REJ09B0332-0500

Everywhere you imagine. RENESAS

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

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

32

SH7065

Hardware Manual

Hardware Manual

Renesas 32-Bit RISC Microcomputer
SuperH™ RISC engine Family/SH7000 Series

SH7065	HD6437065A
	HD64F7065SF
	HD64F7065AF

Rev. 5.00
Revision Date: Sep 11, 2006

Renesas Technology
www.renesas.com

Keep safety first in your circuit designs!

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

Notes regarding these materials

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

General Precautions on Handling of Product

1. Treatment of NC Pins

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

2. Treatment of Unused Input Pins

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

3. Processing before Initialization

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

4. Prohibition of Access to Undefined or Reserved Addresses

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

Preface

The SH7065 is a microprocessor that integrates peripheral functions necessary for system configuration with a 32-bit internal architecture SH2-DSP CPU as its core.

On-chip peripheral functions include large-capacity ROM and RAM, an interrupt controller, four kinds of timers, a serial communication interface, user break controller (UBC), bus state controller (BSC), direct memory access controller (DMAC), A/D converter, D/A converter, and I/O ports, enabling the SH7065 to be used as a microcontroller for electronic products requiring high speed and low power consumption. Flash memory (F-ZTAT™*) and mask ROM are available as on-chip ROM, enabling users to respond quickly and flexibly to changing application specifications and the demands of the transition from initial to full-fledged volume production.

Note: * F-ZTAT is a trademark of Renesas Technology Corp.

Intended Readership: This manual is intended for users undertaking the design of an application system using the SH7065. Readers using this manual require a basic knowledge of electrical circuits, logic circuits, and microcomputers.

Purpose: The purpose of this manual is to give users an understanding of the hardware functions and electrical characteristics of the SH7065. Details of execution instructions can be found in the SH-1, SH-2, SH-DSP Programming Manual, which should be read in conjunction with the present manual.

Using this Manual:

- For an overall understanding of the SH7065's functions
  Follow the Table of Contents. This manual is broadly divided into sections on the CPU, system control functions, peripheral functions, and electrical characteristics.
- For a detailed understanding of CPU functions
  Refer to the separate publication SH-1, SH-2, SH-DSP Programming Manual.

  Note on bit notation: Bits are shown in high-to-low order from left to right.

Related Material: The latest information is available at our Web Site. Please make sure that you have the most up-to-date information available.
http://www.renesas.com/

User’s Manuals on the SH7065:

Manual Title	Document No.
SH7065 Hardware Manual	This manual
SH-1, SH-2, SH-DSP Software Manual	REJ09B0171-0500

Users manuals for development tools:

Manual Title	Document No.
C/C++ Compiler, Assembler, Optimizing Linkage Editor User’s Manual	REJ10B0047-0100
Simulator/Debugger User’s Manual	REJ10B0210-0200
High-performance Embedded Workshop User’s Manual	REJ10B0025-0200

Application Note:

Manual Title	Document No.
C/C++ Compiler	REJ05B0463-0300

Main Revisions for This Edition

<table>
<tr><th>Item</th><th>Page</th><th>Revision (See Manual for Details)</th></tr>
<tr><td>All</td><td>—</td><td>• Notification of change in company name amended
(Before) Hitachi, Ltd. → (After) Renesas Technology Corp.</td></tr>
<tr><td>9.3.4 Types of DMA Transfer
Relationship between DMA Transfer Type, Request Mode, and Bus Mode
Table 9.6 Relationship between DMA Transfer Type, Request Mode, and Bus Mode</td><td>340</td><td>Table amended
<table>
<tr><th>Address Mode</th><th>Type of Transfer</th><th>Request Mode</th><th>Bus Mode</th><th>Transfer Size (Bits)</th><th>Usable Channels</th></tr>
<tr><td>Dual</td><td>External memory and external memory</td><td>Any*1</td><td>B/C</td><td>8/16/32</td><td>0–3</td></tr>
<tr><td></td><td>External memory and memory-mapped external device</td><td>Any*1</td><td>B/C</td><td>8/16/32</td><td>0–3</td></tr>
<tr><td></td><td>Memory-mapped external device and memory-mapped external device</td><td>Any*1</td><td>B/C</td><td>8/16/32</td><td>0–3</td></tr>
<tr><td></td><td>External memory and on-chip memory</td><td>Any*1</td><td>B/C</td><td>8/16/32</td><td>0–3</td></tr>
<tr><td></td><td>External memory and on-chip peripheral module</td><td>Any*2</td><td>B/C</td><td>8/16/32*3</td><td>0–3</td></tr>
<tr><td></td><td>Memory-mapped external device and on-chip memory</td><td>Any*1</td><td>B/C</td><td>8/16/32</td><td>0–3</td></tr>
<tr><td></td><td>Memory-mapped external device and on-chip peripheral module</td><td>Any*2</td><td>B/C</td><td>8/16/32*3</td><td>0–3</td></tr>
<tr><td></td><td>On-chip memory and on-chip memory</td><td>Any*1</td><td>B/C</td><td>8/16/32</td><td>0–3</td></tr>
<tr><td></td><td>On-chip memory and on-chip peripheral module</td><td>Any*2</td><td>B/C</td><td>8/16/32*3</td><td>0–3</td></tr>
<tr><td></td><td>On-chip peripheral module and on-chip peripheral module</td><td>Any*2</td><td>B/C</td><td>8/16/32*3</td><td>0–3</td></tr>
</table>
</td></tr>
<tr><td>11.6 Usage Notes</td><td>484</td><td>Description added
Pay Attention to the Notices Below, When a Value Is Written into the Timer General Register U (TGRU), Timer General Register V (TGRV), Timer General Register W (TGRW), and in Case of Written into Free Operation Address (*): …
Writing Operation into Timer Period Data Register (TPDR) and Timer Dead Time Data Register (TDDR) When MMT Is Operating: …
Notes on Halting TCNT Counter Operation: …</td></tr>
<tr><td>15.7.2 Handling of Analog Input Pins
Figure 15.8 Example of Analog Input Pin Protection Circuit
Figure 15.9 Analog Input Pin Equivalent Circuit
Table 15.5 Analog Input Pin Specifications</td><td>619, 620</td><td>Description of preliminary deleted</td></tr>
</table>

Item	Page	Revision (See Manual for Details)
22.3.1 Clock Timing Table 22.4 Clock Timing	796	Table amended

Item	Symbol	Min	Max	Unit	Figure
Operating frequency (master clock)	f_{OP}	20	60	MHz	Figure 22.2
Clock cycle time	t_{cyc}	16.7	50	ns	
Clock low-level pulse width	t_{CL}	4.4	—	ns	
Clock high-level pulse width	t_{CH}	4.4	—	ns	
Clock rise time	t_{CR}	—	4	ns	
Clock fall time	t_{CF}	—	4	ns	
EXTAL/CKIO clock input frequency	f_{EX}	5	30	MHz	Figure 22.3
EXTAL/CKIO clock input cycle time	t_{EXcyc}	33.3	200	ns	
EXTAL/CKIO clock input low-level pulse width	t_{EXL}	11.6	—	ns	
EXTAL/CKIO clock input high-level pulse width	t_{EXH}	11.6	—	ns	
EXTAL/CKIO clock input rise time	t_{EXR}	—	5	ns	
EXTAL/CKIO clock input fall time	t_{EXF}	—	5	ns	
Reset oscillation settling time	t_{OSC1}	10	—	ms	Figure 22.4
Standby recovery oscillation settling time	t_{OSC2}	10	—	ms	

Contents

Section 8 Bus State Controller (BSC) 219

Section 9 Direct Memory Access Controller (DMAC) 305

Section 1 Overview

1.1 Features of SH7065

The SH7065 is a CMOS single-chip microcomputer featuring an SH2-DSP core—a functionally enhanced version of the SuperH RISC engine using an original Renesas Technology architecture—with the same signal processing capability as a general-purpose digital signal processor (DSP), together with peripheral functions required for system configuration.

The SH2-DSP core offers enhancement of the DSP functions (multiply and multiply-and-accumulate) of the SuperH RISC engine, and provides full DSP type data bus functionality, enabling efficient execution of various kinds of signal processing and image processing. With this CPU, it has become possible to create low-cost, high-performance/high-functionality systems even for applications such as realtime control, which could not previously be handled by microcomputers because of their high-speed processing requirements.

In addition, the SH7065 includes on-chip peripheral functions necessary for system configuration, such as large-capacity ROM and RAM, timers, a serial communication interface (SCI), A/D converter, D/A converter, interrupt controller (INTC), and I/O ports. An external memory access support function allows efficient connection of memory and peripheral LSIs, greatly reducing system cost.

There are two versions of the SH7065, with different kinds of on-chip ROM: an F-ZTAT version with on-chip flash memory, and a mask ROM version. In the F-ZTAT version, programs can be written and rewritten with a Renesas-recommended ROM programmer, or on-board.

Table 1.1 Features

Item	Specifications
CPU	• Original Renesas Technology architecture • 32-bit internal configuration • General register machine — Sixteen 32-bit general registers — Six 32-bit control registers (including three added for DSP use) — Ten 32-bit system registers (including six added for DSP use) • RISC (reduced instruction set computer) type instruction set — Fixed 16-bit instruction length for improved code efficiency — Load-store architecture (basic operations are executed between registers) — Delayed branch instructions reduce pipeline disruption during branches — C-oriented instruction set • Instruction execution time: One instruction per cycle • Address space: Architecture supports 4 Gbytes • Enhanced on-chip multiplier: — 16 × 16 → 32 multiply operations executed in one to three cycles — 32 × 32 → 64 multiply operations executed in two to four cycles — 32 × 32 + 64 → 64 multiply-and-accumulate operations executed in two to four cycles • Five-stage pipeline

Item	Specifications
DSP	• DSP engine — Multiplier — Arithmetic logic unit (ALU) — Shifter — DSP registers • Multiplier — 16 bits × 16 bits → 32 bits — Single-cycle multiplier • DSP registers — Two 40-bit data registers — Six 32-bit data registers — Modulo register (MOD, 32 bits) added to control registers — Repeat counter (RC) added to status register (SR) — Repeat start register (RS, 32 bits) and repeat end register (RE, 32 bits) added to control registers • DSP data bus — Extended Harvard architecture — Simultaneous access to two data buses and one instruction bus • Parallel processing — Maximum of four parallel processes — ALU operations, multiplication, and two loads or stores • Address processors — Two address processors — Address operations to access two memories • DSP data addressing modes — Increment and index — Each with or without modulo addressing • Repeat control: Zero-overhead repeat (loop) control • Instruction set — 16-bit length (in case of load or store only) — 32-bit length (including ALU operations and multiplication) — Added system control instructions for accessing DSP registers • Fifth and last pipeline stage is DSP stage

Item	Specifications
Interrupt controller (INTC)	• Nine external interrupt pins (NMI, $\overline{\text{IRQ0}}$ to $\overline{\text{IRQ7}}$) 15 external interrupt sources (encoded input) can also be selected for pins $\overline{\text{IRQ0}}$ to $\overline{\text{IRQ3}}$ • 16 programmable priority levels • NMI noise canceler function • Interrupt acceptance can be reported externally ($\overline{\text{IRQOUT}}$ pin)
User break controller (UBC)	• Requests an interrupt when the CPU or DMAC generates a bus cycle with specific set conditions • Simplifies configuration of an on-chip debugger
Bus state controller (BSC)	• Supports external expansion memory access — 32-bit external data bus • Address space divided into six areas (four areas in SRAM space, two areas in DRAM space), with the following parameters settable for each area: — Bus size (8/16/32 bits) — Number of wait cycles — SRAM, DRAM, and EDO DRAM easily connectable by space type setting — Output of $\overline{\text{RAS}}$ and $\overline{\text{CAS}}$ signals for DRAM and EDO DRAM — Addressing multiplexing supported internally, allowing direct connection of DRAM and EDO DRAM • DRAM and EDO DRAM burst access functions — DRAM and EDO DRAM fast access mode supported • DRAM and EDO DRAM refresh functions — Programmable refresh interval — CAS-before-RAS refreshing and self-refreshing supported — Up to eight consecutive CAS-before-RAS refreshes possible • Wait cycles can be inserted using an external $\overline{\text{WAIT}}$ signal • Can access I/O devices that use address/data multiplexing • Big-endian or little-endian mode can be set independently for each area

Item	Specifications
Direct memory access controller (DMAC) (4 channels)	• DMA transfer possible for the following devices: — External memory, external I/O, on-chip supporting modules (excluding DMAC, BSC, UBC) • DMA transfer requests by external pins (for two channels) and on-chip peripheral modules, plus auto-request • Cycle steal or burst transfer • Relative channel priorities can be set • Selection of dual or single address mode transfer • Chain mode transfer possible • Transfer data width: 8/16/32 bits • 4-Gbyte address space, maximum 4G (4,294,967,296) transfers • $\overline{\text{TEND}}$ output can be asserted for each channel at the end of DMA transfer
Timer pulse unit (TPU) (6 channels)	• Maximum 16 kinds of waveform output or maximum 16 kinds of input/output processing based on six 16-bit timer channels • 16 dual-function output compare registers/input capture registers • Total of 16 independent comparators • Selection of eight counter input clocks • Input capture function • Pulse output modes — One-shot, toggle, PWM • Phase counting mode — Two-phase encoder count processing capability
Motor management timer (MMT) (1 channel)	• Non-overlap waveform output for 6-phase inverter control • Dead times generated by dead time counters • Any PWM duty from 0% to 100% can be set • Toggle output possible in synchronization with PWM cycle • Data transfer can be performed by DMAC activation • A/D converter conversion start trigger can be generated • Output-off functions

Item	Specifications
Compare-match timer (CMT) (2 channels)	• 16-bit free-running counter • One compare register • Interrupt request generated by compare-match
Watchdog timer (WDT) (1 channel)	• Can be switched between watchdog timer and interval timer function • Internal reset, external signal, or interrupt generated by count overflow
Serial communication interface (SCI) (3 channels)	For each channel: • Selection of asynchronous or synchronous mode • Simultaneous transmission/reception (full-duplex) capability • Built-in dedicated baud rate generator • Multiprocessor communication function • Separate 16-stage FIFO registers for transmission and reception, enabling continuous high-speed communication • Selection of MSB-first or LSB-first transfer • Selection of base clock of 4/8/16 times the bit rate in asynchronous mode • Built-in IrDA interface (conforming to IrDA 1.0)
I/O ports	• Total of 118 port pins: 110 input/output, 8 input • Input/output voltage level for some ports can be set by I/O circuit power supply PV_{CC}
A/D converter	• 10 bits × 4 channels × 2 modules • Conversion can be activated by external trigger
D/A converter	• 8 bits × 2 channels
On-chip memory	• ROM: 256 kbytes • X-RAM: 4 kbytes • Y-RAM: 4 kbytes

Item	Specifications
Operating modes	• Operating modes — Expanded ROMless mode — Expanded ROM mode — Single-chip mode • Processing states — Program execution state — Exception handling state — Bus-released state • Power-down modes — Sleep mode — Hardware standby mode — Software standby mode — Module standby function — Module clock division function
Clock pulse generator (CPG)	• Built-in clock pulse generator • Selection of crystal or external clock as clock source • Built-in clock-multiplication PLL circuits • Built-in PLL circuit for phase synchronization between external clock and internal clock • Internal clock and on-chip peripheral module clock frequencies can be scaled independently
Package	176-pin plastic LQFP (LQFP2424-176), 0.5 mm pitch
Product lineup	SH7065: 256 kB flash/mask Operating frequency: 60 MHz (max.)

1.2 Block Diagram

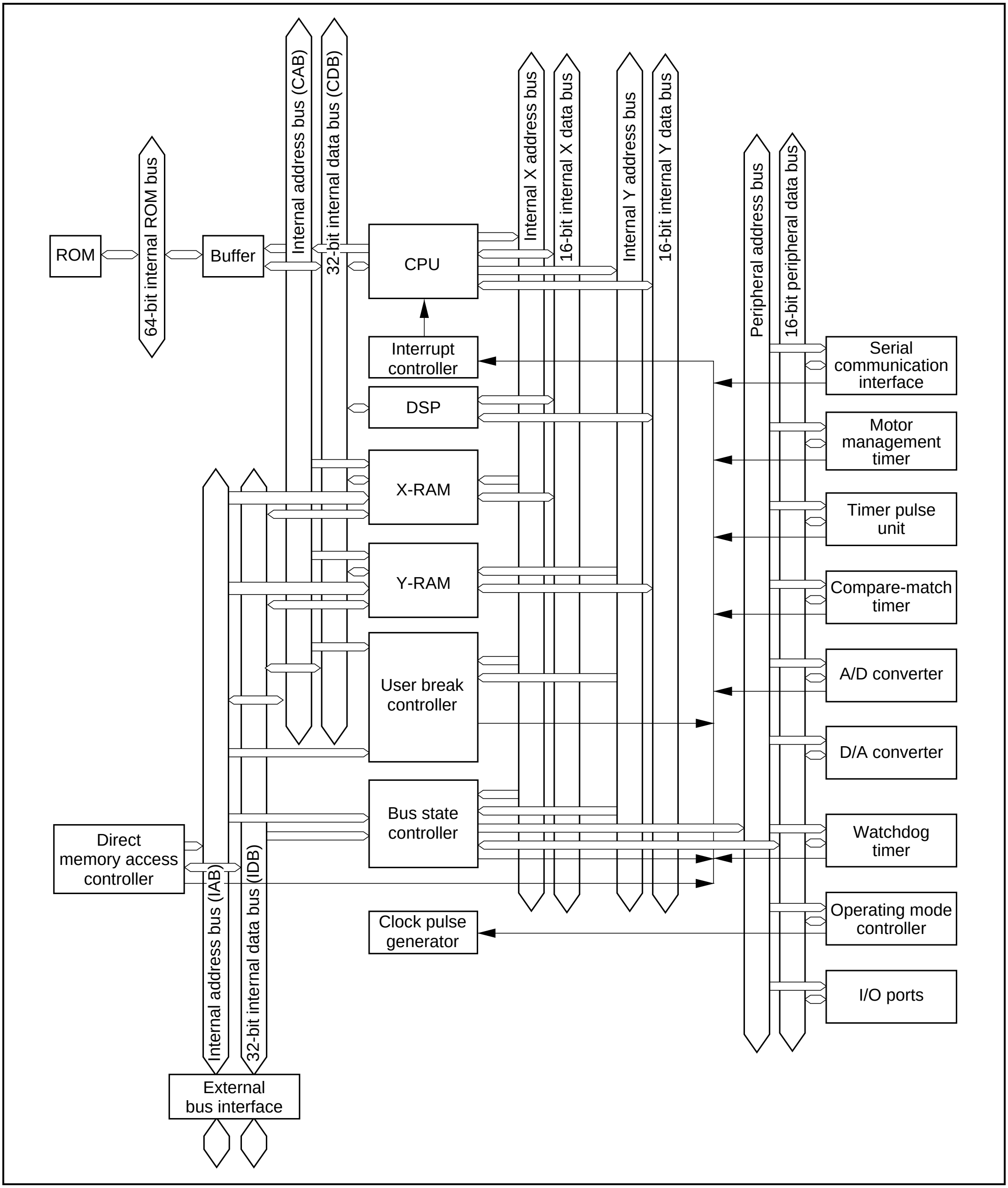
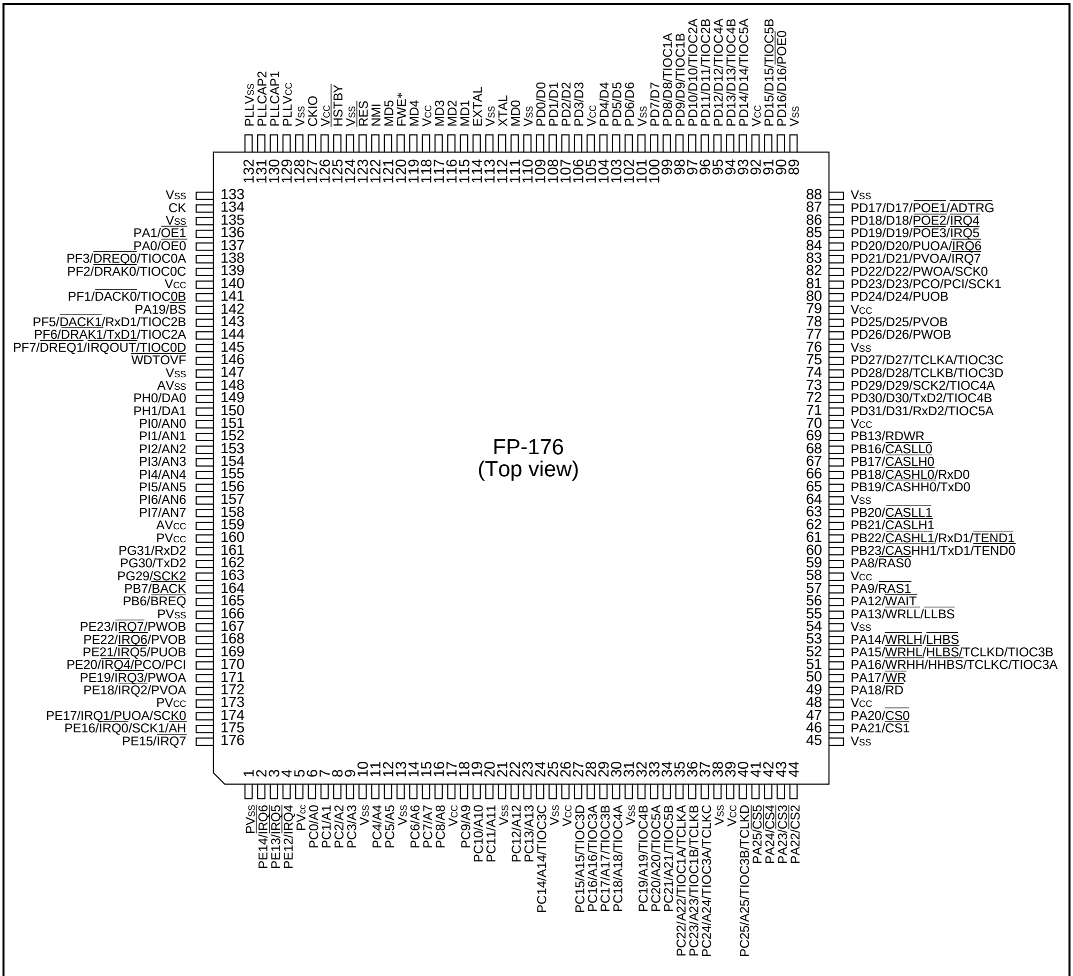


Figure 1.1 Block Diagram

1.3 Pin Arrangement and Pin Functions

1.3.1 Pin Arrangement



Note: * Vss in the mask version (can be pulled down with a resistance of 5.0 kΩ to 10 kΩ)

Figure 1.2 Pin Arrangement

1.3.2 Pin Functions

Table 1.2 summarizes the pin functions.

Table 1.2 Pin Functions

Type	Symbol	I/O	Name	Function
Power supply	Vcc	Input	Power supply	For connection to the power supply. Connect all V_{CC} pins to the system power supply. The chip will not operate if there are any open pins. Apply the same voltage to all V_{CC} pins.*
	Vss	Input	Ground	For connection to ground. Connect all V_{SS} pins to the system ground. The chip will not operate if there are any open pins.
	PVcc	Input	I/O circuit power supply	Power supply for the I/O circuits. The chip will not operate if there are any open pins. Apply the same voltage to all PV_{CC} pins.*
	PVss	Input	I/O circuit ground	Ground for the I/O circuits. The chip will not operate if there are any open pins.
Clock	PLLVcc	Input	PLL power supply	On-chip PLL oscillator power supply. The chip will not operate if there are any open pins.
	PLLVss	Input	PLL ground	On-chip PLL oscillator ground. The chip will not operate if there are any open pins.
	PLLCAP1	Input	PLL capacitance	On-chip PLL oscillator 1 external capacitance pin.
	PLLCAP2	Input	PLL capacitance	On-chip PLL oscillator 2 external capacitance pin.
	EXTAL	Input	External clock	For connection to a crystal resonator. An external clock can also be input to the EXTAL pin.
	XTAL	Output	Crystal	For connection to a crystal resonator
	CKIO	I/O	System clock I/O	Used as external clock input or internal clock output pin.
	CK	Output	System clock output	Internal clock output pin.

Type	Symbol	I/O	Name	Function
System control	$\overline{\text{RES}}$	Input	Power-on reset	Executes a power-on reset when driven low.
	$\overline{\text{WDTOVF}}$	Output	Watchdog timer overflow	WDT overflow output signal
	$\overline{\text{BREQ}}$	Input	Bus request	Driven low when an external device requests release of the bus.
	$\overline{\text{BACK}}$	Output	Bus request acknowledge	Indicates that the bus has been granted to an external device. The device that output the $\overline{\text{BREQ}}$ signal recognizes that the bus has been acquired when it receives the $\overline{\text{BACK}}$ signal.
	$\overline{\text{HSTBY}}$	Input	Hardware standby	Hardware standby input pin. Drive high when not used.
Operating mode control	MD0–MD5	Input	Mode setting	These pins determine the operating mode. Do not change the input values during operation.
	FWE	Input	Flash write enable	On-chip flash memory program/erase hardware protection pin.
Interrupts	NMI	Input	Nonmaskable interrupt	Nonmaskable interrupt request pin. Acceptance at the rising edge or falling edge can be selected.
	$\overline{\text{IRQ0}}$–$\overline{\text{IRQ7}}$	Input	Interrupt request 0 to 7	Maskable interrupt request pins. Level input or edge input can be selected.
	$\overline{\text{IRQOUT}}$	Output	Interrupt request output	Indicates that an interrupt request has been generated. Enables interrupt generation to be recognized in the bus-released state.
Address bus	A0–A25	Output	Address bus	Address output pins.
Data bus	D0–D31	I/O	Data bus	32-bit bidirectional data bus.
Bus control	$\overline{\text{CS0}}$–$\overline{\text{CS5}}$	Output	Chip select 0 to 5	Chip select signals for external memory or devices.
	$\overline{\text{RD}}$	Output	Read	Indicates reading from an external device.
	RDWR	Output	Read/write	Used as the DRAM write directive signal.
	$\overline{\text{WRLL}}$	Output	LL write	Indicates writing of bits 7 to 0 of external data.
	$\overline{\text{WRLH}}$	Output	LH write	Indicates writing of bits 15 to 8 of external data.

Type	Symbol	I/O	Name	Function
Bus control	$\overline{\mathrm{WRHL}}$	Output	HL write	Indicates writing of bits 23 to 16 of external data.
	$\overline{\mathrm{WRHH}}$	Output	HH write	Indicates writing of bits 31 to 24 of external data.
	$\overline{\mathrm{WAIT}}$	Input	Wait	Input for wait cycle insertion in bus cycles during external space access
	$\overline{\mathrm{LLBS}}$	Output	LL byte strobe	Indicates access to bits 7 to 0 of external data.
	$\overline{\mathrm{LHBS}}$	Output	LH byte strobe	Indicates access to bits 15 to 8 of external data.
	$\overline{\mathrm{HLBS}}$	Output	HL byte strobe	Indicates access to bits 23 to 16 of external data.
	$\overline{\mathrm{HHBS}}$	Output	HH byte strobe	Indicates access to bits 31 to 24 of external data.
	$\overline{\mathrm{WR}}$	Output	Write	Indicates the data bus input/output direction. Also used as the write directive for byte-strobe type memory.
	$\overline{\mathrm{RAS0}}$–$\overline{\mathrm{RAS1}}$	Output	Row address strobe 0, 1	DRAM row address strobe timing signals
	$\overline{\mathrm{CASLL0}}$–$\overline{\mathrm{CASLL1}}$	Output	LL column address strobe 0, 1	Output when accessing bits 7 to 0 of DRAM data.
	$\overline{\mathrm{CASLH0}}$–$\overline{\mathrm{CASLH1}}$	Output	LH column address strobe 0, 1	Output when accessing bits 15 to 8 of DRAM data.
	$\overline{\mathrm{CASHL0}}$–$\overline{\mathrm{CASHL1}}$	Output	HL column address strobe 0, 1	Output when accessing bits 23 to 16 of DRAM data.
	$\overline{\mathrm{CASHH0}}$–$\overline{\mathrm{CASHH1}}$	Output	HH column address strobe 0, 1	Output when accessing bits 31 to 24 of DRAM data.
	$\overline{\mathrm{OE0}}$–$\overline{\mathrm{OE1}}$	Output	Output enable 0, 1	Output enable signal for use of EDO DRAM in RAS down mode.
	$\overline{\mathrm{AH}}$	Output	Address hold	Address hold timing signal for a device using a multiplexed address/data bus.
	$\overline{\mathrm{BS}}$	Output	Bus cycle start	Indicates the start of a bus cycle.

Type	Symbol	I/O	Name	Function
Direct memory access controller (DMAC)	$\overline{\text{DREQ0}}$– $\overline{\text{DREQ1}}$	Input	DMA transfer request (channels 0, 1)	Input pins for external requests for DMA transfer.
	$\overline{\text{DRAK0}}$– $\overline{\text{DRAK1}}$	Output	DREQ request acknowledgment (channels 0, 1)	These pins output the input sampling acknowledgment for external requests for DMA transfer.
	$\overline{\text{DACK0}}$– $\overline{\text{DACK1}}$	Output	DMA transfer strobe (channels 0, 1)	These pins output a strobe to the external I/O in external DMA transfer requests.
	$\overline{\text{TEND0}}$– $\overline{\text{TEND1}}$	Output	DMA transfer end (channels 0, 1)	These pins go low at the end of DMA transfer.
Timer pulse unit (TPU)	TCLKA– TCLKD	Input	TPU timer clock input	TPU counter external clock Input pins.
	TIOC0A– TIOC0D	I/O	TPU input capture/output compare (channel 0)	Channel 0 input capture input/output compare output/PWM output pins.
	TIOC1A– TIOC1B	I/O	TPU input capture/output compare (channel 1)	Channel 1 input capture input/output compare output/PWM output pins.
	TIOC2A– TIOC2B	I/O	TPU input capture/output compare (channel 2)	Channel 2 input capture input/output compare output/PWM output pins.
	TIOC3A– TIOC3D	I/O	TPU input capture/output compare (channel 3)	Channel 3 input capture input/output compare output/PWM output pins.
	TIOC4A– TIOC4B	I/O	TPU input capture/output compare (channel 4)	Channel 4 input capture input/output compare output/PWM output pins.
	TIOC5A– TIOC5B	I/O	TPU input capture/output compare (channel 5)	Channel 5 input capture input/output compare output/PWM output pins.

