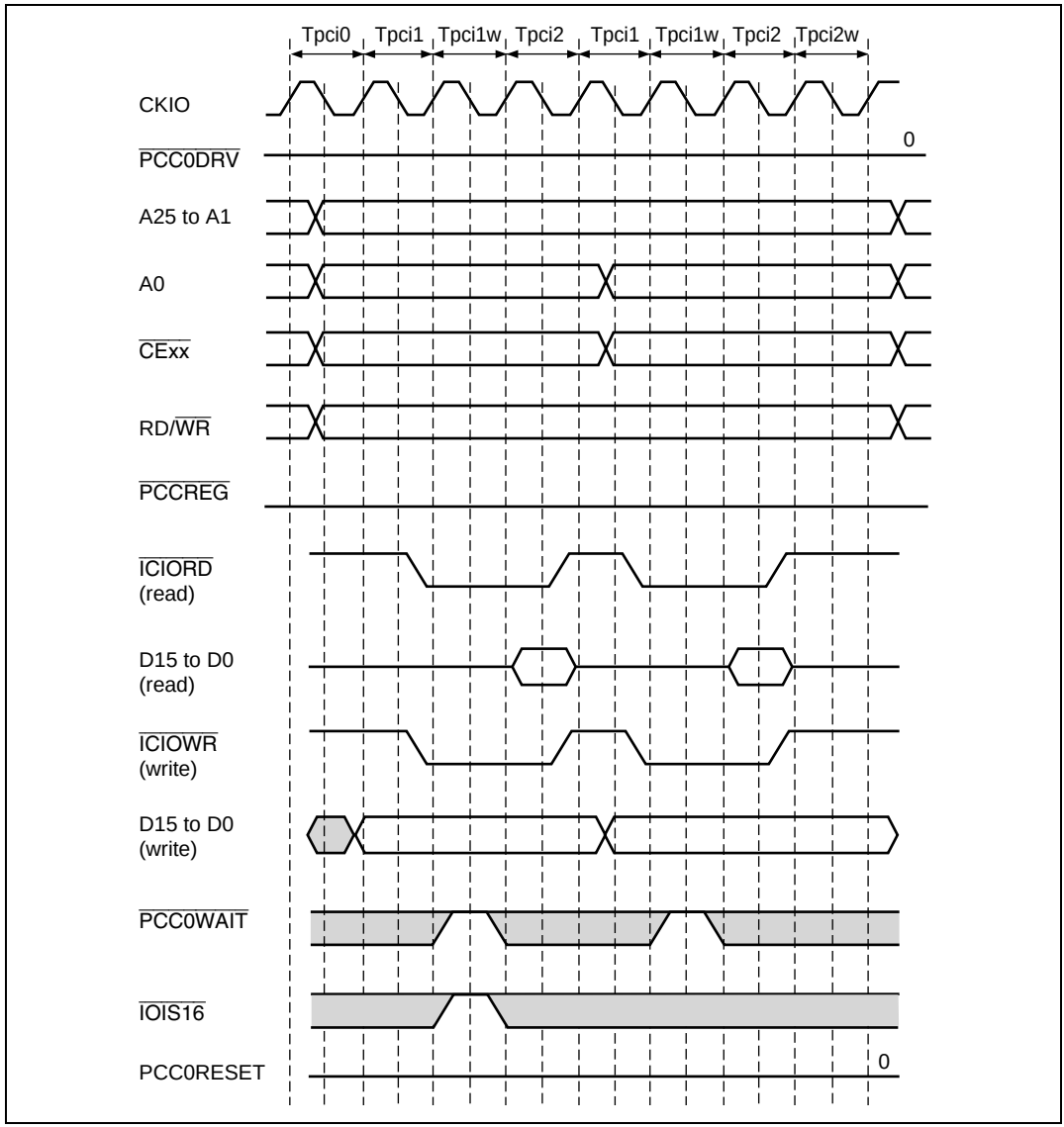


Figure 30.9 Dynamic Bus Sizing Timing for PCMCIA I/O Card Interface

Refer to section 12, Bus State Controller (BSC), and section 32, Electrical Characteristics, for more details.

30.3.3 Usage Notes

External Bus Frequency Limit when Using PC Card: According to the PC card standard, the attribute memory access time is specified as 600 ns (3.3 V)/300 ns (5 V). Therefore, when the SH7727 accesses attribute memory, the bus cycle must be coordinated with the PC card interface timing. In the SH7727, the timing can be adjusted by setting the TED and TEH values in the PCR register, and the number of waits and number of idle states in the WCR1 and WCR2 registers, allowing a PC card to be used within the above frequency ranges.

The common memory access time and I/O access time (based on the IORD and IOWR signals) are also similarly specified (see table below), and a PC card must be used within the above ranges in order to satisfy all these specifications.

PC Card Space	Access Time (5 V Operation)	Access Time (3.3 V Operation)
Attribute memory	300 ns	600 ns
Common memory	250 ns	600 ns
I/O space (–$\overline{\text{IORD}}$/–$\overline{\text{IOWR}}$ pulse width)	165 ns	165 ns

Pin Function Control and Card Type Switching: When setting pin function controller pin functions to dedicated PC card use ("other function"), the disabled state should first be set in the card status change interrupt enable register (PCC0CSCIER). Also, the card status change register (PCC0CSCR) must be cleared after the setting has been made. However, this restriction does not apply to the card detection pins ($\overline{\text{CD1}}$, $\overline{\text{CD2}}$).

When changing the card type bit (P0PCCT) in the area 6 general control register (PCC0GCR), the disabled state should first be set in the card status change interrupt enable register (PCC0CSCIER). Also, the card status change register (PCC0CSCR) must be cleared after the setting has been made.

Reason: When PC card controller settings are modified, the functions of PC card pins that generate various interrupts change, with the result that unnecessary interrupts may be generated.

Setting Procedure when Using PC Card Controller: The following steps should be followed when using a card controller:

1. Drive pin $\overline{\text{ASEMD0}}$ high.
2. Set bit 0 (A6PCM) in bus control register 1 (BCR1) in the bus state controller to 1.
3. Set bit 4 (P0USE) in the area 6 general control register in the PC card controller to 1.
4. Set the pin function controller to custom PC card pin functions (“other functions”).

Section 31 User-Debugging Interface (H-UDI)

31.1 Overview

The SH7727 incorporates a user-debugging interface (H-UDI) and advanced user debugger (AUD) for program debugging.

31.2 User Debugging Interface (H-UDI)

The H-UDI (user-debugging interface) performs on-chip debugging which is supported by the SH7727. The H-UDI described here is a serial interface which is compatible with JTAG (Joint Test Action Group, IEEE Standard 1149.1 and IEEE Standard Test Access Port and Boundary-Scan Architecture) specifications.

The H-UDI in the SH7727 supports a boundary scan mode, and is also used for emulator connection.

When using an emulator, H-UDI functions should not be used. Refer to the emulator manual for the method of connecting the emulator.

31.2.1 Pin Description

TCK: H-UDI serial data input/output clock pin. Data is serially supplied to the H-UDI from the data input pin (TDI), and output from the data output pin (TDO), in synchronization with this clock.

TMS: Mode select input pin. The state of the TAP control circuit is determined by changing this signal in synchronization with TCK. The protocol conforms to the JTAG standard (IEEE Std. 1149.1).

$\overline{\text{TRST}}$: H-UDI reset input pin. Input is accepted asynchronously with respect to TCK, and when low, the H-UDI is reset. See section 31.4.2, Reset Configuration, for more information.

TDI: H-UDI serial data input pin. Data transfer to the H-UDI is executed by changing this signal in synchronization with TCK.

TDO: H-UDI serial data output pin. Data output from the H-UDI is executed by reading this signal in synchronization with TCK.

$\overline{\text{ASEMD0}}$: The ASE mode select pin. If a low level is input to the $\overline{\text{ASEMD0}}$ pin while the $\overline{\text{RESETP}}$ pin is asserted, ASE mode is entered, and if a high level is input, normal mode is

entered. When using the user system alone and not using an emulator or the H-UDI, a high level should be input. In ASE mode, boundary scan and emulator functions can be used. Hold the input level to the $\overline{\text{ASEMD0}}$ pin for at least one cycle after the $\overline{\text{RESETP}}$ pin is negated.

$\overline{\text{ASEBRKAK}}$: Dedicated emulator pin.

31.2.2 Block Diagram

Figure 31.1 shows the block diagram of the H-UDI.

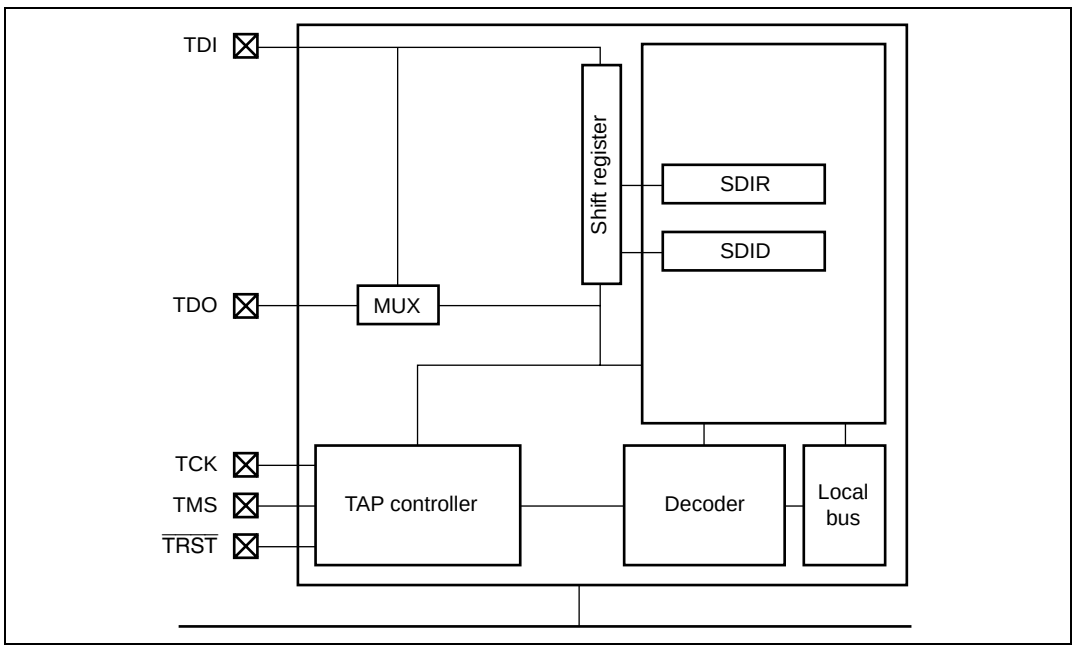


Figure 31.1 H-UDI Block Diagram

31.3 Register Descriptions

The H-UDI has the following registers.

- SDIR: instruction register
- SDID: device identification register
- SDBSR: boundary-scan register

Table 31.1 shows H-UDI register configuration.

Table 31.1 H-UDI Registers

		CPU Side			H-UDI Side		Initial
Name	**Abbreviation**	**R/W**	**Size**	**Address**	**R/W**	**Size**	**Value***
Instruction register	SDIR	R	16	H'04000200	R/W	16	H'FFFF
Device Identification register	SDID	—	—	—	R	32	H'0004200F
Boundary-scan register	SDBSR	—	—	—	R/W	—	Undefined

Note: * Initialized when $\overline{\text{TRST}}$ pin is low or when TAP is in the test-logic-reset state.

31.3.1 Bypass Register (SDBPR)

The bypass register (SDBPR) is a 1-bit register that cannot be accessed by the CPU. Setting the SDIR register to the bypass mode makes the SDBPR register to be connected between the TDI and TDO H-UDI pins.

31.3.2 Instruction Register (SDIR)

The instruction register (SDIR) is a 16-bit read-only register. The register is in bypass mode in its initial state. It is initialized by $\overline{\text{TRST}}$ or in the TAP test-logic-reset state, and can be written by the H-UDI irrespective of the CPU mode. Operation is not guaranteed when a reserved command is set to this register.

Bit:	15	14	13	12	11	10	9	8
	TI3	TI2	TI1	TI0	—	—	—	—
Initial value:	1	1	1	1	1	1	1	1

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

Bits 15 to 12—Test Instruction Bits (TI3 to TI0): Cannot be written by the CPU.

Table 31.2 H-UDI Commands

TI3	TI2	TI1	TI0	Description
0	0	0	0	EXTEST
0	1	0	0	SAMPLE/PRELOAD
0	1	0	1	Reserved
0	1	1	0	H-UDI reset negate
0	1	1	1	H-UDI reset assert
1	0	0	—	Reserved
1	0	1	—	H-UDI interrupt
1	1	0	—	Reserved
1	1	1	0	Reserved
1	1	1	1	Bypass mode (initial value)
0	0	0	1	Recovery from sleep

Bits 11 to 0—Reserved: Always read 1.

31.3.3 Boundary-Scan Register (SDBSR)

Boundary-scan register (SDBSR) is a shift register allocated on the PAD to control the I/O pins in this LSI.

The boundary-scan test is available in compliance with the JTAG standard by using the EXTEST and SAMPLE/PRELOAD instructions. Table 31.3 lists the correspondence between the SH7727 pins and the boundary-scan register.

Table 31.3 Correspondence between SH7727 Pins and Boundary-Scan Register

Bit	Pin Name	I/O	Bit	Pin Name	I/O
from TDI			359	MD1	IN
391	D31/PTB7	IN	358	MD2	IN
390	D30/PTB6	IN	357	NMI	IN
389	D29/PTB5	IN	356	IRQ0/$\overline{\text{IRL0}}$/PTH0	IN
388	D28/PTB4	IN	355	IRQ1/$\overline{\text{IRL1}}$/PTH1	IN
387	D27/PTB3	IN	354	IRQ2/$\overline{\text{IRL2}}$/PTH2	IN
386	D26/PTB2	IN	353	IRQ3/$\overline{\text{IRL3}}$/PTH3	IN
385	D25/PTB1	IN	352	IRQ4/PTH4	IN
384	D24/PTB0	IN	351	MD5	IN
383	D23/PTA7	IN	350	$\overline{\text{BREQ}}$	IN
382	D22/PTA6	IN	349	VEPWC	OUT
381	D21/PTA5	IN	348	VCPWC	OUT
380	D20/PTA4	IN	347	$\overline{\text{BACK}}$	OUT
379	D19/PTA3	IN	346	D31/PTB7	OUT
378	D18/PTA2	IN	345	D30/PTB6	OUT
377	D17/PTA1	IN	344	D29/PTB5	OUT
376	D16/PTA0	IN	343	D28/PTB4	OUT
375	D15	IN	342	D27/PTB3	OUT
374	D14	IN	341	D26/PTB2	OUT
373	D13	IN	340	D25/PTB1	OUT
372	D12	IN	339	D24/PTB0	OUT
371	D11	IN	338	D23/PTA7	OUT
370	D10	IN	337	D22/PTA6	OUT
369	D9	IN	336	D21/PTA5	OUT
368	D8	IN	335	D20/PTA4	OUT
367	D7	IN	334	D19/PTA3	OUT
366	D6	IN	333	D18/PTA2	OUT
365	D5	IN	332	D17/PTA1	OUT
364	D4	IN	331	D16/PTA0	OUT
363	D3	IN	330	D15	OUT
362	D2	IN	329	D14	OUT
361	D1	IN	328	D13	OUT
360	D0	IN	327	D12	OUT

Bit	Pin Name	I/O
326	D11	OUT
325	D10	OUT
324	D9	OUT
323	D8	OUT
322	D7	OUT
321	D6	OUT
320	D5	OUT
319	D4	OUT
318	D3	OUT
317	D2	OUT
316	D1	OUT
315	D0	OUT
314	VEPWC	Control
313	VCPWC	Control
312	$\overline{\text{BACK}}$	Control
311	D31/PTB7	Control
310	D30/PTB6	Control
309	D29/PTB5	Control
308	D28/PTB4	Control
307	D27/PTB3	Control
306	D26/PTB2	Control
305	D25/PTB1	Control
304	D24/PTB0	Control
303	D23/PTA7	Control
302	D22/PTA6	Control
301	D21/PTA5	Control
300	D20/PTA4	Control
299	D19/PTA3	Control
298	D18/PTA2	Control
297	D17/PTA1	Control
296	D16/PTA0	Control
295	D15	Control
294	D14	Control
293	D13	Control
292	D12	Control

Bit	Pin Name	I/O
291	D11	Control
290	D10	Control
289	D9	Control
288	D8	Control
287	D7	Control
286	D6	Control
285	D5	Control
284	D4	Control
283	D3	Control
282	D2	Control
281	D1	Control
280	D0	Control
279	$\overline{\text{BS}}$/PTK4	IN
278	$\overline{\text{WE2}}$/DQMUL/$\overline{\text{ICIORD}}$/PTK6	IN
277	$\overline{\text{WE3}}$/DQMUU/$\overline{\text{ICIORD}}$/PTK7	IN
276	$\overline{\text{AUDSYNC}}$/PTE7/PCC0RDY	IN
275	$\overline{\text{CS4}}$/PTK2	IN
274	$\overline{\text{CS5}}$/$\overline{\text{CE1A}}$/PTK3	IN
273	$\overline{\text{CE2A}}$/PTE4	IN
272	$\overline{\text{CE2B}}$/PTE5	IN
271	AFE_HC1/USB1d_DPLS/PTK0	IN
270	$\overline{\text{AFE_RLYCNT}}$/USB1d_DMNS/PTK1	IN
269	AFE_SCLK/USB1d_TXDPLS	IN
268	PTM7/PINT7/AFE_FS/USB1d_RCV	IN
267	PTM6/PINT6/AFE_RXIN/USB1d_SPEED	IN
266	PTM5/PINT5/AFE_TXOUT/USB1d_TXSE0	IN
265	A0	OUT
264	A1	OUT
263	A2	OUT
262	A3	OUT
261	A4	OUT
260	A5	OUT

Bit	Pin Name	I/O
259	A6	OUT
258	A7	OUT
257	A8	OUT
256	A9	OUT
255	A10	OUT
254	A11	OUT
253	A12	OUT
252	A13	OUT
251	A14	OUT
250	A15	OUT
249	A16	OUT
248	A17	OUT
247	A18	OUT
246	A19	OUT
245	A20	OUT
244	A21	OUT
243	A22	OUT
242	A23	OUT
241	A24	OUT
240	A25	OUT
239	$\overline{BS}$/PTK4	OUT
238	$\overline{RD}$	OUT
237	$\overline{WE0}$/DQMLL	OUT
236	$\overline{WE1}$/DQMLU/$\overline{WE}$	OUT
235	$\overline{WE2}$/DQMUL/$\overline{ICIORD}$/PTK6	OUT
234	$\overline{WE3}$/DQMUU/$\overline{ICIOWR}$/PTK7	OUT
233	RD/$\overline{WR}$	OUT
232	$\overline{AUDSYNC}$/PTE7/PCC0RDY	OUT
231	$\overline{CS0}$	OUT
230	$\overline{CS2}$	OUT
229	$\overline{CS3}$	OUT
228	$\overline{CS4}$/PTK2	OUT
227	$\overline{CS5}$/$\overline{CE1A}$/PTK3	OUT
226	$\overline{CS6}$/$\overline{CE1B}$	OUT
225	$\overline{CE2A}$/PTE4	OUT
224	$\overline{CE2B}$/PTE5	OUT
223	AFE_HC1/USB1d_DPLS/PTK0	OUT
222	$\overline{AFE_RLYCNT}$/$\overline{USB1d}$_DMNS/ PTK1	OUT
221	AFE_SCLK/USB1d_TXDPLS	OUT
220	PTM7/PINT7/AFE_FS/ USB1d_RCV	OUT
219	PTM6/PINT6/AFE_RXIN/ USB1d_SPEED	OUT
218	PTM5/PINT5/AFE_TXOUT/ USB1d_TXSE0	OUT
217	A0	Control
216	A1	Control
215	A2	Control
214	A3	Control
213	A4	Control
212	A5	Control
211	A6	Control
210	A7	Control
209	A8	Control
208	A9	Control
207	A10	Control
206	A11	Control
205	A12	Control
204	A13	Control
203	A14	Control
202	A15	Control
201	A16	Control
200	A17	Control
199	A18	Control
198	A19	Control
197	A20	Control
196	A21	Control
195	A22	Control
194	A23	Control
193	A24	Control

Bit	Pin Name	I/O
192	A25	Control
191	$\overline{BS}$/PTK4	Control
190	$\overline{RD}$	Control
189	$\overline{WE0}$/DQMLL	Control
188	$\overline{WE1}$/DQMLU/WE	Control
187	$\overline{WE2}$/DQMUL/$\overline{ICIORD}$/PTK6	Control
186	$\overline{WE3}$/DQMUU/$\overline{ICIOWR}$/PTK7	Control
185	RD/$\overline{WR}$	Control
184	$\overline{AUDSYNC}$/PTE7/PCC0RDY	Control
183	$\overline{CS0}$	Control
182	$\overline{CS2}$	Control
181	$\overline{CS3}$	Control
180	$\overline{CS4}$/PTK2	Control
179	$\overline{CS5}$/$\overline{CE1A}$/PTK3	Control
178	$\overline{CS6}$/$\overline{CE1B}$	Control
177	$\overline{CE2A}$/PTE4	Control
176	$\overline{CE2B}$/PTE5	Control
175	AFE_HC1/USB1d_DPLS/PTK0	Control
174	$\overline{AFE_RLYCNT}$/$\overline{USB1d_DMNS}$/ PTK1	Control
173	AFE_SCLK/USB1d_TXDPLS	Control
172	PTM7/PINT7/AFE_FS/ USB1d_RCV	Control
171	PTM6/PINT6/AFE_RXIN/ USB1d_SPEED	Control
170	PTM5/PINT5/AFE_TXOUT/ USB1d_TXSE0	Control
169	PTM4/PINT4/$\overline{AFE_RDET}$	IN
168	Reserved/USB1d_SUSPEND0	IN
167	$\overline{USB1_ovr_crnt}$/USBF_VBUS	IN
166	$\overline{USB2_ovr_crnt}$	IN
165	$\overline{RTS2}$/USB1d_TXENL	IN
164	PTE2/USB1_PWR_EN	IN
163	PTE1/USB2_PWR_EN	IN
162	CKE/PTK5	IN
161	$\overline{RAS3}$/PTJ0	IN

Bit	Pin Name	I/O
160	Reserved/PTJ1	IN
159	Reserved/$\overline{CAS}$/PTJ2	IN
158	Reserved/PTJ3	IN
157	Reserved/PTJ4	IN
156	ReservedPTJ5	IN
155	CL1/PTD5	IN
154	DON/PTD7	IN
153	M_DISP/PTE6	IN
152	FLM/PTE3	IN
151	$\overline{WAIT}$	IN
150	AUDCK/PTH6/$\overline{PCC0WAIT}$	IN
149	$\overline{IOIS16}$/PTG7	IN
148	$\overline{ASEBRKAK}$/PTG5	IN
147	PTG4	IN
146	AUDATA3/PTG3/PCC0BVD2	IN
145	AUDATA2/PTG2/PCC0BVD1	IN
144	AUDATA1/PTG1/$\overline{PCC0CD2}$	IN
143	AUDATA0/PTG0/$\overline{PCC0CD1}$	IN
142	PTH5/$\overline{ADTRG}$	IN
141	PTF3/PINT11/Reserved	IN
140	PTF2/$\overline{PCCREG}$/Reserved	IN
139	PTF1/$\overline{PCC0VS1}$/Reserved	IN
138	PTF0/$\overline{PCC0VS2}$/Reserved	IN
137	MD0	IN
136	PTM4/PINT4/$\overline{AFE_RDET}$	OUT
135	Reserved/USB1d_SUSPEND0	OUT
134	$\overline{RTS2}$/USB1d_TXENL	OUT
133	PTE2/USB1_PWR_EN	OUT
132	PTE1/USB2_PWR_EN	OUT
131	CKE/PTK5	OUT
130	$\overline{RAS3}$/PTJ0	OUT
129	Reserved/PTJ1	OUT
128	Reserved/$\overline{CAS}$/PTJ2	OUT
127	Reserved/PTJ3	OUT
126	Reserved/PTJ4	OUT

Bit	Pin Name	I/O
125	Reserved/PTJ5	OUT
124	CL1/PTD5	OUT
123	DON/PTD7	OUT
122	M_DISP/PTE6	OUT
121	FLM/PTE3	OUT
120	PCC0RESET/DRAK0	OUT
119	$\overline{\text{PCC0DRV}}$/DACK0	OUT
118	ASEBRKAK/PTG5	OUT
117	AUDATA3/PTG3/PCC0BVD2	OUT
116	AUDATA2/PTG2/PCC0BVD1	OUT
115	AUDATA1/PTG1/$\overline{\text{PCC0CD2}}$	OUT
114	AUDATA0/PTG0/$\overline{\text{PCC0CD1}}$	OUT
113	PTF3/PINT11/Reserved	OUT
112	PTF2/$\overline{\text{PCCREG}}$/Reserved	OUT
111	PTF1/$\overline{\text{PCC0VS1}}$/Reserved	OUT
110	PTF0/$\overline{\text{PCC0VS2}}$/Reserved	OUT
109	PTM4/PINT4/$\overline{\text{AFE_RDET}}$	Control
108	Reserved/USB1d_SUSPEND0	Control
107	$\overline{\text{RTS2}}$/USB1d_TXENL	Control
106	PTE2/USB1_PWR_EN	Control
105	PTE1/USB2_PWR_EN	Control
104	CKE/PTK5	Control
103	$\overline{\text{RAS3}}$/PTJ0	Control
102	Reserved/PTJ1	Control
101	Reserved/$\overline{\text{CAS}}$/PTJ2	Control
100	Reserved/PTJ3	Control
99	Reserved/PTJ4	Control
98	Reserved/PTJ5	Control
97	CL1/PTD5	Control
96	DON/PTD7	Control
95	M_DISP/PTE6	Control
94	FLM/PTE3	Control
93	PCC0RESET/DRAK0	Control
92	$\overline{\text{PCC0DRV}}$/DACK0	Control
91	$\overline{\text{ASEBRKAK}}$/PTG5	Control

Bit	Pin Name	I/O
90	AUDATA3/PTG3/PCC0BVD2	Control
89	AUDATA2/PTG2/PCC0BVD1	Control
88	AUDATA1/PTG1/$\overline{\text{PCC0CD2}}$	Control
87	AUDATA0/PTG0/$\overline{\text{PCC0CD1}}$	Control
86	PTF3/PINT11/Reserved	Control
85	PTF2/$\overline{\text{PCCREG}}$/Reserved	Control
84	PTF1/$\overline{\text{PCC0VS1}}$/Reserved	Control
83	PTF0/$\overline{\text{PCC0VS2}}$/Reserved	Control
82	PTM3/LCD15/PINT10	IN
81	PTM2/LCD14/PINT9	IN
80	PTM1/LCD13/PINT8	IN
79	PTM0/LCD12	IN
78	STATUS0/PTJ6	IN
77	STATUS1/PTJ7	IN
76	CL2/PTH7	IN
75	SCK0/SCPT1	IN
74	SIOMCLK/SCPT3	IN
73	SCK_SIO/SCPT5	IN
72	SIOFSYNC/SCPT6	IN
71	RxD0/SCPT0	IN
70	RxD2/SCPT4	IN
69	LCD7/PTD3	IN
68	LCD6/PTD2	IN
67	LCD1/PTD1	IN
66	LCD0/PTD0	IN
65	$\overline{\text{DREQ0}}$/PTD4	IN
64	LCLK/UCLK/PTD6	IN
63	RxD_SIO/SCPT2	IN
62	$\overline{\text{CTS2}}$/IRQ5/SCPT7	IN
61	LCD11/PTC7/PINT3	IN
60	LCD10/PTC6/PINT2	IN
59	LCD9/PTC5/PINT1	IN
58	LCD8/PTC4/PINT0	IN
57	LCD5/PTC3	IN
56	LCD4/PTC2	IN

Bit	Pin Name	I/O
55	LCD3/PTC1	IN
54	LCD2/PTC0	IN
53	MD3	IN
52	MD4	IN
51	PTM3/LCD15/PINT10	OUT
50	PTM2/LCD14/PINT9	OUT
49	PTM1/LCD13/PINT8	OUT
48	PTM0/LCD12	OUT
47	STATUS0/PTJ6	OUT
46	STATUS1/PTJ7	OUT
45	CL2/PTH7	OUT
44	TxD0/SCPT0	OUT
43	SCK0/SCPT1	OUT
42	TxD_SIO/SCPT2	OUT
41	SIOMCLK/SCPT3	OUT
40	TxD2/SCPT4	OUT
39	SCK_SIO/SCPT5	OUT
38	SIOFSYNC/SCPT6	OUT
37	LCD11/PTC7/PINT3	OUT
36	LCD10/PTC6/PINT2	OUT
35	LCD9/PTC5/PINT1	OUT
34	LCD8/PTC4/PINT0	OUT
33	LCD7/PTD3	OUT
32	LCD6/PTD2	OUT
31	LCD5/PTC3	OUT
30	LCD4/PTC2	OUT
29	LCD3/PTC1	OUT
28	LCD2/PTC0	OUT
27	LCD1/PTD1	OUT
26	LCD0/PTD0	OUT
25	PTM3/LCD15/PINT10	Control
24	PTM2/LCD14/PINT9	Control
23	PTM1/LCD13/PINT8	Control
22	PTM0/LCD12	Control
21	STATUS0/PTJ6	Control
20	STATUS1/PTJ7	Control
19	CL2/PTH7	Control
18	TxD0/SCPT0	Control
17	SCK0/SCPT1	Control
16	TxD_SIO/SCPT2	Control
15	SIOMCLK/SCPT3	Control
14	TxD2/SCPT4	Control
13	SCK_SIO/SCPT5	Control
12	SIOFSYNC/SCPT6	Control
11	LCD11/PTC7/PINT3	Control
10	LCD10/PTC6/PINT2	Control
9	LCD9/PTC5/PINT1	Control
8	LCD8/PTC4/PINT0	Control
7	LCD7/PTD3	Control
6	LCD6/PTD2	Control
5	LCD5/PTC3	Control
4	LCD4/PTC2	Control
3	LCD3/PTC1	Control
2	LCD2/PTC0	Control
1	LCD1/PTD1	Control
0	LCD0/PTD0	Control
to TDO		

Note: Control is a signal that is active in its low level. Setting the control signal to low level drives the corresponding pin with the OUT value.

31.4 H-UDI Operations

31.4.1 TAP Controller

Figure 31.2 shows the internal states of TAP controller. State transitions basically conform with the JTAG standard.

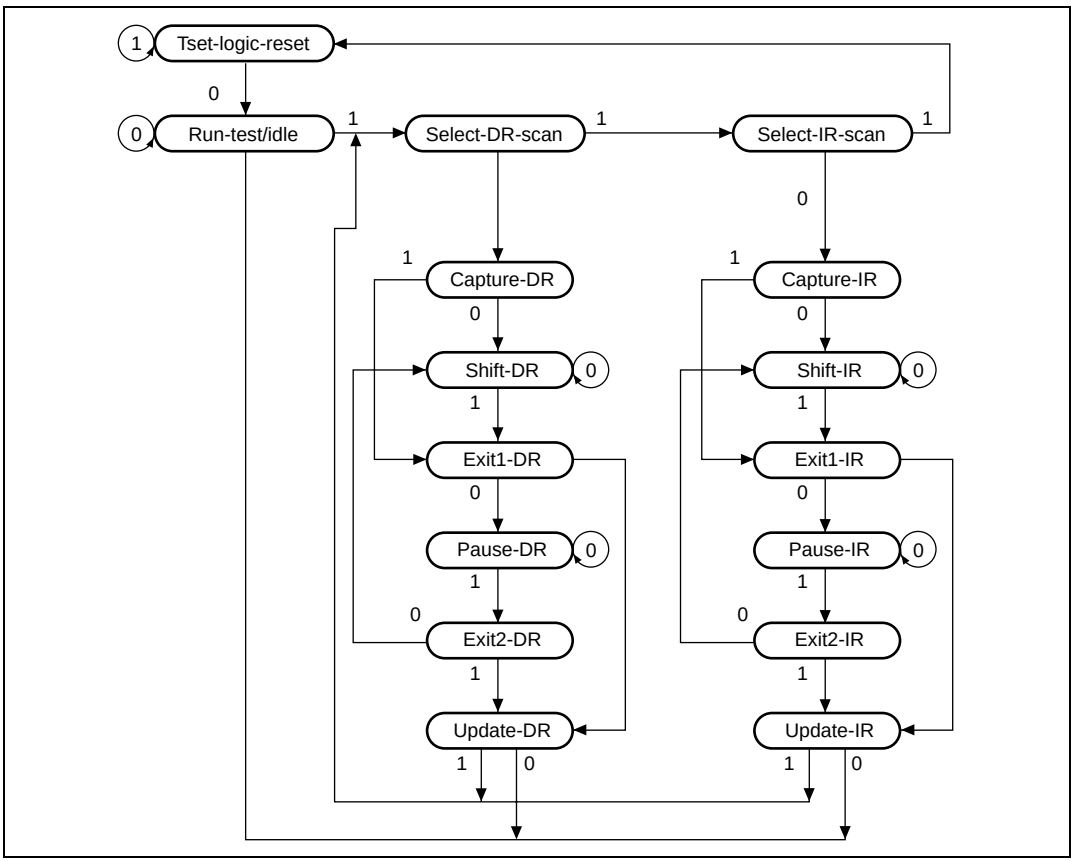


Figure 31.2 TAP Controller State Transitions

Note: The transition condition is the TMS value on the rising edge of TCK. The TDI value is sampled on the rising edge of TCK; shifting occurs on the falling edge of TCK. The TDO value changes on the TCK falling edge. The TDO is at high impedance, except with shift-DR (shift-SR) and shift-IR states. During the change to $\overline{\text{TRST}}$ = 0, there is a transition to test-logic-reset asynchronously with TCK.

31.4.2 Reset Configuration

Table 31.4 Reset Configuration

$\overline{\text{ASDMD0}}$*1	$\overline{\text{RESETP}}$	$\overline{\text{TRST}}$	Chip State
H	L	L	Normal reset and H-UDI reset
		H	Normal reset
	H	L	H-UDI reset only
		H	Normal operation
L	L	L	Reset hold*2
		H	ASE user mode*3: Normal reset ASE break mode*3: $\overline{\text{RESETP}}$ assertion masked
	H	L	H-UDI reset only
		H	Normal operation

Notes: 1. Performs main chip mode and ASE mode settings
$\overline{\text{ASEMD0}}$ = H, main chip mode
$\overline{\text{ASEMD0}}$ = L, ASE mode
When user system is used alone without using emulator or H-UDI, set $\overline{\text{ASEMD0}}$ to H.

2. During ASE mode, reset hold is enabled by setting $\overline{\text{RESETP}}$ and $\overline{\text{TRST}}$ pins at low level for a constant cycle. In this state, the CPU does not start up, even if $\overline{\text{RESETP}}$ is set to high level. When $\overline{\text{TRST}}$ is set to high level, H-UDI operation is enabled, but the CPU does not start up. The reset hold state is cancelled by the following:
   - Boot request from H-UDI (boot sequence)
   - Another $\overline{\text{RESETP}}$ assert (power-on reset)

3. There are two ASE modes, one for executing software in the emulator's firmware (ASE break mode) and one for executing user software (ASE user mode).

31.4.3 H-UDI Reset

An H-UDI reset is executed by setting an H-UDI reset assert command in SDIR. An H-UDI reset is of the same kind as a power-on reset. An H-UDI reset is released by inputting an H-UDI reset negate command.

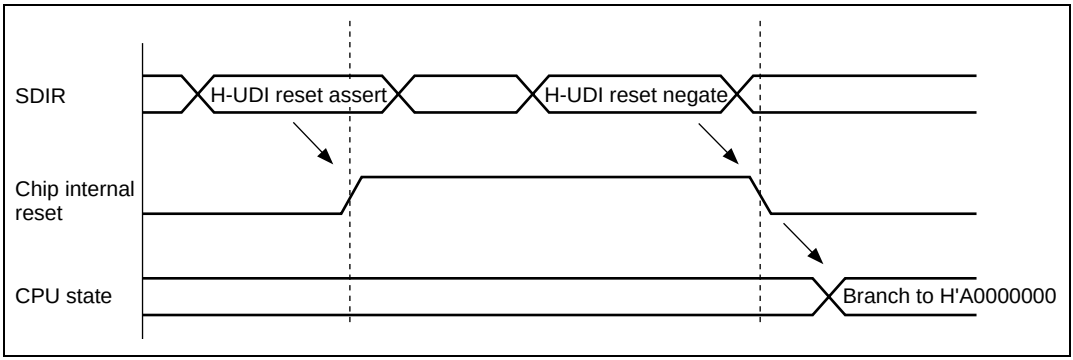


Figure 31.3 H-UDI Reset

31.4.4 H-UDI Interrupt

The H-UDI interrupt function generates an interrupt by setting a command from the H-UDI in the SDIR. An H-UDI interrupt is a general exception/interrupt operation, resulting in a branch to an address based on the VBR value plus offset, and return by the RTE instruction. This interrupt request has a fixed priority level of 15.

H-UDI interrupts are not accepted in sleep mode or standby mode.

31.4.5 Bypass

Setting the command from H-UDI to SDIR allows to set the H-UDI pins to the bypass mode that conforms with the JTAG standard.

31.4.6 Using H-UDI to Recover from Sleep Mode

It is possible to recover from sleep mode by setting a command (0001) from the H-UDI in SDIR.

31.5 Usage Notes

1. An H-UDI command other than an H-UDI interrupt, once set, will not be modified as long as another command is not re-issued from the H-UDI. An H-UDI interrupt command, however, will be changed to a bypass command once set.
2. Because chip operations are suspended in standby mode, H-UDI commands are not accepted. However, the TAP controller remains in operation at this time.
3. The H-UDI is used for emulator connection. Therefore, H-UDI functions cannot be used when using an emulator.

31.6 Advanced User Debugger (AUD)

The AUD is a function exclusively for use by an emulator. Refer to the User's Manual for the relevant emulator for details of the AUD.

Section 32 Electrical Characteristics

32.1 Absolute Maximum Ratings

Table 32.1 shows the absolute maximum ratings.

Table 32.1 Absolute Maximum Ratings

Item	Symbol	Rating	Unit
Power supply voltage (I/O)	VccQ	–0.3 to 4.2	V
Power supply voltage (internal)	Vcc Vcc – PLL1 Vcc – PLL2 Vcc – RTC	–0.3 to 2.5	V
Input voltage (except port L)	Vin	–0.3 to VccQ + 0.3	V
Input voltage (port L)	Vin	–0.3 to AVcc + 0.3	V
Analog power-supply voltage	AVcc	–0.3 to 4.6	V
USB power-supply voltage	AVcc_USB	–0.3 to 4.2	V
Analog input voltage	V_{AN}	–0.3 to AVcc + 0.3	V
Operating temperature	Topr	–20 to 75	°C
Storage temperature	Tstr	–55 to 125	°C

Caution: Operating the chip in excess of the absolute maximum rating may result in permanent damage.

- Order of turning on 1.9 V power (Vcc, Vcc-PLL1, Vcc-PLL2, Vcc-RTC) and 3.3 V power (VccQ, AVcc, AVcc_USB):
  1. First turn on the 3.3 V power, then turn on the 1.9 V power within 1 ms. This interval should be as short as possible.
  2. Until Voltage is applied to all power supplies, a low level is input at the $\overline{\text{RESETP}}$ pin and, a maximum of 4 CKIO clock cycles have been generated, internal circuits remain unsettled, and so pin states are also undefined. The system design must ensure that these undefined states do not cause erroneous system operation.
     Note that the $\overline{\text{RESETP}}$ pin will not accept low level input while the CA pin is low level.
  3. The state of pins MD5 to MD0 should be established after power supply startup and maintained until after the $\overline{\text{RESETP}}$ pin is negated. Read-in of pins MD5 to MD0 is always performed while the $\overline{\text{RESETP}}$ pin is asserted.

