To our customers,

Old Company Name in Catalogs and Other Documents

On April 1st, 2010, NEC Electronics Corporation merged with Renesas Technology Corporation, and Renesas Electronics Corporation took over all the business of both companies. Therefore, although the old company name remains in this document, it is a valid Renesas Electronics document. We appreciate your understanding.

Renesas Electronics website: http://www.renesas.com

April 1st, 2010
Renesas Electronics Corporation

Issued by: Renesas Electronics Corporation (http://www.renesas.com)

Send any inquiries to http://www.renesas.com/inquiry.

Notice

1. All information included in this document is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas Electronics products listed herein, please confirm the latest product information with a Renesas Electronics sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas Electronics such as that disclosed through our website.
2. Renesas Electronics does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties by or arising from the use of Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
3. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part.
4. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
5. When exporting the products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations. You should not use Renesas Electronics products or the technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. Renesas Electronics products and technology may not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations.
6. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
7. Renesas Electronics products are classified according to the following three quality grades: "Standard", "High Quality", and "Specific". The recommended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below. You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application categorized as "Specific" without the prior written consent of Renesas Electronics. Further, you may not use any Renesas Electronics product for any application for which it is not intended without the prior written consent of Renesas Electronics. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for an application categorized as "Specific" or for which the product is not intended where you have failed to obtain the prior written consent of Renesas Electronics. The quality grade of each Renesas Electronics product is "Standard" unless otherwise expressly specified in a Renesas Electronics data sheets or data books, etc.

   "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; and industrial robots.

   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; safety equipment; and medical equipment not specifically designed for life support.

   "Specific": Aircraft; aerospace equipment; submersible repeaters; nuclear reactor control systems; medical equipment or systems for life support (e.g. artificial life support devices or systems), surgical implantations, or healthcare intervention (e.g. excision, etc.), and any other applications or purposes that pose a direct threat to human life.
8. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

(Note 1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its majority-owned subsidiaries.

(Note 2) "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics.

32

Hardware Manual
Renesas 32-Bit RISC Microcomputer
SuperH™ RISC engine Family/SH7750 Series

SH7760 HD6417760BL2
HD6417760BL2
HD6417760BL2
HD6417760BL2
HD6417760BP2
HD6417760BP2

Renesas Electronics
www.renesas.com

Rev.2.00

accuracy or completeness of the information contained in this document nor grants any license to any intellectual property rights or any other rights of Renesas or any third party with respect to the information in this document.

2. Renesas shall have no liability for damages or infringement of any intellectual property or other rights arising out of the use of any information in this document, including, but not limited to, product data, diagrams, charts, programs, algorithms, and application circuit examples.
3. You should not use the products or the technology described in this document for the purpose of military applications such as the development of weapons of mass destruction or for the purpose of any other military use. When exporting the products or technology described herein, you should follow the applicable export control laws and regulations, and procedures required by such laws and regulations.
4. All information included in this document such as product data, diagrams, charts, programs, algorithms, and application circuit examples, is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas products listed in this document, please confirm the latest product information with a Renesas sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas such as that disclosed through our website. (http://www.renesas.com)
5. Renesas has used reasonable care in compiling the information included in this document, but Renesas assumes no liability whatsoever for any damages incurred as a result of errors or omissions in the information included in this document.
6. When using or otherwise relying on the information in this document, you should evaluate the information in light of the total system before deciding about the applicability of such information to the intended application. Renesas makes no representations, warranties or guaranties regarding the suitability of its products for any particular application and specifically disclaims any liability arising out of the application and use of the information in this document or Renesas products.
7. With the exception of products specified by Renesas as suitable for automobile applications, Renesas products are not designed, manufactured or tested for applications or otherwise in systems the failure or malfunction of which may cause a direct threat to human life or create a risk of human injury or which require especially high quality and reliability such as safety systems, or equipment or systems for transportation and traffic, healthcare, combustion control, aerospace and aeronautics, nuclear power, or undersea communication transmission. If you are considering the use of our products for such purposes, please contact a Renesas sales office beforehand. Renesas shall have no liability for damages arising out of the uses set forth above.
8. Notwithstanding the preceding paragraph, you should not use Renesas products for the purposes listed below:
   (1) artificial life support devices or systems
   (2) surgical implantations
   (3) healthcare intervention (e.g., excision, administration of medication, etc.)
   (4) any other purposes that pose a direct threat to human life

   Renesas shall have no liability for damages arising out of the uses set forth in the above and purchasers who elect to use Renesas products in any of the foregoing applications shall indemnify and hold harmless Renesas Technology Corp., its affiliated companies and their officers, directors, and employees against any and all damages arising out of such applications.
9. You should use the products described herein within the range specified by Renesas, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas shall have no liability for malfunctions or damages arising out of the use of Renesas products beyond such specified ranges.
10. Although Renesas endeavors to improve the quality and reliability of its products, IC products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Please be sure to implement safety measures to guard against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other applicable measures. Among others, since the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
11. In case Renesas products listed in this document are detached from the products to which the Renesas products are attached or affixed, the risk of accident such as swallowing by infants and small children is very high. You should implement safety measures so that Renesas products may not be easily detached from your products. Renesas shall have no liability for damages arising out of such detachment.
12. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written approval from Renesas.
13. Please contact a Renesas sales office if you have any questions regarding the information contained in this document, Renesas semiconductor products, or if you have any other inquiries.

on the products covered by this manual, refer to the relevant sections of the manual. If the descriptions under General Precautions in the Handling of MPU/MCU Products and in the body of the manual differ from each other, the description in the body of the manual takes precedence.

1. Handling of Unused Pins

   Handle unused pins in accord with the directions given under Handling of Unused Pins in the manual.

   — The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of LSI, an associated shoot-through current flows internally, and malfunctions may occur due to the false recognition of the pin state as an input signal. Unused pins should be handled as described under Handling of Unused Pins in the manual.

2. Processing at Power-on

   The state of the product is undefined at the moment when power is supplied.

   — The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the moment when power is supplied.

   In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the moment when power is supplied until the reset process is completed.

   In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the moment when power is supplied until the power reaches the level at which resetting has been specified.

3. Prohibition of Access to Reserved Addresses

   Access to reserved addresses is prohibited.

   — The reserved addresses are provided for the possible future expansion of functions. Do not access these addresses; the correct operation of LSI is not guaranteed if they are accessed.

4. Clock Signals

   After applying a reset, only release the reset line after the operating clock signal has become stable. When switching the clock signal during program execution, wait until the target clock signal has stabilized.

   — When the clock signal is generated with an external resonator (or from an external oscillator) during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Moreover, when switching to a clock signal produced with an external resonator (or by an external oscillator) while program execution is in progress, wait until the target clock signal is stable.

5. Differences between Products

   Before changing from one product to another, i.e. to one with a different type number, confirm that the change will not lead to problems.

   — The characteristics of MPU/MCU in the same group but having different type numbers may differ because of the differences in internal memory capacity and layout pattern. When changing to products of different type numbers, implement a system-evaluation test for each of the products.

The SH7760 RISC (Reduced Instruction Set Computer) microcomputer includes a Renesas Technology-original RISC CPU as its core, and the peripheral functions required to configure a system.

Target Users: This manual was written for users who will be using this LSI in the design of application systems. Users of this manual are expected to understand the fundamentals of electrical circuits, logical circuits, and microcomputers.

Objective: This manual was written to explain the hardware functions and electrical characteristics of this LSI to the above users.

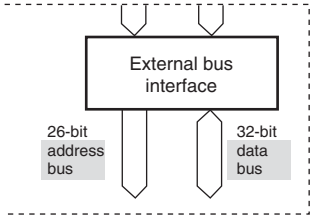
Notes on reading this manual:

- In order to understand the overall functions of the chip
  Read the manual according to the contents. This manual can be roughly categorized into parts on the CPU, system control functions, and peripheral functions.

  Rules: Bit order: The MSB is on the left and the LSB is on the right.

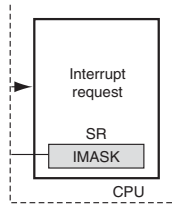
Related Manuals: The latest versions of all related manuals are available from our web site. Please ensure you have the latest versions of all documents you require. http://www.renesas.com

Refer to the SH-4 Software Manual for details on the instruction set.

Item	Page	Revision (See Manual for Details)
All	—	Company name and brand names amended (Before) Hitachi, Ltd. → (After) Renesas Technology Corp.
1.1 SH7760 Features Table 1.1 Features	2	Table amended **Item** **Features** LSI • Packages: 256-pin BGA (Size: 17 × 17 mm, pin pitch: 0.8 mm) CPU • RISC-type instruction set (upward-compatible with SH-1, SH-2, and SH-3)
	3	Table amended **Item** **Features** FPU • Floating-point registers: 32 bits × 16 × 2 banks — (single-precision: 32-bit × 16 or double-precision: 64-bit × 8) × 2 banks
	9	"Product lineup" added
1.2 Block Diagram Figure 1.1 SH7760 Block Diagram	10	Figure amended 
1.3 Pin Arrangement Figure 1.2 SH7760 Pin Arrangement (BP-256F/BP-256FV)	11	Figure added
Figure 1.3 SH7760 Pin Arrangement (BP-256B/BP-256BV)	12	Figure title amended
1.4 Pin Description	13	Description amended Table 1.2 lists the pin configuration of the BP-256F (21 mm*1), and table 1.3 lists the pin configuration of the BP-256B (17 mm*2).
		Notes added Notes: 1. HD6417760BP200AD, HD6417760BP200ADV 2. HD6417760BL200A, HD6417760BL200AV, HD6417760BL200AD, HD6417760BL200ADV
Table 1.2 Pin Configuration (BP-256F: 21 mm)	13 to 21	Table added

Configuration (B: 256B: 17 mm)

1.5 Pin Function Table 1.4 Pin Functions	31	Table amended Pin No. / MFI mode (MD7=0) / LCDC mode (MD7=1) / Other modes / Register 17 mm*1 21mm*2 Function Pin Name I/O Function Pin Name I/O Function Pin Name I/O GPIO GPIO Setting IP Selection Notes added Notes: 1. HD6417760BL200A, HD6417760BL200AV, HD6417760BL200AD, HD6417760BL200ADV 2. HD6417760BP200AD, HD6417760BP200ADV
Table 1.5 Pin Functions	32 to 34	Table amended Pin No. / Register 17 mm*1 21 mm*2 Function 1 Pin Name I/O Function 2 Pin Name I/O Function 3 Pin Name I/O GPIO GPIO Setting IP Selection
	34	Notes added Notes: 1. HD6417760BL200A, HD6417760BL200AV, HD6417760BL200AD, HD6417760BL200ADV 2. HD6417760BP200AD, HD6417760BP200ADV
Table 1.6 Pin Functions	35 to 38	Table amended Pin No. / Memory Interface 17 mm*1 21 mm*2 Function Pin Name I/O SRAM SDRAM PCMCIA MPX Remarks
	38	Notes added Notes: 1. HD6417760BL200A, HD6417760BL200AV, HD6417760BL200AD, HD6417760BL200ADV 2. HD6417760BP200AD, HD6417760BP200ADV
2.2.1 Privileged Mode and Banks Table 2.1 Initial Register Values	41	Table amended Type / Registers / Initial Value* Control registers SR MD bit = 1, RB bit = 1, BL bit = 1, FD bit = 0, IMASK = 1111 (H'F), reserved bits = 0, others = undefined
3.1 Features	51	Description amended • Designed to meet IEEE754 standard
3.7 Usage Notes	63 to 66	Newly added
4.4 Usage Note	84 to 86	Newly added
Section 5 Pipelining	87	Description amended The definitions in this section may not be applied to the SH-4 products other than the SH7760.
5.4 Usage Note	112	Newly added

Entry Assistance Register (PTEA)		When performing access from the CPU in SH7760 to the PCMCIA interface area with the AT bit in MMUCR cleared to 0, access is always performed using the values of the SA and TC bits in this register.
7.3.9 Note on Using Cache Enhanced Mode	164 to 166	Newly added
8.5.1 Resets (1) Power-On Reset	195	Description amended SR.IMASK = B'1111;
(2) Manual Reset	196	Description amended SR.IMASK = B'1111;
(3) H-UDI Reset	197	Description amended SR.IMASK = B'1111;
(4) Instruction TLB Multiple-Hit Exception	198	Description amended SR.IMASK = B'1111;
(5) Data TLB Multiple-Hit Exception	199	Description amended SR.IMASK = B'1111;
8.7.1 Restrictions on First Instruction in Exception Handling Routine	220	Note added • When the UBDE bit in BRCR is set to 1 and the user break debug support function* is used, do not locate a BT, BF, BT/S, BF/S, BRA, or BSR instruction at the address indicated by DBR. Note: * See section 31.5, User Break Debug Support Function.
9.1 Features Figure 9.1 Block Diagram of INTC	222	Figure amended 
9.2 Input/Output Pins Table 9.1 Pin Configuration	223	Table amended **Pin Name** / **Abbreviation** / **I/O** / **Function** IRL interrupt input pins / $\overline{IRL3}$ to $\overline{IRL0}$ / Input / Input of IRL interrupt request signals (maskable by the IMASK bits in SR)

Exception Handling and Priority Table 9.7 Interrupt Exception Handling Sources and Priority Order		Interrupt Source: MFI, MFII; Vector Code: H'E80
10.5 Register Descriptions Table 10.6 Register Configuration (2)	259	Note amended Notes: 1. For details, refer to the descriptions of SDMR.
10.5.5 Wait Control Register 1 (WCR1) Table 10.7 Idle Insertion between Accesses	273	Note added Notes: 2. On the MPX interface, a WCR1 idle wait may be inserted before an access (either read or write) to the same area after a write access. An example of idle wait insertion in accesses to the same area is shown below. (a) Synchronous DRAM set to RAS down mode (b) Synchronous DRAM accessed by on-chip DMAC Under conditions other than conditions (a) and (b) above, an idle wait is also inserted between an MPX interface write access and an immediately following access to the same area.
10.5.7 Wait Control Register 3 (WCR3)	282	Table and note amended

Bit	Bit Name	Initial Value	R/W	Description
4n + 1	AnH1*	All 1	R/W	Area n Data Hold Time
4n	AnH0*	All 1	R/W	For writing, specifies the number of cycles to be inserted during the data hold time after the write strobe is negated. For reading, specifies the number of cycles to be inserted during the data hold time after the data sampling timing. Valid only for SRAM interface, byte control SRAM interface, and burst ROM interface: Cycles to be inserted during the data hold time 00: 0 01: 1 10: 2 11: 3
4m + 3	AmRDH	All 0	R/W	Read-Strobe Negate Timing For reading, these bits specify the timing for the negation of read strobe. These bits should be cleared to 0 when byte control SRAM interface is in use. See figure 10.12.

Notes: n = 0 to 6; m = 1 and 4

* For area 1, only use the combinations listed in table 10.9 for the settings.

Register 4 (WCR4)

WCR4 is a 32-bit readable/writable register that specifies the negation period for the $\overline{CS1}$ signal. Specifying bits CSH1 and CSH0 can insert the negation cycles from 0 to 3. If the $\overline{CS1}$ negate period is set, it must be set to match the WCR3 data hold time (A1H[1:0]). If the $\overline{CS1}$ negate period is not specified (CSH[1:0] = 00), there is no need to match the WCR3 data hold time (A1H[1:0]).

283 Table amended and note added

Bit	Bit Name	Initial Value	R/W	Description
31 to 2		All 0	R	Reserved These bits are always read as 0, and the write value should always be 0.
1	CSH1*	0	R/W	CS Hold Cycle Setting
0	CSH0*	0	R/W	Specifies the number of wait cycles inserted during data hold after $\overline{CS1}$ is negated. Wait cycles to be inserted 00: 0 01: 1 10: 2 11: 3 If a value other than 00 is set, set WCR3.A1RDH to 1.

Note: * Only use the combinations listed in table 10.9 for the settings.

Table 10.9 WCR3 and WCR4 Settings for Area 1 — Newly added

10.5.9 Memory Control Register (MCR) 285 Table amended

Bit	Bit Name	Initial Value	R/W	Description
30	MRSET	0	R/W	Mode Register Set Set this bit to 1 to make the mode register setting for synchronous DRAM. See the description of power-on sequence in section 10.6.4, (10) Power-On Sequence. 0: All-bank precharge 1: Mode register setting

10.5.12 Refresh Timer Control/Status Register (RTCSR) 294 Table amended

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	—	All 0		Reserved These bits are always read as 0.

10.6.2 Areas 308 Description amended

(5) Area 4

In addition, any number of wait cycles can be inserted in each bus cycle by the external wait pin ($\overline{RDY}$).

Control Register (CHCR)

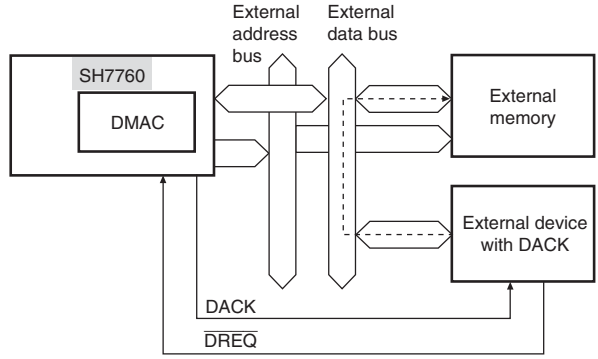
Bit	Bit Name	Initial Value	R/W	Description
3	CHSET	0	R/W	Channel Setting Since the internal state of the acceptance unit for the corresponding channel external and DMABRG requests are cleared when 1 is written to this bit in DMABRG mode, write 1 to this bit when setting up the corresponding channel. Note, however, that this bit always reads out as 0. Note: This operation is invalid in external request 2-channel mode.

11.4.4 Types of DMA Transfer

Figure 11.6 Data Flow in Single Address Mode

432

Figure amended



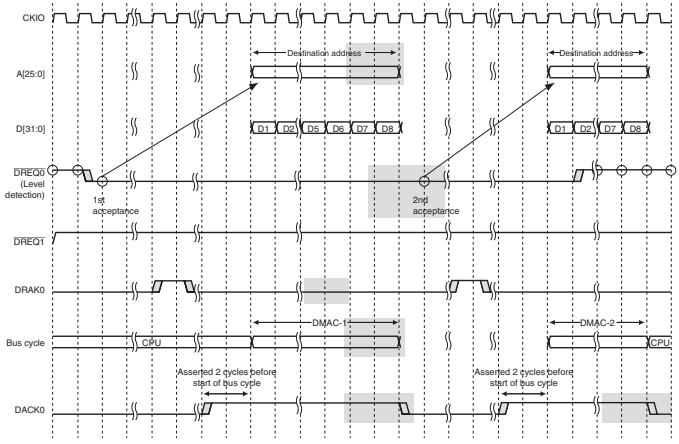
- - - ► : Data flow

11.4.5 Number of Bus Cycles and DREQ Pin Sampling Timing

Figure 11.30 Single Address Mode/Burst Mode in DMABRG Mode External Device → External Bus/DREQ (Level Detection)/32 Byte Block Transfer (Bus Width: 32 bits, SDRAM: row hit write)

460

Figure amended



11.4.6 Ending DMA Transfer

(3) Notes on Transfer End

463

Description amended

- External requests

  See item 5 in External Request Acceptance Conditions in section 11.4.2, DMA Transfer Requests (2) External Request Mode.

Reset

Do not access the registers of the HAC, SSI, USB, LCDC, and DMAC (except for DMAPCR) while the BRGRST bit is 1. Operation is not guaranteed when these registers are accessed in this state.

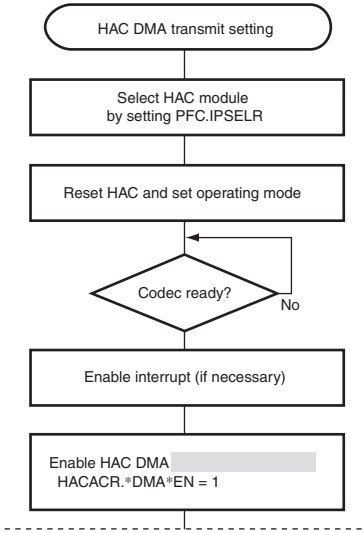
Note: * Make sure to write 1 to the CHSET bit in CHCR0 before re-specifying the DMAC registers in the case of DMAC reactivation (DMA transfer will be resumed).

11.6.5 DMA Audio Transmit Operation

Figure 11.32 Example of HAC DMA Transfer Operation Flow

471

Figure amended



11.6.11 LCDC DMA Transfer

Figure 11.38 Example of LCDC Data Transfer Flow

478

Description amended

[1] Set DMAOR, DMARCR, and DMARSRA so that DMABRG can be used.

12.3 Clock Operating Modes

Table 12.2 Clock Operating Modes

491

Table amended

Clock Operating Mode	Pin Combination					Frequency (vs. Input Clock)			FRQCR Initial Value
	MD2	MD1	MD0	PLL1	PLL2	CPU Clock	Bus Clock	Peripheral Clock	
0	0	0	0	On (×12)	On	12	3	3	H'0E1A
1	0	0	1	On (×12)	On	12	3/2	3/2	H'0E2C
2	0	1	0	On (×6)	On	6	2	1	H'0E13
3	0	1	1	On (×12)	On	12	4	2	H'0E13
4	1	0	0	On (×6)	On	6	3	3/2	H'0E0A
6	1	1	0	Off (×6)	Off	1	1/2	1/2	H'0808

14.4 Usage Note

531

Newly added

Table 15.2 Register Configuration (2)

Ch.	Register Name	Abbrev.	Power-on Reset by RESET Pin/ WDT/H-UDI	Manual Reset by RESET Pin/WDT/ Multiple Exception	Sleep by Sleep Instruction/ Deep Sleep		by Hardware	by Software/ Each Module
Common	Timer start register	TSTR	H'00	H'00	Retained	*		H'00

15.6.3 External Clock Frequency — 545

Description amended

Ensure that the external clock (TCLK) frequency for each channel does not exceed Pck/8.

16.3.4 IRQ Status Register (CMTIRQS) — 557

Table amended

Bit	Bit Name	Initial Value	R/W	Description
11	IO3	0	R/WC0*	Channel 3 to 0 Interrupt Overflow
10	IO2	0	R/WC0*	A bit for each channel indicates if the up-counters or updown-counters have wrapped i.e. overflowed from H'FFFF to H'0000 or underflowed from H'0000 to H'FFFF.
9	IO1	0	R/WC0*	
8	IO0	0	R/WC0*	0: The count has not overflowed or underflowed 1: The count has overflowed or underflowed
7	IC3	0	R/WC0*	Channel 3 to 0 Interrupt Compare
6	IC2	0	R/WC0*	A bit for each channel indicates whether in timer mode, the free-running timer has become equal to the channel times.
5	IC1	0	R/WC0*	
4	IC0	0	R/WC0*	0: Timer has not become equal to the channel time value 1: Timer has become equal to the channel time value
3	IE3	0	R/WC0*	Channel 3 to 0 Interrupt Edge
2	IE2	0	R/WC0*	A bit for each channel indicates whether an edge that wil cause an action (active edge) has been defected.
1	IE1	0	R/WC0*	
0	IE0	0	R/WC0*	0: Channel 3 to 0 has not received an active edge 1: Channel 3 to 0 has received an active edge

Note added

Note: * Writing 0 to clear the bit to 0 is allowed.

16.4.4 16-Bit Timer: Input Capture — 562

Description amended

The counters will be cleared to H'0000 by disabling the timer enable bits.

16.4.5 16-Bit Timer: Output Compare — 563

Description amended

The counters will be cleared to H'0000 by disabling the timer enable bits.

16.4.7 Counter: Up-Counter with Capture — 565

Description amended

The counter will be cleared to H'0000 by disabling the timer enable bit.

Descriptions

Table 17.2 Register Configuration (1)

Ch.	Register Name	Abbrev.	R/W	P4 Address	Area 7 Address	Size	Sync Clock
0	Transmit FIFO data count register 0	SCTFDR0	R	H'FE60 001C	H'1E60 001C	16	Pck
	Receive FIFO data count register 0	SCRFDR0	R	H'FE60 0020	H'1E60 0020	16	Pck
1	Transmit FIFO data count register 1	SCTFDR1	R	H'FE61 001C	H'1E61 001C	16	Pck
	Receive FIFO data count register 1	SCRFDR1	R	H'FE61 0020	H'1E61 0020	16	Pck

575

Ch.	Register Name	Abbrev.	R/W	P4 Address	Area 7 Address	Size	Sync Clock
2	Transmit FIFO data count register 2	SCTFDR2	R	H'FE62 001C	H'1E62 001C	16	Pck
	Receive FIFO data count register 2	SCRFDR2	R	H'FE62 0020	H'1E62 0020	16	Pck

17.3.14 Serial Error Register (SCRER)

601

Figure and table amended

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	PER6	PER5	PER4	PER3	PER2	PER1	PER0	-	FER6	FER5	FER4	FER3	FER2	FER1	FER0
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	Bit Name	Initial Value	R/W	Description
15	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
14	PER6	0	R	Number of Parity Errors These bits indicate the number of data bytes in which a parity error occurred in the receive data stored in SCFRDR. After the ER bit in SCFSR is set, the value indicated by bits PER6 to PER0 is the number of data bytes in which a parity error occurred. If all 128 bytes of receive data in SCFRDR have parity errors, the value indicated by bits PER6 to PER0 will be 0.
13	PER5	0	R	
12	PER4	0	R	
11	PER3	0	R	
10	PER2	0	R	
9	PER1	0	R	
8	PER0	0	R	
7	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
6	FER6	0	R	Number of Framing Errors These bits indicate the number of data bytes in which a framing error occurred in the receive data stored in SCFRDR. After the ER bit in SCFSR is set, the value indicated by bits FER6 to FER0 is the number of data bytes in which a framing error occurred. If all 128 bytes of receive data in SCFRDR have framing errors, the value indicated by bits FER6 to FER0 will be 0.
5	FER5	0	R	
4	FER4	0	R	
3	FER3	0	R	
2	FER2	0	R	
1	FER1	0	R	
0	FER0	0	R	

17.6 Usage Notes

(7) Notes on the TEND Flag

626

Description added

19.3.1 Slave Control Register (ICSCR)

667

Table amended

Bit	Bit Name	Initial Value	R/W	Description
2	SIE	0	R/W	Slave Interface Enable Ensure to set this bit to 1 to have the slave to operate. If this bit is low, the slave interface is reset.

Register (ICMSR)

Bit	Bit Name	Initial Value	R/W	Description
1	FSB	0	R/W	Force Stop onto the Bus Setting FSB to 1 will have the master issue a stop onto the bus at the end of the current transfer. If ESG is also 1, the master immediately issues a start and begins transmitting a new data packet. If ESG is 0, the master enters the idle state. Set FSB to 1 when the TEND flag is set to 1 during transmission in the FIFO buffer mode, or when the RDF flag is set to 1 during reception in the FIFO buffer mode. In single buffer mode, when the last bit of a byte is transmitted/received, the I²C module latches the FSB value and enters the STOP state. Therefore, to stop the transfer after a specified number of bytes are transferred, the FSB bit must be set to 1 before the last byte is transferred. Note: Check section 19.7, Usage Notes, when using this bit.

19.3.6 Master Status Register (ICMSR) 676

Table amended

Bit	Bit Name	Initial Value	R/W	Description
2	MDT	0	R/W*	Master Data Transmission The master has transmitted a byte of data to the slave on the bus. This status bit becomes 1 after the falling edge of SCL during the last data bit transmission.
0	MAT	0	R/W*	Master Address Transmission The master has transmitted the slave address byte of a data packet. This bit becomes 1 after the falling edge of SCL during the output of the ack bit which is sent after an address.

19.3.14 Receive FIFO Data Count Register (ICRFDR) 687

Description amended

H'0000 0000 indicates that ICRXD contains no receive data, while H'0000 0010 indicates that it holds 16 bytes of receive data.

19.3.15 Transmit FIFO Data Count Register (ICTFDR) 687

Description amended

H'0000 0000 indicates that ICTXD contains no transmit data, while H'0000 0010 indicates that it holds 16 bytes of transmit data.

(Single Buffer Mode)		This section describes the transmit procedure and operations in master transmit mode. Figures 19.9 to 19.11 are the timing charts in master transmit mode. Setting the MDBS bit to 1 in the master control register has the I2C module enter single buffer mode. 1. For initial setting, set clock control bits in the clock control register and interrupt generation bits in the master interrupt enable register, according to the slave address, transmit data, and the transmit speed. Since the slave mode is also required even when only the master mode is used, set the device address to the slave address register. Do not modify either the master control register MDBS bit or the slave control register SDBS bit during operation. Incorrect operation may occur if these bits are changed during operation.
19.5.1 Master Transmitter Operation (FIFO Buffer Mode)	697, 698	Description amended Notes: 2. FSB must be set to 1 at least one SCL clock cycle after the transmit FIFO data empty flag (TDFE) is set, and within 9 SCL clock cycles following that flag being set. (See figure 19.15.)* For example, to transfer 3 bytes of data, after 3 bytes of data have been written to the FIFO and transferred, verify that TDFE = 1 either by poling or with the transmit FIFO data empty (TXI) interrupt and then set FSB after the first SCL clock cycle and before the ninth SCL clock cycle completes following TDFE being set to 1. Note that care is required concerning the timing with which FSB is set in application system, in particular, check the interrupt response and handling times carefully. Note: * If FSB is not set with this timing, the stop condition (P) may not be issued correctly.
19.6.1 Master Transmitter (Single Buffer Mode) (4) Monitor the progress of data byte transmission:	699	Description amended (a) Wait for a master event (the MDE bit in the master status register). (b) Load the next data byte into the transmit data register.* Note: * There is no need to observe the limitation that "execution must continue until the first data byte has been output" in this case.
19.6.4 Master Transmitter (FIFO Buffer Mode)	702	Description amended 8. Set FSB to 1 after 1 SCL clock period has completed and before 9 SCL clock periods have completed after TDFE was set to 1. (See figure 19.15)

Receive (I²C Bus Mode)

Verify that RDF is 0. (If RDF is 1, read the data from ICRXD and then clear RDF to 0.)

4. ICMCR=H'0000 0009 (set ESG) //ESG=1, MIE=1, MDBS=0. (At this point, the slave address is output onto I_2C bus.)
5. Wait for MAT, and clear ESG.
6. Wait for RDF, and read the data received from ICRXD.
   ICFSR=H'0000 0000 (clear the flag)
   (Repeat)
7. Wait for RDF, and set FSB to 1.
8. Wait for one bit period after setting RDF to 1 and read the data received from ICRXD.

19.7.1 Restriction 1	703, 704	Description replaced
19.7.2 Restriction 2	704, 705	Description replaced
20.3 Register Descriptions Table 20.2 Register Configuration (1)	709	Table amended

Ch.	Register Name	Abbrev.	R/W	P4 Address	Area 7 Address	Size	Sync Clock
0	Transmit data register 0	SSITDR0	R/W	H'FE68 0008	H'1E68 0008	32	Pck
1	Transmit data register 1	SSITDR1	R/W	H'FE69 0008	H'1E69 0008	32	Pck

20.3.1 Control Register (SSICR) — 711 — Table amended

Bit	Bit Name	Initial Value	R/W	Description
15	SCKD	0	R/W	Serial Bit Clock Direction 0: Serial clock input, slave mode 1: Serial clock output, master mode Note: In uncompressed mode (CPEN = 0), (SCKD, SWSD) may only be set to (0, 0) or (1, 1).
14	SWSD	0	R/W	Serial WS Direction 0: Serial word select input, slave mode 1: Serial word select output, master mode Note: In uncompressed mode (CPEN = 0), (SCKD, SWSD) may only be set to (0, 0) or (1, 1).

713

Bit	Bit Name	Initial Value	R/W	Description
11	SPDP	0	R/W	Serial Padding Polarity This bit is ignored if CPEN = 1. 0: Padding bits are low 1: Padding bits are high Note: When MUEN = 1, the padding bits will be at the low level. (The muting function takes priority.)

Bit	Bit Name	Initial Value	R/W	Description
8	DEL	0	R/W	Serial Data Delay 0: 1 clock cycle delay between SSI_WS and SSI_SDATA 1: No delay between SSI_WS and SSI_SDATA This bit must be set to 1 when CPEN = 1. A one-clock cycle delay is not supported when the SSI module is configured to be a slave transmitter (SWSD = 0 and TRMD = 1). In this situation, this bit should be set to 0.

715

Bit	Bit Name	Initial Value	R/W	Description
2	CPEN	0	R/W	Compressed Mode Enable 0: Compressed mode disabled 1: Compressed mode enabled Note: In compressed mode (CPEN = 1), only use operations other than slave transmitter (SWSD = 0 and TRMD = 1).

20.3.2 Status Register (SSISR) 720

Table amended

Bit	Bit Name	Initial Value	R/W	Description
0	IDST	1	R	Idle Mode Status Flag Indicates that the serial bus activity has ceased. This bit is cleared if EN = 1 and the Serial Bus is currently active. This bit can be set to 1 automatically under the following conditions. SSI = Serial bus master transmitter (SWSD = 1 and TRMD = 1): This bit is set to 1 if the EN bit is cleared and the current system word is completed. It can also be set to 1 when the EN bit has been cleared and the data that has been written to SSITDR is output on the serial data input/output pin (SSI_SDATA), j.e., the serial data of the system word length is output. SSI = Serial bus master receiver (SWSD = 1 and TRMD = 0): This bit is set to 1 if the EN bit is cleared and the current system word is completed. SSI = Slave transmitter/ receiver (SWSD = 0): This bit is set to 1 if the EN bit is cleared and the current system word is completed. To terminate the transfer, clear SSICR.EN to 0 and continue to input the WS signal until SSICR.IDST becomes 1. Note: If the external device stops the serial bus clock before the current system word is completed then this bit will never be set.

20.3.3 Transmit Data Register (SSITDR) 721

Figure amended

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	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
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

Table 20.3 Bus Formats of SSI Module

Bus Format	TPMD	CPEN	SCKD	SWSD	EN	MUEN	DIEN	IIEN	OIEN	UIEN	DEL	PDTA	SDTA	SPDP	SWSP	SCKP	SWL[2:0]	DWL[2:0]	CHNL[1:0]
Non-Compressed Slave Receiver	0	0	0	0	Control bits						Configuration bits								
Non-Compressed Slave Transmitter	1	0	0	0															
Non-Compressed Master Receiver	0	0	1	1															
Non-Compressed Master Transmitter	1	0	1	1															
Compressed Slave Receiver	0	1	0/1	0	Control bits						1	Ignored			Configu-ration bits		Ignored		
Compressed Slave Transmitter	Cannot be used																		
Compressed Master Receiver	0	1	0/1	1	Control bits						1	Ignored			Configu-ration bits		Ignored		
Compressed Master Transmitter	1	1	0/1	1							1								

20.4.2 Non-Compressed Modes Figure 20.6 Multichannel Format (4 Channels, No Padding)	727	Figure amended SCKP = 0, SWSP = 0, DEL = 1, CHNL = 01, SPDP = don't care, SDTA = don't care…
Figure 20.7 Multichannel Format (6 Channels with High Padding)		Figure amended SCKP = 0, SWSP = 0, DEL = 1, CHNL = 10, SPDP = 1,…
Figure 20.8 Multichannel Format (8 Channels, with Padding Bits First, Followed by Serial Data, with Padding)	728	Figure amended SCKP = 0, SWSP = 0, DEL = 1, CHNL = 11, SPDP = 0, …
20.4.3 Compressed Modes Figure 20.18 Compressed Data Format, Master Transmitter, Burst Mode Disabled	732	Figure amended TRMD = 1, CPEN = 1, SCKD = 1, SWSD = 1, SWSP = 0, BREN = 0
Figure 20.19 Compressed Data Format, Master Transmitter, and Burst Mode Enabled		Figure amended TRMD = 1, CPEN = 1, SCKD = 1, SWSD = 1, SWSP = 0, BREN = 1
(2) Slave Transmitter	733	Description amended This mode cannot be used.

Stopping USB Module Slave Mode Operation		
Section 21 USB Host Module (USB)	743	Description amended The USB Host Controller module supports Open Host Controller Interface (Open HCI) Specification*2 for the Universal Serial Bus (USB) as well as the Universal Serial Bus specification ver.1.1*1.
21.1 Features	743	Note added • Supports the register set compliant with the Open Host Controller Interface (Open HCI) Specification Release 1.0*2 • Compatible with the Universal Serial Bus Ver.1.1*1 Specifications. Notes: 1. Moreover, refer to the USB Host Electrical Characteristics section for the electrical characteristics of USB Host. 2. Part of registers is not supported. For details, see section 21.3, Register Descriptions and section 21.6, Restrictions on HcRhDescriptorA.
21.4 Memory	782	Description amended When USB HC (Host controller) reads the data at the following 4 Kbyte boundary address in the USB shared memory, the wrong data may be returned to USB HC. The wrong access address whish USB HC (Host controller) is using. H'0000 0FFC—H'0000 0FFF H'0000 1FFC—H'0000 1FFF
21.5.1 Storage Format of Transfer Data	783	Description amended The USB host assumes that data will be stored sequentially, in little endian order, from low to high addresses, regardless of the CPU endian setting. Figure 21.3 shows USB read operation.

of Transfer Data		The data in memory mentioned above and the data read by the USB host must always correspond. When reading data from external memory, the USB host always reads data in longword units regardless of the endian setting. The USB host assumes that read data is in the little endian order, that is, the byte order that places the first byte in the lowest address and the last byte in the highest address. That is, during operation of this IC, data must be stored sequentially in longword units in little endian order from low to high addresses, regardless of whether the endian setting is little endian or big endian. An example of failure is shown in figure 21.4. In this example, USB host controller does not receive #H'12, which is the expected transfer data. The USB host controller, when writing, stores data sequentially starting with the low order bits in memory in little endian order, so that the data is read/written correctly from both sides regardless of the endian setting. That is, the data is always aligned in little endian format.

22.5.4 Interrupt Request Register (CANIRR)

818 Table amended

Bit	Bit Name	Initial Value	R/W	Description
15	—	0	R	Reserved The write value should always be 0. The read value is not guaranteed.

819 Table amended

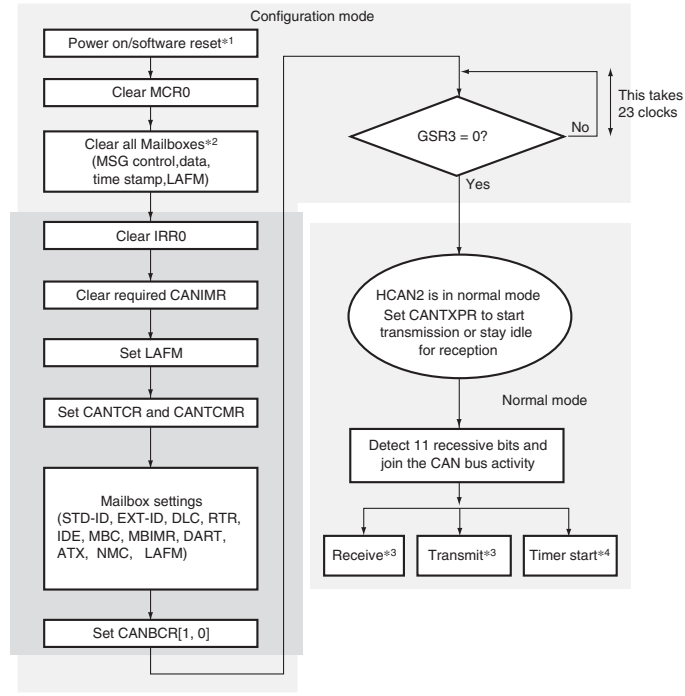
Bit	Bit Name	Initial Value	R/W	Description
11, 10	—	All 0	R	Reserved The write value should always be 0. The read value is not guaranteed.

22.5.5 Interrupt Mask Register (CANIMR)

824 Table amended

Bit	Bit Name	Initial Value	R/W	Description
15, 11, 10	—	All 1	R	Reserved The write value should always be 1. The read value is not guaranteed.

Figure 22.5 Reset Sequence

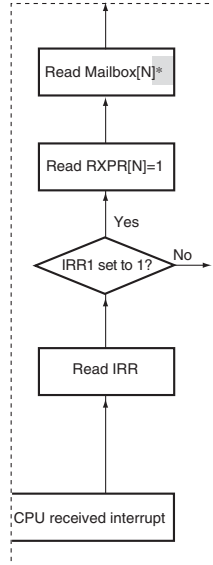


22.6.4 Message Reception Sequence

845

Figure amended

Figure 22.8 Message Receive Sequence



23.4.5 HSPI Software Reset	864	Description deleted The data transmission after the HSPI software reset should protect transmitting and receiving protocol of HSPI, and please perform it from the first. A guarantee of operation is not offered other than it.
24.1 Features	867	Description deleted For details of multiplexed pins, refer to table 24.1, Multiplexed Pins Controlled by Port Control Registers. For pin multiplexing in this LSI, refer to table 1.3 and 1.4. By default, each pin of the ports is pulled up.
Table 24.1 Multiplexed Pins Controlled by Port Control Registers	868	Table amended

Pin Name	Port	GPIO	MFI Mode (MD7=0)	LCD Mode (MD7 = 1)	Register Setting
VCPWC/$\overline{\text{IRQ4}}$	E	PTE1 input/output	IRQ	LCDC	
VEPWC/$\overline{\text{IRQ5}}$	E	PTE0 input/output	IRQ	LCDC	

24.2.21 GPIO Interrupt Control Register (GPIOIC)	892	Table amended		

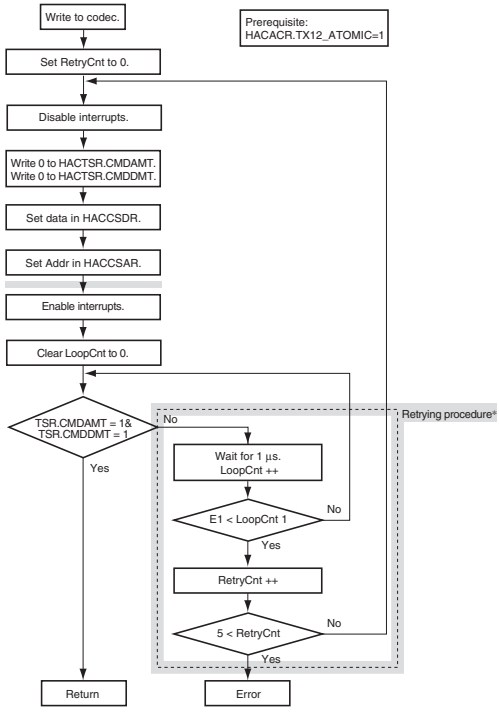
Bit Name	Pin	Pin Name	Port	Interrupt Type
PTIREN15	T2	MFI-RW/LCD_FLM	PTE2	GPIO interrupt
PTIREN14	T1	MFI-RS/LCD_M_DISP	PTE3	
PTIREN13	R2	MFI-MD/LCD_CL2	PTE4	
PTIREN12	R1	MFI-E/LCD_CL1	PTE5	
PTIREN11	P2	$\overline{\text{MFI-CS}}$/LCD_DON	PTE6	
PTIREN10	P1	$\overline{\text{MFI-INT}}$/LCD_CLK	PTE7	
PTIREN9	A13	$\overline{\text{SCIF2\_CTS}}$	PTH6	
STBRT8	A14	$\overline{\text{SCIF2\_RTS}}$	PTH5	
STBRT7	C18	USB_PENC	PTH1	
STBRT6	J20	$\overline{\text{FCE}}$/AUDATA[3]	PTK7	
STBIRQ5	E2	VEPWC/$\overline{\text{IRQ5}}$	PTE0	IRQ interrupt
STBIRQ4	E1	VCPWC/$\overline{\text{IRQ4}}$	PTE1	
STBIRL3	M19	$\overline{\text{IRL3}}$		IRL interrupt
STBIRL2	M20	$\overline{\text{IRL2}}$		
STBIRL1	N19	$\overline{\text{IRL1}}$		
STBIRL0	N20	$\overline{\text{IRL0}}$		

24.2.35 Peripheral Module Select Register (IPSELR)	905	Table amended		

Bit	Bit Name	Initial Value	R/W	Description
11	IPSELR11	0	R/W	Out of the modules SSI[1]/[0] and HAC[1]/[0], select the one using the pins SSI0_SCK/HAC_SD_IN0/$\overline{\text{BS2}}$, SSI0_WS/HAC_SYNC0, SSI0_SDATA/HAC_SD_OUT0, SSI1_SCK/HAC_SD_IN1, SSI1_SDATA/HAC_SD_OUT1, SSI1_WS/HAC_SYNC1, HAC_BIT_CLK0, and HAC_BIT_CLK1 00: SSI[0], SSI[1] 01: HAC[0]. SSI[1] 10:Setting prohibited 11:HAC[0], HAC[1]
10	IPSELR10	0	R/W	

Register (HACTCR)					
	31	CMDAMT	1	R/W	Command Address Empty 0: CSAR Tx buffer contains untransmitted data. 1: CSAR Tx buffer is empty and ready to store data.*1 For details of HAC initialization steps, see the operational flow in 25.5.5.

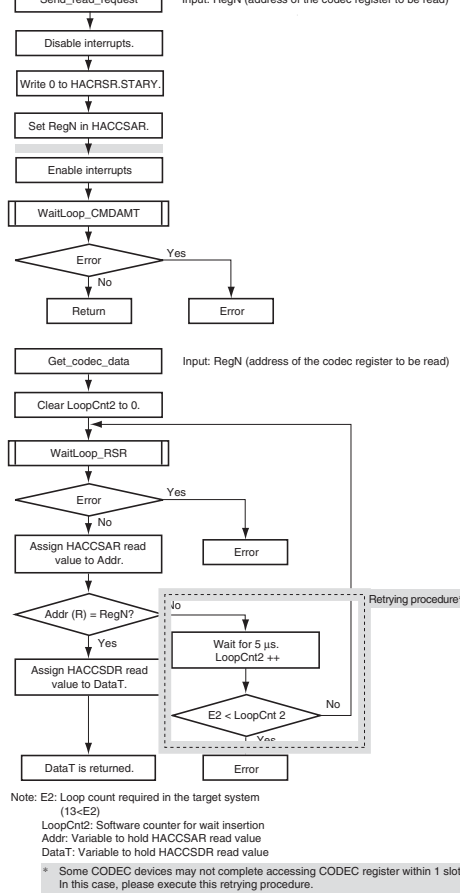
25.5.5 Restrictions Related to HACTCR.CMDAMT	—	Deleted
25.5.5 Initialization Sequence Figure 25.4 Sample Flowchart for Off-Chip Codec Register Write	930	Figure amended



Notes: E1: Loop count required in the target system
(21<E1<1000)
Input: Addr: Address of codec register to be written to
Data: Data to be written to codec register
RetryCnt: Software counter for error detection
LoopCnt: Software counter for wait insertion

* Some CODEC devices may not complete accessing CODEC register within 1 slot time. In this case, please execute this retrying procedure.

Figure 25.6 Sample Flowchart for Off-Chip Codec Register Read (cont)



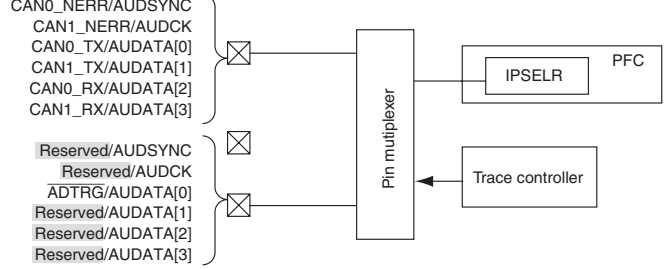
Note: E2: Loop count required in the target system (13<E2)
LoopCnt2: Software counter for wait insertion
Addr: Variable to hold HACCSAR read value
DataT: Variable to hold HACCSDR read value

* Some CODEC devices may not complete accessing CODEC register within 1 slot time. In this case, please execute this retrying procedure.

26.8 Usage Note	994	Newly added
Section 28 User Debug Interface (H-UDI)	1019	The H-UDI consists of six pins: TCK, TMS, TDI, TDO, TRST, and ASEBRK/BRKACK. The pin functions and serial communication protocol comprise a subset of the JTAG standard.

Debug Interface (H-UDI)

Figure 28.1 H-UDI Block Diagram



28.1 Input/Output Pins 1022

Table 28.1 Pin Configuration

Figure amended

Pin Name	Abbreviation	I/O	Function	When Not in Use
Emulator	AUDSYNC/ AUDCK/ AUDATA[3] to AUDATA[0]	Output	Emulator Connection When bit 13 of IPSELR in the PFC is set to 1, signals are output to the following pins. CAN0_TX/AUDATA[0] CAN1_TX/AUDATA[1] CAN0_RX/AUDATA[2] CAN1_RX/AUDATA[3] CAN0_NERR/AUDCK CAN1_NERR/AUDSYNC When bit 12 of IPSELR in the PFC is set to 1, signals are output to the following pins. ADTRG/AUDATA[0] Reserved/AUDATA[1] Reserved/AUDATA[2] Reserved/AUDATA[3] Reserved/AUDCK Reserved/AUDSYNC	Open*4

29.1 Features 1039

Description amended

- A/D conversion can be externally triggered (except in multi mode)

29.3.2 A/D Control/Status Register (ADCSR) 1045

Table amended

Bit	Bit Name	Initial Value	R/W	Description
13	ADST	0	R/W	A/D Start Starts or stops A/D conversion. This bit remains set to 1 during A/D conversion. It can also be set to 1 by external trigger input (ADTRG) pin (except in multi mode). 0: A/D conversion is stopped 1:

1046

Bit	Bit Name	Initial Value	R/W	Description
11	TRGE1	0	R/W	Trigger Enable External trigger input permits or prohibits A/D conversion. These bits must be set while conversion is stopped. 00: When an external trigger is input, A/D conversion does not start 01: Setting prohibited 10: Setting prohibited 11: A/D conversion starts at the falling edge of an input signal from the external trigger input pin (ADTRG) (except in multi mode) Note: Clear bits TRGE1 and TRGE0 to 0 before switching the trigger signal.
10	TRGE0	0	R/W	

1. When the ADST bit in ADCSR is set to 1 by software, A/D conversion starts with the first channel (AN0).

29.4.5 External Trigger Input Timing	1055	Description amended A/D conversion can also be started by an external trigger input (except in multi mode)….
29.7.5 Notice of the DMA transmission of A/D converter	1059	Newly added
29.7.6 Notice of Scan mode and Multi mode of A/D conversion	1060, 1061	Newly added
29.7.7 Notice of Multi mode of A/D conversion	1061	Newly added
30.3 Register Configuration Table 30.2 Register Configuration (1)	1065	Table amended

Register Name	Abbrev.	R/W	P4 Address	Area 7 Address	Size	Sync Clock	Modification During Display*2
LCDC input clock register	LDICKR	R/W	H'FE30 0C00	H'1E30 0C00	16	Pck	No
LCDC module type register	LDMTR	R/W	H'FE30 0C02	H'1E30 0C02	16	Pck	No
LCDC data format register	LDDFR	R/W	H'FE30 0C04	H'1E30 0C04	16	Pck	No
LCDC scan mode register	LDSMR	R/W	H'FE30 0C06	H'1E30 0C06	16	Pck	No
LCDC display start address register - upper	LDSARU	R/W	H'FE30 0C08	H'1E30 0C08	32	Pck	Yes
LCDC display start address register - lower	LDSARL	R/W	H'FE30 0C0C	H'1E30 0C0C	32	Pck	Yes
LCDC display line address offset register	LDLAOR	R/W	H'FE30 0C10	H'1E30 0C10	16	Pck	Yes
LCDC palette control register	LDPALCR	R/W	H'FE30 0C12	H'1E30 0C12	16	Pck	No
Palette data registers 00 to FF	LDPR00 to FF*1	R/W	H'FE30 0800	H'1E30 0800	32	Pck	Yes
LCDC horizontal character number register	LDHCNR	R/W	H'FE30 0C14	H'1E30 0C14	16	Pck	No
LCDC horizontal synchronization signal register	LDHSYNR	R/W	H'FE30 0C16	H'1E30 0C16	16	Pck	No
LCDC vertical display line number register	LDVDLNR	R/W	H'FE30 0C18	H'1E30 0C18	16	Pck	No
LCDC vertical total line number register	LDVTLNR	R/W	H'FE30 0C1A	H'1E30 0C1A	16	Pck	No
LCDC vertical synchronization signal register	LDVSYNR	R/W	H'FE30 0C1C	H'1E30 0C1C	16	Pck	No
LCDC AC modulation signal toggle line number register	LDACLNR	R/W	H'FE30 0C1E	H'1E30 0C1E	16	Pck	Yes
LCDC interrupt control register	LDINTR	R/W	H'FE30 0C20	H'1E30 0C20	16	Pck	No
LCDC power management mode register	LDPMMR	R/W	H'FE30 0C24	H'1E30 0C24	16	Pck	No

1066

Register Name	Abbrev.	R/W	P4 Address	Area 7 Address	Size	Sync Clock	Modification During Display*2
LCDC power supply sequence period register	LDPSPR	R/W	H'FE30 0C26	H'1E30 0C26	16	Pck	No
LCDC control register	LDCNTR	R/W	H'FE30 0C28	H'1E30 0C28	16	Pck	No

Table 30.2 Register Configuration (1)

Notes: 1. There are 256 registers: LDPR00, LDPR01, ……, LDPRFF. These registers are allocated to H'FE30 0800, H'FE30 0804, ……, H'FE30 0BFC.

2. Modification During Display - Yes: Modification during display on the LCD allowed, No: Modification during display on the LCD not allowed.

30.3.10 LCDC Horizontal Character Number Register (LDHCNR) — 1080

Note amended

Note: The values set in HDCN and HTCN must satisfy the relationship of HTCN ≥ HDCN + 3.

30.3.17 LCDC Power Management Mode Register (LDPMMR) — 1088

Table amended

Bit	Bit Name	Initial Value	R/W	Description
4	DONE	1	R/W	LCD_DON Pin Enable Sets whether or not to enable a power-supply control sequence using the LCD_DON pin. 0: Disabled: LCD_DON pin output is masked and fixed low 1: Enabled: LCD_DON pin output is asserted and negated according to the power-on or power-off sequence

---

30.4.6 Power Supply Control Sequence Processing — 1099

Figure 30.4 Power-Supply Control Sequence and States of the LCD Module

Figure amended

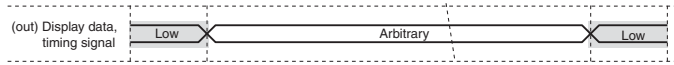


Figure 30.5 Power-Supply Control Sequence and States of the LCD Module

Figure amended

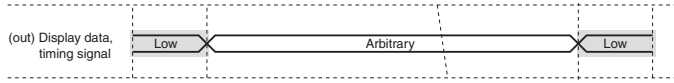


Figure 30.6 Power-Supply Control Sequence and States of the LCD Module — 1100

Figure amended

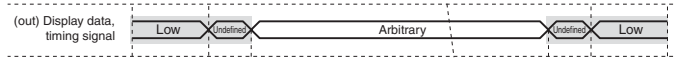
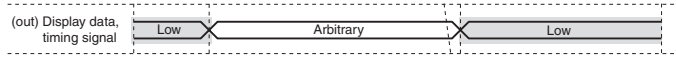


Figure 30.7 Power-Supply Control Sequence and States of the LCD Module

Figure amended



30.6 Usage Notes — 1117

Newly added

32.2 Register Bits — 1178

Table amended

Abbrev.	Bit 31/ 23/15/7	Bit 30/ 22/14/6	Bit 29/ 21/13/5	Bit 28/ 20/12/4	Bit 27/ 19/11/3	Bit 26/ 18/10/2	Bit 25/ 17/9/ 1	Bit 24/ 16/8/0	Module
SCRER0	—	PER6	PER5	PER4	PER3	PER2	PER1	PER0	
	—	FER6	FER5	FER4	FER3	FER2	FER1	FER0	

Abbrev.	Bit 31/ 23/15/7	Bit 30/ 22/14/6	Bit 29/ 21/13/5	Bit 28/ 20/12/4	Bit 27/ 19/11/3	Bit 26/ 18/10/2	Bit 25/ 17/9/ 1	Bit 24/ 16/8/0	Module
SCRER1	—	PER6	PER5	PER4	PER3	PER2	PER1	PER0	
	—	FER6	FER5	FER4	FER3	FER2	FER1	FER0	

1180

Abbrev.	Bit 31/ 23/15/7	Bit 30/ 22/14/6	Bit 29/ 21/13/5	Bit 28/ 20/12/4	Bit 27/ 19/11/3	Bit 26/ 18/10/2	Bit 25/ 17/9/ 1	Bit 24/ 16/8/0	Module
SCRER2	—	PER6	PER5	PER4	PER3	PER2	PER1	PER0	
	—	FER6	FER5	FER4	FER3	FER2	FER1	FER0	

33.1 Absolute Maximum Ratings

Table 33.1 Absolute Maximum Ratings

1225

Table amended

Item	Symbol	Value	Unit
I/O, CPG, ADC power supply voltage	V_{DDQ} V_{DD-CPG} AV_{CC-ADC}	0.3 to 4.6	V
Internal power supply voltage	V_{DD} $V_{DD-PLL1/2/3}$	0.3 to 2.1	V
Input voltage	V_{in}	–0.3 to V_{DDQ}+0.3, –0.5 to 5.5*1	V
Operating temperature*2	T_{opr}	–20 to 75/–40 to 85	°C
Storage temperature	T_{stg}	–55 to 125	°C

Note added

Notes: 1. The LSI may be permanently damaged if the maximum ratings are exceeded.

2. The LSI may be permanently damaged if any of the VSS pins are not connected to GND.

3. For the powering-on and powering-off sequence, see Appendix F, Power-On and Power-Off Procedures.

*1 For I2C0_SCL, I2C0_SDA, I2C1_SCL, I2C1_SDA.

*2 See table G.1.

33.2 DC Characteristics

Table 33.2 DC Characteristics (T_a = –20 to 75°C/–40 to 85°C)

1226

Table amended

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Input voltage	I2C1_SCL, I2C1_SDA I2C0_SCL, I2C0_SDA	V_{IH}	V_{DDQ} × 0.7	—	5.5		
	Other input pins		2.2	—	V_{DDQ} +0.3		

1227

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Input voltage	I2C1_SCL, I2C1_SDA I2C0_SCL, I2C0_SDA	V_{IL}	–0.5	—	V_{DDQ} × 0.3	V	
Output voltage	All output pins*	V_{OH}	V_{DDQ} ×0.8			V	

Characteristics Table 33.2 DC Characteristics (T_a = –20 to 75°C/–40 to 85°C)		Notes: 1. Regardless of whether or not the PLL is used, please supply the same voltage to V_{DDQ}, AV_{CC_ADC}, $V_{DD\text{-}CPG}$, supply the same voltage $V_{DD\text{-}PLL\ 1/2/3}$ and V_{DD}, connect V_{SS}, $V_{SS\text{-}CPG}$, and $V_{DD\text{-}PLL\ 1/2/3}$ to GND. The LSI may be damage when not filling this. … * I2Cn_ SCL and I2Cn_SDA pins are removed.
33.3.1 Clock and Control Signal Timing Table 33.5 Clock and Control Signal Timing	1229	Table condition amended (V_{DDQ} = 3.0 to 3.6 V, V_{DD} = 1.5 V, T_a = –20 to 75°C/–40 to 85°C, C_L = 30 pF, PLL2 on)
33.3.2 Control Signal Timing Table 33.6 Control Signal Timing	1236	Table condition and table amended (V_{DDQ} = 3.0 to 3.6 V, V_{DD} = 1.5 V, T_a = –20 to 75°C/–40 to 85°C, C_L = 30 pF, PLL2 on)

Item	Symbol	Min.	Max.	Unit	Figure
Bus tri-state delay time to standby mode	t_{BOFF2}	—	2	t_{cyc}	33.16 (2)
Bus buffer on time	t_{BON1}	—	12	ns	33.15
Bus buffer on time from standby	t_{BON2}	—	2	t_{cyc}	33.16 (2)
STATUS 0/1 delay time	t_{STD1}	—	6	ns	33.16 (1)
	t_{STD2}	—	2	t_{cyc}	33.16 (1) (2)
	t_{STD3}	—	2	t_{cyc}	33.16 (2)

Figure 33.16 (1) Pin Drive Timing for Reset or Sleep Mode Figure 33.16 (2) Pin Drive Timing for Software Standby Mode	1237, 1238	Figure replaced
33.3.3 Bus Timing Table 33.7 Bus Timing	1239	Table condition amended (V_{DDQ} = 3.0 to 3.6 V, V_{DD} = 1.5 V, T_a = –20 to 75°C/–40 to 85°C, C_L = 30 pF, PLL2 on)
33.3.4 INTC Module Signal Timing Table 33.8 INTC Module Signal Timing	1272	Table condition amended (V_{DDQ} = 3.0 to 3.6 V, V_{DD} = 1.5 V, T_a = –20 to 75°C/–40 to 85°C, C_L = 30 pF, PLL2 on)
33.3.5 DMAC Module Signal Timing Table 33.9 DMAC Module Signal Timing	1272	Table condition amended (V_{DDQ} = 3.0 to 3.6 V, V_{DD} = 1.5 V, T_a = –20 to 75°C/–40 to 85°C, C_L = 30 pF, PLL2 on)
33.3.6 TMU Module Signal Timing Table 33.10 TMU Module Signal Timing	1273	Table condition amended (V_{DDQ} = 3.0 to 3.6 V, V_{DD} = 1.5 V, T_a = –20 to 75°C/–40 to 85°C, C_L = 30 pF, PLL2 on)

Table 33.11 SCIF Module Signal Timing

(V_{DDQ} = 3.0 to 3.6 V, V_{DD} = 1.5 V, T_a = –20 to 75°C/–40 to 85°C, C_L = 30 pF, PLL2 on)

33.3.8 H-UDI Module Signal Timing — 1275

Table 33.12 H-UDI Module Signal Timing

Table condition amended

(V_{DDQ} = 3.0 to 3.6 V, V_{DD} = 1.5 V, T_a = –20 to 75°C/–40 to 85°C, C_L = 30 pF, PLL2 on)

33.3.9 CMT Module Signal Timing — 1276

Table 33.13 CMT Module Signal Timing

Table condition added and table amended

(V_{DDQ} = 3.0 to 3.6 V, V_{DD} = 1.5 V, T_a = –20 to 75°C/–40 to 85°C, C_L = 30 pF, PLL2 on)

Item	Symbol	Min.	Max.	Unit	Figure
CMT_CTR output delay time	t_{TMD}	—	8	ns	33.58
CMT_CTR input setup time	t_{TMS}	6	—	ns	33.58
CMT_CTR input hold time	t_{TMH}	2	—	ns	33.58

33.3.10 HCAN2 Module Signal Timing — 1277

Table 33.14 HCAN2 Module Signal Timing

Table condition added and table amended

(V_{DDQ} = 3.0 to 3.6 V, V_{DD} = 1.5 V, T_a = –20 to 75°C/–40 to 85°C, C_L = 30 pF, PLL2 on)

Item	Symbol	Min.	Max.	Unit	Figure
CAN_TX output delay time	t_{CAND}	—	6	ns	33.60
CAN_RX input setup time	t_{CANS}	4	—	ns	33.60
CAN_RX input hold time	t_{CANH}	2.5	—	ns	33.60

33.3.11 GPIO Signal Timing — 1278

Table 33.15 GPIO Signal Timing

Table condition added and table amended

(V_{DDQ} = 3.0 to 3.6 V, V_{DD} = 1.5 V, T_a = –20 to 75°C/–40 to 85°C, C_L = 30 pF, PLL2 on)

Item	Symbol	Min.	Max.	Unit	Figure
GPIO output delay time	t_{IOPD}	—	9	ns	33.61
GPIO input setup time	t_{IOPS}	7	—	ns	33.61
GPIO input hold time	t_{IOPH}	5	—	ns	33.61

33.3.12 I^2C Electrical Characteristics — 1279

Table 33.17 I^2C DC Characteristics

Table condition added and table amended

(V_{DDQ} = 3.0 to 3.6 V, V_{DD} = 1.5 V, T_a = –20 to 75°C/–40 to 85°C, C_L = 30 pF, PLL2 on)

Item	symbol	Min.	Max.	Unit	Test Condition
Input Voltage	V_{IH}	$V_{DDQ} \times 0.7$	5.5	V	
	V_{IL}	–0.5	$V_{DDQ} \times 0.3$	V	

Table 33.18 I^2C Bus Interface Module Signal Timing — 1280

Table condition added and table amended

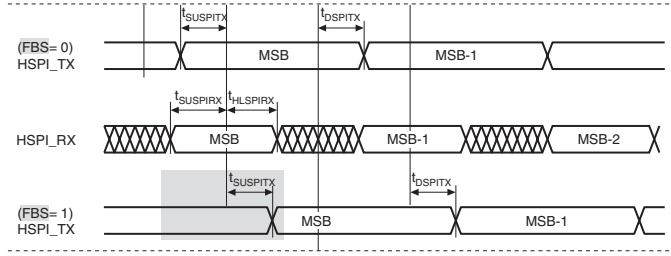
(V_{DDQ} = 3.0 to 3.6 V, V_{DD} = 1.5 V, T_a = –20 to 75°C/–40 to 85°C, C_L = 30 pF, PLL2 on)

Item	Symbol	Min.	Typ.	Max.	Unit
I2Cn_SCL frequency	t_{SCL}	—	—	400	kHz
I2Cn_SCL/I2Cn_SDA rise time	t_{SCr}	20 +0.1 cb*	—	300	ns
I2Cn_SCL/I2Cn_SDA fall time	t_{SCf}	20 +0.1 cb*	—	300	ns

Table 33.18 I²C Bus Interface Module Signal Timing		Notes: t_{Pcyc}: one Pck cycle time * Cb is the total capacity of one bus line. (max. 400 pF)
Table 33.19 I²C Schmitt characteristics	1280	Table condition added and table amended (V_{DDQ} = 3.0 to 3.6 V, V_{DD} = 1.5 V, T_a = –20 to 75°C/–40 to 85°C, C_L = 30 pF, PLL2 on)

Item	Symbol	Min.	Max.	Unit	Notes
Threshold voltage	VTT+	$V_{DDQ} \times 0.7$	—	V	Threshold voltage: L → H
	VTT–	—	$V_{DDQ} \times 0.3$	V	Threshold voltage: H → L
	ΔVTT	$V_{DDQ} \times 0.05$	—	V	reference value between VTT+ and VTT–

33.3.13 HSPI Module Signal Timing Table 33.20 HSPI Module Signal Timing	1282	Table condition added (V_{DDQ} = 3.0 to 3.6 V, V_{DD} = 1.5 V, T_a = –20 to 75°C/–40 to 85°C, C_L = 30 pF, PLL2 on)
Figure 33.64 HSPI Data Output/Input Timing	1283	Figure amended



33.3.14 USB Electrical Characteristics Table 33.21 USB DC characteristics	1284	Table condition added (V_{DDQ} = 3.0 to 3.6 V, V_{DD} = 1.5 V, T_a = –20 to 75°C/–40 to 85°C)
Table 33.22 USB AC characteristics	1285	Table condition added (V_{DDQ} = 3.0 to 3.6 V, V_{DD} = 1.5 V, T_a = –20 to 75°C/–40 to 85°C, C_L = 30 pF, PLL2 on)
33.3.15 MFI Electrical Characteristics Table 33.23 AC Characteristics of 68 Series Bus	1287	Table condition added (V_{DDQ} = 3.0 to 3.6 V, V_{DD} = 1.5 V, T_a = –20 to 75°C/–40 to 85°C, C_L = 30 pF, PLL2 on)
Table 33.24 AC Characteristics of 80 Series Bus	1288	Table condition added (V_{DDQ} = 3.0 to 3.6 V, V_{DD} = 1.5 V, T_a = –20 to 75°C/–40 to 85°C, C_L = 30 pF, PLL2 on)

Table 33.25 SIM Module Signal Timing		(V_{DDQ} = 3.0 to 3.6 V, V_{DD} = 1.5 V, T_a = –20 to 75°C/–40 to 85°C, C_L = 30 pF, PLL2 on)
33.3.17 MMCIF Module Signal Timing Table 33.26 MMCIF Module Signal Timing	1290	Table condition added (V_{DDQ} = 3.0 to 3.6 V, V_{DD} = 1.5 V, T_a = –20 to 75°C/–40 to 85°C, C_L = 30 pF, PLL2 on)
Figure 33.78 MMCIF Receive Timing (falling edge sampling)	—	Deleted
33.3.18 LCDC Module Signal Timing Table 33.27 LCDC Module Signal Timing	1292	Table condition added (V_{DDQ} = 3.0 to 3.6 V, V_{DD} = 1.5 V, T_a = –20 to 75°C/–40 to 85°C, C_L = 30 pF, PLL2 on)
33.3.19 HAC Interface Module Signal Timing Table 33.28 HAC Interface Module Signal Timing	1293	Table condition added and table amended (V_{DDQ} = 3.0 to 3.6 V, V_{DD} = 1.5 V, T_a = –20 to 75°C/–40 to 85°C, C_L = 30 pF, PLL2 on)

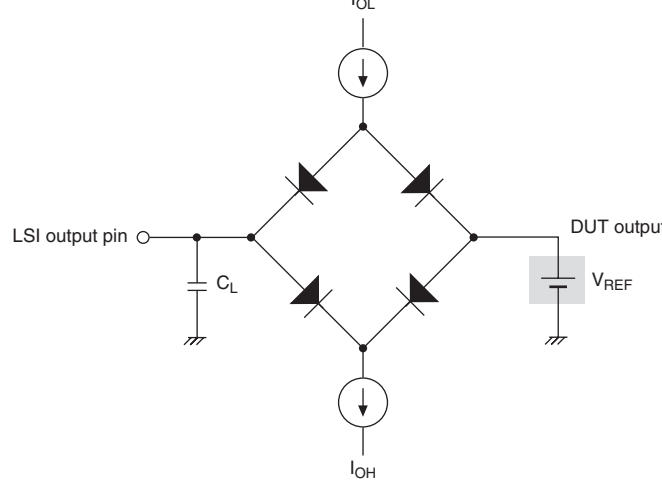
Item	Symbol	Min.	Max.	Unit	Figure
HAC_BIT_CLK input high level width	t_{ICL_HIGH}	$2 \times t_{Pcyc}$	—	ns	33.81
HAC_BIT_CLK input low level width	t_{ICL_LOW}	$2 \times t_{Pcyc}$	—	ns	33.81

33.3.20 SSI Interface Module Signal Timing Table 33.29 SSI Interface Module Signal Timing	1294	Table condition added and table amended (V_{DDQ} = 3.0 to 3.6 V, V_{DD} = 1.5 V, T_a = –20 to 75°C/–40 to 85°C, C_L = 30 pF, PLL2 on)

Item	Symbol	Min.	Max.	Unit	Notes	Figure
Output cycle time	t_{OSCK}	40	710	ns	output	33.83
Input cycle time	t_{ISCK}	80	3300	ns	input	33.83
Input high level width/Output high level width	t_{IHC}/t_{OHC}	30	—	ns	input, output	33.83
Input low level width/Output low level width	t_{ILC}/t_{OLC}	20	—	ns	input, output	33.83

33.4 A/D Converter Characteristics Table 33.30 A/D Converter Characteristics	1296	Table condition added (T_a = 25°C)

Figure 33.88 Output Load Circuit



A. Package Dimensions Figure A.1 Package Dimensions (BP-256F/BP-256FV)	1299	Figure replaced
Figure A.2 Package Dimensions (BP-256B/BP-256BV)	1300	Figure added
B. Mode Pin Settings Table B.1 Clock Operating Modes (SH7760)	1301	Table amended

Clock Operating Mode	External Pin Combination MD2	MD1	MD0	PLL1	PLL2	Frequency (vs. Input Clock) CPU Clock	Bus Clock	Peripheral Module Clock	FRQCR Initial Value
0	0	0	0	On (×12)	On	12	3	3	H'0E1A
1			1	On (×12)	On	12	3/2	3/2	H'0E2C
2		1	0	On (×6)	On	6	2	1	H'0E13
3			1	On (×12)	On	12	4	2	H'0E13
4	1	0	0	On (×6)	On	6	3	3/2	H'0E0A
6		1	0	Off (×6)	Off	1	1/2	1/2	H'0808

C.1 Pin States Table C.1 Pin States in Reset, Power-Down State, and Bus-Released State	1304	Table amended

Signal Name	Pin Name	I/O	Reset Standby Power-on	Manual	Sleep	Bus Software	Hardware	Release
MD2*9		I	I	I	I	Z	Z	I
MD3/$\overline{CE2A}$*9		I/O	PI*7	I	I/O*15	Z*16/H	Z	I
MD4/$\overline{CE2B}$*9		I/O	PI*7	I	I/O*15	Z*16/H	Z	I

Unused Pins		• When USB is not used USB_DP: Pull-down outside the LSI. USB_DM: Pull-down outside the LSI. USB_PENC: Pull-up outside the LSI. $\overline{\text{USB\_OVC}}$: Pull-up outside the LSI. UCLK: Pull-up outside the LSI. • When ADC is not used $AV_{CC\text{-}ADC}$: Power supply $AV_{SS\text{-}ADC}$: Power supply AN0 to AN3: Pull-down outside the LSI. • Hardware Standby is not used CA: Pull-up* outside the LSI Note: * To prevent unwanted effects on other pins when using external pull-up resistor, use independent pull-up resistor for individual pin. For other unused pins, it is recommended to handle the pin individually.
F. Power-on and Power-off Procedures	1325 to 1327	Description replaced
G. Product Lineup	1328	Newly added

Section 1 Overview

Section 2 Programming Model

Section 3 Floating-Point Unit (FPU)

Section 5 Pipelining

Section 6 Memory Management Unit (MMU)

Section 7 Caches

Section 8 Exceptions

Section 9 Interrupt Controller (INTC)

Section 10 Bus State Controller (BSC)

Section 11 Direct Memory Access Controller (DMAC)

Section 12 Clock Pulse Generator (CPG)

Section 13 Watchdog Timer (WDT)

Section 14 Power-Down Modes

Section 15 Timer Unit (TMU)

Section 17 Serial Communication Interface with FIFO (SCIF)

Section 18 SIM Card Module (SIM)

Section 19 I^2C Bus Interface

Section 20 Serial Sound Interface (SSI) Module

Section 21 USB Host Module (USB)

Section 22 Controller Area Network 2 (HCAN2)

Section 23 Serial Protocol Interface (HSPI)

Section 26 Multimedia Card Interface (MMCIF)

Section 27 Multifunctional Interface (MFI)

Section 28 User Debug Interface (H-UDI)

Section 29 A/D Converter (ADC)

Section 31 User Break Controller (UBC)

Section 33 Electrical Characteristics

Appendix

Section 1 Overview

Section 2 Programming Model

Section 3 Floating-Point Unit (FPU)

Section 4 Instruction Set

Section 5 Pipelining

Section 6 Memory Management Unit (MMU)

Section 8 Exceptions

Section 9 Interrupt Controller (INTC)

Section 10 Bus State Controller (BSC)

Section 11 Direct Memory Access Controller (DMAC)

Section 12 Clock Pulse Generator (CPG)

Section 13 Watchdog Timer (WDT)

Section 14 Power-Down Modes

Section 33 Electrical Characteristics

Appendix

1.1 SH7760 Features

This LSI is a microcomputer, featuring an LCD controller, USB host, and other peripheral functions. The SuperH™ RISC engine is a Renesas Technology-original 32-bit RISC (Reduced Instruction Set Computer) microcomputer. The SuperH™ RISC engine employs a fixed-length 16-bit instruction set, allowing an approximately 50% reduction in program size over a 32-bit instruction set.

This LSI features the SH-4 CPU, which at the instruction set level is upwardly compatible with the SH-1, SH-2, and SH-3 microcomputers. This LSI has an instruction cache, an operand cache that can be switched between copy-back and write-through modes, a 4-entry full-associative instruction TLB (translation look aside buffer), and MMU (memory management unit) with 64-entry full-associative shared TLB. The sizes of the instruction cache and operand cache are 16 Kbytes and 32 Kbytes.

This LSI also features the bus state controller (BSC) that can connect to synchronous DRAM. Also, because of its on-chip functions, such as an LCD controller, a USB host, timers, and serial communication functions, required for multimedia and OA equipment, this LSI enables a dramatic reduction in system costs.

Table 1.1 Features

Item	Features
LSI	• Operating frequency: 200 MHz • Performance: 360MIPS, 1.4 GFLOPS • Voltage: 1.5 V (internal), 3.3 V (I/O) • Superscalar architecture: Parallel execution of two instructions • Packages: 256-pin BGA (Size: 17 × 17 mm, pin pitch: 0.8 mm) • External buses: — Separate 26-bit address and 32-bit data buses — External bus frequency: 67MHz • Choice of MFI mode or LCD mode: — MFI mode: 8-/16-bit parallel interface (supports 68-/80-family interface) — LCD mode: LCD controller/data output
CPU	• Original Renesas Technology SuperH architecture • 32-bit internal data bus • General register file: — Sixteen 32-bit general registers (and eight 32-bit shadow registers) — Seven 32-bit control registers — Four 32-bit system registers • RISC-type instruction set (upward-compatible with SH-1, SH-2, and SH-3) — Fixed 16-bit instruction length for improved code efficiency — Load-store architecture — Delayed branch instructions — Conditional execution — C-based instruction set • Superscalar architecture (providing simultaneous execution of two instructions) including FPU • Instruction execution time: Maximum 2 instructions/cycle • Virtual address space: 4 Gbytes (448-Mbyte external memory space) • Space identifier ASIDs: 8 bits, 256 virtual address spaces • On-chip multiplier • 5-stage pipeline

FPU	• On-chip floating-point coprocessor • Supports single-precision (32 bits) and double-precision (64 bits) • Supports IEEE754-compliant data types and exceptions • Two rounding modes: Round to Nearest and Round to Zero • Handling of denormalized numbers: Truncation to zero or interrupt generation for compliance with IEEE754 • Floating-point registers: 32 bits × 16 × 2 banks — (single-precision: 32-bit × 16 or double-precision: 64-bit × 8) × 2 banks • 32-bit CPU-FPU floating-point communication register (FPUL) • Supports FMAC (multiply-and-accumulate) instruction • Supports FDIV (divide) and FSQRT (square root) instructions • Supports FLDI0/FLDI1 (load constant 0/1) instructions • Instruction execution times: — Latency (FMAC/FADD/FSUB/FMUL): 3 cycles (single-precision), 8 cycles (double-precision) — Pitch (FMAC/FADD/FSUB/FMUL): 1 cycle (single-precision), 6 cycles (double-precision) Note: FMAC is supported for single-precision only. • 3-D graphics instructions (single-precision only): — 4-dimensional vector conversion and matrix operations (FTRV): 4 cycles (pitch), 7 cycles (latency) — 4-dimensional vector inner product (FIPR): 1 cycle (pitch), 4 cycles (latency) • 5-stage pipeline

Clock pulse generator (CPG)	• Choice of main clock: 1, 6, or 12 times EXTAL • Clock modes: — CPU frequency: 1, 1/2, 1/3, 1/4, 1/6, or 1/8 times main clock — Bus frequency: 1, 1/2, 1/3, 1/4, 1/6, or 1/8 times main clock — Peripheral frequency: 1/2, 1/3, 1/4, 1/6, or 1/8 times main clock • Power-down modes: — Sleep mode — Deep sleep mode — Standby mode — Hardware standby mode — Module standby mode • Single-channel watchdog timer
Memory management unit (MMU)	• 4-Gbyte address space, 256 address space identifiers (8-bit ASIDs) • Single virtual mode and multiple virtual memory mode • Supports multiple page sizes: 1 Kbyte, 4 Kbytes, 64 Kbytes, 1 Mbyte • 4-entry fully-associative TLB for instructions • 64-entry fully-associative TLB for instructions and operands • Supports software-controlled replacement and random-counter replacement algorithm • TLB contents can be accessed directly by address mapping

Cache memory	• Instruction cache (IC) — 16-Kbyte, 2-way set associative (LRU) — 256 entries, 32-byte block length — Cache-double-mode (16-Kbyte cache) — Index mode • Operand cache (OC) — 32-Kbyte, 2-way set associative (LRU) — 512 entries, 32-byte block length — Cache-double-mode (32-Kbyte cache) — Index mode — RAM mode (16-Kbyte cache + 16-Kbyte RAM) — Choice of write method (copy-back or write-through) • Single-stage copy-back buffer, single-stage write-through buffer • Cache memory contents can be accessed directly by address mapping (usable as on-chip memory) • Store queue (32 bytes × 2 entries)
Interrupt controller (INTC)	• Nine independent external interrupts: NMI, IRL3 to IRL0, and IRQ7 to IRQ4 • 15-level signed external interrupts: IRL3 to IRL0 • On-chip peripheral module interrupts: Priority level can be set for each module
User break controller (UBC)	• Supports debugging by means of user break interrupts • Two break channels • Address, data value, access type, and data size can all be set as break conditions • Supports sequential break function

Bus state controller (BSC)	• Supports external memory access • External memory space divided into seven areas, each of up to 64 Mbytes, with the following parameters settable for each area: — Bus size (8, 16, or 32 bits) — Number of wait cycles (hardware wait function also supported) — SRAM, synchronous DRAM, or burst ROM — Supports PCMCIA interface (only in little endian mode) • Synchronous DRAM refresh functions: — Programmable refresh interval — Supports auto refresh mode and self-refresh mode • Synchronous DRAM burst access function • Big endian or little endian mode can be set
Direct memory access controller (DMAC)	• 8-channel physical address DMA controller • Transfer data size: 8, 16, 32, or 64 bits, or 32 bytes • Address modes: — 1-bus-cycle single address mode — 2-bus-cycle dual address mode • Transfer requests: External, peripheral module, or auto-requests • Choice of DACK or DRAK (four external pins) • Bus modes: Cycle-steal or burst mode • Supports on-chip FIFO bridge (16-stage × 32-bit FIFO × 7) to achieve high-speed transfer for HAC/SSI, USB and LCDC
Timer unit (TMU)	• 3-channel auto-reload 32-bit timer • Input-capture function (only channel 2) • Choice of six types counter input clocks (external and peripheral clocks)
Compare match timer (CMT)	• 4-channel auto-reload 32-bit timers • Choice of 16 or 32 bits • Choice of 1-shot or free-running operation • Choice of an interrupt source or DMA transfer request from compare match or overflow

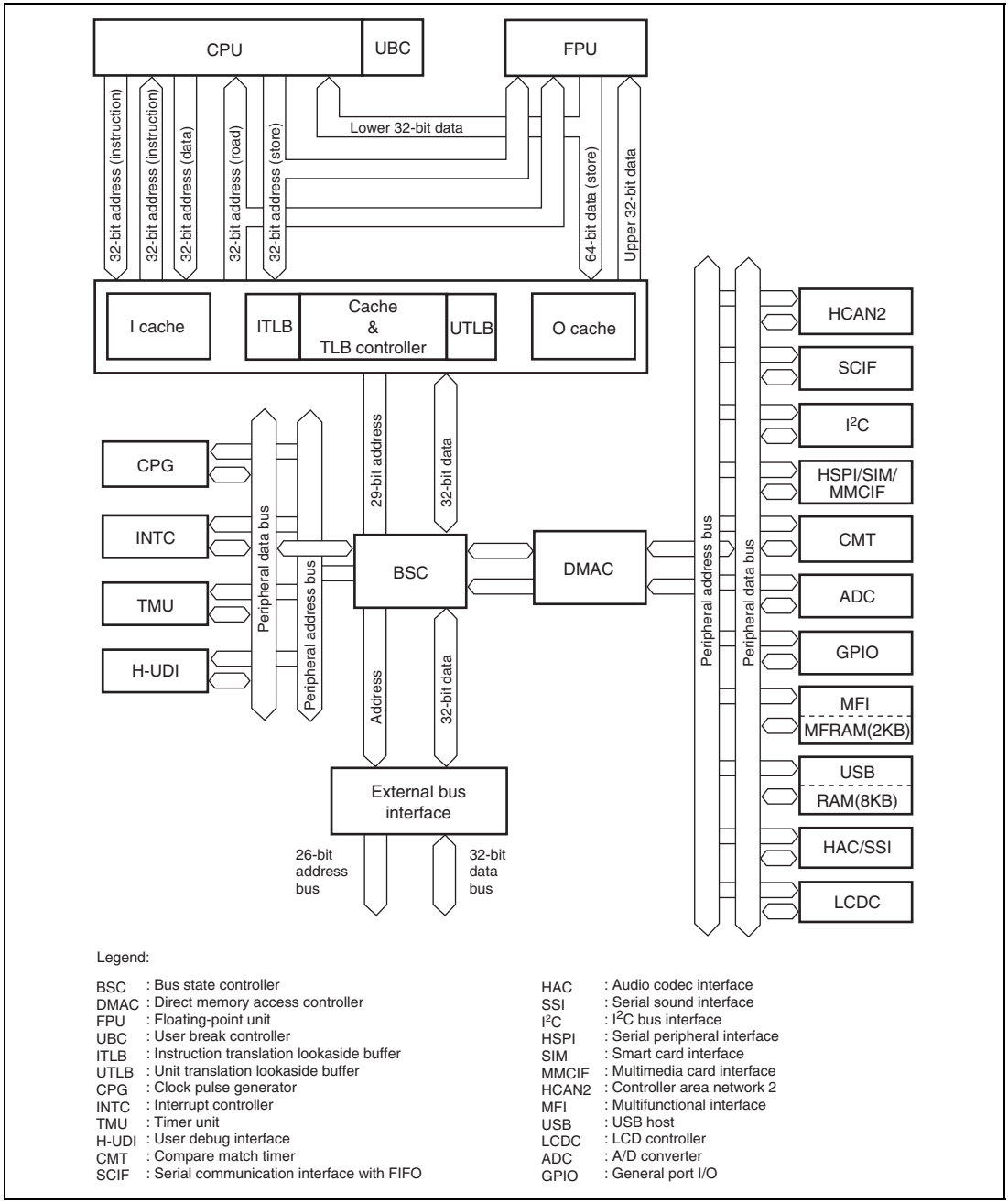
Serial communication interface (SCIF)	• Three full-duplex communications channels • On-chip 128-byte FIFOs for all channels • Choice of asynchronous mode or synchronous mode • Can select any bit rate generated by on-chip baud-rate generator • On-chip modem control function ($\overline{\text{SCIF\_RTS}}$ and $\overline{\text{SCIF\_CTS}}$) for channel 1 and 2
Audio codec interface (HAC)	• Digital interface for audio codec • Supports transfer for slot 1 to slot 4 • Choice of 16- or 20-bit DMA transfer • Supports various sampling rates by adjusting slot data • Generates interrupt: data ready, data request, overflow, and underrun
Serial sound interface (SSI)	• 2-channel bi-directional transfer (maximum) • Support multi-channel and compressed-data transfer • Selectable frame size
I^2C bus interface (I^2C)	• 2 channels (maximum) • Master/slave • 16-byte FIFO • Supports high-speed mode (400 kbits/sec) • Supports version 1.0
Multimedia card interface (MMCIF)	• Supports MMC mode • A maximum bit rate of 20 Mbps at 20 MHz of peripheral clock • Interface with MCCLK output for transfer clock output, MCCMD I/O for command output/response input, MCDAT I/O (data I/O) • Four interrupt sources
Smart card interface (SIM)	• Supports ISO/IEC7816-3 (Identification card) • Asynchronous half-duplex transfer (8 bits) • Can select any bit rate generated by on-chip baud-rate generator • Generates and checks parity bit • Four interrupt sources

Controller area network 2 (HCAN2)	• 2 channels (maximum) • Supports CAN specification 2.0A and 2.0B — Standard data and remote frame (11-bit ID) — Extended data and remote frame (29-bit ID) • 32 independent message buffers using standard (11-bit) and extended (29-bit) ID format • 31 Mailboxes can be used for transmission or reception • One Mailbox can be used for only reception • Message reception filtering by IDs: — Standard message ID — Extended message ID • Local reception filter for reception-only Mailbox (standard and extended message ID) can be specified • Power-down sleep mode • A maximum of 1-Mbit/s CAN data transfer rate can be specified • Transmit message queue having internal priority sorting mechanism which handle priority-inversion issue of real time applications • Data buffer access without hand-shaking
Serial peripheral interface (HSPI)	• 1 channel • Master/slave mode • Selectable bit rate generated by on-chip baud-rate generator
Multifunctional interface (MFI)	• 2-Kbyte internal memory can be read from or written to via the MFI pin in 32-bit units or by the CPU in 8-/16-/32-bit units. • Choice of 8- or 16-bit parallel interface • Supports 68-/80-family interface (can be switched during reset) • Endians can be switched
USB host	• 1 channel • Supports USB version 1.1 and OHCI 1.0 • Supports data transfer rate of 1.5 Mbps and 12Mbps • On-chip 8-Kbyte SRAM as shared memory defined in OHCI specification

LCD controller (LCDC)	• Supports 16 × 1 to 1024 × 1024 dots (8 bpp: a maximum of 640 ×480 dots, 16 bpp: a maximum of 400 × 240 dots) • Supports 4, 8, 15, and 16 bpp color modes • Supports 1, 2, 4, and 6 bpp grayscale modes. • Supports TFT/DSTN/STN display • Selectable signal polarities • 24-bit color palette memory (16 bits of 24 bits are valid: R: 5/G: 6/B: 5) • Unified graphics memory architecture
A/D converter (ADC)	• 10-bit resolution • 4-channel input • Three types of conversion modes — Single mode: 1-channel A/D conversion — Multi mode: 1- to 4-channel A/D conversion — Scan mode: 1- to 4-channel A/D conversion • Conversion time: 8 μs for are channel (maximum) • Absolute error ± 4LSB
General I/O (GPIO)	• 70 general I/O port (69 for I/O and one for output)

Product lineup

Group	Voltage	Operating Frequency	Part No.	Package
SH7760	1.5 V	200 MHz	HD6417760BL200A	256-pin BGA (BP-256B)
			HD6417760BL200AV	
			HD6417760BL200AD	
			HD6417760BL200ADV	
			HD6417760BP200AD	256-pin BGA (BP-256F)
			HD6417760BP200ADV	



Legend:

BSC	: Bus state controller
DMAC	: Direct memory access controller
FPU	: Floating-point unit
UBC	: User break controller
ITLB	: Instruction translation lookaside buffer
UTLB	: Unit translation lookaside buffer
CPG	: Clock pulse generator
INTC	: Interrupt controller
TMU	: Timer unit
H-UDI	: User debug interface
CMT	: Compare match timer
SCIF	: Serial communication interface with FIFO
HAC	: Audio codec interface
SSI	: Serial sound interface
I^2C	: I^2C bus interface
HSPI	: Serial peripheral interface
SIM	: Smart card interface
MMCIF	: Multimedia card interface
HCAN2	: Controller area network 2
MFI	: Multifunctional interface
USB	: USB host
LCDC	: LCD controller
ADC	: A/D converter
GPIO	: General port I/O

Figure 1.1 SH7760 Block Diagram

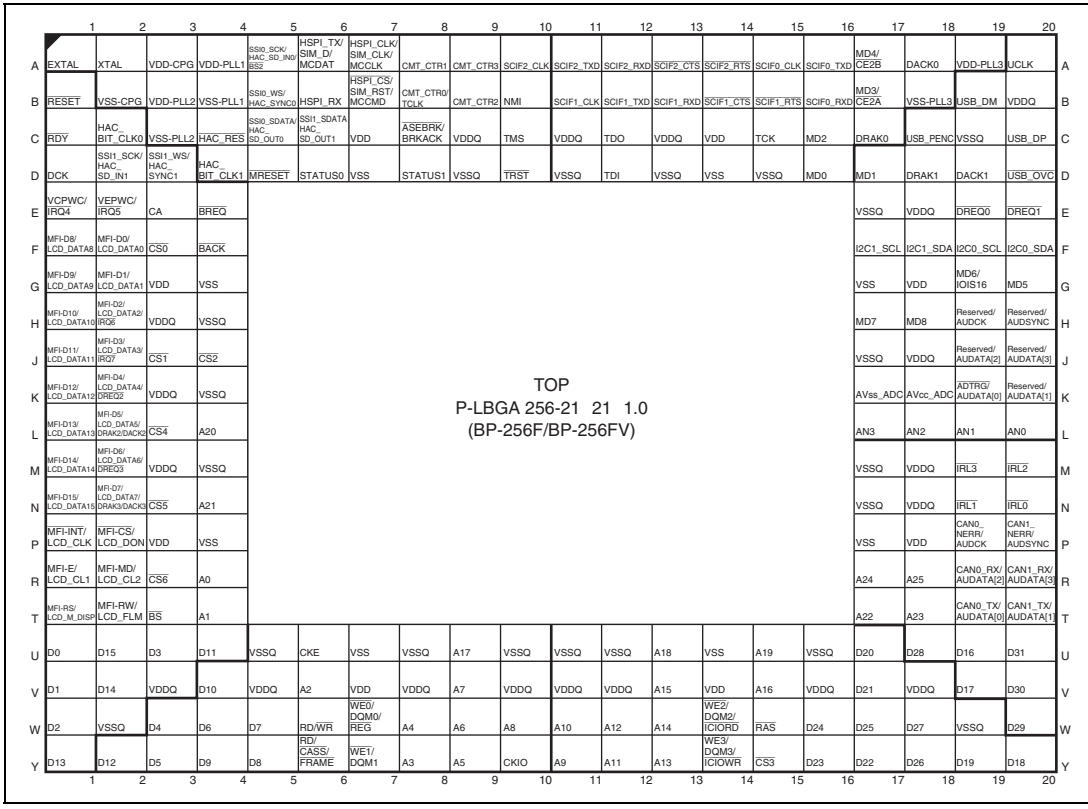


Figure 1.2 SH7760 Pin Arrangement (BP-256F/BP-256FV)

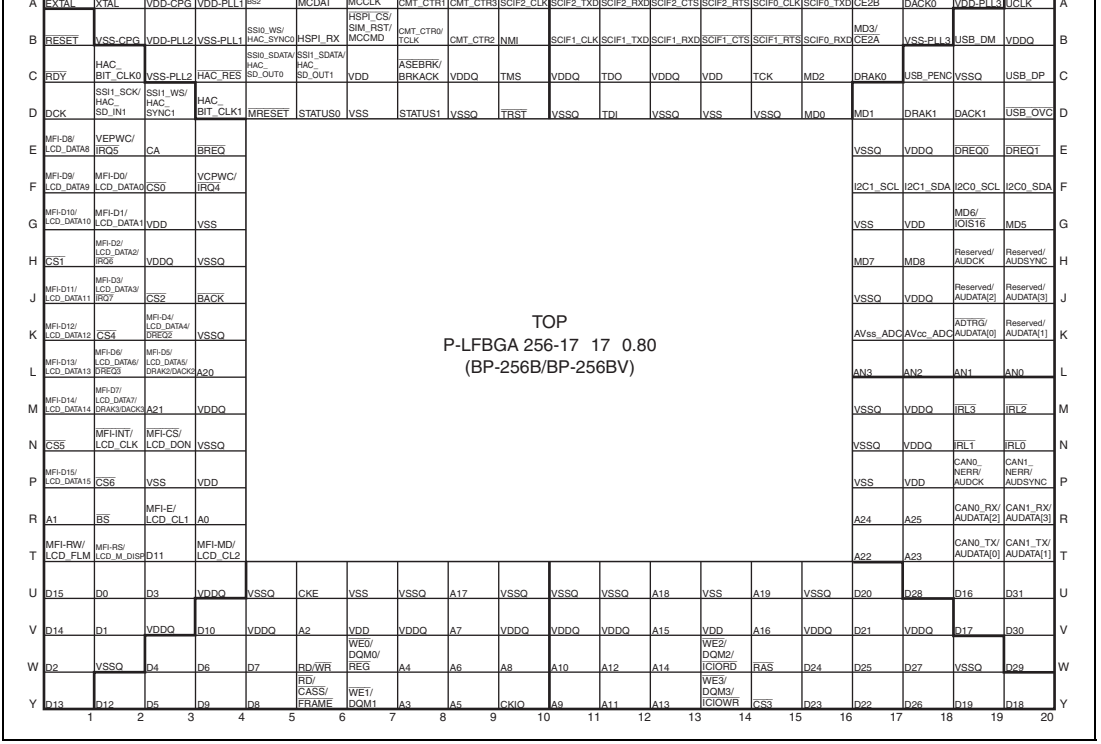


Figure 1.3 SH7760 Pin Arrangement (BP-256B/BP-256BV)

Table 1.2 lists the pin configuration of the BP-256F (21 mm*[1]), and table 1.3 lists the pin configuration of the BP-256B (17 mm*[2]). In the I/O column, I, O, and IO indicate input, output, and input/output, respectively. In the GPIO column, O indicates a pin which also functions as a general I/O port.

Notes: 1. HD6417760BP200AD, HD6417760BP200ADV
2. HD6417760BL200A, HD6417760BL200AV, HD6417760BL200AD, HD6417760BL200ADV

Table 1.2 Pin Configuration (BP-256F: 21 mm)

Pin No.	Pin Name	I/O	Function	GPIO
A1	EXTAL	I	External input clock/crystal resonator	
A2	XTAL	O	Crystal resonator	
A3	VDD-CPG	—	CPG VDD	
A4	VDD-PLL1	—	PLL1 VDD	
A5	SSI0_SCK/HAC_SD_IN0/$\overline{BS2}$	IO/I/O	SSI serial clock input/HAC serial data/bus start 2	O
A6	HSPI_TX/SIM_D/MCDAT	O/IO/IO	HSPI transmit data/SIM data transfer/MMCIF data	O
A7	HSPI_CLK/SIM_CLK/MCCLK	IO/O/O	HSPI serial clock/SIM clock/MMCIF clock	O
A8	CMT_CTR1	IO	CMT counter	O
A9	CMT_CTR3	IO	CMT counter	O
A10	SCIF2_CLK	IO	SCIF serial clock	O
A11	SCIF2_TXD	O	SCIF transmit data	O
A12	SCIF2_RXD	I	SCIF receive data	O
A13	$\overline{SCIF2\_CTS}$	IO	SCIF modem control	O *[1]
A14	$\overline{SCIF2\_RTS}$	IO	SCIF modem control	O *[2]
A15	SCIF0_CLK	IO	SCIF serial clock	O
A16	SCIF0_TXD	O	SCIF transmit data	O
A17	MD4/$\overline{CE2B}$	IO	Mode control 4/PCMCIA-CE	
A18	DACK0	O	DMAC0 bus acknowledge	
A19	VDD-PLL3	—	PLL3 VDD	
A20	UCLK	I	USB operation clock	O
B1	$\overline{RESET}$	I	Reset	
B2	VSS-CPG	—	CPG GND	
B3	VDD-PLL2	—	PLL2 VDD	

B4	VSS-PLL1	—	PLL1 GND	
B5	SSI0_WS/HAC_SYNC0	IO/O	SSI word selection/HAC from sync output	O
B6	HSPI_RX	I	HSPI receive data input	O
B7	$\overline{\text{HSPI\_CS}}$/SIM_RST/MCCMD	IO/O/IO	HSPI chip selection/SIM reset/MMCIF command/response	O
B8	CMT_CTR0/TCLK	IO/I	CMT counter/TMU clock	O
B9	CMT_CTR2	IO	CMT counter	O
B10	NMI	I	Non-maskable interrupt	
B11	SCIF1_CLK	IO	SCIF serial clock	O
B12	SCIF1_TXD	O	SCIF transmit data	O
B13	SCIF1_RXD	I	SCIF receive data	O
B14	$\overline{\text{SCIF1\_CTS}}$	IO	SCIF modem control	O
B15	$\overline{\text{SCIF1\_RTS}}$	IO	SCIF modem control	O
B16	SCIF0_RXD	I	SCIF receive data	O
B17	MD3/$\overline{\text{CE2A}}$	IO	Mode control 3/PCMCIA-CE	
B18	VSS-PLL3	—	PLL3 GND	
B19	USB_DM	IO	USB D-transceiver	
B20	VDDQ	—	USB analog VDD	
C1	$\overline{\text{RDY}}$	I	Bus ready	
C2	HAC_BIT_CLK0	I	HAC serial data clock/SSI divider input clock	O
C3	VSS-PLL2	—	PLL2 GND	
C4	$\overline{\text{HAC\_RES}}$	O	HAC reset	O
C5	SSI0_SDATA/HAC_SD_OUT0	IO/O	SSI serial data/HAC serial data	O
C6	SSI1_SDATA/HAC-SD_OUT1	IO/O	SSI serial data/HAC serial data	O
C7	VDD	—	Internal VDD	
C8	$\overline{\text{ASEBRK}}$/BRKACK	I/O	H-UDI emulator	
C9	VDDQ	—	I/O VDD	
C10	TMS	I	H-UDI mode	
C11	VDDQ	—	I/O VDD	
C12	TDO	O	H_UDI data	
C13	VDDQ	—	I/O VDD	
C14	VDD	—	Internal VDD	
C15	TCK	I	H-UDI clock	

C16	MD2	I	Mode control 2	
C17	DRAK0	O	DMAC 1 request acknowledgement	
C18	USB_PENC	O	USB power-on enable control	O *2
C19	VSSQ	—	USB analog VSS	
C20	USB_DP	IO	USB D+ transceiver	
D1	DCK	O	Clock	O *3
D2	SSI1_SCK/HAC_SD_IN1	IO/I	SSI clock/HAC serial data	O
D3	SSI1_WS/HAC_SYNC1	IO/O	SSI word selection/HAC frame sync output	O
D4	HAC_BIT_CLK1	—	HAC serial data clock/SSI divider input clock	O
D5	$\overline{\text{MRESET}}$	I	Manual reset	
D6	STATUS0	O	Status 0	
D7	VSS	—	Internal GND	
D8	STATUS1	O	Status 1	
D9	VSSQ	—	I/O GND	
D10	$\overline{\text{TRST}}$	I	H-UDI reset	
D11	VSSQ	—	I/O GND	
D12	TDI	I	H-UDI data	
D13	VSSQ	—	I/O GND	
D14	VSS	—	Internal GND	
D15	VSSQ	—	I/O GND	
D16	MD0	I	Mode control 0	
D17	MD1	I	Mode control 1	
D18	DRAK1	O	DMAC 1 request acknowledgement	
D19	DACK1	O	DMAC 1 bus acknowledgement	
D20	$\overline{\text{USB\_OVC}}$	I	USB overcurrent detection	O
E1	VCPWC/$\overline{\text{IRQ4}}$	O/I	LCD panel power supply control (VCC)/external interrupt request 4	O
E2	VEPWC/$\overline{\text{IRQ5}}$	O/I	LCD panel power supply control (VEE)/external interrupt request 5	O
E3	CA	I	Chip active	
E4	$\overline{\text{BREQ}}$	I	Bus request	
E17	VSSQ	—	I/O GND	
E18	VDDQ	—	I/O VDD	
E19	$\overline{\text{DREQ0}}$	I	DMAC 0 request	

E20	DREQ1	I	DMAC 1 request	
F1	MFI-D8/LCD_DATA8	IO/O	MFI data/LCD panel data	O
F2	MFI-D0/LCD_DATA0	IO/O	MFI data/LCD panel data	O
F3	$\overline{CS0}$	O	Chip select 1	
F4	$\overline{BACK}$	O	Bus acknowledgement	
F17	I2C1_SCL	IO	I^2C serial clock	
F18	I2C1_SDA	IO	I^2C serial data	
F19	I2C0_SCL	IO	I^2C serial clock	
F20	I2C0_SDA	IO	I^2C serial data	
G1	MFI-D9/LCD_DATA9	IO/O	MFI data/LCD panel data	O
G2	MFI-D1/LCD_DATA1	IO/O	MFI data/LCD panel data	O
G3	VDD	—	Internal VDD	
G4	VSS	—	Internal GND	
G17	VSS	—	Internal GND	
G18	VDD	—	Internal VDD	
G19	MD6/$\overline{IOIS16}$	I/I	Mode control 6/IOIS16 (PCMCIA)	
G20	MD5	I	Mode control 5	
H1	MFI-D10/LCD_DATA10	IO/O	MFI data/LCD panel data	O
H2	MFI-D2/LCD_DATA2/$\overline{IRQ6}$	IO/O/I	MFI data/LCD panel data/external interrupt request 6	O
H3	VDDQ	—	IO VDD	
H4	VSSQ	—	IO GND	
H17	MD7	I	Mode control 7	
H18	MD8	I	Mode control 8	
H19	Reserved/AUDCK	O	Reserved/H-UDI emulator	O
H20	Reserved/AUDSYNC	O	Reserved/H-UDI emulator	O
J1	MFI-D11/LCD_DATA11	IO/O	MFI data/LCD panel data	O
J2	MFI-D3/LCD_DATA3/$\overline{IRQ7}$	IO/O/I	MFI data/LCD panel data/external interrupt request 7	O
J3	$\overline{CS1}$	O	Chip select 1	
J4	$\overline{CS2}$	O	Chip select 2	
J17	VSSQ	—	IO GND	
J18	VDDQ	—	IO VDD	
J19	Reserved/AUDATA[2]	O	Reserved/H-UDI emulator	O

J20	Reserved/AUDATA[3]	O	Reserved/H-UDI emulator	O *2
K1	MFI-D12/LCD_DATA12	IO/O	MFI data/LCD panel data	O
K2	MFI-D4/LCD_DATA4/$\overline{\text{DREQ2}}$	IO/O/I	MFI data/LCD panel data/DMAC2 request	O
K3	VDDQ	—	IO VDD	O
K4	VSSQ	—	IO GND	
K17	AVss_ADC	—	ADC analog GND	
K18	AVcc_ADC	—	ADC analog VCC	
K19	$\overline{\text{ADTRG}}$/AUDATA[0]	I/O	A/D external trigger/H-UDI emulator	O
K20	Reserved/AUDATA[1]	O	Reserved/H-UDI emulator	O
L1	MFI-D13/LCD_DATA13	IO/O	MFI data/LCD panel data	O
L2	MFI-D5/LCD_DATA5/DRAK2/DACK2	IO/O/O/O	MFI data/LCD panel data/DMAC2 request acknowledgement/DMAC2 bus acknowledgement	O
L3	$\overline{\text{CS4}}$	—	Chip select 4	O
L4	A20	O	Address bus	
L17	AN3	I	ADC analog input	
L18	AN2	I	ADC analog input	
L19	AN1	I	ADC analog input	
L20	AN0	I	ADC analog input	
M1	MFI-D14/LCD_DATA14	IO/O	MFI data/LCD panel data	O
M2	MFI-D6/LCD_DATA6/$\overline{\text{DREQ3}}$	IO/O/I	MFI data/LCD panel data/DMAC3 request	O
M3	VDDQ	—	IO VDD	
M4	VSSQ	—	IO GND	
M17	VSSQ	—	IO GND	
M18	VDDQ	—	IO VDD	
M19	$\overline{\text{IRL3}}$	I	IRL interrupt request 3	
M20	$\overline{\text{IRL2}}$	I	IRL interrupt request 2	
N1	MFI-D15/LCD_DATA15	IO/O	MFI data/LCD panel data	O
N2	MFI-D7/LCD_DATA7/DRAK3/DACK3	IO/O/O/O	MFI data/LCD panel data/DMAC3 request acknowledgement/DMAC3 bus acknowledgement	O
N3	$\overline{\text{CS5}}$	O	Chip select 5	
N4	A21	O	Address bus	
N17	VSSQ	—	IO GND	
N18	VDDQ	—	IO VDD	
N19	$\overline{\text{IRL1}}$	I	IRL interrupt request 1	

N20	IRL0	I	IRL interrupt request 0	
P1	$\overline{\text{MFI-INT}}$/LCD_CLK	O/I	MFI interrupt/LCD clock	O *1
P2	$\overline{\text{MFI-CS}}$/LCD_DON	I/O	MFI chip selection/LCD display-on signal	O *1
P3	VDD	—	Internal VDD	
P4	VSS	—	Internal GND	
P17	VSS	—	Internal GND	
P18	VDD	—	Internal VDD	
P19	CAN0_NERR/AUDCK	I/O	HCAN0 bus error signal/H-UDI emulator	O
P20	CAN1_NERR/AUDSYNC	I/O	HCAN1 bus error signal/H-UDI emulator	O
R1	MFI-E/LCD_CL1	I/O	MFI enable/ LCD shift clock 1	O *1
R2	MFI-MD/LCD_CL2	I/O	MFI mode/LCD shift clock 2	O *1
R3	$\overline{\text{CS6}}$	O	Chip select 6	
R4	A0	O	Address bus	
R17	A24	O	Address bus	
R18	A25	O	Address bus	
R19	CAN0_RX/AUDATA[2]	I/O	HCAN0 bus data receive signal/H-UDI emulator	O
R20	CAN1_RX/AUDATA[3]	I/O	HCAN1 bus data receive signal/H-UDI emulator	O
T1	MFI-RS/LCD_M_DISP	I/O	MFI register select/LCD current-alternating signal/DISP signal	O *1
T2	MFI-RW/LCD_FLM	I/O	MFI read/write/LCD first line marker	O *1
T3	$\overline{\text{BS}}$	O	Bus start	
T4	A1	O	Address bus	
T17	A22	O	Address bus	
T18	A23	O	Address bus	
T19	CAN0_TX/AUDATA[0]	O/O	HCAN0 bus data transmit signal/H-UDI emulator	O
T20	CAN1_TX/AUDATA[1]	O/O	HCAN1 bus data transmit signal/H-UDI emulator	O
U1	D0	IO	Data bus	
U2	D15	IO	Data bus	
U3	D3	IO	Data bus	
U4	D11	IO	Data bus	
U5	VSSQ	—	IO GND	
U6	CKE	O	Clock output enable	
U7	VSS	—	Internal GND	

U8	VSSQ	—	IO GND
U9	A17	O	Address bus
U10	VSSQ	—	IO GND
U11	VSSQ	—	IO GND
U12	VSSQ	—	IO GND
U13	A18	O	Address bus
U14	VSS	—	Internal GND
U15	A19	O	Address bus
U16	VSSQ	—	IO GND
U17	D20	IO	Data bus
U18	D28	IO	Data bus
U19	D16	IO	Data bus
U20	D31	IO	Data bus
V1	D1	IO	Data bus
V2	D14	IO	Data bus
V3	VDDQ	—	IO VDD
V4	D10	IO	Data bus
V5	VDDQ	—	IO VDD
V6	A2	O	Address bus
V7	VDD	—	Internal VDD
V8	VDDQ	—	IO VDD
V9	A7	O	Address bus
V10	VDDQ	—	IO VDD
V11	VDDQ	—	IO VDD
V12	VDDQ	—	IO VDD
V13	A15	O	Address bus
V14	VDD	—	Internal VDD
V15	A16	O	Address bus
V16	VDDQ	—	IO VDD
V17	D21	IO	Data bus
V18	VDDQ	—	IO VDD
V19	D17	IO	Data bus
V20	D30	IO	Data bus

W1	D2	IO	Data bus
W2	VSSQ	—	IO GND
W3	D4	IO	Data bus
W4	D6	IO	Data bus
W5	D7	IO	Data bus
W6	RD/$\overline{WR}$	O	Read/write
W7	$\overline{WE0}$/DQM0/$\overline{REG}$	O/O	Selection signal for D7 to D0/REG
W8	A4	O	Address bus
W9	A6	O	Address bus
W10	A8	O	Address bus
W11	A10	O	Address bus
W12	A12	O	Address bus
W13	A14	O	Address bus
W14	$\overline{WE2}$/DQM2/$\overline{ICIORD}$	O/O	Selection signal for D23 to D16/ICIORD
W15	$\overline{RAS}$	O	RAS
W16	D24	IO	Data bus
W17	D25	IO	Data bus
W18	D27	IO	Data bus
W19	VSSQ	—	IO GND
W20	D29	IO	Data bus
Y1	D13	IO	Data bus
Y2	D12	IO	Data bus
Y3	D5	IO	Data bus
Y4	D9	IO	Data bus
Y5	D8	IO	Data bus
Y6	$\overline{RD}$/$\overline{CASS}$/$\overline{FRAME}$	O/O/O	Read/CAS/FRAME
Y7	$\overline{WE1}$/DQM1	O	Selection signal for D15 to D8
Y8	A3	O	Address bus
Y9	A5	—	Address bus
Y10	CKIO	O	Clock output
Y11	A9	O	Address bus
Y12	A11	O	Address bus
Y13	A13	O	Address bus

Y14	WE3/DQM3/ICIOWR	O/O	Selection signal for D31 to D24/ICIOWR
Y15	$\overline{CS3}$	O	Chip select 3
Y16	D23	IO	Data bus
Y17	D22	IO	Data bus
Y18	D26	IO	Data bus
Y19	D19	IO	Data bus
Y20	D18	IO	Data bus

Legend: In the I/O column, I, O, IO, and — indicate input, output, input/output, and no direction, respectively.

Notes: 1. Can be used as a GPIO interrupt pin.
2. Can be used as a GPIO interrupt pin. When an interrupt occurs, this LSI exits standby mode.
3. Only outputs.

Pin No.	Pin Name	I/O	Function	GPIO
A1	EXTAL	I	External input clock/crystal resonator	
A2	XTAL	O	Crystal resonator	
A3	VDD-CPG	—	CPG VDD	
A4	VDD-PLL1	—	PLL1 VDD	
A5	SSI0_SCK/HAC_SD_IN0/$\overline{BS2}$	IO/I/O	SSI serial clock input/HAC serial data/bus start 2	O
A6	HSPI_TX/SIM_D/MCDAT	O/IO/IO	HSPI transmit data/SIM data transfer/MMCIF data	O
A7	HSPI_CLK/SIM_CLK/MCCLK	IO/O/O	HSPI serial clock/SIM clock/MMCIF clock	O
A8	CMT_CTR1	IO	CMT counter	O
A9	CMT_CTR3	IO	CMT counter	O
A10	SCIF2_CLK	IO	SCIF serial clock	O
A11	SCIF2_TXD	O	SCIF transmit data	O
A12	SCIF2_RXD	I	SCIF receive data	O
A13	$\overline{SCIF2\_CTS}$	IO	SCIF modem control	O *1
A14	$\overline{SCIF2\_RTS}$	IO	SCIF modem control	O *2
A15	SCIF0_CLK	IO	SCIF serial clock	O
A16	SCIF0_TXD	O	SCIF transmit data	O
A17	MD4/$\overline{CE2B}$	IO	Mode control 4/PCMCIA-CE	
A18	DACK0	O	DMAC0 bus acknowledge	
A19	VDD-PLL3	—	PLL3 VDD	
A20	UCLK	I	USB operation clock	O
B1	$\overline{RESET}$	I	Reset	
B2	VSS-CPG	—	CPG GND	
B3	VDD-PLL2	—	PLL2 VDD	
B4	VSS-PLL1	—	PLL1 GND	
B5	SSI0_WS/HAC_SYNC0	IO/O	SSI word selection/HAC from sync output	O
B6	HSPI_RX	I	HSPI receive data input	O
B7	$\overline{HSPI\_CS}$/SIM_RST/MCCMD	IO/O/IO	HSPI chip selection/SIM reset/MMCIF command/response	O
B8	CMT_CTR0/TCLK	IO/I	CMT counter/TMU clock	O
B9	CMT_CTR2	IO	CMT counter	O
B10	NMI	I	Non-maskerable interrupt	
B11	SCIF1_CLK	IO	SCIF serial clock	O

B12	SCIF1_TXD	O	SCIF transmit data	O
B13	SCIF1_RXD	I	SCIF receive data	O
B14	$\overline{\text{SCIF1\_CTS}}$	IO	SCIF modem control	O
B15	$\overline{\text{SCIF1\_RTS}}$	IO	SCIF modem control	O
B16	SCIF0_RXD	I	SCIF receive data	O
B17	MD3/$\overline{\text{CE2A}}$	IO	Mode control 3/PCMCIA-CE	
B18	VSS-PLL3	—	PLL3 GND	
B19	USB_DM	IO	USB D-transceiver	
B20	VDDQ	—	USB analog VDD	
C1	$\overline{\text{RDY}}$	I	Bus ready	
C2	HAC_BIT_CLK0	I	HAC serial data clock/SSI divider input clock	O
C3	VSS-PLL2	—	PLL2 GND	
C4	$\overline{\text{HAC\_RES}}$	O	HAC reset	O
C5	SSI0_SDATA/HAC_SD_OUT0	IO/O	SSI serial data/HAC serial data	O
C6	SSI1_SDATA/HAC-SD_OUT1	IO/O	SSI serial data/HAC serial data	O
C7	VDD	—	Internal VDD	
C8	$\overline{\text{ASEBRK}}$/BRKACK	I/O	H-UDI emulator	
C9	VDDQ	—	I/O VDD	
C10	TMS	I	H-UDI mode	
C11	VDDQ	—	I/O VDD	
C12	TDO	O	H_UDI data	
C13	VDDQ	—	I/O VDD	
C14	VDD	—	Internal VDD	
C15	TCK	I	H-UDI clock	
C16	MD2	I	Mode control 2	
C17	DRAK0	O	DMAC 1 request acknowledgement	
C18	USB_PENC	O	USB power-on enable control	O *2
C19	VSSQ	—	USB analog VSS	
C20	USB_DP	IO	USB D+ transceiver	
D1	DCK	O	Clock	O *3
D2	SSI1_SCK/HAC_SD_IN1	IO/I	SSI clock/HAC serial data	O
D3	SSI1_WS/HAC_SYNC1	IO/O	SSI word selection/HAC frame sync output	O
D4	HAC_BIT_CLK1	—	HAC serial data clock/SSI divider input clock	O

D5	MRESET	I	Manual reset	
D6	STATUS0	O	Status 0	
D7	VSS	—	Internal GND	
D8	STATUS1	O	Status 1	
D9	VSSQ	—	I/O GND	
D10	$\overline{\text{TRST}}$	I	H-UDI reset	
D11	VSSQ	—	I/O GND	
D12	TDI	I	H-UDI data	
D13	VSSQ	—	I/O GND	
D14	VSS	—	Internal GND	
D15	VSSQ	—	I/O GND	
D16	MD0	I	Mode control 0	
D17	MD1	I	Mode control 1	
D18	DRAK1	O	DMAC 1 request acknowledgement	
D19	DACK1	O	DMAC 1 bus acknowledgement	
D20	$\overline{\text{USB\_OVC}}$	I	USB overcurrent detection	O
E1	MFI-D8/LCD_DATA8	IO/O	MFI data/LCD panel data	O
E2	VEPWC/$\overline{\text{IRQ5}}$	O/I	LCD panel power supply control (VEE)/external interrupt request 5	O
E3	CA	I	Chip active	
E4	$\overline{\text{BREQ}}$	I	Bus request	
E17	VSSQ	—	I/O GND	
E18	VDDQ	—	I/O VDD	
E19	$\overline{\text{DREQ0}}$	I	DMAC 0 request	
E20	$\overline{\text{DREQ1}}$	I	DMAC 1 request	
F1	MFI-D9/LCD_DATA9	IO/O	MFI data/LCD panel data	O
F2	MFI-D0/LCD_DATA0	IO/O	MFI data/LCD panel data	O
F3	$\overline{\text{CS0}}$	O	Chip select 1	
F4	VCPWC/$\overline{\text{IRQ4}}$	O/I	LCD panel power supply control (VCC)/external interrupt request 4	O
F17	I2C1_SCL	IO	I^2C serial clock	
F18	I2C1_SDA	IO	I^2C serial data	
F19	I2C0_SCL	IO	I^2C serial clock	
F20	I2C0_SDA	IO	I^2C serial data	

G1	MFI-D10/LCD_DATA10	IO/O	MFI data/LCD panel data	O	
G2	MFI-D1/LCD_DATA1	IO/O	MFI data/LCD panel data	O	
G3	VDD	—	Internal VDD		
G4	VSS	—	Internal GND		
G17	VSS	—	Internal GND		
G18	VDD	—	Internal VDD		
G19	MD6/$\overline{\text{IOIS16}}$	I/I	Mode control 6/IOIS16 (PCMCIA)		
G20	MD5	I	Mode control 5		
H1	$\overline{\text{CS1}}$	O	Chip select 1		
H2	MFI-D2/LCD_DATA2/$\overline{\text{IRQ6}}$	IO/O/I	MFI data/LCD panel data/external interrupt request 6	O	
H3	VDDQ	—	IO VDD		
H4	VSSQ	—	IO GND		
H17	MD7	I	Mode control 7		
H18	MD8	I	Mode control 8		
H19	Reserved/AUDCK	O	Reserved/H-UDI emulator	O	
H20	Reserved/AUDSYNC	O	Reserved/H-UDI emulator	O	
J1	MFI-D11/LCD_DATA11	IO/O	MFI data/LCD panel data	O	
J2	MFI-D3/LCD_DATA3/$\overline{\text{IRQ7}}$	IO/O/I	MFI data/LCD panel data/external interrupt request 7	O	
J3	$\overline{\text{CS2}}$	O	Chip select 2		
J4	$\overline{\text{BACK}}$	O	Bus acknowledgement		
J17	VSSQ	—	IO GND		
J18	VDDQ	—	IO VDD		
J19	Reserved/AUDATA[2]	O	Reserved/H-UDI emulator	O	
J20	Reserved/AUDATA[3]	O	Reserved/H-UDI emulator	O	*2
K1	MFI-D12/LCD_DATA12	IO/O	MFI data/LCD panel data	O	
K2	$\overline{\text{CS4}}$	—	Chip select 4		
K3	MFI-D4/LCD_DATA4/$\overline{\text{DREQ2}}$	IO/O/I	MFI data/LCD panel data/DMAC2 request	O	
K4	VSSQ	—	IO GND		
K17	AVss_ADC	—	ADC analog GND		
K18	AVcc_ADC	—	ADC analog VCC		
K19	$\overline{\text{ADTRG}}$/AUDATA[0]	I/O	A/D external trigger/H-UDI emulator	O	
K20	Reserved/AUDATA[1]	O	Reserved/H-UDI emulator	O	

L1	MFI-D13/LCD_DATA13	IO/O	MFI data/LCD panel data	O
L2	MFI-D6/LCD_DATA6/$\overline{\text{DREQ3}}$	IO/O/I	MFI data/LCD panel data/DMAC3 request	O
L3	MFI-D5/LCD_DATA5/DRAK2/DACK2	IO/O/O/O	MFI data/LCD panel data/DMAC2 request acknowledgement/DMAC2 bus acknowledgement	O
L4	A20	O	Address bus	
L17	AN3	I	ADC analog input	
L18	AN2	I	ADC analog input	
L19	AN1	I	ADC analog input	
L20	AN0	I	ADC analog input	
M1	MFI-D14/LCD_DATA14	IO/O	MFI data/LCD panel data	O
M2	MFI-D7/LCD_DATA7/DRAK3/DACK3	IO/O/O/O	MFI data/LCD panel data/DMAC3 request acknowledgement/DMAC3 bus acknowledgement	O
M3	A21	O	Address bus	
M4	VDDQ	—	IO VDD	
M17	VSSQ	—	IO GND	
M18	VDDQ	—	IO VDD	
M19	$\overline{\text{IRL3}}$	I	IRL interrupt request 3	
M20	$\overline{\text{IRL2}}$	I	IRL interrupt request 2	
N1	$\overline{\text{CS5}}$	O	Chip select 5	
N2	$\overline{\text{MFI-INT}}$/LCD_CLK	O/I	MFI interrupt/LCD clock	O *1
N3	$\overline{\text{MFI-CS}}$/LCD_DON	I/O	MFI chip selection/LCD display-on signal	O *1
N4	VSSQ	—	IO GND	
N17	VSSQ	—	IO GND	
N18	VDDQ	—	IO VDD	
N19	$\overline{\text{IRL1}}$	I	IRL interrupt request 1	
N20	$\overline{\text{IRL0}}$	I	IRL interrupt request 0	
P1	MFI-D15/LCD_DATA15	IO/O	MFI data/LCD panel data	O
P2	$\overline{\text{CS6}}$	O	Chip select 6	
P3	VSS	—	Internal GND	
P4	VDD	—	Internal VDD	
P17	VSS	—	Internal GND	
P18	VDD	—	Internal VDD	
P19	CAN0_NERR/AUDCK	I/O	HCAN0 bus error signal/H-UDI emulator	O
P20	CAN1_NERR/AUDSYNC	I/O	HCAN1 bus error signal/H-UDI emulator	O

R1	A1	O	Address bus	
R2	$\overline{BS}$	O	Bus start	
R3	MFI-E/LCD_CL1	I/O	MFI enable/ LCD shift clock 1	O *1
R4	A0	O	Address bus	
R17	A24	O	Address bus	
R18	A25	O	Address bus	
R19	CAN0_RX/AUDATA[2]	I/O	HCAN0 bus data receive signal/H-UDI emulator	O
R20	CAN1_RX/AUDATA[3]	I/O	HCAN1 bus data receive signal/H-UDI emulator	O
T1	MFI-RW/LCD_FLM	I/O	MFI read/write/LCD first line marker	O *1
T2	MFI-RS/LCD_M_DISP	I/O	MFI register select/LCD current-alternating signal/DISP signal	O *1
T3	D11	IO	Data bus	
T4	MFI-MD/LCD_CL2	I/O	MFI mode/LCD shift clock 2	O *1
T17	A22	O	Address bus	
T18	A23	O	Address bus	
T19	CAN0_TX/AUDATA[0]	O/O	HCAN0 bus data transmit signal/H-UDI emulator	O
T20	CAN1_TX/AUDATA[1]	O/O	HCAN1 bus data transmit signal/H-UDI emulator	O
U1	D15	IO	Data bus	
U2	D0	IO	Data bus	
U3	D3	IO	Data bus	
U4	VDDQ	—	IO VDD	
U5	VSSQ	—	IO GND	
U6	CKE	O	Clock output enable	
U7	VSS	—	Internal GND	
U8	VSSQ	—	IO GND	
U9	A17	O	Address bus	
U10	VSSQ	—	IO GND	
U11	VSSQ	—	IO GND	
U12	VSSQ	—	IO GND	
U13	A18	O	Address bus	
U14	VSS	—	Internal GND	
U15	A19	O	Address bus	
U16	VSSQ	—	IO GND	

U17	D20	IO	Data bus
U18	D28	IO	Data bus
U19	D16	IO	Data bus
U20	D31	IO	Data bus
V1	D14	IO	Data bus
V2	D1	IO	Data bus
V3	VDDQ	—	IO VDD
V4	D10	IO	Data bus
V5	VDDQ	—	IO VDD
V6	A2	O	Address bus
V7	VDD	—	Internal VDD
V8	VDDQ	—	IO VDD
V9	A7	O	Address bus
V10	VDDQ	—	IO VDD
V11	VDDQ	—	IO VDD
V12	VDDQ	—	IO VDD
V13	A15	O	Address bus
V14	VDD	—	Internal VDD
V15	A16	O	Address bus
V16	VDDQ	—	IO VDD
V17	D21	IO	Data bus
V18	VDDQ	—	IO VDD
V19	D17	IO	Data bus
V20	D30	IO	Data bus
W1	D2	IO	Data bus
W2	VSSQ	—	IO GND
W3	D4	IO	Data bus
W4	D6	IO	Data bus
W5	D7	IO	Data bus
W6	RD/$\overline{WR}$	O	Read/write
W7	$\overline{WE0}$/DQM0/$\overline{REG}$	O/O	Selection signal for D7 to D0/REG
W8	A4	O	Address bus
W9	A6	O	Address bus

W10	A8	O	Address bus
W11	A10	O	Address bus
W12	A12	O	Address bus
W13	A14	O	Address bus
W14	$\overline{WE2}$/DQM2/$\overline{ICIORD}$	O/O	Selection signal for D23 to D16/ICIORD
W15	$\overline{RAS}$	O	RAS
W16	D24	IO	Data bus
W17	D25	IO	Data bus
W18	D27	IO	Data bus
W19	VSSQ	—	IO GND
W20	D29	IO	Data bus
Y1	D13	IO	Data bus
Y2	D12	IO	Data bus
Y3	D5	IO	Data bus
Y4	D9	IO	Data bus
Y5	D8	IO	Data bus
Y6	$\overline{RD}$/$\overline{CASS}$/$\overline{FRAME}$	O/O/O	Read/CAS/FRAME
Y7	$\overline{WE1}$/DQM1	O	Selection signal for D15 to D8
Y8	A3	O	Address bus
Y9	A5	—	Address bus
Y10	CKIO	O	Clock output
Y11	A9	O	Address bus
Y12	A11	O	Address bus
Y13	A13	O	Address bus
Y14	$\overline{WE3}$/DQM3/$\overline{ICIOWR}$	O/O	Selection signal for D31 to D24/ICIOWR
Y15	$\overline{CS3}$	O	Chip select 3
Y16	D23	IO	Data bus
Y17	D22	IO	Data bus
Y18	D26	IO	Data bus
Y19	D19	IO	Data bus
Y20	D18	IO	Data bus

Notes: 1. Can be used as a GPIO interrupt pin.
2. Can be used as a GPIO interrupt pin. When an interrupt occurs, this LSI exits standby mode.
3. Only outputs.

Pin No.		MFI mode (MD7=0)			LCDC mode (MD7=1)			Other modes				Register	
17 mm*1	21mm*2	Function	Pin Name	I/O	Function	Pin Name	I/O	Function	Pin Name	I/O	GPIO	GPIO Setting	IP Selection
N2	P1	MFI	$\overline{\text{MFI-INT}}$	O	LCDC	LCD_CLK	I				PTE7	PECR[15:14]	
N3	P2		$\overline{\text{MFI-CS}}$	I		LCD_DON	O				PTE6	PECR[13:12]	
R3	R1		MFI-E	I		LCD_CL1	O				PTE5	PECR[11:10]	
T4	R2		MFI-MD	I		LCD_CL2	O				PTE4	PECR[9:8]	
T2	T1		MFI-RS	I		LCD_M_DISP	O				PTE3	PECR[7:6]	
T1	T2		MFI-RW	I		LCD_FLM	O				PTE2	PECR[5:4]	
F2	F2		MFI-D0	IO		LCD_DATA0	O				PTC7	PCCR[15:14]	
G2	G2		MFI-D1	IO		LCD_DATA1	O				PTC6	PCCR[13:12]	
H2	H2		MFI-D2	IO		LCD_DATA2	O	INTC	$\overline{\text{IRQ6}}$	I	PTC5	PCCR[11:10]	MODSELR[7]
J2	J2		MFI-D3	IO		LCD_DATA3	O		$\overline{\text{IRQ7}}$	I	PTC4	PCCR[9:8]	MODSELR[6]
K3	K2		MFI-D4	IO		LCD_DATA4	O	DMAC	$\overline{\text{DREQ2}}$	I	PTC3	PCCR[7:6]	MODSELR[5]
L3	L2		MFI-D5	IO		LCD_DATA5	O		DRAK2/	O/O	PTC2	PCCR[5:4]	MODSELR[4]
									DACK2				DMARCR[22]*3
L2	M2		MFI-D6	IO		LCD_DATA6	O		$\overline{\text{DREQ3}}$	I	PTC1	PCCR[3:2]	MODSELR[3]
M2	N2		MFI-D7	IO		LCD_DATA7	O		DRAK3/	O/O	PTC0	PCCR[1:0]	MODSELR[2]
									DACK3				DMARCR[23]*4
E1	F1		MFI-D8	IO		LCD_DATA8	O				PTD7	PDCR[15:14]	IPSELR[9]
F1	G1		MFI-D9	IO		LCD_DATA9	O				PTD6	PDCR[13:12]	IPSELR[9]
G1	H1		MFI-D10	IO		LCD_DATA10	O				PTD5	PDCR[11:10]	IPSELR[9]
J1	J1		MFI-D11	IO		LCD_DATA11	O				PTD4	PDCR[9:8]	IPSELR[9]
K1	K1		MFI-D12	IO		LCD_DATA12	O				PTD3	PDCR[7:6]	IPSELR[9]
L1	L1		MFI-D13	IO		LCD_DATA13	O				PTD2	PDCR[5:4]	IPSELR[9]
M1	M1		MFI-D14	IO		LCD_DATA14	O				PTD1	PDCR[3:2]	IPSELR[9]
P1	N1		MFI-D15	IO		LCD_DATA15	O				PTD0	PDCR[1:0]	IPSELR[9]
F4	E1	INTC	$\overline{\text{IRQ4}}$	I		VCPWC	O				PTE1	PECR[3:2]	
E2	E2		$\overline{\text{IRQ5}}$	I		VEPWC	O				PTE0	PECR[1:0]	

Notes: 1. HD6417760BL200A, HD6417760BL200AV, HD6417760BL200AD, HD6417760BL200ADV
2. HD6417760BP200AD, HD6417760BP200ADV
3. DRAK2/DACK2 can be selected only in DMABRG mode
4. DRAK3/DACK3 can be selected only in DMABRG mode

Pin No.												Register	
17 mm*1	21 mm*2	Function 1	Pin Name	I/O	Function 2	Pin Name	I/O	Function 3	Pin Name	I/O	GPIO	GPIO Setting	IP Selection
F19	F19	I²C (0)	I2C0_SCL	IO									
F20	F20		I2C0_SDA	IO									
F17	F17	I²C (1)	I2C1_SCL	IO									
F18	F18		I2C1_SDA	IO									
B6	B6	HSPI	HSPI_RX	I							PTF2	PFCR[5:4]	
A6	A6		HSPI_TX	O	SIM	SIM_D	IO	MMCIF	MCDAT	IO	PTF3	PFCR[7:6]	IPSELR[15:14]
A7	A7		HSPI_CLK	IO		SIM_CLK	O		MCCLK	O	PTF1	PFCR[3:2]	IPSELR[15:14]
B7	B7		$\overline{\text{HSPI\_CS}}$	IO		SIM_RST	O		MCCMD	IO	PTF0	PFCR[1:0]	IPSELR[15:14]
P19	P19	HCAN2 (0)	CAN0_NERR	I	AUD	AUDCK	O				PTA7	PACR[15:14]	IPSELR[13]
R19	R19		CAN0_RX	I		AUDATA[2]	O				PTA6	PACR[13:12]	IPSELR[13]
T19	T19		CAN0_TX	O		AUDATA[0]	O				PTA5	PACR[11:10]	IPSELR[13]
P20	P20	HCAN2 (1)	CAN1_NERR	I		AUDSYNC	O				PTA4	PACR[9:8]	IPSELR[13]
R20	R20		CAN1_RX	I		AUDATA[3]	O				PTA3	PACR[7:6]	IPSELR[13]
T20	T20		CAN1_TX	O		AUDATA[1]	O				PTA2	PACR[5:4]	IPSELR[13]
J20	J20		Reserved*4			AUDATA[3]	O				PTK7	PKCR[15:14]	IPSELR[12]
J19	J19		Reserved*4			AUDATA[2]	O				PTK6	PKCR[13:12]	IPSELR[12]
K20	K20		Reserved*4			AUDATA[1]	O				PTK5	PKCR[11:10]	IPSELR[12]
H19	H19		Reserved*4			AUDCK	O				PTK4	PKCR[9:8]	IPSELR[12]
H20	H20		Reserved*4			AUDSYNC	O				PTK3	PKCR[7:6]	IPSELR[12]
K19	K19	ADC	$\overline{\text{ADTRG}}$	I		AUDATA[0]	O				PTK2	PKCR[5:4]	IPSELR[12]
L20	L20	ADC (0)	AN0	I									
L19	L19	ADC (1)	AN1	I									
L18	L18	ADC (2)	AN2	I									
L17	L17	ADC (3)	AN3	I									
A5	A5	SSI (0)	SSI_SCK	IO	HAC (0)	HAC_SD_IN0	I	BS*3	$\overline{\text{BS2}}$	O	PTB7	PBCR[15:14]	IPSELR[11:10]
B5	B5		SSI0_WS	IO		HAC_SYNC0	O				PTB6	PBCR[13:12]	IPSELR[11:10]
C5	C5		SSI0_SDATA	IO		HAC_SD_OUT0	O				PTB5	PBCR[11:10]	IPSELR[11:10]

Pin No.												Register	
17 mm*1	21 mm*2	Function 1	Pin Name	I/O	Function 2	Pin Name	I/O	Function 3	Pin Name	I/O	GPIO	GPIO Setting	IP Selection
C2	C2	SSI (0)	HAC_BIT_CLK0	I	HAC (0)	HAC_BIT_CLK0	I				PTJ7	PJCR[15:14]	
C4	C4	HAC (0/1)	$\overline{\text{HAC\_RES}}$	O							PTJ6	PJCR[13:12]	
D3	D3	SSI (1)	SSI1_WS	IO	HAC (1)	HAC_SYNC1	O				PTJ5	PJCR[11:10]	IPSELR[11:10]
D2	D2		SSI1_SCK	IO		HAC_SD_IN1	I				PTJ4	PJCR[9:8]	IPSELR[11:10]
C6	C6		SSI1_DATA	IO		HAC_SD_OUT	O				PTJ3	PJCR[7:6]	IPSELR[11:10]
D4	D4		HAC_BIT_CLK1	I		HAC_BIT_CLK1	I				PTJ2	PJCR[5:4]	
B8	B8	CMT (0)	CMT_CTR0	IO	TMU	TCLK	I				PTB4	PBCR[9:8]	
A8	A8	CMT (1)	CMT_CTR1	IO							PTB3	PBCR[7:6]	
B9	B9	CMT (2)	CMT_CTR2	IO							PTB2	PBCR[5:4]	
A9	A9	CMT (3)	CMT_CTR3	IO							PTB1	PBCR[3:2]	
A15	A15	SCIF (0)	SCIF0_CLK	IO							PTG7	PGCR[15:14]	
B16	B16		SCIF0_RXD	I							PTG6	PGCR[13:12]	
A16	A16		SCIF0_TXD	O							PTG5	PGCR[11:10]	
B11	B11	SCIF (1)	SCIF1_CLK	IO							PTG4	PGCR[9:8]	
B14	B14		$\overline{\text{SCIF1\_CTS}}$	IO							PTG3	PGCR[7:6]	
B15	B15		$\overline{\text{SCIF1\_RTS}}$	IO							PTG2	PGCR[5:4]	
B13	B13		SCIF1_RXD	I							PTG1	PGCR[3:2]	
B12	B12		SCIF1_TXD	O							PTG0	PGCR[1:0]	
A10	A10	SCIF (2)	SCIF2_CLK	IO							PTH7	PHCR[15:14]	
A13	A13		$\overline{\text{SCIF2\_CTS}}$	IO							PTH6	PFCR[13:12]	
A14	A14		$\overline{\text{SCIF2\_RTS}}$	IO							PTH5	PHCR[11:10]	
A12	A12		SCIF2_RXD	I							PTH4	PHCR[9:8]	
A11	A11		SCIF2_TXD	O							PTH3	PHCR[7:6]	
A20	A20	USB	UCLK	I							PTH2	PHCR[5:4]	
C18	C18		USB_PENC	O							PTH1	PHCR[3:2]	
D20	D20		$\overline{\text{USB\_OVC}}$	I							PTH0	PHCR[1:0]	
C20	C20		USB_DP	IO									
B19	B19		USB_DM	IO									

Pin No.												Register	
17 mm*1	21 mm*2	Function 1	Pin Name	I/O	Function 2	Pin Name	I/O	Function 3	Pin Name	I/O	GPIO	GPIO Setting	IP Selection
C15	C15	H-UDI	TCK	I									
C10	C10		TMS	I									
D12	D12		TDI	I									
C12	C12		TDO	O									
D10	D10		$\overline{\text{TRST}}$	I									
C8	C8		$\overline{\text{ASEBRK}}$	I	H-UDI	BRKACK	O						
B10	B10	INTC	NMI	I									
N20	N20		$\overline{\text{IRL0}}$	I									
N19	N19		$\overline{\text{IRL1}}$	I									
M20	M20		$\overline{\text{IRL2}}$	I									
M19	M19		$\overline{\text{IRL3}}$	I									
A18	A18	DMAC	DACK0	O									
D19	D19		DACK1	O									
C17	C17		DRAK0	O									
D18	D18		DRAK1	O									
E19	E19		$\overline{\text{DREQ0}}$	I									
E20	E20		$\overline{\text{DREQ1}}$	I									
D1	D1	CPG	DCK	O							PTJ1	PJCR[3]	

Notes: 1. HD6417760BL200A, HD6417760BL200AV, HD6417760BL200AD, HD6417760BL200ADV
2. HD6417760BP200AD, HD6417760BP200ADV
3. For details of the BS settings, see MODSELR[1].
4. Refer to section 24, Pin Function Controller(PFC) for processing of Reserved terminal.

Pin No.					Memory Interface				
17 mm*1	21 mm*2	Function	Pin Name	I/O	SRAM	SDRAM	PCMCIA	MPX	Remarks
R4	R4	Address	A0	O					
R1	T4		A1	O					
V6	V6		A2	O					
Y8	Y8		A3	O					
W8	W8		A4	O					
Y9	Y9		A5	O					
W9	W9		A6	O					
V9	V9		A7	O					
W10	W10		A8	O					
Y11	Y11		A9	O					
W11	W11		A10	O					
Y12	Y12		A11	O					
W12	W12		A12	O					
Y13	Y13		A13	O					
W13	W13		A14	O					
V13	V13		A15	O					
V15	V15		A16	O					
U9	U9		A17	O					
U13	U13		A18	O					
U15	U15		A19	O					
L4	L4		A20	O					
M3	N4		A21	O					
T17	T17		A22	O					
T18	T18		A23	O					
R17	R17		A24	O					
R18	R18		A25	O					
U2	U1	Data	D0	I/O				A0	
V2	V1		D1	I/O				A1	
W1	W1		D2	I/O				A2	
U3	U3		D3	I/O				A3	
W3	W3		D4	I/O				A4	

17 mm*	21 mm*	Function	Pin Name	I/O	SRAM	SDRAM	PCMCIA	MPX	Remarks
Y3	Y3	Data	D5	I/O				A5	
W4	W4		D6	I/O				A6	
W5	W5		D7	I/O				A7	
Y5	Y5		D8	I/O				A8	
Y4	Y4		D9	I/O				A9	
V4	V4		D10	I/O				A10	
T3	U4		D11	I/O				A11	
Y2	Y2		D12	I/O				A12	
Y1	Y1		D13	I/O				A13	
V1	V2		D14	I/O				A14	
U1	U2		D15	I/O				A15	
U19	U19		D16	I/O				A16	
V19	V19		D17	I/O				A17	
Y20	Y20		D18	I/O				A18	
Y19	Y19		D19	I/O				A19	
U17	U17		D20	I/O				A20	
V17	V17		D21	I/O				A21	
Y17	Y17		D22	I/O				A22	
Y16	Y16		D23	I/O				A23	
W16	W16		D24	I/O				A24	
W17	W17		D25	I/O				A25	
Y18	Y18		D26	I/O					
W18	W18		D27	I/O					
U18	U18		D28	I/O					
W20	W20		D29	I/O				ACCSIZE 0	
V20	V20		D30	I/O				ACCSIZE 1	
U20	U20		D31	I/O				ACCSIZE 2	
F3	F3	Chip select	$\overline{CS0}$	O	$\overline{CS0}$			$\overline{CS0}$	
H1	J3		$\overline{CS1}$	O	$\overline{CS1}$			$\overline{CS1}$	
J3	J4		$\overline{CS2}$	O	$\overline{CS2}$	$\overline{CS2}$		$\overline{CS2}$	

17 mm*	21 mm*	Function	Pin Name	I/O	SRAM	SDRAM	PCMCIA	MPX	Remarks
Y15	Y15	Chip select	$\overline{CS3}$	O	$\overline{CS3}$	$\overline{CS3}$		$\overline{CS3}$	
K2	L3		$\overline{CS4}$	O	$\overline{CS4}$			$\overline{CS4}$	
N1	N3		$\overline{CS5}$	O	$\overline{CS5}$		$\overline{CE1A}$	$\overline{CS5}$	
P2	R3		$\overline{CS6}$	O	$\overline{CS6}$		$\overline{CE1B}$	$\overline{CS6}$	
W6	W6	Read/Write	RD/$\overline{WR}$	O	RD/$\overline{WR}$	RD/$\overline{WR}$		RD/$\overline{WR}$	
W15	W15	$\overline{RAS}$	$\overline{RAS}$	O		$\overline{RAS}$			
Y6	Y6	Read/$\overline{CAS}$/$\overline{FRAME}$	$\overline{RD}$/$\overline{CASS}$/ $\overline{FRAME}$	O	$\overline{OE}$	$\overline{CAS}$	$\overline{OE}$	$\overline{FRAME}$	
W7	W7	Selection signal for D7 to D0/$\overline{REG}$	$\overline{WE0}$/DQM0/ $\overline{REG}$	O	$\overline{WE0}$	DQM0	$\overline{REG}$		
Y7	Y7	Selection signal for D15 to D8	$\overline{WE1}$/DQM1	O	$\overline{WE1}$	DQM1	$\overline{WE1}$		
W14	W14	Selection signal for D23 to D16/$\overline{ICIORD}$	$\overline{WE2}$/DQM2/ $\overline{ICIORD}$	O	$\overline{WE2}$	DQM2	$\overline{ICIORD}$		
Y14	Y14	Selection signal for D31 to D24/$\overline{ICIOWR}$	$\overline{WE3}$/DQM3/ $\overline{ICIOWR}$	O	$\overline{WE3}$	DQM3	$\overline{ICIOWR}$		
Y10	Y10	Clock output	CKIO	O		CKIO		CKIO	
U6	U6	Clock output enable	CKE	O		CKE			
D16	D16	Mode	MD0	I					Reset; MD0
D17	D17	Mode	MD1	I					Reset; MD1
C16	C16	Mode	MD2	I					Reset; MD2
B17	B17	Mode/PCMCIA-CE	MD3/$\overline{CE2A}$	I/O			$\overline{CE2A}$		Reset; MD3
A17	A17	Mode/PCMCIA-CE	MD4/$\overline{CE2B}$	I/O			$\overline{CE2B}$		Reset; MD4
G20	G20	Mode	MD5	I					Reset; MD5
G19	G19	Mode/$\overline{IOIS16}$	MD6/$\overline{IOIS16}$	I/I			$\overline{IOIS16}$		Reset; MD6
H17	H17	Mode	MD7	I					Reset; MD7
H18	H18	Mode	MD8	I					Reset; MD8
B1	B1	Reset	$\overline{RESET}$	I			$\overline{RESET}$		
D5	D5	Manual reset	$\overline{MRESET}$	I					
C1	C1	Bus ready	$\overline{RDY}$	I	$\overline{RDY}$		$\overline{RDY}$	$\overline{RDY}$	
R2	T3	Bus start	$\overline{BS}$	O	($\overline{BS}$)	($\overline{BS}$)	($\overline{BS}$)	($\overline{BS}$)	
E4	E4	Bus request	$\overline{BREQ}$	I					
J4	F4	Bus acknowledgement	$\overline{BACK}$	O					

17 mm*	21 mm*	Function	Pin Name	I/O	SRAM	SDRAM	PCMCIA	MPX	Remarks
E3	E3	Chip active	CA	I					
D6	D6	Status 0	STATUS0	O					
D8	D8	Status 1	STATUS1	O					
A1	A1	External input clock/crystal resonator	EXTAL	I					
A2	A2	Crystal resonator	XTAL	O					

Notes: 1. HD6417760BL200A, HD6417760BL200AV, HD6417760BL200AD, HD6417760BL200ADV

2. HD6417760BP200AD, HD6417760BP200ADV

2.1 Data Formats

The data formats supported in the SH-4 are shown in figure 2.1.

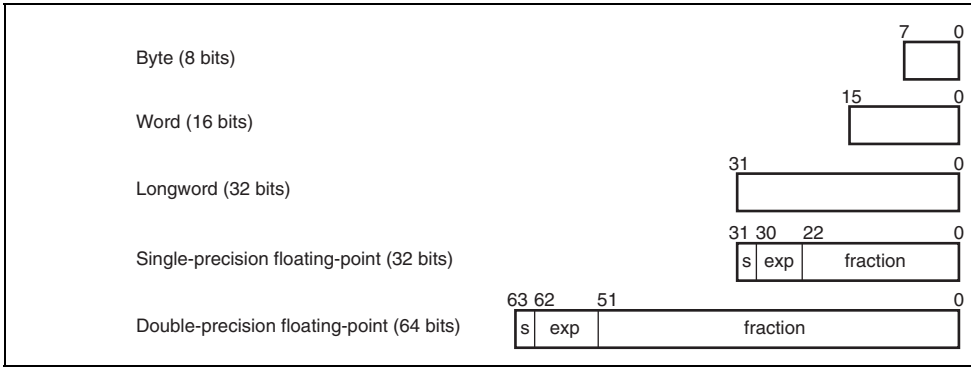


Figure 2.1 Data Formats

2.2.1 Privileged Mode and Banks

Processor Modes: This LSI has two processor modes, user mode and privileged mode. This LSI normally operates in user mode, and switches to privileged mode when an exception occurs or an interrupt is accepted. There are four kinds of registers—general registers, system registers, control registers, and floating-point registers—and the registers that can be accessed differ in the two processor modes.

General Registers: There are 16 general registers, designated R0 to R15. General registers R0 to R7 are banked registers which are switched by a processor mode change.

In privileged mode, the register bank bit (RB) in the status register (SR) defines which banked register set is accessed as general registers, and which set is accessed only through the load control register (LDC) and store control register (STC) instructions.

When the RB bit is 1 (that is, when bank 1 is selected), the 16 registers comprising bank 1 general registers R0_BANK1 to R7_BANK1 and non-banked general registers R8 to R15 can be accessed as general registers R0 to R15. In this case, the eight registers comprising bank 0 general registers R0_BANK0 to R7_BANK0 are accessed by the LDC/STC instructions. When the RB bit is 0 (that is, when bank 0 is selected), the 16 registers comprising bank 0 general registers R0_BANK0 to R7_BANK0 and non-banked general registers R8 to R15 can be accessed as general registers R0 to R15. In this case, the eight registers comprising bank 1 general registers R0_BANK1 to R7_BANK1 are accessed by the LDC/STC instructions.

In user mode, the 16 registers comprising bank 0 general registers R0_BANK0 to R7_BANK0 and non-banked general registers R8 to R15 can be accessed as general registers R0 to R15. The eight registers comprising bank 1 general registers R0_BANK1 to R7_BANK1 cannot be accessed.

Control Registers: Control registers comprise the global base register (GBR) and status register (SR), which can be accessed in both processor modes, and the saved status register (SSR), saved program counter (SPC), vector base register (VBR), saved general register 15 (SGR), and debug base register (DBR), which can only be accessed in privileged mode. Some bits of the status register (such as the RB bit) can only be accessed in privileged mode.

System Registers: System registers comprise the multiply-and-accumulate registers (MACH/MACL), the procedure register (PR), the program counter (PC), the floating-point status/control register (FPSCR), and the floating-point communication register (FPUL). Access to these registers does not depend on the processor mode.

FPR15_BANK0 or FPR0_BANK1–FPR15_BANK1).

FR0–FR15 can be used as the eight registers DR0/2/4/6/8/10/12/14 (double-precision floating-point registers, or pair registers) or the four registers FV0/4/8/12 (register vectors), while XF0–XF15 can be used as the eight registers XD0/2/4/6/8/10/12/14 (register pairs) or register matrix XMTRX.

Register values after a reset are shown in table 2.1.

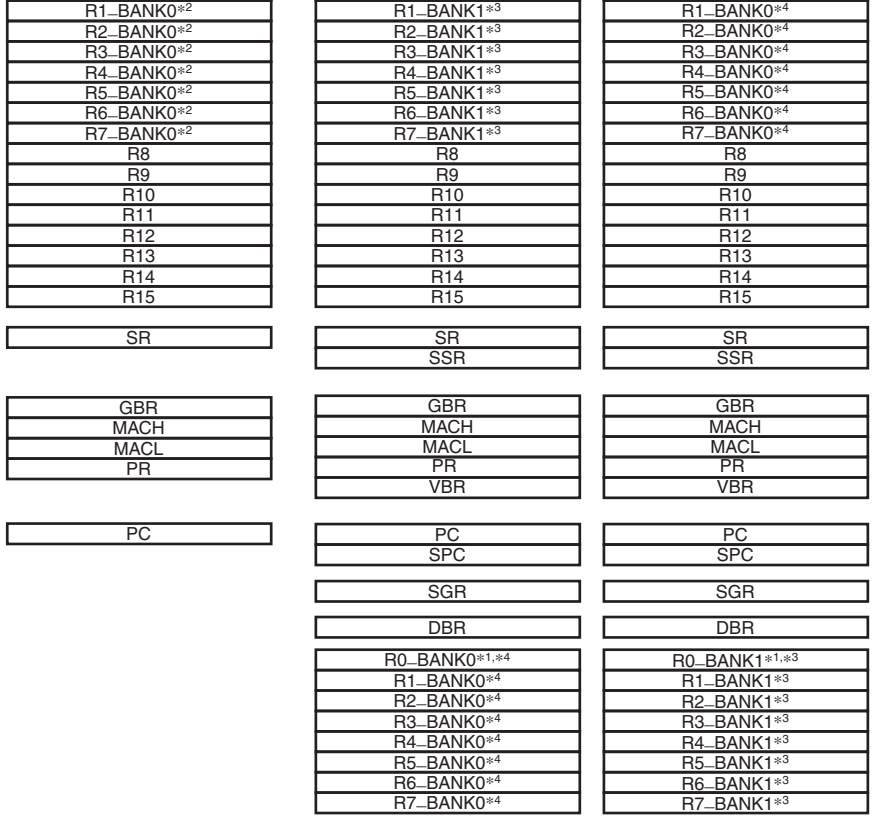
Table 2.1 Initial Register Values

Type	Registers	Initial Value*
General registers	R0_BANK0 to R7_BANK0, R0_BANK1 to R7_BANK1, R8 to R15	Undefined
Control registers	SR	MD bit = 1, RB bit = 1, BL bit = 1, FD bit = 0, IMASK = 1111 (H'F), reserved bits = 0, others = undefined
	GBR, SSR, SPC, SGR, DBR	Undefined
	VBR	H'0000 0000
System registers	MACH, MACL, PR, FPUL	Undefined
	PC	H'A000 0000
	FPSCR	H'0004 0001
Floating-point registers	FR0 to FR15, XF0 to XF15	Undefined

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

The CPU register configuration in each processing mode is shown in figure 2.2.

User mode and privileged mode are switched by the processing mode bit (MD) in the status register.



(a) Register configuration in user mode

(b) Register configuration in privileged mode (RB = 1)

(c) Register configuration in privileged mode (RB = 0)

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

Figure 2.2 CPU Register Configuration in Each Processing Mode

Figure 2.3 shows the relationship between the processing modes and general registers. The SH-4 has twenty-four 32-bit general registers (R0_BANK0 to R7_BANK0, R0_BANK1 to R7_BANK1, and R8 to R15). However, only 16 of these can be accessed as general registers R0 to R15 in one processing mode. The SH-4 has two processing modes, user mode and privileged mode.

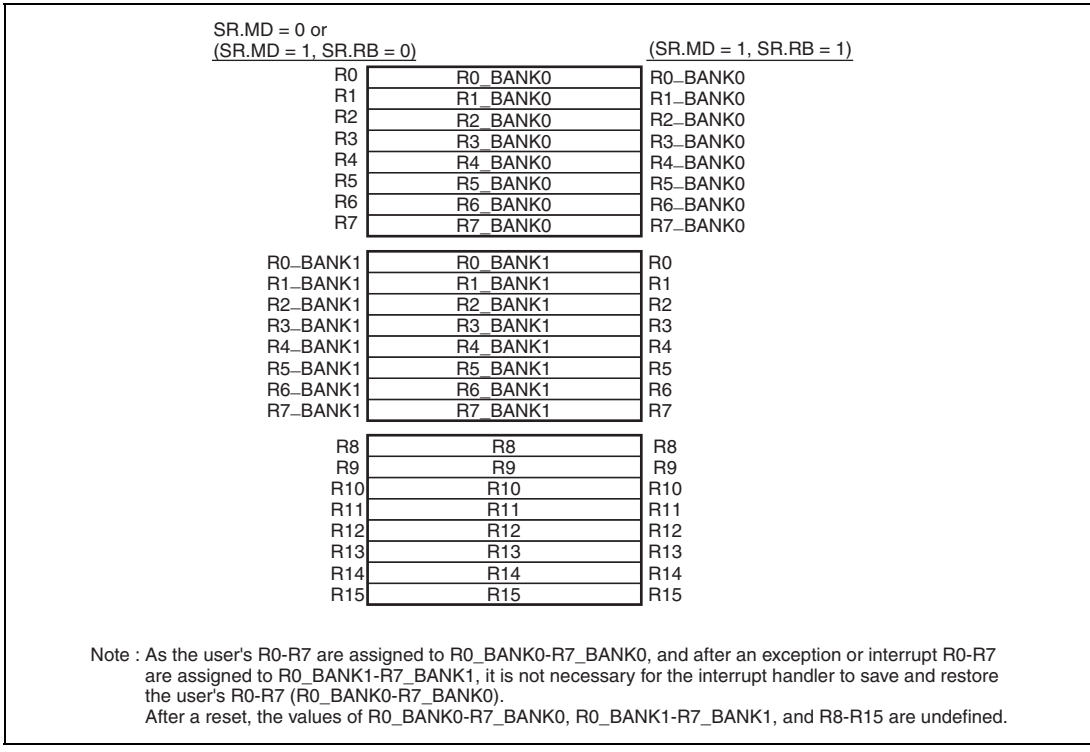
- R0_BANK0 to R7_BANK0

  Allocated to R0 to R7 in user mode (SR.MD = 0)

  Allocated to R0 to R7 when SR.RB = 0 in privileged mode (SR.MD = 1).
- R0_BANK1 to R7_BANK1

  Cannot be accessed in user mode.

  Allocated to R0 to R7 when SR.RB = 1 in privileged mode.



Note : As the user's R0-R7 are assigned to R0_BANK0-R7_BANK0, and after an exception or interrupt R0-R7 are assigned to R0_BANK1-R7_BANK1, it is not necessary for the interrupt handler to save and restore the user's R0-R7 (R0_BANK0-R7_BANK0).
After a reset, the values of R0_BANK0-R7_BANK0, R0_BANK1-R7_BANK1, and R8-R15 are undefined.

Figure 2.3 General Registers

The control registers are 32 bits long. They consist of the status register (SR), global base register (GBR), saved status register (SSR), saved program counter (SPC), vector base register (VBR), saved general register 15 (SGR), and debug base register (DBR). SR and GBR can be accessed in both processing modes, but SSR, SPC, VBR, SGR, and DBR can only be accessed in privileged mode.

Status Register (SR):

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	MD	RB	BL	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	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
	FD	-	-	-	-	-	M	Q	IMASK3	IMASK2	IMASK1	IMASK0	-	-	S	T
Initial value:	0	0	0	0	0	0	-	-	1	1	1	1	0	0	-	-
R/W:	R/W	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
30	MD	1	R/W	Processing Mode Selects the processing mode. 0: User mode (Some instructions cannot be executed and some resources cannot be accessed.) 1: Privileged mode
29	RB	1	R/W	Privileged Mode General Register Bank Soecification Bit This bit is set to 1 by an exception or interrupt. 0: R0_BANK0 to R7_BANK0 are accessed as general registers R0 to R7 and R0_BANK1 to R7_BANK1 can be accessed using LDC/STC instructions 1: R0_BANK1 to R7_BANK1 are accessed as general r can be accessed using LDC/STC instructions
28	BL	1	R/W	Exception/Interrupt Block Bit This bit is set to 1 by a reset, an exception, or an interrupt. While this bit is set to 1, an interrupt request is masked. In this case, this processor enters the reset state when a general exception other than a user break occurs.

27 to 16	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
15	FD	0	R/W	FPU Disable Bit A reset clears this bit to 0. When this bit is set to 1 and an FPU instruction is not in a delay slot, a general FPU disable exception occurs. When this bit is set to 1 and an FPU instruction is in a delay slot, a slot FPU disable exception occurs. (FPU instructions: H'F*** instructions and LDS (.L)/STS(.L) instructions using FPUL/FPSCR)
14 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9	M	—	R/W	M Bit
8	Q	—	R/W	Q Bit Used by the DIV0S, DIV0U, and DIV1 instructions.
7	IMASK3	1	R/W	Interrupt Mask Level Bits An interrupt whose priority is equal to or less than the value of the IMASK bits is masked. These bits are not modified by an interrupt.
6	IMASK2	1	R/W	
5	IMASK1	1	R/W	
4	IMASK0	1	R/W	
3, 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1	S	—	R/W	S Bit Used by the MAC instruction.
0	T	—	R/W	T Bit Indicates true/false or carry/borrow.

Saved Status Register (SSR): The contents of SR are saved to SSR in the event of an exception or interrupt.

Saved Program Counter (SPC): The address of an instruction at which an interrupt or exception occurs is saved to SPC.

Global Base Register (GBR): GBR is referenced as the base address in a GBR-referencing MOV instruction.

Vector Base Register (VBR): VBR is referenced as the branch destination base address in the event of an exception or interrupt.

exception or interrupt.

Debug Base Register (DBR): When the user break debugging function is enabled (BRCR.UBDE = 1), DBR is referenced as the branch destination address of the user break handler instead of VBR.

2.2.4 System Registers

The system registers are 32 bits long. They consist of two multiply-and-accumulate registers (MACH and MACL), the procedure register (PR), the program counter (PC), the floating-point status/control register (FPSCR), and the floating-point communication register (FPUL). For details on FPSCR and FPUL, see section 3, Floating-Point Unit (FPU).

Multiply-and-Accumulate Registers (MACH and MACL): MACH and MACL are used for the added value in a MAC instruction, and to store the operation result of a MAC or MUL instruction.

Procedure Register (PR): The return address is stored in PR in a subroutine call using a BSR, BSRF, or JSR instruction. PR is referenced by the subroutine return instruction (RTS).

Program Counter (PC): PC indicates the address of the instruction currently being executed.

2.2.5 FPU Registers

See section 3, Floating-Point Unit (FPU).

2.3 Memory-Mapped Registers

For details on the control registers mapped to memory, see section 32, List of Registers. The control registers are double-mapped to the following two memory areas. All registers have two addresses.

H'1C00 0000 to H'1FFF FFFF
H'FC00 0000 to H'FFFF FFFF

These two areas are used as follows.

- H'1C00 0000 to H'1FFF FFFF

  This area must be accessed using the address translation function of the MMU.

  Setting the page number of this area to the corresponding field of the TLB enables access to a memory-mapped register.

  The operation of an access to this area without using the address translation function of the MMU is not guaranteed.

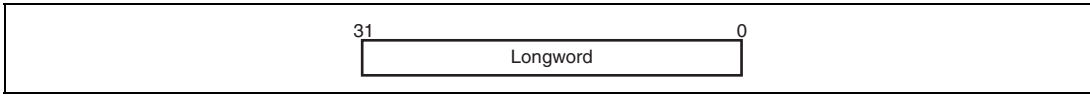
Access to area H'FC00 0000 to H'FFFF FFFF in user mode will cause an address error. Memory-mapped registers can be referenced in user mode by means of access that involves address translation.

Note: Do not access addresses to which registers are not mapped in either area. The operation of an access to an address with no register mapped is undefined. Also, memory-mapped registers must be accessed using a fixed data size. The operation of an access using an invalid data size is undefined.

2.4 Data Formats

2.4.1 Data Format in Registers

Register operands are always longwords (32 bits). When a 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.



2.4.2 Data Formats in Memory

Memory data formats are classified into bytes, words, and longwords. Memory can be accessed in an 8-bit byte, 16-bit word, or 32-bit longword form. A memory operand less than 32 bits in length is sign-extended before being loaded into a register.

A word operand must be accessed starting from a word boundary (even address of a 2-byte unit: address 2n), and a longword operand starting from a longword boundary (even address of a 4-byte unit: address 4n). An address error will result if this rule is not observed. A byte operand can be accessed from any address.

Big endian or little endian byte order can be selected for the data format. The endian should be set with the MD5 external pin after a power-on reset. Big endian is selected when the MD5 pin is low, and little endian when high. The endian cannot be changed dynamically. Bit positions are numbered left to right from most-significant to least-significant. Thus, in a 32-bit longword, the leftmost bit, bit 31, is the most significant bit and the rightmost bit, bit 0, is the least significant bit.

The data format in memory is shown in figure 2.4.

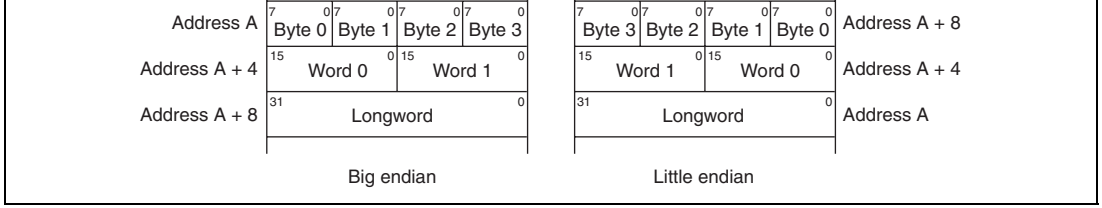


Figure 2.4 Data Formats in Memory

Note: The SH-4 does not support endian conversion for the 64-bit data format. Therefore, if double-precision floating-point format (64-bit) access is performed in little endian mode, the upper and lower 32 bits will be reversed.

2.5 Processing States

This LSI has five processing states: the reset state, exception-handling state, bus-released state, program execution state, and power-down state.

Reset State: In this state the CPU is reset. The power-on reset state is entered when the $\overline{\text{RESET}}$ pin goes low. The manual reset state is entered when the $\overline{\text{RESET}}$ pin is high and the $\overline{\text{MRESET}}$ pin is low. For more information on resets, see section 8, Exceptions.

In the power-on reset state, the internal state of the CPU and the on-chip peripheral module registers are initialized. In the manual reset state, the internal state of the CPU and registers of on-chip peripheral modules other than the BSC are initialized. Since the BSC is not initialized in the manual reset state, refreshing operations continue. For details, see register descriptions for each section.

Exception-Handling State: This is a transient state during which the CPU's processing state flow is altered by a reset, general exception, or interrupt exception handling source.

In the case of a reset, the CPU branches to address H'A000 0000 and starts executing the user-coded exception handling program.

In the case of a general exception or interrupt, the PC is saved in the SPC, the SR is saved in the SSR, and the R15 is saved in SGR. The CPU branches to the start address of the user-coded exception handling routine found from the sum of the contents of the vector base address and the vector offset. See section 8, Exceptions, for more information on resets, general exceptions, and interrupts.

Program Execution State: In this state, the CPU executes program instructions in sequence.

modes in the power-down state: sleep mode, deep sleep mode, and standby mode. For details on power-down states, see section 14, Power-Down Modes.

Bus-Released State: In this state, the CPU has released the bus to a device that requested it.

Transitions between the states are shown in figure 2.5.

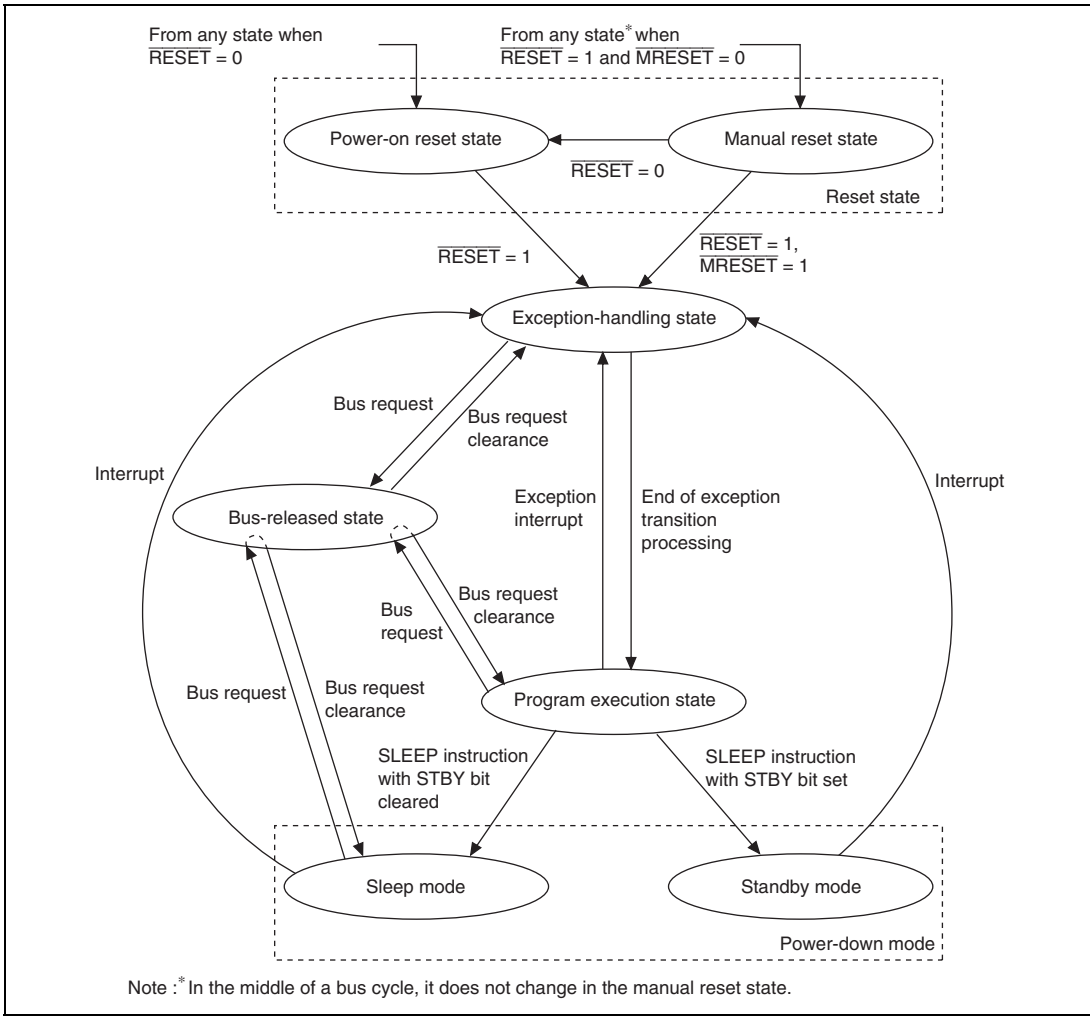


Figure 2.5 Processing State Transitions

There are two processing modes: user mode and privileged mode. The processing mode is determined by the processing mode bit (MD) in the status register (SR). User mode is selected when the MD bit is cleared to 0, and privileged mode when the MD bit is set to 1. When the reset state or exception state is entered, the MD bit is set to 1. When exception handling ends, the MD bit is cleared to 0 and user mode is entered. There are certain registers and bits which can only be accessed in privileged mode.

3.1 Features

The FPU has the following features.

- Designed to meet IEEE754 standard
- 32 single-precision floating-point registers (can also be referenced as 16 double-precision registers)
- Two rounding modes: Round to Nearest and Round to Zero
- Two denormalization modes: Flush to Zero and Treat Denormalized Number
- Six exception sources: FPU Error, Invalid Operation, Divide By Zero, Overflow, Underflow, and Inexact
- Comprehensive instructions: Single-precision, double-precision, graphics support, and system control

When the FD bit in SR is set to 1, the FPU cannot be used, and an attempt to execute an FPU instruction will cause an FPU disable exception.

3.2 Data Formats

3.2.1 Floating-Point Format

A floating-point number consists of the following three fields:

- Sign (s)
- Exponent (e)
- Fraction (f)

The FPU can handle single-precision and double-precision floating-point numbers, using the formats shown in figures 3.1 and 3.2.

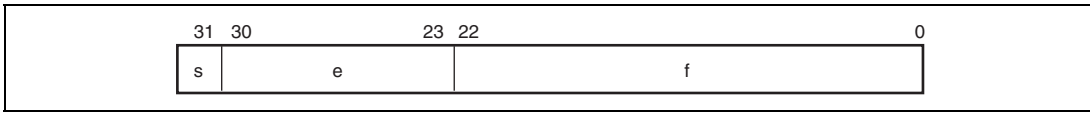


Figure 3.1 Format of Single-Precision Floating-Point Number

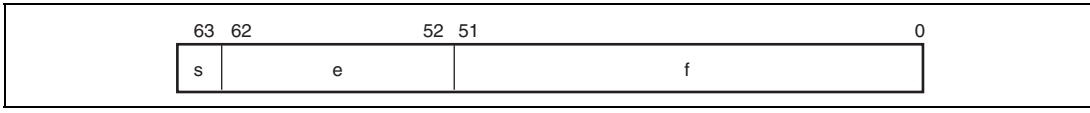


Figure 3.2 Format of Double-Precision Floating-Point Number

$e = E + bias$

The range of unbiased exponent E is $E_{min} - 1$ to $E_{max} + 1$. The two values $E_{min} - 1$ and $E_{max} + 1$ are distinguished as follows. $E_{min} - 1$ indicates zero (both positive and negative sign) and a denormalized number, and $E_{max} + 1$ indicates positive or negative infinity or a non-number (NaN). Table 3.1 shows E_{min} and E_{max} values.

Table 3.1 Floating-Point Number Formats and Parameters

Parameter	Single-Precision	Double-Precision
Total bit width	32 bits	64 bits
Sign bit	1 bit	1 bit
Exponent field	8 bits	11 bits
Fraction field	23 bits	52 bits
Precision	24 bits	53 bits
Bias	+127	+1023
E_{max}	+127	+1023
E_{min}	–126	–1022

Floating-point number value v is determined as follows:

If $E = E_{max} + 1$ and $f \neq 0$, v is a non-number (NaN) irrespective of sign s
If $E = E_{max} + 1$ and $f = 0$, $v = (-1)^s$ (infinity) [positive or negative infinity]
If $E_{min} \leq E \leq E_{max}$, $v = (-1)^s 2^E$ (1.f) [normalized number]
If $E = E_{min} - 1$ and $f \neq 0$, $v = (-1)^s 2^{Emin}$ (0.f) [denormalized number]
If $E = E_{min} - 1$ and $f = 0$, $v = (-1)^s 0$ [positive or negative zero]

Table 3.2 shows the ranges of the various numbers in hexadecimal notation.

Type	Single-Precision	Double-Precision
Signaling non-number	H'7FFF FFFF to H'7FC0 0000	H'7FFF FFFF FFFF FFFF to H'7FF8 0000 0000 0000
Quiet non-number	H'7FBF FFFF to H'7F80 0001	H'7FF7 FFFF FFFF FFFF to H'7FF0 0000 0000 0001
Positive infinity	H'7F80 0000	H'7FF0 0000 0000 0000
Positive normalized number	H'7F7F FFFF to H'0080 0000	H'7FEF FFFF FFFF FFFF to H'0010 0000 0000 0000
Positive denormalized number	H'007F FFFF to H'0000 0001	H'000F FFFF FFFF FFFF to H'0000 0000 0000 0001
Positive zero	H'0000 0000	H'0000 0000 0000 0000
Negative zero	H'8000 0000	H'8000 0000 0000 0000
Negative denormalized number	H'8000 0001 to H'807F FFFF	H'8000 0000 0000 0001 to H'800F FFFF FFFF FFFF
Negative normalized number	H'8080 0000 to H'FF7F FFFF	H'8010 0000 0000 0000 to H'FFEF FFFF FFFF FFFF
Negative infinity	H'FF80 0000	H'FFF0 0000 0000 0000
Quiet non-number	H'FF80 0001 to H'FFBF FFFF	H'FFF0 0000 0000 0001 to H'FFF7 FFFF FFFF FFFF
Signaling non-number	H'FFC0 0000 to H'FFFF FFFF	H'FFF8 0000 0000 0000 to H'FFFF FFFF FFFF FFFF

3.2.2 Non-Numbers (NaN)

Figure 3.3 shows the bit pattern of a non-number (NaN). A value is NaN in the following case:

- Sign bit: Don't care
- Exponent field: All bits are 1
- Fraction field: At least one bit is 1

The NaN is a signaling NaN (sNaN) if the MSB of the fraction field is 1, and a quiet NaN (qNaN) if the MSB is 0.

x	11111111	Nxxxxxxxxxxxxxxxxxxxxx

N = 1:sNaN
N = 0:qNaN

Figure 3.3 Single-Precision NaN Bit Pattern

An sNaN is input in an operation, except copy, FABS, and FNEG, that generates a floating-point value.

- When the EN.V bit in FPSCR is 0, the operation result (output) is a qNaN.
- When the EN.V bit in FPSCR is 1, an invalid operation exception will be generated. In this case, the contents of the operation destination register are unchanged.

If a qNaN is input in an operation that generates a floating-point value, and an sNaN has not been input in that operation, the output will always be a qNaN irrespective of the setting of the EN.V bit in FPSCR. An exception will not be generated in this case.

The qNAN values as operation results are as follows:

- Single-precision qNaN: H'7FBF FFFF
- Double-precision qNaN: H'7FF7 FFFF FFFF FFFF

See the individual instruction descriptions for details of floating-point operations when a non-number (NaN) is input.

3.2.3 Denormalized Numbers

For a denormalized number floating-point value, the exponent field is expressed as 0, and the fraction field as a non-zero value.

When the DN bit in FPSCR of the FPU is 1, a denormalized number (source operand or operation result) is always flushed to 0 in a floating-point operation that generates a value (an operation other than copy, FNEG, or FABS).

When the DN bit in FPSCR is 0, a denormalized number (source operand or operation result) is processed as it is. See the individual instruction descriptions for details of floating-point operations when a denormalized number is input.

3.3.1 Floating-Point Registers

Figure 3.4 shows the floating-point register configuration. There are thirty-two 32-bit floating-point registers, referenced by specifying FR0 to FR15, DR0/2/4/6/8/10/12/14, FV0/4/8/12, XF0 to XF15, XD0/2/4/6/8/10/12/14, or XMTRX.

1. Floating-point registers, FPRi_BANKj (32 registers)
   FPR0_BANK0 to FPR15_BANK0
   FPR0_BANK1 to FPR15_BANK1
2. Single-precision floating-point registers, FRi (16 registers)
   When FPSCR.FR = 0, FR0 to FR15 indicate FPR0_BANK0 to FPR15_BANK0;
   when FPSCR.FR = 1, FR0 to FR15 indicate FPR0_BANK1 to FPR15_BANK1.
3. Double-precision floating-point registers, DRi (8 registers): A DR register comprises two FR registers.
   DR0 = {FR0, FR1}, DR2 = {FR2, FR3}, DR4 = {FR4, FR5}, DR6 = {FR6, FR7},
   DR8 = {FR8, FR9}, DR10 = {FR10, FR11}, DR12 = {FR12, FR13}, DR14 = {FR14, FR15}
4. Single-precision floating-point vector registers, FVi (4 registers): An FV register comprises four FR registers.
   FV0 = {FR0, FR1, FR2, FR3}, FV4 = {FR4, FR5, FR6, FR7},
   FV8 = {FR8, FR9, FR10, FR11}, FV12 = {FR12, FR13, FR14, FR15}
5. Single-precision floating-point extended registers, XFi (16 registers)
   When FPSCR.FR = 0, XF0 to XF15 indicate FPR0_BANK1 to FPR15_BANK1;
   when FPSCR.FR = 1, XF0 to XF15 indicate FPR0_BANK0 to FPR15_BANK0.
6. Double-precision floating-point extended registers, XDi (8 registers): An XD register comprises two XF registers.
   XD0 = {XF0, XF1}, XD2 = {XF2, XF3}, XD4 = {XF4, XF5}, XD6 = {XF6, XF7},
   XD8 = {XF8, XF9}, XD10 = {XF10, XF11}, XD12 = {XF12, XF13}, XD14 = {XF14, XF15}
7. Single-precision floating-point extended register matrix, XMTRX: XMTRX comprises all 16 XF registers.

$$\text{XMTRX} = \begin{bmatrix} \text{XF0} & \text{XF4} & \text{XF8} & \text{XF12} \\ \text{XF1} & \text{XF5} & \text{XF9} & \text{XF13} \\ \text{XF2} & \text{XF6} & \text{XF10} & \text{XF14} \\ \text{XF3} & \text{XF7} & \text{XF11} & \text{XF15} \end{bmatrix}$$

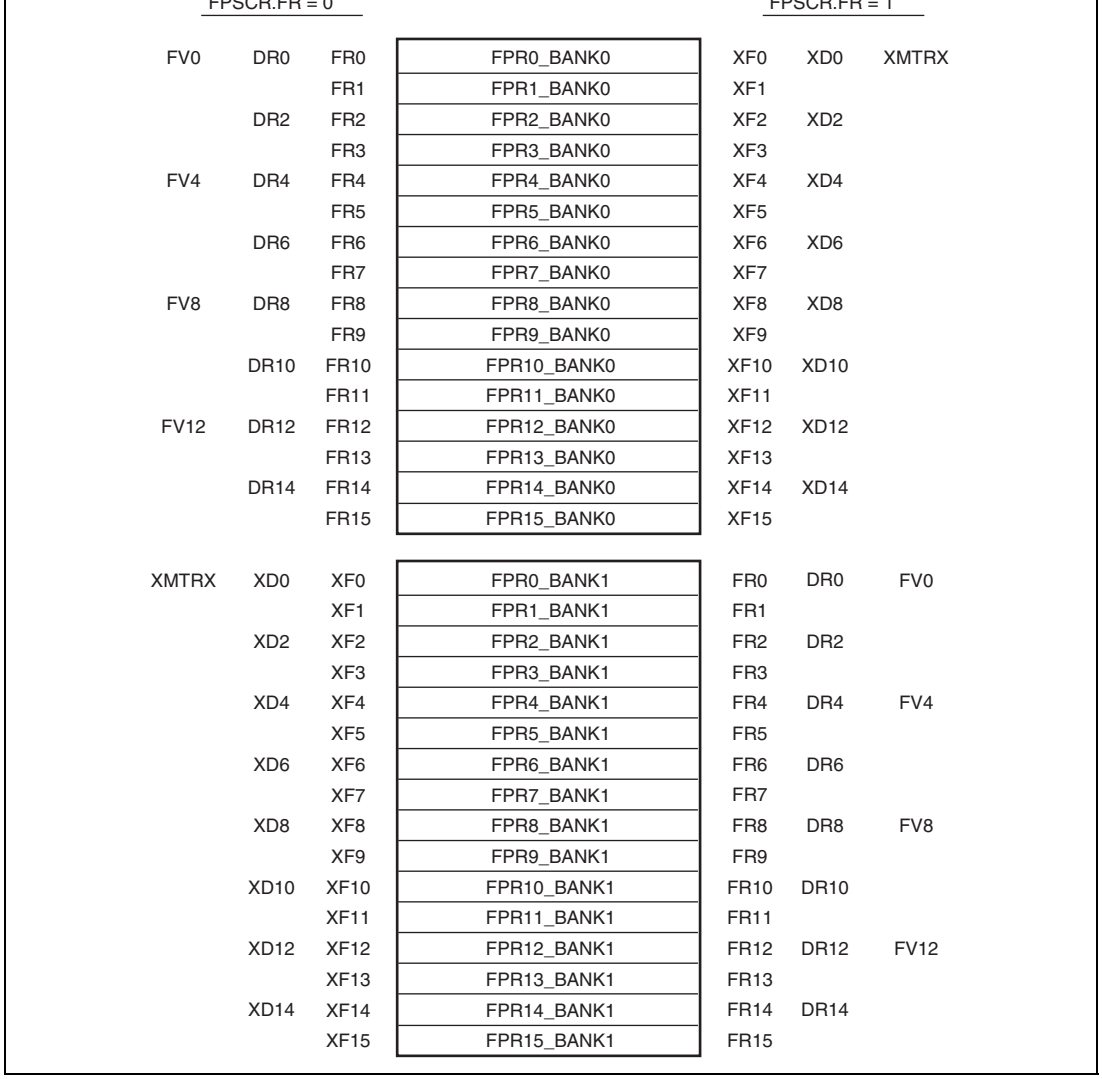


Figure 3.4 Floating-Point Registers

FPSCR is a 32-bit register that controls floating-point instructions, sets FPU exceptions, and selects the rounding mode. Do not set the SZ and PR bits to 1 simultaneously; this setting is reserved.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	FR	SZ	PR	DN	Cause	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	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
	Cause				Enable					Flag					RM1	RM0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	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

Bit	Bit Name	Initial Value	R/W	Description
31 to 22	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
21	FR	0	R/W	Floating-Point Register Bank 0: FPR0_BANK0 to FPR15_BANK0 are assigned to FR0 to FR15 and FPR0_BANK1 to FPR15_BANK1 are assigned to XF0 to XF15 1: FPR0_BANK0 to FPR15_BANK0 are assigned to XF0 to XF15 and FPR0_BANK1 to FPR15_BANK1 are assigned to FR0 to FR15
20	SZ	0	R/W	Transfer Size Mode 0: Data size of FMOV instruction is 32-bits 1: Data size of FMOV instruction is a 32-bit register pair (64 bits)
19	PR	0	R/W	Precision Mode 0: Floating-point instructions are executed as single-precision operations 1: Floating-point instructions are executed as double-precision operations (graphics support instructions are undefined)
18	DN	1	R/W	Denormalization Mode 0: Denormalized number is treated as such 1: Denormalized number is treated as zero

17 to 12	Cause	All 0	R/W	FPU Exception Cause Field
11 to 7	Enable	All 0	R/W	FPU Exception Enable Field
6 to 2	Flag	All 0	R/W	FPU Exception Flag Field When an FPU exception occurs, the bits corresponding to the FPU exception cause field and FPU exception flag field are set to 1. Each time an FPU operation instruction is executed, the FPU exception cause field is cleared to 0. The FPU exception flag field remains set to 1 until it is cleared to 0 by software. For bit allocations of each field, see table 3.3.
1	RM1	0	R/W	Rounding Mode
0	RM0	1	R/W	These bits select the rounding mode. 00: Round to Nearest 01: Round to Zero 10: Reserved 11: Reserved

Table 3.3 Bit Allocation for FPU Exception Handling

Field Name		FPU Error (E)	Invalid Operation (V)	Division by Zero (Z)	Overflow (O)	Underflow (U)	Inexact (I)
Cause	FPU exception cause field	Bit 17	Bit 16	Bit 15	Bit 14	Bit 13	Bit 12
Enable	FPU exception enable field	None	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7
Flag	FPU exception flag field	None	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2

3.3.3 Floating-Point Communication Register (FPUL)

Information is transferred between the FPU and CPU via FPUL. FPUL is a 32-bit system register that is accessed from the CPU side by means of LDS and STS instructions. For example, to convert the integer stored in general register R1 to a single-precision floating-point number, the processing flow is as follows:

R1 → (LDS instruction) → FPUL → (single-precision FLOAT instruction) → FR1

In a floating-point instruction, rounding is performed when generating the final operation result from the intermediate result. Therefore, the result of combination instructions such as FMAC, FTRV, and FIPR will differ from the result when using a basic instruction such as FADD, FSUB, or FMUL. Rounding is performed once in FMAC, but twice in FADD, FSUB, and FMUL.

Which of the two rounding methods is to be used is determined by the RM bits in FPSCR.

FPSCR.RM[1:0] = 00: Round to Nearest
FPSCR.RM[1:0] = 01: Round to Zero

Round to Nearest: The operation result is rounded to the nearest expressible value. If there are two nearest expressible values, the one with an LSB of 0 is selected.

If the unrounded value is $2^{Emax} (2 - 2^{-P})$ or more, the result will be infinity with the same sign as the unrounded value. The values of Emax and P, respectively, are 127 and 24 for single-precision, and 1023 and 53 for double-precision.

Round to Zero: The digits below the round bit of the unrounded value are discarded.

If the unrounded value is larger than the maximum expressible absolute value, the value will become the maximum expressible absolute value.

3.5 Floating-Point Exceptions

3.5.1 General FPU Disable Exceptions and Slot FPU Disable Exceptions

FPU-related exceptions include general FPU disable exceptions and slot FPU disable exceptions. These exceptions occur if an FPU instruction is executed when the FD bit of SR is set to 1.

3.5.2 FPU Exception Sources

The exception sources are as follows:

- FPU error (E): When FPSCR.DN = 0 and a denormalized number is input
- Invalid operation (V): In case of an invalid operation, such as NaN input
- Division by zero (Z): Division with a zero divisor
- Overflow (O): When the operation result overflows
- Underflow (U): When the operation result underflows
- Inexact exception (I): When overflow, underflow, or rounding occurs

to sources V, Z, O, U, and I, but not E. Thus, FPU errors cannot be disabled.

When an FPU exception occurs, the corresponding bit in the FPU exception cause field is set to 1, and 1 is added to the corresponding bit in the FPU exception flag field. When an FPU exception does not occur, the corresponding bit in the FPU exception cause field is cleared to 0, but the corresponding bit in the FPU exception flag field remains unchanged.

3.5.3 FPU Exception Handling

FPU exception handling is initiated in the following cases:

- FPU error (E): FPSCR.DN = 0 and a denormalized number is input
- Invalid operation (V): FPSCR.Enable.V = 1 and (instruction = FTRV or invalid operation)
- Division by zero (Z): FPSCR.Enable.Z = 1 and division with a zero divisor
- Overflow (O): FPSCR.Enable.O = 1 and instruction with possibility of operation result overflow
- Underflow (U): FPSCR.Enable.U = 1 and instruction with possibility of operation result underflow
- Inexact exception (I): FPSCR.Enable.I = 1 and instruction with possibility of inexact operation result

These possibilities are shown in the individual instruction descriptions. All exception events that originate in the FPU are assigned as the same exception event. The meaning of an exception is determined by software by reading from FPSCR and interpreting the information it contains. If no bits are set in the FPU exception cause field of FPSCR when one or more of bits O, U, I, and V (in case of FTRV only) are set in the FPU exception enable field, this indicates that an actual exception source is not generated. Also, the destination register is not changed by any FPU exception handling operation.

Except for the above, the FPU disables exception handling. In every processing, the bit corresponding to source V, Z, O, U, or I is set to 1, and a default value is generated as the operation result.

- Invalid operation (V): qNaN is generated as the result.
- Division by zero (Z): Infinity with the same sign as the unrounded value is generated.
- Overflow (O):

  When rounding mode = RZ, the maximum normalized number, with the same sign as the unrounded value, is generated.
  When rounding mode = RN, infinity with the same sign as the unrounded value is generated.

When FPSCR.DN = 1, zero with the same sign as the unrounded value, is generated.

- Inexact exception (I): An inexact result is generated.

3.6 Graphics Support Functions

The FPU supports two kinds of graphics functions: new instructions for geometric operations, and pair single-precision transfer instructions that enable high-speed data transfer.

3.6.1 Geometric Operation Instructions

Geometric operation instructions perform approximate-value computations. To enable high-speed computation with a minimum of hardware, the FPU ignores comparatively small values in the partial computation results of four multiplications. Consequently, the error shown below is produced in the result of the computation:

$$\text{Maximum error} = \text{MAX (individual multiplication result} \times 2^{-\text{MIN (number of multiplier significant digits}-1,\ \text{number of multiplicand significant digits}-1)}) + \text{MAX (result value} \times 2^{-23},\ 2^{-149})$$

The number of significant digits is 24 for a normalized number and 23 for a denormalized number (number of leading zeros in the fractional part).

In a future version of the SuperH RISC engine family, the above error is guaranteed, but the same result is not guaranteed.

FIPR FVm, FVn (m, n: 0, 4, 8, 12): This instruction is basically used for the following purposes:

- Inner product ($m \neq n$):
  This operation is generally used for surface/rear surface determination for polygon surfaces.
- Sum of square of elements ($m = n$):
  This operation is generally used to find the length of a vector.

Since approximate-value computations are performed to enable high-speed computation, the inexact exception (I) bit in both the FPU exception cause field and flag field are always set to 1 when an FIPR instruction is executed. Therefore, if the I bit is set in the FPU exception enable field, FPU exception handling will be executed.

FTRV XMTRX, FVn (n: 0, 4, 8, 12): This instruction is basically used for the following purposes:

- Matrix $(4 \times 4) \cdot$ vector (4):

parallel movement basically requires a 4 × 4 matrix, the FPU supports 4-dimensional operations.

- Matrix (4 × 4) × matrix (4 × 4):

  This operation requires the execution of four FTRV instructions.

Since approximate-value computations are performed to enable high-speed computation, the inexact exception (I) bit in both the FPU exception cause field and flag field are always set to 1 when an FTRV instruction is executed. Therefore, if the I bit is set in the FPU exception enable field, FPU exception handling will be executed. It is not possible to check all data types in the registers beforehand when executing an FTRV instruction. If the V bit is set in the FPU exception enable field, FPU exception handling will be executed.

FRCHG: This instruction modifies banked registers. For example, when the FTRV instruction is executed, matrix elements must be set in an array in the background bank. However, to create the actual elements of a translation matrix, it is easier to use registers in the foreground bank. When the LDS instruction is used on FPSCR, this instruction takes four to five cycles in order to maintain the FPU state. With the FRCHG instruction, the FR bit in FPSCR can be changed in one cycle.

3.6.2 Pair Single-Precision Data Transfer

In addition to the powerful new geometric operation instructions, the FPU also supports high-speed data transfer instructions.

When the SZ bit is 1, the FPU can perform data transfer by means of pair single-precision data transfer instructions.

- FMOV DRm/XDm, DRn/XDRn (m, n: 0, 2, 4, 6, 8, 10, 12, 14)
- FMOV DRm/XDm, @Rn (m: 0, 2, 4, 6, 8, 10, 12, 14; n: 0 to 15)

These instructions enable two single-precision (2 × 32-bit) data items to be transferred; that is, the transfer performance of these instructions is doubled.

- FSCHG

This instruction changes the value of the SZ bit in FPSCR, enabling fast switching between use and non-use of pair single-precision data transfer.

Programming Note: When the SZ bit is 1 and big-endian mode is used, FMOV can be used for a double-precision floating-point load or store. In little-endian mode, a double-precision floating-

cleared to 0.

3.7 Usage Notes

3.7.1 Rounding Mode and Underflow Flag

When using the Round to Nearest rounding mode, the underflow flag may not be set in cases defined as underflow by the IEEE754 standard.

Under the IEEE754 standard, when the Round to Nearest rounding mode is used and infinite-precision operation result *x* is (i) or (ii) (single-precision) or (iii) or (iv) (double-precision), there are cases where “the result after rounding is a normalized number, but an underflow results.”

In such cases where “the result after rounding is a normalized number, but an underflow results,” the FPU does not set the underflow flag to 1. In these cases the operation result, the value written to FRn, is correct. Also, if an FPU exception occurs, the underflow flag is not set to 1 but the inexact flag is set to 1 in such cases. Generation of FPU exceptions can be enabled by setting the enable field to 1.

(i) H'007FFFFF < x < H'00800000

(ii) H'807FFFFF > x > H'80800000

(iii) H'000FFFFF FFFFFFFF < x < H'00100000 00000000

(iv) H'800FFFFF FFFFFFFF > x > H'80100000 00000000

Examples

- Single-precision

  When FPSCR.RM = 00 (Round to Nearest) and FPSCR.PR = 0 (single-precision), and the FMUL instruction (H'00FFF000 * H'3F000800) is executed.

  a. According to IEEE754 standard

     Operation result: H'00800000

     FPSCR: H'0004300C

  b. FPU

     Operation result: H'00800000

     FPSCR: H'00041004

- Double-precision

  When FPSCR.RM = 00 (Round to Nearest) and FPSCR.PR = 1 (double-precision), and the FDIV instruction (H'001FFFFF FFFFFFFF / H'40000000 00000000) is executed.

  a. According to IEEE754 standard

FPSCR: H'000C300C

b. FPU

Operation result: H'00100000 00000000

FPSCR: H'000C1004

Workarounds

1. Use FPSCR.RM = 01, that is to say Round to Zero rather than Round to Nearest mode.
2. Use FPSCR.RM = 00, that is to say Round to Nearest mode, and set the enable field to 1 to enable generation of inexact exceptions so that the exception handling routine can be used to check whether or not an underflow has occurred.

3.7.2 Setting of Overflow Flag by FIPR or FTRV Instruction

When the maximum error produced by the FIPR or FTRV instruction exceeds the maximum value expressible as a normalized number (H'7F7FFFFF), the overflow flag may be set, even through the operation result is a positive or negative zero (H'00000000 or H'80000000).

Example: The operation result (FR7) after executing the instruction FIPR FV4, FV0 is H'00000000 (positive zero), but the overflow flag may be set nevertheless.

FPSCR = H'00040001
FR0 = H'FF7EF631 , FR1 = H80000000 , FR2 = H'8087F451 , FR3 = H'7F7EF631
FR4 = H'7F7EF631 , FR5 = H'0087F451 , FR6 = H'7F7EF631 , FR7 = H'7F7EF631

Workaround: Avoid using the FIPR and FTRV instructions, and use the FADD, FMUL, and FMAC instructions instead.

When two or more data items used in an operation by the FIPR or FTRV instruction are infinity, and all of the infinity items in the multiplication results have the same sign, the sign of the operation result may be incorrect.

Workarounds

1. Do not use infinity. If conditions (a) to (c) below are satisfied, infinity is never used in operations.
   a. Use Round to Zero (FPSCR.RM = 01) as the rounding mode.
   b. Do not divide by zero.
   c. Do not transfer a value of positive or negative infinity to FR0 to FR15 or to XF0 to XF15.
2. Avoid using the FIPR and FTRV instructions, and use the FADD, FMUL, and FMAC instructions instead.

3.7.4 Notes on Double-Precision FADD and FSUB Instructions

Description: If the input data for a double-precision FADD instruction or a double-precision FSUB instruction satisfies all of the conditions listed below, the inexact bits (FPSCR.Flag.I and FPSCR.Cause.I) may not be set even through the operation result is inexact.

Condition 1: The operation instruction is a double-precision FADD instruction or a double-precision FSUB instruction.

Condition 2: The difference between the DRn and DRm exponents is between 43 and 50.

Condition 3: At least one of bits 31 to 24 of the mantissa portion of whichever of DRn and DRm has the smaller absolute value is 1.

Condition 4: Bits 23 to 0 of the mantissa portion of whichever of DRn and DRm has the smaller absolute value are all 0.

Condition 5: Bits 40 to 32 of the mantissa portion of whichever of DRn and DRm has the smaller absolute value are all 0.

In addition, the result of an operation meeting the above conditions may have a rounding error. Specifically, in a case where the closest expressible value less than the unrounded value should be selected, the closest expressible value greater than the unrounded value is selected instead. Conversely, in a case where the closest expressible value greater than the unrounded value should be selected, the closest expressible value less than the unrounded value is selected instead.

data DR0 = H'C1F00000 00000000, DR2 = H'C4B250D2 0CC1F974, and FPSCR = H'000C0001, the correct operation result is DR2 = H'C4B250D2 0CC1F973, and FPSCR.Flag.I and FPSCR.Cause.I should be set to 1. However, the result actually produced by the FPU is DR2 = H'C4B250D2 0CC1F974, and FPSCR.Flag.I and FPSCR.Cause.I are not set to 1.

Effects: In addition to the problem described above, the numerical size of the result of the operation may contain a minute operation error equivalent to 1/256 of the LSB of the mantissa of the unrounded value. This is can be described as within the scope of the subsequent rounding mechanism. Strictly speaking, it consists of the following.

a: The infinite-precision operation result
b: The closest expressible value less than *a*
c: The closest expressible value greater than *a*
d: The operation result when *a* is rounded correctly
e: The operation result when *a* is rounded by the FPU

- The rounding error when rounding is performed correctly in Round to Nearest mode is:
  $0 \leq | d - a | \leq (1/2) \times (c - b)$

  And the rounding error when rounding is performed by the FPU is:
  $0 \leq | e - a | < (129/256) \times (c - b)$

  If $c - b$ is considered the LSB of the mantissa, the range of rounding error is equivalent to 1/256 of the LSB of the mantissa of the correctly rounded value.
- The rounding error when rounding is performed correctly in Round to Zero mode is:
  $(-1) \times (c - b) < | d | - | a | \leq 0$

  And the rounding error when rounding is performed by the FPU is:
  $(-1) \times (c - b) < | e | - | a | < (1/256) \times (c - b)$

  If $c - b$ is considered the LSB of the mantissa, the range of rounding error is equivalent to 1/256 of the LSB of the mantissa of the correctly rounded value.

4.1 Execution Environment

PC: At the start of instruction execution, the PC indicates the address of the instruction itself.

- Data sizes and data types

  The SH-4 instruction set is implemented with 16-bit fixed-length instructions. The SH-4 can use byte (8-bit), word (16-bit), longword (32-bit), and quadword (64-bit) data sizes for memory access. Single-precision floating-point data (32 bits) can be moved to and from memory using longword or quadword size. Double-precision floating-point data (64 bits) can be moved to and from memory using longword size. When a double-precision floating-point operation is specified (PR in FPSCR = 1), the result of an operation using quadword access will be undefined. When the SH-4 moves byte-size or word-size data from memory to a register, the data is sign-extended.

Load-Store Architecture: The SH-4 has a load-store architecture in which operations are basically executed using registers. Except for bit-manipulation operations such as logical AND that are executed directly in memory, operands in an operation that requires memory access are loaded into registers and the operation is executed between the registers.

Delayed Branches: Except for the two branch instructions BF and BT, the SH-4 branch instructions and RTE are delayed branches. In a delayed branch, the instruction following the branch is executed before the branch destination instruction. This execution slot following a delayed branch is called a delay slot. For example, the BRA execution sequence is as follows:

Static Sequence	**Dynamic Sequence**	
BRA TARGET	BRA TARGET	
ADD R1, R0 next_2	ADD R1, R0 target_instr	ADD in delay slot is executed before branching to TARGET

Delay Slot: A slot illegal instruction exception may occur when a specific instruction is executed in a delay slot. For details, see section 8, Exceptions. The instruction following BF/S or BT/S for which the branch is not taken is also a delay slot instruction.

T Bit: The T bit in SR is used to show the result of a compare operation, and is referenced by a conditional branch instruction. An example of the use of a conditional branch instruction is shown below.

```
CMP/EQ  R1, R0      ; If R0 = R1, T bit is set to 1
BT      TARGET      ; Branches to TARGET if T bit = 1 (R0 = R1)
```

In an RTE delay slot, the SR bits are referenced as follows. In instruction access, the MD bit is used before modification, and in data access, the MD bit is accessed after modification. The other bits—S, T, M, Q, FD, BL, and RB—after modification are used for delay slot instruction execution. The STC and STC.L SR instructions access all SR bits after modification.

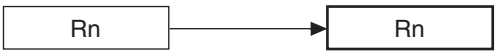
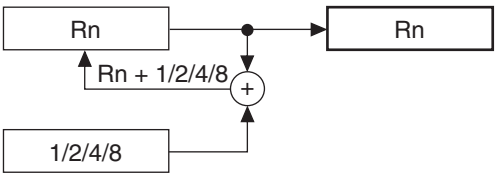
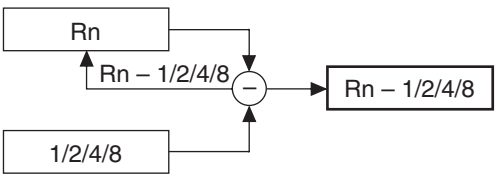
Constant Values: An 8-bit constant value can be specified by the instruction code and an immediate value. 16-bit and 32-bit constant values can be defined as literal constant values in memory, and can be referenced by a PC-relative load instruction.

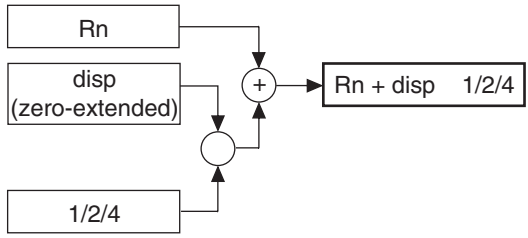
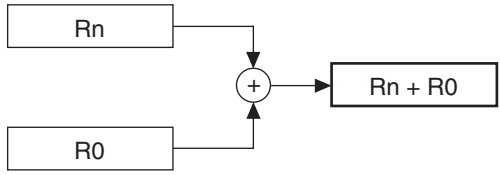
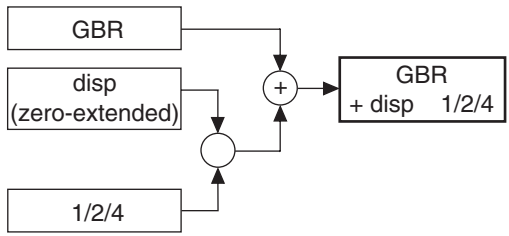
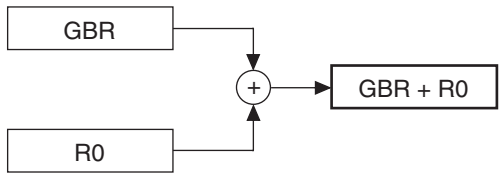
```
MOV.W   @(disp, PC), Rn
MOV.L   @(disp, PC), Rn
```

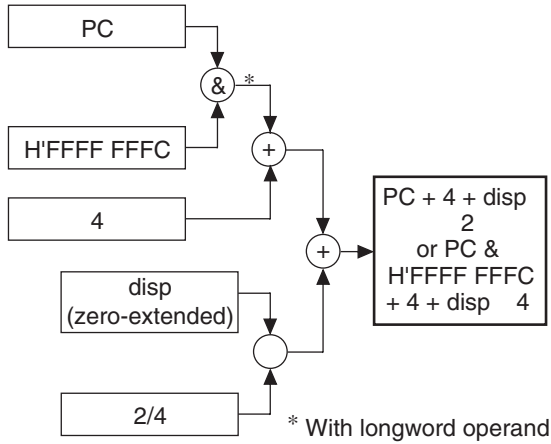
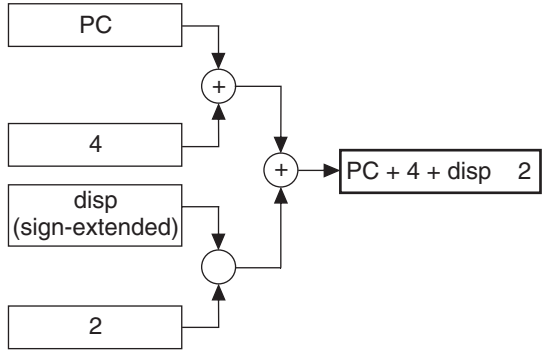
There are no PC-relative load instructions for floating-point operations. However, it is possible to set 0.0 or 1.0 by using the FLDI0 or FLDI1 instruction on a single-precision floating-point register.

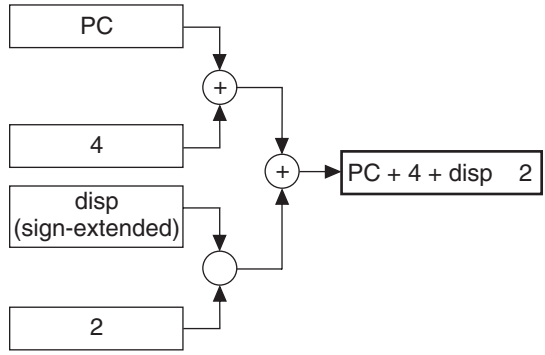
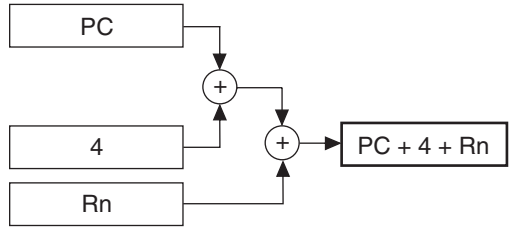
Addressing modes and effective address calculation methods are shown in table 4.1. When a location in virtual memory space is accessed (AT in MMUCR = 1), the effective address is translated into a physical memory address. If multiple virtual memory space systems are selected (SV in MMUCR = 0), the least significant bit of PTEH is also referenced as the access ASID. For details, see section 6, Memory Management Unit (MMU).

Table 4.1 Addressing Modes and Effective Addresses

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 → EA (EA: effective address)
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, 8 for a quadword operand. 	Rn → EA After instruction execution Byte: Rn + 1 → Rn Word: Rn + 2 → Rn Longword: Rn + 4 → Rn Quadword: Rn + 8 → 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, 8 for a quadword operand. 	Byte: Rn – 1 → Rn Word: Rn – 2 → Rn Longword: Rn – 4 → Rn Quadword: Rn – 8 → Rn Rn → EA (Instruction executed with Rn after calculation)

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 → EA Word: Rn + disp × 2 → EA Longword: Rn + disp × 4 → EA
Indexed register indirect	@(R0, Rn)	Effective address is sum of register Rn and R0 contents. 	Rn + R0 → EA
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 → EA Word: GBR + disp × 2 → EA Longword: GBR + disp × 4 → EA
Indexed GBR indirect	@(R0, GBR)	Effective address is sum of register GBR and R0 contents. 	GBR + R0 → EA

PC-relative with displacement	@(disp:8, PC)	Effective address is PC + 4 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 + 4 + disp × 2 → EA Longword: PC & H'FFFF FFFC + 4 + disp × 4 → EA
PC-relative	disp:8	Effective address is PC + 4 with 8-bit displacement disp added after being sign-extended and multiplied by 2. 	PC + 4 + disp × 2 → Branch-Target

PC-relative	disp:12	Effective address is PC + 4 with 12-bit displacement disp added after being sign-extended and multiplied by 2. 	PC + 4 + disp × 2 → Branch-Target
	Rn	Effective address is sum of PC + 4 and Rn. 	PC + 4 + Rn → Branch-Target
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.	—

Note: For the addressing modes below that use a displacement (disp), the assembler descriptions in this manual show the value before scaling (×1, ×2, or ×4) is performed according to the operand size. This is done to clarify the operation of the LSI. Refer to the relevant assembler notation rules for the actual assembler descriptions.

@ (disp:4, Rn) ; Register indirect with displacement
@ (disp:8, GBR) ; GBR indirect with displacement
@ (disp:8, PC) ; PC-relative with displacement
disp:8, disp:12 ; PC-relative

Table 4.2 shows the notation used in the SH instruction lists shown in tables 4.3 to 4.12.

Table 4.2 Notation Used in Instruction List

Item	Format	Description	
Instruction mnemonic	OP.Sz SRC, DEST	OP: Sz: SRC: DEST:	Operation code Size Source operand Source and/or destination operand
Summary of operation		→, ← (xx) M/Q/T & | ^ ~ <<n, >>n	Transfer direction Memory operand SR flag bits Logical AND of individual bits Logical OR of individual bits Logical exclusive-OR of individual bits Logical NOT of individual bits n-bit shift
Instruction code	MSB ↔ LSB	mmmm: nnnn: 0000: 0001: : 1111: mmm: nnn: 000: 001: : 111: mm: nn: 00: 01: 10: 11: iiii: dddd:	Register number (Rm, FRm) Register number (Rn, FRn) R0, FR0 R1, FR1 R15, FR15 Register number (DRm, XDm, Rm_BANK) Register number (DRm, XDm, Rn_BANK) DR0, XD0, R0_BANK DR2, XD2, R1_BANK DR14, XD14, R7_BANK Register number (FVm) Register number (FVn) FV0 FV4 FV8 FV12 Immediate data Displacement
Privileged mode		"Privileged" means the instruction can only be executed in privileged mode.	
T bit	Value of T bit after instruction execution	—: No change	

Note: Scaling (×1, ×2, ×4, or ×8) is executed according to the size of the instruction operand.

Instruction		Operation	Instruction Code	Privileged	T Bit
MOV	#imm,Rn	imm → sign extension → Rn	1110nnnniiiiiiii	—	—
MOV.W	@(disp,PC),Rn	(disp × 2 + PC + 4) → sign extension → Rn	1001nnnndddddddd	—	—
MOV.L	@(disp,PC),Rn	(disp × 4 + PC & H'FFFF FFFC + 4) → Rn	1101nnnndddddddd	—	—
MOV	Rm,Rn	Rm → Rn	0110nnnnmmmm0011	—	—
MOV.B	Rm,@Rn	Rm → (Rn)	0010nnnnmmmm0000	—	—
MOV.W	Rm,@Rn	Rm → (Rn)	0010nnnnmmmm0001	—	—
MOV.L	Rm,@Rn	Rm → (Rn)	0010nnnnmmmm0010	—	—
MOV.B	@Rm,Rn	(Rm) → sign extension → Rn	0110nnnnmmmm0000	—	—
MOV.W	@Rm,Rn	(Rm) → sign extension → Rn	0110nnnnmmmm0001	—	—
MOV.L	@Rm,Rn	(Rm) → Rn	0110nnnnmmmm0010	—	—
MOV.B	Rm,@-Rn	Rn-1 → Rn, Rm → (Rn)	0010nnnnmmmm0100	—	—
MOV.W	Rm,@-Rn	Rn-2 → Rn, Rm → (Rn)	0010nnnnmmmm0101	—	—
MOV.L	Rm,@-Rn	Rn-4 → Rn, Rm → (Rn)	0010nnnnmmmm0110	—	—
MOV.B	@Rm+,Rn	(Rm)→ sign extension → Rn, Rm + 1 → Rm	0110nnnnmmmm0100	—	—
MOV.W	@Rm+,Rn	(Rm) → sign extension → Rn, Rm + 2 → Rm	0110nnnnmmmm0101	—	—
MOV.L	@Rm+,Rn	(Rm) → Rn, Rm + 4 → Rm	0110nnnnmmmm0110	—	—
MOV.B	R0,@(disp,Rn)	R0 → (disp + Rn)	10000000nnnndddd	—	—
MOV.W	R0,@(disp,Rn)	R0 → (disp × 2 + Rn)	10000001nnnndddd	—	—
MOV.L	Rm,@(disp,Rn)	Rm → (disp × 4 + Rn)	0001nnnnmmmmdddd	—	—
MOV.B	@(disp,Rm),R0	(disp + Rm) → sign extension → R0	10000100mmmmdddd	—	—
MOV.W	@(disp,Rm),R0	(disp × 2 + Rm) → sign extension → R0	10000101mmmmdddd	—	—
MOV.L	@(disp,Rm),Rn	(disp × 4 + Rm) → Rn	0101nnnnmmmmdddd	—	—
MOV.B	Rm,@(R0,Rn)	Rm → (R0 + Rn)	0000nnnnmmmm0100	—	—
MOV.W	Rm,@(R0,Rn)	Rm → (R0 + Rn)	0000nnnnmmmm0101	—	—
MOV.L	Rm,@(R0,Rn)	Rm → (R0 + Rn)	0000nnnnmmmm0110	—	—
MOV.B	@(R0,Rm),Rn	(R0 + Rm) → sign extension → Rn	0000nnnnmmmm1100	—	—
MOV.W	@(R0,Rm),Rn	(R0 + Rm) → sign extension → Rn	0000nnnnmmmm1101	—	—
MOV.L	@(R0,Rm),Rn	(R0 + Rm) → Rn	0000nnnnmmmm1110	—	—

MOV.W	R0,@(disp,GBR)	R0 → (disp × 2 + GBR)	11000001dddddddd	—	—
MOV.L	R0,@(disp,GBR)	R0 → (disp × 4 + GBR)	11000010dddddddd	—	—
MOV.B	@(disp,GBR),R0	(disp + GBR) → sign extension → R0	11000100dddddddd	—	—
MOV.W	@(disp,GBR),R0	(disp × 2 + GBR) → sign extension → R0	11000101dddddddd	—	—
MOV.L	@(disp,GBR),R0	(disp × 4 + GBR) → R0	11000110dddddddd	—	—
MOVA	@(disp,PC),R0	disp × 4 + PC & H'FFFF FFFC + 4 → R0	11000111dddddddd	—	—
MOVT	Rn	T → Rn	0000nnnn00101001	—	—
SWAP.B	Rm,Rn	Rm → swap lower 2 bytes → Rn	0110nnnnmmmm1000	—	—
SWAP.W	Rm,Rn	Rm → swap upper/lower words → Rn	0110nnnnmmmm1001	—	—
XTRCT	Rm,Rn	Rm:Rn middle 32 bits → Rn	0010nnnnmmmm1101	—	—

Table 4.4 Arithmetic Operation Instructions

Instruction		Operation	Instruction Code	Privileged	T Bit
ADD	Rm,Rn	Rn + Rm → Rn	0011nnnnmmmm1100	—	—
ADD	#imm,Rn	Rn + imm → Rn	0111nnnniiiiiiii	—	—
ADDC	Rm,Rn	Rn + Rm + T → Rn, carry → T	0011nnnnmmmm1110	—	Carry
ADDV	Rm,Rn	Rn + Rm → Rn, overflow → T	0011nnnnmmmm1111	—	Overflow
CMP/EQ	#imm,R0	When R0 = imm, 1 → T Otherwise, 0 → T	10001000iiiiiiii	—	Comparison result
CMP/EQ	Rm,Rn	When Rn = Rm, 1 → T Otherwise, 0 → T	0011nnnnmmmm0000	—	Comparison result
CMP/HS	Rm,Rn	When Rn ≥ Rm (unsigned), 1 → T Otherwise, 0 → T	0011nnnnmmmm0010	—	Comparison result
CMP/GE	Rm,Rn	When Rn ≥ Rm (signed), 1 → T Otherwise, 0 → T	0011nnnnmmmm0011	—	Comparison result
CMP/HI	Rm,Rn	When Rn > Rm (unsigned), 1 → T Otherwise, 0 → T	0011nnnnmmmm0110	—	Comparison result
CMP/GT	Rm,Rn	When Rn > Rm (signed), 1 → T Otherwise, 0 → T	0011nnnnmmmm0111	—	Comparison result
CMP/PZ	Rn	When Rn ≥ 0, 1 → T Otherwise, 0 → T	0100nnnn00010001	—	Comparison result

CMP/PL	Rn	When Rn > 0, 1 → T Otherwise, 0 → T	0100nnnn00010101	—	Comparison result
CMP/STR	Rm,Rn	When any bytes are equal, 1 → T Otherwise, 0 → T	0010nnnnmmmm1100	—	Comparison result
DIV1	Rm,Rn	1-step division (Rn ÷ Rm)	0011nnnnmmmm0100	—	Calculation result
DIV0S	Rm,Rn	MSB of Rn → Q, MSB of Rm → M, M^Q → T	0010nnnnmmmm0111	—	Calculation result
DIV0U		0 → M/Q/T	0000000000011001	—	0
DMULS.L	Rm,Rn	Signed, Rn × Rm → MAC, 32 × 32 → 64 bits	0011nnnnmmmm1101	—	—
DMULU.L	Rm,Rn	Unsigned, Rn × Rm → MAC, 32 × 32 → 64 bits	0011nnnnmmmm0101	—	—
DT	Rn	Rn – 1 → Rn; when Rn = 0, 1 → T When Rn ≠ 0, 0 → T	0100nnnn00010000	—	Comparison result
EXTS.B	Rm,Rn	Rm sign-extended from byte → Rn	0110nnnnmmmm1110	—	—
EXTS.W	Rm,Rn	Rm sign-extended from word → Rn	0110nnnnmmmm1111	—	—
EXTU.B	Rm,Rn	Rm zero-extended from byte → Rn	0110nnnnmmmm1100	—	—
EXTU.W	Rm,Rn	Rm zero-extended from word → Rn	0110nnnnmmmm1101	—	—
MAC.L	@Rm+,@Rn+	Signed, (Rn) × (Rm) + MAC → MAC Rn + 4 → Rn, Rm + 4 → Rm 32 × 32 + 64 → 64 bits	0000nnnnmmmm1111	—	—
MAC.W	@Rm+,@Rn+	Signed, (Rn) × (Rm) + MAC → MAC Rn + 2 → Rn, Rm + 2 → Rm 16 × 16 + 64 → 64 bits	0100nnnnmmmm1111	—	—
MUL.L	Rm,Rn	Rn × Rm → MACL 32 × 32 → 32 bits	0000nnnnmmmm0111	—	—
MULS.W	Rm,Rn	Signed, Rn × Rm → MACL 16 × 16 → 32 bits	0010nnnnmmmm1111	—	—
MULU.W	Rm,Rn	Unsigned, Rn × Rm → MACL 16 × 16 → 32 bits	0010nnnnmmmm1110	—	—
NEG	Rm,Rn	0 – Rm → Rn	0110nnnnmmmm1011	—	—
NEGC	Rm,Rn	0 – Rm – T → Rn, borrow → T	0110nnnnmmmm1010	—	Borrow
SUB	Rm,Rn	Rn – Rm → Rn	0011nnnnmmmm1000	—	—

SUBC	Rm,Rn	Rn – Rm – T → Rn, borrow → T	0011nnnnmmmm1010	—	Borrow
SUBV	Rm,Rn	Rn – Rm → Rn, underflow → T	0011nnnnmmmm1011	—	Underflow

Table 4.5 Logic Operation Instructions

Instruction		Operation	Instruction Code	Privileged	T Bit
AND	Rm,Rn	Rn & Rm → Rn	0010nnnnmmmm1001	—	—
AND	#imm,R0	R0 & imm → R0	11001001iiiiiiii	—	—
AND.B	#imm,@(R0,GBR)	(R0 + GBR) & imm → (R0 + GBR)	11001101iiiiiiii	—	—
NOT	Rm,Rn	~Rm → Rn	0110nnnnmmmm0111	—	—
OR	Rm,Rn	Rn | Rm → Rn	0010nnnnmmmm1011	—	—
OR	#imm,R0	R0 | imm → R0	11001011iiiiiiii	—	—
OR.B	#imm,@(R0,GBR)	(R0 + GBR) | imm → (R0 + GBR)	11001111iiiiiiii	—	
TAS.B	@Rn	When (Rn) = 0, 1 → T Otherwise, 0 → T In both cases, 1 → MSB of (Rn)	0100nnnn00011011	—	Test result
TST	Rm,Rn	Rn & Rm; when result = 0, 1 → T Otherwise, 0 → T	0010nnnnmmmm1000	—	Test result
TST	#imm,R0	R0 & imm; when result = 0, 1 → T Otherwise, 0 → T	11001000iiiiiiii	—	Test result
TST.B	#imm,@(R0,GBR)	(R0 + GBR) & imm; when result = 0, 1 → T Otherwise, 0 → T	11001100iiiiiiii	—	Test result
XOR	Rm,Rn	Rn ∧ Rm → Rn	0010nnnnmmmm1010	—	—
XOR	#imm,R0	R0 ∧ imm → R0	11001010iiiiiiii	—	—
XOR.B	#imm,@(R0,GBR)	(R0 + GBR) ∧ imm → (R0 + GBR)	11001110iiiiiiii	—	—

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

Instruction		Operation	Instruction Code	Privileged	T Bit
BF	label	When T = 0, disp × 2 + PC + 4 → PC When T = 1, nop	10001011dddddddd	—	—
BF/S	label	Delayed branch; when T = 0, disp × 2 + PC + 4 → PC When T = 1, nop	10001111dddddddd	—	—
BT	label	When T = 1, disp × 2 + PC + 4 → PC When T = 0, nop	10001001dddddddd	—	—
BT/S	label	Delayed branch; when T = 1, disp × 2 + PC + 4 → PC When T = 0, nop	10001101dddddddd	—	—
BRA	label	Delayed branch, disp × 2 + PC + 4 → PC	1010dddddddddddd	—	—
BRAF	Rn	Delayed branch, Rn + PC + 4 → PC	0000nnnn00100011	—	—
BSR	label	Delayed branch, PC + 4 → PR, disp × 2 + PC + 4 → PC	1011dddddddddddd	—	—
BSRF	Rn	Delayed branch, PC + 4 → PR, Rn + PC + 4 → PC	0000nnnn00000011	—	—
JMP	@Rn	Delayed branch, Rn → PC	0100nnnn00101011	—	—
JSR	@Rn	Delayed branch, PC + 4 → PR, Rn → PC	0100nnnn00001011	—	—
RTS		Delayed branch, PR → PC	0000000000001011	—	—

Instruction		Operation	Instruction Code	Privileged	T Bit
CLRMAC		0 → MACH, MACL	0000000000101000	—	—
CLRS		0 → S	0000000001001000	—	—
CLRT		0 → T	0000000000001000	—	0
LDC	Rm,SR	Rm → SR	0100mmmm00001110	Privileged	LSB
LDC	Rm,GBR	Rm → GBR	0100mmmm00011110	—	—
LDC	Rm,VBR	Rm → VBR	0100mmmm00101110	Privileged	—
LDC	Rm,SSR	Rm → SSR	0100mmmm00111110	Privileged	—
LDC	Rm,SPC	Rm → SPC	0100mmmm01001110	Privileged	—
LDC	Rm,DBR	Rm → DBR	0100mmmm11111010	Privileged	—
LDC	Rm,Rn_BANK	Rm → Rn_BANK (n = 0 to 7)	0100mmmm1nnn1110	Privileged	—
LDC.L	@Rm+,SR	(Rm) → SR, Rm + 4 → Rm	0100mmmm00000111	Privileged	LSB
LDC.L	@Rm+,GBR	(Rm) → GBR, Rm + 4 → Rm	0100mmmm00010111	—	—
LDC.L	@Rm+,VBR	(Rm) → VBR, Rm + 4 → Rm	0100mmmm00100111	Privileged	—
LDC.L	@Rm+,SSR	(Rm) → SSR, Rm + 4 → Rm	0100mmmm00110111	Privileged	—
LDC.L	@Rm+,SPC	(Rm) → SPC, Rm + 4 → Rm	0100mmmm01000111	Privileged	—
LDC.L	@Rm+,DBR	(Rm) → DBR, Rm + 4 → Rm	0100mmmm11110110	Privileged	—
LDC.L	@Rm+,Rn_BANK	(Rm) → Rn_BANK, Rm + 4 → Rm	0100mmmm1nnn0111	Privileged	—
LDS	Rm,MACH	Rm → MACH	0100mmmm00001010	—	—
LDS	Rm,MACL	Rm → MACL	0100mmmm00011010	—	—
LDS	Rm,PR	Rm → PR	0100mmmm00101010	—	—
LDS.L	@Rm+,MACH	(Rm) → MACH, Rm + 4 → Rm	0100mmmm00000110	—	—
LDS.L	@Rm+,MACL	(Rm) → MACL, Rm + 4 → Rm	0100mmmm00010110	—	—
LDS.L	@Rm+,PR	(Rm) → PR, Rm + 4 → Rm	0100mmmm00100110	—	—
LDTLB		PTEH/PTEL → TLB	0000000000111000	Privileged	—
MOVCA.L	R0,@Rn	R0 → (Rn) (without fetching cache block)	0000nnnn11000011	—	—
NOP		No operation	0000000000001001	—	—
OCBI	@Rn	Invalidates operand cache block	0000nnnn10010011	—	—
OCBP	@Rn	Writes back and invalidates operand cache block	0000nnnn10100011	—	—
OCBWB	@Rn	Writes back operand cache block	0000nnnn10110011	—	—
PREF	@Rn	(Rn) → operand cache	0000nnnn10000011	—	—
RTE		Delayed branch, SSR/SPC → SR/PC	0000000000101011	Privileged	—

SETT		1 → T	0000000000011000	—	1
SLEEP		Sleep or standby	0000000000011011	Privileged	—
STC	SR,Rn	SR → Rn	0000nnnn00000010	Privileged	—
STC	GBR,Rn	GBR → Rn	0000nnnn00010010	—	—
STC	VBR,Rn	VBR → Rn	0000nnnn00100010	Privileged	—
STC	SSR,Rn	SSR → Rn	0000nnnn00110010	Privileged	—
STC	SPC,Rn	SPC → Rn	0000nnnn01000010	Privileged	—
STC	SGR,Rn	SGR → Rn	0000nnnn00111010	Privileged	—
STC	DBR,Rn	DBR → Rn	0000nnnn11111010	Privileged	—
STC	Rm_BANK,Rn	Rm_BANK → Rn (m = 0 to 7)	0000nnnn1mmm0010	Privileged	—
STC.L	SR,@-Rn	Rn – 4 → Rn, SR → (Rn)	0100nnnn00000011	Privileged	—
STC.L	GBR,@-Rn	Rn – 4 → Rn, GBR → (Rn)	0100nnnn00010011	—	—
STC.L	VBR,@-Rn	Rn – 4 → Rn, VBR → (Rn)	0100nnnn00100011	Privileged	—
STC.L	SSR,@-Rn	Rn – 4 → Rn, SSR → (Rn)	0100nnnn00110011	Privileged	—
STC.L	SPC,@-Rn	Rn – 4 → Rn, SPC → (Rn)	0100nnnn01000011	Privileged	—
STC.L	SGR,@-Rn	Rn – 4 → Rn, SGR → (Rn)	0100nnnn00110010	Privileged	—
STC.L	DBR,@-Rn	Rn – 4 → Rn, DBR → (Rn)	0100nnnn11110010	Privileged	—
STC.L	Rm_BANK,@-Rn	Rn – 4 → Rn, Rm_BANK → (Rn) (m = 0 to 7)	0100nnnn1mmm0011	Privileged	—
STS	MACH,Rn	MACH → Rn	0000nnnn00001010	—	—
STS	MACL,Rn	MACL → Rn	0000nnnn00011010	—	—
STS	PR,Rn	PR → Rn	0000nnnn00101010	—	—
STS.L	MACH,@-Rn	Rn – 4 → Rn, MACH → (Rn)	0100nnnn00000010	—	—
STS.L	MACL,@-Rn	Rn – 4 → Rn, MACL → (Rn)	0100nnnn00010010	—	—
STS.L	PR,@-Rn	Rn – 4 → Rn, PR → (Rn)	0100nnnn00100010	—	—
TRAPA	#imm	PC + 2 → SPC, SR → SSR, #imm << 2 → TRA, H'160 → EXPEVT, VBR + H'0100 → PC	11000011iiiiiiii	—	—

Instruction		Operation	Instruction Code	Privileged	T Bit
FLDI0	FRn	H'0000 0000 → FRn	1111nnnn10001101	—	—
FLDI1	FRn	H'3F80 0000 → FRn	1111nnnn10011101	—	—
FMOV	FRm,FRn	FRm → FRn	1111nnnnmmmm1100	—	—
FMOV.S	@Rm,FRn	(Rm) → FRn	1111nnnnmmmm1000	—	—
FMOV.S	@(R0,Rm),FRn	(R0 + Rm) → FRn	1111nnnnmmmm0110	—	—
FMOV.S	@Rm+,FRn	(Rm) → FRn, Rm + 4 → Rm	1111nnnnmmmm1001	—	—
FMOV.S	FRm,@Rn	FRm → (Rn)	1111nnnnmmmm1010	—	—
FMOV.S	FRm,@-Rn	Rn-4 → Rn, FRm → (Rn)	1111nnnnmmmm1011	—	—
FMOV.S	FRm,@(R0,Rn)	FRm → (R0 + Rn)	1111nnnnmmmm0111	—	—
FMOV	DRm,DRn	DRm → DRn	1111nnn0mmm01100	—	—
FMOV	@Rm,DRn	(Rm) → DRn	1111nnn0mmmm1000	—	—
FMOV	@(R0,Rm),DRn	(R0 + Rm) → DRn	1111nnn0mmmm0110	—	—
FMOV	@Rm+,DRn	(Rm) → DRn, Rm + 8 → Rm	1111nnn0mmmm1001	—	—
FMOV	DRm,@Rn	DRm → (Rn)	1111nnnnmmm01010	—	—
FMOV	DRm,@-Rn	Rn-8 → Rn, DRm → (Rn)	1111nnnnmmm01011	—	—
FMOV	DRm,@(R0,Rn)	DRm → (R0 + Rn)	1111nnnnmmm00111	—	—
FLDS	FRm,FPUL	FRm → FPUL	1111mmmm00011101	—	—
FSTS	FPUL,FRn	FPUL → FRn	1111nnnn00001101	—	—
FABS	FRn	FRn & H'7FFF FFFF → FRn	1111nnnn01011101	—	—
FADD	FRm,FRn	FRn + FRm → FRn	1111nnnnmmmm0000	—	—
FCMP/EQ	FRm,FRn	When FRn = FRm, 1 → T Otherwise, 0 → T	1111nnnnmmmm0100	—	Comparison result
FCMP/GT	FRm,FRn	When FRn > FRm, 1 → T Otherwise, 0 → T	1111nnnnmmmm0101	—	Comparison result
FDIV	FRm,FRn	FRn/FRm → FRn	1111nnnnmmmm0011	—	—
FLOAT	FPUL,FRn	(float) FPUL → FRn	1111nnnn00101101	—	—
FMAC	FR0,FRm,FRn	FR0*FRm + FRn → FRn	1111nnnnmmmm1110	—	—
FMUL	FRm,FRn	FRn*FRm → FRn	1111nnnnmmmm0010	—	—
FNEG	FRn	FRn ∧ H'8000 0000 → FRn	1111nnnn01001101	—	—
FSQRT	FRn	√FRn → FRn	1111nnnn01101101	—	—
FSUB	FRm,FRn	FRn – FRm → FRn	1111nnnnmmmm0001	—	—
FTRC	FRm,FPUL	(long) FRm → FPUL	1111mmmm00111101	—	—

Instruction		Operation	Instruction Code	Privileged	T Bit
FABS	DRn	DRn & H'7FFF FFFF FFFF FFFF → DRn	1111nnn001011101	—	—
FADD	DRm,DRn	DRn + DRm → DRn	1111nnn0mmm00000	—	—
FCMP/EQ	DRm,DRn	When DRn = DRm, 1 → T Otherwise, 0 → T	1111nnn0mmm00100	—	Comparison result
FCMP/GT	DRm,DRn	When DRn > DRm, 1 → T Otherwise, 0 → T	1111nnn0mmm00101	—	Comparison result
FDIV	DRm,DRn	DRn /DRm → DRn	1111nnn0mmm00011	—	—
FCNVDS	DRm,FPUL	double_to_ float[DRm] → FPUL	1111mmm010111101	—	—
FCNVSD	FPUL,DRn	float_to_ double [FPUL] → DRn	1111nnn010101101	—	—
FLOAT	FPUL,DRn	(float)FPUL → DRn	1111nnn000101101	—	—
FMUL	DRm,DRn	DRn *DRm → DRn	1111nnn0mmm00010	—	—
FNEG	DRn	DRn ^ H'8000 0000 0000 0000 → DRn	1111nnn001001101	—	—
FSQRT	DRn	√DRn → DRn	1111nnn001101101	—	—
FSUB	DRm,DRn	DRn – DRm → DRn	1111nnn0mmm00001	—	—
FTRC	DRm,FPUL	(long) DRm → FPUL	1111mmm000111101	—	—

Table 4.11 Floating-Point Control Instructions

Instruction		Operation	Instruction Code	Privileged	T Bit
LDS	Rm,FPSCR	Rm → FPSCR	0100mmmm01101010	—	—
LDS	Rm,FPUL	Rm → FPUL	0100mmmm01011010	—	—
LDS.L	@Rm+,FPSCR	(Rm) → FPSCR, Rm+4 → Rm	0100mmmm01100110	—	—
LDS.L	@Rm+,FPUL	(Rm) → FPUL, Rm+4 → Rm	0100mmmm01010110	—	—
STS	FPSCR,Rn	FPSCR → Rn	0000nnnn01101010	—	—
STS	FPUL,Rn	FPUL → Rn	0000nnnn01011010	—	—
STS.L	FPSCR,@-Rn	Rn – 4 → Rn, FPSCR → (Rn)	0100nnnn01100010	—	—
STS.L	FPUL,@-Rn	Rn – 4 → Rn, FPUL → (Rn)	0100nnnn01010010	—	—

Instruction		Operation	Instruction Code	Privileged	T Bit
FMOV	DRm,XDn	DRm → XDn	1111nnn1mmm01100	—	—
FMOV	XDm,DRn	XDm → DRn	1111nnn0mmm11100	—	—
FMOV	XDm,XDn	XDm → XDn	1111nnn1mmm11100	—	—
FMOV	@Rm,XDn	(Rm) → XDn	1111nnn1mmmm1000	—	—
FMOV	@Rm+,XDn	(Rm) → XDn, Rm + 8 → Rm	1111nnn1mmmm1001	—	—
FMOV	@(R0,Rm),XDn	(R0 + Rm) → XDn	1111nnn1mmmm0110	—	—
FMOV	XDm,@Rn	XDm → (Rn)	1111nnnnmmm11010	—	—
FMOV	XDm,@-Rn	Rn – 8 → Rn, XDm → (Rn)	1111nnnnmmm11011	—	—
FMOV	XDm,@(R0,Rn)	XDm → (R0 + Rn)	1111nnnnmmm10111	—	—
FIPR	FVm,FVn	inner_product [FVm, FVn] → FR[n+3]	1111nnmm11101101	—	—
FTRV	XMTRX,FVn	transform_vector [XMTRX, FVn] → FVn	1111nn0111111101	—	—
FRCHG		~FPSCR.FR → FPSCR.FR	1111101111111101	—	—
FSCHG		~FPSCR.SZ → FPSCR.SZ	1111001111111101	—	—

4.4 Usage Note

4.4.1 Notes on TRAPA Instruction, SLEEP Instruction, and Undefined Instruction (H'FFFD)

- Incorrect data may be written to the cache when a TRAPA instruction or undefined instruction code H'FFFD is executed.
- The ITLB hit judgment may be incorrect when a TRAPA instruction or undefined instruction code H'FFFD is executed, causing a multi-hit exception to occur after re-registration.
- Incorrect data may be written to an FPU-related register or to the MACH or MACL register when a TRAPA instruction, SLEEP instruction, or undefined instruction code H'FFFD is executed.

Conditions under which Problem Occurs

1. Incorrect data may be written to the instruction cache when the following three conditions occur at the same time.
   a. The instruction cache is enabled (CCR.ICE = 1).
   b. A TRAPA instruction or undefined instruction code H'FFFD in a cache-enabled area is executed.

(read or write) an address (H'F0000000 to H'F7FFFFFF) mapped to the internal cache or internal TLB.

2. Incorrect data may be written to the operand cache when the following three conditions occur at the same time.
   a. The operand cache is enabled (CCR.OCE = 1).
   b. Undefined instruction code H'FFFD is executed.
   c. The four words of data following the undefined instruction code H'FFFD mentioned in b. contain code that can be interpreted as an OCBI, OCBP, OCBWB, or TAS.B instruction accessing an address (H'E0000000 to H'E3FFFFFF) mapped to the internal store queue.
3. The ITLB hit judgment may be incorrect when the following three conditions occur at the same time. If an ITLB hit is erroneously judged to be a miss, ITLB re-registration is performed. This can cause an ITLB multi-hit exception to occur.
   a. The MMU enabled (MMUCR.AT = 1).
   b. A TRAPA instruction or undefined instruction code H'FFFD in a TLB conversion area (area U0, P0, or P3) is executed.
   c. The four words of data following the TRAPA instruction or undefined instruction code H'FFFD mentioned in b. contain code that can be interpreted as an instruction to access (read or write) an address (H'F0000000 to H'F7FFFFFF) mapped to the internal cache or internal TLB.
4. Incorrect data may be written to an FPU-related register (FR0 to FR15, XF0 to XF15, FPSCR, or FPUL) or to the MACH or MACL register when the following two conditions occur at the same time.
   a. A TRAPA instruction, SLEEP instruction, or undefined instruction code H'FFFD is executed
   b. The eight words of data following the TRAPA instruction, SLEEP instruction, or undefined instruction code H'FFFD mentioned in a. contain H'Fxxx (an instruction with H'F as the first four bits), excluding H'FFFD, and the code can be interpreted, in combination with FPSCR.PR at that point, as an undefined instruction.

   Example: Instruction H'FxxE (x: any hexadecimal digit) is defined here as undefined when FPSCR.PR is set to 1.

Note: The number of instructions following the instructions mentioned above that may be affected by the problem is as follows: in the case of 1. to 3., the number of instructions that can be executed in 2xIck, and in the case of 4., the number of instructions that can be executed in 4xIck. The maximum number of instructions that can be executed in 2xIck or 4xIck is four or eight, respectively. Therefore, the affected codes are those occurring in “the four words (or eight words) of data following the instruction.”

a. Include a NOP instruction in the eight words of data following each TRAPA instruction, SLEEP instruction, or undefined instruction code H'FFFD.

b. Include an OR R0,R0 instruction in the five words of data following each TRAPA instruction, SLEEP instruction, or undefined instruction code H'FFFD. This workaround also applies to cases where “the eight words of data following the … instruction … contain H'Fxxx,” as mentioned in condition 4. b., because two OR instructions are never executed simultaneously, so a minimum of 5xIck is required for execution.

The SH-4 is a 2-ILP (instruction-level-parallelism) superscalar pipelining microprocessor. Instruction execution is pipelined, and two instructions can be executed in parallel. The execution cycles depend on the implementation of a processor. The definitions in this section may not be applied to the SH-4 products other than the SH7760.

5.1 Pipelines

Figure 5.1 shows the basic pipelines. Normally, a pipeline consists of five or six stages: instruction fetch (I), decode and register read (D), execution (EX/SX/F0/F1/F2/F3), data access (NA/MA), and write-back (S/FS). An instruction is executed as a combination of basic pipelines. Figure 5.2 shows the instruction execution patterns.

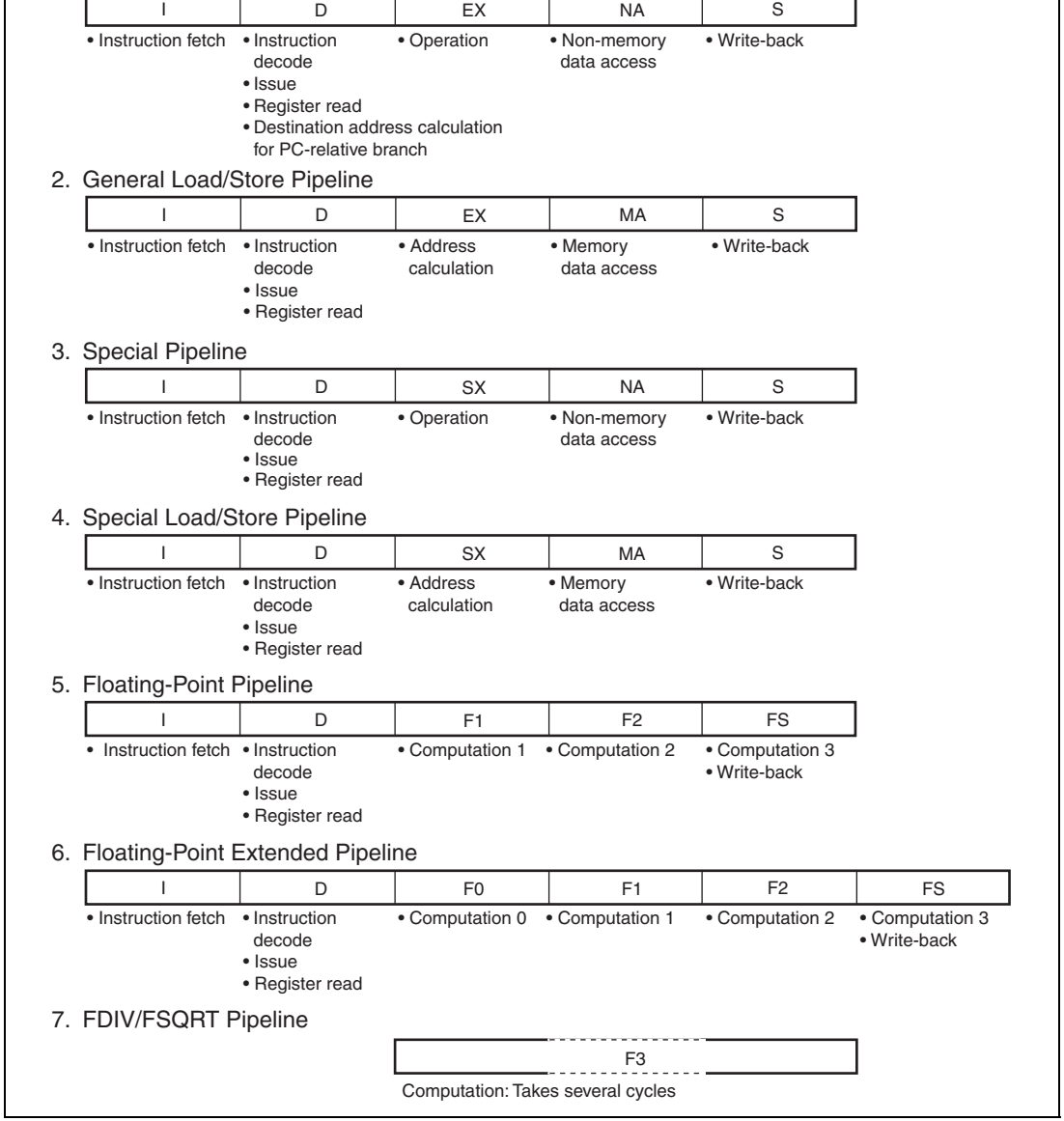
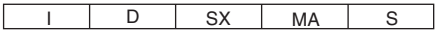


Figure 5.1 Basic Pipelines

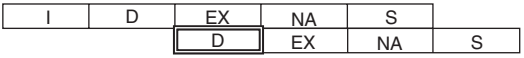
EXT[SU].[BW], MOV, MOV#, MOVA, MOVT, SWAP.[BW], XTRCT, ABS*, CMP*,
DIV*, DT, NEG*, SUB*, AND, AND#, NOT, OR, OR#, TST, TST#, XOR, XOR#,
ROT*, SHA*, SHL*, BF*, BT*, BRA, NOP, CLRS, CLRT, SETS, SETT,
LDS to FPUL, STS from FPUL/FPSCR, FLDI0, FLDI1, FMOV, FLDS, FSTS,
single-/double-precision FABS/FNEG

2. Load/store: 1 issue cycle
MOV.[BWL]. FMOV*@, LDS.L to FPUL, LDTLB, PREF, STS.L from FPUL/FPSCR

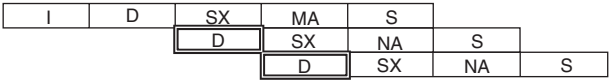
3. GBR-based load/store: 1 issue cycle
MOV.[BWL]@(d,GBR)



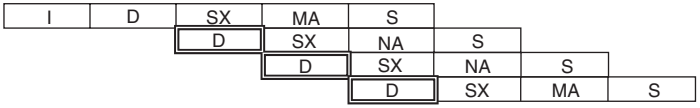
4. JMP, RTS, BRAF: 2 issue cycles



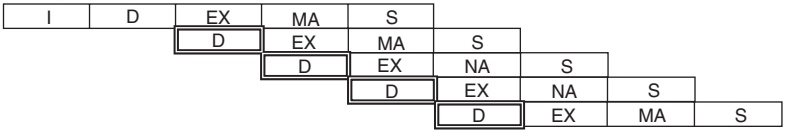
5. TST.B: 3 issue cycles



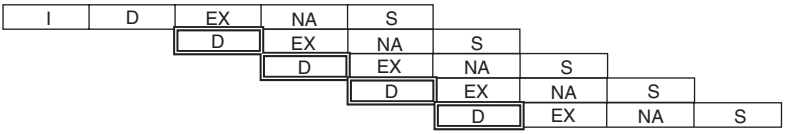
6. AND.B, OR.B, XOR.B: 4 issue cycles



7. TAS.B: 5 issue cycles



8. RTE: 5 issue cycles



9. SLEEP: 4 issue cycles

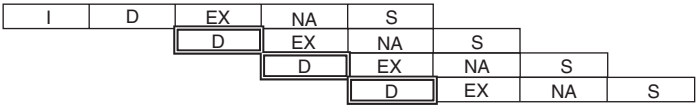


Figure 5.2 Instruction Execution Patterns

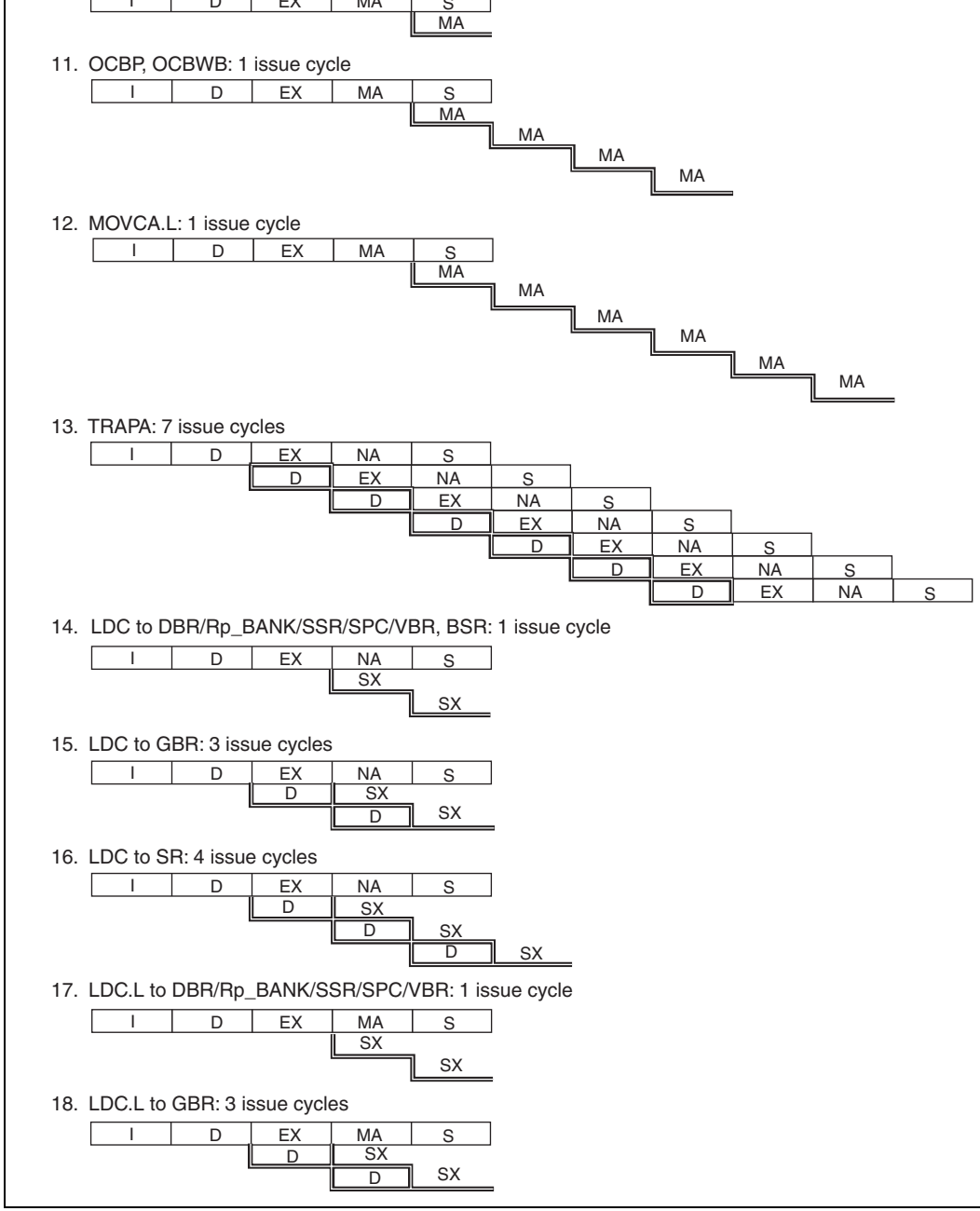
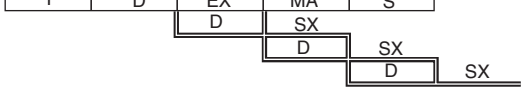
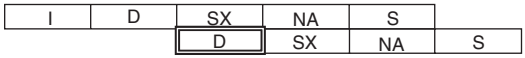


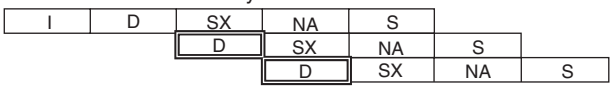
Figure 5.2 Instruction Execution Patterns (cont)



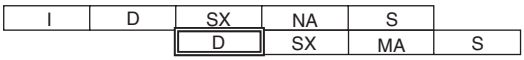
20. STC from DBR/GBR/Rp_BANK/SR/SSR/SPC/VBR: 2 issue cycles



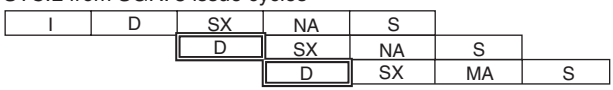
21. STC. from SGR: 3 issue cycles



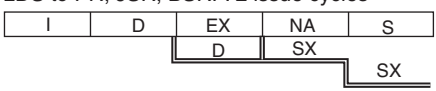
22. STC.L from DBR/GBR/Rp_BANK/SR/SSR/SPC/VBR: 2 issue cycles



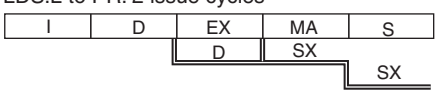
23. STC.L from SGR: 3 issue cycles



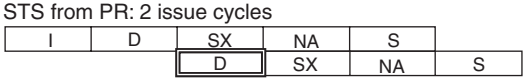
24. LDS to PR, JSR, BSRF: 2 issue cycles



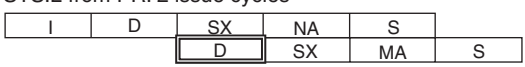
25. LDS.L to PR: 2 issue cycles



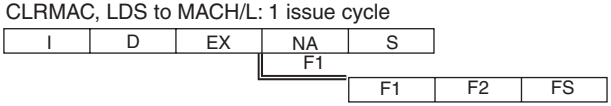
26. STS from PR: 2 issue cycles



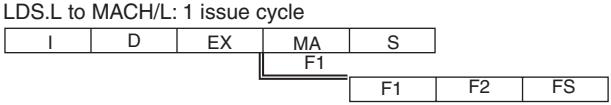
27. STS.L from PR: 2 issue cycles



28. CLRMAC, LDS to MACH/L: 1 issue cycle



29. LDS.L to MACH/L: 1 issue cycle



30. STS from MACH/L: 1 issue cycle

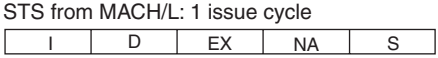
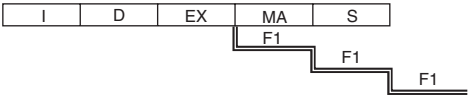


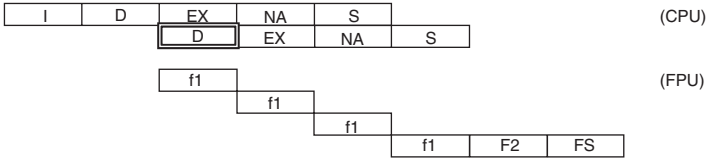
Figure 5.2 Instruction Execution Patterns (cont)

32. LDS to FPSCR: 1 issue cycle

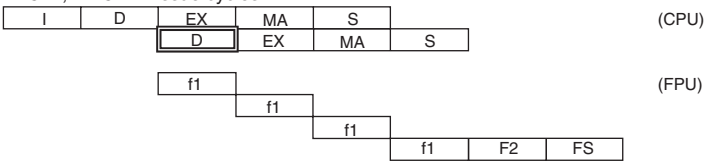
33. LDS.L to FPSCR: 1 issue cycle



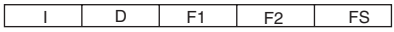
34. Fixed-point multiplication: 2 issue cycles
DMULS.L, DMULU.L, MUL.L, MULS.W, MULU.W



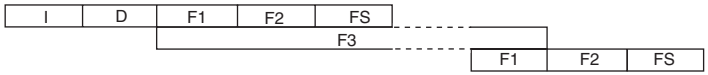
35. MAC.W, MAC.L: 2 issue cycles



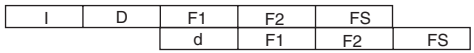
36. Single-precision floating-point computation: 1 issue cycle
FCMP/EQ, FCMP/GT, FADD, FLOAT, FMAC, FMUL, FSUB, FTRC, FRCHG, FSCHG



37. Single-precision FDIV/SQRT: 1 issue cycle



38. Double-precision floating-point computation 1: 1 issue cycle
FCNVDS, FCNVSD, FLOAT, FTRC



39. Double-precision floating-point computation 2: 1 issue cycle
FADD, FMUL, FSUB

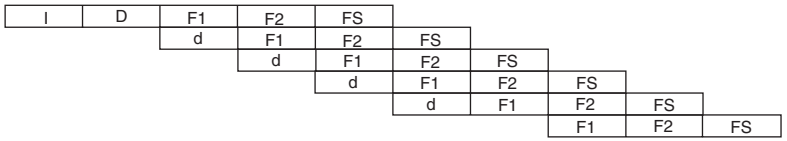
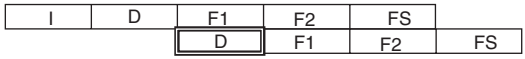
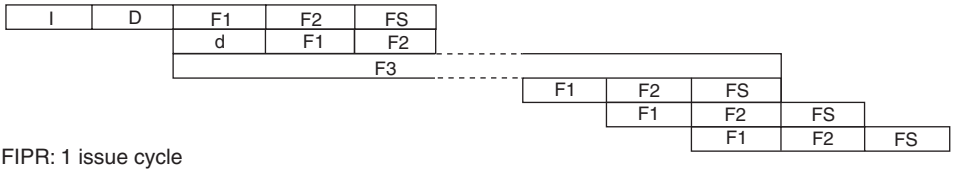


Figure 5.2 Instruction Execution Patterns (cont)

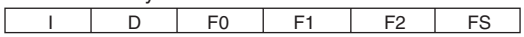
FCMP/EQ, FCMP/GT



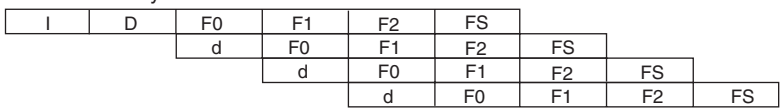
41. Double-precision FDIV/FSQRT: 1 issue cycle
    FDIV, FSQRT



42. FIPR: 1 issue cycle



43. FTRV: 1 issue cycle



Notes: ?? : Cannot overlap a stage of the same kind, except when two instructions are executed in parallel.

D : Locks D-stage.

d : Register read only

?? : Locks, but no operation is executed.

f1 : Can overlap another f1, but not another F1.

Figure 5.2 Instruction Execution Patterns (cont)

Instructions are categorized into six groups according to the internal function blocks used, as shown in table 5.1. Table 5.2 shows the parallel-executability of pairs of instructions in terms of groups. For example, ADD in the EX group and BRA in the BR group can be executed in parallel.

Table 5.1 Instruction Groups

1. MT Group

CLRT		CMP/HI	Rm,Rn	MOV	Rm,Rn
CMP/EQ	#imm,R0	CMP/HS	Rm,Rn	NOP	
CMP/EQ	Rm,Rn	CMP/PL	Rn	SETT	
CMP/GE	Rm,Rn	CMP/PZ	Rn	TST	#imm,R0
CMP/GT	Rm,Rn	CMP/STR	Rm,Rn	TST	Rm,Rn

2. EX Group

ADD	#imm,Rn	MOVT	Rn	SHLL2	Rn
ADD	Rm,Rn	NEG	Rm,Rn	SHLL8	Rn
ADDC	Rm,Rn	NEGC	Rm,Rn	SHLR	Rn
ADDV	Rm,Rn	NOT	Rm,Rn	SHLR16	Rn
AND	#imm,R0	OR	#imm,R0	SHLR2	Rn
AND	Rm,Rn	OR	Rm,Rn	SHLR8	Rn
DIV0S	Rm,Rn	ROTCL	Rn	SUB	Rm,Rn
DIV0U		ROTCR	Rn	SUBC	Rm,Rn
DIV1	Rm,Rn	ROTL	Rn	SUBV	Rm,Rn
DT	Rn	ROTR	Rn	SWAP.B	Rm,Rn
EXTS.B	Rm,Rn	SHAD	Rm,Rn	SWAP.W	Rm,Rn
EXTS.W	Rm,Rn	SHAL	Rn	XOR	#imm,R0
EXTU.B	Rm,Rn	SHAR	Rn	XOR	Rm,Rn
EXTU.W	Rm,Rn	SHLD	Rm,Rn	XTRCT	Rm,Rn
MOV	#imm,Rn	SHLL	Rn		
MOVA	@(disp,PC),R0	SHLL16	Rn		

3. BR Group

BF	disp	BRA	disp	BT	disp
BF/S	disp	BSR	disp	BT/S	disp

FABS	DRn	FMOV.S	@Rm+,FRn	MOV.L	R0,@(disp,GBR)
FABS	FRn	FMOV.S	FRm,@(R0,Rn)	MOV.L	Rm,@(disp,Rn)
FLDI0	FRn	FMOV.S	FRm,@-Rn	MOV.L	Rm,@(R0,Rn)
FLDI1	FRn	FMOV.S	FRm,@Rn	MOV.L	Rm,@-Rn
FLDS	FRm,FPUL	FNEG	DRn	MOV.L	Rm,@Rn
FMOV	@(R0,Rm),DRn	FNEG	FRn	MOV.W	@(disp,GBR),R0
FMOV	@(R0,Rm),XDn	FSTS	FPUL,FRn	MOV.W	@(disp,PC),Rn
FMOV	@Rm,DRn	LDS	Rm,FPUL	MOV.W	@(disp,Rm),R0
FMOV	@Rm,XDn	MOV.B	@(disp,GBR),R0	MOV.W	@(R0,Rm),Rn
FMOV	@Rm+,DRn	MOV.B	@(disp,Rm),R0	MOV.W	@Rm,Rn
FMOV	@Rm+,XDn	MOV.B	@(R0,Rm),Rn	MOV.W	@Rm+,Rn
FMOV	DRm,@(R0,Rn)	MOV.B	@Rm,Rn	MOV.W	R0,@(disp,GBR)
FMOV	DRm,@-Rn	MOV.B	@Rm+,Rn	MOV.W	R0,@(disp,Rn)
FMOV	DRm,@Rn	MOV.B	R0,@(disp,GBR)	MOV.W	Rm,@(R0,Rn)
FMOV	DRm,DRn	MOV.B	R0,@(disp,Rn)	MOV.W	Rm,@-Rn
FMOV	DRm,XDn	MOV.B	Rm,@(R0,Rn)	MOV.W	Rm,@Rn
FMOV	FRm,FRn	MOV.B	Rm,@-Rn	MOVCA.L	R0,@Rn
FMOV	XDm,@(R0,Rn)	MOV.B	Rm,@Rn	OCBI	@Rn
FMOV	XDm,@-Rn	MOV.L	@(disp,GBR),R0	OCBP	@Rn
FMOV	XDm,@Rn	MOV.L	@(disp,PC),Rn	OCBWB	@Rn
FMOV	XDm,DRn	MOV.L	@(disp,Rm),Rn	PREF	@Rn
FMOV	XDm,XDn	MOV.L	@(R0,Rm),Rn	STS	FPUL,Rn
FMOV.S	@(R0,Rm),FRn	MOV.L	@Rm,Rn		
FMOV.S	@Rm,FRn	MOV.L	@Rm+,Rn		

5. FE Group

FADD	DRm,DRn	FIPR	FVm,FVn	FSQRT	DRn
FADD	FRm,FRn	FLOAT	FPUL,DRn	FSQRT	FRn
FCMP/EQ	FRm,FRn	FLOAT	FPUL,FRn	FSUB	DRm,DRn
FCMP/GT	FRm,FRn	FMAC	FR0,FRm,FRn	FSUB	FRm,FRn
FCNVDS	DRm,FPUL	FMUL	DRm,DRn	FTRC	DRm,FPUL
FCNVSD	FPUL,DRn	FMUL	FRm,FRn	FTRC	FRm,FPUL
FDIV	DRm,DRn	FRCHG		FTRV	XMTRX,FVn
FDIV	FRm,FRn	FSCHG			

AND.B	#imm,@(R0,GBR)	LDS	Rm,FPSCR	STC	SR,Rn
BRAF	Rm	LDS	Rm,MACH	STC	SSR,Rn
BSRF	Rm	LDS	Rm,MACL	STC	VBR,Rn
CLRMAC		LDS	Rm,PR	STC.L	DBR,@-Rn
CLRS		LDS.L	@Rm+,FPSCR	STC.L	GBR,@-Rn
DMULS.L	Rm,Rn	LDS.L	@Rm+,FPUL	STC.L	Rp_BANK,@-Rn
DMULU.L	Rm,Rn	LDS.L	@Rm+,MACH	STC.L	SGR,@-Rn
FCMP/EQ	DRm,DRn	LDS.L	@Rm+,MACL	STC.L	SPC,@-Rn
FCMP/GT	DRm,DRn	LDS.L	@Rm+,PR	STC.L	SR,@-Rn
JMP	@Rn	LDTLB		STC.L	SSR,@-Rn
JSR	@Rn	MAC.L	@Rm+,@Rn+	STC.L	VBR,@-Rn
LDC	Rm,DBR	MAC.W	@Rm+,@Rn+	STS	FPSCR,Rn
LDC	Rm,GBR	MUL.L	Rm,Rn	STS	MACH,Rn
LDC	Rm,Rp_BANK	MULS.W	Rm,Rn	STS	MACL,Rn
LDC	Rm,SPC	MULU.W	Rm,Rn	STS	PR,Rn
LDC	Rm,SR	OR.B	#imm,@(R0,GBR)	STS.L	FPSCR,@-Rn
LDC	Rm,SSR	RTE		STS.L	FPUL,@-Rn
LDC	Rm,VBR	RTS		STS.L	MACH,@-Rn
LDC.L	@Rm+,DBR	SETS		STS.L	MACL,@-Rn
LDC.L	@Rm+,GBR	SLEEP		STS.L	PR,@-Rn
LDC.L	@Rm+,Rp_BANK	STC	DBR,Rn	TAS.B	@Rn
LDC.L	@Rm+,SPC	STC	GBR,Rn	TRAPA	#imm
LDC.L	@Rm+,SR	STC	Rp_BANK,Rn	TST.B	#imm,@(R0,GBR)
LDC.L	@Rm+,SSR	STC	SGR,Rn	XOR.B	#imm,@(R0,GBR)
LDC.L	@Rm+,VBR	STC	SPC,Rn		

		2nd Instruction					
		MT	EX	BR	LS	FE	CO
1st Instruction	MT	O	O	O	O	O	X
	EX	O	X	O	O	O	X
	BR	O	O	X	O	O	X
	LS	O	O	O	X	O	X
	FE	O	O	O	O	X	X
	CO	X	X	X	X	X	X

Legend:
O: Can be executed in parallel
X: Cannot be executed in parallel

5.3 Execution Cycles and Pipeline Stalling

This LSI has three basic clocks: CPU clock (Ick), bus clock (Bck), and peripheral clock (Pck). Each hardware unit operates on one of these clocks, as follows:

- CPU clock: CPU, FPU, MMU, cache
- Bus clock: External bus controller
- Peripheral clock: Peripheral units

The frequency ratios of the three clocks are determined with the frequency control register (FRQCR). In this section, machine cycles are based on the CPU clock unless otherwise specified. For details on FRQCR, see section 12, Clock Pulse Generator (CPG).

Instruction execution cycles are summarized in table 5.3. Penalty cycles due to a pipeline stall are not considered in this table.

- Issue rate: Interval between the issue of an instruction and that of the next instruction
- Latency: Interval between the issue of an instruction and the generation of its result (completion)
- Instruction execution pattern (see figure 5.2)
- Locked pipeline stage: Pipeline stage which has been locked
- Lock start: Interval between the issue of an instruction and the start of locking (see table 5.3)
- Lock cycle: Period of locking (see table 5.3)

in figure 5.2. One instruction is separated from the next by the number of machine cycles for its issue rate. Normally, execution, data access, and write-back stages cannot be overlapped onto the same stages of another instruction; the only exception is when two instructions are executed in parallel under parallel-executability conditions. See (a) to (d) in figure 5.3 for some simple examples.

Latency is the interval between issue and completion of an instruction, and is also the interval between the execution of two instructions with an interdependent relationship. When there is interdependency between two instructions fetched simultaneously, the latter of the two is stalled for the following number of cycles:

- (Latency) cycles when there is flow dependency (read-after-write)
- (Latency – 1) or (latency – 2) cycles when there is output dependency (write-after-write)
  - Single/double-precision FDIV or FSQRT is the preceding instruction: (latency – 1) cycles
  - Other instructions in the FE group is the preceding instruction: (latency – 2) cycles
- Five or two cycles when there is anti-flow dependency (write-after-read), as in the following cases:
  - FTRV is the preceding instruction: 5 cycles
  - Double-precision FADD, FSUB, or FMUL is the preceding instruction: 2 cycles

In the case of flow dependency, the latency may be exceptionally increased or decreased, depending on the combination of sequential instructions (figure 5.3 (e)).

- When a floating-point computation is followed by a floating-point register store, latency of the floating-point computation may be decreased by one cycle.
- If there is a load of the shift amount immediately before an SHAD or SHLD instruction, latency of the load is increased by one cycle.
- If an instruction with latency of less than two cycles, including write-back to a floating-point register, is followed by a double-precision floating-point instruction, FIPR, or FTRV, latency of the first instruction is increased to two cycles.

The number of cycles in a pipeline stall due to flow dependency will vary depending on the combination of interdependent instructions or the fetch timing (see figure 5.3 (e)).

Output dependency occurs when the destination operands are the same in a preceding FE group instruction and a following LS group instruction.

For the stall cycles of an instruction with output dependency, the longest latency to the last write-back among all the destination operands must be applied instead of "latency" (see figure 5.3 (f)). A stall due to output dependency with respect to FPSCR, which reflects the result of a floating-point

floating-point registers, FADD is not stalled even if both instructions update the cause field of FPSCR.

Anti-flow dependency can occur only between a preceding double-precision FADD, FMUL, FSUB, or FTRV and a following FMOV, FLDI0, FLDI1, FABS, FNEG, or FSTS (see figure 5.3 (g)).

If an executing instruction locks any resource—i.e. a function block that performs a basic operation—a following instruction that happens to attempt to use the locked resource must be stalled (figure 5.3 (h)). This kind of stall can be compensated by inserting one or more instructions independent of the locked resource to separate the interfering instructions. For example, when a load instruction and an ADD instruction that references the loaded value are consecutive, the 2-cycle stall of the ADD is eliminated by inserting three instructions without dependency. Software performance can be improved by such instruction scheduling.

Other penalties arise in the event of exceptions or external data accesses, as follows.

- Instruction TLB miss
- Instruction access to external memory (instruction cache miss, etc.)
- Data access to external memory (operand cache miss, etc.)
- Data access to a memory-mapped control register

During the penalty cycles of an instruction TLB miss or external instruction access, no instruction is issued, but execution of instructions that have already been issued continues. The penalty for a data access is a pipeline freeze: that is, the execution of uncompleted instructions is interrupted until the arrival of the requested data. The number of penalty cycles for instruction and data accesses is largely dependent on the user's memory subsystems.

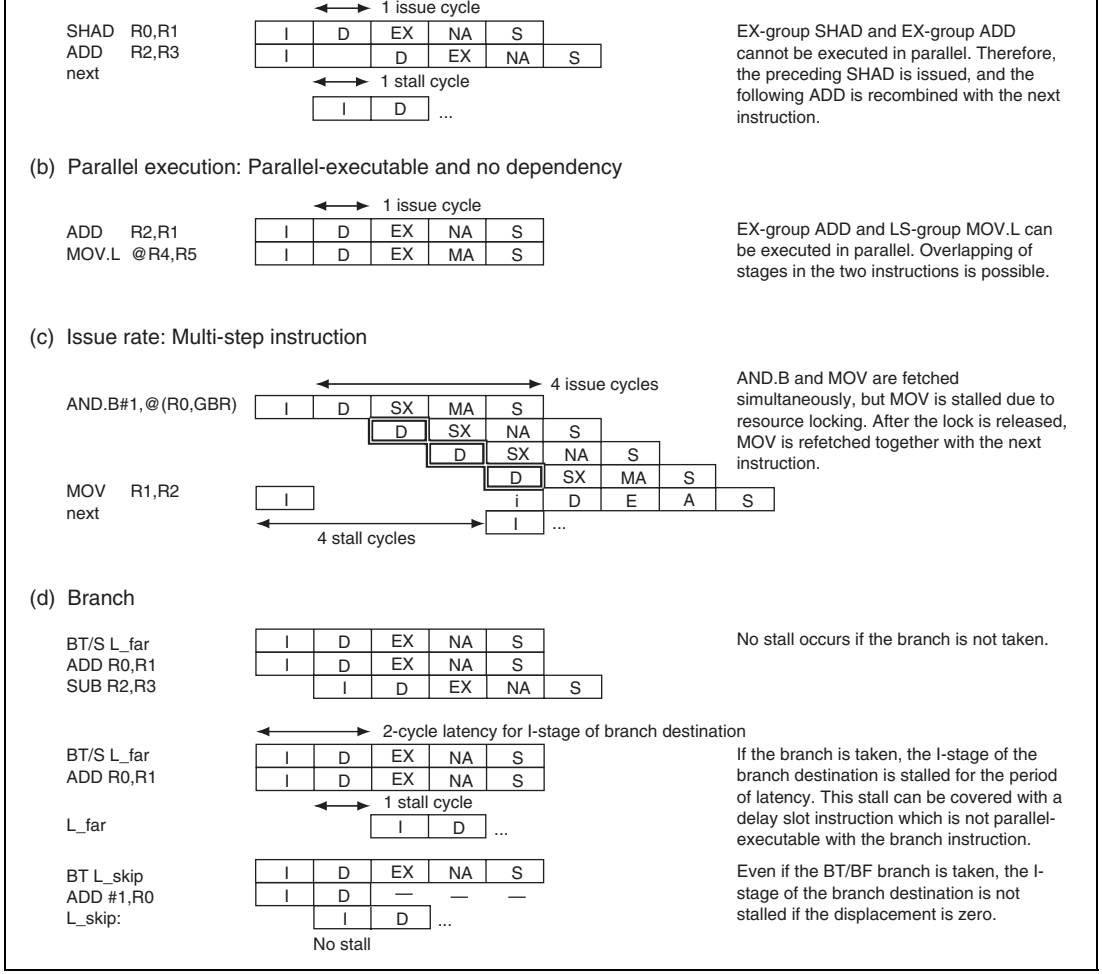


Figure 5.3 Examples of Pipelined Execution

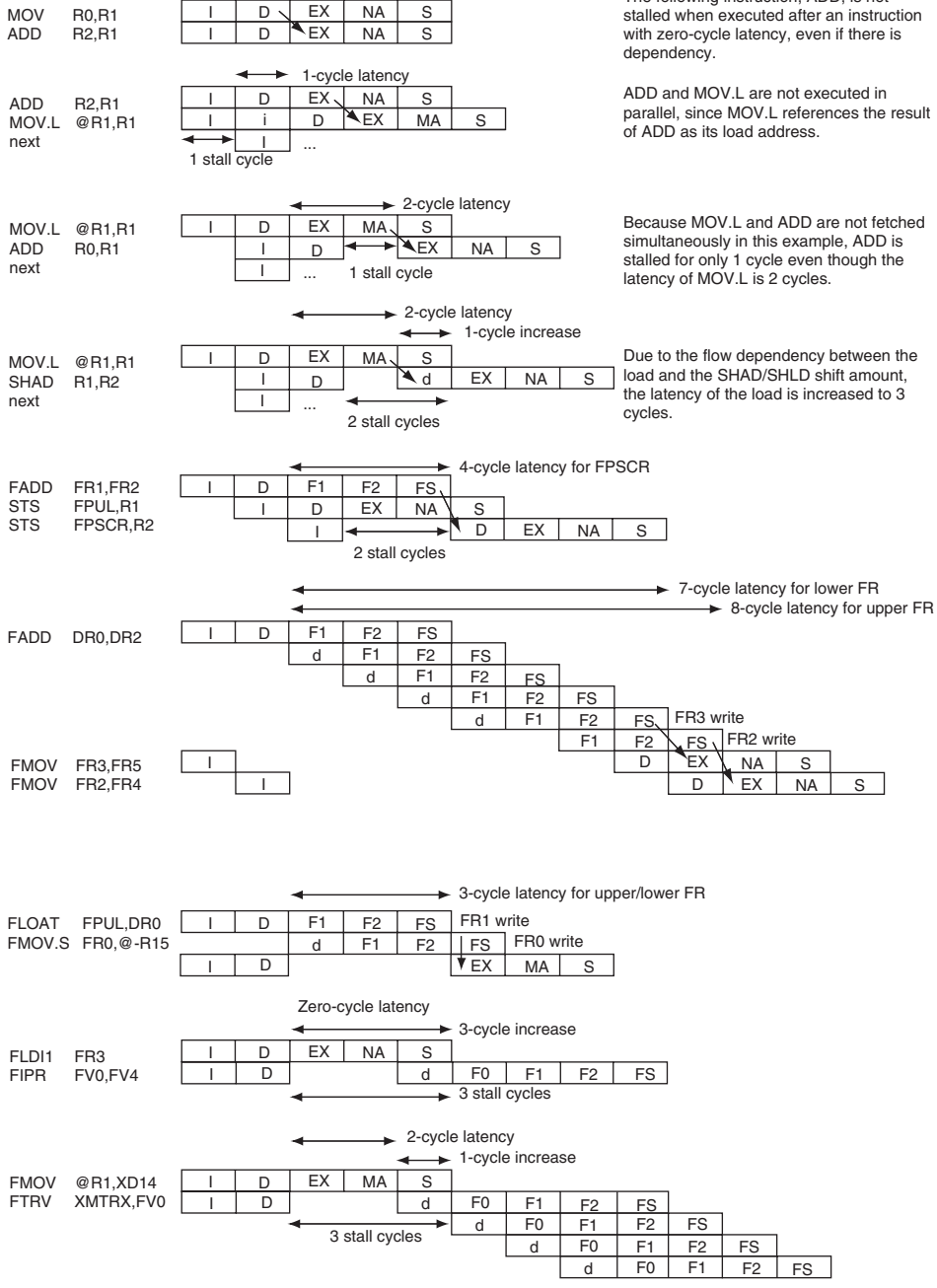


Figure 5.3 Examples of Pipelined Execution (cont)

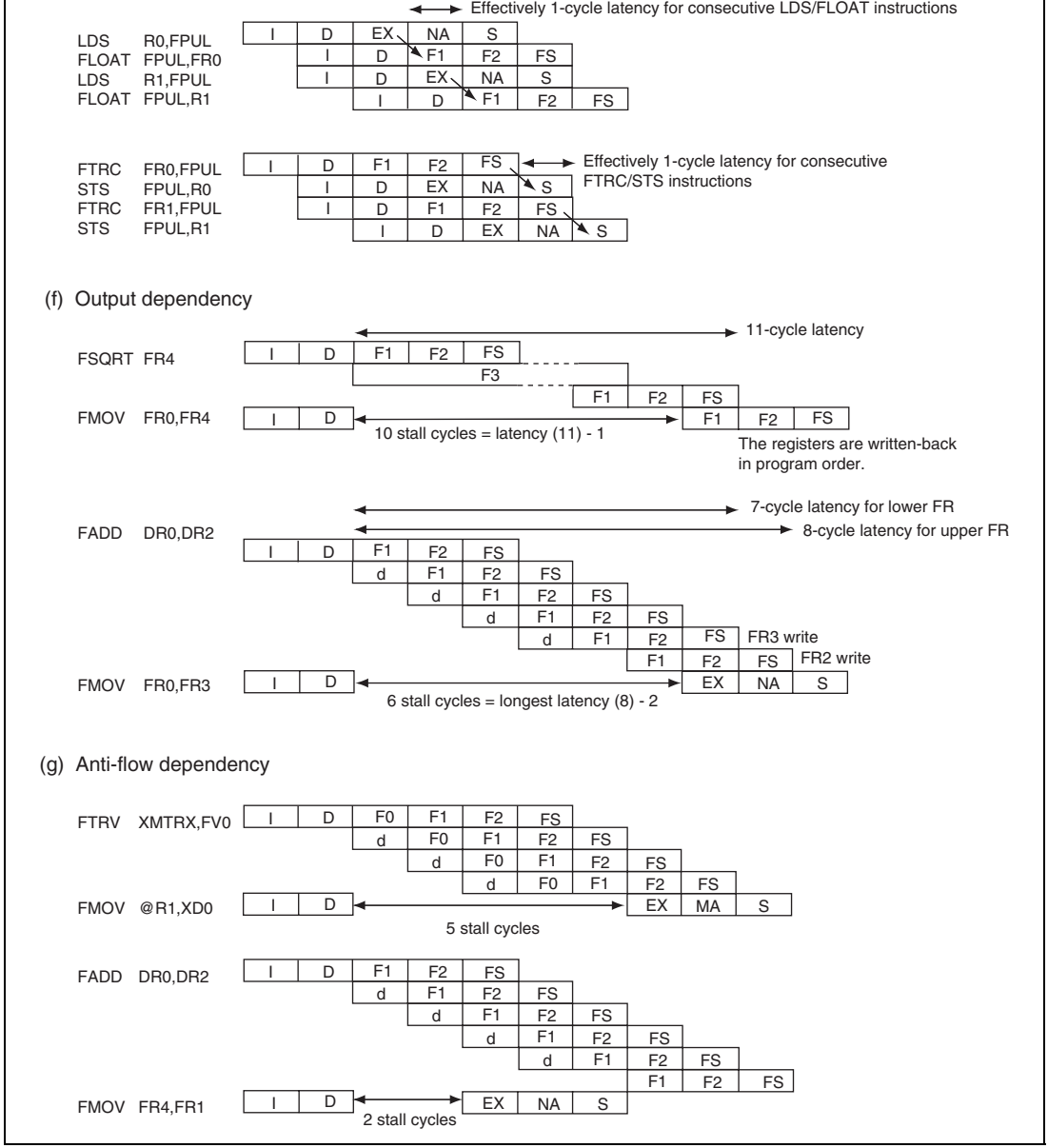


Figure 5.3 Examples of Pipelined Execution (cont)

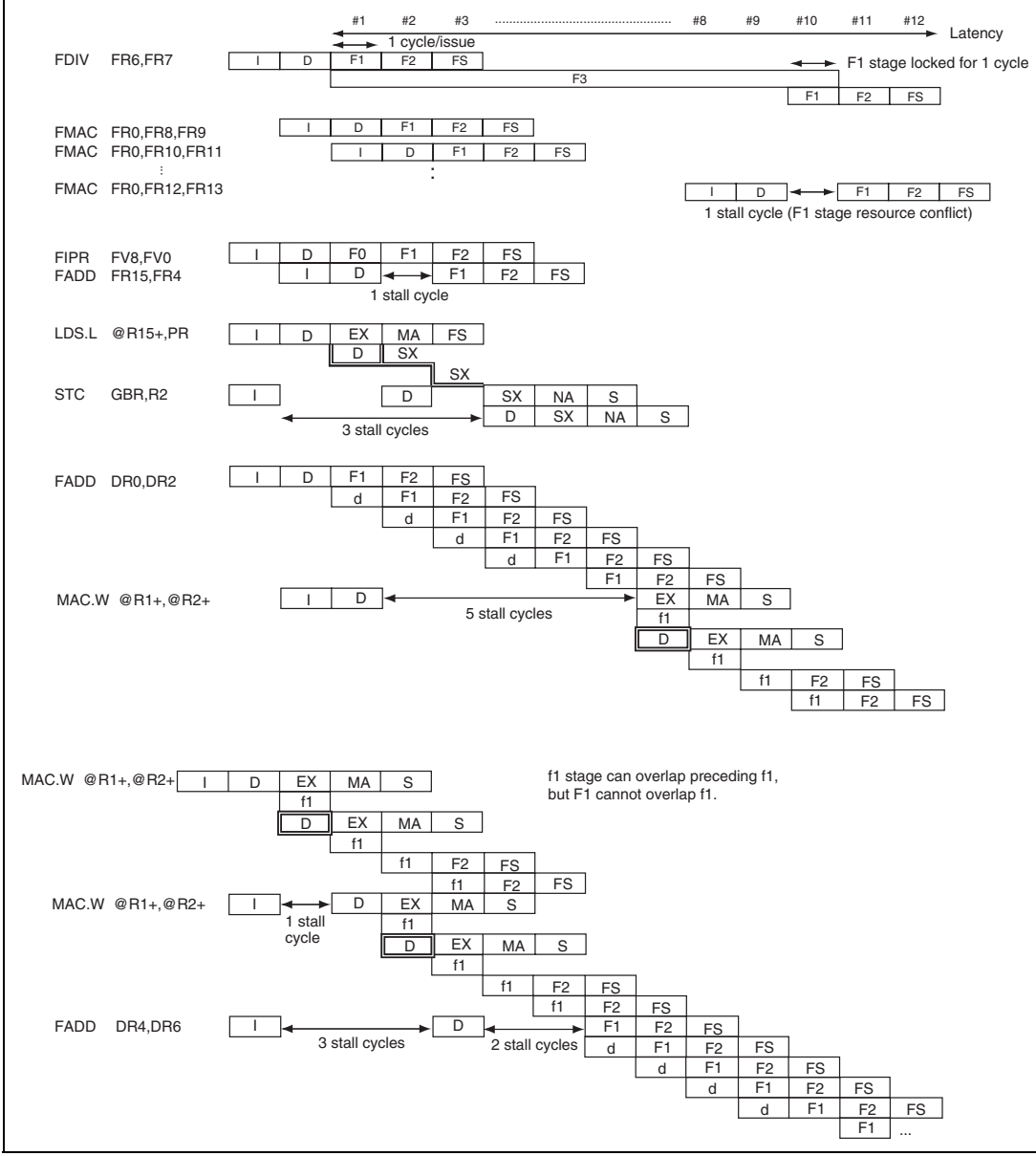


Figure 5.3 Examples of Pipelined Execution (cont)

Functional Category	No.	Instruction		Instruction Group	Issue Rate	Latency	Execution Pattern	Lock		
								Stage	Start	Cycles
Data transfer instructions	1	EXTS.B	Rm,Rn	EX	1	1	#1	—	—	—
	2	EXTS.W	Rm,Rn	EX	1	1	#1	—	—	—
	3	EXTU.B	Rm,Rn	EX	1	1	#1	—	—	—
	4	EXTU.W	Rm,Rn	EX	1	1	#1	—	—	—
	5	MOV	Rm,Rn	MT	1	0	#1	—	—	—
	6	MOV	#imm,Rn	EX	1	1	#1	—	—	—
	7	MOVA	@(disp,PC),R0	EX	1	1	#1	—	—	—
	8	MOV.W	@(disp,PC),Rn	LS	1	2	#2	—	—	—
	9	MOV.L	@(disp,PC),Rn	LS	1	2	#2	—	—	—
	10	MOV.B	@Rm,Rn	LS	1	2	#2	—	—	—
	11	MOV.W	@Rm,Rn	LS	1	2	#2	—	—	—
	12	MOV.L	@Rm,Rn	LS	1	2	#2	—	—	—
	13	MOV.B	@Rm+,Rn	LS	1	1/2	#2	—	—	—
	14	MOV.W	@Rm+,Rn	LS	1	1/2	#2	—	—	—
	15	MOV.L	@Rm+,Rn	LS	1	1/2	#2	—	—	—
	16	MOV.B	@(disp,Rm),R0	LS	1	2	#2	—	—	—
	17	MOV.W	@(disp,Rm),R0	LS	1	2	#2	—	—	—
	18	MOV.L	@(disp,Rm),Rn	LS	1	2	#2	—	—	—
	19	MOV.B	@(R0,Rm),Rn	LS	1	2	#2	—	—	—
	20	MOV.W	@(R0,Rm),Rn	LS	1	2	#2	—	—	—
	21	MOV.L	@(R0,Rm),Rn	LS	1	2	#2	—	—	—
	22	MOV.B	@(disp,GBR),R0	LS	1	2	#3	—	—	—
	23	MOV.W	@(disp,GBR),R0	LS	1	2	#3	—	—	—
	24	MOV.L	@(disp,GBR),R0	LS	1	2	#3	—	—	—
	25	MOV.B	Rm,@Rn	LS	1	1	#2	—	—	—
	26	MOV.W	Rm,@Rn	LS	1	1	#2	—	—	—
	27	MOV.L	Rm,@Rn	LS	1	1	#2	—	—	—
	28	MOV.B	Rm,@-Rn	LS	1	1/1	#2	—	—	—
	29	MOV.W	Rm,@-Rn	LS	1	1/1	#2	—	—	—
	30	MOV.L	Rm,@-Rn	LS	1	1/1	#2	—	—	—
	31	MOV.B	R0,@(disp,Rn)	LS	1	1	#2	—	—	—
	32	MOV.W	R0,@(disp,Rn)	LS	1	1	#2	—	—	—
	33	MOV.L	Rm,@(disp,Rn)	LS	1	1	#2	—	—	—

Category	No.	Instruction		Group	Rate	Latency	Pattern	Stage	Start	Cycles
Data transfer instructions	34	MOV.B	Rm,@(R0,Rn)	LS	1	1	#2	—	—	—
	35	MOV.W	Rm,@(R0,Rn)	LS	1	1	#2	—	—	—
	36	MOV.L	Rm,@(R0,Rn)	LS	1	1	#2	—	—	—
	37	MOV.B	R0,@(disp,GBR)	LS	1	1	#3	—	—	—
	38	MOV.W	R0,@(disp,GBR)	LS	1	1	#3	—	—	—
	39	MOV.L	R0,@(disp,GBR)	LS	1	1	#3	—	—	—
	40	MOVCA.L	R0,@Rn	LS	1	3 to 7	#12	MA	4	3 to 7
	41	MOVT	Rn	EX	1	1	#1	—	—	—
	42	OCBI	@Rn	LS	1	1 to 2	#10	MA	4	1 to 2
	43	OCBP	@Rn	LS	1	1 to 5	#11	MA	4	1 to 5
	44	OCBWB	@Rn	LS	1	1 to 5	#11	MA	4	1 to 5
	45	PREF	@Rn	LS	1	1	#2	—	—	—
	46	SWAP.B	Rm,Rn	EX	1	1	#1	—	—	—
	47	SWAP.W	Rm,Rn	EX	1	1	#1	—	—	—
	48	XTRCT	Rm,Rn	EX	1	1	#1	—	—	—
Fixed-point arithmetic instructions	49	ADD	Rm,Rn	EX	1	1	#1	—	—	—
	50	ADD	#imm,Rn	EX	1	1	#1	—	—	—
	51	ADDC	Rm,Rn	EX	1	1	#1	—	—	—
	52	ADDV	Rm,Rn	EX	1	1	#1	—	—	—
	53	CMP/EQ	#imm,R0	MT	1	1	#1	—	—	—
	54	CMP/EQ	Rm,Rn	MT	1	1	#1	—	—	—
	55	CMP/GE	Rm,Rn	MT	1	1	#1	—	—	—
	56	CMP/GT	Rm,Rn	MT	1	1	#1	—	—	—
	57	CMP/HI	Rm,Rn	MT	1	1	#1	—	—	—
	58	CMP/HS	Rm,Rn	MT	1	1	#1	—	—	—
	59	CMP/PL	Rn	MT	1	1	#1	—	—	—
	60	CMP/PZ	Rn	MT	1	1	#1	—	—	—
	61	CMP/STR	Rm,Rn	MT	1	1	#1	—	—	—
	62	DIV0S	Rm,Rn	EX	1	1	#1	—	—	—
	63	DIV0U		EX	1	1	#1	—	—	—
	64	DIV1	Rm,Rn	EX	1	1	#1	—	—	—
	65	DMULS.L	Rm,Rn	CO	2	4/4	#34	F1	4	2
	66	DMULU.L	Rm,Rn	CO	2	4/4	#34	F1	4	2
	67	DT	Rn	EX	1	1	#1	—	—	—

Category	No.	Instruction		Group	Rate	Latency	Pattern	Stage	Start	Cycles
Fixed-point arithmetic instructions	68	MAC.L	@Rm+,@Rn+	CO	2	2/2/4/4	#35	F1	4	2
	69	MAC.W	@Rm+,@Rn+	CO	2	2/2/4/4	#35	F1	4	2
	70	MUL.L	Rm,Rn	CO	2	4/4	#34	F1	4	2
	71	MULS.W	Rm,Rn	CO	2	4/4	#34	F1	4	2
	72	MULU.W	Rm,Rn	CO	2	4/4	#34	F1	4	2
	73	NEG	Rm,Rn	EX	1	1	#1	—	—	—
	74	NEGC	Rm,Rn	EX	1	1	#1	—	—	—
	75	SUB	Rm,Rn	EX	1	1	#1	—	—	—
	76	SUBC	Rm,Rn	EX	1	1	#1	—	—	—
	77	SUBV	Rm,Rn	EX	1	1	#1	—	—	—
Logical instructions	78	AND	Rm,Rn	EX	1	1	#1	—	—	—
	79	AND	#imm,R0	EX	1	1	#1	—	—	—
	80	AND.B	#imm,@(R0,GBR)	CO	4	4	#6	—	—	—
	81	NOT	Rm,Rn	EX	1	1	#1	—	—	—
	82	OR	Rm,Rn	EX	1	1	#1	—	—	—
	83	OR	#imm,R0	EX	1	1	#1	—	—	—
	84	OR.B	#imm,@(R0,GBR)	CO	4	4	#6	—	—	—
	85	TAS.B	@Rn	CO	5	5	#7	—	—	—
	86	TST	Rm,Rn	MT	1	1	#1	—	—	—
	87	TST	#imm,R0	MT	1	1	#1	—	—	—
	88	TST.B	#imm,@(R0,GBR)	CO	3	3	#5	—	—	—
	89	XOR	Rm,Rn	EX	1	1	#1	—	—	—
	90	XOR	#imm,R0	EX	1	1	#1	—	—	—
	91	XOR.B	#imm,@(R0,GBR)	CO	4	4	#6	—	—	—
Shift instructions	92	ROTL	Rn	EX	1	1	#1	—	—	—
	93	ROTR	Rn	EX	1	1	#1	—	—	—
	94	ROTCL	Rn	EX	1	1	#1	—	—	—
	95	ROTCR	Rn	EX	1	1	#1	—	—	—
	96	SHAD	Rm,Rn	EX	1	1	#1	—	—	—
	97	SHAL	Rn	EX	1	1	#1	—	—	—
	98	SHAR	Rn	EX	1	1	#1	—	—	—
	99	SHLD	Rm,Rn	EX	1	1	#1	—	—	—
	100	SHLL	Rn	EX	1	1	#1	—	—	—
	101	SHLL2	Rn	EX	1	1	#1	—	—	—

Category	No.	Instruction		Group	Rate	Latency	Pattern	Stage	Start	Cycles
Shift instructions	102	SHLL8	Rn	EX	1	1	#1	—	—	—
	103	SHLL16	Rn	EX	1	1	#1	—	—	—
	104	SHLR	Rn	EX	1	1	#1	—	—	—
	105	SHLR2	Rn	EX	1	1	#1	—	—	—
	106	SHLR8	Rn	EX	1	1	#1	—	—	—
	107	SHLR16	Rn	EX	1	1	#1	—	—	—
Branch instructions	108	BF	disp	BR	1	2 (or 1)	#1	—	—	—
	109	BF/S	disp	BR	1	2 (or 1)	#1	—	—	—
	110	BT	disp	BR	1	2 (or 1)	#1	—	—	—
	111	BT/S	disp	BR	1	2 (or 1)	#1	—	—	—
	112	BRA	disp	BR	1	2	#1	—	—	—
	113	BRAF	Rm	CO	2	3	#4	—	—	—
	114	BSR	disp	BR	1	2	#14	SX	3	2
	115	BSRF	Rm	CO	2	3	#24	SX	3	2
	116	JMP	@Rn	CO	2	3	#4	—	—	—
	117	JSR	@Rn	CO	2	3	#24	SX	3	2
	118	RTS		CO	2	3	#4	—	—	—
System control instructions	119	NOP		MT	1	0	#1	—	—	—
	120	CLRMAC		CO	1	3	#28	F1	3	2
	121	CLRS		CO	1	1	#1	—	—	—
	122	CLRT		MT	1	1	#1	—	—	—
	123	SETS		CO	1	1	#1	—	—	—
	124	SETT		MT	1	1	#1	—	—	—
	125	TRAPA	#imm	CO	7	7	#13	—	—	—
	126	RTE		CO	5	5	#8	—	—	—
	127	SLEEP		CO	4	4	#9	—	—	—
	128	LDTLB		CO	1	1	#2	—	—	—
	129	LDC	Rm,DBR	CO	1	3	#14	SX	3	2
	130	LDC	Rm,GBR	CO	3	3	#15	SX	3	2
	131	LDC	Rm,Rp_BANK	CO	1	3	#14	SX	3	2
	132	LDC	Rm,SR	CO	4	4	#16	SX	3	2
	133	LDC	Rm,SSR	CO	1	3	#14	SX	3	2
	134	LDC	Rm,SPC	CO	1	3	#14	SX	3	2
	135	LDC	Rm,VBR	CO	1	3	#14	SX	3	2

Category	No.	Instruction		Group	Rate	Latency	Pattern	Stage	Start	Cycles
System control instructions	136	LDC.L	@Rm+,DBR	CO	1	1/3	#17	SX	3	2
	137	LDC.L	@Rm+,GBR	CO	3	3/3	#18	SX	3	2
	138	LDC.L	@Rm+,Rp_BANK	CO	1	1/3	#17	SX	3	2
	139	LDC.L	@Rm+,SR	CO	4	4/4	#19	SX	3	2
	140	LDC.L	@Rm+,SSR	CO	1	1/3	#17	SX	3	2
	141	LDC.L	@Rm+,SPC	CO	1	1/3	#17	SX	3	2
	142	LDC.L	@Rm+,VBR	CO	1	1/3	#17	SX	3	2
	143	LDS	Rm,MACH	CO	1	3	#28	F1	3	2
	144	LDS	Rm,MACL	CO	1	3	#28	F1	3	2
	145	LDS	Rm,PR	CO	2	3	#24	SX	3	2
	146	LDS.L	@Rm+,MACH	CO	1	1/3	#29	F1	3	2
	147	LDS.L	@Rm+,MACL	CO	1	1/3	#29	F1	3	2
	148	LDS.L	@Rm+,PR	CO	2	2/3	#25	SX	3	2
	149	STC	DBR,Rn	CO	2	2	#20	—	—	—
	150	STC	SGR,Rn	CO	3	3	#21	—	—	—
	151	STC	GBR,Rn	CO	2	2	#20	—	—	—
	152	STC	Rp_BANK,Rn	CO	2	2	#20	—	—	—
	153	STC	SR,Rn	CO	2	2	#20	—	—	—
	154	STC	SSR,Rn	CO	2	2	#20	—	—	—
	155	STC	SPC,Rn	CO	2	2	#20	—	—	—
	156	STC	VBR,Rn	CO	2	2	#20	—	—	—
	157	STC.L	DBR,@-Rn	CO	2	2/2	#22	—	—	—
	158	STC.L	SGR,@-Rn	CO	3	3/3	#23	—	—	—
	159	STC.L	GBR,@-Rn	CO	2	2/2	#22	—	—	—
	160	STC.L	Rp_BANK,@-Rn	CO	2	2/2	#22	—	—	—
	161	STC.L	SR,@-Rn	CO	2	2/2	#22	—	—	—
	162	STC.L	SSR,@-Rn	CO	2	2/2	#22	—	—	—
	163	STC.L	SPC,@-Rn	CO	2	2/2	#22	—	—	—
	164	STC.L	VBR,@-Rn	CO	2	2/2	#22	—	—	—
	165	STS	MACH,Rn	CO	1	3	#30	—	—	—
	166	STS	MACL,Rn	CO	1	3	#30	—	—	—
	167	STS	PR,Rn	CO	2	2	#26	—	—	—
	168	STS.L	MACH,@-Rn	CO	1	1/1	#31	—	—	—
	169	STS.L	MACL,@-Rn	CO	1	1/1	#31	—	—	—
	170	STS.L	PR,@-Rn	CO	2	2/2	#27	—	—	—

Category	No.	Instruction		Group	Rate	Latency	Pattern	Stage	Start	Cycles
Single-precision floating-point instructions	171	FLDI0	FRn	LS	1	0	#1	—	—	—
	172	FLDI1	FRn	LS	1	0	#1	—	—	—
	173	FMOV	FRm,FRn	LS	1	0	#1	—	—	—
	174	FMOV.S	@Rm,FRn	LS	1	2	#2	—	—	—
	175	FMOV.S	@Rm+,FRn	LS	1	1/2	#2	—	—	—
	176	FMOV.S	@(R0,Rm),FRn	LS	1	2	#2	—	—	—
	177	FMOV.S	FRm,@Rn	LS	1	1	#2	—	—	—
	178	FMOV.S	FRm,@-Rn	LS	1	1/1	#2	—	—	—
	179	FMOV.S	FRm,@(R0,Rn)	LS	1	1	#2	—	—	—
	180	FLDS	FRm,FPUL	LS	1	0	#1	—	—	—
	181	FSTS	FPUL,FRn	LS	1	0	#1	—	—	—
	182	FABS	FRn	LS	1	0	#1	—	—	—
	183	FADD	FRm,FRn	FE	1	3/4	#36	—	—	—
	184	FCMP/EQ	FRm,FRn	FE	1	2/4	#36	—	—	—
	185	FCMP/GT	FRm,FRn	FE	1	2/4	#36	—	—	—
	186	FDIV	FRm,FRn	FE	1	12/13	#37	F3	2	10
								F1	11	1
	187	FLOAT	FPUL,FRn	FE	1	3/4	#36	—	—	—
	188	FMAC	FR0,FRm,FRn	FE	1	3/4	#36	—	—	—
	189	FMUL	FRm,FRn	FE	1	3/4	#36	—	—	—
	190	FNEG	FRn	LS	1	0	#1	—	—	—
	191	FSQRT	FRn	FE	1	11/12	#37	F3	2	9
								F1	10	1
	192	FSUB	FRm,FRn	FE	1	3/4	#36	—	—	—
	193	FTRC	FRm,FPUL	FE	1	3/4	#36	—	—	—
	194	FMOV	DRm,DRn	LS	1	0	#1	—	—	—
	195	FMOV	@Rm,DRn	LS	1	2	#2	—	—	—
	196	FMOV	@Rm+,DRn	LS	1	1/2	#2	—	—	—
	197	FMOV	@(R0,Rm),DRn	LS	1	2	#2	—	—	—
	198	FMOV	DRm,@Rn	LS	1	1	#2	—	—	—
	199	FMOV	DRm,@-Rn	LS	1	1/1	#2	—	—	—
	200	FMOV	DRm,@(R0,Rn)	LS	1	1	#2	—	—	—

Category	No.	Instruction		Group	Rate	Latency	Pattern	Stage	Start	Cycles
Double-precision floating-point instructions	201	FABS	DRn	LS	1	0	#1	—	—	—
	202	FADD	DRm,DRn	FE	1	(7, 8)/9	#39	F1	2	6
	203	FCMP/EQ	DRm,DRn	CO	2	3/5	#40	F1	2	2
	204	FCMP/GT	DRm,DRn	CO	2	3/5	#40	F1	2	2
	205	FCNVDS	DRm,FPUL	FE	1	4/5	#38	F1	2	2
	206	FCNVSD	FPUL,DRn	FE	1	(3, 4)/5	#38	F1	2	2
	207	FDIV	DRm,DRn	FE	1	(24, 25)/26	#41	F3	2	23
								F1	22	3
								F1	2	2
	208	FLOAT	FPUL,DRn	FE	1	(3, 4)/5	#38	F1	2	2
	209	FMUL	DRm,DRn	FE	1	(7, 8)/9	#39	F1	2	6
	210	FNEG	DRn	LS	1	0	#1	—	—	—
	211	FSQRT	DRn	FE	1	(23, 24)/25	#41	F3	2	22
								F1	21	3
								F1	2	2
	212	FSUB	DRm,DRn	FE	1	(7, 8)/9	#39	F1	2	6
	213	FTRC	DRm,FPUL	FE	1	4/5	#38	F1	2	2
FPU system control instructions	214	LDS	Rm,FPUL	LS	1	1	#1	—	—	—
	215	LDS	Rm,FPSCR	CO	1	4	#32	F1	3	3
	216	LDS.L	@Rm+,FPUL	CO	1	1/2	#2	—	—	—
	217	LDS.L	@Rm+,FPSCR	CO	1	1/4	#33	F1	3	3
	218	STS	FPUL,Rn	LS	1	3	#1	—	—	—
	219	STS	FPSCR,Rn	CO	1	3	#1	—	—	—
	220	STS.L	FPUL,@-Rn	CO	1	1/1	#2	—	—	—
	221	STS.L	FPSCR,@-Rn	CO	1	1/1	#2	—	—	—
Graphics acceleration instructions	222	FMOV	DRm,XDn	LS	1	0	#1	—	—	—
	223	FMOV	XDm,DRn	LS	1	0	#1	—	—	—
	224	FMOV	XDm,XDn	LS	1	0	#1	—	—	—
	225	FMOV	@Rm,XDn	LS	1	2	#2	—	—	—
	226	FMOV	@Rm+,XDn	LS	1	1/2	#2	—	—	—
	227	FMOV	@(R0,Rm),XDn	LS	1	2	#2	—	—	—
	228	FMOV	XDm,@Rn	LS	1	1	#2	—	—	—
	229	FMOV	XDm,@-Rm	LS	1	1/1	#2	—	—	—
	230	FMOV	XDm,@(R0,Rn)	LS	1	1	#2	—	—	—

Category	No.	Instruction		Group	Rate	Latency	Pattern	Stage	Start	Cycles
Graphics acceleration instructions	231	FIPR	FVm,FVn	FE	1	4/5	#42	F1	3	1
	232	FRCHG		FE	1	1/4	#36	—	—	—
	233	FSCHG		FE	1	1/4	#36	—	—	—
	234	FTRV	XMTRX,FVn	FE	1	(5, 5, 6, 7)/8	#43	F0	2	4
								F1	3	4

Notes: 1. See table 5.1 for the instruction groups.

2. Latency "L1/L2... ": Latency corresponding to a write to each register, including MACH/MACL/FPSCR.

   Example: MOV.B @Rm+, Rn "1/2": Latency for Rm is 1 cycle and latency for Rn is 2 cycles.

3. Branch latency: Interval until the branch destination instruction is fetched
4. Conditional branch latency "2 (or 1) ": Latency is 2 for a non-zero displacement, and 1 for a zero displacement.
5. Double-precision floating-point instruction latency " (L1, L2)/L3": L1 is the latency for FR [n+1], L2 that for FR [n], and L3 that for FPSCR.
6. FTRV latency " (L1, L2, L3, L4)/L5": L1 is the latency for FR [n], L2 that for FR [n+1], L3 that for FR [n+2], L4 that for FR [n+3], and L5 that for FPSCR.
7. Latency "L1/L2/L3/L4" of MAC.L and MAC.W instructions: L1 is the latency for Rm, L2 that for Rn, L3 that for MACH, and L4 that for MACL.
8. Latency "L1/L2" of MUL.L, MULS.W, MULU.W, DMULS.L, and DMULU.L instructions: L1 is the latency for MACH, and L2 that for MACL.
9. Execution pattern: Instruction execution pattern number (see figure 5.2)
10. Lock/stage: Stage locked by the instruction
11. Lock/start: Locking start cycle; 1 is the first D-stage of the instruction.
12. Lock/cycles: Number of cycles locked

Exceptions:

1. When a floating-point computation instruction is followed by an FMOV store instruction, an STS FPUL, Rn instruction, or an STS.L FPUL, @-Rn instruction, latency of the floating-point computation is decreased by 1 cycle.
2. When the preceding instruction loads the shift amount of the following SHAD/SHLD, latency of the load is increased by 1 cycle.
3. When an LS group instruction with latency of less than 3 cycles is followed by a double-precision floating-point instruction, FIPR, or FTRV, latency of the first instruction is increased to 3 cycles.

   Example: In the case of FMOV FR4,FR0 and FIPR FV0,FV4, FIPR is stalled for 2 cycles.

4. When MAC.W/MAC.L/MUL.L/MULS.W/MULU.W/DMULS.L/DMULU.L is followed by an STS.L MACH/MACL, @-Rn instruction, latency of MAC.W/MAC.L/MUL.L/MULS.W/MULU.W/DMULS.L/DMULU.L is 5 cycles.

6. When an LDS to MACH/MACL is followed by an STS.L MACH/MACL, @-Rn instruction, latency of the LDS to MACH/MACL is 4 cycles.
7. When an LDS to MACH/MACL is followed by MAC.W/MAC.L/MUL.L/MULS.W/ MULU.W/DMULS.L/DMULU.L, latency of the LDS to MACH/MACL is 1 cycle.
8. When an FSCHG or FRCHG instruction is followed by an LS group instruction that reads from or writes to a floating-point register, the aforementioned LS group instructions cannot be executed in parallel.
9. When a single-precision FTRC instruction is followed by an STS FPUL, Rn instruction, latency of the single-precision FTRC instruction is 1 cycle.

5.4 Usage Note

The following are additional notes on pipeline operation and the method of calculating the number of clock cycles.

The number of states (CPU clock cycles) required for stages where an external bus access, etc., occurs may include an increased number of cycles, in addition to the number of memory access cycles set by the bus state controller (BSC), etc.

For example, the occurrence of the following may result in idle cycles as observed from the external bus.

1. Transfer of data from the logical address bus to the physical address bus
2. Transfer of data between buses using different operation clocks

The stages where external memory access occurs include some instruction fetch (I) and some memory access (MA) stages.

The SH-4 supports an 8-bit address space identifier, a 32-bit virtual address space, and a 29-bit external memory space. Address translation from virtual addresses to physical addresses is enabled by the memory management unit (MMU) in the SH-4. The MMU performs high-speed address translation by caching user-created address translation table information in an address translation buffer (translation lookaside buffer: TLB).

The SH-4 has four instruction TLB (ITLB) entries and 64 unified TLB (UTLB) entries. UTLB copies are stored in the ITLB by hardware. A paging system is used for address translation, with four page sizes (1, 4, and 64 Kbytes, and 1 Mbyte) supported. It is possible to set the virtual address space access right and implement memory protection independently for privileged mode and user mode.

6.1 Overview of the MMU

The MMU was conceived as a means of making efficient use of physical memory. As shown in figure 6.1, when a process is smaller in size than the physical memory, the entire process can be mapped onto physical memory, but if the process increases in size to the point where it does not fit into physical memory, it becomes necessary to divide the process into smaller parts, and map the parts requiring execution onto physical memory as occasion arises ((1) in figure 6.1). Having this mapping onto physical memory executed consciously by the process itself imposes a heavy burden on the process. The virtual memory system was devised as a means of handling all physical memory mapping to reduce this burden ((2) in figure 6.1). With a virtual memory system, the size of the available virtual memory is much larger than the actual physical memory, and processes are mapped onto this virtual memory. Thus processes only have to consider their operation in virtual memory, and mapping from virtual memory to physical memory is handled by the MMU. The MMU is normally managed by the OS, and physical memory switching is carried out so as to enable the virtual memory required by a process to be mapped smoothly onto physical memory. Physical memory switching is performed via secondary storage, etc.

The virtual memory system that came into being in this way works to best effect in a time sharing system (TSS) that allows a number of processes to run simultaneously ((3) in figure 6.1). Running a number of processes in a TSS did not increase efficiency since each process had to take account of physical memory mapping. Efficiency is improved and the load on each process reduced by the use of a virtual memory system ((4) in figure 6.1). In this virtual memory system, virtual memory is allocated to each process. The task of the MMU is to map a number of virtual memory areas onto physical memory in an efficient manner. It is also provided with memory protection functions to prevent a process from inadvertently accessing another process's physical memory.

it may happen that the translation information has not been recorded in the MMU, or the virtual memory of a different process is accessed by mistake. In such cases, 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 be implemented by software alone, having address translation performed by software each time a process accessed physical memory would be very inefficient. For this reason, a buffer for address translation (the translation lookaside buffer: TLB) is provided by hardware, and frequently used address translation information is placed here. The TLB can be described as a cache for address translation information. However, unlike a cache, if address translation fails—that is, if an exception occurs—switching of the address translation information is normally performed by software. Thus memory management can be performed in a flexible manner by software.

There are two methods by which the MMU can perform mapping from virtual memory to physical memory: the paging method, using fixed-length address translation, and the segment method, using variable-length address translation. With the paging method, the unit of translation is a fixed-size address space called a page (usually from 1 to 64 Kbytes in size).

In the following descriptions, the address space in virtual memory in the SH-4 is referred to as virtual address space, and the address space in physical memory as physical address space.

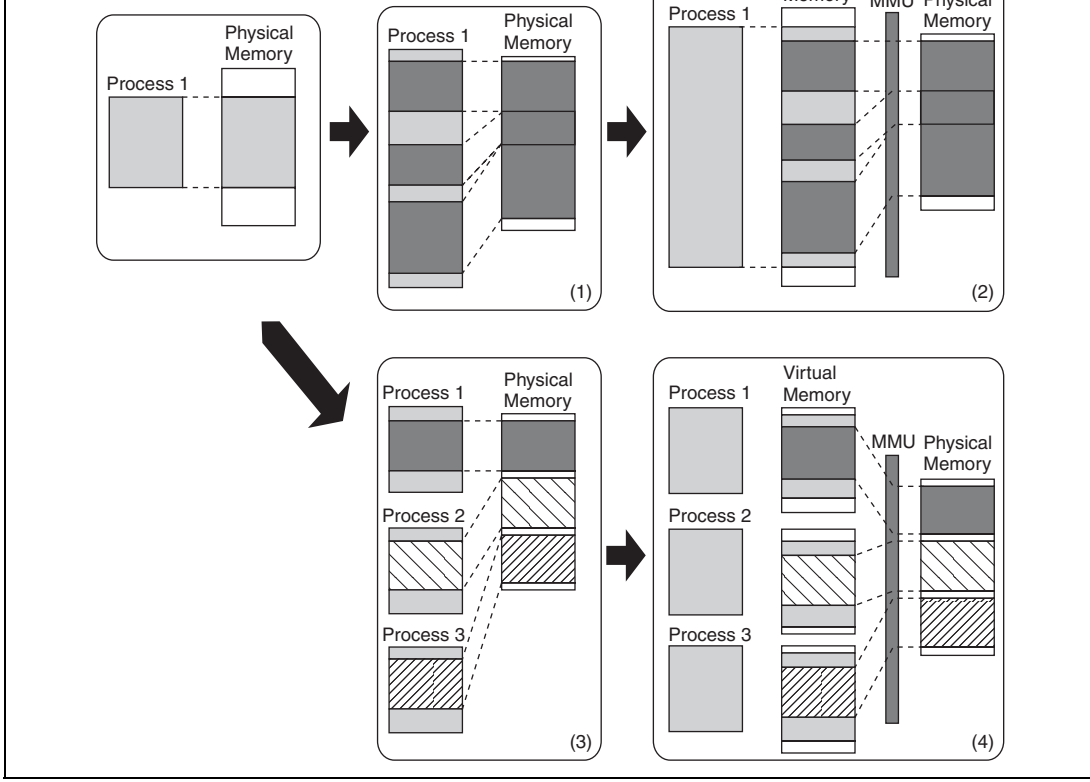


Figure 6.1 Role of the MMU

6.1.1 Address Spaces

(1) Physical Address Space

The SH-4 supports a 32-bit physical address space, and can access a 4-Gbyte address space. When the AT bit in MMUCR is cleared to 0 and the MMU is disabled, the address space is this physical address space. The physical address space is divided into a number of areas, as shown in figure 6.2. The physical address space is permanently mapped onto a 29-bit external memory space; this correspondence can be implemented by ignoring the upper 3 bits of the physical address space addresses. In privileged mode, the 4-Gbyte space from the P0 area to the P4 area can be accessed. In user mode, a 2-Gbyte space in the U0 area can be accessed. Accessing the P1 to P4 areas (except the store queue area) in user mode will cause an address error.

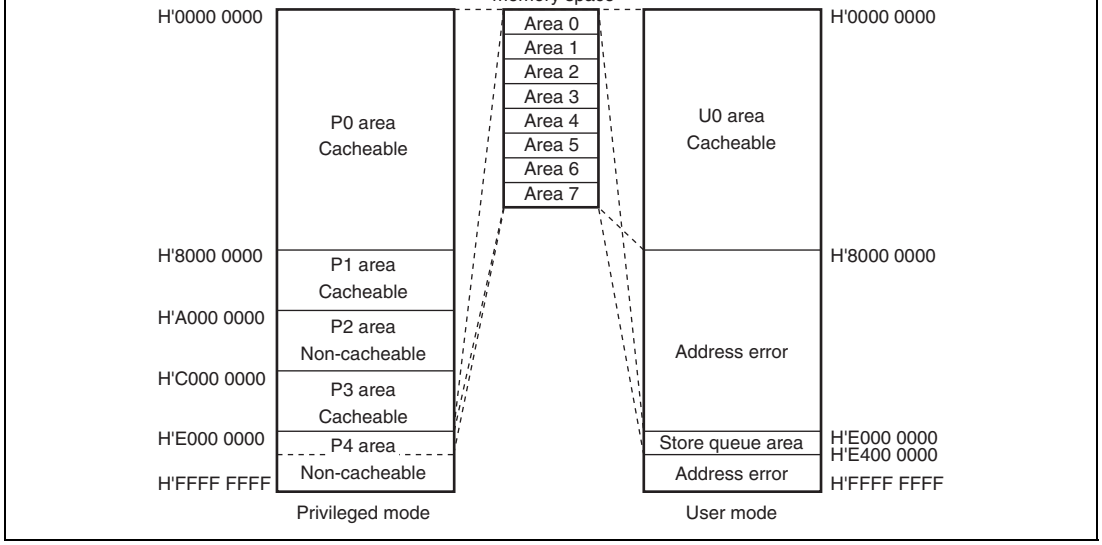


Figure 6.2 Physical Address Space (AT = 0 in MMUCR)

Access to a PCMCIA interface area by the CPU in SH7760 is always performed using the values of the SA and TC bits in PTEA.

Access to a PCMCIA interface area by the DMAC is always performed using the SSAn, DSAn, STC, and DTC values in CHCRn of the DMAC. For details, see section 11, Direct Memory Access Controller (DMAC).

P0, P1, P3, and U0 Areas: The P0, P1, P3, and U0 areas can be accessed using the cache. Whether or not the cache is used is determined by the CCR setting. When the cache is used, switching between the copy-back method and the write-through method for write accesses is specified by the WT bit in CCR, except for in the P1 area. Switching in the P1 area is determined by the CB bit in CCR. Replacing the upper 3 bits of an address in these areas with 0s gives the corresponding external memory space address. However, since area 7 in the external memory space is a reserved area, a reserved area will exist in these areas.

P2 Area: The P2 area cannot be accessed using the cache. In the P2 area, clearing the upper 3 bits of an address to 0 gives the corresponding external memory space address. However, since area 7 in the external memory space is a reserved area, a reserved area will exist in this area.

P4 Area: The P4 area is mapped onto SH-4 on-chip I/O memory. The P4 area cannot be accessed using the cache. The P4 area is shown in detail in figure 6.3.

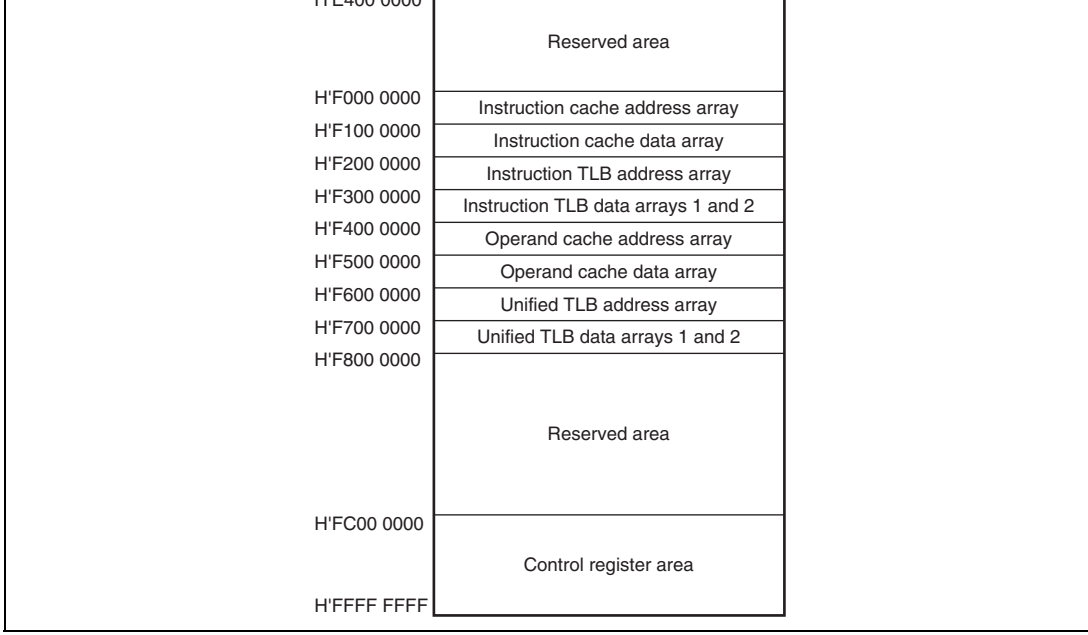


Figure 6.3 P4 Area

The area from H'E000 0000 to H'E3FF FFFF comprises addresses for accessing the store queues (SQs). When the MMU is disabled (AT bit in MMUCR = 0), the SQ access right is specified by the SQMD bit in MMUCR. For details, see section 7.7, Store Queues.

The area from H'F000 0000 to H'F0FF FFFF is used for direct access to the instruction cache address array. For details, see section 7.6.1, IC Address Array.

The area from H'F100 0000 to H'F1FF FFFF is used for direct access to the instruction cache data array. For details, see section 7.6.2, IC Data Array.

The area from H'F200 0000 to H'F2FF FFFF is used for direct access to the instruction TLB address array. For details, see section 6.6.1, ITLB Address Array.

The area from H'F300 0000 to H'F3FF FFFF is used for direct access to instruction TLB data arrays 1 and 2. For details, see section 6.6.2, ITLB Data Array 1, and section 6.6.3, ITLB Data Array 2.

The area from H'F400 0000 to H'F4FF FFFF is used for direct access to the operand cache address array. For details, see section 7.6.3, OC Address Array.

array. For details, see section 7.3.4, UC Data Array.

The area from H'F600 0000 to H'F6FF FFFF is used for direct access to the unified TLB address array. For details, see section 6.6.4, UTLB Address Array.

The area from H'F700 0000 to H'F7FF FFFF is used for direct access to unified TLB data arrays 1 and 2. For details, see section 6.6.5, UTLB Data Array 1, and section 6.6.6, UTLB Data Array 2.

The area from H'FC00 0000 to H'FFFF FFFF is the on-chip peripheral module control register area.

(2) External Memory Space

The SH-4 supports a 29-bit external memory space. The external memory space is divided into eight areas as shown in figure 6.4. Areas 0 to 6 relate to memory, such as SRAM, synchronous DRAM, and PCMCIA. Area 7 is a reserved area. For details, see section 10, Bus State Controller (BSC).

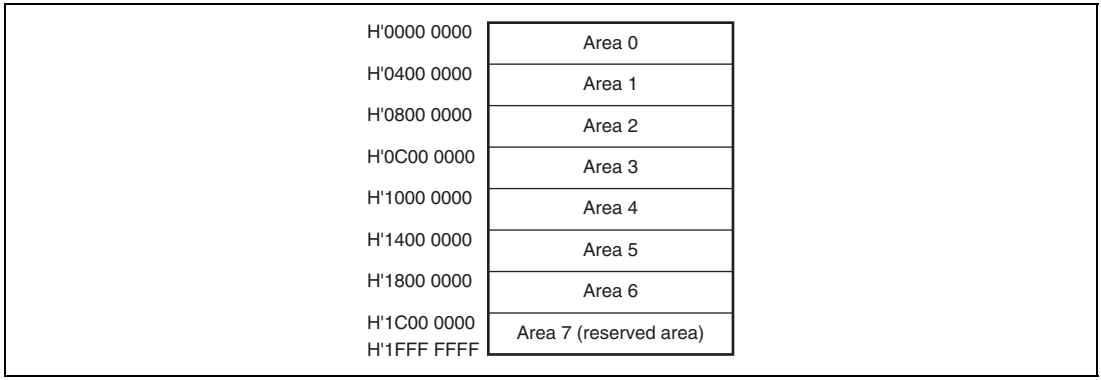


Figure 6.4 External Memory Space

(3) Virtual Address Space

Setting the AT bit in MMUCR to 1 enables the P0, P3, and U0 areas of the physical address space in the SH-4 to be mapped onto any external memory space in 1-, 4-, or 64-Kbyte, or 1-Mbyte page units. By using an 8-bit address space identifier, the P0, U0, P3, and store queue areas can be increased to a maximum of 256. This is called the virtual address space. Mapping from the virtual address space to the 29-bit external memory space is carried out using the TLB. Only when area 7 in the external memory space is accessed using the virtual address space, addresses H'1C00 0000 to H'1FFF FFFF of area 7 are not designated as a reserved area, but are equivalent to the P4 area control register area in the physical address space. The virtual address space is shown in figure 6.5.

cleared to 0 for that page. In this case, access to the area is performed using the SA and TC bit values specified in page units for each TLB page.

Note that the CPU cannot access a PCMCIA interface area through access of the P1, P2, or P4 area. Access to a PCMCIA interface area by the DMAC is always performed using the SSAn, DSAn, STC, and DTC values in CHCRn of the DMAC. For details, see section 11, Direct Memory Access Controller (DMAC).

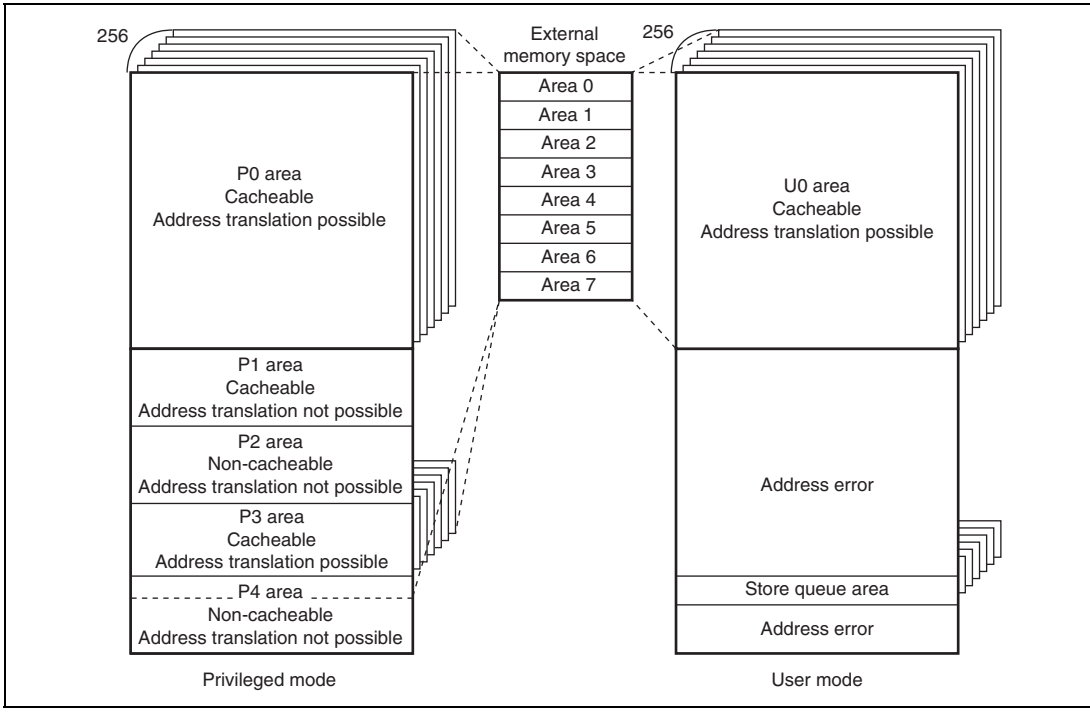


Figure 6.5 Virtual Address Space (AT = 1 in MMUCR)

P0, P3, and U0 Areas: The P0 area (excluding addresses H'7C00 0000 to H'7FFF FFFF), P3 area, and U0 area (excluding addresses H'7C00 0000 to H'7FFF FFFF) allow access using the cache and address translation using the TLB. These areas can be mapped onto any external memory space in 1-, 4-, or 64-Kbyte, or 1-Mbyte page units. When CCR is in the cache enabled state and the TLB cacheability bit (C bit) is 1, accesses can be performed using the cache. In write accesses to the cache, switching between the copy-back method and the write-through method is indicated by the TLB write-through bit (WT bit), and is specified in page units.

allocated to the control register area. This enables control registers to be accessed from the U0 area even in user mode. In this case, the C bit for the corresponding page must be cleared to 0.