Type	Symbol	I/O	Name	Function
Motor management timer (MMT)	PCI	Input	Counter clear input	Counter clear input pin.
	PCO	Output	PWM cycle output	Pin for toggle output synchronized with PWM cycle.
	PUOA–PUOB	Output	PWM U-phase output	PWM U-phase waveform output pin.
	PVOA–PVOB	Output	PWM V-phase output	PWM V-phase waveform output pin.
	PWOA–PWOB	Output	PWM W-phase output	PWM W-phase waveform output pin.
	$\overline{\text{POE0}}$–$\overline{\text{POE3}}$	Input	Port output enable input	These pins input request signals to place large-current pins in the high-impedance state.
Serial communication interface (SCI)	TxD0–TxD2	Output	Transmit data (channels 0 to 2)	Transmit data output pins.
	RxD0–RxD2	Input	Receive data (channels 0 to 2)	Receive data input pins.
	SCK0–SCK2	I/O	Serial clock (channels 0 to 2)	Clock input/output pins.
Analog power supply	AVcc	Input	Analog power supply	For connection to analog power supply.
	AVss	Input	Analog ground	For connection to analog power supply ground.
A/D converter	AN0–AN7	Input	Analog input	Analog signal input pins
	$\overline{\text{ADTRG}}$	Input	A/D conversion trigger input	External input for starting A/D conversion
D/A converter	DA0–DA1	Output	Analog output	D/A converter analog signal output pins

Type	Symbol	I/O	Name	Function
I/O ports	PA × 18	I/O	General port	General input/output port pins. Input or output can be specified bit by bit.
	PB × 11	I/O	General port	General input/output port pins. Input or output can be specified bit by bit.
	PC × 26	I/O	General port	General input/output port pins. Input or output can be specified bit by bit.
	PD × 32	I/O	General port	General input/output port pins. Input or output can be specified bit by bit.
	PE × 12	I/O	General port	General input/output port pins. Input or output can be specified bit by bit.
	PF × 6	I/O	General port	General input/output port pins. Input or output can be specified bit by bit.
	PG × 3	I/O	General port	General input/output port pins. Input or output can be specified bit by bit.
	PH × 2	I/O	General port	General input/output port pins. Input or output can be specified bit by bit.
	PI × 8	Input	General port	General input port pins.

Notes: Unused input pins must be pulled up or pulled down with a resistance of 4.7 kΩ to 10 kΩ.

* The following power-on/power-off order is recommended when applying a 5 V voltage to power supply voltage pin PV_{CC}. When PV_{CC} is also used with the same 3 V voltage as V_{CC}, etc., simultaneous powering on and off is recommended for all power supplies.

1. Powering on

   (1) Turn on the 5 V power (PV_{CC}) first, then the 3 V power (V_{CC}, $PLLV_{CC}$, AV_{CC}).

   (2) Pin states are undefined while only 5 V power (PV_{CC}) is on, as reset input is invalid.

2. Powering off

   (1) Power off in the reverse order to powering on: Turn off the 3 V power first, then the 5 V power.

   (2) Pin states are undefined while only 5 V power is being supplied.

3. Power-on/off interval

   To minimize the length of time during which pin states are undefined, the power-on/off interval should be kept as short as possible. Also, the system design should ensure that erroneous system operation will not result from pin states becoming undefined.

Table 1.3 Pin Function List

No.*	Control Power Supply	Function 1	Function 2	Function 3	Function 4	Function 5
1	—	PLLVcc	—	—	—	—
2	—	PLVss	—	—	—	—
3	—	PLLCAP1	—	—	—	—
4	—	PLLCAP2	—	—	—	—
5	—	AVcc	—	—	—	—
6	—	AVss	—	—	—	—
7	Vcc	EXTAL	—	—	—	—
8	Vcc	XTAL	—	—	—	—
9	Vcc	CKIO	—	—	—	—
10	Vcc	CK	—	—	—	—
11	Vcc	$\overline{\text{RES}}$	—	—	—	—
12	Vcc	$\overline{\text{WDTOVF}}$	—	—	—	—
13	Vcc	$\overline{\text{HSTBY}}$	—	—	—	—
14	Vcc	MD5	—	—	—	—
15	Vcc	MD4	—	—	—	—
16	Vcc	MD3	—	—	—	—
17	Vcc	MD2	—	—	—	—
18	Vcc	MD1	—	—	—	—
19	Vcc	MD0	—	—	—	—
20	Vcc	NMI	—	—	—	—
21	Vcc	FWE	—	—	—	—
22	Vcc	General input/output (PA25)	$\overline{\text{CS5}}$	—	—	—
23	Vcc	General input/output (PA24)	$\overline{\text{CS4}}$	—	—	—
24	Vcc	General input/output (PA23)	$\overline{\text{CS3}}$	—	—	—
25	Vcc	General input/output (PA22)	$\overline{\text{CS2}}$	—	—	—
26	Vcc	General input/output (PA21)	$\overline{\text{CS1}}$	—	—	—
27	Vcc	General input/output (PA20)	$\overline{\text{CS0}}$	—	—	—
28	Vcc	General input/output (PA19)	$\overline{\text{BS}}$	—	—	—
29	Vcc	General input/output (PA18)	$\overline{\text{RD}}$	—	—	—

No.*	Control Power Supply	Function 1	Function 2	Function 3	Function 4	Function 5
30	Vcc	General input/output (PA17)	$\overline{\text{WR}}$	—	—	—
31	Vcc	General input/output (PA16)	$\overline{\text{WRHH}}$	$\overline{\text{HHBS}}$	TCLKC	TIOC3A
32	Vcc	General input/output (PA15)	$\overline{\text{WRHL}}$	$\overline{\text{HLBS}}$	TCLKD	TIOC3B
33	Vcc	General input/output (PA14)	$\overline{\text{WRLH}}$	$\overline{\text{LHBS}}$	—	—
34	Vcc	General input/output (PA13)	$\overline{\text{WRLL}}$	$\overline{\text{LLBS}}$	—	—
35	Vcc	General input/output (PA12)	$\overline{\text{WAIT}}$	—	—	—
36	Vcc	General input/output (PA9)	$\overline{\text{RAS1}}$	—	—	—
37	Vcc	General input/output (PA8)	$\overline{\text{RAS0}}$	—	—	—
38	Vcc	General input/output (PB23)	$\overline{\text{CASHH1}}$	TxD1	$\overline{\text{TEND0}}$	—
39	Vcc	General input/output (PB22)	$\overline{\text{CASHL1}}$	RxD1	$\overline{\text{TEND1}}$	—
40	Vcc	General input/output (PB21)	$\overline{\text{CASLH1}}$	—	—	—
41	Vcc	General input/output (PB20)	$\overline{\text{CASLL1}}$	—	—	—
42	Vcc	General input/output (PB19)	$\overline{\text{CASHH0}}$	TxD0	—	—
43	Vcc	General input/output (PB18)	$\overline{\text{CASHL0}}$	RxD0	—	—
44	Vcc	General input/output (PB17)	$\overline{\text{CASLH0}}$	—	—	—
45	Vcc	General input/output (PB16)	$\overline{\text{CASLL0}}$	—	—	—
46	Vcc	General input/output (PB13)	RDWR	—	—	—
47	Vcc	General input/output (PC25)	A25	TIOC3B	TCLKD	—
48	Vcc	General input/output (PC24)	A24	TIOC3A	TCLKC	—
49	Vcc	General input/output (PC23)	A23	TIOC1B	TCLKB	—
50	Vcc	General input/output (PC22)	A22	TIOC1A	TCLKA	—
51	Vcc	General input/output (PC21)	A21	TIOC5B	—	—
52	Vcc	General input/output (PC20)	A20	TIOC5A	—	—
53	Vcc	General input/output (PC19)	A19	TIOC4B	—	—
54	Vcc	General input/output (PC18)	A18	TIOC4A	—	—
55	Vcc	General input/output (PC17)	A17	TIOC3B	—	—
56	Vcc	General input/output (PC16)	A16	TIOC3A	—	—
57	Vcc	General input/output (PC15)	A15	TIOC3D	—	—
58	Vcc	General input/output (PC14)	A14	TIOC3C	—	—
59	Vcc	General input/output (PC13)	A13	—	—	—
60	Vcc	General input/output (PC12)	A12	—	—	—

No.*	Control Power Supply	Function 1	Function 2	Function 3	Function 4	Function 5
61	Vcc	General input/output (PC11)	A11	—	—	—
62	Vcc	General input/output (PC10)	A10	—	—	—
63	Vcc	General input/output (PC9)	A9	—	—	—
64	Vcc	General input/output (PC8)	A8	—	—	—
65	Vcc	General input/output (PC7)	A7	—	—	—
66	Vcc	General input/output (PC6)	A6	—	—	—
67	Vcc	General input/output (PC5)	A5	—	—	—
68	Vcc	General input/output (PC4)	A4	—	—	—
69	Vcc	General input/output (PC3)	A3	—	—	—
70	Vcc	General input/output (PC2)	A2	—	—	—
71	Vcc	General input/output (PC1)	A1	—	—	—
72	Vcc	General input/output (PC0)	A0	—	—	—
73	Vcc	General input/output (PD31)	D31	RxD2	TIOC5A	—
74	Vcc	General input/output (PD30)	D30	TxD2	TIOC4B	—
75	Vcc	General input/output (PD29)	D29	SCK2	TIOC4A	—
76	Vcc	General input/output (PD28)	D28	TCLKB	TIOC3D	—
77	Vcc	General input/output (PD27)	D27	TCLKA	TIOC3C	—
78	Vcc	General input/output (PD26)	D26	PWOB	—	—
79	Vcc	General input/output (PD25)	D25	PVOB	—	—
80	Vcc	General input/output (PD24)	D24	PUOB	—	—
81	Vcc	General input/output (PD23)	D23	PCO	PCI	SCK1
82	Vcc	General input/output (PD22)	D22	PWOA	SCK0	—
83	Vcc	General input/output (PD21)	D21	PVOA	$\overline{\text{IRQ7}}$	—
84	Vcc	General input/output (PD20)	D20	PUOA	$\overline{\text{IRQ6}}$	—
85	Vcc	General input/output (PD19)	D19	$\overline{\text{POE3}}$	$\overline{\text{IRQ5}}$	—
86	Vcc	General input/output (PD18)	D18	$\overline{\text{POE2}}$	$\overline{\text{IRQ4}}$	—
87	Vcc	General input/output (PD17)	D17	$\overline{\text{POE1}}$	$\overline{\text{ADTRG}}$	—
88	Vcc	General input/output (PD16)	D16	$\overline{\text{POE0}}$	—	—
89	Vcc	General input/output (PD15)	D15	TIOC5B	—	—
90	Vcc	General input/output (PD14)	D14	TIOC5A	—	—

No.*	Control Power Supply	Function 1	Function 2	Function 3	Function 4	Function 5
91	Vcc	General input/output (PD13)	D13	TIOC4B	—	—
92	Vcc	General input/output (PD12)	D12	TIOC4A	—	—
93	Vcc	General input/output (PD11)	D11	TIOC2B	—	—
94	Vcc	General input/output (PD10)	D10	TIOC2A	—	—
95	Vcc	General input/output (PD9)	D9	TIOC1B	—	—
96	Vcc	General input/output (PD8)	D8	TIOC1A	—	—
97	Vcc	General input/output (PD7)	D7	—	—	—
98	Vcc	General input/output (PD6)	D6	—	—	—
99	Vcc	General input/output (PD5)	D5	—	—	—
100	Vcc	General input/output (PD4)	D4	—	—	—
101	Vcc	General input/output (PD3)	D3	—	—	—
102	Vcc	General input/output (PD2)	D2	—	—	—
103	Vcc	General input/output (PD1)	D1	—	—	—
104	Vcc	General input/output (PD0)	D0	—	—	—
105	Vcc	General input/output (PA1)	$\overline{\text{OE1}}$	—	—	—
106	Vcc	General input/output (PA0)	$\overline{\text{OE0}}$	—	—	—
107	PVcc	General input/output (PE23)	$\overline{\text{IRQ7}}$	PWOB	—	—
108	PVcc	General input/output (PE22)	$\overline{\text{IRQ6}}$	PVOB	—	—
109	PVcc	General input/output (PE21)	$\overline{\text{IRQ5}}$	PUOB	—	—
110	PVcc	General input/output (PE20)	$\overline{\text{IRQ4}}$	PCO	PCI	—
111	PVcc	General input/output (PE19)	$\overline{\text{IRQ3}}$	PWOA	—	—
112	PVcc	General input/output (PE18)	$\overline{\text{IRQ2}}$	PVOA	—	—
113	PVcc	General input/output (PE17)	$\overline{\text{IRQ1}}$	PUOA	SCK0	—
114	PVcc	General input/output (PE16)	$\overline{\text{IRQ0}}$	SCK1	$\overline{\text{AH}}$	—
115	Vcc	General input/output (PF7)	$\overline{\text{DREQ1}}$	$\overline{\text{IRQOUT}}$	TIOC0D	—
116	Vcc	General input/output (PF6)	$\overline{\text{DRAK1}}$	TxD1	TIOC2A	—
117	Vcc	General input/output (PF5)	$\overline{\text{DACK1}}$	RxD1	TIOC2B	—
118	AVcc	General input (PI7)	AN7	—	—	—
119	AVcc	General input (PI6)	AN6	—	—	—
120	AVcc	General input (PI5)	AN5	—	—	—

No.*	Control Power Supply	Function 1	Function 2	Function 3	Function 4	Function 5
121	AVcc	General input (PI4)	AN4	—	—	—
122	AVcc	General input (PI3)	AN3	—	—	—
123	AVcc	General input (PI2)	AN2	—	—	—
124	AVcc	General input (PI1)	AN1	—	—	—
125	AVcc	General input (PI0)	AN0	—	—	—
126	AVcc	General input/output (PH1)	DA1	—	—	—
127	AVcc	General input/output (PH0)	DA0	—	—	—
128	PVcc	General input/output (PE12)	$\overline{\text{IRQ4}}$	—	—	—
129	PVcc	General input/output (PE13)	$\overline{\text{IRQ5}}$	—	—	—
130	PVcc	General input/output (PE14)	$\overline{\text{IRQ6}}$	—	—	—
131	PVcc	General input/output (PE15)	$\overline{\text{IRQ7}}$	—	—	—
132	PVcc	General input/output (PG31)	RxD2	—	—	—
133	PVcc	General input/output (PG30)	TxD2	—	—	—
134	PVcc	General input/output (PG29)	SCK2	—	—	—
135	Vcc	General input/output (PF2)	$\overline{\text{DRAK0}}$	TIOC0C	—	—
136	Vcc	General input/output (PF1)	$\overline{\text{DACK0}}$	TIOC0B	—	—
137	Vcc	General input/output (PF3)	$\overline{\text{DREQ0}}$	TIOC0A	—	—
138	PVcc	General input/output (PB7)	$\overline{\text{BACK}}$	—	—	—
139	PVcc	General input/output (PB6)	$\overline{\text{BREQ}}$	—	—	—

Note: * These numbers are not the package pin numbers.

Section 2 CPU

2.1 Register Configuration

The SH7065 has sixteen 32-bit general registers, six 32-bit control registers, and ten 32-bit system registers.

As the SH7065 is upward-compatible with the SH-1 and SH-2 at the object code level, a number of registers have been added to those provided in previous SuperH microcomputers. The additions comprise three control registers (the repeat start register (RS), repeat end register (RE), and modulo register (MOD)), one system register (the DSP status register (DSR)), and six registers (A0, A1, X0, X1, Y0, and Y1) within the DSP data registers.

With SuperH microcomputer type instructions, general registers are used in the same way as in the SH-1 and SH-2, but with DSP type instructions, general registers are used as address and index registers for accessing memory.

2.1.1 General Registers

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

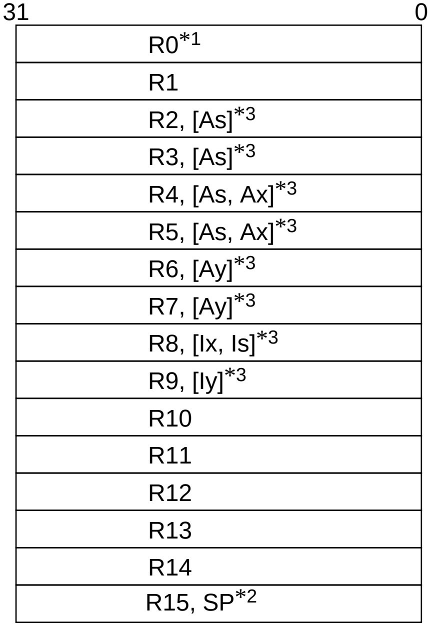
With SuperH microcomputer type instructions, R0 is used as an index register. With a number of instructions, R0 is the only register that can be used. R15 is used as the stack pointer (SP). In exception handling, R15 is used to reference the stack when saving and restoring the status register (SR) and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

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

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

DSP type instructions can access can access X and Y data memory simultaneously. Two sets of address pointers are provided to specify the X and Y data memory addresses.

The general registers are shown in figure 2.1.



31 0

R0*1

R1

R2, [As]*3

R3, [As]*3

R4, [As, Ax]*3

R5, [As, Ax]*3

R6, [Ay]*3

R7, [Ay]*3

R8, [Ix, Is]*3

R9, [Iy]*3

R10

R11

R12

R13

R14

R15, SP*2

Notes: 1. The R0 register is used as the index register in indexed register indirect addressing mode and indexed GBR indirect addressing mode.
With certain instructions, R0 only is used as the source register and destination register.
2. The R15 register is used as the stack pointer (SP) during exception handling.
3. Used as the memory address register or memory index register with DSP type instructions.

Figure 2.1 General Register Configuration

In assembler, the symbols R2, R3 ... R9 are used. If it is wished to use a name that indicates the role of a register for DSP type instructions, a different register name (alias) can be used. The coding in assembler is as follows.

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

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

```
Ax0:  .REG  (R4)
Ax1:  .REG  (R5)
Ix:   .REG  (R8)
Ay0:  .REG  (R6)
Ay1:  .REG  (R7)
Iy:   .REG  (R9)
As0:  .REG  (R4); Definition when an alias is required for single data transfer.
As1:  .REG  (R5); Definition when an alias is required for single data transfer.
As2:  .REG  (R2); Definition when an alias is required for single data transfer.
As3:  .REG  (R3); Definition when an alias is required for single data transfer.
Is:   .REG  (R8); Definition when an alias is required for single data transfer.
```

2.1.2 Control Registers

There are six 32-bit control registers: the status register (SR), repeat start register (RS), repeat end register (RE), global base register (GBR), vector base register (VBR), and modulo register (MOD).

The SR register shows the processing status.

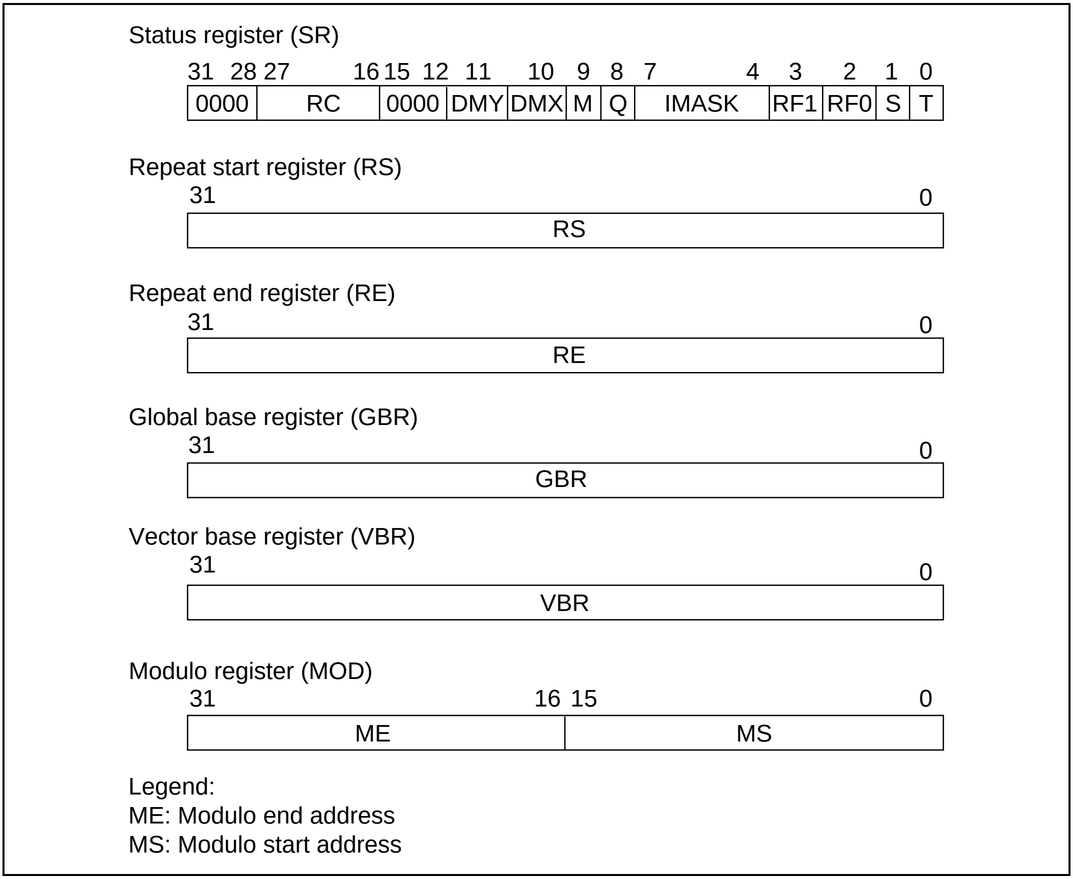
The GBR register is used as the base address in GBR indirect addressing mode, and is used for data transfer involving on-chip peripheral module registers, etc.

The VBR register is used as the base address of the exception handling vector area, including interrupts.

The RS register and RE register are used to control program repeats (loops). The number of loops is specified in the repeat counter (RC) in the SR register, the repeat start address is specified in the RS register, and the repeat end address is specified in the RE register. However, the address values stored in the RS register and RE register are not necessarily the same as the physical repeat start address and end address.

The MOD register is used in modulo addressing for repeat data buffering. The modulo addressing specification is made with the DMX or DMY bit in the SR register, the modulo end address (ME) is specified in the upper 16 bits of the MOD register, and the modulo start address (MS) in the lower 16 bits. The DMX and DMY bits cannot both specify modulo addressing simultaneously. Modulo addressing can be used with the X and Y data transfer instructions (MOVX, MOVY), but not with the single data transfer instruction (MOVS).

Figure 2.2 shows the control register, and table 2.1 shows the bits in the SR register.



Legend:
ME: Modulo end address
MS: Modulo start address

Figure 2.2 Control Register Configuration

Table 2.1 SR Register Bits

Bits	Name (Abbreviation)	Function
27–16	Repeat counter (RC)	These bits specify number of repeats in repeat (loop) control (2 to 4095).
11	Y pointer modulo addressing specification (DMY)	1: Modulo addressing mode is enabled for Y memory address pointer Ay (R6, R7).
10	X pointer modulo addressing specification (DMX)	1: Modulo addressing mode is enabled for X memory address pointer Ax (R4, R5).
9	M bit	Used by DIV0S/U and DIV1 instructions.
8	Q bit	
7–4	Interrupt request mask (IMASK)	These bits show the interrupt request acceptance level (0 to 15).
3, 2	Repeat flags (RF1, RF0)	Used for zero-overhead repeat (loop) control. Set as follows when the SETRC instruction is used. 1-step repeat: 00 RE – RS = –4 2-step repeat: 01 RE – RS = –2 3-step repeat: 11 RE – RS = 0 4 or more steps: 10 RE – RS > 0
1	Saturation operation bit (S)	Used with MAC and DSP instructions. 1: Specifies a saturation operation (preventing overflow)
0	T bit	With MOVT, CMP/cond, TAS, TST, BT, BT/S, BF, BF/S, SETT, CLRT, and DT instructions: 0: Indicates True 1: Indicates False
		With ADDV/C, SUBV/C, DIV0U/S, DIV1, NEGC, SHAR/L, SHLR/L, ROTR/L. and ROTCR/L instructions: 1: Indicates occurrence of carry, borrow, overflow, or underflow
31–28, 15–12	0 bits	0: Always read as 0. Only 0 should be written to these bits.

Special load/store instructions are provided for accessing the RS, RE, and MOD registers. For example, the coding for accessing the RS register is as follows.

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

The instructions for setting an address in the RS and RE registers for zero-overhead repeat control are as follows.

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

The GBR and VBR registers are the same as the previous SuperH microcomputer registers. In the SH7065, four control bits (DMX, DMY, RF1, and RF0) and an RC counter have been added to the SR register, and the RS, RE, and MOD registers are provided as new registers.

2.1.3 System Registers

There are four 32-bit system registers: the multiply and accumulate register high (MACH), multiply and accumulate register low (MACL), procedure register (PR), and program counter (PC).

MACH and MACL store the results of multiply or multiply and accumulate operations*, PR stores the return destination address of a subroutine procedure, and PC shows the executing program address and controls the processing flow. PC shows the address 4 bytes ahead of the currently executing instruction. These registers are the same as the SuperH microcomputer registers.

Note: * These registers are used only when executing an instruction supported by the SH-1 and SH-2. They are not used with the new multiply instruction provided in the SH-DSP (PMULS).

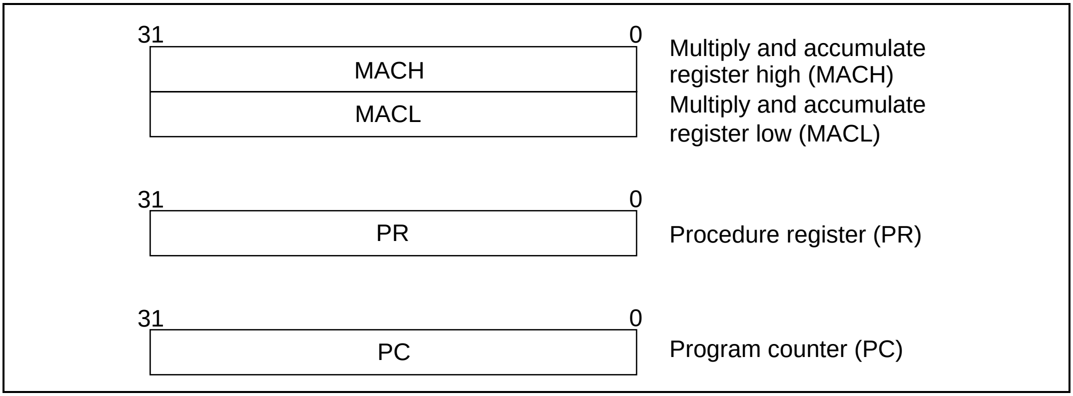

31
0
MACH
MACL
Multiply and accumulate register high (MACH)
Multiply and accumulate register low (MACL)
31
0
PR
Procedure register (PR)
31
0
PC
Program counter (PC)

Figure 2.3 System Register Configuration

In the SH7065, of the DSP unit registers (DSP registers) described below, the DSP status register (DSR) and five of the eight data registers (A0, X0, X1, Y0, and Y1) are treated as system registers. A0 is a 40-bit register, but when data is output from the A0 register the guard bit field (A0G) is ignored, and when data is input to the A0 register the MSB is copied into the guard bit field (A0G).

2.1.4 DSP Registers

The DSP unit has eight data registers and one control register as DSP registers.

The DSP data registers comprise two 40-bit registers, A0 and A1, and six 32-bit registers, M0, M1, X0, X1, Y0, and Y1. Registers A0 and A1 each have an 8-bit guard bit field, designated A0G and A1G, respectively.

The DSP data registers are used as DSP instruction operands in DSP data transfer and processing. Instructions that access the DSP data registers are of three types, for DSP data processing, and X and Y data transfer processing.

The control register is the 32-bit DSP status register (DSR), which shows operation results. The DSR register contains bits that indicate the result of an operation—the Signed Greater Than bit (GT), Zero Value bit (Z), Negative Value bit (N), Overflow bit (V), and DSP Condition bit (DC)—and also Condition Select bits (CS) that control the DC bit setting.

The DC bit is a status flag that closely resembles the T bit of the SuperH microcomputer CPU core. In the case of a conditional DSP type instruction, execution during DSP data processing is controlled in accordance with the DC bit. This control extends only to DSP unit execution, and only DSP registers are updated. It has no effect on address calculation or SuperH microcomputer

CPU core execution instructions such as load/store instructions. The CS control bits (bits 2 to 0) specify the conditions for setting the DC bit.

DSP type instructions include unconditional DSP type instructions and conditional DSP type instructions. In unconditional DSP type data processing, with the exception of the PMULS, MOVX, MOVY, and MOVS instructions, the status bits and DC bit are updated. Conditional DSP type instructions are executed in accordance with the DC bit setting, but the DSR register is not updated regardless of whether or not these instructions are executed.

The DSP registers are shown in figure 2.4, and the DSR register bit functions are summarized in table 2.2.