  Waveforms at power-on are shown in the following figure 32.1.

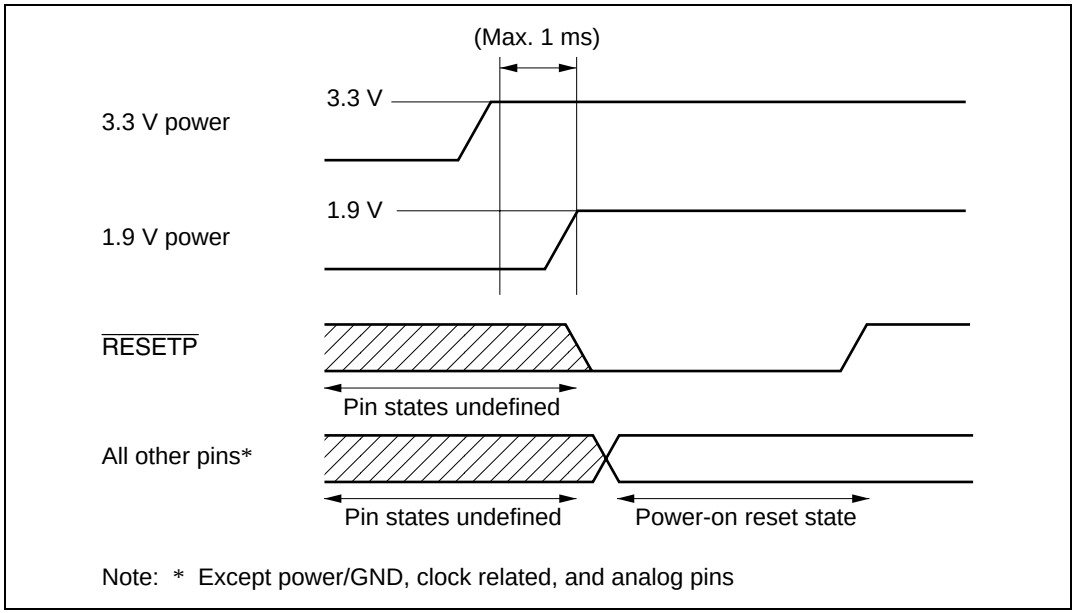


Figure 32.1 Power-On Sequence

- Power-off order
  1. In the reverse order of powering-on, first turn off the 1.9 V power, then turn off the 3.3 V power within 1 ms. This interval should be as short as possible.
  2. Pin states are undefined while only the 1.9 V power is off. The system design must ensure that these undefined states do not cause erroneous system operation.

32.2 DC Characteristics

Tables 32.2 and 32.3 list DC characteristics.

Table 32.2 DC Characteristics (1)

Condition: Ta = –20 to 75°C

Item		Symbol	Min	Typ	Max	Unit	Measurement Conditions
Power supply voltage		$V_{CC}Q$	3.0	—	3.6	V	160 MHz products
			2.6	—	3.6		100 MHz products
		V_{CC},	1.70	—	2.05		160 MHz products
		V_{CC}-PLL1, V_{CC}-PLL2, V_{CC}-RTC*1	1.60	—	2.05		100 MHz products
Analog (A/D, D/A) power-supply voltage		AV_{CC}*2	3.0	3.3	3.6	V	Same potential as VccQ when not used
Analog USB power-supply voltage		AV_{CC}_USB	3.0	3.3	3.6	V	Same potential as VccQ when not used
Analog (A/D, D/A) power-supply current	During A/D conversion	AI_{CC}	—	0.8	2	mA	
	During A/D and D/A conversion		—	2.4	6	mA	
	Idle		—	0.01	5.0	mA	Ta = 25°C
Current dissipation	Normal operation	I_{CC}*3	—	330	650	mA	Ta = 25°C, Vcc = 1.9 V, Iϕ = 160 MHz, X/Y memory on, cache on, and Renesas test program operating
			—	260	—		Ta = 25°C, Vcc = 1.9 V, Iϕ = 160 MHz, X/Y memory off, cache on, and Renesas test program operating

Item		Symbol	Min	Typ	Max	Unit	Measurement Conditions
Current dissipation	Normal operation	I_{CC}*3	—	250	450	mA	Ta = 25°C, Vcc = 1.9 V, Iϕ = 100 MHz, X/Y memory on, cache on, and Renesas test program operating
			—	190	—		Ta = 25°C, Vcc = 1.9 V, Iϕ = 100 MHz, X/Y memory off, cache on, and Renesas test program operating
		$I_{CC}Q$	—	20	—		$V_{CC}Q$ = 3.3 V Bϕ = 33 MHz
	In sleep mode*1	I_{SLEEP}	—	40	50	mA	Total Vcc + VccQ, Vcc = 1.9 V, VccQ = 3.3 V, Bϕ = 33 MHz, and no external bus cycles except refresh cycles
	In standby mode	I_{CC}	—	30	120	µA	Ta = 25°C (with RTC clock input)*4
		$I_{CC}Q$	—	10	30		Ta = 25°C (with RTC clock input)*4

Notes: 1. Regardless of whether the PLL or RTC is used, connect Vcc–PLL and Vcc–RTC to Vcc, and Vss–PLL and Vss–RTC to Vss.

2. AVcc conditions must be: VccQ – 0.3 V ≤ AVcc ≤ VccQ + 0.3 V. If the A/D and D/A converters are not used, do not leave the AVcc and AVss pins open. Connect AVcc to VccQ, and connect AVss to VssQ.
3. Current dissipation values are for V_{IH} min = VccQ – 0.5 V and V_{IL} max = 0.5 V with all output pins in the no-load state.
4. There is no stipulation regarding the power supply in standby mode when there is no RTC clock input.

Table 32.2 DC Characteristics (2)

Condition: Ta = –20 to 75°C

Item		Symbol	Min	Typ	Max	Unit	Measurement Conditions
Input high voltage	$\overline{\text{RESETP}}$, $\overline{\text{RESETM}}$, NMI, IRQ5 to IRQ0, MD5 to MD0, $\overline{\text{IRL3}}$ to $\overline{\text{IRL0}}$, PINT15 to PINT0, $\overline{\text{ASEMD0}}$, $\overline{\text{ADTRG}}$, $\overline{\text{TRST}}$, EXTAL, CKIO, CA	V_{IH}	$V_{CC}Q \times 0.9$	—	$V_{CC}Q + 0.3$	V	
	EXTAL2		—	—	—		Connect to Vcc when no crystal oscillator is connected
	Port L		2.0	—	$AV_{CC} + 0.3$		
	Other input pins		2.0	—	$V_{CC}Q + 0.3$		
Input low voltage	$\overline{\text{RESETP}}$, $\overline{\text{RESETM}}$, NMI	V_{IL}	–0.3	—	$V_{CC}Q \times 0.1$	V	
	$\overline{\text{BREQ}}$, IRQ5 to IRQ0, MD5 to MD0		–0.3	—	0.5		Standby mode
			–0.3	—	$V_{CC}Q \times 0.2$		Normal operation
	Port L		–0.3	—	$AV_{CC} \times 0.2$		
	Other input pin		–0.3	—	$V_{CC}Q \times 0.2$		
Input leak current	All input pins	|Iin|	—	—	1.0	µA	Vin = 0.5 to $V_{CC}Q$ – 0.5 V
Three-state leak current	I/O, all output pins (off condition)	|Isti|	—	—	1.0	µA	Vin = 0.5 to $V_{CC}Q$ – 0.5 V
Output high voltage	All output pins	V_{OH}	2.4	—	—	V	$V_{CC}Q$ = 3.0 V, I_{OH} = –200 µA
			2.0	—	—	V	$V_{CC}Q$ = 3.0 V, I_{OH} = –2mA
Output low voltage	All output pins	V_{OL}	—	—	0.55	V	$V_{CC}Q$ = 3.6 V, I_{OL} = 1.6 mA

Item		Symbol	Min	Typ	Max	Unit	Measurement Conditions
Pull-up resistance	Port pin	Ppull	30	60	120	kΩ	
Pin capacity	Pins other than analog pins	C	—	—	10	pF	
	Analog pins*	C_{AN}	—	—	20	pF	

Note: * There are four analog pins, USB1_P, USB2_P, USB1_M, and USB2_M.

Table 32.3 Permitted Output Current Values

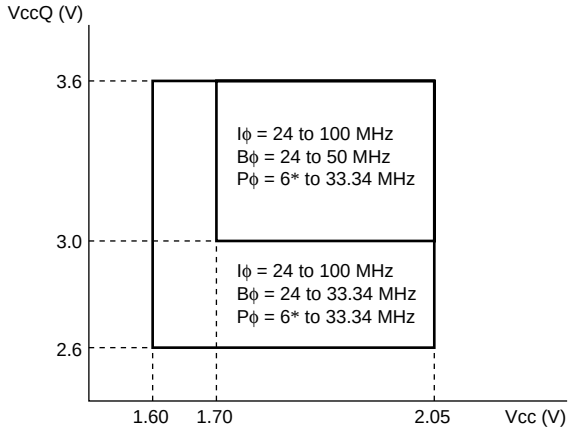
Conditions: VccQ = 2.6 to 3.6 V, Vcc = 1.60 to 2.05 V, AVcc = 3.0 to 3.6 V, Ta = –20 to 75°C

Item	Symbol	Min	Typ	Max	Unit
Output low-level permissible current (per pin)	I_{OL}	—	—	2.0	mA
Output low-level permissible current (total)	$\sum I_{OL}$	—	—	120	mA
Output high-level permissible current (per pin)	$-I_{OH}$	—	—	2.0	mA
Output high-level permissible current (total)	$\sum (-I_{OH})$	—	—	40	mA

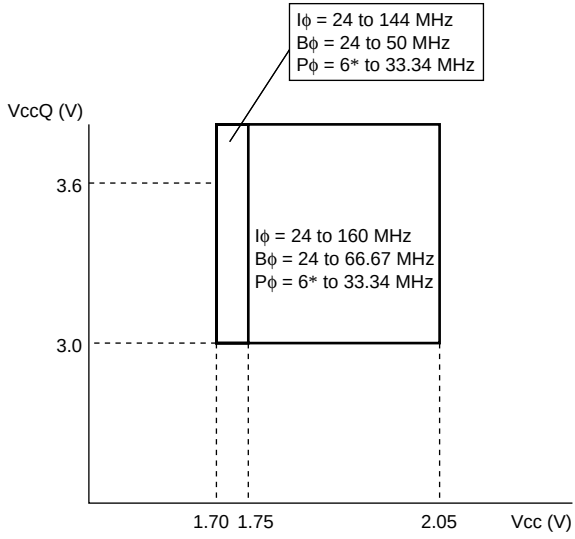
Caution: To ensure LSI reliability, do not exceed the value for output current given in table 32.3.

32.3 AC Characteristics

In general, inputting for this LSI should be clock synchronous. Keep the setup and hold times for each input signal unless otherwise specified. Regarding the power supply and frequency specifications of the individual products, refer to figure 32.2, tables 32.2 and 32.4.



(1) 100 MHz products



(2) 160 MHz products

Note: * When using the USB function controller, select a peripheral clock (Pϕ) setting of 13 MHz or above.

Figure 32.2 Power Supply Voltage and Operating Frequency

Table 32.4 Maximum Operating Frequencies (1)

Item	Symbol	Min	Max	Unit	Power Supply Voltage Conditions	Reference	Products
CPU, cache, TLB (Iϕ)	f	24	100	MHz	V_{CC} = 1.60 to 2.05 V $V_{CC}Q$ = 2.6 to 3.6 V	—	100 MHz products
External bus (Bϕ) or CKIO I/O frequency		24	33.34		V_{CC} = 1.60 to 2.05 V $V_{CC}Q$ = 2.6 to 3.6 V	Table 32.5	
		24	50		V_{CC} = 1.70 to 2.05 V $V_{CC}Q$= 3.0 to 3.6 V	Table 32.6	
Peripheral modules (Pϕ)		6	33.34		V_{CC} = 1.60 to 2.05 V $V_{CC}Q$ = 2.6 to 3.6 V	—	

Table 32.4 Maximum Operating Frequencies (2)

Item	Symbol	Min	Max	Unit	Power Supply Voltage Conditions	Reference	Products
CPU, cache, TLB (Iϕ)	f	24	144	MHz	V_{CC} = 1.70 to 2.05 V $V_{CC}Q$= 3.0 to 3.6 V	—	160 MHz products
		24	160		V_{CC} = 1.75 to 2.05 V $V_{CC}Q$= 3.0 to 3.6 V	—	
External bus (Bϕ) or CKIO I/O frequency		24	50		V_{CC} = 1.70 to 2.05 V $V_{CC}Q$= 3.0 to 3.6 V	Table 32.7	
		24	66.67		V_{CC} = 1.75 to 2.05 V $V_{CC}Q$= 3.0 to 3.6 V	Table 32.8	
Peripheral modules (Pϕ)		6	33.34		V_{CC} = 1.70 to 2.05 V $V_{CC}Q$= 3.0 to 3.6 V	—	

32.3.1 Clock Timing

Table 32.5 Clock Timing (1)

Conditions: VccQ = 2.6 to 3.6 V, Vcc = 1.60 to 2.05 V, AVcc = 3.3 ± 0.3 V, Ta = –20 to 75°C, 100 MHz products

Item	Symbol	Min	Max	Unit	Figure
EXTAL clock input frequency	f_{EX}	6	33.34	MHz	32.3
EXTAL clock input cycle time	t_{EXcyc}	30	166.7	ns	
EXTAL clock input low pulse width	t_{EXL}	9	—	ns	
EXTAL clock input high pulse width	t_{EXH}	9	—	ns	
EXTAL clock input rise time	t_{EXR}	—	6	ns	
EXTAL clock input fall time	t_{EXF}	—	6	ns	
CKIO clock input frequency	f_{CKI}	24	33.34	MHz	32.4
CKIO clock input cycle time	t_{CKIcyc}	30	41.7	ns	
CKIO clock input low pulse width	t_{CKIL}	9	—	ns	
CKIO clock input high pulse width	t_{CKIH}	9	—	ns	
CKIO clock input rise time	t_{CKIR}	—	6	ns	
CKIO clock input fall time	t_{CKIF}	—	6	ns	
CKIO clock output frequency	f_{OP}	24	33.34	MHz	32.5
CKIO clock output cycle time	t_{cyc}	30	41.7	ns	
CKIO clock output low pulse width	t_{CKOL}	8	—	ns	
CKIO clock output high pulse width	t_{CKOH}	8	—	ns	
CKIO clock output rise time	t_{CKOR}	—	6	ns	
CKIO clock output fall time	t_{CKOF}	—	6	ns	
CKIO2 clock output delay time	t_{CK2D}	—	2.5	ns	
CKIO2 clock output rise time	t_{CK2OR}	—	7	ns	
CKIO2 clock output fall time	t_{CK2OF}	—	7	ns	
Power-on oscillation settling time	t_{OSC1}	10	—	ms	32.6
$\overline{\text{RESETP}}$ setup time (At power on and cancellation of standby mode)	t_{RESPS}	20	—	ns	32.6, 32.7
$\overline{\text{RESETM}}$ setup time (At cancellation of standby mode)	t_{RESMS}	0	—	ns	
$\overline{\text{RESETP}}$ assert time (At power on and cancellation of standby mode)	t_{RESPW}	20	—	t_{cyc}	
$\overline{\text{RESETM}}$ assert time (At cancellation of standby mode)	t_{RESMW}	20	—	t_{cyc}	
Standby return oscillation settling time 1	t_{OSC2}	10	—	ms	32.7
Standby return oscillation settling time 2	t_{OSC3}	10	—	ms	32.8
Standby return oscillation settling time 3	t_{OSC4}	11	—	ms	32.9
PLL synchronization settling time 1 (At cancellation of standby mode)	t_{PLL1}	100	—	µs	32.10, 32.11
PLL synchronization settling time 2 (At multiplier change)	t_{PLL2}	100	—	µs	32.12
IRQ/IRL interrupt determination time (RTC used and standby mode)	t_{IRQSTB}	100	—	µs	32.11

Table 32.6 Clock Timing (2)

Conditions: VccQ = 3.0 to 3.6 V, Vcc = 1.70 to 2.05 V, AVcc = 3.3 ± 0.3 V, Ta = –20 to 75°C, 100 MHz products

Item	Symbol	Min	Max	Unit	Figure
EXTAL clock input frequency	f_{EX}	6	50	MHz	32.3
EXTAL clock input cycle time	t_{EXcyc}	20	166.7	ns	
EXTAL clock input low pulse width	t_{EXL}	4	—	ns	
EXTAL clock input high pulse width	t_{EXH}	4	—	ns	
EXTAL clock input rise time	t_{EXR}	—	6	ns	
EXTAL clock input fall time	t_{EXF}	—	6	ns	
CKIO clock input frequency	f_{CKI}	24	50	MHz	32.4
CKIO clock input cycle time	t_{CKIcyc}	20	41.7	ns	
CKIO clock input low pulse width	t_{CKIL}	4	—	ns	
CKIO clock input high pulse width	t_{CKIH}	4	—	ns	
CKIO clock input rise time	t_{CKIR}	—	6	ns	
CKIO clock input fall time	t_{CKIF}	—	6	ns	
CKIO clock output frequency	f_{OP}	24	50	MHz	32.5
CKIO clock output cycle time	t_{cyc}	20	41.7	ns	
CKIO clock output low pulse width	t_{CKOL}	3	—	ns	
CKIO clock output high pulse width	t_{CKOH}	3	—	ns	
CKIO clock output rise time	t_{CKOR}	—	5	ns	
CKIO clock output fall time	t_{CKOF}	—	5	ns	
CKIO2 clock output delay time	t_{CK2D}	—	2.5	ns	
CKIO2 clock output rise time	t_{CK2OR}	—	7	ns	
CKIO2 clock output fall time	t_{CK2OF}	—	7	ns	
Power-on oscillation settling time	t_{OSC1}	10	—	ms	32.6
$\overline{\text{RESETP}}$ setup time (At power on and cancellation of standby mode)	t_{RESPS}	20	—	ns	32.6, 32.7
$\overline{\text{RESETM}}$ setup time (At cancellation of standby mode)	t_{RESMS}	0	—	ns	
$\overline{\text{RESETP}}$ assert time (At power on and cancellation of standby mode)	t_{RESPW}	20	—	t_{cyc}	
$\overline{\text{RESETM}}$ assert time (At cancellation of standby mode)	t_{RESMW}	20	—	t_{cyc}	
Standby return oscillation settling time 1	t_{OSC2}	10	—	ms	32.7
Standby return oscillation settling time 2	t_{OSC3}	10	—	ms	32.8
Standby return oscillation settling time 3	t_{OSC4}	11	—	ms	32.9
PLL synchronization settling time 1 (At cancellation of standby mode)	t_{PLL1}	100	—	µs	32.10, 32.11
PLL synchronization settling time 2 (At multiplier change)	t_{PLL2}	100	—	µs	32.12
IRQ/IRL interrupt determination time (RTC used and standby mode)	t_{IRQSTB}	100	—	µs	32.11

Table 32.7 Clock Timing (3)

Conditions: VccQ = 3.0 to 3.6 V, Vcc = 1.70 to 2.05 V, AVcc = 3.3 ± 0.3 V, Ta = –20 to 75°C, 160 MHz products

Item	Symbol	Min	Max	Unit	Figure
EXTAL clock input frequency	f_{EX}	6	50	MHz	32.3
EXTAL clock input cycle time	t_{EXcyc}	20	166.7	ns	
EXTAL clock input low pulse width	t_{EXL}	4	—	ns	
EXTAL clock input high pulse width	t_{EXH}	4	—	ns	
EXTAL clock input rise time	t_{EXR}	—	6	ns	
EXTAL clock input fall time	t_{EXF}	—	6	ns	
CKIO clock input frequency	f_{CKI}	24	50	MHz	32.4
CKIO clock input cycle time	t_{CKIcyc}	20	41.7	ns	
CKIO clock input low pulse width	t_{CKIL}	4	—	ns	
CKIO clock input high pulse width	t_{CKIH}	4	—	ns	
CKIO clock input rise time	t_{CKIR}	—	6	ns	
CKIO clock input fall time	t_{CKIF}	—	6	ns	
CKIO clock output frequency	f_{OP}	24	50	MHz	32.5
CKIO clock output cycle time	t_{cyc}	20	41.7	ns	
CKIO clock output low pulse width	t_{CKOL}	3	—	ns	
CKIO clock output high pulse width	t_{CKOH}	3	—	ns	
CKIO clock output rise time	t_{CKOR}	—	5	ns	
CKIO clock output fall time	t_{CKOF}	—	5	ns	
CKIO2 clock output delay time	t_{CK2D}	—	2.5	ns	
CKIO2 clock output rise time	t_{CK2OR}	—	7	ns	
CKIO2 clock output fall time	t_{CK2OF}	—	7	ns	
Power-on oscillation settling time	t_{OSC1}	10	—	ms	32.6
$\overline{\text{RESETP}}$ setup time (At power on and cancellation of standby mode)	t_{RESPS}	20	—	ns	32.6, 32.7
$\overline{\text{RESETM}}$ setup time (At cancellation of standby mode)	t_{RESMS}	0	—	ns	
$\overline{\text{RESETP}}$ assert time (At power on and cancellation of standby mode)	t_{RESPW}	20	—	t_{cyc}	
$\overline{\text{RESETM}}$ assert time (At cancellation of standby mode)	t_{RESMW}	20	—	t_{cyc}	
Standby return oscillation settling time 1	t_{OSC2}	10	—	ms	32.7
Standby return oscillation settling time 2	t_{OSC3}	10	—	ms	32.8
Standby return oscillation settling time 3	t_{OSC4}	11	—	ms	32.9
PLL synchronization settling time 1 (At cancellation of standby mode)	t_{PLL1}	100	—	µs	32.10, 32.11
PLL synchronization settling time 2 (At multiplier change)	t_{PLL2}	100	—	µs	32.12
IRQ/IRL interrupt determination time (RTC used and standby mode)	t_{IRQSTB}	100	—	µs	32.11

Table 32.8 Clock Timing (4)

Conditions: VccQ = 3.0 to 3.6 V, Vcc = 1.75 to 2.05 V, AVcc = 3.3 ± 0.3 V, Ta = –20 to 75°C, 160 MHz products

Item	Symbol	Min	Max	Unit	Figure
EXTAL clock input frequency	f_{EX}	6	66.67	MHz	32.3
EXTAL clock input cycle time	t_{EXcyc}	15	166.7	ns	
EXTAL clock input low pulse width	t_{EXL}	1.5	—	ns	
EXTAL clock input high pulse width	t_{EXH}	1.5	—	ns	
EXTAL clock input rise time	t_{EXR}	—	6	ns	
EXTAL clock input fall time	t_{EXF}	—	6	ns	
CKIO clock input frequency	f_{CKI}	24	66.67	MHz	32.4
CKIO clock input cycle time	t_{CKIcyc}	15	41.7	ns	
CKIO clock input low pulse width	t_{CKIL}	1.5	—	ns	
CKIO clock input high pulse width	t_{CKIH}	1.5	—	ns	
CKIO clock input rise time	t_{CKIR}	—	6	ns	
CKIO clock input fall time	t_{CKIF}	—	6	ns	
CKIO clock output frequency	f_{OP}	24	66.67	MHz	32.5
CKIO clock output cycle time	t_{cyc}	15	41.7	ns	
CKIO clock output low pulse width	t_{CKOL}	3	—	ns	
CKIO clock output high pulse width	t_{CKOH}	3	—	ns	
CKIO clock output rise time	t_{CKOR}	—	5	ns	
CKIO clock output fall time	t_{CKOF}	—	5	ns	
CKIO2 clock output delay time	t_{CK2D}	—	2.5	ns	
CKIO2 clock output rise time	t_{CK2OR}	—	7	ns	
CKIO2 clock output fall time	t_{CK2OF}	—	7	ns	
Power-on oscillation settling time	t_{OSC1}	10	—	ms	32.6
$\overline{\text{RESETP}}$ setup time (At power on and cancellation of standby mode)	t_{RESPS}	20	—	ns	32.6, 32.7
$\overline{\text{RESETM}}$ setup time (At cancellation of standby mode)	t_{RESMS}	0	—	ns	
$\overline{\text{RESETP}}$ assert time (At power on and cancellation of standby mode)	t_{RESPW}	20	—	t_{cyc}	
$\overline{\text{RESETM}}$ assert time (At cancellation of standby mode)	t_{RESMW}	20	—	t_{cyc}	
Standby return oscillation settling time 1	t_{OSC2}	10	—	ms	32.7
Standby return oscillation settling time 2	t_{OSC3}	10	—	ms	32.8
Standby return oscillation settling time 3	t_{OSC4}	11	—	ms	32.9
PLL synchronization settling time 1 (At cancellation of standby mode)	t_{PLL1}	100	—	µs	32.10, 32.11
PLL synchronization settling time 2 (At multiplier change)	t_{PLL2}	100	—	µs	32.12
IRQ/IRL interrupt determination time (RTC used and standby mode)	t_{IRQSTB}	100	—	µs	32.11

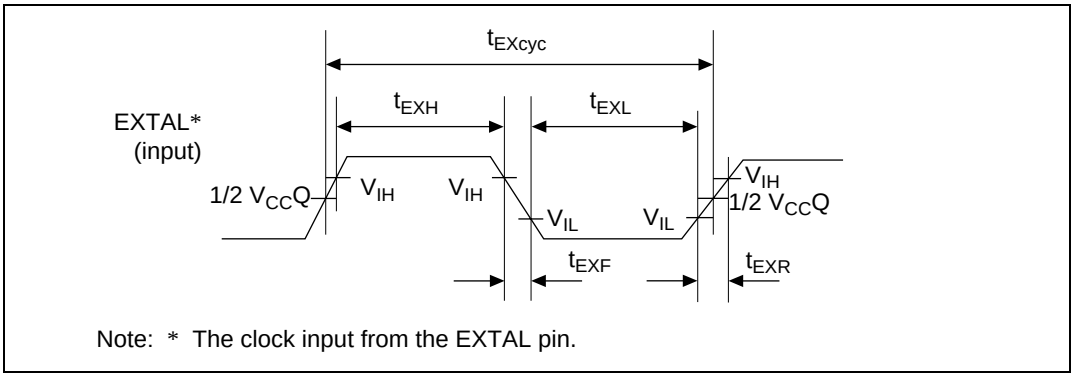


Figure 32.3 EXTAL Clock Input Timing

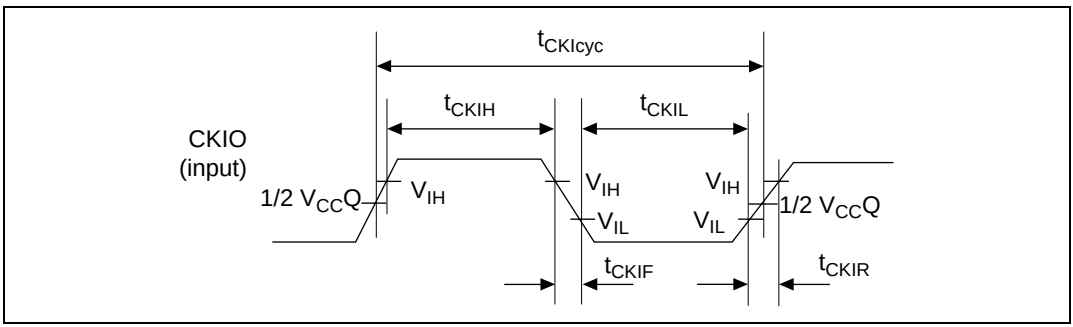


Figure 32.4 CKIO Clock Input Timing

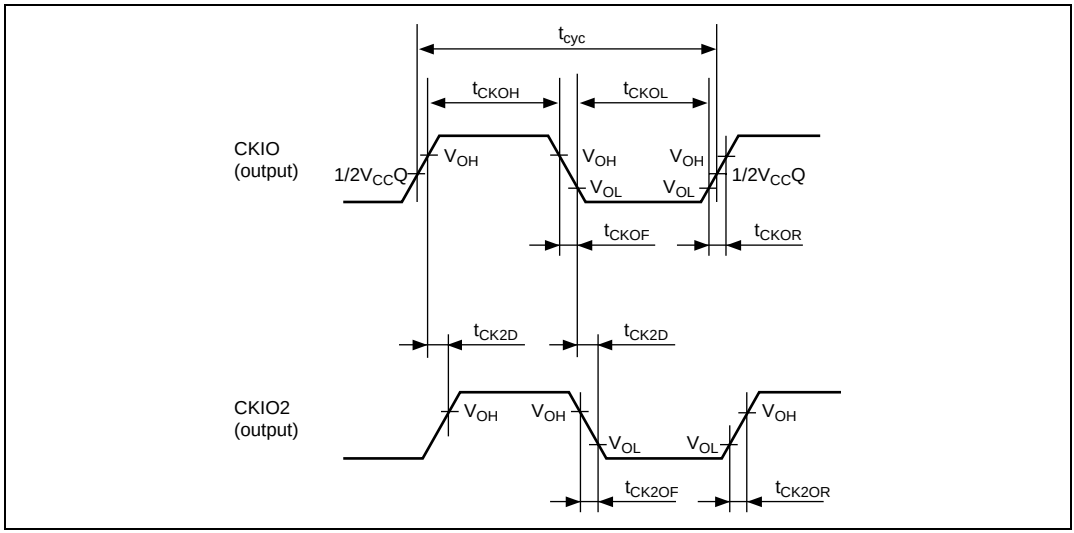


Figure 32.5 CKIO Clock Output Timing

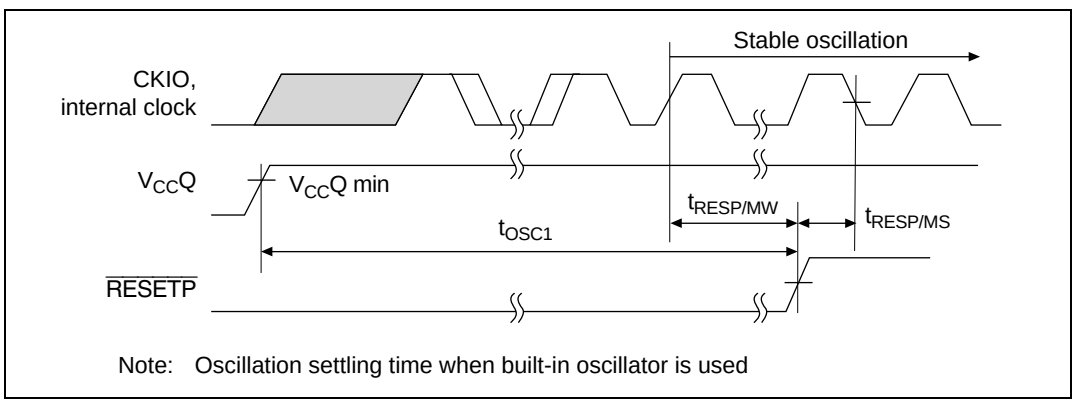


Figure 32.6 Power-on Oscillation Settling Time

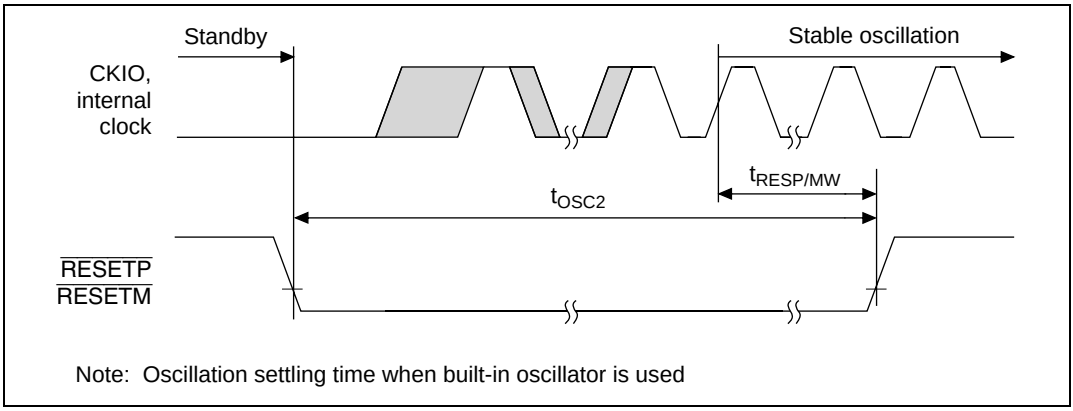


Figure 32.7 Oscillation Settling Time at Standby Return (Return by Reset)

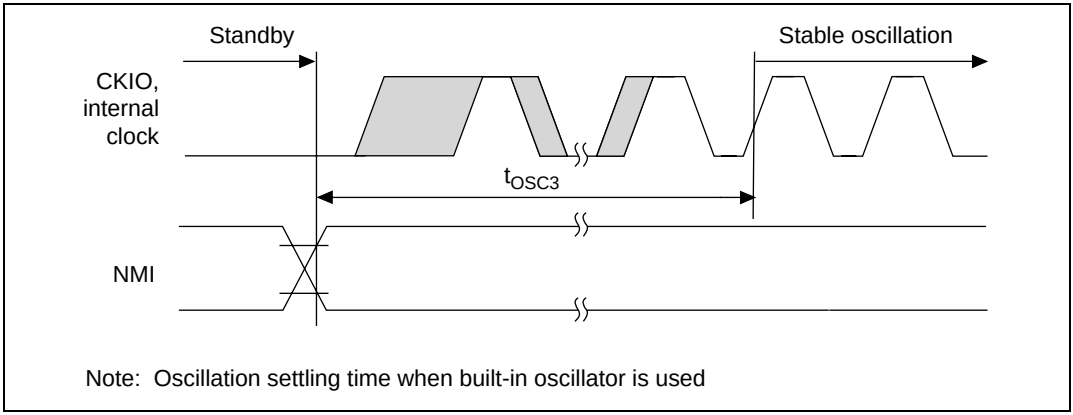


Figure 32.8 Oscillation Settling Time at Standby Return (Return by NMI)

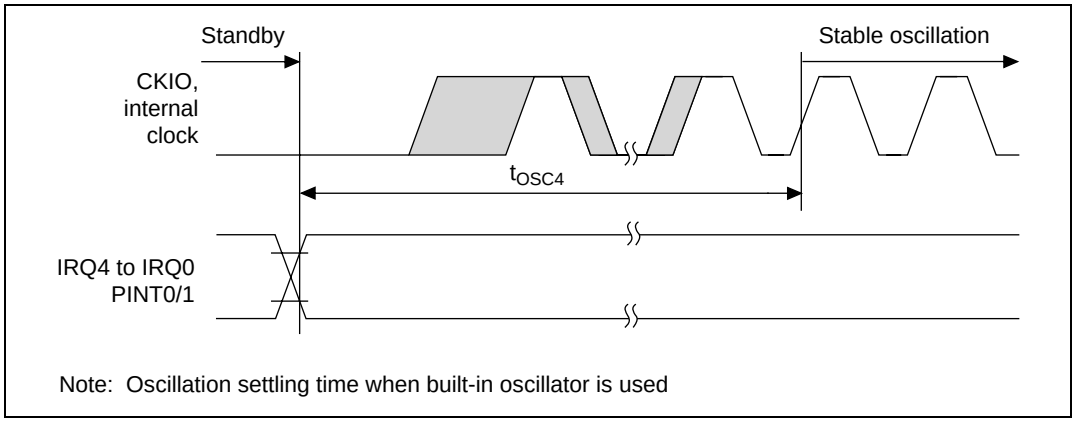


Figure 32.9 Oscillation Settling Time at Standby Return (Return by IRQ4 to IRQ0)

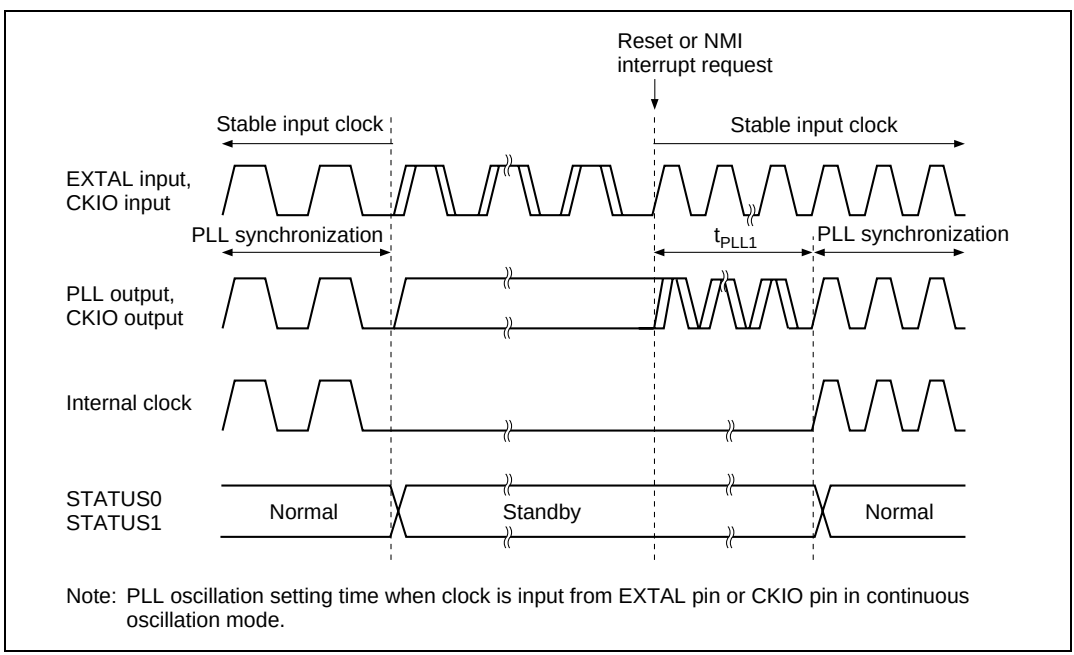
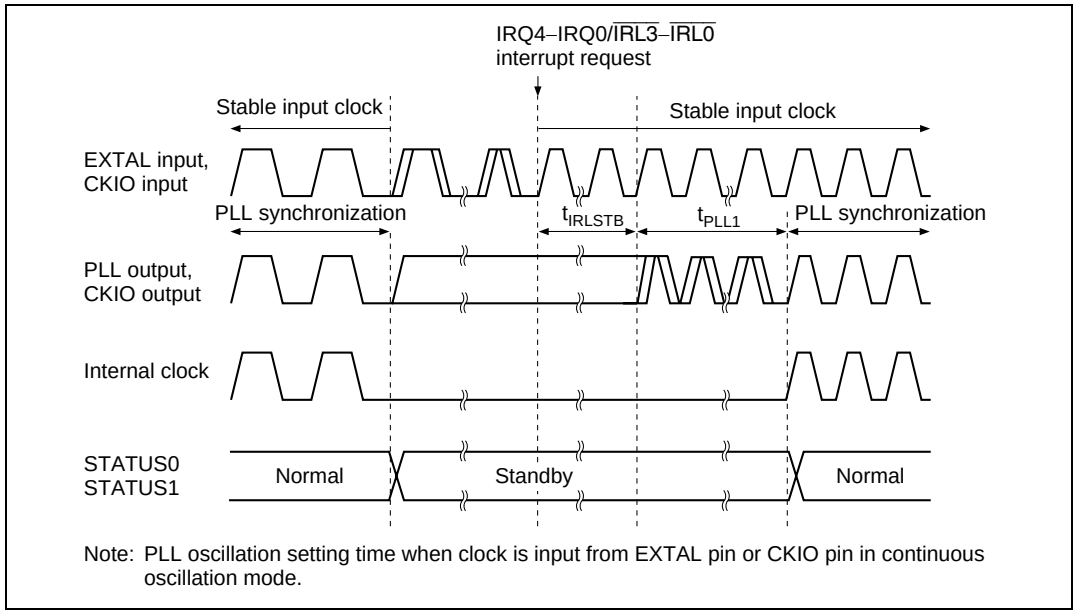
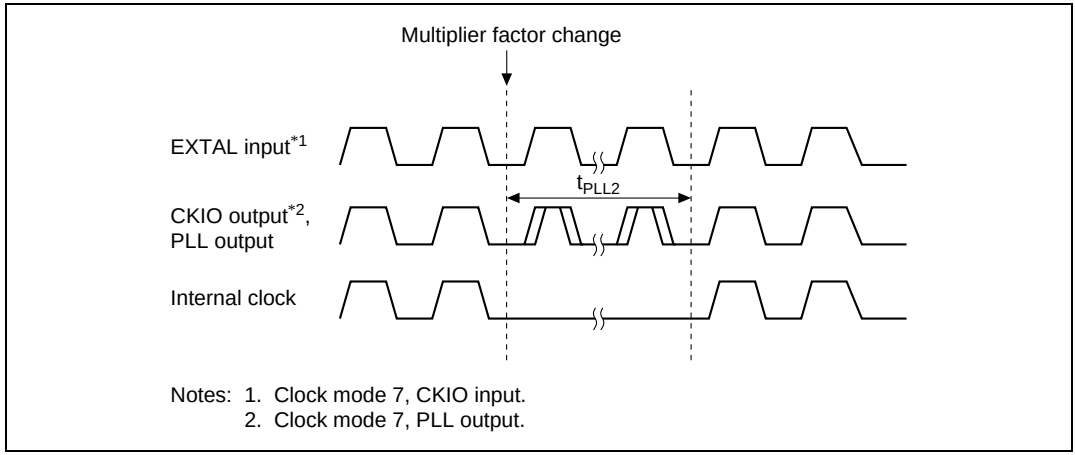


Figure 32.10 PLL Synchronization Settling Time by Reset or NMI Interrupt



Note: PLL oscillation setting time when clock is input from EXTAL pin or CKIO pin in continuous oscillation mode.