P1, P2, and P4 Areas: Address translation using the TLB cannot be performed for the P1, P2, or P4 area (except for the store queue area). Accesses to these areas are the same as for the physical address space. The store queue area can be mapped onto any external memory space by the MMU. However, operation in the case of an exception differs from that for normal P0, U0, and P3 areas. For details, see section 7.7, Store Queues.

(4) On-Chip RAM Space

In the SH-4, half of the operand cache can be used as on-chip RAM. This can be done by changing the CCR settings.

When the operand cache is used as on-chip RAM (ORA bit in CCR = 1), addresses H'7C00 0000 to H'7FFF FFFF in the P0 and U0 areas are an on-chip RAM area. Data accesses (byte/word/longword/quadword) can be used in this area. This area can only be used in RAM mode.

(5) Address Translation

When the MMU is used, the virtual address space is divided into units called pages, and translation to physical addresses is carried out in these page units. The address translation table in external memory contains the physical addresses corresponding to virtual addresses and additional information such as memory protection codes. Fast address translation is achieved by caching the contents of the address translation table located in external memory into the TLB. In the SH-4, basically, the ITLB is used for instruction accesses and the UTLB for data accesses. In the event of an access to an area other than the P4 area, the accessed virtual address is translated to a physical address. If the virtual address belongs to the P1 or P2 area, the physical address is uniquely determined without accessing the TLB. If the virtual address belongs to the P0, U0, or P3 area, the TLB is searched using the virtual address, and if the virtual address is recorded in the TLB, a TLB hit is made and the corresponding physical address is read from the TLB. If the accessed virtual address is not recorded in the TLB, a TLB miss exception is generated and processing switches to the TLB miss exception handling routine. In the TLB miss exception handling routine, the address translation table in external memory is searched, and the corresponding physical address and page management information are recorded in the TLB. After the return from the exception handling routine, the instruction which caused the TLB miss exception is re-executed.

There are two virtual memory systems, single virtual memory and multiple virtual memory, either of which can be selected with the SV bit in MMUCR. In the single virtual memory system, a number of processes run simultaneously, using virtual address space on an exclusive basis, and the physical address corresponding to a particular virtual address is uniquely determined. In the multiple virtual memory system, a number of processes run while sharing the virtual address space, and particular virtual addresses may be translated into different physical addresses depending on the process. The only difference between the single virtual memory and multiple virtual memory systems in terms of operation is in the TLB address comparison method (see section 6.3.3, Address Translation Method).

(7) Address Space Identifier (ASID)

In multiple virtual memory mode, an 8-bit address space identifier (ASID) is used to distinguish between multiple processes running simultaneously while sharing the virtual address space. Software can set the 8-bit ASID of the currently executing process in PTEH in the MMU. The TLB does not have to be purged when processes are switched by means of ASID.

In single virtual memory mode, ASID is used to provide memory protection for multiple processes running simultaneously while using the virtual address space on an exclusive basis.

Note: Two or more entries with the same virtual page number (VPN) but different ASID must not be set in the TLB simultaneously as a single virtual memory mode setting.

6.2 Register Descriptions

The following registers are related to MMU processing. For details on the addresses of these registers and the state of registers in each operating mode, see section 32, List of Registers.

Table 6.1 Register Configuration (1)

Register Name	Abbrev.	R/W	P4 Address	Area 7 Address	Size	Sync Clock
Page table entry high register	PTEH	R/W	H'FF00 0000	H'1F00 0000	32	Ick
Page table entry low register	PTEL	R/W	H'FF00 0004	H'1F00 0004	32	Ick
Page table entry assistance register	PTEA	R/W	H'FF00 0034	H'1F00 0034	32	Ick
Translation table base register	TTB	R/W	H'FF00 0008	H'1F00 0008	32	Ick
TLB exception address register	TEA	R/W	H'FF00 000C	H'1F00 000C	32	Ick
MMU control register	MMUCR	R/W	H'FF00 0010	H'1F00 0010	32	Ick

Register Name	Abbrev.	Power-on Reset by RESET Pin/WDT/ H-UDI	Manual Reset by RESET Pin/WDT/ Multiple Exception	Sleep by Sleep Instruction/ Deep Sleep	Standby by Hardware	Standby by Software/ Each Module
Page table entry high register	PTEH	Undefined	Undefined	Retained	*	Retained
Page table entry low register	PTEL	Undefined	Undefined	Retained		Retained
Page table entry assistance register	PTEA	Undefined	Undefined	Retained		Retained
Translation table base register	TTB	Undefined	Undefined	Retained		Retained
TLB exception address register	TEA	Undefined	Retained	Retained		Retained
MMU control register	MMUCR	H'0000 0000	H'0000 0000	Retained		Retained

Note: * After exiting hardware standby mode, this LSI enters the power-on reset state caused by the RESET pin.

6.2.1 Page Table Entry High Register (PTEH)

PTEH can be accessed in longwords from H'FF00 0000 in the P4 area and from H'1F00 0000 in area 7. PTEH consists of the virtual page number (VPN) and address space identifier (ASID). When an MMU exception or address error exception occurs, the VPN of the virtual address at which the exception occurred is set in the VPN bit by hardware. VPN varies according to the page size, but the VPN set by hardware when an exception occurs consists of the upper 22 bits of the virtual address which caused the exception. VPN setting can also be carried out by software. The number of the currently executing process is set in the ASID bit by software. ASID is not updated by hardware. VPN and ASID are recorded in the UTLB by means of the LDLTB instruction.

After the ASID field in PTEH has been rewritten, a branch instruction to the P0, P3, or U0 area that uses the updated ASID value should be located at least six instructions after the PTEH update instruction.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	VPN															
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

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	VPN						-	-	ASID							
Initial value:	-	-	-	-	-	-	0	0	-	-	-	-	-	-	-	-
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

31 to 10	VPN	—	R/W	Virtual Page Number
9, 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7 to 0	ASID	—	R/W	Address Space Identifier

6.2.2 Page Table Entry Low Register (PTEL)

PTEL can be accessed in longwords from H'FF00 0004 in the P4 area and from H'1F00 0004 in area 7. PTEL is used to hold the physical page number and page management information to be recorded in the UTLB by means of the LDTLB instruction. The contents of this register are not changed unless a software directive is issued.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	PPN												
Initial value:	0	0	0	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W:	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	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PPN						-	V	SZ1	PR1	PR0	SZ0	C	D	SH	WT
Initial value:	-	-	-	-	-	-	0	-	-	-	-	-	-	-	-	-
R/W:	R/W	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

Bit	Bit Name	Initial Value	R/W	Description
31 to 29	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
28 to 10	PPN	—	R/W	Physical Page Number
9	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
8	V	—	R/W	Page Management Information For details, see section 6.3, TLB Functions.
7	SZ1	—	R/W	
6	PR1	—	R/W	
5	PR0			
4	SZ0	—	R/W	
3	C	—	R/W	
2	D	—	R/W	
1	SH	—	R/W	
0	WT	—	R/W	

PTEA can be accessed in longwords from H'FF00 0034 in the P4 area and from H'1F00 0034 in area 7. PTEA is used to store assistance bits for PCMCIA access to the UTLB by means of the LDTLB instruction. When performing access from the CPU in SH7760 to the PCMCIA interface area with the AT bit in MMUCR cleared to 0, access is always performed using the values of the SA and TC bits in this register. The access to a PCMCIA interface area by the DMAC is always performed using the SSAn, DSAn, STC, and DTC values in CHCRn of the DMAC. The contents of this register are not changed unless a software directive is issued.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
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
	-	-	-	-	-	-	-	-	-	-	-	-	TC	SA2	SA1	SA0
Initial value:	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/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
3	TC	—	R/W	PCMCIA Access Assistance Bits For details, see section 6.3, TLB Functions.
2	SA2	—	R/W	
1	SA1	—	R/W	
0	SA0	—	R/W	

6.2.4 Translation Table Base Register (TTB)

TTB can be accessed in longwords from H'FF00 0008 in the P4 area and from H'1F00 0008 in area 7. TTB is used, for example, to hold the base address of the currently used page table. The contents of TTB are not changed unless a software directive is issued. This register can be used freely by software.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TTB															
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

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TTB															
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

TEA can be accessed in longwords from H'FF00 000C in the P4 area and from H'1F00 000C in area 7. After an MMU exception or address error exception occurs, the virtual address at which the exception occurred is set in TEA by hardware. The contents of this register can be changed by software.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	Virtual address at which MMU exception or address error occurred															
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

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Virtual address at which MMU exception or address error occurred															
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

6.2.6 MMU Control Register (MMUCR)

MMUCR can be accessed in longwords from H'FF00 0010 in the P4 area and from H'1F00 0010 in area 7. The individual bits perform MMU settings as shown below. Therefore, MMUCR rewriting should be performed by a program in the P1 or P2 area. After MMUCR is updated, an instruction that performs data access to the P0, P3, U0, or store queue area should be located at least four instructions after the MMUCR update instruction. Also, a branch instruction to the P0, P3, or U0 area should be located at least eight instructions after the MMUCR update instruction. MMUCR contents can be changed by software. However, the LRUI bits and URC bits may also be updated by hardware.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	LRUI						-	-	URB						-	-
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/W	R/W	R/W	R/W	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	URC						SQMD	SV	-	-	-	-	-	TI	-	AT
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	R	R	R	R	R/W	R	R/W

31 to 26	LRUI	All 0	R/W	Least Recently Used ITLB These bits indicate the ITLB entry to be replaced. The LRU (least recently used) method is used to decide the ITLB entry to be replaced in the event of an ITLB miss. The entry to be purged from the ITLB can be confirmed using the LRUI bits. LRUI is updated by means of the algorithm shown below. x means that updating is not performed. 000xxx: ITLB entry 0 is used 1xx00x: ITLB entry 1 is used x1x1x0: ITLB entry 2 is used xx1x11: ITLB entry 3 is used xxxxxx: Other than above When the LRUI bit settings are as shown below, the corresponding ITLB entry is updated by an ITLB miss. Ensure that values for which "Setting prohibited" is indicated below are not set at the discretion of software. After a power-on or manual reset, the LRUI bits are initialized to 0, and therefore a prohibited setting is never made by a hardware update. x means "don't care". 111xxx: ITLB entry 0 is updated 0xx11x: ITLB entry 1 is updated x0x0x1: ITLB entry 2 is updated xx0x00: ITLB entry 3 is updated Other than above: Setting prohibited
25, 24	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
23 to 18	URB	All 0	R/W	UTLB Replace Boundary These bits indicate the UTLB entry boundary at which replacement is to be performed. Valid only when URB > 0.
17, 16	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

15 to 10	URC	All 0	R/W	UTLB Replace Counter These bits serve as a random counter for indicating the UTLB entry for which replacement is to be performed with an LDTLB instruction. This bit is incremented each time the UTLB is accessed. If URB > 0, URC is cleared to 0 when the condition URC = URB is satisfied. Also note that if a value is written to URC by software which results in the condition of URC > URB, incrementing is first performed in excess of URB until URC = H'3F. URC is not incremented by an LDTLB instruction.
9	SQMD	0	R/W	Store Queue Mode Bit Specifies the right of access to the store queues. 0: User/privileged access possible 1: Privileged access possible (address error exception in case of user access)
8	SV	0	R/W	Single Virtual Memory Mode/Multiple Virtual Memory Mode Switching Bit When this bit is changed, ensure that 1 is also written to the TI bit. 0: Multiple virtual memory mode 1: Single virtual memory mode
7 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2	TI	0	R/W	TLB Invalidate Bit Writing 1 to this bit invalidates (clears to 0) all valid UTLB/ITLB bits. This bit is always read as 0.
1	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
0	AT	0	R/W	Address Translation Enable Bit These bits enable or disable the MMU. 0: MMU disabled 1: MMU enabled MMU exceptions are not generated when the AT bit is 0. In the case of software that does not use the MMU, the AT bit should be cleared to 0.

6.3.1 Unified TLB (UTLB) Configuration

The UTLB is used for the following two purposes:

1. To translate a virtual address to a physical address in a data access
2. As a table of address translation information to be recorded in the ITLB in the event of an ITLB miss

The UTLB is so called because of its use for the above two purposes. Information in the address translation table located in external memory is cached into the UTLB. The address translation table contains virtual page numbers and address space identifiers, and corresponding physical page numbers and page management information. Figure 6.6 shows the UTLB configuration. The UTLB consists of 64 fully-associative type entries. Figure 6.7 shows the relationship between the page size and address format.

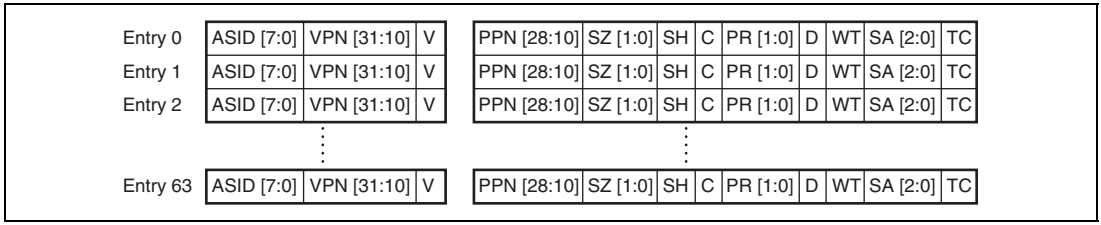


Figure 6.6 UTLB Configuration

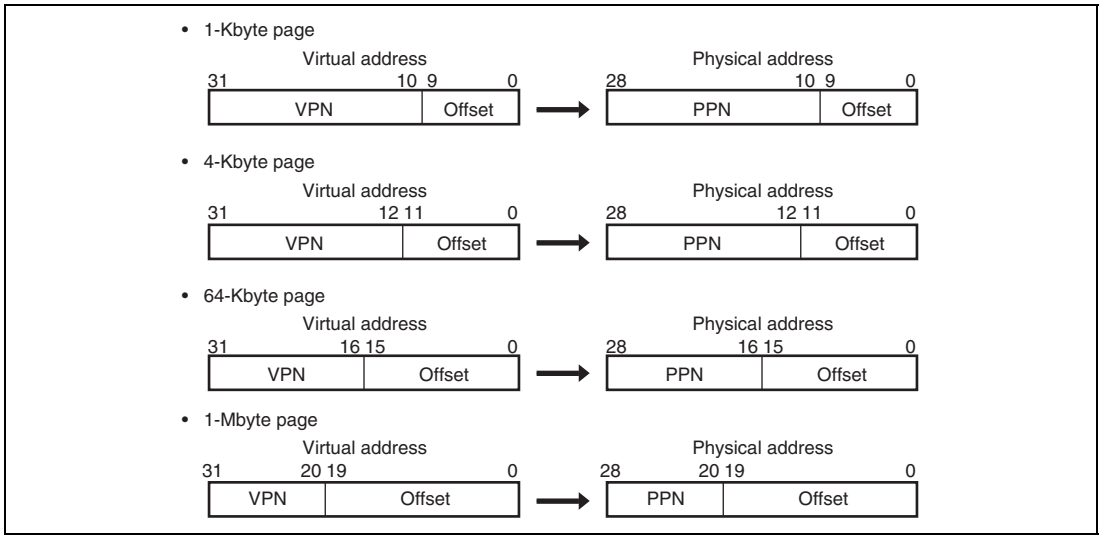


Figure 6.7 Relationship between Page Size and Address Format

- VPN: Virtual page number
  For 1-Kbyte page: Upper 22 bits of virtual address
  For 4-Kbyte page: Upper 20 bits of virtual address
  For 64-Kbyte page: Upper 16 bits of virtual address
  For 1-Mbyte page: Upper 12 bits of virtual address

- 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, this identifier is compared with the ASID in PTEH when address comparison is performed.

- SH: Share status bit
  When 0, pages are not shared by processes.
  When 1, pages are shared by processes.

- SZ[1:0]: Page size bits
  Specify the page size.
  00: 1-Kbyte page
  01: 4-Kbyte page
  10: 64-Kbyte page
  11: 1-Mbyte page

- V: Validity bit
  Indicates whether the 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
  Upper 22 bits of the physical address of the physical page number.
  With a 1-Kbyte page, PPN[28:10] are valid.
  With a 4-Kbyte page, PPN[28:12] are valid.
  With a 64-Kbyte page, PPN[28:16] are valid.
  With a 1-Mbyte page, PPN[28:20] are valid.

Avoiding Synonym Problems).

- PR[1:0]: Protection key data
  2-bit data expressing the page access right as a code.
  00: Can be read from only in privileged mode
  01: Can be read from and written to in privileged mode
  10: Can be read from only in privileged or user mode
  11: Can be read from and written to in privileged mode or user mode

- C: Cacheability bit
  Indicates whether a page is cacheable.
  0: Not cacheable
  1: Cacheable
  When the control register space is mapped, this bit must be cleared to 0.
  When performing PCMCIA space mapping in the cache enabled state, either clear this bit to 0 or set the WT bit to 1.

- D: Dirty bit
  Indicates whether a write has been performed to a page.
  0: Write has not been performed
  1: Write has been performed

- WT: Write-through bit
  Specifies the cache write mode.
  0: Copy-back mode
  1: Write-through mode
  When performing PCMCIA space mapping in the cache enabled state, either set this bit to 1 or clear the C bit to 0.

- SA[2:0]: Space attribute bits
  Valid only when the page is mapped onto PCMCIA connected to area 5 or 6.
  000: Undefined
  001: Variable-size I/O space (base size according to $\overline{\text{IOIS16}}$ signal)
  010: 8-bit I/O space
  011: 16-bit I/O space
  100: 8-bit shared memory space
  101: 16-bit shared memory space

111: 16-bit attribute memory space

- TC: Timing control bit

  Used to select wait control register bits in the bus control unit for areas 5 and 6.

  0: WCR2 (A5W2 to A5W0) and PCR (A5PCW1 to A5PCW0, A5TED2 to A5TED0, A5TEH2 to A5TEH0) are used

  1: WCR2 (A6W2 to A6W0) and PCR (A6PCW1 to A6PCW0, A6TED2 to A6TED0, A6TEH2 to A6TEH0) are used

6.3.2 Instruction TLB (ITLB) Configuration

The ITLB is used to translate a virtual address to a physical address in an instruction access. Information in the address translation table located in the UTLB is cached into the ITLB. Figure 6.8 shows the ITLB configuration. The ITLB consists of four fully-associative type entries.

Entry 0	ASID [7:0]	VPN [31:10]	V	PPN [28:10]	SZ [1:0]	SH	C	PR	SA [2:0]	TC
Entry 1	ASID [7:0]	VPN [31:10]	V	PPN [28:10]	SZ [1:0]	SH	C	PR	SA [2:0]	TC
Entry 2	ASID [7:0]	VPN [31:10]	V	PPN [28:10]	SZ [1:0]	SH	C	PR	SA [2:0]	TC
Entry 3	ASID [7:0]	VPN [31:10]	V	PPN [28:10]	SZ [1:0]	SH	C	PR	SA [2:0]	TC

Notes: 1. The D and WT bits are not supported.
2. There is only one PR bit, corresponding to the upper bit of the PR bits in the UTLB.

Figure 6.8 ITLB Configuration

Figure 6.9 shows a flowchart of a memory access using the UTLB.

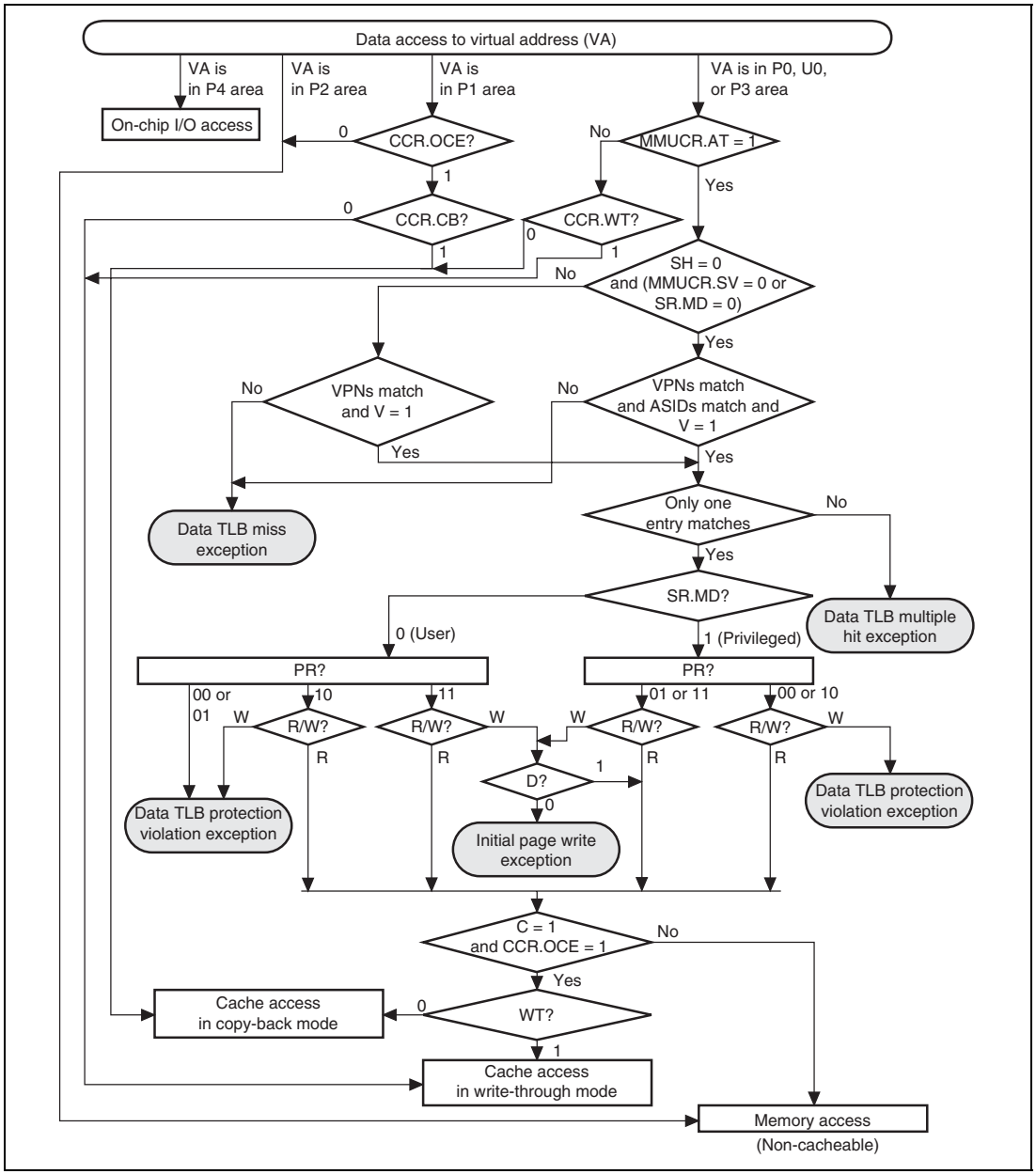


Figure 6.9 Flowchart of Memory Access Using UTLB

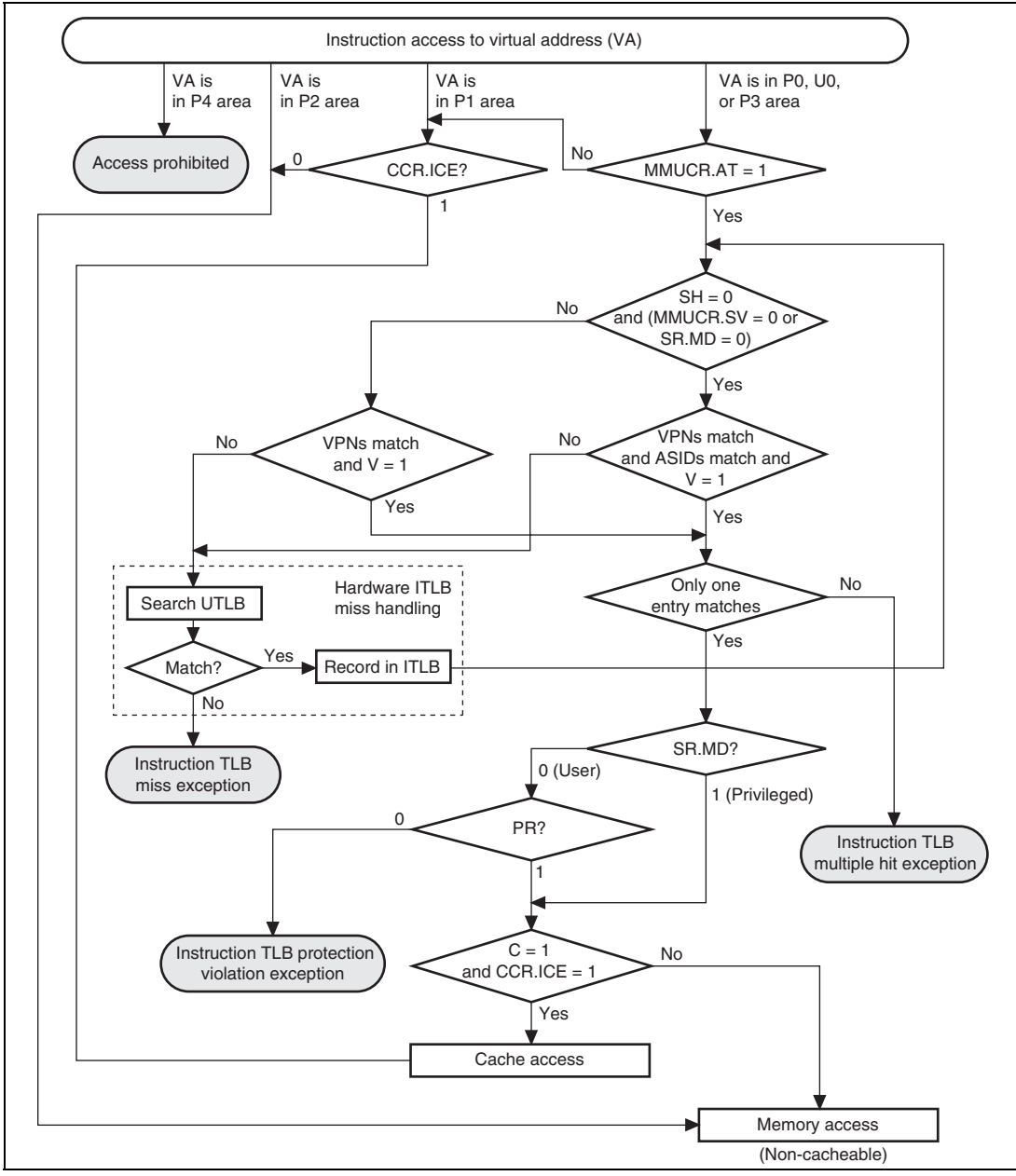


Figure 6.10 Flowchart of Memory Access Using ITLB

6.4.1 MMU Hardware Management

The SH-4 supports the following MMU functions.

1. The MMU decodes the virtual address to be accessed by software, and performs address translation by controlling the UTLB/ITLB in accordance with the MMUCR settings.
2. The MMU determines the cache access status on the basis of the page management information read during address translation (C, WT, SA, and TC bits).
3. If address translation cannot be performed normally in a data access or instruction access, the MMU notifies software by means of an MMU exception.
4. If address translation information is not recorded in the ITLB in an instruction access, the MMU searches the UTLB. If the necessary address translation information is recorded in the UTLB, the MMU copies this information into the ITLB in accordance with the LRUI bit setting in MMUCR.

6.4.2 MMU Software Management

Software processing for the MMU consists of the following:

1. Setting of MMU-related registers. Some registers are also partially updated by hardware automatically.
2. Recording, deletion, and reading of TLB entries. There are two methods of recording UTLB entries: by using the LDTLB instruction, or by writing directly to the memory-mapped UTLB. ITLB entries can only be recorded by writing directly to the memory-mapped ITLB. Deleting or reading UTLB/ITLB entries is enabled by accessing the memory-mapped UTLB/ITLB.
3. MMU exception handling. When an MMU exception occurs, processing is performed based on information set by hardware.

6.4.3 MMU Instruction (LDTLB)

A TLB load instruction (LDTLB) is provided for recording UTLB entries. When an LDTLB instruction is issued, the SH-4 copies the contents of PTEH, PTEL, and PTEA to the UTLB entry indicated by the URC bit in MMUCR. ITLB entries are not updated by the LDTLB instruction, and therefore address translation information purged from the UTLB entry may still remain in the ITLB entry. As the LDTLB instruction changes address translation information, ensure that it is issued by a program in the P1 or P2 area. The operation of the LDTLB instruction is shown in figure 6.11.

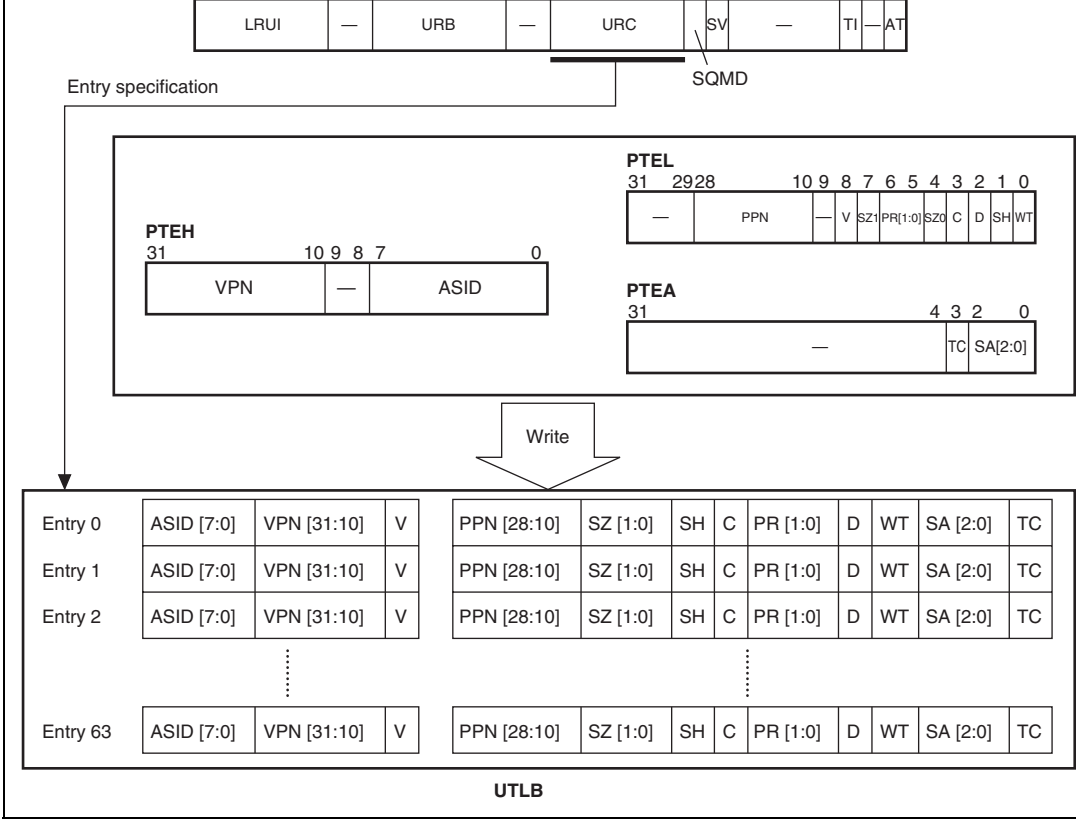


Figure 6.11 Operation of LDTLB Instruction

6.4.4 Hardware ITLB Miss Handling

In an instruction access, the SH-4 searches the ITLB. If it cannot find the necessary address translation information (ITLB miss occurred), the UTLB is searched by hardware, and if the necessary address translation information is present, it is recorded in the ITLB. This procedure is known as hardware ITLB miss handling. If the necessary address translation information is not found in the UTLB search, an instruction TLB miss exception is generated and processing passes to software.

When 1- or 4-Kbyte pages are recorded in TLB entries, a synonym problem may arise. The problem is that, when a number of virtual addresses are mapped onto a single physical address, the same physical address data is recorded in a number of cache entries, and it becomes impossible to guarantee data integrity. This problem does not occur with the instruction TLB and instruction cache because data is only read in these cases. In the SH-4, entry specification is performed using bits 13 to 5 of the virtual address in order to achieve fast operand cache operation. However, bits 13 to 10 of the virtual address in the case of a 1-Kbyte page, and bits 13 and 12 of the virtual address in the case of a 4-Kbyte page, are subject to address translation. As a result, bits 13 to 10 of the physical address after translation may differ from bits 13 of 10 of the virtual address.

Consequently, the following restrictions apply to the recording of address translation information in UTLB entries.

- When address translation information whereby a number of 1-Kbyte page UTLB entries are translated into the same physical address is recorded in the UTLB, ensure that the VPN[13:10] values are the same.
- When address translation information whereby a number of 4-Kbyte page UTLB entries are translated into the same physical address is recorded in the UTLB, ensure that the VPN[13:12] values are the same.
- Do not use 1-Kbyte page UTLB entry physical addresses with UTLB entries of a different page size.
- Do not use 4-Kbyte page UTLB entry physical addresses with UTLB entries of a different page size.

The above restrictions apply only when performing accesses using the cache. When cache index mode is used, VPN[25] is used for the entry address instead of VPN[13], and therefore the above restrictions apply to VPN[25].

Note: When multiple items of address translation information use the same physical memory to provide for future expansion of the SuperH RISC engine family, ensure that the VPN[20:10] values are the same. Also, do not use the same physical address for address translation information of different page sizes.

There are seven MMU exceptions: instruction TLB multiple hit exception, instruction TLB miss exception, instruction TLB protection violation exception, data TLB multiple hit exception, data TLB miss exception, data TLB protection violation exception, and initial page write exception. Refer to figures 6.9 and 6.10 for the conditions under which each of these exceptions occurs.

6.5.1 Instruction TLB Multiple Hit Exception

An instruction TLB multiple hit exception occurs when more than one ITLB entry matches the virtual address to which an instruction access has been made. If multiple hits occur when the UTLB is searched by hardware in hardware ITLB miss handling, a data TLB multiple hit exception will result.

When an instruction TLB multiple hit exception occurs, a reset is executed and cache coherency is not guaranteed.

Hardware Processing: In the event of an instruction TLB multiple hit exception, hardware carries out the following processing:

1. Sets the virtual address at which the exception occurred in TEA.
2. Sets exception code H'140 in EXPEVT.
3. Branches to the reset handling routine (H'A000 0000).

Software Processing (Reset Routine): The ITLB entries which caused the multiple hit exception are checked in the reset handling routine. This exception is intended for use in program debugging, and should not normally be generated.

6.5.2 Instruction TLB Miss Exception

An instruction TLB miss exception occurs when address translation information for the virtual address to which an instruction access is made is not found in the UTLB entries by the hardware ITLB miss handling routine. The instruction TLB miss exception processing carried out by hardware and software is shown below. This is the same as the processing for a data TLB miss exception.

Hardware Processing: In the event of an instruction TLB miss exception, hardware carries out the following processing:

1. Sets the VPN of the virtual address at which the exception occurred in PTEH.
2. Sets the virtual address at which the exception occurred in TEA.
3. Sets exception code H'040 in EXPEVT.

delayed branch instruction in SPC.

5. Sets the SR contents at the time of the exception in SSR. The R15 contents at this time are saved in SGR.
6. Sets the MD bit in SR to 1, and switches to privileged mode.
7. Sets the BL bit in SR to 1, and masks subsequent exception requests.
8. Sets the RB bit in SR to 1.
9. Branches to the address obtained by adding offset H'0000 0400 to the contents of VBR, and starts the instruction TLB miss exception handling routine.

Software Processing (Instruction TLB Miss Exception Handling Routine): Software is responsible for searching the external memory page table and assigning the necessary page table entry. Software should carry out the following processing in order to find and assign the necessary page table entry.

1. Write to PTEL the values of the PPN, PR, SZ, C, D, SH, V, and WT bits in the page table entry recorded in the external memory address translation table. If necessary, the values of the SA and TC bits should be written to PTEA.
2. When the entry to be replaced in entry replacement is specified by software, write that value to the URC bits in MMUCR. If URC is greater than URB at this time, the value should be changed to an appropriate value after issuing an LDTLB instruction.
3. Execute the LDTLB instruction and write the contents of PTEH, PTEL, and PTEA to the TLB.
4. Finally, execute the exception handling return instruction (RTE), terminate the exception handling routine, and return control to the normal flow. The RTE instruction should be issued at least one instruction after the LDTLB instruction.

6.5.3 Instruction TLB Protection Violation Exception

An instruction TLB protection violation exception occurs when, even though an ITLB entry contains address translation information matching the virtual address to which an instruction access is made, the actual access type is not permitted by the access right specified by the PR bit. The instruction TLB protection violation exception processing carried out by hardware and software is shown below.

Hardware Processing: In the event of an instruction TLB protection violation exception, hardware carries out the following processing:

1. Sets the VPN of the virtual address at which the exception occurred in PTEH.
2. Sets the virtual address at which the exception occurred in TEA.
3. Sets exception code H'0A0 in EXPEVT.

delayed branch instruction in SPC.

5. Sets the SR contents at the time of the exception in SSR. The R15 contents at this time are saved in SGR.
6. Sets the MD bit in SR to 1, and switches to privileged mode.
7. Sets the BL bit in SR to 1, and masks subsequent exception requests.
8. Sets the RB bit in SR to 1.
9. Branches to the address obtained by adding offset H'0000 0100 to the contents of VBR, and starts the instruction TLB protection violation exception handling routine.

Software Processing (Instruction TLB Protection Violation Exception Handling Routine): Resolve the instruction TLB protection violation, execute the exception handling return instruction (RTE), terminate the exception handling routine, and return control to the normal flow. The RTE instruction should be issued at least one instruction after the LDTLB instruction.

6.5.4 Data TLB Multiple Hit Exception

A data TLB multiple hit exception occurs when more than one UTLB entry matches the virtual address to which a data access has been made. A data TLB multiple hit exception is also generated if multiple hits occur when the UTLB is searched in hardware ITLB miss handling.

When a data TLB multiple hit exception occurs, a reset is executed, and cache coherency is not guaranteed. The contents of PPN in the UTLB prior to the exception may also be corrupted.

Hardware Processing: In the event of a data TLB multiple hit exception, hardware carries out the following processing:

1. Sets the virtual address at which the exception occurred in TEA.
2. Sets exception code H'140 in EXPEVT.
3. Branches to the reset handling routine (H'A000 0000).

Software Processing (Reset Routine): The UTLB entries which caused the multiple hit exception are checked in the reset handling routine. This exception is intended for use in program debugging, and should not normally be generated.

A data TLB miss exception occurs when address translation information for the virtual address to which a data access is made is not found in the UTLB entries. The data TLB miss exception processing carried out by hardware and software is shown below.

Hardware Processing: In the event of a data TLB miss exception, hardware carries out the following processing:

1. Sets the VPN of the virtual address at which the exception occurred in PTEH.
2. Sets the virtual address at which the exception occurred in TEA.
3. Sets exception code H'040 in the case of a read, or H'060 in the case of a write in EXPEVT (OCBP, OCBWB: read; OCBI, MOVCA.L: write).
4. Sets the PC value indicating the address of the instruction at which the exception occurred in SPC. If the exception occurred at a delay slot, sets the PC value indicating the address of the delayed branch instruction in SPC.
5. Sets the SR contents at the time of the exception in SSR. The R15 contents at this time are saved in SGR.
6. Sets the MD bit in SR to 1, and switches to privileged mode.
7. Sets the BL bit in SR to 1, and masks subsequent exception requests.
8. Sets the RB bit in SR to 1.
9. Branches to the address obtained by adding offset H'0000 0400 to the contents of VBR, and starts the data TLB miss exception handling routine.

Software Processing (Data TLB Miss Exception Handling Routine): Software is responsible for searching the external memory page table and assigning the necessary page table entry. Software should carry out the following processing in order to find and assign the necessary page table entry.

1. Write to PTEL the values of the PPN, PR, SZ, C, D, SH, V, and WT bits in the page table entry recorded in the external memory address translation table. If necessary, the values of the SA and TC bits should be written to PTEA.
2. When the entry to be replaced in entry replacement is specified by software, write that value to the URC bits in MMUCR. If URC is greater than URB at this time, the value should be changed to an appropriate value after issuing an LDTLB instruction.
3. Execute the LDTLB instruction and write the contents of PTEH, PTEL, and PTEA to the UTLB.
4. Finally, execute the exception handling return instruction (RTE), terminate the exception handling routine, and return control to the normal flow. The RTE instruction should be issued at least one instruction after the LDTLB instruction.

A data TLB protection violation exception occurs when, even though a UTLB entry contains address translation information matching the virtual address to which a data access is made, the actual access type is not permitted by the access right specified by the PR bit. The data TLB protection violation exception processing carried out by hardware and software is shown below.

Hardware Processing: In the event of a data TLB protection violation exception, hardware carries out the following processing:

1. Sets the VPN of the virtual address at which the exception occurred in PTEH.
2. Sets the virtual address at which the exception occurred in TEA.
3. Sets exception code H'0A0 in the case of a read, or H'0C0 in the case of a write in EXPEVT (OCBP, OCBWB: read; OCBI, MOVCA.L: write).
4. Sets the PC value indicating the address of the instruction at which the exception occurred in SPC. If the exception occurred at a delay slot, sets the PC value indicating the address of the delayed branch instruction in SPC.
5. Sets the SR contents at the time of the exception in SSR. The R15 contents at this time are saved in SGR.
6. Sets the MD bit in SR to 1, and switches to privileged mode.
7. Sets the BL bit in SR to 1, and masks subsequent exception requests.
8. Sets the RB bit in SR to 1.
9. Branches to the address obtained by adding offset H'0000 0100 to the contents of VBR, and starts the data TLB protection violation exception handling routine.

Software Processing (Data TLB Protection Violation Exception Handling Routine): Resolve the data TLB protection violation, execute the exception handling return instruction (RTE), terminate the exception handling routine, and return control to the normal flow. The RTE instruction should be issued at least one instruction after the LDTLB instruction.

An initial page write exception occurs when the D bit is 0 even though a UTLB entry contains address translation information matching the virtual address to which a data access (write) is made, and the access is permitted. The initial page write exception processing carried out by hardware and software is shown below.

Hardware Processing: In the event of an initial page write exception, hardware carries out the following processing:

1. Sets the VPN of the virtual address at which the exception occurred in PTEH.
2. Sets the virtual address at which the exception occurred in TEA.
3. Sets exception code H'080 in EXPEVT.
4. Sets the PC value indicating the address of the instruction at which the exception occurred in SPC. If the exception occurred at a delay slot, sets the PC value indicating the address of the delayed branch instruction in SPC.
5. Sets the SR contents at the time of the exception in SSR. The R15 contents at this time are saved in SGR.
6. Sets the MD bit in SR to 1, and switches to privileged mode.
7. Sets the BL bit in SR to 1, and masks subsequent exception requests.
8. Sets the RB bit in SR to 1.
9. Branches to the address obtained by adding offset H'0000 0100 to the contents of VBR, and starts the initial page write exception handling routine.

Software Processing (Initial Page Write Exception Handling Routine): Software is responsible for the following processing:

1. Retrieve the necessary page table entry from external memory.
2. Write 1 to the D bit in the external memory page table entry.
3. Write to PTEL the values of the PPN, PR, SZ, C, D, WT, SH, and V bits in the page table entry recorded in external memory. If necessary, the values of the SA and TC bits should be written to PTEA.
4. When the entry to be replaced in entry replacement is specified by software, write that value to the URC bits in MMUCR. If URC is greater than URB at this time, the value should be changed to an appropriate value after issuing an LDTLB instruction.
5. Execute the LDTLB instruction and write the contents of PTEH, PTEL, and PTEA to the UTLB.
6. Finally, execute the exception handling return instruction (RTE), terminate the exception handling routine, and return control to the normal flow. The RTE instruction should be issued at least one instruction after the LDTLB instruction.

To enable the ITLB and UTLB to be managed by software, their contents are allowed to be read from and written to by a P2 area program with a MOV instruction in privileged mode. Operation is not guaranteed if access is made from a program in another area. A branch to an area other than the P2 area should be made at least eight instructions after this MOV instruction. The ITLB and UTLB are allocated to the P4 area in the physical address space. VPN, V, and ASID in the ITLB can be accessed as an address array, PPN, V, SZ, PR, C, and SH as data array 1, and SA and TC as data array 2.

VPN, D, V, and ASID in the UTLB can be accessed as an address array, PPN, V, SZ, PR, C, D, WT, and SH as data array 1, and SA and TC as data array 2. V and D can be accessed from both the address array side and the data array side. Only longword access is possible. Instruction fetches cannot be performed in these areas. For reserved bits, a write value of 0 should be specified; their read value is undefined.

6.6.1 ITLB Address Array

The ITLB address array is allocated to addresses H'F200 0000 to H'F2FF FFFF in the P4 area. An address array access requires a 32-bit address field specification (when reading or writing) and a 32-bit data field specification (when writing). Information for selecting the entry to be accessed is specified in the address field, and VPN, V, and ASID to be written to the address array are specified in the data field.

In the address field, bits [31:24] have the value H'F2 indicating the ITLB address array and the entry is specified by bits [9:8]. As only longword access is used, 0 should be specified for address field bits [1:0].

In the data field, bits [31:10] indicate VPN, bit [8] indicates V, and bits [7:0] indicate ASID.

The following two kinds of operation can be used on the ITLB address array:

1. ITLB address array read

   VPN, V, and ASID are read into the data field from the ITLB entry corresponding to the entry set in the address field.
2. ITLB address array write

   VPN, V, and ASID specified in the data field are written to the ITLB entry corresponding to the entry set in the address field.

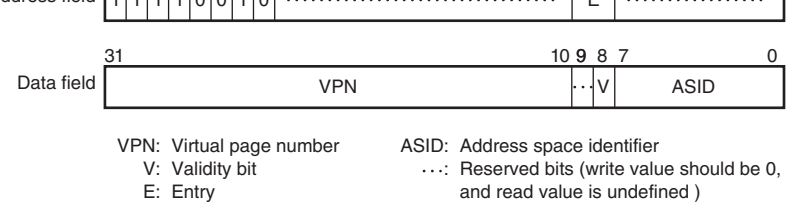


Figure 6.12 Memory-Mapped ITLB Address Array

6.6.2 ITLB Data Array 1

ITLB data array 1 is allocated to addresses H'F300 0000 to H'F37F FFFF in the P4 area. A data array access requires a 32-bit address field specification (when reading or writing) and a 32-bit data field specification (when writing). Information for selecting the entry to be accessed is specified in the address field, and PPN, V, SZ, PR, C, and SH to be written to the data array are specified in the data field.

In the address field, bits [31:23] have the value H'F30 indicating ITLB data array 1 and the entry is specified by bits [9:8].

In the data field, bits [28:10] indicate PPN, bit [8] indicates V, bits [7] and [4] indicate SZ, bit [6] indicates PR, bit [3] indicates C, and bit [1] indicates SH.

The following two kinds of operation can be used on ITLB data array 1:

1. ITLB data array 1 read

   PPN, V, SZ, PR, C, and SH are read into the data field from the ITLB entry corresponding to the entry set in the address field.
2. ITLB data array 1 write

   PPN, V, SZ, PR, C, and SH specified in the data field are written to the ITLB entry corresponding to the entry set in the address field.

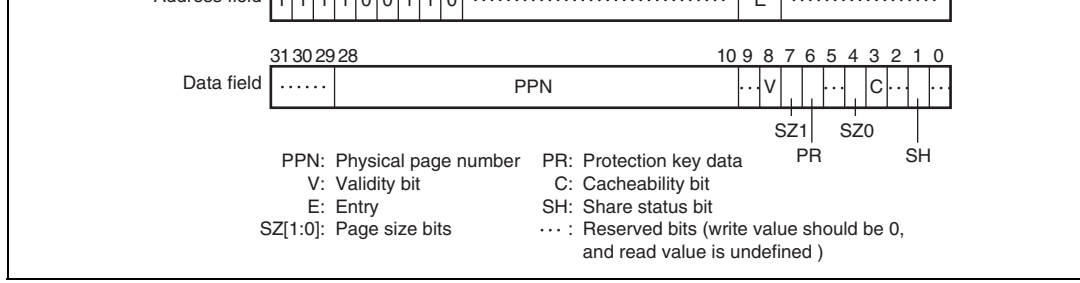


Figure 6.13 Memory-Mapped ITLB Data Array 1

6.6.3 ITLB Data Array 2

ITLB data array 2 is allocated to addresses H'F380 0000 to H'F3FF FFFF in the P4 area. A data array access requires a 32-bit address field specification (when reading or writing) and a 32-bit data field specification (when writing). Information for selecting the entry to be accessed is specified in the address field, and SA and TC to be written to data array 2 are specified in the data field.

In the address field, bits [31:23] have the value H'F38 indicating ITLB data array 2 and the entry is specified by bits [9:8].

In the data field, bit [3] indicates TC and bits [2:0] indicate SA.

The following two kinds of operation can be used on ITLB data array 2:

1. ITLB data array 2 read

   SA and TC are read into the data field from the ITLB entry corresponding to the entry set in the data field.
2. ITLB data array 2 write

   SA and TC specified in the data field are written to the ITLB entry corresponding to the entry set in the address field.

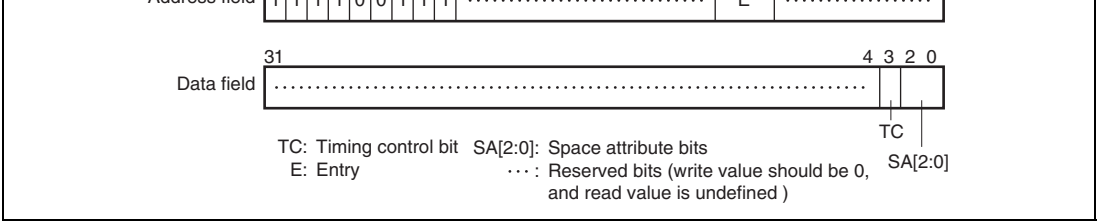


Figure 6.14 Memory-Mapped ITLB Data Array 2

6.6.4 UTLB Address Array

The UTLB address array is allocated to addresses H'F600 0000 to H'F6FF FFFF in the P4 area. An address array access requires a 32-bit address field specification (when reading or writing) and a 32-bit data field specification (when writing). Information for selecting the entry to be accessed is specified in the address field, and VPN, D, V, and ASID to be written to the address array are specified in the data field.

In the address field, bits [31:24] have the value H'F6 indicating the UTLB address array and the entry is specified by bits [13:8]. Bit [7] that is the association bit (A bit) in the address field specifies whether address comparison is performed in a write to the UTLB address array.

In the data field, bits [31:10] indicate VPN, bit [9] indicates D, bit [8] indicates V, and bits [7:0] indicate ASID.

The following three kinds of operation can be used on the UTLB address array:

1. UTLB address array read

   VPN, D, V, and ASID are read into the data field from the UTLB entry corresponding to the entry set in the address field. In a read, associative operation is not performed regardless of whether the association bit specified in the address field is 1 or 0.
2. UTLB address array write (non-associative)

   VPN, D, V, and ASID specified in the data field are written to the UTLB entry corresponding to the entry set in the address field. The A bit in the address field should be cleared to 0.
3. UTLB address array write (associative)

usual address comparison rules are followed, but if a UTLB miss occurs, the result is no operation, and an exception is not generated. If the comparison identifies a UTLB entry corresponding to the VPN specified in the data field, D and V specified in the data field are written to that entry. If there is more than one matching entry, a data TLB multiple hit exception occurs. This associative operation is simultaneously carried out on the ITLB, and if a matching entry is found in the ITLB, V is written to that entry. Even if the UTLB comparison results in no operation, a write to the ITLB side only is performed as long as there is an ITLB match. If there is a match in both the UTLB and ITLB, the UTLB information is also written to the ITLB.

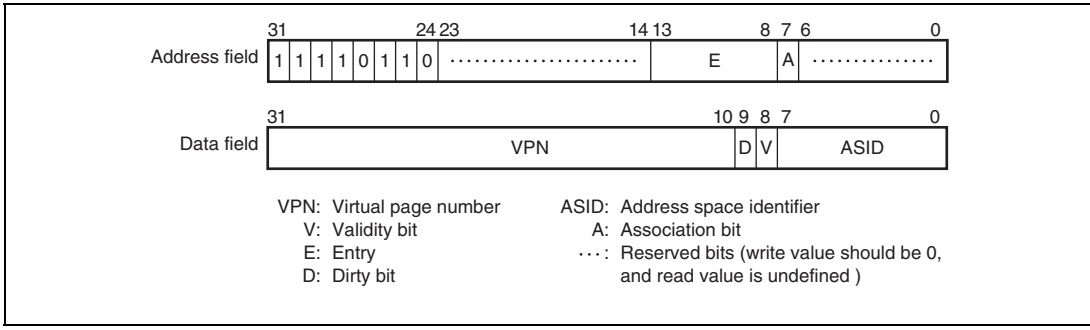


Figure 6.15 Memory-Mapped UTLB Address Array

6.6.5 UTLB Data Array 1

UTLB data array 1 is allocated to addresses H'F700 0000 to H'F77F FFFF in the P4 area. A data array access requires a 32-bit address field specification (when reading or writing) and a 32-bit data field specification (when writing). Information for selecting the entry to be accessed is specified in the address field, and PPN, V, SZ, PR, C, D, SH, and WT to be written to data array 1 are specified in the data field.

In the address field, bits [31:23] have the value H'F70 indicating UTLB data array 1 and the entry is specified by bits [13:8].

In the data field, bits [28:10] indicate PPN, bit [8] indicates V, bits [7] and [4] indicate SZ, bits [6:5] indicate PR, bit [3] indicates C, bit [2] indicates D, bit [1] indicates SH, and bit [0] indicates WT.

1. UTLB data array 1 read

   PPN, V, SZ, PR, C, D, SH, and WT are read into the data field from the UTLB entry corresponding to the entry set in the address field.

2. UTLB data array 1 write

   PPN, V, SZ, PR, C, D, SH, and WT specified in the data field are written to the UTLB entry corresponding to the entry set in the address field.

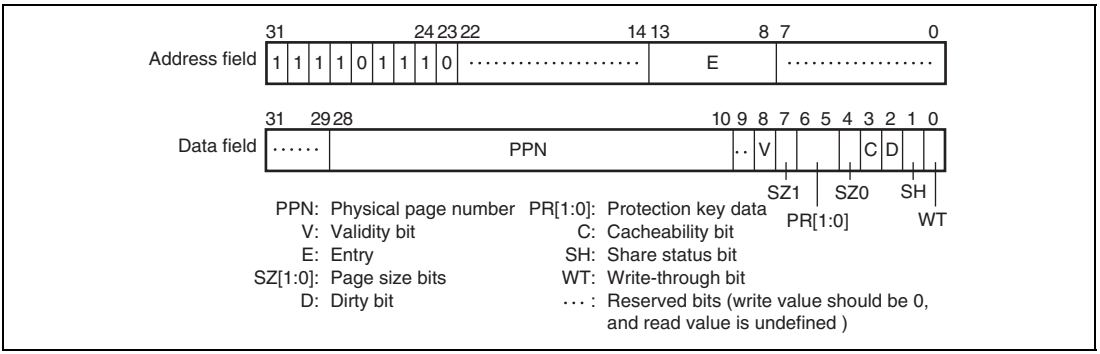


Figure 6.16 Memory-Mapped UTLB Data Array 1

6.6.6 UTLB Data Array 2

UTLB data array 2 is allocated to addresses H'F780 0000 to H'F7FF FFFF in the P4 area. A data array access requires a 32-bit address field specification (when reading or writing) and a 32-bit data field specification (when writing). Information for selecting the entry to be accessed is specified in the address field, and SA and TC to be written to data array 2 are specified in the data field.

In the address field, bits [31:23] have the value H'F78 indicating UTLB data array 2 and the entry is specified by bits [13:8].

In the data field, bit [3] indicates TC and bits [2:0] indicate SA.

The following two kinds of operation can be used on UTLB data array 2:

1. UTLB data array 2 read

   SA and TC are read into the data field from the UTLB entry corresponding to the entry set in the address field.

SA and TC specified in the data field are written to the UTLB entry corresponding to the entry set in the address field.

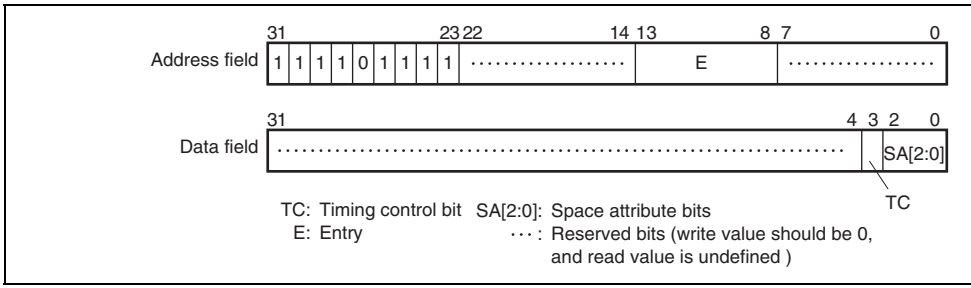


Figure 6.17 Memory-Mapped UTLB Data Array 2

7.1 Features

This LSI has an on-chip 16-Kbyte instruction cache (IC) for instructions and an on-chip 32-Kbyte operand cache (OC) for data. Half of the memory of the operand cache (16 Kbytes) may alternatively be used as on-chip RAM. When the EMODE bit in CCR is 0, this LSI's cache behaves as shown in table 7.1. The features of the cache when the EMODE bit in CCR is 1 are given in table 7.2. The EMODE bit is initialized to 0 after a power-on reset or manual reset.

This LSI supports two 32-byte store queues (SQs) to perform high-speed writes to external memory. The features of the store queues are given in table 7.3.

Table 7.1 Cache Features (EMODE = 0)

Item	Instruction Cache	Operand Cache
Capacity	8-Kbyte cache	16-Kbyte cache or 8-Kbyte cache + 8-Kbyte RAM
Type	Direct mapping	Direct mapping
Line size	32 bytes	32 bytes
Entries	256 entires	512 entires
Write method	—	Copy-back/write-through selectable

Table 7.2 Cache Features (EMODE = 1)

Item	Instruction Cache	Operand Cache
Capacity	16-Kbyte cache	32-Kbyte cache or 16-Kbyte cache + 16-Kbyte RAM
Type	2-way set-associative	2-way set-associative
Line size	32 bytes	32 bytes
Entries	256 entries/way	512 entries/way
Write method	—	Copy-back/write-through selectable
Replacement method	LRU (least-recently-used) algorithm	LRU (least-recently-used) algorithm

Item	Store Queues
Capacity	32 bytes × 2
Addresses	H'E000 0000 to H'E3FF FFFF
Write	Store instruction (1-cycle write)
Write-back	Prefetch instruction (PREF instruction)
Access right	MMU is off: Determined by SQMD bit in MMUCR MMU is on: Determined by PR for each page

of 512 cache lines. Figure 7.1 shows the configuration of the operand cache.

The instruction cache uses the 2-way set-associative method, and each way is configured of 256 cache lines. Figure 7.2 shows the configuration of the instruction cache.

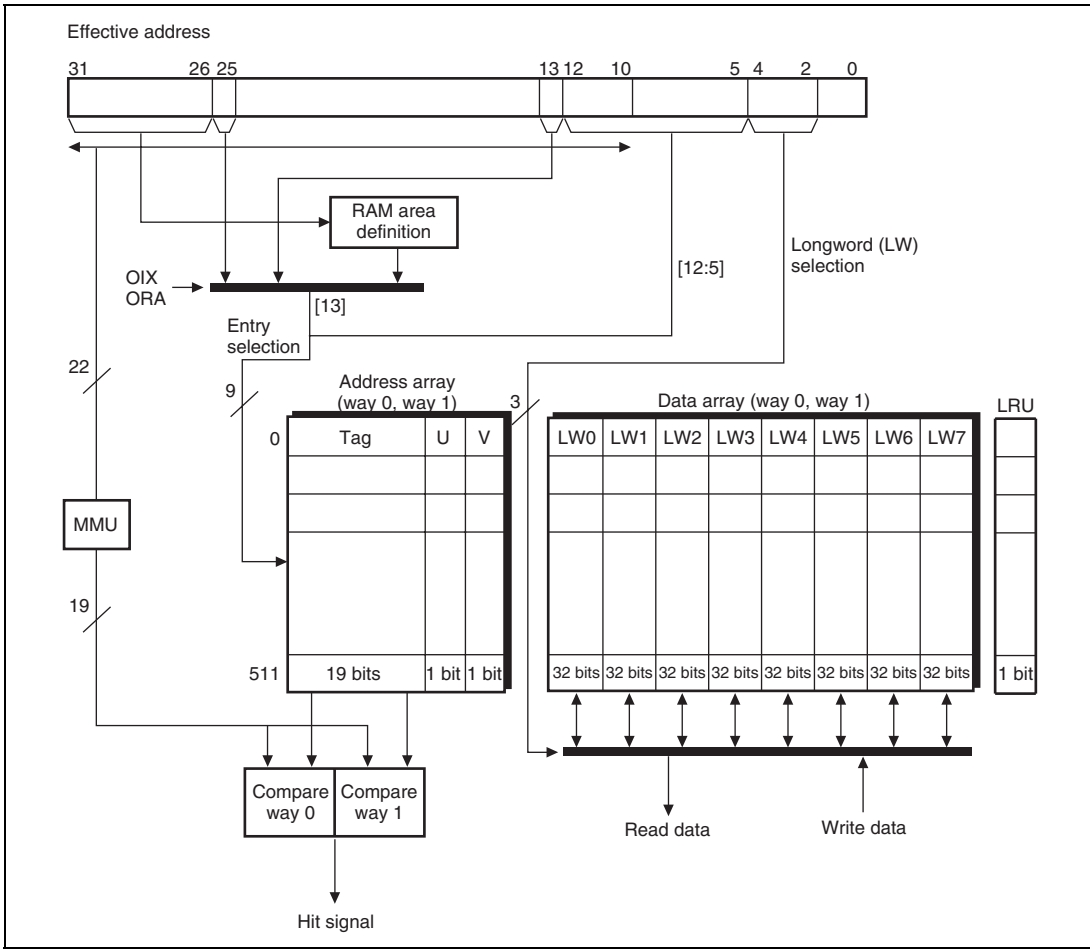


Figure 7.1 Configuration of Operand Cache

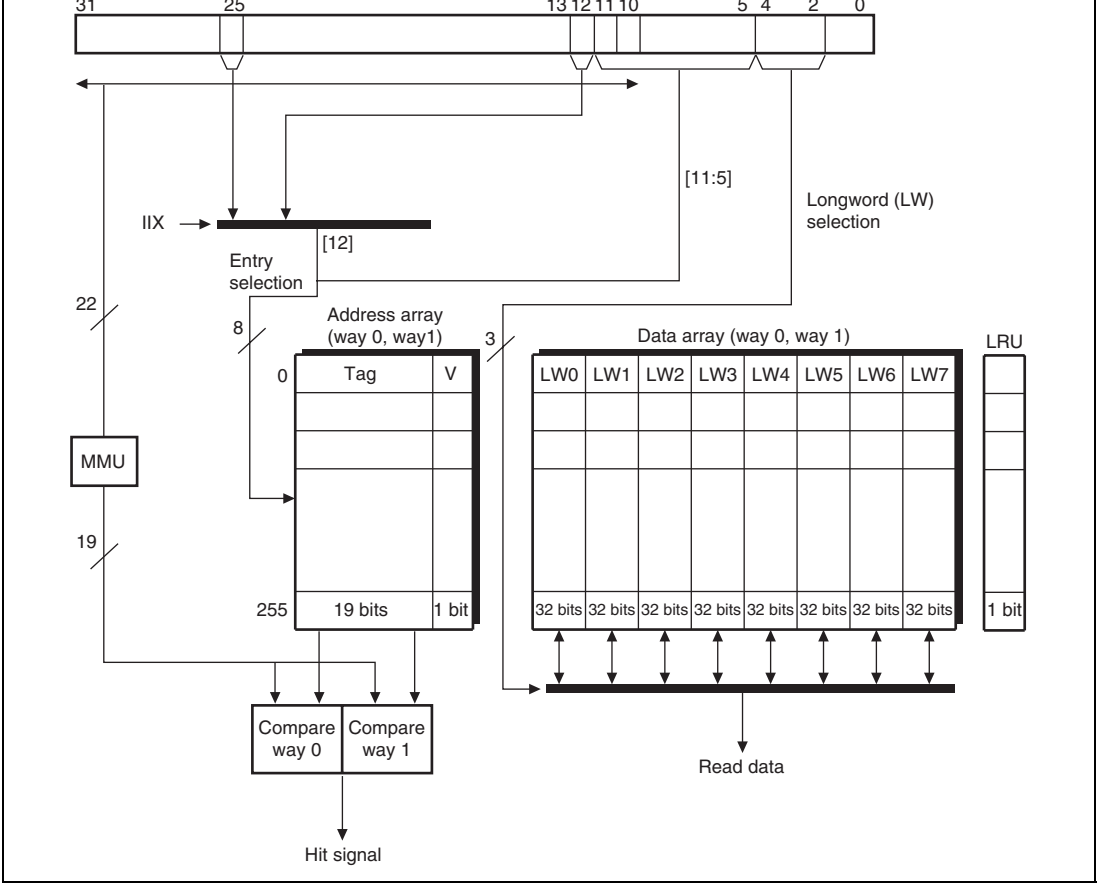


Figure 7.2 Configuration of Instruction Cache

- Tag

  Stores the upper 19 bits of the 29-bit external address of the data line to be cached. The tag is not initialized by a power-on or manual reset.

- V bit (validity bit)

  Indicates that valid data is stored in the cache line. When this bit is 1, the cache line data is valid. The V bit is initialized to 0 by a power-on reset, but retains its value in a manual reset.

- U bit (dirty bit)

  The U bit is set to 1 if data is written to the cache line while the cache is being used in copy-back mode. That is, the U bit indicates a mismatch between the data in the cache line and the data in external memory. The U bit is never set to 1 while the cache is being used in write-through mode, unless it is modified by accessing the memory-mapped cache (see section 7.5,

- Data array

  The data field holds 32 bytes (256 bits) of data per cache line. The data array is not initialized by a power-on or manual reset.

- LRU

  In a 2-way set-associative method, up to 2 items of data can be registered in the cache at each entry address. When an entry is registered, the LRU bit indicates which of the 2 ways it is to be registered in. The LRU bit is a single bit of each entry, and its usage is controlled by hardware. The LRU (least-recently-used) algorithm is used for way selection, and selects the less recently accessed way. The LRU bits are initialized to 0 by a power-on reset but not by a manual reset. The LRU bits cannot be read from or written to by software.

7.2 Register Descriptions

The following registers are related to cache. For details on the addresses of these registers and the state of registers in each operating mode, see section 32, List of Registers.

Table 7.4 Register Configuration (1)

Register Name	Abbrev.	R/W	P4 Address	Area 7 Address	Size	Sync Clock
Cache control register	CCR	R/W	H'FF00 001C	H'1F00 001C	32	Ick
Queue address control register 0	QACR0	R/W	H'FF00 0038	H'1F00 0038	32	Ick
Queue address control register 1	QACR1	R/W	H'FF00 003C	H'1F00 003C	32	Ick

Table 7.4 Register Configuration (2)

					Standby	
Register Name	Abbrev.	Power-on Reset by $\overline{\text{RESET}}$ Pin/WDT/H-UDI	Manual Reset by $\overline{\text{RESET}}$ Pin/WDT/ Multiple Exception	Sleep by Sleep Instruction/ Deep Sleep	by Hardware	by Software/ Each Module
Cache control register	CCR	H'0000 0000	H'0000 0000	Retained	*	Retained
Queue address control register 0	QACR0	Undefined	Undefined	Retained		Retained
Queue address control register 1	QACR1	Undefined	Undefined	Retained		Retained

Note: * After exiting hardware standby mode, this LSI enters the power-on reset state caused by the $\overline{\text{RESET}}$ pin.

CCR selects the cache operating mode, whether all cache entries are disabled, and the cache write mode.

CCR can be accessed in longwords from H'FF00 001C in the P4 area and from H'1F00 001C in area 7. CCR modifications must only be made by a program in the non-cached P2 area. After CCR is updated, an instruction that performs data access to the P0, P1, P3, or U0 area should be located at least four instructions after the CCR update instruction. Also, a branch instruction to the P0, P1, P3, or U0 area should be located at least eight instructions after the CCR update instruction.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	EMODE	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	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
	IIX	-	-	-	ICI	-	-	ICE	OIX	-	ORA	-	OCI	CB	WT	OCE
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R	R	R	R/W	R	R	R/W	R/W	R	R/W	R	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31	EMODE	0	R/W	Double-Size Cache Mode Bit This bit selects whether double-size cache mode is used or not. Do not write to this bit while cache is being used. 0: Cache direct mapping mode 1: Double-size cache mode
30 to 16	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
15	IIX	0	R/W	IC Index Enable Bit 0: Effective address bits [12:5] used for IC entry selection 1: Effective address bits [25] and [11:5] used for IC entry selection
14 to 12	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
11	ICI	0	R/W	IC Invalidation Bit When 1 is written to this bit, the V bits of all IC entries are cleared to 0. This bit is always read as 0.

10, 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	ICE	0	R/W	IC Enable Bit Selects whether the IC is used. Note however when address translation is performed, the IC cannot be used unless the C bit in the page management information is also 1. 0: IC not used 1: IC used
7	OIX	0	R/W	OC Index Enable Bit 0: Effective address bits [13:5] used for OC entry selection 1: Effective address bits [25] and [12:5] used for OC entry selection When the ORA bit is 1, this bit should be cleared to 0.
6	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
5	ORA	0	R/W	OCRAM Bit When the OC is enabled (OCE = 1), this bit specifies whether half of the OC is to be used as RAM. When the OC is disabled (OCE = 0), this bit should be cleared to 0. 0: Normal mode (the entire OC is used as a cache) 1: RAM mode (half of the OC is used as a cache and the other half is used as RAM) When the OIX bit is 1, this bit should be cleared to 0.
4	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
3	OCI	0	R/W	OC Invalidation Bit When 1 is written to this bit, the V and U bits of all OC entries are cleared to 0. This bit is always read as 0.
2	CB	0	R/W	Copy-Back Bit Indicates the P1 area cache write mode. 0: Write-through mode 1: Copy-back mode

1	WT	0	R/W	Write-Through Mode Indicates the P0, U0, and P3 area cache write mode. When address translation is performed, the value of the WT bit in the page management information has priority. 0: Copy-back mode 1: Write-through mode
0	OCE	0	R/W	OC Enable Bit Selects whether the OC is used. Note however when address translation is performed, the OC cannot be used unless the C bit in the page management information is also 1. 0: OC not used 1: OC used

7.2.2 Queue Address Control Register 0 (QACR0)

QACR0 can be accessed in longwords from H'FF00 0038 in the P4 area and from H'1F00 0038 in area 7. QACR0 specifies the area onto which store queue 0 (SQ0) is mapped when the MMU is off.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
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
	-	-	-	-	-	-	-	-	-	-	-	AREA0			-	-
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W:	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 5	—	—	R	Reserved These bits are always read as 0. The write value should always be 0.
4 to 2	AREA0	—	R/W	When the MMU is off, these bits generate external address bits [28:26] for SQ0.
1, 0	—	—	R	Reserved These bits are always read as 0. The write value should always be 0.

QACR1 can be accessed in longwords from H'FF00 003C in the P4 area and from H'1F00 003C in area 7. QACR1 specifies the area onto which store queue 1 (SQ1) is mapped when the MMU is off.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
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
	-	-	-	-	-	-	-	-	-	-	-	AREA1			-	-
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W:	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 5	—	—	R	Reserved These bits are always read as 0. The write value should always be 0.
4 to 2	AREA1	—	R/W	When the MMU is off, these bits generate external address bits [28:26] for SQ1.
1, 0	—	—	R	Reserved These bits are always read as 0. The write value should always be 0.

7.3 Operand Cache Operation

7.3.1 Read Operation

When the OC is enabled (OCE = 1 in CCR) and data is read by means of an effective address from a cacheable area, the cache operates as follows:

1. The tag, V bit, and U bit are read from the cache line indexed by effective address bits [13:5].
2. The tag is compared with bits [28:10] of the address resulting from effective address translation by the MMU:
   - If the tag matches and the V bit is 1 → 3.
   - If the tag matches and the V bit is 0 → 4.
   - If the tag does not match and the V bit is 0 → 4.
   - If the tag does not match, the V bit is 1, and the U bit is 0 → 4.
   - If the tag does not match, the V bit is 1, and the U bit is 1 → 5.

The data indexed by effective address bits [4:0] is read from the data field of the cache line indexed by effective address bits [13:5] in accordance with the access size (quadword/longword/word/byte).

4. Cache miss (no write-back)

   Data is read into the cache line from the external memory space corresponding to the effective address. Data reading is performed, using the wraparound method, in order from the longword data corresponding to the effective address, and when the corresponding data arrives in the cache, the read data is returned to the CPU. While the remaining one cache line of data is being read, the CPU can execute the next processing. When reading of one line of data is completed, the tag corresponding to the effective address is recorded in the cache, and 1 is written to the V bit.

5. Cache miss (with write-back)

   The tag and data field of the cache line indexed by effective address bits [13:5] are saved in the write-back buffer. Then data is read into the cache line from the external memory space corresponding to the effective address. Data reading is performed, using the wraparound method, in order from the longword data corresponding to the effective address, and when the corresponding data arrives in the cache, the read data is returned to the CPU. While the remaining one cache line of data is being read, the CPU can execute the next processing. When reading of one line of data is completed, the tag corresponding to the effective address is recorded in the cache, 1 is written to the V bit, and 0 to the U bit. The data in the write-back buffer is then written back to external memory.

7.3.2 Write Operation

When the OC is enabled (OCE = 1 in CCR) and data is written by means of an effective address to a cacheable area, the cache operates as follows:

1. The tag, V bit, and U bit are read from the cache line indexed by effective address bits [13:5].
2. The tag is compared with bits [28:10] of the address resulting from effective address translation by the MMU:

	Copy-back	Write-through
• If the tag matches and the V bit is 1	→ 3.	→ 4.
• If the tag matches and the V bit is 0	→ 5.	→ 6.
• If the tag does not match and the V bit is 0	→ 5.	→ 6.
• If the tag does not match, the V bit is 1, and the U bit is 0	→ 5.	→ 6.
• If the tag does not match, the V bit is 1, and the U bit is 1	→ 7.	→ 6.

A data write in accordance with the access size (quadword/longword/word/byte) is performed for the data indexed by effective address bits [4:0] of the data field of the cache line indexed by effective address bits [13:5]. Then 1 is written to the U bit.

4. Cache hit (write-through)

   A data write in accordance with the access size (quadword/longword/word/byte) is performed for the data indexed by effective address bits [4:0] of the data field of the cache line indexed by effective address bits [13:5]. A write is also performed to the corresponding external memory using the specified access size.

5. Cache miss (no copy-back/write-back)

   A data write in accordance with the access size (quadword/longword/word/byte) is performed for the data indexed by effective address bits [4:0] of the data field of the cache line indexed by effective address bits [13:5]. Then, data is read into the cache line from the external memory space corresponding to the effective address. Data reading is performed, using the wraparound method, in order from the longword data corresponding to the effective address, and one cache line of data is read excluding the written data. During this time, the CPU can execute the next processing. When reading of one line of data is completed, the tag corresponding to the effective address is recorded in the cache, and 1 is written to the V bit and U bit.

6. Cache miss (write-through)

   A write of the specified access size is performed to the external memory corresponding to the effective address. In this case, a write to cache is not performed.

7. Cache miss (with copy-back/write-back)

   The tag and data field of the cache line indexed by effective address bits [13:5] are first saved in the write-back buffer, and then a data write in accordance with the access size (quadword/longword/word/byte) is performed for the data indexed by effective address bits [4:0] of the data field of the cache line indexed by effective address bits [13:5]. Then, data is read into the cache line from the external memory space corresponding to the effective address. Data reading is performed, using the wraparound method, in order from the longword data corresponding to the effective address, and one cache line of data is read excluding the written data. During this time, the CPU can execute the next processing. When reading of one line of data is completed, the tag corresponding to the effective address is recorded in the cache, and 1 is written to the V bit and U bit. The data in the write-back buffer is then written back to external memory.

In order to give priority to data reads to the cache and improve performance, this LSI has a write-back buffer which holds the relevant cache entry when it becomes necessary to purge a dirty cache entry into external memory as the result of a cache miss. The write-back buffer contains one cache line of data and the physical address of the purge destination.

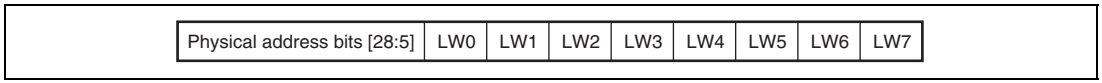


Figure 7.3 Configuration of Write-Back Buffer

7.3.4 Write-Through Buffer

This LSI has a 64-bit buffer for holding write data when writing data in write-through mode or writing to a non-cacheable area. This allows the CPU to proceed to the next operation as soon as the write to the write-through buffer is completed, without waiting for completion of the write to external memory.

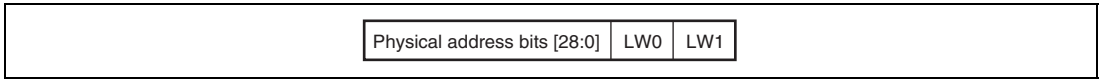


Figure 7.4 Configuration of Write-Through Buffer

7.3.5 RAM Mode

Setting the ORA bit in CCR to 1 enables half of the operand cache to be used as RAM. In cache direct mapping mode, the 8-Kbyte area otherwise used for OC entries 256 to 511 is designated as a RAM area. In double-size cache mode, a total of 16 Kbytes, comprising entries 256 to 511 in both of the ways of the operand cache, is designated as a RAM area. Other entries can still be used as cache. RAM can be accessed using addresses H'7C00 0000 to H'7FFF FFFF. Byte-, word-, longword-, and quadword-size data reads and writes can be performed in the operand cache RAM area. Instruction fetches cannot be performed in this area. This LSI cannot be used in OC index mode when RAM mode is selected.

Examples of RAM usage is shown below.

- In cache direct mapping mode (EMODE = 0 in CCR)
  H'7C00 0000 to H'7C00 1FFF (8 Kbytes): RAM area (entries 256 to 511)
  H'7C00 2000 to H'7C00 3FFF (8 Kbytes): RAM area (entries 256 to 511)
  : : :
  In the same pattern, shadows of the RAM area are created in 8-Kbyte blocks until H'7FFF FFFF is reached.

In this mode, the 8-Kbytes comprising entries 256 to 511 of OC way 0 are designated as RAM area 1 and the 8-Kbytes comprising entries 256 to 511 of OC way 1 are designated as RAM area 2.

H'7C00 0000 to H'7C00 1FFF (8 Kbytes): Corresponds to RAM area 1

H'7C00 2000 to H'7C00 3FFF (8 Kbytes): Corresponds to RAM area 2

H'7C00 4000 to H'7C00 5FFF (8 Kbytes): Corresponds to RAM area 1

H'7C00 6000 to H'7C00 7FFF (8 Kbytes): Corresponds to RAM area 2

: : :

In the same pattern, shadows of the RAM area are created in 16-Kbyte blocks until H'7FFF FFFF is reached.

7.3.6 OC Index Mode

Setting the OIX bit in CCR to 1 enables OC indexing to be performed using bit [25] of the effective address. This is called OC index mode. In normal mode, with the OIX bit in CCR cleared to 0, OC indexing is performed using bits [13:5] of the effective address. Using index mode allows the OC to be handled as two 8-Kbyte areas by means of effective address bit [25], providing efficient use of the cache. This LSI cannot be used in RAM mode when OC index mode is selected.

7.3.7 Coherency between Cache and External Memory

Coherency between cache and external memory should be assured by software. In this LSI, the following four new instructions are supported for cache operations. Details of these instructions are given in the Programming Manual.

- Invalidate instruction : OCBI @Rn Cache invalidation (no write-back)
- Purge instruction : OCBP @Rn Cache invalidation (with write-back)
- Write-back instruction : OCBWB @Rn Cache write-back
- Allocate instruction : MOVCA.L R0,@Rn Cache allocation

This LSI supports a prefetch instruction to reduce the cache fill penalty incurred as the result of a cache miss. If it is known that a cache miss will result from a read or write operation, it is possible to fill the cache with data beforehand by means of the prefetch instruction to prevent a cache miss due to the read or write operation, and so improve software performance. If a prefetch instruction is executed for data already held in the cache, or if the prefetch address results in a UTLB miss or a protection violation, the result is no operation, and an exception is not generated. Details of the prefetch instruction are given in the Programming Manual.

- Prefetch instruction : PREF @Rn

7.3.9 Note on Using Cache Enhanced Mode

When cache enhanced mode (CCR.EMODE = 1) is specified and OC RAM mode (CCR.ORA = 1) is selected, in which half of the operand cache is used as internal RAM, internal RAM data may be updated incorrectly.

Conditions Under which Problem Occurs: Incorrect data may be written to RAM when the following four conditions are satisfied.

Condition 1: Cache enhanced mode (CCR.EMODE = 1) is specified.

Condition 2: The RAM mode (CCR.ORA = 1) in which half of the operand cache is used as internal RAM is specified.

Condition 3: An exception or an interrupt occurs.

Note: This includes a break triggered by a debugging tool swapping an instruction (a break occurring when a TRAPA instruction or undefined instruction code H'FFFD is swapped for an instruction).

Condition 4: A store instruction (MOV, FMOV, AND.B, OR, B, XOR.B, MOVCA.L, STC.L, or STS.L) that accesses internal RAM (H'7C000000 to H'7FFFFFFF) exists within four instructions after the instruction associated with the exception or interrupt described in condition 3. This includes cases where the store instruction that accesses internal RAM itself generates an exception.

Description: When the problem occurs, 8 bytes of incorrect data is written to the 8-byte boundary that includes an address that differs by H'2000 from the address accessed by the store instruction that accesses internal RAM mentioned in condition 4. For example, when a long word is stored at address H'7C000204, the 8 bytes of data in the internal RAM mapped to addresses H'7C002200 to H'7C002207 becomes corrupted.

Example 1 A store instruction accessing internal RAM occurs within four instructions after an instruction generating a TLB miss exception.

MOV.L #H'0C400000, R0	R0 is an address causing a TLB miss.
MOV.L #H'7C000204, R1	R1 is an address mapped to internal RAM.
MOV.L @R0, R2	TLB miss exception occurs.
NOP	1st word
NOP	2nd word
NOP	3rd word
MOV.L R3, @R1	Store instruction accessing internal RAM

Example 2 A store instruction accessing internal RAM occurs within four instructions after an instruction causing an interrupt to be accepted.

MOV.L #H'7C002000, R1	R1 is an address mapped to internal RAM.
MOV.L #H'12345678, R0	An interrupt is accepted after this instruction.
NOP	1st word
NOP	2nd word
NOP	3rd word
MOV.L R0, @R1	Store instruction accessing internal RAM

Example 3 A debugging tool generates a break to swap an instruction.

Original Instruction String	**After Instruction Swap Break**	
MOV.L #H'C000000, R0	MOV.L #H'7C000000, R0	Contains address corresponding to R0.
ADD R0, R0	TRAPA #H'01	R0 address is not a problem in original instruction string.
MOV.L R1, @R0	MOV.L R1, @R0	Internal RAM is accessed by a store operation because ADD is not executed. The store is cancelled, but 2LW starting at H'7C002000 is corrupted.

workarounds can be used to avoid the problem.

Workaround 1:

Use only 8 Kbytes of the 16-Kbyte internal RAM area. In this case, RAM areas for which address bits [12:0] are identical and only bit [13] differs must not be used. For example, the 8-Kbyte RAM area from H'7C000000 to H'7C001FFF or from H'7C001000 to H'7C002FFF may be used.

Note: When a break is used to swap instructions by a debugging tool, etc., a memory access address may be changed when an instruction following the instruction generating the break is swapped for another instruction, causing the unused 8-Kbyte RAM area to be accessed. This will result in the problem described above. However, this phenomenon only occurs during debugging when a break is used to swap instructions. Using a break with no instruction swapping will not cause the problem.

Workaround 2:

Ensure that there are no instructions that generate an interrupt or exception within four instructions after an instruction that accesses internal RAM. For example, the internal RAM area can be used as a data table that is accessed only by load instructions, with writes to the internal RAM area only being performed when the table is generated. It this case, set SR.BL to 1 to disable interrupts while writing to the table. Also take measures to ensure that no exceptions due to TLB misses, etc., occur while writing to the table.

Note: The problem still may occur when a break is used to swap instructions by a debugging tool. This phenomenon only occurs during debugging when a break is used to swap instructions. Using a break with no instruction swapping will not cause the problem.

7.4.1 Read Operation

When the IC is enabled (ICE = 1 in CCR) and instruction fetches are performed by means of an effective address from a cacheable area, the instruction cache operates as follows:

1. The tag and V bit are read from the cache line indexed by effective address bits [12:5].
2. The tag is compared with bits [28:10] of the address resulting from effective address translation by the MMU:
   - If the tag matches and the V bit is 1 → 3.
   - If the tag matches and the V bit is 0 → 4.
   - If the tag does not match and the V bit is 0 → 4.
   - If the tag does not match and the V bit is 1 → 4.

3. Cache hit

   The data indexed by effective address bits [4:2] is read as an instruction from the data field of the cache line indexed by effective address bits [12:5].
4. Cache miss

   Data is read into the cache line from the external memory space corresponding to the effective address. Data reading is performed, using the wraparound method, in order from the longword data corresponding to the effective address, and when the corresponding data arrives in the cache, the read data is returned to the CPU as an instruction. When reading of one line of data is completed, the tag corresponding to the effective address is recorded in the cache, and 1 is written to the V bit.

7.4.2 IC Index Mode

Setting the IIX bit in CCR to 1 enables IC indexing to be performed using bit [25] of the effective address. This is called IC index mode. In normal mode, with the IIX bit in CCR cleared to 0, IC indexing is performed using bits [12:5] of the effective address. Using index mode allows the IC to be handled as two 4-Kbyte areas by means of effective address bit [25], providing efficient use of the cache.

Mode)

To enable the IC and OC to be managed by software, the IC contents can be read from or written to by a program in the P2 area by means of a MOV instruction in privileged mode. Operation is not guaranteed if access is made from a program in another area. In this case, a branch to the P0, U0, P1, or P3 area should be made at least eight instructions after this MOV instruction. In privileged mode, the OC contents can be read from or written to by a program in the P1 or P2 area by means of a MOV instruction. Operation is not guaranteed if access is made from a program in another area. In this case, a branch to the P0, U0, or P3 area should be made at least eight instructions after this MOV instruction The IC and OC are allocated to the P4 area in the physical address space. Only data accesses can be used on both the IC address array and data array and the OC address array and data array, and accesses are always longword-size. Instruction fetches cannot be performed in these areas. For reserved bits, a write value of 0 should be specified and the read value is undefined.

7.5.1 IC Address Array

The IC address array is allocated to addresses H'F000 0000 to H'F0FF FFFF in the P4 area. An address array access requires a 32-bit address field specification (when reading or writing) and a 32-bit data field specification. The entry to be accessed is specified in the address field, and the write tag and V bit are specified in the data field.

In the address field, bits [31:24] have the value H'F0 indicating the IC address array, and the entry is specified by bits [12:5]. The IIX bit in CCR has no effect on this entry specification. The association bit (A bit) [3] in the address field specifies whether or not association is performed when writing to the IC address array. As only longword access is used, 0 should be specified for address field bits [1:0].

In the data field, the tag is indicated by bits [31:10], and the V bit by bit [0]. As the IC address array tag is 19 bits in length, data field bits [31:29] are not used in the case of a write in which association is not performed. Data field bits [31:29] are used for the virtual address specification only in the case of a write in which association is performed.

The following three kinds of operation can be used on the IC address array:

1. IC address array read

   The tag and V bit are read into the data field from the IC entry corresponding to the entry set in the address field. In a read, associative operation is not performed regardless of whether the association bit specified in the address field is 1 or 0.

The tag and V bit specified in the data field are written to the IC entry corresponding to the entry set in the address field. The A bit in the address field should be cleared to 0.

3. IC address array write (associative)

   When a write is performed with the A bit in the address field set to 1, the tag stored in the entry specified in the address field is compared with the tag specified in the data field. If the MMU is enabled at this time, comparison is performed after the virtual address specified by data field bits [31:10] has been translated to a physical address using the ITLB. If the addresses match and the V bit is 1, the V bit specified in the data field is written into the IC entry. In other cases, no operation is performed. This operation is used to invalidate a specific IC entry. If an ITLB miss occurs during address translation, or the comparison shows a mismatch, an exception is not generated, no operation is performed, and the write is not executed. If an instruction TLB multiple hit exception occurs during address translation, processing switches to the instruction TLB multiple hit exception handling routine.

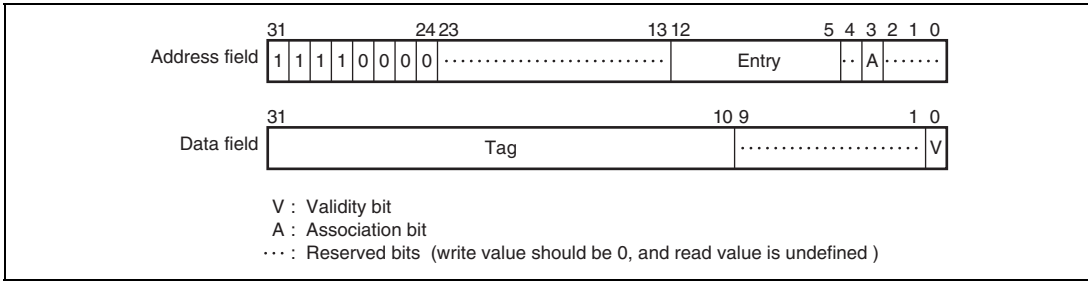


Figure 7.5 Memory-Mapped IC Address Array

7.5.2 IC Data Array

The IC data array is allocated to addresses H'F100 0000 to H'F1FF FFFF in the P4 area. A data array access requires a 32-bit address field specification (when reading or writing) and a 32-bit data field specification. The entry to be accessed is specified in the address field, and the longword data to be written is specified in the data field.

In the address field, bits [31:24] have the value H'F1 indicating the IC data array, and the entry is specified by bits [12:5]. The IIX bit in CCR has no effect on this entry specification. Address field bits [4:2] are used for the longword data specification in the entry. As only longword access is used, 0 should be specified for address field bits [1:0].

The data field is used for the longword data specification.

1. IC data array read

   Longword data is read into the data field from the data specified by the longword specification bits in the address field in the IC entry corresponding to the entry set in the address field.

2. IC data array write

   The longword data specified in the data field is written for the data specified by the longword specification bits in the address field in the IC entry corresponding to the entry set in the address field.

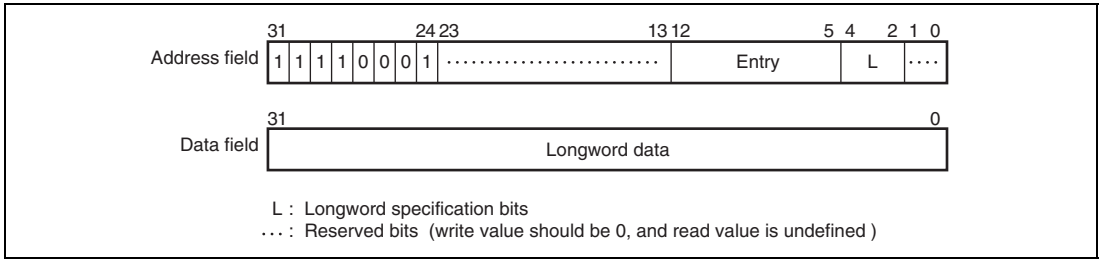


Figure 7.6 Memory-Mapped IC Data Array

7.5.3 OC Address Array

The OC address array is allocated to addresses H'F400 0000 to H'F4FF FFFF in the P4 area. An address array access requires a 32-bit address field specification (when reading or writing) and a 32-bit data field specification. The entry to be accessed is specified in the address field, and the write tag, U bit, and V bit are specified in the data field.

In the address field, bits [31:24] have the value H'F4 indicating the OC address array, and the entry is specified by bits [13:5]. The OIX and ORA bits in CCR have no effect on this entry specification. The association bit (A bit) [3] in the address field specifies whether or not association is performed when writing to the OC address array. As only longword access is used, 0 should be specified for address field bits [1:0].

In the data field, the tag is indicated by bits [31:10], the U bit by bit [1], and the V bit by bit [0]. As the OC address array tag is 19 bits in length, data field bits [31:29] are not used in the case of a write in which association is not performed. Data field bits [31:29] are used for the virtual address specification only in the case of a write in which association is performed.

1. OC address array read

   The tag, U bit, and V bit are read into the data field from the OC entry corresponding to the entry set in the address field. In a read, associative operation is not performed regardless of whether the association bit specified in the address field is 1 or 0.

2. OC address array write (non-associative)

   The tag, U bit, and V bit specified in the data field are written to the OC entry corresponding to the entry set in the address field. The A bit in the address field should be cleared to 0.

   When a write is performed to a cache line for which the U bit and V bit are both 1, after write-back of that cache line, the tag, U bit, and V bit specified in the data field are written.

3. OC address array write (associative)

   When a write is performed with the A bit in the address field set to 1, the tag stored in the entry specified in the address field is compared with the tag specified in the data field. If the MMU is enabled at this time, comparison is performed after the virtual address specified by data field bits [31:10] has been translated to a physical address using the UTLB. If the addresses match and the V bit is 1, the U bit and V bit specified in the data field are written into the OC entry. In other cases, no operation is performed. This operation is used to invalidate a specific OC entry. If the OC entry U bit is 1, and 0 is written to the V bit or to the U bit, write-back is performed. If a UTLB miss occurs during address translation, or the comparison shows a mismatch, an exception is not generated, no operation is performed, and the write is not executed. If a data TLB multiple hit exception occurs during address translation, processing switches to the data TLB multiple hit exception handling routine.

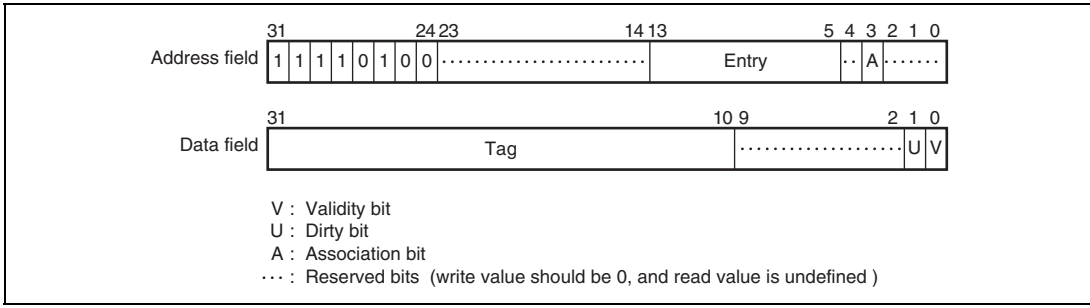


Figure 7.7 Memory-Mapped OC Address Array

The OC data array is allocated to addresses H'F500 0000 to H'F5FF FFFF in the P4 area. A data array access requires a 32-bit address field specification (when reading or writing) and a 32-bit data field specification. The entry to be accessed is specified in the address field, and the longword data to be written is specified in the data field.

In the address field, bits [31:24] have the value H'F5 indicating the OC data array, and the entry is specified by bits [13:5]. The OIX and ORA bits in CCR have no effect on this entry specification. Address field bits [4:2] are used for the longword data specification in the entry. As only longword access is used, 0 should be specified for address field bits [1:0].

The data field is used for the longword data specification.

The following two kinds of operation can be used on the OC data array:

1. OC data array read

   Longword data is read into the data field from the data specified by the longword specification bits in the address field in the OC entry corresponding to the entry set in the address field.
2. OC data array write

   The longword data specified in the data field is written for the data specified by the longword specification bits in the address field in the OC entry corresponding the entry set in the address field. This write does not set the U bit to 1 on the address array side.

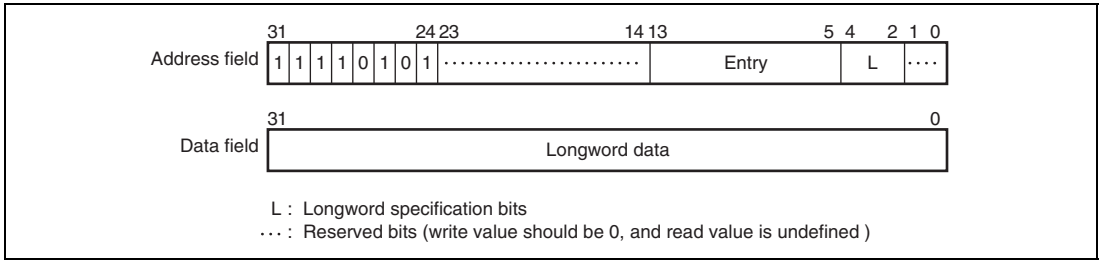


Figure 7.8 Memory-Mapped OC Data Array

To enable the IC and OC to be managed by software, the IC contents can be read from or written to by a program in the P2 area by means of a MOV instruction in privileged mode. Operation is not guaranteed if access is made from a program in another area. In this case, a branch to the P0, U0, P1, or P3 area should be made at least eight instructions after this MOV instruction. In privileged mode, the OC contents can be read from or written to by a program in the P1 or P2 area by means of a MOV instruction. Operation is not guaranteed if access is made from a program in another area. In this case, a branch to the P0, U0, or P3 area should be made at least eight instructions after this MOV instruction. The IC and OC are allocated to the P4 area in the physical address space. Only data accesses can be used on both the IC address array and data array and the OC address array and data array, and accesses are always longword-size. Instruction fetches cannot be performed in these areas. For reserved bits, a write value of 0 should be specified; their read value is undefined.

7.6.1 IC Address Array

The IC address array is allocated to addresses H'F000 0000 to H'F0FF FFFF in the P4 area. An address array access requires a 32-bit address field specification (when reading or writing) and a 32-bit data field specification. The way and entry to be accessed are specified in the address field, and the write tag and V bit are specified in the data field.

In the address field, bits [31:24] have the value H'F0 indicating the IC address array, the way is specified by bit [13], and the entry by bits [12:5]. The IIX bit in CCR has no effect on this entry specification. The association bit (A bit) [3] in the address field specifies whether or not association is performed when writing to the IC address array. As only longword access is used, 0 should be specified for address field bits [1:0].

In the data field, the tag is indicated by bits [31:10], and the V bit by bit [0]. As the IC address array tag is 19 bits in length, data field bits [31:29] are not used in the case of a write in which association is not performed. Data field bits [31:29] are used for the virtual address specification only in the case of a write in which association is performed.

The following three kinds of operation can be used on the IC address array:

1. IC address array read

   The tag and V bit are read into the data field from the IC entry corresponding to the way and entry set in the address field. In a read, associative operation is not performed regardless of whether the association bit specified in the address field is 1 or 0.
2. IC address array write (non-associative)

   The tag and V bit specified in the data field are written to the IC entry corresponding to the way and entry set in the address field. The A bit in the address field should be cleared to 0.

When a write is performed with the A bit in the address field set to 1, the tag for each of the ways stored in the entry specified in the address field is compared with the tag specified in the data field. The way number set by bit [13] is not used. If the MMU is enabled at this time, comparison is performed after the virtual address specified by data field bits [31:10] has been translated to a physical address using the ITLB. If the addresses match and the V bit for that way is 1, the V bit specified in the data field is written into the IC entry. In other cases, no operation is performed. This operation is used to invalidate a specific IC entry. If an ITLB miss occurs during address translation, or the comparison shows a mismatch, an exception is not generated, no operation is performed, and the write is not executed. If an instruction TLB multiple hit exception occurs during address translation, processing switches to the instruction TLB multiple hit exception handling routine.

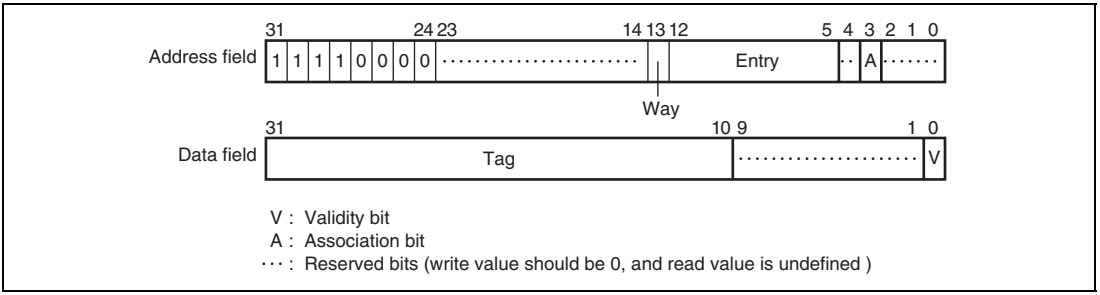


Figure 7.9 Memory-Mapped IC Address Array

7.6.2 IC Data Array

The IC data array is allocated to addresses H'F100 0000 to H'F1FF FFFF in the P4 area. A data array access requires a 32-bit address field specification (when reading or writing) and a 32-bit data field specification. The way and entry to be accessed are specified in the address field, and the longword data to be written is specified in the data field.

In the address field, bits [31:24] have the value H'F1 indicating the IC data array, the way is specified by bit [13], and the entry by bits [12:5]. The IIX bit in CCR has no effect on this entry specification. Address field bits [4:2] are used for the longword data specification in the entry. As only longword access is used, 0 should be specified for address field bits [1:0].

The data field is used for the longword data specification.

1. IC data array read

   Longword data is read into the data field from the data specified by the longword specification bits in the address field in the IC entry corresponding to the way and entry set in the address field.

2. IC data array write

   The longword data specified in the data field is written for the data specified by the longword specification bits in the address field in the IC entry corresponding to the way and entry set in the address field.

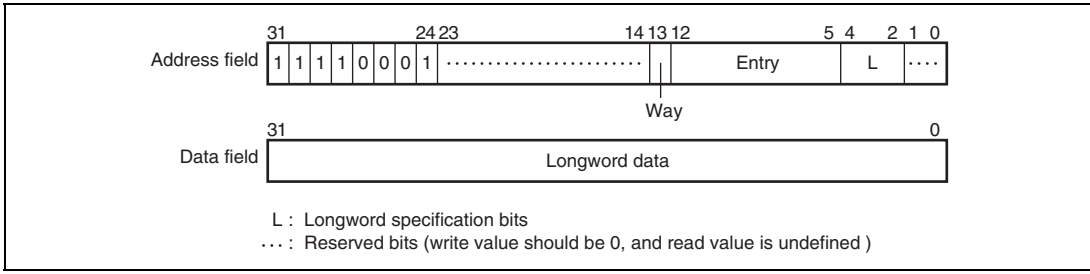


Figure 7.10 Memory-Mapped IC Data Array

7.6.3 OC Address Array

The OC address array is allocated to addresses H'F400 0000 to H'F4FF FFFF in the P4 area. An address array access requires a 32-bit address field specification (when reading or writing) and a 32-bit data field specification. The way and entry to be accessed are specified in the address field, and the write tag, U bit, and V bit are specified in the data field.

In the address field, bits [31:24] have the value H'F4 indicating the OC address array, the way is specified by bit [14], and the entry by bits [13:5]. The OIX bit in CCR has no effect on this entry specification. In RAM mode (ORA = 1 in CCR), the OC address arrays are only accessible in the memory-mapped cache area, and bit [13] is used to specify the way. For details about address mapping, see section 7.6.5, Summary of Memory-Mapping of OC. The association bit (A bit) [3] in the address field specifies whether or not association is performed when writing to the OC address array. As only longword access is used, 0 should be specified for address field bits [1:0].

In the data field, the tag is indicated by bits [31:10], the U bit by bit [1], and the V bit by bit [0]. As the OC address array tag is 19 bits in length, data field bits [31:29] are not used in the case of a write in which association is not performed. Data field bits [31:29] are used for the virtual address specification only in the case of a write in which association is performed.

1. OC address array read

   The tag, U bit, and V bit are read into the data field from the OC entry corresponding to the way and entry set in the address field. In a read, associative operation is not performed regardless of whether the association bit specified in the address field is 1 or 0.

2. OC address array write (non-associative)

   The tag, U bit, and V bit specified in the data field are written to the OC entry corresponding to the way and entry set in the address field. The A bit in the address field should be cleared to 0.

   When a write is performed to a cache line for which the U bit and V bit are both 1, after write-back of that cache line, the tag, U bit, and V bit specified in the data field are written.

3. OC address array write (associative)

   When a write is performed with the A bit in the address field set to 1, the tag for each of the ways stored in the entry specified in the address field is compared with the tag specified in the data field. The way number set by bit [14] is not used. If the MMU is enabled at this time, comparison is performed after the virtual address specified by data field bits [31:10] has been translated to a physical address using the UTLB. If the addresses match and the V bit for that way is 1, the U bit and V bit specified in the data field are written into the OC entry. In other cases, no operation is performed. This operation is used to invalidate a specific OC entry. If the OC entry U bit is 1, and 0 is written to the V bit or to the U bit, write-back is performed. If a UTLB miss occurs during address translation, or the comparison shows a mismatch, an exception is not generated, no operation is performed, and the write is not executed. If a data TLB multiple hit exception occurs during address translation, processing switches to the data TLB multiple hit exception handling routine.

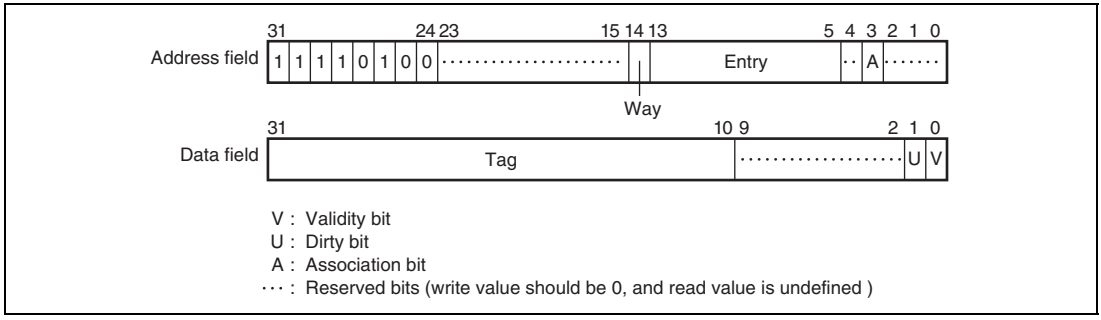


Figure 7.11 Memory-Mapped OC Address Array

The OC data array is allocated to addresses H'F500 0000 to H'F5FF FFFF in the P4 area. A data array access requires a 32-bit address field specification (when reading or writing) and a 32-bit data field specification. The way and entry to be accessed are specified in the address field, and the longword data to be written is specified in the data field.

In the address field, bits [31:24] have the value H'F5 indicating the OC data array, the way is specified by bit [14], and the entry by bits [13:5]. The OIX bit in CCR has no effect on this entry specification. In RAM mode (ORA = 1 in CCR), the OC data arrays are only accessible in the memory-mapped cache area, and bit [13] is used to specify the way. For details about address mapping, see section 7.6.5, Summary of Memory-Mapping of OC. Address field bits [4:2] are used for the longword data specification in the entry. As only longword access is used, 0 should be specified for address field bits [1:0].

The data field is used for the longword data specification.

The following two kinds of operation can be used on the OC data array:

1. OC data array read

   Longword data is read into the data field from the data specified by the longword specification bits in the address field in the OC entry corresponding to the way and entry set in the address field.
2. OC data array write

   The longword data specified in the data field is written for the data specified by the longword specification bits in the address field in the OC entry corresponding to the way and entry set in the address field. This write does not set the U bit to 1 on the address array side.

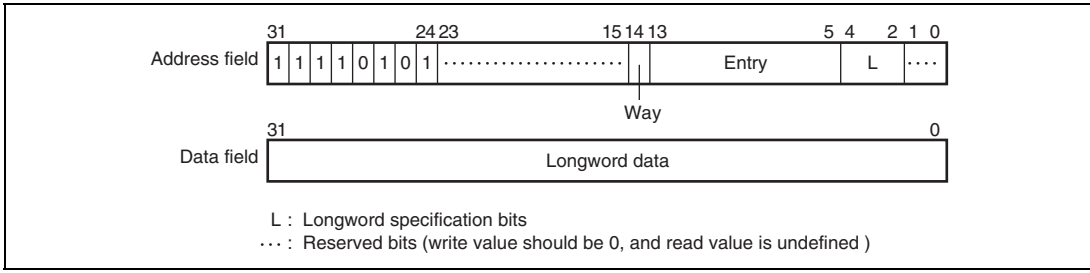


Figure 7.12 Memory-Mapped OC Data Array

The address ranges to which the OC is memory-mapped in double-size cache mode of this LSI are summarized below in the example of data array access.

- In normal mode (ORA = 0 in CCR)

  H'F500 0000 to H'F500 3FFF (16 Kbytes): Way 0 (entries 0 to 511)

  H'F500 4000 to H'F500 7FFF (16 Kbytes): Way 1 (entries 0 to 511)

  In the same pattern, shadows of the cache area are created in 32-Kbyte blocks until H'F5FF FFFF.
- In RAM mode (ORA = 1 in CCR)

  H'F500 0000 to H'F500 1FFF (8 Kbytes): Way 0 (entries 0 to 255)

  H'F500 2000 to H'F500 3FFF (8 Kbytes): Way 1 (entries 0 to 255)

  In the same pattern, shadows of the cache area are created in 16-Kbyte blocks until H'F5FF FFFF.

7.7 Store Queues

This LSI supports two 32-byte store queues (SQs) to perform high-speed writes to external memory. If the SQs are not used, power-down modes, in which SQ functions are stopped, can be used to reduce power consumption. The queue address control registers (QACR0 and QACR1) cannot be accessed while SQ functions are stopped. See section 14, Power-Down Modes, for the procedure for stopping SQ functions.

7.7.1 SQ Configuration

There are two 32-byte store queues, SQ0 and SQ1, as shown in figure 7.13. These two store queues can be set independently.

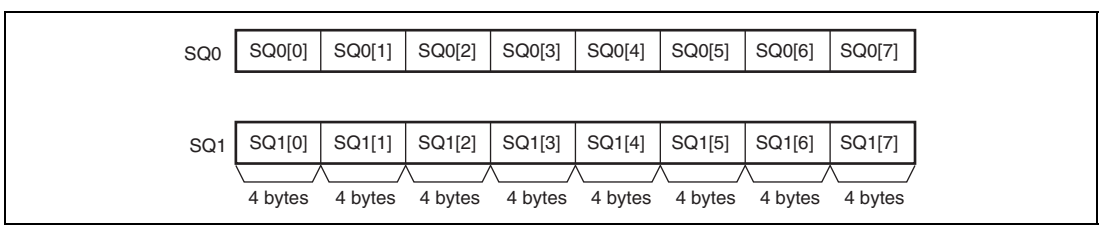


Figure 7.13 Store Queue Configuration

A write to the SQs can be performed using a store instruction for addresses H'E000 0000 to H'E3FF FFFC in the P4 area. A longword or quadword access size can be used. The meanings of the address bits are as follows:

[31:26]	: 111000	Store queue specification
[25:6]	: Don't care	Used for external memory transfer/access right
[5]	: 0/1	0: SQ0 specification 1: SQ1 specification
[4:2]	: LW specification	Specifies longword position in SQ0/SQ1
[1:0]	: 00	Fixed at 0

7.7.3 Transfer to External Memory

Transfer from the SQs to external memory can be performed with a prefetch instruction (PREF). Issuing a PREF instruction for addresses H'E000 0000 to H'E3FF FFFC in the P4 area starts a transfer from the SQs to external memory. The transfer length is fixed at 32 bytes, and the start address is always at a 32-byte boundary. While the contents of one SQ are being transferred to external memory, the other SQ can be written to without a penalty cycle. However, writing to the SQ involved in the transfer to external memory is kept waiting until the transfer is completed.

The external address bits [28:0] of the SQ transfer destination are specified as shown below, according to whether the MMU is on or off.

- When MMU is on (AT = 1 in MMUCR)

  The SQ area (H'E000 0000 to H'E3FF FFFF) is set in VPN of the UTLB, and the transfer destination external address in PPN. The ASID, V, SZ, SH, PR, and D bits have the same meaning as for normal address translation, but the C and WT bits have no meaning with regard to this page. Transfer to the PCMCIA interface area by means of the SQs is not allowed. When a prefetch instruction is issued for the SQ area, address translation is performed and external address bits [28:10] are generated in accordance with the SZ bit specification. For external address bits [9:5], the address prior to address translation is generated in the same way as when the MMU is off. External address bits [4:0] are fixed at 0. Transfer from the SQs to external memory is performed to this address.

The SQ area (H'E000 0000 to H'E3FF FFFF) is specified as the address at which a PREF instruction is issued. The meanings of address bits [31:0] are as follows:

[31:26]	: 111000	Store queue specification
[25:6]	: Address	External address bits [25:6]
[5]	: 0/1	0: SQ0 specification 1: SQ1 specification and external address bit [5]
[4:2]	: Don't care	No meaning in a prefetch
[1:0]	: 00	Fixed at 0

External address bits [28:26], which cannot be generated from the above address, are generated from QACR0 and QACR1.

QACR0[4:2] : External address bits [28:26] corresponding to SQ0
QACR1[4:2] : External address bits [28:26] corresponding to SQ1

External address bits [4:0] are always fixed at 0 since burst transfer starts at a 32-byte boundary.

Data transfer to a PCMCIA interface area in this LSI is always performed using the values of the SA and TC bits in PTEA.

7.7.4 Determination of SQ Access Exception

Determination of an exception in a write to an SQ or transfer to external memory (PREF instruction) is performed as follows according to whether the MMU is on or off. If an exception occurs during a write to an SQ, the SQ contents before the write are retained. If an exception occurs in a data transfer from an SQ to external memory, the transfer to external memory will be aborted.

- When MMU is on (AT = 1 in MMUCR)

  Operation is in accordance with the address translation information recorded in the UTLB, and the SQMD bit in MMUCR. Write type exception judgment is performed for writes to the SQs, and read type exception judgment for transfer from the SQs to external memory (using a PREF instruction). As a result, a TLB miss exception, protection violation exception, or initial page write exception is generated as required. However, if SQ access is enabled in privileged mode only by the SQMD bit in MMUCR, an address error will occur even if address translation is successful in user mode.

Operation is in accordance with the SQMD bit in MMUCR.

0: Privileged/user mode access possible

1: Privileged mode access possible

If the SQ area is accessed in user mode when the SQMD bit in MMUCR is set to 1, an address error will occur.

7.7.5 Reading from SQ

In privileged mode in this LSI, reading the contents of the SQs may be performed by means of a load instruction for addresses H'FF00 1000 to H'FF00 103C in the P4 area. Only longword access is possible.

[31:6]	: H'FF00 1000	Store queue specification
[5]	: 0/1	0: SQ0 specification
		1: SQ1 specification
[4:2]	: LW specification	Specifies longword position in SQ0/SQ1
[1:0]	: 00	Fixed at 0

8.1 Exception Handling Functions

Exception handling processing is handled by a special routine, separate from normal program processing, which is executed by the CPU in case of abnormal events. For example, if the executing instruction ends abnormally, appropriate action must be taken in order to return to the original program sequence, or report the abnormality before terminating the processing. The process of generating an exception handling request in response to abnormal termination, and passing control to a user-written exception handling routine, in order to support such functions, is given the generic name of exception handling.

The SH-4 exception handling is of three kinds: for resets, general exceptions, and interrupts.

8.1.1 Exception Handling Flow

In exception handling, the contents of the program counter (PC), status register (SR), and R15 are saved in the saved program counter (SPC), saved status register (SSR), and saved general register15 (SGR), and the CPU starts execution of the appropriate exception handling routine according to the vector address. An exception handling routine is a program written by the user to handle a specific exception. The exception handling routine is terminated and control returned to the original program by executing a return-from-exception instruction (RTE). This instruction restores the PC and SR contents and returns control to the normal processing routine at the point at which the exception occurred. The SGR contents are not written back to R15 with an RTE instruction.

The basic processing flow is as follows.

1. The PC, SR, and R15 contents are saved in SPC, SSR, and SGR, respectively.
2. The block bit (BL) in SR is set to 1.
3. The mode bit (MD) in SR is set to 1.
4. The register bank bit (RB) in SR is set to 1.
5. In a reset, the FPU disable bit (FD) in SR is cleared to 0.
6. The exception code is written to bits 11 to 0 of the exception event register (EXPEVT) or interrupt event register (INTEVT).
7. The CPU branches to the determined exception handling vector address, and the exception handling routine begins.

The reset vector address is fixed at H'A000 0000. Exception and interrupt vector addresses are determined by adding the offset for the specific event to the vector base address, which is set by software in the vector base register (VBR). In the case of the TLB miss exception, for example, the offset is H'0000 0400, so if H'9C08 0000 is set in VBR, the exception handling vector address will be H'9C08 0400. If a further exception occurs at the exception handling vector address, a duplicate exception will result, and recovery will be difficult; therefore, fixed physical addresses (in P1 and P2 areas) should be specified for vector addresses.

Table 8.1 shows the types of exceptions, with their relative priorities, vector addresses, and exception/interrupt codes.

Table 8.1 Exception Sources and Priorities

Exception Category	Execution Mode	Exception	Priority Level	Priority Order	Vector Address	Offset	Exception Code
Reset	Abort type	Power-on reset	1	1	H'A000 0000	—	H'000
		Manual reset	1	2	H'A000 0000	—	H'020
		H-UDI reset	1	1	H'A000 0000	—	H'000
		Instruction TLB multiple-hit exception	1	3	H'A000 0000	—	H'140
		Data TLB multiple-hit exception	1	4	H'A000 0000	—	H'140
General exception	Re-execution type	User break before instruction execution*1	2	0	(VBR/DBR)	H'100/—	H'1E0
		Instruction address error	2	1	(VBR)	H'100	H'0E0
		Instruction TLB miss exception	2	2	(VBR)	H'400	H'040
		Instruction TLB protection violation exception	2	3	(VBR)	H'100	H'0A0
		General illegal instruction exception	2	4	(VBR)	H'100	H'180
		Slot illegal instruction exception	2	4	(VBR)	H'100	H'1A0
		General FPU disable exception	2	4	(VBR)	H'100	H'800
		Slot FPU disable exception	2	4	(VBR)	H'100	H'820
		Data address error (read)	2	5	(VBR)	H'100	H'0E0
		Data address error (write)	2	5	(VBR)	H'100	H'100
		Data TLB miss exception (read)	2	6	(VBR)	H'400	H'040
		Data TLB miss exception (write)	2	6	(VBR)	H'400	H'060
		Data TLB protection violation exception (read)	2	7	(VBR)	H'100	H'0A0
		Data TLB protection violation exception (write)	2	7	(VBR)	H'100	H'0C0
		FPU exception	2	8	(VBR)	H'100	H'120
		Initial page write exception	2	9	(VBR)	H'100	H'080
	Completion type	Unconditional trap (TRAPA)	2	4	(VBR)	H'100	H'160
		User break after instruction execution*1	2	10	(VBR/DBR)	H'100/—	H'1E0

Interrupt	Completion type	NMI			3	—	(VBR)	H'600	H'1C0
		External interrupts	IRL3 to IRL0	0	4	*2	(VBR)	H'600	H'200
				1					H'220
				2					H'240
				3					H'260
				4					H'280
				5					H'2A0
				6					H'2C0
				7					H'2E0
				8					H'300
				9					H'320
				A					H'340
				B					H'360
				C					H'380
				D					H'3A0
				E					H'3C0
			IRL	IRL0					H'240
				IRL1					H'2A0
				IRL2					H'300
				IRL3					H'360
		Peripheral module interrupt (module/ source)	DMAC	DMTE0	4	*2	(VBR)	H'600	H'640
				DMTE1					H'660
				DMTE2					H'680
				DMTE3					H'6A0
				DMTE4					H'780
				DMTE5					H'7A0
				DMTE6					H'7C0
				DMTE7					H'7E0
				DMAE					H'6C0
			IRQ*3	IRQ4					H'800
				IRQ5					H'820
				IRQ6					H'840
				IRQ7					H'860

Interrupt	Completion type	Peripheral module interrupt (module/ source)	HCAN2	CANI0	4	*2	(VBR)	H'600	H'900
				CANI1					H'920
			SSI	SSII0					H'940
				SSII1					H'960
			HAC	HACI0					H'980
				HACI1					H'9A0
			IIC	IICI0					H'9C0
				IICI1					H'9E0
			USB	USBI					H'A00
			LCDC	VINT					H'A20
			DMABRG	DMABRGI0					H'A80
				DMABRGI1					H'AA0
				DMABRGI2					H'AC0
			SCIF	ERI0					H'880
				RXI0					H'8A0
				BRI0					H'8C0
				TXI0					H'8E0
				ERI1					H'B00
				RXI1					H'B20
				BRI1					H'B40
				TXI1					H'B60
				ERI2					H'B80
				RXI2					H'BA0
				BRI2					H'BC0
				TXI2					H'BE0
			SIM	SIMERI					H'C00
				SIMRXI					H'C20
				SIMTXI					H'C40
				SIMTEI					H'C60
			HSPI	SPII					H'C80

Interrupt	Completion type	Peripheral module interrupt (module/ source)	MMCIF	MMCI0	4	*2	(VBR)	H'600	H'D00
				MMCI1					H'D20
				MMCI2					H'D40
				MMCI3					H'D60
			MFI	MFII					H'E80
			—	—					H'F00
				—					H'F20
				—					H'F40
				—					H'F60
			H-UDI	H-UDI					H'600
			GPIO	GPIOI					H'620
			ADC	ADI					H'F80
			CMT	CMTI					H'FA0
			TMU	TUNI0					H'400
				TUNI1					H'420
				TUNI2					H'440
				TICPI2					H'460
			WDT	ITI					H'560
			REF	RCMI					H'580
				ROVI					H'5A0

Priority : Priority is first assigned by priority level, then by priority order within each level (the lowest number represents the highest priority). For details of the interrupt priority level, see section 9, Interrupt Controller (INTC).

Exception transition destination : Control passes to H'A000 0000 in a reset, and to [VBR + offset] in other cases.

Exception code : Stored in EXPEVT for a reset or general exception, and in INTEVT for an interrupt.

IRL : Interrupt request level (pins IRL3 to IRL0).

Module/source : See the sections on the relevant peripheral modules.

Notes: 1. When UBDE in BRCR = 1, PC = DBR. In other cases, PC = VBR + H'100.

2. The priority order of external interrupts and peripheral module interrupts can be set by software.

3. IRQ is external interrupt.

8.3.1 Exception Flow

Figure 8.1 shows an outline flowchart of the basic operations in instruction execution and exception handling. For the sake of clarity, the following description assumes that instructions are executed sequentially, one by one. Figure 8.1 shows the relative priority order of the different kinds of exceptions (reset, general exception, and interrupt). Register settings in the event of an exception are shown only for SSR, SPC, SGR, EXPEVT/INTEVT, SR, and PC. However, other registers may be set automatically by hardware, depending on the exception. For details, see section 8.5, Operation. Also, see section 8.5.4, Priority Order with Multiple Exceptions, for exception handling during execution of a delayed branch instruction and a delay slot instruction, or in the case of instructions in which two data accesses are performed.

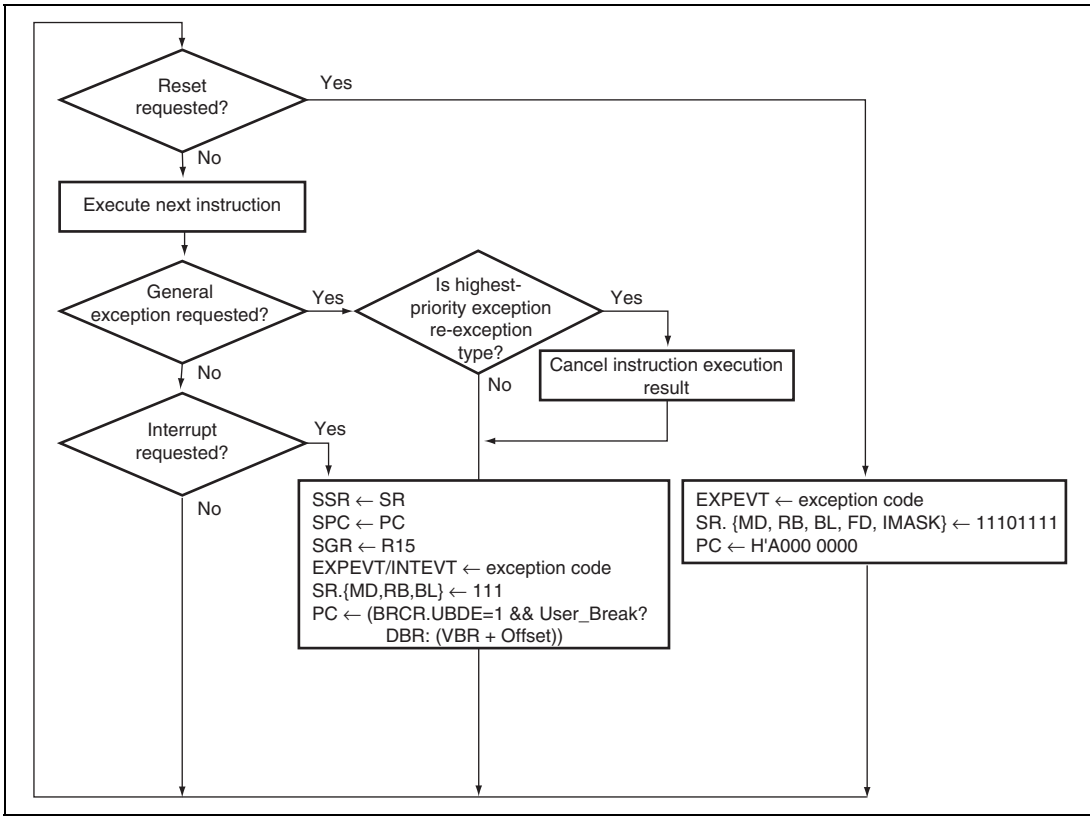


Figure 8.1 Instruction Execution and Exception Handling

A priority ranking is provided for all exceptions for use in determining which of two or more simultaneously generated exceptions should be accepted. Five of the general exceptions—general illegal instruction exception, slot illegal instruction exception, general FPU disable exception, slot FPU disable exception, and unconditional trap exception—are detected in the process of instruction decoding, and do not occur simultaneously in the instruction pipeline. These exceptions therefore all have the same priority. General exceptions are detected in the order of instruction execution. However, exception handling is performed in the order of instruction flow (program order). Thus, an exception for an earlier instruction is accepted before that for a later instruction. An example of the order of acceptance for general exceptions is shown in figure 8.2.

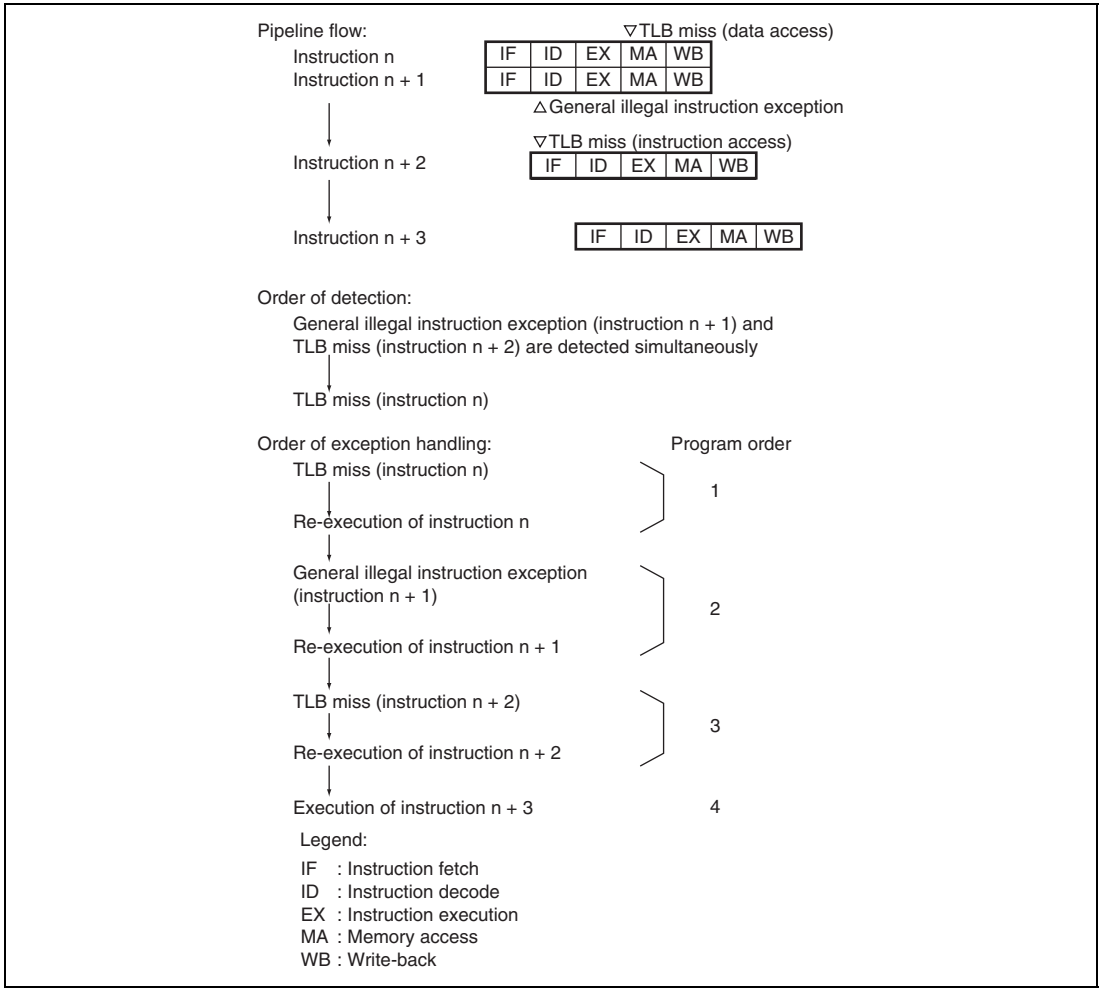


Figure 8.2 Example of General Exception Acceptance Order

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

When the BL bit in SR is 1 and an exception other than a user break is generated, the CPU's internal registers and the registers of the other modules are set to their states following a manual reset, and the CPU branches to the same address as in a reset (H'A000 0000). For the operation in the event of a user break, see section 31, User Break Controller (UBC). If an ordinary interrupt occurs, the interrupt request is held pending and is accepted after the BL bit has been cleared to 0 by software. If a nonmaskable interrupt (NMI) occurs, it can be held pending or accepted according to the setting made by software.

Thus, normally, SPC and SSR are saved and then the BL bit in SR is cleared to 0, to enable multiple exception state acceptance.

8.3.4 Return from Exception Handling

The RTE instruction is used to return from exception handling. When the RTE instruction is executed, the SPC contents are restored to PC and the SSR contents to SR, and the CPU returns from the exception handling routine by branching to the SPC address. If SPC and SSR were saved to external memory, set the BL bit in SR to 1 before restoring the SPC and SSR contents and issuing the RTE instruction.

There are three registers related to exception handling. These are allocated to memory, and can be accessed by specifying the P4 address or area 7 address. For details on the addresses of these registers and the state of registers in each operating mode, see section 32, List of Registers.

Table 8.2 Register Configuration (1)

Register Name	Abbrev.	R/W	P4 Address	Area 7 Address	Size	Sync Clock
TRAPA exception register	TRA	R/W	H'FF00 0020	H'1F00 0020	32	Ick
Exception event register	EXPEVT	R/W	H'FF00 0024	H'1F00 0024	32	Ick
Interrupt event register	INTEVT	R/W	H'FF00 0028	H'1F00 0028	32	Ick

Table 8.2 Register Configuration (2)

Register Name	Abbrev.	Power-on Reset by $\overline{\text{RESET}}$ Pin/WDT/ H-UDI	Manual Reset by $\overline{\text{RESET}}$ Pin/WDT/ Multiple Exception	Sleep by Sleep Instruction/ Deep Sleep	Standby by Hardware	Standby by Software/ Each Module
TRAPA exception register	TRA	Undefined	Undefined	Retained	*	Retained
Exception event register	EXPEVT	H'0000 0000	H'0000 0020	Retained		Retained
Interrupt event register	INTEVT	Undefined	Undefined	Retained		Retained

Note: * After exiting hardware standby mode, this LSI enters the power-on reset state caused by the $\overline{\text{RESET}}$ pin.

EXPEVT consists of 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.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
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
	-	-	-	-												
Initial value:	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

Bit	Bit Name	Initial Value	R/W	Description
31 to 12	—	All 0	R	Reserved These bits are always read as 0, and the write value should always be 0.
11 to 0		*	R/W	12-bit exception code

Note: * H'000 is set in a power-on reset, and H'020 in a manual reset.

8.4.2 Interrupt Event Register (INTEVT)

INTEVT consists of a 14-bit interrupt exception code. The interrupt exception code is set automatically by hardware when an exception occurs. INTEVT can also be modified by software.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
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
	-	-														
Initial value:	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/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 14	—	All 0	R	Reserved These bits are always read as 0, and the write value should always be 0.
13 to 0		—	R/W	14-bit interrupt exception code

TRA consists of 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.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
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
	-	-	-	-	-	-	imm								-	-
Initial value:	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	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 10	—	All 0	R	Reserved These bits are always read as 0, and the write value should always be 0.
9 to 2	imm	—	R/W	8-bit immediate data
1, 0	—	All 0	R	Reserved These bits are always read as 0, and the write value should always be 0.

8.5.1 Resets

(1) Power-On Reset

- Sources:
  - — $\overline{\text{RESET}}$ pin low level
  - — When the watchdog timer overflows while the WT/$\overline{\text{IT}}$ bit is set to 1 and the RSTS bit is cleared to 0 in WTCSR. For details, see section 13, Watchdog timer (WDT).
- Transition address: H'A000 0000
- Transition operations:

  Exception code H'000 is set in EXPEVT, initialization of VBR and SR is performed, and a branch is made to PC = H'A000 0000.

  In the initialization processing, the VBR register is set to H'0000 0000, and in SR, the MD, RB, and BL bits are set to 1, the FD bit is cleared to 0, and the interrupt mask level bits (IMASK3 to IMASK0) are set to B'1111.

  CPU and on-chip peripheral module initialization is performed. For details, see the register descriptions in the relevant sections. For some CPU functions, the $\overline{\text{TRST}}$ pin and $\overline{\text{RESET}}$ pin must be driven low. It is therefore essential to execute a power-on reset and drive the $\overline{\text{TRST}}$ pin low when powering on.

  If the $\overline{\text{RESET}}$ pin is driven high before the $\overline{\text{MRESET}}$ pin while both these pins are low, a manual reset may occur after the power-on reset operation. The $\overline{\text{RESET}}$ pin must be driven high at the same time as, or after, the $\overline{\text{MRESET}}$ pin.

```
Power_on_reset()
{
    EXPEVT = H'0000 0000;
    VBR = H'0000 0000;
    SR.MD = 1;
    SR.RB = 1;
    SR.BL = 1;
    SR.IMASK = B'1111;
    SR.FD = 0;
    Initialize_CPU();
    Initialize_Module(PowerOn);
    PC = H'A000 0000;
}
```

- Sources:
  - $\overline{\text{MRESET}}$ pin low level and $\overline{\text{RESET}}$ pin high level
  - When a general exception other than a user break occurs while the BL bit is set to 1 in SR
  - When the watchdog timer overflows while the RSTS bit is set to 1 in WTCSR. For details, see section 13, Watchdog Timer (WDT).
- Transition address: H'A000 0000
- Transition operations:

  Exception code H'020 is set in EXPEVT, initialization of VBR and SR is performed, and a branch is made to PC = H'A000 0000.

  In the initialization processing, the VBR register is set to H'0000 0000, and in SR, the MD, RB, and BL bits are set to 1, the FD bit is cleared to 0, and the interrupt mask level bits (IMASK3 to IMASK0) are set to B'1111.

  CPU and on-chip peripheral module initialization is performed. For details, see the register descriptions in the relevant sections.

```
Manual_reset()
{
    EXPEVT = H'0000 0020;
    VBR = H'0000 0000;
    SR.MD = 1;
    SR.RB = 1;
    SR.BL = 1;
    SR.IMASK = B'1111;
    SR.FD = 0;
    Initialize_CPU();
    Initialize_Module(Manual);
    PC = H'A000 0000;
}
```

- Source: SDIR.TI7–TI4 = B'0110 (negation) or B'0111 (assertion)
- Transition address: H'A000 0000
- Transition operations:

  Exception code H'000 is set in EXPEVT, initialization of VBR and SR is performed, and a branch is made to PC = H'A000 0000.

  In the initialization processing, the VBR register is set to H'0000 0000, and in SR, the MD, RB, and BL bits are set to 1, the FD bit is cleared to 0, and the interrupt mask level bits (IMASK3 to IMASK0) are set to B'1111.

  CPU and on-chip peripheral module initialization is performed. For details, see the register descriptions in the relevant sections.

```
H-UDI_reset()
{
    EXPEVT = H'0000 0000;
    VBR = H'0000 0000;
    SR.MD = 1;
    SR.RB = 1;
    SR.BL = 1;
    SR.IMASK = B'1111;
    SR.FD = 0;
    Initialize_CPU();
    Initialize_Module(PowerOn);
    PC = H'A000 0000;
}
```

- Source: Multiple ITLB address matches
- Transition address: H'A000 0000
- Transition operations:

  The virtual address (32 bits) at which this exception occurred is set in TEA, and the corresponding virtual page number (22 bits) is set in PTEH [31:10]. ASID in PTEH indicates the ASID when this exception occurred.

  Exception code H'140 is set in EXPEVT, initialization of VBR and SR is performed, and a branch is made to PC = H'A000 0000.

  In the initialization processing, the VBR register is set to H'0000 0000, and in SR, the MD, RB, and BL bits are set to 1, the FD bit is cleared to 0, and the interrupt mask level bits (IMASK3 to IMASK0) are set to B'1111.

  CPU and on-chip peripheral module initialization is performed in the same way as in a manual reset. For details, see the register descriptions in the relevant sections.

```
TLB_multi_hit()
{
    TEA = EXCEPTION_ADDRESS;
    PTEH.VPN = PAGE_NUMBER;
    EXPEVT = H'0000 0140;
    VBR = H'0000 0000;
    SR.MD = 1;
    SR.RB = 1;
    SR.BL = 1;
    SR.IMASK = B'1111;
    SR.FD = 0;
    Initialize_CPU();
    Initialize_Module(Manual);
    PC = H'A000 0000;
}
```

- Source: Multiple UTLB address matches
- Transition address: H'A000 0000
- Transition operations:

  The virtual address (32 bits) at which this exception occurred is set in TEA, and the corresponding virtual page number (22 bits) is set in PTEH [31:10]. ASID in PTEH indicates the ASID when this exception occurred.

  Exception code H'140 is set in EXPEVT, initialization of VBR and SR is performed, and a branch is made to PC = H'A000 0000.

  In the initialization processing, the VBR register is set to H'0000 0000, and in SR, the MD, RB, and BL bits are set to 1, the FD bit is cleared to 0, and the interrupt mask level bits (IMASK3 to IMASK0) are set to B'1111.

  CPU and on-chip peripheral module initialization is performed in the same way as in a manual reset. For details, see the register descriptions in the relevant sections.

```
TLB_multi_hit()
{
    TEA = EXCEPTION_ADDRESS;
    PTEH.VPN = PAGE_NUMBER;
    EXPEVT = H'0000 0140;
    VBR = H'0000 0000;
    SR.MD = 1;
    SR.RB = 1;
    SR.BL = 1;
    SR.IMASK = B'1111;
    SR.FD = 0;
    Initialize_CPU();
    Initialize_Module(Manual);
    PC = H'A000 0000;
}
```

(1) Data TLB Miss Exception

- Source: Address mismatch in UTLB address comparison
- Transition address: VBR + H'0000 0400
- Transition operations:

  The virtual address (32 bits) at which this exception occurred is set in TEA, and the corresponding virtual page number (22 bits) is set in PTEH [31:10]. ASID in PTEH indicates the ASID when this exception occurred.

  The PC and SR contents for the instruction at which this exception occurred are saved in SPC and SSR. The R15 contents at this time are saved in SGR.

  Exception code H'040 (for a read access) or H'060 (for a write access) is set in EXPEVT. The BL, MD, and RB bits are set to 1 in SR, and a branch is made to PC = VBR + H'0400.

  To speed up TLB miss processing, the offset is separate from that of other exceptions.

```
Data_TLB_miss_exception()
{
    TEA = EXCEPTION_ADDRESS;
    PTEH.VPN = PAGE_NUMBER;
    SPC = PC;
    SSR = SR;
    SGR = R15;
    EXPEVT = read_access ? H'0000 0040 : H'0000 0060;
    SR.MD = 1;
    SR.RB = 1;
    SR.BL = 1;
    PC = VBR + H'0000 0400;
}
```

- Source: Address mismatch in ITLB address comparison
- Transition address: VBR + H'0000 0400
- Transition operations:

  The virtual address (32 bits) at which this exception occurred is set in TEA, and the corresponding virtual page number (22 bits) is set in PTEH [31:10]. ASID in PTEH indicates the ASID when this exception occurred.

  The PC and SR contents for the instruction at which this exception occurred are saved in SPC and SSR. The R15 contents at this time are saved in SGR.

  Exception code H'040 is set in EXPEVT. The BL, MD, and RB bits are set to 1 in SR, and a branch is made to PC = VBR + H'0400.

  To speed up TLB miss processing, the offset is separate from that of other exceptions.

```
ITLB_miss_exception()
{
    TEA = EXCEPTION_ADDRESS;
    PTEH.VPN = PAGE_NUMBER;
    SPC = PC;
    SSR = SR;
    SGR = R15;
    EXPEVT = H'0000 0040;
    SR.MD = 1;
    SR.RB = 1;
    SR.BL = 1;
    PC = VBR + H'0000 0400;
}
```

- Source: TLB is hit in a store access, but dirty bit D = 0
- Transition address: VBR + H'0000 0100
- Transition operations:

  The virtual address (32 bits) at which this exception occurred is set in TEA, and the corresponding virtual page number (22 bits) is set in PTEH [31:10]. ASID in PTEH indicates the ASID when this exception occurred.

  The PC and SR contents for the instruction at which this exception occurred are saved in SPC and SSR. The R15 contents at this time are saved in SGR.

  Exception code H'080 is set in EXPEVT. The BL, MD, and RB bits are set to 1 in SR, and a branch is made to PC = VBR + H'0100.

```
Initial_write_exception()
{
    TEA = EXCEPTION_ADDRESS;
    PTEH.VPN = PAGE_NUMBER;
    SPC = PC;
    SSR = SR;
    SGR = R15;
    EXPEVT = H'0000 0080;
    SR.MD = 1;
    SR.RB = 1;
    SR.BL = 1;
    PC = VBR + H'0000 0100;
}
```

- Source: The access does not accord with the UTLB protection information (PR bits) shown below.

PR	Privileged Mode	User Mode
00	Only read access possible	Access not possible
01	Read/write access possible	Access not possible
10	Only read access possible	Only read access possible
11	Read/write access possible	Read/write access possible

- Transition address: VBR + H'0000 0100
- Transition operations:

  The virtual address (32 bits) at which this exception occurred is set in TEA, and the corresponding virtual page number (22 bits) is set in PTEH [31:10]. ASID in PTEH indicates the ASID when this exception occurred.

  The PC and SR contents for the instruction at which this exception occurred are saved in SPC and SSR. The R15 contents at this time are saved in SGR.

  Exception code H'0A0 (for a read access) or H'0C0 (for a write access) is set in EXPEVT. The BL, MD, and RB bits are set to 1 in SR, and a branch is made to PC = VBR + H'0100.

```
Data_TLB_protection_violation_exception()
{
    TEA = EXCEPTION_ADDRESS;
    PTEH.VPN = PAGE_NUMBER;
    SPC = PC;
    SSR = SR;
    SGR = R15;
    EXPEVT = read_access ? H'0000 00A0 : H'0000 00C0;
    SR.MD = 1;
    SR.RB = 1;
    SR.BL = 1;
    PC = VBR + H'0000 0100;
}
```

- Source: The access does not accord with the ITLB protection information (PR bits) shown below.

PR	Privileged Mode	User Mode
0	Access possible	Access not possible
1	Access possible	Access possible

- Transition address: VBR + H'0000 0100
- Transition operations:

  The virtual address (32 bits) at which this exception occurred is set in TEA, and the corresponding virtual page number (22 bits) is set in PTEH [31:10]. ASID in PTEH indicates the ASID when this exception occurred.

  The PC and SR contents for the instruction at which this exception occurred are saved in SPC and SSR. The R15 contents at this time are saved in SGR.

  Exception code H'0A0 is set in EXPEVT. The BL, MD, and RB bits are set to 1 in SR, and a branch is made to PC = VBR + H'0100.

```
ITLB_protection_violation_exception()
{
    TEA = EXCEPTION_ADDRESS;
    PTEH.VPN = PAGE_NUMBER;
    SPC = PC;
    SSR = SR;
    SGR = R15;
    EXPEVT = H'0000 00A0;
    SR.MD = 1;
    SR.RB = 1;
    SR.BL = 1;
    PC = VBR + H'0000 0100;
}
```

- Sources:
  - Word data access from other than a word boundary (2n +1)
  - Longword data access from other than a longword data boundary (4n +1, 4n + 2, or 4n +3)
  - Quadword data access from other than a quadword data boundary (8n +1, 8n + 2, 8n +3, 8n + 4, 8n + 5, 8n + 6, or 8n + 7)
  - Access to area H'8000 0000–H'FFFF FFFF in user mode
- Transition address: VBR + H'0000 0100
- Transition operations:

  The virtual address (32 bits) at which this exception occurred is set in TEA, and the corresponding virtual page number (22 bits) is set in PTEH [31:10]. ASID in PTEH indicates the ASID when this exception occurred.

  The PC and SR contents for the instruction at which this exception occurred are saved in SPC and SSR. The R15 contents at this time are saved in SGR.

  Exception code H'0E0 (for a read access) or H'100 (for a write access) is set in EXPEVT. The BL, MD, and RB bits are set to 1 in SR, and a branch is made to PC = VBR + H'0100. For details, see section 6, Memory Management Unit (MMU).

```
Data_address_error()
{
    TEA = EXCEPTION_ADDRESS;
    PTEH.VPN = PAGE_NUMBER;
    SPC = PC;
    SSR = SR;
    SGR = R15;
    EXPEVT = read_access? H'0000 00E0: H'0000 0100;
    SR.MD = 1;
    SR.RB = 1;
    SR.BL = 1;
    PC = VBR + H'0000 0100;
}
```

- Sources:
  - Instruction fetch from other than a word boundary (2n +1)
  - Instruction fetch from area H'8000 0000–H'FFFF FFFF in user mode
- Transition address: VBR + H'0000 0100
- Transition operations:

  The virtual address (32 bits) at which this exception occurred is set in TEA, and the corresponding virtual page number (22 bits) is set in PTEH [31:10]. ASID in PTEH indicates the ASID when this exception occurred.

  The PC and SR contents for the instruction at which this exception occurred are saved in the SPC and SSR. The R15 contents at this time are saved in SGR.

  Exception code H'0E0 is set in EXPEVT. The BL, MD, and RB bits are set to 1 in SR, and a branch is made to PC = VBR + H'0100. For details, see section 6, Memory Management Unit (MMU).

```
Instruction_address_error()
{
    TEA = EXCEPTION_ADDRESS;
    PTEH.VPN = PAGE_NUMBER;
    SPC = PC;
    SSR = SR;
    SGR = R15;
    EXPEVT = H'0000 00E0;
    SR.MD = 1;
    SR.RB = 1;
    SR.BL = 1;
    PC = VBR + H'0000 0100;
}
```

- Source: Execution of TRAPA instruction
- Transition address: VBR + H'0000 0100
- Transition operations:

  As this is a processing-completion-type exception, the PC contents for the instruction following the TRAPA instruction are saved in SPC. The value of SR and R15 when the TRAPA instruction is executed are saved in SSR and SGR. The 8-bit immediate value in the TRAPA instruction is multiplied by 4, and the result is set in TRA [9:0]. Exception code H'160 is set in EXPEVT. The BL, MD, and RB bits are set to 1 in SR, and a branch is made to PC = VBR + H'0100.

```
TRAPA_exception()
{
    SPC = PC + 2;
    SSR = SR;
    SGR = R15;
    TRA = imm << 2;
    EXPEVT = H'0000 0160;
    SR.MD = 1;
    SR.RB = 1;
    SR.BL = 1;
    PC = VBR + H'0000 0100;
}
```

- Sources:
  - Decoding of an undefined instruction not in a delay slot

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

    Undefined instruction: H'FFFD
  - Decoding in user mode of a privileged instruction not in a delay slot

    Privileged instructions: LDC, STC, RTE, LDTLB, SLEEP, but excluding LDC/STC instructions that access GBR
- Transition address: VBR + H'0000 0100
- Transition operations:

  The PC and SR contents for the instruction at which this exception occurred are saved in SPC and SSR. The R15 contents at this time are saved in SGR.

  Exception code H'180 is set in EXPEVT. The BL, MD, and RB bits are set to 1 in SR, and a branch is made to PC = VBR + H'0100. Operation is not guaranteed if an undefined code other than H'FFFD is decoded.

```
General_illegal_instruction_exception()
{
    SPC = PC;
    SSR = SR;
    SGR = R15;
    EXPEVT = H'0000 0180;
    SR.MD = 1;
    SR.RB = 1;
    SR.BL = 1;
    PC = VBR + H'0000 0100;
}
```

- Sources:
  - — Decoding of an undefined instruction in a delay slot

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

    Undefined instruction: H'FFFD
  - — Decoding of an instruction that modifies PC in a delay slot

    Instructions that modify PC: JMP, JSR, BRA, BRAF, BSR, BSRF, RTS, RTE, BT, BF, BT/S, BF/S, TRAPA, LDC Rm,SR, LDC.L @Rm+,SR
  - — Decoding in user mode of a privileged instruction in a delay slot

    Privileged instructions: LDC, STC, RTE, LDTLB, SLEEP, but excluding LDC/STC instructions that access GBR
  - — Decoding of a PC-relative MOV instruction or MOVA instruction in a delay slot
- Transition address: VBR + H'0000 0100
- Transition operations:

  The PC contents for the preceding delayed branch instruction are saved in SPC. The SR and R15 contents when this exception occurred are saved in SSR and SGR.

  Exception code H'1A0 is set in EXPEVT. The BL, MD, and RB bits are set to 1 in SR, and a branch is made to PC = VBR + H'0100. Operation is not guaranteed if an undefined code other than H'FFFD is decoded.

```
Slot_illegal_instruction_exception()
{
    SPC = PC - 2;
    SSR = SR;
    SGR = R15;
    EXPEVT = H'0000 01A0;
    SR.MD = 1;
    SR.RB = 1;
    SR.BL = 1;
    PC = VBR + H'0000 0100;
}
```

- Source: Decoding of an FPU instruction* not in a delay slot with SR.FD =1
- Transition address: VBR + H'0000 0100
- Transition operations:

  The PC and SR contents for the instruction at which this exception occurred are saved in SPC and SSR. The R15 contents at this time are saved in SGR.

  Exception code H'800 is set in EXPEVT. The BL, MD, and RB bits are set to 1 in SR, and a branch is made to PC = VBR + H'0100.

Note: * FPU instructions are instructions in which the first 4 bits of the instruction code are F (but excluding undefined instruction H'FFFD), and the LDS, STS, LDS.L, and STS.L instructions corresponding to FPUL and FPSCR.

```
General_fpu_disable_exception()
{
    SPC = PC;
    SSR = SR;
    SGR = R15;
    EXPEVT = H'0000 0800;
    SR.MD = 1;
    SR.RB = 1;
    SR.BL = 1;
    PC = VBR + H'0000 0100;
}
```

- Source: Decoding of an FPU instruction in a delay slot with SR.FD =1
- Transition address: VBR + H'0000 0100
- Transition operations:

  The PC contents for the preceding delayed branch instruction are saved in SPC. The SR and R15 contents when this exception occurred are saved in SSR and SGR.

  Exception code H'820 is set in EXPEVT. The BL, MD, and RB bits are set to 1 in SR, and a branch is made to PC = VBR + H'0100.

```
Slot_fpu_disable_exception()
{
    SPC = PC - 2;
    SSR = SR;
    SGR = R15;
    EXPEVT = H'0000 0820;
    SR.MD = 1;
    SR.RB = 1;
    SR.BL = 1;
    PC = VBR + H'0000 0100;
}
```

- Source: Fulfilling of a break condition set in the user break controller
- Transition address: VBR + H'0000 0100, or DBR
- Transition operations:

  In the case of a post-execution break, the PC contents for the instruction following the instruction at which the breakpoint is set are set in SPC. In the case of a pre-execution break, the PC contents for the instruction at which the breakpoint is set are set in SPC.

  The SR and R15 contents when the break occurred are saved in SSR and SGR. Exception code H'1E0 is set in EXPEVT.

  The BL, MD, and RB bits are set to 1 in SR, and a branch is made to PC = VBR + H'0100. It is also possible to branch to PC = DBR.

  For details of PC, etc., when a data break is set, see section 31, User Break Controller (UBC).

```
User_break_exception()
{
    SPC = (pre_execution break? PC : PC + 2);
    SSR = SR;
    SGR = R15;
    EXPEVT = H'0000 01E0;
    SR.MD = 1;
    SR.RB = 1;
    SR.BL = 1;
    PC = (BRCR.UBDE==1 ? DBR : VBR + H'0000 0100);
}
```

- Source: Exception due to execution of a floating-point operation
- Transition address: VBR + H'0000 0100
- Transition operations:

  The PC and SR contents for the instruction at which this exception occurred are saved in SPC and SSR . The R15 contents at this time are saved in SGR. Exception code H'120 is set in EXPEVT. The BL, MD, and RB bits are set to 1 in SR, and a branch is made to PC = VBR + H'0100.

```
FPU_exception()
{
    SPC = PC;
    SSR = SR;
    SGR = R15;
    EXPEVT = H'0000 0120;
    SR.MD = 1;
    SR.RB = 1;
    SR.BL = 1;
    PC = VBR + H'0000 0100;
}
```

(1) NMI

- Source: NMI pin edge detection
- Transition address: VBR + H'0000 0600
- Transition operations:

  The PC and SR contents for the instruction at which this exception is accepted are saved in SPC and SSR. The R15 contents at this time are saved in SGR.

  Exception code H'1C0 is set in INTEVT. The BL, MD, and RB bits are set to 1 in SR, and a branch is made to PC = VBR + H'0600. When the BL bit in SR is 0, this interrupt is not masked by the interrupt mask bits in SR, and is accepted at the highest priority level. When the BL bit in SR is 1, a software setting can specify whether this interrupt is to be masked or accepted. For details, see section 9, Interrupt Controller (INTC).

```
NMI()
{
    SPC = PC;
    SSR = SR;
    SGR = R15;
    INTEVT = H'0000 01C0;
    SR.MD = 1;
    SR.RB = 1;
    SR.BL = 1;
    PC = VBR + H'0000 0600;
}
```

- Source: The interrupt mask level bits (IMASK3 to IMASK0) setting in SR is smaller than the IRL (3–0) level, and the BL bit in SR is 0 (accepted at instruction boundary).
- Transition address: VBR + H'0000 0600
- Transition operations:

  The PC contents immediately after the instruction at which the interrupt is accepted are set in SPC. The SR and R15 contents at the time of acceptance are set in SSR and SGR.

  The code corresponding to the IRL (3–0) level is set in INTEVT. See table 9.7, for the corresponding codes. The BL, MD, and RB bits are set to 1 in SR, and a branch is made to VBR + H'0600. The acceptance level is not set in the interrupt mask level bits (IMASK3 to IMASK0) in SR. When the BL bit in SR is 1, the interrupt is masked. For details, see section 9, Interrupt Controller (INTC).

```
IRL()
{
    SPC = PC;
    SSR = SR;
    SGR = R15;
    INTEVT = H'0000 0200 – H'0000 03C0;
    SR.MD = 1;
    SR.RB = 1;
    SR.BL = 1;
    PC = VBR + H'0000 0600;
}
```

- Source: The interrupt mask level bits (IMASK3 to IMASK0) setting in SR is smaller than the peripheral module (DMAC, IRQ, HCAN2, SSI, HAC, I^2C, USB, LCDC, DMABRG, SCIF, SIM, HSPI, MMCIF, MFI, H-UDI, ADC, CMT, TMU, WDT, or REF) interrupt level, and the BL bit in SR is 0 (accepted at instruction boundary).
- Transition address: VBR + H'0000 0600
- Transition operations:

  The PC contents immediately after the instruction at which the interrupt is accepted are set in SPC. The SR and R15 contents at the time of acceptance are set in SSR and SGR.

  The code corresponding to the interrupt source is set in INTEVT. The BL, MD, and RB bits are set to 1 in SR, and a branch is made to VBR + H'0600. The module interrupt levels should be set as values between B'0000 and B'1111 in the interrupt priority level setting registers (IPRA–IPRC) in the interrupt controller. For details, see section 9, Interrupt Controller (INTC).

```
Module_interruption()
{
    SPC = PC;
    SSR = SR;
    SGR = R15;
    INTEVT = H'0000 0400 – H'0000 0FA0;
    SR.MD = 1;
    SR.RB = 1;
    SR.BL = 1;
    PC = VBR + H'0000 0600;
}
```

- Source: The interrupt mask bit setting in SR is smaller than the IRL (3–0) level, and the BL bit in SR is 0 (accepted at instruction boundary).
- Transition address: VBR + H'0000 0600
- Transition operations:

  The PC contents immediately after the instruction at which the interrupt is accepted are set in SPC. The SR and R15 contents at the time of acceptance are set in SSR and SGR.

  The code corresponding to the interrupt source is set in INTEVT. The BL, MD, and RB bits are set to 1 in SR, and a branch is made to VBR + H'0600. The IRQ interrupt levels should be set as values between B'0000 and B'1111 in the interrupt priority level setting register 00 (INTPRI00) in the interrupt controller. For details, see section 9, Interrupt Controller (INTC).

```
IRQ()
{
    SPC = PC;
    SSR = SR;
    SGR = R15;
    INTEVT = H'0000 0800 – H'0000 0860;
    SR.MD = 1;
    SR.RB = 1;
    SR.BL = 1;
    PC = VBR + H'0000 0600;
}
```

With some instructions, such as instructions that make two accesses to memory, and the indivisible pair comprising a delayed branch instruction and delay slot instruction, multiple exceptions occur. Care is required in these cases, as the exception priority order differs from the normal order.

- Instructions that make two accesses to memory

  With MAC instructions, memory-to-memory arithmetic/logic instructions, and TAS instructions, two data transfers are performed by a single instruction, and an exception will be detected for each of these data transfers. In these cases, therefore, the following order is used to determine priority.

  1. Data address error in first data transfer
  2. TLB miss in first data transfer
  3. TLB protection violation in first data transfer
  4. Initial page write exception in first data transfer
  5. Data address error in second data transfer
  6. TLB miss in second data transfer
  7. TLB protection violation in second data transfer
  8. Initial page write exception in second data transfer

- Indivisible delayed branch instruction and delay slot instruction

  As a delayed branch instruction and its associated delay slot instruction are indivisible, they are treated as a single instruction. Consequently, the priority order for exceptions that occur in these instructions differs from the usual priority order. The priority order shown below is for the case where the delay slot instruction has only one data transfer.

  1. A check is performed for the interrupt type and re-execution type exceptions of priority levels 1 and 2 in the delayed branch instruction.
  2. A check is performed for the interrupt type and re-execution type exceptions of priority levels 1 and 2 in the delay slot instruction.
  3. A check is performed for the completion type exception of priority level 2 in the delayed branch instruction.
  4. A check is performed for the completion type exception of priority level 2 in the delay slot instruction.
  5. A check is performed for priority level 3 in the delayed branch instruction and priority level 3 in the delay slot instruction. (There is no priority ranking between these two.)
  6. A check is performed for priority level 4 in the delayed branch instruction and priority level 4 in the delay slot instruction. (There is no priority ranking between these two.)

the above case (instructions that make two accesses to memory).

If the accepted exception (the highest-priority exception) is a delay slot instruction re-execution type exception, the branch instruction PR register write operation (PC → PR operation performed in a BSR, BSRF, or JSR instruction) is not disabled.

8.6 Usage Notes

1. Return from exception handling
   A. Check the BL bit in SR with software. If SPC and SSR have been saved to external memory, set the BL bit in SR to 1 before restoring them.
   B. Issue an RTE instruction. When RTE is executed, the SPC contents are saved in PC, the SSR contents are saved in SR, and branch is made to the SPC address to return from the exception handling routine.
2. If an exception or interrupt occurs when BL bit in SR = 1
   A. Exception

      When an exception other than a user break occurs, a manual reset is executed. The value in EXPEVT at this time is H'0000 0020; the SPC and SSR contents are undefined.
   B. Interrupt

      If an ordinary interrupt occurs, the interrupt request is held pending and is accepted after the BL bit in SR has been cleared to 0 by software. If a nonmaskable interrupt (NMI) occurs, it can be held pending or accepted according to the setting made by software.

      In sleep or standby mode, however, an interrupt is accepted even if the BL bit in SR is set to 1.
3. SPC when an exception occurs
   A. Re-execution type exception

      The PC value for the instruction at which the exception occurred is set in SPC, and the instruction is re-executed after returning from the exception handling routine. If an exception occurs in a delay slot instruction, however, the PC value for the delay slot instruction is saved in SPC regardless of whether or not the preceding delay slot instruction condition is satisfied.
   B. Completion type exception or interrupt

      The PC value for the instruction following that at which the exception occurred is set in SPC. If an exception occurs in a branch instruction with delay slot, however, the PC value for the branch destination is saved in SPC.
4. An exception must not be generated in an RTE instruction delay slot, as the operation cannot be guaranteed in this case.

8.7.1 Restrictions on First Instruction in Exception Handling Routine

- Do not locate a BT, BF, BT/S, BF/S, BRA, or BSR instruction at address VBR + H'100, VBR + H'400, or VBR + H'600.
- When the UBDE bit in BRCR is set to 1 and the user break debug support function* is used, do not locate a BT, BF, BT/S, BF/S, BRA, or BSR instruction at the address indicated by DBR.

Note: * See section 31.5, User Break Debug Support Function.

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 handle interrupt requests according to user-set priority.

9.1 Features

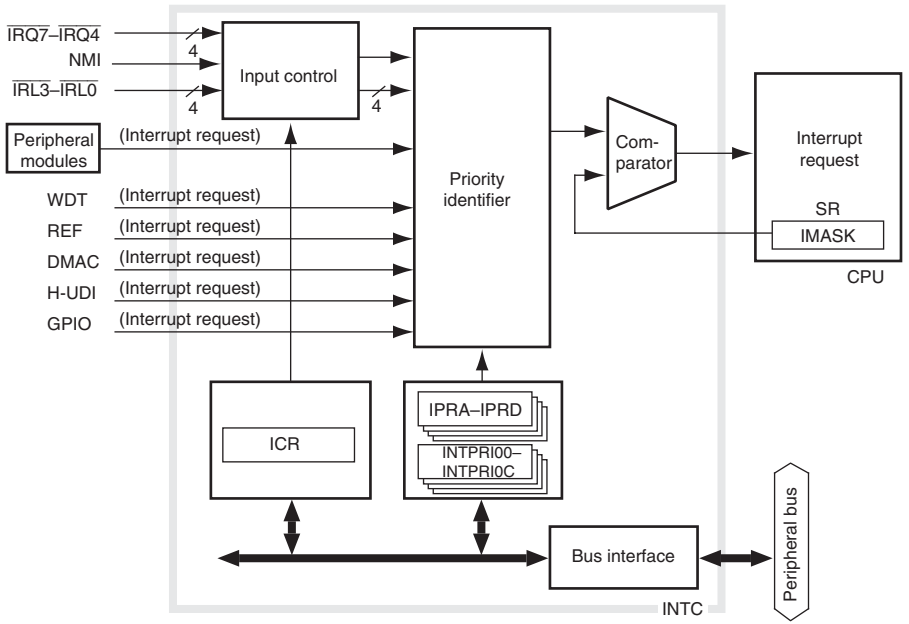
The INTC has the following features.

- Fifteen interrupt priority levels can be set

  By setting the eight interrupt priority level setting registers, the priorities of peripheral module interrupts can be selected from 15 levels for different request sources.
- NMI noise canceler function

  The NMI input level bit indicates the NMI pin state. The pin state can be checked by reading this bit in the interrupt exception handling routine, enabling it to be used as a noise canceler.
- NMI request masking when the BL bit in SR is set to 1.

  It is possible to select whether or not NMI requests are to be masked when the BL bit in SR is set to 1.



Note : Peripheral modules: The following 14 modules can output interrupt requests.
The number of channel is enclosed within parentheses.

HCAN2 (2)	: Controller Area Network 2
SSI (2)	: Serial Sound Interface
HAC (2)	: Audio Codec Interface
I^2C (2)	: I^2C Bus Interface
USB	: USB Host
LCDC	: LCD Controller
SCIF (3)	: Serial Communication Interface with FIFO
SIM	: SIM Card Module
HSPI	: Serial Peripheral Interface
MMCIF	: Multimedia Card Interface
MFI	: Multifunctional Interface
ADC	: A/D Converter
CMT	: Compare Match Timer
TMU (3)	: Timer Unit

Legend:

WDT	: Watchdog timer
REF	: Memory refresh controller section of the bus state controller
DMAC	: Direct memory access controller
H-UDI	: User debug interface
GPIO	: General I/O ports
ICR	: Interrupt control register
IPRA–IPRD	: Interrupt priority level setting registers A–D
INTPRI00-0C	: Interrupt priority level setting registers 00-0C
SR	: Status register

Figure 9.1 Block Diagram of INTC

Table 9.1 shows the INTC pin configuration.

Table 9.1 Pin Configuration

Pin Name	Abbreviation	I/O	Function
Nonmaskable interrupt input pin	NMI	Input	Input of nonmaskable interrupt request signal
IRL interrupt input pins	$\overline{\text{IRL3}}$ to $\overline{\text{IRL0}}$	Input	Input of IRL interrupt request signals (maskable by the IMASK bits in SR)
IRQ interrupt input pins	$\overline{\text{IRQ7}}$ to $\overline{\text{IRQ4}}$	Input	Input of IRQ interrupt request signals

9.3 Register Descriptions

The INTC has the following registers. For details on the addresses of these registers and the state of registers in each operating mode, see section 32, List of Registers.

Table 9.2 Register Configuration (1)

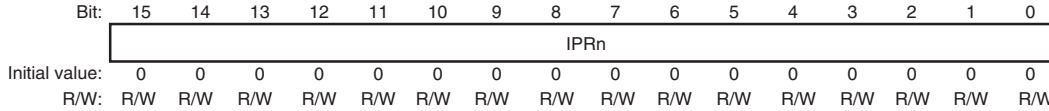
Register Name	Abbrev.	R/W	P4 Address	Area 7 Address	Size	Sync Clock
Interrupt control register	ICR	R/W	H'FFD0 0000	H'1FD0 0000	16	Pck
Interrupt priority level setting register A	IPRA	R/W	H'FFD0 0004	H'1FD0 0004	16	Pck
Interrupt priority level setting register B	IPRB	R/W	H'FFD0 0008	H'1FD0 0008	16	Pck
Interrupt priority level setting register C	IPRC	R/W	H'FFD0 000C	H'1FD0 000C	16	Pck
Interrupt priority level setting register D	IPRD	R/W	H'FFD0 0010	H'1FD0 0010	16	Pck
Interrupt priority level setting register 00	INTPRI00	R/W	H'FE08 0000	H'1E08 0000	32	Pck
Interrupt priority level setting register 04	INTPRI04	R/W	H'FE08 0004	H'1E08 0004	32	Pck
Interrupt priority level setting register 08	INTPRI08	R/W	H'FE08 0008	H'1E08 0008	32	Pck
Interrupt priority level setting register 0C	INTPRI0C	R/W	H'FE08 000C	H'1E08 000C	32	Pck
Interrupt source register 00	INTREQ00	R	H'FE08 0020	H'1E08 0020	32	Pck
Interrupt source register 04	INTREQ04	R	H'FE08 0024	H'1E08 0024	32	Pck
Interrupt mask register 00	INTMSK00	R/W	H'FE08 0040	H'1E08 0040	32	Pck
Interrupt mask register 04	INTMSK04	R/W	H'FE08 0044	H'1E08 0044	32	Pck
Interrupt mask clear register 00	INTMSK CLR00	W	H'FE08 0060	H'1E08 0060	32	Pck
Interrupt mask clear register 04	INTMSK CLR04	W	H'FE08 0064	H'1E08 0064	32	Pck

Register Name	Abbrev.	Power-on Reset by RESET Pin/WDT/ H-UDI	Manual Reset by RESET Pin/WDT/ Multiple Exception	Sleep by Sleep Instruction/ Deep Sleep	Standby by Hardware	Standby by Software/ Each Module
Interrupt control register	ICR	H'0000*1	H'0000*1	Retained	*3	Retained
		H'8000*2	H'8000*2	Retained		Retained
Interrupt priority level setting register A	IPRA	H'0000	H'0000	Retained		Retained
Interrupt priority level setting register B	IPRB	H'0000	H'0000	Retained		Retained
Interrupt priority level setting register C	IPRC	H'0000	H'0000	Retained		Retained
Interrupt priority level setting register D	IPRD	H'DA74	H'DA74	Retained		Retained
Interrupt priority level setting register 00	INTPRI00	H'0000 0000	Retained	Retained		Retained
Interrupt priority level setting register 04	INTPRI04	H'0000 0000	Retained	Retained		Retained
Interrupt priority level setting register 08	INTPRI08	H'0000 0000	Retained	Retained		Retained
Interrupt priority level setting register 0C	INTPRI0C	H'0000 0000	Retained	Retained		Retained
Interrupt source register 00	INTREQ00	H'0000 0000	Retained	Retained		Retained
Interrupt source register 04	INTREQ04	H'0000 0000	Retained	Retained		Retained
Interrupt mask register 00	INTMSK00	H'F3FF 7FFF	Retained	Retained		Retained
Interrupt mask register 04	INTMSK04	H'00FF FFFF	Retained	Retained		Retained
Interrupt mask clear register 00	INTMSK CLR00	—	—	—		—
Interrupt mask clear register 04	INTMSK CLR04	—	—	—		—

Notes: 1. The NMI pin is a low level.
2. The NMI pin is a high level.
3. After exiting hardware standby mode, this LSI enters the power-on reset state by the RESET pin.

IPRA to IPRD are 16-bit readable/writable registers that set priority levels from 15 to 0 for the peripheral module interrupts.

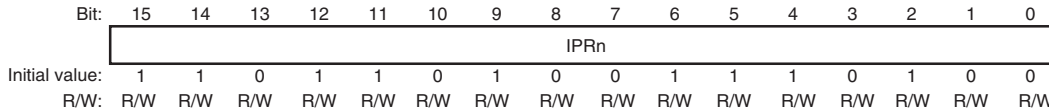
- IPRA to IPRC



Bit	Bit Name	Initial Value	R/W	Description
15 to 0	IPRn	All 0	R/W	These bits set the priority level for each interrupt source in 4-bit units. For details, see table 9.3, Interrupt Request Sources and IPRA to IPRD.

Note: n = 15 to 0

- IPRD



Bit	Bit Name	Initial Value	R/W	Description
15 to 0	IPRn	H'DA74	R/W	These bits set the priority level for each interrupt source in 4-bit units. For details, see table 9.3, Interrupt Request Sources and IPRA to IPRD.

Note: n = 15 to 0

Register	Bits			
	15 to 12	11 to 8	7 to 4	3 to 0
IPRA	TMU0	TMU1	TMU2	Reserved*2
IPRB	WDT	REF*1	Reserved*2	Reserved*2
IPRC	GPIO (IRL, IRQ)*3	DMAC	Reserved*2	H-UDI
IPRD	IRL0	IRL1	IRL2	IRL3

Notes: 1. REF is the memory refresh control unit in the bus state controller (BSC). See section 10, Bus State Controller (BSC), for details.
2. Reserved: These bits are always read as 0. The write value should always be 0.
3. These bits set the interrupt priority level of the GPIO in normal mode and that of the GPIO, IRL, IRQ4, and IRQ5 in software standby mode. To exit software standby mode by an IRQ4 or IRQ5 interrupt, INTPRI00 must also be set. The same value must be set in both registers. Note that software standby mode cannot be exited by an IRQ6 or IRQ7 interrupt.

As shown in table 9.3, four peripheral modules are assigned to each register. Setting a value from H'F (1111) to H'0 (0000) in each of the 4-bit groups, 15 to 12, 11 to 8, 7 to 4, and 3 to 0, configures interrupt priority level for each group. Setting H'F designates priority level 15 (the highest level), and setting H'0 designates priority level 0 (requests are masked).

9.3.2 Interrupt Priority Level Setting Registers 00 to 0C (INTPRI00 to INTPRI0C)

INTPRI00 to INTPRI0C are 32-bit readable/writable registers that set priority levels from 15 to 0 for the peripheral module interrupts.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	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
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	Bit Name	Initial Value	R/W	Description
31 to 0		All 0	R/W	These bits set the priority level for each interrupt source in 4-bit units. For details, see table 9.4, Interrupt Request Sources and INTPRI00 to INTPRI0C.

	Bits							
Register	**31 to 28**	**27 to 24**	**23 to 20**	**19 to 16**	**15 to 12**	**11 to 8**	**7 to 4**	**3 to 0**
INTPRI00	IRQ4*3	IRQ5*3	IRQ6	IRQ7	Reserved*2	Reserved*2	Reserved*2	Reserved*2
INTPRI04	HCAN2(0)	HCAN2(1)	SSI(0)	SSI(1)	HAC(0)	HAC(1)	I^2C(0)	I^2C(1)
INTPRI08	USB	LCDC	DMABRG	SCIF(0)	SCIF(1)	SCIF(2)	SIM	HSPI
INTPRI0C	Reserved*2	Reserved*2	MMCIF	Reserved*2	MFI	Reserved*2	ADC	CMT

Notes: 1. As shown in table 9.4, eight peripheral modules are assigned to each register. Setting a value from H'F (1111) to H'0 (0000) in each of the 4-bit groups configures interrupt priority level for each group. Setting H'F designates priority level 15 (the highest level), and setting H'0 designates priority level 0 (requests are masked).

2. Reserved: These bits are always read as 0. The write value should always be 0.
3. To enable an IRQ4 or IRQ5 interrupt in software standby mode, setting must be made in this register as well as in IPRC. The same value must be set in both registers. Note that software standby mode cannot be exited by an IRQ6 or IRQ7 interrupt.

9.3.3 Interrupt Control Register (ICR)

ICR sets the input signal detection mode for external interrupt input pin NMI and indicates the input signal level at the NMI pin.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	NMIL	MAI	-	-	-	-	NMIB	NMIE	IRLM	-	-	-	-	-	-	-
Initial value:	0/1*	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	-	-	-	-	R/W	R/W	R/W	-	-	-	-	-	-	-

Bit	**Bit Name**	**Initial Value**	**R/W**	**Description**
15	NMIL	0/1*	R	NMI Input Level Sets the level of the signal input at the NMI pin. This bit can be read to determine the NMI pin level. It cannot be modified. 0: NMI pin input level is low 1: NMI pin input level is high

14	MAI	0	R/W	NMI Interrupt Mask Specifies whether or not all interrupts are to be masked while the NMI pin input level is low, irrespective of the BL bit in SR of the CPU. NMI interrupts are accepted in normal operation and in sleep mode. In standby mode, all interrupts are masked, and standby is not cleared, while the NMI pin is low. 0: Interrupts enabled even while NMI pin is low 1: Interrupts disabled while NMI pin is low
13 to 10	—	All 0	—	Reserved These bits are always read as 0.The write value should always be 0.
9	NMIB	0	R/W	NMI Block Mode Specifies whether an NMI request is to be held pending or detected immediately while the BL bit in SR of the CPU is set to 1. If interrupt requests are enabled while the BL bit is 1, the previous exception information will be lost, and so must be saved beforehand. This bit is cleared automatically by NMI acceptance. 0: NMI interrupt requests held pending while the BL bit in SR is set to 1 1: NMI interrupt requests detected while the BL bit in SR is set to 1
8	NMIE	0	R/W	NMI Edge Select Specifies whether the falling or rising edge of the interrupt request signal to the NMI pin is detected. 0: Interrupt request detected on falling edge of NMI input 1: Interrupt request detected on rising edge of NMI input
7	IRLM	0	R/W	IRL Pin Mode Specifies whether pins $\overline{\text{IRL3}}$ to $\overline{\text{IRL0}}$ are to be used as level-encoded interrupt requests or as four independent interrupt requests. 0: $\overline{\text{IRL}}$ pins used as level-encoded interrupt requests 1: $\overline{\text{IRL}}$ pins used as four independent interrupt requests (level-sensing IRQ mode)

6 to 0	—	All 0	—	Reserved These bits are always read as 0.The write value should always be 0.

Note: * 1 when NMI pin input is high, 0 when low.

9.3.4 Interrupt Source Registers 00, 04 (INTREQ00, INTREQ04)

INTREQ00 and INTREQ04 are 32-bit read-only registers that indicate the origin of the interrupt request sent to the INTC. The states of the bits in these registers are not affected by masking of the corresponding interrupts by the settings in INTPRI00 and INTPRI04.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
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
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	Bit Name	Initial Value	R/W	Description
31 to 0		All 0	R	Interrupt Requests 31 to 0 Each bit indicates that there is an interrupt request relevant to that bit. For the correspondence between the bits and interrupt sources, see table 9.5. 0: There is no interrupt request that corresponds to this bit 1: There is an interrupt request that corresponds to this bit.

- INTREQ00, INTMSK00, and INTMSKCLR00

Bit Number	Module	Interrupt	Bit Number	Module	Interrupt
31	IRQ	IRQ4	15	—	—
30	IRQ	IRQ5	14	DMABRG	DMABRGI0
29	IRQ	IRQ6	13	DMABRG	DMABRGI1
28	IRQ	IRQ7	12	DMABRG	DMABRGI2
27	—	—	11	SCIF(0)	ERI0
26	—	—	10	SCIF(0)	RXI0
25	HCAN2(0)	HCANI0	9	SCIF(0)	BRI0
24	HCAN2(1)	HCANI1	8	SCIF(0)	TXI0
23	SSI(0)	SSII0	7	SCIF(1)	ERI1
22	SSI(1)	SSII1	6	SCIF(1)	RXI1
21	HAC(0)	HACI0	5	SCIF(1)	BRI1
20	HAC(1)	HACI1	4	SCIF(1)	TXI1
19	$I^2C(0)$	IICI0	3	SCIF(2)	ERI2
18	$I^2C(1)$	IICI1	2	SCIF(2)	RXI2
17	USB	USBI	1	SCIF(2)	BRI2
16	LCDC	VINT	0	SCIF(2)	TXI2

- INTREQ04, INTMSK04, and INTMSKCLR04

Bit Number	Module	Interrupt	Bit Number	Module	Interrupt
31	—	—	15	MMCIF	MMCI3
30	—	—	14	—	—
29	—	—	13	—	—
28	—	—	12	—	—
27	—	—	11	—	—
26	—	—	10	—	—
25	—	—	9	—	—
24	—	—	8	—	—
23	SIM	SIMERI	7	—	—
22	SIM	SIMRXI	6	MFI	MFII
21	SIM	SIMTXI	5	—	—
20	SIM	SIMTEI	4	—	—
19	HSPI	HSPII	3	—	—
18	MMCIF	MMCI0	2	—	—
17	MMCIF	MMCI1	1	ADC	ADI
16	MMCIF	MMCI2	0	CMT	CMTI

9.3.5 Interrupt Mask Registers 00, 04 (INTMSK00, INTMSK04)

INTMSK00 and INTMSK04 are 32-bit registers that set the masking of individual interrupt requests. Writing 1 to the corresponding bits in INTMSK00 and INTMSK04 masks interrupt requests.

To cancel masking of an interrupt, write 1 to the corresponding bit in INTMSKCLR00 and INTMSKCLR04. Note that writing 0 to the related bit in INTMSK00 and INTMSK04 does not cancel masking of the interrupt.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial value:	1	1	1	1	0	0	1	1	1	1	1	1	1	1	1	1
R/W:	R/W	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

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 28		All 1	R/W	Interrupt Masks 31 to 28 These bits set the masking of the interrupt request that corresponds to the given bit. For the correspondence between bits and interrupt sources, see table 9.5. 0: Corresponding interrupt requests are accepted 1: Corresponding interrupt requests are masked
27, 26		All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
25 to 16		All 1	R/W	Interrupt Masks 25 to 16 These bits set the masking of the interrupt request that corresponds to the given bit. For the correspondence between bits and interrupt sources, see table 9.5. 0: Corresponding interrupt requests are accepted 1: Corresponding interrupt requests are masked
15		0	R	Reserved This bit is always read as 0. The write value should always be 0.
14 to 0		All 1	R/W	Interrupt Masks 14 to 0 These bits set the masking of the interrupt request that corresponds to the given bit. For the correspondence between bits and interrupt sources, see table 9.5. 0: Corresponding interrupt requests are accepted 1: Corresponding interrupt requests are masked

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial value:	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
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
Initial value:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W:	R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24		All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
23 to 15		All 1	R/W	Interrupt Masks 23 to 15 These bits set the masking of the interrupt request that corresponds to the given bit. For the correspondence between bits and interrupt sources, see table 9.5. 0: Corresponding interrupt requests are accepted 1: Corresponding interrupt requests are masked
14 to 7		All 1	R	Reserved These bits are read as 1s at the first time. After that, these bits are read as 0s. The write value should always be 0.
6 to 0		All 1	R/W	Interrupt Masks 6 to 0 These bits set the masking of the interrupt request that corresponds to the given bit. For the correspondence between bits and interrupt sources, see table 9.5. 0: Corresponding interrupt requests are accepted 1: Corresponding interrupt requests are masked

INTMSKCLR00 and INTMSKCLR04 are 32-bit read-only registers that clear the masking of individual interrupt requests.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
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
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W:	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	—	—	W	Interrupt Mask Clear 31 to 0 Each bit selects whether or not to clear the masking of the interrupt source that corresponds to that bit. For the correspondence between the bits and interrupt sources, see table 9.5. 0: Masking of corresponding interrupt is not changed 1: Masking of corresponding interrupt is cleared

There are four types of interrupt sources: NMI, IRQ, IRL, and peripheral modules. Each interrupt has a priority level (16 to 0), with level 16 as the highest and level 1 as the lowest. When level 0 is set, the interrupt is masked and interrupt requests are ignored.

9.4.1 NMI Interrupt

The NMI interrupt has the highest priority level of 16. It is always accepted unless the BL bit in SR of the CPU is set to 1. In sleep or standby mode, the interrupt is accepted even if the BL bit is set to 1.

A setting can also be made to have the NMI interrupt accepted even if the BL bit is set to 1.

Input from the NMI pin is edge-detected. The NMI edge select bit (NMIE) in ICR is used to select either rising or falling edge as the detection edge. When the NMIE bit in ICR is modified, the NMI interrupt is not detected for a maximum of six bus clock cycles after the modification.

NMI interrupt exception handling does not affect the interrupt mask level bits (IMASK3 to IMASK0) in SR.

9.4.2 IRQ Interrupts

IRQ interrupts are input by level at pins $\overline{\text{IRQ7}}$ to $\overline{\text{IRQ4}}$. After an IRQ interrupt is accepted, the pin level must be retained until the interrupt processing starts.

9.4.3 IRL Interrupts

IRL interrupts are input by level at pins $\overline{\text{IRL3}}$ to $\overline{\text{IRL0}}$.

The priority level is the 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 9.2 shows an example of IRL interrupt connection, and table 9.6 shows the correspondence between the $\overline{\text{IRL}}$ pins and interrupt levels.

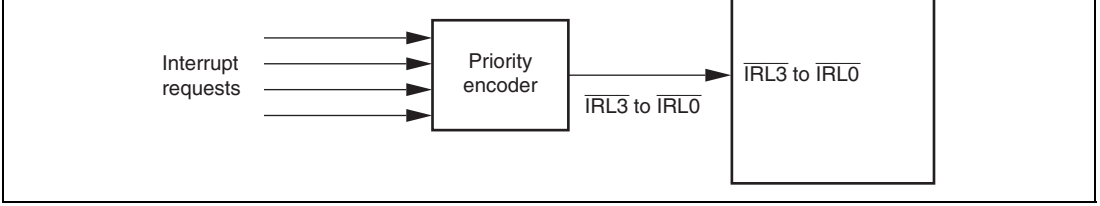


Figure 9.2 Example of IRL Interrupt Connection

Table 9.6 $\overline{\text{IRL3}}$ 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

IRL interrupt detection requires a built-in noise-cancellation feature. The IRL interrupt is not detected unless the levels sampled at every bus clock cycle remain unchanged for three consecutive cycles, so that no transient level on the $\overline{\text{IRL}}$ pin change is detected.

The priority level of the IRL interrupt must not be lowered until the interrupt is accepted and the interrupt processing starts. However, the priority level can be changed to a higher one.

The interrupt mask level bits (IMASK3 to IMASK0) in the status register (SR) are not affected by IRL interrupt processing.

ICR to 1.

When independent interrupt requests are used, the interrupt priority levels can be set in interrupt priority level setting register D (IPRD).

9.4.4 Peripheral Module Interrupts

Peripheral module interrupts are interrupts generated by peripheral modules.

Not every interrupt source is assigned a different interrupt vector, but sources are reflected in the interrupt event register (INTEVT), so it is easy to identify sources by using the INTEVT value as a branch offset in the exception handling routine.

A priority level from 15 to 0 can be set for each module by means of IPRA to IPRD and INTPRI00 to INTPRI0C.

The interrupt mask level bits (IMASK3 to IMASK0) in SR are not affected by peripheral module interrupt processing.

Updating of the interrupt source flag and interrupt enable flag of a peripheral module should only be carried out when the BL bit in SR is set to 1. To prevent erroneous interrupt acceptance from an interrupt source that should have been updated, first read the on-chip peripheral register containing the relevant flag, then clear the BL bit to 0. This will secure the necessary timing internally. When updating a number of flags, there is no problem if only the register containing the last flag updated is read from.

If flag updating is performed while the BL bit is cleared to 0, the program may jump to the interrupt handling routine when the INTEVT value is 0. In this case, interrupt processing is initiated due to the timing relationship between the flag update and interrupt request recognition within this LSI. Processing can be continued without any problem by executing an RTE instruction.

Table 9.7 lists the codes for the interrupt event register (INTEVT), and the order of interrupt priority.

Each interrupt source is assigned a unique INTEVT code. The start address of the exception handling routine is common to each interrupt source. Therefore, to identify the interrupt source, branching is performed at the start of the exception handling routine using the INTEVT value. For instance, the INTEVT value is used as a branch offset .

The priority order of the peripheral modules is specified as desired by setting priority levels from 15 to 0 in IPRA to IPRD and INTPRI00 to INTPRI0C. The priority order of the peripheral modules is set to 0 by a reset.

When the priorities for multiple interrupt sources are set to the same level and such interrupts are generated simultaneously, they are handled according to the default priority order shown in table 9.7.

Updating of IPRA to IPRD and INTPRI00 to INTPRI0C should only be carried out when the BL bit in SR is set to 1. To prevent erroneous interrupt acceptance, first read one of the interrupt priority level setting registers, then clear the BL bit to 0. This will secure the necessary timing internally.

Interrupt Source		INTEVT Code	Interrupt Priority (Initial Value)	Relevant IPR (Bit Numbers)	Priority within IPR Setting Unit	Default Priority
NMI		H'1C0	16	—	—	High
IRL3 to IRL0	IRL3 to IRL0 = 0	H'200	15	—	—	
	IRL3 to IRL0 = 1	H'220	14	—	—	
	IRL3 to IRL0 = 2	H'240	13	—	—	
	IRL3 to IRL0 = 3	H'260	12	—	—	
	IRL3 to IRL0 = 4	H'280	11	—	—	
	IRL3 to IRL0 = 5	H'2A0	10	—	—	
	IRL3 to IRL0 = 6	H'2C0	9	—	—	
	IRL3 to IRL0 = 7	H'2E0	8	—	—	
	IRL3 to IRL0 = 8	H'300	7	—	—	
	IRL3 to IRL0 = 9	H'320	6	—	—	
	IRL3 to IRL0 = A	H'340	5	—	—	
	IRL3 to IRL0 = B	H'360	4	—	—	
	IRL3 to IRL0 = C	H'380	3	—	—	
	IRL3 to IRL0 = D	H'3A0	2	—	—	
	IRL3 to IRL0 = E	H'3C0	1	—	—	
IRL	IRL0	H'240	15 to 0 (13)	IPRD (15 to 12)	—	
	IRL1	H'2A0	15 to 0 (10)	IPRD (11 to 8)	—	
	IRL2	H'300	15 to 0 (7)	IPRD (7 to 4)	—	
	IRL3	H'360	15 to 0 (4)	IPRD (3 to 0)	—	
H-UDI	H-UDI	H'600	15 to 0 (0)	IPRC (3 to 0)	—	
GPIO	GPIOI	H'620	15 to 0 (0)	IPRC (15 to 12)	—	Low

DMAC	DMTE0	H'640	15 to 0 (0)	IPRC (11 to 8)	High	High
	DMTE1	H'660				
	DMTE2	H'680				
	DMTE3	H'6A0				
	DMTE4	H'780				
	DMTE5	H'7A0				
	DMTE6	H'7C0				
	DMTE7	H'7E0				
	DMAE	H'6C0			Low	
IRQ	IRQ4	H'800	15 to 0 (0)	INTPRI00 (31 to 28)	High	
	IRQ5	H'820	15 to 0 (0)	INTPRI00 (27 to 24)		
	IRQ6	H'840	15 to 0 (0)	INTPRI00 (23 to 20)		
	IRQ7	H'860	15 to 0 (0)	INTPRI00 (19 to 16)	Low	
HCAN2(0)	CANI0	H'900	15 to 0 (0)	INTPRI04 (31 to 28)		
HCAN2(1)	CANI1	H'920	15 to 0 (0)	INTPRI04 (27 to 24)		
SSI(0)	SSII0	H'940	15 to 0 (0)	INTPRI04 (23 to 20)		
SSI(1)	SSII1	H'960	15 to 0 (0)	INTPRI04 (19 to 16)		
HAC(0)	HACI0	H'980	15 to 0 (0)	INTPRI04 (15 to 12)		
HAC(1)	HACI1	H'9A0	15 to 0 (0)	INTPRI04 (11 to 8)		
$I^2C(0)$	IICI0	H'9C0	15 to 0 (0)	INTPRI04 (7 to 4)		
$I^2C(1)$	IICI1	H'9E0	15 to 0 (0)	INTPRI04 (3 to 0)		Low

USB	USBI	H'A00	15 to 0 (0)	INTPRI08 (31 to 28)		High
LCDC	VINT	H'A20	15 to 0 (0)	INTPRI08 (27 to 24)		
DMABRG	DMABRGI0	H'A80	15 to 0 (0)	INTPRI08 (23 to 20)	High	
	DMABRGI1	H'AA0				
	DMABRGI2	H'AC0			Low	
SCIF(0)	ERI0	H'880	15 to 0 (0)	INTPRI08 (19 to 16)	High	
	RXI0	H'8A0				
	BRI0	H'8C0				
	TXI0	H'8E0			Low	
SCIF(1)	ERI1	H'B00	15 to 0 (0)	INTPRI08 (15 to 12)	High	
	RXI1	H'B20				
	BRI1	H'B40				
	TXI1	H'B60			Low	
SCIF(2)	ERI2	H'B80	15 to 0 (0)	INTPRI08 (11 to 8)	High	
	RXI2	H'BA0				
	BRI2	H'BC0				
	TXI2	H'BE0			Low	
SIM	SIMERI	H'C00	15 to 0 (0)	INTPRI08 (7 to 4)	High	
	SIMRXI	H'C20				
	SIMTXI	H'C40				
	SIMTEI	H'C60			Low	
HSPI	HSPII	H'C80	15 to 0 (0)	INTPRI08 (3 to 0)		
MMCIF	MMCI0	H'D00	15 to 0 (0)	INTPRI0C (23 to 20)	High	
	MMCI1	H'D20				
	MMCI2	H'D40				
	MMCI3	H'D60			Low	Low

MFI	MFII	H'E80	15 to 0 (0)	INTPRI0C (15 to 12)		High
—	—	H'F00	15 to 0 (0)	INTPRI0C (11 to 8)	High	
	—	H'F20				
	—	H'F40				
	—	H'F60			Low	
ADC	ADI	H'F80	15 to 0 (0)	INTPRI0C (7 to 4)		
CMT	CMTI	H'FA0	15 to 0 (0)	INTPRI0C (3 to 0)		
TMU0	TUNI0	H'400	15 to 0 (0)	IPRA (15 to 12)	—	
TMU1	TUNI1	H'420	15 to 0 (0)	IPRA (11 to 8)	—	
TMU2	TUNI2	H'440	15 to 0 (0)	IPRA (7 to 4)	High	
	TICPI2	H'460			Low	
WDT	ITI	H'560	15 to 0 (0)	IPRB (15 to 12)	—	
REF	RCMI	H'580	15 to 0 (0)	IPRB (11 to 8)	High	
	ROVI	H'5A0			Low	Low

Note: TUNI0 to TUNI2 : Underflow interrupts
TICPI2 : Input capture interrupt
ERI : Receive-error interrupt
RXI : Receive-data-full interrupt
TXI : Transmit-data-empty interrupt
SIMTEI : Transmit-end interrupt
BRI : Break interrupt request
ITI : Interval timer interrupt
RCMI : Compare-match interrupt
ROVI : Refresh count overflow interrupt
H-UDI : User debug interface
DMTE0 to DMTE7 : DMAC transfer end interrupts
DMAE : DMAC address error interrupt
VINT : Vertical synchronization interrupt

9.5.1 Interrupt Operation Sequence

The sequence of operations when an interrupt is generated is described below. Figure 9.3 shows a flowchart of the interrupt 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, according to the priority levels set in IPRA to IPRD and INTPRI00 to INTPRI0C. 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 priority according to table 9.4 is selected.
3. The priority level of the interrupt selected by the interrupt controller is compared with the interrupt mask level (IMASK3 to IMASK0) in SR of the CPU. If the request's priority level is higher than the level in bits IMASK3 to IMASK0, the interrupt controller accepts the interrupt and sends an interrupt request signal to the CPU.
4. The CPU accepts an interrupt between instructions.
5. The interrupt source code is set in the interrupt event register (INTEVT).
6. The contents of the status register (SR) and program counter (PC) are saved to SSR and SPC, respectively. The R15 contents at this time are saved in SGR.
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-related exception handling routine (the sum of the value set in the vector base register (VBR) and H'0000 0600).

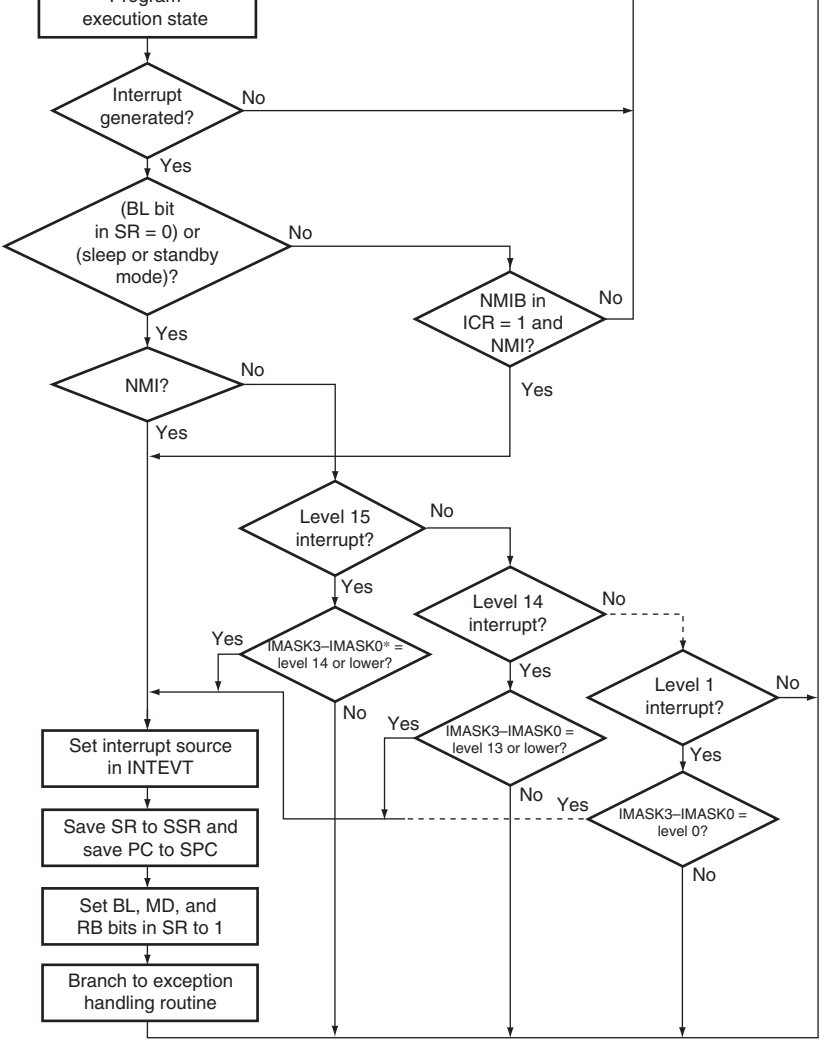
The exception handling routine may branch with the INTEVT value as its offset in order to identify the interrupt source. This enables it to easily branch to the handling routine for the particular interrupt source.

Notes: 1. In this LSI, the interrupt mask level bits (IMASK3 to IMASK0) in the status register (SR) of the CPU are not changed by acceptance of an interrupt.

2. Clear the interrupt source flag during the interrupt handling routine.

   To ensure that the cleared interrupt source is not inadvertently accepted again, read the interrupt source flag after it has been cleared, wait for the interval shown in table 9.8, and then clear the BL bit or execute an RTE instruction.

3. For some interrupt sources, the interrupt masks (INTMSK00 and INTMSK04) must be cleared using INTMSKCLR00 and INTMSKCLR04.



Note: * IMASK3–IMASK0: Interrupt mask level bits in status register (SR)

Figure 9.3 Interrupt Operation Flowchart

When handling multiple interrupts, the interrupt handling routine should include the following procedures:

1. Branch to the interrupt handling routine of each interrupt source using the INTEVT value as an offset to identify the interrupt source.
2. Clear the interrupt source in the corresponding interrupt handling routine.
3. Save SPC and SSR in the stack.
4. Clear the BL bit in SR, and set the accepted interrupt level in the interrupt mask level bits (IMASK3 to IMASK0) in SR.
5. Write the actual processing.
6. Set the BL bit in SR to 1.
7. Restore SSR and SPC from memory.
8. Execute the RTE instruction.

When these procedures are followed in order, an interrupt of higher priority than the one being handled can be accepted immediately after step 4. This enables the interrupt response time to be shortened for urgent processing.

9.5.3 Interrupt Masking with MAI Bit

By setting the MAI bit to 1 in ICR, interrupts can be masked while the NMI pin is low, irrespective of the BL and IMASK bits in SR.

- In normal operation and sleep mode

  All interrupts are masked while the NMI pin is low. However, an NMI interrupt only is generated by a transition at the NMI pin.
- In standby mode

  All interrupts are masked while the NMI pin is low, and an NMI interrupt is not generated by a transition at the NMI pin. Therefore, standby mode cannot be cleared by an NMI interrupt while the MAI bit is set to 1.

The time from interrupt request generation until interrupt exception handling is performed and fetching of the first instruction of the exception handling routine is started (the interrupt response time) is shown in table 9.8.

Table 9.8 Interrupt Response Time

Item		Number of States: NMI	IRL IRQ	Peripheral Modules	Notes
Time for priority decision and SR mask bit comparison		1 Icyc + 4 Bcyc	1 Icyc + 7 Bcyc	1 Icyc + 2 Bcyc	
Wait time until end of sequence being executed by CPU		S – 1 (≥ 0) × Icyc	S – 1 (≥ 0) × Icyc	S – 1 (≥ 0) × Icyc	
Time from interrupt exception handling (save of SR and PC) until fetch of first instruction of exception handling routine is started		4 × Icyc	4 × Icyc	4 × Icyc	
Response time	Total	5 Icyc + 4 Bcyc + (S – 1) Icyc	5 Icyc + 7 Bcyc + (S – 1) Icyc	5 Icyc + 2 Bcyc + (S – 1) Icyc	
	Minimum case	13 Icyc	19 Icyc	9 Icyc	When Icyc: Bcyc = 2:1
	Maximum case	36 + S Icyc	60 + S Icyc	20 + S Icyc	When Icyc: Bcyc = 8:1

Icyc : One cycle of internal clock supplied to CPU, etc.

Bcyc : One CKIO cycle

S : Number of instruction execution states

The BSC divides the off-chip memory space and outputs control signals complying with specifications of various types of memory and bus interfaces. It enables the connection of synchronous DRAM, SRAM, ROM, etc., to this LSI. It also supports the PCMCIA interface protocol, which implements simplified system design and high-speed data transfers by a compact system.

10.1 Features

The BSC has the following features:

- Divides the off-chip memory space into seven areas for management.
  - Maximum 64 Mbytes for each of areas 0 to 6
  - Bus width of each area can be controlled by register settings (except area 0, which uses an off-chip pin setting)
  - Wait-cycle insertion by $\overline{\text{RDY}}$ pin
  - Wait-cycle insertion can be controlled by program
  - Types of memory are specifiable for connection to each area
  - Output the control signals of memory to each area
  - Automatic wait cycle insertion to prevent data bus collisions in case of consecutive memory accesses to different areas, or a read access followed by a write access to the same area
  - Write strobe setup time and hold time periods can be inserted in a write cycle to enable connection to low-speed memory
- SRAM interface
  - Wait-cycle insertion can be controlled by program
  - Wait-cycle insertion by $\overline{\text{RDY}}$ pin
  - Connectable areas: 0 to 6
  - Settable bus widths: 32, 16, 8
- Synchronous DRAM interface
  - Row address/column address multiplexing according to synchronous DRAM capacity
  - Burst operation
  - Auto-refresh and self-refresh
  - Synchronous DRAM control signal timing can be controlled by register settings
  - Consecutive accesses to the same row address
  - Connectable areas: 2, 3
  - Settable bus width: 32

— Wait-cycle insertion can be controlled by program
— Burst transfer for the number of times specified by the register
— Connectable areas: 0, 5, 6
— Settable bus widths: 32, 16, 8

- MPX interface
  — Address/data multiplexing
  — Peripheral LSI, which requires address/data multiplexing, can be connected
  — Connectable areas: 0 to 6
  — Settable bus width: 32
- Byte control SRAM interface
  — SRAM interface with byte control
  — Connectable areas: 1, 4
  — Settable bus widths: 32, 16
- PCMCIA interface (valid only in little-endian mode)
  — Wait-cycle insertion can be controlled by program
  — Bus sizing function for I/O bus width
  — Connectable areas: 5, 6
  — Settable bus widths: 16, 8
- Refresh counter can be used as interval timer
  — Interrupt request generated by compare-match
  — Interrupt request generated by refresh counter overflow

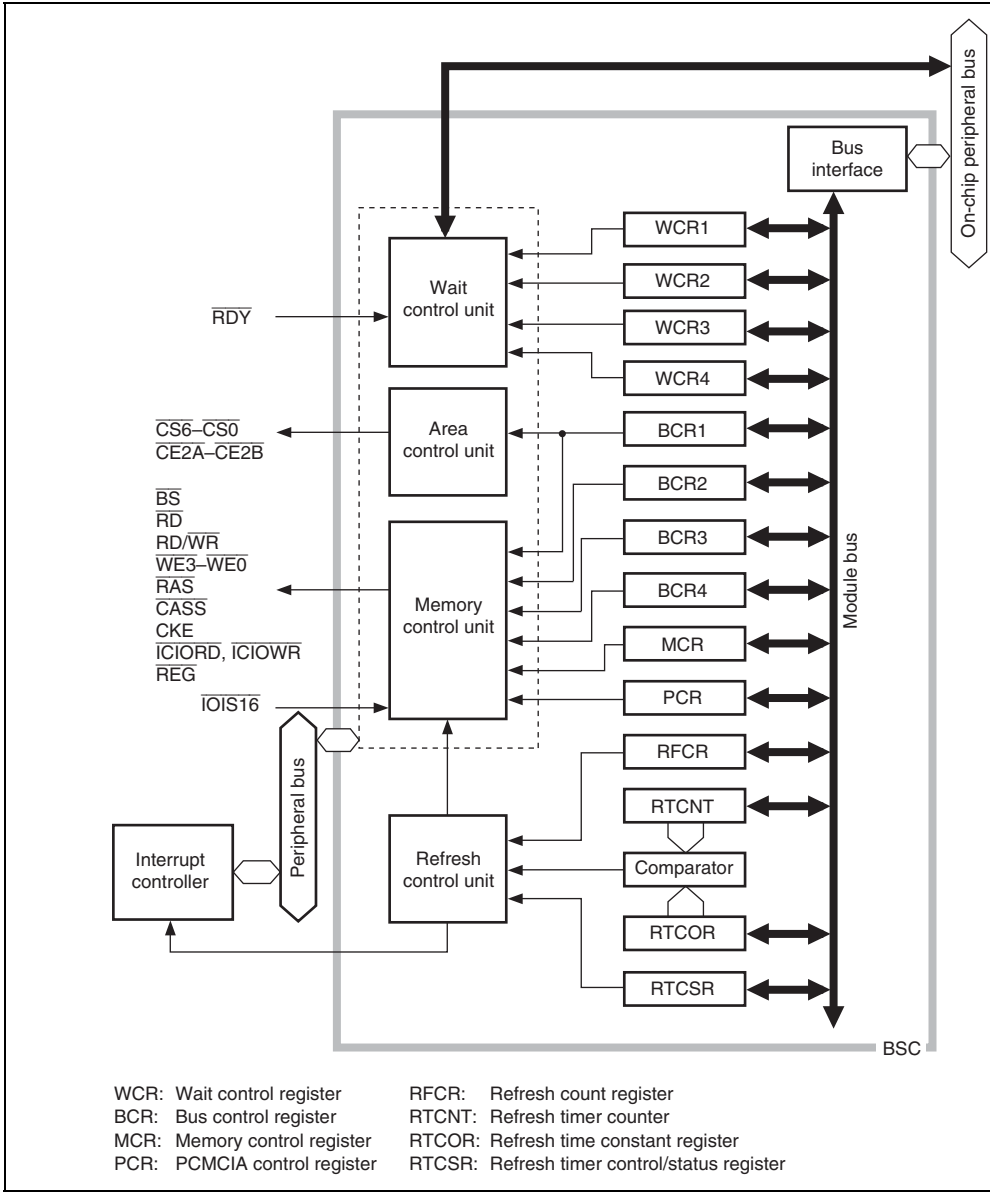


Figure 10.1 Block Diagram of BSC

Table 10.1 shows the BSC pin configuration.

Table 10.1 Pin Configuration

Name	Signals	I/O	Description
Address bus	A25 to A0	Output	Address output
Data bus	D31 to D0	Input/ Output	Data input/output
Bus cycle start	$\overline{BS}$	Output	Signal that indicates the start of a bus cycle When setting synchronous DRAM interface or MPX interface: asserted once for a burst transfer For other burst transfers: asserted each data cycle
Chip select 6 to 0	$\overline{CS6}$ to $\overline{CS0}$	Output	Chip select signals that indicate the area being accessed $\overline{CS5}$ and $\overline{CS6}$ are also used as PCMCIA $\overline{CE1A}$ and $\overline{CE1B}$
Read/write	RD/$\overline{WR}$	Output	Data bus input/output direction designation signal Also used as the DRAM/PCMCIA interface write designation signal
Row address strobe	$\overline{RAS}$	Output	$\overline{RAS}$ signal when setting synchronous DRAM interface
Read/column address strobe/ cycle frame	$\overline{RD}$/$\overline{CASS}$/ $\overline{FRAME}$	Output	Strobe signal that indicates a read cycle When setting synchronous DRAM interface: $\overline{CAS}$ signal When setting MPX interface: $\overline{FRAME}$ signal
Data enable 0	$\overline{WE0}$/ DQM0/ $\overline{REG}$	Output	When setting PCMCIA interface: $\overline{REG}$ signal When setting SRAM interface: write strobe signal for D7 to D0 When setting synchronous DRAM interface: selection signal for D7 to D0
Data enable 1	$\overline{WE1}$/ DQM1	Output	When setting PCMCIA interface: write strobe signal When setting SRAM interface: write strobe signal for D15 to D8 When setting synchronous DRAM interface: selection signal for D15 to D8

Data enable 2	WE2/ DQM2/ ICIORD	Output	When setting PCMCIA interface: ICIORD signal When setting SRAM interface: write strobe signal for D23 to D16 When setting synchronous DRAM interface: selection signal for D23 to D16
Data enable 3	WE3/ DQM3/ ICIOWR	Output	When setting PCMCIA interface: ICIOWR signal When setting SRAM interface: write strobe signal for D31 to D24 When setting synchronous DRAM interface: selection signal for D31 to D24
Ready	RDY	Input	Wait-cycle request signal
Area 0 MPX interface setting/ 16-bit I/O	MD6/IOIS16	Input	At power-on reset: Designates area 0 bus as MPX interface (1: SRAM, 0: MPX) When setting PCMCIA interface: 16-bit I/O designation signal. Valid only in little-endian mode.
Clock enable	CKE	Output	Synchronous DRAM clock enable control signal
Bus release request	BREQ	Input	Bus release request signal
Bus request acknowledge	BACK	Output	Bus request acknowledge signal
Area 0 bus width/PCMCIA card select	MD3/CE2A* MD4/CE2B*	Input/ Output	At power-on reset: area 0 bus width specification signal When using PCMCIA: CE2A, CE2B
Endian switchover	MD5	Input	Endian setting at a power-on reset

Note: * The input/output switching is specified by the A56PCM bit in bus control register 1 (BCR1).

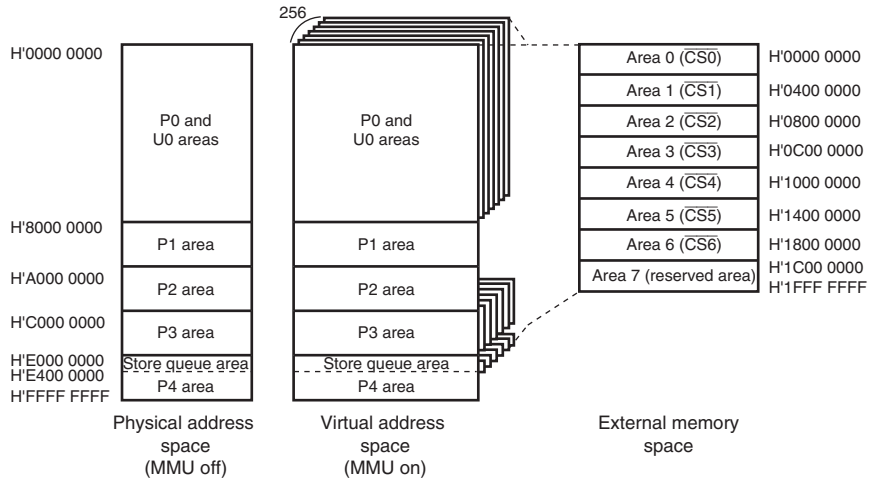
10.3 Overview of Areas

(1) Space Divisions

The architecture of this LSI provides a 32-bit virtual address space. The virtual address space is divided into five areas according to the upper address value. Off-chip memory space comprises a 29-bit address space, divided into eight areas.

The virtual address space can be allocated to any off-chip address by means of the memory management unit (MMU). Details are given in section 6, Memory Management Unit (MMU). This section describes the areas into which the off-chip address space is divided.

off-chip address space as shown in table 10.2, and chip select signals ($\overline{CS0}$ to $\overline{CS6}$, $\overline{CE2A}$, $\overline{CE2B}$) are output for each of these areas. $\overline{CS0}$ is asserted when accessing area 0, and $\overline{CS6}$ when accessing area 6. When synchronous DRAM is connected to area 2 or 3, signals such as $\overline{RAS}$, $\overline{CASS}$, RD/$\overline{WR}$, and DQM are also asserted. When the PCMCIA interface is selected for area 5 or 6, $\overline{CE2A}$ or $\overline{CE2B}$ is asserted in addition to $\overline{CS5}$ or $\overline{CS6}$ for the byte to be accessed.



Notes: 1. When the MMU is off (MMUCR.AT = 0), the top 3 bits of the 32-bit address are ignored, and memory is mapped onto a fixed 29-bit external address.
2. When the MMU is on (MMUCR.AT = 1), the P0, U0, P3, and store queue areas can be mapped onto any external address using the TLB.

Figure 10.2 Correspondence between Virtual Address Space and Off-chip Memory Space

Area	Off-chip Addresses	Size	Connectable Memory	Specifiable Bus Width*1*2	Access Size
0	H'0000 0000 to H'03FF FFFF	64 Mbytes	SRAM	8, 16, 32	8, 16, 32, 64*6 bits, 32 bytes
			Burst ROM	8, 16, 32	
			MPX	32	
1	H'0400 0000 to H'07FF FFFF	64 Mbytes	SRAM	8, 16, 32	8, 16, 32, 64*6 bits, 32 bytes
			MPX	32	
			Byte control SRAM	16, 32	
2	H'0800 0000 to H'0BFF FFFF	64 Mbytes	SRAM	8, 16, 32	8, 16, 32, 64*6 bits, 32 bytes
			Synchronous DRAM	32*3	
			MPX	32	
3	H'0C00 0000 to H'0FFF FFFF	64 Mbytes	SRAM	8, 16, 32	8, 16, 32, 64*6 bits, 32 bytes
			Synchronous DRAM	32*3	
			MPX	32	
4	H'1000 0000 to H'13FF FFFF	64 Mbytes	SRAM	8, 16, 32	8, 16, 32, 64*6 bits, 32 bytes
			MPX	32	
			Byte control RAM	16, 32	
5	H'1400 0000 to H'17FF FFFF	64 Mbytes	SRAM	8, 16, 32	8, 16, 32, 64*6 bits, 32 bytes
			MPX	32	
			Burst ROM	8, 16, 32	
			PCMCIA	8, 16*4	
6	H'1800 0000 to H'1BFF FFFF	64 Mbytes	SRAM	8, 16, 32	8, 16, 32, 64*6 bits, 32 bytes
			MPX	32	
			Burst ROM	8, 16, 32	
			PCMCIA	8, 16*4	
7*5	H'1C00 0000 to H'1FFF FFFF	64 Mbytes	—	—	—

Notes: 1. The memory bus width in area 0 is specified by off-chip pins.
2. The memory bus width in areas other than area 0 is specified by the register.
3. With synchronous DRAM interface, the bus width is 32 bits only.
4. With PCMCIA interface, the bus width is 8 or 16 bits only.
5. Do not access a reserved area, as operation cannot be guaranteed in this case.
6. A 64-bit access size applies only to transfer by the DMAC (CHCRn.TS = 000).
In the case of access to off-chip memory by means of FMOV (FPSCR.SZ = 1), two 32-bit access size transfers are performed.

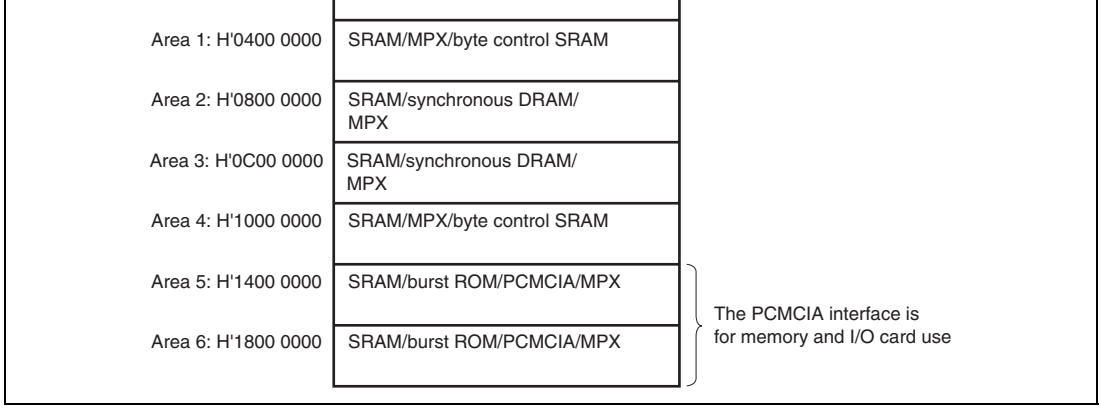


Figure 10.3 Off-chip Memory Space Allocation

(2) Memory Bus Width

In this LSI, the memory bus width can be set independently for each area. For area 0, a bus width of 8, 16, or 32 bits can be selected at a power-on reset by a $\overline{\text{RESET}}$ signal, using off-chip pins. The correspondence between the off-chip pins (MD4 and MD3) and the bus width at a power-on reset by a $\overline{\text{RESET}}$ signal is shown below.

Table 10.3 Correspondence between Off-chip Pins (MD4 and MD3) and Bus Width

MD4	MD3	Bus Width
0	0	Reserved (Setting prohibited)
	1	8 bits
1	0	16 bits
	1	32 bits

When SRAM interface is used in areas 1 to 6, a bus width of 8, 16, or 32 bits can be selected by BCR2. When burst ROM interface is used, a bus width of 8, 16, or 32 bits can be selected. When byte control SRAM interface is used, a bus width of 16 or 32 bits can be selected. When the MPX interface is used, a bus width of 32 bits should be selected. For the synchronous DRAM interface, a bus width of 32 bits should be selected by the MCR register.

When using the PCMCIA interface, a bus width of 8 or 16 bits should be selected.

The addresses of area 7 (H'1C00 0000 to H'1FFF FFFF) are reserved and must not be used.

This LSI supports PCMCIA interface specifications for off-chip memory space areas 5 and 6.

The IC memory card interface and I/O card interface prescribed in JEIDA specifications version 4.2 (PCMCIA2.1) are supported.

The PCMCIA interface is only supported in little endian mode.

Table 10.4 PCMCIA Interface Features

Item	Features
Access	Random access
Data bus	8/16 bits
Memory type	Masked ROM, OTPROM, EPROM, EEPROM, flash memory, SRAM
Common memory capacity	Max. 64 Mbytes
Attribute memory capacity	Max. 64 Mbytes
Others	Dynamic bus sizing for I/O bus width, access to PCMCIA interface from address translation areas

Table 10.5 PCMCIA Support Interfaces

	IC Memory Card Interface			I/O Card Interface			
Pin	Signal Name	I/O	Function	Signal Name	I/O	Function	Corresponding Pin for this LSI
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	$\overline{CS5}$ or $\overline{CS6}$
8	A10	I	Address	A10	I	Address	A10
9	$\overline{OE}$	I	Output enable	$\overline{OE}$	I	Output enable	$\overline{RD}$
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

Pin	Signal Name	I/O	Function	Signal Name	I/O	Function	Corresponding Pin for this LSI
14	A14	I	Address	A14	I	Address	A14
15	$\overline{WE}$/$\overline{PGM}$	I	Write enable	$\overline{WE}$/$\overline{PGM}$	I	Write enable	$\overline{WE1}$
16	$\overline{RDY}$/$\overline{BSY}$	O	Ready/busy	$\overline{IREQ}$	O	Interrupt request	Sensed on port
17	VCC		Operating power supply	VCC		Operating power supply	—
18	VPP1		Programming power supply	VPP1		Programming/ 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
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	$\overline{WP}$*[1]	O	Write protect	$\overline{IOIS16}$	O	16-bit I/O port	$\overline{IOIS16}$
34	GND		Ground	GND		Ground	—
35	GND		Ground	GND		Ground	—
36	$\overline{CD1}$	O	Card detection	$\overline{CD1}$	O	Card detection	Sensed on port
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	$\overline{CE2}$	I	Card enable	$\overline{CE2}$	I	Card enable	$\overline{CE2A}$ or $\overline{CE2B}$
43	RFSH	I	Refresh request	RFSH	I	Refresh request	Output from port

Pin	Signal Name	I/O	Function	Signal Name	I/O	Function	Corresponding Pin for this LSI
44	RFU		Reserved	$\overline{\text{IORD}}$	I	I/O read	$\overline{\text{ICIORD}}$
45	RFU		Reserved	$\overline{\text{IOWR}}$	I	I/O write	$\overline{\text{ICIOWR}}$
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/ 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	RFU		Reserved	RFU		Reserved	—
58	RESET	I	Reset	RESET	I	Reset	Output from port
59	$\overline{\text{WAIT}}$	O	Wait request	$\overline{\text{WAIT}}$	O	Wait request	$\overline{\text{RDY}}$*[2]
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	$\overline{\text{REG}}$
62	BVD2	O	Battery voltage detection	$\overline{\text{SPKR}}$	O	Digital speech signal	Sensed on port
63	BVD1	O	Battery voltage detection	$\overline{\text{STSCHG}}$	O	Card status change	Sensed on port
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	Sensed on port
68	GND		Ground	GND		Ground	—

Notes: 1. $\overline{\text{WP}}$ is not supported.
2. Check the polarity.

The BSC has the following registers. The synchronous DRAM mode register in synchronous DRAM can be accessed as a register for this LSI. The following registers control memory interfaces, wait-cycles, and refresh cycles, etc.

Table 10.6 Register Configuration (1)

Register Name	Abbrev.	R/W	P4 Address	Area 7 Address	Size	Sync Clock
Bus control register 1	BCR1	R/W	H'FF80 0000	H'1F80 0000	32	Bck
Bus control register 2	BCR2	R/W	H'FF80 0004	H'1F80 0004	16	Bck
Bus control register 3	BCR3	R/W	H'FF80 0050	H'1F80 0050	16	Bck
Bus control register 4	BCR4	R/W	H'FE0A 00F0	H'1E0A 00F0	32	Bck
Wait control register 1	WCR1	R/W	H'FF80 0008	H'1F80 0008	32	Bck
Wait control register 2	WCR2	R/W	H'FF80 000C	H'1F80 000C	32	Bck
Wait control register 3	WCR3	R/W	H'FF80 0010	H'1F80 0010	32	Bck
Wait control register 4	WCR4	R/W	H'FE0A 0028	H'1E0A 0028	32	Bck
Memory control register	MCR	R/W	H'FF80 0014	H'1F80 0014	32	Bck
PCMCIA control register	PCR	R/W	H'FF80 0018	H'1F80 0018	16	Bck
Refresh timer control/status register	RTCSR	R/W	H'FF80 001C	H'1F80 001C	16	Bck
Refresh timer counter	RTCNT	R/W	H'FF80 0020	H'1F80 0020	16	Bck
Refresh timer constant register	RTCOR	R/W	H'FF80 0024	H'1F80 0024	16	Bck
Refresh count register	RFCR	R/W	H'FF80 0028	H'1F80 0028	16	Bck
Synchronous DRAM mode register (for area 2)	SDMR2	W	H'FF90 xxxx*[1]	H'1F90 xxxx	8	Bck
Synchronous DRAM mode register (for area 3)	SDMR3	W	H'FF94 xxxx*[1]	H'1F94 xxxx	8	Bck

Register Name	Abbrev.	Power-on Reset by RESET Pin/WDT/ H-UDI	Manual Reset by RESET Pin/WDT/ Multiple Exception	Sleep by Sleep Instruction/ Deep Sleep	Standby by Hardware	Standby by Software/ Each Module
Bus control register 1	BCR1	H'0000 0000	Retained	Retained	*2	Retained
Bus control register 2	BCR2	H'3FFC	Retained	Retained		Retained
Bus control register 3	BCR3	H'0001	Retained	Retained		Retained
Bus control register 4	BCR4	H'0000 0000	Retained	Retained		Retained
Wait control register 1	WCR1	H'7777 7777	Retained	Retained		Retained
Wait control register 2	WCR2	H'FFFE EFFF	Retained	Retained		Retained
Wait control register 3	WCR3	H'0777 7777	Retained	Retained		Retained
Wait control register 4	WCR4	H'0000 0000	Retained	Retained		Retained
Memory control register	MCR	H'0000 0000	Retained	Retained		Retained
PCMCIA control register	PCR	H'0000	Retained	Retained		Retained
Refresh timer control/status register	RTCSR	H'0000	Retained	Retained		Retained
Refresh timer counter	RTCNT	H'0000	Retained	Retained		Retained
Refresh timer constant register	RTCOR	H'0000	Retained	Retained		Retained
Refresh count register	RFCR	H'0000	Retained	Retained		Retained
Synchronous DRAM mode register (for area 2)	SDMR2	Write only				
Synchronous DRAM mode register (for area 3)	SDMR3	Write only				

Notes: 1. For details, refer to the descriptions of SDMR.

2. After exiting hardware standby mode, this LSI enters the power-on reset state by the RESET pin.

BCR1 is a 32-bit readable/writable register that specifies the function, bus cycle status, etc., of each area. Do not access off-chip memory space other than area 0 until register initialization is complete.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ENDIAN	-	A0MPX	-	-	DPUP	-	OPUP	-	-	A1MBC	A4MBC	BREQEN		MEMMPX	DMABST
Initial value:	0/1	0	0/1	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R	R/W	R	R	R/W	R/W	R/W	R	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	HIZMEM	HIZCNT	A0BST2	A0BST1	A0BST0	A5BST2	A5BST1	A5BST0	A6BST2	A6BST1	A6BST0	DRAMTP2	DRAMTP1	DRAMTP0	-	A56PCM
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	RW	R/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/W

Bit	Bit Name	Initial Value	R/W	Description
31	ENDIAN	0/1	R	Endian Flag The value of the endian setting off-chip pin (MD5) is sampled at a power-on reset by the $\overline{\text{RESET}}$ pin. This bit determines the endian mode of all spaces. 0: Indicates that pin MD5 is low at a power-on reset and big-endian mode is specified for this LSI. 1: Indicates that pin MD5 is high at a power-on reset and little-endian mode is specified for this LSI.
30	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
29	A0MPX	0/1	R	Area 0 Memory Type The value of the area 0 memory type setting off-chip pin (MD6) is sampled at a power-on reset by the $\overline{\text{RESET}}$ pin. This bit determines the memory type of area 0. 0: Indicates that pin MD6 is high and area 0 is specified as SRAM interface 1: Indicates that pin MD6 is low, and area 0 is specified as MPX interface
28, 27	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

26	DPUP	0	R/W	Data Pin Pull-Up Resistor Control Specifies the pull-up resistor status of the data pins (D31 to D0). It is initialized by a power-on reset. The pins are not pulled up when access is performed or when the bus is released, even if the pull-up resistor is on. 0: Cycle is provided for turning on the pull-up resistor for data pins (D31 to D0) before and after memory access.* 1: Pull-up resistor is off for data pins (D31 to D0). Note: *It is recommended that a pull-up resistor be externally attached for the data pins if it is required.
25	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
24	OPUP	0	R/W	Control Output Pin Pull-Up Resistor Control Specifies the pull-up resistor status (A[25:0], $\overline{BS}$, $\overline{CSn}$, $\overline{RD}$, $\overline{WEn}$, RD/$\overline{WR}$, $\overline{RAS}$, $\overline{CE2A}$, $\overline{CE2B}$) when the control output pins are high-impedance. This bit is initialized by a power-on reset. 0: Pull-up resistor is on for control output pins (A[25:0], $\overline{BS}$, $\overline{CSn}$, $\overline{RD}$, $\overline{WEn}$, RD/$\overline{WR}$, $\overline{RAS}$, $\overline{CE2A}$, $\overline{CE2B}$) 1: Pull-up resistor is off for control output pins (A[25:0], $\overline{BS}$, $\overline{CSn}$, $\overline{RD}$, $\overline{WEn}$, RD/$\overline{WR}$, $\overline{RAS}$, $\overline{CE2A}$, $\overline{CE2B}$)
23, 22	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
21	A1MBC	0	R/W	Area 1 SRAM Byte Control Mode MPX interface has priority when MPX interface is set. This bit is initialized by a power-on reset. 0: Area 1 SRAM is set to normal mode 1: Area 1 SRAM is set to byte control mode
20	A4MBC	0	R/W	Area 4 SRAM Byte Control Mode MPX interface has priority when MPX interface is set. This bit is initialized by a power-on reset. 0: Area 4 SRAM is set to normal mode 1: Area 4 SRAM is set to byte control mode

19	BREQEN	0	R/W	BREQ Enable Indicates whether off-chip requests can be accepted. BREQEN is initialized to the off-chip request acceptance disabled state by a power-on reset. 0: Off-chip requests are not accepted 1: Off-chip requests are accepted
18	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
17	MEMMPX	0	R/W	Area 1 to 6 MPX Bus Setting Sets the MPX interface when areas 1 to 6 are specified as SRAM interface (or burst ROM interface). This bit is initialized by a power-on reset. 0: SRAM interface (or burst ROM interface) is selected when areas 1 to 6 are specified as SRAM interface (or burst ROM interface) 1: MPX interface is selected when areas 1 to 6 are specified as SRAM interface (or burst ROM interface)
16	DMABST	0	R/W	DMAC Burst Mode Transfer Priority Setting Specifies the priority of burst mode transfers by the DMAC. When OFF, the priority is as follows: bus released, refresh, DMAC, CPU. When ON, bus release and refresh operations are not performed until the end of the DMAC burst mode transfer. This bit is initialized at a power-on reset. 0: DMAC burst mode transfer priority setting OFF 1: DMAC burst mode transfer priority setting ON
15	HIZMEM	0	R/W	High Impedance (High-Z) Control Specifies the state of address and other signals (A[25:0], $\overline{BS}$, $\overline{CSn}$, RD/$\overline{WR}$, $\overline{CE2A}$, $\overline{CE2B}$) in software standby mode and the bus-released state. 0: The A[25:0], $\overline{BS}$, $\overline{CSn}$, RD/$\overline{WR}$, $\overline{CE2A}$, and $\overline{CE2B}$ signals made to the high-impedance state in software standby mode and the bus-released state 1: The A[25:0], $\overline{BS}$, $\overline{CSn}$, RD/$\overline{WR}$, $\overline{CE2A}$, and $\overline{CE2B}$ signals driven in software standby mode and made to the high-impedance state in the bus-released state

14	HIZCNT	0	R/W	High Impedance (High-Z) Control Specifies the state of the $\overline{RAS}$ and $\overline{CAS}$ signals in software standby mode and the bus-released state. 0: The $\overline{RAS}$, $\overline{WEn}$ /DQMn, and $\overline{RD}$/$\overline{CASS}$/$\overline{FRAME}$ signals made to the high-impedance in software standby mode and the bus-released state 1: The $\overline{RAS}$, $\overline{WEn}$ /DQMn, and $\overline{RD}$/$\overline{CASS}$/$\overline{FRAME}$ signals driven in software standby mode and the bus-released state
13 12 11	A0BST2 A0BST1 A0BST0	0 0 0	R/W R/W R/W	Area 0 Burst ROM Control These bits specify whether burst ROM interface is used in area 0. When burst ROM interface is used, they also specify the number of accesses in a burst. When area 0 is used as an MPX interface area, the settings of these bits are ignored. 000: Area 0 is accessed as SRAM interface. 001: Area 0 is accessed as burst ROM interface (4 consecutive accesses). Can be used with 8-, 16-, or 32-bit bus width 010: Area 0 is accessed as burst ROM interface (8 consecutive accesses). Can be used with 8-, 16-, or 32-bit bus width 011: Area 0 is accessed as burst ROM interface (16 consecutive accesses). Can only be used with 8- or 16-bit bus width. The setting of 32-bit bus width is prohibited. 100: Area 0 is accessed as burst ROM interface (32 consecutive accesses). Can only be used with 8-bit bus width 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited

10 9 8	A5BST2 A5BST1 A5BST0	0 0 0	R/W R/W R/W	Area 5 Burst ROM Control These bits specify whether burst ROM interface is used in area 5. When burst ROM interface is used, they also specify the number of accesses in a burst. When area 5 is an MPX interface area, the settings of these bits are ignored. When area 5 is used as a PCMCIA area, these bits should be cleared to 0. 000: Area 5 is accessed as SRAM interface. 001: Area 5 is accessed as burst ROM interface (4 consecutive accesses). Can be used with 8-, 16-, or 32-bit bus width 010: Area 5 is accessed as burst ROM interface (8 consecutive accesses). Can be used with 8-, 16-, or 32-bit bus width 011: Area 5 is accessed as burst ROM interface (16 consecutive accesses). Can only be used with 8- or 16-bit bus width. The setting of 32-bit bus width is prohibited. 100: Area 5 is accessed as burst ROM interface (32 consecutive accesses). Can only be used with 8-bit bus width 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited

7 6 5	A6BST2 A6BST1 A6BST0	0 0 0	R/W R/W R/W	Area 6 Burst ROM Control These bits specify whether burst ROM interface is used in area 6. When burst ROM interface is used, they also specify the number of accesses in a burst. When area 6 is an MPX interface area, the settings of these bits are ignored. When area 6 is used as a PCMCIA area, these bits should be cleared to 0. 000: Area 6 is accessed as SRAM interface. 001: Area 6 is accessed as burst ROM interface (4 consecutive accesses). Can be used with 8-, 16-, or 32-bit bus width 010: Area 6 is accessed as burst ROM interface (8 consecutive accesses). Can be used with 8-, 16-, or 32-bit bus width 011: Area 6 is accessed as burst ROM interface (16 consecutive accesses). Can only be used with 8- or 16-bit bus width. The setting of 32-bit bus width is prohibited. 100: Area 6 is accessed as burst ROM interface (32 consecutive accesses). Can only be used with 8-bit bus width 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited

4 3 2	DRAMTP2 DRAMTP1 DRAMTP0	0 0 0	R/W R/W R/W	Areas 2 and 3 Memory Type These bits specify the type of memory connected to areas 2 and 3. Memory types such as ROM, SRAM, and flash ROM can be connected as an SRAM interface. Synchronous DRAM can also be connected. 000: Areas 2 and 3 are accessed as an SRAM interface or MPX interface* 001: Setting prohibited 010: Area 2 is accessed as an SRAM interface or MPX interface* and area 3 as a synchronous DRAM interface 011: Areas 2 and 3 are accessed as a synchronous DRAM interface 100: Setting prohibited 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited Note: * The MEMMPX bit setting selects the SRAM interface or MPX interface.
1	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
0	A56PCM	0	R/W	Area 5 and 6 Bus Type Specifies whether areas 5 and 6 are accessed as PCMCIA interface. The setting of this bit has priority over the MEMMPX bit. When this bit is 1, the MD3 pin is designated for output as the $\overline{CE2A}$ pin, and the MD4 pin is designated for output as the $\overline{CE2B}$ pin. 0: Areas 5 and 6 are accessed as SRAM interface 1: Areas 5 and 6 are accessed as PCMCIA interface

BCR2 is a 16-bit readable/writable register that specifies the bus width for each area and whether the GPIO interrupt is used.

Do not access off-chip memory space other than area 0 until register initialization is complete.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	A0SZ1	A0SZ0	A6SZ1	A6SZ0	A5SZ1	A5SZ0	A4SZ1	A4SZ0	A3SZ1	A3SZ0	A2SZ1	A2SZ0	A1SZ1	A1SZ0	-	STBI RLEN
Initial value:	0/1	0/1	1	1	1	1	1	1	1	1	1	1	1	1	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/W	R/W	-	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	A0SZ1	0/1	R	Area 0 Bus Width
14	A0SZ0	0/1	R	The off-chip pins (MD4 and MD3) that specify the bus width are sampled at a power-on reset by the $\overline{\text{RESET}}$ signal. 00: Setting prohibited 01: 8 bits 10: 16 bits 11: 32 bits
2n + 1	AnSZ1	All 1	R/W	Bus Width Setting
2n	AnSZ0	All 1	R/W	Specifies the bus width of area n. 00: Setting prohibited 01: Bus width is 8 bits 10: Bus width is 16 bits 11: Bus width is 32 bits
1	—	0	—	Reserved This bit is always read as 0. The write value should always be 0.
0	STBIRLEN	0	R/W	GPIO Interrupt Enable 0: GPIO interrupt is not used 1: GPIO interrupt is used

Note: n = 1 to 6

BCR3 is a 16-bit readable/writable register that specifies the selection of either the MPX interface or the SRAM interface and specifies the burst length when the synchronous DRAM interface is used.

Do not access off-chip memory space other than area 0 until register initialization is complete.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MEM MODE	A1 MPX	A4 MPX	-	-	-	-	-	-	-	-	-	-	-	-	SDBL
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W:	R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	MEM MODE	0	R/W	A1MPX/A4MPX Enable Determines whether to use A1MPX and A4MPX or to use MEMMPX for selecting the MPX interface or the SRAM interface. 0: MPX or SRAM interface is selected by MEMMPX 1: MPX or SRAM interface is selected by A1MPX and A4MPX
14	A1MPX	0	R/W	MPX-Interface Setting for Area 1 Specifies the type of memory connected to area 1. This setting is validated by the MEMMODE bit. 0: SRAM/byte control SRAM interface is selected for area 1 1: MPX interface is selected for area 1
13	A4MPX	0	R/W	MPX-Interface Setting for Area 4 Specifies the type of memory connected to area 4. This setting is validated by the MEMMODE bit. 0: SRAM/byte control SRAM interface is selected for area 4 1: MPX interface is selected for area 4
12 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	SDBL	1	R/W	Burst Length Sets the burst length when the synchronous DRAM interface is used. The burst-length setting is only valid when the bus width is 32 bits. 0: Burst length is 8 1: Burst length is 4

BCR4 is a 32-bit readable/writable register that enables asynchronous input for pins corresponding to individual bits. When asynchronous input is set (ASYNCn = 1), the sampling timing is one cycle earlier than when synchronous input is set (ASYNCn = 0). With the synchronous input setting, ensure that setup and hold times are observed.

The timings shown in this section and section 33, Electrical Characteristics, are all for the case where synchronous input is set (ASYNCn = 0).

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
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
	-	-	-	-	-	-	-	-	-	ASYNC6	ASYNC5	ASYNC4	ASYNC3	ASYNC2	ASYNC1	ASYNC0
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/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 7	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
6	ASYNC6	0	R/W	Asynchronous Input 6 0: The $\overline{\text{DREQ3}}$ pin can be used for synchronous input with the CKIO signal. 1: The $\overline{\text{DREQ3}}$ pin can be used for asynchronous input with the CKIO signal.
5	ASYNC5	0	R/W	Asynchronous Input 5 0: The $\overline{\text{DREQ2}}$ pin can be used for synchronous input with the CKIO signal. 1: The $\overline{\text{DREQ2}}$ pin can be used for asynchronous input with the CKIO signal.
4	ASYNC4	0	R/W	Asynchronous Input 4 0: The $\overline{\text{IOIS16}}$ pin can be used for synchronous input with the CKIO signal. 1: The $\overline{\text{IOIS16}}$ pin can be used for asynchronous input with the CKIO signal.

3	ASYNC3	0	R/W	Asynchronous Input 3 0: The $\overline{\text{DREQ1}}$ pin can be used for synchronous input with the CKIO signal. 1: The $\overline{\text{DREQ1}}$ pin can be used for asynchronous input with the CKIO signal.
2	ASYNC2	0	R/W	Asynchronous Input 2 0: The $\overline{\text{DREQ0}}$ pin can be used for synchronous input with the CKIO signal. 1: The $\overline{\text{DREQ0}}$ pin can be used for asynchronous input with the CKIO signal.
1	ASYNC1	0	R/W	Asynchronous Input 1 0: The $\overline{\text{BREQ}}$ pin can be used for synchronous input with the CKIO signal. 1: The $\overline{\text{BREQ}}$ pin can be used for asynchronous input with the CKIO signal.
0	ASYNC0	0	R/W	Asynchronous Input 0 0: The $\overline{\text{RDY}}$ pin can be used for synchronous input with the CKIO signal. 1: The $\overline{\text{RDY}}$ pin can be used for asynchronous input with the CKIO signal.

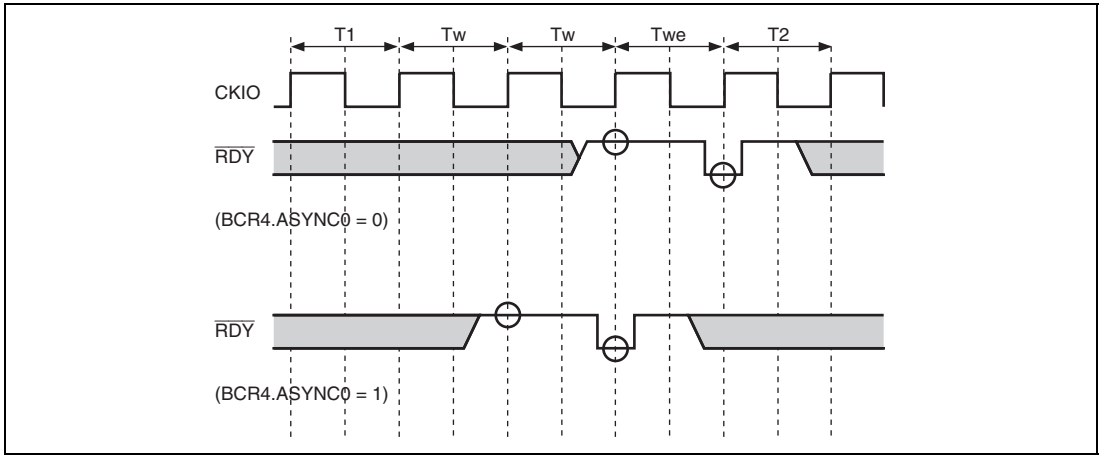


Figure 10.4 Example of $\overline{\text{RDY}}$ Sampling Timing

WCR1 is a 32-bit readable/writable register that specifies the number of idle state insertion cycles for each area. With some types of memory, data bus drive does not go off immediately after the off-chip read signal goes off. As a result, there is a possibility of a data bus collision when consecutive memory accesses are performed on memory in different areas, or when a memory write is performed immediately after a read. In this LSI, idle cycles corresponding to the number of cycles set in WCR1 are automatically inserted if there is a possibility of this kind of data bus collision.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	DMAIW2	DMAIW1	DMAIW0	-	A6IW2	A6IW1	A6IW0	-	A5IW2	A5IW1	A5IW0	-	A4IW2	A4IW1	A4IW0
Initial value:	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	A3IW2	A3IW1	A3IW0	-	A2IW2	A2IW1	A2IW0	-	A1IW2	A1IW1	A1IW0	-	A0IW2	A0IW1	A0IW0
Initial value:	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
30 29 28	DMAIW2 DMAIW1 DMAIW0	1 1 1	R/W R/W R/W	DMAIW-DACK Device Inter-Cycle Idle Setting These bits specify the number of idle cycles between bus cycles to be inserted when switching from a device with DACK to another space, or from a read access to a write access on the same device. The DMAIW bits are valid only for DMA single address transfer; with DMA dual address transfer, inter-area idle cycles specified by the AnIW2 to AnIW0 bits are inserted.
4n + 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

4n + 2	AnIW2	All 1	R/W	Area n Inter-Cycle Idle Setting
4n + 1	AnIW1	All 1	R/W	These bits specify the number of idle cycles between bus cycles to be inserted when switching from off-chip memory space area n to another space, or from a read access to a write access in the same space. For the type of idle cycles to be inserted, see table 10.7.
4n	AnIW0	All 1	R/W	

	Idle cycles to be inserted
000:	0
001:	1
010:	2
011:	3
100:	6
101:	9
110:	12
111:	15

Note: n = 0 to 6

Preceding Cycle	Following Cycle: Same Area: Read: CPU	Same Area: Read: DMA	Same Area: Write: CPU	Same Area: Write: DMA	Different Area: Read: CPU	Different Area: Read: DMA	Different Area: Write: CPU	Different Area: Write: DMA	Same Area: MPX Address Output	Different Area: MPX Address Output
Read			M	M	M	M	M	M	M (1)	M (1)
Write					M	M	M	M	*2	M
DMA read (memory → device)			M	M	M	M	M	M		M (1)
DMA write (device → memory)	D	D	D	D*1	D	D	D	D		D (1)

DMA in the table indicates DMA single-address transfer. DMA dual-address transfer is in accordance with the CPU.

M, D: Idle wait always inserted by WCR1

(M(1): One cycle inserted in MPX access even if WCR1 is cleared to 0)

M: Idle cycles according to setting of AnIW2 to AnIW0 (areas 0 to 6)

D: Idle cycles according to setting of DMAIW2 to DMAIW0

Notes: When synchronous DRAM is used in RAS down mode, bits DMAIW2 to DAMIW0 and bits A3IW2 to A3IW0 should be both 000.

1. Inserted when device is switched
2. On the MPX interface, a WCR1 idle wait may be inserted before an access (either read or write) to the same area after a write access. An example of idle wait insertion in accesses to the same area is shown below.

   (a) Synchronous DRAM set to RAS down mode

   (b) Synchronous DRAM accessed by on-chip DMAC

   Under conditions other than conditions (a) and (b) above, an idle wait is also inserted between an MPX interface write access and an immediately following access to the same area.

   Under above conditions (a) and (b), an idle wait may be inserted in a same-area access following an MPX interface write access, depending on the synchronous DRAM pipeline access situation. An idle wait is not inserted when the WCR1 setting is 0. The setting for the number of idle state cycles inserted after a power-on reset is the default value of 15 (the maximum value), so ensure that the optimum value is set.

WCR2 is a 32-bit readable/writable register that specifies the number of wait cycles to be inserted for each area. It also specifies the data access pitch when performing burst ROM memory access.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	A6W2	A6W1	A6W0	A6B2	A6B1	A6B0	A5W2	A5W1	A5W0	A5B2	A5B1	A5B0	A4W2	A4W1	A4W0	-
Initial value:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	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

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	A3W2	A3W1	A3W0	-	A2W2	A2W1	A2W0	A1W2	A1W1	A1W0	A0W2	A0W1	A0W0	A0B2	A0B1	A0B0
Initial value:	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1
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	R/W

Bit	Bit Name	Initial Value	R/W	Description
31	A6W2	1	R/W	Area 6 Wait Control
30	A6W1	1	R/W	These bits specify the number of wait cycles to be inserted for area 6. For the case where an MPX interface setting is made, see table 10.7.
29	A6W0	1	R/W	

	Inserted wait cycles	$\overline{\text{RDY}}$ pin
000:	0	Disabled
001:	1	Enabled
010:	2	Enabled
011:	3	Enabled
100:	6	Enabled
101:	9	Enabled
110:	12	Enabled
111:	15	Enabled

28 27 26	A6B2 A6B1 A6B0	1 1 1	R/W R/W R/W	Area 6 Burst Pitch These bits specify the number of wait cycles to be inserted for the second and following data accesses in burst transfer when area 6 is specified as burst ROM area.

	Inserted wait cycles	$\overline{\text{RDY}}$ pin
000:	0	Disabled
001:	1	Enabled
010:	2	Enabled
011:	3	Enabled
100:	4	Enabled
101:	5	Enabled
110:	6	Enabled
111:	7	Enabled

25 24 23	A5W2 A5W1 A5W0	1 1 1	R/W R/W R/W	Area 5 Wait Control These bits specify the number of wait cycles to be inserted for area 5. For the case where an MPX interface setting is made, see table 10.7.

	Inserted wait cycles	$\overline{\text{RDY}}$ pin
000:	0	Disabled
001:	1	Enabled
010:	2	Enabled
011:	3	Enabled
100:	6	Enabled
101:	9	Enabled
110:	12	Enabled
111:	15	Enabled

22 21 20	A5B2 A5B1 A5B0	1 1 1	R/W R/W R/W	Area 5 Burst Pitch These bits specify the number of wait cycles to be inserted for the second and following data accesses in burst transfer when area 5 is specified as burst ROM area. Inserted wait cycles — $\overline{RDY}$ pin 000: 0 — Disabled 001: 1 — Enabled 010: 2 — Enabled 011: 3 — Enabled 100: 4 — Enabled 101: 5 — Enabled 110: 6 — Enabled 111: 7 — Enabled
19 18 17	A4W2 A4W1 A4W0	1 1 1	R/W R/W R/W	Area 4 Wait Control These bits specify the number of wait cycles to be inserted for area 4. For the case where an MPX interface setting is made, see table 10.7. Inserted wait cycles — $\overline{RDY}$ pin 000: 0 — Disabled 001: 1 — Enabled 010: 2 — Enabled 011: 3 — Enabled 100: 6 — Enabled 101: 9 — Enabled 110: 12 — Enabled 111: 15 — Enabled
16	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

15 14 13	A3W2 A3W1 A3W0	1 1 1	R/W R/W R/W	Area 3 Wait Control These bits specify the number of wait cycles to be inserted for area 3. An external wait input is available for SRAM and MPX interfaces and is not available for synchronous DRAM interface. For the case where an MPX interface setting is made, see table 10.7. • When SRAM interface is in use: Inserted wait cycles / $\overline{\text{RDY}}$ pin 000: 0 Disabled 001: 1 Enabled 010: 2 Enabled 011: 3 Enabled 100: 6 Enabled 101: 9 Enabled 110: 12 Enabled 111: 15 Enabled • When synchronous DRAM interface is in use*[1]: Synchronous DRAM $\overline{\text{CAS}}$ latency cycles 000: Setting prohibited 001: 1*[2] 010: 2 011: 3 100: 4*[2] 101: 5*[2] 110: Setting prohibited 111: Setting prohibited
12	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

11 10 9	A2W2 A2W1 A2W0	1 1 1	R/W R/W R/W	Area 2 Wait Control These bits specify the number of wait cycles to be inserted for area 2. An external wait input is available for SRAM and MPX interfaces and is not available for synchronous DRAM interface. • When SRAM interface is in use: Inserted wait cycles — $\overline{RDY}$ pin 000: 0 — Disabled 001: 1 — Enabled 010: 2 — Enabled 011: 3 — Enabled 100: 6 — Enabled 101: 9 — Enabled 110: 12 — Enabled 111: 15 — Enabled • When synchronous DRAM interface is in use*1: Synchronous DRAM $\overline{CAS}$ latency cycles 000: Setting prohibited 001: 1*2 010: 2*2 011: 3*2 100: 4*2 101: 5*2 110: Setting prohibited 111: Setting prohibited

8 7 6	A1W2 A1W1 A1W0	1 1 1	R/W R/W R/W	Area 1 Wait Control These bits specify the number of wait cycles to be inserted for area 1. For the case where an MPX interface setting is made, see table 10.8.

	Inserted wait cycles	$\overline{\text{RDY}}$ pin
000:	0	Disabled
001:	1	Enabled
010:	2	Enabled
011:	3	Enabled
100:	6	Enabled
101:	9	Enabled
110:	12	Enabled
111:	15	Enabled

5 4 3	A0W2 A0W1 A0W0	1 1 1	R/W R/W R/W	Area 0 Wait Control These bits specify the number of wait cycles to be inserted for area 0. For the case where an MPX interface setting is made, see table 10.8.

	Inserted wait cycles	$\overline{\text{RDY}}$ pin
000:	0	Disabled
001:	1	Enabled
010:	2	Enabled
011:	3	Enabled
100:	6	Enabled
101:	9	Enabled
110:	12	Enabled
111:	15	Enabled

2	A0B2	1	R/W	Area 0 Burst Pitch
1	A0B1	1	R/W	These bits specify the number of wait cycles to be inserted for the second and following data accesses in burst transfer when area 0 is specified as burst ROM area.
0	A0B0	1	R/W	

	Inserted wait cycles	$\overline{\text{RDY}}$ pin
000:	0	Disabled
001:	1	Enabled
010:	2	Enabled
011:	3	Enabled
100:	4	Enabled
101:	5	Enabled
110:	6	Enabled
111:	7	Enabled

Notes: 1. External wait input is always ignored
2. Inhibited in RAS down mode

Table 10.8 MPX Interface Setting

			Description			
			Inserted Wait Cycles			
			1st Data		2nd Data	
AnW2	AnW1	AnW0	Read	Write	and After	$\overline{\text{RDY}}$ Pin
0	0	0	1	0	0	Enabled
0	0	1	1	1	0	Enabled
0	1	0	2	2	0	Enabled
0	1	1	3	3	0	Enabled
1	0	0	1	0	1	Enabled
1	0	1	1	1	1	Enabled
1	1	0	2	2	1	Enabled
1	1	1	3	3	1	Enabled

Note: n = 0 to 6

WCR3 is a 32-bit readable/writable register that specifies the cycles to be inserted for each area during the address setup time before the read/write strobe is asserted and during the data-hold time after the write strobe is negated.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	A6S0	A6H1	A6H0	-	A5S0	A5H1	A5H0	A4 RDH	A4S0	A4H1	A4H0
Initial value:	0	0	0	0	0	1	1	1	0	1	1	1	0	1	1	1
R/W:	R	R	R	R	R	R/W	R/W	R/W	R	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
	-	A3S0	A3H1	A3H0	-	A2S0	A2H1	A2H0	A1 RDH	A1S0	A1H1	A1H0	-	A0S0	A0H1	A0H0
Initial value:	0	1	1	1	0	1	1	1	1	1	1	1	0	1	1	1
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 27, 23, 15, 11, 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4n + 2	AnS0	All 1	R/W	Area n Write Strobe Setup Time Specifies the number of cycles to be inserted during the address setup time before the read/write strobe is asserted. Valid only for SRAM interface, byte control SRAM interface, and burst ROM interface: Cycles to be inserted during the setup time 0: 0 1: 1

4n + 1	AnH1*	All 1	R/W	Area n Data Hold Time
4n	AnH0*	All 1	R/W	For writing, specifies the number of cycles to be inserted during the data hold time after the write strobe is negated. For reading, specifies the number of cycles to be inserted during the data hold time after the data sampling timing. Valid only for SRAM interface, byte control SRAM interface, and burst ROM interface: Cycles to be inserted during the data hold time 00: 0 01: 1 10: 2 11: 3
4m + 3	AmRDH	All 0	R/W	Read-Strobe Negate Timing For reading, these bits specify the timing for the negation of read strobe. These bits should be cleared to 0 when byte control SRAM interface is in use. See figure 10.12.

Notes: n = 0 to 6; m = 1 and 4

* For area 1, only use the combinations listed in table 10.9 for the settings.

10.5.8 Wait Control Register 4 (WCR4)

WCR4 is a 32-bit readable/writable register that specifies the negation period for the $\overline{CS1}$ signal. Specifying bits CSH1 and CSH0 can insert the negation cycles from 0 to 3. If the $\overline{CS1}$ negate period is set, it must be set to match the WCR3 data hold time (A1H[1:0]). If the $\overline{CS1}$ negate period is not specified (CSH[1:0] = 00), there is no need to match the WCR3 data hold time (A1H[1:0]).

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
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
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	CSH1	CSH0
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/W	R/W

31 to 2	—	All 0	R	Reserved These bits are always read as 0, and the write value should always be 0.
1 0	CSH1* CSH0*	0 0	R/W R/W	CS Hold Cycle Setting Specifies the number of wait cycles inserted during data hold after $\overline{CS1}$ is negated. Wait cycles to be inserted 00: 0 01: 1 10: 2 11: 3 If a value other than 00 is set, set WCR3.A1RDH to 1.

Note: * Only use the combinations listed in table 10.9 for the settings.

Table 10.9 WCR3 and WCR4 Settings for Area 1

WCR3			WCR4	
A1RDH	**A1H1**	**A1H0**	**CSH1**	**CSH0**
0	0	0	0	0
		1		
	1	0		
		1		
1	0	0	0	0
		1		
	1	0		
		1		
	0	1	0	1
	1	0	1	0
		1		1

Note: Values other than the combinations listed above are illegal settings.

MCR is a 32-bit readable/writable register that specifies $\overline{RAS}$ and $\overline{CASS}$ timing, burst control, address multiplexing, and refresh control for synchronous DRAM (areas 2 and 3).

Write bits 31 to 3 when making the initial settings after a power-on reset and do not modify the settings from then onward. When writing to bits RFSH and RMODE, write the same values without changing other bits. When using synchronous DRAM, do not access areas 2 and 3 before register initialization is complete.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RASD	MR SET	TRC2	TRC1	TRC0	-	-	-	-	-	TPC2	TPC1	TPC0	-	RCD1	RCD0
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	R	R	R	R	R/W	R/W	R/W	R	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TRWL 2	TRWL 1	TRWL 0	TRAS 2	TRAS 1	TRAS 0	-	SZ1	SZ0	AMX EXT	AMX2	AMX1	AMX0	RFSH	RM ODE	-
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/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

31	RASD	0	R/W	RAS Down Sets RAS down mode. Do not set RAS down mode when specifying areas 2 and 3 as synchronous DRAM interface. 0: Normal mode 1: RAS down mode Note: When using synchronous DRAM in RAS down mode, set the DMAIW2 to DMAIW0 bits to 000 and A3IW2 to A3IW0 bits to 000.
30	MRSET	0	R/W	Mode Register Set Set this bit to 1 to make the mode register setting for synchronous DRAM. See the description of power-on sequence in section 10.6.4, (10) Power-On Sequence. 0: All-bank precharge 1: Mode register setting
29 28 27	TRC2 TRC1 TRC0	0 0 0	R/W R/W R/W	RAS Precharge Time after Refresh (Both auto- and self-refresh are available for synchronous DRAM) RAS precharge time immediately after refresh 000: 0 001: 3 010: 6 011: 9 100: 12 101: 15 110: 18 111: 21
26 to 22	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

21 20 19	TPC2 TPC1 TPC0	0 0 0	R/W R/W R/W	RAS Precharge Period When synchronous DRAM interface is in use, these bits specify the minimum number of cycles until the next bank active command is issued after precharging. RAS Precharge Time (SDRAM) 000: 1*[1] 001: 2 010: 3 011: 4*[1] 100: 5*[1] 101: 6*[1] 110: 7*[1] 111: 8*[1]
18	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
17 16	RCD1 RCD0	0 0	R/W R/W	RAS-CAS Delay When using the synchronous DRAM interface, specify ACTIVE to READ or WRITE delay in these bits. 00: Setting prohibited 01: 2 cycles 10: 3 cycles 11: 4 cycles*[1]

15 14 13	TRWL2 TRWL1 TRWL0	0 0 0	R/W R/W R/W	Write Precharge Delay Specify the synchronous DRAM write precharge delay in these bits. In auto-precharge mode, specify the time after a write cycle before the next bank active command is issued. After a write cycle, the next active command is not issued for a period of TPC + TRWL. In RAS down mode, specify the time after a write cycle before the next precharge command is issued. After a write cycle, the next precharge command is not issued for a period of TRWL. This setting is valid only when synchronous DRAM interface is in use. For details on these bit settings and the period in which no command is issued, refer to section 33.3.3, Bus Timing. Write Precharge ACT Delay Time 000: 1 001: 2 010: 3*[1] 011: 4*[1] 100: 5*[1] 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited
12 11 10	TRAS2 TRAS1 TRAS0	0 0 0	R/W R/W R/W	Refresh Period When the synchronous DRAM interface is in use, the bank active command is not issued for a period of TRC*[2] + TRAS after an auto-refresh command is issued. Command Issuance Gap after Synchronous DRAM Refresh 000: 4 + TRC 001: 5 + TRC 010: 6 + TRC 011: 7 + TRC 100: 8 + TRC 101: 9 + TRC 110: 10 + TRC 111: 11 + TRC

9	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
8 7	SZ1 SZ0	0 0	R/W R/W	Memory Data Size These bits specify the bus width of synchronous DRAM. This setting has priority over the BCR2 register setting. Synchronous DRAM 00: Setting prohibited 01: Setting prohibited 10: Setting prohibited 11: 32 bits
6 5 4 3	AMXEXT AMX2 AMX1 AMX0	0 0 0 0	R/W R/W R/W R/W	Address Multiplexing These bits specify address multiplexing for synchronous DRAM. For details, refer to appendix D, Address Multiplexing for Synchronous DRAM. Synchronous DRAM structure example / Bank 0000: (512k × 16 bits × 2) × 2 — a[21]*3 1000: (512k × 16 bits × 2) × 2 — a[20]*3 0001: (1M × 8 bits × 2) × 4 — a[22]*3 1001: (1M × 8 bits × 2) × 4 — a[21]*3 0010: (1M × 16 bits × 4) × 2 — a[23:22]*3 0011: (2M × 8 bits × 4) × 4 — a[24:23]*3 0100: (512k × 32 bits × 4) × 1 — a[22:21]*3 0101: (1M × 32 bits × 2) × 1 — a[22]*3 0110: (4M × 4 bits × 4) × 8 — a[25:24]*3 1110: (4M × 16 bits × 4) × 2 — a[25:24]*3 0111 (256k × 32 bits × 2) × 1 — a[20]*3 Other settings are prohibited.
2	RFSH	0	R/W	Refresh Control Specifies refresh control. Selects whether refreshing is performed for synchronous DRAM. When the refresh function is not used, the refresh request cycle generation timer can be used as an interval timer. 0: Refresh is not performed 1: Refresh is performed

1	RMODE	0	R/W	Refresh Mode Specifies whether normal refreshing or self-refreshing is performed when the RFSH bit is set to 1. When RFSH bit = 1 and RMODE = 0, auto-refreshing is performed for synchronous DRAM at the interval specified in refresh-related registers RTCNT, RTCOR, and RTCSR. If a refresh request is issued during an off-chip bus cycle, the refresh cycle is executed when the bus cycle ends. When RFSH = 1 and RMODE =1, if a refresh request is issued during an off-chip bus cycle, the synchronous DRAM waits until the bus cycle ends before entering the self-refresh state. All refresh requests for memory in the self-refresh state are ignored. 0: Auto refresh (when RFSH = 1) 1: Self refresh (when RFSH = 1)
0	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

Notes: 1. Inhibited in RAS down mode
2. Bits 29 to 27. RAS precharge period after refresh
3. a[x]: Off-chip address; not address pin

10.5.10 PCMCIA Control Register (PCR)

PCR is a 16-bit readable/writable register that specifies the $\overline{OE}$ and $\overline{WE}$ signal assertion/negation timing for areas 5 and 6 specified as the PCMCIA interface. The $\overline{OE}$ and $\overline{WE}$ signal assertion width is specified by the wait control bits in WCR2.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	A5 PCW1	A5 PCW0	A6 PCW1	A6 PCW0	A5 TED2	A5 TED1	A5 TED0	A6 TED2	A6 TED1	A6 TED0	A5 TEH2	A5 TEH1	A5 TEH0	A6 TEH2	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

15 14	A5PCW1 A5PCW0	0 0	R/W R/W	PCMCIA Wait A5 In a low-speed PCMCIA wait cycle, these bits specify the number of wait cycles to be added to the number of wait cycles specified by WCR2. The setting of these bits is selected when the PCMCIA interface access TC bit is 0. Wait cycles to be inserted 00: 0 01: 15 10: 30 11: 50
13 12	A6PCW1 A6PCW0	0 0	R/W R/W	PCMCIA Wait A6 In a low-speed PCMCIA wait cycle, these bits specify the number of wait cycles to be added to the number of wait cycles specified by WCR2. The setting of these bits is selected when the PCMCIA interface access TC bit is 1. Wait cycles to be inserted 00: 0 01: 15 10: 30 11: 50
11 10 9	A5TED2 A5TED1 A5TED0	0 0 0	R/W R/W R/W	Address-$\overline{OE}$/$\overline{WE}$ Assertion Delay A5 These bits set the delay time from address output to $\overline{OE}$/$\overline{WE}$ assertion in the connected PCMCIA interface. The setting of these bits is selected when the PCMCIA interface access TC bit is 0. Wait cycles to be inserted 000: 0 001: 1 010: 2 011: 3 100: 6 101: 9 110: 12 111: 15

8 7 6	A6TED2 A6TED1 A6TED0	0 0 0	R/W R/W R/W	Address-$\overline{OE}$/$\overline{WE}$ Assertion Delay A6 These bits set the delay time from address output to $\overline{OE}$/$\overline{WE}$ assertion in the connected PCMCIA interface. The setting of these bits is selected when the PCMCIA interface access TC bit is 1. Wait cycles to be inserted 000: 0 001: 1 010: 2 011: 3 100: 6 101: 9 110: 12 111: 15
5 4 3	A5TEH2 A5TEH1 A5TEH0	0 0 0	R/W R/W R/W	$\overline{OE}$/$\overline{WE}$ Negation-Address Delay A5 These bits set the address hold delay time after $\overline{OE}$/$\overline{WE}$ negation in the connected PCMCIA interface. PCMCIA interface. The setting of these bits is selected when the PCMCIA interface access TC bit is 0. Wait cycles to be inserted 000: 0 001: 1 010: 2 011: 3 100: 6 101: 9 110: 12 111: 15

2	A6TEH2	0	R/W	$\overline{OE}$/$\overline{WE}$ Negation-Address Delay A6
1	A6TEH1	0	R/W	These bits set the address hold delay time after $\overline{OE}$/$\overline{WE}$ negation in the connected PCMCIA interface. The setting of these bits is selected when the PCMCIA interface access TC bit is 1.
0	A6TEH0	0	R/W	

	Wait cycles to be Inserted
000:	0
001:	1
010:	2
011:	3
100:	6
101:	9
110:	12
111:	15

10.5.11 Synchronous DRAM Mode Register (SDMR)

SDMR is a 16-bit write-only virtual register that is written to via the synchronous DRAM address bus, and sets the mode of the area 2 and area 3 synchronous DRAM.

Settings for the SDMR register must be made before accessing synchronous DRAM.

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

Since the address bus, not the data bus, is used to write to the synchronous DRAM mode register, if the value to be set is "X" and the SDMR register address is "Y", value "X" is written to the synchronous DRAM mode register by performing a write to address X + Y. When the synchronous DRAM bus width is set to 32 bits, as A0 of the synchronous DRAM is connected to A2 of this LSI, and A1 of the synchronous DRAM is connected to A3 of this LSI, the value actually written to the synchronous DRAM is the value of "X" shifted 2 bits to the right.

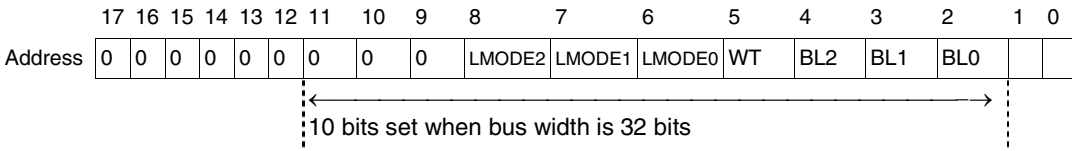
For example, to write H'0230 to SDMR in area 2, arbitrary data is written to address H'FF90 0000 (address "Y") + H'08C0 (value "X") (= H'FF90 08C0). As a result, H'0230 is written to the SDMR register. The range of value "X" is H'0000 to H'0FFC.

SDMR register. The range of value "X" is H'0000 to H'0FFC.

The lower 16 bits of the address are set in the synchronous DRAM mode register. The burst length is 4 and 8. Setting to SDMR writes into the following addresses in bytes.

Bus Width	Burst Length	CAS Latency	Area 2	Area 3
32	4	1	H'FF90 0048	H'FF94 0048
		2	H'FF90 0088	H'FF94 0088
		3	H'FF90 00C8	H'FF94 00C8
32	8	1	H'FF90 004C	H'FF94 004C
		2	H'FF90 008C	H'FF94 008C
		3	H'FF90 00CC	H'FF94 00CC

For a 32-bit bus:



LMODE: RAS-CAS latency
BL: Burst length
WT: Wrap type (0: Sequential)

BL	LMODE
000: Setting prohibited	000: Setting prohibited
001: Setting prohibited	001: 1
010: 4	010: 2
011: 8	011: 3
100: Setting prohibited	100: Setting prohibited
101: Setting prohibited	101: Setting prohibited
110: Setting prohibited	110: Setting prohibited
111: Setting prohibited	111: Setting prohibited

RTCSR is a 16-bit readable/writable register that specifies the refresh cycle and whether interrupts are to be generated.

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/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	—	All 0	—	Reserved These bits are always read as 0.
7	CMF	0	R/W	Compare-Match Flag Status flag that indicates a match between RTCNT and RTCOR values. 0: RTCNT and RTCOR values do not match [Clearing condition] When 0 is written to CMF 1: RTCNT and RTCOR values match [Setting condition] When RTCNT = RTCOR*
6	CMIE	0	R/W	Compare-Match Interrupt Enable Controls whether or not an interrupt request is enabled when the CMF bit is set to 1 in RTCSR. Do not set this bit to 1 when auto-refreshing is used. 0: Interrupt request by CMF disabled 1: Interrupt request by CMF enabled

5 4 3	CKS2 CKS1 CKS0	0 0 0	R/W R/W R/W	Clock Select Bits These bits select the input clock for RTCNT. The source clock is the off-chip bus clock (CKIO). The RTCNT count clock is obtained by dividing CKIO by the specified factor. 000: Clock input disabled 001: CKIO/4 010: CKIO/16 011: CKIO/64 100: CKIO/256 101: CKIO/1024 110: CKIO/2048 111: CKIO/4096
2	OVF	0	R/W	Refresh Count Overflow Flag Status flag that indicates that the number of refresh cycles in RFCR has exceeded the number specified by the LMTS bit in RTCSR. 0: RFCR has not overflowed the count limit indicated by LMTS [Clearing condition] When 0 is written to OVF 1: RFCR has overflowed the count limit specified by LMTS [Setting condition] When RFCR overflows the count limit specified by LMTS*
1	OVIE	0	R/W	Refresh Count Overflow Interrupt Enable Controls whether or not an interrupt request is enabled when the OVF bit is set to 1 in RTCSR. 0: Interrupt requests by OVF disabled 1: Interrupt requests by OVF enabled
0	LMTS	0	R/W	Refresh Count Overflow Limit Select Specifies the count limit to be compared with the RFCR value. If the RFCR value exceeds the value specified by LMTS, the OVF bit is set. 0: Count limit is 1024 1: Count limit is 512

Note: * If 1 is written, the original value is retained.

RTCNT is an 8-bit readable/writable counter that is incremented by the input clock (selected by bits CKS2 to CKS0 in RTCSR). When the RTCNT value matches the RTCOR value, the CMF bit is set in RTCSR and the RTCNT value is cleared.

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

10.5.14 Refresh Time Constant Register (RTCOR)

RTCOR is a readable/writable register that specifies the upper limit of RTCNT. The RTCOR and RTCNT values (lower 8 bits) are constantly compared, and when they match, the CMF bit is set in RTCSR and the RTCNT value is cleared to 0. If the RFSH bit in MCR has been set to 1 and auto-refresh has been selected as the refresh mode, a memory refresh cycle is generated when the CMF bit is set.

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

10.5.15 Refresh Count Register (RFCR)

RFCR is a 10-bit readable/writable counter that counts the number of refresh cycles by being incremented each time the RTCOR and RTCNT values match. If the RFCR value exceeds the count limit specified by the LMTS bit in RTCSR, the OVF bit in RTCSR is set and the RFCR value is cleared.

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

RTCSR, RTCNT, RTCOR, and RFCR 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. Perform reads and writes using the following methods:

(1) Write to RTCSR, RTCNT, RTCOR, or RFCR

When writing to RTCSR, RTCNT, RTCOR, and RFCR, use only word transfer instructions. They cannot be written to 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 six bits and the write data in the remaining bits, as shown in figure 10.5.

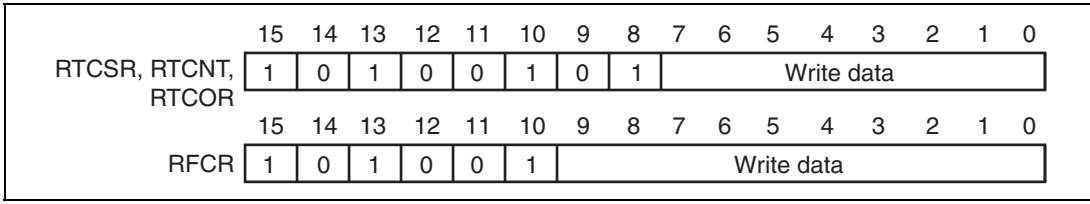


Figure 10.5 Write to RTCSR, RTCNT, RTCOR, or RFCR

(2) Read from RTCSR, RTCNT, RTCOR, or RFCR

RTCSR, RTCNT, RTCOR, or RFCR should be read in 16-bit units. Each reserved bit is always read as 0.

10.6.1 Endian/Access Size and Data Alignment

This LSI supports both big-endian mode, in which upper byte in a string of byte data is at an address 0, and little-endian mode, in which lower byte in a string of byte data is at an address 0. The mode is specified by the MD5 pin at a power-on reset by the $\overline{\text{RESET}}$ pin. When the MD5 pin is low, big-endian mode is specified, and when the MD5 pin is high, little-endian mode is specified.

A data bus width of 8, 16, or 32 bits can be selected for the normal memory interface, 32 bits for the synchronous DRAM interface, and 8 or 16 bits for the PCMCIA interface. Data alignment is carried out according to the data bus width and endian mode of each device. Accordingly, when the data bus width is narrower than the access size, multiple bus cycles are automatically generated to reach the access size. In this case, access is performed by automatically incrementing addresses to the bus width. For example, when a long word access is performed at the area with an 8-bit bus width in the SRAM interface, each address is incremented one by one, and then access is performed four times. In the 32-byte transfer, a total of 32-byte data is continuously transferred according to the set bus width. The first access is performed on the data for which there was an access request, and the remaining accesses are performed using wraparound on 32-byte boundary data. During these transfers, the bus is not released and refresh operation is not performed. In this LSI, data alignment and data length conversion between different interfaces is performed automatically. Quadword access is used only in transfer by the DMAC.

The relationships between the endian mode, device data length, and access unit are shown in tables 10.10 to 10.15.

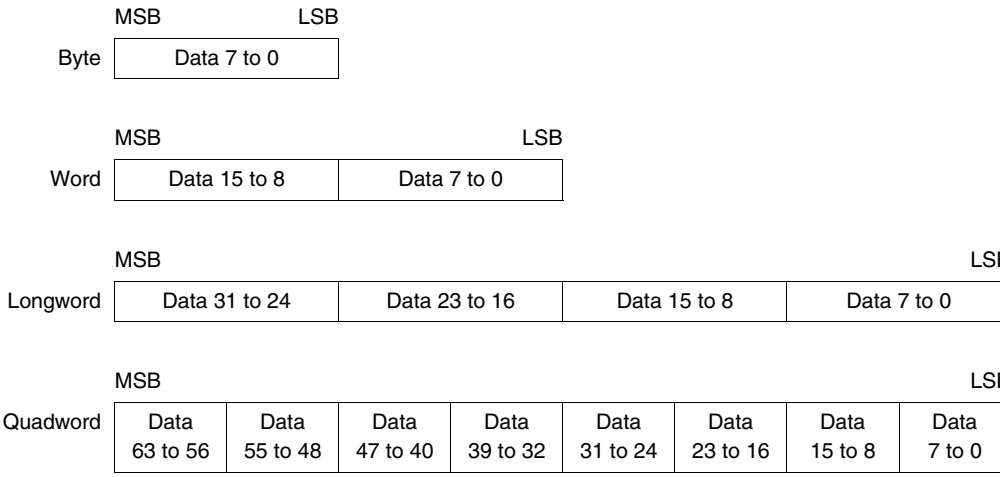


Table 10.10 32-Bit Off-chip Device/Big-Endian Access and Data Alignment

Operation			Data Bus				Strobe Signals			
Access Size	Address	No.	D31 to D24	D23 to D16	D15 to D8	D7 to D0	$\overline{WE3}$, DQM3	$\overline{WE2}$, DQM2	$\overline{WE1}$, DQM1	$\overline{WE0}$, DQM0
Byte	4n	1	Data 7 to 0	—	—	—	Asserted			
	4n+1	1	—	Data 7 to 0	—	—		Asserted		
	4n+2	1	—	—	Data 7 to 0	—			Asserted	
	4n+3	1	—	—	—	Data 7 to 0				Asserted
Word	4n	1	Data 15 to 8	Data 7 to 0	—	—	Asserted	Asserted		
	4n+2	1	—	—	Data 15 to 8	Data 7 to 0			Asserted	Asserted
Long-word	4n	1	Data 31 to 24	Data 23 to 16	Data 15 to 8	Data 7 to 0	Asserted	Asserted	Asserted	Asserted
Quad-word	8n	1	Data 63 to 56	Data 55 to 48	Data 47 to 40	Data 39 to 32	Asserted	Asserted	Asserted	Asserted
	8n+4	2	Data 31 to 24	Data 23 to 16	Data 15 to 8	Data 7 to 0	Asserted	Asserted	Asserted	Asserted

Operation			Data Bus				Strobe Signals			
Access Size	Address	No.	D31 to D24	D23 to D16	D15 to D8	D7 to D0	$\overline{WE3}$, DQM3	$\overline{WE2}$, DQM2	$\overline{WE1}$, DQM1	$\overline{WE0}$, DQM0
Byte	2n	1	—	—	Data 7 to 0	—			Asserted	
	2n+1	1	—	—	—	Data 7 to 0				Asserted
Word	2n	1	—	—	Data 15 to 8	Data 7 to 0			Asserted	Asserted
Long-word	4n	1	—	—	Data 31 to 24	Data 23 to 16			Asserted	Asserted
	4n+2	2	—	—	Data 15 to 8	Data 7 to 0			Asserted	Asserted
Quad-word	8n	1	—	—	Data 63 to 56	Data 55 to 48			Asserted	Asserted
	8n+2	2	—	—	Data 47 to 40	Data 39 to 32			Asserted	Asserted
	8n+4	3	—	—	Data 31 to 24	Data 23 to 16			Asserted	Asserted
	8n+6	4	—	—	Data 15 to 8	Data 7 to 0			Asserted	Asserted

Operation			Data Bus				Strobe Signals			
Access Size	**Address**	**No.**	**D31 to D24**	**D23 to D16**	**D15 to D8**	**D7 to D0**	**$\overline{WE3}$, DQM3**	**$\overline{WE2}$, DQM2**	**$\overline{WE1}$, DQM1**	**$\overline{WE0}$, DQM0**
Byte	n	1	—	—	—	Data 7 to 0				Asserted
Word	2n	1	—	—	—	Data 15 to 8				Asserted
	2n+1	2	—	—	—	Data 7 to 0				Asserted
Long-word	4n	1	—	—	—	Data 31 to 24				Asserted
	4n+1	2	—	—	—	Data 23 to 16				Asserted
	4n+2	3	—	—	—	Data 15 to 8				Asserted
	4n+3	4	—	—	—	Data 7 to 0				Asserted
Quad-word	8n	1	—	—	—	Data 63 to 56				Asserted
	8n+1	2	—	—	—	Data 55 to 48				Asserted
	8n+2	3	—	—	—	Data 47 to 40				Asserted
	8n+3	4	—	—	—	Data 39 to 32				Asserted
	8n+4	5	—	—	—	Data 31 to 24				Asserted
	8n+5	6	—	—	—	Data 23 to 16				Asserted
	8n+6	7	—	—	—	Data 15 to 8				Asserted
	8n+7	8	—	—	—	Data 7 to 0				Asserted

Operation			Data Bus				Strobe Signals			
Access Size	**Address**	**No.**	**D31 to D24**	**D23 to D16**	**D15 to D8**	**D7 to D0**	**$\overline{WE3}$, DQM3**	**$\overline{WE2}$, DQM2**	**$\overline{WE1}$, DQM1**	**$\overline{WE0}$, DQM0**
Byte	4n	1	—	—	—	Data 7 to 0				Asserted
	4n+1	1	—	—	Data 7 to 0	—			Asserted	
	4n+2	1	—	Data 7 to 0	—	—		Asserted		
	4n+3	1	Data 7 to 0	—	—	—	Asserted			
Word	4n	1	—	—	Data 15 to 8	Data 7 to 0			Asserted	Asserted
	4n+2	1	Data 15 to 8	Data 7 to 0	—	—	Asserted	Asserted		
Long-word	4n	1	Data 31 to 24	Data 23 to 16	Data 15 to 8	Data 7 to 0	Asserted	Asserted	Asserted	Asserted
Quad-word	8n	1	Data 31 to 24	Data 23 to 16	Data 15 to 8	Data 7 to 0	Asserted	Asserted	Asserted	Asserted
	8n+4	2	Data 63 to 56	Data 55 to 48	Data 47 to 40	Data 39 to 32	Asserted	Asserted	Asserted	Asserted

Operation			Data Bus				Strobe Signals			
Access Size	Address	No.	D31 to D24	D23 to D16	D15 to D8	D7 to D0	$\overline{WE3}$, DQM3	$\overline{WE2}$, DQM2	$\overline{WE1}$, DQM1	$\overline{WE0}$, DQM0
Byte	2n	1	—	—	—	Data 7 to 0				Asserted
	2n+1	1	—	—	Data 7 to 0	—			Asserted	
Word	2n	1	—	—	Data 15 to 8	Data 7 to 0			Asserted	Asserted
Long-word	4n	1	—	—	Data 15 to 8	Data 7 to 0			Asserted	Asserted
	4n+2	2	—	—	Data 31 to 24	Data 23 to 16			Asserted	Asserted
Quad-word	8n	1	—	—	Data 15 to 8	Data 7 to 0			Asserted	Asserted
	8n+2	2	—	—	Data 31 to 24	Data 23 to 16			Asserted	Asserted
	8n+4	3	—	—	Data 47 to 40	Data 39 to 32			Asserted	Asserted
	8n+6	4	—	—	Data 63 to 56	Data 55 to 48			Asserted	Asserted

Operation			Data Bus				Strobe Signals			
Access Size	Address	No.	D31 to D24	D23 to D16	D15 to D8	D7 to D0	$\overline{WE3}$, DQM3	$\overline{WE2}$, DQM2	$\overline{WE1}$, DQM1	$\overline{WE0}$, DQM0
Byte	n	1	—	—	—	Data 7 to 0				Asserted
Word	2n	1	—	—	—	Data 7 to 0				Asserted
	2n+1	2	—	—	—	Data 15 to 8				Asserted
Long-word	4n	1	—	—	—	Data 7 to 0				Asserted
	4n+1	2	—	—	—	Data 15 to 8				Asserted
	4n+2	3	—	—	—	Data 23 to 16				Asserted
	4n+3	4	—	—	—	Data 31 to 24				Asserted
Quad-word	8n	1	—	—	—	Data 7 to 0				Asserted
	8n+1	2	—	—	—	Data 15 to 8				Asserted
	8n+2	3	—	—	—	Data 23 to 16				Asserted
	8n+3	4	—	—	—	Data 31 to 24				Asserted
	8n+4	5	—	—	—	Data 39 to 32				Asserted
	8n+5	6	—	—	—	Data 47 to 40				Asserted
	8n+6	7	—	—	—	Data 55 to 48				Asserted
	8n+7	8	—	—	—	Data 63 to 56				Asserted

(1) Area 0

For area 0, off-chip address bits A28 to A26 are 000.

The interfaces that can be set for this area are SRAM, MPX, and burst ROM.

A bus width of 8, 16, or 32 bits is selectable with pins MD4 and MD3 at a power-on reset caused by the $\overline{\text{RESET}}$ pin.

When area 0 is accessed, the $\overline{\text{CS0}}$ signal is asserted. In addition, the $\overline{\text{RD}}$ signal, which can be used as $\overline{\text{OE}}$, and write control signals $\overline{\text{WE0}}$ to $\overline{\text{WE3}}$ are asserted.

As regards the number of bus cycles, 0 to 15 wait cycles can be selected with bits A0W2 to A0W0 in WCR2. In addition, any number of wait cycles can be inserted in each bus cycle by the external wait pin ($\overline{\text{RDY}}$).

When the burst ROM interface is in use, the number of bus cycles for burst transfer is selected in the range of 2 to 9 according to the number of wait cycles.

The setup time of the address and $\overline{\text{CS}}$ signal with respect to the read/write strobe can be specified by bit A0S0 in WCR3 within a range of 0 to 1 cycle. The data-hold time of the address and $\overline{\text{CS}}$ signal with respect to the read/write strobe can be specified by bits A0H1 and A0H0 within a range of 0 to 3 cycles.

(2) Area 1

For area 1, off-chip address bits A28 to A26 are 001.

The interfaces that can be set for this area are SRAM, MPX, and byte control SRAM.

When the SRAM interface is in use, a bus width of 8, 16, or 32 bits is selectable with bits A1SZ1 and A1SZ0 in BCR2. When the MPX interface is in use, a bus width of 32 bits should be selected by bits A1SZ1 and A1SZ0 in BCR2. When the byte control SRAM interface is in use, select a bus width of 16 or 32 bits.

When area 1 is accessed, the $\overline{\text{CS1}}$ signal is asserted. In addition, the $\overline{\text{RD}}$ signal, which can be used as $\overline{\text{OE}}$, and write control signals $\overline{\text{WE0}}$ to $\overline{\text{WE3}}$ are asserted.

As regards the number of bus cycles, 0 to 15 wait cycles is selectable with bits A1W2 to A1W0 in WCR2. In addition, any number of wait cycles can be inserted in each bus cycle by the external wait pin ($\overline{\text{RDY}}$).

by bit A1S0 in WCR3 within a range of 0 to 1 cycle. The data-hold time of the address and CS signal with respect to the read/write strobe can be specified by bits A1H1 and A1H0 within a range of 0 to 3 cycles.

(3) Area 2

For area 2, off-chip address bits A28 to A26 are 010.

The interfaces that can be set for this area are SRAM, MPX, and synchronous DRAM.

When the SRAM interface is in use, a bus width of 8, 16, or 32 bits is selectable with bits A2SZ1 and A2SZ0 in BCR2. When the MPX interface is in use, a bus width of 32 bits should be selected by bits A2SZ1 and A2SZ0 in BCR2. When the synchronous DRAM interface is in use, select 32 bits by the SZ bits in MCR.

When area 2 is accessed, the $\overline{\text{CS2}}$ signal is asserted. When the SRAM interface is in use, the $\overline{\text{RD}}$ signal, which can be used as $\overline{\text{OE}}$, and write control signals $\overline{\text{WE0}}$ to $\overline{\text{WE3}}$ are asserted.

As regards the number of bus cycles, 0 to 15 wait cycles is selectable with bits A2W2 to A2W0 in WCR2. In addition, any number of wait cycles can be inserted in each bus cycle by the external wait pin ($\overline{\text{RDY}}$).

The setup time of the address and $\overline{\text{CS}}$ signal with respect to the read/write strobe can be specified by bit A2S0 in WCR3 within a range of 0 to 1 cycle. The data-hold time of the address and $\overline{\text{CS}}$ signal with respect to the read/write strobe can be specified by bits A2H1 and A2H0 within a range of 0 to 3 cycles.

When the synchronous DRAM interface is in use, the $\overline{\text{RAS}}$, $\overline{\text{CAS}}$, and RD/$\overline{\text{WR}}$ signals and byte control signals DQM0 to DQM3 are asserted, and address multiplexing is performed. Timing control for signals $\overline{\text{RAS}}$, $\overline{\text{CAS}}$, and data, and address multiplexing control, can be specified by MCR.

(4) Area 3

For area 3, off-chip address bits A28 to A26 are 011.

The interfaces that can be set for this area are SRAM, MPX, and synchronous DRAM.

When the SRAM interface is in use, a bus width of 8, 16, or 32 bits is selectable with bits A3SZ1 and A3SZ0 in BCR2. When the MPX interface is in use, a bus width of 32 bits should be selected by bits A3SZ1 and A3SZ0 in BCR2. When the synchronous DRAM interface is in use, select 32 bits by the SZ bits in MCR.

signal, which can be used as OE, and write control signals WE0 to WE3 are asserted.

As regards the number of bus cycles, 0 to 15 wait cycles is selectable with bits A3W2 to A3W0 in WCR2. In addition, any number of wait cycles can be inserted in each bus cycle by the external wait pin ($\overline{\text{RDY}}$).

The setup time of the address and $\overline{\text{CS}}$ signal with respect to the read/write strobe can be specified by bit A3S0 in WCR3 within a range of 0 to 1 cycle. The data-hold time of the address and $\overline{\text{CS}}$ signal with respect to the read/write strobe can be specified by bits A3H1 and A3H0 within a range of 0 to 3 cycles.

When the synchronous DRAM interface is in use, the $\overline{\text{RAS}}$, $\overline{\text{CAS}}$, and RD/$\overline{\text{WR}}$ signals and byte control signals DQM0 to DQM3 are asserted, and address multiplexing is performed. Timing control for signals $\overline{\text{RAS}}$, $\overline{\text{CAS}}$, and data, and address multiplexing control, can be specified by MCR.

(5) Area 4

For area 4, off-chip address bits A28 to A26 are 100.

The interfaces that can be set for this area are SRAM, MPX, and byte control SRAM.

When the SRAM interface is in use, a bus width of 8, 16, or 32 bits is selectable with bits A4SZ1 and A4SZ0 in BCR2. When the MPX interface is in use, a bus width of 32 bits should be selected by bits A4SZ1 and A4SZ0 in BCR2. When the byte control SRAM interface is in use, select a bus width of 16 or 32 bits.

When area 4 is accessed, the $\overline{\text{CS4}}$ signal is asserted. In addition, the $\overline{\text{RD}}$ signal, which can be used as $\overline{\text{OE}}$, and write control signals $\overline{\text{WE0}}$ to $\overline{\text{WE3}}$ are asserted.

As regards the number of bus cycles, 0 to 15 wait cycles is selectable with bits A4W2 to A4W0 in WCR2. In addition, any number of wait cycles can be inserted in each bus cycle by the external wait pin ($\overline{\text{RDY}}$).

The setup time of the address and $\overline{\text{CS}}$ signal with respect to the read/write strobe can be specified by bit A4S0 in WCR3 within a range of 0 to 1 cycle. The data-hold time of the address and $\overline{\text{CS}}$ signal with respect to the read/write strobe can be specified by bits A4H1 and A4H0 within a range of 0 to 3 cycles.

For area 5, off-chip address bits A28 to A26 are 101.

The interfaces that can be set for this area are SRAM, MPX, burst ROM, and PCMCIA.

When the SRAM interface is in use, a bus width of 8, 16, or 32 bits is selectable with bits A5SZ1 and A5SZ0 in BCR2. When the burst ROM interface is in use, a bus width of 8, 16, or 32 bits is selectable with bits A5SZ1 and A5SZ0 in BCR2. When the MPX interface is in use, a bus width of 32 bits should be selected by bits A5SZ1 and A5SZ0 in BCR2. When the PCMCIA interface is in use, select 8 or 16 bits by bits A5SZ1 and A5SZ0 in BCR2.

When area 5 is accessed while the SRAM interface is in use, the $\overline{\text{CS5}}$ signal is asserted. the $\overline{\text{RD}}$ signal, which can be used as $\overline{\text{OE}}$, and write control signals $\overline{\text{WE0}}$ to $\overline{\text{WE3}}$ are asserted. While the PCMCIA interface is in use, the $\overline{\text{CE1A}}$ and $\overline{\text{CE2A}}$ signals, the $\overline{\text{RD}}$ signal, which can be used as $\overline{\text{OE}}$, the $\overline{\text{WE1}}$, $\overline{\text{WE2}}$, $\overline{\text{WE3}}$, and $\overline{\text{WE0}}$ signals, which can be used as $\overline{\text{WE}}$, $\overline{\text{ICIORD}}$, $\overline{\text{ICIOWR}}$, and $\overline{\text{REG}}$, respectively, are asserted.

As regards the number of bus cycles, 0 to 15 wait cycles is selectable with bits A5W2 to A5W0 in WCR2. In addition, any number of wait cycles can be inserted in each bus cycle by the external wait pin ($\overline{\text{RDY}}$).

When the burst ROM interface is in use, the number of bus cycles for burst transfer is selected in the range of 2 to 9 according to the number of wait cycles.

The setup time of the address and CS signal with respect to the read/write strobe can be specified by bit A5S0 in WCR3 within a range of 0 to 1 cycle. The data-hold time of the address and CS signal with respect to the read/write strobe can be specified by bits A5H1 and A5H0 within a range of 0 to 3 cycles.

For a PCMCIA interface, the setup time of the address, $\overline{\text{CE1A}}$, and $\overline{\text{CE2A}}$ signals with respect to the read/write strobe can be specified by bits A5TED1 and A5TED0 in PCR within a range of 0 to 15 cycles. The hold time of the address, $\overline{\text{CE1A}}$, and $\overline{\text{CE2A}}$ signals can be specified by bits A5TEH1 and A5TEH0 in PCR within a range of 0 to 15 cycles. The number of wait cycles can be specified by bits A5PCW1 and A5PCW0 within a range of 0 to 50 cycles. The number of wait cycles specified by PCR is added to the value specified by WCR2.

For area 6, off-chip address bits A28 to A26 are 110.

The interfaces that can be set for this area are SRAM, MPX, burst ROM, and PCMCIA.

When the SRAM interface is in use, a bus width of 8, 16, or 32 bits is selectable with bits A6SZ1 and A6SZ0 in BCR2. When the burst ROM interface is in use, a bus width of 8, 16, or 32 bits is selectable with bits A6SZ1 and A6SZ0 in BCR2. When the MPX interface is in use, a bus width of 32 bits should be selected by bits A6SZ1 and A6SZ0 in BCR2. When the PCMCIA interface is in use, select 8 or 16 bits by bits A6SZ1 and A6SZ0 in BCR2.

When area 6 is accessed while the SRAM interface is in use, the $\overline{\text{CS6}}$ signal is asserted. the $\overline{\text{RD}}$ signal, which can be used as $\overline{\text{OE}}$, and write control signals $\overline{\text{WE0}}$ to $\overline{\text{WE3}}$ are asserted. While the PCMCIA interface is in use, the $\overline{\text{CE1B}}$ and $\overline{\text{CE2B}}$ signals, the $\overline{\text{RD}}$ signal, which can be used as $\overline{\text{OE}}$, the $\overline{\text{WE1}}$, $\overline{\text{WE2}}$, $\overline{\text{WE3}}$, and $\overline{\text{WE0}}$ signals, which can be used as $\overline{\text{WE}}$, $\overline{\text{ICIORD}}$, $\overline{\text{ICIOWR}}$, and $\overline{\text{REG}}$, respectively, are asserted.

As regards the number of bus cycles, 0 to 15 wait cycles is selectable with bits A6W2 to A6W0 in WCR2. In addition, any number of wait cycles can be inserted in each bus cycle by the external wait pin ($\overline{\text{RDY}}$).

When the burst ROM interface is in use, the number of bus cycles for burst transfer is selected in the range of 2 to 9 according to the number of wait cycles.

The setup time of the address and $\overline{\text{CS}}$ signal with respect to the read/write strobe can be specified by bit A6S0 in WCR3 within a range of 0 to 1 cycle. The data-hold time of the address and $\overline{\text{CS}}$ signal with respect to the read/write strobe can be specified by bits A6H1 and A6H0 within a range of 0 to 3 cycles.

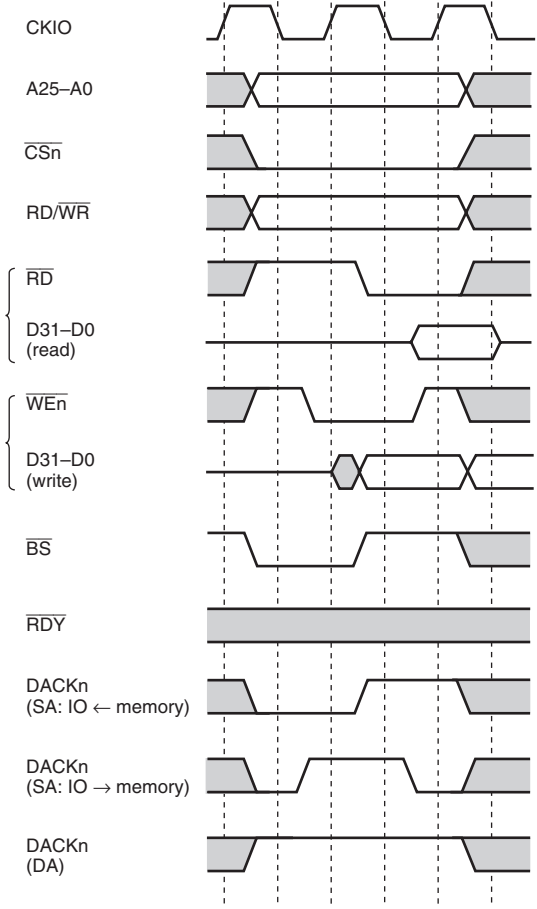
The setup time of the address, $\overline{\text{CE1B}}$, and $\overline{\text{CE2B}}$ signals with respect to the read/write strobe can be specified by bits A6TED1 and A6TED0 in PCR within a range of 0 to 15 cycles. The hold time of the address, $\overline{\text{CE1B}}$, and $\overline{\text{CE2B}}$ signals can be specified by bits A6TEH1 and A6TEH0 in PCR within a range of 0 to 15 cycles. The number of wait cycles can be specified by bits A6PCW1 and A6PCW0 within a range of 0 to 50 cycles. The number of wait cycles specified by PCR is added to the value specified by WCR2.

(1) Basic Timing

The strobe signals for the SRAM interface of this LSI are output primarily based on the SRAM connection. Figure 10.6 shows the basic timing of SRAM interface. 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 asserted at the rising edge of the clock in the T1 state, and negated at the rising edge of the clock after the T2 state. Therefore, there is no negation period in the case of access at minimum pitch.

When reading, specifying an access size is not needed. The output addresses on the address pins (A25 to A0) are correct, but since the access size is not specified, 32-bit data is always output when a 32-bit device is in use, and 16-bit data is output when a 16-bit device is in use. When writing, only the $\overline{WEn}$ signal corresponding to the byte to be written is asserted. For details, see section 10.6.1, Endian/Access Size and Data Alignment.

In 32-byte transfer, a total of 32 bytes are transferred consecutively according to the set bus width. The first access is performed on the data for which there was an access request, and the remaining accesses are performed in wraparound mode on the data at the 32-byte boundary. The bus is not released during this transfer.



SA: Single address DMA
DA: Dual address DMA

Note: For DACKn, an example is shown where the acknowledge level (AL) bit in CHCRn of the DMAC is cleared to 0.

Figure 10.6 Basic Timing of SRAM Interface

SRAM.

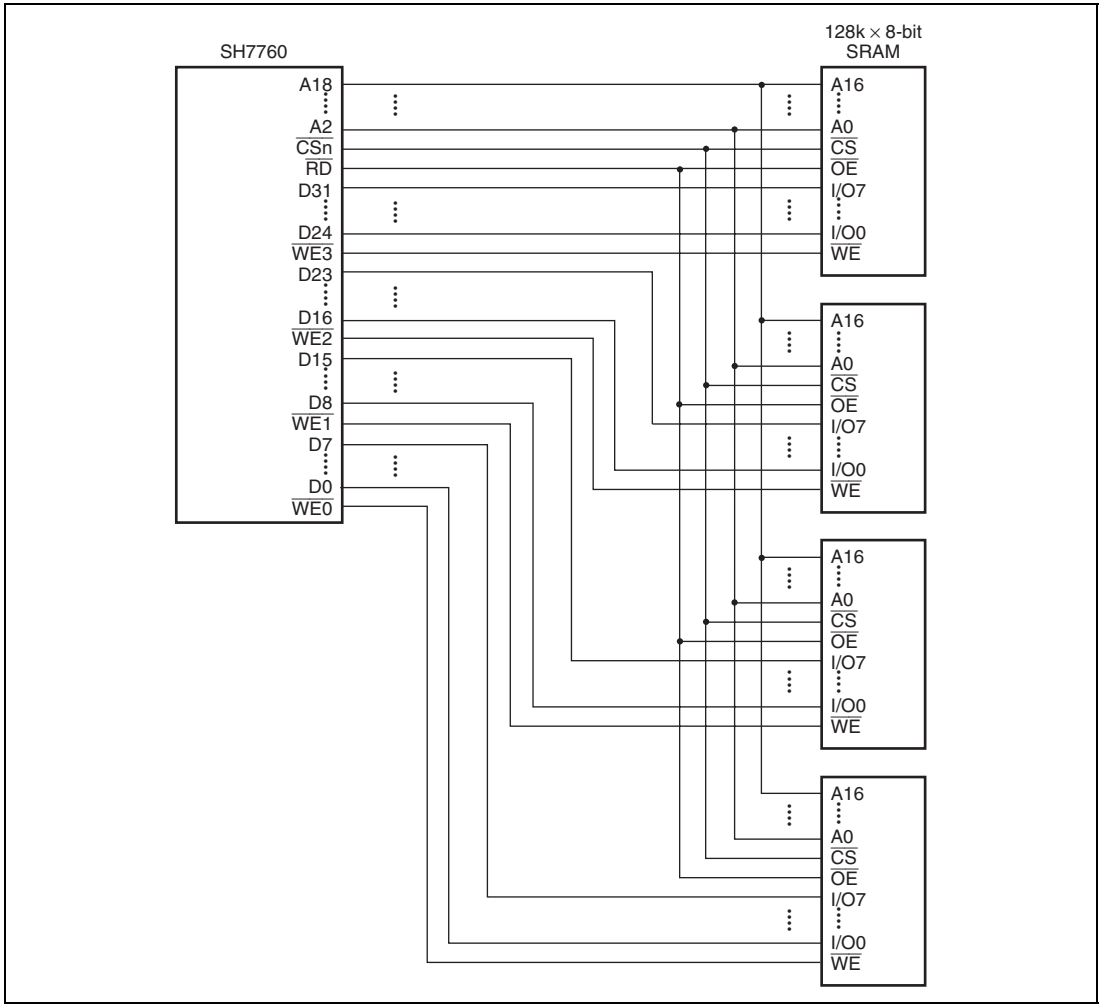


Figure 10.7 Example of 32-Bit Data Width SRAM Connection

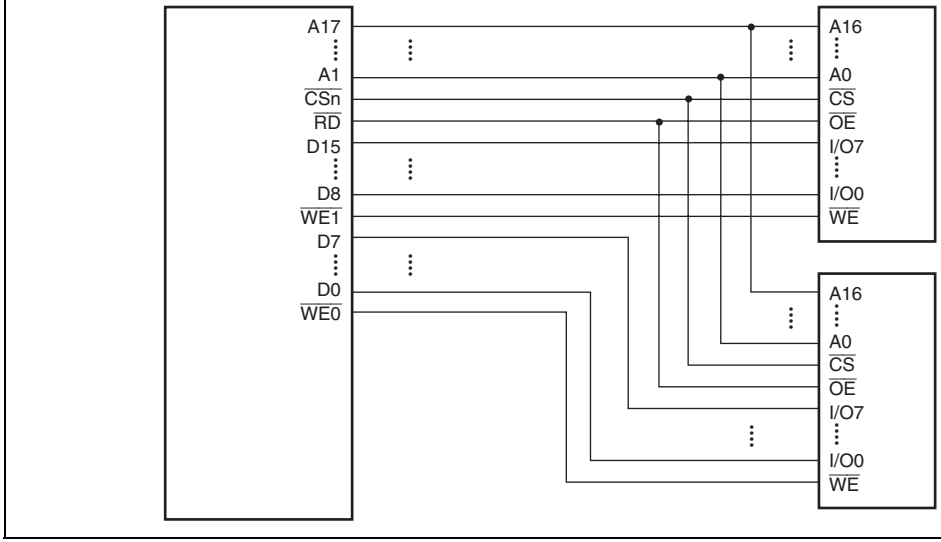


Figure 10.8 Example of 16-Bit Data Width SRAM Connection

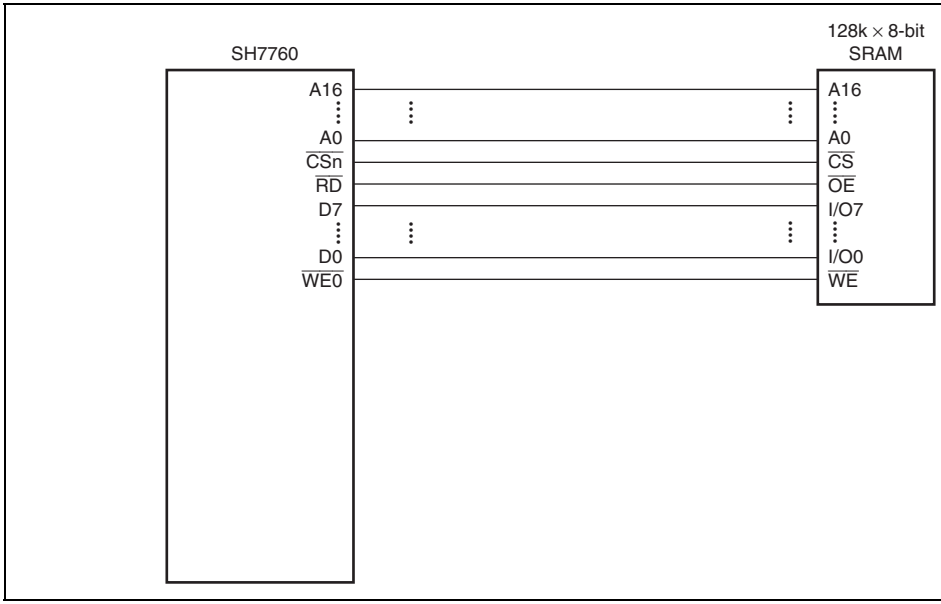
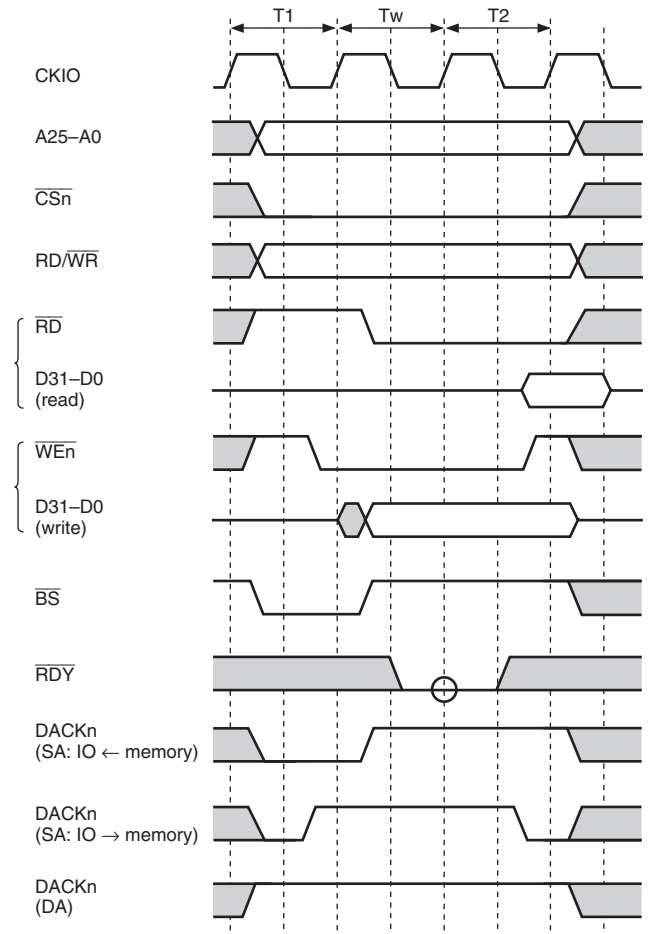


Figure 10.9 Example of 8-Bit Data Width SRAM Connection

Wait-state insertion for the SRAM interface can be controlled by WCR2. If the wait-control bits for each area in WCR2 are not zero, a software wait is inserted in accordance with the wait-control bits. For details, see section 10.5.6, Wait Control Register 2 (WCR2).

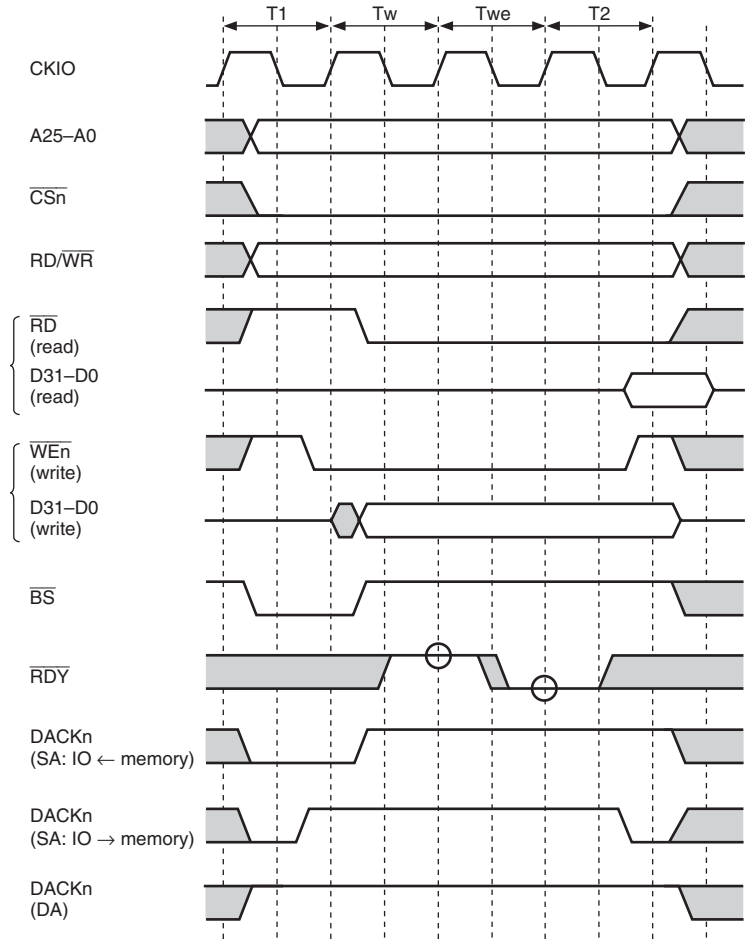
A specified number of Tw cycles are inserted as wait cycles using the SRAM interface wait timing shown in figure 10.10.



Note: For DACKn, an example is shown where the acknowledge level (AL) bit in CHCRn of the DMAC is cleared to 0.

Figure 10.10 SRAM Interface Wait Timing (Software Wait Only)

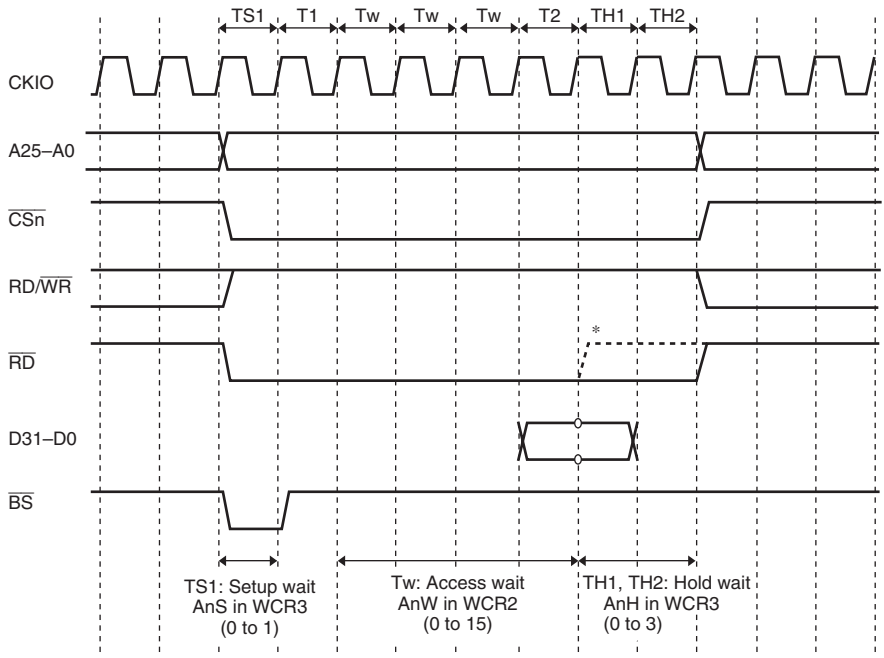
sampled. RDY signal sampling is shown in figure 10.11. A single wait cycle as a software wait is specified in figure 10.11. Sampling is performed at the transition from the Tw state to the T2 state; therefore, the $\overline{\text{RDY}}$ signal has no effect if asserted in the T1 cycle or in the first Tw cycle. The $\overline{\text{RDY}}$ signal is sampled on the rising edge of the clock.



Note: For DACKn, an example is shown where the acknowledge level (AL) bit in CHCRn of the DMAC is cleared to 0.

Figure 10.11 SRAM Interface Wait Timing (Wait Cycle Insertion by $\overline{\text{RDY}}$ Signal)

When the SRAM interface is in use, timing for the negation of the strobe during read operations can be specified by the A1RDH and A4RDH bits in WCR3. When the byte control SRAM interface is in use, AnRDH should be cleared to 0.



Note: * Where the AnRDH bit is set to 1.

Figure 10.12 SRAM Interface Wait State Timing (Read Strobe Negate Timing Setting; AnS = 1, AnW = 011, AnH = 10)

Figure 10.13 shows the SRAM read timing when the division ratio is set to CKIO/2 by DIV[1:0] in DCKDR.

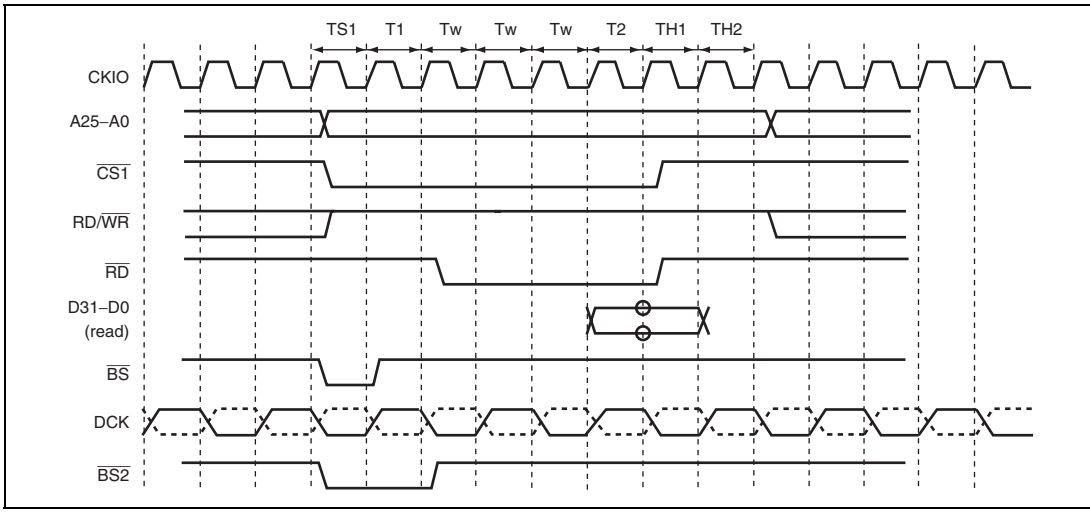


Figure 10.13 DCK, BS2, and CS1 Timing when Reading SRAM Interface (DCKDR = H'0002, A1RDH = 1 and A1H[1:0] = 10 in WCR3, CSH[1:0] in WCR4 = 10, Three Wait Cycles)

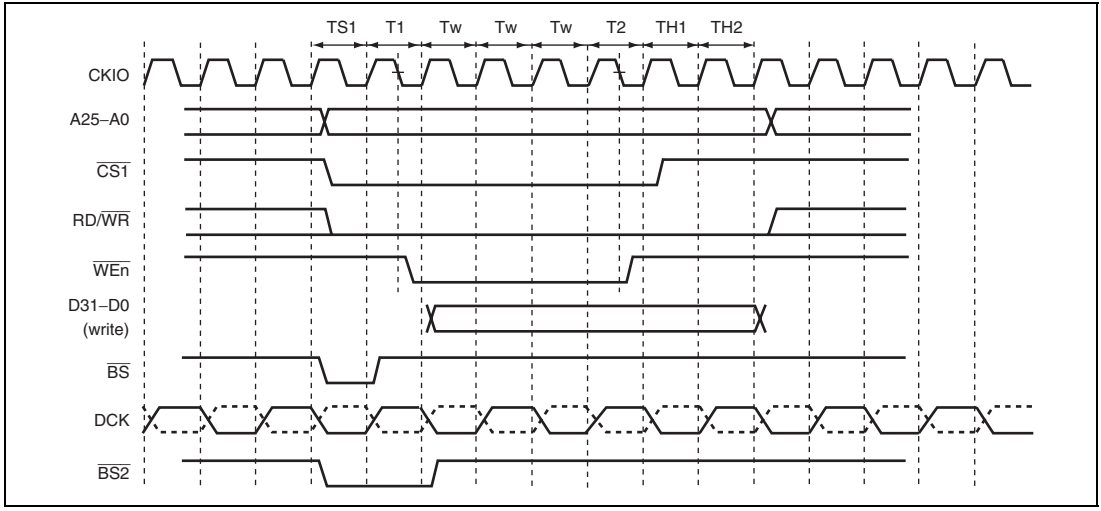


Figure 10.14 DCK, BS2, and CS1 Timing when Writing to SRAM Interface (DCKDR = H'0002, A1RDH = 1 and A1H[1:0] = 10 in WCR3, CSH[1:0] in WCR4 = 10, Three Wait Cycles)

The negate timing for CS1 and RD should be set as follows:

Number of Inserted Wait Cycles during Data Holding	A1H[1:0]		CSH[1:0]	
0	0	0	0	0
1	0	1	0	1
2	1	0	1	0
3	1	1	1	1

10.6.4 Synchronous DRAM Interface

(1) Synchronous DRAM Connection System

Since synchronous DRAM is selectable with the CS signal, it can be connected to off-chip memory space areas 2 and 3 and share usage of RAS and other control signals. If bits DRAMTP2 to DRAMTP0 in BCR1 are 010, area 3 becomes a synchronous DRAM interface. If set to 011, areas 2 and 3 both become synchronous DRAM interfaces.

This LSI supports burst read and burst write modes with a burst length of 4 as a synchronous DRAM operating mode. The data bus width is 32 bits, and the SZ bits in MCR must be set to 11.

write-through area and read/write operations in the non-cacheable area, 16-byte data is also read in a single read because the synchronous DRAM is accessed by burst read/write operations with a burst length of 4. Transfer of 16-byte data is also performed in a single write, but DQMn is not asserted when unnecessary data is transferred.

This LSI also supports read and burst read and burst write modes with a burst length of 8 as a synchronous DRAM operating mode. The data bus width is 32 bits, and the SZ bits in MCR must be set to 11. A 32-byte burst transfer is performed in a cache fill/copy-back cycle. For write operations in the write-through area and read/write operations in the non-cacheable area, 32-byte data is also read in a single read because the synchronous DRAM is accessed by burst read/write operations with a burst length of 8. Transfer of 32-byte data is also performed in a single write, but DQMn is not asserted when unnecessary data is transferred. For details of setting a burst length of 8, refer to (11) Changing the Burst Length, in section 10.6.4. For details of burst length, refer to section 10.5.11, Synchronous DRAM Mode Register (SDMR), and (10) Power-on Sequence, in section 10.6.4.

The control signals for connecting synchronous DRAM are $\overline{\text{RAS}}$, $\overline{\text{CASS}}$, RD/$\overline{\text{WR}}$, $\overline{\text{CS2}}$ or $\overline{\text{CS3}}$, DQM0 to DQM3, and CKE. All signals other than $\overline{\text{CS2}}$ and $\overline{\text{CS3}}$ are common to all areas, and signals other than CKE are valid and latched only when $\overline{\text{CS2}}$ or $\overline{\text{CS3}}$ is asserted. Synchronous DRAM can therefore be connected in parallel to multiple areas. CKE is negated (to low level) when the frequency is changed, when the clock is unstable during stopping of the clock or restarting of the clock supply, or when self-refreshing is performed. Otherwise, CKE is always asserted (to high level).

RAS, CASS, RD/WR, and specific address signals specify commands for synchronous DRAM. The commands are NOP, auto-refresh (REF), self-refresh (SELF), precharge all banks (PALL), precharge specified bank (PRE), row address strobe bank active (ACTV), read (READ), read with precharge (READA), write (WRIT), write with precharge (WRITA), and mode register setting (MRS).

Bytes are specified by DQM0 to DQM3. A read/write is performed for the byte where the corresponding DQM signal is low. When the bus width is 32 bits in big-endian mode, DQM3 specifies an access to address 4n and DQM0 specifies an access to address 4n + 3. In little-endian mode, DQM3 specifies an access to address 4n + 3 and DQM0 specifies an access to address 4n.

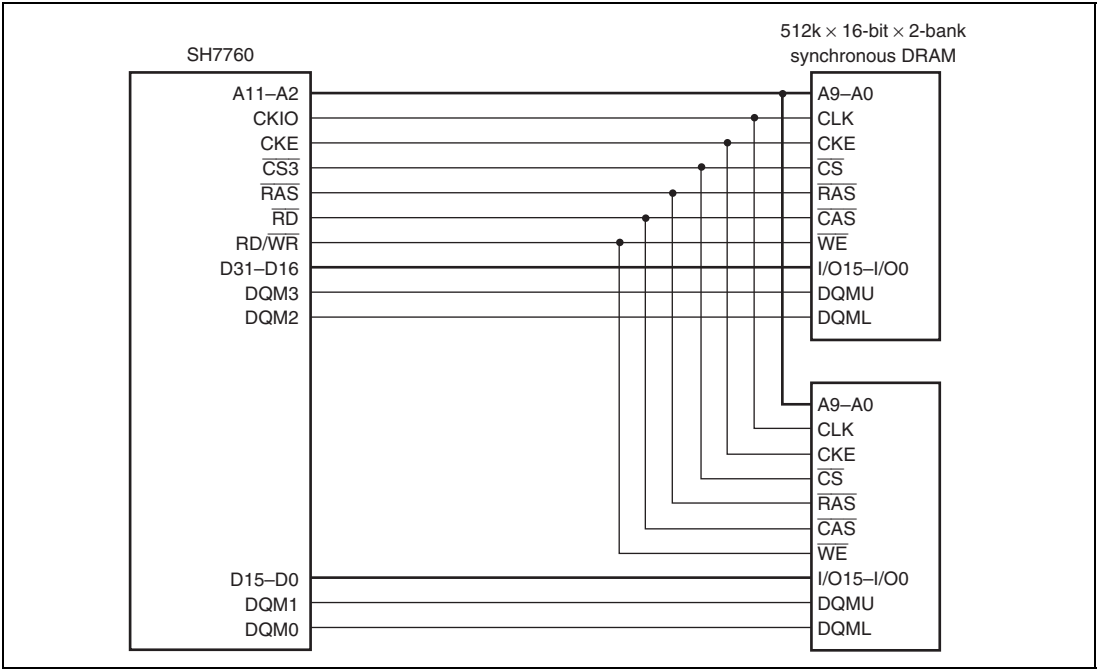


Figure 10.15 Connection Example of Synchronous DRAM with 32-Bit Data Width (Area 3)

(2) Address Multiplexing

Address multiplexing is performed so that synchronous DRAM can be connected without off-chip multiplexing circuitry in accordance with the address multiplexing bits AMXEXT and AMX2 to AMX0 in MCR. Table 10.16 shows the relationship between the address multiplexing bits and the bits output on the address pins. The address signals output on address pins A25 to A18, A1, and A0 are not guaranteed.

A0, which serves as the LSB of the synchronous DRAM address pin, specifies the longword address connected to this LSI. Therefore, be sure to first connect pin A0 of the synchronous DRAM to pin A2 of this LSI, and then connect pin A1 to pin A3.

Address Pins (32-Bit Bus Width, AMX2 to AMX0 = 000, AMXEXT = 0)

Address Pin of SH7760			Synchronous DRAM Address Pin	Function
	RAS Cycle	CAS Cycle		
A13	A21	A21	A11	Select bank address BANK
A12	A20	H/L	A10	Address precharge setting
A11	A19	0	A9	Address
A10	A18	0	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	Not used	Not used	Not used	
A0	Not used	Not used	Not used	

(3) Burst Read

The timing chart for a burst read is shown in figure 10.16. The example below assumes that two 512k × 16-bit × 2-bank synchronous DRAMs are connected, and a 32-bit data width is used. The burst length is 4. An ACTV command is output in the Tr cycle and then a READ command is issued in the Tcl cycle. After 4 cycles, a READA command is issued and the read data is fetched on the rising edge of the off-chip command clock (CKIO) from cycle Td1 to cycle Td8. The Tpc cycle is used to wait for completion of auto-precharge based on the READA command inside the synchronous DRAM, and no new access commands can be issued to the same bank during this waiting period. In this LSI, the number of Tpc cycles is determined based on the bits TPC2 to TPC0 in MCR that are specified, and no commands are issued for the synchronous DRAM during this period.

The example in figure 10.16 shows the basic cycle. To connect slower synchronous DRAM, setting bits WCR2 and MCR can extend the cycle. The bits RCD1 and RCD0 in MCR can be used to specify the number of cycles from the ACTV command output cycle Tr to the READ command output cycle Tc1, with the values of 0 to 3 corresponding to 2 to 4 cycles, respectively. For 2 or more cycles, a Trw cycle, which issues an NOP command for the synchronous DRAM, is inserted

latch cycle Td1 as 1 to 5 cycles independently for areas 2 and 3. This number of cycles corresponds to the number of synchronous DRAM CAS latency cycles.