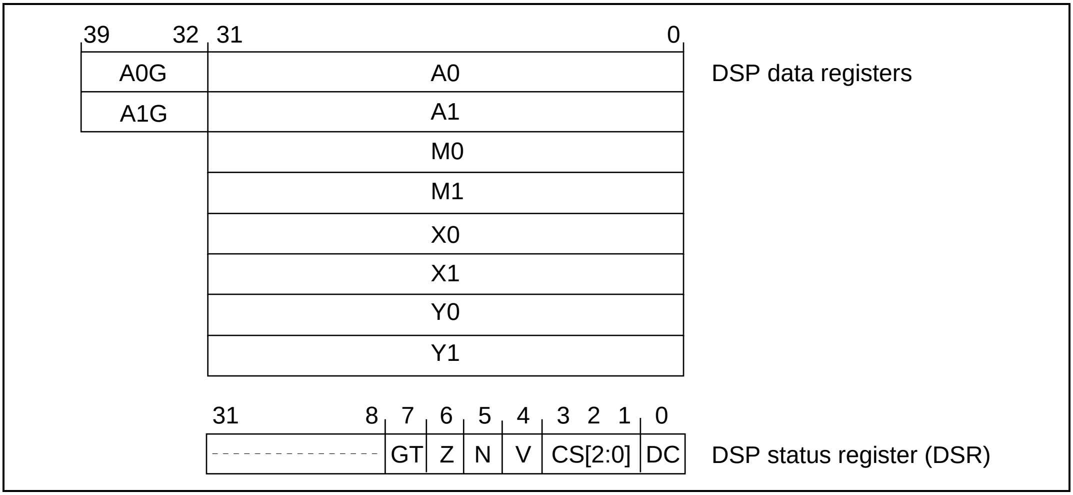


Figure 2.4   DSP Register Configuration

Table 2.2 DSR Register Bits

Bits	Name (Abbreviation)	Function
31–8	Reserved	0: Always read as 0. The write value should also be 0.
7	Signed Greater Than (GT)	Indicates that the operation result is positive (except zero) or that operand 1 is greater than operand 2. 1: Operation result is positive or operand 1 is greater than operand 2
6	Zero Value (Z)	Indicates that the operation result is zero (0) or that operand 1 is equal to operand 2. 1: Operation result is zero (0) or operands are equal
5	Negative Value (N)	Indicates that the operation result is negative or that operand 1 is smaller than operand 2. 1: Operation result is negative or operand 1 is smaller than operand 2
4	Overflow (V)	Indicates that the operation result has overflowed. 1: Operation result has overflowed
3–1	Condition Select (CS)	These bits specify the mode for selecting the operation result status to be set in the DC bit. Do not set these bits to 110 or 111. 000: Carry/borrow mode 001: Negative value mode 010: Zero mode 011: Overflow mode 100: Signed greater than mode 101: Signed greater than or equal to mode
0	DSP Condition (DC)	Sets the status of the operation result in the mode specified by the CS bits. 0: Specified mode status has not occurred (false) 1: Specified mode status has occurred

The DSR register is treated as a system register by CPU core instructions. The following load/store instructions are used for data transfer to and from the DSR register.

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

The A0, X0, X1, Y0, and Y1 registers are also treated as system registers by CPU core instructions. The following load/store instructions are used for data transfer to and from these registers.

```
STS    Dm,Rn;
STS.L  Dm,@-Rn;
LDS    Rn,Dm;
LDS.L  @Rn+,Dm;
(Dm: A0, X0, X1, Y0, or Y1)
```

2.1.5 Notes on Guard Bits and Overflow Treatment

Data operations in the DSP unit are basically 32-bit operations, but these operations are always executed with a 40-bit length including the 8-bit guard field. If the guard bit field does not match the value of the MSB of the 32-bit field, the operation result is treated as overflow. In this case, the N bit shows the correct status of the operation result regardless of whether or not overflow has occurred. This also applies when the destination operand is a 32-bit register. The 8-bit guard bit field is always assumed to present, and each status flag is updated.

If overflow occurs that prevents the result from being indicated correctly despite the use of the guard bits, the N flag will not be able to show the correct status.

2.1.6 Initial Register Values

Register values after a reset are shown in table 2.3.

Table 2.3 Initial Register Values

Type	Registers	Initial Value
General registers	R0–R14	Undefined
	R15 (SP)	SP value in vector address table
Control registers	SR	I3 to I0 = 1111 (H'F); reserved bits, RC, DMY, and DMX cleared to 0; other bits undefined
	RS RE	Undefined
	GBR	Undefined
	VBR	H'0000 0000
	MOD	Undefined
System registers	MACH, MACL, PR	Undefined
	PC	PC value in vector address table
DSP registers	A0, A0G, A1, A1G, M0, M1, X0, X1, Y0, Y1	Undefined
	DSR	H'0000 0000

2.2 Data Formats

2.2.1 Register Data Formats

The register operand data size is always longword (32 bits). When data in memory is loaded into a register, if the memory operand data size is byte (8 bits) or word (16 bits), it is sign-extended to longword length.

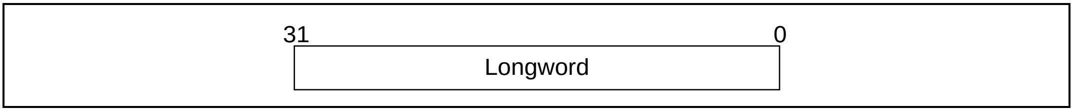


Figure 2.5 Register Data Format

2.2.2 Memory Data Formats

Byte, word, and longword data formats can be used.

Byte data can be located at any address, while word data must start at address 2n and longword data at address 4n. If data is accessed other than at these boundaries, an address error will result, and the result of the access cannot be guaranteed. In particular, since the program counter (PC) and status register (SR) are stored in longword format in the stack area indicated by the stack pointer (SP: R15), the setting musty be made so that stack pointer value is 4n.

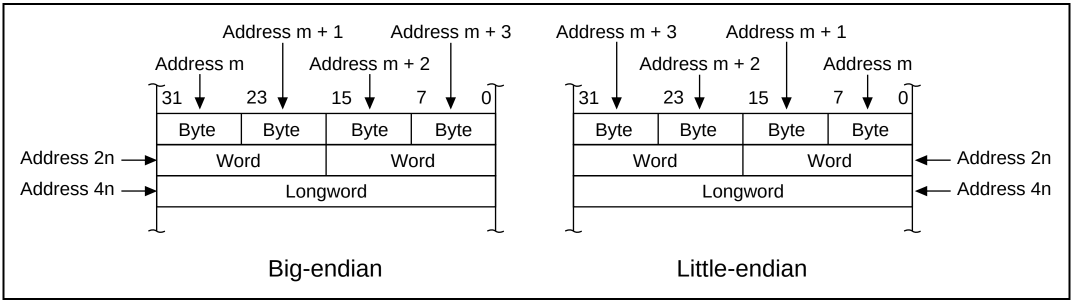


Figure 2.6 Memory Data Format

2.2.3 Immediate Data Formats

Byte immediate data is placed inside the instruction code.

With the MOV, ADD, and CMP/EQ instructions, immediate data is sign-extended and a longword operation is performed with a register. With the TST, AND, OR, and XOR instructions, on the other hand, a longword operation is performed after zero-extending the immediate data. Therefore, when immediate data is used with an AND instruction, the upper 24 bits of the destination register are always cleared.

Word and longword immediate data should be placed in a table in memory, not inside the instruction code. The table in memory should be referenced with an immediate data transfer instruction (MOV) using PC relative addressing mode with displacement.

2.2.4 DSP Type Data Formats

The SH7065 has three different data formats for instructions: fixed-point data format, integer data format, and logical data format.

In the DSP type fixed-point data format, there is a binary point between bit 31 and bit 30. There are three kinds of format—with guard bits, without guard bits, and multiplication input—each with a different valid bit length and range of expressable values.

In the DSP type integer data format, there is a binary point between bit 16 and bit 15. There are three kinds of format—with guard bits, without guard bits, and shift amount—each with a different valid bit length and range of expressable values. The shift amount for an arithmetic shift (PSHA) is a 7-bit area, and values from –64 to +63 can be expressed, but only values from –32 to +32 are actually valid. Similarly, the shift amount for a logical shift (PSHL) is a 6-bit area, but only values from –16 to +16 are actually valid.

There is no radix point in the DSP type logical data format.

The data format and valid data length are determined by the DSP register.

The three DSP type data formats and the position of the binary point in each are shown in figure 2.7, together with a SuperH type data format for reference.

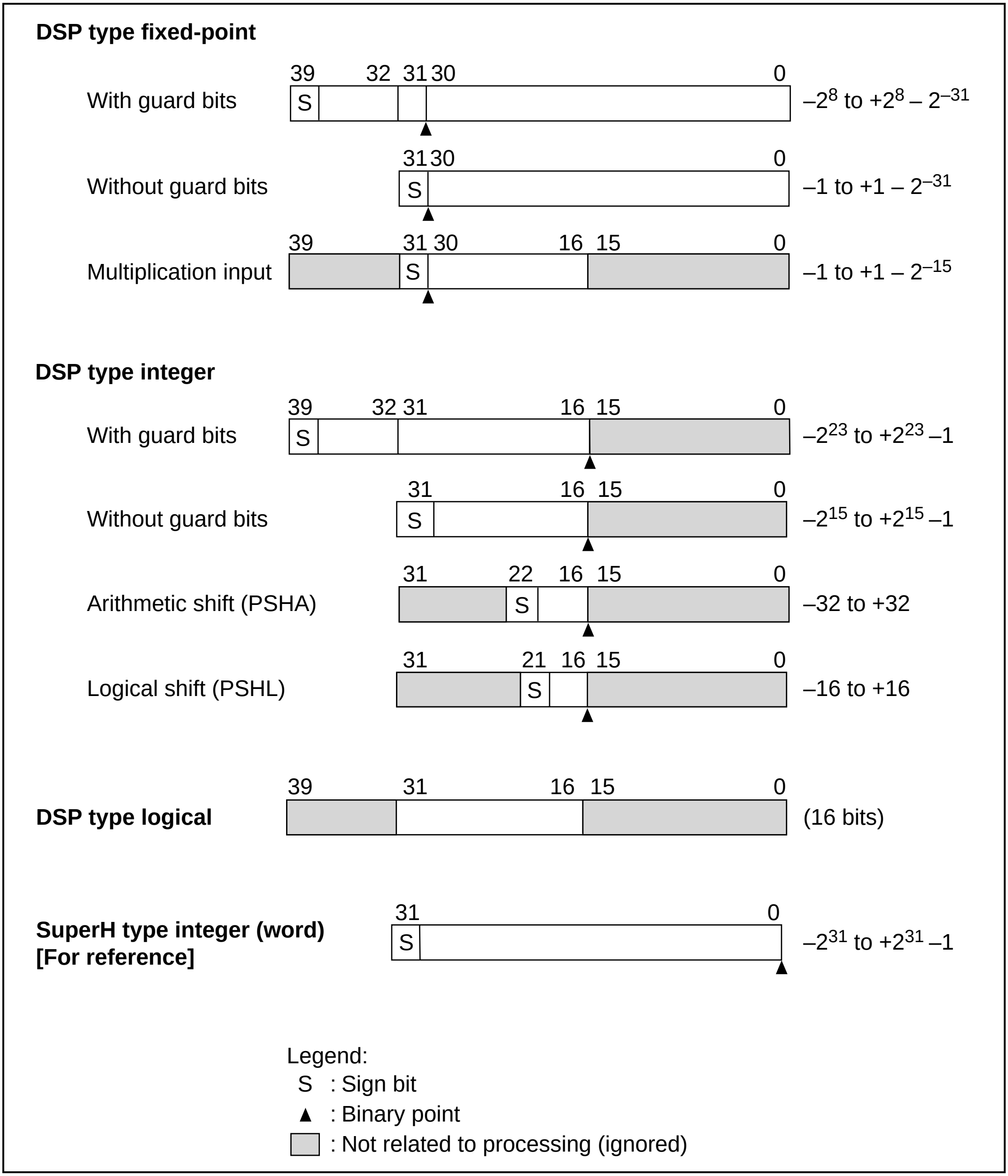


Figure 2.7 DSP Type Data Formats

2.2.5 DSP Type Instructions and Data Formats

The data format and valid data length are determined by the DSP type instruction and DSP register. There are three types of instruction that access DSP data registers: DSP data processing instructions, X and Y data transfer processing instructions, and single data transfer processing instructions.

DSP Data Processing: When the A0 or A1 register is used as the source register in DSP fixed-point data processing, the guard bits (bits 39 to 32) are valid. When a register other than A0 or A1 (register M0, M1, X0, X1, Y0, or Y1) is used as the source register, the sign-extension of that register data is used as the data in bits 39 to 32. When the A0 or A1 register is used as the destination register, the guard bits (bits 39 to 32) are valid. When a register other than A0 or A1 is used as the destination register, bits 39 to 32 of the result data are ignored.

In DSP integer data processing, the situation is the same as for DSP fixed-point data processing, except that the lower word (lower 16 bits: bits 15 to 0) of the source register is ignored, and the lower word of the destination register is cleared to 0.

In DSP logical data processing, the upper word (upper 16 bits: bits 31 to 16) of the source register is valid. The lower word and the guard bits of the A0 and A1 registers are ignored. The upper word of the destination register is valid. The lower word and the guard bits of the A0 and A1 registers are cleared to 0.

X and Y Data Transfer: The MOVX.W and MOVY.W instructions access X and Y memory via the 16-bit X and Y data buses. The data loaded into a register and the data stored from a register is always the upper word (upper 16 bits: bits 31 to 16).

In a load, MOVX.W loads X memory with the X0 or X1 register as the destination register, while MOVY.W loads Y memory with the Y0 or Y1 register as the destination register. Data is loaded into the upper word of the register, while the lower word is cleared to 0.

Data in the upper word of the A0 or A1 register can be stored in X or Y memory with a data transfer instruction, but data cannot be stored from any other register. The guard bits and lower word of the A0 or A1 register are ignored.

Single Data Transfer: The MOVS.W and MOVS.L instructions can access any memory via the data bus (CDB). All the DSP registers are connected to the CDB bus, and are used as the source and destination registers in a data transfer. There are two data transfer modes: word and longword. In word mode, with the exception of the A0G and A1G registers, a load is performed to, or store performed from, the upper word of a DSP register. In longword mode, with the exception of the A0G and A1G registers, a load is performed to, or store performed from, the 32 bits of a DSP register. In a single data transfer, the A0G and A1G registers can be handled as independent registers. The load and store data length for the A0G and A1G registers is 8 bits.

When a DSP register is used as the source register in word mode, if data is stored from a register other than A0G or A1G, the upper word of the register is transferred. In the case of the A0 and A1 registers, the guard bits are ignored. When the A0G or A1G register is used as the source register in word mode, only 8 bits of data are stored from the register, and the upper bits are sign-extended.

When a DSP register is used as the destination register in word mode, with the exception of the A0G and A1G registers, data is loaded into the upper word of the register. When data is loaded into a register other than A0G or A1G, the lower word of the register is cleared to 0. In the case of the A0 and A1 registers, the data sign is extended and loaded into the guard bits, and the lower word is cleared to 0. When the A0G or A1G register is used as the destination register in word mode, the lowest 8 bits of the data are loaded into the register, and the A0 or A1 register is not cleared to 0, but retains its prior value.

When a DSP register is used as the source register in longword mode, if data is stored from a register other than A0G or A1G, the 32 bits of the register are transferred. When the A0 or A1 register is used as the source register, the guard bits are ignored. When the A0G or A1G register is used as the source register in longword mode, only 8 bits of data are stored from the register, and the upper bits are sign-extended.

When a DSP register is used as the destination register in longword mode, with the exception of the A0G and A1G registers, data is loaded into the 32 bits of the register. In the case of the A0 and A1 registers, the data sign is extended and loaded into the guard bits. When the A0G or A1G register is used as the destination register in longword mode, the lowest 8 bits of the data are loaded into the register, and the A0 or A1 register is not cleared to 0, but retains its prior value.

The register data formats used with DSP instructions are shown in tables 2.4 and 2.5. With some instructions, not all registers can be accessed. For example, with the PMULS instruction, the A1 register can be specified as the source register, but the A0 register cannot. See the descriptions of the instructions for details.

The relationship between the DSP registers and the buses in data transfer is shown in figure 2.8.

Table 2.4 DSP Instruction Source Register Data Formats

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

Note: * The sign is extended and stored in the ALU guard bits.

Table 2.5 DSP Instruction Destination Register Data Formats

Registers	Instructions		Guard Bits	Register Bits	
			39 32	31 16	15 0
A0, A1	DSP operations	Fixed-point, PSHA, PMULS	(Sign extension)	40-bit result	
		Integer, PDMSB	(Sign extension)	24-bit result	Cleared to 0
		Logical, PSHL	Cleared to 0	16-bit result	Cleared to 0
	Data transfer	MOVS.W	Sign extension	16-bit result	Cleared to 0
		MOVS.L	Sign extension	32-bit data	
A0G, A1G	Data transfer	MOVS.W	Data	Not updated	
		MOVS.L	Data	Not updated	
X0, X1 Y0, Y1 M0, M1	DSP operations	Fixed-point, PSHA, PMULS	▨	32-bit result	
		Integer, logical, PDMSB, PSHL	▨	16-bit result	Cleared to 0
	Data transfer	MOVX.W, MOVY.W, MOVS.W	▨	16-bit data	Cleared to 0
		MOVS.L	▨	32-bit data	

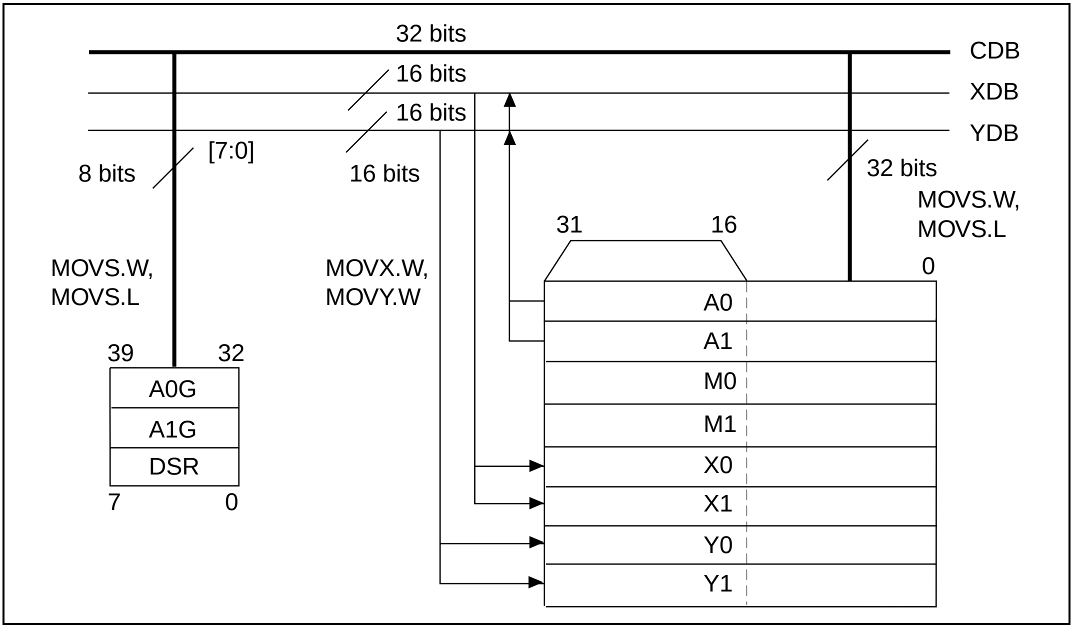


Figure 2.8 Relationship between DSP Registers and Buses in Data Transfer

2.3 Features of CPU Core Instructions

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

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

One Instruction per State: Pipelining is used, and basic instructions can be executed in one state (equivalent to 16.7 ns at 60-MHz operation).

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

Table 2.6 Word Data Sign Extension

SH7065 CPU		Description	Example of Other CPU	
MOV.W	@(disp,PC),R1	Sign-extended to 32 bits, R1 becomes H'00001234, and is then operated on by the ADD instruction.	ADD.W	#H'1234,R0
ADD	R1,R0			
. . . .				
.DATA.W	H'1234			

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

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

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

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

Table 2.7 Delayed Branch Instructions

SH7065 CPU		Description	Example of Other CPU	
BRA	TRGET	ADD is executed before branch to TRGET.	ADD.W	R1,R0
ADD	R1,R0		BRA	TRGET

Multiply/Multiply and Accumulate Operations: A 16 × 16 → 32 multiply operation is executed in 1 to 3 states, and a 16 × 16 + 64 → 64 multiply and accumulate operation in 2 to 3 states. A 32 × 32 → 64 multiply operation and a 32 × 32 + 64 → 64 multiply and accumulate operation are each executed in 2 to 4 states.

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

Table 2.8 T Bit

SH7065 CPU		Description	Example of Other CPU	
CMP/GE	R1,R0	If R0 ≥ R1, the T bit is set.	CMP.W	R1,R0
BT	TRGET0	A branch is made to TRGET0 if R0 ≥ R1, or to TRGET1 if R0 < R1.	BGE	TRGET0
BF	TRGET1		BLT	TRGET1
ADD	#–1,R0	The T bit is not set by ADD.	SUB.W	#1,R0
CMP/EQ	#0,R0	If R0 = 0, the T bit is set.	BEQ	TRGET
BT	TRGET	A branch is made if R0 = 0.		

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

Table 2.9 Immediate Data Referencing

Type	SH7065 CPU	Example of Other CPU
8-bit immediate	MOV #H'12,R0	MOV.B #H'12,R0
16-bit immediate	MOV.W @(disp,PC),R0 . . . . .DATA.W H'1234	MOV.W #H'1234,R0
32-bit immediate	MOV.L @(disp,PC),R0 . . . . .DATA.L H'12345678	MOV.L #H'12345678,R0

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

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

Table 2.10 Absolute Address Referencing

Type	SH7065 CPU	Example of Other CPU
Absolute address	MOV.L @(disp,PC),R1 MOV.B @R1,R0 . . . . .DATA.L H'12345678	MOV.B @H'12345678,R0

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

Table 2.11 Displacement Referencing

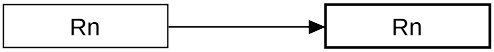
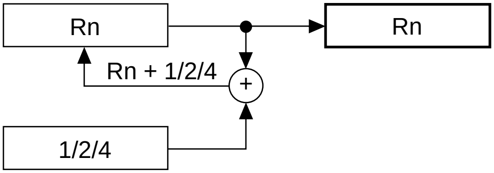
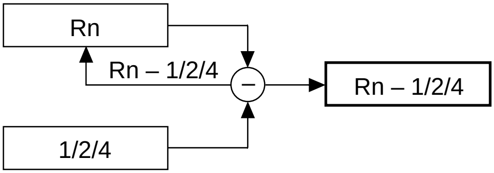
Type	SH7065 CPU	Example of Other CPU
16-bit displacement	MOV.W @(disp,PC),R0 MOV.W @(R0,R1),R2 . . . . .DATA.W H'1234	MOV.W @(H'1234,R1),R2

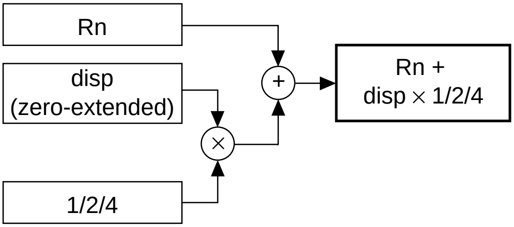
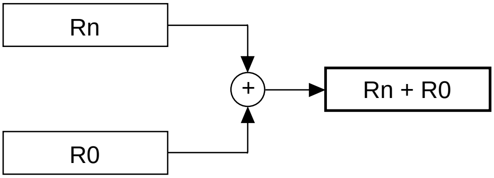
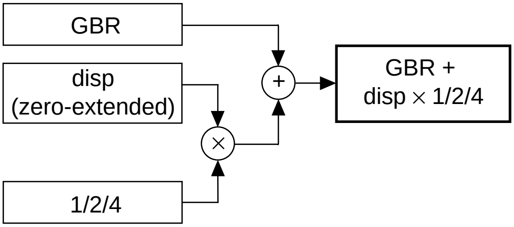
2.4 Instruction Formats

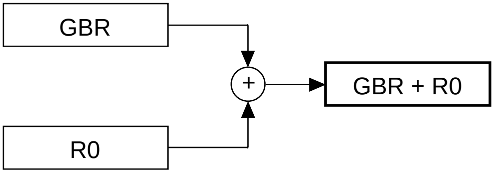
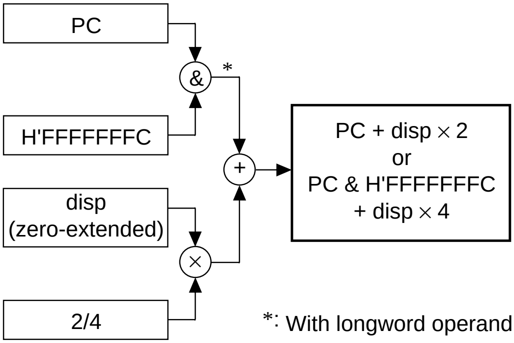
2.4.1 CPU Instruction Addressing Modes

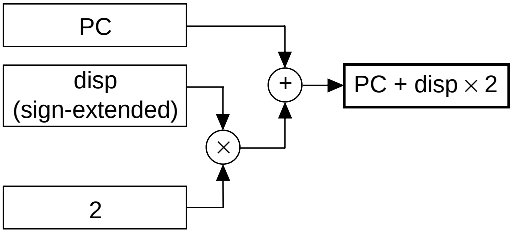
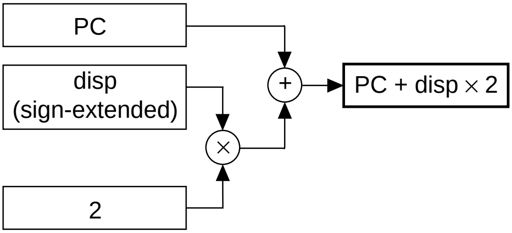
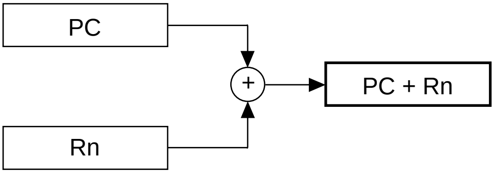
The following table shows addressing modes and effective address calculation methods for instructions executed by the CPU core.

Table 2.12 Addressing Modes and Effective Addresses for CPU Instructions

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

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

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

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

2.4.2 DSP Data Addressing

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

Table 2.13 Overview of Data Transfer Instructions

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

X/Y Data Addressing

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

1. Non-update address register addressing:

   The Ax and Ay registers are address pointers. They are not updated.

2. Addition index register addressing:

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

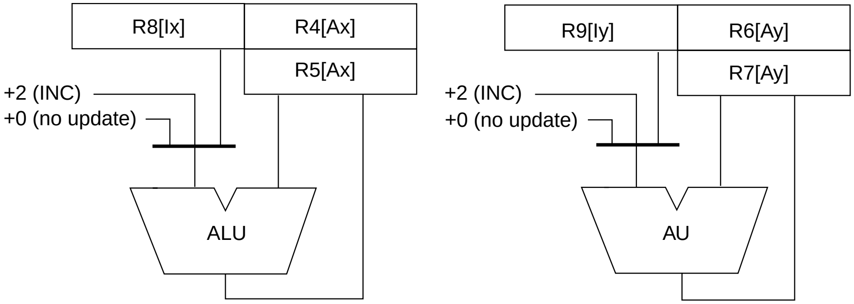
3. Increment address register addressing:

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

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

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

X/Y data transfer addressing is shown in figure 2.9. When accessing X and Y memory using the X and Y buses, the upper word of Ax (R4 or R5) and Ay (R6 or R7) is ignored. The result of @AY+ or @Ay+Iy is stored in the lower word of Ay, while the upper word retains its original value.



AU: Adder provided for DSP addressing

Note: Three address processing methods:
1. Increment
2. Index register addition (Ix/Iy)
3. No update
Post-updating is used in all cases.
The address pointer can be decremented by setting –2/–4 in the index register.

Figure 2.9 X and Y Data Transfer Addressing

Single Data Addressing

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

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

1. Non-update address register addressing:

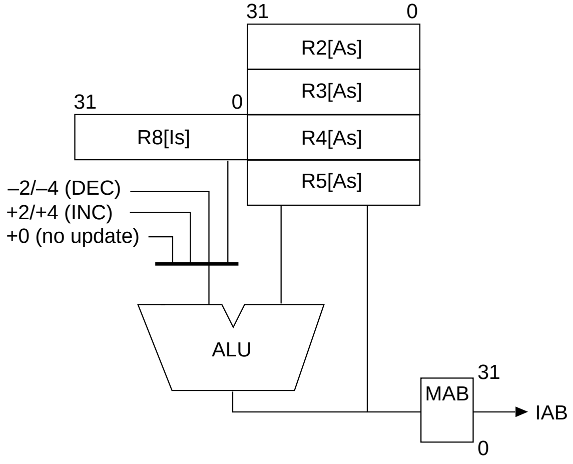
   The As register is an address pointer. It is not updated.
2. Addition index register addressing:

   The As register is an address pointer. After a data transfer, the value of the Is register is added to the As register (post-updating).
3. Increment address register addressing:

   The As register is an address pointer. After a data transfer, the As register is incremented by 2 or 4 (post-updating).
4. Decrement address register addressing:

   The As register is an address pointer. Before a data transfer, –2 or –4 is added to the As register (i.e. 2 or 4 is subtracted) (pre-updating).

The R8 register is the index register (Is) for the address pointer (As). Single data transfer addressing is shown in figure 2.10.