Figure 32.11 PLL Synchronization Settling Time by IRQ/IRL and PINT0/1 Interrupt



Notes: 1. Clock mode 7, CKIO input.
2. Clock mode 7, PLL output.

Figure 32.12 PLL Sync Stabilization Time at Frequency Multiplier Factor Change

32.3.2 Control Signal Timing

Table 32.9 Control Signal Timing

Conditions: VccQ = 2.6 to 3.6 V, Vcc = 1.60 to 2.05 V, AVcc = 3.3 ± 0.3 V, Ta = –20 to 75°C

		Min	Max		Figure
$\overline{\text{RESETP}}$ pulse width	t_{RESPW}	20*2	—	tcyc	32.13
$\overline{\text{RESETP}}$ setup time*1	t_{RESPS}	23	—	ns	32.14
$\overline{\text{RESETP}}$ hold time	t_{RESPH}	2	—	ns	
$\overline{\text{RESETM}}$ pulse width	t_{RESMW}	12*3	—	tcyc	
$\overline{\text{RESETM}}$ setup time	t_{RESMS}	3	—	ns	
$\overline{\text{RESETM}}$ hold time	t_{RESMH}	34	—	ns	
$\overline{\text{BREQ}}$ setup time	t_{BREQS}	10	—	ns	32.15
$\overline{\text{BREQ}}$ hold time	t_{BREQH}	3	—	ns	
NMI setup time *1	t_{NMIS}	10	—	ns	32.14
NMI hold time	t_{NMIH}	4	—	ns	
IRQ5–IRQ0 setup time *1	t_{IRQS}	10	—	ns	
IRQ5–IRQ0 hold time	t_{IRQH}	4	—	ns	
$\overline{\text{BACK}}$ delay time	t_{BACKD}	—	10	ns	32.15
STATUS1, STATUS0 delay time	t_{STD}	—	16	ns	32.16
Bus tri-state delay time 1	t_{BOFF1}	0	15	ns	
Bus tri-state delay time 2	t_{BOFF2}	0	15	ns	
Bus buffer-on time 1	t_{BON1}	0	15	ns	
Bus buffer-on time 2	t_{BON2}	0	15	ns	

Notes: 1. $\overline{\text{RESETP}}$, NMI and IRQ5 to IRQ0 are asynchronous. Changes are detected at the clock fall when the setup shown is used. When the setup cannot be used, detection can be delayed until the next clock falls. When using as IRL, please observe the setup time.

2. In the standby mode, $t_{RESPW} = t_{OSC2}$ (10 ms). In the sleep mode, $t_{RESPW} = t_{PLL1}$ (100 µs). When the clock multiplication ratio is changed, $t_{RESPW} = t_{PLL1}$ (100 µs).
3. In the standby mode, $t_{RESMW} = t_{OSC2}$ (10 ms). In the sleep mode, $\overline{\text{RESETM}}$ must be kept low until STATUS (0-1) changes to reset (HH). When the clock multiplication ratio is changed, $\overline{\text{RESETM}}$ must be kept low until STATUS (0-1) changes to reset (HH).

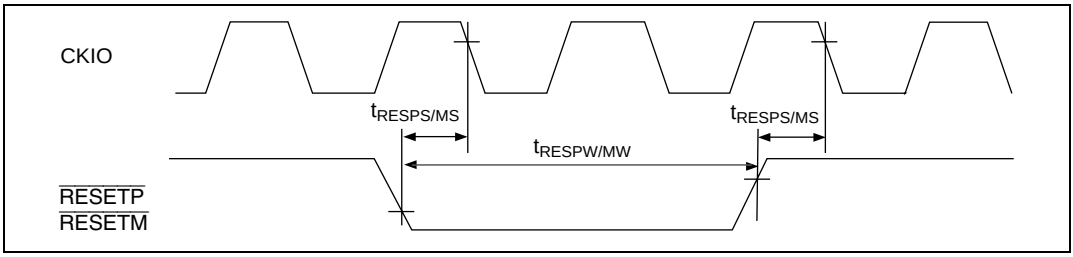


Figure 32.13 Reset Input Timing

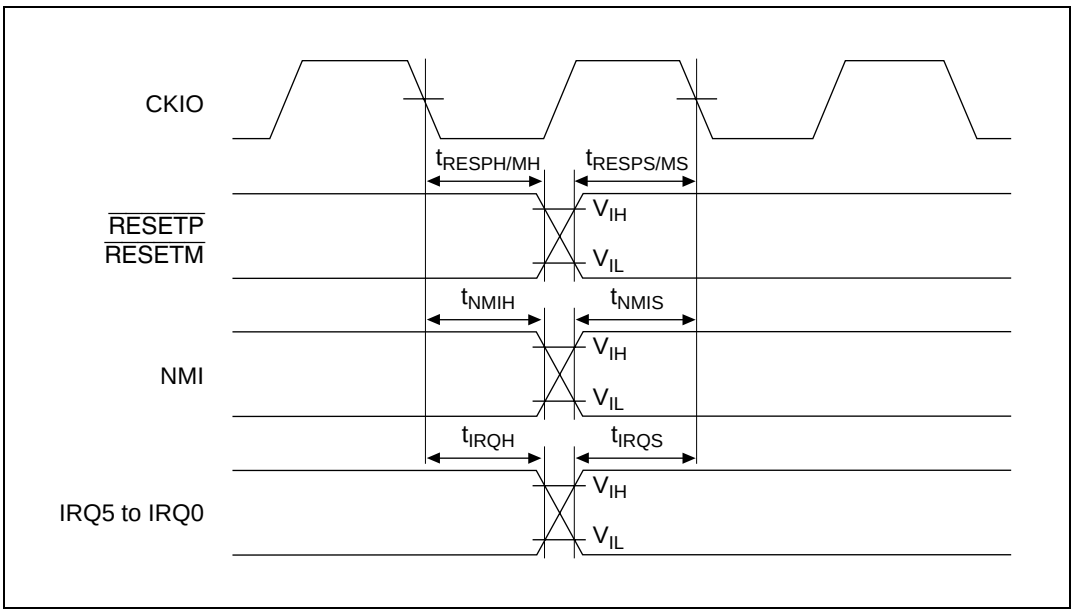


Figure 32.14 Interrupt signal Input Timing

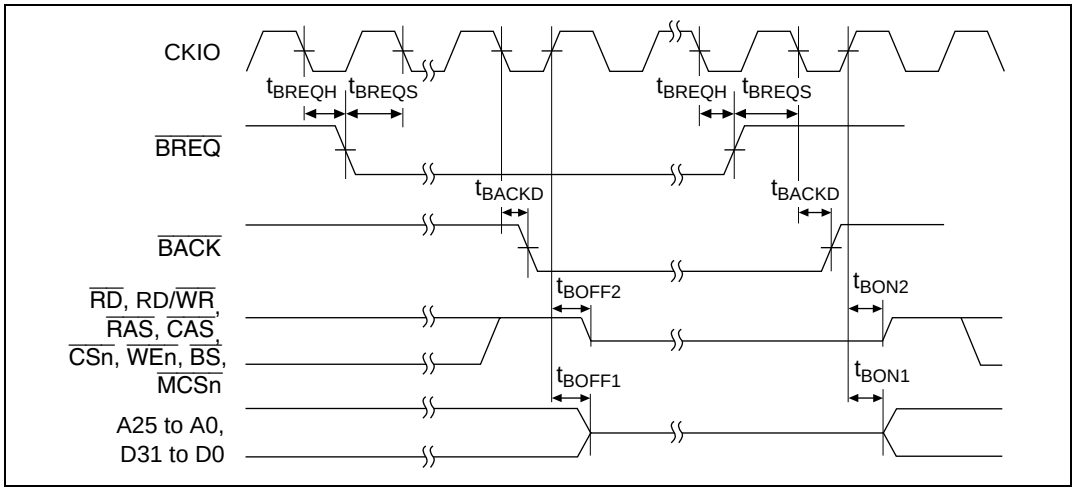


Figure 32.15 Bus Release Timing

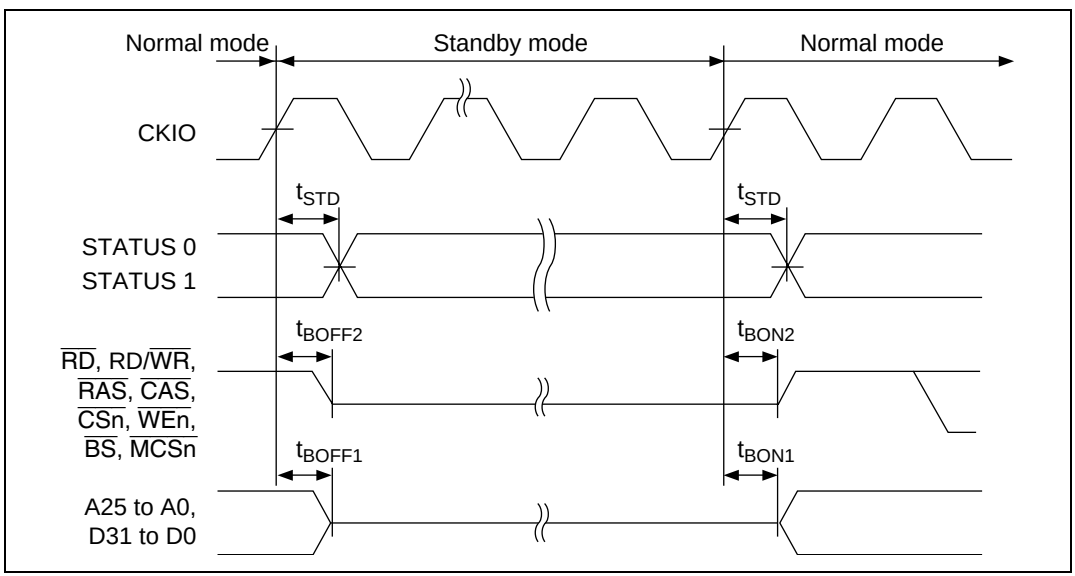


Figure 32.16 Pin Drive Timing at Standby

32.3.3 AC Bus Timing

Table 32.10 Bus Timing

Conditions: Clock Modes 0/1/2/7, VccQ = 2.6 to 3.6 V, Vcc = 1.60 to 2.05 V, AVcc = 3.3 ± 0.3 V, Ta = –20 to 75°C

Item	Symbol	Min	Max	Unit	Conditions	Figure
Address delay time	t_{AD}	1.5	13	ns	Vcc = 1.70 to 2.05 V VccQ = 3.0 to 3.6 V	32.17 to 32.30, 32.33 to 32.40
		1.5	16		Other than the above	
Address setup time	t_{AS}	0	—	ns		32.17 to 32.19
Address hold time	t_{AH}*1	7	—	ns		32.17 to 32.22
$\overline{BS}$ delay time	t_{BSD}	—	12	ns		32.17 to 32.30, 32.34 to 32.40
$\overline{CS}$ delay time 1	t_{CSD1}	1.5	12	ns		32.17 to 32.40
$\overline{CS}$ delay time 2	t_{CSD2}	1	12	ns		32.17 to 32.22
Read/write delay time	t_{RWD}	1.5	10	ns		32.17 to 32.40
Read/write hold time	t_{RWH}	0	—	ns		32.17 to 32.22
Read strobe delay time	t_{RSD}	—	10	ns		32.17 to 32.22, 32.34 to 32.37
Read data setup time 1	t_{RDS1}	6	—	ns		32.17 to 32.22, 32.34 to 32.40
Read data setup time 2	t_{RDS2}	7	—	ns		32.23 to 32.26
Read data hold time 1	t_{RDH1}*2	0	—	ns		32.17 to 32.22, 32.34 to 32.40
Read data hold time 2	t_{RDH2}	2	—	ns		32.23 to 32.26
Write enable delay time	t_{WED}	1	10	ns		32.17 to 32.19, 32.34, 32.35
Write data delay time 1	t_{WDD1}	—	14	ns		32.17 to 32.19, 32.34, 32.35, 32.38 to 32.40
Write data delay time 2	t_{WDD2}	—	13	ns		32.27 to 32.30

Item	Symbol	Min	Max	Unit	Conditions	Figure
Write data hold time 1	t_{WDH1}	1.5	—	ns		32.17 to 32.19, 32.34, 32.35, 32.38 to 32.40
Write data hold time 2	t_{WDH2}	1.5	—	ns		32.27 to 32.30
Write data hold time 3	t_{WDH3}	2	—	ns		32.17 to 32.19
Write data hold time 4	t_{WDH4}	2	—	ns		32.34, 32.35, 32.38 to 32.40
$\overline{WAIT}$ setup time	t_{WTS}	5	—	ns	Vcc = 1.70 to 2.05 V VccQ = 3.0 to 3.6 V	32.18 to 32.22, 32.35, 32.37, 32.39, 32.40
		6	—		Other than the above	
$\overline{WAIT}$ hold time	t_{WTH}	0	—	ns		32.18 to 32.22, 32.35, 32.37, 32.39, 32.40
$\overline{RAS}$ delay time 2	t_{RASD2}	1.5	12	ns		32.23 to 32.33
$\overline{CAS}$ delay time 2	t_{CASD2}	1.5	12	ns		32.23 to 32.33
DQM delay time	t_{DQMD}	1.5	10	ns		32.23 to 32.30
CKE delay time	t_{CKED}	1.5	12	ns		32.32
$\overline{ICIORD}$ delay time	t_{ICRSD}	—	12	ns		32.38 to 32.40
$\overline{ICIOWR}$ delay time	t_{ICWSD}	—	12	ns		32.38 to 32.40
$\overline{IOIS16}$ setup time	t_{IO16S}	12	—	ns		32.39, 32.40
$\overline{IOIS16}$ hold time	t_{IO16H}	4	—	ns		32.39, 32.40
DACK delay time 1	t_{DAKD1}	—	10	ns		32.17 to 32.30, 32.33 to 32.40

Notes: 1. t_{AH}: This is to deal with the latest negate timing of $\overline{CS}$n, RD, or $\overline{WE}$n.
2. t_{RDH1}: This is to deal with the earliest negate timing of $\overline{CS}$n or RD.

32.3.4 Basic Timing

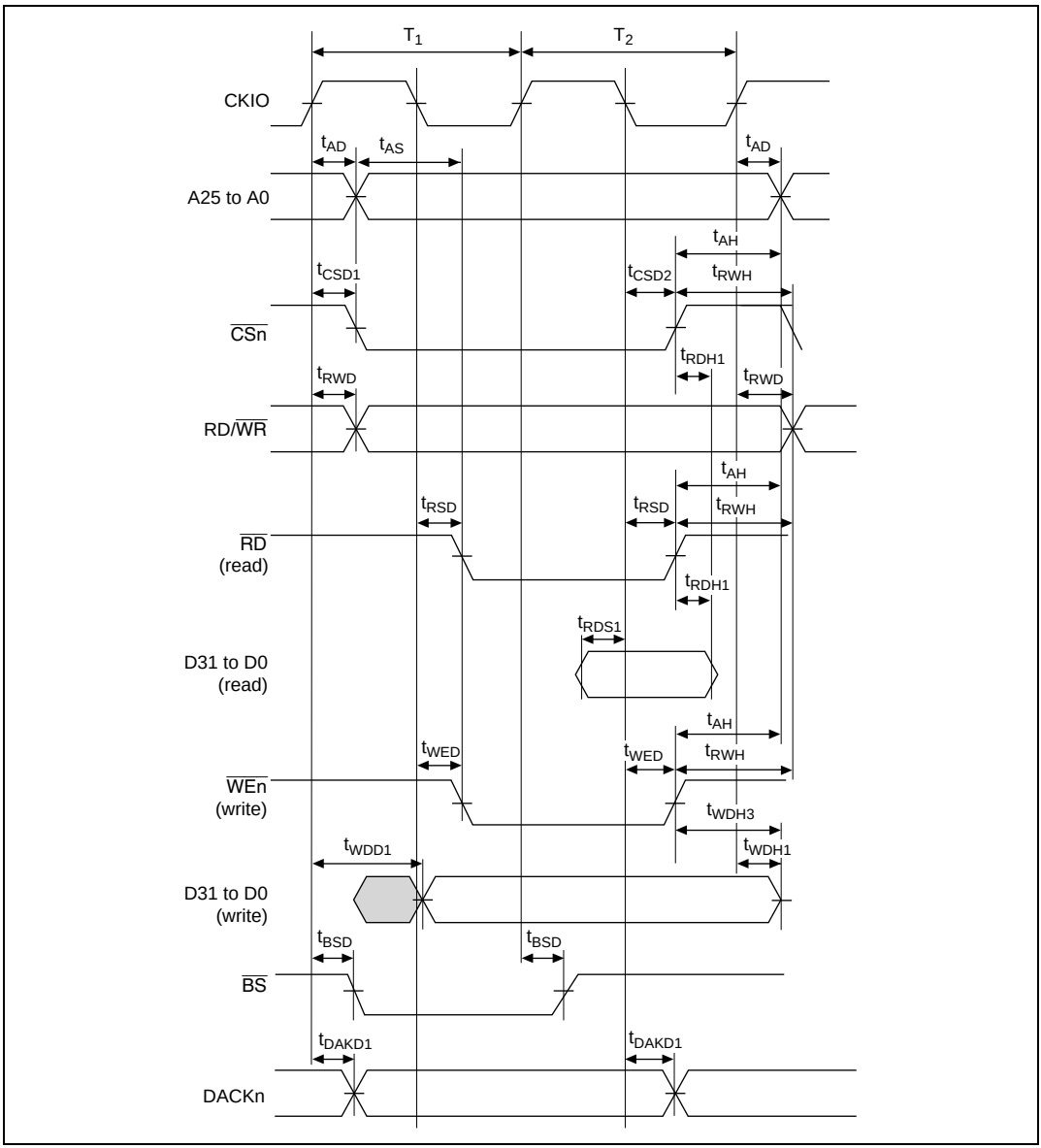


Figure 32.17 Basic Bus Cycle (No Wait)

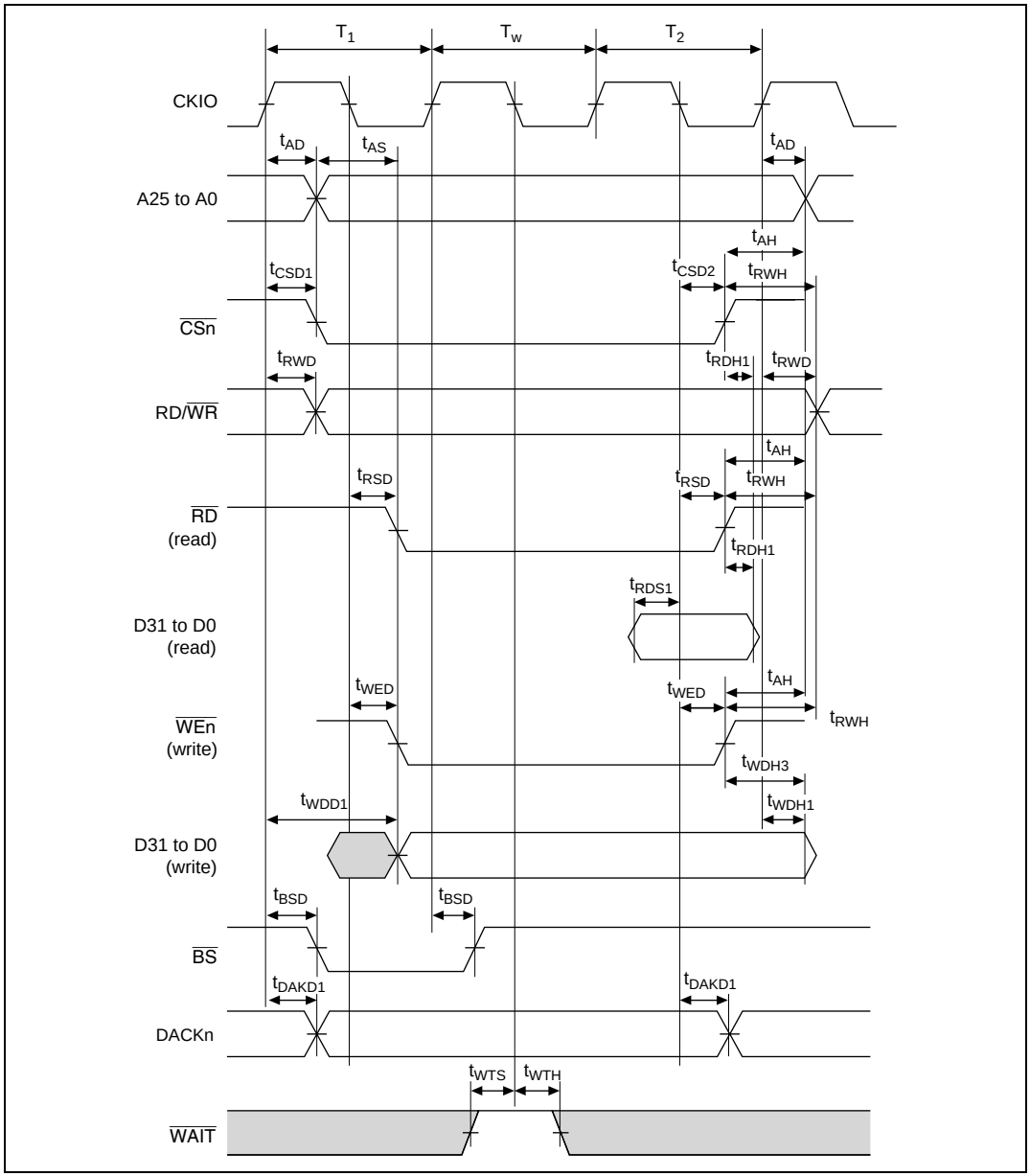
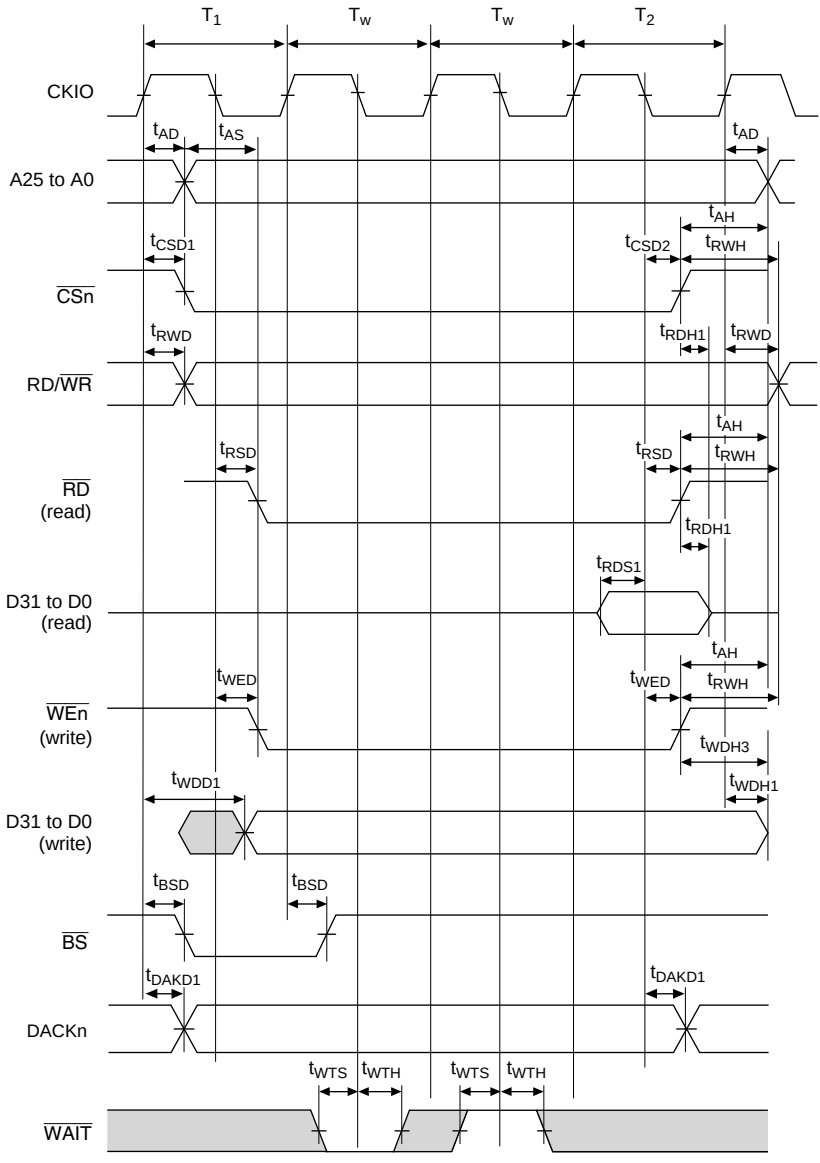


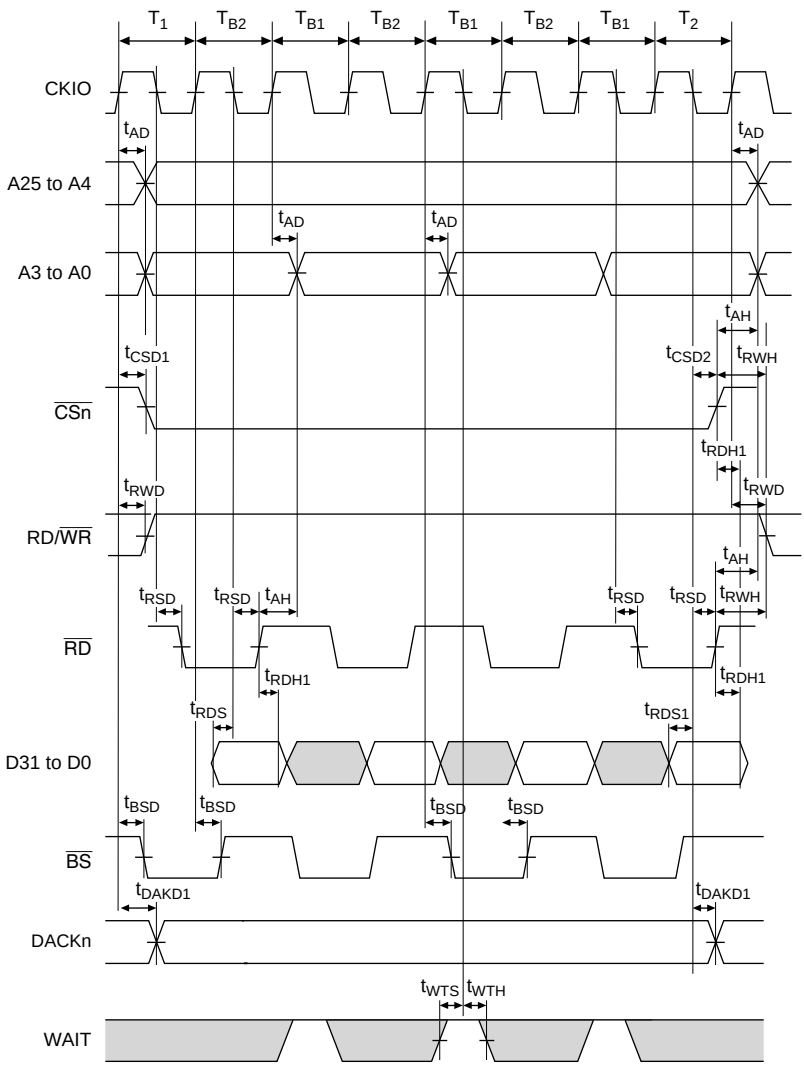
Figure 32.18 Basic Bus Cycle (One Wait)



Notes: t_{RDH1}: Specified based on the earliest negate timing of $\overline{CSn}$ or $\overline{RD}$.
t_{AH}: Specified based on the latest negate timing of $\overline{CSn}$, $\overline{RD}$, or $\overline{WEn}$.

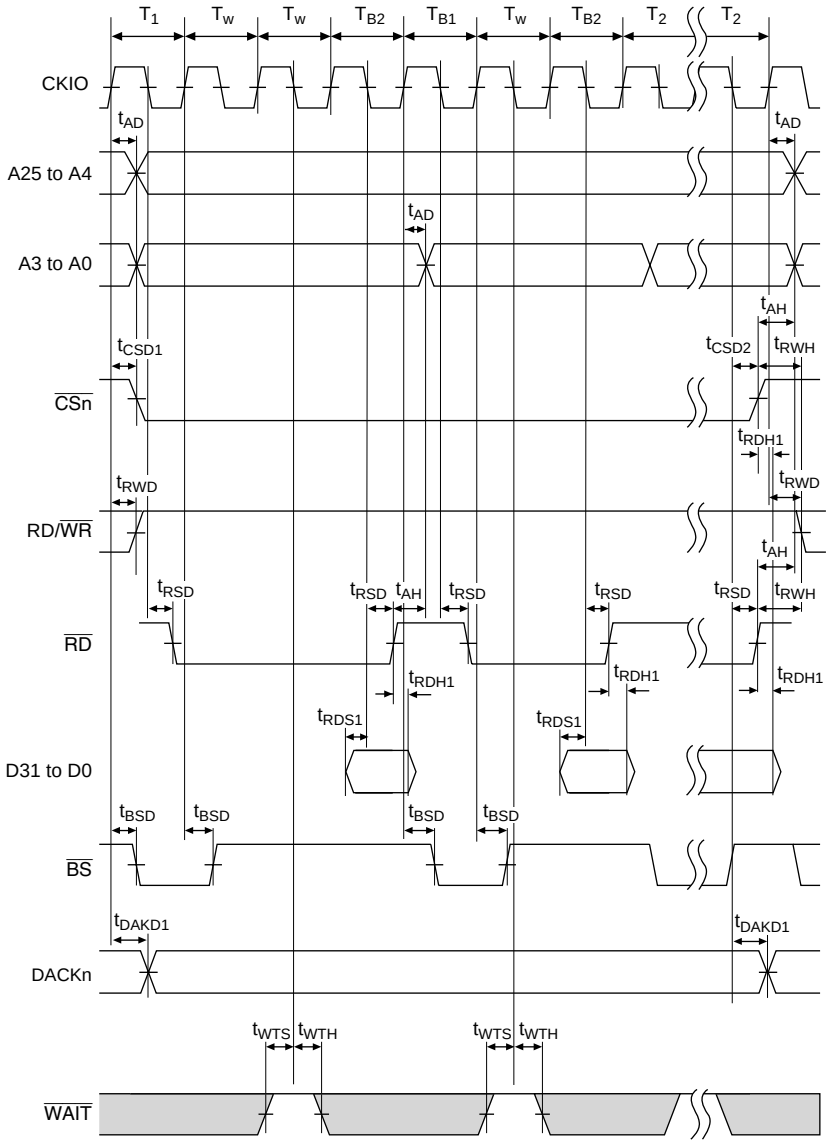
Figure 32.19 Basic Bus Cycle (External Wait, WAITSEL = 1)

32.3.5 Burst ROM Timing



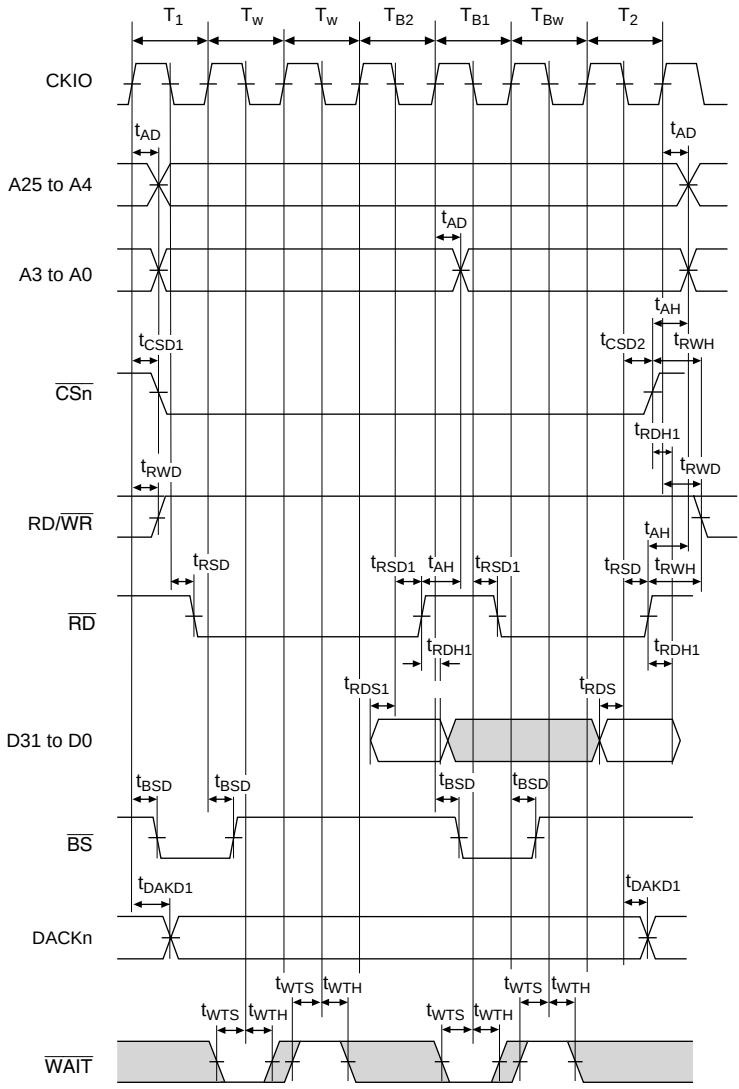
Note: In the write cycle, the basic bus cycle is performed.

Figure 32.20 Burst ROM Bus Cycle (No Wait)



Note: In the write cycle, the basic bus cycle is performed.

Figure 32.21 Burst ROM Bus Cycle (Two Waits)



Note: In the write cycle, the basic bus cycle is performed.