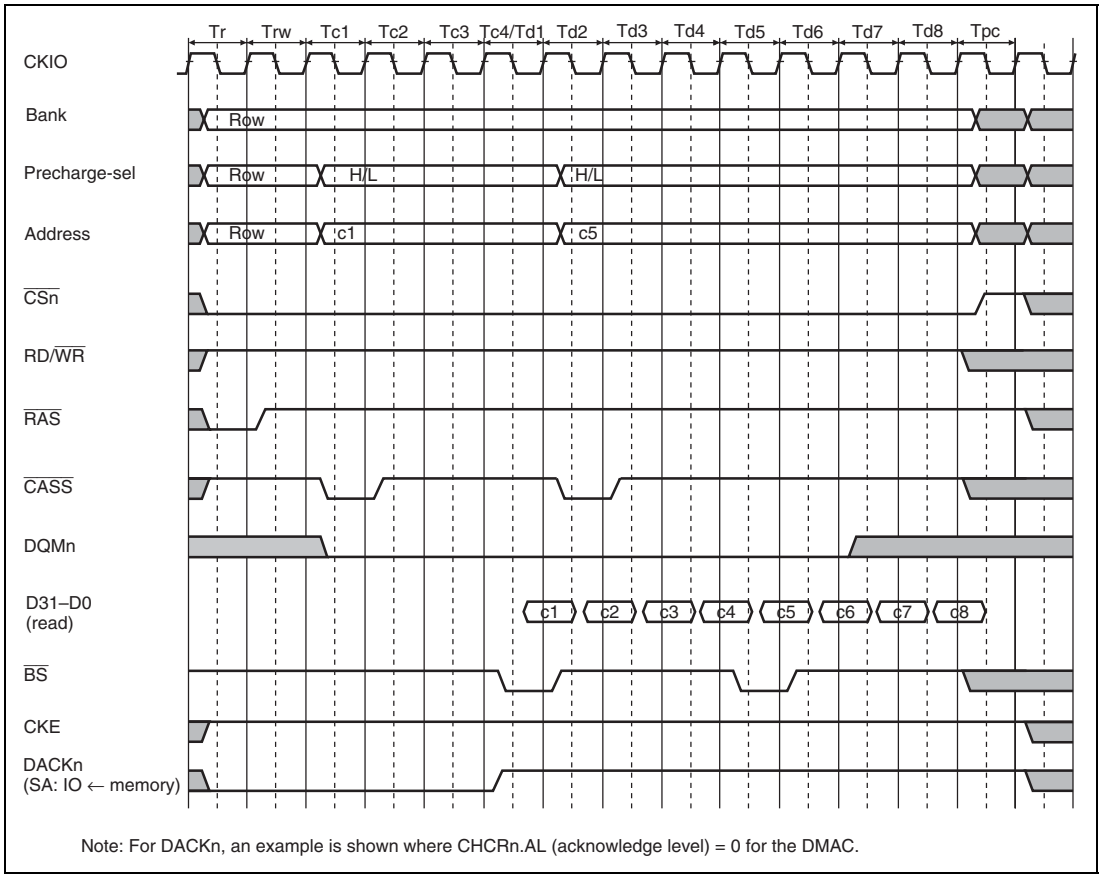


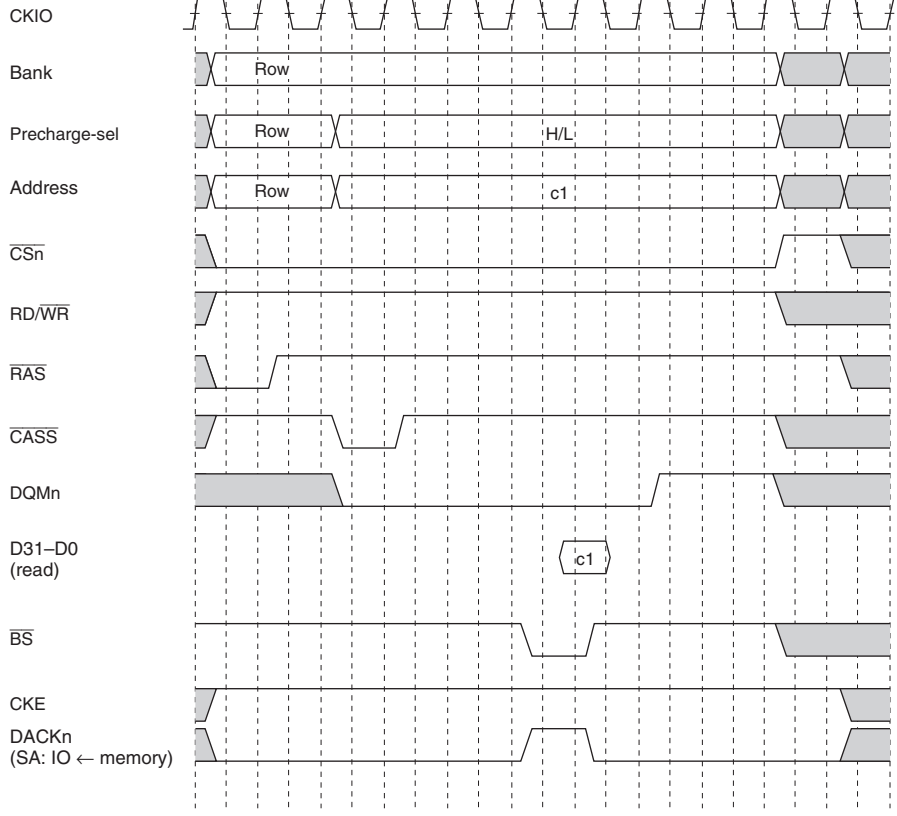
Figure 10.16 Basic Timing for Synchronous DRAM Burst Read

In a synchronous DRAM cycle, the $\overline{BS}$ signal is asserted for one cycle at the beginning of each data transfer cycle corresponding to a READ or READA command. When the data is accessed in the fill operation for a cache miss, the 64-bit boundary data including the missing data are first read by the initial READ command, and then the 16-byte boundary data including the missing data are read in wraparound mode. READA commands that are subsequently issued are used to read the 16 bytes of data, which is the remainder of the 32-byte boundary data.

The timing chart for a single read is shown in figure 10.17. In this LSI, since synchronous DRAM is set to burst read/burst write mode, read data continues to be output even after the required data has been received. To prevent data collisions, after the required data is read in cycle Td1, dummy read cycles Td2 to Td4 are performed, and this LSI waits for the end of the synchronous DRAM operation.

There are 4 burst transfers during a read. In cache-through and other DMA read cycles, the $\overline{BS}$ signal is asserted and data is latched only in cycle Td1 of cycles Td1 to Td4.

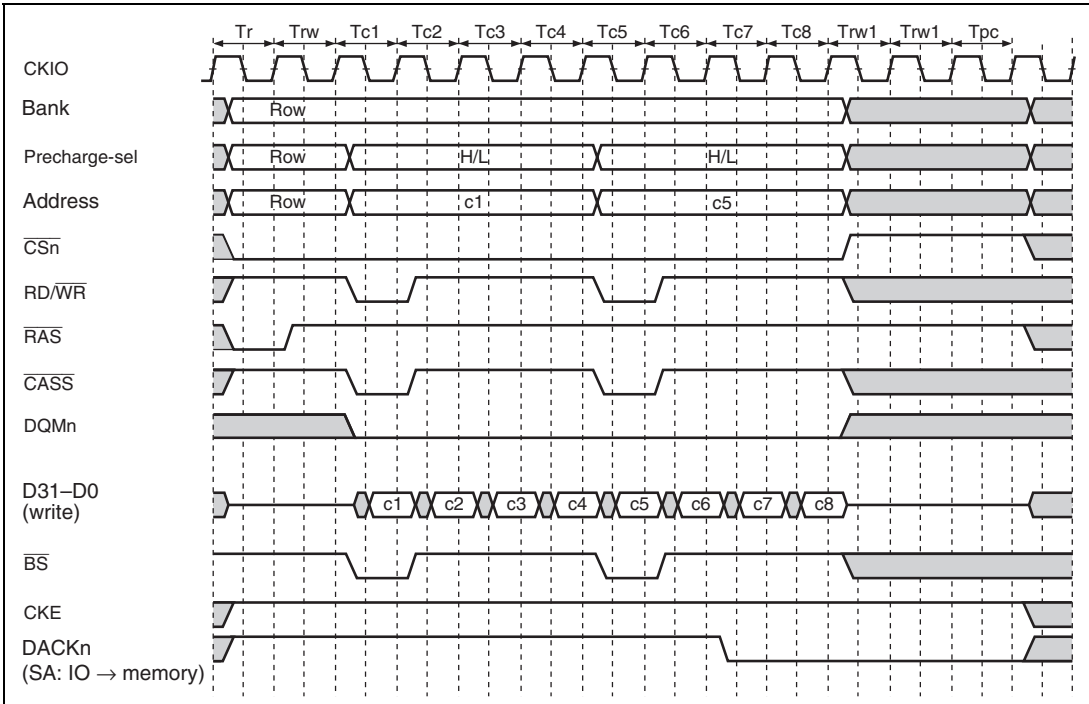
Such dummy cycles increase the memory access time and tend to reduce program execution speed and DMA transfer speed. It is important both to avoid access to unnecessary cache-through areas and to use a data structure that allows data to be placed at a 32-byte boundary for transfer in 32-byte units when carrying out DMA transfer with synchronous DRAM specified as the source.



Note: For DACKn, an example is shown where CHCRn.AL (acknowledge level) = 0 for the DMAC.

Figure 10.17 Basic Timing for Synchronous DRAM Single Read

The timing chart for a burst write is shown in figure 10.18. In this LSI, a burst write occurs only in the event of 32-byte transfer. In a burst write operation, the WRIT command is issued in cycle Tc1 following the Tr cycle where the ACTV command is output, and then 4 cycles later, the WRITA command is issued. In the write cycle, the write data is output at the same time as the write command. For 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 cycle until precharging is started following the write command for delaying issuance of a new command for the synchronous DRAM during this period. Bits TRWL2 to TRWL0 in MCR can be used to specify the number of Trwl cycles. Access is started from 16-byte boundary data, and 32-byte boundary data is written in wraparound mode. DACK is asserted two cycles before the data write cycle.

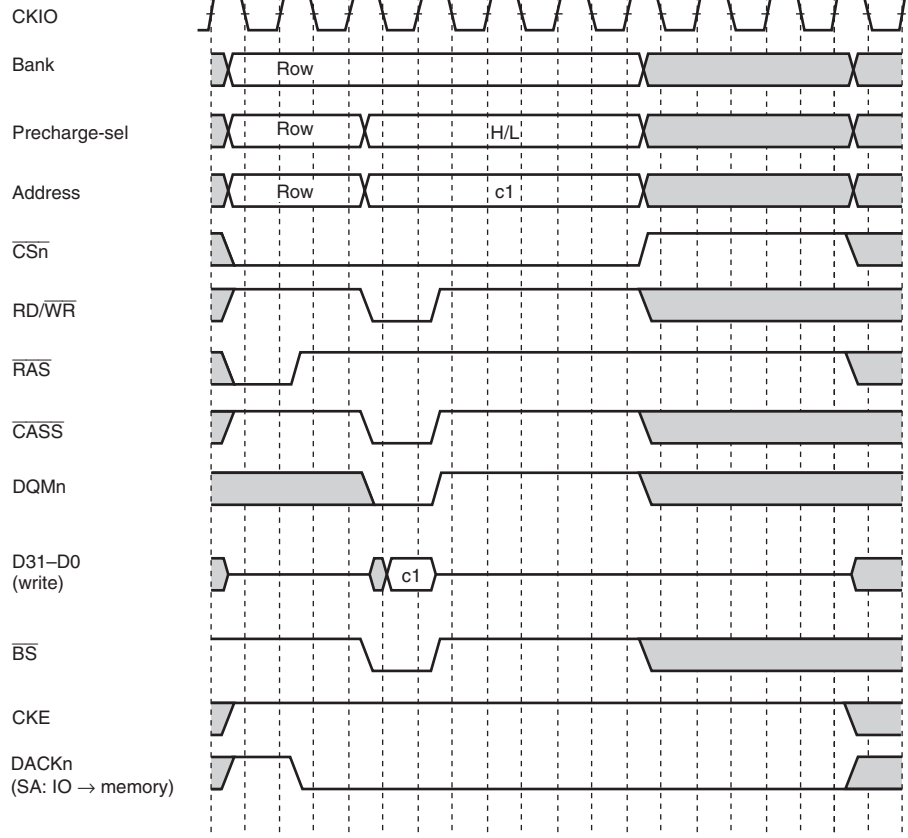


Note: For DACKn, an example is shown where CHCRn.AL (acknowledge level) = 0 for the DMAC.

Figure 10.18 Basic Timing for Synchronous DRAM Burst Write

The basic timing chart for single write access is shown in figure 10.19. In a single write operation, a WRITA command that performs auto-precharge is issued in cycle Tc1 following the Tr cycle where the ACTV command is output. In the write cycle, the write data is output at the same time as the write command. For 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 synchronous DRAM 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 cycle until precharging is started following the write command for delaying issuance of a new command for the synchronous DRAM during this period. Bits TRWL2 to TRWL0 in MCR can be used to specify the number of Trwl cycles. DACK is asserted two cycles before the data write cycle.

This LSI supports 4- or 8-burst-length read and write operations of synchronous DRAM. Dummy cycles are therefore generated even with single write operations.



Note: For DACKn, an example is shown where CHCRn.AL (acknowledge level) = 0 for the DMAC.

Figure 10.19 Basic Timing for Synchronous DRAM Single Write

The synchronous DRAM bank function is used to support high-speed accesses to the same row address. When the RASD bit in MCR is 1, the read/write commands perform access using commands without auto-precharge (READ, WRIT). In this case, precharging is not performed when the access ends. When accessing the same row address in the same bank, it is possible to issue the READ or WRIT command immediately without issuing an ACTV command in the same way as in the DRAM RAS down state. Since the synchronous DRAM is internally divided into two or four banks, one row address in each bank can be activated. If the next access is to a different row address, a PRE command is first issued to precharge the relevant bank, and then when precharging is completed, the access is performed by issuing an ACTV command followed by a READ or WRIT command. If this is followed by an access to a different row address, the access time will be longer because of the precharging performed after the access request is issued.

In a write, when auto-precharge is performed, a command cannot be issued for a period of Trwl + Tpc cycles after issuance of the WRITA command. When RAS down mode is used, READ or WRIT commands can be issued successively if the row address is the same. The number of cycles can thus be reduced by Trwl + Tpc cycles for each write. Bits TPC2 to TPC0 in MCR are used to determine the number of cycles between issuance of the PRE command and the ACTV command.

There is a limit on the time t_{RAS} that each bank can be kept in the active state. If execution of a program cannot guarantee that this time value can be observed so that an access to a different row address occurs by a cache miss, auto-refresh must be set and a refresh cycle must be used that is no more than the maximum value of t_{RAS}. This makes it possible to observe the restrictions on the maximum active state time for each bank. If auto-refresh is not used, measures must be taken in the program to ensure that the banks do not remain active for longer than the prescribed time.

A burst read cycle without auto-precharge is shown in figure 10.20, a burst read cycle for the same row address in figure 10.21, and a burst read cycle for different row addresses in figure 10.22. Similarly, a burst write cycle without auto-precharge is shown in figure 10.23, a burst write cycle for the same row address in figure 10.24, and a burst write cycle for different row addresses in figure 10.25.

When synchronous DRAM is read, there is a 2-cycle latency for the DQMn signal that specifies the bytes. As a result, when the READ command is issued in figure 10.20, if the Tc cycle is executed immediately, the DQMn signal is not specified for the cycle Td1 data output. Therefore, the CAS latency should not be set to 1.

When RAS down mode is set, and if only accesses to the respective banks in area 3 are considered, as long as accesses to the same row address continue, the operation starts with the cycle in figure 10.20 or 10.23, and repeats the cycle in figure 10.21 or 10.24. An access to a different area during this time has no effect. If there is an access to a different row address in the

that in figure 10.21 or 10.24. In RAS down mode, a PALL command is also issued before a refresh cycle or before bus release due to bus arbitration.

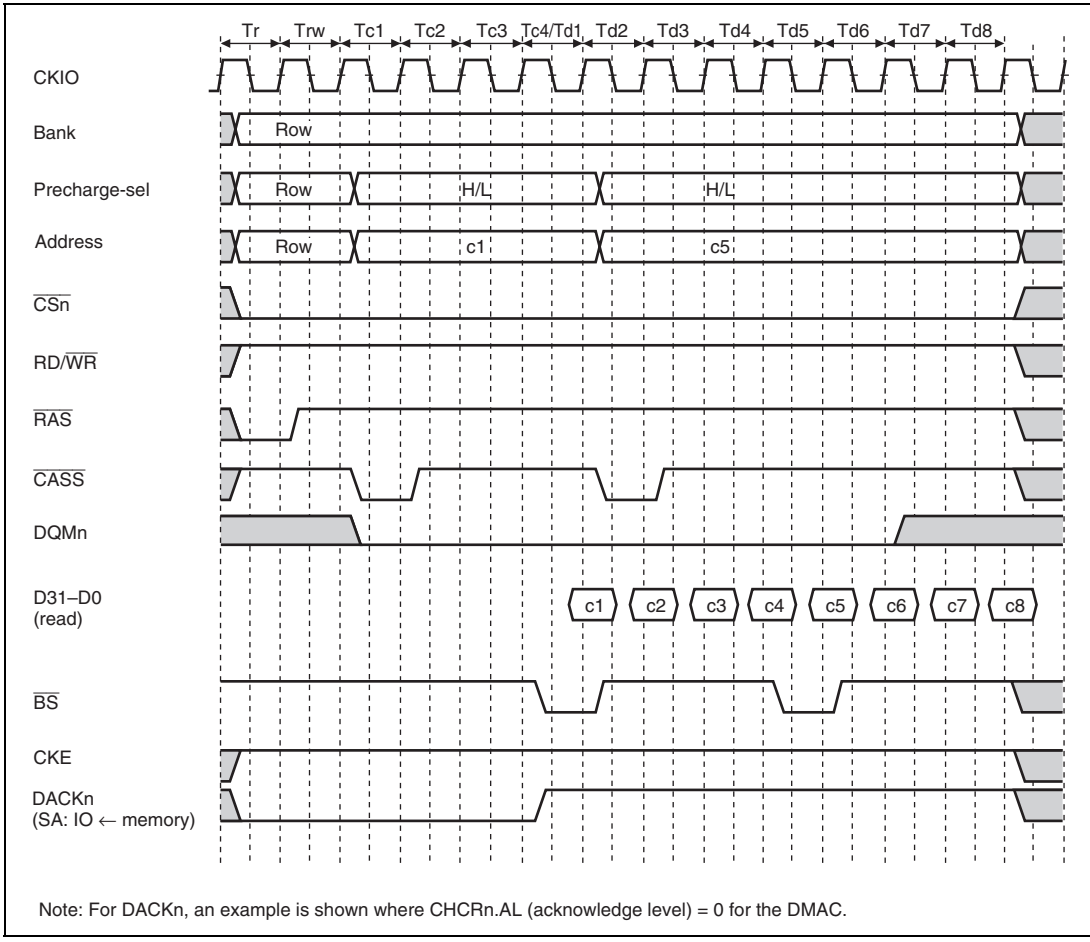
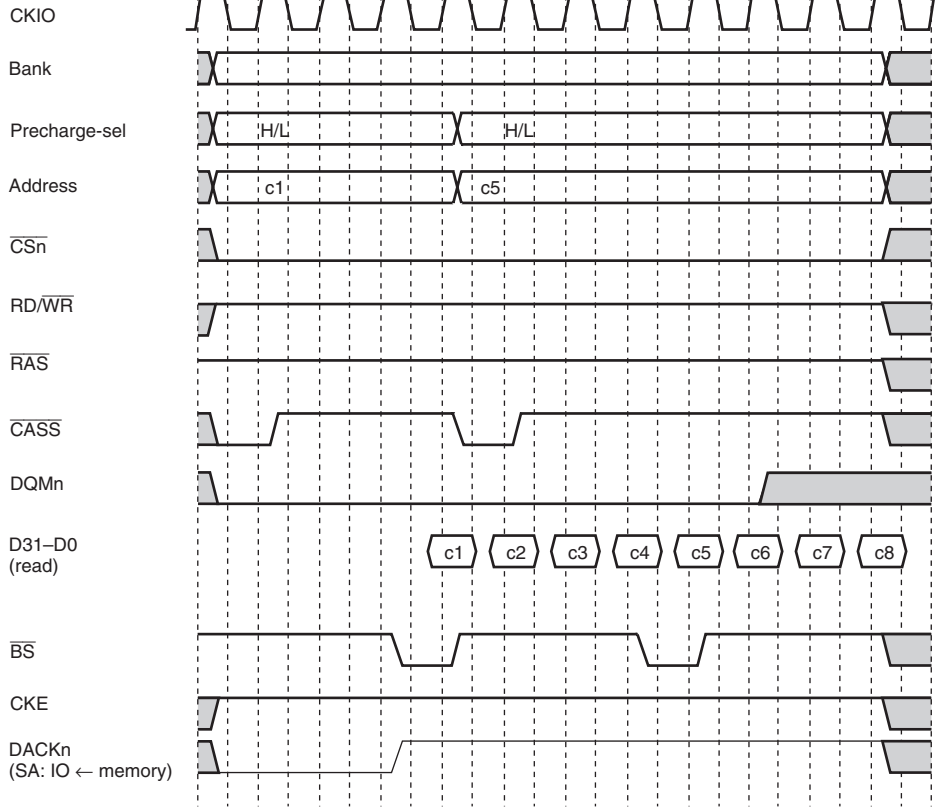
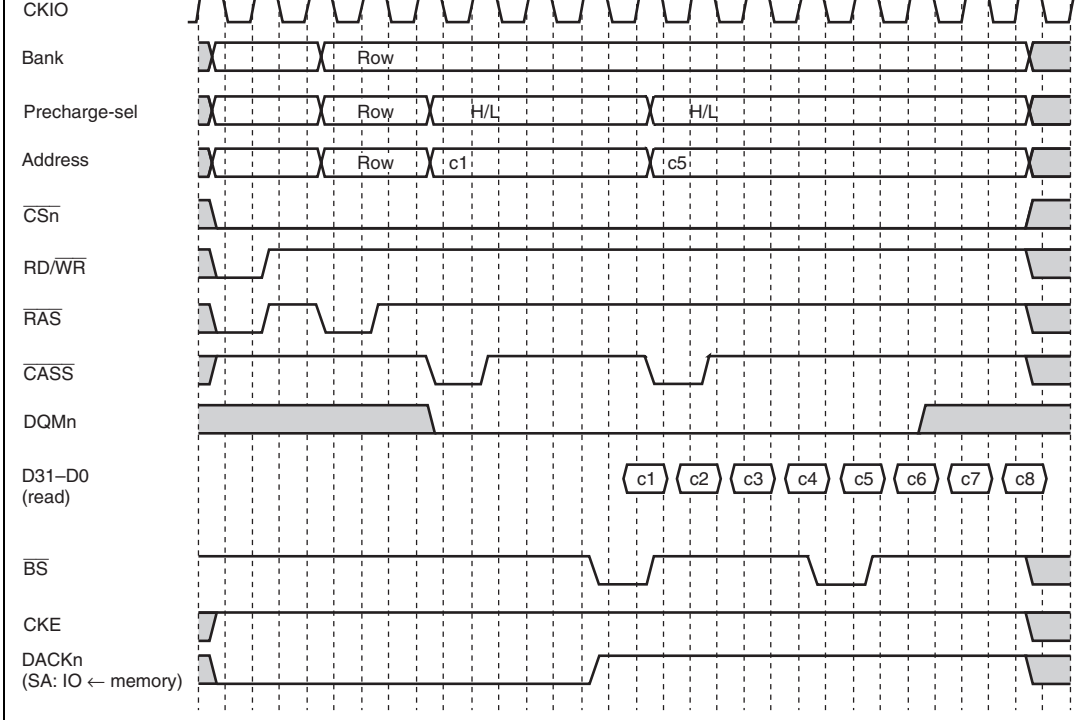


Figure 10.20 Burst Read Timing



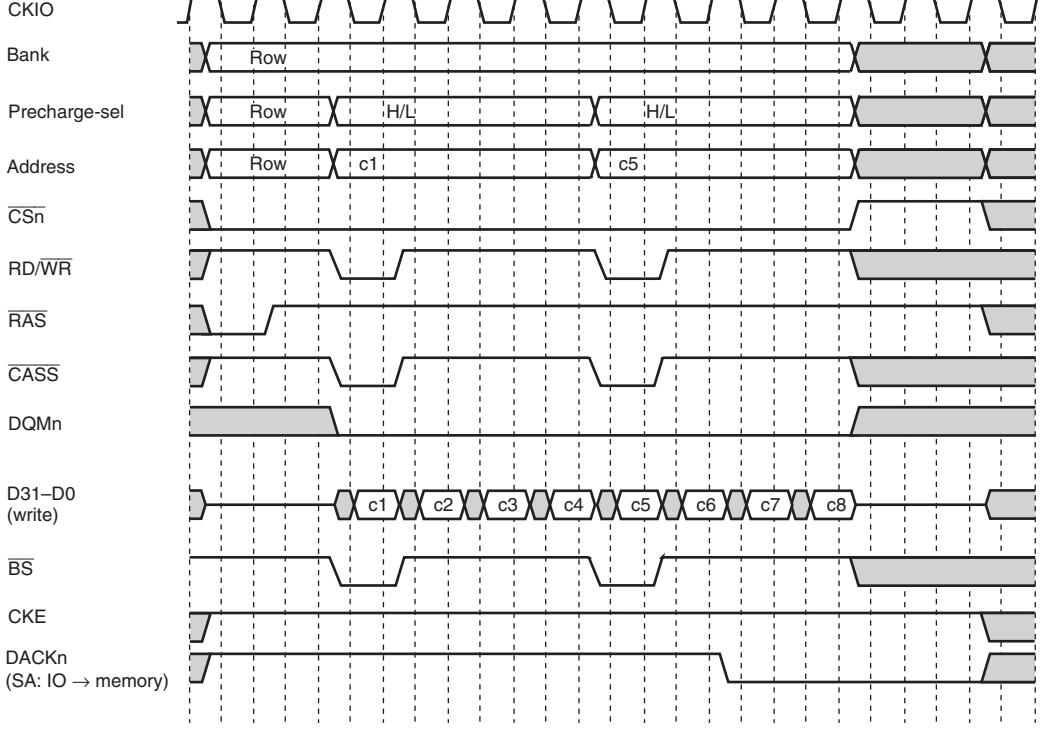
Note: For DACKn, an example is shown where CHCRn.AL (acknowledge level) = 0 for the DMAC.

Figure 10.21 Burst Read Timing (RAS Down, Same Row Address)



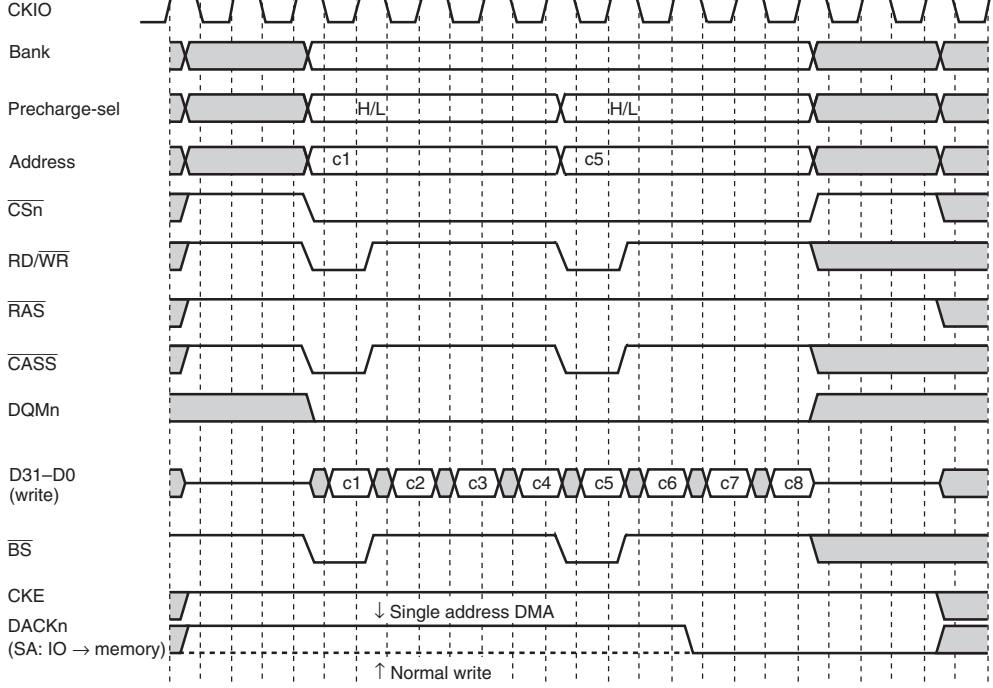
Note: For DACKn, an example is shown where CHCRn.AL (acknowledge level) = 0 for the DMAC.

Figure 10.22 Burst Read Timing (RAS Down, Different Row Addresses)



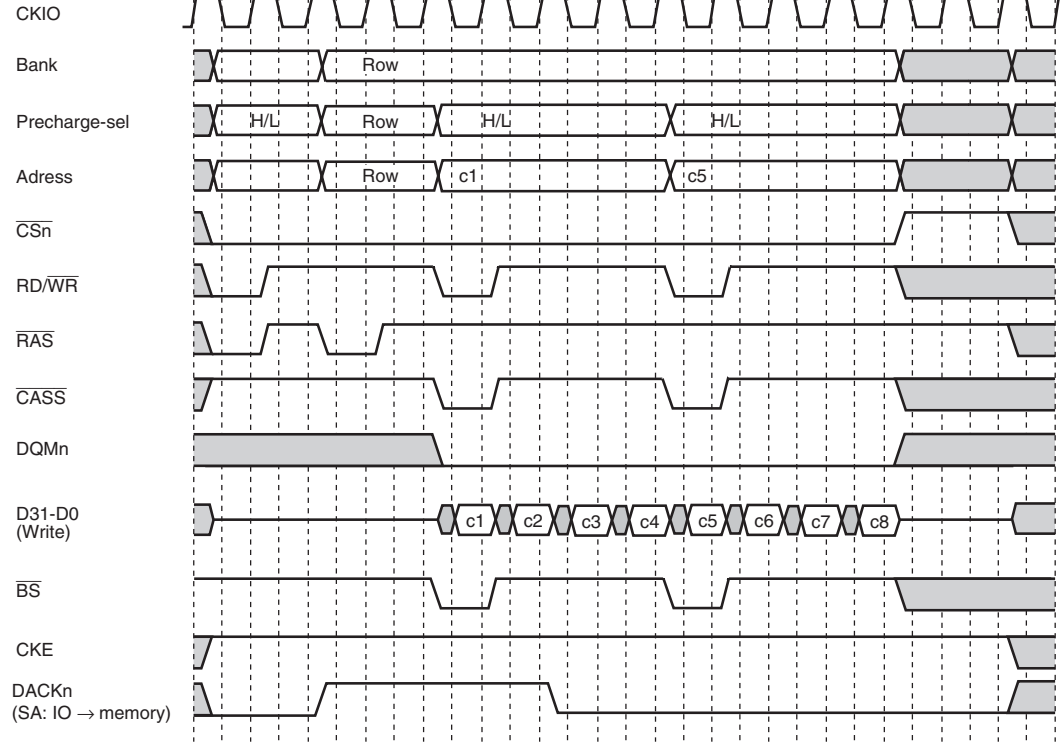
Note: For DACKn, an example is shown where CHCRn.AL (acknowledge level) = 0 for the DMAC.

Figure 10.23 Burst Write Timing



Note: The (Tnop) cycle is inserted only for SA-DMA. The DACKn signal is output as indicated by the solid line.
In the case of a normal write, the (Tnop) cycle is deleted and the DACKn signal is output as indicated by the dotted line.
For DACKn, an example is shown where CHCRn.AL (acknowledge level) = 0 for the DMAC.

Figure 10.24 Burst Write Timing (Same Row Address)



Note: For DACKn, an example is shown where CHCRn.AL (acknowledge level) = 0 for the DMAC.

Figure 10.25 Burst Write Timing (Different Row Addresses)

(8) Pipelined Access

When the RASD bit in MCR is set to 1, pipelined access is performed for faster access to synchronous DRAM between an access by the CPU and an access by the DMAC or for consecutive accesses by the DMAC. Since synchronous DRAM is internally divided into two or four banks, after a READ or WRIT command is issued for one bank it is possible to issue a PRE, ACTV, or other command during the CAS latency cycle, data latch cycle, or data write cycle for shortening the access cycle.

When a read access is followed by another read access to the same row address, after a READ command has been issued, another READ command is issued before the end of the data latch cycle so that read data is on the data bus continuously. When an access is made to another row address and a different bank, the PRE command or ACTV command can be issued during the CAS latency cycle or data latch cycle. If there are consecutive access requests for different row

depending on the bank and row address, but since the write data is output at the same time as the WRIT command, the PRE, ACTV, and WRIT commands are issued so that one or two dummy cycles occur automatically on the data bus. Similarly, when a read access follows a write access, or a write access follows a write access, the PRE, ACTV, or READ command is issued during the data write cycle for the preceding access. However, a PRE command cannot be issued for different row addresses in the same bank, and so the PRE command is issued following the number of Trwl cycles specified by the TRWL bit in MCR after the end of the last data write cycle.

Figure 10.26 shows a burst read cycle for different banks and row addresses from the preceding burst read cycle.

Pipelined access is enabled only for consecutive access to area 3, and is aborted when there is an access to another area. Pipelined access is also aborted in the event of a refresh cycle, or bus release due to bus arbitration. The cases where pipelined access is available are shown in table 10.17. In this table, DMAC dual indicates transfer by DMAC dual address mode, and DMAC single indicates transfer by DMAC single address mode.

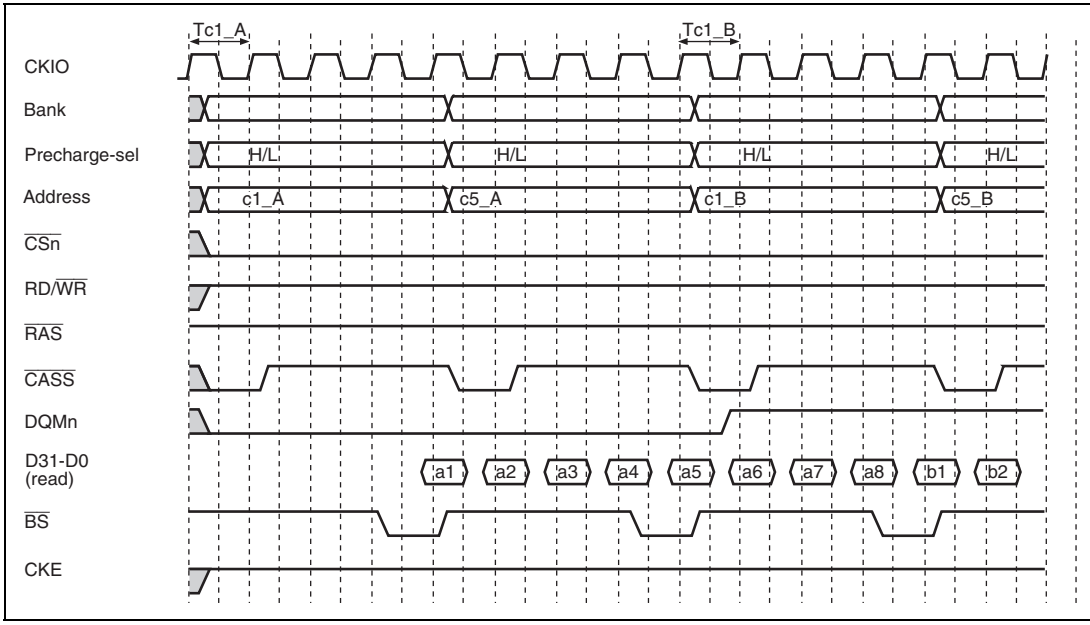


Figure 10.26 Burst Read Cycle for Different Bank and Row Address From Preceding Burst Read Cycle

		Next Access					
		CPU		DMAC Dual		DMAC Single	
Previous Access		Read	Write	Read	Write	Read	Write
CPU	Read	X	X	O	X	O	O
	Write	X	X	O	X	O	O
DMAC dual	Read	X	X	X	X	X	X
	Write	O	O	O	X	O	O
DMAC single	Read	O	O	O	X	O	O
	Write	O	O	O	X	O	O

O: Pipelined access available

X: Pipelined access not available

(9) Refreshing

The bus state controller is provided with a function for controlling refreshing of the synchronous DRAM. 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 of time, both the RMODE bit and the RFSH bit can be set to 1 to activate self-refresh mode, which uses low power consumption for data retention.

(a) 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. Bits CKS2 to CKS0 and RTCOR should be set to satisfy the refresh interval specification for the synchronous DRAM that is used. First make the settings for RTCOR, RTCNT, and the RMODE and RFSH bits in MCR, and then make the CKS2 to CKS0 setting last. 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 10.27 shows the auto-refresh operation and figure 10.29 shows the synchronous DRAM auto-refresh timing.

First, an REF command is issued in the TRr2 cycle. A new command is not output for the duration of (TRr cycles) + (number of cycles specified by bits TRAS2 to TRAS0 in MCR) + (number of cycles specified by bits TRC2 to TRC0 in MCR). Bits TRAS2 to TRAS0 and TRC2 to TRC0 must be set to satisfy the synchronous DRAM refresh cycle time specification (active–active command delay time).

if both areas 2 and 3 are set to the synchronous DRAM, auto-refreshing of area 2 is performed after area 3.

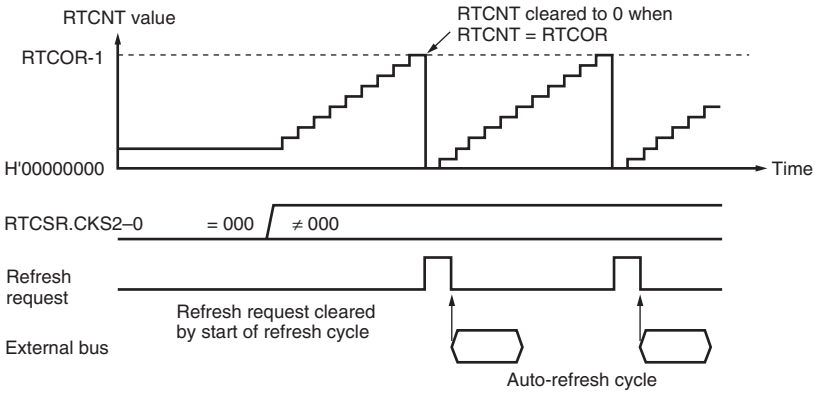


Figure 10.27 Auto-Refresh Operation

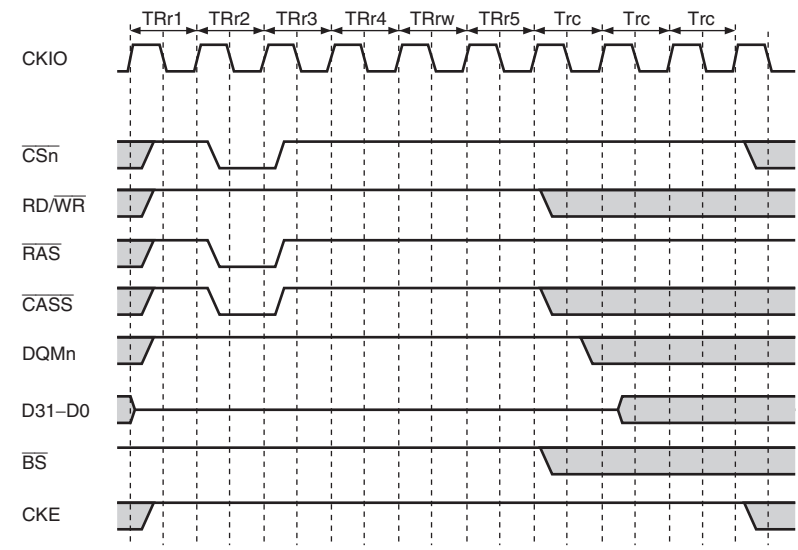


Figure 10.28 Synchronous DRAM Auto-Refresh Timing

Self-refresh mode is a type of software standby mode where 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 canceled by clearing the RMODE bit to 0. After self-refresh mode has been cancelled, the issuing of commands is prohibited for the number of cycles specified by bits TRC2 to TRC0 in MCR. Self-refresh timing is shown in figure 10.29. Settings must be made so that self-refresh cancellation and data retention are performed correctly and auto-refreshing is performed at the correct intervals. When self-refreshing is activated from a state where auto-refreshing is set or when canceling software standby mode using a method other than a power-on reset, setting the RFSH bit to 1 and clearing the RMODE bit to 0 when self-refresh mode is cleared will restart auto-refreshing. If the transition from canceling self-refresh mode to starting 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 enables 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 software standby function and is maintained even after recovery from software standby mode using a method other than power-on reset.

In the case of a power-on reset, the self-refresh state is cancelled since the bus state controller's registers are initialized.

Self-refreshing is continued in sleep mode, in software standby mode, and at a manual reset.

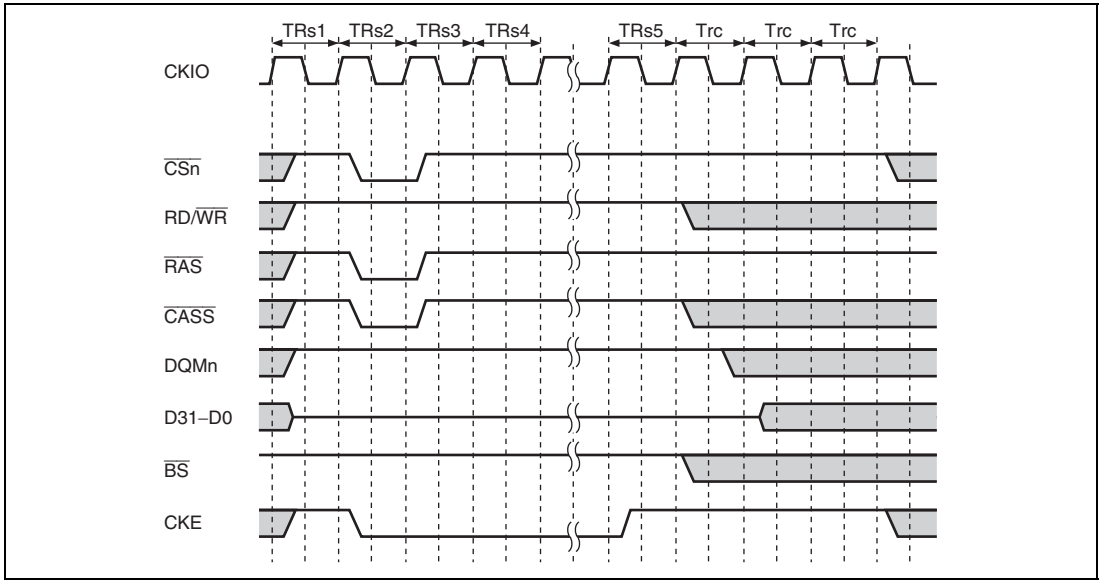


Figure 10.29 Synchronous DRAM Self-Refresh Timing

If a refresh request is generated during execution of a bus cycle, execution of the refresh is deferred until the bus cycle is completed. Refresh operations are deferred during multiple bus cycles generated due to a smaller data bus width than the access size (such as when performing longword access to 8-bit bus width memory) and during a 32-byte transfer such as a cache fill or write-back. Refresh operations are also deferred between read and write cycles during execution of a TAS instruction and between read and write cycles when DMAC dual address transfer is executed. If a refresh request occurs when the bus has been released by the bus arbitration function, refresh execution is deferred until the bus is acquired. If a match between RTCNT and RTCOR occurs while a refresh is waiting to be executed so that a new refresh request is generated, the previous refresh request is eliminated. In order for refreshing to be performed correctly, care must be taken to ensure that no bus cycle or bus mastership occurs that is longer than the refresh interval. When a refresh request is generated, the $\overline{\text{BACK}}$ signal is negated (driven high). Therefore, correct refreshing can be performed by monitoring the $\overline{\text{BACK}}$ signal using a bus arbiter or bus master other than this LSI requesting the bus, and returning the bus to this LSI.

(10) Power-On Sequence

To use synchronous DRAM, the mode must first be set after power is supplied. To initialize synchronous DRAM correctly, the bus state controller registers must first be set, and then writing must be performed to the synchronous DRAM mode register. In the synchronous DRAM mode register setting, the address signal value at that time is latched by a combination of the $\overline{\text{RAS}}$, $\overline{\text{CAS}}$, and RD/$\overline{\text{WR}}$ signals. If the value to be set is X, the bus state controller operates so that the value X is written to the synchronous DRAM mode register by performing a write to address H'FF90 0000 + X for area 2 synchronous DRAM, and to address H'FF94 0000 + 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/burst write, CAS latency 1 to 3, wrap type = sequential, and burst length 4 or 8, which are supported by this LSI, arbitrary data is written in byte-size access to the following addresses.

Bus Width	Burst Length	CAS Latency	Area 2	Area 3
32	4	1	H'FF90 0048	H'FF94 0048
		2	H'FF90 0088	H'FF94 0088
		3	H'FF90 00C8	H'FF94 00C8
32	8	1	H'FF90 004C	H'FF94 004C
		2	H'FF90 008C	H'FF94 008C
		3	H'FF90 00CC	H'FF94 00CC

setting command is issued. The timing for the precharge all banks command is shown in figure 10.30(1), and the timing for the mode register setting command is shown in figure 10.30(2).

Before setting the mode register, a 200-μs idle time (this is required for the synchronous DRAM and varies depending on the memory manufacturer) after power is supplied must be guaranteed. There is no problem in making the precharge all banks setting immediately if the reset signal pulse width is greater than this idle time.

First, a precharge all banks (PALL) command is issued in the TRp1 cycle by writing to address H'FF90 0000 + X or H'FF94 0000 + X while the MRSET bit in MCR is cleared to 0. Next, the number of dummy auto-refresh cycles specified by the manufacturer (usually 8) or more must be executed. This is achieved automatically while performing various kinds of initialization after the auto-refresh setting is made, but a more accurate way is to change the RTCOR value only while these dummy cycles are being executed to set a short interval that generates refresh requests. 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. After auto-refreshing has been executed at least the prescribed number of times, a mode register write command is issued in the TMw1 cycle by setting MCR.MRSET to 1 and writing to address H'FF90 0000 + X or H'FF94 0000 + X.

The synchronous DRAM mode register should be configured only once after power-on reset and before synchronous DRAM access, and the setting should not be changed once it is made.

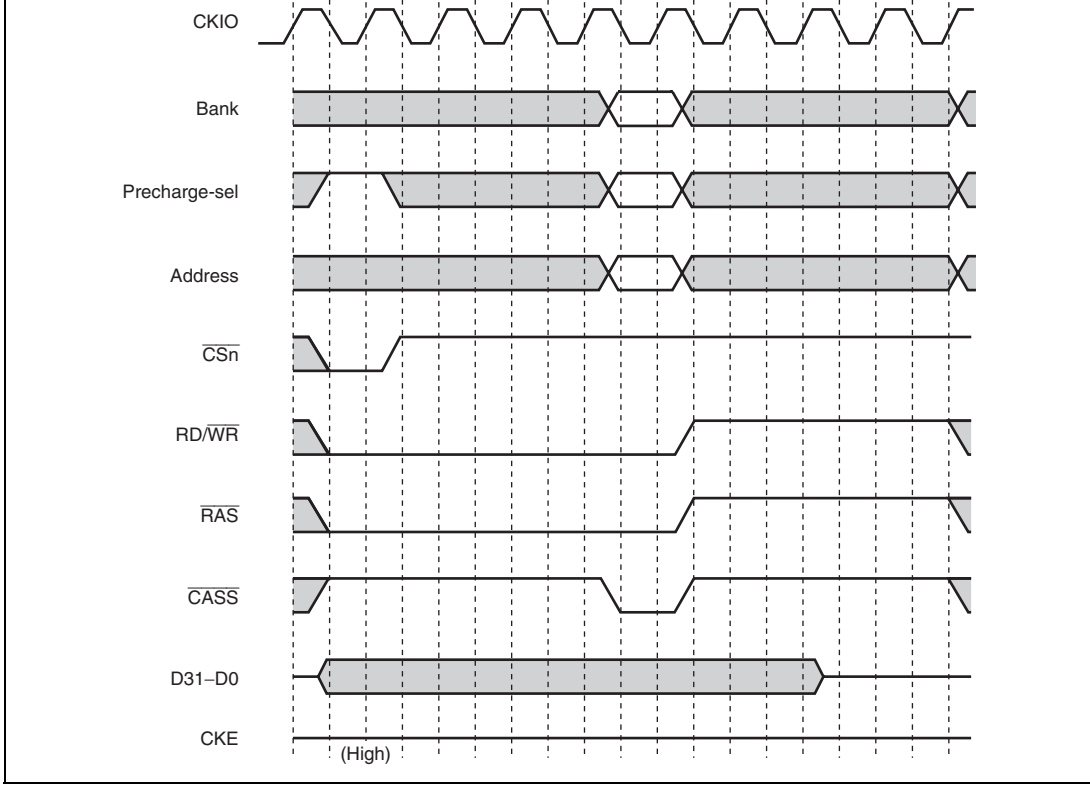


Figure 10.30 (1) Synchronous DRAM Mode Write Timing (PALL)

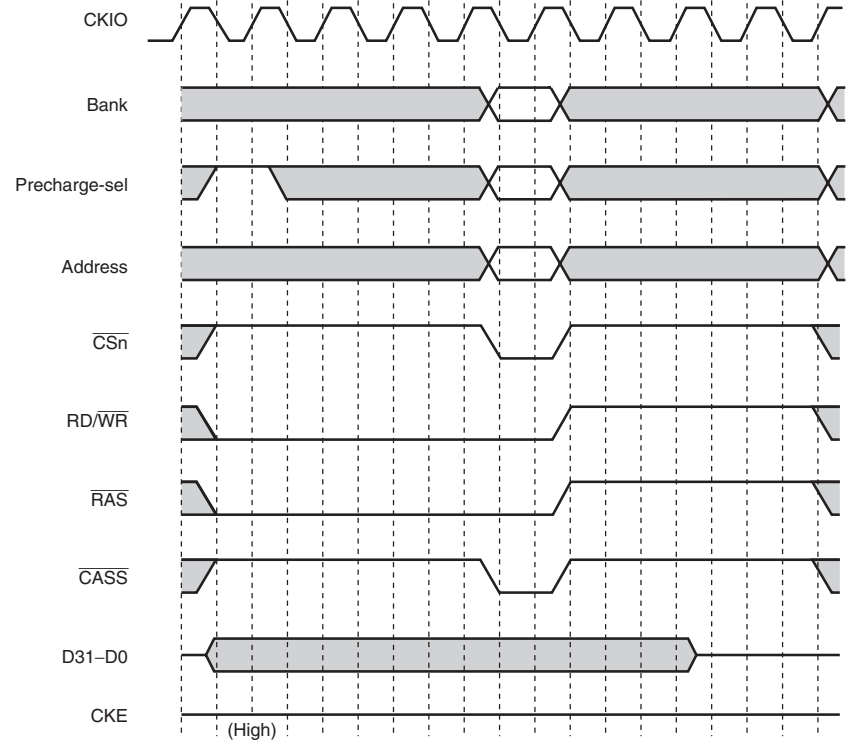


Figure 10.30 (2) Synchronous DRAM Mode Write Timing (Mode Register Setting)

(11) Changing the Burst Length

When synchronous DRAM is connected to this LSI with a 32-bit memory bus width, a burst length of either 4 or 8 is specifiable with the SDBL bit in BCR3. For more details, see the description of the BCR3 register.

(a) Burst Read

Figure 10.31 is the timing chart for burst-read operations. In the example shown below, two synchronous DRAMs of 512k × 16 bits × 2 banks are assumed to be connected and used with a 32-bit data width and a burst length of 8. After the Tr cycle which outputs an ACTV command, a READA command is issued in cycle Tc1. During cycles Td1 to Td8, the read data are fetched at the rising edges of the off-chip command clock (CKIO). Tpc is the cycle used to wait for completion of auto-precharging, which is triggered by the READA command, in the synchronous DRAM. During this cycle, no new command that accesses the same bank can be issued. In this LSI, bits TPC2 to TPC0 in MCR are used to determine the number of Tpc cycles, and no commands are issued for the synchronous DRAM during these cycles.

Bits RCD1 and RCD0 in MCR can be used to specify the number of cycles from the ACTV command output cycle Tr to the READA command output cycle Tc1, where setting values of 1, 2, or 3 correspond to 2, 3, or 4 cycles, respectively. When two or more cycles are specified, the Trw cycle for issuing of NOP commands to the synchronous DRAM is inserted between the Tr and Tc cycles. Bits A2W2 to A2W0 and A3W2 to A3W0 in WCR2 can be used to set the number of cycles from the READA command output cycle Tc1 to cycle Td1 where the first read data is received. The number of cycles from 1 to 5 is specifiable independently for areas 2 and 3. Note that this number of cycles is equal to the number of CAS latency cycles of the synchronous DRAM.

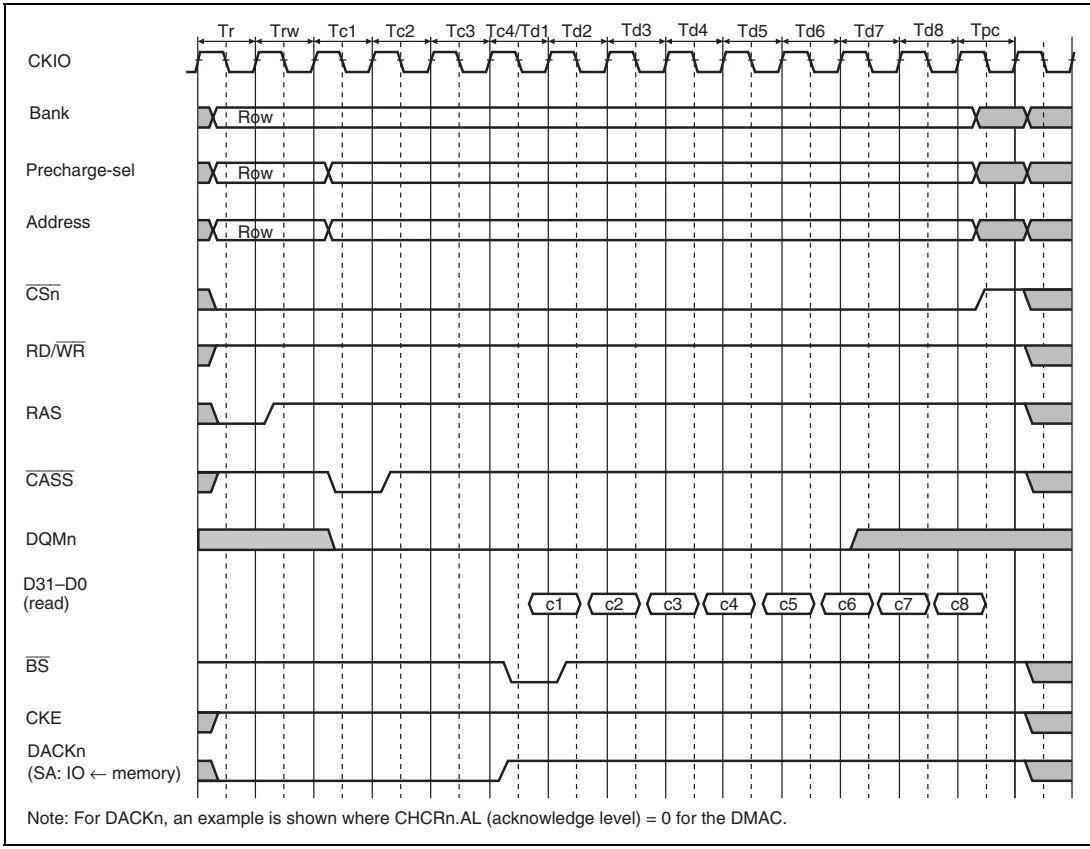
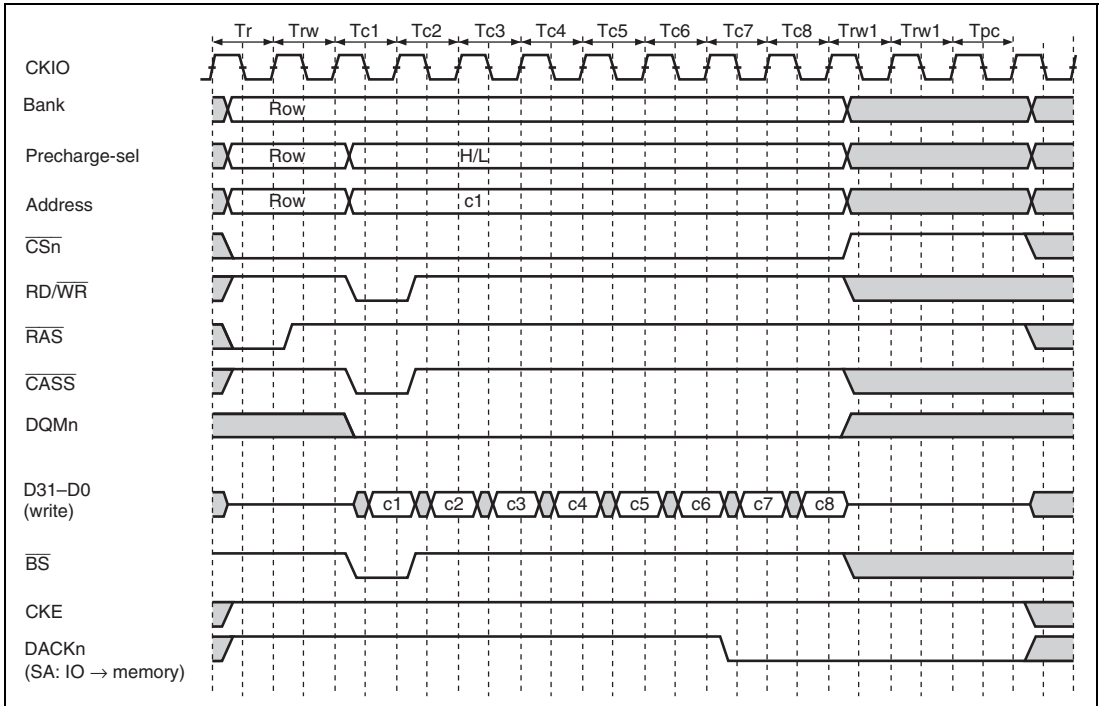


Figure 10.31 Basic Timing of a Burst Read from Synchronous DRAM (Burst Length = 8)

32-bit boundary data including the missing data are first read, and then the 32-byte boundary data including the missing data are read in wraparound mode.

(b) Burst Write

Figure 10.32 is the timing chart for a burst-write operation with a burst length of 8. In this LSI, a burst write takes place when a copy-back of the cache or a 32-byte transfer of data by the DMAC occurs. In a burst-write operation, a WRITA command that performs auto precharging is issued during the Tc1 cycle after the Tr cycle where the ACTV command is output. During the write cycle, the write data is output simultaneously with the write command. For a write command with an auto precharge, since precharging of the relevant bank in the synchronous DRAM is performed after completion of the write command, no new command for the same bank can be issued until precharging has been completed. As a result, besides the precharge waiting cycle Tpc in read access, Trwl cycles are added to provide waiting time until precharging starts after the write command has been issued, and these Trwl cycles delay the issuing of new commands to the same bank. Bits TRWL2 to TRWL0 in MCR can be used to select the number of Trwl cycles. The 32-byte boundary data is written in wraparound mode.



Note: For DACKn, an example is shown where CHCRn.AL (acknowledge level) = 0 for the DMAC.

Figure 10.32 Basic Timing of a Burst Write to Synchronous DRAM

Setting bits A0BST2 to A0BST0, A5BST2 to A5BST0, and A6BST2 to A6BST0 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 burst access function. The timing for burst access to burst ROM is shown in figure 10.33. No wait cycle is set. Basically, access is performed in the same way as for SRAM interface, but when the first cycle ends, only the address is changed, and then the next access is executed. When ROM having an 8-bit data width is connected, bits A0BST2 to A0BST0, A5BST2 to A5BST0, or A6BST2 to A6BST0 can be used to set the number of consecutive accesses to 4, 8, 16, or 32. When ROM having a 16-bit data width is connected, 4, 8, or 16 accesses can be set in the same way. When ROM having a 32-bit data width is connected, 4 or 8 accesses can be set.

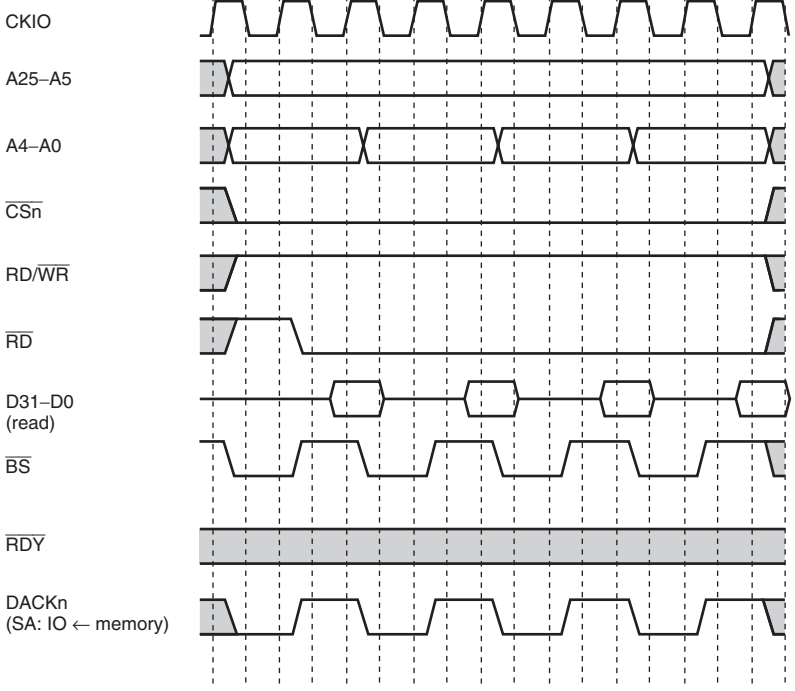
The $\overline{\text{RDY}}$ pin sampling is always performed when one or more wait cycles are set. The timing in this case is shown in figure 10.34.

The second and subsequent access cycles also comprise two cycles when a burst ROM setting is made and the wait is specified at 0.

The write operation for a burst ROM interface is performed as a SRAM interface.

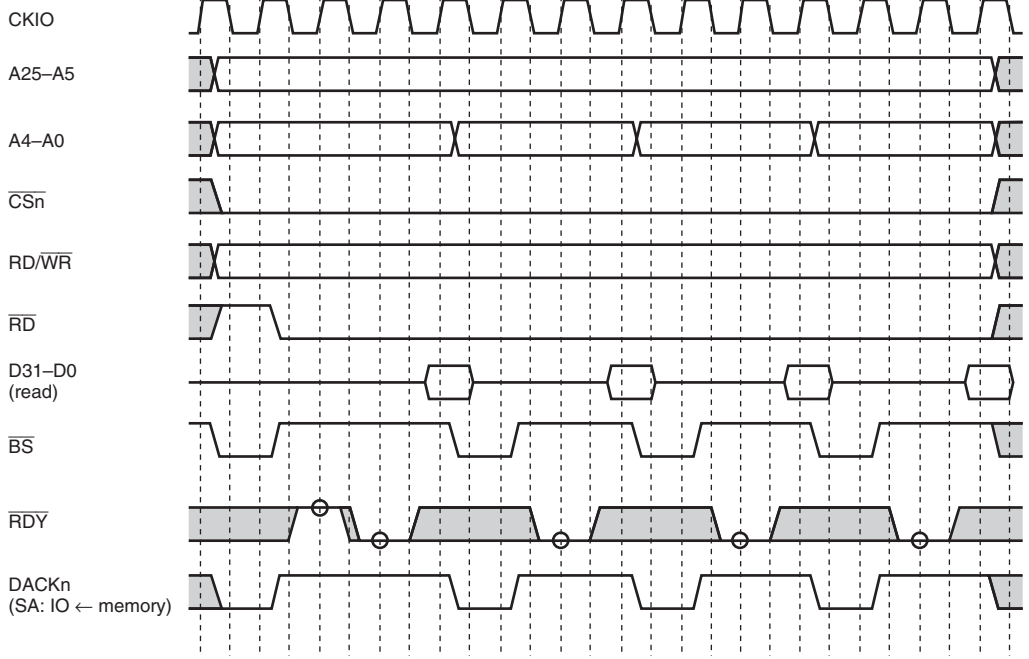
In 32-byte transfer, a total of 32 bytes are transferred consecutively according to the bus width that was set. The first access is performed on the data where there was an access request, and the remaining accesses are performed on the 32-byte boundary data. The bus is not released during this operation.

Figure 10.35 shows the timing when the burst ROM is set and a setup/hold is specified by WCR3.



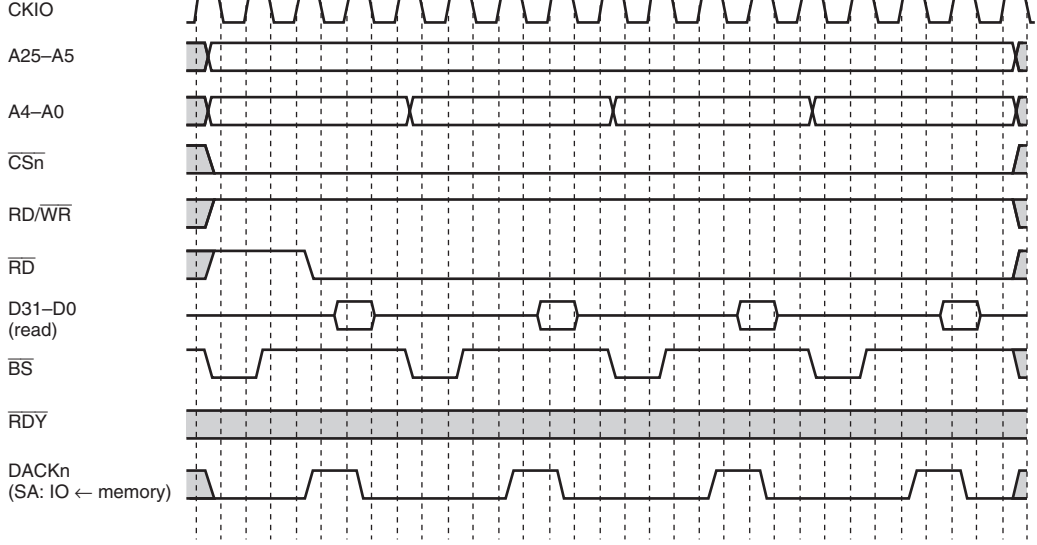
Note: For DACKn, an example is shown where CHCRn.AL (acknowledge level) = 0 for the DMAC.

Figure 10.33 Burst ROM Basic Access Timing



Note: For DACKn, an example is shown where CHCRn.AL (acknowledge level) = 0 for the DMAC.

Figure 10.34 Burst ROM Wait Access Timing



Note: For DACKn, an example is shown where CHCRn.AL (acknowledge level) = 0 for the DMAC.

Figure 10.35 Burst ROM Wait Access Timing

10.6.6 PCMCIA Interface

In this LSI, setting the A56PCM bit in BCR1 to 1 allows the bus interface for off-chip memory space areas 5 and 6 to become an IC memory card interface or I/O card interface as stipulated in JEIDA specification version 4.2 (PCMCIA2.1).

Figure 10.36 shows an example of PCMCIA card connection to this LSI. To enable hot swapping of the PCMCIA cards (i.e. insertion or removal while system power is being supplied), a tri-state buffer must be connected between this LSI bus interface and the PCMCIA cards.

Since operation in big endian mode is not explicitly stipulated in the JEIDA/PCMCIA standard, this LSI supports only little-endian mode setting and the little-endian mode PCMCIA interface.

When the MMU is on, PCMCIA interface memory space can be set in MMU page units, and there is a choice of 8-bit shared memory, 16-bit shared memory, 8-bit attribute memory, 16-bit attribute memory, 8-bit I/O space, 16-bit I/O space, or dynamic bus sizing. See section 6, Memory Management Unit (MMU), for details of the setting method. When the MMU is off, the memory space is always used for access in the setting of bits SA2 to SA0 in PTEA.

0	0	0	Reserved (Setting prohibited)
		1	Dynamic I/O bus sizing
	1	0	8-bit I/O space
		1	16-bit I/O space
1	0	0	8-bit shared memory
		1	16-bit shared memory
	1	0	8-bit attribute memory
		1	16-bit attribute memory

When the MMU is on, wait cycles in a bus access can be set in MMU page units. For details of the setting method, see section 6, Memory Management Unit (MMU). When the MMU is off (MMUCR.AT = 0), access is always performed according to the TC bit in PTEA. When the TC bit is cleared to 0, bits A5W2 to A5W0 in WCR2 and bits A5PCW1 to A5PCW0, A5TED2 to A5TED0, and A5TEH2 to A5TEH0 in PCR are selected. When TC is set to 1, bits A6W2 to A6W0 in WCR2 and bits A6PCW1 to A6PCW0, A6TED2 to A6TED0, and A6TEH2 to A6TEH0 in PCR are selected.

Access to a PCMCIA interface area by the DMAC is always performed using the DMAC's CHCRn.SSAn, CHCRn.DSAn, CHCRn.STC, and CHCRn.DTC values.

Bits AnPCW1 and AnPCW0 in PCR can be used to set the number of wait cycles as 0, 15, 30, or 50 to be inserted in a low-speed bus cycle. This value is added to the number of inserted wait cycles specified by WCR2. Bits AnTED2 to AnTED0 in PCR (with a setting range from 0 to 15) can be used to retain the $\overline{RD}$ and $\overline{WE1}$ signal addresses and the $\overline{CS}$, $\overline{CE2A}$, $\overline{CE2B}$, and $\overline{REG}$ setup times. Bits AnTEH2 to AnTEH0 in PCR (with a setting range from 0 to 15) can also be used to retain the $\overline{RD}$ and $\overline{WE1}$ signal addresses and the $\overline{CS}$, $\overline{CE2A}$, $\overline{CE2B}$, and $\overline{REG}$ data hold times.

Bits A5IW2 to A5IW0 and A6IW2 to A6IW0 in WCR1 are used to set the number of wait cycles between cycles. The selected number of wait cycles between cycles depends only on the area accessed (area 5 or 6). When area 5 is accessed, bits A5IW2 to A5IW0 are selected, and when area 6 is accessed, bits A6IW2 to A6IW0 are selected.

In 32-byte transfer, a total of 32 bytes are transferred consecutively according to the bus width that was set. The first access is performed on the data where there was an access request, and the remaining accesses are performed on the 32-byte boundary data in wraparound mode. The bus is not released during this operation.

Bus Width (Bits)	Read/ Write	Access Size (Bits)*[1]	Odd/ Even	IOIS16	Access	CE2	CE1	A0	D15 to D8	D7 to D0
8	Read	8	Even	*	—	1	0	0	Invalid	Read data
			Odd	*	—	1	0	1	Invalid	Read data
		16	Even	*	First	1	0	0	Invalid	Lower read data
			Even	*	Second	1	0	1	Invalid	Upper read data
			Odd	*	—	—	—	—	—	—
	Write	8	Even	*	—	1	0	0	Invalid	Write data
			Odd	*	—	1	0	1	Invalid	Write data
		16	Even	*	First	1	0	0	Invalid	Lower write data
			Even	*	Second	1	0	1	Invalid	Upper write data
			Odd	*	—	—	—	—	—	—
16	Read	8	Even	*	—	1	0	0	Invalid	Read data
			Odd	*	—	0	1	1	Read data	Invalid
		16	Even	*	—	0	0	0	Upper read data	Lower read data
			Odd	*	—	—	—	—	—	—
	Write	8	Even	*	—	1	0	0	Invalid	Write data
			Odd	*	—	0	1	1	Write data	Invalid
		16	Even	*	—	0	0	0	Upper write data	Lower write data
			Odd	*	—	—	—	—	—	—

(Bits)	Write	(Bits)*[1]	Even	IOIS16	Access	CE2	CE1	A0	D15 to D8	D7 to D0
Dynamic bus sizing*[2]	Read	8	Even	0	—	1	0	0	Invalid	Read data
			Odd	0	—	0	1	1	Read data	Invalid
		16	Even	0	—	0	0	0	Upper read data	Lower read data
			Odd	0	—	—	—	—	—	—
	Write	8	Even	0	—	1	0	0	Invalid	Write data
			Odd	0	—	0	1	1	Write data	Invalid
		16	Even	0	—	0	0	0	Upper write data	Lower write data
			Odd	0	—	—	—	—	—	—
	Read	8	Even	1	—	1	0	0	Invalid	Read data
			Odd	1	First	0	1	1	Ignored	Invalid
			Odd	1	Second	1	0	1	Invalid	Read data
		16	Even	1	First	0	0	0	Invalid	Lower read data
			Even	1	Second	1	0	1	Invalid	Upper read data
			Odd	1	—	—	—	—	—	—
	Write	8	Even	1	—	1	0	0	Invalid	Write data
			Odd	1	First	0	1	1	Invalid	Write data
			Odd	1	Second	1	0	1	Invalid	Write data
		16	Even	1	First	0	0	0	Upper write data	Lower write data
			Even	1	Second	1	0	1	Invalid	Upper write data
			Odd	1	—	—	—	—	—	—

Notes: * Don't Care

1. In 32-bit/64-bit/32-byte transfer, the addresses are automatically divided into increments equal to the bus width, and the above accesses are repeated until the transfer data size is reached.
2. PCMCIA I/O card interface only

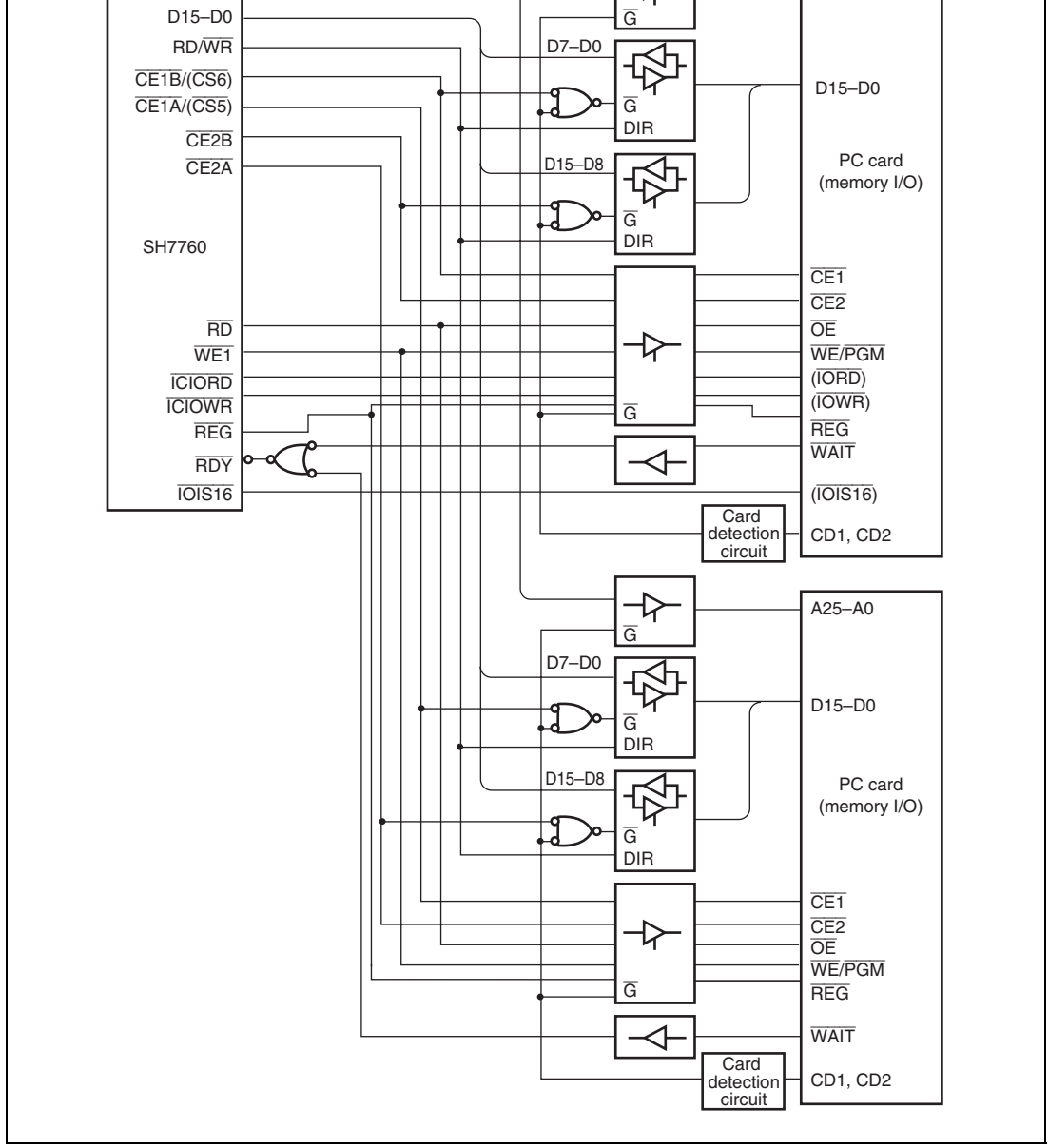
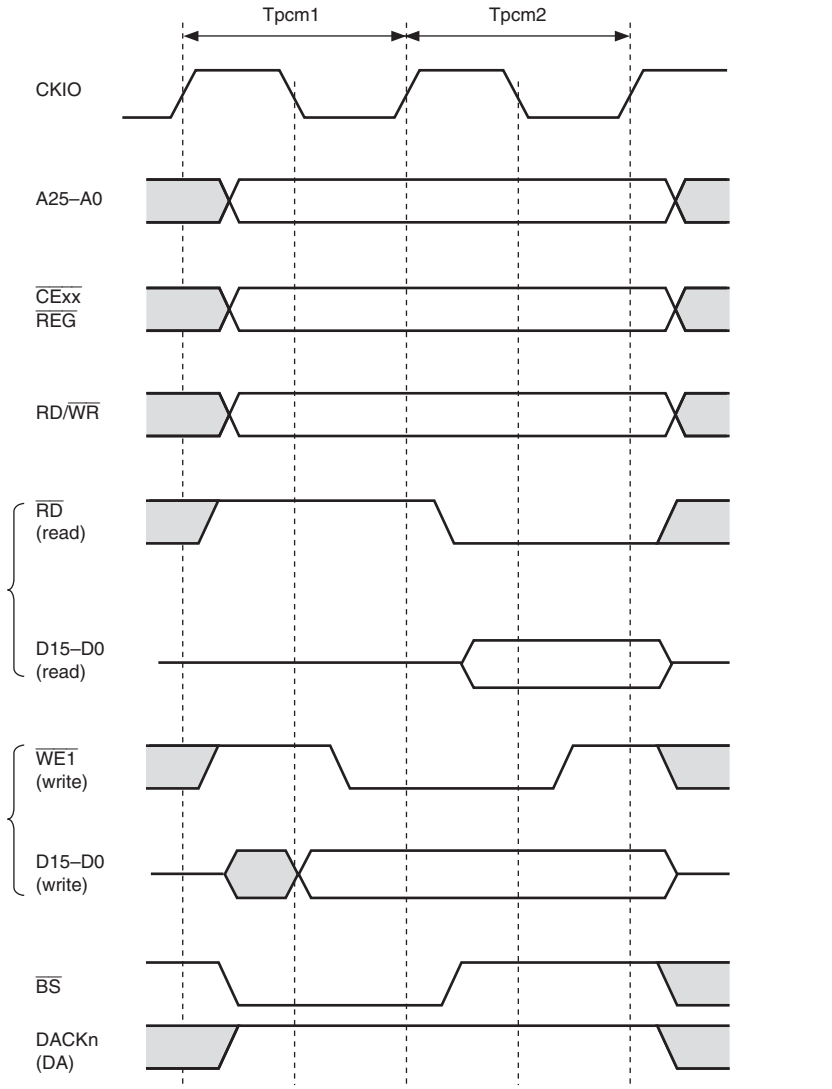


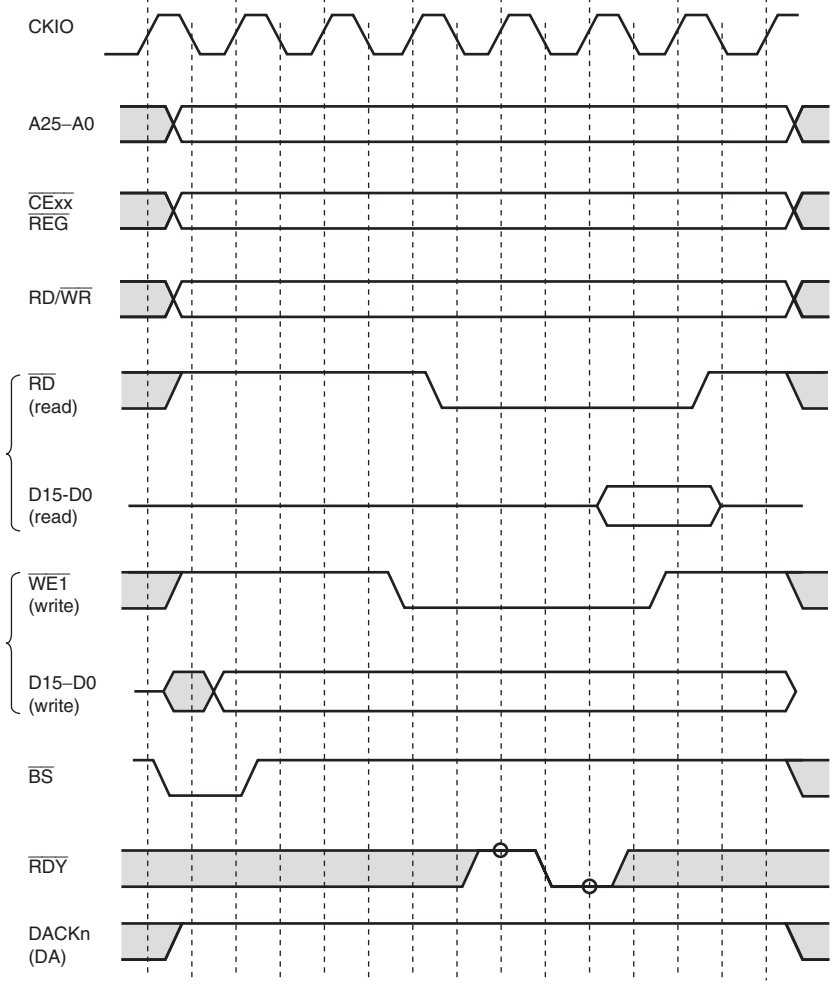
Figure 10.36 Example of PCMCIA Interface

Figure 10.37 shows the basic timing for the PCMCIA memory card interface, and figure 10.38 shows the wait timing for the PCMCIA memory card interface.



Note: For DACKn, an example is shown where CHCRn.AL (acknowledge level) = 0 for the DMAC.

Figure 10.37 Basic Timing for PCMCIA Memory Card Interface



Note: For DACKn, an example is shown where CHCRn.AL (acknowledge level) = 0 for the DMAC.

Figure 10.38 Wait Timing for PCMCIA Memory Card Interface

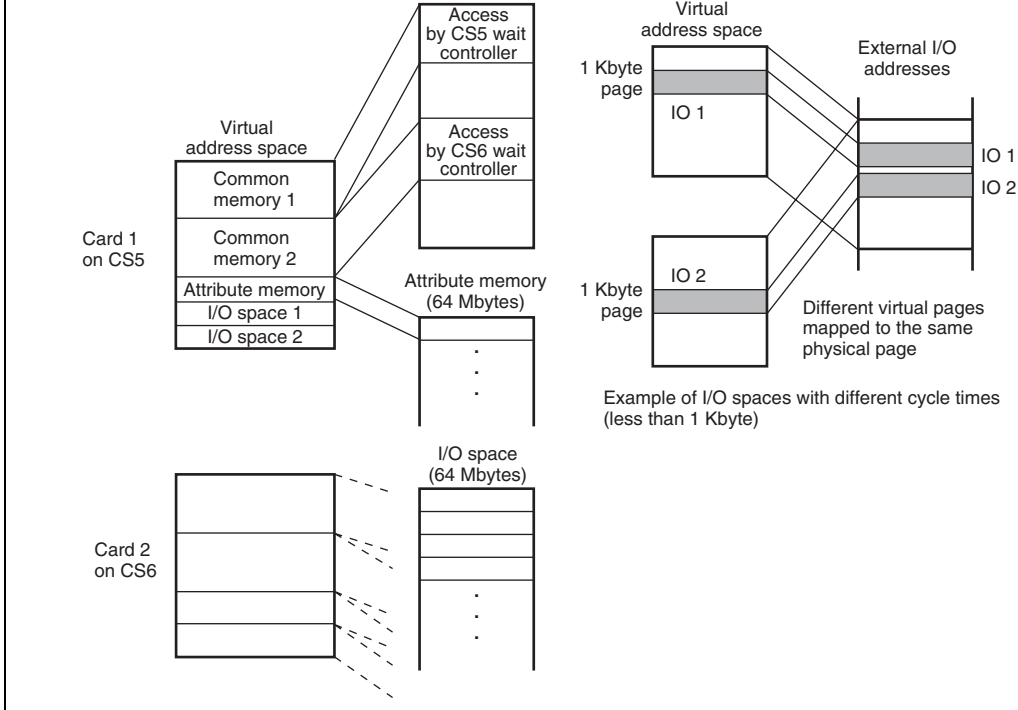


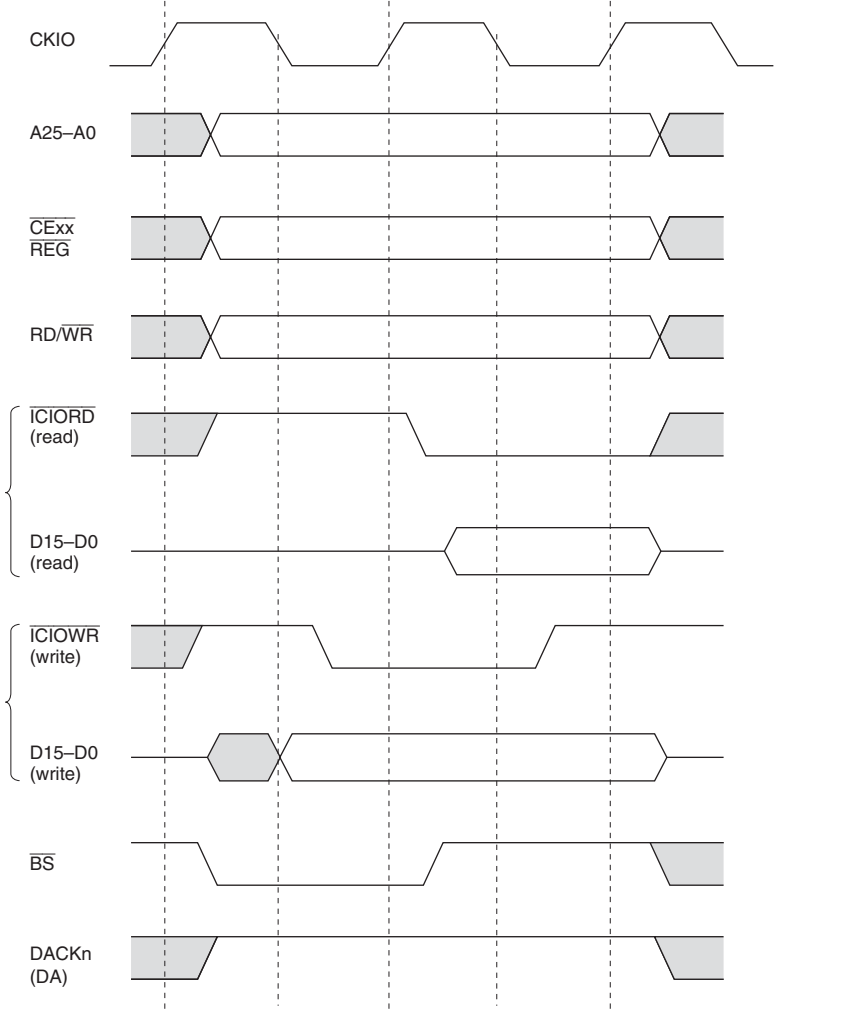
Figure 10.39 PCMCIA Space Allocation

(2) I/O Card Interface Timing

Figures 10.40 and 10.41 show the timing for the PCMCIA I/O card interface.

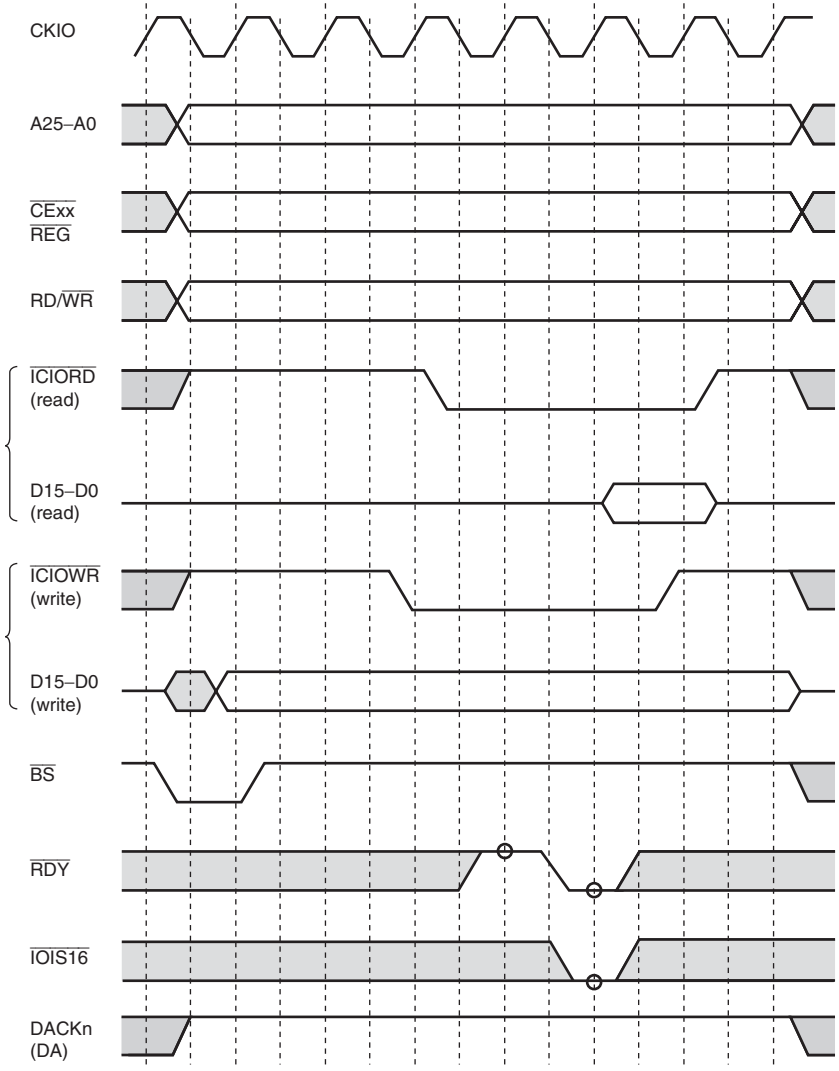
When accessing a PCMCIA card as an I/O card interface, it is possible to perform dynamic sizing of the I/O bus width using the $\overline{\text{IOIS16}}$ pin. With a 16-bit bus width selected, if the $\overline{\text{IOIS16}}$ signal is high during a word-size I/O bus cycle, the I/O port is recognized as 8 bits in width. In this case, a data access for only 8 bits is performed in the I/O bus cycle being executed, and this is automatically followed by a data access for the remaining 8 bits. Dynamic bus sizing is also performed for byte-size access to address 2n + 1.

Figure 10.42 shows the basic timing for dynamic bus sizing.



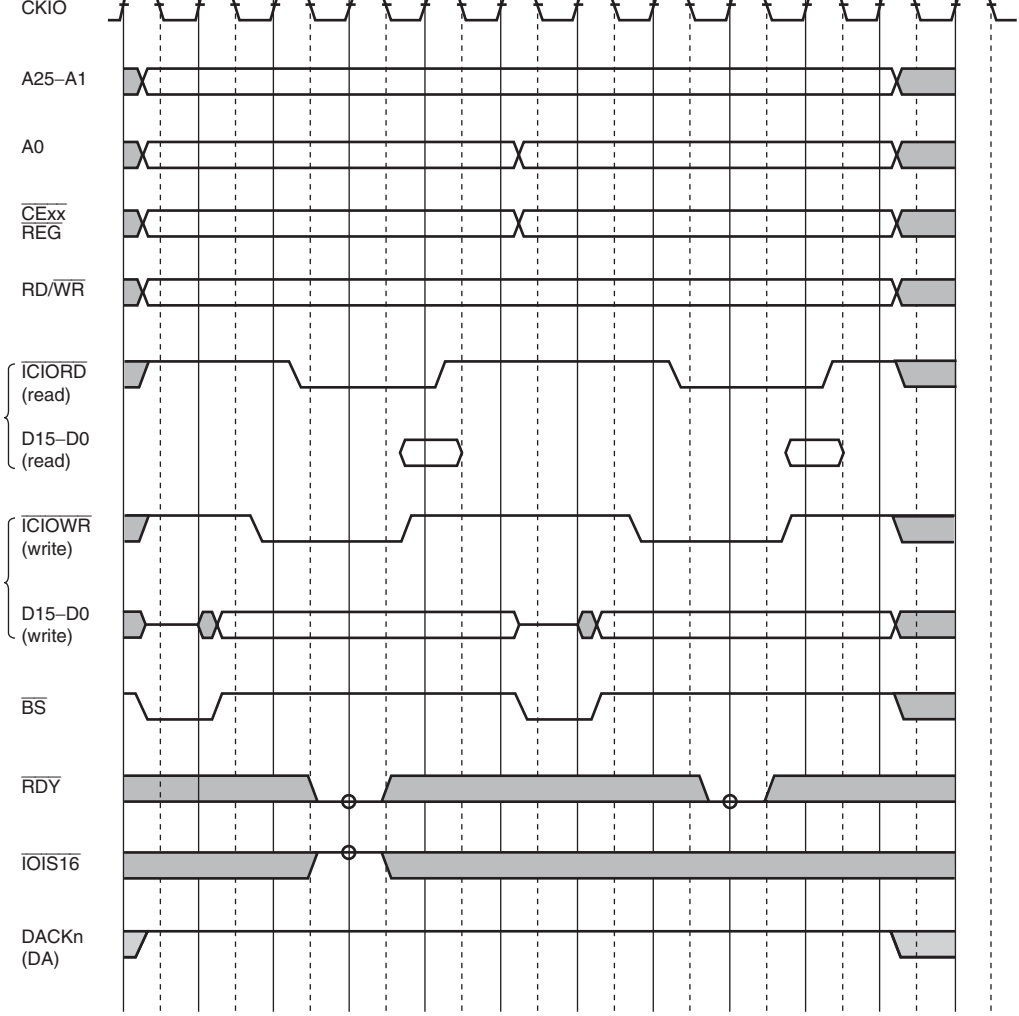
Note: For DACKn, an example is shown where CHCRn.AL (acknowledge level) = 0 for the DMAC.

Figure 10.40 Basic Timing for PCMCIA I/O Card Interface



Note: For DACKn, an example is shown where CHCRn.AL (acknowledge level) = 0 for the DMAC.

Figure 10.41 Wait Timing for PCMCIA I/O Card Interface



Note: For DACKn, an example is shown where CHCRn.AL (acknowledge level) = 0 for the DMAC.

Figure 10.42 Dynamic Bus Sizing Timing for PCMCIA I/O Card Interface

If the MD6 pin is cleared to 0 at a power-on reset by the $\overline{\text{RESET}}$ pin, the MPX interface is selected for area 0. The MPX interface is selected for areas 1 to 6 by the MPX bit in BCR1 and bits MEMMODE, A4MPX, and A1MPX in BCR3. The MPX interface provides an address/data multiplex type bus protocol and allows easy connection with off-chip memory controller chips using a single 32-bit address/data multiplex bus. A bus cycle consists of an address phase and a data phase, with address information output to D25 to D0 and the access size output to D31 to D29 in the address phase. The $\overline{\text{BS}}$ signal is asserted for one cycle to indicate the address phase. The $\overline{\text{CSn}}$ signal is asserted at the rising edge in Tm1 and is negated after the end of the last data transfer in the data phase. Therefore, a negation cycle does not occur in the case of minimum pitch access. The $\overline{\text{FRAME}}$ signal is asserted at the rising edge in Tm1 and negated at the start of the last data transfer cycle in the data phase. Therefore, an off-chip device for the MPX interface must internally store the address information and access size output in the address phase and perform data input/output for the data phase. For details of access sizes and data alignment, see section 10.6.1, Endian/Access Size and Data Alignment.

Values output to address pins A25 to A20 are not guaranteed.

In 32-byte transfer, a total of 32 bytes are transferred consecutively according the bus width that was set. The first access is performed on the data where there was an access request, and the remaining accesses are performed on 32-byte boundary data. If the access size exceeds the set bus width in this way, the address is output once, and then the burst access is performed with multiple continuous data cycles. The bus is not released during this operation.

D31	D30	D29	Access Size
0	0	0	Byte
		1	Word
	1	0	Longword
		1	Quadword
1	x	x	32-byte burst

x: Don't care

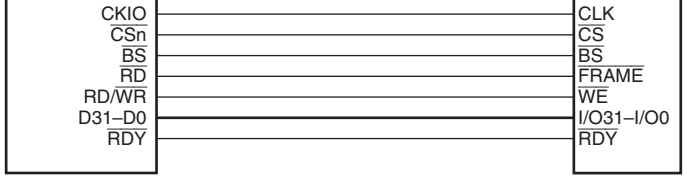


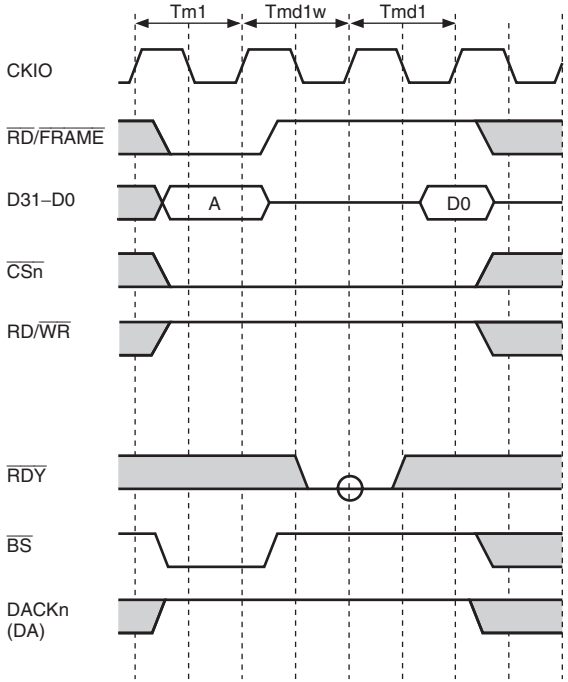
Figure 10.43 Example of 32-Bit Data Width MPX Connection

The MPX interface timing is shown below.

When the MPX interface is used for areas 1 to 6, a bus size of 32 bits should be specified by BCR2.

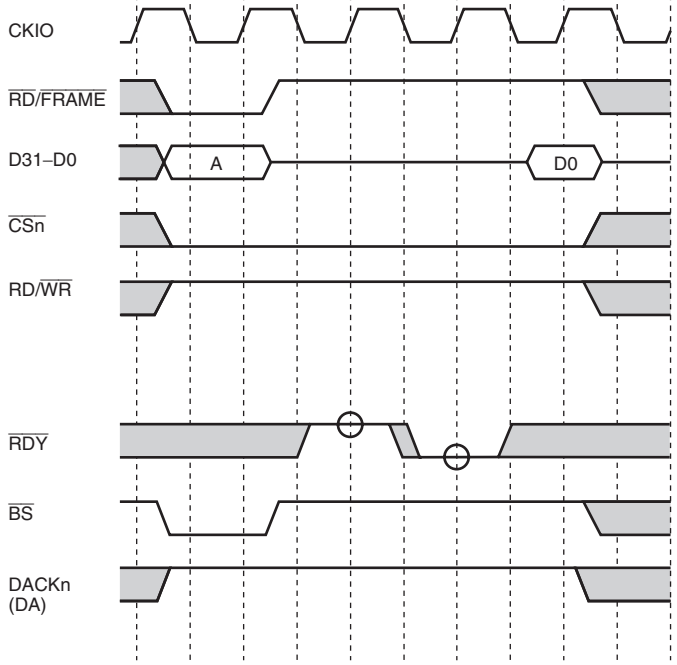
In wait control, waits can be specified by WCR2 and waits can be inserted by the RDY pin.

In a read, one wait cycle is automatically inserted after address output, even if WCR2 is cleared to 0.



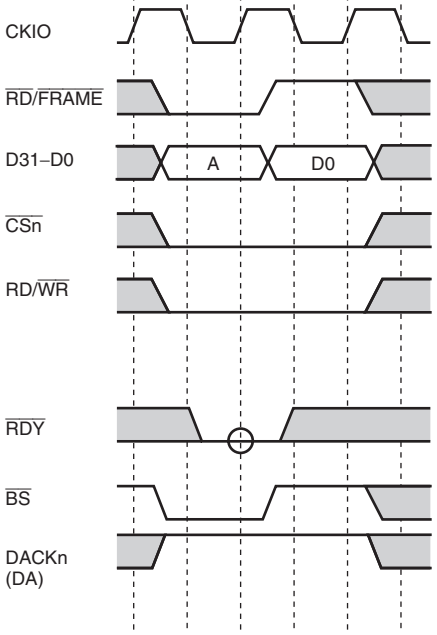
Note: For DACKn, an example is shown where CHCRn.AL (acknowledge level)= 0 for the DMAC.

Figure 10.44 MPX Interface Timing 1 (Single Read Cycle, AnW = 0, No External Wait)



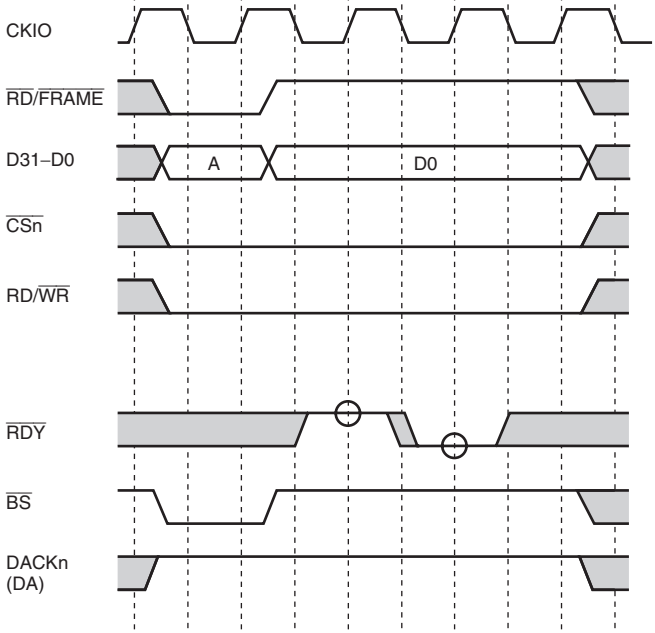
Note: For DACKn, an example is shown where CHCRn.AL (acknowledge level) = 0 for the DMAC.

Figure 10.45 MPX Interface Timing 2 (Single Read, AnW = 0, One External Wait Inserted)



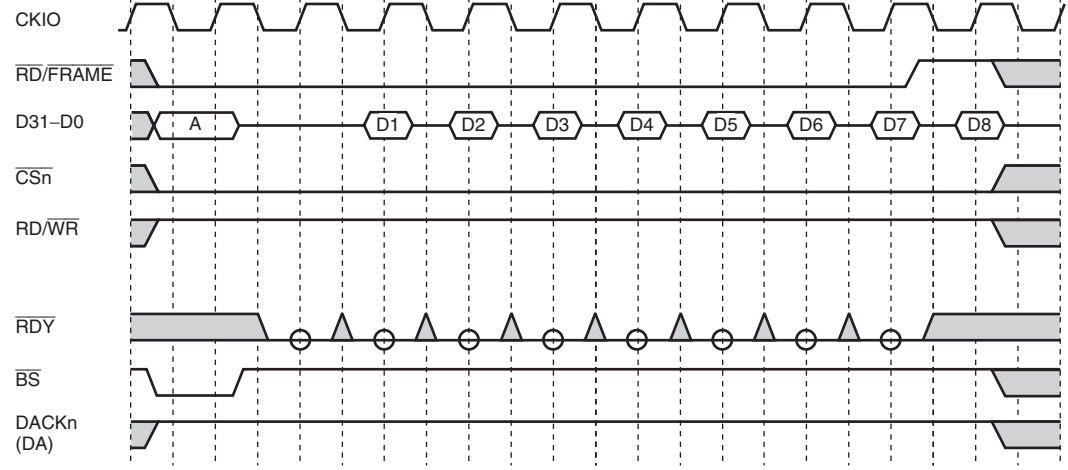
Note: For DACKn, an example is shown where CHCRn.AL (acknowledge level) = 0 for the DMAC.

Figure 10.46 MPX Interface Timing 3 (Single Write Cycle, AnW = 0, No External Wait)



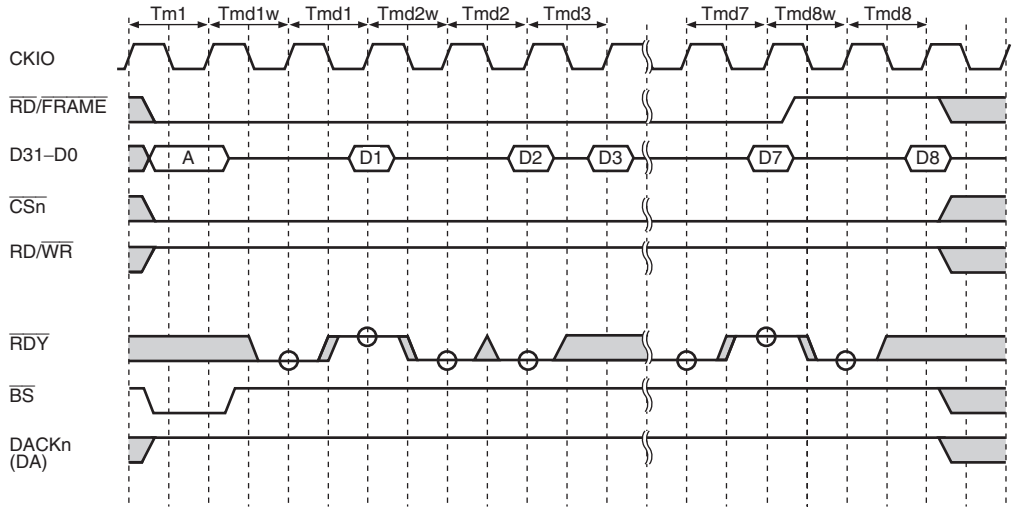
Note: For DACKn, an example is shown where CHCRn.AL (acknowledge level) = 0 for the DMAC.

Figure 10.47 MPX Interface Timing 4 (Single Write, AnW = 1, One External Wait Inserted)



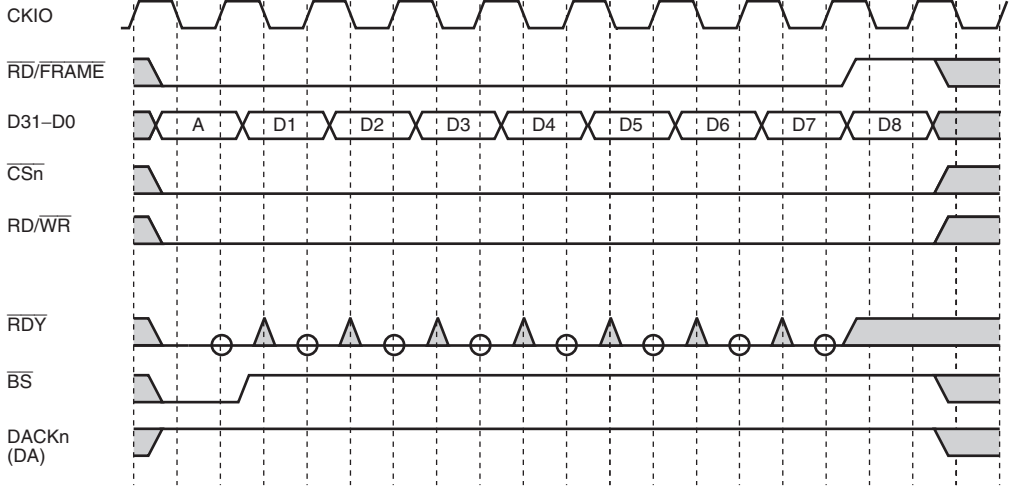
Note: For DACKn, an example is shown where CHCRn.AL (acknowledge level) = 0 for the DMAC.

Figure 10.48 MPX Interface Timing 5 (Burst Read Cycle, AnW = 0, No External Wait, 32-Bit Bus Width, 32-Byte Data Transfer)



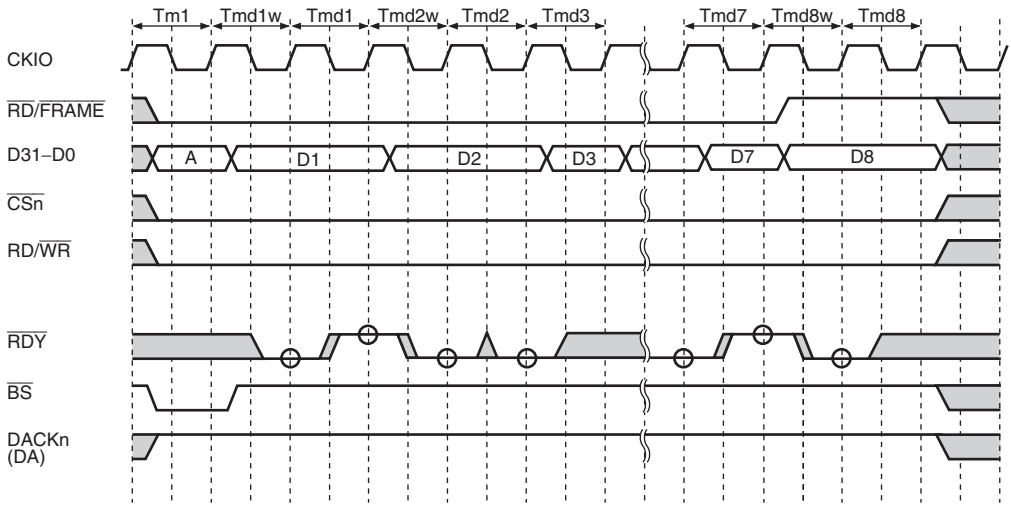
Note: For DACKn, an example is shown where CHCRn.AL (acknowledge level) = 0 for the DMAC.

Figure 10.49 MPX Interface Timing 6 (Burst Read Cycle, AnW = 0, External Wait Control, 32-Bit Bus Width, 32-Byte Data Transfer)



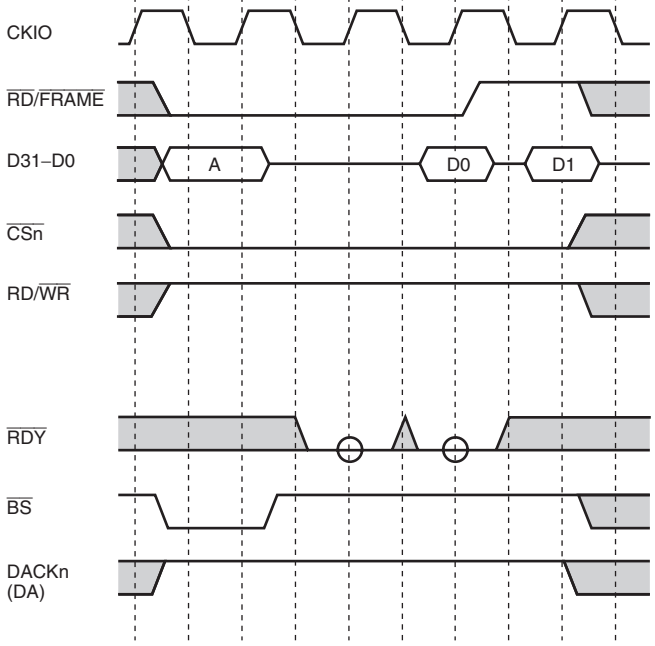
Note: For DACKn, an example is shown where CHCRn.AL (acknowledge level) = 0 for the DMAC.

Figure 10.50 MPX Interface Timing 7 (Burst Write Cycle, AnW = 0, No External Wait, 32-Bit Bus Width, 32-Byte Data Transfer)



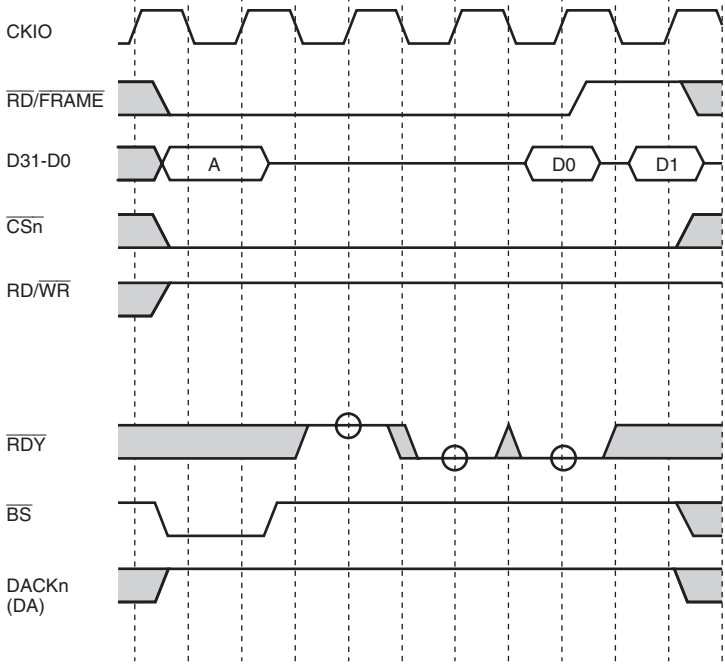
Note: For DACKn, an example is shown where CHCRn.AL (acknowledge level) = 0 for the DMAC.

Figure 10.51 MPX Interface Timing 8 (Burst Write Cycle, AnW = 1, External Wait Control, 32-Bit Bus Width, 32-Byte Data Transfer)



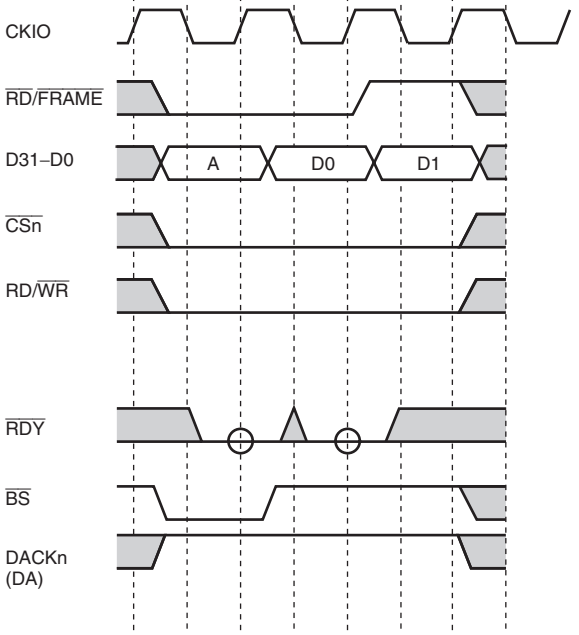
Note: For DACKn, an example is shown where CHCRn.AL (acknowledge level) = 0 for the DMAC.

Figure 10.52 MPX Interface Timing 9 (Burst Read Cycle, AnW = 0, No External Wait, 32-Bit Bus Width, 64-Bit Data Transfer)



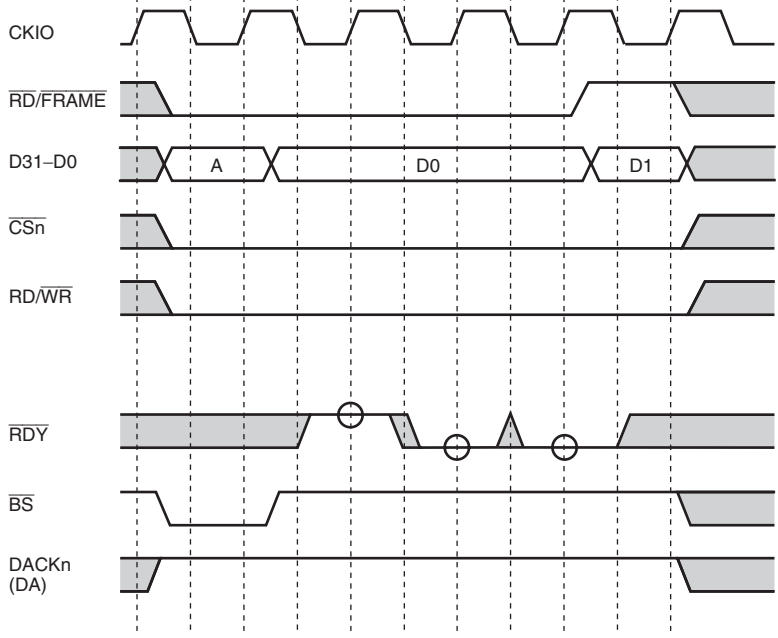
Note: For DACKn, an example is shown where CHCRn.AL (acknowledge level) = 0 for the DMAC.

Figure 10.53 MPX Interface Timing 10 (Burst Read Cycle, AnW = 0, One External Wait Inserted, 32-Bit Bus Width, 64-Bit Data Transfer)



Note: For DACKn, an example is shown where CHCRn.AL (acknowledge level) = 0 for the DMAC.

Figure 10.54 MPX Interface Timing 11 (Burst Write Cycle, AnW = 0, No External Wait, 32-Bit Bus Width, 64-Bit Data Transfer)



Note: For DACKn, an example is shown where CHCRn.AL (acknowledge level) = 0 for the DMAC.

Figure 10.55 MPX Interface Timing 12 (Burst Write Cycle, AnW = 1, One External Wait Inserted, 32-Bit Bus Width, 64-Bit Data Transfer)

10.6.8 Byte Control SRAM Interface

The byte control SRAM interface is a memory interface that outputs a byte select strobe ($\overline{\text{WEn}}$) in both read and write bus cycles. This interface has 16-bit data pins and can be connected to SRAM having upper byte select strobe and lower byte select strobe functions such as UB and LB.

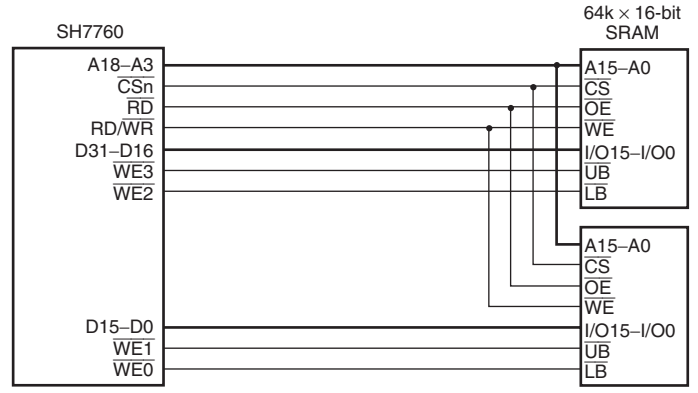
Areas 1 and 4 can be specified as a byte control SRAM interface. However, when these areas are set to MPX mode, MPX mode has priority.

The write timing for the byte control SRAM interface is identical to that of the normal SRAM interface.

In read operations, on the other hand, the $\overline{\text{WEn}}$ pin timing is different. In a read access, only the $\overline{\text{WE}}$ signal for the byte being read is asserted. Assertion is synchronized with the falling edge of the CKIO clock in the same way as the $\overline{\text{WE}}$ signal, while negation is synchronized with the rising edge of the CKIO clock in the same way as the $\overline{\text{RD}}$ signal.

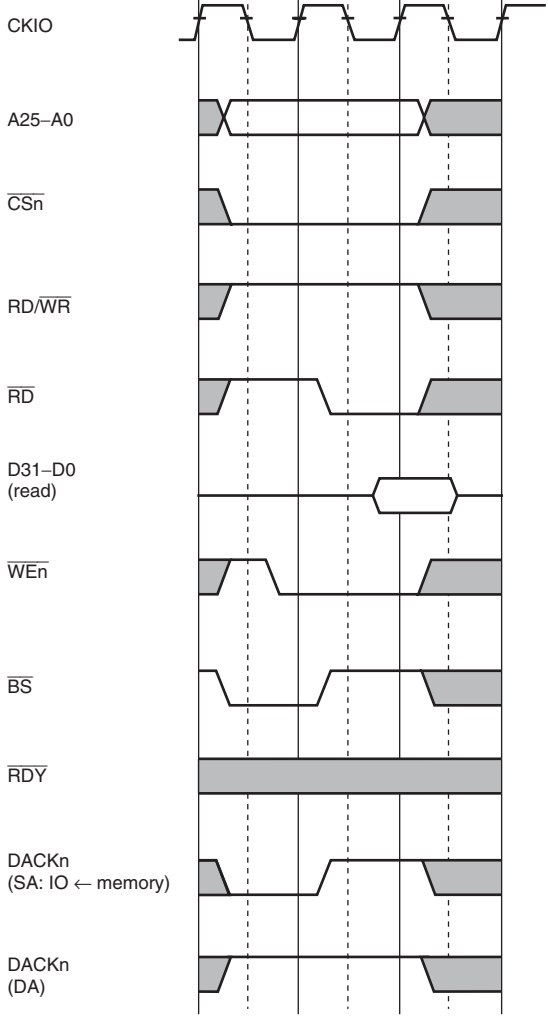
was set. The first access is performed on the data where there was an access request. The remaining accesses are performed on the 32-byte boundary data in wraparound mode. The bus is not released during this operation.

Figure 10.56 shows an example of a byte control SRAM connection, and figures 10.57 to 10.59 show examples of byte control SRAM read cycles.



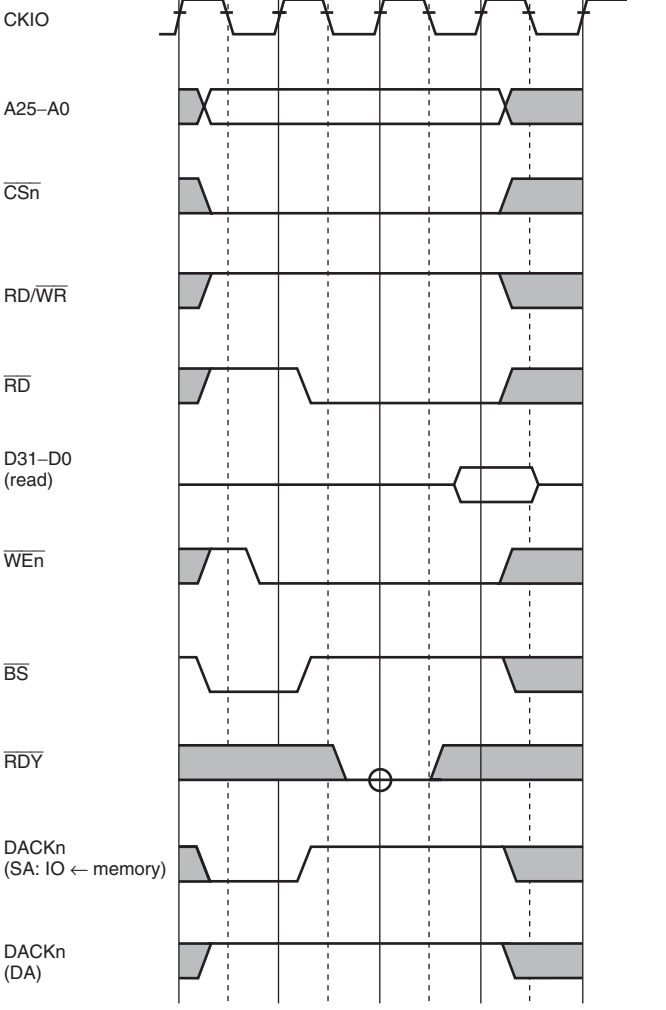
Note: For DACKn, an example is shown where CHCRn.AL (acknowledge level) = 0 for the DMAC.

Figure 10.56 Example of 32-Bit Data Width Byte Control SRAM



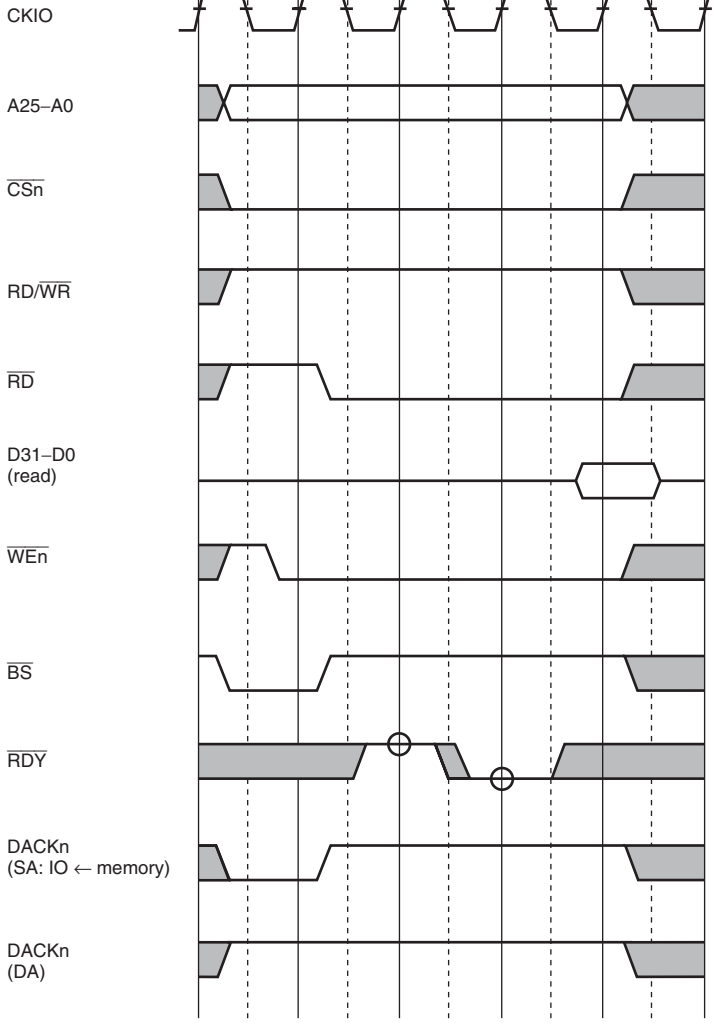
Note: For DACKn, an example is shown where CHCRn.AL (acknowledge level) = 0 for the DMAC.

Figure 10.57 Byte Control SRAM Basic Read Cycle (No Wait)



Note: For DACKn, an example is shown where CHCRn.AL (acknowledge level) = 0 for the DMAC.

Figure 10.58 Byte Control SRAM Basic Read Cycle (One Internal Wait Cycle)



Note: For DACKn, an example is shown where CHCRn.AL (acknowledge level) = 0 for the DMAC.

Figure 10.59 Byte Control SRAM Basic Read Cycle (One Internal Wait + One External Wait)

A problem associated with higher operating frequencies for off-chip memory buses is that the data buffer turn-off after completion of a read from a low-speed device may be too slow, causing a collision with the data in the next access, and resulting in lower reliability or malfunctions. To prevent this problem, this module provides a data collision prevention function. It stores the preceding access area and the type of read/write and inserts a wait cycle before the access cycle if there is a possibility of a bus collision when the next access is started. The process for wait cycle insertion consists of inserting idle cycles between access cycles as shown in section 10.5.5, Wait Control Register (WCR1). When this LSI performs consecutive write cycles, the data transfer direction is fixed (from this LSI to other memory), and there is no problem. Also, for read accesses to the same area, generally, data is output from the same data buffer, and no wait cycle is inserted. If bits AnIW2 to AnIW0 (n = 0 to 6) in WCR1 are used to set idle cycles between accesses, the number of inserted idle cycles is only the specified number of idle cycles minus the number of idle cycles specified by the bits.

When bus arbitration is performed, the bus is released after wait cycles are inserted between cycles.

In single address mode DMA transfer from an I/O device to memory, the I/O device speed determines the data on the bus. When a low-speed I/O device is used, a wait time equivalent to the output buffer turn-off time must be inserted between cycles. When a high-speed memory is used, the memory may be unable to run at full speed since insertion of waits between cycles may be required to adjust to the speed of a low-speed device for enabling DMA transfer.

Waits between cycles can be specified with bits DMAIW2 to DMAIW0 in wait control register 1 (WCR1) for single address mode DMA transfer from an I/O device to memory. The number of waits that can be inserted should be in the range from 0 to 15. A number of waits specified by the DMAIW2 to DMAIW0 bits are inserted in single address DMA transfers to all areas.

In dual address mode DMA transfer, the normal wait between cycles specified by bits AnIW2 to AnIW0 (n = 0 to 6) is inserted.

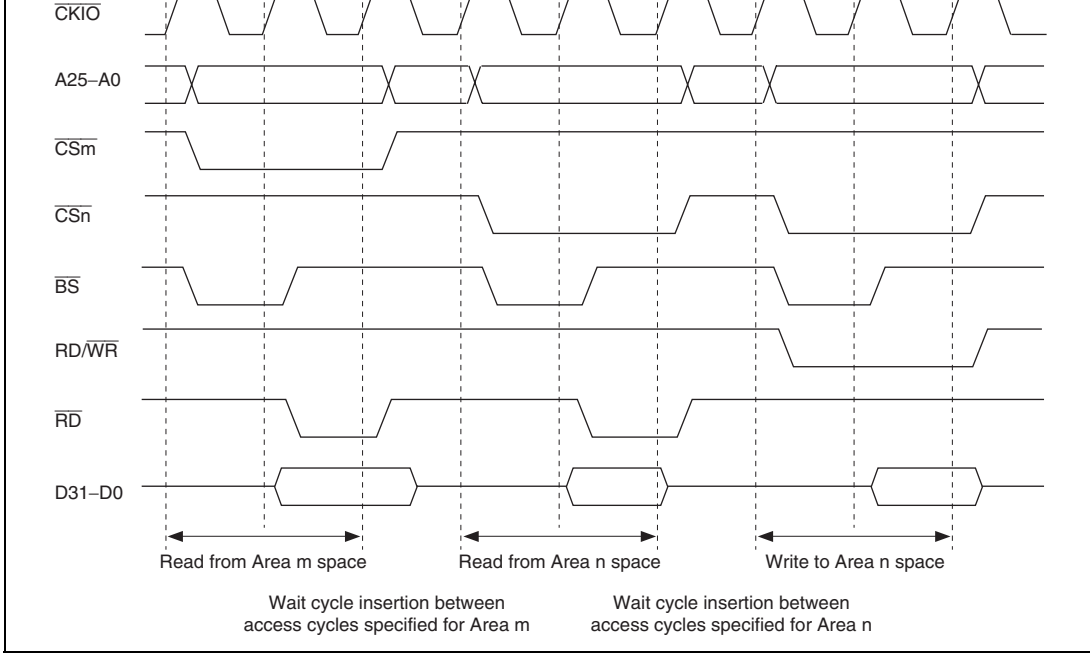


Figure 10.60 Wait Cycles between Access Cycles

10.6.10 Bus Arbitration

This LSI provides a bus arbitration function that allows an off-chip device to control the bus in response to a bus request.

In normal operation, this LSI controls the bus, and it releases the bus to transfer the right to access the bus when an off-chip device issues a bus request. In the description below, an off-chip device that issues bus requests is referred to as a slave.

This LSI has two internal bus masters: the CPU and DMAC. When synchronous DRAM is connected and refresh control is performed, the refresh request serves as the third bus master. Bus requests from off-chip devices are also added when this LSI is in master mode. If requests occur simultaneously, the priority from highest to lowest is based on the following order: bus request from an off-chip device, the refresh request, the DMAC, and the CPU.

To prevent malfunctions of connected devices when the right to access the bus is transferred from master to slave, all bus control signals are negated before the bus is released. When the right of access to the bus is received, bus control signals switch from the negated level to start driving the

access the bus, output buffer collisions can be avoided.

The right of access to the bus is transferred at the end of bus cycles.

When the bus release request signal ($\overline{\text{BREQ}}$) is asserted, this LSI releases the bus as soon as the current bus cycle being executed ends and outputs the bus request acknowledge signal ($\overline{\text{BACK}}$). However, the bus is not released during multiple bus cycles generated due to a smaller data bus width than the access size (such as when performing longword access to 8-bit bus width memory) or during a 32-byte transfer such as a cache fill or write-back. The bus is also not released between read and write cycles during execution of a TAS instruction, or between read and write cycles when DMAC dual address transfer is executed. When $\overline{\text{BREQ}}$ is negated, $\overline{\text{BACK}}$ is negated and use of the bus is resumed.

When a refresh request is generated, this LSI performs a refresh operation as soon as the current bus cycle being executed ends. However, refresh operations are deferred during multiple bus cycles generated due to a smaller data bus width than the access size (such as when performing longword access to 8-bit bus width memory) and during a 32-byte transfer such as a cache fill or write-back. The refresh operation is also deferred between read and write cycles during execution of a TAS instruction, and between read and write cycles when DMAC dual address transfer is executed. Refresh operations are also deferred in the bus release state.

Since the CPU in this LSI is connected to cache memory by a dedicated peripheral bus, the CPU can still read from cache memory when the bus is being used by another bus master inside or outside this LSI. When writing from the CPU, an off-chip write cycle is generated when write-through has been set for the cache in this LSI, or when an access is made to a cache-off area. This results in a delay until the bus is returned.

When this LSI wants to take back the bus in response to an internal memory refresh request, it negates $\overline{\text{BACK}}$. A device that asserts the off-chip bus release request receives the $\overline{\text{BACK}}$ negation, and then negates $\overline{\text{BREQ}}$ to release the bus. In this way, the bus is returned to this LSI, and then processing is performed.

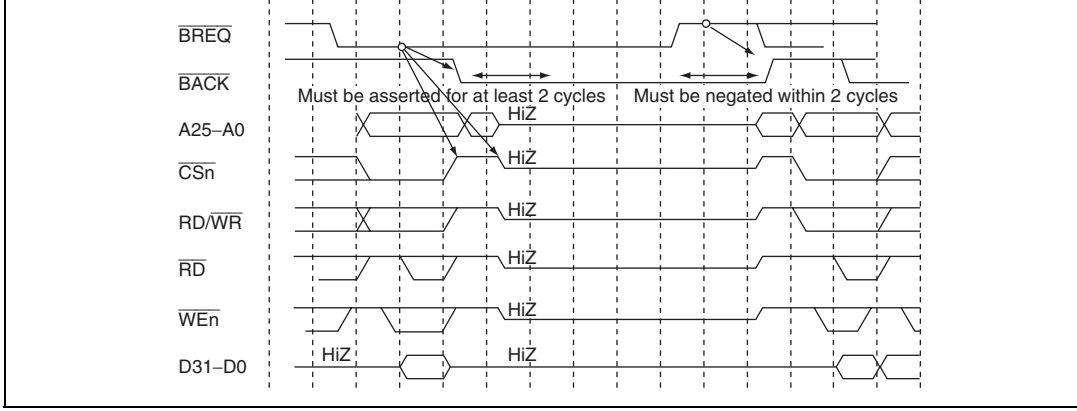


Figure 10.61 Arbitration Sequence

10.6.11 Bus Release and Acquire Sequences

This LSI controls the bus unless it receives a bus request.

As soon as an assertion (low level) of the bus request signal ($\overline{\text{BREQ}}$) is received from an off-chip device and the current bus cycle being executed ends, this LSI releases the bus and asserts (low level) the bus request acknowledge signal ($\overline{\text{BACK}}$). If a bus request has not been issued due to a refresh request, this LSI receives the $\overline{\text{BREQ}}$ negation (high level) indicating that the slave has released the bus, and then negates (drives high) the $\overline{\text{BACK}}$ signal and resumes use of the bus.

If a bus request is issued due to a refresh request in the bus release state, this LSI negates the bus request acknowledge signal ($\overline{\text{BACK}}$) and then receives the $\overline{\text{BREQ}}$ negation indicating that the slave has released the bus, and resumes use of the bus.

When the bus is released, all bus control output signals and input/output signals pertaining to the bus interface go to the high-impedance state except for CKE in the synchronous DRAM interface, $\overline{\text{BACK}}$ (bus request acknowledge) in bus arbitration, and DACK0 and DACK1 for DMA transfer control.

For synchronous DRAM, a precharge command is issued for the active bank, and the bus is released after precharging is completed.

The following is the specific bus release sequence.

First, the bus request acknowledge signal is asserted at the rising edge of the clock. The address bus and data bus go to the high-impedance state in synchronization with the assertion of $\overline{\text{BACK}}$. At the same time, the bus control signals ($\overline{\text{BS}}$, $\overline{\text{CSn}}$, $\overline{\text{RAS}}$, $\overline{\text{WEn}}$, $\overline{\text{RD}}$, RD/$\overline{\text{WR}}$, $\overline{\text{CE2A}}$, and $\overline{\text{CE2B}}$)

before going to high-impedance. The bus request signal is sampled at the rising edge of the clock.

The following is the specific bus reacquiring sequence from the slave.

As soon as $\overline{\text{BREQ}}$ negation is detected at the rising edge of the clock, $\overline{\text{BACK}}$ is negated and the bus control signal driving is started from next rising edge of the clock. Driving of the address bus and data bus also starts at the same rising edge of the clock. The bus control signals are asserted and the bus cycle is actually started, at the earliest, at the next rising edge of the clock where the driving of the bus control signals was started.

In order to reacquire the bus and start execution of a refresh operation or bus access, the $\overline{\text{BREQ}}$ signal must be negated for at least two cycles.

If a refresh request is generated when $\overline{\text{BACK}}$ has been asserted and the bus has been released, the $\overline{\text{BACK}}$ signal is negated even while the $\overline{\text{BREQ}}$ signal is asserted to request a slave to release the bus. When this LSI is used in master mode with slaves designed independently by the user, consecutive bus accesses may be attempted to reduce the overhead due to arbitration. When connecting a slave where the total duration of consecutive accesses exceeds the refresh cycle, it should be designed so that the bus is released as soon as possible after negation of the $\overline{\text{BACK}}$ signal is detected.

10.7 Usage Notes

10.7.1 Refresh

Auto refresh operations are not carried out when this LSI enters software standby, hardware standby, or deep-sleep mode. If the memory system requires refresh operations, set the memory in the self-refresh state prior to making the transition to software standby, hardware standby, or deep-sleep mode.

10.7.2 Bus Arbitration

The bus is not released when this LSI enters software standby or deep-sleep mode. In systems performing bus arbitration, clear the bus release request enable bit (BCR1.BREQEN) to 0 for the processor in master mode before making the transition to software standby or deep-sleep mode. Correct operation is not guaranteed when a transition is made to software standby mode or deep-sleep mode with BCR1.BREQEN = 1.

This LSI includes an on-chip eight-channel direct memory access controller (DMAC). The DMAC can be used in place of the CPU to perform high-speed data transfers among external devices equipped with DACK (DMA transfer end notification), external memories, memory-mapped external devices, and on-chip peripheral modules. Using the DMAC reduces the burden on the CPU and increases the operating efficiency of this LSI.

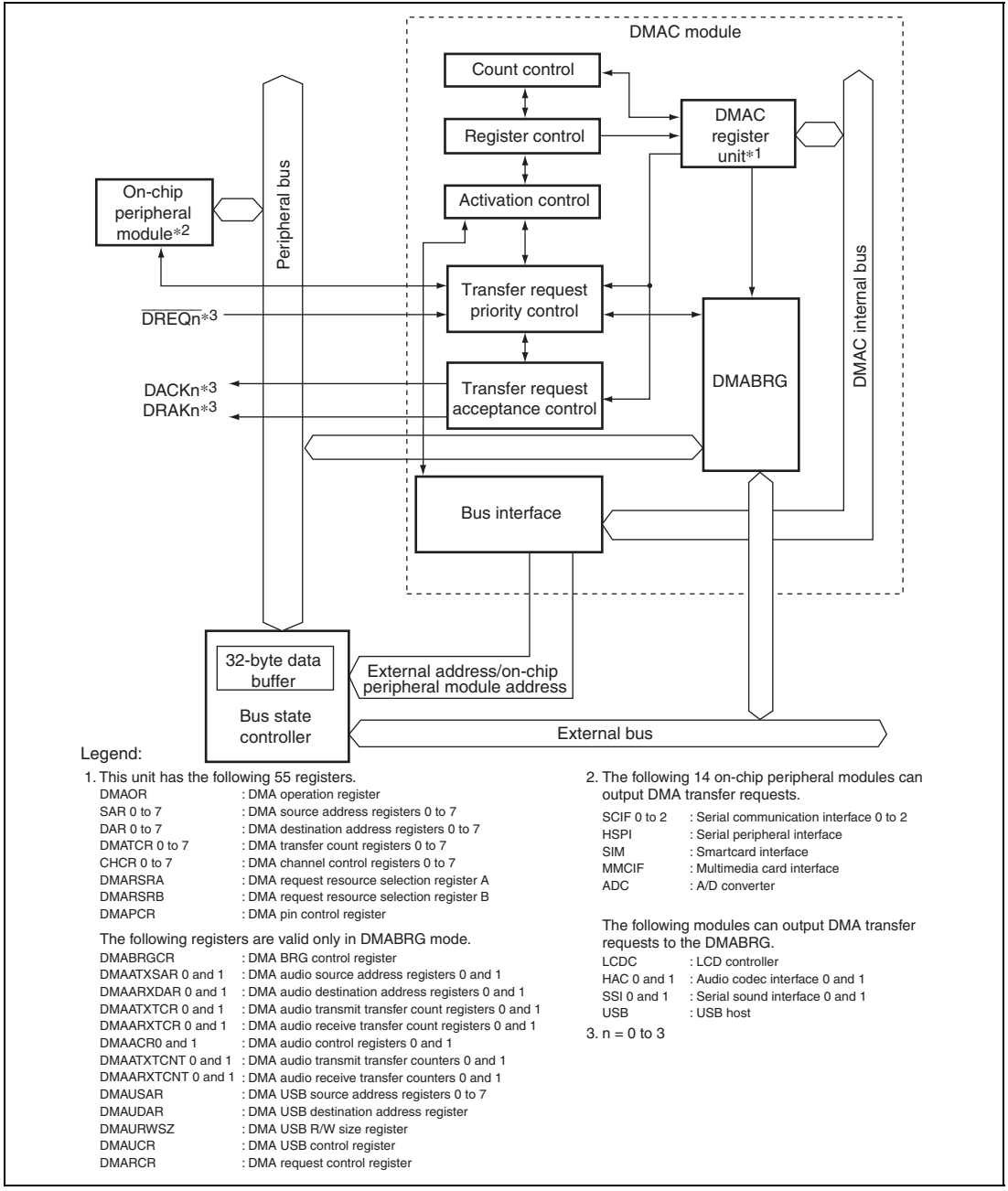
11.1 Features

The DMAC has the following features.

- Number of channels: Eight channels
- Address space: Physical address space
- Selection of data length: 8-bit, 16-bit, 32-bit, 64-bit, or 32-byte transfer data length
- Maximum number of transfers: 16 M (16,777,216) transfers
- Selection of DMA mode: External request 2-channel mode or DMABRG mode
- Selection of address mode: Single address mode*[1] or dual address mode
- Selection of transfer requests: External request*[2], requests from on-chip peripheral modules*[3], or auto-request
- Selection of bus mode: Cycle steal mode or burst mode
- Selection of priority order: Fixed priority mode or round robin mode
- Channel functions: Different transfer modes (address mode, bus mode, and transfer requests) can be set for each channel.
- Interrupt request: Interrupt request can be sent to the CPU on completion of data transfer.

Notes: 1. In DMABRG mode, only synchronous DRAM can be specified.

2. External request 2-channel mode: DREQ0 (corresponds to channel 0) and DREQ1 (corresponds to channel 1)
   DMABRG mode: DREQ0 to DREQ3 (can be set for all channels)
3. External request 2-channel mode: Transfer requests cannot be accepted from the LCDC, HAC, SSI, and USB.
   DMABRG mode: Transfer requests can be accepted from all on-chip peripheral modules with the DMA transfer request function. (Note that transfer requests from the LCDC, HAC, SSI, and USB can only be accepted in channel 0.)



Legend:

1. This unit has the following 55 registers.

DMAOR	: DMA operation register
SAR 0 to 7	: DMA source address registers 0 to 7
DAR 0 to 7	: DMA destination address registers 0 to 7
DMATCR 0 to 7	: DMA transfer count registers 0 to 7
CHCR 0 to 7	: DMA channel control registers 0 to 7
DMARSRA	: DMA request resource selection register A
DMARSRB	: DMA request resource selection register B
DMAPCR	: DMA pin control register

The following registers are valid only in DMABRG mode.

DMABRGCR	: DMA BRG control register
DMAATXSAR 0 and 1	: DMA audio source address registers 0 and 1
DMAARXDAR 0 and 1	: DMA audio destination address registers 0 and 1
DMAATXTCR 0 and 1	: DMA audio transmit transfer count registers 0 and 1
DMAARXTCR 0 and 1	: DMA audio receive transfer count registers 0 and 1
DMAACR0 and 1	: DMA audio control registers 0 and 1
DMAATXTCNT 0 and 1	: DMA audio transmit transfer counters 0 and 1
DMAARXTCNT 0 and 1	: DMA audio receive transfer counters 0 and 1
DMAUSAR	: DMA USB source address registers 0 to 7
DMAUDAR	: DMA USB destination address register
DMAURWSZ	: DMA USB R/W size register
DMAUCR	: DMA USB control register
DMARCR	: DMA request control register

2. The following 14 on-chip peripheral modules can output DMA transfer requests.

SCIF 0 to 2	: Serial communication interface 0 to 2
HSPI	: Serial peripheral interface
SIM	: Smartcard interface
MMCIF	: Multimedia card interface
ADC	: A/D converter

The following modules can output DMA transfer requests to the DMABRG.

LCDC	: LCD controller
HAC 0 and 1	: Audio codec interface 0 and 1
SSI 0 and 1	: Serial sound interface 0 and 1
USB	: USB host

3. n = 0 to 3

Figure 11.1 DMAC Block Diagram

For data transfer in DMABRG mode between synchronous DRAM and the LCDC, HAC, SSI, or USB, the DMABRG performs a high-speed data transfer via the DMABRG internal FIFO (32-bit 16-stage) using DMAC channel 0. The DMABRG transfers a maximum of 32-byte data in a single transfer.

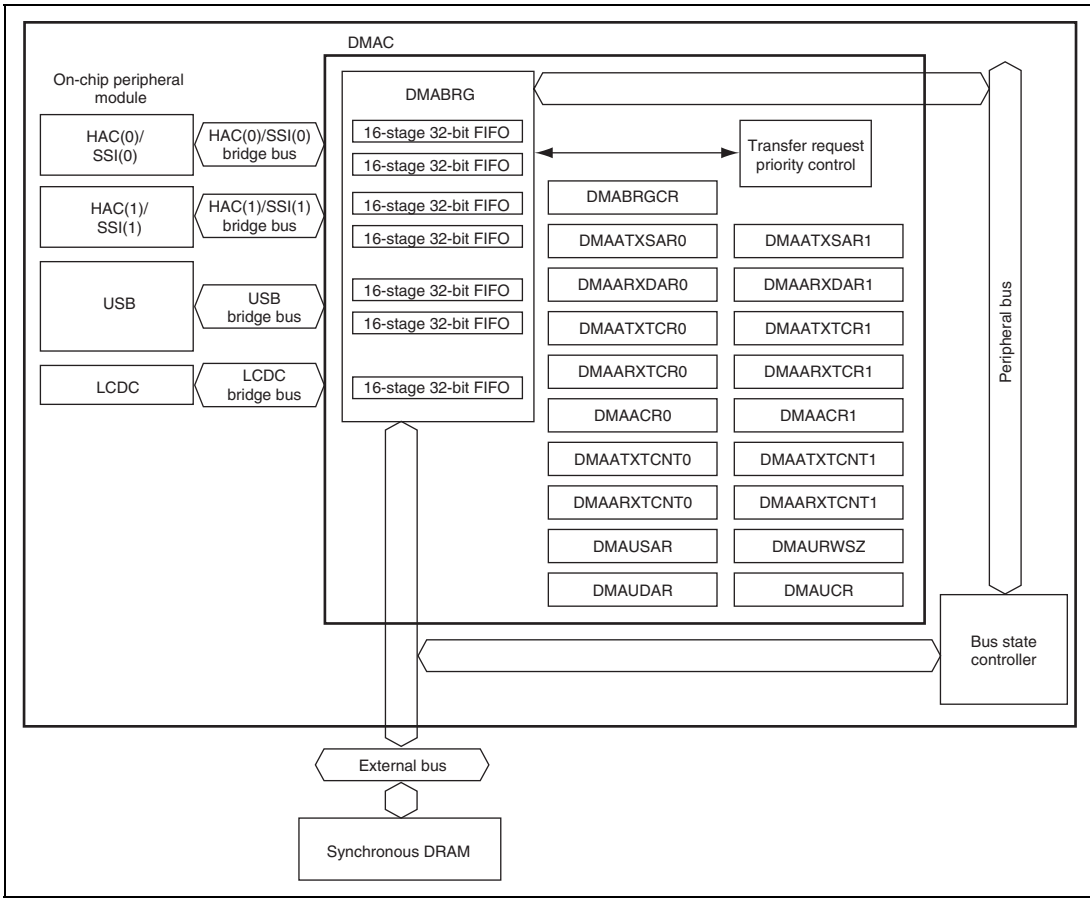


Figure 11.2 DMABRG Block Diagram

Table 11.1 shows the DMAC pin configuration.

Table 11.1 Pin Configuration

Pin Name		Abbreviation	I/O	Function
DMA transfer request	$\overline{\text{DREQ0}}$	$\overline{\text{DREQ0}}$	Input	DMA transfer request input from external device
$\overline{\text{DREQ}}$ acceptance confirmation	DRAK0	DRAK0	Output	Notifies acceptance of DMA transfer request and start of execution to external device which has output $\overline{\text{DREQ0}}$*1
DMA transfer end notification	DACK0	DACK0	Output	Strobe output to external device which has output $\overline{\text{DREQ0}}$, regarding DMA transfer request
DMA transfer request	$\overline{\text{DREQ1}}$	$\overline{\text{DREQ1}}$	Input	DMA transfer request input from external device
$\overline{\text{DREQ}}$ acceptance confirmation	DRAK1	DRAK1	Output	Notifies acceptance of DMA transfer request and start of execution to external device which has output $\overline{\text{DREQ1}}$*1
DMA transfer end notification	DACK1	DACK1	Output	Strobe output to external device which has output $\overline{\text{DREQ1}}$, regarding DMA transfer request
DMA transfer request	$\overline{\text{DREQ2}}$	$\overline{\text{DREQ2}}$	Input	DMA transfer request input from external device
$\overline{\text{DREQ}}$ acceptance confirmation	DRAK2/ DACK2	DRAK2	Output	Notifies acceptance of DMA transfer request to external device which has output $\overline{\text{DREQ2}}$*2
DMA transfer end notification		DACK2	Output	Strobe output to external device which has output $\overline{\text{DREQ2}}$, regarding DMA transfer request*2
DMA transfer request	$\overline{\text{DREQ3}}$	$\overline{\text{DREQ3}}$	Input	DMA transfer request input from external device

DREQ acceptance confirmation	DRAK3/ DACK3	DRAK3	Output	Notifies acceptance of DMA transfer request to external device which has output $\overline{\text{DREQ3}}$*[3]
DMA transfer end notification		DACK3	Output	Strobe output to external device which has output $\overline{\text{DREQ3}}$, regarding DMA transfer request*[3]

Notes: 1. Pin DRAK0 or DRAK1 indicates the start of execution only in external request 2-channel mode.
2. Pins DRAK2 and DACK2 are multiplexed.
3. Pins DRAK3 and DACK3 are multiplexed.

11.3 Register Descriptions

The DMAC has the following registers. For details of register addresses and register states during each process, see section 32, List of Registers. For details regarding the DMA pin control register (DMAPCR), see section 24.2.34, DMA Pin Control Register (DMAPCR), in section 24, Pin Function Controller (PFC). In later descriptions, channel numbers are not explicitly mentioned.

Table 11.2 Register Configuration (1)

Ch.	Register Name	Abbrev.	R/W	P4 Address	Area 7 Address	Size	Sync Clock
0	DMA source address register 0	SAR0	R/W	H'FFA0 0000	H'1FA0 0000	32	Bck
	DMA destination address register 0	DAR0	R/W	H'FFA0 0004	H'1FA0 0004	32	Bck
	DMA transfer count register 0	DMATCR0	R/W	H'FFA0 0008	H'1FA0 0008	32	Bck
	DMA channel control register 0	CHCR0	R/W	H'FFA0 000C	H'1FA0 000C	32	Bck
1	DMA source address register 1	SAR1	R/W	H'FFA0 0010	H'1FA0 0010	32	Bck
	DMA destination address register 1	DAR1	R/W	H'FFA0 0014	H'1FA0 0014	32	Bck
	DMA transfer count register 1	DMATCR1	R/W	H'FFA0 0018	H'1FA0 0018	32	Bck
	DMA channel control register 1	CHCR1	R/W	H'FFA0 001C	H'1FA0 001C	32	Bck
2	DMA source address register 2	SAR2	R/W	H'FFA0 0020	H'1FA0 0020	32	Bck
	DMA destination address register 2	DAR2	R/W	H'FFA0 0024	H'1FA0 0024	32	Bck
	DMA transfer count register 2	DMATCR2	R/W	H'FFA0 0028	H'1FA0 0028	32	Bck
	DMA channel control register 2	CHCR2	R/W	H'FFA0 002C	H'1FA0 002C	32	Bck

3	DMA source address register 3	SAR3	R/W	H'FFA0 0030	H'1FA0 0030	32	Bck
	DMA destination address register 3	DAR3	R/W	H'FFA0 0034	H'1FA0 0034	32	Bck
	DMA transfer count register 3	DMATCR3	R/W	H'FFA0 0038	H'1FA0 0038	32	Bck
	DMA channel control register 3	CHCR3	R/W	H'FFA0 003C	H'1FA0 003C	32	Bck
4	DMA source address register 4	SAR4	R/W	H'FFA0 0050	H'1FA0 0050	32	Bck
	DMA destination address register 4	DAR4	R/W	H'FFA0 0054	H'1FA0 0054	32	Bck
	DMA transfer count register 4	DMATCR4	R/W	H'FFA0 0058	H'1FA0 0058	32	Bck
	DMA channel control register 4	CHCR4	R/W	H'FFA0 005C	H'1FA0 005C	32	Bck
5	DMA source address register 5	SAR5	R/W	H'FFA0 0060	H'1FA0 0060	32	Bck
	DMA destination address register 5	DAR5	R/W	H'FFA0 0064	H'1FA0 0064	32	Bck
	DMA transfer count register 5	DMATCR5	R/W	H'FFA0 0068	H'1FA0 0068	32	Bck
	DMA channel control register 5	CHCR5	R/W	H'FFA0 006C	H'1FA0 006C	32	Bck
6	DMA source address register 6	SAR6	R/W	H'FFA0 0070	H'1FA0 0070	32	Bck
	DMA destination address register 6	DAR6	R/W	H'FFA0 0074	H'1FA0 0074	32	Bck
	DMA transfer count register 6	DMATCR6	R/W	H'FFA0 0078	H'1FA0 0078	32	Bck
	DMA channel control register 6	CHCR6	R/W	H'FFA0 007C	H'1FA0 007C	32	Bck
7	DMA source address register 7	SAR7	R/W	H'FFA0 0080	H'1FA0 0080	32	Bck
	DMA destination address register 7	DAR7	R/W	H'FFA0 0084	H'1FA0 0084	32	Bck
	DMA transfer count register 7	DMATCR7	R/W	H'FFA0 0088	H'1FA0 0088	32	Bck
	DMA channel control register 7	CHCR7	R/W	H'FFA0 008C	H'1FA0 008C	32	Bck
Common	DMA operation register	DMAOR	R/W	H'FFA0 0040	H'1FA0 0040	32	Bck
	DMA request resource selection register A	DMARSRA	R/W	H'FE09 0000	H'1E09 0000	32	Pck
	DMA request resource selection register B	DMARSRB	R/W	H'FE09 0004	H'1E09 0004	32	Pck
	DMA request control register	DMARCR	R/W	H'FE09 0008	H'1E09 0008	32	Pck
DMABRG	DMA BRG control register	DMABRGCR	R/W	H'FE3C 0000	H'1E3C 0000	32	Pck
	DMA audio source address register 0	DMAATXSAR0	R/W	H'FE3C 0040	H'1E3C 0040	32	Pck
	DMA audio destination address register 0	DMAARXDAR0	R/W	H'FE3C 0044	H'1E3C 0044	32	Pck
	DMA audio transmit transfer count register 0	DMAATXTCR0	R/W	H'FE3C 0048	H'1E3C 0048	32	Pck

DMABRG	DMA audio receive transfer count register 0	DMAARXTCR0	R/W	H'FE3C 004C	H'1E3C 004C	32	Pck
	DMA audio control register 0	DMAACR0	R/W	H'FE3C 0050	H'1E3C 0050	32	Pck
	DMA audio transmit transfer counter 0	DMAATXTCNT0	R	H'FE3C 0054	H'1E3C 0054	32	Pck
	DMA audio receive transfer counter 0	DMAARXTCNT0	R	H'FE3C 0058	H'1E3C 0058	32	Pck
	DMA audio source address register 1	DMAATXSAR1	R/W	H'FE3C 0060	H'1E3C 0060	32	Pck
	DMA audio destination address register 1	DMAARXDAR1	R/W	H'FE3C 0064	H'1E3C 0064	32	Pck
	DMA audio transmit transfer count register 1	DMAATXTCR1	R/W	H'FE3C 0068	H'1E3C 0068	32	Pck
	DMA audio receive transfer count register 1	DMAARXTCR1	R/W	H'FE3C 006C	H'1E3C 006C	32	Pck
	DMA audio control register 1	DMAACR1	R/W	H'FE3C 0070	H'1E3C 0070	32	Pck
	DMA audio transmit transfer counter 1	DMAATXTCNT1	R	H'FE3C 0074	H'1E3C 0074	32	Pck
	DMA audio receive transfer counter 1	DMAARXTCNT1	R	H'FE3C 0078	H'1E3C 0078	32	Pck
	DAM USB source address register	DMAUSAR	R/W	H'FE3C 0080	H'1E3C 0080	32	Pck
	DMA USB destination register	DMAUDAR	R/W	H'FE3C 0084	H'1E3C 0084	32	Pck
	DMA USB R/W size register	DMAURWSZ	R/W	H'FE3C 0088	H'1E3C 0088	32	Pck
	DMA USB control register	DMAUCR	R/W	H'FE3C 008C	H'1E3C 008C	32	Pck

Ch.	Register Name	Abbrev.	Power-on Reset by $\overline{\text{RESET}}$ Pin/WDT/ H-UDI	Manual Reset by $\overline{\text{RESET}}$ Pin/WDT/ Multiple Exception	Sleep by Sleep Instruction/ Deep Sleep	Standby by Hardware	Standby by Software/ Each Module
0	DMA source address register 0	SAR0	Undefined	Undefined	Retained	*	Retained
	DMA destination address register 0	DAR0	Undefined	Undefined	Retained		Retained
	DMA transfer count register 0	DMATCR0	Undefined	Undefined	Retained		Retained
	DMA channel control register 0	CHCR0	H'0000 0000	H'0000 0000	Retained		Retained
1	DMA source address register 1	SAR1	Undefined	Undefined	Retained		Retained
	DMA destination address register 1	DAR1	Undefined	Undefined	Retained		Retained
	DMA transfer count register 1	DMATCR1	Undefined	Undefined	Retained		Retained
	DMA channel control register 1	CHCR1	H'0000 0000	H'0000 0000	Retained		Retained
2	DMA source address register 2	SAR2	Undefined	Undefined	Retained		Retained
	DMA destination address register 2	DAR2	Undefined	Undefined	Retained		Retained
	DMA transfer count register 2	DMATCR2	Undefined	Undefined	Retained		Retained
	DMA channel control register 2	CHCR2	H'0000 0000	H'0000 0000	Retained		Retained
3	DMA source address register 3	SAR3	Undefined	Undefined	Retained		Retained
	DMA destination address register 3	DAR3	Undefined	Undefined	Retained		Retained
	DMA transfer count register 3	DMATCR3	Undefined	Undefined	Retained		Retained
	DMA channel control register 3	CHCR3	H'0000 0000	H'0000 0000	Retained		Retained
4	DMA source address register 4	SAR4	Undefined	Undefined	Retained		Retained
	DMA destination address register 4	DAR4	Undefined	Undefined	Retained		Retained
	DMA transfer count register 4	DMATCR4	Undefined	Undefined	Retained		Retained
	DMA channel control register 4	CHCR4	H'0000 0000	H'0000 0000	Retained		Retained
5	DMA source address register 5	SAR5	Undefined	Undefined	Retained		Retained
	DMA destination address register 5	DAR5	Undefined	Undefined	Retained		Retained
	DMA transfer count register 5	DMATCR5	Undefined	Undefined	Retained		Retained
	DMA channel control register 5	CHCR5	H'0000 0000	H'0000 0000	Retained		Retained
6	DMA source address register 6	SAR6	Undefined	Undefined	Retained		Retained
	DMA destination address register 6	DAR6	Undefined	Undefined	Retained		Retained
	DMA transfer count register 6	DMATCR6	Undefined	Undefined	Retained		Retained
	DMA channel control register 6	CHCR6	H'0000 0000	H'0000 0000	Retained		Retained

Ch.	Register Name	Abbrev.	Reset by RESET Pin/WDT/ H-UDI	Reset by RESET Pin/WDT/ Multiple Exception	Sleep by Sleep Instruction/ Deep Sleep	Module Stop by Hardware	Module Stop by Software/ Each Module
7	DMA source address register 7	SAR7	Undefined	Undefined	Retained	*	Retained
	DMA destination address register 7	DAR7	Undefined	Undefined	Retained		Retained
	DMA transfer count register 7	DMATCR7	Undefined	Undefined	Retained		Retained
	DMA channel control register 7	CHCR7	H'0000 0000	H'0000 0000	Retained		Retained
Common	DMA operation register	DMAOR	H'0000 0000	H'0000 0000	Retained		Retained
	DMA request resource selection register A	DMARSRA	H'0000 0000	H'0000 0000	Retained		Retained
	DMA request resource selection register B	DMARSRB	H'0000 0000	H'0000 0000	Retained		Retained
	DMA request control register	DMARCR	H'0000 0000	H'0000 0000	Retained		Retained
DMABRG	DMA BRG control register	DMABRG CR	H'0000 0000	H'0000 0000	Retained		Retained
	DMA audio source address register 0	DMAATX SAR0	Undefined	Undefined	Retained		Retained
	DMA audio destination address register 0	DMAARX DAR0	Undefined	Undefined	Retained		Retained
	DMA audio transmit transfer count register 0	DMAATX TCR0	Undefined	Undefined	Retained		Retained
	DMA audio receive transfer count register 0	DMAARX TCR0	Undefined	Undefined	Retained		Retained
	DMA audio control register 0	DMAACR0	H'0000 0000	H'0000 0000	Retained		Retained
	DMA audio transmit transfer counter 0	DMAATX TCNT0	Undefined	Undefined	Retained		Retained
	DMA audio receive transfer counter 0	DMAARX TCNT0	Undefined	Undefined	Retained		Retained
	DMA audio source address register 1	DMAATX SAR1	Undefined	Undefined	Retained		Retained
	DMA audio destination address register 1	DMAARX DAR1	Undefined	Undefined	Retained		Retained
	DMA audio transmit transfer count register 1	DMAATX TCR1	Undefined	Undefined	Retained		Retained
	DMA audio receive transfer count register 1	DMAARX TCR1	Undefined	Undefined	Retained		Retained

Ch.	Register Name	Abbrev.	Reset by RESET Pin/WDT/ H-UDI	Reset by RESET Pin/WDT/ Multiple Exception	Sleep by Sleep Instruction/ Deep Sleep	by Hardware	by Software/ Each Module
DMABRG	DMA audio control register 1	DMAACR1	H'0000 0000	H'0000 0000	Retained	*	Retained
	DMA audio transmit transfer counter 1	DMAATX TCNT1	Undefined	Undefined	Retained		Retained
	DMA audio receive transfer counter 1	DMAARX TCNT1	Undefined	Undefined	Retained		Retained
	DAM USB source address register	DMAUSAR	H'0000 0000	H'0000 0000	Retained		Retained
	DMA USB destination register	DMAUDAR	H'0000 0000	H'0000 0000	Retained		Retained
	DMA USB R/W size register	DMAURWSZ	H'0000 0000	H'0000 0000	Retained		Retained
	DMA USB control register	DMAUCR	H'0000 0000	H'0000 0000	Retained		Retained

Note: * After exiting hardware standby mode, this LSI enters the power-on reset state by the RESET pin.

11.3.1 DMA Source Address Register (SAR)

SAR is a 32-bit readable/writable register that specifies the source address of a DMA transfer. During a DMA transfer, they indicate the next source address. In single address mode, the SAR value is ignored when an external device with DACK has been specified as the transfer source.

A 16-bit, 32-bit, 64-bit, or 32-byte boundary address should be specified when performing a 16-bit, 32-bit, 64-bit, or 32-byte data transfer, respectively. If a different address is specified, an address error will be detected and the DMAC will halt.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
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

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
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

Notes: 1. Make the setting of bit 0, bits 1 and 0, bits 2 to 0, or bits 4 to 0 to match the boundary when specifying a 16-bit, 32-bit, 64-bit, or 32-byte boundary address, respectively. If an address is specified regardless of the boundary, an address error will be detected and the DMAC stops operation on all channels (AE (address error flag) bit in DMAOR is 1). The DMAC will also detect an address error and stop operation if an area 7 address

2. An external address is 29 bits long. Bits 31 to 29 in both SAR and DAR are not used in DMA transfers. Therefore, clearing bits 31 to 29 to 0 in both SAR and DAR is recommended.

11.3.2 DMA Destination Address Register (DAR)

DAR is a 32-bit readable/writable register that specifies the destination address of a DMA transfer. During a DMA transfer, they indicate the next destination address. In single address mode, the DAR value is ignored when a device with DACK has been specified as the transfer destination.

A 16-bit, 32-bit, 64-bit, or 32-byte boundary address should be specified when performing a 16-bit, 32-bit, 64-bit, or 32-byte data transfer, respectively. If a different address is specified, an address error will be detected and the DMAC will halt.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
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

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
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

Notes: 1. Make the setting of bit 0, bits 1 and 0, bits 2 to 0, or bits 4 to 0 to match the boundary when specifying a 16-bit, 32-bit, 64-bit, or 32-byte boundary address, respectively. If an address is specified regardless of the boundary, an address error will be detected and the DMAC stops operation on all channels (AE (address error flag) bit in DMAOR is 1). The DMAC will also detect an address error and stop operation if an area 7 address is specified for a data transfer via the external bus or if an address for an on-chip peripheral module that does not exist is specified.

2. An external address is 29 bits long. Bits 31 to 29 in both SAR and DAR are not used in DMA transfers. Therefore, clearing bits 31 to 29 to 0 in both SAR and DAR is recommended.

DMATCR is a 32-bit readable/writable register that specifies the transfer count for the corresponding channel. Specifying H'0000 0001 gives a transfer count of 1, while H'0000 0000 gives the maximum setting (16,777,216). During DMAC operation, the remaining number of transfers is shown.

The upper eight bits of DMATCR are reserved. They are always read as 0 and the write value should always be 0.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial value:	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

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
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

CHCR is a 32-bit readable/writable register that specifies the operating mode and transfer method for each channel. Bits 31 to 28 and 27 to 24 are only valid when the source and destination addresses are in the CS5 or CS6 space and the relevant space has been specified as a PCMCIA interface space. In other cases, these bits should be cleared to 0.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SSA2	SSA1	SSA0	STC	DSA2	DSA1	DSA0	DTC	-	-	-	-	DS	RL	AM	AL
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	R	R	R	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
	DM1	DM0	SM1	SM0	RS3	RS2	RS1	RS0	TM	TS2	TS1	TS0	CHSET	IE	TE	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

Bit	Bit Name	Initial Value	R/W	Description
31	SSA2	0	R/W	Source Address Space Attribute Specification
30	SSA1	0	R/W	These bits specify the space attribute for the source address when accessing a PCMCIA interface area. These bits are only valid in the case of page mapping to PCMCIA connected to areas 5 and 6.
29	SSA0	0	R/W	000: Reserved in PCMCIA access 001: Dynamic bus sizing I/O space 010: 8-bit I/O space 011: 16-bit I/O space 100: 8-bit common memory space 101: 16-bit common memory space 110: 8-bit attribute memory space 111: 16-bit attribute memory space

28	STC	0	R/W	Source Address Wait Control Select Specifies the CS5 or CS6 space wait cycle control for the source address when accessing a PCMCIA interface area. 0: CS5 space wait cycle selection Settings of bits A5W2 to A5W0 in WCR2 and bits A5PCW1 and A5PCW0, A5TED2 to A5TED0, and A5TEH2 to A5TEH0 in PCR are selected 1: CS6 space wait cycle selection Settings of bits A6W2 to A6W0 in WCR2 and bits A6PCW1 and A6PCW0, A6TED2 to A6TED0, and A6TEH2 to A6TEH0 in PCR are selected
27 26 25	DSA2 DSA1 DSA0	0 0 0	R/W R/W R/W	Destination Address Space Attribute Specification These bits specify the space attribute for the destination address when accessing a PCMCIA interface area. These bits are only valid in the case of page mapping to PCMCIA connected to areas 5 and 6. 000: Reserved in PCMCIA access 001: Dynamic bus sizing I/O space 010: 8-bit I/O space 011: 16-bit I/O space 100: 8-bit common memory space 101: 16-bit common memory space 110: 8-bit attribute memory space 111: 16-bit attribute memory space

24	DTC	0	R/W	Destination Address Wait Control Select Specifies the CS5 or CS6 space wait cycle control for the destination address when accessing a PCMCIA interface area. 0: CS5 space wait cycle selection Settings of bits A5W2 to A5W0 in WCR2 and bits A5PCW1 and A5PCW0, A5TED2 to A5TED0, and A5TEH2 to A5TEH0 in PCR are selected 1: CS6 space wait cycle selection Settings of bits A6W2 to A6W0 in WCR2 and bits A6PCW1 and A6PCW0, A6TED2 to A6TED0, and A6TEH2 to A6TEH0 in PCR are selected
23 to 20	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
19	DS	0	R/W	$\overline{\text{DREQ}}$ Select Specifies either low level detection or falling edge detection as the sampling method for the $\overline{\text{DREQ}}$ pin. In external request 2-channel mode, this bit is valid only in CHCR0 and CHCR1. In DMABRG mode, this bit is valid in CHCR0 to CHCR7 and each of bits DS3 to DS0 in DMARCR should be specified to the same as this bit. 0: Low level detection 1: Falling edge detection Level detection burst mode when TM = 1 and DS = 0 Edge detection burst mode when TM = 1 and DS = 1
18	RL	0	R/W	Request Check Level Selects whether the DRAK signal that notifies an external device of the acceptance of $\overline{\text{DREQ}}$ is an active-high or active-low output. In external request 2-channel mode, this bit is valid only in CHCR0 and CHCR1. In DMABRG mode, this bit is invalid and the DRAK polarity is specified by bits RL3 to RL0 in DMARCR. 0: DRAK is an active-high output 1: DRAK is an active-low output

17	AM	0	R/W	Acknowledge Mode In dual address mode, selects whether DACK is output in the data read cycle or write cycle. In single address mode, DACK is always output regardless of the setting of this bit. In external request 2-channel mode, this bit is valid only in CHCR0 and CHCR1. In DMABRG mode, this bit is valid in CHCR0 to CHCR7. 0: DACK is output in read cycles 1: DACK is output in write cycles
16	AL	0	R/W	Acknowledge Level Specifies the DACK signal as active-high or active-low. In external request 2-channel mode, this bit is valid only in CHCR0 and CHCR1. In DMABRG mode, this bit is invalid and the DACK polarity is specified by bits AL3 to AL0 in DMARCR. 0: Active-high output 1: Active-low output
15 14	DM1 DM0	0 0	R/W R/W	Destination Address Mode 1 and 0 These bits specify incrementing/decrementing of the DMA transfer destination address. The specification of these bits is ignored when data is transferred from external memory to an external device in single address mode. 00: Destination address fixed 01: Destination address incremented (+1 in 8-bit transfer, +2 in 16-bit transfer, +4 in 32-bit transfer, +8 in 64-bit transfer, +32 in 32-byte burst transfer) 10: Destination address decremented (–1 in 8-bit transfer, –2 in 16-bit transfer, –4 in 32-bit transfer, –8 in 64-bit transfer, –32 in 32-byte burst transfer) 11: Setting prohibited

13	SM1	0	R/W	Source Address Mode 1 and 0
12	SM0	0	R/W	These bits specify incrementing/decrementing of the DMA transfer source address. The specification of these bits is ignored when data is transferred from an external device to external memory in single address mode. 00: Source address fixed 01: Source address incremented (+1 in 8-bit transfer, +2 in 16-bit transfer, +4 in 32-bit transfer, +8 in 64-bit transfer, +32 in 32-byte burst transfer) 10: Source address decremented (–1 in 8-bit transfer, –2 in 16-bit transfer, –4 in 32-bit transfer, –8 in 64-bit transfer, –32 in 32-byte burst transfer) 11: Setting prohibited
11	RS3	0	R/W	Resource Select 3 to 0 These bits specify the transfer request source. Setting bits DMS1 and DMS0 in DMAOR specifies the transfer request source in external request 2-channel mode or DMABRG mode. See tables 11.3 and 11.4 for settings of bits RS3 to RS0.
10	RS2	0	R/W	
9	RS1	0	R/W	
8	RS0	0	R/W	
7	TM	0	R/W	Transmit Mode Specifies the bus mode for transfer. 0: Cycle steal mode 1: Burst mode
6	TS2	0	R/W	Transmit Size 2 to 0 These bits specify the transfer data size (access size). 000: Quadword size (64-bit) specification 001: Byte size (8-bit) specification 010: Word size (16-bit) specification 011: Longword size (32-bit) specification 100: 32-byte block transfer specification Other than above: Setting prohibited
5	TS1	0	R/W	
4	TS0	0	R/W	

3	CHSET	0	R/W	Channel Setting Since the internal state of the acceptance unit for the corresponding channel external and DMABRG requests are cleared when 1 is written to this bit in DMABRG mode, write 1 to this bit when setting up the corresponding channel. Note, however, that this bit always reads out as 0. Note: This operation is invalid in external request 2-channel mode.
2	IE	0	R/W	Interrupt Enable When this bit is set to 1, an interrupt request (DMTE) is generated after completing a number of data transfers specified in DMATCR (when TE = 1). When a DMABRG request DMA transfer is executed in DMABRG mode, the DMTE0 signal cannot be generated. 0: Interrupt request is not generated after completing a number of transfers specified in DMATCR 1: Interrupt request is generated after completing a number of transfers specified in DMATCR
1	TE	0	R/W	Transfer End This bit is set to 1 after the number of transfers specified in DMATCR. If the IE bit is set to 1 at this time, an interrupt request (DMTE) is generated. If data transfer ends before this bit is set to 1 due to an NMI interrupt, address error, or clearing of the DE bit or the DME bit in DMAOR, etc., this bit is not set to 1. When this bit is 1, the transfer enabled state is not entered even if the DE bit is set to 1. 0: Number of transfers specified in DMATCR not completed [Clearing condition] When 0 is written to TE after reading TE = 1 1: Number of transfers specified in DMATCR completed

0	DE	0	R/W	DMAC Enable Enables operation of the corresponding channel. When auto-request is specified by bits RS3 to RS0, setting this bit to 1 starts the transfer. When an external request or on-chip peripheral module request is generated, a transfer request after setting this bit to 1 starts the transfer. During a transfer, clearing this bit to 0 stops the transfer. The transfer enable state is not entered by setting this bit to 1 if the TE bit is set to 1, the DME bit in DMAOR is cleared to 0, or the NMIF or AE bit in DMAOR is set to 1. 0: Operation of corresponding channel is disabled 1: Operation of corresponding channel is enabled

Bit 11: RS3	Bit 10: RS2	Bit 9: RS1	Bit 8: RS0	Description
0	0	0	0	External request*[1] Dual address mode External address space → external address space
			1	Setting prohibited
		1	0	External request*[1] Single address mode External address space → external device
			1	External request*[1] Single address mode External device → external address space
	1	0	0	Auto-request (external address space → external address space)
			1	Auto-request (external address space → on-chip peripheral module)
		1	0	Auto-request (on-chip peripheral module → external address space)
			1	On-chip peripheral module request*[2]*[3] External address space → on-chip peripheral module
1	0	0	0	Setting prohibited
			1	Setting prohibited
		1	0	Setting prohibited
			1	Setting prohibited
	1	0	0	TMU channel 2 (input capture interrupt) External address space → external address space
			1	TMU channel 2 (input capture interrupt) External address space → on-chip peripheral module
		1	0	TMU channel 2 (input capture interrupt) On-chip peripheral module → external address space
			1	On-chip peripheral module request*[2]*[3] On-chip peripheral module → external address space

Notes: 1. External request specifications are valid only for channels 0 and 1. $\overline{\text{DREQ0}}$ and $\overline{\text{DREQ1}}$ correspond to channel 0 and channel 1, respectively.
2. DMARSRA and DMARSRB values should be specified in addition to setting this bit.
3. On-chip peripheral modules except for LCDC, HAC, SSI, USB, and TMU

Bit 11: RS3	Bit 10: RS2	Bit 9: RS1	Bit 8: RS0	Description
0	0	0	0	External request*1*2 Dual address mode External address space → external address space
			1	Setting prohibited
		1	0	External request*1*2, DMABRG request*2*3 Single address mode*4, External address space → external device
			1	External request*1*2, DMABRG request*2*3 Single address mode*4, External device → external address space
	1	0	0	Auto-request (external address space → external address space)
			1	Auto-request (external address space → on-chip peripheral module)
		1	0	Auto-request (on-chip peripheral module → external address space)
			1	On-chip peripheral module request*2*5 External address space → on-chip peripheral module
1	0	0	0	Setting prohibited
			1	Setting prohibited
		1	0	Setting prohibited
			1	Setting prohibited
	1	0	0	TMU channel 2 (input capture interrupt) External address space → external address space
			1	TMU channel 2 (input capture interrupt) External address space → on-chip peripheral module
		1	0	TMU channel 2 (input capture interrupt) On-chip peripheral module → external address space
			1	On-chip peripheral module request*2*5 On-chip peripheral module → external address space

Notes: 1. External requests can be accepted in all channels. $\overline{\text{DREQ0}}$ to $\overline{\text{DREQ3}}$ can be used. Note that the $\overline{\text{DREQ}}$ pin number and channel number do not match.

2. DMARSRA and DMARSRB values should be specified in addition to setting this bit.

3. DMABRG requests can only be accepted in channel 0. A DMABRG request is a transfer request from the LCDC, HAC(0)/(1), SSI(0)/(1), and USB. This bit is automatically set when a DMABRG request is issued.

4. Only single address mode for synchronous DRAM can be set.

5. On-chip peripheral modules except for LCDC, HAC, SSI, USB, and TMU.

DMAOR is a 32-bit readable/writable register that specifies the DMA mode and channel priorities, and enables or disables DMA transfer.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
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
	DMS1	DMS0	-	-	-	-	PR1	PR0	-	-	-	-	-	AE	NMIF	DME
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	R	R	R	R/W	R/W	R	R	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
15	DMS1	0	R/W	DMA Mode Select 1 and 0
14	DMS0	0	R/W	These bits select the transfer mode. Set these bits when DMAOR.DME = 0. 00: External-request 2-channel mode 11: DMABRG mode* Other than above: Setting prohibited Note: * Make this setting when CHCR0 has its initial value of H'0000 0000.
13 to 10	—	All 0	R	Reserved These bits are always read as 0.The write value should always be 0.
9	PR1	0	R/W	Priority Mode 1 and 0
8	PR0	0	R/W	These bits determine the order of priority for channel execution when transfer requests are made for multiple channels simultaneously. 00: CH0 > CH1 > CH2 > CH3 > CH4 > CH5 > CH6 > CH7 01: CH0 > CH2 > CH3 > CH4 > CH5 > CH6 > CH7 > CH1 10: CH2 > CH0 > CH1 > CH3 > CH4 > CH5 > CH6 > CH7 11: Round robin mode

7 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2	AE	0	R/W	Address Error Flag Indicates that an address error has occurred during DMA transfer. Setting this bit during data transfer will suspend transfers on all channels and generate an interrupt request (DMAE). The CPU cannot write 1 to this bit. Write AE=0 after reading AE=1 to clear this bit. 0: No address error, DMA transfer enabled [Clearing condition] When 0 is written to the AE bit after reading AE = 1 1: Address error, DMA transfer disabled [Setting condition] When an address error is caused by the DMAC
1	NMIF	0	R/W	NMI Flag Indicates that NMI has been input. It is possible to set this bit regardless of whether or not the DMAC is operating. Setting this bit during data transfer will suspend transfers on all channels. The CPU cannot write 1 to this bit. Write NMIF=0 after reading NMIF=1 to clear this bit. 0: No NMI input, DMA transfer enabled [Clearing condition] When 0 is written to NMIF after reading NMIF = 1 1: NMI input, DMA transfer disabled [Setting condition] When an NMI interrupt is generated
0	DME	0	R/W	DMAC Master Enable Enables activation of the entire DMAC. Setting the DME bit and the DE bit in CHCR for the corresponding channel to 1 will enable that channel for translfer. Clearing this bit during data transfer will suspend transfers on all channels. Even if the DME bit has been set to 1, transfer is not enabled when TE is 1 or DE is 0 in CHCR, or when the NMIF or AE bit in DMAOR is 1. 0: Operation disabled on all channels 1: Operation enabled on all channels

DMARSRA and DMARSRB are 32-bit readable/writable registers that specify the transfer request source for each channel, together with the RS bits in each CHCRn. When the channel is not used, or DMA transfer initiated by an auto-request or a TMU input capture interrupt is used, the DMARSR corresponding to that channel should be set to H'00.

To resume a DMA transfer that has stopped due to an address error (AE = 1 in DMAOR) or an NMI interrupt (NMIF = 1 in DMAOR), re-specify this register value, regardless of whether or not a transfer request source to any channel has been changed, before specifying AE = 0 in DMAOR or NMIF = 0 in DMAOR.

- DMARSRA

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CH0 WEN	CH0 RS6	CH0 RS5	CH0 RS4	CH0 RS3	CH0 RS2	CH0 RS1	CH0 RS0	CH1 WEN	CH1 RS6	CH1 RS5	CH1 RS4	CH1 RS3	CH1 RS2	CH1 RS1	CH1 RS0
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
	CH2 WEN	CH2 RS6	CH2 RS5	CH2 RS4	CH2 RS3	CH2 RS2	CH2 RS1	CH2 RS0	CH3 WEN	CH3 RS6	CH3 RS5	CH3 RS4	CH3 RS3	CH3 RS2	CH3 RS1	CH3 RS0
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

- DMARSRB

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CH4 WEN	CH4 RS6	CH4 RS5	CH4 RS4	CH4 RS3	CH4 RS2	CH4 RS1	CH4 RS0	CH5 WEN	CH5 RS6	CH5 RS5	CH5 RS4	CH5 RS3	CH5 RS2	CH5 RS1	CH5 RS0
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
	CH6 WEN	CH6 RS6	CH6 RS5	CH6 RS4	CH6 RS3	CH6 RS2	CH6 RS1	CH6 RS0	CH7 WEN	CH7 RS6	CH7 RS5	CH7 RS4	CH7 RS3	CH7 RS2	CH7 RS1	CH7 RS0
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	Bit Name	Initial Value	R/W	Description
31	CH0WEN	0	R/W	Bits CHnRS6 to CHnRS0 specify transfer request sources to each channel. DMARSRA bits are allocated to channels 0 to 3. When writing to the CHnRS6 to CHnRS0 bits for each channel, simultaneously write 1 to the CHnWEN bit. Clearing the CHnWEN bit to 0 will not change the values in the CHnRS6 to CHnRS0 bits of each channel and retain the previous values. The CHnWEN bit is write-enabled, but it does not retain the written value and is always read as 0. CHnRS[6:0] H'00: Unused or auto-request, TMU input capture interrupt H'10: DREQ0*[1] H'11: DREQ1*[1] H'12: DREQ2*[1] H'13: DREQ3*[1] H'14: DMABRG (LCDC reception, USB transmission/reception, HAC transmission/reception, SSI transmission/reception)*[1]*[2] H'20: SCIF(0) Transmit-data-empty H'21: SCIF(0) Receive-data-full H'22: SCIF(1) Transmit-data-empty H'23: SCIF(1) Receive-data-full H'24: SCIF(2) Transmit-data-empty H'25: SCIF(2) Receive-data-full H'26: HSPI Transmit data H'27: HSPI Receive data H'28: SIM Transmit data empty H'29: SIM Receive-data-full H'2B: MMC FIFO ready H'2C: ADC AD conversion end data transfer H'2D: Setting prohibited H'2E: Setting prohibited H'7F: --*[3] Other than above: Setting prohibited
30	CH0RS6	0	R/W	
29	CH0RS5	0	R/W	
28	CH0RS4	0	R/W	
27	CH0RS3	0	R/W	
26	CH0RS2	0	R/W	
25	CH0RS1	0	R/W	
24	CH0RS0	0	R/W	
23	CH1WEN	0	R/W	
22	CH1RS6	0	R/W	
21	CH1RS5	0	R/W	
20	CH1RS4	0	R/W	
19	CH1RS3	0	R/W	
18	CH1RS2	0	R/W	
17	CH1RS1	0	R/W	
16	CH1RS0	0	R/W	
15	CH2WEN	0	R/W	
14	CH2RS6	0	R/W	
13	CH2RS5	0	R/W	
12	CH2RS4	0	R/W	
11	CH2RS3	0	R/W	
10	CH2RS2	0	R/W	
9	CH2RS1	0	R/W	
8	CH2RS0	0	R/W	
7	CH3WEN	0	R/W	
6	CH3RS6	0	R/W	
5	CH3RS5	0	R/W	
4	CH3RS4	0	R/W	
3	CH3RS3	0	R/W	
2	CH3RS2	0	R/W	
1	CH3RS1	0	R/W	
0	CH3RS0	0	R/W	

Notes: 1. This setting is valid only in DMABRG mode. In external request 2-channel mode, DREQ0 or DREQ1 can be accepted (only for channel 0 or 1) regardless of this setting.

2. The DMABRG settings are valid only in channel 0. In DMABRG mode, when selecting channel 0 transfer request source in DMABRG, don't change the channel 0 transfer request source after the DMABRG DMA transfer starts. Normal operation is not guaranteed if the transfer request source is changed.

3. Use this setting when the DMA transfer is complete with the request in DMAC retained (DMARCR.REXn = 1). See (3) Notes on Transfer End in section 11.4.6, Ending DMA Transfer.

Note: n = 0 to 3

Bit	Bit Name	Initial Value	R/W	Description
31	CH4WEN	0	R/(W)	CHnRS6 to CHnRS0 specify transfer request sources to each channel. DMARSRB bits are allocated to channels 0 to 3. When writing to the CHnRS6 to CHnRS0 bits for each channel, simultaneously write 1 to the CHnWEN bit. Clearing the CHnWEN bit to 0 will not change the values in the CHnRS6 to CHnRS0 bits of each channel and retain the previous values. The CHnWEN bit is write-enabled, but it does not retain the written value and is always read as 0. CHnRS[6:0] H'00: Unused or auto-request, TMU input capture interrupt H'10: DREQ0*1 H'11: DREQ1*1 H'12: DREQ2*1 H'13: DREQ3*1 H'20: SCIF(0) Transmit-data-empty H'21: SCIF(0) Receive-data-full H'22: SCIF(1) Transmit-data-empty H'23: SCIF(1) Receive-data-full H'24: SCIF(2) Transmit-data-empty H'25: SCIF(2) Receive-data-full H'26: HSPI Transmit data H'27: HSPI Receive data H'28: SIM Transmit data empty H'29: SIM Receive-data-full H'2B: MMC FIFO ready H'2C: ADC AD conversion end data transfer H'2D: Setting prohibited H'2E: Setting prohibited H'7F: --*2 Other than above: Setting prohibited Notes: 1. This setting is valid only in DMABRG mode. It is invalid in external request 2-channel mode (channels 2 to 7 cannot accept external requests). 2. Use this setting when the DMA transfer is complete with the request in DMAC retained (DMARCR.REXn = 1). See (3) Notes on Transfer End in section 11.4.6, Ending DMA Transfer.
30	CH4RS6	0	R/W	
29	CH4RS5	0	R/W	
28	CH4RS4	0	R/W	
27	CH4RS3	0	R/W	
26	CH4RS2	0	R/W	
25	CH4RS1	0	R/W	
24	CH4RS0	0	R/W	
23	CH5WEN	0	R/(W)	
22	CH5RS6	0	R/W	
21	CH5RS5	0	R/W	
20	CH5RS4	0	R/W	
19	CH5RS3	0	R/W	
18	CH5RS2	0	R/(W)	
17	CH5RS1	0	R/(W)	
16	CH5RS0	0	R/W	
15	CH6WEN	0	R/(W)	
14	CH6RS6	0	R/W	
13	CH6RS5	0	R/W	
12	CH6RS4	0	R/W	
11	CH6RS3	0	R/W	
10	CH6RS2	0	R/W	
9	CH6RS1	0	R/W	
8	CH6RS0	0	R/W	
7	CH7WEN	0	R/(W)	
6	CH7RS6	0	R/W	
5	CH7RS5	0	R/W	
4	CH7RS4	0	R/W	
3	CH7RS3	0	R/W	
2	CH7RS2	0	R/(W)	
1	CH7RS1	0	R/(W)	
0	CH7RS0	0	R/W	

Note: n = 4 to 7

Refer to section 24.2.34, DMA Pin Control Register (DMAPCR), in section 24, Pin Function Controller (PFC) for details regarding DMAPCR.

11.3.8 DMA Request Control Register (DMARCR)

DMARCR is a 32-bit readable/writable register that switches between DRAK2 and DACK2 and between DRAK3 and DACK3 in DMABRG mode, selects the interrupt level for DRAK and DACK, detects $\overline{\text{DREQ0}}$ to $\overline{\text{DREQ3}}$, and specifies the acceptance priority of requests from $\overline{\text{DREQ0}}$ to $\overline{\text{DREQ3}}$ and DMABRG. The numbers 0, 1, 2, and 3 in the bit names of DS, RL and AL correspond to $\overline{\text{DREQ0}}$, $\overline{\text{DREQ1}}$, $\overline{\text{DREQ2}}$, and $\overline{\text{DREQ3}}$ (channel numbers are not indicated). This DMARCR setting is invalid in external request 2-channel mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	REX7	REX6	REX5	REX4	REX3	REX2	REX1	REX0	R/A3	R/A2	-	-	-	-	RPR1	RPR0
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	R	R	R	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	DS3	RL3	AL3	-	DS2	RL2	AL2	-	DS1	RL1	AL1	-	DS0	RL0	AL0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31	REX7	0	R	Channel 7 Request Reception*4*7 0: Channel 7 does not accept a transfer request 1: Channel 7 accepts a transfer request
30	REX6	0	R	Channel 6 Request Reception*4*7 0: Channel 6 does not accept a transfer request 1: Channel 6 accepts a transfer request
29	REX5	0	R	Channel 5 Request Reception*4*7 0: Channel 5 does not accept a transfer request 1: Channel 5 accepts a transfer request
28	REX4	0	R	Channel 4 Request Reception*4*7 0: Channel 4 does not accept a transfer request 1: Channel 4 accepts a transfer request
27	REX3	0	R	Channel 3 Request Reception*4*7 0: Channel 3 does not accept a transfer request 1: Channel 3 accepts a transfer request

26	REX2	0	R	Channel 2 Request Reception*4*7 0: Channel 2 does not accept a transfer request 1: Channel 2 accepts a transfer request
25	REX1	0	R	Channel 1 request Reception*4*5*7 0: Channel 1 does not accept a transfer request 1: Channel 1 accepts a transfer request
24	REX0	0	R	Channel 0 Request Reception*4*6*7 0: Channel 0 does not accept a transfer request 1: Channel 0 accepts a transfer request
23	R/A3	0	R/W	DRAK3/DACK3 Select 0: Outputs DRAK3 1: Outputs DACK3
22	R/A2	0	R/W	DRAK2/DACK2 Select 0: Outputs DRAK2 1: Outputs DACK2
21 to 18	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
17 16	RPR1 RPR0	0 0	R/W R/W	Request Priority 1 and 0 Select the request priority order. 00: DMABRG*1 > DREQ0 > DREQ1 > DREQ2 > DREQ3 01: DMABRG*1 > DREQ0 > DREQ1 > DREQ2 > DREQ3 10: DREQ0 > DMABRG*1 > DREQ1 > DREQ2 > DREQ3 11: Round-robin (Initial setting: DMABRG*1 > DREQ0 > DREQ1 > DREQ2 > DREQ3) Note: Setting RPR[1:0] = 01 will make all channels disabled to receive a transfer request from on-chip peripheral modules except for LCDC, HAC, SSI, USB, and TMU or an external request ($\overline{\text{DREQ}}$), after having accepted a DMABRG request (REX0 = 1). When the DMABRG request is cleared (REX0 = 0), transfer requests are acceptable.

15	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
14	DS3	0	R/W	$\overline{\text{DREQ3}}$ Select 0: Low-level detection 1: Falling-edge detection
13	RL3	0	R/W	Request Check Level 3 0: DRAK3 high-active 1: DRAK3 low-active
12	AL3	0	R/W	Acknowledge Level 3 0: DACK3 high-active 1: DACK3 low-active
11	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
10	DS2	0	R/W	$\overline{\text{DREQ2}}$ Select 0: Low-level detection 1: Falling-edge detection
9	RL2	0	R/W	Request Check Level 2 0: DRAK2 high-active 1: DRAK2 low-active
8	AL2	0	R/W	Acknowledge Level 2 0: DACK2 high-active 1: DACK2 low-active
7	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
6	DS1	0	R/W	$\overline{\text{DREQ1}}$ Select 0: Low-level detection 1: Falling-edge detection
5	RL1	0	R/W	Request Check Level 1 0: DRAK1 high-active 1: DRAK1 low-active

4	AL1	0	R/W	Acknowledge Level 1 0: DACK1 high-active 1: DACK1 low-active
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
2	DS0	0	R/W	$\overline{\text{DREQ0}}$ Select 0: Low-level detection 1: Falling-edge detection
1	RL0	0	R/W	Request Check Level 0 0: DRAK0 high-active 1: DRAK0 low-active
0	AL0	0	R/W	Acknowledge Level 0 0: DACK0 high-active 1: DACK0 low-active

Notes: 1. Internal priority of the DMABRG: LCDC > A–B–C–D*2*3 > USB
2. A = HAC(0)/SSI(0) transmission, B = HAC(0)/SSI(0) reception, C = HAC(1)/SSI(1) transmission, and D = HAC(1)/SSI(1) reception
3. A–B–C–D is the round-robin method.
4. This bit is not set to 1 when TMU channel 2 (input capture interrupt) or auto-request has been accepted.
5. This bit is not set to 1 when $\overline{\text{DREQ1}}$ is accepted in external request 2-channel mode.
6. This bit is not set to 1 when $\overline{\text{DREQ0}}$ is accepted in external request 2-channel mode.
7. An address error (DMAOR.AE=1) or an NMI interrupt (DMAOR.NMIF=1) may result in REXn=1 although the corresponding channel accepts a DMA transfer request. For details of an address error or NMI interrupt, refer to (2) Ending Transfer Simultaneously on All Channels, in section 11.4.6 Ending DMA Transfer.

DMABRGCR is a 32-bit readable/writable register that has enable bits which generate HAC, SSI, and USB interrupts, and flags which indicate interrupts that have been generated.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	A1RXHE	A1RXEE	A1TXHE	A1TXEE	A0RXHE	A0RXEE	A0TXHE	A0TXEE	A1RXHF	A1RXEF	A1TXHF	A1TXEF	A0RXHF	A0RXEF	A0TXHF	A0TXEF
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
	-	-	-	-	-	-	UAE	UTE	-	-	-	-	-	-	UAF	UTF
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	R	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31	A1RXHE	0	R/W	HAC(1)/SSI(1) Receive Side Half Data Transfer End Interrupt Enable 0: Disabled 1: Enabled
30	A1RXEE	0	R/W	HAC(1)/SSI(1) Receive Side All Data Transfer End Interrupt Enable 0: Disabled 1: Enabled
29	A1TXHE	0	R/W	HAC(1)/SSI(1) Transmit Side Half Data Transfer End Interrupt Enable 0: Disabled 1: Enabled
28	A1TXEE	0	R/W	HAC(1)/SSI(1) Transmit Side All Data Transfer End Interrupt Enable 0: Disabled 1: Enabled
27	A0RXHE	0	R/W	HAC(0)/SSI(0) Receive Side Half Data Transfer End Interrupt Enable 0: Disabled 1: Enabled
26	A0RXEE	0	R/W	HAC(0)/SSI(0) Receive Side All Data Transfer End Interrupt Enable 0: Disabled 1: Enabled

25	A0TXHE	0	R/W	HAC(0)/SSI(0) Transmit Side Half Data Transfer End Interrupt Enable 0: Disabled 1: Enabled
24	A0TXEE	0	R/W	HAC(0)/SSI(0) Transmit Side All Data Transfer End Interrupt Enable 0: Disabled 1: Enabled
23	A1RXHF	0	R/W	HAC(1)/SSI(1) Receive Side Half Data Transfer End Interrupt Flag 0: An interrupt has not occurred. [Clearing condition] When 0 is written to A1RXHF after reading A1RXHF = 1 1: An interrupt has occurred.
22	A1RXEF	0	R/W	HAC(1)/SSI(1) Receive Side All Data Transfer End Interrupt Flag 0: An interrupt has not occurred. [Clearing condition] When 0 is written to A1RXEF after reading A1RXEF = 1 1: An interrupt has occurred.
21	A1TXHF	0	R/W	HAC(1)/SSI(1) Transmit Side Half Data Transfer End Interrupt Flag 0: An interrupt has not occurred. [Clearing condition] When 0 is written to A1TXHF after reading A1TXHF = 1 1: An interrupt has occurred.
20	A1TXEF	0	R/W	HAC(1)/SSI(1) Transmit Side All Data Transfer End Interrupt Flag 0: An interrupt has not occurred. [Clearing condition] When 0 is written to A1TXEF after reading A1TXEF = 1 1: An interrupt has occurred.

19	A0RXHF	0	R/W	HAC(0)/SSI(0) Receive Side Half Data Transfer End Interrupt Flag 0: An interrupt has not occurred. [Clearing condition] When 0 is written to A0RXHF after reading A0RXHF = 1 1: An interrupt has occurred.
18	A0RXEF	0	R/W	HAC(0)/SSI(0) Receive Side All Data Transfer End Interrupt Flag 0: An interrupt has not occurred. [Clearing condition] When 0 is written to A0RXEF after reading A0RXEF = 1 1: An interrupt has occurred.
17	A0TXHF	0	R/W	HAC(0)/SSI(0) Transmit Side Half Data Transfer End Interrupt Flag 0: An interrupt has not occurred. [Clearing condition] When 0 is written to A0TXHF after reading A0TXHF = 1 1: An interrupt has occurred.
16	A0TXEF	0	R/W	HAC(0)/SSI(0) Transmit Side All Data Transfer End Interrupt Flag 0: An interrupt has not occurred. [Clearing condition] When 0 is written to A0TXEF after reading A0TXEF = 1 1: An interrupt has occurred.
15 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9	UAE	0	R/W	USB Address Error Interrupt Enable 0: Disabled 1: Enabled
8	UTE	0	R/W	USB Transfer End Interrupt Enable 0: Disabled 1: Enabled
7 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

1	UAF	0	R/W	USB Address Error Interrupt Flag 0: An interrupt has not occurred. [Clearing condition] When 0 is written to UAF after reading UAF = 1 1: An interrupt has occurred.
0	UTF	0	R/W	USB Transfer End Interrupt Flag 0: An interrupt has not occurred. [Clearing condition] When 0 is written to UTF after reading UTF = 1 1: An interrupt has occurred.

11.3.10 DMA Audio Source Address Register (DMAATXSAR)

DMAATXSAR is a 32-bit readable/writable register that specifies the source start address of a DMA transfer from synchronous DRAM to the HAC or SSI codec. DMAATXSAR0 corresponds to HAC(0) or SSI(0) and DMAATXSAR1 corresponds to HAC(1) or SSI(1). During a DMA transfer, the register value is not modified.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W:	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	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
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	R

DMAARXDAR is a 32-bit readable/writable register that specifies the destination start address of a DMA transfer from the HAC or SSI codec to synchronous DRAM. DMAARXDAR0 corresponds to HAC(0) or SSI(0) and DMAARXDAR1 corresponds to HAC(1) or SSI(1). During a DMA transfer, the register value is not modified.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W:	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	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
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	R

11.3.12 DMA Audio Transmit Transfer Count Register (DMAATXTCR)

DMAATXTCR is a 32-bit readable/writable register that specifies the number of bytes of source audio data transferred in a DMA transfer to the HAC or SSI codec. DMAATXTCR0 corresponds to HAC(0) or SSI(0) and DMAATXTCR1 corresponds to HAC(1) or SSI(1). During a DMA transfer the register value is not modified. This register should be set from H'0000 0004 to H'03FF FFFC (bits 31 to 26, 1, and 0 are fixed at 0). Do not set H'0000 0000. Normal operation is not guaranteed when H'0000 0000 is set.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
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

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
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	R

DMAARXTCR is a 32-bit readable/writable register that specifies the number of bytes of received audio data transferred in a DMA transfer from the HAC or SSI codec. DMAARXTCR0 corresponds to HAC(0) or SSI(0) and DMAARXTCR1 corresponds to HAC(1) or SSI(1). During a DMA transfer the register value is not modified. This register should be set from H'0000 0004 to H'03FF FFFC (bits 31 to 26, 1, and 0 are fixed at 0). Do not set H'0000 0000. Normal operation is not guaranteed when H'0000 0000 is set.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
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

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
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	R

DMAACR is a 32-bit readable/writable register that specifies the DMA operating mode of the HAC or SSI codec. DMAACR0 corresponds to HAC(0) or SSI(0) and DMAACR1 corresponds to HAC(1) or SSI(1).

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	RAM1	RAM0	-	-	-	-	-	RAR	RDS	RDE
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	R	R	R	R	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	TAM1	TAM0	-	-	-	-	-	TAR	TDS	TDE
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	R	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 26	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
25	RAM1	0	R/W	Receive Data Alignment Setting
24	RAM0	0	R/W	These bits specify the data alignment method for writing receive data to an external memory. For details of the data alignment method for the receive slot data and external bus, see table 11.5 (1). 00: Alignment control is not performed 01: Longword data is transferred as four byte-data 10: Longword data is transferred as two word-data 11: Setting prohibited
23 to 19	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
18	RAR	0	R/W	DMA Auto Reload Setting Specifies the use or unuse of auto address reload to continue a DMA transfer when the number of bytes in the receive DMA transfer reaches the number of transfer bytes specified by DMAARXTCRn. 0: Address of receive DMA not auto reloaded 1: Address of receive DMA auto reloaded

17	RDS	0	R/W	HAC/SSI Receive DMA Termination Setting this bit to 1 forcibly terminates the receive DMA transfer. • When writing 0: Write operation is ignored 1: Receive DMA transfer is forcibly terminated • When reading 0: Transfer is completed 1: Transfer is being performed
16	RDE	0	R/W	HAC/SSI Receive DMA Transfer Activation Control Controls the receive DMA transfer activation. Write operation is ignored during transfer. To reactivate a receive DMA transfer, read this bit as 0 and then write 1 to it. • When writing 0: Write operation is ignored 1: Receive DMA transfer is activated • When reading 0: Transfer is completed 1: Transfer is being performed
15 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9 8	TAM1 TAM0	0 0	R/W R/W	Transmit Data Alignment Setting These bits specify the data alignment method for reading transmit data from an external memory. For details of the data alignment method for the transmit slot data and external bus, see Table 11.5 (2). 00: Alignment control is not performed 01: A longword is transferred as four bytes 10: A longword is transferred as two words 11: Setting prohibited
7 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

2	TAR	0	R/W	HAC/SSI Transmit DMA Auto Reload Setting Specifies the use or unuse of auto address reload to continue a DMA transfer when the number of bytes in the transmit DMA transfer reaches the number of transfer bytes specified by DMAARXTCRn. 0: Address of transmit DMA not auto reloaded 1: Address of transmit DMA auto reloaded
1	TDS	0	R/W	HAC/SSI Transmit DMA Termination Setting this bit to 1 forcibly terminates a transmit DMA transfer. • When writing 0: Write operation is ignored 1: Transmit DMA transfer is forcibly terminated • When reading 0: Transfer is completed 1: Transfer is being performed
0	TDE	0	R/W	HAC/SSI Transmit DMA Transfer Activation Control Controls the transmit DMA transfer activation. Write operation is ignored during transfer. To reactivate a transmit DMA transfer, read this bit as 0 and then write 1 to it. • When writing 0: Write operation is ignored 1: Transmit DMA transfer is activated • When reading 0: Transfer is completed 1: Transfer is being performed

			Data Bus				Slot Data			
RAM1	RAM0	Size	31 to 24	23 to 16	15 to 8	7 to 0	31 to 24	23 to 16	15 to 8	7 to 0
0	0	No Alignment (One longword)	D3	D2	D1	D0	D3	D2	D1	D0
0	1	Longword (four bytes)	D0	D1	D2	D3	D3	D2	D1	D0
1	0	Longword (two words)	D1	D0	D3	D2	D3	D2	D1	D0

Table 11.5 (2) Data Alignment for Transmit Slot Data and External Bus

			Data Bus				Slot Data			
TAM1	TAM0	Size	31 to 24	23 to 16	15 to 8	7 to 0	31 to 24	23 to 16	15 to 8	7 to 0
0	0	No Alignment (One longword)	D3	D2	D1	D0	D3	D2	D1	D0
0	1	Longword (four bytes)	D3	D2	D1	D0	D0	D1	D2	D3
1	0	Longword (two words)	D3	D2	D1	D0	D1	D0	D3	D2

11.3.15 DMA Audio Transmit Transfer Counter (DMAATXTCNT)

DMAATXTCNT is a 32-bit read-only register that indicates the number of bytes remaining to be transferred in the transfer bytes specified by DMAATXTCR. This register is write-prohibited. DMAATXTCNT0 corresponds to HAC(0) or SSI(0) and DMAATXTCNT1 corresponds to HAC(1) or SSI(1). Writing 1 to the TDE bit in DMAACR sets the DMAATXTCR value in this register. On forced termination, the number of transfer bytes remaining at that time is indicated.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
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
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

DMAARXTCNT is a 32-bit read-only register that indicates the number of remaining bytes of transfer bytes specified by DMAARXTCR. This register is write-prohibited. DMAARXTCNT0 corresponds to HAC(0) or SSI(0) and DMAARXTCNT1 corresponds to HAC(1) or SSI(1). Writing 1 to the RDE bit in DMAACR sets the DMAARXTCR value in this register. On forced termination, the number of transfer bytes remaining at that time is indicated.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
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
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

11.3.17 DMA USB Source Address Register (DMAUSAR)

DMAUSAR is a 32-bit readable/writable register that specifies the source address of a DMA transfer. The settings in this register are valid only for the DMA transfer between the USB internal shared memory (hereafter referred to as shared memory) and synchronous DRAM. During USB DMA transfer, the register value can be read but cannot be modified. The address should be specified as a 32-bit boundary.

Specify this register value as a synchronous DRAM address for a DMA transfer from synchronous DRAM to the shared memory, and as a shared memory address for a transfer from the shared memory to synchronous DRAM. In the case of a transfer from the shared memory to synchronous DRAM, the address should be within the transfer source, that is the shared memory area (H'FE34 1000 to H'FE34 2FFC). When an address outside of this area is specified, the DMAC detects a USB address error and terminates the USB DMA transfer.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	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
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	R

DMAUDAR is a 32-bit readable/writable register that specifies the destination address of a DMA transfer. The settings in this register are valid only for the DMA transfer between the shared memory and synchronous DRAM. During USB DMA transfer, the register value can be read but cannot be modified. The address should be specified as a 32-bit boundary.

Specify this register value as a shared memory address for a DMA transfer from synchronous DRAM to the shared memory, and as a synchronous DRAM address for a transfer from the shared memory to synchronous DRAM. In the case of a transfer from synchronous DRAM to the shared memory, the address should be within the transfer destination, that is the shared memory area (H'FE34 1000 to H'FE34 2FFC). When an address outside of this area is specified, the DMAC detects a USB address error and terminates the USB DMA transfer.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	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
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	R

DMAURWSZ is a 32-bit readable/writable register that specifies the transfer direction and data size. During USB DMA transfer, the register value can be read but cannot be modified.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	RW
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

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	SZ12	SZ11	SZ10	SZ9	SZ8	SZ7	SZ6	SZ5	SZ4	SZ3	SZ2	SZ1	SZ0
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/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 17	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
16	RW	0	R/W	Transfer Direction 0: Specifies a DMA transfer from synchronous DRAM to the shared memory 1: Specifies a DMA transfer from the shared memory to synchronous DRAM
15 to 13	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
12 to 0	SZ12 to SZ0	All 0	R/W	Transfer Data Size Specifies the number of bytes to be transferred in a USB DMA transfer. Up to 8191 bytes can be specified. Setting these bits as H'0000 (SZ[12:0] = H'0000) will not perform transfer, but setting the START bit in DMAUCR to 1 sets the UTF bit in DMABRGCR to 1.

DMAUCR is a 32-bit readable/writable register that specifies the start of USB DMA transfer between the shared memory and synchronous DRAM, and the data alignment mode. The setting of the data alignment mode is also valid for accesses to the USB from the CPU. For details of the data alignment mode, see section 11.6.13, USB Endian Conversion Function.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	CVRT1	CVRT0
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/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	START	-
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/W	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 18	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
17 16	CVRT1 CVRT0	0 0	R/W R/W	Alignment Mode 00: Alignment is not performed 01: Byte boundary mode 10: Longword/word boundary mode 11: Setting prohibited
15 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1	START	0	R/W	DMA Transfer Start Setting this bit to 1 starts a USB DMA transfer. When the USB DMA transfer is completed, this bit is automatically cleared to 0. • When writing 0: Invalid 1: Starts a USB DMA transfer • When reading 0: USB DMA transfer is stopped 1: USB DMA transfer is being performed
0	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

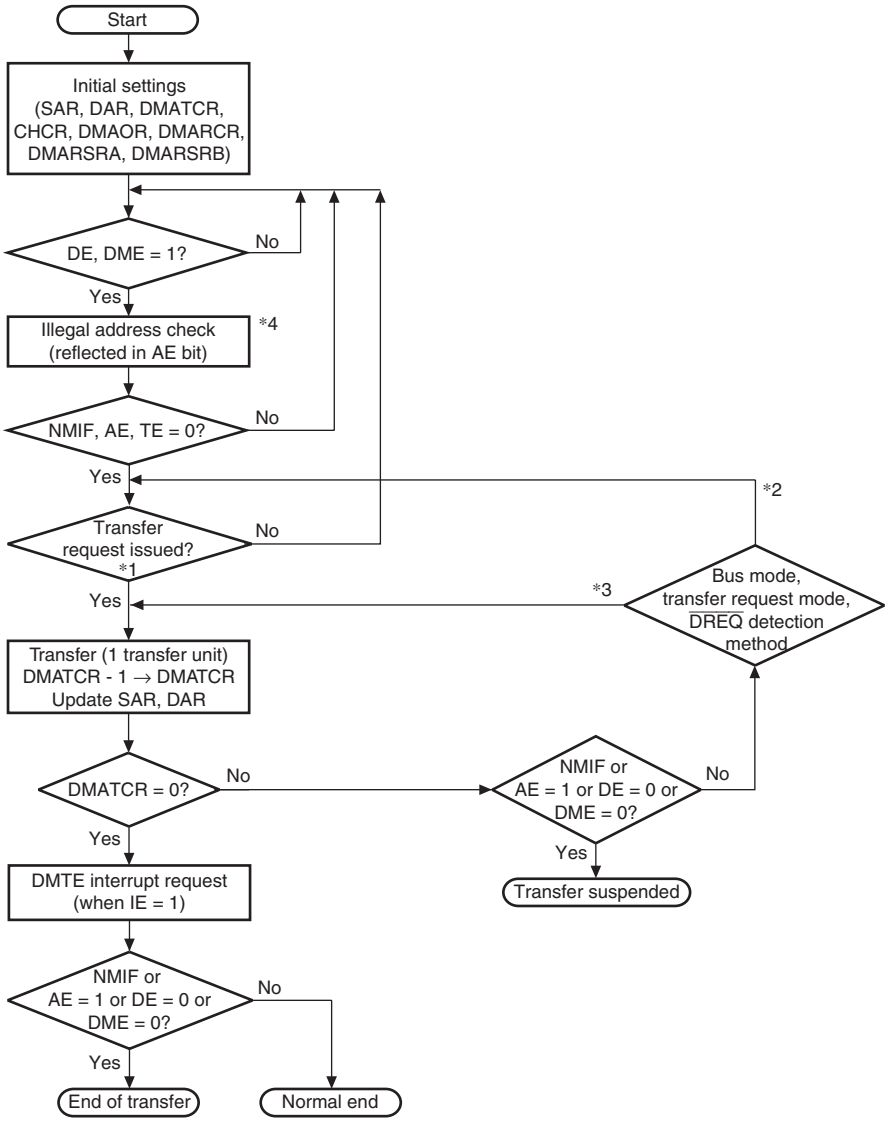
When a DMA transfer request is issued, the DMAC starts the transfer according to the predetermined channel priority order. It ends the transfer when the transfer end conditions are satisfied. Transfers can be requested in three modes: auto-request, external request, and on-chip peripheral module request. There are two modes for DMA transfer: single address mode and dual address mode. Either burst mode or cycle steal mode can be selected as the bus mode.

11.4.1 DMA Transfer Procedure

After the desired transfer conditions have been set in SAR, DAR, DMATCR, CHCR, DMAOR, DMARCR, DMARSRA, and DMARSRB, the DMAC transfers data according to the following procedure:

1. The DMAC checks to see if transfer is enabled (DE = 1, DME = 1, TE = 0, NMIF = 0, AE = 0).
2. When a transfer request is issued and transfer has been enabled, the DMAC transfers one transfer unit of data (determined by bits TS2 to TS0). In auto-request mode, the transfer begins automatically when the DE and DME bits are set to 1. The DMATCR value is decremented by 1 for each transfer. The actual transfer flow depends on the address mode and bus mode.
3. When the specified number of transfers have been completed (when the DMATCR value reaches 0), the transfer ends normally. If the IE bit in CHCR is set to 1 at this time, the DMAC sends a DMTE interrupt request to the CPU.
4. If a DMAC address error or NMI interrupt occurs, the DMAC suspends the transfer. It also suspends the transfer when the DE bit in CHCR or the DME bit in DMAOR is cleared to 0. In the event of an address error, the DMAC issues a forced DMAE interrupt request to the CPU.

For details of DMA transfer end and suspension, see section 11.4.6, Ending DMA Transfer.



Notes: 1. In auto-request mode, transfer begins when the NMIF, AE, and TE bits are all 0 and the DE and DME bits are set to 1.
2. $\overline{\text{DREQ}}$ level detection (external request) in burst mode, or cycle steal mode.
3. $\overline{\text{DREQ}}$ edge detection (external request) in burst mode, or auto-request mode in burst mode.
4. An illegal address is detected by comparing bits TS2–TS0 in CHCRn with SARn and DARn.

Figure 11.3 DMAC Transfer Flowchart

DMA transfer requests are basically generated at either the data transfer source or destination, but they can also be issued by external devices or on-chip peripheral modules that are neither the source nor the destination.

Transfers can be requested in three modes: auto-request, external request, and on-chip peripheral module request. The transfer request mode is selected by bits RS3 to RS0 in CHCR0 to CHCR7 and settings of DMARSRA and DMARSRB.

(1) Auto-Request Mode

The DMAC can automatically generate a transfer request signal internally in Auto-Request Mode when receiving no transfer request signal from an external source, as in a memory-to-memory transfer or a transfer between memory and an on-chip peripheral module unable to request a transfer. Setting the DE bit in CHCR of the channel to be used and the DME bit in DMAOR to 1 starts the transfer. However, the TE bit in CHCR of the channel to be used and the NMIF and AE bits in DMAOR must all be 0.

(2) External Request Mode

In this mode, the DMAC performs a transfer in response to a transfer request signal ($\overline{\text{DREQ}}$) from an external device. Select one of the modes shown in table 11.6 according to the application system. If DMA transfer is enabled (DE = 1, DME = 1, TE = 0, NMIF = 0, AE = 0), transfer starts when $\overline{\text{DREQ}}$ is input. The DS bits in CHCR0 to CHCR7 and bits DS3 to DS0 in DMARCR are used to select either falling edge detection or low level detection of the $\overline{\text{DREQ}}$ signal (level detection with DS = 0, edge detection with DS = 1).

The source of the transfer request does not have to be the data transfer source or destination.

Table 11.6 Selecting External Request Mode with RS Bits

RS3	RS2	RS1	RS0	Address Mode	Transfer Source	Transfer Destination
0	0	0	0	Dual address mode	External memory, memory-mapped external device, or on-chip peripheral module	External memory, memory-mapped external device, or on-chip peripheral module
		1	0	Single address mode	External memory or memory-mapped external device	External device with DACK
			1	Single address mode	External device with DACK	External memory or memory-mapped external device

1. When at least either of DMAOR.DME and CHCR.DE is 0, and DMAOR.NMIF, DMAOR.AE, and CHCR.TE are all 0, the DMAC will hold an input external request ($\overline{\text{DREQ}}$: edge detection) until DMA transfer is either executed or canceled. Since DMA transfer is not enabled in this case (DME = 0 or DE = 0), DMA transfer is not initiated. DMA transfer is started after it is enabled (DME = 1, DE = 1, NMIF = 0, AE = 0, TE = 0).
2. If an external request ($\overline{\text{DREQ}}$) is input while DMA transfer is enabled (DME = 1, DE = 1, NMIF = 0, AE = 0, TE = 0), DMA transfer is started.
3. An external request ($\overline{\text{DREQ}}$) will be ignored if it is input with TE = 1, NMIF = 1, or AE = 1 during a power-on reset or manual reset, in deep sleep mode or standby mode, or while the DMAC is in the module standby state. Write 1 to CHCRn or CHSET or re-specify the channel resource in DMARSRA or DMARSRB before enabling DMAC transfer in order to resume DMA transfer in DMABRG mode.
4. A previously input external request will be canceled by the occurrence of an NMI interrupt (NMIF = 1) or address error (AE = 1), or by a power-on reset or manual reset.
5. In this LSI, it is possible to cancel a previously input external request ($\overline{\text{DREQ}}$). In external request 2-channel mode, drive the $\overline{\text{DREQ}}$ pin high after clearing the DS bit in CHCRn to 0. In DMABRG mode, set the CHSET bit in CHCRn to 1.

- Usage Notes

  The DMAC detects an external request ($\overline{\text{DREQ}}$) at a low level or falling edge. Ensure to hold the external request ($\overline{\text{DREQ}}$) signal high when there is no DMA transfer request from an external device after a power-on reset or manual reset.

  When DMA transfer is resumed, check whether a DMA transfer request is being held.

(3) On-Chip Peripheral Module Request Mode

In this mode, the DMAC performs a transfer in response to a transfer request signal (interrupt request signal) from an on-chip peripheral module.

To output a transfer request from an on-chip peripheral module, set the DMA transfer request enable bit for that module.

Receiveing simultaneous transfer requests on two or more channels, the DMAC selects a channel according to a predetermined priority system, either in a fixed mode or round robin mode. The mode is selected with priority bits PR1 and PR0 in DMAOR.

(1) Fixed Mode

In this mode, the relative channel priorities remain fixed. The following priority orders are available in fixed mode:

- CH0 > CH1 > CH2 > CH3 > CH4 > CH5 > CH6 > CH7
- CH0 > CH2 > CH3 > CH4 > CH5 > CH6 > CH7 > CH1
- CH2 > CH0 > CH1 > CH3 > CH4 > CH5 > CH6 > CH7

Select one of these priority orders by setting the PR1 and PR0 bits in DMAOR.

(2) Round Robin Mode

In round robin mode, each time the transfer of one transfer unit (byte, word, longword, quadword, or 32 bytes) ends on a given channel, that channel is assigned the lowest priority level. This is illustrated in figure 11.4. The order of priority in round robin mode immediately after a reset is CH0 > CH1 > CH2 > CH3 > CH4 > CH5 > CH6 > CH7. If no transfer request is accepted for any channel during DMA transfer, the priority order becomes CH0 > CH1 > CH2 > CH3 > CH4 > CH5 > CH6 > CH7.

Figure 11.4 Round Robin Mode

for channels 0 and 3, and channel 1 generates a transfer request during a transfer on channel 0. The operation of the DMAC in this case is as follows.

1. Transfer requests are issued simultaneously for channels 0 and 3.
2. Since channel 0 has a higher priority level than channel 3, the channel 0 transfer is executed first (channel 3 is on transfer standby).
3. A transfer request is issued for channel 1 during the channel 0 transfer (channels 1 and 3 are on transfer standby).
4. At the end of the channel 0 transfer, channel 0 shifts to the lowest priority level.
5. At this point, channel 1 has a higher priority level than channel 3, so the channel 1 transfer is started (channel 3 is on transfer standby).
6. At the end of the channel 1 transfer, channel 1 shifts to the lowest priority level.
7. The channel 3 transfer is started.
8. At the end of the channel 3 transfer, the channel 3 and channel 2 priority levels are lowered, giving channel 3 the lowest priority.

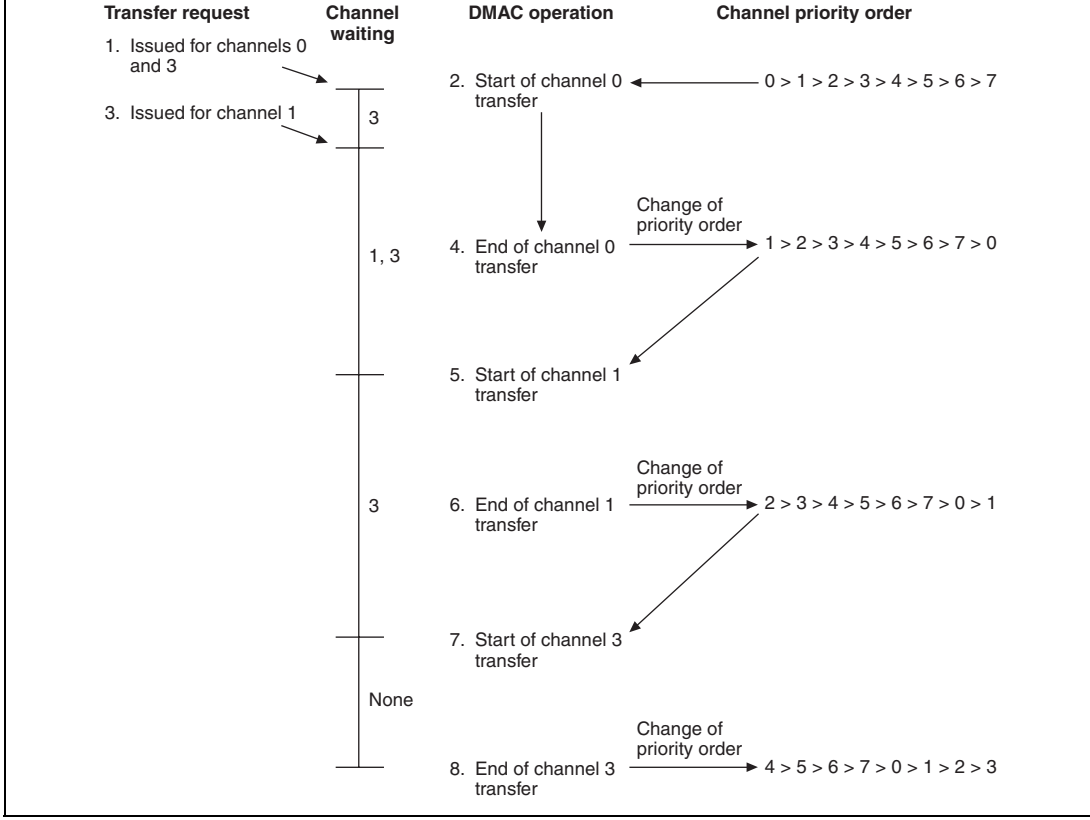


Figure 11.5 Example of Changes in Priority Order in Round Robin Mode

11.4.4 Types of DMA Transfer

The DMAC supports the types of transfer shown in table 11.7. It can operate in single address mode, in which either the transfer source or the transfer destination is accessed using the acknowledge signal, or in dual address mode, in which both the transfer source and transfer destination addresses are output. The actual transfer operation timing depends on the bus mode, which can be either burst mode or cycle steal mode.

	Transfer Destination			
Transfer Source	**External Device with DACK**	**External Memory**	**Memory-Mapped External Device**	**On-Chip Peripheral Module**
External device with DACK	Not available	Single address mode	Single address mode	Not available
External memory	Single address mode	Dual address mode	Dual address mode	Dual address mode
Memory-mapped external device	Single address mode	Dual address mode	Dual address mode	Dual address mode
On-chip peripheral module	Not available	Dual address mode	Dual address mode	Not available

(1) Address Modes

- Single Address Mode

  In single address mode, both the transfer source and the transfer destination are external; one is accessed by the DACK signal and the other by an address. In this mode, the DMAC performs a DMA transfer in one bus cycle by simultaneously outputting the external device strobe signal (DACK) to either the transfer source or transfer destination external device to access it, while outputting an address to the other side of the transfer. Figure 11.6 shows an example of a transfer between external memory and an external device with DACK in which the external device outputs data to the data bus and that data is written to external memory in the same bus cycle.

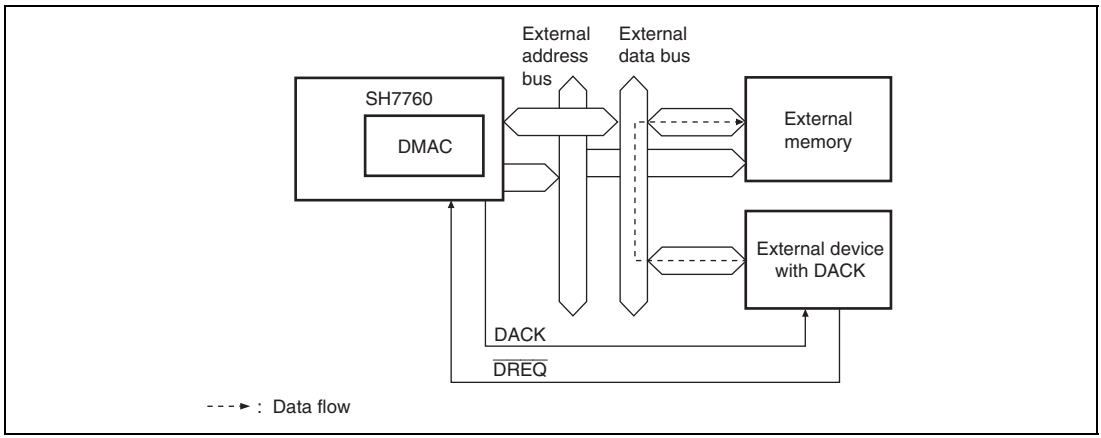


Figure 11.6 Data Flow in Single Address Mode

external device with DACK and external memory. Only the external request signal ($\overline{\text{DREQ}}$) is used in both of these cases.

Figure 11.7 shows the transfer timing for single address mode.

The access timing depends on the type of external memory. For details, see the descriptions of the memory interfaces in section 10, Bus State Controller (BSC).

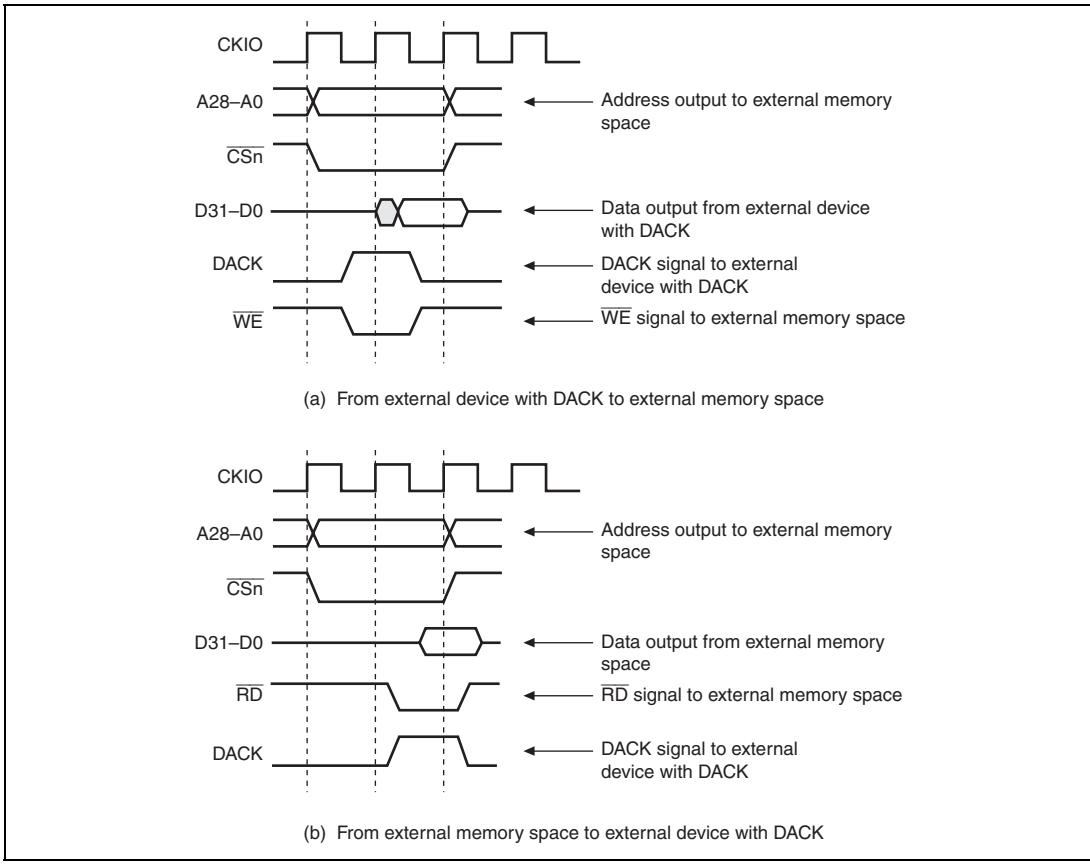


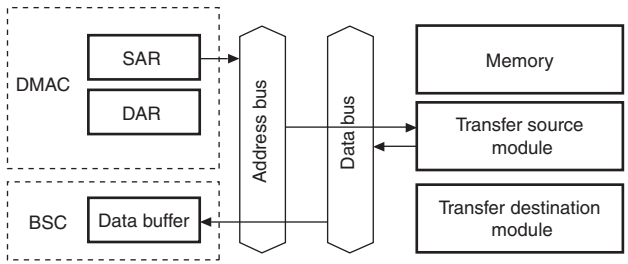
Figure 11.7 DMA Transfer Timing in Single Address Mode

Dual address mode is used to access both the transfer source and the transfer destination by address. The transfer source and destination can be either on-chip peripheral module or external address.

Even if the operand cache is used in RAM mode, the RAM cannot be set as the transfer source or transfer destination.

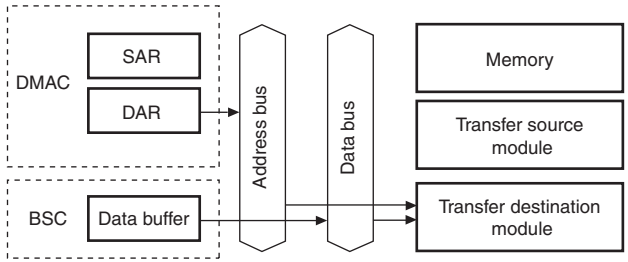
Since in dual address mode, data corresponding to the size specified by the TS bit in CHCRn is read from the transfer source in the data read cycle and is written to the transfer destination in the data write cycle, it is transferred in two bus cycles. In this process, the transfer data is temporarily stored in the data buffer in the bus state controller (BSC).

In a transfer between external memories such as that shown in figure 11.8, data is read from external memory into the BSC's data buffer in the read cycle, then written to the other external memory in the write cycle. Figure 11.9 shows the timing for this operation. The DACK output timing is the same as that of $\overline{\text{CSn}}$ in a read or write cycle specified by the AM bit in CHCRn.



Taking the SAR value as the address, data is read from the transfer source module and stored temporarily in the data buffer in the bus state controller (BSC).

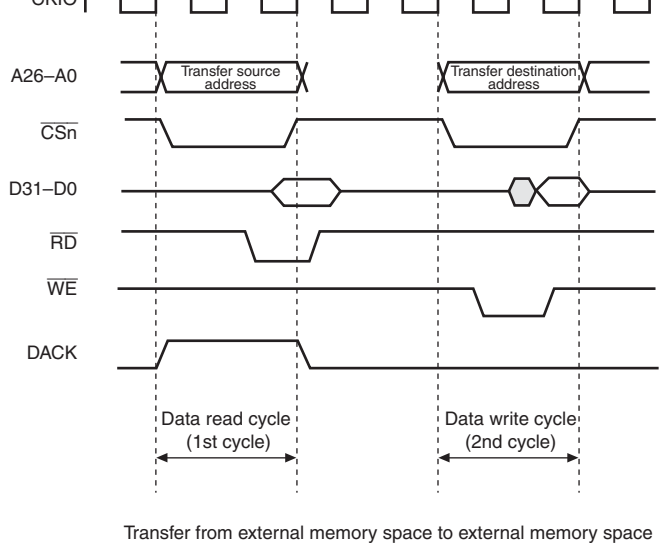
1st bus cycle



Taking the DAR value as the address, the data stored in the BSC's data buffer is written to the transfer destination module.

2nd bus cycle

Figure 11.8 Operation in Dual Address Mode



Transfer from external memory space to external memory space

Figure 11.9 Example of Transfer Timing in Dual Address Mode

(2) Bus Modes

There are two bus modes: cycle steal mode and burst mode. The bus mode is selected for each channel with the TM bit in CHCR0 to CHCR7.

- Cycle Steal Mode

  In cycle steal mode, the DMAC releases the bus to the CPU at the end of each transfer-unit (8-bit, 16-bit, 32-bit, 64-bit, or 32-byte) transfer. When the next transfer request is issued, the DMAC reacquires the bus from the CPU and carries out another transfer-unit transfer. At the end of this transfer, the bus is again given to the CPU. This is repeated until the transfer end condition is satisfied.

  In cycle steal mode, areas for transfer has no limitation by the settings of transfer request source, transfer source, and transfer destination.

  Figure 11.10 shows an example of DMA transfer timing in cycle steal mode. The following transfer conditions are used in this example:

  — Dual address mode

  — $\overline{\text{DREQ}}$ level detection

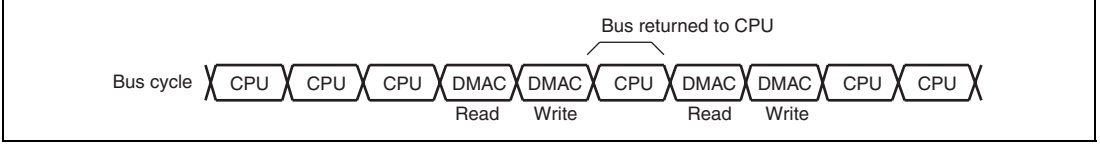


Figure 11.10 Example of DMA Transfer in Cycle Steal Mode

- Burst Mode

  In burst mode, once the DMAC has acquired the bus it holds the bus and transfers data continuously until the transfer end condition is satisfied. With $\overline{\text{DREQ}}$ low level detection in external request mode, however, when $\overline{\text{DREQ}}$ is driven high the bus passes to another bus master after the end of the DMAC transfer request that has already been accepted, even if the transfer end condition has not been satisfied.

  Figure 11.11 shows an example of DMA transfer timing in burst mode. The following transfer conditions are used in this example:

  — Single address mode

  — $\overline{\text{DREQ}}$ level detection (DS = 0 and TM = 1 in CHCRn, external request 2-channel mode)

  Note: Specify $\overline{\text{DREQ}}$ edge detection when performing burst transfer in DMABRG mode. Operations in burst mode with $\overline{\text{DREQ}}$ level detection in DMABRG mode are the same as those in cycle steal mode.

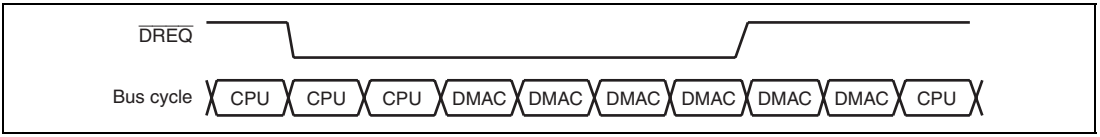


Figure 11.11 Example of DMA Transfer in Burst Mode

Table 11.8 shows the relationship between the type of DMA transfer, the request mode, and the bus mode.

Table 11.8 Relationship between DMA Transfer Type, Request Mode, and Bus Mode

Address Mode	Type of Transfer	Request Mode	Bus Mode	Transfer Size (Bits)
Single	External device with DACK and external memory	External	B/C	8/16/32/64/32B
	External device with DACK and memory-mapped external device	External	B/C	8/16/32/64/32B
Dual	External memory and external memory	Internal*[1], external*[5]	B/C	8/16/32/64/32B
	External memory and memory-mapped external device	Internal*[1], external*[5]	B/C	8/16/32/64/32B
	Memory-mapped external device and memory-mapped external device	Internal*[1], external*[5]	B/C	8/16/32/64/32B
	External memory and on-chip peripheral module	Internal*[2]	B/C*[3]	8/16/32/64*[4]
	Memory-mapped external device and on-chip peripheral module	Internal*[2]	B/C*[3]	8/16/32/64*[4]

Legend:

32B: 32-byte burst transfer

B: Burst

C: Cycle steal

External: External request

Internal: Auto-request and on-chip peripheral module request

Notes: 1. External request, auto-request, or on-chip peripheral module request possible.

2. Auto-request or on-chip peripheral module request possible.

3. Only cycle steal mode when the transfer request source is an on-chip peripheral module other than the DMABRG.

4. Access size permitted for the on-chip peripheral module register that is the transfer source or transfer destination.

5. See tables 11.9 (1) and 11.9 (2) for the transfer sources and transfer destinations in DMA transfer by means of an external request.

Table 11.9 (1) shows the memory interfaces that can be specified for the transfer source and transfer destination in DMA transfer initiated by an external request in external request 2-channel mode supported by this LSI.

Table 11.9 (1) External Request Transfer Sources and Destinations in External Request 2-Channel Mode

	Transfer Direction (Settable Memory Interface)				Address Mode	Usable DMAC Channels
	Transfer Source		Transfer Destination			
1	Synchronous DRAM		External device with DACK		Single	0, 1
2	External device with DACK		Synchronous DRAM		Single	0, 1
3	SRAM-type		External device with DACK		Single	0, 1
4	External device with DACK		SRAM-type		Single	0, 1
5	Synchronous DRAM		SRAM-type, MPX, PCMCIA	O	Dual	0, 1
6	SRAM-type, MPX, PCMCIA	O	Synchronous DRAM		Dual	0, 1
7	SRAM-type, PCMCIA, MPX		SRAM-type, MPX, PCMCIA	O	Dual	0, 1
8	SRAM-type, MPX, PCMCIA	O	SRAM-type, PCMCIA, MPX		Dual	0, 1

Legend:

O: DACK output setting in dual address mode transfer

Notes: 1. SRAM-type in the table indicates an SRAM, byte control SRAM, or burst ROM.

2. Memory interfaces in which transfer is possible in single address mode are SRAM, byte control SRAM, burst ROM, and synchronous DRAM.

3. When performing dual address mode transfer, make the DACK output setting for the SRAM, byte control SRAM, burst ROM, PCMCIA, or MPX interface.

- DMABRG Mode

  Table 11.9 (2) shows the memory interfaces that can be specified for the transfer source and transfer destination in DMA transfer initiated by an external request in DMABRG mode supported by this LSI.

	Transfer Direction (Settable Memory Interface)				Address Mode	Usable DMAC Channels
	Transfer Source		Transfer Destination			
1	Synchronous DRAM		External device with DACK		Single	All
2	External device with DACK		Synchronous DRAM		Single	All
3	Synchronous DRAM		SRAM-type, MPX, PCMCIA	O	Dual	All
4	SRAM-type, MPX, PCMCIA	O	Synchronous DRAM		Dual	All
5	SRAM-type, PCMCIA, MPX		SRAM-type, MPX, PCMCIA	O	Dual	All
6	SRAM-type, MPX, PCMCIA	O	SRAM-type, PCMCIA, MPX		Dual	All

Legend:

O: DACK output setting in dual address mode transfer

Notes: 1. SRAM-type in the table indicates an SRAM, byte control SRAM, or burst ROM.
2. Memory interface in which transfer is possible in single address mode is synchronous DRAM.
3. When performing dual address mode transfer, make the DACK output setting for the SRAM, byte control SRAM, burst ROM, PCMCIA, or MPX interface.

(4) Bus Mode and Channel Priority Order

When, for example, channel 1 is transferring data in burst mode, and a transfer request is issued to channel 0, which has a higher priority, the channel 0 transfer is started immediately.

If fixed mode has been set for the priority levels (CH0 > CH1), transfer on channel 1 is continued after transfer on channel 0 is completely finished, whether cycle steal mode or burst mode is set for channel 0.

If round robin mode has been set for the priority levels, transfer on channel 1 is restarted after one transfer unit of data is transferred on channel 0, whether cycle steal mode or burst mode is set for channel 0. Channel execution alternates in the order: channel 1 → channel 0 → channel 1 → channel 0.

An example of round robin mode operation is shown in figure 11.12.

Since channel 1 is in burst mode (in the case of edge sensing) regardless of whether fixed mode or round robin mode is set for the priority order, the bus is not released to the CPU until channel 1 transfer ends.

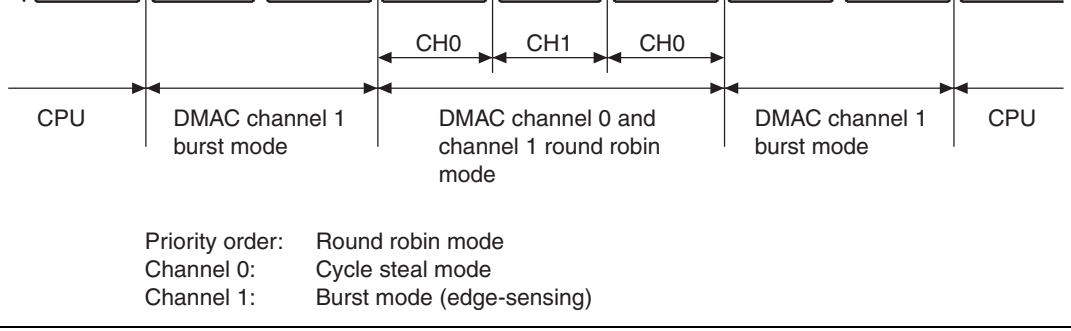


Figure 11.12 Bus Handling with Two DMAC Channels Operating

11.4.5 Number of Bus Cycles and $\overline{\text{DREQ}}$ Pin Sampling Timing

(1) Number of Bus Cycles

The number of bus cycles when the DMAC is the bus master is controlled by the bus state controller (BSC) just as it is when the CPU is the bus master. See section 10, Bus State Controller (BSC), for details.

(2) $\overline{\text{DREQ}}$ Pin Sampling Timing

In external request mode, the DMAC samples the $\overline{\text{DREQ}}$ pin at the rising edge of a CKIO clock signal. When detecting a $\overline{\text{DREQ}}$ input, the DMAC generates a bus cycle and performs DMA transfer after four CKIO cycles at the earliest.

In the case of $\overline{\text{DREQ}}$ falling edge sampling, the DMAC detects a $\overline{\text{DREQ}}$ input after two CKIO cycles (in the case of low-level sampling, one CKIO cycle).

The second and subsequent $\overline{\text{DREQ}}$ sampling operations are performed one cycle after the start of the first DMAC transfer bus cycle (in the case of external request 2-channel mode and single address mode).

DRAK is output for one cycle only, once each time $\overline{\text{DREQ}}$ is detected, regardless of the transfer mode or $\overline{\text{DREQ}}$ detection method. In the case of burst mode edge detection, $\overline{\text{DREQ}}$ is sampled in the first cycle only, and so DRAK is output in the first cycle only.

- Cycle Steal Mode

  In cycle steal mode, The $\overline{\text{DREQ}}$ sampling timing differs for dual address mode and single address mode, and for level detection and edge detection of $\overline{\text{DREQ}}$.

  For example, in figure 11.13 (external request 2-channel mode, cycle steal mode, dual address mode, level detection), DMAC transfer begins, at the earliest, four CKIO cycles after the first sampling operation. The second sampling operation is performed one cycle after the start of the first DMAC transfer write cycle. If $\overline{\text{DREQ}}$ is not detected at this time, sampling is executed in every subsequent cycle.

  In figure 11.15 (external request 2-channel mode, cycle steal mode, dual address mode, edge detection), DMAC transfer begins, at the earliest, five CKIO cycles after the first sampling operation. The second sampling operation begins from the cycle in which the first DMAC transfer read cycle ends. If $\overline{\text{DREQ}}$ is not detected at this time, sampling is executed in every subsequent cycle.

  For details of the timing for various types of memory access, see section 10, Bus State Controller (BSC).

  Figure 11.21 shows external request 2-channel mode, cycle steal mode, single address mode, and level detection. In this case, too, transfer is started, at the earliest, four CKIO cycles after the first $\overline{\text{DREQ}}$ sampling operation. The second sampling operation is performed one cycle after the start of the first DMAC transfer bus cycle.

  Figure 11.23 shows external request 2-channel mode, cycle steal mode, single address mode, and edge detection. In this case, transfer is started, at the earliest, five CKIO cycles after the first $\overline{\text{DREQ}}$ sampling operation. The second sampling begins one cycle after the first assertion of DRAK.

  In single address mode, the DACK signal is output every DMAC transfer cycle.

- Burst Mode, Dual Address Mode, Level Detection

  $\overline{\text{DREQ}}$ sampling timing in burst mode using dual address mode and level detection is virtually the same as for cycle steal mode.

  For example, in figure 11.17, DMAC transfer begins, at the earliest, four CKIO cycles after the first sampling operation. The second sampling operation is performed one cycle after the start of the first DMAC transfer write cycle.

  In the case of dual address mode transfer initiated by an external request, the DACK signal can be output in either the read cycle or the write cycle of the DMAC transfer according to the specification of the AM bit in CHCR.

$\overline{\text{DREQ}}$ sampling timing in burst mode using single address mode and level detection is shown in figure 11.20.

In the example shown in figure 11.25, DMAC transfer begins, at the earliest, four CKIO cycles after the first sampling operation, and the second sampling operation begins one cycle after the start of the first DMAC transfer bus cycle.

In single address mode, the DACK signal is output every DMAC transfer cycle.

In figure 11.29, with a 32-byte data size, 32-bit bus width, and SDRAM: row hit write, DMAC transfer begins, at the earliest, six CKIO cycles after the first sampling operation. The second sampling operation begins one cycle after DACK is asserted for the first DMAC transfer.

- Burst Mode, Dual Address Mode, Edge Detection

  In burst mode using dual address mode and edge detection, $\overline{\text{DREQ}}$ sampling is performed in the first cycle only.

  For example, in the case shown in figure 11.19, DMAC transfer begins, at the earliest, five CKIO cycles after the first sampling operation. DMAC transfer then continues until the end of the number of data transfers set in DMATCR. $\overline{\text{DREQ}}$ is not sampled during this time, and therefore DRAK is output in the first cycle only. In the case of dual address mode transfer initiated by an external request, the DACK signal can be output in either the read cycle or the write cycle of the DMAC transfer according to the specification of the AM bit in CHCR.

- Burst Mode, Single Address Mode, Edge Detection

  In burst mode using single address mode and edge detection, $\overline{\text{DREQ}}$ sampling is performed only in the first cycle.

  For example, in the case shown in figure 11.27, DMAC transfer begins, at the earliest, five cycles after the first sampling operation. DMAC transfer then continues until the end of the number of data transfers set in DMATCR. $\overline{\text{DREQ}}$ is not sampled during this time, and therefore DRAK is output in the first cycle only.

  In single address mode, the DACK signal is output every DMAC transfer cycle.

(4) Suspension of DMA Transfer with DREQ Level Detection

With $\overline{\text{DREQ}}$ level detection in burst mode or cycle steal mode, and in dual address mode or single address mode, the external device for which DMA transfer is being executed can determine at the rising edge of CKIO that DRAK has been asserted, and suspend DMA transfer by negating $\overline{\text{DREQ}}$. In this case, the next DRAK signal is not output.

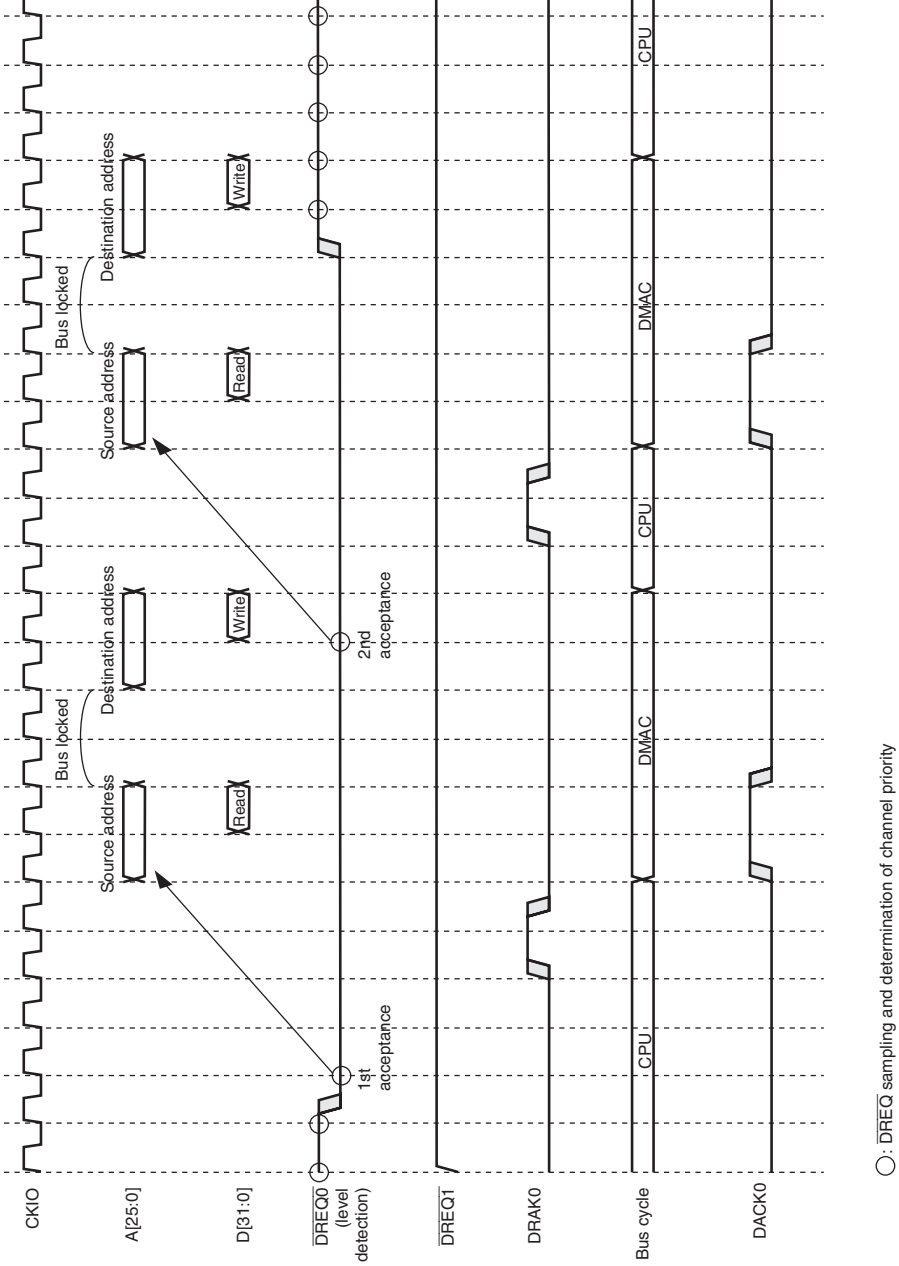


Figure 11.13 Dual Address Mode/Cycle Steal Mode in External Request 2-Channel Mode External Bus → External Bus/DREQ (Level Detection), DACK (Read Cycle)

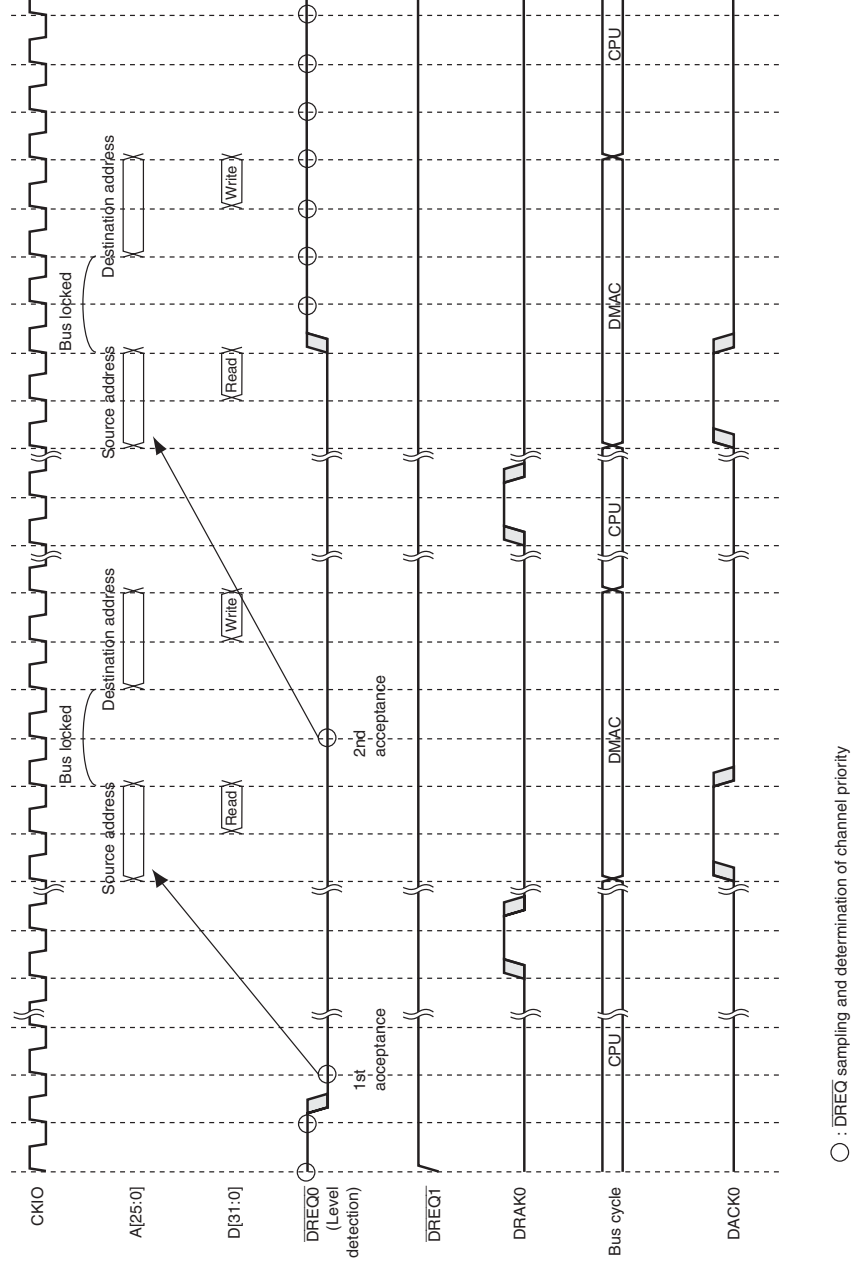


Figure 11.14 Dual Address Mode/Cycle Steal Mode in DMABRG Mode
External Bus → External Bus/DREQ (Level Detection), DACK (Read Cycle)

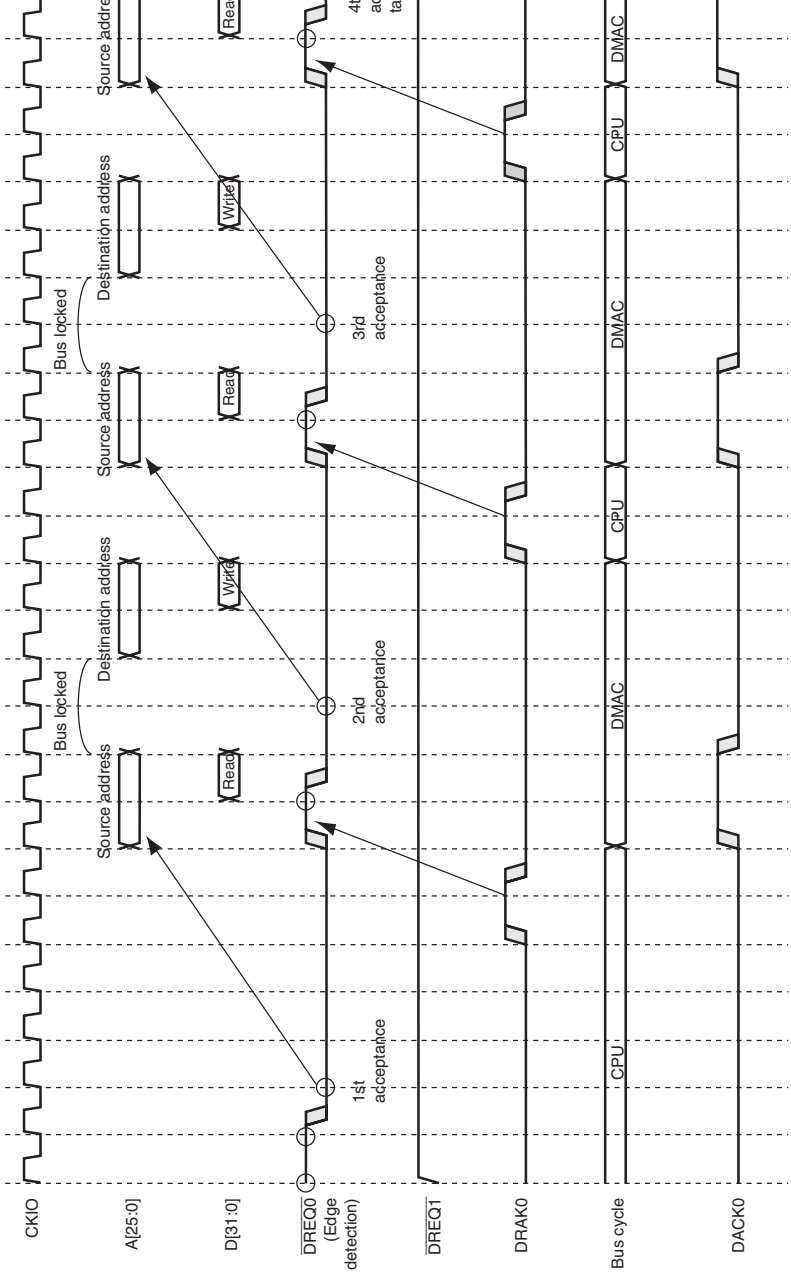


Figure 11.15 Dual Address Mode/Cycle Steal Mode in External Request 2-Channel Mode External Bus → External Bus/DREQ (Edge Detection), DACK (Read Cycle)

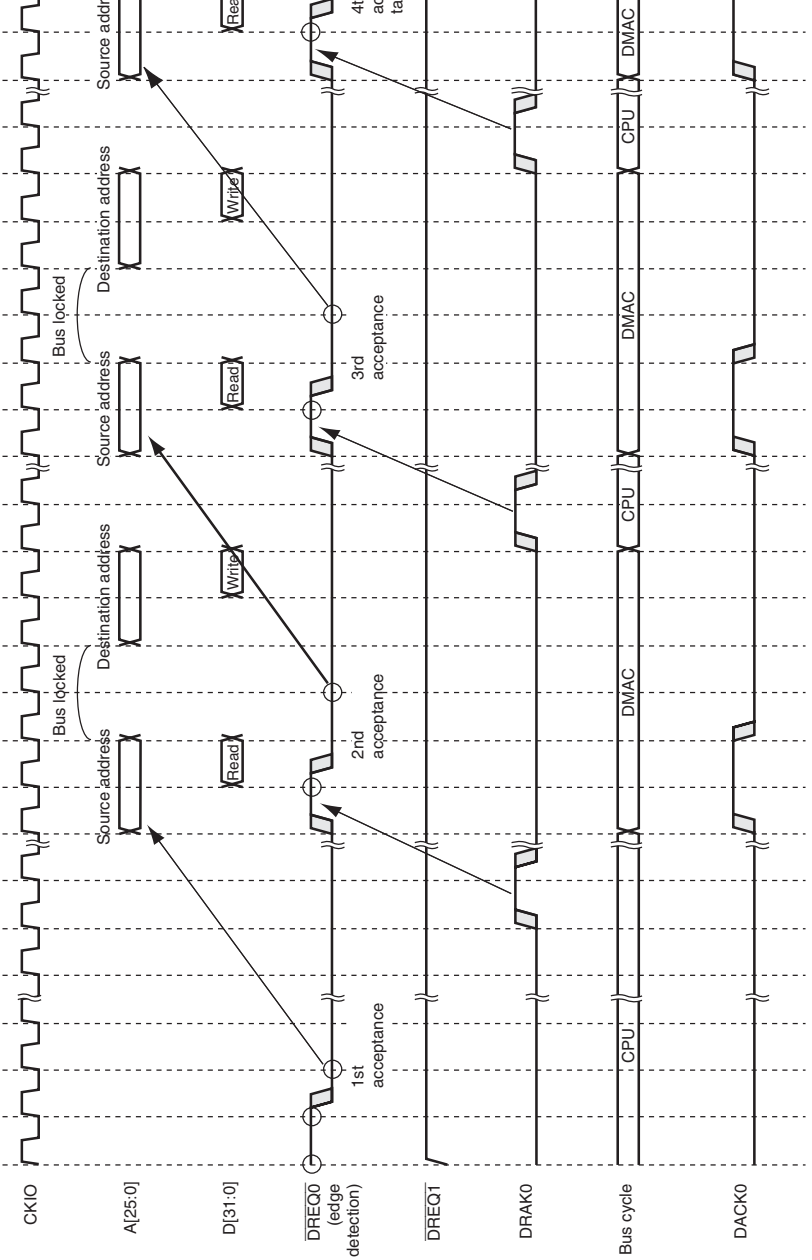


Figure 11.16 Dual Address Mode/Cycle Steal Mode in DMABRG Mode
External Bus → External Bus/$\overline{\text{DREQ}}$ (Edge Detection), DACK (Read Cycle)

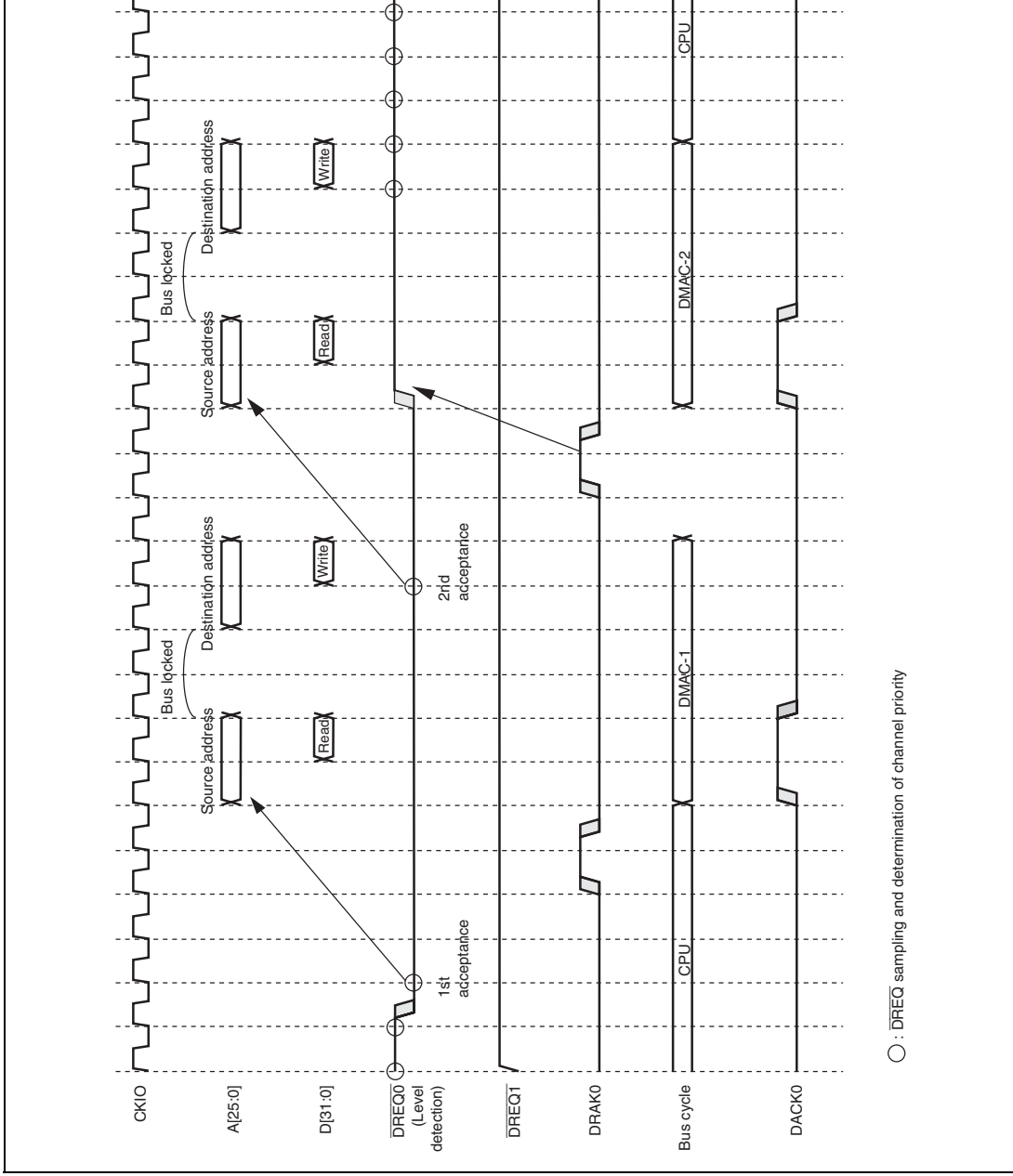


Figure 11.17 Dual Address Mode/Burst Mode in External Request 2-Channel Mode External Bus → External Device/$\overline{\text{DREQ}}$ (Level Detection), DACK (Read Cycle)

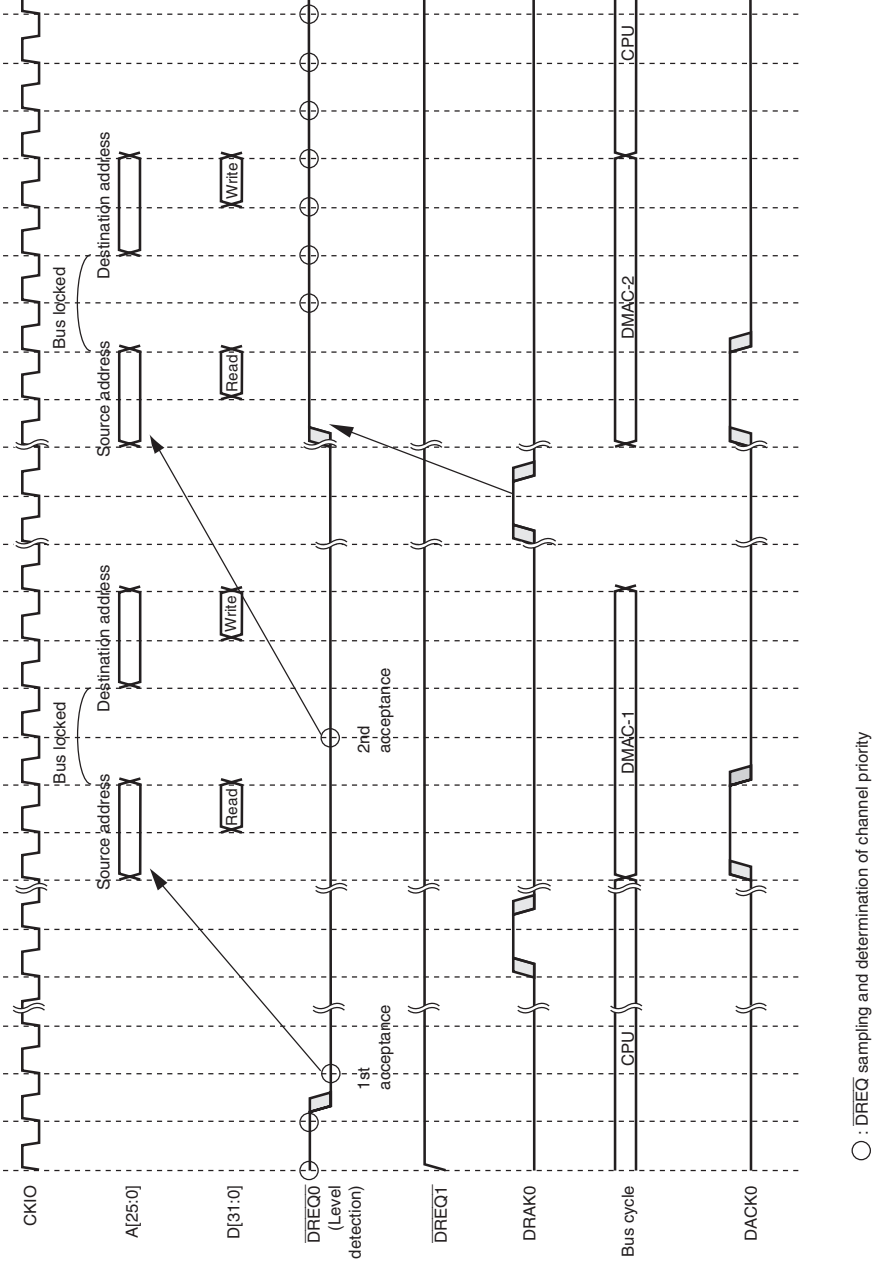


Figure 11.18 Dual Address Mode/Burst Modes in DMABRG Mode
External Bus → External Bus/ DREQ (Level Detection), DACK (Read Cycle)

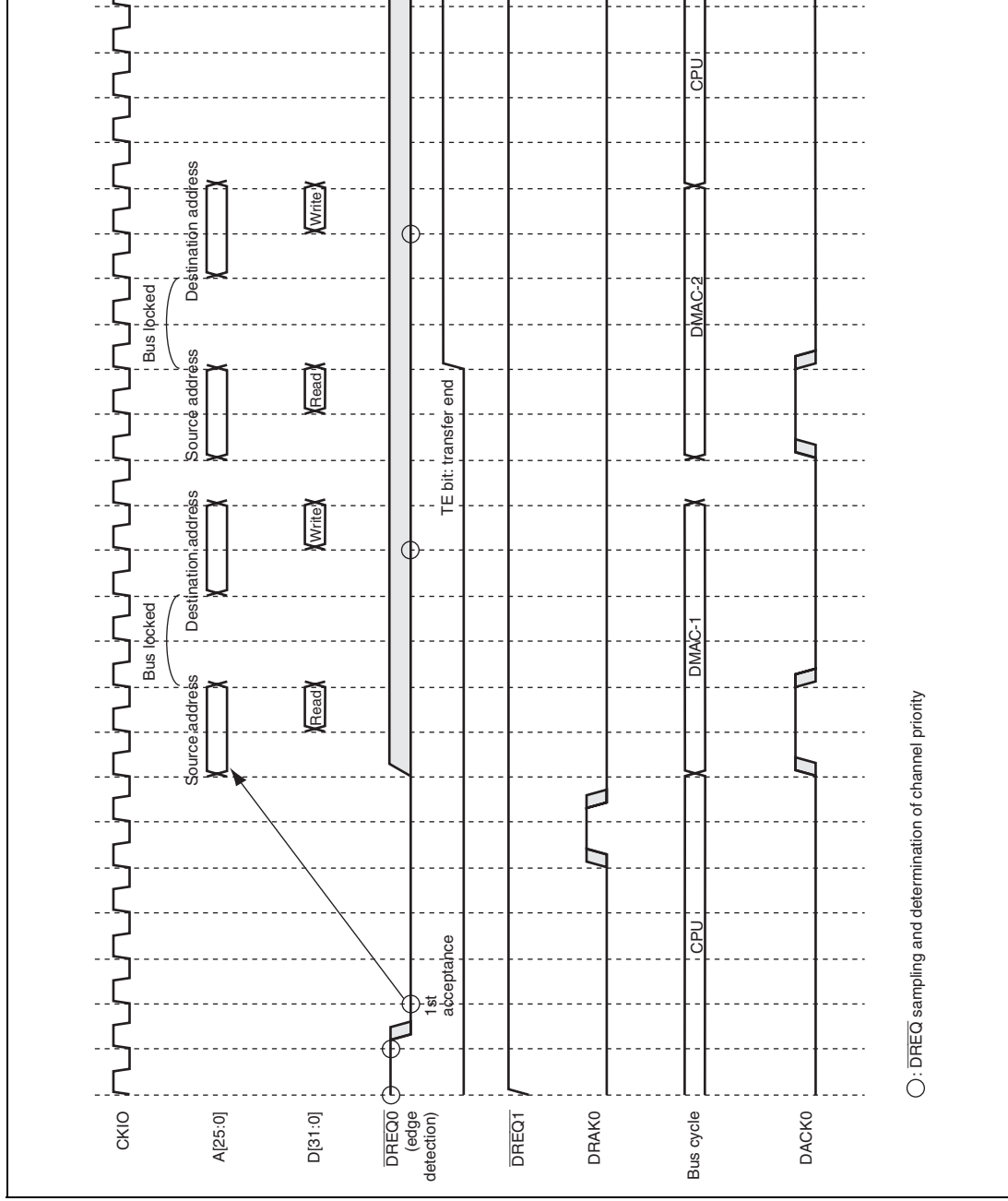


Figure 11.19 Dual Address Mode/Burst Mode in External Request 2-Channel Mode
External Bus → External Bus/DREQ (Edge Detection), DACK (Read Cycle)

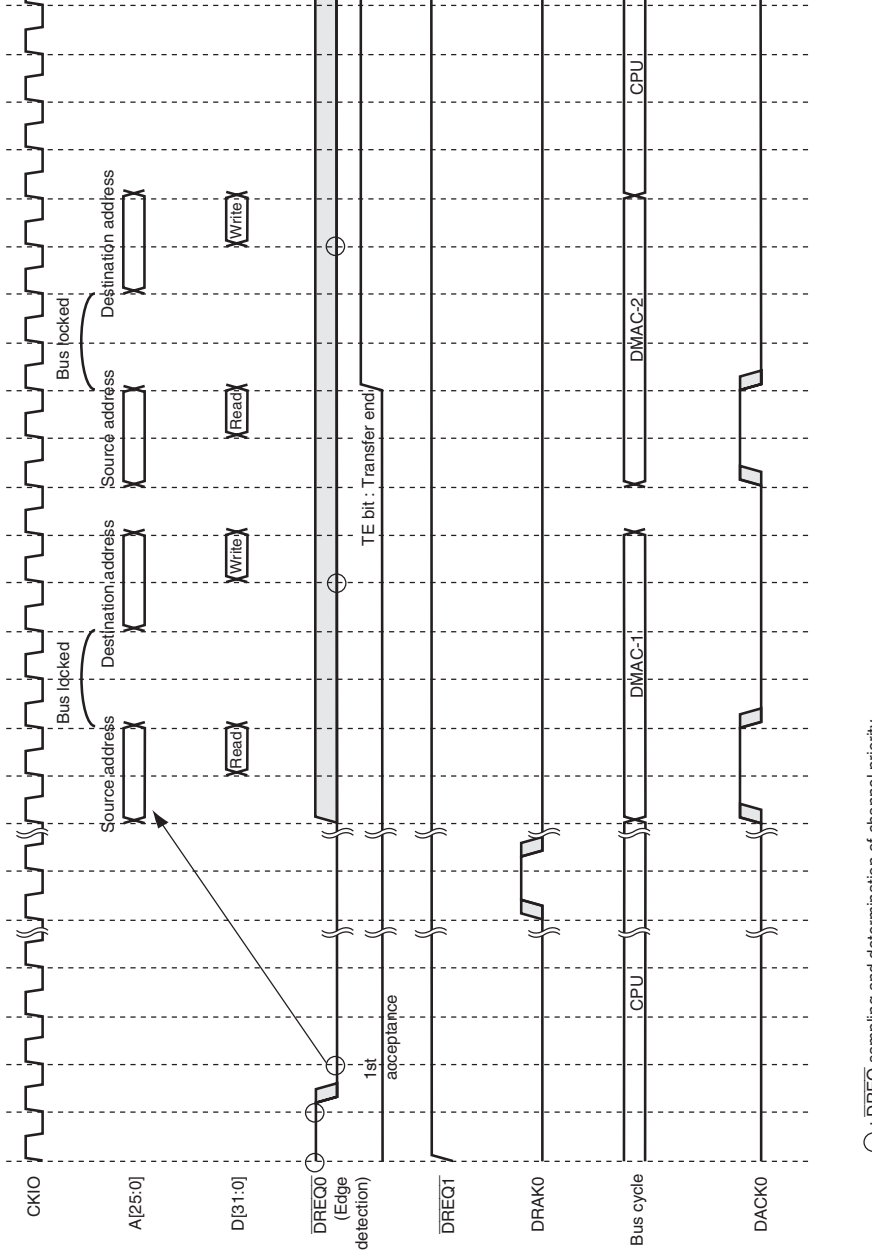


Figure 11.20 Dual Address Mode/Burst Modes in DMABRG Mode
External Bus → External Bus/DREQ (Edge Detection), DACK (Read Cycle)

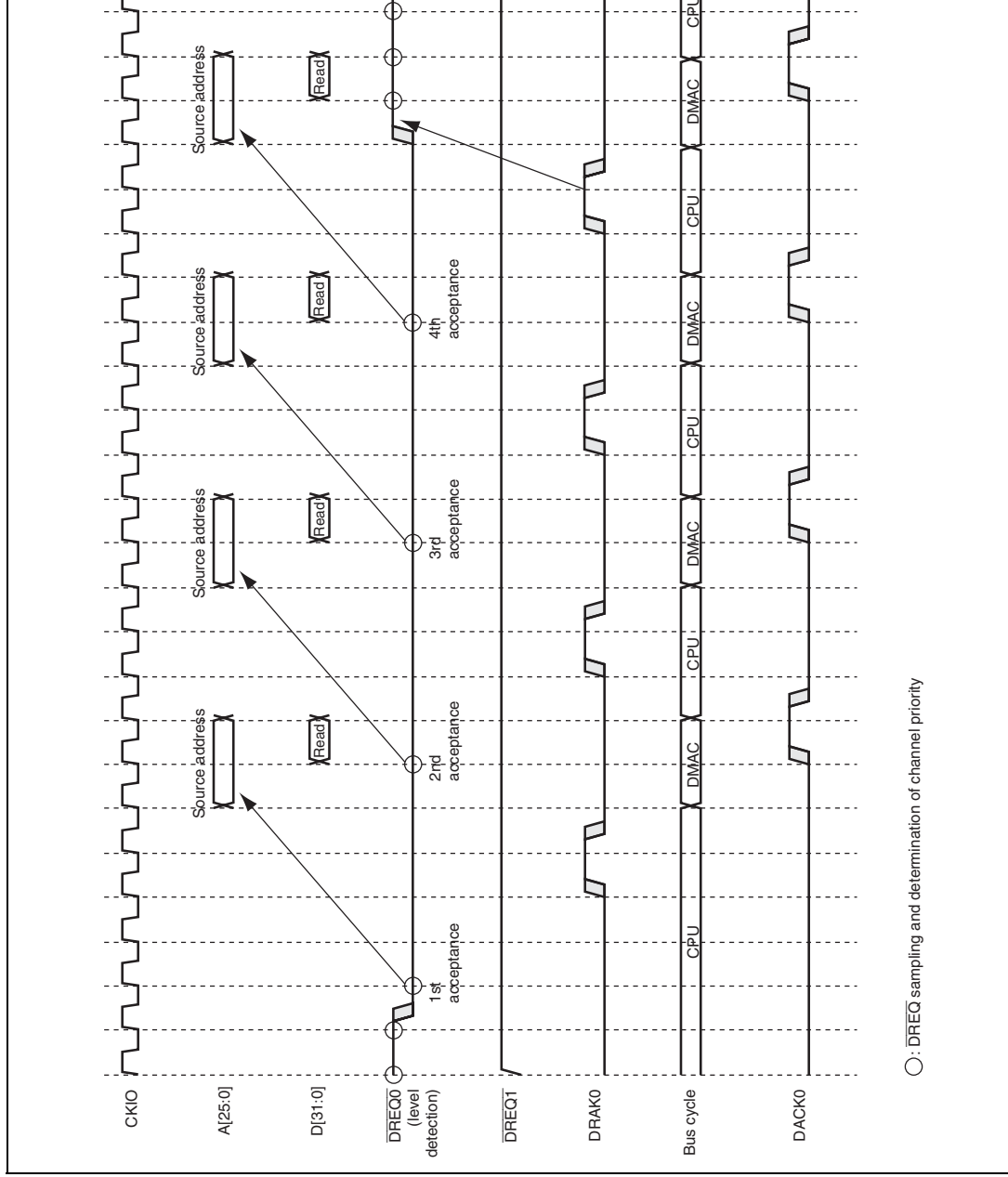


Figure 11.21 Single Address Mode/Cycle Steal Mode in External Request 2-Channel Mode External Bus → External Device/ $\overline{\text{DREQ}}$ (Level Detection)

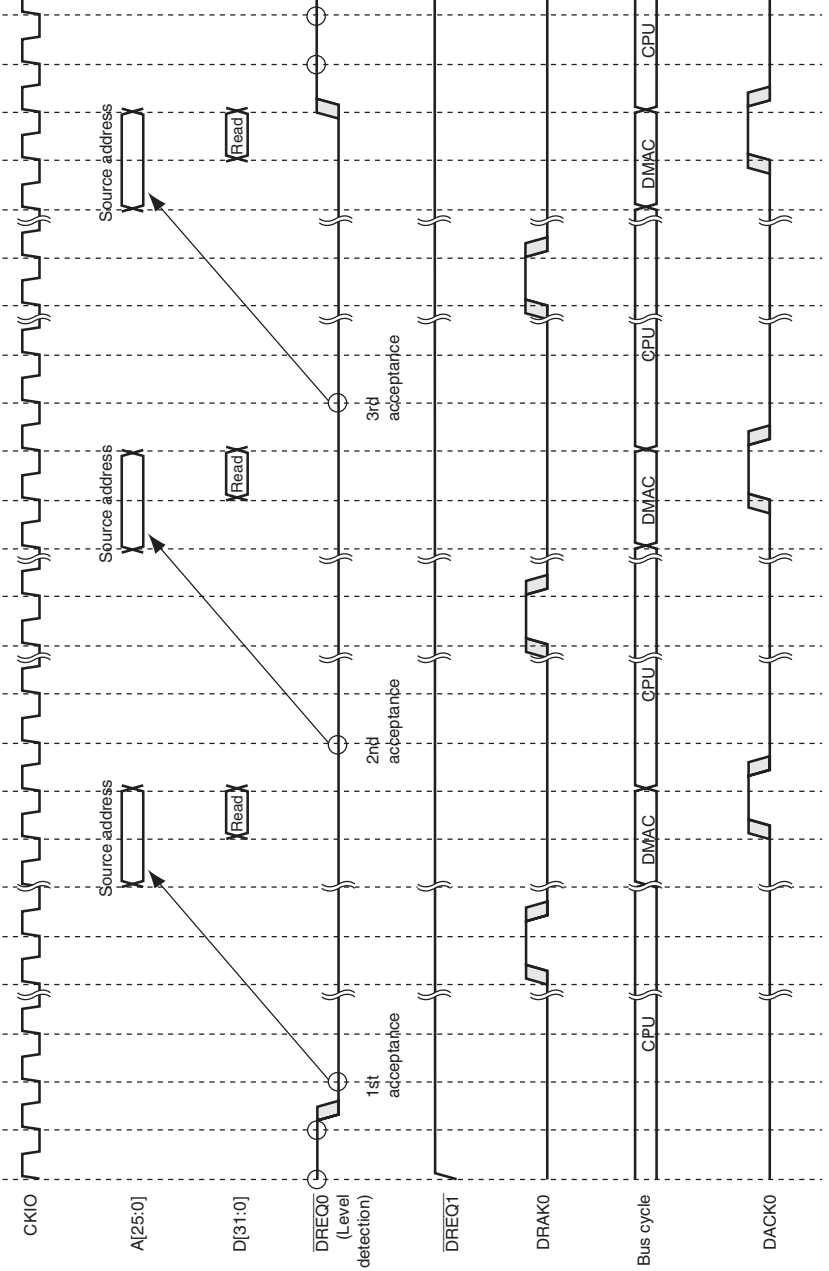


Figure 11.22 Single Address Mode/Cycle Steal Mode in External Request 2-Channel Mode External Bus → External Device/ DREQ (Level Detection)

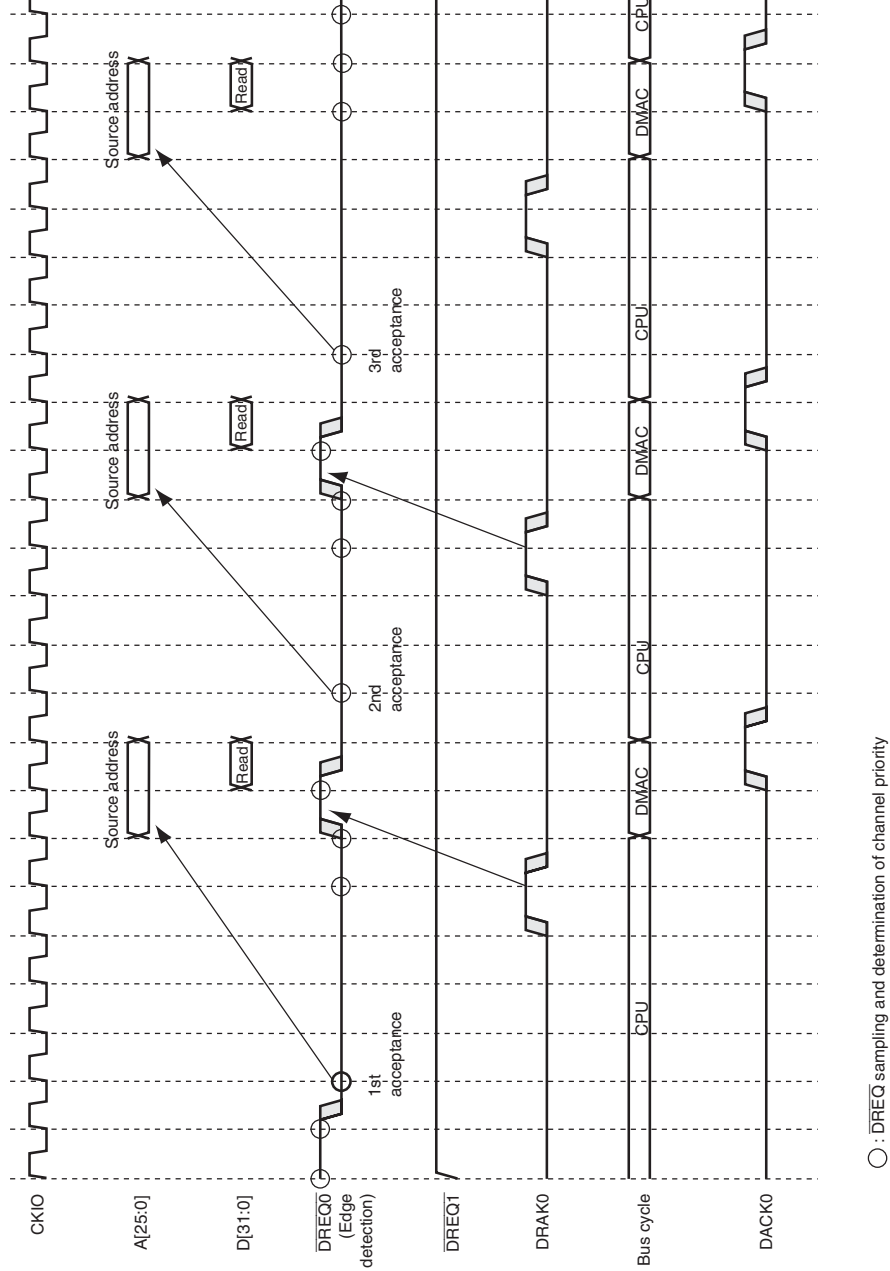


Figure 11.23 Single Address Mode/Cycle Steal Mode in External Request 2-Channel Mode External Bus → External Device/ $\overline{\text{DREQ}}$ (Edge Detection)

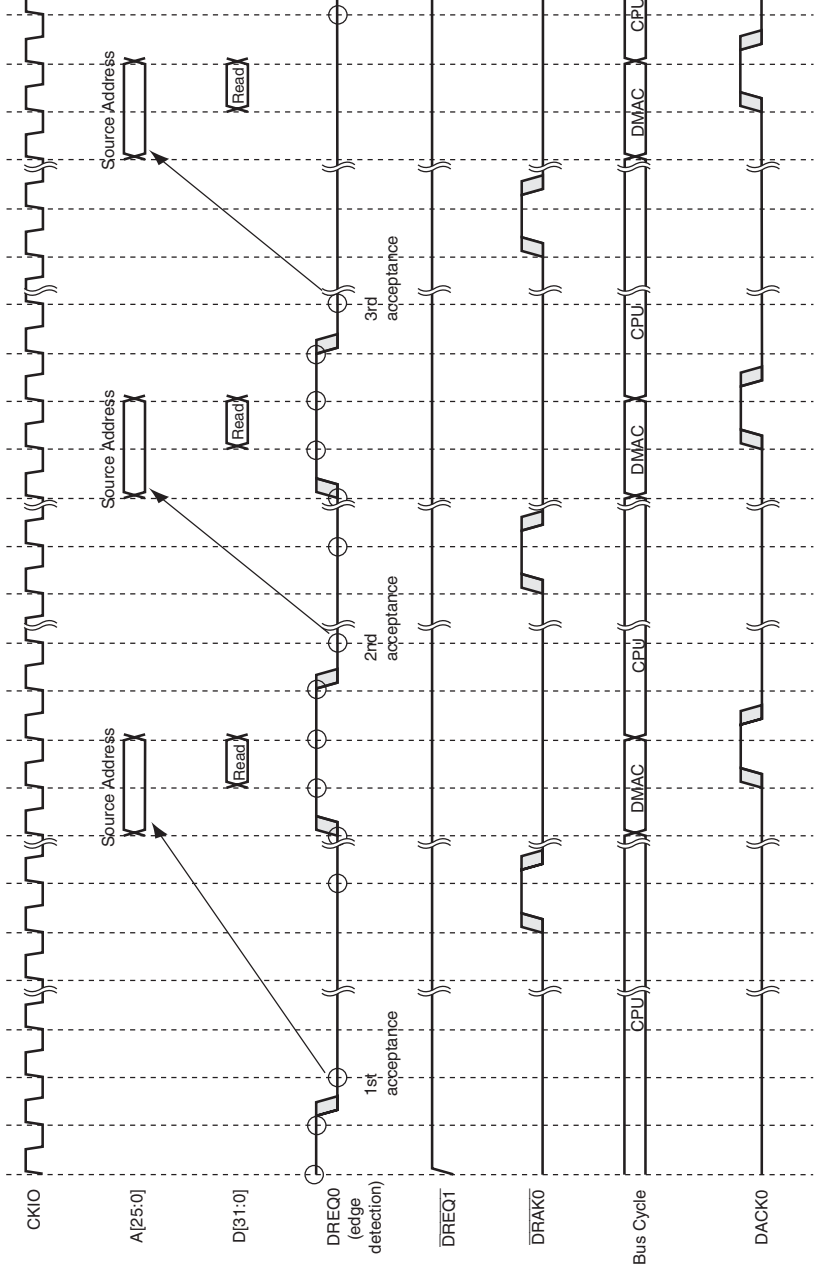


Figure 11.24 Single Address Mode/Cycle Steal Mode in DMABRG Mode
External Bus → External Device/ DREQ (Edge Detection)

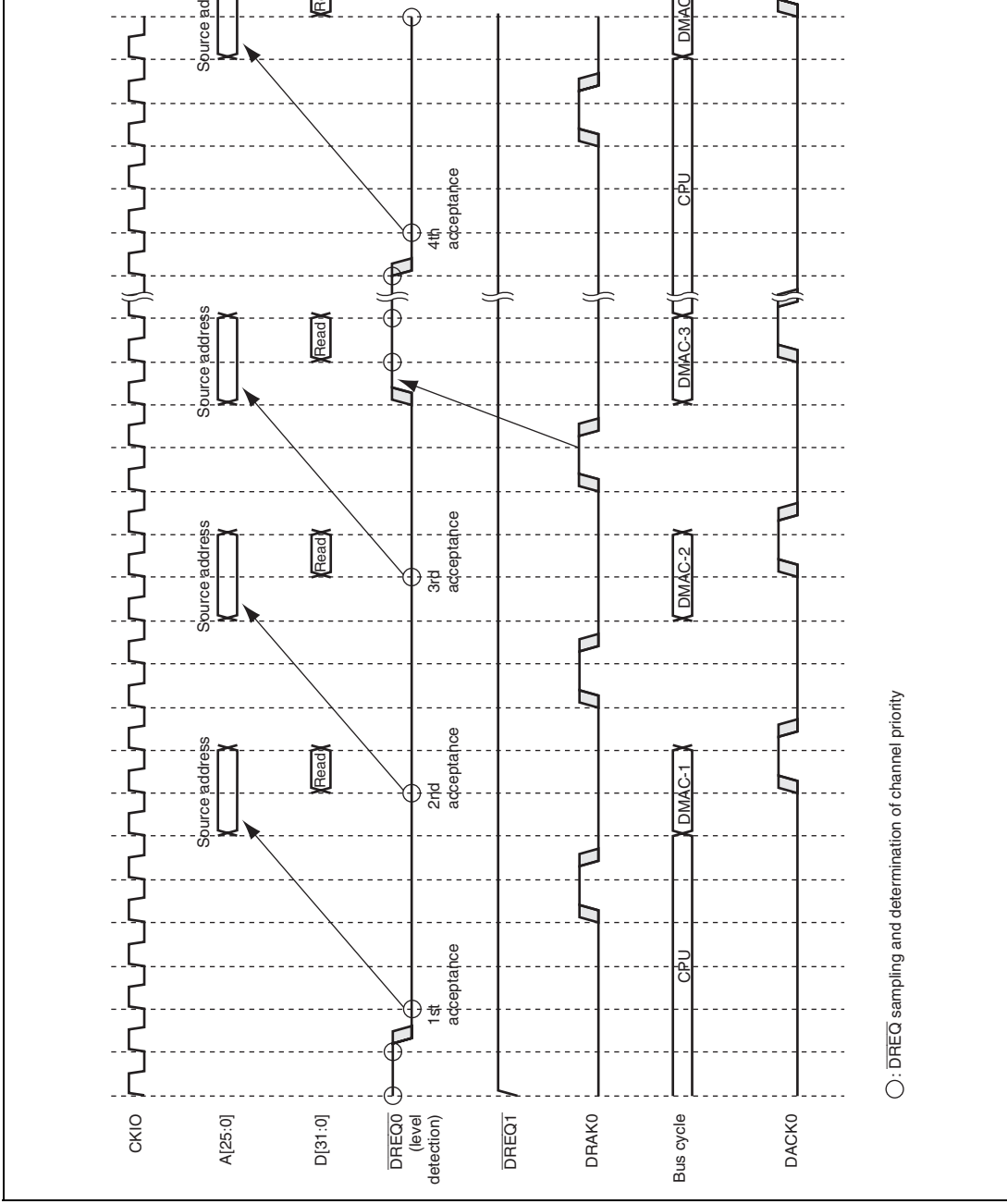


Figure 11.25 Single Address Mode/Burst Mode in External Request 2-Channel Mode
External Bus → External Device/ $\overline{\text{DREQ}}$ (Level Detection)

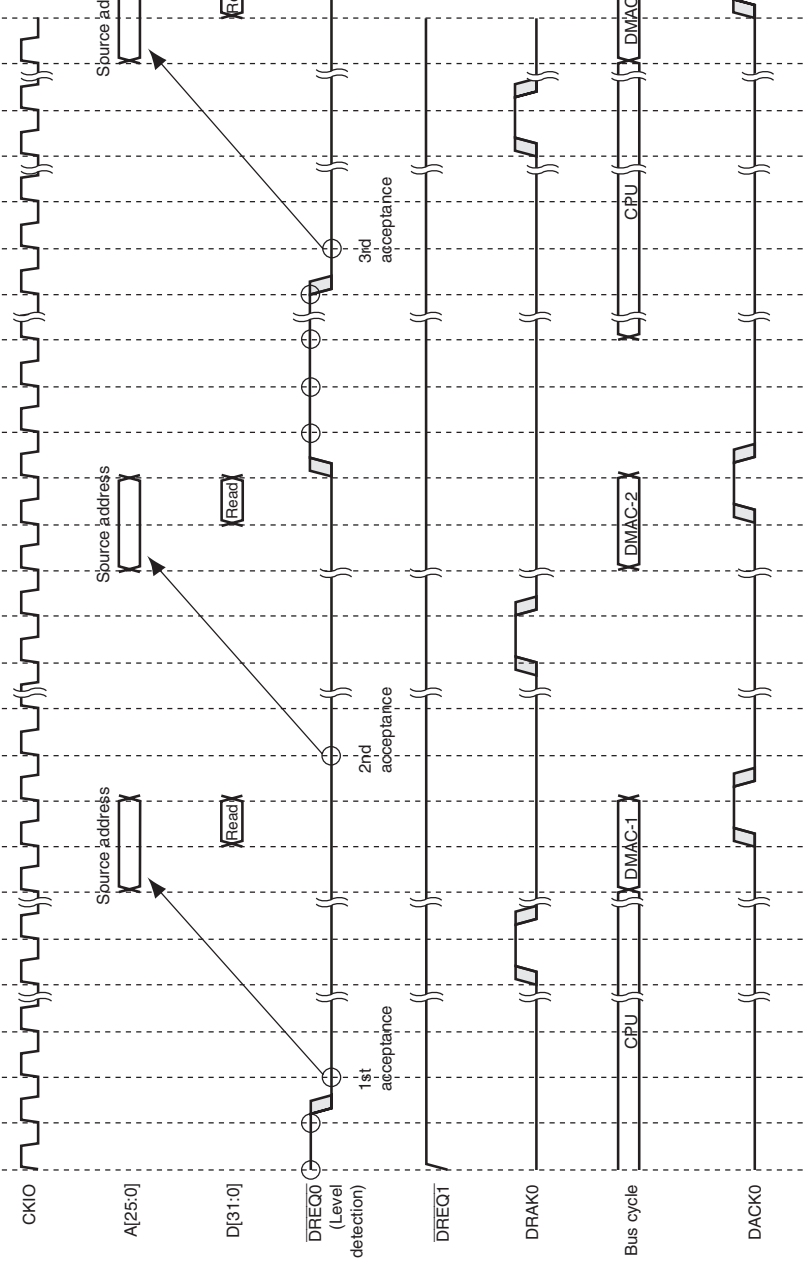


Figure 11.26 Single Address Mode/Burst Mode in DMABRG Mode
External Bus → External Device/ $\overline{\text{DREQ}}$ (Level Detection)

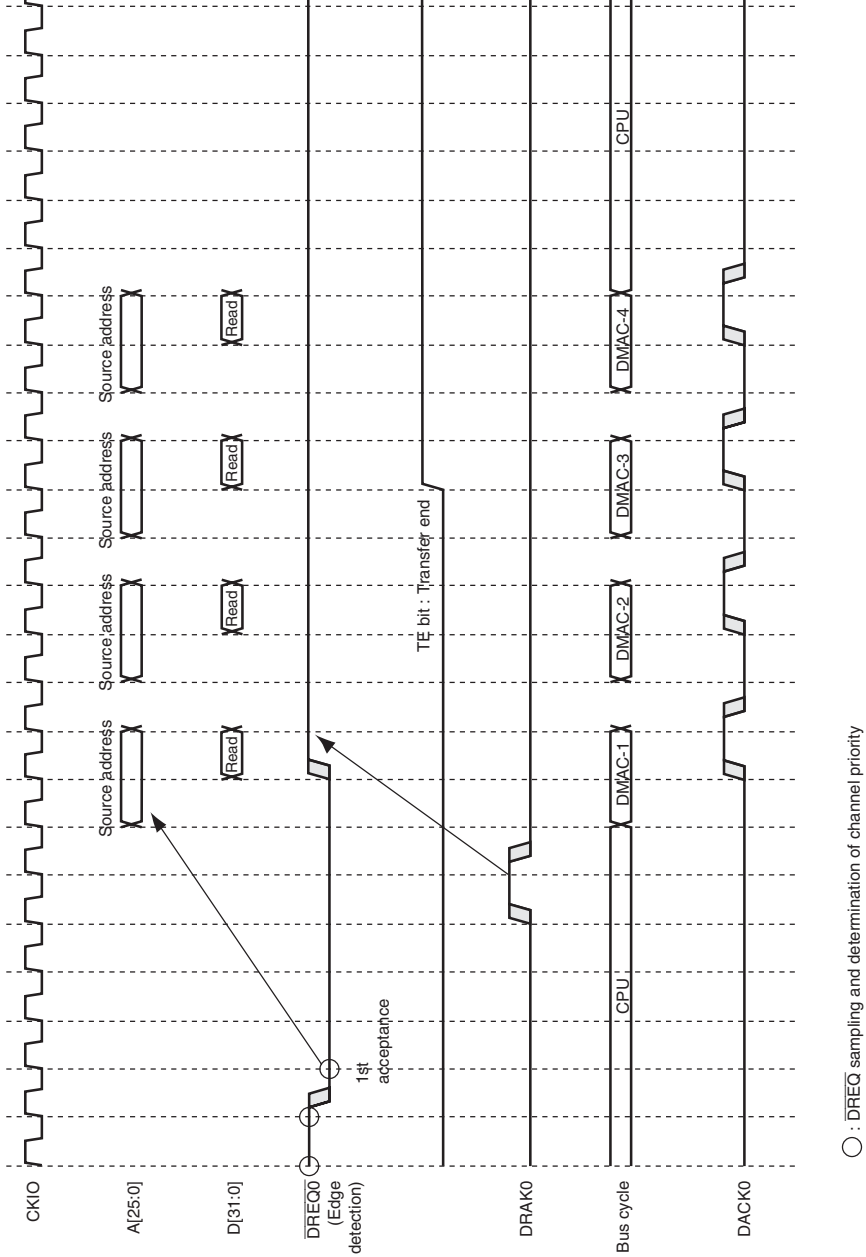


Figure 11.27 Single Address Mode/Burst Mode in External Request 2-Channel Mode External Bus → External Device/ $\overline{\text{DREQ}}$ (Edge Detection)

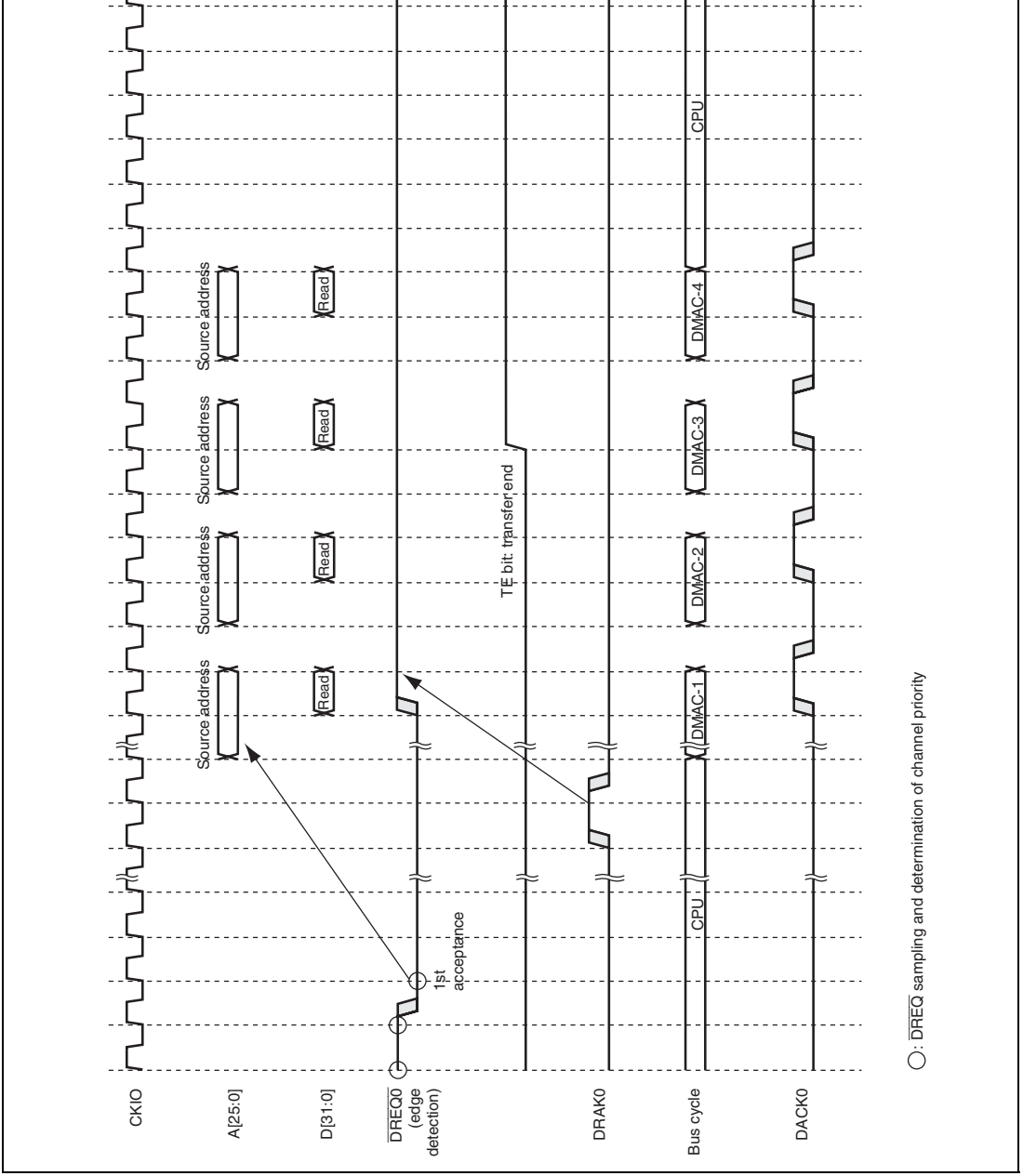


Figure 11.28 Single Address Mode/Burst Mode in DMABRG Mode
External Bus → External Device/ $\overline{\text{DREQ}}$ (Edge Detection)

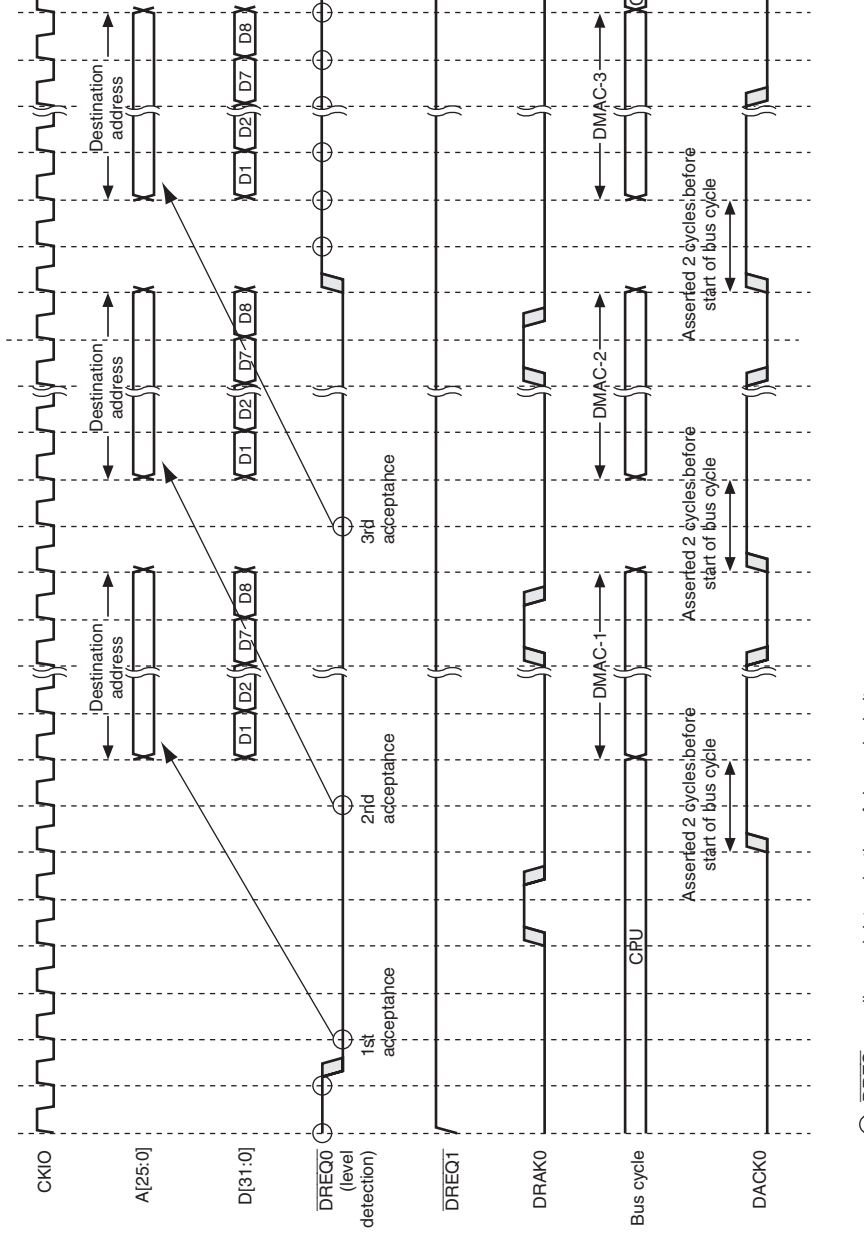


Figure 11.29 Single Address Mode/Burst Mode in External Request 2-Channel Mode External Device → External Bus/ $\overline{\text{DREQ}}$ (Level Detection)/32 Byte Block Transfer (Bus Width: 32 bits, SDRAM: row hit write)

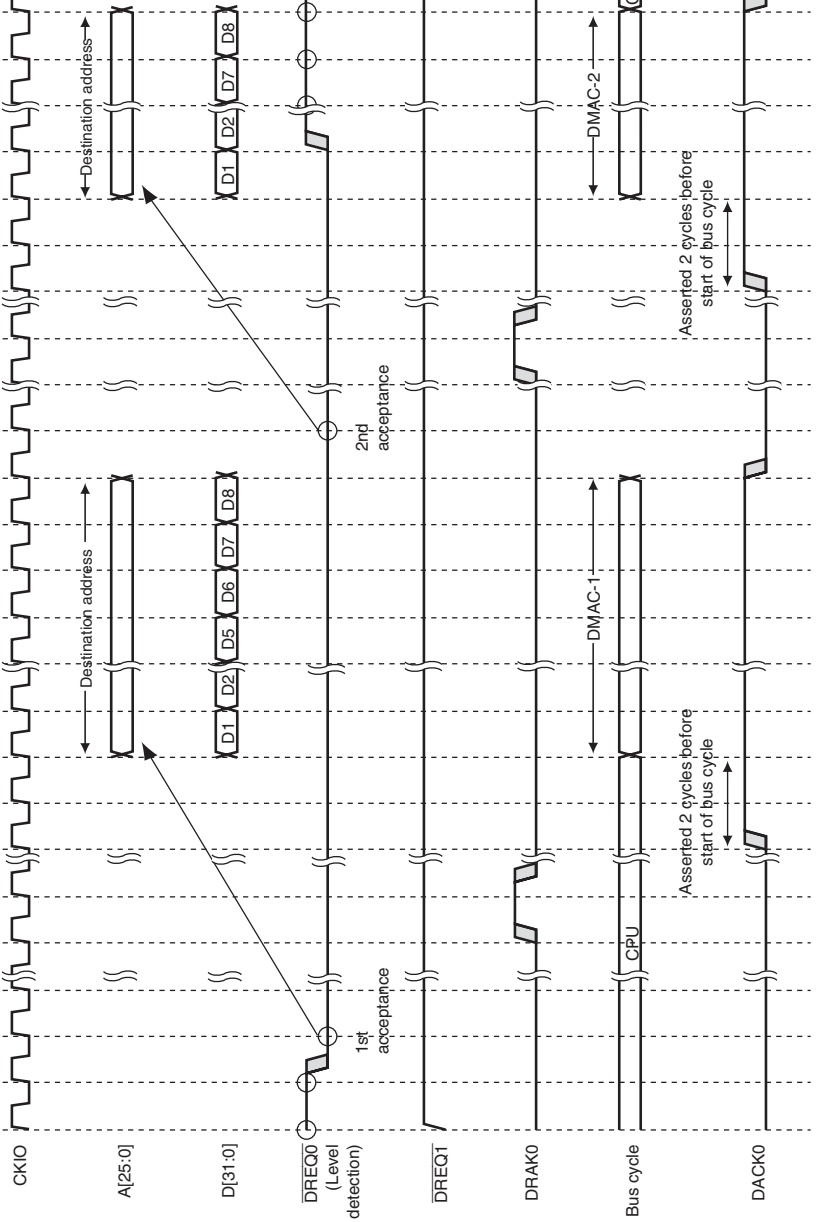


Figure 11.30 Single Address Mode/Burst Mode in DMABRG Mode
External Device → External Bus/ DREQ (Level Detection)/32 Byte Block Transfer
(Bus Width: 32 bits, SDRAM: row hit write)

The conditions for ending DMA transfer are different for ending on individual channels and for ending on all channels simultaneously. Following are the procedures for ending transfer, except for ending transfer when the DMATCR value reaches 0.

1. Cycle steal mode (external request, on-chip peripheral module request, auto-request)

   When transfer end conditions are met, the DMAC waits until all ongoing DMA transfers requested before transfer end conditions are complete, and then stops the operation. In cycle steal mode, the operation is the same for both edge and level transfer request detection.

2. Burst mode, edge detection (external request, DMABRG request, on-chip peripheral module request, auto-request)

   It generates the same delay between the time transfer end conditions are met and the time the DMAC stops the operation as in cycle steal mode. In burst mode with edge detection, only the first transfer request activates the DMAC, but the timing of stop request (DE = 0 in CHCR, DME = 0 in DMAOR) sampling is the same as the transfer request sampling timing shown in Burst Mode, Single Address Mode, Edge Detection and Suspension of DMA Transfer with $\overline{\text{DREQ}}$ Level Detection under Operation in section 11.4.5 (3) Operation. Therefore, a transfer request is regarded as having been issued until a stop request is detected, and the corresponding processing is executed before the DMAC stops.

3. Burst mode, level detection (external request)

   It generates the same delay between the time transfer end conditions are met and the time the DMAC stops the operation as in cycle steal mode. As in the case of burst mode with edge detection, the timing of stop request (DE = 0 in CHCR, DME = 0 in DMAOR) sampling is the same as the transfer request sampling timing shown in Burst Mode, Single Address Mode, Edge Detection and Suspension of DMA Transfer in Case of $\overline{\text{DREQ}}$ Level Detection under Operation in section 11.4.5 (3) Operation. Therefore, a transfer request is regarded as having been issued until a stop request is detected, and the corresponding processing is executed before the DMAC stops.

4. Bus timing for transfer suspension

   The DMAC suspends the operation after processing for one bus cycle unit is complete. In dual address mode transfer, the DMAC executes write cycle processing even if a transfer end condition is satisfied during the read cycle. It suspends the operation after completing the transfers mentioned above in 1, 2, and 3.

Transfer ends on the corresponding channel when either of the following conditions is satisfied:

- The DMATCR value reaches 0.
- The DE bit in CHCR is cleared to 0.

1. End of transfer when DMATCR = 0

   When the DMATCR value reaches 0, the DMAC terminates DMA transfer on the corresponding channel and sets the TE bit in CHCR. If the IE bit is set at this time, an interrupt (DMTE) request is sent to the CPU (an interrupt (DMTE) request can not be sent to the CPU for a DMA transfer end when DMATCR = 0 in a DMABRG request).

   Transfer ending with DMATCR = 0 does not follow the procedures described in 1, 2, 3, and 4 in section 11.4.6.

2. End of transfer when DE = 0 in CHCR

   When the DE bit in CHCR is cleared to 0, DMA transfer is suspended on the corresponding channel. (During a DMA transfer in a DMABRG request, do not clear the DE bit to 0 by accessing from the CPU.) The TE bit is not set in this case. Transfer ending in this case follows the procedures described in 1, 2, 3, and 4 in section 11.4.6.

(2) Conditions for Ending Transfer Simultaneously on All Channels

Transfer ends on all channels simultaneously when either of the following conditions is satisfied:

- The AE or NMIF bit in DMAOR is set to 1.
- The DME bit in DMAOR is cleared to 0.

1. End of transfer with DMAOR.AE = 1

   If the AE bit in DMAOR is set to 1 due to an address error, DMA transfer is suspended on all channels in accordance with the procedures in 1, 2, 3, and 4 in section 11.4.6, and the bus is passed to the CPU. Therefore, when the AE bit is set to 1, SAR, DAR, and DMATCR values indicate the addresses for the DMA transfer to be performed next and the remaining number of transfers. The TE bit is not set to 1 in this case. To resume DMA transfer, first correct the channel settings that caused the address error. Next re-specify DMARSRA/DMARSRB even if there is no change in resource. After that read AE = 1 and then write AE = 0. Acceptance of external requests is suspended while the AE bit is set to 1, so a DMA transfer request must be reissued when resuming transfer. Acceptance of on-chip peripheral module requests is also suspended, so when resuming transfer, the DMA transfer request enable bit for the relevant on-chip peripheral module must be cleared to 0 before the new setting is made. DMABRG must be reset for DMABRG requests. See section 11.6.2, DMABRG Reset for the procedure.

If the NMIF bit in DMAOR is set to 1 due to an NMI interrupt, DMA transfer is suspended on all channels in accordance with the procedures in 1, 2, 3, and 4 in section 11.4.6, and the bus is passed to the CPU. Therefore, when NMIF is set to 1, the SAR, DAR, and DMATCR values indicate the addresses for the DMA transfer to be performed next and the remaining number of transfers. The TE bit is not set to 1 in this case. To resume DMA transfer after NMI interrupt handling is completed, first re-specify DMARSRA/DMARSRB even when there is no resource change. After that read NMIF = 1 and then write NMIF = 0. Acceptance of external requests is suspended while the NMIF bit is set to 1, so a DMA transfer request must be reissued when resuming transfer. Acceptance of on-chip peripheral module requests is also suspended, so when resuming transfer, the DMA transfer request enable bit for the relevant on-chip peripheral module must be cleared to 0 before the new setting is made. DMABRG must be reset for DMABRG requests. See section 11.6.2, DMABRG Reset for the procedure.

3. End of transfer with DMAOR.DME = 0

   If the DME bit in DMAOR is cleared to 0, DMA transfer is suspended on all channels in accordance with the procedures in 1, 2, 3, and 4 in section 11.4.6, and the bus is passed to the CPU. The TE bit is not set to 1 in this case. When the DME bit is cleared to 0, the SAR, DAR, and DMATCR values indicate the addresses for the DMA transfer to be performed next and the remaining number of transfers. When resuming transfer, set DME to 1. Operation will then be resumed from the next transfer.

(3) Notes on Transfer End

When DMA transfer ends, requests may be retained in DMAC. Following are examples of cancellation of requests retained in DMAC.

- External requests

  See item 5 in External Request Acceptance Conditions in section 11.4.2, DMA Transfer Requests (2) External Request Mode.

- On-chip peripheral module requests

  Retained requests may be processed if DMA transfer occurs. If DMARCR.REXn = 1 when DMA transfer ends then external requests will be retained in DMAC. Examples of processing are shown below.

  1. After DMA transfer ends, set the corresponding resources in DMARSRA or DMARSRB to H'00. (Write H'80.)
  2. Read Bit REXn corresponding to the channel in DMARCR.

     REXn = 0: The DMAC has not accepted (retained) a transfer request. Go to 9.

     REXn = 1: The DMAC has accepted (retained) a transfer request. Go to 3.

4. Specify external address space in the corresponding channel SARn (the lower 6 bits are 32-bit boundary), and P4 address H'FE09 0020 in DARn.
5. Specify H'0000 0001 in DMATCRn of the corresponding channel.
6. Specify CHCRn.DM [1:0] and SM [1:0] = 00 in the corresponding channel, and CHCRn.RS [3:0] = 0111.
7. Setting CHCRn.TE = 0 for the corresponding channel carries out DMA transfer for requests that were retained in DMAC.
8. Confirm that the corresponding channel DMARCR.REXn = 0.

Note: When DMA transfer ends while DMAOR.AE = 1 or DMAOR.NMIF = 1, requests may be cleared even if DMARCR.REXn = 1. In that case, see "1. End of transfer with DMAOR.AE = 1" and "2. End of transfer with DMAOR.NMIF = 1" in (2) **Conditions for Ending Transfer Simultaneously on All Channels** of this section.

11.4.7 Interrupt-Request Codes

When the number of transfers specified in DMATCR has been finished and the IE bit in CHCR is set to 1, a transfer-end interrupt request can be sent to the CPU from each channel. Table 11.10 lists the interrupt-request codes that are associated with these DMAC interrupts.

Table 11.10 DMAC Interrupt-Request Codes

Interrupt Source	Description	INTEVT Code	Priority
DMTE0	CH0 transfer-end interrupt*	H'640	High
DMTE1	CH1 transfer-end interrupt	H'660	
DMTE2	CH2 transfer-end interrupt	H'680	
DMTE3	CH3 transfer-end interrupt	H'6A0	
DMTE4	CH4 transfer-end interrupt	H'780	
DMTE5	CH5 transfer-end interrupt	H'7A0	
DMTE6	CH6 transfer-end interrupt	H'7C0	
DMTE7	CH7 transfer-end interrupt	H'7E0	
DMAE	Address error interrupt	H'6C0	
DMABRGI0	USB address error interrupt	H'A80	
DMABRGI1	All data transfer end interrupt	H'AA0	
DMABRGI2	Half data transfer end interrupt	H'AC0	Low

Note: * A CH0 transfer-end interrupt cannot be generated when the DMABRG in DMABRG mode is used.

11.5.1 Examples of Transfer between External Memory and an External Device with DACK

(1) External Request 2-Channel Mode

Examples of transfer of data in external memory to an external device with DACK using DMAC channel 1 in external request 2-channel mode are considered here.

Table 11.11 (1) shows the transfer conditions and the corresponding register settings.

Table 11.11 (1) Conditions for Transfer between External Memory and External Device with DACK, and Corresponding Register Settings

Transfer Condition	Register	Setting
Transfer source: external memory	SAR1	H'0C00 0000
Transfer destination: external device with DACK	DAR1	(Accessed by DACK)
Number of transfers: 32	DMATCR1	H'0000 0020
Transfer source address: decremented	CHCR1	H'0000 22A5*
Transfer destination address: (setting invalid)		
Transfer request source: external pin ($\overline{\text{DREQ1}}$) edge detection		
Bus mode: burst		
Transfer unit: word		
No interrupt request at end of transfer		
External request 2-channel mode	DMAOR	H'0000 0201
Channel priority order: 2 > 0 > 1 > 3 > 4 > 5 > 6 > 7		

Note: * When $\overline{\text{DREQ1}}$ is specified as a DMA transfer request source in external request 2-channel mode, only channel 1 accepts the DMA transfer request ($\overline{\text{DREQ0}}$ is accepted only by channel 0).

Examples of data transfer from external memory to an external device with DACK using DMAC channel 1 in DMABRG mode are considered here.

Table 11.11 (2) shows the transfer conditions and the corresponding register settings.

Table 11.11 (2) Conditions for Transfer between External Memory and External Device with DACK, and Corresponding Register Settings

Transfer Condition	Register	Setting
Transfer source: external memory	SAR1	H'0C00 0000
Transfer destination: external device with DACK	DAR1	(Accessed by DACK)
Number of transfers: 32	DMATCR1	H'0000 0020
Transfer source address: decremented	CHCR1	H'0000 22A5*[1] (H'0000 22AD*[2] when writing)
Transfer destination address: (setting invalid)		
Transfer request source: external pin ($\overline{\text{DREQ1}}$) edge detection	DMARCR	H'0003 0040
Bus mode: burst	DMARSRA	H'0011 0000 (H'0091 0000*[3] when writing)
Transfer unit: word		
Request reception priority: round robin		
No interrupt request at end of transfer		
DMABRG mode	DMAOR	H'0000 C201
Channel priority order: 2 > 0 > 1 > 3 > 4 > 5 > 6 > 7		

Notes: 1. When $\overline{\text{DREQ0}}$ to $\overline{\text{DREQ3}}$ are specified as DMA transfer request sources in DMABRG mode, any channels can accept the requests (a limitation on the use of channels in DMABRG mode is only for a DMABRG request).
2. Always write 1 to the CHSET bit when modifying the CHCRn value in DMABRG mode.
3. Always write 1 to the CHnWEN bit of the corresponding channel when modifying the DMARSRA or DMARSRB value.

The DMABRG has independent FIFOs (32-bit 16-stage) for the LCDC, HAC, SSI, and USB with which it performs DMA transfers between the LCDC, HAC, SSI, and USB and synchronous DRAM. The DMABRG transfers a maximum of 32-byte data in a single DMA transfer.

11.6.1 DMABRG Request

DMA transfer by the DMABRG is performed using DMAC channel 0. The independent FIFOs (32-bit 16-stage) for the LCDC, HAC, SSI, and USB generate DMABRG requests. The LCDC, HAC, SSI, and USB that are connected to the DMABRG can operate at the same time.

CHCR0*, SAR0, and DAR0 are automatically set according to the LCDC or DMABRG register settings. CHCR0, SAR0, and DAR0 do not have to be set (rewritten) by the CPU.

Note: * If CHCR0.DE = 1 is set by the CPU, an address error may occur (DMAOR.AE = 1) and the DMAC will stop operation. When using DMABRG requests, make sure the CPU does not set CHCR0.DE = 1.

11.6.2 DMABRG Reset

The DMAC of this LSI suspends a DMA transfer when the following conditions are met.

(1) NMI interrupt occurred
(2) DMA address error occurred

When the DMAC suspends a DMA transfer by the above conditions while using the DMABRG (LCDC, HAC, SSI, or USB), reset the DMABRG (CHCR.CHSET = 1), re-specify the DMAC registers*, and then reactivate the DMAC.