Note: Four address processing methods:

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

Figure 2.10 Single Data Transfer Addressing

Modulo Addressing

Like other DSPs, the SH7065 has a modulo addressing mode. Address registers are updated in the same way in this mode. When the address pointer value reaches the preset modulo end address, the address pointer value becomes the modulo start address.

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

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

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

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

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

The maximum modulo size is 64 kbytes. This is sufficient to access the X and Y memory. A block diagram of modulo addressing is shown in figure 2.11.

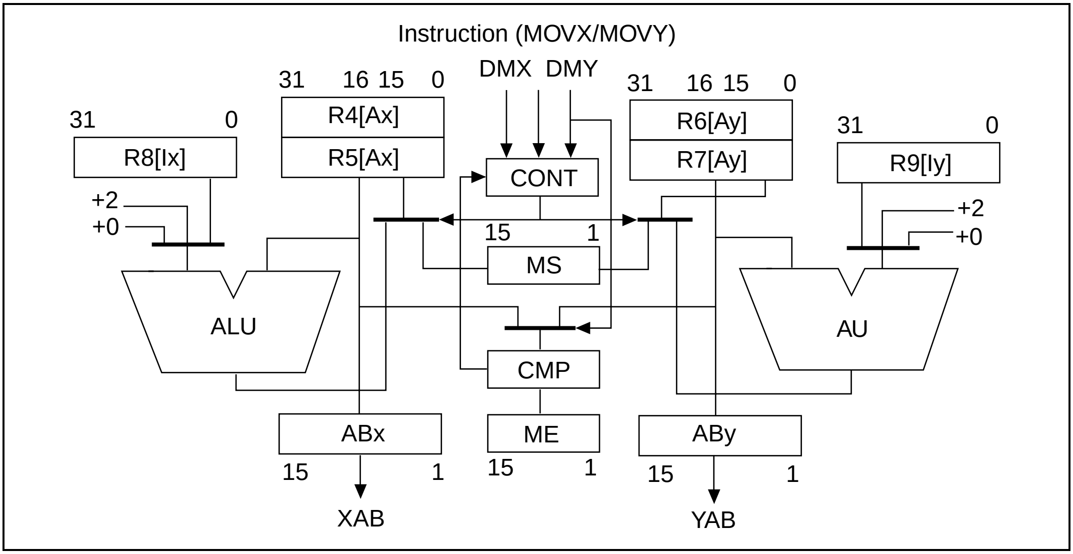


Figure 2.11 Modulo Addressing

An example of modulo addressing is given below.

```
MS = H'C008;  ME = H'C00C;  R4 = H'C008;
              DMX = 1;      DMY = 0;     (Modulo addressing setting for address
                                          register Ax (R4, R5))
```

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

```
       R4: H'C008
Inc.   R4: H'C00A
Inc.   R4: H'C00C
Inc.   R4: H'C008   (Reaches modulo end address, so becomes modulo start address)
```

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

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

DSP Addressing Operations

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

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

    /* Ax is one of R4,5 */
    if ( DMX==0 || DMX==1 && DMY == 1 )} Ax=Ax+(+2 or R8[Ix] or +0);
    /* Inc,Index,Not-Update */
    else if (! not-update) Ax=modulo( Ax, (+2 or R8[Ix]) );

    /* Ay is one of R6,7 */
    if ( DMY==0 ) Ay=Ay+(+2 or R9[Iy] or +0); /* Inc,Index,Not-Update */
    else if (! not-update) Ay=modulo( Ay, (+2 or R9[Iy]) );
}
else if ( Operation is MOVS.W or MOVS.L ) {
  if ( Addressing is Nop, Inc, Add-index-reg ) {
    MAB=As;
    /* memory access cycle uses MAB. The address to be used has not been updated */
    /* As is one of R2-5 */
    As=As+(+2 or +4 or R8[Is] or +0); /* Inc,Index,Not-Update */
  else { /* Decrement, Pre-update */
      /* As is one of R2–5 */
      As=As+(-2 or -4);
      MAB=As;
      /* memory access cycle uses MAB. The address to be used has been updated */
}

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

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

2.4.3 CPU Instruction Formats

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

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

Table 2.14 CPU Instruction Formats

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

Instruction Formats		Source Operand	Destination Operand	Sample Instruction
nm type	15 0 xxxx | nnnn | mmmm | xxxx	mmmm: register direct	nnnn: register direct	ADD Rm,Rn
		mmmm: register direct	nnnn: register indirect	MOV.L Rm,@Rn
		mmmm: post-increment register indirect (multiply and accumulate operation) nnnn: * post-increment register indirect (multiply and accumulate operation)	MACH, MACL	MAC.W @Rm+,@Rn+
		mmmm: post-increment register indirect	nnnn: register direct	MOV.L @Rm+,Rn
		mmmm: register direct	nnnn: pre-decrement register indirect	MOV.L Rm,@-Rn
		mmmm: register direct	nnnn: indexed register indirect	MOV.L Rm,@(R0,Rn)
md type	15 0 xxxx xxxx | mmmm | dddd	mmmmdddd: register indirect with displacement	R0 (register direct)	MOV.B @(disp,Rm),R0
nd4 type	15 0 xxxx xxxx | nnnn | dddd	R0 (register direct)	nnnndddd: register indirect with displacement	MOV.B R0,@(disp,Rn)
nmd type	15 0 xxxx | nnnn | mmmm | dddd	mmmm: register direct	nnnndddd: register indirect with displacement	MOV.L Rm,@(disp,Rn)
		mmmmdddd: register indirect with displacement	nnnn: register direct	MOV.L @(disp,Rm),Rn

Instruction Formats		Source Operand	Destination Operand	Sample Instruction
d type	15 0 xxxx xxxx dddd dddd	dddddddd: GBR indirect with displacement	R0 (register direct)	MOV.L @(disp,GBR),R0
		R0 (register direct)	dddddddd: GBR indirect with displacement	MOV.L R0,@(disp,GBR)
		dddddddd: PC-relative with displacement	R0 (register direct)	MOVA @(disp,PC),R0
		dddddddd: PC-relative	—	BF label
d12 type	15 0 xxxx dddd dddd dddd	dddddddddddd: PC-relative	—	BRA label (label=disp+PC)
nd8 type	15 0 xxxx nnnn dddd dddd	dddddddd: PC-relative with displacement	nnnn: register direct	MOV.L @(disp,PC),Rn
i type	15 0 xxxx xxxx iiii iiii	iiiiiiii: immediate	Indexed GBR indirect	AND.B #imm,@(R0,GBR)
		iiiiiiii: immediate	R0 (register direct)	AND #imm,R0
		iiiiiiii: immediate	—	TRAPA #imm
ni type	15 0 xxxx nnnn iiii iiii	iiiiiiii: immediate	nnnn: register direct	AD #imm,Rn

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

2.4.4 DSP Instruction Formats

The SH7065 includes new instructions for digital signal processing. The new instructions are of the following two kinds.

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

The instruction formats are shown in figure 2.12.

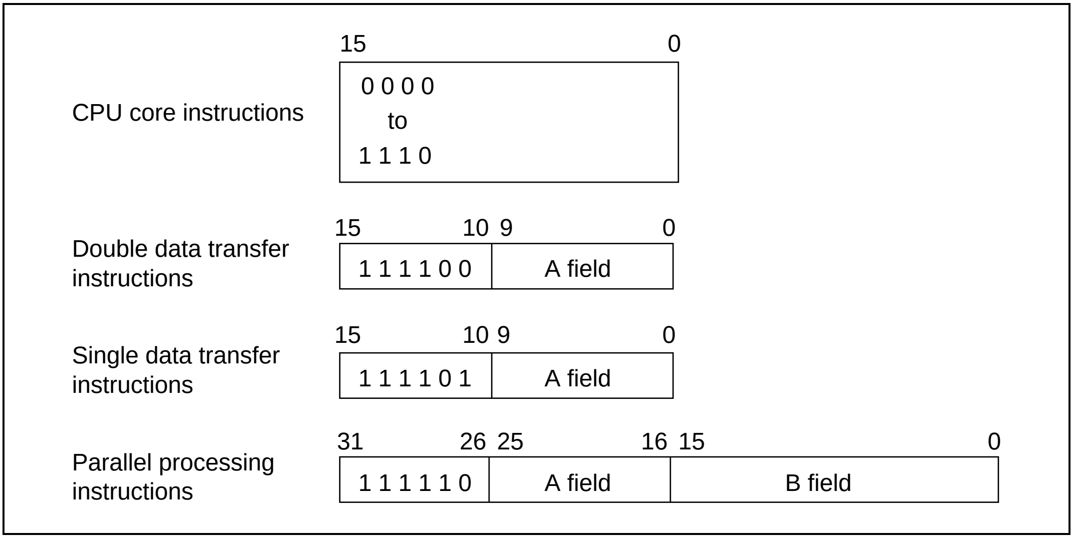


Figure 2.12 DSP Instruction Formats

Double and Single Data Transfer Instructions

The format of double data transfer instructions is shown in table 2.15, and that of single data transfer instructions in table 2.16.

Table 2.15 Double Data Transfer Instruction Formats

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

Legend:

Ax: 0 = R4, 1 = R5

Ay: 0 = R6, 1 = R7

Dx: 0 = X0, 1 = X1

Dy: 0 = Y0, 1 = Y1

Da: 0 = A0, 1 = A1

Table 2.16 Single Data Transfer Instruction Formats

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

Note: * Codes reserved for system use.

Parallel Processing Instructions

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

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

Table 2.17 A Field Parallel Data Transfer Instructions

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

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

Table 2.18 B Field ALU Operation Instructions and Multiply Instructions

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

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

Notes: 1. Codes reserved for system use.

2. [if cc]: DCT (DC bit True), DCF (DC bit False) or none (unconditional instruction)

2.5 Instruction Set

SH7065 instructions can be divided into three kinds: CPU instructions executed by the CPU core, and DSP data transfer instructions and DSP operation instructions executed by the DSP unit. CPU instructions include several for supporting DSP functions. The instruction sets for each of these three kinds of instructions are described below.

2.5.1 CPU Instruction Set

The CPU instructions are listed by type in table 2.19.

Table 2.19 CPU Instruction Types

Type	Kinds of Instruction	Op Code	Function	Number of Instructions
Data transfer instructions	5	MOV	Data transfer Immediate data transfer Peripheral module data transfer Structure data transfer	39
		MOVA	Effective address transfer	
		MOVT	T bit transfer	
		SWAP	Upper/lower swap	
		XTRCT	Extraction of middle of linked registers	
Arithmetic operation instructions	21	ADD	Binary addition	33
		ADDC	Binary addition with carry	
		ADDV	Binary addition with overflow	
		CMP/cond	Comparison	
		DIV1	Division	
		DIV0S	Signed division initialization	
		DIV0U	Unsigned division initialization	
		DMULS	Signed double-precision multiplication	
		DMULU	Unsigned double-precision multiplication	
		DT	Decrement and test	

Type	Kinds of Instruction	Op Code	Function	Number of Instructions
Arithmetic operation instructions	21	EXTS	Sign extension	33
		EXTU	Zero extension	
		MAC	Multiply and accumulate, double-precision multiply and accumulate	
		MUL	Double-precision multiplication	
		MULS	Signed multiplication	
		MULU	Unsigned multiplication	
		NEG	Sign inversion	
		NEGC	Sign inversion with borrow	
		SUB	Binary subtraction	
		SUBC	Binary subtraction with borrow	
		SUBV	Binary subtraction with underflow	
Logic operation instructions	6	AND	Logical AND	14
		NOT	Bit inversion	
		OR	Logical OR	
		TAS	Memory test and bit set	
		TST	Logical AND T bit state	
		XOR	Exclusive logical OR	
Shift instructions	10	ROTCL	1-bit left shift with T bit	14
		ROTCR	1-bit right shift with T bit	
		ROTL	1-bit left shift	
		ROTR	1-bit right shift	
		SHAL	Arithmetic 1-bit left shift	
		SHAR	Arithmetic 1-bit right shift	
		SHLL	Logical 1-bit left shift	
		SHLLn	Logical n-bit left shift	
		SHLR	Logical 1-bit right shift	
		SHLRn	Logical n-bit right shift	

Type	Kinds of Instruction	Op Code	Function	Number of Instructions
Branch instructions	9	BF	Condition branch, delayed conditional branch (branches if T = 0)	11
		BT	Condition branch, delayed conditional branch (branches if T = 1)	
		BRA	Unconditional branch	
		BRAF	Unconditional branch	
		BSR	Branch to subroutine procedure	
		BSRF	Branch to subroutine procedure	
		JMP	Unconditional branch	
		JSR	Branch to subroutine procedure	
		RTS	Return from subroutine procedure	
System control instructions	14	CLRMAC	MAC register clear	71
		CLRT	T bit clear	
		LDC	Load into control register	
		LDRE	Load into repeat end register	
		LDRS	Load into repeat start register	
		LDS	Load into system register	
		NOP	No operation	
		RTE	Return from exception handling	
		SETRC	Repeat count setting	
		SETT	T bit setting	
		SLEEP	Transition to power-down mode	
		STC	Store from control register	
		STS	Store from system register	
		TRAPA	Trap exception handling	
	Total: 65			Total: 182

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

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

Notes: 1. The table shows the minimum number of execution states. In practice, the number of instruction execution states will be increased in cases such as the following:

- When there is conflict between an instruction fetch and a data access
- When the destination register of a load instruction (memory → register) is also used by the following instruction

2. Scaling (×1, ×2, or ×4) is executed according to the instruction operand size. See the *SH-1/SH-2/SH-DSP Software Manual* for details.

Table 2.20 Data Transfer Instructions

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

Instruction		Instruction Code	Operation	Execution States	T Bit
MOV.W	Rm,@(R0,Rn)	0000nnnnmmmm0101	Rm → (R0 + Rn)	1	—
MOV.L	Rm,@(R0,Rn)	0000nnnnmmmm0110	Rm → (R0 + Rn)	1	—
MOV.B	@(R0,Rm),Rn	0000nnnnmmmm1100	(R0 + Rm) → sign extension → Rn	1	—
MOV.W	@(R0,Rm),Rn	0000nnnnmmmm1101	(R0 + Rm) → sign extension → Rn	1	—
MOV.L	@(R0,Rm),Rn	0000nnnnmmmm1110	(R0 + Rm) → Rn	1	—
MOV.B	R0,@(disp,GBR)	11000000dddddddd	R0 → (disp + GBR)	1	—
MOV.W	R0,@(disp,GBR)	11000001dddddddd	R0 → (disp × 2 + GBR)	1	—
MOV.L	R0,@(disp,GBR)	11000010dddddddd	R0 → (disp × 4 + GBR)	1	—
MOV.B	@(disp,GBR),R0	11000100dddddddd	(disp + GBR) → sign extension → R0	1	—
MOV.W	@(disp,GBR),R0	11000101dddddddd	(disp × 2 + GBR) → sign extension → R0	1	—
MOV.L	@(disp,GBR),R0	11000110dddddddd	(disp × 4 + GBR) → R0	1	—
MOVA	@(disp,PC),R0	11000111dddddddd	disp × 4 + PC → R0	1	—
MOVT	Rn	0000nnnn00101001	T → Rn	1	—
SWAP.B	Rm,Rn	0110nnnnmmmm1000	Rm → swap lower 2 bytes → Rn	1	—
SWAP.W	Rm,Rn	0110nnnnmmmm1001	Rm → swap upper/lower words → Rn	1	—
XTRCT	Rm,Rn	0010nnnnmmmm1101	Middle 32 bits of Rm and Rn → Rn	1	—

Table 2.21 Arithmetic Operation Instructions

Instruction		Instruction Code	Operation	Execution States	T Bit
ADD	Rm,Rn	`0011nnnnmmmm1100`	Rn + Rm → Rn	1	—
ADD	#imm,Rn	`0111nnnniiiiiiii`	Rn + imm → Rn	1	—
ADDC	Rm,Rn	`0011nnnnmmmm1110`	Rn + Rm + T → Rn, carry → T	1	Carry
ADDV	Rm,Rn	`0011nnnnmmmm1111`	Rn + Rm→ Rn, overflow → T	1	Overflow
CMP/EQ	#imm,R0	`10001000iiiiiiii`	When R0 = imm, 1 → T	1	Comparison result
CMP/EQ	Rm,Rn	`0011nnnnmmmm0000`	When Rn = Rm, 1 → T	1	Comparison result
CMP/HS	Rm,Rn	`0011nnnnmmmm0010`	When Rn ≥ Rm (unsigned), 1 → T	1	Comparison result
CMP/GE	Rm,Rn	`0011nnnnmmmm0011`	When Rn ≥ Rm (signed), 1 → T	1	Comparison result
CMP/HI	Rm,Rn	`0011nnnnmmmm0110`	When Rn > Rm (unsigned), 1 → T	1	Comparison result
CMP/GT	Rm,Rn	`0011nnnnmmmm0111`	When Rn > Rm (signed), 1 → T	1	Comparison result
CMP/PL	Rn	`0100nnnn00010101`	When Rn > 0, 1 → T	1	Comparison result
CMP/PZ	Rn	`0100nnnn00010001`	When Rn ≥ 0, 1 → T	1	Comparison result
CMP/STR	Rm,Rn	`0010nnnnmmmm1100`	When any bytes are equal, 1→ T	1	Comparison result
DIV1	Rm,Rn	`0011nnnnmmmm0100`	1-step division (Rn ÷ Rm)	1	Calculation result
DIV0S	Rm,Rn	`0010nnnnmmmm0111`	MSB of Rn → Q, MSB of Rm → M, M^Q → T	1	Calculation result
DIV0U		`0000000000011001`	0 → M/Q/T	1	0
DMULS.L	Rm,Rn	`0011nnnnmmmm1101`	Signed, Rn × Rm → MACH, MACL 32 × 32 → 64 bits	2–4*	—

Instruction		Instruction Code	Operation	Execution States	T Bit
DMULU.L	Rm,Rn	`0011nnnnmmmm0101`	Unsigned, Rn × Rm → MACH, MACL 32 × 32 → 64 bits	2–4*	—
DT	Rn	`0100nnnn00010000`	Rn – 1 → Rn; when Rn = 0, 1 → T When Rn ≠ 0, 0 → T	1	Comparison result
EXTS.B	Rm,Rn	`0110nnnnmmmm1110`	Rm sign-extended from byte → Rn	1	—
EXTS.W	Rm,Rn	`0110nnnnmmmm1111`	Rm sign-extended from word → Rn	1	—
EXTU.B	Rm,Rn	`0110nnnnmmmm1100`	Rm zero-extended from byte → Rn	1	—
EXTU.W	Rm,Rn	`0110nnnnmmmm1101`	Rm zero-extended from word → Rn	1	—
MAC.L	@Rm+,@Rn+	`0000nnnnmmmm1111`	Signed, (Rn) × (Rm) + MAC → MAC 32 × 32 + 64 → 64 bits	3/(2–4)*	—
MAC.W	@Rm+,@Rn+	`0100nnnnmmmm1111`	Signed, (Rn) × (Rm) + MAC → MAC 16 × 16 + 64 → 64 bits	3/(2)*	—
MUL.L	Rm,Rn	`0000nnnnmmmm0111`	Rn × Rm → MACL 32 × 32 → 32 bits	2–4*	—
MULS.W	Rm,Rn	`0010nnnnmmmm1111`	Signed, Rn × Rm → MAC 16 × 16 → 32 bits	1–3*	—
MULU.W	Rm,Rn	`0010nnnnmmmm1110`	Unsigned, Rn × Rm → MAC 16 × 16 → 32 bits	1–3*	—
NEG	Rm,Rn	`0110nnnnmmmm1011`	0 – Rm → Rn	1	—
NEGC	Rm,Rn	`0110nnnnmmmm1010`	0 – Rm – T → Rn, borrow → T	1	Borrow
SUB	Rm,Rn	`0011nnnnmmmm1000`	Rn – Rm → Rn	1	—

Instruction		Instruction Code	Operation	Execution States	T Bit
SUBC	Rm,Rn	0011nnnnmmmm1010	Rn – Rm – T → Rn, borrow → T	1	Borrow
SUBV	Rm,Rn	0011nnnnmmmm1011	Rn – Rm → Rn, underflow → T	1	Underflow

Note: * The normal number of execution states is shown. The number in parentheses is the number of execution cycles in the case of contention with preceding or following instructions.

Table 2.22 Logic Operation Instructions

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

Table 2.23 Shift Instructions

Instruction		Instruction Code	Operation	Execution States	T Bit
ROTL	Rn	0100nnnn00000100	T ← Rn ← MSB	1	MSB
ROTR	Rn	0100nnnn00000101	LSB→ Rn→ T	1	LSB
ROTCL	Rn	0100nnnn00100100	T ← Rn ← T	1	MSB
ROTCR	Rn	0100nnnn00100101	T→ Rn→ T	1	LSB
SHAL	Rn	0100nnnn00100000	T ← Rn ← 0	1	MSB
SHAR	Rn	0100nnnn00100001	MSB→ Rn→ T	1	LSB
SHLL	Rn	0100nnnn00000000	T ← Rn ← 0	1	MSB
SHLR	Rn	0100nnnn00000001	0→ Rn→ T	1	LSB
SHLL2	Rn	0100nnnn00001000	Rn << 2 → Rn	1	—
SHLR2	Rn	0100nnnn00001001	Rn >> 2 → Rn	1	—
SHLL8	Rn	0100nnnn00011000	Rn << 8 → Rn	1	—
SHLR8	Rn	0100nnnn00011001	Rn >> 8 → Rn	1	—
SHLL16	Rn	0100nnnn00101000	Rn << 16 → Rn	1	—
SHLR16	Rn	0100nnnn00101001	Rn >> 16 → Rn	1	—

Table 2.24 Branch Instructions

Instruction		Instruction Code	Operation	Execution States	T Bit
BF	label	10001011dddddddd	When T = 0, disp × 2 + PC → PC; when T = 1, nop	3/1*	—

BF/S	label	10001111dddddddd	Delayed branch;
			when T = 0,
			disp × 2 + PC → PC;
			when T = 1, nop
				2/1*	—

BT	label	10001001dddddddd	When T = 1, disp × 2 + PC → PC; when T = 0, nop	3/1*	—

BT/S	label	10001101dddddddd	Delayed branch;
			when T = 1,
			disp × 2 + PC → PC;
			when T = 0, nop
				2/1*	—

BRA	label	1010dddddddddddd	Delayed branch, disp × 2 + PC → PC	2	—
BRAF	Rm	0000mmmm00100011	Delayed branch, Rm + PC → PC	2	—
BSR	label	1011dddddddddddd	Delayed branch, PC → PR, disp × 2 + PC→ PC	2	—
BSRF	Rm	0000mmmm00000011	Delayed branch, PC→ PR, Rm + PC→ PC	2	—
JMP	@Rm	0100mmmm00101011	Delayed branch, Rm→ PC	2	—
JSR	@Rm	0100mmmm00001011	Delayed branch, PC→ PR, Rm→ PC	2	—
RTS		0000000000001011	Delayed branch, PR→ PC	2	—

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

Table 2.25 System Control Instructions

Instruction		Instruction Code	Operation	Execution States	T Bit
CLRMAC		0000000000101000	0 → MACH, MACL	1	—
CLRT		0000000000001000	0 → T	1	0
LDC	Rm,SR	0100mmmm00001110	Rm → SR	1	LSB
LDC	Rm,GBR	0100mmmm00011110	Rm → GBR	1	—
LDC	Rm,VBR	0100mmmm00101110	Rm → VBR	1	—
LDC	Rm,MOD	0100mmmm01011110	Rm → MOD	1	—
LDC	Rm,RE	0100mmmm01111110	Rm → RE	1	—
LDC	Rm,RS	0100mmmm01101110	Rm → RS	1	—
LDC.L	@Rm+,SR	0100mmmm00000111	(Rm) → SR, Rm + 4 → Rm	3	LSB
LDC.L	@Rm+,GBR	0100mmmm00010111	(Rm) → GBR, Rm + 4 → Rm	3	—
LDC.L	@Rm+,VBR	0100mmmm00100111	(Rm) → VBR, Rm + 4 → Rm	3	—
LDC.L	@Rm+,MOD	0100mmmm01010111	(Rm) → MOD, Rm + 4 → Rm	3	—
LDC.L	@Rm+,RE	0100mmmm01110111	(Rm) → RE, Rm + 4 → Rm	3	—
LDC.L	@Rm+,RS	0100mmmm01100111	(Rm) → RS, Rm + 4 → Rm	3	—
LDRE	@(disp,PC)	10001110dddddddd	disp × 2 + PC → RE	1	—
LDRS	@(disp,PC)	10001100dddddddd	disp × 2 + PC → RS	1	—
LDS	Rm,MACH	0100mmmm00001010	Rm → MACH	1	—
LDS	Rm,MACL	0100mmmm00011010	Rm → MACL	1	—
LDS	Rm,PR	0100mmmm00101010	Rm → PR	1	—
LDS	Rm,DSR	0100mmmm01101010	Rm → DSR	1	—
LDS	Rm,A0	0100mmmm01111010	Rm → A0	1	—
LDS	Rm,X0	0100mmmm10001010	Rm → X0	1	—
LDS	Rm,X1	0100mmmm10011010	Rm → X1	1	—
LDS	Rm,Y0	0100mmmm10101010	Rm → Y0	1	—
LDS	Rm,Y1	0100mmmm10111010	Rm → Y1	1	—
LDS.L	@Rm+,MACH	0100mmmm00000110	(Rm) → MACH, Rm + 4 → Rm	1	—
LDS.L	@Rm+,MACL	0100mmmm00010110	(Rm) → MACL, Rm + 4 → Rm	1	—
LDS.L	@Rm+,PR	0100mmmm00100110	(Rm) → PR, Rm + 4 → Rm	1	—
LDS.L	@Rm+,DSR	0100mmmm01100110	(Rm) → DSR, Rm + 4 → Rm	1	—

Instruction		Instruction Code	Operation	Execution States	T Bit
LDS.L	@Rm+,A0	`0100mmmm01110110`	(Rm) → A0, Rm + 4 → Rm	1	—
LDS.L	@Rm+,X0	`0100mmmm10000110`	(Rm) → X0, Rm + 4 → Rm	1	—
LDS.L	@Rm+,X1	`0100mmmm10010110`	(Rm) → X1, Rm + 4 → Rm	1	—
LDS.L	@Rm+,Y0	`0100mmmm10100110`	(Rm) → Y0, Rm + 4 → Rm	1	—
LDS.L	@Rm+,Y1	`0100mmmm10110110`	(Rm) → Y1, Rm + 4 → Rm	1	—
NOP		`0000000000001001`	No operation	1	—
RTE		`0000000000101011`	Delayed branch, stack area → PC/SR	4	LSB
SETRC	Rm	`0100mmmm00010100`	RE – RS operation result (repeat status) → RF1, RF0 Rm [11:0] → RC (SR [27:16])	1	—
SETRC	#imm	`10000010iiiiiiii`	RE – RS operation result (repeat status) → RF1, RF0 imm → RC (SR [23:16]), zeros → SR [27:24]	1	1
SETT		`0000000000011000`	1 → T	1	1
SLEEP		`0000000000011011`	Sleep	3*	—
STC	SR,Rn	`0000nnnn00000010`	SR → Rn	1	—
STC	GBR,Rn	`0000nnnn00010010`	GBR → Rn	1	—
STC	VBR,Rn	`0000nnnn00100010`	VBR → Rn	1	—
STC	MOD,Rn	`0000nnnn01010010`	MOD → Rn	1	—
STC	RE,Rn	`0000nnnn01110010`	RE → Rn	1	—
STC	RS,Rn	`0000nnnn01100010`	RS → Rn	1	—
STC.L	SR,@-Rn	`0100nnnn00000011`	Rn – 4 → Rn, SR → (Rn)	2	—
STC.L	GBR,@-Rn	`0100nnnn00010011`	Rn – 4 → Rn, GBR → (Rn)	2	—
STC.L	VBR,@-Rn	`0100nnnn00100011`	Rn – 4 → Rn, VBR → (Rn)	2	—
STC.L	MOD,@-Rn	`0100nnnn01010011`	Rn – 4 → Rn, MOD → (Rn)	2	—
STC.L	RE,@-Rn	`0100nnnn01110011`	Rn – 4 → Rn, RE → (Rn)	2	—
STC.L	RS,@-Rn	`0100nnnn01100011`	Rn – 4 → Rn, RS → (Rn)	2	—
STS	MACH,Rn	`0000nnnn00001010`	MACH → Rn	1	—
STS	MACL,Rn	`0000nnnn00011010`	MACL → Rn	1	—
STS	PR,Rn	`0000nnnn00101010`	PR → Rn	1	—

Instruction		Instruction Code	Operation	Execution States	T Bit
STS	DSR,Rn	`0000nnnn01101010`	DSR → Rn	1	—
STS	A0,Rn	`0000nnnn01111010`	A0 → Rn	1	—
STS	X0,Rn	`0000nnnn10001010`	X0 → Rn	1	—
STS	X1,Rn	`0000nnnn10011010`	X1 → Rn	1	—
STS	Y0,Rn	`0000nnnn10101010`	Y0 → Rn	1	—
STS	Y1,Rn	`0000nnnn10111010`	Y1 → Rn	1	—
STS.L	MACH,@-Rn	`0100nnnn00000010`	Rn – 4 → Rn, MACH → (Rn)	1	—
STS.L	MACL,@-Rn	`0100nnnn00010010`	Rn – 4 → Rn, MACL → (Rn)	1	—
STS.L	PR,@-Rn	`0100nnnn00100010`	Rn – 4 → Rn, PR → (Rn)	1	—
STS.L	DSR,@-Rn	`0100nnnn01100010`	Rn – 4 → Rn, DSR → (Rn)	1	—
STS.L	A0,@-Rn	`0100nnnn01110010`	Rn – 4 → Rn, A0 → (Rn)	1	—
STS.L	X0,@-Rn	`0100nnnn10000010`	Rn – 4 → Rn, X0 → (Rn)	1	—
STS.L	X1,@-Rn	`0100nnnn10010010`	Rn – 4 → Rn, X1 → (Rn)	1	—
STS.L	Y0,@-Rn	`0100nnnn10100010`	Rn – 4 → Rn, Y0 → (Rn)	1	—
STS.L	Y1,@-Rn	`0100nnnn10110010`	Rn – 4 → Rn, Y1 → (Rn)	1	—
TRAPA	#imm	`11000011iiiiiiii`	PC/SR → stack area, (imm × 4 + VBR) → PC	8	—

Note: * Number of states until transition to sleep state.