Figure 32.22 Burst ROM Bus Cycle (External Wait, WAITSEL = 1)

32.3.6 Synchronous DRAM Timing

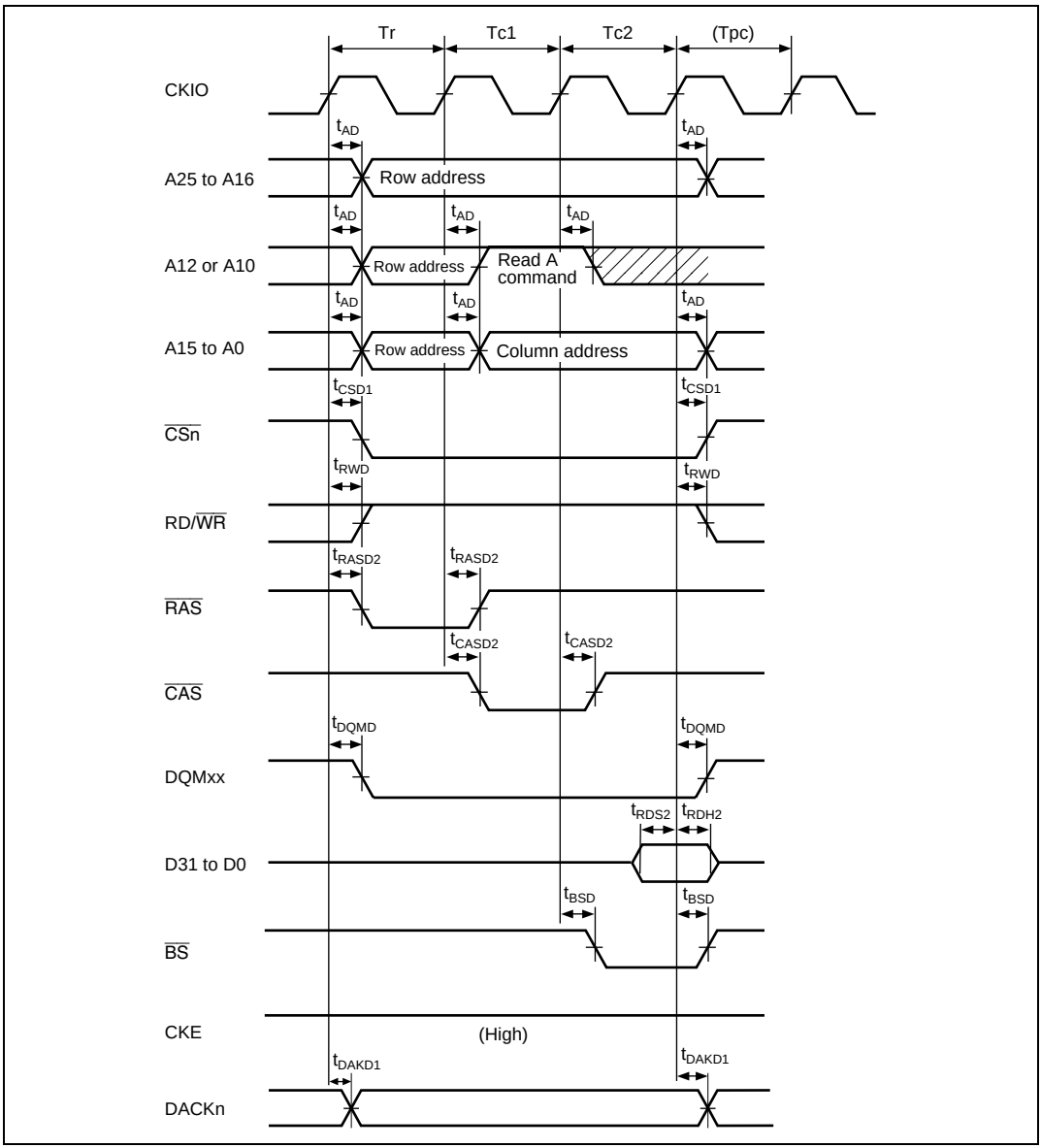


Figure 32.23 Synchronous DRAM Read Bus Cycle (RCD = 0, CAS Latency = 1, TPC = 0)

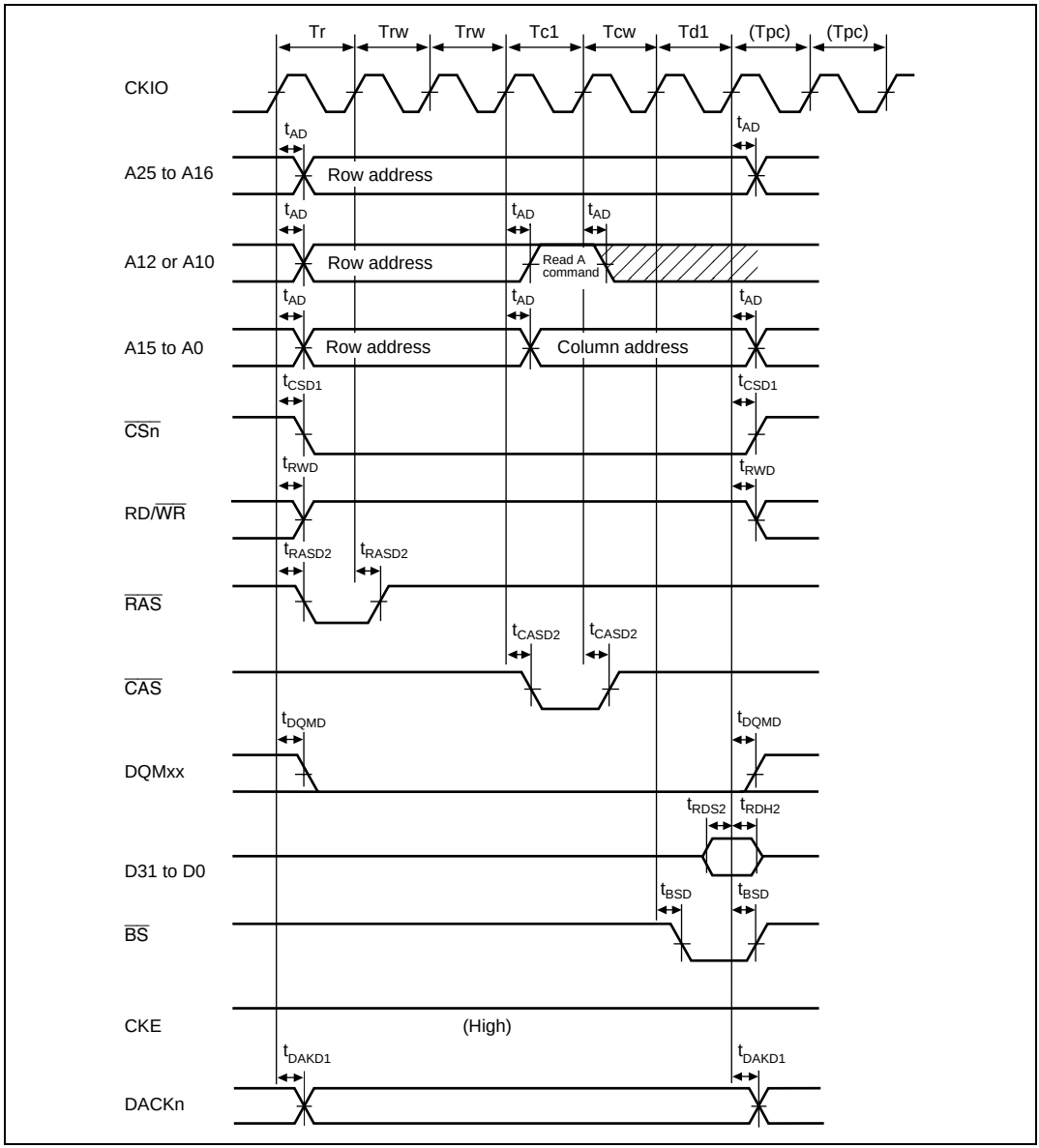


Figure 32.24 Synchronous DRAM Read Bus Cycle (RCD = 2, CAS Latency = 2, TPC = 1)

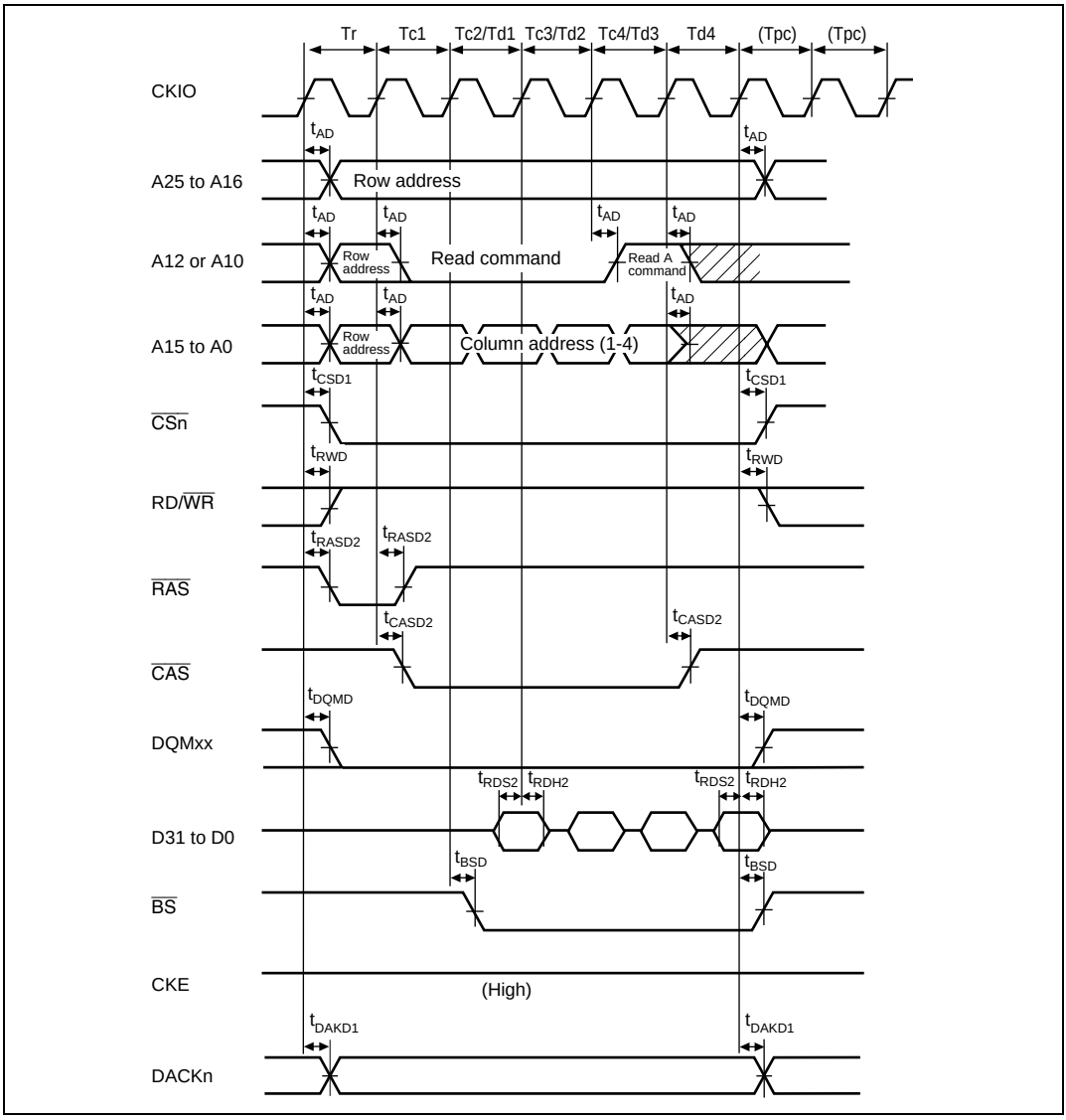


Figure 32.25 Synchronous DRAM Read Bus Cycle (Burst Read (Single Read × 4), RCD = 0, CAS Latency = 1, TPC = 1)

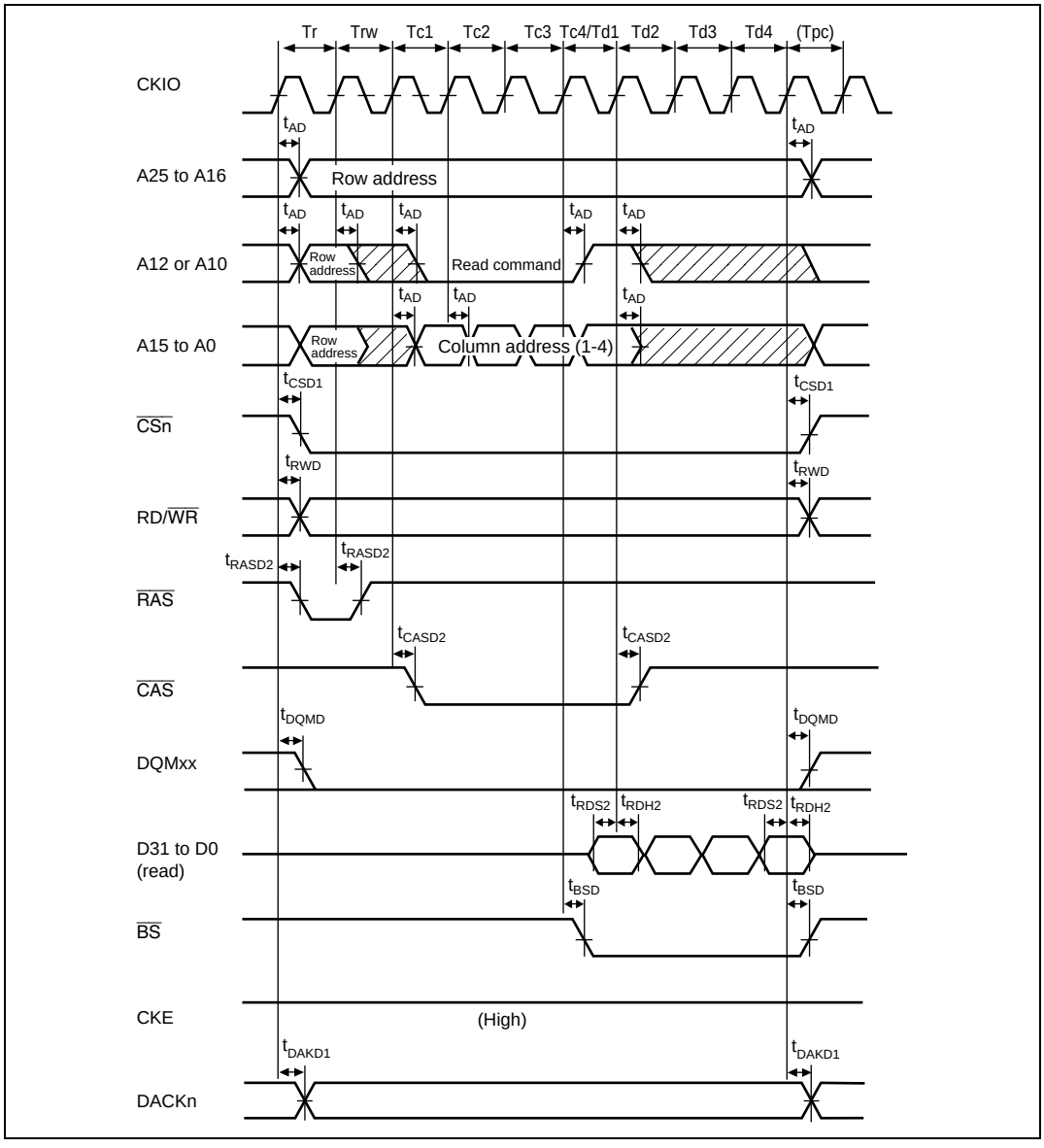


Figure 32.26 Synchronous DRAM Read Bus Cycle (Burst Read (Single Read × 4), RCD = 1, CAS Latency = 3, TPC = 0)

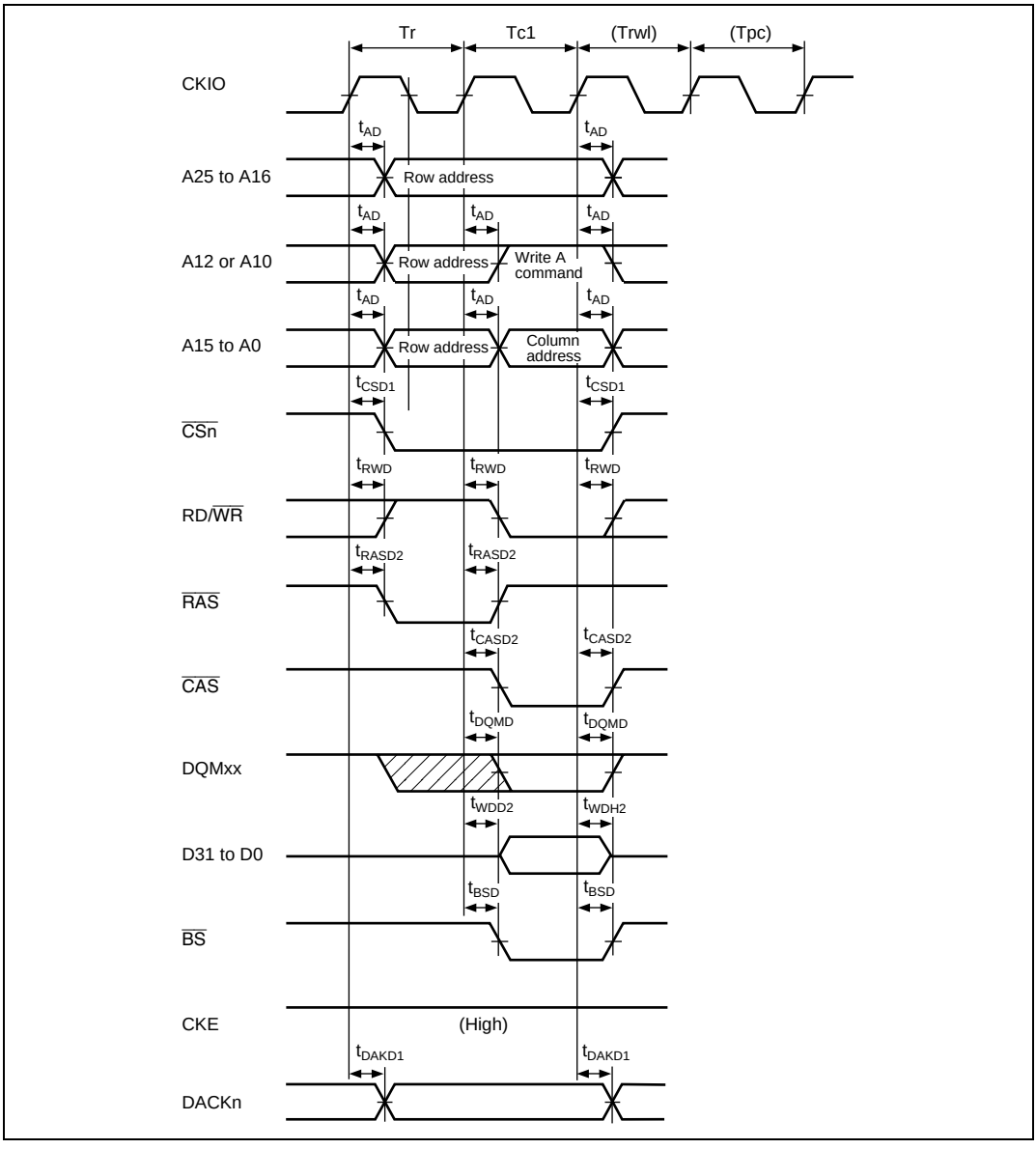


Figure 32.27 Synchronous DRAM Write Bus Cycle (RCD = 0, TPC = 0, TRWL = 0)

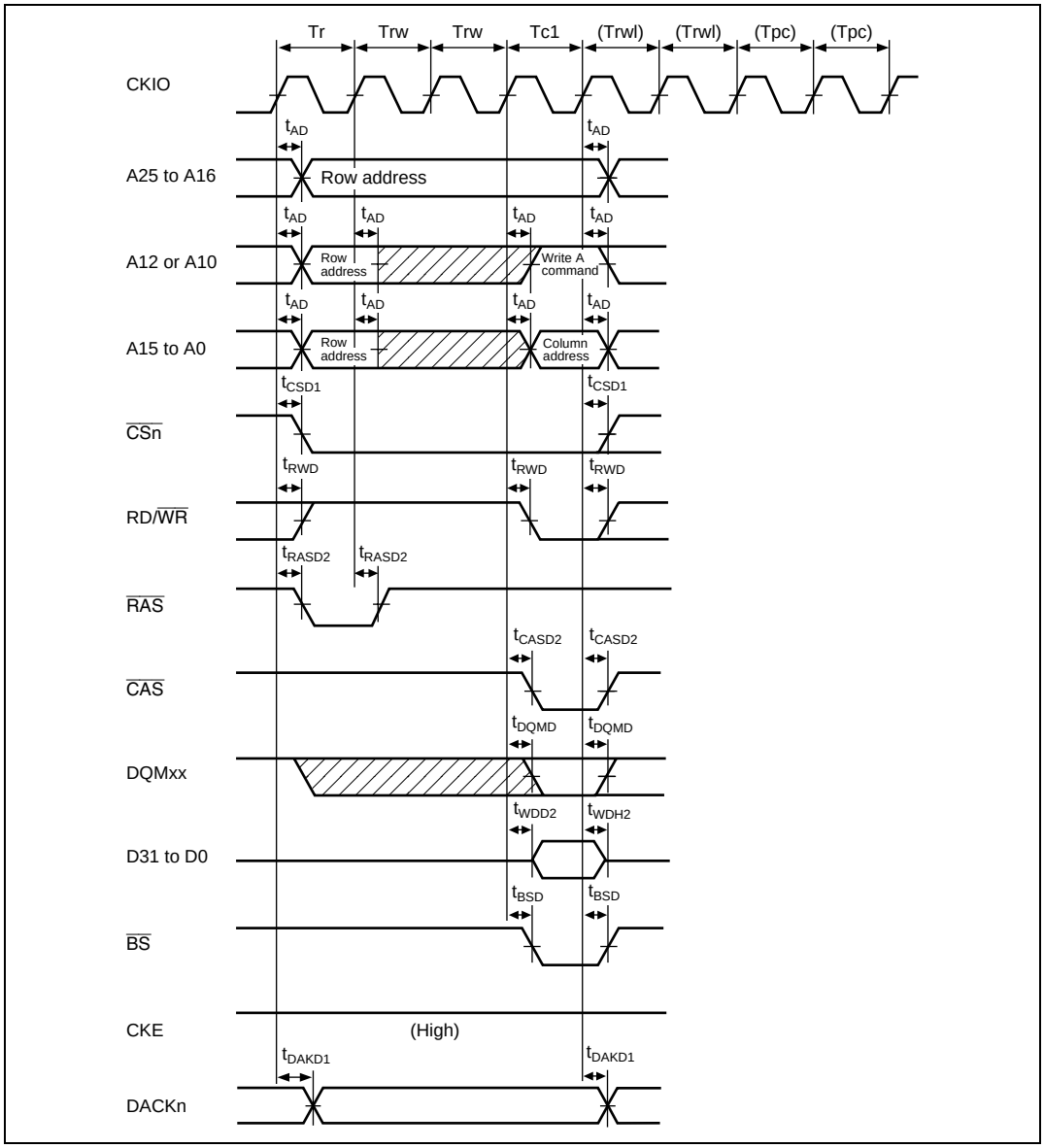


Figure 32.28 Synchronous DRAM Write Bus Cycle (RCD = 2, TPC = 1, TRWL = 1)

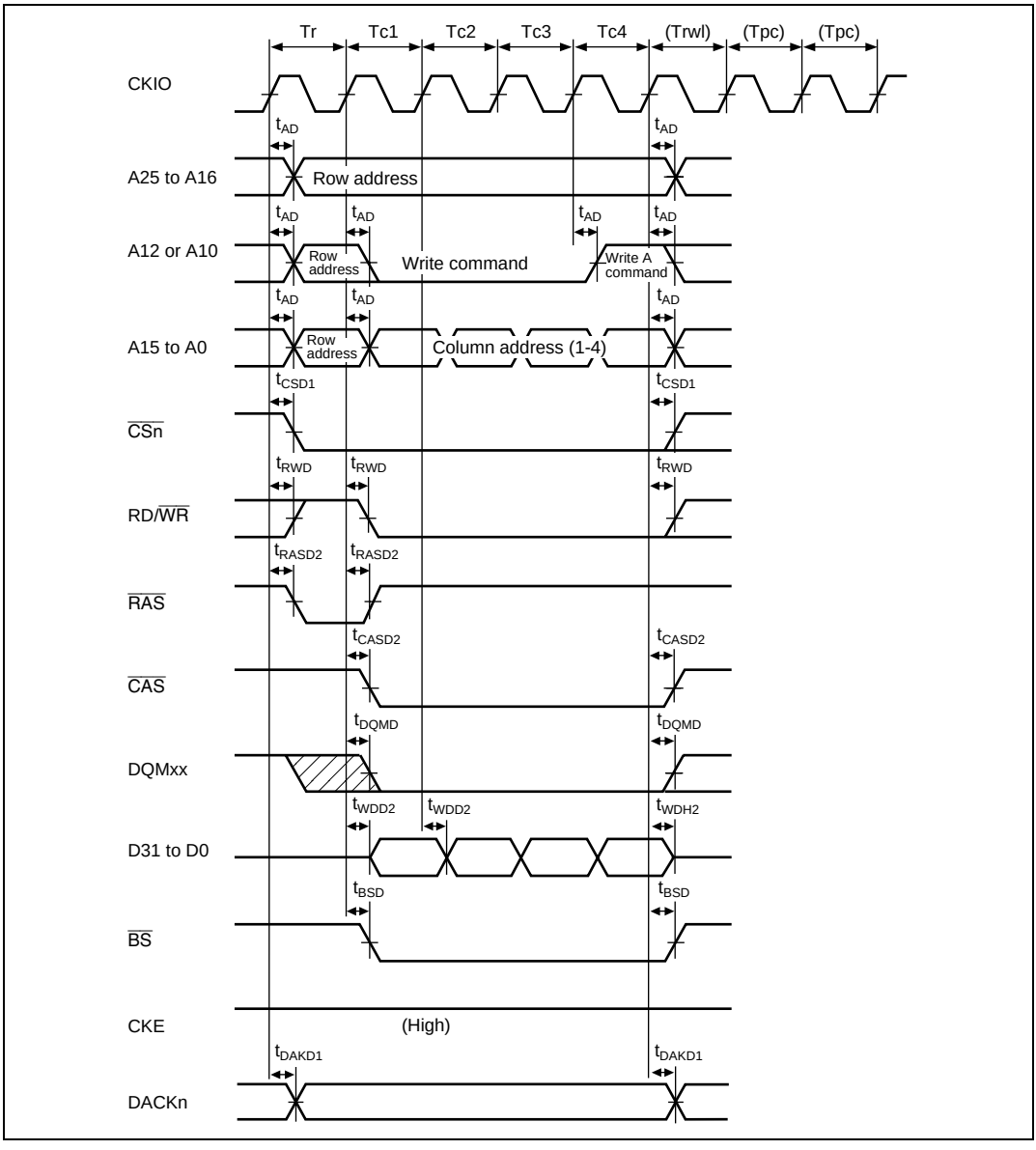


Figure 32.29 Synchronous DRAM Write Bus Cycle (Burst Mode (Single Write × 4), RCD = 0, TPC = 1, TRWL = 0)

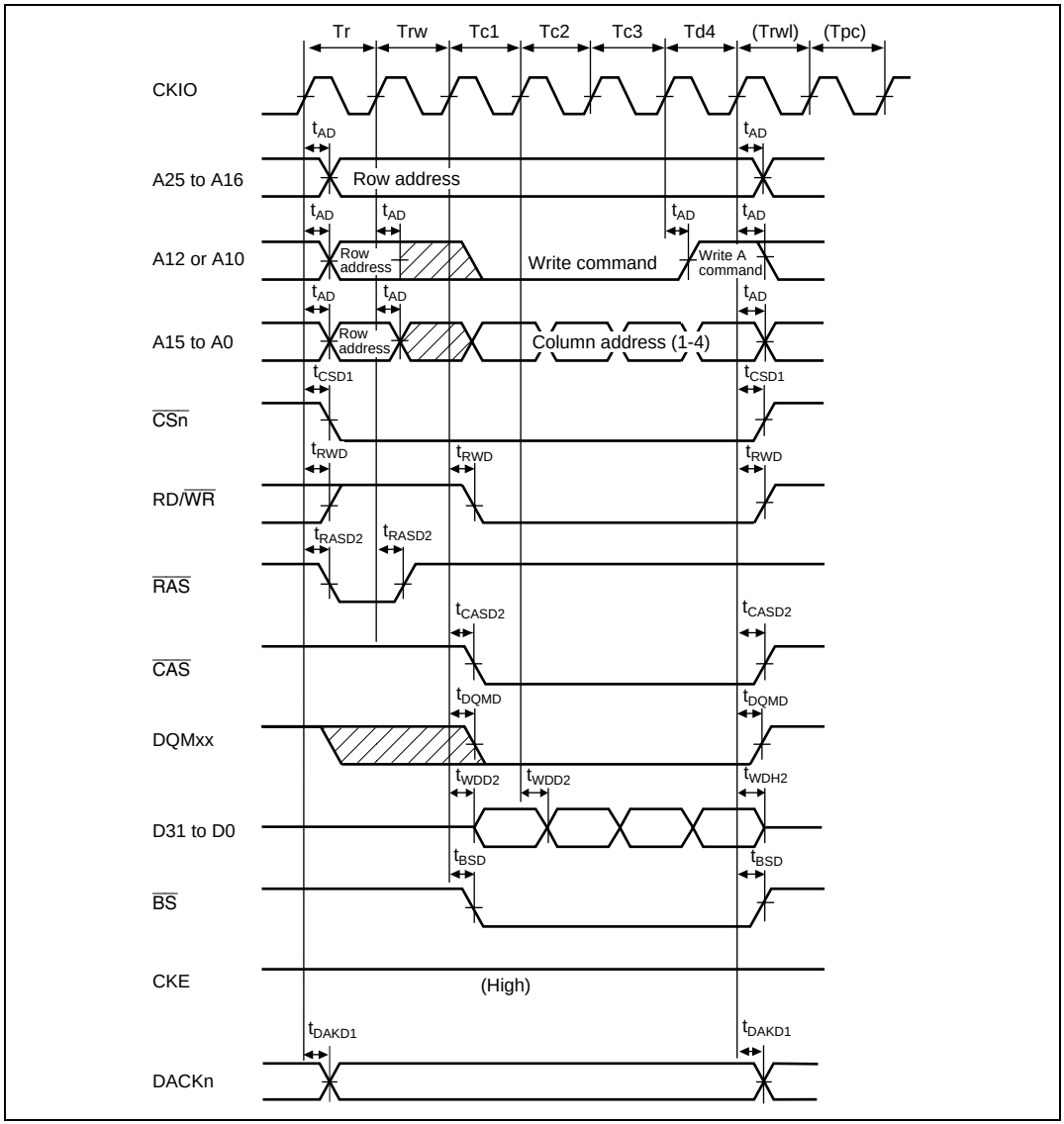


Figure 32.30 Synchronous DRAM Write Bus Cycle (Burst Mode (Single Write × 4), RCD = 1, TPC = 0, TRWL = 0)

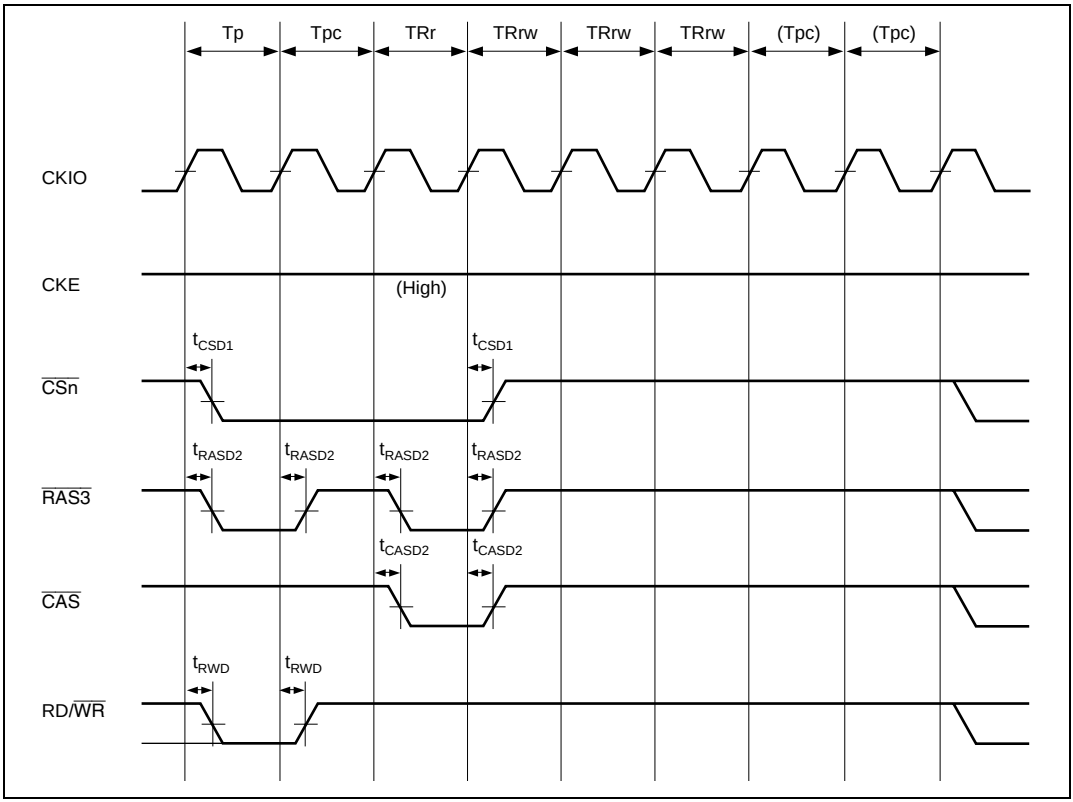


Figure 32.31 Synchronous DRAM Auto-Refresh Cycle (TRAS = 1, TPC = 1)

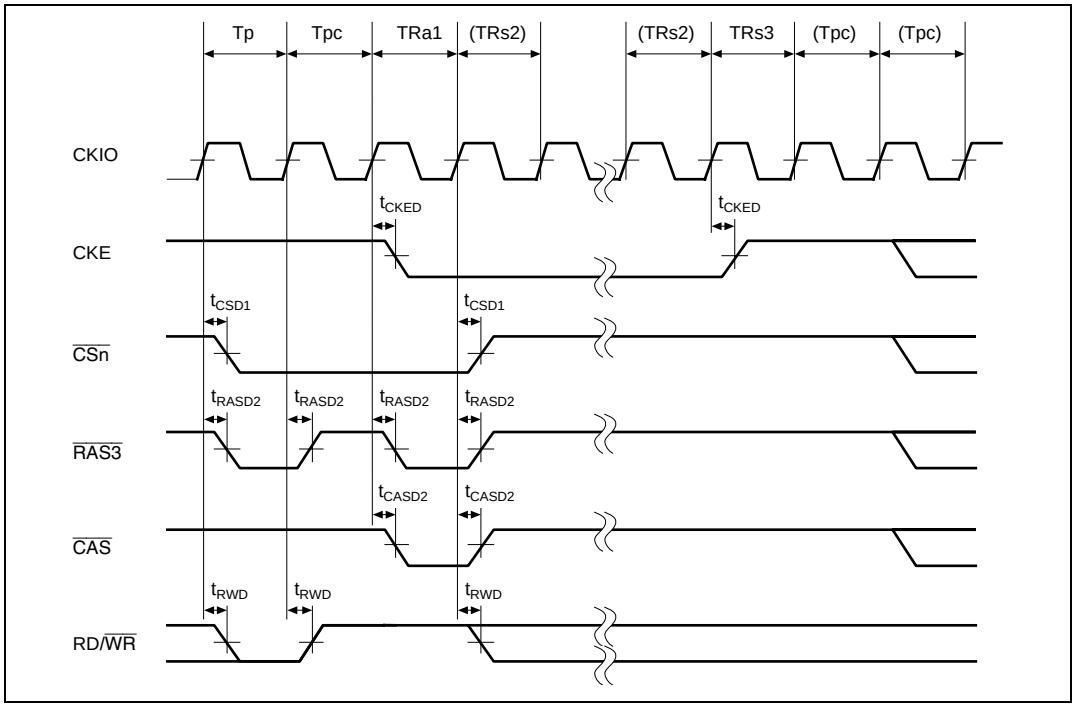


Figure 32.32 Synchronous DRAM Self-Refresh Cycle (TPC = 0)

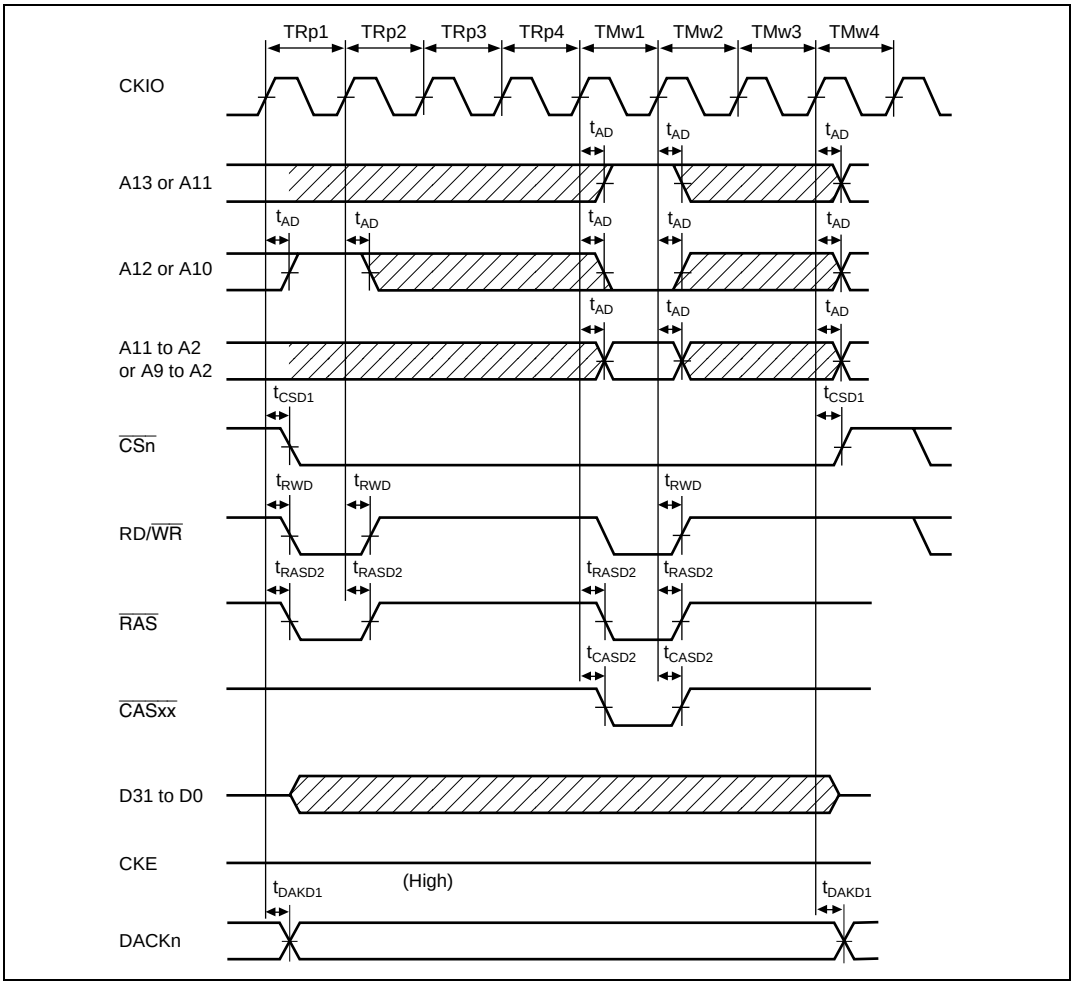


Figure 32.33 Synchronous DRAM Mode Register Write Cycle

32.3.7 PCMCIA Timing

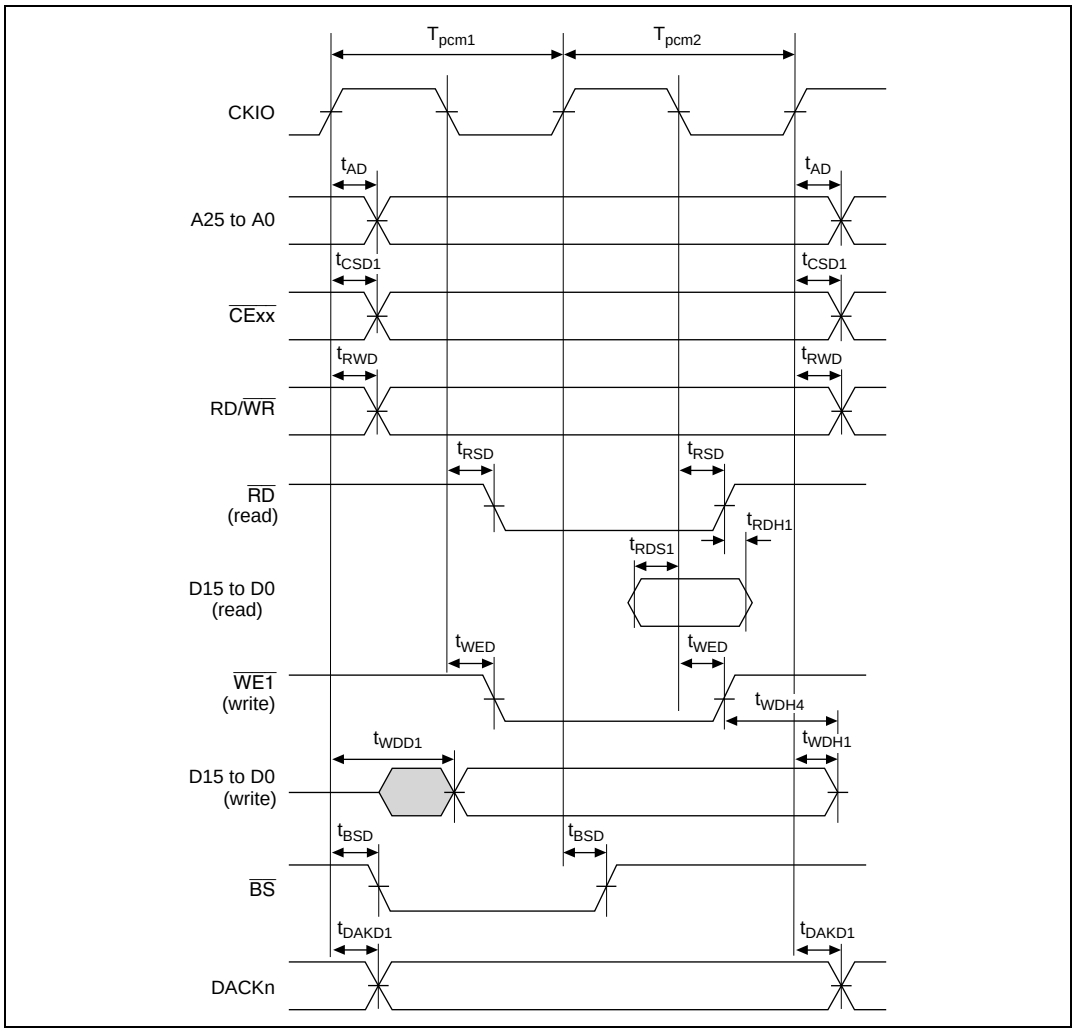


Figure 32.34 PCMCIA Memory Bus Cycle (TED = 0, TEH = 0, No Wait)