Setting the BRGRST bit in DMAPCR to 1 resets the DMABRG. The reset is canceled by clearing the BRGRST bit to 0. Resetting the DMABRG forcibly terminates DMA transfer for the HAC, SSI, USB, or LCDC. In this case, a transfer end interrupt is not generated.

Resetting the DMABRG initializes the following registers to the state of a power-on reset.

- DMABRGCR
- DMAACR
- DMAUSAR
- DMAUDAR
- DMAURWSZ
- DMAUCR

- DMAATXSAR(0/1)
- DMAARXDAR(0/1)
- DMAATXTCR(0/1)
- DMAARXTCR(0/1)
- DMAATXTCNT(0/1)
- DMAARXTCNT(0/1)

Do not access the registers of the HAC, SSI, USB, LCDC, and DMAC (except for DMAPCR) while the BRGRST bit is 1. Operation is not guaranteed when these registers are accessed in this state.

Note: * Make sure to write 1 to the CHSET bit in CHCR0 before re-specifying the DMAC registers in the case of DMAC reactivation (DMA transfer will be resumed).

11.6.3 DMA Transfer Operating Mode for HAC and SSI

This LSI has two audio codec interfaces. The HAC and SSI are assigned to the audio codec interfaces. This assignment is selected by the IPSELR11 and IPSELR10 bits in IPSELR of the PFC. For details see section 24.2.35, Peripheral Module Select Register (IPSELR).

Figure 11.31 shows a configuration of the DMA for the HAC and SSI. This LSI transfers data by the DMA transfer request from the audio codec via DMAC channel 0. A transfer between synchronous DRAM and the audio codec is performed by using a 32-byte 2-stage FIFO for each interface.

Audio data for transfer is stored in the transmit/receive buffer of the synchronous DRAM. The transmit/receive buffer is defined by specifying the start address to DMAARXDAR or DMAATXSAR and the number of transfer bytes to DMAARXTCR or DMAATXTCR.

When half of the data is transferred (A0TXH, A0RXH, A1TXH, or A1RXH interrupt is used) or all data is transferred (A0TXE, A0RXE, A1TXE, or A1RXE interrupt is used), an interrupt can be generated. Double buffer control for audio data can be used by switching halved transmit/receive buffers.

DMAARXDAR, DMAATXSAR, DMAARXTCR, and DMAATXTCR have the auto-reload function. When the same buffer is repeatedly used in the auto-reload function, it is not necessary to re-specify the registers.

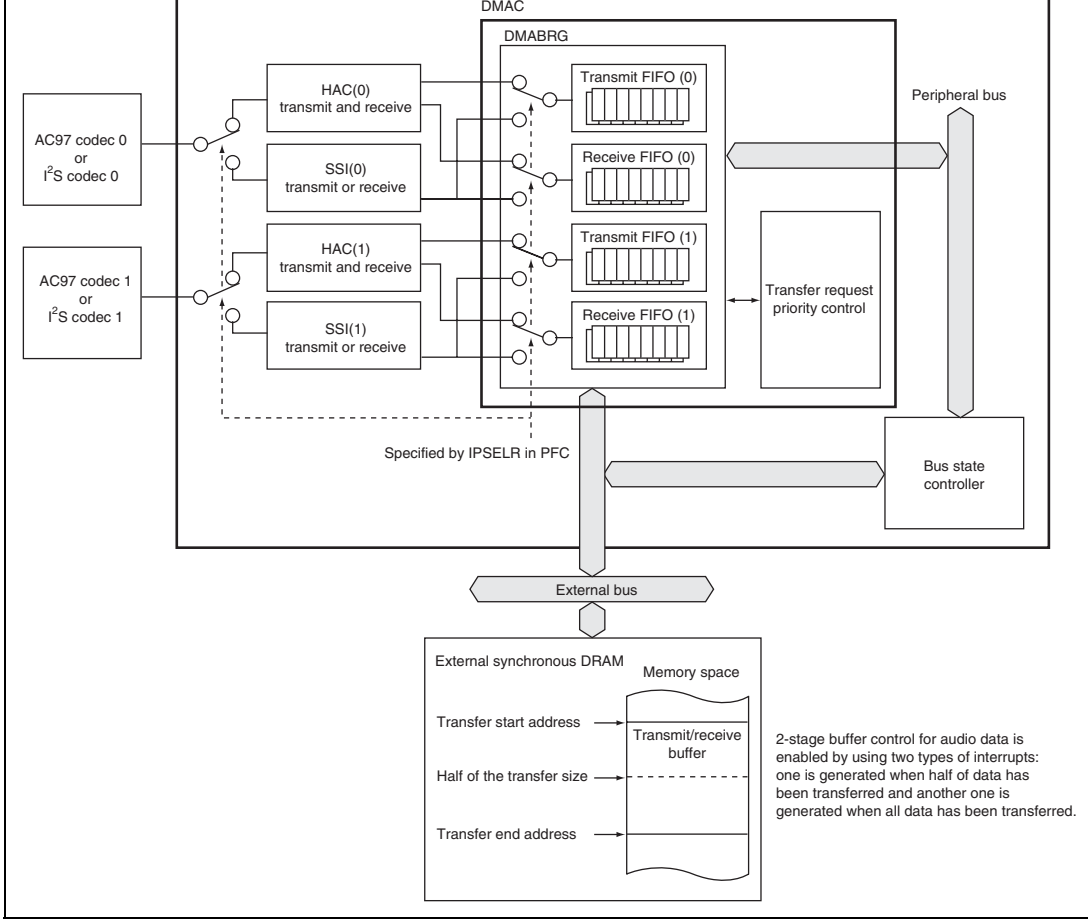


Figure 11.31 Configuration of DMA for HAC/SSI

To receive audio data in DMA transfer, specify DMABRG mode in bits DMS1 and DMS0 in DMAOR and a transfer request source and request acceptance priority order in DMARSRA and DMARCR. Then, specify the start address of a receive buffer storing receive audio data in DMAARXDAR and the number of transfer bytes in DMAARXTCR. Writing 1 to the RDE bit in DMAACR starts receiving the data.

11.6.5 DMA Audio Transmit Operation

To transmit audio data in DMA transfer, specify DMABRG mode in bits DMS1 and DMS0 in DMAOR and a transfer request source and request acceptance priority order in DMARSRA and DMARCR. Specify the start address of a transmit buffer storing transmit audio data in DMAATXSAR and the number of transfer bytes in DMAATXTCR. Writing 1 to the TDE bit in DMAACR starts transmitting the data.

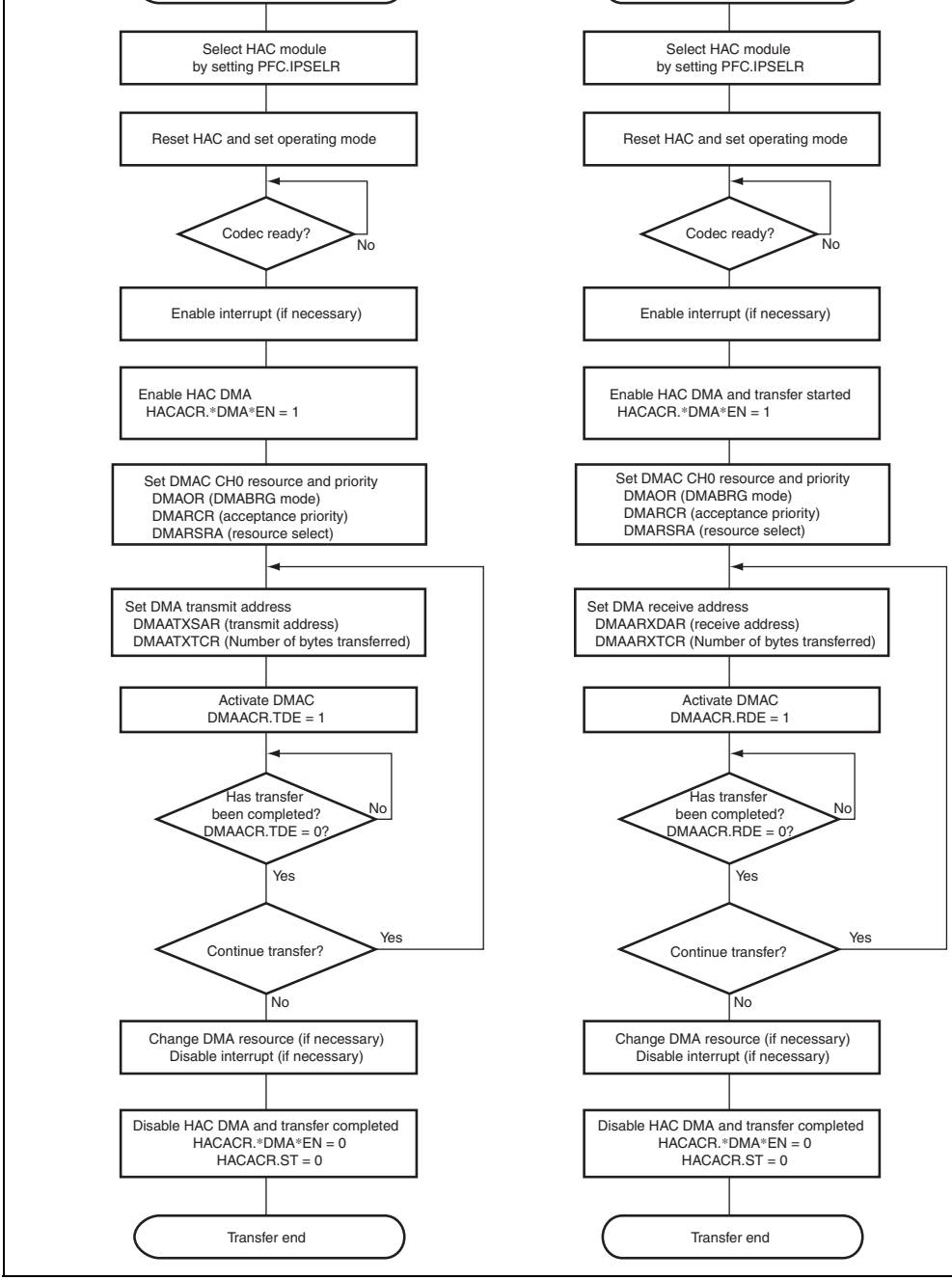


Figure 11.32 Example of HAC DMA Transfer Operation Flow

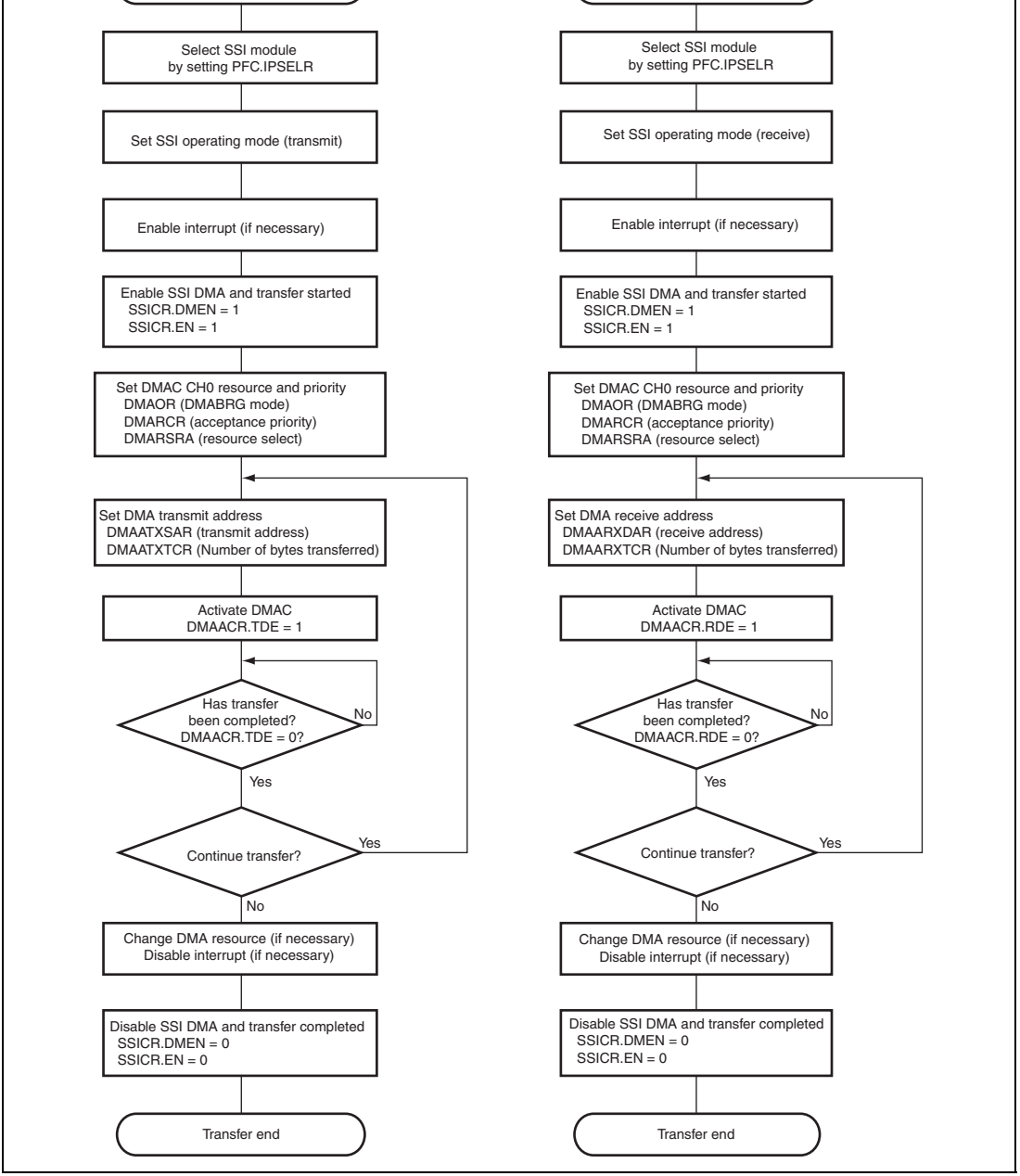


Figure 11.33 Example of SSI DMA Transfer Operation Flow

The DMAC stops a DMA transfer for the HAC or SSI when the transfer of data bytes specified by DMAATXTCR or DMAARXTCR is complete. When the transfer is complete, the settings specified before the transfer are read out from DMAARXDAR or DMAATXSAR. When restarting a transfer with the same start address and the same transfer bytes, write 1 to the DMA activation bit (RDE or TDE bit) in DMAACR to reactivate the DMAC. It is not necessary to re-specify the DMAARXDAR or DMAATXSAR value.

When the auto reload setting bit (RAR or TAR bit) in DMAACR is 1, the DMAC is automatically reactivated and performs transfers between the transmit/receive buffer and audio codec repeatedly.

To terminate a DMA transfer with the auto reload enabled, write 1 to the DMA forced termination bit (RDS or TDS bit) in DMAACR.

11.6.7 Forced Termination of DMA Audio Transfer

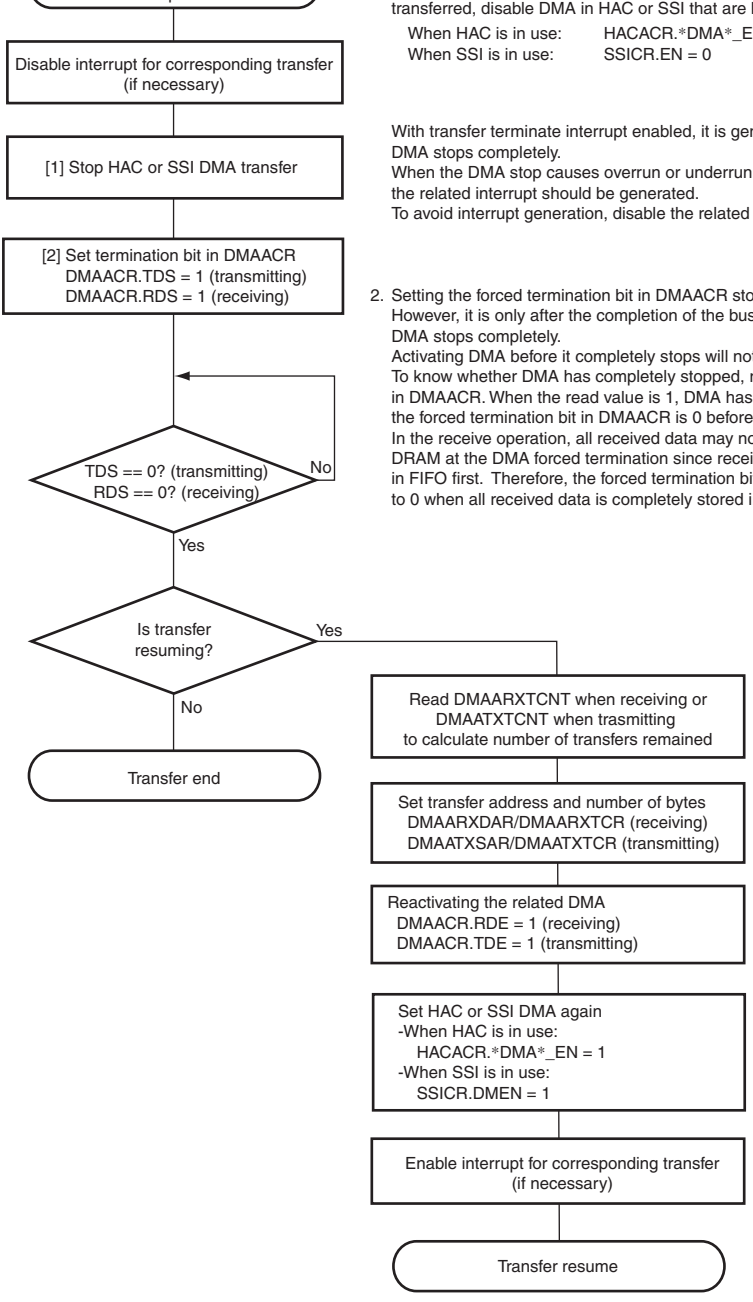
To forcibly terminate a DMA transfer while the transfer of data bytes specified by DMAATXTCR or DMAARXTCR is incomplete, write 1 to the DMA termination bit (RDS or TDS bit) in DMAACR. In a forced termination, a transfer end interrupt is also generated.

In a forced termination, the number of transfer bytes remaining on termination is indicated in DMAATXTCNT or DMAARXTCNT. The DMA audio transfer counter loads the DMAATXTCR or DMAARXTCR value on activation of the DMAC (when the RDE or TDE bit in DMAACR is set to 1) and is decremented every time a DMA transfer is performed. When resuming the DMA transfer after a forced termination, check the DMA transfer counter value for transfer progress, re-specify the start address and number of transfer bytes, and then reactivate the DMAC.

Since DMA audio data is transferred using FIFO, all received data may not be stored in the receive buffer at forced termination. When the DMA forced termination bit (RDS bit) in DMAACR is read as 0, the data is completely stored.

Re-specifying the registers with the RDS or TDS bit set to 1 does not activate the DMA. Clear the DMA enable bit of the SSI or HAC to 0 before forcibly terminating a transfer.

Figure 11.34 shows the forced termination procedure for the DMA audio transfer.



transferred, disable DMA in HAC or SSI that are being used.

When HAC is in use: HACACR.*DMA*_EN = 0
When SSI is in use: SSICR.EN = 0

With transfer terminate interrupt enabled, it is generated when the terminated DMA stops completely.
When the DMA stop causes overrun or underrun in HAC or SSI, the related interrupt should be generated.
To avoid interrupt generation, disable the related interrupts beforehand.

2. Setting the forced termination bit in DMAACR stops DMA in HAC or SSI. However, it is only after the completion of the bus cycle being performed that DMA stops completely.
Activating DMA before it completely stops will not take effect.
To know whether DMA has completely stopped, read the forced termination bit in DMAACR. When the read value is 1, DMA has not stopped. Make sure that the forced termination bit in DMAACR is 0 before activating DMA again.
In the receive operation, all received data may not be stored in synchronous DRAM at the DMA forced termination since received data is temporarily stored in FIFO first. Therefore, the forced termination bit in DMAACR will be cleared to 0 when all received data is completely stored in synchronous DRAM.

Figure 11.34 Forced Termination and Resume Procedures for DMA Audio Transfer

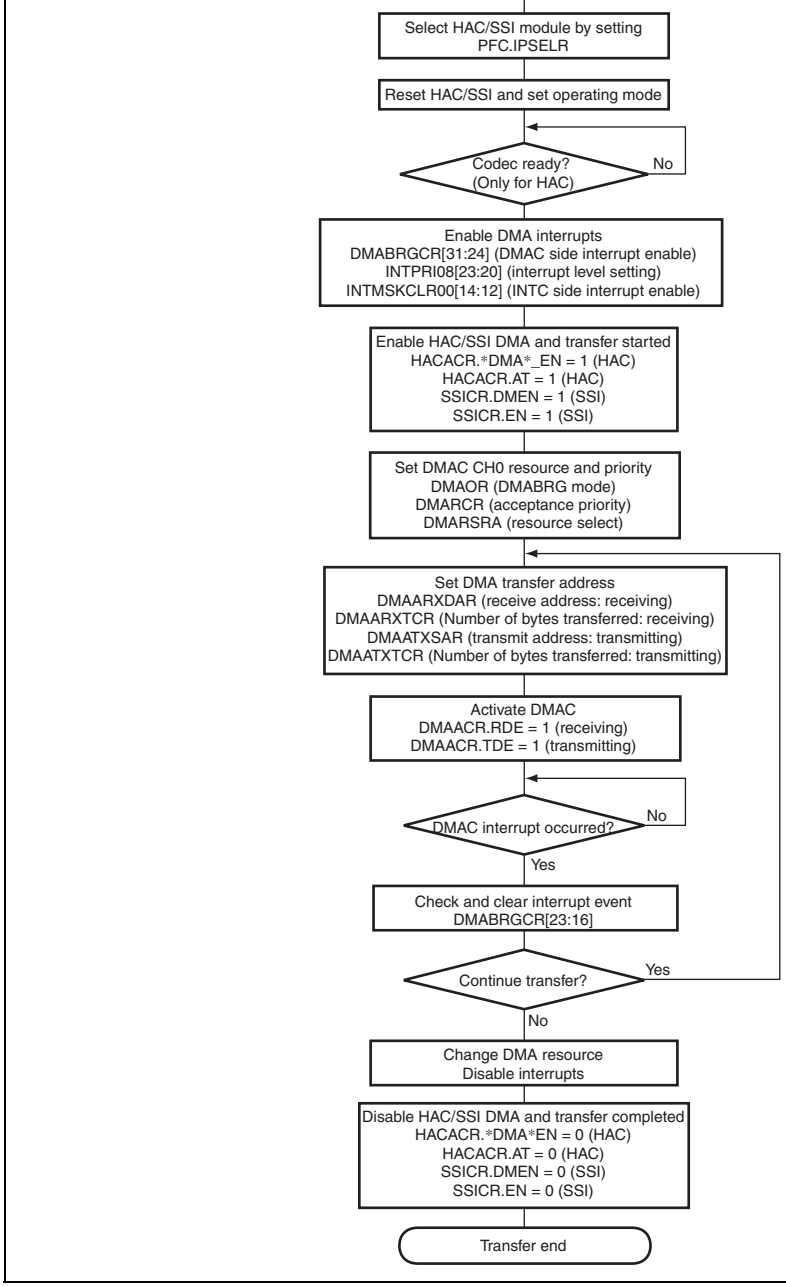


Figure 11.35 HAC/SSI DMA Transfer Operation Flow Using an Interrupt

There are two types of transfer end interrupts.

- A0TXH, A0RXH, A1TXH, or A1RXH (half data transfer end interrupt)
  An interrupt is generated when a half of the transfer size specified by DMAARXTCR or DMAATXTCR is completed.*[1]*[2]
- A0TXE, A0RXE, A1TXE, or A1RXE (all data transfer end interrupt)
  An interrupt is generated when the whole transfer size specified by DMAARXTCR or DMAATXTCR is completed.

Using half data transfer end and all data transfer end interrupts makes transfers for consecutive audio data efficient since half of the transmit/receive buffer can be accessed by the CPU during a transfer of the other half.*[3]

In addition, by enabling the auto reload function (TAR/RAR bit in DMAACR), it is not necessary to re-specify the registers for the second transfer or later.

Notes: 1. When the transfer size specified in DMAARXTCR or DMAATXTCR is 4 bytes, a half data transfer end interrupt is not generated.
2. When the transfer size specified in DMAARXTCR or DMAATXTCR is 8n + 4 bytes (n is an integer = 1 or greater) (transfer count is an odd number), a half data transfer end interrupt is generated when n + 1 transfers are completed.
3. The DMABRG for the HAC or SSI has FIFOs that stores a maximum of 64-byte data that is pre-fetched on transmit. If a half of the transfer size of audio data is less than 64 bytes, the remaining data in the transmit/receive buffer may already be stored in the FIFO when a half data transfer end interrupt is generated. When using the double buffer control by a half data transfer end interrupt, configure the transmit/receive buffer in synchronous DRAM having the size of 128 bytes or more.

11.6.9 HAC/SSI Endian Conversion Function

Data is transferred between the HAC or SSI and a transmit/receive buffer in 32-bit (longword) units. When data less than 32 bits is transferred, the byte order of audio data in the transmit/receive buffer in synchronous DRAM may differ from the DMA transfer order, depending on the MD5 pin level which specifies the endian type.

When SSI handles the transfer of 8-bit (byte) audio data, data transfer starts from the least-significant byte as shown in figure 11.36: first the left channel data is input to or output from bits 7 to 0, secondly the right channel data is input to or output from bits 15 to 8, and then the next left channel data is input to or output from bits 23 to 16. Selecting big endian mode (MD5 = 0) requires the conversion for alignment such that the least-significant byte is stored in the highest address in the transmit/receive buffer in synchronous DRAM (DMAACR.TAM[1:0]/DMAACR.RAM[1:0] = 01).

Slot data	31-24	23-16	15-8	7-0
Transfer data	R1	L1	R0	L0

External bus	31-24	23-16	15-8	7-0
Address on memory side	+0	+1	+2	+3
TAM[1:0]/RAM[1:0] = 01 TAM[1:0]/RAM[1:0] = 00	L0	R0	L1	R1

Big endian (conversion needed)

Slot data	31-24	23-16	15-8	7-0
Transfer data	R1	L1	R0	L0

External bus	31-24	23-16	15-8	7-0
Address on memory side	+3	+2	+1	+0
TAM[1:0]/RAM[1:0] = 01 TAM[1:0]/RAM[1:0] = 00	R1	L1	R0	L0

Little endian (conversion not needed)

Figure 11.36 8-Bit Data Transfer for SSI

(2) 16-Bit Data Transfer for HAC/SSI

When HAC or SSI handles the transfer of 16-bit (word) audio data, data transfer starts from the least-significant word as shown in figure 11.37: first the left channel data is input to or output from bits 15 to 0 and then the right channel data is input to or output from bits 31 to 16. Selecting big endian mode (MD5 = 0) requires the conversion for alignment such that the least-significant word is stored in the highest address in a transmit/receive buffer in synchronous DRAM (DMAACR.TAM[1:0]/DMAACR.RAM[1:0] = 10).

Slot data	31-24	23-16	15-8	7-0
Transfer data	R		L	

External bus	31-24	23-16	15-8	7-0
Address on memory side	+0		+2	
TAM[1:0]/RAM[1:0] = 01 TAM[1:0]/RAM[1:0] = 00	L		R	

Big endian (conversion needed)

Slot data	31-24	23-16	15-8	7-0
Transfer data	R		L	

External bus	31-24	23-16	15-8	7-0
Address on memory side	+2		+0	
TTAM[1:0]/RAM[1:0] = 01 TAM[1:0]/RAM[1:0] = 00	R		L	

Little endian (conversion not needed)

Figure 11.37 16-Bit Data Transfer for HAC/SSI

When HAC handles the transfer of 16-bit (word) audio data, it performs the alignment such that the left channel data is the most-significant word and the right channel data is the least-significant word.

When SSI handles the transfer of 16-bit audio data, however, it performs the alignment such that the left channel data is the least-significant word and the right channel data is the most-significant word. Therefore, the difference must be taken into consideration when performing transfers between HAC and SSI. For DMA transfer of audio data, specify DMAACR.TAM[1:0]/DMAACR.RAM[1:0] = 10 for one channel to adjust the alignment.

11.6.11 LCDC DMA Transfer

Figure 11.38 shows a DMA transfer flow for the LCDC.

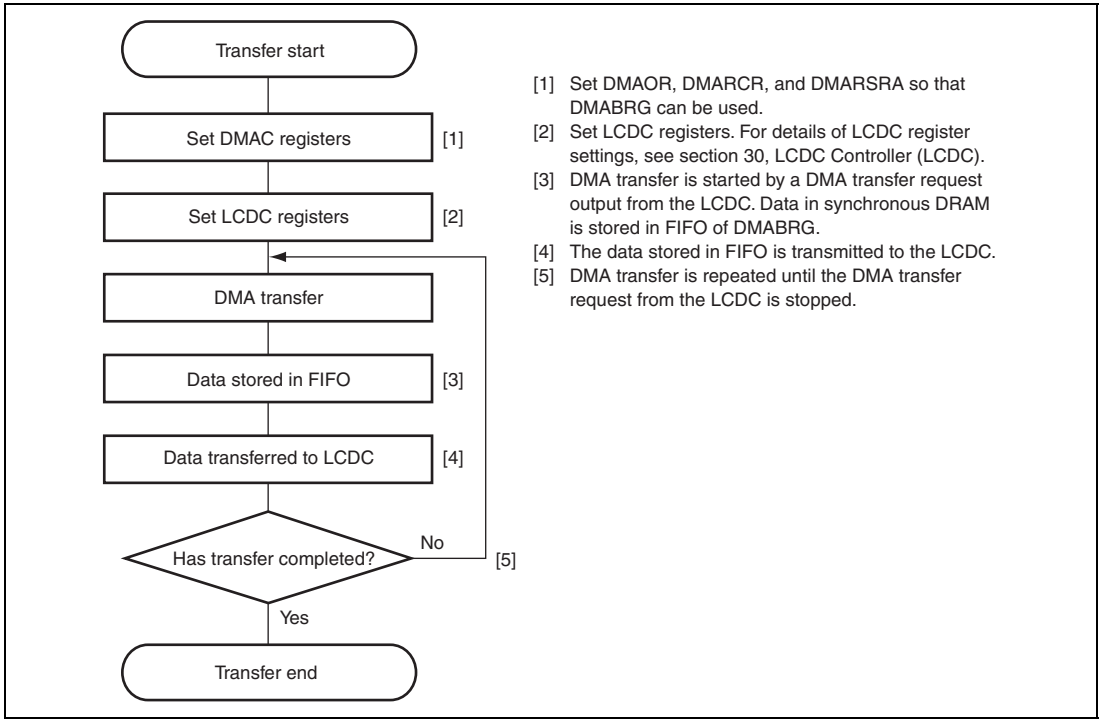


Figure 11.38 Example of LCDC Data Transfer Flow

The USB contains an 8-Kbyte shared memory. It is possible to perform a DMA transfer between the USB internal shared memory and synchronous DRAM by using the DMABRG.

Figure 11.39 shows a DMA transfer flow between the shared memory and synchronous DRAM. On this transfer, specifying the transfer size and number of transfers is not needed. The DMABRG converts the number of transfer bytes specified by the SZ bits in DMAURWSZ into the appropriate transfer data size and the number of transfers to perform a DMA transfer. When the number of bytes actually transferred reaches the number of transfer bytes specified by the SZ bits in DMAURWSZ, the UTF bit in DMABRGCR is set to 1 and the DMA transfer is successfully completed.

When the transfer is continued beyond the shared memory area (H'FE34 1000 to H'FE34 2FFF), a USB address error is generated. When a USB address error is detected, the UAF bit in DMABRGCR is set to 1 and operation ends abnormally.

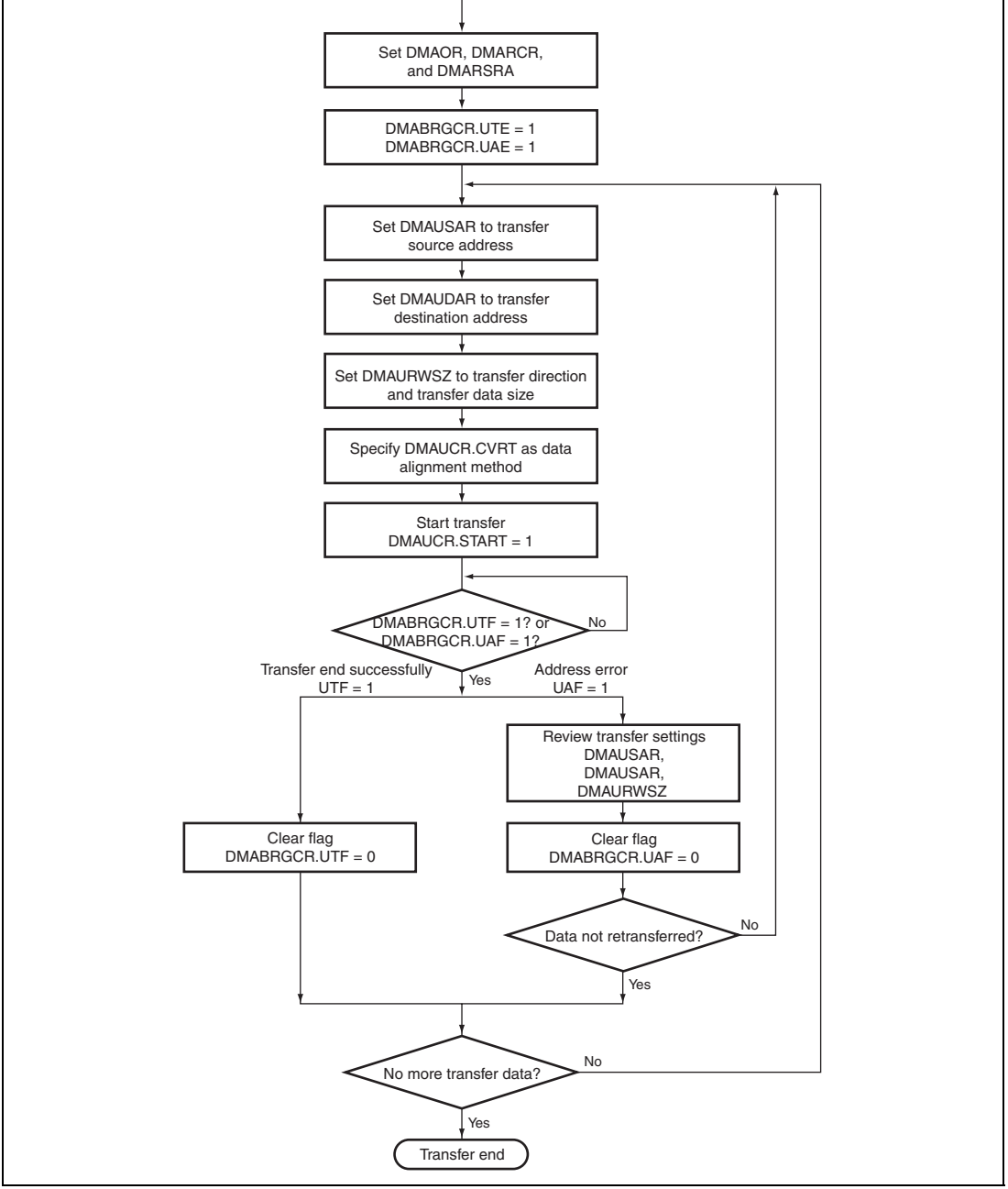


Figure 11.39 DMA Transfer Flow
Shared Memory ↔ Synchronous DRAM

The DMABRG supports the endian conversion function for data transfers between the peripheral bus and USB bridge bus, and the external bus and USB bridge bus shown in figure 11.40. The data alignment method of the USB is specified by the CVRT bits in DMAUCR. The data alignment method between the peripheral bus and USB bridge bus differs from that between the external bus and USB bridge bus. Table 11.12 shows the data alignment method between the peripheral bus and USB bridge bus and table 11.13 shows that between the external bus and USB bridge bus.

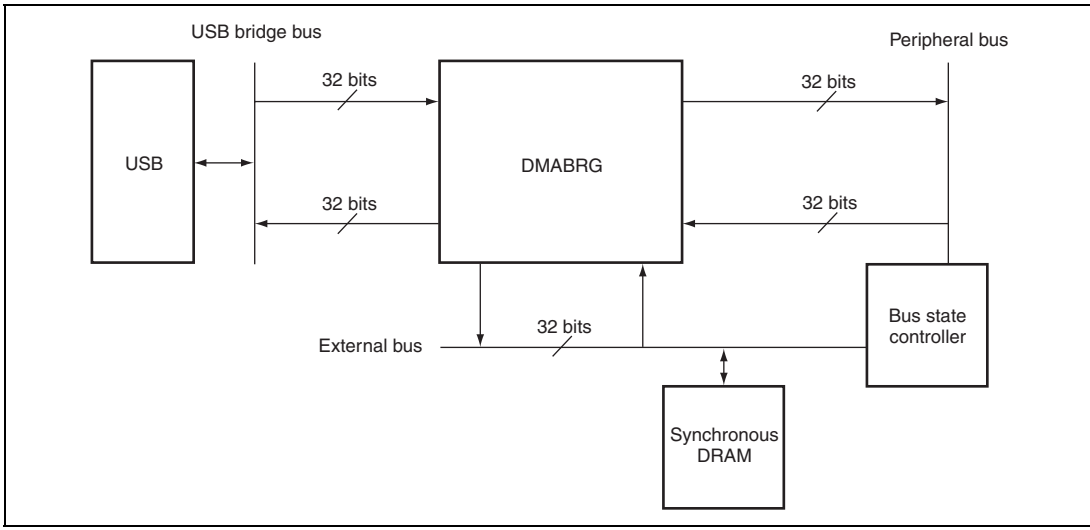
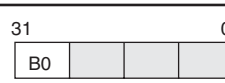
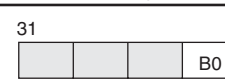
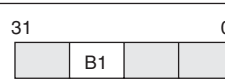
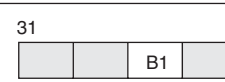
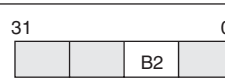
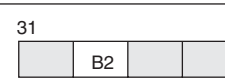
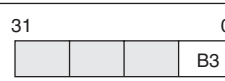
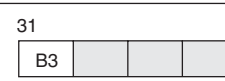
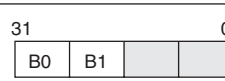
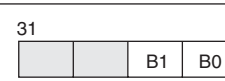
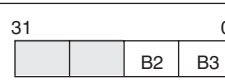
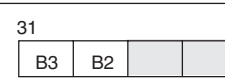
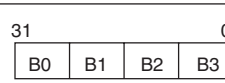
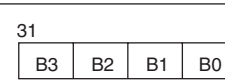
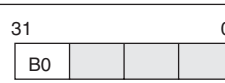
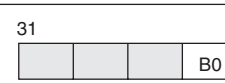
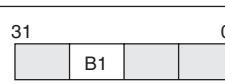
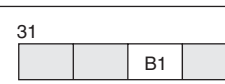
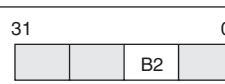
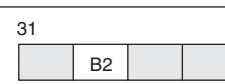
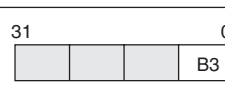
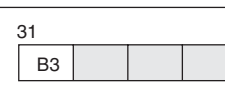
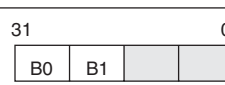
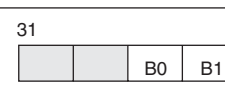
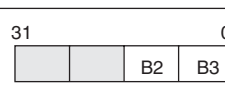
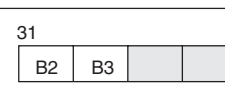
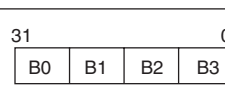
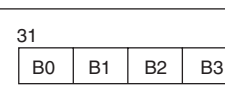
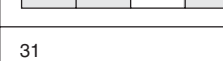
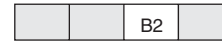
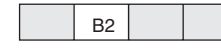
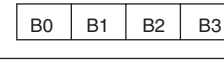
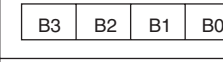
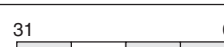
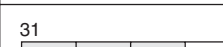
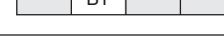
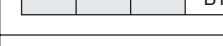


Figure 11.40 Bus Arrangement for Data Alignment

Transfer Mode	Access Size	Address	Peripheral Bus	USB Bridge Bus
Byte boundary mode	Byte	4n + 0		
		4n + 1		
		4n + 2		
		4n + 3		
	Word	4n + 0		
		4n + 2		
	Longword	4n + 0		
Word/longword boundary mode	Byte	4n + 0		
		4n + 1		
		4n + 2		
		4n + 3		
	Word	4n + 0		
		4n + 2		
	Longword	4n + 0		

Transfer Mode	Access Size	Address	External Bus	USB Bridge Bus
Byte boundary mode	Byte	4n + 0		
		4n + 1		
		4n + 2		
		4n + 3		
	Longword	4n + 0		
Word/longword boundary mode	Byte	4n + 0		
		4n + 1		
		4n + 2		
		4n + 3		
	Longword	4n + 0		

The DMABRG issues three interrupts: a USB address error interrupt, an all data transfer end interrupt, and a half data transfer end interrupt. The DMABRG generates a USB address error interrupt request for a DMA transfer request from the USB, an all data transfer end interrupt request for a DMA transfer request from the HAC, SSI, or USB, and a half data transfer end interrupt request for a transfer request from the HAC or SSI. A DMABRG interrupt request is not generated for a DMA transfer request from the LCDC. When a reset is cancelled, the interrupt priority is in the following order: a USB address error interrupt, an all data transfer end interrupt, and a half data transfer end interrupt.

(1) USB Address Error Interrupt Request (DMABRGI0)

When a USB address error occurs with the UAE bit in DMABRGCR set to 1, the DMABRG sets the UAF bit in DMABRGCR to 1 and outputs an interrupt request to the INTC.

(2) All Data Transfer End Interrupt Request (DMABRGI1)

- When all data transfer is completed on the receive side for channel 1 of the HAC or SSI with the A1RXEE bit in DMABRGCR set to 1, the DMABRG sets the A1RXEF bit in DMABRGCR to 1 and outputs an interrupt request to the INTC.
- When all data transfer is completed on the transmit side for channel 1 of the HAC or SSI with the A1TXEE bit in DMABRGCR set to 1, the DMABRG sets the A1TXEF bit in DMABRGCR to 1 and outputs an interrupt request to the INTC.
- When all data transfer is completed on the receive side for channel 0 of the HAC or SSI with the A0RXEE bit in DMABRGCR set to 1, the A0RXEF bit in DMABRGCR is set to 1 and an interrupt request is output to the INTC.
- When all data transfer is completed on the transmit side for channel 0 of the HAC or SSI with the A0TXEE bit in DMABRGCR set to 1, the DMABRG sets the A0TXEF bit in DMABRGCR to 1 and outputs an interrupt request to the INTC.
- When USB data transfer is completed with the UTE bit in DMABRGCR set to 1, the DMABRG sets the UTF bit in DMABRGCR to 1 and outputs an interrupt request to the INTC.

(3) Half Data Transfer End Interrupt Request (DMABRGI2)

- When data transfer of half of the bytes specified in DMAARXTCR is completed on the receive side for channel 1 of the HAC or SSI with the A1RXHE bit in DMABRGCR set to 1, the DMABRG sets the A1RXHF bit in DMABRGCR to 1 and outputs an interrupt request to the INTC.

the A1TXHF bit in DMABRGCR is set to 1 and an interrupt request is output to the INTC.

- When data transfer of half of the bytes specified in DMAARXTCR is completed on the receive side for channel 0 of the HAC or SSI with the A0RXHE bit in DMABRGCR set to 1, the DMABRG sets the A0RXHF bit in DMABRGCR to 1 and outputs an interrupt request to the INTC.
- When data transfer of half of the bytes specified in DMAARXTCR is completed on the transmit side for channel 0 of the HAC or SSI with the A0TXHE bit in DMABRGCR set to 1, the DMABRG sets the A0TXHF bit in DMABRGCR to 1 and outputs an interrupt request to the INTC.

The DMABRG outputs three types of interrupt requests to the INTC: an all data transfer end interrupt, a half data transfer end interrupt and an address error interrupt. To know which interrupt the DMABRG has issued, read interrupt flag bits in DMABRGCR. An interrupt flag bit that is set to 1 indicates the corresponding interrupt has been output.

11.7 Usage Notes

1. When modifying SAR, DAR, DMATCR, and CHCR, first clear the DE bit for the relevant channel.
2. Inputting an NMI interrupt with the DMAC not operating sets the NMIF bit in DMAOR.
   — When DMA transfer is not correctly performed, take the following actions:

   Read the NMIF, AE, and DME bits in DMAOR, the DE and TE bits in CHCR, and DMATCR on this LSI. If the NMIF bit was set before the transfer, the DMATCR indicates the transfer count that has been specified. If the NMIF bit was set during the transfer, when the DE bit is 1 and the TE bit is 0 in CHCR, the DMATCR indicates the number of transfers remaining.

   Also, the next addresses to be accessed can be found by reading SAR and DAR.

   If the AE bit has been set, an address error has occurred. Check the settings in CHCR, SAR, and DAR.
3. Check that DMA transfer is not in progress before making a transition to module standby state, standby mode, or deep sleep mode.

   Either check CHCR.TE = 1, or set DMAOR.DME = 0 to terminate DMA transfer. Setting DMAOR.DME = 0 stops the transfer on the completion of the DMA bus cycle currently being performed. Note, therefore, that transfer may not end immediately, depending on the transfer data size. DMA operation is not guaranteed if module standby state, standby mode, or deep sleep mode is entered without confirming that DMA transfer has ended.
4. Do not specify a DMAC, cache, BSC, or UBC control register as the DMAC transfer source or destination.

channel before setting the DE bit to 1 in CHCR, or make the register settings with the DE bit in CHCR cleared to 0, then set the DE bit to 1. It does not matter whether setting of the DME bit in DMAOR to 1 is carried out first or last. To operate the relevant channel, the DME and DE bits must both be set to 1. The DMAC may not operate normally if the SAR, DAR, and DMATCR settings are not made (with the exception of the unused register in single address mode).

6. After the DMATCR count reaches 0 and DMA transfer ends successfully, always write 0 to DMATCR even when executing the maximum number of transfers on the same channel.
7. When using falling edge detection for external requests, hold the external request pin high to make DMAC settings.
8. When using the DMAC in single address mode, specify an external address as the address. Specifying an on-chip peripheral module address causes an address error and stops transfers on all channels.

This LSI incorporates a clock pulse generator (CPG) that generates a CPU clock (Ick), peripheral clock (Pck), bus clock (Bck), and module clock (Fck).

The CPG generates the clocks supplied inside the processor and performs power-down mode control.

12.1 Features

The CPG has the following features.

- Six clock modes

  Any of six clock operating modes can be selected, with different division ratio combinations of the CPU clock, bus clock, and peripheral clock after a power-on reset.
- Five clocks

  The CPG can generate individually the CPU clock (Ick) used by the CPU, FPU, caches, and TLB, the peripheral clock (Pck) used by the peripheral modules, the bus clock (Bck) used by the external bus interface, the module clock (Fck), and the DCK clock (DCK).
- Frequency change function

  The PLL circuits and a frequency divider in the CPG enable the CPU clock, bus clock, peripheral clock, module clock, and DCK clock frequencies to be changed independently. Frequency changes are performed by software in accordance with the settings in FRQCR, MCKCR, and DCKDR.
- PLL on/off control

  Power consumption can be reduced by stopping the PLL circuits during low-frequency operation.
- Power-down mode control

  It is possible to stop the clock in sleep mode, deep sleep mode, hardware standby mode, and software standby mode, and to stop specific modules with the module standby function.

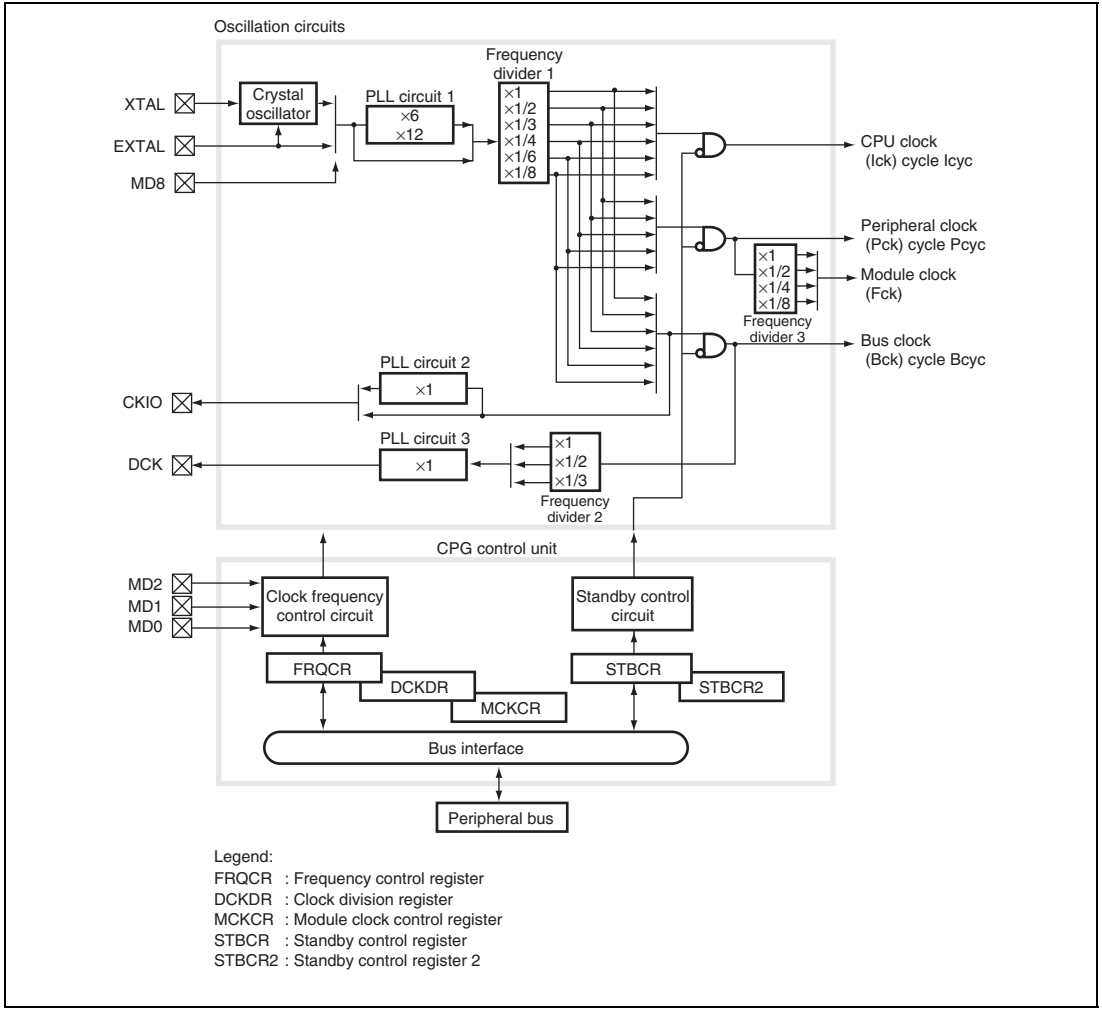


Figure 12.1 Block Diagram of CPG

Each of the CPG blocks functions as described below.

(1) PLL Circuit 1

PLL circuit 1 has a function for multiplying the clock frequency from the EXTAL pin or crystal oscillator by 6 or 12. Starting and stopping of this circuit is controlled by the setting of the frequency control register.

PLL circuit 2 coordinates the phases of the bus clock and the clock signal output from the CKIO pin. Starting and stopping of this circuit is controlled by the setting of the frequency control register.

(3) PLL Circuit 3

PLL circuit 3 coordinates the phases of the bus clock and the clock signal output from the DCK pin. Starting and stopping of this circuit is controlled by the setting of the clock division register.

(4) Crystal Oscillator

Oscillation circuits for when a crystal resonator is connected to the XTAL and EXTAL pins. Usage of the crystal oscillator is enabled by the MD8 pin setting.

(5) Frequency Divider 1

Frequency divider 1 generates the CPU clock (Ick), bus clock (Bck), and peripheral clock (Pck). The division ratio is set in the frequency control register.

(6) Frequency Divider 2

Frequency divider 2 generates the clock output from the DCK pin. The division ratio is set in the clock division register.

(7) Frequency Divider 3

(Frequency divider 3 generates the module clock (Fck). The division ratio is set in the module clock control register.

(8) Clock Frequency Control Circuit

The clock frequency control circuit controls the clock frequency by means of the MD pins, frequency control register, clock division register, and module clock control register.

(9) Standby Control Circuit

The standby control circuit controls the state of the on-chip oscillation circuits and other modules when the clock is switched or in sleep and standby modes.

(10) Frequency Control Register (FRQCR)

FRQCR contains control bits for the clock output from the CKIO pin, on/off of PLL circuits 1 and 2, and frequency division ratios of the CPU clock, bus clock, and peripheral clock.

DCKDR contains control bits for the clock output from the DCK pin, on/off of PLL circuit 3, division ratio of the DCK output clock, and enable/disable of the DCK output clock.

(12) Module Clock Control Register (MCKCR)

MCKCR contains control bits for the division ratio of the module clock.

(13) Standby Control Register (STBCR)

STBCR contains power-saving mode control bits. For further information on STBCR, see section 14, Power-Down Modes.

(14) Standby Control Register 2 (STBCR2)

STBCR2 contains power-saving mode control bits. For further information on STBCR2, see section 14, Power-Down Modes.

12.2 Input/Output Pins

Table 12.1 shows the CPG pin configuration and function.

Table 12.1 Pin Configuration and Function of an Oscillation Circuit

Pin Name	Abbreviation	I/O	Function
Mode control pins	MD0	Input	These bits set clock operating mode.
	MD1		
	MD2		
Crystal I/O pins (clock input pins)	XTAL	Output	Connects crystal resonator.
	EXTAL	Input	Connects crystal resonator, or used as external clock input pin.
	MD8	Input	Selects use/non-use of crystal resonator. When MD8 = 0, external clock is input from the EXTAL pin. When MD8 = 1, crystal resonator is connected directly to the EXTAL and XTAL pins.
Clock output pins	CKIO	Output	Used as external clock output pins. Level can also be fixed.
	DCK	Output	
CKIO enable pin	CKE	Output	0 when CKIO output clock is unstable*.

Note: * Set to 1 by a power-on reset.

Table 12.2 shows the relationship between the combinations of mode control pin (MD2 to MD0) settings and clock operating modes.

Table 12.3 shows the FRQCR settings and internal clock frequencies.

Table 12.2 Clock Operating Modes

Clock Operating Mode	Pin Combination					Frequency (vs. Input Clock)			
	MD2	MD1	MD0	PLL1	PLL2	CPU Clock	Bus Clock	Peripheral Clock	FRQCR Initial Value
0	0	0	0	On (×12)	On	12	3	3	H'0E1A
1	0	0	1	On (×12)	On	12	3/2	3/2	H'0E2C
2	0	1	0	On (×6)	On	6	2	1	H'0E13
3	0	1	1	On (×12)	On	12	4	2	H'0E13
4	1	0	0	On (×6)	On	6	3	3/2	H'0E0A
6	1	1	0	Off (×6)	Off	1	1/2	1/2	H'0808

Notes: 1. The multiplication factor of PLL 1 is solely determined by the clock operating mode.

2. For the ranges of input clock frequency, see the descriptions of the EXTAL clock input frequency (f_{EX}) and CKIO clock output (f_{OP}) in section 33.3.1, Clock and Control Signal Timing.

FRQCR (Lower 9 Bits)	Frequency Division Ratio		
	CPU Clock	Bus Clock	Peripheral Clock
H'000	1	1	1/2
H'002			1/4
H'004			1/8
H'008		1/2	1/2
H'00A			1/4
H'00C			1/8
H'011		1/3	1/3
H'013			1/6
H'01A		1/4	1/4
H'01C			1/8
H'023		1/6	1/6
H'02C		1/8	1/8
H'048	1/2	1/2	1/2
H'04A			1/4
H'04C			1/8
H'05A		1/4	1/4
H'05C			1/8
H'063		1/6	1/6
H'06C		1/8	1/8
H'091	1/3	1/3	1/3
H'093			1/6
H'0A3		1/6	1/6
H'0DA	1/4	1/4	1/4
H'0DC			1/8
H'0EC		1/8	
H'123	1/6	1/6	1/6
H'16C	1/8	1/8	1/8

Note: Do not set the lower nine bits of FRQCR to values other than those shown in this table.

The CPG has the following registers. For details on the addresses of these registers and the state of registers in each operating mode, see section 32, List of Registers.

Table 12.4 Register Configuration (1)

Register Name	Abbrev.	R/W	P4 Address	Area 7 Address	Size	Sync Clock
Frequency control register	FRQCR	R/W	H'FFC0 0000	H'1FC0 0000	16	Pck
Clock division register	DCKDR	R/W	H'FE0A 0020	H'1E0A 0020	32	Pck
Module clock control register	MCKCR	R/W	H'FE0A 0024	H'1E0A 0024	32	Pck

Table 12.4 Register Configuration (2)

		Power-on Reset		Manual Reset by		Standby	
Register Name	Abbrev.	by $\overline{\text{RESET}}$ Pin	by WDT/ H-UDI	$\overline{\text{RESET}}$ Pin/WDT/ Multiple Exception	Sleep by Sleep Instruction/ Deep Sleep	by Hardware	by Software/ Each Module
Frequency control register	FRQCR	*1	Retained	Retained	Retained	*2	Retained
Clock division register	DCKDR	H'0000 0001		Retained	Retained		Retained
Module clock control register	MCKCR	H'0000 0000		Retained	Retained		Retained

Notes: 1. The initial values of bits 11 to 9 are 1, and those of bits 8 to 0 are undefined.
2. After exiting hardware standby mode, this LSI enters the power-on reset state caused by the $\overline{\text{RESET}}$ pin.

FRQCR is a 16-bit readable/writable register that specifies use/non-use of clock output from the CKIO pin, on/off control of PLL circuits 1 and 2, and the frequency division ratios of the CPU clock, bus clock, and peripheral clock. FRQCR can only be accessed in words.

FRQCR is initialized only by a power-on reset via the $\overline{\text{RESET}}$ pin. The initial value of each bit is determined by the clock operating mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	CKO EN	PLL1 EN	PLL2 EN	IFC2	IFC1	IFC0	BFC2	BFC1	BFC0	PFC2	PFC1	PFC0
Initial value:	0	0	0	0	1	1	1	-	-	-	-	-	-	-	-	-
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

Bit	Bit Name	Initial Value	R/W	Description
15 to 12	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
11	CKOEN	1	R/W	Clock Output Enable Specifies whether a clock is output from the CKIO pin or the CKIO pin is placed in the high-impedance state. When the CKIO pin goes to the high-impedance state, operation continues at the operating frequency before this state was entered. When the CKIO pin becomes high-impedance, it is pulled up. Note that the CKIO pin is not pulled up in hardware standby mode. 0: CKIO pin goes to high-impedance state 1: Clock is output from CKIO pin
10	PLL1EN	1	R/W	PLL Circuit 1 Enable Specifies whether PLL circuit 1 is on or off. 0: PLL circuit 1 is not used 1: PLL circuit 1 is used
9	PLL2EN	1	R/W	PLL Circuit 2 Enable Specifies whether PLL circuit 2 is on or off. 0: PLL circuit 2 is not used 1: PLL circuit 2 is used

8	IFC2	—	R/W	CPU Clock Frequency Division Ratio Setting
7	IFC1	—	R/W	These bits specify the CPU clock frequency
6	IFC0	—	R/W	division ratio with respect to the input clock or PLL circuit 1 output frequency. 000: ×1 001: ×1/2 010: ×1/3 011: ×1/4 100: ×1/6 101: ×1/8 Other than above: Setting prohibited
5	BFC2	—	R/W	Bus Clock Frequency Division Ratio Setting
4	BFC1	—	R/W	These bits specify the bus clock frequency
3	BFC0	—	R/W	division ratio with respect to the input clock or PLL circuit 1 output frequency. 000: ×1 001: ×1/2 010: ×1/3 011: ×1/4 100: ×1/6 101: ×1/8 Other than above: Setting prohibited
2	PFC2	—	R/W	Peripheral Clock Frequency Division Ratio
1	PFC1	—	R/W	Setting
0	PFC0	—	R/W	These bits specify the peripheral clock frequency division ratio with respect to the input clock or PLL circuit 1 output frequency. 000: ×1/2 001: ×1/3 010: ×1/4 011: ×1/6 100: ×1/8 Other than above: Setting prohibited

DCKDR is a 32-bit readable/writable register that specifies use/non-use of clock output from the DCK pin and the DCK clock frequency division ratio.

By setting the DIV1 and DIV0 bits, the CKIO clock is divided by 1, 2, or 3 and supplied to the DCK pin. This division ratio setting also allows the DCK clock to be extended to become one to three CKIO cycles even while $\overline{BS2}$ is being asserted. For details on adjustment of the $\overline{CS}$ negate time, see the description of WCR4.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
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
	-	-	-	-	-	-	-	-	DCK EN	-	-	-	PLL3 EN	DCK OUT	DIV1	DIV0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7	DCKEN	0	R	Indicates whether the DCK output clock is stable (usable). 0: Unstable (unusable) 1: Stable (usable)
6 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
3	PLL3EN	0	R/W	PLL Circuit 3 Enable Specifies whether PLL circuit 3 is on or off. Writing 1 to this bit turns the on-chip PLL circuit 3 on. 0: PLL circuit 3 is not used. In this case, the DCK output is fixed at 1. 1: PLL circuit 3 is used.
2	DCKOUT	0	R/W	DCK Output Control Controls the DCK pin state. Writing 1 to this bit sets the DCK pin at the output state. 0: DCK pin goes to high-impedance state 1: Clock is output from DCK pin

1	DIV1	0	R/W	Frequency Division Ratio Setting
0	DIV0	1	R/W	These bits specify the DCK clock frequency division ratio with respect to the CKIO clock. 00: Setting prohibited 01: CKIO × 1/1 10: CKIO × 1/2 11: CKIO × 1/3

12.4.3 Module Clock Control Register (MCKCR)

MCKCR is a 32-bit readable/writable register that specifies the frequency division ratio of the module clock (Fck).

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
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
	-	-	-	-	-	-	-	-	-	-	-	-	FLM CK3	FLM CK2	FLM CK1	FLM CK0
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/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
3	FLMCK3	0	R/W	Module Clock Frequency Division Ratio Setting
2	FLMCK2	0	R/W	These bits specify the Fck clock frequency
1	FLMCK1	0	R/W	division ratio with respect to the peripheral clock.
0	FLMCK0	0	R/W	0000: Peripheral clock × 1/1 0001: Setting prohibited 0010: Setting prohibited 0011: Setting prohibited Other than above: Setting prohibited

There are two methods of changing the clock frequency: by switching off and on PLL circuit 1, and by changing the frequency division ratio of each clock. In both cases, control is performed by software by means of FRQCR, MCKCR, and DCKDR. These methods are described below.

12.5.1 Switching between PLL Circuit 1 On/Off (When PLL Circuit 2 Is Off)

When PLL circuit 1 is turned on, the oscillation stabilization time for PLL circuit 1 is required. The oscillation stabilization time is counted by the on-chip WDT.

1. Set a value in WDT to provide the specified oscillation stabilization time, and stop the WDT. The following settings are necessary:
   TME bit in WTCSR = 0: WDT stopped
   CKS2 to CKS0 bits in WTCSR: WDT count clock frequency division ratio
   WTCNT: Initial counter value
2. Set the PLL1EN bit to 1.
3. This LSI operation stops temporarily, and the WDT starts counting up. The internal clock stops and an unstable clock is output to the CKIO pin.
4. After the WDT count overflows, a clock begins to be supplied within the chip, and this LSI resumes operation. The WDT stops after overflowing.

12.5.2 Switching between PLL Circuit 1 On/Off (When PLL Circuit 2 Is On)

When PLL circuit 2 is on, the oscillation stabilization time for PLL circuit 1 and PLL circuit 2 is required.

1. Make WDT settings as in step 1 in section 12.5.1.
2. Set the PLL1EN bit to 1.
3. This LSI operation stops temporarily, PLL circuit 1 starts oscillation, and the WDT starts counting up. The internal clock stops and an unstable clock is output to the CKIO pin.
4. After the WDT count overflows, PLL circuit 2 starts oscillation. The WDT resumes its up-count from the value set in step 1 above. Even during this time, the internal clock is stopped and an unstable clock is output to the CKIO pin.
5. After the WDT count overflows, a clock begins to be supplied within the chip, and this LSI resumes operation. The WDT stops after overflowing.

If PLL circuit 2 is on when the bus clock frequency division ratio is changed, the oscillation stabilization time for PLL circuit 2 is required.

1. Make WDT settings as in step 1 in section 12.5.1, Switching between PLL Circuit 1 On/Off (When PLL Circuit 2 is Off).
2. Set the BFC2 to BFC0 bits to the desired value.
3. This LSI stops temporarily, and the WDT starts counting up. The internal clock stops and an unstable clock is output to the CKIO pin.
4. After the WDT count overflows, a clock begins to be supplied within the chip, and this LSI resumes operation. The WDT stops after overflowing.

12.5.4 Changing Bus Clock Frequency Division Ratio (When PLL Circuit 2 Is Off)

If PLL circuit 2 is off when the bus clock frequency division ratio is changed, WDT counting is not performed.

1. Set the BFC2 to BFC0 bits to the desired value.
2. The specified clock is switched to immediately.

12.5.5 Changing Frequency Division Ratio of CPU Clock or Peripheral Clock

When the frequency division ratio of the CPU clock or peripheral clock is changed, WDT counting is not performed.

1. Set the IFC2 to IFC0 or PFC2 to PFC0 bits to the desired value.
2. The specified clock is switched to immediately.

12.5.6 Switching between PLL Circuit 3 On/Off

When PLL circuit 3 is turned on, the oscillation stabilization time for PLL circuit 3 is required. The oscillation stabilization time is counted by an on-chip fixed timer. After counting has finished (oscillation stabilized), the DCKEN bit in DCKDR is set to 1. At this timing the DCK oscillation stabilization end can be notified to external devices by using the GPIO to output the value in the DCKEN bit.

Before turning on/off PLL circuit 1 or 2, changing the bus clock frequency division ratio, or entering standby mode, make sure to stop PLL circuit 3 (clear the DCKEN bit to 0). After changing these settings, start PLL circuit 3.

1. Set the PLL3EN bit in DCKDR to 1.
2. The on-chip fixed timer starts counting up. At this time, an unstable clock is output to the DCK pin. The DCKEN bit is cleared to 0 to indicate that the DCK cannot be used.
3. After the on-chip fixed timer finishes counting, the DCKEN bit is set to 1 to indicate that DCK oscillation has become stable. For the time until the on-chip fixed timer finishes counting (oscillation stabilization time), refer to section 33, Electrical Characteristics.

- Turning Off PLL Circuit 3
  1. Clear the PLL3EN bit in DCKDR to 0.

12.5.7 Changing DCK Output Clock Division Ratio

Before changing the DCK clock frequency division ratio, be sure to stop oscillation of PLL circuit 3.

1. Clear the PLL3EN bit in DCKDR to 0. Then set the clock frequency division ratio with the DIV0 and DIV1 bits in DCKDR.
2. Set the PLL3EN bit to 1.
3. The on-chip fixed timer starts counting up. At this time, an unstable clock is output to the DCK pin. The DCKEN bit is cleared to 0 to indicate that the DCK cannot be used.
4. After the on-chip fixed timer finishes counting, the DCKEN bit is set to 1 to indicate that DCK oscillation has become stable.

12.5.8 Controlling DCK Output Clock

The DCK pin can be switched between clock output and the high-impedance state by means of the DCKOUT bit in DCKDR. When the DCK pin goes to the high-impedance state, it is pulled up. Make the DCKOUT bit setting after stopping PLL circuit 3.

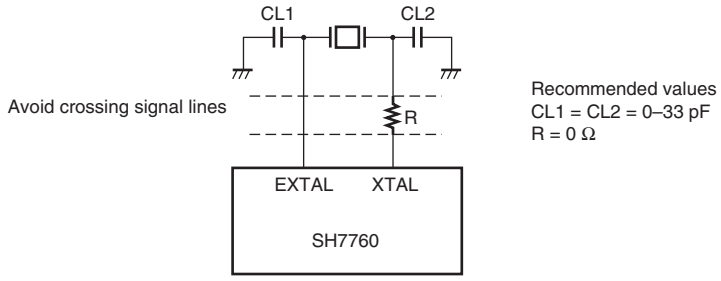
1. Clear the PLL3EN bit in DCKDR to 0. Then set the DCK pin state with the DCKOUT bit.

The CKIO pin can be switched between clock output and the high-impedance state by means of the CKOEN bit in FRQCR. When the CKIO pin goes to the high-impedance state, it is pulled up.

12.6 Usage Notes

(1) Using a crystal resonator

Place the crystal resonator and capacitors CL1 and CL2 close to the EXTAL and XTAL pins. To prevent induction from interfering with correct oscillation, ensure that no other signal lines cross the signal lines for these pins.



Note: The values for CL1, CL2, and the damping resistance should be determined after consultation with the crystal resonator manufacturer.

Figure 12.2 Points for Attention when Using Crystal Resonator

(2) Inputting external clock from EXTAL pin

Make no connection to the XTAL pin.

(3) Using a PLL oscillation circuit

Separate VDD-CPG and VSS-CPG from the other VDD and VSS lines at the board power supply source, and insert resistors RCB and RB and bypass capacitors CPB and CB, close to the pins.

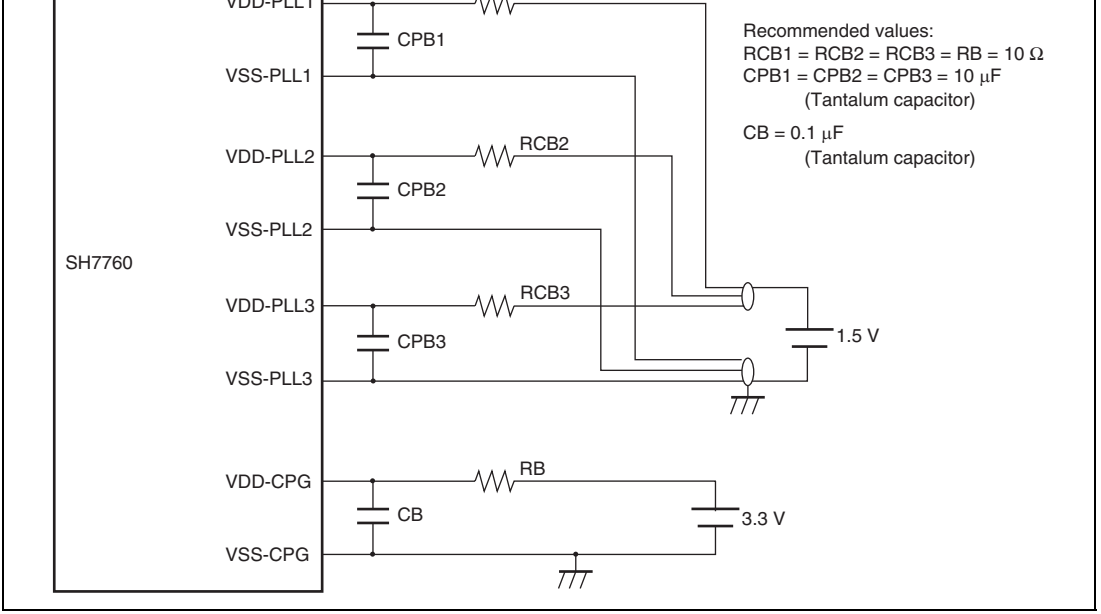


Figure 12.3 Points for Attention when Using PLL Oscillation Circuit

The WDT is a single-channel timer used to count the clock stabilization time when exiting standby mode or a temporary standby state in which the clock frequency is changed. It can be used as a normal watchdog timer or an interval timer.

13.1 Features

The WDT has the following features.

- Can be used to secure clock stabilization time

  Used when exiting standby mode or a temporary standby state in which the clock frequency is changed.
- Can be switched between watchdog timer mode and interval timer mode
- Internal reset generation in watchdog timer mode

  An internal reset is executed on counter overflow.
- Interrupt generation in interval timer mode

  An interval timer interrupt is generated on counter overflow.
- Selection of eight counter input clocks

  Any of eight clocks can be selected, scaled from the ×1 clock of frequency divider 1 shown in figure 12.1 in Section 12, Clock Pulse Generator (CPG).
- Power-on reset or manual reset can be selected.

Figure 13.1 shows a block diagram of the WDT.

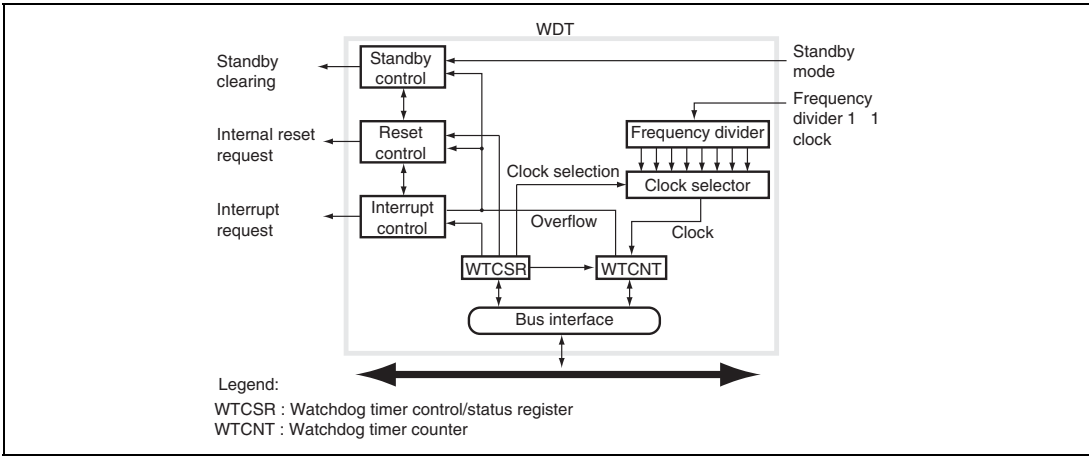


Figure 13.1 Block Diagram of WDT

The WDT has the following registers. These registers control clock selection and timer mode switching. For details on the addresses of these registers and the state of registers in each operating mode, see section 32, List of Registers.

Table 13.1 Register Configuration (1)

Register Name	Abbrev.	R/W	P4 Address	Area 7 Address	Size	Sync Clock
Watchdog timer counter	WTCNT	R/W	H'FFC0 0008	H'1FC0 0008	8/16*[1]	Pck
Watchdog timer control/status register	WTCSR	R/W	H'FFC0 000C	H'1FC0 000C	8/16*[1]	Pck

Table 13.1 Register Configuration (2)

		Power-on Reset		Manual Reset by $\overline{\text{RESET}}$ Pin/WDT/ Multiple Exception	Sleep by Sleep Instruction/ Deep Sleep	Standby	
Register Name	Abbrev.	by $\overline{\text{RESET}}$ Pin	by WDT/ H-UDI			by Hardware	by Software/ Each Module
Watchdog timer counter	WTCNT	H'00	Retained	Retained	Retained	*[2]	Retained
Watchdog timer control/status register	WTCSR	H'00	Retained	Retained	Retained		Retained

Notes: 1. Read: Byte access
Write: Word access
2. After exiting hardware standby mode, this LSI enters the power-on reset state caused by the $\overline{\text{RESET}}$ pin.

WTCNT is an 8-bit readable/writable counter that counts up on the selected clock. When WTCNT overflows, a reset is generated in watchdog timer mode, or an interrupt is generated in interval timer mode. WTCNT is initialized to H'00 only by a power-on reset via the $\overline{\text{RESET}}$ pin.

To write to WTCNT, use a word-size access with the upper byte set to H'5A. To read from WTCNT, use a byte-size access.

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

13.2.2 Watchdog Timer Control/Status Register (WTCSR)

WTCSR is an 8-bit readable/writable register containing bits for selecting the clock used for counting and timer mode, and overflow flags.

To write to WTCSR, use a word-size access with the upper byte set to H'A5. To read from WTCSR, use a byte-size access.

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	Bit Name	Initial Value	R/W	Description
7	TME	0	R/W	Timer Enable Specifies starting and stopping of timer operation. Clear this bit to 0 when using the WDT in standby mode or to change a clock frequency. 0: Up-count stopped, WTCNT value retained 1: Up-count started
6	WT/$\overline{\text{IT}}$	0	R/W	Timer Mode Select Specifies whether the WDT is used as a watchdog timer or interval timer. Up counting may not be performed correctly if this bit is modified while the WDT is running. 0: Interval timer mode 1: Watchdog timer mode

5	RSTS	0	R/W	Reset Select Specifies the kind of reset to be performed when WTCNT overflows in watchdog timer mode. This setting is ignored in interval timer mode. 0: Power-on reset 1: Manual reset
4	WOVF	0	R/W	Watchdog Timer Overflow Flag Indicates that WTCNT has overflowed in watchdog timer mode. This flag is not set in interval timer mode. 0: No overflow 1: WTCNT has overflowed in watchdog timer mode
3	IOVF	0	R/W	Interval Timer Overflow Flag Indicates that WTCNT has overflowed in interval timer mode. This flag is not set in watchdog timer mode. 0: No overflow 1: WTCNT has overflowed in interval timer mode
2 1 0	CKS2 CKS1 CKS0	0 0 0	R/W R/W R/W	Clock Select 2 to 0 These bits select the clock used for the WTCNT count from eight clocks obtained by dividing the input clock of Frequency divider 1×1 clock. When PLL1 is switched on or off, the clock after the switching is used. The overflow cycles shown below are for use of a 33-MHz input clock and PLL circuit 1 on (×6). Clock Division Ratio — Overflow Cycle 000: 1/32 — 41 µs 001: 1/64 — 82 µs 010: 1/128 — 164 µs 011: 1/256 — 328 µs 100: 1/512 — 656 µs 101: 1/1024 — 1.31 ms 110: 1/2048 — 2.62 ms 111: 1/4096 — 5.25 ms Up counting may not be performed correctly if bits CKS2 to CKS0 are modified while the WDT is running. Always stop the WDT before modifying these bits.

WTCNT and WTCSR differ from other registers in being more difficult to write to. The procedure for writing to these registers is given below.

(1) Writing to WTCNT and WTCSR

These registers must be written to with a word transfer instruction. They cannot be written to with a byte or longword transfer instruction.

As shown in figure 13.2, when writing to WTCNT, perform the transfer with the upper byte set to H'5A and the lower byte containing the write data. When writing to WTCSR, perform the transfer with the upper byte set to H'A5 and the lower byte containing the write data. This transfer procedure writes the lower byte data to WTCNT or WTCSR.

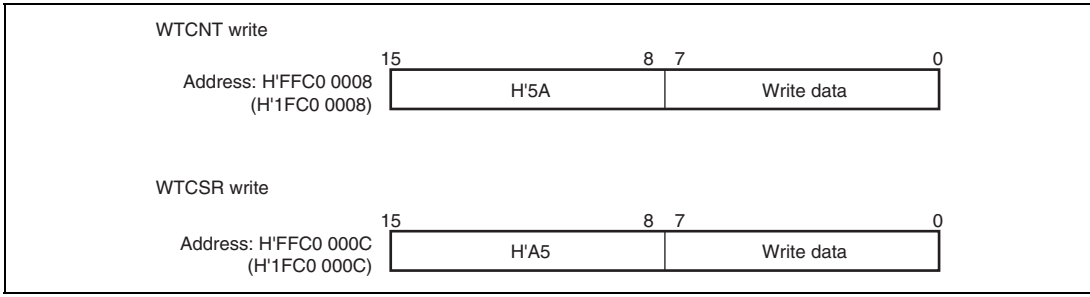


Figure 13.2 Writing to WTCNT and WTCSR

13.3 Operation

13.3.1 Standby Clearing Procedure

The WDT is used when clearing standby mode by means of an NMI or other interrupt. The procedure is shown below. (As the WDT does not operate when standby mode is cleared with a reset, the $\overline{\text{RESET}}$ pin should be held low until the clock stabilizes.)

1. Be sure to clear the TME bit in WTCSR to 0 before making a transition to software standby mode. If the TME bit is set to 1, an inadvertent reset or interval timer interrupt may be caused when the count overflows.
2. Select the count clock to be used with bits CKS2 to CKS0 in WTCSR, and set the initial value in WTCNT. Make these settings so that the time until the count overflows is at least as long as the clock oscillation stabilization time. Make a transition to software standby mode, and stop the clock, by executing a SLEEP instruction.
3. The WDT starts counting on detection of an NMI signal transition edge or an interrupt.

operation. The WOVF flag in WTCSR is not set at this time.

5. The counter stops at a value of H'00 to H'01. The value at which the counter stops depends on the clock ratio.

13.3.2 Frequency Changing Procedure

The WDT is used when changing the clock frequency by means of the PLL. It is not used when the frequency is changed simply by switching between frequency dividers.

1. Be sure to clear the TME bit in WTCSR to 0 before making a frequency change. If the TME bit is set to 1, an inadvertent reset or interval timer interrupt may be caused when the count overflows.
2. Select the count clock to be used with bits CKS2 to CKS0 in WTCSR, and set the initial value in WTCNT. Make these settings so that the time until the count overflows is at least as long as the clock oscillation stabilization time.
3. When FRQCR is modified, the clock stops. The WDT starts counting. For details of FRQCR, see section 12.4.1, Frequency Control Register (FRQCR).
4. When the WDT count overflows, the CPG resumes clock supply and the processor resumes operation. The WOVF flag in WTCSR is not set at this time.
5. The counter stops at a value of H'00 to H'01. The value at which the counter stops depends on the clock ratio.
6. When re-setting WTCNT immediately after modifying FRQCR, first read the counter and confirm that its value is as described in step 5 above.

13.3.3 Using Watchdog Timer Mode

1. Set the WT/$\overline{\text{IT}}$ bit in WTCSR to 1, select the type of reset with the RSTS bit, and the count clock with bits CKS2 to CKS0, and set the initial value in WTCNT.
2. When the TME bit in WTCSR is set to 1, the count starts in watchdog timer mode.
3. During operation in watchdog timer mode, write H'00 to the counter periodically so that it does not overflow.
4. When the counter overflows, the WDT sets the WOVF flag in WTCSR to 1, and generates a reset of the type specified by the RSTS bit. The counter then continues counting.

When the WDT is operating in interval timer mode, an interval timer interrupt is generated each time the counter overflows. This enables interrupts to be generated at fixed intervals.

1. Clear the WT/$\overline{IT}$ bit in WTCSR to 0, select the count clock with bits CKS2 to CKS0, and set the initial value in WTCNT.
2. When the TME bit in WTCSR is set to 1, the count starts in interval timer mode.
3. When the counter overflows, the WDT sets the IOVF flag in WTCSR to 1, and sends an interval timer interrupt request to INTC. The counter continues counting.

In power-down modes, some of the on-chip peripheral modules and the CPU functions are halted, enabling power consumption to be reduced.

The following power-down modes and functions are provided.

- Sleep mode
- Deep sleep mode
- Software standby mode
- Hardware standby mode
- Module standby function

Table 14.1 shows the conditions for entering these modes from the program execution state, the status of the CPU and peripheral modules in each mode, and the method of exiting each mode.

Power-Down Mode	Entering Conditions	Status						Exiting Method
		CPG	CPU	On-Chip Memory	Peripheral Modules	Pins	External Memory	
Sleep mode	SLEEP instruction executed while STBY bit is 0 in STBCR	Operating	Halted (registers retained)	Retained	Operating	Retained	Refreshing	• Interrupt • Reset
Deep sleep mode	SLEEP instruction executed while STBY bit is 0 in STBCR, and DSLP bit is 1 in STBCR2	Operating	Halted (registers retained)	Retained	Operating (DMA halted)	Retained	Self-refreshing	• Interrupt • Reset
Software standby mode	SLEEP instruction executed while STBY bit is 1 in STBCR	Halted	Halted (registers retained)	Retained	Halted	Retained	Self-refreshing	• Interrupt • Reset
Hardware standby mode	Setting CA pin to low level	Halted	Halted	Undefined	Halted	High-impedance state	Undefined	• Power-on reset
Module standby function	• Setting MSTP bit to 1 in STBCR • Setting CSTP bit to 1 in CLKSTP00	Operating	Operating	Retained	Specified modules halted	Retained	Refreshing	• Clearing MSTP and CSTP bits to 0 • Reset

Table 14.2 shows the pins used for power-down mode control.

Table 14.2 Pin Configuration

Pin Name	Abbreviation	I/O	Function
Processing status 1	STATUS1	Output	Indicate the processor's operating status.
Processing status 0	STATUS0		STATUS1 STATUS0 Operating Status High High Reset High Low Sleep mode Low High Standby mode Low Low Normal operation
Hardware standby request	CA	Input	A transition to hardware standby mode is made by inputting a low-level to the pin.

14.2 Register Descriptions

The following registers are used for power-down mode control. For details on the addresses of these registers and the state of registers in each operating mode, see section 32, List of Registers.

Table 14.3 Register Configuration (1)

Register Name	Abbrev.	R/W	P4 Address	Area 7 Address	Size	Sync Clock
Standby control register	STBCR	R/W	H'FFC0 0004	H'1FC0 0004	8	Pck
Standby control register 2	STBCR2	R/W	H'FFC0 0010	H'1FC0 0010	8	Pck
Clock stop register 00	CLKSTP00	R/W	H'FE0A 0000	H'1E0A 0000	32	Pck
Clock stop clear register 00	CLKSTPCLR00	W	H'FE0A 0010	H'1E0A 0010	32	Pck

Register Name	Abbrev.	Power-on Reset by RESET Pin/ WDT/H-UDI	Manual Reset by RESET Pin/WDT/ Multiple Exception	Sleep by Sleep Instruction/ Deep Sleep	Standby by Hardware	Standby by Software/ Each Module
Standby control register	STBCR	H'00	Retained	Retained	*	Retained
Standby control register 2	STBCR2	H'00	Retained	Retained		Retained
Clock stop register 00	CLKSTP00	H'0000 0000	Retained	Retained		Retained
Clock stop clear register 00	CLKSTPCLR00	—	—	Retained		Retained

Note: * After exiting hardware standby mode, this LSI enters the power-on reset state caused by the RESET pin.

14.2.1 Standby Control Register (STBCR)

STBCR is an 8-bit readable/writable register that specifies the power-down mode status.

Bit:	7	6	5	4	3	2	1	0
	STBY	-	-	MSTP4	-	MSTP2	-	-
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R	R	R/W	R	R/W	R	R

Bit	Bit Name	Initial Value	R/W	Description
7	STBY	0	R/W	Standby Specifies a transition to software standby mode. 0: Transition to sleep mode on execution of SLEEP instruction 1: Transition to software standby mode on execution of SLEEP instruction
6, 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

4	MSTP4	0	R/W	Module Stop 4 Specifies stopping of the clock supply to the DMAC among the peripheral modules. When DMA transfer is used, stop the transfer before setting this bit to 1. When DMA transfer is performed after clearing this bit to 0, DMAC settings must be made again. 0: DMAC operates 1: DMAC clock supply is stopped
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
2	MSTP2	0	R/W	Module Stop 2 Specifies stopping of the clock supply to the TMU among the peripheral modules. 0: TMU operates 1: TMU clock supply is stopped
1, 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

14.2.2 Standby Control Register 2 (STBCR2)

STBCR2 is an 8-bit readable/writable register that specifies the sleep mode and deep sleep mode transition conditions.

Bit:	7	6	5	4	3	2	1	0
	DSLP	STHZ	-	-	-	-	MSTP6	MSTP5
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	Bit Name	Initial value	R/W	Description
7	DSLP	0	R/W	Deep Sleep Specifies a transition to deep sleep mode 0: Transition to sleep mode or standby mode on execution of SLEEP instruction, according to setting of the STBY bit in STBCR 1: Transition to deep sleep mode on execution of SLEEP instruction when the STBY bit in STBCR is 0.

6	STHZ	0	R/W	STATUS Pin High-Impedance Control Selects whether the STATUS0 and STATUS1 pins are set to high-impedance in hardware standby mode. 0: Sets STATUS0 and STATUS1 pins to high-impedance in hardware standby mode 1: Drives STATUS0 pin low and STATUS1 pin high in hardware standby mode
5 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1	MSTP6	0	R/W	Module Stop 6 Specifies that the clock supply to the store queue (SQ) in the cache controller (CCN) is stopped. Setting the MSTP6 bit to 1 stops the clock supply to the SQ, and the SQ functions are therefore unavailable. 0: SQ operating 1: Specifies stopping of the clock supply to the SQ among the peripheral module.
0	MSTP5	0	R/W	Module Stop 5 Specifies stopping of the clock supply to the UBC among the peripheral module. 0: UBC operating 1: UBC clock supply is stopped

CLKSTP00 is a 32-bit readable/writable register that controls the operating clock for peripheral modules. The clock supply is stopped by writing 1 to the corresponding bit in CLKSTP00.

The clock supply is restarted by writing 1 to the corresponding bit in CLKSTPCLR00. Writing 0 to CLKSTP00 will not change the bit value.

Table 14.4 shows which module each bit is assigned to.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CSTP31	CSTP30	CSTP29	CSTP28	CSTP27	CSTP26	CSTP25	CSTP24	CSTP23	CSTP22	-	CSTP20	CSTP19	-	CSTP17	CSTP16
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
	CSTP15	CSTP14	CSTP13	CSTP12	CSTP11	CSTP10	CSTP9	CSTP8	-	-	-	-	-	-	-	CSTP0
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

Table 14.4 Bit Assignment of CLKSTP00 and CLKSTPCLR00

Bit No.	Bit Name	Module	Bit No.	Bit Name	Module	Bit No.	Bit Name	Module	Bit No.	Bit Name	Module
31	CSTP31	SSI(1)	23	CSTP23	MFI	15	CSTP15	HAC(0)	7	—	—
30	CSTP30	SSI(0)	22	CSTP22	HSPI	14	CSTP14	ADC	6	—	—
29	CSTP29	SCIF(2)	21	—	—	13	CSTP13	HCAN2(1)	5	—	—
28	CSTP28	SCIF(1)	20	CSTP20	LCDC	12	CSTP12	HCAN2(0)	4	—	—
27	CSTP27	SCIF(0)	19	CSTP19	USB	11	CSTP11	DMABRG	3	—	—
26	CSTP26	I^2C(1)	18	—	—	10	CSTP10	GPIO	2	—	—
25	CSTP25	I^2C(0)	17	CSTP17	CMT	9	CSTP9	MMCIF	1	—	—
24	CSTP24	SIM	16	CSTP16	HAC(1)	8	CSTP8	DMAC	0	CSTP0	INTC

CLKSTPCLR00 is a 32-bit write-only register that is used to clear corresponding bits in CLKSTP00. Write 1 to the corresponding bits in CLKSTPCLR00 to restart supplying the clock. Table 14.4 shows which module each bit is assigned to.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
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
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

14.3 Operation

14.3.1 Sleep Mode

(1) Transition to Sleep Mode

If a SLEEP instruction is executed when the STBY bit in STBCR is cleared to 0, the chip switches from the program execution state to sleep mode. After execution of the SLEEP instruction, the CPU halts but its register contents are retained. The peripheral modules continue to operate, and the clock continues to be output from the CKIO pin.

In sleep mode, a high-level signal is output at the STATUS1 pin, and a low-level signal at the STATUS0 pin.

(2) Exit from Sleep Mode

Sleep mode is exited by means of an interrupt (NMI, IRL, IRQ, GPIO, or peripheral module) or a reset.

In sleep mode, interrupts are accepted even if the BL bit in SR is 1. If necessary, SPC and SSR should be saved to the stack before executing the SLEEP instruction.

(a) Exit by interrupt

When an NMI, IRL, IRQ, GPIO, or peripheral module interrupt is generated, sleep mode is exited and the interrupt exception handling is executed. The code corresponding to the interrupt source is set in INTEVT.

(b) Exit by reset

Sleep mode is exited by means of a power-on or manual reset.

(1) Transition to Deep Sleep Mode

If a SLEEP instruction is executed when the STBY bit in STBCR is cleared to 0 and the DSLP bit in STBCR2 is set to 1, the chip switches from the program execution state to deep sleep mode. After execution of the SLEEP instruction, the CPU halts but its register contents are retained. Except for the DMAC, peripheral modules continue to operate. The clock continues to be output to the CKIO pin, but all bus access (including auto refresh) stops. When using memory that requires refreshing, select self-refreshing mode prior to making the transition to deep sleep mode. Terminate DMA transfers prior to making the transition to deep sleep mode. If you make a transition to deep sleep mode while DMA transfers are in progress, the results of those transfers cannot be guaranteed.

In deep sleep mode, a high-level signal is output at the STATUS1 pin, and a low-level signal at the STATUS0 pin.

(2) Exit from Deep Sleep Mode

As with sleep mode, deep sleep mode is exited by means of an interrupt (NMI, IRL, IRQ, GPIO, or peripheral module) or a reset.

14.3.3 Software Standby Mode

(1) Transition to Software Standby Mode

If a SLEEP instruction is executed when the STBY bit in STBCR is set to 1, the chip switches from the program execution state to software standby mode. In software standby mode, the clock and peripheral modules halt as well as the CPU. Clock output from the CKIO pin is also stopped.

The CPU and cache register contents are retained. Some peripheral module registers are initialized.

The procedure for a transition to software standby mode is shown below.

(a) Clear the TME bit in WTCSR of the WDT to 0, and stop the WDT.
Set the initial value for up counting in WTCNT of the WDT, and set the clock to be used for up counting in bits CKS2 to CKS0 in WTCSR.
(b) Set the STBY bit in STBCR to 1, then execute a SLEEP instruction.
(c) When software standby mode is entered and the LSI's internal clock stops, a low-level signal is output at the STATUS1 pin, and a high-level signal at the STATUS0 pin.

Software standby mode is exited by means of an interrupt (NMI, IRL, IRQ*, or GPIO) or a reset via the $\overline{\text{RESET}}$ and $\overline{\text{MRESET}}$ pins.

Note: * Software standby mode can be cleared by an IRQ4 or IRQ5 interrupt, but not by an IRQ6 or IRQ7 interrupt.

(a) Exit by interrupt

A hot start can be performed by means of the on-chip WDT. When an NMI, IRL, IRQ, or GPIO interrupt is detected, the WDT starts counting. After the count overflows, clocks are supplied to the entire LSI, software standby mode is exited, and the STATUS1 and STATUS0 pins both go low. Interrupt exception handling is then executed, and the code corresponding to the interrupt source is set in INTEVT. In standby mode, interrupts are accepted even if the BL bit in the SR register is 1, and so, if necessary, SPC and SSR should be saved to the stack before executing the SLEEP instruction.

The phase of the CKIO pin clock output may be unstable immediately after an interrupt is detected, until software standby mode is exited.

(b) Exit by reset

Software standby mode is exited by means of a reset (power-on or manual) via the $\overline{\text{RESET}}$ pin. The $\overline{\text{RESET}}$ pin should be held low until clock oscillation stabilizes. The internal clock continues to be output at the CKIO pin.

(3) Clock Pause Function

In software standby mode, it is possible to stop or change the frequency of the clock input from the EXTAL pin. This function is used as follows.

(a) Enter software standby mode following the transition procedure described above.

(b) When software standby mode is entered and the LSI's internal clock stops, a low-level signal is output at the STATUS1 pin, and a high-level signal at the STATUS0 pin.

(c) The input clock is stopped, or its frequency changed, after the STATUS1 pin goes low and the STATUS0 pin high.

(d) When the frequency is changed, input an NMI interrupt after the change. When the clock is stopped, input an NMI interrupt after applying the clock.

(e) After the time set in the WDT, clock supply begins inside the LSI, the STATUS1 and STATUS0 pins both go low, and operation is resumed from interrupt exception handling.

(1) Transition to Module Standby Function

Setting 1 to the MSTP6 to MSTP4 and MSTP2 bits in STBCR and STBCR2, and the CSTP31 to CSTP0 bits in CLKSTP00 enables the clock supply to the corresponding peripheral modules to be halted. Use of this function allows power consumption in sleep mode to be further reduced.

In the module standby state, the peripheral module external pins retain their states prior to halting of the modules, and most registers retain their states prior to halting of the modules.

(2) Exit from Module Standby Function

In the case of STBCR and STBCR2, the module standby function is exited by writing 0 to the MSTP6 to MSTP4 and MSTP2 bits. In the case of CLKSTP00, the module standby function is exited by writing 1 to the corresponding bit in CLKSTPCLR00.

The module standby function is also exited by means of a power-on reset via the $\overline{\text{RESET}}$ pin or a power-on reset caused by watchdog timer overflow.

14.3.5 Hardware Standby Mode

(1) Transition to Hardware Standby Mode

Setting the CA pin level low effects a transition to hardware standby mode. Note that the CA pin must be continuously held low while in hardware standby mode. In this mode, all modules stop, as in software standby mode selected using the SLEEP instruction.

Hardware standby mode differs from software standby mode in the following points:

(a) Interrupts and manual resets are not available.
(b) All output pins other than the STATUS pins are in the high-impedance state and the pull-up resistance is off.

Operation when a low-level is input to the CA pin in software standby mode depends on the CPG status, as follows:

(a) In software standby mode

The clock remains stopped and a transition is made to the hardware standby state.

(b) When WDT is operating at the time software standby mode is exited by interrupt

After software standby mode is momentarily exited and the CPU restarts operation, a transition is made to hardware standby mode.

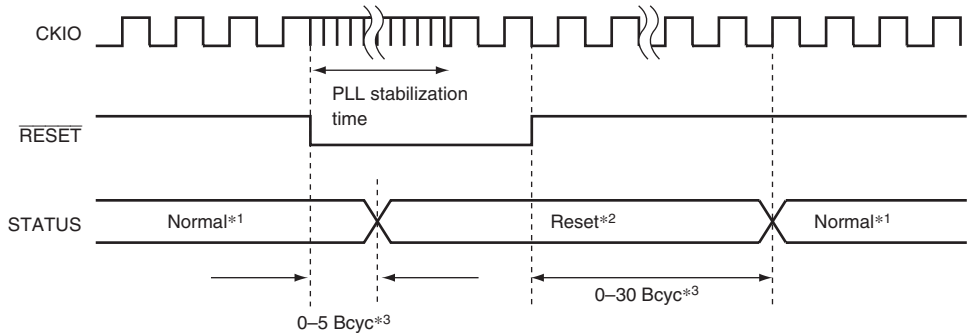
Hardware standby mode is exited by means of a power-on reset via the $\overline{\text{RESET}}$ pin.

14.3.6 STATUS Pin Change Timing

The timing of STATUS1 and STATUS0 pin changes is shown below.

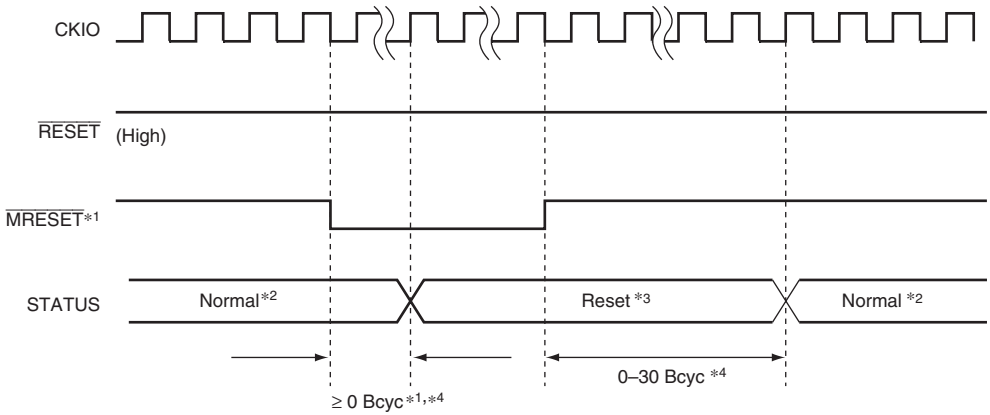
(1) In Reset

(a) Power-on reset



Notes: 1. Normal : LL (STATUS1 is low and STATUS0 is low)
2. Reset : HH (STATUS1 is high and STATUS0 is high)
3. Bcyc : Bus clock cycle

Figure 14.1 STATUS Output in Power-On Reset



Notes: 1. In a manual reset, STATUS = HH (reset) is set and an internal reset started after waiting until the end of the currently executing bus cycle
2. Normal :LL (STATUS1 is low and STATUS0 is low)
3. Reset :HH (STATUS1 is high and STATUS0 is high)
4. Bcyc : Bus clock cycle

Figure 14.2 STATUS Output in Manual Reset

(a) Software standby → Interrupt

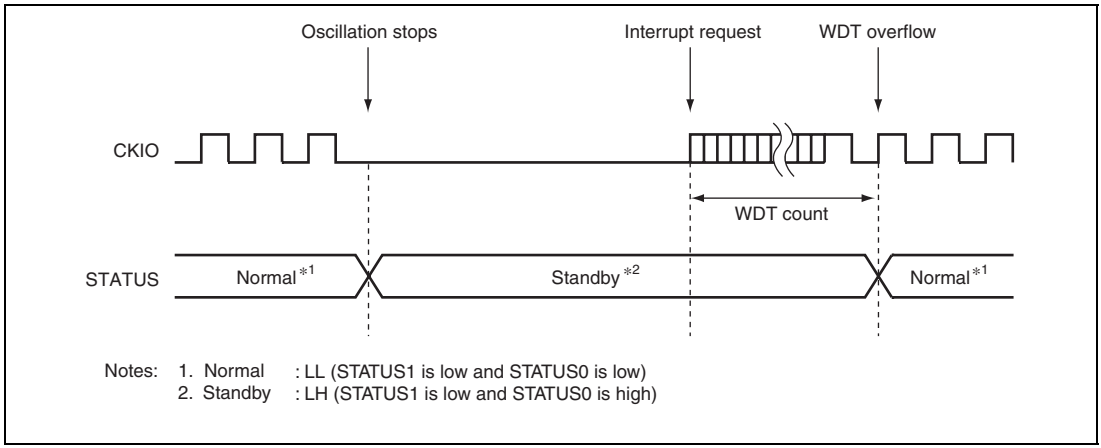


Figure 14.3 STATUS Output in Sequence of Software Standby → Interrupt

(b) Software standby → Power-on reset

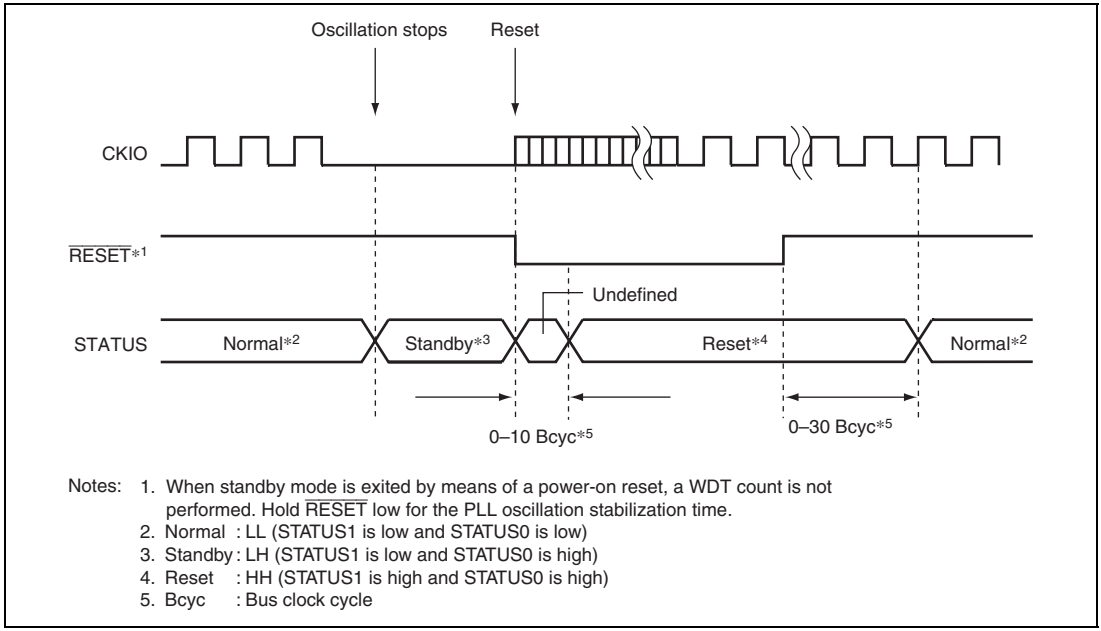
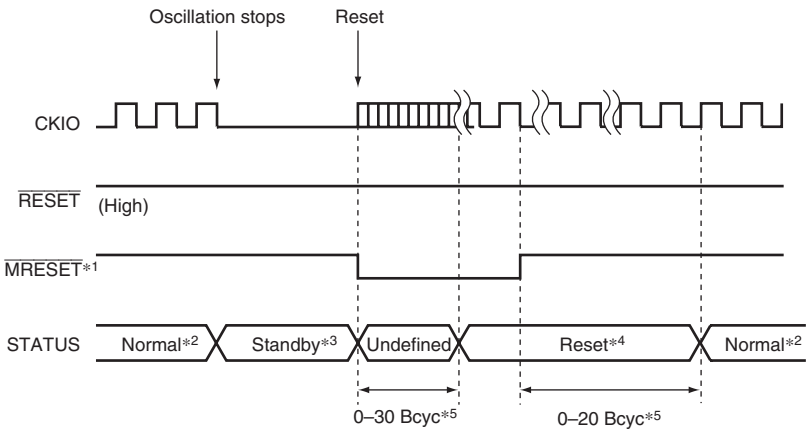


Figure 14.4 STATUS Output in Sequence of Software Standby → Power-On Reset

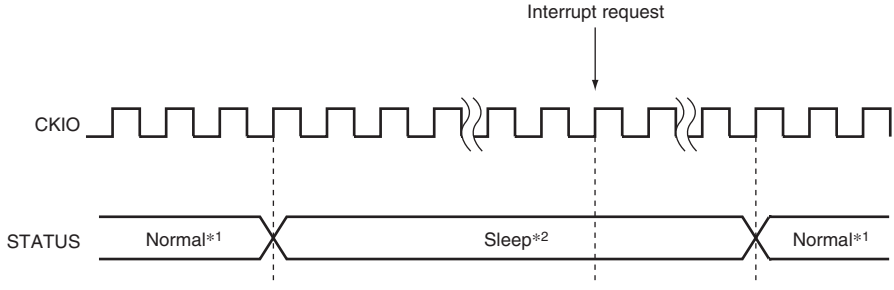


Notes: 1. When standby mode is exited by means of a manual reset, a WDT count is not performed. Hold $\overline{\text{MRESET}}$ low for the PLL oscillation stabilization time.
2. Normal : LL (STATUS1 is low and STATUS0 is low)
3. Standby : LH (STATUS1 is low and STATUS0 is high)
4. Reset : HH (STATUS1 is high and STATUS0 is high)
5. Bcyc : Bus clock cycle

Figure 14.5 STATUS Output in Sequence of Software Standby → Manual Reset

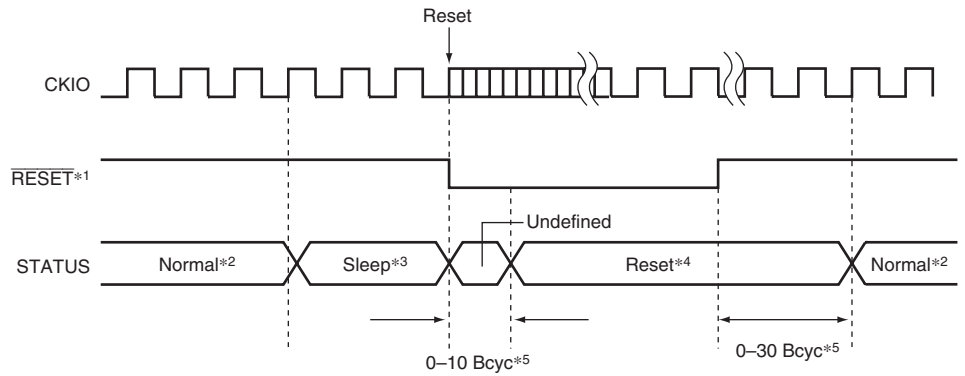
(3) In Exit from Sleep Mode

(a) Sleep → Interrupt



Notes: 1. Normal : LL (STATUS1 is low and STATUS0 is low)
2. Sleep : HL (STATUS1 is high and STATUS0 is low)

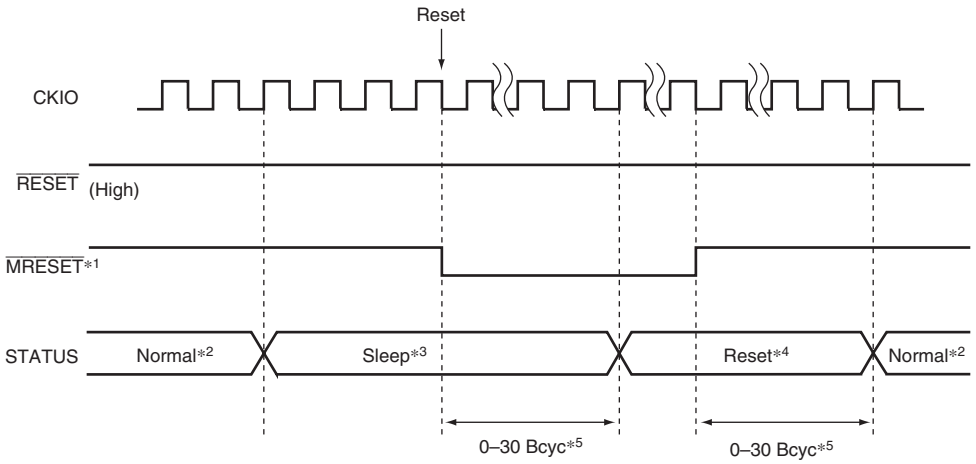
Figure 14.6 STATUS Output in Sequence of Sleep → Interrupt



Notes: 1. When sleep mode is exited by means of a power-on reset, hold RESET low for the oscillation stabilization time.
2. Normal : LL (STATUS1 is low and STATUS0 is low)
3. Sleep : HL (STATUS1 is high and STATUS0 is low)
4. Reset : HH (STATUS1 is high and STATUS0 is high)
5. Bcyc : Bus clock cycle

Figure 14.7 STATUS Output in Sequence of Sleep → Power-On Reset

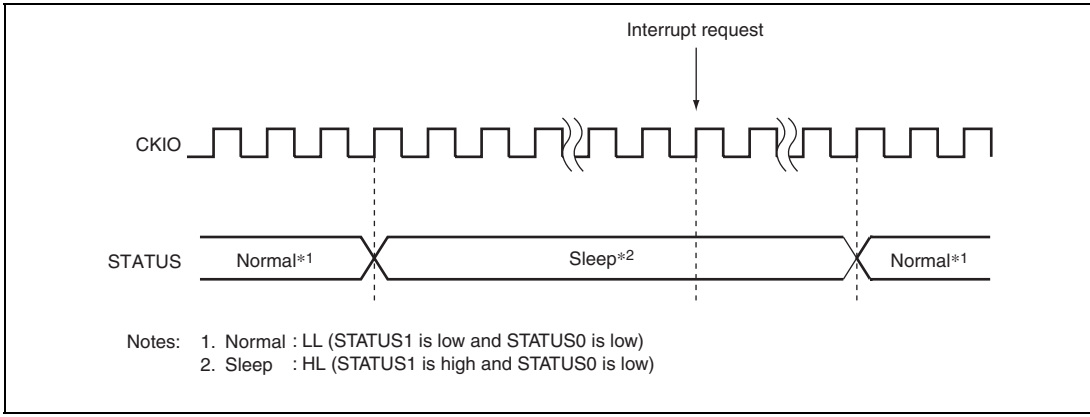
(c) Sleep → Manual reset



Notes: 1. Hold MRESET low until STATUS = reset.
2. Normal : LL (STATUS1 is low and STATUS0 is low)
3. Sleep : HL (STATUS1 is high and STATUS0 is low)
4. Reset : HH (STATUS1 is high and STATUS0 is high)
5. Bcyc : Bus clock cycle

Figure 14.8 STATUS Output in Sequence of Sleep → Manual Reset

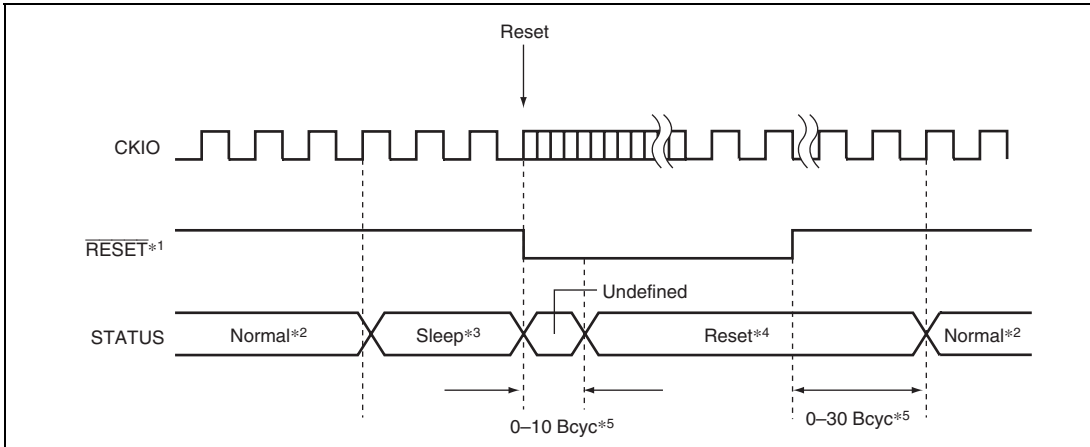
(a) Deep sleep → Interrupt



Notes: 1. Normal : LL (STATUS1 is low and STATUS0 is low)
2. Sleep : HL (STATUS1 is high and STATUS0 is low)

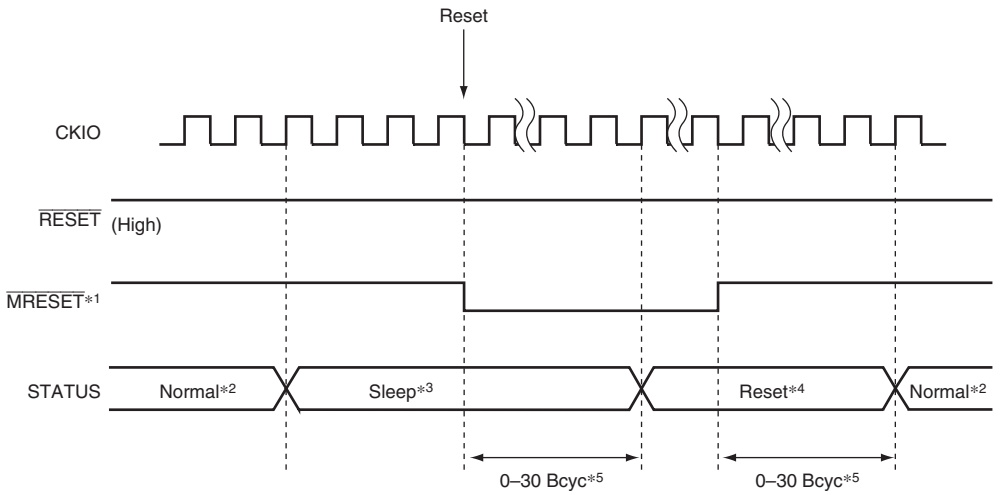
Figure 14.9 STATUS Output in Sequence of Deep Sleep → Interrupt

(b) Deep sleep → Power-on reset



Notes: 1. When deep sleep mode is exited by means of a power-on reset, hold $\overline{\text{RESET}}$ low for the oscillation stabilization time.
2. Normal : LL (STATUS1 is low and STATUS0 is low)
3. Sleep : HL (STATUS1 is high and STATUS0 is low)
4. Reset : HH (STATUS1 is high and STATUS0 is high)
5. Bcyc : Bus clock cycle

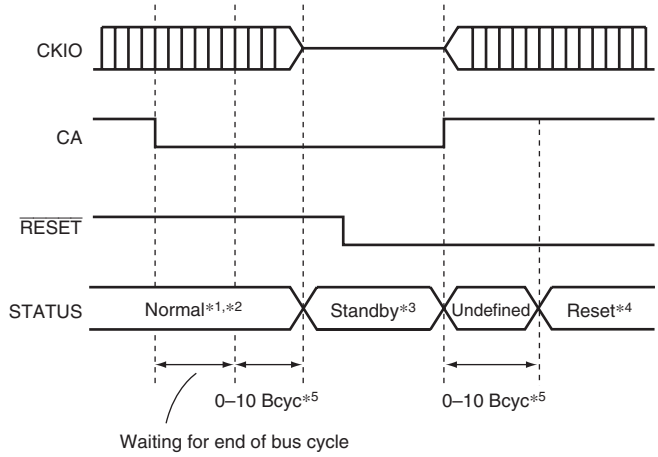
Figure 14.10 STATUS Output in Sequence of Deep Sleep → Power-On Reset



Notes: 1. Hold $\overline{\text{MRESET}}$ low until STATUS = reset.
2. Normal : LL (STATUS1 is low and STATUS0 is low)
3. Sleep : HL (STATUS1 is high and STATUS0 is low)
4. Reset : HH (STATUS1 is high and STATUS0 is high)
5. Bcyc : Bus clock cycle

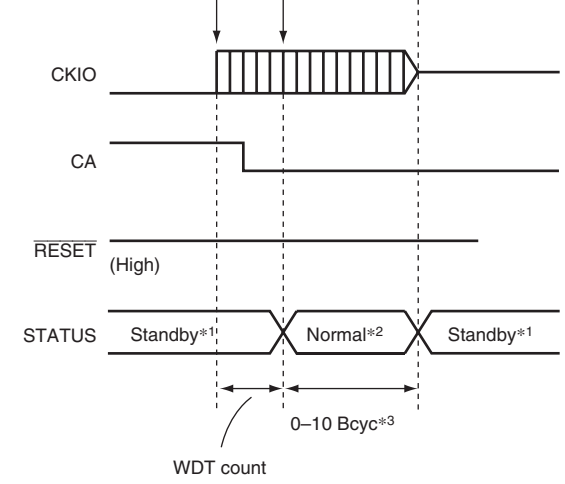
Figure 14.11 STATUS Output in Sequence of Deep Sleep → Manual Reset

Figure 14.12 shows the timing of the signals of the respective pins in hardware standby mode.
The CA pin level must be kept low while in hardware standby mode.
After setting the $\overline{\text{RESET}}$ pin level low, the clock starts when the CA pin level is switched to high.



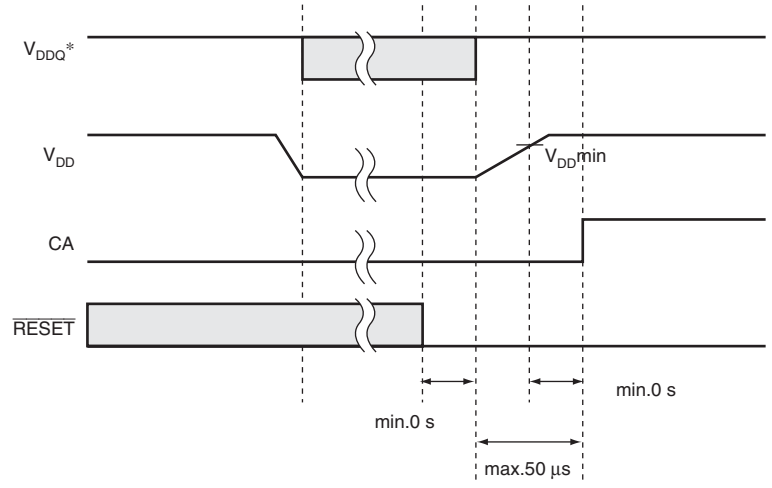
Notes:
1. Same in sleep and reset.
2. Normal : LL (STATUS1 is low and STATUS0 is low)
3. Standby : LH (STATUS1 is low and STATUS0 is high)
4. Reset : HH (STATUS1 is high and STATUS0 is high)
5. Bcyc : Bus clock cycle

Figure 14.12 Hardware Standby Mode Timing (When CA = Low in Normal Operation)



Notes: 1. Standby : LH (STATUS1 is low and STATUS0 is high)
2. Normal : LL (STATUS1 is low and STATUS0 is low)
3. Bcyc : Bus clock cycle

Figure 14.13 Hardware Standby Mode Timing (When CA = Low in WDT Operation)



Note: * V_{DDQ}, $V_{DD\text{-}CPG}$

Figure 14.14 Timing when Power Is Off

14.4.1 Note on Current Consumption

After a power-on reset, the current consumption may exceed the maximum value for sleep mode or standby mode during the period until one or more of the arithmetic operation or floating-point operation instructions listed below is executed.

1. Arithmetic operation instructions
   MAC.W, MAC.L
2. Floating-point operation instructions
   — When FPSCR.PR = 0
   FADD, FSUB, FMUL, FMAC, FLOAT, FTRC, FDIV, FSQRT, FIPR, FTRV
   — When FPSCR.PR = 1
   FADD, FSUB, FMUL, FLOAT, FTRC, FDIV, FSQRT, FCNVSD, FCNVDS

Workaround: After a power-on reset, execute one or more of the above instructions before transitioning to sleep mode or standby mode.

Example: To reduce the effect on FPSCR, arrange the following two instructions starting at H'A0000000.

```
Address         Instruction String
H'A0000000      FLDI1       FR0
H'A0000002      FADD        FR0, FR0 ; FLDI1 FR0 loads 1 into FR0,
        :         :                  ; so the cause and flag bits of FPSCR are not set to 1.
```

This LSI includes an on-chip 32-bit timer unit (TMU) which has three channels (channels 0 to 2).

15.1 Features

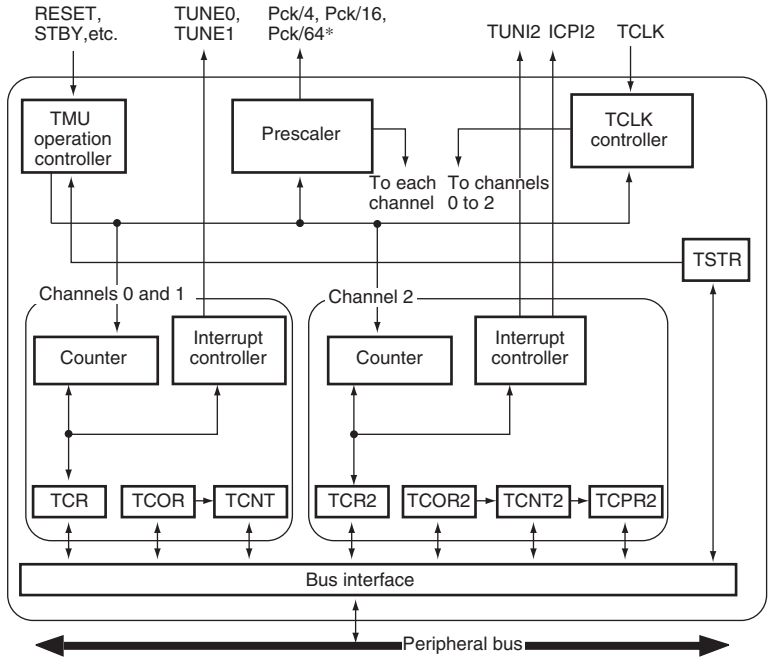
The TMU has the following features.

- Auto-reload type 32-bit down-counter provided for each channel
- Input capture function provided in channel 2
- Selection of rising edge or falling edge as external clock input edge when external clock is selected or input capture function is used for each channel
- 32-bit timer constant register for auto-reload use, readable/writable at any time, and 32-bit down-counter provided for each channel
- Selection of six counter input clocks for each channel

  External clock (TCLK), five peripheral clocks (Pck/4, Pck/16, Pck/64, Pck/256, and Pck/1024) (Pck is the peripheral clock)
- Two interrupt sources

  One underflow source (each channel) and one input capture source (channel 2)
- DMAC data transfer request capability

  In channel 2, a data transfer request is sent to the DMAC when an input capture interrupt is generated.



Note: * Internal signals with 1/4, 1/16, or 1/64 of the Pck frequency and supplied to the on-chip peripheral modules.

Legend:
TSTR : Timer start register
TCOR : Timer constant register
TCNT : Timer counter
TCR : Timer control register
TCPR2 : Input capture register 2 (only in channel 2)

Figure 15.1 Block Diagram of TMU

15.2 Input/Output Pins

Table 15.1 shows the TMU pin configuration.

Table 15.1 Pin Configuration

Pin Name	Abbreviation	I/O	Function
Clock input	TCLK	Input	External clock input pin/input capture control input pin

The TMU has the following registers. For details on the addresses of these registers and the state of registers in each operating mode, see section 32, List of Registers.

Table 15.2 Register Configuration (1)

Ch.	Register Name	Abbrev.	R/W	P4 Address	Area 7 Address	Size	Sync Clock
Common	Timer start register	TSTR	R/W	H'FFD8 0004	H'1FD8 0004	8	Pck
0	Timer constant register 0	TCOR0	R/W	H'FFD8 0008	H'1FD8 0008	32	Pck
	Timer counter 0	TCNT0	R/W	H'FFD8 000C	H'1FD8 000C	32	Pck
	Timer control register 0	TCR0	R/W	H'FFD8 0010	H'1FD8 0010	16	Pck
1	Timer constant register 1	TCOR1	R/W	H'FFD8 0014	H'1FD8 0014	32	Pck
	Timer counter 1	TCNT1	R/W	H'FFD8 0018	H'1FD8 0018	32	Pck
	Timer control register 1	TCR1	R/W	H'FFD8 001C	H'1FD8 001C	16	Pck
2	Timer constant register 2	TCOR2	R/W	H'FFD8 0020	H'1FD8 0020	32	Pck
	Timer counter 2	TCNT2	R/W	H'FFD8 0024	H'1FD8 0024	32	Pck
	Timer control register 2	TCR2	R/W	H'FFD8 0028	H'1FD8 0028	16	Pck
	Input capture register 2	TCPR2	R	H'FFD8 002C	H'1FD8 002C	32	Pck

Ch.	Register Name	Abbrev.	Power-on Reset by RESET Pin/ WDT/H-UDI	Manual Reset by RESET Pin/WDT/ Multiple Exception	Sleep by Sleep Instruction/ Deep Sleep	Standby by Hardware	Standby by Software/ Each Module
Common	Timer start register	TSTR	H'00	H'00	Retained	*	H'00
0	Timer constant register 0	TCOR0	H'FFFF FFFF	H'FFFF FFFF	Retained		Retained
	Timer counter 0	TCNT0	H'FFFF FFFF	H'FFFF FFFF	Retained		Retained
	Timer control register 0	TCR0	H'0000	H'0000	Retained		Retained
1	Timer constant register 1	TCOR1	H'FFFF FFFF	H'FFFF FFFF	Retained		Retained
	Timer counter 1	TCNT1	H'FFFF FFFF	H'FFFF FFFF	Retained		Retained
	Timer control register 1	TCR1	H'0000	H'0000	Retained		Retained
2	Timer constant register 2	TCOR2	H'FFFF FFFF	H'FFFF FFFF	Retained		Retained
	Timer counter 2	TCNT2	H'FFFF FFFF	H'FFFF FFFF	Retained		Retained
	Timer control register 2	TCR2	H'0000	H'0000	Retained		Retained
	Input capture register 2	TCPR2	Retained	Retained	Retained		Retained

Note: * After exiting hardware standby mode, this LSI enters the power-on reset state caused by the RESET pin.

15.3.1 Timer Start Register (TSTR)

TSTR is an 8-bit readable/writable register that specifies whether TCNT in each channel is operated or stopped.

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

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2	STR2	0	R/W	Counter Start 2 Specifies whether TCNT2 is operated or stopped. 0: TCNT2 count operation is stopped 1: TCNT2 performs count operation

1	STR1	0	R/W	Counter Start 1 Specifies whether TCNT1 is operated or stopped. 0: TCNT1 count operation is stopped 1: TCNT1 performs count operation
0	STR0	0	R/W	Counter Start 0 Specifies whether TCNT0 is operated or stopped. 0: TCNT0 count operation is stopped 1: TCNT0 performs count operation

15.3.2 Timer Constant Register (TCORn) (n = 0 to 2)

The TCOR registers are 32-bit readable/writable registers. When a TCNT counter underflows while counting down, the TCOR value is set in that TCNT, which continues counting down from the set value.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
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

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
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

15.3.3 Timer Counter (TCNTn) (n = 0 to 2)

The TCNT registers are 32-bit readable/writable registers. Each TCNT counts down on the input clock selected by the TPSC2 to TPSC0 bits in TCR.

When a TCNT counter underflows while counting down, the UNF flag is set in TCR of the corresponding channel. At the same time, the TCOR value is set in TCNT, and the count-down operation continues from the set value.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
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

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
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

The TCR registers are 16-bit readable/writable registers. Each TCR selects the count clock, specifies the edge when an external clock is selected, and controls interrupt generation when the flag indicating TCNT underflow is set to 1. TCR2 is also used for input capture control and control of interrupt generation in the event of input capture.

- TCR0 and TCR1

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	UNF	-	-	UNIE	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0
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/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

- TCR2

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	ICPF	UNF	ICPE1	ICPE0	UNIE	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0
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

Bit	Bit Name	Initial Value	R/W	Description
15 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9	ICPF*[1]	0	R/W	Input Capture Interrupt Flag Status flag, provided in channel 2 only, which indicates the occurrence of input capture. 0: Input capture has not occurred [Clearing condition] When 0 is written to ICPF 1: Input capture has occurred [Setting condition] When input capture occurs*[2]
8	UNF	0	R/W	Underflow Flag Status flag that indicates the occurrence of TCNT underflow. 0: TCNT has not underflowed [Clearing condition] When 0 is written to UNF 1: TCNT has underflowed [Setting condition] When TCNT underflows*[2]

7	ICPE1*1	0	R/W	Input Capture Control
6	ICPE0*1	0	R/W	These bits, provided in channel 2 only, specify whether the input capture function is used, and control enabling or disabling of interrupt generation when the function is used. When the input capture function is used, a data transfer request is sent to the DMAC in the event of input capture. The CKEG bits specify whether the rising edge or falling edge of the TCLK pin is used to set the TCNT2 value in TCPR2. The TCNT2 value is set in TCPR2 only when the ICPF bit in TCR2 is 0. When the ICPF bit is 1, TCPR2 is not set in the event of input capture. When input capture occurs, a DMAC transfer request is generated regardless of the value of the ICPF bit. However, a new DMAC transfer request is not generated until processing of the previous request is finished. 00: Input capture function is not used. 01: Setting prohibited 10: Input capture function is used, but interrupt due to input capture (TICPI2) is not enabled. Data transfer request is sent to the DMAC in the event of input capture. 11: Input capture function is used, and interrupt due to input capture (TICPI2) is enabled. Data transfer request is sent to the DMAC in the event of input capture.
5	UNIE	0	R/W	Underflow Interrupt Control Controls enabling or disabling of interrupt generation when the UNF status flag is set to 1, indicating TCNT underflow. 0: Interrupt due to underflow (TUNI) is disabled 1: Interrupt due to underflow (TUNI) is enabled
4	CKEG1	0	R/W	Clock Edge 1 and 0
3	CKEG0	0	R/W	These bits select the external clock input edge when an external clock is selected or the input capture function is used. 00: Count/input capture register set on rising edge 01: Count/input capture register set on falling edge 1X: Count/input capture register set on both rising and falling edges

2	TPSC2	0	R/W	Timer Prescaler 2 to 0
1	TPSC1	0	R/W	These bits select the TCNT count clock.
0	TPSC0	0	R/W	000: Counts on Pck/4 001: Counts on Pck/16 010: Counts on Pck/64 011: Counts on Pck/256 100: Counts on Pck/1024 101: Setting prohibited 110: Setting prohibited 111: Counts on external clock (TCLK)

Notes: X: Don't care

1. Reserved bit in channel 0 or 1 (initial value is 0, and can only be read).
2. Writing 1 does not change the value; the previous value is retained.

15.3.5 Input Capture Register 2 (TCPR2)

TCPR2 is a 32-bit read-only register for use with the input capture function, provided only in channel 2. The input capture function is controlled by means of the ICPE and CKEG bits in TCR2. When input capture occurs, the TCNT2 value is copied into TCPR2. The value is set in TCPR2 only when the ICPF bit in TCR2 is 0.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
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
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

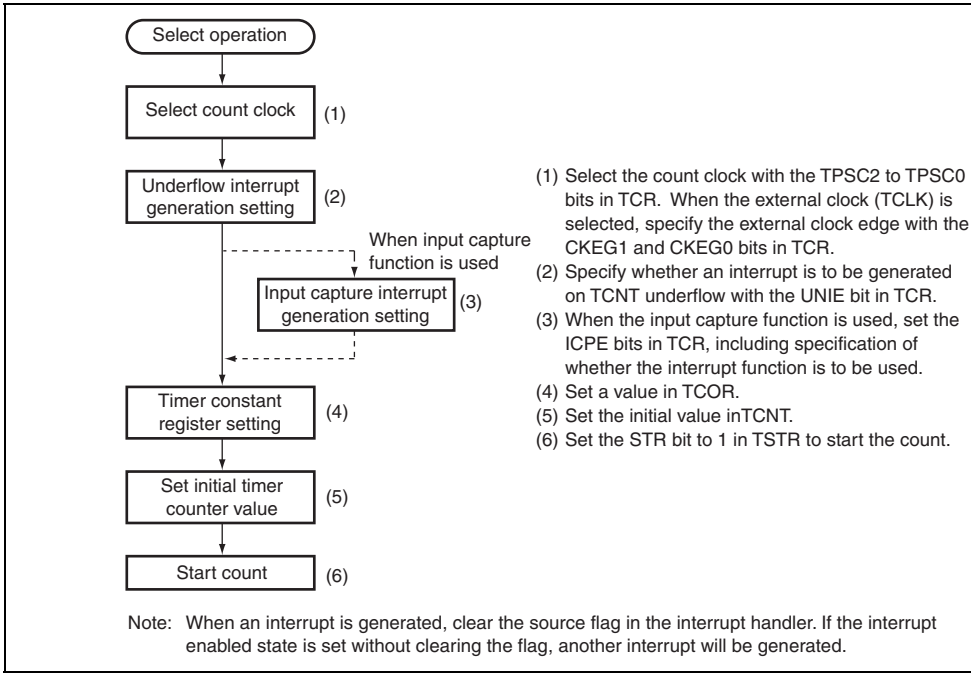
Each channel has a 32-bit timer counter (TCNT) and a 32-bit timer constant register (TCOR). Each TCNT performs count-down operation. The channels have an auto-reload function that allows cyclic count operations, and can also perform external event counting. Channel 2 also has an input capture function.

15.4.1 Counter Operation

When one of bits STR0 to STR2 in TSTR is set to 1, the TCNT for the corresponding channel starts counting. When TCNT underflows, the UNF flag in TCR is set. If the UNIE bit in TCR is set to 1 at this time, an interrupt request is sent to the CPU. At the same time, the value is copied from TCOR into TCNT, and the count-down continues (auto-reload function).

(1) Example of Count Operation Setting Procedure

Figure 15.2 shows an example of the count operation setting procedure.



Note: When an interrupt is generated, clear the source flag in the interrupt handler. If the interrupt enabled state is set without clearing the flag, another interrupt will be generated.

Figure 15.2 Example of Count Operation Setting Procedure

Figure 15.3 shows the TCNT auto-reload operation.

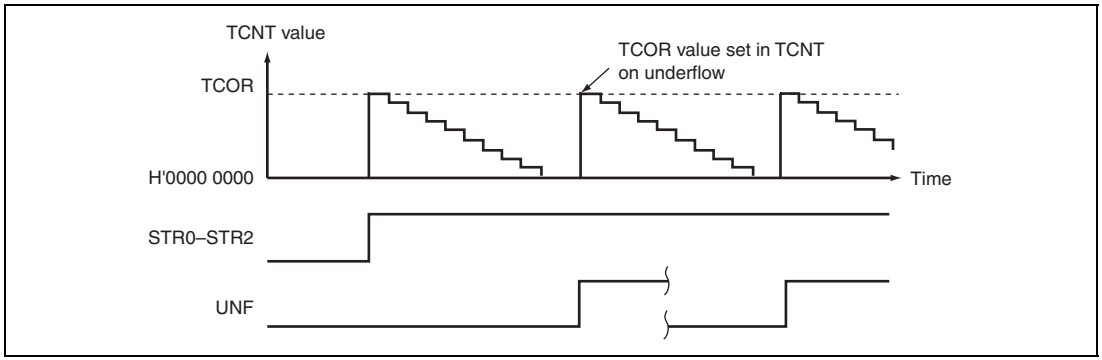


Figure 15.3 TCNT Auto-Reload Operation

(3) TCNT Count Timing

- Operating on internal clock

  Any of five count clocks (Pck/4, Pck/16, Pck/64, Pck/256, or Pck/1024) scaled from the peripheral clock can be selected as the count clock by means of the TPSC2 to TPSC0 bits in TCR.

  Figure 15.4 shows the timing in this case.

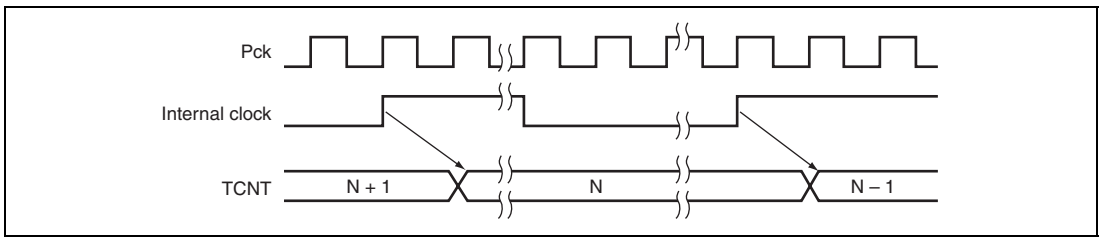


Figure 15.4 Count Timing when Operating on Internal Clock

The external clock pin (TCLK) input can be selected as the timer clock by means of the TPSC2 to TPSC0 bits in TCR. The detected edge (rising, falling, or both edges) can be selected with the CKEG1 and CKEG0 bits in TCR.

Figure 15.5 shows the timing for both-edge detection.

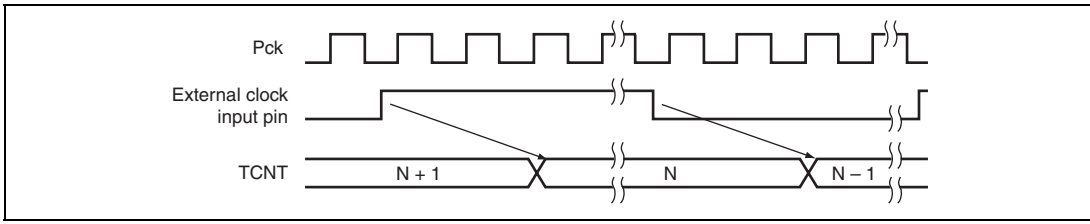


Figure 15.5 Count Timing when Operating on External Clock

15.4.2 Input Capture Function

Channel 2 has an input capture function.

The procedure for using the input capture function is as follows:

1. Use bits TPSC2 to TPSC0 in TCR to set an internal clock as the timer operating clock.
2. Use bits ICPE1 and ICPE0 in TCR to specify whether to use the input capture function and when using this function whether to generate interrupts.
3. Use bits CKEG1 and CKEG0 in TCR to specify whether the rising or falling edge of the TCLK pin is to be used to set the TCNT value in TCPR2.

When input capture occurs, the TCNT2 value is set in TCPR2 only when the ICPF bit in TCR2 is 0. A new DMAC transfer request is not generated until processing of the previous request is finished.

Figure 15.6 shows the operation timing when the input capture function is used (with TCLK rising edge detection).

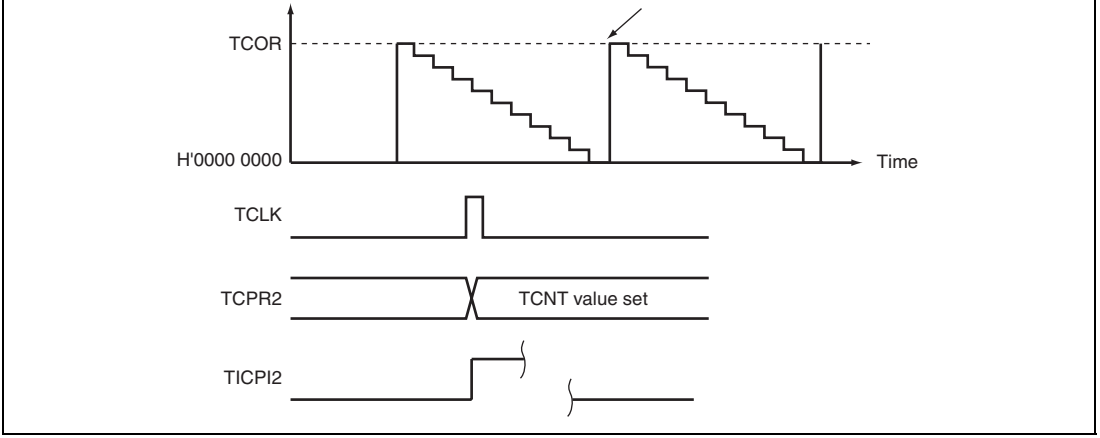


Figure 15.6 Operation Timing when Using Input Capture Function

15.5 Interrupts

There are four TMU interrupt sources: underflow interrupts and the input capture interrupt when the input capture function is used. Underflow interrupts are generated on each of the channels, and input capture interrupts on channel 2 only.

An underflow interrupt request is generated (for each channel) when both the UNF bit and the interrupt enable bit (UNIE) for that channel are set to 1.

When the input capture function is used and an input capture request is generated, an interrupt is requested if the ICPF bit in TCR2 is 1 and the input capture control bits (ICPE1 and ICPE0) in TCR2 are both set to 11.

The TMU interrupt sources are summarized in table 15.3.

Table 15.3 TMU Interrupt Sources

Channel	Interrupt Source	Description	Priority
0	TUNI0	Underflow interrupt 0	High
1	TUNI1	Underflow interrupt 1	↕
2	TUNI2	Underflow interrupt 2	↕
	TICPI2	Input capture interrupt 2	Low

15.6.1 Register Writes

When writing to a TMU register, timer count operation must be stopped by clearing the start bit (STR2 to STR0) for the relevant channel in TSTR.

Note that TSTR can be written to, and the UNF and ICPF bits in TCR can be cleared while the count is in progress. When the flags (UNF and ICPF) are cleared while the count is in progress, make sure not to change the values of bits other than those being cleared.

15.6.2 Reading from TCNT

Reading from TCNT is performed synchronously with the timer count operation. Note that when the timer count operation is performed simultaneously with reading from a register, the synchronous processing causes the TCNT value before the count-down operation to be read as the TCNT value.

15.6.3 External Clock Frequency

Ensure that the external clock (TCLK) frequency for each channel does not exceed Pck/8.

The unit has 2 major modes of operation. It can be configured as a four-channel timer/counter unit that contains a 32 bit free running timer as a common time-stamp for four 32-bit capture/compare registers.

Alternatively it can be configured as four 16-bit timer/counters with count enables. In this mode there are four independent 16-bit incrementing/decrementing blocks. Each channel can then be set up to output a signal when the compare time is reached or to store the timer value when an input edge is received. This latter case also supports up/down counting on some channels.

16.1 Features

- 32-bit free-running timer
- Four channels of output compare or input capture
- 4-channel 16-bit timer/counter
- Interrupt on capture, compare, and overflow
- Programmable pin/edge polarity
- Programmable timer clock
- Independent clocks for 16-bit timers
- Rotary switches supported

Figure 16.1 shows a block diagram of the CMT.

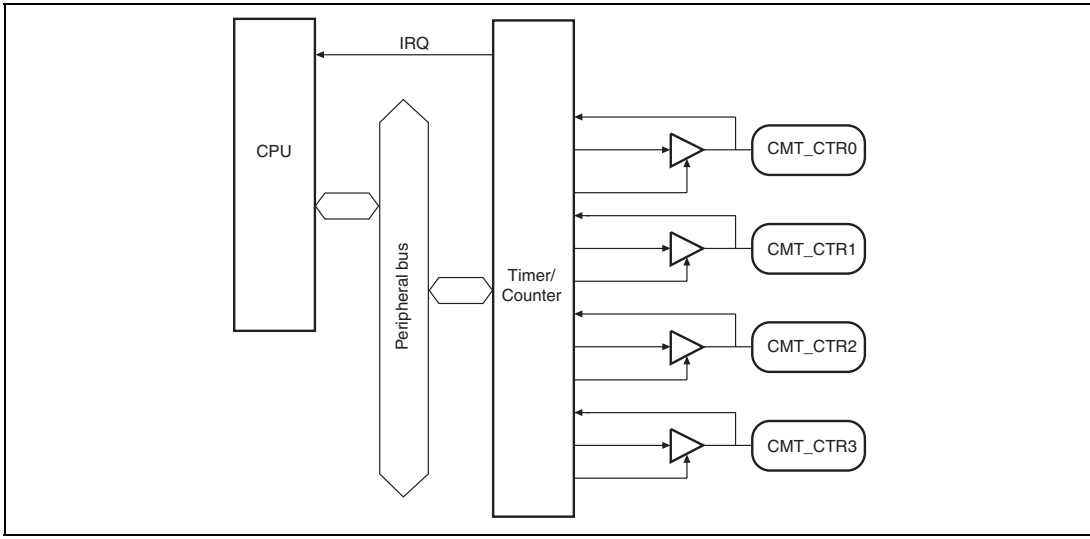


Figure 16.1 Block Diagram of CMT

Table 16.1 shows the CMT pin configuration.

Table 16.1 Pin Configuration

Pin Name	Number of Pins	I/O	Function
CMT_CTR0 to CMT_CTR3	4	Input/Output	Multi-function timer/counter input/output

16.3 Register Descriptions

The CMT has the following registers. For details on the addresses of these registers and the state of registers in each operating mode, see section 32, List of Registers.

Table 16.2 Register Configuration (1)

Ch.	Register Name	Abbrev.	R/W	P4 Address	Area 7 Address	Size	Sync Clock
Common	Configuration register	CMTCFG	R/W	H'FE1C 0000	H'1E1C 0000	32	Pck
	Free-running timer	CMTFRT	R	H'FE1C 0004	H'1E1C 0004	32	Pck
	Control register	CMTCTL	R/W	H'FE1C 0008	H'1E1C 0008	32	Pck
	IRQ status register	CMTIRQS	R/W	H'FE1C 000C	H'1E1C 000C	32	Pck
0	Channel 0 time register	CMTCH0T	R/W	H'FE1C 0010	H'1E1C 0010	32	Pck
	Channel 0 stop time register	CMTCH0ST	R/W	H'FE1C 0020	H'1E1C 0020	32	Pck
	Channel 0 timer/counter	CMTCH0C	R/W	H'FE1C 0030	H'1E1C 0030	32	Pck
1	Channel 1 time register	CMTCH1T	R/W	H'FE1C 0014	H'1E1C 0014	32	Pck
	Channel 1 stop time register	CMTCH1ST	R/W	H'FE1C 0024	H'1E1C 0024	32	Pck
	Channel 1 timer/counter	CMTCH1C	R/W	H'FE1C 0034	H'1E1C 0034	32	Pck
2	Channel 2 time register	CMTCH2T	R/W	H'FE1C 0018	H'1E1C 0018	32	Pck
	Channel 2 stop time register	CMTCH2ST	R/W	H'FE1C 0028	H'1E1C 0028	32	Pck
	Channel 2 timer/counter	CMTCH2C	R/W	H'FE1C 0038	H'1E1C 0038	32	Pck
3	Channel 3 time register	CMTCH3T	R/W	H'FE1C 001C	H'1E1C 001C	32	Pck
	Channel 3 stop time register	CMTCH3ST	R/W	H'FE1C 002C	H'1E1C 002C	32	Pck
	Channel 3 timer/counter	CMTCH3C	R/W	H'FE1C 003C	H'1E1C 003C	32	Pck

Ch.	Register Name	Abbrev.	Power-on Reset by RESET Pin/WDT/ H-UDI	Manual Reset by RESET Pin/WDT/ Multiple Exception	Sleep by Sleep Instruction/ Deep Sleep	Standby by Hardware	Standby by Software/ Each Module
Common	Configuration register	CMTCFG	H'0000 0000	H'0000 0000	Retained	*	Retained
	Free-running timer	CMTFRT	H'0000 0000	H'0000 0000	Retained		Retained
	Control register	CMTCTL	H'0000 0000	H'0000 0000	Retained		Retained
	IRQ status register	CMTIRQS	H'0000 0000	H'0000 0000	Retained		Retained
0	Channel 0 time register	CMTCH0T	H'0000 0000	H'0000 0000	Retained		Retained
	Channel 0 stop time register	CMTCH0ST	H'0000 0000	H'0000 0000	Retained		Retained
	Channel 0 timer/counter	CMTCH0C	H'0000 0000	H'0000 0000	Retained		Retained
1	Channel 1 time register	CMTCH1T	H'0000 0000	H'0000 0000	Retained		Retained
	Channel 1 stop time register	CMTCH1ST	H'0000 0000	H'0000 0000	Retained		Retained
	Channel 1 timer/counter	CMTCH1C	H'0000 0000	H'0000 0000	Retained		Retained
2	Channel 2 time register	CMTCH2T	H'0000 0000	H'0000 0000	Retained		Retained
	Channel 2 stop time register	CMTCH2ST	H'0000 0000	H'0000 0000	Retained		Retained
	Channel 2 timer/counter	CMTCH2C	H'0000 0000	H'0000 0000	Retained		Retained
3	Channel 3 time register	CMTCH3T	H'0000 0000	H'0000 0000	Retained		Retained
	Channel 3 stop time register	CMTCH3ST	H'0000 0000	H'0000 0000	Retained		Retained
	Channel 3 timer/counter	CMTCH3C	H'0000 0000	H'0000 0000	Retained		Retained

Note: * After exiting hardware standby mode, this LSI enters the power-on reset state caused by the RESET pin.

CMTCFG is a 32-bit readable/writable register. The possible operations for a pin are timer compare, timer input capture, up or down count, and capture input, where one pin is used for capture while the second is used to enable the count.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	ROT2	ROT0
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ED3		ED2		ED1		ED0		-	FRCM	FRTM	T23			T01	
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	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 18	—	—	R	Reserved These bits can only be read from. The write value should always be 0.
17	ROT2	0	R/W	Channel 2, 3 Rotation Enable Only set to 1 when operating in updown-counter mode (T01=11, T23=011), otherwise this bit is disabled (ROT2=0). When this bit is set to 1, this indicates that pins of channels 2 and 3 are operating in rotary mode. This is an encoding of CMT_CTR2 pin and CMT_CTR3 pin to generate up and down signals to the counter. Counter 3 needs to be disabled by clearing to 0 the TE3 bit in CMTCTL.
16	ROT0	0	R/W	Channel 0,1 Rotation Enable Only set to 1 when operating in updown-counter mode (T01=11, T23=011), otherwise this bit is disabled (ROT0=0). When this bit is set to 1, this indicates that pins of channels 0 and 1 are operating in rotary mode. This is an encoding of CMT_CTR0 pin and CMT_CTR1 pin to generate up and down signals to the counter. Counter 1 needs to be disabled by clearing to 0 the TE1 bit in CMTCTL.

15, 14	ED3	All 0	R/W	Channel 3 Pin Active Control [Input capture mode] 00: Edge detection of CMT_CTR3 pin disabled 01: Edge detection on rising edge of CMT_CTR3 pin input 10: Edge detection on falling edge of CMT_CTR3 pin input 11: Edge detection on either edge of CMT_CTR3 pin input [Output compare mode] 00: Reserved* 01: 1 is output to CMT_CTR3 pin during active period 10: 0 is output to CMT_CTR3 pin during active period 11: Reserved
13, 12	ED2	All 0	R/W	Channel 2 Pin Active Control [Input capture mode] 00: Edge detection of CMT_CTR2 pin disabled 01: Edge detection on rising edge of CMT_CTR2 pin input 10: Edge detection on falling edge of CMT_CTR2 pin input 11: Edge detection on either edge of CMT_CTR2 pin input [Output compare mode] 00: Reserved* 01: 1 is output to CMT_CTR2 pin during active period 10: 0 is output to CMT_CTR2 pin during active period 11: Reserved
11, 10	ED1	All 0	R/W	Channel 1 Pin Active Control [Input capture mode] 00: Edge detection of CMT_CTR1 pin disabled 01: Edge detection on rising edge of CMT_CTR1 pin input 10: Edge detection on falling edge of CMT_CTR1 pin input 11: Edge detection on either edge of CMT_CTR1 pin input [Output compare mode] 00: Reserved* 01: 1 is output to CMT_CTR1 pin during active period 10: 0 is output to CMT_CTR1 pin during active period 11: Reserved

9, 8	ED0	All 0	R/W	Channel 0 Pin Active Control [Input capture mode] 00: Edge detection of CMT_CTR0 pin disabled 01: Edge detection on rising edge of CMT_CTR0 pin input 10: Edge detection on falling edge of CMT_CTR0 pin input 11: Edge detection on either edge of CMT_CTR0 pin input [Output compare mode] 00: Reserved* 01: 1 is output to CMT_CTR0 pin during active period 10: 0 is output to CMT_CTR0 pin during active period 11: Reserved
7	—	0	R	Reserved This bit can only be read from. The write value should always be 0.
6	FRCM	0	R/W	Free-Running Control Mode In 16-bit timer/counter mode, when bits T23 are set to 100, this bit determines whether the up-counter uses a free-running counter or input capture on channel 3. 0: External clock (up-counter with input capture) 1: Internal clock (free-running up-counter)
5	FRTM	0	R/W	Free-Running Timer Mode Determines whether the timer works as a common 32-bit free-running timer or four independent 16-bit timers/counters. When this bit is set to 1, settings of bits 4 to 0 in this register become disabled. 0: 16-bit timer/counter mode 1: 32-bit free-running timer (FRT) mode

Note: * When these channels be used, be sure to set up values other than Reserved.

4 to 2	T23	All 0	R/W	Timer 2, 3 Configuration These bits are only used in 16-bit timer/counter mode. These bits control the use of CMT_CTR2 and CMT_CTR3 pins. CMT_CTR2 pin is mapped to timer 2/counter 2 and CMT_CTR3 pin is mapped to timer 3/counter 3. Configuration modes for channels 2 and 3: 000: Timers 2 and 3 001: Up-counter 2 and timer 3 010: Up-counters 2 and 3 011: Updown-counter 2 100: Up-counter with input capture 101: Reserved 110: Reserved 111: Reserved Note: Up-counter 2 is a sub-set of updown-counter 2.
1, 0	T01	All 0	R/W	Timer 0,1 Configuration These bits are only used in 16-bit timer/counter mode. These bits control the use of CMT_CTR0 and CMT_CTR1 pins. CMT_CTR0 pin is mapped to timer 0/counter 0 and CMT_CTR1 pin is mapped to timer 1/counter 1. Configuration modes for channels 0 and 1: 00: Timers 0 and 1 01: Up-counter 0 and timer 1 10: Up-counters 0 and 1 11: Updown-counter 0 Note: Up-counter 0 is a sub-set of updown-counter 0.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	FRT															
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
	FRT															
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	Bit Name	Initial Value	R/W	Description
31 to 0	FRT	All 0	R	Free-Running Timer These bits indicate the current value of the free-running timer (FRT).

16.3.3 Control Register (CMTCTL)

CMTCTL is a 32-bit readable/writable register that controls interrupts, makes settings for the clocks, and selects the operating mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TE3	TE2	TE1	TE0	IOE3	IOE2	IOE1	IOE0	ICE3	ICE2	ICE1	ICE0	IEE3	IEE2	IEE1	IEE0
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
	CC3		CC2		CC1		CC0		SI3	SI2	SI1	SI0	OP3	OP2	OP1	OP0
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	Bit Name	Initial Value	R/W	Description
31	TE3	0	R/W	Timer Enable Enables the counting of each of the 16-bit counters. If these bits are inactive when operating in timer mode or in counter mode, the counters are reset to 0. In updown-counter mode, a counter for each pair (counter 1 or counter 3) needs to be disabled (TE1=0, TE3=0). 0: Counting disabled; counter will be reset to H'000 1: Counter is incremented
30	TE2	0	R/W	
29	TE1	0	R/W	
28	TE0	0	R/W	

27	IOE3	0	R/W	Channel 3 to 0 Interrupt Overflow Enable
26	IOE2	0	R/W	These bits enable an interrupt to be generated when the relevant IOn bit is set in the IRQ status register.
25	IOE1	0	R/W	
24	IOE0	0	R/W	0: Interrupt generation disabled 1: Interrupt generation enabled
23	ICE3	0	R/W	Channel 3 to 0 Interrupt Compare Enable
22	ICE2	0	R/W	These bits enable an interrupt to be generated when the relevant ICn bit is set in the IRQ status register.
21	ICE1	0	R/W	
20	ICE0	0	R/W	0: Interrupt generation disabled 1: Interrupt generation enabled
19	IEE3	0	R/W	Channel 3 to 0 Interrupt Edge Enable
18	IEE2	0	R/W	These bits enable an interrupt to be generated when the relevant IEn bit is set in the IRQ status register.
17	IEE1	0	R/W	
16	IEE0	0	R/W	0: Interrupt generation disabled 1: Interrupt generation enabled When a channel is in outup compare mode, the corresponding IEEn has to be set to 0.
15, 14	CC3	All 0	R/W	Timer Clock Control Channel 3 These bits specify the clock input for the 16-bit timer/counter in channel 3.* 00: Clock for timer 3 is 1/32 of source clock 01: Clock for timer 3 is 1/128 of source clock 10: Clock for timer 3 is 1/512 of source clock 11: Clock for timer 3 is 1/1024 of source clock Set the same value as the CC0 bit when using 16-bit input capture mode.
13, 12	CC2	All 0	R/W	Timer Clock Control Channel 2 These bits specify the clock input for the 16-bit timer/counter in channel 2.* 00: Clock for timer 2 is 1/32 of source clock 01: Clock for timer 2 is 1/128 of source clock 10: Clock for timer 2 is 1/512 of source clock 11: Clock for timer 2 is 1/1024 of source clock Set the same value as the CC0 bit when using 16-bit input capture mode.

11, 10	CC1	All 0	R/W	Timer Clock Control Channel 1 These bits specify the clock input for the 16-bit timer/counter in channel 1.* 00: Clock for timer 1 is 1/32 of source clock 01: Clock for timer 1 is 1/128 of source clock 10: Clock for timer 1 is 1/512 of source clock 11: Clock for timer 1 is 1/1024 of source clock Set the same value as the CC0 bit when using 16-bit input capture mode.
9, 8	CC0	All 0	R/W	Free-Running Timer Clock Control This clock is used for the 32-bit free-running timer (FRT) and also for the 16-bit timer/counter in channel 0.* 00: Clock for FRT and timer 0 is 1/32 of source clock 01: Clock for FRT and timer 0 is 1/128 of source clock 10: Clock for FRT and timer 0 is 1/512 of source clock 11: Clock for FRT and timer 0 is 1/1024 of source clock
7	SI3	0	R/W	Channel 3 to 0 Stop Ignore For the channel n, these bits determine whether in output compare mode with 32-bit free-running timer mode, the output remains active for half the maximum time or until the stop value is reached. 0: Output remains active until the channel n stop time value is reached 1: Output remains active for half the total time of the FRT
6	SI2	0	R/W	
5	SI1	0	R/W	
4	SI0	0	R/W	
3	OP3	0	R/W	Channel 3 to 0 Operation For the channel n, if in timer mode, these bits determine whether the timer is used in output compare or input capture mode. 0: Input capture mode 1: Output compare mode When a channel is in output compare mode, the corresponding IEEn bits has to be set to 0.
2	OP2	0	R/W	
1	OP1	0	R/W	
0	OP0	0	R/W	

Notes: n = 3 to 0

* The source clock is the peripheral clock (Pck). The clock which divided from the source clock is the timer/counter resolution of the channel.

CMTIRQS, once set, can only be cleared by a write. Writing 0 to these bits clears the interrupt status bits. These conditions only create an interrupt if the relevant interrupt enable bit is set.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
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
	-	-	-	-	IO3	IO2	IO1	IO0	IC3	IC2	IC1	IC0	IE3	IE2	IE1	IE0
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

Bit	Bit Name	Initial Value	R/W	Description
31 to 12	—	—	R	Reserved These bits can only be read from. The write value should always be 0.
11	IO3	0	R/WC0*	Channel 3 to 0 Interrupt Overflow A bit for each channel indicates if the up-counters or updown-counters have wrapped i.e. overflowed from H'FFFF to H'0000 or underflowed from H'0000 to H'FFFF. 0: The count has not overflowed or underflowed 1: The count has overflowed or underflowed
10	IO2	0	R/WC0*	
9	IO1	0	R/WC0*	
8	IO0	0	R/WC0*	
7	IC3	0	R/WC0*	Channel 3 to 0 Interrupt Compare A bit for each channel indicates whether in timer mode, the free-running timer has become equal to the channel times. 0: Timer has not become equal to the channel time value 1: Timer has become equal to the channel time value
6	IC2	0	R/WC0*	
5	IC1	0	R/WC0*	
4	IC0	0	R/WC0*	
3	IE3	0	R/WC0*	Channel 3 to 0 Interrupt Edge A bit for each channel indicates whether an edge that wil cause an action (active edge) has been defected. 0: Channel 3 to 0 has not received an active edge 1: Channel 3 to 0 has received an active edge
2	IE2	0	R/WC0*	
1	IE1	0	R/WC0*	
0	IE0	0	R/WC0*	

Note: * Writing 0 to clear the bit to 0 is allowed.

In output compare mode, these registers specify the value to be compared with the free-running timer. In input capture mode, this register stores the free-running timer values or the 16-bit timer values on the active edge of the input. Every time an edge is detected, these registers are updated and the new captured value will be saved.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	Channel n time															
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
	Channel n time															
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

16.3.6 Channels 0 to 3 Stop Time Registers (CMTCH0ST to CMTCH3ST)

In output compare mode, these registers specify the value to be compared with the free-running timer. When this value is reached, the timer output is reset to the inactive state.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	Channel n stop time															
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
	Channel n stop time															
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

16.3.7 Channels 0 to 3 Counters (CMTCH0C to CMTCH3C)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
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
	Channel n counter															
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

31 to 16	—	—	R	Reserved These bits can only be read from. The write value should always be 0.
15 to 0	Channel n counter	All 0	R/W	Channel n Counter A register for each channel indicates the current value of the counters. This register can be written to for setting the counter values. Reading from these registers does not affect the count value.

Note: n = 3 to 0

16.4 Operation

The timer consists of two sections of which the first is a 32-bit free-running timer for which the clock can be set to operate between approximately 1 MHz and 30 kHz. The second section consists of four 16-bit counters with the ability to count up and the ability to count down. This counting is controlled through edge detection on the input pins. The timer and counters can be used as counters which count based on inputs or can operate as timers with input capture/output compare. They differ from the 32-bit timer of the FRT in that they are reset to H'0000 when a capture or compare occurs on that channel.

The timer/counter consists of four channels, each of which can be configured as a timer or a counter. In timer mode, each of the four channels has two operating modes: input capture mode and output compare mode.

16.4.1 Edge Detection

The timers and counters are based on edge detection on the input pins. An active edge can be programmed to be a rising edge, falling edge, or both edges. In addition, the edge detection logic can operate in rotary switch mode where the combination of two inputs indicates whether the switch has been turned right or left. This switch indicates increment or decrement of the updown-counter. Edge detection operating on two inputs functions as a pair. The outputs can either work independently for the timers or the up-counters or can work as pairs to indicate up and down to the updown-counters.

In order for an edge to be detected, the pulse must last for at least two cycles of the clock divided from the source clock for that channel, as shown in figure 16.2.

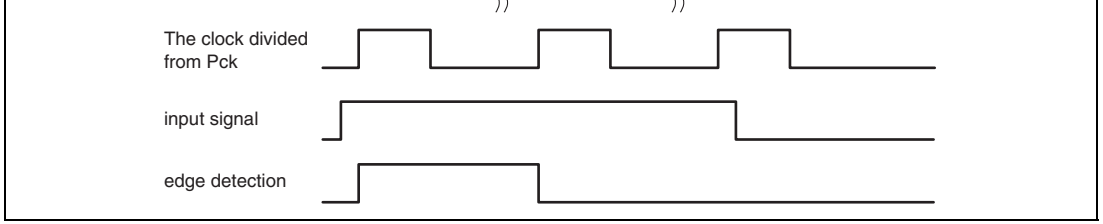


Figure 16.2 Edge Detection

The timing diagram in figure 16.2 shows an edge detection (rising edge). The input pin must be asserted for at least two clock resolution cycles.

16.4.2 32-Bit Timer: Input Capture

When operating in input capture mode, the channel will detect an edge on the input signal. The selection of rising or falling edge is programmable. When this edge is detected, the current value of the free-running timer (FRT) is captured into the channel time register for that channel. In addition, the interrupt edge bit for that channel will be set. In this case, if the interrupt enable bit for that channel is set, an interrupt will be generated.

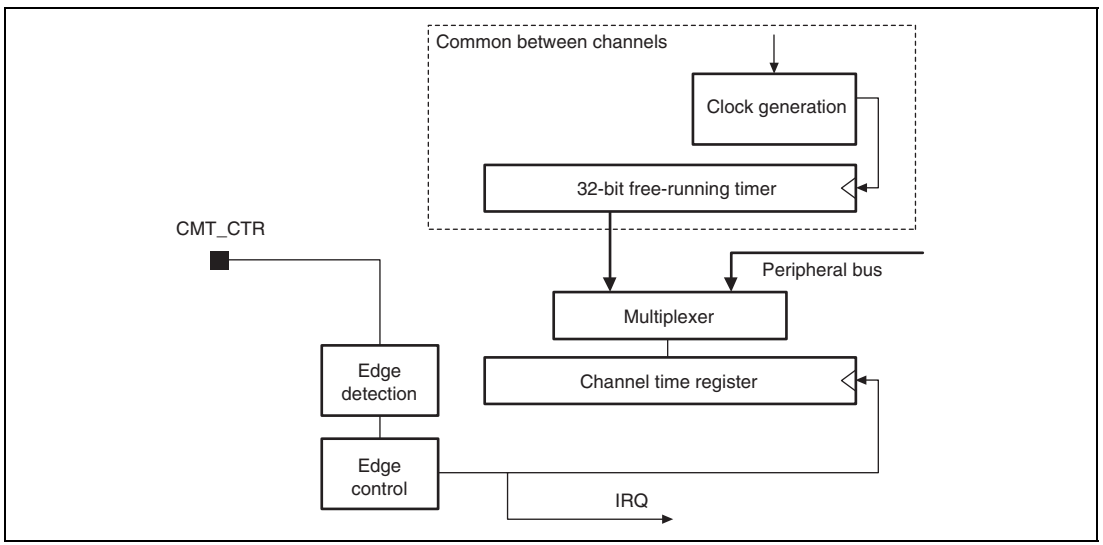
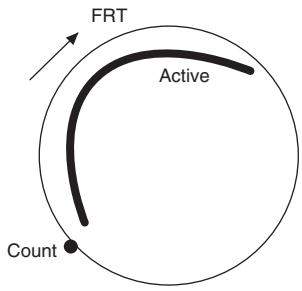


Figure 16.3 32-Bit Timer Mode: Input Capture

When operating in output compare mode, the channel time register is compared with the FRT. When the free-running timer becomes equal to the channel time register, the output will be set to the active state as defined in the pin active control bit. The output will remain in this state until the time reaches half of the maximum channel time, as shown in figure 16.4, or it reaches the channel stop time value specified by the corresponding stop ignore bit in the control register (CMTCTL). At the point where the channel time register is equal to the FRT, the interrupt compare bit will be set, and an interrupt generated if the interrupt enable bit is set.



Note: The output is active until the free-running timer becomes half of the count value.

Figure 16.4 Output Pin Assertion Period

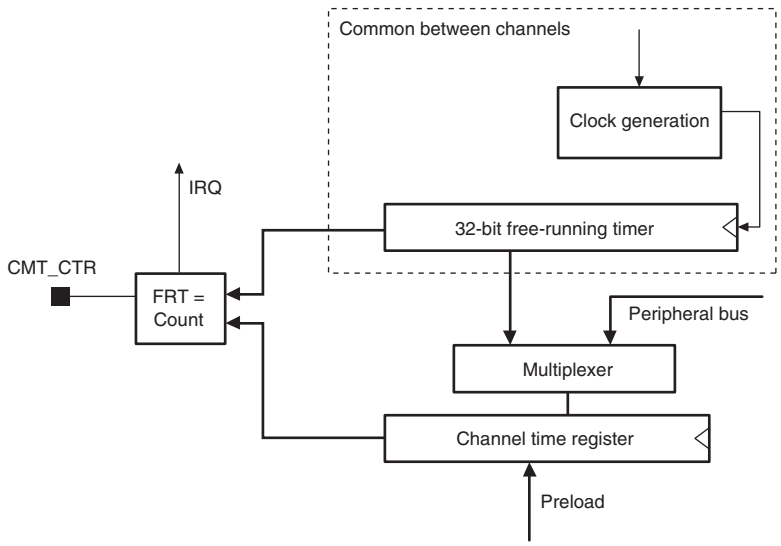


Figure 16.5 32-Bit Timer Mode: Output Compare

When operating in input capture mode, the 16-bit counters will be free-running using the clocks defined by the timer clock control bits. When an active edge defined by the pin active control bit is received, the channel time will be set to the value of that channel's 16-bit counter and the interrupt edge bit will be set. The counter will then be reset to H'0000 and will begin counting again. The counters will be cleared to H'0000 by disabling the timer enable bits. Channel 0 is always enabled for this mode. The other three channels will be enabled if they use the same clock source as channel 0.

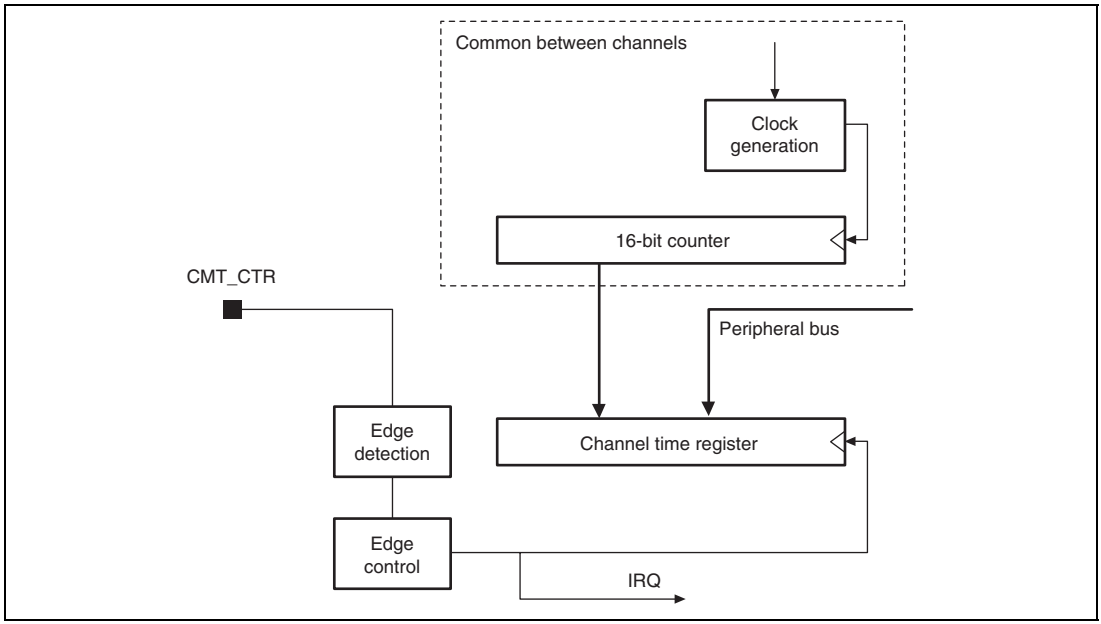


Figure 16.6 16-Bit Timer Mode: Input Capture

When operating in output compare mode, the 16-bit counters will be free-running using the clocks defined by the timer clock control bits. Bits 15 to 0 in the channel time register are compared with the 16-bit counter for that channel. When the values become equal, the output will be inverted from its current state (i.e. toggled), and the interrupt compare bit will be set. The counter will then be reset to H'0000 and will begin counting again. Each time a match on the counter occurs, the output will be toggled. The counters will be cleared to H'0000 by disabling the timer enable bits.

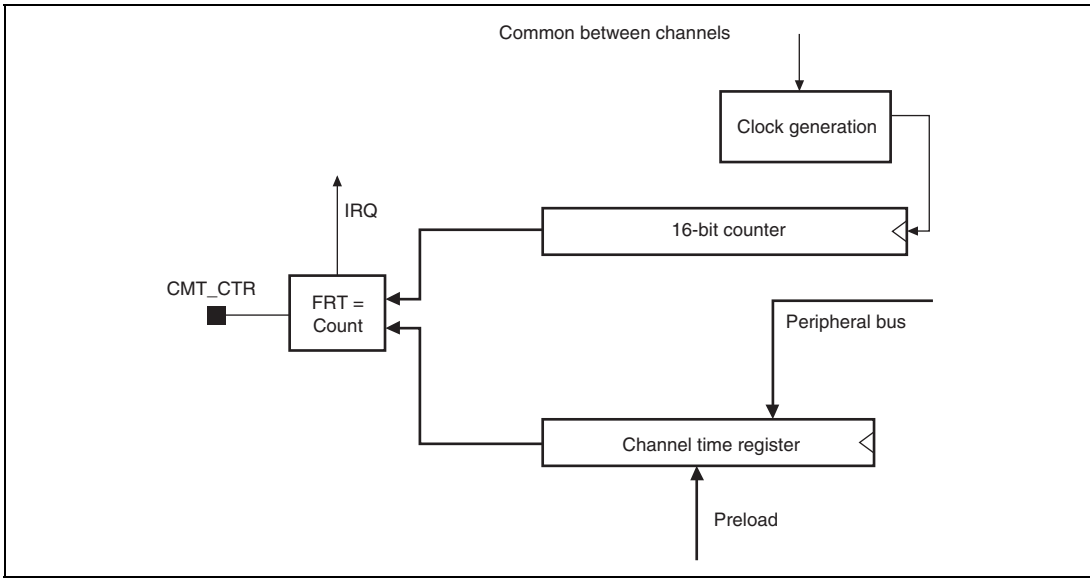


Figure 16.7 16-Bit Timer Mode: Output Compare

Each of the pins can be connected to each of the four up-counters. These counters count up when an active edge is detected on the input pins. The counters can be written to by software to be preloaded and the current value can be read. Two up-counters can also be configured to count both up and down. In this case since two pins are required, the second up-counter within the pair is not available. The pins are then referred to as up (pins 0 and 2) and down (pins 1 and 3). An active edge on these pins will cause the counter either to count up or down, or if both edges are active then the count will remain unchanged. In either up-counter or updown-counter mode, the counter will generate an interrupt if an edge is detected or if the count overflows or underflows.

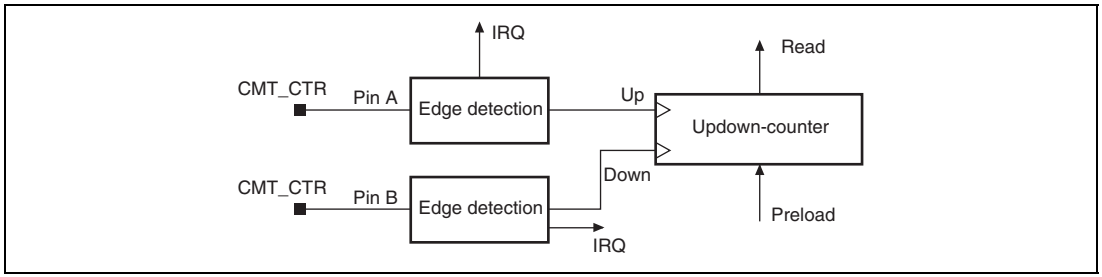


Figure 16.8 Updown-Counter Mode

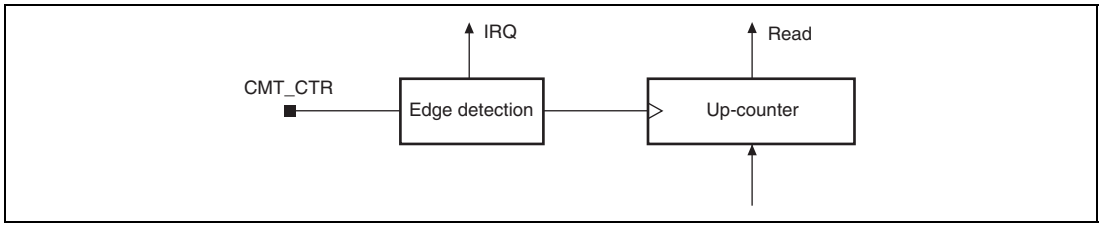


Figure 16.9 Up-Counter Mode

In this mode, the 16-bit counter of channel 2 will operate either as a free-running up-counter or as an up-counter with input capture, depending on the setting of the FRCM bit. If the FRCM bit is cleared to 0, the counter counts up when an active edge is detected on the input pin of channel 3. If the FRCM bit is set to 1, the counter is a free-running counter. The counter will be cleared to H'0000 by disabling the timer enable bit.

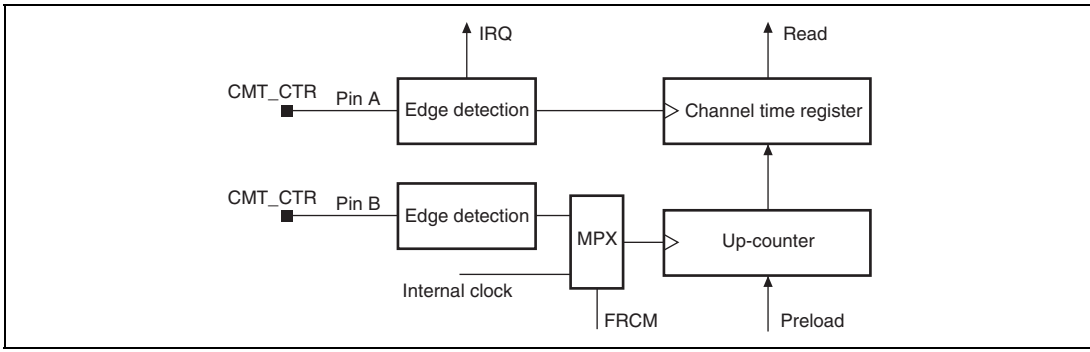


Figure 16.10 Up-Counter with Capture Mode

16.4.8 Interrupts

The Status Register will have the interrupt status bits set for timer operation on either input capture or output compare regardless of the state of the interrupt enable bits. The counters will set an interrupt status bit if the count changes or the counter underflows or overflows. If the interrupt enable bit for a type of interrupt and the interrupt status bit of the same type for the same channel is set then an interrupt is generated.

Each of the two updown-counters can operate in rotary mode. This treats the two input signals as encoded, as shown in figure 16.11. A rotary switch generates the following waveforms depending on the direction. The direction is determined by the value of A when a falling edge is detected on the B input; if A is 1, the direction is left (down is 1) and if A is 0, the direction is right (up is 1). A is pin 0 and pin 2. B is pin 1 and pin 3. The interrupt edge bit in channels 3 to 0 will be set whenever a change in the counter value occurs. If a counter overflow or underflow occurs, the interrupt overflow bit in channels 3 to 0 will be set as well.

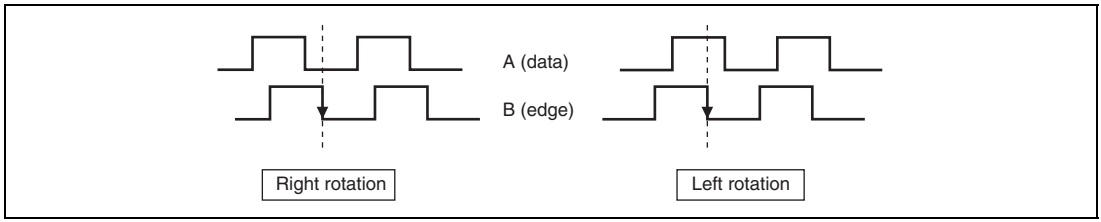


Figure 16.11 Rotary Mode

16.4.10 Timer Frequency

The frequency of the free running timer and the 16-bit timers can be altered under software control to be 1 of 4 frequencies. Each 16-bit timer can have an independent clock.

16.4.11 Standby Mode

CMT allows clock gating to reduce power consumption. The module standby mode can be executed by controlling bit 17 in the Clock Stop Register 00 (CLKSTP00).

To wake up the module, bit 17 in the Clock Stop Clear Register 00 (CLKSTPCLR00) must be enabled. After enabling this bit all access to CMT can be possible.

To power down the module, the following procedure is required.

1. All channels needs to be in input capture mode (CMTCTL.OP3 - OP0=0000).
2. The active edge for each channel needs to be disabled (CMTCFG.ED3 - ED0=00).
3. Disable bit 17 in the Clock Stop Register 00 (CLKSTP00).

(SCIF)

This LSI is equipped with a 3-channel serial communication interface with built-in FIFO buffers (Serial Communication Interface with FIFO: SCIF). The SCIF can perform both asynchronous and synchronous serial communications.

128-stage FIFO buffers are provided for both transmission and reception, enabling fast, efficient, and continuous communication.

Channels 1 and 2 have modem control functions ($\overline{\text{SCIF1\_RTS}}$, $\overline{\text{SCIF2\_RTS}}$, $\overline{\text{SCIF1\_CTS}}$, and $\overline{\text{SCIF2\_CTS}}$).

17.1 Features

The SCIF has the following features.

- Asynchronous serial communication mode

  Serial data communication is executed using an asynchronous system in which synchronization is achieved character by character. Serial data communication can be carried out with standard asynchronous communication chips such as a Universal Asynchronous Receiver/Transmitter (UART) or Asynchronous Communication Interface Adapter (ACIA).

  There is a choice of 8 serial data transfer formats.

  — Data length: 7 or 8 bits

  — Stop bit length: 1 or 2 bits

  — Parity: Even/odd/none

  — Receive error detection: Parity, framing, and overrun errors

  — Break detection: A break is detected when a framing error lasts for more than 1 frame length at Space 0 (low level). When a framing error occurs, a break can also be detected by reading the SCIF_RXD pin level directly from the serial port register (SCSPTR).
- Synchronous serial communication mode

  Serial data communication is synchronized with a clock. Serial data communication can be carried out with other LSIs that have a synchronous communication function.

  There is a single serial data communication format.

  — Data length: 8 bits

  — Receive error detection: Overrun errors

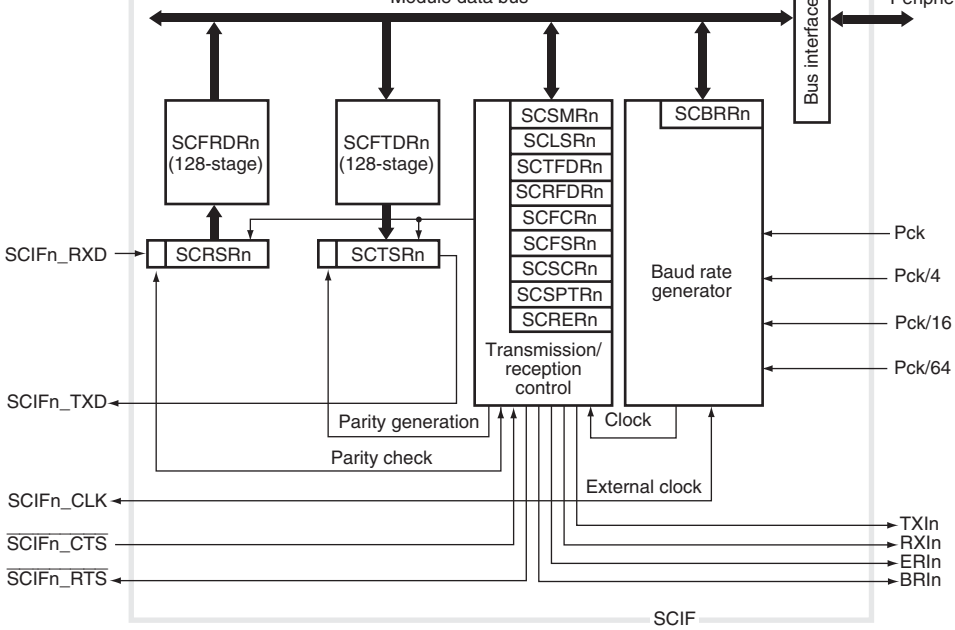
The transmitter and receiver are independent units, enabling transmission and reception to be performed simultaneously.

The transmitter and receiver both have a 128-stage FIFO buffer structure, enabling continuous serial data transmission and reception.

- On-chip baud rate generator allows any bit rate to be selected.
- Choice of serial clock source: internal clock from baud rate generator or external clock from SCIF_CLK pin
- Four interrupt sources

  There are four interrupt sources—transmit-FIFO-data-empty, break, receive-FIFO-data-full, and receive-error—that can issue requests independently.
- The DMA controller (DMAC) can be activated to execute a data transfer by issuing a DMA transfer request in the event of a transmit-FIFO-data-empty or receive-FIFO-data-full interrupt.
- When not in use, the SCIF can be stopped by halting its clock supply to reduce power consumption.
- In asynchronous mode, modem control functions ($\overline{\text{SCIF\_RTS}}$ and $\overline{\text{SCIF\_CTS}}$) are provided.(only in channels 1 and 2)
- The amount of data in the transmit/receive FIFO registers, and the number of receive errors in the receive data in the receive FIFO register, can be ascertained.
- In asynchronous mode, a timeout error (DR) can be detected during reception.

Figure 17.1 shows a block diagram of the SCIF. Figures 17.2 to 17.6 show block diagrams of the I/O ports in SCIF. There are three channels in this LSI. In figures 17.1 to 17.6, the channels are omitted and explained. Note that the $\overline{\text{SCIF\_CTS}}$ and $\overline{\text{SCIF\_RTS}}$ pins are available only in channels 1 and 2 and not in channel 0.

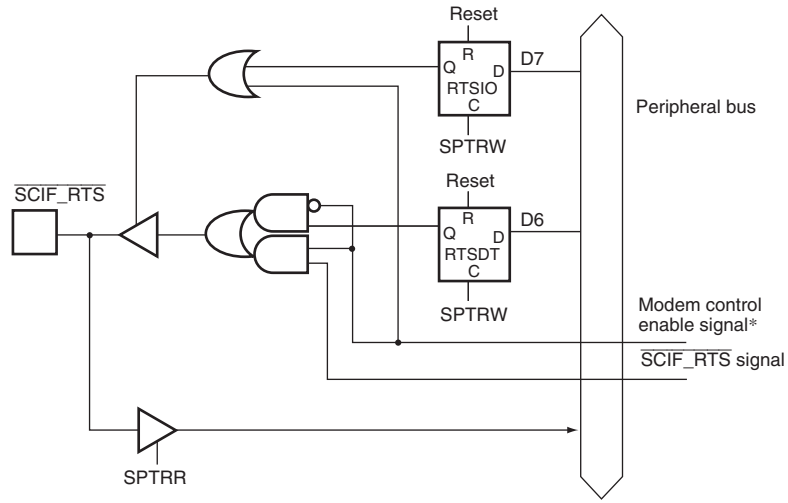


Note: n = 0 to 2
Channel 0 does not have $\overline{\text{SCIF0\_CTS}}$ and $\overline{\text{SCIF0\_RTS}}$.

Legend:

SCRSRn	:Receive shift register	SCBRRn	:Bit rate register
SCFRDRn	:Receive FIFO data register	SCSPTRn	:Serial port register
SCTSRn	:Transmit shift register	SCFCRn	:FIFO control register
SCFTDRn	:Transmit FIFO data register	SCTFDRn	:Transmit FIFO data count register
SCSMRn	:Serial mode register	SCRFDRn	:Receive FIFO data count register
SCSCRn	:Serial control register	SCLSRn	:Line status register
SCFSRn	:Serial status register	SCRERn	:Serial error register

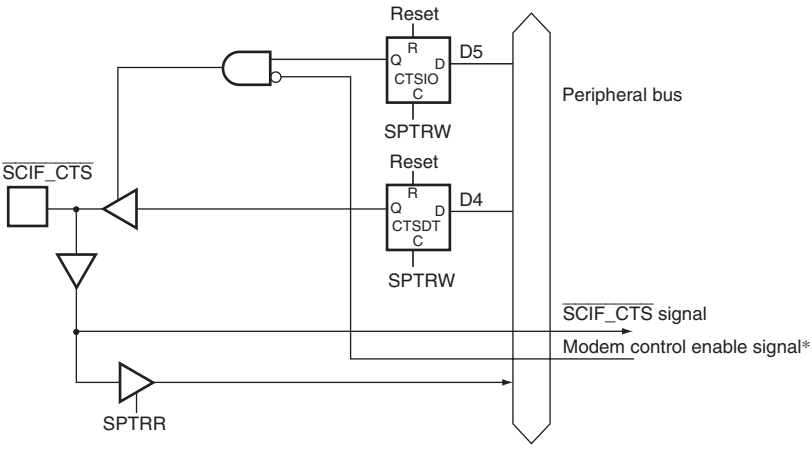
Figure 17.1 Block Diagram of SCIF



SPTRW: Write to SCSPTR
SPTRR: Read from SCSPTR

Note: * The $\overline{\text{SCIF\_RTS}}$ pin function is designated as modem control by the MCE bit in SCFCR.

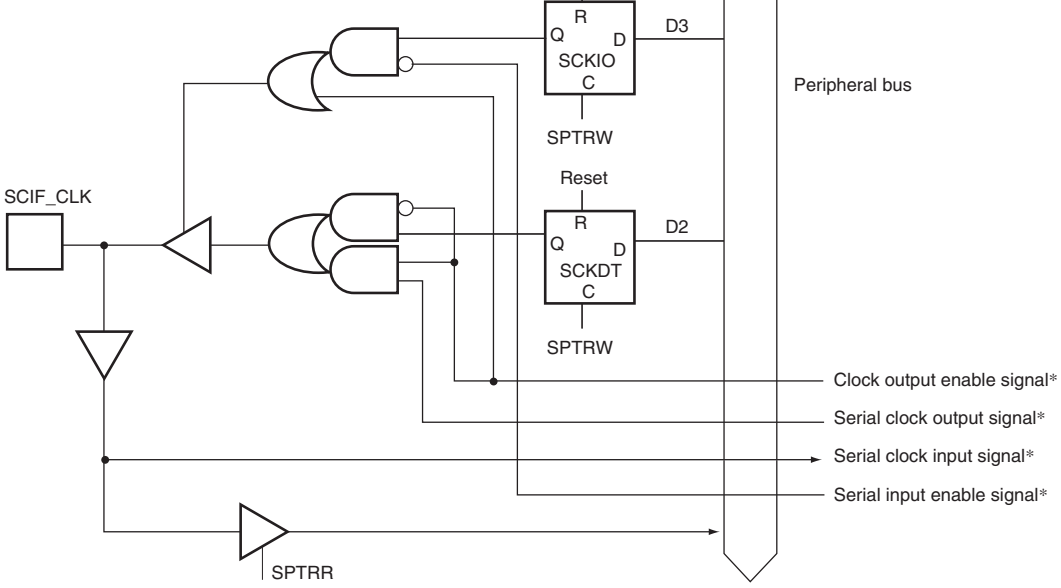
Figure 17.2 $\overline{\text{SCIF\_RTS}}$ Pin (Only in Channels 1 and 2)



SPTRW: Write to SCSPTR
SPTRR: Read from SCSPTR

Note: * The $\overline{\text{SCIF\_CTS}}$ pin function is designated as modem control by the MCE bit in SCFCR.

Figure 17.3 $\overline{\text{SCIF\_CTS}}$ Pin (Only in Channels 1 and 2)



SPTRW: Write to SCSPTR
SPTRR: Read from SCSPTR

Note: * The SCIF_CLK pin function is designated as internal clock output or external clock input by the C/$\overline{A}$ bit in SCSMR and the CKE1 and CKE0 bits in SCSCR.

Figure 17.4 SCIF_CLK Pin

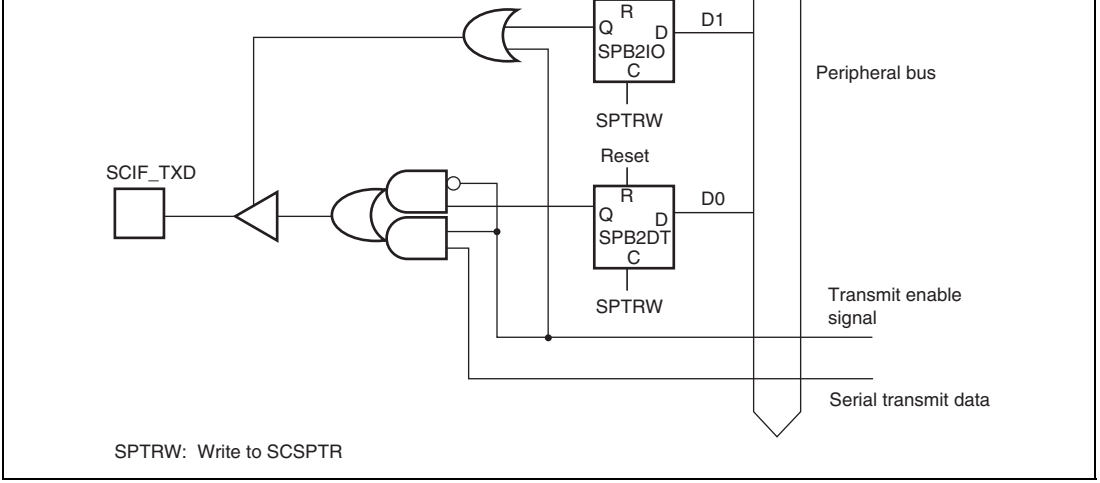


Figure 17.5 SCIF_TXD Pin

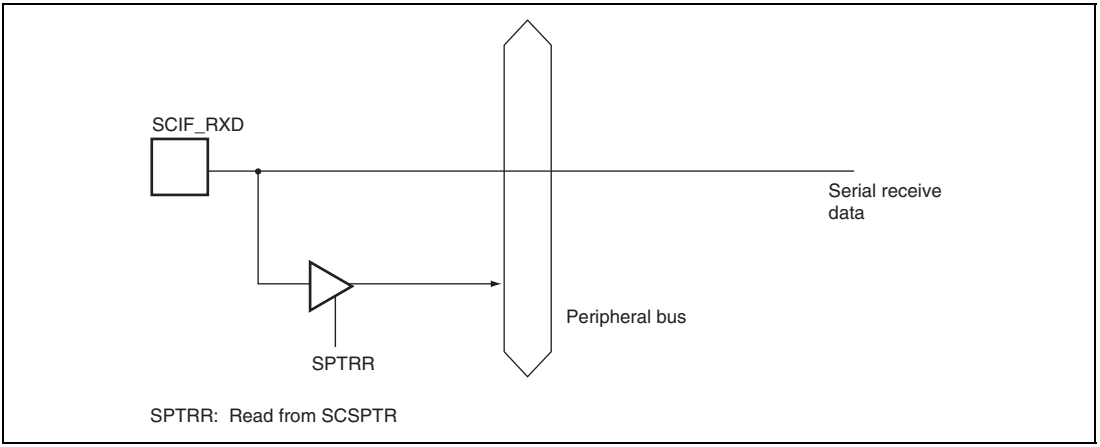


Figure 17.6 SCIF_RXD Pin

Table 17.1 shows the SCIF pin configuration. Since the pin functions are the same in each channel, the channel number is omitted in the description below. The modem control pins are available only in channels 1 and 2, and not in channel 0.

Table 17.1 Pin Configuration

Pin Name	Abbreviation	I/O	Function
Serial clock pin	SCIF0_CLK to SCIF2_CLK	Input/Output	Clock input/output
Receive data pin	SCIF0_RXD to SCIF2_RXD	Input/Output	Receive data input
Transmit data pin	SCIF0_TXD to SCIF2_TXD	Input/Output	Transmit data output
Modem control pin	$\overline{\text{SCIF1\_CTS}}$, $\overline{\text{SCIF2\_CTS}}$	Input/Output	Transmission enabled
Modem control pin	$\overline{\text{SCIF1\_RTS}}$, $\overline{\text{SCIF2\_RTS}}$	Input/Output	Transmission request

Note: These pins are made to function as serial pins by performing SCIF operation settings with the C/$\overline{\text{A}}$ bit in SCSMR, the TE, RE, CKE1, and CKE0 bits in SCSCR, and the MCE bit in SCFCR. Break state transmission and detection can be set in SCSPTR of the SCIF.

The SCIF has the following registers. Since the register functions are the same in each channel, the channel number is omitted in the description below. For details on the addresses of these registers and the state of registers in each operating mode, see section 32, List of Registers.

Table 17.2 Register Configuration (1)

Ch.	Register Name	Abbrev.	R/W	P4 Address	Area 7 Address	Size	Sync Clock
0	Serial mode register 0	SCSMR0	R/W	H'FE60 0000	H'1E60 0000	16	Pck
	Bit rate register 0	SCBRR0	R/W	H'FE60 0004	H'1E60 0004	8	Pck
	Serial control register 0	SCSCR0	R/W	H'FE60 0008	H'1E60 0008	16	Pck
	Transmit FIFO data register 0	SCFTDR0	W	H'FE60 000C	H'1E60 000C	8	Pck
	Serial status register 0	SCFSR0	R/W*1	H'FE60 0010	H'1E60 0010	16	Pck
	Receive FIFO data register 0	SCFRDR0	R	H'FE60 0014	H'1E60 0014	8	Pck
	FIFO control register 0	SCFCR0	R/W	H'FE60 0018	H'1E60 0018	16	Pck
	Transmit FIFO data count register 0	SCTFDR0	R	H'FE60 001C	H'1E60 001C	16	Pck
	Receive FIFO data count register 0	SCRFDR0	R	H'FE60 0020	H'1E60 0020	16	Pck
	Serial port register 0	SCSPTR0	R/W	H'FE60 0024	H'1E60 0024	16	Pck
	Line status register 0	SCLSR0	R/W*4	H'FE60 0028	H'1E60 0028	16	Pck
	Serial error register 0	SCRER0	R	H'FE60 002C	H'1E60 002C	16	Pck
1	Serial mode register 1	SCSMR1	R/W	H'FE61 0000	H'1E61 0000	16	Pck
	Bit rate register 1	SCBRR1	R/W	H'FE61 0004	H'1E61 0004	8	Pck
	Serial control register 1	SCSCR1	R/W	H'FE61 0008	H'1E61 0008	16	Pck
	Transmit FIFO data register 1	SCFTDR1	W	H'FE61 000C	H'1E61 000C	8	Pck
	Serial status register 1	SCFSR1	R/W*1	H'FE61 0010	H'1E61 0010	16	Pck
	Receive FIFO data register 1	SCFRDR1	R	H'FE61 0014	H'1E61 0014	8	Pck
	FIFO control register 1	SCFCR1	R/W	H'FE61 0018	H'1E61 0018	16	Pck
	Transmit FIFO data count register 1	SCTFDR1	R	H'FE61 001C	H'1E61 001C	16	Pck
	Receive FIFO data count register 1	SCRFDR1	R	H'FE61 0020	H'1E61 0020	16	Pck
	Serial port register 1	SCSPTR1	R/W	H'FE61 0024	H'1E61 0024	16	Pck
	Line status register 1	SCLSR1	R/W*4	H'FE61 0028	H'1E61 0028	16	Pck

1	Serial error register 1	SCRER1	R	H'FE61 002C	H'1E61 002C	16	Pck
2	Serial mode register 2	SCSMR2	R/W	H'FE62 0000	H'1E62 0000	16	Pck
	Bit rate register 2	SCBRR2	R/W	H'FE62 0004	H'1E62 0004	8	Pck
	Serial control register 2	SCSCR2	R/W	H'FE62 0008	H'1E62 0008	16	Pck
	Transmit FIFO data register 2	SCFTDR2	W	H'FE62 000C	H'1E62 000C	8	Pck
	Serial status register 2	SCFSR2	R/W*[1]	H'FE62 0010	H'1E62 0010	16	Pck
	Receive FIFO data register 2	SCFRDR2	R	H'FE62 0014	H'1E62 0014	8	Pck
	FIFO control register 2	SCFCR2	R/W	H'FE62 0018	H'1E62 0018	16	Pck
	Transmit FIFO data count register 2	SCTFDR2	R	H'FE62 001C	H'1E62 001C	16	Pck
	Receive FIFO data count register 2	SCRFDR2	R	H'FE62 0020	H'1E62 0020	16	Pck
	Serial port register 2	SCSPTR2	R/W	H'FE62 0024	H'1E62 0024	16	Pck
	Line status register 2	SCLSR2	R/W*[4]	H'FE62 0028	H'1E62 0028	16	Pck
	Serial error register 2	SCRER2	R	H'FE62 002C	H'1E62 002C	16	Pck

Ch.	Register Name	Abbrev.	Power-on Reset by RESET Pin/WDT/ H-UDI	Manual Reset by RESET Pin/WDT/ Multiple Exception	Sleep by Sleep Instruction/ Deep Sleep	Standby by Hardware	Standby by Software/ Each Module
0	Serial mode register 0	SCSMR0	H'0000	H'0000	Retained	*5	Retained
	Bit rate register 0	SCBRR0	H'FF	H'FF	Retained		Retained
	Serial control register 0	SCSCR0	H'0000	H'0000	Retained		Retained
	Transmit FIFO data register 0	SCFTDR0	Undefined	Undefined	Retained		Retained
	Serial status register 0	SCFSR0	H'0060	H'0060	Retained		Retained
	Receive FIFO data register 0	SCFRDR0	Undefined	Undefined	Retained		Retained
	FIFO control register 0	SCFCR0	H'0000	H'0000	Retained		Retained
	Transmit FIFO data count register 0	SCTFDR0	H'0000	H'0000	Retained		Retained
	Receive FIFO data count register 0	SCRFDR0	H'0000	H'0000	Retained		Retained
	Serial port register 0	SCSPTR0	H'0000*2	H'0000*2	Retained		Retained
	Line status register 0	SCLSR0	H'0000	H'0000	Retained		Retained
	Serial error register 0	SCRER0	H'0000	H'0000	Retained		Retained
1	Serial mode register 1	SCSMR1	H'0000	H'0000	Retained		Retained
	Bit rate register 1	SCBRR1	H'FF	H'FF	Retained		Retained
	Serial control register 1	SCSCR1	H'0000	H'0000	Retained		Retained
	Transmit FIFO data register 1	SCFTDR1	Undefined	Undefined	Retained		Retained
	Serial status register 1	SCFSR1	H'0060	H'0060	Retained		Retained
	Receive FIFO data register 1	SCFRDR1	Undefined	Undefined	Retained		Retained
	FIFO control register 1	SCFCR1	H'0000	H'0000	Retained		Retained
	Transmit FIFO data count register 1	SCTFDR1	H'0000	H'0000	Retained		Retained
	Receive FIFO data count register 1	SCRFDR1	H'0000	H'0000	Retained		Retained
	Serial port register 1	SCSPTR1	H'0000*3	H'0000*3	Retained		Retained
	Line status register 1	SCLSR1	H'0000	H'0000	Retained		Retained
	Serial error register 1	SCRER1	H'0000	H'0000	Retained		Retained

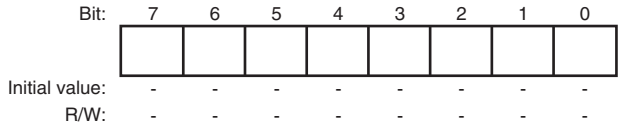
Ch.	Register Name	Abbrev.	Reset by RESET Pin/WDT/ H-UDI	by RESET Pin/WDT/ Multiple Exception	Sleep by Sleep Instruction/ Deep Sleep	by Hardware	by Software/ Each Module
2	Serial mode register 2	SCSMR2	H'0000	H'0000	Retained	*5	Retained
	Bit rate register 2	SCBRR2	H'FF	H'FF	Retained		Retained
	Serial control register 2	SCSCR2	H'0000	H'0000	Retained		Retained
	Transmit FIFO data register 2	SCFTDR2	Undefined	Undefined	Retained		Retained
	Serial status register 2	SCFSR2	H'0060	H'0060	Retained		Retained
	Receive FIFO data register 2	SCFRDR2	Undefined	Undefined	Retained		Retained
	FIFO control register 2	SCFCR2	H'0000	H'0000	Retained		Retained
	Transmit FIFO data count register 2	SCTFDR2	H'0000	H'0000	Retained		Retained
	Receive FIFO data count register 2	SCRFDR2	H'0000	H'0000	Retained		Retained
	Serial port register 2	SCSPTR2	H'0000*3	H'0000*3	Retained		Retained
	Line status register 2	SCLSR2	H'0000	H'0000	Retained		Retained
	Serial error register 2	SCRER2	H'0000	H'0000	Retained		Retained

Notes: 1. To clear the flags, 0s can only be written to bits 7 to 4, 1, and 0.
2. Bits 2 and 0 are undefined.
3. Bits 6, 4, 2, and 0 are undefined.
4. To clear the flag, 0 can only be written to bit 0.
5. After exiting hardware standby mode, this LSI enters the power-on reset state caused by the RESET pin.

SCRSR is the register used to receive serial data.

The SCIF sets serial data input from the SCIF_RXD pin in SCRSR in the order received, starting with the LSB (bit 0), and converts it to parallel data. When one byte of data has been received, it is transferred to SCFRDR, automatically.

SCRSR cannot be directly read from and written to by the CPU.



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

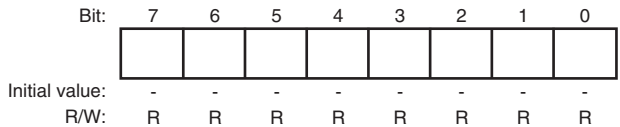
17.3.2 Receive FIFO Data Register (SCFRDR)

SCFRDR is an 8-bit FIFO register of 128 stages that stores received serial data.

When the SCIF has received one byte of serial data, it transfers the received data from SCRSR to SCFRDR where it is stored, and completes the receive operation. SCRSR is then enabled for reception, and consecutive receive operations can be performed until SCFRDR is full (128 data bytes).

SCFRDR is a read-only register, and cannot be written to by the CPU.

If a read is performed when there is no receive data in SCFRDR, an undefined value will be returned. When SCFRDR is full of receive data, subsequent serial data is lost.



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

SCTSR is the register used to transmit serial data.

To perform serial data transmission, the SCIF first transfers transmit data from SCFTDR to SCTSR, then sends the data to the SCIF_TXD pin starting with the LSB (bit 0).

When transmission of one byte is completed, the next transmit data is transferred from SCFTDR to SCTSR, and transmission started, automatically.

SCTSR cannot be directly read from and written to by the CPU.

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

17.3.4 Transmit FIFO Data Register (SCFTDR)

SCFTDR is an 8-bit FIFO register of 128 stages that stores data for serial transmission.

If SCTSR is empty when transmit data has been written to SCFTDR, the SCIF transfers the transmit data written in SCFTDR to SCTSR and starts serial transmission.

SCFTDR is a write-only register, and cannot be read by the CPU.

The next data cannot be written when SCFTDR is filled with 128 bytes of transmit data. Data written in this case is ignored.

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

SCSMR is a 16-bit register used to set the SCIF's serial transfer format and select the baud rate generator clock source.

SCSMR can always be read from and written to by the CPU.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	-	CKS1	CKS0
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	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7	C/$\overline{A}$	0	R/W	Communication Mode Selects asynchronous mode or synchronous mode as the SCIF operating mode. 0: Asynchronous mode 1: Synchronous mode
6	CHR	0	R/W	Character Length Selects 7 or 8 bits as the asynchronous mode data length. In synchronous mode, the data length is fixed at 8 bits regardless of the CHR bit setting. When 7-bit data is selected, the MSB (bit 7) of SCFTDR is not transmitted. 0: 8-bit data 1: 7-bit data

5	PE	0	R/W	Parity Enable In asynchronous mode, selects whether or not parity bit addition is performed in transmission, and parity bit checking is performed in reception. In synchronous mode, parity bit addition and checking is disabled regardless of the PE bit setting. 0: Parity bit addition and checking disabled 1: Parity bit addition and checking enabled* Note: * When the PE bit is set to 1, the parity (even or odd) specified by the O/$\overline{E}$ bit is added to transmit data before transmission. In reception, the parity bit is checked for the parity (even or odd) specified by the O/$\overline{E}$ bit.
4	O/$\overline{E}$	0	R/W	Parity Mode Selects either even or odd parity for use in parity addition and checking. In asynchronous mode, the O/$\overline{E}$ bit setting is only valid when the PE bit is set to 1, enabling parity bit addition and checking. In synchronous mode or when parity addition and checking is disabled in asynchronous mode, the O/$\overline{E}$ bit setting is invalid. 0: Even parity 1: Odd parity When even parity is set, parity bit addition is performed in transmission so that the total number of 1-bits in the transmit character plus the parity bit is even. In reception, a check is performed to see if the total number of 1-bits in the receive character plus the parity bit is even. When odd parity is set, parity bit addition is performed in transmission so that the total number of 1-bits in the transmit character plus the parity bit is odd. In reception, a check is performed to see if the total number of 1-bits in the receive character plus the parity bit is odd.

3	STOP	0	R/W	Stop Bit Length In asynchronous mode, selects 1 or 2 bits as the stop bit length. The stop bit setting is valid only in asynchronous mode. Since the stop bit is not added in synchronous mode, the STOP bit setting is invalid. 0: 1 stop bit*[1] 1: 2 stop bits*[2] In reception, 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; if it is 0, it is treated as the start bit of the next transmit character. Notes: 1. In transmission, a single 1-bit (stop bit) is added to the end of a transmit character before it is sent. 2. In transmission, two 1-bits (stop bits) are added to the end of a transmit character before it is sent.
2	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
1	CKS1	0	R/W	Clock Select 1 and 0 These bits select the clock source for the on-chip baud rate generator. The clock source can be selected from Pck, Pck/4, Pck/16, and Pck/64, according to the setting of bits CKS1 and CKS0. For details of the relationship between clock sources, bit rate register settings, and baud rate, see section17.3.8, Bit Rate Register (SCBRR). 00: Pck clock 01: Pck/4 clock 10: Pck/16 clock 11: Pck/64 clock
0	CKS0	0	R/W	

Note: Pck = Peripheral Clock

SCSCR is a register used to enable/disable transmission/reception by SCIF, serial clock output, interrupt requests, and to select transmission/reception clock source for the SCIF.

SCSCR can always be read from and written to by the CPU.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	TIE	RIE	TE	RE	REIE	-	CKE1	CKE0
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	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7	TIE	0	R/W	Transmit Interrupt Enable Enables or disables transmit-FIFO-data-empty interrupt (TXI) request generation when serial transmit data is transferred from SCFTDR to SCTSR, the number of data bytes in SCFTDR falls to or below the transmit trigger set number, and the TDFE flag in SCFSR is set to 1. TXI interrupt requests can be cleared using the following methods: Either by reading 1 from the TDFE flag, writing transmit data exceeding the transmit trigger set number to SCFTDR and then clearing the TDFE flag to 0, or by clearing the TIE bit to 0. 0: Transmit-FIFO-data-empty interrupt (TXI) request disabled 1: Transmit-FIFO-data-empty interrupt (TXI) request enabled

6	RIE	0	R/W	Receive Interrupt Enable Enables or disables generation of a receive-data-full interrupt (RXI) request when the RDF flag or DR flag in SCFSR is set to 1, a receive-error interrupt (ERI) request when the ER flag in SCFSR is set to 1, and a break interrupt (BRI) request when the BRK flag in SCFSR or the ORER flag in SCLSR is set to 1. 0: Receive-data-full interrupt (RXI) request, receive-error interrupt (ERI) request, and break interrupt (BRI) request disabled 1: Receive-data-full interrupt (RXI) request, receive-error interrupt (ERI) request, and break interrupt (BRI) request enabled Note: An RXI interrupt request can be cleared by reading 1 from the RDF or DR flag, then clearing the flag to 0, or by clearing the RIE bit to 0. ERI and BRI interrupt requests can be cleared by reading 1 from the ER, BRK, or ORER flag, then clearing the flag to 0, or by clearing the RIE and REIE bits to 0.
5	TE	0	R/W	Transmit Enable Enables or disables the start of serial transmission by the SCIF. Serial transmission is started when transmit data is written to SCFTDR while the TE bit is set to 1. 0: Transmission disabled 1: Transmission enabled* Note: SCSMR and SCFCR settings must be made, the transmission format decided, and the transmit FIFO reset, before the TE bit is set to 1.

4	RE	0	R/W	Receive Enable Enables or disables the start of serial reception by the SCIF. Serial reception is started when a start bit is detected in this state in asynchronous mode or a synchronization clock is input while the RE bit is set to 1. It should be noted that clearing the RE bit to 0 does not affect the DR, ER, BRK, RDF, FER, PER, and ORER flags, which retain their states. Serial reception begins once the start bit is detected in these states. 0: Reception disabled 1: Reception enabled* Note: * SCSMR and SCFCR settings must be made, the reception format decided, and the receive FIFO reset, before the RE bit is set to 1.
3	REIE	0	R/W	Receive Error Interrupt Enable Enables or disables generation of receive-error interrupt (ERI) and break interrupt (BRI) requests. The REIE bit setting is valid only when the RIE bit is 0. Receive-error interrupt (ERI) and break interrupt (BRI) requests can be cleared by reading 1 from the ER, BRK, or ORER flag, then clearing the flag to 0, or by clearing the RIE and REIE bits to 0. When REIE is set to 1, ERI and BRI interrupt requests will be generated even if RIE is cleared to 0. In DMAC transfer, this setting is made if the interrupt controller is to be notified of ERI and BRI interrupt requests. 0: Receive-error interrupt (ERI) and break interrupt (BRI) requests disabled 1: Receive-error interrupt (ERI) and break interrupt (BRI) requests enabled
2	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

1	CKE1	0	R/W	Clock Enable 1, 0 0 CKE0 0 R/W These bits select the SCIF clock source and whether to enable or disable the clock output from the SCIF_CLK pin. The CKE1 and CKE0 bits are used together to specify whether the SCIF_CLK pin functions as a serial clock output pin or a serial clock input pin. Note however that the CKE0 bit setting is valid only when an internal clock is selected as the SCIF clock source (CKE1 = 0). When an external clock is selected (CKE1 = 1), the CKE0 bit setting is invalid. The CKE1 and CKE0 bits must be set before determining the SCIF's operating mode with SCSMR. Asynchronous mode 00: Internal clock/SCIF_CLK pin functions as port 01: Internal clock/SCIF_CLK pin functions as clock output*1 1X: External clock/SCIF_CLK pin functions as clock input*2 Synchronous mode 0X: Internal clock/SCIF_CLK pin functions as synchronization clock output 1X: External clock/SCIF_CLK pin functions as synchronization clock input

Notes: X: Don't care
1. Outputs a clock with a frequency 16 times the bit rate.
2. Inputs a clock with a frequency 16 times the bit rate.

17.3.7 Serial Status Register (SCFSR)

SCFSR is a 16-bit register that consists of status flags that indicate the operating status of the SCIF.

SCFSR can be read from or written to by the CPU at all times. However, 1 cannot be written to flags ER, TEND, TDFE, BRK, RDF, and DR. Also note that in order to clear these flags they must be read as 1 beforehand. The FER flag and PER flag are read-only flags and cannot be modified.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	ER	TEND	TDFE	BRK	FER	PER	RDF	DR
Initial value:	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W*1	R/W*1	R/W*1	R/W*1	R	R	R/W*1	R/W*1

15 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7	ER	0	R/W*[1]	Receive Error Indicates that a framing error or parity error occurred during reception. The ER flag is not affected and retains its previous state when the RE bit in SCSCR is cleared to 0. When a receive error occurs, the receive data is still transferred to SCFRDR, and reception continues. The FER and PER bits in SCFSR can be used to determine whether there is a receive error in the readout data from SCFRDR. 0: No framing error or parity error occurred during reception [Clearing conditions] • Power-on reset or manual reset • When 0 is written to ER after reading ER = 1 1: A framing error or parity error occurred during reception [Setting conditions] • When the SCIF checks whether the stop bit at the end of the receive data is 1 when reception ends, and the stop bit is 0*[2] • When, in reception, the number of 1-bits in the receive data plus the parity bit does not match the parity setting (even or odd) specified by the O/$\overline{E}$ bit in SCSMR

6	TEND	1	R/W*	Transmit End Indicates that transmission has been ended without valid data in SCFTDR on transmission of the last bit of the transmit character. 0: Transmission is in progress [Clearing conditions] • When transmit data is written to SCFTDR, and 0 is written to TEND after reading TEND = 1 • When data is written to SCFTDR by the DMAC 1: Transmission has been ended [Setting conditions] • Power-on reset or manual reset • When the TE bit in SCSCR is 0 • When there is no transmit data in SCFTDR on transmission of the last bit of a 1-byte serial transmit character

5	TDFE	1	R/W*	Transmit FIFO Data Empty Indicates that data has been transferred from SCFTDR to SCTSR, the number of data bytes in SCFTDR has fallen to or below the transmit trigger data number set by bits TTRG1 and TTRG0 in SCFCR, and new transmit data can be written to SCFTDR. 0: A number of transmit data bytes exceeding the transmit trigger set number have been written to SCFTDR [Clearing conditions] • When transmit data exceeding the transmit trigger set number is written to SCFTDR after reading TDFE = 1, and 0 is written to TDFE • When transmit data exceeding the transmit trigger set number is written to SCFTDR by the DMAC 1: The number of transmit data bytes in SCFTDR does not exceed the transmit trigger set number [Setting conditions] • Power-on reset or manual reset • When the number of SCFTDR transmit data bytes falls to or below the transmit trigger set number as the result of a transmit operation*[3]

Notes: 1. Only 0 can be written, to clear the flag.

2. In 2-stop-bit mode, only the first stop bit is checked for a value of 1; the second stop bit is not checked.

3. As SCFTDR is a 128-byte FIFO register, the maximum number of bytes that can be written when TDFE = 1 is 128 − (transmit trigger set number). Data written in excess of this will be ignored.
The upper bits of SCFDR indicate the number of data bytes transmitted to SCFTDR.

4	BRK	0	R/W*1	Break Detect Indicates that a receive data break signal has been detected. 0: A break signal has not been received [Clearing conditions] • Power-on reset or manual reset • When 0 is written to BRK after reading BRK = 1 1: A break signal has been received*2 [Setting condition] • When data with a framing error is received, followed by the space "0" level (low level) for at least one frame length
3	FER	0	R	Framing Error In asynchronous mode, indicates whether or not a framing error has been found in the data that is to be read next from SCFRDR. 0: There is no framing error that is to be read from SCFRDR [Clearing conditions] • Power-on reset or manual reset • When there is no framing error in the data that is to be read next from SCFRDR 1: There is a framing error that is to be read from SCFRDR [Setting condition] • When there is a framing error in the data that is to be read next from SCFRDR

2	PER	0	R	Parity Error In asynchronous mode, indicates whether or not a parity error has been found in the data that is to be read next from SCFRDR. 0: There is no parity error that is to be read from SCFRDR [Clearing conditions] • Power-on reset or manual reset • When there is no parity error in the data that is to be read next from SCFRDR 1: There is a parity error in the receive data that is to be read from SCFRDR [Setting condition] • When there is a parity error in the data that is to be read next from SCFRDR
1	RDF	0	R/W*[1]	Receive FIFO Data Full Indicates that the received data has been transferred from SCRSR to SCFRDR, and the number of receive data bytes in SCFRDR is equal to or greater than the receive trigger number set by bits RTRG1 and RTRG0 in SCFCR. 0: The number of receive data bytes in SCFRDR is less than the receive trigger set number [Clearing conditions] • Power-on reset or manual reset • When SCFRDR is read until the number of receive data bytes in SCFRDR falls below the receive trigger set number after reading RDF = 1, and 0 is written to RDF • When SCFRDR is read by the DMAC until the number of receive data bytes in SCFRDR falls below the receive trigger set number 1: The number of receive data bytes in SCFRDR is equal to or greater than the receive trigger set number [Setting condition] • When SCFRDR contains at least the receive trigger set number of receive data bytes*[3]

0	DR	0	R/W[1]	Receive Data Ready In asynchronous mode, indicates that there are fewer than the receive trigger set number of data bytes in SCFRDR, and no further data has arrived for at least 15 etu after the stop bit of the last data received. This is not set when using synchronous mode. 0: Reception is in progress or has ended normally and there is no receive data left in SCFRDR [Clearing conditions] • Power-on reset or manual reset • When all the receive data in SCFRDR has been read after reading DR = 1, and 0 is written to DR • When all the receive data in SCFRDR has been read by the DMAC 1: No further receive data has arrived [Setting condition] • When SCFRDR contains fewer than the receive trigger set number of receive data bytes, and no further data has arrived for at least 15 etu after the stop bit of the last data received[4]

etu: Elementary time unit (time for transfer of 1 bit)

Notes: 1. Only 0 can be written, to clear the flag.

2. When a break is detected, the receive data (H'00) following detection is not transferred to SCFRDR. When the break ends and the receive signal returns to mark "1", receive data transfer is resumed.
3. SCFRDR is a 128-byte FIFO register. When RDF = 1, at least the receive trigger set number of data bytes can be read. If all the data in SCFRDR is read and another read is performed, the data value will be undefined. The number of receive data bytes in SCFRDR is indicated by SCRFDR.
4. Equivalent to 1.5 frames with an 8-bit, 1-stop-bit format.

SCBRR is an 8-bit register that set the serial transmission/reception bit rate in accordance with the baud rate generator operating clock selected by bits CKS1 and CKS0 in SCSMR.

SCBRR can always be read from and written to by the CPU.

The SCBRR setting is found from the following equation.

Asynchronous mode:

$$N = \frac{Pck}{64 \times 2^{2n-1} \times B} \times 10^6 - 1$$

Synchronous mode:

$$N = \frac{Pck}{8 \times 2^{2n-1} \times B} \times 10^6 - 1$$

Where B: Bit rate (bits/s)
N: SCBRR setting for baud rate generator ($0 \leq N \leq 255$)
Pck: Peripheral module operating frequency (MHz)
n: Baud rate generator input clock (n = 0 to 3)
(See table 17.3 for the relation between n and the clock.)

Table 17.3 SCSMR Settings

		SCSMR Setting	
n	**Clock**	**CKS1**	**CKS0**
0	Pck	0	0
1	Pck/4	0	1
2	Pck/16	1	0
3	Pck/64	1	1

The bit rate error in asynchronous mode is found from the following equation:

$$\text{Error (\%)} = \left\{ \frac{Pck \times 10^6}{(N + 1) \times B \times 64 \times 2^{2n-1}} - 1 \right\} \times 100$$

SCFCR performs data count resetting and trigger data number setting for transmit and receive FIFO registers, and also contains a loopback test enable bit.

SCFCR can always be read from and written to by the CPU.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	RSTRG2*1	RSTRG1*1	RSTRG0*1	RTRG1	RTRG0	TTRG1	TTRG0	MCE*1	TFRST	RFRST	LOOP
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/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 9 8	RSTRG2*1 RSTRG1*1 RSTRG0*1	0 0 0	R/W R/W R/W	$\overline{\text{SCIF\_RTS}}$ Output Active Trigger The $\overline{\text{SCIF\_RTS}}$ signal becomes high when the number of receive data stored in SCFRDR exceeds the trigger number shown below. 000:127 001:1 010:16 011:32 100:64 101:96 110:108 111:120
7 6	RTRG1 RTRG0	0 0	R/W R/W	Receive FIFO Data Number Trigger These bits are used to set the number of receive data bytes that sets the RDF flag in SCFSR. The RDF flag is set when the number of receive data bytes in SCFRDR is equal to or greater than the trigger set number shown below. 00:1 01:16 10:64 11:96

5	TTRGT	0	R/W	Transmit FIFO Data Number Trigger
4	TTRG0	0	R/W	These bits are used to set the number of remaining transmit data bytes that sets the TDFE flag in SCFSR. The TDFE flag is set when the number of transmit data bytes in SCFTDR is equal to or less than the trigger set number shown below. 00:64 (64)*2 01:32 (96) 10:4 (124) 11:0 (128)
3	MCE*1	0	R/W	Modem Control Enable Enables the $\overline{\text{SCIF\_CTS}}$ and $\overline{\text{SCIF\_RTS}}$ modem control signals. Always set the MCE bit to 0 in synchronous mode. 0: Modem signals disabled*3 1: Modem signals enabled
2	TFRST	0	R/W	Transmit FIFO Data Register Reset Invalidates the transmit data in the transmit FIFO data register and resets it to the empty state. 0: Reset operation disabled*4 1: Reset operation enabled
1	RFRST	0	R/W	Receive FIFO Data Register Reset Invalidates the receive data in the receive FIFO data register and resets it to the empty state. 0: Reset operation disabled*4 1: Reset operation enabled
0	LOOP	0	R/W	Loopback Test Internally connects the transmit output pin (SCIF_TXD) and receive input pin (SCIF_RXD), and the $\overline{\text{SCIF\_RTS}}$ pin and $\overline{\text{SCIF\_CTS}}$ pin, enabling loopback testing. 0: Loopback test disabled 1: Loopback test enabled

Notes: 1. Reserved bit in channel 0.
2. Figures in parentheses are the number of empty bytes in SCFTDR when the flag is set.
3. $\overline{\text{SCIF\_CTS}}$ is fixed at active-0 regardless of the input value, and $\overline{\text{SCIF\_RTS}}$ output is also fixed at 0.
4. A reset operation is performed in the event of a power-on reset or manual reset.

SCTFDR is a 16-bit register that indicates the number of transmit data bytes stored in SCFTDR.

SCTFDR can always be read from the CPU.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	T7	T6	T5	T4	T3	T2	T1	T0
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	Bit Name	Initial Value	R/W	Description
15 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7 to 0	T7 to T0	All 0	R	These bits show the number of untransmitted data bytes in SCFTDR. A value of H'0000 indicates that there is no transmit data, and a value of H'0080 indicates that SCFTDR is full of transmit data.

17.3.11 Receive FIFO Data Count Register (SCRFDR)

SCRFDR is a 16-bit register that indicates the number of receive data bytes stored in SCFRDR.

SCRFDR can always be read from the CPU.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	R7	R6	R5	R4	R3	R2	R1	R0
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	Bit Name	Initial Value	R/W	Description
15 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7 to 0	R7 to R0	All 0	R	These bits show the number of receive data bytes in SCFRDR. A value of H'0000 indicates that there is no receive data, and a value of H'0080 indicates that SCFRDR is full of receive data.

SCSPTR is a 16-bit readable/writable register that controls input/output and data for the port pins multiplexed with the serial communication interface (SCIF) pins at all times. Input data can be read from the SCIF_RXD pin, output data written to the SCIF_TXD pin, and breaks in serial transmission/reception controlled, by means of bits 1 and 0.

All SCSPTR bits except bits 6, 4, 2, and 0 are initialized to 0 by a power-on reset or manual reset; the value of bits 6, 4, 2, and 0 is undefined. SCSPTR is not initialized in standby mode or in the module standby state.

Note that when reading data via a serial port pin in the SCIF, the peripheral clock value from 2 cycles before is read.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	RT SIO*	RT SDT*	CT SIO*	CT SDT*	SCK IO	SCK DT	SPB2 IO	SPB2 DT
Initial value:	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

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7	RTSIO*	0	R/W	Serial Port $\overline{\text{SCIF\_RTS}}$ Port Input/Output Specifies the serial port $\overline{\text{SCIF\_RTS}}$ pin input/output condition. When actually setting the $\overline{\text{SCIF\_RTS}}$ pin as a port output pin to output the value set by the RTSDT bit, the MCE bit in SCFCR should be cleared to 0. 0: RTSDT bit value is not output to $\overline{\text{SCIF\_RTS}}$ pin 1: RTSDT bit value is output to $\overline{\text{SCIF\_RTS}}$ pin

6	RTSDT*	—	R/W	Serial Port SCIF_RTS Port Data Specifies the serial port SCIF_RTS pin input/output data. Input or output is specified by the RTSIO bit. In output mode, the RTSDT bit value is output to the SCIF_RTS pin. The SCIF_RTS pin value is read from the RTSDT bit regardless of the value of the RTSIO bit. The initial value of this bit after a power-on reset or manual reset is undefined. 0: Input/output data is low-level 1: Input/output data is high-level
5	CTSIO*	0	R/W	Serial Port SCIF_CTS Port Input/Output Specifies the serial port SCIF_CTS pin input/output condition. When actually setting the SCIF_CTS pin as a port output pin to output the value set by the CTSDT bit, the MCE bit in SCFCR should be cleared to 0. 0: CTSDT bit value is not output to SCIF_CTS pin 1: CTSDT bit value is output to SCIF_CTS pin
4	CTSDT*	—	R/W	Serial Port SCIF_CTS Port Data Specifies the serial port SCIF_CTS pin input/output data. Input or output is specified by the CTSIO bit. In output mode, the CTSDT bit value is output to the SCIF_CTS pin. The SCIF_CTS pin value is read from the CTSDT bit regardless of the value of the CTSIO bit. The initial value of this bit after a power-on reset or manual reset is undefined. 0: Input/output data is low-level 1: Input/output data is high-level

3	SCKIO	0	R/W	Serial Port Clock Port Input/Output Specifies the serial port SCIF_CLK pin input/output condition. When actually setting the SCIF_CLK pin as a port output pin to output the value set by the SCKDT bit, the CKE1 and CKE0 bits in SCSCR should be cleared to 0. 0: SCKDT bit value is not output to SCIF_CLK pin 1: SCKDT bit value is output to SCIF_CLK pin
2	SCKDT	—	R/W	Serial Port Clock Port Data Specifies the serial port SCIF_CLK pin input/output data. Input or output is specified by the SCKIO bit. In output mode, the SCKDT bit value is output to the SCIF_CLK pin. The SCIF_CLK pin value is read from the SCKDT bit regardless of the value of the SCKIO bit. The initial value of this bit after a power-on reset or manual reset is undefined. 0: Input/output data is low-level 1: Input/output data is high-level
1	SPB2IO	0	R/W	Serial Port Break Input/Output Specifies the serial port SCIF_TXD pin output condition. When actually setting the SCIF_TXD pin as a port output pin to output the value set by the SPB2DT bit, the TE bit in SCSCR should be cleared to 0. 0: SPB2DT bit value is not output to the SCIF_TXD pin 1: SPB2DT bit value is output to the SCIF_TXD pin
0	SPB2DT	—	R/W	Serial Port Break Data Specifies the serial port SCIF_RXD pin input data and SCIF_TXD pin output data. The SCIF_TXD pin output condition is specified by the SPB2IO bit. When the SCIF_TXD pin is designated as an output, the value of the SPB2DT bit is output to the SCIF_TXD pin. The SCIF_RXD pin value is read from the SPB2DT bit regardless of the value of the SPB2IO bit. The initial value of this bit after a power-on reset or manual reset is undefined. 0: Input/output data is low-level 1: Input/output data is high-level

Note: * Only channels 1 and 2. Reserved bit in channel 0.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	ORER
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*1

Bit	Bit Name	Initial Value	R/W	Description
15 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	ORER	0	R/W*1	Overrun Error Indicates that an overrun error occurred during reception, causing abnormal termination. 0: Reception in progress, or reception has ended normally*2 [Clearing conditions] • Power-on reset or manual reset • When 0 is written to ORER after reading ORER = 1 1: An overrun error occurred during reception*3 [Setting condition] • When the next serial reception is completed while SCFRDR receives 128-byte data (SCFRDR is full)

Notes: 1. Only 0 can be written, to clear the flag.
2. The ORER flag is not affected and retains its previous state when the RE bit in SCSCR is cleared to 0.
3. The receive data prior to the overrun error is retained in SCFRDR, and the data received subsequently is lost. Serial reception cannot be continued while the ORER flag is set to 1.

SCRER is a 16-bit register that indicates the number of receive errors in the data in SCFRDR. SCRER can always be read from the CPU.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	PER6	PER5	PER4	PER3	PER2	PER1	PER0	-	FER6	FER5	FER4	FER3	FER2	FER1	FER0
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	Bit Name	Initial Value	R/W	Description
15	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
14	PER6	0	R	Number of Parity Errors These bits indicate the number of data bytes in which a parity error occurred in the receive data stored in SCFRDR. After the ER bit in SCFSR is set, the value indicated by bits PER6 to PER0 is the number of data bytes in which a parity error occurred. If all 128 bytes of receive data in SCFRDR have parity errors, the value indicated by bits PER6 to PER0 will be 0.
13	PER5	0	R	
12	PER4	0	R	
11	PER3	0	R	
10	PER2	0	R	
9	PER1	0	R	
8	PER0	0	R	
7	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
6	FER6	0	R	Number of Framing Errors These bits indicate the number of data bytes in which a framing error occurred in the receive data stored in SCFRDR. After the ER bit in SCFSR is set, the value indicated by bits FER6 to FER0 is the number of data bytes in which a framing error occurred. If all 128 bytes of receive data in SCFRDR have framing errors, the value indicated by bits FER6 to FER0 will be 0.
5	FER5	0	R	
4	FER4	0	R	
3	FER3	0	R	
2	FER2	0	R	
1	FER1	0	R	
0	FER0	0	R	

17.4.1 Overview

The SCIF can carry out serial communication in asynchronous mode, in which synchronization is achieved character by character and in synchronous mode, in which synchronization is achieved with clock pulses. For details on asynchronous mode, see section 17.4.2, Operation in Asynchronous Mode.

128-stage FIFO buffers are provided for both transmission and reception, reducing the CPU overhead, and enabling fast and continuous communication to be performed.

$\overline{\text{SCIF\_RTS}}$ and $\overline{\text{SCIF\_CTS}}$ signals are also provided as modem control signals.

The serial transfer format is selected using SCSMR, as shown in table 17.4. The SCIF clock source is determined by the combination of the C/$\overline{\text{A}}$ bit in SCSMR and the CKE1 and CKE0 bits in SCSCR, as shown in table 17.5.

Asynchronous Mode:

- Data length: Choice of 7 or 8 bits
- Choice of parity addition and addition of 1 or 2 stop bits (the combination of these parameters determines the transfer format and character length)
- Detection of framing errors, parity errors, receive-FIFO-data-full state, overrun errors, receive-data-ready state, and breaks, during reception
- Indication of the number of data bytes stored in the transmit and receive FIFO registers
- Choice of internal or external clock as SCIF clock source

  When internal clock is selected: The SCIF operates on the baud rate generator clock and can output a clock with frequency of 16 times the bit rate.

  When external clock is selected: A clock with a frequency of 16 times the bit rate must be input (the on-chip baud rate generator is not used).

Synchronous Mode:

- Data length: Fixed at 8 bits
- Detection of overrun errors during reception
- Choice of internal or external clock as SCIF clock source

  When internal clock is selected: The SCIF operates on the baud rate generator clock and a serial clock is output to external devices.

  When external clock is selected: The on-chip baud rate generator is not used and the SCIF operates on the input serial clock.

SCSMR Settings					SCIF Transfer Format		
Bit 7: C/$\overline{A}$	**Bit 6: CHR**	**Bit 5: PE**	**Bit 3: STOP**	**Mode**	**Data Length**	**Parity Bit**	**Stop Bit Length**
0	0	0	0	Asynchronous mode	8-bit data	No	1 bit
			1				2 bits
		1	0			Yes	1 bit
			1				2 bits
	1	0	0		7-bit data	No	1 bit
			1				2 bits
		1	0			Yes	1 bit
			1				2 bits
1	*	*	*	Synchronous mode	8-bit data	No	No

Note: * Don't care

Table 17.5 SCSMR and SCSCR Settings for SCIF Clock Source Selection

SCSMR	SCSCR Settings				
Bit 7: C/$\overline{A}$	**Bit 1: CKE1**	**Bit 0: CKE0**	**Mode**	**Clock Source**	**SCIF_CLK Pin Function**
0	0	0	Asynchronous mode	Internal	SCIF does not use SCIF_CLK pin
		1			Outputs clock with frequency of 16 times the bit rate
	1	0		External	Inputs clock with frequency of 16
		1			times the bit rate
1	0	0	Synchronous mode	Internal	Outputs synchronization clock
		1			
	1	0		External	Inputs synchronization clock
		1			

In asynchronous mode, a character that consists of data with a start bit indicating the start of communication and a stop bit indicating the end of communication is transmitted or received. In this mode, serial communication is performed with synchronization achieved character by character.

Inside the SCIF, the transmitter and receiver are independent units, enabling full-duplex communication. Both the transmitter and receiver have a 128-stage FIFO buffer structure, so that data can be read or written during transmission or reception, enabling continuous data transmission and reception.

Figure 17.7 shows the general format for asynchronous serial communication.

In asynchronous serial communication, the transmission line is usually held in the mark state (high level). The SCIF monitors the transmission line, and when it goes to the space state (low level), recognizes a start bit and starts serial communication.

One character in serial communication consists of a start bit (low level), followed by transmit/receive data (LSB-first; from the lowest bit), a parity bit (high or low level), and finally stop bits (high level).

In reception in asynchronous mode, the SCIF synchronizes with the fall of the start bit. Receive data can be latched at the middle of each bit because the SCIF samples data at the eighth clock which has a frequency of 16 times the bit rate.

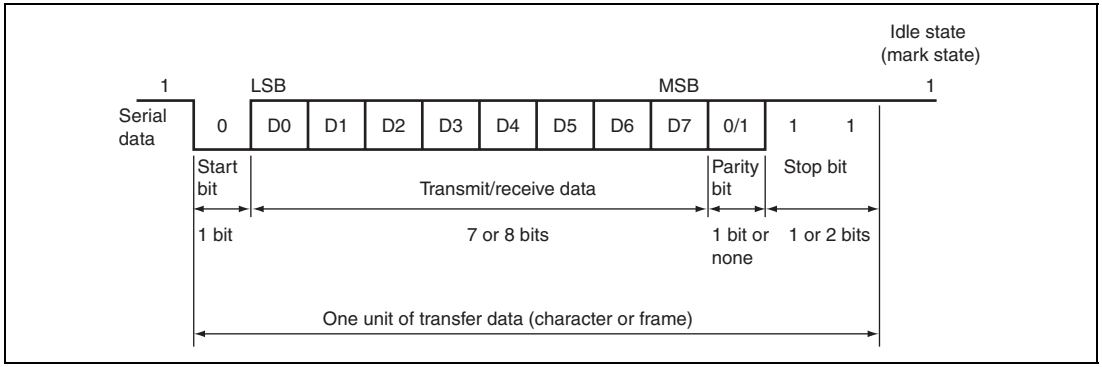


Figure 17.7 Data Format in Asynchronous Communication (Example with 8-Bit Data, Parity, and Two Stop Bits)

Table 17.6 shows the data transfer formats that can be used. Any of 8 transfer formats can be selected according to the SCSMR settings.

Table 17.6 Serial Transfer Formats (Asynchronous Mode)

SCSMR Settings			Serial Transfer Format and Frame Length											
CHR	PE	STOP	1	2	3	4	5	6	7	8	9	10	11	12
0	0	0	S	8-bit data								STOP		
0	0	1	S	8-bit data								STOP	STOP	
0	1	0	S	8-bit data								P	STOP	
0	1	1	S	8-bit data								P	STOP	STOP
1	0	0	S	7-bit data							STOP			
1	0	1	S	7-bit data							STOP	STOP		
1	1	0	S	7-bit data							P	STOP		
1	1	1	S	7-bit data							P	STOP	STOP	

Legend:
S : Start bit
STOP : Stop bit
P : Parity bit

Either an internal clock generated by the on-chip baud rate generator or an external clock input at the SCIF_CLK pin can be selected as the SCIF's serial clock, according to the settings of the C/$\overline{A}$ bit in SCSMR and the CKE1 and CKE0 bits in SCSCR. For details of SCIF clock source selection, see table 17.5.

When an external clock is input at the SCIF_CLK pin, the clock frequency should be 16 times the bit rate used.

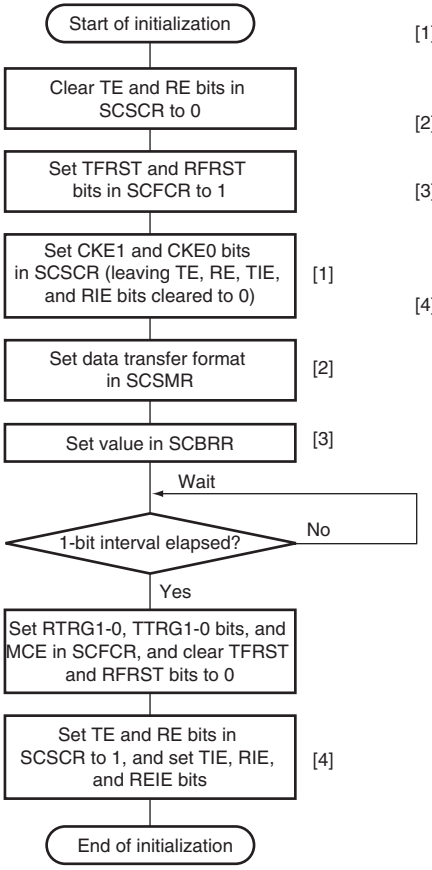
When the SCIF is operated on an internal clock, a clock whose frequency is 16 times the bit rate is output from the SCIF_CLK pin.

(3) SCIF Initialization (Asynchronous Mode)

Before transmitting and receiving data, it is necessary to clear the TE and RE bits in SCSCR to 0, then initialize the SCIF as described below.

When the operating mode or transfer format, etc., is changed, the TE and RE bits must be cleared to 0 before making the change using the following procedure.

1. When the TE bit is cleared to 0, SCTSR is initialized. Note that clearing the TE and RE bits to 0 does not change the contents of SCFSR, SCFTDR, or SCFRDR.
2. The TE bit should be cleared to 0 after all transmit data has been sent and the TEND flag in SCFSR has been set. TEND can also be cleared to 0 during transmission, but the data being transmitted will go to the mark state after the clearance. Before setting TE again to start transmission, the TFRST bit in SCFCR should first be set to 1 to reset SCFTDR.
3. When an external clock is used the clock should not be stopped during operation, including initialization, since operation will be unreliable in this case.



[1] Set the clock selection in SCSCR. Be sure to clear bits TIE, RIE, TE, and RE to 0.

[2] Set the data transfer format in SCSMR.

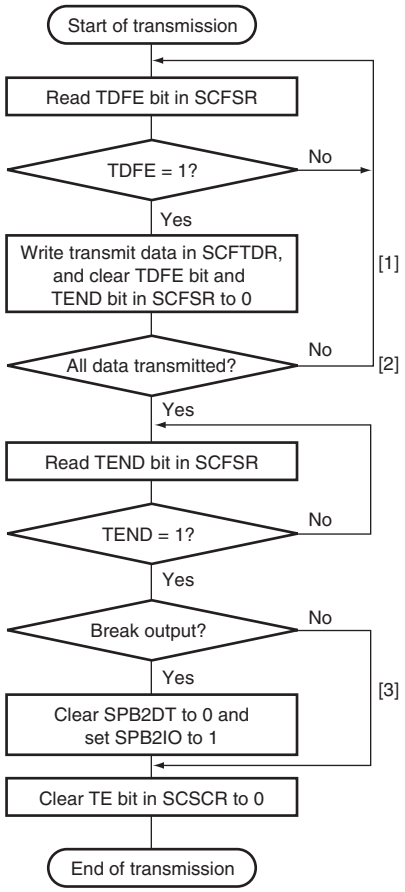
[3] Write a value corresponding to the bit rate into SCBRR. (Not necessary if an external clock is used.)

[4] Wait at least one bit interval, then set the TE bit or RE bit in SCSCR to 1. Also set the RIE, REIE, and TIE bits.
Setting the TE and RE bits enables the SCIF_TXD and SCIF_RXD pins to be used. When transmitting, the SCIF will go to the mark state; when receiving, it will go to the idle state, waiting for a start bit.

Figure 17.8 Sample SCIF Initialization Flowchart

Figure 17.9 shows a sample flowchart for serial transmission.

Use the following procedure for serial data transmission after enabling the SCIF for transmission.



[1] SCIF status check and transmit data write:

Read SCFSR and check that the TDFE flag is set to 1, then write transmit data to SCFTDR, and clear the TDFE and TEND flags to 0.

The number of transmit data bytes that can be written is 128 - (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 SCFTDR, and then clear the TDFE flag to 0.

[3] Break output at the end of serial transmission:

To output a break in serial transmission, clear the SPB2DT bit to 0 and set the SPB2IO bit to 1 in SCSPTR, then clear the TE bit in SCSCR to 0.

In [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 SCTFDR.

Figure 17.9 Sample Serial Transmission Flowchart

In serial transmission, the SCIF operates as described below.

1. When data is written into SCFTDR, the SCIF transfers the data from SCFTDR to SCTSR and starts transmitting. Confirm that the TDFE flag in SCFSR is set to 1 before writing transmit data to SCFTDR. The number of data bytes that can be written is at least 128 – (transmit trigger setting).

number of transmit data bytes in SCFTDR falls to or below the transmit trigger number set in SCFCR, the TDFE flag is set. If the TIE bit in SCSCR 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 SCIF_TXD pin in the following order.

A. Start bit: One 0-bit 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 1-bits (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 SCFTDR transmit data at the timing for sending the stop bit. If data is present, the data is transferred from SCFTDR to SCTSR, the stop bit is sent, and then serial transmission of the next frame is started.

   If there is no transmit data after the stop bit is sent, the TEND flag in SCFSR is set to 1, the stop bit is sent, and then the line goes to the mark state in which 1 is output from the SCIF_TXD pin.

Figure 17.10 shows an example of the operation for transmission in asynchronous mode.

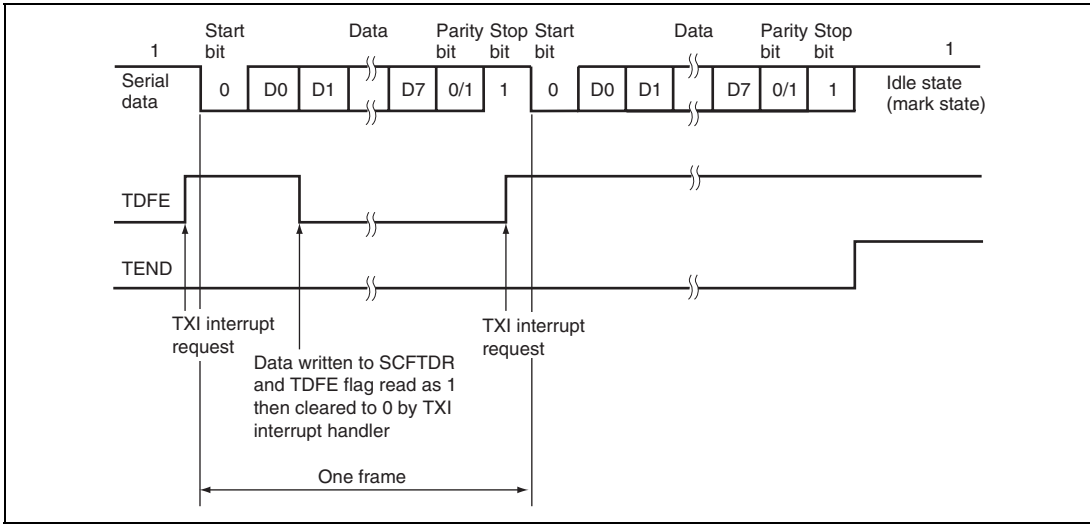


Figure 17.10 Sample SCIF Transmission Operation (Example with 8-Bit Data, Parity, One Stop Bit)

the SCIF_CTS input value. When SCIF_CTS is set to 1 during transmission, the line goes to the mark state after transmission of one frame. When SCIF_CTS is set to 0, the next transmit data is output starting from the start bit.

Figure 17.11 shows an example of the operation when modem control is used.

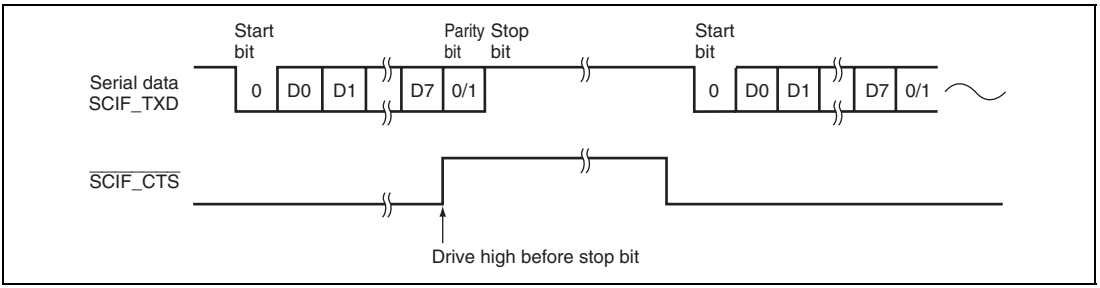
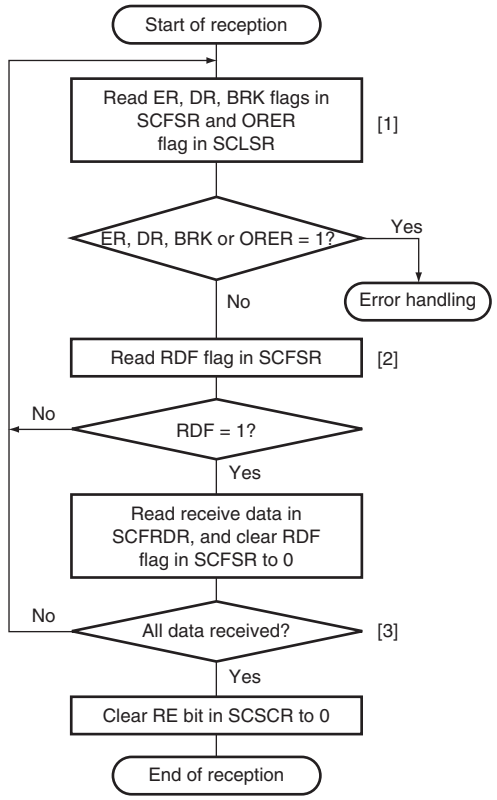


Figure 17.11 Sample Operation Using Modem Control (SCIF_CTS) (Only in Channels 1 and 2)

Figure 17.12 shows a sample flowchart for serial reception.

Use the following procedure for serial data reception after enabling the SCIF for reception.



[1] Receive error handling and break detection:

Read the DR, ER, and BRK flags in SCFSR, and the ORER flag in SCLSR, to identify any error, perform the appropriate error handling, then clear the DR, ER, BRK, and ORER flags to 0. In the case of a framing error, a break can also be detected by reading the value of the SCIF_RXD pin.

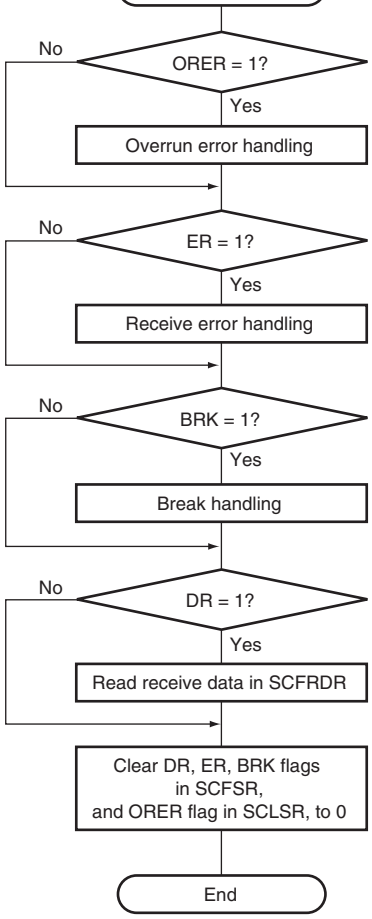
[2] SCIF status check and receive data read:

Read SCFSR and check that RDF = 1, then read the receive data in SCFRDR, 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 also 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 SCFRDR, read 1 from the RDF flag, then clear the RDF flag to 0. The number of receive data bytes in SCFRDR can be ascertained by reading from SCRFDR.

Figure 17.12 Sample Serial Reception Flowchart (1)



has occurred in the receive data that is to be read from SCFRDR can be ascertained from the FER and PER bits in SCFSR.

[2] When a break signal is received, receive data (H'00) is not transferred to SCFRDR.
However, note that the last data in SCFRDR is H'00, and the break data in which a framing error occurred is stored.
When a break handling is completed and a receive signal returns to 1, the receive data transfer resumes.

Figure 17.12 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 SCRSR 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 SCRSR to SCFRDR.*

D. Break check: The SCIF checks that the BRK flag is 0, indicating that the break state is not set.*

If (b), (c), and (d) checks are passed, the receive data is stored in SCFRDR.

Note: * Reception continues even when a parity error or framing error occurs.

4. If the RIE bit in SCSCR 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 or REIE bit in SCSCR is set to 1 when the ER flag changes to 1, a receive-error interrupt (ERI) request is generated.

   If the RIE bit or REIE bit in SCSCR is set to 1 when the BRK or ORER flag changes to 1, a break reception interrupt (BRI) request is generated.

Figure 17.13 shows an example of the operation for reception in asynchronous mode.

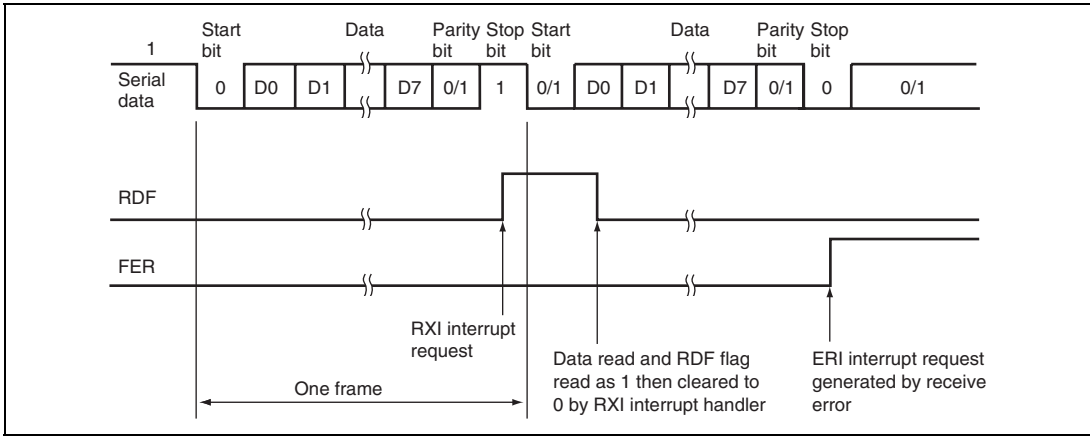


Figure 17.13 Sample SCIF Receive Operation (Example with 8-Bit Data, Parity, One Stop Bit)

5. When modem control is enabled, the $\overline{\text{SCIF\_RTS}}$ signal is output when SCFRDR is empty. When $\overline{\text{SCIF\_RTS}}$ is 0, reception is possible. When $\overline{\text{SCIF\_RTS}}$ is 1, this indicates that SCFRDR contains bytes of data equal to or more than the $\overline{\text{SCIF\_RTS}}$ output active trigger number.The $\overline{\text{SCIF\_RTS}}$ output active trigger value is specified by bits 10 to 8 in the FIFO control register (SCFCR). For details, see section 17.3.9, FIFO control register (SCFCR). In addition, $\overline{\text{SCIF\_RTS}}$ is also 1 when the RE bit in SCSCR is cleared to 0.
   Figure 17.14 shows an example of the operation when modem control is used.

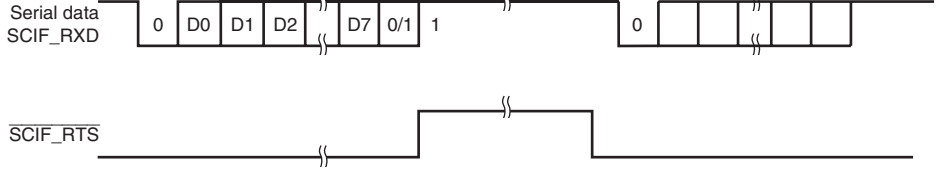


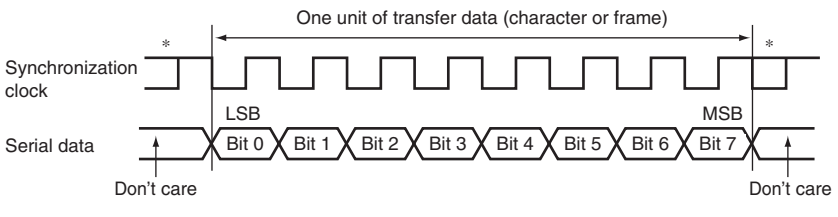
Figure 17.14 Sample Operation Using Modem Control (SCIF_RTS) (Only in Channels 1 and 2)

17.4.3 Operation in Synchronous Mode

Synchronous mode, in which data is transmitted or received in synchronization with clock pulses, is suitable for fast serial communication.

Since the transmitter and receiver are independent units in the SCIF, full-duplex communication can be achieved by sharing the clock. Both the transmitter and receiver have a 128-stage FIFO buffer structure, so that data can be read or written during transmission or reception, enabling continuous data transfer and reception.

Figure 17.15 shows the general format for synchronous communication.



Note: * High except in continuous transfer

Figure 17.15 Data Format in Synchronous Communication

In synchronous serial communication, data on the communication line is output from one fall of the synchronization clock to the next fall. Data is guaranteed to be accurate at the start of the synchronization clock.

In serial communication, each character is output starting with the LSB and ending with the MSB. After the MSB is output, the communication line remains in the state of the last data.

In synchronous mode, the SCIF receives data in synchronization with the rise of the synchronization clock.

A fixed 8-bit data format is used. No parity bit can be added.

(2) Clock

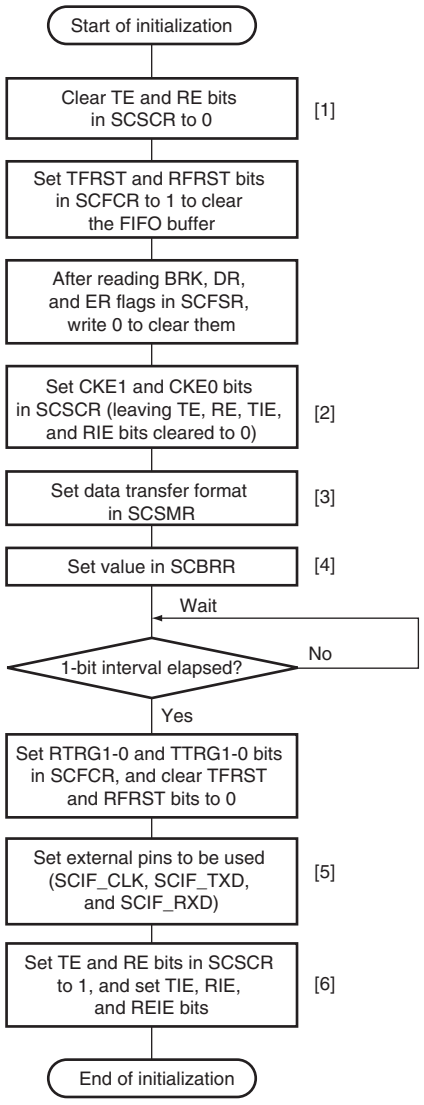
Either an internal clock generated by the on-chip baud rate generator or an external synchronization clock input at the SCIF_CLK pin can be selected as the SCIF's serial clock, according to the settings of the C/$\overline{A}$ bit in SCSMR and the CKE1 and CKE0 bits in SCSCR. For details of SCIF clock source selection, see table 17.5.

When the SCIF is operated on an internal clock, the synchronization clock is output from the SCIF_CLK pin. Eight synchronization clock pulses are output in the transfer of one character, and when no transfer is performed the clock is fixed high. When an internal clock is selected in a receive operation only, as long as the RE bit in SCSCR is set to 1, clock pulses are output until the number of receive data bytes in the receive FIFO data register reaches the receive trigger number.

(3) SCIF Initialization (Synchronous Mode)

Before transmitting and receiving data, it is necessary to clear the TE and RE bits in SCSCR to 0, then initialize the SCIF as described below.

When changing the operating mode or transfer format, etc., 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, SCTSR is initialized. Note that clearing the RE bit to 0 does not initialize the RDF, PER, FER, or ORER flag state or change the contents of SCFRDR.



[1] Leave the TE and RE bits cleared to 0 until the initialization almost ends. Be sure to clear the TIE, RIE, TE, and RE bits to 0.

[2] Set the CKE1 and CKE0 bits.

[3] Set the data transfer format in SCSMR.

[4] Write a value corresponding to the bit rate into SCBRR. This is not necessary if an external clock is used. Wait at least one bit interval after this write before moving to the next step.

[5] Set the external pins to be used. Set SCIF_RXD input for reception and SCIF_TXD output for transmission. The input/output of the SCIF_CLK pin must match the setting of the CKE1 and CKE0 bits.

[6] Set the TE or RE bit in SCSCR to 1. Also set the TIE, RIE, and REIE bits to enable the SCIF_TXD, SCIF_RXD, and SCIF_CLK pins to be used. When transmitting, the SCIF_TXD pin will go to the mark state. When receiving in clocked synchronous mode with the synchronization clock output (clock master) selected, a clock starts to be output from the SCIF_CLK pin at this point.

Figure 17.16 Sample SCIF Initialization Flowchart

Figure 17.17 shows a sample flowchart for serial transmission.

Use the following procedure for serial data transmission after enabling the SCIF for transmission.

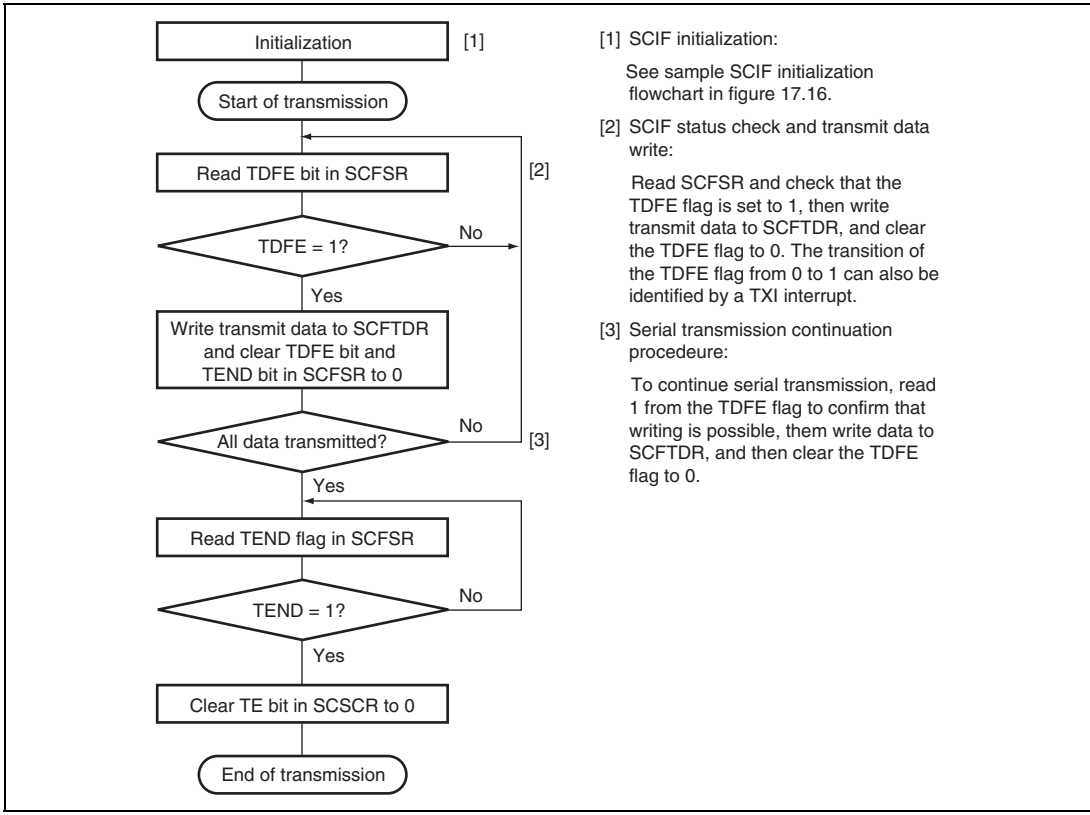


Figure 17.17 Sample Serial Transmission Flowchart

In serial transmission, the SCIF operates as described below.

1. When data is written into SCFTDR, the SCIF transfers the data from SCFTDR to SCTSR and starts transmitting. Confirm that the TDFE flag in SCFSR is set to 1 before writing transmit data to SCFTDR. The number of data bytes that can be written is at least 128 (transmit trigger setting).
2. When data is transferred from SCFTDR to SCTSR and transmission is started, consecutive transmit operations are performed until there is no transmit data left in SCFTDR. When the number of transmit data bytes in SCFTDR falls to or below the transmit trigger number set in

data-empty interrupt (TXI) request is generated.

If clock output mode is selected, the SCIF outputs eight synchronization clock pulses for each data.

When the external clock is selected, data is output in synchronization with the input clock.

The serial transmit data is sent from the SCIF_TXD pin in the LSB-first order.

3. The SCIF checks the SCFTDR transmit data at the timing for sending the last bit. If data is present, the data is transferred from SCFTDR to SCTSR, and then serial transmission of the next frame is started. If there is no transmit data, the TEND flag in SCFSR is set to 1 after the last bit is sent, and the transmit data pin (SCIF_TXD pin) retains the output state of the last bit.
4. After serial transmission ends, the SCIF_CLK pin is fixed high.

Figure 17.18 shows an example of the operation for transmission in synchronous mode.

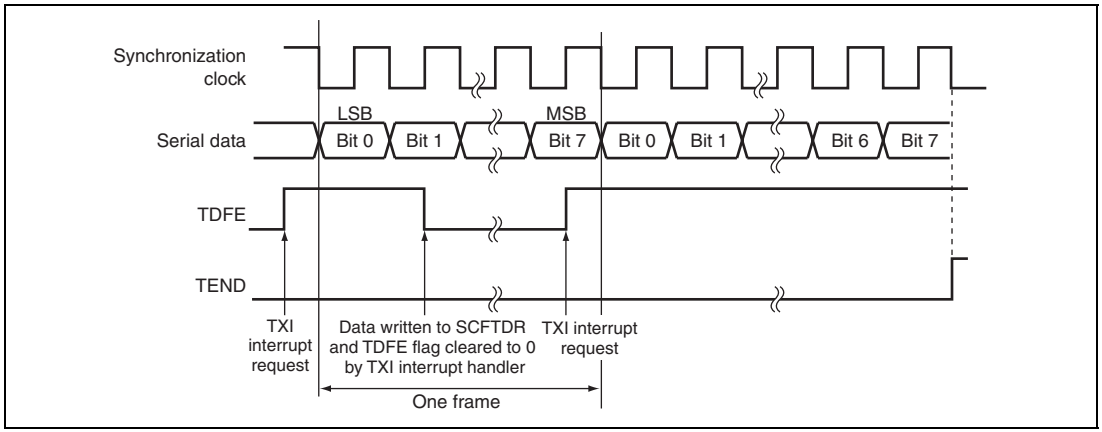


Figure 17.18 Sample SCIF Transmission Operation in Synchronous Mode

Figure 17.19 shows a sample flowchart for serial reception.

Use the following procedure for serial data reception after enabling the SCIF for reception.

When switching the operating mode from asynchronous mode to synchronous mode without initializing the SCIF, make sure that the ORER, PER7 to PER0, and FER7 to FER0 flags are cleared to 0.

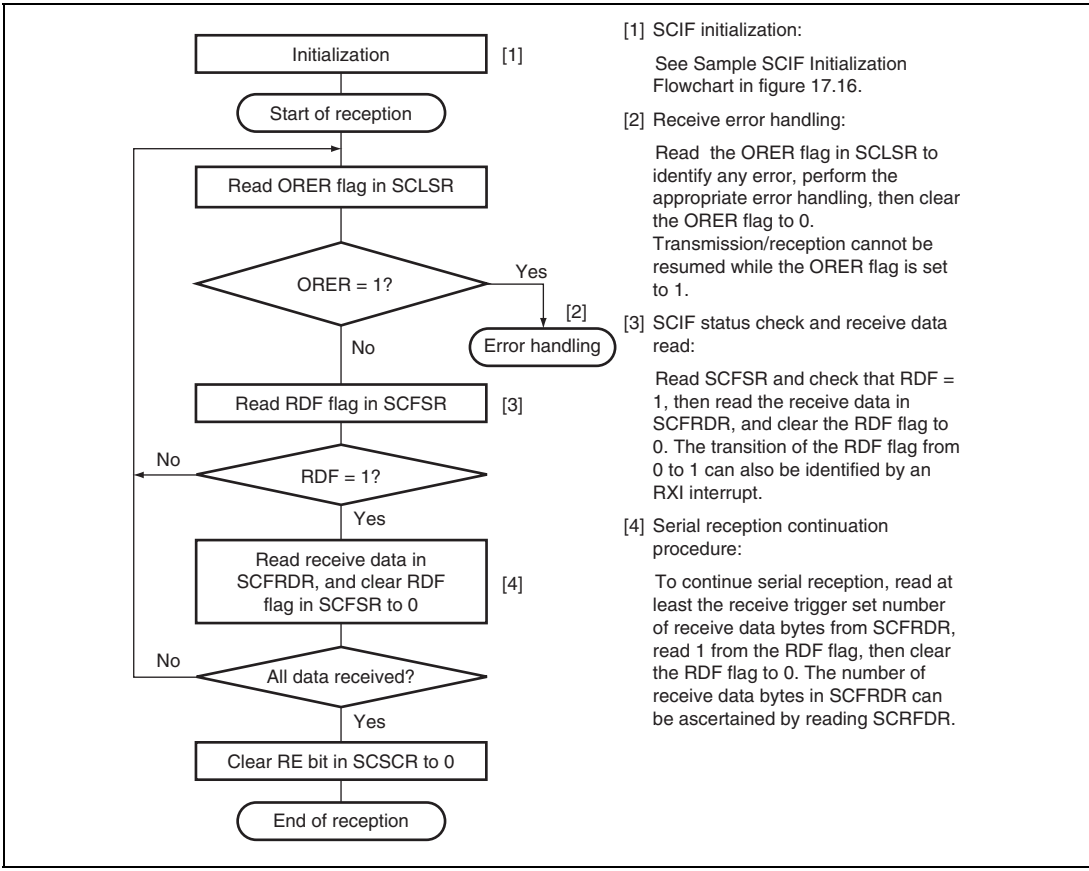


Figure 17.19 Sample Serial Reception Flowchart (1)

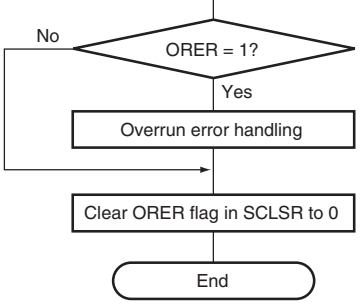


Figure 17.19 Sample Serial Reception Flowchart (2)

In serial reception, the SCIF operates as described below.

1. The SCIF is initialized internally in synchronization with the input or output of the synchronization clock.
2. The received data is stored in SCRSR in LSB-to-MSB order.

   After receiving the data, the SCIF checks whether the receive data can be transferred from SCRSR to SCFRDR. If this check is passed, the receive data is stored in SCFRDR. If an overrun error is detected in the error check, reception cannot continue.
3. If the RIE bit in SCSCR is set to 1 when the RDF flag changes to 1, a receive-FIFO-data-full interrupt (RXI) request is generated.

   If the RIE bit in SCSCR is set to 1 when the ORER flag changes to 1, a break interrupt (BRI) request is generated.

Figure 17.20 shows an example of the operation for reception in synchronous mode.

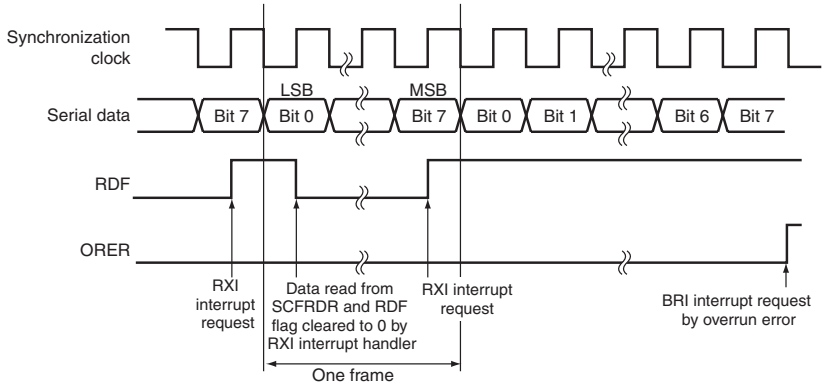
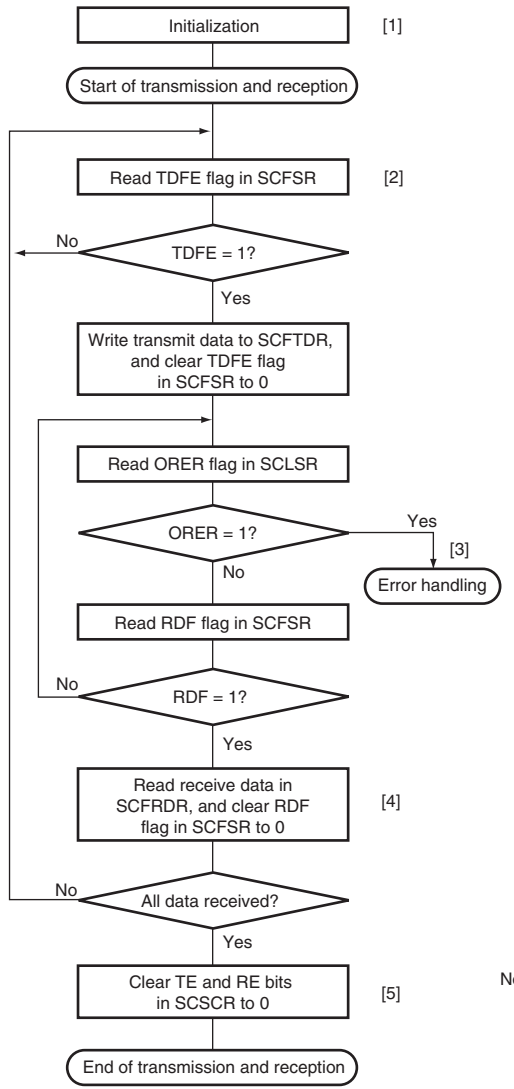


Figure 17.20 Sample SCIF Reception Operation in Synchronous Mode

Figure 17.21 shows a sample flowchart for simultaneous serial data transmission and reception.

Use the following procedure for simultaneous serial transmission and reception after enabling the SCIF for both transmission and reception.



[1] SCIF initialization:

See Sample SCIF Initialization Flowchart in figure 17.16.

[2] SCIF status check and transmit data write:

Read SCFSR and check that the TDFE flag is set to 1, then write transmit data to SCFTDR, and clear the TDFE flag to 0. The transition of the TDFE flag from 0 to 1 can also be identified by a TXI interrupt.

[3] Receive error handling:

Read the ORER flag in SCLSR to identify any error, perform the appropriate error handling, then clear the ORER flag to 0. Reception cannot be resumed while the ORER flag is set to 1.

[4] SCIF status check and receive data read:

Read SCFSR and check that RDF = 1, then read the receive data in SCFRDR, and clear the RDF flag to 0. The transition of the RDF flag from 0 to 1 can also be identified by an RXI interrupt.

[5] Serial transmission and reception continuation procedure:

To continue serial transmission and reception, read RDF flag and SCFRDR, and clear the RDF flag to 0 before receiving the MSB in the current frame. Similarly, read 1 from the TDFE flag to confirm that writing is possible before transmitting the MSB in the current frame. Then write data to SCFTDR and clear the TDFE flag to 0.

Note: When switching from a transmit operation or receive operation to simultaneous transmission and reception operations, clear the TE and RE bits to 0, and then set them simultaneously to 1.

Figure 17.21 Sample Simultaneous Serial Transmission and Reception Flowchart

The SCIF has four interrupt sources: transmit-FIFO-data-empty interrupt (TXI) request, receive-error interrupt (ERI) request, receive-FIFO-data-full interrupt (RXI) request, and break interrupt (BRI) request.

Table 17.7 shows the interrupt sources and their order of priority. The interrupt sources are enabled or disabled by means of the TIE, RIE, and REIE bits in SCSCR. A separate interrupt request is sent to the interrupt controller for each of these interrupt sources.

If the TDFE flag in SCFSR is set to 1 when a TXI interrupt is enabled by the TIE bit, a TXI interrupt request and a transmit-FIFO-data-empty request for DMA transfer are generated. If the TDFE flag is set to 1 when a TXI interrupt is disabled by the TIE bit, only a transmit-FIFO-data-empty request for DMA transfer is generated. A transmit-FIFO-data-empty request can activate the DMAC to perform data transfer.

If the RDF or DR flag in SCFSR is set to 1 when an RXI interrupt is enabled by the RIE bit, an RXI interrupt request and a receive-FIFO-data-full request for DMA transfer are generated. If the RDF or DR flag is set to 1 when an RXI interrupt is disabled by the RIE bit, only a receive-FIFO-data-full request for DMA transfer is generated. A receive-FIFO-data-full request can activate the DMAC to perform data transfer. Note that generation of an RXI interrupt request or a receive-FIFO-data-full request by setting the DR flag to 1 occurs only in asynchronous mode.

When the BRK flag in SCFSR or the ORER flag in SCLSR is set to 1, a BRI interrupt request is generated.

If transmission/reception is carried out using the DMAC, set and enable the DMAC before making the SCIF settings. Also make settings to inhibit output of RXI and TXI interrupt requests to the interrupt controller. If output of interrupt requests is enabled, these interrupt requests to the interrupt controller can be cleared by the DMAC regardless of the interrupt handler.

By setting the REIE bit to 1 while the RIE bit is cleared to 0 in SCSCR, it is possible to output ERI interrupt requests, but not RXI interrupt requests.

Interrupt Source	Description	DMAC Activation	Priority on Reset Release
ERI	Interrupt initiated by receive error flag (ER)	Not possible	High
RXI	Interrupt initiated by receive FIFO data full flag (RDF) or receive data ready flag (DR)*	Possible	
BRI	Interrupt initiated by break flag (BRK) or overrun error flag (ORER)	Not possible	
TXI	Interrupt initiated by transmit FIFO data empty flag (TDFE)	Possible	Low

Note: * An RXI interrupt by setting of the DR flag is available only in asynchronous mode.

17.6 Usage Notes

Note the following when using the SCIF.

(1) SCFTDR Writing and the TDFE Flag

The TDFE flag in SCFSR is set when the number of transmit data bytes written in SCFTDR has fallen to or below the transmit trigger number set by bits TTRG1 and TTRG0 in SCFCR. After TDFE is set, transmit data up to the number of empty bytes in SCFTDR can be written, allowing efficient continuous transmission.

However, if the number of data bytes written in SCFTDR is equal to or less than the transmit trigger number, the TDFE flag will be set to 1 again, even after being read as 1 and cleared to 0. TDFE clearing should therefore be carried out when SCFTDR contains more than the transmit trigger number of transmit data bytes.

The number of transmit data bytes in SCFTDR can be found from SCTFDR.

(2) SCFRDR Reading and the RDF Flag

The RDF flag in SCFSR is set when the number of receive data bytes in SCFRDR has become equal to or greater than the receive trigger number set by bits RTRG1 and RTRG0 in SCFCR. After RDF is set, receive data equivalent to the trigger number can be read from SCFRDR, allowing efficient continuous reception.

However, if the number of data bytes read in SCFRDR is equal to or greater than the trigger number, the RDF flag will be set to 1 again even if it is cleared to 0. After the receive data is read, clear the RDF flag readout to 0 in order to reduce the number of data bytes in SCFRDR to less than the trigger number.

(3) Break Detection and Processing

If a framing error (FER) is detected**,** break signals can also be detected by reading the SCIF_RXD pin value directly. In the break state the input from the SCIF_RXD pin consists of all 0s, so the FER flag is set and the parity error flag (PER) may also be set.

Although the SCIF stops transferring receive data to SCFRDR after receiving a break, the receive operation continues.

(4) Sending a Break Signal

The input/output condition and level of the SCIF_TXD pin are determined by bits SPB2IO and SPB2DT in SCSPTR. This feature can be used to send a break signal.

After the serial transmitter is initialized and until the TE bit is set to 1 (enabling transmission), the SCIF_TXD pin function is not selected and the value of the SPB2DT bit substitutes for the mark state. The SPB2IO and SPB2DT bits should therefore be set to 1 (designating output and high level) in the beginning.

To send a break signal during serial transmission, clear the SPB2DT bit to 0 (designating low level), and 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 SCIF_TXD pin.

(5) Receive Data Sampling Timing and Receive Margin in Asynchronous Mode

In asynchronous mode, the SCIF operates on a base clock with a frequency of 16 times the bit 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 17.22.

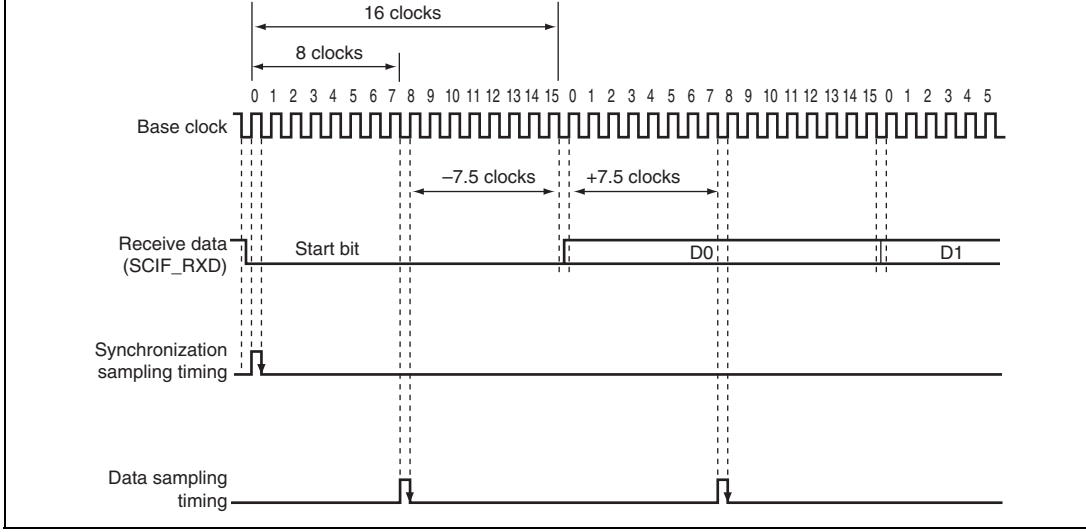


Figure 17.22 Receive Data Sampling Timing in Asynchronous Mode

Thus, the reception margin in asynchronous mode is given by formula (1).

$$M = \left| (0.5 - \frac{1}{2N}) - (L - 0.5)\,F - \frac{|D - 0.5|}{N}(1 + F) \right| \times 100\ \% \quad \ldots\ldots\ldots\ldots\ (1)$$

M: Receive margin (%)
N: Ratio of bit rate to clock (N = 16)
D: Clock duty (D = 0 to 1.0)
L: Frame length (L = 9 to 12)
F: Absolute value of clock rate deviation

From equation (1), if F = 0 and D = 0.5, the reception margin is 46.875%, as given by formula (2).

When D = 0.5 and F = 0:

$$M = (0.5 - 1 / (2 \times 16)) \times 100\% = 46.875\% \quad \ldots\ldots\ldots\ldots\ (2)$$

However, this is a theoretical value. A reasonable margin to allow in system designs is 20% to 30%.

When using an external clock as the synchronization clock, after SCFTDR is updated by the DMAC, an external clock should be input after at least five peripheral clock (Pck) cycles. A malfunction may occur when the transfer clock is input within four cycles after updating SCFTDR (see figure 17.23).

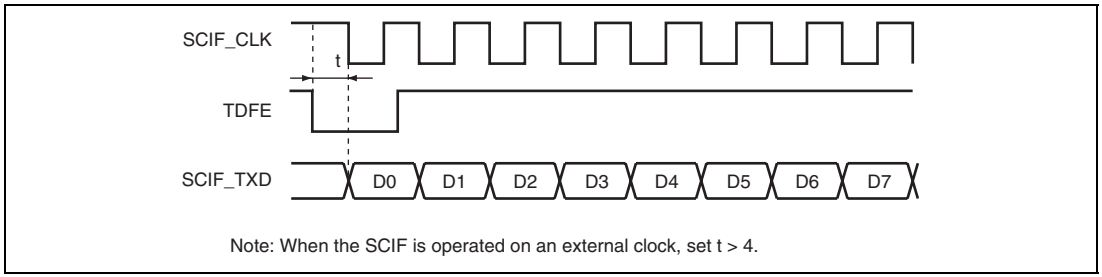


Figure 17.23 Example of Synchronization Clock Transfer by DMAC

(7) Note on the TEND Flag

In clock synchronous mode, when data is written to the transmit FIFO data register (SCFTDR) using the DMAC, there are cases where the transmit end flag (SCFSR.TEND) will not be cleared.

Workaround: Applications should read SCFSR.TEND using the following procedure.

1. Verify that the data transfer has completed (CHCR.TE = 1) in the DMAC.
2. Read the TEND flag.
3. If the TEND flag is 1, clear it to 0 and then read the TEND flag again.
4. Use the TEND flag value read out the second time.

The smart card interface supports IC cards (smart cards) conforming to the ISO/IEC 7816-3 (Identification Card) specification.

18.1 Features

The SIM has the following features.

- Communication functions
  Asynchronous half-duplex transmission
  Protocol selectable between T = 0 and T = 1 modes
  Data length: 8 bits
  Parity bit generation and check
  Extra guard time selectable
  Output clock cycles per etu selectable
  Transmission of error signal (parity error) in receiver mode with T = 0
  Detection of error signal and automatic character repetition in transmitter mode with T = 0
  Minimum character interval 11 etu selectable for T = 1
  Direct convention/inverse convention selectable
  Output clock fixable in high or low state
- Baud rate freely specifiable with internal baud rate generator
- Four types of interrupt factor
  Transmit data empty, receive data full, transmit/receive error, transmission complete
- DMA transfer
  The direct memory access controller (DMAC) can be activated to be used for data transfer using a transmit data empty DMA transfer request or a receive data full DMA transfer request. The receiver can observe the interval between the start of two successive characters.

Legend: etu (Elementary time unit): Time for 1-bit transfer

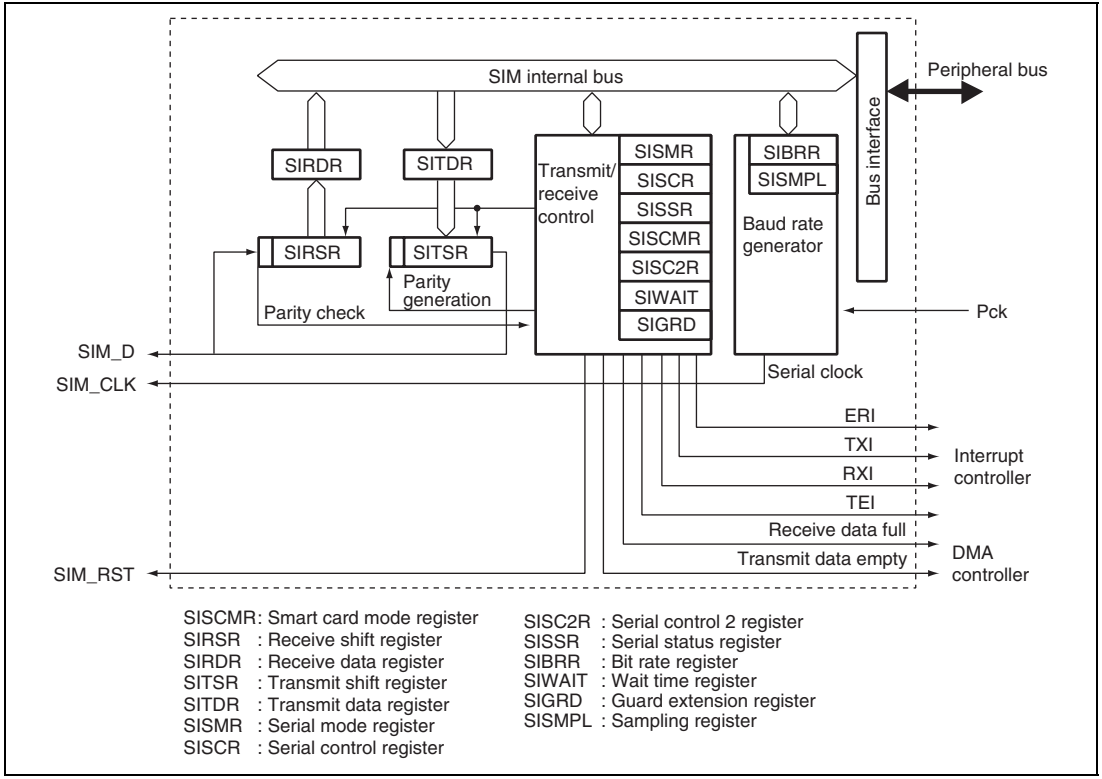


Figure 18.1 Block Diagram of SIM

18.2 Input/Output Pins

The pin configuration of the SIM is shown in table 18.1.

Table 18.1 Pin Configuration

Name	Abbreviation	I/O	Function
Transmit/receive data	SIM_D	Input/Output	Transmit/receive data input/output
Clock output	SIM_CLK	Output	Clock output
Smart card reset	SIM_RST	Output	Smart card reset output

SIM has the following registers. For the addresses of these registers and the register states in various processing states, refer to section 32, List of Registers.

Table 18.2 Register Configuration (1)

Register Name	Abbrev.	R/W	P4 Address	Area 7 Address	Size	Sync Clock
Serial mode register	SISMR	R/W	H'FE48 0000	H'1E48 0000	8	Pck
Bit rate register	SIBRR	R/W	H'FE48 0002	H'1E48 0002	8	Pck
Serial control register	SISCR	R/W	H'FE48 0004	H'1E48 0004	8	Pck
Transmit data register	SITDR	R/W	H'FE48 0006	H'1E48 0006	8	Pck
Serial status register	SISSR	R/W	H'FE48 0008	H'1E48 0008	8	Pck
Receive data register	SIRDR	R	H'FE48 000A	H'1E48 000A	8	Pck
Smart card mode register	SISCMR	R/W	H'FE48 000C	H'1E48 000C	8	Pck
Serial control 2 register	SISC2R	R/W	H'FE48 000E	H'1E48 000E	8	Pck
Wait time register	SIWAIT	R/W	H'FE48 0010	H'1E48 0010	16	Pck
Guard extension register	SIGRD	R/W	H'FE48 0012	H'1E48 0012	8	Pck
Sampling register	SISMPL	R/W	H'FE48 0014	H'1E48 0014	16	Pck

Register Name	Abbrev.	Power-on Reset by $\overline{\text{RESET}}$ Pin/WDT/ H-UDI	Manual Reset by $\overline{\text{RESET}}$ Pin/WDT/ Multiple Exception	Sleep by Sleep Instruction/ Deep Sleep	Standby by Hardware	Standby by Software/ Each Module
Serial mode register	SISMR	H'20	H'20	Retained	*	Retained
Bit rate register	SIBRR	H'07	H'07	Retained		Retained
Serial control register	SISCR	H'00	H'00	Retained		Retained
Transmit data register	SITDR	H'FF	H'FF	Retained		Retained
Serial status register	SISSR	H'84	H'84	Retained		Retained
Receive data register	SIRDR	H'00	H'00	Retained		Retained
Smart card mode register	SISCMR	H'01	H'01	Retained		Retained
Serial control 2 register	SISC2R	H'00	H'00	Retained		Retained
Wait time register	SIWAIT	H'0000	H'0000	Retained		Retained
Guard extension register	SIGRD	H'00	H'00	Retained		Retained
Sampling register	SISMPL	H'0173	H'0173	Retained		Retained

Note: * After exiting hardware standby mode, this LSI enters the power-on reset state by the $\overline{\text{RESET}}$ pin.

SISMR is an 8-bit readable/writable register that selects settings for the communication format of the smart card interface.

Bit:	7	6	5	4	3	2	1	0
	-	-	PE	O/$\overline{E}$	-	-	-	-
Initial value:	0	0	1	0	0	0	0	0
R/W:	R	R	R	R/W	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 0	R	Reserved These bits are always read as 0, and the write value should always be 0.
5	PE	1	R	Parity Enable This bit is always read as 1.The write value should always be 1.
4	O/$\overline{E}$	0	R/W	Parity Mode Selects whether even or odd parity is to be used when adding a parity bit and checking parity. 0: Even parity*[1] 1: Odd parity*[2] Notes: 1. When even parity is specified, the transmitter will add a parity bit if the number of transmitted characters (1s) is odd, so that the total number of set bits (1s) is always even. The receiver checks whether the total number of set bits (1s), including a parity bit and received characters, is even. 2. When odd parity is specified, the transmitter will add a parity bit if the number of transmitted characters (1s) is even, so that the total number of set bits (1s) is always odd. The receiver checks whether the total number of set bits (1s), including a parity bit and received characters, is odd.
3 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

SIBRR is an 8-bit readable/writable register that sets the transmission/reception bit rate.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	-	BRR2	BRR1	BRR0
Initial value:	0	0	0	0	0	1	1	1
R/W:	R	R	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Explanation
7 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2	BRR2	1	R/W	These bits set the transmit/receive bit rates 2 to 0.
1	BRR1	1	R/W	
0	BRR0	1	R/W	

The SIBRR setting can be determined from the following formula.

$$\text{SIM\_CLK} = \frac{\text{Pck}}{2\ (\text{brr} + 1)}$$

The units of Pck (peripheral clock) and SIM_CLK output clock are MHz.

brr is the value specified in BRR[2:0].

18.3.3 Serial Control Register (SISCR)

SISCR is an 8-bit readable/writable register that enables or disables transmission or reception operation, the serial clock output, and interrupt requests for the smart card interface.

Bit:	7	6	5	4	3	2	1	0
	TIE	RIE	TE	RE	WAIT_IE	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

7	TIE	0	R/W	Transmit Interrupt Enable Enables or disables transmit data empty interrupt (SIMTXI) requests when serial transmit data is transferred from SITDR to SITSR, and the TDRE flag in SISSR is set to 1. 0: Disables transmit data empty interrupt (SIMTXI) requests* 1: Enables transmit data empty interrupt (SIMTXI) requests Note: * SIMTXI can be canceled by clearing either the TDRE flag or TIE to 0.
6	RIE	0	R/W	Receive interrupt enable Enables or disables receive data full interrupt (SIMRXI) requests, and transmit/receive error interrupt (SIMERI) requests due to parity errors, overrun errors, and error signal status, when serial receive data is transferred from SIRSR to SIRDR, and the RDRF flag in SISSR register is set to 1. 0: Disables receive data full interrupt (SIMRXI) requests and transmit/receive error interrupt (SIMERI) requests*[1]*[2] 1: Enables receive data full interrupt (SIMRXI) requests and transmit/receive error interrupt (SIMERI) requests*[2] Notes: 1. SIMRXI and SIMERI interrupt requests can be canceled by clearing the RDRF flag or the PER, ORER or ERS flag or the RIE bit to 0. 2. Wait error interrupt (SIMERI) requests are enabled and disabled using the WAIT_IE bit in SISCR.
5	TE	0	R/W	Transmit Enable Enables/disables serial transmission operations. 0: Disables transmission*[1] 1: Enables transmission*[2] Notes: 1. The TDRE flag in SISSR register is fixed at 1. 2. In this state, if transmit data is written to SITDR, the transmission operation is started. Before setting the TE bit to 1, SISMR and SISCMR must always be set to determine the transmission format.

4	RE	0	R/W	Receive Enable Enables/disables serial reception. 0: Disables reception*[1] 1: Enables reception*[2] Notes: 1. Note that clearing the RE bit to 0 has no effect on the RDRF, PER, ERS, ORER, or WAIT_ER flags. The state of these flags will be maintained. 2. If the start bit is detected in this state, serial reception is initiated. Before setting the RE bit to 1, SISMR and SISCMR registers must always be set, to determine the reception format.
3	WAIT_IE	0	R/W	Wait Enable Enables or disables wait error interrupt requests. 0: Disables wait error interrupt (SIMERI) requests 1: Enables wait error interrupt (SIMERI) requests
2	TEIE	0	R/W	Transmit end interrupt enable Enables or disables transmission end interrupt (SIMTEI) requests when transmission ends and the TEND flag is set to 1. 0: Disables transmission end interrupt (SIMTEI) requests* 1: Enables transmission end interrupt (SIMTEI) requests* Note: * After the 1 in the TDRE flag in SISSR is read, SIMTEI can be cancelled either by writing transmit data to SITDR and then clearing the TEND bit, or by clearing the TEIE bit to 0.
1 0	CKE1 CKE0	0 0	R/W R/W	Clock Enable 1, 0 Select the clock source for the smart card interface, and enables or disables clock output from the SIM_CLK pin. 00: Output pin fixed at low level output 01: Output pin set for clock output 10: Output pin fixed at high level output 11: Output pin set for clock output

SITSR is a shift register for transmitting serial data.

The smart card interface transfers transmit data from SITDR to the SITSR register, and then sends the data in order from the LSB or MSB to the SIM_D pin to perform serial data transmission.

After one-byte data has been transmitted, the next data to be transmitted is automatically transferred from SITDR to SITSR, and transmission begins. When the TDRE flag in the SISSR is set to 1, no data is transferred from SITDR to SITSR.

It is not possible to read/write a SITSR directly from a CPU or DMAC.

18.3.5 Transmit Data Register (SITDR)

SITDR is an 8-bit readable/writable register that stores data for serial transmission.

When the smart card interface detects a vacancy in SITSR, transmit data written to SITDR is transferred to SITSR, and serial transmission is initiated. Consecutive serial transmission is possible by writing the next piece of transmit data to SITDR while SITSR serial data is being transmitted.

Bit:	7	6	5	4	3	2	1	0
	SITD7	SITD6	SITD5	SITD4	SITD3	SITD2	SITD1	SITD0
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	Bit Name	Initial Value	R/W	Description
7 to 0	SITD 7 to 0	All 1	R/W	Transmit Data Store data for serial transmission.

SISSR is an 8-bit readable/writable register that indicates the operating state of the smart card interface.

Bit:	7	6	5	4	3	2	1	0
	TDRE	RDRF	ORER	ERS	PER	TEND	WAIT_ER	-
Initial value:	1	0	0	0	0	1	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R	R/W	R

Bit	Bit Name	Initial Value	R/W	Description
7	TDRE	1	R/W	Transmit Data Register Empty Indicates that data was transferred from SITDR to SITSR, and that the next set of serial transmit data can be written to SITDR. 0: Indicates that valid transmit data is written to the SITDR. [Clearing Conditions] • When the TE bit in SISCR is 1, and data is written to SITDR • When 0 is written to the TDRE bit 1: Indicates that there is no valid transmit data in SITDR. [Setting Conditions]* • On reset • When the TE bit in SISCR is 0 • When data is transferred from SITDR to SITSR, and data can be written to SITDR Note: * Writing 1 will retain the original value.

6	RDRF	0	R/W	Receive Data Register Full Indicates that received data is stored in SIRDR*[1]. 0: Indicates that no valid received data is stored in SIRDR. [Clearing Conditions] • On reset • When data is read from SIRDR • When 0 is written to the RDRF bit 1: Indicates that valid received data is stored in SIRDR. [Setting Condition]*[2] • When serial reception is completed normally, and received data is transferred from SIRSR to SIRDR. Notes: 1. In the T = 0 mode, when the receiver detects a parity error, it retains the previous state without affecting the SIRDR contents and RDRF flag. On the other hand, in the T = 1 mode, when the receiver detects a parity error, it transfers the received data to SIRDR, and sets the RDRF flag to 1. In both the T = 0 and T = 1 modes, clearing the RE bit in SISCR to 0 will retain the previous state without affecting the SIRDR contents and RDRF flag. 2. Writing 1 will retain the original value.

5	ORER	0	R/W	Overrun Error Indicates that an overrun error resulting in abnormal termination has occurred during reception.. 0: Indicates that reception is in progress, or that reception was completed normally*[1] [Clearing Conditions] • On reset • When 0 is written to ORER 1: Indicates that an overrun error occurred during reception*[2] [Setting Condition]*[3] • When the next serial reception is completed in the RDRF = 1 state. Notes: 1. Clearing the RE bit in SISCR to 0 will retain the previous state without affecting the ORER flag. 2. SIRDR loses the data received before the overrun error but retains the data received at the time the overrun error occurred. Furthermore, if ORER is set to 1, then subsequent serial reception cannot continue. 3. Writing 1 will retain the original value.
4	ERS	0	R/W	Error Signal Status This flag indicates the status of error signals returned from the receiver during transmission. In the T = 1 mode, this bit is not set. 0: Indicates that an error signal indicating detection of a parity error was not sent from the receiver. [Clearing Conditions] • On reset • When 0 is written to ERS 1: Indicates that an error signal indicating detection of a parity error was sent from the receiver. [Setting Condition] • When an error signal is sampled. Note: Clearing the TE bit in SISCR to 0 will retain the previous state without affecting the ERS flag.

3	PER	0	R/W	Parity Error Indicates that a parity error has occurred during reception, resulting in abnormal termination. 0: Indicates that reception is in progress, or that reception was completed normally.*[1] [Clearing Conditions] • On reset • When 0 is written to PER 1: Indicates that a parity error occurred during reception.*[2] [Setting Condition] • When the number of logic 1 digits combined in the receive data and parity bit does not match the setting of the even/odd parity specified by the serial mode register's (SISMR) O/E bit during reception. Notes: 1. Clearing the RE bit in SISCR to 0 will retain the previous state without affecting the PER flag. 2. In the T = 0 mode, the receiver does not transfer the data received when a parity error occurs to SIRDR, and sets the RDRF flag. On the other hand, in the T = 1 mode, the data received when a parity error occurs is sent to SIRDR, and the RDRF flag is set. When a parity error occurs, clear the PER flag to 0 before the sampling timing for the next parity bit. When PER is set to 1, data reception can continue. However, the PER flag will not be cleared when reception is completed successfully.

2	TEND	1	R	Transmit End Indicates that transmission is ended. The TEND flag is read-only, and cannot be written. 0: Indicates that transmission is in progress [Clearing Condition] • Transmit data is transferred from SITDR to SITSR, and serial transmission is initiated. 1: Indicates that transmission is ended [Setting Conditions] • On reset • When ERS = 0 (normal transmission) after transmission of one byte of serial characters and a parity bit Note: The TEND flag is set 1 etu before the end of the guardtime.

1	WAIT_ER	0	R/W	Wait Error Indicates the wait timer error status. 0: Indicates that the interval between the start of two successive characters has not exceeded the etu set by SIWAIT. [Clearing Conditions] • On reset • When 0 is written to WAIT_ER while its value is 1 1: Indicates that the interval between the start of two successive characters has exceeded the etu set by SIWAIT. [Setting Conditions] • In T = 0 mode, when the interval between the start of two successive characters exceeds the etu (value of 60 x SIWAIT: working wait time). • In T = 1 mode, when the interval between the start of two successive characters exceeds the etu (SIWAIT value: Guardtime). Notes: 1. Even if the RE bit in SISCR is cleared to 0, the WAIT_ER flag is unaffected, and the previous state is maintained. 2. In T = 0 mode, changing the RE bit from 0 to 1 may not set the WAIT_ER bit, even if the setting conditions for the WAIT_ER bit are satisfied. In this condition, the WAIT_ER bit is set at the timing of 60 × (SCWAIT + n) etu after the last transmission or reception. n is a whole number and it depends on the timing at which the RE bit is set. 3. In T = 0 mode, to avoid making the WAIT_ER bit set at the timing of 60 × (SCWAIT + n) etu after the last transmission or reception, the following procedure should be followed: Change the protocol bit (PB) in the smart card mode register (SISCMR) from 0 to 1 and again change the PB bit to 0. In T = 1 mode, to avoid making the WAIT_ER bit set at the timing of (SCWAIT) etu after the last reception, the following procedure should be followed: Change the PB bit in SISCMR from 1 to 0 and again change the PB bit to 1.

0	—	0	R	Reserved This bit is always read as 0. The write value should also always be 0.

Legend: etu: Elementary time unit

18.3.7 Receive Shift Register (SIRSR)

SIRSR is a register for reception of serial data.

The smart card interface receives serial data input from the SIM_D pin in order, from the LSB or MSB, and sets it in SIRSR, converting it to parallel data. When reception of one byte of data is completed, the data is automatically transferred to SIRDR. The CPU cannot directly read or write SIRSR.

18.3.8 Receive Data Register (SIRDR)

SIRDR is an 8-bit read-only register that stores received serial data.

When reception of one byte of serial data is completed, the smart card interface transfers the received serial data from SIRSR to SIRDR for storage, and completes the reception operation. Thereafter, SIRSR can receive data. In this way, SIRSR and SIRDR constitute a double buffer, enabling continuous reception of data. SIRDR cannot be written from the CPU.

Bit:	7	6	5	4	3	2	1	0
	SIRD7	SIRD6	SIRD5	SIRD4	SIRD3	SIRD2	SIRD1	SIRD0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	SIRD 7 to 0	All 0	R	Receive Data These bits store received serial data.

SISCMR is an 8-bit readable/writable register that selects smart card interface functions.

Bit:	7	6	5	4	3	2	1	0
	-	LCB	PB	-	SDIR	SINV	RST	SMIF
Initial value:	0	0	0	0	0	0	0	1
R/W:	R	R/W	R/W	R	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
6	LCB	0	R/W	Last Character When this bit is set to 1, the guardtime is 2 etu, and the setting of the guard extension register is invalid. 0: The guardtime is determined by the guard register value 1: The guardtime is 2 etu
5	PB	0	R/W	Protocol Selects the T = 0 or T = 1 protocol. 0: The smart card interface operates according to the T = 0 protocol 1: The smart card interface operates according to the T = 1 protocol
4	—	0	R	Reserved This bit is always read as 0. The write value should also always be 0.
3	SDIR	0	R/W	Smart Card Data Transfer Direction Selects the format for serial/parallel conversion. 0: Transmit the SITDR contents LSB-first Received data is stored in SIRDR LSB-first. 1: Transmit the SITDR contents MSB-first Received data is stored in SIRDR MSB-first.
2	SINV	0	R/W	Smart Card Data Inversion Specifies inversion of the data logic level. In combination with the function of bit 3, used for transmission to and reception from an inverse convention card. SINV does not affect the parity bit. 0: Transmit the SITDR contents without change Store received data in SIRDR without change 1: Invert and transmit the SITDR data Invert and store received data in SIRDR

1	RST	0	R/W	Smart Card Reset Controls the output of the SIM_RST pin of the smart card interface. 0: The SIM_RST pin of the smart card interface outputs low level. 1: The SIM_RST pin of the smart card interface outputs high level.
0	SMIF	1	R/W	Smart Card Interface Mode Select This bit is always read as 1. The write value should also always be 1.

Legend: etu: Elementary time unit

18.3.10 Serial Control 2 Register (SISC2R)

SISC2R is an 8-bit readable/writable register that enables or disables receive data full interrupt (SIMRXI) requests.

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

Bit	Bit Name	Initial Value	R/W	Description
7	EIO	0	R/W	Error Interrupt Only When the EIO bit is 1, even if the RIE bit is set to 1, a receive data full interrupt (SIMRXI) request is not sent to the CPU. When the DMAC is used with this setting, the CPU processes only SIMERI requests. Receive data full interrupt (SIMRXI) requests are enabled or disabled by the RIE bit setting.
6 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should also always be 0.

SIGRD is an 8-bit readable/writable register that sets the time added for guardtime.

Bit:	7	6	5	4	3	2	1	0
	SIGRD7	SIGRD6	SIGRD5	SIGRD4	SIGRD3	SIGRD2	SIGRD1	SIGRD0
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	Bit Name	Initial Value	R/W	Description
7 to 0	SIGRD 7 to 0	All 0	R/W	Guard Extension Indicates the time added for guardtime after transmitting a character to the smart card. The interval between the start of two successive characters is 12 etu (no addition) when this register value is H'00, is 13 etu when the value is H'01, and so on, up to 266 etu for H'FE. If the value of this register is H'FF, the interval between the start of two successive characters is 11 etu in T = 1 mode and is 12 etu in T = 0 mode.

Legend: etu: Elementary time unit

SIWAIT is a 16-bit readable/writable register. If the interval between the start of two successive characters exceeds the value set (in etu units), a wait time error is generated.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SIWAIT15	SIWAIT14	SIWAIT13	SIWAIT12	SIWAIT11	SIWAIT10	SIWAIT9	SIWAIT8	SIWAIT7	SIWAIT6	SIWAIT5	SIWAIT4	SIWAIT3	SIWAIT2	SIWAIT1	SIWAIT0
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	Bit Name	Initial Value	R/W	Description
15 to 0	SIWAIT 15 to 0	All 0	R/W	Wait Time Register T=0 In this mode, the Wait Time Register can set a working wait time. If the interval between a character being received and the start character that has previously been received/transmitted exceeds (60 x the value set in this register) etu, the WAIT_ER flag is set to 1. However, when SIWAIT=H'0000, the WAIT_ER flag will be set after 60 etu. T = 1 In this mode, the Wait Time Register can set a character wait time. If the interval between the start of two successive characters received exceeds the etu (the value set in this register), the WAIT_ER bit flag is set to 1. However, when SIWAIT=H'0000, the WAIT_ER flag will be set after 1 etu.

Legend: etu: Elementary time unit

SISMPL is a 16-bit readable/writable register that sets the number of serial clock cycles per etu.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	SISMPL10	SISMPL9	SISMPL8	SISMPL7	SISMPL6	SISMPL5	SISMPL4	SISMPL3	SISMPL2	SISMPL1	SISMPL0
Initial value:	0	0	0	0	0	0	0	1	0	1	1	1	0	0	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

Bit	Bit Name	Initial Value	R/W	Description
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	SISMPL 10 to 0	H'173	R/W	Setting for the number of serial clock cycles per etu The number of serial clock cycles per etu is (SISMPL value + 1). The value written to SISMPL should always be H'0007 or greater.

Legend: etu: Elementary time unit

18.4 Operation

The main functions of the SMI are as follows.

1. One frame consists of eight data bits and one parity bit.
2. The transmitter inserts the guardtime, specified by SIGRD and the LCB and PB bits in SISCMR, between the end of each parity bit and the beginning of the next frame.
3. When detecting a parity error, the receiver in T = 0 mode outputs low level for 1 etu as an error signal, after 10.5 etu has passed since the start bit was received.
4. When sampling an error signal, the transmitter in T = 0 mode automatically repeats the disputed data after a delay of at least 2 etu.
5. Only asynchronous communication functions are supported; there is no clock-synchronized communication function.

18.4.1 Data Format

Figure 18.2 shows the data format used by the smart card interface. The smart card interface performs a parity check for each frame during reception.

When detecting a parity error, the receiver in T = 0 mode returns an error signal to the transmitter, requesting data repetition. The transmitter samples error signals and repeats the disputed data.

transmitter does not sample error signals or repeat the disputed data.

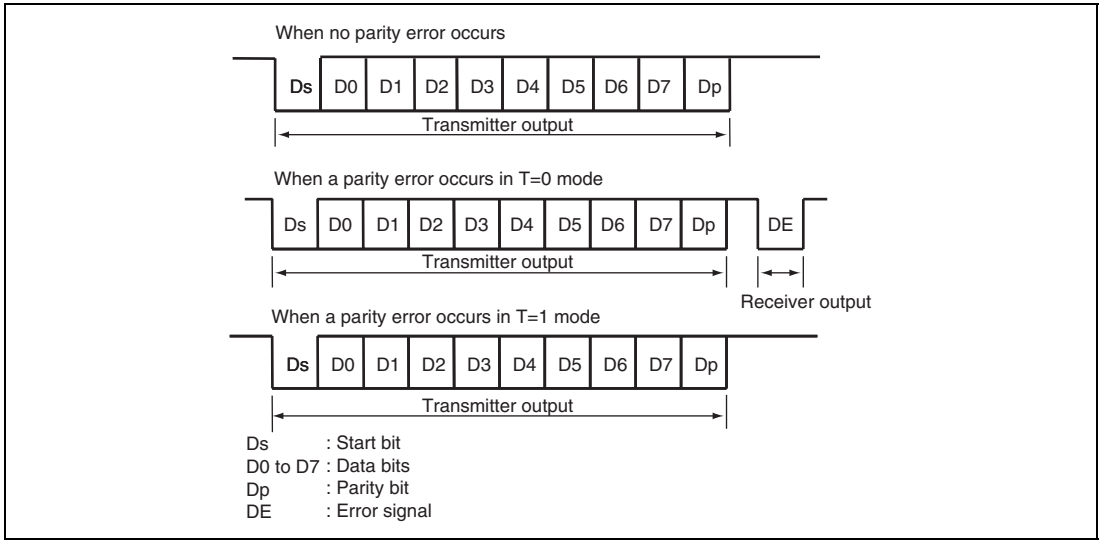


Figure 18.2 Data Format Used by the Smart Card Interface

The operation sequence is as follows.

1. When the SIM is not in use, the data line is in the high-impedance state, fixed at high level by the pull-up resistance.
2. The transmitter starts transmission of one frame of data. The data frame begins with the start bit (Ds: low level). This is followed by eight data bits (D0 to D7) and the parity bit (Dp).
3. The SIM then returns the data line to high impedance. The data line is held at high level by the pull-up resistance.
4. The receiver performs a parity check.

   If detecting no parity error and receiving the data normally, the receiver waits for the next frame, without taking further action.

   If a parity error has occurred, the receiver outputs an error signal (DE: low level) for requesting data repetition in T = 0 mode. After outputting an error signal for the specified duration, the receiver again sets the signal line to high impedance. The signal line returns to high level by the pull-up resistance. In T = 1 mode, however, the receiver outputs no error signal when a parity error occurs.
5. If the transmitter does not receive an error signal, it transmits the next frame.

   If the transmitter in T = 0 mode receives an error signal, it repeats the disputed data as in step 2 above. In T = 1 mode, however, the transmitter receives no error signals and performs no repetition.

Table 18.3 shows a map of the bits in the registers used by the smart card interface.

Set 0 or 1 to a bit if indicated as 0 or 1 in the following table. Set a bit without 0/1 indication according to below instructions.

Table 18.3 Register Settings for the Smart Card Interface

	Bit							
Register	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
SISMR	0	0	PE	O/$\overline{E}$	0	0	0	0
SIBRR	0	0	0	0	0	BRR2	BRR1	BRR0
SISCR	TIE	RIE	TE	RE	WAIT_IE	TEIE	CKE1	CKE0
SITDR	SITD[7:0]							
SISSR	TDRE	RDRF	ORER	ERS	PER	TEND	WAIT_ER	0
SIRDR	SIRD[7:0]							
SISCMR	0	LCB	PB	0	SDIR	SINV	RST	SMIF
SISC2R	EIO	0	0	0	0	0	0	0
SIWAIT	SIWAIT[15:0] (16-bit register)							
SIGRD	SIGRD[7:0]							
SISMPL	SISCMPL[10:0] (16-bit register, but bits 15 to 11 are 0)							

(1) Serial mode register (SISMR) setting

When the IC card is set for the direct convention, the O/$\overline{E}$ bit is set to 0; for the inverse convention, it is set to 1.

(2) Bit rate register (SIBRR) setting

Sets the bit rate. For the method of computing settings, refer to section 18.4.3, Clocks.

(3) Serial control register (SISCR) settings

The different interrupts can be enabled and disabled using the TIE, RIE, TEIE, and WAIT_IE bits.

By setting either the TE or RE bit to 1, transmission or reception is selected.

The CKE1 and CKE0 bits are used to select the clock output state. For details, refer to section 18.4.3, Clocks.

When the IC card is set for the direct convention, both the SDIR and SINV bits are set to 0; for the inverse convention, both are set to 1. The SMIF bit is always set to 1.

Figure 18.3 shows the register settings and initial character waveform examples for two types of IC cards (a direct-convention type and an inverse-convention type).

For the direct-convention type, the logical level 1 is assigned to the Z state, and the logical level 0 to the A state, and transmission and reception are performed LSB-first. The data of the above initial character is then H'3B. Even parity is used according to the smart card specification, and so the parity bit is 1.

For the inverse-convention type, the logical level 1 is assigned to the A state, and the logical level 0 to the Z state, and transmission and reception are performed MSB-first. The data of the above initial character is then H'3F.

Even parity is used according to the smart card specification, and so the parity bit is 0, assigned to the Z state.

Inversion by the SINV bit setting only applies to data bits D7 to D0. To invert the parity bit, specify odd parity through the SISMR.O/$\overline{E}$ bit, both for transmission and reception.

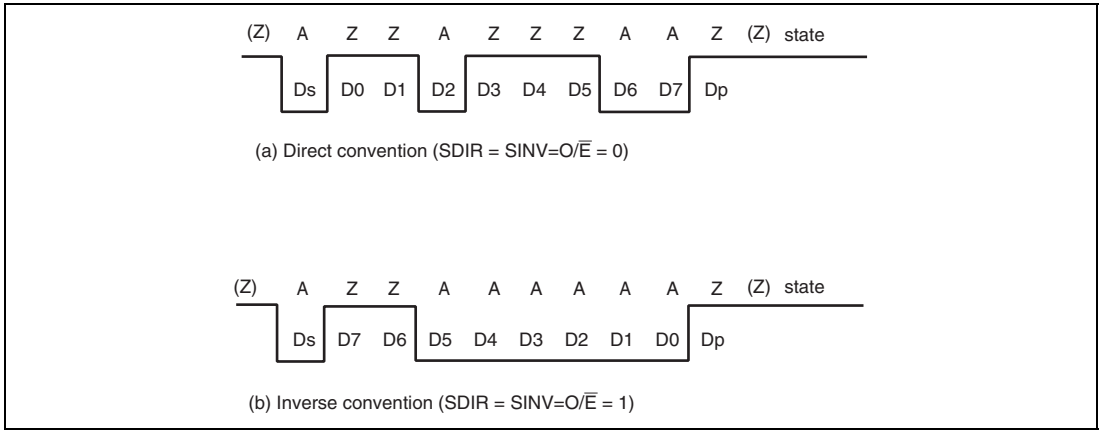


Figure 18.3 Examples of Initial Character Waveforms

Only the internal clock generated by the on-chip baud rate generator can be used as the transmission/reception clock in the smart card interface. The bit rate is set using SIBRR and SISMPL, using the formula indicated below. Examples of bit rates are shown in table 18.4.

Here, when CKE0 = 1 is used to select the clock output, a clock signal is output from the SIM_CLK pin with frequency equal to (SISMPL+1) times the bit rate.

$$B = Pck \times 10^6 / \{(S+1) \times 2\ (N+1)\}$$

Where

B: the bit rate (bits/s)

Pck: the operating frequency of the peripheral module [MHz]

S: the SISMPL setting ($0 \leq S \leq 2047$)

N: the SIBRR setting ($0 \leq N \leq 7$).

Table 18.4 Example of Bit Rates (bits/s) for SIBRR Settings (Pck = 33.3 MHz, SISMPL = 371)

SIBRR Setting	SIM_CLK Frequency (MHz)	Bit Rate (bits/s)
7	2.06	5544
6	2.36	6336
5	2.75	7392
4	3.30	8871
3	4.13	11089
2	5.50	14785
1	8.25	22177
0	16.50	44355

(1) Initialization

Prior to data transmission and reception, the procedure shown in figure 18.4 should be used to initialize the smart card interface. Initialization is also necessary when switching from transmitter mode to receiver mode, and when switching from receiver mode to transmitter mode. An example of the initialization process is shown in the flow chart of figure 18.4.

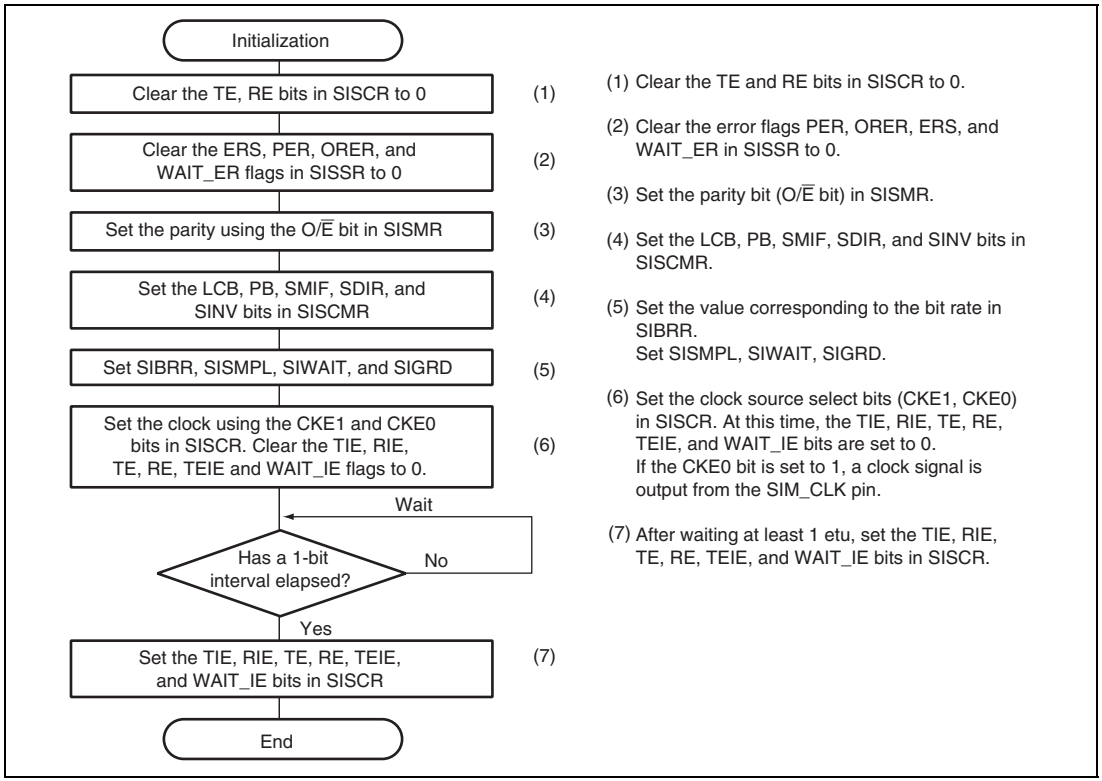
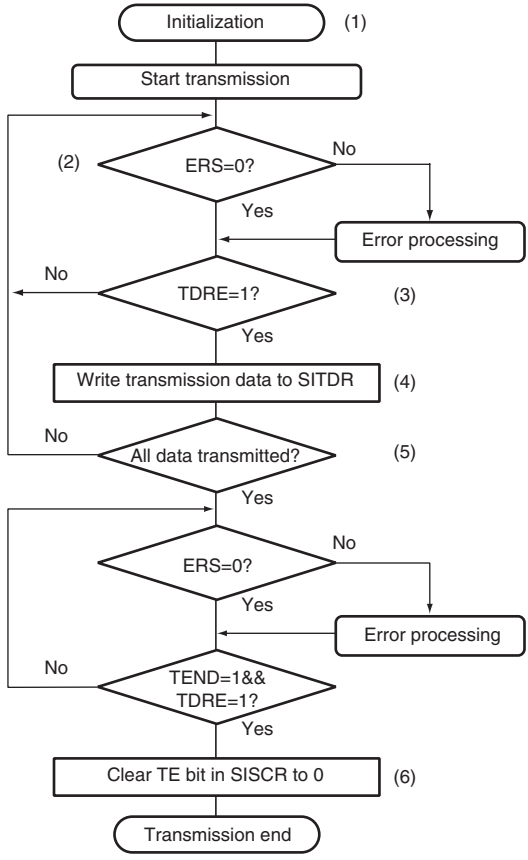


Figure 18.4 Example of Initialization Flow

Data transmission in smart card mode includes error signal sampling and repetition processing. An example of transmission processing is shown in figure 18.5.



(1) Follow the procedure of item 1 in section 18.4.4, and initialize the smart card interface.

(2) Confirm that the ERS bit (error flag) in SISSR is cleared to 0.

(3) Repeat (2) and (3) in the figure until it can be confirmed that the TDRE flag in SISSR is set to 1.

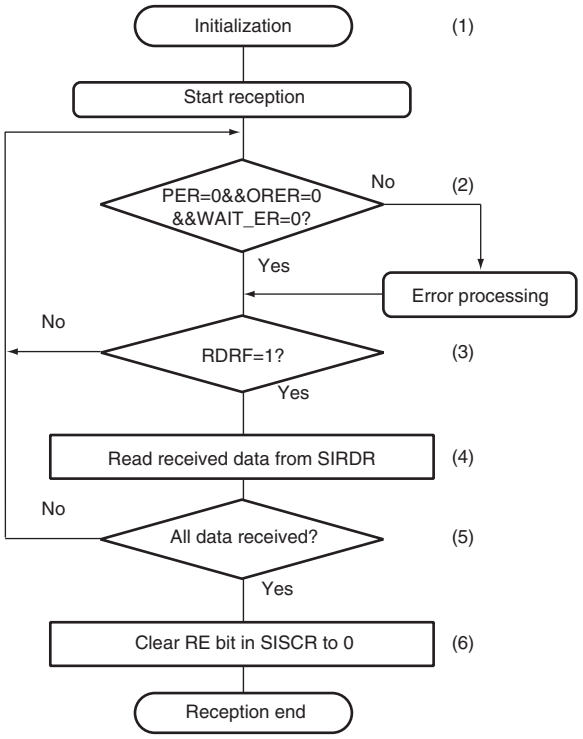
(4) Write the transmission data to SITDR, and perform the transmission. At this time, the TDRE flag is automatically cleared to 0. When transmission of the start bit is begun, the TEND flag is automatically cleared to 0, and the TDRE flag is automatically set to 1.

(5) When performing continuous data transmission, return to (2).

(6) When transmission is ended, clear the TE bit to 0.

Figure 18.5 Example of Transmission Processing

An example of data reception processing in smart card mode is shown in figure 18.6.



(1) Follow the procedure of item 1 in section 18.4.4, and initialize the smart card interface.

(2) Confirm that the PER, ORER, and WAIT_ER flags in SISSR are 0. If one of these flags is set, after performing the prescribed reception error processing, clear the PER, ORER, and WAIT_ER flags to 0.

(3) Repeat (2) and (3) in the figure until it can be confirmed that the RDRF flag is set to 1.

(4) Read received data from SIRDR.

(5) When performing continuous data reception, return to step (2).

(6) When reception is ended, clear the RE bit to 0.

Note: If during reception a parity error occurs and PER is set to 1, in T=0 mode the received data is not transferred to SIRDR, and so this data cannot be read.

Figure 18.6 Example of Reception Processing

When switching from receiver mode to transmitter mode, after confirming that reception has been completed, and then set RE = 0 and TE = 1. Completion of reception can be confirmed through the RDRF flag.