Caution:

- The table shows the minimum number of execution states. In practice, the number of instruction execution states will be increased in cases such as the following:
  - When there is conflict between an instruction fetch and a data access
  - When the destination register of a load instruction (memory → register) is also used by the following instruction
  - When the branch destination address of a branch instruction is address 4n + 2
  - Depending on the number of cycles of the instruction fetch destination or data access destination (see 8.4, Number of Access Cycles (SH7065A) in section 8, Bus State Controller (BSC), for details).

CPU Instructions Supporting DSP Functions

A number of system control instructions have been added to the CPU core instructions to support DSP functions. The RS, RE, and MOD registers have been added, supporting repeat control and modulo addressing, and a repeat counter (RC) has been added to the status register (SR). LDC and STC instructions have been added in order to access these new additions, while LDS and STS instructions have been added to access DSP registers DSR, A0, X0, X1, Y0, and Y1.

Another addition is the SETRC instruction which sets a value in the repeat counter (RC: bits 27 to 16) and the repeat flags (RF1, RF0: bits 3 and 2) in the SR register. When an immediate operand is used in the SETRC instruction, 8-bit immediate data is stored in bits 23 to 16 of SR, and bits 27 to 24 are cleared to 0. When the operand is a register, the 12 bits from bit 11 to bit 0 of the register are stored in bits 27 to 16 of SR. According to the set values of RS and RE, 1-instruction repeat (00), 2-instruction repeat (01), 3-instruction repeat (11), or 4-plus-instruction repeat (10) is set.

In addition to the LDC instruction, the LDRS and LDRE instructions have been added as instructions that set the repeat start address and repeat end address in the RS and RE registers.

The added instructions are shown in table 2.26.

Table 2.26 Added CPU Instructions

Instruction		Instruction Code	Operation	Execution States	T Bit
LDC	Rm,MOD	`0100mmmm01011110`	Rm → MOD	1	—
LDC	Rm,RE	`0100mmmm01111110`	Rm → RE	1	—
LDC	Rm,RS	`0100mmmm01101110`	Rm → RS	1	—
LDC.L	@Rm+,MOD	`0100mmmm01010111`	(Rm) → MOD, Rm + 4 → Rm	3	—
LDC.L	@Rm+,RE	`0100mmmm01110111`	(Rm) → RE, Rm + 4 → Rm	3	—
LDC.L	@Rm+,RS	`0100mmmm01100111`	(Rm) → RS, Rm + 4 → Rm	3	—
STC	MOD,Rn	`0000nnnn01010010`	MOD → Rn	1	—
STC	RE,Rn	`0000nnnn01110010`	RE → Rn	1	—
STC	RS,Rn	`0000nnnn01100010`	RS → Rn	1	—
STC.L	MOD,@-Rn	`0100nnnn01010011`	Rn – 4 → Rn, MOD → (Rn)	2	—
STC.L	RE,@-Rn	`0100nnnn01110011`	Rn – 4 → Rn, RE → (Rn)	2	—
STC.L	RS,@-Rn	`0100nnnn01100011`	Rn – 4 → Rn, RS → (Rn)	2	—
LDS	Rm,DSR	`0100mmmm01101010`	Rm → DSR	1	—

Instruction		Instruction Code	Operation	Execution States	T Bit
LDS.L	@Rm+,DSR	0100mmmm01100110	(Rm) → DSR, Rm + 4 → Rm	1	—
LDS	Rm,A0	0100mmmm01111010	Rm → A0	1	—
LDS.L	@Rm+,A0	0100mmmm01110110	(Rm) → A0, Rm + 4 → Rm	1	—
LDS	Rm,X0	0100mmmm10001010	Rm → X0	1	—
LDS.L	@Rm+,X0	0100mmmm10000110	(Rm) → X0, Rm + 4→ Rm	1	—
LDS	Rm,X1	0100mmmm10011010	Rm → X1	1	—
LDS.L	@Rm+,X1	0100mmmm10010110	(Rm) → X1, Rm + 4→ Rm	1	—
LDS	Rm,Y0	0100mmmm10101010	Rm → Y0	1	—
LDS.L	@Rm+,Y0	0100mmmm10100110	(Rm) → Y0, Rm + 4 → Rm	1	—
LDS	Rm,Y1	0100mmmm10111010	Rm → Y1	1	—
LDS.L	@Rm+,Y1	0100mmmm10110110	(Rm) → Y1, Rm + 4 → Rm	1	—
STS	DSR,Rn	0000nnnn01101010	DSR → Rn	1	—
STS.L	DSR,@-Rn	0100nnnn01100010	Rn – 4 → Rn, DSR → (Rn)	1	—
STS	A0,Rn	0000nnnn01111010	A0 → Rn	1	—
STS.L	A0,@-Rn	0100nnnn01110010	Rn – 4 → Rn, A0 → (Rn)	1	—
STS	X0,Rn	0000nnnn10001010	X0 → Rn	1	—
STS.L	X0,@-Rn	0100nnnn10000010	Rn – 4 → Rn, X0 → (Rn)	1	—
STS	X1,Rn	0000nnnn10011010	X1 → Rn	1	—
STS.L	X1,@-Rn	0100nnnn10010010	Rn – 4 → Rn, X1 → (Rn)	1	—
STS	Y0,Rn	0000nnnn10101010	Y0 → Rn	1	—
STS.L	Y0,@-Rn	0100nnnn10100010	Rn – 4 → Rn, Y0 → (Rn)	1	—
STS	Y1,Rn	0000nnnn10111010	Y1→ Rn	1	—
STS.L	Y1,@-Rn	0100nnnn10110010	Rn – 4 → Rn, Y1→ (Rn)	1	—
SETRC	Rm	0100mmmm00010100	Rm [11:0] → RC (SR [27:16])	1	—
SETRC	#imm	10000010iiiiiiii	imm → RC (SR [23:16]), 0 → SR [27:24]	1	—
LDRS	@(disp,PC)	10001100dddddddd	disp × 2 + PC → RS	1	—
LDRE	@(disp,PC)	10001110dddddddd	disp × 2 + PC → RE	1	—

2.5.2 DSP Data Transfer Instruction Set

The DSP data transfer instructions are listed by type in table 2.27.

Table 2.27 DSP Data Transfer Instruction Types

Type	Kinds of Instruction	Op Code	Function	Number of Instructions
Double data transfer instructions	4	NOPX	X memory no operation	14
		MOVX	X memory data transfer	
		NOPY	Y memory no operation	
		MOVY	Y memory data transfer	
Single data transfer instruction	1	MOVS	Single data transfer	16
	Total: 5			Total: 30

Data transfer instructions are divided into two groups: double data transfer and single data transfer. Double data transfer can be executed by DSP parallel processing instructions in combination with DSP operation instructions. Parallel processing instructions are 32 bits long, and incorporate a double data transfer instruction in the A field. Double data transfer instructions that are not parallel processing instructions, and single data transfer instructions, are 16 bits long.

In double data transfer, X memory and Y memory can be simultaneously accessed in parallel. Instructions are specified one by one from the X and Y memory data accesses, respectively. The Ax pointer is used to access the X memory, and the Ay pointer to access the Y memory. Double data transfer can only be used to access X and Y memory.

In single data transfer, access is possible from any area. In single data transfer, the Ax pointer and two other pointers are used as the As pointer.

Table 2.28 Double Data Transfer Instructions (X Memory Data)

Instruction	Operation	Instruction Code	Execution States	DC Bit
NOPX	No operation	`1111000*0*0*00**`	1	—
MOVX.W @Ax,Dx	(Ax) → MSW of Dx, 0 → LSW of Dx	`111100A*D*0*01**`	1	—
MOVX.W @Ax+,Dx	(Ax) → MSW of Dx, 0 → LSW of Dx, Ax + 2→ Ax	`111100A*D*0*10**`	1	—
MOVX.W @Ax+Ix,Dx	(Ax) → MSW of Dx, 0 → LSW of Dx, Ax + Ix→ Ax	`111100A*D*0*11**`	1	—
MOVX.W Da,@Ax	MSW of Da → (Ax)	`111100A*D*1*01**`	1	—
MOVX.W Da,@Ax+	MSW of Da → (Ax), Ax + 2 → Ax	`111100A*D*1*10**`	1	—
MOVX.W Da,@Ax+Ix	MSW of Da → (Ax), Ax + Ix → Ax	`111100A*D*1*11**`	1	—

Table 2.29 Double Data Transfer Instructions (Y Memory Data)

Instruction	Operation	Instruction Code	Execution States	DC Bit
NOPY	No operation	`111100*0*0*0**00`	1	—
MOVY.W @Ay,Dy	(Ay) → MSW of Dy, 0 → LSW of Dy	`111100*A*D*0**01`	1	—
MOVY.W @Ay+,Dy	(Ay) → MSW of Dy, 0 → LSW of Dy, Ay + 2 → Ay	`111100*A*D*0**10`	1	—
MOVY.W @Ay+Iy,Dy	(Ay) → MSW of Dy, 0 → LSW of Dy, Ay + Iy→ Ay	`111100*A*D*0**11`	1	—
MOVY.W Da,@Ay	MSW of Da → (Ay)	`111100*A*D*1**01`	1	—
MOVY.W Da,@Ay+	MSW of Da → (Ay), Ay + 2 → Ay	`111100*A*D*1**10`	1	—
MOVY.W Da,@Ay+Iy	MSW of Da → (Ay), Ay + Iy → Ay	`111100*A*D*1**11`	1	—

Table 2.30 Single Data Transfer Instructions

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

Note: * When guard bit register A0G or A1G is specified as source operand Ds, the data is sign-extended before being transferred.

The correspondence between DSP data transfer operands and registers is shown in table 2.31. CPU core registers are used as a pointer address that indicates a memory address.

Table 2.31 Correspondence between DSP Data Transfer Operands and Registers

Operand	SH (CPU Core) Register									
	R0	R1	R2 (As2)	R3 (As3)	R4 (Ax0, As0)	R5 (Ax1, As1)	R6 (Ay0)	R7 (Ay1)	R8 (Ix,Is)	R9 (Iy)
Ax					Yes	Yes				
Ix, (Is)									Yes	
Dx										
Ay							Yes	Yes		
Iy										Yes
Dy										
Da										
As			Yes	Yes	Yes	Yes				
Ds										

Operand	DSP Register									
	X0	X1	Y0	Y1	M0	M1	A0	A1	A0G	A1G
Ax										
Ix, (Is)										
Dx	Yes	Yes								
Ay										
Iy										
Dy			Yes	Yes						
Da							Yes	Yes		
As										
Ds	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes

Legend:

Yes: Settable register

2.5.3 DSP Operation Instruction Set

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

B field data operation instructions are divided into three: double data operation instructions, conditional single data operation instructions, and unconditional single data operation instructions. The formats of the DSP operation instructions are shown in table 2.32. The respective operands are selected independently from the DSP registers. The correspondence between DSP operation instruction operands and registers is shown in table 2.33.

Table 2.32 DSP Operation Instruction Formats

Type		Instruction Formats		Instructions
Double data operation instructions (6 operands)			ALUop. Sx, Sy, Du	PADD PMULS,
			MLTop. Se, Sf, Dg	PSUB PMULS
Conditional single data operation instructions	3 operands		ALUop. Sx, Sy, Dz	PADD, PAND, POR, PSHA,
		DCT	ALUop. Sx, Sy, Dz	PSHL, PSUB, PXOR
		DCF	ALUop. Sx, Sy, Dz	
	2 operands		ALUop. Sx, Dz	PCOPY, PDEC, PDMSB, PINC,
		DCT	ALUop. Sx, Dz	PLDS, PSTS, PNEG
		DCF	ALUop. Sx, Dz	
			ALUop. Sy, Dz	
		DCT	ALUop. Sy, Dz	
		DCF	ALUop. Sy, Dz	
	1 operand		ALUop. Dz	PCLR
		DCT	ALUop. Dz	
		DCF	ALUop. Dz	
Unconditional single data operation instructions	3 operands		ALUop. Sx, Sy, Du	PADDC, PSUBC, PMULS
			MLTop. Se, Sf, Dg	
	2 operands		ALUop. Sx, Dz	PCMP, PABS, PRND
			ALUop. Sy, Dz	
	1 operand		ALUop. Dz	PSHA #imm, PSHL #imm

Table 2.33 Correspondence between DSP Instruction Operands and Registers

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

Legend:
Yes: Settable register

When writing parallel instructions, the B field instruction is written first, followed by the A field instruction. A sample parallel processing program is shown in figure 2.13.

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

Figure 2.13 Sample Parallel Processing Program

Square brackets mean that the contents can be omitted. The no operation instructions NOPX and NOPY can be omitted. A semicolon is the instruction line delimiter, but this can also be omitted. If the semicolon delimiter is used, the area to the right of the semicolon can be used as a comment field.

The DSR register status codes (DC, N, Z, V, and GT) are always updated by unconditional ALU operation instructions and shift operation instructions. Conditional instructions do not update the status codes even if the condition is satisfied. Multiply instructions, too, do not update the status codes. The definition of the DC bit is determined by the specification of the CS bit in the DSR register.

The DSP operation instructions are listed by type in table 2.34.

Table 2.34 DSP Operation Instruction Types

Type		Kinds of Instruction	Op Code	Function	Number of Instructions
ALU arithmetic operation instructions	ALU fixed-point operation instructions	11	PABS	Absolute value operation	28
			PADD	Addition	
			PADD PMULS	Addition and signed multiplication	
			PADDC	Addition with carry	
			PCLR	Clear	
			PCMP	Comparison	
			PCOPY	Copy	
			PNEG	Signed inversion	
			PSUB	Subtraction	
			PSUB PMULS	Subtraction and signed multiplication	
			PSUBC	Subtraction with borrow	
	ALU integer operation instructions	2	PDEC	Decrement	12
			PINC	Increment	
	MSB detection instructions	1	PDMSB	MSB detection	6
	Rounding operation instructions	1	PRND	Rounding operation	2
ALU logic operation instructions		3	PAND	Logical AND operation	9
			POR	Logical OR operation	
			PXOR	Exclusive logical OR operation	
Fixed-point multiply instruction		1	PMULS	Signed multiplication	1
Shift	Arithmetic shift operation instructions	1	PSHA	Arithmetic shift	4
	Logical shift operation instructions	1	PSHL	Logical shift	4
System control instructions		2	PLDS	System register load	12
			PSTS	Store from system register	
		Total: 23			Total: 78

ALU Arithmetic Operation Instructions

Table 2.35 ALU Fixed-Point Operation Instructions

Instruction	Operation	Instruction Code	Execution States	DC Bit
PABS Sx,Dz	If Sx ≥ 0, Sx → Dz If Sx < 0, 0 – Sx → Dz	111110********** 10001000xx00zzzz	1	Updated
PABS Sy,Dz	If Sy ≥ 0, Sy → Dz If Sy < 0, 0 – Sy → Dz	111110********** 1010100000yyzzzz	1	Updated
PADD Sx,Sy,Dz	Sx + Sy → Dz	111110********** 10110001xxyyzzzz	1	Updated
DCT PADD Sx,Sy,Dz	If DC = 1, Sx + Sy → Dz If DC = 0, nop	111110********** 10110010xxyyzzzz	1	—
DCF PADD Sx,Sy,Dz	If DC = 0, Sx + Sy → Dz If DC = 1, nop	111110********** 10110011xxyyzzzz	1	—
PADD Sx,Sy,Du PMULS Se,Sf,Dg	Sx + Sy → Du Se upper word × Sf upper word → Dg	111110********** 0111eeffxxyygguu	1	Updated
PADDC Sx,Sy,Dz	Sx + Sy + DC → Dz	111110********** 10110000xxyyzzzz	1	Updated
PCLR Dz	H'00000000 → Dz	111110********** 100011010000zzzz	1	Updated
DCT PCLR Dz	If DC = 1, H'00000000 → Dz If DC = 0, nop	111110********** 100011100000zzzz	1	—
DCF PCLR Dz	If DC = 0, H'00000000 → Dz If DC = 1, nop	111110********** 100011110000zzzz	1	—
PCMP Sx,Sy	Sx – Sy	111110********** 10000100xxyy0000	1	Updated
PCOPY Sx,Dz	Sx → Dz	111110********** 11011001xx00zzzz	1	Updated
PCOPY Sy,Dz	Sy → Dz	111110********** 1111100100yyzzzz	1	Updated
DCT PCOPY Sx,Dz	If DC = 1, Sx → Dz If DC = 0, nop	111110********** 11011010xx00zzzz	1	—

Instruction	Operation	Instruction Code	Execution States	DC Bit
DCT PCOPY Sy,Dz	If DC = 1, Sy → Dz	111110**********	1	—
	If DC = 0, nop	1111101000yyzzzz		
DCF PCOPY Sx,Dz	If DC = 0, Sx → Dz	111110**********	1	—
	If DC = 1, nop	11011011xx00zzzz		
DCF PCOPY Sy,Dz	If DC = 0, Sy → Dz	111110**********	1	—
	If DC = 1, nop	1111101100yyzzzz		
PNEG Sx,Dz	0 – Sx → Dz	111110**********	1	Updated
		11001001xx00zzzz		
PNEG Sy,Dz	0 – Sy → Dz	111110**********	1	Updated
		1110100100yyzzzz		
DCT PNEG Sx,Dz	If DC = 1, 0 – Sx → Dz	111110**********	1	—
	If DC = 0, nop	11001010xx00zzzz		
DCT PNEG Sy,Dz	If DC = 1, 0 – Sy → Dz	111110**********	1	—
	If DC = 0, nop	1110101000yyzzzz		
DCF PNEG Sx,Dz	If DC = 0, 0 – Sx → Dz	111110**********	1	—
	If DC = 1, nop	11001011xx00zzzz		
DCF PNEG Sy,Dz	If DC = 0, 0 – Sy → Dz	111110**********	1	—
	If DC = 1, nop	1110101100yyzzzz		
PSUB Sx,Sy,Dz	Sx – Sy → Dz	111110**********	1	Updated
		10100001xxyyzzzz		
DCT PSUB Sx,Sy,Dz	If DC = 1, Sx – Sy → Dz	111110**********	1	—
	If DC = 0, nop	10100010xxyyzzzz		
DCF PSUB Sx,Sy,Dz	If DC = 0, Sx – Sy → Dz	111110**********	1	—
	If DC = 1, nop	10100011xxyyzzzz		
PSUB Sx,Sy,Du	Sx – Sy → Du	111110**********	1	Updated
PMULS Se,Sf,Dg	Se upper word × Sf upper word → Dg	0110eeffxxyygguu		
PSUBC Sx,Sy,Dz	Sx – Sy – DC → Dz	111110**********	1	Updated
		10100000xxyyzzzz		

Table 2.36 ALU Integer Operation Instructions

Instruction	Operation	Instruction Code	Execution States	DC Bit

PDEC Sx,Dz	Sx upper word – 1 → Dz upper word
	Dz lower word is cleared
		`111110**********`
		`10001001xx00zzzz`
			1	Updated

PDEC Sy,Dz	Sy upper word – 1 → Dz upper word
	Dz lower word is cleared
		`111110**********`
		`1010100100yyzzzz`
			1	Updated

DCT PDEC Sx,Dz	If DC = 1, Sx upper word – 1 → Dz upper word, Dz lower word is cleared
	If DC = 0, nop
		`111110**********`
		`10001010xx00zzzz`
			1	—

DCT PDEC Sy,Dz	If DC = 1, Sy upper word – 1 → Dz upper word, Dz lower word is cleared
	If DC = 0, nop
		`111110**********`
		`1010101000yyzzzz`
			1	—

DCF PDEC Sx,Dz	If DC = 0, Sx upper word – 1 → Dz upper word, Dz lower word is cleared
	If DC = 1, nop
		`111110**********`
		`10001011xx00zzzz`
			1	—

DCF PDEC Sy,Dz	If DC = 0, Sy upper word – 1 → Dz upper word, Dz lower word is cleared
	If DC = 1, nop
		`111110**********`
		`1010101100yyzzzz`
			1	—

PINC Sx,Dz	Sx upper word + 1 → Dz upper word
	Dz lower word is cleared
		`111110**********`
		`10011001xx00zzzz`
			1	Updated

PINC Sy,Dz	Sy upper word + 1 → Dz upper word
	Dz lower word is cleared
		`111110**********`
		`1011100100yyzzzz`
			1	Updated

Instruction	Operation	Instruction Code	Execution States	DC Bit
DCT PINC Sx,Dz	If DC = 1, Sx upper word + 1 → Dz upper word, Dz lower word is cleared If DC = 0, nop	111110********** 10011010xx00zzzz	1	—
DCT PINC Sy,Dz	If DC = 1, Sy upper word + 1 → Dz upper word, Dz lower word is cleared If DC = 0, nop	111110********** 1011101000yyzzzz	1	—
DCF PINC Sx,Dz	If DC = 0, Sx upper word + 1 → Dz upper word, Dz lower word is cleared If DC = 1, nop	111110********** 10011011xx00zzzz	1	—
DCF PINC Sy,Dz	If DC = 0, Sy upper word + 1 → Dz upper word, Dz lower word is cleared If DC = 1, nop	111110********** 1011101100yyzzzz	1	—

Table 2.37 MSB Detection Instructions

Instruction	Operation	Instruction Code	Execution States	DC Bit
PDMSB Sx,Dz	MSB position of Sx data → Dz upper word, Dz lower word is cleared	111110********** 10011101xx00zzzz	1	Updated
PDMSB Sy,Dz	MSB position of Sy data → Dz upper word, Dz lower word is cleared	111110********** 1011110100yyzzzz	1	Updated
DCT PDMSB Sx,Dz	If DC = 1, MSB position of Sx data → Dz upper word, Dz lower word is cleared If DC = 0, nop	111110********** 10011110xx00zzzz	1	—
DCT PDMSB Sy,Dz	If DC = 1, MSB position of Sy data → Dz upper word, Dz lower word is cleared If DC = 0, nop	111110********** 1011111000yyzzzz	1	—
DCF PDMSB Sx,Dz	If DC = 0, MSB position of Sx data → Dz upper word, Dz lower word is cleared If DC = 1, nop	111110********** 10011111xx00zzzz	1	—
DCF PDMSB Sy,Dz	If DC = 0, MSB position of Sy data → Dz upper word, Dz lower word is cleared If DC = 1, nop	111110********** 1011111100yyzzzz	1	—

Table 2.38 Rounding Operation Instructions

Instruction	Operation	Instruction Code	Execution States	DC Bit
PRND Sx,Dz	Sx + H'00008000 → Dz Dz lower word is cleared	111110********** 10011000xx00zzzz	1	Updated
PRND Sy,Dz	Sy + H'00008000 → Dz Dz lower word is cleared	111110********** 1011100000yyzzzz	1	Updated

ALU Logic Operation Instructions

Table 2.39 ALU Logic Operation Instructions

Instruction	Operation	Instruction Code	Execution States	DC Bit
PAND Sx,Sy,Dz	Sx & Sy → Dz Dz lower word is cleared	111110********** 10010101xxyyzzzz	1	Updated
DCT PAND Sx,Sy,Dz	If DC = 1, Sx&Sy → Dz, Dz lower word is cleared If DC = 0, nop	111110********** 10010110xxyyzzzz	1	—
DCF PAND Sx,Sy,Dz	If DC = 0, Sx&Sy → Dz, Dz lower word is cleared If DC = 1, nop	111110********** 10010111xxyyzzzz	1	—
POR Sx,Sy,Dz	Sx | Sy → Dz Dz lower word is cleared	111110********** 10110101xxyyzzzz	1	Updated
DCT POR Sx,Sy,Dz	If DC = 1, Sx|Sy → Dz, Dz lower word is cleared If DC = 0, nop	111110********** 10110110xxyyzzzz	1	—
DCF POR Sx,Sy,Dz	If DC = 0, Sx|Sy → Dz, Dz lower word is cleared If DC = 1, nop	111110********** 10110111xxyyzzzz	1	—
PXOR Sx,Sy,Dz	Sx ^ Sy → Dz Dz lower word is cleared	111110********** 10100101xxyyzzzz	1	Updated
DCT PXOR Sx,Sy,Dz	If DC = 1, Sx^Sy → Dz, Dz lower word is cleared If DC = 0, nop	111110********** 10100110xxyyzzzz	1	—
DCF PXOR Sx,Sy,Dz	If DC = 0, Sx^Sy → Dz, Dz lower word is cleared If DC = 1, nop	111110********** 10100111xxyyzzzz	1	—

Fixed-Point Multiply Instruction

Table 2.40 Fixed-Point Multiply Instruction

Instruction	Operation	Instruction Code	Execution States	DC Bit

PMULS Se,Sf,Dg	Se upper word × Sf → upper word → Dg	`111110**********`
		`0100eeff0000gg00`
			1	—

Shift Operation Instructions

Table 2.41 Arithmetic Shift Operation Instructions

Instruction	Operation	Instruction Code	Execution States	DC Bit

PSHA Sx,Sy,Dz	If Sy ≥ 0, Sx<<Sy → Dz
	If Sy < 0, Sx>>Sy → Dz
		`111110**********`
		`10010001xxyyzzzz`
			1	Updated

DCT PSHA Sx,Sy,Dz	If DC = 1 & Sy ≥ 0, Sx<<Sy → Dz
	If DC = 1 & Sy < 0, Sx>>Sy → Dz
	If DC = 0, nop
		`111110**********`
		`10010010xxyyzzzz`
			1	—

DCF PSHA Sx,Sy,Dz	If DC = 0 & Sy ≥ 0, Sx<<Sy → Dz
	If DC = 0 & Sy < 0, Sx>>Sy → Dz
	If DC = 1, nop
		`111110**********`
		`10010011xxyyzzzz`
			1	—

PSHA #imm,Dz	If imm ≥ 0, Dz<<imm → Dz
	If imm < 0, Dz>>imm → Dz
		`111110**********`
		`00010iiiiiiizzzz`
			1	Updated

Table 2.42 Logical Shift Operation Instructions

Instruction	Operation	Instruction Code	Execution States	DC Bit

PSHL Sx,Sy,Dz	If Sy ≥ 0, Sx<<Sy → Dz, Dz lower word is cleared
	If Sy < 0, Sx>>Sy → Dz, Dz lower word is cleared
		`111110**********`
		`10000001xxyyzzzz`
			1	Updated

DCT PSHL Sx,Sy,Dz	If DC = 1 & Sy ≥ 0, Sx<<Sy → Dz, Dz lower word is cleared
	If DC = 1 & Sy < 0, Sx>>Sy → Dz, Dz lower word is cleared
	If DC = 0, nop
		`111110**********`
		`10000010xxyyzzzz`
			1	—

DCF PSHL Sx,Sy,Dz	If DC = 0 & Sy ≥ 0, Sx<<Sy → Dz, Dz lower word is cleared
	If DC = 0 & Sy < 0, Sx>>Sy → Dz, Dz lower word is cleared
	If DC = 1, nop
		`111110**********`
		`10000011xxyyzzzz`
			1	—

PSHL #imm,Dz	If imm ≥ 0, Dz<<imm → Dz, Dz lower word is cleared
	If imm < 0, Dz>>imm → Dz, Dz lower word is cleared
		`111110**********`
		`00000iiiiiiizzzz`
			1	Updated

System Control Instructions

Table 2.43 System Control Instructions

Instruction	Operation	Instruction Code	Execution States	DC Bit
PLDS Dz,MACH	Dz → MACH	111110********** 111011010000zzzz	1	—
PLDS Dz,MACL	Dz → MACL	111110********** 111111010000zzzz	1	—
DCT PLDS Dz,MACH	If DC = 1, Dz → MACH If DC = 0, nop	111110********** 111011100000zzzz	1	—
DCT PLDS Dz,MACL	If DC = 1, Dz → MACL If DC = 0, nop	111110********** 111111100000zzzz	1	—
DCF PLDS Dz,MACH	If DC = 0, Dz → MACH If DC = 1, nop	111110********** 111011110000zzzz	1	—
DCF PLDS Dz,MACL	If DC = 0, Dz → MACL If DC = 1, nop	111110********** 111111110000zzzz	1	—
PSTS MACH,Dz	MACH → Dz	111110********** 110011010000zzzz	1	—
PSTS MACL,Dz	MACL → Dz	111110********** 110111010000zzzz	1	—
DCT PSTS MACH,Dz	If DC = 1, MACH → Dz If DC = 0, nop	111110********** 110011100000zzzz	1	—
DCT PSTS MACL,Dz	If DC = 1, MACL → Dz If DC = 0, nop	111110********** 110111100000zzzz	1	—
DCF PSTS MACH,Dz	If DC = 0, MACH → Dz If DC = 1, nop	111110********** 110011110000zzzz	1	—
DCF PSTS MACL,Dz	If DC = 0, MACL → Dz If DC = 1, nop	111110********** 110111110000zzzz	1	—

NOPX and NOPY Instruction Codes

When there is no data transfer instruction to be parallel-processed simultaneously with a DSP operation instruction, an NOPX or NOPY instruction can be written as the data transfer instruction, or the instruction can be omitted. The instruction code is the same whether an NOPX or NOPY instruction is written or the instruction is omitted. Examples of NOPX and NOPY instruction codes are shown in table 2.44.