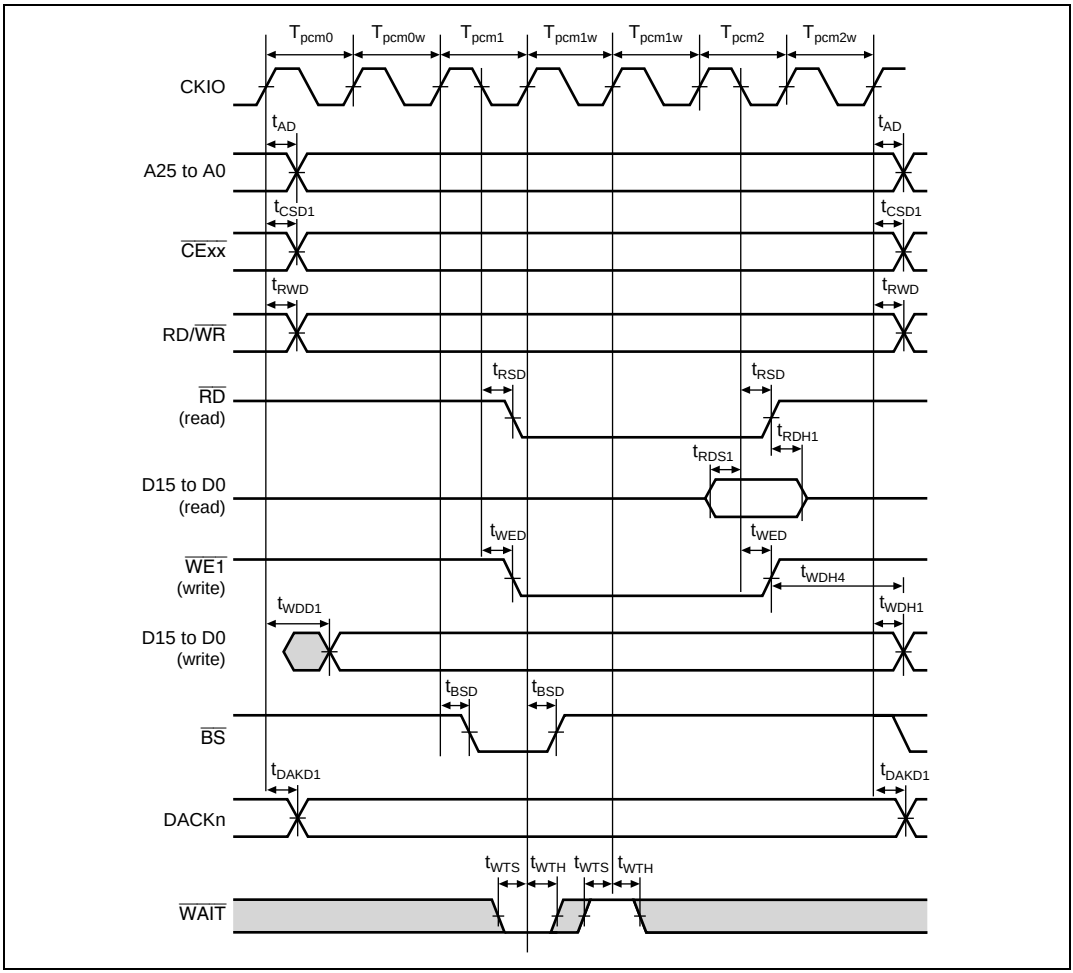
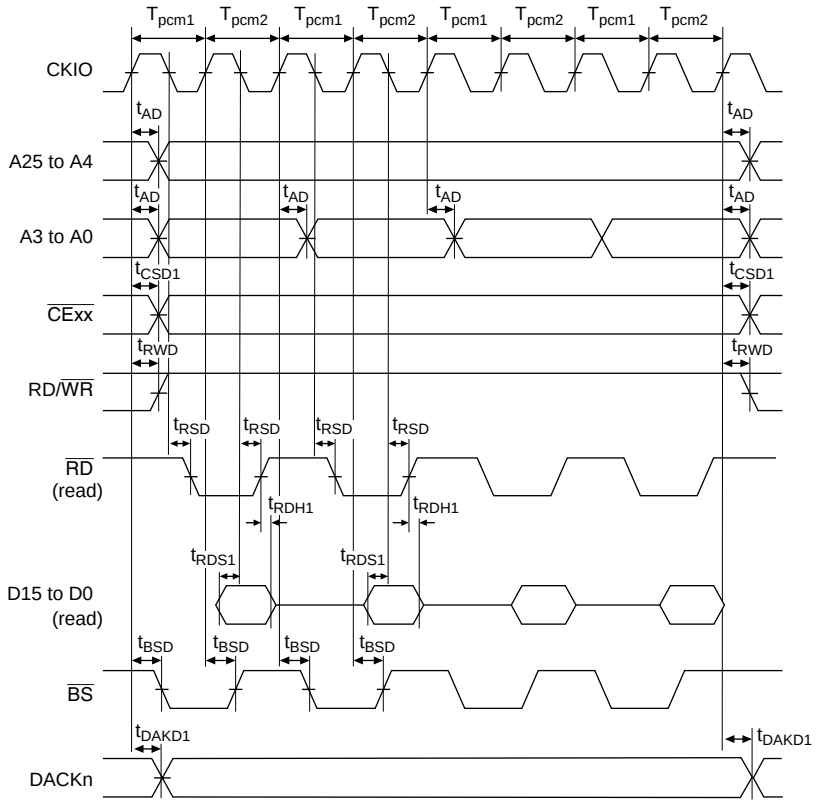
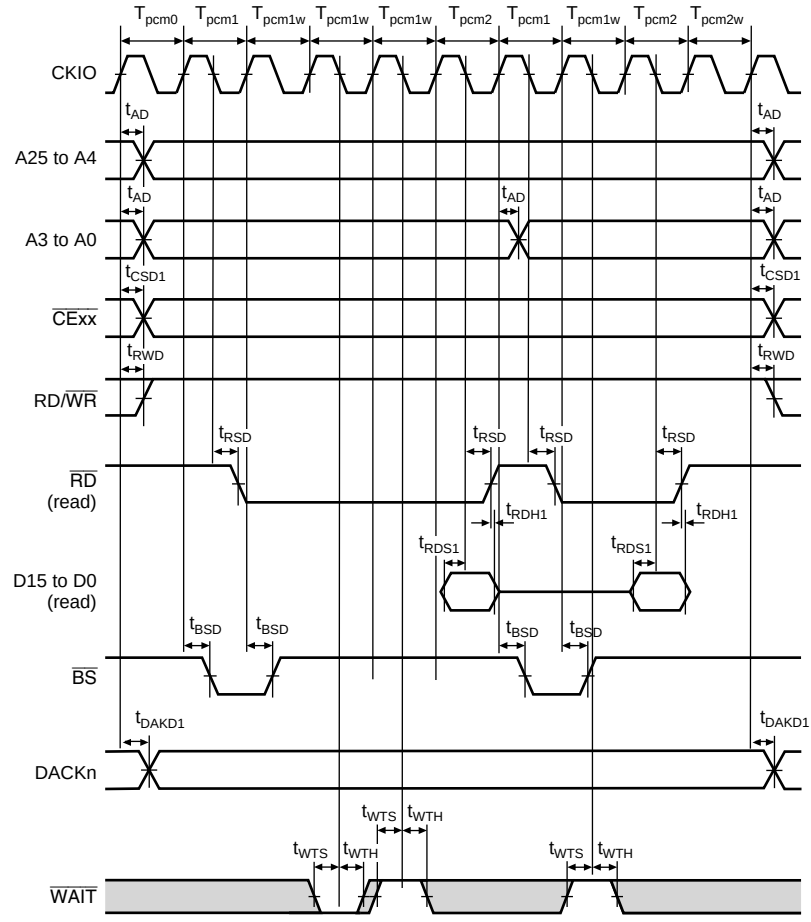


Figure 32.35 PCMCIA Memory Bus Cycle
(TED = 2, TEH = 1, One Wait, External Wait, WAITSEL = 1)



Note: Even though burst mode is set, write cycle operation is the same as in normal mode.

Figure 32.36 PCMCIA Memory Bus Cycle
(Burst Read, TED = 0, TEH = 0, No Wait)



Note: Even though burst mode is set, the write cycle operation is the same as in normal mode.

Figure 32.37 PCMCIA Memory Bus Cycle
(Burst Read, TED = 1, TEH = 1, Two Waits, Burst Pitch = 3, WAITSEL = 1)

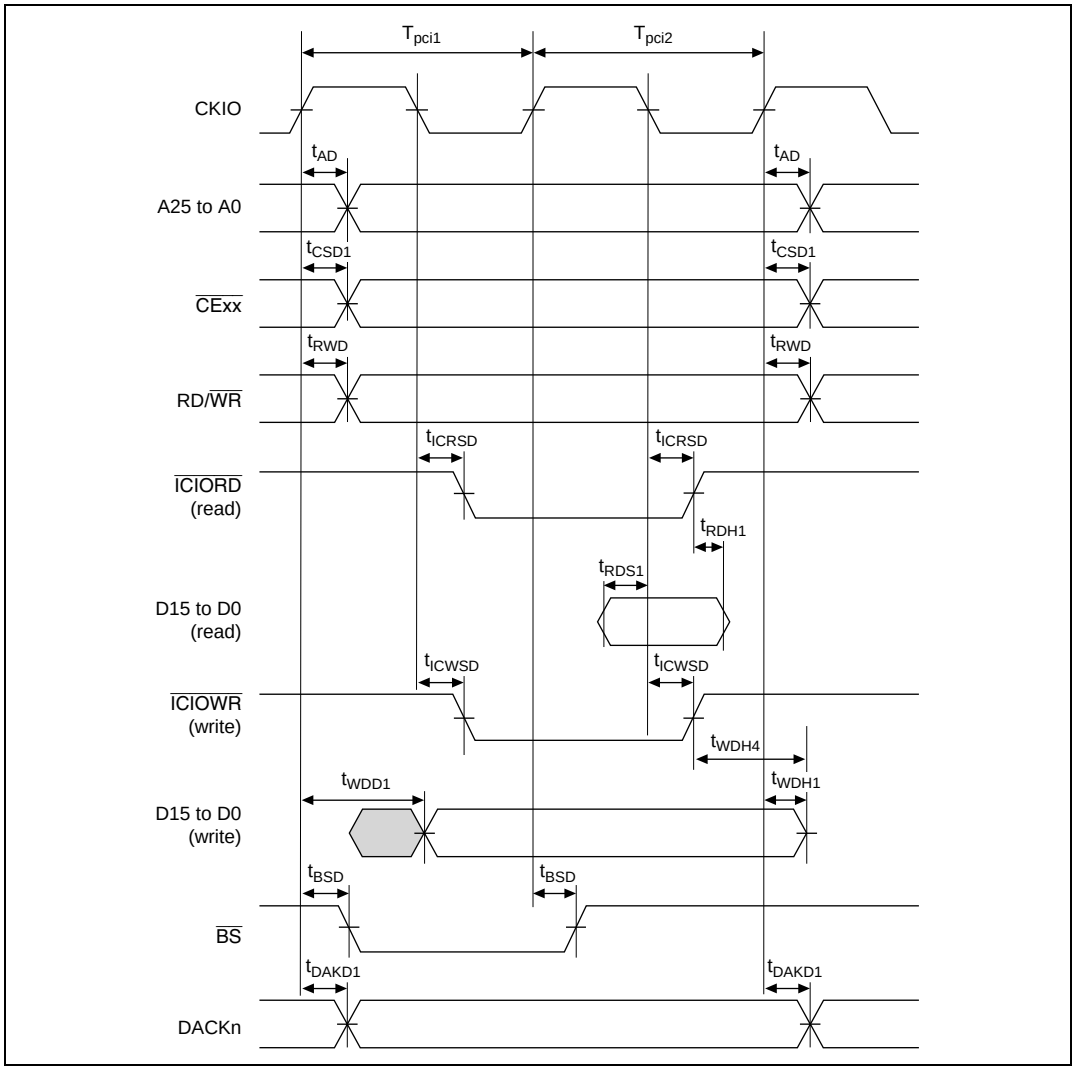


Figure 32.38 PCMCIA I/O Bus Cycle (TED = 0, TEH = 0, No Wait)

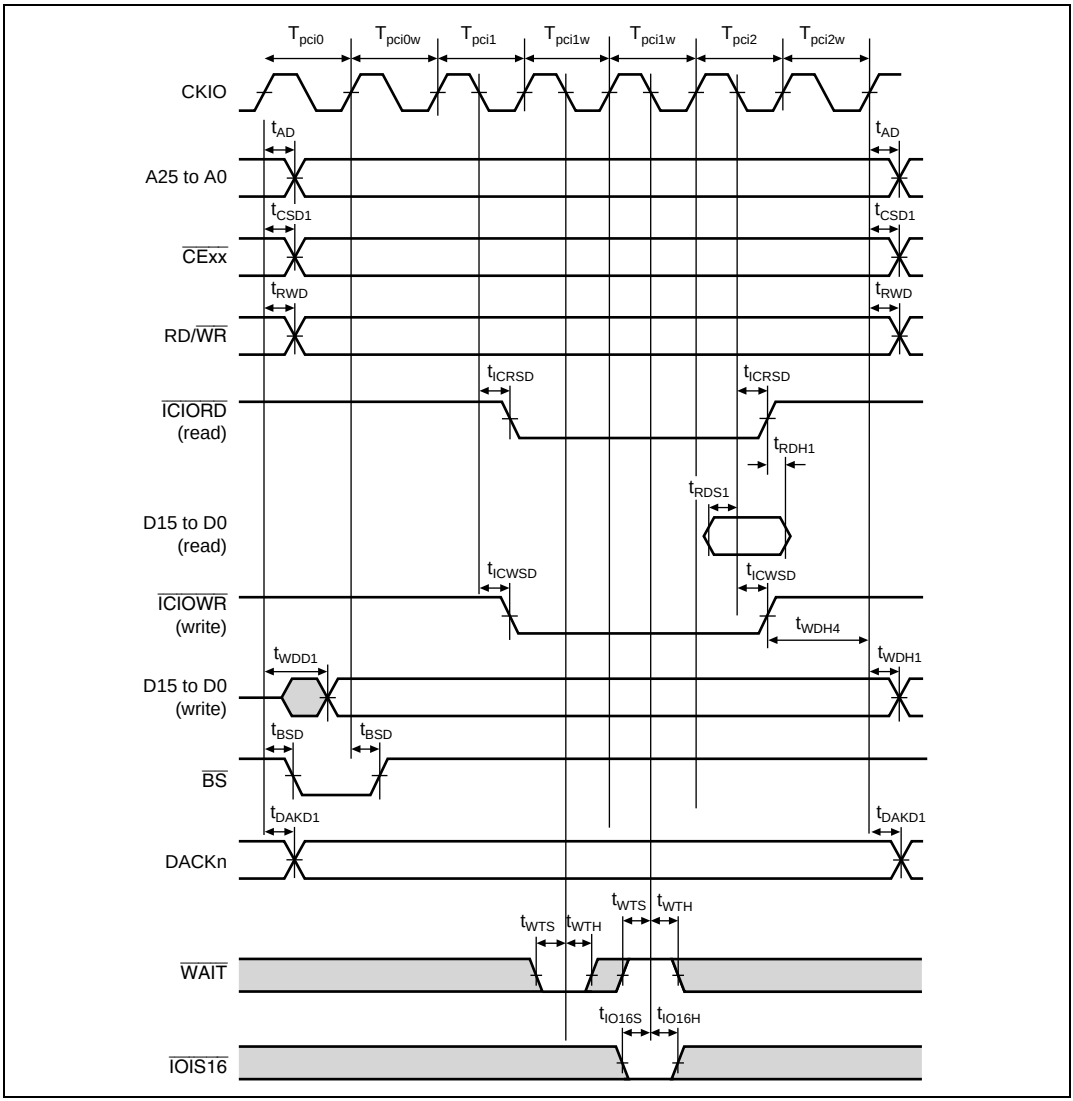


Figure 32.39 PCMCIA I/O Bus Cycle
(TED = 2, TEH = 1, One Wait, External Wait, WAITSEL = 1)

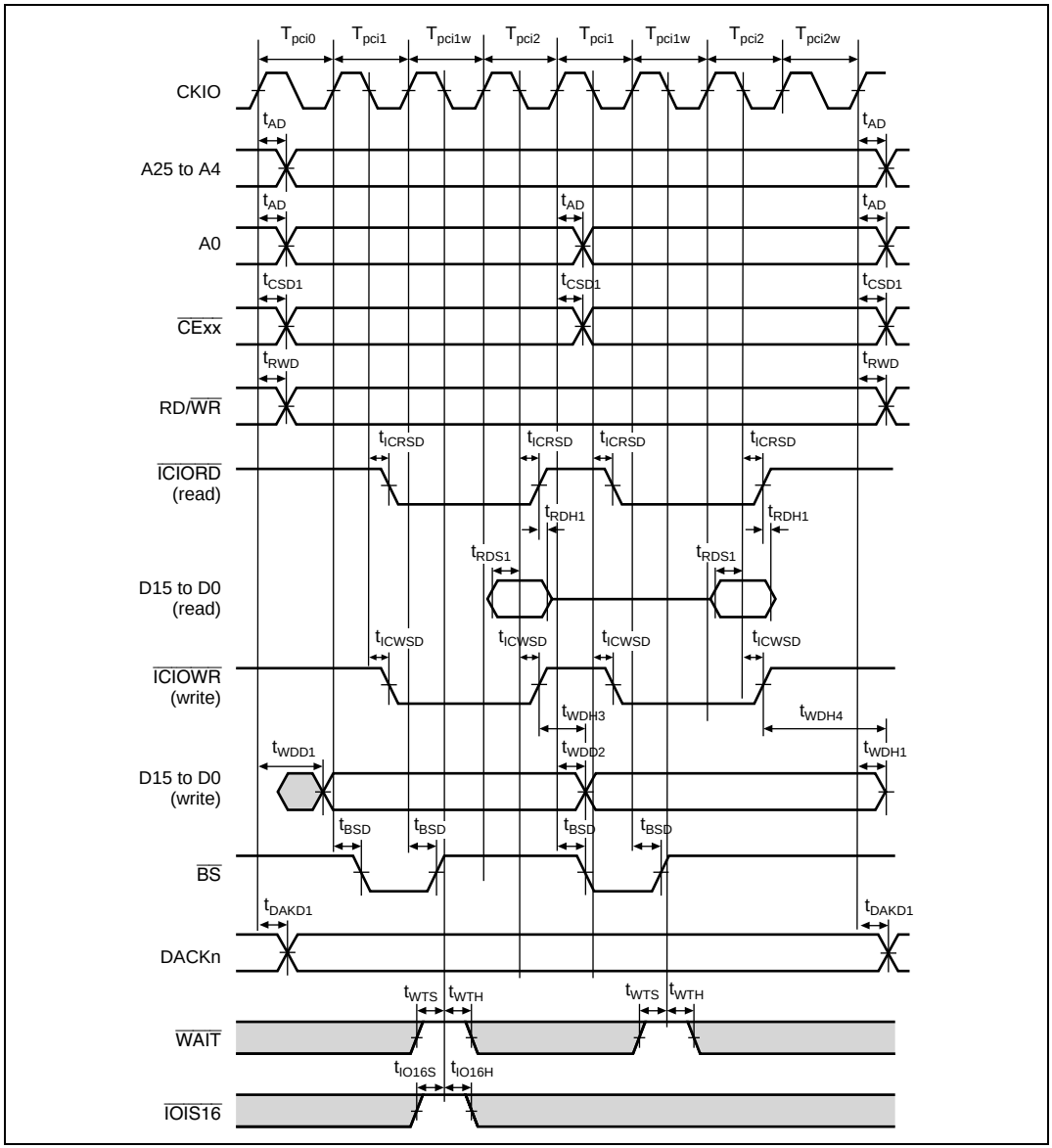


Figure 32.40 PCMCIA I/O Bus Cycle
(TED = 1, TEH = 1, One Wait, Bus Sizing, WAITSEL = 1)

32.3.8 Peripheral Module Signal Timing

Table 32.11 Peripheral Module Signal Timing

Conditions: VccQ = 2.6 to 3.6 V, Vcc = 1.60 to 2.05 V, AVcc = 3.3 ± 0.3 V, Ta = –20 to 75°C

Module	Item		Symbol	Min	Max	Unit	Figure
RTC	Oscillation settling time		t_{ROSC}	—	3	s	32.41
SCI	Input clock cycle	Asynchronization	t_{SCYC}	4	—	P_{cyc}*	32.42
		Clock synchronization		6	—		32.43
	Input clock rise time		t_{SCKR}	—	1.5	P_{cyc}*	32.42
	Input clock fall time		t_{SCKF}	—	1.5		
	Input clock pulse width		t_{SCKW}	0.4	0.6	t_{Scyc}	
	Transmission data delay time		t_{TXD}	—	100	ns	32.43
	Receive data setup time (clock synchronization)		t_{RXS}	100	—		
	Receive data hold time (clock synchronization)		t_{RXH}	100	—		
	$\overline{\text{RTS}}$ delay time		t_{RTSD}	—	100		
	$\overline{\text{CTS}}$ setup time (clock synchronization)		t_{CTSS}	100	—		
	$\overline{\text{CTS}}$ hold time (clock synchronization)		t_{CTSH}	100	—		
Port	Output data delay time		t_{PORTD}	—	26	ns	32.44
	Input data setup time (1)		t_{PORTS1}	15	—		
	Input data hold time (1)		t_{PORTH1}	8	—		
	Input data setup time (2)		t_{PORTS2}	t_{cyc} + 15	—		
	Input data hold time (2)		t_{PORTH2}	8	—		
	Input data setup time (3)		t_{PORTS3}	3 t_{cyc} + 15	—		
	Input data hold time (3)		t_{PORTH3}	8	—		
DMAC	$\overline{\text{DREQ}}$ setup time		t_{DRQS}	8	—	ns	32.45
	$\overline{\text{DREQ}}$ hold time		t_{DREQH}	8	—		
	DRAK delay time		t_{DRAKD}	—	14		32.46

Note: * P_{cyc} stands for “peripheral clock (Pϕ) cycle.”

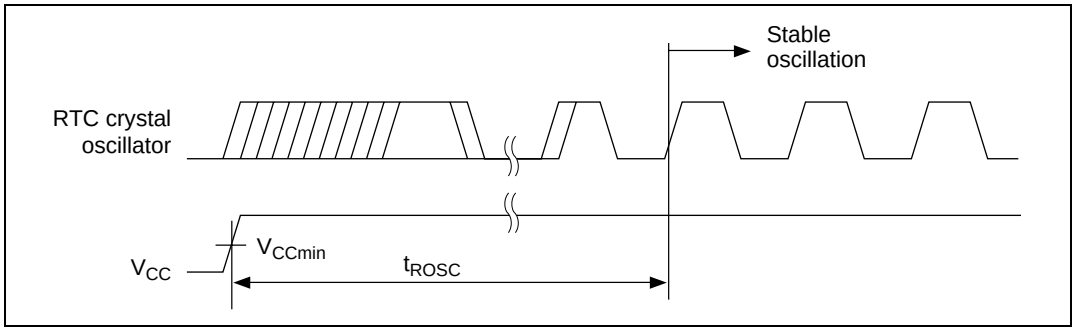


Figure 32.41 Oscillation Settling Time at RTC Crystal Oscillator Power-on

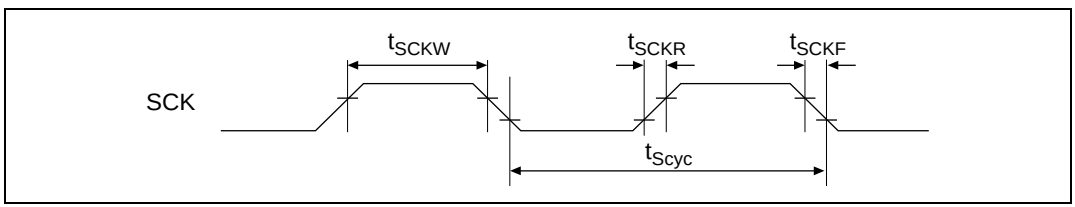


Figure 32.42 SCK Input Clock Timing

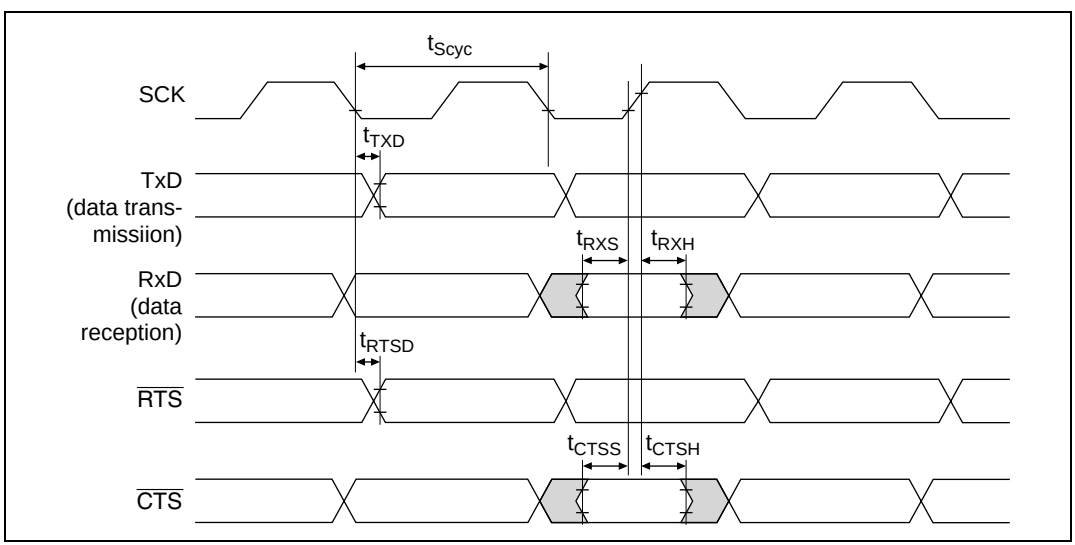


Figure 32.43 SCI I/O Timing in Clock Synchronous Mode

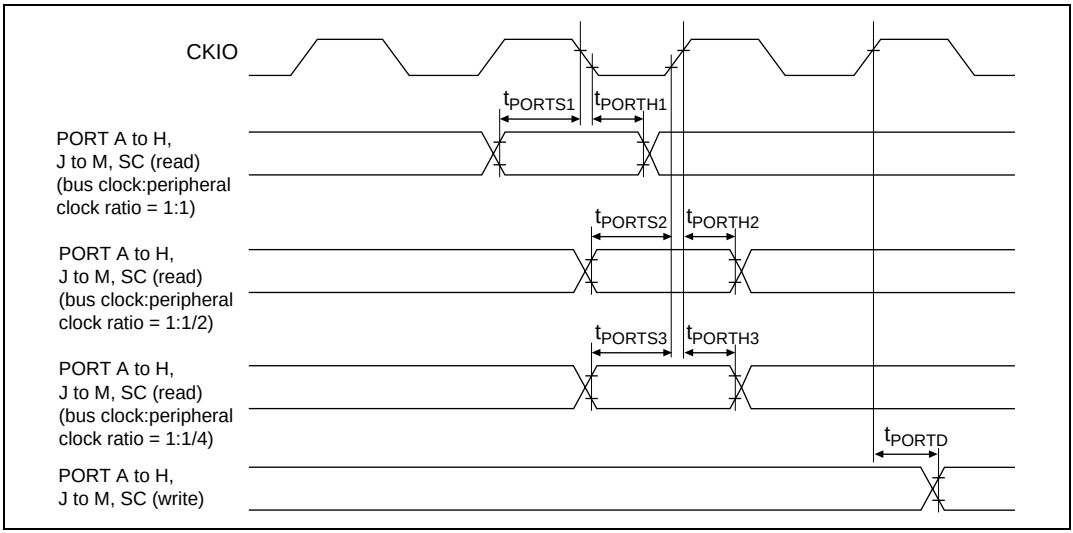


Figure 32.44 I/O Port Timing

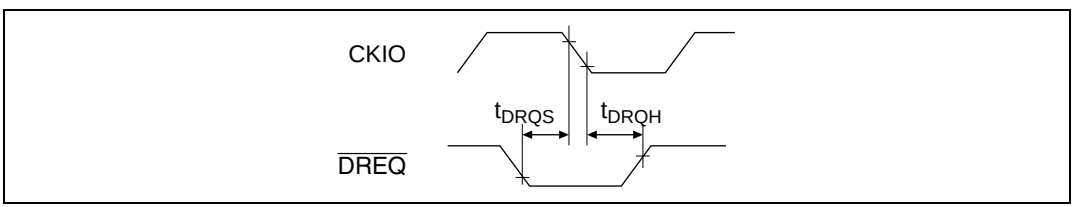


Figure 32.45 $\overline{\text{DREQ}}$ Input Timing

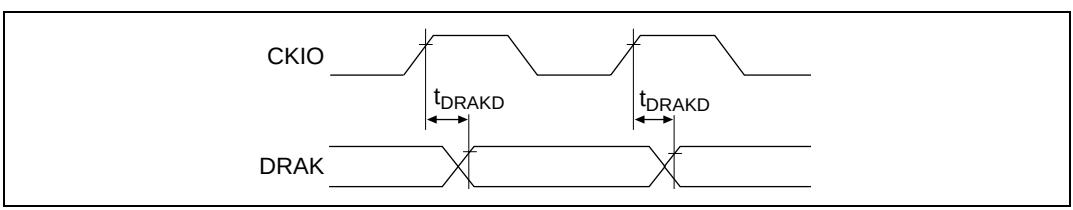


Figure 32.46 DRAK Output Timing

32.3.9 H-UDI-Related Pin Timing

Table 32.12 H-UDI-Related Pin Timing

Conditions: VccQ = 2.6 to 3.6 V, Vcc = 1.60 to 2.05 V, AVcc = 3.3 ± 0.3 V, Ta = –20 to 75°C

Item	Symbol	Min	Max	Unit	Figure
TCK cycle time	t_{TCKcyc}	50	—	ns	Figure 32.47
TCK high pulse width	t_{TCKH}	12	—	ns	
TCK low pulse width	t_{TCKL}	12	—	ns	
TCK rise/fall time	t_{TCKf}	—	4	ns	
$\overline{\text{TRST}}$ setup time	t_{TRSTS}	12	—	ns	Figure 32.48
$\overline{\text{TRST}}$ hold time	t_{TRSTH}	50	—	t_{cyc}	
TDI setup time	t_{TDIS}	10	—	ns	Figure 32.49
TDI hold time	t_{TDIH}	10	—	ns	
TMS setup time	t_{TMSS}	10	—	ns	
TMS hold time	t_{TMSH}	10	—	ns	
TDO delay time	t_{TDOD}	—	19	ns	
$\overline{\text{ASEMD0}}$ setup time	t_{ASEMDH}	12	—	ns	Figure 32.50
$\overline{\text{ASEMD0}}$ hold time	t_{ASEMDS}	12	—	ns	

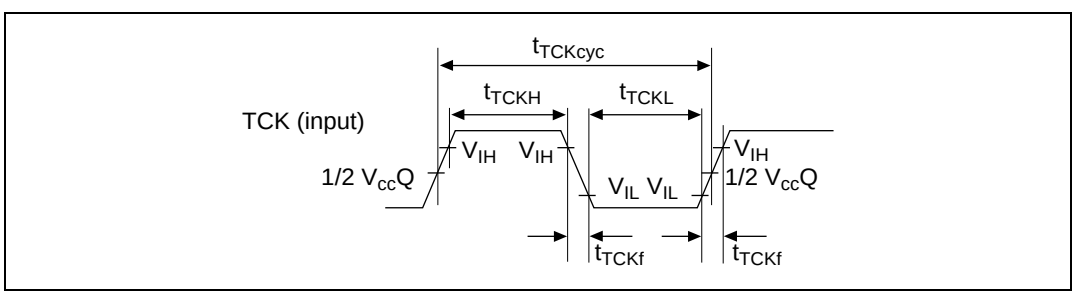


Figure 32.47 TCK Input Timing

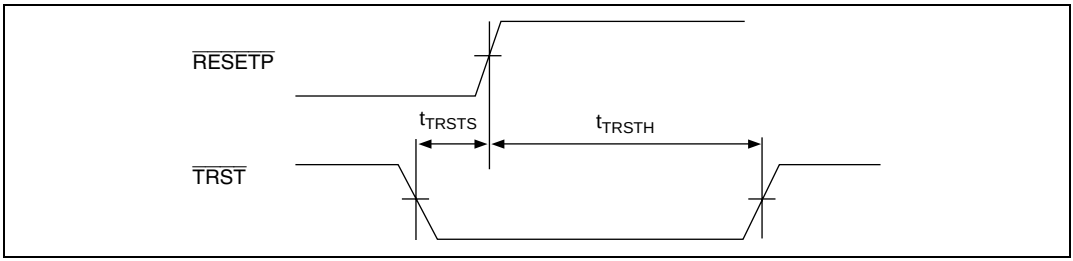


Figure 32.48 $\overline{\text{TRST}}$ Input Timing (Reset Hold)

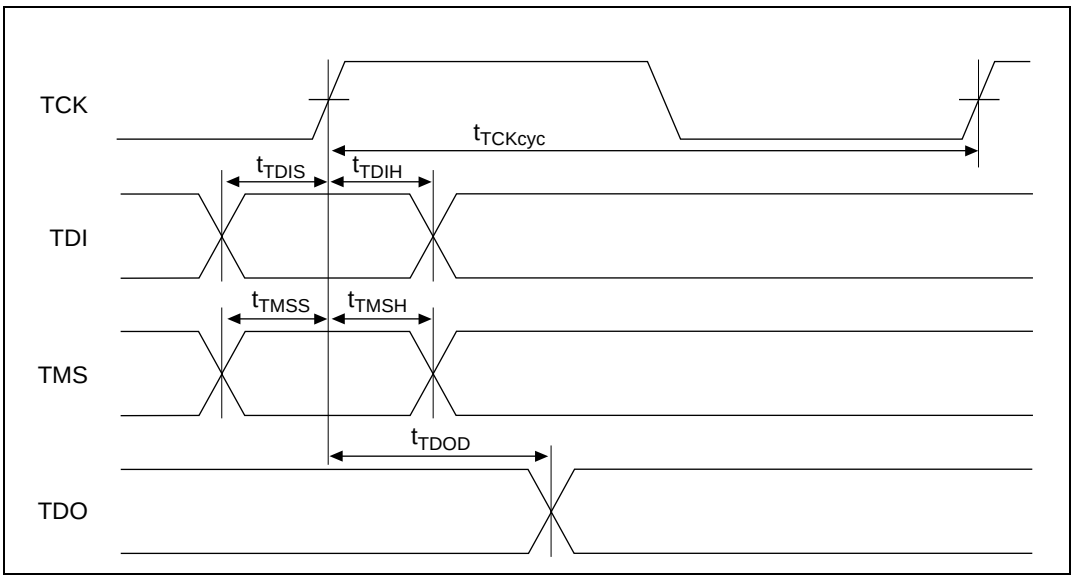


Figure 32.49 H-UDI Data Transfer Timing

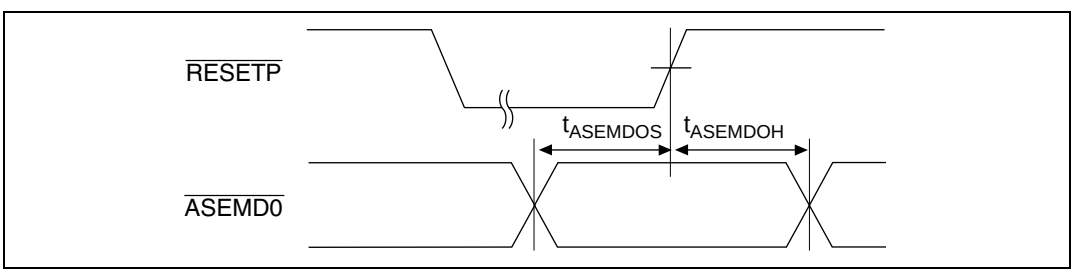


Figure 32.50 $\overline{\text{ASEMD0}}$ Input Timing

32.3.10 LCDC Timing

Table 32.13 LCDC Timing

Conditions: VccQ = 2.6 to 3.6 V, Vcc = 1.60 to 2.05 V, AVcc = 3.3 ± 0.3 V, Ta = –20 to 75°C

Item	Symbol	Min	Max	Unit	Figure
LCLK input clock frequency	t_{FREQ}	—	50	MHz	32.51
LCLK input clock rise time	t_R	—	3	ns	
LCLK input clock fall time	t_F	—	3	ns	
LCLK input clock duty	t_{DUTY}	90	110	%	
Clock (CL2/DCLK) cycle time	t_{CC}	25	—	ns	
Clock (CL2/DCLK) high-level width	t_{CHW}	7	—	ns	
Clock (CL2/DCLK) low-level width	t_{CLW}	7	—	ns	
Clock (CL2/DCLK) transition time (rise, fall)	t_{CT}	—	3	ns	
Data (LCD) delay time	t_{DD}	–3.5	3	ns	
Data (LCD) transition time (rise, fall)	t_{DT}	—	3	ns	
Display enable (M/DISP) delay time	t_{ID}	–3.5	3	ns	
Display enable (M/DISP) transition time (rise, fall)	t_{IT}	—	3	ns	
Horizontal sync. signal (CL1/Hsync) delay time	t_{HD}	–3.5	3	ns	
Horizontal sync. singal (CL1/Hsync) transition time	t_{HT}	—	3	ns	
Vertical sync. signal (FLM/Vsync) delay time	t_{VD}	–3.5	3	ns	
Vertical sync. signal (FLM/Vsync) transition time	t_{VT}	—	3	ns	

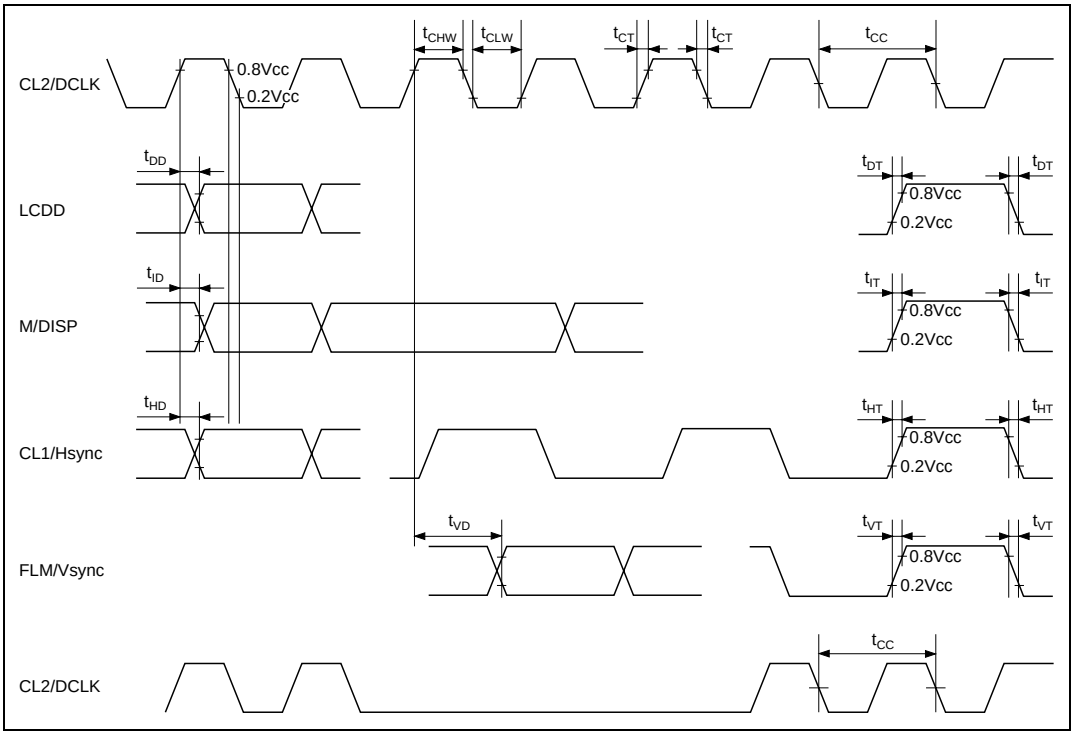


Figure 32.51 LCDC AC Specification

32.3.11 SIOF Module Signal Timing

Table 32.14 SIOF Module Signal Timing

Conditions: $V_{CC}Q$ = 2.6 to 3.6 V, V_{CC} = 1.60 to 2.05 V, AV_{CC} = 3.3 ± 0.3 V, Ta = –20 to 75°C