When switching from transmitter mode to receiver mode, after confirming that transmission has been completed, and then set TE = 0 and RE = 1. Completion of transmission can be confirmed through the TDRE and TEND flags.

(5) Interrupt operations

The smart card interface has four types of interrupt sources: transmit data empty interrupt (SIMTXI) requests, transmit/receive error interrupt (SIMERI) requests, receive data full interrupt (SIMRXI) requests, and transmission end interrupt (SIMTEI) requests.

Table 18.5 describes the interrupt sources for the smart card interface. Each of the interrupt requests can be enabled or disabled using the TIE, RIE, TEIE, and WAIT_IE bits in SISCR and the EIO bit in SISC2R. In addition, each interrupt request can be sent independently to the interrupt controller.

Table 18.5 Smart Card Interface Interrupt Sources

Operating State		Flags	Mask Bits	Interrupt Sources
Transmitter mode	Normal operation	TDRE	TIE	SIMTXI
		TEND	TEIE	SIMTEI
	Error	ERS	RIE	SIMERI
Receiver mode	Normal operation	RDRF	RIE, EIO	SIMRXI
	Error	ORER, PER	RIE	SIMERI
		WAIT_ER	WAIT_IE	SIMERI

(6) Data transfer using DMAC

The smart card interface enables reception and transmission in T=0 and T=1 modes using DMAC.

In transmission, when the TDRE flag in SISSR is set to 1, a transmit data empty DMA transfer request is issued. If a transmit data empty DMA transfer request is set in advance as a DMAC startup factor, the DMAC can be started and made to transfer data when a transmit data empty DMA transfer request occurs.

number of bytes assigned to the DMAC.

For error handling with an interruput request to the CPU in transmission using the DMAC, set the TIE bit to 0 to disable an SIMTXI request, and set the RIE bit to 1 to enable an SIMERI request. Clear the ERS flag by sending an interrupt request to the CPU since it is not automatically cleared once set when an error signal was received.

The receiver issues a receive data full DMA transfer request when the RDRF flag in SISSR is set to 1. It is possible to start the DMAC to transfer data with a receive data full DMA transfer request by setting this request as a DMAC startup factor in advance.

In T = 0 mode, if a parity error occurs during reception, a data repeat request is issued. Since the RDRF flag is not set and a DMA transfer request is not issued, it is possible to receive the number of bytes assigned to the DMAC.

For error handling with an interrupt to the CPU in reception using the DMAC, set the RIE and EIO bits to 1 to disable an SIMRXI interrupt and to enable only an SIMERI request.

Clear the PER, ORER, and WAIT_ER flags by sending an interrupt request to the CPU since they are not automatically cleared once set by a receive error.

For transmission/reception using the DMAC, the DMAC should be configured to be enabled before configuring the smart card interface.

18.5 Usage Notes

The following matters should be noted when using the smart card interface.

18.5.1 Receive data Timing

When the SISMPL register holds its initial value, the smart card interface operates at a serial clock frequency (SIM_CLK) 372 times the transfer rate.

During reception, the smart card interface samples the falling edge of the start bit using the serial SIM_CLK for internal synchronization. Received data is captured internally at the center of one etu (at the rising edge of the 186th SIM_CLK pulse when one etu takes 372 SIM_CLK cycles). This is shown in figure 18.7.

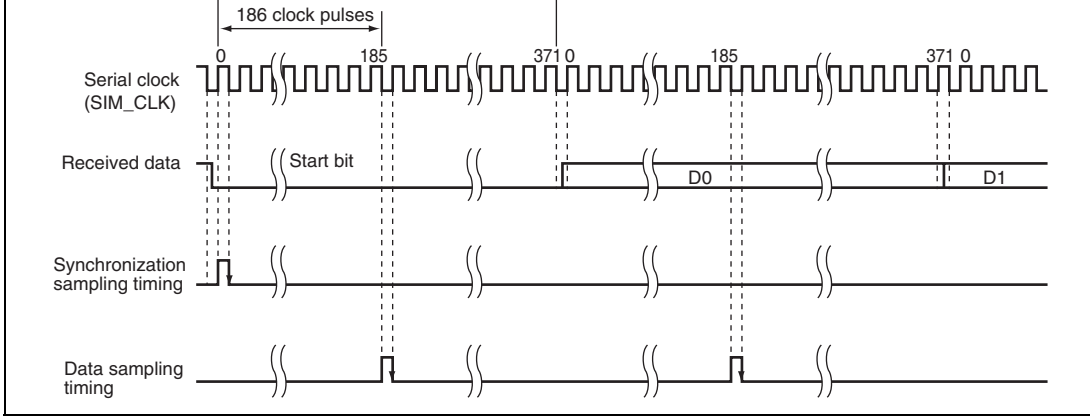


Figure 18.7 Received Data Sampling Timing in Smart Card Mode

18.5.2 Repetition when the Smart Card Interface Is in Receiver Mode (T = 0)

Figure 18.8 shows repetition operations when the smart card interface is in receiver mode. (1) to (5) in figure 18.8 correspond to items 1 to 5 described below.

1. If checking of the received parity bit detects an error, the PER bit in SISSR is automatically set to 1. If the RIE bit in SISCR is set for enable, an SIMERI request is issued. The PER bit in SISSR should be cleared to 0 before the sampling timing for the next parity bit.
2. The RDRF bit in SISSR is not set for frames in which a parity error occurs.
3. If checking of the received parity bit detects no error, the PER bit in SISSR is not set.
4. If checking of the received parity bit detects no error, it is assumed that reception was completed normally, and the RDRF bit in SISSR is automatically set to 1. If the RIE bit in SISCR is 1 and the EIO bit is 0, an SIMRXI request is generated.
5. If a normal frame is received, the pin is maintained in a high-impedance state at the timing for transmission of error signals.

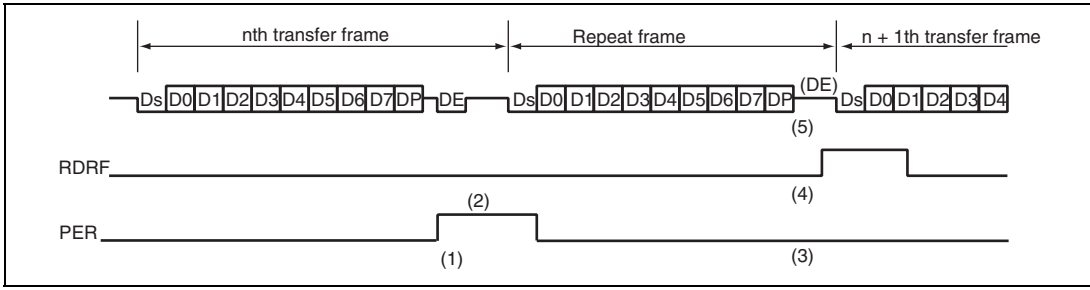


Figure 18.8 Repetition in the Smart Card Interface Receiver Mode

Figure 18.9 shows repetition operations when the smart card interface is in transmitter mode. (1) to (4) in figure 18.9 correspond to items 1 to 4 described below.

1. After completion of transmission of one frame, if an error signal is returned from the receiver, the ERS bit in SISSR is set to 1. If the RIE bit in SISCR is set to enable, an SIMERI request is issued. The ERS bit in SISSR should be cleared to 0 before the sampling timing for the next parity bit.
2. In T = 0 mode, the TEND bit in SISSR is not set for a frame when an error signal indicating an error is received.
3. If no error signal is returned from the receiver, the ERS bit in SISSR is not set.
4. If an error signal is not returned from the receiver, it is assumed that transmission of one frame, including repetition, is completed, and the TEND bit in SISSR is set to 1. At this time, if the TIE bit in SISCR is set to enable, a TEI interrupt request is issued.

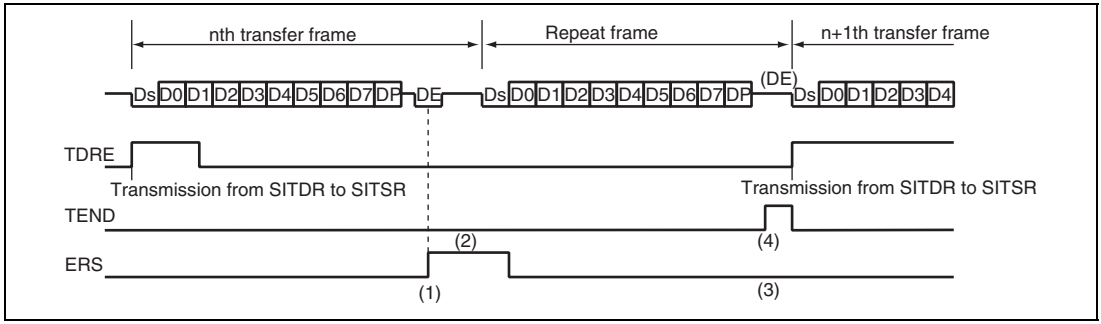


Figure 18.9 Repetition Standby Mode (clock stopped) when the Smart Card Interface Is in Transmitter Mode

If the TEIE bit is always set to 1 during continuous transmission, unnecessary transmit end interrupts (SIMTEI) occur because the TEND bit is set to 1 every time transmission is completed.

To avoid unnecessary SIMTEI requests, the TEIE bit in SISCR should be set to 1 only after the last transmit data is written to SITDR and SITSR starts transmission.

Figure 18.10 shows the signal waveforms when TEIE is set to 1.

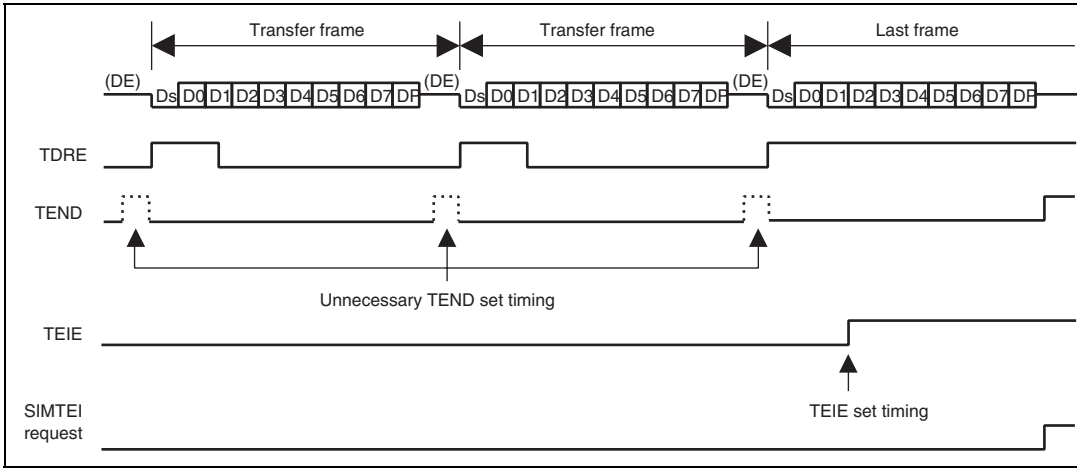


Figure 18.10 TEIE Set Timing

18.5.5 Standby Mode Switching

When switching between the smart card interface mode and standby mode, in order to maintain the clock duty, the following switching procedure should be used. Figure 18.11 shows standby mode switching procedure. (1) to (7) in figure 18.11 correspond to items 1 to 7 described below.

- When changing from smart card interface mode to standby mode:
  1. Write 0 to the TE and RE bits in SISCR, to stop transmission and reception operations. At the same time, set the CKE1 bit to the value for the output-fixed state in standby mode.
  2. Write 0 to the CKE0 bit in SISCR to stop the clock.
  3. Wait for one clock cycle of the serial clock. During this interval, the duty is maintained, and the clock output is fixed at the specified level.
  4. Make the transition to standby mode.

5. Cancel the standby state.
6. Set the CKE1 bit in SISCR to the value of the output-fixed state at the beginning of standby (the current SIM_CLK pin state).
7. Write 1 to the CKE0 bit in SISCR to output a clock signal. Clock signal generation begins at normal duty.

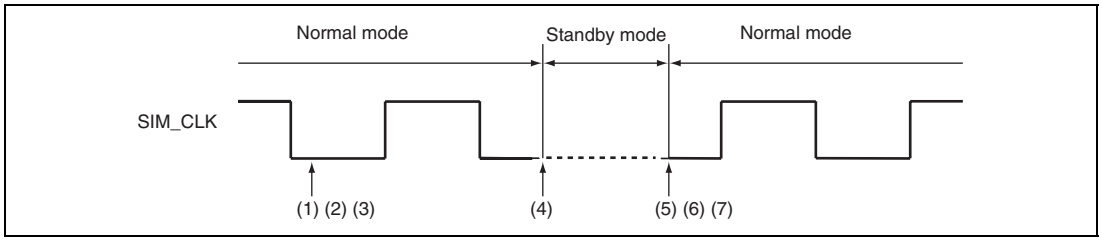


Figure 18.11 Procedure for Stopping the Clock and Restarting

18.5.6 Power-On and Clock Output

In order to maintain the clock duty from power-on, the following switching procedure should be used.

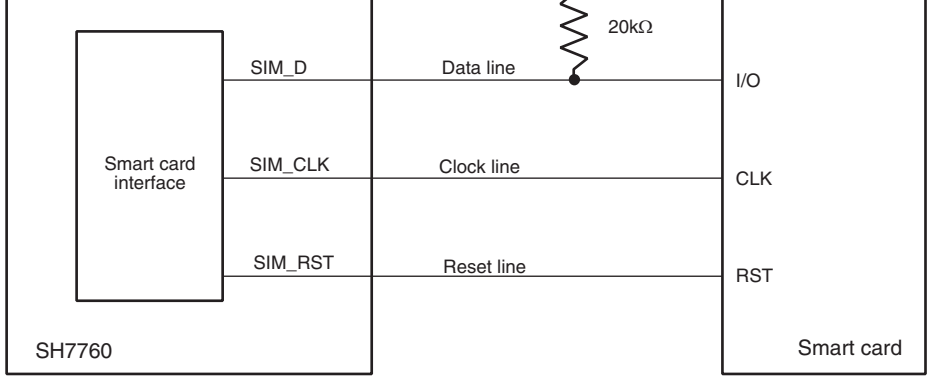
1. The clock output pin is set to an input port as the initial state and is pulled up in the chip.
2. Select the SIM through IPSELR in the PFC.
3. Use the CKE1 bit in SISCR to specify the fixed output.
4. Set the CKE0 bit in SISCR to 1 to start clock output.

18.5.7 Pin Connections

An example of pin connections for the smart card interface is shown in figure 18.12.

In communication with the smart card, transmission and reception are performed using a single data transmission line. The data transmission line should be pulled up by a resistance on the power supply V_{DDQ} side.

When using the clock generated by the smart card interface with the IC card, the SIM_CLK pin output is input to the CLK pin of the IC card. If an internal clock of the IC card is used, this connection is not needed.



Note: For details, refer to ISO/IEC7816-3.

Figure 18.12 Example of Smart Card Interface Pin Connections

This LSI contains an on-chip 2-channel I²C bus interface compatible with the Philips I²C bus (Inter IC Bus) interface. It should be noted, however, the register structure used to control the I²C bus differs from that of the Philips implementation.

19.1 Features

- Supports the Philips I²C bus interface.
- Multi-master capability.
- 7- or 10-bit address compatible master.
- 7-bit slave address.
- Fast mode compatible.
- Adaptable to a wide range of system clock frequencies.
- Supports 16-stage FIFO buffer mode.

Figure 19.1 shows a connection example for the I²C bus interface.

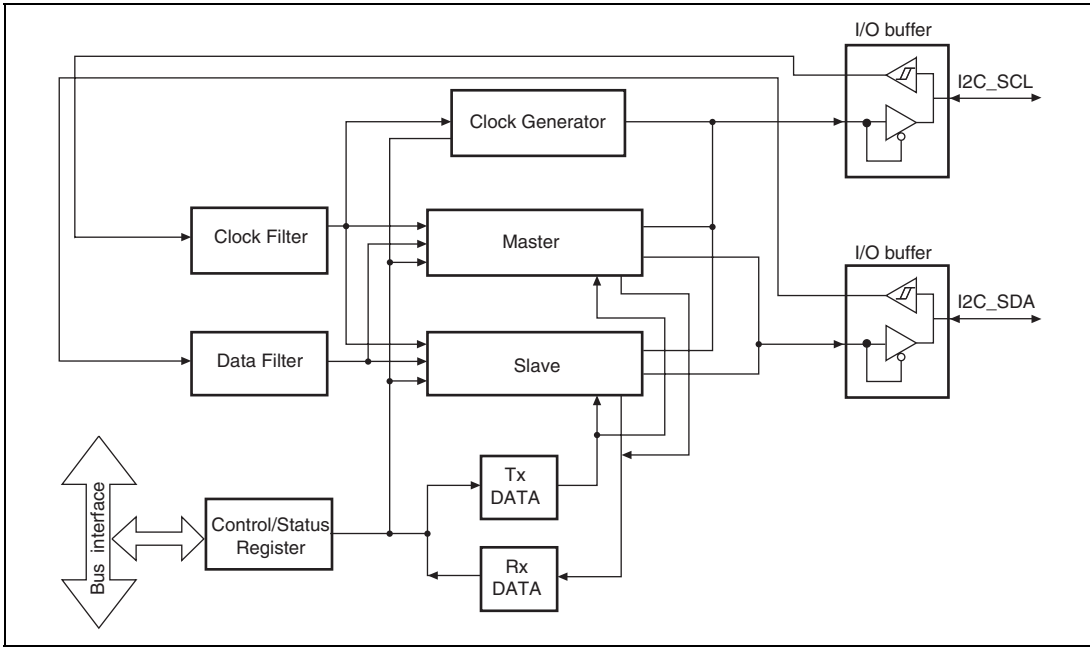


Figure 19.1 I²C Bus Interface Block Diagram

Table 19.1 lists the pins used in the I^2C bus interface.

Table 19.1 I^2C Bus Interface

Channel	Pin Name	Direction	Description
0	I2C0_SCL	Input/Output	I^2C 0 serial clock input output pin*
	I2C0_SDA	Input/Output	I^2C 0 serial data input output pin*
1	I2C1_SCL	Input/Output	I^2C 1 serial clock input output pin*
	I2C1_SDA	Input/Output	I^2C 1 serial data input output pin*

Note: * SCL/SDA output on the I^2C bus open drain pin.

19.3 Register Descriptions

All registers in the I^2C bus interface module are mapped to the register bus interface. For address and processing status of these registers, refer to section 32, List of Registers. Note that the channel numbers are omitted in the description.

Ch.	Register Name	Abbrev.	R/W	P4 Address	Area 7 Address	Size	Sync Clock
0	Slave control register 0	ICSCR0	R/W	H'FE14 0000	H'1E14 0000	32	Pck
	Master control register 0	ICMCR0	R/W	H'FE14 0004	H'1E14 0004	32	Pck
	Slave status register 0	ICSSR0	R/(W)*1	H'FE14 0008	H'1E14 0008	32	Pck
	Master status register 0	ICMSR0	R/(W)*2	H'FE14 000C	H'1E14 000C	32	Pck
	Slave interrupt enable register 0	ICSIER0	R/W	H'FE14 0010	H'1E14 0010	32	Pck
	Master interrupt enable register 0	ICMIER0	R/W	H'FE14 0014	H'1E14 0014	32	Pck
	Clock control register 0	ICCCR0	R/W	H'FE14 0018	H'1E14 0018	32	Pck
	Slave address enable register 0	ICSAR0	R/W	H'FE14 001C	H'1E14 001C	32	Pck
	Master address enable register 0	ICMAR0	R/W	H'FE14 0020	H'1E14 0020	32	Pck
	Receive data register 0	ICRXD0	R/W	H'FE14 0024	H'1E14 0024	32	Pck
	Transmit data register 0	ICTXD0	R/W	H'FE14 0024	H'1E14 0024	32	Pck
	FIFO control register 0	ICFCR0	R/W	H'FE14 0028	H'1E14 0028	32	Pck
	FIFO status register 0	ICFSR0	R/W	H'FE14 002C	H'1E14 002C	32	Pck
	FIFO interrupt enable register 0	ICFIER0	R/W	H'FE14 0030	H'1E14 0030	32	Pck
	Receive FIFO data count register 0	ICRFDR0	R	H'FE14 0034	H'1E14 0034	32	Pck
	Transmit FIFO data count register 0	ICTFDR0	R	H'FE14 0038	H'1E14 0038	32	Pck
1	Slave control register 1	ICSCR1	R/W	H'FE15 0000	H'1E15 0000	32	Pck
	Master control register 1	ICMCR1	R/W	H'FE15 0004	H'1E15 0004	32	Pck
	Slave status register 1	ICSSR1	R/(W)*1	H'FE15 0008	H'1E15 0008	32	Pck
	Master status register 1	ICMSR1	R/(W)*2	H'FE15 000C	H'1E15 000C	32	Pck
	Slave interrupt enable register 1	ICSIER1	R/W	H'FE15 0010	H'1E15 0010	32	Pck
	Master interrupt enable register 1	ICMIER1	R/W	H'FE15 0014	H'1E15 0014	32	Pck
	Clock control register 1	ICCCR1	R/W	H'FE15 0018	H'1E15 0018	32	Pck
	Slave address enable register 1	ICSAR1	R/W	H'FE15 001C	H'1E15 001C	32	Pck
	Master address enable register 1	ICMAR1	R/W	H'FE15 0020	H'1E15 0020	32	Pck
	Receive data register 1	ICRXD1	R/W	H'FE15 0024	H'1E15 0024	32	Pck
	Transmit data register 1	ICTXD1	R/W	H'FE15 0024	H'1E15 0024	32	Pck
	FIFO control register 1	ICFCR1	R/W	H'FE15 0028	H'1E15 0028	32	Pck
	FIFO status register 1	ICFSR1	R/W	H'FE15 002C	H'1E15 002C	32	Pck
	FIFO interrupt enable register 1	ICFIER1	R/W	H'FE15 0030	H'1E15 0030	32	Pck
	Receive FIFO data count register 1	ICRFDR1	R	H'FE15 0034	H'1E15 0034	32	Pck
	Transmit FIFO data count register 1	ICTFDR1	R	H'FE15 0038	H'1E15 0038	32	Pck

Ch.	Register Name	Abbrev.	Power-on Reset by RESET Pin/WDT/ H-UDI	Manual Reset by RESET Pin/WDT/ Multiple Exception	Sleep by Sleep Instruction/ Deep Sleep	Standby by Hardware	Standby by Software/ Each Module
0	Slave control register 0	ICSCR0	H'0000 0000	H'0000 0000	Retained	*4	Retained
	Master control register 0	ICMCR0	H'0000 0000*3	H'0000 0000*3	Retained		Retained
	Slave status register 0	ICSSR0	H'0000 0000	H'0000 0000	Retained		Retained
	Master status register 0	ICMSR0	H'0000 0000	H'0000 0000	Retained		Retained
	Slave interrupt enable register 0	ICSIER0	H'0000 0000	H'0000 0000	Retained		Retained
	Master interrupt enable register 0	ICMIER0	H'0000 0000	H'0000 0000	Retained		Retained
	Clock control register 0	ICCCR0	H'0000 0000	H'0000 0000	Retained		Retained
	Slave address enable register 0	ICSAR0	H'0000 0000	H'0000 0000	Retained		Retained
	Master address enable register 0	ICMAR0	H'0000 0000	H'0000 0000	Retained		Retained
	Receive data register 0	ICRXD0	Undefined	Undefined	Retained		Retained
	Transmit data register 0	ICTXD0	Undefined	Undefined	Retained		Retained
	FIFO control register 0	ICFCR0	H'0000 0000	H'0000 0000	Retained		Retained
	FIFO status register 0	ICFSR0	H'0000 0001	H'0000 0001	Retained		Retained
	FIFO interrupt enable register 0	ICFIER0	H'0000 0000	H'0000 0000	Retained		Retained
	Receive FIFO data count register 0	ICRFDR0	H'0000 0000	H'0000 0000	Retained		Retained
	Transmit FIFO data count register 0	ICTFDR0	H'0000 0000	H'0000 0000	Retained		Retained
1	Slave control register 1	ICSCR1	H'0000 0000	H'0000 0000	Retained		Retained
	Master control register 1	ICMCR1	H'0000 0000*3	H'0000 0000*3	Retained		Retained
	Slave status register 1	ICSSR1	H'0000 0000	H'0000 0000	Retained		Retained
	Master status register 1	ICMSR1	H'0000 0000	H'0000 0000	Retained		Retained
	Slave interrupt enable register 1	ICSIER1	H'0000 0000	H'0000 0000	Retained		Retained
	Master interrupt enable register 1	ICMIER1	H'0000 0000	H'0000 0000	Retained		Retained
	Clock control register 1	ICCCR1	H'0000 0000	H'0000 0000	Retained		Retained
	Slave address enable register 1	ICSAR1	H'0000 0000	H'0000 0000	Retained		Retained
	Master address enable register 1	ICMAR1	H'0000 0000	H'0000 0000	Retained		Retained
	Receive data register 1	ICRXD1	Undefined	Undefined	Retained		Retained
	Transmit data register 1	ICTXD1	Undefined	Undefined	Retained		Retained
	FIFO control register 1	ICFCR1	H'0000 0000	H'0000 0000	Retained		Retained
	FIFO status register 1	ICFSR1	H'0000 0001	H'0000 0001	Retained		Retained
	FIFO interrupt enable register 1	ICFIER1	H'0000 0000	H'0000 0000	Retained		Retained
	Receive FIFO data count register 1	ICRFDR1	H'0000 0000	H'0000 0000	Retained		Retained
	Transmit FIFO data count register 1	ICTFDR1	H'0000 0000	H'0000 0000	Retained		Retained

Notes: 1. Bits 4 to 0 allow only a write of 0 to clear the flags.
2. Bits 6 to 0 allow only a write of 0 to clear the flags.
3. Bits 6 and 5 are undefined.
4. After exiting hardware standby mode, this LSI enters the power-on reset state by the RESET pin.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
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
	-	-	-	-	-	-	-	-	-	-	-	-	SDBS	SIE	GCAE	FNA
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/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 4	—	All 0	R	Reserved These bits are always read as 0, and the write value should always be 0.
3	SDBS	0	R/W	Slave Data Buffer Select This bit is used to select the data buffer. The data buffer has two modes; the FIFO buffer mode and the single buffer mode. Clearing SDBS to 0 will select the FIFO buffer mode. In the receive mode, while the RDF flag is 1 with the receive byte count stored in the FIFO buffer equal to or greater than the byte count specified by RTRG3 to RTRG0, SCL is held low. Reading the receive data from the FIFO buffer will clear the RDF flag to 0 and release SCL from low level. Setting SDBS to 1 will select the single buffer mode. SCL will be held low from the moment the receive data register receives a data packet until SDR is cleared to 0. 0: FIFO buffer mode 1: Single buffer mode
2	SIE	0	R/W	Slave Interface Enable Ensure to set this bit to 1 to have the slave to operate. If this bit is low, the slave interface is reset.
1	GCAE	0	R/W	General Call Acknowledgement Enable Ensure to set this bit to 1 when the master requires the slave to acknowledge a transmission of a general call address.

0	FNA	0	R/W	Force Non-Acknowledge In the slave receiver mode, the level on the FNA bit is sent to the transmitter as the acknowledge signal. FNA is 0 while the data packet is being received, and set to 1 on completion of data reception. The force non-acknowledge is sent to the master during slave reception. After having received the last required byte in a data packet, the slave communicates to the master with not driving acknowledge (NACK). The master issues a stop on to the bus after receiving a NACK. Setting FNA to 1 will not effect the acknowledging of slave addresses. This I^2C module assumes that the slave receiver returns NACK by FNA = 1 in order to handle an error associated with system failure. Therefore, the low period of SCL is not extended for returning NACK. The FNA value on reception of the last bit of one byte determines whether or not the slave receiver returns NACK. Namely, to have the slave return NACK after transfer of a specific byte, FNA must be set to 1 on completion of the transfer of the preceding byte.

19.3.2 Slave Status Register (ICSSR)

The status bits (bit 0 to bit 6) of the slave status register are cleared by writing 0 to the respective status bit positions in the receive status state. Each bit is held at 1 until reset by a write of 0; however, the GCAR and STM bits are the exception.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
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
	-	-	-	-	-	-	-	-	-	GCAR	STM	SSR	SDE	SDT	SDR	SAR
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/W*	R/W*	R/W*	R/W*

31 to 7	—	All 0	R	Reserved These bits are always read as 0, and the write value should always be 0.
6	GCAR	0	R	General Call Address Received Indicates that the address received from the bus is a general call address (00H). This bit does not cause an interrupt. This bit is automatically cleared to 0 by hardware when the SIE bit (bit 2 in the slave control register) is 0 or when the SSR bit (bit 4 in the slave status register) is set to 1.
5	STM	0	R	Slave Transmit Mode Current slave transmit mode (read or write). Setting this bit to 1 indicates a write operation, and clearing this bit to 0 indicates a read. This status bit does not cause an interrupt. This bit is automatically cleared by hardware when the SIE bit (bit 2 in the slave control register) is 0 or when the SSR bit (bit 4 in the slave status register) is set to 1.
4	SSR	0	R/W*	Slave Stop Received A stop has been output to the bus. This status bit becomes active after the rising edge of SDA during the stop bit.
3	SDE	0	R/W*	Slave Data Empty Transmit data has been loaded into the shift register. At the start of data byte transmission, the contents of the ICTXD register are loaded into a shift register ready for the data to be passed onto the bus. This status bit indicates that this has taken place and that the ICTXD register is again ready to receive further data. This status bit becomes active on the falling edge of SCL before the first data bit. In the single buffer mode, this bit must be reset each time a new data has been written to the ICTXD register. This is because the slave holds SCL low to stall the bus, if it reaches the start of a slave transmit cycle and this status bit is still set. In the FIFO buffer mode, this bit is not used.

2	SDT	0	R/W*	Slave Data Transmitted A byte of data has been transmitted on the bus. This status bit becomes active after the falling edge of SCL during the last data bit.
1	SDR	0	R/W*	Slave Data Received A byte of data has been received from the bus and is available in the receive data register. This bit becomes active after the falling edge of SCL during the last data bit. After data has been read from the ICRXD register, this bit must be reset in single buffer mode. This bit is not used in the FIFO buffer mode. When SDBS is set to 1, SCL will be held low from the moment the receive data register acquires the data packet up until SDR is cleared.
0	SAR	0	R/W*	Slave Address Received Indicates that the slave has recognized its own address on the bus (defined by the contents of the slave address register). If the general call acknowledgement enable bit in the slave control register is enabled, then this status bit could also indicate the reception of a general call address on the bus. In that case, bit GCAR of this register is used to differentiate the receipt of a general call address. Bit STM indicates whether the access is a read (high) or a write (low). This status becomes active after the falling edge of SCL during the last address bit. The slave holds SCL low at the start of the ACK phase until this bit is reset by software.

Note: * This bit can be written or read. When 0 is written to, the bit is initialized. When 1 is written to, it is ignored.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
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
	-	-	-	-	-	-	-	-	-	-	-	SSRE	SDEE	SDTE	SDRE	SARE
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/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 5	—	All 0	R	Reserved These bits are always read as 0, and the write value should always be 0.
4	SSRE	0	R/W	Slave Stop Receive Interrupt Enable 0: The SSR interrupt is disabled 1: The SSR interrupt is enabled
3	SDEE	0	R/W	Slave Data Empty Interrupt Enable 0: The SDE interrupt is disabled 1: The SDE interrupt is enabled
2	SDTE	0	R/W	Slave Data Transmit Interrupt Enable 0: The SDT interrupt is disabled 1: The SDT interrupt is enabled
1	SDRE	0	R/W	Slave Data Receive Interrupt Enable 0: The SDR interrupt is disabled 1: The SDR interrupt is enabled
0	SARE	0	R/W	Slave Address Receive Interrupt Enable 0: The SAR interrupt is disabled 1: The SAR interrupt is enabled

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
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
	-	-	-	-	-	-	-	-	-	SADD0_6	SADD0_5	SADD0_4	SADD0_3	SADD0_2	SADD0_1	SADD0_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/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 7	—	All 0	R	Reserved These bits are always read as 0, and the write value should always be 0.
6 to 0	SADD0_6 to SADD0_0	All 0	R/W	Slave Address This is the unique 7-bit address allocated to the slave on the I^2C bus. The slave interface looks for a match between this address and the first seven bits transmitted as the slave address, at the beginning of a data packet transmission.

19.3.5 Master Control Register (ICMCR)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
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
	-	-	-	-	-	-	-	-	MDBS	FSCL	FSDA	OBPC	MIE	TSBE	FSB	ESG
Initial value:	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

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	All 0	R	Reserved These bits are always read as 0, and the write value should always be 0.

7	MDBS	0	R/W	Master Data Buffer Select This bit is used to select the data buffer. The data buffer has the FIFO buffer mode and the single buffer mode. Clearing MDBS to 0 will select theFIFO buffer mode. In the receive mode, while the RDF flag is 1 with the receive byte count in the FIFO buffer reaches the byte count specified by RTRG3 to RTRG0, SCL is held low. Reading the receive data from the FIFO buffer will clear the RDF flag to 0 and release SCL from low level. Setting MDBS to 1 will select the single buffer mode. SCL will be held low from the moment the receive data register receives a data packet until the MDR flag is cleared to 0.
6	FSCL	—	R/W	Force SCL Controls the state of the I2C_SCL pin. Reading this bit will return the value reflecting the current state of I2C_SCL. When OBPC is 1, this bit directly controls SCL on the bus. Since this bit reflects the value on the I2C_SCL pin directly, the read value (level) of this bit (including the reset level) changes depending on the I2C_SCL level.
5	FSDA	—	R/W	Force SDA Controls the state of the I2C_SDA pin. Reading this bit will return the value reflecting the busy state on the I^2C bus. When OBPC is 1, this bit directly controls SDA on the bus. The read value (level) of this bit (including the reset level) reflects the busy state on the I^2C bus. 0: The I^2C bus is not busy 1: The I^2C bus is busy
4	OBPC	0	R/W	Override Bus Pin Control Setting OBPC to 1 will have FSDA and FSCL in this register control the SDA and SCL lines directly. This mode is used for testing purposes only.
3	MIE	0	R/W	Master Interface Enable Setting MIE to 1 will enable the master interface.

2	TSBE	0	R/W	Start Byte Transmission Enable Setting TSBE to 1 will have the master transmit a start byte (01H) onto the bus after each start or restart is issued. The start byte is used for interfacing with a microcomputer with a lower operating frequency which supports to the I^2C bus interface.
1	FSB	0	R/W	Force Stop onto the Bus Setting FSB to 1 will have the master issue a stop onto the bus at the end of the current transfer. If ESG is also 1, the master immediately issues a start and begins transmitting a new data packet. If ESG is 0, the master enters the idle state. Set FSB to 1 when the TEND flag is set to 1 during transmission in the FIFO buffer mode, or when the RDF flag is set to 1 during reception in the FIFO buffer mode. In single buffer mode, when the last bit of a byte is transmitted/received, the I^2C module latches the FSB value and enters the STOP state. Therefore, to stop the transfer after a specified number of bytes are transferred, the FSB bit must be set to 1 before the last byte is transferred. Note: Check section 19.7, Usage Notes, when using this bit.
0	ESG	0	R/W	Enable Start Generation Setting ESG to 1 will have the master start transmission of a data packet. If the bus is idle when ESG is set to 1, the master issues a start onto the bus and then issues the slave address. If the master is transferring data when ESG is set to 1, the master issues a restart before transmitting the slave address at the end of that data byte transfer. In the case of data packet transmission, ESG must be reset by software after the slave address is transmitted; if not reset, a restart is issued after each time the transmission is completed.

The status bits of the master status register (bits 0 to 6) are cleared by writing 0 to the respective status bit positions in the reception states. Each status bit remains 1 until reset by writing 0.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
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
	-	-	-	-	-	-	-	-	-	MNR	MAL	MST	MDE	MDT	MDR	MAT
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/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*

Bit	Bit Name	Initial Value	R/W	Description
31 to 7	—	All 0	R	Reserved These bits are always read as 0, and the write value should always be 0.
6	MNR	0	R/W*	Master NACK Received MNR = 1 indicates that the master has received a NACK response (SDA is high during the acknowledge cycle on the bus) during an address or data transmission.
5	MAL	0	R/W*	Master Arbitration Lost MAL = 1 in the multiple-master system indicates that the master has lost bus arbitration for other masters. In this case, MIE is reset and master interface is disabled.
4	MST	0	R/W*	Master Stop Transmission MST = 1 indicates that the master has sent a stop onto the bus. A stop can be sent either as a result of the setting of the forced stop bit in the control register, or from a NACK being received from a slave during data packet reception from a slave.

3	MDE	0	R/W*	Master Data Empty At the start of a byte-data transmission, the contents of the transmit data register are loaded to a shift register, which is ready for passing data onto the bus. MDE = 1 indicates that this load operation has taken place and the transmit data register is ready to receive further data. In the master transmit mode, the MDE and MAT bits are simultaneously set to 1 after the slave address has been sent. In this case, clear the MDE and MAT bits to 0 after clearing the ESG bit of the ICMCR to 0. The data transmission is then resumed.
2	MDT	0	R/W*	Master Data Transmission The master has transmitted a byte of data to the slave on the bus. This status bit becomes 1 after the falling edge of SCL during the last data bit transmission.
1	MDR	0	R/W*	Master Data Reception The master has received a byte of data from the bus and the receive data register is ready. This status bit becomes active after the falling edge of SCL during the last data bit reception. In the single buffer mode, this status bit must be reset after data has been read from the receive data register. In the FIFO buffer mode, this bit is not used. When MDBS = 1, SCL is held low from the moment the receive data register starts receiving data packets until the MDR bit is cleared to 0. In the master receive mode, the MDE and MAT bits are simultaneously set to 1 after the slave address has been sent. In this case, clear the MDE and MAT bits to 0 after clearing the ESG bit of the ICMCR to 0. The data transmission is then resumed.
0	MAT	0	R/W*	Master Address Transmission The master has transmitted the slave address byte of a data packet. This bit becomes 1 after the falling edge of SCL during the output of the ack bit which is sent after an address.

Note: * This bit can be written or read. When 0 is written to, the bit is initialized. When 1 is written to, it is ignored.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
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
	-	-	-	-	-	-	-	-	-	MNRE	MALE	MSTE	MDEE	MDTE	MDRE	MATE
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/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 7	—	All 0	R	Reserved These bits are always read as 0, and the write value should always be 0.
6	MNRE	0	R/W	Master NACK Received Interrupt Enable 0: The MNR interrupt is disabled 1: The MNR interrupt is enabled
5	MALE	0	R/W	Master Arbitration Lost Interrupt Enable 0: The MAL interrupt is disabled 1: The MAL interrupt is enabled
4	MSTE	0	R/W	Master Stop Transmission Interrupt Enable 0: The MST interrupt is disabled 1: The MST interrupt is enabled
3	MDEE	0	R/W	Master Data Empty Interrupt Enable 0: The MDE interrupt is disabled 1: The MDE interrupt is enabled
2	MDTE	0	R/W	Master Data Transmission Interrupt Enable 0: The MDT interrupt is disabled 1: The MDT interrupt is enabled
1	MDRE	0	R/W	Master Data Receive Interrupt Enable 0: The MDR interrupt is disabled 1: The MDR interrupt is enabled
0	MATE	0	R/W	Master Address Transmission Interrupt Enable 0: The MAT interrupt is disabled 1: The MAT interrupt is enabled

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
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
	-	-	-	-	-	-	-	-	SADD1_6	SADD1_5	SADD1_4	SADD1_3	SADD1_2	SADD1_1	SADD1_0	STM1
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

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	All 0	R	Reserved These bits are always read as 0, and the write value should always be 0.
7 to 1	SADD1_6 to SADD1_0	All 0	R/W	Slave Address This is the address of the slave the master is to communicate with.
0	STM1	0	R/W	Slave Transfer Mode This bit indicates in which mode the slave is to operate. This bit sets the operating mode (transmit or receive mode) of the slave to the external slave device specified by the slave address (SADD1) sent from the master. The slave device is automatically set to the transmit/receive mode by hardware according to the received STM1 value. 0: Write operation 1: Read operation

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
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
	-	-	-	-	-	-	-	-	SCGD						CDF	
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

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	All 0	R	Reserved These bits are always read as 0, and the write value should always be 0.
7 to 2	SCGD	All 0	R/W	SCL Clock Generation Divider In master mode operation, the SCL clock is generated from the internal clock frequency using the SCGD value as the division ratio. In slave mode operation, if SCL is held low to stall the bus by data overflow, this clock is also generated from the internal clock. Accordingly, SCGD must be programmed for master and slave operating modes. The formula expressing the relationship is: Equation 2 SCL rate calculation SCL freq = IICck / (20 + SCGD * 8) Recommended settings for CDF and SCGD for various CPU rates and the two I^2C bus speeds are given in table 19.3.
1, 0	CDF	All 0	R/W	Clock Division Factor The internal clocks for most of the blocks in the I^2C bus interface module are divided from peripheral bus clock. The internal I^2C clock is generated from the peripheral clock using the value of CDF as the division ratio: Equation 1 I^2C internal clock frequency calculation IICck = Pck / (1 + CDF) The minimum setup and hold times on the SMA line relative to the SCL line on the bus should be met. The clock frequency is to ensure that the glitch filtering will operate with glitches of up to 50 ns (as described in the fast mode I^2C specifications). Note: CDF must be set to a value that the clock frequency (IICck) is less than 20 MHz.

	100 kHz		400 kHz	
Pck	**CDF**	**SCGD**	**CDF**	**SCGD**
33 MHz	3	8	2	1
Error	–1.79%		–1.79%	
25 MHz	2	8	0	6
Error	0.97%		–8.08824%	

19.3.10 Receive/Transmit Data Registers (ICRXD/ICTXD)

Reading from or writing to this register means accessing different physical internal registers. When data is to be transmitted, data in the shift register is loaded to TXD. After data has been received into the shift register from the I^2C bus, data is loaded to RXD.

- Receive Data Register (ICRXD) (Single Buffer Mode)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
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
	-	-	-	-	-	-	-	-	RXD							
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

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	All 0	R	Reserved These bits are always read as 0, and the write value should always be 0.
7 to 0	RXD	All 0	R/W	Read Receive Data Data received by master or slave.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
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
	-	-	-	-	-	-	-	-	TXD							
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

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	All 0	R	Reserved These bits are always read as 0, and the write value should always be 0.
7 to 0	TXD	All 0	R/W	Write Transmit Data Data transmitted by master or slave.

When the FIFO buffer is selected by the SDBS bit of ICSCR or the MDBS bit of ICMCR, the operation is as follows:

- **Receive Data Register (ICRXD) (FIFO Buffer Mode)**

  ICRXD is a 16-stage FIFO register for storing the receive data. When 1-byte data is received, the receive data is transferred to ICRXD from the shift register, and reception ends. After that, the ICRXD is ready to receive and consecutive receive operations of up to 16 bytes of data are possible. When 16 bytes of receive data are stored, the FIFO receive register is full.

  ICRXD is read only and cannot be written to by the CPU. When the receive FIFO register is completely empty, reading ICRXD will return an undefined value. When the receive FIFO register is full, the subsequent data is lost.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
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
	-	-	-	-	-	-	-	-	RXD							
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

ICTXD is a 16-stage FIFO register for storing the transmit data. When the data is written to the ICTXD and the shift register is empty, the data is transferred from ICTXD to the shift register and transmission starts.

ICTXD is write only and cannot be read from by the CPU. When the register is full with 16 bytes of data, the subsequent data cannot be written to, and the written value is ignored.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
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
	-	-	-	-	-	-	-	-	TXD							
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W:	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

19.3.11 FIFO Control Register (ICFCR)

ICFCR is a register for resetting the byte count and setting the number of trigger data in transmit and receive FIFO registers, respectively. ICFCR can always be read from and written to by the CPU.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
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
	-	-	-	-	-	-	-	-	RTRG3	RTRG2	RTRG1	RTRG0	TTRG1	TTRG0	RFRST	TFRST
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

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	All 0	R	Reserved. These bits are always read as 0. The write value should always be 0.

7	RTRG3	0	R/W	Receive FIFO Data Count Trigger
6	RTRG2	0	R/W	These bits specify the receive byte count to set ICFSR.RDF. The RDF flag is set to 1 when the receive byte count in ICRXD reaches the trigger byte count set in these bits. The following settings are available: 0000: 1 0110: 7 1100: 13 0001: 2 0111: 8 1101: 14 0010: 3 1000: 9 1110: 15 0011: 4 1001: 10 1111: 16 0100: 5 1010: 11 0101: 6 1011: 12
5	RTRG1	0	R/W	
4	RTRG0	0	R/W	
3	TTRG1	0	R/W	Transmit FIFO Data Count Trigger
2	TTRG0	0	R/W	These bits specify the receive byte count to set ICFSR. TDFE. The RDF flag is set to 1 when the receive byte count in ICTXD is equal to or smaller than the trigger byte count set in these bits. The following settings are available: 00: 8 (8)*1 01: 4 (12) 10: 2 (14) 11: 0 (16)
1	RFRST	0	R/W	Receive FIFO Data Register Reset Disables the receive data in ICRXD and resets ICRXD to the empty state. 0: The reset operation*2 is disabled 1: The reset operation is enabled
0	TFRST	0	R/W	Transmit FIFO Data Register Reset Disables the receive data in ICTXD and resets ICTXD to the empty state. 0: The reset operation*2 is disabled 1: The reset operation is enabled

Notes: 1. Values in parentheses ( ) indicate the number of empty stages in the transmit FIFO data register (ICTXD) in each case.

2. At power-on reset and manual reset, the reset operation is performed.

ICFSR is a 32-bit register that indicates the operation state of the I^{2}C. ICFSR can always be written to and read from the CPU. However, writing 1 to the TEND, RDF and TDFE flags is not allowed. It is possible to write 0 to clear the flags.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
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
	-	-	-	-	-	-	-	-	-	-	-	-	-	TEND	RDF	TDFE
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 3	—	All 0	R	Reserved. These bits are always read as 0, and the write value should always be 0.
2	TEND	0	R/W	Transmit End Indicates the end of transmission with no valid data in ICTXD when the last bit of the character is transmitted. 0: Indicates transmission in progress [Clear conditions] • Power-on reset, or manual reset • When transmit data is written to ICTXD and 0 to the TEND flag. 1: Indicates transmission end [Set condition] • When there is no transmit data in ICTXD when the last bit of the 1-byte character is transmitted.

1	RDF	0	R/W	Receive FIFO Data Full Indicates that the receive data is transferred from the shift register to ICRXD, and the byte count in ICRXD reaches the receive trigger byte count set in the RTRG3 to RTRG0 bits in ICFCR. When RDF is set to 1, the receive operation stops. Reading the receive data from ICRXD and clearing RDF to 0 will resume a receive operation. 0: Indicates that the byte count in ICRXD is smaller than the receive trigger byte count. [Clear conditions] • Power-on reset, or manual reset • Reading from ICRXD has made the byte count in ICRXD smaller than the receive trigger byte count, and 0 is written to RDF. 1: Indicates that the byte count in ICRXD reaches the receive trigger byte count. [Set condition] • The byte count in ICRXD exceeds the receive trigger byte count.*1
0	TDFE	1	R/W	Transmit FIFO Data Empty Indicates that the transmit data can be written to ICTXD, the data is transferred from ICTXD to the shift register, and the byte count in ICTXD is equal to or smaller than the trigger byte count set in the TTRG1 and TTRG0 bits of ICFCR. 0: Indicates that the byte count in ICTXD exceeds the transmit trigger byte count. [Clear condition] • When the byte count in ICTXD exceeds the transmit trigger byte count, and 0 is written to the TDFE bit. 1: Indicates that the byte count in ICTXD is equivalent to or smaller than the transmit trigger byte count. [Set conditions] • Power-on reset, or manual reset • The byte count in ICTXD becomes equivalent to or smaller than the transmit trigger byte count.*2

The receive byte count in ICRXD is indicated by ICRFDR.

2. ICTXD is a 16-byte FIFO register. When TDFE = 1, the maximum byte count that can be written to ICTXD is 16 minus the transmit trigger byte count. If the written byte count exceeds the maximum, the excess data is ignored. The byte count in ICTXD is indicated by ICTFDR.

19.3.13 FIFO Interrupt Enable Register (ICFIER)

ICFIER is a register that enables or disables interrupt request from the FIFO operation source. ICFIER can always be read from and written to by the CPU.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
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
	-	-	-	-	-	-	-	-	-	-	-	-	-	TEIE	RXIE	TXIE
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/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 3	—	All 0	R	Reserved. These bits are always read as 0, and the write value should always be 0.
2	TEIE	0	R/W	Transmit End Interrupt Enable Enables or disables the generation of the transmit end interrupt (TEI) when the TEND flag of ICFSR is set to 1. 0: The transmit end interrupt (TEI) is disabled 1: The transmit end interrupt (TEI) is enabled
1	RXIE	0	R/W	Receive Interrupt Enable Enables or disables the generation of the receive data full interrupt (RXI) when the RDF flag of ICFSR is set to 1. 0: The receive data full interrupt (RXI) is disabled 1: The receive data full interrupt (RXI) is enabled

0	TXIE	0	R/W	Transmit Interrupt Enable Enables or disables the generation of the transmit FIFO data empty interrupt (TXI) when serial transmit data is transferred from ICTXD to the shift register, the byte count in ICTXD is equal to or smaller than the transmit trigger byte count, and the TDFE flag of ICFSR is set to 1. 0: The transmit FIFO data empty interrupt (TXI) is disabled 1: The transmit FIFO data empty interrupt (TXI) is enabled

19.3.14 Receive FIFO Data Count Register (ICRFDR)

ICRFDR is a 32-bit register that indicates the byte count in ICRXD. The lower 5 bits indicate the number of receive bytes in ICRXD. ICRFDR can always be read by the CPU. H'0000 0000 indicates that ICRXD contains no receive data, while H'0000 0010 indicates that it holds 16 bytes of receive data.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
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
	-	-	-	-	-	-	-	-	-	-	-	R4	R3	R2	R1	R0
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

19.3.15 Transmit FIFO Data Count Register (ICTFDR)

ICTFDR is a 32-bit register that indicates the byte count stored in ICTXD. The lower 5 bits indicates the number of transmit bytes in ICTXD. ICTFDR can always be read by the CPU. H'0000 0000 indicates that ICTXD contains no transmit data, while H'0000 0010 indicates that it holds 16 bytes of transmit data.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
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
	-	-	-	-	-	-	-	-	-	-	-	T4	T3	T2	T1	T0
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

19.4.1 Data and Clock Filters

These blocks filter out glitches on signals coming from the I^2C bus. A glitch with a duration of up to one clock cycle is rejected. (See the description of the clock control register for details of internal clock). This is specified for the faster I^2C bus rate (400 kHz) but does not violate the slower I^2C bus rate specification.

These blocks also perform resynchronization of bus signals to the internal clock.

19.4.2 Clock Generator

The clock generator has two functions. Firstly, it generates the SCL I^2C bus clock under command of the master or slave interface. Secondly, it controls the internal clock rate, used by filtering blocks and the master and slave interfaces. This operates as a clock-enable signal to the registers in the filter, master, and slave interfaces.

19.4.3 Master and Slave Interfaces

Master and slave interfaces run independently and in parallel. The master interface controls the transmission of address and data on the I^2C bus. The slave interface monitors the I^2C bus and takes part in transmissions if its programmed address is seen on the bus. Both interfaces communicate with the control register and the status registers independently. Only one interrupt line comes from the I^2C bus interface module; the source could be either the master or the slave.

19.4.4 Software Status Interlocking

To make software interface to the I^2C bus interface module as robust and simple as possible, some statuses are interlocked in the master and slave interface operations. The status bits involved are described below.

(1) MDR and SDR (Single Buffer Mode)

MDR and SDR are set to 1 when data is received. Clear these bits to 0 after reading the receive data register. If data is received while MDR and SDR are 1, hardware recognizes that unread data remains in the receive data register, automatically holds SCL at low level and suspends data transfer. In this case, clearing these bits to 0 after reading the receive data will resume transfer.

When receiving data consecutively, be sure to clear MDR and SDR to 0 after reading the receive data register.

When the slave or master is about to start transmission of data (from the transmit data register) onto the I^2C bus, the MDE and SDE status bits may still remain 1. In such case, the SCL line must be held low until these bits are reset to 0. The MDE and SDE bits being set to 1 indicate that the data has already been transmitted from the transmit data register onto the I^2C bus.

To write data into the transmit data register that is ready for the next transmission, the software must clear MDE and SDE to 0. However, this is not required for the first byte transmission onto the bus.

(3) MAL

When the master has lost arbitration, the MAL bit in the master status register is set to 1 and the MIE bit in the master control register is reset to 0. At this point, the master mode is disabled and the I^2C bus interface is set to operate in the slave mode. When master operation is restarted, data transfer from the master begins after the MAL bit has been cleared to 0.

(4) SAR

The SAR status bit is set to 1 when the slave has recognized its address output to the I^2C bus. At this point, the slave interface drives the SCL line to be low until the SAR status bit is reset to 0.

This is particularly important when a slave transmit is about to take place on the bus. When the slave transmits the data from the transmit data register, the software responds to the SAR status by writing the required data into the transmit data register and resetting the SAR status bit to 0. This allows the slave interface to carry on access.

When the slave is about to receive data, the software may not have completed reading of data loaded by the previous access from the receive data register. The problem is that the new access may overwrite the valid data still held in the receive data register. However, this can be avoided by using the SAR status bit. The software should reset the SAR bit to 0 (if it is set to 1) only after completing a read from the receive data register. Then the receive data register will not be overwritten.

(5) Writing to ICTXD and the TDFE Flag (FIFO Buffer Mode)

The TDFE flag of ICFSR is set to 1 when the byte count in ICTXD is equal to or smaller than the transmit trigger byte count by TTRG1 and TTRG0 bits of ICFCR. After TDFE is set, the transmit data can be written for the number of empty bytes in ICTXD. This allows efficient continuous transmission. When the byte count in ICTXD is below the transmit trigger count, the TDFE flag is automatically set to 1 even if the flag is cleared to 0. Accordingly, clear the TDFE flag to 0 only

ICTADR can be informed by ICTFDR.

(6) Reading from ICRXD and the RDF Flag (FIFO Buffer Mode)

The RDF flag of ICFSR is set to 1 when the receive byte count in ICRXD reaches the receive trigger byte count by RTRG3 to RTRG0 bits of ICFCR. After RDF is set, the receive data for the trigger byte count can be read from ICRXD. This allows efficient continuous reception.

When the byte count in ICRXD is equivalent to or greater than the trigger count after a read, the RDF flag is cleared to 0 even if it is set to 1 again. Accordingly, read the RDF flag as 1 and then clear it to 0 after reading all data. The receive byte count in ICRXD can be informed by ICRFDR.

19.4.5 I²C Bus Data Format

Figure 19.2 shows the bus timing of the I²C bus interface. Table 19.4 describes legend in figure 19.2.

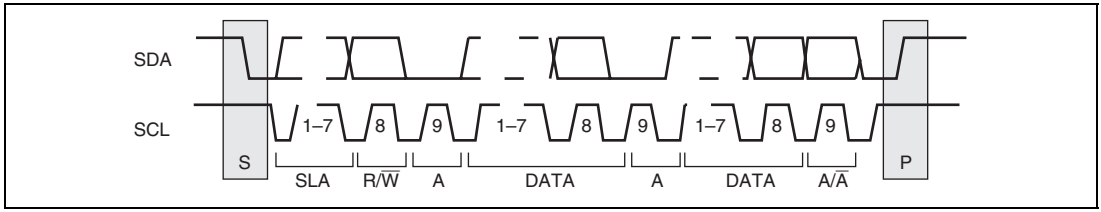


Figure 19.2 I²C Bus Timing

Table 19.4 Legend in I²C Bus Data Format

Symbol	Description
S	Start condition. The master device drives SDA from high to low level while SCL is high level.
SLA	Slave address. The slave address is selected by the master device.
R/$\overline{W}$	Indicates the direction of data transfer: from the slave device to the master device when R/$\overline{W}$ is 1, or from the master device to the slave device when R/$\overline{W}$ is 0.
A	Data acknowledge. Data receive device drives SDA to low level. The slave device returns a data acknowledge signal in master transmit mode.
DATA	Transfer data. The data consists of 8 bits, which are transferred from MSB.
P	Stop condition. The master device drives SDA from low to high level while SCL is high level.

Figure 19.3 shows the format of data transfer from the master to the slave device (master data transmit format). Figure 19.4 shows the data transfer format (master data receive format) in which the master device read data on and after the second byte from the slave device.

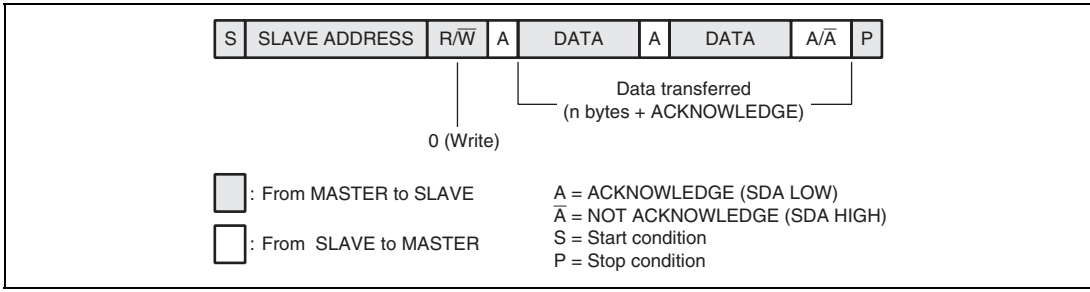


Figure 19.3 Master Data Transmit format

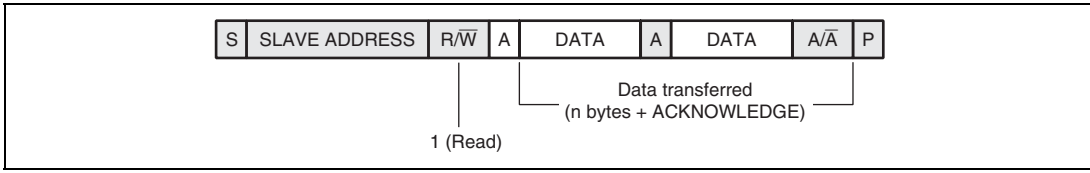


Figure 19.4 Master Data Receive format

Figure 19.5 shows the combination transfer format in which the data transfer direction changes during one transfer.

When changing the direction at the first transfer, retransmit command (Sr), the slave address and the R/$\overline{W}$ signal are transmitted. In this case, the R/$\overline{W}$ signal is set to the direction opposite to the first transfer direction.

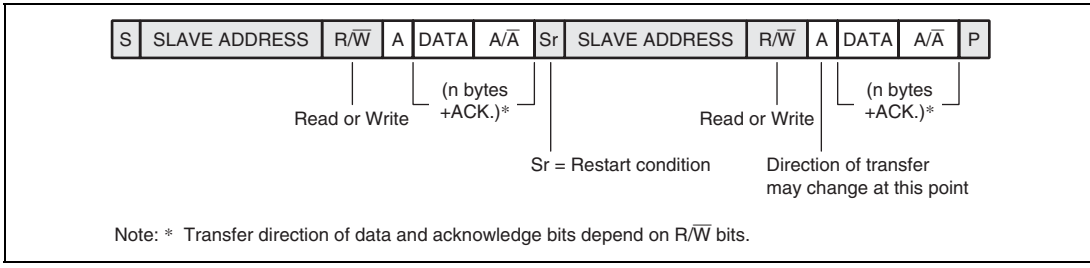


Figure 19.5 Combination Transfer Format of Master Transfer

Description is given below on the 10-bit address transfer format supported in the master mode.

This format has three transfer methods as the 7-bit address transfer format.

Figure 19.6 shows the data transfer format. The set value of the master address register is output in one byte following the first transfer condition (S). The value set in transmit data register (ICTXD) is transmitted as a slave address in the second byte. Data transfer on and after the third byte is done in the same way as the 7-bit address data transmission.

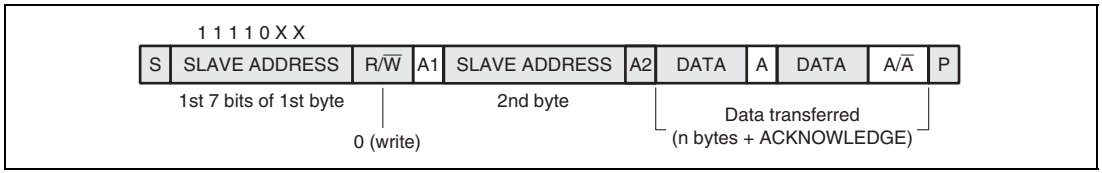


Figure 19.6 10-Bit Address Data Transfer Format

Figure 19.7 show the data receive format. Address transmission of two bytes in the data receive format is done in the same way as in the data transfer format. Then, retransmit condition (Sr) is transmitted and the value set in the master address register is output. At this time, STM1 must be set to 1 (receive mode). Data transfer is done in the same way as in the 7-bit address data receive format.

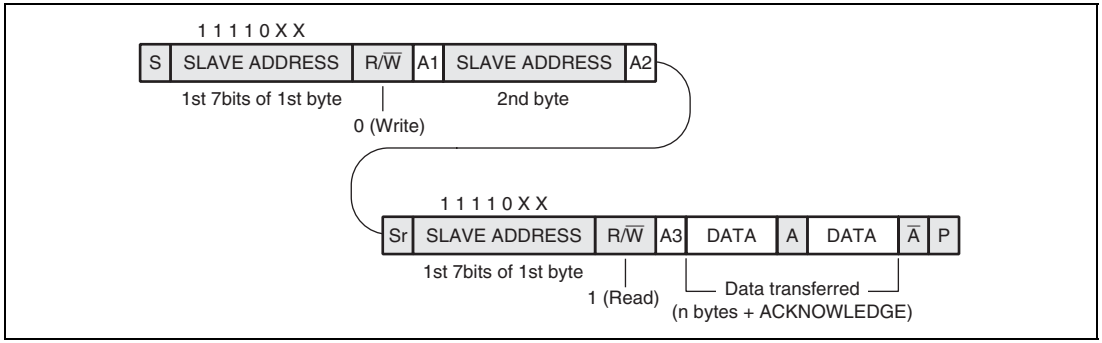


Figure 19.7 10-Bit Address Data Receive Format

In the data transmit/receive combination format, data is transmitted after an address is transmitted with the first two bytes. Then, retransmit condition (Sr) is transmitted instead of stop condition (P). After Sr is transmitted, the procedure is the same as that in the data receive format.

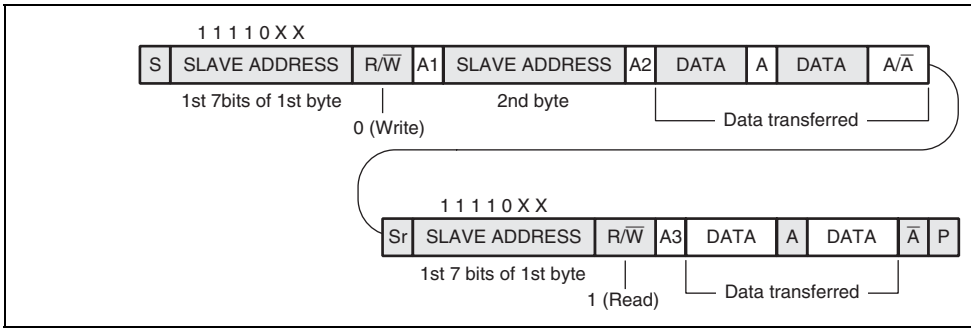


Figure 19.8 10-Bit Address Transmit/Receive Combination Format

19.4.8 Master Transmit Operation (Single Buffer Mode)

This section describes the transmit procedure and operations in master transmit mode. Figures 19.9 to 19.11 are the timing charts in master transmit mode. Setting the MDBS bit to 1 in the master control register has the I^2C module enter single buffer mode.

1. For initial setting, set clock control bits in the clock control register and interrupt generation bits in the master interrupt enable register, according to the slave address, transmit data, and the transmit speed. Since the slave mode is also required even when only the master mode is used, set the device address to the slave address register.

   Do not modify either the master control register MDBS bit or the slave control register SDBS bit during operation. Incorrect operation may occur if these bits are changed during operation.
2. Monitor the FSDA bit in the master control register. Confirm that the bit is low, which means that the other I^2C device is not using the bus. After confirmation, set the MIE bit and ESG bit in the master control register to 1 to start master transmission.
3. After the signals for indicating the transmit start condition, the slave address, and the data transfer direction are transmitted, an interrupt indicated by the MDE bit and the MAT bit in the master status register is generated in the timing of (1) in figure 19.9. At this time, clear the ESG bit to 0. The master device holds SCL low to suspend data transmission until the MDE bit is cleared to 0.
4. Interrupt indicated by the SAR is generated in the timing of (3) in figure 19.9. If the IRQ processing of the slave device is delayed, the slave device extends the SCL period to suspend data transmission (in the timing of (7) in figure 19.9). The slave device drives SDA low at the ninth clock and returns ACK.

An interrupt by MDE (bit 5) is generated at the ninth clock before starting data transfer (in the timing of (2) in figure 19.9). An interrupt by MDT (bit 2) is generated at the eighth clock after 1-byte data transfer (in the timing of (4) in figure 19.9). Clear MDE to 0 after setting transmit data. An interrupt by the SDR bit (slave data reception) is generated at the eighth clock (in the timing of (6) in figure 19.9). Clear the SDR bit to 0 after the slave device reads the receive data. If this processing is delayed, the slave device extends the SCL period to suspend data transmission (in the timing of (8) in figure 19.9).

6. To end data transfer, an interrupt by the MNR (bit 6) in the master status register is generated at the ninth clock while the ACK signal from the slave device is 1 (NACK) (in the timing of (5) in figure 19.9). The master device outputs a data transfer end condition when receiving the NACK.
   When the master device completes the data transmission, set FSB (bit 1) to 1 in the master control register to output a force stop condition. When the last bit of a byte is transmitted/received, the I^2C module latches the FSB value and enters the stop state. Therefore, to stop the transfer after a specified byte is transferred, the FSB bit must be set to 1 before the last byte data is transferred.
7. The FSB bit must be set to 1 before the last byte is transferred. In master transmit mode, after the last byte is set, the MST (master stop transmission) bit is checked using an interrupt or polling. At this time, the MNR (master NACK reception) is checked. When the NACK is returned, an error routine is executed to re-transmit the last byte data.

The timings of (1) to (6) in figure 19.9 are generated after the falling edge of the clock signal.

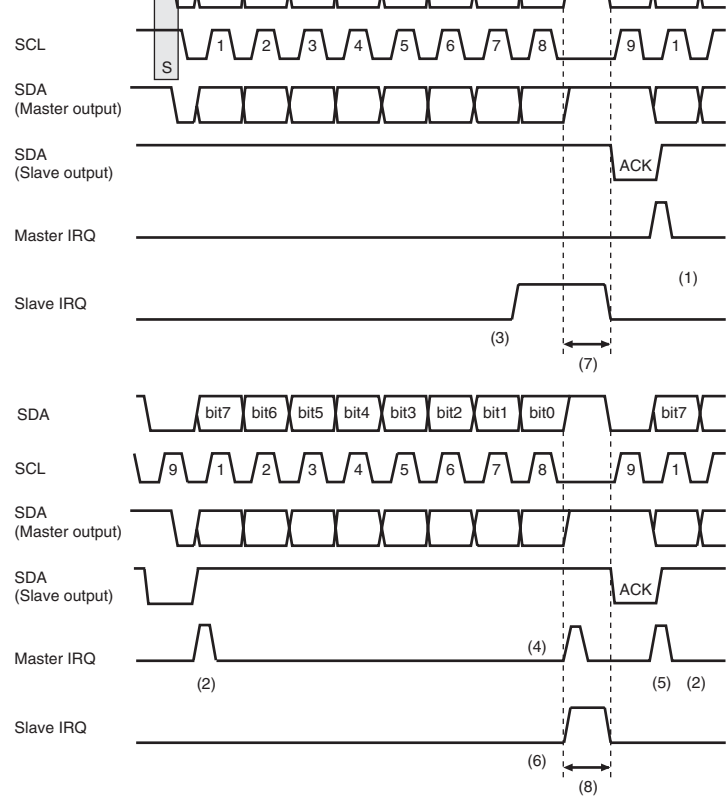


Figure 19.9 Data Transfer Mode Timing Chart

19.4.9 Master Receiver Operation (Single Buffer Mode)

Data receive procedure and operation in master receiver mode is described below. Figure 19.10 shows the operation timing in master data receive mode. Setting the MDBS bit in the master control register makes the I^2C module enter single buffer mode.

1. In master data receive mode, operation is the same as that in master data transmit mode as to transmit of a slave address and a 1-byte signal indicating the data transfer direction. At this time, however, select 1 (receive) for the data transfer direction.
2. The slave device automatically enters data transmission mode by the signal that indicates the data transfer direction and transmits 1-byte data in synchronization with the SCL clock output from the master device. The master device generates an interrupt by MDR (bit 1) at the eighth clock (in the timing of (2) in figure 19.10). Clear the MDR bit to 0 after the master device

suspend data transmission. (in the timing of (3) in figure 19.10.)

3. The slave device generates an interrupt by SDT (bit 2) indicating 1-byte data transfer end at the eighth clock (in the timing of (2) in figure 19.10) and an interrupt by SDE (bit 3) indicating data empty at the ninth clock (in the timing of (1) in figure 19.10). Clear SDE to 0 after writing slave transmit data to TXD.
4. To end data transfer, set FSB (bit 1) to 1 in the master control register in the master device to output a transfer end (force stop). When the last bit of a byte is transmitted/received, the I^2C module latches the FSB value and enters the stop state. Therefore, to stop the transfer after a specified byte data is transferred, the FSB bit must be set before the last byte is transferred. If the last byte is not correct, the protocol layer notifies the slave device that re-transmission is needed.

The timings of (1) to (3) in figure 19.10 are generated at the falling edge of the clock signal.

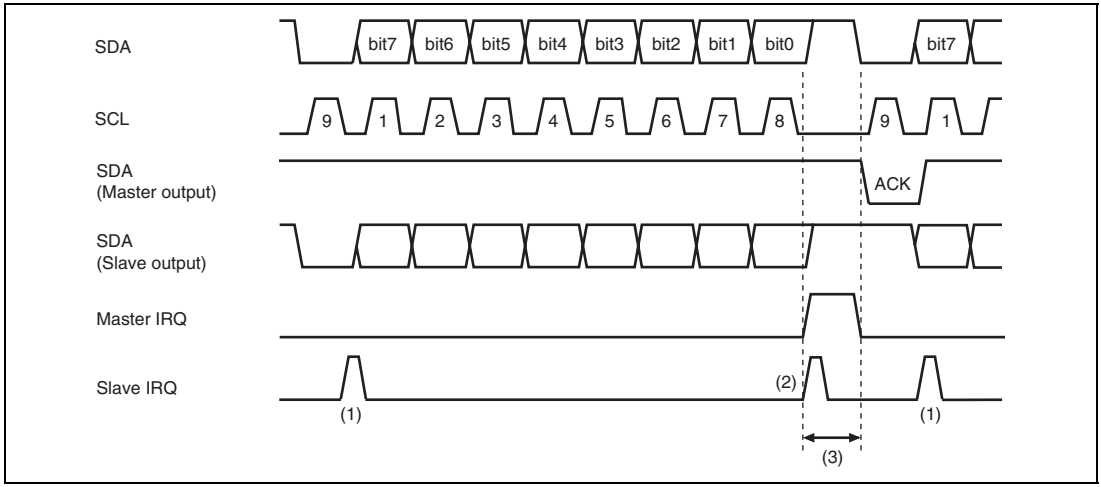


Figure 19.10 Data Receive Mode Timing Chart

Communications are unavailable in standby mode because clock supply stops. When entering the standby mode, terminate communications and check the status of the registers using the steps described below.

Carry out the following steps if communications are in progress:

1. When completing communication in I^2C master mode, check that the MST bit of ICMSR is 1 and clear the MIE bit of ICMCR to 0.
2. When completing communication in I^2C slave mode, check that SSR bit of ICSSR is 1 and clear SIE bit of ICSCR to 0.

Carry out the following steps if communications are NOT in progress:

1. Check that the MIE bit of ICMCR is 0.
2. Check that the SIE bit of ICSCR is 0.
3. Monitor the status of the FSCL and FSDA bits of ICMCR to check that FSCL is 1 and FSDA is 0. (Determine the timing for monitoring according to the SCL frequency to be used). When the MIE and SIE bits are 1, check that communications are not in progress and clear these bits to 0.

19.5 FIFO Mode Operation

In FIFO mode, the 16-stage FIFO buffer can be used. Registers related to the FIFO mode are ICFCR, ICFSR, ICFIER, ICRFDR, and ICTFDR. For details, refer to the description of these registers.

In FIFO mode, interrupt overhead can be reduced since transfer is performed in units of bytes specified by ICFCR.

19.5.1 Master Transmitter Operation (FIFO Buffer Mode)

1. The MDE bit and the MAT bit are set to 1 at the same timing, as in single buffer mode. At this time, the ESG bit should be cleared to 0. The master device holds SCL low level until the MDE bit is cleared to 0 in order to suspend data transfer.
2. FSB must be set to 1 at least one SCL clock cycle after the transmit FIFO data empty flag (TDFE) is set, and within 9 SCL clock cycles following that flag being set. (See figure 19.15.)* For example, to transfer 3 bytes of data, after 3 bytes of data have been written to the FIFO and transferred, verify that TDFE = 1 either by poling or with the transmit FIFO data empty (TXI) interrupt and then set FSB after the first SCL clock cycle and before the ninth SCL clock cycle completes following TDFE being set to 1. Note that care is required

interrupt response and handling times carefully.

Note: * If FSB is not set with this timing, the stop condition (P) may not be issued correctly.

19.5.2 Master Receiver Operation (FIFO Buffer Mode)

1. Clear the ESG bit to 0 when the RDF bit is set to 1 by receiving data for the byte count specified by the register.
2. The receive byte count specified by the RTRG bits in ICFCR can be selected from 1 to 16.
3. When the receive byte count reaches the specified count, an RDF interrupt is generated (RXIE = 1) and receive operation is stopped. The ACK signal is automatically returned until the RDF bit is set to 1.
4. Read all the receive data in the receive FIFO by the CPU when the RDF interrupt occurs. (Dummy-reading or reading before an RDF interrupt generation is not allowed.) However, if issuing a STOP condition is needed, carry out step 6.
5. To resume receive operations, read all the receive data in the receive FIFO with FSB = 0 and then clear the RDF flag in ICFSR to 0. (Modify the RTRG bits before clearing the RDF flag, if necessary.)
6. In order to issue a STOP condition, set the FSB bit to 1 and wait at least on bit period. Then read all the receive data in the receive FIFO and clear the RDF flag to 0. (The FSB bit must be set only in this timing).
7. In order to forcibly stop the receive operation before the receive byte count reaches the RTRG setting, manually control (start/stop) the FSCL bit and the FSDA bit in ICMCR and read data for the receive byte count indicated by ICRFDR from the receive FIFO.

19.6 Programming Examples

19.6.1 Master Transmitter (Single Buffer Mode)

In order to set up the master interface to transmit a data packet on the I^2C bus, take the following steps.

(1) Load the clock control register:

(a) Set SCL clock generation divider (SCGD) to 01h.
(SCL frequency of 400 kHz)

(b) Set clock division ratio to 2h.
(Peripheral clock: 33 MHz and I^2C internal clock (IICck): 11 MHz)

(a) Set address of slave being accessed to the master address register and the STM1 bit (write mode: 0).

(b) Set the first byte to be transmitted to the transmit data register.

(c) Set the master control register to 89h.
(MDBS = 1, MIE = 1, and ESG = 1)

(3) Wait for the address to be output:

(a) Wait for master events (interrupts by the MAT and MDE bits in the master status register).

(b) Set the master control register to 88h (the master device holds the SCL low level until the MDE bit is cleared in order to suspend the data transmission).
If only one data byte is to be transmitted, set the master control register 8Ah. (This enables the stop generation). This generates a stop on the bus as soon as one byte has been transmitted.

(c) Reset the MAT and MDE bits to 0.

(4) Monitor the progress of data byte transmission:

(a) Wait for a master event (the MDE bit in the master status register).

(b) Load the next data byte into the transmit data register.*

Note: * There is no need to observe the limitation that "execution must continue until the first data byte has been output" in this case.

(c) Reset the MDE bit.
Clear the MDE bit after setting the last transmit byte. After transmission of the last byte is started, MDE is generated. Before clearing the MDE bit, set 8Ah in the master control register (this must be done before the last transmission byte is completely output).
(Set the force stop bit.)

(5) Wait for the end of transmission:

(a) Wait for a master event (the MST bit in the master status register).

(b) Reset the MST bit after checking MNR (master NACK received).

In order to set up the master interface to receive a data packet on the I^2C bus, take the following steps.

(1) Load the clock control register:

(a) Set SCL clock generation divider (SCGD) to 01h.
(SCL frequency of 400 kHz)

(b) Set clock division factor (CDF) is set to 2h.
(Off-chip clock(sysclockfreq): 33MHz, on-chip clock (clockfreq): 11 MHz)

(2) Load the master control register and address:

(a) Set address of slave being accessed to master address register and the STM1 bit (read mode: 1).

(b) Set the Master Control Register to 89h.
(MDBS = 1, MIE = 1, and ESG = 1)

(3) Wait for the address to be output:

(a) Wait for master events (interrupts by the MAT bit and MDR bit in the master status register).

(b) Set the master control register to 88h (the master device keeps the SCL low level until the MDR bit is cleared in order to suspend the data reception).
If only one data byte is to be transmitted, set the master control register 8Ah. (This enables the stop generation). This generates a stop on the bus as soon as one byte has been received.

(c) Reset the MAT bit to 0**.**

(4) Monitor the progress of data byte reception:

(a) Wait for a master event (the MDR bit in the master status register).

(b) Read data from receive data register.
If the byte preceding the last byte transmitted by the slave device is to be received, for the last one-byte receive interrupt, i.e., MDR interrupt,

(c) Set the master control register to 8Ah.
(Set the force stop control bit).

(d) Reset the MDR bit.

(5) Wait for the end of transmission:

(a) Execute processing of the last byte receive interrupt (MDR), i.e., extract the data and clear MDR.

(c) Reset the MST bit to 0.

19.6.3 Master Transmitter—Restart—Master Receiver (Single Buffer Mode)

In order to set up the master interface for transmitting data packets to the I^2C bus, issuing a restart, and reading data back from the slave, take the following steps.

(1) Load the clock control register:

(a) Set SCL clock generation divider (SCGD) 01h.
(SCL frequency of 400 kHz)

(b) Set clock division ratio (CDF) is set to 2h.
(Off-chip clock(sysclockfreq): 33MHz, on-chip clock (clockfreq): 11 MHz)

(2) Load the master control register and address:

(a) Set address of slave being accessed to the master address register and the STM1 bit (write mode: 0).

(b) Set the master control register to 89h.
(MDBS = 1, MIE = 1, and ESG = 1)

(3) Wait for the address to be output:

(a) Wait for master device's events (interrupts by the MAT bit and the MDE bit in the master status register).

(b) Set address of slave being accessed to the master address register and the STM1 bit (read mode: 1).

If the enable start generation bit in the master control register is still set to 1, the master will issue a restart at the end of the byte transmission. Since the new address has been loaded as described above, the bus direction will be turned around.

(c) Reset the MAT bit to 0.

(4) Wait for the address to be output:

(a) Wait for master device's events (interrupts by the MAT bit and the MDR bit of the master status register).

(b) Set the master control register to 88h.
(The master device holds the SCL low level in order to suspend the data reception until the MDR bit is cleared to 0).

(c) Reset the MAT bit to 0.

(a) Wait for a master event (the MDR bit in the master status register).

Read data from the receive data register.

If next byte is to be the data immediately preceding the last byte transmitted by the slave device, for the receive interrupt for the byte immediately preceding the last one-byte, i.e., MDR interrupt,

(b) Set the master control register 8Ah.
(Set the force stop control bit.)

(c) Reset the MDR bit to 0**.**

(6) Wait for the end of transmission:

(a) Execute processing of the last byte receive interrupt (MDR), i.e., extract the data and clear MDR.

(b) Wait for the master device's event (the MST bit in the master status register).

(c) Reset the MST bit to 0.

19.6.4 Master Transmitter (FIFO Buffer Mode)

Operation example:

1. Set the clock rate to ICCCR.
2. Set the slave address, etc. to ICMAR.
3. Write transmission data to ICTXD (up to 16 bytes can be written).
4. Clear the TDFE flag.
5. ICMCR=H'0000 0009 (set ESG) //ESG=1, MIE=1, MDBS=0. (At this point, the slave address is output onto I^2C bus.)
6. Wait for MAT to be set to 1, and clear ESG, MAT, and MDE to 0. (Transmission data has been output until FIFO becomes empty.)
7. Wait for TDFE to be set to 1, and write subsequent transmit data to ICTXD.
   ICFSR=H'0000 0000 (Clear the flag.)
   (Repeat)
8. Set FSB to 1 after 1 SCL clock period has completed and before 9 SCL clock periods have completed after TDFE was set to 1.
   (See figure 19.15)
9. Clear the TEND flag to 0.

Operation example:

1. Set the clock rate to ICCCR.
2. Set the slave address, etc. to ICMAR.
3. Set the RDF trigger value to ICFCR.
   Verify that RDF is 0.
   (If RDF is 1, read the data from ICRXD and then clear RDF to 0.)
4. ICMCR=H'0000 0009 (set ESG) //ESG=1, MIE=1, MDBS=0. (At this point, the slave address is output onto I^2C bus.)
5. Wait for MAT, and clear ESG.
6. Wait for RDF, and read the data received from ICRXD.
   ICFSR=H'0000 0000 (clear the flag)
   (Repeat)
7. Wait for RDF, and set FSB to 1.
8. Wait for one bit period after setting RDF to 1 and read the data received from ICRXD.
9. ICFSR=H'0000 0000 (Clear the flag.)

19.7 Usage Notes

19.7.1 Restriction 1

For the STOP operation in I^2C master mode, there is a timing restriction on FSB setting.

Single buffer mode assumes to carry out a predefined number of byte transfers. The current circuit fetches the value of the FSB bit when the last bit of one byte is transmitted or received, and proceeds to the STOP operation. Consequently, to stop communications after the transfer of a specified number of bytes, the FSB bit should be set to 1 before the last byte is transferred.

This timing, however, contains a problem. In transmission mode, FSB is set to 1 BEFORE an ACK/NACK to the last byte is checked. For this, the following software actions are required.

- Software Actions
  The FSB bit must be set before the last eight bits are transmitted or received as mentioned above. In the case of transmission, ACK/NACK to the last byte needs to be checked. For example, the software should operate as follows.

polling after the last byte is set. At the same time, it checks MNR (master NACK received). If NACK is returned, it makes a branch to the error routine where the last byte is retransmitted. In master receiver mode, the software ends reception after confirming that the last byte has been received. However, if the last byte has any defect, it issues a retransmission request by the upper protocol.

19.7.2 Restriction 2

There are restrictions on the timing with which FSB may be set during I²C bus interface FIFO buffer mode master transmission.

When a stop condition (P) is issued in FIFO buffer mode, that stop condition (P) may not be issued correctly if the timing with which FSB is set to 1 meets either of the conditions (1) or (2) below.

(1) If that setting occurs during or before the transmission of the last bit (bit 8) of the next to last data item (data item (n-1)).
(2) If that setting occurs during or after the transmission of the last bit of the last data item (data item n).

If FSB is set to one with the above timing, in case (1) the stop condition (P) may be issued after the transmission of data item (n-1) and data item n may not be issued. Also, in case (2), a stop condition (P) may not be issued after transmission of data item n.

- Software Workaround

  Set FSB to 1 after the last bit of data item (n-1), and before the last bit of data item n, has been transmitted. In particular, as shown in the figure 19.11, set FSB within the 8 SCL clock cycle period following one clock SCL cycle after TDFE is set to 1 ((1) in the figure) and through the ninth SCL clock cycle after that setting ((2) in the figure).

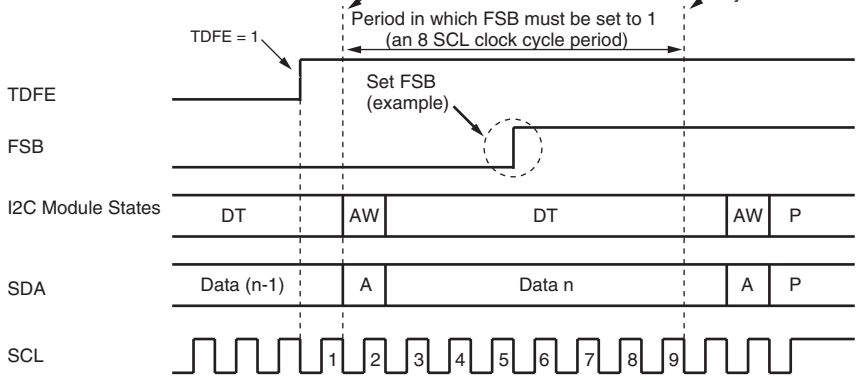


Figure 19.11 Period for Setting FSB = 1 for which a Stop Condition (P) will be Correctly Issued

The serial sound interface (SSI) module is a module designed to send or receive audio data interface with a variety of devices offering Philips format compatibility. It also provides additional modes for other common formats, as well as support for a burst and multi-channel mode.

20.1 Features

The SSI has the following features.

- Number of channels: Two channels (maximum)
- Operating modes: Compressed mode and non-compressed mode
  The compressed mode is used for continuous bit stream transfer
  The non-compressed mode supports all serial audio streams divided into channels.
- The SSI module is configured as any of a transmitter or receiver. The serial bus format can be used in the compressed and non-compressed mode.
- Asynchronous transfer between the buffer and the shift register
- Division ratios of the serial bus interface clock can be selected.
- Data transmission/reception can be controlled from the DMAC or interrupt.

Figure 20.1 is a block diagram of the SSI module.

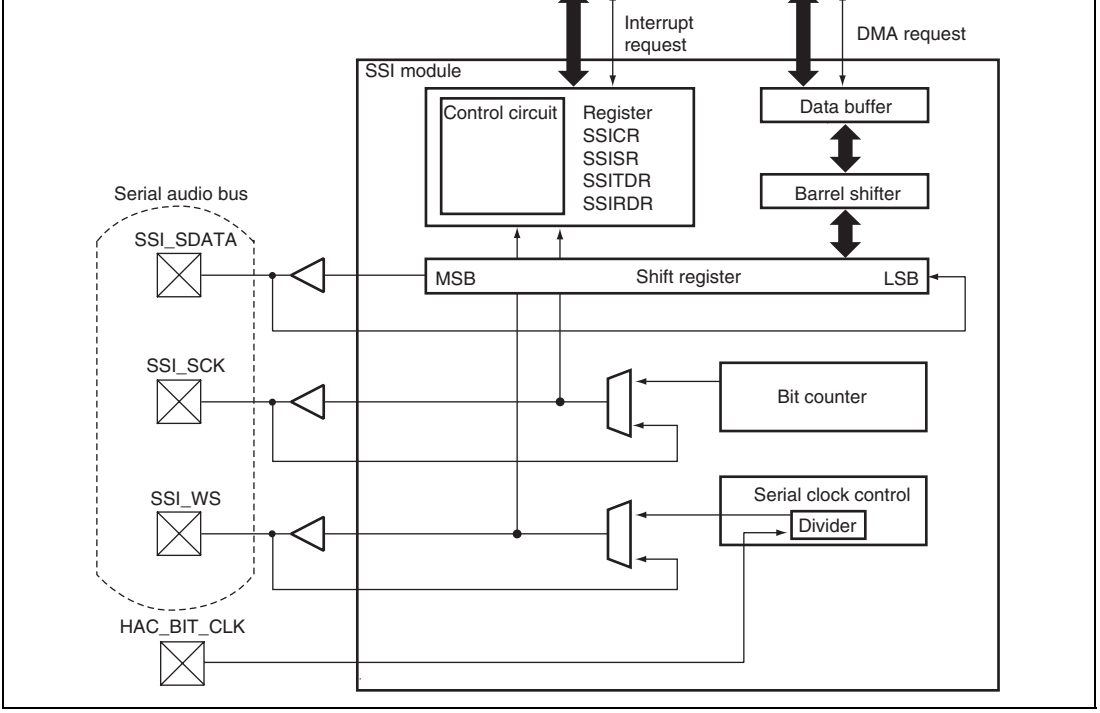


Figure 20.1 Block Diagram of SSI Module

20.2 Input/Output Pins

Table 20.1 lists the pin configurations relating to the SSI module.

Table 20.1 Pin Configuration

Name	Number of Pins	I/O	Function
SSI_SCK0	1	Input/Output	Serial bit clock
SSI_WS0	1	Input/Output	Word select
SSI_SDATA0	1	Input/Output	Serial data input/output
HAC_BIT_CLK0	1	Input	Divider input clock (oversampling clock)
SSI_SCK1	1	Input/Output	Serial bit clock
SSI_WS1	1	Input/Output	Word select
SSI_SDATA1	1	Input/Output	Serial data input/output
HAC_BIT_CLK1	1	Input	Divider input clock (oversampling clock)

The SSI module has the following registers. As for addresses of these registers and register status in each processing state, see section 32 List of Registers. Note that the initial value of SSISR is determined depending on the settings of the SCKD and CKDV bits in SSICR after the clock is supplied. Distinction with channels is omitted for the description of this document.

Table 20.2 Register Configuration (1)

Ch.	Register Name	Abbrev.	R/W	P4 Address	Area 7 Address	Size	Sync Clock
0	Control register 0	SSICR0	R/W	H'FE68 0000	H'1E68 0000	32	Pck
	Status register 0	SSISR0	R/W*1	H'FE68 0004	H'1E68 0004	32	Pck
	Transmit data register 0	SSITDR0	R/W	H'FE68 0008	H'1E68 0008	32	Pck
	Receive data register 0	SSIRDR0	R	H'FE68 000C	H'1E68 000C	32	Pck
1	Control register 1	SSICR1	R/W	H'FE69 0000	H'1E69 0000	32	Pck
	Status register 1	SSISR1	R/W*1	H'FE69 0004	H'1E69 0004	32	Pck
	Transmit data register 1	SSITDR1	R/W	H'FE69 0008	H'1E69 0008	32	Pck
	Receive data register 1	SSIRDR1	R	H'FE69 000C	H'1E69 000C	32	Pck

Table 20.2 Register Configuration (2)

Ch.	Register Name	Abbrev.	Power-on Reset by $\overline{\text{RESET}}$ Pin/WDT/ H-UDI	Manual Reset by $\overline{\text{RESET}}$ Pin/WDT/ Multiple Exception	Sleep by Sleep Instruction/ Deep Sleep	Standby by Hardware	Standby by Software/ Each Module
0	Control register 0	SSICR0	H'0000 0000	H'0000 0000	Retained	*2	Retained
	Status register 0	SSISR0	H'0200 0003	H'0200 0003	Retained		Retained
	Transmit data register 0	SSITDR0	H'0000 0000	H'0000 0000	Retained		Retained
	Receive data register 0	SSIRDR0	H'0000 0000	H'0000 0000	Retained		Retained
1	Control register 1	SSICR1	H'0000 0000	H'0000 0000	Retained		Retained
	Status register 1	SSISR1	H'0200 0003	H'0200 0003	Retained		Retained
	Transmit data register 1	SSITDR1	H'0000 0000	H'0000 0000	Retained		Retained
	Receive data register 1	SSIRDR1	H'0000 0000	H'0000 0000	Retained		Retained

Notes: 1. Bits 27 and 26 allow a write of 0 only to clear the flag.
2. When exiting hardware standby mode, this module is entered in the power-on reset state by the $\overline{\text{RESET}}$ pin.

SSICR is a 32-bit readable/writable register that controls the IRQ, selects each polarity status, and sets operating mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	DMEN	UIEN	OIEN	IIEN	DIEN	CHNL1	CHNL0	DWL2	DWL1	DWL0	SWL2	SWL1	SWL0
Initial value:	-	-	-	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	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	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SCKD	SWSD	SCKP	SWSP	SPDP	SDTA	PDTA	DEL	BREN	CKDV			MUEN	CPEN	TRMD	EN
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	Bit Name	Initial Value	R/W	Description
31 to 29	—	—	R	Reserved These bits are always read as an undefined value. The write value should always be 0.
28	DMEN	0	R/W	DMA Enable Enables or disables the DMA request. 0: DMA request disabled. 1: DMA request enabled.
27	UIEN	0	R/W	Underflow Interrupt Enable 0: Underflow interrupt disabled 1: Underflow interrupt enabled
26	OIEN	0	R/W	Overflow Interrupt Enable 0: Overflow interrupt disabled 1: Overflow interrupt enabled
25	IIEN	0	R/W	Idle Mode Interrupt Enable 0: Idle interrupt disabled 1: Idle interrupt enabled
24	DIEN	0	R/W	Data Interrupt Enable 0: Data interrupt disabled 1: Data interrupt enabled
23 22	CHNL1 CHNL0	0 0	R/W R/W	Channels These bits indicate the number of channels in each system word. These bits are ignored if CPEN = 1. 00: 1 channel per system word 01: 2 channels per system word 10: 3 channels per system word 11: 4 channels per system word

21	DWL2	0	R/W	Data Word Length
20	DWL1	0	R/W	These bits indicate the encoded number of bits in
19	DWL0	0	R/W	a data word. These bits are ignored if CPEN = 1. 000: 8 Bits 001: 16 Bits 010: 18 Bits 011: 20 Bits 100: 22 Bits 101: 24 Bits 110: 32 Bits 111: Setting prohibited
18	SWL2	0	R/W	System Word Length
17	SWL1	0	R/W	These bits indicate the encoded number of bits in
16	SWL0	0	R/W	a system word. These bits are ignored if CPEN = 1. 000: 8 Bits 001: 16 Bits 010: 24 Bits 011: 32 Bits 100: 48 Bits 101: 64 Bits 110: 128 Bits 111: 256 Bits
15	SCKD	0	R/W	Serial Bit Clock Direction 0: Serial clock input, slave mode 1: Serial clock output, master mode Note: In non-compressed mode (CPEN = 0), (SCKD, SWSD) may only be set to (0, 0) or (1, 1).
14	SWSD	0	R/W	Serial WS Direction 0: Serial word select input, slave mode 1: Serial word select output, master mode Note: In non-compressed mode (CPEN = 0), (SCKD, SWSD) may only be set to (0, 0) or (1, 1).

13	SCKP	0	R/W	Serial Clock Polarity 0: SSI_WS and SSI_SDATA change on falling edge of SSI_SCK (sampled on rising edge of SCK) 1: SSI_WS and SSI_SDATA change on rising edge of SSI_SCK (sampled on falling edge of SCK)

	SCKP = 0	SCKP = 1
SSI_SDATA input sampling timing in receive mode (TRMD = 0)	SSI_SCK rising edge	SSI_SCK falling edge
SSI_SDATA output change timing in transmit mode (TRMD = 1)	SSI_SCK falling edge	SSI_SCK rising edge
SSI_WS input sampling in slave mode (SWSD = 0)	SSI_SCK rising edge	SSI_SCK falling edge
SSI_WS output change timing in master mode (SWSD = 1)	SSI_SCK falling edge	SSI_SCK rising edge

12	SWSP	0	R/W	Serial WS Polarity The function of this bit depends on whether the SSI module is in non-compressed mode or compressed mode. CPEN = 0 (Non compressed mode): 0: SSI_WS is low for the first channel, high for the second channel 1: SSI_WS is high for the first channel, low for the second channel CPEN = 1 (Compressed mode): 0: SSI_WS is active high flow control. WS = high means data should be transferred, low means data should not be transferred. 1: SSI_WS is active low flow control. WS = low means data should be transferred, high means data should not be transferred.

11	SPDP	0	R/W	Serial Padding Polarity This bit is ignored if CPEN = 1. 0: Padding bits are low 1: Padding bits are high Note: When MUEN = 1, the padding bits will be at the low level. (The muting function takes priority.)
10	SDTA	0	R/W	Serial Data Alignment This bit is ignored if CPEN = 1. 0: Serial data is transmitted/ received first, followed by padding bits. 1: Padding bits are transmitted/ received first, followed by serial data.
9	PDTA	0	R/W	Parallel Data Alignment This bit is ignored if CPEN = 1. If the data word length = 32, 16 or 8 then this bit has no meaning. This bit is applied to SSIRDR in receive mode and to SSITDR in transmit mode. 0: Parallel data (SSITDR or SSIRDR) is left aligned 1: Parallel data (SSITDR or SSIRDR) is right aligned • DWL = 000 (data word length: 8 bits), PDTA ignored All data bits in SSIRDR or SSITDR are used on the audio serial bus. Four data words are transmitted/received in each 32-bit access. The first data word is stored in bits 7 to 0, the second from bits 15 to 8, the third from bits 23 to 16 and the last data word is stored in bits 31 to 24. • DWL = 001 (data word length: 16 bits), PDTA ignored All data bits in SSIRDR or SSITDR are used on the audio serial bus. Two data words are transmitted/received in each 32-bit access. The first data word is stored in bits 15 to 0 and the second data word is stored in bits 31 to 16.

9				• DWL = 010, 011, 100, 101 (data word length: 18, 20, 22 and 24 bits), PDTA = 0 (left aligned) The data bits which are used in SSIRDR or SSITDR are the following: Bits 31 to (32 – number of bits having data word length specified by DWL). If DWL = 011 then data word length is 20 bits and bits 31 to 12 are used of either SSIRDR or SSITDR. All other bits are ignored or reserved. • DWL = 010, 011, 100, 101 (data word length: 18, 20, 22 and 24 bits), PDTA = 1 (right aligned) The data bits which are used in SSIRDR or SSITDR are the following: Bits (number of bits having data word length specified by DWL - 1) to 0. If DWL = 011 then data word length is 20 bits and bits 19 to 0 are used of either SSIRDR or SSITDR. All other bits are ignored or reserved. • DWL = 110 (data word length: 32 bits), PDTA ignored All data bits in SSIRDR or SSITDR are used on the audio serial bus.
8	DEL	0	R/W	Serial Data Delay 0: 1 clock cycle delay between SSI_WS and SSI_SDATA 1: No delay between SSI_WS and SSI_SDATA This bit must be set to 1 when CPEN = 1. A one-clock cycle delay is not supported when the SSI module is configured to be a slave transmitter (SWSD = 0 and TRMD = 1). In this situation, this bit should be set to 0.

7	BREN	0	R/W	Burst Mode Enable 0: Burst mode is disabled. 1: Burst mode is enabled. Burst mode is used in conjunction with compressed mode (CPEN = 1). When burst mode is enabled the SSI_SCK signal is gated. Clock pulses are output only when there is valid serial data being output on SSI_SDATA.
6 to 4	CKDV	All 0	R/W	Serial Oversampling Clock Division Ratio These bits define the division ratio between oversampling Clock (HAC_BIT_CLK) and the serial bit clock. These bits are ignored if SCKD = 0. The Serial Bit Clock is used for the shift register and is provided on the SSI_SCK pin. 000: (Serial bit clock frequency = oversampling clock frequency/1) 001: (Serial bit clock frequency = oversampling clock frequency/2) 010: (Serial bit clock frequency = oversampling clock frequency/4) 011: (Serial bit clock frequency = oversampling clock frequency/8) 100: (Serial bit clock frequency = oversampling clock frequency/16) 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited
3	MUEN	0	R/W	Mute Enable When in transmit mode (TRMD = 1), by making MUEN = 1, the output of SSI_SDATA will be in low level. 0: The SSI module is not muted. 1: The SSI module is muted.
2	CPEN	0	R/W	Compressed Mode Enable 0: Compressed mode disabled 1: Compressed mode enabled Note: In compressed mode (CPEN = 1), only use operations other than slave transmitter (SWSD = 0 and TRMD = 1).
1	TRMD	0	R/W	Transmit/Receive Mode Select 0: The SSI module is in receive mode 1: The SSI module is in transmit mode

0	EN	0	R/W	SSI Module Enable 0: The SSI module is disabled 1: The SSI module is enabled

20.3.2 Status Register (SSISR)

SSISR is configured by status flags that indicate the operating status of the SSI module and bits that indicate the current channel number and word number.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	DMRQ	UIRQ	OIRQ	IIRQ	DIRQ	-	-	-	-	-	-	-	-
Initial value:	-	-	-	0	0	0	1	0	-	-	-	-	-	-	-	-
R/W:	R	R	R	R	R/W*	R/W*	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
	-	-	-	-	-	-	-	-	-	-	-	-	CHNO1	CHNO0	SWNO	IDST
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	0	0	1	1
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 29	—	—	R	Reserved These bits are always read as an undefined value. The write value should always be 0.
28	DMRQ	0	R	DMA Request Status Flag This status flag allows the CPU to see the status of the DMA request of SSI module. TRMD = 0 (Receive Mode): • If DMRQ = 1 then SSIRDR has unread data. • If SSIRDR is read then DMRQ = 0 until there is new unread data. TRMD = 1 (Transmit Mode): • If DMRQ = 1, SSITDR requests data to be written to continue the transmission onto the audio serial bus. • Once data is written to SSITDR then DMRQ = 0 until further transmit data is requested.

27	UIRQ	0	R/W*	Underflow Error Interrupt Status Flag This status flag indicates that the data has been supplied at a lower rate than the required rate. This bit is set to 1 regardless of the setting of UIEN bit. In order to clear it to 0, write 0 in it. If UIRQ = 1 and UIEN = 1, then an interrupt will be generated. When TRMD = 0 (Receive Mode): If UIRQ = 1, it indicates that SSIRDR was read out before DMRQ and DIRQ bits would indicate the existence of new unread data. In this instance, the same received data may be stored twice by the host, which can lead to destruction of multi-channel data. When TRMD = 1 (Transmit Mode): If UIRQ = 1, it indicates that the transmitted data was not written in SSITDR. By this, the same data may be transmitted one time too often, which can lead to destruction of multi-channel data. Consequently, erroneous SSI data will be output, which makes this error more serious than underflow in the receive mode. Note: When underflow error occurs, the data in the data buffer will be transmitted until the next data is written in.

26	OIRQ	0	R/W*	Overflow Error Interrupt Status Flag This status flag indicates that the data has been supplied at a higher rate than the required rate. This bit is set to 1 regardless of the setting of OIEN bit. In order to clear it to 0, write 0 in it. If OIRQ = 1 and OIEN = 1, then an interrupt will be generated. When TRMD = 0 (Receive Mode): If OIRQ = 1, it indicates that the previous unread data had not been read out before new unread data was written in SSIRDR. This may cause the loss of data, which can lead to destruction of multi-channel data. Note: When overflow error occurs, the data in the data buffer will be overwritten by the next data sent from the SSI interface. When TRMD = 1 (Transmit Mode): If OIRQ = 1, it indicates that SSITDR had data written in before the data in SSITDR was transferred to the shift register. This may cause the loss of data, which can lead to destruction of multi-channel data.
25	IIRQ	1	R	Idle Mode Interrupt Status Flag This status flag indicates whether the SSI module is in the idle status. This bit is set to 1 regardless of the setting of IIEN bit, so that polling will be possible. The interrupt can be masked by clearing IIEN bit to 0, but writing 0 in this bit will not clear the interrupt. If IIRQ = 1 and IIEN = 1, then an interrupt will be generated. 0: The SSI module is not in the idle status. 1: The SSI module is in the idle status.

24	DIRQ	0	R	Data Interrupt Status Flag This status flag indicates that the SSI module requires that data be either read out or written in. This bit is set to 1 regardless of the setting of DIEN bit, so that polling will be possible. The interrupt can be masked by clearing DIEN bit to 0, but writing 0 in this bit will not clear the interrupt. If DIRQ = 1 and DIEN = 1, then an interrupt will be generated. When TRMD = 0 (Receive Mode): 0: No unread data exists in SSIRDR. 1: Unread data exists in SSIRDR. When TRMD = 1 (Transmit Mode): 0: The transmit buffer is full. 1: The transmit buffer is empty, and requires that data be written in SSITDR.
23 to 4	—	—	R	Reserved These bits are always read as an undefined value. The write value should always be 0.
3	CHNO1	0	R	Channel Number The number indicates the current channel. When TRMD = 0 (Receive Mode): This bit indicates to which channel the current data in SSIRDR belongs. When the data in SSIRDR is updated by transfer from the shift register, this value will change. When TRMD = 1 (Transmit Mode): This bit indicates the data of which channel should be written in SSITDR. When data is copied to the shift register, regardless whether the data is written in SSITDR, this value will change.
2	CHNO0	0	R	

1	SWNO	1	R	Serial Word Number The number indicates the current word number. When TRMD = 0 (Receive Mode): This bit indicates which system word the current data in SSIRDR is. Regardless whether the data has been read out from SSIRDR, when the data in SSIRDR is updated by transfer from the shift register, this value will change. When TRMD = 1 (Transmit Mode): This bit indicates which system word should be written in SSITDR. When data is copied to the shift register, regardless whether the data is written in SSITDR, this value will change.
0	IDST	1	R	Idle Mode Status Flag Indicates that the serial bus activity has ceased. This bit is cleared if EN = 1 and the Serial Bus is currently active. This bit can be set to 1 automatically under the following conditions. SSI = Serial bus master transmitter (SWSD = 1 and TRMD = 1): This bit is set to 1 if the EN bit is cleared and the current system word is completed. It can also be set to 1 when the EN bit has been cleared and the data that has been written to SSITDR is output on the serial data input/output pin (SSI_SDATA), j.e., the serial data of the system word length is output. SSI = Serial bus master receiver (SWSD = 1 and TRMD = 0): This bit is set to 1 if the EN bit is cleared and the current system word is completed. SSI = Slave transmitter/ receiver (SWSD = 0): This bit is set to 1 if the EN bit is cleared and the current system word is completed. To terminate the transfer, clear SSICR.EN to 0 and continue to input the WS signal until SSICR.IDST becomes 1. Note: If the external device stops the serial bus clock before the current system word is completed then this bit will never be set.

Note: * These bits are readable/writable bits. If writing 0, these bits are initialized, although writing 1 is ignored.

SSITDR is a 32-bit register that stores data to be transmitted.

Data written to SSITDR is transferred to the shift register as it is required for transmission. If the data word length is less than 32 bits then its alignment should be as defined by the PDTA control bit.

Reading this register will return the data in the buffer.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	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
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

20.3.4 Receive Data Register (SSIRDR)

SSIRDR is a 32-bit register that stores the received data.

Data in SSIRDR is transferred from the shift register as each data word is received. If the data word length is less than 32 bits then its alignment should be as defined by the PDTA control bit in SSICR.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
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
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

20.4.1 Bus Format

The SSI module can operate as a transmitter or a receiver and can be configured into many serial bus formats in either mode.

The bus formats can be one of eight major modes as shown in table 20.3.

Table 20.3 Bus Formats of SSI Module

Bus Format	TPMD	CPEN	SCKD	SWSD	EN	MUEN	DIEN	IIEN	OIEN	UIEN	DEL	PDTA	SDTA	SPDP	SWSP	SCKP	SWL[2:0]	DWL[2:0]	CHNL[1:0]
Non-Compressed Slave Receiver	0	0	0	0	Control bits						Configuration bits								
Non-Compressed Slave Transmitter	1	0	0	0															
Non-Compressed Master Receiver	0	0	1	1															
Non-Compressed Master Transmitter	1	0	1	1															
Compressed Slave Receiver	0	1	0/1	0	Control bits						1	Ignored			Configu-ration bits		Ignored		
Compressed Slave Transmitter	Cannot be used																		
Compressed Master Receiver	0	1	0/1	1	Control bits						1	Ignored			Configu-ration bits		Ignored		
Compressed Master Transmitter	1	1	0/1	1							1								

20.4.2 Non-Compressed Modes

The non-compressed mode is designed to support all serial audio streams which are split into channels. It can support Philips, Sony and Matsushita modes as well as many more variants on these modes.

(1) Slave Receiver

This mode allows the SSI module to receive serial data from another device. The clock and word select signals used for the serial data stream are also supplied from an external device. If these

guaranteed.

(2) Slave Transmitter

This mode allows the SSI module to transmit serial data to another device. The clock and word select signals used for the serial data stream are also supplied from an external device. If these signals do not conform to the format as specified in the SSI module then operation is not guaranteed.

(3) Master Receiver

This mode allows the SSI module to receive serial data from another device. The clock and word select signals are internally derived from the HAC_BIT_CLK input clock. The format of these signals is as defined in the SSI module. If the incoming data does not conform to the defined format then operation is not guaranteed.

(4) Master Transmitter

This mode allows the SSI module to transmit serial data to another device. The clock and word select signals are internally derived from the HAC_BIT_CLK input clock. The format of these signals is as defined in the configuration bits in the SSI module.

(5) Configuration Fields - Word Length Related

All configuration bits relating to the word length of SSICR are valid in non-compressed modes.

There are many configurations that the SSI module can support and it is not sensible to show all of the Serial Data formats in this document. Some of the combinations are shown below for the popular formats by Philips, Sony, and Matsushita.

Figures 20.2 and 20.3 show the supported Philips protocol both with padding and without. Padding occurs when the data word length is smaller than the system word length.

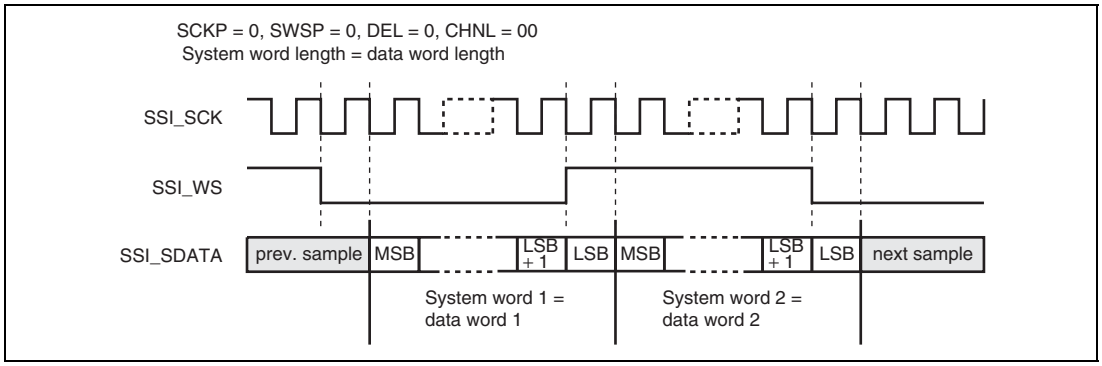


Figure 20.2 Philips Format (with no Padding)

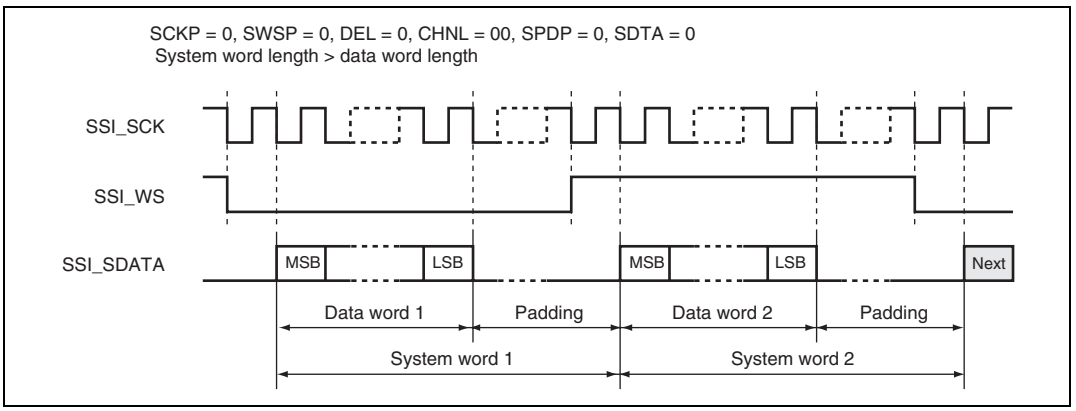


Figure 20.3 Philips Format (with Padding)

Figure 20.4 shows the format used by Sony. Figure 20.5 shows the format used by Matsushita. Padding is assumed in both cases, but may not be present in a final implementation if the system word length equals the data word length.

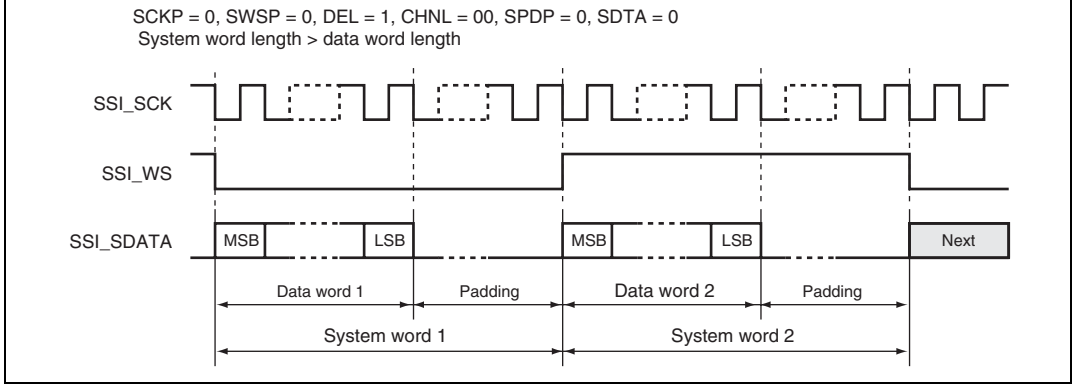


Figure 20.4 Sony Format (with Serial Data First, Followed by Padding Bits)

3. Matsushita Format

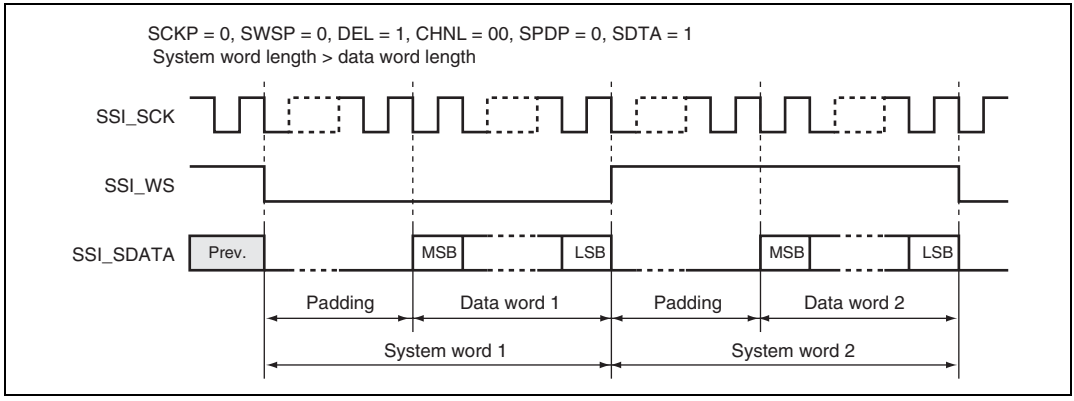


Figure 20.5 Matsushita Format (with Padding Bits First, Followed by Serial Data)

(6) Multi-Channel Formats

Some devices extend the definition of the specification by Philips and allow more than 2 channels to be transferred within two system words.

The SSI module supports the transfer of 4, 6 and 8 channels by the use of the CHNL, SWL and DWL bits. It is important that the system word length (SWL) is greater than or equal to the number of channels (CHNL) times the data word length (DWL).

Table 20.4 shows the number of padding bits for each of the valid configurations. If a setup is not valid it does not have a number in the following table and has instead a dash.

Padding Bits Per System Word			DWL[2:0]	000	001	010	011	100	101	110
CHNL [1:0]	Decoded Channels per System Word	SWL [2:0]	Decoded Word Length	8	16	18	20	22	24	32
00	1	000	8	0	—	—	—	—	—	—
		001	16	8	0	—	—	—	—	—
		010	24	16	8	6	4	2	0	—
		011	32	24	16	14	12	10	8	0
		100	48	40	32	30	28	26	24	16
		101	64	56	48	46	44	42	40	32
		110	128	120	112	110	108	106	104	96
		111	256	248	240	238	236	234	232	224
01	2	000	8	—	—	—	—	—	—	—
		001	16	0	—	—	—	—	—	—
		010	24	8	—	—	—	—	—	—
		011	32	16	0	—	—	—	—	—
		100	48	32	16	12	8	4	0	—
		101	64	48	32	28	24	20	16	0
		110	128	112	96	92	88	84	80	64
		111	256	240	224	220	216	212	208	192
10	3	000	8	—	—	—	—	—	—	—
		001	16	—	—	—	—	—	—	—
		010	24	0	—	—	—	—	—	—
		011	32	8	—	—	—	—	—	—
		100	48	24	0	—	—	—	—	—
		101	64	40	16	10	4	—	—	—
		110	128	104	80	74	68	62	56	32
		111	256	232	208	202	196	190	184	160
11	4	000	8	—	—	—	—	—	—	—
		001	16	—	—	—	—	—	—	—
		010	24	—	—	—	—	—	—	—
		011	32	0	—	—	—	—	—	—
		100	48	16	—	—	—	—	—	—
		101	64	32	0	—	—	—	—	—
		110	128	96	64	56	48	40	32	0
		111	256	224	192	184	176	168	160	128

is transmitted in order on the serial audio bus.

In the case of the SSI module configured as a receiver each word received on the Serial Audio Bus is presented for reading in order by SSIRDR.

Figures 20.6 to 20.8 show how 4, 6 and 8 channels are transferred on the serial audio bus.

Note that there are no padding bits in the first example, serial data is transmitted/received first and followed by padding bits in the second example, and padding bits are transmitted/received first and followed by serial data in the third example. This selection is purely arbitrary.

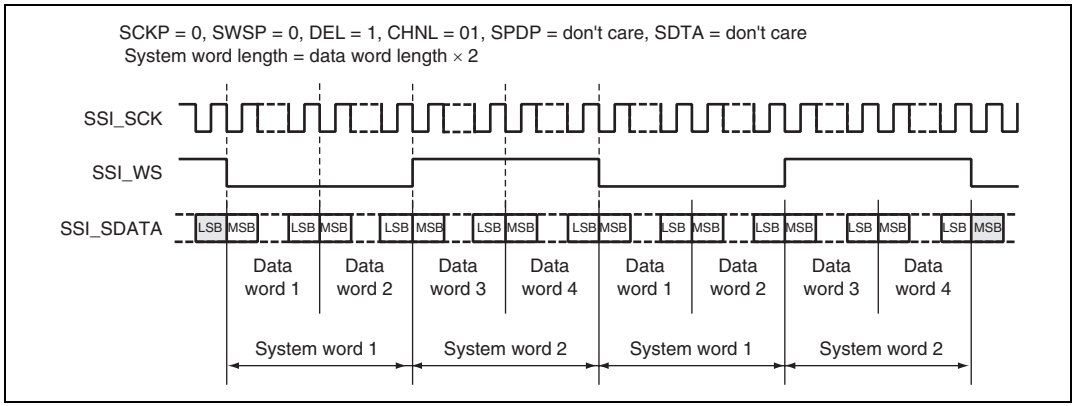


Figure 20.6 Multichannel Format (4 Channels, No Padding)

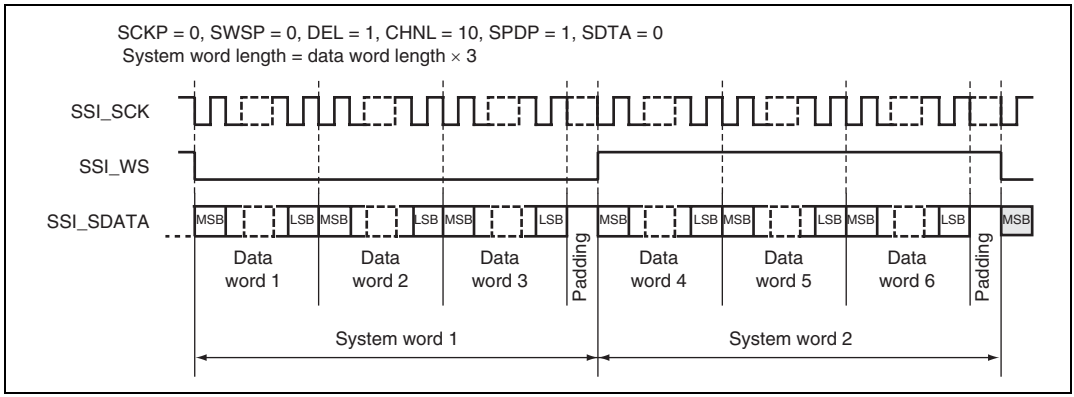


Figure 20.7 Multichannel Format (6 Channels with High Padding)

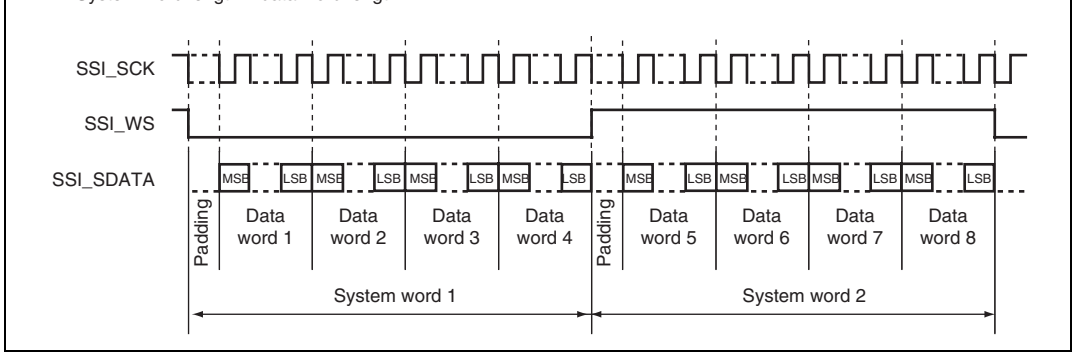


Figure 20.8 Multichannel Format (8 Channels, with Padding Bits First, Followed by Serial Data, with Padding)

(7) Configuration Fields - Signal Format Fields

There are several more configuration bits in non-compressed mode which will now be demonstrated. These bits are NOT mutually exclusive, however some configurations will probably not be useful for any other device.

They are demonstrated by referring to the following basic sample format shown in figure 20.9.

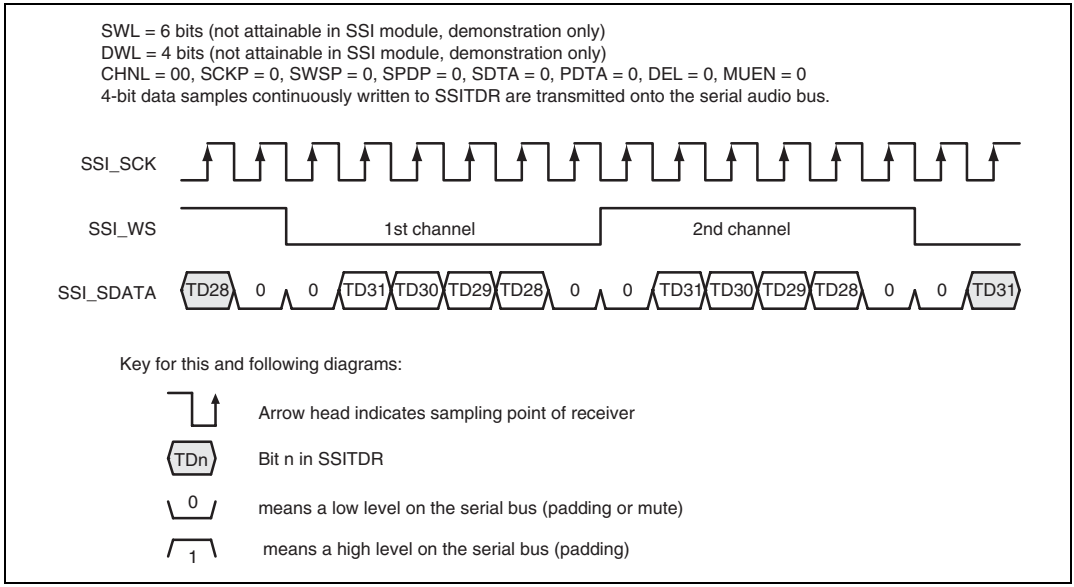


Figure 20.9 Basic Sample Format (Transmit Mode with Example System/Data Word Length)

these are possible with the SSI module but are used only for clarification of the other configuration bits.

1. Inverted Clock

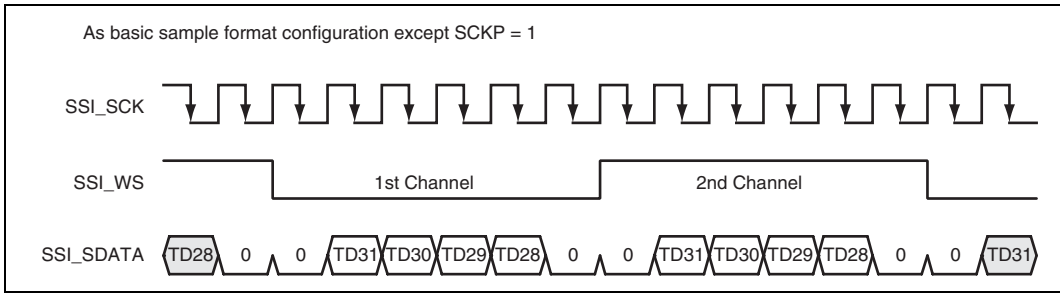


Figure 20.10 Inverted Clock

2. Inverted Word Select

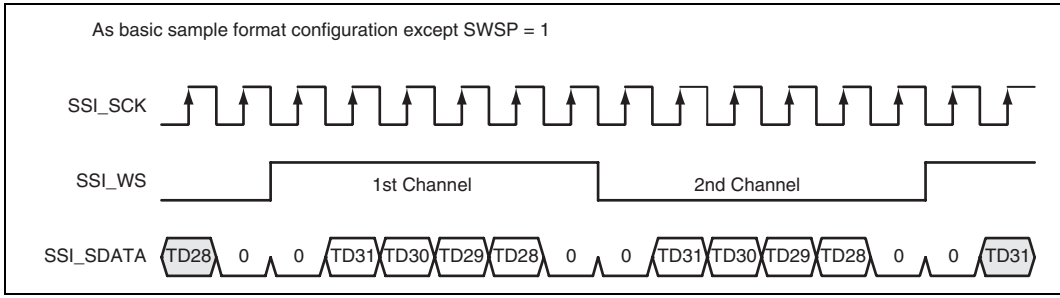


Figure 20.11 Inverted Word Select

3. Inverted Padding Polarity

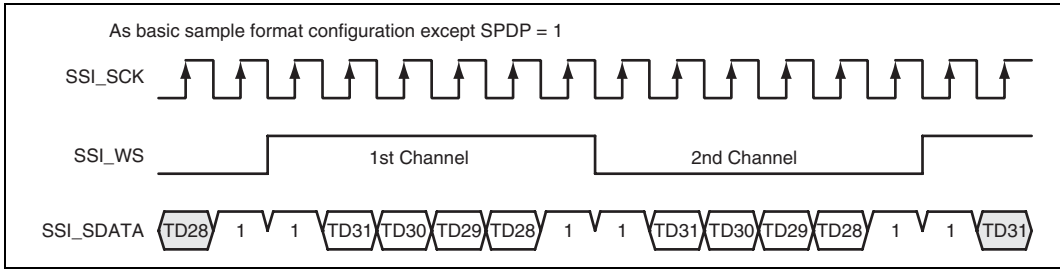


Figure 20.12 Inverted Padding Polarity

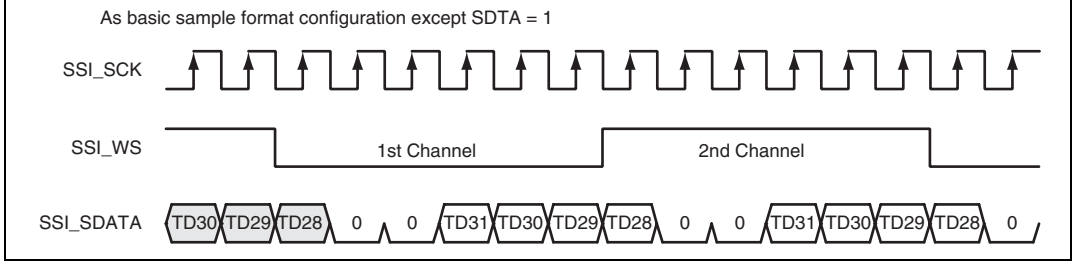


Figure 20.13 Padding Bits First, Followed by Serial Data, with Delay

5. Padding Bits First, Followed by Serial Data, without Delay

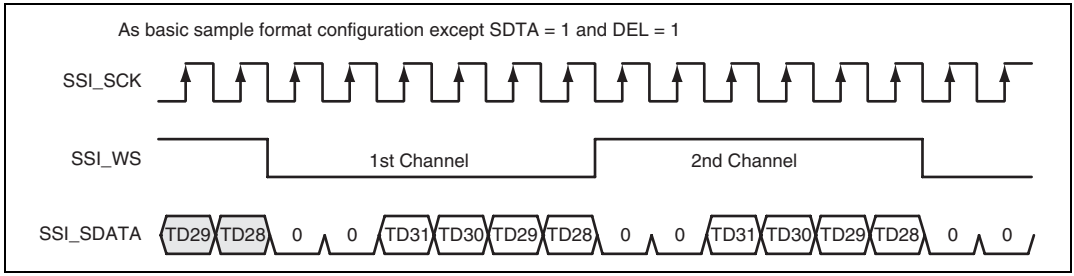


Figure 20.14 Padding Bits First, Followed by Serial Data, without Delay

6. Serial Data First, Followed by Padding Bits, without Delay

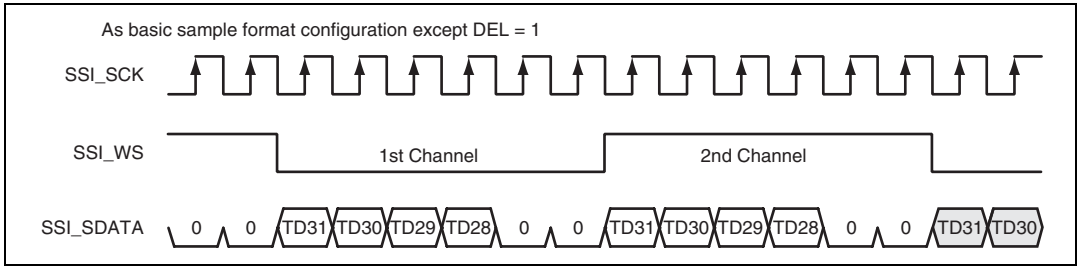


Figure 20.15 Serial Data First, Followed by Padding Bits, without Delay

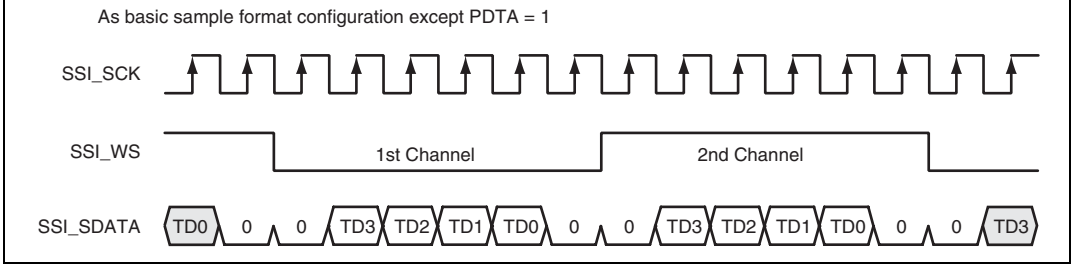


Figure 20.16 Parallel Right Aligned with Delay

8. Mute Enabled

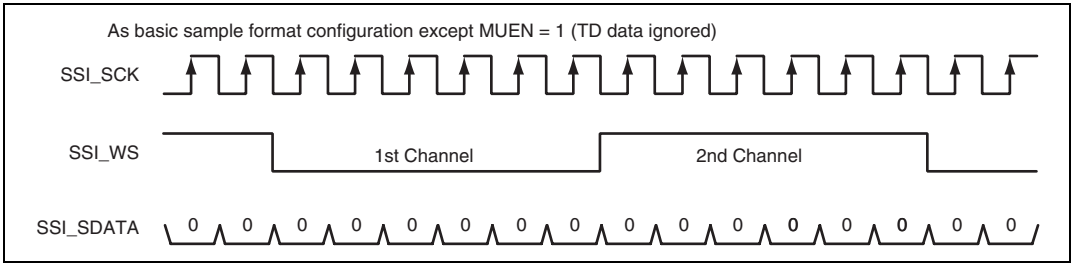


Figure 20.17 Mute Enabled

20.4.3 Compressed Modes

The compressed mode is used to transfer a continuous bit stream. This would typically be a compressed bit stream which requires downstream decoding.

In streaming mode (burst mode not enabled) there is no concept of a data word. However in order to receive and transmit it is necessary to transfer between the serial bus and word formatted memory. Therefore the word boundary selection is arbitrary during receive/transmit and must be dealt with by another module. When burst mode is enabled then data bits being transmitted can be identified by virtue of the fact that the serial clock output is only activated when there is a word to be output and only the required number of clock pulses necessary to clock out each 32-bit word are generated. The serial bit clock stops at a low level when SSICR.SCKP = 0, and at a high level when SSICR.SCKP = 1. Note burst mode is only valid in the context of the SSI module being a transmitter of data. Burst mode data cannot be received by this module.

Data is transmitted and received in blocks of 32 bits, and the first bit received/transmitted bit is bit 31 when stored in memory.

transmitter can transmit another data burst.

Figures 20.18 and 20.19 show the compressed mode data transfer, with burst mode disabled, and enabled, respectively.

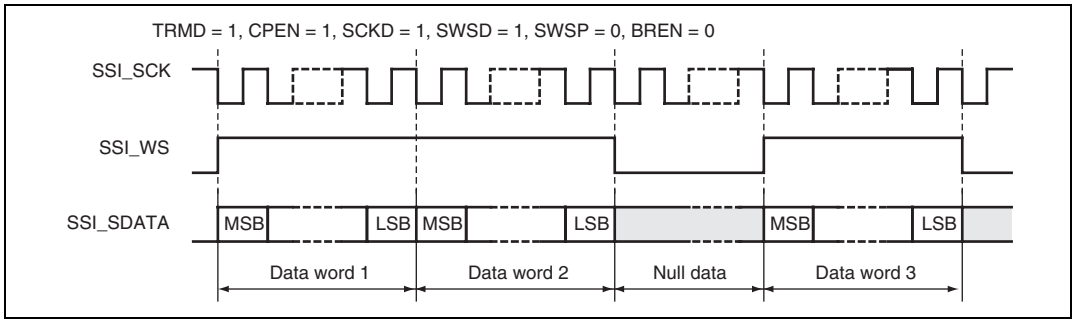


Figure 20.18 Compressed Data Format, Master Transmitter, Burst Mode Disabled

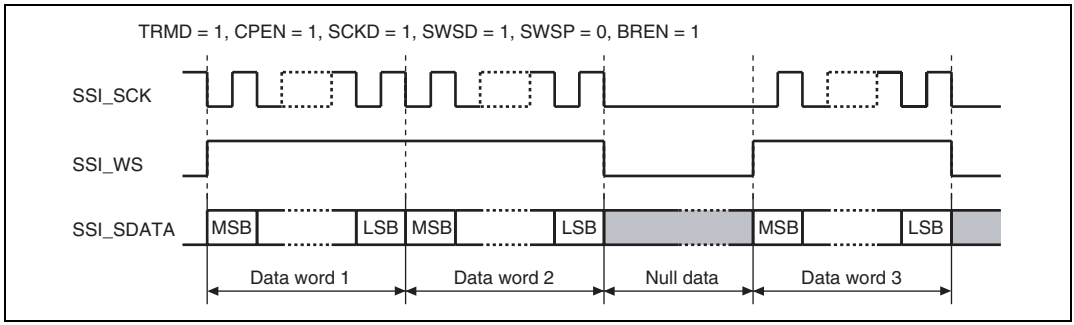


Figure 20.19 Compressed Data Format, Master Transmitter, and Burst Mode Enabled

(1) Slave Receiver

This mode allows the module to receive a serial bit stream from another device and store it in memory.

The shift register clock can be supplied from an external device or from an internal clock.

The word select pin is used as an input flow control. Assuming that SWSP = 0 if SSI_WS is high then the module will receive the bit stream in blocks of 32 bits, one data bit per clock. If SSI_WS goes low then the module will complete the current 32-bit block and then stop any further reception, until SSI_WS goes high again.

This mode cannot be used.

(3) Master Receiver

This mode allows the SSI module to receive a serial bit stream from another device and store it in memory.

The shift register clock can be supplied from an external device or from an internal clock.

The word select pin is used as an output flow control. The module always asserts the word select signal to indicate it can receive more data continuously. It is the responsibility of the host CPU to ensure it can transmit data to the SSI module in time to ensure no data is lost.

(4) Master Transmitter

This mode allows the module to transmit a serial bit stream from internal memory to another device.

The shift register clock can be supplied from an external device or from an internal clock.

The word select pin is used as an output flow control. The module always asserts the word select signal to indicate it will transmit more data continuously. Word select signal is not asserted until the first word is ready to transmit however. It is the responsibility of the receiving device to ensure it can receive the serial data in time to ensure no data is lost.

When the configuration for data transfer is completed, the SSI module can work with the minimum interaction with CPU. The CPU specifies settings for the SSI module and DMAC then handles overflow/ underflow interrupts if required.

20.4.4 Operation Modes

There are three modes of operation: configuration, enabled and disabled. Figure 20.20 shows the transition diagram between these operation modes.

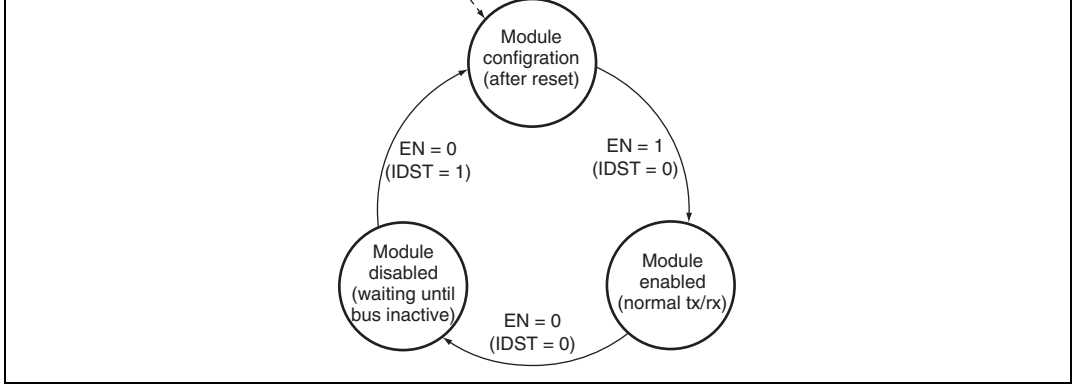


Figure 20.20 Transition Diagram between Operation Modes

(1) Configuration Mode

This mode is entered after the module is released from reset. All required settings in the control register should be defined in this mode, before the SSI module is enabled by setting the EN bit.

Setting the EN bit causes the SSI module to enter the module enabled mode.

(2) Module Enabled Mode

Operation of the module in this mode depends on the selected operating mode. For details, see section 20.4.5, Transmit Operation and section 20.4.6, Receive Operation.

20.4.5 Transmit Operation

Transmission can be controlled in one of two ways: either DMA or an interrupt driven.

DMA driven is preferred to reduce the CPU load. In DMA control mode, an underflow or overflow of data or DMAC transfer end is notified by using an interrupt.

The alternative is using the interrupts that the SSI module generates to supply data as required. This mode has a higher interrupt load as the SSI module is only double buffered and will require data to be written at least every system word period.

When the SSI module has been enabled for transmission, at least one longword must be written to the transmit register before disabling the transmitter (In 16-bit mode, two 16-bit words will be transmitted; in 8-bit mode, 4 bytes will be transmitted. For all other data sizes, one data word will be transmitted, e.g., 18 bits for 18-bit mode.)

When disabling the SSI module, the SSI clock* must be supplied continuously until the module enters in the idle state, indicated by the IIRQ bit.

Figure 20.21 shows the transmit operation in the DMA controller mode. Figure 20.22 shows the transmit operation in the Interrupt controller mode.

Note: * SCKD = 0: Clock input through the SSI_SCK pin
SCKD = 1: Clock input through the HAC_BIT_CLK pin

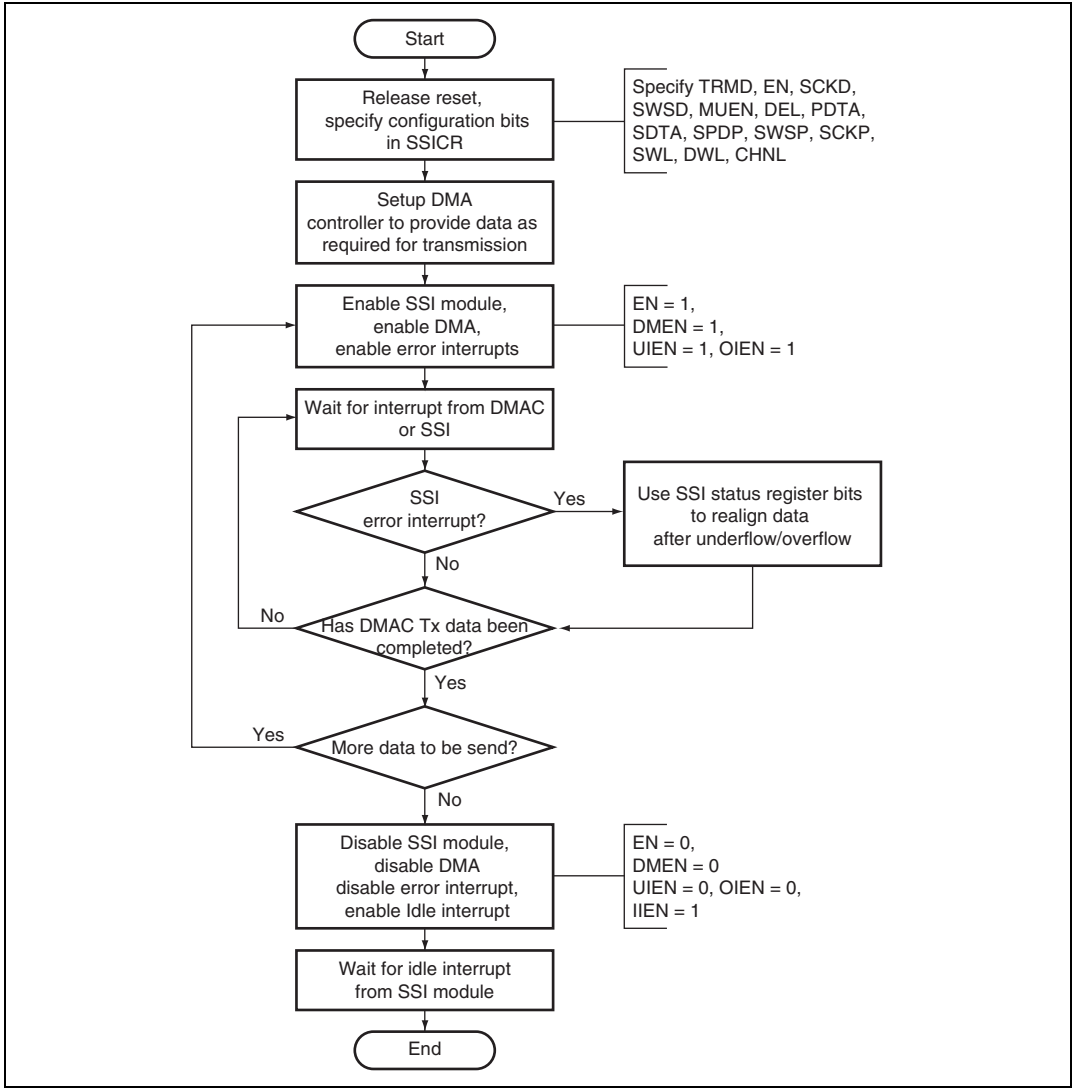


Figure 20.21 Transmission Using DMA Controller

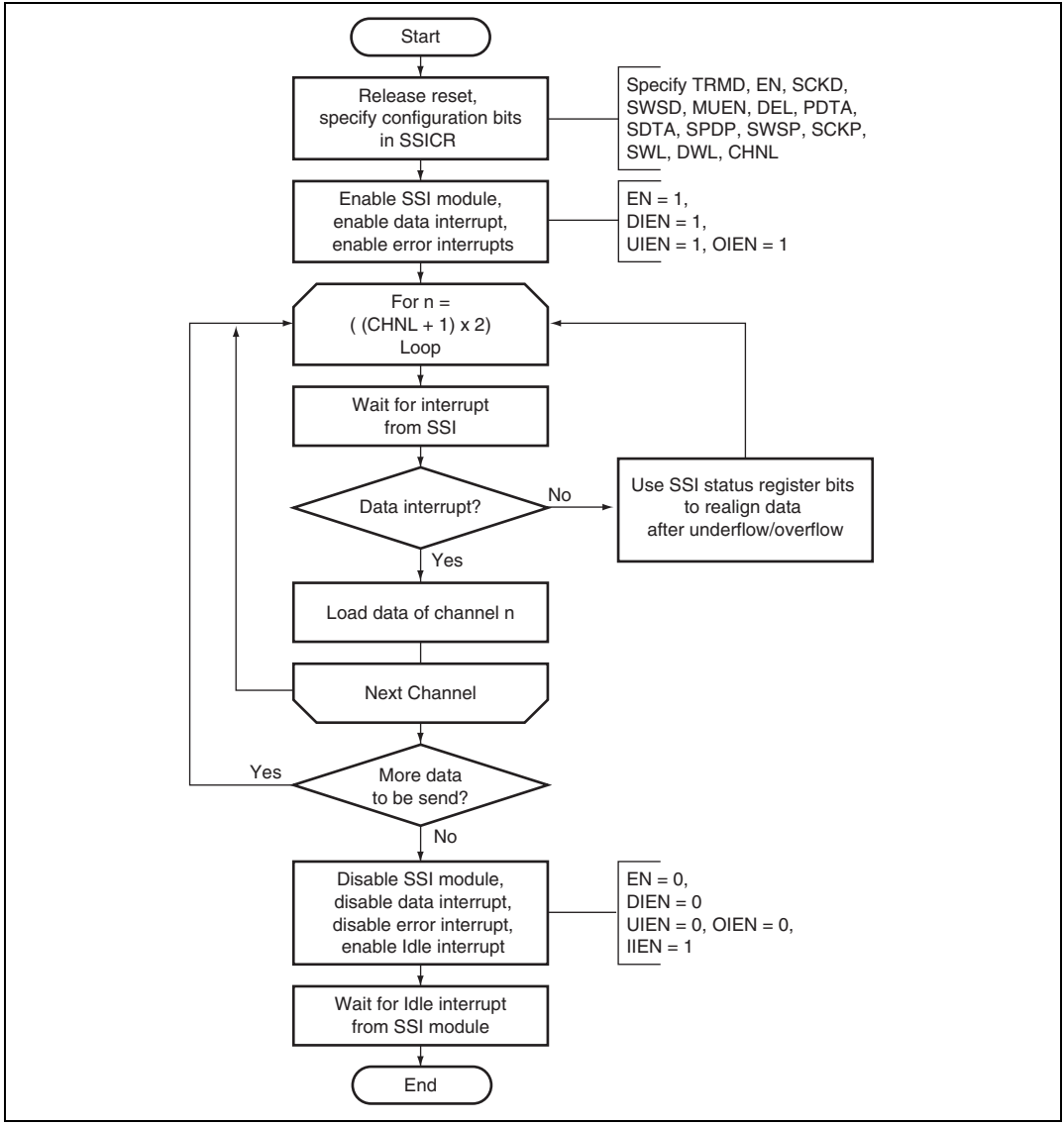


Figure 20.22 Transmission using Interrupt Data Flow Control

As with transmission the reception can be controlled in one of two ways: either DMA or an interrupt driven.

Figures 20.23 and 20.24 show the flow of operation.

When disabling the SSI module, the SSI clock* must be supplied continuously until the module enters in the idle state, which is indicated by the IIRQ bit.

Note: * SCKD = 0: Clock input through the SSI_SCK pin
SCKD = 1: Clock input through the HAC_BIT_CLK pin

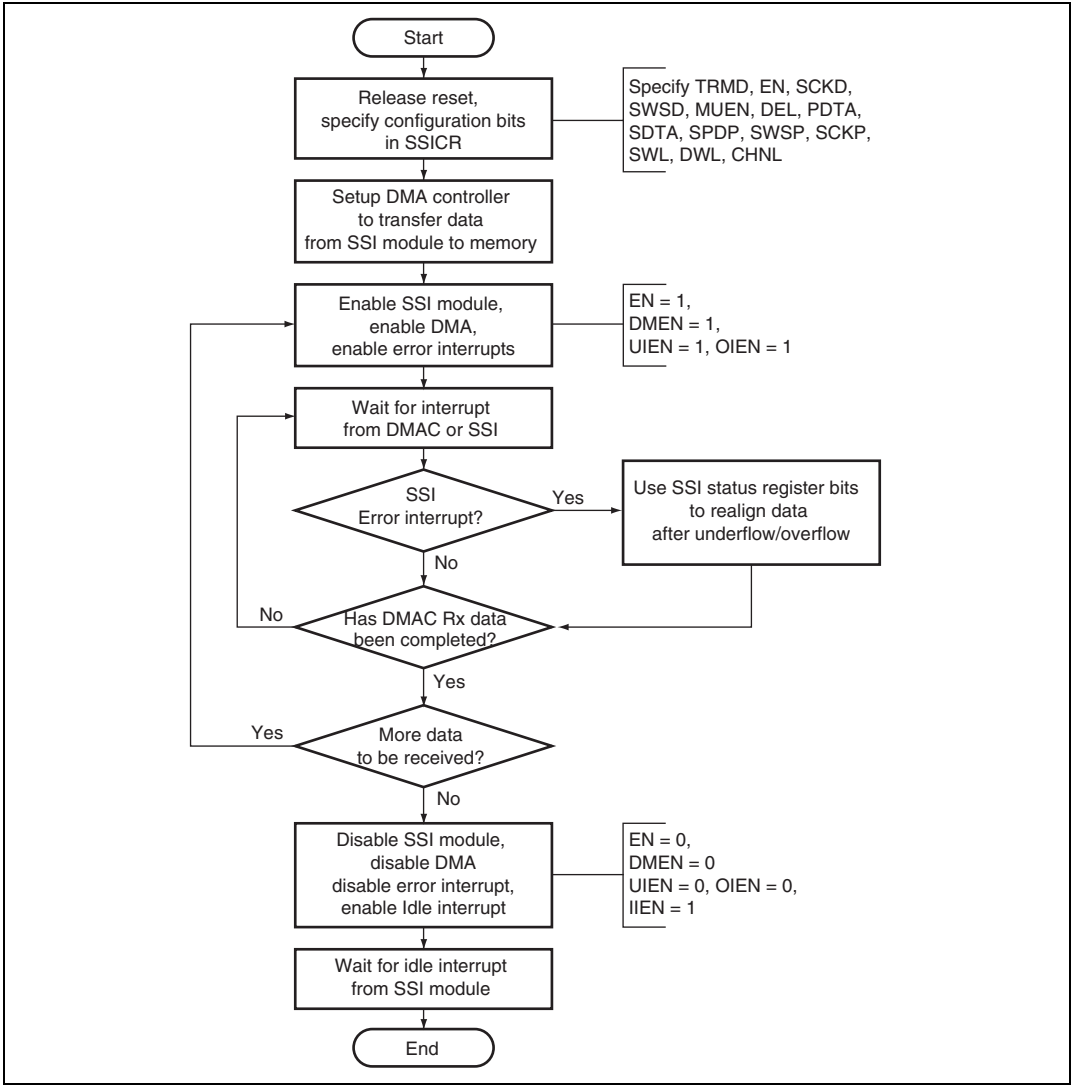


Figure 20.23 Reception using DMA Controller

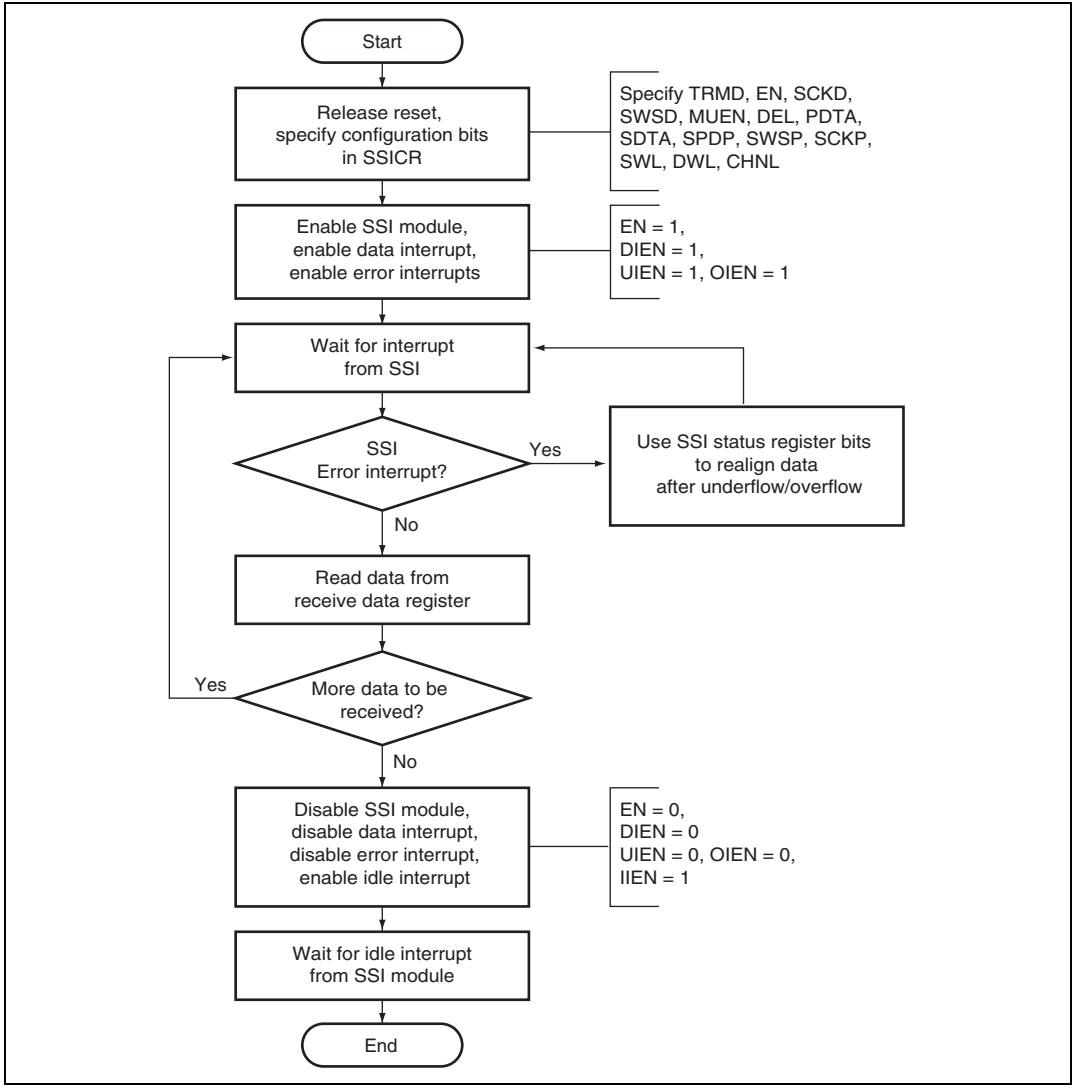


Figure 20.24 Reception using Interrupt Data Flow Control

When an underflow or overflow error condition is met, the CHNO[1:0] and SWNO bits can be used to recover the SSI module to a known status. When an underflow or overflow occurs, the host CPU can read the number of channels and the number of system words to determine what point the serial audio stream has reached. In the transmitter case, the host CPU can skip forward through the data it wants to transmit until it finds the sample data that matches what the SSI module is expecting to transmit next, and so resynchronize with the audio data stream. In the

number of received data, and so resynchronize with the audio data stream.

20.4.7 Serial Clock Control

This function is used to control and select which clock is used for the serial bus interface.

If the serial clock direction is set to input (SCKD = 0), the SSI module is in clock slave mode, then the bit clock that is used in the shift register is derived from the SSI_SCK pin.

If the serial clock direction is set to output (SCKD = 1), the SSI Module is in clock master mode, and the shift register uses the bit clock derived from the HAC_BIT_CLK input pin or its clock divided. This input clock is then divided by the ratio in the serial oversampling clock division ratio (CKDV) bit in SSICR and used as the bit clock in the shift register.

In either case, the SSI_SCK pin output is the same as the bit clock.

20.5 Usage Note

20.5.1 Restrictions when an Overflow Occurs during Receive DMA Operation

If an overflow occurs during receive DMA operation, the module must be reactivated. If an overflow occurs, recover the module according to the following procedure.

1. Ensure an overflow occurs through an overflow error interrupt or overflow error status flag (the OIRQ bit in SSISR).
2. Terminate the DMA by writing 1 to the RDS bit in DMAACR. At this time, confirm the DMA is terminated by reading in the RDS bit (0 can be read).
3. Disable the DMA in the SSI module to halt its operation by writing 0 to the EN bit and DMEN bit in SSICR.
4. Confirm the remaining number of the DMA by reading the DMAARXTCNT to reset the start address of the DMA and number of transfers (MDAARXDAR/DMAARXTCR).
5. Clear the overflow status flag by writing 0 to the OIRQ bit.
6. Reactivate the DMAC by writing 1 to the RDE bit in DMAACR.
7. Reactivate the module by enabling the SSI module and DMA again.

To terminate a slave mode data transfer, after setting SSICR.EN to 0, supply the SSI_WS signal at the active level (alternate continuous input at the high or low level for the system word length) until SSICR.IDST becomes 1 (for five or more system words). If the SSI_WS signal is stopped before IDST becomes 1, the SSI module will not be able to correctly terminate the transfer and will stop in the transfer interrupted state. If SSICR.EN is set to 1 from this state, incorrect data may be transmitted due to restarting from the interrupted state. Note that to restart the data transfer, if the module is in the IDST = 1 state, it can be restarted from the first or second WS edge from EN = 1.

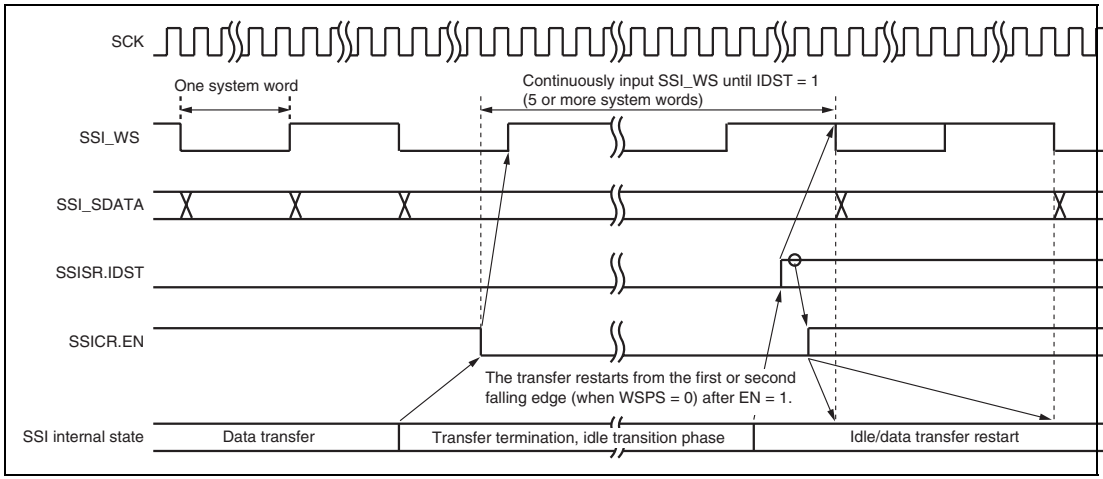


Figure 20.25 Slave Mode SSI Transfer Termination/Restart Timing

The USB Host Controller module supports Open Host Controller Interface (Open HCI) Specification*[2] for the Universal Serial Bus (USB) as well as the Universal Serial Bus specification ver.1.1*[1].

The Open HCI Specification for the USB is a register-level description of Host Controller for the USB.

It is necessary to refer Open HCI specification to develop drivers for this USB Host Controller and hardware.

Abbreviations used in this section are shown below.

HC: host controller

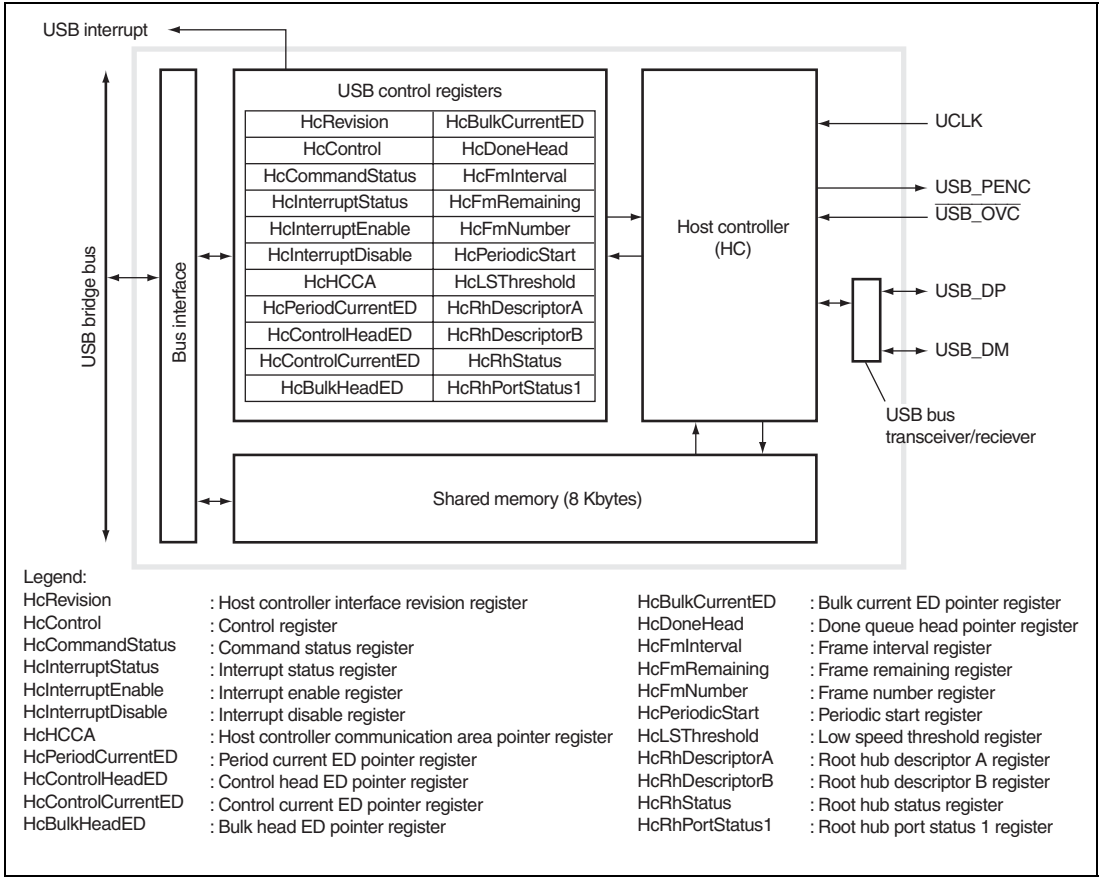
HCD: host controller driver

21.1 Features

The USB host module has the following features.

- Supports the register set compliant with the Open Host Controller Interface (Open HCI) Specification Release 1.0*[2]
- Compatible with the Universal Serial Bus Ver.1.1*[1] Specifications.
- Supports full speed (12 Mbps) mode and low speed (1.5 Mbps) mode as USB transfer speeds
- Supports four transfer modes (control transfer, bulk transfer, interrupt transfer, and isochronous transfer)
- Supports the overcurrent detection circuit
- Supports 127 endpoints at the maximum
- One USB port
- On-chip 8-Kbyte SRAM used as shared memory which is defined in the OpenHCI specification

Notes: 1. Moreover, refer to the USB Host Electrical Characteristics section for the electrical characteristics of USB Host.
2. Part of registers is not supported. For details, see section 21.3, Register Descriptions and section 21.6, Restrictions on HcRhDescriptorA.



Legend:

HcRevision	: Host controller interface revision register	HcBulkCurrentED	: Bulk current ED pointer register
HcControl	: Control register	HcDoneHead	: Done queue head pointer register
HcCommandStatus	: Command status register	HcFmInterval	: Frame interval register
HcInterruptStatus	: Interrupt status register	HcFmRemaining	: Frame remaining register
HcInterruptEnable	: Interrupt enable register	HcFmNumber	: Frame number register
HcInterruptDisable	: Interrupt disable register	HcPeriodicStart	: Periodic start register
HcHCCA	: Host controller communication area pointer register	HcLSThreshold	: Low speed threshold register
HcPeriodCurrentED	: Period current ED pointer register	HcRhDescriptorA	: Root hub descriptor A register
HcControlHeadED	: Control head ED pointer register	HcRhDescriptorB	: Root hub descriptor B register
HcControlCurrentED	: Control current ED pointer register	HcRhStatus	: Root hub status register
HcBulkHeadED	: Bulk head ED pointer register	HcRhPortStatus1	: Root hub port status 1 register

Figure 21.1 Block Diagram of USB Host Module

Table 21.1 shows the pin configuration of the USB host module.

Table 21.1 Pin Configuration

Pin Name	Abbreviation	I/O	Function
D+	USB_DP	Input/Output	Root hub port D+ transceiver pin
D−	USB_DM	Input/Output	Root hub port D− transceiver pin
Power enable	USB_PENC	Output	Power-on enable control pin for root hub port
Overcurrent	$\overline{\text{USB\_OVC}}$	Input	Overcurrent detection pin for root hub port
UCLK	UCLK	Input	USB operating clock (input clock of 48.0000 MHz ±0.05%)

21.3 Register Descriptions

The USB host module has the following registers. These registers can be accessed only in longwords. Access in bytes or words is prohibited. For details on register addresses and register states during each process, see section 32, List of Registers. When a clock is not input from the UCLK pin, these registers cannot be accessed.

Register Name	Abbreviation	R/W	P4 Address	Area 7 Address	Size	Sync Clock
Host controller interface revision register	HcRevision	R	H'FE34 0000	H'1E34 0000	32	Pck
Control register	HcControl	R/W	H'FE34 0004	H'1E34 0004	32	Pck
Command status register	HcCommandStatus	R/W	H'FE34 0008	H'1E34 0008	32	Pck
Interrupt status register	HcInterruptStatus	R/W	H'FE34 000C	H'1E34 000C	32	Pck
Interrupt enable register	HcInterruptEnable	R/W	H'FE34 0010	H'1E34 0010	32	Pck
Interrupt disable register	HcInterruptDisable	R/W	H'FE34 0014	H'1E34 0014	32	Pck
Host controller communication area pointer register	HcHCCA	R/W	H'FE34 0018	H'1E34 0018	32	Pck
Period current ED pointer register	HcPeriodCurrentED	R	H'FE34 001C	H'1E34 001C	32	Pck
Control head ED pointer register	HcControlHeadED	R/W	H'FE34 0020	H'1E34 0020	32	Pck
Control current ED pointer register	HcControlCurrentED	R/W	H'FE34 0024	H'1E34 0024	32	Pck
Bulk head ED pointer register	HcBulkHeadED	R/W	H'FE34 0028	H'1E34 0028	32	Pck
Bulk current ED pointer register	HcBulkCurrentED	R/W	H'FE34 002C	H'1E34 002C	32	Pck
Done queue head pointer register	HcDoneHead	R	H'FE34 0030	H'1E34 0030	32	Pck
Frame interval register	HcFmInterval	R/W	H'FE34 0034	H'1E34 0034	32	Pck
Frame remaining register	HcFmRemaining	R	H'FE34 0038	H'1E34 0038	32	Pck
Frame number register	HcFmNumber	R	H'FE34 003C	H'1E34 003C	32	Pck
Periodic start register	HcPeriodicStart	R/W	H'FE34 0040	H'1E34 0040	32	Pck
Low speed threshold register	HcLSThreshold	R/W	H'FE34 0044	H'1E34 0044	32	Pck
Root hub descriptor A register	HcRhDescriptorA	R/W	H'FE34 0048	H'1E34 0048	32	Pck
Root hub descriptor B register	HcRhDescriptorB	R/W	H'FE34 004C	H'1E34 004C	32	Pck
Root hub status register	HcRhStatus	R/W	H'FE34 0050	H'1E34 0050	32	Pck
Root hub port status 1 register	HcRhPortStatus1	R/W	H'FE34 0054	H'1E34 0054	32	Pck
	Shared memory start	R/W	H'FE34 1000	H'1E34 1000	32	Pck
	Shared memory end	R/W	H'FE34 2FFF	H'1E34 2FFF	32	Pck

Register Name	Abbrev.	Power-on Reset by RESET Pin/WDT/H-UDI	Manual Reset by RESET Pin/WDT/ Multiple Exception	Sleep by Sleep Instruction/ Deep Sleep	Standby by Hardware	Standby by Software/ Each Module
Host controller interface revision register	HcRevision	H'0000 0010	H'0000 0010	Retained	*	Retained
Control register	HcControl	H'0000 0000	H'0000 0000	Retained		Retained
Command status register	HcCommandStatus	H'0000 0000	H'0000 0000	Retained		Retained
Interrupt status register	HcInterruptStatus	H'0000 0000	H'0000 0000	Retained		Retained
Interrupt enable register	HcInterruptEnable	H'0000 0000	H'0000 0000	Retained		Retained
Interrupt disable register	HcInterruptDisable	H'0000 0000	H'0000 0000	Retained		Retained
Host controller communication area pointer register	HcHCCA	H'0000 0000	H'0000 0000	Retained		Retained
Period current ED pointer register	HcPeriodCurrentED	H'0000 0000	H'0000 0000	Retained		Retained
Control head ED pointer register	HcControlHeadED	H'0000 0000	H'0000 0000	Retained		Retained
Control current ED pointer register	HcControlCurrentED	H'0000 0000	H'0000 0000	Retained		Retained
Bulk head ED pointer register	HcBulkHeadED	H'0000 0000	H'0000 0000	Retained		Retained
Bulk current ED pointer register	HcBulkCurrentED	H'0000 0000	H'0000 0000	Retained		Retained
Done queue head pointer register	HcDoneHead	H'0000 0000	H'0000 0000	Retained		Retained
Frame interval register	HcFmInterval	H'0000 2EDF	H'0000 2EDF	Retained		Retained
Frame remaining register	HcFmRemaining	H'0000 0000	H'0000 0000	Retained		Retained
Frame number register	HcFmNumber	H'0000 0000	H'0000 0000	Retained		Retained
Periodic start register	HcPeriodicStart	H'0000 0000	H'0000 0000	Retained		Retained
Low speed threshold register	HcLSThreshold	H'0000 0628	H'0000 0628	Retained		Retained
Root hub descriptor A register	HcRhDescriptorA	H'0200 1202	H'0200 1202	Retained		Retained
Root hub descriptor B register	HcRhDescriptorB	H'0000 0000	H'0000 0000	Retained		Retained
Root hub status register	HcRhStatus	H'0000 0000	H'0000 0000	Retained		Retained
Root hub port status 1 register	HcRhPortStatus1	H'0000 0100	H'0000 0100	Retained		Retained
	Shared memory start	Undefined	Undefined	Retained		Retained
	Shared memory end	Undefined	Undefined	Retained		Retained

Note: * After exiting hardware standby mode, this LSI enters the power-on reset state by the RESET pin.

HcRevision stores the version number of the host controller interface specifications.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
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
	-	-	-	-	-	-	-	-	REV							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	All 0	R	Reserved These bits are always read as 0. Always write 0 to these bits.
7 to 0	REV	H'10	R	OpenHCI Revision H'10: Indicates the OpenHCI specification version 1.0.

21.3.2 Control Register (HcControl)

HcControl defines the operation mode for HC. The bits of this register, except for HCFS and RWC, are modified only by HCD.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
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
	-	-	-	-	-	RWE	RWC	IR	HCFS1	HCFS0	BLE	CLE	IE	PLE	CBSR1	CBSR0
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/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

31 to 11	—	All 0	R	Reserved These bits are always read as 0. Always write 0 to these bits.
10	RWE	0	R/W	Remote Wakeup Enable This bit is used by HCD to enable/disable the remote wakeup function simultaneously with the detection of an upstream resume signal. 0: Remote wakeup function is not supported 1: Remote wakeup function is supported This function is not supported. Always write 0 to this bit.
9	RWC	0	R/W	Remote Wakeup Connected This bit indicates whether or not HC supports a remote wakeup signal. When the remote wakeup is supported and used in the system, HC must set this bit during POST in the system firmware. HC clears the bit simultaneously with the hardware reset, but not with the software reset. 0: Remote wakeup signal is not supported 1: Remote wakeup signal is supported This function is not supported. Always write 0 to this bit.
8	IR	0	R/W	Interrupt Routing This bit determines the routing of interrupts generated by the event registered in HcInterruptStatus. HCD clears this bit to 0 simultaneously with the hardware reset, but not with 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 1: Setting prohibited

7	HCFS1	0	R/W	Host Controller Functional State
6	HCFS0	0	R/W	HCD determines whether HC has started to route SOF after having read the SF bit of HcInterruptStatus. This bit can be changed by HC only in the USB suspend state. HC can change from the USB suspend state to the USB resume state after having detected the resume signal from the downstream port. In HC, USB suspend is entered after the software reset like USB reset is entered after the hardware reset. The former resets the root hub. 00: USB reset state 01: USB resume state 10: USB operation state 11: USB suspend state
5	BLE	0	R/W	Bulk List Enable This bit is set to enable the processing of the bulk list in the next frame. HC checks this bit when the processing of the list has been determined. When this bit is 0 (disabling), HCD can modify the list. When HcBulkCurrentED indicates ED to be deleted, HCD should increment the pointer by updating HcBulkCurrentED before re-enabling the list processing. 0: Bulk list processing is disabled in the next frame 1: Bulk list processing is enabled in the next frame
4	CLE	0	R/W	Control List Enable This bit is set to 1 to enable processing of the control list in the next frame. When cleared to 0 by HCD, the control list is not processed after the next SOF. HC must check this bit whenever the list is to be processed. When this bit is 0 (disabling), HCD can modify the list. When HcControlCurrentED indicates ED to be deleted, HCD should increment the pointer by updating HcControlCurrentED before re-enabling list processing. 0: Control list processing is disabled in the next frame 1: Control list processing is enabled in the next frame

3	IE	0	R/W	Isochronous Enable 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). When this bit is 1 (enabling), HC continues to process ED. When this bit is 0 (disabling), HC stops the processing of the periodic list (currently including 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: Isochronous list processing is disabled in the next frame 1: Isochronous list processing is enabled in the next frame
2	PLE	0	R/W	Periodic List Enable This bit is set to 1 to enable processing of the periodic list. When HCD clears it to 0, no periodic list processing is carried out after next SOF. HC must check this bit before HC starts to process the list. 0: Periodic list processing is disabled in the next frame 1: Periodic list processing is enabled in the next frame
1	CBSR1	0	R/W	Control Bulk Service Ratio
0	CBSR0	0	R/W	This specifies the service ratio between Control and Bulk EDs. Before processing any of the nonperiodic lists, HC must compare the ratio specified with its internal count on how many nonempty Control EDs have been processed, in determining whether to continue serving another Control ED or switching to Bulk EDs. The internal count will be retained when crossing the frame boundary. In case of reset, HCD must restore this value. 00: The ratio of Control ED to Bulk ED is 1:1 01: The ratio of Control ED to Bulk ED is 2:1 10: The ratio of Control ED to Bulk ED is 3:1 11: The ratio of Control ED to Bulk ED is 4:1

HcCommandStatus indicates the current status of HC. HC reads this register to receive a command issued by HCD. HCD sets each bit by writing 1 and HC clears it by writing 0.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	SOC1	SOC0
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
	-	-	-	-	-	-	-	-	-	-	-	-	OCR	BLF	CLF	HCR
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/W	R/W	R/W	R/W

Bits	Bit Name	Initial Value	R/W	Description
31 to 18	—	All 0	R	Reserved These bits are always read as 0. Always write 0 to these bits.
17	SOC1	0	R	Scheduling Overrun Count
16	SOC0	0	R	These bits are incremented in each SchedulingOverrun error. These bits are initialized to B'00 and wrap around at B'11. These bits are incremented when scheduling overrun is detected even though the SO bit in HcInterruptStatus is set. These bits are used by HCD to monitor any persistent scheduling problem.
15 to 4	—	All 0	R	Reserved These bits are always read as 0. Always write 0 to these bits.
3	OCR	0	R/W	Ownership Change Request This bit is set to 1 by OS HCD to request a change of control of HC. When this bit is1, HC sets the OC bit in the HcInterruptStatus. After a change, this bit is cleared to 0 and remains 0 until the next request from OS HCD. 0: Do not request the change of the control of HC 1: Request the change of the control of HC

2	BLF	0	R/W	Bulk List Filled This bit is used to indicate whether there are any TDs in the list. HCD sets this bit to 1 for the list when adding TD to ED. When starting to process the head of the list, HC checks BF. As long as BLF is 0, HC does not start to process the list. When BLF is 1, HC starts to process the list to set BF through 0. When HC finds TD in the list, HC sets BLF to 1. When TD is not in the list and HCD does not set BLF, HC completes processing the list. When list processing is stopped, BLF remains 0. 0: TD is not found in the Bulk list 1: TD is found in the Bulk list
1	CLF	0	R/W	Control List Filled This bit is used to indicate whether there are any TDs in the control list. HCD sets this bit when adding TD to ED in the control list. When HC starts to process the head of the control list, it checks CLF. As long as CLF is 0, HC does not start to process the control list. When CF is 1, HC starts to process the control list and clears CLF to 0. When HC finds TD in the list, HC sets CLF to 1. When TD is not found in the control list and HCD does not set CLF, HC completes the processing of the control list. When control list processing is stopped, CLF remains 0. 0: TD is not found in the Control list 1: TD is found in the Control list

0	HCR	0	R/W	Host Controller Reset HCD sets this bit to 1 to start software reset of HC. The HC changes to the USB suspend state in which most operational registers are reset, regardless of which functional state the HC is in. However, access to the InterrupRouting field in HcControl or access without host bus is allowed. HC clears this bit to 0 upon completion of the reset operation. The reset operation must be completed within 10μs. Setting this bit to 1 does not either reset the root hub or issue the subsequent reset signal to the downstream port. 0: Software reset of HC is cancelled 1: Software reset of HC is started

21.3.4 Interrupt Status Register (HcInterruptStatus)

HcInterruptStatus indicates the status of various events that cause hardware interrupts. To generate an interrupt, HC sets the corresponding bit in this register to 1 when HcInterruptEnable enables a hardware interrupt with HcInterruptEnable.MIE = 1. HCD can clear a bit to 0 by writing 1 to release the interrupt status. However, HCD cannot set any of these bits to 1.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	OC	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	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
	-	-	-	-	-	-	-	-	-	RHSC	FNO	UE	RD	SF	WDH	SO
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/W	R/W	R/W	R/W	R/W	R/W	R/W

31	—	0	R	Reserved These bits are always read as 0. Always write 0 to these bits.
30	OC	0	R/W	Ownership Change HC sets this bit to 1 when HCD sets the OCR bit in the HcCommandStatus register. This event generates a system management interrupt (SMI) at once when not masked. When there is no SMI pin, this bit is set to 0. 0: No change requested 1: There is an HC change request
29 to 7	—	All 0	R	Reserved These bits are always read as 0. Always write 0 to these bits.
6	RHSC	0	R/W	Root Hub Status Change HC sets this bit to 1 when the contents of either HcRhStatus or any HcRhPortStatus1 have changed. 0: HcRhStatus or HcRhPortStatus contents are not changed 1: HcRhStatus or HcRhPortStatus contents are changed
5	FNO	0	R/W	Frame Number Overflow HC sets this bit to 1 after MSB bit in HcFmNumber changes from 0 to 1 or from 1 to 0 and the HccaFrameNumber is updated. 0: No change 1: MSB in HcFmNumber register is changed and HccaFrameNumber is updated
4	UE	0	R/W	Unrecoverable Error HC sets this bit to 1 when detecting a system error irrelevant to USB. HCD clears this bit after HC is reset. 0: Detected no system error irrelevant to USB 1: Detected a system error irrelevant to USB

3	RD	0	R/W	Resume Detected HC sets this bit to 1 when detecting the transmission of a resume signal by a USB device. This bit is not set when HCD sets USB resume state. 0: A USB device is not asserting a resume signal 1: A USB device is asserting a resume signal
2	SF	0	R/W	Start of Frame HC sets this bit to 1 when each frame is started and after the HccaFrameNumber is updated. HC simultaneously generates the SOF token. 0: Frame is not started or HccaFrameNumber is not updated 1: Frame is started and HccaFrameNumber is updated
1	WDH	0	R/W	Writeback Done Head HC sets this bit to 1 immediately after writing 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: HccaDoneHead is retained 1: The value in HcDoneHead is written to HccaDonehead
0	SO	0	R/W	Scheduling Overrun HC sets this bit to 1 when the USB schedule has overrun in the current frame after HccaFrameNumber is updated. Scheduling overrun also increments the SOC bit in HcCommandStatus. 0: The USB schedule has not overrun in the current frame 1: The USB schedule has overrun in the current frame

21.3.5 Interrupt Enable Register (HcInterruptEnable)

Each enable bit in HcInterruptEnable corresponds to the related hardware interrupt bit in HcInterruptStatus. A hardware interrupt is generated when bits in HcInterruptStatus are set to1, the corresponding bits in HcInterruptEnable are set to 1, and HcInterruptEnable.MIE = 1.

Writing 1 to a bit in this register sets the corresponding bit to 1, while writing 0 does not clear the bit to 0 but leaves it unchanged. Reading this register will return the current value of this register.

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	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
	-	-	-	-	-	-	-	-	-	RHSC	FNO	UE	RD	SF	WDH	SO
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/W	R/W	R/W	R/W	R/W	R/W	R/W

Bits	Bit Name	Initial Value	R/W	Description
31	MIE	0	R/W	Master Interrupt Enable Setting this bit to 1 will enable an interrupt generation by the event specified in another bit in this register. HCD uses this bit to enable the master interrupt. To use HCD for interrupt detection, use the USB bit of the INTC. 0: Operation is not affected 1: Interrupt generation is enabled
30	OC	0	R/W	Ownership Change 0: Operation is not affected 1: Interrupt generation to change HC control is enabled
29 to 7	—	All 0	R	Reserved These bits are always read as 0. Always write 0 to this bit.
6	RHSC	0	R/W	Root Hub Status Change 0: Operation is not affected 1: Interrupt generation due to Root Hub Status Change is enabled
5	FNO	0	R/W	Frame Number Overflow 0: Operation is not affected 1: Interrupt generation due to Frame Number Overflow is enabled
4	UE	0	R/W	Unrecoverable Error 0: Operation is not affected 1: Interrupt generation due to unrecoverable error is enabled

3	RD	0	R/W	Resume Detected 0: Operation is not affected 1: Interrupt generation due to Resume Detected is enabled
2	SF	0	R/W	Start of Frame 0: Operation is not affected 1: Interrupt generation due to Start of Frame is enabled
1	WDH	0	R/W	Writeback Done Head 0: Operation is not affected 1: Interrupt generation for HcDoneHead Writeback is enabled
0	SO	0	R/W	Scheduling Overrun 0: Operation is not affected 1: Interrupt generation due to Scheduling Overrun is enabled

21.3.6 Interrupt Disable Register (HcInterruptDisable)

Each disable bit in the HcInterruptDisable corresponds to the related interrupt bit in HcInterruptStatus. Writing 1 to a bit in the HcInterruptDisable clears the corresponding bit in HcInterruptEnable to 0. Writing 0 to a bit leaves the corresponding bit in HcInterruptEnable unchanged. Reading this register will return the current value of HcInterruptEnable.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MIE	OC	-	-	-	-	-	-	-	-	-	-	-	-	-	-
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	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
	-	-	-	-	-	-	-	-	-	RHSC	FNO	UE	RD	SF	WDH	SO
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/W	R/W	R/W	R/W	R/W	R/W	R/W

31	MIE	0	R/W	Master Interrupt Enable 0: Operation is not affected 1: Interrupt generation for other events is disabled
30	OC	0	R/W	Ownership Change 0: Operation is not affected 1: Interrupt generation for HC control change is disabled
29 to 7	—	All 0	R	Reserved These bits are always read as 0. Always write 0 to this bit.
6	RHSC	0	R/W	Root Hub Status Change 0: Operation is not affected 1: Interrupt generation due to Root Hub Status Change is disabled
5	FNO	0	R/W	Frame Number Overflow 0: Operation is not affected 1: Interrupt generation due to Frame Number Overflow is disabled
4	UE	0	R/W	Unrecoverable Error 0: Operation is not affected 1: Interrupt generation due to unrecoverable error is disabled
3	RD	0	R/W	Resume Detected 0: Operation is not affected 1: Interrupt generation due to Resume Detected is disabled
2	SF	0	R/W	Start of Frame 0: Operation is not affected 1: Interrupt generation due to Start of Frame is disabled
1	WDH	0	R/W	Writeback Done Head 0: Operation is not affected 1: Interrupt generation due to HcDonehead Writeback is disabled
0	SO	0	R/W	Scheduling Overrun 0: Operation is not affected 1: Interrupt generation due to Scheduling Overrun is disabled

HcHCCA stores physical addresses of the host controller communication area (HCCA). HCD determines the alignment restrictions by writing 1 to all bits in HcHCCA and by reading the contents of HcHCCA. Alignment is evaluated by checking the number of 0s in the low-order bits. The minimum alignment is 256 bytes. Consequently, bits 0 to 7 must always return 0 when they are read. This area is used to retain the control structures and interrupt table that are accessed by HC and HCD.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	HCCA															
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
	HCCA								-	-	-	-	-	-	-	-
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	R	R	R	R	R	R	R

Bits	Bit Name	Initial Value	R/W	Description
31 to 8	HCCA	All 0	R/W	Host Controller Communication Area Address These bits store the base address of HCCA.
7 to 0	—	All 0	R	Reserved These bits are always read as 0. Always write 0 to this bit.

21.3.8 Period Current ED Pointer Register (HcPeriodCurrentED)

HcPeriodCurrentED stores the address of the isochronous ED or interrupt ED to be processed in the current frame.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	PCED															
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
	PCED												-	-	-	-
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

31 to 4	PCED	All 0	R	Period Current ED Address These bits store the start address of the periodic list processed in the current frame.
3 to 0	—	All 0	R	Reserved These bits are always read as 0. Always write 0 to this bit.

21.3.9 Control Head ED Pointer Register (HcControlHeadED)

HcControlHeadED stores the address of the first ED to be processed in the control list.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CHED															
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
	CHED												-	-	-	-
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

Bit	Bit Name	Initial Value	R/W	Description
31 to 4	CHED	All 0	R/W	Control Head ED Address These bits store the start address of the control list.
3 to 0	—	All 0	R	Reserved These bits are always read as 0. Always write 0 to this bit.

21.3.10 Control Current ED Pointer Register (HcControlCurrentED)

HcControlCurrentED stores the address of the ED to be processed in the current frame in the control list.

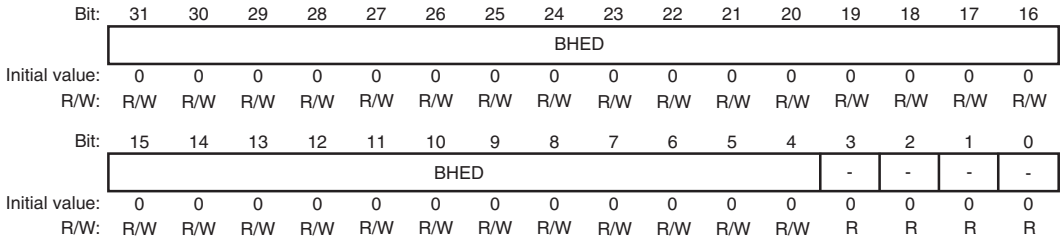
Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CCED															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CCED												-	-	-	-
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

31 to 4	CCED	All 0	R/W	Control Current ED Address These bits store the start address of the control list to be processed in the current frame.
3 to 0	—	All 0	R	Reserved These bits are always read as 0. Always write 0 to this bit.

21.3.11 Bulk Head ED Pointer Register (HcBulkHeadED)

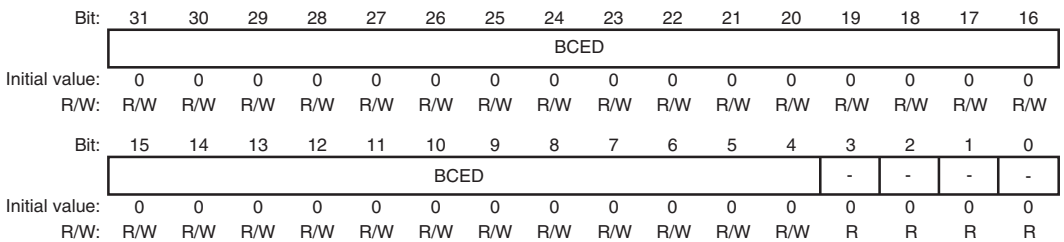
HcBulkHeadED stores the address of the first ED to be processed in the bulk list.



Bit	Bit Name	Initial Value	R/W	Description
31 to 4	BHED	All 0	R/W	Bulk Head ED Address These bits store the start address of the bulk list.
3 to 0	—	All 0	R	Reserved These bits are always read as 0. Always write 0 to this bit.

21.3.12 Bulk Current ED Pointer Register (HcBulkCurrentED)

HcBulkCurrentED stores the address of the ED to be processed in the current frame in the bulk list.



31 to 4	BCED	All 0	R/W	Bulk Current ED Address These bits store the start address of the bulk list to be processed in the current frame.
3 to 0	—	All 0	R	Reserved These bits are always read as 0. Always write 0 to this bit.

21.3.13 Done Queue Head Pointer Register (HcDoneHead)

HcDoneHead stores the address of the last processed TD in the done queue.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DH															
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
	DH												-	-	-	-
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	Bit Name	Initial Value	R/W	Description
31 to 4	DH	All 0	R	Done Queue Pointer Address These bits store the address of the last processed TD.
3 to 0	—	All 0	R	Reserved These bits are always read as 0.

HcFmInterval consists of a 14-bit FI value that indicates the frame bit time interval (interval between two consecutive SOFs) and a 15-bit FSMPS value that indicates the maximum packet size that is transmitted and received at full speed by HC without causing scheduling overrun. HCD minutely adjusts the frame interval by updating the value in each SOF.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	FIT	FSMPS														
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
	-	-	FI													
Initial value:	0	0	1	0	1	1	1	0	1	1	0	1	1	1	1	1
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	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31	FIT	0	R/W	Frame Interval Toggle This bit is toggled by HCD whenever it loads a new value into FI.
30 to 16	FSMPS	All 0	R/W	Largest Data Packet This bit specifies a value to be loaded into the Largest Data Packet Counter at the beginning of each frame. The counter value represents the maximum amount of data in bits transmittable/receivable by HC in a single transaction at any given time without causing scheduling overrun. The bit value is calculated by HCD.
15, 14	—	All 0	R	Reserved These bits are always read as 0. Always write 0 to this bit.
13 to 0	FI	H'2EDF	R/W	Frame Interval This bit specifies the interval between two consecutive SOFs in bit times. The nominal value is set to be 11999. HCD should store the current value of this bit before resetting HC. Setting the HCR bit in HcCommandStatus will have the HC reset this bit to its nominal value. HCD may choose to restore the stored value upon the completion of the reset sequence.

HcFmRemaining contains a 14-bit down-counter that indicates bit times remaining until the current frame is complete. HcFmRemaining is a read-only register. Operation is not guaranteed when writing.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	FRT	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
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
	-	-	FR													
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	Bit Name	Initial Value	R/W	Description
31	FRT	0	R	Frame Remaining Toggle When FR reaches H'0000, this bit stores the FIT value in HcFmInterval. This bit is used by HCD for the synchronization between the FI and FR bits.
30 to 14	—	All 0	R	Reserved These bits are always read as 0.
13 to 0	FR	All 0	R	Frame Bit Times Remaining This counter is decremented at each bit time. When it reaches zero, it is re-set to the FI value in HcFmInterval that is loaded at the next bit time boundary. When entering the USB operational state, HC re-loads the contents of the FI bit in HcFmInterval to this counter and uses the updated value from the next SOF.

HcFmNumber contains a 16-bit counter. This register is referenced for the timing between events occurring in HC and HCD. HCD uses 16-bit value specified in this register and generates a 32-bit frame number without requiring frequent access to the register.

HcFmNumber is a read-only register. Operation is not guaranteed when writing.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
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
	FN															
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	Bit Name	Initial Value	R/W	Description
31 to 16	—	All 0	R	Reserved These bits are always read as 0.
15 to 0	FN	All 0	R	Frame Number This is incremented when HcFmRemaining is re-loaded. It will be rolled over to H'0 after H'FFFF. When HC enters the USB operational state, this will be automatically incremented. HC increments the FN at each frame boundary and sends a SOF. Then HC writes the FN contents to HCCA before reading the first ED in that Frame. After writing to HCCA, the HC sets HcInterruptStatus.SF = 1.

21.3.17 Periodic Start Register (HcPeriodicStart)

HcPeriodicStart indicates the earliest time when HC should start to process the periodic list.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
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
	-	-	PS													
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/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

31 to 14	—	All 0	R	Reserved These bits are always read as 0. Always write 0 to this bit.
13 to 0	PS	All 0	R/W	Periodic Start After a hardware reset, this bit is cleared. Then, HCD sets this bit to 1 during the HC initialization. The value is calculated roughly as 10% subtracted from the HcFmInterval value. When HcFmRemaining reaches the specified value, processing of the periodic lists will have priority over Control/Bulk processing. HC will therefore start processing the Interrupt list after completing the current Control or Bulk transaction that is in progress.

21.3.18 Low Speed Threshold Register (HcLSThreshold)

HcLSThreshold stores an 11-bit LST value that is used by HC to determine whether or not to authorize the transfer of the LS packet of up to 8 bytes, before EOF. HC and HCD cannot change this value.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
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
	-	-	-	-	LST											
Initial value:	0	0	0	0	0	1	1	0	0	0	1	0	1	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

Bit	Bit Name	Initial Value	R/W	Description
31 to 12	—	All 0	R	Reserved These bits are always read as 0. Always write 0 to this bit.
11 to 0	LST	H'628	R/W	LS Threshold This field contains a value which is compared to the FR bit prior to initiating a Low Speed transaction. The transaction is started only if the value of the FR bit is equivalent to or larger than that of this bit. HCD calculates the value of this bit taking transmission and setup overhead into consideration.

HcRhDescriptorA is the first register of two describing the characteristics of the root hub. The descriptor length (11), descriptor type (TBD), and hub controller current (0) bits in the hub Class Descriptor are emulated by HCD. All other bits are allocated in HcRhDescriptorA and HcRhDescriptorB.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	POTPGT								-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	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	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	NOCP	OCPM	DT	NPS	PSM	NDP							
Initial value:	0	0	0	1	0	0	1	0	0	0	0	0	0	0	1	0
R/W:	R	R	R	R/W	R/W	R	R/W	R/W	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	POTPGT	H'02	R/W	Power On To Power Good Time This bit specifies the HCD wait time before accessing a powered-on port of the root hub. It is implementation-specific. The unit of time is 2 ms. The duration is calculated as POTPGT × 2 ms.
23 to 13	—	All 0	R	Reserved These bits are always read as 0. Always write 0 to this bit.
12	NOCP	1	R/W	No Overcurrent Protection This bit describes the overcurrent condition reporting method for root hub ports. When this bit is cleared to 0, the OCPM bit specifies either global or per-port reporting. 0: Overcurrent status info is collected 1: Overcurrent protection is not supported Note: There are restrictions on this bit. For details, see section 21.6, Restrictions on HcRhDescriptorA.

11	OCPM	0	R/W	Overcurrent Protection Mode This bit specifies the overcurrent condition reporting method for root hub ports. At reset, this bit should reflect the same mode as in PSM. This bit is valid only when the NOCP bit is cleared to 0. 0: Overcurrent status info is collected for all ports 1: Overcurrent status info is collected for ports individually Note: There are restrictions on this bit. For details, see section 21.6, Restrictions on HcRhDescriptorA.
10	DT	1	R	Device Type This bit specifies that the root hub is not a compound device. Always write 0 to this bit. 0: Root hub is not a compound device 1: Setting prohibited
9	NPS	1	R/W	No Power Switching This bit selects whether ports are power switched or ports are always powered. It is implementation specific. When this bit is cleared to 0, the PSM bit specifies global or per-port switching. 0: Ports are power switched 1: Ports are always powered on when HC is powered on Note: Since the initial value is 1, this bit should be cleared to 0 in advance (0 is written by HCD) to enable power switching.

8	PSM	0	R/W	Power Switching Mode This bit specifies the root hub port power switching control method. It is implementation specific. This bit is valid only when the NPS bit is cleared to 0. 0: All ports are powered at the same time. 1: Each port is powered individually. This mode allows port power to be controlled by either the global switch or per-port switching. When HcRhDescriptorB.PPCM = H'0002, the port responds only to port power commands (Set/ClearPortPower). When the port mask is cleared, then the port is controlled only by the global power switch (Set/ClearGlobalPower).
7 to 0	NDP	H'02	R	Number of Downstream Ports These bits specify the number of downstream ports supported by the root hub. It is implementation-specific. The value of these bits is H'2. However, this LSI supports only 1 port for downstream.

21.3.20 Root Hub Descriptor B Register (HcRhDescriptorB)

HcRhDescriptorB is the second register of two describing the characteristics of the root hub. Set the bits of this register on initial setup so as to cater for system implementation.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	PPCM															
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
	DR															
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

31 to 16	PPCM	All 0	R/W	Port Power Control Mask Each bit indicates whether or not the corresponding port is affected by a global power control command when HcRhDescriptorA.PSM = 1. When a bit is set to 1, the corresponding port's power state is affected only by per-port power control (Set/ClearPortPower). When a bit is cleared to 0, the corresponding port is controlled by the global power switch (Set/ClearGlobalPower). If the device is configured to global switching mode (PSM = 0), settings in these bits are not valid. H'0000: Port 1 is affected by global power control H'0002: Port 1 is masked from global power control Other than above: Setting prohibited Bit 16: Reserved Bit 17: Ganged-power mask on Port 1 Bit 18: Ganged-power mask on Port 2 ... Bit 31: Ganged-power mask on Port 15 Note: Before setting these bits, clear HcRhDescriptorA.NPS = 0 and HcRhPortStatus.PPS = 0 to turn off the power of all ports.
15 to 0	DR	All 0	R/W	Device Removable Each bit is used only for a port of the root hub. When a bit is cleared to 0, the attached device is removable from the corresponding port. When a bit is set to 1, the attached device is not removable from the corresponding port. H'0000: Device connected to port 1 can be removed H'0002: Device connected to port 1 cannot be removed Other than above: Setting prohibited Bit 0: Reserved Bit 1: Device attached to Port 1 Bit 2: Device attached to Port 2 ... Bit 15: Device attached to Port 15

HcRhStatus is divided into two parts. The high-order word in a longword represents the hub status change bits and the low-order word in a longword represents the hub status bits.

Since this register functions differently in read than in write, functional descriptions will be made separately below. Note that bit titles in read are different from those in write so that bit titles can always suit the functions. For the bit name, the bit title for a read operation is used. Taking bit 0 as an example, the bit name is LPS and the bit title in read is Local Power Status while that in write is Clear Global Power.

- Read

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CRWE	-	-	-	-	-	-	-	-	-	-	-	-	-	OCIC	LPSC
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
	DRWE	-	-	-	-	-	-	-	-	-	-	-	-	-	OCI	LPS
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	Bit Name	Initial Value	R/W	Description
31	CRWE	0	R	Reserved These bits are always read as 0.
30 to 18	—	All 0	R	Reserved These bits are always read as 0.
17	OCIC	0	R	Overcurrent Indicator Change This bit is set to 1 by hardware when a change has occurred to the OCI bit of this register. 0: Operation is not affected 1: Overcurrent indicator has changed
16	LPSC	0	R	Local Power Status Change The root hub does not support the local power status function, thus, this bit is always read as 0.
15	DRWE	0	R	Device Remote Wakeup Enable When this bit is 1, the CSC bit is enabled as a resume event, causing a USB suspend to USB resume state transition and setting the RD interrupt. 0: Does not generate device remote wakeup event 1: Generates device remote wakeup event

14 to 2	—	All 0	R	Reserved These bits are always read as 0.
1	OCI	0	R	Overcurrent Indicator This bit reports overcurrent conditions when the global reporting is implemented. When set, an overcurrent condition exists. When cleared, all power operations are normal. If per-port overcurrent protection is implemented, this bit is always 0. 0: No port is in overcurrent state 1: A port is in overcurrent state
0	LPS	0	R	Local Power Status The root hub does not support the local power status function, thus, this bit is always read as 0.

- Write

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CRWE	-	-	-	-	-	-	-	-	-	-	-	-	-	OCIC	LPSC
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	W	R	R	R	R	R	R	R	R	R	R	R	R	R	W	W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DRWE	-	-	-	-	-	-	-	-	-	-	-	-	-	OCI	LPS
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W

Bit	Bit Name	Initial Value	R/W	Description
31	CRWE	0	W	Clear Remote Wakeup Enable Writing 1 to this bit clears the DRWE bit to 0. Writing 0 has no effect. 0: Operation is not affected 1: The DRWE bit is cleared to 0
30 to 18	—	All 0	R	Reserved Always write 0 to this bit.
17	OCIC	0	W	Overcurrent Indicator Change HCD clears this bit to 0 by writing 1. Writing 0 has no effect. 0: Operation is not affected 1: The OCIC bit is cleared to 0

16	LPSC	0	W	Set Global Power In global power mode (PSM = 0), writing 1 to this bit will power all ports on (and clear the PPS bit to 0). In per-port power mode, it will set the PPS bit to 1 only on ports whose PPCM bit is not set. Writing 0 has no effect. 0: Operation is not affected 1: Power of all ports is turned on
15	DRWE	0	W	Set Remote Wakeup Enable Writing 1 to this bit sets the DRWE bit to 1. Writing 0 has no effect. 0: Operation is not affected 1: Device remote wakeup is enabled
14 to 2	—	All 0	R	Reserved Always write 0 to this bit.
1	OCI	0	R	Reserved Always write 0 to this bit.
0	LPS	0	W	Clear Global Power In global power mode (PSM = 0), writing 1 to this bit will power all ports off (and clear the PPS bit to 0). In per-port power mode, it will clear the PPS bit to 0 only on ports whose PPCM bit is not set. Writing 0 has no effect. 0: Operation is not affected 1: Power of all ports is turned off

21.3.22 Root Hub Port Status 1 Register (HcRhPortStatus1)

HcRhPortStatus1 is used for controlling and reporting the port event for each port. The upper word indicates the change of the port status and the lower word indicates the port status.

Some status bits are implemented with special write behavior (see below). If a transaction (token through handshake) is in progress when a write to change port status occurs, the resulting port status change must be postponed until the transaction completes. Reserved bits should always be written 0.

Since this register functions differently in read than in write, functional descriptions will be made separately below. Note that bit titles in read are different from those in write so that the bit titles can always suit the functions. For the bit name, the bit title for a read operation is used. Taking a

that in write is Clear Port Enable.

- Read

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	PRSC	OCIC	PSSC	PESC	CSC
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
	-	-	-	-	-	-	LSDA	PPS	-	-	-	PRS	POCI	PSS	PES	CCS
Initial value:	0	0	0	0	0	0	0	1	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	Bit Name	Initial Value	R/W	Description
31 to 21	—	All 0	R	Reserved These bits are always read as 0.
20	PRSC	0	R	Port Reset Status Change This bit is set to 1 at the end of the 10-ms port reset signal. 0: Port reset is not complete 1: Port reset is complete
19	OCIC	0	R	Overcurrent Indicator Change This bit is valid only if overcurrent conditions are reported on a per-port basis. This bit is set when root hub changes the POCI bit. 0: POCI has not changed 1: POCI has changed
18	PSSC	0	R	Port Suspend Status Change This bit is set to 1 when the full resume sequence has been completed. This sequence includes the 20-ms resume pulse, LS EOP, and 3-ms resychronization delay. 0: Port resume is incomplete, or PRSC bit is set to 1 1: Port resume is complete
17	PESC	0	R	Port Enable Status Change This bit is set to 1 when a hardware event clears the PES bit to 0. Writing 1 by HCD does not set this bit to 1. 0: PES has not changed 1: PES has changed

16	CSC	0	R	Connector Status Change This bit is set to 1 when a connect or disconnect event occurs. If CCS is cleared to 0 when a write to PPS, PES, or PPS occurs, this bit is set to 1 to force the driver to re-evaluate the connection status since these writes should not occur if the port is disconnected. 0: CCS has not changed 1: CCS has changed Note: When the DR bit is 1, this bit is set to 1 only after a root hub reset to inform the system that the device is attached.
15 to 10	—	All 0	R	Reserved These bits are always read as 0.
9	LSDA	0	R	Low-Speed Device Attached This bit indicates the speed of the device attached to this port. When this bit is 1, a low speed device is attached to this port. When this bit is 0, a full speed device is attached to this port. This bit is valid only when the CCS is set to 1. 0: Full-speed device is attached 1: Low-speed device is attached

8	PPS	1	R	Port Power Status This bit reflects the port's power status, regardless of the type of power switching implemented. Since the initial value of the NPS bit in HcRhDescriptorA is 1, this bit is initially set to 1. The NPS bit should be cleared to 0 beforehand in order to enable power switching. This bit is cleared to 0 if an overcurrent condition is detected. HCD sets this bit to 1 by writing PPS or LPSC. HCD clears this bit to 0 by writing LSDA or OCI. Which power control switches are enabled is determined by PSM and PPCM. In global switching mode (PSM = 0), only Set/ClearGlobalPower command controls this bit. In per-port power switching (PSM = 1), when the PPCM bit for the port is set to 1, only Set/ClearPortPower commands are enabled. When port power is disabled, CCS, PES, PSS, and PRS should be reset. 0: Power of port is turned off 1: Power of port is turned on Note: This bit is always read as 1 when power switching is not supported.
7 to 5	—	All 0	R	Reserved These bits are always read as 0.
4	PRS	0	R	Port Reset Status When this bit is set by a write to this bit, port reset signal is asserted. When reset is completed, this bit is cleared when PRSC is set. This bit cannot be set if CCS is cleared. 0: Port is not in reset state 1: Port is in reset state
3	POCI	0	R	Port Overcurrent Indicator This bit is only valid when the root hub is configured in such a way that overcurrent conditions are reported on a per-port basis. When per-port overcurrent reporting is not supported, this bit is cleared to 0. When this bit is 0, all power operations are normal for this port. When this bit is 1, an overcurrent condition exists on this port. This bit always reflects the overcurrent input signal 0: Port is not in overcurrent state 1: Port is in overcurrent state

2	PSS	0	R	Port Suspend Status This bit indicates the port is suspended or is in the resume sequence. It is set to 1 by a SetSuspendState write and cleared to 0 when PSSC is set to 1 at the end of the resume interval. This bit cannot be set to 1 when CCS is 0. This bit is also cleared to 0 when PRSC is set to 1 at the end of the port reset or when HC enters the USB resume state. When an upstream resume is in progress, it should propagate to HC. 0: Port is not in suspend state 1: Port is in suspend state
1	PES	0	R	Port Enable Status This bit indicates whether the port is enabled or disabled. The root hub may clear this bit to 0 when detecting an overcurrent condition, disconnect event, switched-off power, or operational bus error such as babble. This change also sets PESC to 1. HCD sets this bit to 1 by writing 1 to this bit and clears to 0 this bit by writing CSS. This bit cannot be set to 1 when CCS is 0. This bit is also set, if not already, on completion of a port reset when PRSC is set or port suspend when PSSC is set. 0: Port is disabled 1: Port is enabled
0	CCS	0	R	Current Connect Status This bit reflects the current state of the downstream port. 0: Device is not connected to port 1: Device is connected to port Note: This bit is always read 1 when the attached device is non-removable (DR = H'0002).

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	PRSC	OCIC	PSSC	PESC	CSC
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	W	W	W	W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	LSDA	PPS	-	-	-	PRS	POCI	PSS	PES	CCS
Initial value:	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	W	W	R	R	R	W	W	W	W	W

Bit	Bit Name	Initial Value	R/W	Description
31 to 21	—	All 0	R	Reserved Always write 0 to this bit.
20	PRSC	0	W	Port Reset Status Change HCD writes 1 to this bit to clear this bit to 0. Writing 0 has no effect. 0: Operation is not affected 1: The PRSC bit is cleared to 0
19	OCIC	0	W	Overcurrent Indicator Change HCD writes 1 to this bit to clear this bit to 0. Writing 0 has no effect. 0: Operation is not affected 1: The OCIC bit is cleared to 0
18	PSSC	0	W	Port Suspend Status Change HCD writes 1 to this bit to clear this bit to 0. Writing 0 has no effect. Setting PRSC to 1 will also clear this bit to 0. 0: Operation is not affected 1: The PSSC bit is cleared to 0
17	PESC	0	W	Port Enable Status Change HCD writes 1 to this bit to clear this bit to 0. Writing 0 has no effect. 0: Operation is not affected 1: The PESC bit is cleared to 0
16	CSC	0	W	Connect Status Change HCD writes 1 to this bit to clear this bit to 0. Writing 0 has no effect. 0: Operation is not affected 1: The CSC bit is cleared to 0

15 to 10	—	All 0	R	Reserved Always write 0 to this bit.
9	LSDA	0	W	Clear Port Power HCD writes 1 to this bit to clear the PPS bit to 0. Writing 0 has no effect. 0: Operation is not affected 1: The PPS bit is cleared to 0 and the power of the port is turned off
8	PPS	1	W	Set Port Power HCD writes 1 to this bit to set this bit to 1. Writing 0 has no effect. 0: Operation is not affected 1: The PPS bit is set to 1 and the power of the port is turned on
7 to 5	—	All 0	R	Reserved Always write 0 to this bit.
4	PRS	0	W	Set Port Clear HCD writes 1 to this bit to set the port reset signal. Writing 0 has no effect. Clearing CCS to 0 will not set this bit to 1 but will set CSC to 1 instead. This informs the driver that a disconnected port is reset. 0: Operation is not affected 1: The PRS bit is set to 1
3	POCI	0	W	Clear Suspend Status HCD writes 1 to this bit to initiate a resume. Writing 0 has no effect. Setting both PSS and this bit to 1 starts a resume. 0: Operation is not affected 1: Resume is started
2	PSS	0	W	Set Port Suspend HCD sets this bit to 1 by writing 1. Writing 0 has no effect. Clearing CCS to 0 does not set PSS to 1 but sets CSC to 1 instead. It informs the driver that a disconnected port is suspended. 0: Operation is not affected 1: The PPS bit is set to 1 and the port is suspended

1	PES	0	W	Set Port Enable HCD sets this bit to this bit by writing 1. Writing 0 has no effect. Clearing CCS to 0 does not set PES to 1 but sets CSC to 1 instead. It informs the driver that a disconnected port is enabled. 0: Operation is not affected 1: The PES bit is set to 1 and the port is enabled
0	CCS	0	W	Clear Port Enable HCD writes 1 to this bit to clear the PES bit to 0. Writing 0 has no effect. CCS is not affected by any write. 0: Operation is not affected 1: The PES bit is cleared to 0 and the port is disabled

The USB host has on-chip 8-Kbyte SRAM as a shared memory which is defined in the OpenHCI specification. Write access to the shared memory is possible in bytes, words, or longwords. The data format is little endian. The MSB is bit 31 and the LSB is bit 0. The user can use the endian conversion in bytes or in words by rewriting an appropriate value in the CVRT bit in DMAUCR. Figure 21.2 shows a memory map of the shared memory.

When area P4 is selected as the register area, the CPU sees the shared memory area to be mapped into H'FE34 1000 to H'FE34 2FFF, while the HC sees it to be mapped into H'0000 0000 to H'0000 1FFF. Therefore when the HC accesses the shared memory, the area from H'0000 0000 to H'0000 1FFF must be specified.

When USB HC (Host controller) reads the data at the following 4 Kbyte boundary address in the USB shared memory, the wrong data may be returned to USB HC.

The wrong access address whish USB HC (Host controller) is using.

H'0000 0FFC— H'0000 0FFF
H'0000 1FFC— H'0000 1FFF

The shared memory area is mapped into H'FE34 1000 to H'FE34 2FFF in area P4 or into H'1E34 1000 to H'1E34 2FFF in area 7. For details, see section 32, List of Registers.

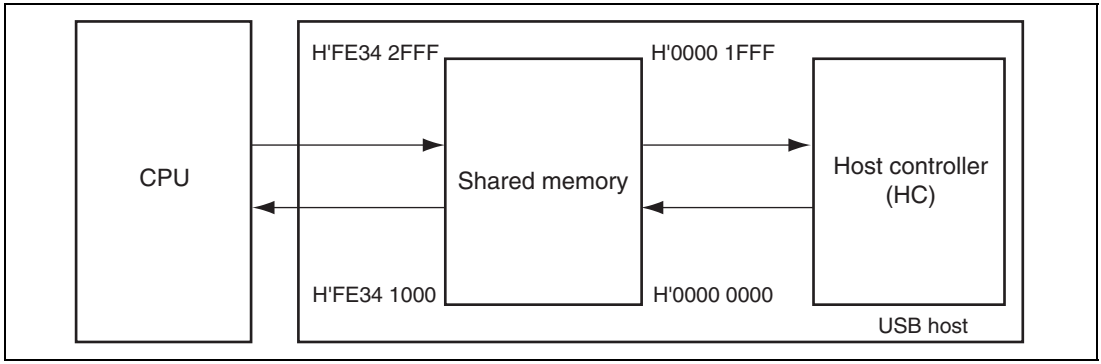


Figure 21.2 Memory Map of Shared Memory

21.5.1 Storage Format of Transfer Data

The USB host assumes that data will be stored sequentially, in little endian order, from low to high addresses, regardless of the CPU endian setting. Figure 21.3 shows USB read operation.

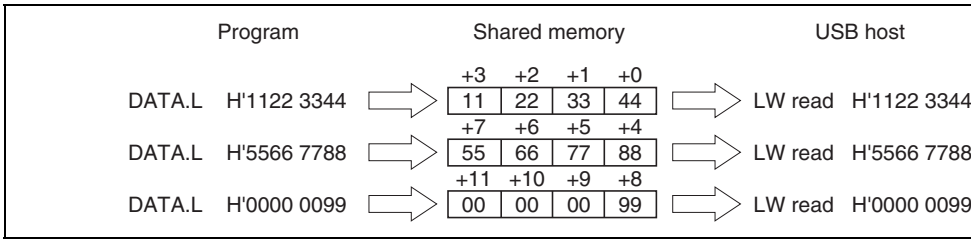


Figure 21.3 USB Read Operation

The data in memory mentioned above and the data read by the USB host must always correspond. When reading data from external memory, the USB host always reads data in longword units regardless of the endian setting. The USB host assumes that read data is in the little endian order, that is, the byte order that places the first byte in the lowest address and the last byte in the highest address. That is, during operation of this IC, data must be stored sequentially in longword units in little endian order from low to high addresses, regardless of whether the endian setting is little endian or big endian.

An example of failure is shown in figure 21.4.

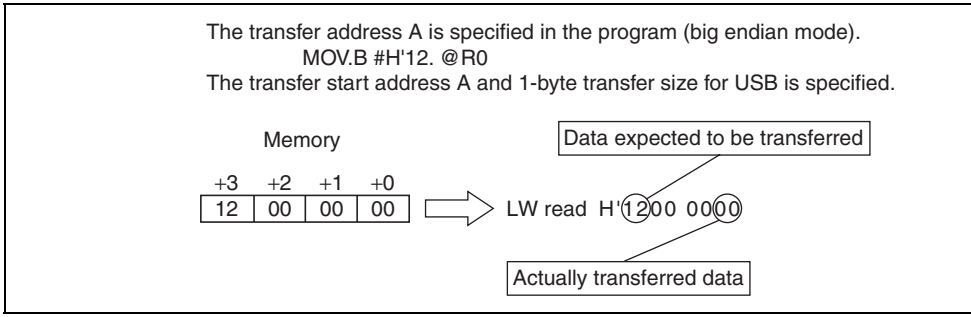


Figure 21.4 Example of Transfer Failure

In this example, USB host controller does not receive #H'12, which is the expected transfer data.

The USB host controller, when writing, stores data sequentially starting with the low order bits in memory in little endian order, so that the data is read/written correctly from both sides regardless of the endian setting. That is, the data is always aligned in little endian format.

ED (endpoint descriptor) and TD (transfer descriptor) that define each transfer transaction of the USB host controller must be placed such that each Dword is aligned on a longword boundary (addresses 4n to 4n + 3) of the memory. Each descriptor must be aligned on a memory address boundary prescribed by the OpenHCI specification Ver.1.0.

21.6 Restrictions on HcRhDescriptorA

When modifying initial settings of the NOCP or OCPM bits in HcRhDescriptorA after reset, keep the following in mind.

1. The initial values are NOCP=1 and OCPM=0, and the USB host controller does not detect overcurrent. To use the overcurrent detection, set NOCP = 0 and OCPM = 1 simultaneously. Modify these bits only once during USB host controller initialization. Do not modify them more than once.
2. Making the settings in step 1 above will not change the settings in the OCI and OCIC bits in HcRhStatus for overcurrent condition information. These bits should be ignored.
3. Making the settings in step 1 above will set HcInterruptStatus.RHSC = 1 even if a port is not in overcurrent condition. Therefore, perform interrupt handling processing shown in figure 21.5.

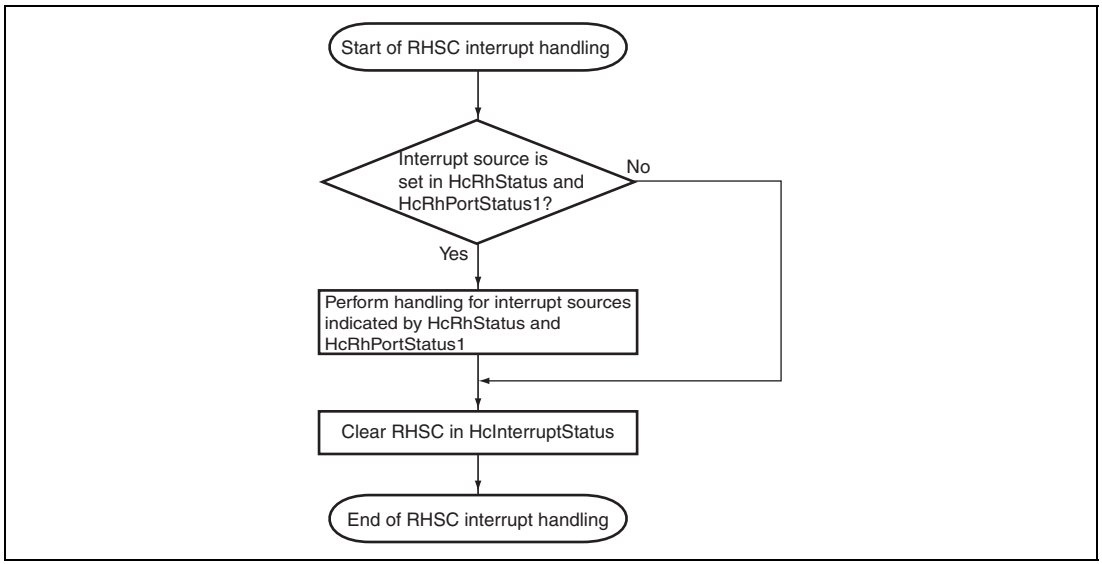


Figure 21.5 Example of RHSC Interrupt Handling

The Controller Area Network 2 (HCAN2) is a module that controls the Controller Area Network (CAN) provided for realtime communication in automobiles or industrial equipment systems. For details on the **CAN specification, refer to the CAN Specification Version 2.0, Robert Bosch GmbH, 1991**.

The section places no constraints upon the implementation of the HCAN2 module in terms of process, packaging or power supply criteria. These issues are resolved appropriately in implementation specifications.

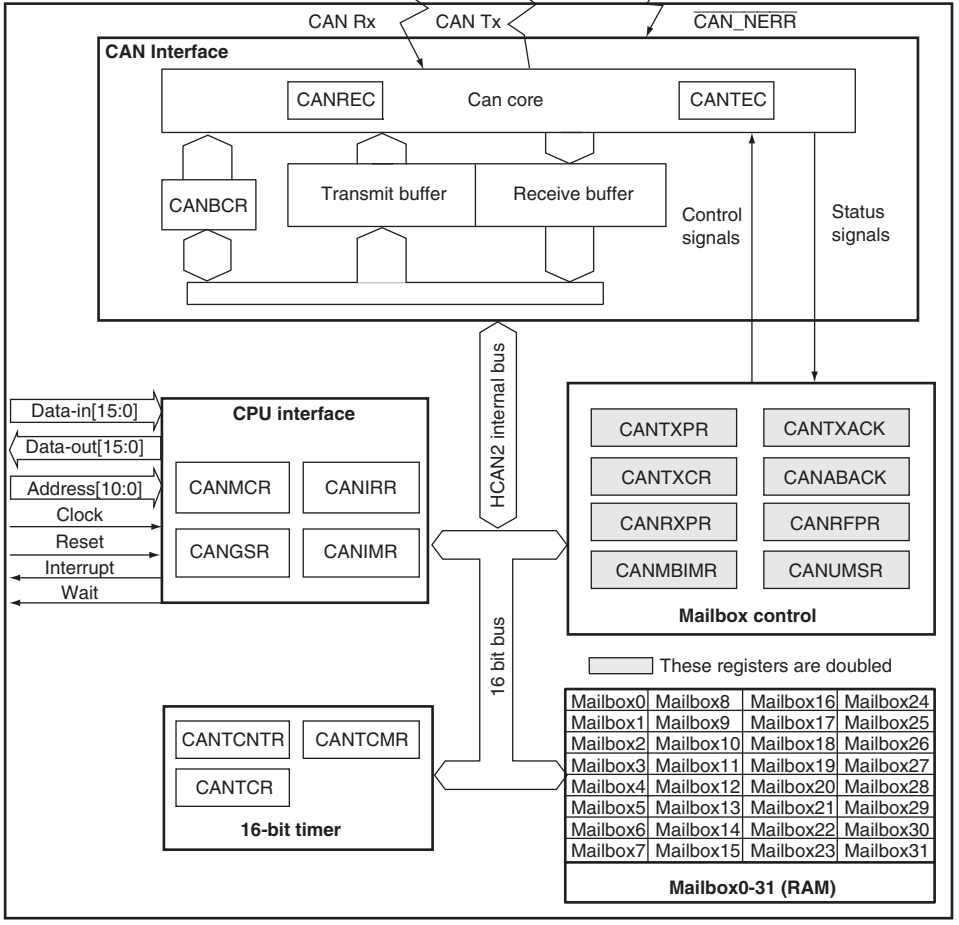
22.1 Features

- Supports CAN specification 2.0A/2.0B and ISO-11898:
- 31 programmable transmit/receive Mailboxes and one receive-only Mailbox
- Sleep mode for low power consumption and automatic recovery from sleep mode by detecting the CAN bus being active
- Programmable receive filter masks (standard identifier and extended identifier) are supported by all Mailboxes
- Programmable CAN data rate up to 1 Mbit/s
- Transmit message queuing with internal priority sorting mechanism against a priority inversion problem of the real time application
- Data buffer access without using handshake
- Flexible CPU interface
- Flexible interrupt structure
- A 16-bit free running timer with flexible clock sources and pre-scaler, timer compare match register
- Supports flexible time stamp function for both transmission and reception (the stamp timing is programmable)

22.2 Architecture

22.2.1 Block diagram

HCAN2 offers a flexible and sophisticated method for the organization and control of CAN frames, which is compliant to CAN2.0B Active and ISO-11898. The module is formed from 5 different functional entities. These are the Micro Processor Interface (MPCI), Mailbox, Mailbox Control, Timer and CAN Interface. The figure below shows the block diagram of the module. The bus interface timing is designed based on SH internal bus interface.



Note: Since the HCAN2 is designed on the basis of a 16-bit bus system, longword (32-bit) accesses are prohibited. All registers can be accessed with words and mailboxes can be accessed with words or bytes.

Legend:

CANTCNTR	: Timer counter register	CANTEC	: Transmit error counter
CANTCR	: Timer control register	CANTXPR	: Transmit pending request register
CANTCMR	: Timer compare match register	CANTXCR	: Transmit cancel register
CANMCR	: Master control register	CANTXACK	: Transmit acknowledge register
CANGSR	: General status register	CANABACK	: Abort acknowledge register
CANIRR	: Interrupt request register	CANRXPR	: Receive data frame pending register
CANIMR	: Interrupt mask register	CANRFPR	: Remote frame request pending register
CANBCR	: Bit configuration register	CANMBIMR	: Mailbox interrupt mask register
CANREC	: Receive error counter	CANUMSR	: Unread message status register

Figure 22.1 Block Diagram of HCAN2 Module

(1) Micro Processor Interface (MPI)

The MPI allows communication between the CPU and the HCAN2 registers/mailboxes to control the timer unit, memory interface and data controller, etc. It also contains the wakeup control logic that detects the CAN bus activities and notifies the MPI and the other parts of HCAN2 so that the HCAN2 can automatically exit the Sleep mode.

The MPI has four registers CANMCR, CANIRR, CANGSR and CANIMR.

(2) Mailbox

Mailboxes are essentially RAM configured as message buffers. There are 32 Mailboxes, and each mailbox stores the following information.

- CAN message control (identifier, dlc, rtr, ide, etc)
- CAN message data (for CAN data frames)
- Time Stamp for message receive/transmit
- Local Acceptance Filter Mask for mailboxes configured to receive
- 3-bit width Mailbox Configuration, Disable Automatic Re-Transmission bit, Auto-Transmission of response to remote frame request and New Message Control bit.

(3) Mailbox Control

The Mailbox control supports the following functions.

For received message, it compares the IDs and generates appropriate RAM address and data to store messages from the CAN interface into the Mailbox, and set/clear the appropriate registers accordingly.

To transmit message, it runs the internal arbitration to pick the correct priority message and loads the message from the Mailbox into the buffer of the CAN Interface, and set/clear appropriate registers accordingly.

Arbitrate Mailbox accesses between the host CPU and the Mailbox Control.

The Mailbox control has registers CANTXPR, CANTXCR, CANTXACK, CANABACK, CANRXPR, CANRFPR, and CANMBIMR.

With the use of the Time Stamp field, the Timer module allows HCAN2 to record the time in which the messages are transmitted and received.

The timer is a 16-bit free running counter that can be controlled by the host CPU. It provides one 16-bit Compare Match Register that holds a reference timer value. When the value of the free running 16-bit timer matches the reference value hold in the compare match register, an interrupt signal is generated.

The clock period of this Timer offers a wide selection derived from the system clock.

The timer has registers CANTCNTR, CANTCR, and CANTCMR.

(5) CAN Interface

The CAN interface supports the CAN Bus Data Link Controller specification described in **CAN Specification Version 2.0, Robert Bosch GmbH, 1991**. This block fulfils all the functions of the standard DLC as specified by the OSI 7 Layer Reference model. This functional entity also provides the registers and logic specific to a given CAN bus, which includes the Receive Error Counter, Transmit Error Counter, the Bit Configuration Registers and various useful test modes. This block also contains functional entities to hold the data received and the data to be transmitted for the CAN Data Link Controller.

22.3 Input/Output Pins

Table 22.1 summarizes the pins of the HCAN2.

Table 22.1 Pin Configuration

Pin Name	I/O	Function
CAN0_RX	Input	CAN bus receive signal of channel 0
CAN0_TX	Output	CAN bus transmit signal of channel 0
CAN0_NERR	Input	CAN bus error of channel 0
CAN1_RX	Input	CAN bus receive signal of channel 1
CAN1_TX	Output	CAN bus transmit signal of channel 1
CAN1_NERR	Input	CAN bus error of channel 1

22.4.1 Memory map

The diagram of the memory map is shown in Fig. 22.2.

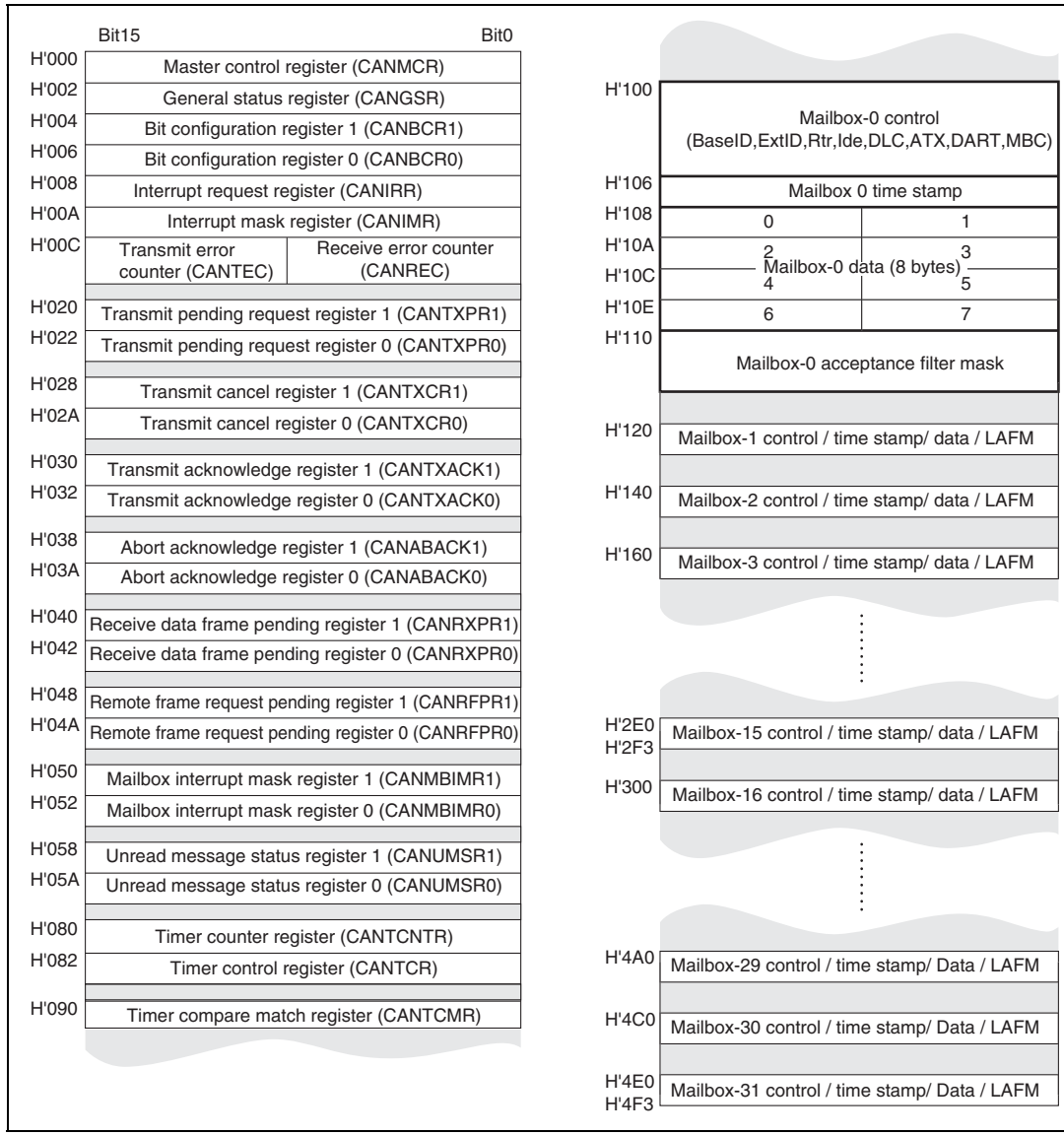


Figure 22.2 HCAN2 Memory Map

Mailboxes function as message buffers to transmit/receive CAN frames. Each Mailbox is comprised of four identical storage fields: 1) Message Control, 2) Message Data, 3) Time Stamp and 4) Local Acceptance Filter Mask. Table 22.2 shows the address map for the control, data, timestamp and LAFM address for each mailbox.

Notes: 1. The Message Control, Timestamp and LAFM fields can only be accessed with 16 bits, whereas the Message Data area can be accessed in units of 16 bits or 8 bits. Also, unused parts of Mailboxes must be initialized during the configuration to their inactive state as they are in effect configured of RAM. When the LAFM is not used to receive messages, it must be cleared.

2. Unused Mailboxes can be used as extra memory. However, it is important in such case to disable the related mailbox (setting MBC to B'111) in order to avoid that the mailbox joins the search for a matching identifier during the reception of messages, and even store a wrong message in the worst case.

Mailbox	Address			
	Control	Time Stamp	Data	LAFM
	6 Bytes	2 Bytes	8 Bytes	4 Bytes
0 (receive-only)	100 to 105	106 to 107	108 to 10F	110 to 113
1	120 to 125	126 to 127	128 to 12F	130 to 133
2	140 to 145	146 to 147	148 to 14F	150 to 153
3	160 to 165	166 to 167	168 to 16F	170 to 173
4	180 to 185	186 to 187	188 to 18F	190 to 193
5	1A0 to 1A5	1A6 to 1A7	1A8 to 1AF	1B0 to 1B3
6	1C0 to 1C5	1C6 to 1C7	1C8 to 1CF	1D0 to 1D3
7	1E0 to IE5	1E6 to 1E7	1E8 to 1EF	1F0 to 1F3
8	200 to 205	206 to 207	208 to 20F	210 to 213
9	220 to 225	226 to 227	228 to 22F	230 to 233
10	240 to 245	246 to 247	248 to 24F	250 to 253
11	260 to 265	266 to 267	268 to 26F	270 to 273
12	280 to 285	286 to 287	288 to 28F	290 to 293
13	2A0 to 2A5	2A6 to 2A7	2A8 to 2AF	2B0 to 2B3
14	2C0 to 2C5	2C6 to 2C7	2C8 to 2CF	2D0 to 2D3
15	2E0 to 2E5	2E6 to 2E7	2E8 to 2EF	2F0 to 2F3
16	300 to 305	306 to 307	308 to 30F	310 to 313
17	320 to 325	326 to 327	328 to 32F	330 to 333
18	340 to 345	346 to 347	348 to 34F	350 to353
19	360 to 365	366 to 367	368 to 36F	370 to 373
20	380 to 385	386 to 387	388 to 38F	390 to 393
21	3A0 to 3A5	3A6 to 3A7	3A8 to 3AF	3B0 to 3B3
22	3C0 to 3C5	3C6 to 3C7	3C8 to 3CF	3D0 to 3D3
23	3E0 to 3E5	3E6 to 3E7	3E8 to 3EF	3F0 to 3F3
24	400 to 405	406 to 407	408 to 40F	410 to 413
25	420 to 425	426 to 427	428 to 42F	430 to 433
26	440 to 445	446 to 447	448 to 44F	450 to 453
27	460 to 465	466 to 467	468 to 46F	470 to 473
28	480 to 485	486 to 487	488 to 48F	490 to 493
29	4A0 to 4A5	4A6 to 4A7	4A8 to 4AF	4B0 to 4B3
30	4C0 to 4C5	4C6 to 4C7	4C8 to 4CF	4D0 to 4D3
31	4E0 to 4E5	4E6 to 4E7	4E8 to 4EF	4F0 to 4F3

structure of a Mailbox in detail.

Address	Data bus																Access Size	Field Name
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
H'100 + N*32	0	STDID[10:0]											RTR	IDE	EXTID [17:16]		16 bits	Control
H'102 + N*32	EXTID[15:0]																16 bits	
H'104 + N*32			NMC	ATX	DART	MBC[2:0]			0		CBE		DLC[3:0]				8/16 bits	
H'106 + N*32	TimeStamp[15:0]																16 bits	Time stamp
H'108 + N*32	MSG_DATA_0 (first Rx/Tx byte)								MSG_DATA_1								8/16 bits	Data
H'10A + N*32	MSG_DATA_2								MSG_DATA_3								8/16 bits	
H'10C + N*32	MSG_DATA_4								MSG_DATA_5								8/16 bits	
H'10E + N*32	MSG_DATA_6								MSG_DATA_7								8/16 bits	
H'110 + N*32	Local acceptance filter mask 0 (LAFM0)																16 bits	LAFM
H'112 + N*32	Local acceptance filter mask 1 (LAFM1)																16 bits	

Notes: 1. All bits shadowed in gray are reserved and the write value should always be 0. The read value is not guaranteed.
2. ATX and DART are not supported by mailbox 0, and the MBC setting of mailbox 0 is restricted.

Figure 22.3 Mailbox N Structure

Address	Bit	Bit Name	Description
H'100 + N*32	15	—	Reserved The write value should always be 0. The read value is not guaranteed.
	14 to 4	STDID10 to STDID0	Standard ID These bits set Identifiers (Standard Identifiers) of Data Frames and remote Frames.
	3	RTR	Remote Transmission Request Used to distinguish between Data Frames and Remote Frames. This bit is overwritten by the received CAN frames depending on Data Frame or Remote Frame. Important: Note that when the ATX bit is set with the setting MBC = B'001, the RTR bit will never be set. When a Remote Frame is received, the host CPU can be notified by CANRFPR set or IRR2 (Remote Frame Request Interrupt), however, as HCAN2 needs to transmit the current message as a Data Frame, the RTR bit remains 0. 0: Data Frame 1: Remote Frame
	2	IDE	Identifier Extension Used to distinguish between the standard format and extended format of CAN data frames and remote frames. 0: Standard format 1: Extended format
	1 0	EXTID17 EXTID16	Identifier Extension These bits set Identifiers (Extended identifiers) for Data Frames and Remote Frames.
H'102 + N*32	15 to 0	EXTID15 to EXTID0	
H'104 + N*32	15, 14	—	Reserved The write value should always be 0. The read value is not guaranteed.

H'04 + N*32	13	NMC	New Message Control When this bit is cleared to 0, the Mailbox of which CANRXPR/CANRFPR bit is already set does not store the new message but maintains the old one and sets the corresponding CANUMSR bit. When this bit is set to 1, the Mailbox of which CANRXPR/CANRFPR bit is already set overwrites with the new message and sets the corresponding CANUMSR bit.
	12	ATX	Automatic Transmission of Data Frame When this bit is set to 1 and a Remote Frame is received into the Mailbox, a Data Frame is transmitted from the same Mailbox using the current contents of the message data by setting the corresponding CANTXPR automatically. The scheduling of transmission is still governed by CAN identifier. In order to use this function, MBC[2:0] needs be programmed to B'001. Important : Note that when this function is used, the RTR bit will never be set despite receiving a Remote Frame. When a Remote Frame is received, the host CPU will be notified by the corresponding CANRFPR set, however, as the HCAN2 needs to transmit the current message as a Data Frame, the RTR bit remains 0. If a mailbox is configured to receive Remote Frames, then the RTR bit is overwritten by the received CAN Frames.
	11	DART	Disable Automatic Re-Transmission When this bit is set, it disables the automatic re-transmission of a message in the event of an error on the CAN bus or an arbitration lost on the CAN bus. When this function is used, the corresponding CANTXCR bit is automatically set at the start of transmission. When this bit is cleared, the HCAN2 tries to transmit the message as many times as required until it is successfully transmitted or it is cancelled by CANTXCR.
	10 9 8	MBC2 MBC1 MBC0	Mailbox Configuration These bits configure the nature of each Mailbox as shown in Table 22.3. When MBC is set to B'111, the Mailbox is inactive, i.e., it does not receive or transmit a message regardless of CANTXPR or other settings. The setting of MBC = 110 is prohibited.
	7, 6	—	Reserved The write value should always be 0. The read value is not guaranteed.

H'104 + N*32	5	CBE	CAN Bus Error An external fault-tolerant CAN transceiver can be used together with the HCAN2 module. In such case the error output pin of the transceiver must be connected to the CAN_NERR pin. The value of CAN_NERR is stored into the CBE bit at the end of each Transmit/Receive (if the message is stored). This bit reports the inverted value of the CAN_NERR pin. Then, using a transceiver with error pin active low, CBE shows a potential physical error with the CAN bus when set to 1. If a transceiver with error pin active high is used the notation must be inverted. As the CAN_NERR value will be updated after transmission or reception in the correspondent Mailbox non-interrupt is dedicated for this function but instead the interrupt for successful transmission (IRR8) or reception (IRR2) should be considered.
	4	—	Reserved The write value should always be 0. The read value is not guaranteed.
	3 2 1 0	DLC3 DLC2 DLC1 DLC0	Data Length Code These bits encode the number of bytes from 0, 1, 2, … 8 that will be transmitted in a Data Frame. 0000: Data length = 0 byte 0001: Data length = 1 byte 0010: Data length = 2 bytes 0011: Data length = 3 bytes 0100: Data length = 4 bytes 0101: Data length = 5 bytes 0110: Data length = 6 bytes 0111: Data length = 7 bytes 1xxx: Data length = 8 bytes

MBC2	MBC1	MBC0	Data Frame Transmit	Remote Frame Transmit	Data Frame Receive	Remote Frame Receive	Remarks
0	0	0	Yes	Yes	No	No	• Not allowed for Mailbox 0
0	0	1	Yes	Yes	No	Yes	• Can be used with ATX • Not allowed for Mailbox 0 • LAFM can be used
0	1	0	No	No	Yes	Yes	• Allowed for Mailbox 0 • LAFM can be used
0	1	1	No	No	Yes	No	• Allowed for Mailbox 0 • LAFM can be used
1	0	0	No	Yes	Yes	Yes	• Not allowed for Mailbox 0 • LAFM can be used
1	0	1	No	Yes	Yes	No	• Not allowed for Mailbox 0 • LAFM can be used
1	1	0	Setting prohibited				
1	1	1	Mailbox inactive				

(2) (Mailbox) Timestamp Field

Storage for the Timestamp on message for transmit/receive. The Timestamp is a function useful for monitoring when a message is transmitted/received: it can be used to verify if the reception/transmission of messages are within an expected schedule.

(3) Message Data Field

Storage for the CAN message data that is transmitted or received. MSG_DATA_0 corresponds to the first data byte that is transmitted or received. The bit order on the bus is bit 7 through to bit 0.

(4) Local Acceptance Filter Mask (LAFM)

This area is used as the Local Acceptance Filter Mask (LAFM) for a receive mailboxes.

- LAFM
  The LAFM is comprised of two 16-bit read/write areas as follows. It allows a Mailbox to accept more than one identifier for receive.

H'112 + N*32	EXTID_LAFM[15:0]	16 bits	

Figure 22.4 Acceptance Filter

If a bit is set in the LAFM, the corresponding bit of the received CAN identifier is ignored when the HCAN2 searches a Mailbox with the matching CAN identifier. If the bit is cleared, the corresponding bit of the received CAN identifier must match the STD_ID/EXT_ID set in the mailbox, in which the CAN frame is to be stored. The structure of the LAFM is the same as the Message Control in a Mailbox. If this function is not required, it must be filled with 0.

Notes: 1. HCAN2 starts to find a matching identifier from Mailbox 31 down to Mailbox 0. As soon as HCAN2 finds one, it stops the search and stores the message into the Mailbox. This means that a received message can only be stored in 1 Mailbox.

2. When a message is received and a matching Mailbox is found, the whole message is stored into the Mailbox. This means that, if the LAFM is used, the STD_ID, RTR, IDE, and EXT_ID may differ to the ones originally set as they are updated with the STD_ID, RTR, IDE, and EXT_ID of the received message.

Address	Bit	Bit Name	Description
H'110 + N*32	15	—	Reserved The write value should always be 0. The read value is not guaranteed.
	14 to 4	STDID_LAFM10 to STDID_LAFM0	Filter Mask [10:0] bits for the CAN Base Identifier [10:0] 0: Corresponding CAN base ID set in Mailbox 0 is enabled. 1: Corresponding CAN base ID set in Mailbox 0 is disabled.
	3, 2	—	Reserved The write value should always be 0. The read value is not guaranteed.
	1 0	EXTID_LAFM17, EXTID_LAFM16	Filter Mask [17:16] bits for the CAN Extended Identifier [17:16] 0: Corresponding Extended CAN base ID is enabled. 1: Corresponding Extended CAN base ID is disabled.
H'112 + N*32	15 to 0	EXTID_LAFM15 to EXTID_LAFM0	Filter Mask [15:0] bits for the CAN Extended Identifier [15:0] 0: Corresponding Extended CAN base ID is enabled. 1: Corresponding Extended CAN base ID is disabled.

The HCAN2 has the following registers. For more information on addresses of registers and register states in each processing, refer to section 32, List of Registers.

Table 22.4 Register Configuration (1)

Ch.	Register Name	Abbrev.	R/W	P4 Address	Area 7 Address	Size	Sync Clock
0	Master control register	CAN0MCR	R/W	H'FE38 0000	H'1E38 0000	16	Pck
	General status register	CAN0GSR	R	H'FE38 0002	H'1E38 0002	16	Pck
	Bit configuration register 1	CAN0BCR1	R/W	H'FE38 0004	H'1E38 0004	16	Pck
	Bit configuration register 0	CAN0BCR0	R/W	H'FE38 0006	H'1E38 0006	16	Pck
	Interrupt request register	CAN0IRR	R/W	H'FE38 0008	H'1E38 0008	16	Pck
	Interrupt mask register	CAN0IMR	R/W	H'FE38 000A	H'1E38 000A	16	Pck
	Error counter	CAN0TECREC	R/W*1	H'FE38 000C	H'1E38 000C	16	Pck
	Transmit pending request register 1	CAN0TXPR1	R/W*2	H'FE38 0020	H'1E38 0020	16	Pck
	Transmit pending request register 0	CAN0TXPR0	R/W*2	H'FE38 0022	H'1E38 0022	16	Pck
	Transmit cancel register 1	CAN0TXCR1	R/W*2	H'FE38 0028	H'1E38 0028	16	Pck
	Transmit cancel register 0	CAN0TXCR0	R/W*2	H'FE38 002A	H'1E38 002A	16	Pck
	Transmit acknowledge register 1	CAN0TXACK1	R/W	H'FE38 0030	H'1E38 0030	16	Pck
	Transmit acknowledge register 0	CAN0TXACK0	R/W	H'FE38 0032	H'1E38 0032	16	Pck
	Abort acknowledge register 1	CAN0ABACK1	R/W	H'FE38 0038	H'1E38 0038	16	Pck
	Abort acknowledge register 0	CAN0ABACK0	R/W	H'FE38 003A	H'1E38 003A	16	Pck
	Receive data frame pending register 1	CAN0RXPR1	R/W	H'FE38 0040	H'1E38 0040	16	Pck
	Receive data frame pending register 0	CAN0RXPR0	R/W	H'FE38 0042	H'1E38 0042	16	Pck
	Remote frame request pending register 1	CAN0RFPR1	R/W	H'FE38 0048	H'1E38 0048	16	Pck
	Remote frame request pending register 0	CAN0RFPR0	R/W	H'FE38 004A	H'1E38 004A	16	Pck
	Mailbox interrupt mask register 1	CAN0MBIMR1	R/W	H'FE38 0050	H'1E38 0050	16	Pck
	Mailbox interrupt mask register 0	CAN0MBIMR0	R/W	H'FE38 0052	H'1E38 0052	16	Pck
	Unread message status register 1	CAN0UMSR1	R/W	H'FE38 0058	H'1E38 0058	16	Pck
	Unread message status register 0	CAN0UMSR0	R/W	H'FE38 005A	H'1E38 005A	16	Pck
	Timer counter register	CAN0TCNTR	R/W	H'FE38 0080	H'1E38 0080	16	Pck

0	Timer control register	CAN0TCR	R/W	H'FE38 0082	H'1E38 0082	16	Pck
	Timer compare match register	CAN0TCMR	R/W	H'FE38 0090	H'1E38 0090	16	Pck
	Mailbox 0	CAN0MB0	R/W*4*5	H'FE38 0100	H'1E38 0100	16	Pck
	Mailbox 1	CAN0MB1	R/W*4*5	H'FE38 0120	H'1E38 0120	16	Pck
	Mailbox 2	CAN0MB2	R/W*4*5	H'FE38 0140	H'1E38 0140	16	Pck
	Mailbox 3	CAN0MB3	R/W*4*5	H'FE38 0160	H'1E38 0160	16	Pck
	Mailbox 4	CAN0MB4	R/W*4*5	H'FE38 0180	H'1E38 0180	16	Pck
	Mailbox 5	CAN0MB5	R/W*4*5	H'FE38 01A0	H'1E38 01A0	16	Pck
	Mailbox 6	CAN0MB6	R/W*4*5	H'FE38 01C0	H'1E38 01C0	16	Pck
	Mailbox 7	CAN0MB7	R/W*4*5	H'FE38 01E0	H'1E38 01E0	16	Pck
	Mailbox 8	CAN0MB8	R/W*4*5	H'FE38 0200	H'1E38 0200	16	Pck
	Mailbox 9	CAN0MB9	R/W*4*5	H'FE38 0220	H'1E38 0220	16	Pck
	Mailbox 10	CAN0MB10	R/W*4*5	H'FE38 0240	H'1E38 0240	16	Pck
	Mailbox 11	CAN0MB11	R/W*4*5	H'FE38 0260	H'1E38 0260	16	Pck
	Mailbox 12	CAN0MB12	R/W*4*5	H'FE38 0280	H'1E38 0280	16	Pck
	Mailbox 13	CAN0MB13	R/W*4*5	H'FE38 02A0	H'1E38 02A0	16	Pck
	Mailbox 14	CAN0MB14	R/W*4*5	H'FE38 02C0	H'1E38 02C0	16	Pck
	Mailbox 15	CAN0MB15	R/W*4*5	H'FE38 02E0	H'1E38 02E0	16	Pck
	Mailbox 16	CAN0MB16	R/W*4*5	H'FE38 0300	H'1E38 0300	16	Pck
	Mailbox 17	CAN0MB17	R/W*4*5	H'FE38 0320	H'1E38 0320	16	Pck
	Mailbox 18	CAN0MB18	R/W*4*5	H'FE38 0340	H'1E38 0340	16	Pck
	Mailbox 19	CAN0MB19	R/W*4*5	H'FE38 0360	H'1E38 0360	16	Pck
	Mailbox 20	CAN0MB20	R/W*4*5	H'FE38 0380	H'1E38 0380	16	Pck
	Mailbox 21	CAN0MB21	R/W*4*5	H'FE38 03A0	H'1E38 03A0	16	Pck
	Mailbox 22	CAN0MB22	R/W*4*5	H'FE38 03C0	H'1E38 03C0	16	Pck
	Mailbox 23	CAN0MB23	R/W*4*5	H'FE38 03E0	H'1E38 03E0	16	Pck
	Mailbox 24	CAN0MB24	R/W*4*5	H'FE38 0400	H'1E39 0400	16	Pck
	Mailbox 25	CAN0MB25	R/W*4*5	H'FE38 0420	H'1E38 0420	16	Pck
	Mailbox 26	CAN0MB26	R/W*4*5	H'FE38 0440	H'1E38 0440	16	Pck
	Mailbox 27	CAN0MB27	R/W*4*5	H'FE38 0460	H'1E38 0460	16	Pck
	Mailbox 28	CAN0MB28	R/W*4*5	H'FE38 0480	H'1E38 0480	16	Pck
	Mailbox 29	CAN0MB29	R/W*4*5	H'FE38 04A0	H'1E38 04A0	16	Pck
	Mailbox 30	CAN0MB30	R/W*4*5	H'FE38 04C0	H'1E38 04C0	16	Pck

0	Mailbox 31	CAN0MB31	R/W*4*5	H'FE38 04E0	H'1E38 04E0	16	Pck
1	Master control register	CAN1MCR	R/W	H'FE39 0000	H'1E39 0000	16	Pck
	General status register	CAN1GSR	R	H'FE39 0002	H'1E39 0002	16	Pck
	Bit configuration register 1	CAN1BCR1	R/W	H'FE39 0004	H'1E39 0004	16	Pck
	Bit configuration register 0	CAN1BCR0	R/W	H'FE39 0006	H'1E39 0006	16	Pck
	Interrupt request register	CAN1IRR	R/W	H'FE39 0008	H'1E39 0008	16	Pck
	Interrupt mask register	CAN1IMR	R/W	H'FE39 000A	H'1E39 000A	16	Pck
	Error counter	CAN1TECREC	R/W*1	H'FE39 000C	H'1E39 000C	16	Pck
	Transmit pending request register 1	CAN1TXPR1	R/W*2	H'FE39 0020	H'1E39 0020	16	Pck
	Transmit pending request register 0	CAN1TXPR0	R/W*2	H'FE39 0022	H'1E39 0022	16	Pck
	Transmit cancel register 1	CAN1TXCR1	R/W*2	H'FE39 0028	H'1E39 0028	16	Pck
	Transmit cancel register 0	CAN1TXCR0	R/W*2	H'FE39 002A	H'1E39 002A	16	Pck
	Transmit acknowledge register 1	CAN1TXACK1	R/W	H'FE39 0030	H'1E39 0030	16	Pck
	Transmit acknowledge register 0	CAN1TXACK0	R/W	H'FE39 0032	H'1E39 0032	16	Pck
	Abort acknowledge register 1	CAN1ABACK1	R/W	H'FE39 0038	H'1E39 0038	16	Pck
	Abort acknowledge register 0	CAN1ABACK0	R/W	H'FE39 003A	H'1E39 003A	16	Pck
	Receive data frame pending register 1	CAN1RXPR1	R/W	H'FE39 0040	H'1E39 0040	16	Pck
	Receive data frame pending register 0	CAN1RXPR0	R/W	H'FE39 0042	H'1E39 0042	16	Pck
	Remote frame request pending register 1	CAN1RFPR1	R/W	H'FE39 0048	H'1E39 0048	16	Pck
	Remote frame request pending register 0	CAN1RFPR0	R/W	H'FE39 004A	H'1E39 004A	16	Pck
	Mailbox interrupt mask register 1	CAN1MBIMR1	R/W	H'FE39 0050	H'1E39 0050	16	Pck
	Mailbox interrupt mask register 0	CAN1MBIMR0	R/W	H'FE39 0052	H'1E39 0052	16	Pck
	Unread message status register 1	CAN1UMSR1	R/W	H'FE39 0058	H'1E39 0058	16	Pck
	Unread message status register 0	CAN1UMSR0	R/W	H'FE39 005A	H'1E39 005A	16	Pck
	Timer counter register	CAN1TCNTR	R/W	H'FE39 0080	H'1E39 0080	16	Pck
	Timer control register	CAN1TCR	R/W	H'FE39 0082	H'1E39 0082	16	Pck
	Timer compare match register	CAN1TCMR	R/W	H'FE39 0090	H'1E39 0090	16	Pck
	Mailbox 0	CAN1MB0	R/W*4*5	H'FE39 0100	H'1E39 0100	16	Pck
	Mailbox 1	CAN1MB1	R/W*4*5	H'FE39 0120	H'1E39 0120	16	Pck
	Mailbox 2	CAN1MB2	R/W*4*5	H'FE39 0140	H'1E39 0140	16	Pck
	Mailbox 3	CAN1MB3	R/W*4*5	H'FE39 0160	H'1E39 0160	16	Pck
	Mailbox 4	CAN1MB4	R/W*4*5	H'FE39 0180	H'1E39 0180	16	Pck

1	Mailbox 5	CAN1MB5	R/W*4*5	H'FE39 01A0	H'1E39 01A0	16	Pck
	Mailbox 6	CAN1MB6	R/W*4*5	H'FE39 01C0	H'1E39 01C0	16	Pck
	Mailbox 7	CAN1MB7	R/W*4*5	H'FE39 01E0	H'1E39 01E0	16	Pck
	Mailbox 8	CAN1MB8	R/W*4*5	H'FE39 0200	H'1E39 0200	16	Pck
	Mailbox 9	CAN1MB9	R/W*4*5	H'FE39 0220	H'1E39 0220	16	Pck
	Mailbox 10	CAN1MB10	R/W*4*5	H'FE39 0240	H'1E39 0240	16	Pck
	Mailbox 11	CAN1MB11	R/W*4*5	H'FE39 0260	H'1E39 0260	16	Pck
	Mailbox 12	CAN1MB12	R/W*4*5	H'FE39 0280	H'1E39 0280	16	Pck
	Mailbox 13	CAN1MB13	R/W*4*5	H'FE39 02A0	H'1E39 02A0	16	Pck
	Mailbox 14	CAN1MB14	R/W*4*5	H'FE39 02C0	H'1E39 02C0	16	Pck
	Mailbox 15	CAN1MB15	R/W*4*5	H'FE39 02E0	H'1E39 02E0	16	Pck
	Mailbox 16	CAN1MB16	R/W*4*5	H'FE39 0300	H'1E39 0300	16	Pck
	Mailbox 17	CAN1MB17	R/W*4*5	H'FE39 0320	H'1E39 0320	16	Pck
	Mailbox 18	CAN1MB18	R/W*4*5	H'FE39 0340	H'1E39 0340	16	Pck
	Mailbox 19	CAN1MB19	R/W*4*5	H'FE39 0360	H'1E39 0360	16	Pck
	Mailbox 20	CAN1MB20	R/W*4*5	H'FE39 0380	H'1E38 0380	16	Pck
	Mailbox 21	CAN1MB21	R/W*4*5	H'1E39 03A0	H'1E38 03A0	16	Pck
	Mailbox 22	CAN1MB22	R/W*4*5	H'1E39 03C0	H'1E38 03C0	16	Pck
	Mailbox 23	CAN1MB23	R/W*4*5	H'1E39 03E0	H'1E38 03E0	16	Pck
	Mailbox 24	CAN1MB24	R/W*4*5	H'FE39 0400	H'1E39 0400	16	Pck
	Mailbox 25	CAN1MB25	R/W*4*5	H'FE39 0420	H'1E39 0420	16	Pck
	Mailbox 26	CAN1MB26	R/W*4*5	H'FE39 0440	H'1E39 0440	16	Pck
	Mailbox 27	CAN1MB27	R/W*4*5	H'FE39 0460	H'1E39 0460	16	Pck
	Mailbox 28	CAN1MB28	R/W*4*5	H'FE39 0480	H'1E39 0480	16	Pck
	Mailbox 29	CAN1MB29	R/W*4*5	H'FE39 04A0	H'1E39 04A0	16	Pck
	Mailbox 30	CAN1MB30	R/W*4*5	H'FE39 04C0	H'1E39 04C0	16	Pck
	Mailbox 31	CAN1MB31	R/W*4*5	H'FE39 04E0	H'1E39 04E0	16	Pck

Ch.	Register Name	Abbrev.	Power-on Reset by $\overline{\text{RESET}}$ Pin/WDT/ H-UDI	Manual Reset by $\overline{\text{RESET}}$ Pin/WDT/ Multiple Exception	Sleep by Sleep Instruction/ Deep Sleep	Standby by Hardware	Standby by Software/ Each Module
0	Master control register	CAN0MCR	H'0001	H'0001	Retained	*6	Retained
	General status register	CAN0GSR	H'000C	H'000C	Retained		Retained
	Bit configuration register 1	CAN0BCR1	H'0000	H'0000	Retained		Retained
	Bit configuration register 0	CAN0BCR0	H'0000	H'0000	Retained		Retained
	Interrupt request register	CAN0IRR	H'0001	H'0001	Retained		Retained
	Interrupt mask register	CAN0IMR	H'FFFF	H'FFFF	Retained		Retained
	Error counter	CAN0TECREC	H'0000	H'0000	Retained		Retained
	Transmit pending request register 1	CAN0TXPR1	H'0000	H'0000	Retained		Retained
	Transmit pending request register 0	CAN0TXPR0	H'0000	H'0000	Retained		Retained
	Transmit cancel register 1	CAN0TXCR1	H'0000	H'0000	Retained		Retained
	Transmit cancel register 0	CAN0TXCR0	H'0000	H'0000	Retained		Retained
	Transmit acknowledge register 1	CAN0TXACK1	H'0000	H'0000	Retained		Retained
	Transmit acknowledge register 0	CAN0TXACK0	H'0000	H'0000	Retained		Retained
	Abort acknowledge register 1	CAN0ABACK1	H'0000	H'0000	Retained		Retained
	Abort acknowledge register 0	CAN0ABACK0	H'0000	H'0000	Retained		Retained
	Receive data frame pending register 1	CAN0RXPR1	H'0000	H'0000	Retained		Retained
	Receive data frame pending register 0	CAN0RXPR0	H'0000	H'0000	Retained		Retained
	Remote frame request pending register 1	CAN0RFPR1	H'0000	H'0000	Retained		Retained
	Remote frame request pending register 0	CAN0RFPR0	H'0000	H'0000	Retained		Retained
	Mailbox interrupt mask register 1	CAN0MBIMR1	H'FFFF	H'FFFF	Retained		Retained
	Mailbox interrupt mask register 0	CAN0MBIMR0	H'FFFF	H'FFFF	Retained		Retained
	Unread message status register 1	CAN0UMSR1	H'0000	H'0000	Retained		Retained
	Unread message status register 0	CAN0UMSR0	H'0000	H'0000	Retained		Retained
	Timer counter register	CAN0TCNTR	H'0000	H'0000	Retained		Retained
	Timer control register	CAN0TCR	H'0000	H'0000	Retained		Retained
	Timer compare match register	CAN0TCMR	H'0000	H'0000	Retained		Retained
	Mailbox 0	CAN0MB0	Undefined	Undefined	Retained		Retained
	Mailbox 1	CAN0MB1	Undefined	Undefined	Retained		Retained

Ch.	Register Name	Abbrev.	Reset by RESET Pin/WDT/ H-UDI	Reset by RESET Pin/WDT/ Multiple Exception	Sleep by Sleep Instruction/ Deep Sleep	by Hardware	by Software/ Each Module
0	Mailbox 2	CAN0MB2	Undefined	Undefined	Retained	*6	Retained
	Mailbox 3	CAN0MB3	Undefined	Undefined	Retained		Retained
	Mailbox 4	CAN0MB4	Undefined	Undefined	Retained		Retained
	Mailbox 5	CAN0MB5	Undefined	Undefined	Retained		Retained
	Mailbox 6	CAN0MB6	Undefined	Undefined	Retained		Retained
	Mailbox 7	CAN0MB7	Undefined	Undefined	Retained		Retained
	Mailbox 8	CAN0MB8	Undefined	Undefined	Retained		Retained
	Mailbox 9	CAN0MB9	Undefined	Undefined	Retained		Retained
	Mailbox 10	CAN0MB10	Undefined	Undefined	Retained		Retained
	Mailbox 11	CAN0MB11	Undefined	Undefined	Retained		Retained
	Mailbox 21	CAN0MB21	Undefined	Undefined	Retained		Retained
	Mailbox 22	CAN0MB22	Undefined	Undefined	Retained		Retained
	Mailbox 23	CAN0MB23	Undefined	Undefined	Retained		Retained
	Mailbox 24	CAN0MB24	Undefined	Undefined	Retained		Retained
	Mailbox 25	CAN0MB25	Undefined	Undefined	Retained		Retained
	Mailbox 26	CAN0MB26	Undefined	Undefined	Retained		Retained
	Mailbox 27	CAN0MB27	Undefined	Undefined	Retained		Retained
	Mailbox 28	CAN0MB28	Undefined	Undefined	Retained		Retained
	Mailbox 29	CAN0MB29	Undefined	Undefined	Retained		Retained
	Mailbox 30	CAN0MB30	Undefined	Undefined	Retained		Retained
	Mailbox 31	CAN0MB31	Undefined	Undefined	Retained		Retained
1	Master control register	CAN1MCR	H'0001	H'0001	Retained		Retained
	General status register	CAN1GSR	H'000C	H'000C	Retained		Retained
	Interrupt mask register	CAN1IMR	H'FFFF	H'FFFF	Retained		Retained
	Error counter	CAN1TECREC	H'0000	H'0000	Retained		Retained
	Transmit pending request register 1	CAN1TXPR1	H'0000	H'0000	Retained		Retained
	Transmit pending request register 0	CAN1TXPR0	H'0000	H'0000	Retained		Retained
	Transmit cancel register 1	CAN1TXCR1	H'0000	H'0000	Retained		Retained
	Transmit cancel register 0	CAN1TXCR0	H'0000	H'0000	Retained		Retained
	Transmit acknowledge register 1	CAN1TXACK1	H'0000	H'0000	Retained		Retained

Ch.	Register Name	Abbrev.	Reset by RESET Pin/WDT/ H-UDI	Reset by RESET Pin/WDT/ Multiple Exception	Sleep by Sleep Instruction/ Deep Sleep	by Hardware	by Software/ Each Module
1	Transmit acknowledge register 0	CAN1TXACK0	H'0000	H'0000	Retained	*6	Retained
	Abort acknowledge register 1	CAN1ABACK1	H'0000	H'0000	Retained		Retained
	Abort acknowledge register 0	CAN1ABACK0	H'0000	H'0000	Retained		Retained
	Receive data frame pending register 1	CAN1RXPR1	H'0000	H'0000	Retained		Retained
	Receive data frame pending register 0	CAN1RXPR0	H'0000	H'0000	Retained		Retained
	Remote frame request pending register 1	CAN1RFPR1	H'0000	H'0000	Retained		Retained
	Remote frame request pending register 0	CAN1RFPR0	H'0000	H'0000	Retained		Retained
	Mailbox interrupt mask register 1	CAN1MBIMR1	H'FFFF	H'FFFF	Retained		Retained
	Mailbox interrupt mask register 0	CAN1MBIMR0	H'FFFF	H'FFFF	Retained		Retained
	Unread message status register 1	CAN1UMSR1	H'0000	H'0000	Retained		Retained
	Unread message status register 0	CAN1UMSR0	H'0000	H'0000	Retained		Retained
	Timer counter register	CAN1TCNTR	H'0000	H'0000	Retained		Retained
	Timer control register	CAN1TCR	H'0000	H'0000	Retained		Retained
	Timer compare match register	CAN1TCMR	H'0000	H'0000	Retained		Retained
	Mailbox 0	CAN1MB0	Undefined	Undefined	Retained		Retained
	Mailbox 1	CAN1MB1	Undefined	Undefined	Retained		Retained
	Mailbox 2	CAN1MB2	Undefined	Undefined	Retained		Retained
	Mailbox 3	CAN1MB3	Undefined	Undefined	Retained		Retained
	Mailbox 4	CAN1MB4	Undefined	Undefined	Retained		Retained
	Mailbox 5	CAN1MB5	Undefined	Undefined	Retained		Retained
	Mailbox 6	CAN1MB6	Undefined	Undefined	Retained		Retained
	Mailbox 7	CAN1MB7	Undefined	Undefined	Retained		Retained
	Mailbox 8	CAN1MB8	Undefined	Undefined	Retained		Retained
	Mailbox 9	CAN1MB9	Undefined	Undefined	Retained		Retained
	Mailbox 10	CAN1MB10	Undefined	Undefined	Retained		Retained
	Mailbox 11	CAN1MB11	Undefined	Undefined	Retained		Retained
	Mailbox 12	CAN1MB12	Undefined	Undefined	Retained		Retained
	Mailbox 13	CAN1MB13	Undefined	Undefined	Retained		Retained

Ch.	Register Name	Abbrev.	Reset by RESET Pin/WDT/ H-UDI	by RESET Pin/WDT/ Multiple Exception	Sleep by Sleep Instruction/ Deep Sleep	by Hardware	by Software/ Each Module
1	Mailbox 14	CAN1MB14	Undefined	Undefined	Retained	*6	Retained
	Mailbox 15	CAN1MB15	Undefined	Undefined	Retained		Retained
	Mailbox 16	CAN1MB16	Undefined	Undefined	Retained		Retained
	Mailbox 17	CAN1MB17	Undefined	Undefined	Retained		Retained
	Mailbox 18	CAN1MB18	Undefined	Undefined	Retained		Retained
	Mailbox 19	CAN1MB19	Undefined	Undefined	Retained		Retained
	Mailbox 20	CAN1MB20	Undefined	Undefined	Retained		Retained
	Mailbox 21	CAN1MB21	Undefined	Undefined	Retained		Retained
	Mailbox 22	CAN1MB22	Undefined	Undefined	Retained		Retained
	Mailbox 23	CAN1MB23	Undefined	Undefined	Retained		Retained
	Mailbox 24	CAN1MB24	Undefined	Undefined	Retained		Retained
	Mailbox 25	CAN1MB25	Undefined	Undefined	Retained		Retained
	Mailbox 26	CAN1MB26	Undefined	Undefined	Retained		Retained
	Mailbox 27	CAN1MB27	Undefined	Undefined	Retained		Retained
	Mailbox 28	CAN1MB28	Undefined	Undefined	Retained		Retained
	Mailbox 29	CAN1MB29	Undefined	Undefined	Retained		Retained
	Mailbox 30	CAN1MB30	Undefined	Undefined	Retained		Retained
	Mailbox 31	CAN1MB31	Undefined	Undefined	Retained		Retained

Notes: 1. Can be written to when MCR15 = MCR = 1 (Test Mode).
2. Only allows writing a 1 to the Mailbox designated for transmission.
3. Cannot be modified.
4. The Message Control, Message Data, Timestamp and LAFM fields can only be accessed with 16 bits, whereas the Message Data area can be accessed with 16 bits or 8 bits. Also, unused parts of Mailboxes must be initialized during the configuration to their inactive state as they are configured of RAM.
5. Unused Mailboxes can be used as memory. However, it is important in such case to disable the related mailbox (setting MBC to B'111) in order to avoid that the mailbox joins the search for a matching identifier during the reception of messages, and even store a wrong message in the worst case.
6. After exiting hardware standby mode, this LSI enters the Power-On Reset state by the RESET pin.

CANMCR is a 16-bit read/write register that controls HCAN2.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TST7	TST6	TST5	TST4	TST3	TST2	TST1	TST0	MCR7	-	MCR5	-	-	MCR2	MCR1	MCR0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	TST7	0	R/W	Test Mode Enables/disables the Test Modes settable by TST6 to TST0 bits. When this bit is set, the following TST6 to TST0 bits become effective. 0: HCAN2 is in Normal Mode. 1: HCAN2 is in Test Mode.
14	TST6	0	R/W	Write CAN Error Counters Enables the Transmit Error Counter (TEC) and Receive Error Counter (REC) registers to be writable. Only TEC can be directly written to. The same value written into TEC is automatically written into REC. The maximum value that can be written into TEC/REC is D'255 (H'FF). This means that the HCAN2 cannot be forced into the Bus Off state. Before writing into TEC/REC, the HCAN2 needs to be put into Halt Mode. When writing into TEC/REC, the TST7 bit needs be set to 1. 0: TEC and REC is not writable but read-only. 1: TEC and REC is writable with the same value at the same time.
13	TST5	0	R/W	Force to Error Passive This bit can force HCAN2 to become Error Passive node, regardless of the Error Counters. 0: State of HCAN2 depends on the Error Counters. 1: HCAN2 behaves as an Error Passive node regardless of the Error Counters.

12	TST4	0	R/W	Auto Acknowledge Mode Allows HCAN2 to generate its own Acknowledge bit in order to enable Self Test. In order to achieve the Self Test mode, the message transmitted needs to be read back, and there are two settings for this. One is to set [Enable Internal Loop = 1 & Disable Tx Output = 1 & Disable Rx Input = 1], so that the Tx value can be internally provided to the Rx. The other way is to set [Enable Internal Loop = 0 & Disable Tx Output = 0 & Disable Rx Input = 0] and connect the Tx and Rx onto the CAN bus so that the transmitted data can be received via the CAN bus. 0: HCAN2 does not generate its own Acknowledge bit. 1: HCAN2 generates its own Acknowledge bit.
11	TST3	0	R/W	Disable Error Counters Enables/disables the Error Counters (TEC/REC) to be functional. When this bit is disabled (set to 1), the Error Counters (TEC/REC) remain unchanged and hold their current value. When this bit is enabled (cleared to 0), the Error Counters (TEC/REC) function according to the CAN specification. 0: Error Counters (TEC/REC) function according to the CAN specification. 1: Error Counters (TEC/REC) remain unchanged and holds the current value.
10	TST2	0	R/W	Disable Rx InputControls the Rx to be supplied into the CAN Interface block. When this bit is enabled (cleared to 0), the Rx pin value is supplied into the CAN Interface block. When this bit is disabled (set to 1), the Rx value for the CAN block always remains recessive or the Tx value internally connected if Enable Internal Loop = 1. 0: External Rx pin value is supplied for the CAN Interface block. 1: Internal Loop Enable = 0: Rx value always remains recessive for the CAN Interface block. Internal Loop Enable = 1: Tx value is internally supplied for the CAN Interface block.

9	TST1	0	R/W	Disable Tx Output Controls the Tx pin to output transmit data or recessive bits. When this bit is enabled (cleared to 0), the internal transmit output value appears on the Tx pin. When this bit is disabled (set to 1), the Tx Output pin always remains recessive or Tx value is internally looped back the internal Rx if Enable Internal Loop = 1. 0: External Tx pin value is supplied for the CAN Interface block. 1: Internal Loop Enable = 0: Tx is always recessive on the Tx pin. Internal Loop Enable = 1: Tx is internally looped back the internal Rx.
8	TST0	0	R/W	Enable Internal Loop Enables/ disables the internal Tx looped back to the internal Rx. For details, refer to 22.6 Operation. 0: Rx is fed from the Rx pin. 1: Rx is fed from the internal Tx signal.
7	MCR7	0	R/W	Auto- Wake Mode Enables/disables the Auto wake mode. If this bit is set, HCAN2 automatically cancels the Sleep Mode (MCR5) by detecting CAN bus activity (dominant bit). If MCR7 is cleared HCAN2 does not automatically cancel the Sleep Mode. 0: HCAN2 does not automatically cancel Sleep Mode. 1: HCAN2 automatically cancels Sleep Mode when the CAN bus active is detected.
6	—	0	R	Reserved The write value should always be 0. The read value is not guaranteed.

5	MCR5	0	R/W	Sleep Mode Enables/disables Sleep mode transition. If this bit is set, the Sleep Mode is enabled. The HCAN2 waits for the completion of the current bus activity before shutting down. Until this mode is terminated HCAN2 will ignore all CAN bus activities. The two Error Counters (TEC and REC) will remain the same values during Sleep mode. Sleep mode will be exited in two ways: • By writing a 0 to this bit. • If MCR7 is enabled after detecting the dominant bit on the CAN bus. When leaving this mode, the HCAN2 will synchronize to the CAN bus (by checking for 11 recessive bits) before re-initializing. This means that, when the second method above is used, the HCAN2 will miss the first message to receive, however, CAN transceivers have the same feature, and the software needs to be designed in this manner. Important: This mode is same as setting the module to the Halt mode and stopping the clock. This means that, the interrupt is generated from IRR0 when entering the Sleep mode. During the Sleep mode, only the MPI block is accessible, i.e., CANMCR/CANGSR/CANIRR/CANIMR are accessible. However, for example, IRR1 cannot be cleared as it is an OR'ed signal of CANRXPR that cannot be cleared during the Sleep mode, therefore, it is recommended to set the Halt mode first and then transit to the Sleep mode. 0: HCAN2 sleep mode is released. 1: Transition to HCAN2 sleep mode is enabled.

4,3	—	All 0	R	Reserved The write value should always be 0. The read value is not guaranteed.
2	MCR2	0	R/W	Message Transmission Priority Selects the order of transmission for pending transmit data. When this bit is set, pending transmit data are sent in order of the bit position in CANTXPR. The order of transmission starts from Mailbox 31 as the highest priority, and then down to Mailbox 1 (if those mailboxes are configured for transmission). If this bit is cleared, all transmit messages are queued with respect to their priority (by running internal arbitration). The highest priority message has the Arbitration Field with the lowest digital value and is transmitted first. The internal arbitration includes the RTR bit and the IDE bit. 0: Transmission order is determined by the message identifier priority. 1: Transmission order is determined by the Mailbox number priority (Mailbox 31 → Mailbox 1).
1	MCR1	0	R/W	Halt Request When this bit is set, the CAN controller completes its current operation and then to be cut off the CAN bus. HCAN2 remains in this Halt Mode until this bit is cleared. During the Halt Mode, the CAN Interface does not join the CAN bus activity or does not store messages nor transmit messages. All the registers and Mailbox contents retain. HCAN2 will complete the current operation if it is a transmitter or a receiver, and then enter the Halt Mode. If the CAN bus is in idle or intermission state, HCAN2 will enter the Halt Mode immediately. Entering the Halt Mode is notified by IRR0 and GSR4. In the Halt Mode, HCAN2 configuration can be modified as it does not join the bus activity. This bit has be cleared by writing a 0 to re-join the CAN bus. After this bit is cleared, the CAN Interface waits until it detects 11 recessive bits, and then joins the CAN bus. 0: Normal operating mode 1: Halt Mode transition request.

0	MCR0	1	R/W	Reset Request Controls resetting of the HCAN2 module. After detecting a Reset request, the HCAN2 controller enters its reset routine, re-initializing the internal logic, and then setting GSR3 and IRR0 to notify the Reset Mode. During the re-initialization, all the registers are cleared. This bit has to be cleared by writing a 0 to join the CAN bus. After this bit is cleared, the HCAN2 module needs to be re-configured, waits until it detects 11 recessive bits, and then joins the CAN bus. After a Power-On Reset, this bit and GSR3 are always set. This means that a Reset request has been made. 0: CAN Interface normal operating mode (MCR0 = 0 and GSR3 = 0) Setting condition: When 0 is written after the HCAN2 Reset. 1: CAN Interface Reset Mode request.

CANGSR is a 16-bit read-only register that indicates the status of the HCAN2.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	GSR5	GSR4	GSR3	GSR2	GSR1	GSR0
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/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 6	—	All 0	—	Reserved The write value should always be 0. The read value is not guaranteed.
5	GSR5	0	R	Error Passive Status Indicates that the CAN Interface is Error Passive or not. This bit will be set as soon as the HCAN2 enters the Error Passive state and is cleared when the module returns to the Error Active state (This means that GSR5 will stay high during the Error Passive and Bus Off). Consequently, to find out the accurate state, both GSR5 and GSR0 must be considered. 0: HCAN2 is not Error Passive. Setting condition: HCAN2 is in Error Active state. 1: HCAN is Error Passive (if GSR.0 = 0) Setting condition: TEC ≥ 128 or REC ≥ 128
4	GSR4	0	R	Halt/Sleep Status Indicates whether the CAN Interface is in the Halt/Sleep state or not. 0: HCAN2 is not in the Halt state or Sleep state. 1: HCAN2 is in the Halt mode (if MCR1 = 1) or Sleep mode (if MCR5 = 1). Setting condition: when MCR1 is set and the CAN bus is in intermission or idle state.
3	GSR3	1	R	Reset Status Indicates whether the CAN interface is in the Reset state (Configuration mode) or not. 0: Normal operating state Setting condition: After the HCAN2 internal Reset 1: Reset state (Configuration mode)

2	GSR2	1	R	Message Transmission Complete Flag Flag that indicates to the host CPU if HCAN2 is processing transmission requests or a transmission is completed. This bit is an OR'ed signal of all the CANTXPR bits. Please note the difference to the meaning of IRR8 (Slot Empty) that is an OR'ed signal of all the CANTXACK/CANABACK bits. 0: Transmission in progress 1: There is no message requested for transmission.
1	GSR1	0	R	Transmit/Receive Warning Flag Flag that indicates an error warning. 0: Clearing condition: TEC < 96 or REC < 96 or TEC ≥ 256 1: Setting condition: 96 ≤ TEC < 256 or 96 ≤ REC < 256
0	GSR0	0	R	Bus Off Flag Flag that indicates that the HCAN2 is in the Bus Off state. 0: Clearing condition: Recovery from the Bus Off State 1: Setting condition: TEC ≥ 256 (Bus Off state)

22.5.3 Bit Configuration Registers 1 and 0 (CANBCR1, CANBCR0)

The CANBCR registers are 16-bit read/write registers that is used to set CAN bit timing parameters and the baud rate pre-scaler for the CAN interface.

For the following description, the timequanta is defined as follows:

$$\text{Timequanta} = \frac{\text{BRP}}{f_{clk}}$$

Where: BRP (Baud Rate Predivider) is a value stored in CANBCR0 and f_{clk} is the peripheral clock frequency.

- CANBCR1

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TSEG1_3	TSEG1_2	TSEG1_1	TSEG1_0	-	TSEG2_2	TSEG2_1	TSEG2_0	-	-	SJW1	SJW0	-	-	EG	BSP
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	R/W	R/W	R/W	R	R	R/W	R/W	R	R	R/W	R/W

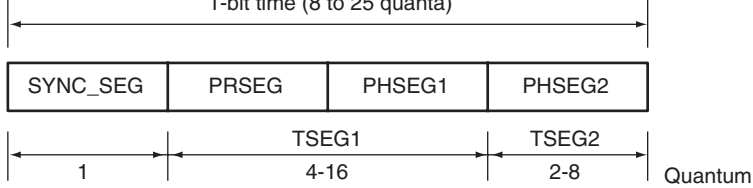
15	TSEG1_3	0	R/W	Time Segment 1 These bits are used to set the segment for absorbing output buffer, CAN bus, and input buffer delay. A value from 4 to 16 can be set. 0000: Setting prohibited 0001: Setting prohibited 0010: Setting prohibited 0011: PRSEG + PHSEG1 = 4 time quanta 0100: PRSEG + PHSEG1 = 5 time quanta : 1111: PRSEG + PHSEG1 = 16 time quanta
14	TSEG1_2	0	R/W	
13	TSEG1_1	0	R/W	
12	TSEG1_0	0	R/W	
11	—	0	—	Reserved The write value should always be 0. The read value is not guaranteed.
10	TSEG2_2	0	R/W	Time Segment 2 These bits are used to set the segment for correcting a 1-bit time error. A value from 2 to 8 time quanta can be set as shown below. 000: Setting prohibited 001: PHSEG2 = 2 time quanta (Conditionally prohibited. See table 22.2.) 010: PHSEG2 = 3 time quanta 011: PHSEG2 = 4 time quanta 100: PHSEG2 = 5 time quanta 101: PHSEG2 = 6 time quanta 110: PHSEG2 = 7 time quanta 111: PHSEG2 = 8 time quanta
9	TSEG2_1	0	R/W	
8	TSEG2_0	0	R/W	
7, 6	—	All 0	—	Reserved The write value should always be 0. The read value is not guaranteed.
5	SJW1	0	R/W	ReSynchronization Jump Width These bits set the synchronization jump width. 00: Synchronization jump width = 1 time quantum 01: Synchronization jump width = 2 time quanta 10: Synchronization jump width = 3 time quanta 11: Synchronization jump width = 4 time quanta
4	SJW0	0	R/W	
3, 2	—	All 0	—	Reserved The write value should always be 0. The read value is not guaranteed.

1	EG	0	R/W	Edge Select Selects at which edge is to be used for re-synchronization. In order to comply to the standard CAN, this bit should be cleared to 0. 0: Re-synchronization is performed at falling edge of Rx. 1: Re-synchronization is performed at both rising and falling edge of Rx.
0	BSP	0	R/W	Bit Sample Point Sets the point at which data is sampled. Three-time sampling is only available when the BRP is programmed to be less than 4. 0: Bit sampling at one point (end of Time Segment 1) 1: Bit sampling at three points (end of Time Segment 1, and 1 time quantum before and after)

- CANBCR0

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	BRP7	BRP6	BRP5	BRP4	BRP3	BRP2	BRP1	BRP0
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

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	—	All 0	—	Reserved The write value should always be 0. The read value is not guaranteed.
7 to 0	BRP7 to BRP0	All 0	R/W	Baud Rate Pre-scale These bits are used to set the clock used for the Time Quantum. 00000000: 1 × System clock 00000001: 2 × System clock 00000010: 3 × System clock : (BRP+1) × System clock 11111111: 256 × System clock



SYNC_SEG: Segment for establishing synchronization of nodes on the CAN bus. (Normal bit edge transitions occur in this segment.)

PRSEG: Segment for compensating for physical delay between networks

PHSEG1: Buffer segment for phase drift (positive) (This segment is extended when synchronization (resynchronization) is established.)

PHSEG2: Buffer segment for phase drift (negative) (This segment is shortened when synchronization (resynchronization) is established.)

The HCAN2 Bit Rate calculation is:

$$\text{Bit rate} = \frac{f_{clk}}{BRP \quad (TSEG1 + TSEG2 + 1)}$$

where BRP, TSEG1 and TSEG2 are derived values from the descriptions of the tables above, but not the actual programmed values. The "+ 1" is for the SYNC_SEG and fixed to 1 time quantum.

f_{CLK} = Pck (peripheral clock (Pck/2 or Pck/3))

BCR Setting Constraints

TSEG1 > TSEG2 ≥ SJW (SJW = 1 to 4)
TSEG1 + TSEG2 + 1 = 8 to 25 time quanta

These constraints allow the setting range shown in the table below for TSET1 and TSEG2 in the Bit Configuration Register.

Table 22.5 shows the settings of TSEG1 and TSEG2 in CANBCR1. That allow the above-described settings.

			TSEG2 (Bits 10 to 8 in CANBCR1)						
			001	010	011	100	101	110	111
			2	3	4	5	6	7	8
TSEG1 (Bits 15 to 12 in CANBCR1)	0011	4	No	Yes	No	No	No	No	No
	0100	5	Yes	Yes	Yes	No	No	No	No
	0101	6	Yes	Yes	Yes	Yes	No	No	No
	0110	7	Yes	Yes	Yes	Yes	Yes	No	No
	0111	8	Yes	Yes	Yes	Yes	Yes	Yes	No
	1000	9	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	1001	10	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	1010	11	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	1011	12	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	1100	13	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	1101	14	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	1110	15	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	1111	16	Yes	Yes	Yes	Yes	Yes	Yes	Yes

Examples:

1. To have a Bit rate of 1Mbps with a frequency of fclk = 40MHz, it is possible to set: BRP = 4, TSEG1 = 6, TSEG2 = 3.

   Then the configuration to write is CANBCR1 = 5200 and CANBCR0 = 0003.

2. To have a Bit rate of 500kpbs with a frequency of 35MHz, it is possible to set: BPR = 5, TSEG1 = 8, TSEG2 = 5.

   Then the configuration to write is CANBCR1 = 7400 and CANBCR0 = 0004.

CANIRR is a 16-bit read/write-clearable register containing status flags for the various interrupt sources.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	IRR14	IRR13	IRR12	-	-	IRR9	IRR8	IRR7	IRR6	IRR5	IRR4	IRR3	IRR2	IRR1	IRR0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W:	R	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	—	0	R	Reserved The write value should always be 0. The read value is not guaranteed.
14	IRR14	0	R/W	Timer Compare Match Interrupt Indicates that a Compare-Match condition occurred to CANTCMR. When the value set in CANTCMR matches the Timer value (CANTCMR =CANTCNTR) this bit is set. Please note that this bit is not set if the CANTCMR value is H'0000. 0: Timer Compare Match has not occurred for CANTCMR. Clearing condition: Write a 1 to this bit. 1: Timer Compare Match has occurred for CANTCMR. Setting condition: CANTCMR matches to the Timer value (CANCMR = CANTCNTR).
13	IRR13	0	R/W	Timer Overrun Interrupt Indicates that the Timer has overrun and is reset to 0. Please note that this bit is set even when the CANTCMR is enabled to clear-set the Timer value and its value is set to H'FFFF. 0: Timer has not overrun. Clearing condition: Write a 1 to this bit. 1: Timer has overrun. Setting condition: When the timer (CANTCNTR) changes from H'FFFF to H'0000.

12	IRR12	0	R/W	Wake-up on Bus Activity Indicates that a CAN bus activity is present. When the HCAN is in sleep mode and a recessive to dominant bit transition takes place on the CAN bus, this bit is set. The operation of this interrupt is configured in the Master Control Register. (MCR7 – Auto-wake Mode). This interrupt is cleared by writing a 1 to this bit position. Writing a 0 has no effect. 0: Bus idle state Clearing condition: Write a 1 to this bit. 1: CAN bus activity is detected in HCAN2 sleep mode. Setting condition: Bit transition, from recessive to dominant, is detected in sleep mode.
11, 10	—	All 0	R	Reserved The write value should always be 0. The read value is not guaranteed.
9	IRR9	0	R/W	Message Overrun/Overwrite Interrupt Flag Indicates that a message has been received but the existing message in the matching Mailbox has not been read due to the corresponding CANRXPR or CANRFPR set to 1. The received message is either abandoned (overrun) or overwritten depeding on the value of the NMC (New Message Control) bit. This bit is cleared by writing a 1 to the correspondent bit position in CANUMSR (Unread Message Status Register). Writing a 0 has no effect. 0: No message overrun/overwrite Clearing condition: Clean all bits in CANUMSR. 1: Receive message overrun and its storage has been rejected or message overwrite. Setting condition: Message is received while the corresponding CANRXPR or CANRFPR =1 and CANMBIMR = 0.

8	IRR8	0	R/W	Mailbox Empty Interrupt Flag Indicates that message transmission or transmission cancellation has been successfully ended, and the Mailbox is now ready to accept a new message data for the next transmission. This bit is set when at least one CANTXPR bit is cleared. This bit is set by an OR'ed signal of the CANTXACK and CANABACK bits, therefore, this bit is automatically cleared when all the CANTXACK and CANABACK bits are cleared. Writing a 0 has no effect. Note that this bit does not indicate that all CANTXPR bits are reset, whereas GSR2 does. 0: Messages set for transmission or transmission cancellation is not in progress. Clearing condition: All the CANTXACK and CANABACK bits are cleared. 1: Message has been transmitted or aborted, and a new message can be stored. Setting condition: One of the CANTXPR bits is cleared by completion of transmission or completion of transmission abort (i.e. in case of CANMBIMR = 0, the CANTXACK or CANABACK bit is set).
7	IRR7	0	R/W	Overload Frame Indicates that the HCAN2 has transmitted an overload frame. It remains latched until reset by writing a 1 to this bit position. Writing a 0 has no effect. 0: Clearing condition: Write a 1 to this bit. 1: Setting condition: Overload frame is transmitted.

6	IRR6	0	R/W	Bus Off Interrupt Flag This bit is set when the HCAN2 enters the Bus-off state or when the HCAN2 leaves Bus-off state and returns to Error-Active. This is because that the existing condition is that 11 recessive bits x 128 have been received at the node of TEC ≥ 256 or at the end of Bus-off. This bit remains latched even though the HCAN2 node leaves the Bus-off condition, and needs to be explicitly cleared by software. The software is expected to read the GSR0 to judge whether HCAN2 has become Bus-off or error active, GSR0 should be read. This bit is cleared by writing a 1. Writing a 0 has no effect. 0: Clearing condition: Write a 1 to this bit. 1: Bus off state caused by a transmit error or error active state returned from Bus-off. Setting condition: TEC ≥ 256 or the end of bus-off after receiving 128 x 11 bits
5	IRR5	0	R/W	Error Passive Interrupt Flag Indicates the Error Passive state caused by the transmit/receive error counter. This bit is cleared by writing a 1, while writing a 0 has no effect. If this bit is cleared, the node may still be error passive. 0: Clearing condition: Write a 1 to this bit. 1: Error passive state is caused by a transmit/receive error. Setting condition: TEC ≥ 128 or REC ≥ 128
4	IRR4	0	R/W	Receive Overload Warning Interrupt Flag This bit becomes set and latches if the receive error counter (REC) reaches a value greater than 96. This bit is cleared by writing a 1. Writing a 0 has no effect. When the interrupt is cleared, the REC still holds its value greater than 96. 0: Clearing condition: Write a 1 to this bit. 1: Error warning state is caused by a receive error. Setting condition: REC ≥ 96

3	IRR3	0	R/W	Transmit Overload Warning Interrupt Flag This bit becomes set and latches if the transmit error counter (TEC) reaches a value greater than 96. This bit is cleared by writing a 1. Writing a 0 has no effect. When the interrupt is cleared, the TEC still holds a value greater than 96. 0: Clearing condition: Write a 1 to this bit. 1: Error warning state is caused by a transmit error. Setting condition: TEC ≥ 96
2	IRR2	0	R	Remote Frame Request Interrupt Flag Indicates that a Remote Frame has been received in a Mailbox. This bit is set if at least one receive mailbox contains a Remote Frame transmission request. This bit is cleared by ensuring all bits in the Remote Request Pending Register (CANRFPR) are cleared. Writing to this bit has no effect. 0: Clearing condition: Clearing all bits in CANRFPR. 1: At least one remote request is pending Setting conditions: When remote frame is received and the corresponding CANMBIMR = 0.
1	IRR1	0	R	Data Frame Received Interrupt Flag Indicates that there is a pending Data Frame received. If this bit is set at least on receive mailbox contains a pending message. This bit is cleared when all bits in the Receive Message Pending Register (CANRXPR) are cleared, i.e. there is no pending message in any receiving mailbox. It is a logical OR from each configured receive mailbox. Writing to this bit has no effect. 0: Clearing condition: Clearing all bits in CANRXPR. 1: Data Frame is received and stored in Mailbox Setting conditions: When data is received and the corresponding CANMBIMR = 0.

0	IRR0	1	R/W	Reset/Halt/Sleep Interrupt Flag Indicates that the CAN Interface has been reset or halted and the HCAN2 is now in Configuration mode or HCAN2 is asleep. An interrupt signal will be generated through this bit to notify the change of the HCAN2's state to the host processor if a MCR0 (Software reset) or MCR1 (Halt) or MCR5 (Sleep) request is made. The GSR may be read after this bit is set to figure out which state HCAN2 is in. Important: When a Sleep mode request needs to be made, the Halt mode should be used beforehand. Refer to the MCR5 description. 0: Clearing condition: Write a 1 to this bit. 1: Transition to Software reset mode, Halt mode, or Sleep mode. Setting condition: When reset/halt processing completed after Software reset (MCR0) or Halt mode (MCR1) or Sleep mode (MCR5) is requested.

22.5.5 Interrupt Mask Register (CANIMR)

CANIMR is a 16-bit register that prevents all interrupts corresponding interrupts in the CANIRR from generating on output signal on the IRQ. An interrupt request is masked if the corresponding bit position is set to 1. This register can be read or written at any time. The CANIMR directly controls the generation of IRQ, but does not prevent the setting of the corresponding bit in the CANIRR.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	IMR14	IMR13	IMR12	-	-	IMR9	IMR8	IMR7	IMR6	IMR5	IMR4	IMR3	IMR2	IMR1	IMR0
Initial value:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W:	R	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

15, 11, 10	—	All 1	R	Reserved The write value should always be 1. The read value is not guaranteed.
14 to 12, 9 to 0	IMR14 to IMR12, IMR9 to IMR0	All 1	R/W	Masks interrupt sources corresponding to IRR14 to IRR12, IRR9 to IRR0. When the bit is set, the interrupt is masked, however, the CANIRR bit setting is retained. 0: Corresponding CANIRR bit is not masked (IRQ is generated for interrupt conditions). 1: Corresponding interrupt of CANIRR bit is masked.

22.5.6 Transmit Error Counter and Receive Error Counter (CANTECREC)

CANTECREC is 16 bit read/(write) register and consists of the transmit error counter (TEC) and receive error counter (REC) that function as a counter indicating the number of transmit/receive message errors on the CAN interface. The counter value is stipulated in CAN Specification Version 2.0, Robert Bosch GmbH, 1991 and Implementation Guide for the CAN Protocol, CAN Specification 2.0 Addendum, CAN In Automation, Erlangen, Germany. In the normal mode, this register is read-only and can only be modified by the CAN interface. This register can be cleared by a Reset request (MCR0) or Bus off.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TEC7	TEC6	TEC5	TEC4	TEC3	TEC2	TEC1	TEC0	REC7	REC6	REC5	REC4	REC3	REC2	REC1	REC0
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	Bit Name	Initial Value	R/W	Description
15 to 8	TEC7 to TEC0	All 0	R/W*	Transmit error counter
7 to 0	REC7 to REC0	All 0	R/W*	Receive error counter

Note: * It is possible to write the value only in test mode with MCR15 = MCR14 = 1.

The CANTXPR registers are two 16-bit read/conditionally-write registers that contain any transmit pending flags for the CAN module. CANTXPR1 controls Mailboxes 31 to 16, and CANTXPR0 controls Mailboxes 15 to 1. The host CPU may set the CANTXPR bits to affect any message being considered for transmission by writing a 1 to the corresponding bit location. Writing a 0 has no effect. CANTXPR cannot be cleared by writing a 0, but it must be cleared by setting the corresponding CANTXCR bits. CANTXPR may be read by the host CPU to determine which, if any, transmissions are pending. There is a transmit pending bit for all Mailboxes except for the Mailbox 0. Writing a 1 to a bit location when the mailbox is configured to receive will have no effect, and will be automatically cleared when an internal arbitration for transmission runs.

The HCAN2 will clear a transmit pending flag after successful transmission of its corresponding or when a transmission abort is normally requested from TXCR. The CANTXPR flag is not cleared if the message is not transmitted due to the CAN node losing the arbitration process or due to errors on the CAN bus, and HCAN2 automatically tries to transmit it again until its DART (Disable Automatic Re-Transmission) bit is set in the Message-Control of the corresponding Mailbox. In such case (DART is set), the transmission is cleared and notified through Mailbox Empty Interrupt Flag (IRR8) and the correspondent bit in the Abort Acknowledgement Register (CANABACK).

If the status of CANTXPR changes, the HCAN2 shall ensure that in the identifier priority scheme (MCR2 = 0), the highest priority message is always presented for transmission in an intelligent way even under circumstances such as bus arbitration losses or errors on the CAN bus. Please refer to section 22.6 Operation in details.

When the HCAN2 changes state of nay CANTXPR bit position to 0, an empty slot interrupt (IRR8) may be generated. This indicates that either successful or an aborted mailbox transmission has been made. If a message transmission is successful it is signaled in the CANTXACK, and if a message transmission abortion is successful it is signaled in the CANABACK. By checking these registers, the contents of the Message-Data of the corresponding Mailbox may be modified to prepare for the next transmission.

- CANTXPR1

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TXPR1_15	TXPR1_14	TXPR1_13	TXPR1_12	TXPR1_11	TXPR1_10	TXPR1_9	TXPR1_8	TXPR1_7	TXPR1_6	TXPR1_5	TXPR1_4	TXPR1_3	TXPR1_2	TXPR1_1	TXPR1_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*

15 to 0	TXPR1[15:0]	All 0	R/W*	Indicates that the corresponding Mailbox is requested to transmit a CAN Frame. Bit15 to 0 correspond to Mailboxes 31 to 16, respectively. When multiple bits are set, the order of the transmissions is governed by the MCR2 (CAN ID or Mailbox number. 0: Transmit message idle state in corresponding mailbox. Clearing condition: Completion of message transmission or message transmission abortion (automatically cleared) 1: Transmission request made fo corresponding mailbox

Note: * A write of 1 only is allowed to the Mailbox designated for transmission.

- CANTXPR0

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TXPR0_15	TXPR0_14	TXPR0_13	TXPR0_12	TXPR0_11	TXPR0_10	TXPR0_9	TXPR0_8	TXPR0_7	TXPR0_6	TXPR0_5	TXPR0_4	TXPR0_3	TXPR0_2	TXPR0_1	-
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

Bit	Bit Name	Initial Value	R/W	Description
15 to 1	TXPR0[15:1]	All 0	R/W*	Indicating that the corresponding Mailbox is requested to transmit a CAN Frame. Bits15 to 1 correspond to Mailboxes 15 to 1, respectively. When multiple bits are set, the order of the transmission is governed by the MCR2 (CAN-ID or Mailbox number). 0: Transmit message idle state in corresponding mailbox. Clearing condition: Completion of message transmission or message transmission abortion (automatically cleared) 1: Transmission request for the corresponding Mailbox
0	—	0	R	Reserved This bit is always 0 as this Mailbox is receive-only. Writing a 0 to this bit has no effect. The read value is not guaranteed.

Note: * A write of 1 only is allowed for the Mailbox designated for transmission.

The CANTXCR are two 16-bit read/conditionally-write registers. The CANTXCR1 controls Mailbox 31 to Mailbox 16, and the CANTXCR0 controls Mailbox 15 to Mailbox 1. These registers are used by the microprocessor to request the pending transmission requests in the CANTXPR to be cancelled. To clear the corresponding bit in CANTXPR, the host CPU must write a 1 to the corresponding CANTXCR bit. Writing a 0 has no effect.

When an abort has succeeded, the CAN controller clears the corresponding CANTXPR and CANTXCR bits, and sets the corresponding CANABACK bit. However, once a Mailbox has started transmission, the transmission cannot be canceled by this bit. In such case, if the transmission is finishes in success, the CAN controller clears the corresponding CANTXPR and CANTXCR bits, and sets the corresponding CANTXACK bit, however, if the transmission fails due to a bus arbitration loss or an error on the bus, the CAN controller clears the corresponding CANTXPR and CANTXCR bits, and sets the corresponding CANABACK bit. If an attempt is made by the host CPU to clear Mailbox transmission that is not transmit-pending, it shall have no effect, and will be automatically cleared when an internal arbitration for transmission runs.

- CANTXCR1

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TXCR1_15	TXCR1_14	TXCR1_13	TXCR1_12	TXCR1_11	TXCR1_10	TXCR1_9	TXCR1_8	TXCR1_7	TXCR1_6	TXCR1_5	TXCR1_4	TXCR1_3	TXCR1_2	TXCR1_1	TXCR1_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	Bit Name	Initial Value	R/W	Description
15 to 0	TXCR1[15:0]	All 0	R/W*	Requests the corresponding Mailbox that is in the queue for transmission, to cancel its transmission. Bits 15 to 0 correspond to Mailboxes 31 to 16 (and TXPR1[15:0]) respectively. 0: Transmit message cancellation idle state in corresponding mailbox. Clearing condition: Completion of transmit message cancellation (automatically cleared) 1: Transmission cancellation request made for the corresponding Mailbox.

Note: * A write of 1 only is allowed for the Mailbox designated for transmission when it is in the wait state.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TXCR0_15	TXCR0_14	TXPR0_13	TXCR0_12	TXCR0_11	TXCR0_10	TXCR0_9	TXCR0_8	TXCR0_7	TXCR0_6	TXCR0_5	TXCR0_4	TXCR0_3	TXCR0_2	TXCR0_1	-
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

Bit	Bit Name	Initial Value	R/W	Description
15 to 1	TXCR0[15:1]	All 0	R/W*	Requests the corresponding Mailbox that is in the queue for transmission, to cancel its transmission. Bits 15 to 1 correspond to Mailboxes 15 to 1 (and TXPR1[15:1]) respectively. 0: Transmit message cancellation idle state in corresponding mailbox. Clearing condition: Completion of transmit message cancellation (automatically cleared) 1: Transmission cancellation request made for the corresponding Mailbox.
0	—	0	R	Reserved This bit is always 0 as this Mailbox is receive-only. Writing a 1 to this bit has no effect. This bit is always read as 0.

Note: * A write of 1 only is allowed for the Mailbox designated for transmission when it is in the wait state.

The CANTXACK are two16-bit read/conditionally-write registers that are used to signal to the CPU that a Mailbox transmission has been successfully made. When transmission has succeeded, the HCAN2 sets the corresponding bit in CANTXACK. The host CPU can clear the CANTXACK bit by writing a 1 to the corresponding bit location. Writing a 0 has no effect.

- CANTXACK1

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TXACK1_15	TXACK1_14	TXACK1_13	TXACK1_12	TXACK1_11	TXACK1_10	TXACK1_9	TXACK1_8	TXACK1_7	TXACK1_6	TXACK1_5	TXACK1_4	TXACK1_3	TXACK1_2	TXACK1_1	TXACK1_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	Bit Name	Initial Value	R/W	Description
15 to 0	TXACK1[15:0]	All 0	R/W*	Notifies that requested transmission of the corresponding Mailbox has been finished successfully. Bits 15 to 0 correspond to Mailboxes 31 to 16 respectively. 0: Clearing condition: Write a 1 to this bit. 1: The corresponding Mailbox has successfully transmitted message (Data or Remote Frame). Setting condition: Completion of message transmission for the corresponding Mailbox

Note: * Only a write of 1 is allowed to clear the bit.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TXACK0_15	TXACK0_14	TXACK0_13	TXACK0_12	TXACK0_11	TXACK0_10	TXACK0_9	TXACK0_8	TXACK0_7	TXACK0_6	TXACK0_5	TXACK0_4	TXACK0_3	TXACK0_2	TXACK0_1	-
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

Bit	Bit Name	Initial Value	R/W	Description
15 to 1	TXACK0[15:1]	All 0	R/W*	Notifies that requested transmission of the corresponding Mailbox has been finished successfully. Bits 15 to 1 correspond to Mailboxes 15 to 1 respectively. 0: Clearing condition: Write a 1 to this bit. 1: The corresponding Mailbox has successfully transmitted messages (Data or Remote Frame). Setting condition: Completion of message transmission for the corresponding Mailbox
0	—	0	R	Reserved This bit is always 0 as this Mailbox is receive-only. Writing a 1 to this bit has no effect. This bit is always read as 0.

Note: * Only a write of 1 only is allowed to clear the bit.

22.5.10 Abort Acknowledge Registers 1 and 0 (CANABACK1, CANABACK0)

The CANABACK registers are two 16-bit read/conditionally-write registers that are used to signal to the CPU that a mailbox transmission has been aborted as per its each request. When an abort has succeeded, the HCAN2 sets the corresponding bit in CANABACK. The host CPU may clear the Abort Acknowledge bit by writing a 1 to the corresponding bit. Writing a 0 has no effect. A CANABACK bits position is set by the HCAN2 to acknowledge that a CANTXPR bit has been cleared by the corresponding CANTXCR bit.

- CANABACK1

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ABACK1_15	ABACK1_14	ABACK1_13	ABACK1_12	ABACK1_11	ABACK1_10	ABACK1_9	ABACK1_8	ABACK1_7	ABACK1_6	ABACK1_5	ABACK1_4	ABACK1_3	ABACK1_2	ABACK1_1	ABACK1_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*

15 to 0	ABACK1[15:0]	All 0	R/W*	Notifies that requested transmission cancellation requested of the corresponding Mailbox has been performed successfully. Bits 15 to 0 correspond to Mailboxes 31 to 16 respectively. 0: Clearing condition: Write a 1 to this bit. 1: Corresponding Mailbox has cancelled transmission of message (Data or Remote Frame). Setting condition: Completion of transmission cancellation for the corresponding Mailbox

Note: * Only a write of 1 is allowed to clear the bit.

- CANABACK0

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ABACK0_15	ABACK0_14	ABACK0_13	ABACK0_12	ABACK0_11	ABACK0_10	ABACK0_9	ABACK0_8	ABACK0_7	ABACK0_6	ABACK0_5	ABACK0_4	ABACK0_3	ABACK0_2	ABACK0_1	-
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

Bit	Bit Name	Initial Value	R/W	Description
15 to 1	ABACK0[15:1]	All 0	R/W*	Notifies that requested transmission cancellation requested of the corresponding Mailbox has been performed successfully. Bits 15 to 1 correspond to Mailboxes 15 to 1 respectively. 0: Clearing condition: Write a 1 to this bit. 1: Corresponding Mailbox has cancelled transmission of message (Data or Remote Frame). Setting condition: Completion of transmission cancellation for the corresponding Mailbox
0	—	0	R	Reserved This bit is always 0 as this is receive-only. Writing a 1 to this bit has no effect. This bit is always read as 0.

Note: * Only a write of 1 is allowed to clear the bit.

The CANRXPR are two 16-bit read/conditionally-write registers that contain the receive Data Frame pending flags associated with the configured Receive Mailboxes. When a CAN Data Frame is successfully stored in a receive Mailbox, the corresponding bit is set in CANRXPR. The bit may be cleared by writing a 1 to the corresponding bit position. Writing a 0 has no effect. However, the bit may only be set if the Mailbox is configured by its MBC (Mailbox Configuration) to receive Data Frame. When a CANRXPR bit is set, it also set IRR1 (Data Frame Received Interrupt Flag) if its CANMBIMR (Mailbox Interrupt Mask Register) is not set, and the interrupt signal is generated if IMR1 is not set. Please note that those bits are only set by receiving Data Frame and not by receiving Remote Frame.

- CANRXPR1

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RXPR1_15	RXPR1_14	RXPR1_13	RXPR1_12	RXPR1_11	RXPR1_10	RXPR1_9	RXPR1_8	RXPR1_7	RXPR1_6	RXPR1_5	RXPR1_4	RXPR1_3	RXPR1_2	RXPR1_1	RXPR1_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	Bit Name	Initial Value	R/W	Description
15 to 0	RXPR1[15:0]	All 0	R/W*	Configurable receive Mailbox locations corresponding to Mailbox position from 31 to 16 respectively. 0: Clearing condition: Write a 1 to this bit. 1: Corresponding Mailbox received a CAN Data Frame. Setting condition: Completion of Data Frame receive on corresponding Mailbox

Note: * Only a write of 1 is allowed to clear the bit.

- CANRXPR0

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RXPR0_15	RXPR0_14	RXPR0_13	RXPR0_12	RXPR0_11	RXPR0_10	RXPR0_9	RXPR0_8	RXPR0_7	RXPR0_6	RXPR0_5	RXPR0_4	RXPR0_3	RXPR0_2	RXPR0_1	RXPR0_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*

15 to 0	RXFPR0[15:0]	All 0	R/W*	Configurable receive Mailbox locations corresponding to Mailbox position from 15 to 0 respectively. 0: Clearing condition: Write a 1 to this bit. 1: Corresponding Mailbox received a CAN Data Frame. Setting condition: Completion of Data Frame receive on corresponding Mailbox

Note: * Only a write of 1 is allowed to clear the bit.

22.5.12 Remote Frame Request Pending Registers 1 and 0 (CANRFPR1, CANRFPR0)

The CANRFPR are two 16-bit read/conditionally-write registers that contains the received Remote Frame pending flags associated with the configured Receive Mailboxes. When a CAN Remote Frame is successfully stored in a receive Mailbox, the corresponding bit is set in CANRFPR. The bit may be cleared by writing a 1 to the corresponding bit position. Writing a 0 has no effect. There is a bit position for all mailboxes. However, the bit may only be set if the Mailbox is configured by its MBC (Mailbox Configuration) to receive Remote Frames. When a CANRFPR bit is set, IRR2 (Remote Frame Request Interrupt Flag) is also set CANMBIMR (Mailbox Interrupt Mask Register) is not set, and the interrupt signal is generated if IMR2 is not set. Note that these bits are only set by receiving Remote Frames and not by receiving Data Frames.

- CANRFPR1

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RFPR1_15	RFPR1_14	RFPR1_13	RFPR1_12	RFPR1_11	RFPR1_10	RFPR1_9	RFPR1_8	RFPR1_7	RFPR1_6	RFPR1_5	RFPR1_4	RFPR1_3	RFPR1_2	RFPR1_1	RFPR1_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	Bit Name	Initial Value	R/W	Description
15 to 0	RFPR1[15:0]	All 0	R/W*	Remote request pending flags for Mailboxes 31 to 16 respectively. 0: Clearing condition: Write a 1 to this bit. 1: Corresponding Mailbox received Remote Frame. Setting condition: Completion of remote frame reception in the corresponding Mailbox

Note: * Only a write of 1 is allowed to clear the bit.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RFPR0_15	RFPR0_14	RFPR0_13	RFPR0_12	RFPR0_11	RFPR0_10	RFPR0_9	RFPR0_8	RFPR0_7	RFPR0_6	RFPR0_5	RFPR0_4	RFPR0_3	RFPR0_2	RFPR0_1	RFPR0_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	Bit Name	Initial Value	R/W	Description
15 to 0	RFPR0[15:0]	All 0	R/W*	Remote request pending flags for Mailboxes 15 to 0 respectively. 0: Clearing condition: Write a 1 to this bit. 1: Corresponding Mailbox received Remote Frame Setting condition: Completion of remote frame reception in the corresponding Mailbox

Note: * Only a write of 1 is allowed to clear the bit.

22.5.13 Mailbox Interrupt Mask Registers 1 and 0 (CANMBIMR1, CANMBIMR0)

The CANMBIMR are two 16-bit read/write registers. The CANMBIMR only prevents the setting of IRR related to the Mailbox activities (IRR1: Data Frame Received Interrupt, IRR2: Remote Frame Request Interrupt, IRR8: Mailbox Empty Interrupt, and IRR9: Message Overrun Interrupt). If a Mailbox is configured as receive, a mask at the corresponding bit position prevents the generation of receive interrupts (IRR1 and IRR2 and IRR9) but does not prevents the settings of the corresponding bit in CANRXPR or CANRFPR or CANUMSR. Similarly, when a mailbox has been configured for transmission, a mask prevents the generation of in Interrupt signal and setting of an Mailbox Empty Interrupt due to successful transmission or transmission abortion (IRR8), however, it does not prevent the HCAN2 from clearing the corresponding CANTXPR/CANTXCR bit and setting the CANTXACK bit for abortion of transmission.

A mask is set by writing a 1 to the corresponding bit for the Mailbox activity to be masked. At reset, all Mailbox interrupts are masked.

- CANMBIMR1

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MBIMR1_15	MBIMR1_14	MBIMR1_13	MBIMR1_12	MBIMR1_11	MBIMR1_10	MBIMR1_9	MBIMR1_8	MBIMR1_7	MBIMR1_6	MBIMR1_5	MBIMR1_4	MBIMR1_3	MBIMR1_2	MBIMR1_1	MBIMR1_0
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

15 to 0	MBIMR1[15:0]	All 1	R/W	Enable or disable interrupt requests from individual Mailboxes 31 to 16 respectively. 0: Interrupt requests from IRR1/IRR2/IRR8/ IRR9 enabled. 1: Interrupt requests from IRR1/IRR2/IRR8/ IRR9 disabled.

- CANMBIMR0

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MBIMR0_15	MBIMR0_14	MBIMR0_13	MBIMR0_12	MBIMR0_11	MBIMR0_10	MBIMR0_9	MBIMR0_8	MBIMR0_7	MBIMR0_6	MBIMR0_5	MBIMR0_4	MBIMR0_3	MBIMR0_2	MBIMR0_1	MBIMR0_0
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

Bit	Bit Name	Initial Value	R/W	Description
15 to 0	MBIMR0[15:0]	All 1	R/W	Enable or disable interrupt requests from individual Mailboxes 15 to 0 respectively. 0: Interrupt requests from IRR1/IRR2/IRR8/ IRR9 enabled. 1: Interrupt requests from IRR1/IRR2/IRR8/ IRR9 disabled.

22.5.14 Unread Message Status Registers 1 and 0 (CANUMSR1, CANUMSR0)

The CANUMSR are two 16-bit read/write registers and record any receive mailboxes that have been emptied prior to a new message received. If the host CPU has not cleared the corresponding bit in CANRXPR or CANRFPR when a new message for that mailbox is received, the corresponding CANUMSR bit is set to 1. This bit may be cleared by writing a 1 to the corresponding bit location in the CANUMSR. Writing a 0 has no effect.

If a mailbox is configured as a transmit box, the corresponding CANUMSR bit will not be set.

- CANUMSR1

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	UMSR1_15	UMSR1_14	UMSR1_13	UMSR1_12	UMSR1_11	UMSR1_10	UMSR1_9	UMSR1_8	UMSR1_7	UMSR1_6	UMSR1_5	UMSR1_4	UMSR1_3	UMSR1_2	UMSR1_1	UMSR1_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

15 to 0	UMSR1[15:0]	All 0	R/W	Indicate that an unread received message has been overwritten for Mailboxes 31 to 16. 0: Clearing condition: Write a 1 to this bit. 1: Unread received message is overwritten by a new message or overrun condition. Setting condition: A new message is received before CANRXPR or CANRFPR is cleared.

- CANUMSR0

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	UMSR0_15	UMSR0_14	UMSR0_13	UMSR0_12	UMSR0_11	UMSR0_10	UMSR0_9	UMSR0_8	UMSR0_7	UMSR0_6	UMSR0_5	UMSR0_4	UMSR0_3	UMSR0_2	UMSR0_1	UMSR0_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	Bit Name	Initial Value	R/W	Description
15 to 0	UMSR0[15:0]	All 0	R/W	Indicate that an unread received message has been overwritten for Mailboxes 15 to 0. 0: Clearing condition: Write a 1 to this bit. 1: Unread received message is overwritten by a new message or overrun condition. Setting condition: A new message is received before CANRXPR or CANRFPR is cleared.

22.5.15 Timer Counter Register (CANTCNTR)

CANTCNTR is a 16-bit read/write register that allows the CPU to monitor and modify the value of the Free Running Timer Counter. When the Timer rolls over or meets CANTCMR and TCR11 is set to 1, CANTCNTR is set to 0 and starts running again.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TCNTR 15	TCNTR 14	TCNTR 13	TCNTR 12	TCNTR 11	TCNTR 10	TCNTR 9	TCNTR 8	TCNTR 7	TCNTR 6	TCNTR 5	TCNTR 4	TCNTR 3	TCNTR 2	TCNTR 1	TCNTR 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	Bit Name	Initial Value	R/W	Description
15 to 0	TCNTR[15:0]	All 0	R/W*	Indicates the value of 16 bit Free Running Timer

Note: * The register can be cleared by the Compare Match condition.

CANTCR is a 16-bit read/write register and provides functions to control the operation of the Timer.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TCR15	-	TCR13	TCR12	TCR11	-	-	-	-	-	TPSC5	TPSC4	TPSC3	TPSC2	TPSC1	TPSC0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R	R/W	R/W	R/W	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	TCR15	0	R/W	Enable Timer When this bit is set, the timer is running. When this bit is cleared the timer completes the current cycle (notified by Timer overrun or a compare match on CANTCMR) and is cleared to 0. 0: Timer stops running and is cleared at the end of the current cycle. 1: Timer is running.
14	—	0	—	Reserved The write value should always be 0. The read value is not guaranteed.
13	TCR13	0	R/W	TimeStamp Control for Reception Specifies if the Timestamp in the message control of each Mailbox is recorded at the StartOfFrame (SOF) or EndOfFrame (EOF). 0: Timestamp is recorded at the SOF of every message received. 1: Timestamp is recorded at the EOF of every message received.
12	TCR12	0	R/W	TimeStamp Control for Transmission Specifies if the Timestamp of each transmit Mailbox is recorded at the point that the corresponding CANTXPR bit is set or the corresponding CANTXACK is set when a transmission request is made. 0: Timestamp is recorded at the point that the CANTXPR bit is set for message transmission. 1: Timestamp is recorded at the point that the CANTXACK bit is set for message transmission.

11	TCR11	0	R/W	Timer Clear-Set- Control by CANTCMR Specifies if the Timer is to be cleared and set to 0 when CANTCMR matches to CANTCNTR. Please note that the CANTCMR is also capable to generate an interrupt signal to the host CPU via IRR15. 0: Timer is not cleared by CANTCMR. 1: Timer is cleared by CANTCMR.
10 to 6	—	All 0	—	Reserved The write value should always be 0. The read value cannot be guaranteed.
5	TCR5	0	R/W	HCAN2 Timer Prescaler This control fields allows the timer source clock (2 × Peripheral clock) to be divided before it is used for the timer. The following relationship exists between the source clock period and timer period: 000000: 1 × Source clock 000001: 2 × Source clock 000010: 4 × Source clock 000011: 6 × Source clock 000100: 8 × Source clock : 111111: 126 × Source clock
4	TCR4	0	R/W	
3	TCR3	0	—	
2	TCR2	0	—	
1	TCR1	0	—	
0	TCR0	0	—	

The CANTCMR is a 16-bit read/write register capable of generating interrupt signals and clearing/ setting the timer value.

If a compare match occurs, an interrupt flag is set to Bit14 in CANIRR and this bit cannot be prevented from being set in CANIRR except when the CANTCMR value is H'0000. Bit14 of CANIMR can prevent the generation of the interrupt signal. When Bit11 of CANTCR is set to 1, a compare match with CANTCMR will clear the timer to 0 (Timer Clear/ Set function).

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TCMR 15	TCMR 14	TCMR 13	TCMR 12	TCMR 11	TCMR 10	TCMR 9	TCMR 8	TCMR 7	TCMR 6	TCMR 5	TCMR 4	TCMR 3	TCMR 2	TCMR 1	TCMR 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	Bit Name	Initial Value	R/W	Description
15 to 0	TCMR[15:0]	0	R/W*	Indicates the value of Free Running Timer.

Note: * This register is cleared by the Compare Match condition.

22.6.1 Test Mode Settings

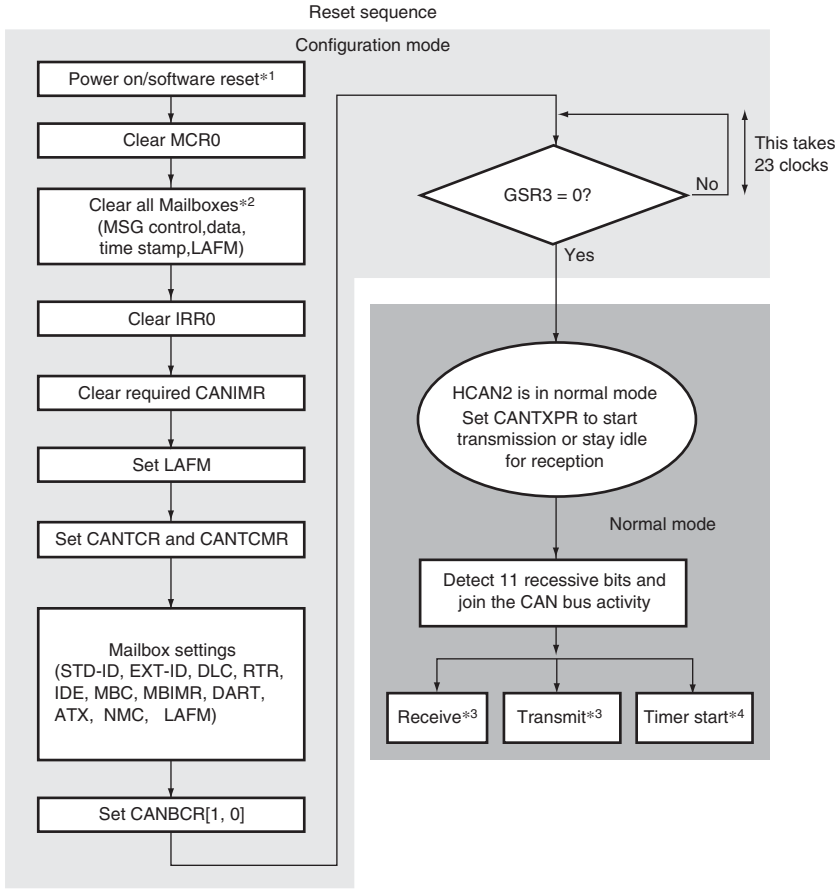
The HCAN2 has various test modes. The register TST[7:0] (MCR[15:8]) is used to select the HCAN2 test mode. The default (initialized) settings allow HCAN2 to operate in Normal mode. The following table is examples for test modes.

Bit 15: TST7	Bit 14: TST6	Bit 13: TST5	Bit 12: TST4	Bit 11: TST3	Bit 10: TST2	Bit 9: TST1	Bit 8: TST0	Description
0	0	0	0	0	0	0	0	Normal mode (initial value)
1	0	0	0	1	0	1	0	Listen-Only Mode (Receive-Only Mode)
1	0	0	1	—	0	0	0	Self Test Mode 1 (External)
1	0	0	1	—	1	1	1	Self Test Mode 2 (Internal)
1	1	0	—	—	—	—	—	Error Passive Mode 1
1	—	1	—	—	—	—	—	Error Passive Mode 2

- Normal Mode: HCAN2 operates in the normal mode.
- Listen-Only Mode: ISO-11898 requires this mode for baud rate detection etc. The Error Counters are disabled so that TEC/REC does not increase the value, and the Tx Output is disabled so that HCAN2 does not generate error frames.
- Self Test Mode 1: HCAN2 generates its own Acknowledge bit. The Rx/Tx pin must be connected to the CAN bus.
- Self Test Mode 2: HCAN2 generates its own Acknowledge bit. The Rx/Tx pin does not need to be connected to the CAN bus or any external devices, as the internal Tx is looped back to the internal Rx.
- Error Passive Mode 1: HCAN2 can be forced to become Error Passive node by writing a value (greater than 127) into the Error Counter (MCR1 must be 1 when writing to the Error Counter). The value written into TEC is written into REC, so only the same value can be set to these registers. Also, HCAN2 needs to be put into Halt Mode when writing into TEC/REC.
- Error Passive Mode 2: HCAN2 can be forced to become an Error Passive node by setting the TST5.

- Reset Sequence

  Following sequence is an example to configure the HCAN2 after Hardware/Software Reset. After reset, all the registers are initialized, therefore, the HCAN2 needs to be configured before joining the CAN bus activity. Read the notes carefully.



Notes: 1. Software reset could be performed at any time by setting MCR0=1.
2. Mailboxes are comprised of RAMs. Therefore, initialize all the mailboxes first even if some of them are not used.
3. If CANTXPR is not set, HCAN2 will receive the next incoming message. If CANTXPR is set, HCAN2 will start transmission of the message and will be arbitrated by the CAN bus. If it loses the arbitration, it will become a receiver.
4. Timer can be started at any time after CANTCR is set.

Figure 22.5 Reset Sequence

(1) Event Trigger Transmission

- Message Transmission Request

  The following is an example to transmit a CAN frame onto the bus. Note that IRR8 is set when one of the CANTXACK or CANABACK bits is set, meaning one of the Mailbox has completed its transmission or transmission abortion and is now ready to be updated for the next transmission, whereas the GSR3 means that there is currently no transmission request made (CANTXPR = H'0000).