Table 2.44 Sample NOPX and NOPY Instruction Codes

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

2.6 Usage Note

If a CPU instruction double-precision multiplication (MUL.L/DMUL.L/DMULS.L) or double-precision multiply-and-accumulate operation (MAC.L) and a DSP computational instruction are executed in combination, erroneous operation may occur.

1. Conditions for occurrence

   If conditions a. and b. below occur simultaneously, the instructions noted in b. (2) may operate erroneously.

   a. Execution of instruction from on-chip X/Y memory

   b. Execution of the following instruction sequence in the order (1) (2) (3)

      (1) Double-precision multiplication (MUL.L/DMUL.L/DMULS.L) or double-precision multiply-and-accumulate operation (MAC.L)

      (2) DSP computational instruction other than PMULS, PSTS, or PLDS

      (3) A PMULS, PSTS, or PLDS instruction

   The instructions in b. (1) above require a number of cycles for execution. Consequently, if an instruction of the kind noted in b. (3) that uses the same resources is executed during execution of the b. (1) instruction, the start of execution of (3) is suppressed until the executing computational operation ends. As the instructions noted in (2) have no correlation with the instructions noted in (1), the instruction is executed, but thereafter it may not be possible for execution of a (2) instruction to be completed correctly due to the operation that suppresses execution of (3) instructions. If there is no (2) instruction and a (1) instruction is followed immediately by a (3) instruction, there is no problem and the instructions will be executed normally. Also, there is no problem, and operation is normal, if instructions are executed from on-chip ROM or external memory in b. above.

   This also applies to the case where b. (1) above is immediately preceded by a delayed branch instruction, the b. (1) instruction is in the delay slot, and b. (2) and (3) are written at the branch destination.

Note: A “DSP computational instruction other than PMULS, PSTS, or PLDS” refers to any of the following instructions.
PABS, PADD, PADDC, PAND, PCLR, PCMP, PCOPY, PDEC, PDMSB, PINC, PNEG, POR, PRND, PSHA, PSHL, PSUB, PSUBC, PXOR

2. Programming countermeasure

   This restriction can be avoided by using any of methods (1) to (3) below.

   (1) Do not execute an instruction sequence conforming to condition b. above in on-chip X/Y memory.

   (2) If an instruction sequence conforming to condition b. above is present in instruction code and there is no problem if the order of instructions is changed, switch round instructions (2) and (3).

   (3) If an instruction sequence conforming to condition b. above is present in instruction code and there is a problem with changing the order of instructions, insert one or more NOP instructions, or CPU instructions unrelated to the multiplier, between instructions (1) and (2).

Section 3 Operating Modes

3.1 Operating Mode Selection

The SH7065 has ten operating modes. The settings of the mode pins (MD5 to MD0) determine the mode in which the chip operates. The mode pin settings must not be changed while the chip is operating (while power is being supplied).

The method of selecting the operating mode is shown in table 3.1

Table 3.1 Operating Mode Selection

Operating Mode No.	Mode Name	Pin Settings							On-Chip ROM	CS0 Bus Width (Bits)
		FWE	MD5	MD4	MD3	MD2	MD1*1	MD0		
0	Single-chip mode	0	Used for clock mode selection			0	0	0	Enabled	—
1	MCU mode 1	0				0	0	1	Enabled	8/16/32
2	MCU mode 2	0				0	1	0	Disabled	32
3	MCU mode 3	0				0	1	1	Disabled	16
4	MCU mode 4	0				1	0	0	Disabled	8
F0*3	User program mode (single-chip)	1				0	0	0	Enabled	—
F1*3	User program mode	1				0	0	1	Enabled	8/16/32
F2*3	Boot mode (single-chip)	1				0	1	0	Enabled	—
F3*3	Boot mode	1				0	1	1	Enabled	8/16/32
F7*3	PROM mode (programmer mode)	*2				1	1	1	Enabled	—
Other than the above	Reserved (Do not set)								—	—

Notes: 1. In the F-ZTAT version, it is possible to change MD1 during a power-on reset.
2. 0 or 1
3. F0 to F7 can only be used in the F-ZTAT version.

Table 3.2 shows the correspondence between the settings of mode pins MD5 to MD3 and the clock operating mode.

Table 3.2 Clock Operating Mode Selection

Mode No.	Pin Settings			Clock I/O		PLL Circuit 1	PLL Circuit 2	Initial State of CKIO Pin	Initial State of CK Pin	Initial Clock Ratio (Input Clock = 1)					Initial Value of FRQCR Register
	MD5	MD4	MD3	Supply Source	Output					CKM	CKP	CKE	CKIO	CK	
0	0	0	0	EXTAL or crystal resonator	CKIO, CK	ON (· 2)	ON (· 2)	Output	Output	· 1	· 1	· 1	· 2	· 1	H'C0AA
1	0	0	1							· 4	· 2	· 2	· 2	· 2	H'C045
2	0	1	0			ON (· 1)	ON (· 4)	Output	Output	· 1	· 1	· 1	· 4	· 1	H'C0AA
3	0	1	1							· 4	· 2	· 4	· 4	· 2	H'C044
4	1	0	0			OFF	ON (· 4)	Hi-Z	Output	· 1	· 1	· 1	· 4	· 1	H'40AA
5	1	0	1							· 4	· 2	· 2	· 4	· 2	H'4045
6	1	1	0	CKIO	CK	ON (· 2)	OFF	Input	Output	· 2	· 1	· 1	· 1	· 1	H'4045
7	1	1	1			ON (· 1)				· 1	· 1	· 1	· 1	· 1	H'4000

3.1.1 Operating Modes

Mode 0 (Single-Chip Mode): In single-chip mode any port can be used, but external addresses cannot be used.

Mode 1 (MCU Mode 1): In mode 1, on-chip ROM is enabled. The bus width for on-chip ROM space is 32 bits.

Mode 2 (MCU Mode 2): In mode 2, external memory space with a 32-bit CS0 space bus width is used.

Mode 3 (MCU Mode 3): In mode 3, external memory space with a 16-bit CS0 space bus width is used.

Mode 4 (MCU Mode 4): In mode 4, external memory space with an 8-bit CS0 space bus width is used.

Modes F0 and F1 (User Program Modes): In the user program modes, on-chip flash memory can be programmed, erased, and verified on-board. For details, see section 19, 256 kB Flash Memory (F-ZTAT).

Modes F2 and F3 (Boot Modes): In the boot modes, on-chip flash memory can be programmed, erased, and verified on-board. For details, see section 19, 256 kB Flash Memory (F-ZTAT).

Mode F7 (PROM (Programmer) Mode): In PROM (programmer) mode, on-chip flash memory can be programmed using a PROM programmer recommended by Renesas. For details, see section 19, 256 kB Flash Memory (F-ZTAT).

3.1.2 Pin Configuration

The functions of pins relating to the operating modes are shown in table 3.3.

Table 3.3 Pin Functions

Pin Name	I/O	Function
MD0	Input	The level at this pin is used in the operating mode specification.
MD1	Input	The level at this pin is used in the operating mode specification.
MD2	Input	The level at this pin is used in the operating mode specification.
MD3	Input	The level at this pin is used in the clock mode specification.
MD4	Input	The level at this pin is used in the clock mode specification.
MD5	Input	The level at this pin is used in the clock mode specification.
FWE	Input	Used for hardware protection against on-chip flash memory programming/erasing. In the mask ROM version this pin is V_{SS}.

3.1.3 Register Configuration

Table 3.4 summarizes the registers relating to the operating modes.

Table 3.4 Registers

Name	Abbreviation	R/W	Initial Value	Address	Access Size
Mode status register	MSR	R	—	H'FFFF1020	8, 16, 32
Mode control register	MODECR	R/W	H'001C	H'FFFF102A	8, 16, 32

3.2 Register Descriptions

3.2.1 Mode Status Register (MSR)

The mode status register (MSR) is a 16-bit read-only register used to monitor the operating mode status.

Bit:	15	14	13	12	11	10	9	8
	—	—	—	—	—	—	—	—
Initial value:	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
R/W:	R	R	R	R	R	R	R	R

Bit:	7	6	5	4	3	2	1	0
	—	—	MD5	MD4	MD3	MD2	MD1	MD0
Initial value:	Undefined	Undefined	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R

Bits 15 to 6—Reserved: These bits always return an undefined value when read.

Bits 5 to 0—Mode (MD5 to MD0): These bits indicate the mode pin states in a power-on reset.

Bits 5 to 0: **MD5 to MD0**	**Description**
0 or 1	Indicates the mode pin state

3.2.2 Mode Control Register (MODECR)

The mode control register (MODECR) is a 16-bit readable/writable register that selects the on-chip ROM access mode.

MODECR is initialized to H'001C by a power-on reset, but is not initialized in standby mode.

Bit:	15	14	13	12	11	10	9	8
	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

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

Bits 15 to 5—Reserved: These bits are always read as 0 and should only be written with 0.

Bit 4—ROM Access Mode (ROMMD): Selects the on-chip ROM access mode. Normally, high-speed mode should be used. For details, see section 8.4, Number of Access Cycles (SH7065A), On-Chip ROM.

Bit 4: ROMMD	Description	
0	On-chip ROM is accessed in high-speed mode	
1	On-chip ROM is accessed in low-speed mode	(Initial value)

Bits 3 and 2—Reserved: These bits are always read as 1 and should only be written with 1.

Bits 1 and 0—Reserved: These bits are always read as 0 and should only be written with 0.

Section 4 Clock Pulse Generator (CPG) and Power-Down Modes

4.1 Overview

The SH7065 has an on-chip clock pulse generator (CPG) which is used to generate the clocks supplied internally and control the power-down modes. In the power-down modes, the operation of the on-chip peripheral modules and CPU, or of all functions, is halted. It is also possible to select a division ratio for the clock supplied to individual modules, even during operation. These features enable power consumption to be reduced.

4.1.1 Features

The CPG has the following features:

- Eight clock modes

  Any of eight clock operating modes can be selected, differentiated by frequency range, power consumption, and use of a crystal resonator or external clock input.
- Three clocks

  The CPG can generate independently a master clock (CKM) used by the CPU, etc., a peripheral clock (CKP) used by the peripheral modules, and an external bus clock (CKE) used by the external bus interface.
- Frequency modification function

  PLL (phase-locked loop) circuits and frequency dividers in the CPG enable the frequencies of the master clock, peripheral clock, and external bus clock to be changed independently. Frequency changes are performed by software in accordance with the settings in the frequency control register (FRQCR).

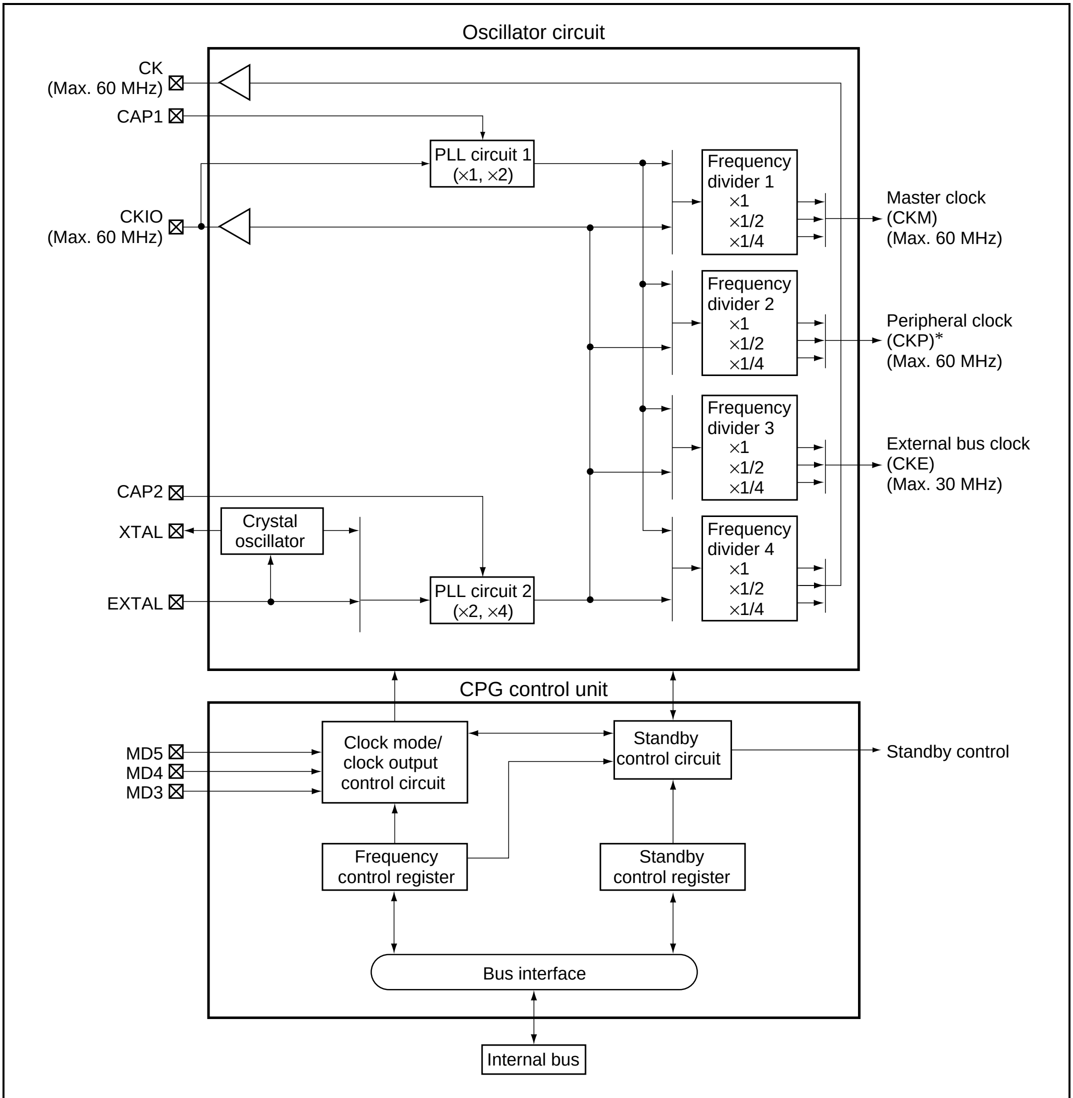
The power-down modes include the following modes and functions:

- Power-down mode control

  It is possible to stop the clock in sleep mode, software standby mode, and hardware standby mode, to stop the clock supply to specific modules with the module standby function, and to divide the frequency of clocks supplied to specific modules with the module clock division function.

4.1.2 Block Diagram of CPG

Figure 4.1 shows a block diagram of the CPG.



Note: * Peripheral clock CKP is divided by the module clock division function before being input to individual modules. The maximum operating frequency of the peripheral modules is 30 MHz.
For details, see section 4.14, Module Clock Division Function.

Figure 4.1 Block Diagram of CPG

The function of each of the CPG blocks is described below.

PLL Circuit 1: PLL circuit 1 has the function of multiplying the frequency of the clock from the CKIO pin or PLL circuit 2 by a factor of 1 or 2. The phase of the rising edge of the external bus clock (CKE) is controlled so that it matches the phase of the rising edge at the CKIO pin. The multiplication factor is determined by the clock operating mode.

PLL Circuit 2: PLL circuit 2 has the function of multiplying the frequency of the input clock from the crystal oscillator or EXTAL pin by a factor of 2 or 4. The multiplication factor is determined by the clock operating mode.

Crystal Oscillator: This is the oscillator circuit used when a crystal resonator is connected to the XTAL and EXTAL pins. Use of the crystal oscillator can be selected by a clock operating mode setting,

Frequency Divider 1: Frequency divider 1 has the function of generating the master clock (CKM). The master clock (CKM) operating frequency can be selected from 4, 2, 1, 1/2, or 1/4 times the input clock frequency according to the clock mode. The division ratio is set in the frequency control register.

Frequency Divider 2: Frequency divider 2 has the function of generating the peripheral clock (CKP). The peripheral clock (CKP) operating frequency can be selected from 4, 2, 1, 1/2, or 1/4 times the input clock frequency according to the clock mode. The division ratio is set in the frequency control register.

Frequency Divider 3: Frequency divider 3 has the function of generating the external bus clock (CKE). The external bus clock (CKE) operating frequency can be selected from 4, 2, 1, 1/2, or 1/4 times the input clock frequency according to the clock mode. The division ratio is set in the frequency control register.

Frequency Divider 4: Frequency divider 4 has the function of generating external clock output (CK). The external clock output (CK) operating frequency can be selected from 4, 2, 1, 1/2, or 1/4 times the input clock frequency according to the clock mode. The division ratio is set in the frequency control register.

Clock Mode/Clock Output Control Circuit: The clock mode/clock output control circuit controls the clock mode and the clock output from the CK/CKIO pin by means of the frequency control register.

Standby Control Circuit: The standby control circuit controls the state of the on-chip oscillator circuit and other modules when the clock is switched and in sleep and standby modes.

Frequency Control Register: The frequency control register contains control bits for on/off control of the clock output from the CK/CKIO pin, and the frequency division ratios for the master clock, peripheral clock, external bus clock, and clock output.

Standby Control Register: The standby control register contains power-down mode control bits.

4.1.3 CPG Pin Configuration

Table 4.1 shows the CPG pins and their functions.

Table 4.1 CPG Pins

Pin Name	Abbreviation	I/O	Function
Mode control pins	MD5–MD3	Input	Set clock operating mode.
Crystal input/output pins (clock input pins)	XTAL	Output	Connects crystal resonator.
	EXTAL	Input	Connects crystal resonator, or used as external clock input pin.
Clock input/output pin	CKIO	I/O	Used as external clock input or external clock output pin. In output mode, can be fixed in high-impedance state.
	CK	Output	Used as external clock output pin. Can be fixed in high-impedance state.
PLL capacitance connection pins	CAP1	Input	Connects capacitance (recommended value: 470 pF) for PLL circuit 1 operation.
	CAP2	Input	Connects capacitance (recommended value: 470 pF) for PLL circuit 2 operation.

4.1.4 CPG Register Configuration

Table 4.2 shows the CPG register configuration.

Table 4.2 CPG Register

Name	Abbreviation	R/W	Initial Value	Address	Access Size
Frequency control register	FRQCR	R/W	Depends on clock mode	H'FFFF 1028	8, 16, 32

4.2 Clock Operating Modes

Table 4.3 shows the clock operating modes corresponding to various combinations of mode control pin (MD5 to MD3) settings.

Table 4.3 Clock Operating Mode Settings

Mode No.	Pin Settings			Clock Input/Output		PLL Circuit 1	PLL Circuit 2	CKIO Pin Initial State	CK Pin Initial State	Clock Ratio Initial Value (Input Clock = 1)					FRQCR Register Initial Value
	MD5	MD4	MD3	Supply Source	Output					CKM*1	CKP*2	CKE*3	CKIO	CK	
0	0	0	0	EXTAL or crystal resonator	CKIO, CK	ON (×2)	ON (×2)	Output	Output	×1	×1	×1	×2	×1	H'C0AA
1	0	0	1							×4	×2	×2	×2	×2	H'C045
2	0	1	0			ON (×1)	ON (×4)	Output	Output	×1	×1	×1	×4	×1	H'C0AA
3	0	1	1							×4	×2	×4	×4	×2	H'C044
4	1	0	0			OFF	ON (×4)	Hi-Z	Output	×1	×1	×1	×4	×1	H'40AA
5	1	0	1							×4	×2	×2	×4	×2	H'4045
6	1	1	0	CKIO	CK	ON (×2)	OFF	Input	Output	×2	×1	×1	×1	×1	H'4045
7	1	1	1			ON (×1)				×1	×1	×1	×1	×1	H'4000

Notes: 1. Master clock
2. Peripheral clock
3. External bus clock

Modes 0 and 1

An external clock is input from the EXTAL pin, or a crystal resonator is connected, and PLL circuits 1 and 2 operate. The frequency multiplication factor is fixed at 2 for both PLL circuit 1 and PLL circuit 2.

When the CKE clock is multiplied by 2 by means of a setting in the frequency control register (FRQCR), a clock in phase with the CKE clock is output from the CKIO pin. When the CKE clock is multiplied by 4, switching of CKIO output coincides with the rise of the CKE clock. When the CKE clock is multiplied by 1, the rise of the CKIO output coincides with switching of the CKE clock.

A clock with the frequency set by FRQCR (without phase coordination) is output from the CK pin.

The CK pin can be set to the high-impedance state by means of a setting in FRQCR.

Modes 2 and 3

An external clock is input from the EXTAL pin, or a crystal resonator is connected, and PLL circuits 1 and 2 operate. The frequency multiplication factor is fixed at 1 for PLL circuit 1 and 4 for PLL circuit 2.

When the CKE clock is multiplied by 4 by means of a setting in the frequency control register (FRQCR), a clock in phase with the CKE clock is output from the CKIO pin. When the CKE clock is multiplied by 1 or 2, the rise of the CKIO output coincides with switching of the CKE clock.

A clock with the frequency set by FRQCR (without phase coordination) is output from the CK pin.

The CK pin can be set to the high-impedance state by means of a setting in FRQCR.

Modes 4 and 5

An external clock is input from the EXTAL pin, or a crystal resonator is connected, and PLL circuit 2 operates. The frequency multiplication factor is fixed at 4.

The CKIO pin is in the high-impedance state. PLL circuit 1 is off, and phase coordination is not performed.

A clock with the frequency set by FRQCR (without phase coordination) is output from the CK pin.

The CK pin can be set to the high-impedance state and a clock output from the CKIO pin by means of settings in FRQCR.

Mode 6

An external clock is input from the CKIO pin, and PLL circuit 1 operates. The frequency multiplication factor is fixed at 2. PLL circuit 2 is off.

When the CKE clock is multiplied by 1 by means of a setting in FRQCR, a CKE clock in phase with the CKIO pin is output. When the CKE clock is multiplied by 1/2, the rise of the CKIO output coincides with switching of the CKE clock. When the CKE clock is multiplied by 2, switching of CKIO output coincides with the rise of the CKE clock.

A clock with the frequency set by FRQCR (without phase coordination) is output from the CK pin.

The CK pin can be set to the high-impedance state by means of a setting in FRQCR.

Mode 7

An external clock is input from the CKIO pin, and PLL circuit 1 operates. The frequency multiplication factor is fixed at 1. PLL circuit 2 is off.

When the CKE clock is multiplied by 1 by means of a setting in FRQCR, a CKE clock in phase with the CKIO pin is output. When the CKE clock is multiplied by 1/2 or 1/4, the rise of the CKIO output coincides with switching of the CKE clock.

A clock with the frequency set by FRQCR (without phase coordination) is output from the CK pin.

The CK pin can be set to the high-impedance state by means of a setting in FRQCR.

4.3 CPG Register Description

4.3.1 Frequency Control Register (FRQCR)

The frequency control register (FRQCR) is a 16-bit readable/writable register used for on/off control of clock output from the CKIO/CK pin, and specification of the frequency division ratios for the master clock, peripheral clock, external bus clock, and clock output.

FRQCR is initialized to a value determined by the clock mode in a power-on reset, but is not initialized in standby mode.

Bit:	15	14	13	12	11	10	9	8
	CKIOOE	CKOE	—	—	—	—	—	—
Initial value:	—	1	0	0	0	0	0	0
R/W:	R/W	R/W	R	R	R	R	R	R

Bit:	7	6	5	4	3	2	1	0
	FR7	FR6	FR5	FR4	FR3	FR2	FR1	FR0
Initial value:	—	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 15—CKIO Output Enable (CKIOOE): Specifies whether the CKIO pin outputs a clock or goes to the high-impedance state. The initial value is determined by the clock operating mode. In clock modes 6 and 7, CKIO is an input pin, and the initial value of this bit is 0.

Do not write 0 to this bit in clock operating mode 0, 1, 2, or 3, and do not write 1 in clock operating mode 6 or 7.

Bit 15: CKIOOE	Description
0	CKIO pin goes to high-impedance state (initial value in clock operating modes 4, 5, 6, and 7)
1	CKIO pin outputs a clock (initial value in clock operating modes 0, 1, 2, and 3)

Bit 14—CK Output Enable (CKOE): Specifies whether the CK pin outputs a clock or goes to the high-impedance state.

Bit 14: CKOE	Description	
0	CK pin goes to high-impedance state	
1	CK pin outputs a clock	(Initial value)

Bits 13 to 8—Reserved: These bits are always read as 0, and should only be written with 0.

Bits 7 to 0—Frequency Setting (FR7 to FR0): These bits set the frequency of the master clock (CKM), peripheral clock (CKP), external bus clock (CKE), and clock output (CK). The initial value is determined by the clock operating mode.

Table 4.8 shows settings of FR7 to FR0 and the corresponding frequency ratios of the master clock (CKM), peripheral clock (CKP), external bus clock (CKE), clock output (CK), and CKIO pin, taking the external input clock frequency as 1.

Table 4.4 Frequency Divider 1 Control (CKM)

FR5	FR4	Frequency Divider 1 Control
0	0	×1
	1	×1/2
1	0	×1/4
	1	Do not set

Table 4.5 Frequency Divider 2 Control (CKP)

FR3	FR2	Frequency Divider 2 Control
0	0	×1
	1	×1/2
1	0	×1/4
	1	Do not set

Table 4.6 Frequency Divider 3 Control (CKE)

FR1	FR0	Frequency Divider 3 Control
0	0	×1
	1	×1/2
1	0	×1/4
	1	Do not set

Table 4.7 Frequency Divider 4 Control (CK Pin)

FR7	FR6	Frequency Divider 4 Control
0	0	×1
	1	×1/2
1	0	×1/4
	1	Do not set

Table 4.8 (1) FR Register Values and Frequency Ratios (Input Clock = 1)

Clock Modes 0 and 1

FR Register Values								PLL Circuit 1 Multiplication Factor	PLL Circuit 2 Multiplication Factor	Clock Ratio					Clock Input Frequency Range (MHz)
FR 7	FR 6	FR 5	FR 4	FR 3	FR 2	FR 1	FR 0			CKM	CKP	CKE	CKIO	CK	
0	0	0	0	0	0	0	0	×2	×2	×4	×4	×4	×2	×4	5–15
0	1													×2	
1	0													×1	
0	0					0	1					×2		×4	
0	1													×2	
1	0													×1	
0	0					1	0					×1		×4	
0	1													×2	
1	0													×1	
0	0			0	1	0	0				×2	×4		×4	
0	1													×2	
1	0													×1	
0	0					0	1					×2		×4	
0	1													×2	
1	0													×1	
0	0					1	0					×1		×4	
0	1													×2	
1	0													×1	
0	0			1	0	0	0				×1	×4		×4	
0	1													×2	
1	0													×1	
0	0					0	1					×2		×4	
0	1													×2	
1	0													×1	
0	0					1	0					×1		×4	
0	1													×2	
1	0													×1	

FR Register Values								PLL Circuit 1 Multiplication Factor	PLL Circuit 2 Multiplication Factor	Clock Ratio					Clock Input Frequency Range (MHz)
FR 7	FR 6	FR 5	FR 4	FR 3	FR 2	FR 1	FR 0			CKM	CKP	CKE	CKIO	CK	
0	0	0	1	0	0	0	0	×2	×2	×2	×4	×4	×2	×4	5–15
0	1													×2	
1	0													×1	
0	0					0	1					×2		×4	
0	1													×2	
1	0													×1	
0	0					1	0					×1		×4	
0	1													×2	
1	0													×1	
0	0			0	1	0	0				×2	×4		×4	
0	1													×2	
1	0													×1	
0	0					0	1					×2		×4	
0	1													×2	
1	0													×1	
0	0					1	0					×1		×4	
0	1													×2	
1	0													×1	
0	0			1	0	0	0				×1	×4		×4	
0	1													×2	
1	0													×1	
0	0					0	1					×2		×4	
0	1													×2	
1	0													×1	
0	0					1	0					×1		×4	
0	1													×2	
1	0													×1	

FR Register Values								PLL Circuit 1 Multiplication Factor	PLL Circuit 2 Multiplication Factor	Clock Ratio					Clock Input Frequency Range (MHz)
FR 7	FR 6	FR 5	FR 4	FR 3	FR 2	FR 1	FR 0			CKM	CKP	CKE	CKIO	CK	
0	0	1	0	0	0	0	0	×2	×2	×1	×4	×4	×2	×4	5–15
0	1													×2	
1	0													×1	
0	0					0	1					×2		×4	
0	1													×2	
1	0													×1	
0	0					1	0					×1		×4	
0	1													×2	
1	0													×1	
0	0			0	1	0	0				×2	×4		×4	
0	1													×2	
1	0													×1	
0	0					0	1					×2		×4	
0	1													×2	
1	0													×1	
0	0					1	0					×1		×4	
0	1													×2	
1	0													×1	
0	0			1	0	0	0				×1	×4		×4	
0	1													×2	
1	0													×1	
0	0					0	1					×2		×4	
0	1													×2	
1	0													×1	
0	0					1	0					×1		×4	
0	1													×2	
1	0													×1	

Notes: The lower and upper limits of the clock input frequency range are determined by the following conditions:

1. Lower-limit frequency

   The output frequency of each PLL before division must be at least 10 MHz. The specific clock input frequency lower limits are as follows: 2.5 MHz in clock modes 2, 3, 4, and 5 (as the PLL2 multiplication factor is ×4), 5 MHz in clock modes 0 and 1 (as the PLL2 multiplication factor is ×2), 5 MHz also in clock mode 6 (as the PLL1 multiplication factor is ×2), and 10 MHz in clock mode 7 (as the PLL1 multiplication factor is ×1).