Item	Symbol	Min	Max	Unit	Figure
SIOMCLK clock input cycle time	t_{MCYC}	30	—	ns	32.52
SIOMCLK input high-level width	t_{MWH}	0.4 × t_{MCYC}	—	ns	32.52
SIOMCLK input low-level width	t_{MWL}	0.4 × t_{MCYC}	—	ns	32.52
SCK_SIO clock cycle time	t_{SICYC}	2 × t_{PCYC}	—	ns	32.53 to 32.57
SCK_SIO output high-level width	t_{SWHO}	0.4 × t_{SICYC}	—	ns	32.53 to 32.56
SCK_SIO output low-level width	t_{SWLO}	0.4 × t_{SICYC}	—	ns	32.53 to 32.56
SIOFSYNC output delay time	t_{FSD}	—	20	ns	32.53 to 32.56
SCK_SIO input high-level width	t_{SWHI}	0.4 × t_{SICYC}	—	ns	32.57
SCK_SIO input low-level width	t_{SWLI}	0.4 × t_{SICYC}	—	ns	32.57
SIOFSYNC input setup time	t_{FSS}	20	—	ns	32.57
SIOFSYNC input hold time	t_{FSH}	20	—	ns	32.57
TXD_SIO output delay time	t_{STDD}	—	20	ns	32.53 to 32.57
RXD_SIO input setup time	t_{SRDS}	20	—	ns	32.53 to 32.57
RXD_SIO input hold time	t_{SRDH}	20	—	ns	32.53 to 32.57

Note: t_{PCYC} is the cycle time (ns) of the peripheral clock (Pϕ)

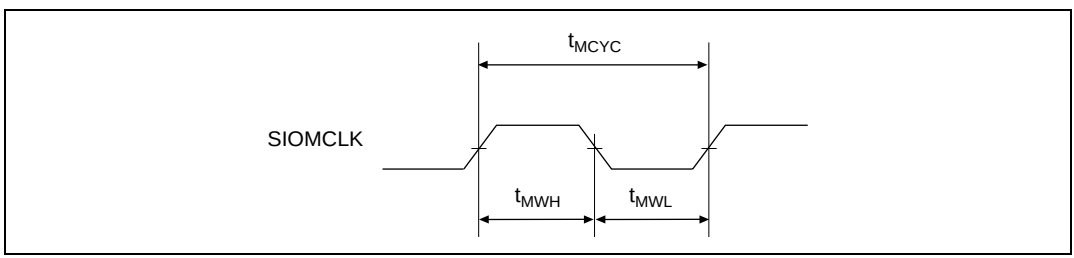


Figure 32.52 SIOMCLK Input Timing

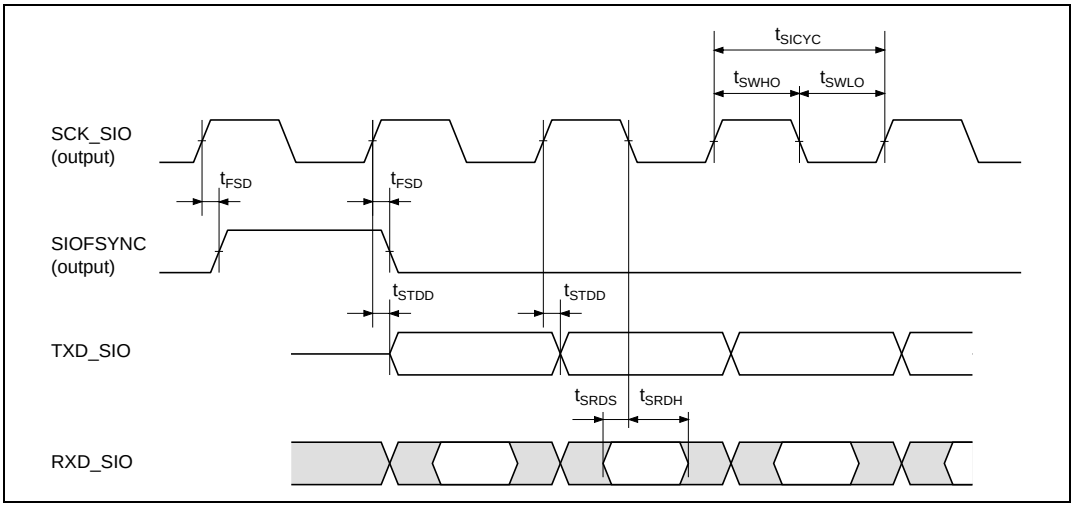


Figure 32.53 SIOF Transmit/Receive Timing (Master Mode 1: Fall Sampling Time)

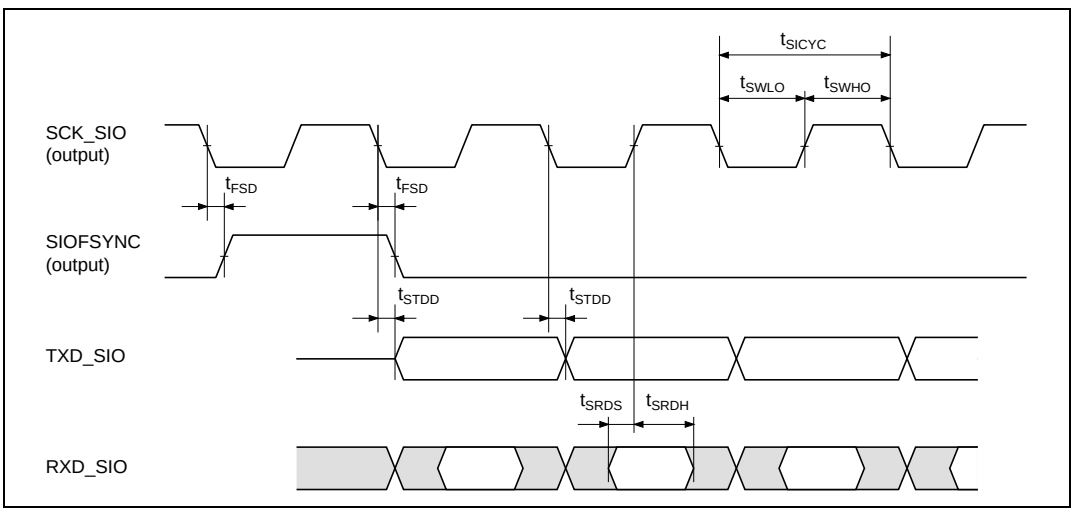


Figure 32.54 SIOF Transmit/Receive Timing (Master Mode 1: Rise Sampling Time)

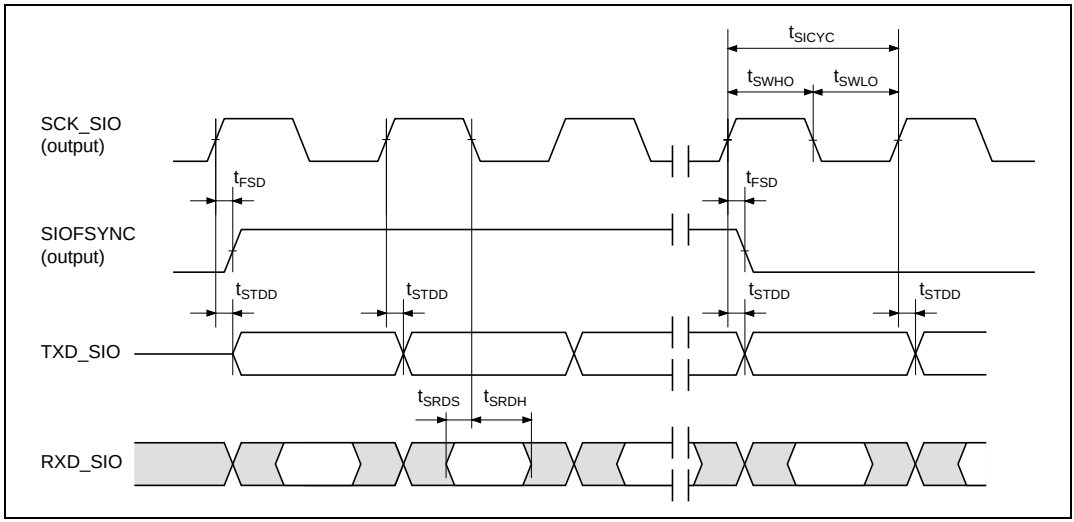


Figure 32.55 SIOF Transmit/Receive Timing (Master Mode 2: Fall Sampling Time)

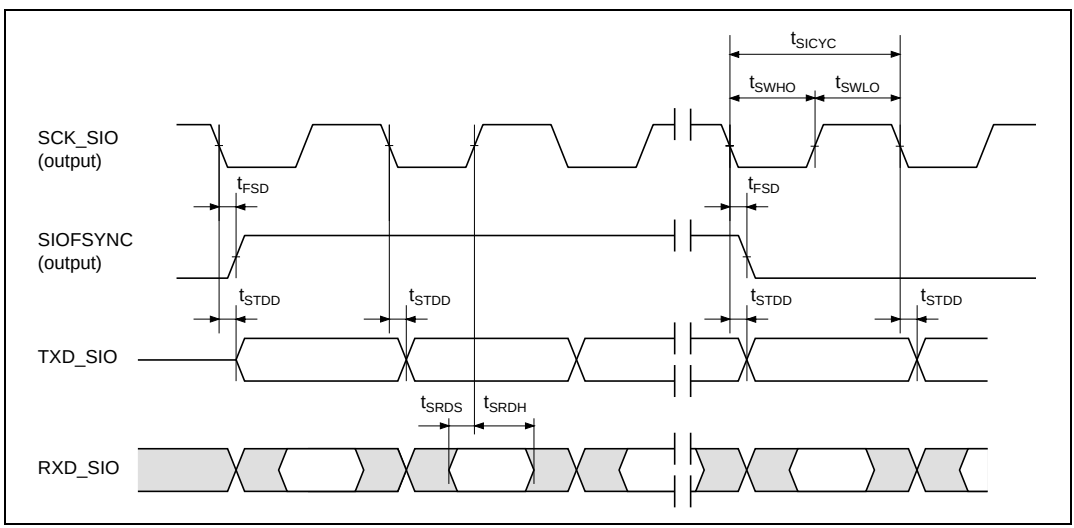


Figure 32.56 SIOF Transmit/Receive Timing (Master Mode 2: Rise Sampling Time)

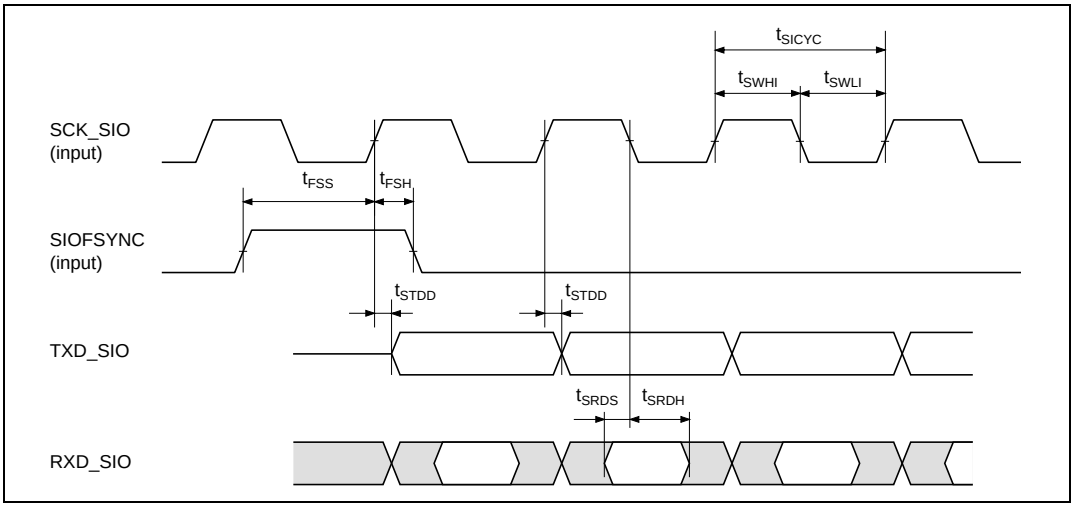


Figure 32.57 SIOF Transmit/Receive Timing (Slave Mode 1 and Slave Mode 2)

32.3.12 USB Module Signal Timing

Table 32.15 USB Module Signal Timing

Conditions: $V_{CC}Q$ = 2.6 to 3.6 V, V_{CC} = 1.60 to 2.05 V, AV_{CC} = 3.3 ± 0.3 V, T_a = –20 to 75°C

Item	Symbol	Min	Max	Unit	Figure
UCLK external input clock frequency (48 MHz)	t_{FREQ}	47.9	48.1	MHz	32.58
Clock rise time	t_{R48}	—	6	ns	
Clock fall time	t_{F48}	—	6	ns	

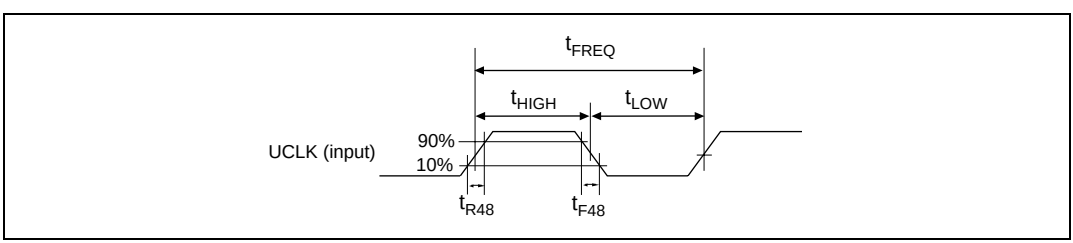


Figure 32.58 USB Clock Timing

Table 32.16 USB Electrical Characteristics (Full-Speed)

Item	Symbol	Min	Max	Unit	State[*1]
Transition time (rise)[*2]	t_R	4	20	ns	CL = 50 pF
Transition time (fall)[*2]	t_F	4	20	ns	CL = 50 pF
Rise/fall time matching	t_{RFM}	90	111	%	(TR/TF)
Output signal crossover power supply voltage	V_{CRS}	1.3	2.0	V	—

Notes: Measured with edge control C_{EDGE} = 47 pF and connection of direct resister Rs = 27 Ω.
1. Value when CL = 50 pF unless specified otherwise.
2. Value within 10% to 90% of the signal power supply voltage.

Table 32.17 USB Electrical Characteristics (Low-Speed)

Item	Symbol	Min	Max	Unit	State
Transition time (rise)*	t_R	75	—	ns	CL = 200 pF
		—	300	ns	CL = 600 pF
Transition time (fall)*	t_F	75	—	ns	CL = 200 pF
		—	300	ns	CL = 600 pF
Rise/fall time matching	t_{RFM}	80	125	%	(TR/TF)
Output signal crossover power supply voltage	V_{CRS}	1.3	2.2	V	—

Notes: Measured with edge control C_{EDGE} = 47 pF and connection of direct resister Rs = 27 Ω.
* Value within 10% to 90% of the signal power supply voltage.

32.3.13 AFEIF Module Signal Timing

Table 32.18 AFEIF Module Signal Timing

Conditions: VccQ = 2.6 to 3.6 V, Vcc = 1.60 to 2.05 V, AVcc = 3.3 ± 0.3 V, Ta = –20 to 75°C

Item	Symbol	Min	Max	Unit
AFE_SCLK clock input cycle time	t_{ASCYC}	$8 \times t_{PCYC}$	—	ns
AFE_SCLK input high-level width	t_{ASWH}	$0.4 \times t_{ASCYC}$	—	ns
AFE_SCLK input low-level width	t_{ASWL}	$0.4 \times t_{ASCYC}$	—	ns
AFE_FS input time	t_{AFSD}	0	50	ns
AFE_TXOUT output delay time	t_{ATDD}	—	t_{PCYC} + 20	ns
AFE_RXIN input setup time	t_{ARDS}	20	—	ns
AFE_RXIN input hold time	t_{ARDH}	$2 \times t_{PCYC}$ + 20	—	ns
AFE_HC1 output delay time	t_{AHCD}	—	$3 \times t_{PCYC}$ + 20	ns
AFE_RLYC output delay time	t_{ARLYD}	—	t_{PCYC} + 20	ns

Note: t_{PCYC} is the cycle time (ns) of the peripheral clock (Pϕ).

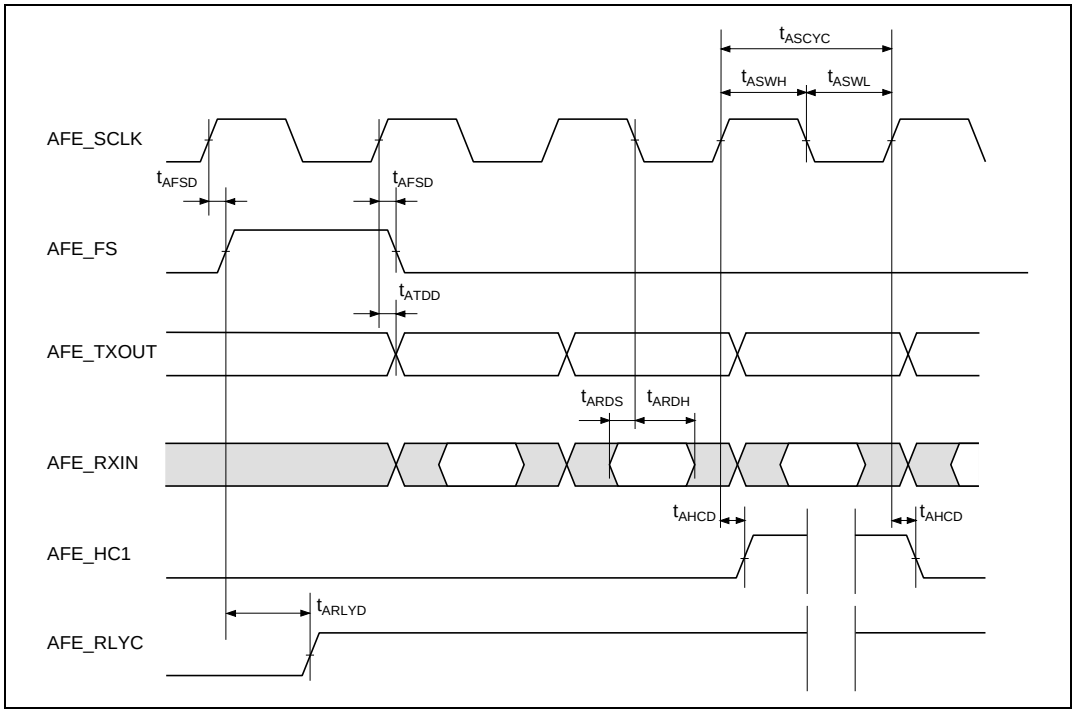
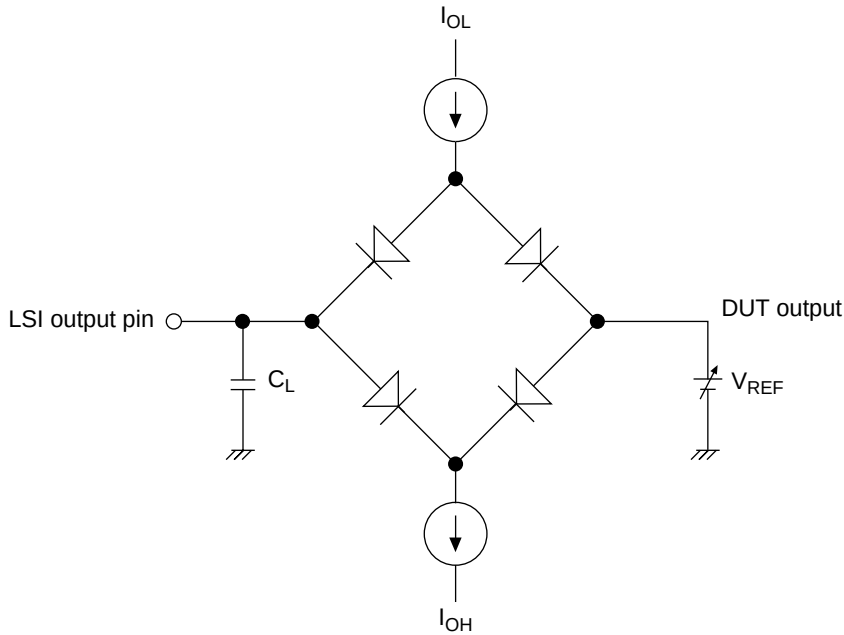


Figure 32.59 AFEIF Module AC Timing

32.3.14 AC Characteristics Measurement Conditions

- I/O signal reference level: 1.2 VccQ
- Input pulse level: VssQ to 3.0 V (where $\overline{\text{RESETP}}$, $\overline{\text{RESETM}}$, $\overline{\text{ASEMD0}}$, $\overline{\text{IRL3}}$ to $\overline{\text{IRL0}}$, $\overline{\text{ADTRG}}$, PINT[15] to PINT[0], CA, NMI, IRQ5 to IRQ0, CKIO, and MD5 to MD0 are within VssQ to VccQ)
- Input rise and fall times: 1 ns



Notes: 1. C_L is the total value that includes the capacitance of measurement instruments, etc., and is set as follows for each pin.
30pF: CKIO, $\overline{\text{RASxx}}$, $\overline{\text{CASxx}}$, $\overline{\text{CS0}}$, $\overline{\text{CS2}}$ to $\overline{\text{CS6}}$, $\overline{\text{CE2A}}$, $\overline{\text{CE2B}}$, $\overline{\text{BACK}}$
50pF: All other pins
2. I_{OL} and I_{OH} are the values shown in table 32.3.

Figure 32.60 Output Load Circuit

32.3.15 Delay Time Variation Due to Load Capacitance

A graph (reference data) of the variation in delay time when a load capacitance greater than that stipulated (30 pF or 50 pF) is connected to this LSI's pins is shown below. The graph shown in figure 32.61 should be taken into consideration in the design process if the stipulated capacitance is exceeded in connecting an external device.

If the connected load capacitance exceeds the range shown in figure 32.61, the graph will not be a straight line.

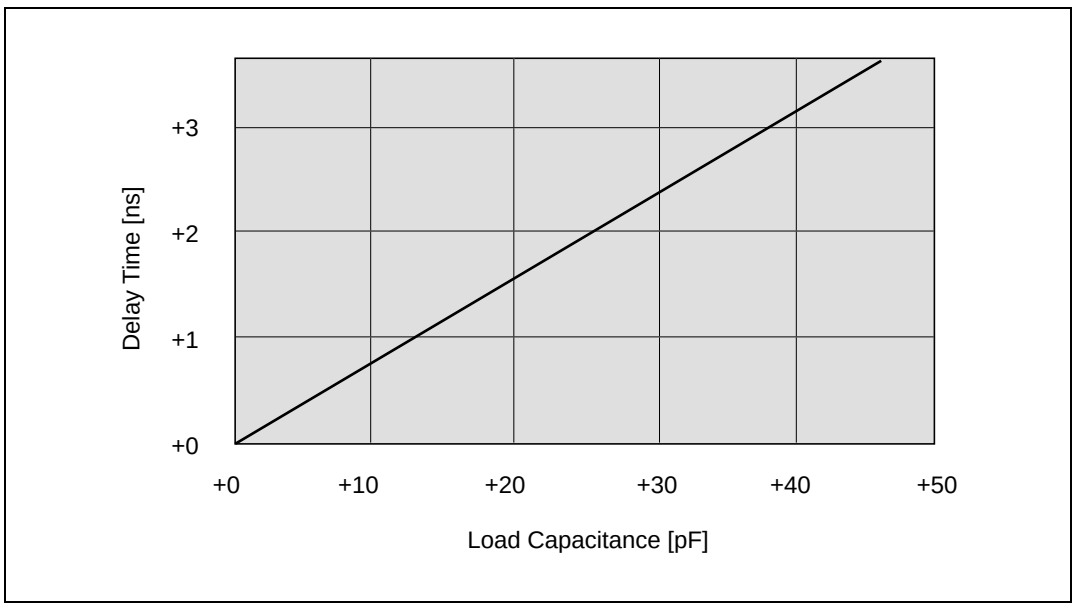


Figure 32.61 Load Capacitance vs. Delay Time

32.4 A/D Converter Characteristics

Table 32.19 lists the A/D converter characteristics.

Table 32.19 A/D Converter Characteristics

Conditions: VccQ = 2.6 to 3.6 V, Vcc = 1.60 to 2.05 V, AVcc = 3.3 ± 0.3 V, Ta = –20 to 75°C

Item	Min	Typ	Max	Unit
Resolution	10	10	10	bits
Conversion time	15	—	—	µs
Analog input capacitance	—	—	20	pF
Permissible signal-source (single-source) impedance	—	—	5	kΩ
Nonlinearity error	—	—	±3.0	LSB
Offset error	—	—	±2.0	LSB
Full-scale error	—	—	±2.0	LSB
Quantization error	—	—	±0.5	LSB
Absolute accuracy	—	—	±4.0	LSB

32.5 D/A Converter Characteristics

Table 32.20 lists the D/A converter characteristics.

Table 32.20 D/A Converter Characteristics

Conditions: VccQ = 2.6 to 3.6 V, Vcc = 1.60 to 2.05 V, AVcc = 3.3 ± 0.3 V, Ta = –20 to 75°C

Item	Min	Typ	Max	Unit	Test Conditions
Resolution	8	8	8	bits	
Conversion time	—	—	10.0	µs	20 pF capacitive load
Absolute accuracy	—	±2.5	±4.0	LSB	2 MΩ resistance load

Appendix A Pin Functions

A.1 Pin Functions

Table A.1 Pin Functions

Type	Signal Name (Initial Status: Bold)	Pin No. (HQFP)	I/O	Power-On Reset	Manual Reset	Standby	Release/ Open Bus Privileges
Clock and oscillation related	**XTAL2**	2	O	O	O	O	O
	EXTAL2	3	I	I	I	I	I
	XTAL	179	O	O	O	O	O
	EXTAL	180	I	I	I	I	I
	CAP1, CAP2	171, 174	—	—	—	—	—
	CKIO2	19	O(Z)	O	O(Z)	O(Z)	O(Z)
	CKIO	189	IO	IO	IO	IO	IO
System control	**STATUS0**/PTJ[6], **STATUS1**/PTJ[7]	185, 186	O/IO	O	O/P	O/K	O/P
	$\overline{\text{RESETP}}$, **$\overline{\text{RESETM}}$**	220, 147	I	I	I	I	I
	CA	221	I	I	I	I	I
	$\overline{\text{Scan_testen}}$	224	I	I	I	I	I
	MD0 to MD5	169, 5, 6, 222, 223, 15	I	I	I	I	I
	$\overline{\text{ASEMD0}}$	150	I	I	I	I	I
Interrupts	**NMI**	7	I	I	I	I	I
	$\overline{\text{IRQ0}}$/**$\overline{\text{IRL0}}$**/PTH[0] to **$\overline{\text{IRQ3}}$**/**$\overline{\text{IRL3}}$**/PTH[3]	8, 9, 10, 11	I/I/I	V	I/I/I	I/I/Z	I/I/I
	$\overline{\text{IRQ4}}$/PTH[4]	12	I/I	V	I/I	I/Z	I/I
Bus functions	**$\overline{\text{BREQ}}$**	16	I	I	I	I	I
	$\overline{\text{BACK}}$	17	O	H	H	H	L
	D31/PTB[7] to **D24**/PTB[0], **D23**/PTA[7] to **D16**/PTA[0]	21 to 28, 30, 32 to 34, 36, 38 to 40	IO/IO	Z	K/P	Z/K	Z/P
	D15 to D0	41, 43, 45 to 52, 54, 56 to 60	IO	Z	K	Z	Z

Type	Signal Name (Initial Status: Bold)	Pin No. (HQFP)	I/O	Power-On Reset	Manual Reset	Standby	Release/ Open Bus Privileges
Bus functions	**A0 to A25**	61 to 63, 65, 67 to 74, 76, 78 to 85, 87, 89, 90, 92, 94	O	Z	O	Z(L)	Z
	$\overline{\text{BS}}$/PTK[4]	95	O/IO	H	O/P	Z(H)/K	Z/P
	$\overline{\text{RD}}$	96	O	H	O	Z(H)	Z
	$\overline{\text{WE0}}$/DQMLL	97	O/O	H	O/O	Z(H)/Z(H)	Z/Z
	$\overline{\text{WE1}}$/DQMLU/$\overline{\text{WE}}$	98	O/O/O	H	O/O/O	Z(H)/Z(H)/ Z(H)	Z/Z/Z
	$\overline{\text{WE2}}$/DQMUL/$\overline{\text{ICIORD}}$/ PTK[6], **$\overline{\text{WE3}}$**/DQMUU/$\overline{\text{ICIOWR}}$/ PTK[7]	99, 101	O/O/O/ IO	H	O/O/O/P	Z(H)/Z(H)/ O/K	Z/Z/Z/P
	RDWR	103	O	H	O	Z(H)	Z
	$\overline{\text{CS0}}$, $\overline{\text{CS2}}$, $\overline{\text{CS3}}$	105, 106, 107	O	H	O	Z(H)	Z
	$\overline{\text{CS4}}$/PTK[2]	108	O/IO	H	O/P	Z(H)/K	Z/P
	$\overline{\text{CS5}}$/$\overline{\text{CE1A}}$/PTK[3]	109	O/O/IO	H	O/O/P	Z(H)/Z(H)/ K	Z/Z/P
	$\overline{\text{CS6}}$/$\overline{\text{CE1B}}$	110	O/O	H	O/O	Z(H)/Z(H)	Z/Z
	$\overline{\text{CE2A}}$/**PTE[4],** $\overline{\text{CE2B}}$/**PTE[5]**	111, 112	O/IO	V	O/P	Z(H)/K	Z/P
	CKE/PTK[5]	128	O/IO	H	O/P	O/K	O/P
	Reserved/$\overline{\text{CAS}}$/PTJ[2]	131	O/O/IO	H	O/O/P	O/Z(H)/K	O/Z(H)/P
	$\overline{\text{WAIT}}$	146	I	Z	Z	Z	Z
	$\overline{\text{IOIS16}}$/**PTG[7]**	149	I/I	V	I/I	Z/Z(V)	I/I
	$\overline{\text{RAS3}}$/PTJ[0], **Reserved**/PTJ[1], **Reserved**/PTJ[3]*3, **Reserved**/PTJ[4]*3, **Reserved**/PTJ[5]*3	129, 130, 133, 135, 136	O/IO	H	O/P	Z(H)/K	Z(H)/P
AFE/USB digital/port related	**AFE_HC1**/ USB1d_DPLS/PTK[0]	113	O/I/IO	L	O/I/P	Z/Z/K	O/I/P
	$\overline{\text{AFE_RLYCNT}}$/ USB1d_DMNS/PTK[1]	114	O/I/IO	L	O/I/P	O/Z/K	O/I/P
	AFE_SCLK/ USB1d_TXDPLS	116	I/O	I	I/O	Z/O	I/O

Type	Signal Name (Initial Status: Bold)	Pin No. (HQFP)	I/O	Power-On Reset	Manual Reset	Standby	Release/ Open Bus Privileges
AFE/USB digital/port related	**PTM[7]/PINT[7]**/ AFE_FS/USB1d_RCV, **PTM[6]/PINT[6]**/ AFE_RXIN/USB1d_SPEED, **PTM[5]/PINT[5]**/ AFE_TXOUT/ USB1d_TXSE0	118, 119, 120	I/I/I/O	V	I/I/I/O	Z(V)/I/Z/O	I/I/I/O
	PTM[4]/PINT[4]/ $\overline{\text{AFE_RDET}}$	121	I/I/I	V	I/I/I	Z(V)/I/Z	I/I/I
	USB1d_SUSPEND	122	O	O	O	O	O
JTAG	**PTE[0]/TDO***1	143	IO/O	V/O*1	P/O	K/O	P/O
	PTF[7]/PINT[15]/$\overline{\text{TRST}}$*1, **PTF[6]/PINT[14]/TMS***1, **PTF[5]/PINT[13]/TDI***1, **PTF[4]/PINT[12]/TCK***1	160, 162, 163, 164	I/I/I	V/I*1	I/I/I	Z/I/I	I/I/I
DMAC	$\overline{\text{DREQ0}}$/**PTD[4]**	218	I/I	V	Z/I	Z/Z(V)	I/I
	DRAK0/PCC0RESET	144	O/O	H	O/O	Z(H)/Z	O/O
	DACK0/PCC0DRV	145	O/O	H	O/O	Z/Z	O/O
Port/PCC/ AUD related	**PTE[7]**/PCC0RDY/ **AUDSYNC***1	104	IO/I/O	V/O*1	P/I/O	K/Z/O	P/I/O
	$\overline{\text{PCC0WAIT}}$/ **PTH[6]/AUDCK***1	176	I/I/I	V/V*1	I/I/I	Z/Z(V)/Z	I/I/I
	PCC0BVD2/ **PTG[3]/AUDATA[3]***1, PCCBVD1/ **PTG[2]/AUDATA[2]***1, $\overline{\text{PCC0CD2}}$/ **PTG[1]/AUDATA[1]***1, $\overline{\text{PCC0CD1}}$/ **PTG[0]/AUDATA[0]***1	153, 154, 156, 158	I/I/O	V/O*1	I/I/O	Z/Z(V)/O	I/I/O
	PTG[4]	152	I	V	I	Z(V)	I
	PTF[3]/PINT[11]	165	I/I	V	I/I	Z(V)/I	I/I
	PCCREG/**PTF[2]**	166	O/I	V	O/I	Z/Z(V)	O/I
	PTG[5]/$\overline{\text{ASEBRKAK}}$*1	151	I/O	V/O*1	I/O	Z(V)/O	I/O
	$\overline{\text{PCC0VS1}}$/**PTF[1]**, $\overline{\text{PCC0VS2}}$/**PTF[0]**	167, 168	I/I	V	I/I	Z/Z(V)	I/I

Type	Signal Name (Initial Status: Bold)	Pin No. (HQFP)	I/O	Power-On Reset	Manual Reset	Standby	Release/ Open Bus Privileges
LCDC related	**PTD[5]**/CL1, **PTD[7]**/DON, **PTE[6]**/M_DISP, **PTE[3]**/FLM, PTH[7]/**CL2**	138, 140, 141, 142, 187	IO/O	V	P/L	K/L*2	P/O
	VEPWC, VCPWC	13, 14	O	L	L	L*2	O
	LCD15/**PTM[3]/PINT[10]** to LCD13/**PTM[1]/PINT[8]**	181, 182, 183	O/I/I	V	L/I/I	K*2/Z(V)/I	O/I/I
	LCD12/**PTM[0]**	184	O/I	V	L/I	K*2/Z(V)	O/I
	LCD11/**PTC[7]/PINT[3]** to LCD8/**PTC[4]/PINT[0]**	204, 205, 206, 208	O/IO/I	V	L/P/I	K*2/K/I	O/P/I
	LCD7/**PTD[3]** to LCD0/**PTD[0]**	210 to 217	O/IO	V	L/P	K*2/K	O/P
	LCLK/UCLK/**PTD[6]**	219	I/I/I	V	I/I/I	Z/Z/Z(V)	I/I/I
Serial related	**SIOMCLK**/SCPT[3]	194	I/IO	I	Z/P	Z/K	I/P
	SCK_SIO/SCPT[5]	196	IO/IO	I	Z/P	Z/K	IO/P
	SIOFSYNC/SCPT[6]	197	IO/IO	I	Z/P	K/K	IO/P
	RxD0/SCPT[0], **RxD2**/SCPT[4]	198, 201	I/I	Z	Z/I	Z/Z	I/Z
	RxD_SIO/SCPT[2]	199	I/I	I	Z/I	Z/Z	I/Z
	SCPT[7]/$\overline{\text{CTS2}}$/$\overline{\text{IRQ5}}$	203	I/I/I	V	I/Z/I	Z(V)/Z/I	I/I/I
	TxD0/SCPT[0], **TxD2**/SCPT[4]	191, 195	O/O	Z	Z/O	Z/K	O/Z
	SCK0/SCPT[1]	192	IO/IO	V	Z/P	V/K	IO/P
	TxD_SIO/SCPT[2]	193	O/O	H	Z/O	Z/K	O/O
	$\overline{\textbf{RTS2}}$/USB1d_TXENL	125	O/O	L	O/O	Z/O	O/O
USB related	**PTE[2]**/USB1_pwr_en, **PTE[1]**/USB2_pwr_en	126, 127	IO/O	V	P/O	K/O	P/O
	$\overline{\textbf{USB1_ovr_crnt}}$/ USBF_VBUS	123	I/I	I	Z/Z	Z/Z	I/I
	$\overline{\textbf{USB2_ovr_crnt}}$	124	I	I	Z	Z	I
	USB1_P(analog), USB1_M(analog), USB2_P(analog), USB2_M(analog)	226, 227, 229, 230	IO	L	L	K	IO

Type	Signal Name (Initial Status: Bold)	Pin No. (HQFP)	I/O	Power-On Reset	Manual Reset	Standby	Release/ Open Bus Privileges
ADC	**$\overline{\text{ADTRG}}$**/PTH[5]	148	I/I	V	I/I	I/Z	I/I
	AN[2]/PTL[2], AN[3]/PTL[3], AN[4]/PTL[4], AN[5]/PTL[5]	233, 234, 235, 236	I/I	Z	Z/I	Z/Z	I/I
ADC/DAC	**AN[6]**/PTL[6]/DA[1], **AN[7]**/PTL[7]/DA[0]	238, 239	I/I/O	Z	Z/I/Z	Z/Z/O	I/I/O
Power supply	**AVcc_USB**	225, 231	—	—	—	—	—
	AVss_USB	228	—	—	—	—	—
	AVss	232, 240	—	—	—	—	—
	AVcc	237	—	—	—	—	—
	Vcc-PLL1, Vcc-PLL2	170, 175	—	—	—	—	—
	Vss-PLL1, Vss-PLL2	172, 173	—	—	—	—	—
	Vcc-RTC	1	—	—	—	—	—
	Vss-RTC	4	—	—	—	—	—
	Vcc	37, 93, 139, 157, 178, 202	—	—	—	—	—
	Vss	35, 91, 137, 155, 177, 200	—	—	—	—	—
	VccQ	20, 31, 44, 55, 66, 77, 88, 102, 117, 134, 161, 190, 209	—	—	—	—	—
	VssQ	18, 29, 42, 53, 64, 75, 86, 100, 115, 132, 159, 188, 207	—	—	—	—	—

Notes: 1. The initial status is determined from the level of $\overline{\text{ASEMD0}}$ when resetting.

2. In accordance with the power management sequence described in section 25.3.6, Power Management Registers of this manual, make sure to clear the DON register to 0 and interrupt the power supply before entering the standby mode.

3. The initial status of PTJ3 to PTJ5 is high level output, but it may change briefly to low level when resetting.

I: Input (input buffer on)
O: Output (output buffer on)
H: High-level output
L: Low-level output
Z: High-impedance (input and output buffers off)
P: I or O according to setting of register
K: Input pins are high-impedance, output pin status is retained
V: Input/output buffer off, pull-up MOS on
( ): Based on status of internal registers (Refer to the register tables for information on individual pins.)