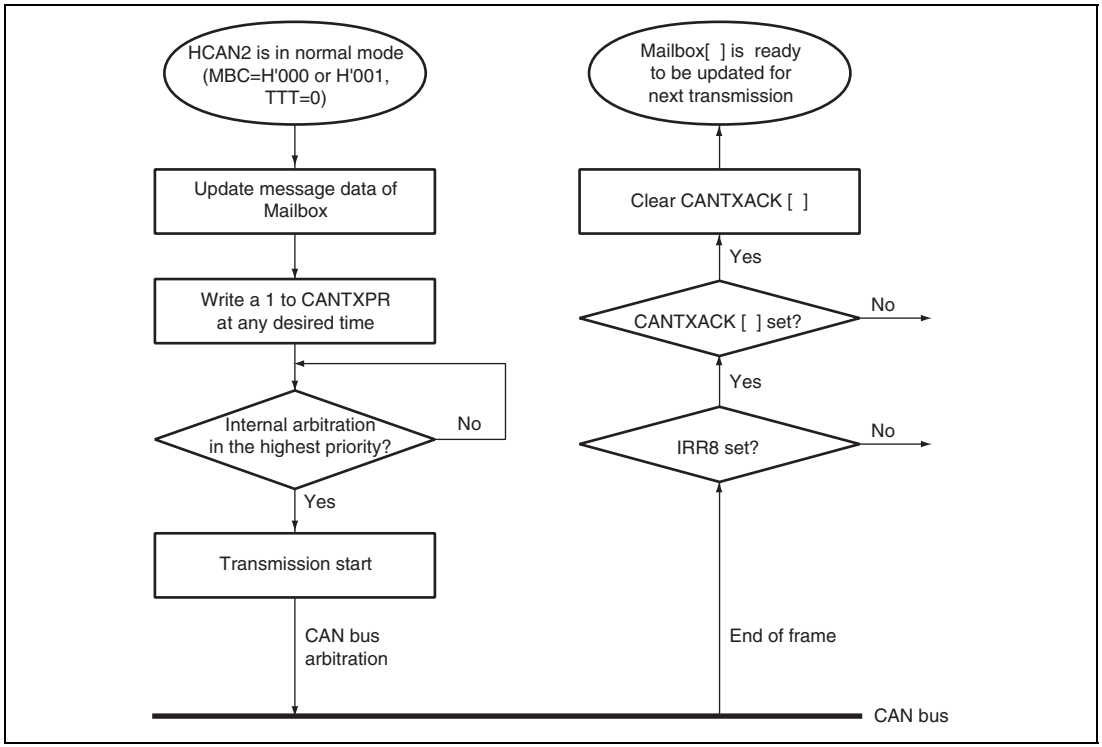


Figure 22.6 Transmission Request

- Internal Arbitration for Transmission

  The following diagram explains how HCAN2 manages to schedule transmission-requested messages in the correct order based on the CAN identifier. ‘Internal arbitration' picks up the highest priority message among messages transmitted-request messages.

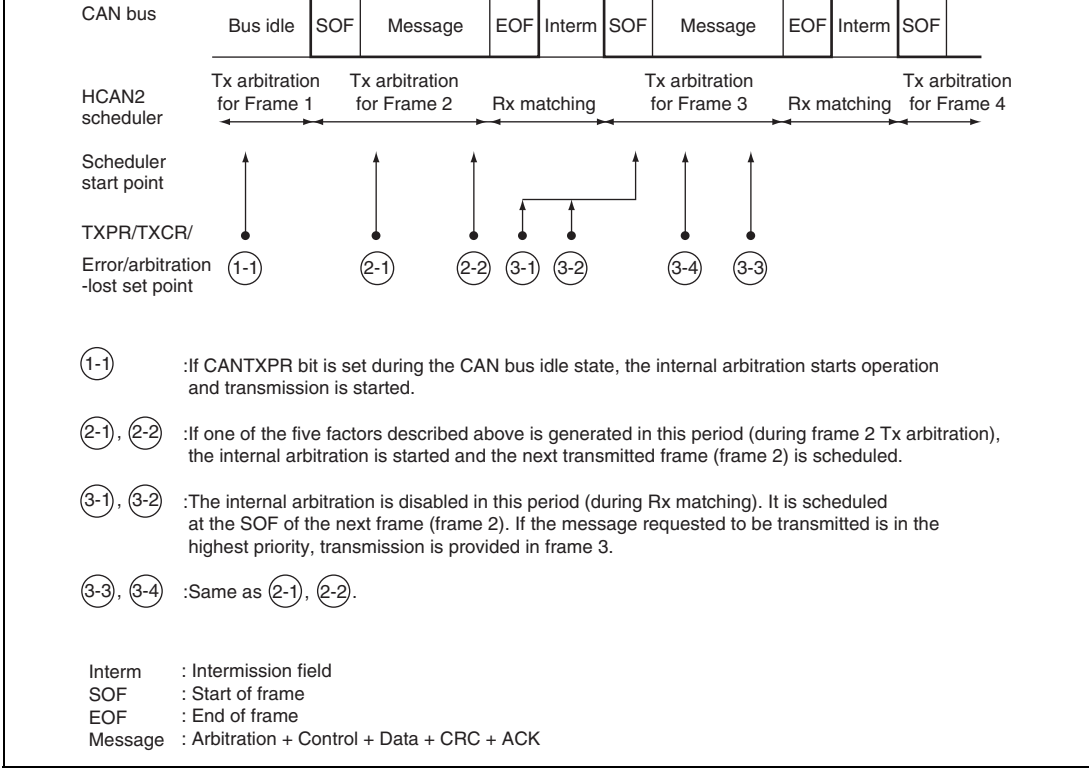


Figure 22.7 Internal Arbitration for Transmission

The HCAN2 Scheduler, which runs internal arbitration, has two states, Tx Arbitration State and Rx Matching State. HCAN2 Scheduler is in the Rx Matching State if the CAN bus is in the EOF or Intermission cycle, or otherwise is in the Tx Arbitration State. When a transmission (or transmission abortion) request is made in the Tx Arbitration State, the internal arbitration starts immediately. When a transmission (or transmission abortion) request is made in the Rx Matching State, the internal arbitration waits until the Rx Matching State (i.e. Intermission field) is finished, and then starts running as soon as the HCAN2 scheduler state becomes the ‘Tx arbitration state'.

There are 5 factors that can run internal arbitration, which are:

- CANTXPR is set.
- CANTXCR is set (not that, if CANTXCR is set for the message currently under transmission, HCAN2 does not stop the transmission but completes. If the message loses the bus arbitration or causes an error on the bus, the HCAN2 will cancel the transmission request.)
- Error occurs on the CAN bus.

- Mailbox with the setting of MBC = B'001 receives a Remote Frame.

Whenever these factors happen, the internal arbitration starts running to ensure that the highest priority message is always transmitted first. The followings are examples set in the diagram.

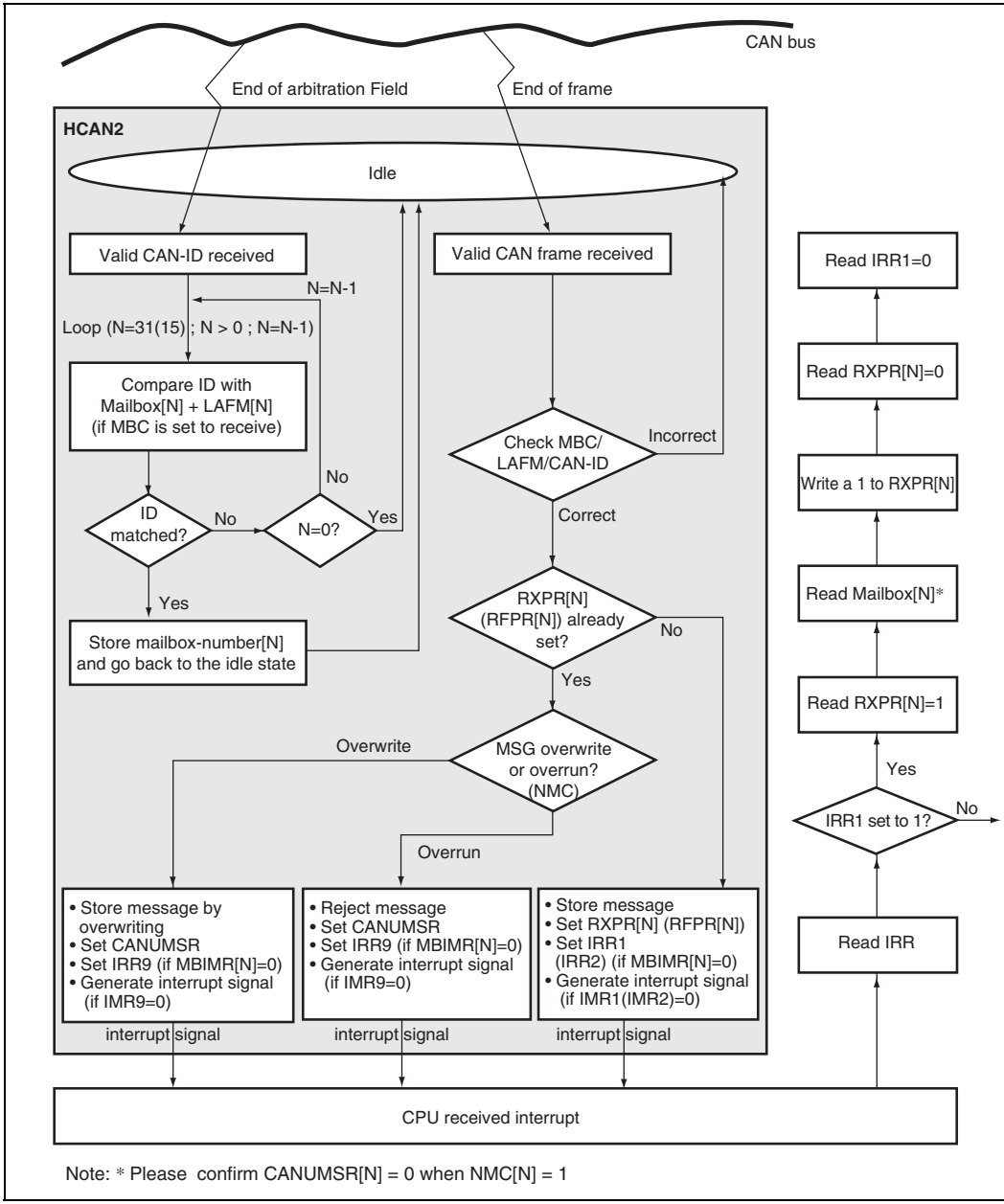
1-1 : When a TXPR bit(s) is set while the CAN bus is idle, the internal arbitration starts running immediately and the transmission is started.

2-1, 2-2 : During this period (Tx Arb for Frame-2), whenever or however many times any of the 4 factors occurs, the internal arbitration starts running and scheduled for the next frame (Frame-2) to be transmitted.

3-1, 3-2 : During this period (Rx Matching), any internal arbitration is not allowed to run, but scheduled later at the SOF of the next frame (Frame-2). If the transmit-requested message has the highest priority, the transmission will be set for the Frame-3.

3-3, 3-4 : This is the same case as 2-1,2-2.

The diagram below shows the message receive sequence.



Note: * Please confirm CANUMSR[N] = 0 when NMC[N] = 1

Figure 22.8 Message Receive Sequence

comparing the received identifier to the identifiers set in Mailboxes, starting from Mailbox 31 down to Mailbox 0. It first checks the MBC if it is configured as a receive box, and reads LAFM, and reads the CAN-ID of Mailbox 31 (if configured as receive) to finally compares them to the received ID. If it does not match, the same check takes place at Mailbox 30 (if configured as a receive). Once HCAN2 finds a matching identifier, it stores the number of Mailbox [N] into an internal buffer, stops the search, and gets back to idle state, waiting for an EndOfFrame (EOF) to come. When an EOF is notified by the CAN interface logic, HCAN2 this time only reads the MBC, LAFM and CAN-ID of Mailbox [N] to confirm the matching condition again (i.e., there has been no modification to the configuration of Mailbox [N]). This re-confirmation guarantees the data consistency even when a Mailbox is re-configured during receiving message. If it still matches, then the message is written or abandoned, depending on the NMC bit. If it is written into the corresponding Mailbox, including the CAN-ID, i.e., there is a possibility that the CAN-ID is overwritten by a different CAN-ID of the received message due to the LAFM used. This also implies that, if the identifier of the received message matches the ID + LAFM of 2 or more Mailboxes, the higher numbered Mailbox will always store the relevant messages and the lower numbered Mailbox will never receive messages. Therefore, the settings of the identifiers and LAFMs need to be carefully selected.

22.6.5 Reconfiguration of Mailbox

When re-configuration of Mailboxes is required, the following procedures should be taken.

(1) Change ID of transmit box or Change transmit box to receive box

Confirm that the corresponding CANTXPR is not set. The identifier and corresponding MBC field can be changed at any time. When both of them need to be changed, please change the identifier first and then the corresponding MBC field.

(2) Change ID of receive box or Change receive box to transmit box

- Method-1: Using Halt Mode
  The advantage of this method is that HCAN2 will not lose a message if the message is currently on the CAN bus and the HCAN2 is a receiver. HCAN2 will be in Halt Mode after completing of the reception. The disadvantage is that it might take long if HCAN2 is receiving a message (as the transmission to the halt state is delayed until the end of the reception), and also HCAN2 will not be able to receive/transmit messages during Halt mode.
- Method-2: Without Using Halt Mode
  The advantage of this method is that the re-configuration is done instantly, and the software overhead can be substantially reduced because of no interruption. CANRXPR (CANRFPR) needs to be read before and after the re-configuration is to check if a message is received or not during this period. Please note that CANMBIMR does not prevent the CANRXPR

ID or for the new ID. Therefore, if a message is received a message during this period, it is better to abandon this message, and this is the disadvantage of this method.

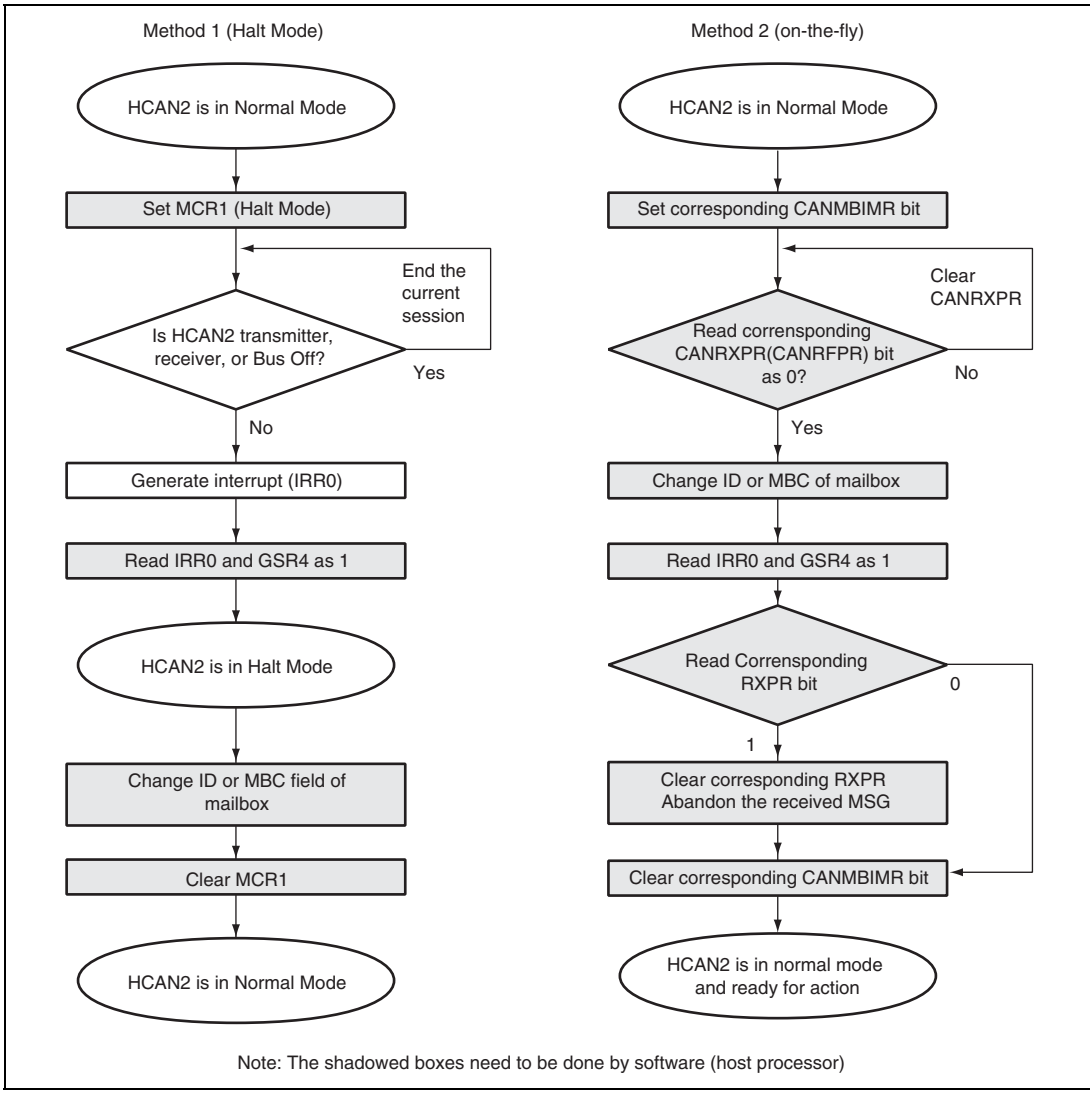


Figure 22.9 Changing ID of Receive Box or Changing Receive Box to Transmit Box

This HCAN2 supports clock gating to reduce power consumption. The module standby mode can be controlled by CLKSTP00 (bit 12 for channel 0 and bit 13 for channel 1) in the Low Power Consumption Module.

To set one of HCAN2 channels to standby mode, the following procedure is required:

1. Set HCAN2 to Halt Mode (MCR1 = 1).
2. Wait for the Halt Mode Interrupt (IRR0).
3. Clear all pending interrupt requests.
4. Disable the requested channel by setting the corresponding bit to 1 in CLKSTP00 register in the Power Control (channel 0 for CSTP12, channel 1 for CSTP13).

To wake-up from the Standby mode, the following procedure is required:

1. Enable the requested channel by setting the corresponding bit to 1 in CLKSTPCLR00 register in the Power Control (channel 0 for bit 12, channel 1 for bit 13).
2. Modify HCAN2 configurations, if necessary
3. Release HCAN2 Halt Mode by clearing MCR1.
4. After 11 recessive bits are detected on the CAN bus, the HCAN2 is able to join the communication.

22.7 Usage Notes

22.7.1 Auto-Acknowledge Mode Usage Note

Although transmission is possible in the Self Tests with the master control register (MCR) TST4 bit set to 1 (auto- acknowledge mode), reception of the transmitted data is not possible.

22.7.2 Mailbox Access during HCAN2 Sleep Mode

Do not access mailboxes when the HCAN2 module is in the sleep state. The CPU may stop if a mailbox is accessed during HCAN2 sleep mode. The CPU will not stop if registers other than the mailbox registers are accessed during HCAN2 sleep mode. Furthermore, the CPU will not stop if a mailbox is accessed during any state other than HCAN2 sleep mode.

This LSI incorporates one channel of the Serial Protocol Interface (HSPI).

23.1 Features

The HSPI has the following features.

- Operating mode: Master mode or Slave mode.
- The transmit and receive sections within the module are double buffered to allow duplex communication.
- A flexible peripheral clock division strategy allows a wide range of bit rates to be supported.
- The programmable clock control logic allows setting for two different transmit protocols and accommodates transmit and receive functions on either edge of the serial bit clock.
- Error detection logic is provided for warning of the receive buffer overflow.
- The HSPI has a facility to generate the chip select signal to slave modules when configured as a master mode either automatically as part of the data transfer process, or under the manual control of the host processor.
- The HSPI supports DMA transfer of both receive and transmit data independently via two DMA channels if implemented in the system.

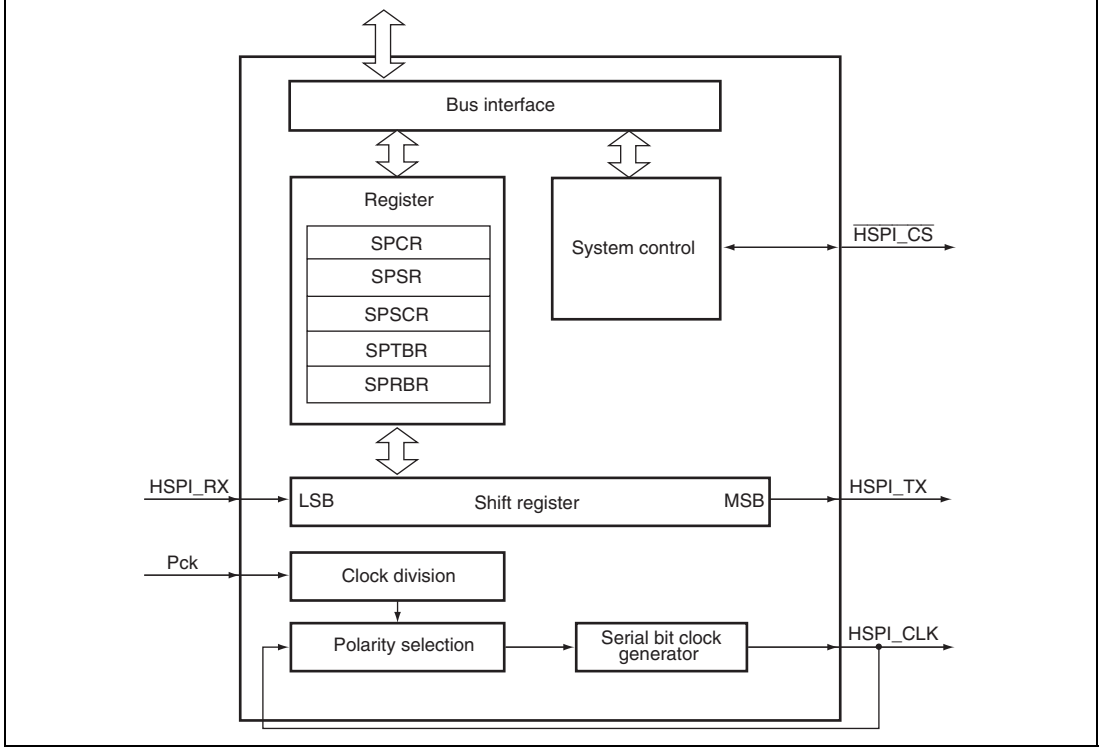


Figure 23.1 Block Diagram of HSPI

23.2 Input/Output Pins

The input/output pins of the HSPI is shown in table 23.1.

Table 23.1 Pin Configuration

Name	Abbreviation	I/O	Function
Serial bit clock pin	HSPI_CLK	Input/Output	Clock input/output
Transmit data pin	HSPI_TX	Output	Transmit data output
Receive data pin	HSPI_RX	Input	Receive data input
Chip select pin	$\overline{\text{HSPI\_CS}}$	Input/Output	Chip select

The HSPI has the following registers. For details on addresses of these registers and register status in each processing state, refer to section 32, List of Registers.

Table 23.2 Register Configuration (1)

Register Name	Abbrev.	R/W	P4 Address	Area 7 Address	Size	Sync Clock
Control register	SPCR	R/W	H'FE18 0000	H'1E18 0000	32	Pck
Status register	SPSR	R*2	H'FE18 0004	H'1E18 0004	32	Pck
System control register	SPSCR	R/W	H'FE18 0008	H'1E18 0008	32	Pck
Transmit buffer register	SPTBR	R/W	H'FE18 000C	H'1E18 000C	32	Pck
Receive buffer register	SPRBR	R	H'FE18 0010	H'1E18 0010	32	Pck

Table 23.2 Register Configuration (2)

Register Name	Abbrev.	Power-on Reset by $\overline{\text{RESET}}$ Pin/WDT/ H-UDI	Manual Reset by $\overline{\text{RESET}}$ Pin/WDT/ Multiple Exception	Sleep by Sleep Instruction/ Deep Sleep	Standby by Hardware	Standby by Software/ Each Module
Control register	SPCR	H'0000 0000*1	H'0000 0000*1	Retained	*3	Retained
Status register	SPSR	H'0000 0120*1	H'0000 0120*1	Retained		Retained
System control register	SPSCR	H'0000 0040*1	H'0000 0040*1	Retained		Retained
Transmit buffer register	SPTBR	H'0000 0000*1	H'0000 0000*1	Retained		Retained
Receive buffer register	SPRBR	H'0000 0000*1	H'0000 0000*1	Retained		Retained

Notes: 1. Reserved bits are read as undefined values.
2. To clear the flag, only 0s are written to bits 4 and 3.
3. After exiting hardware standby mode, this LSI enters the power-on reset state caused by the $\overline{\text{RESET}}$ pin.

SPCR is a 32-bit readable/writable register that controls the transfer data of shift timing and specifies the clock polarity and frequency.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
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
	-	-	-	-	-	-	-	-	FBS	CLKP	IDIV	CLKC4	CLKC3	CLKC2	CLKC1	CLKC0
Initial value:	-	-	-	-	-	-	-	-	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

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	All —	R	Reserved These bits are always read as an undefined value. The write value should always be 0.
7	FBS	0	R/W	First Bit Start Controls the timing relationship between each bit of transferred data and the serial bit clock. 0: The first bit transmitted from the HSPI module is set up such that it can be sampled by the receiving device on the first edge of HSPI_CLK after the $\overline{\text{HSPI\_CS}}$ pin goes low. Similarly the first received bit is sampled on the first edge of HSPI_CLK after the $\overline{\text{HSPI\_CS}}$ pin goes low. 1: The first bit transmitted from the HSPI module is set up such that it can be sampled by the receiving device on the second edge of HSPI_CLK after the $\overline{\text{HSPI\_CS}}$ pin goes low. Similarly the first received bit is sampled on the second edge of HSPI_CLK after the $\overline{\text{HSPI\_CS}}$ pin goes low.
6	CLKP	0	R/W	Serial Clock Polarity 0: HSPI_CLK signal is not inverted and so is low when inactive. 1: HSPI_CLK signal is inverted and so is high when inactive.

5	IDIV	0	R/W	Initial Clock Division Ratio 0: The peripheral clock is divided by a factor of 4 initially to create an intermediate frequency, which is further divided to create the serial bit clock when master mode. 1: The peripheral clock is divided by a factor of 32 initially to create an intermediate frequency, which is further divided to create the serial bit clock when master mode.
4 to 0	CLKC 4 to 0	All 0	R/W	Clock Division Count These bits determine the number of intermediate frequency cycles long both the high and low periods of the serial bit clock. 00000: 1 intermediate frequency cycle. Serial bit clock frequency = Intermediate frequency / 2. 00001: 2 Intermediate frequency cycles. Serial bit clock frequency = Intermediate frequency / 4. 00010: 3 intermediate frequency cycles. Serial bit clock frequency = Intermediate frequency / 6. : 11111: 32 intermediate frequency cycles. Serial bit clock frequency = Intermediate frequency / 64.

The serial bit clock frequency can be computed using the following formula:

$$\text{Serial bit clock frequency} = \frac{\text{Pck}}{(\text{Initial clock division} \quad (\text{Clock division count} + 1) \quad 2}$$

When the HSPI is configured as a slave, the IDIV and CLKC bits are ignored and the HSPI synchronizes to the externally supplied serial bit clock. The maximum value of the external serial bit clock that the module can operate with is peripheral clock frequency / 8.

If any of the FBS, CLKP, IDIV or CLKC bit values are changed, then the HSPI will undergo the HSPI software reset.

SPSR is a 32-bit readable/writable register. The status flag in SPSR can confirm whether the system correctly operates or not. If the ROIE bit in SPSCR is set to 1, an interrupt request is generated due to the occurrence of the receive buffer overrun error or the warning of the receive buffer overrun error. When the TFIE bit in SPSCR is set to 1, an interrupt request is generated by the transmit complete status flag. If the appropriate enable bit in SPSCR is set to 1, an interrupt request is generated due to the receive FIFO halfway, receive FIFO full, transmit FIFO empty, or transmit FIFO halfway flag. If the RNIE bit in SPSCR is set to 1, an interrupt request is generated when the receive FIFO is not empty.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
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
	-	-	-	-	-	TXFU	TXHA	TXEM	RXFU	RXHA	RXEM	RXOO	RXOW	RXFL	TXFN	TXFL
Initial value:	-	-	-	-	-	0	0	1	0	0	1	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R/W*	R/W*	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 11	—	All —	R	Reserved These bits are always read as an undefined value. The write value should always be 0.
10	TXFU	0	R	Transmit FIFO Full Flag This status flag is enabled only to operation in FIFO mode. The flag is set to 1 when the transmit FIFO is full of bytes for transmission and cannot accept any more. It is cleared to 0 when data is transmitted from the transmit FIFO.
9	TXHA	0	R	Transmit FIFO Halfway Flag This status flag is enabled only to operation in FIFO mode. The flag is set to 1 when the transmit FIFO reaches the halfway point, that is, it has 4 bytes of data and 4 spaces for more data. It is cleared to 0 when more data is written to the transmit FIFO. It remains set to 1 until cleared to 0 regardless of the subsequent FIFO levels. If TXHA = 1 and THIE = 1 then the interrupt is generated.

8	TXEM	1	R	Transmit FIFO Empty Flag This status flag is enabled only to operation in FIFO mode. The flag is set to 1 when the transmit FIFO is empty of data to transmit. It is cleared to 0 when more data is written to the transmit FIFO. If TXEM = 1 and TEIE = 1 then the interrupt is generated.
7	RXFU	0	R	Receive FIFO Full Flag This status flag is enabled only to operation in FIFO mode. The flag is set to 1 when the receive FIFO is full of received bytes and cannot accept any more. It is cleared to 0 when data is read out of the receive FIFO. If RXFU = 1 and RFIE = 1 then the interrupt is generated.
6	RXHA	0	R	Receive FIFO Halfway Flag This status flag is enabled only to operation in FIFO mode. The flag is set to 1 when the receive FIFO reaches the halfway point, that is, it has 4 bytes of data and 4 spaces for more data. It is cleared to 0 when more data is read from the receive FIFO. It remain set to 1 until cleared to 0 regardless of the subsequent FIFO levels. If RXHA = 1 and RHIE = 1 then the interrupt is generated.
5	RXEM	1	R	Receive FIFO Empty Flag This status flag is enabled only to operation in FIFO mode. The flag is set to 1 when the receive FIFO is empty of received data. It is cleared to 0 when more data is received into to the receive FIFO. If RXEM = 0 and RNIE = 1 then the interrupt is generated.
4	RXOO	0	R/W*	Receive Buffer Overrun Occurred Flag This status flag is set to 1 when new data has been received but the previous received data has not been read from SPRBR. The previously received data will not be overwritten by the newly received data. The RXOO flag remain set to 1 until writing a 0 to its bit position. If RXOO = 1 and ROIE = 1 then the interrupt is generated.

3	RXOW	0	R/W*	Receive Buffer Overrun Warning Flag This status flag is set to 1 when a new serial data transfer starts and the previous received data has not been read from SPRBR. The RXOW remain set to 1 until writing a 0 to its bit position. If RXOW= 1 and ROIE = 1 then the interrupt is generated.
2	RXFL	0	R	Receive Buffer Full Status Flag This status flag indicates that new data is available in the SPRBR and has not yet been read. It is set to 1 at the completion of a serial bus transfer at the point the shift register contents are loaded into the SPRBR. This bit is cleared to 0 by reading SPRBR. If RXFL = 1 and RXDE = 1 then the DMA transfer request enabled.
1	TXFN	0	R	Transmit Complete Status Flag This status flag indicates that the last transmission has completed. It is set to 1 when SPTBR is able to write more data from the peripheral bus. This bit is cleared to 0 by writing more data SPTBR. If TXFN = 1 and TFIE = 1 then the interrupt is generated.
0	TXFL	0	R	Transmit Buffer Full Status Flag This status flag indicates SPTBR has transmitted data. It is set to 1 when SPTBR is written with data from the peripheral bus. This bit is cleared to 0 when SPTBR is able to accept more data from the peripheral bus. If TXFL = 0 (i.e. the SPTBR is empty) and TXDE = 1 then the DMA transfer request enabled.

Note:* These bits are readable/writable bits. When writing 0, these bits are initialized, while writing 1 is ignored.

SPSCR is a 32-bit readable/writable register that enables or disables interrupts or FIFO mode, selects either LSB first or MSB first in transmitting/receiving date, and master or slave mode.

If any of the FFEN, LMSB, CSA or MASL bit values are changed, then the module will undergo the HSPI software reset.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
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
	-	-	TEIE	THIE	RNIE	RHIE	RFIE	FFEN	LMSB	CSV	CSA	TFIE	ROIE	RXDE	TXDE	MASL
Initial value:	-	-	0	0	0	0	0	0	0	1	0	0	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/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 14	—	All —	R	Reserved These bits are always read as an undefined value. The write value should always be 0.
13	TEIE	0	R/W	Transmit FIFO Empty Interrupt Enable 0:Transmit FIFO empty interrupt disabled 1:Transmit FIFO empty interrupt enabled
12	THIE	0	R/W	Transmit FIFO Halfway Interrupt Enable 0:Transmit FIFO halfway interrupt disabled 1:Transmit FIFO halfway interrupt enabled
11	RNIE	0	R/W	Receive FIFO Not Empty Interrupt Enable 0: Receive FIFO not empty interrupt disabled 1: Receive FIFO not empty interrupt enabled
10	RHIE	0	R/W	Receive FIFO Halfway Interrupt Enable 0: Receive FIFO halfway interrupt disabled 1: Receive FIFO halfway interrupt enabled
9	RFIE	0	R/W	Receive FIFO Full Interrupt Enable 0: Receive FIFO full interrupt disabled 1: Receive FIFO full interrupt enabled

8	FFEN	0	R/W	FIFO Mode Enable Enables or disables the FIFO mode. When FIFO mode is enabled two 8-entry deep FIFOs are made available, one for transmit data and one for receive data. These FIFOs are read and written via SPTBR and SPRBR. When FIFO mode is disabled the SPTBR and SPRBR are used directly so new data must be written to SPTBR and read from SPRBR for each and every transfer. FIFO mode must be disabled if DMA requests are also going to be used to service SPTBR and SPRBR. 0: FIFO mode disabled 1: FIFO mode enabled
7	LMSB	0	R/W	LSB/MSB First Control 0: Data is transmitted and received most significant bit (MSB) first. 1: Data is transmitted and received least significant bit (LSB) first.
6	CSV	1	R/W	Chip Select Value Controls the value output from the chip select when the HSPI is a master and the chip select generation has been selected. 0: Chip select output is low. 1: Chip select output is high.
5	CSA	0	R/W	Automatic/Manual Chip Select 0: Chip select output is automatically generated during data transfer. 1: Chip select output is manually controlled, with its value being determined by the CSV bit.
4	TFIE	0	R/W	Transmit Complete Interrupt Enable 0: Transmit complete interrupt disabled 1: Transmit complete interrupt enabled
3	ROIE	0	R/W	Receive Overrun Occurred / Warning Interrupt Enable 0: Receive overrun occurred / warning interrupt disabled 1: Receive overrun occurred / warning interrupt enabled

2	RXDE	0	R/W	Receive DMA Enable 0: Receive DMA transfer request disabled 1: Receive DMA transfer request enabled
1	TXDE	0	R/W	Transmit DMA Enable 0: Transmit DMA transfer request disabled 1: Transmit DMA transfer request enabled
0	MASL	0	R/W	Master/Slave Select Bit 0: HSPI module configured as a slave 1: HSPI module configured as a master

23.3.4 Transmit Buffer Register (SPTBR)

SPTBR is a 32-bit readable/writable register that stores data to be transmitted.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
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
	-	-	-	-	-	-	-	-	TD							
Initial value:	-	-	-	-	-	-	-	-	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

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	All —	R	Reserved These bits are always read as an undefined value. The write value should always be 0.
7 to 0	TD	All 0	R/W	Transmit Data Data written to this register is transferred to the shift register for transmission. When reading these bits, always read as data stored in the transmit buffer.

SPRBR is a 32-bit read-only register that stores the number of received data.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
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
	-	-	-	-	-	-	-	-	RD							
Initial value:	-	-	-	-	-	-	-	-	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	Bit Name	Initial Value	R/W	Description
31 to 8	—	All —	R	Reserved These bits are always read as an undefined value. The write value should always be 0.
7 to 0	RD	All 0	R	Receive Data Data from the shift register, which is stored as each byte, is received, if the previously received data has been read.

23.4.1 Operation Overview without DMA (FIFO Mode Disabled)

Figure 23.2 shows the flow of a transmit/receive operation procedure.

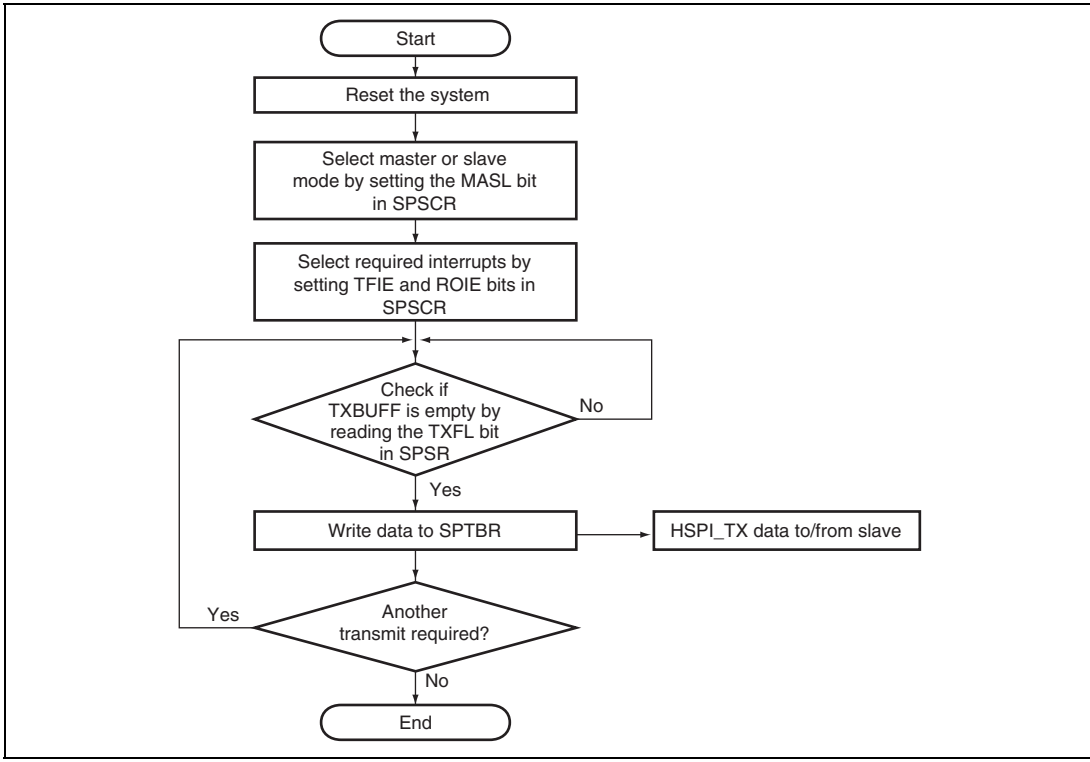


Figure 23.2 Operational Flowchart

Depending on the settings of SPCR, the master transmits data to the slave on either the falling or rising edge of HSPI_CLK and samples data from the slave on the opposite edge. The data transfer between the master and slave completes when the transmit complete status flag (TXFN) in SPSR is set to 1. This flag should be used to identify when an HSPI transfer event (byte transmitted and byte received) has occurred, even in the case where the HSPI module is being used to receive data only (null data being transmitted). By default data is transmitted MSB first, but LSB first is also possible depending on how the LMSB bit in SPSCR is set.

During the transmit function the slave responds by sending data to the master synchronized with the HSPI_CLK from the master transmitted. Data from the slave is sampled and transferred to the shift register in the module and on completion of the transmit function, is transferred to SPRBR.

pin must be driven high between successive bytes. When the FBS = 1, the $\overline{\text{HSPI\_CS}}$ pin can stay low for several byte transmissions. In this case, if the system is configured such that the FBS is always 1, then the $\overline{\text{HSPI\_CS}}$ line can be fixed at ground (if the HSPI will only be used as a slave).

23.4.2 Operation Overview with DMA

The operation of the HHSPI when DMA is used to perform transmit and receive data transfers is simpler than when DMA is not used. The HSPI must be configured as in the case for transfers without DMA. FIFO mode must be disabled. The DMA controller (DMAC) should then be configured to transfer the required amount of data. DMA requests can then be enabled in the HSPI module and the transfers will then take place without further processor intervention.

When the DMAC indicates that all transfers have ended then the DMA request signals in the HSPI module should be disabled to remove any remaining DMA requests. This is necessary as the HSPI module will always request data to transmit.

23.4.3 Operation with FIFO Mode Enabled

In order to reduce the interrupt overhead on the processor in the case for operation without DMA mode, FIFO mode has been provided. When FIFO mode is enabled, up to 8 bytes can be written in advance for transmission and up to 8 bytes can be received before the receive FIFO needs to be read. To transfer the specified amount of data between the HSPI module and an external device, follow the following procedure:

1. Set up the module for the required HSPI transfer characteristics (master/slave, clock polarity etc.) and enable FIFO mode.
2. Write bytes into the transmit FIFO via SPTBR. If more than 8 bytes are to be transmitted then enable the transmit FIFO halfway interrupt to keep track of the FIFO level as data is transmitted.
3. Respond to the transmit FIFO halfway interrupt when it occurs by writing more data to the transmit FIFO and reading data from the receive FIFO via SPRBR.
4. When all of the transmit data has been written into the transmit FIFO, disable the transmit FIFO halfway interrupt and read the contents of the receive FIFO until it is empty. Enable the receive FIFO not empty interrupt to keep track of when the final bytes of the transfer are received.
5. Respond to the receive FIFO not empty interrupt until all the expected data has been received.
6. Disable the module until it is required again.

this is the case, follow the following procedure.

1. Set up the module for the required HSPI transfer characteristics (master/slave, clock polarity etc.) and enable FIFO mode.
2. Fill the transmit FIFO with the data to transmit. Enable the receive FIFO not empty interrupt.
3. Respond to the receive FIFO not empty interrupt and read data from the receive FIFO until it is empty. Write more data to the transmit FIFO if required.
4. Disable the module when the transfer is to stop.

23.4.4 Timing Diagrams

The following diagrams explain the timing relationship of all shift and sample processes in the HSPI. Figure 23.3 shows the conditions when FBS = 0, while figure 23.4 shows the conditions when FBS = 1. It can be seen that if CLKP in SPCR is 0 then transmit data is shifted on the falling edge of HSPI_CLK and receive data is sampled on the rising edge. The opposite is true when CLKP = 1.

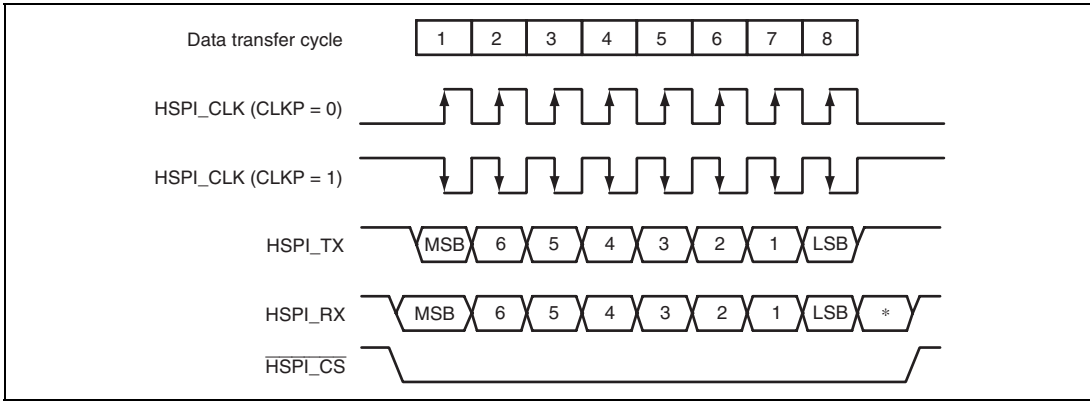


Figure 23.3 Timing Conditions when FBS = 0

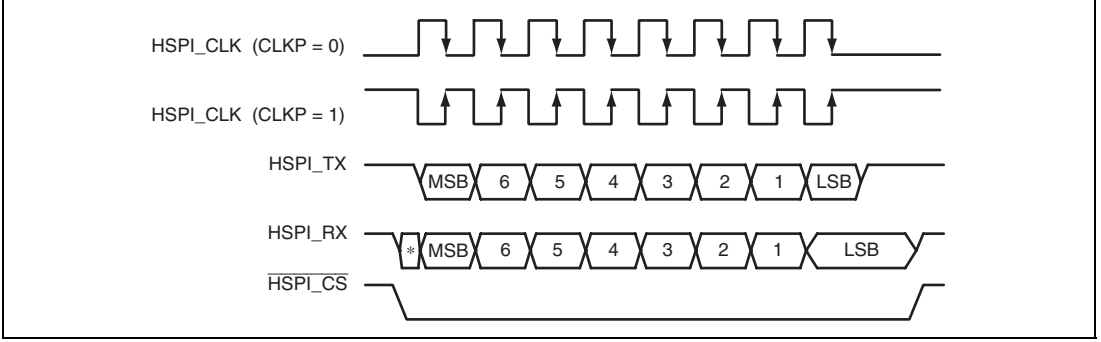


Figure 23.4 Timing Conditions when FBS = 1

23.4.5 HSPI Software Reset

If any of the FBS, CLKP, IDIV or CLKC bit values are changed, then the HSPI software reset is generated. The receive and transmit FIFO pointers can be initialized by the HSPI software reset. The data transmission after the HSPI software reset should protect transmitting and receiving protocol of HSPI, and please perform it from the first. A guarantee of operation is not offered other than it.

23.4.6 Clock Polarity and Transmit Control

SPCR also allows the user to define the shift timing for transmit data and polarity. The FBS bit in SPCR allows selection between two different transfer formats. The MSB or LSB is valid on the falling edge of $\overline{\text{HSPI\_CS}}$. The CLKP bit in SPCR allows for control of the polarity select block which controls which edges of HSPI_CLK shift and sample data in the master and slave.

23.4.7 Transmit and Receive Routines

The master and slave can be considered linked together as a circular shift register synchronized with HSPI_CLK. The transmit byte from the master is replaced with the receive byte from the slave in eight HSPI_CLK cycles. Both the transmit and receive functions are double buffered to allow for continuous reads and writes. When FIFO mode is enabled eight entry FIFOs are available for both transmit and receive data.

The HSPI operates synchronized with the bus clock.

Module standby mode can be enabled or disabled by controlling the CSTP22 bit in the clock stop register (CLKSTP00) in the CPG module.

To power down the module, follow the following procedure.

1. Ensure all data transfers have taken place. That is, the transmit buffer (or FIFOs) should be empty and the receive buffer (or FIFOs) should have been read until they are empty.
2. Disable all DMA requests and interrupt requests. Disable FIFO mode.
3. Set the CSTP22 bit in clock stop clear register (CLKSTP00) to 1.

To wake up the HSPI, the CSTP22 bit in the clock stop register (CLKSTPCLR00) should be set to 1.

24.1 Features

The PFC has the following features.

- Individual control of pull-up of each port pin used by a peripheral module
- Individual control of high-impedance (Hi-Z) state of pins used by the SCIF in software standby mode
- Applicable modules are selectable by the MFI mode/LCD mode

This LSI has ten general ports (A to H, J, and K), which provide 69 input/output pins and one output pin in total.

The GPIO (general port for input/output) has the following features.

- Each port pin is multiplexed pin, for which the port control register can set the pin function and pull-up MOS control individually.
- Each port has a data register that stores data for the pins.
- GPIO interrupts are supported.

For details of multiplexed pins, refer to table 24.1, Multiplexed Pins Controlled by Port Control Registers. For pin multiplexing in this LSI, refer to table 1.3 and 1.4. By default, each pin of the ports is pulled up.

Table 24.1 Multiplexed Pins Controlled by Port Control Registers

Pin Name	Port	GPIO	MFI Mode (MD7=0)	LCD Mode (MD7 = 1)	Register Setting
CAN0_NERR/AUDCK*1	A	PTA7 input/output	HCAN2[0]/AUD		AUD
CAN0_RX/AUDATA[2]*1	A	PTA6 input/output	HCAN2[0]/AUD		AUD
CAN0_TX/AUDATA[0]*1	A	PTA5 input/output	HCAN2[0]/AUD		AUD
CAN1_NERR/AUDSYNC*1	A	PTA4 input/output	HCAN2[1]/AUD		AUD
CAN1_RX/AUDATA[3]*1	A	PTA3 input/output	HCAN2[1]/AUD		AUD
CAN1_TX/AUDATA[1]*1	A	PTA2 input/output	HCAN2[1]/AUD		AUD
SSI0_SCK/HAC_SD_IN0/$\overline{BS2}$*1*2	B	PTB7 input/output	SSI[0]/HAC[0]		$\overline{BS2}$
SSI0_WS/HAC_SYNC0*1	B	PTB6 input/output	SSI[0]/HAC[0]		
SSI0_SDATA/HAC_SD_OUT0*1	B	PTB5 input/output	SSI[0]/HAC[0]		

CMT_CTR0/TCLK	B	PTB4 input/output		CMT	
CMT_CTR1	B	PTB3 input/output		CMT	
CMT_CTR2	B	PTB2 input/output		CMT	
CMT_CTR3	B	PTB1 input/output		CMT	
MFI-D0/LCD_DATA0	C	PTC7 input/output	MFI	LCDC	
MFI-D1/LCD_DATA1	C	PTC6 input/output	MFI	LCDC	
MFI-D2/LCD_DATA2/$\overline{\text{IRQ6}}$*2	C	PTC5 input/output	MFI	LCDC	IRQ
MFI-D3/LCD_DATA3/$\overline{\text{IRQ7}}$*2	C	PTC4 input/output	MFI	LCDC	IRQ
MFI-D4/LCD_DATA4/$\overline{\text{DREQ2}}$*2	C	PTC3 input/output	MFI	LCDC	DMAC
MFI-D5/LCD_DATA5/DRAK2/ DACK2*2	C	PTC2 input/output	MFI	LCDC	DMAC
MFI-D6/LCD_DATA6/$\overline{\text{DREQ3}}$*2	C	PTC1 input/output	MFI	LCDC	DMAC
MFI-D7/LCD_DATA7/DRAK3/ DACK3*2	C	PTC0 input/output	MFI	LCDC	DMAC
MFI-D8/LCD_DATA8*1	D	PTD7 input/output	MFI	LCDC	
MFI-D9/LCD_DATA9*1	D	PTD6 input/output	MFI	LCDC	
MFI-D10/LCD_DATA10*1	D	PTD5 input/output	MFI	LCDC	
MFI-D11/LCD_DATA11*1	D	PTD4 input/output	MFI	LCDC	
MFI-D12/LCD_DATA12*1	D	PTD3 input/output	MFI	LCDC	
MFI-D13/LCD_DATA13*1	D	PTD2 input/output	MFI	LCDC	
MFI-D14/LCD_DATA14*1	D	PTD1 input/output	MFI	LCDC	
MFI-D15/LCD_DATA15*1	D	PTD0 input/output	MFI	LCDC	
$\overline{\text{MFI-INT}}$/LCD_CLK*3	E	PTE7 input/output	MFI	LCDC	
$\overline{\text{MFI-CS}}$/LCD_DON*3	E	PTE6 input/output	MFI	LCDC	
MFI-E/LCD_CL1*3	E	PTE5 input/output	MFI	LCDC	
MFI-MD/LCD_CL2*3	E	PTE4 input/output	MFI	LCDC	
MFI-RS/LCD_M_DISP*3	E	PTE3 input/output	MFI	LCDC	
MFI-RW/LCD_FLM*3	E	PTE2 input/output	MFI	LCDC	
VCPWC/$\overline{\text{IRQ4}}$	E	PTE1 input/output	IRQ	LCDC	
VEPWC/$\overline{\text{IRQ5}}$	E	PTE0 input/output	IRQ	LCDC	
HSPI_TX/SIM_D/MCDAT*1	F	PTF3 input/output		HSPI/MMCIF/SIM	
HSPI_RX*1	F	PTF2 input/output		HSPI/MMCIF/SIM	
HSPI_CLK/SIM_CLK/MCCLK*1	F	PTF1 input/output		HSPI/MMCIF/SIM	

$\overline{\text{HSPI\_CS}}$/SIM_RST/MCCMD*1	F	PTF0 input/output	HSPI/MMCIF/SIM	
SCIF0_CLK	G	PTG7 input/output	SCIF[0]	
SCIF0_RXD	G	PTG6 input/output	SCIF[0]	
SCIF0_TXD	G	PTG5 input/output	SCIF[0]	
SCIF1-CLK	G	PTG4 input/output	SCIF[1]	
$\overline{\text{SCIF1\_CTS}}$	G	PTG3 input/output	SCIF[1]	
$\overline{\text{SCIF1\_RTS}}$	G	PTG2 input/output	SCIF[1]	
SCIF1_RXD	G	PTG1 input/output	SCIF[1]	
SCIF1_TXD	G	PTG0 input/output	SCIF[1]	
SCIF2-CLK	H	PTH7 input/output	SCIF[2]	
$\overline{\text{SCIF2\_CTS}}$*3	H	PTH6 input/output	SCIF[2]	
$\overline{\text{SCIF2\_RTS}}$*3	H	PTH5 input/output	SCIF[2]	
SCIF2_RXD	H	PTH4 input/output	SCIF[2]	
SCIF2_TXD	H	PTH3 input/output	SCIF[2]	
UCLK	H	PTH2 input/output	USB	
USB_PENC*3	H	PTH1 input/output	USB	
$\overline{\text{USB\_OVC}}$	H	PTH0 input/output	USB	
HAC_BIT_CLK0	J	PTJ7 input/output	HAC[0]/SSI[0]	
$\overline{\text{HAC\_RES}}$	J	PTJ6 input/output	HAC[1][0]	
SSI1_WS/HAC_SYNC1*1	J	PTJ5 input/output	HAC[1]/SSI[1]	
SSI1_SCK/HAC_SD_IN1	J	PTJ4 input/output	HAC[1]/SSI[1]	
SSI1_SDATA/HAC_SD_OUT1*1	J	PTJ3 input/output	HAC[1]/SSI[1]	
HAC_BIT_CLK1	J	PTJ2 input/output	HAC[1]/SSI[1]	
DCK	J	PTJ1 output	DCK	
Reserved/AUDATA[3] *1*3	K	PTK7 input/output	Reserved/AUD	AUD
Reserved/AUDATA[2] *1	K	PTK6 input/output	Reserved/AUD	AUD
Reserved/AUDATA[1] *1	K	PTK5 input/output	Reserved/AUD	AUD
Reserved/AUDCK *1	K	PTK4 input/output	Reserved/AUD	AUD
Reserved/AUDSYNC *1	K	PTK3 input/output	Reserved/AUD	AUD
$\overline{\text{ADTRG}}$/AUDATA[0]*1	K	PTK2 input/output	ADC/AUD	AUD

Notes: 1. A module that uses this pin is selected by IPSELR in PFC.
2. A module that uses this pin is selected by MODSELR in PFC.
3. GPIO interrupts are supported.

The PFC has the following set of registers. For details of the addresses of these registers and the states in each operating mode, see section 32, List of Registers.

Table 24.2 Register Configuration (1)

Register Name	Abbrev.	R/W	P4 Address	Area 7 Address	Size	Sync Clock
Input pin pull-up control register	INPUPA	R/W	H'FE40 0028	H'1E40 0028	16	Pck
DMA pin control register	DMAPCR	R/W	H'FE40 002C	H'1E40 002C	16	Pck
SCIF Hi-Z control register	SCIHZR	R/W	H'FE40 0030	H'1E40 0030	16	Pck
Peripheral module select register	IPSELR	R/W	H'FE40 0034	H'1E40 0034	16	Pck
Port A pull-up control register	PAPUPR	R/W	H'FE40 0080	H'1E40 0080	8	Pck
Port B pull-up control register	PBPUPR	R/W	H'FE40 0084	H'1E40 0084	8	Pck
Port C pull-up control register	PCPUPR	R/W	H'FE40 0088	H'1E40 0088	8	Pck
Port D pull-up control register	PDPUPR	R/W	H'FE40 008C	H'1E40 008C	8	Pck
Port E pull-up control register	PEPUPR	R/W	H'FE40 0090	H'1E40 0090	8	Pck
Port F pull-up control register	PFPUPR	R/W	H'FE40 0094	H'1E40 0094	8	Pck
Port G pull-up control register	PGPUPR	R/W	H'FE40 0098	H'1E40 0098	8	Pck
Port H pull-up control register	PHPUPR	R/W	H'FE40 009C	H'1E40 009C	8	Pck
Port J pull-up control register	PJPUPR	R/W	H'FE40 00A0	H'1E40 00A0	8	Pck
Port K pull-up control register	PKPUPR	R/W	H'FE40 00A4	H'1E40 00A4	8	Pck
Mode pin pull-up control register	MDPUPR	R/W	H'FE40 00A8	H'1E40 00A8	8	Pck
Mode select register	MODSELR	R/W	H'FE40 00AC	H'1E40 00AC	8	Pck
Port A control register	PACR	R/W	H'FE40 0000	H'1E40 0000	16	Pck
Port B control register	PBCR	R/W	H'FE40 0004	H'1E40 0004	16	Pck
Port C control register	PCCR	R/W	H'FE40 0008	H'1E40 0008	16	Pck
Port D control register	PDCR	R/W	H'FE40 000C	H'1E40 000C	16	Pck
Port E control register	PECR	R/W	H'FE40 0010	H'1E40 0010	16	Pck
Port F control register	PFCR	R/W	H'FE40 0014	H'1E40 0014	16	Pck
Port G control register	PGCR	R/W	H'FE40 0018	H'1E40 0018	16	Pck
Port H control register	PHCR	R/W	H'FE40 001C	H'1E40 001C	16	Pck
Port J control register	PJCR	R/W	H'FE40 0020	H'1E40 0020	16	Pck
Port K control register	PKCR	R/W	H'FE40 0024	H'1E40 0024	16	Pck
Port A data register	PADR	R/W	H'FE40 0040	H'1E40 0040	8	Pck

Port B data register	PBDR	R/W	H'FE40 0044	H'1E40 0044	8	Pck
Port C data register	PCDR	R/W	H'FE40 0048	H'1E40 0048	8	Pck
Port D data register	PDDR	R/W	H'FE40 004C	H'1E40 004C	8	Pck
Port E data register	PEDR	R/W	H'FE40 0050	H'1E40 0050	8	Pck
Port F data register	PFDR	R/W	H'FE40 0054	H'1E40 0054	8	Pck
Port G data register	PGDR	R/W	H'FE40 0058	H'1E40 0058	8	Pck
Port H data register	PHDR	R/W	H'FE40 005C	H'1E40 005C	8	Pck
Port J data register	PJDR	R/W	H'FE40 0060	H'1E40 0060	8	Pck
Port K data register	PKDR	R/W	H'FE40 0064	H'1E40 0064	8	Pck
GPIO interrupt control register	GPIOIC	R/W	H'FE80 0048	H'1E80 0048	16	Bck

Table 24.2 Register Configuration (2)

Register Name	Abbrev.	Power-on Reset by $\overline{\text{RESET}}$ Pin/WDT/ H-UDI	Manual Reset by $\overline{\text{RESET}}$ Pin/WDT/ Multiple Exception	Sleep by Sleep Instruction/ Deep Sleep	Standby by Hardware	Standby by Software/ Each Module
Input pin pull-up control register	INPUPA	H'FF00	Retained	Retained	*	Retained
DMA pin control register	DMAPCR	H'A550	Retained	Retained		Retained
SCIF Hi-Z control register	SCIHZR	H'0000	Retained	Retained		Retained
Peripheral module select register	IPSELR	H'0003	Retained	Retained		Retained
Port A pull-up control register	PAPUPR	H'FC	Retained	Retained		Retained
Port B pull-up control register	PBPUPR	H'FE	Retained	Retained		Retained
Port C pull-up control register	PCPUPR	H'FF	Retained	Retained		Retained
Port D pull-up control register	PDPUPR	H'FF	Retained	Retained		Retained
Port E pull-up control register	PEPUPR	H'FF	Retained	Retained		Retained
Port F pull-up control register	PFPUPR	H'0F	Retained	Retained		Retained
Port G pull-up control register	PGPUPR	H'FF	Retained	Retained		Retained
Port H pull-up control register	PHPUPR	H'FF	Retained	Retained		Retained
Port J pull-up control register	PJPUPR	H'FC	Retained	Retained		Retained
Port K pull-up control register	PKPUPR	H'FC	Retained	Retained		Retained
Mode pin pull-up control register	MDPUPR	H'38	Retained	Retained		Retained
Mode select register	MODSELR	H'00	Retained	Retained		Retained

Register Name	Abbrev.	Reset by RESET Pin/WDT/ H-UDI	by RESET Pin/WDT/ Multiple Exception	Sleep by Sleep Instruction/ by Deep Sleep	Hardware	by Software/ Each Module
Port A control register	PACR	H'0000	Retained	Retained	*	Retained
Port B control register	PBCR	H'0000	Retained	Retained		Retained
Port C control register	PCCR	H'FFFF	Retained	Retained		Retained
Port D control register	PDCR	H'FFFF	Retained	Retained		Retained
Port E control register	PECR	H'0000	Retained	Retained		Retained
Port F control register	PFCR	H'0000	Retained	Retained		Retained
Port G control register	PGCR	H'0000	Retained	Retained		Retained
Port H control register	PHCR	H'003C	Retained	Retained		Retained
Port J control register	PJCR	H'0000	Retained	Retained		Retained
Port K control register	PKCR	H'0000	Retained	Retained		Retained
Port A data register	PADR	H'00	Retained	Retained		Retained
Port B data register	PBDR	H'00	Retained	Retained		Retained
Port C data register	PCDR	H'00	Retained	Retained		Retained
Port D data register	PDDR	H'00	Retained	Retained		Retained
Port E data register	PEDR	H'00	Retained	Retained		Retained
Port F data register	PFDR	H'00	Retained	Retained		Retained
Port G data register	PGDR	H'00	Retained	Retained		Retained
Port H data register	PHDR	H'00	Retained	Retained		Retained
Port J data register	PJDR	H'00	Retained	Retained		Retained
Port K data register	PKDR	H'00	Retained	Retained		Retained
GPIO interrupt control register	GPIOIC	H'0000	Retained	Retained		Retained

Note: * After exiting hardware standby mode, this LSI enters the power-on reset state caused by the RESET pin.

PACR is a 16-bit readable/writable register that selects the pin function and input pull-up MOS control.

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	-	-	-	-
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

Bit	Bit Name	Initial value	R/W	Description
15	PA7MD1	0	R/W	PTA7 Mode
14	PA7MD0	0	R/W	00: Peripheral module (HCAN2[0]/AUD) 01: Port output 10: Port input (pull-up MOS: Off) 11: Port input (pull-up MOS: On)
13	PA6MD1	0	R/W	PTA6 Mode
12	PA6MD0	0	R/W	00: Peripheral module (HCAN2[0]/AUD) 01: Port output 10: Port input (pull-up MOS: Off) 11: Port input (pull-up MOS: On)
11	PA5MD1	0	R/W	PTA5 Mode
10	PA5MD0	0	R/W	00: Peripheral module (HCAN2[0]/AUD) 01: Port output 10: Port input (pull-up MOS: Off) 11: Port input (pull-up MOS: On)
9	PA4MD1	0	R/W	PTA4 Mode
8	PA4MD0	0	R/W	00: Peripheral module (HCAN2[1]/AUD) 01: Port output 10: Port input (pull-up MOS: Off) 11: Port input (pull-up MOS: On)
7	PA3MD1	0	R/W	PTA3 Mode
6	PA3MD0	0	R/W	00: Peripheral module (HCAN2[1]/AUD) 01: Port output 10: Port input (pull-up MOS: Off) 11: Port input (pull-up MOS: On)
5	PA2MD1	0	R/W	PTA2 Mode
4	PA2MD0	0	R/W	00: Peripheral module (HCAN2[1]/AUD) 01: Port output 10: Port input (pull-up MOS: Off) 11: Port input (pull-up MOS: On)

3 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

24.2.2 Port B Control Register (PBCR)

PBCR is a 16-bit readable/writable register that selects the pin function and input pull-up MOS control.

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	-	-
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	R

Bit	Bit Name	Initial value	R/W	Description
15 14	PB7MD1 PB7MD0	0 0	R/W R/W	PTB7 Mode 00: Peripheral module (SSI[0]/HAC[0]/$\overline{BS2}$) 01: Port output 10: Port input (pull-up MOS: Off) 11: Port input (pull-up MOS: On)
13 12	PB6MD1 PB6MD0	0 0	R/W R/W	PTB6 Mode 00: Peripheral module (SSI[0]/HAC[0]) 01: Port output 10: Port input (pull-up MOS: Off) 11: Port input (pull-up MOS: On)
11 10	PB5MD1 PB5MD0	0 0	R/W R/W	PTB5 Mode 00: Peripheral module (SSI[0]/HAC[0]) 01: Port output 10: Port input (pull-up MOS: Off) 11: Port input (pull-up MOS: On)
9 8	PB4MD1 PB4MD0	0 0	R/W R/W	PTB4 Mode 00: Peripheral module (CMT) 01: Port output 10: Port input (pull-up MOS: Off) 11: Port input (pull-up MOS: On)
7 6	PB3MD1 PB3MD0	0 0	R/W R/W	PTB3 Mode 00: Peripheral module (CMT) 01: Port output 10: Port input (pull-up MOS: Off) 11: Port input (pull-up MOS: On)

5	PB2MD1	0	R/W	PTB2 Mode
4	PB2MD0	0	R/W	00: Peripheral module (CMT) 01: Port output 10: Port input (pull-up MOS: Off) 11: Port input (pull-up MOS: On)
3	PB1MD1	0	R/W	PTB1 Mode
2	PB1MD0	0	R/W	00: Peripheral module (CMT) 01: Port output 10: Port input (pull-up MOS: Off) 11: Port input (pull-up MOS: On)
1, 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

24.2.3 Port C Control Register (PCCR)

PCCR is a 16-bit readable/writable register that selects the pin function and input pull-up MOS control.

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	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

Bit	Bit Name	Initial value	R/W	Description
15	PC7MD1	1	R/W	PTC7 Mode
14	PC7MD0	1	R/W	00: Peripheral module (MFI/LCDC) 01: Port output 10: Port input (pull-up MOS: Off) 11: Port input (pull-up MOS: On)
13	PC6MD1	1	R/W	PTC6 Mode
12	PC6MD0	1	R/W	00: Peripheral module (MFI/LCDC) 01: Port output 10: Port input (pull-up MOS: Off) 11: Port input (pull-up MOS: On)
11	PC5MD1	1	R/W	PTC5 Mode
10	PC5MD0	1	R/W	00: Peripheral module (MFI/LCDC/IRQ) 01: Port output 10: Port input (pull-up MOS: Off) 11: Port input (pull-up MOS: On)

9	PC4MD1	1	R/W	PTC4 Mode
8	PC4MD0	1	R/W	00: Peripheral module (MFI/LCDC/IRQ) 01: Port output 10: Port input (pull-up MOS: Off) 11: Port input (pull-up MOS: On)
7	PC3MD1	1	R/W	PTC3 Mode
6	PC3MD0	1	R/W	00: Peripheral module (MFI/LCDC/DMAC) 01: Port output 10: Port input (pull-up MOS: Off) 11: Port input (pull-up MOS: On)
5	PC2MD1	1	R/W	PTC2 Mode
4	PC2MD0	1	R/W	00: Peripheral module (MFI/LCDC/DMAC) 01: Port output 10: Port input (pull-up MOS: Off) 11: Port input (pull-up MOS: On)
3	PC1MD1	1	R/W	PTC1 Mode
2	PC1MD0	1	R/W	00: Peripheral module (MFI/LCDC/DMAC) 01: Port output 10: Port input (pull-up MOS: Off) 11: Port input (pull-up MOS: On)
1	PC0MD1	1	R/W	PTC0 Mode
0	PC0MD0	1	R/W	00: Peripheral module (MFI/LCDC/DMAC) 01: Port output 10: Port input (pull-up MOS: Off) 11: Port input (pull-up MOS: On)

24.2.4 Port D Control Register (PDCR)

PDCR is a 16-bit readable/writable register that selects the pin function and input pull-up MOS control.

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	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

15	PD7MD1	1	R/W	PTD7 Mode
14	PD7MD0	1	R/W	00: Peripheral module (MFI/LCDC) 01: Port output 10: Port input (pull-up MOS: Off) 11: Port input (pull-up MOS: On)
13	PD6MD1	1	R/W	PTD6 Mode
12	PD6MD0	1	R/W	00: Peripheral module (MFI/LCDC) 01: Port output 10: Port input (pull-up MOS: Off) 11: Port input (pull-up MOS: On)
11	PD5MD1	1	R/W	PTD5 Mode
10	PD5MD0	1	R/W	00: Peripheral module (MFI/LCDC) 01: Port output 10: Port input (pull-up MOS: Off) 11: Port input (pull-up MOS: On)
9	PD4MD1	1	R/W	PTD4 Mode
8	PD4MD0	1	R/W	00: Peripheral module (MFI/LCDC) 01: Port output 10: Port input (pull-up MOS: Off) 11: Port input (pull-up MOS: On)
7	PD3MD1	1	R/W	PTD3 Mode
6	PD3MD0	1	R/W	00: Peripheral module (MFI/LCDC) 01: Port output 10: Port input (pull-up MOS: Off) 11: Port input (pull-up MOS: On)
5	PD2MD1	1	R/W	PTD2 Mode
4	PD2MD0	1	R/W	00: Peripheral module (MFI/LCDC) 01: Port output 10: Port input (pull-up MOS: Off) 11: Port input (pull-up MOS: On)
3	PD1MD1	1	R/W	PTD1 Mode
2	PD1MD0	1	R/W	00: Peripheral module (MFI/LCDC) 01: Port output 10: Port input (pull-up MOS: Off) 11: Port input (pull-up MOS: On)
1	PD0MD1	1	R/W	PTD0 Mode
0	PD0MD0	1	R/W	00: Peripheral module (MFI/LCDC) 01: Port output 10: Port input (pull-up MOS: Off) 11: Port input (pull-up MOS: On)

PECR is a 16-bit readable/writable register that selects the pin function and input pull-up MOS control.

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:	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	Bit Name	Initial value	R/W	Description
15 14	PE7MD1 PE7MD0	0 0	R/W R/W	PTE7 Mode 00: Peripheral module (MFI/LCDC) 01: Port output 10: Port input (pull-up MOS: Off) 11: Port input (pull-up MOS: On)
13 12	PE6MD1 PE6MD0	0 0	R/W R/W	PTE6 Mode 00: Peripheral module (MFI/LCDC) 01: Port output 10: Port input (pull-up MOS: Off) 11: Port input (pull-up MOS: On)
11 10	PE5MD1 PE5MD0	0 0	R/W R/W	PTE5 Mode 00: Peripheral module (MFI/LCDC) 01: Port output 10: Port input (pull-up MOS: Off) 11: Port input (pull-up MOS: On)
9 8	PE4MD1 PE4MD0	0 0	R/W R/W	PTE4 Mode 00: Peripheral module (MFI/LCDC) 01: Port output 10: Port input (pull-up MOS: Off) 11: Port input (pull-up MOS: On)
7 6	PE3MD1 PE3MD0	0 0	R/W R/W	PTE3 Mode 00: Peripheral module (MFI/LCDC) 01: Port output 10: Port input (pull-up MOS: Off) 11: Port input (pull-up MOS: On)
5 4	PE2MD1 PE2MD0	0 0	R/W R/W	PTE2 Mode 00: Peripheral module (MFI/LCDC) 01: Port output 10: Port input (pull-up MOS: Off) 11: Port input (pull-up MOS: On)

3	PE1MD1	0	R/W	PTE1 Mode
2	PE1MD0	0	R/W	00: Peripheral module (IRQ/LCDC) 01: Port output 10: Port input (pull-up MOS: Off) 11: Port input (pull-up MOS: On)
1	PE0MD1	0	R/W	PTE0 Mode
0	PE0MD0	0	R/W	00: Peripheral module (IRQ/LCDC) 01: Port output 10: Port input (pull-up MOS: Off) 11: Port input (pull-up MOS: On)

24.2.6 Port F Control Register (PFCR)

PFCR is a 16-bit readable/writable register that selects the pin function and input pull-up MOS control.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	PF3 MD1	PF3 MD0	PF2 MD1	PF2 MD0	PF1 MD1	PF1 MD0	PF0 MD1	PF0 MD0
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

Bit	Bit Name	Initial value	R/W	Description
15 to 8	—	All 0	R	Reserved These bits are always read as 0, and the write value should always be 0.
7	PF3MD1	0	R/W	PTF3 Mode
6	PF3MD0	0	R/W	00: Peripheral module (HSPI/MMCIF/SIM) 01: Port output 10: Port input (pull-up MOS: Off) 11: Port input (pull-up MOS: On)
5	PF2MD1	0	R/W	PTF2 Mode
4	PF2MD0	0	R/W	00: Peripheral module (HSPI/MMCIF/SIM) 01: Port output 10: Port input (pull-up MOS: Off) 11: Port input (pull-up MOS: On)
3	PF1MD1	0	R/W	PTF1 Mode
2	PF1MD0	0	R/W	00: Peripheral module (HSPI/MMCIF/SIM) 01: Port output 10: Port input (pull-up MOS: Off) 11: Port input (pull-up MOS: On)

1	PF0MD1	0	R/W	PTF0 Mode
0	PF0MD0	0	R/W	00: Peripheral module (HSPI/MMCIF/SIM) 01: Port output 10: Port input (pull-up MOS: Off) 11: Port input (pull-up MOS: On)

24.2.7 Port G Control Register (PGCR)

PGCR is a 16-bit readable/writable register that selects the pin function and input pull-up MOS control.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PG7 MD1	PG7 MD0	PG6 MD1	PG6 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:	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	Bit Name	Initial value	R/W	Description
15	PG7MD1	0	R/W	PTG7 Mode
14	PG7MD0	0	R/W	00: Peripheral module (SCIF[0]) 01: Port output 10: Port input (pull-up MOS: Off) 11: Port input (pull-up MOS: On)
13	PG6MD1	0	R/W	PTG6 Mode
12	PG6MD0	0	R/W	00: Peripheral module (SCIF[0]) 01: Port output 10: Port input (pull-up MOS: Off) 11: Port input (pull-up MOS: On)
11	PG5MD1	0	R/W	PTG5 Mode
10	PG5MD0	0	R/W	00: Peripheral module (SCIF[0]) 01: Port output 10: Port input (pull-up MOS: Off) 11: Port input (pull-up MOS: On)
9	PG4MD1	0	R/W	PTG4 Mode
8	PG4MD0	0	R/W	00: Peripheral module (SCIF[1]) 01: Port output 10: Port input (pull-up MOS: Off) 11: Port input (pull-up MOS: On)

7	PG3MD1	0	R/W	PTG3 Mode
6	PG3MD0	0	R/W	00: Peripheral module (SCIF[1]) 01: Port output 10: Port input (pull-up MOS: Off) 11: Port input (pull-up MOS: On)
5	PG2MD1	0	R/W	PTG2 Mode
4	PG2MD0	0	R/W	00: Peripheral module (SCIF[1]) 01: Port output 10: Port input (pull-up MOS: Off) 11: Port input (pull-up MOS: On)
3	PG1MD1	0	R/W	PTG1 Mode
2	PG1MD0	0	R/W	00: Peripheral module (SCIF[1]) 01: Port output 10: Port input (pull-up MOS: Off) 11: Port input (pull-up MOS: On)
1	PG0MD1	0	R/W	PTG0 Mode
0	PG0MD0	0	R/W	00: Peripheral module (SCIF[1]) 01: Port output 10: Port input (pull-up MOS: Off) 11: Port input (pull-up MOS: On)

24.2.8 Port H Control Register (PHCR)

PHCR is a 16-bit readable/writable register that selects the pin function and input pull-up MOS control.

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:	0	0	0	0	0	0	0	0	0	0	1	1	1	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

Bit	Bit Name	Initial value	R/W	Description
15	PH7MD1	0	R/W	PTH7 Mode
14	PH7MD0	0	R/W	00: Peripheral module (SCIF[2]) 01: Port output 10: Port input (pull-up MOS: Off) 11: Port input (pull-up MOS: On)

13	PH6MD1	0	R/W	PTH6 Mode
12	PH6MD0	0	R/W	00: Peripheral module (SCIF[2]) 01: Port output 10: Port input (pull-up MOS: Off) 11: Port input (pull-up MOS: On)
11	PH5MD1	0	R/W	PTH5 Mode
10	PH5MD0	0	R/W	00: Peripheral module (SCIF[2]) 01: Port output 10: Port input (pull-up MOS: Off) 11: Port input (pull-up MOS: On)
9	PH4MD1	0	R/W	PTH4 Mode
8	PH4MD0	0	R/W	00: Peripheral module (SCIF[2]) 01: Port output 10: Port input (pull-up MOS: Off) 11: Port input (pull-up MOS: On)
7	PH3MD1	0	R/W	PTH3 Mode
6	PH3MD0	0	R/W	00: Peripheral module (SCIF[2]) 01: Port output 10: Port input (pull-up MOS: Off) 11: Port input (pull-up MOS: On)
5	PH2MD1	1	R/W	PTH2 Mode
4	PH2MD0	1	R/W	00: Peripheral module (USB) 01: Port output 10: Port input (pull-up MOS: Off) 11: Port input (pull-up MOS: On)
3	PH1MD1	1	R/W	PTH1 Mode
2	PH1MD0	1	R/W	00: Peripheral module (USB) 01: Port output 10: Port input (pull-up MOS: Off) 11: Port input (pull-up MOS: On)
1	PH0MD1	0	R/W	PTH0 Mode
0	PH0MD0	0	R/W	00: Peripheral module (USB) 01: Port output 10: Port input (pull-up MOS: Off) 11: Port input (pull-up MOS: On)

PJCR is a 16-bit readable/writable register that selects the pin function and input pull-up MOS control.

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	-	-
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	R

Bit	Bit Name	Initial value	R/W	Description
15	PJ7MD1	0	R/W	PTJ7 Mode
14	PJ7MD0	0	R/W	00: Peripheral module (HAC[0]/SSI[0]) 01: Port output 10: Port input (pull-up MOS: Off) 11: Port input (pull-up MOS: On)
13	PJ6MD1	0	R/W	PTJ6 Mode
12	PJ6MD0	0	R/W	00: Peripheral module (HAC[0][1]) 01: Port output 10: Port input (pull-up MOS: Off) 11: Port input (pull-up MOS: On)
11	PJ5MD1	0	R/W	PTJ5 Mode
10	PJ5MD0	0	R/W	00: Peripheral module (SSI[1]/HAC[1]) 01: Port output 10: Port input (pull-up MOS: Off) 11: Port input (pull-up MOS: On)
9	PJ4MD1	0	R/W	PTJ4 Mode*
8	PJ4MD0	0	R/W	00: Peripheral module (SSI[1]/HAC[1]) 01: Port output 10: Port input (pull-up MOS: Off) 11: Port input (pull-up MOS: On)
7	PJ3MD1	0	R/W	PTJ3 Mode*
6	PJ3MD0	0	R/W	00: Peripheral module (SSI[1]/HAC[1]) 01: Port output 10: Port input (pull-up MOS: Off) 11: Port input (pull-up MOS: On)
5	PJ2MD1	0	R/W	PTJ2 Mode
4	PJ2MD0	0	R/W	00: Peripheral module (HAC[1]/SSI[1]) 01: Port output 10: Port input (pull-up MOS: Off) 11: Port input (pull-up MOS: On)

3	PJ1MD1	0	R/W	PTJ1 Mode
2	PJ1MD0	0	R/W	00: Peripheral module (DCK) 01: Port output Other than above: Setting prohibited
1, 0	—	All 0	R	Reserved These bits are always read as 0, and the write value should always be 0.

Note: *The following settings are prohibited: the combination of port3 set to output and port 4 to input or port3 to input and port4 to output.

24.2.10 Port K Control Register (PKCR)

PKCR is a 16-bit readable/writable register that selects the pin function and input pull-up MOS control.

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	-	-	-	-
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

Bit	Bit Name	Initial value	R/W	Description
15	PK7MD1	0	R/W	PTK7 Mode
14	PK7MD0	0	R/W	00: Peripheral module (Reserved/AUD) 01: Port output 10: Port input (pull-up MOS: Off) 11: Port input (pull-up MOS: On)
13	PK6MD1	0	R/W	PTK6 Mode
12	PK6MD0	0	R/W	00: Peripheral module (Reserved/AUD) 01: Port output 10: Port input (pull-up MOS: Off) 11: Port input (pull-up MOS: On)
11	PK5MD1	0	R/W	PTK5 Mode
10	PK5MD0	0	R/W	00: Peripheral module (Reserved/AUD) 01: Port output 10: Port input (pull-up MOS: Off) 11: Port input (pull-up MOS: On)

9	PK4MD1	0	R/W	PTK4 Mode
8	PK4MD0	0	R/W	00: Peripheral module (Reserved/AUD) 01: Port output 10: Port input (pull-up MOS: Off) 11: Port input (pull-up MOS: On)
7	PK3MD1	0	R/W	PTK3 Mode
6	PK3MD0	0	R/W	00: Peripheral module (Reserved/AUD) 01: Port output 10: Port input (pull-up MOS: Off) 11: Port input (pull-up MOS: On)
5	PK2MD1	0	R/W	PTK2 Mode
4	PK2MD0	0	R/W	00: Peripheral module (ADC/AUD) 01: Port output 10: Port input (pull-up MOS: Off) 11: Port input (pull-up MOS: On)
3 to 0	—	All 0	R	Reserved These bits are always read as 0, and the write value should always be 0.

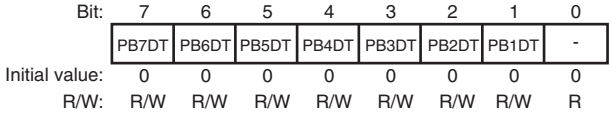
24.2.11 Port A Data Register (PADR)

PADR is an 8-bit readable/writable register that stores port A data.

Bit:	7	6	5	4	3	2	1	0
	PA7DT	PA6DT	PA5DT	PA4DT	PA3DT	PA2DT	-	-
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	R

Bit	Bit Name	Initial value	R/W	Description
7	PA7DT	0	R/W	These bits store output data of a pin which is used as a general output port. When the pin functions as a general output port, if the port is read, the value of this corresponding register will be read out. When the pin functions as a general input port, if the port is read, the status of the corresponding pin will be read out.
6	PA6DT	0	R/W	
5	PA5DT	0	R/W	
4	PA4DT	0	R/W	
3	PA3DT	0	R/W	
2	PA2DT	0	R/W	
1, 0	—	All 0	R	Reserved These bits are always read as 0, and the write value should always be 0.

PBDR is an 8-bit readable/writable register that stores port B data.



Bit:	7	6	5	4	3	2	1	0
	PB7DT	PB6DT	PB5DT	PB4DT	PB3DT	PB2DT	PB1DT	-
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	Bit Name	Initial value	R/W	Description
7	PB7DT	0	R/W	These bits store output data of a pin which is used as a general output port. When the pin functions as a general output port, if the port is read, the value of this corresponding register will be read out. When the pin functions as a general input port, if the port is read, the status of the corresponding pin will be read out.
6	PB6DT	0	R/W	
5	PB5DT	0	R/W	
4	PB4DT	0	R/W	
3	PB3DT	0	R/W	
2	PB2DT	0	R/W	
1	PB1DT	0	R/W	
0	—	0	R	Reserved This bit is always read as 0, and the write value should always be 0.

PCDR is an 8-bit readable/writable register that stores port C data.

Bit:	7	6	5	4	3	2	1	0
	PC7DT	PC6DT	PC5DT	PC4DT	PC3DT	PC2DT	PC1DT	PC0DT
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	Bit Name	Initial value	R/W	Description
7	PC7DT	0	R/W	These bits store output data of a pin which is used as a general output port. When the pin functions as a general output port, if the port is read, the value of this corresponding register will be read out. When the pin functions as a general input port, if the port is read, the status of the corresponding pin will be read out.
6	PC6DT	0	R/W	
5	PC5DT	0	R/W	
4	PC4DT	0	R/W	
3	PC3DT	0	R/W	
2	PC2DT	0	R/W	
1	PC1DT	0	R/W	
0	PC0DT	0	R/W	

24.2.14 Port D Data Register (PDDR)

PDDR is an 8-bit readable/writable register that stores port D data.

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	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial value	R/W	Description
7	PD7DT	0	R/W	These bits store output data of a pin which is used as a general output port. When the pin functions as a general output port, if the port is read, the value of this corresponding register will be read out. When the pin functions as a general input port, if the port is read, the status of the corresponding pin will be read out.
6	PD6DT	0	R/W	
5	PD5DT	0	R/W	
4	PD4DT	0	R/W	
3	PD3DT	0	R/W	
2	PD2DT	0	R/W	
1	PD1DT	0	R/W	
0	PD0DT	0	R/W	

PEDR is an 8-bit readable/writable register that stores port E data.

Bit:	7	6	5	4	3	2	1	0
	PE7DT	PE6DT	PE5DT	PE4DT	PE3DT	PE2DT	PE1DT	PE0DT
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	Bit Name	Initial value	R/W	Description
7	PE7DT	0	R/W	These bits store output data of a pin which is used as a general output port. When the pin functions as a general output port, if the port is read, the value of this corresponding register will be read out. When the pin functions as a general input port, if the port is read, the status of the corresponding pin will be read out.
6	PE6DT	0	R/W	
5	PE5DT	0	R/W	
4	PE4DT	0	R/W	
3	PE3DT	0	R/W	
2	PE2DT	0	R/W	
1	PE1DT	0	R/W	
0	PE0DT	0	R/W	

24.2.16 Port F Data Register (PFDR)

PFDR is an 8-bit readable/writable register that stores port F data.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	PF3DT	PF2DT	PF1DT	PF0DT
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial value	R/W	Description
7 to 4	—	All 0	R	Reserved These bits are always read as 0, and the write value should always be 0.
3	PF3DT	0	R/W	These bits store output data of a pin which is used as a general output port. When the pin functions as a general output port, if the port is read, the value of this corresponding register will be read out. When the pin functions as a general input port, if the port is read, the status of the corresponding pin will be read out.
2	PF2DT	0	R/W	
1	PF1DT	0	R/W	
0	PF0DT	0	R/W	

PGDR is an 8-bit readable/writable register that stores port G data.

Bit:	7	6	5	4	3	2	1	0
	PG7DT	PG6DT	PG5DT	PG4DT	PG3DT	PG2DT	PG1DT	PG0DT
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	Bit Name	Initial value	R/W	Description
7	PG7DT	0	R/W	These bits store output data of a pin which is used as a general output port. When the pin functions as a general output port, if the port is read, the value of this corresponding register will be read out. When the pin functions as a general input port, if the port is read, the status of the corresponding pin will be read out.
6	PG6DT	0	R/W	
5	PG5DT	0	R/W	
4	PG4DT	0	R/W	
3	PG3DT	0	R/W	
2	PG2DT	0	R/W	
1	PG1DT	0	R/W	
0	PG0DT	0	R/W	

24.2.18 Port H Data Register (PHDR)

PHDR is an 8-bit readable/writable register that stores port H data.

Bit:	7	6	5	4	3	2	1	0
	PH7DT	PH6DT	PH5DT	PH4DT	PH3DT	PH2DT	PH1DT	PH0DT
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	Bit Name	Initial value	R/W	Description
7	PH7DT	0	R/W	These bits store output data of a pin which is used as a general output port. When the pin functions as a general output port, if the port is read, the value of this corresponding register will be read out. When the pin functions as a general input port, if the port is read, the status of the corresponding pin will be read out.
6	PH6DT	0	R/W	
5	PH5DT	0	R/W	
4	PH4DT	0	R/W	
3	PH3DT	0	R/W	
2	PH2DT	0	R/W	
1	PH1DT	0	R/W	
0	PH0DT	0	R/W	

PJDR is an 8-bit readable/writable register that stores port J data.

Bit:	7	6	5	4	3	2	1	0
	PJ7DT	PJ6DT	PJ5DT	PJ4DT	PJ3DT	PJ2DT	PJ1DT	-
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	Bit Name	Initial value	R/W	Description
7	PJ7DT	0	R/W	These bits store output data of a pin which is used as a general output port. When the pin functions as a general output port, if the port is read, the value of this corresponding register will be read out. When the pin functions as a general input port, if the port is read, the status of the corresponding pin will be read out. However, Bit 1 is exclusively used as an output port. Note: When port3 and port4 function as general input ports, the state of port3 is read from the PJ4DT bit and the state of port4 is read from the PJ3DT bit.
6	PJ6DT	0	R/W	
5	PJ5DT	0	R/W	
4	PJ4DT	0	R/W	
3	PJ3DT	0	R/W	
2	PJ2DT	0	R/W	
1	PJ1DT	0	R/W	
0	—	0	R	Reserved This bit is always read as 0, and the write value should always be 0.

24.2.20 Port K Data Register (PKDR)

PKDR is an 8-bit readable/writable register that stores port K data.

Bit:	7	6	5	4	3	2	1	0
	PK7DT	PK6DT	PK5DT	PK4DT	PK3DT	PK2DT	-	-
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	R

Bit	Bit Name	Initial value	R/W	Description
7	PK7DT	0	R/W	These bits store output data of a pin which is used as a general output port. When the pin functions as a general output port, if the port is read, the corresponding value of this register will be read out. When the pin functions as a general input port, if the port is read, the status of the corresponding pin will be read out.
6	PK6DT	0	R/W	
5	PK5DT	0	R/W	
4	PK4DT	0	R/W	
3	PK3DT	0	R/W	
2	PK2DT	0	R/W	

1, 0	—	All 0	R	Reserved These bits are always read as 0, and the write value should always be 0.

24.2.21 GPIO Interrupt Control Register (GPIOIC)

GPIOIC is a 16-bit readable/writable register that controls interrupt inputs.

When an IRQ or IRL interrupt is used to cancel software standby mode, the standby cancellation IRL enable bit (STBIRLEN), described in section 10.5.2 Bus Control Register 2 (BCR2), should be set to 1.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PTIR EN15	PTIR EN14	PTIR EN13	PTIR EN12	PTIR EN11	PTIR EN10	PTIR EN9	STB RT8	STB RT7	STB RT6	STB IRQ5	STB IRQ4	STB IRL3	STB IRL2	STB IRL1	STB IRL0
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	Bit Name	Initial value	R/W	Description
15	PTIREN15	0	R/W	Port Interrupt Enable
14	PTIREN14	0	R/W	The setting whether to use the port as a GPIO
13	PTIREN13	0	R/W	interrupt can be set for each bit.
12	PTIREN12	0	R/W	0: Uses the port as a normal I/O port
11	PTIREN11	0	R/W	1: Uses the port as a GPIO interrupt
10	PTIREN10	0	R/W	
9	PTIREN9	0	R/W	
8	STBRT8	0	R/W	Standby Cancellation Setting Bit
7	STBRT7	0	R/W	In standby mode, if an interrupt to return from
6	STBRT6	0	R/W	standby mode is detected while the corresponding STBRT bit is set to 1, the standby mode is cancelled. 0: Disables canceling standby mode by detection of an interrupt 1: Enables canceling standby mode by detection of an interrupt
5	STBIRQ5	0	R/W	Standby Cancellation IRQ Setting Bit
4	STBIRQ4	0	R/W	In standby mode, if an IRQ interrupt (IRQ is in the low level) is detected while the corresponding STBIRQ bit is set to 1, the standby mode is cancelled. 0: Disables canceling standby mode by detection of an IRQ interrupt 1: Enables canceling standby mode by detection of an IRQ interrupt

3	STBIRL3	0	R/W	Standby Cancellation IRL Setting Bit
2	STBIRL2	0	R/W	In standby mode, if an IRL interrupt (IRL is in the
1	STBIRL1	0	R/W	low level) is detected while the corresponding
0	STBIRL0	0	R/W	STBIRL bit is set to 1, the standby mode is cancelled. 0: Disables canceling standby mode by detection of an IRL interrupt 1: Enables canceling standby mode by detection of an IRL interrupt

The following table shows the relationship between the GPIOIC bits and the corresponding pins to which interrupts are generated.

Bit Name	Pin	Pin Name	Port	Interrupt Type
PTIREN15	T2	MFI-RW/LCD_FLM	PTE2	GPIO interrupt
PTIREN14	T1	MFI-RS/LCD_M_DISP	PTE3	
PTIREN13	R2	MFI-MD/LCD_CL2	PTE4	
PTIREN12	R1	MFI-E/LCD_CL1	PTE5	
PTIREN11	P2	$\overline{\text{MFI-CS}}$/LCD_DON	PTE6	
PTIREN10	P1	$\overline{\text{MFI-INT}}$/LCD_CLK	PTE7	
PTIREN9	A13	$\overline{\text{SCIF2\_CTS}}$	PTH6	
STBRT8	A14	$\overline{\text{SCIF2\_RTS}}$	PTH5	
STBRT7	C18	USB_PENC	PTH1	
STBRT6	J20	$\overline{\text{FCE}}$/AUDATA[3]	PTK7	
STBIRQ5	E2	VEPWC/$\overline{\text{IRQ5}}$	PTE0	IRQ interrupt
STBIRQ4	E1	VCPWC/$\overline{\text{IRQ4}}$	PTE1	
STBIRL3	M19	$\overline{\text{IRL3}}$		IRL interrupt
STBIRL2	M20	$\overline{\text{IRL2}}$		
STBIRL1	N19	$\overline{\text{IRL1}}$		
STBIRL0	N20	$\overline{\text{IRL0}}$		

When a port pin is used for a GPIO interrupt source, it must be specified for an input pin through the corresponding port control register. The GPIO interrupt is a low-active level-sensed interrupt. After a GPIO interrupt is accepted, the GPIO pin level must be retained until the interrupt processing starts.

The values of the bits specified as GPIO interrupt sources are ORed to detect an interrupt. The bit that has received an interrupt can be checked by reading the port control registers.

to section 9, Interrupt Controller (INTC).

24.2.22 Port A Pull-Up Control Register (PAPUPR)

PAPUPR is an 8-bit readable/writable register that individually controls the pull-up for pins PTA7 to PTA2 corresponding to each bit in the register when the given pin is used by a peripheral module. However, for the pins set to the GPIO in the PACR, the settings in this register will be invalid.

Bit:	7	6	5	4	3	2	1	0
	PA7 PUPR	PA6 PUPR	PA5 PUPR	PA4 PUPR	PA3 PUPR	PA2 PUPR	-	-
Initial value:	1	1	1	1	1	1	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R	R

Bit	Bit Name	Initial Value	R/W	Description
7	PA7PUPR	1	R/W	Sets individual pull-up control for given pins of Port A 0: PTAn pull-up off 1: PTAn pull-up on
6	PA6PUPR	1	R/W	
5	PA5PUPR	1	R/W	
4	PA4PUPR	1	R/W	
3	PA3PUPR	1	R/W	
2	PA2PUPR	1	R/W	
1, 0	—	All 0	R	Reserved These bits are always read as 0, and the write value should always be 0.

n = 7 to 2

24.2.23 Port B Pull-Up Control Register (PBPUPR)

PBPUPR is an 8-bit readable/writable register that individually controls the pull-up for pins PTB7 to PTB1 corresponding to each bit in the register when the given pin is used by a peripheral module. However, for the pins set to the GPIO in the PBCR, the settings in this register will be invalid.

Bit:	7	6	5	4	3	2	1	0
	PB7 PUPR	PB6 PUPR	PB5 PUPR	PB4 PUPR	PB3 PUPR	PB2 PUPR	PB1 PUPR	-
Initial value:	1	1	1	1	1	1	1	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

7	PB7PUPR	1	R/W	Sets individual pull-up control for pins of Port B
6	PB6PUPR	1	R/W	0: PTBn pull-up off
5	PB5PUPR	1	R/W	1: PTBn pull-up on
4	PB4PUPR	1	R/W	
3	PB3PUPR	1	R/W	
2	PB2PUPR	1	R/W	
1	PB1PUPR	1	R/W	
0	—	0	R	Reserved This bit is always read as 0, and the write value should always be 0.

n = 7 to 1

24.2.24 Port C Pull-Up Control Register (PCPUPR)

PCPUPR is an 8-bit readable/writable register that individually controls the pull-up for pins PTC7 to PTC0 corresponding to each bit in the register when the given pin is used by a peripheral module. However, for the pins set to the GPIO in the PCCR, the settings in this register will be invalid.

Bit:	7	6	5	4	3	2	1	0
	PC7PUPR	PC6PUPR	PC5PUPR	PC4PUPR	PC3PUPR	PC2PUPR	PC1PUPR	PC0PUPR
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	Bit Name	Initial Value	R/W	Description
7	PC7PUPR	1	R/W	Sets individual pull-up control for each pin of Port C
6	PC6PUPR	1	R/W	0: PTCn pull-up off
5	PC5PUPR	1	R/W	1: PTCn pull-up on
4	PC4PUPR	1	R/W	
3	PC3PUPR	1	R/W	
2	PC2PUPR	1	R/W	
1	PC1PUPR	1	R/W	
0	PC0PUPR	1	R/W	

n = 7 to 0

PDPUPR is an 8-bit readable/writable register that individually controls the pull-up for pins PTD7 to PTD0 corresponding to each bit in the register when the given pin is used by a peripheral module. However, for the pins set to the GPIO in the PDCR, the settings in this register will be invalid.

Bit:	7	6	5	4	3	2	1	0
	PD7 PUPR	PD6 PUPR	PD5 PUPR	PD4 PUPR	PD3 PUPR	PD2 PUPR	PD1 PUPR	PD0 PUPR
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	Bit Name	Initial Value	R/W	Description
7	PD7PUPR	1	R/W	Sets individual pull-up control for each pin of Port D
6	PD6PUPR	1	R/W	0: PTDn pull-up off
5	PD5PUPR	1	R/W	1: PTDn pull-up on
4	PD4PUPR	1	R/W	
3	PD3PUPR	1	R/W	
2	PD2PUPR	1	R/W	
1	PD1PUPR	1	R/W	
0	PD0PUPR	1	R/W	

n = 7 to 0

24.2.26 Port E Pull-Up Control Register (PEPUPR)

PEPUPR is an 8-bit readable/writable register that individually controls the pull-up for pins PTE7 to PTE0 corresponding to each bit in the register when the given pin is used by a peripheral module. However, for the pins set to the GPIO in the PECR, the settings in this register will be invalid.

Bit:	7	6	5	4	3	2	1	0
	PE7 PUPR	PE6 PUPR	PE5 PUPR	PE4 PUPR	PE3 PUPR	PE2 PUPR	PE1 PUPR	PE0 PUPR
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

7	PE7PUPR	1	R/W	Sets individual pull-up control for each pin of Port E
6	PE6PUPR	1	R/W	0: PTEn pull-up off
5	PE5PUPR	1	R/W	1: PTEn pull-up on
4	PE4PUPR	1	R/W	
3	PE3PUPR	1	R/W	
2	PE2PUPR	1	R/W	
1	PE1PUPR	1	R/W	
0	PE0PUPR	1	R/W	

n = 7 to 0

24.2.27 Port F Pull-Up Control Register (PFPUPR)

PFPUPR is an 8-bit readable/writable register that individually controls the pull-up for pins PTF3 to PTF0 corresponding to each bit in the register when the given pin is used by a peripheral module. However, for the pins set to the GPIO in the PFCR, the settings in this register will be invalid.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	PF3 PUPR	PF2 PUPR	PF1 PUPR	PF0 PUPR
Initial value:	0	0	0	0	1	1	1	1
R/W:	R	R	R	R	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 0	R	Reserved These bits are always read as 0, and the write value should always be 0.
3	PF3PUPR	1	R/W	Sets individual pull-up control of each pin of Port F
2	PF2PUPR	1	R/W	0: PTFn pull-up off
1	PF1PUPR	1	R/W	1: PTFn pull-up on
0	PF0PUPR	1	R/W	

n = 3 to 0

PGPUPR is an 8-bit readable/writable register that individually controls the pull-up for pins PTG7 to PTG0 corresponding to each bit in the register when the given pin is used by a peripheral module. However, for the pins set to the GPIO in the PGCR, the settings in this register will be invalid.

Bit:	7	6	5	4	3	2	1	0
	PG7 PUPR	PG6 PUPR	PG5 PUPR	PG4 PUPR	PG3 PUPR	PG2 PUPR	PG1 PUPR	PG0 PUPR
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	Bit Name	Initial Value	R/W	Description
7	PG7PUPR	1	R/W	Sets individual pull-up control for each pin of Port G
6	PG6PUPR	1	R/W	0: PTGn pull-up off
5	PG5PUPR	1	R/W	1: PTGn pull-up on
4	PG4PUPR	1	R/W	
3	PG3PUPR	1	R/W	
2	PG2PUPR	1	R/W	
1	PG1PUPR	1	R/W	
0	PG0PUPR	1	R/W	

n = 7 to 0

24.2.29 Port H Pull-Up Control Register (PHPUPR)

PHPUPR is an 8-bit readable/writable register that individually controls the pull-up for pins PTH7 to PTH0 corresponding to each bit in the register when the given pin is used by a peripheral module. However, for the pins set to the GPIO in the PHCR, the settings in this register will be invalid.

Bit:	7	6	5	4	3	2	1	0
	PH7 PUPR	PH6 PUPR	PH5 PUPR	PH4 PUPR	PH3 PUPR	PH2 PUPR	PH1 PUPR	PH0 PUPR
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

7	PH7PUPR	1	R/W	Sets individual pull-up control for each pin of Port H
6	PH6PUPR	1	R/W	0: PTHn pull-up off
5	PH5PUPR	1	R/W	1: PTHn pull-up on
4	PH4PUPR	1	R/W	
3	PH3PUPR	1	R/W	
2	PH2PUPR	1	R/W	
1	PH1PUPR	1	R/W	
0	PH0PUPR	1	R/W	

n = 7 to 0

24.2.30 Port J Pull-Up Control Register (PJPUPR)

PJPUPR is an 8-bit readable/writable register. Each bit of this register corresponds to PTJ7 to PTJ2, and when the pins of Port J are used by the peripheral modules, the pull-up control is performed individually. However, for the pins set to the GPIO in the PJCR, the settings in this register will be invalid.

Bit:	7	6	5	4	3	2	1	0
	PJ7 PUPR	PJ6 PUPR	PJ5 PUPR	PJ4 PUPR	PJ3 PUPR	PJ2 PUPR	-	-
Initial value:	1	1	1	1	1	1	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R	R

Bit	Bit Name	Initial Value	R/W	Description
7	PJ7PUPR	1	R/W	Pull-up control of the pins of Port J can be set individually.
6	PJ6PUPR	1	R/W	
5	PJ5PUPR	1	R/W	0: PTJn pull-up off
4	PJ4PUPR	1	R/W	1: PTJn pull-up on
3	PJ3PUPR	1	R/W	
2	PJ2PUPR	1	R/W	
1, 0	—	All 0	R	Reserved This bit is always read as 0, and the write value should always be 0.

n = 7 to 2

PKPUPR is an 8-bit readable/writable register that individually controls the pull-up for pins PTK7 to PTK2 corresponding to each bit in the register when the given pin is used by a peripheral module. However, for the pins set to the GPIO in the PKCR, the settings in this register will be invalid.

Bit:	7	6	5	4	3	2	1	0
	PK7 PUPR	PK6 PUPR	PK5 PUPR	PK4 PUPR	PK3 PUPR	PK2 PUPR	-	-
Initial value:	1	1	1	1	1	1	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R	R

Bit	Bit Name	Initial Value	R/W	Description
7	PK7PUPR	1	R/W	Sets individual pull-up control of pins of Port K 0: PTKn pull-up off 1: PTKn pull-up on
6	PK6PUPR	1	R/W	
5	PK5PUPR	1	R/W	
4	PK4PUPR	1	R/W	
3	PK3PUPR	1	R/W	
2	PK2PUPR	1	R/W	
1, 0	—	All 0	R	Reserved These bits are always read as 0, and the write value should always be 0.

n = 7 to 2

MDPUPR is an 8-bit readable/writable register that individually controls the pull-up for the pins connected to each bit of the register field.

Bit:	7	6	5	4	3	2	1	0
	MD PUPR7	MD PUPR6	MD PUPR5	MD PUPR4	MD PUPR3	MD PUPR2	MD PUPR1	MD PUPR0
Initial value:	0	0	1	1	1	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	MDPUPR7	0	R/W	Controls pull-up of MD8 0: MD8 pull-up off 1: MD8 pull-up on
6	MDPUPR6	0	R/W	Controls pull-up of MD7 0: MD7 pull-up off 1: MD7 pull-up on
5	MDPUPR5	1	R/W	Controls pull-up of MD5 0: MD5 pull-up off 1: MD5 pull-up on
4	MDPUPR4	1	R/W	Controls pull-up of MD4/$\overline{\text{CE2B}}$ 0: MD4/$\overline{\text{CE2B}}$ pull-up off 1: MD4/$\overline{\text{CE2B}}$ pull-up on
3	MDPUPR3	1	R/W	Controls pull-up of MD3/$\overline{\text{CE2A}}$ 0: MD3/$\overline{\text{CE2A}}$ pull-up off 1: MD3/$\overline{\text{CE2A}}$ pull-up on
2	MDPUPR2	0	R/W	Controls pull-up of MD2 0: MD2 pull-up off 1: MD2 pull-up on
1	MDPUPR1	0	R/W	Controls pull-up of MD1 0: MD1 pull-up off 1: MD1 pull-up on
0	MDPUPR0	0	R/W	Controls pull-up of MD0 0: MD0 pull-up off 1: MD0 pull-up on

INPUPA is a 16-bit readable/writable register that individually controls the pull-up for the pin connected to each bit of the register field.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MD6 PUP	RDY PUP	BREQ PUP	IRL0 PUP	IRL1 PUP	IRL2 PUP	IRL3 PUP	NMI PUP	-	-	-	-	-	-	-	-
Initial value:	1	1	1	1	1	1	1	1	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	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15	MD6PUP	1	R/W	Controls pull-up of MD6/$\overline{\text{IOIS16}}$ 0: MD6/$\overline{\text{IOIS16}}$ pull-up off 1: MD6/$\overline{\text{IOIS16}}$ pull-up on
14	RDYPUP	1	R/W	Controls pull-up of $\overline{\text{RDY}}$ 0: $\overline{\text{RDY}}$ pull-up off 1: $\overline{\text{RDY}}$ pull-up on
13	BREQPUP	1	R/W	Controls pull-up of $\overline{\text{BREQ}}$ 0: $\overline{\text{BREQ}}$ pull-up off 1: $\overline{\text{BREQ}}$ pull-up on
12	IRL0PUP	1	R/W	Controls pull-up of $\overline{\text{IRL0}}$ 0: $\overline{\text{IRL0}}$ pull-up off 1: $\overline{\text{IRL0}}$ pull-up on
11	IRL1PUP	1	R/W	Controls pull-up of $\overline{\text{IRL1}}$ 0: $\overline{\text{IRL1}}$ pull-up off 1: $\overline{\text{IRL1}}$ pull-up on
10	IRL2PUP	1	R/W	Controls pull-up of $\overline{\text{IRL2}}$ 0: $\overline{\text{IRL2}}$ pull-up off 1: $\overline{\text{IRL2}}$ pull-up on
9	IRL3PUP	1	R/W	Controls pull-up of $\overline{\text{IRL3}}$ 0: $\overline{\text{IRL3}}$ pull-up off 1: $\overline{\text{IRL3}}$ pull-up on
8	NMIPUP	1	R/W	Controls pull-up of NMI 0: NMI pull-up off 1: NMI pull-up on
7 to 0	—	All 0	R	Reserved These bits are always read as 0, and the write value should always be 0.

DMAPCR is a 16-bit readable/writable register that controls the states of the DMAC pins ($\overline{\text{DREQ0}}$, $\overline{\text{DREQ1}}$, DRAK0, DRAK1, DACK0, and DACK1) and DMABRG reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DREQP0	-	DREQP1	-	DACKP0	DACKD0	DACKP1	DACKD1	DRAKP0	DRAKD0	DRAKP1	DRAKD1	-	-	-	BRGRST
Initial value:	1	0	1	0	0	1	0	1	0	1	0	1	0	0	0	0
R/W:	R/W	R	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R/W

Bit	Bit Name	Initial value	R/W	Description
15	DREQP0	1	R/W	Controls pull-up for $\overline{\text{DREQ0}}$. 0: $\overline{\text{DREQ0}}$ pull-up off 1: $\overline{\text{DREQ0}}$ pull-up on
14	—	0	R	Reserved This bit is always read as 0, and the write value should always be 0.
13	DREQP1	1	R/W	Controls the pull-up for $\overline{\text{DREQ1}}$. 0: $\overline{\text{DREQ1}}$ pull-up off 1: $\overline{\text{DREQ1}}$ pull-up on
12	—	0	R	Reserved This bit is always read as 0, and the write value should always be 0.
11 10	DACKP0 DACKD0	0 1	R/W R/W	Controls the pin state for DACK0 in software standby mode. 00: Hi-Z state 01: Output 10: Hi-Z state with pull-up on 11: Setting prohibited
9 8	DACKP1 DACKD1	0 1	R/W R/W	Controls the pin state for DACK1 in software standby mode. 00: Hi-Z state 01: Output 10: Hi-Z state with pull-up on 11: Setting prohibited
7 6	DRAKP0 DRAKD0	0 1	R/W R/W	Controls the pin state for DRAK0 in software standby mode. 00: Hi-Z state 01: Output 10: Hi-Z state with pull-up on 11: Setting prohibited

5 4	DRAKP1 DRAKD1	0 1	R/W R/W	Controls the pin state for DRAK1 in software standby mode. 00: Hi-Z state 01: Output 10: Hi-Z state with pull-up on 11: Setting prohibited
3 to 1	—	All 0	R	Reserved These bits are always read as 0, and the write value should always be 0.
0	BRGRST	0	R/W	Controls a DMABRG reset. 0: Cancels DMABRG reset 1: Resets DMABRG Note: For the BRGRST usage, refer to section 11.6.2, DMABRG Reset.

IPSELR is a 16-bit readable/writable register. Modules using pins multiplexed are specified by this register when the modules are not dependent on either MFI mode or LCD mode. For details of pin multiplexing, see table 24.1.

This register is valid only when peripheral modules are selected by PACR, PBCR (PTB7 to PTB5), PDCR, PFCR, or PJCR (PJT5 to PJT3) of the GPIO.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	IPSELR 15	IPSELR 14	IPSELR 13	IPSELR 12	IPSELR 11	IPSELR 10	IPSELR 9	-	-	-	-	-	-	-	LCDMD 1	LCDMD 0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	IPSELR15	0	R/W	Out of the modules HSPI, MMCIF, and SIM, select the one using the pins HSPI_TX/SIM_D/MCDAT, HSPI_RX, HSPI_CLK/SIM_CLK/MCCLK, and $\overline{\text{HSPI\_CS}}$/SIM_RST/MCCMD.
14	IPSELR14	0	R/W	00: Selects HSPI 01: Selects MMCIF 10: Setting prohibited 11: Selects SIM
13	IPSELR13	0	R/W	Out of the modules HCAN2 and AUD, select the one using the pins CAN0_NERR/AUDCK, CAN0_RX/AUDATA[2], CAN0_TX/AUDATA[0], CAN1_NERR/AUDSYNC, CAN1_RX/AUDATA[3], and CAN1_TX/AUDATA[1]. 0: Selects HCAN2 1: Selects AUD
12	IPSELR12	0	R/W	Out of the modules ADC and AUD, select the one using the pins Reserved/AUDATA[3], Reserved/AUDATA[2], Reserved/AUDATA[1], Reserved/AUDCK, Reserved/AUDSYNC, and $\overline{\text{ADTRG}}$/AUDATA[0]. 0: Reserved/ADC* 1: AUD Note: *Pull up the reserved pins internally in the PKCR register settings.

11	IPSELR11	0	R/W	Out of the modules SSI[1]/[0] and HAC[1]/[0], select the one using the pins
10	IPSELR10	0	R/W	SSI0_SCK/HAC_SD_IN0/$\overline{BS2}$, SSI0_WS/HAC_SYNC0, SSI0_SDATA/HAC_SD_OUT0, SSI1_SCK/HAC_SD_IN1, SSI1_SDATA/HAC_SD_OUT1, SSI1_WS/HAC_SYNC1, HAC_BIT_CLK0, and HAC_BIT_CLK1 00: SSI[0], SSI[1] 01: HAC[0]. SSI[1] 10:Setting prohibited 11:HAC[0], HAC[1]
9	IPSELR9	0	R/W	Select the pins MFI-D8/LCD_DATA8 to MFI-D15/LCD_DATA15 of MFI/LCDC. 0: MFI/LCDC 1: Setting prohibited
8 to 2	—	All 0	R	Reserved These bits are always read as 0, and the write value should always be 0.
1	LCDMD1	1	R/W	LDCD mode settings
0	LCDMD0	1	R/W	00: Mode 1 LCD_CL1 and LCD_FLM output LCD_CL2 output 01: Mode 2 LCD_CL1 and LCD_FLM output LCD_CL2 Hiz Other than above: Setting prohibited Note: When using the LCDC, be sure to set these bits to B'00 or B'01.

SCIHZR is a 16-bit readable/writable register that controls the pin state of the individual port pins in software standby mode when a Peripheral module (SCIF0 to SCIF2) is selected by the PGCR or PHCR. When each bit is 0, the corresponding pin retains the state before the software standby mode is entered. This register settings is ignored when the pins are specified as the GPIO port by PGCR or PHCR. This register setting is also ignored when the SCIF controls the pins.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SCI CLK0	SCI RXD0	SCI TXD0	SCI CLK1	SCI CTS1	SCI RTS1	SCI RXD1	SCI TXD1	SCI CLK2	SCI CTS2	SCI RTS2	SCI RXD2	SCI TXD2	-	-	-
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	R	R

Bit	Bit Name	Initial Value	R/W	Description
15	SCICLK0	0	R/W	0: Sets SCIF0_CLK Hi-Z state to off 1: Sets SCIF0_CLK Hi-Z state to on
14	SCIRXD0	0	R/W	0: Sets SCIF0_RXD Hi-Z state to off 1: Sets SCIF0_RXD Hi-Z state to on
13	SCITXD0	0	R/W	0: Sets SCIF0_TXD Hi-Z state to off 1: Sets SCIF0_TXD Hi-Z state to on
12	SCICLK1	0	R/W	0: Sets SCIF1_CLK Hi-Z state to off 1: Sets SCIF1_CLK Hi-Z state to on
11	SCICTS1	0	R/W	0: Sets $\overline{\text{SCIF1\_CTS}}$ Hi-Z state to off 1: Sets $\overline{\text{SCIF1\_CTS}}$ Hi-Z state to on
10	SCIRTS1	0	R/W	0: Sets $\overline{\text{SCIF1\_RTS}}$ Hi-Z state to off 1: Sets $\overline{\text{SCIF1\_RTS}}$ Hi-Z state to on
9	SCIRXD1	0	R/W	0: Sets SCIF1_RXD Hi-Z state to off 1: Sets SCIF1_RXD Hi-Z state to on
8	SCITXD1	0	R/W	0: Sets SCIF1_TXD Hi-Z state to off 1: Sets SCIF1_TXD Hi-Z state to on
7	SCICLK2	0	R/W	0: Sets SCIF2_CLK Hi-Z state to off 1: Sets SCIF2_CLK Hi-Z state to on
6	SCICTS2	0	R/W	0: Sets $\overline{\text{SCIF2\_CTS}}$ Hi-Z state to off 1: Sets $\overline{\text{SCIF2\_CTS}}$ Hi-Z state to on
5	SCIRTS2	0	R/W	0: Sets $\overline{\text{SCIF2\_RTS}}$ Hi-Z state to off 1: Sets $\overline{\text{SCIF2\_RTS}}$ Hi-Z state to on
4	SCIRXD2	0	R/W	0: Sets SCIF2_RXD Hi-Z state to off 1: Sets SCIF2_RXD Hi-Z state to on

3	SCITXD2	0	R/W	0: Sets SCIF2_TXD Hi-Z state to off 1: Sets SCIF2_TXD Hi-Z state to on
2 to 0	—	All 0	R	Reserved These bits are always read as 0, and the write value should always be 0.

24.2.37 Mode Select Register (MODSELR)

MODESELR is an 8-bit readable/writable register that individually sets the mode of pins MFI-D2 to MFI-D7 and SSI0_SCK. When MFI mode/LCD mode is used, modules should be selected in IPSELR. However, when these pins are used as GPIOs, the settings in this register will be invalid.

Bit:	7	6	5	4	3	2	1	0
	MOD SELR7	MOD SELR6	MOD SELR5	MOD SELR4	MOD SELR3	MOD SELR2	MOD SELR1	-
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	Bit Name	Initial Value	R/W	Description
7	MODSELR7	0	R/W	Selects the mode of pin MFI-D2/LCD_DATA2/$\overline{\text{IRQ6}}$ 0: MFI mode/LCD mode (MFI/LCDC) 1: $\overline{\text{IRQ6}}$
6	MODSELR6	0	R/W	Selects the mode of pin MFI-D3/LCD_DATA3/$\overline{\text{IRQ7}}$ 0: MFI mode/LCD mode (MFI/LCDC) 1: $\overline{\text{IRQ7}}$
5	MODSELR5	0	R/W	Selects the mode of pin MFI-D4/LCD_DATA4/$\overline{\text{DREQ2}}$ 0: MFI mode/LCD mode (MFI/LCDC) 1: $\overline{\text{DREQ2}}$
4	MODSELR4	0	R/W	Selects the mode of pin MFI-D5/LCD_DATA5/DRAK2/DACK2 0: MFI mode/LCD mode (MFI/LCDC) 1: DRAK2/DACK2
3	MODSELR3	0	R/W	Selects the mode of pin MFI-D6/LCD_DATA6/$\overline{\text{DREQ3}}$ 0: MFI mode/LCD mode (MFI/LCDC) 1: $\overline{\text{DREQ3}}$

2	MODSELR2	0	R/W	Selects the mode of pin MFI-D7/LCD_DATA7/DRAK3/DACK3 0: MFI mode/LCD mode (MFI/LCDC) 1: DRAK3/DACK3
1	MODSELR1	0	R/W	Selects the mode of pin SSI0_SCK/HAC_SD_IN0/$\overline{BS2}$ 0: MFI mode/LCD mode (SSI/HAC) 1: $\overline{BS2}$
0	—	0	R	Reserved This bit is always read as 0, and the write value should always be 0.

The HAC, the audio codec digital controller interface, supports bidirectional data transfer compliant with Audio Codec 97 (AC'97) Version 2.1. The HAC provides serial transmission to /reception from the AC97 codec. Each channel of the HAC can be connected to a single audio codec device.

The HAC carries out data extraction from/insertion into audio frames. For data slots within both receive and transmit frames, the PIO transfer by the CPU or the DMA transfer by the DMAC can be used.

25.1 Features

The HAC has the following features:

- Digital interface to a single AC'97 version 2.1 Audio Codec
- PIO transfer of status slots 1 and 2 in Rx frames
- PIO transfer of command slots 1 and 2 in Tx frames
- PIO transfer of data slots 3 and 4 in Rx frames
- PIO transfer of data slots 3 and 4 in Tx frames
- Selectable 16-bit or 20-bit DMA transfer of data slots 3 and 4 in Rx frames
- Selectable 16-bit or 20-bit DMA transfer of data slots 3 and 4 in Tx frames
- Accommodates various sampling rates by qualifying slot data with tag bits and monitoring the Tx frame request bits of Rx frames
- Generates data ready, data request, overrun and underrun interrupts
- Supports cold reset, warm reset, and power-down mode

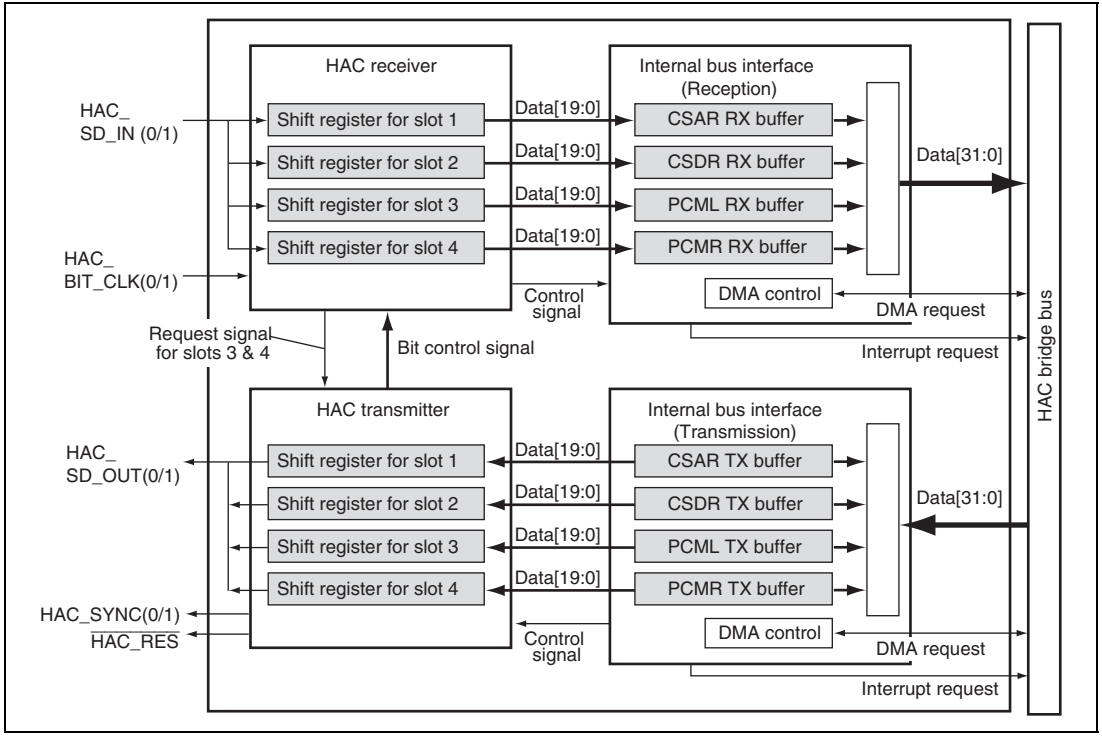


Figure 25.1 Block Diagram

25.2 Input/Output Pins

Table 25.1 describes the HAC pin configuration.

Table 25.1 Pin Configuration

Name	# of Pins	I/O	Function
HAC_BIT_CLK (0/1)	1	Input	HAC serial data clock
HAC_SD_IN (0/1)	1	Input	HAC serial data incoming to Rx frame
HAC_SD_OUT (0/1)	1	Output	HAC serial data outgoing from Tx frame
HAC_SYNC (0/1)	1	Output	HAC frame sync
$\overline{\text{HAC\_RES}}$	1	Output	HAC reset (negative logic signal) (common to channels 0 and 1)

This section describes the HAC registers. For details of register addresses and register statuses in each processing, see section 32, List of Registers. Since these registers function in the same way in both channels 0 and 1, they are not discriminated by channel number in the description below.

Table 25.2 Register Configuration (1)

Ch.	Register Name	Abbrev.	R/W	P4 Address	Area 7 Address	Size	Sync Clock
0	Control and status register 0	HACCR0	R/W	H'FE24 0008	H'1E24 0008	32	Pck
	Command/status address register 0	HACCSAR0	R/W	H'FE24 0020	H'1E24 0020	32	Pck
	Command/status data register 0	HACCSDR0	R/W	H'FE24 0024	H'1E24 0024	32	Pck
	PCM left channel register 0	HACPCML0	R/W	H'FE24 0028	H'1E24 0028	32	Pck
	PCM right channel register 0	HACPCMR0	R/W	H'FE24 002C	H'1E24 002C	32	Pck
	TX interrupt enable register 0	HACTIER0	R/W	H'FE24 0050	H'1E24 0050	32	Pck
	TX status register 0	HACTSR0	R/W	H'FE24 0054	H'1E24 0054	32	Pck
	RX interrupt enable register 0	HACRIER0	R/W	H'FE24 0058	H'1E24 0058	32	Pck
	RX status register 0	HACRSR0	R/W	H'FE24 005C	H'1E24 005C	32	Pck
	HAC control register 0	HACACR0	R/W	H'FE24 0060	H'1E24 0060	32	Pck
1	Control and status register 1	HACCR1	R/W	H'FE25 0008	H'1E25 0008	32	Pck
	Command/status address register 1	HACCSAR1	R/W	H'FE25 0020	H'1E25 0020	32	Pck
	Command/status data register 1	HACCSDR1	R/W	H'FE25 0024	H'1E25 0024	32	Pck
	PCM left channel register 1	HACPCML1	R/W	H'FE25 0028	H'1E25 0028	32	Pck
	PCM right channel register 1	HACPCMR1	R/W	H'FE25 002C	H'1E25 002C	32	Pck
	TX interrupt enable register 1	HACTIER1	R/W	H'FE25 0050	H'1E25 0050	32	Pck
	TX status register 1	HACTSR1	R/W	H'FE25 0054	H'1E25 0054	32	Pck
	RX interrupt enable register 1	HACRIER1	R/W	H'FE25 0058	H'1E25 0058	32	Pck
	RX status register 1	HACRSR1	R/W	H'FE25 005C	H'1E25 005C	32	Pck
	HAC control register 1	HACACR1	R/W	H'FE25 0060	H'1E25 0060	32	Pck

Ch.	Register Name	Abbrev.	Power-on Reset by RESET Pin/WDT/ H-UDI	Manual Reset by RESET Pin/WDT/ Multiple Exceptions	Sleep by Sleep Instruction/ Deep Sleep	Standby	
						by Hardware	by Software /Each Module
0	Control and status register 0	HACCR0	H'0000 0200	H'0000 0200	Retained	*	Retained
	Command/status address register 0	HACCSAR0	H'0000 0000	H'0000 0000	Retained		Retained
	Command/status data register 0	HACCSDR0	H'0000 0000	H'0000 0000	Retained		Retained
	PCM left channel register 0	HACPCML0	H'0000 0000	H'0000 0000	Retained		Retained
	PCM right channel register 0	HACPCMR0	H'0000 0000	H'0000 0000	Retained		Retained
	TX interrupt enable register 0	HACTIER0	H'0000 0000	H'0000 0000	Retained		Retained
	TX status register 0	HACTSR0	H'0000 0000	H'0000 0000	Retained		Retained
	RX interrupt enable register 0	HACRIER0	H'0000 0000	H'0000 0000	Retained		Retained
	RX status register 0	HACRSR0	H'0000 0000	H'0000 0000	Retained		Retained
	HAC control register 0	HACACR0	H'8400 0000	H'8400 0000	Retained		Retained
1	Control and status register 1	HACCR1	H'0000 0200	H'0000 0200	Retained		Retained
	Command/status address register 1	HACCSAR1	H'0000 0000	H'0000 0000	Retained		Retained
	Command/status data register 1	HACCSDR1	H'0000 0000	H'0000 0000	Retained		Retained
	PCM left channel register 1	HACPCML1	H'0000 0000	H'0000 0000	Retained		Retained
	PCM right channel register 1	HACPCMR1	H'0000 0000	H'0000 0000	Retained		Retained
	TX interrupt enable register 1	HACTIER1	H'0000 0000	H'0000 0000	Retained		Retained
	TX status register 1	HACTSR1	H'0000 0000	H'0000 0000	Retained		Retained
	RX interrupt enable register 1	HACRIER1	H'0000 0000	H'0000 0000	Retained		Retained
	RX status register 1	HACRSR1	H'0000 0000	H'0000 0000	Retained		Retained
	HAC control register 1	HACACR1	H'8400 0000	H'8400 0000	Retained		Retained

Note: * After exiting hardware standby mode, this LSI enters the power-on reset state by the RESET pin.

HACCR is a 32-bit read/write register for controlling input/output and monitoring the interface status.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
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
	CR	-	-	-	CDRT	WMRT	-	-	-	-	ST	-	-	-	-	-
Initial value:	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	W	W	R	R	R	R	W	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	—	All 0	R	Reserved Always 0 for read and write.
15	CR	0	R	Codec Ready 0: The HAC-connected codec is not ready. 1: The HAC-connected codec is ready.
14 to 12	—	All 0	R	Reserved Always read as 0. Write prohibited.
11	CDRT	0	W	HAC Cold Reset Use a cold reset only after power-on, or only to exit from the power-down mode by the power-down command. [Write] 0: Always write 0 to this bit before writing 1 again. 1: Performs a cold reset on the HAC. [Read] Always read as 0.
10	WMRT	0	W	HAC Warm Reset Use a warm reset only after power-up, or only to exit from the power-down mode by the power-down command. [Write] 0: Always write 0 to this bit before writing 1 again. 1: Performs a warm reset on the HAC. [Read] Always read as 0.
9	—	1	R	Reserved Always 1 for read and write.

Bit	Bit Name	Initial Value	R/W	Description
8 to 6	—	All 0	R	Reserved Always 0 for read and write.
5	ST	0	W	Start Transfer [Write] 1: Starts data transmission/reception. 0: Stops data transmission/reception at the end of the current frame. Do not take this action to terminate transmission/reception in normal operation. [Read access] Always read as 0.
4 to 0	—	All 0	R	Reserved Always 0 for read and write.

To place the off-chip codec device into the power-down mode, write 1 to bit 12 of the register index 26 in the off-chip codec via the HAC. When entering the power-down mode, the off-chip codec stops HAC_BIT_CLK and suspends the normal operation. The off-chip codec acts in the same manner at power-on. To resume the normal operation, perform a cold reset or a warm reset on the off-chip codec.

25.3.2 Command/Status Address Register (HACCSAR)

HACCSAR is a 32-bit read/write register that specifies the address of the codec register to be read /written. When requesting a write to/read from a codec register, write the command register address to HACCSAR. Then the HAC transmits this register address to the codec via slot 1.

After the codec has responded to a read request (HACRSR.STARY = 1), the status address received via slot 1 can be read out from HACCSAR.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	RW	CA6/ SA6	CA5/ SA5	CA4/ SA4
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/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CA3/ SA3	CA2/ SA2	CA1/ SA1	CA0/ SA0	SLR EQ3	SLR EQ4	SLR EQ5	SLR EQ6	SLR EQ7	SLR EQ8	SLR EQ9	SLR EQ10	SLR EQ11	SLR EQ12	-	-
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	R	R	R	R	R	R	R	R	R	R	R

31 to 20	—	All 0	R	Reserved Always 0 for read and write.
19	RW	0	R/W	Codec Read/Write Command 0: Notifies the off-chip codec device of a write access to the register specified in the address field (CA6/SA6 to CA0/SA0). Write the data to HACCSDR in advance. When HACACR.TX12_ATOMIC is 1, the HAC transmits HACCSAR and HACCSDR as a pair in the same Tx frame. When HACACR.TX12_ATOMIC is 0, transmission of HACCSAR and HACCSDR in the same Tx frame is not guaranteed. 1: Notifies the off-chip codec device of a read access to the register specified in the address field (CA6/SA6 to CA0/SA0).
18 17 16 15 14 13 12	CA6/SA6 CA5/SA5 CA4/SA4 CA3/SA3 CA2/SA2 CA1/SA1 CA0/SA0	0 0 0 0 0 0 0	R/W R/W R/W R/W R/W R/W R/W	Codec Control Register Addresses 6 to 0/ Codec Status Register Addresses 6 to 0 [Write] Specify the address of the codec register to be written. [Read] Indicate the status address received via slot 1, corresponding to the codec register whose data has been returned in HACCSDR.
11 10 9 8 7 6 5 4 3 2	SLREQ3 SLREQ4 SLREQ5 SLREQ6 SLREQ7 SLREQ8 SLREQ9 SLREQ10 SLREQ11 SLREQ12	0 0 0 0 0 0 0 0 0 0	R R R R R R R R R R	Slot Requests 3 to 12 Valid only in the Rx frame. Indicate whether the codec is requesting slot data in the next Tx frame. Automatically set by hardware, and correspond to bits 11 to 2 of slot 1 in the Rx frame. 0: Slot data is requested. 1: Slot data is not requested.
1, 0	—	All 0	R	Reserved Always 0 for read and write.

HACCSDR is a 32-bit read/write data register used for accessing the codec register. Write the command data to HACCSDR. The HAC then transmits the data to the codec via slot 2.

After the codec has responded to a read request (HACRSR.STDRY = 1), the status data received via slot 2 can be read out from HACCSDR. In both read and write, HACCSAR stores the related codec register address.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	CD15/ SD15	CD14/ SD14	CD13/ SD13	CD12/ SD12
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/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CD11/ SD11	CD10/ SD10	CD9/ SD9	CD8/ SD8	CD7/ SD7	CD6/ SD6	CD5/ SD5	CD4/ SD4	CD3/ SD3	CD2/ SD2	CD1/ SD1	CD0/ SD0	-	-	-	-
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

Bit	Bit Name	Initial Value	R/W	Description
31 to 20	—	All 0	R	Reserved Always 0 for read and write.
19	CD15/SD15	0	R/W	Command Data 15 to 0/Status Data 15 to 0 Write data to these bits and then write the codec register address in HACCSAR. The HAC then transmits the data to the codec. Read these bits to get the contents of the codec register indicated by HACCSAR.
18	CD14/SD14	0	R/W	
17	CD13/SD13	0	R/W	
16	CD12/SD12	0	R/W	
15	CD11/SD11	0	R/W	
14	CD10/SD10	0	R/W	
13	CD9/SD9	0	R/W	
12	CD8/SD8	0	R/W	
11	CD7/SD7	0	R/W	
10	CD6/SD6	0	R/W	
9	CD5/SD5	0	R/W	
8	CD4/SD4	0	R/W	
7	CD3/SD3	0	R/W	
6	CD2/SD2	0	R/W	
5	CD1/SD1	0	R/W	
4	CD0/SD0	0	R/W	
3 to 0	—	All 0	R	Reserved Always 0 for read and write.

HACPCML is a 32-bit read/write data register used for accessing the left channel of the codec in digital audio recording or stream playback. To transmit the PCM playback left channel data to the codec, write the data to HACPCML. To receive the PCM record left channel data from the codec, read HACPCML. The data is left justified to accommodate a codec with ADC/DAC resolution of 20 bits or less.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	D19	D18	D17	D16
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/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
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	Bit Name	Initial Value	R/W	Description
31 to 20	—	All 0	R	Reserved Always 0 for read and write.
19 to 0	D19 to D0	All 0	R/W	Data 19 to 0 Write the PCM playback left channel data to these bits. The HAC then transmits the data to the codec on an on-demand basis. Read these bits to get the PCM record left channel data from the codec.

In 16-bit packed DMA mode, HACPCML is defined as follows:

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	LD15	LD14	LD13	LD12	LD11	LD10	LD9	LD8	LD7	LD6	LD5	LD4	LD3	LD2	LD1	LD0
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
	RD15	RD14	RD13	RD12	RD11	RD10	RD9	RD8	RD7	RD6	RD5	RD4	RD3	RD2	RD1	RD0
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

31 to 16	LD15 to LD0	All 0	R/W	Left Data 15 to 0 Write the PCM playback left channel data to these bits. The HAC then transmits the data to the codec on an on-demand basis. Read these bits to get the PCM record left channel data from the codec.
15 to 0	RD15 to RD0	All 0	R/W	Right Data 15 to 0 Write the PCM playback right channel data to these bits. The HAC then transmits the data to the codec on an on-demand basis. Read these bits to get the PCM record right channel data from the codec.

25.3.5 PCM Right Channel Register (HACPCMR)

HACPCMR is a 32-bit read/write register used for accessing the right channel of the codec in digital audio recording or stream playback. To transmit the PCM playback right channel data to the codec, write the data to HACPCMR. To receive the PCM record right channel data from the codec, read HACPCMR. The data is left justified to accommodate a codec with ADC/DAC resolution of 20-bit or less.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	D19	D18	D17	D16
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/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
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	Bit Name	Initial Value	R/W	Description
31 to 20	—	All 0	R	Reserved Always 0 for read and write.
19 to 0	D19 to D0	All 0	R/W	Data 19 to 0 Write the PCM playback right channel data to these bits. The HAC then transmits the data to the codec on an on-demand basis. Read these bits to get the PCM record right channel data from the codec.

HACTIER is a 32-bit read/write register that enables or disables HAC TX interrupts.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	PLTFRQIE	PRTFRQIE	-	-	-	-	-	-	-	-	-	-	-	-
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/W	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
	-	-	-	-	-	-	PLTFUNIE	PRTFUNIE	-	-	-	-	-	-	-	-
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	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31, 30	—	All 0	R	Reserved Always 0 for read and write.
29	PLTFRQIE	0	R/W	PCML TX Request Interrupt Enable 0: Disables PCML TX request interrupts 1: Enables PCML TX request interrupts
28	PRTFRQIE	0	R/W	PCMR TX Request Interrupt Enable 0: Disables PCMR TX request interrupts 1: Enables PCMR TX request interrupts
27 to 10	—	All 0	R	Reserved Always 0 for read and write.
9	PLTFUNIE	0	R/W	PCML TX Underrun Interrupt Enable 0: Disables PCML TX underrun interrupts 1: Enables PCML TX underrun interrupts
8	PRTFUNIE	0	R/W	PCMR TX Underrun Interrupt Enable 0: Disables PCMR TX underrun interrupts 1: Enables PCMR TX underrun interrupts
7 to 0	—	All 0	R	Reserved Always 0 for read and write.

HACTSR is a 32-bit read/write register that indicates the status of the HAC TX controller. Writing 0 to the bit will initialize it.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CMD AMT	CMD DMT	PLT FRQ	PRT FRQ	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	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
	-	-	-	-	-	-	PLT FUN	PRT FUN	-	-	-	-	-	-	-	-
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	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W*2	Description
31	CMDAMT	1	R/W	Command Address Empty 0: CSAR Tx buffer contains untransmitted data. 1: CSAR Tx buffer is empty and ready to store data.*1 For details of HAC initialization steps, see the operational flow in 25.5.5.
30	CMDDMT	1	R/W	Command Data Empty 0: CSDR Tx buffer contains untransmitted data. 1: CSDR Tx buffer is empty and ready to store data. *1
29	PLTFRQ	1	R/W	PCML TX Request 0: PCML Tx buffer contains untransmitted data. 1: PCML TX buffer is empty and needs to store data. In DMA mode, writing to HACPCML will automatically clear this bit to 0.
28	PRTFRQ	1	R/W	PCMR TX Request 0: PCMR Tx buffer contains untransmitted data. 1: PCMR TX buffer is empty and needs to store data. In DMA mode, writing to HACPCMR will automatically clear this bit to 0.
27 to 10	—	All 0	R	Reserved Always 0 for read and write.

9	PLTFUN	0	R/W	PCML TX Underrun 0: No PCML TX underrun has occurred. 1: PCML TX underrun has occurred because the codec has requested slot 3 with PLTFRQ = 1.
8	PRTFUN	0	R/W	PCMR TX Underrun 0: No PCMR TX underrun has occurred. 1: PCMR TX underrun has occurred because the codec has requested slot 4 with PRTFRQ = 1.
7 to 0	—	All 0	R	Reserved Always 0 for read and write.

Notes: 1. CMDAMT and CMDDMT have no associated interrupts. Poll these bits until they are read as 1 before writing a new command to HACCSAR/HACCSDR. When bit 19 (RW) of HACCSAR is 0 and TX12_ATOMIC is 1, take the following steps:

1. Initialize CMDDMT and CMDAMT before first accessing a codec register after HAC initialization by any reset event.
2. After making the settings in HACCSDR and HACCSAR, poll CMDDMT and CMDAMT until they are cleared to 1, and then initialize these bits.
3. Now the next write to a register is available.

2. These bits are read/write. Writing 0 to the bit initializes it but writing 1 has no effect.

25.3.8 RX Interrupt Enable Register (HACRIER)

HACRIER is a 32-bit read/write register that enables or disables HAC RX interrupts.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	STARYIE	STDRYIE	PLRFRQIE	PRRFRQIE	-	-	-
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/W	R/W	R/W	R/W	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	PLRFOVIE	PRRFOVIE	-	-	-	-	-	-	-	-	-	-	-	-
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/W	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 23	—	All 0	R	Reserved Always 0 for read and write.
22	STARYIE	0	R/W	Status Address Ready Interrupt Enable 0: Disables status address ready interrupts. 1: Enables status address ready interrupts.

21	STDRYIE	0	R/W	Status Data Ready Interrupt Enable 0: Disables status data ready interrupts. 1: Enables status data ready interrupts.
20	PLRFRQIE	0	R/W	PCML RX Request Interrupt Enable 0: Disables PCML RX request interrupts. 1: Enables PCML RX request interrupts.
19	PRRFRQIE	0	R/W	PCMR RX Request Interrupt Enable 0: Disables PCMR RX request interrupts. 1: Enables PCMR RX request interrupts.
18 to 14	—	All 0	R	Reserved Always 0 for read and write.
13	PLRFOVIE	0	R/W	PCML RX Overrun Interrupt Enable 0: Disables PCML RX overrun interrupts. 1: Enables PCML RX overrun interrupts.
12	PRRFOVIE	0	R/W	PCMR RX Overrun Interrupt Enable 0: Disables PCMR RX overrun interrupts. 1: Enables PCMR RX overrun interrupts.
11 to 0	—	All 0	R	Reserved Always 0 for read and write.

25.3.9 RX Status Register (HACRSR)

HACRSR is a 32-bit read/write register that indicates the status of the HAC RX controller. Writing 0 to the bit will initialize it.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	STARY	STDRY	PLR FRQ	PRR FRQ	-	-	-
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/W	R/W	R/W	R/W	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	PLR FOV	PRR FOV	-	-	-	-	-	-	-	-	-	-	-	-
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/W	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W*	Description
31 to 23	—	All 0	R	Reserved Always 0 for read and write.

22	STARY	0	R/W	Status Address Ready 0: HACCSAR (status address) is not ready. 1: HACCSAR (status address) is ready.
21	STDRY	0	R/W	Status Data Ready 0: HACCSDR (status data) is not ready. 1: HACCSDR (status data) is ready.
20	PLRFRQ	0	R/W	PCML RX Request 0: PCML RX data is not ready. 1: PCML RX data is ready and must be read. In DMA mode, reading HACPCML automatically clears this bit to 0.
19	PRRFRQ	0	R/W	PCMR RX Request 0: PCMR RX data is not ready. 1: PCMR RX data is ready and must be read. In DMA mode, reading HACPCMR automatically clears this bit to 0.
18 to 14	—	All 0	R	Reserved Always 0 for read and write.
13	PLRFOV	0	R/W	PCML RX Overrun 0: No PCML RX data overrun has occurred. 1: PCML RX data overrun has occurred because the HAC has received new data from slot 3 with PLRFRQ = 1.
12	PRRFOV	0	R/W	PCMR RX Overrun 0: No PCMR RX data overrun has occurred. 1: PCMR RX data overrun has occurred because the HAC has received new data from slot 4 with PRRFRQ = 1.
11 to 0	—	All 0	R	Reserved Always 0 for read and write.

Note: * This register is read/write. Writing 0 to the bit initializes it but writing 1 has no effect.

HACACR is a 32-bit read/write register used for controlling the HAC interface.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	DMA RX16	DMA TX16	-	-	TX12_ ATOMIC	-	RXDMAL _EN	TXDMAL _EN	RXDMAR _EN	TXDMAR _EN	-	-	-	-	-
Initial value:	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R	R	R/W	R	R/W	R/W	R/W	R/W	R	R	R	R	R

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	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31	—	1	R	Reserved Always 1 for read and write.
30	DMARX16	0	R/W	16-bit RX DMA Enable 0: Disables 16-bit packed RX DMA mode. Enables the RXDMAL_EN and RXDMAR_EN settings. 1: Enables 16-bit packed RX DMA mode. Disables the RXDMAL_EN and RXDMAR_EN settings.
29	DMATX16	0	R/W	16-bit TX DMA Enable 0: Disables 16-bit packed TX DMA mode. Enables the TXDMAL_EN and TXDMAR_EN settings. 1: Enables 16-bit packed TX DMA mode. Disables the TXDMAL_EN and TXDMAR_EN settings.
28, 27	—	All 0	R	Reserved Always 0 for read and write.
26	TX12_ATOMIC	1	R/W	TX Slot 1 and 2 Atomic Control 0: Transmits TX data in HACCSAR and that in HACCSDR separately. (Setting prohibited) 1: Transmits TX data in HACCSAR and that in HACCSDR in the same frame if bit 19 in HACCSAR is 0 (write). (HACCSAR must be written last.)
25	—	0	R	Reserved Always 0 for read and write.

24	RXDMAL_EN	0	R/W	RX DMA Left Enable 0: Disables 20-bit RX DMA for HACPCML. 1: Enables 20-bit RX DMA is for HACPCML.
23	TXDMAL_EN	0	R/W	TX DMA Left Enable 0: Disables 20-bit TX DMA for HACPCML. 1: Enables 20-bit TX DMA for HACPCML.
22	RXDMAR_EN	0	R/W	RX DMA Right Enable 0: Disables 20-bit RX DMA for HACPCMR. 1: Enables 20-bit RX DMA for HACPCMR.
21	TXDMAR_EN	0	R/W	TX DMA Right Enable 0: Disables 20-bit TX DMA for HACPCMR. 1: Enables 20-bit TX DMA for HACPCMR.
20 to 0	—	All 0	R	Reserved Always 0 for read and write.

Figure 25.2 shows the AC97 frame slot structure. This LSI supports slots 0 to 4 only. Slots 5 to 12 are out of scope.

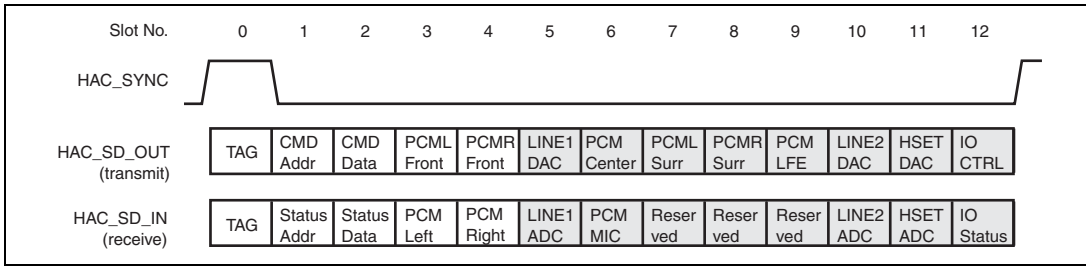


Figure 25.2 AC97 Frame Slot Structure

Table 25.3 AC97 Transmit Frame Structure

Slot	Name	Description
0	SDATA_OUT TAG	Codec IDs and Tags indicating valid data
1	Control CMD Addr write port	Read/write command and register address
2	Control DATA write port	Register write data
3	PCM L DAC playback	Left channel PCM output data
4	PCM R DAC playback	Right channel PCM output data
5	Modem Line 1 DAC	Modem 1 output data (unsupported)
6	PCM Center	Center channel PCM data (unsupported)
7	PCM Surround L	Surround left channel PCM data (unsupported)
8	PCM Surround R	Surround right channel PCM data (unsupported)
9	PCM LFE	LFE channel PCM data (unsupported)
10	Modem Line 2 DAC	Modem 2 output data (unsupported)
11	Modem handset DAC	Modem handset output data (unsupported)
12	Modem IO control	Modem control IO output (unsupported)

Slot	Name	Description
0	SDATA_IN TAG	Tags indicating valid data
1	Status ADDR read port	Register address and slot request
2	Status DATA read port	Register read data
3	PCM L ADC record	Left channel PCM input data
4	PCM R ADC record	Right channel PCM input data
5	Modem Line 1 ADC	Modem 1 input data (unsupported)
6	Dedicated Microphone ADC	Optional PCM data (unsupported)
7 to 9	Reserved	Reserved
10	Modem Line 2 ADC	Modem 2 input data (unsupported)
11	Modem handset input DAC	Modem handset input data (unsupported)
12	Modem IO status	Modem control IO input (unsupported)

25.5 Operation

25.5.1 Receiver

The HAC receiver receives serial audio data input on the HAC_SD_IN pin, synchronous to HAC_BIT_CLK. From slot 0, the receiver extracts tag bits that indicate which other slots contain valid data. It will update the receive data only when receiving valid slot data indicated by the tag bits.

Supporting data only in slots 1 to 4, the receiver ignores tag bits and data related to slots 5 to 12. It loads valid slot data to the corresponding shift register to hold the data for PIO or DMA transfer, and sets the corresponding status bits. It is possible to read 20-bit data within a 32-bit register using PIO.

In the case of RX overrun, the new data will overwrite the current data in the RX buffer of the HAC.

The HAC transmitter outputs serial audio data on the HAC_SD_OUT pin, synchronous to HAC_BIT_CLK. The transmitter sets the tag bits in slot 0 to indicate which slots in the current frame contain valid data. It loads data slots to the current TX frame in response to the corresponding slot request bits from the previous RX frame.

The transmitter supports data only in slots 1 to 4. The TX buffer holds data that has been transferred using PIO or DMA, and sets the corresponding status bit. It is possible to write 20-bit data within a 32-bit register using PIO.

In the case of a TX underrun, the HAC will transmit the current TX buffer data until the next data arrives.

25.5.3 DMA

The HAC supports DMA transfer for slots 3 and 4 of both the RX and TX frames. Specify the slot data size for DMA transfer, 16 or 20 bits, with the DMARX16 and DMATX16 bits in HACACR.

When the data size is 20 bits, transfer of data slots 3 and 4 requires two local bus access cycles. Since each of the receiver and transmitter has its DMA request, the stereo mode generates a DMA request for slots 3 and 4 separately. The mono mode generates a DMA request for just one slot.

When the data size is 16 bits, data from slots 3 and 4 are packed into a single 32-bit quantity (left data and right data are in PCML), which requires only one local bus access cycle.

It may be necessary to halt a DMA transfer before the end count is reached, depending on system applications. If so, clear the corresponding DMA bit in HACACR to 0 (DMA disabled). To resume a DMA transfer, reprogram the DMAC and then set the corresponding DMA bit to 1 (DMA enabled).

25.5.4 Interrupts

Interrupts can be used for flag events from the receiver and transmitter. Make the setting for each interrupt in the corresponding interrupt enable register. Interrupts include a request to the CPU to read/write slot data, overrun and underrun. To get the interrupt source, read the status register. Writing 0 to the bit will clear the corresponding interrupt.

Figure 25.3 shows an example of the initialization sequence.

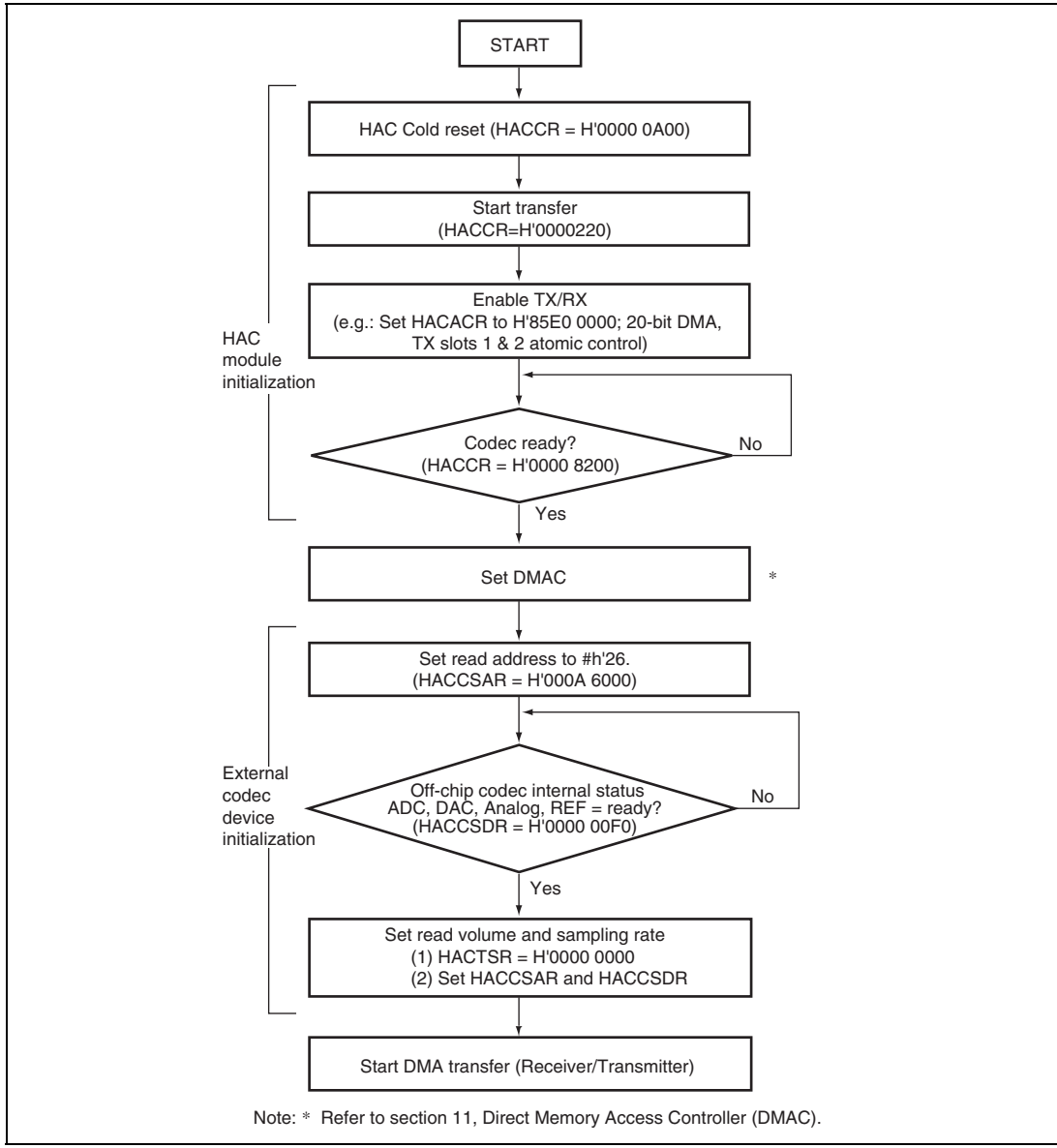
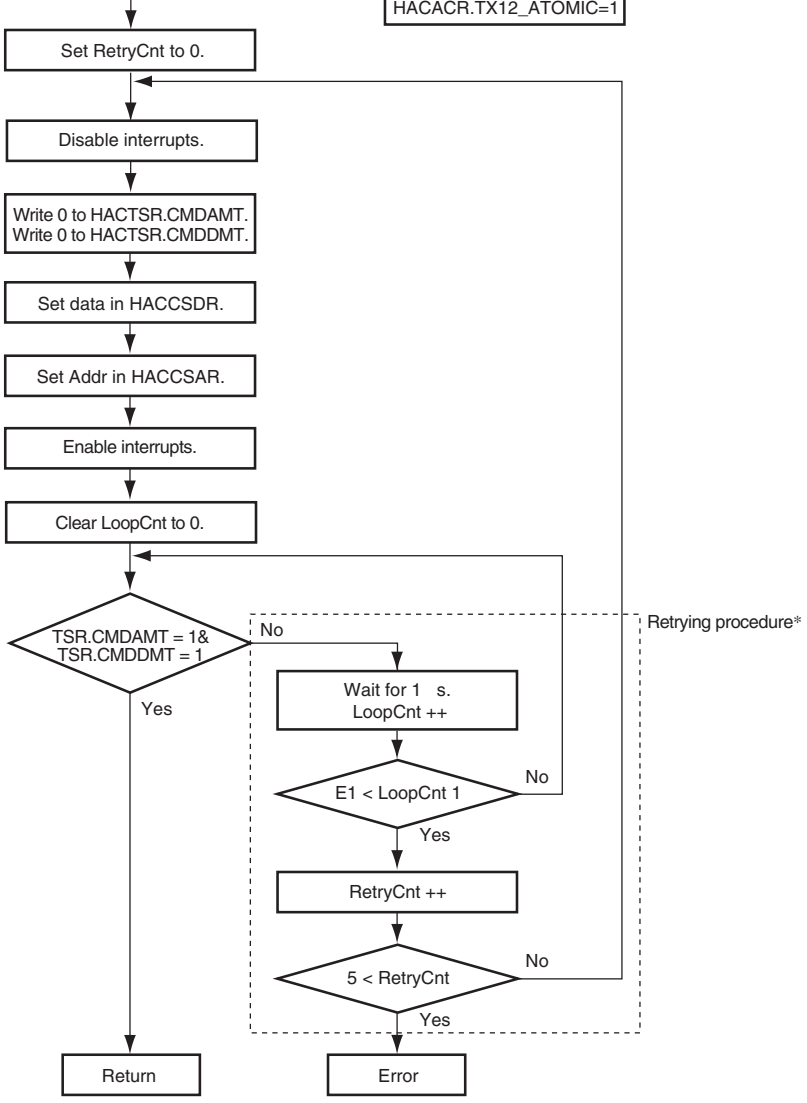


Figure 25.3 Initialization Sequence



Notes: E1: Loop count required in the target system
(21<E1<1000)
Input: Addr: Address of codec register to be written to
Data: Data to be written to codec register
RetryCnt: Software counter for error detection
LoopCnt: Software counter for wait insertion

* Some CODEC devices may not complete accessing CODEC register within 1 slot time.
In this case, please execute this retrying procedure.

Figure 25.4 Sample Flowchart for Off-Chip Codec Register Write

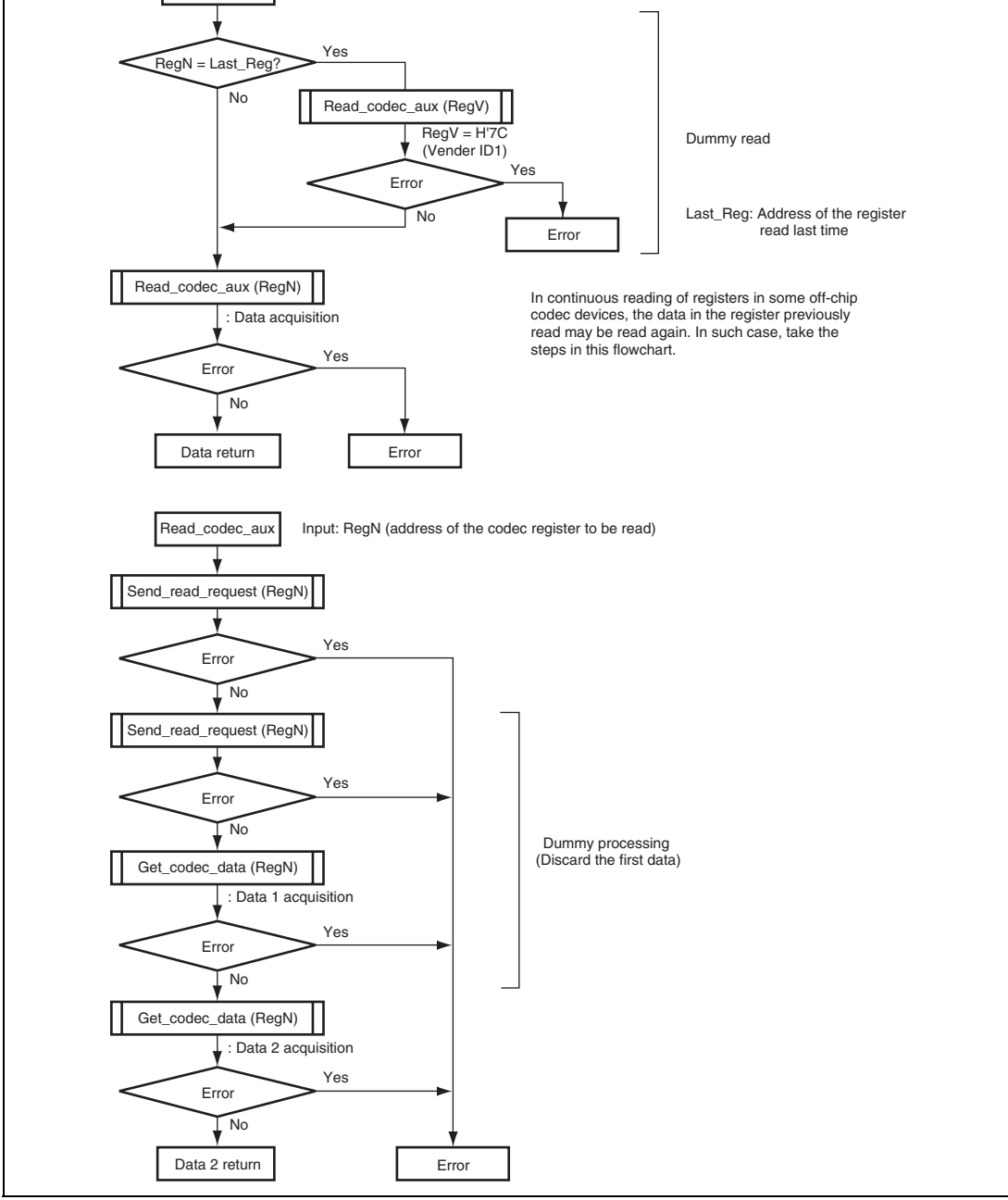
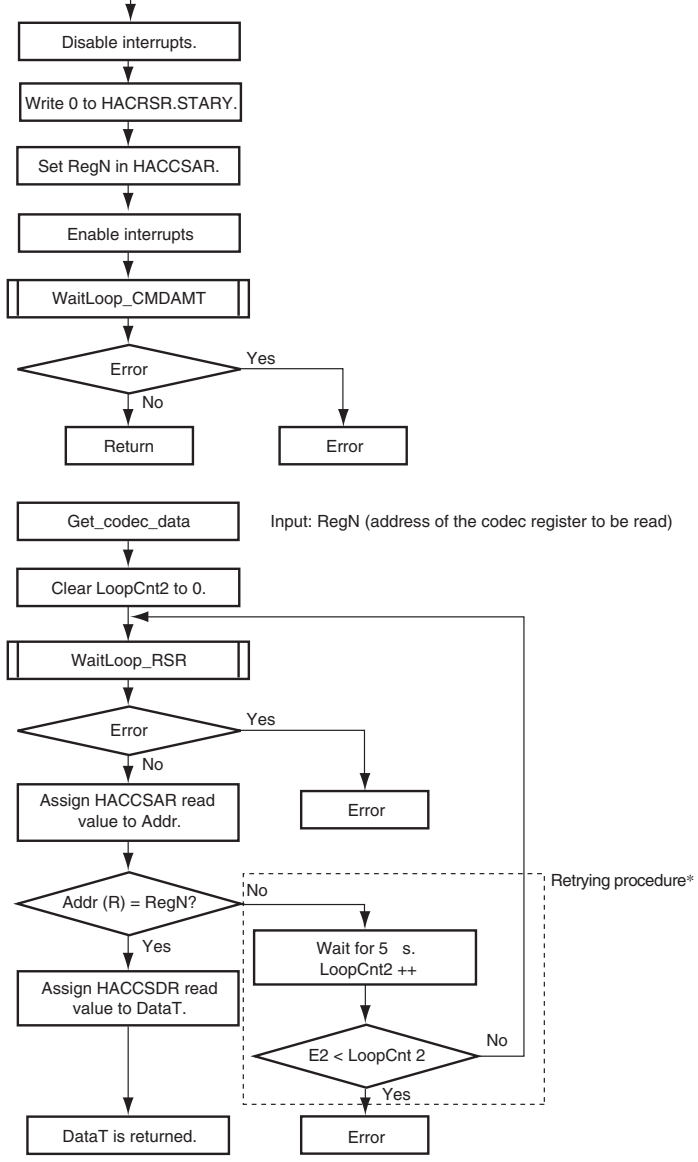


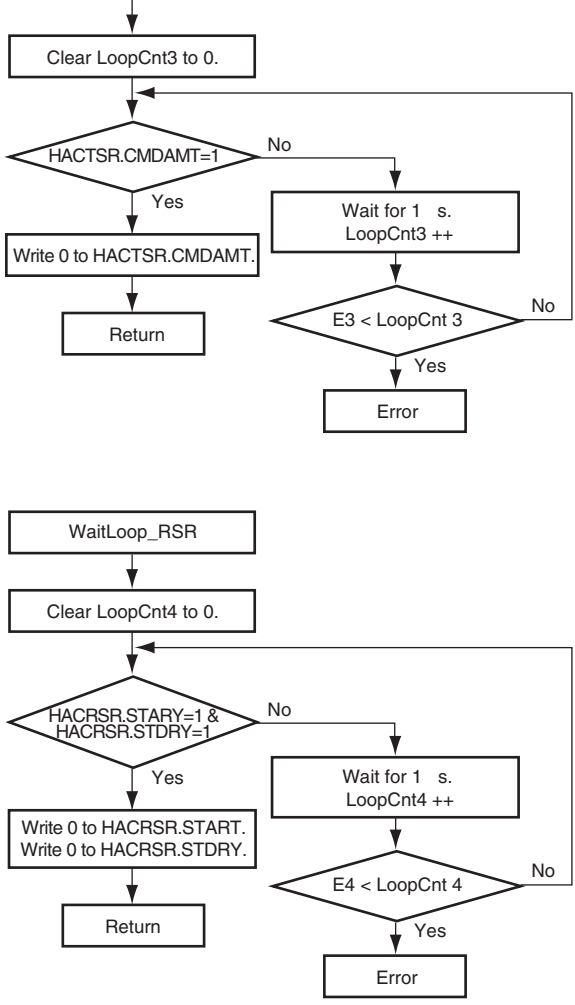
Figure 25.5 Sample Flowchart for Off-Chip Codec Register Read



Notes: E2: Loop count required in the target system
(13<E2)
LoopCnt2: Software counter for wait insertion
Addr: Variable to hold HACCSAR read value
DataT: Variable to hold HACCSDR read value

* Some CODEC devices may not complete accessing CODEC register within 1 slot time.
In this case, please execute this retrying procedure.

Figure 25.6 Sample Flowchart for Off-Chip Codec Register Read (cont)



Note: E3, E4: Loop count required in the target system
(21<E3, 21<E4<1000)
LoopCnt 3: Software counter for wait insertion
LoopCnt 4: Software counter for wait insertion

Figure 25.7 Sample Flowchart for Off-Chip Codec Register Read (cont)

It is possible to stop the supply of clock to the HAC using the CSTP15 and CSTP16 bits in CLKSTP00, which is a register used in power-down modes.

To cancel module standby mode and resume the supply of clock to the HAC, write 1 to the corresponding bits in CLKSTPCLR00. It enables all accesses to the HAC.

To place the HAC into the power-down mode, take the following steps:

1. Check that all data transfers have ended. Also check that the transmit buffer is empty and the receive buffer has been read out to be empty.
2. Disable all DMA requests and interrupt requests.
3. Place the codec into power-down mode.
4. Write 1 to the CSTP15 and CSTP16 bits in CLKSTP00.

25.5.7 Notes

The HAC_SYNC signal is generated by the HAC to indicate the position of slot 0 within a frame.

When using two channels of the HAC simultaneously, connect the $\overline{\text{HAC\_RES}}$ pin to the reset pins on both of the connected codecs.

25.5.8 Reference

AC'97 Component Specification, Revision 2.1

This LSI supports a multimedia card interface (MMCIF). The MMC mode interface can be utilized. The MMCIF is a clock-synchronous serial interface that transmits/receives data that is distinguished in terms of command and response. A number of commands/responses are predefined in the multimedia card. As the MMCIF specifies a command code and command type/response type upon the issuance of a command, commands extended by the secure multimedia card (Secure-MMC) and additional commands can be supported in the future within the range of combinations of currently defined command types/response types.

26.1 Features

The MMCIF has the following features:

- Interface that complies with The MultiMediaCard System Specification Version 2.11
- Supports MMC mode
- 20-Mbps bit rate (max.) for the card interface at a peripheral operating clock of 20 MHz
- Incorporates 64 data-transfer FIFOs of 16 bits
- Supports DMA transfer
- Four interrupt sources
  FIFO empty/full, command/response/data transfer complete, transfer error, and FIFO ready
- Interface via the MCCLK output (transfer clock output) pin, the MCCMD input/output (command output/response input) pin, and the MCDAT input/output (data input/output) pin

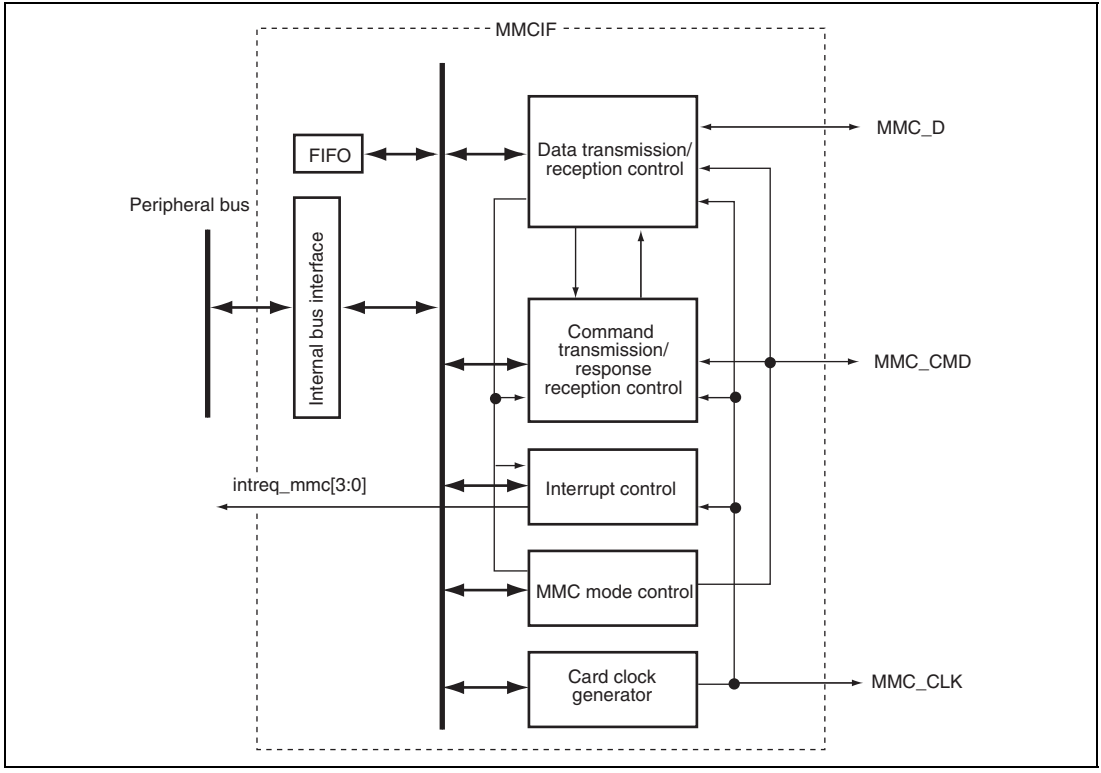


Figure 26.1 Block Diagram of MMCIF

26.2 Input/Output Pins

Table 26.1 summarizes the pins of the MMCIF.

Table 26.1 Pin Configuration

Symbol	I/O	Function
MCCLK	Output	Card clock output
MCCMD	Input/Output	Command output/response input
MCDAT	Input/Output	Data input/output

Note: For insertion/detachment of a card or for signals switching over between open-drain and CMOS modes, use ports of this LSI.

The MMCIF has the following registers. For more information on addresses of registers and register states in each processing, refer to section 32, List of Registers.

Table 26.2 Register Configuration (1)

Register Name	Abbrev.	R/W	P4 Address	Area 7 Address	Size	Sync Clock
Command register 0	CMDR0	R/W	H'FE50 0000	H'1E50 0000	8	Pck
Command register 1	CMDR1	R/W	H'FE50 0001	H'1E50 0001	8	Pck
Command register 2	CMDR2	R/W	H'FE50 0002	H'1E50 0002	8	Pck
Command register 3	CMDR3	R/W	H'FE50 0003	H'1E50 0003	8	Pck
Command register 4	CMDR4	R/W	H'FE50 0004	H'1E50 0004	8	Pck
Command register 5	CMDR5	—	H'FE50 0005	H'1E50 0005	8	Pck
Command start register	CMDSTRT	R/W	H'FE50 0006	H'1E50 0006	8	Pck
Operation control register	OPCR	R/W	H'FE50 000A	H'1E50 000A	8	Pck
Card status register	CSTR	R	H'FE50 000B	H'1E50 000B	8	Pck
Interrupt control register 0	INTCR0	R/W	H'FE50 000C	H'1E50 000C	8	Pck
Interrupt control register 1	INTCR1	R/W	H'FE50 000D	H'1E50 000D	8	Pck
Interrupt status register 0	INTSTR0	R/W	H'FE50 000E	H'1E50 000E	8	Pck
Interrupt status register 1	INTSTR1	R/W	H'FE50 000F	H'1E50 000F	8	Pck
Transfer clock control register	CLKON	R/W	H'FE50 0010	H'1E50 0010	8	Pck
Command timeout control register	CTOCR	R/W	H'FE50 0011	H'1E50 0011	8	Pck
Transfer byte number count register	TBCR	R/W	H'FE50 0014	H'1E50 0014	8	Pck
Mode register	MODER	R/W	H'FE50 0016	H'1E50 0016	8	Pck
Command type register	CMDTYR	R/W	H'FE50 0018	H'1E50 0018	8	Pck
Response type register	RSPTYR	R/W	H'FE50 0019	H'1E50 0019	8	Pck
Response register 0	RSPR0	R/W	H'FE50 0020	H'1E50 0020	8	Pck
Response register 1	RSPR1	R/W	H'FE50 0021	H'1E50 0021	8	Pck
Response register 2	RSPR2	R/W	H'FE50 0022	H'1E50 0022	8	Pck
Response register 3	RSPR3	R/W	H'FE50 0023	H'1E50 0023	8	Pck
Response register 4	RSPR4	R/W	H'FE50 0024	H'1E50 0024	8	Pck
Response register 5	RSPR5	R/W	H'FE50 0025	H'1E50 0025	8	Pck
Response register 6	RSPR6	R/W	H'FE50 0026	H'1E50 0026	8	Pck
Response register 7	RSPR7	R/W	H'FE50 0027	H'1E50 0027	8	Pck

Register Name	Abbrev.	R/W	P4 Address	Area 7 Address	Size	Sync Clock
Response register 8	RSPR8	R/W	H'FE50 0028	H'1E50 0028	8	Pck
Response register 9	RSPR9	R/W	H'FE50 0029	H'1E50 0029	8	Pck
Response register 10	RSPR10	R/W	H'FE50 002A	H'1E50 002A	8	Pck
Response register 11	RSPR11	R/W	H'FE50 002B	H'1E50 002B	8	Pck
Response register 12	RSPR12	R/W	H'FE50 002C	H'1E50 002C	8	Pck
Response register 13	RSPR13	R/W	H'FE50 002D	H'1E50 002D	8	Pck
Response register 14	RSPR14	R/W	H'FE50 002E	H'1E50 002E	8	Pck
Response register 15	RSPR15	R/W	H'FE50 002F	H'1E50 002F	8	Pck
Response register 16	RSPR16	R/W	H'FE50 0030	H'1E50 0030	8	Pck
Data timeout register	DTOUTR	R/W	H'FE50 0032	H'1E50 0032	16	Pck
Data register	DR	R/W	H'FE50 0040	H'1E50 0040	16	Pck
FIFO pointer clear register	FIFOCLR	W	H'FE50 0042	H'1E50 0042	8	Pck
DMA control register	DMACR	R/W	H'FE50 0044	H'1E50 0044	8	Pck
Interrupt control register 2	INTCR2	R/W	H'FE50 0046	H'1E50 0046	8	Pck
Interrupt status register 2	INTSTR2	R/W	H'FE50 0048	H'1E50 0048	8	Pck
Receive data timing select register	RDTIMSEL	R/W	H'FE50 004A	H'1E50 004A	8	Pck

Table 26.2 Register Configuration (2)

Register Name	Abbrev.	Power-on Reset by $\overline{\text{RESET}}$ Pin/WDT/ H-UDI	Manual Reset by $\overline{\text{RESET}}$ Pin/WDT/ Multiple Exception	Standby: Sleep by Sleep Instruction/ Deep Sleep	Standby: by Hardware	Standby: by Software/ Each Module
Command register 0	CMDR0	H'00	H'00	Retained	*	Retained
Command register 1	CMDR1	H'00	H'00	Retained		Retained
Command register 2	CMDR2	H'00	H'00	Retained		Retained
Command register 3	CMDR3	H'00	H'00	Retained		Retained
Command register 4	CMDR4	H'00	H'00	Retained		Retained
Command register 5	CMDR5	H'00	H'00	Retained		Retained
Command start register	CMDSTRT	H'00	H'00	Retained		Retained
Operation control register	OPCR	H'00	H'00	Retained		Retained
Card status register	CSTR	H'0x	H'0x	Retained		Retained

Register Name	Abbrev.	Reset by RESET Pin/WDT/ H-UDI	Reset by RESET Pin/WDT/ Multiple Exception	Sleep by Sleep Instruction/ Deep Sleep		Reset by Software/ by Hardware Each Module
Interrupt control register 0	INTCR0	H'00	H'00	Retained	*	Retained
Interrupt control register 1	INTCR1	H'00	H'00	Retained		Retained
Interrupt status register 0	INTSTR0	H'00	H'00	Retained		Retained
Interrupt status register 1	INTSTR1	H'00	H'00	Retained		Retained
Transfer clock control register	CLKON	H'00	H'00	Retained		Retained
Command timeout control register	CTOCR	H'00	H'00	Retained		Retained
Transfer byte number count register	TBCR	H'00	H'00	Retained		Retained
Mode register	MODER	H'00	H'00	Retained		Retained
Command type register	CMDTYR	H'00	H'00	Retained		Retained
Response type register	RSPTYR	H'00	H'00	Retained		Retained
Response register 0	RSPR0	H'00	H'00	Retained		Retained
Response register 1	RSPR1	H'00	H'00	Retained		Retained
Response register 2	RSPR2	H'00	H'00	Retained		Retained
Response register 3	RSPR3	H'00	H'00	Retained		Retained
Response register 4	RSPR4	H'00	H'00	Retained		Retained
Response register 5	RSPR5	H'00	H'00	Retained		Retained
Response register 6	RSPR6	H'00	H'00	Retained		Retained
Response register 7	RSPR7	H'00	H'00	Retained		Retained
Response register 8	RSPR8	H'00	H'00	Retained		Retained
Response register 9	RSPR9	H'00	H'00	Retained		Retained
Response register 10	RSPR10	H'00	H'00	Retained		Retained
Response register 11	RSPR11	H'00	H'00	Retained		Retained
Response register 12	RSPR12	H'00	H'00	Retained		Retained
Response register 13	RSPR13	H'00	H'00	Retained		Retained
Response register 14	RSPR14	H'00	H'00	Retained		Retained
Response register 15	RSPR15	H'00	H'00	Retained		Retained
Response register 16	RSPR16	H'00	H'00	Retained		Retained
Data timeout register	DTOUTR	H'FFFF	H'FFFF	Retained		Retained
Data register	DR	H'xxxx	H'xxxx	Retained		Retained
FIFO pointer clear register	FIFOCLR	H'00	H'00	Retained		Retained

Register Name	Abbrev.	Reset by RESET Pin/WDT/ H-UDI	Reset by RESET Pin/WDT/ Multiple Exception	Sleep by Sleep Instruction/ by Deep Sleep	Hardware	Software/ Each Module
DMA control register	DMACR	H'00	H'00	Retained	*	Retained
Interrupt control register 2	INTCR2	H'00	H'00	Retained		Retained
Interrupt status register 2	INTSTR2	H'0x	H'0x	Retained		Retained
Receive data timing select register	RDTIMSEL	H'00	H'00	Retained		Retained

Note: * After exiting hardware standby mode, this LSI enters the power-on reset state caused by the RESET pin.

26.3.1 Mode Register (MODER)

MODER is an 8-bit readable/writable register that specifies the MMCIF operating mode. The following sequence should be repeated when the MMCIF uses the multimedia card: Send a command, wait for the end of the command sequence and the end of the data busy state, and send a next command.

The series of operations from command sending, command response reception, data transmission/reception, and data response reception is called as the command sequence. The command sequence starts from sending a command by setting the START bit in CMDSTRT to 1, and ends when all necessary data transmission/reception and response reception have been completed. The multimedia card supports the data busy state such that only the specific command is accepted to write/erase data to/from the flash memory in the card during command sequence execution and after command sequence execution has ended. The data busy state is indicated by a low level output from the card side to the MCDAT pin.

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

Bit	Bit Name	Initial Value	R/W	Description
7 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	MODE	0	R/W	Operating Mode Specifies the MMCIF operating mode. 0: Operates in MMC mode 1: Setting prohibited

CMDTYR is an 8-bit readable/writable register that specifies the command format in conjunction with RSPTYR. Bits TY1 and TY0 specify the existence and direction of transfer data, and bits TY4 to TY2 specify the additional settings. All of bits TY4 to TY2 should be cleared to 0 or only one of them should be set to 1. Bits TY4 to TY2 can only be set to 1 if the corresponding settings in TY1 and TY0 allow that setting.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	TY4	TY3	TY2	TY1	TY0
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

Bit	Bit Name	Initial Value	R/W	Description
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	TY4	0	R/W	Type 4 Set this bit to 1 when specifying the CMD12 command. Bits TY1 and TY0 should be set to 00.
3	TY3	0	R/W	Type 3 Set this bit to 1 when specifying stream transfer. Bits TY1 and TY0 should be set to 01 or 10. The command sequence of the stream transfer specified by this bit ends when it is aborted by the CMD12 command.
2	TY2	0	R/W	Type 2 Set this bit to 1 when specifying multiblock transfer. Bits TY1 and TY0 should be set to 01 or 10. The command sequence of the multiblock transfer specified by this bit ends when it is aborted by the CMD12 command.
1 0	TY1 TY0	0 0	R/W R/W	Types 1 and 0 These bits specify the existence and direction of transfer data. 00: A command without data transfer 01: A command with read data reception 10: A command with write data transmission 11: Setting prohibited

RSPTYR is an 8-bit readable/writable register that specifies command format in conjunction with CMDTYR. Bits RTY2 to RTY0 specify the number of response bytes, and bits RTY5 and RTY4 specify the additional settings.

Bit:	7	6	5	4	3	2	1	0
	-	-	RTY5	RTY4	-	RTY2	RTY1	RTY0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5	RTY5	0	R/W	Response Type 5 Sets data busy status from the MMC card. 0: A command without data busy 1: A command with data busy
4	RTY4	0	R/W	Response Type 4 Specifies that the command response CRC is checked through CRC7. Bits RTY2 to RTY0 should be set to 100 or 101. 0: Does not check CRC through CRC7 1: Checks CRC through CRC7
3	—	0	R	Reserved These bits are always read as 0. The write value should always be 0.
2 1 0	RTY2 RTY1 RTY0	0 0 0	R/W R/W R/W	Response Types 2 to 0 These bits specify the number of command response bytes. 000: A command needs no command response. 001: Setting prohibited 010: Setting prohibited 011: Setting prohibited 100: A command needs 6-byte command responses. Specified by R1, R1b, R3, R4, and R5 responses. 101: A command needs a 17-byte command response. Specified by R2 response. 110: Setting prohibited 111: Setting prohibited

MultiMediaCard System Specification Version 2.11 and the settings of the CMDTYR and RSPTYR registers.

Table 26.3 Correspondence between Commands and Settings of CMDTYR and RSPTYR

CMD INDEX	Abbreviation	resp	CMDTYR				RSPTYR		
			1 and 0	2	3	4	2 to 0	4	5
CMD0	GO_IDLE_STATE	—	00				000		
CMD1	SEND_OP_COND	R3	00				100		
CMD2	ALL_SEND_CID	R2	00				101		
CMD3	SET_RELATIVE_ADDR	R1	00				100	*	
CMD4	SET_DSR	—	00				000		
CMD7	SELECT/DESELECT_CARD	R1b	00				100	*	1
CMD9	SEND_CSD	R2	00				101		
CMD10	SEND_CID	R2	00				101		
CMD11	READ_DAT_UNTIL_STOP	R1	01		1		100	*	
CMD12	STOP_TRANSMISSION	R1b	00			1	100	*	1
CMD13	SEND_STATUS	R1	00				100	*	
CMD15	GO_INACTIVE_STATE	—	00				000		
CMD16	SET_BLOCKLEN	R1	00				100	*	
CMD17	READ_SINGLE_BLOCK	R1	01				100	*	
CMD18	READ_MULTIPLE_BLOCK	R1	01	1			100	*	
CMD20	WRITE_DAT_UNTIL_STOP	R1	10		1		100	*	
CMD24	WRITE_BLOCK	R1	10				100	*	
CMD25	WRITE_MULTIPLE_BLOCK	R1	10	1			100	*	
CMD26	PROGRAM_CID	R1	10				100	*	
CMD27	PROGRAM_CSD	R1	10				100	*	
CMD28	SET_WRITE_PROT	R1b	00				100	*	1
CMD29	CLR_WRITE_PROT	R1b	00				100	*	1
CMD30	SEND_WRITE_PROT	R1	01				100	*	
CMD32	TAG_SECTOR_START	R1	00				100	*	
CMD33	TAG_SECTOR_END	R1	00				100	*	
CMD34	UNTAG_SECTOR	R1	00				100	*	

INDEX	Abbreviation	resp	1 and 0	2	3	4	2 to 0	4	5
CMD35	TAG_ERASE_GROUP_ START	R1	00				100	*	
CMD36	TAG_ERASE_GROUP_ END	R1	00				100	*	
CMD37	UNTAG_ERASE_GROUP	R1	00				100	*	
CMD38	ERASE	R1b	00				100	*	1
CMD39	FAST_IO	R4	00				100	*	
CMD40	GO_IRQ_STATE	R5	00				100	*	
CMD42	LOCK_UNLOCK	R1b	10				100	*	1
CMD55	APP_CMD	R1	00				100	*	
CMD56	GEN_CMD	R1b	10 or 11				100	*	1

Notes: * Command response CRC can be checked.

1. A blank means value 0.

TBCR is an 8-bit readable/writable register that specifies the number of bytes to be transferred (block size) for each single block transfer command. TBCR specifies the number of data block bytes not including the start bit, end bit, and CRC.

The multiblock transfer command corresponds to the number of bytes of each data block. This setting is ignored by the stream transfer command.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	C3	C2	C1	C0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
3	CS3	0	R/W	Transfer Data Block Size Four or more bytes should be set before executing a command with data transfer. 0000: 1 byte (for forced erase) 0001: Setting prohibited 0010: 4 bytes 0011: 8 bytes 0100: 16 bytes 0101: 32 bytes 0110: 64 bytes 0111: 128 bytes 1000: 256 bytes 1001: 512 bytes 1010: 1024 bytes 1011: 2048 bytes 1100 to 1111: Setting prohibited
2	CS2	0	R/W	
1	CS1	0	R/W	
0	CS0	0	R/W	

The CMDR registers are six 8-bit registers. A command is written to CMDR as shown in table 26.4, and the command is transmitted when the START bit in CMDSTRT is set to 1.

Table 26.4 CMDR Configuration

Register	Contents	Operation
CMDR0	Start bit, Host bit, and command index	Write command indexes. Clear the Start bit to 0 and set the Host bit to 1.
CMDR1 to CMDR4	Command argument	Write command arguments.
CMDR5	CRC and End bit	Setting of CRC is unnecessary (automatic calculation). End bit is fixed to 1 and its setting is unnecessary.

- CMDR0

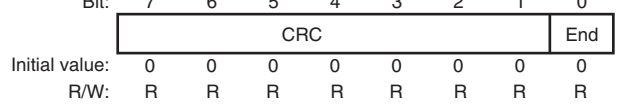
Bit:	7	6	5	4	3	2	1	0
	Start	Host	INDEX					
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	Start	0	R/W	Start bit (This bit should be cleared to 0)
6	Host	0	R/W	Transmission bit (This bit should be set to 1)
5 to 0	INDEX	All 0	R/W	Command indexes

- CMDR1 to CMDR4

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	Bit Name	Initial Value	R/W	Description
7 to 0	—	All 0	R/W	Command arguments See specifications for the MMC card.



Bit	Bit Name	Initial Value	R/W	Description
7 to 1	CRC	All 0	R	These bits are always read as 0. The write value should always be 0.
0	End	0	R	This bit is always read as 0. The write value should always be 0.

26.3.6 Response Registers 0 to 16 (RSPR0 to RSPR16)

RSPR0 to RSPR16 are command response registers, which are seventeen 8-bit registers.

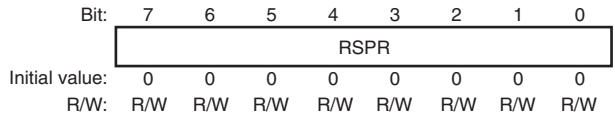
The number of command response bytes differs according to the command. The number of command response bytes can be specified by RSPTYR in the MMCIF. The command response is shifted-in from bit 0 in RSPR16, and shifted to the number of command response bytes × 8 bits. Table 26.5 summarizes the correspondence between the number of command response bytes and valid RSPR register.

	MMC Mode Response	
RSPR registers	6 bytes (R1, R1b, R3, R4, R5)	17 bytes (R2)
RSPR0	—	1st byte
RSPR1	—	2nd byte
RSPR2	—	3rd byte
RSPR3	—	4th byte
RSPR4	—	5th byte
RSPR5	—	6th byte
RSPR6	—	7th byte
RSPR7	—	8th byte
RSPR8	—	9th byte
RSPR9	—	10th byte
RSPR10	—	11th byte
RSPR11	1st byte	12th byte
RSPR12	2nd byte	13th byte
RSPR13	3rd byte	14th byte
RSPR14	4th byte	15th byte
RSPR15	5th byte	16th byte
RSPR16	6th byte	17th byte

RSPR0 to RSPR16 are simple shift registers. A command response that has been shifted in is not automatically cleared, and it is continuously shifted until it is shifted out from bit 7 in RSPR0. To clear unnecessary bytes to H'00, write an arbitrary value to each RSPR.

Clearing an RSPR is completed two transfer clock cycles after an arbitrary value is written to the RSPR.

- RSPR0 to RSPR16



7 to 0	RSPR	All 0	R/W	These bits are cleared to H'00 by writing an arbitrary value. RSPR0 to RSPR16 comprise a continuous 17-byte shift register.

26.3.7 Command Start Register (CMDSTRT)

CMDSTRT is an 8-bit readable/writable register that triggers the start of command transmission, representing the start of a command sequence. The following operations should have been completed before the command sequence starts.

- Analysis of prior command response, clearing the command response register write if necessary
- Analysis/transfer of receive data of prior command if necessary
- Preparation of transmit data of the next command if necessary
- Setting of CMDTYR, RSPTYR, and TBCR
- Setting of CMDR0 to CMDR4
  The CMDR0 to CMDR4, CMDTYR, RSPTYR, and TBCR registers should not be changed until command transmission has ended (the CWRE flag in CSTR has been set to 1).

Command sequences are controlled by the sequencers in both the MMCIF side and the MMC card side. Normally, these operate synchronously. However, if an error occurs or a command is aborted, these may become temporarily unsynchronized. Be careful when setting the CMDOFF bit in OPCR, issuing the CMD12 command, or processing an error in MMC mode. A new command sequence should be started only after the end of the command sequence on both the MMCIF and card sides is confirmed.

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

Bit	Bit Name	Initial Value	R/W	Description
7 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	START	0	R/W	Starts command transmission when 1 is written. This bit is automatically cleared. When 0 is written to this bit, its previous value is retained.

OPCR is an 8-bit readable/writable register that aborts command operation, and suspends or continues data transfer.

Bit:	7	6	5	4	3	2	1	0
	CMD OFF	-	RD_ CONTI	DATAEN	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R	R/W	R/W	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7	CMDOFF	0	R/W	Command Off Aborts all command operations (MMCIF command sequence) when 1 is written after a command is transmitted. This bit is then cleared automatically. Write enabled period: From command transmission completion to command sequence end Write of 0: Operation is not affected. Write of 1: Command sequence is forcibly aborted.* * The transfer clock output resumes if the transfer clock has been halted during the command sequence.
6	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
5	RD_ CONTI	0	R/W	Read Continue Transfer clock output and read data reception are resumed when 1 is written while the transfer clock has been halted by FIFO full or termination of block reading in multiblock read. This bit is cleared automatically when 1 is written and reading is resumed Write enabled period: While MCCLK for read data reception is halted Write of 0: Operation is not affected. Write of 1: Resumes MCCLK output and read data reception.

4	DATAEN	0	R/W	Data Enable Starts write data transmission by a command with write data. This bit is cleared automatically when 1 is written. Resumes transfer clock output and write data transmission when the transfer clock has been halted by FIFO empty or termination of one block writing in multiblock write. Write enabled period: (1) after receiving a response to a command with write data, (2) while transfer clock is halted by FIFO empty, (3) when one block writing in multiblock write is terminated Write of 0: Operation is not affected. Write of 1: Starts or resumes transfer clock output and write data transmission.
3 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Some states of the multimedia card cause command sequence on the multimedia card side to stop. Table 26.6 shows the card states in which a command sequence is halted. In this case, the command sequence should also be aborted by setting the CMDOFF bit to 1 on the MMCIF side.

Table 26.6 Card States in which Command Sequence is Halted

Card Operating Mode		Error Status
MMC mode	Command response	When the error detection bit in the card status (32 bits) in the command response data transmitted by the card is set.
	Data status	When the CRCERI bit is set due to an error in the CRC status to be transmitted from the card is set while block data is transmitted to the card.

In write data transmission, the contents of the command response and data response should be analyzed, and then transmission should be triggered. In addition, the transfer clock (MCCLK) output should be temporarily halted by FIFO full/empty, and it should be resumed when the preparation has been completed.

issuing the CMD12 command, and the transfer clock output should be resumed. To continue to the next block, the RD_CONTI and DATAEN bits should be set to 1. To issue the CMD12 command, the CMDOFF bit should be set to 1 to abort the command sequence on the MMCIF side.

Note: The FIFO full interrupt source must be cleared (FIFO data read) only after five or more transfer clock cycles have been passed since the interrupt occurred.

26.3.9 Command Timeout Control Register (CTOCR)

CTOCR specifies the period to generate a timeout for the command response.

The counter (CTOUTC), to which the peripheral bus does not have access, counts the transfer clock to monitor the command timeout. The initial value of CTOUTC is 0, and CTOUTC starts counting the transfer clock from the start of command transmission. CTOUTC is cleared and stops counting the transfer clock when command response reception has been completed, or when the command sequence has been aborted by setting the CMDOFF bit to 1.

When the command response cannot be received, CTOUTC continues counting the transfer clock, and enters the command timeout error state when the number of transfer clock cycles reaches the number specified in CTOCR. When the CTERIE bit in INTCR1 is set to 1, the CTERI flag in INTSTR1 is set. As CTOUTC continues counting transfer clock, the CTERI flag setting condition is repeatedly generated. To perform command timeout error handling, the command sequence should be aborted by setting the CMDOFF bit to 1, and then the CTERI flag should be cleared to prevent extra-interrupt generation.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	CTSEL1	CTSEL0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

1	CTSEL1	0	R/W	Command Timeout Select
0	CTSEL0	0	R/W	00: 128 transfer clock cycles from command transmission completion to response reception completion 01: 256 transfer clock cycles from command transmission completion to response reception completion 10: Setting prohibited 11: Setting prohibited

Note: If R2 response (17-byte command response) is requested and CTSEL0 is cleared to 0, a timeout is generated during response reception. Therefore, set CTSEL0 to 1.

26.3.10 Data Timeout Register (DTOUTR)

DTOUTR specifies the period to generate a data timeout. The 16-bit counter (DTOUTC) and a prescaler, to which the peripheral bus does not have access, count the peripheral clock to monitor the data timeout. The prescaler always counts the peripheral clock, and outputs a count pulse for every 10,000 peripheral clock cycles. The initial value of DTOUTC is 0, and DTOUTC starts counting the prescaler output from the start of the command sequence. DTOUTC is cleared when the command sequence has ended, or when the command sequence has been aborted by setting the CMDOFF bit to 1, after which the DTOUTC stops counting the prescaler output.

When the command sequence does not end, DTOUTC continues counting the prescaler output, and enters the data timeout error states when the number of prescaler outputs reaches the number specified in DTOUTR. When the DTERIE bit in INTCR1 is set to 1, the DTERI flag in INTSTR1 is set. As DTOUTC continues counting prescaler output, the DTERI flag setting condition is repeatedly generated. To perform data timeout error handling, the command sequence should be aborted by setting the CMDOFF bit to 1, and then the DTERI flag should be cleared to prevent extra-interrupt generation.

For a command with data busy status, data timeout cannot be monitored since the command sequence is terminated before entering the data busy state. Timeout in the data busy state should be monitored by firmware.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DTOUTR															
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

15 to 0	DTOUTR	All 1	R/W	Data Timeout Time/10,000 Data timeout time: Peripheral clock cycle x DTOUTR setting value x 10,000.

26.3.11 Card Status Register (CSTR)

CSTR indicates the MMCIF status during command sequence execution.

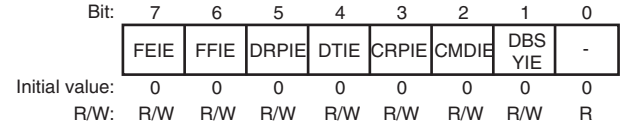
Bit:	7	6	5	4	3	2	1	0
	BUSY	FIFO_ FULL	FIFO_ EMPTY	CWRE	DTBUSY	DTBUSY _TU	-	REQ
Initial value:	0	0	0	0	0	-	0	0
R/W:	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7	BUSY	0	R	Command Busy Indicates command execution status. When the CMDOFF bit in OPCR is set to 1, this bit is cleared to 0 because the MMCIF command sequence is aborted. 0: Idle state waiting for a command, or data busy state 1: Command sequence execution in progress
6	FIFO_ FULL	0	R	FIFO Full This bit is set to 1 when the FIFO becomes full while data is being received from the card, and cleared to 0 when RD_CONTI is set to 1 or the command sequence is completed. Indicates whether the FIFO is empty or not. 0: The FIFO is empty. 1: The FIFO is full.
5	FIFO_ EMPTY	0	R	FIFO Empty This bit is set to 1 when the FIFO becomes empty while data is being sent to the card, and cleared to 0 when DATA_EN is set to 1 or the command sequence is completed. Indicates whether the FIFO holds data or not. 0: The FIFO includes data. 1: The FIFO is empty.

4	CWRE	0	R	Command Register Write Enable Indicates whether the CMDR command is being transmitted or has been transmitted. 0: The CMDR command has been transmitted, or the START bit in CMDSTRT has not been set yet, so the new command can be written. 1: The CMDR command is waiting for transmission or is being transmitted. If a new command is written, a malfunction will result.
3	DTBUSY	0	R	Data Busy Indicates command execution status. Indicates that the card is in the busy state after the command sequence of a command without data transfer which includes the busy state in the response, or a command with write data has been ended. 0: Idle state waiting for a command, or command sequence execution in progress 1: Card is in the data busy state after command sequence termination.
2	DTBUSY_TU	—	R	Data Busy Pin Status Indicates the MCDAT pin level. By reading this bit, the MCDAT level can be monitored. 0: A low level is input to the MCDAT pin. 1: A high level is input to the MCDAT pin.
1	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
0	REQ	0	R	Interrupt Request Indicates whether an interrupt is requested or not. An interrupt request is the logical OR of the INTSTR0 and INTSTR1 flags. The INTSTR0 and INTSTR1 flags set is controlled by the enable bits in INTCR0 and INTCR1. 0: No interrupt requested. 1: Interrupt requested.

The INTCR registers enable or disable interrupts.

- INTCR0



Bit:	7	6	5	4	3	2	1	0
	FEIE	FFIE	DRPIE	DTIE	CRPIE	CMDIE	DBS YIE	-
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	Bit Name	Initial Value	R/W	Description
7	FEIE	0	R/W	FIFO Empty Interrupt Enable 0: Disables FIFO empty interrupt (disables FEI flag setting). 1. Enables FIFO empty interrupt (enables FEI flag setting).
6	FFIE	0	R/W	FIFO Full Interrupt Enable 0: Disables FIFO full interrupt (disables FFI flag setting). 1: Enables FIFO full interrupt (enables FFI flag setting).
5	DRPIE	0	R/W	Data Response Interrupt Enable 0: Disables data response interrupt (disables DPRI flag setting). 1: Enables data response interrupt (enables DPRI flag setting).
4	DTIE	0	R/W	Data Transfer End Interrupt Enable 0: Disables data transfer end interrupt (disables DTI flag setting). 1: Enables data transfer end interrupt (enables DTI flag setting).
3	CRPIE	0	R/W	Command Response Receive End Interrupt Enable 0: Disables command response receive end interrupt (disables CRPI flag setting). 1: Enables command response receive end interrupt (enables CRPI flag setting).
2	CMDIE	0	R/W	Command Transmit End Interrupt Enable 0: Disables command transmit end interrupt (disables CMDI flag setting). 1: Enables command transmit end interrupt (enables CMDI flag setting).

1	DBSYIE	0	R/W	Data Busy End Interrupt Enable 0: Disables data busy end interrupt (disables DBSYI flag setting). 1: Enables data busy end interrupt (enables DBSYI flag setting).
0	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

- INTCR1

Bit:	7	6	5	4	3	2	1	0
	INTRQ2E	INTRQ1E	INTRQ0E	-	-	CRCERIE	DTERIE	CTERIE
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	INTRQ2E	0	R/W	MMCI2 Interrupt Enable 0: Disables MMCI2 interrupt. 1: Enables MMCI2 interrupt.
6	INTRQ1E	0	R/W	MMCI1 Interrupt Enable 0: Disables MMCI1 interrupt. 1: Enables MMCI1 interrupt.
5	INTRQ0E	0	R/W	MMCI0 Interrupt Enable 0: Disables MMCI0 interrupt. 1: Enables MMCI0 interrupt.
4, 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2	CRCERIE	0	R/W	CRC Error Interrupt Enable 0: Disables CRC error interrupt (disables CRCERI flag setting). 1: Enables CRC error interrupt (enables CRCERI flag setting).

1	DTERIE	0	R/W	Data Timeout Error Interrupt Enable 0: Disables data timeout error interrupt (disables DTERI flag setting). 1: Enables data timeout error interrupt (enables DTERI flag setting).
0	CTERIE	0	R/W	Command Timeout Error Interrupt Enable 0: Disables command timeout error interrupt (disables CTERI flag setting). 1: Enables command timeout error interrupt (enables CTERI flag setting).

- INTCR2

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

Bit	Bit Name	Initial Value	R/W	Description
7 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	FRDYIE	0	R/W	FIFO Ready Interrupt Enable 0: Disables FIFO ready interrupt (disables flag setting). 1: Enables FIFO ready interrupt (enables flag setting).

26.3.13 Interrupt Status Registers 0 to 2 (INTSTR0 to INTSTR2)

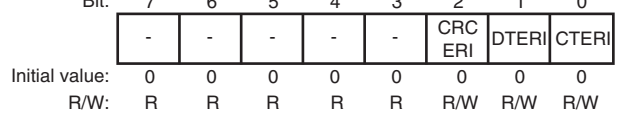
The INTSTR registers enable or disable MMCIF interrupts MMCI3 to MMCI0.

- INTSTR0

Bit:	7	6	5	4	3	2	1	0
	FEI	FFI	DRPI	DTI	CRPI	CMDI	DBSYI	-
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

7	FEI	0	R/W	FIFO Empty Interrupt 0: No interrupt [Clearing condition] Write 0 after reading FEI = 1. 1: Interrupt requested [Setting condition] When FIFO becomes empty while FEIE = 1 and data is being transmitted (when the FIFO_EMPTY bit in CSTR is set)	MMCI0
6	FFI	0	R/W	FIFO Full Interrupt 0: No interrupt [Clearing condition] Write 0 after reading FFI = 1. 1: Interrupt requested [Setting condition] When FIFO becomes full while FFIE = 1 and data is being received (when the FIFO_FULL bit in CSTR is set)	MMCI0
5	DRPI	0	R/W	Data Response Interrupt 0: No interrupt [Clearing condition] Write 0 after reading DRPI = 1. 1: Interrupt requested [Setting condition] When the CRC status is received while DRPIE = 1.	MMCI1
4	DTI	0	R/W	Data Transfer End Interrupt 0: No interrupt [Clearing condition] Write 0 after reading DTI = 1. 1: Interrupt requested [Setting condition] When the number of bytes of data transfer specified in TBCR ends while DTIE = 1.	MMCI1

3	CRPI	0	R/W	Command Response Receive End Interrupt 0: No interrupt [Clearing condition] Write 0 after reading CRPI = 1. 1: Interrupt requested [Setting condition] When command response reception ends while CRPIE = 1.	MMCI1
2	CMDI	0	R/W	Command Transmit End Interrupt 0: No interrupt [Clearing condition] Write 0 after reading CMDI = 1. 1: Interrupt requested [Setting condition] When command transmission ends while CMDIE = 1. (When the CWRE bit in CSTR is cleared.)	MMCI1
1	DBSYI	0	R/W	Data Busy End Interrupt 0: No interrupt [Clearing condition] Write 0 after reading DBSYI = 1. 1: Interrupt requested [Setting condition] When data busy state is canceled while DBSYIE = 1. (When the DTBUSY bit in CSTR is cleared.)	MMCI1
0	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.	—



Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	-	CRCERI	DTERI	CTERI
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description	Interrupt outputs
7 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.	—
2	CRCERI	0	R/W	CRC Error Interrupt 0: No interrupt [Clearing condition] Write 0 after reading CRCERI = 1. 1: Interrupt requested [Setting condition] When a CRC error for command response or receive data or a CRC status error for transmit data response is detected while CRCERIE = 1. For the command response, CRC is checked when the RTY4 in RSPTYR is enabled.	MMCI2
1	DTERI	0	R/W	Data Timeout Error Interrupt 0: No interrupt [Clearing condition] Write 0 after reading DTERI = 1. 1: Interrupt requested [Setting condition] When a data timeout error specified in DTOUTR occurs while DTERIE = 1.	MMCI2
0	CTERI	0	R/W	Command Timeout Error Interrupt 0: No interrupt [Clearing condition] Write 0 after reading CTERI = 1. 1: Interrupt requested [Setting condition] When a command timeout error specified in TOCR occurs while CTERIE = 1.	MMCI2

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	FRDY_TU	FRDYI
Initial value:	0	0	0	0	0	0	-	0
R/W:	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description	Interrupt outputs
7 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.	—
1	FRDY_TU	—	R	FIFO Ready Flag Regardless of set values of DMAEN and FRDYIE, this bit is read as 0 when FIFO data amount matches the condition set in DMACR[2:0], and otherwise, read as 1.	—
0	FRDYI	0	R/W	FIFO Ready Interrupt 0: No interrupt [Clearing condition] Write 0 after reading FRDYI = 1. 1: Interrupt requested [Setting condition] When remained FIFO data does not match the assert condition set in DMACR while DMAEN = 1 and FRDYIE = 1. Note: FRDYI will be set on the setting condition after clearing. To clear it, disable the flag setting by FRDYIE in INTCR2.	MMCI3

CLKON controls the transfer clock frequency and clock ON/OFF.

Bits CSEL2 to CSEL0 must be set to 100 for the peripheral clock to be 20-MHz in order to achieve a 20-Mbps transfer clock in the MMCIF. At this time, bits CSEL2 to CSEL0 should be set to 000 for the 200-kbps transfer clock in Card Identification Mode in MMC mode.

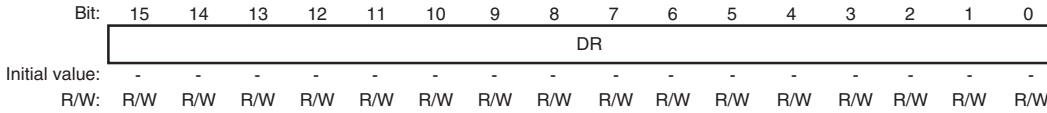
In a command sequence, do not perform clock ON/OFF or frequency modification.

Bit:	7	6	5	4	3	2	1	0
	CLKON	-	-	-	-	CSEL2	CSEL1	CSEL0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	CLKON	0	R/W	Clock On 0: Fixes the transfer clock output from the MCCLK pin to low level. 1: Outputs the transfer clock from the MCCLK pin.
6 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2 1 0	CSEL2 CSEL1 CSEL0	0 0 0	R/W R/W R/W	Transfer Clock Frequency Select 000: Uses the 1/100-divided peripheral clock as a transfer clock. 001: Uses the 1/8-divided peripheral clock as a transfer clock. 010: Uses the 1/4-divided peripheral clock as a transfer clock. 011: Uses the 1/2-divided peripheral clock as a transfer clock. 100: Use the peripheral clock as a transfer clock. 101 to 111: Setting prohibited

DR is a register for reading/writing FIFO data.

Word/byte access is enabled to addresses of this register.



Bit	Bit Name	Initial Value	R/W	Description
15 to 0	DR	—	R/W	Register for reading/writing FIFO data. Word/byte access is enabled. When DR is accessed in words, the upper and lower bytes are transmitted or received in that order. Word access and byte access can be done in random order. However, (DR address + 1) cannot be accessed in bytes.

The following shows examples of DR access.

When data is written to DR in the following steps 1 to 4, the transmit data is stored in the FIFO as shown in figure 26.2.

1. Write word data H'0123 to DR.
2. Write byte data H'45 to DR.
3. Write word data H'6789 to DR.
4. Write byte data H'AB to DR.

When the receive data is stored in the FIFO as shown in figure 26.2 (for example, after data is started to be received while the FIFO is empty and data is received in the order of H'01, H'23, ..., H'AB), data can be read from DR in the following steps 5 to 8.

5. Read byte data H'01 from DR.
6. Read word data H'2345 from DR.
7. Read byte data H'67 from DR.
8. Ready word data H'89AB from DR.

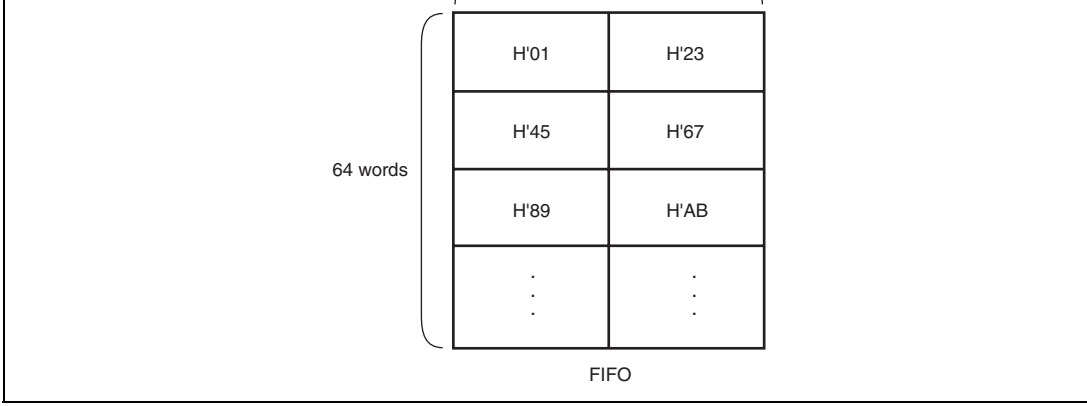


Figure 26.2 DR Access Example

26.3.16 FIFO Pointer Clear Register (FIFOCLR)

The FIFO write/read pointer is cleared by writing an arbitrary value to FIFOCLR.

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

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	FIFOCLR	H'00	W	The FIFO pointer is cleared by writing an arbitrary value to this register.

DMACR sets DMA request signal output. DMAEN enables or disables a DMA request signal. The DMA request signal is output based on a value that has been set to SET2 to SET0.

Bit:	7	6	5	4	3	2	1	0
	DMAEN	-	-	-	-	SET2	SET1	SET0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	DMAEN	0	R/W	DMA Enable 0: Disables output of DMA request signal. 1: Enables output of DMA request signal.
6 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2 1 0	SET2 SET1 SET0	0 0 0	R/W R/W R/W	DMA Request Signal Assert Condition Sets DMA request signal assert condition. 000: Not output 001: FIFO remained data is 1/4 or less of FIFO capacity. 010: FIFO remained data is 1/2 or less of FIFO capacity. 011: FIFO remained data is 3/4 or less of FIFO capacity. 100: FIFO remained data is 1 byte or more. 101: FIFO remained data is 1/4 or more of FIFO capacity. 110: FIFO remained data is 1/2 or more of FIFO capacity. 111: FIFO remained data is 3/4 or more of FIFO capacity.

RDTIMSEL selects the acquisition timing of receive response or receive data from the card. Consider the timing according to the multimedia card standards to set this register.

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

Bit	Bit Name	Initial Value	R/W	Description
7 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	RTSEL	0	R/W	Receive Data Timing Selection 0: Receives data at the falling of MCCLK 1: Receives data at the rising of MCCLK

26.4 Operation

The multimedia card is an external storage media that can be easily connected or disconnected. The MMCIF operates in MMC mode.

Insert a card and supply power to it. Then operate the MMCIF by applying the transfer clock after setting an appropriate transfer clock frequency.

Do not connect or disconnect the card during command sequence execution or in the data busy state.

26.4.1 Operations in MMC Mode

MMC mode is an operating mode in which the transfer clock is output from the MCCLK pin, command transmission/response receive occurs via the MCCMD pin, and data is transmitted/received via the MCDAT pin. In this mode the next command can be issued while data is being transmitted/received.

This feature is efficient for multiblock or stream transfer. In this case, the next command is the CMD12 command, which aborts the current command sequence.

In MMC mode, broadcast commands that simultaneously issue commands to multiple cards are supported. After information of the inserted cards is recognized by a broadcast command, a

deselected, and then various commands are issued to the selected card.

Commands in MMC mode are basically classified into three types: broadcast, relative address, and flash memory operation commands. The card can be operated by issuing these commands appropriately according to the card state.

(1) Operation of Broadcast Commands

The CMD0, CMD1, CMD2, and CMD4 are broadcast commands. These commands and the CMD3 command comprise a sequence assigning relative addresses to individual cards. In this sequence, the CMD output format is open drain, and the command response is wired-OR. During the issuance of this command sequence, the CSEL2 to CSEL0 bits in the CLKON register should be set to 000 and the transfer clock frequency should be set sufficiently slow.

- All cards are initialized to the idle state by the CMD0.
- The operation condition registers (OCR) of all cards are read via wired-OR and cards that cannot operate are deactivated by the CMD1.
  The cards that are not deactivated enter the ready state.
- The card identifications (CID) of all cards in the ready state are read via wired-OR by the CMD2.
  Each card compares it's CID and data on the MCCMD, and if they are different, the card aborts the CID output. Only one card in which the CID can be entirely output enters the acknowledge state. When the R2 response is necessary, set CTOCR to H'01.
- A relative address (RCA) is given to the card in the acknowledge state by the CMD3.
  The card to which the RCA is given enters the standby state.
- By repeating CMD2 and CMD3, RCAs are given to all cards in the ready state to make them enter the standby state.

(2) Operation of Relative Address Commands

The CMD7, CMD9, CMD10, CMD13, CMD15, CMD39, and CMD55 are relative address commands that address the card by RCA. The relative address commands are used to read card administration information and original information, and to change the specific card states.

The CMD7 sets one addressed card to the transfer state, and the other cards to the standby state. Only the card in the transfer state can execute flash-memory operation commands, other than broadcast or relative-address commands.

Some broadcast commands do not require a command response.

Figure 26.3 shows an example of the command sequence for commands that do not require a command response.

Figure 26.4 shows the operational flow for commands that do not require a command response.

- Make settings to issue the command.
- Set the START bit in CMDSTRT to 1 to start command transmission. MCCMD must be kept driven until the end bit output is completed.
- The end of the command sequence is detected by poling the BUSY flag in CSTR or by the command transmit end interrupt (CMDI).

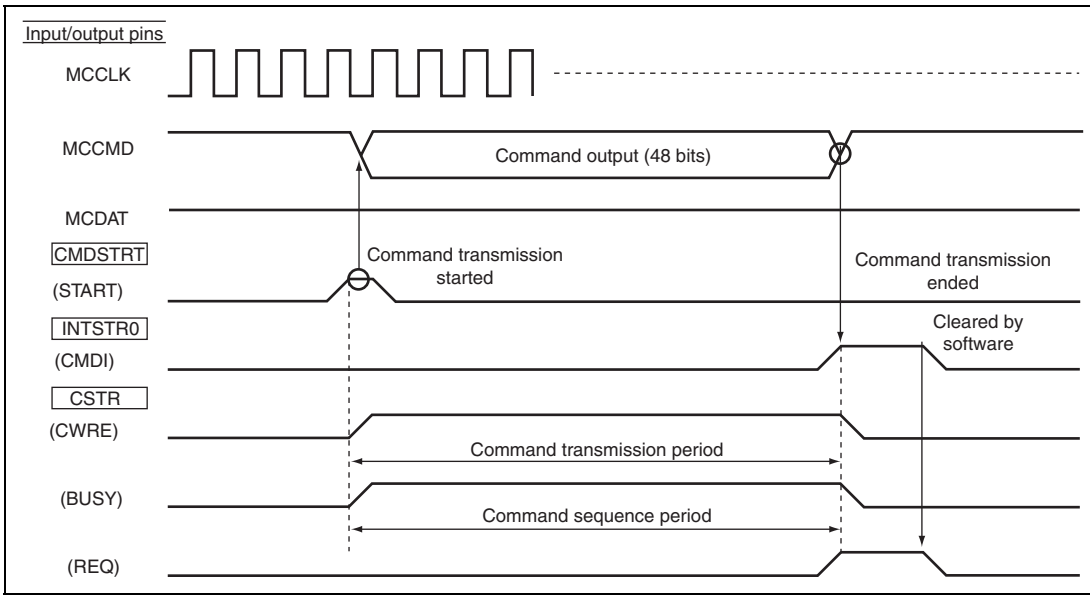


Figure 26.3 Example of Command Sequence for Commands Not Requiring Command Response

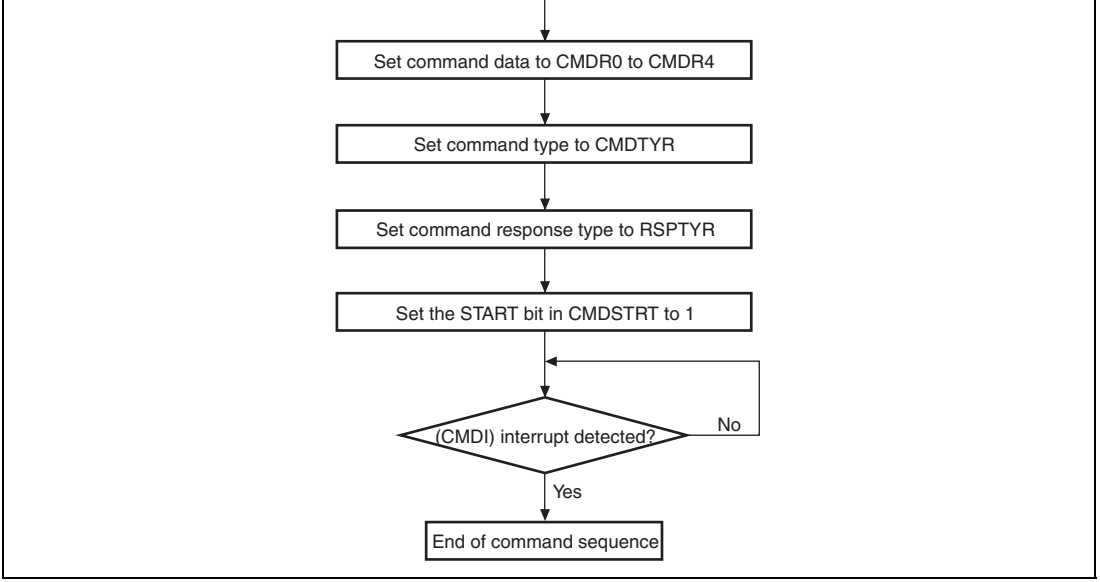


Figure 26.4 Example of Operational Flow for Commands Not Requiring Command Response

(4) Operation of Commands without Data Transfer

Broadcast, relative address, and flash memory operation commands include a number of commands that do not include data transfer. Such commands execute the desired data transfer using command arguments and command responses. For a command that is related to time-consuming processing such as flash memory write/erase, the card indicates the data busy state via the MCDAT.

Figures 26.5 and 26.6 show examples of the command sequence for commands without data transfer.

Figure 26.7 shows the operational flow for commands without data transfer.

- Make settings to issue the command.
- Set the START bit in CMDSTRT to 1 to start command transmission. MCCMD must be kept driven until the end bit output is completed.
  Command transmission completion can be confirmed by the command transmit end interrupt (CMDI).

If the card returns no command response, the command response is detected by the command timeout error (CTERI).

- The end of the command sequence is detected by poling the BUSY flag in CSTR or by the command response receive end interrupt (CRPI).
- The end of data busy state is detected by the data busy end interrupt (DBSYI).

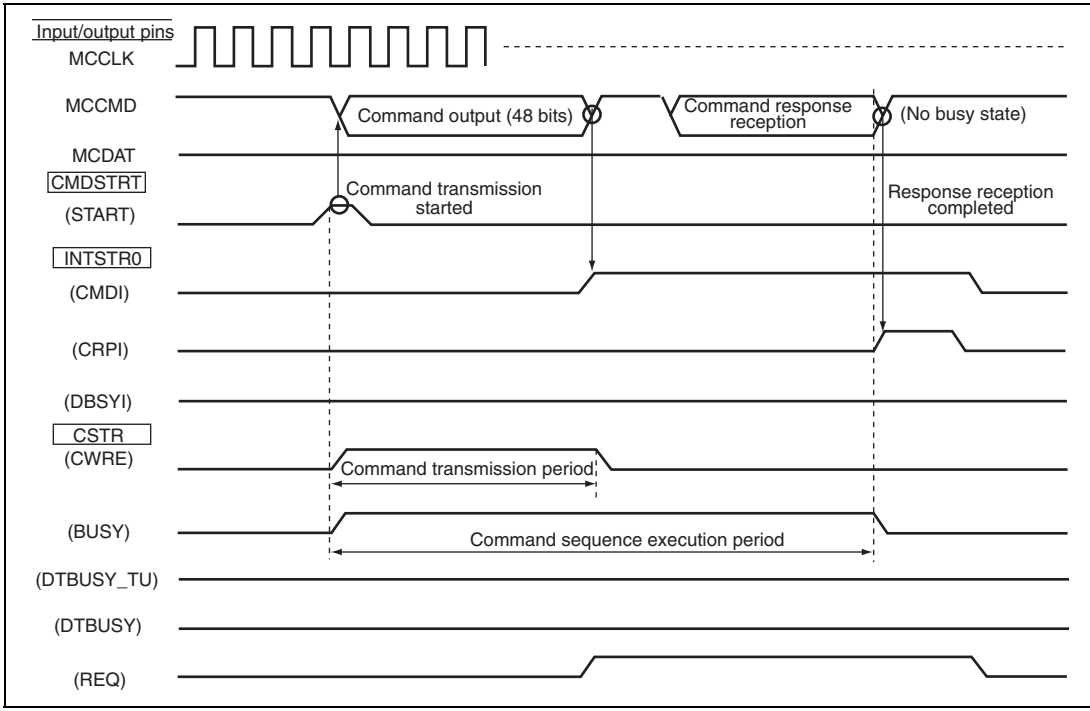


Figure 26.5 Example of Command Sequence for Commands without Data Transfer (No Data Busy State)

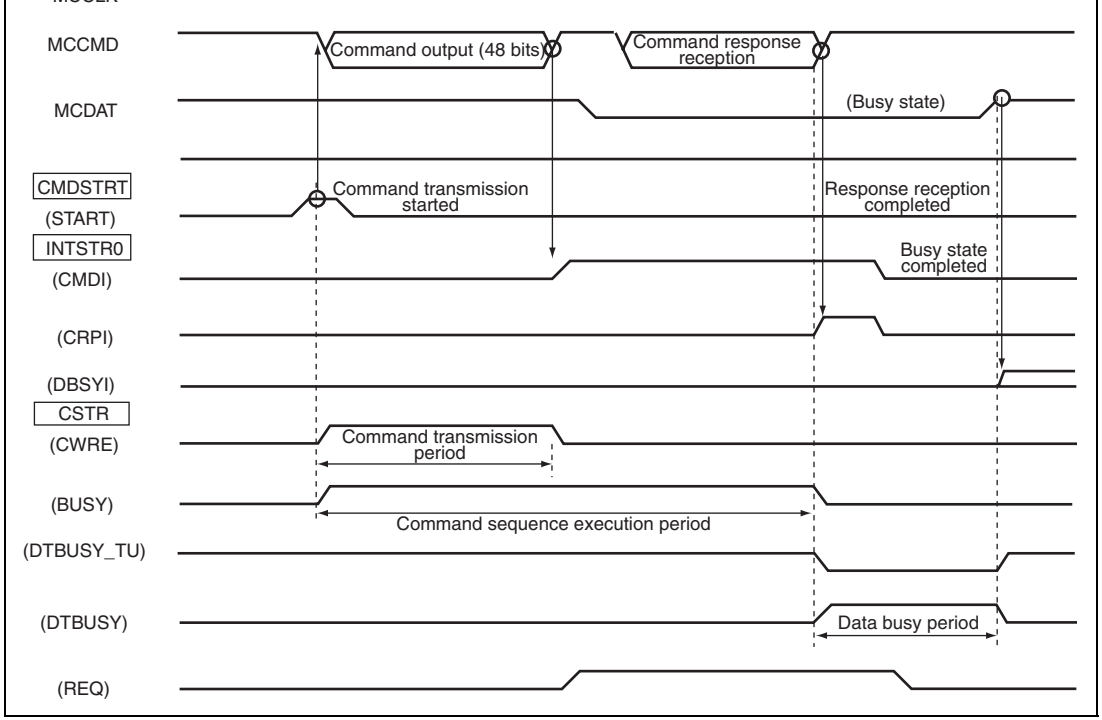


Figure 26.6 Example of Command Sequence for Commands without Data Transfer (with Data Busy State)

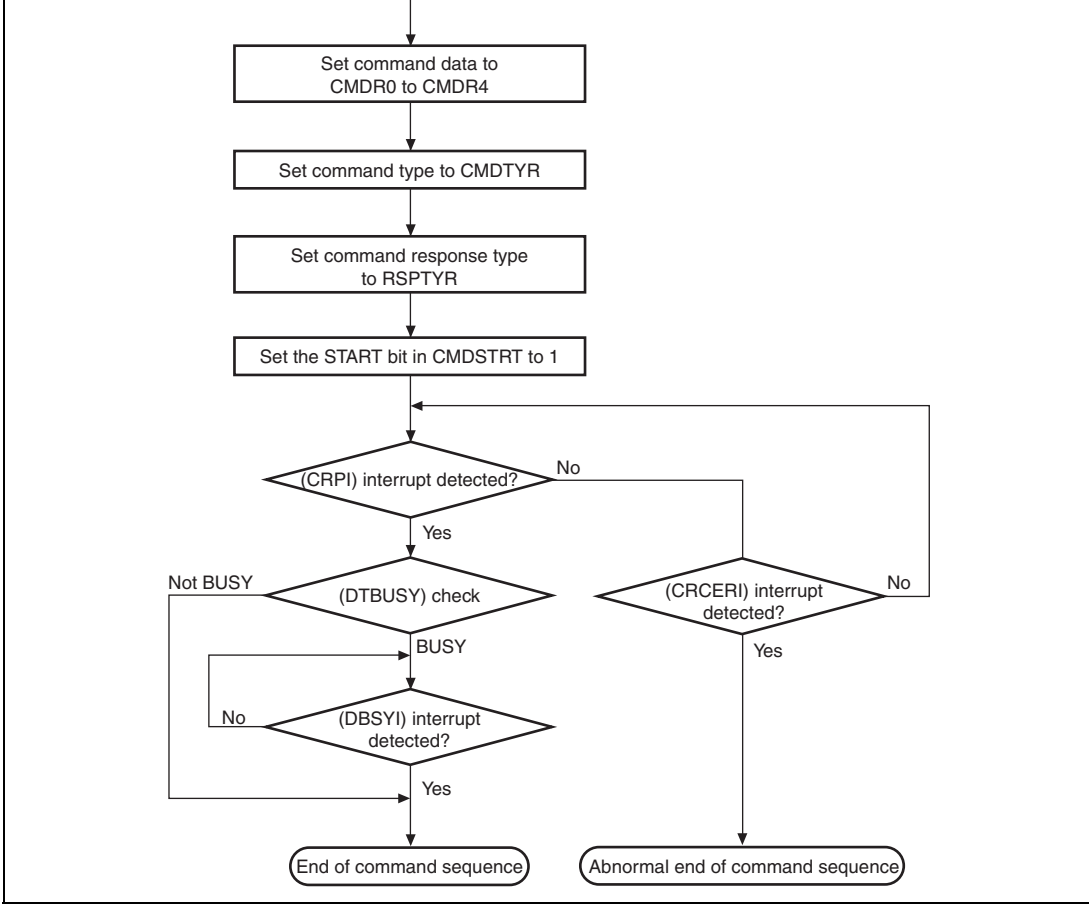


Figure 26.7 Example of Operational Flow for Commands without Data Transfer

Flash memory operation commands include a number of commands involving read data. Such commands confirm the card status by the command argument and command response, and receive card information and flash memory data from the MCDAT pin.

The number of bytes of the flash memory to be read is specified by CMD16 as a block size, or if not specified, reading is continued until it is aborted by CMD12 during multiblock or stream transfer. In multiblock transfer, the transfer operation is suspended for every block and an instruction to continue or end the command sequence is waited for.

Whether the command sequence is suspended or not during the sequence depends on the size of the block and FIFO. The command sequence ends without suspending the data transfer when block size ≤ FIFO size. When block size > FIFO size, the command sequence is suspended by FIFO full. Once the command sequence is suspended, data in FIFO is processed before the command sequence is continued. In multiblock transfer, the command sequence is suspended for every block.

In multiblock transfer, when a block size is set to 4 or 8 bytes and the CMDOFF bit of OPCR is set to 1 for only one block, the command response may not be received correctly. Therefore, when a block size is set to 4 or 8 bytes, the command sequence must be completed after reading at least two blocks.

Figures 26.8 to 26.11 show examples of the command sequence for commands with read data.

Figures 26.12 to 26.14 show the operational flows for commands with read data.

- Make settings to issue the command, and clear FIFO.
- Set the START bit in CMDSTRT to 1 to start command transmission. MCCMD must be kept driven until the end bit output is completed.
  Command transmission completion can be confirmed by the command transmit end interrupt (CMDI).
- The command response is received from the card.
  If the card does not return the command response, the command response is detected by the command timeout error (CTERI).
- Read data is received from the card.
- The inter-block suspension in multiblock transfer and suspension by the FIFO full are detected by the data transfer end interrupt (DTI) and FIFO full interrupt (FFI), respectively.
  To continue the command sequence, the RD_CONTI bit in OPCR should be set to 1. To end the command sequence, the CMDOFF bit in OPCR should be set to 1, and the CMD12 should be issued.

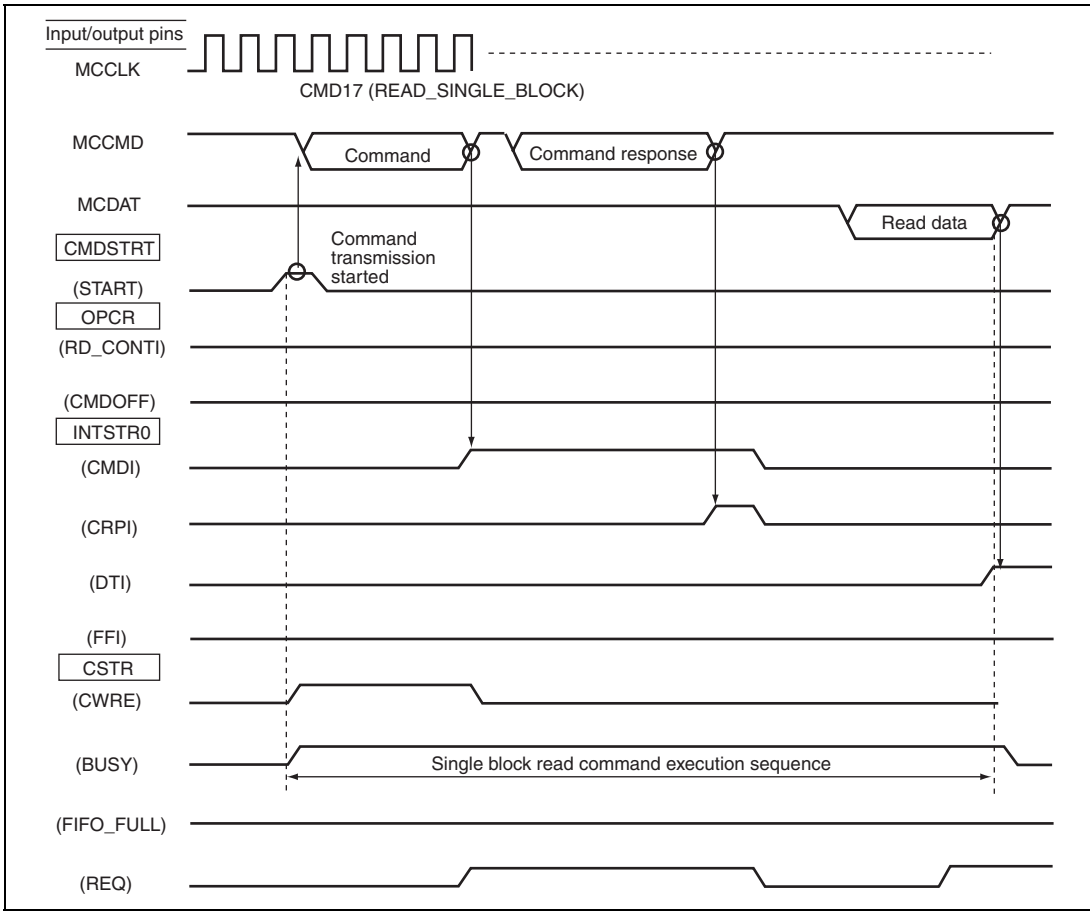


Figure 26.8 Example of Command Sequence for Commands with Read Data (Block Size ≤ FIFO Size)

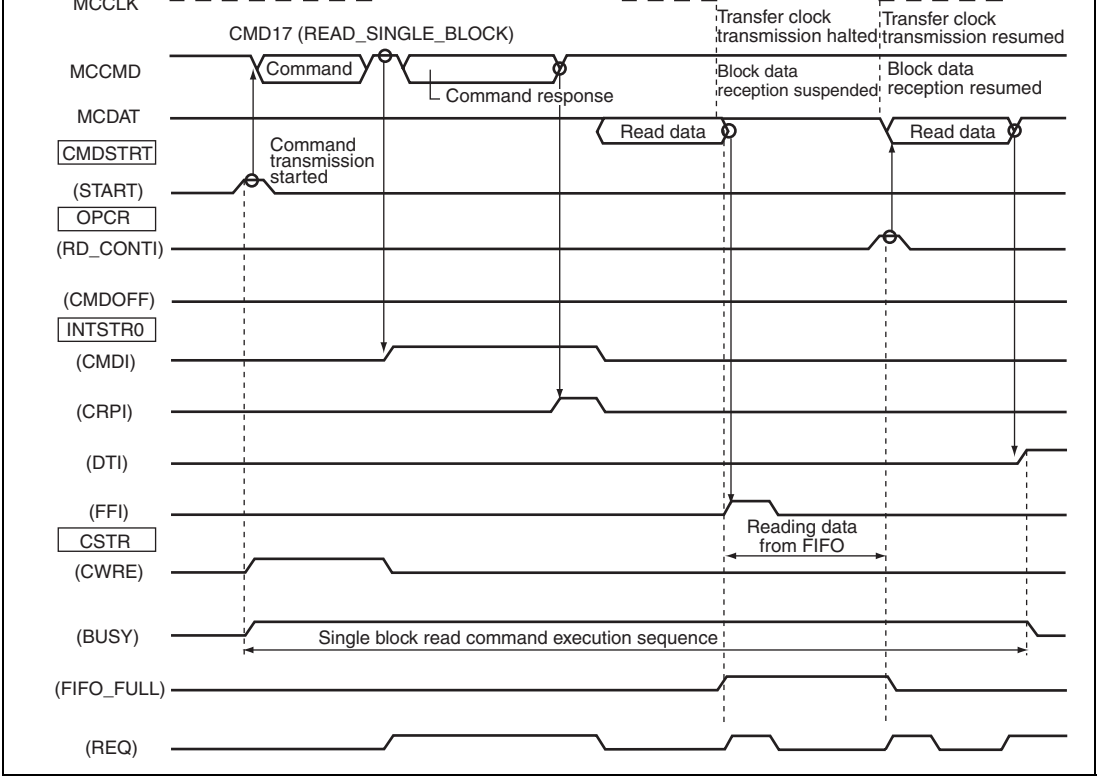


Figure 26.9 Example of Command Sequence for Commands with Read Data (Block Size > FIFO Size)

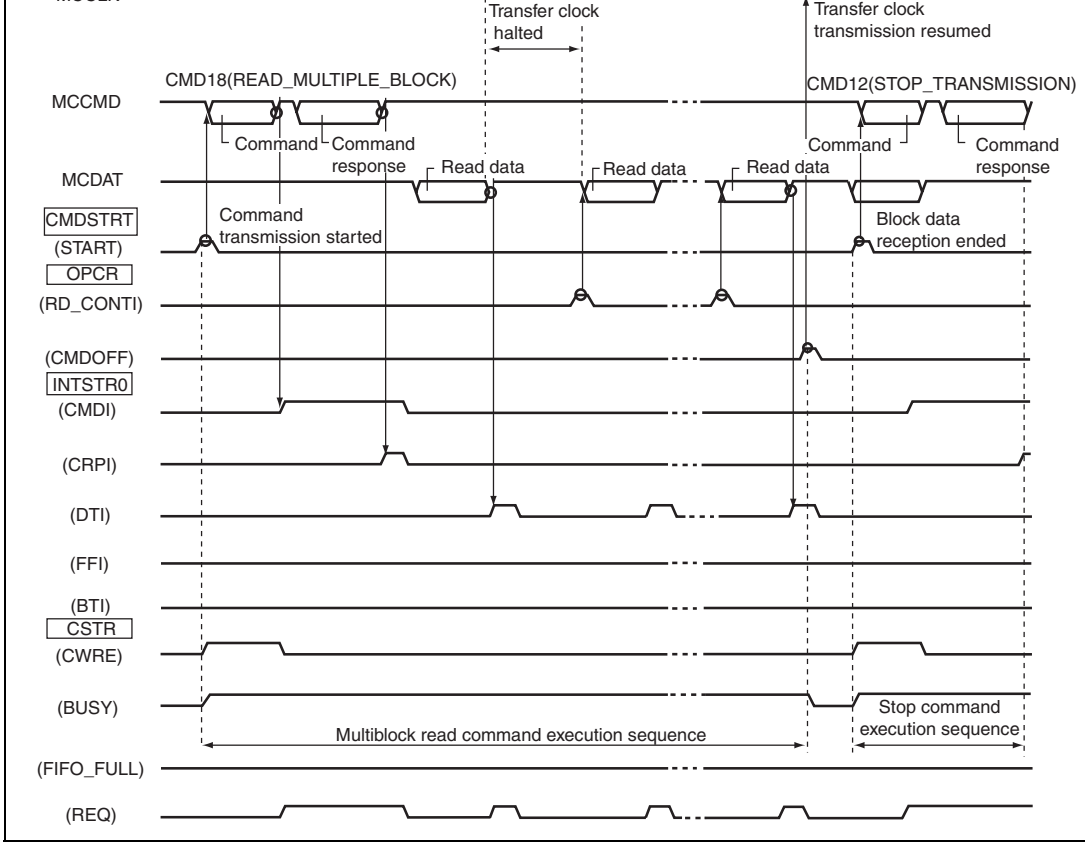


Figure 26.10 Example of Command Sequence for Commands with Read Data (Multiblock Transfer)

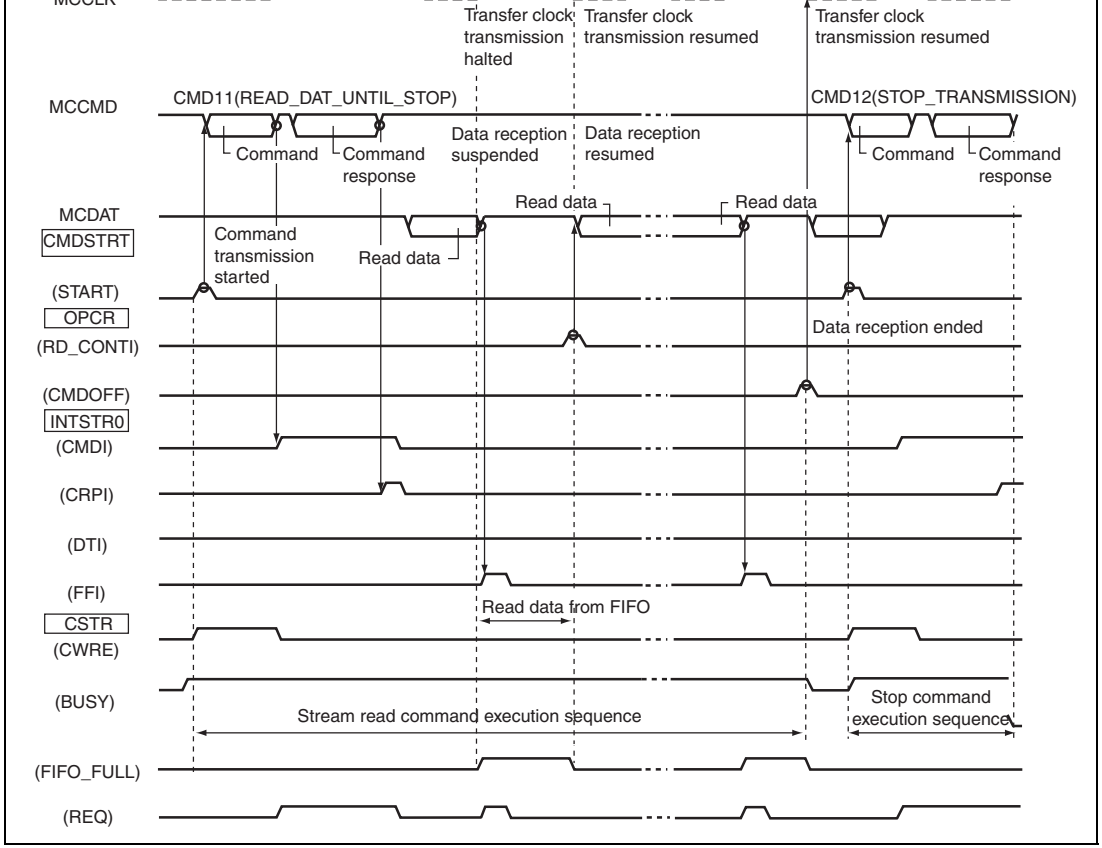


Figure 26.11 Example of Command Sequence for Commands with Read Data (Stream Transfer)

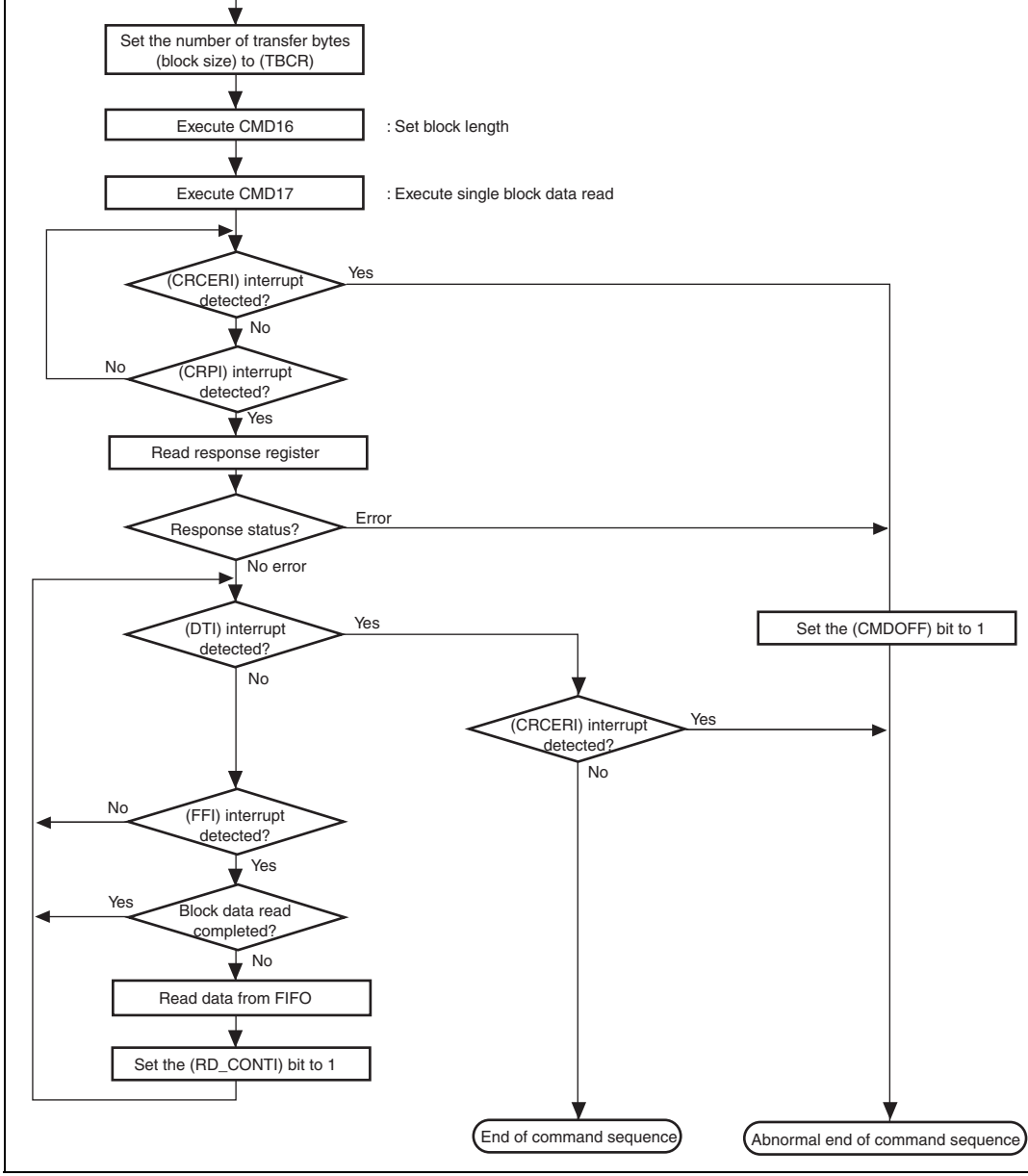


Figure 26.12 Example of Operational Flow for Commands with Read Data (Single Block Transfer)

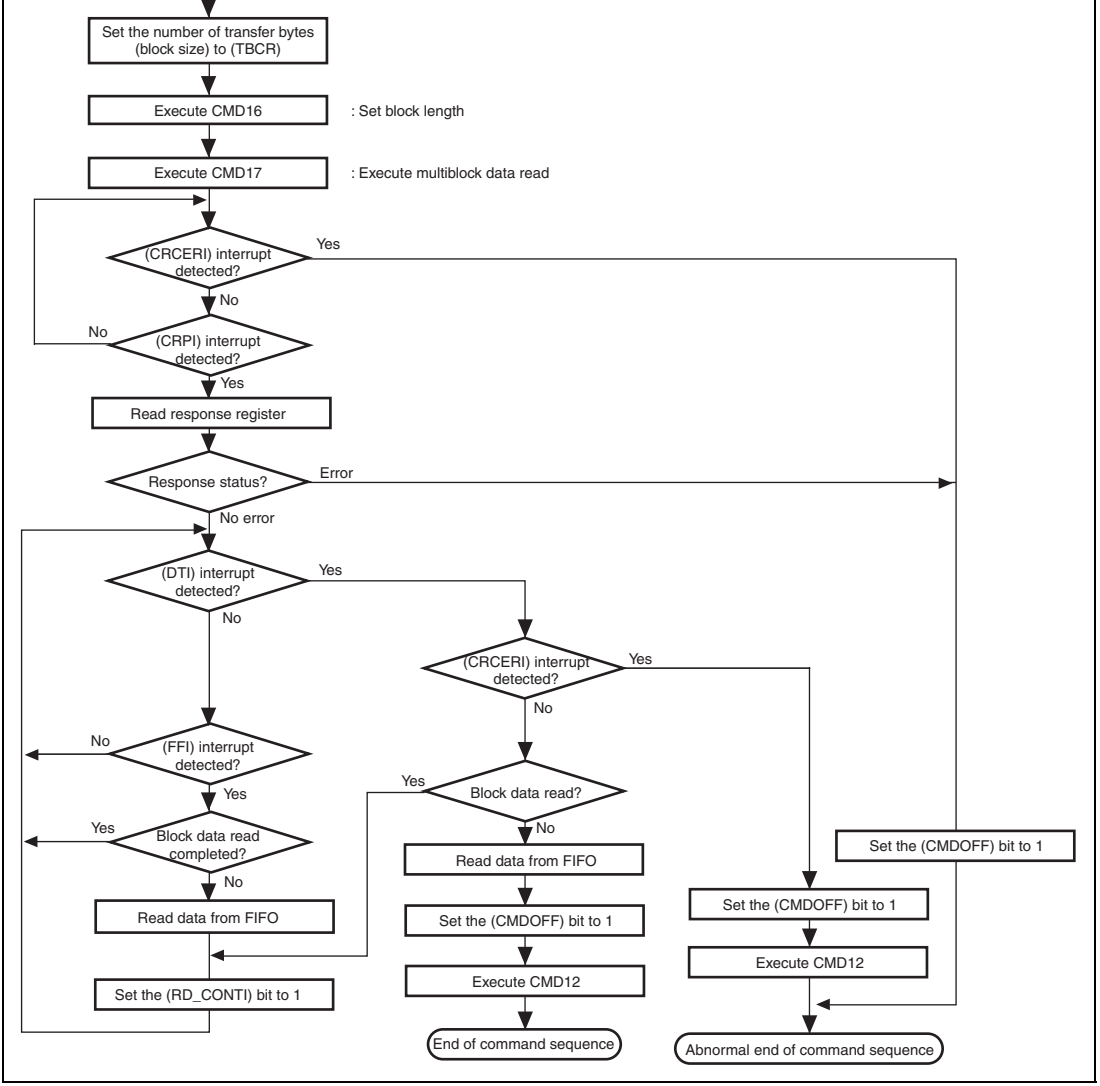


Figure 26.13 Example of Operational Flow for Commands with Read Data (Multiblock Transfer)

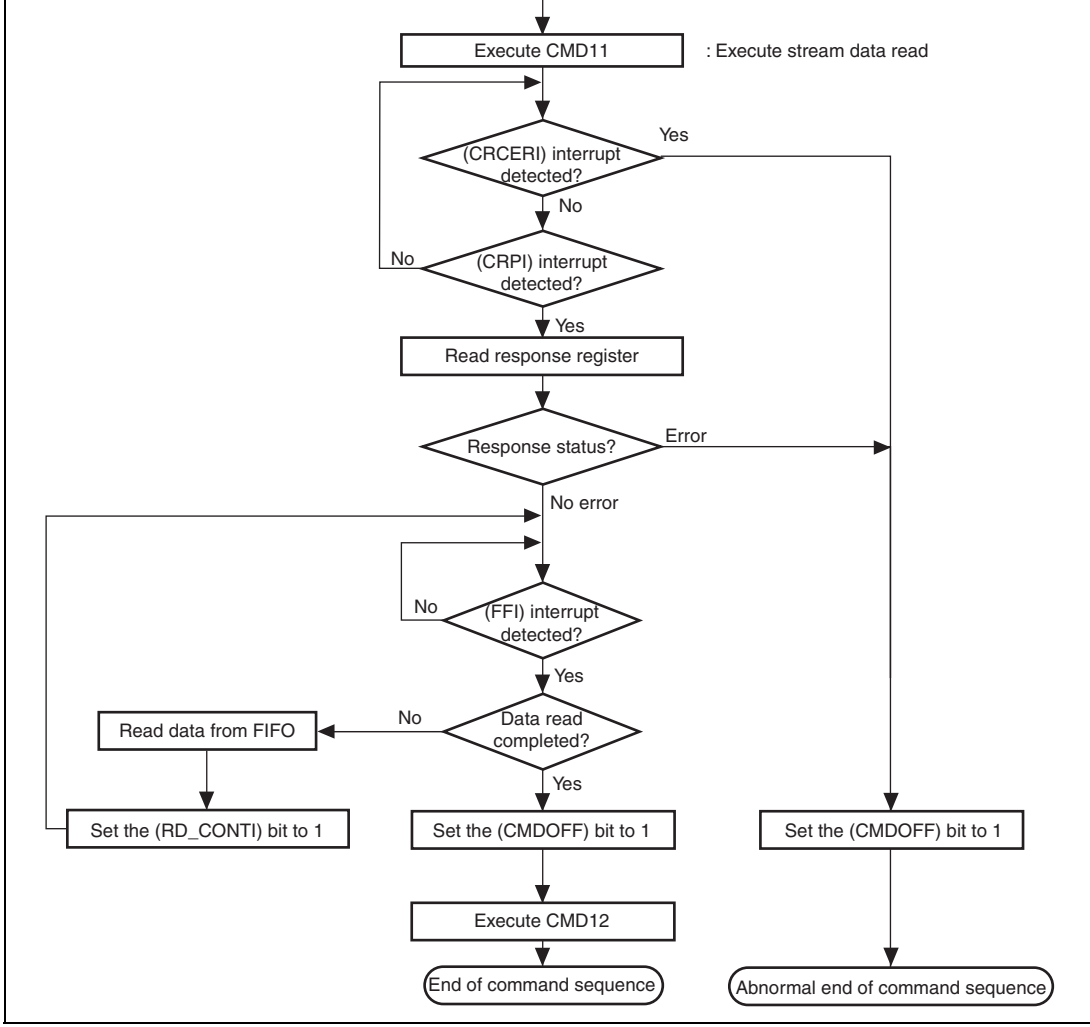


Figure 26.14 Example of Operational Flow for Commands with Read Data (Stream Transfer)

Flash memory operation commands include a number of commands involving write data. Such commands confirm the card status by the command argument and command response, and transmit card information and flash memory data via the MCDAT pin. For a command that is related to time-consuming processing such as flash memory write, the card indicates the data busy state via the MCDAT pin.

The number of bytes of flash memory to be written is specified by CMD16 as a block size, or if not specified, writing is continued until it is aborted by CMD12 during multiblock or stream transfer. In multiblock transfer, the transfer operation is suspended for every block and an instruction to continue or end the command sequence is waited for.

The suspension of the command sequence depends on the size of the block and FIFO. The command sequence ends without suspending the data transfer when block size ≤ FIFO size. When block size > FIFO size, the command sequence is suspended by FIFO empty. Once the command sequence is suspended, the next data is written to the FIFO before the command sequence is continued. In multiblock transfer, the command sequence is suspended for every block.

Figures 26.15 to 26.18 show examples of the command sequence for commands with write data.

Figures 26.19 to 26.21 show the operational flows for commands with write data.

- Make settings to issue a command, and set write data to FIFO.
- Set the START bit in CMDSTRT to 1 to start command transmission. MCCMD must be kept driven until the end bit output is completed.
- Command transmission completion can be confirmed by the command transmit end interrupt (CMDI).
- The command response is received from the card.
- If the card returns no command response, the command response is detected by the command timeout error (CTERI).
- Set the DATAEN bit in OPCR to 1 to start write data transmission. MCDAT must be kept driven until the end bit output is completed.
- Inter-block suspension in multiblock transfer and suspension according to the FIFO empty are detected by the data transfer end interrupt (DTI), data response interrupt (DRPI), and FIFO empty interrupt (FEI), respectively. In addition, after the end of data transfer (DRPI detection), the data busy state is checked through DTBUSY in CSTR. If the card is in data busy state, cancellation of the data busy state is detected by the data busy end interrupt (DBSYI).
  To continue the command sequence, write data should be written to the FIFO, and the DATAEN bit in OPCR should be set to 1. To end the command sequence, the CMDOFF bit in OPCR should be set to 1, and the CMD12 should be issued.

- The data busy state is checked through DTBUSY in CSTR. If the card is in data busy state, the end of the data busy state is detected by the data busy end interrupt (DBSYI).

Note: In a write to the card by stream transfer, the MMCIF continues data transfer to the card even after a FIFO empty interrupt is detected. In this case, complete the command sequence after at least 24 transfer clock cycles.

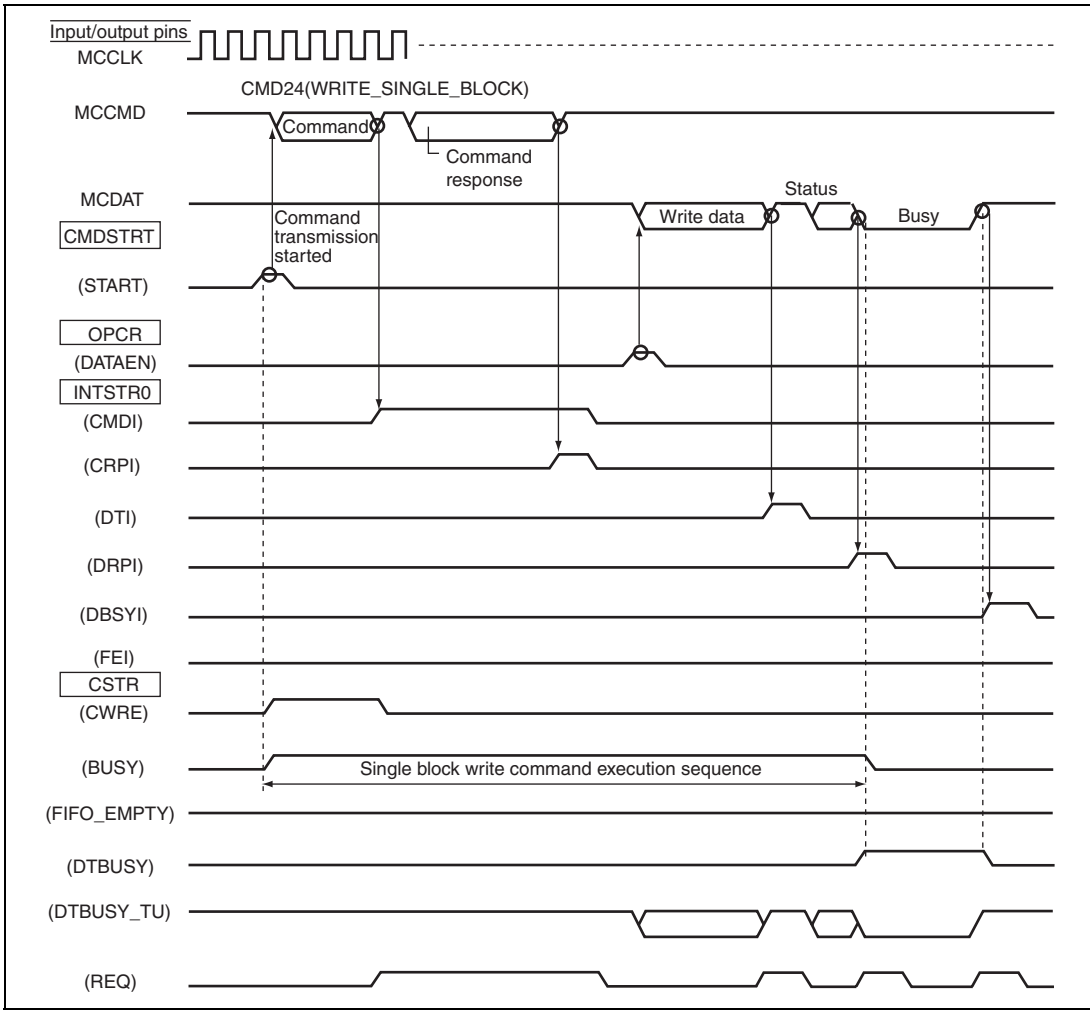


Figure 26.15 Example of Command Sequence for Commands with Write Data (Block Size ≤ FIFO Size)

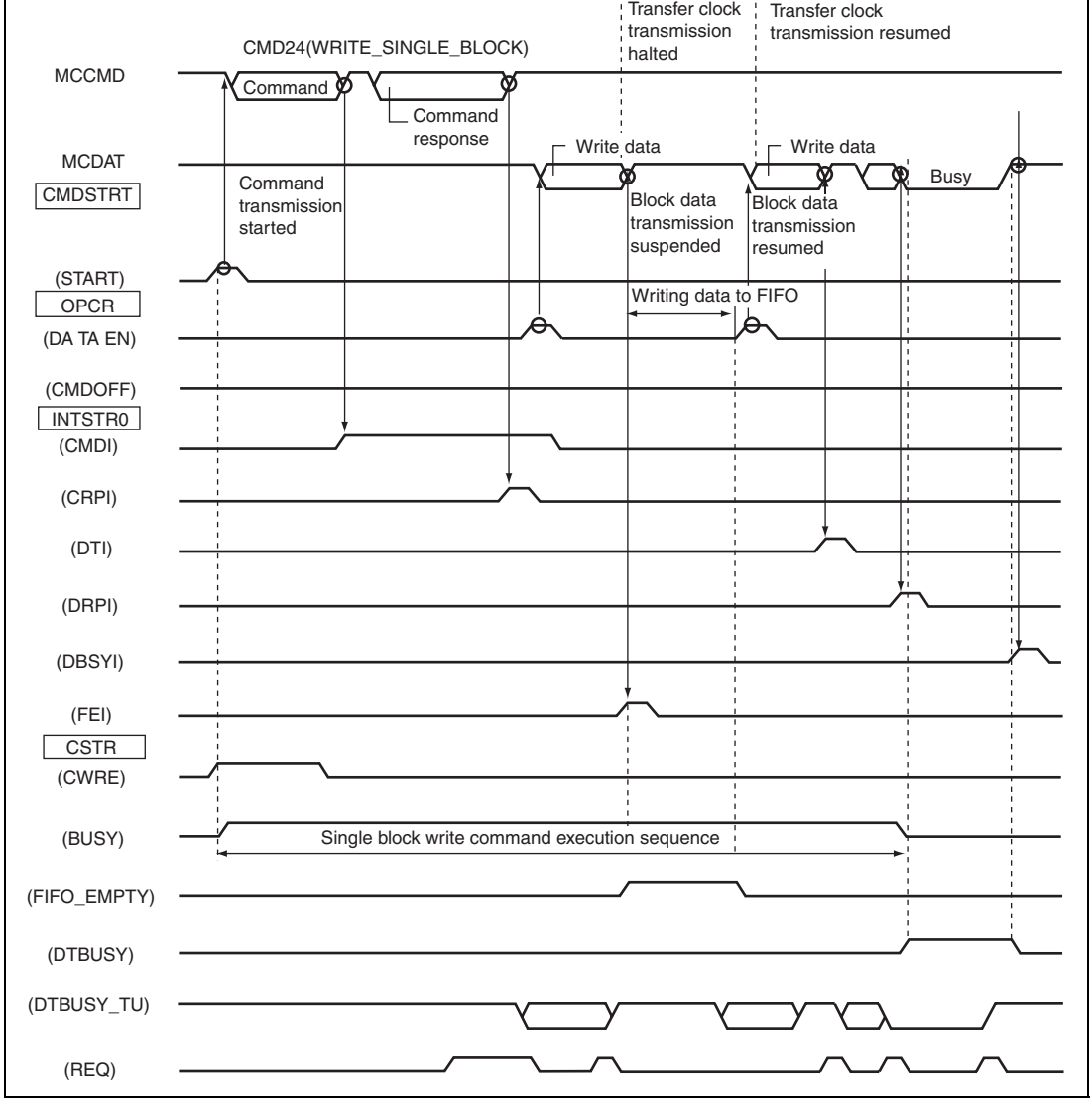


Figure 26.16 Example of Command Sequence for Commands with Write Data (Block Size > FIFO Size)

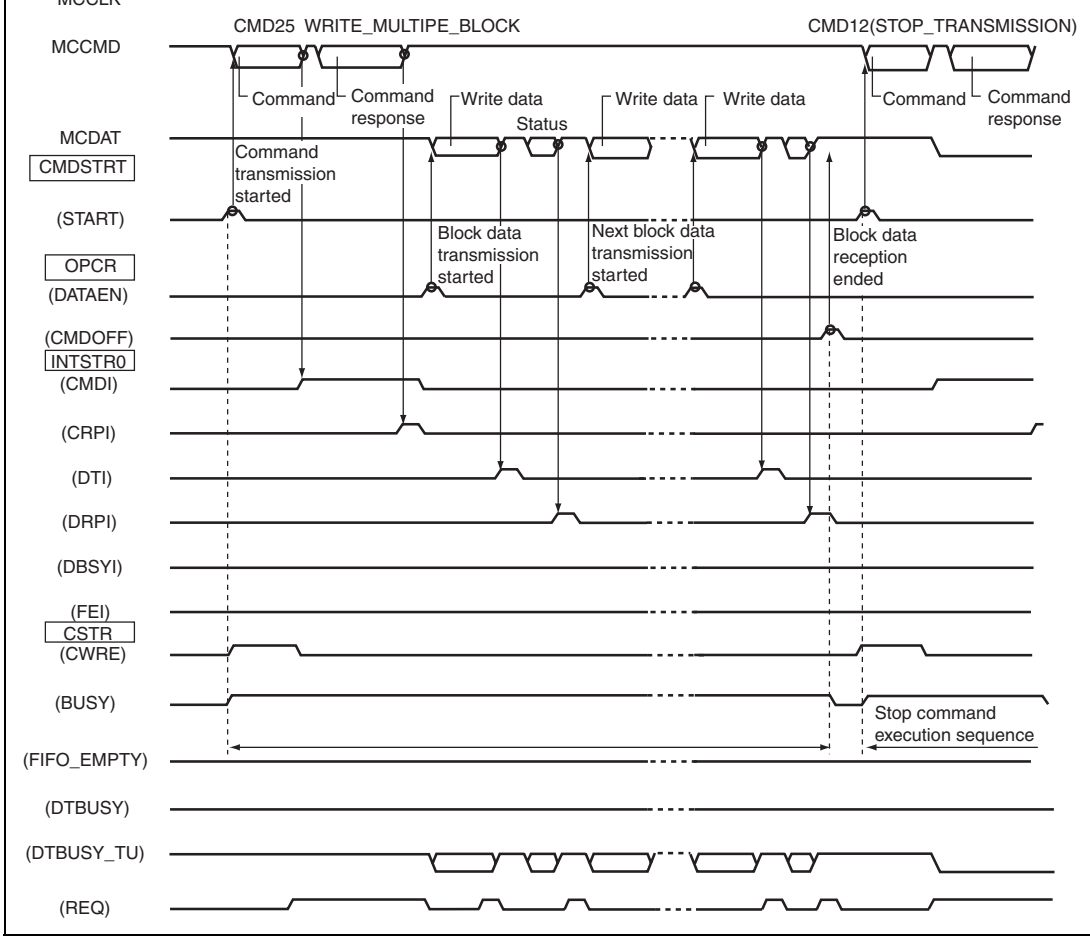


Figure 26.17 Example of Command Sequence for Commands with Write Data (Multiblock Transfer)

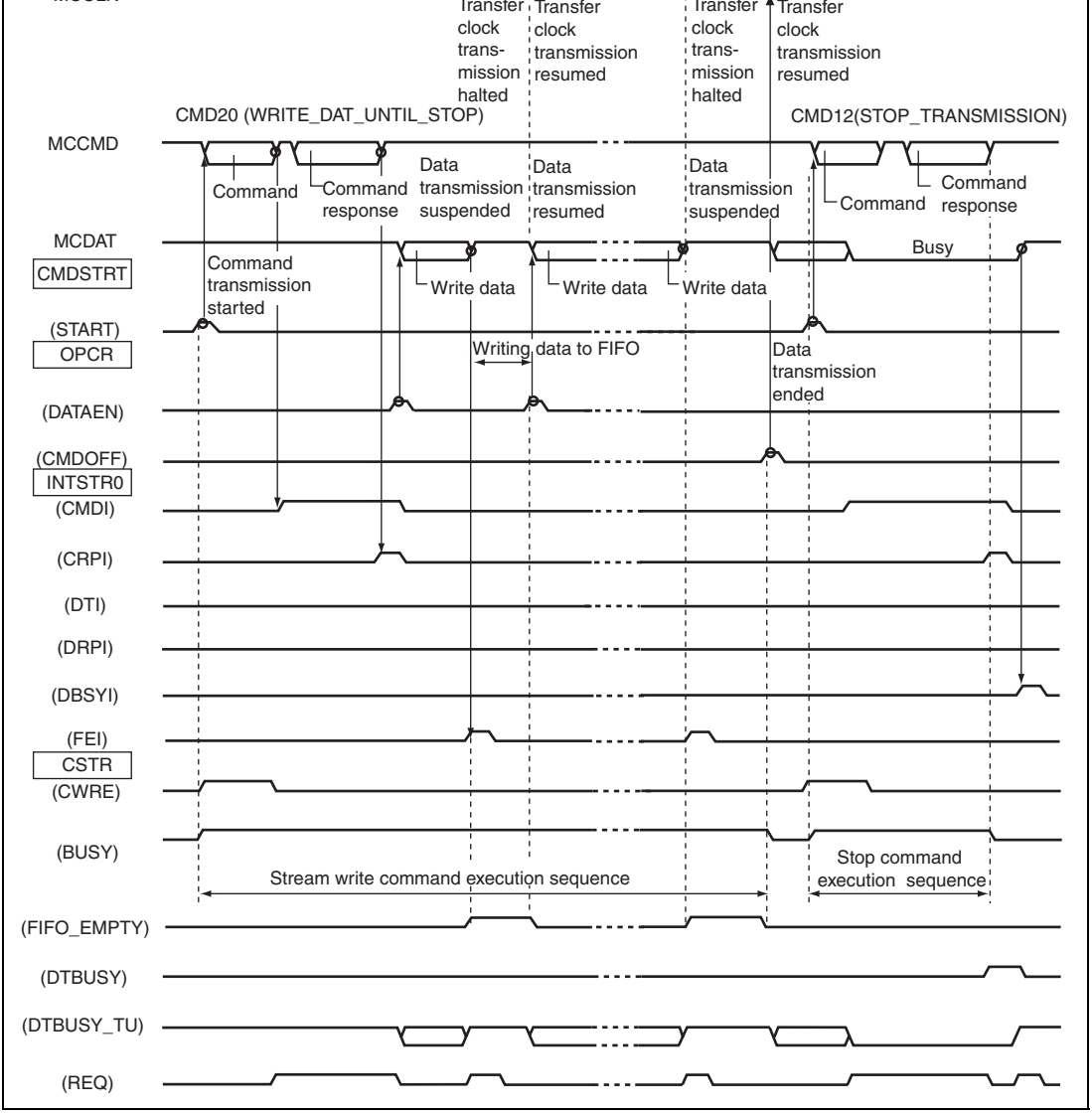


Figure 26.18 Example of Command Sequence for Commands with Write Data (Stream Transfer)

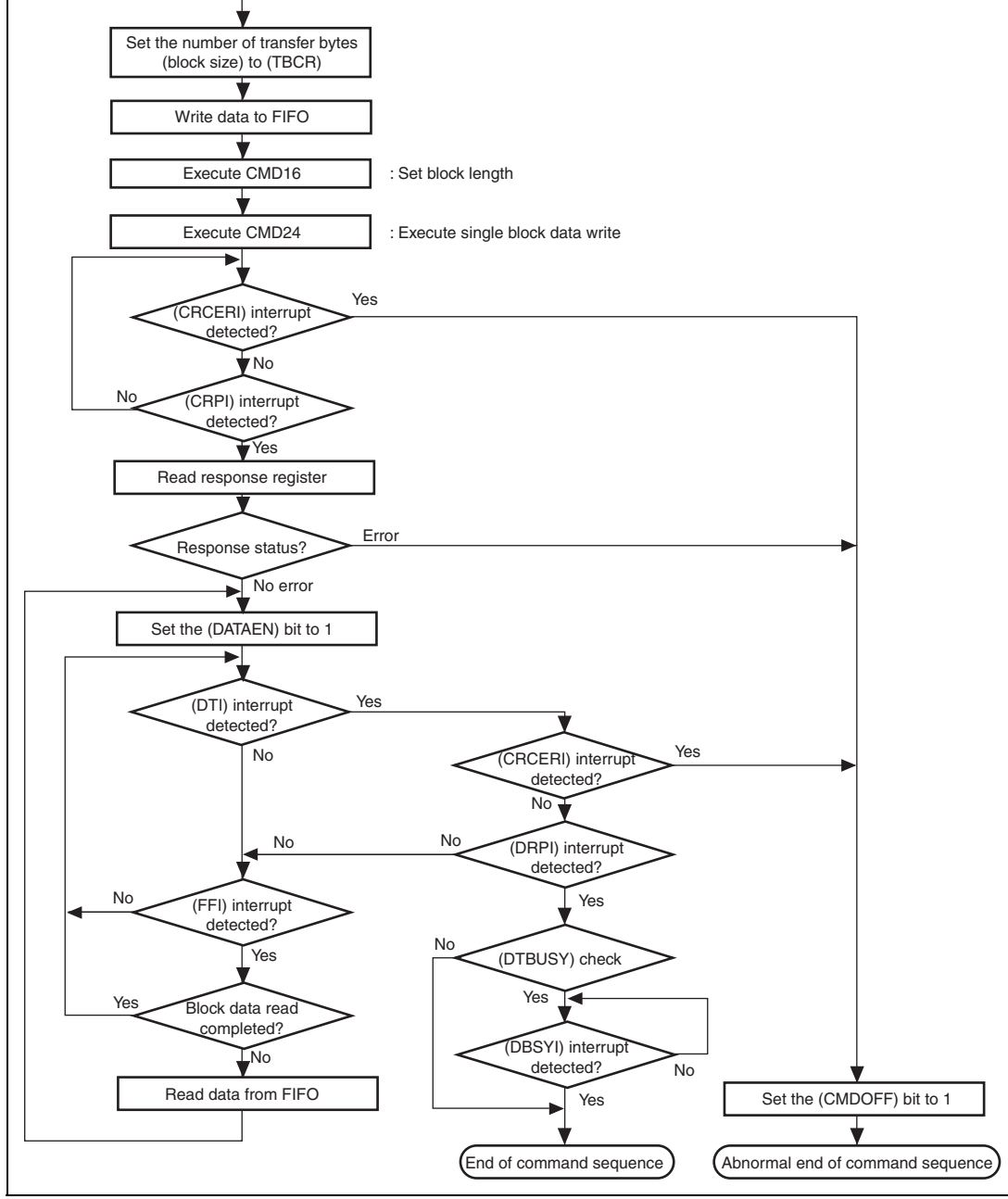


Figure 26.19 Example of Operational Flow for Commands with Write Data (Single Block Transfer)

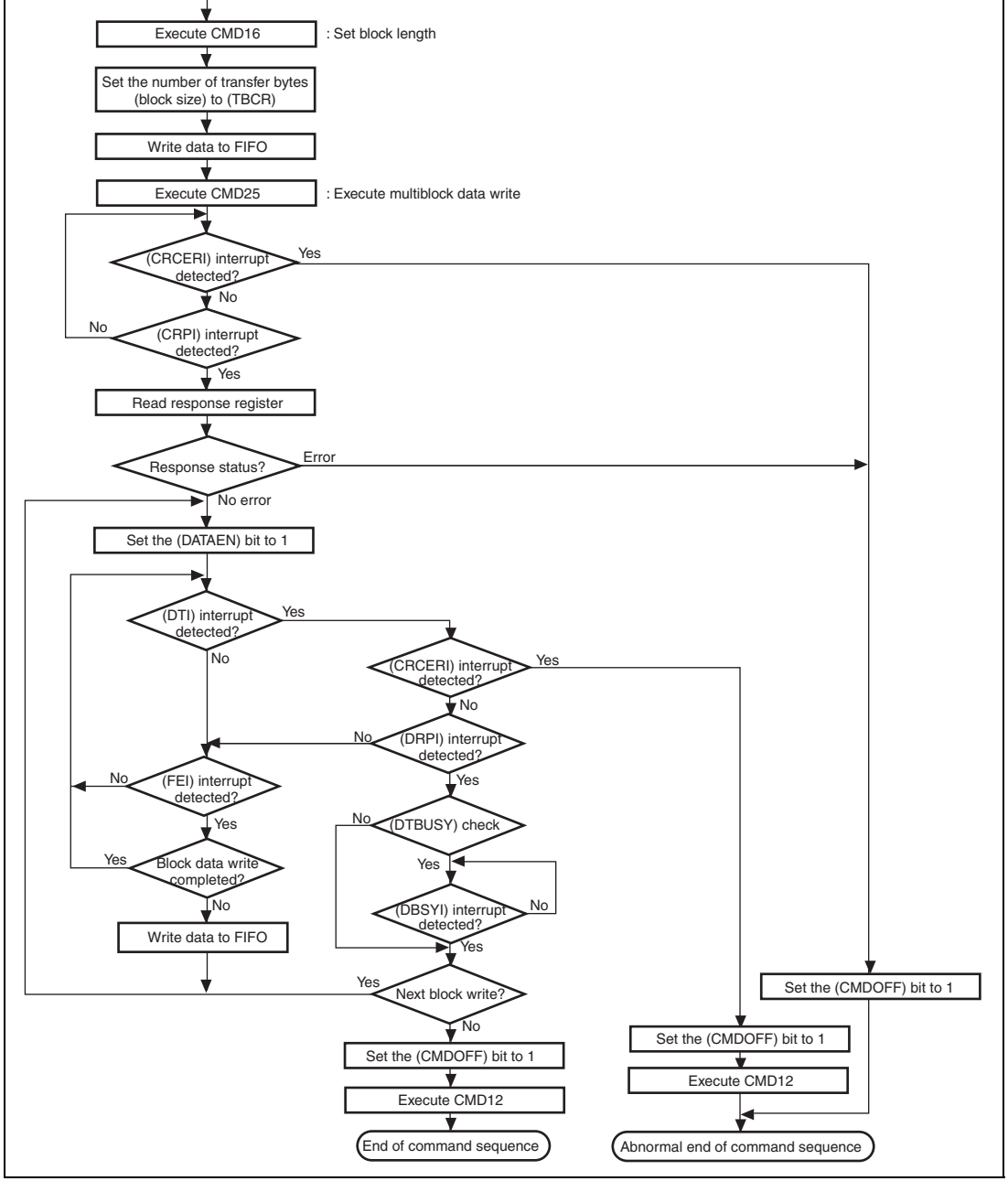


Figure 26.20 Example of Operational Flow for Commands with Write Data (Multiblock Transfer)

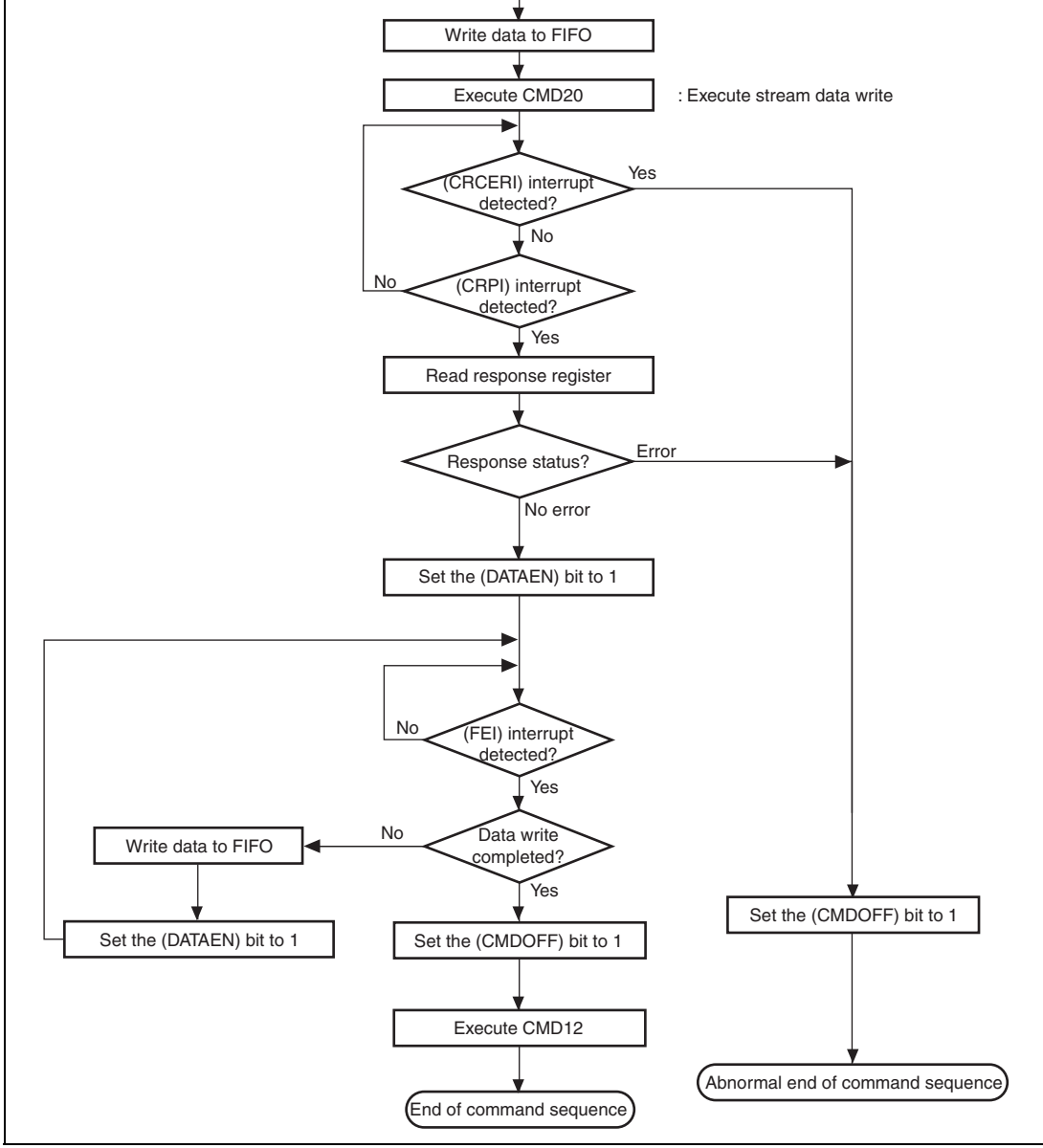


Figure 26.21 Example of Operational Flow for Commands with Write Data (Stream Transfer)

Table 26.7 lists the MMCIF interrupt sources. The interrupt sources are classified into four groups, and four interrupt vectors are assigned. Each interrupt source can be individually enabled by the enable bits in INTCR0 to INTCR2. Disabled interrupt sources do not set the flag.

Table 26.7 MMCIF Interrupt Sources

Name	Interrupt source	Interrupt flag
MMCI0	FIFO empty	FEI
	FIFO full	FFI
MMCI1	Data response	DRPI
	Data transfer end	DTI
	Command response receive end	CRPI
	Command transmit end	CMDI
	Data busy end	DBSYI
MMCI2	CRC error	CRCERI
	Data timeout error	DTERI
	Command timeout error	CTERI
MMCI3	FIFO ready	FRDYI

26.6.1 Operation in Read Sequence

In order to transfer data in FIFO with the DMAC, set MMCIF (DMACR) after setting the DMAC*. Transmit the read command after setting DMACR.

Figure 26.22 shows the operational flow for a read sequence.

- Clear FIFO and make settings in DMACR.
- Read command transmission is started.
- Read data is received from the card.
- After the read sequence, data remains in FIFO. If necessary, write 100 to SET[2:0] in DMACR to read all data from FIFO.
- Confirm that the DMAC transfer is completed and set the DMAEN bit in DMACR to 0.
- When the DMAEN bit in DMACR is set to 1, the FIFO_FULL bit in CSTR and FFI bit in INTSTR0 can not be set.

Note: * Access from the DMAC to FIFO must be done in bytes or words.

26.6.2 Operation in Write Sequence

To transfer data to FIFO with the DMAC, set MMCIF (DMACR) after setting the DMAC. Then, start transfer to the card after a FIFO ready interrupt. Figure 26.23 shows the operational flow for write sequence.

- Make settings in DMACR, and set write data to FIFO.
- After receiving write command response, confirm whether data above the condition of DMACR setting is written to FIFO by a FIFO ready interrupt (FRDYI). Then, set 1 to the DATAEN bit in OPCR to start write-data transmission.
  In a write to the card by stream transfer, the MMCIF continues data transfer to the card even after a FIFO empty interrupt is detected. Therefore, complete the write sequence after at least 24 card clock cycles.
- Confirm that the DMAC transfer is all completed and be sure to set the DMAEN bit in DMACR to 0.
- When the DMAEN bit in DMACR is set to 1, the FIFO_EMPTY bit in CSTR and the FEI bit in INSTR0 can not be set.
- Some combinations of DMACR settings and data transfer count will generate no FIFO ready interrupt (FRDYI) and data will remain in FIFO. In this case, set the DATAEN bit in OPCR to 1 to start write-data transmission.

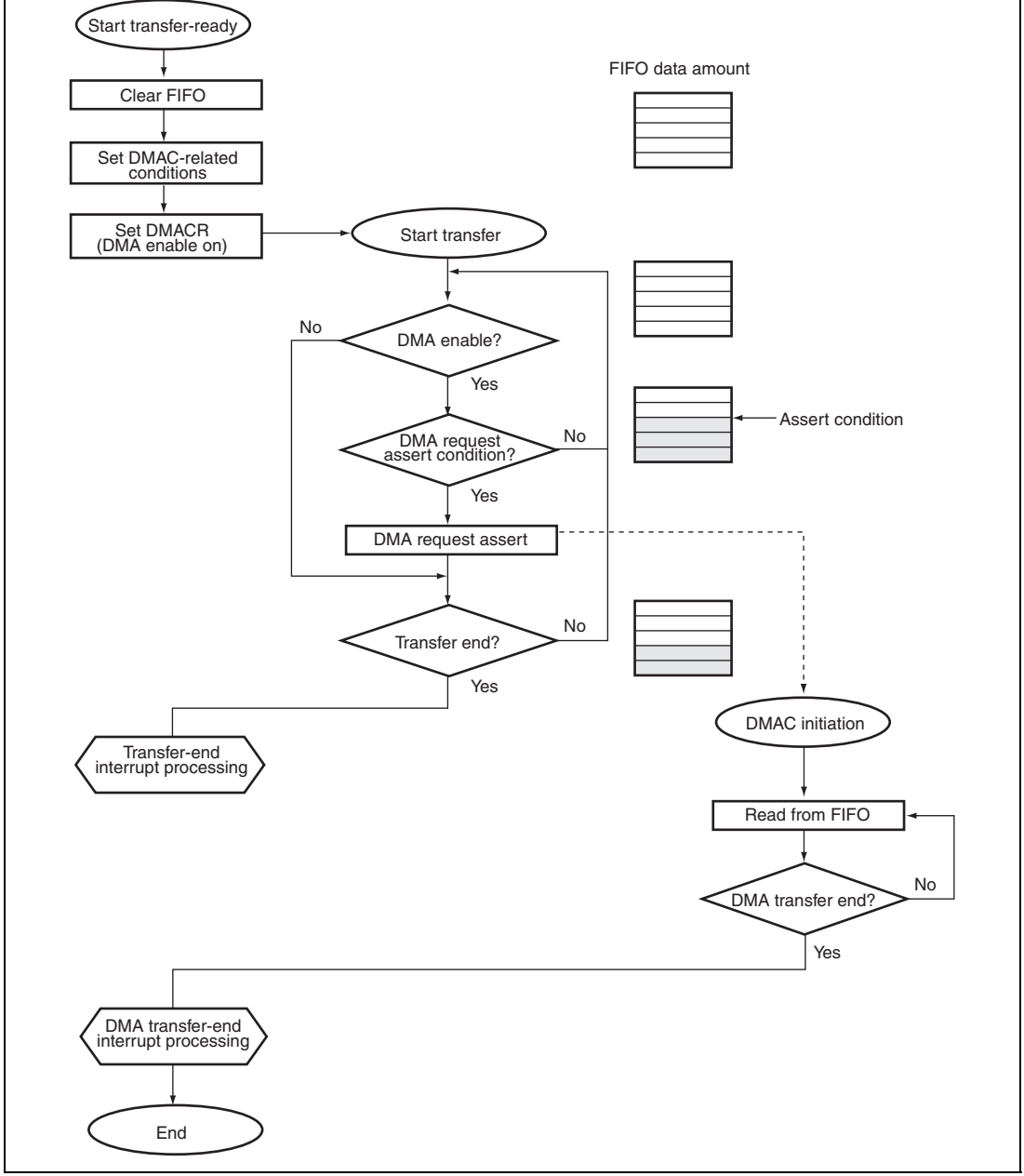


Figure 26.22 Example of Read Sequence Flow

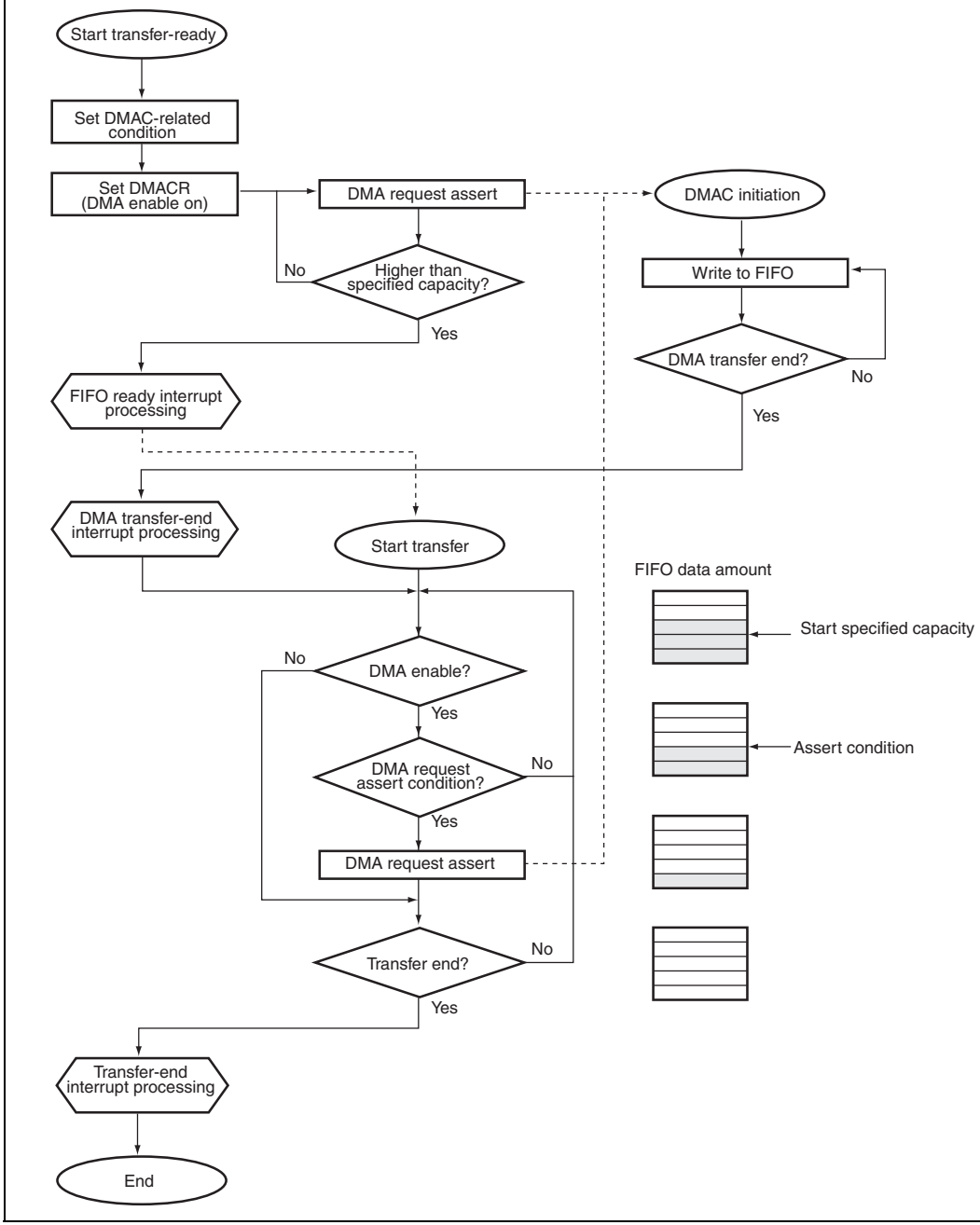


Figure 26.23 Example of Write Sequence Flow

When the little endian is specified, the access size for registers or that for memory where the corresponding data is stored should be fixed. For example, if data read from the MMCIF with the word size is written to memory and then it is read from memory with the byte size, data misalignment occurs.

26.8 Usage Note

26.8.1 Notice about The MMCIF transfer data block size in multiblock read command

(1) Summary

The MMCIF may not receive the command response correctly, when it finishes receiving the end bit of the first data block before receiving the end bit of the command response. It may occur after MMCIF issued CMD18 to the receive device.

(2) Conditions under which the phenomenon occurs

This phenomenon occurs when all of the following operations occur.

1. The multiblock read command (CMD18) is issued.
2. The transfer data block size is setting to 1, 2, 4, or 8 byte(s).
3. The end bit of the command response is received after the end bit of the data block is received.

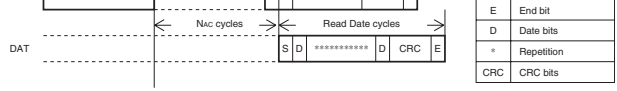
(3) Workaround

1. The transfer data block size in Transfer Byte Number Count Register (TBCR) must be more than 16 bytes when CMD18 is issued.
2. Confirm that the end bit of the command response is received before the end bit of the first data block is received, if the transfer data block size is 1, 2, 4, or 8 bytes when CMD18 is transmitted.

   Keep the below formula in order to confirm that the end bit of the command response is received before the end bit of the first data block is received.

   (N_{AC} cycles + Read Data cycles) > (N_{CR} Cycles + Response cycles)

   Please confirm the N_{AC} and N_{CR} in the spec of target devices to communicate.



E	End bit
D	Date bits
*	Repetition
CRC	CRC bits

Figure 26.24 Timing of Receiving the Command Response and Data

This LSI incorporates a multifunctional interface (MFI) for use in high-speed transfer of data to external devices which cannot share an external bus. The MFI is a parallel interface with selectable 8-bit/16-bit bus width, and can be directly connected to 68/80-series system interfaces.

The MFI allows external devices to read from and write to 2-Kbyte on-chip RAM exclusively for MFI use (MFRAM), in 32-bit units. Access to this MFRAM is available via the MFI and the CPU of this LSI. The MFI supports interrupts issued to this LSI by an external device, and those sent from this LSI to the external device. Using the MFRAM and these interrupt functions enables software-based data transfer between external devices and the on-chip CPU and connection to external devices not having bus privileges.

27.1 Features

- Provides reading from/writing to the 2-Kbyte on-chip MFRAM in 32-bit units via MFI pins, and in 8-, 16-, or 32-bit units from the on-chip CPU.
- Supports a high-speed asynchronous interface with selectable 8-bit/16-bit bus width; allows selection of 68- or 80-series during reset period.
- Automatic address increments and endian settings are configurable.
- Writing to specific bits of MFI on-chip registers from an external device will issue interrupts to this LSI. Conversely, this LSI is able to send interrupts from the on-chip CPU to the external device.
- Provides 7 interrupt source bits each for internal interrupts and for external interrupts. It allows software-based control of 128 different interrupts with high-speed data transfer using interrupts.

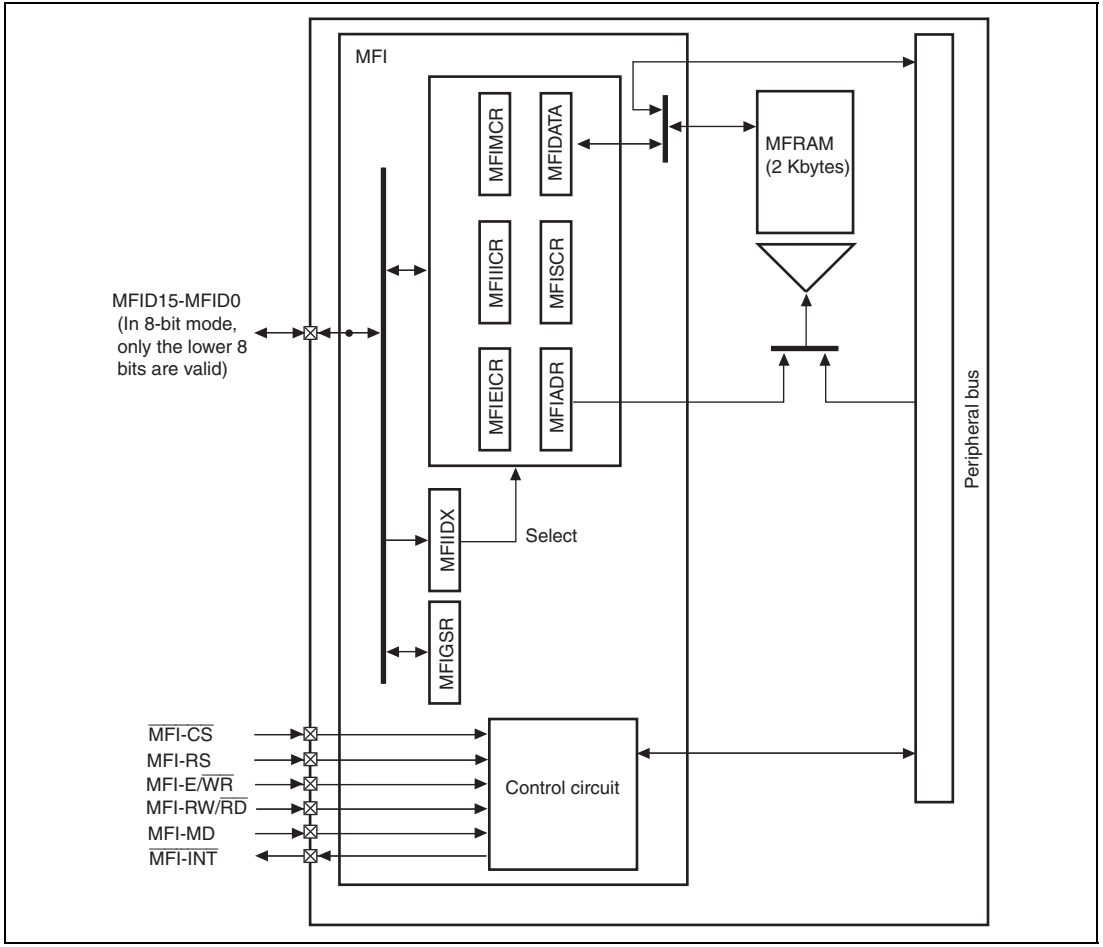


Figure 27.1 MFI block diagram

Table 27.1 shows the MFI pin configuration. To use the MFI , set pin MD7 to MFI mode (MD7 = 0).

Table 27.1 Pin Configuration

Name	Abbreviation	I/O	Description
MFI data	MFI-D15 to MFI-D0	Input/ Output	Address/data/command input/output to the MFI Pins MFI-D7 to MFI-D0 are valid in 8-bit mode.
MFI chip select	$\overline{\text{MFI-CS}}$	Input	Chip select input to the MFI
MFI register select	MFI-RS	Input	Selects MFI access types 0: Normal access 1: Index/status register access
MFI enable/write	MFI-E/$\overline{\text{WR}}$	Input	For a 68-series interface, enable signal to start data writing and reading For an 80-series interface, write strobe signal; data writing at low level
MFI read-write/read	MFI-RW/$\overline{\text{RD}}$	Input	For a 68-series interface, select signal for data writing or reading For an 80-series interface, read strobe signal; data reading at low level
MFI mode	MFI-MD	Input	Selects the MFI 68/80-series interface mode 0: 80-series interface 1: 68-series interface This pin is sampled at a power-on reset by the $\overline{\text{RESET}}$ pin. Do not change the level of this pin after a power-on reset.
MFI interrupt	$\overline{\text{MFI-INT}}$	Output	Interrupt request to an external device from the MFI

The MFI has the following registers. For information on the register addresses and register states during various processing states, refer to section 32, List of Registers.

Table 27.2 Register Configuration (1)

Register Name	Abbrev.	R/W	P4 Address	Area 7 Address	Size	Sync Clock
MFI index register	MFIIDX	R/W*1	H'FE2C 0000	H'1E2C 0000	32	Pck
MFI general status register	MFIGSR	R/W	H'FE2C 0004	H'1E2C 0004	32	Pck
MFI status/control register	MFISCR	R/W*2	H'FE2C 0008	H'1E2C 0008	32	Pck
MFI memory control register	MFIMCR	R/W*4	H'FE2C 000C	H'1E2C 000C	32	Pck
MFI on-chip interrupt control register	MFIIICR	R/W	H'FE2C 0010	H'1E2C 0010	32	Pck
MFI external interrupt control register	MFIEICR	R/W	H'FE2C 0014	H'1E2C 0014	32	Pck
MFI address register	MFIADR	R/W*5	H'FE2C 0018	H'1E2C 0018	32	Pck
MFI data register	MFIDATA	R/W	H'FE2C 001C	H'1E2C 001C	32	Pck
	MFRAM Start	R/W	H'FE2E 0000	H'1E2E 0000	32	Pck
	MFRAM End	R/W	H'FE2E 07FF	H'1E2E 07FF	32	Pck

Register Name	Abbrev.	Power-on Reset by $\overline{\text{RESET}}$ Pin/WDT/ H-UDI	Manual Reset by $\overline{\text{RESET}}$ Pin/WDT/ Multiple Exception	Sleep by Sleep Instruction /Deep Sleep	Standby	
					by Hardware	by Software /Each Module
MFI index register	MFIIDX	H'0000	H'0000	Retained	*6	Retained
MFI general status register	MFIGSR	H'0000	H'0000	Retained		Retained
MFI status/control register	MFISCR	H'0040/H'0050*3	H'0040/H'0050*3	Retained		Retained
MFI memory control register	MFIMCR	H'0000	H'0000	Retained		Retained
MFI on-chip interrupt control register	MFIIICR	H'0000	H'0000	Retained		Retained
MFI external interrupt control register	MFIEICR	H'0000	H'0000	Retained		Retained
MFI address register	MFIADR	H'0000	H'0000	Retained		Retained
MFI data register	MFIDATA	H'0000	H'0000	Retained		Retained
	MFRAM Start	Undefined	Undefined	Retained		Retained
	MFRAM End	Undefined	Undefined	Retained		Retained

Notes: 1. The external device can write to this register only when the MFI-RS pin is driven high. The on-chip CPU cannot write to this register.

2. The external device can write to bit 6 only via the MFI. The on-chip CPU cannot write to this bit. Perform a reading of MFISCR after changing this bit to check for malfunctions.
3. 80-series interface: 0040; 68-series interface: 0050
4. The external device can write to bits 7, 5, 3, and 0 only via the MFI. The on-chip CPU cannot write to these bits.
5. The external device can write to bits 10 to 2 only via the MFI. The on-chip CPU cannot write to these bits.
6. After exiting hardware standby mode, this LSI enters the power-on reset state by the $\overline{\text{RESET}}$ pin.

The MFIIDX is a 32-bit register which is used to specify the register for reading from and writing to via the MFI. Set this register holding the MFI-RS pin low. The external device can write to this register only when the MFI-RS pin is driven high.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
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
	-	-	-	-	-	-	-	-	REG5	REG4	REG3	REG2	REG1	REG0	BYTE1	BYTE0
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*

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7 to 2	REG5 to REG0	All 0	R/W*	Specifies MFI on-chip register 5 to 0. 000000: MFIGSR 000001: MFISCR 000010: MFIMCR 000011: MFIIICR 000100: MFIEICR 000101: MFIADR 000110: MFIDATA Settings other than the above are prohibited.

1	BYTE1	0	R/W*	Specifies byte position for on-chip register.
0	BYTE0	0	R/W*	Specifies which 8 or 16 bits of the 32-bit register are to be accessed. • MFISCR.BO = 0 8-bit bus / 16-bit bus 00: Register bits 31 to 24 / Register bits 31 to 16 01: Register bits 23 to 16 / Setting prohibited 10: Register bits 15 to 8 / Register bits 15 to 0 11: Register bits 7 to 0 / Setting prohibited • MFISCR.BO = 1 8-bit bus / 16-bit bus 00: Register bits 7 to 0 / Register bits 15 to 0 01: Register bits 15 to 8 / Setting prohibited 10: Register bits 23 to 16 / Register bits 31 to 16 11: Register bits 31 to 24 / Setting prohibited However, with MFIDATA selected by bits REG5 to REG0, each time reading from or writing to MFIDATA from the external device occurs, bits BYTE1 and BYTE0 change according to the following rules. 8-bit bus: 00 → 01 → 10 → 11 → 00 → 01... etc. 16-bit bus: 00 → 10 → 00 → 10... etc.

Note: * The external device can write to these bits via the MFI only when the MFI-RS pin is 1. The on-chip CPU cannot write to these bits.

27.3.2 MFI General Status Register (MFIGSR)

The MFIGSR is a 32-bit register which an MFI-connected external device uses to indicate its status to the on-chip CPU and vice versa. When the MFI-RS pin is driven high, this register is read-only via the MFI. To write to the MFIGSR from the MFI, specify MFIGSR setting bits REG5 to REG0, drive the MFI-RS pin low and then perform writing. In this state, the MFIGSR can also be read.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
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
	-	-	-	-	-	-	-	-	STATUS7	STATUS6	STATUS5	STATUS4	STATUS3	STATUS2	STATUS1	STATUS0
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

31 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7 to 0	STATUS7 to STATUS0	All 0	R/W	General status These bits can be read from and written to by the software of an MFI-connected external device and by the on-chip CPU. These bits are not modified by hardware other than a power-on reset.

27.3.3 MFI Status/Control Register (MFISCR)

The MFISCR is a 32-bit readable/writable register which is used to control the MFI mode and state.

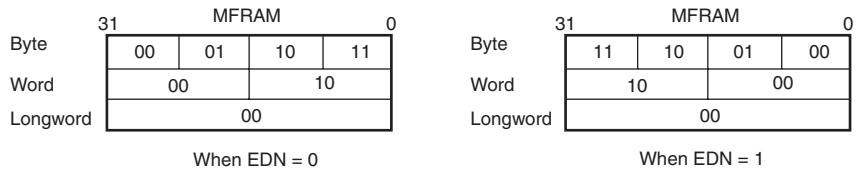
Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	SCRMD2	-	SCRMD0	-	-	EDN	BO
Initial value:	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R/W*1	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 7	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
6	SCRMD2	1	R/W*	MFI mode 2 Specifies the MFI bus width. Changing this bit immediately takes effect to change the bus width. 0: 8-bit mode 1: 16-bit mode Note: In order to check for malfunctions, perform a dummy reading of MFISCR after changing this bit.

5	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
4	SCRMD0	0	R	MFI mode 0 The MFI-MD pin value is sampled at a power-on reset by the $\overline{\text{RESET}}$ pin. Indicates whether the MFI is the 68- or the 80-series interface. Indicates the value of the MFI-MD signal. 0: 80-series interface 1: 68-series interface
3, 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1	EDN	0	R/W	Endian setting (for MFRAM access) Specifies the byte order when accessing the MFRAM from the on-chip CPU. See figure 27.2. (Can be set independently from the MD5 pin setting of this LSI.) 0: Big endian 1: Little endian
0	BO	0	R/W	Byte order Specifies the byte order of MFIDATA corresponding to MFIADR. 0: Big endian 1: Little endian

Note: * The external device can write to this bit via the MFI. The on-chip CPU cannot write to this bit.



Relation between lower two bits of address and access size used when accessing MFRAM

Figure 27.2 Differences in EDN Bit Settings

The MFIMCR is a 32-bit register that the external device uses to control the MFRAM via the MFI.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
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
	-	-	-	-	-	-	-	-	LOCK	-	WT*3	-	RD*3	-	-	AI/AD
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*1	R	R/W*1	R	R/W*1	R	R	R/W*1

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7	LOCK	0	R/W*1	Lock This bit is used to lock read/write operations during continuous access. Writing 1 to the LOCK bit retains the values of the RD and WT bits simultaneously set until clearing the LOCK bit to 0. Setting both the RD and LOCK bits simultaneously to 1 puts the MFI in the continuous read mode; setting both the WT and LOCK bits simultaneously to 1 results in the continuous write mode. Do not set the RD and WT bits simultaneously to 1.
6	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
5	WT*3	0	R/W*1	Write Setting this bit to 1 writes the MFIDATA value to the MFRAM address indicated by MFIADR.*2 • Setting both the WT and LOCK bits simultaneously to 1 results in the continuous write mode and enables high-speed data transfer *4. The WT value remains 1 until the WT bit is next written to 0, or until the LOCK bit is cleared to 0. • If not setting the LOCK bit simultaneously to 1, writing to MFRAM is performed only once. The WT bit is automatically cleared to 0.

4	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
3	RD*3	0	R/W*1	Read Setting this bit to 1 reads the MFRAM data indicated by MFIADR into MFIDATA.*2 • Setting the RD and LOCK bits simultaneously to 1 results in the continuous read mode, and enables high-speed data transfer. The RD bit remains 1 until the RD bit is next written to 0, or until the LOCK bit is cleared to 0. • If not setting the LOCK bit simultaneously to 1, reading of MFRAM is performed only once. The RD bit is automatically cleared to 0
2, 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	AI/AD	0	R/W*1	Address auto-increment/decrement This bit is valid only when the LOCK bit is 1. Each time an MFRAM read or write operation occurs, the value in MFIADR is automatically changed by +4 or by -4. 0: Auto-increment (+4) 1: Auto-decrement (-4)

Notes: 1. The external device can write to this bit via the MFI. The on-chip CPU cannot write to this bit.
2. If the on-chip CPU and the external device via the MFI access MFRAM concurrently, the access via the MFI is handled first.
3. Do not set the WT and RD bits simultaneously to 1.
4. Performs continuous writing to MFRAM in 32-bit units. Data with a length of less than 32 bits is not written to MFRAM.

The MFIIICR is a 32-bit register that an MFI-connected external device uses to issue interrupts to the on-chip CPU.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
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
	-	-	-	-	-	-	-	-	IIC6	IIC5	IIC4	IIC3	IIC2	IIC1	IIC0	IIR
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

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7	IIC6	0	R/W	Internal interrupt source Bits used to specify the interrupt source generated by the IIR. Both the MFI-connected external device and the on-chip CPU can write to these bits. Using these bits enables fast interrupt handling. These bits are completely under software control, and their values have no effect on the operation of the LSI.
6	IIC5	0	R/W	
5	IIC4	0	R/W	
4	IIC3	0	R/W	
3	IIC2	0	R/W	
2	IIC1	0	R/W	
1	IIC0	0	R/W	
0	IIR	0	R/W	Internal interrupt request While this bit is 1, an interrupt request is issued to the on-chip CPU.

The MFIEICR is a 32-bit register that the on-chip CPU uses to issue interrupts an MFI-connected external device.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
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
	-	-	-	-	-	-	-	-	EIC6	EIC5	EIC4	EIC3	EIC2	EIC1	EIC0	EIR
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

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7	EIC6	0	R/W	External interrupt source Bits used to specify the interrupt source generated by the EIR. Both the MFI-connected external device and the on-chip CPU can write to these bits. Using these bits enables fast interrupt handling. These bits are completely under software control, and their values have no effect on the operation of the LSI.
6	EIC5	0	R/W	
5	EIC4	0	R/W	
4	EIC3	0	R/W	
3	EIC2	0	R/W	
2	EIC1	0	R/W	
1	EIC0	0	R/W	
0	EIR	0	R/W	External interrupt request While this bit is 1, the $\overline{\text{MFI-INT}}$ pin is asserted low and interrupt request is issued to the external device from this LSI.

The MFIADR is a 32-bit register which indicates the address in the MFRAM to be accessed by the external device via the MFI.

Specifying continuous access to the MFRAM in the LOCK bit in MFIMCR automatically performs auto-increment (+4) or auto-decrement (-4) of the address according to the AI/AD bit in MFIMCR, and updates MFIADR each time the external device accesses the MFRAM.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
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
	-	-	-	-	-	A10	A9	A8	A7	A6	A5	A4	A3	A2	-	-
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/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 2	A10 to A2	All 0	R/W*	Address Specifies the memory space in the 2-Kbyte MFRAM to be accessed by the external device via the MFI, with 32-bit alignment.
1, 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Note: * The external device can write to these bits via the MFI. The on-chip CPU cannot write to these bits.

The MFIDATA is a 32-bit register which is used to hold data to be written to the MFRAM and data read from the MFRAM. It is possible to use the MFDATA for data transfer between an MFI-connected external device and the software on this LSI if this register is not used for accessing the MFRAM.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MFI DATA31	MFI DATA30	MFI DATA29	MFI DATA28	MFI DATA27	MFI DATA26	MFI DATA25	MFI DATA24	MFI DATA23	MFI DATA22	MFI DATA21	MFI DATA20	MFI DATA19	MFI DATA18	MFI DATA17	MFI DATA16
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
	MFI DATA15	MFI DATA14	MFI DATA13	MFI DATA12	MFI DATA11	MFI DATA10	MFI DATA9	MFI DATA8	MFI DATA7	MFI DATA6	MFI DATA5	MFI DATA4	MFI DATA3	MFI DATA2	MFI DATA1	MFI DATA0
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	Bit Name	Initial Value	R/W	Description
31 to 0	MFIDATA31 to MFIDATA0	All 0	R/W	32-bit data

27.4.1 Overview

Access to the MFI is controlled by a combination of settings in the $\overline{\text{MFI-CS}}$, MFI-RS, MFI-E/$\overline{\text{WR}}$, and MFI-RW/$\overline{\text{RD}}$ pins. Table 27.3 shows the relationship between combinations of these signals and MFI operations. For access to MFIIDX and MFIGSR, refer to table 27.4.

Table 27.3 MFI Operations

		68 Series		80 Series		
$\overline{\text{MFI-CS}}$	MFI-RS	MFI-E /$\overline{\text{WR}}$	MFI-RW /$\overline{\text{RD}}$	MFI-E /$\overline{\text{WR}}$	MFI-RW /$\overline{\text{RD}}$	Operation
1	x	x	x	x	x	No operation (NOP)
0	0	1	1	1	0	Read from register specified by MFIIDX[7:0]
0	0	1	0	0	1	Write to register specified by MFIIDX[7:0]
0	1	1	1	1	0	Read from MFIGSR[7:0]*
0	1	1	0	0	1	Write to MFIIDX[7:0]*
0	x	0	x	1	1	No operation (NOP)
0	x	—	—	0	0	Setting prohibited

Notes: * Performs reading and writing with MFI-D7 to MFI-D0 when using the MFI with a 16-bit width.

x: Don't care

Table 27.4 Access to MFIIDX and MFIGSR

		External device*	
	CPU	8 bits	16 bits
MFIIDX	[31:0]	[7:0]	[15:0]
MFIGSR	[31:0]	[7:0]	[15:0]

Note: * When the MFI-RS pin is driven high.

Figure 27.3 shows an example of the recommended connections between the MFI and an external device.

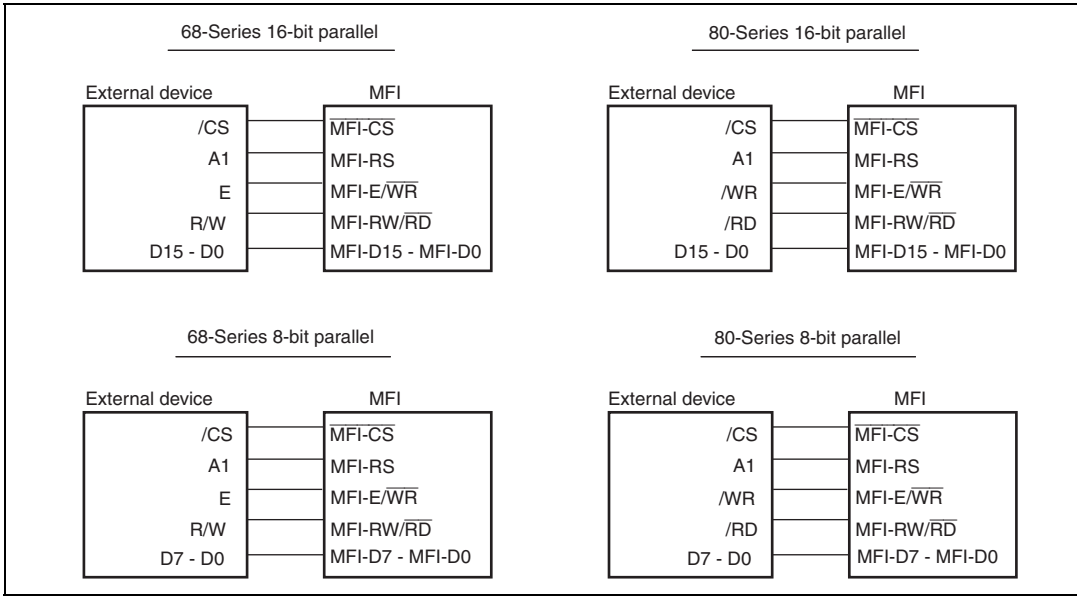


Figure 27.3 Example of MFI Connections

27.4.3 Memory Map

Table 27.5 shows the memory space in the internal MFRAM.

Table 27.5 Memory Map

	Start address	End address	Size
Access by external device via MFI	H'0000	H'07FF	2 Kbytes
Access by on-chip CPU	H'FE2E 0000	H'FE2E 07FF	2 Kbytes

This section describes the 8-bit parallel interface using the MFI. The MFI interface enables access to the 68 and 80 series. Access via the MFI is complete within a fixed time.

27.5.1 68-Series 8-Bit Parallel Interface

Figure 27.4 shows the basic read/write sequence for the 68-series 8-bit parallel interface. MFI access is limited to the period during which the MFI-E/$\overline{\text{WR}}$ signal is driven high and the $\overline{\text{MFI-CS}}$ signal is simultaneously driven low. During this period, a write operation is performed with the MFI-RW/$\overline{\text{RD}}$ signal driven low; a read operation is performed with this signal driven high. The $\overline{\text{MFI-RS}}$ signal indicates whether this is normal access or index/status register access; the low level indicates normal access, and the high level indicates an index/status register access.

For details of the timing, refer to section 33, Electrical Characteristics.

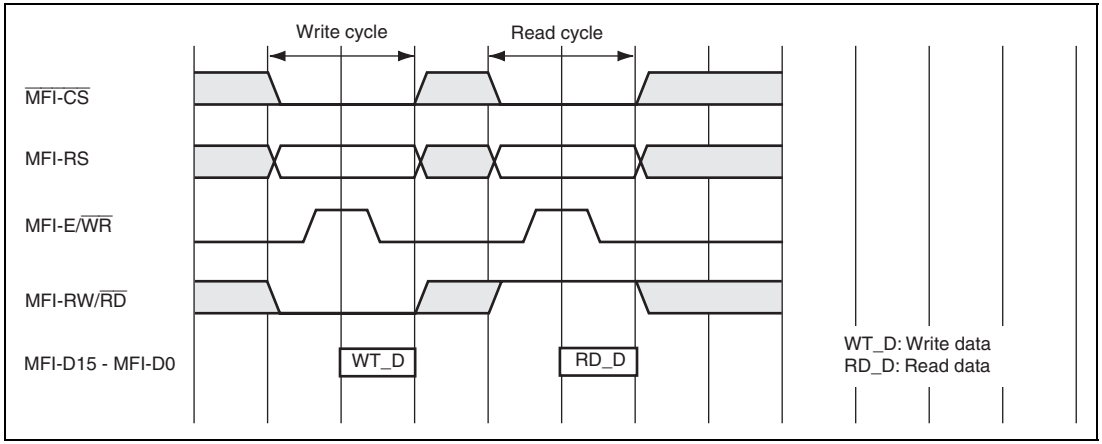


Figure 27.4 Basic Timing for the MFI 68-Series Interface

Figure 27.5 shows the basic read/write sequence for the 80-series 8-bit parallel interface. In the 80-series interface, read operations are limited to the period during which both the MFI-RW/$\overline{\text{RD}}$ and $\overline{\text{MFI-CS}}$ signals are driven low. Write operations are limited to the period during which both the MFI-E/$\overline{\text{WR}}$ and the $\overline{\text{MFI-CS}}$ signals are driven low.

The MFI-RS signal has the same function as in the 68-series interface.

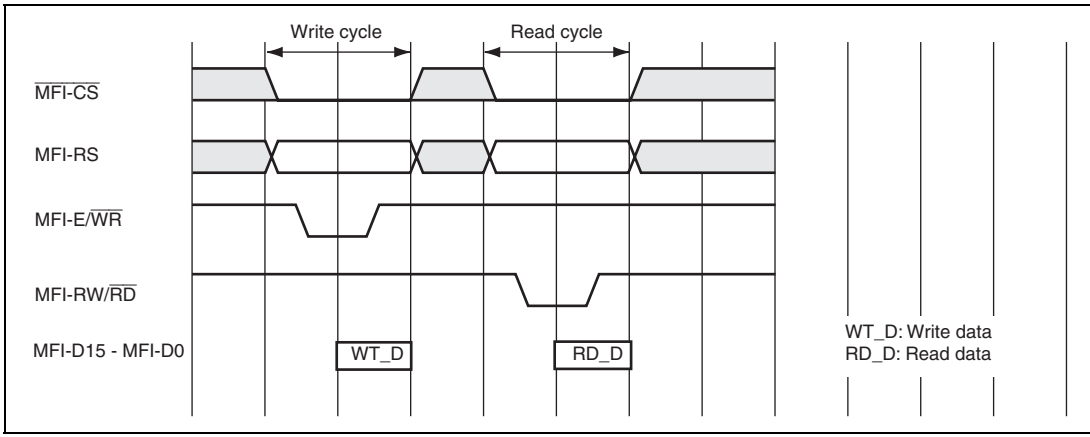


Figure 27.5 Basic Timing of the MFI 80-Series Interface

This section describes the details of the interface, for the case of the 68-series interface.

27.6.1 Writing to MFIIDX/Reading from MFIGSR

Figure 27.6 shows writing to MFIIDX and reading from MFIGSR.

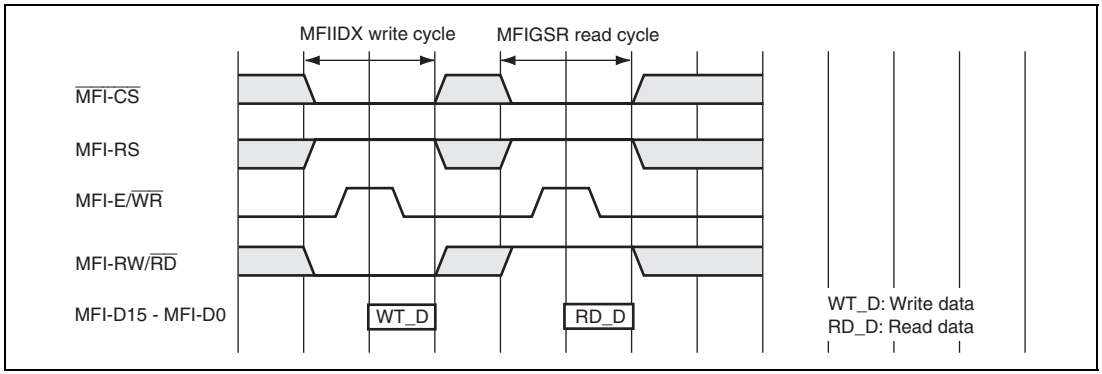


Figure 27.6 Writing to MFIIDX and Reading from MFIGSR

27.6.2 Reading from/Writing to MFI Register

As shown in figure 27.7, to read from/write to the MFI registers, specify a register and a byte position to be accessed by writing to MFIIDX while the MFI-RS is driven high. Then, read from or write to the register specified by MFIIDX with the MFI-RS pin driven low.

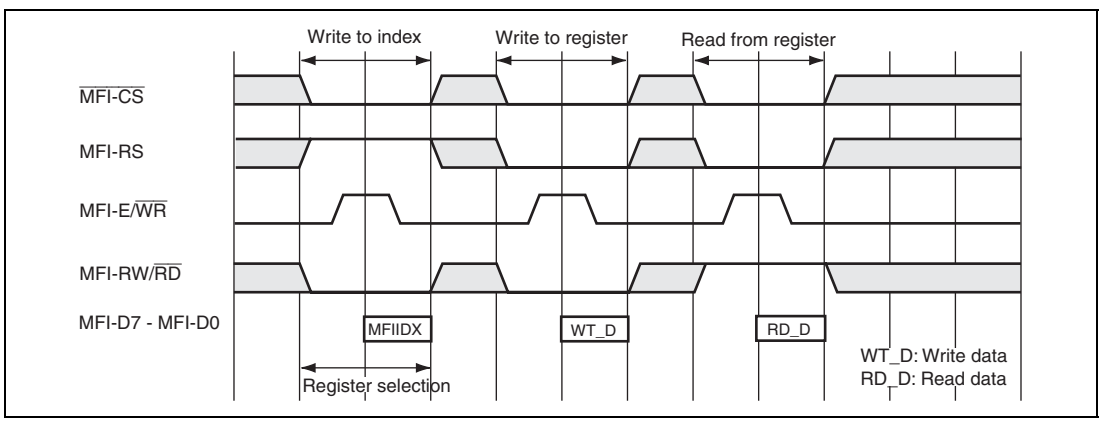


Figure 27.7 MFI Register Settings

Figure 27.8 shows the timing chart for continuous data transfer to the MFRAM via the MFI. As shown in this timing chart, setting the starting address and the data to be written first will enable continuous data transfer subsequently. It provides effective data transfer to this LSI using DMA transfer and others.

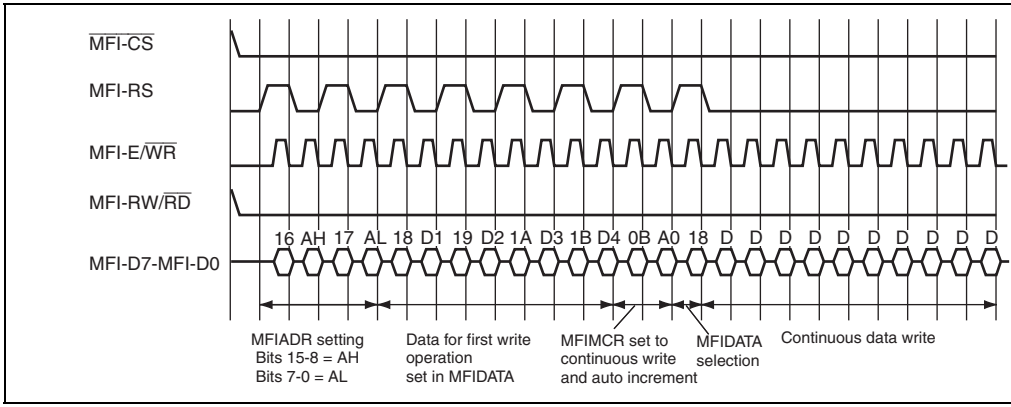


Figure 27.8 Continuous Data Writing to MFRAM (8-Bit Bus Width, MFISCR.SCRMD2 = 0)

27.6.4 Continuous Reading from MFRAM via MFI

Figure 27.9 shows the timing chart for continuous reading from the MFRAM via the MFI. As shown in this timing chart, setting the start address will enable continuous data reading subsequently. It provides effective data transfer from this LSI using DMA transfer and others.