2. Upper-limit frequency

   (1) Clock upper limits after division according to FRQCR register setting

   CKM ≤ 60 MHz, CKP ≤ 60 MHz, CKE ≤ 30 MHz

   (2) Clock upper limits after division according to MCLKCR1-5 register setting

   Mϕ ≤ 60 MHz, Pϕ ≤ 30 MHz

   The frequency that satisfies both (1) and (2) above is the clock input frequency upper limit.

Table 4.8 (2) FR Register Values and Frequency Ratios (Input Clock = 1)

Clock Modes 2 and 3

FR Register Values								PLL Circuit 1 Multiplication Factor	PLL Circuit 2 Multiplication Factor	Clock Ratio					Clock Input Frequency Range (MHz)
FR 7	FR 6	FR 5	FR 4	FR 3	FR 2	FR 1	FR 0			CKM	CKP	CKE	CKIO	CK	
0	0	0	0	0	0	0	0	×1	×4	×4	×4	×4	×4	×4	2.5–15
0	1													×2	
1	0													×1	
0	0					0	1					×2		×4	
0	1													×2	
1	0													×1	
0	0					1	0					×1		×4	
0	1													×2	
1	0													×1	
0	0			0	1	0	0				×2	×4		×4	
0	1													×2	
1	0													×1	
0	0					0	1					×2		×4	
0	1													×2	
1	0													×1	
0	0					1	0					×1		×4	
0	1													×2	
1	0													×1	
0	0			1	0	0	0				×1	×4		×4	
0	1													×2	
1	0													×1	
0	0					0	1					×2		×4	
0	1													×2	
1	0													×1	
0	0					1	0					×1		×4	
0	1													×2	
1	0													×1	

FR Register Values								PLL Circuit 1 Multiplication Factor	PLL Circuit 2 Multiplication Factor	Clock Ratio					Clock Input Frequency Range (MHz)
FR 7	FR 6	FR 5	FR 4	FR 3	FR 2	FR 1	FR 0			CKM	CKP	CKE	CKIO	CK	
0	0	0	1	0	0	0	0	×1	×4	×2	×4	×4	×4	×4	2.5–15
0	1													×2	
1	0													×1	
0	0					0	1					×2		×4	
0	1													×2	
1	0													×1	
0	0					1	0					×1		×4	
0	1													×2	
1	0													×1	
0	0			0	1	0	0				×2	×4		×4	
0	1													×2	
1	0													×1	
0	0					0	1					×2		×4	
0	1													×2	
1	0													×1	
0	0					1	0					×1		×4	
0	1													×2	
1	0													×1	
0	0			1	0	0	0				×1	×4		×4	
0	1													×2	
1	0													×1	
0	0					0	1					×2		×4	
0	1													×2	
1	0													×1	
0	0					1	0					×1		×4	
0	1													×2	
1	0													×1	

FR Register Values								PLL Circuit 1 Multiplication Factor	PLL Circuit 2 Multiplication Factor	Clock Ratio					Clock Input Frequency Range (MHz)
FR 7	FR 6	FR 5	FR 4	FR 3	FR 2	FR 1	FR 0			CKM	CKP	CKE	CKIO	CK	
0	0	1	0	0	0	0	0	×1	×4	×1	×4	×4	×4	×4	2.5–15
0	1													×2	
1	0													×1	
0	0					0	1					×2		×4	
0	1													×2	
1	0													×1	
0	0					1	0					×1		×4	
0	1													×2	
1	0													×1	
0	0			0	1	0	0				×2	×4		×4	
0	1													×2	
1	0													×1	
0	0					0	1					×2		×4	
0	1													×2	
1	0													×1	
0	0					1	0					×1		×4	
0	1													×2	
1	0													×1	
0	0			1	0	0	0				×1	×4		×4	
0	1													×2	
1	0													×1	
0	0					0	1					×2		×4	
0	1													×2	
1	0													×1	
0	0					1	0					×1		×4	
0	1													×2	
1	0													×1	

Notes: The lower and upper limits of the clock input frequency range are determined by the following conditions:

1. Lower-limit frequency

   The output frequency of each PLL before division must be at least 10 MHz. The specific clock input frequency lower limits are as follows: 2.5 MHz in clock modes 2, 3, 4, and 5 (as the PLL2 multiplication factor is ×4), 5 MHz in clock modes 0 and 1 (as the PLL2 multiplication factor is ×2), 5 MHz also in clock mode 6 (as the PLL1 multiplication factor is ×2), and 10 MHz in clock mode 7 (as the PLL1 multiplication factor is ×1).

2. Upper-limit frequency

   (1) Clock upper limits after division according to FRQCR register setting

   CKM ≤ 60 MHz, CKP ≤ 60 MHz, CKE ≤ 30 MHz

   (2) Clock upper limits after division according to MCLKCR1-5 register setting

   Mφ ≤ 60 MHz, Pφ ≤ 30 MHz

   The frequency that satisfies both (1) and (2) above is the clock input frequency upper limit.

Table 4.8 (3) FR Register Values and Frequency Ratios (Input Clock = 1)

Clock Modes 4 and 5

FR Register Values								PLL Circuit 1 Multiplication Factor	PLL Circuit 2 Multiplication Factor	Clock Ratio					Clock Input Frequency Range (MHz)
FR 7	FR 6	FR 5	FR 4	FR 3	FR 2	FR 1	FR 0			CKM	CKP	CKE	CKIO	CK	
0	0	0	0	0	0	0	0	—	×4	×4	×4	×4	×4	×4	2.5–15
0	1													×2	
1	0													×1	
0	0					0	1					×2		×4	
0	1													×2	
1	0													×1	
0	0					1	0					×1		×4	
0	1													×2	
1	0													×1	
0	0			0	1	0	0				×2	×4		×4	
0	1													×2	
1	0													×1	
0	0					0	1					×2		×4	
0	1													×2	
1	0													×1	
0	0					1	0					×1		×4	
0	1													×2	
1	0													×1	
0	0			1	0	0	0				×1	×4		×4	
0	1													×2	
1	0													×1	
0	0					0	1					×2		×4	
0	1													×2	
1	0													×1	
0	0					1	0					×1		×4	
0	1													×2	
1	0													×1	

FR Register Values								PLL Circuit 1 Multiplication Factor	PLL Circuit 2 Multiplication Factor	Clock Ratio					Clock Input Frequency Range (MHz)
FR 7	FR 6	FR 5	FR 4	FR 3	FR 2	FR 1	FR 0			CKM	CKP	CKE	CKIO	CK	
0	0	0	1	0	0	0	0	—	×4	×2	×4	×4	×4	×4	2.5–15
0	1													×2	
1	0													×1	
0	0					0	1					×2		×4	
0	1													×2	
1	0													×1	
0	0					1	0					×1		×4	
0	1													×2	
1	0													×1	
0	0			0	1	0	0				×2	×4		×4	
0	1													×2	
1	0													×1	
0	0					0	1					×2		×4	
0	1													×2	
1	0													×1	
0	0					1	0					×1		×4	
0	1													×2	
1	0													×1	
0	0			1	0	0	0				×1	×4		×4	
0	1													×2	
1	0													×1	
0	0					0	1					×2		×4	
0	1													×2	
1	0													×1	
0	0					1	0					×1		×4	
0	1													×2	
1	0													×1	

FR Register Values								PLL Circuit 1 Multiplication Factor	PLL Circuit 2 Multiplication Factor	Clock Ratio					Clock Input Frequency Range (MHz)
FR 7	FR 6	FR 5	FR 4	FR 3	FR 2	FR 1	FR 0			CKM	CKP	CKE	CKIO	CK	
0	0	1	0	0	0	0	0	—	×4	×1	×4	×4	×4	×4	2.5–15
0	1													×2	
1	0													×1	
0	0					0	1					×2		×4	
0	1													×2	
1	0													×1	
0	0					1	0					×1		×4	
0	1													×2	
1	0													×1	
0	0			0	1	0	0				×2	×4		×4	
0	1													×2	
1	0													×1	
0	0					0	1					×2		×4	
0	1													×2	
1	0													×1	
0	0					1	0					×1		×4	
0	1													×2	
1	0													×1	
0	0			1	0	0	0				×1	×4		×4	
0	1													×2	
1	0													×1	
0	0					0	1					×2		×4	
0	1													×2	
1	0													×1	
0	0					1	0					×1		×4	
0	1													×2	
1	0													×1	

Notes: The lower and upper limits of the clock input frequency range are determined by the following conditions:

1. Lower-limit frequency

   The output frequency of each PLL before division must be at least 10 MHz. The specific clock input frequency lower limits are as follows: 2.5 MHz in clock modes 2, 3, 4, and 5 (as the PLL2 multiplication factor is ×4), 5 MHz in clock modes 0 and 1 (as the PLL2 multiplication factor is ×2), 5 MHz also in clock mode 6 (as the PLL1 multiplication factor is ×2), and 10 MHz in clock mode 7 (as the PLL1 multiplication factor is ×1).

2. Upper-limit frequency

   (1) Clock upper limits after division according to FRQCR register setting

   CKM ≤ 60 MHz, CKP ≤ 60 MHz, CKE ≤ 30 MHz

   (2) Clock upper limits after division according to MCLKCR1-5 register setting

   Mϕ ≤ 60 MHz, Pϕ ≤ 30 MHz

   The frequency that satisfies both (1) and (2) above is the clock input frequency upper limit.

Table 4.8 (4) FR Register Values and Frequency Ratios (Input Clock = 1)

Clock Mode 6

FR Register Values								PLL Circuit 1 Multiplication Factor	PLL Circuit 2 Multiplication Factor	Clock Ratio					Clock Input Frequency Range (MHz)
FR 7	FR 6	FR 5	FR 4	FR 3	FR 2	FR 1	FR 0			CKM	CKP	CKE	CKIO	CK	
0	0	0	0	0	0	0	0	×2	—	×2	×2	×2	—	×2	5–30
0	1													×1	
1	0													×1/2	
0	0					0	1					×1		×2	
0	1													×1	
1	0													×1/2	
0	0					1	0					×1/2		×2	
0	1													×1	
1	0													×1/2	
0	0			0	1	0	0				×1	×2		×2	
0	1													×1	
1	0													×1/2	
0	0					0	1					×1		×2	
0	1													×1	
1	0													×1/2	
0	0					1	0					×1/2		×2	
0	1													×1	
1	0													×1/2	
0	0			1	0	0	0				×1/2	×2		×2	
0	1													×1	
1	0													×1/2	
0	0					0	1					×1		×2	
0	1													×1	
1	0													×1/2	
0	0					1	0					×1/2		×2	
0	1													×1	
1	0													×1/2	

FR Register Values								PLL Circuit 1 Multiplication Factor	PLL Circuit 2 Multiplication Factor	Clock Ratio					Clock Input Frequency Range (MHz)
FR 7	FR 6	FR 5	FR 4	FR 3	FR 2	FR 1	FR 0			CKM	CKP	CKE	CKIO	CK	
0	0	0	1	0	0	0	0	×2	—	×1	×2	×2	—	×2	5–30
0	1													×1	
1	0													×1/2	
0	0					0	1					×1		×2	
0	1													×1	
1	0													×1/2	
0	0					1	0					×1/2		×2	
0	1													×1	
1	0													×1/2	
0	0			0	1	0	0				×1	×2		×2	
0	1													×1	
1	0													×1/2	
0	0					0	1					×1		×2	
0	1													×1	
1	0													×1/2	
0	0					1	0					×1/2		×2	
0	1													×1	
1	0													×1/2	
0	0			1	0	0	0				×1/2	×2		×2	
0	1													×1	
1	0													×1/2	
0	0					0	1					×1		×2	
0	1													×1	
1	0													×1/2	
0	0					1	0					×1/2		×2	
0	1													×1	
1	0													×1/2	

FR Register Values								PLL Circuit 1 Multiplication Factor	PLL Circuit 2 Multiplication Factor	Clock Ratio					Clock Input Frequency Range (MHz)
FR 7	**FR 6**	**FR 5**	**FR 4**	**FR 3**	**FR 2**	**FR 1**	**FR 0**			**CKM**	**CKP**	**CKE**	**CKIO**	**CK**	
0	0	1	0	0	0	0	0	×2	—	×1/2	×2	×2	—	×2	5–30
0	1													×1	
1	0													×1/2	
0	0					0	1					×1		×2	
0	1													×1	
1	0													×1/2	
0	0					1	0					×1/2		×2	
0	1													×1	
1	0													×1/2	
0	0			0	1	0	0				×1	×2		×2	
0	1													×1	
1	0													×1/2	
0	0					0	1					×1		×2	
0	1													×1	
1	0													×1/2	
0	0					1	0					×1/2		×2	
0	1													×1	
1	0													×1/2	
0	0			1	0	0	0				×1/2	×2		×2	
0	1													×1	
1	0													×1/2	
0	0					0	1					×1		×2	
0	1													×1	
1	0													×1/2	
0	0					1	0					×1/2		×2	
0	1													×1	
1	0													×1/2	

Notes: The lower and upper limits of the clock input frequency range are determined by the following conditions:

1. Lower-limit frequency

   The output frequency of each PLL before division must be at least 10 MHz. The specific clock input frequency lower limits are as follows: 2.5 MHz in clock modes 2, 3, 4, and 5 (as the PLL2 multiplication factor is ×4), 5 MHz in clock modes 0 and 1 (as the PLL2 multiplication factor is ×2), 5 MHz also in clock mode 6 (as the PLL1 multiplication factor is ×2), and 10 MHz in clock mode 7 (as the PLL1 multiplication factor is ×1).

2. Upper-limit frequency

   (1) Clock upper limits after division according to FRQCR register setting

   CKM ≤ 60 MHz, CKP ≤ 60 MHz, CKE ≤ 30 MHz

   (2) Clock upper limits after division according to MCLKCR1-5 register setting

   $M\phi \leq 60$ MHz, $P\phi \leq 30$ MHz

   The frequency that satisfies both (1) and (2) above is the clock input frequency upper limit.

Table 4.8 (5) FR Register Values and Frequency Ratios (Input Clock = 1)

Clock Mode 7

FR Register Values								PLL Circuit 1 Multiplication Factor	PLL Circuit 2 Multiplication Factor	Clock Ratio					Clock Input Frequency Range (MHz)
FR 7	FR 6	FR 5	FR 4	FR 3	FR 2	FR 1	FR 0			CKM	CKP	CKE	CKIO	CK	
0	0	0	0	0	0	0	0	×1	—	×1	×1	×1	—	×1	10–30
0	1													×1/2	
1	0													×1/4	
0	0					0	1					×1/2		×1	
0	1													×1/2	
1	0													×1/4	
0	0					1	0					×1/4		×1	
0	1													×1/2	
1	0													×1/4	
0	0			0	1	0	0				×1/2	×1		×1	
0	1													×1/2	
1	0													×1/4	
0	0					0	1					×1/2		×1	
0	1													×1/2	
1	0													×1/4	
0	0					1	0					×1/4		×1	
0	1													×1/2	
1	0													×1/4	
0	0			1	0	0	0				×1/4	×1		×1	
0	1													×1/2	
1	0													×1/4	
0	0					0	1					×1/2		×1	
0	1													×1/2	
1	0													×1/4	
0	0					1	0					×1/4		×1	
0	1													×1/2	
1	0													×1/4	

FR Register Values								PLL Circuit 1 Multiplication Factor	PLL Circuit 2 Multiplication Factor	Clock Ratio					Clock Input Frequency Range (MHz)
FR 7	FR 6	FR 5	FR 4	FR 3	FR 2	FR 1	FR 0			CKM	CKP	CKE	CKIO	CK	
0	0	0	1	0	0	0	0	×1	—	×1/2	×1	×1	—	×1	10–30
0	1													×1/2	
1	0													×1/4	
0	0					0	1					×1/2		×1	
0	1													×1/2	
1	0													×1/4	
0	0					1	0					×1/4		×1	
0	1													×1/2	
1	0													×1/4	
0	0			0	1	0	0				×1/2	×1		×1	
0	1													×1/2	
1	0													×1/4	
0	0					0	1					×1/2		×1	
0	1													×1/2	
1	0													×1/4	
0	0					1	0					×1/4		×1	
0	1													×1/2	
1	0													×1/4	
0	0			1	0	0	0				×1/4	×1		×1	
0	1													×1/2	
1	0													×1/4	
0	0					0	1					×1/2		×1	
0	1													×1/2	
1	0													×1/4	
0	0					1	0					×1/4		×1	
0	1													×1/2	
1	0													×1/4	

FR Register Values								PLL Circuit 1 Multiplication Factor	PLL Circuit 2 Multiplication Factor	Clock Ratio					Clock Input Frequency Range (MHz)
FR 7	FR 6	FR 5	FR 4	FR 3	FR 2	FR 1	FR 0			CKM	CKP	CKE	CKIO	CK	
0	0	1	0	0	0	0	0	×1	—	×1/4	×1	×1	—	×1	10–30
0	1													×1/2	
1	0													×1/4	
0	0					0	1					×1/2		×1	
0	1													×1/2	
1	0													×1/4	
0	0					1	0					×1/4		×1	
0	1													×1/2	
1	0													×1/4	
0	0			0	1	0	0				×1/2	×1		×1	
0	1													×1/2	
1	0													×1/4	
0	0					0	1					×1/2		×1	
0	1													×1/2	
1	0													×1/4	
0	0					1	0					×1/4		×1	
0	1													×1/2	
1	0													×1/4	
0	0			1	0	0	0				×1/4	×1		×1	
0	1													×1/2	
1	0													×1/4	
0	0					0	1					×1/2		×1	
0	1													×1/2	
1	0													×1/4	
0	0					1	0					×1/4		×1	
0	1													×1/2	
1	0													×1/4	

Notes: The lower and upper limits of the clock input frequency range are determined by the following conditions:

1. Lower-limit frequency

   The output frequency of each PLL before division must be at least 10 MHz. The specific clock input frequency lower limits are as follows: 2.5 MHz in clock modes 2, 3, 4, and 5 (as the PLL2 multiplication factor is ×4), 5 MHz in clock modes 0 and 1 (as the PLL2 multiplication factor is ×2), 5 MHz also in clock mode 6 (as the PLL1 multiplication factor is ×2), and 10 MHz in clock mode 7 (as the PLL1 multiplication factor is ×1).

2. Upper-limit frequency

   (1) Clock upper limits after division according to FRQCR register setting

   CKM ≤ 60 MHz, CKP ≤ 60 MHz, CKE ≤ 30 MHz

   (2) Clock upper limits after division according to MCLKCR1-5 register setting

   Mϕ ≤ 60 MHz, Pϕ ≤ 30 MHz

   The frequency that satisfies both (1) and (2) above is the clock input frequency upper limit.

4.4 Changing the Frequency

Changes in the master clock, peripheral clock, external bus clock, and clock output frequencies are controlled by software by means of the frequency control register.

The method of changing the frequencies is described below.

A frequency change is carried out by writing the required value in bits FR7 to FR0 in the FRQCR register. The write to FRQCR must be executed by a program in on-chip RAM or on-chip ROM. Also note that the DMAC must not be used to access FRQCR.

If the frequency ratio of Mϕ (the clock resulting from master clock (CKM) division) to CKE (the external bus clock) changes as a result of the frequency change, after the change FRQCR must be read before making an external CS space access. (The FRQCR value read at this time will be undefined.)

4.5 Output Clock Control

The CKIO and CK pins can be switched between clock output and the high-impedance state by means of the CKIOOE and CKOE bits in the FRQCR register. The initial values depend on the clock mode. Table 4.9 shows the correspondence between the clock mode, the state of the CKIO and CK pins, and the initial value of the CKIOOE and CKOE bits.

When the CKIOOE and CKOE bits are modified, the CKIO or CK output is changed immediately.

Table 4.9 Clock Modes, CKIO and CK Pin States, and Initial Value of CKIOOE and CKOE bits

Clock	Initial Pin State*		Initial Value		Bit Value Modification	
Mode	CKIO	CK	CKIOOE	CKOE	CKIOOE	CKOE
0	External clock output	External clock output	1	1	Not possible	Possible
1	External clock output	External clock output	1	1	Not possible	Possible
2	External clock output	External clock output	1	1	Not possible	Possible
3	External clock output	External clock output	1	1	Not possible	Possible
4	High impedance	External clock output	0	1	Possible	Possible
5	High impedance	External clock output	0	1	Possible	Possible
6	Clock input	External clock output	0	1	Not possible	Possible
7	Clock input	External clock output	0	1	Not possible	Possible

Note: * If hardware standby mode is entered after power is applied without executing a power-on reset, the pin states will be undefined. In this case, the $\overline{\text{RES}}$ pin must be driven low in hardware standby mode in order to fix the initial pin states according to the clock mode. When hardware standby mode is entered after a power-on reset, the prior pin states are retained.

4.6 Oscillator

There are two ways of supplying a clock: by connecting a crystal resonator and by inputting an external clock.

4.6.1 Connecting a Crystal Resonator

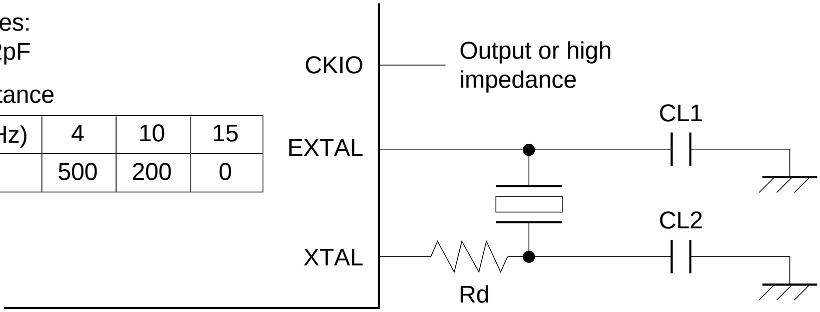
Figure 4.2 shows an example of crystal resonator connection. The values of damping resistance Rd and load capacitances CL1 and CL2 should be decided after investigating the components in collaboration with the manufacturer of the crystal resonator to be used. The crystal resonator should be an AT-cut parallel-resonance type. Place the crystal resonator and its load capacitors as close as possible to the XTAL and EXTAL pins.

Other signal lines should be routed away from the oscillator circuit to prevent induction from interfering with correct oscillation.

Reference Values:
CL1 = CL2 = 22pF

Damping Resistance

Frequency (MHz)	4	10	15
Rd (Ω)	500	200	0



Notes: 1. The CKIO pin is an output in clock modes 0, 1, 2, and 3, and is high-impedance in clock modes 4 and 5.

2. The values of CL1 and CL2 and damping resistance Rd should be decided after consultation with the manufacturer of the crystal resonator to be used.

Figure 4.2 Example of Crystal Resonator Connection

4.6.2 External Clock Input Methods

An external clock is input from the EXTAL pin or the CKIO pin, depending on the clock mode.

Clock Input from EXTAL Pin

This method can be used in clock modes 0, 1, 2, 3, 4, and 5.

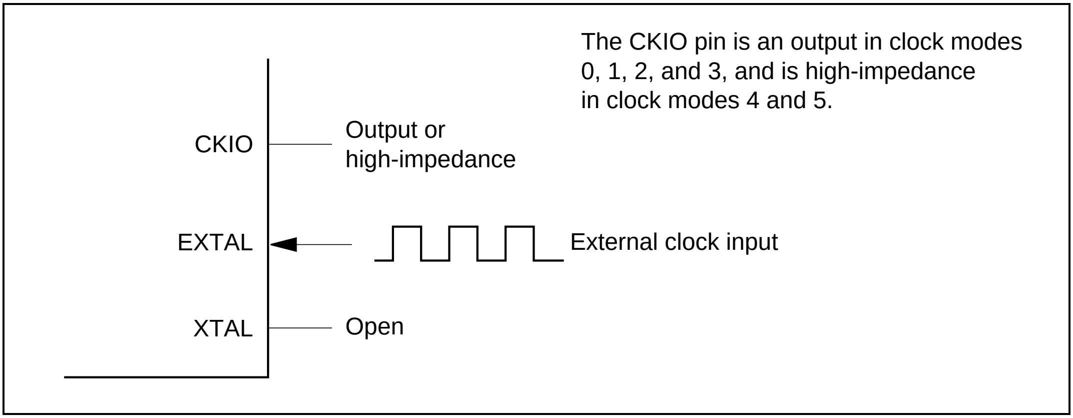


Figure 4.3 External Clock Input Method

Clock Input from CKIO Pin

This method can be used in clock modes 6 and 7.

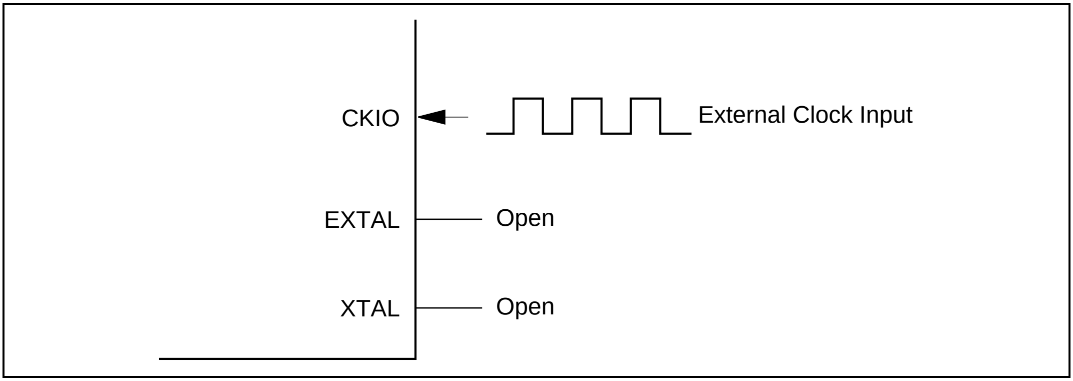
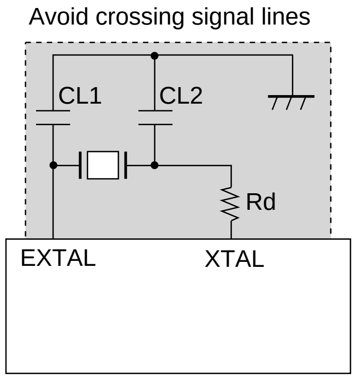


Figure 4.4 External Clock Input Method

4.6.3 Notes on Board Design

When Using a Crystal Resonator

Place the crystal resonator, capacitors CL1 and CL2, and damping resistance Rd as close as possible to the EXTAL and XTAL pins. To prevent induction from interfering with correct oscillation, use a common grounding point for the capacitors connected to the resonator, and do not locate a wiring pattern near these components.



Note: The values for CL1, CL2, and the damping resistance should be determined after consultation with the crystal resonator manufacturer.

Figure 4.5 Points for Attention when Using Crystal Resonator

Bypass Capacitors

As far as possible, insert a laminated ceramic capacitor of 0.01 to 0.1 µF as a bypass capacitor for each V_{SS}/V_{CC} pair. Mount the bypass capacitors as close as possible to the SH7065's power supply pins, and use components with a frequency characteristic suitable for the SH7065 operating frequency, as well as a suitable capacitance value.

Table 4.10 Bypass Capacitor Power Supply Pairs (Recommended)

PLLV$_{CC}$ Pair		**AV$_{CC}$ Pair**		**PV$_{CC}$ Pair**	
129(PLLV$_{CC}$)—132(PLLV$_{SS}$) (See figure 4.6)	◎	159(AV$_{CC}$)—148(AV$_{SS}$) (See figure 15.8)	◎	5(PV$_{CC}$)—1(PV$_{SS}$)	○
				173(PV$_{CC}$)—166(PV$_{SS}$)	○
V$_{CC}$ Pair					
26(V$_{CC}$)—31(V$_{SS}$)	◎	17(V$_{CC}$)—21(V$_{SS}$)	○	39(V$_{CC}$)—38(V$_{SS}$)	△
58(V$_{CC}$)—54(V$_{SS}$)	◎	70(V$_{CC}$)—64(V$_{SS}$)	○	48(V$_{CC}$)—45(V$_{SS}$)	△
79(V$_{CC}$)—76(V$_{SS}$)	◎	118(V$_{CC}$)—124(V$_{SS}$)	○	92(V$_{CC}$)—89(V$_{SS}$)	△
105(V$_{CC}$)—101(V$_{SS}$)	◎	—	—	140(V$_{CC}$)—135(V$_{SS}$)	△

Note: Priority order for inserting bypass capacitors:

◎: Must be inserted

○: Insert as far as possible

△: Insert if possible

When Using PLL Oscillator Circuits

Keep the wiring from the PLL V_{CC} and V_{SS} connection pattern to the power supply pins short, and make the pattern width large, to minimize the inductance component.

Ground the oscillation stabilization capacitors C1 and C2 to V_{SS} (PLL). Place C1 and C2 close to the CAP1 and CAP2 pins and do not locate a wiring pattern in the vicinity.