A.2 Treatment of Unused Pins

Table A.2 Treatment of Unused Pins

Type	Signal Name (Initial Status: Bold)	Pin No (HQFP)	Pin No (CSP)	I/O	Treatment when Not Used
Clock and oscillation related	**XTAL2**	2	B4	O	Open
	EXTAL2	3	A2	I	Connect to Vcc
	XTAL	179	V4	O	Open
	EXTAL	180	V2	I	Pull up
	CAP1, CAP2	171, 174	U3, W4	—	Open
	CKIO2	19	C7	O	Open
	CKIO	189	R1	IO	Open
System control	**STATUS0**/PTJ[6], **STATUS1**/PTJ[7]	185, 186	U1, R2	O/IO	Open
	$\overline{\text{RESETP}}$, **$\overline{\text{RESETM}}$**	220, 147	H1, T11	I	Pull up
	CA	221	G4	I	Pull up
	$\overline{\text{Scan_testen}}$	224	G1	I	Pull up
	MD0 to MD5	169, 5, 6, 222, 223, 15	T5, A3, B5, G3, G2, C6	I	Always used
	$\overline{\text{ASEMD0}}$	150	T10	I	Pull up
Interrupts	**NMI**	7	A4	I	Pull up
	$\overline{\text{IRQ0}}$/**$\overline{\text{IRL0}}$**/PTH[0] to **$\overline{\text{IRQ3}}$**/**$\overline{\text{IRL3}}$**/PTH[3]	8, 9, 10, 11	C4, A5, D4, C5	I/I/I	Pull up
	$\overline{\text{IRQ4}}$/PTH[4]	12	D5	I/I	Pull up
Bus functions	**$\overline{\text{BREQ}}$**	16	D6	I	Pull up
	$\overline{\text{BACK}}$	17	A7	O	Open
	D31/PTB[7] to **D24**/PTB[0], **D23**/PTA[7] to **D16**/PTA[0]	21 to 28, 30, 32 to 34, 36, 38 to 40	A8, B8, C8, D8, A9, B9, D9, C9, D10, B10, C11, D11, A11, C12, B12, A12	IO/IO	Open
	D15 to D0	41, 43, 45 to 52, 54, 56 to 60	D13, D11, D14, C14, B14, A14, D15, C15, C17, A15, A16, A17, B17, A18, B16, B18	IO	Open

Type	Signal Name (Initial Status: Bold)	Pin No (HQFP)	Pin No (CSP)	I/O	Treatment when Not Used
Bus functions	**A0 to A25**	61 to 63, 65, 67 to 74, 76, 78 to 85, 87, 89, 90, 92, 94	A19, D18, B19, C19, D19, D17, E19, D16, E17, E16, F19, F18, F16, G18, G17, G16, H19, H18, H17, H16, J19, J16, K19, K16, K18,L16	O	Open
	$\overline{\text{BS}}$/PTK[4]	95	L18	O/IO	Open
	RD/$\overline{\text{WR}}$	96	L19	O	Open
	$\overline{\text{WE0}}$/DQMLL	97	M16	O/O	Open
	$\overline{\text{WE1}}$/DQMLU/$\overline{\text{WE}}$	98	M17	O/O/O	Open
	$\overline{\text{WE2}}$/DQMUL/$\overline{\text{ICIORD}}$/PTK[6], **$\overline{\text{WE3}}$**/DQMUU/$\overline{\text{ICIOWR}}$/PTK[7]	99, 101	M18	O/O/O/IO	Open
	RD/$\overline{\text{WR}}$	103	N18	O	Open
	$\overline{\text{CS0}}$, $\overline{\text{CS2}}$, $\overline{\text{CS3}}$	105, 106, 107	P16, P17, P18	O	Open
	$\overline{\text{CS4}}$/PTK[2]	108	P19	O/IO	Open
	$\overline{\text{CS5}}$/$\overline{\text{CE1A}}$/PTK[3]	109	R16	O/O/IO	Open
	$\overline{\text{CS6}}$/$\overline{\text{CE1B}}$	110	R17	O/O	Open
	$\overline{\text{CE2A}}$/PTE[4], $\overline{\text{CE2B}}$/PTE[5]	111, 112	U17, R19	O/IO	Open
	CKE/PTK[5]	128	U16	O/IO	Open
	Reserved/$\overline{\text{CAS}}$/PTJ[2]	131	U15	O/O/IO	Open
	$\overline{\text{WAIT}}$	146	V11	I	Pull up
	$\overline{\text{IOIS16}}$/PTG[7]	149	W10	I/I	Open
	$\overline{\text{RAS3}}$/PTJ[0], **Reserved**/PTJ[1], **Reserved**/PTJ[3], **Reserved**/PTJ[4], **Reserved**/PTJ[5]	129, 130, 133, 135, 136	W15, T16, W14, U14, T14	O/IO	Open

Type	Signal Name (Initial Status: Bold)	Pin No (HQFP)	Pin No (CSP)	I/O	Treatment when Not Used
AFE/USB digital/port related	**AFE_HC1**/USB1d_DPLS/ PTK[0], **$\overline{\text{AFE_RLYCNT}}$**/USB1d_DMNS/ PTK[1]	113, 114	T17, T19	O/I/IO	Open
	AFE_SCLK/USB1d_TXDPLS	116	U19	I/IO	Pull up
	PTM[7]/PINT[7]/AFE_FS/ USB1d_RCV, **PTM[6]/PINT[6]**/AFE_RXIN/ USB1d_SPEED, **PTM[5]/PINT[5]**/AFE_TXOUT/ USB1d_TXSE0	118, 119, 120	V19, T18, V18	I/I/I/O	Open
	PTM[4]/PINT[4]/$\overline{\text{AFE_RDET}}$	121	W19	I/I/I	Open
	Reserved/USB1d_SUSPEND	122	V16	O	Open
JTAG	**PTE[0]/TDO***1	143	U12	IO/O	Open
	PTF[7]/PINT[15]/$\overline{\text{TRST}}$*1, **PTF[6]/PINT[14]/TMS***1, **PTF[5]/PINT[13]/TDI***1, **PTF[4]/PINT[12]/TCK***1	160, 162, 163, 164	W8, U7, V7, W7	I/I/I	Pull up
DMAC	**$\overline{\text{DREQ0}}$**/PTD[4]	218	H3	I/I	Pull up
	DRAK0/PCC0RESET, **DACK0**/PCC0DRV	144, 145	T12, W11	O/O	Open
Port/PCC/ AUD related	**PTE[7]**/PCC0RDY/**AUDSYNC**	104	N19	IO/I/O	Open
	$\overline{\text{PCC0WAIT}}$/**PTH[6]/AUDCK***1	176	W3	I/I/I	Pull up
	PCC0BVD2/**PTG[3]/ AUDATA[3]***1, PCCBVD1/**PTG[2]/ AUDATA[2]***1, $\overline{\text{PCC0CD2}}$/**PTG[1]/ AUDATA[1]***1, $\overline{\text{PCC0CD1}}$/**PTG[0]/ AUDATA[0]***1	153, 154, 156, 158	U9, T9, W9, U8	I/I/O	Open
	PTG[4]	152	V10	I	Open
	PTF[3]/PINT[11]	165	T6	I/I	Open
	PCCREG/**PTF[2]**	166	U6	O/I	Open
	PTG[5]/$\overline{\text{ASEBRKAK}}$*1	151	U10	I/O	Open
	$\overline{\text{PCC0VS1}}$/**PTF[1]**, $\overline{\text{PCC0VS2}}$/**PTF[0]**	167, 168	V6, W6	I/I	Open

Type	Signal Name (Initial Status: Bold)	Pin No (HQFP)	Pin No (CSP)	I/O	Treatment when Not Used
LCDC related	**PTD[5]**/CL1, **PTD[7]**/DON, **PTE[6]**/M_DISP, **PTE[3]**/FLM, **CL2**/PTH[7]	138, 140, 141, 142, 187	V13, T13, W12, V12, T1	IO/O	Open
	VEPWC, VCPWC	13, 14	A6, B6	O	Open
	LCD15/**PTM[3]/PINT[10]** to LCD13/**PTM[1]/PINT[8]**	181, 182, 183	W1, T2, V1	O/I/I	Open
	LCD12/**PTM[0]**	184	U2	O/I	Open
	LCD11/**PTC[7]/PINT[3]** to LCD8/**PTC[4]/PINT[0]**	204, 205, 206, 208	M4, L1, L2, L3	O/IO/I	Open
	LCD7/**PTD[3]** to LCD0/**PTD[0]**	210 to 217	K4, K3, K2, J3, J4, J2, J1, H4	O/IO	Open
	LCLK/UCLK/**PTD[6]**	219	H2	I/I/I	Pull up
Serial related	**SIOMCLK**/SCPT[3]	194	P2	I/IO	Pull up
	SCK_SIO/SCPT[5], **SIOFSYNC**/SCPT[6]	196, 197	P4, N1	IO/IO	Pull up
	RxD0/SCPT[0], **RxD2**/SCPT[4], **RxD_SIO**/SCPT[2]	198, 201, 199	N2, M1, N3	I/I	Pull up
	SCPT[7]/$\overline{\text{CTS2}}$/$\overline{\text{IRQ5}}$	203	M3	I/I/I	Pull up
	TxD0/SCPT[0], **TxD2**/SCPT[4]	191, 195, 193	R3, P3, P1	O/O	Open
	SCK0/SCPT[1]	192	R4	IO/IO	Open
	TxD_SIO/SCPT[2]	193	P1	O/O	Open
	$\overline{\textbf{RTS2}}$/USB1d_TXENL	125	W17	O/O	Open
USB related	**PTE[2]**/USB1_pwr_en, **PTE[1]**/USB2_pwr_en	126, 127	V15, W16	IO/O	Open
	$\overline{\textbf{USB1_ovr_crnt}}$/USBF_VBUS	123	W18	I/I	Pull up
	$\overline{\textbf{USB2_ovr_crnt}}$	124	V17	I	Pull up
	USB1_P(analog), USB1_M(analog), USB2_P(analog), USB2_M(analog)	226, 227, 229, 230	F3, F2, E4, E3	IO	Open*2
ADC	$\overline{\textbf{ADTRG}}$/PTH[5]	148	U11	I/I	Pull up
	AN[2]/PTL[2], AN[3]/PTL[3], AN[4]/PTL[4], AN[5]/PTL[5]	233, 234, 235, 236	D3, D1, E2, C1	I/I	Open
ADC/DAC	**AN[6]**/PTL[6]/DA[1], **AN[7]**/PTL[7]/DA[0]	238, 239	B1, D2	I/I/O	Open*3

Type	Signal Name (Initial Status: Bold)	Pin No (HQFP)	Pin No (CSP)	I/O	Treatment when Not Used
Power supply	**AVcc_USB**	225, 231	F4, C3	—	VccQ
	AVss_USB	228	F1	—	VssQ
	AVss	232, 240	E1, B2	—	VssQ
	AVcc	237	C2	—	VccQ
	Vcc-PLL1, Vcc-PLL2	170, 175	U5, V5	—	Vcc
	Vss-PLL1, Vss-PLL2	172, 173	W5, U4	—	Vcc
	Vcc-RTC	1	A1	—	Vcc
	Vss-RTC	4	B3	—	Vcc
	Vcc	37, 93, 139, 157, 178, 202	D2, L17, U13, T8, W2, M2	—	Vcc
	Vss	35, 91, 137, 155, 177, 200	B11, K17, W13, V9, V3, N4	—	Vss
	VccQ	20, 31, 44, 55, 66, 77, 88, 102, 117, 134, 161, 190, 209	D7, C10, A13, B15, E18, G19, J17, N17, U18, V14, T7, T4, K1	—	VccQ
	VssQ	18, 29, 42, 53, 64, 75, 86, 100, 115, 132, 159, 188, 207	B7, A10, C13, C16, C18, F17, J18, M19, R18, T15, V8, T3, L4	—	VssQ

Notes: For unused pins, the above table shows examples of processing. The indicated settings may not be suitable in some cases.

1. Valid when $\overline{\text{ASEMD0}}$ is low level.
2. In cases where either USB function is used, this should be fixed as per the not used and not active examples.
3. A/D pin functions are assumed.

A.3 Pin Status when Accessing Address Spaces

Table A.3 Pin Status (Normal Memory/Little Endian)

Pin		8-Bit Bus Width	16-Bit Bus Width		
		Byte/Word/ Longword Access	Byte Access (Address 2n)	Byte Access (Address 2n+1)	Word/Longword Access
$\overline{CS6}$ to $\overline{CS2}$, $\overline{CS0}$		Enabled	Enabled	Enabled	Enabled
$\overline{RD}$	R	Low	Low	Low	Low
	W	High	High	High	High
RD/$\overline{WR}$	R	High	High	High	High
	W	Low	Low	Low	Low
$\overline{BS}$		Enabled	Enabled	Enabled	Enabled
$\overline{RAS3}$		High	High	High	High
$\overline{CAS}$		High	High	High	High
$\overline{WE0}$/DQMLL	R	High	High	High	High
	W	Low	Low	High	Low
$\overline{WE1}$/DQMLU/$\overline{WE}$	R	High	High	High	High
	W	High	High	Low	Low
$\overline{WE2}$/DQMUL/$\overline{ICIORD}$	R	High	High	High	High
	W	High	High	High	High
$\overline{WE3}$/DQMUU/$\overline{ICIOWR}$	R	High	High	High	High
	W	High	High	High	High
$\overline{CE2A}$		High	High	High	High
$\overline{CE2B}$		High	High	High	High
CKE		Disabled	Disabled	Disabled	Disabled
$\overline{WAIT}$		Enabled*1	Enabled*1	Enabled*1	Enabled*1
$\overline{IOIS16}$		Disabled	Disabled	Disabled	Disabled
A25 to A0		Address	Address	Address	Address
D7 to D0		Valid data	Valid data	Invalid data	Valid data
D15 to D8		High-Z*2	Invalid data	Valid data	Valid data
D31 to D16		High-Z*2	High-Z*2	High-Z*2	High-Z*2

Table A.3 Pin Status (Normal Memory/Little Endian) (cont)

Pin		32-Bit Bus Width						
		Byte Access (Address 4n)	Byte Access (Address 4n+1)	Byte Access (Address 4n+2)	Byte Access (Address 4n+3)	Word Access (Address 4n)	Word Access (Address 4n+2)	Longword Access
$\overline{CS6}$ to $\overline{CS2}$, $\overline{CS0}$		Enabled	Enabled	Enabled	Enabled	Enabled	Enabled	Enabled
$\overline{RD}$	R	Low	Low	Low	Low	Low	Low	Low
	W	High	High	High	High	High	High	High
RD/$\overline{WR}$	R	High	High	High	High	High	High	High
	W	Low	Low	Low	Low	Low	Low	Low
$\overline{BS}$		Enabled	Enabled	Enabled	Enabled	Enabled	Enabled	Enabled
$\overline{RAS3}$		High	High	High	High	High	High	High
$\overline{CAS}$		High	High	High	High	High	High	High
$\overline{WE0}$/DQMLL	R	High	High	High	High	High	High	High
	W	Low	High	High	High	Low	High	Low
$\overline{WE1}$/DQMLU/$\overline{WE}$	R	High	High	High	High	High	High	High
	W	High	Low	High	High	Low	High	Low
$\overline{WE2}$/DQMUL/$\overline{ICIORD}$	R	High	High	High	High	High	High	High
	W	High	High	Low	High	High	Low	Low
$\overline{WE3}$/DQMUU/$\overline{ICIOWR}$	R	High	High	High	High	High	High	High
	W	High	High	High	Low	High	Low	Low
$\overline{CE2A}$		High	High	High	High	High	High	High
$\overline{CE2B}$		High	High	High	High	High	High	High
CKE		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
$\overline{WAIT}$		Enabled*1	Enabled*1	Enabled*1	Enabled*1	Enabled*1	Enabled*1	Enabled*1
$\overline{IOIS16}$		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
A25 to A0		Address	Address	Address	Address	Address	Address	Address
D7 to D0		Valid data	Invalid data	Invalid data	Invalid data	Valid data	Invalid data	Valid data
D15 to D8		Invalid data	Valid data	Invalid data	Invalid data	Valid data	Invalid data	Valid data
D23 to D16		Invalid data	Invalid data	Valid data	Invalid data	Invalid data	Valid data	Valid data
D31 to D24		Invalid data	Invalid data	Invalid data	Valid data	Invalid data	Valid data	Valid data

Notes: 1. Disabled when the wait setting of the WCR2 register is 0.

2. Unused pins can be switched to port function, pull-up.

Table A.4 Pin Status (Normal Memory/Big Endian)

Pin		8-Bit Bus Width	16-Bit Bus Width		
		Byte/Word/ Longword Access	Byte Access (Address 2n)	Byte Access (Address 2n+1)	Word/Longword Access
$\overline{CS6}$ to $\overline{CS2}$, $\overline{CS0}$		Enabled	Enabled	Enabled	Enabled
$\overline{RD}$	R	Low	Low	Low	Low
	W	High	High	High	High
RD/$\overline{WR}$	R	High	High	High	High
	W	Low	Low	Low	Low
$\overline{BS}$		Enabled	Enabled	Enabled	Enabled
$\overline{RAS3}$		High	High	High	High
$\overline{CAS}$		High	High	High	High
$\overline{WE0}$/DQMLL	R	High	High	High	High
	W	Low	High	Low	Low
$\overline{WE1}$/DQMLU/$\overline{WE}$	R	High	High	High	High
	W	High	Low	High	Low
$\overline{WE2}$/DQMUL/$\overline{ICIORD}$	R	High	High	High	High
	W	High	High	High	High
$\overline{WE3}$/DQMUU/$\overline{ICIOWR}$	R	High	High	High	High
	W	High	High	High	High
$\overline{CE2A}$		High	High	High	High
$\overline{CE2B}$		High	High	High	High
CKE		Disabled	Disabled	Disabled	Disabled
$\overline{WAIT}$		Enabled*1	Enabled*1	Enabled*1	Enabled*1
$\overline{IOIS16}$		Disabled	Disabled	Disabled	Disabled
A25 to A0		Address	Address	Address	Address
D7 to D0		Valid data	Invalid data	Valid data	Valid data
D15 to D8		High-Z*2	Valid data	Invalid data	Valid data
D31 to D16		High-Z*2	High-Z*2	High-Z*2	High-Z*2

Table A.4 Pin Status (Normal Memory/Big Endian) (cont)

Pin		32-Bit Bus Width						
		Byte Access (Address 4n)	Byte Access (Address 4n+1)	Byte Access (Address 4n+2)	Byte Access (Address 4n+3)	Word Access (Address 4n)	Word Access (Address 4n+2)	Longword Access
$\overline{CS6}$ to $\overline{CS2}$, $\overline{CS0}$		Enabled	Enabled	Enabled	Enabled	Enabled	Enabled	Enabled
$\overline{RD}$	R	Low	Low	Low	Low	Low	Low	Low
	W	High	High	High	High	High	High	High
RD/$\overline{WR}$	R	High	High	High	High	High	High	High
	W	Low	Low	Low	Low	Low	Low	Low
$\overline{BS}$		Enabled	Enabled	Enabled	Enabled	Enabled	Enabled	Enabled
$\overline{RAS3}$		High	High	High	High	High	High	High
$\overline{CAS}$		High	High	High	High	High	High	High
$\overline{WE0}$/DQMLL	R	High	High	High	High	High	High	High
	W	High	High	High	Low	High	Low	Low
$\overline{WE1}$/DQMLU/$\overline{WE}$	R	High	High	High	High	High	High	High
	W	High	High	Low	High	High	Low	Low
$\overline{WE2}$/DQMUL/$\overline{ICIORD}$	R	High	High	High	High	High	High	High
	W	High	Low	High	High	Low	High	Low
$\overline{WE3}$/DQMUU/$\overline{ICIOWR}$	R	High	High	High	High	High	High	High
	W	Low	High	High	High	Low	High	Low
$\overline{CE2A}$		High	High	High	High	High	High	High
$\overline{CE2B}$		High	High	High	High	High	High	High
CKE		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
$\overline{WAIT}$		Enabled*1	Enabled*1	Enabled*1	Enabled*1	Enabled*1	Enabled*1	Enabled*1
$\overline{IOIS16}$		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
A25 to A0		Address	Address	Address	Address	Address	Address	Address
D7 to D0		Invalid data	Invalid data	Invalid data	Valid data	Invalid data	Valid data	Valid data
D15 to D8		Invalid data	Invalid data	Valid data	Invalid data	Invalid data	Valid data	Valid data
D23 to D16		Invalid data	Valid data	Invalid data	Invalid data	Valid data	Invalid data	Valid data
D31 to D24		Valid data	Invalid data	Invalid data	Invalid data	Valid data	Invalid data	Valid data

Notes: 1. Disabled when the wait setting of the WCR2 register is 0.

2. Unused pins can be switched to port function, pull-up.

Table A.5 Pin Status (Burst ROM/Little Endian)

Pin		8-Bit Bus Width	16-Bit Bus Width		
		Byte/Word/ Longword Access	Byte Access (Address 2n)	Byte Access (Address 2n+1)	Word/Longword Access
$\overline{CS6}$ to $\overline{CS2}$, $\overline{CS0}$		Enabled	Enabled	Enabled	Enabled
$\overline{RD}$	R	Low	Low	Low	Low
	W	—	—	—	—
RD/$\overline{WR}$	R	High	High	High	High
	W	—	—	—	—
$\overline{BS}$		Enabled	Enabled	Enabled	Enabled
$\overline{RAS3}$		High	High	High	High
$\overline{CAS}$		High	High	High	High
$\overline{WE0}$/DQMLL	R	High	High	High	High
	W	—	—	—	—
$\overline{WE1}$/DQMLU/$\overline{WE}$	R	High	High	High	High
	W	—	—	—	—
$\overline{WE2}$/DQMUL/$\overline{ICIORD}$	R	High	High	High	High
	W	—	—	—	—
$\overline{WE3}$/DQMUU/$\overline{ICIOWR}$	R	High	High	High	High
	W	—	—	—	—
$\overline{CE2A}$		High	High	High	High
$\overline{CE2B}$		High	High	High	High
CKE		Disabled	Disabled	Disabled	Disabled
$\overline{WAIT}$		Enabled*1	Enabled*1	Enabled*1	Enabled*1
$\overline{IOIS16}$		Disabled	Disabled	Disabled	Disabled
A25 to A0		Address	Address	Address	Address
D7 to D0		Valid data	Valid data	Invalid data	Valid data
D15 to D8		High-Z*2	Invalid data	Valid data	Valid data
D31 to D16		High-Z*2	High-Z*2	High-Z*2	High-Z*2

Table A.5 Pin Status (Burst ROM/Little Endian) (cont)

Pin		32-Bit Bus Width						
		Byte Access (Address 4n)	Byte Access (Address 4n+1)	Byte Access (Address 4n+2)	Byte Access (Address 4n+3)	Word Access (Address 4n)	Word Access (Address 4n+2)	Longword Access
$\overline{CS6}$ to $\overline{CS2}$, $\overline{CS0}$		Enabled	Enabled	Enabled	Enabled	Enabled	Enabled	Enabled
$\overline{RD}$	R	Low	Low	Low	Low	Low	Low	Low
	W	—	—	—	—	—	—	—
RD/$\overline{WR}$	R	High	High	High	High	High	High	High
	W	—	—	—	—	—	—	—
$\overline{BS}$		Enabled	Enabled	Enabled	Enabled	Enabled	Enabled	Enabled
$\overline{RAS3}$		High	High	High	High	High	High	High
$\overline{CAS}$		High	High	High	High	High	High	High
$\overline{WE0}$/DQMLL	R	High	High	High	High	High	High	High
	W	—	—	—	—	—	—	—
$\overline{WE1}$/DQMLU/$\overline{WE}$	R	High	High	High	High	High	High	High
	W	—	—	—	—	—	—	—
$\overline{WE2}$/DQMUL/$\overline{ICIORD}$	R	High	High	High	High	High	High	High
	W	—	—	—	—	—	—	—
$\overline{WE3}$/DQMUU/$\overline{ICIOWR}$	R	High	High	High	High	High	High	High
	W	—	—	—	—	—	—	—
$\overline{CE2A}$		High	High	High	High	High	High	High
$\overline{CE2B}$		High	High	High	High	High	High	High
CKE		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
$\overline{WAIT}$		Enabled*1	Enabled*1	Enabled*1	Enabled*1	Enabled*1	Enabled*1	Enabled*1
$\overline{IOIS16}$		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
A25 to A0		Address	Address	Address	Address	Address	Address	Address
D7 to D0		Valid data	Invalid data	Invalid data	Invalid data	Valid data	Invalid data	Valid data
D15 to D8		Invalid data	Valid data	Invalid data	Invalid data	Valid data	Invalid data	Valid data
D23 to D16		Invalid data	Invalid data	Valid data	Invalid data	Invalid data	Valid data	Valid data
D31 to D24		Invalid data	Invalid data	Invalid data	Valid data	Invalid data	Valid data	Valid data

Notes: 1. Disabled when the wait setting of the WCR2 register is 0.

2. Unused pins can be switched to port function, pull-up.

Table A.6 Pin Status (Burst ROM/Big Endian)

Pin		8-Bit Bus Width	16-Bit Bus Width		
		Byte/Word/ Longword Access	Byte Access (Address 2n)	Byte Access (Address 2n+1)	Word/Longword Access
$\overline{CS6}$ to $\overline{CS2}$, $\overline{CS0}$		Enabled	Enabled	Enabled	Enabled
$\overline{RD}$	R	Low	Low	Low	Low
	W	—	—	—	—
RD/$\overline{WR}$	R	High	High	High	High
	W	—	—	—	—
$\overline{BS}$		Enabled	Enabled	Enabled	Enabled
$\overline{RAS3L}$		High	High	High	High
$\overline{CAS}$		High	High	High	High
$\overline{WE0}$/DQMLL	R	High	High	High	High
	W	—	—	—	—
$\overline{WE1}$/DQMLU/$\overline{WE}$	R	High	High	High	High
	W	—	—	—	—
$\overline{WE2}$/DQMUL/$\overline{ICIORD}$	R	High	High	High	High
	W	—	—	—	—
$\overline{WE3}$/DQMUU/$\overline{ICIOWR}$	R	High	High	High	High
	W	—	—	—	—
$\overline{CE2A}$		High	High	High	High
$\overline{CE2B}$		High	High	High	High
CKE		Disabled	Disabled	Disabled	Disabled
$\overline{WAIT}$		Enabled*1	Enabled*1	Enabled*1	Enabled*1
$\overline{IOIS16}$		Disabled	Disabled	Disabled	Disabled
A25 to A0		Address	Address	Address	Address
D7 to D0		Valid data	Invalid data	Valid data	Valid data
D15 to D8		High-Z*2	Valid data	Invalid data	Valid data
D31 to D16		High-Z*2	High-Z*2	High-Z*2	High-Z*2

Table A.6 Pin Status (Burst ROM/Big Endian) (cont)

Pin		32-Bit Bus Width						
		Byte Access (Address 4n)	Byte Access (Address 4n+1)	Byte Access (Address 4n+2)	Byte Access (Address 4n+3)	Word Access (Address 4n)	Word Access (Address 4n+2)	Longword Access
$\overline{CS6}$ to $\overline{CS2}$, $\overline{CS0}$		Enabled	Enabled	Enabled	Enabled	Enabled	Enabled	Enabled
$\overline{RD}$	R	Low	Low	Low	Low	Low	Low	Low
	W	—	—	—	—	—	—	—
RD/$\overline{WR}$	R	High	High	High	High	High	High	High
	W	—	—	—	—	—	—	—
$\overline{BS}$		Enabled	Enabled	Enabled	Enabled	Enabled	Enabled	Enabled
$\overline{RAS3}$		High	High	High	High	High	High	High
$\overline{CAS}$		High	High	High	High	High	High	High
$\overline{WE0}$/DQMLL	R	High	High	High	High	High	High	High
	W	—	—	—	—	—	—	—
$\overline{WE1}$/DQMLU/$\overline{WE}$	R	High	High	High	High	High	High	High
	W	—	—	—	—	—	—	—
$\overline{WE2}$/DQMUL/$\overline{ICIORD}$	R	High	High	High	High	High	High	High
	W	—	—	—	—	—	—	—
$\overline{WE3}$/DQMUU/$\overline{ICIOWR}$	R	High	High	High	High	High	High	High
	W	—	—	—	—	—	—	—
$\overline{CE2A}$		High	High	High	High	High	High	High
$\overline{CE2B}$		High	High	High	High	High	High	High
CKE		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
$\overline{WAIT}$		Enabled*1	Enabled*1	Enabled*1	Enabled*1	Enabled*1	Enabled*1	Enabled*1
$\overline{IOIS16}$		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
A25 to A0		Address	Address	Address	Address	Address	Address	Address
D7 to D0		Invalid data	Invalid data	Invalid data	Valid data	Invalid data	Valid data	Invalid data
D15 to D8		Invalid data	Invalid data	Valid data	Invalid data	Invalid data	Valid data	Valid data
D23 to D16		Invalid data	Valid data	Invalid data	Invalid data	Valid data	Invalid data	Valid data
D31 to D24		Valid data	Invalid data	Invalid data	Invalid data	Valid data	Invalid data	Valid data

Notes: 1. Disabled when the wait setting of the WCR2 register is 0.

2. Unused pins can be switched to port function, pull-up.

Table A.7 Pin Status (Synchronous DRAM/Little Endian)

Pin		32-Bit Bus Width						
		Byte Access (Address 4n)	Byte Access (Address 4n+1)	Byte Access (Address 4n+2)	Byte Access (Address 4n+3)	Word Access (Address 4n)	Word Access (Address 4n+2)	Longword Access
$\overline{CS6}$ to $\overline{CS2}$, $\overline{CS0}$		Enabled	Enabled	Enabled	Enabled	Enabled	Enabled	Enabled
$\overline{RD}$	R	High	High	High	High	High	High	High
	W	High	High	High	High	High	High	High
RD/$\overline{WR}$	R	High	High	High	High	High	High	High
	W	Low	Low	Low	Low	Low	Low	Low
$\overline{BS}$		Enabled	Enabled	Enabled	Enabled	Enabled	Enabled	Enabled
$\overline{RAS3}$		Low/ High*1	Low/ High*1	Low/ High*1	Low/ High*1	Low/ High*1	Low/ High*1	Low/ High*1
$\overline{CAS}$		High/ Low*1	High/ Low*1	High/ Low*1	High/ Low*1	High/ Low*1	High/ Low*1	High/ Low*1
$\overline{WE0}$/DQMLL	R	Low	High	High	High	Low	High	Low
	W	Low	High	High	High	Low	High	Low
$\overline{WE1}$/DQMLU/$\overline{WE}$	R	High	Low	High	High	Low	High	Low
	W	High	Low	High	High	Low	High	Low
$\overline{WE2}$/DQMUL/$\overline{ICIORD}$	R	High	High	Low	High	High	Low	Low
	W	High	High	Low	High	High	Low	Low
$\overline{WE3}$/DQMUU/$\overline{ICIOWR}$	R	High	High	High	Low	High	Low	Low
	W	High	High	High	Low	High	Low	Low
$\overline{CE2A}$		High	High	High	High	High	High	High
$\overline{CE2B}$		High	High	High	High	High	High	High
CKE		High*2	High*2	High*2	High*2	High*2	High*2	High*2
$\overline{WAIT}$		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
$\overline{IOIS16}$		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
A25 to A0		Address command	Address command	Address command	Address command	Address command	Address command	Address command
D7 to D0		Valid data	Invalid data	Invalid data	Invalid data	Valid data	Invalid data	Valid data
D15 to D8		Invalid data	Valid data	Invalid data	Invalid data	Valid data	Invalid data	Valid data
D23 to D16		Invalid data	Invalid data	Valid data	Invalid data	Invalid data	Valid data	Valid data
D31 to D24		Invalid data	Invalid data	Invalid data	Valid data	Invalid data	Valid data	Valid data

Notes: 1. Low 32MB access/High 32MB access

2. Normally High, but Low during self-refresh.

Table A.8 Pin Status (Synchronous DRAM/Big Endian)

Pin		32-Bit Bus Width						
		Byte Access (Address 4n)	Byte Access (Address 4n+1)	Byte Access (Address 4n+2)	Byte Access (Address 4n+3)	Word Access (Address 4n)	Word Access (Address 4n+2)	Longword Access
$\overline{CS6}$ to $\overline{CS2}$, $\overline{CS0}$		Enabled	Enabled	Enabled	Enabled	Enabled	Enabled	Enabled
$\overline{RD}$	R	High	High	High	High	High	High	High
	W	High	High	High	High	High	High	High
RD/$\overline{WR}$	R	High	High	High	High	High	High	High
	W	Low	Low	Low	Low	Low	Low	Low
$\overline{BS}$		Enabled	Enabled	Enabled	Enabled	Enabled	Enabled	Enabled
$\overline{RAS3}$		Low/ High*1	Low/ High*1	Low/ High*1	Low/ High*1	Low/ High*1	Low/ High*1	Low/ High*1
$\overline{CAS}$		High/ Low*1	High/ Low*1	High/ Low*1	High/ Low*1	High/ Low*1	High/ Low*1	High/ Low*1
$\overline{WE0}$/DQMLL	R	High	High	High	Low	High	Low	Low
	W	High	High	High	Low	High	Low	Low
$\overline{WE1}$/DQMLU/$\overline{WE}$	R	High	High	Low	High	High	Low	Low
	W	High	High	Low	High	High	Low	Low
$\overline{WE2}$/DQMUL/$\overline{ICIORD}$	R	High	Low	High	High	Low	High	Low
	W	High	Low	High	High	Low	High	Low
$\overline{WE3}$/DQMUU/$\overline{ICIOWR}$	R	Low	High	High	High	Low	High	Low
	W	Low	High	High	High	Low	High	Low
$\overline{CE2A}$		High	High	High	High	High	High	High
$\overline{CE2B}$		High	High	High	High	High	High	High
CKE		High*2	High*2	High*2	High*2	High*2	High*2	High*2
$\overline{WAIT}$		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
$\overline{IOIS16}$		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
A25 to A0		Address command	Address command	Address command	Address command	Address command	Address command	Address command
D7 to D0		Valid data	Invalid data	Invalid data	Invalid data	Valid data	Invalid data	Valid data
D15 to D8		Invalid data	Valid data	Invalid data	Invalid data	Valid data	Invalid data	Valid data
D23 to D16		Invalid data	Invalid data	Valid data	Invalid data	Invalid data	Valid data	Valid data
D31 to D24		Invalid data	Invalid data	Invalid data	Valid data	Invalid data	Valid data	Valid data

Notes: 1. Low 32MB access/High 32MB access

2. Normally High, but Low during self-refresh.

Table A.9 Pin Status (PCMCIA/Little Endian)

Pin		PCMCIA Memory Interface (Area 5)				PCMCIA/IO Interface (Area 5)			
		8-Bit Bus Width	16-Bit Bus Width			8-Bit Bus Width	16-Bit Bus Width		
		Byte/ Word/ Longword Access	Byte Access (Address 2n)	Byte Access (Address 2n+1)	Word/ Longword Access	Byte/ Word/ Longword Access	Byte Access (Address 2n)	Byte Access (Address 2n+1)	Word/ Longword Access
$\overline{CS6}$ to $\overline{CS2}$, $\overline{CS0}$		Enabled	Enabled	High	Enabled	Enabled	Enabled	High	Enabled
$\overline{RD}$	R	Low	Low	Low	Low	High	High	High	High
	W	High	High	High	High	High	High	High	High
RD/$\overline{WR}$	R	High	High	High	High	High	High	High	High
	W	Low	Low	Low	Low	Low	Low	Low	Low
$\overline{BS}$		Enabled	Enabled	Enabled	Enabled	Enabled	Enabled	Enabled	Enabled
$\overline{RAS3}$		High	High	High	High	High	High	High	High
$\overline{CAS}$		High	High	High	High	High	High	High	High
$\overline{WE0}$/DQMLL	R	High	High	High	High	High	High	High	High
	W	High	High	High	High	High	High	High	High
$\overline{WE1}$/DQMLU/$\overline{WE}$	R	High	High	High	High	High	High	High	High
	W	Low	Low	Low	Low	High	High	High	High
$\overline{WE2}$/DQMUL/$\overline{ICIORD}$	R	High	High	High	High	Low	Low	Low	Low
	W	High	High	High	High	High	High	High	High
$\overline{WE3}$/DQMUU/$\overline{ICIOWR}$	R	High	High	High	High	High	High	High	High
	W	High	High	High	High	Low	Low	Low	Low
$\overline{CE2A}$		High	High	Low	Low	High	High	Low	Low
$\overline{CE2B}$		High	High	High	High	High	High	High	High
CKE		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
$\overline{WAIT}$		Enabled*1	Enabled*1	Enabled*1	Enabled*1	Enabled*1	Enabled*1	Enabled*1	Enabled*1
$\overline{IOIS16}$		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Enabled	Enabled
A25 to A0		Address	Address	Address	Address	Address	Address	Address	Address
D7 to D0		Valid data	Valid data	Invalid data	Valid data	Valid data	Valid data	Invalid data	Valid data
D15 to D8		High-Z*2	Invalid data	Valid data	Valid data	High-Z*2	Invalid data	Valid data	Valid data
D31 to D16		High-Z*2	High-Z*2	High-Z*2	High-Z*2	High-Z*2	High-Z*2	High-Z*2	High-Z*2

Table A.9 Pin Status (PCMCIA/Little Endian) (cont)

Pin		PCMCIA Memory Interface (Area 6)				PCMCIA/IO Interface (Area 6)			
		8-Bit Bus Width	16-Bit Bus Width			8-Bit Bus Width	16-Bit Bus Width		
		Byte/ Word/ Longword Access	Byte Access (Address 2n)	Byte Access (Address 2n+1)	Word/ Longword Access	Byte/ Word/ Longword Access	Byte Access (Address 2n)	Byte Access (Address 2n+1)	Word/ Longword Access
$\overline{CS6}$ to $\overline{CS2}$, $\overline{CS0}$		Enabled	Enabled	High	Enabled	Enabled	Enabled	High	Enabled
$\overline{RD}$	R	Low	Low	Low	Low	High	High	High	High
	W	High	High	High	High	High	High	High	High
RD/$\overline{WR}$	R	High	High	High	High	High	High	High	High
	W	Low	Low	Low	Low	Low	Low	Low	Low
$\overline{BS}$		Enabled	Enabled	Enabled	Enabled	Enabled	Enabled	Enabled	Enabled
$\overline{RAS3}$		High	High	High	High	High	High	High	High
$\overline{CAS}$		High	High	High	High	High	High	High	High
$\overline{WE0}$/DQMLL	R	High	High	High	High	High	High	High	High
	W	High	High	High	High	High	High	High	High
$\overline{WE1}$/DQMLU/$\overline{WE}$	R	High	High	High	High	High	High	High	High
	W	Low	Low	Low	Low	High	High	High	High
$\overline{WE2}$/DQMUL/$\overline{ICIORD}$	R	High	High	High	High	Low	Low	Low	Low
	W	High	High	High	High	High	High	High	High
$\overline{WE3}$/DQMUU/$\overline{ICIOWR}$	R	High	High	High	High	High	High	High	High
	W	High	High	High	High	Low	Low	Low	Low
$\overline{CE2A}$		High	High	High	High	High	High	High	High
$\overline{CE2B}$		High	High	Low	Low	High	High	Low	Low
CKE		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
$\overline{WAIT}$		Enabled*1	Enabled*1	Enabled*1	Enabled*1	Enabled*1	Enabled*1	Enabled*1	Enabled*1
$\overline{IOIS16}$		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Enabled	Enabled
A25 to A0		Address	Address	Address	Address	Address	Address	Address	Address
D7 to D0		Valid data	Valid data	Invalid data	Valid data	Valid data	Valid data	Invalid data	Valid data
D15 to D8		High-Z*2	Invalid data	Valid data	Valid data	High-Z*2	Invalid data	Valid data	Valid data
D31 to D16		High-Z*2	High-Z*2	High-Z*2	High-Z*2	High-Z*2	High-Z*2	High-Z*2	High-Z*2

Notes: 1. Disabled when the wait setting of the WCR2 register is 0.