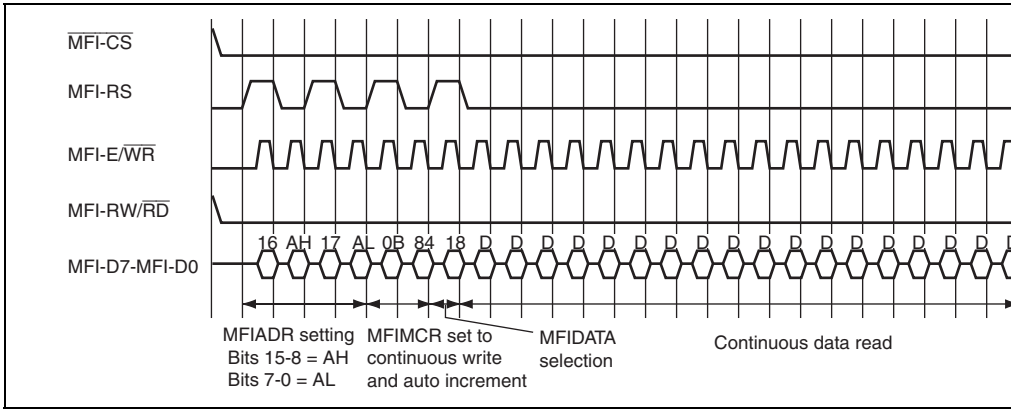


Figure 27.9 Continuous Data Reading from MFRAM (8-Bit Bus Width, MFISCR.SCRMD2 = 0)

The H-UDI is serial input/output interface using the pin functions and transfer protocol compliant with JTAG (IEEE 1149.4: IEEE Standards Test Access Port and Boundary-Scan Architecture) standards.

The H-UDI is also used for emulator connection. Do not use H-UDI functions when using an emulator. Refer to the appropriate emulator manual for the method of connecting the emulator.

The H-UDI consists of six pins: TCK, TMS, TDI, TDO, $\overline{\text{TRST}}$, and $\overline{\text{ASEBRK}}$/BRKACK. The pin functions and serial communication protocol comprise a subset of the JTAG standard. This LSI has additional six pins for emulator connection: (AUDSYNC, AUDCK, and AUDATA[3] to AUDATA[0]). The pins for emulator connection can also be multiplexed for other functions and are assigned to the module specified by the settings of IPSELR in the PFC.

The H-UDI contains two separate TAP controllers, one for controlling the boundary-scan function and another for other functions. Asserting $\overline{\text{TRST}}$, for example at a power-on reset, activates the boundary-scan TAP controller. To use the TAP controller for other functions, input a switchover command to the H-UDI. The CPU has no access to the boundary-scan TAP controller.

Figure 28.1 shows a block diagram of the H-UDI. To initialize the TAP (Test Access Port) controller, control registers and boundary-scan TAP controller, assert $\overline{\text{TRST}}$ active low, or set the TMS pin to 1 and apply TCK for 5 or more cycles. This initialization sequence is independent of the reset pin for this LSI. Other circuits are initialized by the reset pin.

The H-UDI has four registers: SDIR, SDDR (SDDRH and SDDRL), and SDINT. SDBSR configures the JTAG-compliant boundary-scan system, SDIR is used for commands, SDDR is used for data, and SDINT is used for H-UDI interrupts. SDIR is directly accessed from the TDI and TDO pins.

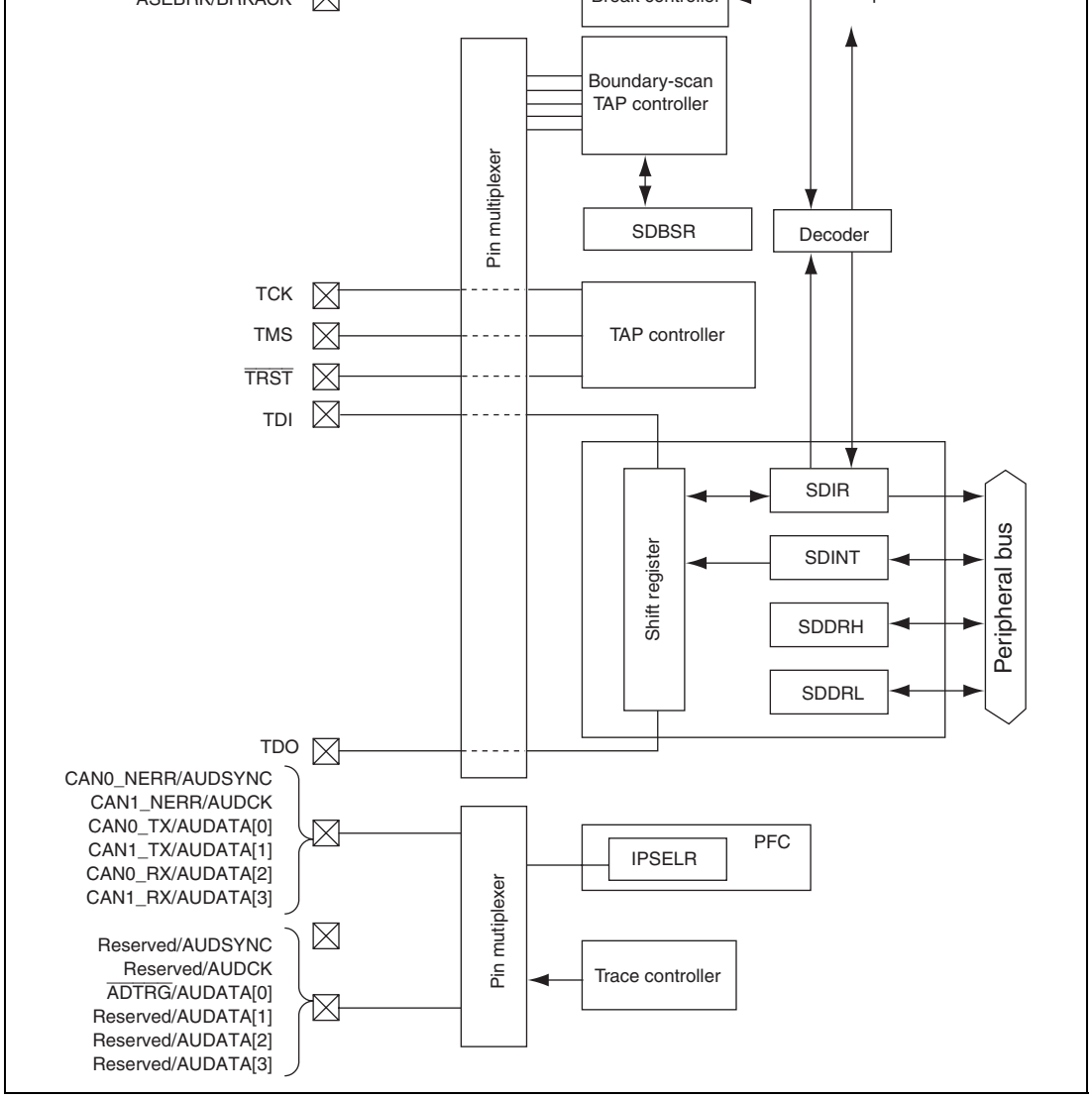


Figure 28.1 H-UDI Block Diagram

Table 28.1 shows the pin configuration for the H-UDI.

Table 28.1 Pin Configuration

Pin Name	Abbreviation	I/O	Function	When Not in Use
Clock	TCK	Input	Functions as the serial clock input pin prescribed in the JTAG standards. Data input to the H-UDI via the TDI pin or data output via the TDO pin is performed in sync with this signal.	Open*1
Mode	TMS	Input	Mode Select Input Changing this signal in sync with a TCK signal determines the significance of data input via the TDI pin. Its protocol conforms to the JTAG standards (IEEE standards 1149.1).	Open*1
Reset	$\overline{\text{TRST}}$	Input	H-UDI Reset Input This signal is received asynchronously with a TCK signal. Asserting this signal low resets the JTAG interface circuit. At power- on, assert the $\overline{\text{TRST}}$ pin low for a given period, whether or not JTAG is used. This differs from the IEEE standards.	Connected to ground *2*3
Data Input	TDI	Input	Data Input Entering this signal in sync with a TCK signal will send data to the H-UDI circuit.	Open*1
Data Output	TDO	Output	Data Output Reading this signal in sync with a TCK signal will read out data from the H-UDI circuit.	Open
Emulator	$\overline{\text{ASEBRK}}$/ BRKACK	Input/ Output	Emulator Connection Only	Open*1

Emulator	AUDSYNC/ AUDCK/ AUDATA[3] to AUDATA[0]	Output	Emulator Connection When bit 13 of IPSELR in the PFC is set to 1, signals are output to the following pins. CAN0_TX/AUDATA[0] CAN1_TX/AUDATA[1] CAN0_RX/AUDATA[2] CAN1_RX/AUDATA[3] CAN0_NERR/AUDCK CAN1_NERR/AUDSYNC When bit 12 of IPSELR in the PFC is set to 1, signals are output to the following pins. $\overline{\text{ADTRG}}$/AUDATA[0] Reserved/AUDATA[1] Reserved/AUDATA[2] Reserved/AUDATA[3] Reserved/AUDCK Reserved/AUDSYNC	Open*4

Notes: 1. This pin is pulled up in this LSI. Using external pull-up resistors will not affect the use of interrupts or resets via the H-UDI or emulators on the board.

2. Design the $\overline{\text{TRST}}$ pin so that it can retain low while the $\overline{\text{RESET}}$ pin is asserted low at a power on reset and can control reset independently, to use interrupts or resets via the H-UDI or emulators on the board.

3. This pin should be connected to ground, the $\overline{\text{RESET}}$, or another pin that operates in the same manner as the $\overline{\text{RESET}}$ pin. However, note that connecting this pin to a ground pin will cause the following problem. Since the $\overline{\text{TRST}}$ pin is pulled up within this LSI, a weak current flows when the pin is externally connected to ground. The value of the current is determined by a resistance of the pull-up MOS for the port pin. Although this current does not affect the operation of this LSI, it consumes unnecessary power.

4. Pull up these pins when not using them as emulator pins and they are not in the output state.

The maximum frequency of a TCK (TMS, TDI, and TDO) signal is 20 MHz, or 2 MHz when boundary scan function is used. Set the TCK clock or the CPG of this LSI so that the frequency of TCK is lower than the frequency of the peripheral clock of this LSI.

and BYPASS)

The H-UDI contains two separate TAP controllers: one for controlling the boundary-scan function and another for controlling the H-UDI reset and interrupt functions. Assertion of $\overline{\text{TRST}}$, for example at power-on reset, activates the boundary-scan TAP controller and enables the boundary-scan function prescribed in the JTAG standards. Executing a switchover command to the H-UDI allows usage of the H-UDI reset and H-UDI interrupts. This LSI, however, has the following limitations:

- Clock-related pins (EXTAL, XTAL, and CKIO) are out of the scope of the boundary-scan test.
- Reset-related pins ($\overline{\text{RESET}}$, $\overline{\text{MRESET}}$, and CA) are out of the scope of the boundary-scan test.
- H-UDI-related pins (TCK, TDI, TDO, TMS, $\overline{\text{TRST}}$, and $\overline{\text{ASEBRK}}$/BRKACK) are out of the scope of the boundary-scan test.
- Analog pins (AN0 to AN3, USB_DM, and USB_DP) are out of the scope of the boundary-scan test.
- I^2C pins (I2C0_SDA, I2C0_SCL, I2C1_SDA, and I2C1_SCL) are out of the scope of the boundary-scan test.
- To perform EXTEST, assert $\overline{\text{MRESET}}$ pin low, negate the $\overline{\text{RESET}}$ pin high and assert the CA pin low. To perform SAMPLE/PRELOAD, assert the CA pin high and negate the $\overline{\text{RESET}}$ pin low.
- To perform the boundary scan (EXTEST, SAMPLE/PRELOAD, or BYPASS), supply a clock signal to the EXTAL pin and perform a power-on reset with the $\overline{\text{RESET}}$ pin. The input clock frequency should be in the range of 1 to 34 MHz. Perform the boundary scan after the power-on oscillation settling time (t_{OSC1}) has elapsed. The supply of a clock signal to the EXTAL pin may be suspended after t_{OSC1} has elapsed. For details of power-on oscillation settling time, see section 33, Electrical Characteristics.
- During the boundary scan (EXTEST, SAMPLE/PRELOAD, and BYPASS), the maximum TCK signal frequency is 2 MHz.
- The external controller has 3-bit access to the boundary-scan TAP controller via the H-UDI.

Table 28.2 shows the commands supported by the boundary-scan TAP controller.

Bit 2	Bit 1	Bit 0	Description
0	0	0	EXTEST
0	0	1	SAMPLE/PRELOAD
0	1	1	H-UDI (switchover command)
Other than above			BYPASS mode

28.2.1 Boundary Scan Register (SDBSR)

SDBSR is a shift register, located on the PAD, for controlling the input/output pins.

Using the EXTEST and SAMPLE/PRELOAD commands, a boundary-scan test complying with the JTAG standards (IEEE1149.1) can be carried out. Table 28.3 shows the correspondence between pins of this LSI and the SDBSR values.

Table 28.3 SDBSR Configuration

Bit	Abbreviation	I/O*	Bit	Abbreviation	I/O*
	From TDI		480	CMT_CTR1	OUT
499	SCIF2_TXD	IN	479	CMT_CTR1	Control
498	SCIF2_TXD	OUT	478	CMT_CTR0/TCLK	IN
497	SCIF2_TXD	Control	477	CMT_CTR0/TCLK	OUT
496	SCIF1_CLK	IN	476	CMT_CTR0/TCLK	Control
495	SCIF1_CLK	OUT	475	STATUS1	IN
494	SCIF1_CLK	Control	474	STATUS1	OUT
493	SCIF2_CLK	IN	473	STATUS1	Control
492	SCIF2_CLK	OUT	472	HSPI_CLK	IN
491	SCIF2_CLK	Control	471	HSPI_CLK	OUT
490	NMI	IN	470	HSPI_CLK	Control
489	NMI	OUT	469	$\overline{\text{HSPI\_CS}}$	IN
488	NMI	Control	468	$\overline{\text{HSPI\_CS}}$	OUT
487	CMT_CTR3	IN	467	$\overline{\text{HSPI\_CS}}$	Control
486	CMT_CTR3	OUT	466	HSPI_TX	IN
485	CMT_CTR3	Control	465	HSPI_TX	OUT
484	CMT_CTR2	IN	464	HSPI_TX	Control
483	CMT_CTR2	OUT	463	HSPI_RX	IN
482	CMT_CTR2	Control	462	HSPI_RX	OUT
481	CMT_CTR1	IN	461	HSPI_RX	Control

460	SSI1_SDATA/HAC_SD_OUT1	IN	423	VCPWC/$\overline{\text{IRQ4}}$	OUT
459	SSI1_SDATA/HAC_SD_OUT1	OUT	422	VCPWC/$\overline{\text{IRQ4}}$	Control
458	SSI1_SDATA/HAC_SD_OUT1	Control	421	VEPWC/$\overline{\text{IRQ5}}$	IN
457	STATUS0	IN	420	VEPWC/$\overline{\text{IRQ5}}$	OUT
456	STATUS0	OUT	419	VEPWC/$\overline{\text{IRQ5}}$	Control
455	STATUS0	Control	418	$\overline{\text{BREQ}}$	IN
454	SSI0_SCK/HAC_SD_IN0/$\overline{\text{BS2}}$	IN	417	$\overline{\text{BREQ}}$	OUT
453	SSI0_SCK/HAC_SD_IN0/$\overline{\text{BS2}}$	OUT	416	$\overline{\text{BREQ}}$	Control
452	SSI0_SCK/HAC_SD_IN0/$\overline{\text{BS2}}$	Control	415	MFI-D8/LCD_DATA8	IN
451	SSI0_WS/HAC_SYNC0	IN	414	MFI-D8/LCD_DATA8	OUT
450	SSI0_WS/HAC_SYNC0	OUT	413	MFI-D8/LCD_DATA8	Control
449	SSI0_WS/HAC_SYNC0	Control	412	MFI-D0/LCD_DATA0	IN
448	SSI0_SDATA/HAC_SD_OUT0	IN	411	MFI-D0/LCD_DATA0	OUT
447	SSI0_SDATA/HAC_SD_OUT0	OUT	410	MFI-D0/LCD_DATA0	Control
446	SSI0_SDATA/HAC_SD_OUT0	Control	409	$\overline{\text{CS0}}$	IN
445	$\overline{\text{HAC\_RES}}$	IN	408	$\overline{\text{CS0}}$	OUT
444	$\overline{\text{HAC\_RES}}$	OUT	407	$\overline{\text{CS0}}$	Control
443	$\overline{\text{HAC\_RES}}$	Control	406	$\overline{\text{BACK}}$	IN
442	$\overline{\text{RDY}}$	IN	405	$\overline{\text{BACK}}$	OUT
441	$\overline{\text{RDY}}$	OUT	404	$\overline{\text{BACK}}$	Control
440	$\overline{\text{RDY}}$	Control	403	MFI-D9/LCD_DATA9	IN
439	HAC_BIT_CLK0	IN	402	MFI-D9/LCD_DATA9	OUT
438	HAC_BIT_CLK0	OUT	401	MFI-D9/LCD_DATA9	Control
437	HAC_BIT_CLK0	Control	400	MFI-D1/LCD_DATA1	IN
436	DCK	IN	399	MFI-D1/LCD_DATA1	OUT
435	DCK	OUT	398	MFI-D1/LCD_DATA1	Control
434	DCK	Control	397	MFI-D10/LCD_DATA10	IN
433	SSI1_SCK/HAC_SD_IN1	IN	396	MFI-D10/LCD_DATA10	OUT
432	SSI1_SCK/HAC_SD_IN1	OUT	395	MFI-D10/LCD_DATA10	Control
431	SSI1_SCK/HAC_SD_IN1	Control	394	MFI-D2/LCD_DATA2/$\overline{\text{IRQ6}}$	IN
430	SSI1_WS/HAC_SYNC1	IN	393	MFI-D2/LCD_DATA2/$\overline{\text{IRQ6}}$	OUT
429	SSI1_WS/HAC_SYNC1	OUT	392	MFI-D2/LCD_DATA2/$\overline{\text{IRQ6}}$	Control
428	SSI1_WS/HAC_SYNC1	Control	391	MFI-D11/LCD_DATA11	IN
427	HAC_BIT_CLK1	IN	390	MFI-D11/LCD_DATA11	OUT
426	HAC_BIT_CLK1	OUT	389	MFI-D11/LCD_DATA11	Control
425	HAC_BIT_CLK1	Control	388	MFI-D3/LCD_DATA3/$\overline{\text{IRQ7}}$	IN
424	VCPWC/$\overline{\text{IRQ4}}$	IN	387	MFI-D3/LCD_DATA3/$\overline{\text{IRQ7}}$	OUT

386	MFI-D5/LCD_DATA5/IRQ7	Control	360	MFI-D14/LCD_DATA14	OUT
385	$\overline{CS1}$	IN	359	MFI-D14/LCD_DATA14	Control
384	$\overline{CS1}$	OUT	358	MFI-D6/LCD_DATA6/$\overline{DREQ3}$	IN
383	$\overline{CS1}$	Control	357	MFI-D6/LCD_DATA6/$\overline{DREQ3}$	OUT
382	$\overline{CS2}$	IN	356	MFI-D6/LCD_DATA6/$\overline{DREQ3}$	Control
381	$\overline{CS2}$	OUT	355	MFI-D15/LCD_DATA15	IN
380	$\overline{CS2}$	Control	354	MFI-D15/LCD_DATA15	OUT
379	MFI-D12/LCD_DATA12	IN	353	MFI-D15/LCD_DATA15	Control
378	MFI-D12/LCD_DATA12	OUT	352	MFI-D7/LCD_DATA7/DRAK3/ DACK3	IN
377	MFI-D12/LCD_DATA12	Control	351	MFI-D7/LCD_DATA7/DRAK3/ DACK3	OUT
376	MFI-D4/LCD_DATA4/$\overline{DREQ2}$	IN	350	MFI-D7/LCD_DATA7/DRAK3/ DACK3	Control
375	MFI-D4/LCD_DATA4/$\overline{DREQ2}$	OUT	349	$\overline{CS5}$	IN
374	MFI-D4/LCD_DATA4/$\overline{DREQ2}$	Control	348	$\overline{CS5}$	OUT
373	MFI-D13/LCD_DATA13	IN	347	$\overline{CS5}$	Control
372	MFI-D13/LCD_DATA13	OUT	346	A21	IN
371	MFI-D13/LCD_DATA13	Control	345	A21	OUT
370	MFI-D5/LCD_DATA5/DRAK2/ DACK2	IN	344	A21	Control
369	MFI-D5/LCD_DATA5/DRAK2/ DACK2	OUT	343	$\overline{MFI-INT}$/LCD_CLK	IN
368	MFI-D5/LCD_DATA5/DRAK2/ DACK2	Control	342	$\overline{MFI-INT}$/LCD_CLK	OUT
367	$\overline{CS4}$	IN	341	$\overline{MFI-INT}$/LCD_CLK	Control
366	$\overline{CS4}$	OUT	340	$\overline{MFI-CS}$/LCD_DON	IN
365	$\overline{CS4}$	Control	339	$\overline{MFI-CS}$/LCD_DON	OUT
364	A20	IN	338	$\overline{MFI-CS}$/LCD_DON	Control
363	A20	OUT	337	MFI-E/LCD_CL1	IN
362	A20	Control	336	MFI-E/LCD_CL1	OUT
361	MFI-D14/LCD_DATA14	IN	335	MFI-E/LCD_CL1	Control

334	MFI-MD/LCD_CL2	IN	297	D2	OUT
333	MFI-MD/LCD_CL2	OUT	296	D2	Control
332	MFI-MD/LCD_CL2	Control	295	D13	IN
331	$\overline{CS6}$	IN	294	D13	OUT
330	$\overline{CS6}$	OUT	293	D13	Control
329	$\overline{CS6}$	Control	292	D12	IN
328	A0	IN	291	D12	OUT
327	A0	OUT	290	D12	Control
326	A0	Control	289	D5	IN
325	MFI-RS/LCD_M_DISP	IN	288	D5	OUT
324	MFI-RS/LCD_M_DISP	OUT	287	D5	Control
323	MFI-RS/LCD_M_DISP	Control	286	D4	IN
322	MFI-RW/LCD_FLM	IN	285	D4	OUT
321	MFI-RW/LCD_FLM	OUT	284	D4	Control
320	MFI-RW/LCD_FLM	Control	283	D9	IN
319	$\overline{BS}$	IN	282	D9	OUT
318	$\overline{BS}$	OUT	281	D9	Control
317	$\overline{BS}$	Control	280	D6	IN
316	A1	IN	279	D6	OUT
315	A1	OUT	278	D6	Control
314	A1	Control	277	D10	IN
313	D0	IN	276	D10	OUT
312	D0	OUT	275	D10	Control
311	D0	Control	274	D11	IN
310	D15	IN	273	D11	OUT
309	D15	OUT	272	D11	Control
308	D15	Control	271	D8	IN
307	D3	IN	270	D8	OUT
306	D3	OUT	269	D8	Control
305	D3	Control	268	D7	IN
304	D1	IN	267	D7	OUT
303	D1	OUT	266	D7	Control
302	D1	Control	265	$\overline{RD}/\overline{CASS}/\overline{FRAME}$	IN
301	D14	IN	264	$\overline{RD}/\overline{CASS}/\overline{FRAME}$	OUT
300	D14	OUT	263	$\overline{RD}/\overline{CASS}/\overline{FRAME}$	Control
299	D14	Control	262	RD/$\overline{WR}$	IN
298	D2	IN	261	RD/$\overline{WR}$	OUT

260	RD/WR	Control	223	A10	IN
259	A2	IN	222	A10	OUT
258	A2	OUT	221	A10	Control
257	A2	Control	220	A11	IN
256	CKE	IN	219	A11	OUT
255	CKE	OUT	218	A11	Control
254	CKE	Control	217	A12	IN
253	$\overline{WE1}$/DQM1	IN	216	A12	OUT
252	$\overline{WE1}$/DQM1	OUT	215	A12	Control
251	$\overline{WE1}$/DQM1	Control	214	A13	IN
250	$\overline{WE0}$/DQM0	IN	213	A13	OUT
249	$\overline{WE0}$/DQM0	OUT	212	A13	Control
248	$\overline{WE0}$/DQM0	Control	211	A14	IN
247	A3	IN	210	A14	OUT
246	A3	OUT	209	A14	Control
245	A3	Control	208	A15	IN
244	A4	IN	207	A15	OUT
243	A4	OUT	206	A15	Control
242	A4	Control	205	A18	IN
241	A5	IN	204	A18	OUT
240	A5	OUT	203	A18	Control
239	A5	Control	202	$\overline{WE3}$/DQM3	IN
238	A6	IN	201	$\overline{WE3}$/DQM3	OUT
237	A6	OUT	200	$\overline{WE3}$/DQM3	Control
236	A6	Control	199	$\overline{WE2}$/DQM2	IN
235	A7	IN	198	$\overline{WE2}$/DQM2	OUT
234	A7	OUT	197	$\overline{WE2}$/DQM2	Control
233	A7	Control	196	$\overline{CS3}$	IN
228	A17	IN	195	$\overline{CS3}$	OUT
231	A17	OUT	194	$\overline{CS3}$	Control
230	A17	Control	193	$\overline{RAS}$	IN
229	A8	IN	192	$\overline{RAS}$	OUT
228	A8	OUT	191	$\overline{RAS}$	Control
227	A8	Control	190	A16	IN
226	A9	IN	189	A16	OUT
225	A9	OUT	188	A16	Control
224	A9	Control	187	A19	IN

186	A19	OUT	149	D17	Control
185	A19	Control	148	D31	IN
184	D23	IN	147	D31	OUT
183	D23	OUT	146	D31	Control
182	D23	Control	145	D16	IN
181	D24	IN	144	D16	OUT
180	D24	OUT	143	D16	Control
179	D24	Control	142	D28	IN
178	D22	IN	141	D28	OUT
177	D22	OUT	140	D28	Control
176	D22	Control	139	D20	IN
175	D25	IN	138	D20	OUT
174	D25	OUT	137	D20	Control
173	D25	Control	136	CAN1_TX/AUDATA[1]	IN
172	D21	IN	135	CAN1_TX/AUDATA[1]	OUT
171	D21	OUT	134	CAN1_TX/AUDATA[1]	Control
170	D21	Control	133	CAN0_TX/AUDATA[0]	IN
169	D26	IN	132	CAN0_TX/AUDATA[0]	OUT
168	D26	OUT	131	CAN0_TX/AUDATA[0]	Control
167	D26	Control	130	A23	IN
166	D27	IN	129	A23	OUT
165	D27	OUT	128	A23	Control
164	D27	Control	127	A22	IN
163	D19	IN	126	A22	OUT
162	D19	OUT	125	A22	Control
161	D19	Control	124	CAN1_RX/AUDATA[3]	IN
160	D18	IN	123	CAN1_RX/AUDATA[3]	OUT
159	D18	OUT	122	CAN1_RX/AUDATA[3]	Control
158	D18	Control	121	CAN0_RX/AUDATA[2]	IN
157	D29	IN	120	CAN0_RX/AUDATA[2]	OUT
156	D29	OUT	119	CAN0_RX/AUDATA[2]	Control
155	D29	Control	118	A25	IN
154	D30	IN	117	A25	OUT
153	D30	OUT	116	A25	Control
152	D30	Control	115	A24	IN
151	D17	IN	114	A24	OUT
150	D17	OUT	113	A24	Control

112	CAN1_NERR/AUDSYNC	IN	75	MD7	IN
111	CAN1_NERR/AUDSYNC	OUT	74	MD7	OUT
110	CAN1_NERR/AUDSYNC	Control	73	MD7	Control
109	CAN0_NERR/AUDCK	IN	72	MD5	IN
108	CAN0_NERR/AUDCK	OUT	71	MD5	OUT
107	CAN0_NERR/AUDCK	Control	70	MD5	Control
106	$\overline{\text{IRL0}}$	IN	69	MD6/$\overline{\text{IOIS16}}$	IN
105	$\overline{\text{IRL0}}$	OUT	68	MD6/$\overline{\text{IOIS16}}$	OUT
104	$\overline{\text{IRL0}}$	Control	67	MD6/$\overline{\text{IOIS16}}$	Control
103	$\overline{\text{IRL1}}$	IN	66	$\overline{\text{DREQ1}}$	IN
102	$\overline{\text{IRL1}}$	OUT	65	$\overline{\text{DREQ1}}$	OUT
101	$\overline{\text{IRL1}}$	Control	64	$\overline{\text{DREQ1}}$	Control
100	$\overline{\text{IRL2}}$	IN	63	$\overline{\text{DREQ0}}$	IN
99	$\overline{\text{IRL2}}$	OUT	62	$\overline{\text{DREQ0}}$	OUT
98	$\overline{\text{IRL2}}$	Control	61	$\overline{\text{DREQ0}}$	Control
97	$\overline{\text{IRL3}}$	IN	60	$\overline{\text{USB\_OVC}}$	IN
96	$\overline{\text{IRL3}}$	OUT	59	$\overline{\text{USB\_OVC}}$	OUT
95	$\overline{\text{IRL3}}$	Control	58	$\overline{\text{USB\_OVC}}$	Control
94	Reserved/AUDATA[1]	IN	57	DACK1	IN
93	Reserved/AUDATA[1]	OUT	56	DACK1	OUT
92	Reserved/AUDATA[1]	Control	55	DACK1	Control
91	$\overline{\text{ADTRG}}$/AUDATA[0]	IN	54	DRAK1	IN
90	$\overline{\text{ADTRG}}$/AUDATA[0]	OUT	53	DRAK1	OUT
89	$\overline{\text{ADTRG}}$/AUDATA[0]	Control	52	DRAK1	Control
88	Reserved/AUDATA[3]	IN	51	UCLK	IN
87	Reserved/AUDATA[3]	OUT	50	UCLK	OUT
86	Reserved/AUDATA[3]	Control	49	UCLK	Control
85	Reserved/AUDATA[2]	IN	48	DACK0	IN
84	Reserved/AUDATA[2]	OUT	47	DACK0	OUT
83	Reserved/AUDATA[2]	Control	46	DACK0	Control
82	Reserved/AUDSYNC	IN	45	USB_PENC	IN
81	Reserved/AUDSYNC	OUT	44	USB_PENC	OUT
80	Reserved/AUDSYNC	Control	43	USB_PENC	Control
79	Reserved/AUDCK	IN	42	MD4/$\overline{\text{CE2B}}$	IN
78	Reserved/AUDCK	OUT	41	MD4/$\overline{\text{CE2B}}$	OUT
77	Reserved/AUDCK	Control	40	MD4/$\overline{\text{CE2B}}$	Control
76	MD8	IN	39	MD3/$\overline{\text{CE2A}}$	IN

38	MD3/$\overline{CE2A}$	OUT	18	$\overline{SCIF2\_RTS}$	IN
37	MD3/$\overline{CE2A}$	Control	17	$\overline{SCIF2\_RTS}$	OUT
36	DRAK0	IN	16	$\overline{SCIF2\_RTS}$	Control
35	DRAK0	OUT	15	$\overline{SCIF1\_CTS}$	IN
34	DRAK0	Control	14	$\overline{SCIF1\_CTS}$	OUT
33	MD1	IN	13	$\overline{SCIF1\_CTS}$	Control
32	SCIF0_TXD	IN	12	$\overline{SCIF2\_CTS}$	IN
31	SCIF0_TXD	OUT	11	$\overline{SCIF2\_CTS}$	OUT
30	SCIF0_TXD	Control	10	$\overline{SCIF2\_CTS}$	Control
29	SCIF0_RXD	IN	9	SCIF1_RXD	IN
28	SCIF0_RXD	OUT	8	SCIF1_RXD	OUT
27	SCIF0_RXD	Control	7	SCIF1_RXD	Control
26	MD2	IN	6	SCIF2_RXD	IN
25	MD0	IN	5	SCIF2_RXD	OUT
24	SCIF0_CLK	IN	4	SCIF2_RXD	Control
23	SCIF0_CLK	OUT	3	SCIF1_TXD	IN
22	SCIF0_CLK	Control	2	SCIF1_TXD	OUT
21	$\overline{SCIF1\_RTS}$	IN	1	SCIF1_TXD	Control
20	$\overline{SCIF1\_RTS}$	OUT		To TDO	
19	$\overline{SCIF1\_RTS}$	Control			

Note: * Control is an active-high signal. When Control is driven high, the corresponding pin is driven according to the OUT value.

The H-UDI has the following registers. For details of the addresses of SDIR, SDDR (SDDRH and SDDRL), and SDINT and the status in each operating mode, see section 32, List of Registers.

Table 28.4 Register Configuration (1)

		CPU Side					
Register Name	**Abbrev.**	**R/W**	**P4 Address**	**Area 7 Address**	**Size**	**Initial Value*1**	**Sync Clock**
Instruction register	SDIR	R	H'FFF0 0000	H'1FF0 0000	16	H'FFFF	Pck
Data register H	SDDR/SDDRH	R/W	H'FFF0 0008	H'1FF0 0008	32/16	Undefined	Pck
Data register L	SDDRL	R/W	H'FFF0 000A	H'1FF0 000A	16	Undefined	Pck
Interrupt source register	SDINT	R/W	H'FFF0 0014	H'1FF0 0014	16	H'000	Pck
Bypass register	SDBPR	—	—	—	—	Undefined	—
Boundary scan register	SDBSR	—	—	—	—	Undefined	—

Table 28.4 Register Configuration (2)

		H-UDI Side			
Register Name	**Abbrev.**	**R/W**	**Size**	**Initial Value*1**	**Sync Clock**
Instruction register	SDIR	R/W	32	H'FFFF FFFD (Fixed value*2)	Pck
Data register H	SDDR/SDDRH	—	—	—	Pck
Data register L	SDDRL	—	—	—	Pck
Interrupt source register	SDINT	W*3	32	H'0000 0000	Pck
Bypass register	SDBPR	R/W	1	Undefined	—
Boundary scan register	SDBSR	R/W	—	Undefined	—

Register Name	Abbrev.	Power-on Reset by $\overline{\text{RESET}}$ Pin/WDT/ H-UDI	Manual Reset by $\overline{\text{RESET}}$ Pin/WDT/ Multiple Exception	Sleep by Sleep Instruction/ Deep Sleep	Standby	
					by Hardware	by Software/ Each Module
Instruction register	SDIR	H'FFFF*4	Retained	Retained	*5	Retained
Data register H	SDDR/SDDRH	Undefined	Retained	Retained		Retained
Data register L	SDDRL	Undefined	Retained	Retained		Retained
Interrupt source register	SDINT	H'0000	Retained	Retained		Retained

Notes: 1. Initialized when the $\overline{\text{TRST}}$ pin is low or the TAP controller is in the Test-Logic-Reset state.
2. Always read as the fixed value (H'FFFF FFFD) from the H-UDI side.
3. Using the H-UDI interrupt command sets the LSB to 1.
4. Reserved bits are read as undefined values. For details, see each of the register descriptions.
5. After exiting hardware standby mode, this LSI enters the power-on reset state by the $\overline{\text{RESET}}$ pin.

SDIR is a 16-bit register that is read only for the CPU. Set a command via the serial input (TDI). SDIR is initialized when $\overline{\text{TRST}}$ is driven low or the TAP controller enters the Test-Logic-Reset state. The H-UDI can write to SDIR 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	7	6	5	4	3	2	1	0
	TI7	TI6	TI5	TI4	TI3	TI2	TI1	TI0	-	-	-	-	-	-	-	-
Initial value:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	TI7 to TI0	All 1	R	Test Instruction Bit 0110xxxx: H-UDI, reset, negate 0111xxxx: H-UDI, reset, assert 101xxxxx: H-UDI interrupt 11111111: Initial value Other than above: Setting prohibited
7 to 0	—	All 1	R	Reserved These bits are always read as 1. The write value should always be 1.

28.3.2 Data Register H and L (SDDRH, SDDRL)

SDDR is a 32-bit register that comprises two 16-bit registers: SDDRH and SDDRL. SDDRH and SDDRL can be read from/written to by the CPU. The register value is not initialized by the CPU reset but is initialized by $\overline{\text{TRST}}$.

- SDDRH

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
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

- SDDRL

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
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

SDINT is a 16-bit register that can be read from/written to by the CPU. Specifying an H-UDI interrupt command in SDIR via H-UDI pin (Update-IR) sets the INTREQ bit to 1. While SDIR contains an H-UDI interrupt command, SDINT is connected between the TDI and TDO pins. SDINT can be read as a 32-bit register; the upper 16 bits will be 0 and the lower 16 bits represent the SDINT register.

The CPU can write 0 alone to the INTREQ bit. As long as this bit is set to 1, an interrupt request will continue to be generated. Therefore, the INTREQ bit must be cleared to 0 during the interrupt handler. The SDINT register is initialized when $\overline{\text{TRST}}$ is driven low or the TAP controller enters the Test-Logic-Reset state.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	INTREQ
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

Bit	Bit Name	Initial Value	R/W	Description
15 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	INTREQ	0	R/W	Interrupt Request Indicates whether or not an interrupt request by an H-UDI interrupt command has occurred. Clearing this bit to 0 by the CPU cancels an interrupt request. Writing 1 to this bit retains the previous value.

28.4.1 TAP Control

Figure 28.2 shows the internal states of TAP controller. The state transitions conform to the JTAG standards.

- State transitions occur according to the TMS value at the rising edge of a TCK signal.
- The TDI value is sampled at the rising edge of a TCK signal and shifted at the falling edge of a TCK signal.
- The TDO value is changed at the falling edge of a TCK signal. The TDO signal is in a Hi-Z state other than in the Shift-DR or Shift-IR state.
- Clearing $\overline{\text{TRST}}$ to 0 places the TAP controller into the Test-Logic-Reset state asynchronously with a TCK signal.

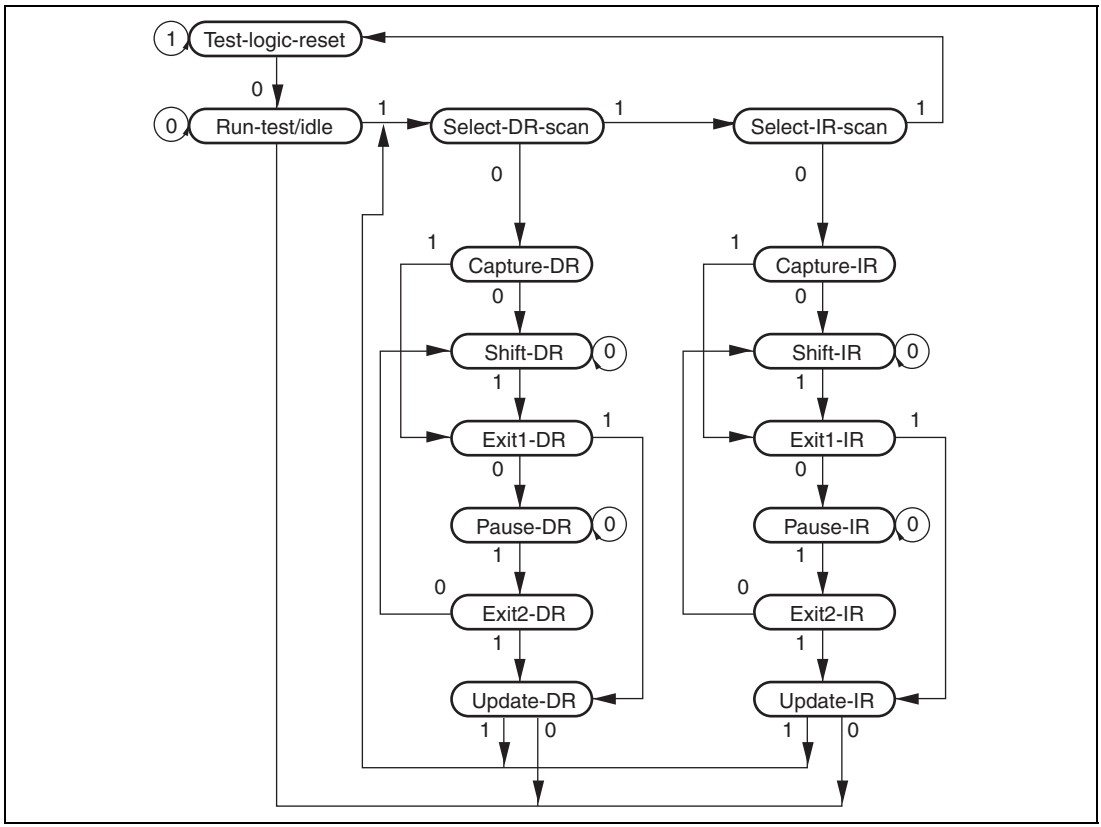


Figure 28.2 TAP Controller State Transitions

A power-on reset can be performed with an SDIR command. Sending an H-UDI reset assert command and then an H-UDI reset negate command from the H-UDI pin resets the H-UDI (see figure 28.3). The required time from the H-UDI reset assert command to H-UDI reset negate command is the same as the time for holding the reset pin low for a power-on reset.

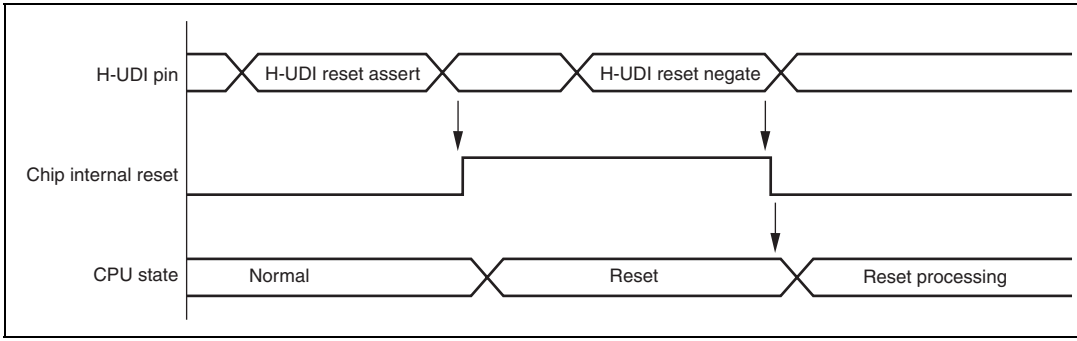


Figure 28.3 H-UDI Reset

28.4.3 H-UDI Interrupt

To generate an interrupt using the H-UDI interrupt function, set the appropriate command in SDIR via the H-UDI. An H-UDI interrupt is a general exception/interrupt operation, resulting in branching to the VBR address. The H-UDI returns from the interrupt handler with a RTE instruction. When an H-UDI interrupt occurs, the exception code H'600 is stored in the interrupt event register (INTEVT). The priority level for the H-UDI interrupt is controlled by bits IPR3 to IPR0 in IPRC.

Specifying the appropriate command (Update-IR) sets the INTREQ bit to 1 and then asserts an H-UDI interrupt request signal. Since this signal is not negated until the INTREQ bit is cleared to 0 by software, the interrupt request will not be missed. While SDIR contains an H-UDI interrupt command, SDINT is connected between the TDI and TDO pins.

- Once an SDIR command is set, it will not be modified unless a $\overline{\text{TRST}}$ signal is asserted, the TAP controller enters the Test-Logic-Reset state, or other command is re-issued via the H-UDI.
- An H-UDI interrupt or an H-UDI reset can be accepted to cancel sleep mode. Neither of these signals can be accepted in standby mode.
- H-UDI functions cannot be used in standby mode. To retain the TAP controller state before and after standby mode, drive a TCK signal high when placing this LSI into standby mode.

  The H-UDI can also be used for emulator connection. Therefore, H-UDI functions cannot be used when an emulator is used.

This LSI includes a 10-bit successive-approximation A/D converter with a selection of up to four analog input channels.

29.1 Features

The A/D converter has the following features.

- 10-bit resolution
- 4 input channels
- High-speed conversion
  - Conversion time: maximum 8 μs per channel (when Pck = 34 MHz with CKSL[1:0] = 01)
- 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 data registers
  - A/D conversion results are transferred for storage into 16-bit data registers corresponding to the channels.
- Sample-and-hold function
- A/D conversion can be externally triggered (except in multi mode)
- 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.
- A/D conversion end signal can be switched
  - Interrupt request or DMAC activation can be selected by the DMASL bit in A/D control/status register (ADCSR).
- Absolute error: ±4 LSB

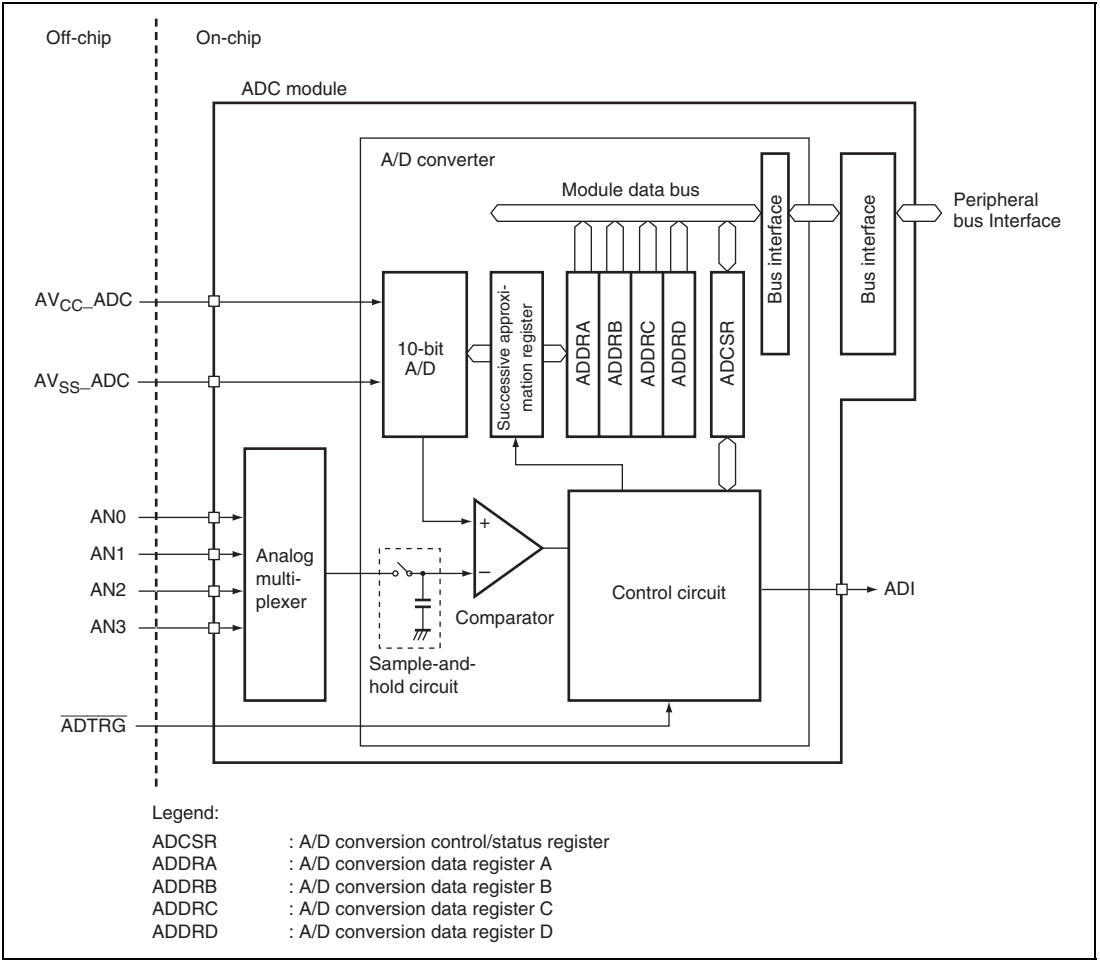


Figure 29.1 A/D Converter Block Diagram

Table 29.1 shows the pin configuration for the A/D converter.

AV_{CC}_ADC and AV_{SS}_ADC are the power supply for the analog circuits in the A/D converter. AV_{CC}_ADC also functions as the A/D converter reference voltage.

Table 29.1 Pin Configuration

Pin Name	Abbreviation	I/O	Function
Analog power-supply	AV_{CC}_ADC	Input	Analog power supply and A/D conversion reference voltage
Analog ground	AV_{SS}_ADC	Input	Analog ground
Analog input pin 0	AN0	Input	Analog input
Analog input pin 1	AN1	Input	Analog input
Analog input pin 2	AN2	Input	Analog input
Analog input pin 3	AN3	Input	Analog input
A/D external trigger input	$\overline{\text{ADTRG}}$	Input	External trigger input for starting A/D conversion

The A/D converter has the following registers. For details of register addresses and register states during each process, see section 32, List of Register.

Table 29.2 Register Configuration (1)

Register Name	Abbrev.	R/W	P4 Address	Area 7 Address	Size	Sync Clock
A/D conversion data register A	ADDRA	R	H'FE28 0000	H'1E28 0000	16	Pck
A/D conversion data register B	ADDRB	R	H'FE28 0002	H'1E28 0002	16	Pck
A/D conversion data register C	ADDRC	R	H'FE28 0004	H'1E28 0004	16	Pck
A/D conversion data register D	ADDRD	R	H'FE28 0006	H'1E28 0006	16	Pck
A/D conversion control/status register	ADCSR	R/W*2	H'FE28 0008	H'1E28 0008	16	Pck

Table 29.2 Register Configuration (2)

Register Name	Abbrev.	Power-on Reset by $\overline{\text{RESET}}$ Pin/WDT/ H-UDI	Manual Reset by $\overline{\text{RESET}}$ Pin/WDT/ Multiple Exception	Standby: Sleep by Sleep Instruction/ by Deep Sleep	Standby: by Hardware	Standby: by Software/ Each Module
A/D conversion data register A	ADDRA	H'0000	H'0000	Retained	*3	H'0000*1
A/D conversion data register B	ADDRB	H'0000	H'0000	Retained		H'0000*1
A/D conversion data register C	ADDRC	H'0000	H'0000	Retained		H'0000*1
A/D conversion data register D	ADDRD	H'0000	H'0000	Retained		H'0000*1
A/D conversion control/status register	ADCSR	H'0040	H'0040	Retained		H'0040*1

Notes: 1. Before entering module standby or software standby mode, check that A/D conversion is not in progress. If standby mode is entered while A/D conversion is in progress, correct register values are not guaranteed.
2. Only 0 can be written to bit 15 for clearing the flag.
3. After exiting hardware standby mode, this LSI enters the power-on reset state by the $\overline{\text{RESET}}$ pin.

ADDR are 16-bit read-only registers that store the results of A/D conversion, comprising 4 registers from A to D.

An A/D conversion produces 10-bit data, which is transferred for storage into the A/D conversion data register corresponding to the selected channel. The upper 8 bits of the result are stored in the upper byte (bits 15 to 8) of the A/D conversion data register. The lower 2 bits are stored in the lower byte (bits 7 and 6). Bits 5 to 0 of an A/D conversion data register are always read as 0.

Table 29.3 indicates the pairings of analog input channels and data registers ADDRA to ADDRD.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0	-	-	-	-	-	-
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	Bit Name	Initial Value	R/W	Description
15 to 6	AD9 to AD0	All 0	R	Bit Data (10 bits)
5 to 0	—	All 0	R	Reserved These bits are always read as 0, and the write value should always be 0.

Table 29.3 Analog Input Channels and Corresponding A/D Conversion Data Registers

Analog Input Channel	A/D Conversion Data Register
AN0	ADDRA
AN1	ADDRB
AN2	ADDRC
AN3	ADDRD

ADCSR is a 16-bit readable/writable register that controls A/D conversion operations and displays the A/D conversion status.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ADF	ADIE	ADST	DMASL	TRGE1	TRGE0	-	-	CKSL1	CKSL0	MDS1	MDS0	-	-	CH1	CH0
Initial value:	0	0	0	0	0	0	0	0	0	1	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/W	R/W	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	ADF	0	R/(W)*	A/D End Flag A status flag that indicates the end of A/D conversion. [Clearing conditions] • When 0 is written after reading ADF = 1 with ADF = 1 • When ADDR is read with DMASL = 1 (DMA transfer) Note: If 1 is written, the previous value is retained. [Setting conditions] • Single mode: A/D conversion ends • Multi mode: A/D conversion has cycled through the selected channels (A/D conversion cycles through the selected channels) • Scan mode: A/D conversion has cycled through the selected channels (A/D conversion is continuously repeated for the selected channels) When operation is stopped during conversion in multi mode or scan mode, the ADF bit is not set.
14	ADIE	0	R/W	A/D Interrupt Enable Enables or disables the interrupt (ADI) requested at the end of A/D conversion. Do not change the ADIE bit setting during A/D conversion. 0: A/D conversion end interrupt (ADI) request is disabled 1: A/D conversion end interrupt (ADI) request is enabled

13	ADST	0	R/W	A/D Start Starts or stops A/D conversion. This bit remains set to 1 during A/D conversion. It can also be set to 1 by external trigger input ($\overline{\text{ADTRG}}$) pin (except in multi mode). 0: A/D conversion is stopped 1: • Single mode: A/D conversion starts. This bit is cleared to 0 automatically when conversion on the specified channel ends. Even when the ADST bit is cleared to 0 (by software), A/D conversion does not stop (0 cannot be written to this bit during A/D conversion). • Multi mode: A/D conversion starts. This bit is cleared to 0 automatically when conversion on the specified channels has been performed for one cycle. When the ADST bit is cleared to 0 (by software), A/D conversion stops when the currently executed channel ends. • Scan mode: A/D conversion starts. A/D conversion continues without a break until the ADST bit is cleared to 0 by software or until all registers are initialized by a power-on or manual reset or in hardware standby, module standby, or software standby mode. For the standby modes, refer to section 29.7.4, Notes on Standby Modes.
12	DMASL	0	R/W	DMAC Select Selects an interrupt or activation of the DMAC due to the end of A/D conversion. Do not change the DMASL bit setting during A/D conversion. 0: An interrupt by the end of A/D conversion is selected 1: Activation of the DMAC by the end of A/D conversion is selected

11 10	TRGE1 TRGE0	0 0	R/W R/W	Trigger Enable External trigger input permits or prohibits A/D conversion. These bits must be set while conversion is stopped. 00: When an external trigger is input, A/D conversion does not start 01: Setting prohibited 10: Setting prohibited 11: A/D conversion starts at the falling edge of an input signal from the external trigger input pin ($\overline{\text{ADTRG}}$) (except in multi mode) Note: Clear bits TRGE1 and TRGE0 to 0 before switching the trigger signal.
9, 8	—	All 0	R	Reserved These bits are always read as 0, and the write value should always be 0.
7 6	CKSL1 CKSL0	0 1	R/W R/W	Clock Select These bits select the A/D conversion clock division ratio. 00: Pck/4 01: Pck/8 10: Pck/16 11: Pck/32 Note: For the Pck and clock division ratio settings, refer to section 29.7.3, Pck and Clock Division Ratio Settings.
5 4	MDS1 MDS0	0 0	R/W R/W	Conversion Mode Select These bits select single mode, multi mode, or scan mode. For details on modes, see section 29.4, Operation. The combination of MDS1 = 0 and MDS0 = 1 should not be selected. 00: Single mode 01: Setting prohibited 10: Multi mode 11: Scan mode
3, 2	—	All 0	R	Reserved These bits are always read as 0, and the write value should always be 0.

1	CH1	0	R/W	Channel Select
0	CH0	0	R/W	These bits select the analog input channels together with the MDS1 bits. Select the input channels after clearing the ADST bit to 0. Single Mode (MDS1 = 0): 00: AN0 01: AN1 10: AN2 11: AN3 Multi Mode or Scan Mode (MDS1 = 1): 00: AN0 01: AN0 and AN1 10: AN0 to AN2 11: AN0 to AN3

Note: * Only 0 can be written for clearing the flag.

29.4 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. To avoid malfunction, switch operating modes while the ADST bit of ADCSR is 0. Changing operating modes and channels and setting the ADST bit can be performed simultaneously.

29.4.1 Single Mode

In single mode, an analog input for the specified channel is converted once as shown below.

1. A/D conversion of the selected channel starts when the ADST bit of ADCSR is set to 1 by software or an external trigger input. The ADST bit holds 1 during A/D conversion and is automatically cleared to 0 when the A/D conversion ends.
2. When A/D conversion ends, the conversion results are transmitted to the A/D conversion data register that corresponds to the channel.
3. When A/D conversion ends, the ADF bit of ADCSR is set to 1. If the ADIE bit in ADCSR is also set to 1, an ADI interrupt is requested at this time.
4. Writing 0 to the ADF bit after reading ADF = 1 clears the ADF bit.

29.2 shows a timing diagram for this example.

1. Select single mode as the operating mode (MDS1 = 0 and MDS0 = 0), AN1 as the input channel (CH1 = 0 and CH0 = 1), and enable A/D interrupt requests (ADIE = 1). Then start A/D conversion (ADST = 1).
2. When A/D conversion is completed, the A/D conversion result is transferred into ADDRB. At the same time, the ADF bit is set to 1, the ADST bit is cleared to 0, and the A/D converter becomes idle.
3. Since ADF = 1, ADIE = 1, and DMASL = 0, an ADI interrupt is generated.
4. The A/D interrupt processing routine starts.
5. The A/D interrupt processing routine reads and processes the A/D conversion result (ADDRB).
6. After reading ADF = 1, write 0 in the ADF bit.
7. Execution of the A/D interrupt processing routine ends. After this, when the ADST bit is set to 1, A/D conversion starts and steps 2 to 7 are repeated.

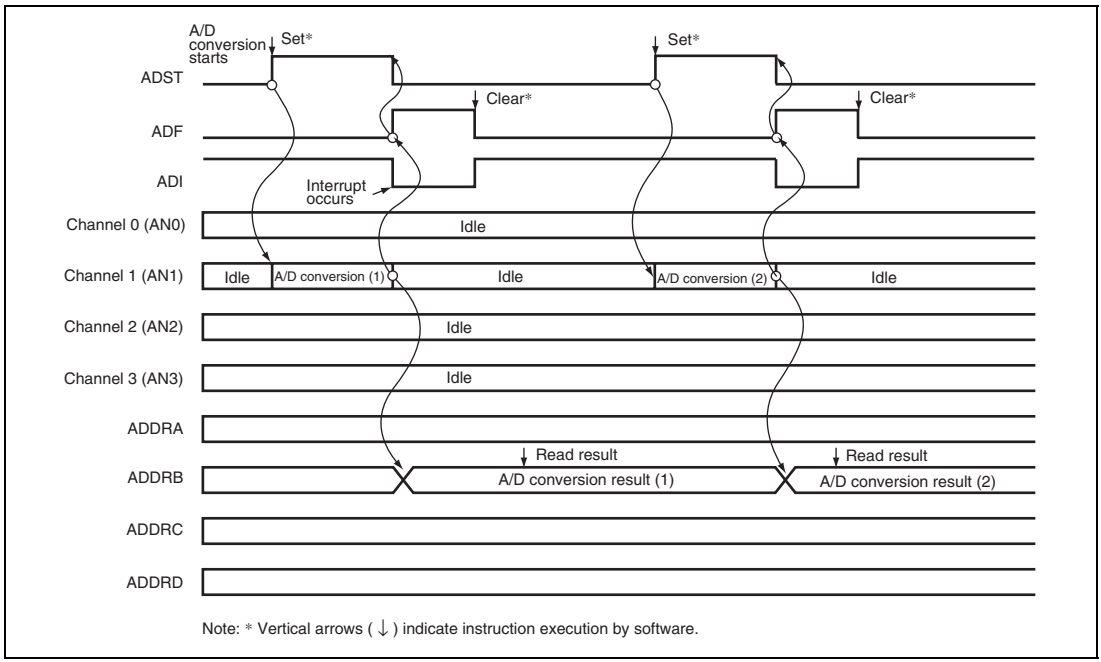


Figure 29.2 Example of A/D Converter Operation (Single Mode, Channel 1 Selected)

In multi mode, analog inputs for the specified channels (one or more) are converted once each as shown below.

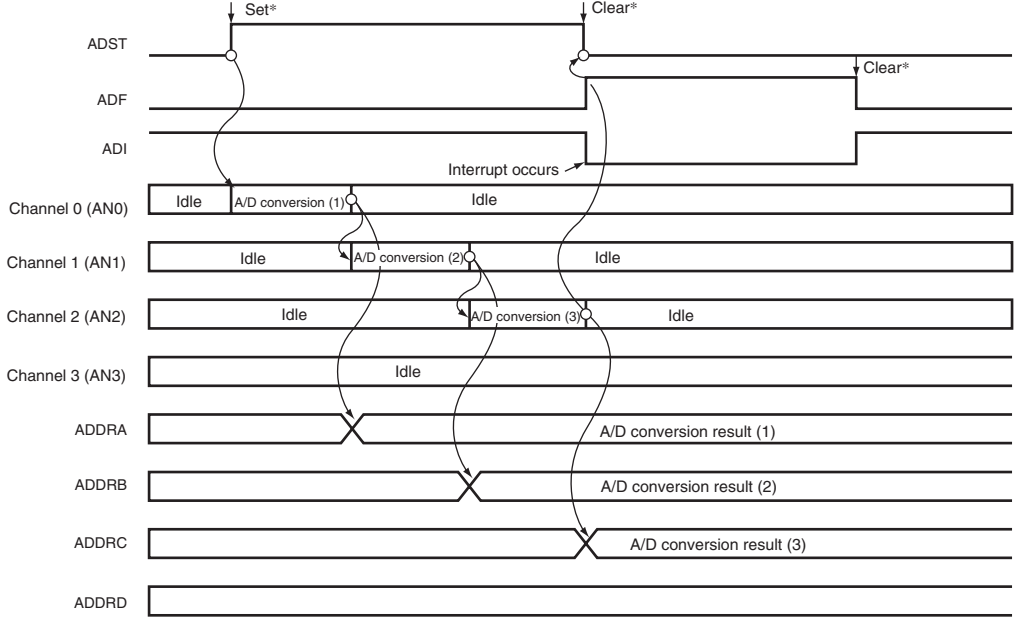
1. When the ADST bit in ADCSR is set to 1 by software, A/D conversion starts with the first channel (AN0).
2. When multiple channels are selected, the input signal for the second channel is converted after the A/D conversion for the first channel ends.
3. When conversion of each channel ends, the conversion results are transmitted to the A/D conversion data register that corresponds to the channel.
4. When conversion of all selected channel ends, the ADF bit of ADCSR is set to 1. If the ADIE bit is also set to 1, an ADI interrupt is requested at this time.
5. When A/D conversion ends, the ADST bit is automatically cleared to 0 and the A/D converter becomes idle. When the ADST bit is cleared to 0 during A/D conversion, the conversion is halted and the A/D converter becomes idle.

   Writing 0 to the ADF bit after reading ADF = 1 clears the ADF bit.

Typical operations when three channels (AN0 to AN2) are selected in multi mode are described below. Figure 29.3 shows a timing diagram for this example.

1. Select multi mode as the operating mode (MDS1 = 1 and MDS0 = 0) and AN0 to AN2 as the analog input channels (CH1 = 1 and CH0 = 0). Then start A/D conversion (ADST = 1).
2. A/D conversion of the first channel (AN0) starts. When the conversion ends, the result is transferred into ADDRA. Next, the second channel (AN1) is selected automatically and A/D conversion starts.
3. Conversion proceeds in the same way up to the third channel (AN2).
4. When conversion of all selected channels (AN0 to AN2) ends, the ADF bit is set to 1 and the ADST bit is cleared to 0 to stop A/D conversion.

   If the DMASL bit is cleared to 0 and the ADIE bit is set to 1 at this time, an ADI interrupt is generated after A/D conversion ends.



Note: * Vertical arrows (↓) indicate instruction execution by software.

Figure 29.3 Example of A/D Converter Operation (Multi Mode, Three Channels AN0 to AN2 Selected)

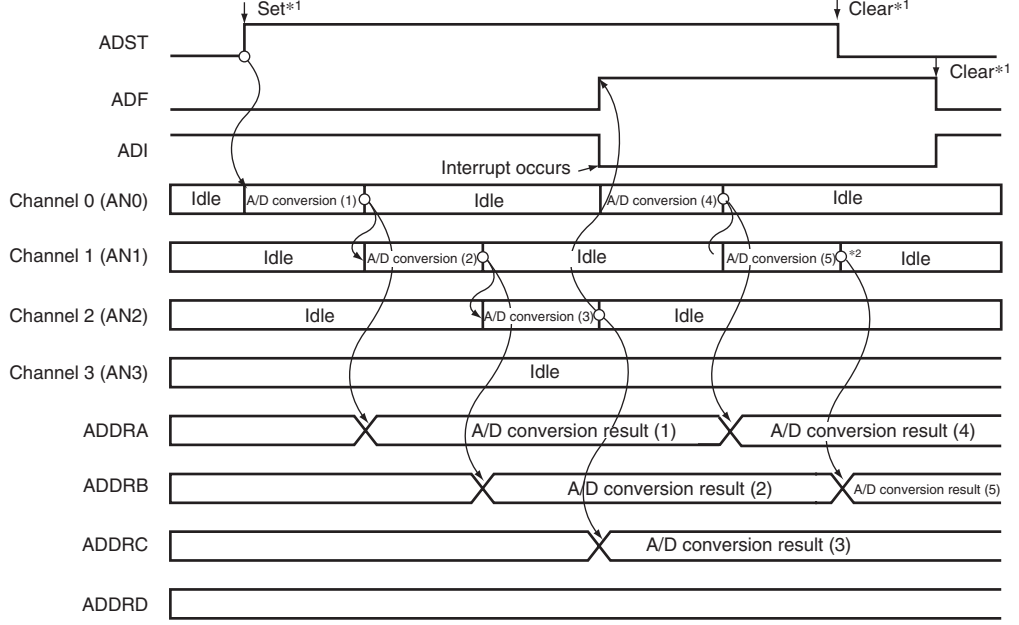
In scan mode, analog inputs for a maximum of four specified channels are converted in succession as shown below.

1. When the ADST bit in ADCSR is set to 1 by software or an external trigger input, A/D conversion starts with the first channel (AN0).
2. A/D conversion for the first channel starts. When multiple channels are selected, the input signal for the second channel (AN1) is converted after the A/D conversion for the first channel ends.
3. When conversion of each channel ends, the conversion results are transmitted to the ADDRA to ADDRD data register that corresponds to the channel.
4. When conversion of all selected channel ends, the ADF bit of ADCSR is set to 1. If the ADIE bit is also set to 1, an ADI interrupt is generated at this time.
5. While the ADST bit is set to 1, it is not automatically cleared, but steps 2 to 4 above are repeated. When the ADST bit is cleared to 0, A/D conversion stops and the A/D converter becomes idle.

Typical operations when three channels (AN0 to AN2) are selected in scan mode are described below. Figure 29.4 shows a timing diagram for this example.

1. Select scan mode as the operating mode (MDS1 = 1 and MDS0 = 1) and AN0 to AN2 as the input channels (CH1 = 1 and CH0 = 0). Then start A/D conversion (ADST = 1).
2. A/D conversion of the first channel (AN0) starts. When the A/D conversion ends, the result is transferred into ADDRA. Next, the second channel (AN1) is selected automatically and A/D conversion starts.
3. Conversion proceeds in the same way up to the third channel (AN2).
4. When conversion of all selected channels (AN0 to AN2) is completed, the ADF bit is set to 1, the first channel (AN0) is selected again, and A/D conversion is consecutively performed. (In multi mode, A/D conversion ends when the selected channels have been cycled through. However, in scan mode, after the selected channels have been cycled through, A/D conversion starts again from the first channel and is consecutively repeated.)

   If the DMASL bit is cleared to 0 and the ADIE bit is set to 1 at this time, an ADI interrupt is generated after A/D conversion ends.
5. While the ADST bit is set to 1, steps 2 to 4 above are repeated. 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 (AN0).



Notes: 1. Vertical arrows (↓) indicate instruction execution by software.
2. Data being converted is ignored.

Figure 29.4 Example of A/D Converter Operation (Scan Mode, Three Channels AN0 to AN2 Selected)

in the corresponding data register is fixed at 256 cycles when Pck/8 is selected as the clock division ratio. Therefore, data is stored in the data registers every 256 cycles.

After an interrupt occurs, data should be read within 256 cycles. After this read, data should be read at the specified intervals.

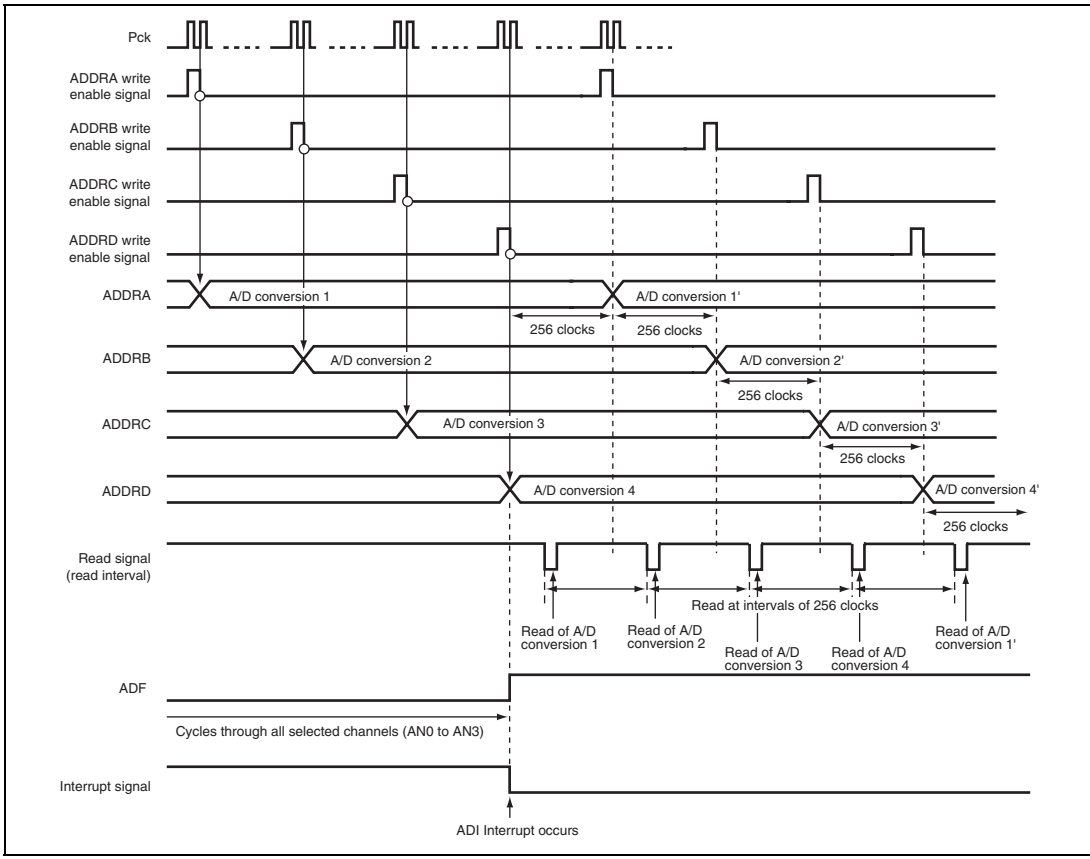


Figure 29.5 Timing for Data Write when Four Channels Are Selected in Multi Mode

Table 29.4 indicates the A/D conversion time.

Table 29.4 A/D Conversion Time

	Pck/4		Pck/8		Pck/16		Pck/32	
Conversion Time Type	**Min**	**Max**	**Min**	**Max**	**Min**	**Max**	**Min**	**Max**
A/D conversion time for the first conversion	131	134	259	266	515	530	1027	1058
A/D conversion time for the second and subsequent conversions (multi mode or scan mode)	—	128	—	256	—	512	—	1024

Notes: 1. To change the channel during A/D conversion in multi mode or scan mode, clear the ADST bit to 0 to stop A/D conversion. Wait for the A/D conversion time shown above before changing the channel.
2. Values in the table are the numbers of states (one state is one peripheral clock (Pck) cycle).
3. The conversion time for the second and subsequent conversions is fixed.

A/D conversion can also be started by an external trigger input (except in multi mode). When the TRGE1 and TRGE0 bits in ADCSR are both set to 1, an external trigger input is enabled at the $\overline{\text{ADTRG}}$ pin.

The ADST bit in ADCSR is set to 1 at the falling edge of the $\overline{\text{ADTRG}}$ pin, thus 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 29.6 shows this timing.

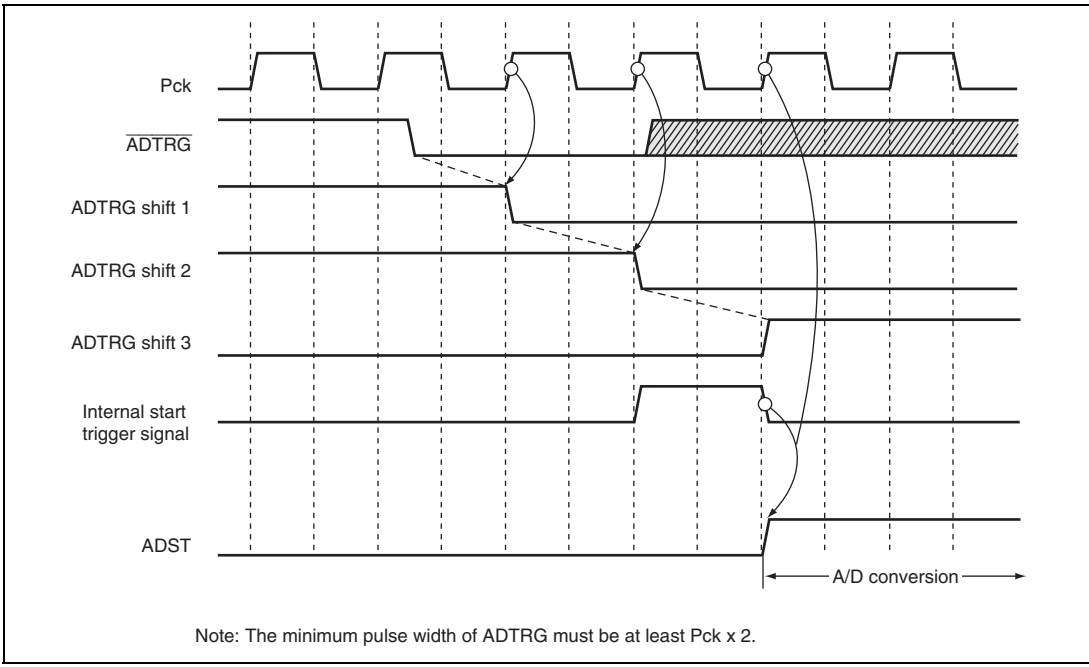


Figure 29.6 External Trigger Input Timing

The A/D converter generates an interrupt (ADI) at the end of A/D conversion.

The ADI interrupt request is enabled/disabled by specifying the ADIE bit in ADCSR. Either interrupt generation or DMAC activation when A/D conversion ends can be selected by the DMASL bit.

Since data for only one channel can be DMA transferred for each interrupt, data for all specified channels can not be DMA transferred when more than one channel is specified in multi or scan mode.

29.6 Definitions of A/D Conversion Accuracy

The A/D converter compares an input for an analog channel to its analog reference voltage 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:

1. Offset error (figure 29.7 (1))

   Deviation between analog input voltage and ideal A/D conversion characteristics when the digital output value changes from the minimum (zero voltage) B'0000000000 (B'000 in figure 29.7) to B'0000000001 (B'001 in figure 29.7)
2. Full-scale error (figure 29.7 (2))

   Deviation between analog input voltage and ideal A/D conversion characteristics when the digital output value changes from the B'1111111110 (B'110 in figure 29.7) to the maximum B'1111111111 (B'111 in figure 29.7).
3. Quantization error (figure 29.7 (3))

   Intrinsic error of the A/D converter and is expressed as 1/2 LSB.
4. Nonlinearity error (figure 29.7 (4))

   Deviation between analog input voltage and ideal A/D conversion characteristics between zero voltage and full-scale voltage. Note that it does not include offset, full-scale, and quantization errors.

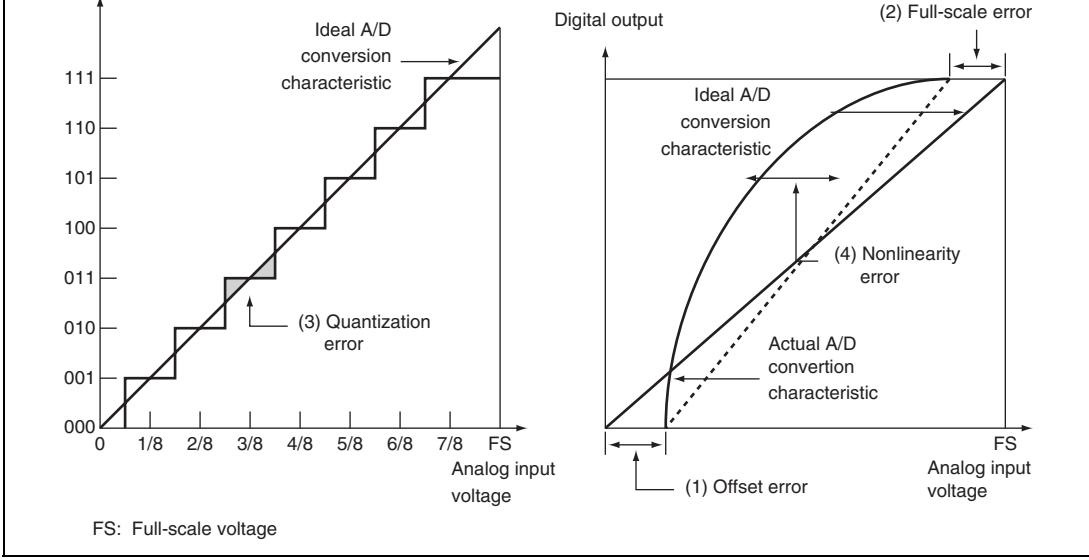


Figure 29.7 Definitions of A/D Conversion Accuracy

29.7 Usage Notes

When using the A/D converter, note the points listed below.

29.7.1 Setting Analog Input Voltage

1. Analog input voltage range

   During A/D conversion, the voltages input to the analog input pins ANn should be in the range AV_{SS}_ADC ≤ ANn ≤ AV_{CC}_ADC (n = 0 to 3).

2. AV_{CC}_ADC and AV_{SS}_ADC input voltages

   AV_{CC}_ADC, AV_{SS}_ADC, and V_{SS} should be related as follows: AV_{CC}_ADC = 3.3 V ± 0.3 V and AV_{SS}_ADC = V_{SS}. (AV_{CC}_ADC = Analog power supply, AV_{SS}_ADC = Analog ground, V_{SS} = Internal digital power supply)

To prevent damage from abnormal voltage such as voltage surges at the analog input pins (AN0 to AN3), connect a protection circuit like the one shown in figure 29.8. The circuit shown also includes a CR filter to suppress noise. This circuit is shown as an example; the circuit constants should be determined according to actual application conditions.

Figure 29.9 shows an equivalent circuit diagram of the analog input pins.

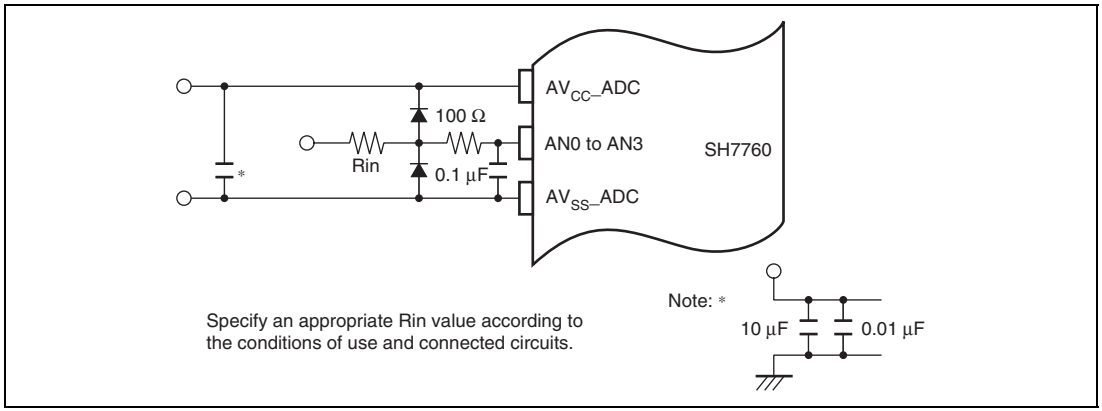


Figure 29.8 Example of Analog Input Pin Protection Circuit

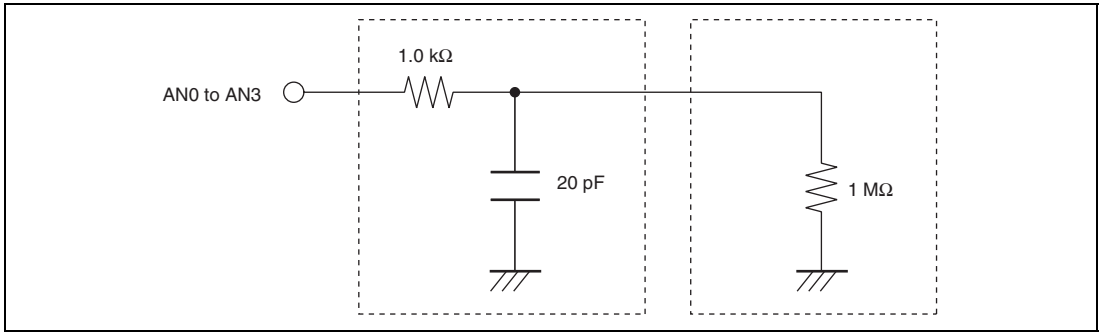


Figure 29.9 Analog Input Pin Equivalent Circuit

Four types of divided clocks can be used as the clock for A/D conversion. Since the internal circuit configuration affects the limits of the interface between the analog and digital sections, be sure to see table 29.5 when setting the input clock and clock division ratio.

Table 29.5 Relationship between Clock Division Ratio and Usable Input Clock Frequency

Clock Division Ratio	Input Clock
Pck/4	18 MHz or lower
Pck/8	34 MHz or lower
Pck/16	34 MHz or lower
Pck/32	34 MHz or lower

29.7.4 Notes on Standby Modes

Before entering hardware standby, module standby, or software standby modes, check that A/D conversion is not in progress, that is, the ADF bit is 1, or in multi mode or scan mode, clear the ADST bit to 0 to stop A/D conversion. Note that A/D conversion does not stop immediately. If module standby or software standby mode is entered without confirming that A/D conversion is stopped, correct AD converter operation is not guaranteed.

In hardware standby, module standby, or software standby mode, all AD converter registers are initialized.

29.7.5 Notice of the DMA transmission of A/D converter

(1) Condition

When DMAC selection bit (DMASL) of ADCSR is set to 1, the value of ADC register read by CPU is unknown. Then, ADC registers can't be read during ADC DMA transmission. But writing register is possible. And while ADC execute DMA transmission, if CPU read MFI register, the value may not be correct. Furthermore, DMA transmitted value of ADC also may not be correct.

(2) Workaround

Please apply one of following 1) or 2) workarounds.

1) Do not read ADC registers by CPU during ADCSR.DMASL = 1.
2) Use ADCSR.DMASL = 0.

(1) Condition

There is the case of stopping or NOT starting of A/D conversion in SCAN mode or MULTI mode, because ADC can not detect the changing of ADST when the period of ADST changing is shorter than the sampling period of the clock cycle selected with CKSL1,0.

For example, the A/D conversion may not be started depending on the setting timing of ADST, even though ADST is set to 1(ADST = 1) to start the A/D conversion again after the A/D start bit (ADCSR.ADST = 0) was cleared to stop conversion during the A/D operation.

(2) Workaround

It is possible to evade by the method of following (a) or (b).

(a) When ADC is used in SCAN mode, or ADC is stopped AD conversion by ADST = 0 during AD operation (ADCSR.ADF = 0) in MULTI mode, please set ADST to 1 after an interval of AD conversion time shown table 29.6 from the time of setting ADST = 0.

Table 29.6 AD Conversion Time

	Pck/4	Pck/8	Pck/16	Pck/32
AD Conversion Time	134	266	530	1058

Unit: cycle

(b) Leave it longer than the 1 cycle of the selected clock with CKSL1, 0 (Table 29.7) of the A/D Control/Status Register (ADCSR) according to Pck in table 29.8 and the setting of the clock dividing frequency.

Table 29.7 ADST Transition Period

	ADCSR.CKSL1, 0			
	2'B00	2'B01	2B'10	2'B11
ADST Transition Period [sec]	4/Pck	8/Pck	16/Pck	32/Pck

Clock Division Ratio	Pck
Pck/4	Less than 18 MHz.
Pck/8	Less than 34 MHz.
Pck/16	Less than 34 MHz.
Pck/32	Less than 34 MHz.

29.7.7 Notice of Multi mode of A/D conversion

(1) Condition

In multi mode of the A/D conversion, it is impossible to start the A/D conversion by the $\overline{\text{ADTRG}}$ input because the A/D start bit (ADCSR.ADST) is not set by the $\overline{\text{ADTRG}}$ trigger input. Use single mode (ADCSR.MDS[1:0] = 00) or scan mode (ADCSR.MDS[1:0] = 11) to start the A/D conversion by the external trigger input to the $\overline{\text{ADTRG}}$ pin, or take the following workround.

(2) Workaround

In the case that the A/D conversion is started by the external trigger inputs in multi mode, use the external interrupt inputs instead of the $\overline{\text{ADTRG}}$ pin input and set the ADCSR.ADST to start A/D conversion in the interrupt handing routine.

Example: A/D conversion start by external trigger input (does not use $\overline{\text{ADTRG}}$) and note

When using the A/D converter in multi mode, use the IRQ, IRL or GPIO interrupt input and set the ADCSR.ADST by the CPU.

Then A/D conversion start timings are delayed with the interrupt response time (refer to table 9.8) from the external interrupt input. Therefore, it is necessary to take the delay of the response time into account for the interrupt signal output timing from the external device to trigger. When the CPU is in power-down modes, the transition time from power-down mode to normal mode is also necessary.

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 palette 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.

30.1 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) *1
- Supports 4/8/15/16-bpp (bits per pixel) color modes
- Supports 1/2/4/6-bpp grayscale modes
- Supports LCD-panel sizes from 16 × 1 to 1024 × 1024*2
- 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 using part of the SDRAM connected to area 3 of the CPU as system memory.
- 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 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 30.4).

Notes: 1. When connecting the LCDC to a TFT panel with an unwired 18-bit bus, the lower bit lines should be connected to GND or to the lowest bit from which data is output.
2. For details, see section 30.4.1, Size of LCD Modules Which Can Be Displayed with this LCDC.

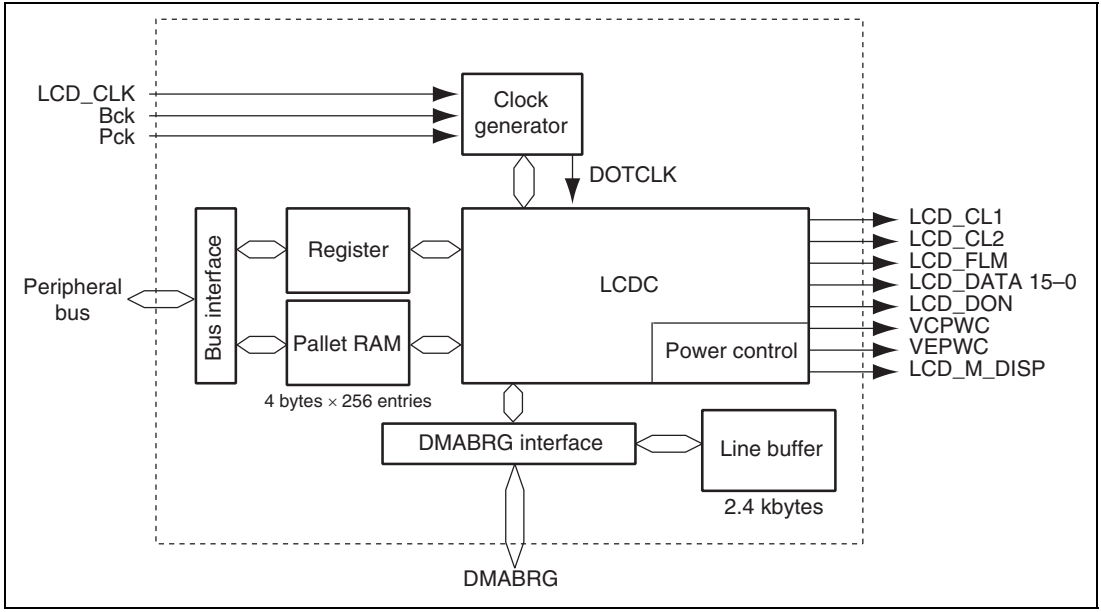


Figure 30.1 LCDC Block Diagram

30.2 Input/Output Pins

Table 30.1 summarizes the LCDC's pin configuration.

Table 30.1 Pin Configuration

Name	I/O	Function
LCD_DATA15 to 0	Output	Data for LCD panel
LCD_DON	Output	Display-on signal (DON)
LCD_CL1	Output	Shift-clock 1 (STN/DSTN)/horizontal sync signal (HSYNC) (TFT)
LCD_CL2	Output	Shift-clock 2 (STN/DSTN)/dot clock (DOTCLK) (TFT)
LCD_M_DISP	Output	LCD current-alternating signal/DISP signal
LCD_FLM	Output	First line marker/vertical sync signal (VSYNC) (TFT)
VCPWC	Output	LCD-module power control (V_{CC})
VEPWC	Output	LCD-module power control (V_{EE})
LCD_CLK	Input	LCD clock-source input

Note: Check the LCD module specifications carefully in section 30.5, Clock and LCD Data Signal Examples, before deciding on the wiring specifications for the LCD module.

The LCDC includes the following registers. For description on the address and processing status of these registers, refer to section 32, List of Registers.

Table 30.2 Register Configuration (1)

Register Name	Abbrev.	R/W	P4 Address	Area 7 Address	Size	Sync Clock	Modification During Display*[2]
LCDC input clock register	LDICKR	R/W	H'FE30 0C00	H'1E30 0C00	16	Pck	No
LCDC module type register	LDMTR	R/W	H'FE30 0C02	H'1E30 0C02	16	Pck	No
LCDC data format register	LDDFR	R/W	H'FE30 0C04	H'1E30 0C04	16	Pck	No
LCDC scan mode register	LDSMR	R/W	H'FE30 0C06	H'1E30 0C06	16	Pck	No
LCDC display start address register - upper	LDSARU	R/W	H'FE30 0C08	H'1E30 0C08	32	Pck	Yes
LCDC display start address register - lower	LDSARL	R/W	H'FE30 0C0C	H'1E30 0C0C	32	Pck	Yes
LCDC display line address offset register	LDLAOR	R/W	H'FE30 0C10	H'1E30 0C10	16	Pck	Yes
LCDC palette control register	LDPALCR	R/W	H'FE30 0C12	H'1E30 0C12	16	Pck	No
Palette data registers 00 to FF	LDPR00 to FF*[1]	R/W	H'FE30 0800	H'1E30 0800	32	Pck	Yes
LCDC horizontal character number register	LDHCNR	R/W	H'FE30 0C14	H'1E30 0C14	16	Pck	No
LCDC horizontal synchronization signal register	LDHSYNR	R/W	H'FE30 0C16	H'1E30 0C16	16	Pck	No
LCDC vertical display line number register	LDVDLNR	R/W	H'FE30 0C18	H'1E30 0C18	16	Pck	No
LCDC vertical total line number register	LDVTLNR	R/W	H'FE30 0C1A	H'1E30 0C1A	16	Pck	No
LCDC vertical synchronization signal register	LDVSYNR	R/W	H'FE30 0C1C	H'1E30 0C1C	16	Pck	No
LCDC AC modulation signal toggle line number register	LDACLNR	R/W	H'FE30 0C1E	H'1E30 0C1E	16	Pck	Yes
LCDC interrupt control register	LDINTR	R/W	H'FE30 0C20	H'1E30 0C20	16	Pck	No
LCDC power management mode register	LDPMMR	R/W	H'FE30 0C24	H'1E30 0C24	16	Pck	No

Register Name	Abbrev.	R/W	P4 Address	Address	Size	Clock	Modification During Display*2
LCDC power supply sequence period register	LDPSPR	R/W	H'FE30 0C26	H'1E30 0C26	16	Pck	No
LCDC control register	LDCNTR	R/W	H'FE30 0C28	H'1E30 0C28	16	Pck	No

Notes: 1. There are 256 registers: LDPR00, LDPR01, ……, LDPRFF. These registers are allocated to H'FE30 0800, H'FE30 0804, ……, H'FE30 0BFC.

2. Modification During Display - Yes: Modification during display on the LCD allowed, No: Modification during display on the LCD not allowed.

Table 30.2 Register Configuration (2)

Register Name	Abbrev.	Power-on Reset by RESET Pin/WDT/ H-UDI	Manual Reset by RESET Pin/WDT/ Multiple Exception	Sleep by Sleep Instruction/ Deep Sleep	Standby by Hardware	Standby by Software/ Each Module
LCDC input clock register	LDICKR	H'0101	H'0101	Retained	*	Retained
LCDC module type register	LDMTR	H'0109	H'0109	Retained		Retained
LCDC data format register	LDDFR	H'000C	H'000C	Retained		Retained
LCDC scan mode register	LDSMR	H'0000	H'0000	Retained		Retained
LCDC display start address register - upper	LDSARU	H'0C00 0000	H'0C00 0000	Retained		Retained
LCDC display start address register - lower	LDSARL	H'0C00 0000	H'0C00 0000	Retained		Retained
LCDC display line address offset register	LDLAOR	H'0280	H'0280	Retained		Retained
LCDC palette control register	LDPALCR	H'0000	H'0000	Retained		Retained
Palette data registers 00 to FF	LDPR00 to FF*1	Undefined	Undefined	Retained		Retained
LCDC horizontal character number register	LDHCNR	H'4F52	H'4F52	Retained		Retained
LCDC horizontal synchronization signal register	LDHSYNR	H'0050	H'0050	Retained		Retained
LCDC vertical display line number register	LDVDLNR	H'01DF	H'01DF	Retained		Retained
LCDC vertical total line number register	LDVTLNR	H'01DF	H'01DF	Retained		Retained
LCDC vertical synchronization signal register	LDVSYNR	H'01DF	H'01DF	Retained		Retained

Register Name	Abbrev.	Reset by RESET Pin/WDT/ H-UDI	Reset by RESET Pin/WDT/ Multiple Exception	Sleep by Sleep Instruction/ Deep Sleep	by Hardware	by Software/ Each Module
LCDC AC modulation signal toggle line number register	LDACLNR	H'000C	H'000C	Retained	*	Retained
LCDC interrupt control register	LDINTR	H'0000	H'0000	Retained		Retained
LCDC power management mode register	LDPMMR	H'0010	H'0010	Retained		Retained
LCDC power supply sequence period register	LDPSPR	H'F60F	H'F60F	Retained		Retained
LCDC control register	LDCNTR	H'0000	H'0000	Retained		Retained

Note: * After exiting hardware standby mode, this LSI enters the power-on reset state caused by the RESET pin.

This LCDC can select bus clock, the peripheral clock, or the external clock as its operation clock source. It is also possible to include a divider of 1/1 to 1/16 in the selected clock and use the division as the LCDC operating clock (DOTCLK). The clock output from the LCDC is used to generate the synchronous clock output (LCD_CL2) for the LCD panel from the operating clock selected in this register. The frequency of the output clock is LCD_CL2 = DOTCLK for a TFT panel, and LCD_CL2 = (DOTCLK/width of data bus output to LCD panel) for an STN or DSTN panel. The LDICKR must be set so that the clock input to the LCDC is 50 MHz or less regardless of the LCD_CL2.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	ICK SEL1	ICK SEL0	-	-	-	-	-	-	-	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

Bit	Bit Name	Initial Value	R/W	Description
15, 14	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
13	ICKSEL1	0	R/W	Input Clock Select Set the clock source for DOTCLK. 00: Bus clock is selected (Bck) 01: Peripheral clock is selected (Pck) 10: External clock is selected (LCD_CLK) 11: Setting prohibited
12	ICKSEL0	0	R/W	
11 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	—	1	R	Reserved This bit is always read as 1. The write value should always be 1.
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	DCDR4	0	R/W	Clock Division Ratio Set the input clock division ratio. For details on the setting, refer to table 30.3.
3	DCDR3	0	R/W	
2	DCDR2	0	R/W	
1	DCDR1	0	R/W	
0	DCDR0	1	R/W	

DCDR[4:0]	Clock Division Ratio	I/O Clock Frequency (MHz)		
		50.000	60.000	66.000
00001	1/1	50.000	60.000	66.000
00010	1/2	25.000	30.000	33.000
00100	1/4	12.500	15.000	16.500
01000	1/8	6.250	7.500	8.250
10000	1/16	3.125	3.750	4.125

Note: Any setting other than above is handled as a clock division ratio of 1/1 (initial value).

30.3.2 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

Bit	Bit Name	Initial Value	R/W	Description
15	FLMPOL	0	R/W	FLM (Vertical Sync Signal) Polarity Select Selects the polarity of the LCD_FLM (vertical sync signal, first line marker) for the LCD module. 0: LCD_FLM pulse is high active 1: LCD_FLM pulse is low active
14	CL1POL	0	R/W	CL1 (Horizontal Sync Signal) Polarity Select Selects the polarity of the LCD_CL1 (horizontal sync signal) for the LCD module. 0: LCD_CL1 pulse is high active 1: LCD_CL1 pulse is low active
13	DISPPOL	0	R/W	DISP (Display Enable) Polarity Select Selects the polarity of the LCD_M_DISP (display enable) for the LCD module. 0: LCD_M_DISP is high active 1: LCD_M_DISP is low active

12	DPOL	0	R/W	Display Data Polarity Select Selects the polarity of the LCD_DATA (display data) for the LCD module. This bit supports reflection of the LCD module. 0: LCD_DATA is high active, transparent-type LCD panel 1: LCD_DATA is low active, reflective-type LCD panel
11	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
10	MCNT	0	R/W	M Signal Control Sets whether or not to output the LCD's current-alternating signal of the LCD module. 0: M (AC line modulation) signal is output 1: M signal is not output
9	CL1CNT	0	R/W	CL1 (Horizontal Sync Signal) Control Sets whether or not to enable LCD_CL1 output during the vertical retrace period. 0: LCD_CL1 is output during vertical retrace period 1: LCD_CL1 is not output during vertical retrace period
8	CL2CNT	1	R/W	LCD_CL2 (Dot Clock of LCD Module) Control Sets whether or not to enable CL2 output during the vertical retrace period. 0: LCD_CL2 is output during vertical retrace period 1: LCD_CL2 is not output during vertical retrace period
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

5	MIFTYP5	0	R/W	Module Interface Type Select
4	MIFTYP4	0	R/W	Set the LCD panel type and data bus width to be 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 more, 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 number of bits. 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 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. 000000: STN monochrome 4-bit data bus module 000001: STN monochrome 8-bit data bus module 001000: STN color 4-bit data bus module 001001: STN color 8-bit data bus module 001010: STN color 12-bit data bus module 001011: STN color 16-bit data bus module 010001: DSTN monochrome 8-bit data bus module 010011: DSTN monochrome 16-bit data bus module 011001: DSTN color 8-bit data bus module 011010: DSTN color 12-bit data bus module 011011: DSTN color 16-bit data bus module 101011: TFT color 16-bit data bus module Settings other than above: Setting prohibited
3	MIFTYP3	1	R/W	
2	MIFTYP2	0	R/W	
1	MIFTYP1	0	R/W	
0	MIFTYP0	1	R/W	

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 COLOR6	DSP COLOR5	DSP COLOR4	DSP COLOR3	DSP COLOR2	DSP COLOR1	DSP COLOR0
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

Bit	Bit Name	Initial Value	R/W	Description
15 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	PABD	0	R/W	Byte Data Pixel Alignment 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 B'0101 which is the normal style handled by a MOV instruction of the this CPU, and should not be selected between B'0101 and B'1010. 0: Big endian for byte data 1: Little endian for byte data
7	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

6	DSPCOLOR6	0	R/W	Display Color Select
5	DSPCOLOR5	0	R/W	Set the number of display colors for the display (0 is written to upper bits of 4 to 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. The number of colors that can be selected in rotation mode is restricted by the display resolution. For details, see table 30.4, Display Resolutions when Using Display Rotation. 0000000: Monochrome, 2 grayscales, 1 bpp (via palette) 0000001: Monochrome, 4 grayscales, 2 bpp (via palette) 0000010: Monochrome, 16 grayscales, 4 bpp (via palette) 0000100: Monochrome, 64 grayscales, 6 bpp (via palette) 0001010: Color, 16 colors, 4 bpp (via palette) 0001100: Color, 256 colors, 8 bpp (via palette) 0011101: Color, 32k colors (RGB: 555), 15 bpp 0101101: Color, 64k colors (RGB: 565), 16 bpp Settings other than above: Setting prohibited
4	DSPCOLOR4	0	R/W	
3	DSPCOLOR3	1	R/W	
2	DSPCOLOR2	1	R/W	
1	DSPCOLOR1	0	R/W	
0	DSPCOLOR0	0	R/W	

LDSMR specifies whether or not to enable the hardware rotation function that is used to rotate the LCD panel. The system memory allocated for display (SDRAM in area 3) is always accessed in units of 32 bytes.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	ROT	-	-	-	-	-	-	-	-	-	-	-	-	-
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	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15, 14	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
13	ROT	0	R/W	Rotation Module Select 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, 256 must be selected 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.) 0: Not rotated 1: Rotated 90 degrees rightwards (left side of image is displayed on the upper side of the LCD module)
12 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

LDSARU sets the start address from which data is fetched by the LCDC for 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	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

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 28	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
27, 26	—	All 1	R	Reserved These bits are always read as 1. The write value should always be 1.
25 to 0	SAU25 to SAU0	All 0	R/W	Start Address for Upper Display Panel Data Fetch The start address for data fetch of the display data must be set within the synchronous DRAM area of area 3.

Notes: 1. The minimum alignment unit of LDSARU is 512 bytes when the hardware rotation function is not used. Write 0 to the lower nine bits. When using the hardware rotation function, 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 lower-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. For the upper left address, calculating backwards using the LDSARU value results in LDSARU − 256 (LDLAOR value) × (320 − 1), so make sure it is set within the 512-byte boundary.
LDSARU = (upper-left address of image) + 256 (LDLAOR value) × 319 (line)

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, or 16 bpp, set the registers so that the start of each line is aligned with the longword boundary (32 bits).

30.3.6 LCDC Display Start Address Register – Lower (LDSARL)

When a DSTN panel is used, LDSARL 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	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

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 28	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
27, 26	—	All 1	R	Reserved These bits are always read as 1. The write value should always be 1.
25 to 0	SAL25 to SAL0	All 0	R/W	Start Address for Lower Panel Display Data Fetch 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

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, or 16 bpp, set the registers so that the start of each line is aligned with the longword boundary (32 bits).

30.3.7 LCDC Display Line Address Offset Register (LDLAOR)

LDLAOR specifies the Y-coordinate increment address width used by the LCDC to read an image recognized by the graphics driver. When the Y-coordinate is increased by 1, this register specifies by how many bytes the address for reading data from memory should be moved, and it needs not conform to the width of the LCD panel. This register corresponds to B when the equation Ax + By + C is used to calculate the memory address of a point (X, Y) in a two dimensional image.

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

Bit	Bit Name	Initial Value	R/W	Description
15 to 10	LAO15 to LAO10	All 0	R/W	Line Address Offset 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. When reading from the register, the lower two bits are always read as 0. In order for VGA (640 × 480 dot) display data to be read continuously without skipping an address between lines, the initial value is set to (× resolution = 640). For details, see table 30.4, Display Resolutions when Using Display Rotation, in section 30.4, Operation. 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, the LDLAOR value should not correspond to the width of the LCD panel (320 in a 320 × 240 panel), but should be a binary exponential (in this example, 256) at least as large as the horizontal width of the image (after rotation, 240 in a 240 × 320 panel).
9	LAO9	1	R/W	
8	LAO8	0	R/W	
7	LAO7	1	R/W	
6 to 0	LAO6 to LAO0	All 0	R/W	

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, color-palette setting 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

Bit	Bit Name	Initial Value	R/W	Description
15 to 5	—	All 0	R	Reserved These bits always read as 0. The write value should always be 0.
4	PALS	0	R	Palette State Indicates the access right state of the palette. 0: Display mode: LCDC uses the palette 1: Color-palette setting mode: The host (CPU) uses the palette
3 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	PALEN	0	R/W	Palette Read/Write Enable Requests the access right to the palette. 0: Request for transition to normal display mode 1: Request for transition to color palette setting mode

LDPR 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 30.4.3, Color Palette Specification.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	PALD nn_23	PALD nn_22	PALD nn_21	PALD nn_20	PALD nn_19	PALD nn_18	PALD nn_17	PALD nn_16
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
	PALD nn_15	PALD nn_14	PALD nn_13	PALD nn_12	PALD nn_11	PALD nn_10	PALD nn_9	PALD nn_8	PALD nn_7	PALD nn_6	PALD nn_5	PALD nn_4	PALD nn_3	PALD nn_2	PALD nn_1	PALD nn_0
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

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	—	—	R	Reserved
23 to 0	PALDnn_23 to PALDnn_0	—	R/W	Palette Data Bits 18 to 16, 9, 8, and 2 to 0 are reserved within each RGB palette and cannot be set. However, these bits can be used by extension, according to the value of the upper bits.

Note: nn = H'00 to H'FF

LDHCNR specifies the LCD module's horizontal size (in the scan direction) and the width of the entire scan, including that of 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	1	0	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

Bit	Bit Name	Initial Value	R/W	Description
15	HDCN7	0	R/W	Horizontal Display Character Number
14	HDCN6	1	R/W	Sets the number of horizontal display characters (unit: character = 8 dots).
13	HDCN5	0	R/W	
12	HDCN4	0	R/W	Specify a value of (number of display characters) −1.
11	HDCN3	1	R/W	
10	HDCN2	1	R/W	Example: For a LCD module with a width of 640 pixels.
9	HDCN1	1	R/W	HDCN = (640/8) −1 = 79 = H'4F
8	HDCN0	1	R/W	
7	HTCN7	0	R/W	Horizontal Total Character Number
6	HTCN6	1	R/W	Sets the number of total horizontal characters (unit: character = 8 dots).
5	HTCN5	0	R/W	
4	HTCN4	1	R/W	Specify a value of (total number of characters) −1.
3	HTCN3	0	R/W	
2	HTCN2	0	R/W	However, the minimum value for the horizontal retrace period is three characters (24 dots).
1	HTCN1	1	R/W	
0	HTCN0	0	R/W	Example: For a LCD module with a width of 640 pixels. HTCN = [(640/8) −1] +3 = 82 = H'52 In this case, the number of total horizontal dots is 664 dots and the number for the horizontal retrace period is 24 dots.

Note: The values set in HDCN and HTCN must satisfy the relationship of HTCN ≥ HDCN + 3. Also, the total number of characters of HTCN must be an even number. (The set value will be an odd number, as it is one less than the actual number.)

LDHSYNR specifies the timing of the generation of the horizontal (scan direction) sync signals 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	Bit Name	Initial Value	R/W	Description
15 14 13 12	HSYNW3 HSYNW2 HSYNW1 HSYNW0	0 0 0 0	R/W R/W R/W R/W	Horizontal Sync Signal Width Sets the width of the horizontal sync signals (CL1 and Hsync) (unit: character = 8 dots). Specify a value of (width of horizontal sync signal) −1. Example: For a horizontal sync signal width of 8 dots. HSYNW = (8 dots/8 dots/character) −1 = 0 = H'0
11 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7 6 5 4 3 2 1 0	HSYNP7 HSYNP6 HSYNP5 HSYNP4 HSYNP3 HSYNP2 HSYNP1 HSYNP0	0 1 0 1 0 0 0 0	R/W R/W R/W R/W R/W R/W R/W R/W	Horizontal Sync Signal Output Position Sets the output position of the horizontal sync signals (unit: character = 8 dots). Specify a value of (position of horizontal sync signal output) −1. Example: For a 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.

Note: The following conditions must be satisfied:
HTCN ≥ HSYNP+HSYNW+1
HSYNP ≥ HDCN+1

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 upper and lower 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

Bit	Bit Name	Initial Value	R/W	Description
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10	VDLN10	0	R/W	Vertical Display Line Number Sets the number of vertical display lines (unit: line). Specify a value of (number of display lines) –1. Example: For an 480–line LCD module VDLN = 480–1 = 479 = H'1DF
9	VDLN9	0	R/W	
8	VDLN8	1	R/W	
7	VDLN7	1	R/W	
6	VDLN6	1	R/W	
5	VDLN5	0	R/W	
4	VDLN4	1	R/W	
3	VDLN3	1	R/W	
2	VDLN2	1	R/W	
1	VDLN1	1	R/W	
0	VDLN0	1	R/W	

LDVTLNR specifies the LCD panel module's entire vertical length, including that of the vertical retrace period.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	VTLN10	VTLN9	VTLN8	VTLN7	VTLN6	VTLN5	VTLN4	VTLN3	VTLN2	VTLN1	VTLN0
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

Bit	Bit Name	Initial Value	R/W	Description
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10	VTLN10	0	R/W	Vertical Total Line Number Sets the total number of vertical display lines (unit: line). Specify a value of (total number of lines) –1. The minimum for the total number of vertical lines is 2 lines. The following conditions must be satisfied: VTLN>=VDLN, VTLN>=1. Example: For a 480-line LCD module and a vertical retrace period of 0 lines. VTLN = (480+0) –1 = 479 = H'1DF
9	VTLN9	0	R/W	
8	VTLN8	1	R/W	
7	VTLN7	1	R/W	
6	VTLN6	1	R/W	
5	VTLN5	0	R/W	
4	VTLN4	1	R/W	
3	VTLN3	1	R/W	
2	VTLN2	1	R/W	
1	VTLN1	1	R/W	
0	VTLN0	1	R/W	

LDVSYNR specifies the vertical (scan direction and vertical direction) sync signal timing of 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	Bit Name	Initial Value	R/W	Description
15	VSYNW3	0	R/W	Vertical Sync Signal Width
14	VSYNW2	0	R/W	Sets the width of the vertical sync signals (FLM and Vsync) (unit: line).
13	VSYNW1	0	R/W	Specify a value of (width of vertical sync signal) −1.
12	VSYNW0	0	R/W	Example: For a vertical sync signal width of 1 line. VSYNW = (1−1) = 0 = H'0
11	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
10	VSYNP10	0	R/W	Vertical Sync Signal Output Position
9	VSYNP9	0	R/W	Sets the output position of the vertical sync signals (FLM and Vsync) (unit: line).
8	VSYNP8	1	R/W	Specify a value of (position of vertical sync signal output) −2.
7	VSYNP7	1	R/W	DSTN should be set to an odd number value. It is handled as (setting value+1)/2.
6	VSYNP6	1	R/W	Example: For a 480-line LCD module and a retrace period of 0 lines (in other words, VTLN= 479 and the vertical sync signal is active for the first line):
5	VSYNP5	0	R/W	• Single display VSYNP = [(1−1)+VTLN]mod(VTLN+1) = [(1−1)+479]mod(479+1) = 479mod480 = 479 =H'1DF
4	VSYNP4	1	R/W	• Dual address VSYNP = [(1−1)×2+VTLN]mod(VTLN+1) = [(1−1)×2+479]mod(479+1) = 479mod480 = 479 =H'1DF
3	VSYNP3	1	R/W	
2	VSYNP2	1	R/W	
1	VSYNP1	1	R/W	
0	VSYNP0	1	R/W	

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

Bit	Bit Name	Initial Value	R/W	Description
15 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	ACLN4	0	R/W	AC Line Number Sets the number of lines where the LCD current-alternating signal of the LCD module is toggled (unit: line). Specify a value of (number of toggled lines) –1. Example: For toggling every 13 lines. ACLN = 13–1 = 12= H'0C
3	ACLN3	1	R/W	
2	ACLN2	1	R/W	
1	ACLN1	0	R/W	
0	ACLN0	0	R/W	

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.

30.3.16 LCDC Interrupt Control Register (LDINTR)

LDINTR regulates the LCD module's Vsync interrupt (LCDCI) activity.

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

Bit	Bit Name	Initial Value	R/W	Description
15 to 13	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

12	VINTSEL	0	R/W	Vsync Interrupt Select Sets the starting point of the LCDC's Vsync interrupt. 0: Vsync interrupt occurs at the beginning of access to synchronous DRAM 1: Vsync interrupt occurs at the beginning of the LCD display vertical retrace period
11 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	VINTE	0	R/W	Vsync Interrupt Enable Sets whether or not to generate LCDC's Vsync interrupts. 0: Vsync interrupts are disabled 1: Vsync interrupts are enabled
7 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	VINTS	0	R/W	Vsync Interrupt State Indicates the LCDC's Vsync interrupt handling state. This bit is set to 1 at the time a Vsync interrupt is generated. During the processing routine for Vsync interrupt, clear the register by entering a value of 0. 0: LCDC did not generate a Vsync interrupt or has been informed that the generated Vsync interrupt has completed 1: LCDC has generated a Vsync interrupt and has not yet been informed that the generated Vsync interrupt has completed 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. When the VINTE bit is set to 0, Vsync interrupts are not generated.

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 function 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

Bit	Bit Name	Initial Value	R/W	Description
15 14 13 12	ONC3 ONC2 ONC1 ONC0	0 0 0 0	R/W R/W R/W R/W	LCDC Power-On Sequence Period Sets the period from VEPWC assertion to LCD_DON assertion in the power-on sequence of the LCD module in frame units. Specify a value of (the period) −1. This period is the (c) period in figures 30.4 to 30.7, Power-Supply Control Sequence and States of the LCD Module. For details on setting this register, see table 30.5, Available Power-Supply Control-Sequence Periods at Typical Frame Rates. (The setting method is common for ONA, ONB, OFFD, OFFE, and OFFF.)
11 10 9 8	OFFD3 OFFD2 OFFD1 OFFD0	0 0 0 0	R/W R/W R/W R/W	LCDC Power-Off Sequence Period Sets the period from LCD_DON negation to VEPWC negation in the power-off sequence of the LCD module in frame units. Specify a value of (the period) −1. This period is the (d) period in figures 30.4 to 30.7, Power-Supply Control Sequence and States of the LCD Module.
7	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
6	VCPE	0	R/W	VCPWC Pin Enable Sets whether or not to enable a power-supply control sequence using the VCPWC pin. 0: Disabled: VCPWC pin output is masked and fixed low 1: Enabled: VCPWC pin output is asserted and negated according to the power-on or power-off sequence

5	VEPE	0	R/W	VEPWC Pin Enable Sets whether or not to enable a power-supply control sequence using the VEPWC pin. 0: Disabled: VEPWC pin output is masked and fixed low 1: Enabled: VEPWC pin output is asserted and negated according to the power-on or power-off sequence
4	DONE	1	R/W	LCD_DON Pin Enable Sets whether or not to enable a power-supply control sequence using the LCD_DON pin. 0: Disabled: LCD_DON pin output is masked and fixed low 1: Enabled: LCD_DON pin output is asserted and negated according to the power-on or power-off sequence
3, 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1 0	LPS1 LPS0	0 0	R R	LCD Module Power-Supply Input State Indicates the power-supply input state of the LCD module when using the power-supply control function. 0: LCD module power off 1: LCD module power on

LDPSPR controls the power supply circuit that provides power to the LCD module. It sets the timing for beginning output to the VEPWC and VCPWC pins and for the timing signals which accompany them.

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

Bit	Bit Name	Initial Value	R/W	Description
15	ONA3	1	R/W	LCDC Power-On Sequence Period
14	ONA2	1	R/W	Sets the period from VCPWC assertion to starting output of the display data (LCD_DATA) and timing signals (LCD_FLM, LCD_CL1, LCD_CL2, and LCD_M_DISP) in the power-on sequence of the LCD module in frame units.
13	ONA1	1	R/W	
12	ONA0	1	R/W	Specify a value of (the period) −1.
				This period is the (a) period in figures 30.4 to 30.7, Power-Supply Control Sequence and States of the LCD Module.
11	ONB3	0	R/W	LCDC Power-On Sequence Period
10	ONB2	1	R/W	Sets the period from starting output of the display data (LCD_DATA) and timing signals (LCD_FLM, LCD_CL1, LCD_CL2, and LCD_M_DISP) to the VEPWC assertion in the power-on sequence of the LCD module in frame units.
9	ONB1	1	R/W	
8	ONB0	0	R/W	Specify a value of (the period) −1.
				This period is the (b) period in figures 30.4 to 30.7, Power-Supply Control Sequence and States of the LCD Module.

7	OFFE3	0	R/W	LCDC Power-Off Sequence Period
6	OFFE2	0	R/W	Sets the period from VEPWC negation to stopping output of the display data (LCD_DATA) and timing signals (LCD_FLM, LCD_CL1, LCD_CL2, and LCD_M_DISP) in the power-off sequence of the LCD module in frame units. Specify a value of (the period) –1. This period is the (e) period in figures 30.4 to 30.7, Power-Supply Control Sequence and States of the LCD Module.
5	OFFE1	0	R/W	
4	OFFE0	0	R/W	
3	OFFF3	1	R/W	LCDC Power-Off Sequence Period Sets the period from stopping output of the display data (LCD_DATA) and timing signals (LCD_FLM, LCD_CL1, LCD_CL2, and LCD_M_DISP) to VCPWC negation in the power-off sequence of the LCD module in frame units. Specify a value of (the period) –1. This period is the (f) period in figures 30.4 to 30.7, Power-Supply Control Sequence and States of the LCD Module.
2	OFFF2	1	R/W	
1	OFFF1	1	R/W	
0	OFFF0	1	R/W	

30.3.19 LCDC Control Register (LDCNTR)

LDCNTR specifies start and stop of display by the LCDC.

The LCDC begins display when a value of 1 is input to both the DON2 bit and the DON bit. Power is then supplied to the LCD module in accordance with the sequence set by the LDPMM and LDCNTR. The sequence ends when the LPS[1:0] value changes from B'00 to B'11. Do not make any action to the DON bit until the sequence ends.

The LCDC stops display when a value of 0 is input to the DON bit. Power to the LCD module is cut off in accordance with the sequence set by the LDPMMR and LDCNTR. The sequence ends when the LPS[1:0] value changes from B'11 to B'00. Do not make any action to the DON bit until the sequence ends.

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

15 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	DON2	0	R/W	Display On 2 Specifies the start of the LCDC display operation. 0: LCDC is being operated or stopped 1: LCDC starts operation This bit is always read as 0. Input 1 only when starting display. If 1 is input at any time other than when starting display, it is not certain to function. The 1 that is input automatically reverts to 0, so it is unnecessary to clear it by inputting 0.
3 to 1	—	All 0	R	Reserved. These bits are always read as 0. The write value should always be 0.
0	DON	0	R/W	Display On Specifies the start and stop of the LCDC display operation. The control sequence state can be checked by referencing the LPS[1:0] of LDPMMR. 0: Display-off mode: LCDC is stopped 1: Display-on mode: LCDC operates

Notes: 1. Write H'0011 to LDCNTR when starting display and H'0000 when completing display. Data other than H'0011 and H'0000 must not be written to.

2. Setting bit DON2 to 1 makes the contents of the palette RAM undefined. Before writing to the palette RAM, set bit DON2 to 1.

30.4.1 Size of LCD Modules Which Can Be Displayed with 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 before display.

This LSI has a maximum 32-byte burst memory read operation and a 2.4-kbyte line buffer, so display failure is unlikely. However, there may be some display problems with certain configurations. A recommended size at the frame rate of 60 Hz is 320 × 240 dots in 16 bpp or 640 × 480 dots in 8-bpp.

Figure 30.2 shows the valid display and the retrace period.

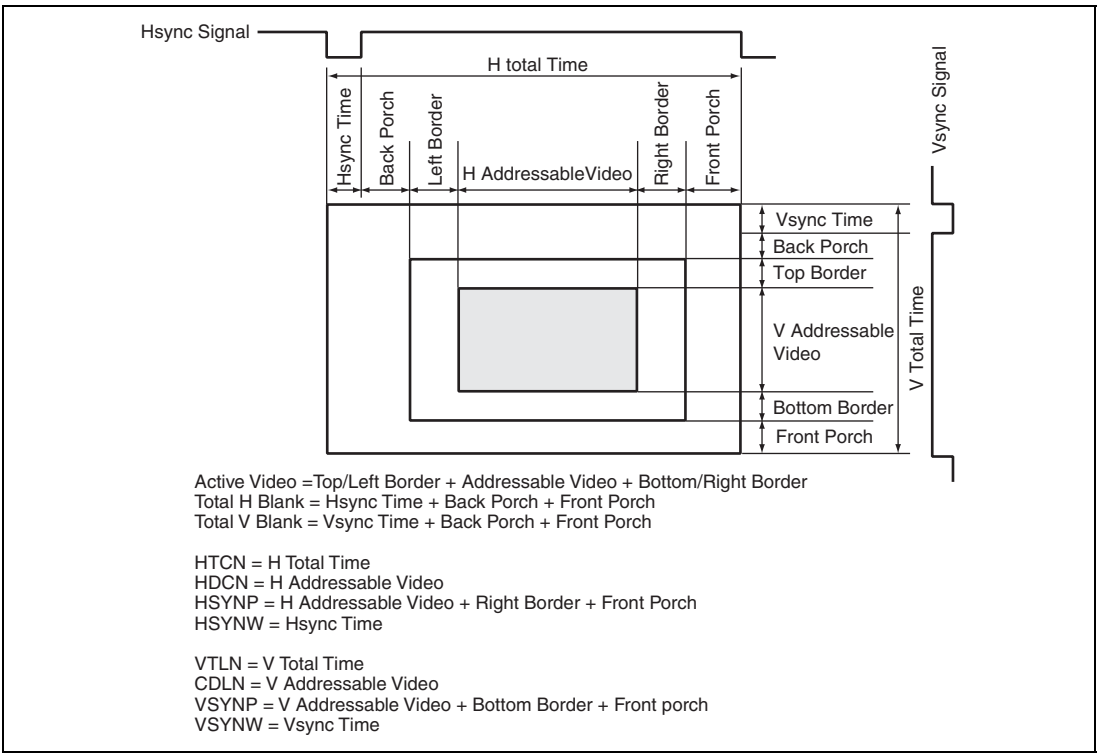


Figure 30.2 Valid Display and the Retrace Period

Table 30.4 Display Resolutions when Using Display Rotation

Image for Display in Memory (X-Resolution × Y-Resolution)	LCD Module (X-Resolution × Y-Resolution)	Number of Colors for Display	
240 × 320	320 × 240	Monochrome	4 bpp
			6 bpp
		Color	8 bpp
			16 bpp
234 × 320	320 × 234	Monochrome	6 bpp
		Color	16 bpp
80 × 160	160 × 80	Monochrome	2 bpp
			4 bpp
			6 bpp
		Color	4 bpp
			8 bpp
			16 bpp
64 × 128	128 × 64	Monochrome	1 bpp
			2 bpp
			4 bpp
			6 bpp
		Color	4 bpp
			8 bpp

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. For each resolution, only the number of colors indicated in the table is supported.

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.

30.4.3 Color Palette Specification

(1) 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.

below:

1. The PALEN bit in the LDPALCR is 0 (initial value); normal display operation
2. Access LDPALCR and set the PALEN bit to 1; enter color-palette setting mode after three cycles of peripheral clock.
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 after a cycle of peripheral clock.

While the PALS bit in LPDALCR is set to 1, the display data for the LCDC (LCD_DATA) will output a value of 0.

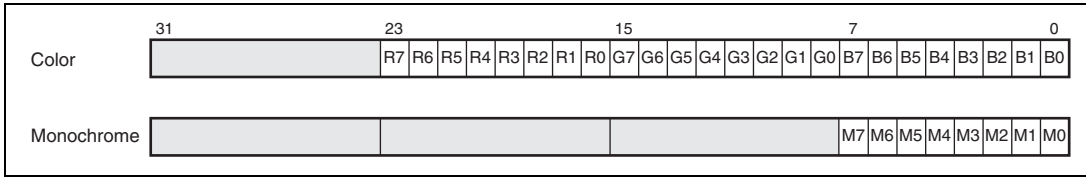


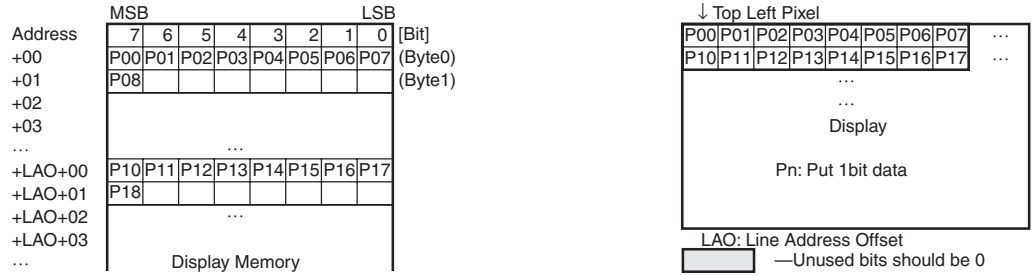
Figure 30.3 Color-Palette Data Format

PALDnn color and gradation data should be set as above.

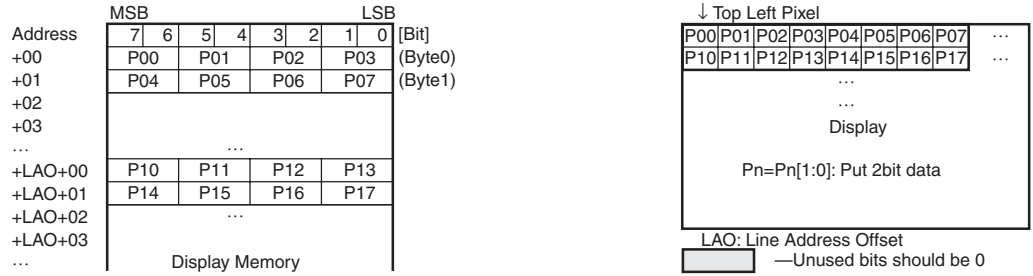
For a color display, PALDnn [23:16], PALDnn [15:8], and PALDnn [7:0] respectively hold the R, G, and B data. However, although there are register bits at PALDnn [18:16], PALDnn [9:8], and PALDnn [2:0], there is no corresponding memory for them. PALDnn [18:16], PALDnn [9:8], and PALDnn [2:0] are thus not available for storing palette data. The numbers of valid bits are thus R: 5, G: 6, and B: 5. However, 24-bit (R: 8 bits, G: 8 bits, and B: 8 bits) data should be written to the palette-data registers. When the values for PALDnn [23:19], PALDnn [15:10], or PALDnn [7:3] are not 0, 1 or 0 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. Data is thus extended to 24 bits.

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]. Data is thus extended to 8 bits.

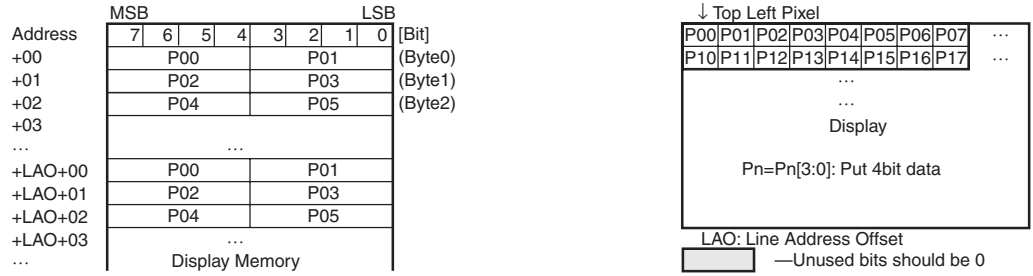
1. Packed 1bpp (Pixel Alignment in Byte is Big Endian) [Windows CE Recommended Format]



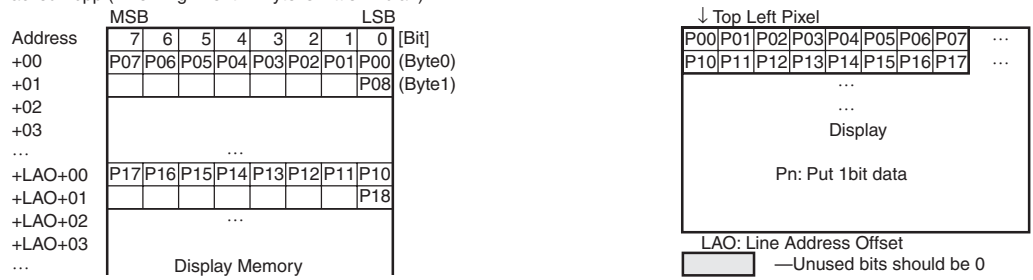
2. Packed 2bpp (Pixel Alignment in Byte is Big Endian) [Windows CE Recommended Format]



3. Packed 4bpp (Pixel Alignment in Byte is Big Endian) [Windows CE Recommended Format]



4. Packed 1bpp (Pixel Alignment in Byte is Little Endian)



Address	7	6	5	4	3	2	1	0	[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								

↓ 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

6. Packed 4bpp (Pixel Alignment in Byte is Little Endian)

	MSB							LSB	
Address	7	6	5	4	3	2	1	0	[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								

↓ 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

7. Unpacked 4bpp [Windows CE Recommended Format]

	MSB							LSB	
Address	7	6	5	4	3	2	1	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[3:0]: Put 4bit data								

LAO: Line Address Offset

—Unused bits should be 0

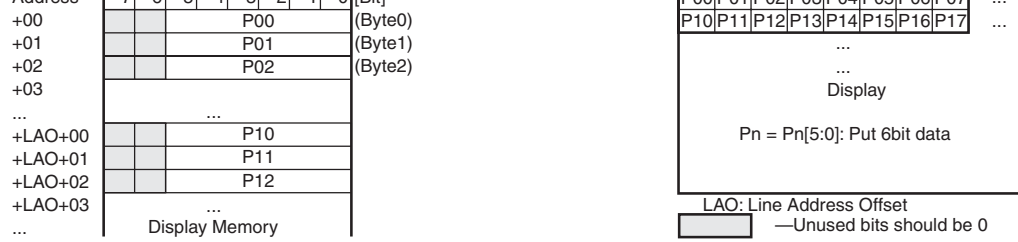
8. Unpacked 5bpp [Windows CE Recommended Format]

	MSB							LSB	
Address	7	6	5	4	3	2	1	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								

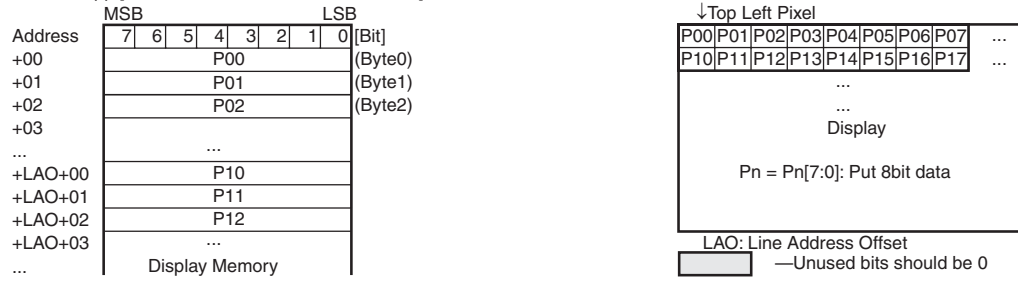
↓ Top Left Pixel								
P00	P01	P02	P03	P04	P05	P06	P07	…
P10	P11	P12	P13	P14	P15	P16	P17	…
…								
…								
Display								
Pn = Pn[4:0]: Put 5bit data								

LAO: Line Address Offset

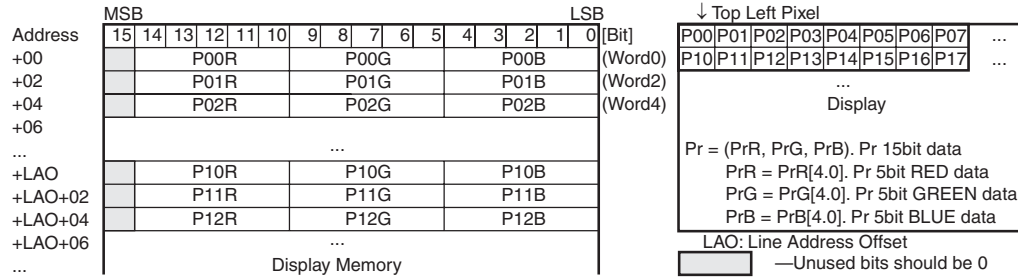
—Unused bits should be 0



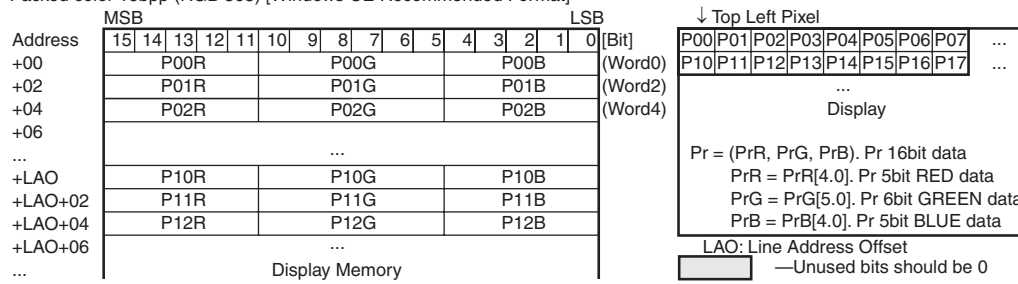
10. Packed 8bpp [Windows CE Recommended Format]



11. Unpacked color 15bpp (RGB 555) [Windows CE Recommended Format]



12. Packed color 16bpp (RGB 565) [Windows CE Recommended Format]



The display resolution is set with the LDHCNR, LDHSYNR, LDVDLNR, LDVTLNR, and LDVSYNR. The LCD current-alternating period for STN or DSTN display is set by using the LDACLNR. The initial values in these registers are set to VGA (640 × 480 dots), a typical resolution for STN or DSTN display.

The clock to be used is set with the LDICKR. The LCD module frame rate is determined by the display interval of one screen (as specified by a size-related register) + retrace line interval (non-display interval), and by the frequency of the clock used.

This LCDC has a Vsync 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 LDINTR.

30.4.6 Power Supply Control Sequence Processing

The LCD module normally requires processing of a specific sequence for cutting off of the input power supply. Settings in LDPMMR, LDPSPR, and LDCNTR, in conjunction with the LCD power-supply control pins (VCPWC, VEPWC, and LCD_DON), are used to provide processing of power-supply control sequences that suits the requirements of the LCD module.

Figures 30.4 to 30.7 are summary timing charts for power-supply control sequences and table 30.5 is a summary of available power-supply control sequence periods.

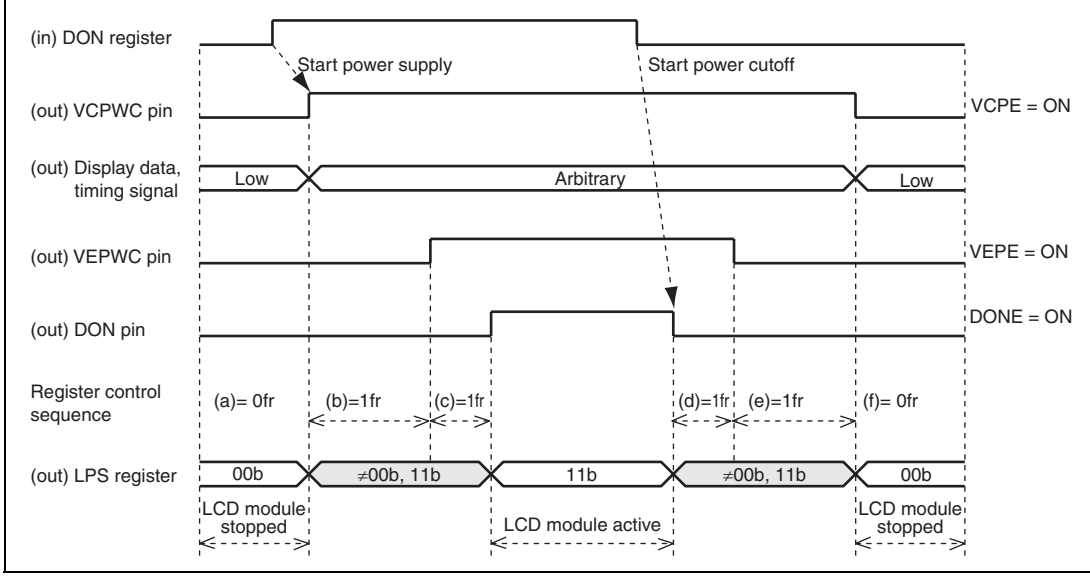


Figure 30.4 Power-Supply Control Sequence and States of the LCD Module

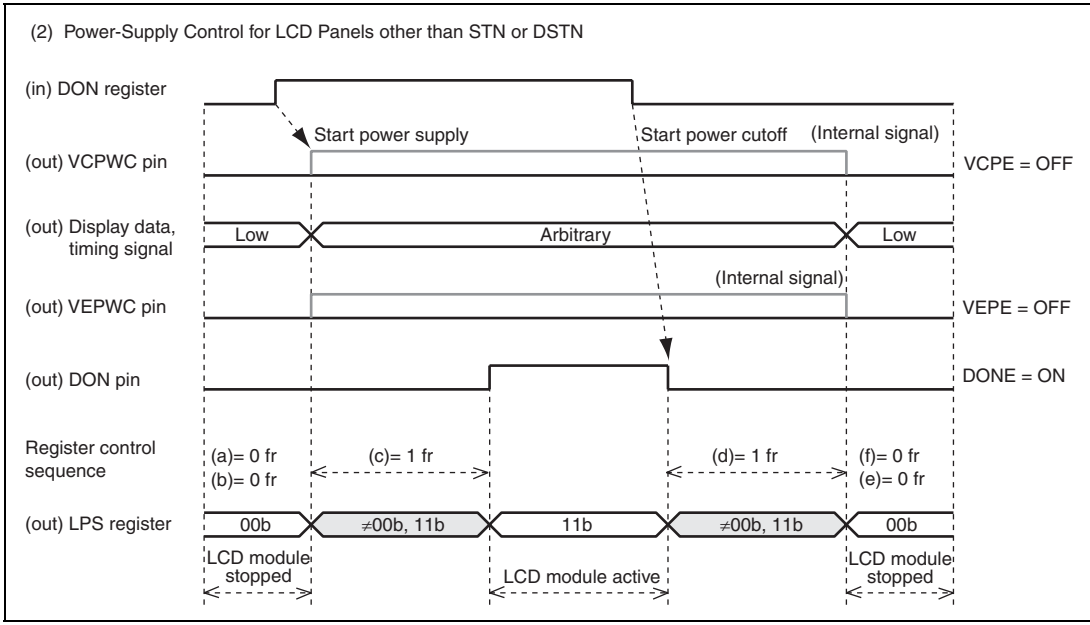


Figure 30.5 Power-Supply Control Sequence and States of the LCD Module

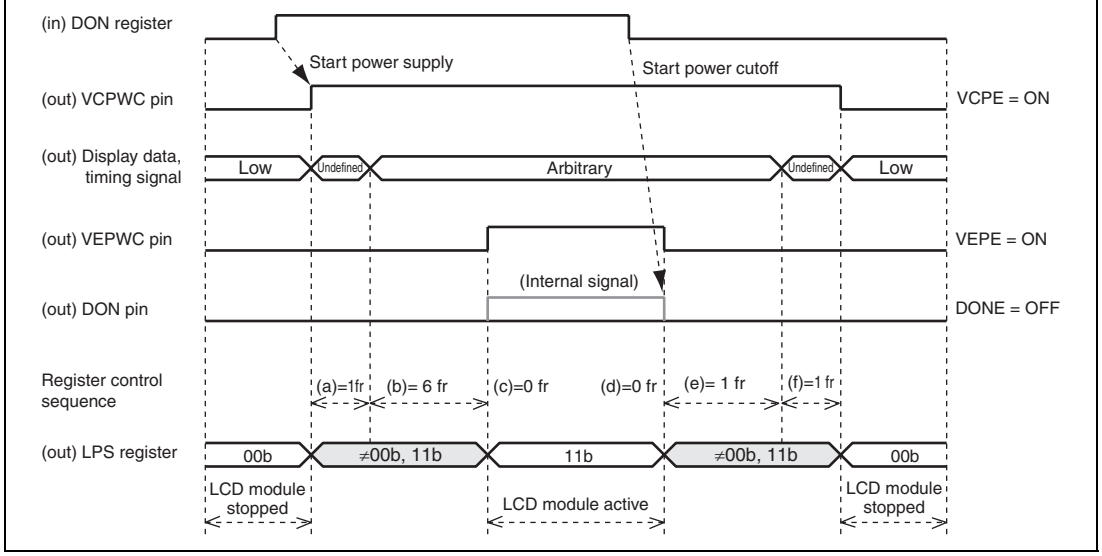


Figure 30.6 Power-Supply Control Sequence and States of the LCD Module

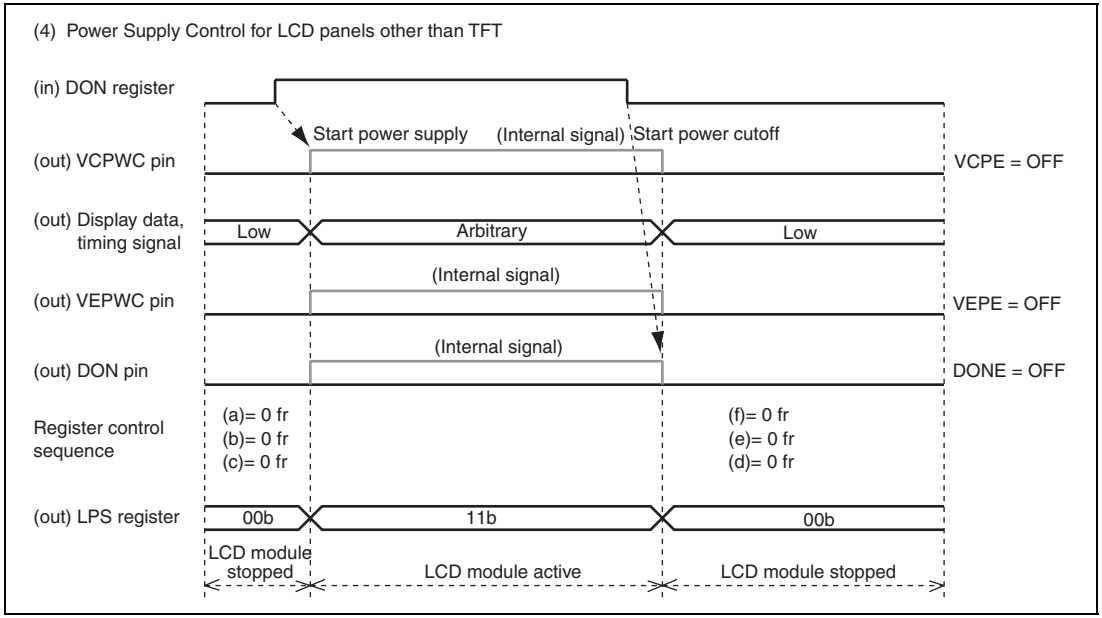


Figure 30.7 Power-Supply Control Sequence and States of the LCD Module

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)

Each of the ONA, ONB, ONC, OFFD, OFFE, and OFFF registers can be used to set the power-supply control-sequence periods, in units of frames, from 0 to 15. The register setting is one less than the value for that register, so the settings for H'0 through H'E correspond to 1 through 15 frames, with H'F set as 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 H'6 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.

Mode		Function
Display on (LCDC active)	Register setting: DON = 1	Timing signals and display data according to the prescribed resolution and number of colors are output to the LCD module.
Display off (LCDC stopped)	Register setting: DON = 0	Register access is enabled. Timing signals and display data according to the prescribed resolution and number of colors are not output to the LCD module.

Table 30.7 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	LCD_CL2, LCD_CL1, LCD_FLM, LCD_M_DISP, LCD_DATA	VEPWC	LCD_DON
Operating State	Supply	Supply	Supply	Supply
(Transitional State)	Supply	Supply	Supply	
	Supply	Supply		
	Supply			
Stopped State				

State	Power Supply for Logic	Display Data, Timing Signal	Power Supply for High-Voltage Systems
Control Pin	VCPWC	LCD_CL2, LCD_CL1, LCD_FLM, LCD_M_DISP, LCD_DATA	VEPWC
Operating State	Supply	Supply	Supply
(Transitional State)	Supply	Supply	
	Supply		
Stopped State			

The table above 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.

Warning regarding display-off mode (LCDC stopped):

When LCD module power-supply control-sequence processing is in use by the LCDC and the supply of power is cut off while the LCDC is in its display-on mode, the LCDC may not function properly. In the worst case, the connected LCD module may be damaged.

30.4.7 Operation for Hardware Rotation

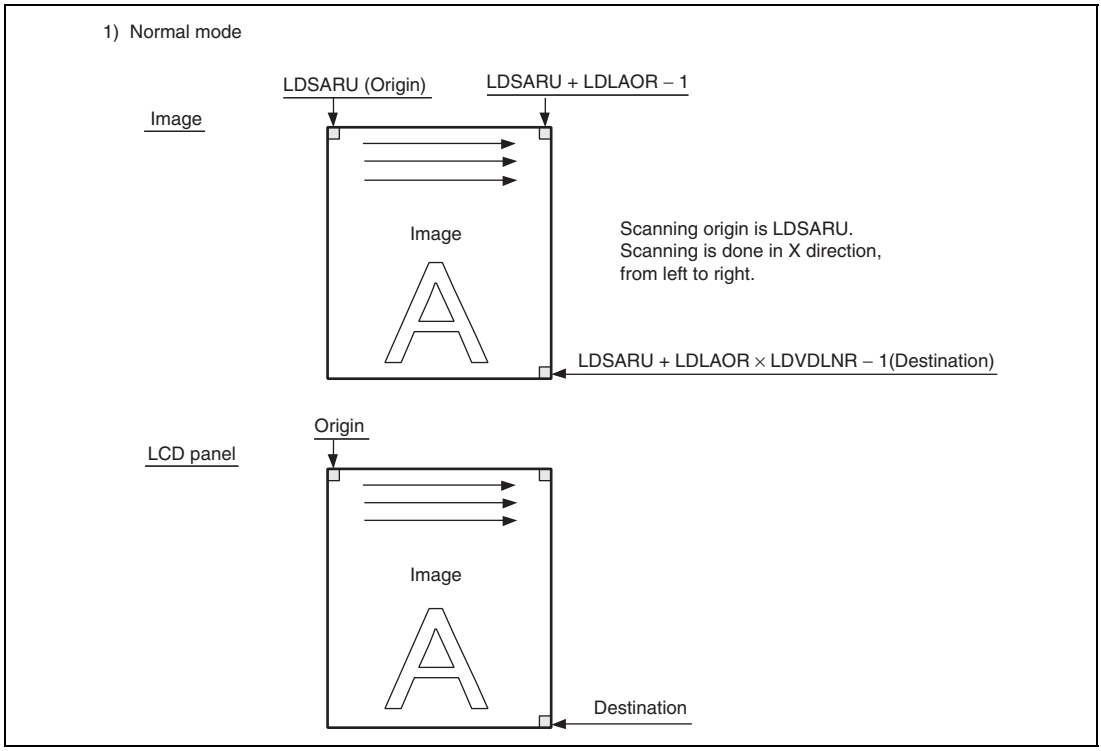
Operation in hardware-rotation mode is described below. Hardware-rotation mode can be thought of as using a landscape-format LCD panel instead of a portrait-format LCD panel by placing the landscape-format LCD panel as if it were a portrait-format panel. Whether the panel is intended for use in landscape or portrait format is thus no problem. The panel must, however, be within 320 pixels wide.

When conducting hardware rotation, the following five changes must be made from the settings for no hardware rotation. (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).

LDLAOR should be set to 256).

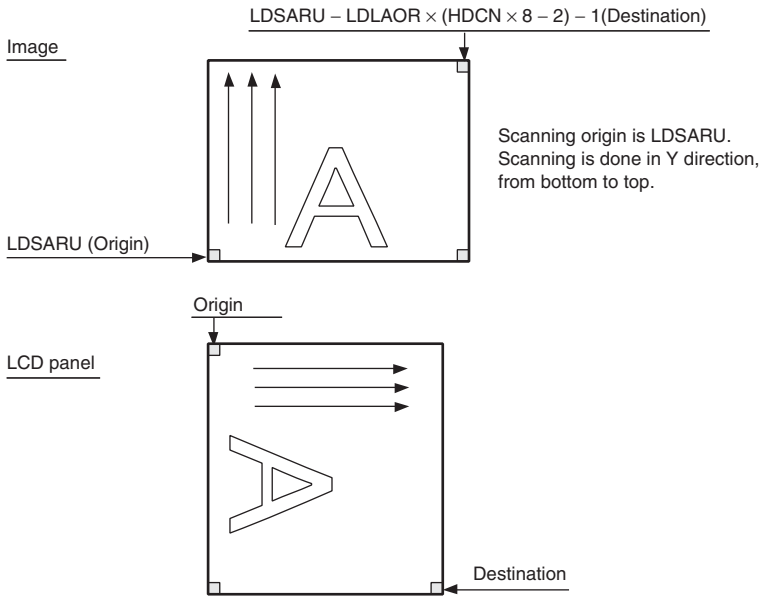
4. Graphics software should also be set as in number 3.
5. LDSARU should be changed to represent the address of the data for the lower-left pixel of the image rather than that of the data for the upper-left pixel of the image.



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.

of the LCD panel before rotation. Rotation is possible regardless of the drawing process carried out by the graphics driver software, but the image size and address offset values for the image managed by the graphics driver software must correspond to those of the actual image.

2) Rotation mode



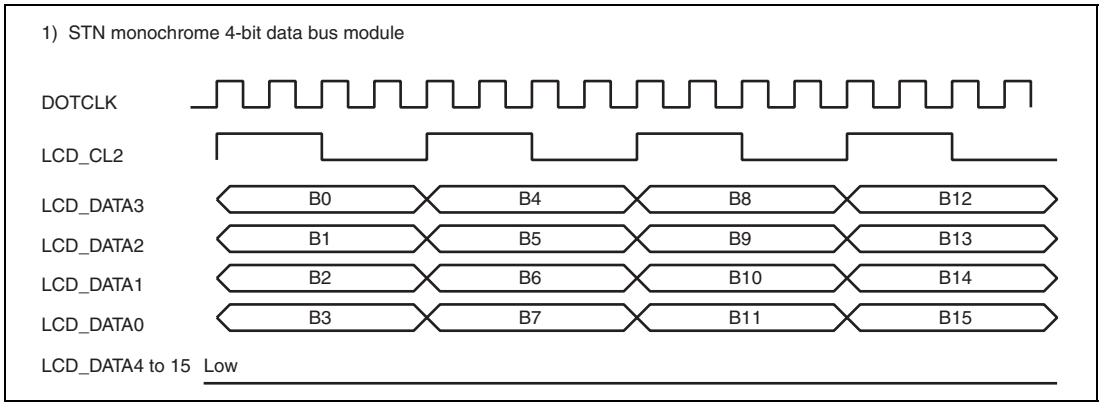


Figure 30.8 Clock and LCD Data Signal Example

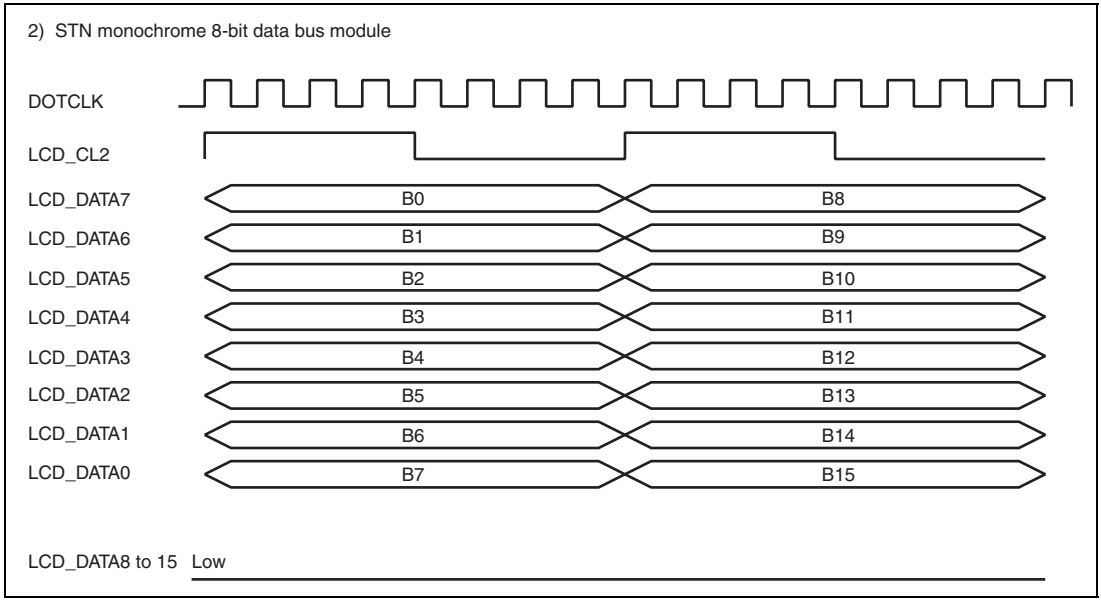


Figure 30.9 Clock and LCD Data Signal Example

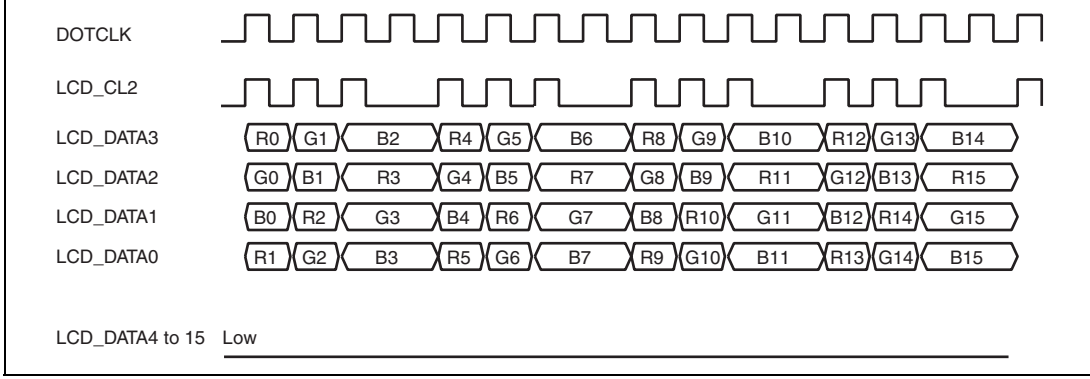


Figure 30.10 Clock and LCD Data Signal Example

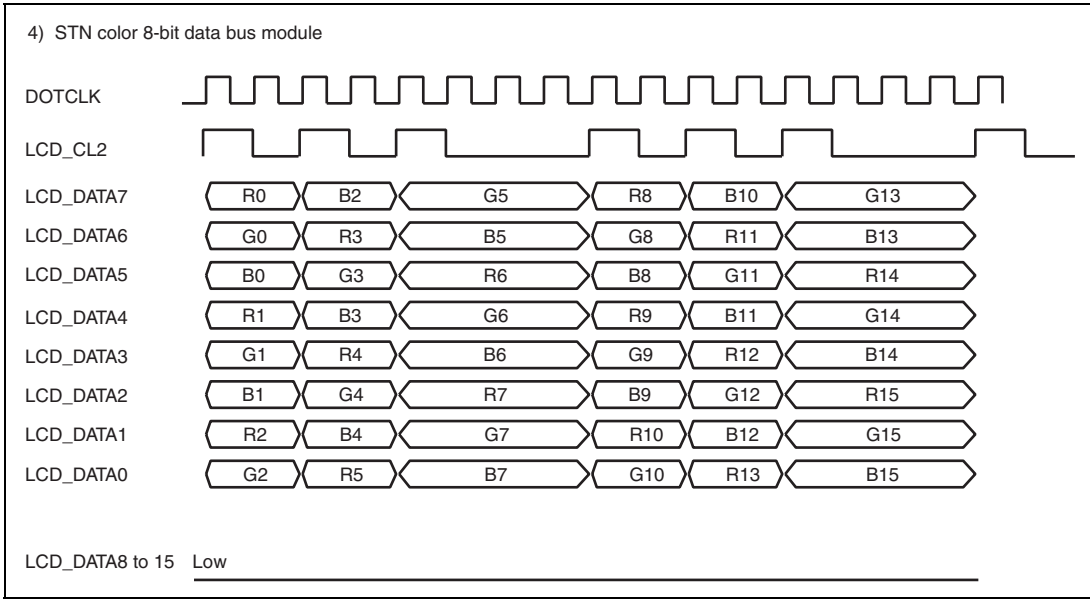


Figure 30.11 Clock and LCD Data Signal Example

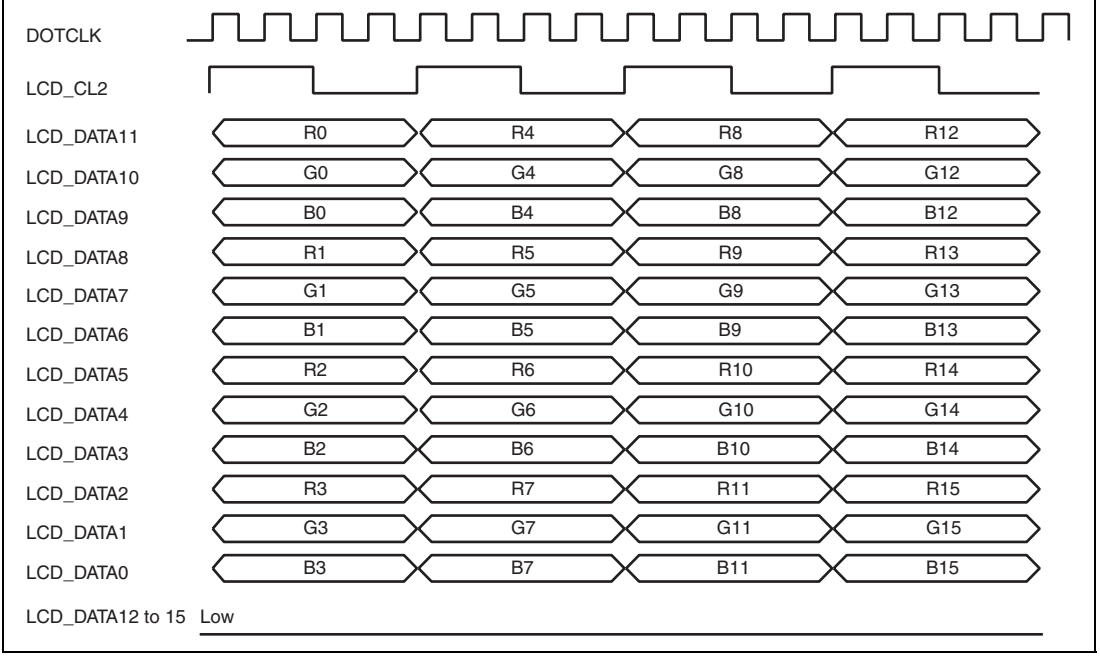


Figure 30.12 Clock and LCD Data Signal Example

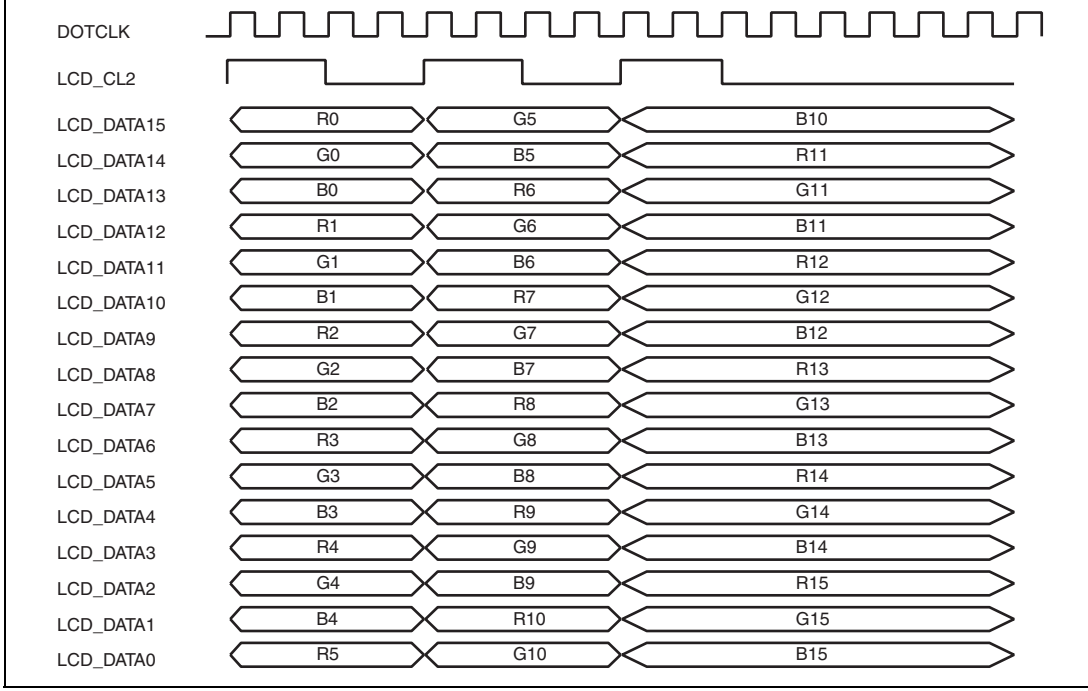


Figure 30.13 Clock and LCD Data Signal Example

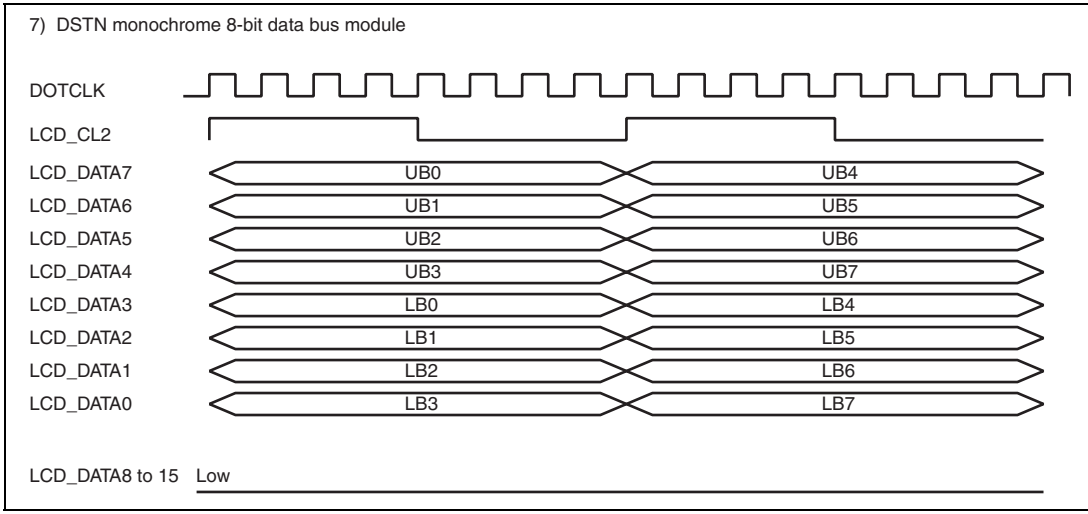


Figure 30.14 Clock and LCD Data Signal Example

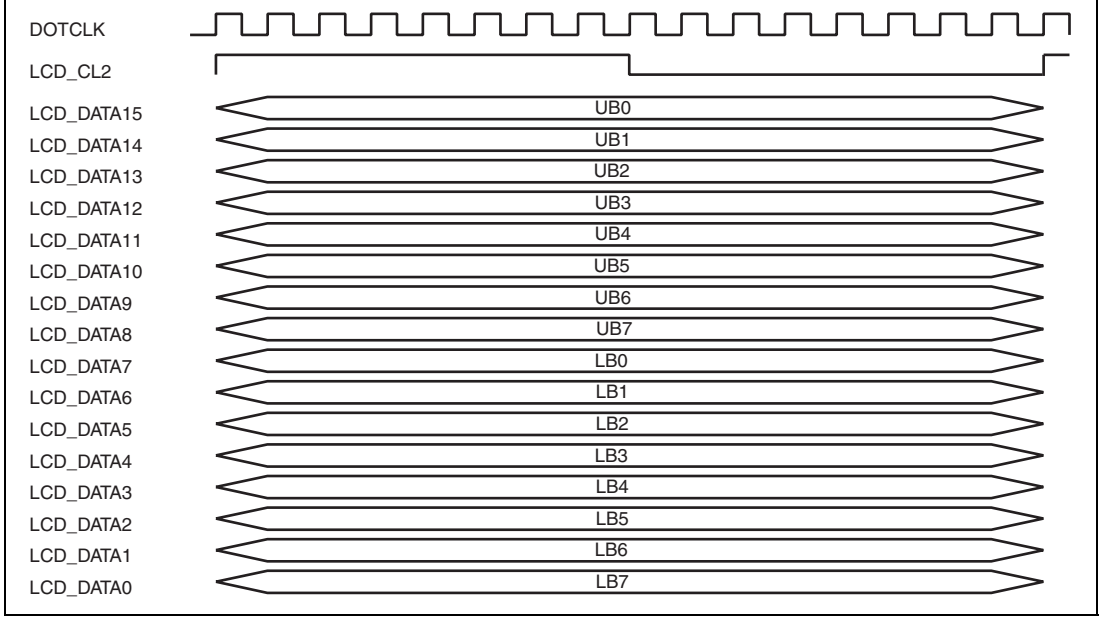


Figure 30.15 Clock and LCD Data Signal Example

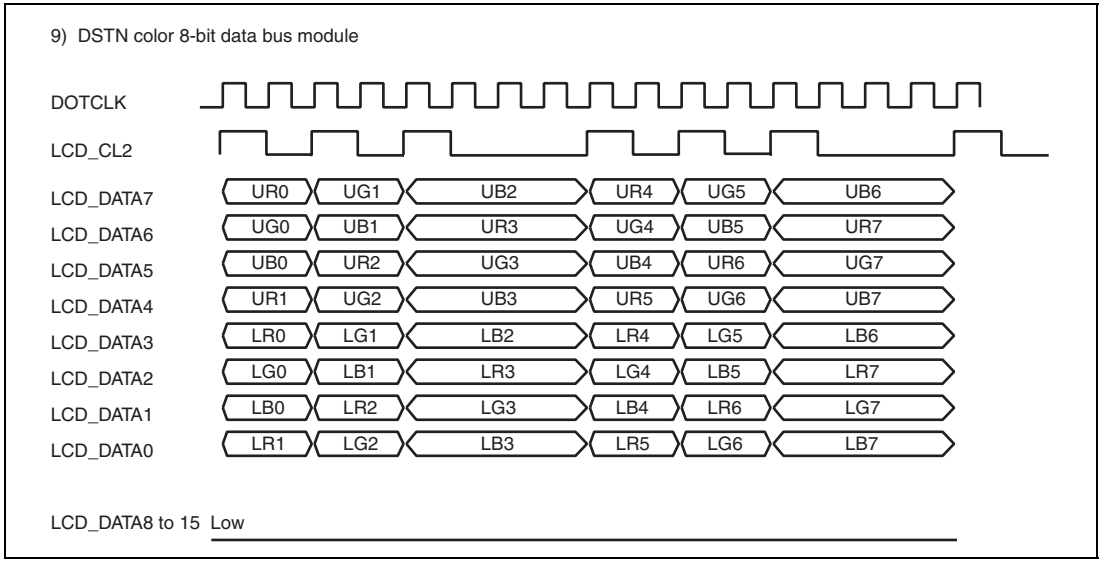


Figure 30.16 Clock and LCD Data Signal Example

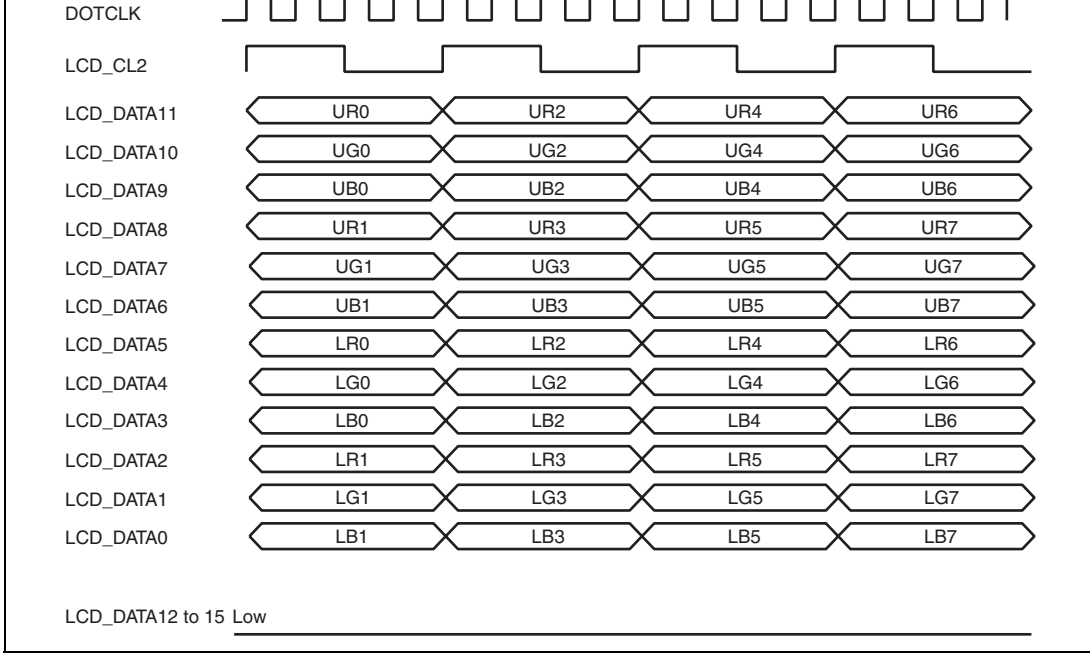


Figure 30.17 Clock and LCD Data Signal Example

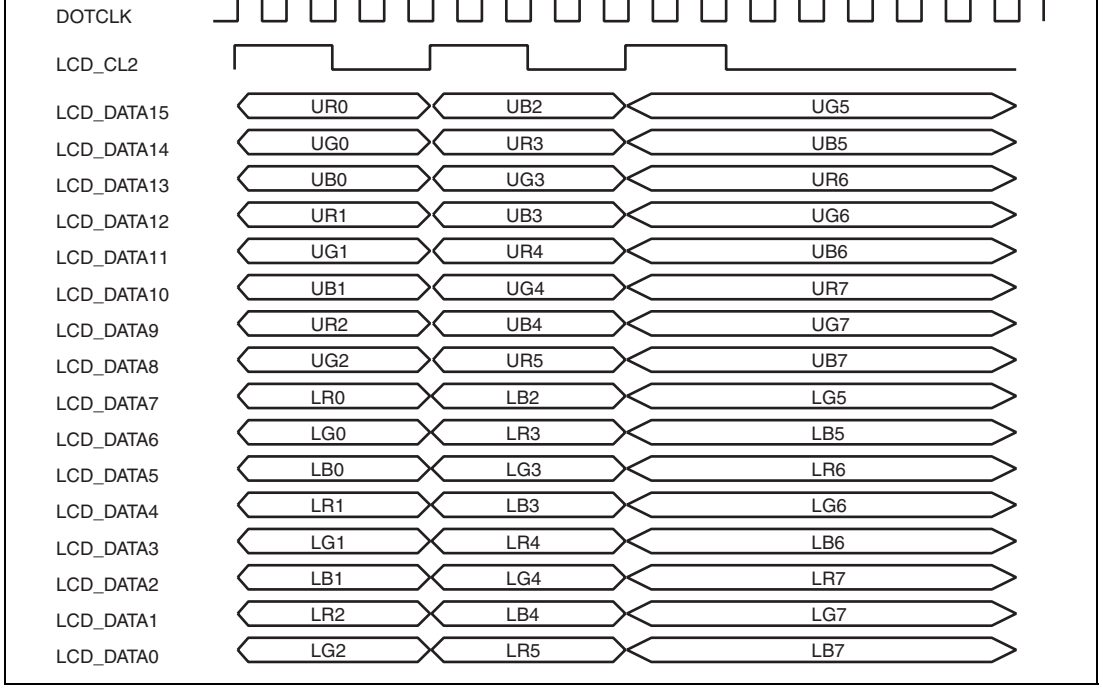


Figure 30.18 Clock and LCD Data Signal Example

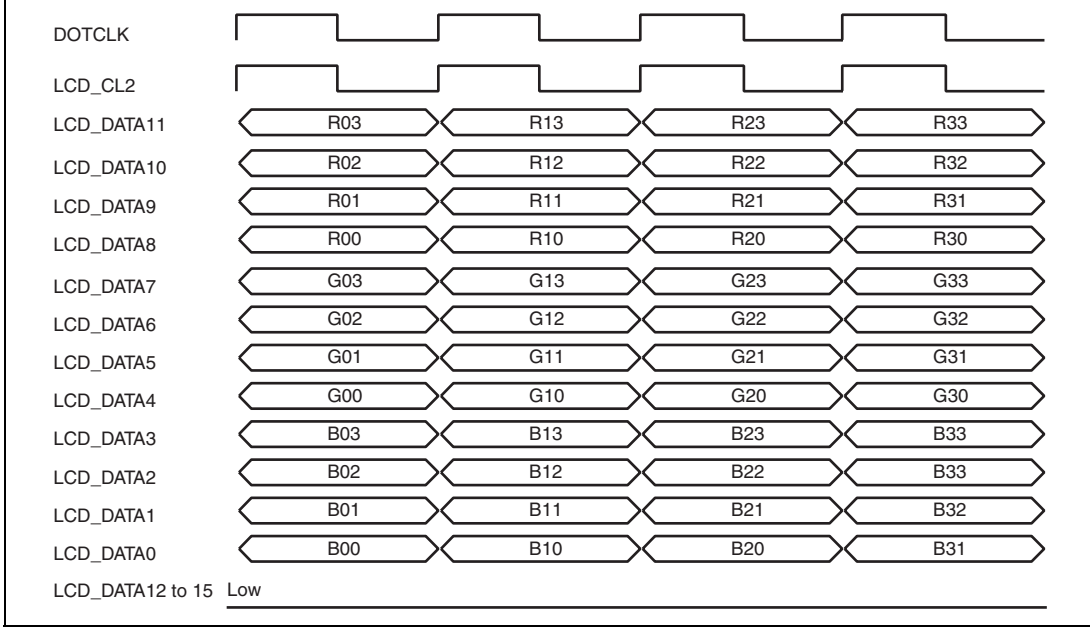


Figure 30.19 Clock and LCD Data Signal Example

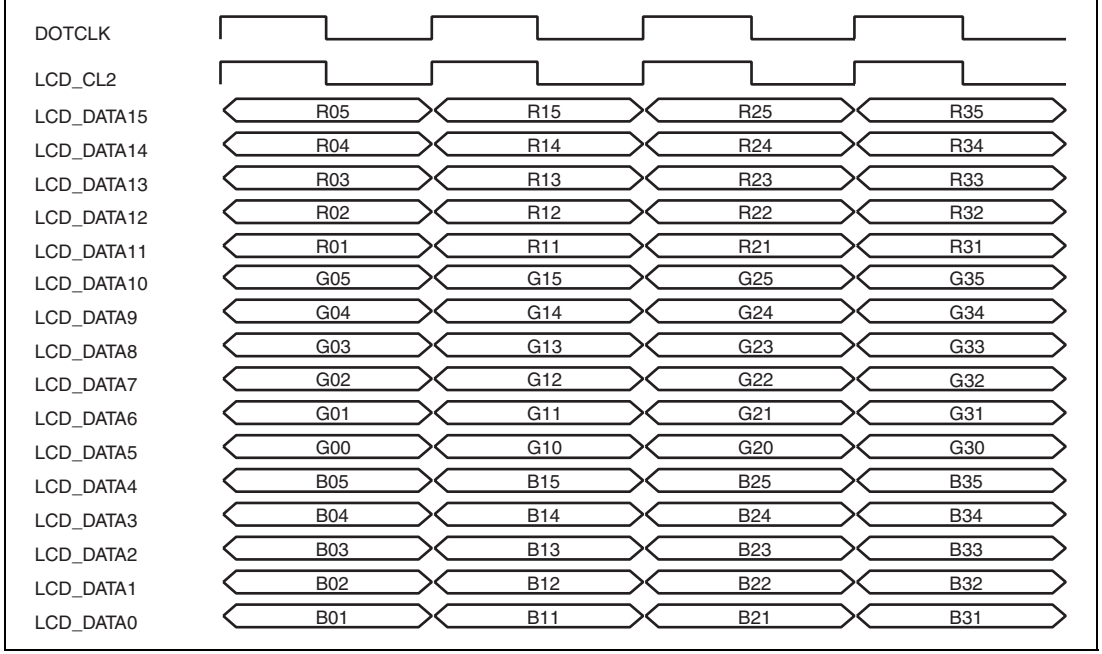


Figure 30.20 Clock and LCD Data Signal Example

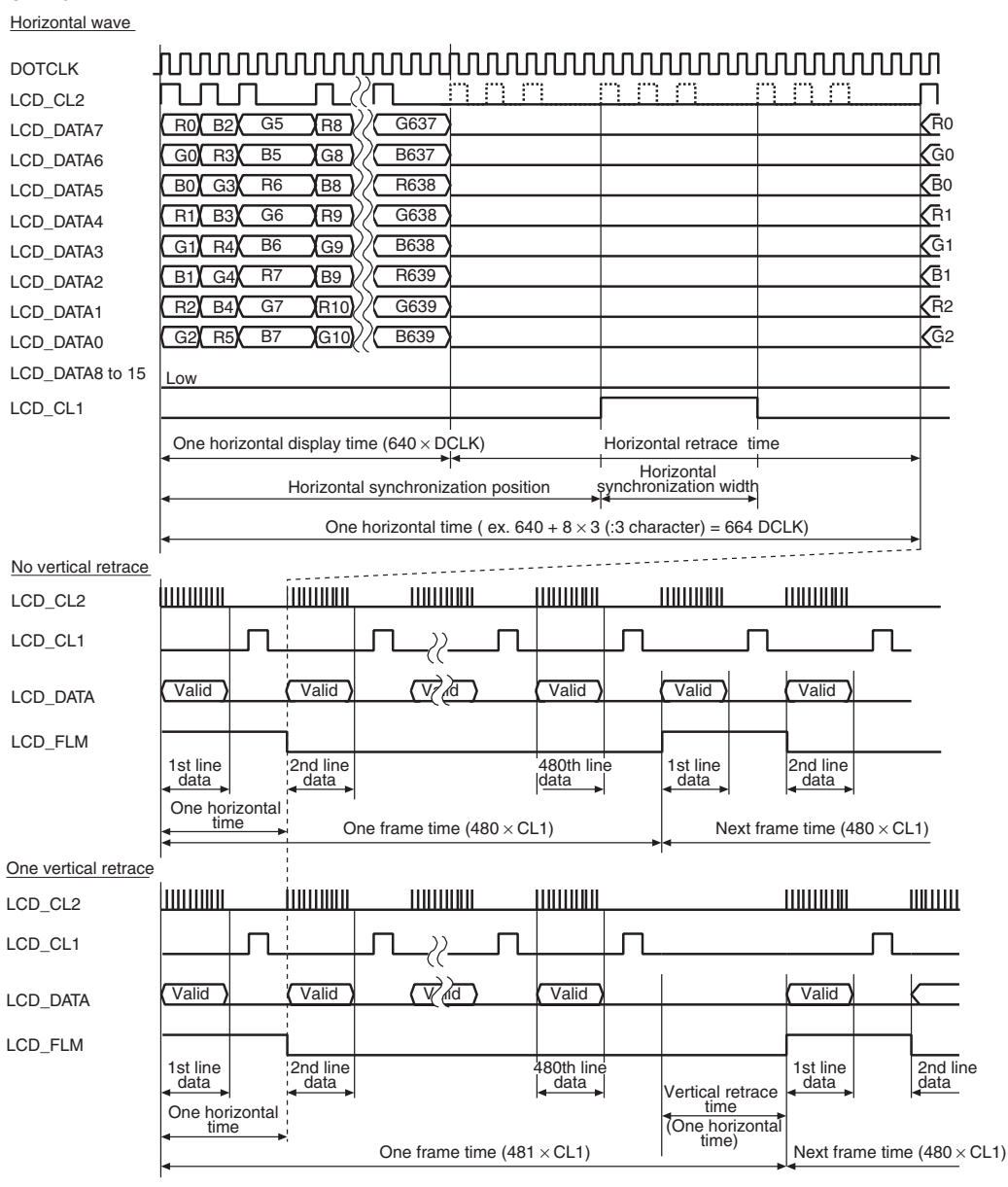


Figure 30.21 Clock and LCD Data Signal Example

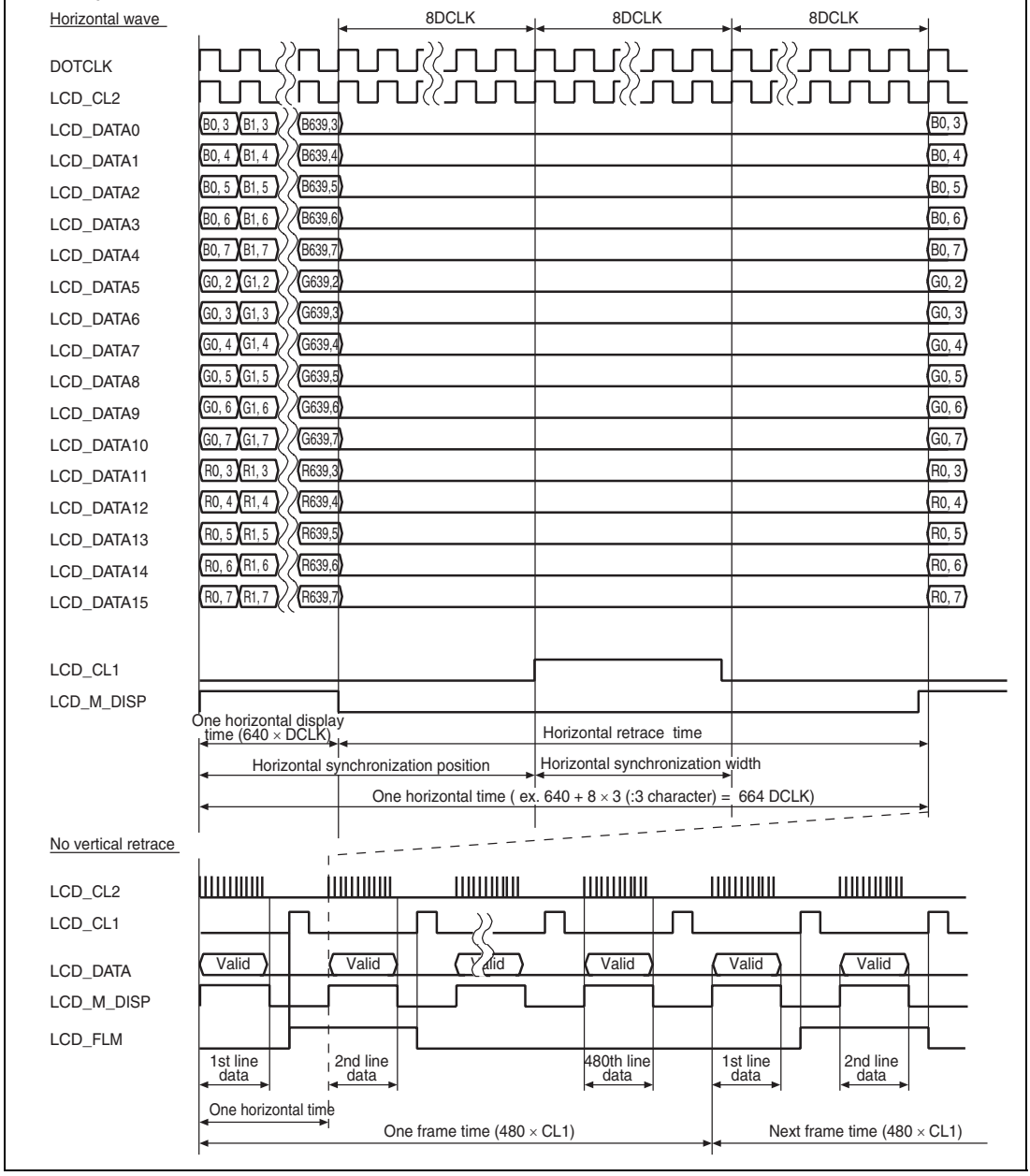


Figure 30.22 Clock and LCD Data Signal Example

30.6.1 LCD Module Sizes that can be displayed with SH7760 LCDC

Refer the following condition about a display size of the LCD Module. There is a possibility not to be displayed correctly when the condition is not satisfied.

Condition: Horizontal resolution [pixel] × Display colors [bpp])/8 × Horizontal frequency [kHZ]/ 1000 < Bck frequency [MHz]/3.6

30.6.2 Hardware Rotation Function

Refer the following condition about a display size of the LCD Module. There is a possibility not to be displayed correctly when the condition is not satisfied.

Condition: Horizontal resolution [pixel] × Display colors [bpp])/8 × Horizontal frequency [kHz]/ 1000 < Bck frequency [MHz]/50

E.g.) Horizontal resolution: 320 pixel, Display colors: 16 bpp, Horizontal frequency: 15.6 kHz, Bck frequency: 66 MHz (320 * 16/8 * 15.6)/1000 = 9.98 [Mbytes/sec] > 66/50 = 1.32: This result doesn't satisfy the conditional in using the hardware rotation function mode, so the setting of LCDC should be changed.

30.6.3 Power Cutoff

Set LDPMMR.OFFD[3:0], LDPSPR.OFFE[3:0], and LDPSPR.OFFF[3:0], so that the following relationship is met when power supply to the device is turned off.

$70 \times 80 \times tBcyc > 0.084 \approx 0.1$ ms

If the above relationship is not met, stop LCDC using the following procedure.

(1) Verify that LDPMMR.LPS = 11.
(2) Set LDCNTR.DON/DON2 to 0. (This turns off LCD display.)
(3) Verify that LDPMMR.LPS = 00.
(4) Read the DMAC SAR0 and verify that the transfer source address is not changing.

The user break controller (UBC) provides functions that simplify program debugging. When break conditions are set in the UBC, a user break interrupt is generated according to the contents of the bus cycle generated by the CPU. This function makes it easy to design an effective self-monitoring debugger, enabling programs to be debugged with the chip alone, without using an in-circuit emulator.

31.1 Features

The UBC has the following features.

- Two break channels (A and B)

  User break interrupts can be generated on independent conditions for channels A and B or on sequential conditions (sequential break setting: channel A → channel B).
- The following break compare conditions can be set:
  - Address (selection of 32-bit virtual address and ASID for comparison):

    Address: All bits compared/lower 10 bits masked/lower 12 bits masked/lower 16 bits masked/lower 20 bits masked/all bits masked

    ASID: All bits compared/all bits masked
  - Data (channel B only, 32-bit mask capability)
  - Bus cycle: Instruction access/operand access
  - Read/write
  - Operand size: Byte/word/longword/quadword
- An instruction access cycle break can be effected before or after the instruction is executed.

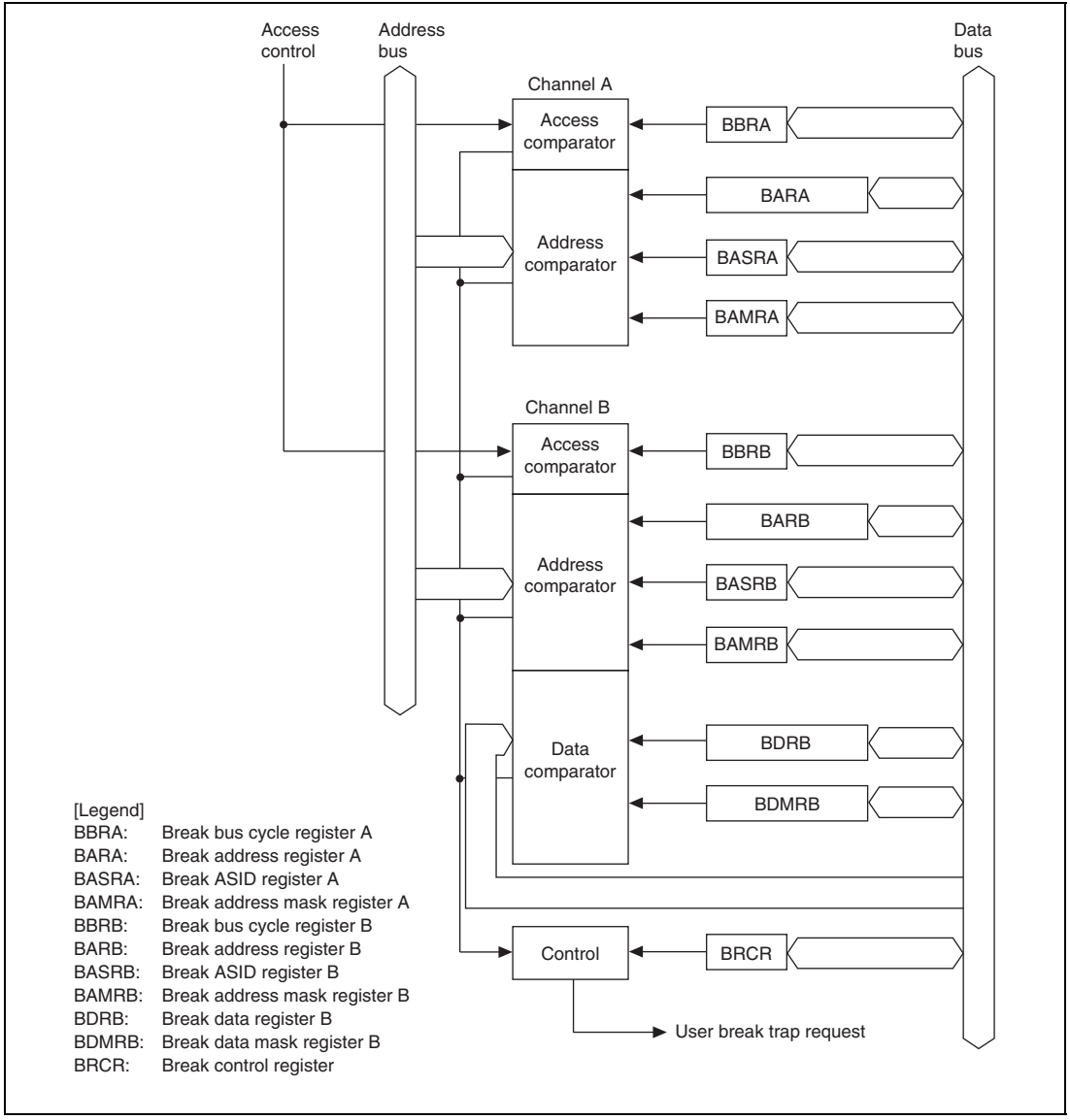


Figure 31.1 Block Diagram of UBC

The UBC has the following registers. Refer to section 32, List of Registers, for the addresses of the registers and the state of each register in each processor state.

Table 31.1 Register Configuration (1)

Register Name	Abbreviation	R/W	P4 Address	Area 7 Address	Size	Sync Clock
Break address register A	BARA	R/W	H'FF20 0000	H'1F20 0000	32	Ick
Break ASID register A	BASRA	R/W	H'FF00 0014	H'FF00 0014	8	Ick
Break address mask register A	BAMRA	R/W	H'FF20 0004	H'1F20 0004	8	Ick
Break bus cycle register A	BBRA	R/W	H'FF20 0008	H'1F20 0008	16	Ick
Break address register B	BARB	R/W	H'FF20 000C	H'1F20 000C	32	Ick
Break ASID register B	BASRB	R/W	H'FF00 0018	H'1F00 0018	8	Ick
Break address mask register B	BAMRB	R/W	H'FF20 0010	H'1F20 0010	8	Ick
Break bus cycle register B	BBRB	R/W	H'FF20 0014	H'1F20 0014	16	Ick
Break data register B	BDRB	R/W	H'FF20 0018	H'1F20 0018	32	Ick
Break data mask register B	BDMRB	R/W	H'FF20 001C	H'1F20 001C	32	Ick
Break control register	BRCR	R/W	H'FF20 0020	H'1F20 0020	16	Ick

Register Name	Abbrev.	Power-on Reset by RESET Pin/WDT/ H-UDI	Manual Reset by RESET Pin/WDT/ Multiple Exception	Sleep by Sleep Instruction/ Deep Sleep	Standby	
					by Hardware	by Software/ Each Module
Break address register A	BARA	Undefined	Retained	Retained	*2	Retained
Break ASID register A	BASRA	Undefined	Retained	Retained		Retained
Break address mask register A	BAMRA	Undefined	Retained	Retained		Retained
Break bus cycle register A	BBRA	H'0000	Retained	Retained		Retained
Break address register B	BARB	Undefined	Retained	Retained		Retained
Break ASID register B	BASRB	Undefined	Retained	Retained		Retained
Break address mask register B	BAMRB	Undefined	Retained	Retained		Retained
Break bus cycle register B	BBRB	H'0000	Retained	Retained		Retained
Break data register B	BDRB	Undefined	Retained	Retained		Retained
Break data mask register B	BDMRB	Undefined	Retained	Retained		Retained
Break control register	BRCR	H'0000*1	Retained	Retained		Retained

Notes: 1. This value includes an undefined bit value. Refer to the register description.
2. After exiting hardware standby mode, this LSI enters the power-on reset state caused by the RESET pin.

The access size must be the same as the control register size. If the access size is different from the register size, no data will be written to the register and an undefined value will be read.

UBC control register contents cannot be transferred to a floating-point register using a floating-point memory data transfer instruction. When a UBC control register is updated, use either of the following methods to make the updated value valid:

1. Execute an RTE instruction after the memory store instruction that updated the register. The updated value will be valid from the RTE instruction destination onward.
2. Execute instructions requiring 5 states for execution after the memory store instruction that updated the register. This LSI executes two instructions in parallel and a minimum of 0.5 states are required for execution of one instruction, 11 instructions must be inserted. The updated value will be valid from the 6th state onward.

BARA and BARB are 32-bit readable/writable registers that specify the virtual addresses used in the channel A and channel B break conditions.

- BARA

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	BAA31	BAA30	BAA29	BAA28	BAA27	BAA26	BAA25	BAA24	BAA23	BAA22	BAA21	BAA20	BAA19	BAA18	BAA17	BAA16
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

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BAA15	BAA14	BAA13	BAA12	BAA11	BAA10	BAA9	BAA8	BAA7	BAA6	BAA5	BAA4	BAA3	BAA2	BAA1	BAA0
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

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	BAA31 to BAA0	—	R/W	Break Address A31 to A0 Stores the virtual addresses used in the channel A break condition.

- BARB

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	BAB31	BAB30	BAB29	BAB28	BAB27	BAB26	BAB25	BAB24	BAB23	BAB22	BAB21	BAB20	BAB19	BAB18	BAB17	BAB16
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

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BAB15	BAB14	BAB13	BAB12	BAB11	BAB10	BAB9	BAB8	BAB7	BAB6	BAB5	BAB4	BAB3	BAB2	BAB1	BAB0
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

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	BAB31 to BAB0	—	R/W	Break Address B31 to B0 Stores the virtual addresses used in the channel B break condition.

BASRA and BASRB are 8-bit readable/writable registers that specify the ASID used in the channel A and channel B break conditions.

- 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

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	BASA7 to BASA0	—	R/W	Break ASID A7 to A0 Stores the ASID (bits 7 to 0) used in the channel A break condition.

- 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

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	BASB7 to BASB0	—	R/W	Break ASID B7 to B0 Stores the ASID (bits 7 to 0) used in the channel B break condition.

BAMRA is an 8-bit readable/writable register that specifies which bits are to be masked in the break ASID set in BASRA and the break address set in BARA.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	BAMA2	BASMA	BAMA1	BAMA0
Initial value:	0	0	0	0	-	-	-	-
R/W:	R	R	R	R	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2	BASMA	—	R/W	Break ASID Mask A Specifies whether all bits of the channel A break ASID (BASA7 to BASA0) set in BASRA are to be masked. 0: All BASRA bits are included in break conditions 1: No BASRA bits are included in break conditions
3 1 0	BAMA2 BAMA1 BAMA0	— — —	R/W R/W R/W	Break Address Mask A2 to A0 These bits specify which bits of the channel A break address (BAA31 to BAA0) set in BARA are to be masked. 000: All BARA bits are included in break conditions 001: Lower 10 bits of BARA are masked, and not included in break conditions 010: Lower 12 bits of BARA are masked, and not included in break conditions 011: All BARA bits are masked, and not included in break conditions 100: Lower 16 bits of BARA are masked, and not included in break conditions 101: Lower 20 bits of BARA are masked, and not included in break conditions 11x: Setting prohibited

Note: x: Don't care

BAMRB is an 8-bit readable/writable register that specifies which bits are to be masked in the break ASID set in BASRA and the break address set in BARB.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	BAMB2	BASMB	BAMB1	BAMB0
Initial value:	0	0	0	0	-	-	-	-
R/W:	R	R	R	R	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2	BASMB	—	R/W	Break ASID Mask B Specifies whether all bits of the channel B break ASID (BASB7 to BASB0) set in BASRB are to be masked. 0: All BASRB bits are included in break conditions 1: No BASRB bits are included in break conditions
3 1 0	BAMB2 BAMB1 BAMB0	— — —	R/W R/W R/W	Break Address Mask B2 to B0 These bits specify which bits of the channel B break address (BAB31 to BAB0) set in BARB are to be masked. 000: All BARB bits are included in break conditions 001: Lower 10 bits of BARB are masked, and not included in break conditions 010: Lower 12 bits of BARB are masked, and not included in break conditions 011: All BARB bits are masked, and not included in break conditions 100: Lower 16 bits of BARB are masked, and not included in break conditions 101: Lower 20 bits of BARB are masked, and not included in break conditions 11x: Setting prohibited

Note: x: Don't care

BBRA is a 16-bit readable/writable register that specifies three conditions from among the channel A break conditions: instruction access/operand access, read/write, and operand size.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	SZA2	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	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 7	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5	IDA1	0	R/W	Instruction Access/Operand Access Select A
4	IDA0	0	R/W	These bits specify whether an instruction access cycle or an operand access cycle is used as the bus cycle in the channel A break conditions. 00: Condition comparison is not performed 01: Instruction access cycle is used as break condition 10: Operand access cycle is used as break condition 11: Instruction access cycle or operand access cycle is used as break condition
3	RWA1	0	R/W	Read/Write Select A
2	RWA0	0	R/W	These bits specify whether a read cycle or write cycle is used as the bus cycle in the channel A break conditions. 00: Condition comparison is not performed 01: Read cycle is used as break condition 10: Write cycle is used as break condition 11: Read cycle or write cycle is used as break condition
6	SZA2	0	R/W	Operand Size Select A
1	SZA1	0	R/W	These bits select the operand size of the bus cycle
0	SZA0	0	R/W	used as a channel A break conditions. 000: Operand size is not included in break conditions 001: Byte access is used as break condition 010: Word access is used as break condition 011: Longword access is used as break condition 100: Quadword access is used as break condition 101: Setting prohibited 11x: Setting prohibited

Note: x: Don't care

BBRB is a 16-bit readable/writable register that specifies three conditions from among the channel B break conditions: instruction access/operand access, read/write, and operand size.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	SZB2	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	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 7	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5 4	IDB1 IDB0	0 0	R/W R/W	Instruction Access/Operand Access Select B These bits specify whether an instruction access cycle or an operand access cycle is used as the bus cycle in the channel B break conditions. 00: Condition comparison is not performed 01: Instruction access cycle is used as break condition 10: Operand access cycle is used as break condition 11: Instruction access cycle or operand access cycle is used as break condition
3 2	RWB1 RWB0	0 0	R/W R/W	Read/Write Select B These bits specify whether a read cycle or write cycle is used as the bus cycle in the channel B break conditions. 00: Condition comparison is not performed 01: Read cycle is used as break condition 10: Write cycle is used as break condition 11: Read cycle or write cycle is used as break condition
6 1 0	SZB2 SZB1 SZB0	0 0 0	R/W R/W R/W	Operand Size Select B These bits select the operand size of the bus cycle used as a channel B break conditions. 000: Operand size is not included in break conditions 001: Byte access is used as break condition 010: Word access is used as break condition 011: Longword access is used as break condition 100: Quadword access is used as break condition 101: Setting prohibited 11x: Setting prohibited

Note: x: Don't care

BDRB is a 32-bit readable/writable register that specifies the data (bits 31 to 0) to be used in the channel B break conditions.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	BDB31	BDB30	BDB29	BDB28	BDB27	BDB26	BDB25	BDB24	BDB23	BDB22	BDB21	BDB20	BDB19	BDB18	BDB17	BDB16
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

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BDB15	BDB14	BDB13	BDB12	BDB11	BDB10	BDB9	BDB8	BDB7	BDB6	BDB5	BDB4	BDB3	BDB2	BDB1	BDB0
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

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	BDB31 to BDB0	—	R/W	Break Data B These bits store the data (bits 31 to 0) to be used in the channel B break conditions.

31.2.8 Break Data Mask Register B (BDMRB)

BDMRB is a 32-bit readable/writable register that specifies which bits of the break data set in BDRB are to be masked. When the value of the data bus is included in the break conditions, its operand size should be specified. When the byte size is specified as an operand size, the same data should be specified to bits 15 to 8 and bits 7 to 0 in both BDRB and BDMRB.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	BDMB31	BDMB30	BDMB29	BDMB28	BDMB27	BDMB26	BDMB25	BDMB24	BDMB23	BDMB22	BDMB21	BDMB20	BDMB19	BDMB18	BDMB17	BDMB16
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

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BDMB15	BDMB14	BDMB13	BDMB12	BDMB11	BDMB10	BDMB9	BDMB8	BDMB7	BDMB6	BDMB5	BDMB4	BDMB3	BDMB2	BDMB1	BDMB0
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

31 to 0	BDMB31 to BDMB0	—	R/W	Break Data Mask B These bits specify whether the corresponding bit of the channel B break data Bn set in BDRB is to be masked. 0: Channel B break data Bn is included in break conditions 1: Channel B break data Bn is masked, and not included in break conditions

Note: n = 31 to 0

31.2.9 Break Control Register (BRCR)

BRCR is a 16-bit readable/writable register that specifies (1) whether channels A and B are to be used as two independent channels or in a sequential condition, (2) whether the break is to be effected before or after instruction execution, (3) whether the BDRB register is to be included in the channel B break conditions, and (4) whether the user break debug function is to be used. BRCR also contains condition match flags. The CMFA, CMFB, and UBDE bits in BRCR are initialized to 0 by a power-on reset, but retain their value in standby mode. The PCBA, DBEB, PCBB, and SEQ bits are undefined after a power-on reset or manual reset, so these bits should be initialized by software as necessary.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CMFA	CMFB	-	-	-	PCBA	-	-	DBEB	PCBB	-	-	SEQ	-	-	UBDE
Initial value:	0	0	0	0	0	-	0	0	-	-	0	0	-	0	0	0
R/W:	R/W	R/W	R	R	R	R/W	R	R	R/W	R/W	R	R	R/W	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	CMFA	0	R/W	Condition Match Flag A Set to 1 when a break condition set for channel A is satisfied. This flag is not cleared to 0. To confirm that the flag is set again after once being set, it should be cleared with a write. 0: Channel A break condition does not matched 1: Channel A break condition has matched
14	CMFB	0	R/W	Condition Match Flag B Set to 1 when a break condition set for channel B is satisfied. This flag is not cleared to 0. To confirm that the flag is set again after once being set, it should be cleared with a write. 0: Channel B break condition is not matched 1: Channel B break condition match has occurred

13 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10	PCBA	—	R/W	Instruction Access Break Select A Specifies whether a channel A instruction access cycle break is to be effected before or after the instruction is executed. 0: Channel A PC break is effected before instruction execution 1: Channel A PC break is effected after instruction execution
9 ,8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7	DBEB	—	R/W	Data Break Enable B Specifies whether the data bus condition is to be included in the channel B break conditions. When the data bus is included in the break conditions, bits IDB1 to IDB0 in BBRB should be set to B'10 or B'11. 0: Data bus condition is not included in channel B break conditions 1: Data bus condition is included in channel B break conditions
6	PCBB	—	R/W	PC Break Select B Specifies whether a channel B instruction access cycle break is to be effected before or after the instruction is executed. 0: Channel B PC break is effected before instruction execution 1: Channel B PC break is effected after instruction execution
5, 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
3	SEQ	—	R/W	Sequence Condition Select Specifies whether the conditions for channels A and B are to be independent or sequential. 0: Channel A and B comparisons are performed as independent conditions 1: Channel A and B comparisons are performed as sequential conditions (channel A → channel B)

2, 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	UBDE	0	R/W	User Break Debug Enable Specifies whether the user break debug function is used or not. 0: User break debug function is not used 1: User break debug function is used

31.3 Operation

31.3.1 Explanation of Terms Relating to Access

An instruction access is an access that obtains an instruction. For example, fetching an instruction from the branch destination when a branch instruction is executed is an instruction access. An operand access is any memory access for the purpose of instruction execution. For example, the access to address (PC+disp×2+4) in the instruction MOV.W@(disp,PC), Rn is an operand access. As the term “data” is used to distinguish data from an address, the term “operand access” is used in this section.

All operand accesses are treated as either read accesses or write accesses. The following instructions require special attention:

- PREF, OCBP, and OCBWB instructions: Treated as read accesses.
- MOVCA.L and OCBI instructions: Treated as write accesses.
- TAS.B instruction: Treated as one read access and one write access.

The operand accesses for the PREF, OCBP, OCBWB, and OCBI instructions are accesses with no access data.

This LSI handles all operand accesses as having a data size. The data size can be byte, word, longword, or quadword. The operand data size for the PREF, OCBP, OCBWB, MOVCA.L, and OCBI instructions is treated as longword.

In this section, “1 (2, 3 …) instruction(s) after…”, as a measure of the distance between two instructions, is defined as follows. A branch is counted as an interval of two instructions.

(1) Example of sequence of instructions with no branch:

100 Instruction A (0 instructions after instruction A)
102 Instruction B (1 instruction after instruction A)
104 Instruction C (2 instructions after instruction A)
106 Instruction D (3 instructions after instruction A)

(2) Example of sequence of instructions with a branch (however, the example of a sequence of instructions with no branch should be applied when the branch destination of a delayed branch instruction is the instruction itself + 4):

100 Instruction A: BT/S L200 (0 instructions after instruction A)
102 Instruction B (1 instruction after instruction A, 0 instructions after instruction B)
L200 200 Instruction C (3 instructions after instruction A, 2 instructions after instruction B)
202 Instruction D (4 instructions after instruction A, 3 instructions after instruction B)

31.3.3 User Break Operation Sequence

The sequence of operations from setting of break conditions to user break exception handling is described below.

1. Specify pre- or post-execution break in the case of an instruction access, inclusion or exclusion of the data bus value in the break conditions in the case of an operand access, and use of independent or sequential channel A and B break conditions in BRCR. Set the break addresses in BARA and BARB, set the ASIDs corresponding to the break space in BASRA and BASRB, and set the address and ASID masking methods in BAMRA and BAMRB. If the data bus value is to be included in the break conditions, also set the break data in BDRB and the data mask in BDMRB.
2. Set the break bus conditions in BBRA and BBRB. If even one of the BBRA/BBRB instruction access/operand access select (the ID bit) and read/write select groups (the RW bit) is set to B'00, a user break interrupt will not be generated on the corresponding channel. Make the BBRA and BBRB settings after all other break-related register settings have been completed. If breaks are enabled with BBRA/BBRB while the break address, data, mask register, or the break control register is in the initial state after a reset, a break may be generated inadvertently.

When the BL bit is 0, exception handling is started and the condition match flag (CMFA/CMFB) for the respective channel is set for the matched condition. When the BL bit is 1, the condition match flag (CMFA/CMFB) for the respective channel is set for the matched condition but exception handling is not started.

The condition match flags (CMFA, CMFB) are set by a break condition match, but are not automatically reset. Therefore, a memory store instruction should be used on BRCR to clear the flags to 0. For details of the exact setting conditions for the condition match flags, see section 31.3.6, Condition Match Flag Setting.

4. When sequential condition mode has been selected and the channel B condition is matched after the channel A condition has been matched, a break is effected at the instruction at which the channel B condition was matched. For details of the operation when the channel A condition match and channel B condition match occur close together, see section 31.3.8, Contiguous A and B Settings for Sequential Conditions. With sequential conditions, only the channel B condition match flag is set. To clear the channel A match when the channel A condition has been matched but the channel B condition has not yet been matched in sequential condition mode, B'0 should be written to the SEQ bit in BRCR.

31.3.4 Instruction Access Cycle Break

1. When an instruction access/read/word setting is made in BBRA/BBRB, an instruction access cycle can be used as a break condition. In this case, breaking before or after execution of the relevant instruction can be selected with the PCBA/PCBB bit in BRCR. When an instruction access cycle is used as a break condition, clear the LSB of BARA/BARB to 0. A break will not be generated if this bit is set to 1.
2. When a pre-execution break is specified, the break is effected when it is confirmed that the instruction is to be fetched and executed. Therefore, an overrun-fetched instruction (an instruction that is fetched but not executed when a branch or exception occurs) cannot be used in a break. However, if a TLB miss or TLB protection violation exception occurs at the time of the fetch of an instruction subject to a break, the break exception handling is carried out first. The instruction TLB exception handling is performed when the instruction is re-executed (see section 8.2, Exception Types and Priorities). Also, since a delayed branch instruction and the delay slot instruction are executed as a single instruction, if a pre-execution break is specified for a delay slot instruction, the break will be effected before execution of the delayed branch instruction. However, a pre-execution break cannot be specified for the delay slot instruction for an RTE instruction.

set for a delayed branch instruction, the delay slot is executed and the break is effected before execution of the instruction at the branch destination (when the branch is made) or the instruction two instructions ahead of the branch instruction (when the branch is not made).

4. When an instruction access cycle is set for channel B, BDRB is ignored in determining whether there is an instruction access match. Therefore, a break condition specified by the DBEB bit in BRCR is not executed.

31.3.5 Operand Access Cycle Break

1. In the case of an operand access cycle break, the bits included in address bus comparison vary as shown below according to the data size specification in BBRA and BBRB.

Data Size	Address Bits Compared
Quadword	Address bits A31 to A3
Longword	Address bits A31 to A2
Word	Address bits A31 to A1
Byte	Address bits A31 to A0
Not included in condition	In quadword access, address bits A31 to A3 In longword access, address bits A31 to A2 In word access, address bits A31 to A1 In byte access, address bits A31 to A0

2. When a data value is included in break conditions in channel B

   When a data value is included in the break conditions, set the DBEB bit in BRCR to 1. In this case, BDRB and BDMRB settings are necessary in addition to the address condition. A user break interrupt is generated when all three conditions, address, ASID, and data, are matched. When a quadword access occurs, the 64-bit access data is divided into an upper 32 bits and lower 32 bits, and interpreted as two 32-bit data units. A break is generated if either of the 32-bit data units satisfies the data match condition.

   Set the IDB1 to IDB0 bits in BBRB to B'10 or B'11. When byte data is specified, the same data should be set to bits 15 to 8 and bits 7 to 0 in BDRB and BDMRB. When word or byte is set, bits 31 to 16 in BDRB and BDMRB are ignored.

3. When the DBEB bit in BRCR is set to 1, a break is not generated by an operand access with no access data (an operand access in a PREF, OCBP, OCBWB, or OCBI instruction).

1. Instruction access with post-execution condition, or operand access
   The flag is set when execution of the instruction that causes the break is completed. As an exception to this, however, in the case of an instruction with more than one operand access the flag may be set on detection of the match condition alone, without waiting for execution of the instruction to be completed.
   A. Example 1:
      100 BT L200 (branch performed)
      102 Instruction (operand access break on channel A) → flag not set
   B. Example 2:
      110 FADD (FPU exception)
      112 Instruction (operand access break on channel A) → flag not set
2. Instruction access with pre-execution condition
   The flag is set when the break match condition is detected.
   A. Example 1:
      110 Instruction (pre-execution break on channel A) → flag set
      112 Instruction (pre-execution break on channel B) → flag not set
   B. Example 2:
      110 Instruction (pre-execution break on channel B, instruction access TLB miss)
      → flag set

31.3.7 Program Counter (PC) Value Saved

1. When instruction access (pre-execution) is set as a break condition, the program counter (PC) value saved to SPC in user break interrupt handling is the address of the instruction at which the break condition match occurred. In this case, a user break interrupt is generated and the fetched instruction is not executed.
2. When instruction access (post-execution) is set as a break condition, the program counter (PC) value saved to SPC in user break interrupt handling is the address of the instruction to be executed after the instruction at which the break condition match occurred. In this case, the fetched instruction is executed, and a user break interrupt is generated before execution of the next instruction.
3. When an instruction access (post-execution) break condition is set for a delayed branch instruction, the delay slot instruction is executed and a user break is effected before execution of the instruction at the branch destination (when the branch is made) or the instruction two instructions ahead of the branch instruction (when the branch is not made). In this case, the PC value saved to SPC is the address of the branch destination (when the branch is made) or the instruction following the delay slot instruction (when the branch is not made).

5. When operand access (address + data) is set as a break condition, execution of the instruction at which the condition match occurred is completed. A user break interrupt is generated before execution of instructions from one instruction later to four instructions later. It is not possible to specify at which instruction, from one later to four later, the interrupt will be generated. The start address of the instruction after the instruction for which execution is completed at the point at which user break interrupt handling is started is saved to SPC. If an instruction between one instruction later and four instructions later causes another exception, control is performed as follows. Designating the exception caused by the break as exception 1, and the exception caused by an instruction between one instruction later and four instructions later as exception 2, memory updates and register updates that can not be performed by exception 2 are guaranteed irrespective of the existence of exception 1. The PC value saved is the address of the first instruction for which execution is suppressed. Whether exception 1 or exception 2 is used for the exception jump destination and the value written to the exception register (EXPEVT/INTEVT) is not guaranteed. However, if exception 2 is from a source which is not synchronized with an instruction (external interrupt or peripheral module interrupt), exception 1 is used for the exception jump destination and the value written to the exception register (EXPEVT/INTEVT).

31.3.8 Contiguous A and B Settings for Sequential Conditions

When channel A match and channel B match timings are close together, a sequential break may not be guaranteed. Rules relating to the guaranteed range are given below.

1. Instruction access matches on both channel A and channel B

Instruction B is 0 instructions after instruction A	Equivalent to setting the same address. This setting is prohibited.
Instruction B is 1 instruction after instruction A	Sequential operation is not guaranteed.
Instruction B is 2 or more instructions after instruction A	Sequential operation is guaranteed.

2. Instruction access match on channel A, operand access match on channel B

Instruction B is 0 or 1 instruction after instruction A	Sequential operation is not guaranteed.
Instruction B is 2 or more instructions after instruction A	Sequential operation is guaranteed.

Instruction B is 0 to 3 instructions after instruction A	Sequential operation is not guaranteed.
Instruction B is 4 or more instructions after instruction A	Sequential operation is guaranteed.

4. Operand access matches on both channel A and channel B

   Do not make a setting such that a single operand access will match the break conditions of both channel A and channel B. There are no other restrictions. For example, sequential operation is guaranteed even if two accesses within a single instruction match channel A and channel B conditions in turn.

31.4 Usage Notes

1. Do not execute a post-execution instruction access break for the SLEEP instruction.
2. Do not make an operand access break setting between 1 and 3 instructions before a SLEEP instruction.
3. The value of the BL bit referenced in a user break exception depends on the break setting, as follows.
   - Pre-execution instruction access break: The BL bit value before the executed instruction is referenced.
   - Post-execution instruction access break: The OR of the BL bit values before and after the executed instruction is referenced.
   - Operand access break (address/data): The BL bit value after the executed instruction is referenced.
   - In the case of an instruction that modifies the BL bit

BL bit	Pre-Execution Instruction Access	Post-Execution Instruction Access	Pre-Execution Instruction Access	Post-Execution Instruction Access	Operand Access (Address/Data)
0 → 0	A	A	A	A	A
1 → 0	M	M	M	M	A
0 → 1	A	M	A	M	M
1 → 1	M	M	M	M	M

A: Accepted
M: Masked

The BL bit value before execution of a delay slot instruction is the same as the BL bit value before execution of an RTE instruction. The BL bit value after execution of a delay slot instruction is the same as the first BL bit value for the first instruction executed on returning by means of an RTE instruction (the same as the value of the BL bit in SSR before execution of the RTE instruction).

— If an interrupt or exception is accepted with the BL bit cleared to 0, the value of the BL bit before execution of the first instruction of the exception handling routine is 1.

4. If channels A and B both match independently at virtually the same time, and, as a result, the SPC value is the same for both user break interrupts, only one user break interrupt is generated, but both the CMFA bit and the CMFB bit are set. For example:

   110 Instruction (post-execution instruction break on channel A) → SPC = 112, CMFA = 1

   112 Instruction (pre-execution instruction break on channel B) → SPC = 112, CMFB = 1

5. The PCBA or PCBB bit in BRCR is valid for an instruction access break setting.
6. When the SEQ bit in BRCR is 1, the internal sequential break state is initialized by a channel B condition match. For example: A → A → B (user break generated) → B (no break generated)
7. In the event of conflict between a re-execution type exception and a post-execution break in a multistep instruction, the re-executing-type exception is generated. In this case, the CMF bit may or may not be set to 1 when the break condition occurs.
8. A post-execution break is classified as a completed-type exception. Consequently, in the event of conflict between a completed-type exception and a post-execution break, the post-execution break is suppressed in accordance with the priorities of the two events. For example, in the case of conflict between a TRAPA instruction and a post-execution break, the user break is suppressed. However, in this case, the CMF bit is set by the occurrence of the break condition.

31.5 User Break Debug Support Function

The user break debug support function enables the processing used in the event of a user break exception to be changed. When a user break exception occurs, if the UBDE bit in BRCR is set to 1, the DBR value will be used as the branch destination address instead of [VBR + offset]. The value of R15 is saved in SGR regardless of the value of the UBDE bit in BRCR or the type of exception event. A flowchart of the user break debug support function is shown in figure 31.2.

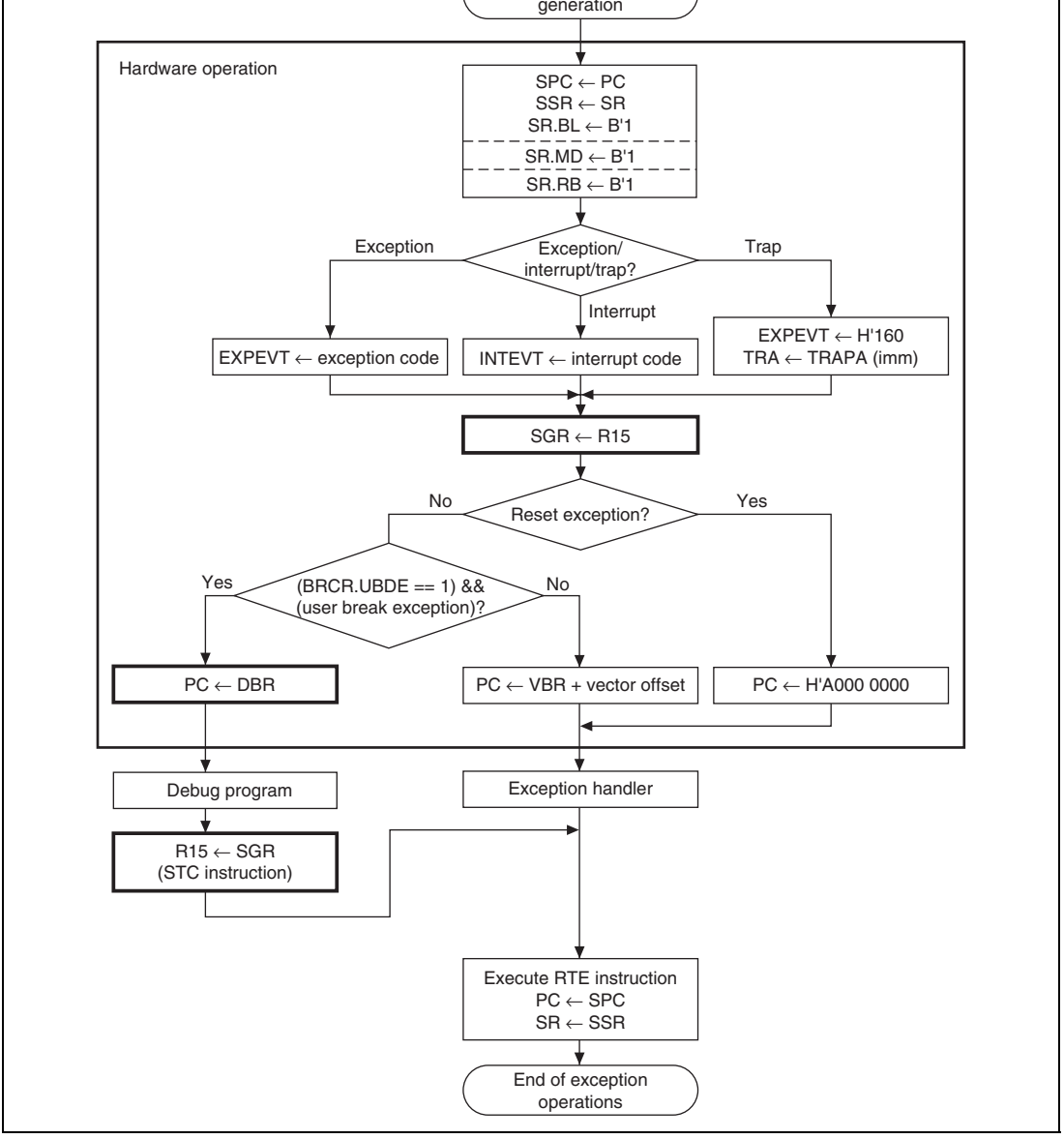


Figure 31.2 User Break Debug Support Function Flowchart

(1) Instruction Access Cycle Break Condition Settings

1. Register settings: BASRA = H'80 / BARA = H'0000 0404 / BAMRA = H'00 /
   BBRA = H'0014 / BASRB = H'70 / BARB = H'0000 8010 / BAMRB = H'01 /
   BBRB = H'0014 / BDRB = H'0000 0000 / BDMRB = H'0000 0000 / BRCR = H'0400
   — Conditions set: Independent channel A/channel B mode
   - Channel A: ASID: H'80 / address: H'0000 0404 / address mask: H'00
     Bus cycle: instruction access (post-instruction-execution), read (operand size not included in conditions)
   - Channel B: ASID: H'70 / address: H'0000 8010 / address mask: H'01
     Data: H'0000 0000 / data mask: H'0000 0000
     Bus cycle: instruction access (pre-instruction-execution), read (operand size not included in conditions)

   A user break is generated after execution of the instruction at address H'0000 0404 with ASID = H'80, or before execution of an instruction at addresses H'0000 8000 to H'0000 83FE with ASID = H'70.
2. Register settings: BASRA = H'80 / BARA = H'0003 7226 / BAMRA = H'00 /
   BBRA = H'0016 / BASRB = H'70 / BARB = H'0003 722E / BAMRB = H'00 /
   BBRB = H'0016 / BDRB = H'00000000 / BDMRB = H'00000000 / BRCR = H'0008
   — Conditions set: Channel A → channel B sequential mode
   - Channel A: ASID: H'80 / address: H'0003 7226 / address mask: H'00
     Bus cycle: instruction access (pre-instruction-execution), read, word
   - Channel B: ASID: H'70 / address: H'0003 722E / address mask: H'00
     Data: H'0000 0000 / data mask: H'0000 0000
     Bus cycle: instruction access (pre-instruction-execution), read, word

   The instruction at address H'0003 7266 with ASID = H'80 is executed, then a user break is generated before execution of the instruction at address H'0003 722E with ASID = H'70.
3. Register settings: BASRA = H'80 / BARA = H'0002 7128 / BAMRA = H'00 /
   BBRA = H'001A / BASRB = H'70 / BARB = H'0003 1415 / BAMRB = H'00 /
   BBRB = H'0014 / BDRB = H'0000 0000 / BDMRB = H'0000 0000 / BRCR = H'0000
   — Conditions set: Independent channel A/channel B mode
   - Channel A: ASID: H'80 / address: H'0002 7128 / address mask: H'00
     Bus cycle: CPU, instruction access (pre-instruction-execution), write, word
   - Channel B: ASID: H'70 / address: H'0003 1415 / address mask: H'00
     Data: H'0000 0000 / data mask: H'0000 0000

included in conditions)

A user break interrupt is not generated on channel A since the instruction access is not a write cycle.

A user break interrupt is not generated on channel B since instruction access is performed on an even address.

(2) Operand Access Cycle Break Condition Settings

1. Register settings: BASRA = H'80 / BARA = H'0012 3456 / BAMRA = H'00 / BBRA = H'0024 / BASRB = H'70/ BARB = H'000A BCDE / BAMRB = H'02 / BBRB = H'002A / BDRB = H'0000 A512 / BDMRB = H'0000 0000 / BRCR = H'0080
   — Conditions set: Independent channel A/channel B mode
   - Channel A: ASID: H'80 / address: H'0012 3456 / address mask: H'00
     Bus cycle: operand access, read (operand size not included in conditions)
   - Channel B: ASID: H'70 / address: H'000A BCDE / address mask: H'02
     Data: H'0000 A512 / data mask: H'0000 0000
     Bus cycle: operand access, write, word
     Data break enabled

On channel A, a user break interrupt is generated in the event of a longword read at address H'0012 3454, a word read at address H'0012 3456, or a byte read at address H'0012 3456, with ASID = H'80.

On channel B, a user break interrupt is generated when H'A512 is written by word access to any address from H'000A B000 to H'000A BFFE with ASID = H'70.

In this LSI, this function stops the clock supplied to the user break controller and is used to minimize power consumption when the chip is operating. Note that, if you use this function, you cannot use the user break controller.

31.7.1 Transition to User Break Controller Stopped State

Setting bit MSTP5 of the STBCR2 (inside the CPG) to 1 stops the clock supply and causes the user break controller to enter the stopped state. Follow steps 1 to 5 below to set bit MSTP5 to 1 and enter the stopped state.

1. Initialize BBRA and BBRB to 0;
2. Initialize BRCR to 0;
3. Make a dummy read of BRCR;
4. Read STBCR2, then set bit MSTP5 in the read data to 1 and write the modified data back.
5. Make two dummy reads of STBCR2.

Make sure that, if an exception or interrupt occurs while performing steps 1 to 5, you do not change the values of these registers in the exception handling routine.

Do not read from or write to BARA, BAMRA, BBRA, BARB, BAMRB, BBRB, BDRB, BDMRB, and BRCR registers while the UBC clock is stopped. If the registers are read from or written to, the value cannot be guaranteed.

31.7.2 Cancelling the User Break Controller Stopped State

The clock supply can be restarted by setting bit MSTP5 of STBCR2 (inside the CPG) to 0. The user break controller can then be operated again. Follow steps 1 and 2 below to clear bit MSTP5 to 0 to cancel the stopped state.

1. Read STBCR2, then clear bit MSTP5 in the read data to 0 and write the modified data back;
2. Make two dummy reads of STBGR2.

As with the transition to the stopped state, if an exception or interrupt occurs while processing steps 1 and 2, make sure that the values in these registers are not changed in the exception handling routine.

The following are example programs:

```
; Transition to user break controller stopped state
; (1) Initialize BBRA and BBRB to 0.
      mov     #0, R0
      mov.l   #BBRA, R1
      mov.w   R0, @R1
      mov.l   #BBRB, R1
      mov.w   R0, @R1
; (2) Initialize BRCR to 0.
      mov.l   #BRCR, R1
      mov.w   R0, @R1
; (3) Dummy read BRCR.
      mov.w   @R1, R0
; (4) Read STBCR2, then set bit MSTP5 in the read data to 1 and write
      it back
      mov.l   #STBCR2, R1
      mov.b   @R1, R0
      or      #H'1, R0
      mov.b   R0, @R1
; (5) Twice dummy read STBCR2.
      mov.b   @R1, R0
      mov.b   @R1, R0
; Canceling user break controller stopped state
; (6) Read STBCR2, then clear bit MSTP5 in the read data to 0 and write
      it back
      mov.l   #STBCR2, R1
      mov.b   @R1, R0
      and     #H'FE, R0
      mov.b   R0, @R1
; (7) Twice dummy read STBCR2.
      mov.b   @R1, R0
```

mov.b @R1, R0

This section gives information on the on-chip I/O registers and is configured as described below.

1. Register Addresses (by functional module, in order of the corresponding section numbers)
   - Descriptions by functional module, in order of the corresponding section numbers
     Entries that consist of — lines are for separation of the functional modules.
   - Access to reserved addresses which are not described in this list is prohibited.
   - When registers consist of 16 or 32 bits, the addresses of the MSBs are given, on the presumption of a big-endian system.

2. Register Bits
   - Bit configurations of the registers are described in the same order as the Register Addresses (by functional module, in order of the corresponding section numbers).
   - Reserved bits are indicated by — in the bit name.
   - No entry in the bit-name column indicates that the whole register is allocated as a counter or for holding data.
   - When registers consist of 16 or 32 bits, bits are described from the MSB side.
     The order in which bytes are described is on the presumption of a big-endian system.

3. Register States in Each Operating Mode
   - Register states are described in the same order as the Register Addresses (by functional module, in order of the corresponding section numbers).
   - For the initial state of each bit, refer to the description of the register in the corresponding section.
   - The register states described are for the basic operating modes. If there is a specific reset for an on-chip module, refer to the section on that on-chip module.

(by functional module, in order of the corresponding section numbers)

Entries under Access size indicates numbers of bits.

Note: Access to undefined or reserved addresses is prohibited. Since operation or continued operation is not guaranteed when these registers are accessed, do not attempt such access.

Module	Register Name	Abbrev.	P4 Address	Area 7 Address	Size	Sync Clock
MMU	Page table entry high register	PTEH	H'FF00 0000	H'1F00 0000	32	Ick
	Page table entry low register	PTEL	H'FF00 0004	H'1F00 0004	32	Ick
	Page table entry assistance register	PTEA	H'FF00 0034	H'1F00 0034	32	Ick
	Translation table base register	TTB	H'FF00 0008	H'1F00 0008	32	Ick
	TLB exception address register	TEA	H'FF00 000C	H'1F00 000C	32	Ick
	MMU control register	MMUCR	H'FF00 0010	H'1F00 0010	32	Ick
Cache	Cache control register	CCR	H'FF00 001C	H'1F00 001C	32	Ick
	Queue address control register 0	QACR0	H'FF00 0038	H'1F00 0038	32	Ick
	Queue address control register 1	QACR1	H'FF00 003C	H'1F00 003C	32	Ick
Exception handling	TRAPA exception register	TRA	H'FF00 0020	H'1F00 0020	32	Ick
	Exception event register	EXPEVT	H'FF00 0024	H'1F00 0024	32	Ick
	Interrupt event register	INTEVT	H'FF00 0028	H'1F00 0028	32	Ick
INTC	Interrupt control register	ICR	H'FFD0 0000	H'1FD0 0000	16	Pck
	Interrupt priority level setting register A	IPRA	H'FFD0 0004	H'1FD0 0004	16	Pck
	Interrupt priority level setting register B	IPRB	H'FFD0 0008	H'1FD0 0008	16	Pck
	Interrupt priority level setting register C	IPRC	H'FFD0 000C	H'1FD0 000C	16	Pck
	Interrupt priority level setting register D	IPRD	H'FFD0 0010	H'1FD0 0010	16	Pck
	Interrupt priority level setting register 00	INTPRI00	H'FE08 0000	H'1E08 0000	32	Pck
	Interrupt priority level setting register 04	INTPRI04	H'FE08 0004	H'1E08 0004	32	Pck
	Interrupt priority level setting register 08	INTPRI08	H'FE08 0008	H'1E08 0008	32	Pck
	Interrupt priority level setting register 0C	INTPRI0C	H'FE08 000C	H'1E08 000C	32	Pck
	Interrupt source register 00	INTREQ00	H'FE08 0020	H'1E08 0020	32	Pck
	Interrupt source register 04	INTREQ04	H'FE08 0024	H'1E08 0024	32	Pck
	Interrupt mask register 00	INTMSK00	H'FE08 0040	H'1E08 0040	32	Pck

INTC	Interrupt mask register 04	INTMSK04	H'FE08 0044	H'1E08 0044	32	Pck
	Interrupt mask clear register 00	INTMSKCLR00	H'FE08 0060	H'1E08 0060	32	Pck
	Interrupt mask clear register 04	INTMSKCLR04	H'FE08 0064	H'1E08 0064	32	Pck
BSC	Bus control register 1	BCR1	H'FF80 0000	H'1F80 0000	32	Bck
	Bus control register 2	BCR2	H'FF80 0004	H'1F80 0004	16	Bck
	Bus control register 3	BCR3	H'FF80 0050	H'1F80 0050	16	Bck
	Bus control register 4	BCR4	H'FE0A 00F0	H'1E0A 00F0	32	Bck
	Wait control register 1	WCR1	H'FF80 0008	H'1F80 0008	32	Bck
	Wait control register 2	WCR2	H'FF80 000C	H'1F80 000C	32	Bck
	Wait control register 3	WCR3	H'FF80 0010	H'1F80 0010	32	Bck
	Wait control register 4	WCR4	H'FE0A 0028	H'1E0A 0028	32	Bck
	Memory control register	MCR	H'FF80 0014	H'1F80 0014	32	Bck
	PCMCIA control register	PCR	H'FF80 0018	H'1F80 0018	16	Bck
	Refresh timer control/status register	RTCSR	H'FF80 001C	H'1F80 001C	16	Bck
	Refresh timer counter	RTCNT	H'FF80 0020	H'1F80 0020	16	Bck
	Refresh time constant register	RTCOR	H'FF80 0024	H'1F80 0024	16	Bck
	Refresh count register	RFCR	H'FF80 0028	H'1F80 0028	16	Bck
	Synchronous DRAM mode register (for Area 2)	SDMR2	H'FF90 xxxx*[1]	H'1F90 xxxx	8	Bck
	Synchronous DRAM mode register (for Area 3)	SDMR3	H'FF94 xxxx*[1]	H'1F94 xxxx	8	Bck
DMAC Channel 0	DMA source address register 0	SAR0	H'FFA0 0000	H'1FA0 0000	32	Bck
	DMA destination address register 0	DAR0	H'FFA0 0004	H'1FA0 0004	32	Bck
	DMA transfer count register 0	DMATCR0	H'FFA0 0008	H'1FA0 0008	32	Bck
	DMA channel control register 0	CHCR0	H'FFA0 000C	H'1FA0 000C	32	Bck
DMAC Channel 1	DMA source address register 1	SAR1	H'FFA0 0010	H'1FA0 0010	32	Bck
	DMA destination address register 1	DAR1	H'FFA0 0014	H'1FA0 0014	32	Bck
	DMA transfer count register 1	DMATCR1	H'FFA0 0018	H'1FA0 0018	32	Bck
	DMA channel control register 1	CHCR1	H'FFA0 001C	H'1FA0 001C	32	Bck
DMAC Channel 2	DMA source address register 2	SAR2	H'FFA0 0020	H'1FA0 0020	32	Bck
	DMA destination address register 2	DAR2	H'FFA0 0024	H'1FA0 0024	32	Bck
	DMA transfer count register 2	DMATCR2	H'FFA0 0028	H'1FA0 0028	32	Bck
	DMA channel control register 2	CHCR2	H'FFA0 002C	H'1FA0 002C	32	Bck

DMAC Channel 3	DMA source address register 3	SAR3	H'FFA0 0030	H'1FA0 0030	32	Bck
	DMA destination address register 3	DAR3	H'FFA0 0034	H'1FA0 0034	32	Bck
	DMA transfer count register 3	DMATCR3	H'FFA0 0038	H'1FA0 0038	32	Bck
	DMA channel control register 3	CHCR3	H'FFA0 003C	H'1FA0 003C	32	Bck
DMAC Channel 4	DMA source address register 4	SAR4	H'FFA0 0050	H'1FA0 0050	32	Bck
	DMA destination address register 4	DAR4	H'FFA0 0054	H'1FA0 0054	32	Bck
	DMA transfer count register 4	DMATCR4	H'FFA0 0058	H'1FA0 0058	32	Bck
	DMA channel control register 4	CHCR4	H'FFA0 005C	H'1FA0 005C	32	Bck
DMAC Channel 5	DMA source address register 5	SAR5	H'FFA0 0060	H'1FA0 0060	32	Bck
	DMA destination address register 5	DAR5	H'FFA0 0064	H'1FA0 0064	32	Bck
	DMA transfer count register 5	DMATCR5	H'FFA0 0068	H'1FA0 0068	32	Bck
	DMA channel control register 5	CHCR5	H'FFA0 006C	H'1FA0 006C	32	Bck
DMAC Channel 6	DMA source address register 6	SAR6	H'FFA0 0070	H'1FA0 0070	32	Bck
	DMA destination address register 6	DAR6	H'FFA0 0074	H'1FA0 0074	32	Bck
	DMA transfer count register 6	DMATCR6	H'FFA0 0078	H'1FA0 0078	32	Bck
	DMA channel control register 6	CHCR6	H'FFA0 007C	H'1FA0 007C	32	Bck
DMAC Channel 7	DMA source address register 7	SAR7	H'FFA0 0080	H'1FA0 0080	32	Bck
	DMA destination address register 7	DAR7	H'FFA0 0084	H'1FA0 0084	32	Bck
	DMA transfer count register 7	DMATCR7	H'FFA0 0088	H'1FA0 0088	32	Bck
	DMA channel control register 7	CHCR7	H'FFA0 008C	H'1FA0 008C	32	Bck
DMAC common	DMA operation register	DMAOR	H'FFA0 0040	H'1FA0 0040	32	Bck
	DMA request resource selection register A	DMARSRA	H'FE09 0000	H'1E09 0000	32	Pck
	DMA request resource selection register B	DMARSRB	H'FE09 0004	H'1E09 0004	32	Pck
DMAC DMABRG	DMA request control register	DMARCR	H'FE09 0008	H'1E09 0008	32	Pck
	DMA BRG control register	DMABRGCR	H'FE3C 0000	H'1E3C 0000	32	Pck
	DMA audio source address register 0	DMAATXSAR0	H'FE3C 0040	H'1E3C 0040	32	Pck
	DMA audio destination address register 0	DMAARXDAR0	H'FE3C 0044	H'1E3C 0044	32	Pck
	DMA audio transmit transfer count register 0	DMAATXTCR0	H'FE3C 0048	H'1E3C 0048	32	Pck
	DMA audio receive transfer count register 0	DMAARXTCR0	H'FE3C 004C	H'1E3C 004C	32	Pck
	DMA audio control register 0	DMAACR0	H'FE3C 0050	H'1E3C 0050	32	Pck
	DMA audio transmit transfer count register 0	DMAATXTCNT0	H'FE3C 0054	H'1E3C 0054	32	Pck
	DMA audio receive transfer count register 0	DMAARXTCNT0	H'FE3C 0058	H'1E3C 0058	32	Pck

DMAC DMABRG	DMA audio source address register 1	DMAATXSAR1	H'FE3C 0060	H'1E3C 0060	32	Pck
	DMA audio destination address register 1	DMAARXDAR1	H'FE3C 0064	H'1E3C 0064	32	Pck
	DMA audio transmit transfer count register 1	DMAATXTCR1	H'FE3C 0068	H'1E3C 0068	32	Pck
	DMA audio receive transfer count register 1	DMAARXTCR1	H'FE3C 006C	H'1E3C 006C	32	Pck
	DMA audio control register 1	DMAACR1	H'FE3C 0070	H'1E3C 0070	32	Pck
	DMA audio transmit transfer count register 1	DMAATXTCNT1	H'FE3C 0074	H'1E3C 0074	32	Pck
	DMA audio receive transfer count register 1	DMAARXTCNT1	H'FE3C 0078	H'1E3C 0078	32	Pck
	DAM USB source address register	DMAUSAR	H'FE3C 0080	H'1E3C 0080	32	Pck
	DMA USB destination address register	DMAUDAR	H'FE3C 0084	H'1E3C 0084	32	Pck
	DMA USB R/W size register	DMAURWSZ	H'FE3C 0088	H'1E3C 0088	32	Pck
	DMA USB control register	DMAUCR	H'FE3C 008C	H'1E3C 008C	32	Pck
CPG	Frequency control register	FRQCR	H'FFC0 0000	H'1FC0 0000	16	Pck
	Clock division register	DCKDR	H'FE0A 0020	H'1E0A 0020	32	Pck
	Module clock control register	MCKCR	H'FE0A 0024	H'1E0A 0024	32	Pck
WDT	Watchdog timer counter	WTCNT	H'FFC0 0008	H'1FC0 0008	8/16*2	Pck
	Watchdog timer control/status register	WTCSR	H'FFC0 000C	H'1FC0 000C	8/16*2	Pck
Power-down	Standby control register	STBCR	H'FFC0 0004	H'1FC0 0004	8	Pck
	Standby control register 2	STBCR2	H'FFC0 0010	H'1FC0 0010	8	Pck
	Clock stop register 00	CLKSTP00	H'FE0A 0000	H'1E0A 0000	32	Pck
	Clock stop clear register 00	CLKSTPCLR00	H'FE0A 0010	H'1E0A 0010	32	Pck
TMU common	Timer start register	TSTR	H'FFD8 0004	H'1FD8 0004	8	Pck
TMU Channel 0	Timer constant register 0	TCOR0	H'FFD8 0008	H'1FD8 0008	32	Pck
	Timer counter 0	TCNT0	H'FFD8 000C	H'1FD8 000C	32	Pck
	Timer control register 0	TCR0	H'FFD8 0010	H'1FD8 0010	16	Pck
TMU Channel 1	Timer constant register 1	TCOR1	H'FFD8 0014	H'1FD8 0014	32	Pck
	Timer counter 1	TCNT1	H'FFD8 0018	H'1FD8 0018	32	Pck
	Timer control register 1	TCR1	H'FFD8 001C	H'1FD8 001C	16	Pck
TMU Channel 2	Timer constant register 2	TCOR2	H'FFD8 0020	H'1FD8 0020	32	Pck
	Timer counter 2	TCNT2	H'FFD8 0024	H'1FD8 0024	32	Pck
	Timer control register 2	TCR2	H'FFD8 0028	H'1FD8 0028	16	Pck
	Input capture register 2	TCPR2	H'FFD8 002C	H'1FD8 002C	32	Pck

CMT common	Configuration register	CMTCFG	H'FE1C 0000	H'1E1C 0000	32	Pck
	Free-running timer	CMTFRT	H'FE1C 0004	H'1E1C 0004	32	Pck
	Control register	CMTCTL	H'FE1C 0008	H'1E1C 0008	32	Pck
	IRQ status register	CMTIRQS	H'FE1C 000C	H'1E1C 000C	32	Pck
CMT Channel 0	Channel 0 time register	CMTCH0T	H'FE1C 0010	H'1E1C 0010	32	Pck
	Channel 0 stop time register	CMTCH0ST	H'FE1C 0020	H'1E1C 0020	32	Pck
	Channel 0 timer/counter	CMTCH0C	H'FE1C 0030	H'1E1C 0030	32	Pck
CMT Channel 1	Channel 1 time register	CMTCH1T	H'FE1C 0014	H'1E1C 0014	32	Pck
	Channel 1 stop time register	CMTCH1ST	H'FE1C 0024	H'1E1C 0024	32	Pck
	Channel 1 timer/counter	CMTCH1C	H'FE1C 0034	H'1E1C 0034	32	Pck
CMT Channel 2	Channel 2 time register	CMTCH2T	H'FE1C 0018	H'1E1C 0018	32	Pck
	Channel 2 stop time register	CMTCH2ST	H'FE1C 0028	H'1E1C 0028	32	Pck
	Channel 2 timer/counter	CMTCH2C	H'FE1C 0038	H'1E1C 0038	32	Pck
CMT Channel 3	Channel 3 time register	CMTCH3T	H'FE1C 001C	H'1E1C 001C	32	Pck
	Channel 3 stop time register	CMTCH3ST	H'FE1C 002C	H'1E1C 002C	32	Pck
	Channel 3 timer/counter	CMTCH3C	H'FE1C 003C	H'1E1C 003C	32	Pck
SCIF Channel 0	Serial mode register 0	SCSMR0	H'FE60 0000	H'1E60 0000	16	Pck
	Bit rate register 0	SCBRR0	H'FE60 0004	H'1E60 0004	8	Pck
	Serial control register 0	SCSCR0	H'FE60 0008	H'1E60 0008	16	Pck
	Transmit FIFO data register 0	SCFTDR0	H'FE60 000C	H'1E60 000C	8	Pck
	Serial status register 0	SCFSR0	H'FE60 0010	H'1E60 0010	16	Pck
	Receive FIFO data register 0	SCFRDR0	H'FE60 0014	H'1E60 0014	8	Pck
	FIFO control register 0	SCFCR0	H'FE60 0018	H'1E60 0018	16	Pck
	Transmit FIFO data count register 0	SCTFDR0	H'FE60 001C	H'1E60 001C	16	Pck
	Receive FIFO data count register 0	SCRFDR0	H'FE60 0020	H'1E60 0020	16	Pck
	Serial port register 0	SCSPTR0	H'FE60 0024	H'1E60 0024	16	Pck
	Line status register 0	SCLSR0	H'FE60 0028	H'1E60 0028	16	Pck
	Serial error register 0	SCRER0	H'FE60 002C	H'1E60 002C	16	Pck
SCIF Channel 1	Serial mode register 1	SCSMR1	H'FE61 0000	H'1E61 0000	16	Pck
	Bit rate register 1	SCBRR1	H'FE61 0004	H'1E61 0004	8	Pck
	Serial control register 1	SCSCR1	H'FE61 0008	H'1E61 0008	16	Pck
	Transmit FIFO data register 1	SCFTDR1	H'FE61 000C	H'1E61 000C	8	Pck

SCIF Channel 1	Serial status register 1	SCFSR1	H'FE61 0010	H'1E61 0010	16	Pck
	Receive FIFO data register 1	SCFRDR1	H'FE61 0014	H'1E61 0014	8	Pck
	FIFO control register 1	SCFCR1	H'FE61 0018	H'1E61 0018	16	Pck
	Transmit FIFO data count register 1	SCTFDR1	H'FE61 001C	H'1E61 001C	16	Pck
	Receive FIFO data count register 1	SCRFDR1	H'FE61 0020	H'1E61 0020	16	Pck
	Serial port register 1	SCSPTR1	H'FE61 0024	H'1E61 0024	16	Pck
	Line status register 1	SCLSR1	H'FE61 0028	H'1E61 0028	16	Pck
	Serial error register 1	SCRER1	H'FE61 002C	H'1E61 002C	16	Pck
SCIF Channel.2	Serial mode register 2	SCSMR2	H'FE62 0000	H'1E62 0000	16	Pck
	Bit rate register 2	SCBRR2	H'FE62 0004	H'1E62 0004	8	Pck
	Serial control register 2	SCSCR2	H'FE62 0008	H'1E62 0008	16	Pck
	Transmit FIFO data register 2	SCFTDR2	H'FE62 000C	H'1E62 000C	8	Pck
	Serial status register 2	SCFSR2	H'FE62 0010	H'1E62 0010	16	Pck
	Receive FIFO data register 2	SCFRDR2	H'FE62 0014	H'1E62 0014	8	Pck
	FIFO control register 2	SCFCR2	H'FE62 0018	H'1E62 0018	16	Pck
	Transmit FIFO data count register 2	SCTFDR2	H'FE62 001C	H'1E62 001C	16	Pck
	Receive FIFO data count register 2	SCRFDR2	H'FE62 0020	H'1E62 0020	16	Pck
	Serial port register 2	SCSPTR2	H'FE62 0024	H'1E62 0024	16	Pck
	Line status register 2	SCLSR2	H'FE62 0028	H'1E62 0028	16	Pck
	Serial error register 2	SCRER2	H'FE62 002C	H'1E62 002C	16	Pck
SIM	Serial mode register	SISMR	H'FE48 0000	H'1E48 0000	8	Pck
	Bit rate register	SIBRR	H'FE48 0002	H'1E48 0002	8	Pck
	Serial control register	SISCR	H'FE48 0004	H'1E48 0004	8	Pck
	Transmit data register	SITDR	H'FE48 0006	H'1E48 0006	8	Pck
	Serial status register	SISSR	H'FE48 0008	H'1E48 0008	8	Pck
	Receive data register	SIRDR	H'FE48 000A	H'1E48 000A	8	Pck
	Smart card mode register	SISCMR	H'FE48 000C	H'1E48 000C	8	Pck
	Serial control 2 register	SISC2R	H'FE48 000E	H'1E48 000E	8	Pck
	Wait time register	SIWAIT	H'FE48 0010	H'1E48 0010	16	Pck
	Guard extension register	SIGRD	H'FE48 0012	H'1E48 0012	8	Pck
	Sampling register	SISMPL	H'FE48 0014	H'1E48 0014	16	Pck

I²C Channel 0	Slave control register 0	ICSCR0	H'FE14 0000	H'1E14 0000	32	Pck
	Master control register 0	ICMCR0	H'FE14 0004	H'1E14 0004	32	Pck
	Slave status register 0	ICSSR0	H'FE14 0008	H'1E14 0008	32	Pck
	Master status register 0	ICMSR0	H'FE14 000C	H'1E14 000C	32	Pck
	Slave interrupt enable register 0	ICSIER0	H'FE14 0010	H'1E14 0010	32	Pck
	Master interrupt enable register 0	ICMIER0	H'FE14 0014	H'1E14 0014	32	Pck
	Clock control register 0	ICCCR0	H'FE14 0018	H'1E14 0018	32	Pck
	Slave address enable register 0	ICSAR0	H'FE14 001C	H'1E14 001C	32	Pck
	Master address enable register 0	ICMAR0	H'FE14 0020	H'1E14 0020	32	Pck
	Receive data register 0	ICRXD0	H'FE14 0024	H'1E14 0024	32	Pck
	Transmit data register 0	ICTXD0	H'FE14 0024	H'1E14 0024	32	Pck
	FIFO control register 0	ICFCR0	H'FE14 0028	H'1E14 0028	32	Pck
	FIFO status register 0	ICFSR0	H'FE14 002C	H'1E14 002C	32	Pck
	FIFO interrupt enable register 0	ICFIER0	H'FE14 0030	H'1E14 0030	32	Pck
	Receive FIFO data count register 0	ICRFDR0	H'FE14 0034	H'1E14 0034	32	Pck
	Transmit FIFO data count register 0	ICTFDR0	H'FE14 0038	H'1E14 0038	32	Pck
I²C Channel 1	Slave control register 1	ICSCR1	H'FE15 0000	H'1E15 0000	32	Pck
	Master control register 1	ICMCR1	H'FE15 0004	H'1E15 0004	32	Pck
	Slave status register 1	ICSSR1	H'FE15 0008	H'1E15 0008	32	Pck
	Master status register 1	ICMSR1	H'FE15 000C	H'1E15 000C	32	Pck
	Slave interrupt enable register 1	ICSIER1	H'FE15 0010	H'1E15 0010	32	Pck
	Master interrupt enable register 1	ICMIER1	H'FE15 0014	H'1E15 0014	32	Pck
	Clock control register 1	ICCCR1	H'FE15 0018	H'1E15 0018	32	Pck
	Slave address enable register 1	ICSAR1	H'FE15 001C	H'1E15 001C	32	Pck
	Master address enable register 1	ICMAR1	H'FE15 0020	H'1E15 0020	32	Pck
	Receive data register 1	ICRXD1	H'FE15 0024	H'1E15 0024	32	Pck
	Transmit data register 1	ICTXD1	H'FE15 0024	H'1E15 0024	32	Pck
	FIFO control register 1	ICFCR1	H'FE15 0028	H'1E15 0028	32	Pck
	FIFO status register 1	ICFSR1	H'FE15 002C	H'1E15 002C	32	Pck
	FIFO interrupt enable register 1	ICFIER1	H'FE15 0030	H'1E15 0030	32	Pck
	Receive FIFO data count register 1	ICRFDR1	H'FE15 0034	H'1E15 0034	32	Pck
	Transmit FIFO data count register 1	ICTFDR1	H'FE15 0038	H'1E15 0038	32	Pck

SSI Channel 0	Control register 0	SSICR0	H'FE68 0000	H'1E68 0000	32	Pck
	Status register 0	SSISR0	H'FE68 0004	H'1E68 0004	32	Pck
	Transmit data register 0	SSITDR0	H'FE68 0008	H'1E68 0008	32	Pck
	Receive data register 0	SSIRDR0	H'FE68 000C	H'1E68 000C	32	Pck
SSI Channel 1	Control register 1	SSICR1	H'FE69 0000	H'1E69 0000	32	Pck
	Status register 1	SSISR1	H'FE69 0004	H'1E69 0004	32	Pck
	Transmit data register 1	SSITDR1	H'FE69 0008	H'1E69 0008	32	Pck
	Receive data register 1	SSIRDR1	H'FE69 000C	H'1E69 000C	32	Pck
USB	Host controller interface revision register	HcRevision	H'FE34 0000	H'1E34 0000	32	Pck
	Control register	HcControl	H'FE34 0004	H'1E34 0004	32	Pck
	Command status register	HcCommandStatus	H'FE34 0008	H'1E34 0008	32	Pck
	Interrupt status register	HcInterruptStatus	H'FE34 000C	H'1E34 000C	32	Pck
	Interrupt enable register	HcInterruptEnable	H'FE34 0010	H'1E34 0010	32	Pck
	Interrupt disable register	HcInterruptDisable	H'FE34 0014	H'1E34 0014	32	Pck
	Host controller communication area pointer register	HcHCCA	H'FE34 0018	H'1E34 0018	32	Pck
	Period current ED pointer register	HcPeriodCurrentED	H'FE34 001C	H'1E34 001C	32	Pck
	Control head ED pointer register	HcControlHeadED	H'FE34 0020	H'1E34 0020	32	Pck
	Control current ED pointer register	HcControlCurrentED	H'FE34 0024	H'1E34 0024	32	Pck
	Bulk head ED pointer register	HcBulkHeadED	H'FE34 0028	H'1E34 0028	32	Pck
	Bulk current ED pointer register	HcBulkCurrentED	H'FE34 002C	H'1E34 002C	32	Pck
	Done queue head pointer register	HcDoneHead	H'FE34 0030	H'1E34 0030	32	Pck
	Frame interval register	HcFmInterval	H'FE34 0034	H'1E34 0034	32	Pck
	Frame remaining register	HcFmRemaining	H'FE34 0038	H'1E34 0038	32	Pck
	Frame number register	HcFmNumber	H'FE34 003C	H'1E34 003C	32	Pck
	Periodic start register	HcPeriodicStart	H'FE34 0040	H'1E34 0040	32	Pck
	Low speed threshold register	HcLSThreshold	H'FE34 0044	H'1E34 0044	32	Pck
	Root hub descriptor A register	HcRhDescriptorA	H'FE34 0048	H'1E34 0048	32	Pck
	Root hub descriptor B register	HcRhDescriptorB	H'FE34 004C	H'1E34 004C	32	Pck
	Root hub status register	HcRhStatus	H'FE34 0050	H'1E34 0050	32	Pck
	Root hub port status 1 register	HcRhPortStatus1	H'FE34 0054	H'1E34 0054	32	Pck

USB		Shared memory area Start	H'FE34 1000	H'1E34 1000	32	Pck
		Shared memory area End	H'FE34 2FFF	H'1E34 2FFF	32	Pck
HCAN2 Channel 0	Master control register	CAN0MCR	H'FE38 0000	H'1E38 0000	16	Pck
	General status register	CAN0GSR	H'FE38 0002	H'1E38 0002	16	Pck
	Bit configuration register 1	CAN0BCR1	H'FE38 0004	H'1E38 0004	16	Pck
	Bit configuration register 0	CAN0BCR0	H'FE38 0006	H'1E38 0006	16	Pck
	Interrupt request register	CAN0IRR	H'FE38 0008	H'1E38 0008	16	Pck
	Interrupt mask register	CAN0IMR	H'FE38 000A	H'1E38 000A	16	Pck
	Error counter	CAN0TECREC	H'FE38 000C	H'1E38 000C	16	Pck
	Transmit pending request register 1	CAN0TXPR1	H'FE38 0020	H'1E38 0020	16	Pck
	Transmit pending request register 0	CAN0TXPR0	H'FE38 0022	H'1E38 0022	16	Pck
	Transmit cancel register 1	CAN0TXCR1	H'FE38 0028	H'1E38 0028	16	Pck
	Transmit cancel register 0	CAN0TXCR0	H'FE38 002A	H'1E38 002A	16	Pck
	Transmit acknowledge register 1	CAN0TXACK1	H'FE38 0030	H'1E38 0030	16	Pck
	Transmit acknowledge register 0	CAN0TXACK0	H'FE38 0032	H'1E38 0032	16	Pck
	Abort acknowledge register 1	CAN0ABACK1	H'FE38 0038	H'1E38 0038	16	Pck
	Abort acknowledge register 0	CAN0ABACK0	H'FE38 003A	H'1E38 003A	16	Pck
	Receive data frame pending register 1	CAN0RXPR1	H'FE38 0040	H'1E38 0040	16	Pck
	Receive data frame pending register 0	CAN0RXPR0	H'FE38 0042	H'1E38 0042	16	Pck
	Remote frame request pending register 1	CAN0RFPR1	H'FE38 0048	H'1E38 0048	16	Pck
	Remote frame request pending register 0	CAN0RFPR0	H'FE38 004A	H'1E38 004A	16	Pck
	Mailbox interrupt mask register 1	CAN0MBIMR1	H'FE38 0050	H'1E38 0050	16	Pck
	Mailbox interrupt mask register 0	CAN0MBIMR0	H'FE38 0052	H'1E38 0052	16	Pck
	Unread message status register 1	CAN0UMSR1	H'FE38 0058	H'1E38 0058	16	Pck
	Unread message status register 0	CAN0UMSR0	H'FE38 005A	H'1E38 005A	16	Pck
	Timer counter register	CAN0TCNTR	H'FE38 0080	H'1E38 0080	16	Pck
	Timer control register	CAN0TCR	H'FE38 0082	H'1E38 0082	16	Pck
	Timer compare match register	CAN0TCMR	H'FE38 0090	H'1E38 0090	16	Pck

HCAN2 Channel 0	Mailbox 0	CAN0MB0	H'FE38 0100	H'1E38 0100	16	Pck
	Mailbox 1	CAN0MB1	H'FE38 0120	H'1E38 0120	16	Pck
	Mailbox 2	CAN0MB2	H'FE38 0140	H'1E38 0140	16	Pck
	Mailbox 3	CAN0MB3	H'FE38 0160	H'1E38 0160	16	Pck
	Mailbox 4	CAN0MB4	H'FE38 0180	H'1E38 0180	16	Pck
	Mailbox 5	CAN0MB5	H'FE38 01A0	H'1E38 01A0	16	Pck
	Mailbox 6	CAN0MB6	H'FE38 01C0	H'1E38 01C0	16	Pck
	Mailbox 7	CAN0MB7	H'FE38 01E0	H'1E38 01E0	16	Pck
	Mailbox 8	CAN0MB8	H'FE38 0200	H'1E38 0200	16	Pck
	Mailbox 9	CAN0MB9	H'FE38 0220	H'1E38 0220	16	Pck
	Mailbox 10	CAN0MB10	H'FE38 0240	H'1E38 0240	16	Pck
	Mailbox 11	CAN0MB11	H'FE38 0260	H'1E38 0260	16	Pck
	Mailbox 12	CAN0MB12	H'FE38 0280	H'1E38 0280	16	Pck
	Mailbox 13	CAN0MB13	H'FE38 02A0	H'1E38 02A0	16	Pck
	Mailbox 14	CAN0MB14	H'FE38 02C0	H'1E38 02C0	16	Pck
	Mailbox 15	CAN0MB15	H'FE38 02E0	H'1E38 02E0	16	Pck
	Mailbox 16	CAN0MB16	H'FE38 0300	H'1E38 0300	16	Pck
	Mailbox 17	CAN0MB17	H'FE38 0320	H'1E38 0320	16	Pck
	Mailbox 18	CAN0MB18	H'FE38 0340	H'1E38 0340	16	Pck
	Mailbox 19	CAN0MB19	H'FE38 0360	H'1E38 0360	16	Pck
	Mailbox 20	CAN0MB20	H'FE38 0380	H'1E38 0380	16	Pck
	Mailbox 21	CAN0MB21	H'FE38 03A0	H'1E38 03A0	16	Pck
	Mailbox 22	CAN0MB22	H'FE38 03C0	H'1E38 03C0	16	Pck
	Mailbox 23	CAN0MB23	H'FE38 03E0	H'1E38 03E0	16	Pck
	Mailbox 24	CAN0MB24	H'FE38 0400	H'1E38 0400	16	Pck
	Mailbox 25	CAN0MB25	H'FE38 0420	H'1E38 0420	16	Pck
	Mailbox 26	CAN0MB26	H'FE38 0440	H'1E38 0440	16	Pck
	Mailbox 27	CAN0MB27	H'FE38 0460	H'1E38 0460	16	Pck
	Mailbox 28	CAN0MB28	H'FE38 0480	H'1E38 0480	16	Pck
	Mailbox 29	CAN0MB29	H'FE38 04A0	H'1E38 04A0	16	Pck
	Mailbox 30	CAN0MB30	H'FE38 04C0	H'1E38 04C0	16	Pck
	Mailbox 31	CAN0MB31	H'FE38 04E0	H'1E38 04E0	16	Pck

HCAN2 Channel 1	Master control register	CAN1MCR	H'FE39 0000	H'1E39 0000	16	Pck
	General status register	CAN1GSR	H'FE39 0002	H'1E39 0002	16	Pck
	Bit configuration register 1	CAN1BCR1	H'FE39 0004	H'1E39 0004	16	Pck
	Bit configuration register 0	CAN1BCR0	H'FE39 0006	H'1E39 0006	16	Pck
	Interrupt request register	CAN1IRR	H'FE39 0008	H'1E39 0008	16	Pck
	Interrupt mask register	CAN1IMR	H'FE39 000A	H'1E39 000A	16	Pck
	Error counter	CAN1TECREC	H'FE39 000C	H'1E39 000C	16	Pck
	Transmit pending request register 1	CAN1TXPR1	H'FE39 0020	H'1E39 0020	16	Pck
	Transmit pending request register 0	CAN1TXPR0	H'FE39 0022	H'1E39 0022	16	Pck
	Transmit cancel register 1	CAN1TXCR1	H'FE39 0028	H'1E39 0028	16	Pck
	Transmit cancel register 0	CAN1TXCR0	H'FE39 002A	H'1E39 002A	16	Pck
	Transmit acknowledge register 1	CAN1TXACK1	H'FE39 0030	H'1E39 0030	16	Pck
	Transmit acknowledge register 0	CAN1TXACK0	H'FE39 0032	H'1E39 0032	16	Pck
	Abort acknowledge register 1	CAN1ABACK1	H'FE39 0038	H'1E39 0038	16	Pck
	Abort acknowledge register 0	CAN1ABACK0	H'FE39 003A	H'1E39 003A	16	Pck
	Receive data frame pending register 1	CAN1RXPR1	H'FE39 0040	H'1E39 0040	16	Pck
	Receive data frame pending register 0	CAN1RXPR0	H'FE39 0042	H'1E39 0042	16	Pck
	Remote frame request pending register 1	CAN1RFPR1	H'FE39 0048	H'1E39 0048	16	Pck
	Remote frame request pending register 0	CAN1RFPR0	H'FE39 004A	H'1E39 004A	16	Pck
	Mailbox interrupt mask register 1	CAN1MBIMR1	H'FE39 0050	H'1E39 0050	16	Pck
	Mailbox interrupt mask register 0	CAN1MBIMR0	H'FE39 0052	H'1E39 0052	16	Pck
	Unread message status register 1	CAN1UMSR1	H'FE39 0058	H'1E39 0058	16	Pck
	Unread message status register 0	CAN1UMSR0	H'FE39 005A	H'1E39 005A	16	Pck
	Timer counter register	CAN1TCNTR	H'FE39 0080	H'1E39 0080	16	Pck
	Timer control register	CAN1TCR	H'FE39 0082	H'1E39 0082	16	Pck
	Timer compare match register	CAN1TCMR	H'FE39 0090	H'1E39 0090	16	Pck
	Mailbox 0	CAN1MB0	H'FE39 0100	H'1E39 0100	16	Pck
	Mailbox 1	CAN1MB1	H'FE39 0120	H'1E39 0120	16	Pck
	Mailbox 2	CAN1MB2	H'FE39 0140	H'1E39 0140	16	Pck
	Mailbox 3	CAN1MB3	H'FE39 0160	H'1E39 0160	16	Pck
	Mailbox 4	CAN1MB4	H'FE39 0180	H'1E39 0180	16	Pck
	Mailbox 5	CAN1MB5	H'FE39 01A0	H'1E39 01A0	16	Pck

HCAN2 Channel 1	Mailbox 6	CAN1MB6	H'FE39 01C0	H'1E39 01C0	16	Pck
	Mailbox 7	CAN1MB7	H'FE39 01E0	H'1E39 01E0	16	Pck
	Mailbox 8	CAN1MB8	H'FE39 0200	H'1E39 0200	16	Pck
	Mailbox 9	CAN1MB9	H'FE39 0220	H'1E39 0220	16	Pck
	Mailbox 10	CAN1MB10	H'FE39 0240	H'1E39 0240	16	Pck
	Mailbox 11	CAN1MB11	H'FE39 0260	H'1E39 0260	16	Pck
	Mailbox 12	CAN1MB12	H'FE39 0280	H'1E39 0280	16	Pck
	Mailbox 13	CAN1MB13	H'FE39 02A0	H'1E39 02A0	16	Pck
	Mailbox 14	CAN1MB14	H'FE39 02C0	H'1E39 02C0	16	Pck
	Mailbox 15	CAN1MB15	H'FE39 02E0	H'1E39 02E0	16	Pck
	Mailbox 16	CAN1MB16	H'FE39 0300	H'1E39 0300	16	Pck
	Mailbox 17	CAN1MB17	H'FE39 0320	H'1E39 0320	16	Pck
	Mailbox 18	CAN1MB18	H'FE39 0340	H'1E39 0340	16	Pck
	Mailbox 19	CAN1MB19	H'FE39 0360	H'1E39 0360	16	Pck
	Mailbox 20	CAN1MB20	H'FE39 0380	H'1E39 0380	16	Pck
	Mailbox 21	CAN1MB21	H'FE39 03A0	H'1E39 03A0	16	Pck
	Mailbox 22	CAN1MB22	H'FE39 03C0	H'1E39 03C0	16	Pck
	Mailbox 23	CAN1MB23	H'FE39 03E0	H'1E39 03E0	16	Pck
	Mailbox 24	CAN1MB24	H'FE39 0400	H'1E39 0400	16	Pck
	Mailbox 25	CAN1MB25	H'FE39 0420	H'1E39 0420	16	Pck
	Mailbox 26	CAN1MB26	H'FE39 0440	H'1E39 0440	16	Pck
	Mailbox 27	CAN1MB27	H'FE39 0460	H'1E39 0460	16	Pck
	Mailbox 28	CAN1MB28	H'FE39 0480	H'1E39 0480	16	Pck
	Mailbox 29	CAN1MB29	H'FE39 04A0	H'1E39 04A0	16	Pck
	Mailbox 30	CAN1MB30	H'FE39 04C0	H'1E39 04C0	16	Pck
	Mailbox 31	CAN1MB31	H'FE39 04E0	H'1E39 04E0	16	Pck

HSPI	Control register	SPCR	H'FE18 0000	H'1E18 0000	32	Pck
	Status register	SPSR	H'FE18 0004	H'1E18 0004	32	Pck
	System control register	SPSCR	H'FE18 0008	H'1E18 0008	32	Pck
	Transmit buffer register	SPTBR	H'FE18 000C	H'1E18 000C	32	Pck
	Receive buffer register	SPRBR	H'FE18 0010	H'1E18 0010	32	Pck
PFC	Input pin pull-up control register	INPUPA	H'FE40 0028	H'1E40 0028	16	Pck
	DMA pin control register	DMAPCR	H'FE40 002C	H'1E40 002C	16	Pck
	SCIF Hi-Z control register	SCIHZR	H'FE40 0030	H'1E40 0030	16	Pck
	Peripheral module select register	IPSELR	H'FE40 0034	H'1E40 0034	16	Pck
	Port A pull-up control register	PAPUPR	H'FE40 0080	H'1E40 0080	8	Pck
	Port B pull-up control register	PBPUPR	H'FE40 0084	H'1E40 0084	8	Pck
	Port C pull-up control register	PCPUPR	H'FE40 0088	H'1E40 0088	8	Pck
	Port D pull-up control register	PDPUPR	H'FE40 008C	H'1E40 008C	8	Pck
	Port E pull-up control register	PEPUPR	H'FE40 0090	H'1E40 0090	8	Pck
	Port F pull-up control register	PFPUPR	H'FE40 0094	H'1E40 0094	8	Pck
	Port G pull-up control register	PGPUPR	H'FE40 0098	H'1E40 0098	8	Pck
	Port H pull-up control register	PHPUPR	H'FE40 009C	H'1E40 009C	8	Pck
	Port J pull-up control register	PJPUPR	H'FE40 00A0	H'1E40 00A0	8	Pck
	Port K pull-up control register	PKPUPR	H'FE40 00A4	H'1E40 00A4	8	Pck
	Mode pin pull-up control register	MDPUPR	H'FE40 00A8	H'1E40 00A8	8	Pck
	Mode select register	MODSELR	H'FE40 00AC	H'1E40 00AC	8	Pck
	Port A control register	PACR	H'FE40 0000	H'1E40 0000	16	Pck
	Port B control register	PBCR	H'FE40 0004	H'1E40 0004	16	Pck
	Port C control register	PCCR	H'FE40 0008	H'1E40 0008	16	Pck
	Port D control register	PDCR	H'FE40 000C	H'1E40 000C	16	Pck
	Port E control register	PECR	H'FE40 0010	H'1E40 0010	16	Pck
	Port F control register	PFCR	H'FE40 0014	H'1E40 0014	16	Pck
	Port G control register	PGCR	H'FE40 0018	H'1E40 0018	16	Pck
	Port H control register	PHCR	H'FE40 001C	H'1E40 001C	16	Pck
	Port J control register	PJCR	H'FE40 0020	H'1E40 0020	16	Pck
	Port K control register	PKCR	H'FE40 0024	H'1E40 0024	16	Pck

PFC	Port A data register	PADR	H'FE40 0040	H'1E40 0040	8	Pck
	Port B data register	PBDR	H'FE40 0044	H'1E40 0044	8	Pck
	Port C data register	PCDR	H'FE40 0048	H'1E40 0048	8	Pck
	Port D data register	PDDR	H'FE40 004C	H'1E40 004C	8	Pck
	Port E data register	PEDR	H'FE40 0050	H'1E40 0050	8	Pck
	Port F data register	PFDR	H'FE40 0054	H'1E40 0054	8	Pck
	Port G data register	PGDR	H'FE40 0058	H'1E40 0058	8	Pck
	Port H data register	PHDR	H'FE40 005C	H'1E40 005C	8	Pck
	Port J data register	PJDR	H'FE40 0060	H'1E40 0060	8	Pck
	Port K data register	PKDR	H'FE40 0064	H'1E40 0064	8	Pck
	GPIO interrupt control register	GPIOIC	H'FF80 0048	H'1F80 0048	16	Bck
HAC Channel 0	Control and status register 0	HACCR0	H'FE24 0008	H'1E24 0008	32	Pck
	Command/status address register 0	HACCSAR0	H'FE24 0020	H'1E24 0020	32	Pck
	Command/status data register 0	HACCSDR0	H'FE24 0024	H'1E24 0024	32	Pck
	PCM left channel register 0	HACPCML0	H'FE24 0028	H'1E24 0028	32	Pck
	PCM right channel register 0	HACPCMR0	H'FE24 002C	H'1E24 002C	32	Pck
	TX interrupt enable register 0	HACTIER0	H'FE24 0050	H'1E24 0050	32	Pck
	TX status register 0	HACTSR0	H'FE24 0054	H'1E24 0054	32	Pck
	RX interrupt enable register 0	HACRIER0	H'FE24 0058	H'1E24 0058	32	Pck
	RX status register 0	HACRSR0	H'FE24 005C	H'1E24 005C	32	Pck
	HAC control register 0	HACACR0	H'FE24 0060	H'1E24 0060	32	Pck
HAC Channel 1	Control and status register 1	HACCR1	H'FE25 0008	H'1E25 0008	32	Pck
	Command/status address register 1	HACCSAR1	H'FE25 0020	H'1E25 0020	32	Pck
	Command/status data register 1	HACCSDR1	H'FE25 0024	H'1E25 0024	32	Pck
	PCM left channel register 1	HACPCML1	H'FE25 0028	H'1E25 0028	32	Pck
	PCM right channel register 1	HACPCMR1	H'FE25 002C	H'1E25 002C	32	Pck
	TX interrupt enable register 1	HACTIER1	H'FE25 0050	H'1E25 0050	32	Pck
	TX status register 1	HACTSR1	H'FE25 0054	H'1E25 0054	32	Pck
	RX interrupt enable register 1	HACRIER1	H'FE25 0058	H'1E25 0058	32	Pck
	RX status register 1	HACRSR1	H'FE25 005C	H'1E25 005C	32	Pck
	HAC control register 1	HACACR1	H'FE25 0060	H'1E25 0060	32	Pck

MMCIF	Command register 0	CMDR0	H'FE50 0000	H'1E50 0000	8	Pck
	Command register 1	CMDR1	H'FE50 0001	H'1E50 0001	8	Pck
	Command register 2	CMDR2	H'FE50 0002	H'1E50 0002	8	Pck
	Command register 3	CMDR3	H'FE50 0003	H'1E50 0003	8	Pck
	Command register 4	CMDR4	H'FE50 0004	H'1E50 0004	8	Pck
	Command register 5	CMDR5	H'FE50 0005	H'1E50 0005	8	Pck
	Command start register	CMDSTRT	H'FE50 0006	H'1E50 0006	8	Pck
	Operation control register	OPCR	H'FE50 000A	H'1E50 000A	8	Pck
	Card status register	CSTR	H'FE50 000B	H'1E50 000B	8	Pck
	Interrupt control register 0	INTCR0	H'FE50 000C	H'1E50 000C	8	Pck
	Interrupt control register 1	INTCR1	H'FE50 000D	H'1E50 000D	8	Pck
	Interrupt status register 0	INTSTR0	H'FE50 000E	H'1E50 000E	8	Pck
	Interrupt status register 1	INTSTR1	H'FE50 000F	H'1E50 000F	8	Pck
	Transfer clock control register	CLKON	H'FE50 0010	H'1E50 0010	8	Pck
	Command timeout control register	CTOCR	H'FE50 0011	H'1E50 0011	8	Pck
	Transfer byte number count register	TBCR	H'FE50 0014	H'1E50 0014	8	Pck
	Mode register	MODER	H'FE50 0016	H'1E50 0016	8	Pck
	Command type register	CMDTYR	H'FE50 0018	H'1E50 0018	8	Pck
	Response type register	RSPTYR	H'FE50 0019	H'1E50 0019	8	Pck
	Response register 0	RSPR0	H'FE50 0020	H'1E50 0020	8	Pck
	Response register 1	RSPR1	H'FE50 0021	H'1E50 0021	8	Pck
	Response register 2	RSPR2	H'FE50 0022	H'1E50 0022	8	Pck
	Response register 3	RSPR3	H'FE50 0023	H'1E50 0023	8	Pck
	Response register 4	RSPR4	H'FE50 0024	H'1E50 0024	8	Pck
	Response register 5	RSPR5	H'FE50 0025	H'1E50 0025	8	Pck
	Response register 6	RSPR6	H'FE50 0026	H'1E50 0026	8	Pck
	Response register 7	RSPR7	H'FE50 0027	H'1E50 0027	8	Pck
	Response register 8	RSPR8	H'FE50 0028	H'1E50 0028	8	Pck
	Response register 9	RSPR9	H'FE50 0029	H'1E50 0029	8	Pck
	Response register 10	RSPR10	H'FE50 002A	H'1E50 002A	8	Pck
	Response register 11	RSPR11	H'FE50 002B	H'1E50 002B	8	Pck
	Response register 12	RSPR12	H'FE50 002C	H'1E50 002C	8	Pck
	Response register 13	RSPR13	H'FE50 002D	H'1E50 002D	8	Pck

MMCIF	Response register 14	RSPR14	H'FE50 002E	H'1E50 002E	8	Pck
	Response register 15	RSPR15	H'FE50 002F	H'1E50 002F	8	Pck
	Response register 16	RSPR16	H'FE50 0030	H'1E50 0030	8	Pck
	Data timeout register	DTOUTR	H'FE50 0032	H'1E50 0032	16	Pck
	Data register	DR	H'FE50 0040	H'1E50 0040	16	Pck
	FIFO pointer clear register	FIFOCLR	H'FE50 0042	H'1E50 0042	8	Pck
	DMA control register	DMACR	H'FE50 0044	H'1E50 0044	8	Pck
	Interrupt control register 2	INTCR2	H'FE50 0046	H'1E50 0046	8	Pck
	Interrupt status register 2	INTSTR2	H'FE50 0048	H'1E50 0048	8	Pck
	Receive data timing select register	RDTIMSEL	H'FE50 004A	H'1E50 004A	8	Pck
MFI	MFI index register	MFIIDX	H'FE2C 0000	H'1E2C 0000	32	Pck
	MFI general status register	MFIGSR	H'FE2C 0004	H'1E2C 0004	32	Pck
	MFI status/control register	MFISCR	H'FE2C 0008	H'1E2C 0008	32	Pck
	MFI memory control register	MFIMCR	H'FE2C 000C	H'1E2C 000C	32	Pck
	MFI internal interrupt control register	MFIIICR	H'FE2C 0010	H'1E2C 0010	32	Pck
	MFI external interrupt control register	MFIEICR	H'FE2C 0014	H'1E2C 0014	32	Pck
	MFI address register	MFIADR	H'FE2C 0018	H'1E2C 0018	32	Pck
	MFI data register	MFIDATA	H'FE2C 001C	H'1E2C 001C	32	Pck
		MFRAM Start	H'FE2E 0000	H'1E2E 0000	32	Pck
		MFRAM End	H'FE2E 07FF	H'1E2E 07FF	32	Pck
H-UDI	Instruction register	SDIR	H'FFF0 0000	H'1FF0 0000	16	Pck
	Data register H	SDDR/SDDRH	H'FFF0 0008	H'1FF0 0008	32/16	Pck
	Data register L	SDDRL	H'FFF0 000A	H'1FF0 000A	16	Pck
	Interrupt source register	SDINT	H'FFF0 0014	H'1FF0 0014	16	Pck
	Bypass register	SDBPR	—	—	—	—
	Boundary scan register	SDBSR	—	—	—	—
ADC	A/D data register A	ADDRA	H'FE28 0000	H'1E28 0000	16	Pck
	A/D data register B	ADDRB	H'FE28 0002	H'1E28 0002	16	Pck
	A/D data register C	ADDRC	H'FE28 0004	H'1E28 0004	16	Pck
	A/D data register D	ADDRD	H'FE28 0006	H'1E28 0006	16	Pck
	A/D control/status register	ADCSR	H'FE28 0008	H'1E28 0008	16	Pck
LCDC	LCDC input clock register	LDICKR	H'FE30 0C00	H'1E30 0C00	16	Pck
	LCDC module type register	LDMTR	H'FE30 0C02	H'1E30 0C02	16	Pck

LCDC	LCDC data format register	LDDFR	H'FE30 0C04	H'1E30 0C04	16	Pck
	LCDC scan mode register	LDSMR	H'FE30 0C06	H'1E30 0C06	16	Pck
	LCDC data fetch start address register for upper display panel	LDSARU	H'FE30 0C08	H'1E30 0C08	32	Pck
	LCDC data fetch start address register for lower display panel	LDSARL	H'FE30 0C0C	H'1E30 0C0C	32	Pck
	LCDC fetch data line address offset register for display panel	LDLAOR	H'FE30 0C10	H'1E30 0C10	16	Pck
	LCDC palette control register	LDPALCR	H'FE30 0C12	H'1E30 0C12	16	Pck
	LCDC palette data register 00 to FF	LDPR00 to FF*3	H'FE30 0800	H'1E30 0800	32	Pck
	LCDC horizontal character number register	LDHCNR	H'FE30 0C14	H'1E30 0C14	16	Pck
	LCDC horizontal synchronization signal register	LDHSYNR	H'FE30 0C16	H'1E30 0C16	16	Pck
	LCDC vertical displayed line number register	LDVDLNR	H'FE30 0C18	H'1E30 0C18	16	Pck
	LCDC vertical total line number register	LDVTLNR	H'FE30 0C1A	H'1E30 0C1A	16	Pck
	LCDC vertical synchronization signal register	LDVSYNR	H'FE30 0C1C	H'1E30 0C1C	16	Pck
	LCDC AC modulation signal toggle line number register	LDACLNR	H'FE30 0C1E	H'1E30 0C1E	16	Pck
	LCDC interrupt control register	LDINTR	H'FE30 0C20	H'1E30 0C20	16	Pck
	LCDC power management mode register	LDPMMR	H'FE30 0C24	H'1E30 0C24	16	Pck
	LCDC power supply sequence period register	LDPSPR	H'FE30 0C26	H'1E30 0C26	16	Pck
	LCDC control register	LDCNTR	H'FE30 0C28	H'1E30 0C28	16	Pck
UBC	Break address register A	BARA	H'FF20 0000	H'1F20 0000	32	Ick
	Break ASID register A	BASRA	H'FF00 0014	H'1F00 0014	8	Ick
	Break address mask register A	BAMRA	H'FF20 0004	H'1F20 0004	8	Ick
	Break bus cycle register A	BBRA	H'FF20 0008	H'1F20 0008	16	Ick
	Break address register B	BARB	H'FF20 000C	H'1F20 000C	32	Ick
	Break ASID register B	BASRB	H'FF00 0018	H'1F00 0018	8	Ick
	Break address mask register B	BAMRB	H'FF20 0010	H'1F20 0010	8	Ick
	Break bus cycle register B	BBRB	H'FF20 0014	H'1F20 0014	16	Ick
	Break data register B	BDRB	H'FF20 0018	H'1F20 0018	32	Ick
	Break data mask register B	BDMRB	H'FF20 001C	H'1F20 001C	32	Ick
	Break control register	BRCR	H'FF20 0020	H'1F20 0020	16	Ick

Notes: 1. For details, refer to the description of SDMR2 and SDMR3.
2. Read: Byte access, Write: Word access
3. There are 256 LDPRxx: LDPR00, LDPR01, ..., LDPRFF. The addresses for LDPRxx are H'FE30 0800, H'FE30 0804, ..., H'FE30 0BFC.

Register addresses and bit names of the on-chip peripheral modules are described below.

Each line covers eight bits, and 16-bit and 32-bit registers are shown as 2 or 4 lines, respectively.