2. Unused pins can be switched to port function, pull-up.

Table A.10 Pin Status (PCMCIA/Big Endian)

Pin		PCMCIA Memory Interface (Area 5)				PCMCIA/IO Interface (Area 5)			
		8-Bit Bus Width	16-Bit Bus Width			8-Bit Bus Width	16-Bit Bus Width		
		Byte/ Word/ Longword Access	Byte Access (Address 2n)	Byte Access (Address 2n+1)	Word/ Longword Access	Byte/ Word/ Longword Access	Byte Access (Address 2n)	Byte Access (Address 2n+1)	Word/ Longword Access
$\overline{CS6}$ to $\overline{CS2}$, $\overline{CS0}$		Enabled	Enabled	High	Enabled	Enabled	Enabled	High	Enabled
$\overline{RD}$	R	Low	Low	Low	Low	High	High	High	High
	W	High	High	High	High	High	High	High	High
RD/$\overline{WR}$	R	High	High	High	High	High	High	High	High
	W	Low	Low	Low	Low	Low	Low	Low	Low
$\overline{BS}$		Enabled	Enabled	Enabled	Enabled	Enabled	Enabled	Enabled	Enabled
$\overline{RAS3}$		High	High	High	High	High	High	High	High
$\overline{CAS}$		High	High	High	High	High	High	High	High
$\overline{WE0}$/DQMLL	R	High	High	High	High	High	High	High	High
	W	High	High	High	High	High	High	High	High
$\overline{WE1}$/DQMLU/$\overline{WE}$	R	High	High	High	High	High	High	High	High
	W	Low	Low	Low	Low	High	High	High	High
$\overline{WE2}$/DQMUL/$\overline{ICIORD}$	R	High	High	High	High	Low	Low	Low	Low
	W	High	High	High	High	High	High	High	High
$\overline{WE3}$/DQMUU/$\overline{ICIOWR}$	R	High	High	High	High	High	High	High	High
	W	High	High	High	High	Low	Low	Low	Low
$\overline{CE2A}$*3		High	High	Low	Low	High	High	Low	Low
$\overline{CE2B}$*3		High	High	High	High	High	High	High	High
CKE		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
$\overline{WAIT}$		Enabled*1	Enabled*1	Enabled*1	Enabled*1	Enabled*1	Enabled*1	Enabled*1	Enabled*1
$\overline{IOIS16}$		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
A25 to A0		Address	Address	Address	Address	Address	Address	Address	Address
D7 to D0		Valid data	Invalid data	Valid data	Valid data	Valid data	Invalid data	Valid data	Valid data
D15 to D8		High-Z*2	Valid data	Invalid data	Valid data	High-Z*2	Valid data	Invalid data	Valid data
D31 to D16		High-Z*2	High-Z*2	High-Z*2	High-Z*2	High-Z*2	High-Z*2	High-Z*2	High-Z*2

Table A.10 Pin Status (PCMCIA/Big Endian) (cont)

Pin		PCMCIA Memory Interface (Area 6)				PCMCIA/IO Interface (Area 6)			
		8-Bit Bus Width	16-Bit Bus Width			8-Bit Bus Width	16-Bit Bus Width		
		Byte/ Word/ Longword Access	Byte Access (Address 2n)	Byte Access (Address 2n+1)	Word/ Longword Access	Byte/ Word/ Longword Access	Byte Access (Address 2n)	Byte Access (Address 2n+1)	Word/ Longword Access
$\overline{CS6}$ to $\overline{CS2}$, $\overline{CS0}$		Enabled	Enabled	High	Enabled	Enabled	Enabled	High	Enabled
$\overline{RD}$	R	Low	Low	Low	Low	High	High	High	High
	W	High	High	High	High	High	High	High	High
RD/$\overline{WR}$	R	High	High	High	High	High	High	High	High
	W	Low	Low	Low	Low	Low	Low	Low	Low
$\overline{BS}$		Enabled	Enabled	Enabled	Enabled	Enabled	Enabled	Enabled	Enabled
$\overline{RAS3}$		High	High	High	High	High	High	High	High
$\overline{CAS}$		High	High	High	High	High	High	High	High
$\overline{WE0}$/DQMLL	R	High	High	High	High	High	High	High	High
	W	High	High	High	High	High	High	High	High
$\overline{WE1}$/DQMLU/$\overline{WE}$	R	High	High	High	High	High	High	High	High
	W	Low	Low	Low	Low	High	High	High	High
$\overline{WE2}$/DQMUL/$\overline{ICIORD}$	R	High	High	High	High	Low	Low	Low	Low
	W	High	High	High	High	High	High	High	High
$\overline{WE3}$/DQMUU/$\overline{ICIOWR}$	R	High	High	High	High	High	High	High	High
	W	High	High	High	High	Low	Low	Low	Low
$\overline{CE2A}$*3		High	High	High	High	High	High	High	High
$\overline{CE2B}$*3		High	High	Low	Low	High	High	Low	Low
CKE		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
$\overline{WAIT}$		Enabled*1	Enabled*1	Enabled*1	Enabled*1	Enabled*1	Enabled*1	Enabled*1	Enabled*1
$\overline{IOIS16}$		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
A25 to A0		Address	Address	Address	Address	Address	Address	Address	Address
D7 to D0		Valid data	Invalid data	Valid data	Valid data	Valid data	Invalid data	Valid data	Valid data
D15 to D8		High-Z*2	Valid data	Invalid data	Valid data	High-Z*2	Valid data	Invalid data	Valid data
D31 to D16		High-Z*2	High-Z*2	High-Z*2	High-Z*2	High-Z*2	High-Z*2	High-Z*2	High-Z*2

Notes: 1. Disabled when the wait setting of the WCR2 register is 0.

2. Unused pins can be switched to port function, pull-up.

3. The operation of the CE pin is the same as when operating in little endian mode.

Appendix B Control Registers

B.1 Register Address Map

Table B.1 Memory-Mapped Control Registers (Address Map)

Control Register	Module*1	Bus*2	Address*4	Size (Bits)	Access Size (Bits)*3
PTEH	CCN	L	H'FFFFFFF0	32	32
PTEL	CCN	L	H'FFFFFFF4	32	32
TTB	CCN	L	H'FFFFFFF8	32	32
TEA	CCN	L	H'FFFFFFFC	32	32
MMUCR	CCN	L	H'FFFFFFE0	32	32
CCR	CCN	L	H'FFFFFFEC	32	32
CCR2	CCN	I	H'040000B0	32	32
TRA	CCN	L	H'FFFFFFD0	32	32
EXPEVT	CCN	L	H'FFFFFFD4	32	32
INTEVT	CCN	L	H'FFFFFFD8	32	32
BASRA	CCN	L	H'FFFFFFE4	8	8
BASRB	CCN	L	H'FFFFFFE8	8	8
BARA	UBC	L	H'FFFFFFB0	32	32
BAMRA	UBC	L	H'FFFFFFB4	32	32
BBRA	UBC	L	H'FFFFFFB8	16	16
BARB	UBC	L	H'FFFFFFA0	32	32
BAMRB	UBC	L	H'FFFFFFA4	32	32
BBRB	UBC	L	H'FFFFFFA8	16	16
BDRB	UBC	L	H'FFFFFF90	32	32
BDMRB	UBC	L	H'FFFFFF94	32	32
BRCR	UBC	L	H'FFFFFF98	32	32
BETR	UBC	L	H'FFFFFF9C	16	16
BRSR	UBC	L	H'FFFFFFAC	32	32
BRDR	UBC	L	H'FFFFFFBC	32	32
FRQCR	CPG	I2	H'FFFFFF80	16	16
STBCR	CPG	I2	H'FFFFFF82	8	8
STBCR2	CPG	I2	H'FFFFFF88	8	8
WTCNT	CPG	I2	H'FFFFFF84	8	16
WTCSR	CPG	I2	H'FFFFFF86	8	16

Control Register	Module*1	Bus*2	Address*4	Size (Bits)	Access Size (Bits)*3
CKIO2CR	CPG	I2	H'0400023A	8	16
BCR1	BSC	I	H'FFFFFF60	16	16
BCR2	BSC	I	H'FFFFFF62	16	16
WCR1	BSC	I	H'FFFFFF64	16	16
WCR2	BSC	I	H'FFFFFF66	16	16
MCR	BSC	I	H'FFFFFF68	16	16
PCR	BSC	I	H'FFFFFF6C	16	16
RTCSR	BSC	I	H'FFFFFF6E	16	16
RTCNT	BSC	I	H'FFFFFF70	16	16
RTCOR	BSC	I	H'FFFFFF72	16	16
RFCR	BSC	I	H'FFFFFF74	16	16
SDMR	BSC	I	H'FFFFD000– H'FFFFEFFF	—	8
R64CNT	RTC	P	H'FFFFFEC0	8	8
RSECCNT	RTC	P	H'FFFFFEC2	8	8
RMINCNT	RTC	P	H'FFFFFEC4	8	8
RHRCNT	RTC	P	H'FFFFFEC6	8	8
RWKCNT	RTC	P	H'FFFFFEC8	8	8
RDAYCNT	RTC	P	H'FFFFFECA	8	8
RMONCNT	RTC	P	H'FFFFFECC	8	8
RYRCNT	RTC	P	H'FFFFFECE	8	8
RSECAR	RTC	P	H'FFFFFED0	8	8
RMINAR	RTC	P	H'FFFFFED2	8	8
RHRAR	RTC	P	H'FFFFFED4	8	8
RWKAR	RTC	P	H'FFFFFED6	8	8
RDAYAR	RTC	P	H'FFFFFED8	8	8
RMONAR	RTC	P	H'FFFFFEDA	8	8
RCR1	RTC	P	H'FFFFFEDC	8	8
RCR2	RTC	P	H'FFFFFEDE	8	8
ICR0	INTC	I2	H'FFFFFEE0	16	16
IPRA	INTC	I2	H'FFFFFEE2	16	16
IPRB	INTC	I2	H'FFFFFEE4	16	16
TSTR	TMU	P	H'FFFFFE92	8	8
TCOR0	TMU	P	H'FFFFFE94	32	32
TCNT0	TMU	P	H'FFFFFE98	32	32

Control Register	Module*1	Bus*2	Address*4	Size (Bits)	Access Size (Bits)*3
TCR0	TMU	P	H'FFFFFE9C	16	16
TCOR1	TMU	P	H'FFFFFEA0	32	32
TCNT1	TMU	P	H'FFFFFEA4	32	32
TCR1	TMU	P	H'FFFFFEA8	16	16
TCOR2	TMU	P	H'FFFFFEAC	32	32
TCNT2	TMU	P	H'FFFFFEB0	32	32
TCR2	TMU	P	H'FFFFFEB4	16	16
SCSMR	SCI	P	H'FFFFFE80	8	8
SCBRR	SCI	P	H'FFFFFE82	8	8
SCSCR	SCI	P	H'FFFFFE84	8	8
SCTDR	SCI	P	H'FFFFFE86	8	8
SCSSR	SCI	P	H'FFFFFE88	8	8
SCRDR	SCI	P	H'FFFFFE8A	8	8
SCSCMR	SCI	P	H'FFFFFE8C	8	8
INTEVT2	INTC	I2	H'04000000	32	32
IRR0	INTC	I2	H'04000004	16	8
IRR1	INTC	I2	H'04000006	16	8
IRR2	INTC	I2	H'04000008	16	8
ICR1	INTC	I2	H'04000010	16	16
ICR2	INTC	I2	H'04000012	16	16
PINTER	INTC	I2	H'04000014	16	16
IPRC	INTC	I2	H'04000016	16	16
IPRD	INTC	I2	H'04000018	16	16
IPRE	INTC	I2	H'0400001A	16	16
SAR0	DMAC	P1	H'04000020	32	16,32
DAR0	DMAC	P1	H'04000024	32	16,32
DMATCR0	DMAC	P1	H'04000028	32	16,32
CHCR0	DMAC	P1	H'0400002C	32	8,16,32
SAR1	DMAC	P1	H'04000030	32	16,32
DAR1	DMAC	P1	H'04000034	32	16,32
DMATCR1	DMAC	P1	H'04000038	32	16,32
CHCR1	DMAC	P1	H'0400003C	32	8,16,32
SAR2	DMAC	P1	H'04000040	32	16,32
DAR2	DMAC	P1	H'04000044	32	16,32
DMATCR2	DMAC	P1	H'04000048	32	16,32

Control Register	Module[*1]	Bus[*2]	Address[*4]	Size (Bits)	Access Size (Bits)[*3]
CHCR2	DMAC	P1	H'0400004C	32	8,16,32
SAR3	DMAC	P1	H'04000050	32	16,32
DAR3	DMAC	P1	H'04000054	32	16,32
DMATCR3	DMAC	P1	H'04000058	32	16,32
CHCR3	DMAC	P1	H'0400005C	32	8,16,32
DMAOR	DMAC	P1	H'04000060	16	8,16
CMSTR	CMT	P1	H'04000070	16	8,16,32
CMCSR	CMT	P1	H'04000072	16	8,16,32
CMCNT	CMT	P1	H'04000074	16	8,16,32
CMCOR	CMT	P1	H'04000076	16	8,16,32
ADDRAH	A/D	P1	H'04000080	8	8,16,32[*5 *6]
ADDRAL	A/D	P1	H'04000082	8	8,16[*5]
ADDRBH	A/D	P1	H'04000084	8	8,16,32[*5 *6]
ADDRBL	A/D	P1	H'04000086	8	8,16[*5]
ADDRCH	A/D	P1	H'04000088	8	8,16,32[*5 *6]
ADDRCL	A/D	P1	H'0400008A	8	8,16[*5]
ADDRDH	A/D	P1	H'0400008C	8	8,16,32[*5 *6]
ADDRDL	A/D	P1	H'0400008E	8	8,16[*5]
ADCSR	A/D	P1	H'04000090	8	8,16,32[*5 *6]
ADCR	A/D	P1	H'04000092	8	8,16
DADR0	D/A	P1	H'040000A0	8	8,16,32[*5 *6]
DADR1	D/A	P1	H'040000A2	8	8,16[*5]
DACR	D/A	P1	H'040000A4	8	8,16,32
SIMDR	SIOF	P2	H'040000C0	16	16
SISCR	SIOF	P2	H'040000C2	16	16
SITDAR	SIOF	P2	H'040000C4	16	16
SIRDAR	SIOF	P2	H'040000C6	16	16
SICDAR	SIOF	P2	H'040000C8	16	16
SICTR	SIOF	P2	H'040000CC	16	16
SIFCTR	SIOF	P2	H'040000D0	16	16
SISTR	SIOF	P2	H'040000D4	16	16
SIIER	SIOF	P2	H'040000D6	16	16
SITDR	SIOF	P2	H'040000E0	32	32
SIRDR	SIOF	P2	H'040000E4	32	32
SITCR	SIOF	P2	H'040000E8	32	32

Control Register	Module*1	Bus*2	Address*4	Size (Bits)	Access Size (Bits)*3
SIRCR	SIOF	P2	H'040000EC	32	32
PACR	PORT	P	H'04000100	16	16
PBCR	PORT	P	H'04000102	16	16
PCCR	PORT	P	H'04000104	16	16
PDCR	PORT	P	H'04000106	16	16
PECR	PORT	P	H'04000108	16	16
PFCR	PORT	P	H'0400010A	16	16
PGCR	PORT	P	H'0400010C	16	16
PHCR	PORT	P	H'0400010E	16	16
PJCR	PORT	P	H'04000110	16	16
PKCR	PORT	P	H'04000112	16	16
PLCR	PORT	P	H'04000114	16	16
SCPCR	PORT	P	H'04000116	16	16
PMCR	PORT	P	H'04000118	16	16
PADR	PORT	P	H'04000120	8	8
PBDR	PORT	P	H'04000122	8	8
PCDR	PORT	P	H'04000124	8	8
PDDR	PORT	P	H'04000126	8	8
PEDR	PORT	P	H'04000128	8	8
PFDR	PORT	P	H'0400012A	8	8
PGDR	PORT	P	H'0400012C	8	8
PHDR	PORT	P	H'0400012E	8	8
PJDR	PORT	P	H'04000130	8	8
PKDR	PORT	P	H'04000132	8	8
PLDR	PORT	P	H'04000134	8	8
SCPDR	PORT	P	H'04000136	8	8
PMDR	PORT	P	H'04000138	8	8
Reserved	—	P	H'04000140	—	Access prohibited
Reserved	—	P	H'04000142	—	Access prohibited
Reserved	—	P	H'04000144	—	Access prohibited
Reserved	—	P	H'04000146	—	Access prohibited
Reserved	—	P	H'04000148	—	Access prohibited
Reserved	—	P	H'0400014A	—	Access prohibited
Reserved	—	P	H'0400014C	—	Access prohibited
Reserved	—	P	H'0400014E	—	Access prohibited

Control Register	Module*1	Bus*2	Address*4	Size (Bits)	Access Size (Bits)*3
SCSMR2	SCIF	P1	H'04000150	8	8
SCBRR2	SCIF	P1	H'04000152	8	8
SCSCR2	SCIF	P1	H'04000154	8	8
SCFTDR2	SCIF	P1	H'04000156	8	8
SCSSR2	SCIF	P1	H'04000158	16	16
SCFRDR2	SCIF	P1	H'0400015A	8	8
SCFCR2	SCIF	P1	H'0400015C	8	8
SCFDR2	SCIF	P1	H'0400015E	16	16
PCC0ISR	PCC	P2	H'04000160	8	8
PCC0GCR	PCC	P2	H'04000162	8	8
PCC0CSCR	PCC	P2	H'04000164	8	8
PCC0CSCIER	PCC	P2	H'04000166	8	8
ACTR1	AFE	P2	H'04000180	16	16
ACTR2	AFE	P2	H'04000182	16	16
ASTR1	AFE	P2	H'04000184	16	16
ASTR2	AFE	P2	H'04000186	16	16
MRCR	AFE	P2	H'04000188	16	16
MPCR	AFE	P2	H'0400018A	16	16
DPNQ	AFE	P2	H'0400018C	16	16
RCNT	AFE	P2	H'0400018E	16	16
ACDR	AFE	P2	H'04000190	16	16
ASDR	AFE	P2	H'04000192	16	16
TDFP	AFE	P2	H'04000194	16	16/32
RDFP	AFE	P2	H'04000198	16	16/32
SDIR	H-UDI	I2	H'04000200	16	16
IPRF	PPCNT	P2	H'04000220	16	16
IPRG	PPCNT	P2	H'04000222	16	16
IRR3	PPCNT	P2	H'04000224	16	16
IRR4	PPCNT	P2	H'04000226	16	16
ICR3	PPCNR	P2	H'04000228	16	16
CHRAR	PPCNT	P2	H'0400022A	16	16
Reserved	—	—	H'0400022C	16	16
Reserved	—	—	H'0400022E	16	16
STBCR3	PPCNT	P2	H'04000230	8	8
SRSTR	PPCNT	P2	H'04000232	8	8

Control Register	Module*1	Bus*2	Address*4	Size (Bits)	Access Size (Bits)*3
Reserved	—	P2	H'04000238	16	16
EXPFC	PPCNT	P2	H'04000234	16	16
EXCPGCR	PPCNT	P2	H'04000236	8	8
USBIFR0	USBF	P2	H'04000240	8	8
USBIFR1	USBF	P2	H'04000241	8	8
USBEPDR0I	USBF	P2	H'04000242	8	8
USBEPDR0O	USBF	P2	H'04000243	8	8
USBTRG	USBF	P2	H'04000244	8	8
USBFCLR	USBF	P2	H'04000245	8	8
USBEPSZ0O	USBF	P2	H'04000246	8	8
USBEPDR0S	USBF	P2	H'04000247	8	8
USBDASTS	USBF	P2	H'04000248	8	8
USBEPDR2	USBF	P2	H'04000249	8	8/32
USBISR0	USBF	P2	H'0400024A	8	8
USBEPSTL	USBF	P2	H'0400024B	8	8
USBIER0	USBF	P2	H'0400024C	8	8
USBIER1	USBF	P2	H'0400024D	8	8
USBEPDR1	USBF	P2	H'0400024E	8	8/32
USBEPSZ1	USBF	P2	H'0400025F	8	8
USBISR1	USBF	P2	H'04000250	8	8
USBDMA	USBF	P2	H'04000251	8	8
USBEPDR3	USBF	P2	H'04000252	8	8
HcRevision (USBHR)	USBH	P2	H'04000400	32	32
HcControl (USBHC)	USBH	P2	H'04000404	32	32
HcCommandStatus (USBHCS)	USBH	P2	H'04000408	32	32
HcInterruptStatus (USBHIS)	USBH	P2	H'0400040C	32	32
HcInterruptEnable (USBHIE)	USBH	P2	H'04000410	32	32
HcInterruptDisable (USBHID)	USBH	P2	H'04000414	32	32
HcHCCA (USBHHCCA)	USBH	P2	H'04000418	32	32

Control Register	Module*1	Bus*2	Address*4	Size (Bits)	Access Size (Bits)*3
HcPeriodCurrentED (USBHPCED)	USBH	P2	H'0400041C	32	32
HcControlHeadED (USBHCHED)	USBH	P2	H'04000420	32	32
HcControlCurrentED (USBHCCED)	USBH	P2	H'04000424	32	32
HcBulkHeadED (USBHBHED)	USBH	P2	H'04000428	32	32
HcBulkCurrentED (USBHBCED)	USBH	P2	H'0400042C	32	32
HcDoneHead (USBHDHED)	USBH	P2	H'04000430	32	32
HcFmInterval (USBHFI)	USBH	P2	H'04000434	32	32
HcFmRemaining (USBHFR)	USBH	P2	H'04000438	32	32
HcFmNumber (USBHFN)	USBH	P2	H'0400043C	32	32
HcPeriodStart (USBHPS)	USBH	P2	H'04000440	32	32
HcLSThreshold (USBHLST)	USBH	P2	H'04000444	32	32
HcRhDescriptorA (USBHRDA)	USBH	P2	H'04000448	32	32
HcRhDescriptorB (USBHRDB)	USBH	P2	H'0400044C	32	32
HcRhStatus (USBHRS)	USBH	P2	H'04000450	32	32
HcRhPortStatus1 (USBHRPS1)	USBH	P2	H'04000454	32	32
HcRhPortStatus2 (USBHRPS2)	USBH	P2	H'04000458	32	32
LDPR00 to LDPRFF	LCDC	P2	H'04000800 to H'04000BFC	32	32
LDICKR	LCDC	P2	H'04000C00	16	16
LDMTR	LCDC	P2	H'04000C02	16	16
LDDFR	LCDC	P2	H'04000C04	16	16
LDSMR	LCDC	P2	H'04000C06	16	16
LDSARU	LCDC	P2	H'04000C08	32	32

Control Register	Module*1	Bus*2	Address*4	Size (Bits)	Access Size (Bits)*3
LDSARL	LCDC	P2	H'04000C0C	32	32
LDLAOR	LCDC	P2	H'04000C10	16	16
LDPALCR	LCDC	P2	H'04000C12	16	16
LDHCNR	LCDC	P2	H'04000C14	16	16
LDHSYNR	LCDC	P2	H'04000C16	16	16
LDVDLNR	LCDC	P2	H'04000C18	16	16
LDVTLNR	LCDC	P2	H'04000C1A	16	16
LDVSYNR	LCDC	P2	H'04000C1C	16	16
LDACLNR	LCDC	P2	H'04000C1E	16	16
LDINTR	LCDC	P2	H'04000C20	16	16
LDMPPR	LCDC	P2	H'04000C24	16	16
LDPSPR	LCDC	P2	H'04000C26	16	16
LDCNTR	LCDC	P2	H'04000C28	16	16

Notes: 1. Modules

CCN: Cache controller
CPG: Clock pulse generator
RTC: Realtime clock
TMU: Timer unit
DMAC: Direct memory access controller
CMT: Compare-match timer
D/A: D/A converter
PORT: Port control
PCC: PC card controller
H-UDI: User-debugging interface
PPCNT: Peripheral bus interface controller
USBF: USB function controller
LCDC: LCD controller
UBC: User break controller
BSC: Bus state controller
INTC: Interrupt controller
SCI: Serial communication interface
A/D: A/D converter
SIOF: Serial I/O
SCIF: Serial communication interface with FIFO
AFE: Analog front end interface
USBH: USB host controller

2. Internal buses:

L: CPU, CCN, cache, TLB, and DSP connected
I: BSC, cache, DMAC, INTC, CPG, and UDI connected
I2: INTC, CPG, and H-UDI connected
P: Peripheral modules (RTC, TMU, SCI) connected
P1: Peripheral modules (DMAC, CMT, A/D, D/A, PORT, SCIF) connected
P2: Peripheral modules (SIOF, PCC, AFE, PPCNT, USBF, USBH, LCDC) connected

3. The access size shown is for control register access (read/write). An incorrect result will be obtained if a different size from that shown is used for access.
4. To exclude area 1 control registers from address translation by the MMU, set the first 3 bits of the logical address to 101, to locate the registers in the P2 space.

5. With 16-bit access, it is not possible to read data in two registers simultaneously.
6. With 32-bit access, it is not possible to read data in the register at [accessed address + 2] simultaneously.

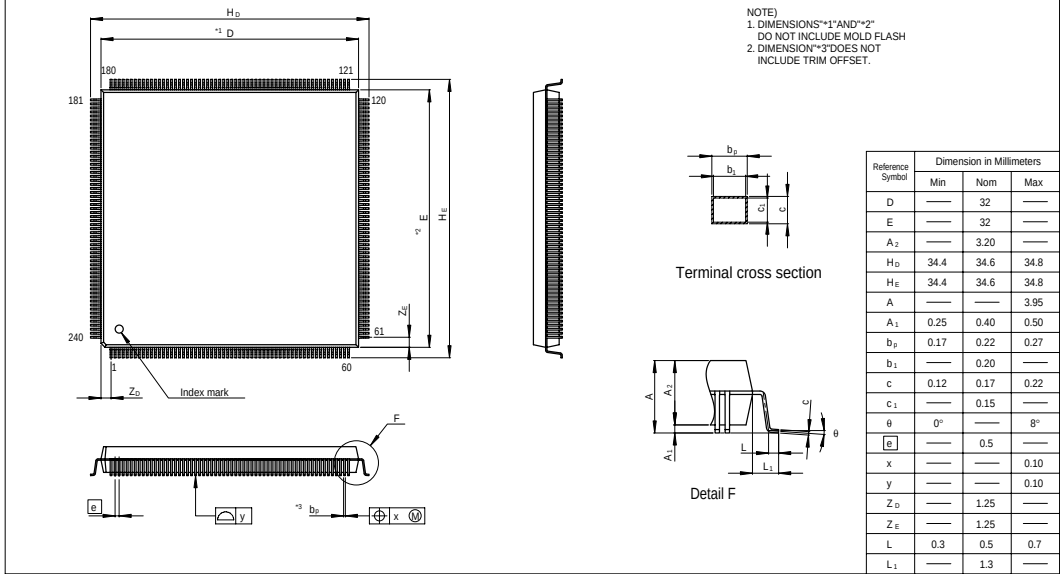
Appendix C Product Lineup

Abbreviation	Power Supply Voltages I/O	Power Supply Voltages Internal	Operation Frequency	Model Name	Package
SH7727	3.3±0.3 V	1.7 to 2.05 V	160 MHz	HD6417727F160C	240-pin plastic HQFP (PRQP0240KC-B)
	3.3±0.3 V	1.7 to 2.05 V	160 MHz	HD6417727BP160C	240-pin CSP (PLBG0240JA-A)
	3.1±0.5 V	1.6 to 2.05 V	100 MHz	HD6417727F100C	240-pin plastic HQFP (PRQP0240KC-B)
	3.1±0.5 V	1.6 to 2.05 V	100 MHz	HD6417727BP100C	240-pin CSP (PLBG0240JA-A)

Appendix D Package Dimensions

The following drawings show the package dimensions of SH7727.

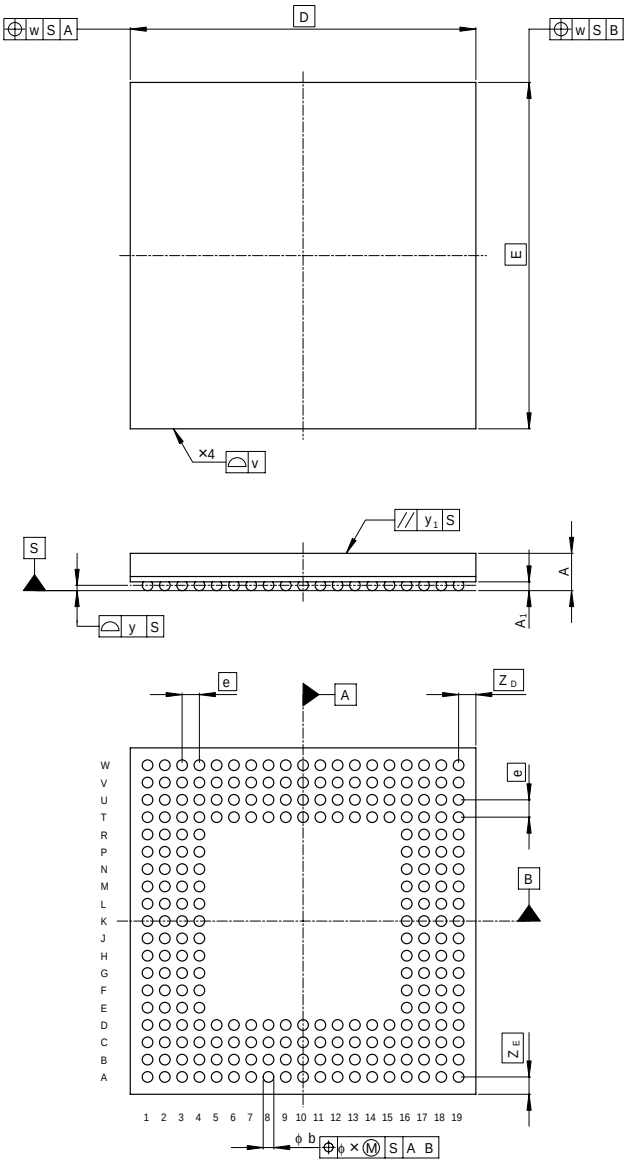
JEITA Package Code	RENESAS Code	Previous Code	MASS[Typ.]
P-HQFP240-32x32-0.50	PRQP0240KC-B	FP-240B/FP-240BV	7.0g



Reference Symbol	Dimension in Millimeters		
	Min	Nom	Max
D	—	32	—
E	—	32	—
A_2	—	3.20	—
H_D	34.4	34.6	34.8
H_E	34.4	34.6	34.8
A	—	—	3.95
A_1	0.25	0.40	0.50
b_p	0.17	0.22	0.27
b_1	—	0.20	—
c	0.12	0.17	0.22
c_1	—	0.15	—
θ	0°	—	8°
e	—	0.5	—
x	—	—	0.10
y	—	—	0.10
Z_D	—	1.25	—
Z_E	—	1.25	—
L	0.3	0.5	0.7
L_1	—	1.3	—

Figure D.1 Package Dimensions (PRQP0240KC-B)

JEITA Package Code	RENESAS Code	Previous Code	MASS[Typ.]
P-LFBGA240-13x13-0.65	PLBG0240JA-A	BP-240A/BP-240AV	0.4g



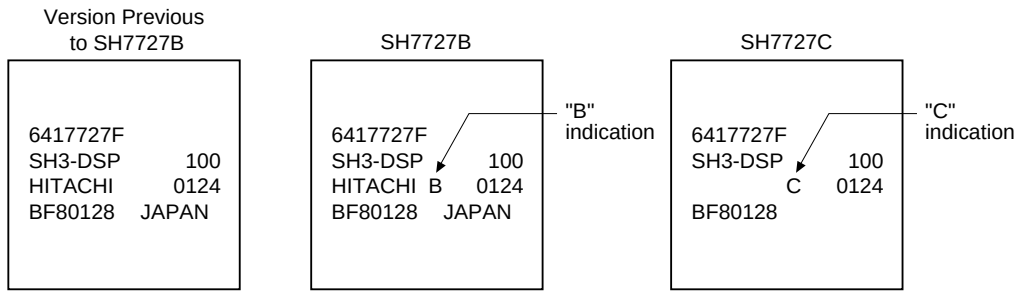
Reference Symbol	Dimension in Millimeters		
	Min	Nom	Max
D	——	13.00	——
E	——	13.00	——
v	——	——	0.15
w	——	——	0.20
A	——	——	1.40
A_1	0.28	0.33	0.38
e	——	0.65	——
b	0.35	0.40	0.45
x	——	——	0.08
y	——	——	0.10
y_1	——	——	0.2
S_D	——	——	——
S_E	——	——	——
Z_D	——	0.65	——
Z_E	——	0.65	——

Figure D.2 Package Dimensions (PLBG0240JA-A)

Appendix E Using Versions Previous to the SH7727C

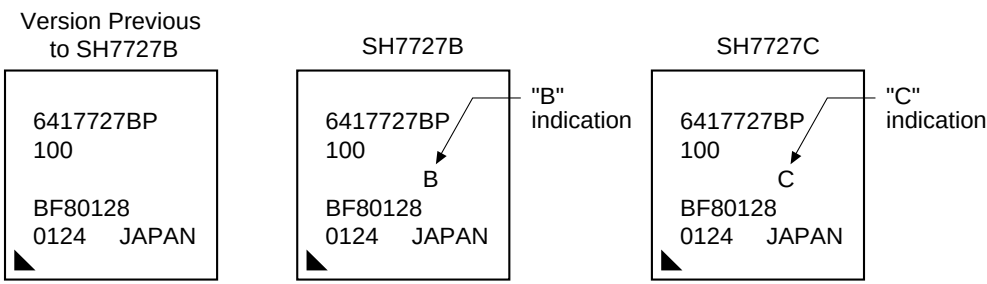
E.1 Determining the Version Number Based on the Markings on the Chip

(1) HQFP-240 Package



Note: Once stocks bearing Hitachi markings are exhausted, chips bearing Renesas markings may begin to appear.

(2) CSP-240 Package



Appendix F Using Port G Control Register (PGCR) with Versions Previous to the SH7727B

Bit:	15	14	13	12	11	10	9	8
	PG7MD1	PG7MD0	—	—	PG5MD1	PG5MD0	PG4MD1	PG4MD0
Initial value:	1	0	1	0	1/0	0	1	0
R/W:	R/W	R/W	R	R	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
	PG3MD1	PG3MD0	PG2MD1	PG2MD0	PG1MD1	PG1MD0	—	PG0MD0
Initial value:	1/0	0	1/0	0	1/0	0	1/0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W

The port G control register (PGCR) is a readable and writeable 16-bit register used to select pin functions. PGCR is initialized to H'AAAA ($\overline{\text{ASEMD0}}$ = 1) or H'A200 ($\overline{\text{ASEMD0}}$ = 0) at power-on reset, but it is not initialized by manual resets or in the standby mode or sleep mode.

Bits 15, 14: PG7 Mode 1, 0 (PG7MD1, PG7MD0)
Bits 13, 12: Reserved
Bits 11, 10: PG5 Mode 1, 0 (PG5MD1, PG5MD0)
Bits 9, 8: PG4 Mode 1, 0 (PG4MD1, PG4MD0)
Bits 7, 6: PG3 Mode 1, 0 (PG3MD1, PG3MD0)
Bits 5, 4: PG2 Mode 1, 0 (PG2MD1, PG2MD0)
Bits 3, 2: PG1 Mode 1, 0 (PG1MD1, PG1MD0)
Bit 1: Reserved
Bits 3, 0: PG0 Mode 1, 0 (PG1MD1, PG0MD0)

These bits are used to select pin functions and input pull-up MOS control settings. In the PG1 and PG0 modes, bit 3 (PG1MD1) is used to select between “other function” and “port input.” When the port input setting is selected (PG1MD1 = 1), pull-up MOS on-off selection is performed using bit 2 (PG1MD0) in the PG1 mode and bit 0 (PG0MD0) in the PG0 mode.

PG0 Mode

Bit 3: PG1MD1	Bit 0: PG0MD0	Pin function	
0	0	Other function (see table 26.1)	(Initial value) $\overline{\text{ASEMD0}}$ = 0
	1	Reserved	
1	0	Port input (pull-up MOS: on)	(Initial value) $\overline{\text{ASEMD0}}$ = 1
	1	Port input (pull-up MOS: off)	

PG1 to PG5 and PG7 Mode

Bit (2n + 1): PGnMD1	Bit 2n: PGnMD0	Pin function	
0	0	Other function (n = 1, 2, 5) (see table 26.1)	(Initial value) $\overline{\text{ASEMD0}}$ = 0
	1	Reserved	
1	0	Port input (pull-up MOS: on)	(Initial value) $\overline{\text{ASEMD0}}$ = 1
	1	Port input (pull-up MOS: off)	

(n = 1, 2, 3, 5)

Bit (2n + 1): PGnMD1	Bit 2n: PGnMD0	Pin function	
0	0	Other function (see table 26.1) (n = 7), Reserved (n = 4)	
	1	Reserved	
1	0	Port input (pull-up MOS: on)	(Initial value)
	1	Port input (pull-up MOS: off)	

(n = 4, 7)

Renesas 32-Bit RISC Microcomputer
Hardware Manual
SH7727 Group

Publication Date: 1st Edition, August 2001
Rev.5.00, December 12, 2005
Published by: Sales Strategic Planning Div.
Renesas Technology Corp.
Edited by: Customer Support Department
Global Strategic Communication Div.
Renesas Solutions Corp.

©2005. Renesas Technology Corp., All rights reserved. Printed in Japan.

Renesas Technology Corp. Sales Strategic Planning Div. Nippon Bldg., 2-6-2, Ohte-machi, Chiyoda-ku, Tokyo 100-0004, Japan

RENESAS SALES OFFICES

http://www.renesas.com

Refer to "**http://www.renesas.com/en/network**" for the latest and detailed information.

Renesas Technology America, Inc.
450 Holger Way, San Jose, CA 95134-1368, U.S.A
Tel: <1> (408) 382-7500, Fax: <1> (408) 382-7501

Renesas Technology Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K.
Tel: <44> (1628) 585-100, Fax: <44> (1628) 585-900

Renesas Technology (Shanghai) Co., Ltd.
Unit 205, AZIA Center, No.133 Yincheng Rd (n), Pudong District, Shanghai 200120, China
Tel: <86> (21) 5877-1818, Fax: <86> (21) 6887-7898

Renesas Technology Hong Kong Ltd.
7th Floor, North Tower, World Finance Centre, Harbour City, 1 Canton Road, Tsimshatsui, Kowloon, Hong Kong
Tel: <852> 2265-6688, Fax: <852> 2730-6071

Renesas Technology Taiwan Co., Ltd.
10th Floor, No.99, Fushing North Road, Taipei, Taiwan
Tel: <886> (2) 2715-2888, Fax: <886> (2) 2713-2999

Renesas Technology Singapore Pte. Ltd.
1 Harbour Front Avenue, #06-10, Keppel Bay Tower, Singapore 098632
Tel: <65> 6213-0200, Fax: <65> 6278-8001

Renesas Technology Korea Co., Ltd.
Kukje Center Bldg. 18th Fl., 191, 2-ka, Hangang-ro, Yongsan-ku, Seoul 140-702, Korea
Tel: <82> (2) 796-3115, Fax: <82> (2) 796-2145

Renesas Technology Malaysia Sdn. Bhd
Unit 906, Block B, Menara Amcorp, Amcorp Trade Centre, No.18, Jalan Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: <603> 7955-9390, Fax: <603> 7955-9510

SH7727 Group
Hardware Manual



Renesas Technology Corp.
2-6-2, Ote-machi, Chiyoda-ku, Tokyo, 100-0004, Japan