Abbrev.	Bit 31/ 23/15/7	Bit 30/ 22/14/6	Bit 29/ 21/13/5	Bit 28/ 20/12/4	Bit 27/ 19/11/3	Bit 26/ 18/10/2	Bit 25/ 17/9/ 1	Bit 24/ 16/8/0	Module
PTEH	VPN	VPN	VPN	VPN	VPN	VPN	VPN	VPN	MMU
	VPN	VPN	VPN	VPN	VPN	VPN	VPN	VPN	
	VPN	VPN	VPN	VPN	VPN	VPN	—	—	
	ASID	ASID	ASID	ASID	ASID	ASID	ASID	ASID	
PTEL	—	—	—	PPN	PPN	PPN	PPN	PPN	
	PPN	PPN	PPN	PPN	PPN	PPN	PPN	PPN	
	PPN	PPN	PPN	PPN	PPN	PPN	—	V	
	SZ1	PR1	PR0	SZ0	C	D	SH	WT	
TTB	TTB	TTB	TTB	TTB	TTB	TTB	TTB	TTB	
	TTB	TTB	TTB	TTB	TTB	TTB	TTB	TTB	
	TTB	TTB	TTB	TTB	TTB	TTB	TTB	TTB	
	TTB	TTB	TTB	TTB	TTB	TTB	TTB	TTB	
TEA	Virtual address at which MMU exception or address error occurred								
	Virtual address at which MMU exception or address error occurred								
	Virtual address at which MMU exception or address error occurred								
	Virtual address at which MMU exception or address error occurred								
MMUCR	LRUI	LRUI	LRUI	LRUI	LRUI	LRUI	—	—	
	URB	LRB	LRB	LRB	LRB	LRB	—	—	
	URC	URC	URC	URC	URC	URC	SQMD	SV	
	—	—	—	—	—	TI	—	AT	
PTEA	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	TC	SA2	SA1	SA0	

CCR	EMODE	—	—	—	—	—	—	—	Cache
	—	—	—	—	—	—	—	—	
	IIX	—	—	—	ICI	—	—	ICE	
	OIX	—	ORA	—	OCI	CB	WT	OCE	
QACR0	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	AREA0	AREA0	AREA0	—	—	
QACR1	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	AREA1	AREA1	AREA1	—	—	
TRA	—	—	—	—	—	—	—	—	Exception handling
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	imm	imm	
	imm	imm	imm	imm	imm	imm	—	—	
EXPEVT	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
INTEVT	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
ICR	NMIL	MAI	—	—	—	—	NMIB	NMIE	INTC
	IRLM	—	—	—	—	—	—	—	
IPRA	IPR15	IPR14	IPR13	IPR12	IPR11	IPR10	IPR9	IPR8	
	IPR7	IPR6	IPR5	IPR4	IPR3	IPR2	IPR1	IPR0	
IPRB	IPR15	IPR14	IPR13	IPR12	IPR11	IPR10	IPR9	IPR8	
	IPR7	IPR6	IPR5	IPR4	IPR3	IPR2	IPR1	IPR0	
IPRC	IPR15	IPR14	IPR13	IPR12	IPR11	IPR10	IPR9	IPR8	
	IPR7	IPR6	IPR5	IPR4	IPR3	IPR2	IPR1	IPR0	
IPRD	IPR15	IPR14	IPR13	IPR12	IPR11	IPR10	IPR9	IPR8	
	IPR7	IPR6	IPR5	IPR4	IPR3	IPR2	IPR1	IPR0	

INTPRI00	—	—	—	—	—	—	—	—	INTC
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
INTPRI04	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
INTPRI08	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
INTPRI0C	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
INTREQ00	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
INTREQ04	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
INTMSK00	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
INTMSK04	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	

INTMSKCLR00	—	—	—	—	—	—	—	—	INTC
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
INTMSKCLR04	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
BCR1	ENDIAN	—	A0MPX	—	—	DPUP	—	OPUP	BSC
	—	—	A1MBC	A4MBC	BREQEN	—	MEMMPX	DMABST	
	HIZMEM	HIZCNT	A0BST2	A0BST1	A0BST0	A5BST2	A5BST1	A5BST0	
	A6BST2	A6BST1	A6BST0	DRAMTP2	DRAMTP1	DRAMTP0	—	A56PCM	
BCR2	A0SZ1	A0SZ0	A6SZ1	A6SZ0	A5SZ1	A5SZ0	A4SZ1	A4SZ0	
	A3SZ1	A3SZ0	A2SZ1	A2SZ0	A1SZ1	A1SZ0	—	STBIRLEN	
BCR3	MEMMODE	A1MPX	A4MPX	—	—	—	—	—	
	—	—	—	—	—	—	—	SDBL	
BCR4	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	ASYNC6	ASYNC5	ASYNC4	ASYNC3	ASYNC2	ASYNC1	ASYNC0	
WCR1	—	DMAIW2	DMAIW1	DMAIW0	—	A6IW2	A6IW1	A6IW0	
	—	A5IW2	A5IW1	A5IW0	—	A4IW2	A4IW1	A4IW0	
	—	A3IW2	A3IW1	A3IW0	—	A2IW2	A2IW1	A2IW0	
	—	A1IW2	A1IW1	A1IW0	—	A0IW2	A0IW1	A0IW0	
WCR2	A6W2	A6W1	A6W0	A6B2	A6B1	A6B0	A5W2	A5W1	
	A5W0	A5B2	A5B1	A5B0	A4W2	A4W1	A4W0	—	
	A3W2	A3W1	A3W0	—	A2W2	A2W1	A2W0	A1W2	
	A1W1	A1W0	A0W2	A0W1	A0W0	A0B2	A0B1	A0B0	
WCR3	—	—	—	—	—	A6S0	A6H1	A6H0	
	—	A5S0	A5H1	A5H0	A4RDH	A4S0	A4H1	A4H0	
	—	A3S0	A3H1	A3H0	—	A2S0	A2H1	A2H0	
	A1RDH	A1S0	A1H1	A1H0	—	A0S0	A0H1	A0H0	

WCR4	—	—	—	—	—	—	—	—	BSC
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	CSH1	CSH0	
MCR	RASD	MRSET	TRC2	TRC1	TRC0	—	—	—	
	—	—	TPC2	TPC1	TPC0	—	RCD1	RCD0	
	TRWL2	TRWL1	TRWL0	TRAS2	TRAS1	TRAS0	—	SZ1	
	SZ0	AMXEXT	AMX2	AMX1	AMX0	RFSH	RMODE	—	
PCR	A5PCW1	A5PCW0	A6PCW1	A6PCW0	A5TED2	A5TED1	A5TED0	A6TED2	
	A6TED1	A6TED0	A5TEH2	A5TEH1	A5TEH0	A6TEH2	A6TEH1	A6TEH0	
RTCSR	—	—	—	—	—	—	—	—	
	CMF	CMIE	CKS2	CKS1	CKS0	OVF	OVIE	LMTS	
RTCNT	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
RTCOR	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
RFCR	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
SDMR2	—	—	—	—	—	—	—	—	
SDMR3	—	—	—	—	—	—	—	—	
SAR0	—	—	—	—	—	—	—	—	DMAC
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
DAR0	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
DMATCR0	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	

CHCR0	SSA2	SSA1	SSA0	STC	DSA2	DSA1	DSA0	DTC	DMAC
	—	—	—	—	DS	RL	AM	AL	
	DM1	DM0	SM1	SM0	RS3	RS2	RS1	RS0	
	TM	TS2	TS1	TS0	CHSET	IE	TE	DE	
SAR1	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
DAR1	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
DMATCR1	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
CHCR1	SSA2	SSA1	SSA0	STC	DSA2	DSA1	DSA0	DTC	
	—	—	—	—	DS	RL	AM	AL	
	DM1	DM0	SM1	SM0	RS3	RS2	RS1	RS0	
	TM	TS2	TS1	TS0	CHSET	IE	TE	DE	
SAR2	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
DAR2	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
DMATCR2	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	

CHCR2	SSA2	SSA1	SSA0	STC	DSA2	DSA1	DSA0	DTC	DMAC
	—	—	—	—	DS	RL	AM	AL	
	DM1	DM0	SM1	SM0	RS3	RS2	RS1	RS0	
	TM	TS2	TS1	TS0	CHSET	IE	TE	DE	
SAR3	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
DAR3	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
DMATCR3	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
CHCR3	SSA2	SSA1	SSA0	STC	DSA2	DSA1	DSA0	DTC	
	—	—	—	—	DS	RL	AM	AL	
	DM1	DM0	SM1	SM0	RS3	RS2	RS1	RS0	
	TM	TS2	TS1	TS0	CHSET	IE	TE	DE	
SAR4	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
DAR4	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
DMATCR4	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	

CHCR4	SSA2	SSA1	SSA0	STC	DSA2	DSA1	DSA0	DTC	DMAC
	—	—	—	—	DS	RL	AM	AL	
	DM1	DM0	SM1	SM0	RS3	RS2	RS1	RS0	
	TM	TS2	TS1	TS0	CHSET	IE	TE	DE	
SAR5	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
DAR5	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
DMATCR5	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
CHCR5	SSA2	SSA1	SSA0	STC	DSA2	DSA1	DSA0	DTC	
	—	—	—	—	DS	RL	AM	AL	
	DM1	DM0	SM1	SM0	RS3	RS2	RS1	RS0	
	TM	TS2	TS1	TS0	CHSET	IE	TE	DE	
SAR6	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
DAR6	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
DMATCR6	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	

CHCR6	SSA2	SSA1	SSA0	STC	DSA2	DSA1	DSA0	DTC	DMAC
	—	—	—	—	DS	RL	AM	AL	
	DM1	DM0	SM1	SM0	RS3	RS2	RS1	RS0	
	TM	TS2	TS1	TS0	CHSET	IE	TE	DE	
SAR7	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
DAR7	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
DMATCR7	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
CHCR7	SSA2	SSA1	SSA0	STC	DSA2	DSA1	DSA0	DTC	
	—	—	—	—	DS	RL	AM	AL	
	DM1	DM0	SM1	SM0	RS3	RS2	RS1	RS0	
	TM	TS2	TS1	TS0	CHSET	IE	TE	DE	
DMAOR	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	DMS1	DMS0	—	—	—	—	PR1	PR0	
	—	—	—	—	—	AE	NMIF	DME	
DMARSRA	CH0WEN	CH0RS6	CH0RS5	CH0RS4	CH0RS3	CH0RS2	CH0RS1	CH0RS0	
	CH1WEN	CH1RS6	CH1RS5	CH1RS4	CH1RS3	CH1RS2	CH1RS1	CH1RS0	
	CH2WEN	CH2RS6	CH2RS5	CH2RS4	CH2RS3	CH2RS2	CH2RS1	CH2RS0	
	CH3WEN	CH3RS6	CH3RS5	CH3RS4	CH3RS3	CH3RS2	CH3RS1	CH3RS0	
DMARSRB	CH4WEN	CH4RS6	CH4RS5	CH4RS4	CH4RS3	CH4RS2	CH4RS1	CH4RS0	
	CH5WEN	CH5RS6	CH5RS5	CH5RS4	CH5RS3	CH5RS2	CH5RS1	CH5RS0	
	CH6WEN	CH6RS6	CH6RS5	CH6RS4	CH6RS3	CH6RS2	CH6RS1	CH6RS0	
	CH7WEN	CH7RS6	CH7RS5	CH7RS4	CH7RS3	CH7RS2	CH7RS1	CH7RS0	

DMARCR	REX7	REX6	REX5	REX4	REX3	REX2	REX1	REX0	DMAC
	R/A3	R/A2	—	—	—	—	RPR1	RPR0	
	—	DS3	RL3	AL3	—	DS2	RL2	AL2	
	—	DS1	RL1	AL1	—	DS0	RL0	AL0	
DMABRGCR	A1RXHE	A1RXEE	A1TXHE	A1TXEE	A0RXHE	A0RXEE	A0TXHE	A0TXEE	
	A1RXHF	A1RXEF	A1TXHF	A1TXEF	A0RXHF	A0RXEF	A0TXHF	A0TXEF	
	—	—	—	—	—	—	UAE	UTE	
	—	—	—	—	—	—	UAF	UTF	
DMAATXSAR0	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
DMAARXDAR0	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
DMAATXTCR0	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
DMAARXTCR0	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
DMAACR0	—	—	—	—	—	—	RAM1	RAM0	
	—	—	—	—	—	RAR	RDS	RDE	
	—	—	—	—	—	—	TAM1	TAM0	
	—	—	—	—	—	TAR	TDS	TDE	
DMAATXTCNT0	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	

DMAARXTCNT0	—	—	—	—	—	—	—	—	DMAC
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
DMAATXSAR1	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
DMAARXDAR1	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
DMAATXTCR1	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
DMAARXTCR1	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
DMAACR1	—	—	—	—	—	—	RAM1	RAM0	
	—	—	—	—	—	RAR	RDS	RDE	
	—	—	—	—	—	—	TAM1	TAM0	
	—	—	—	—	—	TAR	TDS	TDE	
DMAATXTCNT1	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
DMAARXTCNT1	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	

DMAUSAR	—	—	—	—	—	—	—	—	DMAC
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
DMAUDAR	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
DMAURWSZ	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	RW	
	—	—	—	SZ12	SZ11	SZ10	SZ9	SZ8	
	SZ7	SZ6	SZ5	SZ4	SZ3	SZ2	SZ1	SZ0	
DMAUCR	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	CVRT1	CVRT0	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	START	—	
FRQCR	—	—	—	—	CKOEN	PLL1EN	PLL2EN	IFC2	CPG
	IFC1	IFC0	BFC2	BFC1	BFC0	PFC2	PFC1	PFC0	
STBCR	STBY	—	—	MSTP4	—	MSTP2	—	—	
DCKDR	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	DCKEN	—	—	—	PLL3EN	DCKOUT	DIV1	DIV0	
MCKCR	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	FLMCK3	FLMCK2	FLMCK1	FLMCK0	
WTCNT	—	—	—	—	—	—	—	—	WDT
WTCSR	TME	WT/$\overline{IT}$	RSTS	WOVF	IOVF	CKS2	CKS1	CKS0	
STBCR2	DSLP	STHZ	—	—	—	—	MSTP6	MSTP5	Power-down
CLKSTP00	CSTP31	CSTP30	CSTP29	CSTP28	CSTP27	CSTP26	CSTP25	CSTP24	
	CSTP23	CSTP22	—	CSTP20	CSTP19	—	CSTP17	CSTP16	
	CSTP15	CSTP14	CSTP13	CSTP12	CSTP11	CSTP10	CSTP9	CSTP8	
	—	—	—	—	—	—	—	CSTP0	

CLKSTPCLR00	CSTP31	CSTP30	CSTP29	CSTP28	CSTP27	CSTP26	CSTP25	CSTP24	Power-down
	CSTP23	CSTP22	—	CSTP20	CSTP19	—	CSTP17	CSTP16	
	CSTP15	CSTP14	CSTP13	CSTP12	CSTP11	CSTP10	CSTP9	CSTP8	
	—	—	—	—	—	—	—	CSTP0	
TSTR	—	—	—	—	—	STR2	STR1	STR0	TMU
TCOR0	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
TCNT0	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
TCR0	—	—	—	—	—	—	—	UNF	
	—	—	UNIE	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	
TCOR1	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
TCNT1	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
TCR1	—	—	—	—	—	—	—	UNF	
	—	—	UNIE	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	
TCOR2	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
TCNT2	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	

TCR2	—	—	—	—	—	—	ICPF	UNF	TMU
	ICPE1	ICPE0	UNIE	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	
TCPR2	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
CMTCFG	—	—	—	—	—	—	—	—	CMT
	—	—	—	—	—	—	ROT2	ROT0	
	ED3	ED3	ED2	ED2	ED1	ED1	ED0	ED0	
	—	FRCM	FRTM	T23	T23	T23	T01	T01	
CMTFRT	FRT	FRT	FRT	FRT	FRT	FRT	FRT	FRT	
	FRT	FRT	FRT	FRT	FRT	FRT	FRT	FRT	
	FRT	FRT	FRT	FRT	FRT	FRT	FRT	FRT	
	FRT	FRT	FRT	FRT	FRT	FRT	FRT	FRT	
CMTCTL	TE3	TE2	TE1	TE0	IOE3	IOE2	IOE1	IOE0	
	ICE3	ICE2	ICE1	ICE0	IEE3	IEE2	IEE1	IEE0	
	CC3	CC3	CC2	CC2	CC1	CC1	CC0	CC0	
	SI3	SI2	SI1	SI0	OP3	OP2	OP1	OP0	
CMTIRQS	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	IO3	IO2	IO1	IO0	
	IC3	IC2	IC1	IC0	IE3	IE2	IE1	IE0	
CMTCH0T	Channel 0 time 31-24								
	Channel 0 time 23-16								
	Channel 0 time 15-8								
	Channel 0 time 7-0								
CMTCH1T	Channel 1 time 31-24								
	Channel 1 time 23-16								
	Channel 1 time 15-8								
	Channel 1 time 7-0								

CMTCH2T			Channel 2 time 31-24						CMT
			Channel 2 time 23-16						
			Channel 2 time 15-8						
			Channel 2 time 7-0						
CMTCH3T			Channel 3 time 31-24						
			Channel 3 time 23-16						
			Channel 3 time 15-8						
			Channel 3 time 7-0						
CMTCH0ST			Channel 0 stop time 31-24						
			Channel 0 stop time 23-16						
			Channel 0 stop time 15-8						
			Channel 0 stop time 7-0						
CMTCH1ST			Channel 1 stop time 31-24						
			Channel 1 stop time 23-16						
			Channel 1 stop time 15-8						
			Channel 1 stop time 7-0						
CMTCH2ST			Channel 2 stop time 31-24						
			Channel 2 stop time 23-16						
			Channel 2 stop time 15-8						
			Channel 2 stop time 7-0						
CMTCH3ST			Channel 3 stop time 31-24						
			Channel 3 stop time 23-16						
			Channel 3 stop time 15-8						
			Channel 3 stop time 7-0						
CMTCH0C	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
			Channel 0 counter 15-8						
			Channel 0 counter 7-0						
CMTCH1C	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
			Channel 1 counter 15-8						
			Channel 1 counter 7-0						

CMTCH2C	—	—	—	—	—	—	—	—	CMT
	—	—	—	—	—	—	—	—	
	Channel 2 counter 15-8								
	Channel 2 counter 7-0								
CMTCH3C	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	Channel 3 counter 15-8								
	Channel 3 counter 7-0								
SCSMR0	—	—	—	—	—	—	—	—	SCIF(0)
	C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	—	CKS1	CKS0	
SCBRR0	—	—	—	—	—	—	—	—	
SCSCR0	—	—	—	—	—	—	—	—	
	TIE	RIE	TE	RE	REIE	—	CKE1	CKE0	
SCFTDR0	—	—	—	—	—	—	—	—	
SCFSR0	—	—	—	—	—	—	—	—	
	ER	TEND	TDFE	BRK	FER	PER	RDF	DR	
SCFRDR0	—	—	—	—	—	—	—	—	
SCFCR0	—	—	—	—	—	—	—	—	
	RTRG1	RTRG0	TTRG1	TTRG0	—	TFRST	RFRST	LOOP	
SCTFDR0	—	—	—	—	—	—	—	—	
	T7	T6	T5	T4	T3	T2	T1	T0	
SCRFDR0	—	—	—	—	—	—	—	—	
	R7	R6	R5	R4	R3	R2	R1	R0	
SCSPTR0	—	—	—	—	—	—	—	—	
	—	—	—	—	SCKIO	SCKDT	SPB2IO	SPB2DT	
SCLSR0	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	ORER	
SCRER0	—	PER6	PER5	PER4	PER3	PER2	PER1	PER0	
	—	FER6	FER5	FER4	FER3	FER2	FER1	FER0	
SCSMR1	—	—	—	—	—	—	—	—	SCIF(1)
	C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	—	CKS1	CKS0	
SCBRR1	—	—	—	—	—	—	—	—	

SCSCR1	—	—	—	—	—	—	—	—	SCIF(1)
	TIE	RIE	TE	RE	REIE	—	CKE1	CKE0	
SCFTDR1	—	—	—	—	—	—	—	—	
SCFSR1	—	—	—	—	—	—	—	—	
	ER	TEND	TDFE	BRK	FER	PER	RDF	DR	
SCFRDR1	—	—	—	—	—	—	—	—	
SCFCR1	—	—	—	—	—	RSTRG2	RSTRG1	RSTRG0	
	RTRG1	RTRG0	TTRG1	TTRG0	MCE	TFRST	RFRST	LOOP	
SCTFDR1	—	—	—	—	—	—	—	—	
	T7	T6	T5	T4	T3	T2	T1	T0	
SCRFDR1	—	—	—	—	—	—	—	—	
	R7	R6	R5	R4	R3	R2	R1	R0	
SCSPTR1	—	—	—	—	—	—	—	—	
	RTSIO	RTSDT	CTSIO	CTSDT	SCKIO	SCKDT	SPB2IO	SPB2DT	
SCLSR1	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	ORER	
SCRER1	—	PER6	PER5	PER4	PER3	PER2	PER1	PER0	
	—	FER6	FER5	FER4	FER3	FER2	FER1	FER0	
SCSMR2	—	—	—	—	—	—	—	—	SCIF(2)
	C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	—	CKS1	CKS0	
SCBRR2	—	—	—	—	—	—	—	—	
SCSCR2	—	—	—	—	—	—	—	—	
	TIE	RIE	TE	RE	REIE	—	CKE1	CKE0	
SCFTDR2	—	—	—	—	—	—	—	—	
SCFSR2	—	—	—	—	—	—	—	—	
	ER	TEND	TDFE	BRK	FER	PER	RDF	DR	
SCFRDR2	—	—	—	—	—	—	—	—	
SCFCR2	—	—	—	—	—	RSTRG2	RSTRG1	RSTRG0	
	RTRG1	RTRG0	TTRG1	TTRG0	MCE	TFRST	RFRST	LOOP	
SCTFDR2	—	—	—	—	—	—	—	—	
	T7	T6	T5	T4	T3	T2	T1	T0	
SCRFDR2	—	—	—	—	—	—	—	—	
	R7	R6	R5	R4	R3	R2	R1	R0	

SCSPTR2	—	—	—	—	—	—	—	—	SCIF(2)
	RTSIO	RTSDT	CTSIO	CTSDT	SCKIO	SCKDT	SPB2IO	SPB2DT	
SCLSR2	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	ORER	
SCRER2	—	PER6	PER5	PER4	PER3	PER2	PER1	PER0	
	—	FER6	FER5	FER4	FER3	FER2	FER1	FER0	
SISMR	—	—	PE	O/$\overline{E}$	—	—	—	—	SIM
SIBRR	—	—	—	—	—	BRR2	BRR1	BRR0	
SISCR	TIE	RIE	TE	RE	WAIT_IE	TEIE	CKE1	CKE0	
SITDR	SITD7	SITD6	SITD5	SITD4	SITD3	SITD2	SITD1	SITD0	
SISSR	TDRE	RDRF	ORER	ERS	PER	TEND	WAIT_ER	—	
SIRDR	SIRD7	SIRD6	SIRD5	SIRD4	SIRD3	SIRD2	SIRD1	SIRD0	
SISCMR	—	LCB	PB	—	SDIR	SINV	RST	SMIF	
SISC2R	EIO	—	—	—	—	—	—	—	
SIWAIT	SIWAIT15	SIWAIT14	SIWAIT13	SIWAIT12	SIWAIT11	SIWAIT10	SIWAIT9	SIWAIT8	
	SIWAIT7	SIWAIT6	SIWAIT5	SIWAIT4	SIWAIT3	SIWAIT2	SIWAIT1	SIWAIT0	
SIGRD	SIGRD7	SIGRD6	SIGRD5	SIGRD4	SIGRD3	SIGRD2	SIGRD1	SIGRD0	
SISMPL	—	—	—	—	—	SISMPL10	SISMPL9	SISMPL8	
	SISMPL7	SISMPL6	SISMPL5	SISMPL4	SISMPL3	SISMPL2	SISMPL1	SISMPL0	
ICSCR0	—	—	—	—	—	—	—	—	I^2C(0)
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	SDBS	SIE	GCAE	FNA	
ICMCR0	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	MDBS	FSCL	FSDA	OBPC	MIE	TSBE	FSB	ESG	
ICSSR0	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	GCAR	STM	SSR	SDE	SDT	SDR	SAR	

ICMSR0	—	—	—	—	—	—	—	—	I^2C(0)
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	MNR	MAL	MST	MDE	MDT	MDR	MAT	
ICSIER0	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	SSRE	SDEE	SDTE	SDRE	SARE	
ICMIER0	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	MNRE	MALE	MSTE	MDEE	MDTE	MDRE	MATE	
ICCCR0	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	SCGD	SCGD	SCGD	SCGD	SCGD	SCGD	CDF	CDF	
ICSAR0	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	SADD0_6	SADD0_5	SADD0_4	SADD0_3	SADD0_2	SADD0_1	SADD0_0	
ICMAR0	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	SADD1_6	SADD1_5	SADD1_4	SADD1_3	SADD1_2	SADD1_1	SADD1_0	STM1	
ICRXD0	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	RXD	RXD	RXD	RXD	RXD	RXD	RXD	RXD	
ICTXD0	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	TXD	TXD	TXD	TXD	TXD	TXD	TXD	TXD	

ICFCR0	—	—	—	—	—	—	—	—	$I^2C(0)$
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	RTRG3	RTRG2	RTRG1	RTRG0	TTRG1	TTRG0	RFRST	TFRST	
ICFSR0	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	TEND	RDF	TDFE	
ICFIER0	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	TEIE	RXIE	TXIE	
ICRFDR0	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	R4	R3	R2	R1	R0	
ICTFDR0	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	T4	T3	T2	T1	T0	
ICSCR1	—	—	—	—	—	—	—	—	$I^2C(1)$
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	SDBS	SIE	GCAE	FNA	
ICMCR1	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	MDBS	FSCL	FSDA	OBPC	MIE	TSBE	FSB	ESG	
ICSSR1	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	GCAR	STM	SSR	SDE	SDT	SDR	SAR	

ICMSR1	—	—	—	—	—	—	—	—	I²C(1)
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	MNR	MAL	MST	MDE	MDT	MDR	MAT	
ICSIER1	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	SSRE	SDEE	SDTE	SDRE	SARE	
ICMIER1	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	MNRE	MALE	MSTE	MDEE	MDTE	MDRE	MATE	
ICCCR1	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	SCGD	SCGD	SCGD	SCGD	SCGD	SCGD	CDF	CDF	
ICSAR1	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	SADD0_6	SADD0_5	SADD0_4	SADD0_3	SADD0_2	SADD0_1	SADD0_0	
ICMAR1	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	SADD1_6	SADD1_5	SADD1_4	SADD1_3	SADD1_2	SADD1_1	SADD1_0	STM1	
ICRXD1	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	RXD	RXD	RXD	RXD	RXD	RXD	RXD	RXD	
ICTXD1	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	TXD	TXD	TXD	TXD	TXD	TXD	TXD	TXD	

ICFCR1	—	—	—	—	—	—	—	—	I²C(1)
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	RTRG3	RTRG2	RTRG1	RTRG0	TTRG1	TTRG0	RFRST	TFRST	
ICFSR1	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	TEND	RDF	TDFE	
ICFIER1	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	TEIE	RXIE	TXIE	
ICRFDR1	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	R4	R3	R2	R1	R0	
ICTFDR1	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	T4	T3	T2	T1	T0	
SSICR0	—	—	—	DMEN	UIEN	OIEN	IIEN	DIEN	SSI(0)
	CHNL1	CHNL0	DWL2	DWL1	DWL0	SWL2	SWL1	SWL0	
	SCKD	SWSD	SCKP	SWSP	SPDP	SDTA	PDTA	DEL	
	BREN	CKDV	CKDV	CKDV	MUEN	CPEN	TRMD	EN	
SSISR0	—	—	—	DMRQ	UIRQ	OIRQ	IIRQ	DIRQ	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	CHNO1	CHNO0	SWNO	IDST	
SSITDR0	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	

SSIRDR0	—	—	—	—	—	—	—	—	SSI(0)
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
SSICR1	—	—	—	DMEN	UIEN	OIEN	IIEN	DIEN	SSI(1)
	CHNL1	CHNL0	DWL2	DWL1	DWL0	SWL2	SWL1	SWL0	
	SCKD	SWSD	SCKP	SWSP	SPDP	SDTA	PDTA	DEL	
	BREN	CKDV	CKDV	CKDV	MUEN	CPEN	TRMD	EN	
SSISR1	—	—	—	DMRQ	UIRQ	OIRQ	IIRQ	DIRQ	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	CHNO1	CHNO0	SWNO	IDST	
SSITDR1	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
SSIRDR1	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
HcRevision	—	—	—	—	—	—	—	—	USB
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	REV	REV	REV	REV	REV	REV	REV	REV	
HcControl	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	RWE	RWC	IR	
	HCFS1	HCFS0	BLE	CLE	IE	PLE	CBSR1	CBSR0	
HcCommand status	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	SOC1	SOC0	
	—	—	—	—	—	—	—	—	
	—	—	—	—	OCR	BLF	CLF	HCR	

HcInterrupt Status	—	OC	—	—	—	—	—	—	USB
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	RHSC	FNO	UE	RD	SF	WDH	SO	
HcInterrupt Enable	MIE	OC	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	RHSC	FNO	UE	RD	SF	WDH	SO	
HcInterrupt Disable	MIE	OC	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	RHSC	FNO	UE	RD	SF	WDH	SO	
HcHCCA	HCCA	HCCA	HCCA	HCCA	HCCA	HCCA	HCCA	HCCA	
	HCCA	HCCA	HCCA	HCCA	HCCA	HCCA	HCCA	HCCA	
	HCCA	HCCA	HCCA	HCCA	HCCA	HCCA	HCCA	HCCA	
	—	—	—	—	—	—	—	—	
HcPeriod CurrentED	PCED	PCED	PCED	PCED	PCED	PCED	PCED	PCED	
	PCED	PCED	PCED	PCED	PCED	PCED	PCED	PCED	
	PCED	PCED	PCED	PCED	PCED	PCED	PCED	PCED	
	PCED	PCED	PCED	PCED	—	—	—	—	
HcControl HeadED	CHED	CHED	CHED	CHED	CHED	CHED	CHED	CHED	
	CHED	CHED	CHED	CHED	CHED	CHED	CHED	CHED	
	CHED	CHED	CHED	CHED	CHED	CHED	CHED	CHED	
	CHED	CHED	CHED	CHED	—	—	—	—	
HcControl CurrentED	CCED	CCED	CCED	CCED	CCED	CCED	CCED	CCED	
	CCED	CCED	CCED	CCED	CCED	CCED	CCED	CCED	
	CCED	CCED	CCED	CCED	CCED	CCED	CCED	CCED	
	CCED	CCED	CCED	CCED	—	—	—	—	
HcBulk HeadED	BHED	BHED	BHED	BHED	BHED	BHED	BHED	BHED	
	BHED	BHED	BHED	BHED	BHED	BHED	BHED	BHED	
	BHED	BHED	BHED	BHED	BHED	BHED	BHED	BHED	
	BHED	BHED	BHED	BHED	—	—	—	—	

HcBulk CurrentED	BCED	BCED	BCED	BCED	BCED	BCED	BCED	BCED	USB
	BCED	BCED	BCED	BCED	BCED	BCED	BCED	BCED	
	BCED	BCED	BCED	BCED	BCED	BCED	BCED	BCED	
	BCED	BCED	BCED	BCED	—	—	—	—	
HcDone Head	DH	DH	DH	DH	DH	DH	DH	DH	
	DH	DH	DH	DH	DH	DH	DH	DH	
	DH	DH	DH	DH	DH	DH	DH	DH	
	DH	DH	DH	DH	—	—	—	—	
HcFm Interval	FIT	FSMPS	FSMPS	FSMPS	FSMPS	FSMPS	FSMPS	FSMPS	
	FSMPS	FSMPS	FSMPS	FSMPS	FSMPS	FSMPS	FSMPS	FSMPS	
	—	—	FI	FI	FI	FI	FI	FI	
	FI	FI	FI	FI	FI	FI	FI	FI	
HcFm Remaining	FRT	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	FR	FR	FR	FR	FR	FR	
	FR	FR	FR	FR	FR	FR	FR	FR	
HcFmNumber	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	FN	FN	FN	FN	FN	FN	FN	FN	
	FN	FN	FN	FN	FN	FN	FN	FN	
HcPeriodic Start	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	PS	PS	PS	PS	PS	PS	
	PS	PS	PS	PS	PS	PS	PS	PS	
HcLS Threshold	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	LST	LST	LST	LST	
	LST	LST	LST	LST	LST	LST	LST	LST	
HcRh DescriptorA	POTPGT	POTPGT	POTPGT	POTPGT	POTPGT	POTPGT	POTPGT	POTPGT	
	—	—	—	—	—	—	—	—	
	—	—	—	NOCP	OCPM	DT	NPS	PSM	
	NDP	NDP	NDP	NDP	NDP	NDP	NDP	NDP	

HcRh DescriptorB	PPCM	PPCM	PPCM	PPCM	PPCM	PPCM	PPCM	PPCM	USB
	PPCM	PPCM	PPCM	PPCM	PPCM	PPCM	PPCM	PPCM	
	DR	DR	DR	DR	DR	DR	DR	DR	
	DR	DR	DR	DR	DR	DR	DR	DR	
HcRhStatus	CRWE	—	—	—	—	—	—	—	
	—	—	—	—	—	—	OCIC	LPSC	
	DRWE	—	—	—	—	—	—	—	
	—	—	—	—	—	—	OCI	LPS	
HcRhPort Status1	—	—	—	—	—	—	—	—	
	—	—	—	PRSC	OCIC	PSSC	PESC	CSC	
	—	—	—	—	—	—	LSDA	PPS	
	—	—	—	PRS	POCI	PSS	PES	CCS	
Shared Memory Start	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
Shared Memory End	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
CAN0MCR	TST7	TST6	TST5	TST4	TST3	TST2	TST1	TST0	HCAN2(0)
	MCR7	—	MCR5	—	—	MCR2	MCR1	MCR0	
CAN0GSR	—	—	—	—	—	—	—	—	
	—	—	GSR5	GSR4	GSR3	GSR2	GSR1	GSR0	
CAN0BCR1	TSEG1_3	TSEG1_2	TSEG1_1	TSEG1_0	—	TSEG2_2	TSEG2_1	TSEG2_0	
	—	—	SJW1	SJW0	—	—	EG	BSP	
CAN0BCR0	—	—	—	—	—	—	—	—	
	BRP7	BRP6	BRP5	BRP4	BRP3	BRP2	BRP1	BRP0	
CAN0IRR	—	IRR14	IRR13	IRR12	—	—	IRR9	IRR8	
	IRR7	IRR6	IRR5	IRR4	IRR3	IRR2	IRR1	IRR0	
CAN0IMR	—	IMR14	IMR13	IMR12	—	—	IMR9	IMR8	
	IMR7	IMR6	IMR5	IMR4	IMR3	IMR2	IMR1	IMR0	

CAN0TECREC	TEC7	TEC6	TEC5	TEC4	TEC3	TEC2	TEC1	TEC0	HCAN2(0)
	REC7	REC6	REC5	REC4	REC3	REC2	REC1	REC0	
CAN0TXPR1	TXPR1_15	TXPR1_14	TXPR1_13	TXPR1_12	TXPR1_11	TXPR1_10	TXPR1_9	TXPR1_8	
	TXPR1_7	TXPR1_6	TXPR1_5	TXPR1_4	TXPR1_3	TXPR1_2	TXPR1_1	TXPR1_0	
CAN0TXPR0	TXPR0_15	TXPR0_14	TXPR0_13	TXPR0_12	TXPR0_11	TXPR0_10	TXPR0_9	TXPR0_8	
	TXPR0_7	TXPR0_6	TXPR0_5	TXPR0_4	TXPR0_3	TXPR0_2	TXPR0_1	—	
CAN0TXCR1	TXCR1_15	TXCR1_14	TXCR1_13	TXCR1_12	TXCR1_11	TXCR1_10	TXCR1_9	TXCR1_8	
	TXCR1_7	TXCR1_6	TXCR1_5	TXCR1_4	TXCR1_3	TXCR1_2	TXCR1_1	TXCR1_0	
CAN0TXCR0	TXCR0_15	TXCR0_14	TXCR0_13	TXCR0_12	TXCR0_11	TXCR0_10	TXCR0_9	TXCR0_8	
	TXCR0_7	TXCR0_6	TXCR0_5	TXCR0_4	TXCR0_3	TXCR0_2	TXCR0_1	—	
CAN0TXACK1	TXACK1_15	TXACK1_14	TXACK1_13	TXACK1_12	TXACK1_11	TXACK1_10	TXACK1_9	TXACK1_8	
	TXACK1_7	TXACK1_6	TXACK1_5	TXACK1_4	TXACK1_3	TXACK1_2	TXACK1_1	TXACK1_0	
CAN0TXACK0	TXACK0_15	TXACK0_14	TXACK0_13	TXACK0_12	TXACK0_11	TXACK0_10	TXACK0_9	TXACK0_8	
	TXACK0_7	TXACK0_6	TXACK0_5	TXACK0_4	TXACK0_3	TXACK0_2	TXACK0_1	—	
CAN0ABACK1	ABACK1_15	ABACK1_14	ABACK1_13	ABACK1_12	ABACK1_11	ABACK1_10	ABACK1_9	ABACK1_8	
	ABACK1_7	ABACK1_6	ABACK1_5	ABACK1_4	ABACK1_3	ABACK1_2	ABACK1_1	ABACK1_0	
CAN0ABACK0	ABACK0_15	ABACK0_14	ABACK0_13	ABACK0_12	ABACK0_11	ABACK0_10	ABACK0_9	ABACK0_8	
	ABACK0_7	ABACK0_6	ABACK0_5	ABACK0_4	ABACK0_3	ABACK0_2	ABACK0_1	—	
CAN0RXPR1	RXPR1_15	RXPR1_14	RXPR1_13	RXPR1_12	RXPR1_11	RXPR1_10	RXPR1_9	RXPR1_8	
	RXPR1_7	RXPR1_6	RXPR1_5	RXPR1_4	RXPR1_3	RXPR1_2	RXPR1_1	RXPR1_0	
CAN0RXPR0	RXPR0_15	RXPR0_14	RXPR0_13	RXPR0_12	RXPR0_11	RXPR0_10	RXPR0_9	RXPR0_8	
	RXPR0_7	RXPR0_6	RXPR0_5	RXPR0_4	RXPR0_3	RXPR0_2	RXPR0_1	RXPR0_0	
CAN0RFPR1	RFPR1_15	RFPR1_14	RFPR1_13	RFPR1_12	RFPR1_11	RFPR1_10	RFPR1_9	RFPR1_8	
	RFPR1_7	RFPR1_6	RFPR1_5	RFPR1_4	RFPR1_3	RFPR1_2	RFPR1_1	RFPR1_0	
CAN0RFPR0	RFPR0_15	RFPR0_14	RFPR0_13	RFPR0_12	RFPR0_11	RFPR0_10	RFPR0_9	RFPR0_8	
	RFPR0_7	RFPR0_6	RFPR0_5	RFPR0_4	RFPR0_3	RFPR0_2	RFPR0_1	RFPR0_0	
CAN0MBIMR1	MBIMR1_15	MBIMR1_14	MBIMR1_13	MBIMR1_12	MBIMR1_11	MBIMR1_10	MBIMR1_9	MBIMR1_8	
	MBIMR1_7	MBIMR1_6	MBIMR1_5	MBIMR1_4	MBIMR1_3	MBIMR1_2	MBIMR1_1	MBIMR1_0	
CAN0MBIMR0	MBIMR0_15	MBIMR0_14	MBIMR0_13	MBIMR0_12	MBIMR0_11	MBIMR0_10	MBIMR0_9	MBIMR0_8	
	MBIMR0_7	MBIMR0_6	MBIMR0_5	MBIMR0_4	MBIMR0_3	MBIMR0_2	MBIMR0_1	MBIMR0_0	
CAN0UMSR1	UMSR1_15	UMSR1_14	UMSR1_13	UMSR1_12	UMSR1_11	UMSR1_10	UMSR1_9	UMSR1_8	
	UMSR1_7	UMSR1_6	UMSR1_5	UMSR1_4	UMSR1_3	UMSR1_2	UMSR1_1	UMSR1_0	

CAN0UMSR0	UMSR0_15	UMSR0_14	UMSR0_13	UMSR0_12	UMSR0_11	UMSR0_10	UMSR0_9	UMSR0_8 HCAN2(0)
	UMSR0_7	UMSR0_6	UMSR0_5	UMSR0_4	UMSR0_3	UMSR0_2	UMSR0_1	UMSR0_0
CAN0TCNTR	TCNTR15	TCNTR14	TCNTR1 3	TCNTR12	TCNTR11	TCNTR10	TCNTR9	TCNTR8
	TCNTR7	TCNTR6	TCNTR5	TCNTR4	TCNTR3	TCNTR2	TCNTR1	TCNTR0
CAN0TCR	TCR15	—	TCR13	TCR12	TCR11	—	—	—
	—	—	TCR5	TCR4	TCR3	TCR2	TCR1	TCR0
CAN0TCMR0	TCMR15	TCMR14	TCMR13	TCMR12	TCMR11	TCMR10	TCMR9	TCMR8
	TCMR7	TCMR6	TCMR5	TCMR4	TCMR3	TCMR2	TCMR1	TCMR0
CAN0MB0	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
CAN0MB1	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
CAN0MB2	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
CAN0MB3	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
CAN0MB4	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
CAN0MB5	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
CAN0MB6	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
CAN0MB7	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
CAN0MB8	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
CAN0MB9	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
CAN0MB10	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
CAN0MB11	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—

CAN0MB12	—	—	—	—	—	—	—	—	HCAN2(0)
	—	—	—	—	—	—	—	—	
CAN0MB13	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
CAN0MB14	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
CAN0MB15	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
CAN0MB16	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
CAN0MB17	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
CAN0MB18	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
CAN0MB19	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
CAN0MB20	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
CAN0MB21	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
CAN0MB22	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
CAN0MB23	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
CAN0MB24	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
CAN0MB25	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
CAN0MB26	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
CAN0MB27	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	

CAN0MB28	—	—	—	—	—	—	—	—	HCAN2(0)
	—	—	—	—	—	—	—	—	
CAN0MB29	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
CAN0MB30	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
CAN0MB31	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
CAN1MCR	TST7	TST6	TST5	TST4	TST3	TST2	TST1	TST0	HCAN2(1)
	MCR7	—	MCR5	—	—	MCR2	MCR1	MCR0	
CAN1GSR	—	—	—	—	—	—	—	—	
	—	—	GSR5	GSR4	GSR3	GSR2	GSR1	GSR0	
CAN1BCR1	TSEG1_3	TSEG1_2	TSEG1_1	TSEG1_0	—	TSEG2_2	TSEG2_1	TSEG2_0	
	—	—	SJW1	SJW0	—	—	EG	BSP	
CAN1BCR0	—	—	—	—	—	—	—	—	
	BRP7	BRP6	BRP5	BRP4	BRP3	BRP2	BRP1	BRP0	
CAN1IRR	—	IRR14	IRR13	IRR12	—	—	IRR9	IRR8	
	IRR7	IRR6	IRR5	IRR4	IRR3	IRR2	IRR1	IRR0	
CAN1IMR	—	IMR14	IMR13	IMR12	—	—	IMR9	IMR8	
	IMR7	IMR6	IMR5	IMR4	IMR3	IMR2	IMR1	IMR0	
CAN1TECREC	TEC7	TEC6	TEC5	TEC4	TEC3	TEC2	TEC1	TEC0	
	REC7	REC6	REC5	REC4	REC3	REC2	REC1	REC0	
CAN1TXPR1	TXPR1_15	TXPR1_14	TXPR1_13	TXPR1_12	TXPR1_11	TXPR1_10	TXPR1_9	TXPR1_8	
	TXPR1_7	TXPR1_6	TXPR1_5	TXPR1_4	TXPR1_3	TXPR1_2	TXPR1_1	TXPR1_0	
CAN1TXPR0	TXPR0_15	TXPR0_14	TXPR0_13	TXPR0_12	TXPR0_11	TXPR0_10	TXPR0_9	TXPR0_8	
	TXPR0_7	TXPR0_6	TXPR0_5	TXPR0_4	TXPR0_3	TXPR0_2	TXPR0_1	—	
CAN1TXCR1	TXCR1_15	TXCR1_14	TXCR1_13	TXCR1_12	TXCR1_11	TXCR1_10	TXCR1_9	TXCR1_8	
	TXCR1_7	TXCR1_6	TXCR1_5	TXCR1_4	TXCR1_3	TXCR1_2	TXCR1_1	TXCR1_0	
CAN1TXCR0	TXCR0_15	TXCR0_14	TXCR0_13	TXCR0_12	TXCR0_11	TXCR0_10	TXCR0_9	TXCR0_8	
	TXCR0_7	TXCR0_6	TXCR0_5	TXCR0_4	TXCR0_3	TXCR0_2	TXCR0_1	—	

CAN1TXACK1	TXACK1_15	TXACK1_14	TXACK1_13	TXACK1_12	TXACK1_11	TXACK1_10	TXACK1_9	TXACK1_8 HCAN2(1)
	TXACK1_7	TXACK1_6	TXACK1_5	TXACK1_4	TXACK1_3	TXACK1_2	TXACK1_1	TXACK1_0
CAN1TXACK0	TXACK0_15	TXACK0_14	TXACK0_13	TXACK0_12	TXACK0_11	TXACK0_10	TXACK0_9	TXACK0_8
	TXACK0_7	TXACK0_6	TXACK0_5	TXACK0_4	TXACK0_3	TXACK0_2	TXACK0_1	—
CAN1ABACK1	ABACK1_15	ABACK1_14	ABACK1_13	ABACK1_12	ABACK1_11	ABACK1_10	ABACK1_9	ABACK1_8
	ABACK1_7	ABACK1_6	ABACK1_5	ABACK1_4	ABACK1_3	ABACK1_2	ABACK1_1	ABACK1_0
CAN1ABACK0	ABACK0_15	ABACK0_14	ABACK0_13	ABACK0_12	ABACK0_11	ABACK0_10	ABACK0_9	ABACK0_8
	ABACK0_7	ABACK0_6	ABACK0_5	ABACK0_4	ABACK0_3	ABACK0_2	ABACK0_1	—
CAN1RXPR1	RXPR1_15	RXPR1_14	RXPR1_13	RXPR1_12	RXPR1_11	RXPR1_10	RXPR1_9	RXPR1_8
	RXPR1_7	RXPR1_6	RXPR1_5	RXPR1_4	RXPR1_3	RXPR1_2	RXPR1_1	RXPR1_0
CAN1RXPR0	RXPR0_15	RXPR0_14	RXPR0_13	RXPR0_12	RXPR0_11	RXPR0_10	RXPR0_9	RXPR0_8
	RXPR0_7	RXPR0_6	RXPR0_5	RXPR0_4	RXPR0_3	RXPR0_2	RXPR0_1	RXPR0_0
CAN1RFPR1	RFPR1_15	RFPR1_14	RFPR1_13	RFPR1_12	RFPR1_11	RFPR1_10	RFPR1_9	RFPR1_8
	RFPR1_7	RFPR1_6	RFPR1_5	RFPR1_4	RFPR1_3	RFPR1_2	RFPR1_1	RFPR1_0
CAN1RFPR0	RFPR0_15	RFPR0_14	RFPR0_13	RFPR0_12	RFPR0_11	RFPR0_10	RFPR0_9	RFPR0_8
	RFPR0_7	RFPR0_6	RFPR0_5	RFPR0_4	RFPR0_3	RFPR0_2	RFPR0_1	RFPR0_0
CAN1MBIMR1	MBIMR1_15	MBIMR1_14	MBIMR1_13	MBIMR1_12	MBIMR1_11	MBIMR1_10	MBIMR1_9	MBIMR1_8
	MBIMR1_7	MBIMR1_6	MBIMR1_5	MBIMR1_4	MBIMR1_3	MBIMR1_2	MBIMR1_1	MBIMR1_0
CAN1MBIMR0	MBIMR0_15	MBIMR0_14	MBIMR0_13	MBIMR0_12	MBIMR0_11	MBIMR0_10	MBIMR0_9	MBIMR0_8
	MBIMR0_7	MBIMR0_6	MBIMR0_5	MBIMR0_4	MBIMR0_3	MBIMR0_2	MBIMR0_1	MBIMR0_0
CAN1UMSR1	UMSR1_15	UMSR1_14	UMSR1_13	UMSR1_12	UMSR1_11	UMSR1_10	UMSR1_9	UMSR1_8
	UMSR1_7	UMSR1_6	UMSR1_5	UMSR1_4	UMSR1_3	UMSR1_2	UMSR1_1	UMSR1_0
CAN1UMSR0	UMSR0_15	UMSR0_14	UMSR0_13	UMSR0_12	UMSR0_11	UMSR0_10	UMSR0_9	UMSR0_8
	UMSR0_7	UMSR0_6	UMSR0_5	UMSR0_4	UMSR0_3	UMSR0_2	UMSR0_1	UMSR0_0
CAN1TCNTR	TCNTR15	TCNTR14	TCNTR13	TCNTR12	TCNTR11	TCNTR10	TCNTR9	TCNTR8
	TCNTR7	TCNTR6	TCNTR5	TCNTR4	TCNTR3	TCNTR2	TCNTR1	TCNTR0
CAN1TCR	TCR15	—	TCR13	TCR12	TCR11	—	—	—
	—	—	TCR5	TCR4	TCR3	TCR2	TCR1	TCR0
CAN1TCMR	TCMR15	TCMR14	TCMR13	TCMR12	TCMR11	TCMR10	TCMR9	TCMR8
	TCMR7	TCMR6	TCMR5	TCMR4	TCMR3	TCMR2	TCMR1	TCMR0
CAN1MB0	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—

CAN1MB1	—	—	—	—	—	—	—	—	HCAN2(1)
	—	—	—	—	—	—	—	—	
CAN1MB2	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
CAN1MB3	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
CAN1MB4	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
CAN1MB5	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
CAN1MB6	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
CAN1MB7	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
CAN1MB8	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
CAN1MB9	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
CAN1MB10	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
CAN1MB11	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
CAN1MB12	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
CAN1MB13	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
CAN1MB14	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
CAN1MB15	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
CAN1MB16	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	

CAN1MB17	—	—	—	—	—	—	—	—	HCAN2(1)
	—	—	—	—	—	—	—	—	
CAN1MB18	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
CAN1MB19	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
CAN1MB20	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
CAN1MB21	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
CAN1MB22	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
CAN1MB23	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
CAN1MB24	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
CAN1MB25	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
CAN1MB26	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
CAN1MB27	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
CAN1MB28	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
CAN1MB29	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
CAN1MB30	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
CAN1MB31	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	

SPCR	—	—	—	—	—	—	—	—	HSPI
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	FBS	CLKP	IDIV	CLKC4	CLKC3	CLKC2	CLKC1	CLKC0	
SPSR	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	TXFU	TXHA	TXEM	
	RXFU	RXHA	RXEM	RXOO	RXOW	RXFL	TXFN	TXFL	
SPSCR	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	TEIE	THIE	RNIE	RHIE	RFIE	FFEN	
	LMSB	CSV	CSA	TFIE	ROIE	RXDE	TXDE	MASL	
SPTBR	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	TD	TD	TD	TD	TD	TD	TD	TD	
SPRBR	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	RD	RD	RD	RD	RD	RD	RD	RD	
PACR	PA7MD1	PA7MD0	PA6MD1	PA6MD0	PA5MD1	PA5MD0	PA4MD1	PA4MD0	PFC
	PA3MD1	PA3MD0	PA2MD1	PA2MD0	—	—	—	—	
PBCR	PB7MD1	PB7MD0	PB6MD1	PB6MD0	PB5MD1	PB5MD0	PB4MD1	PB4MD0	
	PB3MD1	PB3MD0	PB2MD1	PB2MD0	PB1MD1	PB1MD0	—	—	
PCCR	PC7MD1	PC7MD0	PC6MD1	PC6MD0	PC5MD1	PC5MD0	PC4MD1	PC4MD0	
	PC3MD1	PC3MD0	PC2MD1	PC2MD0	PC1MD1	PC1MD0	PC0MD1	PC0MD0	
PDCR	PD7MD1	PD7MD0	PD6MD1	PD6MD0	PD5MD1	PD5MD0	PD4MD1	PD4MD0	
	PD3MD1	PD3MD0	PD2MD1	PD2MD0	PD1MD1	PD1MD0	PD0MD1	PD0MD0	
PECR	PE7MD1	PE7MD0	PE6MD1	PE6MD0	PE5MD1	PE5MD0	PE4MD1	PE4MD0	
	PE3MD1	PE3MD0	PE2MD1	PE2MD0	PE1MD1	PE1MD0	PE0MD1	PE0MD0	
PFCR	—	—	—	—	—	—	—	—	
	PF3MD1	PF3MD0	PF2MD1	PF2MD0	PF1MD1	PF1MD0	PF0MD1	PF0MD0	

PGCR	PG7MD1	PG7MD0	PG6MD1	PG6MD0	PG5MD1	PG5MD0	PG4MD1	PG4MD0	PFC1
	PG3MD1	PG3MD0	PG2MD1	PG2MD0	PG1MD1	PG1MD0	PG0MD1	PG0MD0	
PHCR	PH7MD1	PH7MD0	PH6MD1	PH6MD0	PH5MD1	PH5MD0	PH4MD1	PH4MD0	
	PH3MD1	PH3MD0	PH2MD1	PH2MD0	PH1MD1	PH1MD0	PH0MD1	PH0MD0	
PJCR	PJ7MD1	PJ7MD0	PJ6MD1	PJ6MD0	PJ5MD1	PJ5MD0	PJ4MD1	PJ4MD0	
	PJ3MD1	PJ3MD0	PJ2MD1	PJ2MD0	PJ1MD1	PJ1MD0	—	—	
PKCR	PK7MD1	PK7MD0	PK6MD1	PK6MD0	PK5MD1	PK5MD0	PK4MD1	PK4MD0	
	PK3MD1	PK3MD0	PK2MD1	PK2MD0	—	—	—	—	
PADR	PA7DT	PA6DT	PA5DT	PA4DT	PA3DT	PA2DT	—	—	
PBDR	PB7DT	PB6DT	PB5DT	PB4DT	PB3DT	PB2DT	PB1DT	—	
PCDR	PC7DT	PC6DT	PC5DT	PC4DT	PC3DT	PC2DT	PC1DT	PC0DT	
PDDR	PD7DT	PD6DT	PD5DT	PD4DT	PD3DT	PD2DT	PD1DT	PD0DT	
PEDR	PE7DT	PE6DT	PE5DT	PE4DT	PE3DT	PE2DT	PE1DT	PE0DT	
PFDR	—	—	—	—	PF3DT	PF2DT	PF1DT	PF0DT	
PGDR	PG7DT	PG6DT	PG5DT	PG4DT	PG3DT	PG2DT	PG1DT	PG0DT	
PHDR	PH7DT	PH6DT	PH5DT	PH4DT	PH3DT	PH2DT	PH1DT	PH0DT	
PJDR	PJ7DT	PJ6DT	PJ5DT	PJ4DT	PJ3DT	PJ2DT	PJ1DT	—	
PKDR	PK7DT	PK6DT	PK5DT	PK4DT	PK3DT	PK2DT	—	—	
INPUPA	MD6PUP	RDYPUP	BREQPUP	IRL0PUP	IRL1PUP	IRL2PUP	IRL3PUP	NMIPUP	
	—	—	—	—	—	—	—	—	
DMAPCR	DREQP0	—	DREQP1	—	DACKP0	DACKD0	DACKP1	DACKD1	
	DRAKP0	DRAKD0	DRAKP1	DRAKD1	—	—	—	BRGRST	
SCIHZR	SCICLK0	SCIRXD0	SCITXD0	SCICLK1	SCICTS1	SCIRTS1	SCIRXD1	SCITXD1	
	SCICLK2	SCICTS2	SCIRTS2	SCIRXD2	SCITXD2	—	—	—	
IPSELR	IPSELR15	IPSELR14	IPSELR13	IPSELR12	IPSELR11	IPSELR10	IPSELR9	—	
	—	—	—	—	—	—	LCDMD1	LCDMD0	
PAPUPR	PA7PUPR	PA6PUPR	PA5PUPR	PA4PUPR	PA3PUPR	PA2PUPR	—	—	
PBPUPR	PB7PUPR	PB6PUPR	PB5PUPR	PB4PUPR	PB3PUPR	PB2PUPR	PB1PUPR	—	
PCPUPR	PC7PUPR	PC6PUPR	PC5PUPR	PC4PUPR	PC3PUPR	PC2PUPR	PC1PUPR	PC0PUPR	
PDPUPR	PD7PUPR	PD6PUPR	PD5PUPR	PD4PUPR	PD3PUPR	PD2PUPR	PD1PUPR	PD0PUPR	
PEPUPR	PE7PUPR	PE6PUPR	PE5PUPR	PE4PUPR	PE3PUPR	PE2PUPR	PE1PUPR	PE0PUPR	
PFPUPR	—	—	—	—	PF3PUPR	PF2PUPR	PF1PUPR	PF0PUPR	
PGPUPR	PG7PUPR	PG6PUPR	PG5PUPR	PE4PUPR	PG3PUPR	PG2PUPR	PG1PUPR	PG0PUPR	

PHPUPR	PH7PUPR	PH6PUPR	PH5PUPR	PH4PUPR	PH3PUPR	PH2PUPR	PH1PUPR	PH0PUPR	PFC
PJPUPR	PJ7PUPR	PJ6PUPR	PJ5PUPR	PJ4PUPR	PH3PUPR	PJ2PUPR	—	—	
PKPUPR	PK7PUPR	PK6PUPR	PK5PUPR	PK4PUPR	PK3PUPR	PK2PUPR	—	—	
MDPUPR	MDPUPR7	MDPUPR6	MDPUPR5	MDPUPR4	MDPUPR3	MDPUPR2	MDPUPR1	MDPUPR0	
MODSELR	MODSELR7	MODSELR6	MODSELR5	MODSELR4	MODSELR3	MODSELR2	MODSELR1	—	
GPIOIC	PTIREN15	PTIREN14	PTIREN13	PTIREN12	PTIREN11	PTIREN10	PTIREN9	STBRT8	
	STBRT7	STBRT6	STBIRQ5	STBIRQ4	STBIRL3	STBIRL2	STBIRL1	STBIRL0	
HACCR0	—	—	—	—	—	—	—	—	HAC(0)
	—	—	—	—	—	—	—	—	
	CR	—	—	—	CDRT	WMRT	—	—	
	—	—	ST	—	—	—	—	—	
HACCSAR0	—	—	—	—	—	—	—	—	
	—	—	—	—	RW	CA6/SA6	CA5/SA5	CA4/SA4	
	CA3/SA3	CA2/SA2	CA1/SA1	CA0/SA0	SLREQ3	SLREQ4	SLREQ5	SLREQ6	
	SLREQ7	SLREQ8	SLREQ9	SLREQ10	SLREQ11	SLREQ12	—	—	
HACCSDR0	—	—	—	—	—	—	—	—	
	—	—	—	—	CD15/SD15	CD14/SD14	CD13/SD13	CD12/SD12	
	CD11/SD11	CD10/SD10	CD9/SD9	CD8/SD8	CD7/SD7	CD6/SD6	CD5/SD5	CD4/SD4	
	CD3/SD3	CD2/SD2	CD1/SD1	CD0/SD0	—	—	—	—	
HACPCML0	—/LD15	—/LD14	—/LD13	—/LD12	—/LD11	—/LD10	—/LD9	—/LD8	
	—/LD7	—/LD6	—/LD5	—/LD4	D19/LD3	D18/LD2	D17/LD1	D16/LD0	
	D15/RD15	D14/RD14	D13/RD13	D12/RD12	D11/RD11	D10/RD10	D9/RD9	D8/RD8	
	D7/RD7	D6/RD6	D5/RD5	D4/RD4	D3/RD3	D2/RD2	D1/RD1	D0/RD0	
HACPCMR0	—	—	—	—	—	—	—	—	
	—	—	—	—	D19	D18	D17	D16	
	D15	D14	D13	D12	D11	D10	D9	D8	
	D7	D6	D5	D4	D3	D2	D1	D0	
HACTIER0	—	—	PLTFRQIE	PRTFRQIE	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	PLTFUNIE	PRTFUNIE	
	—	—	—	—	—	—	—	—	

HACTSR0	CMDAMT	CMDDMT	PLTFRQ	PRTFRQ	—	—	—	—	HAC(0)
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	PLTFUN	PRTFUN	
	—	—	—	—	—	—	—	—	
HACRIER0	—	—	—	—	—	—	—	—	
	—	STARYIE	STDRYIE	PLRFRQIE	PRRFRQIE	—	—	—	
	—	—	PLRFOVIE	PRRFOVIE	—	—	—	—	
	—	—	—	—	—	—	—	—	
HACRSR0	—	—	—	—	—	—	—	—	
	—	STARY	STDRY	PLRFRQ	PRRFRQ	—	—	—	
	—	—	PLRFOV	PRRFOV	—	—	—	—	
	—	—	—	—	—	—	—	—	
HACACR0	—	DMARX16	DMATX16	—	—	TX12_ ATOMIC	—	RXDMAL_ EN	
	TXDMAL_ EN	RXDMAR_ EN	TXDMAR_ EN	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
HACCR1	—	—	—	—	—	—	—	—	HAC(1)
	—	—	—	—	—	—	—	—	
	CR	—	—	—	CDRT	WMRT	—	—	
	—	—	ST	—	—	—	—	—	
HACCSAR1	—	—	—	—	—	—	—	—	
	—	—	—	—	RW	CA6/SA6	CA5/SA5	CA4/SA4	
	CA3/SA3	CA2/SA2	CA1/SA1	CA0/SA0	SLREQ3	SLREQ4	SLREQ5	SLREQ6	
	SLREQ7	SLREQ8	SLREQ9	SLREQ10	SLREQ11	SLREQ12	—	—	
HACCSDR1	—	—	—	—	—	—	—	—	
	—	—	—	—	CD15/SD15	CD14/SD14	CD13/SD13	CD12/SD12	
	CD11/SD11	CD10/SD10	CD9/SD9	CD8/SD8	CD7/SD7	CD6/SD6	CD5/SD5	CD4/SD4	
	CD3/SD3	CD2/SD2	CD1/SD1	CD0/SD0	—	—	—	—	
HACPCML1	—/LD15	—/LD14	—/LD13	—/LD12	—/LD11	—/LD10	—/LD9	—/LD8	
	—/LD7	—/LD6	—/LD5	—/LD4	D19/LD3	D18/LD2	D17/LD1	D16/LD0	
	D15/RD15	D14/RD14	D13/RD13	D12/RD12	D11/RD11	D10/RD10	D9/RD9	D8/RD8	
	D7/RD7	D6/RD6	D5/RD5	D4/RD4	D3/RD3	D2/RD2	D1/RD1	D0/RD0	

HACPCMR1	—	—	—	—	—	—	—	—	HAC(1)
	—	—	—	—	D19	D18	D17	D16	
	D15	D14	D13	D12	D11	D10	D9	D8	
	D7	D6	D5	D4	D3	D2	D1	D0	
HACTIER1	—	—	PLTFRQIE	PRTFRQIE	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	PLTFUNIE	PRTFUNIE	
	—	—	—	—	—	—	—	—	
HACTSR1	CMDAMT	CMDDMT	PLTFRQ	PRTFRQ	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	PLTFUN	PRTFUN	
	—	—	—	—	—	—	—	—	
HACRIER1	—	—	—	—	—	—	—	—	
	—	STARYIE	STDRYIE	PLRFRQIE	PRRFRQIE	—	—	—	
	—	—	PLRFOVIE	PRRFOVIE	—	—	—	—	
	—	—	—	—	—	—	—	—	
HACRSR1	—	—	—	—	—	—	—	—	
	—	STARY	STDRY	PLRFRQ	PRRFRQ	—	—	—	
	—	—	PLRFOV	PRRFOV	—	—	—	—	
	—	—	—	—	—	—	—	—	
HACACR1	—	DMARX16	DMATX16	—	—	TX12_ ATOMIC	—	RXDMAL_ EN	
	TXDMAL_ EN	RXDMAR_ EN	TXDMAR_ EN	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
CMDR0	Start	Host	INDEX	INDEX	INDEX	INDEX	INDEX	INDEX	MMCIF
CMDR1	—	—	—	—	—	—	—	—	
CMDR2	—	—	—	—	—	—	—	—	
CMDR3	—	—	—	—	—	—	—	—	
CMDR4	—	—	—	—	—	—	—	—	
CMDR5	CRC	CRC	CRC	CRC	CRC	CRC	CRC	End	
CMDSTRT	—	—	—	—	—	—	—	START	
OPCR	CMDOFF	—	RD_CONTI	DATAEN	—	—	—	—	

CSTR	BUSY	FIFO_FULL	FIFO_ EMPTY	CWRE	DTBUSY	DTBUSY_ TU	—	REQ	MMCIF
INTCR0	FEIE	FFIE	DRPIE	DTIE	CRPIE	CMDIE	DBSYIE	—	
INTCR1	INTRQ2E	INTRQ1E	INTRQ0E	—	—	CRCERIE	DTERIE	CTERIE	
INTSTR0	FEI	FFI	DRPI	DTI	CRPI	CMDI	DBSYI	—	
INTSTR1	—	—	—	—	—	CRCERI	DTERI	CTERI	
CLKON	CLKON	—	—	—	—	CSEL2	CSEL1	CSEL0	
CTOCR	—	—	—	—	—	—	CTSEL1	CTSEL0	
TBCR	—	—	—	—	C3	C2	C1	C0	
MODER	—	—	—	—	—	—	—	MODE	
CMDTYR	—	—	—	TY4	TY3	TY2	TY1	TY0	
RSPTYR	—	—	RTY5	RTY4	—	RTY2	RTY1	RTY0	
RSPR0	RSPR	RSPR	RSPR	RSPR	RSPR	RSPR	RSPR	RSPR	
RSPR1	RSPR	RSPR	RSPR	RSPR	RSPR	RSPR	RSPR	RSPR	
RSPR2	RSPR	RSPR	RSPR	RSPR	RSPR	RSPR	RSPR	RSPR	
RSPR3	RSPR	RSPR	RSPR	RSPR	RSPR	RSPR	RSPR	RSPR	
RSPR4	RSPR	RSPR	RSPR	RSPR	RSPR	RSPR	RSPR	RSPR	
RSPR5	RSPR	RSPR	RSPR	RSPR	RSPR	RSPR	RSPR	RSPR	
RSPR6	RSPR	RSPR	RSPR	RSPR	RSPR	RSPR	RSPR	RSPR	
RSPR7	RSPR	RSPR	RSPR	RSPR	RSPR	RSPR	RSPR	RSPR	
RSPR8	RSPR	RSPR	RSPR	RSPR	RSPR	RSPR	RSPR	RSPR	
RSPR9	RSPR	RSPR	RSPR	RSPR	RSPR	RSPR	RSPR	RSPR	
RSPR10	RSPR	RSPR	RSPR	RSPR	RSPR	RSPR	RSPR	RSPR	
RSPR11	RSPR	RSPR	RSPR	RSPR	RSPR	RSPR	RSPR	RSPR	
RSPR12	RSPR	RSPR	RSPR	RSPR	RSPR	RSPR	RSPR	RSPR	
RSPR13	RSPR	RSPR	RSPR	RSPR	RSPR	RSPR	RSPR	RSPR	
RSPR14	RSPR	RSPR	RSPR	RSPR	RSPR	RSPR	RSPR	RSPR	
RSPR15	RSPR	RSPR	RSPR	RSPR	RSPR	RSPR	RSPR	RSPR	
RSPR16	RSPR	RSPR	RSPR	RSPR	RSPR	RSPR	RSPR	RSPR	
DTOUTR	DTOUTR	DTOUTR	DTOUTR	DTOUTR	DTOUTR	DTOUTR	DTOUTR	DTOUTR	
	DTOUTR	DTOUTR	DTOUTR	DTOUTR	DTOUTR	DTOUTR	DTOUTR	DTOUTR	
DR	DR	DR	DR	DR	DR	DR	DR	DR	
	DR	DR	DR	DR	DR	DR	DR	DR	

FIFOCLR	FIFOCLR	FIFOCLR	FIFOCLR	FIFOCLR	FIFOCLR	FIFOCLR	FIFOCLR	FIFOCLR	MMCIF
DMACR	DMAEN	—	—	—	—	SET2	SET1	SET0	
INTCR2	—	—	—	—	—	—	—	FRDYIE	
INTSTR2	—	—	—	—	—	—	FRDY_TU	FRDYI	
RDTIMSEL	—	—	—	—	—	—	—	RTSEL	
MFIIDX	—	—	—	—	—	—	—	—	MFI
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	REG5	REG4	REG3	REG2	REG1	REG0	BYTE1	BYTE0	
MFIGSR	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	STATUS7	STATUS6	STATUS5	STATUS4	STATUS3	STATUS2	STATUS1	STATUS0	
MFISCR	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	SCRMD2	—	SCRMD0	—	—	EDN	BO	
MFIMCR	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	LOCK	—	WT	—	RD	—	—	AI/AD	
MFIICR	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	IIC6	IIC5	IIC4	IIC3	IIC2	IIC1	IIC0	IIR	
MFIEICR	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	EIC6	EIC5	EIC4	EIC3	EIC2	EIC1	EIC0	EIR	
MFIADR	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	A10	A9	A8	
	A7	A6	A5	A4	A3	A2	—	—	

MFIDATA	MFIDATA 31	MFIDATA 30	MFIDATA 29	MFIDATA 28	MFIDATA 27	MFIDATA 26	MFIDATA 25	MFIDATA 24	MFI
	MFIDATA 23	MFIDATA 22	MFIDATA 21	MFIDATA 20	MFIDATA 19	MFIDATA 18	MFIDATA 17	MFIDATA 16	
	MFIDATA 15	MFIDATA 14	MFIDATA 13	MFIDATA 12	MFIDATA 11	MFIDATA 10	MFIDATA 9	MFIDATA 8	
	MFIDATA 7	MFIDATA 6	MFIDATA 5	MFIDATA 4	MFIDATA 3	MFIDATA 2	MFIDATA 1	MFIDATA 0	
MFRAM Start	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
MFRAM End	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
SDIR	TI7	TI6	TI5	TI4	TI3	TI2	TI1	TI0	H-UDI
	—	—	—	—	—	—	—	—	
SDDR	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
SDINT	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	INTREQ	
ADDRA	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	ADC
	AD1	AD0	—	—	—	—	—	—	
ADDRB	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
	AD1	AD0	—	—	—	—	—	—	
ADDRC	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
	AD1	AD0	—	—	—	—	—	—	
ADDRD	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
	AD1	AD0	—	—	—	—	—	—	
ADCSR	ADF	ADIE	ADST	DMASL	TRGE1	TRGE0	—	—	
	CKSL1	CKSL0	MDS1	MDS0	—	—	CH1	CH0	

LDICKR	—	—	ICKSEL1	ICKSEL0	—	—	—	—	LCDC
	—	—	—	DCDR4	DCDR3	DCDR2	DCDR1	DCDR0	
LDMTR	FLMPOL	CL1POL	DISPPOL	DPOL	—	MCNT	CL1CNT	CL2CNT	
	—	—	MIFTYP5	MIFTYP4	MIFTYP3	MIFTYP2	MIFTYP1	MIFTYP0	
LDDFR	—	—	—	—	—	—	—	PABD	
	—	DSPCOLOR6	DSPCOLOR5	DSPCOLOR4	DSPCOLOR3	DSPCOLOR2	DSPCOLOR1	DSPCOLOR0	
LDSMR	—	—	ROT	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
LDSARU	—	—	—	—	—	—	SAU25	SAU24	
	SAU23	SAU22	SAU21	SAU20	SAU19	SAU18	SAU17	SAU16	
	SAU15	SAU14	SAU13	SAU12	SAU11	SAU10	SAU9	SAU8	
	SAU7	SAU6	SAU5	SAU4	SAU3	SAU2	SAU1	SAU0	
LDSARL	—	—	—	—	—	—	SAL25	SAL24	
	SAL23	SAL22	SAL21	SAL20	SAL19	SAL18	SAL17	SAL16	
	SAL15	SAL14	SAL13	SAL12	SAL11	SAL10	SAL9	SAL8	
	SAL7	SAL6	SAL5	SAL4	SAL3	SAL2	SAL1	SAL0	
LDLAOR	LAO15	LAO14	LAO13	LAO12	LAO11	LAO10	LAO9	LAO8	
	LAO7	LAO6	LAO5	LAO4	LAO3	LAO2	LAO1	LAO0	
LDPALCR	—	—	—	—	—	—	—	—	
	—	—	—	PALS	—	—	—	PALEN	
LDPR00–FF	—	—	—	—	—	—	—	—	
	PALDnn_23	PALDnn_22	PALDnn_21	PALDnn_20	PALDnn_19	PALDnn_18	PALDnn_17	PALDnn_16	
	PALDnn_15	PALDnn_14	PALDnn_13	PALDnn_12	PALDnn_11	PALDnn_10	PALDnn_9	PALDnn_8	
	PALDnn_7	PALDnn_6	PALDnn_5	PALDnn_4	PALDnn_3	PALDnn_2	PALDnn_1	PALDnn_0	
LDHCNR	HDCN7	HDCN6	HDCN5	HDCN4	HDCN3	HDCN2	HDCN1	HDCN0	
	HTCN7	HTCN6	HTCN5	HTCN4	HTCN3	HTCN2	HTCN1	HTCN0	
LDHSYNR	HSYNW3	HSYNW2	HSYNW1	HSYNW0	—	—	—	—	
	HSYNP7	HSYNP6	HSYNP5	HSYNP4	HSYNP3	HSYNP2	HSYNP1	HSYNP0	
LDVDLNR	—	—	—	—	—	VDLN10	VDLN9	VDLN8	
	VDLN7	VDLN6	VDLN5	VDLN4	VDLN3	VDLN2	VDLN1	VDLN0	
LDVTLNR	—	—	—	—	—	VTLN10	VTLN9	VTLN8	
	VTLN7	VTLN6	VTLN5	VTLN4	VTLN3	VTLN2	VTLN1	VTLN0	

LDVSYNR	VSYNW3	VSYNW2	VSYNW1	VSYNW0	—	VSYNP10	VSYNP9	VSYNP8	LCDC
	VSYNP7	VSYNP6	VSYNP5	VSYNP4	VSYNP3	VSYNP2	VSYNP1	VSYNP0	
LDACLNR	—	—	—	—	—	—	—	—	
	—	—	—	ACLN4	ACLN3	ACLN2	ACLN1	ACLN0	
LDINTR	—	—	—	VINTSEL	—	—	—	VINTE	
	—	—	—	—	—	—	—	VINTS	
LDPMMR	ONC3	ONC2	ONC1	ONC0	OFFD3	OFFD2	OFFD1	OFFD0	
	—	VCPE	VEPE	DONE	—	—	LPS1	LPS0	
LDPSPR	ONA3	ONA2	ONA1	ONA0	ONB3	ONB2	ONB1	ONB0	
	OFFE3	OFFE2	OFFE1	OFFE0	OFFF3	OFFF2	OFFF1	OFFF0	
LDCNTR	—	—	—	—	—	—	—	—	
	—	—	—	DON2	—	—	—	DON	
BASRA	BASA7	BASA6	BASA5	BASA4	BASA3	BASA2	BASA1	BASA0	UBC
BASRB	BASB7	BASB6	BASB5	BASB4	BASB3	BASB2	BASB1	BASB0	
BARA	BAA31	BAA30	BAA29	BAA28	BAA27	BAA26	BAA25	BAA24	
	BAA23	BAA22	BAA21	BAA20	BAA19	BAA18	BAA17	BAA16	
	BAA15	BAA14	BAA13	BAA12	BAA11	BAA10	BAA9	BAA8	
	BAA7	BAA6	BAA5	BAA4	BAA3	BAA2	BAA1	BAA0	
BAMRA	—	—	—	—	BAMA2	BASMA	BAMA1	BAMA0	
BBRA	—	—	—	—	—	—	—	—	
	—	SZA2	IDA1	IDA0	RWA1	RWA0	SZA1	SZA0	
BARB	BAB31	BAB30	BAB29	BAB28	BAB27	BAB26	BAB25	BAB24	
	BAB23	BAB22	BAB21	BAB20	BAB19	BAB18	BAB17	BAB16	
	BAB15	BAB14	BAB13	BAB12	BAB11	BAB10	BAB9	BAB8	
	BAB7	BAB6	BAB5	BAB4	BAB3	BAB2	BAB1	BAB0	
BAMRB	—	—	—	—	BAMB2	BASMB	BAMB1	BAMB0	
BBRB	—	—	—	—	—	—	—	—	
	—	SZB2	IDB1	IDB0	RWB1	RWB0	SZB1	SZB0	
BDRB	BDB31	BDB30	BDB29	BDB28	BDB27	BDB26	BDB25	BDB24	
	BDB23	BDB22	BDB21	BDB20	BDB19	BDB18	BDB17	BDB16	
	BDB15	BDB14	BDB13	BDB12	BDB11	BDB10	BDB9	BDB8	
	BDB7	BDB6	BDB5	BDB4	BDB3	BDB2	BDB1	BDB0	

BDMRB	BDMB31	BDMB30	BDMB29	BDMB28	BDMB27	BDMB26	BDMB25	BDMB24	UBC
	BDMB23	BDMB22	BDMB21	BDMB20	BDMB19	BDMB18	BDMB17	BDMB16	
	BDMB15	BDMB14	BDMB13	BDMB12	BDMB11	BDMB10	BDMB9	BDMB8	
	BDMB7	BDMB6	BDMB5	BDMB4	BDMB3	BDMB2	BDMB1	BDMB0	
BRCR	CMFA	CMFB	—	—	—	PCBA	—	—	
	DBEB	PCBB	—	—	SEQ	—	—	UBDE	

Module	Abbrev.	Power-on Reset by $\overline{\text{RESET}}$ Pin/WDT/ H-UDI	Manual Reset by Sleep $\overline{\text{RESET}}$ Pin/WDT/ Multiple Exception	by Sleep Instruction/ Deep Sleep	Standby by Hardware	Standby by Software/ Each Module
MMU	PTEH	Undefined	Undefined	Retained	Undefined	Retained
	PTEL	Undefined	Undefined	Retained	Undefined	Retained
	PTEA	Undefined	Undefined	Retained	Undefined	Retained
	TTB	Undefined	Retained	Retained	Undefined	Retained
	TEA	Undefined	Undefined	Retained	Undefined	Retained
	MMUCR	H'0000 0000	H'0000 0000	Retained	H'0000 0000	Retained
Cache	CCR	H'0000 0000	H'0000 0000	Retained	H'0000 0000	Retained
	QACR0	Undefined	Undefined	Retained	Undefined	Retained
	QACR1	Undefined	Undefined	Retained	Undefined	Retained
Exception handling	TRA	Undefined	Undefined	Retained	Undefined	Retained
	EXPEVT	H'0000 0000	H'0000 0020	Retained	H'0000 0000	Retained
	INTEVT	Undefined	Undefined	Retained	Undefined	Retained
INTC	ICR	H'0000*1	H'0000*1	Retained	H'0000*1	Retained
		H'8000*2	H'8000*2	Retained	H'8000*2	Retained
	IPRA	H'0000	H'0000	Retained	H'0000	Retained
	IPRB	H'0000	H'0000	Retained	H'0000	Retained
	IPRC	H'0000	H'0000	Retained	H'0000	Retained
	IPRD	H'DA74	H'DA74	Retained	H'DA74	Retained
	INTPRI00	H'0000 0000	Retained	Retained	H'0000 0000	Retained
	INTPRI04	H'0000 0000	Retained	Retained	H'0000 0000	Retained
	INTPRI08	H'0000 0000	Retained	Retained	H'0000 0000	Retained
	INTPRI0C	H'0000 0000	Retained	Retained	H'0000 0000	Retained
	INTREQ00	H'0000 0000	Retained	Retained	H'0000 0000	Retained
	INTREQ04	H'0000 0000	Retained	Retained	H'0000 0000	Retained
	INTMSK00	H'F3FF 7FFF	Retained	Retained	H'F3FF 7FFF	Retained
	INTMSK04	H'00FF FFFF	Retained	Retained	H'00FF FFFF	Retained
	INTMSKCLR00	—	—	—	—	—
	INTMSKCLR04	—	—	—	—	—

Module	Abbrev.	by RESET Pin/WDT/ H-UDI	RESET Pin/WDT/by Sleep Multiple Exception	Instruction/ Deep Sleep	by Hardware	by Software/ Each Module
BSC	BCR1	H'0000 0000	Retained	Retained	H'0000 0000	Retained
	BCR2	H'3FFC	Retained	Retained	H'3FFC	Retained
	BCR3	H'0001	Retained	Retained	H'0001	Retained
	BCR4	H'0000 0000	Retained	Retained	H'0000 0000	Retained
	WCR1	H'7777 7777	Retained	Retained	H'7777 7777	Retained
	WCR2	H'FFFE EFFF	Retained	Retained	H'FFFE EFFF	Retained
	WCR3	H'0777 7777	Retained	Retained	H'0777 7777	Retained
	WCR4	H'0000 0000	Retained	Retained	H'0000 0000	Retained
	MCR	H'0000 0000	Retained	Retained	H'0000 0000	Retained
	PCR	H'0000	Retained	Retained	H'0000	Retained
	RTCSR	H'0000	Retained	Retained	H'0000	Retained
	RTCNT	H'0000	Retained	Retained	H'0000	Retained
	RTCOR	H'0000	Retained	Retained	H'0000	Retained
	RFCR	H'0000	Retained	Retained	H'0000	Retained
	SDMR2	Write only				
	SDMR3	Write only				
DMAC Channel 0	SAR0	Undefined	Undefined	Retained	Undefined	Retained
	DAR0	Undefined	Undefined	Retained	Undefined	Retained
	DMATCR0	Undefined	Undefined	Retained	Undefined	Retained
	CHCR0	H'0000 0000	H'0000 0000	Retained	H'0000 0000	Retained
DMAC Channel 1	SAR1	Undefined	Undefined	Retained	Undefined	Retained
	DAR1	Undefined	Undefined	Retained	Undefined	Retained
	DMATCR1	Undefined	Undefined	Retained	Undefined	Retained
	CHCR1	H'0000 0000	H'0000 0000	Retained	H'0000 0000	Retained
DMAC Channel 2	SAR2	Undefined	Undefined	Retained	Undefined	Retained
	DAR2	Undefined	Undefined	Retained	Undefined	Retained
	DMATCR2	Undefined	Undefined	Retained	Undefined	Retained
	CHCR2	H'0000 0000	H'0000 0000	Retained	H'0000 0000	Retained

Module	Abbrev.	by RESET Pin/WDT/ H-UDI	RESET Pin/WDT/ Multiple Exception	by Sleep Instruction/ Deep Sleep	by Hardware	by Software/ Each Module
DMAC Channel 3	SAR3	Undefined	Undefined	Retained	Undefined	Retained
	DAR3	Undefined	Undefined	Retained	Undefined	Retained
	DMATCR3	Undefined	Undefined	Retained	Undefined	Retained
	CHCR3	H'0000 0000	H'0000 0000	Retained	H'0000 0000	Retained
DMAC Channel 4	SAR4	Undefined	Undefined	Retained	Undefined	Retained
	DAR4	Undefined	Undefined	Retained	Undefined	Retained
	DMATCR4	Undefined	Undefined	Retained	Undefined	Retained
	CHCR4	H'0000 0000	H'0000 0000	Retained	H'0000 0000	Retained
DMAC Channel 5	SAR5	Undefined	Undefined	Retained	Undefined	Retained
	DAR5	Undefined	Undefined	Retained	Undefined	Retained
	DMATCR5	Undefined	Undefined	Retained	Undefined	Retained
	CHCR5	H'0000 0000	H'0000 0000	Retained	H'0000 0000	Retained
DMAC Channel 6	SAR6	Undefined	Undefined	Retained	Undefined	Retained
	DAR6	Undefined	Undefined	Retained	Undefined	Retained
	DMATCR6	Undefined	Undefined	Retained	Undefined	Retained
	CHCR6	H'0000 0000	H'0000 0000	Retained	H'0000 0000	Retained
DMAC Channel 7	SAR7	Undefined	Undefined	Retained	Undefined	Retained
	DAR7	Undefined	Undefined	Retained	Undefined	Retained
	DMATCR7	Undefined	Undefined	Retained	Undefined	Retained
	CHCR7	H'0000 0000	H'0000 0000	Retained	H'0000 0000	Retained
DMAC Common	DMAOR	H'0000 0000	H'0000 0000	Retained	H'0000 0000	Retained
	DMARSRA	H'0000 0000	H'0000 0000	Retained	H'0000 0000	Retained
	DMARSRB	H'0000 0000	H'0000 0000	Retained	H'0000 0000	Retained
DMAC DMABRG	DMARCR	H'0000 0000	H'0000 0000	Retained	H'0000 0000	Retained
	DMABRGCR	H'0000 0000	H'0000 0000	Retained	H'0000 0000	Retained
	DMAATXSAR0	Undefined	Undefined	Retained	Undefined	Retained
	DMAARXDAR0	Undefined	Undefined	Retained	Undefined	Retained
	DMAATXTCR0	Undefined	Undefined	Retained	Undefined	Retained
	DMAARXTCR0	Undefined	Undefined	Retained	Undefined	Retained
	DMAACR0	H'0000 0000	H'0000 0000	Retained	H'0000 0000	Retained
	DMAATXTCNT0	Undefined	Undefined	Retained	Undefined	Retained

Module	Abbrev.	by RESET Pin/WDT/ H-UDI	RESET Pin/WDT/ by Sleep Multiple Exception	Instruction/ Deep Sleep	by Hardware	by Software/ Each Module
DMAC DMABRG	DMAARXTCNT0	Undefined	Undefined	Retained	Undefined	Retained
	DMAATXSAR1	Undefined	Undefined	Retained	Undefined	Retained
	DMAARXDAR1	Undefined	Undefined	Retained	Undefined	Retained
	DMAATXTCR1	Undefined	Undefined	Retained	Undefined	Retained
	DMAARXTCR1	Undefined	Undefined	Retained	Undefined	Retained
	DMAACR1	H'0000 0000	H'0000 0000	Retained	H'0000 0000	Retained
	DMAATXTCNT1	Undefined	Undefined	Retained	Undefined	Retained
	DMAARXTCNT1	Undefined	Undefined	Retained	Undefined	Retained
	DMAUSAR	H'0000 0000	H'0000 0000	Retained	H'0000 0000	Retained
	DMAUDAR	H'0000 0000	H'0000 0000	Retained	H'0000 0000	Retained
	DMAURWSZ	H'0000 0000	H'0000 0000	Retained	H'0000 0000	Retained
	DMAUCR	H'0000 0000	H'0000 0000	Retained	H'0000 0000	Retained
CPG	FRQCR	*3/ Retained	Retained	Retained	*3/ Retained	Retained
	DCKDR	H'0000 0001	Retained	Retained	H'0000 0001	Retained
	MCKCR	H'0000 0000	Retained	Retained	H'0000 0000	Retained
WDT	WTCNT	H'00/ Retained	Retained	Retained	H'00/ Retained	Retained
	WTCSR	H'00/ Retained	Retained	Retained	H'00/ Retained	Retained
Power-down	STBCR	H'00	Retained	Retained	H'00	Retained
	STBCR2	H'00	Retained	Retained	H'00	Retained
	CLKSTP00	H'0000 0000	Retained	Retained	H'0000 0000	Retained
	CLKSTPCLR00	—	—	Retained	—	Retained
TMU Common	TSTR	H'00	H'00	Retained	H'00	Retained
TMU Channel 0	TCOR0	H'FFFF FFFF	H'FFFF FFFF	Retained	H'FFFF FFFF	Retained
	TCNT0	H'FFFF FFFF	H'FFFF FFFF	Retained	H'FFFF FFFF	Retained
	TCR0	H'0000	H'0000	Retained	H'0000	Retained
TMU Channel 1	TCOR1	H'FFFF FFFF	H'FFFF FFFF	Retained	H'FFFF FFFF	Retained
	TCNT1	H'FFFF FFFF	H'FFFF FFFF	Retained	H'FFFF FFFF	Retained
	TCR1	H'0000	H'0000	Retained	H'0000	Retained

Module	Abbrev.	by RESET Pin/WDT/ H-UDI	RESET Pin/WDT/ by Sleep Multiple Exception	Instruction/ Deep Sleep	by Hardware	by Software/ Each Module
TMU Channel 2	TCOR2	H'FFFF FFFF	H'FFFF FFFF	Retained	H'FFFF FFFF	Retained
	TCNT2	H'FFFF FFFF	H'FFFF FFFF	Retained	H'FFFF FFFF	Retained
	TCR2	H'0000	H'0000	Retained	H'0000	Retained
	TCPR2	Retained	Retained	Retained	Retained	Retained
CMT Common	CMTCFG	H'0000 0000	H'0000 0000	Retained	H'0000 0000	Retained
	CMTFRT	H'0000 0000	H'0000 0000	Retained	H'0000 0000	Retained
	CMTCTL	H'0000 0000	H'0000 0000	Retained	H'0000 0000	Retained
	CMTIRQS	H'0000 0000	H'0000 0000	Retained	H'0000 0000	Retained
CMT Channel 0	CMTCH0T	H'0000 0000	H'0000 0000	Retained	H'0000 0000	Retained
	CMTCH0ST	H'0000 0000	H'0000 0000	Retained	H'0000 0000	Retained
	CMTCH0C	H'0000 0000	H'0000 0000	Retained	H'0000 0000	Retained
CMT Channel 1	CMTCH1T	H'0000 0000	H'0000 0000	Retained	H'0000 0000	Retained
	CMTCH1ST	H'0000 0000	H'0000 0000	Retained	H'0000 0000	Retained
	CMTCH1C	H'0000 0000	H'0000 0000	Retained	H'0000 0000	Retained
CMT Channel 2	CMTCH2T	H'0000 0000	H'0000 0000	Retained	H'0000 0000	Retained
	CMTCH2ST	H'0000 0000	H'0000 0000	Retained	H'0000 0000	Retained
	CMTCH2C	H'0000 0000	H'0000 0000	Retained	H'0000 0000	Retained
CMT Channel 3	CMTCH3T	H'0000 0000	H'0000 0000	Retained	H'0000 0000	Retained
	CMTCH3ST	H'0000 0000	H'0000 0000	Retained	H'0000 0000	Retained
	CMTCH3C	H'0000 0000	H'0000 0000	Retained	H'0000 0000	Retained
SCIF Channel 0	SCSMR0	H'0000	H'0000	Retained	H'0000	Retained
	SCBRR0	H'FF	H'FF	Retained	H'FF	Retained
	SCSCR0	H'0000	H'0000	Retained	H'0000	Retained
	SCFTDR0	Undefined	Undefined	Retained	Undefined	Retained
	SCFSR0	H'0060	H'0060	Retained	H'0060	Retained
	SCFRDR0	Undefined	Undefined	Retained	Undefined	Retained
	SCFCR0	H'0000	H'0000	Retained	H'0000	Retained
	SCTFDR0	H'0000	H'0000	Retained	H'0000	Retained
	SCRFDR0	H'0000	H'0000	Retained	H'0000	Retained
	SCSPTR0	H'0000*4	H'0000*4	Retained	H'0000*4	Retained

Module	Abbrev.	by RESET Pin/WDT/ H-UDI	RESET Pin/WDT/by Sleep Multiple Exception	Instruction/ Deep Sleep	by Hardware	by Software/ Each Module
SCIF Channel 0	SCLSR0	H'0000	H'0000	Retained	H'0000	Retained
	SCRER0	H'0000	H'0000	Retained	H'0000	Retained
SCIF Channel 1	SCSMR1	H'0000	H'0000	Retained	H'0000	Retained
	SCBRR1	H'FF	H'FF	Retained	H'FF	Retained
	SCSCR1	H'0000	H'0000	Retained	H'0000	Retained
	SCFTDR1	Undefined	Undefined	Retained	Undefined	Retained
	SCFSR1	H'0060	H'0060	Retained	H'0060	Retained
	SCFRDR1	Undefined	Undefined	Retained	Undefined	Retained
	SCFCR1	H'0000	H'0000	Retained	H'0000	Retained
	SCTFDR1	H'0000	H'0000	Retained	H'0000	Retained
	SCRFDR1	H'0000	H'0000	Retained	H'0000	Retained
	SCSPTR1	H'0000*5	H'0000*5	Retained	H'0000*5	Retained
	SCLSR1	H'0000	H'0000	Retained	H'0000	Retained
	SCRER1	H'0000	H'0000	Retained	H'0000	Retained
SCIF Channel 2	SCSMR2	H'0000	H'0000	Retained	H'0000	Retained
	SCBRR2	H'FF	H'FF	Retained	H'FF	Retained
	SCSCR2	H'0000	H'0000	Retained	H'0000	Retained
	SCFTDR2	Undefined	Undefined	Retained	Undefined	Retained
	SCFSR2	H'0060	H'0060	Retained	H'0060	Retained
	SCFRDR2	Undefined	Undefined	Retained	Undefined	Retained
	SCFCR2	H'0000	H'0000	Retained	H'0000	Retained
	SCTFDR2	H'0000	H'0000	Retained	H'0000	Retained
	SCRFDR2	H'0000	H'0000	Retained	H'0000	Retained
	SCSPTR2	H'0000*5	H'0000*5	Retained	H'0000*5	Retained
	SCLSR2	H'0000	H'0000	Retained	H'0000	Retained
	SCRER2	H'0000	H'0000	Retained	H'0000	Retained
SIM	SISMR	H'20	H'20	Retained	H'20	Retained
	SIBRR	H'07	H'07	Retained	H'07	Retained
	SISCR	H'00	H'00	Retained	H'00	Retained
	SITDR	H'FF	H'FF	Retained	H'FF	Retained
	SISSR	H'84	H'84	Retained	H'84	Retained

Module	Abbrev.	by RESET Pin/WDT/ H-UDI	RESET Pin/WDT/ by Sleep Multiple Exception	Instruction/ Deep Sleep	by Hardware	by Software/ Each Module
SIM	SIRDR	H'00	H'00	Retained	H'00	Retained
	SISCMR	H'01	H'01	Retained	H'01	Retained
	SISC2R	H'00	H'00	Retained	H'00	Retained
	SIWAIT	H'0000	H'0000	Retained	H'0000	Retained
	SIGRD	H'00	H'00	Retained	H'00	Retained
	SISMPL	H'0173	H'0173	Retained	H'0173	Retained
I^2C Channel 0	ICSCR0	H'0000 0000	H'0000 0000	Retained	H'0000 0000	Retained
	ICMCR0	H'0000 0000*6	H'0000 0000*6	Retained	H'0000 0000*6	Retained
	ICSSR0	H'0000 0000	H'0000 0000	Retained	H'0000 0000	Retained
	ICMSR0	H'0000 0000	H'0000 0000	Retained	H'0000 0000	Retained
	ICSIER0	H'0000 0000	H'0000 0000	Retained	H'0000 0000	Retained
	ICMIER0	H'0000 0000	H'0000 0000	Retained	H'0000 0000	Retained
	ICCCR0	H'0000 0000	H'0000 0000	Retained	H'0000 0000	Retained
	ICSAR0	H'0000 0000	H'0000 0000	Retained	H'0000 0000	Retained
	ICMAR0	H'0000 0000	H'0000 0000	Retained	H'0000 0000	Retained
	ICRXD0	Undefined	Undefined	Retained	Undefined	Retained
	ICTXD0	Undefined	Undefined	Retained	Undefined	Retained
	ICFCR0	H'0000 0000	H'0000 0000	Retained	H'0000 0000	Retained
	ICFSR0	H'0000 0001	H'0000 0001	Retained	H'0000 0001	Retained
	ICFIER0	H'0000 0000	H'0000 0000	Retained	H'0000 0000	Retained
	ICRFDR0	H'0000 0000	H'0000 0000	Retained	H'0000 0000	Retained
	ICTFDR0	H'0000 0000	H'0000 0000	Retained	H'0000 0000	Retained
I^2C Channel 1	ICSCR1	H'0000 0000	H'0000 0000	Retained	H'0000 0000	Retained
	ICMCR1	H'0000 0000*6	H'0000 0000*6	Retained	H'0000 0000*6	Retained
	ICSSR1	H'0000 0000	H'0000 0000	Retained	H'0000 0000	Retained
	ICMSR1	H'0000 0000	H'0000 0000	Retained	H'0000 0000	Retained
	ICSIER1	H'0000 0000	H'0000 0000	Retained	H'0000 0000	Retained
	ICMIER1	H'0000 0000	H'0000 0000	Retained	H'0000 0000	Retained
	ICCCR1	H'0000 0000	H'0000 0000	Retained	H'0000 0000	Retained
	ICSAR1	H'0000 0000	H'0000 0000	Retained	H'0000 0000	Retained
	ICMAR1	H'0000 0000	H'0000 0000	Retained	H'0000 0000	Retained

Module	Abbrev.	by RESET Pin/WDT/ H-UDI	RESET Pin/WDT/by Sleep Multiple Exception	Instruction/ Deep Sleep	by Hardware	by Software/ Each Module
I²C Channel 1	ICRXD1	Undefined	Undefined	Retained	Undefined	Retained
	ICTXD1	Undefined	Undefined	Retained	Undefined	Retained
	ICFCR1	H'0000 0000	H'0000 0000	Retained	H'0000 0000	Retained
	ICFSR1	H'0000 0001	H'0000 0001	Retained	H'0000 0001	Retained
	ICFIER1	H'0000 0000	H'0000 0000	Retained	H'0000 0000	Retained
	ICRFDR1	H'0000 0000	H'0000 0000	Retained	H'0000 0000	Retained
	ICTFDR1	H'0000 0000	H'0000 0000	Retained	H'0000 0000	Retained
SSI Channel 0	SSICR0	H'0000 0000	H'0000 0000	Retained	H'0000 0000	Retained
	SSISR0	H'0200 0003	H'0200 0003	Retained	H'0200 0003	Retained
	SSITDR0	H'0000 0000	H'0000 0000	Retained	H'0000 0000	Retained
	SSIRDR0	H'0000 0000	H'0000 0000	Retained	H'0000 0000	Retained
SSI Channel 1	SSICR1	H'0000 0000	H'0000 0000	Retained	H'0000 0000	Retained
	SSISR1	H'0200 0003	H'0200 0003	Retained	H'0200 0003	Retained
	SSITDR1	H'0000 0000	H'0000 0000	Retained	H'0000 0000	Retained
	SSIRDR1	H'0000 0000	H'0000 0000	Retained	H'0000 0000	Retained
USB	HcRevision	H'0000 0010	H'0000 0010	Retained	H'0000 0010	Retained
	HcControl	H'0000 0000	H'0000 0000	Retained	H'0000 0000	Retained
	HcCommandStatus	H'0000 0000	H'0000 0000	Retained	H'0000 0000	Retained
	HcInterruptStatus	H'0000 0000	H'0000 0000	Retained	H'0000 0000	Retained
	HcInterruptEnable	H'0000 0000	H'0000 0000	Retained	H'0000 0000	Retained
	HcInterruptDisable	H'0000 0000	H'0000 0000	Retained	H'0000 0000	Retained
	HcHCCA	H'0000 0000	H'0000 0000	Retained	H'0000 0000	Retained
	HcPeriodCurrentED	H'0000 0000	H'0000 0000	Retained	H'0000 0000	Retained
	HcControlHeadED	H'0000 0000	H'0000 0000	Retained	H'0000 0000	Retained
	HcControlCurrentED	H'0000 0000	H'0000 0000	Retained	H'0000 0000	Retained
	HcBulkHeadED	H'0000 0000	H'0000 0000	Retained	H'0000 0000	Retained
	HcBulkCurrentED	H'0000 0000	H'0000 0000	Retained	H'0000 0000	Retained
	HcDoneHead	H'0000 0000	H'0000 0000	Retained	H'0000 0000	Retained
	HcFmInterval	H'0000 2EDF	H'0000 2EDF	Retained	H'0000 2EDF	Retained
	HcFmRemaining	H'0000 0000	H'0000 0000	Retained	H'0000 0000	Retained
	HcFmNumber	H'0000 0000	H'0000 0000	Retained	H'0000 0000	Retained

Module	Abbrev.	by RESET Pin/WDT/ H-UDI	RESET Pin/WDT/ Multiple Exception	by Sleep Instruction/ Deep Sleep	by Hardware	by Software/ Each Module
USB	HcPeriodicStart	H'0000 0000	H'0000 0000	Retained	H'0000 0000	Retained
	HcLSThreshold	H'0000 0628	H'0000 0628	Retained	H'0000 0628	Retained
	HcRhDescriptorA	H'0200 1202	H'0200 1202	Retained	H'0200 1202	Retained
	HcRhDescriptorB	H'0000 0000	H'0000 0000	Retained	H'0000 0000	Retained
	HcRhStatus	H'0000 0000	H'0000 0000	Retained	H'0000 0000	Retained
	HcRhPortStatus1	H'0000 0100	H'0000 0100	Retained	H'0000 0100	Retained
	Shared memory Start	Undefined	Undefined	Retained	Undefined	Retained
	Shared memory End	Undefined	Undefined	Retained	Undefined	Retained
HCAN2 Channel 0	CAN0MCR	H'0001	H'0001	Retained	H'0001	Retained
	CAN0GSR	H'000C	H'000C	Retained	H'000C	Retained
	CAN0BCR1	H'0000	H'0000	Retained	H'0000	Retained
	CAN0BCR0	H'0000	H'0000	Retained	H'0000	Retained
	CAN0IRR	H'0001	H'0001	Retained	H'0001	Retained
	CAN0IMR	H'FFFF	H'FFFF	Retained	H'FFFF	Retained
	CAN0TECREC	H'0000	H'0000	Retained	H'0000	Retained
	CAN0TXPR1	H'0000	H'0000	Retained	H'0000	Retained
	CAN0TXPR0	H'0000	H'0000	Retained	H'0000	Retained
	CAN0TXCR1	H'0000	H'0000	Retained	H'0000	Retained
	CAN0TXCR0	H'0000	H'0000	Retained	H'0000	Retained
	CAN0TXACK1	H'0000	H'0000	Retained	H'0000	Retained
	CAN0TXACK0	H'0000	H'0000	Retained	H'0000	Retained
	CAN0ABACK1	H'0000	H'0000	Retained	H'0000	Retained
	CAN0ABACK0	H'0000	H'0000	Retained	H'0000	Retained
	CAN0RXPR1	H'0000	H'0000	Retained	H'0000	Retained
	CAN0RXPR0	H'0000	H'0000	Retained	H'0000	Retained
	CAN0RFPR1	H'0000	H'0000	Retained	H'0000	Retained
	CAN0RFPR0	H'0000	H'0000	Retained	H'0000	Retained
	CAN0MBIMR1	H'FFFF	H'FFFF	Retained	H'FFFF	Retained
	CAN0MBIMR0	H'FFFF	H'FFFF	Retained	H'FFFF	Retained
	CAN0UMSR1	H'0000	H'0000	Retained	H'0000	Retained
	CAN0UMSR0	H'0000	H'0000	Retained	H'0000	Retained

Module	Abbrev.	by RESET Pin/WDT/ H-UDI	by RESET Pin/WDT/ Multiple Exception	by Sleep Instruction/ Deep Sleep	by Hardware	by Software/ Each Module
HCAN2 Channel 0	CAN0TCNTR	H'0000	H'0000	Retained	H'0000	Retained
	CAN0TCR	H'0000	H'0000	Retained	H'0000	Retained
	CAN0TCMR	H'0000	H'0000	Retained	H'0000	Retained
	CAN0MB0	Undefined	Undefined	Retained	Undefined	Retained
	CAN0MB1	Undefined	Undefined	Retained	Undefined	Retained
	CAN0MB2	Undefined	Undefined	Retained	Undefined	Retained
	CAN0MB3	Undefined	Undefined	Retained	Undefined	Retained
	CAN0MB4	Undefined	Undefined	Retained	Undefined	Retained
	CAN0MB5	Undefined	Undefined	Retained	Undefined	Retained
	CAN0MB6	Undefined	Undefined	Retained	Undefined	Retained
	CAN0MB7	Undefined	Undefined	Retained	Undefined	Retained
	CAN0MB8	Undefined	Undefined	Retained	Undefined	Retained
	CAN0MB9	Undefined	Undefined	Retained	Undefined	Retained
	CAN0MB10	Undefined	Undefined	Retained	Undefined	Retained
	CAN0MB11	Undefined	Undefined	Retained	Undefined	Retained
	CAN0MB12	Undefined	Undefined	Retained	Undefined	Retained
	CAN0MB13	Undefined	Undefined	Retained	Undefined	Retained
	CAN0MB14	Undefined	Undefined	Retained	Undefined	Retained
	CAN0MB15	Undefined	Undefined	Retained	Undefined	Retained
	CAN0MB16	Undefined	Undefined	Retained	Undefined	Retained
	CAN0MB17	Undefined	Undefined	Retained	Undefined	Retained
	CAN0MB18	Undefined	Undefined	Retained	Undefined	Retained
	CAN0MB19	Undefined	Undefined	Retained	Undefined	Retained
	CAN0MB20	Undefined	Undefined	Retained	Undefined	Retained
	CAN0MB21	Undefined	Undefined	Retained	Undefined	Retained
	CAN0MB22	Undefined	Undefined	Retained	Undefined	Retained
	CAN0MB23	Undefined	Undefined	Retained	Undefined	Retained
	CAN0MB24	Undefined	Undefined	Retained	Undefined	Retained
	CAN0MB25	Undefined	Undefined	Retained	Undefined	Retained
	CAN0MB26	Undefined	Undefined	Retained	Undefined	Retained
	CAN0MB27	Undefined	Undefined	Retained	Undefined	Retained

Module	Abbrev.	by RESET Pin/WDT/ H-UDI	RESET Pin/WDT/ Multiple Exception	by Sleep Instruction/ Deep Sleep	by Hardware	by Software/ Each Module
HCAN2 Channel 0	CAN0MB28	Undefined	Undefined	Retained	Undefined	Retained
	CAN0MB29	Undefined	Undefined	Retained	Undefined	Retained
	CAN0MB30	Undefined	Undefined	Retained	Undefined	Retained
	CAN0MB31	Undefined	Undefined	Retained	Undefined	Retained
HCAN2 Channel 1	CAN1MCR	H'0001	H'0001	Retained	H'0001	Retained
	CAN1GSR	H'000C	H'000C	Retained	H'000C	Retained
	CAN1BCR1	H'0000	H'0000	Retained	H'0000	Retained
	CAN1BCR0	H'0000	H'0000	Retained	H'0000	Retained
	CAN1IRR	H'0001	H'0001	Retained	H'0001	Retained
	CAN1IMR	H'FFFF	H'FFFF	Retained	H'FFFF	Retained
	CAN1TECREC	H'0000	H'0000	Retained	H'0000	Retained
	CAN1TXPR1	H'0000	H'0000	Retained	H'0000	Retained
	CAN1TXPR0	H'0000	H'0000	Retained	H'0000	Retained
	CAN1TXCR1	H'0000	H'0000	Retained	H'0000	Retained
	CAN1TXCR0	H'0000	H'0000	Retained	H'0000	Retained
	CAN1TXACK1	H'0000	H'0000	Retained	H'0000	Retained
	CAN1TXACK0	H'0000	H'0000	Retained	H'0000	Retained
	CAN1ABACK1	H'0000	H'0000	Retained	H'0000	Retained
	CAN1ABACK0	H'0000	H'0000	Retained	H'0000	Retained
	CAN1RXPR1	H'0000	H'0000	Retained	H'0000	Retained
	CAN1RXPR0	H'0000	H'0000	Retained	H'0000	Retained
	CAN1RFPR1	H'0000	H'0000	Retained	H'0000	Retained
	CAN1RFPR0	H'0000	H'0000	Retained	H'0000	Retained
	CAN1MBIMR1	H'FFFF	H'FFFF	Retained	H'FFFF	Retained
	CAN1MBIMR0	H'FFFF	H'FFFF	Retained	H'FFFF	Retained
	CAN1UMSR1	H'0000	H'0000	Retained	H'0000	Retained
	CAN1UMSR0	H'0000	H'0000	Retained	H'0000	Retained
	CAN1TCNTR	H'0000	H'0000	Retained	H'0000	Retained
	CAN1TCR	H'0000	H'0000	Retained	H'0000	Retained
	CAN1TCMR	H'0000	H'0000	Retained	H'0000	Retained

Module	Abbrev.	by RESET Pin/WDT/ H-UDI	by RESET Pin/WDT/ Multiple Exception	by Sleep Instruction/ Deep Sleep	by Hardware	by Software/ Each Module
HCAN2 Channel 1	CAN1MB0	Undefined	Undefined	Retained	Undefined	Retained
	CAN1MB1	Undefined	Undefined	Retained	Undefined	Retained
	CAN1MB2	Undefined	Undefined	Retained	Undefined	Retained
	CAN1MB3	Undefined	Undefined	Retained	Undefined	Retained
	CAN1MB4	Undefined	Undefined	Retained	Undefined	Retained
	CAN1MB5	Undefined	Undefined	Retained	Undefined	Retained
	CAN1MB6	Undefined	Undefined	Retained	Undefined	Retained
	CAN1MB7	Undefined	Undefined	Retained	Undefined	Retained
	CAN1MB8	Undefined	Undefined	Retained	Undefined	Retained
	CAN1MB9	Undefined	Undefined	Retained	Undefined	Retained
	CAN1MB10	Undefined	Undefined	Retained	Undefined	Retained
	CAN1MB11	Undefined	Undefined	Retained	Undefined	Retained
	CAN1MB12	Undefined	Undefined	Retained	Undefined	Retained
	CAN1MB13	Undefined	Undefined	Retained	Undefined	Retained
	CAN1MB14	Undefined	Undefined	Retained	Undefined	Retained
	CAN1MB15	Undefined	Undefined	Retained	Undefined	Retained
	CAN1MB16	Undefined	Undefined	Retained	Undefined	Retained
	CAN1MB17	Undefined	Undefined	Retained	Undefined	Retained
	CAN1MB18	Undefined	Undefined	Retained	Undefined	Retained
	CAN1MB19	Undefined	Undefined	Retained	Undefined	Retained
	CAN1MB20	Undefined	Undefined	Retained	Undefined	Retained
	CAN1MB21	Undefined	Undefined	Retained	Undefined	Retained
	CAN1MB22	Undefined	Undefined	Retained	Undefined	Retained
	CAN1MB23	Undefined	Undefined	Retained	Undefined	Retained
	CAN1MB24	Undefined	Undefined	Retained	Undefined	Retained
	CAN1MB25	Undefined	Undefined	Retained	Undefined	Retained
	CAN1MB26	Undefined	Undefined	Retained	Undefined	Retained
	CAN1MB27	Undefined	Undefined	Retained	Undefined	Retained
	CAN1MB28	Undefined	Undefined	Retained	Undefined	Retained
	CAN1MB29	Undefined	Undefined	Retained	Undefined	Retained
	CAN1MB30	Undefined	Undefined	Retained	Undefined	Retained
	CAN1MB31	Undefined	Undefined	Retained	Undefined	Retained

Module	Abbrev.	by RESET Pin/WDT/ H-UDI	RESET Pin/WDT/ Multiple Exception	by Sleep Instruction/ Deep Sleep	by Hardware	by Software/ Each Module
HSPI	SPCR	H'0000 0000*7	H'0000 0000*7	Retained	H'0000 0000*7	Retained
	SPSR	H'0000 0120*7	H'0000 0120*7	Retained	H'0000 0120*7	Retained
	SPSCR	H'0000 0040*7	H'0000 0040*7	Retained	H'0000 0040*7	Retained
	SPTBR	H'0000 0000*7	H'0000 0000*7	Retained	H'0000 0000*7	Retained
	SPRBR	H'0000 0000*7	H'0000 0000*7	Retained	H'0000 0000*7	Retained
PFC	INPUPA	H'FF00	Retained	Retained	H'FF00	Retained
	DMAPCR	H'A550	Retained	Retained	H'A550	Retained
	SCIHZR	H'0000	Retained	Retained	H'0000	Retained
	IPSELR	H'0003	Retained	Retained	H'0003	Retained
	PAPUPR	H'FC	Retained	Retained	H'FC	Retained
	PBPUPR	H'FE	Retained	Retained	H'FE	Retained
	PCPUPR	H'FF	Retained	Retained	H'FF	Retained
	PDPUPR	H'FF	Retained	Retained	H'FF	Retained
	PEPUPR	H'FF	Retained	Retained	H'FF	Retained
	PFPUPR	H'0F	Retained	Retained	H'0F	Retained
	PGPUPR	H'FF	Retained	Retained	H'FF	Retained
	PHPUPR	H'FF	Retained	Retained	H'FF	Retained
	PJPUPR	H'FC	Retained	Retained	H'FC	Retained
	PKPUPR	H'FC	Retained	Retained	H'FC	Retained
	MDPUPR	H'38	Retained	Retained	H'38	Retained
	MODSELR	H'00	Retained	Retained	H'00	Retained
	PACR	H'0000	Retained	Retained	H'0000	Retained
	PBCR	H'0000	Retained	Retained	H'0000	Retained
	PCCR	H'FFFF	Retained	Retained	H'FFFF	Retained
	PDCR	H'FFFF	Retained	Retained	H'FFFF	Retained
	PECR	H'0000	Retained	Retained	H'0000	Retained
	PFCR	H'0000	Retained	Retained	H'0000	Retained
	PGCR	H'0000	Retained	Retained	H'0000	Retained
	PHCR	H'003C	Retained	Retained	H'003C	Retained
	PJCR	H'0000	Retained	Retained	H'0000	Retained
	PKCR	H'0000	Retained	Retained	H'0000	Retained

Module	Abbrev.	by RESET Pin/WDT/ H-UDI	RESET Pin/WDT/by Sleep Multiple Exception	Instruction/ Deep Sleep	by Hardware	by Software/ Each Module
PFC	PADR	H'00	Retained	Retained	H'00	Retained
	PBDR	H'00	Retained	Retained	H'00	Retained
	PCDR	H'00	Retained	Retained	H'00	Retained
	PDDR	H'00	Retained	Retained	H'00	Retained
	PEDR	H'00	Retained	Retained	H'00	Retained
	PFDR	H'00	Retained	Retained	H'00	Retained
	PGDR	H'00	Retained	Retained	H'00	Retained
	PHDR	H'00	Retained	Retained	H'00	Retained
	PJDR	H'00	Retained	Retained	H'00	Retained
	PKDR	H'00	Retained	Retained	H'00	Retained
	GPIOIC	H'0000	Retained	Retained	H'0000	Retained
HAC Channel 0	HACCR0	H'0000 0200	H'0000 0200	Retained	H'0000 0200	Retained
	HACCSAR0	H'0000 0000	H'0000 0000	Retained	H'0000 0000	Retained
	HACCSDR0	H'0000 0000	H'0000 0000	Retained	H'0000 0000	Retained
	HACPCML0	H'0000 0000	H'0000 0000	Retained	H'0000 0000	Retained
	HACPCMR0	H'0000 0000	H'0000 0000	Retained	H'0000 0000	Retained
	HACTIER0	H'0000 0000	H'0000 0000	Retained	H'0000 0000	Retained
	HACTSR0	H'0000 0000	H'0000 0000	Retained	H'0000 0000	Retained
	HACRIER0	H'0000 0000	H'0000 0000	Retained	H'0000 0000	Retained
	HACRSR0	H'0000 0000	H'0000 0000	Retained	H'0000 0000	Retained
	HACACR0	H'8400 0000	H'8400 0000	Retained	H'8400 0000	Retained
HAC Channel 1	HACCR1	H'0000 0200	H'0000 0200	Retained	H'0000 0200	Retained
	HACCSAR1	H'0000 0000	H'0000 0000	Retained	H'0000 0000	Retained
	HACCSDR1	H'0000 0000	H'0000 0000	Retained	H'0000 0000	Retained
	HACPCML1	H'0000 0000	H'0000 0000	Retained	H'0000 0000	Retained
	HACPCMR1	H'0000 0000	H'0000 0000	Retained	H'0000 0000	Retained
	HACTIER1	H'0000 0000	H'0000 0000	Retained	H'0000 0000	Retained
	HACTSR1	H'0000 0000	H'0000 0000	Retained	H'0000 0000	Retained
	HACRIER1	H'0000 0000	H'0000 0000	Retained	H'0000 0000	Retained
	HACRSR1	H'0000 0000	H'0000 0000	Retained	H'0000 0000	Retained
	HACACR1	H'8400 0000	H'8400 0000	Retained	H'8400 0000	Retained

Module	Abbrev.	by RESET Pin/WDT/ H-UDI	RESET Pin/WDT/ by Sleep Multiple Exception	Instruction/ Deep Sleep	by Hardware	by Software/ Each Module
MMCIF	CMDR0	H'00	H'00	Retained	H'00	Retained
	CMDR1	H'00	H'00	Retained	H'00	Retained
	CMDR2	H'00	H'00	Retained	H'00	Retained
	CMDR3	H'00	H'00	Retained	H'00	Retained
	CMDR4	H'00	H'00	Retained	H'00	Retained
	CMDR5	H'00	H'00	Retained	H'00	Retained
	CMDSTRT	H'00	H'00	Retained	H'00	Retained
	OPCR	H'00	H'00	Retained	H'00	Retained
	CSTR	H'0x	H'0x	Retained	H'0x	Retained
	INTCR0	H'00	H'00	Retained	H'00	Retained
	INTCR1	H'00	H'00	Retained	H'00	Retained
	INTSTR0	H'00	H'00	Retained	H'00	Retained
	INTSTR1	H'00	H'00	Retained	H'00	Retained
	CLKON	H'00	H'00	Retained	H'00	Retained
	CTOCR	H'00	H'00	Retained	H'00	Retained
	TBCR	H'00	H'00	Retained	H'00	Retained
	MODER	H'00	H'00	Retained	H'00	Retained
	CMDTYR	H'00	H'00	Retained	H'00	Retained
	RSPTYR	H'00	H'00	Retained	H'00	Retained
	RSPR0	H'00	H'00	Retained	H'00	Retained
	RSPR1	H'00	H'00	Retained	H'00	Retained
	RSPR2	H'00	H'00	Retained	H'00	Retained
	RSPR3	H'00	H'00	Retained	H'00	Retained
	RSPR4	H'00	H'00	Retained	H'00	Retained
	RSPR5	H'00	H'00	Retained	H'00	Retained
	RSPR6	H'00	H'00	Retained	H'00	Retained
	RSPR7	H'00	H'00	Retained	H'00	Retained
	RSPR8	H'00	H'00	Retained	H'00	Retained
	RSPR9	H'00	H'00	Retained	H'00	Retained
	RSPR10	H'00	H'00	Retained	H'00	Retained
	RSPR11	H'00	H'00	Retained	H'00	Retained

Module	Abbrev.	by RESET Pin/WDT/ H-UDI	RESET Pin/WDT/ by Sleep Multiple Exception	Instruction/ Deep Sleep	by Hardware	by Software/ Each Module
MMCIF	RSPR12	H'00	H'00	Retained	H'00	Retained
	RSPR13	H'00	H'00	Retained	H'00	Retained
	RSPR14	H'00	H'00	Retained	H'00	Retained
	RSPR15	H'00	H'00	Retained	H'00	Retained
	RSPR16	H'00	H'00	Retained	H'00	Retained
	DTOUTR	H'FFFF	H'FFFF	Retained	H'FFFF	Retained
	DR	H'xxxx	H'xxxx	Retained	H'xxxx	Retained
	FIFOCLR	H'00	H'00	Retained	H'00	Retained
	DMACR	H'00	H'00	Retained	H'00	Retained
	INTCR2	H'00	H'00	Retained	H'00	Retained
	INTSTR2	H'0x	H'0x	Retained	H'0x	Retained
	RDTIMSEL	H'00	H'00	Retained	H'00	Retained
MFI	MFIIDX	H'0000	H'0000	Retained	H'0000	Retained
	MFIGSR	H'0000	H'0000	Retained	H'0000	Retained
	MFISCR	H'0040/H'0050*8	H'0040/H'0050*8	Retained	H'0040/H'0050*8	Retained
	MFIMCR	H'0000	H'0000	Retained	H'0000	Retained
	MFIIICR	H'0000	H'0000	Retained	H'0000	Retained
	MFIEICR	H'0000	H'0000	Retained	H'0000	Retained
	MFIADR	H'0000	H'0000	Retained	H'0000	Retained
	MFIDATA	H'0000	H'0000	Retained	H'0000	Retained
	MFRAM Start	Undefined	Undefined	Retained	Undefined	Retained
	MFRAM End	Undefined	Undefined	Retained	Undefined	Retained
H-UDI	SDIR	H'FFFF*12	Retained	Retained	H'FFFF*12	Retained
	SDDR/SDDRH	Undefined	Retained	Retained	Undefined	Retained
	SDDRL	Undefined	Retained	Retained	Undefined	Retained
	SDINT	H'0000	Retained	Retained	H'0000	Retained
ADC	ADDRA	H'0000	H'0000	Retained	H'0000	H'0000*9
	ADDRB	H'0000	H'0000	Retained	H'0000	H'0000*9
	ADDRC	H'0000	H'0000	Retained	H'0000	H'0000*9
	ADDRD	H'0000	H'0000	Retained	H'0000	H'0000*9
	ADCSR	H'0040	H'0040	Retained	H'0040	H'0040*9

Module	Abbrev.	by RESET Pin/WDT/ H-UDI	RESET Pin/WDT/ Multiple Exception	by Sleep Instruction/ Deep Sleep	by Hardware	by Software/ Each Module
LCDC	LDICKR	H'0101	H'0101	Retained	H'0101	Retained
	LDMTR	H'0109	H'0109	Retained	H'0109	Retained
	LDDFR	H'000C	H'000C	Retained	H'000C	Retained
	LDSMR	H'0000	H'0000	Retained	H'0000	Retained
	LDSARU	H'0C00 0000	H'0C00 0000	Retained	H'0C00 0000	Retained
	LDSARL	H'0C00 0000	H'0C00 0000	Retained	H'0C00 0000	Retained
	LDLAOR	H'0280	H'0280	Retained	H'0280	Retained
	LDPALCR	H'0000	H'0000	Retained	H'0000	Retained
	LDPR00 to FF*10	Undefined	Undefined	Retained	Undefined	Retained
	LDHCNR	H'4F52	H'4F52	Retained	H'4F52	Retained
	LDHSYNR	H'0050	H'0050	Retained	H'0050	Retained
	LDVDLNR	H'01DF	H'01DF	Retained	H'01DF	Retained
	LDVTLNR	H'01DF	H'01DF	Retained	H'01DF	Retained
	LDVSYNR	H'01DF	H'01DF	Retained	H'01DF	Retained
	LDACLNR	H'000C	H'000C	Retained	H'000C	Retained
	LDINTR	H'0000	H'0000	Retained	H'0000	Retained
	LDPMMR	H'0010	H'0010	Retained	H'0010	Retained
	LDPSPR	H'F60F	H'F60F	Retained	H'F60F	Retained
	LDCNTR	H'0000	H'0000	Retained	H'0000	Retained
UBC	BARA	Undefined	Retained	Retained	Undefined	Retained
	BASRA	Undefined	Retained	Retained	Undefined	Retained
	BAMRA	Undefined	Retained	Retained	Undefined	Retained
	BBRA	H'0000	Retained	Retained	H'0000	Retained
	BARB	Undefined	Retained	Retained	Undefined	Retained
	BASRB	Undefined	Retained	Retained	Undefined	Retained
	BAMRB	Undefined	Retained	Retained	Undefined	Retained
	BBRB	H'0000	Retained	Retained	H'0000	Retained
	BDRB	Undefined	Retained	Retained	Undefined	Retained
	BDMRB	Undefined	Retained	Retained	Undefined	Retained
	BRCR	H'0000*11	Retained	Retained	H'0000*11	Retained

3. The initial values of bits 11 to 9 are 1 and those of bits 8 to 0 are undefined when the $\overline{\text{RESET}}$ pin resets all of them. The values of bits 11 to 0 are retained when the WDT or the H-UDI resets all of them.
4. Bits 2 and 0 are undefined.
5. Bits 6, 4, 2, and 0 are undefined.
6. Bits 6 and 5 are undefined.
7. Reserved bits are read as undefined values.
8. 80-series interface: 0040; 68-series interface: 0050
9. Before entering module standby or software standby mode, check that A/D conversion is not in progress. If standby mode is entered while A/D conversion is in progress, correct register values are not guaranteed.
10. There are 256 LDPRxx: LDPR00, LDPR01, ..., LDPRFF.
11. Bits 10, 7, 6, and 3 are undefined.
12. Reserved bit are read as undefined values, for details, see each of the register descriptions.

33.1 Absolute Maximum Ratings

Table 33.1 Absolute Maximum Ratings

Item	Symbol	Value	Unit
I/O, CPG, ADC power supply voltage	V_{DDQ} V_{DD-CPG} AV_{CC-ADC}	−0.3 to 4.6	V
Internal power supply voltage	V_{DD} $V_{DD-PLL1/2/3}$	−0.3 to 2.1	V
Input voltage	V_{in}	−0.3 to V_{DDQ}+0.3, −0.5 to 5.5*1	V
Operating temperature*2	T_{opr}	−20 to 75/−40 to 85	°C
Storage temperature	T_{stg}	−55 to 125	°C

Notes: 1. The LSI may be permanently damaged if the maximum ratings are exceeded.
2. The LSI may be permanently damaged if any of the V_{SS} pins are not connected to GND.
3. For the powering-on and powering-off sequence, see Appendix F, Power-On and Power-Off Procedures.

*1 For I2C0_SCL, I2C0_SDA, I2C1_SCL, I2C1_SDA.

*2 See table G.1.

Table 33.2 DC Characteristics (T_a = –20 to 75°C/–40 to 85°C)

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Power supply voltage		V_{DDQ} V_{DD-CPG} AV_{CC-ADC}	3.0	3.3	3.6	V	Normal mode, sleep mode, deep-sleep mode, standby mode
		V_{DD} $V_{DD-PLL1/2/3}$	1.4	1.5	1.6		Normal mode, sleep mode, deep-sleep mode, standby mode
Analog power supply current	During A/D conversion	AI_{cc}	—	6.0	7.5	mA	
	During idle		—	0.2	2.0	µA	
Current dissipation	Normal operation	I_{DD}	—	220	730	mA	Ick = 200 MHz
	Sleep mode		—	90	160		
	Standby mode		—	—	250	µA	T_a = 25°C
			—	—	500		T_a > 50°C
	Normal operation	I_{DDQ}	—	70	190	mA	Ick = 200 MHz
	Sleep mode		—	25	50		Bck = 67 MHz
	Standby mode		—	—	530	µA	T_a = 25°C
			—	—	800		T_a > 50°C
Input voltage	$\overline{\text{RESET}}$, NMI, $\overline{\text{TRST}}$, $\overline{\text{ASEBRK}}$/BRKACK, $\overline{\text{MRESET}}$, CA, $\overline{\text{SCIF2\_RTS}}$, USB_PENC, VEPWC/$\overline{\text{IRQ5}}$, VCPWC/$\overline{\text{IRQ4}}$, $\overline{\text{IRL3}}$, $\overline{\text{IRL2}}$, $\overline{\text{IRL1}}$, $\overline{\text{IRL0}}$, Reserved/AUDATA[3]	V_{IH}	V_{DDQ} × 0.9	—	V_{DDQ} +0.3	V	
	I2C1_SCL, I2C1_SDA I2C0_SCL, I2C0_SDA		V_{DDQ} × 0.7	—	5.5		
	USB_DP, USB_DM		V_{DDQ} × 0.7	—	V_{DDQ} +0.3		
	Other input pins		2.2	—	V_{DDQ} +0.3		

Input voltage	RESET, NMI, TRST, ASEBRK/BRKACK, MRESET, CA, SCIF2_RTS, USB_PENC, VEPWC/IRQ5, VCPWC/IRQ4, IRL3, IRL2, IRL1, IRL0, Reserved/AUDATA[3]	V_{IL}	−0.3	—	V_{DDQ} × 0.1	V	
	I2C1_SCL, I2C1_SDA I2C0_SCL, I2C0_SDA		−0.5	—	V_{DDQ} × 0.3		
	USB_DP, USB_DM		−0.3	—	V_{DDQ} × 0.2		
	Other input pins		−0.3	—	V_{DDQ} × 0.2		
Input leak current	All input pins	|Iin|	—	—	1	µA	V_{IN} = 0.5 to V_{DDQ} −0.5V
Three-state leak current	I/O, all output pins (off condition)	|Isti|	—	—	1	µA	V_{IN} = 0.5 to V_{DDQ} −0.5V
Output voltage	All output pins*	V_{OH}	V_{DDQ} ×0.8	—	—	V	
	I2C1_SCL, I2C1_SDA I2C0_SCL, I2C0_SDA	V_{OL}	0	—	0.4		
	Other pins		—	—	V_{DDQ} × 0.2		
Pull-up resistance	All pins	R_{pull}	20	60	180	kΩ	
Pin capacitance	USB_DP, USB_DM	C_{USB}	—	—	20	pF	
	AN3, AN2, AN1, AN0	C_{ADC}	—	—	20		
	Other pins	C_L	—	—	10		

Notes: 1. Regardless of whether or not the PLL is used, please supply the same voltage to V_{DDQ}, AV_{CC_ADC}, $V_{DD\text{-}CPG}$, supply the same voltage $V_{DD\text{-}PLL\ 1/2/3}$ and V_{DD}, connect V_{SS}, $V_{SS\text{-}CPG}$, and $V_{DD\text{-}PLL\ 1/2/3}$ to GND. The LSI may be damage when not filling this.

2. Regardless of whether or not the A/D converter is used, connect V_{DDQ} and $V_{DD\text{-}CPG}$ to $AV_{CC\text{-}ADC}$, and $AV_{SS\text{-}ADC}$ to GND. The LSI may be damage when not filling this.
3. The current dissipation values are for V_{IH} min = V_{DDQ} −0.5 V and V_{IL} max = 0.5 V with all output pins unload.
4. I_{DD} is the sum total value of the current of V_{DD} and $V_{DD\text{-}PLL\ 1/2/3}$.
5. I_{DDQ} is the sum total value of the current of V_{DDQ} and $V_{DD\text{-}CPG}$.

* I2Cn_ SCL and I2Cn_SDA pins are removed.

Item	Symbol	Min.	Typ.	Max.	Unit
Permissible output low current (per pin)	I_{OL}	—	—	2	mA
Permissible output low current (total)	ΣI_{OL}	—	—	120	
Permissible output high current (per pin)	$-I_{OH}$	—	—	2	mA
Permissible output high current (total)	$\Sigma\lvert-I_{OH}\rvert$	—	—	40	

Note: To protect chip reliability, do not exceed the output current values in table 33.3.

33.3 AC Characteristics

In principle, this LSI's input should be synchronous. Unless specified otherwise, ensure that the setup time and hold times for each input signal are observed.

Table 33.4 Clock Timing

Item		Symbol	Min.	Typ.	Max.	Unit
Operating frequency	CPU, FPU, cache, TLB	f	1	—	200	MHz
	External bus		1	—	67	
	Peripheral modules		1	—	34	

Table 33.5 Clock and Control Signal Timing

(V_{DDQ} = 3.0 to 3.6 V, V_{DD} = 1.5 V, T_a = –20 to 75°C/–40 to 85°C, C_L = 30 pF, PLL2 on)

Item		Symbol	Min.	Max.	Unit	Figure
EXTAL clock input frequency	PLL1 6-times/PLL2 operation	f_{EX}	16	34	MHz	
	PLL1 12-times/PLL2 operation		16	22		
	PLL1/PLL2 not operating		1	34		
DCK clock output		f_{OP2}	22	67	MHz	
DCK clock output cycle time		t_{DCcyc}	15	45	ns	33.4
DCK clock output low-level pulse width		t_{DCOL1}	1	—	ns	33.4
DCK clock output high-level pulse width		t_{DCOH1}	1	—	ns	33.4
DCK clock output rise time		t_{DCOr}	—	3	ns	33.4
DCK clock output fall time		t_{DCOf}	—	3	ns	33.4
DCK clock output low-level pulse width		t_{DCOL2}	3	—	ns	33.5
DCK clock output high-level pulse width		t_{DCOH2}	3	—	ns	33.5
EXTAL clock input cycle time		t_{EXcyc}	30	1000	ns	33.1
EXTAL clock input low-level pulse width		t_{EXL}	3.5	—	ns	33.1
EXTAL clock input high-level pulse width		t_{EXH}	3.5	—	ns	33.1
EXTAL clock input rise time		t_{EXr}	—	4	ns	33.1
EXTAL clock input fall time		t_{EXf}	—	4	ns	33.1
CKIO clock output	PLL1/PLL2 operation	f_{OP}	25	67	MHz	
	PLL1/PLL2 not operating		1	34		
CKIO clock output cycle time		t_{CKOcyc}	15	1000	ns	33.2
CKIO clock output low-level pulse width		t_{CKOL1}	1	—	ns	33.2
CKIO clock output high-level pulse width		t_{CKOH1}	1	—	ns	33.2
CKIO clock output rise time		t_{CKOr}	—	3	ns	33.2
CKIO clock output fall time		t_{CKOf}	—	3	ns	33.2
CKIO clock output low-level pulse width		t_{CKOL2}	3	—	ns	33.3
CKIO clock output high-level pulse width		t_{CKOH2}	3	—	ns	33.3
Power-on oscillation settling time		t_{OSC1}	10	—	ms	33.6, 33.8
Power-on oscillation settling time/mode setting		t_{OSCMD}	10	—	ms	33.6, 33.8

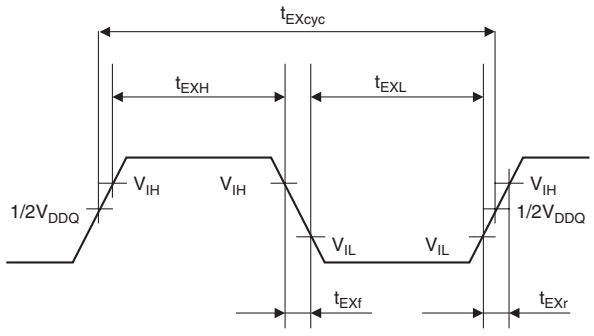
MD reset setup time	t_{MDRS}	3	—	t_{cyc}	33.14
MD reset hold time	t_{MDRH}	20	—	ns	33.6, 33.8
$\overline{\text{RESET}}$ assert time	t_{RESW}	20	—	t_{cyc}	33.6, 33.7, 33.8, 33.9
PLL synchronization settling time	t_{PLL}	200	—	µs	33.12, 33.13
Standby return oscillation settling time 1	t_{OSC2}	5	—	ms	33.7, 33.9
Standby return oscillation settling time 2	t_{OSC3}	5	—	ms	33.10
Standby return oscillation settling time 3	t_{OSC4}	5	—	ms	33.11
Standby return oscillation settling time 1*	t_{OSC2}	2	—	ms	33.7, 33.9
Standby return oscillation settling time 2*	t_{OSC3}	2	—	ms	33.10
Standby return oscillation settling time 3*	t_{OSC4}	2	—	ms	33.11
IRL interrupt determination time (standby mode)	t_{IRLSTB}	—	200	µs	33.13
$\overline{\text{TRST}}$ reset hold time	t_{TRSTRH}	0	—	ns	33.6, 33.8
$\overline{\text{RESET}}$ input rise time	t_{PRr}	—	1	µs	33.14
$\overline{\text{RESET}}$ input fall time	t_{PRf}	—	1	µs	33.14

Notes: 1. When a crystal resonator is connected to EXTAL and XTAL, the maximum frequency is 34MHz. when a 3rd overtone crystal resonator is used, an external tank circuit is necessary.

2. As there is feedback from the CKIO pin when PLL2 is operating, the load copacitance connected to the CKIO pin should be a maximum of 50 pF.

3. t_{cyc} shows 1 cycle time of a CKIO clock.

* When the oscillation settling time of the crystal resonator is 1ms or less.



Note: When the clock is input from the EXTAL pin.

Figure 33.1 EXTAL Clock Input Timing

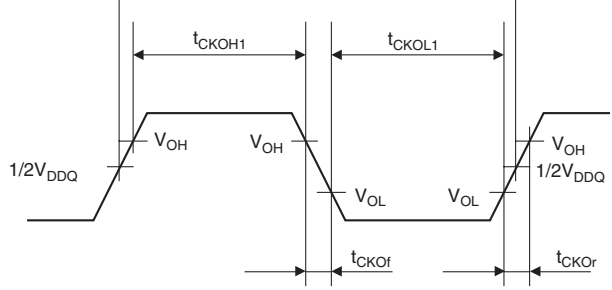


Figure 33.2 CKIO Clock Output Timing (1)

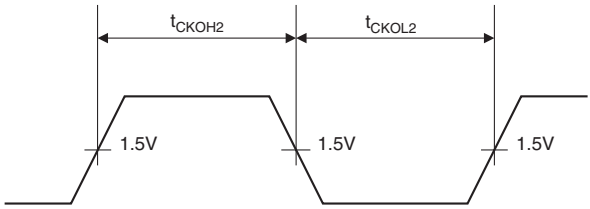


Figure 33.3 CKIO Clock Output Timing (2)

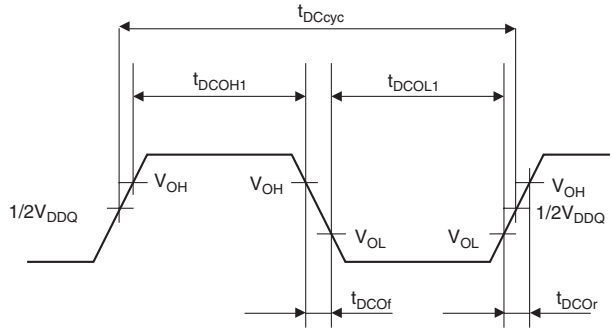


Figure 33.4 DCK Clock Output Timing (1)

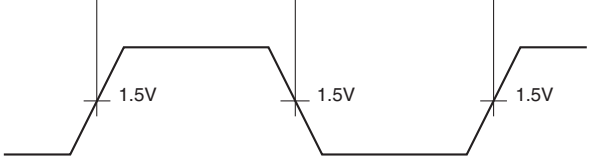
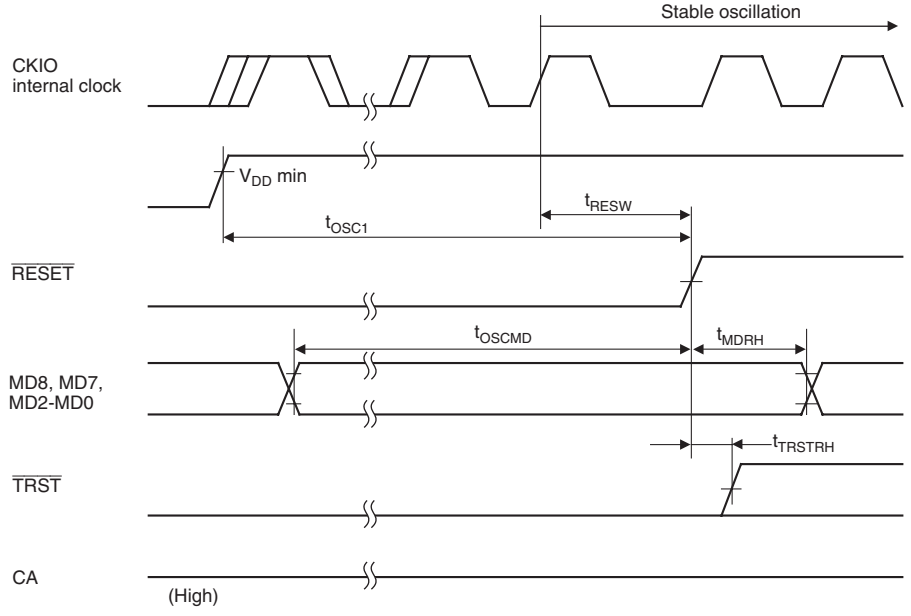
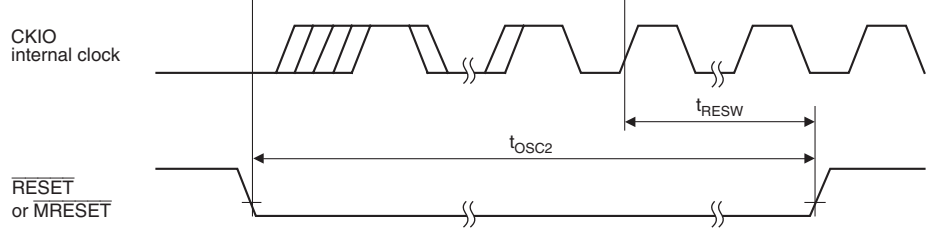


Figure 33.5 DCK Clock Output Timing (2)



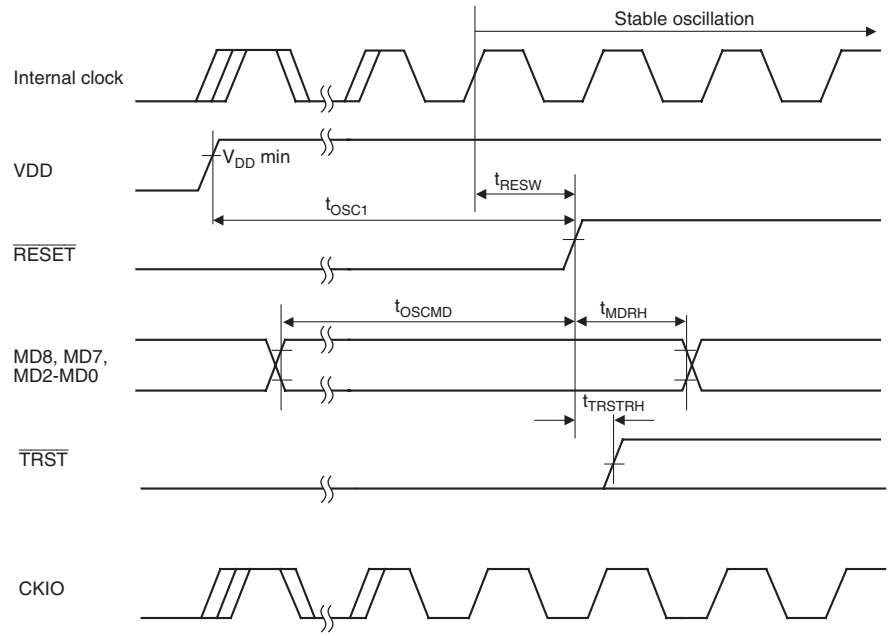
Notes: 1. Oscillation settling time when on-chip resonator is used
2. PLL2 not operating

Figure 33.6 Power-On Oscillation Settling Time (1)



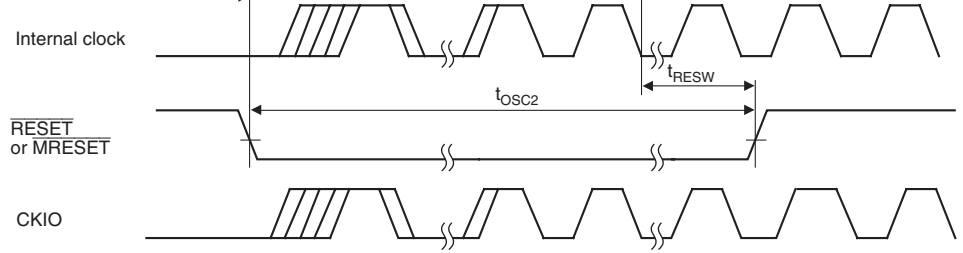
Notes: 1. Oscillation settling time when on-chip resonator is used.
2. PLL2 not operating

Figure 33.7 Standby Return Oscillation Settling Time (Return by RESET or MRESET) (1)



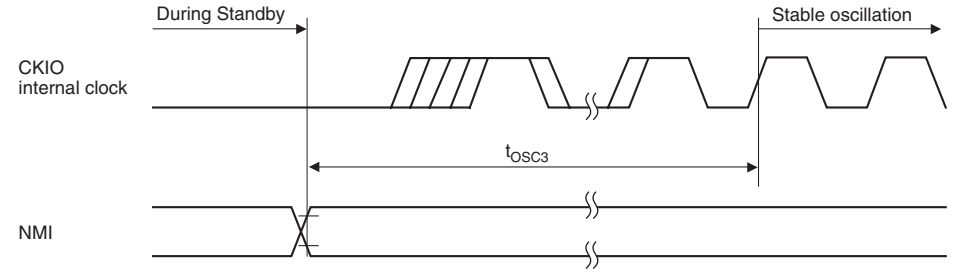
Notes: 1. Oscillation settling time when on-chip resonator is used.
2. PLL2 operating

Figure 33.8 Power-On Oscillation Settling Time (2)



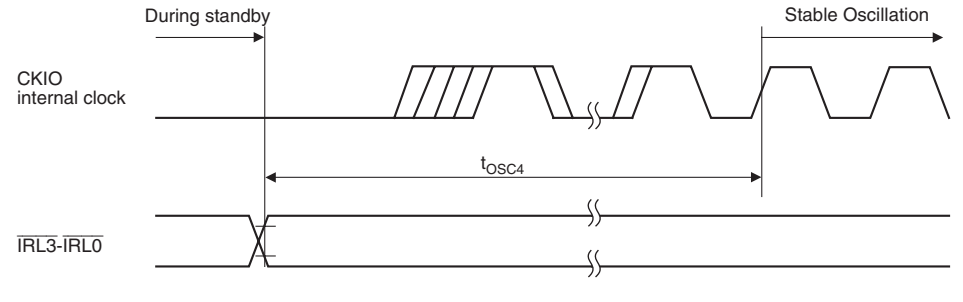
Notes: 1. Oscillation settling time when on-chip resonator is used.
2. PLL2 operating

Figure 33.9 Standby Return Oscillation Settling Time (Return by $\overline{\text{RESET}}$ or $\overline{\text{MRESET}}$) (2)



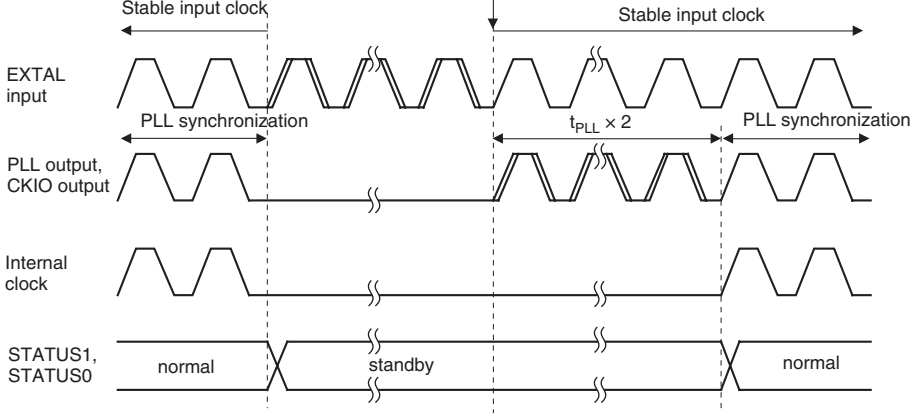
Note: Oscillation settling time when on-chip resonator is used.

Figure 33.10 Standby Return Oscillation Settling Time (Return by NMI)



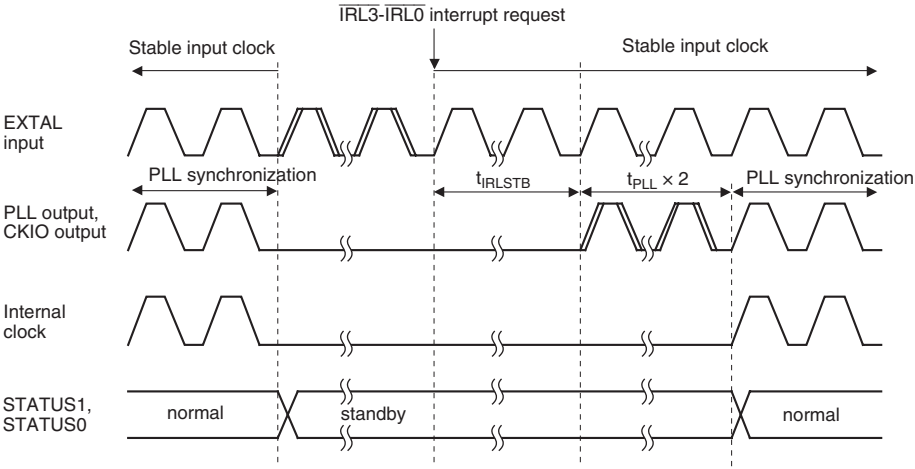
Note: Oscillation settling time when on-chip resonator is used.

Figure 33.11 Standby Return Oscillation Settling Time (Return by $\overline{\text{IRL3}}$ to $\overline{\text{IRL0}}$)



Note: When an external clock is input from EXTAL

Figure 33.12 PLL Synchronization Settling Time in Case of $\overline{\text{RESET}}$, $\overline{\text{MRESET}}$ or NMI Interrupt



Note: When an external clock is input from EXTAL

Figure 33.13 PLL Synchronization Settling Time in Case of IRL Interrupt

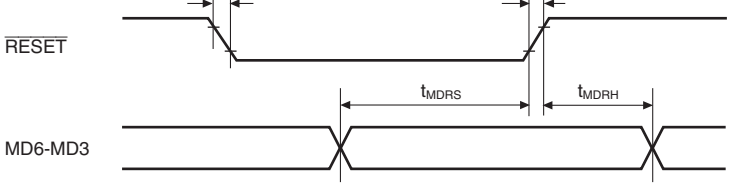


Figure 33.14 MD pins Setup/Hold Timing

33.3.2 Control Signal Timing

Table 33.6 Control Signal Timing

(V_{DDQ} = 3.0 to 3.6 V, V_{DD} = 1.5 V, T_a = –20 to 75°C/–40 to 85°C, C_L = 30 pF, PLL2 on)

Item	Symbol	Min.	Max.	Unit	Figure
$\overline{\text{BREQ}}$ setup time	t_{BREQS}	3	—	ns	33.15
$\overline{\text{BREQ}}$ hold time	t_{BREQH}	1.5	—	ns	33.15
$\overline{\text{BACK}}$ delay time	t_{BACKD}	—	6	ns	33.15
Bus tri-state delay time	t_{BOFF1}	—	12	ns	33.15
Bus tri-state delay time to standby mode	t_{BOFF2}	—	2	t_{cyc}	33.16 (2)
Bus buffer on time	t_{BON1}	—	12	ns	33.15
Bus buffer on time from standby	t_{BON2}	—	2	t_{cyc}	33.16 (2)
STATUS 0/1 delay time	t_{STD1}	—	6	ns	33.16 (1)
	t_{STD2}	—	2	t_{cyc}	33.16 (1) (2)
	t_{STD3}	—	2	t_{cyc}	33.16 (2)

Note: t_{cyc} shows 1 cycle time of a CKIO clock.

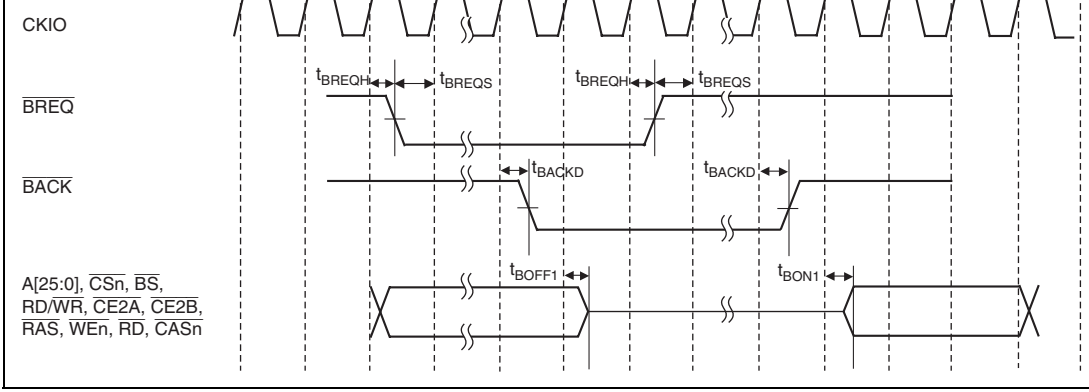


Figure 33.15 Control Signal Timing

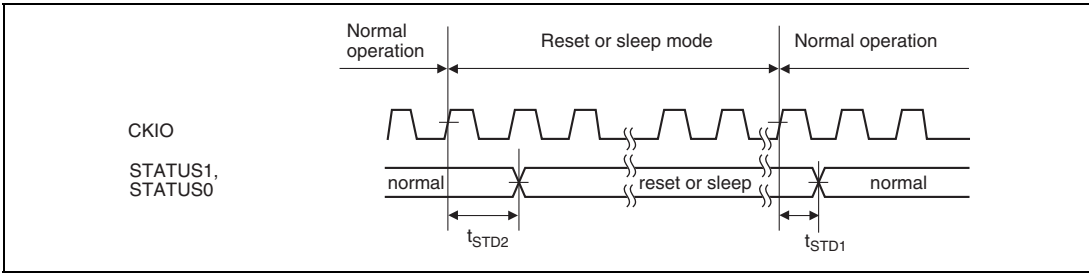
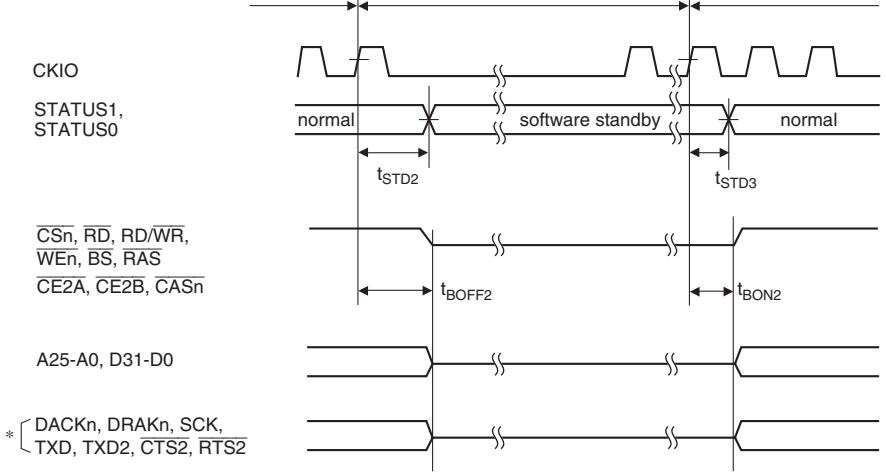


Figure 33.16 (1) Pin Drive Timing for Reset or Sleep Mode



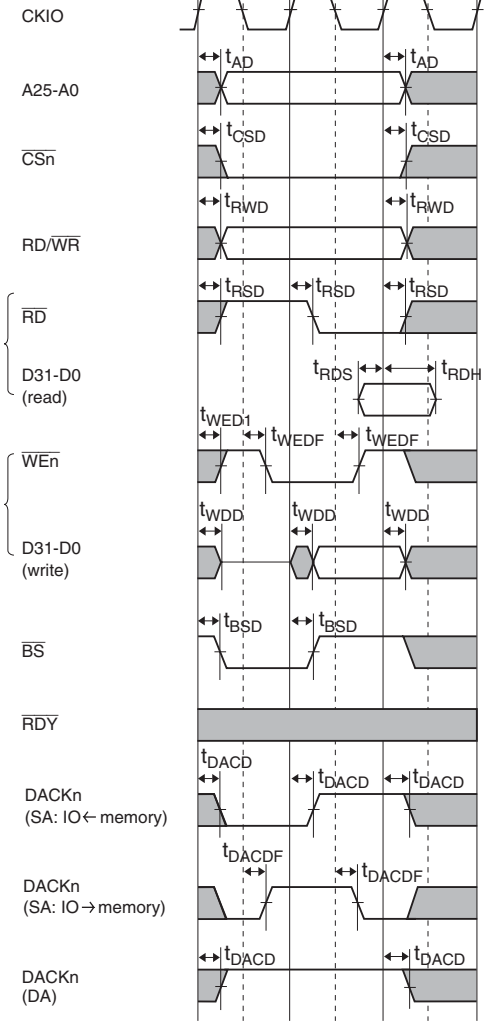
Note: * These pins can be put into state of high-impedance with SCIHZR and DMAPCR.

Figure 33.16 (2) Pin Drive Timing for Software Standby Mode

Table 33.7 Bus Timing

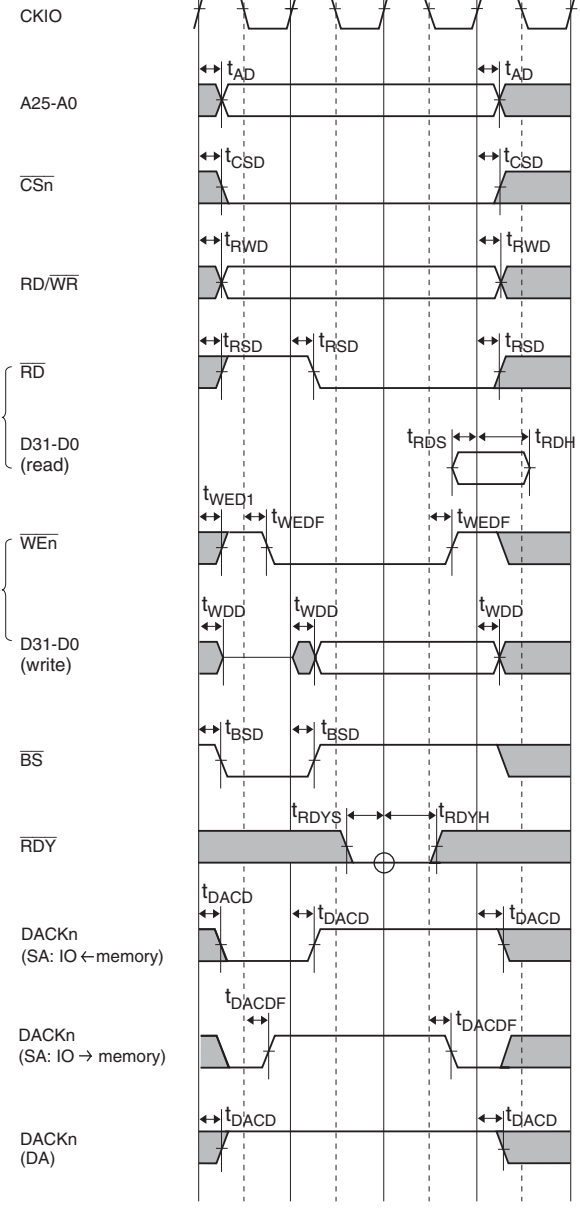
(V_{DDQ} = 3.0 to 3.6 V, V_{DD} = 1.5 V, T_a = –20 to 75°C/–40 to 85°C, C_L = 30 pF, PLL2 on)

Item	Symbol	Min.	Max.	Unit	Notes
Address delay time	t_{AD}	1.5	6	ns	
$\overline{BS}$ delay time	t_{BSD}	1.5	6	ns	
$\overline{CS}$ delay time	t_{CSD}	1.5	6	ns	
$\overline{RW}$ delay time	t_{RWD}	1.5	6	ns	
$\overline{RD}$ delay time	t_{RSD}	1.5	6	ns	
Read data setup time	t_{RDS}	3	—	ns	
Read data hold time	t_{RDH}	1.5	—	ns	
$\overline{WE}$ delay time (falling edge)	t_{WEDF}	1.5	6	ns	Relative to CKIO falling edge
$\overline{WE}$ delay time	t_{WED1}	1.5	6	ns	
Write data delay time	t_{WDD}	1.5	6	ns	
$\overline{RDY}$ setup time	t_{RDYS}	3	—	ns	
$\overline{RDY}$ hold time	t_{RDYH}	1.5	—	ns	
$\overline{RAS}$ delay time	t_{RASD}	1.5	6	ns	
$\overline{CAS}$ delay time 2	t_{CASD2}	1.5	6	ns	SDRAM
CKE delay time	t_{CKED}	1.5	6	ns	SDRAM
DQM delay time	t_{DQMD}	1.5	6	ns	SDRAM
$\overline{FRAME}$ delay time	t_{FMD}	1.5	6	ns	MPX
$\overline{IOIS16}$ setup time	t_{IO16S}	3	—	ns	PCMCIA
$\overline{IOIS16}$ hold time	t_{IO16H}	1.5	—	ns	PCMCIA
$\overline{ICIOWR}$ delay time (falling edge)	t_{ICWSDF}	1.5	6	ns	PCMCIA
$\overline{ICIORD}$ delay time	t_{ICRSD}	1.5	6	ns	PCMCIA
DACK delay time	t_{DACD}	1.5	6	ns	
DACK delay time (falling edge)	t_{DACDF}	1.5	6	ns	Relative to CKIO falling edge



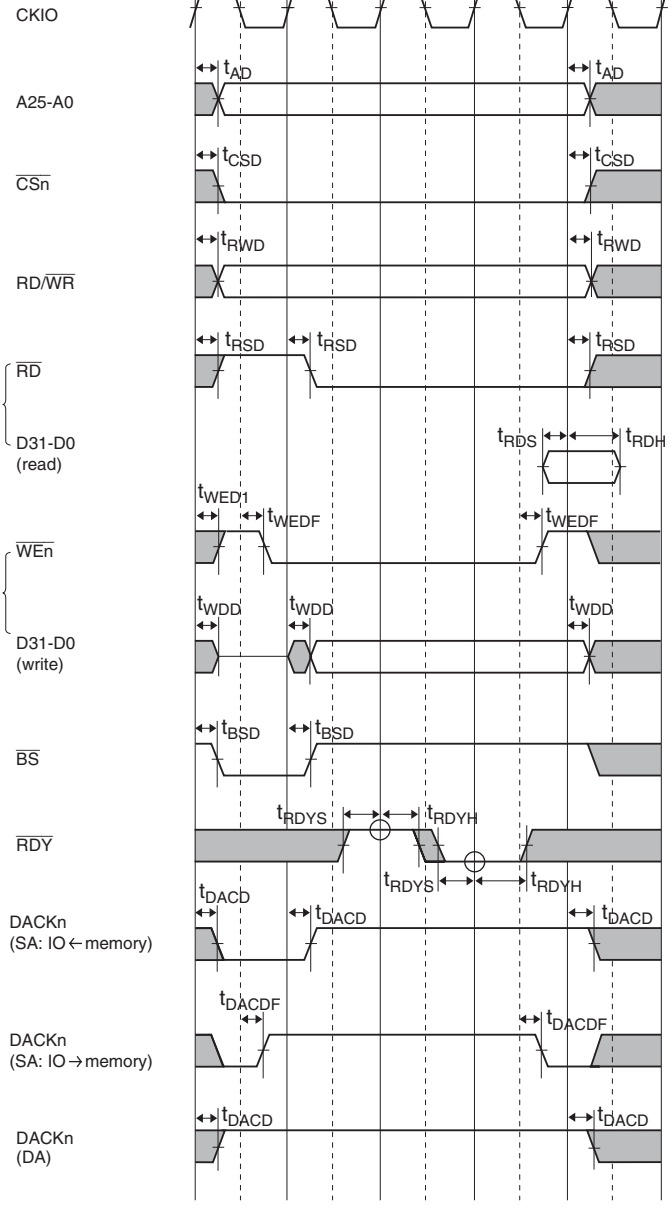
Notes: IO : Dack device
SA : Single address DMA transfer
DA : Dual address DMA transfer
DACK set to active-high

Figure 33.17 SRAM Bus Cycle: Basic Bus Cycle (No Wait)



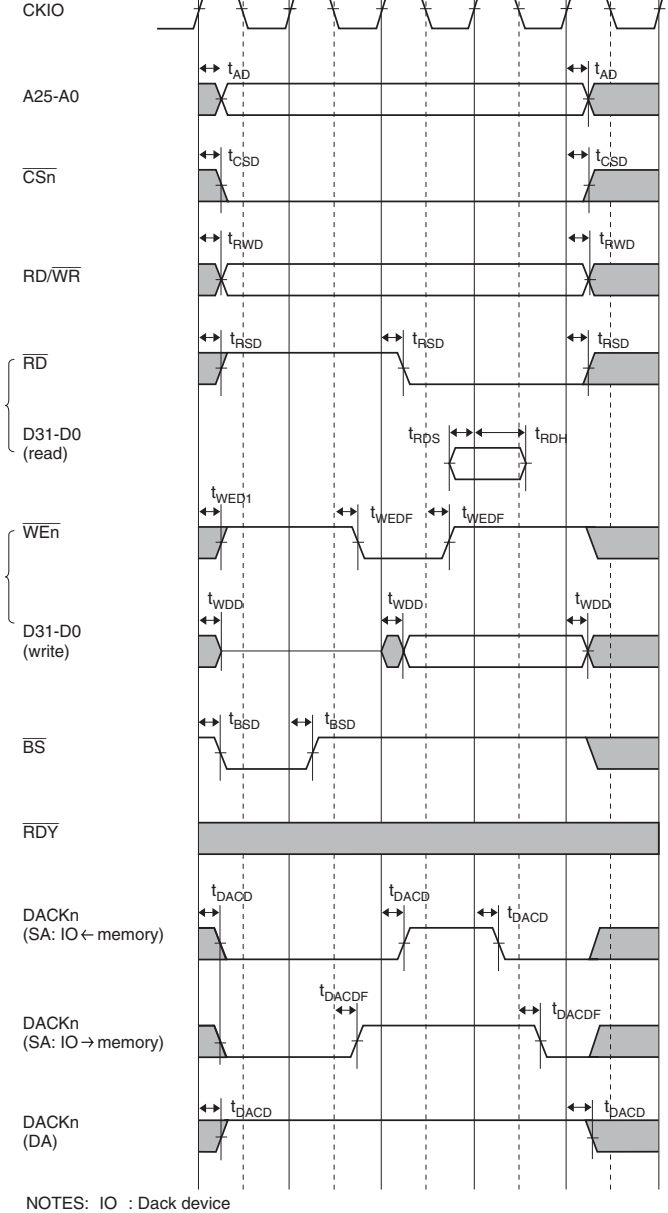
NOTES: IO : Dack device
SA: Single address DMA transfer
DA: Dual address DMA transfer
DACK set to active-high

Figure 33.18 SRAM Bus Cycle: Basic Bus Cycle (One Internal Wait)



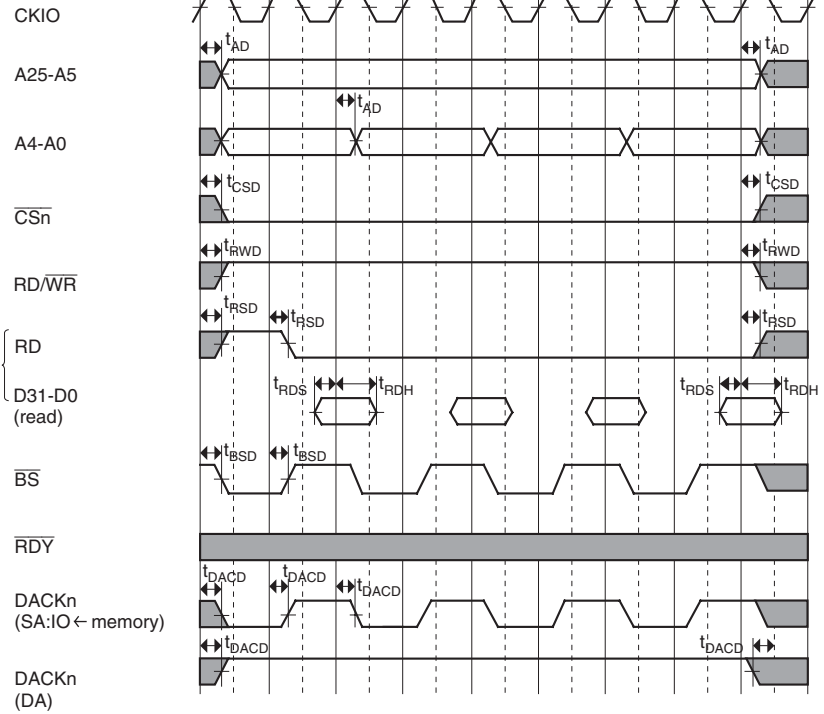
NOTES: IO : Dack device
SA : Single address DMA transfer
DA : Dual address DMA transfer
DACK set to active-high

Figure 33.19 SRAM Bus Cycle: Basic Bus Cycle (One Internal Wait + One External Wait)



NOTES: IO : Dack device
SA : Single address DMA transfer
DA : Dual address DMA transfer
DACK set to active-high

Figure 33.20 SRAM Bus Cycle: Basic Bus Cycle (No Wait, Address Setup/Hold Time Insertion, AnS = 1, AnH = 1)



NOTES: IO : Dack device
SA : Single address DMA transfer
DA : Dual address DMA transfer
DACK set to active-high

Figure 33.21 Burst ROM Bus Cycle (No Wait)

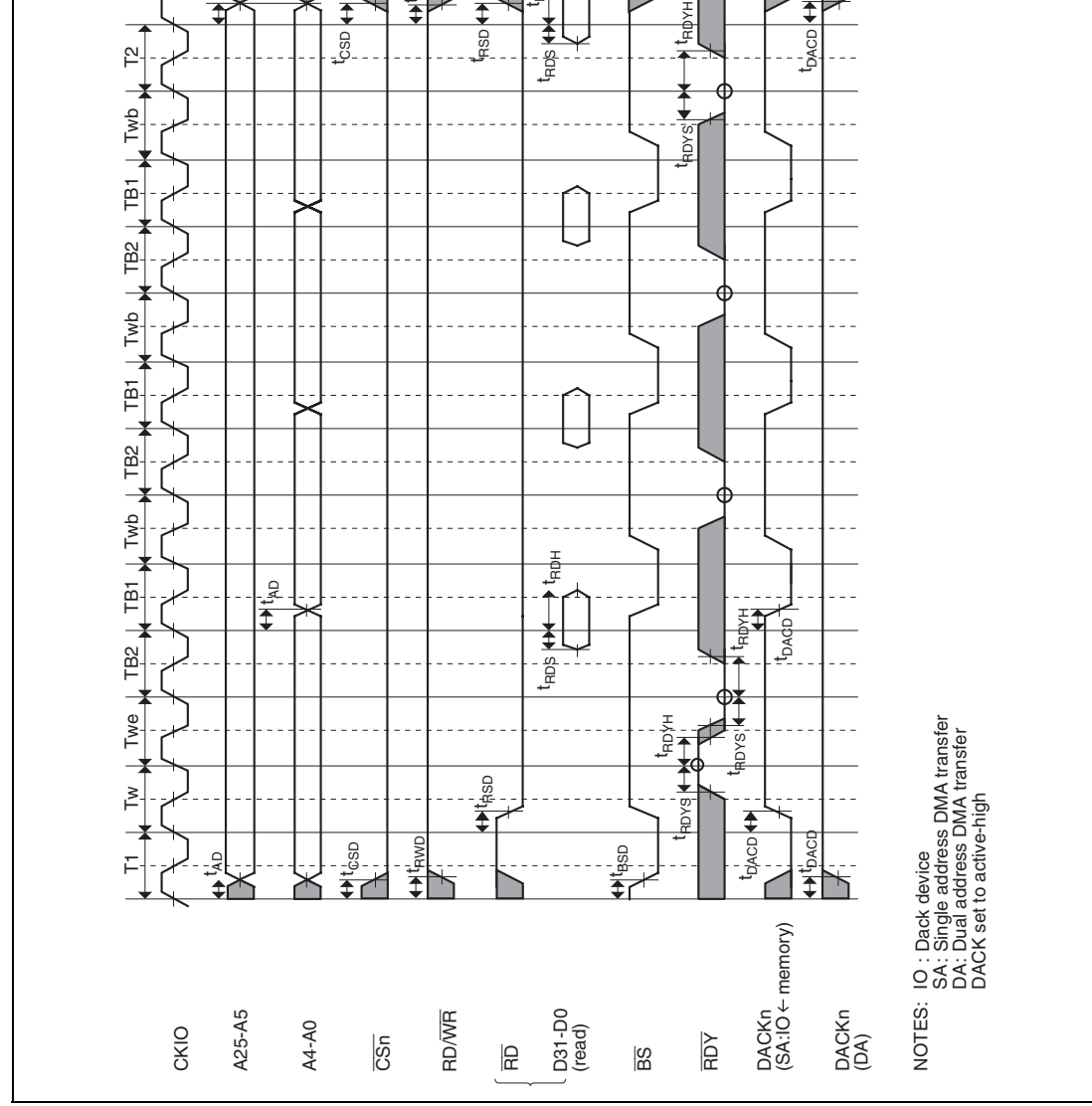


Figure 33.22 Burst ROM Bus Cycle
(1st Data: One Internal Wait + One External Wait ; 2nd/3rd/4th Data: One Internal Wait)

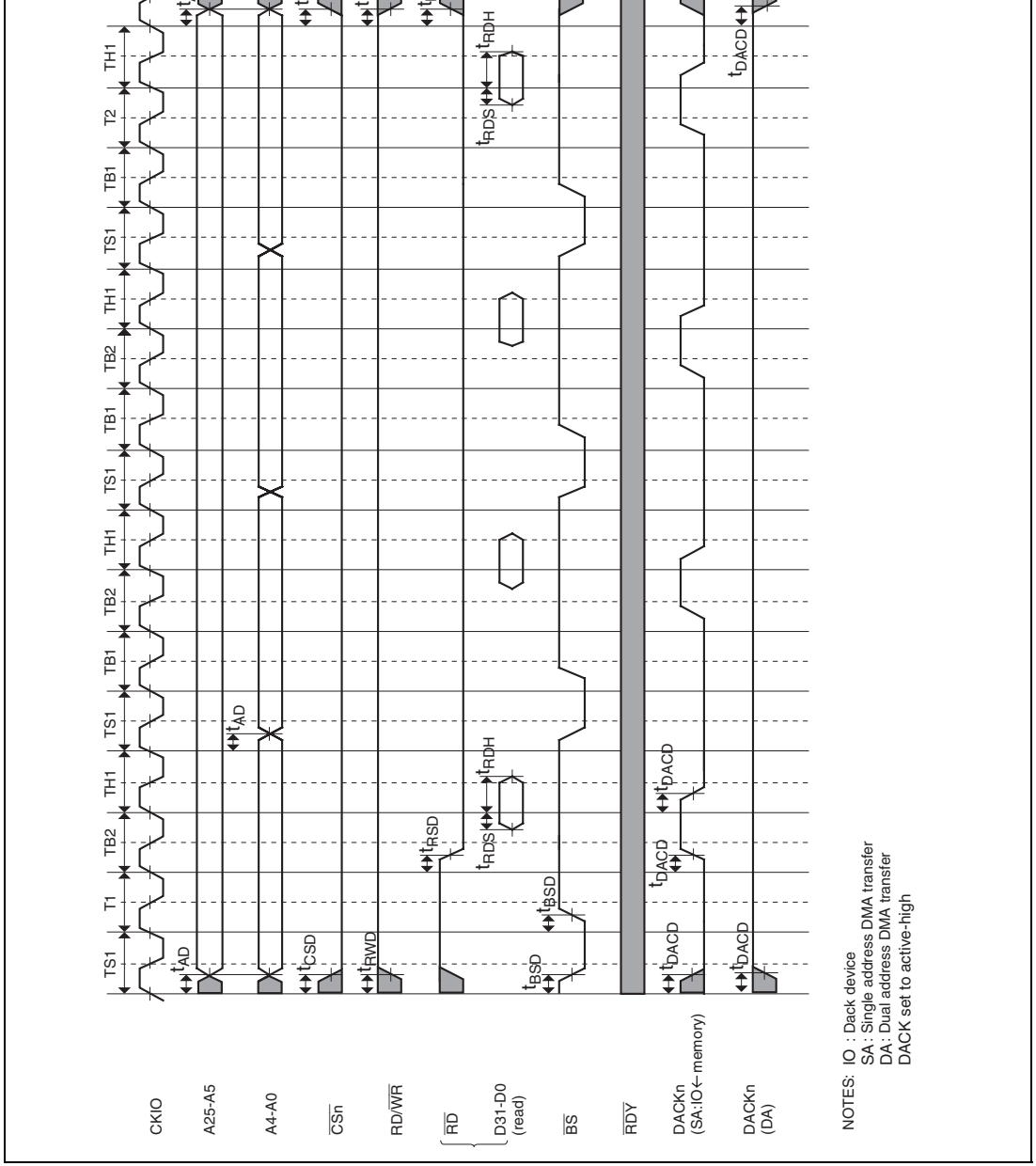


Figure 33.23 Burst ROM Bus Cycle
(No Wait, Address Setup/Hold Time Insertion, AnS = 1, AnH = 1)

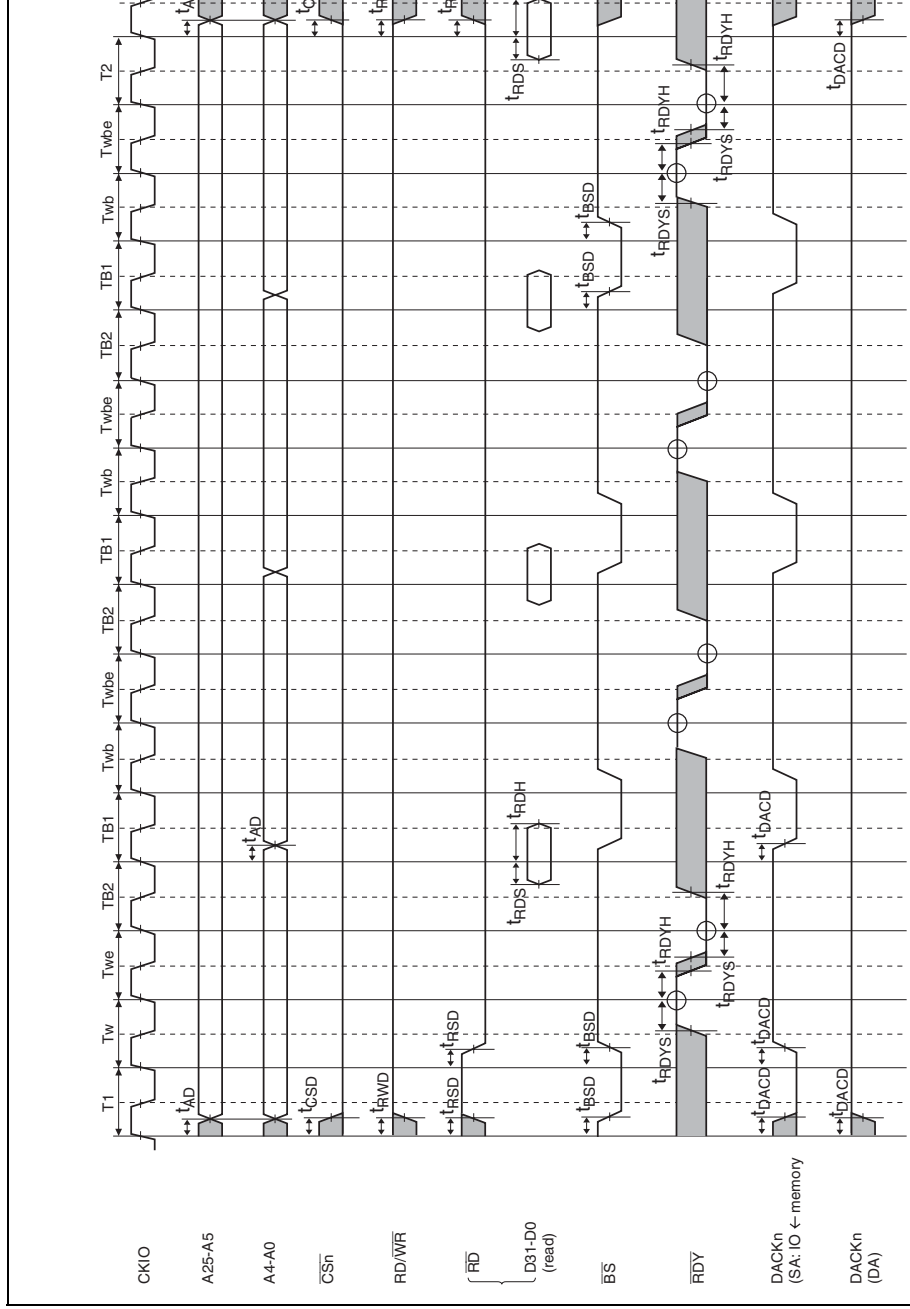


Figure 33.24 Burst ROM Bus Cycle (One Internal Wait + One External Wait)

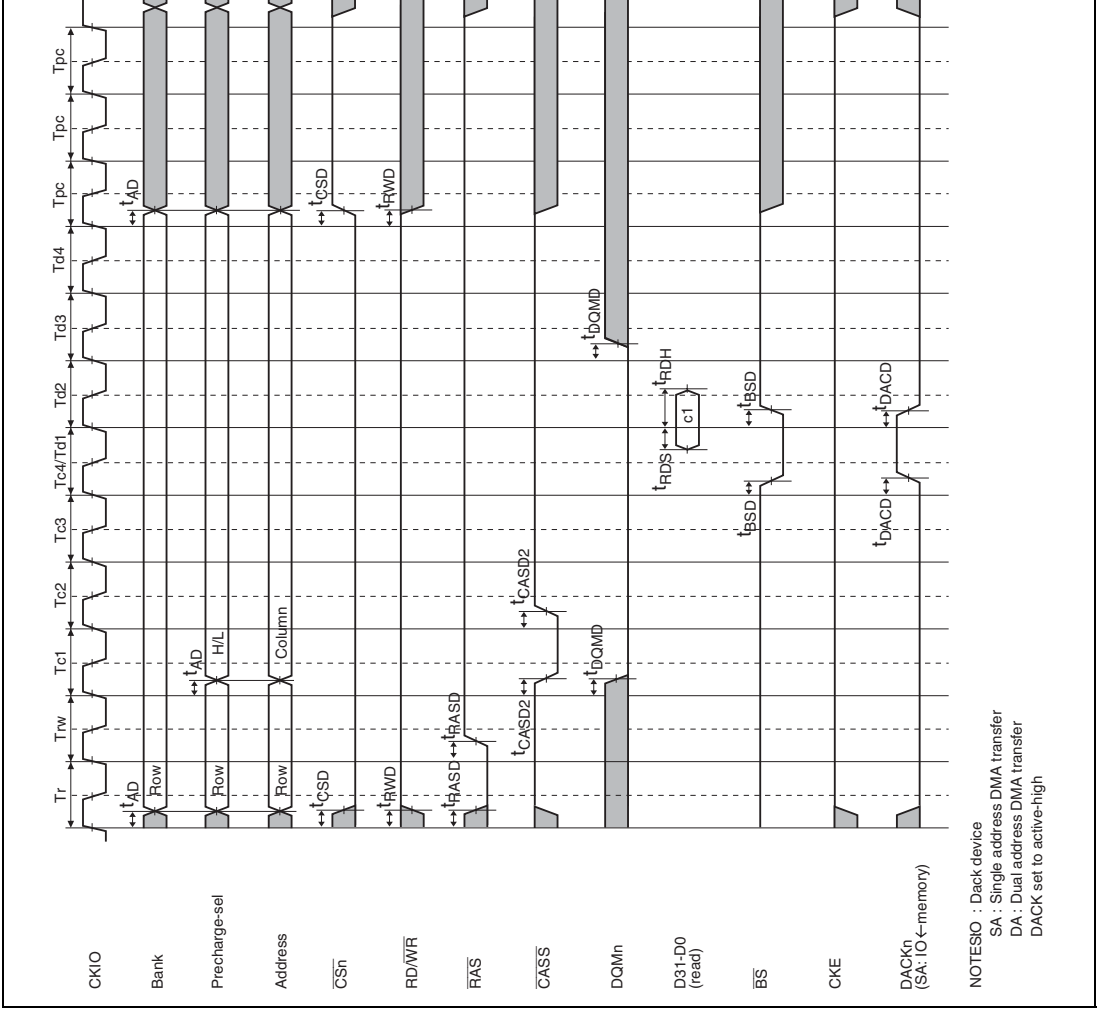


Figure 33.25 Synchronous DRAN Auto-Precharge Read Bus Cycle: Single (RCD[1:0] = 01, CAS Latency = 3, TPC[2:0] = 011)

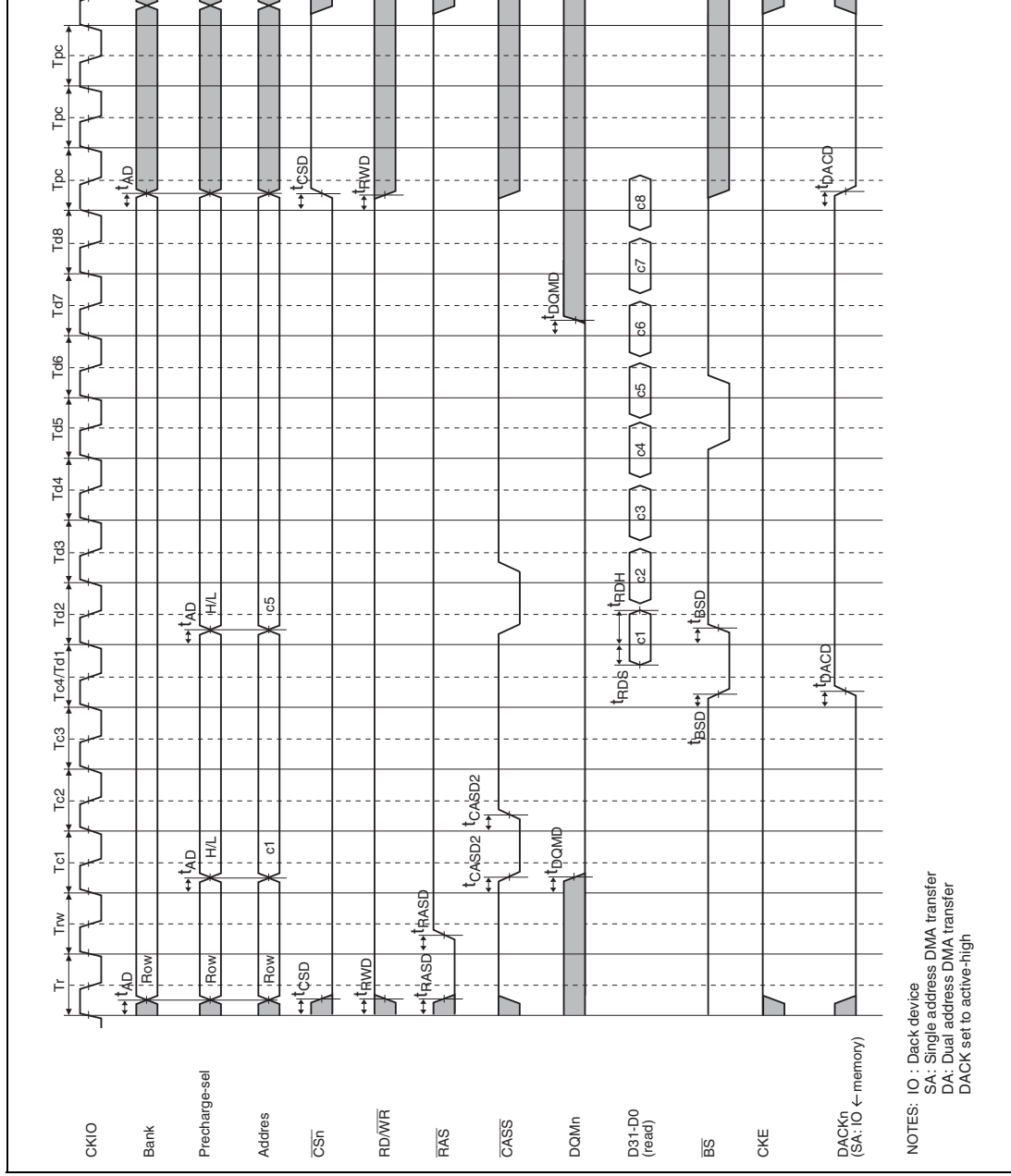


Figure 33.26 Synchronous DRAM Auto-Precharge Read Bus Cycle: Burst (RCD[1:0] = 01, CAS Latency = 3, TPC[2:0] = 011)

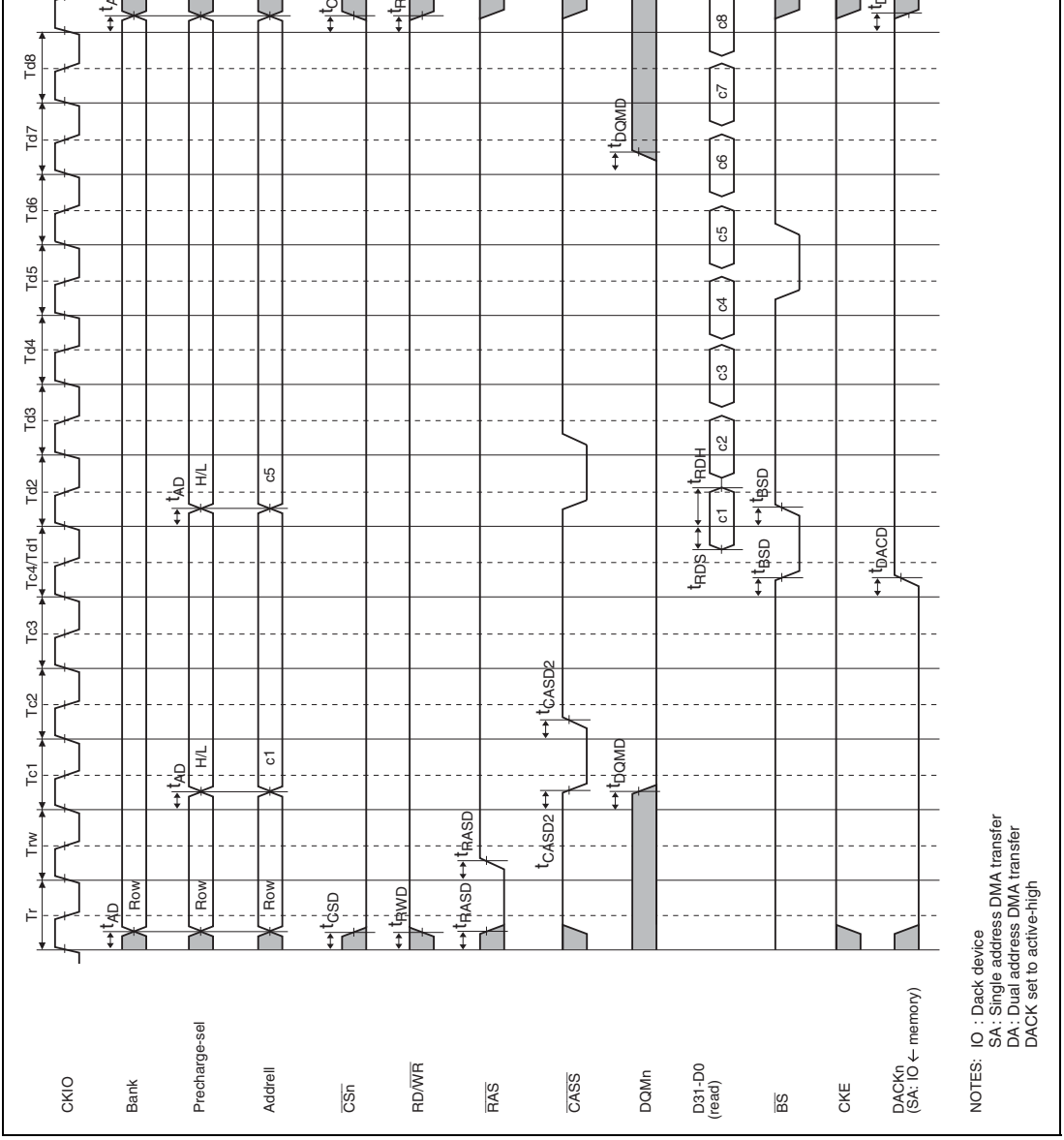


Figure 33.27 Synchronous DRAM Normal Read Bus Cycle: ACT + READ Commands, Burst (RCD[1:0] = 01, CAS Latency = 3)

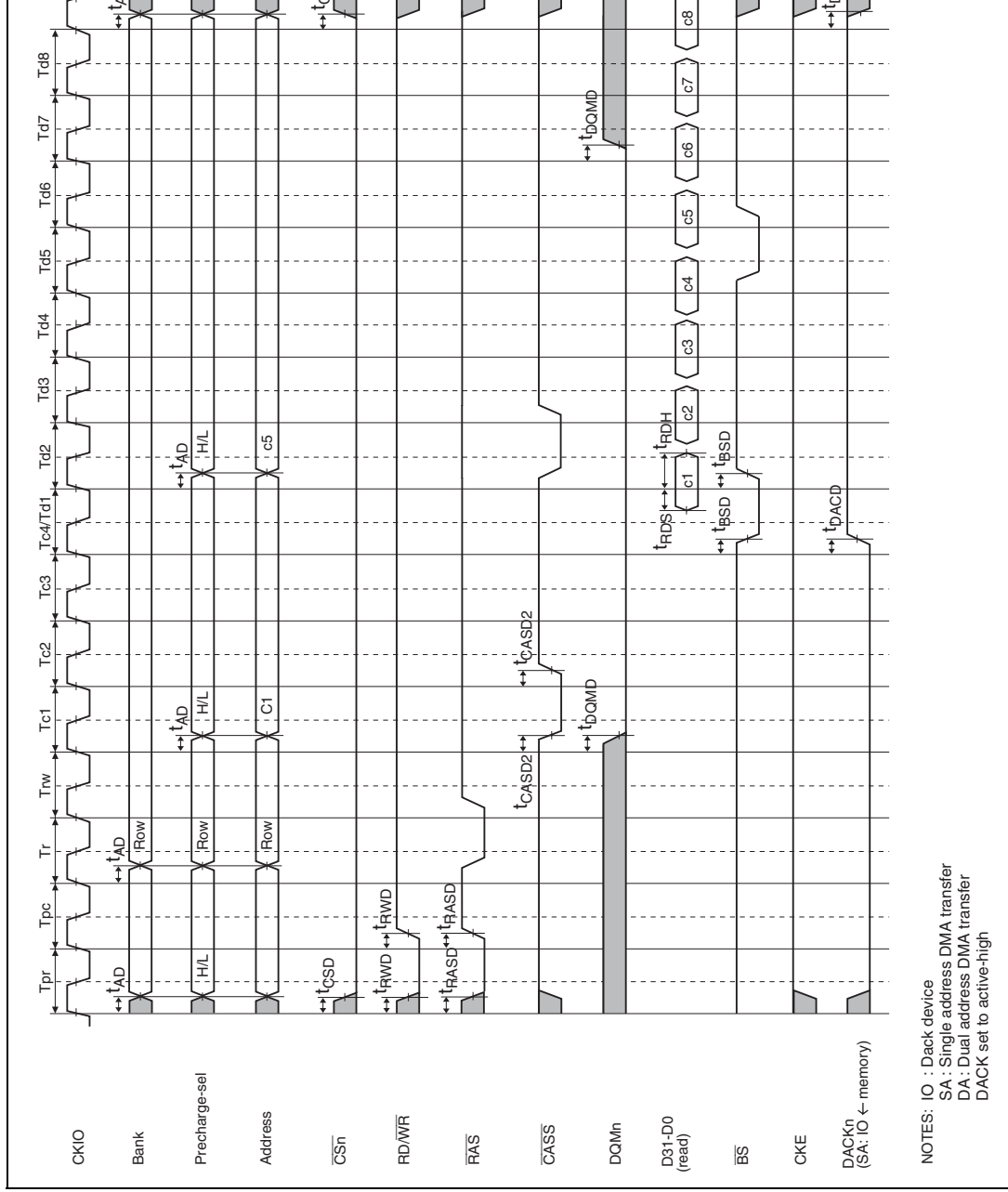
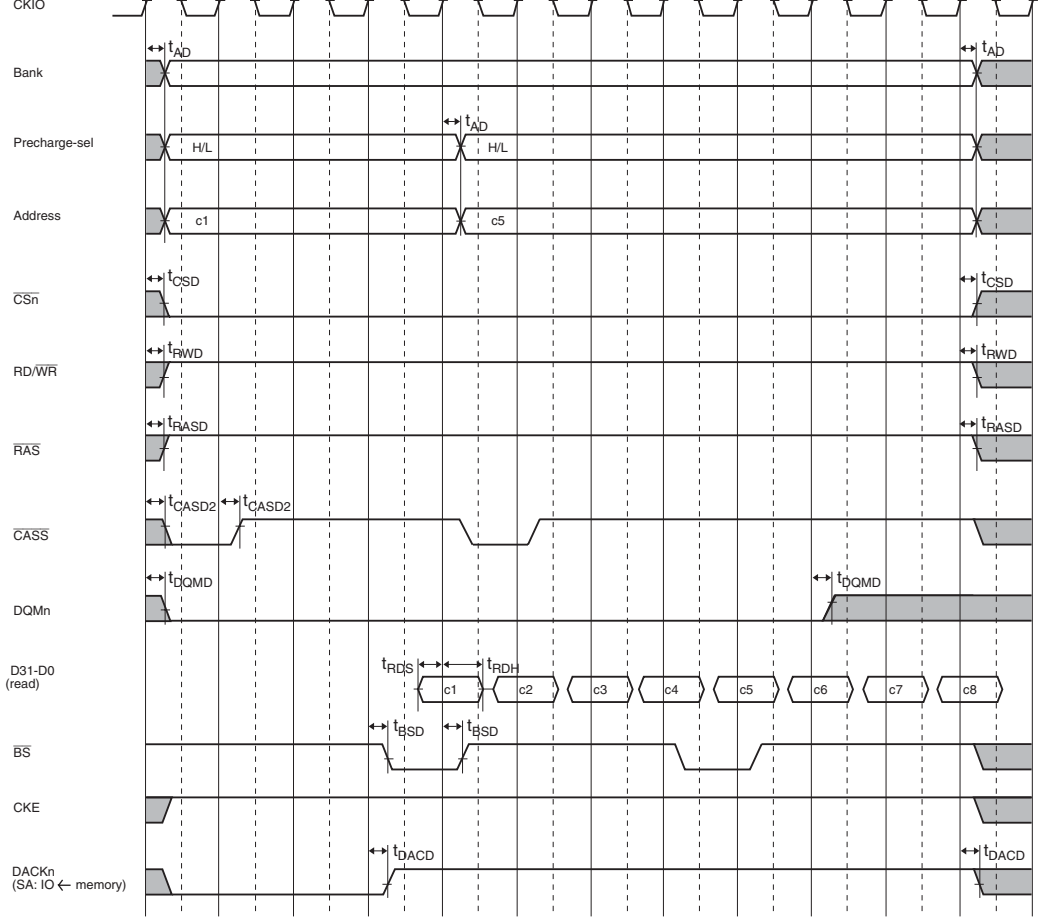
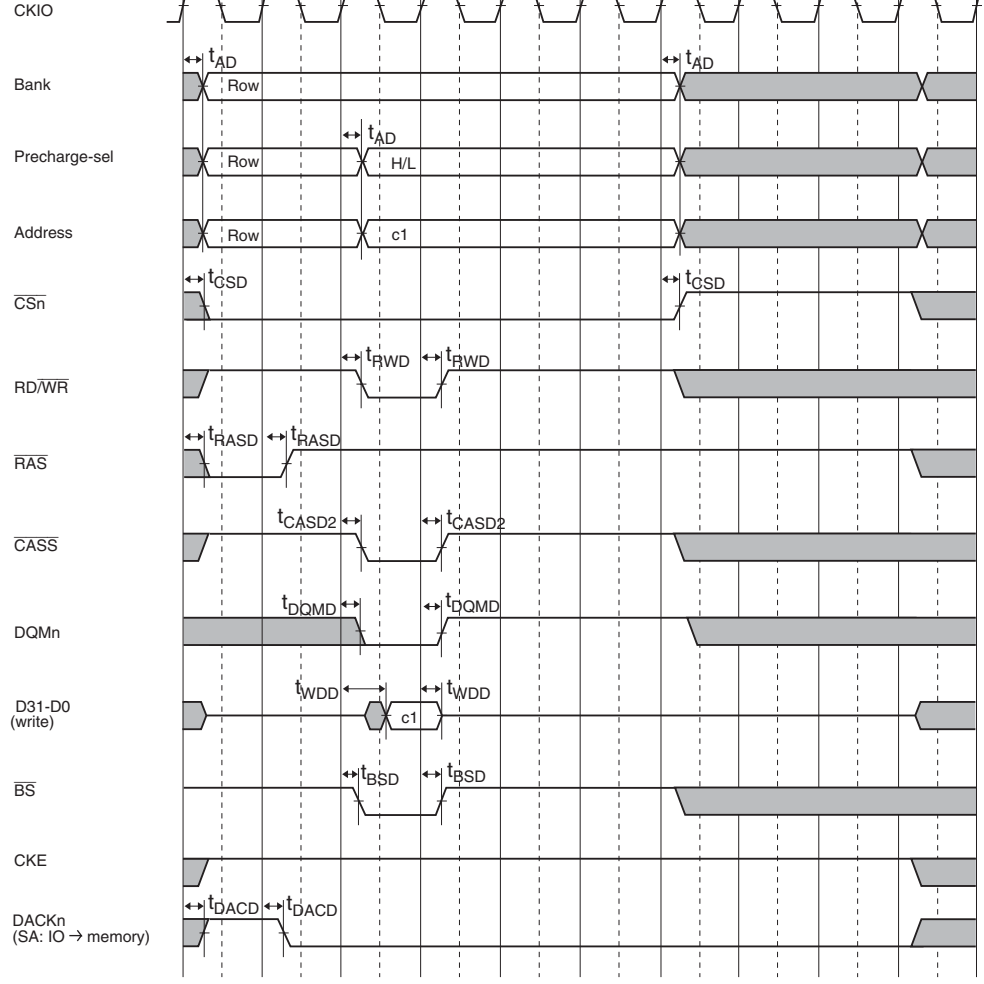


Figure 33.28 Synchronous DRAM Normal Read Bus Cycle: PRE + ACT + READ Commands, Burst (RCD[1:0] = 01, TPC[2:0] = 001, CAS Latency = 3)



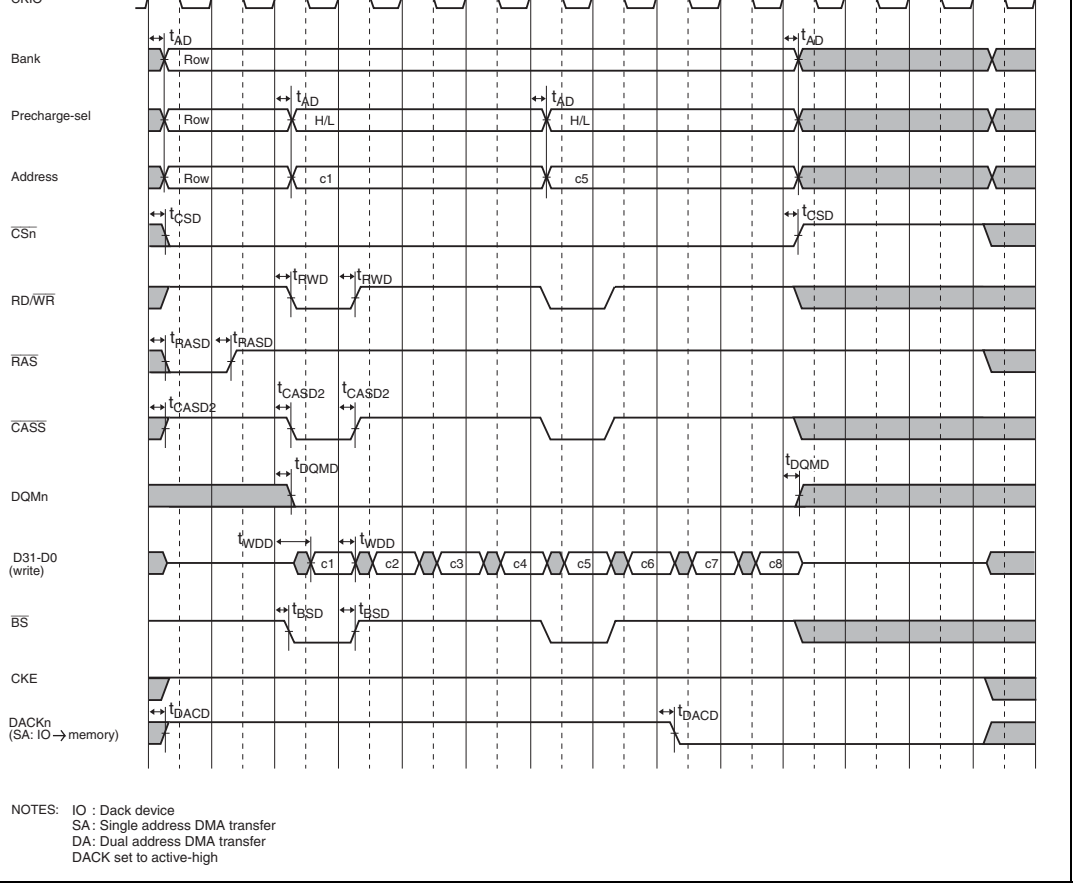
NOTES: IO : Dack device
SA : Single address DMA transfer
DA : Dual address DMA transfer
DACK set to active-high

Figure 33.29 Synchronous DRAM Normal Read Bus Cycle: READ Command, Burst (CAS Latency = 3)



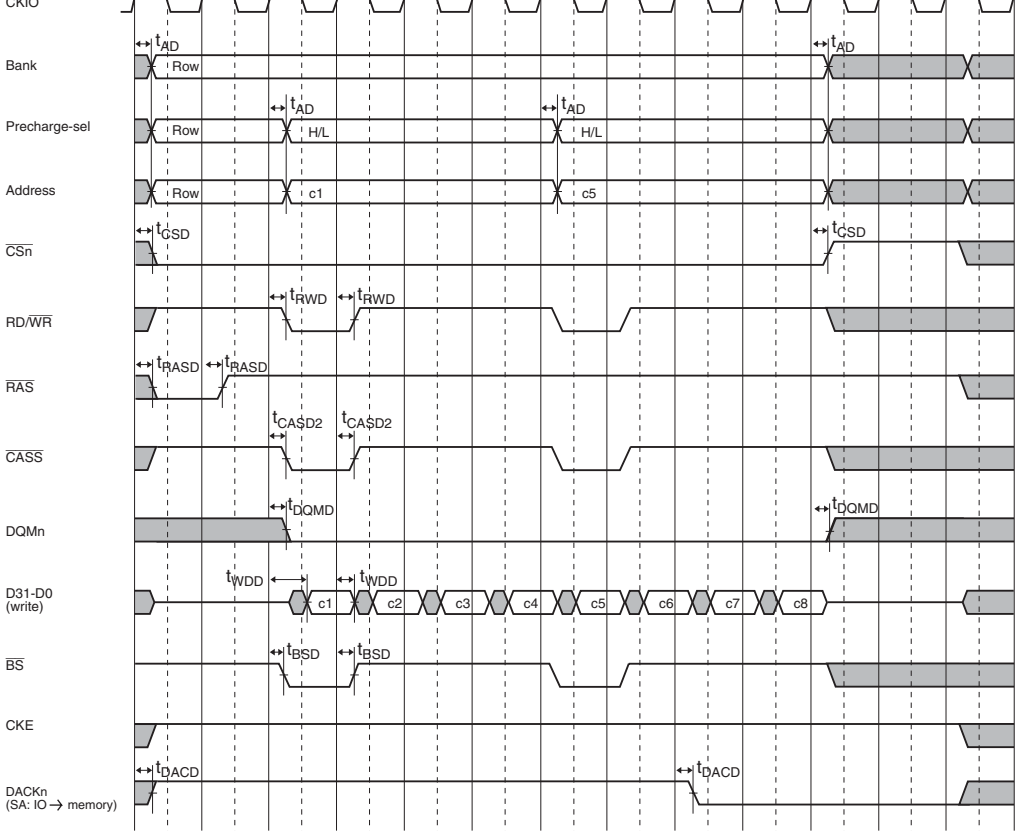
NOTES: IO : Dack device
SA : Single address DMA transfer
DA : Dual address DMA transfer
DACK set to active-high

Figure 33.30 Synchronous DRAM Auto-Precharge Write Bus Cycle: Single (RCD[1:0] = 01, TPC[2:0] = 001, TRWL[2:0] = 010)



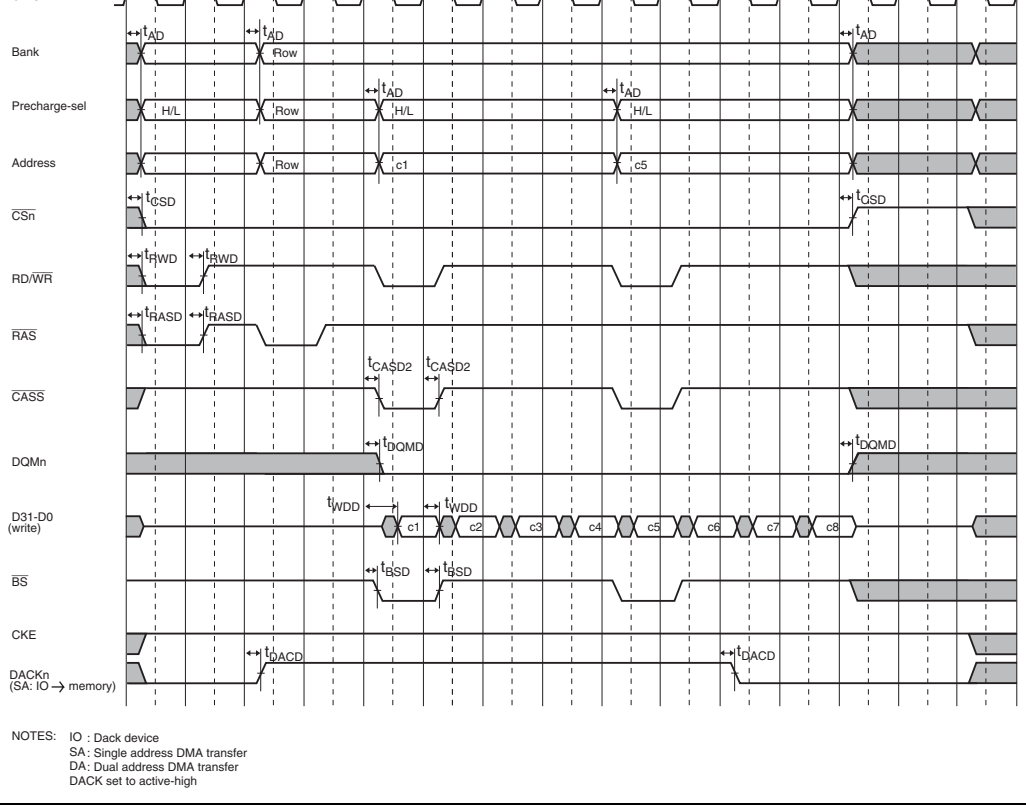
NOTES: IO : Dack device
SA: Single address DMA transfer
DA: Dual address DMA transfer
DACK set to active-high

Figure 33.31 Synchronous DRAM Auto-Precharge Write Bus Cycle: Burst (RCD[1:0] = 01, TPC[2:0] = 001, TRWL[2:0] = 010)



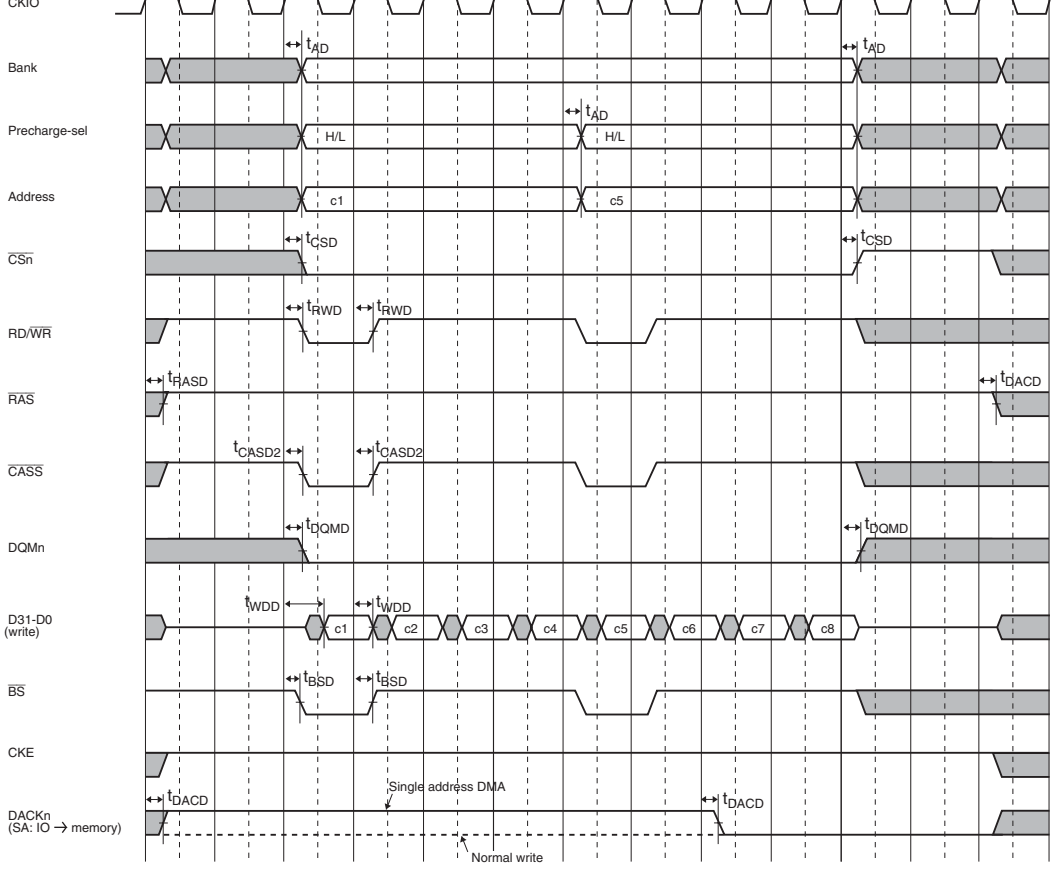
NOTES: IO : Dack device
SA : Single address DMA transfer
DA : Dual address DMA transfer
DACK set to active-high

Figure 33.32 Synchronous DRAM Normal Write Bus Cycle: ACT+WRITE Commands, Burst (RCD[1:0] = 01, TRWL[2:0] = 010)



NOTES: IO : Dack device
SA: Single address DMA transfer
DA: Dual address DMA transfer
DACK set to active-high

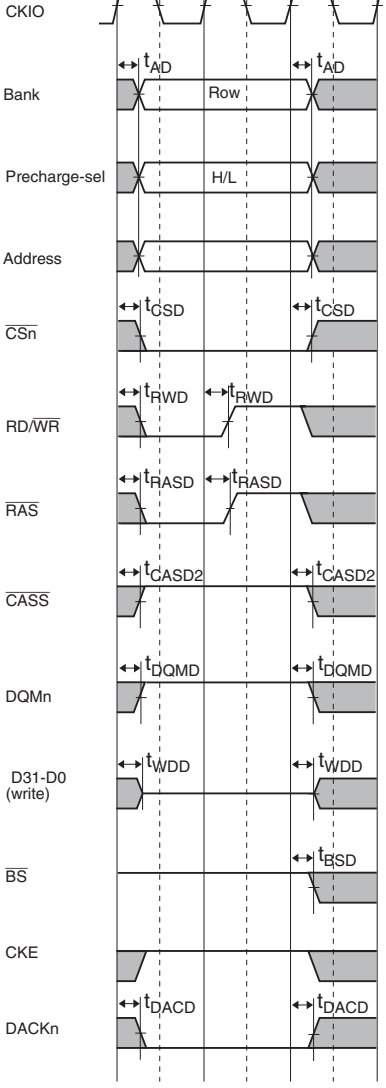
Figure 33.33 Synchronous DRAM Normal Write Bus Cycle: PRE+ACT+WRITE Commands, Burst (RCD[1:0] = 01, TPC[2:0] = 001, TRWL[2:0] = 010)



NOTES: IO : Dack device
SA : Single address DMA transfer
DA : Dual address DMA transfer
DACK set to active-high

In the case of SA-DMA only, the (Tnop) cycle is inserted, and the DACKn signal is output as shown by the solid line. In a normal write, the (Tnop) cycle is omitted and the DACKn signal is output as shown by the dotted line.

Figure 33.34 Synchronous DRAM Normal Write Bus Cycle: WRITE Command, Burst (TRWL[2:0] = 010)



NOTES: IO : Dack device
SA: Single address DMA transfer
DA: Dual address DMA transfer
DACK set to active-high

Figure 33.35 Synchronous DRAM Bus Cycle: Precharge Command (TPC[2:0] = 001)

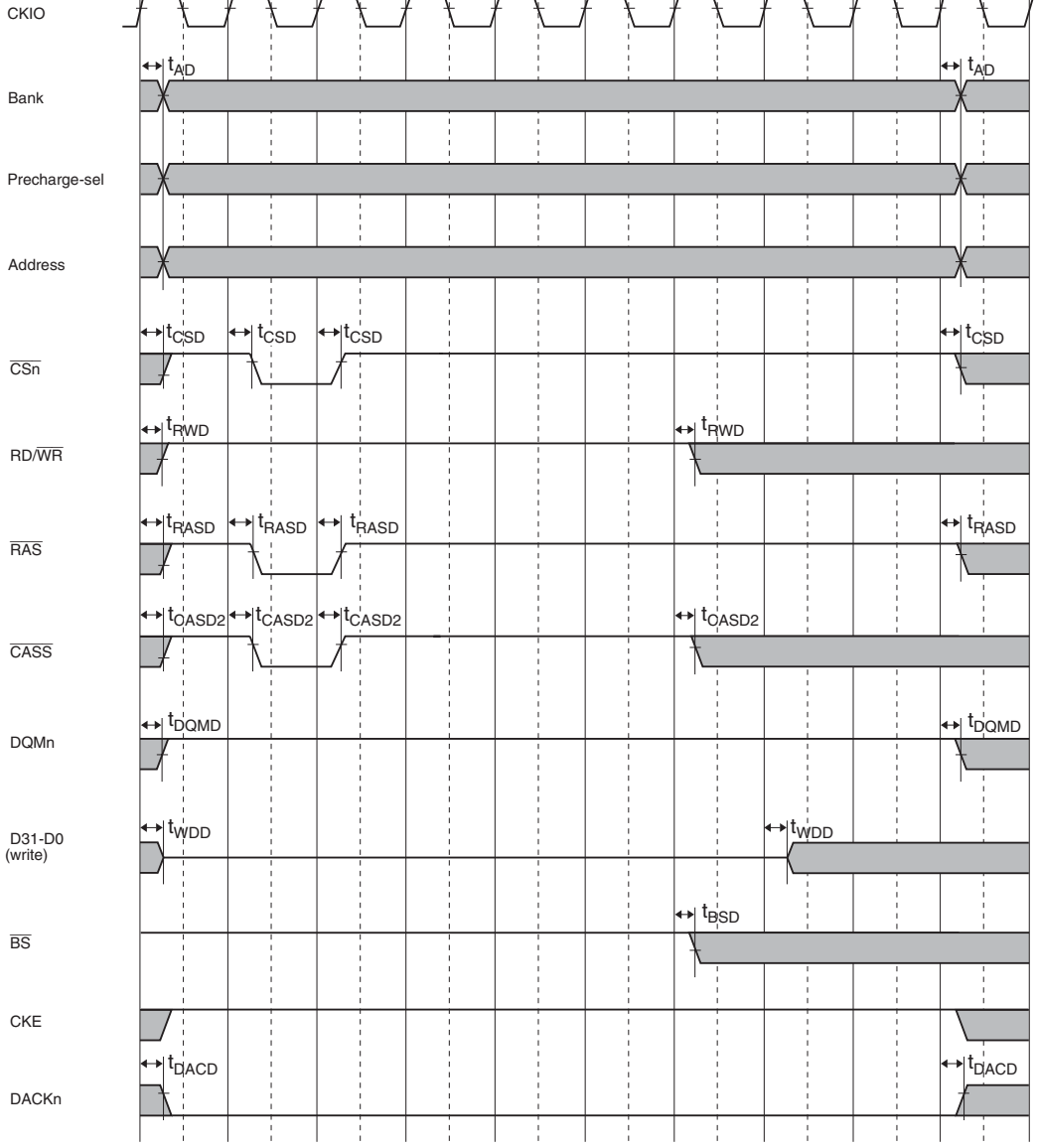
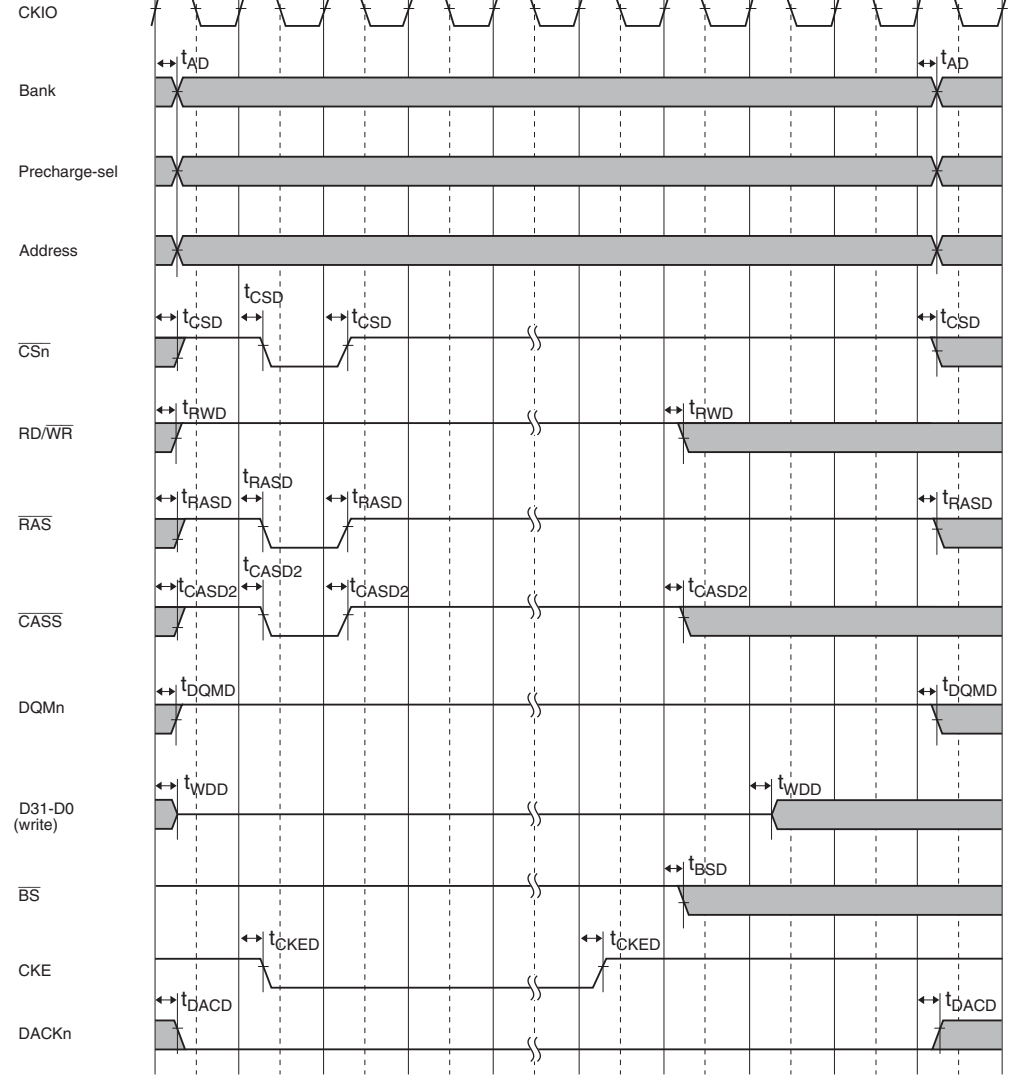
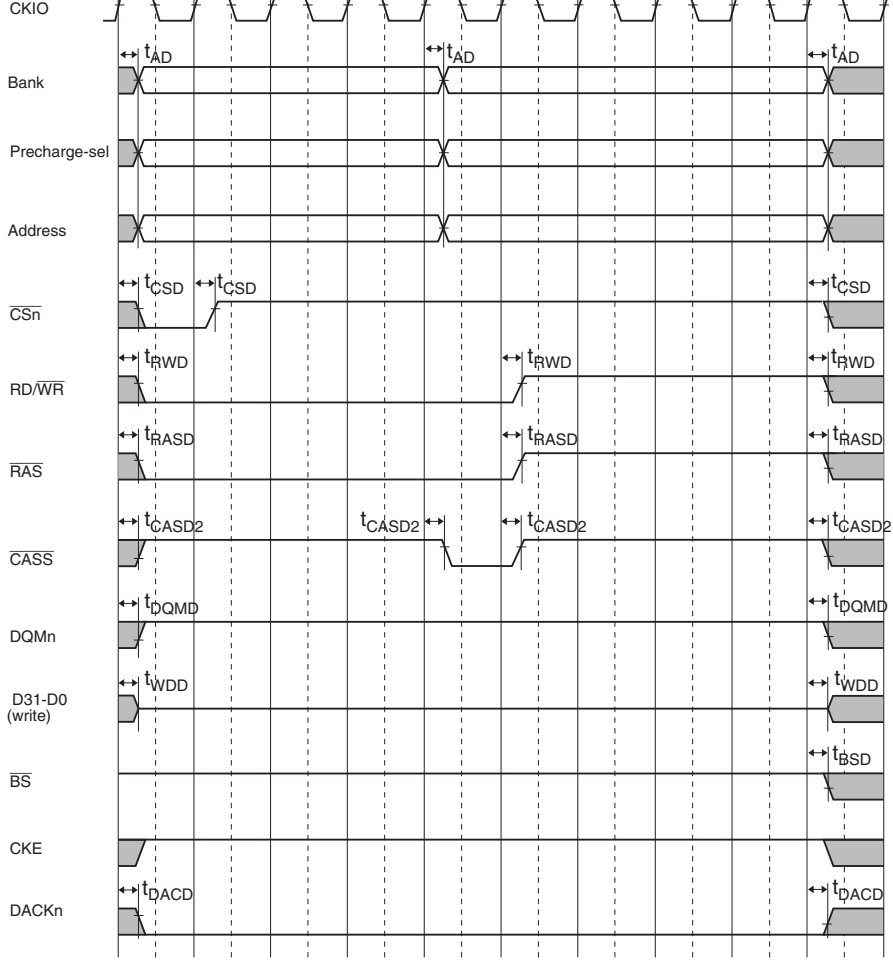


Figure 33.36 Synchronous DRAM Bus Cycle: Auto-Refresh (TRAS = 1, TRC[2:0] = 001)



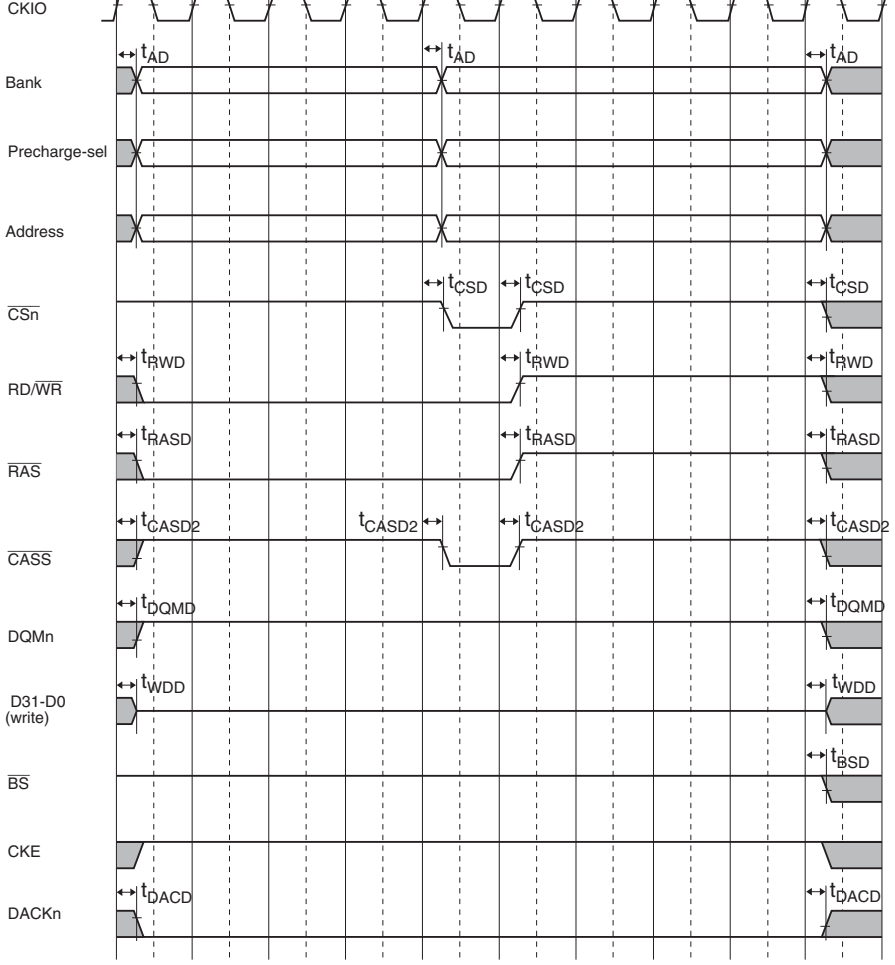
NOTES: IO : Dack device
SA: Single address DMA transfer
DA: Dual address DMA transfer
DACK set to active-high

Figure 33.37 Synchronous DRAM Bus Cycle: Self-Refresh (TRC[2:0] = 001)



NOTES: IO : Dack device
SA: Single address DMA transfer
DA: Dual address DMA transfer
DACK set to active-high

Figure 33.38 Synchronous DRAM Bus Cycle: Mode Register Setting (PALL)



NOTES: IO : Dack device
SA : Single address DMA transfer
DA : Dual address DMA transfer
DACK set to active-high

Figure 33.39 Synchronous DRAM Bus Cycle: Mode Register Setting (SET)

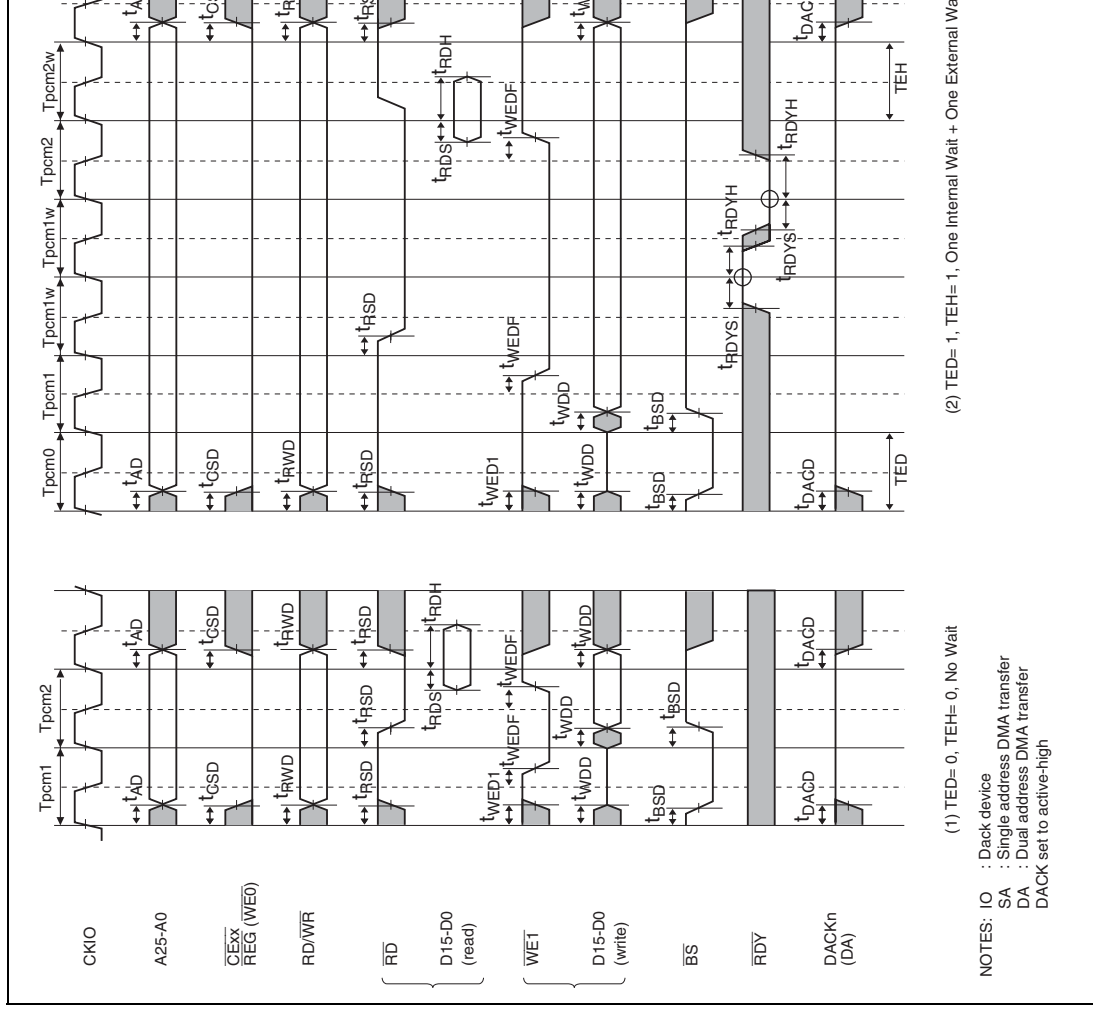


Figure 33.40 PCMCIA Memory Bus Cycle

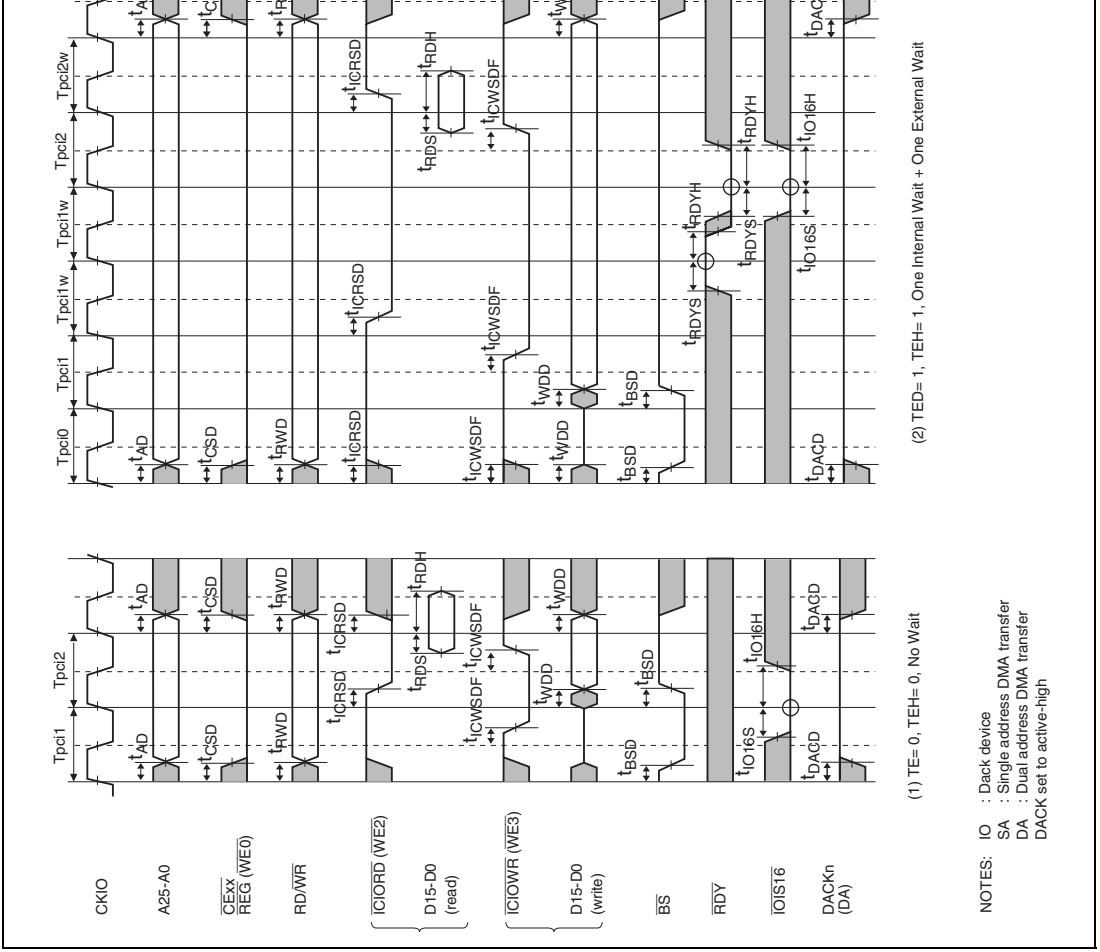


Figure 33.41 PCMCIA I/O Bus Cycle

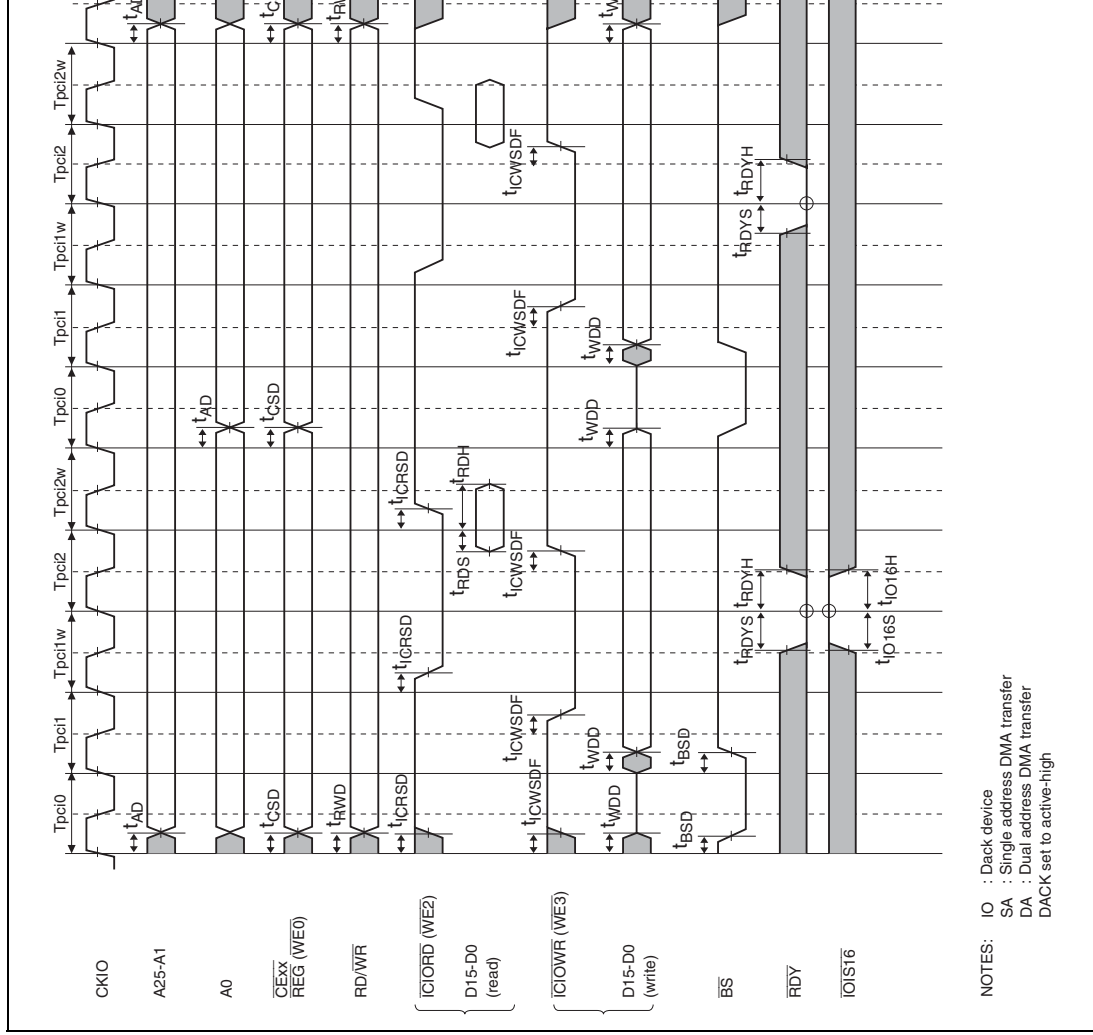


Figure 33.42 PCMCIA I/O Bus Cycle (TED = 1, TEH = 1, One Internal Wait, Bus Sizing)

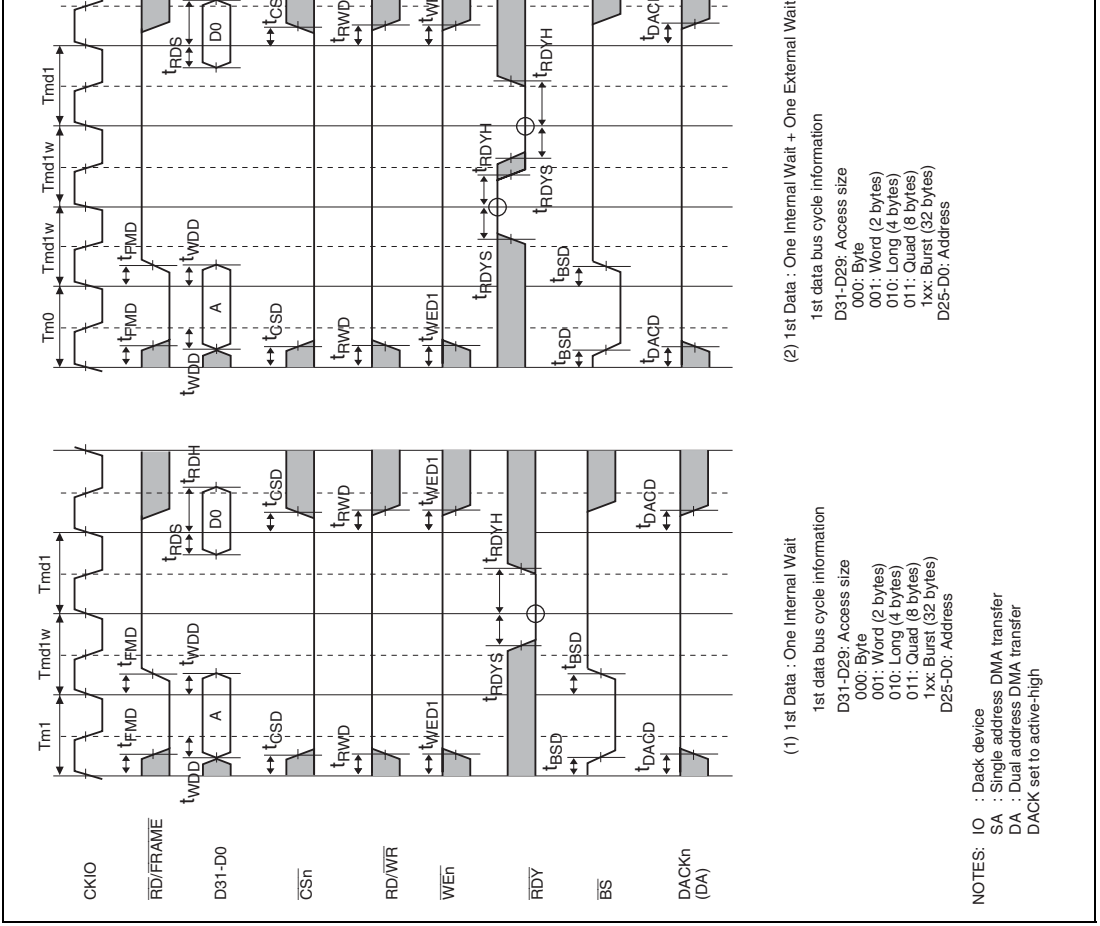


Figure 33.43 MPX Basic Bus Cycle: Read

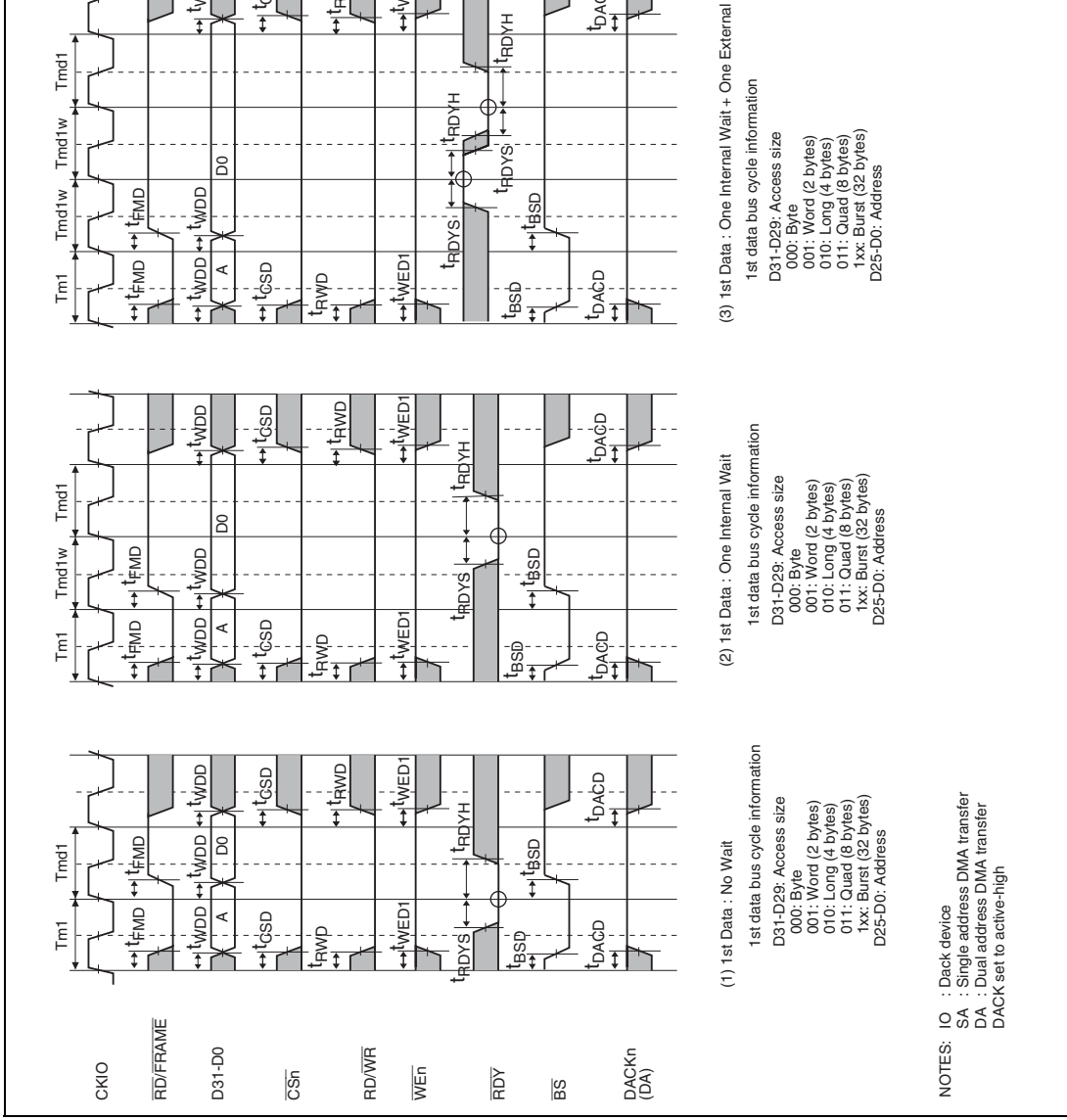


Figure 33.44 MPX Basic Bus Cycle: Write

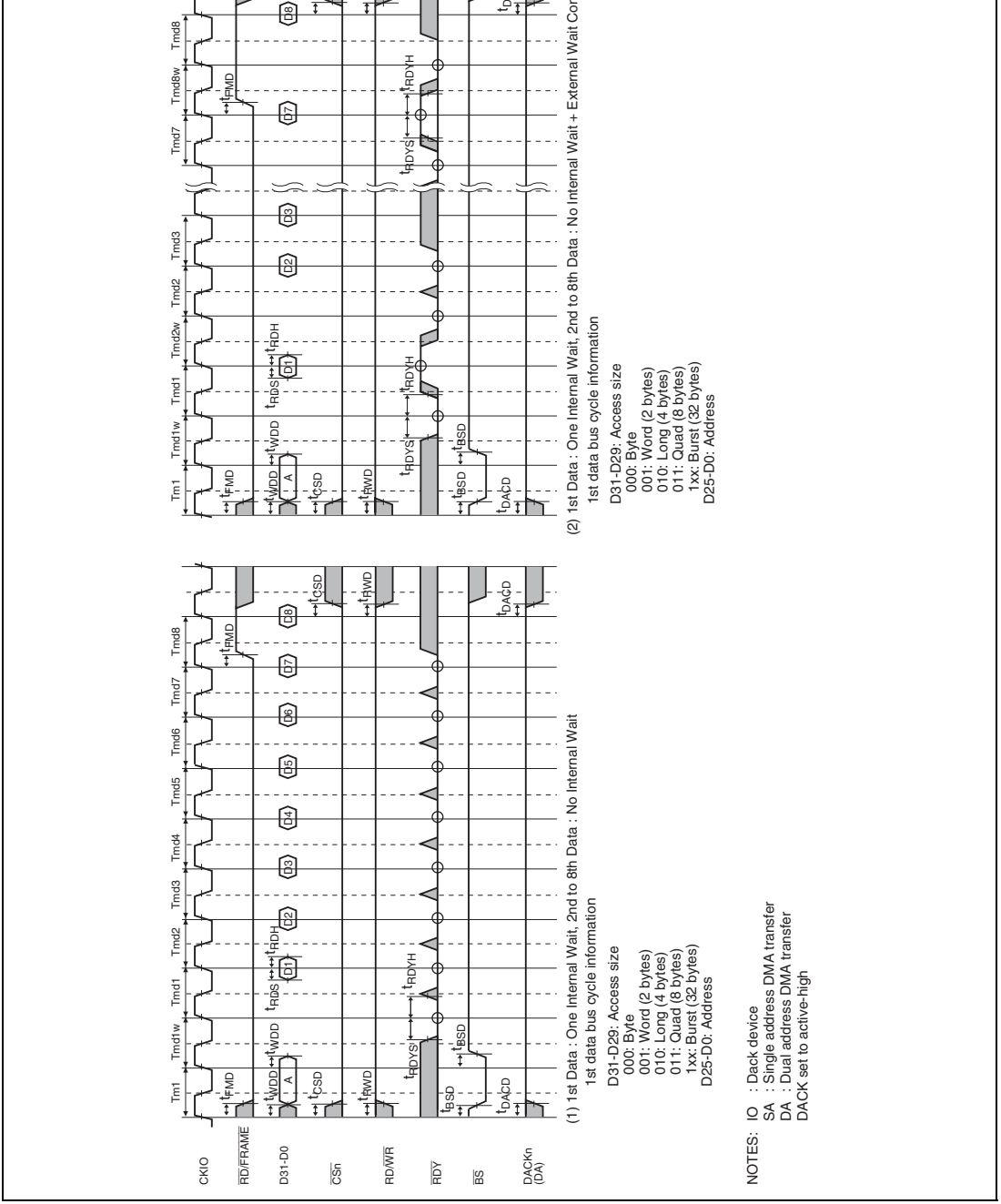


Figure 33.45 MPX Bus Cycle: Burst Read

Figure 33.46 MPX Bus Cycle: Burst Write

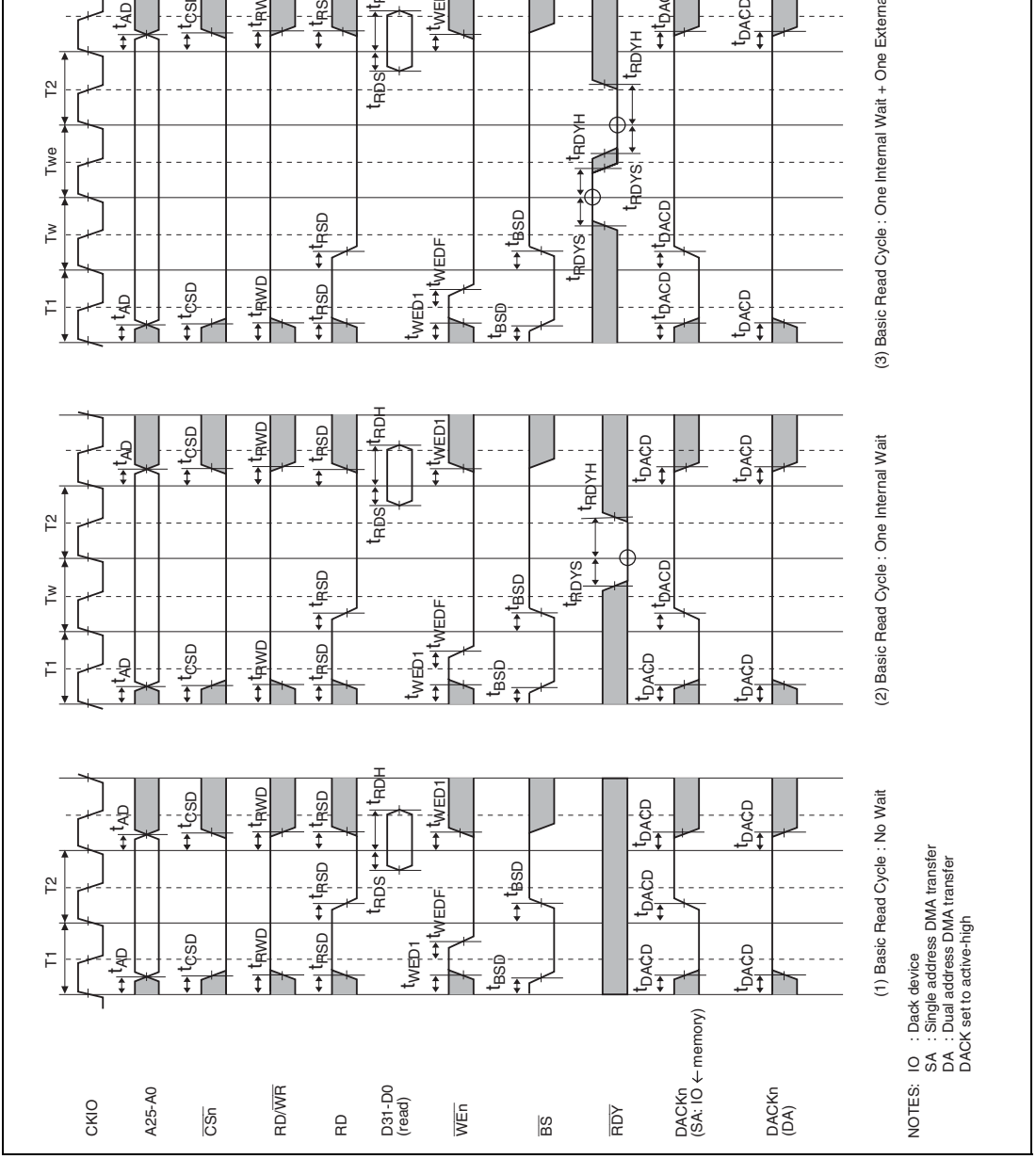
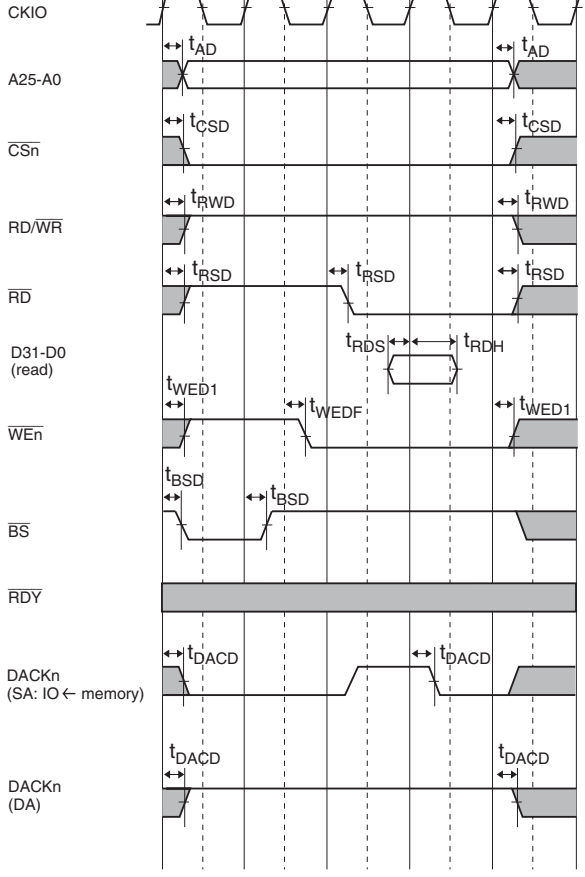


Figure 33.47 Memory Byte Control SRAM Bus Cycle



Notes: IO : Dack device
SA : Single address DMA transfer
DA : Dual address DMA transfer
DACK set to active-high

Figure 33.48 Memory Byte Control SRAM Bus Cycle: Basic Read Cycle (No Wait, Address Setup/Hold Time Insertion, AnS = 1, AnH = 1)

Table 33.8 INTC Module Signal Timing

(V_{DDQ} = 3.0 to 3.6 V, V_{DD} = 1.5 V, T_a = –20 to 75°C/–40 to 85°C, C_L = 30 pF, PLL2 on)

Module	Item	Symbol	Min.	Max.	Unit	Figure	Notes
INTC	NMI pulse width (High)	t_{NMIH}	5	—	t_{cyc}	33.49	normal mode sleep mode
			30	—	ns	33.49	standby mode
	NMI pulse width (Low)	t_{NMIL}	5	—	t_{cyc}	33.49	normal mode sleep mode
			30	—	ns	33.49	standby mode

Note: t_{cyc}: one CKIO cycle time

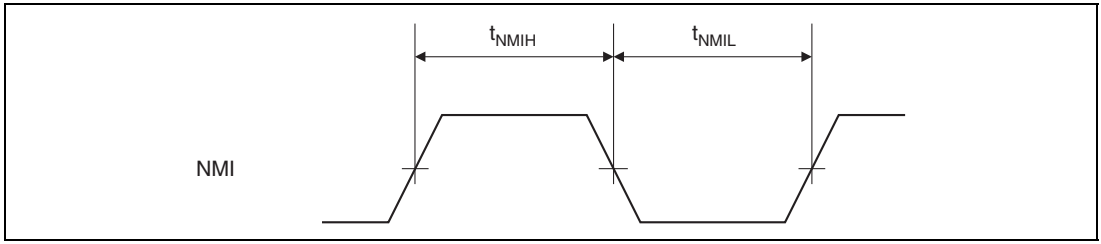


Figure 33.49 NMI Input Timing

33.3.5 DMAC Module Signal Timing

Table 33.9 DMAC Module Signal Timing

(V_{DDQ} = 3.0 to 3.6 V, V_{DD} = 1.5 V, T_a = –20 to 75°C/–40 to 85°C, C_L = 30 pF, PLL2 on)

Module	Item	Symbol	Min.	Max.	Unit	Figure	Notes
DMAC	$\overline{\text{DREQ}}$n setup time	t_{DRQS}	3	—	ns	33.50	
	$\overline{\text{DREQ}}$n hold time	t_{DRQH}	1.5	—	ns	33.50	
	DRAKn delay time	t_{DRAKD}	1.5	6	ns	33.50	

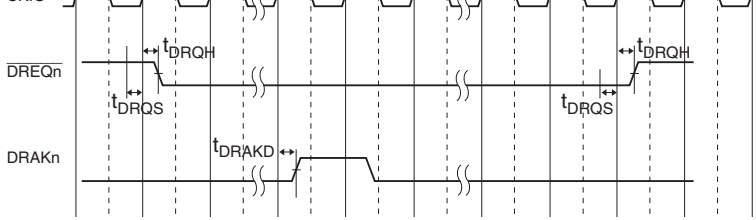


Figure 33.50 $\overline{\text{DREQ}}$/DRAK Timing

33.3.6 TMU Module Signal Timing

Table 33.10 TMU Module Signal Timing

(V_{DDQ} = 3.0 to 3.6 V, V_{DD} = 1.5V, T_a = –20 to 75°C/–40 to 85°C, C_L = 30 pF, PLL2 on)

Module	Item	Symbol	Min.	Max.	Unit	Figure	Notes
TMU	Timer clock pulse width (High)	t_{TCLKWH}	4	—	t_{Pcyc}	33.51	
	Timer clock pulse width (Low)	t_{TCLKWL}	4	—	t_{Pcyc}	33.51	
	Timer clock rise time	t_{TCLKr}	—	0.8	t_{Pcyc}	33.51	
	Timer clock fall time	t_{TCLKf}	—	0.8	t_{Pcyc}	33.51	

Note: t_{Pcyc}: one Pck cycle time

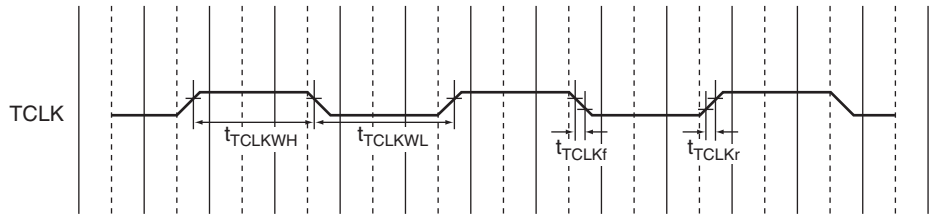


Figure 33.51 TCLK Input Timing

Table 33.11 SCIF Module Signal Timing

(V_{DDQ} = 3.0 to 3.6 V, V_{DD} = 1.5 V, T_a = –20 to 75°C/–40 to 85°C, C_L = 30 pF, PLL2 on)

Module	Item	Symbol	Min.	Max.	Unit	Figure	Notes
SCIF	Input clock cycle (asynchronous)	t_{Scyc}	4	—	t_{Pcyc}	33.52	
	Input clock cycle (synchronous)		10	—	t_{Pcyc}	33.52	
	Input clock pulse width	t_{SCKW}	0.4	0.6	t_{Scyc}	33.52	
	Input clock rise time	t_{SCKr}	—	0.8	t_{Pcyc}	33.52	
	Input clock fall time	t_{SCKf}	—	0.8	t_{Pcyc}	33.52	
	Transfer data delay time	t_{TXD}	—	3	t_{Pcyc}	33.53	
	Receive data setup time (synchronous)	t_{RXS}	3	—	t_{Pcyc}	33.53	
	Receive data hold time (synchronous)	t_{RXH}	1	—	t_{Pcyc}	33.53	

Note: t_{Pcyc}: one Pck cycle time

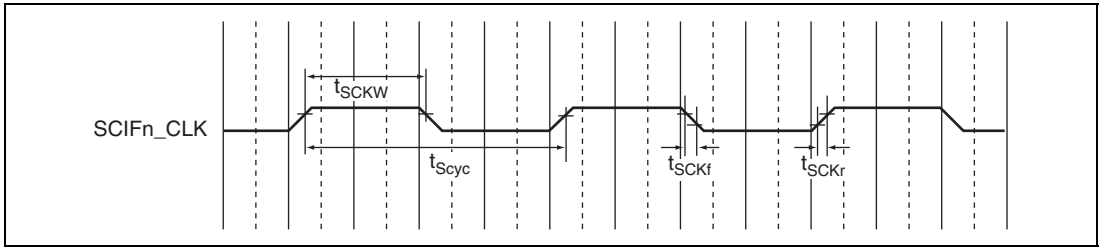


Figure 33.52 SCIFn_CLK Input Clock Timing

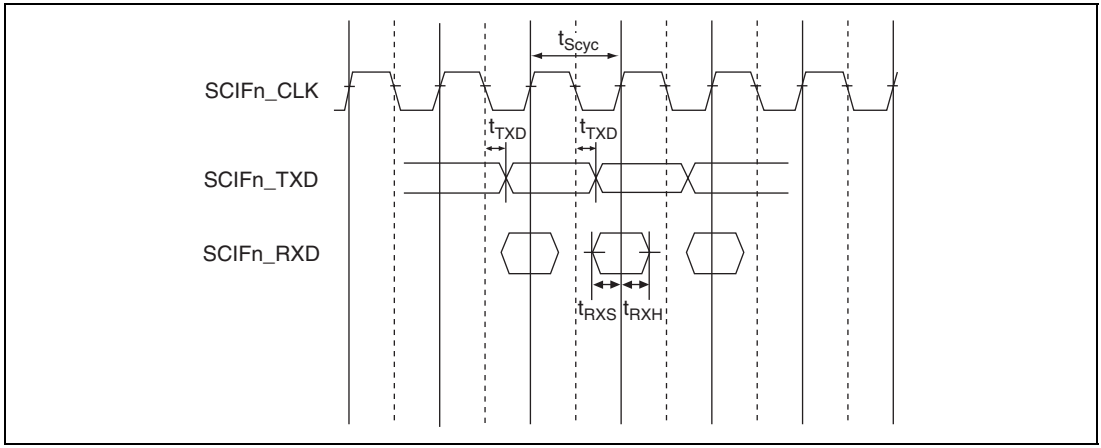


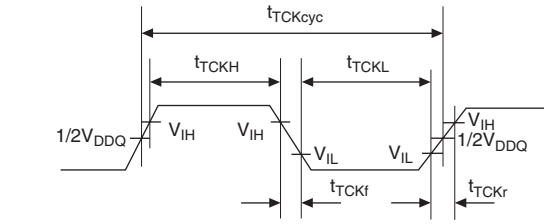
Figure 33.53 SCIF I/O Synchronous Mode Clock Timing

Table 33.12 H-UDI Module Signal Timing

(V_{DDQ} = 3.0 to 3.6 V, V_{DD} = 1.5 V, T_a = –20 to 75°C/–40 to 85°C, C_L = 30 pF, PLL2 on)

Module	Item	Symbol	Min.	Max.	Unit	Figure	Notes
H-UDI	Input clock cycle	t_{TCKcyc}	50	—	ns	33.54, 33.56	
	Input clock pulse width (High)	t_{TCKH}	15	—	ns	33.54	
	Input clock pulse width (Low)	t_{TCKL}	15	—	ns	33.54	
	Input clock rise time	t_{TCKr}	—	10	ns	33.54	
	Input clock fall time	t_{TCKf}	—	10	ns	33.54	
	$\overline{\text{ASEBRK}}$ setup time	$t_{ASEBRKS}$	10	—	t_{cyc}	33.55	
	$\overline{\text{ASEBRK}}$ hold time	$t_{ASEBRKH}$	10	—	t_{cyc}	33.55	
	TDI/TMS setup time	t_{TDIS}	15	—	ns	33.56	
	TDI/TMS hold time	t_{TDIH}	15	—	ns	33.56	
	TDO data delay time	t_{TDO}	0	10	ns	33.56	
	ASE-PINBRK pulse width	t_{PINBRK}	2	—	t_{Pcyc}	33.57	

Notes: 1. t_{cyc}: one CKIO cycle time
2. t_{Pcyc}: one Pck cycle time



Note: When clock is input from TCK pin.

Figure 33.54 TCK Input Timing

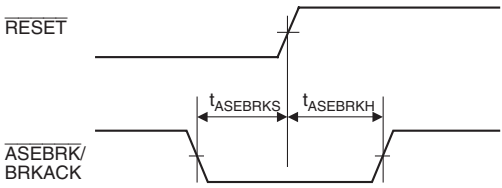


Figure 33.55 $\overline{\text{RESET}}$ Hold Timing

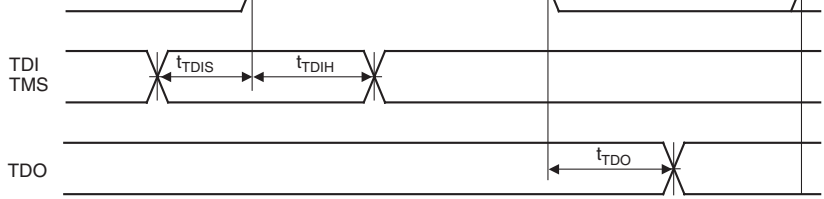


Figure 33.56 H-UDI Data Transfer Timing

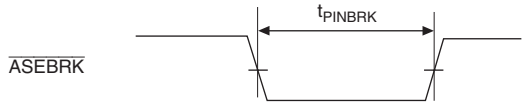


Figure 33.57 Pin Break Timing

33.3.9 CMT Module Signal Timing

Table 33.13 CMT Module Signal Timing

(V_{DDQ} = 3.0 to 3.6 V, V_{DD} = 1.5 V, T_a = –20 to 75°C/–40 to 85°C, C_L = 30 pF, PLL2 on)

Item	Symbol	Min.	Max.	Unit	Figure
CMT_CTR output delay time	t_{TMD}	—	8	ns	33.58
CMT_CTR input setup time	t_{TMS}	6	—	ns	33.58
CMT_CTR input hold time	t_{TMH}	2	—	ns	33.58
Timer clock low level width	t_{TMLOW}	1.5	—	t_{cyc}	33.59
Timer clock high level width	t_{TMHIGH}	1.5	—	t_{cyc}	33.59

Note: t_{cyc}: one CKIO cycle time

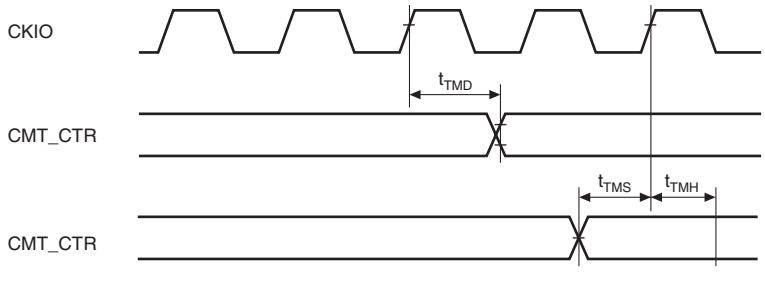


Figure 33.58 CMT Timing (1)

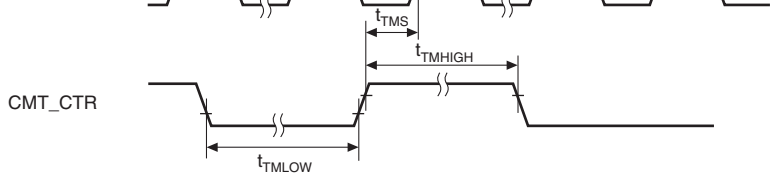


Figure 33.59 CMT Timing (2)

33.3.10 HCAN2 Module Signal Timing

Table 33.14 HCAN2 Module Signal Timing

(V_{DDQ} = 3.0 to 3.6 V, V_{DD}= 1.5 V, T_a = –20 to 75°C/–40 to 85°C, C_L = 30 pF, PLL2 on)

Item	Symbol	Min.	Max.	Unit	Figure
CAN_TX output delay time	t_{CAND}	—	6	ns	33.60
CAN_RX input setup time	t_{CANS}	4	—	ns	33.60
CAN_RX input hold time	t_{CANH}	2.5	—	ns	33.60

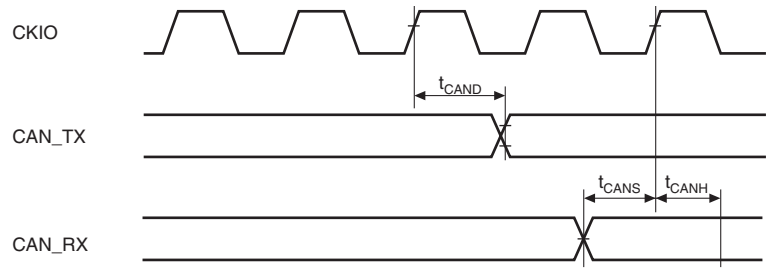


Figure 33.60 HCAN2 Timing

Table 33.15 GPIO Signal Timing

(V_{DDQ} = 3.0 to 3.6 V, V_{DD} = 1.5 V, T_a = –20 to 75°C/–40 to 85°C, C_L = 30 pF, PLL2 on)

Item	Symbol	Min.	Max.	Unit	Figure
GPIO output delay time	t_{IOPD}	—	9	ns	33.61
GPIO input setup time	t_{IOPS}	7	—	ns	33.61
GPIO input hold time	t_{IOPH}	5	—	ns	33.61

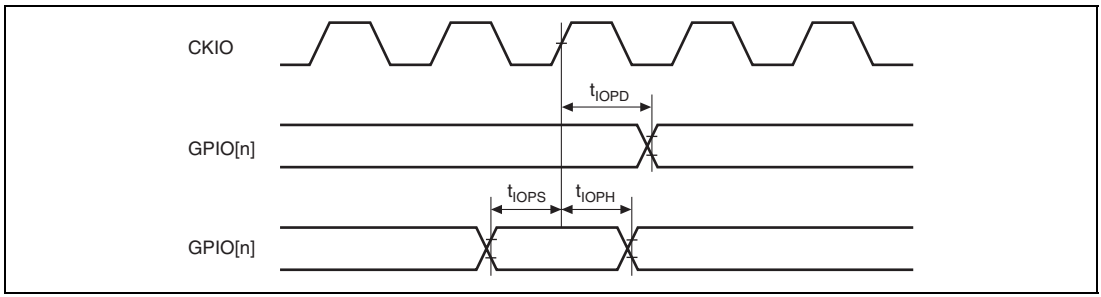


Figure 33.61 GPIO Timing

(1) I²C block diagram and truth table

Figure 33.62 shows a block diagram of I²C I/O buffer and Table 33.16 shows a truth table.

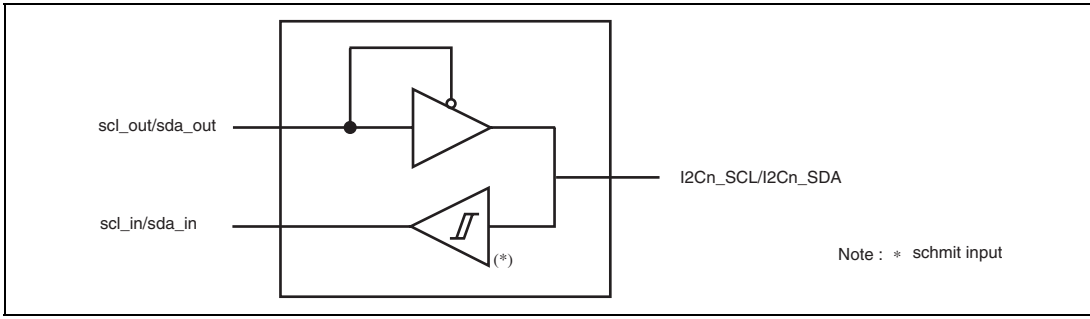


Figure 33.62 Block Diagram of I²C I/O Buffer

Table 33.16 Truth Table of I²C I/O Buffer

scl_out/sda_out	I2Cn_SCL/ I2Cn_SDA	scl_in/sda_in
1	Z	H/L level of I2Cn_SCL/ I2Cn_SDA
0	0	H/L level of I2Cn_SCL/ I2Cn_SDA

Notes: 1. The Output voltages from I2Cn_SCL/ I2Cn_SDA are undefined until the internal logic will be stable after power on.
2. If a external pull down resistance is connected to a 5V tolerant buffer, the value of the resistance must be less than 15 kΩ.

(2) I²C DC characteristics

Table 33.17 I²C DC Characteristics

(V_{DDQ} = 3.0 to 3.6 V, V_{DD} = 1.5 V, T_a = –20 to 75°C/–40 to 85°C, C_L = 30 pF, PLL2 on)

Item	symbol	Min.	Max.	Unit	Test Condition
Input Voltage	V_{IH}	$V_{DDQ} \times 0.7$	5.5	V	
	V_{IL}	–0.5	$V_{DDQ} \times 0.3$	V	
Output Voltage	V_{OL}	0	0.4	V	I_{OL} = 3 mA
Input Current	I_i	–10	10	μA	Input Voltage = $0.1 \times V_{DDQ}$ to $0.9 \times V_{DDQ}$ (max.)

Table 33.18 I²C Bus Interface Module Signal Timing

(V_{DDQ} = 3.0 to 3.6 V, V_{DD} = 1.5 V, T_a = –20 to 75°C/–40 to 85°C, C_L = 30 pF, PLL2 on)

Item	Symbol	Min.	Typ.	Max.	Unit
I2Cn_SCL frequency	t_{Icyc}	—	—	400	kHz
I2Cn_SCL low level pulse width	t_{ICWL}	1.3	—	—	µs
I2Cn_SCL high level pulse width	t_{ICWH}	0.6	—	—	µs
I2Cn_SCL/I2Cn_SDA rise time	t_{ICr}	20 +0.1 Cb*	—	300	ns
I2Cn_SCL/I2Cn_SDA fall time	t_{ICf}	20 +0.1 Cb*	—	300	ns
I2Cn_SDA bus free time	t_{ICBF}	1.3	—	—	µs
I2Cn_SCL start condition hold time	t_{ICH}	0.6	—	—	µs
I2Cn_SCL resend start condition setup time	t_{ICS}	0.6	—	—	µs
I2Cn_SDA stop condition setup time	t_{ICST}	0.6	—	—	µs
I2Cn_SDA setup time	t_{DAS}	100	—	—	ns
I2Cn_SDA hold time	t_{ICDH}	0	—	0.9	µs

Notes: t_{Pcyc}: one Pck cycle time

* Cb is the total capacity of one bus line. (max. 400 pF)

(4) I²C Schmitt characteristics

Table 33.19 I²C Schmitt characteristics

(V_{DDQ} = 3.0 to 3.6 V, V_{DD} = 1.5 V, T_a = –20 to 75°C/–40 to 85°C, C_L = 30 pF, PLL2 on)

Item	Symbol	Min.	Max.	Unit	Notes
Threshold voltage	VTT+	$V_{DDQ} \times 0.7$	—	V	Threshold voltage: L → H
	VTT–	—	$V_{DDQ} \times 0.3$	V	Threshold voltage: H → L
	ΔVTT	$V_{DDQ} \times 0.05$	—	V	reference value between VTT+ and VTT–

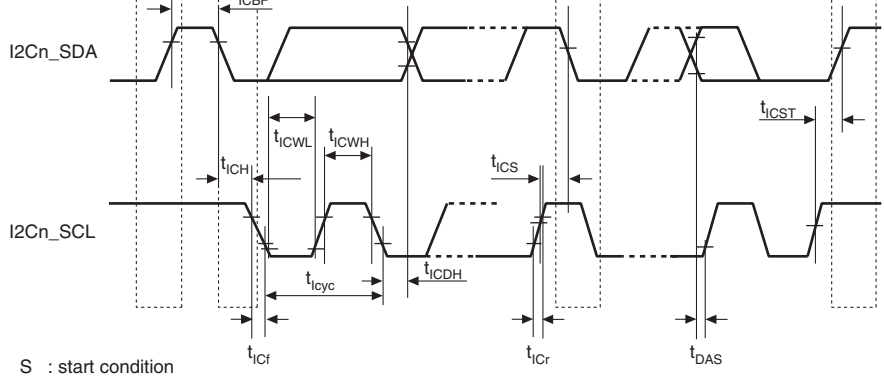


Figure 33.63 I²C Bus Interface Module Signal Timing

(5) Notes on Usage of I²C

- While 5 V interface signal is input to this LSI, the power must be supplied to this LSI.

  If the power supply to SH-4 is switched off, the power supply to the bus pull-up resistance must be switched off.

  The I/O buffer for I²C is 5V tolerant, if the power is supplied to this LSI.

  The I/O buffer may be destroyed, if the 5V interface signal is input while the power supply to this LSI is switched off.
- If an external pull-down resistance is connected to the 5 V tolerant buffer, the value of the resistance must be less than 15 kΩ.

Table 33.20 HSPI Module Signal Timing

(V_{DDQ} = 3.0 to 3.6 V, V_{DD} = 1.5 V, T_a = –20 to 75°C/–40 to 85°C, C_L = 30 pF, PLL2 on)

Item	Symbol	Min.	Max.	Unit	Figure
HSPI_CLK frequency	t_{SPIcyc}	—	Pck/8	Hz	33.64
HSPI clock high level width	t_{SPIHW}	60	—	ns	33.64
HSPI clock low level width	t_{SPILW}	60	—	ns	33.64
HSPI_TX setup time	$t_{SUSPITX}$	—	20	ns	33.64
HSPI_TX delay time	t_{DSPITX}	—	20	ns	33.64
HSPI_RX setup time	$t_{SUSPIRX}$	20	—	ns	33.64
HSPI_RX hold time	$t_{HLSPIRX}$	20	—	ns	33.64
HSPI_CS read time	t_{CSLEAD}	100	—	ns	33.64

Note: Pck: Peripheral clock frequency

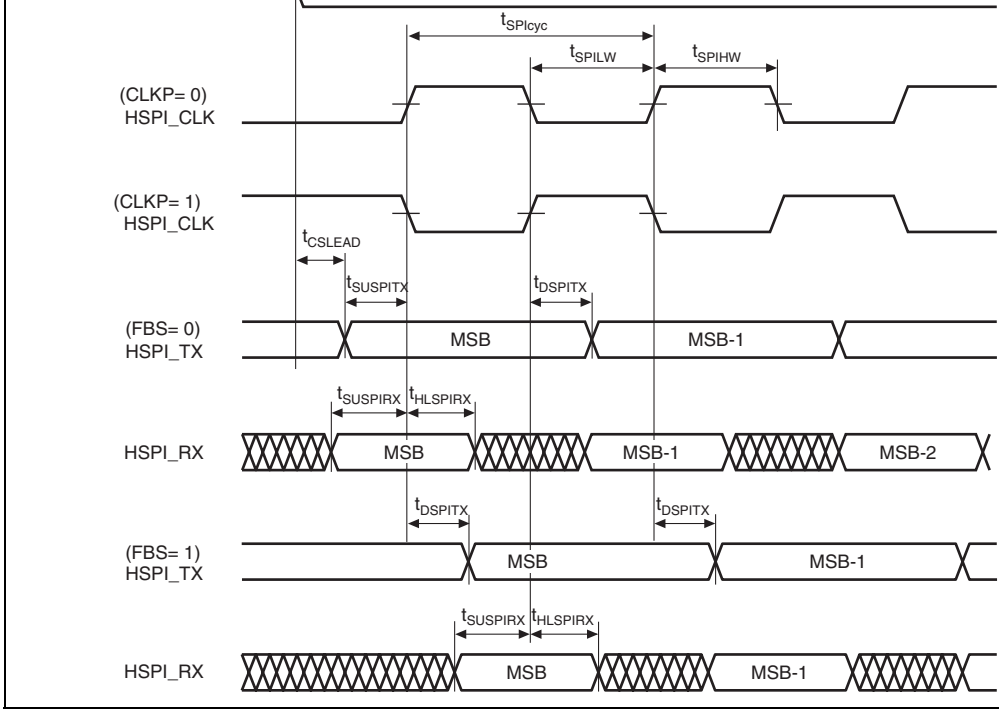


Figure 33.64 HSPI Data Output/Input Timing

(1) USB DC characteristics

Table 33.21 USB DC characteristics

(V_{DDQ} = 3.0 to 3.6 V, V_{DD} = 1.5 V, T_a = –20 to 75°C/–40 to 85°C)

Item	Symbol	Min.	Max.	Unit	Test condition	Figure
Input Voltage	V_{IH}	V_{DDQ} × 0.7	V_{DDQ} +0.3	V		33.65
	V_{IL}	–0.3	V_{DDQ} × 0.2	V		33.65
Output Voltage	V_{OH}	V_{DDQ} × 0.8	—	V	14.25 kΩ-GND	33.66
	V_{OL}	—	V_{DDQ} × 0.2	V	1.425 kΩ-V_{DDQ}	33.66
Differential input sensitivity	VDI	0.2	—	V	|USB_DP–USB_DM|	33.67
Differential common mode range	VCM	0.8	2.5	V		33.67
Output resistance*	ZDRV	28	44	Ω		33.68

Note: * It includes the external resistance value. The recommended external resistance value is 27 Ω ±1%.

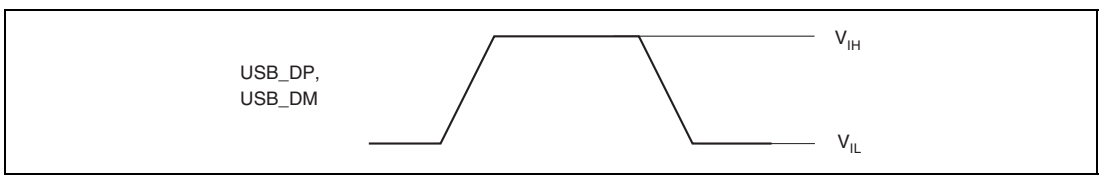


Figure 33.65 Input Voltage (V_{IH}, V_{IL})

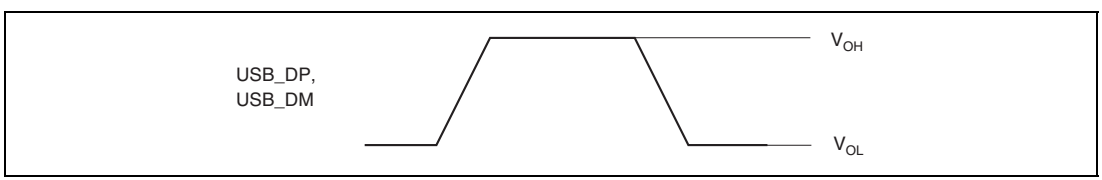


Figure 33.66 Output (V_{OH}, V_{OL})

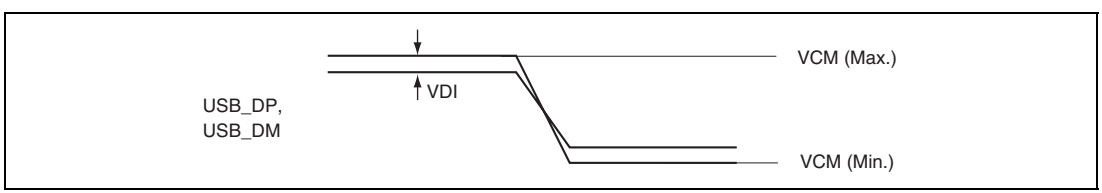


Figure 33.67 Differential Input Sensitivity (VDI), Differential common mode range (VCM)

Table 33.22 USB AC characteristics

(V_{DDQ} = 3.0 to 3.6 V, V_{DD} = 1.5 V, T_a = –20 to 75°C/–40 to 85°C, C_L = 30 pF, PLL2 on)

			Full speed*1		Low speed*2			
Item		**Symbol**	**Min.**	**Max.**	**Min.**	**Max.**	**Unit**	**Notes**
Transition Time	Rise*3	t_r	4	20	75	300	ns	
	Fall*3	t_f	4	20	75	300	ns	
Rise/Fall time matching		t_{RFM}	80	111.1	80	125	%	t_r / t_f
D+D– crossover voltage		V_{CRS}	1.3	2.4	1.1	2.0	V	

Notes: 1. Please refer Figure 33.68 about the test condition.
2. Please refer Figure 33.69 about the test condition.
3. Transition time from 10% level to 90% level (Figure 33.70).

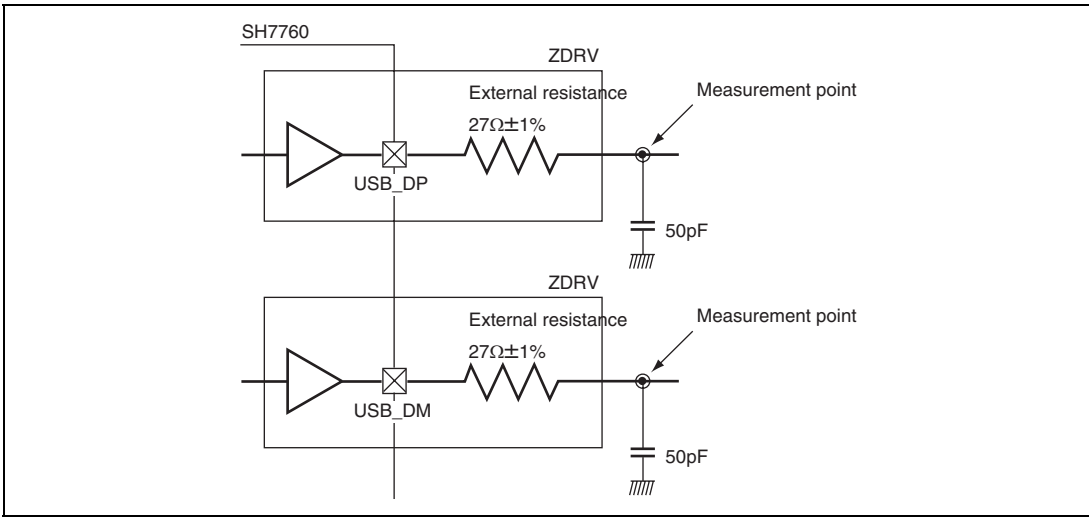


Figure 33.68 Load Condition of AC Characteristics (Full speed)

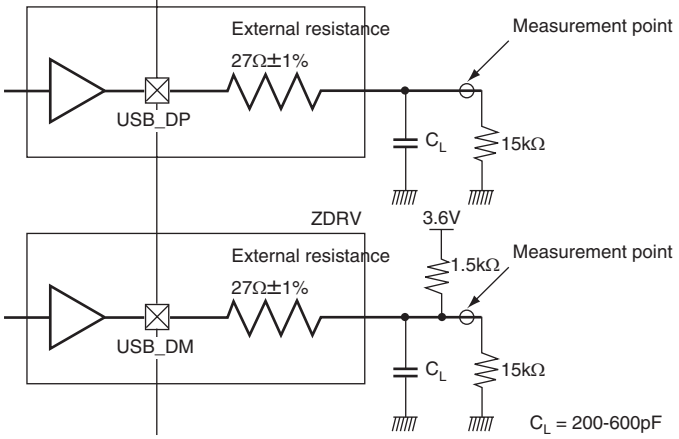


Figure 33.69 Load Condition of AC Characteristics (Low speed)

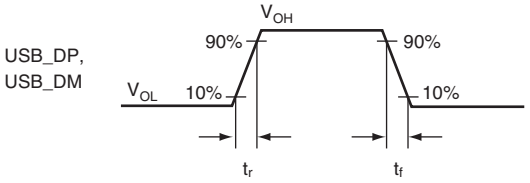


Figure 33.70 t_r, t_f

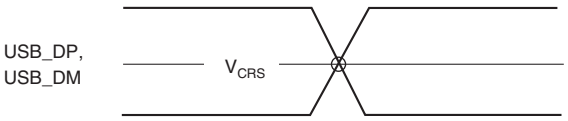


Figure 33.71 V_{CRS}

(1) MFI AC characteristics

Figure 33.72 shows the AC characteristics of 68 series bus. Figure 33.73 and 33.74 show the AC characteristics of 80 series bus.

Table 33.23 AC Characteristics of 68 Series Bus

(V_{DDQ} = 3.0 to 3.6 V, V_{DD} = 1.5 V, T_a = –20 to 75°C/–40 to 85°C, C_L = 30 pF, PLL2 on)

Item	Symbol	Min.	Max.	Unit	Figure
Read bus cycle time	$t_{MFICYCR}$	$4 \times t_{Pcyc} + 10$	—	ns	33.72
Write bus cycle time	$t_{MFICYCW}$	$3 \times t_{Pcyc} + 10$	—	ns	
Address setup time (MFI-RS)	t_{MFIAS}	0	—	ns	
(MFI-RW/$\overline{RD}$)	t_{MFIAS}	10	—	ns	
Address hold time (MFI-RS)	t_{MFIAH}	0	—	ns	
(MFI-RW/$\overline{RD}$)	t_{MFIAH}	10	—	ns	
Enable high width (Read)	t_{MFIWRH}	$2.5 \times t_{Pcyc}$	—	ns	
Enable low width (Write)	t_{MFIWEH}	$1.5 \times t_{Pcyc}$	—	ns	
Enable low width	t_{MFIWEL}	$2.0 \times t_{Pcyc} + 5$	—	ns	
Read data delay time	t_{MFIRDD}	—	$2 \times t_{Pcyc} + 10$	ns	
Read data hold time	t_{MFIRDH}	0	—	ns	
Write data setup time	t_{MFIWDS}	$t_{Pcyc} + 10$	—	ns	
Write data hold time	t_{MFIWDH}	10	—	ns	

Notes: 1. t_{Pcyc}: one Pck cycle time
2. t_{MFIWEH} is the time where the low level of $\overline{MFI\text{-}CS}$ and the high level of MFI-E/$\overline{WR}$ are overlapped.

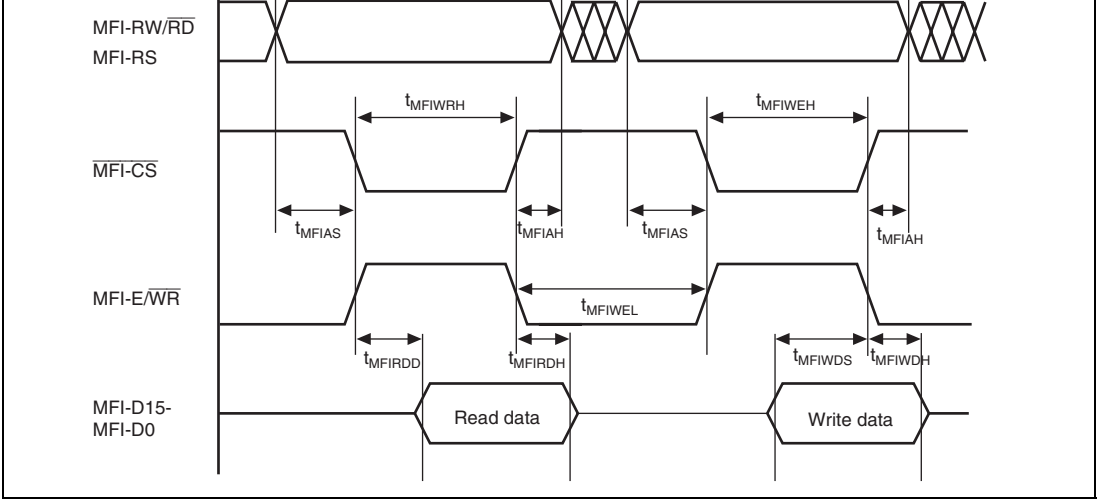


Figure 33.72 AC Characteristics of 68 Series Bus

Table 33.24 AC Characteristics of 80 Series Bus

(V_{DDQ} = 3.0 to 3.6 V, V_{DD} = 1.5 V, T_a = –20 to 75°C/–40 to 85°C, C_L = 30 pF, PLL2 on)

Item	Symbol	Min.	Max.	Unit	Figure
Read bus cycle time	$t_{MFICYCR}$	$4 \times t_{Pcyc}$	—	ns	33.73, 33.74
Write bus cycle time	$t_{MFICYCW}$	$3 \times t_{Pcyc}$	—	ns	
Address setup time	t_{MFIAS}	0	—	ns	
Address hold time	t_{MFIAH}	0	—	ns	
Read low width (Read)	t_{MFIWRL}	$2.5 \times t_{Pcyc}$	—	ns	
Write low width (Write)	t_{MFIWWL}	$1.5 \times t_{Pcyc}$	—	ns	
Read/Write high width	$t_{MFIWRWH}$	$2.0 \times t_{Pcyc} + 5$	—	ns	
Read data delay time	t_{MFIRDD}	—	$2 \times t_{Pcyc} + 10$	ns	
Read data hold time	t_{MFIRDH}	0	—	ns	
Write data setup time	t_{MFIWDS}	$t_{Pcyc} + 10$	—	ns	
Write data hold time	t_{MFIWDH}	10	—	ns	

Notes: 1. $t_{Pcyc:}$ one Pck cycle time
2. t_{MFIWRL} is the time where the low level of $\overline{\text{MFI-CS}}$ and the low level of MFI-RW/$\overline{\text{RD}}$ are overlapped.
3. t_{MFIWWL} is the time where the low level of $\overline{\text{MFI-CS}}$ and the low level of MFI-E/$\overline{\text{WR}}$ are overlapped.

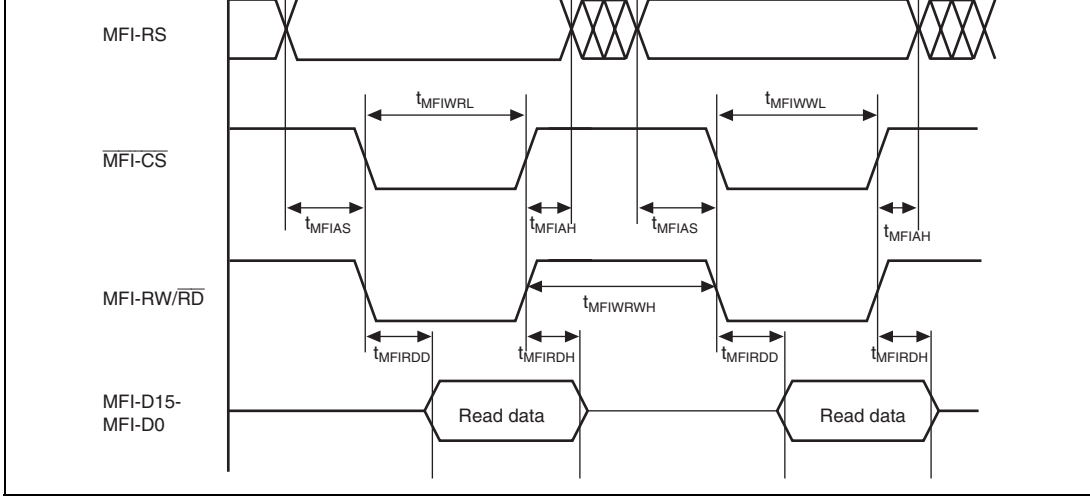


Figure 33.73 AC Characteristics of 80 Series Bus (Read)

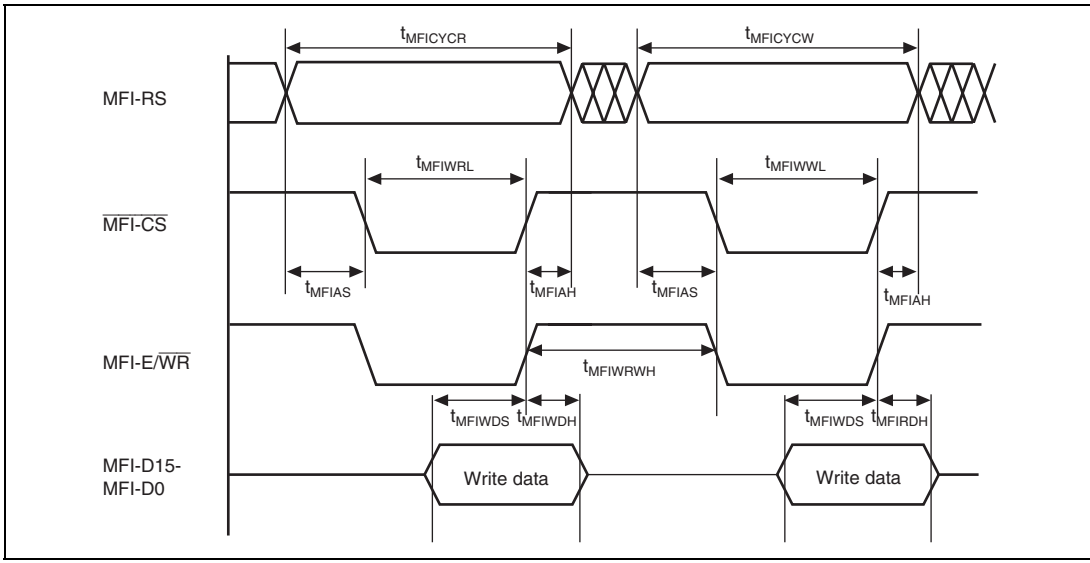


Figure 33.74 AC Characteristics of 80 Series Bus (Write)

Table 33.25 SIM Module Signal Timing

(V_{DDQ} = 3.0 to 3.6 V, V_{DD} = 1.5 V, T_a = –20 to 75°C/–40 to 85°C, C_L = 30 pF, PLL2 on)

Item	Symbol	Min.	Max.	Unit	Figure
SIM_CLK clock cycle time	t_{SMcyc}	t_{Pcyc}/16	t_{Pcyc}/2	ns	33.75
SIM_CLK clock high level width	t_{SMCWH}	0.4 t_{SMcyc}	—	ns	
SIM_CLK clock low level width	t_{SMCWL}	0.4 t_{SMcyc}	—	ns	
SIM_RST reset output delay time	t_{SMRD}	0	20	ns	

Note: t_{Pcyc}: one Pck cycle time

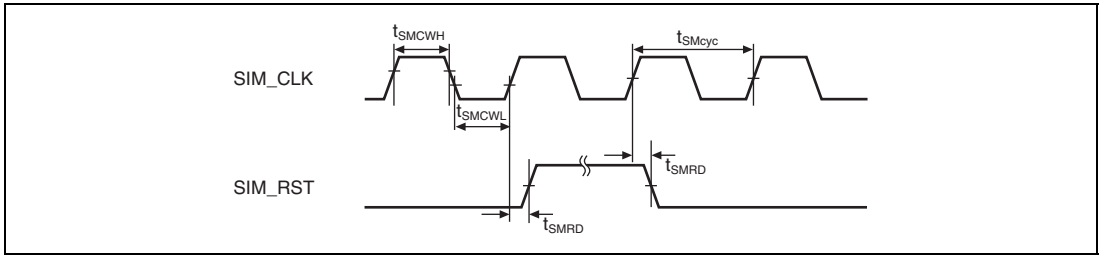


Figure 33.75 SIM Module Signal Timing

33.3.17 MMCIF Module Signal Timing

Table 33.26 MMCIF Module Signal Timing

(V_{DDQ} = 3.0 to 3.6 V, V_{DD} = 1.5 V, T_a = –20 to 75°C/–40 to 85°C, C_L = 30 pF, PLL2 on)

Item	Symbol	Min.	Max.	Unit	Figure
MCCLK clock cycle time	t_{MMcyc}	50	—	ns	33.76
MCCLK clock high level width	t_{MMWH}	0.4 × t_{MMcyc}	—	ns	33.76
MCCLK clock low level width	t_{MMWL}	0.4 × t_{MMcyc}	—	ns	33.76
MCCMD output data delay time	t_{MMTCD}	—	10	ns	33.76
MCCMD input data setup time	t_{MMRCS}	10	—	ns	33.77
MCCMD input data hold time	t_{MMRCH}	10	—	ns	33.77
MCDAT output data delay time	t_{MMTDD}	—	10	ns	33.76
MCDAT input data setup time	t_{MMRDS}	10	—	ns	33.77
MCDAT input data hold time	t_{MMRDH}	10	—	ns	33.77

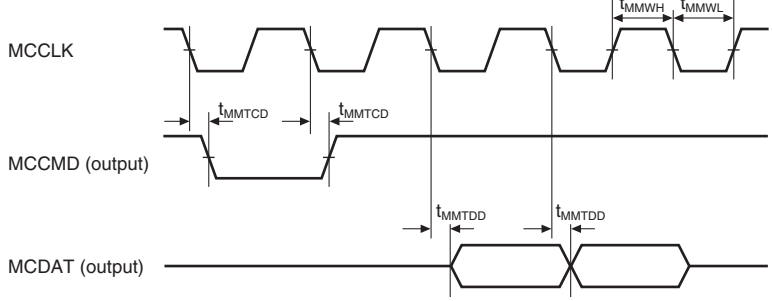


Figure 33.76 MMCIF Transmit Timing

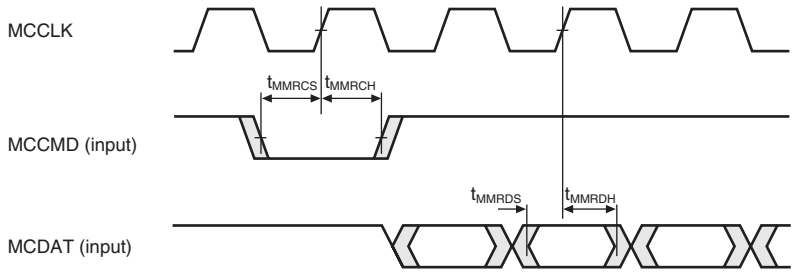


Figure 33.77 MMCIF Receive Timing

Table 33.27 LCDC Module Signal Timing

(V_{DDQ} = 3.0 to 3.6 V, V_{DD} = 1.5 V, T_a = –20 to 75°C/–40 to 85°C, C_L = 30 pF, PLL2 on)

Item	Symbol	Min.	Max.	Unit	Figure
LCD_CLK input clock frequency	t_{FREQ}	—	50	MHz	
LCD_CLK input clock rise time	t_r	—	3	ns	
LCD_CLK input clock fall time	t_f	—	3	ns	
LCD_CLK input clock duty	t_{DUTY}	90	110	%	
Clock (LCD_CL2) cycle time	t_{CC}	25	—	ns	33.78
Clock (LCD_CL2) High level pulse width	t_{CHW}	7	—	ns	
Clock (LCD_CL2) Low level pulse width	t_{CLW}	7	—	ns	
Clock (LCD_CL2) transition time (rise/fall)	t_{CT}	—	3	ns	
Data (LCD_DATA) delay time	t_{DDdo}	–3.5	3	ns	
Display Enable (LCD_M_DISP) delay time	t_{IDdo}	–3.5	3	ns	
Horizontal sync signal (LCD_CL1) delay time	t_{HDdo}	–3.5	3	ns	
Vertical sync signal (LCD_FLM) delay time	t_{VDdo}	–3.5	3	ns	

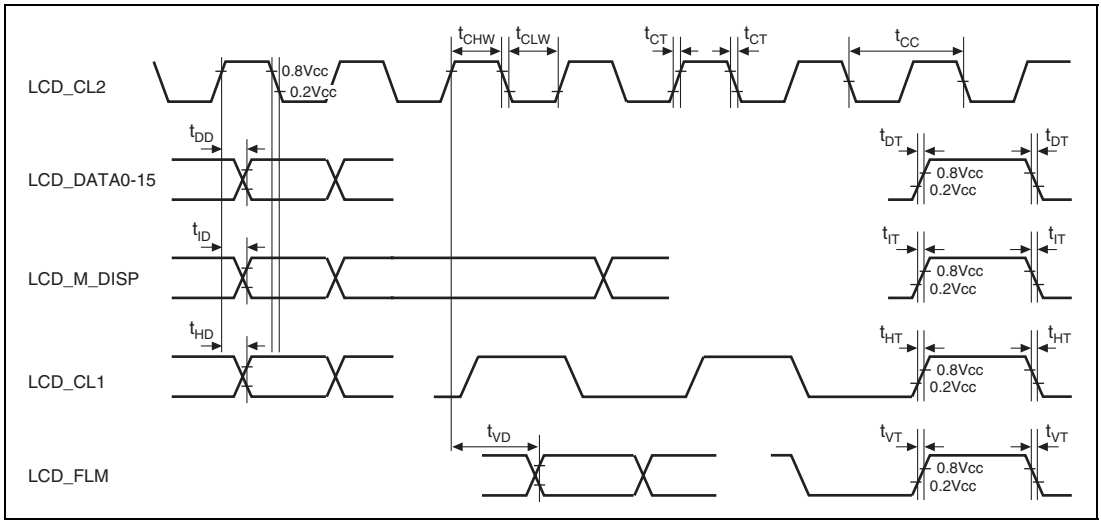


Figure 33.78 LCDC Module Signal Timing

Table 33.28 HAC Interface Module Signal Timing

(V_{DDQ} = 3.0 to 3.6 V, V_{DD}= 1.5 V, T_a = –20 to 75°C/–40 to 85°C, C_L = 30 pF, PLL2 on)

Item	Symbol	Min.	Max.	Unit	Figure
$\overline{\text{HAC\_RES}}$ active low pulse width	t_{RST_LOW}	1000	—	ns	33.79
HAC_SYNC active high pulse width	t_{SYN_HIGH}	1000	—	ns	33.80
HAC_SYNC delay time 1	t_{SYNCD1}	—	15	ns	33.82
HAC_SYNC delay time 2	t_{SYNCD2}	—	15	ns	33.82
HAC_SD_OUT delay time	t_{SDOUTD}	—	15	ns	33.82
HAC_SD_IN setup time	t_{SDINS}	10	—	ns	33.82
HAC_SD_IN hold time	t_{SDINH}	10	—	ns	33.82
HAC_BIT_CLK input high level width	t_{ICL_HIGH}	2 × t_{Pcyc}	—	ns	33.81
HAC_BIT_CLK input low level width	t_{ICL_LOW}	2 × t_{Pcyc}	—	ns	33.81

Note: t_{Pcyc}: one Pck cycle time

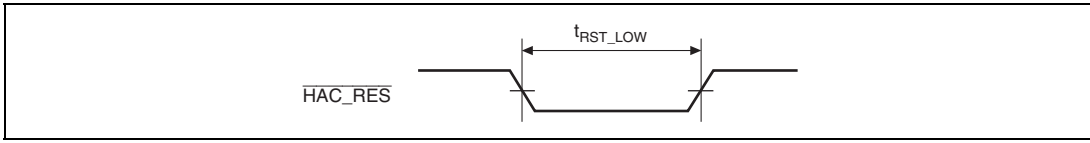


Figure 33.79 HAC Cold Reset Timing

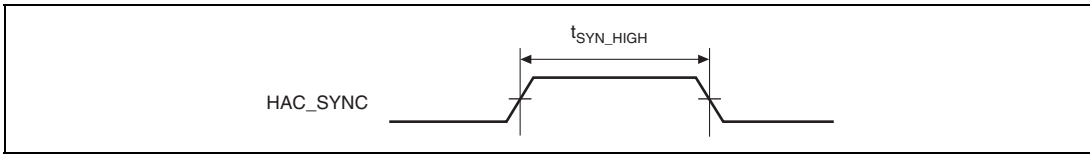


Figure 33.80 HAC Cold Reset Timing

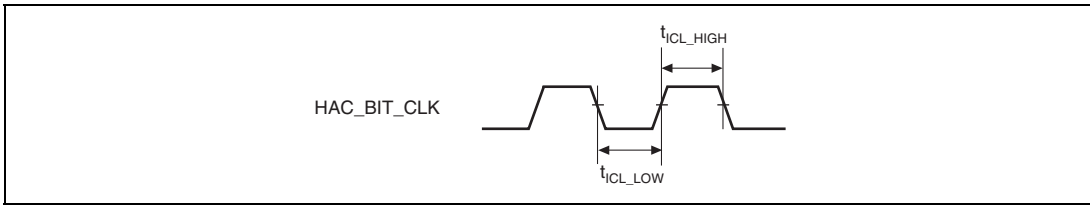


Figure 33.81 HAC Clock Input Timing

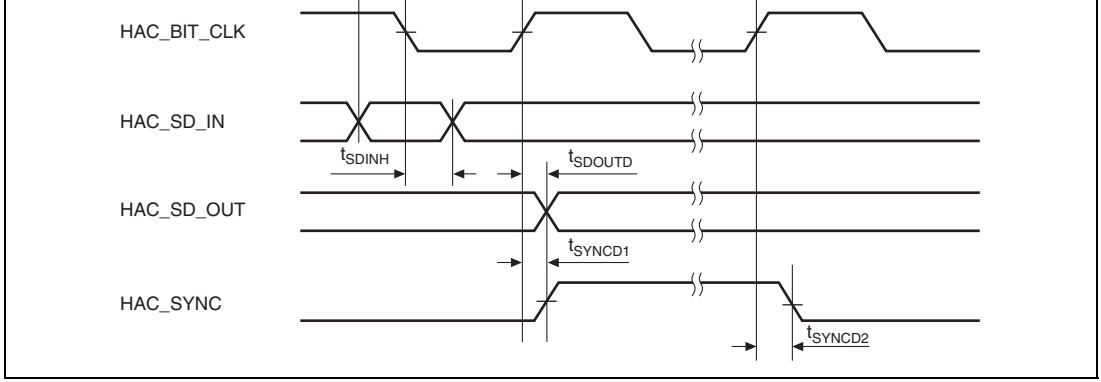


Figure 33.82 HAC Interface Module Signal Timing

33.3.20 SSI Interface Module Signal Timing

Table 33.29 SSI Interface Module Signal Timing

(V_{DDQ} = 3.0 to 3.6 V, V_{DD}= 1.5 V, T_a = –20 to 75°C/–40 to 85°C, C_L = 30 pF, PLL2 on)

Item	Symbol	Min.	Max.	Unit	Notes	Figure
Output cycle time	t_{OSCK}	40	710	ns	output	33.83
Input cycle time	t_{ISCK}	80	3300	ns	input	33.83
Input high level width/Output high level width	t_{IHC}/t_{OHC}	30	—	ns	input, output	33.83
Input low level width/Output low level width	t_{ILC}/t_{OLC}	20	—	ns	input, output	33.83
SCK Output rise time	t_{RC}	—	60	ns	output	33.83
SDATA Output delay time	t_{DTR}	—	50	ns	transmit	33.84, 33.85
SDATA/WS Input setup time	t_{SR}	10	—	ns	receive	33.86, 33.87
SDATA/WS Input hold time	t_{HTR}	10	—	ns	receive	33.86, 33.87

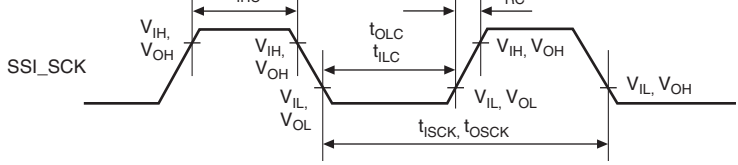


Figure 33.83 SSI Clock Input/Output Timing

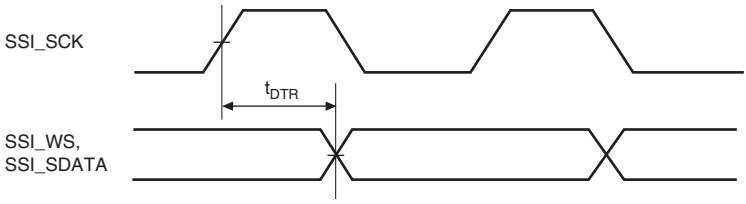


Figure 33.84 SSI Transmit Timing (1)

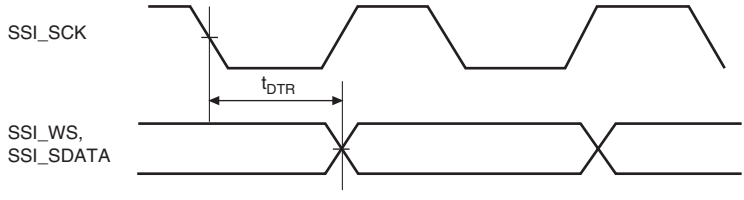


Figure 33.85 SSI Transmit Timing

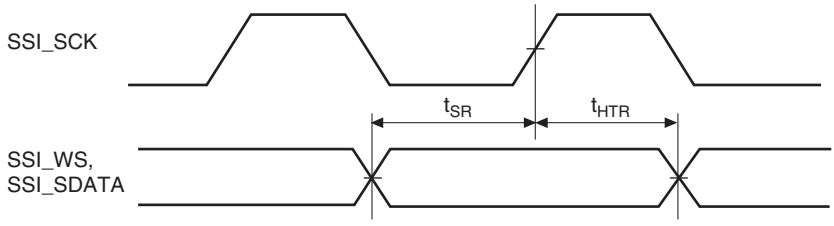


Figure 33.86 SSI Receive Timing (1)

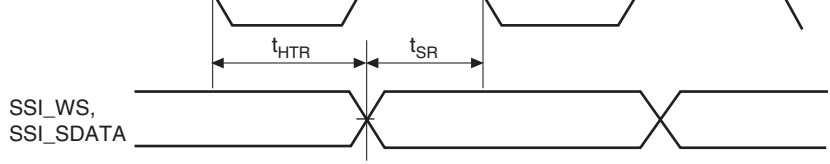


Figure 33.87 SSI Receive Timing (2)

33.4 A/D Converter Characteristics

Table 33.30 shows A/D converter characteristics.

Table 33.30 A/D Converter Characteristics

(T_a = 25°C)

Item	Min.	Max.	Unit
Resolution	10	10	bits
Conversion time (single mode)	8	—	μs
Permitted analog signal source impedance	—	5	kΩ
Non-linear error	—	±4.0	LSB
Offset error	—	±2.0	LSB
Full-scale error	—	±2.0	LSB
Quantization error	—	±0.5	LSB
Absolute error	—	±4.0	LSB

Notes: The input voltage to A/D converter is as follows.

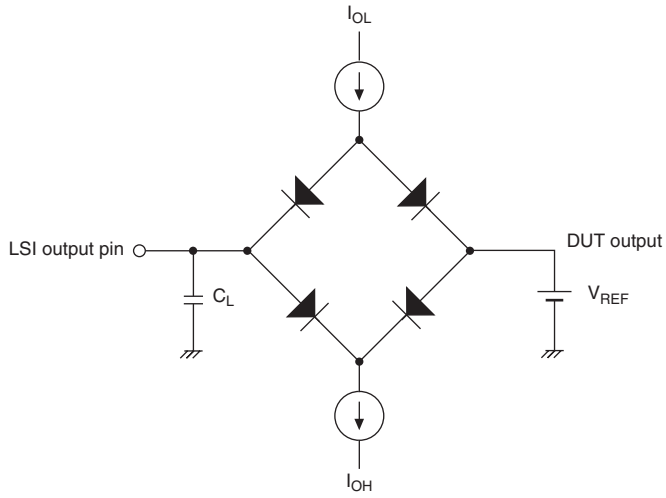
$AV_{SS\_ADC} \leq ANn \leq AV_{CC\_ADC}$

1. ANn: A/D converter
2. AV_{SS_ADC} = GND
3. AV_{CC_ADC} = 3.0 V (Min.), 3.3 V (Typ.), 3.6 V (Max.)
4. n = 0 to 3

The AC characteristic test conditions are as follows:

- Input/output signal reference level: 1.5 V (V_{DDQ} = 3.3 ±0.3V)
- Input pulse level: V_{SSQ} to 3.0 V (V_{SSQ} to V_{DDQ} for ($\overline{\text{RESET}}$, $\overline{\text{TRST}}$, NMI, $\overline{\text{ASEBRK}}$/BRKACK, $\overline{\text{MRESET}}$, CA, $\overline{\text{SCIF2\_RTS}}$, USB_PENC, VEPWC/$\overline{\text{IRQ5}}$, VCPWC/$\overline{\text{IRQ4}}$, $\overline{\text{IRL3}}$, $\overline{\text{IRL2}}$, $\overline{\text{IRL1}}$, $\overline{\text{IRL0}}$, Reserved/AUDATA[3])
- Input rise/fall time: 1 ns

The output load circuit is shown in figure 33.88



Notes: 1. C_L is the total value, including the capacitance of the test jig,etc.
The capacitance of each pin is set to 30 pF.
2. I_{OL} and I_{OH} values are as shown in table 33.3, Permissible Output Currents.

Figure 33.88 Output Load Circuit

Figure 33.89 is a chart showing the changes in the delay time (reference data) when a load capacitance equal to or larger than the stipulated value (30 pF) is connected to the LSI pins. When connecting an external device with a load capacitance exceeding the regulation, use the chart in figure 33.89 as reference for system design.

Note that if the load capacitance to be connected exceeds the range shown in figure 33.89 the graph will not be a straight line.

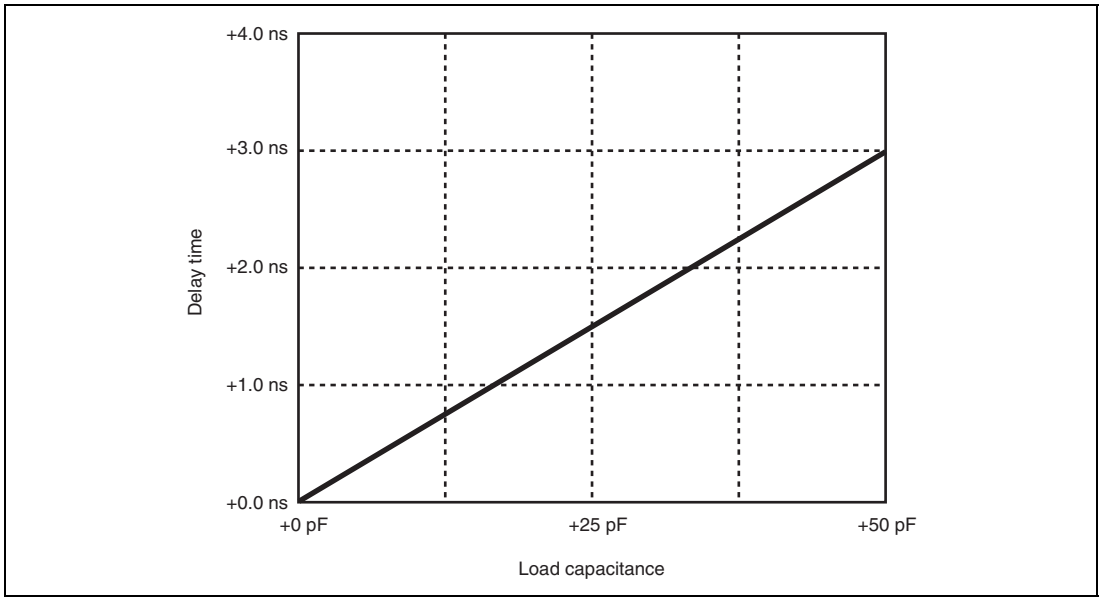
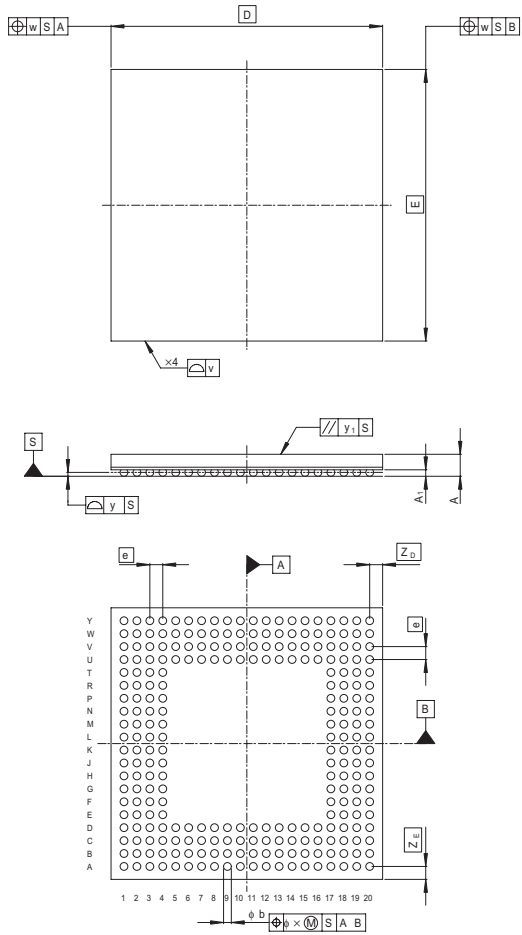


Figure 33.89 Load Capacitance-Delay Time

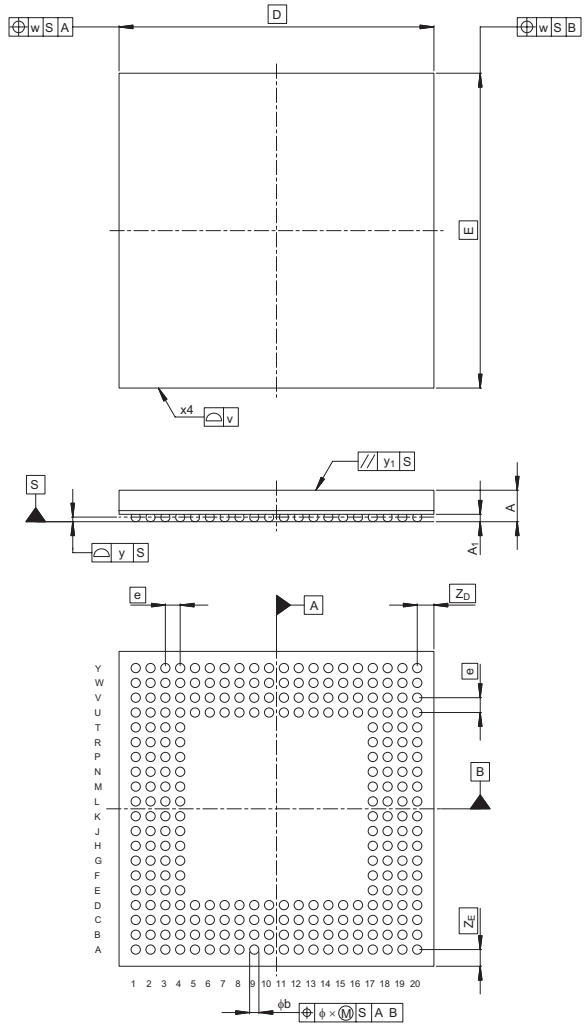
A. Package Dimensions

JEITA Package Code	RENESAS Code	Previous Code	MASS[Typ.]
P-LBGA256-21x21-1.00	PLBG0256FA-A	BP-256F/BP-256FV	1.2g



Reference Symbol	Dimension in Millimeters		
	Min	Nom	Max
D	—	21.0	—
E	—	21.0	—
v	—	—	0.20
w	—	—	0.30
A	—	—	1.70
A_1	0.4	0.5	0.6
e	—	1.00	—
b	0.50	0.60	0.70
x	—	—	0.10
y	—	—	0.15
y_1	—	—	0.35
S_D	—	—	—
S_E	—	—	—
Z_D	—	1.00	—
Z_E	—	1.00	—

Figure A.1 Package Dimensions (BP-256F/BP-256FV)



Reference Symbol	Dimension in Millimeters		
	Min	Nom	Max
D	—	17.0	—
E	—	17.0	—
v	—	—	0.15
w	—	—	0.20
A	—	—	1.70
A_1	0.35	0.40	0.45
e	—	0.80	—
b	0.45	0.50	0.55
x	—	—	0.08
y	—	—	0.10
y_1	—	—	0.2
S_D	—	—	—
S_E	—	—	—
Z_D	—	0.9	—
Z_E	—	0.9	—

Figure A.2 Package Dimensions (BP-256B/BP-256BV)

The MD8–MD0 pin values are input in the event of a power-on reset via the $\overline{\text{RESET}}$ pin.

Table B.1 Clock Operating Modes (SH7760)

Clock Operating Mode	External Pin Combination					Frequency (vs. Input Clock)			
	MD2	MD1	MD0	PLL1	PLL2	CPU Clock	Bus Clock	Peripheral Module Clock	FRQCR Initial Value
0	0	0	0	On (×12)	On	12	3	3	H'0E1A
1			1	On (×12)	On	12	3/2	3/2	H'0E2C
2		1	0	On (×6)	On	6	2	1	H'0E13
3			1	On (×12)	On	12	4	2	H'0E13
4	1	0	0	On (×6)	On	6	3	3/2	H'0E0A
6		1	0	Off (×6)	Off	1	1/2	1/2	H'0808

Notes: 1. The multiplication factor of PLL1 is solely determined by the clock operating mode.

2. For the ranges input clock frequency, see the description of the EXTAL clock input frequency (f_{EX}) and the CKIO clock output (f_{OP}) in section 33.3.1, Clock and Control Signal Timing.

Table B.2 Area 0 Memory Map and Bus Width

Pin Value				
MD6	MD4	MD3	Memory Type	Bus Width
0	0	0	Setting prohibited	Setting prohibited
		1	Setting prohibited	Setting prohibited
	1	0	Setting prohibited	Setting prohibited
		1	MPX interface	32 bits
1	0	0	Setting prohibited	Setting prohibited
		1	SRAM interface	8 bits
	1	0	SRAM interface	16 bits
		1	SRAM interface	32 bits

Pin Value	
MD5	**Endian**
0	Big endian
1	Little endian

Table B.4 MFI Mode/LCD Mode

Pin Value	
MD7	**MFI Mode/LCD Mode**
0	MFI mode
1	LCD mode

Table B.5 Clock Input

Pin Value	
MD8	**Clock Input**
0	External input clock
1	Crystal resonator

C.1 Pin States

Table C.1 Pin States in Reset, Power-Down State, and Bus-Released State

Signal Name	Pin Name	I/O	Reset		Sleep	Standby		Bus Release
			Power-on	Manual		Software	Hardware	
$\overline{RDY}$*9		I	PI	I	I	Z	Z	I
$\overline{CS0}$ – $\overline{CS6}$		O	H	H	O*15	Z*5/H*3	Z	Z*5
$\overline{BS}$		O	H	H	O*15	Z*5/H*3	Z	Z*5
D0 – D31		I/O	Z	Z*6	Z*6	Z*6	Z	Z
A0 – A25		O	PZ	Z*5/O*4	O	Z*5/O*3	Z	Z*5
$\overline{WE3}$/DQM3/$\overline{ICIOWR}$		O	H	O*2	O*15	Z*5/O*1	Z	Z*5/O*1
$\overline{WE2}$/DQM2/$\overline{ICIORD}$		O	H	O*2	O*15	Z*5/O*1	Z	Z*5/O*1
$\overline{WE1}$/DQM1		O	H	O*2	O*15	Z*5/O*1	Z	Z*5/O*1
$\overline{WE0}$/DQM0/$\overline{REG}$		O	H	O*2	O*15	Z*5/O*1	Z	Z*5/O*1
$\overline{RAS}$		O	H	O*2	O*15	Z*5/O*1	Z	Z*5/O*1
$\overline{RD}$/$\overline{CASS}$/$\overline{FRAME}$		O	H	O*2	O*15	Z*5/O*1	Z	Z*5/O*1
RD/$\overline{WR}$		O	H	H	O*15	Z*5/H*3	Z	Z*5
$\overline{BACK}$		O	H	H	O	H	Z	O
$\overline{BREQ}$*9		I	PI	I	I	Z	Z	I
$\overline{DREQ0}$ – $\overline{DREQ1}$*9		I	PI	I	I	Z	Z	I
DACK0 – DACK1		O	L	L	O*15	Z*9/O*10	Z	O
DRAK0 – DRAK1		O	L	L	O*15	Z*9/O*10	Z	O
$\overline{IRL0}$ – $\overline{IRL3}$*9		I	PI	I	I	Z	Z	I
NMI*9		I	PI	I	I	Z	Z	I
$\overline{TRST}$		I	PI	PI	PI	PZ	Z	PI
TCK		I	PI	PI	PI	PZ	Z	PI
TMS		I	PI	PI	PI	PZ	Z	PI
TDI		I	PI	PI	PI	PZ	Z	PI
TDO		O	O	O	O	O	Z	O
$\overline{ASEBRK}$/BRKACK		I/O	PI/O	PI/O	PI/O	PZ/O	Z	PI/O
MD0*9		I	I	I	I	Z	Z	I
MD1*9		I	I	I	I	Z	Z	I

Signal Name	Pin Name	I/O	Power-on	Manual	Sleep	Software	Hardware	Release
MD2*9		I	I	I	I	Z	Z	I
MD3/$\overline{\text{CE2A}}$*9		I/O	PI*7	I	I/O*15	Z*10/H	Z	I
MD4/$\overline{\text{CE2B}}$*9		I/O	PI*7	I	I/O*15	Z*10/H	Z	I
MD5*9		I	PI*7	I	I	Z	Z	I
MD6/$\overline{\text{IOIS16}}$*9		I	PI*7	I	I	Z	Z	I
MD7*9		I	I	I	I	Z	Z	I
MD8*9		I	I	I	I	Z	Z	I
$\overline{\text{RESET}}$		I	I	I	I	I	I	I
$\overline{\text{MRESET}}$		I	PI	PI	PI	PI	I	PI
STATUS0 – STATUS1		O	O	O	O	O	Z/O*8	O
CKIO		O	O	PZ/O*16	PZ/O*16	PZ/O*16	Z	PZ/O*16
CKE		O	H	O*2	O*2	L	Z	O*2
CA		I	I	I	I	I	I	I
CAN0_NERR/ AUDCK*11	CAN0_NERR	I	PZ	I	I	I	Z	I
	AUDCK	O	PZ	O	O	O	Z	O
	PA7	I/O	PZ	I/O	I/O	I/O	Z	I/O
CAN0_RX/ AUDATA[2]*11	CAN0_RX	I	PZ	I	I	Z	Z	I
	AUDATA[2]	O	PZ	O	O	O	Z	O
	PA6	I/O	PZ	I/O	I/O	Z/O	Z	I/O
CAN0_TX/ AUDATA[0]*11	CAN0_TX	O	PZ	O	O	O	Z	O
	AUDATA[0]	O	PZ	O	O	O	Z	O
	PA5	I/O	PZ	I/O	I/O	I/O	Z	I/O
CAN1_NERR/ AUDSYNC*11	CAN1_NERR	I	PZ	I	I	I	Z	I
	AUDSYNC	O	PZ	O	O	O	Z	O
	PA4	I/O	PZ	I/O	I/O	I/O	Z	I/O
CAN1_RX/ AUDATA[3]*11	CAN1_RX	I	PZ	I	I	Z	Z	I
	AUDATA[3]	O	PZ	O	O	O	Z	O
	PA3	I/O	PZ	I/O	I/O	Z/O	Z	I/O
CAN1_TX/ AUDATA[1]*11	CAN1_TX	O	PZ	O	O	O	Z	O
	AUDATA[1]	O	PZ	O	O	O	Z	O
	PA2	I/O	PZ	I/O	I/O	I/O	Z	I/O

Signal Name	Pin Name	I/O	Power-on	Manual	Sleep	Software	Hardware	Release
SSI0_SCK/	SSI0_SCK	I/O	PZ	I	I/O	Z/O	Z	I/O
HAC_SD_IN0/	HAC_SD_IN0	I	PZ	I	I	Z	Z	I
$\overline{BS2}$*11	$\overline{BS2}$	O	PZ	O	O	O	Z	O
	PB7	I/O	PZ	I/O	I/O	Z/O	Z	I/O
SSI0_WS/	SSI0_WS	I/O	PZ	I	I/O	Z/O	Z	I/O
HAC_SYNC0*11	HAC_SYNC0	O	PZ	O	O	O	Z	O
	PB6	I/O	PZ	I/O	I/O	Z/O	Z	I/O
SSI0_SDATA/	SSI0_SDATA	I/O	PZ	I	I/O	Z/O	Z	I/O
HAC_SD_OUT0*11	HAC_SD_OUT0	O	PZ	O	O	O	Z	O
	PB5	I/O	PZ	I/O	I/O	Z/O	Z	I/O
SSI1_SCK/	SSI1_SCK	I/O	PZ	I	I/O	Z/O	Z	I/O
HAC_SD_IN1*11	HAC_SD_IN1	I	PZ	I	I	Z	Z	I
	PJ4	I/O	PZ	I/O	I/O	Z/O	Z	I/O
SSI1_SDATA/	SSI1_SDATA	I/O	PZ	I	I/O	Z/O	Z	I/O
HAC_SD_OUT1*11	HAC_SD_OUT1	O	PZ	O	O	O	Z	O
	PJ3	I/O	PZ	I/O	I/O	Z/O	Z	I/O
SSI1_WS/	SSI1_WS	I/O	PZ	I	I/O	Z/O	Z	I/O
HAC_SYNC1*11	HAC_SYNC1	O	PZ	O	O	O	Z	O
	PJ5	I/O	PZ	I/O	I/O	Z/O	Z	I/O
MFI-D0/	PC7	I/O	PZ	I/O	I/O	I/O	Z	I/O
LCD_DATA0*11	MFI-D0	I/O	PZ	I/O	I/O	I/O	Z	I/O
	LCD_DATA0	O	PZ	O	O	O	Z	O
MFI-D1/	PC6	I/O	PZ	I/O	I/O	I/O	Z	I/O
LCD_DATA1*11	MFI-D1	I/O	PZ	I/O	I/O	I/O	Z	I/O
	LCD_DATA1	O	PZ	O	O	O	Z	O
MFI-D2/	PC5	I/O	PZ	I/O	I/O	I/O	Z	I/O
LCD_DATA2/	MFI-D2	I/O	PZ	I/O	I/O	I/O	Z	I/O
$\overline{IRQ6}$*11	LCD_DATA2	O	PZ	O	O	O	Z	O
	$\overline{IRQ6}$	I	PZ	I	I	I	Z	I
MFI-D3/	PC4	I/O	PZ	I/O	I/O	I/O	Z	I/O
LCD_DATA3/	MFI-D3	I/O	PZ	I/O	I/O	I/O	Z	I/O
$\overline{IRQ7}$*11	LCD_DATA3	O	PZ	O	O	O	Z	O
	$\overline{IRQ7}$	I	PZ	I	I	I	Z	I

Signal Name	Pin Name	I/O	Power-on	Manual	Sleep	Software	Hardware	Release
MFI-D4/ LCD_DATA4/ $\overline{\text{DREQ2}}$*11	PC3	I/O	PZ	I/O	I/O	Z/O	Z	I/O
	MFI-D4	I/O	PZ	I/O	I/O	Z/O	Z	I/O
	LCD_DATA4	O	PZ	O	O	O	Z	O
	$\overline{\text{DREQ2}}$	I	PZ	I	I	Z	Z	I
MFI-D5/ LCD_DATA5/ DRAK2/DACK2*11	PC2	I/O	PZ	I/O	I/O	I/O	Z	I/O
	MFI-D5	I/O	PZ	I/O	I/O	I/O	Z	I/O
	LCD_DATA5	O	PZ	O	O	O	Z	O
	DRAK2/DACK2	O	PZ	O	O	O	Z	O
MFI-D6/ LCD_DATA6/ $\overline{\text{DREQ3}}$*11	PC1	I/O	PZ	I/O	I/O	Z/O	Z	I/O
	MFI-D6	I/O	PZ	I/O	I/O	Z/O	Z	I/O
	LCD_DATA6	O	PZ	O	O	O	Z	O
	$\overline{\text{DREQ3}}$	I	PZ	I	I	Z	Z	I
MFI-D7/ LCD_DATA7/ DRAK3/DACK3*11	PC0	I/O	PZ	I/O	I/O	I/O	Z	I/O
	MFI-D7	I/O	PZ	I/O	I/O	I/O	Z	I/O
	LCD_DATA7	O	PZ	O	O	O	Z	O
	DRAK3/DACK3	O	PZ	O	O	O	Z	O
MFI-D8/ LCD_DATA8*11	PD7	I/O	PZ	I/O	I/O	I/O	Z	I/O
	MFI-D8	I/O	PZ	I/O	I/O	I/O	Z	I/O
	LCD_DATA8	O	PZ	O	O	O	Z	O
MFI-D9/ LCD_DATA9*11	PD6	I/O	PZ	I/O	I/O	I/O	Z	I/O
	MFI-D9	I/O	PZ	I/O	I/O	I/O	Z	I/O
	LCD_DATA9	O	PZ	O	O	O	Z	O
MFI-D10/ LCD_DATA10*11	PD5	I/O	PZ	I/O	I/O	I/O	Z	I/O
	MFI-D10	I/O	PZ	I/O	I/O	I/O	Z	I/O
	LCD_DATA10	O	PZ	O	O	O	Z	O
MFI-D11/ LCD_DATA11*11	PD4	I/O	PZ	I/O	I/O	I/O	Z	I/O
	MFI-D11	I/O	PZ	I/O	I/O	I/O	Z	I/O
	LCD_DATA11	O	PZ	O	O	O	Z	O

Signal Name	Pin Name	I/O	Power-on	Manual	Sleep	Software	Hardware	Release
MFI-D12/ LCD_DATA12*11	PD3	I/O	PZ	I/O	I/O	I/O	Z	I/O
	MFI-D12	I/O	PZ	I/O	I/O	I/O	Z	I/O
	LCD_DATA12	O	PZ	O	O	O	Z	O
MFI-D13/ LCD_DATA13*11	PD2	I/O	PZ	I/O	I/O	I/O	Z	I/O
	MFI-D13	I/O	PZ	I/O	I/O	I/O	Z	I/O
	LCD_DATA13	O	PZ	O	O	O	Z	O
MFI-D14/ LCD_DATA14*11	PD1	I/O	PZ	I/O	I/O	I/O	Z	I/O
	MFI-D14	I/O	PZ	I/O	I/O	I/O	Z	I/O
	LCD_DATA14	O	PZ	O	O	O	Z	O
MFI-D15/ LCD_DATA15*11	PD0	I/O	PZ	I/O	I/O	I/O	Z	I/O
	MFI-D15	I/O	PZ	I/O	I/O	I/O	Z	I/O
	LCD_DATA15	O	PZ	O	O	O	Z	O
$\overline{\text{MFI-INT}}$/ LCD_CLK*11	$\overline{\text{MFI-INT}}$	O	PZ	O	O	O	Z	O
	LCD_CLK	I	PZ	I	I	Z	Z	I
	PE7	I/O	PZ	I/O	I/O	Z/O	Z	I/O
$\overline{\text{MFI-CS}}$/ LCD_DON*11	$\overline{\text{MFI-CS}}$	I	PZ	I	I	I	Z	I
	LCD_DON	O	PZ	O	O	O	Z	O
	PE6	I/O	PZ	I/O	I/O	I/O	Z	I/O
MFI-E/ LCD_CL1*11	MFI-E	I	PZ	I	I	Z	Z	I
	LCD_CL1	O	PZ	O	O	O	Z	O
	PE5	I/O	PZ	I/O	I/O	Z/O	Z	I/O
MFI-MD/ LCD_CL2*11	MFI-MD	I	PZ	I	I	Z	Z	I
	LCD_CL2	O	PZ	O	O	O	Z	O
	PE4	I/O	PZ	I/O	I/O	Z/O	Z	I/O
MFI-RS/ LCD_M_DISP*11	MFI-RS	I	PZ	I	I	I	Z	I
	LCD_M_DISP	O	PZ	O	O	O	Z	O
	PE3	I/O	PZ	I/O	I/O	I/O	Z	I/O

Signal Name	Pin Name	I/O	Power-on	Manual	Sleep	Software	Hardware	Release
MFI-RW/	MFI-RW	I	PZ	I	I	Z	Z	I
LCD_FLM*[11]	LCD_FLM	O	PZ	O	O	O	Z	O
	PE2	I/O	PZ	I/O	I/O	Z/O	Z	I/O
$\overline{\text{HAC\_RES}}$*[11]	$\overline{\text{HAC\_RES}}$	O	PZ	O	O	O	Z	O
	PJ6	I/O	PZ	I/O	I/O	I/O	Z	I/O
HAC_BIT_CLK0*[11]	HAC_BIT_CLK0	I	PZ	I	I	Z	Z	I
	PJ7	I/O	PZ	I/O	I/O	Z/O	Z	I/O
VCPWC/$\overline{\text{IRQ4}}$*[11]	$\overline{\text{IRQ4}}$	I	PZ	I	I	I	Z	I
	VCPWC	O	L	L	O	O	Z	O
	PE1	I/O	PZ	I/O	I/O	I/O	Z	I/O
VEPWC/$\overline{\text{IRQ5}}$*[11]	$\overline{\text{IRQ5}}$	I	PZ	I	I	I	Z	I
	VEPWC	O	L	L	O	O	Z	O
	PE0	I/O	PZ	I/O	I/O	I/O	Z	I/O
I2C0_SCL(O/D)*[14]		I/O	I	I	I/O	I/O	I/O	I/O
I2C0_SDA(O/D)*[14]		I/O	I	I	I/O	I/O	I/O	I/O
I2C1_SCL(O/D)*[14]		I/O	I	I	I/O	I/O	I/O	I/O
I2C1_SDA(O/D)*[14]		I/O	I	I	I/O	I/O	I/O	I/O
HSPI_TX/	HSPI_TX	I/O	PI	I	I/O	Z/O	Z	I/O
SIM_D/	SIM_D	I/O	PI	I	I/O	Z/O	Z	I/O
MCDAT*[11]	MCDAT	I/O	PI	I	I/O	Z/O	Z	I/O
	PF3	I/O	PI	I/O	I/O	Z/O	Z	I/O
HSPI_RX*[11]	HSPI_RX	I	PI	I	I	Z	Z	I
	PF2	I/O	PI	I/O	I/O	Z/O	Z	I/O
HSPI_CLK/	HSPI_CLK	I/O	PI	I	I/O	Z/O	Z	I/O
SIM_CLK/	SIM_CLK	O	PZ	O	O	O	Z	O
MCCLK*[11]	MCCLK	O	PZ	O	O	O	Z	O
	PF1	I/O	PI	I/O	I/O	Z/O	Z	I/O
$\overline{\text{HSPI\_CS}}$/	$\overline{\text{HSPI\_CS}}$	I/O	PI	I	I/O	Z/O	Z	I/O
SIM_RST/	SIM_RST	O	PZ	O	O	O	Z	O
MCCMD*[11]	MCCMD	I/O	PI	I	I/O	Z/O	Z	I/O
	PF0	I/O	PI	I/O	I/O	Z/O	Z	I/O

Signal Name	Pin Name	I/O	Power-on	Manual	Sleep	Software	Hardware	Release
CMT_CTR0/	CMT_CTR0	I/O	PI	I	I/O	Z/O	Z	I/O
TCLK*11	TCLK	I	PI	I	I	Z	Z	I
	PB4	I/O	PI	I/O	I/O	Z/O	Z	I/O
CMT_CTR1*11	CMT_CTR1	I/O	PI	I	I/O	Z/O	Z	I/O
	PB3	I/O	PI	I/O	I/O	Z/O	Z	I/O
CMT_CTR2*11	CMT_CTR2	I/O	PI	I	I/O	Z/O	Z	I/O
	PB2	I/O	PI	I/O	I/O	Z/O	Z	I/O
CMT_CTR3*11	CMT_CTR3	I/O	PI	I	I/O	Z/O	Z	I/O
	PB1	I/O	PI	I/O	I/O	Z/O	Z	I/O
SCIF0_CLK*11	SCIF0_CLK	I/O	PI	I	I/O	Z/O	Z	I/O
	PG7	I/O	PI	I/O	I/O	I/O	Z	I/O
SCIF0_RXD*11	SCIF0_RXD	I	PI	I	I	Z	Z	I
	PG6	I/O	PI	I/O	I/O	I/O	Z	I/O
SCIF0_TXD*11	SCIF0_TXD	O	PZ	Z	Z/O	Z/O	Z	Z/O
	PG5	I/O	PI	I/O	I/O	I/O	Z	I/O
SCIF1_CLK*11	SCIF1_CLK	I/O	PI	I	I/O	Z/O	Z	I/O
	PG4	I/O	PI	I/O	I/O	I/O	Z	I/O
$\overline{\text{SCIF1\_CTS}}$*11	$\overline{\text{SCIF1\_CTS}}$	I/O	PI	I	I/O	Z/O	Z	I/O
	PG3	I/O	PI	I/O	I/O	I/O	Z	I/O
$\overline{\text{SCIF1\_RTS}}$*11	$\overline{\text{SCIF1\_RTS}}$	I/O	PI	I	I/O	Z/O	Z	I/O
	PG2	I/O	PI	I/O	I/O	I/O	Z	I/O
SCIF1_RXD*11	SCIF1_RXD	I	PI	I	I	Z	Z	I
	PG1	I/O	PI	I/O	I/O	I/O	Z	I/O
SCIF1_TXD*11	SCIF1_TXD	O	PZ	Z	Z/O	Z/O	Z	Z/O
	PG0	I/O	PI	I/O	I/O	I/O	Z	I/O
SCIF2_CLK*11	SCIF2_CLK	I/O	PI	I	I/O	Z/O	Z	I/O
	PH7	I/O	PI	I/O	I/O	I/O	Z	I/O
$\overline{\text{SCIF2\_CTS}}$*11	$\overline{\text{SCIF2\_CTS}}$	I/O	PI	I	I/O	Z/O	Z	I/O
	PH6	I/O	PI	I/O	I/O	I/O	Z	I/O
$\overline{\text{SCIF2\_RTS}}$*11	$\overline{\text{SCIF2\_RTS}}$	I/O	PI	I	I/O	Z/O	Z	I/O
	PH5	I/O	PI	I/O	I/O	I/O	Z	I/O

Signal Name	Pin Name	I/O	Power-on	Manual	Sleep	Software	Hardware	Release
SCIF2_RXD*[11]	SCIF2_RXD	I	PI	I	I	Z	Z	I
	PH4	I/O	PI	I/O	I/O	I/O	Z	I/O
SCIF2_TXD*[11]	SCIF2_TXD	O	PZ	Z	Z/O	Z/O	Z	Z/O
	PH3	I/O	PI	I/O	I/O	I/O	Z	I/O
Reserved /AUDATA[3]*[11]	AUDATA[3]	O	PZ	O	O	O	Z	O
	PK7	I/O	PZ	I/O	I/O	I/O	Z	I/O
Reserved/AUDATA[2]*[11]	AUDATA[2]	O	PZ	O	O	O	Z	O
	PK6	I/O	PZ	I/O	I/O	I/O	Z	I/O
Reserved /AUDATA[1]*[11]	AUDATA[1]	O	PZ	O	O	O	Z	O
	PK5	I/O	PZ	I/O	I/O	I/O	Z	I/O
Reserved /AUDCK*[11]	AUDCK	O	PZ	O	O	O	Z	O
	PK4	I/O	PZ	I/O	I/O	I/O	Z	I/O
Reserved /AUDSYNC*[11]	AUDSYNC	O	PZ	O	O	O	Z	O
	PK3	I/O	PZ	I/O	I/O	I/O	Z	I/O
$\overline{\text{ADTRG}}$/AUDATA[0]*[11]	$\overline{\text{ADTRG}}$	I	PZ	I	I	Z	Z	I
	AUDATA[0]	O	PZ	O	O	O	Z	O
	PK2	I/O	PZ	I/O	I/O	Z/O	Z	I/O
AN0*[12]		I	Z	Z	I/Z	Z	Z	I/Z
AN1*[12]		I	Z	Z	I/Z	Z	Z	I/Z
AN2*[12]		I	Z	Z	I/Z	Z	Z	I/Z
AN3*[12]		I	Z	Z	I/Z	Z	Z	I/Z
UCLK*[11]	UCLK	I	PI	I	I	Z	Z	I
	PH2	I/O	PI	I/O	I/O	Z/O	Z	I/O
USB_PENC*[11]	USB_PENC	O	PZ	O	O	O	Z	O
	PH1	I/O	PZ	I/O	I/O	I/O	Z	I/O
$\overline{\text{USB\_OVC}}$*[11]	$\overline{\text{USB\_OVC}}$	I	PI	I	I	Z	Z	I
	PH0	I/O	PI	I/O	I/O	Z/O	Z	I/O
USB_DP*[13]		I/O	O	I	I/O	I	O	I/O
USB_DM*[13]		I/O	O	I	I/O	I	O	I/O

Signal Name	Pin Name	I/O	Power-on	Manual	Sleep	Software	Hardware	Release
HAC_BIT_CLK1*11	HAC_BIT_CLK1	I	PZ	I	I	Z	Z	I
	PJ2	I/O	PZ	I/O	I/O	Z/O	Z	I/O
DCK*11	DCK	O	PZ	PZ/O	PZ/O	PZ/O	Z	PZ/O
	PJ1	O	PZ	O	O	O	Z	O

Legend: I: Input
O: Output
H: High level output
L: Low level output
Z: Hi-Z state
PI: Input pulled up with a built-in pull-up resistance.
PZ: Output pulled up with a built-in pull-up resistance.

Notes:
1. Z (I) or O (refresh) according to the register contents (BCR1.HIZCNT).
2. Depends on the refresh operation.
3. Z (I) or H (retained) according to the register contents (BCR1.HIZMEM).
4. Output when the auto-refresh is selected.
5. Pulled up or not according to the register contents (BCR1.OPUP).
6. Pulled up or not according to the register contents (BCR1.DPUP).
7. Pulled up using the pull-up MOSs. However, the pull-up MOSs cannot be used to pull-up the mode pins at a power-on reset. For this purpose, pull-up or pull-down outside the LSI.
8. Z or O according to the register contents (STBCR2.STHZ).
9. Pulled up or not according to the PFC register settings (see section 24, Pin Function Controller (PFC)). However, the PFC register settings are invalid in hardware standby mode.
10. Hi-Z or not according to the PFC register settings (see section 24, Pin Function Controller (PFC)).
11. Pulled up or not, and the multiplexed functions for IP modules are selected by the PFC register settings (see section 24, Pin Function Controller (PFC)). However, the PFC register settings are invalid in hardware standby mode. For details of the I/O control for the selected IP modules, see the corresponding section. Selection of the GPIO functions and the I/O control of the GPIO are determined according to the GPIO register settings (see section 24, Pin Function Controller (PFC)).
12. According to the ADC register settings. Hi-Z at initialization.
13. Pull-down for USB pins while not used.
14. Only low level output since these pins are open-drain pins. Pulled up when the I^2C is not in use.
15. Depends on the refresh and DMAC operations.
16. Z or O according to the register contents (FRQCR.CKOEN).

- When PLL1 is not used
  V_{DD_PLL1}: Power supply
  V_{SS_PLL1}: Power supply
- When PLL2 is not used
  V_{DD_PLL2}: Power supply
  V_{SS_PLL2}: Power supply
- When PLL3 is not used
  V_{DD_PLL3}: Power supply
  V_{SS_PLL3}: Power supply
- When on-chip crystal oscillator is not used
  XTAL: Leave unconnected
  V_{DD_CPG}: Power supply
  V_{SS_CPG}: Power supply
- When I^2C is not used
  I2C0_SCL: Pull-up outside the LSI.
  I2C0_SDA: Pull-up outside the LSI.
  I2C1_SCL: Pull-up outside the LSI.
  I2C1_SDA: Pull-up outside the LSI.
- When USB is not used
  USB_DP: Pull-down outside the LSI.
  USB_DM: Pull-down outside the LSI.
  USB_PENC: Pull-up outside the LSI.
  $\overline{\text{USB\_OVC}}$: Pull-up outside the LSI.
  UCLK: Pull-up outside the LSI.
- When ADC is not used
  $AV_{CC\text{-}ADC}$: Power supply
  $AV_{SS\text{-}ADC}$: Power supply
  AN0 to AN3: Pull-down outside the LSI.
- Hardware Standby is not used
  CA: Pull-up* outside the LSI

Note: * To prevent unwanted effects on other pins when using external pull-up resistor, use independent pull-up resistor for individual pin. For other unused pins, it is recommended to handle the pin individually.

(1) BUS 32 (16M: 512k × 16b × 2) × 2 *
AMX 0 AMXEXT 0 16M, column-addr-8bit 4MB

SH7760 Series Address Pins			Synchronous DRAM Address Pins	Function
	RAS Cycle	CAS Cycle		
A14				
A13	A21	A21	A11	BANK selects bank address
A12	A20	H/L	A10	Address precharge setting
A11	A19	0	A9	Address
A10	A18	0	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	Not used			
A0	Not used			

SH7760 Series Address Pins			Synchronous DRAM Address Pins	Function
	RAS Cycle	CAS Cycle		
A14				
A13	A20	A20	A11	BANK selects bank address
A12	A21	H/L	A10	Address precharge setting
A11	A19	0	A9	Address
A10	A18	0	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	Not used			
A0	Not used			

SH7760 Series Address Pins			Synchronous DRAM Address Pins	Function
	RAS Cycle	CAS Cycle		
A14				
A13	A22	A22	A11	BANK selects bank address
A12	A21	H/L	A10	Address precharge setting
A11	A20	0	A9	Address
A10	A19	A10	A8	
A9	A18	A9	A7	
A8	A17	A8	A6	
A7	A16	A7	A5	
A6	A15	A6	A4	
A5	A14	A5	A3	
A4	A13	A4	A2	
A3	A12	A3	A1	
A2	A11	A2	A0	
A1	Not used			
A0	Not used			

SH7760 Series Address Pins			Synchronous DRAM Address Pins	Function
	RAS Cycle	CAS Cycle		
A14				
A13	A21	A21	A11	BANK selects bank address
A12	A22	H/L	A10	Address precharge setting
A11	A20	0	A9	Address
A10	A19	A10	A8	
A9	A18	A9	A7	
A8	A17	A8	A6	
A7	A16	A7	A5	
A6	A15	A6	A4	
A5	A14	A5	A3	
A4	A13	A4	A2	
A3	A12	A3	A1	
A2	A11	A2	A0	
A1	Not used			
A0	Not used			

AMX2 = 64M, column addr 8bit = 16MB

SH7760 Series Address Pins			Synchronous DRAM Address Pins	Function
	RAS Cycle	CAS Cycle		
A16				
A15	A23	A23	A13	BANK selects bank address
A14	A22	A22	A12	
A13	A21	0	A11	Address precharge setting
A12	A20	H/L	A10	
A11	A19	0	A9	Address
A10	A18	0	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	Not used			
A0	Not used			

AMX3 = 64M, column addr 9bit = 32MB

SH7760 Series Address Pins			Synchronous DRAM Address Pins	Function
	RAS Cycle	CAS Cycle		
A16				
A15	A24	A24	A13	BANK selects bank address
A14	A23	A23	A12	
A13	A22	0	A11	Address precharge setting
A12	A21	H/L	A10	
A11	A20	0	A9	Address
A10	A19	A10	A8	
A9	A18	A9	A7	
A8	A17	A8	A6	
A7	A16	A7	A5	
A6	A15	A6	A4	
A5	A14	A5	A3	
A4	A13	A4	A2	
A3	A12	A3	A1	
A2	A11	A2	A0	
A1	Not used			
A0	Not used			

AMX = 64M, column addr 8bit 8MB

SH7760 Series Address Pins			Synchronous DRAM Address Pins	Function
	RAS Cycle	CAS Cycle		
A15				
A14	A22	A22	A12	BANK selects bank address
A13	A21	A21	A11	
A12	A20	H/L	A10	Address precharge setting
A11	A19	0	A9	Address
A10	A18	0	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	Not used			
A0	Not used			

AMX3 64M, column addr 8bit 8MB

SH7760 Series Address Pins			Synchronous DRAM Address Pins	Function
	RAS Cycle	CAS Cycle		
A15				
A14	A22	A22	A12	BANK selects bank address
A13	A21	0	A11	
A12	A20	H/L	A10	Address precharge setting
A11	A19	0	A9	Address
A10	A18	0	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	Not used			
A0	Not used			

AMX 6 AMXEXT0 64M, column-addr-10bit 64MB

SH7760 Series Address Pins			Synchronous DRAM Address Pins	Function
	RAS Cycle	CAS Cycle		
A15	A25	A25	A13	BANK selects bank address
A14	A24	A24	A12	
A13	A23	0	A11	
A12	A22	H/L	A10	Address precharge setting
A11	A21	A11	A9	Address
A10	A20	A10	A8	
A9	A19	A9	A7	
A8	A18	A8	A6	
A7	A17	A7	A5	
A6	A16	A6	A4	
A5	A15	A5	A3	
A4	A14	A4	A2	
A3	A13	A3	A1	
A2	A12	A2	A0	
A1	Not used			
A0	Not used			

SH7760 Series Address Pins			Synchronous DRAM Address Pins	Function
	RAS Cycle	CAS Cycle		
A16	A25	A25	A14	BANK selects bank address
A15	A24	A24	A13	
A14	A23	0	A12	
A13	A22	0	A11	
A12	A21	H/L	A10	Address precharge setting
A11	A20	0	A9	Address
A10	A19	A10	A8	
A9	A18	A9	A7	
A8	A17	A8	A6	
A7	A16	A7	A5	
A6	A15	A6	A4	
A5	A14	A5	A3	
A4	A13	A4	A2	
A3	A12	A3	A1	
A2	A11	A2	A0	
A1	Not used			
A0	Not used			

AMX = 1: 16M, column addr 8bit: 2MB

SH7760 Series Address Pins			Synchronous DRAM Address Pins	Function
	RAS Cycle	CAS Cycle		
A13				
A12	A20	A20	A10	BANK selects bank address
A11	A19	H/L	A9	Address precharge setting
A10	A18	0	A8	Address
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	Not used			
A0	Not used			

Note: * Example configurations of synchronous DRAM

This LSI is provided with an internal buffer for holding pre-read instructions, and always performs pre-reading. Therefore, program code must not be located in the last 20-byte area of any memory space. If program code is located in these areas, the memory area will be exceeded and a bus access for instruction pre-reading may be initiated. A case in which this is a problem is shown below.

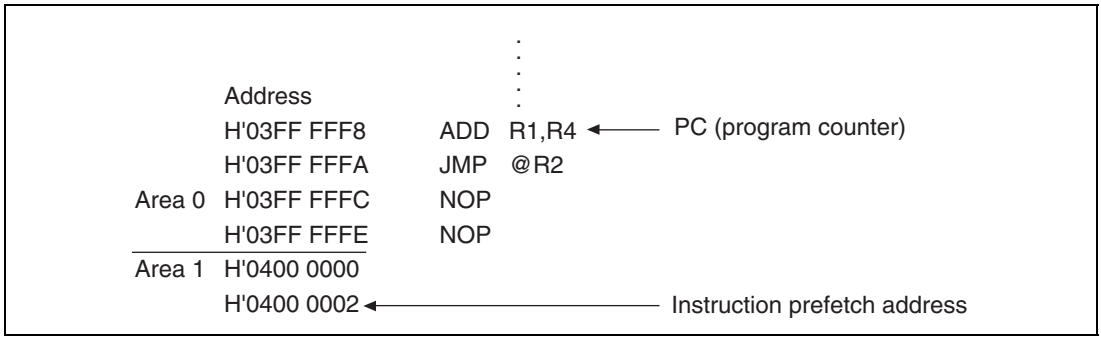


Figure E.1 Instruction Prefetch

Figure E.1 presupposes a case in which the instruction (ADD) indicated by the program counter (PC) and the address H'0400 0002 instruction prefetch are executed simultaneously. It is also assumed that the program branches to an area other than area 1 after executing the following JMP instruction and delay slot instruction.

In this case, the program flow is unpredictable, and a bus access (instruction prefetch) to area 1 may be initiated.

Instruction Prefetch Side Effects

1. It is possible that an external bus access caused by an instruction prefetch may result in misoperation of an external device, such as a FIFO, connected to the area concerned.
2. If there is no device to reply to an external bus request caused by an instruction prefetch, hangup will occur.

Remedies

1. These illegal instruction fetches can be avoided by using the MMU.
2. The problem can be avoided by not locating program code in the last 20 bytes of any area.

F.1 Power-on Procedure

(1) Turn on the power supply of I/O, and CPG to the same timing as the power supply VDDQ.

(2) After turning on a power supply to VDDQ or simultaneously, the input signal to pin ($\overline{\text{RESET}}$, $\overline{\text{MRESET}}$, MD0-MD10, external clock, and etc) must be input. If any signal is input to the LSI pin before the power supply turns on the LSI may be damaged.

(i) Make the $\overline{\text{RESET}}$ signal into the Low level at the power supply VDDQ.

(3) Turn on the power supply so that the voltage of a power supply VDD may keep less than 1.2 V until the voltage of a power supply VDDQ reaches 2 V to avoid an abnormal oscillation of the PLL1/2/3 circuit.

(4) It recommends that a power supply to VDDQ is in advance, and a power supply to VDD is behind.

(5) In addition to the above (1), (2), (3), (4), and following (i), (ii), follow “F.3, The Ratings for Power-on and Power-off”.

(i) In the case of this LSI simple substance, the power supply sequence of a power supply to VDDQ and VDD does not have time restrictions. Refer to figure F.1. In addition, it recommends performing a power-on procedure at the shortest possible time.

(ii) When this LSI is connected with other elements on the mounting board, please keep that

$-0.3\ \text{V} < \text{Vin} < \text{VDDQ} + 0.3\ \text{V}$.

Moreover, as shown in figure F.2, power up time (tpwu) until it goes up from the following condition (A) to an operation voltage value {VDDQ (min), VDD (min)} is 100 ms (max). If it is longer than that time, the LSI may be damaged.

In addition, it recommends performing a power-on procedure at the shortest possible time.

Condition (A): $\text{VDDQ} \geq 1.0\ \text{V}$ or $\text{VDD} \geq 0.5\ \text{V}$

(1) Turn off the power supply of I/O and CPG to the same timing as the power supply VDDQ.

(2) There is no timing regulation of a signal line ($\overline{\text{RESET}}$ and $\overline{\text{MRESET}}$).

(3) The input level to the pin must be lowered in compliance with the I/O, RTC, CPG power supply voltage.

(4) It recommends that turning off the I/O, CPG power supply voltage (VDDQ) after turning off the internal power supply voltage (VDD).

(5) In addition to the above (1), (2), (3), (4), and following (i), (ii), follow "F.3, The Ratings for Power-on and Power-off".

(i) In the case of this LSI simple substance, the power supply sequence of a power supply to VDDQ and VDD does not have time restrictions. Refer to figure F.1. In addition, it recommends performing a power off at the shortest possible time.

(ii) When this LSI is connected with other elements on the mounting board, please keep that –0.3 V < Vin < VDDQ +0.3 V.

Moreover, as shown in figure F.2, power down time (tpwd) until it goes down from an operation voltage value {VDDQ (min), VDD (min)} to the following condition (B) is 150 ms (max).

If it is longer than that time, the LSI may be damaged.

In addition, it recommends performing a power off at the shortest possible time.

Condition (B): VDDQ < 1.0 V and VDD < 0.5 V

F.3 The Ratings for Power-on and Power-off

The LSI may be damaged when not satisfying the conditions of (1), (2), (3), and (4) below.

(1) VDDQ = VDD-CPG = AVCC–ADC

(2) VDD = VDD–PLL1 = VDD–PLL2 = VDD–PLL3

(3) –0.3V < VDD < VDDQ +0.3 V.

–0.3 V < VDD–PLL1 < VDDQ +0.3 V.

–0.3 V < VDD–PLL2 < VDDQ +0.3 V.

–0.3 V < VDD–PLL3 < VDDQ +0.3 V.

(4) VSS = VSSQ = VSS–PLL1=VSS–PLL2 = VSS–PLL3 = VSS–CPG = AVSS–ADC = GND [0V].

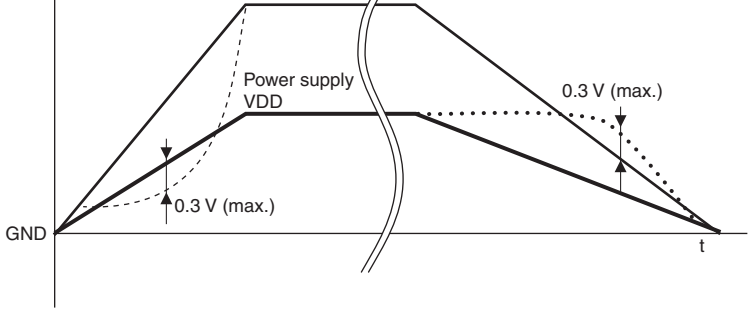


Figure F.1 Power-on and Power-off Procedure (1)

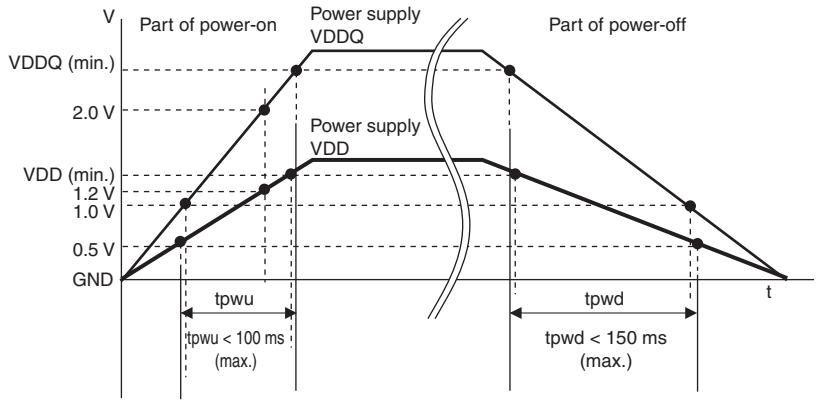


Figure F.2 Power-on and Power-off Procedure (2)

Table G.1 SH7760 Product Lineup

Group	Voltage	Operating Frequency	Operating Temperature	Part Number*	Package
SH7760	1.5 V	200 MHz	–20 to 75°C	HD6417760BL200A	256-pin BGA (BP-256B)
				HD6417760BL200AV	
			–40 to 85°C	HD6417760BL200AD	256-pin BGA (BP-256B)
				HD6417760BL200ADV	
				HD6417760BP200AD	256-pin BGA (BP-256F)
				HD6417760BP200ADV	

Note: * All listed products are available in lead-free versions. Lead-free products have a “V” appended at the end of the part number.

The registers related to the version registers are shown below.

Table H.1 Register Configuration

Register Name	Abbrev.	R/W	Initial value	P4 Address	Area 7 Address	Size
Processor version register	PVR	R	H'0405 01xx	H'FF00 0030	H'1F00 0030	32
Product register	PRR	R	H'0000 05xx	H'FF00 0044	H'1F00 0044	32

Legend: x: Undefined

(1) Processor Version Register (PVR)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	version information															
Initial value:	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	1
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
	version information								-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	1	-	-	-	-	-	-	-	-
R/W:	R	R	R	R	R	R	R	R	-	-	-	-	-	-	-	-

(2) Product Register (PRR)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	version information															
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
	version information								-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	1	0	1	-	-	-	-	-	-	-	-
R/W:	R	R	R	R	R	R	R	R	-	-	-	-	-	-	-	-

Renesas 32-Bit RISC Microcomputer
Hardware Manual
SH7760 Group

Publication Date: 1st Edition, February 2003
Rev.2.00, February 12, 2010
Published by: Sales Strategic Planning Div.
Renesas Technology Corp.
Edited by: Customer Support Department
Global Strategic Communication Div.
Renesas Solutions Corp.

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 204, 205, AZIACenter, No.1233 Lujiazui Ring Rd, Pudong District, Shanghai, China 200120
Tel: <86> (21) 5877-1818, Fax: <86> (21) 6887-7858/7898

Renesas Technology Hong Kong Ltd.
7th Floor, North Tower, World Finance Centre, Harbour City, Canton Road, Tsimshatsui, Kowloon, Hong Kong
Tel: <852> 2265-6688, Fax: <852> 2377-3473

Renesas Technology Taiwan Co., Ltd.
10th Floor, No.99, Fushing North Road, Taipei, Taiwan
Tel: <886> (2) 2715-2888, Fax: <886> (2) 3518-3399

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, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: <603> 7955-9390, Fax: <603> 7955-9510

Renesas Electronics Corporation
1753, Shimonumabe, Nakahara-ku, Kawasaki-shi, Kanagawa 211-8668 Japan

X-ON Electronics

Largest Supplier of Electrical and Electronic Components

Click to view similar products for Microprocessors - MPU ***category:***

Click to view products by Renesas ***manufacturer:***

Other Similar products are found below :

MC68302EH20C MC7457RX1000LC MC7457RX1267LC MC7457VG1267LC A2C00010998 A A2C52004004 R5F117BCGNA#20 R5F52106BDLA#U0 R5S72690W266BG#U0 ADJ3400IAA5DOE MPC8245TVV266D MPC8245TZU300D MPC8260ACVVMHBB MPC8323ECVRAFDCA MPC8536ECVJAVLA BOXNUC5PGYH0AJ 20-668-0024 P1010NSN5DFB P2010NSN2MHC P2020NXE2HHC P5020NSE7QMB P5020NSE7TNB P5020NSE7VNB LS1020ASN7KQB LS1020AXN7HNB LS1020AXN7KQB A2C00010729 A A2C00039344 T1022NSE7MQB T1022NXN7PQB T1023NSE7MQA T1024NXE7PQA T1042NSE7MQB T1042NSN7MQB T1042NXN7WQB T2080NSE8TTB T2080NSN8PTB T2080NXE8TTB T2081NXN8TTB R5F101AFASP#V0 MC68302CEH20C TS68040MF33A MPC8260ACVVMIBB MPC8280CZUUPEA MPC8313ECVRAFFC MPC8313ECVRAGDC MPC8313EVRADDC MPC8313EVRAFFC MPC8313VRADDC MPC8314CVRAGDA