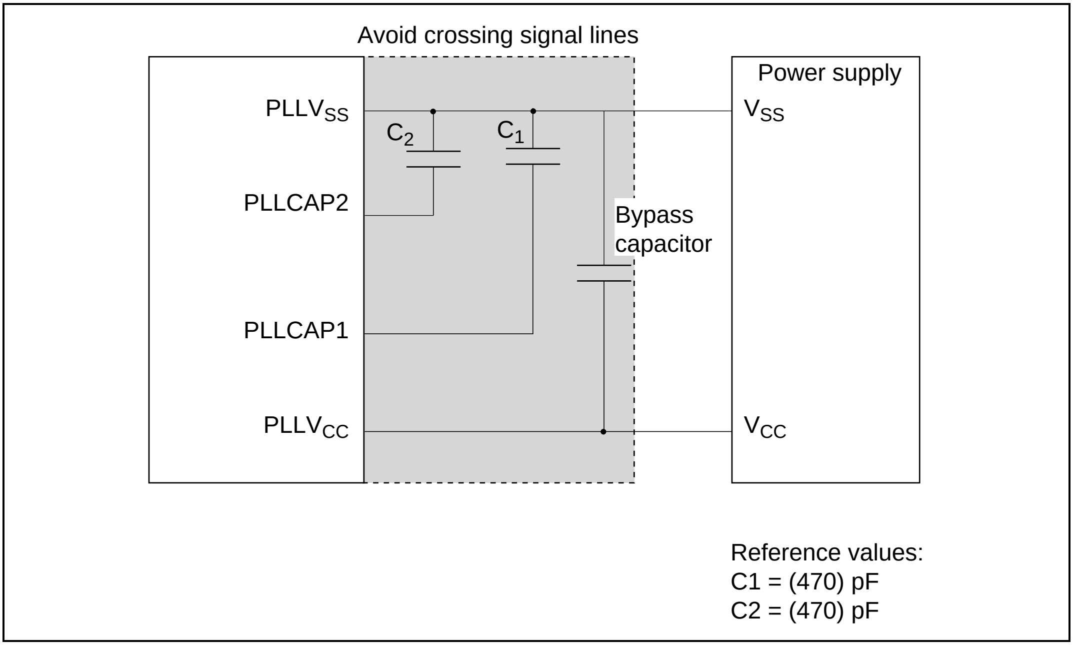


Figure 4.6 Points for Attention when Using PLL Oscillator Circuits

Table 4.11 Capacitance Values (For Reference)

Capacitance Value	Mode 0	Mode 1	Mode 2	Mode 3	Mode 4	Mode 5	Mode 6	Mode 7
C1 = 470 pF	Required	Required	Required	Required	Not required	Not required	Required	Required
C2 = 470 pF	Required	Required	Required	Required	Required	Required	Not required	Not required

4.7 Oscillation Stoppage Detection Function

This CPG is provided with a function that automatically places the timer pins in the high-impedance state when it detects clock stoppage to provide for cases where the oscillator halts due to a system error of some kind. If the CPG detects a change in EXTAL or CKIO due to an oscillator fault or stoppage of the external clock, it places the MMT (motor management timer) 6-phase output pins multiplexed with port E[*1] and the MMT 6-phase output pins multiplexed with port D[*2] in the high-impedance state.

Note that, in a software standby state transition, the pin states of the MMT 6-phase output pins multiplexed with port E[*1] and the MMT 6-phase output pins multiplexed with port D[*2] differ as shown below according to the setting of bit 6 (HIZ) in the standby control register (SBYCR).

1. MMT 6-phase output pins multiplexed with port E[*1]

   These pins go to the high-impedance state regardless of the setting of bit 6 in SBYCR and PFC settings.
2. MMT 6-phase output pins multiplexed with port D[*2]

   These pins go to the high-impedance state when a function other than the data bus function is selected by a PFC setting, and when the setting of bit 6 in SBYCR is 1 (HIZ setting). When the setting of bit 6 in SBYCR is 0, the previous pin states are retained. When the data bus function is selected, the pins always go to the high-impedance state.

While external clock oscillation is stopped, other chip operations are undefined. Also, when external clock oscillation is restarted after being stopped, chip operations, including those of the above 12 pins, are undefined. A power-on reset must therefore be executed when resuming chip operation.

Notes: 1. PE23/$\overline{\mathrm{IRQ7}}$/PWOB, PE22/$\overline{\mathrm{IRQ6}}$/PVOB, PE21/$\overline{\mathrm{IRQ5}}$/PUOB, PE19/$\overline{\mathrm{IRQ3}}$/PWOA, PE18/$\overline{\mathrm{IRQ2}}$/PVOA, PE17/$\overline{\mathrm{IRQ1}}$/PUOA/SCK0

2. PD26/D26/PWOB, PD25/D25/PVOB, PD24/D24/PUOB, PD22/D22/PWOA/SCK0, PD21/D21/PVOA/$\overline{\mathrm{IRQ7}}$, PD20/D20/PUOA/$\overline{\mathrm{IRQ6}}$

4.8 Power-Down Modes

4.8.1 States in Power-Down Modes

Table 4.12 shows the conditions for entering the power-down modes from the program execution state, the state of the CPU and peripheral modules in each mode, and the method of exiting each mode.

Table 4.12 State of CPU and Peripheral Modules in Power-Down Modes

		State							
Power-Down Mode	**Entering Conditions**	**CPG**	**CPU**	**CPU Registers**	**On-Chip Memory**	**On-chip Peripheral Modules**	**Pins**	**Refresh Operations**	**Exiting Conditions**
Sleep	SLEEP instruction executed while SBY bit is 0 in SBYCR	Operating	Halted	Held	Held	Operating	Operating	Refreshing	1. Interrupt 2. DMA address error 3. Power-on reset
Software standby	SLEEP instruction executed while SBY bit is 1 in SBYCR	Halted	Halted	Held	Held	Halted	Halted or high impedance	Self-refreshing	1. NMI interrupt 2. Power-on reset
Hardware standby	Low-level input to $\overline{\text{HSTBY}}$ pin	Halted	Halted	Undefined	Held	Halted	High impedance	Refreshing not possible	High-level input to $\overline{\text{HSTBY}}$ pin during low-level input to $\overline{\text{RES}}$ pin
Module standby function	Setting MSTP bit to 1 in MSTPCR	Operating	Operating	Held	Held	Specified modules halted*	Held or initialized	Refreshing	1. Clearing MSTP bit to 0 2. Power-on reset
Module clock division	Setting MCLK bit to 1 in MCLKCR	Clock to module corresponding to MCLK bit is further divided from master clock (CKM) or peripheral clock (CKP) set in CPG before being supplied							1. Setting MCLK bit to initial value 2. Power-on reset

Note: * See section 4.9.2, Module Stop Control Registers 1 and 2 (MSTPCR1, MSTPCR2).

Register Configuration

Table 4.13 shows the registers used for power-down mode control.

Table 4.13 Power-Down Mode Registers

Name	Abbreviation	R/W	Initial Value	Address	Access Size
Standby control register	SBYCR	R/W	H'1F	H'FFFF 1004	8, 16, 32
Module stop control register 1	MSTPCR1	R/W	H'0000	H'FFFF 1030	8, 16, 32
Module stop control register 2	MSTPCR2	R/W	H'0000	H'FFFF 1032	8, 16, 32
Module clock control register 1	MCLKCR1	R/W	H'8888 (clock modes 1, 3, 5, 6) H'FFFF (clock modes 0, 2, 4, 7)	H'FFFF 1034	8, 16, 32
Module clock control register 2	MCLKCR2	R/W	H'8888 (clock modes 1, 3, 5, 6) H'FFFF (clock modes 0, 2, 4, 7)	H'FFFF 1036	8, 16, 32
Module clock control register 3	MCLKCR3	R/W	H'8888 (clock modes 1, 3, 5, 6) H'FFFF (clock modes 0, 2, 4, 7)	H'FFFF 1038	8, 16, 32
Module clock control register 4	MCLKCR4	R/W	H'8888 (clock modes 1, 3, 5, 6) H'FFFF (clock modes 0, 2, 4, 7)	H'FFFF 103A	8, 16, 32
Module clock control register 5	MCLKCR5	R/W	H'CCCC (clock modes 1, 3, 5, 6) H'FFFF (clock modes 0, 2, 4, 7)	H'FFFF 103C	8, 16, 32

4.8.2 Pin Configuration

Table 4.14 shows the pin used for power-down mode control.

Table 4.14 Power-Down Mode Pin

Pin Name	Abbreviation	I/O	Function
Hardware standby pin	$\overline{\text{HSTBY}}$	Input	Low-level input to this pin places the chip in the hardware standby state.

4.9 Register Descriptions

4.9.1 Standby Control Register (SBYCR)

The standby control register (SBYCR) is an 8-bit readable/writable register that specifies the power-down mode status.

SBYCR is initialized to H'1F by a power-on reset, but is not initialized in software standby mode.

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

Bit 7—Software Standby (SBY): Specifies a transition to software standby mode. The SBY bit cannot be set to 1 while the watchdog timer (WDT) is operating (while the timer enable bit (TME) is set to 1 in the watchdog timer's timer control/status register (TCSR)). When making a transition to software standby mode, the watchdog timer must be stopped by clearing the TME bit to 0 before the SBY bit is set.

Bit 7: SBY	Description	
0	Transition to sleep mode on execution of SLEEP instruction	(Initial value)
1	Transition to software standby mode on execution of SLEEP instruction	

Bit 6—Port High Impedance (HIZ): Selects whether specific output pins retain their state or become high-impedance in software standby mode. See appendix B, Pin States, for the pins that are controlled. The HIZ bit cannot be set to 1 when the TME bit is set to 1 in the watchdog timer's TCSR register. To set output pins to the high-impedance state, the TME bit must be cleared to 0 before the HIZ bit is set.

Bit 6: HIZ	Description	
0	Pin state retained in software standby mode	(Initial value)
1	Pins go to high-impedance state in software standby mode	

Bit 5—Reserved: This bit is always read as 0 and should only be written with 0.

Bits 4 to 0—Reserved: These bits are always read as 1 and should only be written with 1.

4.9.2 Module Stop Control Registers 1 and 2 (MSTPCR1, MSTPCR2)

Module stop control registers 1 and 2 (MSTPCR1, MSTPCR2) are 16-bit readable/writable registers that specify the module stop mode status.

MSTPCR1 and MSTPCR2 are initialized to H'0000 by a power-on reset, but are not initialized in software standby mode.

MSTPCR1

Bit:	15	14	13	12	11	10	9	8
	MSTP15	MSTP14	MSTP13	MSTP12	MSTP11	MSTP10	MSTP9	MSTP8
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
	MSTP7	MSTP6	MSTP5	MSTP4	MSTP3	MSTP2	MSTP1	MSTP0
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

MSTPCR2

Bit:	15	14	13	12	11	10	9	8
	MSTP31	MSTP30	MSTP29	MSTP28	MSTP27	MSTP26	MSTP25	MSTP24
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
	MSTP23	MSTP22	MSTP21	MSTP20	MSTP19	MSTP18	MSTP17	MSTP16
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bits 15 to 0—Module Stop 31 to 0 (MSTP31 to MSTP0): These bits specify stoppage of the clock supply to the corresponding modules. See table 4.16 for the correspondence between the register bits and modules.

Bits 15 to 0: MSTP31 to MSTP0	Description	
0	Clock is supplied to corresponding module	(Initial value)
1	Clock supply to corresponding module is stopped	

4.9.3 Module Clock Control Registers 1 to 5 (MCLKCR1 to MCLKCR5)

Module clock control registers 1 to 5 (MCLKCR1 to MCLKCR5) are 16-bit readable/writable registers that specify the division ratio for the clocks supplied to the modules.

Registers MCLKCR1 to MCLKCR5 are initialized to a value determined by the clock mode in a power-on reset, but are not initialized in software standby mode.

MCLKCR1

Bit:	15	14	13	12	11	10	9	8
	—	MCLK 032	MCLK 031	MCLK 030	—	MCLK 022	MCLK 021	MCLK 020
Initial value:	1	—	—	—	1	—	—	—
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
	—	MCLK 012	MCLK 011	MCLK 010	—	MCLK 002	MCLK 001	MCLK 000
Initial value:	1	—	—	—	1	—	—	—
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W

MCLKCR2

Bit:	15	14	13	12	11	10	9	8
	—	MCLK 072	MCLK 071	MCLK 070	—	MCLK 062	MCLK 061	MCLK 060
Initial value:	1	—	—	—	1	—	—	—
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
	—	MCLK 052	MCLK 051	MCLK 050	—	MCLK 042	MCLK 041	MCLK 040
Initial value:	1	—	—	—	1	—	—	—
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W

MCLKCR3

Bit:	15	14	13	12	11	10	9	8
	—	MCLK 112	MCLK 111	MCLK 110	—	MCLK 102	MCLK 101	MCLK 100
Initial value:	1	—	—	—	1	—	—	—
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
	—	MCLK 092	MCLK 091	MCLK 090	—	MCLK 082	MCLK 081	MCLK 080
Initial value:	1	—	—	—	1	—	—	—
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W

MCLKCR4

Bit:	15	14	13	12	11	10	9	8
	—	MCLK 152	MCLK 151	MCLK 150	—	MCLK 142	MCLK 141	MCLK 140
Initial value:	1	—	—	—	1	—	—	—
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
	—	MCLK 132	MCLK 131	MCLK 130	—	MCLK 122	MCLK 121	MCLK 120
Initial value:	1	—	—	—	1	—	—	—
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W

MCLKCR5

Bit:	15	14	13	12	11	10	9	8
	—	—	MCLK 191	MCLK 190	—	—	MCLK 181	MCLK 180
Initial value:	1	1	—	—	1	1	—	—
R/W:	R	R	R/W	R/W	R	R	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
	—	—	MCLK 171	MCLK 170	—	—	MCLK 161	MCLK 160
Initial value:	1	1	—	—	1	1	—	—
R/W:	R	R	R/W	R/W	R	R	R/W	R/W

Bits 15, 11, 7, and 3—Reserved: These bits are always read as 1 and should only be written with 1.

Bits 14, 10, 6, and 2—Reserved (MCLKCK5 only): These bits are always read as 1 and should only be written with 1.

Other Bits—Module Clock 191 to 000 (MCLK191 to MCLK000): These bits specify the clock division ratio for the corresponding modules. A clock further divided from the master clock (CKM) or peripheral clock (CKP) set in the frequency control register (FRQCR) of the clock pulse generator (CPG) is supplied to the corresponding modules. The initial values depend on the clock mode. See table 4.18 for the correspondence between the register bits and modules.

- MCLK191 to MCLK160

Bit nn1: MCLKnn1	Bit nn0: MCLKnn0	Description
0	0	Clock supplied to module is not divided (Initial value in clock modes 1, 3, 5, 6)
	1	Reserved (Do not set)
1	0	Clock supplied to module is further divided by 8
	1	Clock supplied to module is further divided by 64 (Initial value in clock modes 0, 2, 4, 7)

- MCLK152 to MCLK000

Bit nn2: MCLKnn2	Bit nn1: MCLKnn1	Bit nn0: MCLKnn0	Description
0	0	0	Clock supplied to module is not divided (Initial value in clock modes 1, 3, 5, 6)
		1	Clock supplied to module is further divided by 2
	1	0	Clock supplied to module is further divided by 3
		1	Clock supplied to module is further divided by 5
1	0	0	Reserved (Do not set)
		1	Reserved (Do not set)
	1	0	Clock supplied to module is further divided by 8
		1	Clock supplied to module is further divided by 64 (Initial value in clock modes 0, 2, 4, 7)

4.10 Sleep Mode

4.10.1 Transition to Sleep Mode

If a SLEEP instruction is executed when the SBY bit in SBYCR 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 on-chip peripheral modules continue to operate, and clocks continue to be output from the CKIO and CK pins. In sleep mode, external bus release requests are not accepted.

The CPU regards the SBYCR write as being executed in one cycle, and performs the next processing. However, the write actually takes the number of cycles shown in table 8.12 in section 8, Bus State Controller (BSC). To ensure that the value written from the CPU to SBYCR is reliably reflected in the SLEEP instruction, either read SBYCR or else wait for the number of cycles shown in table 8.12, before executing the SLEEP instruction.

4.10.2 Exit from Sleep Mode

Sleep mode is exited by means of an interrupt (NMI, IRQ, IRL, or on-chip peripheral module), a DMAC address error, a power-on reset, or the $\overline{\text{HSTBY}}$ pin.

Exit by Interrupt: When an NMI, IRQ, IRL, or on-chip peripheral module interrupt is generated, sleep mode is exited and interrupt exception handling is executed. The interrupt request is not accepted and sleep mode is not exited when the priority level of the generated interrupt is not higher than the interrupt mask level set in the CPU's status register (SR), or when an interrupt from an on-chip peripheral module is disabled on the module side.

Exit by DMAC Address Error: When a DMAC address error occurs, sleep mode is exited and DMAC address error exception handling is executed.

Exit by Power-On Reset: When the $\overline{\text{RES}}$ pin is driven low, the SH7065 enters the power-on reset state and exits sleep mode.

Exit by $\overline{\text{HSTBY}}$ Pin: When the $\overline{\text{HSTBY}}$ pin is driven low, the SH7065 enters the hardware standby mode state and exits sleep mode.

4.11 Software Standby Mode

4.11.1 Transition to Software Standby Mode

If a SLEEP instruction is executed when the SBY bit in SBYCR is set to 1, the chip switches from the program execution state to software standby mode. In software standby mode, the clock and on-chip peripheral modules halt as well as the CPU, reducing power consumption to an extremely low level. Clock output from the CKIO and CK pins is also stopped.

CPU register contents and data in on-chip RAM are retained. Some on-chip peripheral module registers are initialized. The state of the peripheral module registers in software standby mode is shown in table 4.15. See appendix B, Pin States, for the pin states.

The CPU regards the SBYCR write as being executed in one cycle, and performs the next processing. However, the write actually takes the number of cycles shown in table 8.12 in section 8, Bus State Controller (BSC). To ensure that the value written from the CPU to SBYCR is reliably reflected in the SLEEP instruction, either read SBYCR or else wait for the number of cycles shown in table 8.12, before executing the SLEEP instruction.

In the software standby state, external bus address/data/bus control signals (except DRAM signals) go to the high-impedance state, i.e. the bus-released state. In the software standby state, the $\overline{\text{BREQ}}$ bus release request input signal is ignored. Note that the following two cases apply to the $\overline{\text{BACK}}$ bus use enable output signal.

1. Transition from bus-released state ($\overline{\text{BREQ}}$ input asserted low) to software standby state
   When the bus release request signal ($\overline{\text{BREQ}}$) is asserted low in the normal state, the $\overline{\text{BACK}}$ pin is set to low output, indicating that the bus has been released. If the software standby state is entered in this state, $\overline{\text{BACK}}$ output goes high, but other address, data, and bus control signals remain in the high-impedance state, i.e. the bus-released state. If the software standby state is

exited while $\overline{\text{BREQ}}$ input is still asserted, $\overline{\text{BACK}}$ output goes low and the bus-released state is maintained. If software standby is exited while $\overline{\text{BREQ}}$ input is negated, $\overline{\text{BACK}}$ output goes high and the chip returns to the normal state (in which the bus is not released).

2. Transition from normal state ($\overline{\text{BREQ}}$ input negated high) to software standby state

   When a transition is made from the normal state to the software standby state, $\overline{\text{BACK}}$ output goes to the Z (high-impedance) state, and the external bus goes to the high-impedance state, i.e. the bus-released state. If this state is exited while $\overline{\text{BREQ}}$ input is negated, $\overline{\text{BACK}}$ output returns to the high level. If $\overline{\text{BREQ}}$ input is in the asserted state when software standby is exited, $\overline{\text{BACK}}$ is output high for 1.5 external clock (CKE) cycles, and then returns to the low level, i.e. the bus-released state.

Table 4.15 State of Registers in Software Standby Mode

Module	Initialized Registers	Registers Retaining Contents
Interrupt controller (INTC)	—	All registers
User break controller (UBC)	—	All registers
Bus state controller (BSC)	—	All registers
Clock pulse generator (CPG)	—	All registers
Direct memory access controller (DMAC)	All registers	—
Timer pulse unit (TPU)	All registers	—
Motor management timer (MMT)	All registers	—
Watchdog timer (WDT)	• OVF, WT/IT, and TME bits in TCSR register • RSTCSR register	• Bits CKS2 to CKS0 in TCSR register • TCNT registers
Serial communication interface (SCI)	All registers	—
A/D converter (A/D)	All registers	—
D/A converter (D/A)	All registers	—
Compare match timer (CMT)	All registers	—
Pin function controller (PFC)	—	All registers
I/O ports (I/O)	—	All registers
Power-down mode related modules	—	All registers

4.11.2 Exit from Software Standby Mode

Software standby mode is exited by means of a power-on reset or the $\overline{\text{HSTBY}}$ pin.

Exit by NMI Interrupt: When a falling edge or rising edge (as selected with the NMI edge select bit (NMIE) in interrupt control register 1 (ICR1) of the interrupt controller (INTC)) is detected in the NMI signal, clock oscillation is started. This clock is supplied only to the watchdog timer (WDT). When the time set in the clock select bits (CKS2 to CKS0) in the WDT's timer control/status register (TCSR) before entering software standby mode has elapsed, WDT overflow occurs. This overflow is taken as an indication that the clock has settled, and the clock is then supplied to the entire chip. Software standby mode is thus exited and NMI exception handling is begun.

When exiting software standby mode by means of an NMI interrupt, set bit CKS2 to CKS0 so that the WDT overflow period is at least as long as the oscillation settling time.

When exiting software standby mode with the NMI pin designated for falling edge detection, ensure that the NMI pin level goes high when software standby mode is entered (when the clock is stopped) and low when recovering from software standby mode (when the clock is restarted after the oscillation settling time). When exiting software standby mode with the NMI pin designated for rising edge detection, ensure that the NMI pin level goes low when software standby mode is entered (when the clock is stopped) and high when recovering from software standby mode (when the clock is restarted after the oscillation settling time).

Exit by Power-On Reset: When the $\overline{\text{RES}}$ pin is driven low, the SH7065 enters the power-on reset state and exits software standby mode. The $\overline{\text{RES}}$ pin must be held low until clock oscillation settles.

Exit by $\overline{\textbf{HSTBY}}$ Pin: When the $\overline{\text{HSTBY}}$ pin is driven low, the SH7065 enters the hardware standby mode state and exits software standby mode.

4.11.3 Software Standby Mode Application Example

In the following example, software standby mode is entered at a falling edge on the NMI pin, and exited at a rising edge. The timing for this example is shown in figure 4.7.

When the NMI pin level changes from high to low while the NMI edge select bit (NMIE) is cleared to 0 (falling edge detection) in the interrupt control register 1 (ICR1), an NMI interrupt is accepted. When the NMIE bit is set to 1 (rising edge detection), the standby bit (SBY) in the standby control register is set to 1, and a SLEEP instruction is executed in the NMI exception service routine, a transition is made to standby mode. When the NMI pin level is subsequently changed from low to high, software standby mode is exited.

After the NMI pin level is changed to high, keep the high level until the NMI exception handling starts.

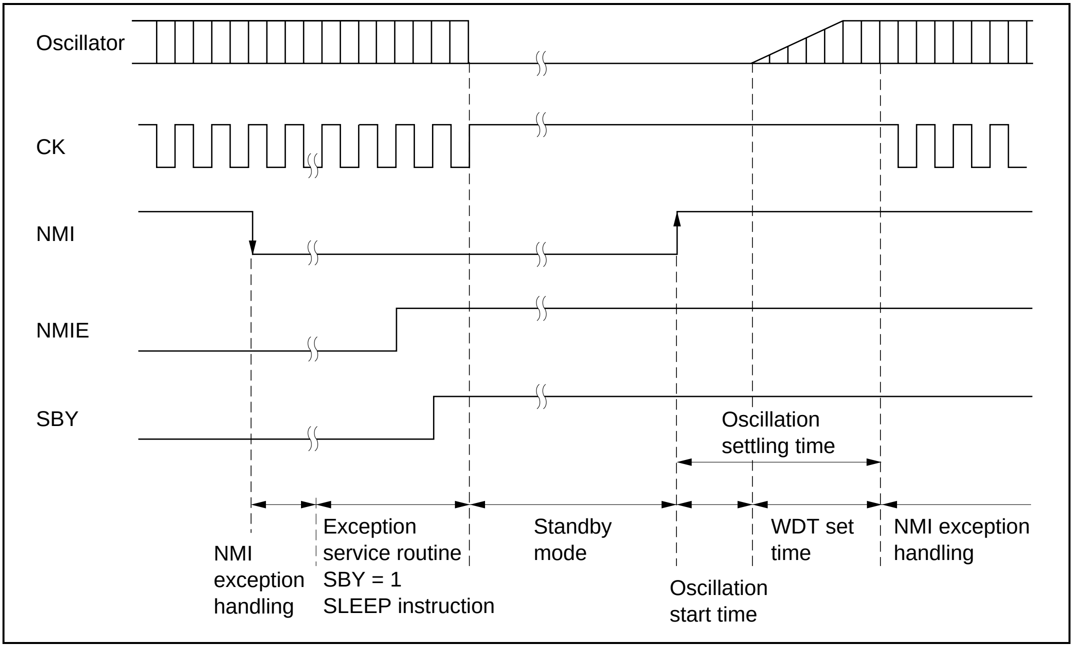


Figure 4.7 NMI Timing in Standby Mode (Application Example)

4.12 Hardware Standby Mode

4.12.1 Transition to Hardware Standby Mode

Regardless of its current state, the chip enters hardware standby mode whenever the $\overline{\text{HSTBY}}$ pin is driven low.

Hardware standby mode reduces power consumption drastically by resetting and halting all functions. As long as the specified voltage is supplied, on-chip RAM data is retained. However, on-chip RAM contents may be lost if an access to on-chip RAM has been initiated when the hardware standby state is entered. To retain RAM contents, the clock supply to RAM should be halted with the module standby function before entering the hardware standby state. I/O ports are placed in the high-impedance state.

The level of the mode pins (MD5 to MD0) should not be changed during hardware standby mode.

4.12.2 Exit from Hardware Standby Mode

Hardware standby mode is exited by means of the $\overline{\text{HSTBY}}$ and $\overline{\text{RES}}$ pins.

When $\overline{\text{HSTBY}}$ is driven high while $\overline{\text{RES}}$ is low, the power-on reset state is entered and hardware standby mode is exited. The $\overline{\text{RES}}$ pin must be held low until clock oscillation settles.

4.12.3 Hardware Standby Mode Timing

Figure 4.8 shows the timing relationships for hardware standby mode.

To enter hardware standby mode, first drive $\overline{\text{RES}}$ low, then drive $\overline{\text{HSTBY}}$ low. To exit hardware standby mode, first drive $\overline{\text{HSTBY}}$ high, wait for the clock to settle, then bring $\overline{\text{RES}}$ from low to high.

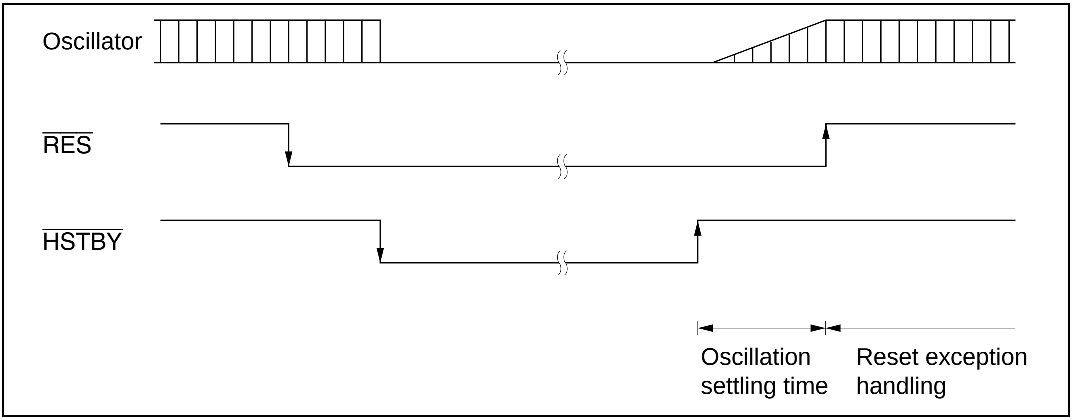


Figure 4.8 Hardware Standby Mode Timing

4.13 Module Standby Function

4.13.1 Transition to Module Standby Function

Setting an MSTP bit to 1 in module stop mode control register 1 or 2 (MSTPCR1, MSTPCR2) enables the clock supply to the corresponding on-chip peripheral module to be halted. Use of this function allows power consumption to be reduced in normal operation and in sleep mode.

The correspondence between the MSTP bits and on-chip peripheral modules is shown in table 4.16.

In the module standby state, the SCI and A/D registers are initialized. Other registers retain their states prior to halting of the module.

Registers of modules set to the module standby state cannot be read or written to.

Table 4.16 MSTP Bits and Corresponding On-Chip Peripheral Modules

Bit*	Description
MSTP31	X-RAM and Y-RAM
MSTP30	On-chip ROM
MSTP29	—
MSTP28	User break controller (UBC)
MSTP27	Direct memory access controller (DMAC)
MSTP26	—
MSTP25	—
MSTP24	—
MSTP23	—
MSTP22	—
MSTP21	—
MSTP20	—
MSTP19	—
MSTP18	—
MSTP17	—
MSTP16	—
MSTP15	Serial communication interface (SCI) channel 0
MSTP14	Serial communication interface (SCI) channel 1
MSTP13	Serial communication interface (SCI) channel 2
MSTP12	—
MSTP11	Compare match timer (CMT)
MSTP10	—
MSTP9	Motor management timer (MMT)
MSTP8	Port output enable (POE)
MSTP7	Timer pulse unit (TPU)
MSTP6	A/D converter (A/D)
MSTP5	D/A converter (D/A)
MSTP4	—
MSTP3	—
MSTP2	—
MSTP1	—
MSTP0	—

Note: * Bits to which an on-chip peripheral module is not assigned must be written with 0.

4.13.2 Exit from Module Standby Function

The module standby function is exited by clearing the MSTP bits to 0, or by a power-on reset.

When the X-RAM/Y-RAM or on-chip ROM module standby function is exited by modification of the module stop control register (MSTPCR2), following the register modification at least one MSTPCR2 register read must be performed before the above memory is accessed.

4.14 Module Clock Division Function

4.14.1 Clock Definitions

Definitions of the clocks used by the SH7065 are given in tables 4.17 and 4.18, and figure 4.9.

Table 4.17 Definitions of Internal Clocks

Abbreviation	Name
CKM	Master clock
CKP	Peripheral clock
CKE	External bus clock

Table 4.18 Definitions of Divided Clocks

Abbreviation	Name
Mϕ	Clock supplied to modules after division of master clock (CKM)
Pϕ	Clock supplied to modules after division of peripheral clock (CKP)

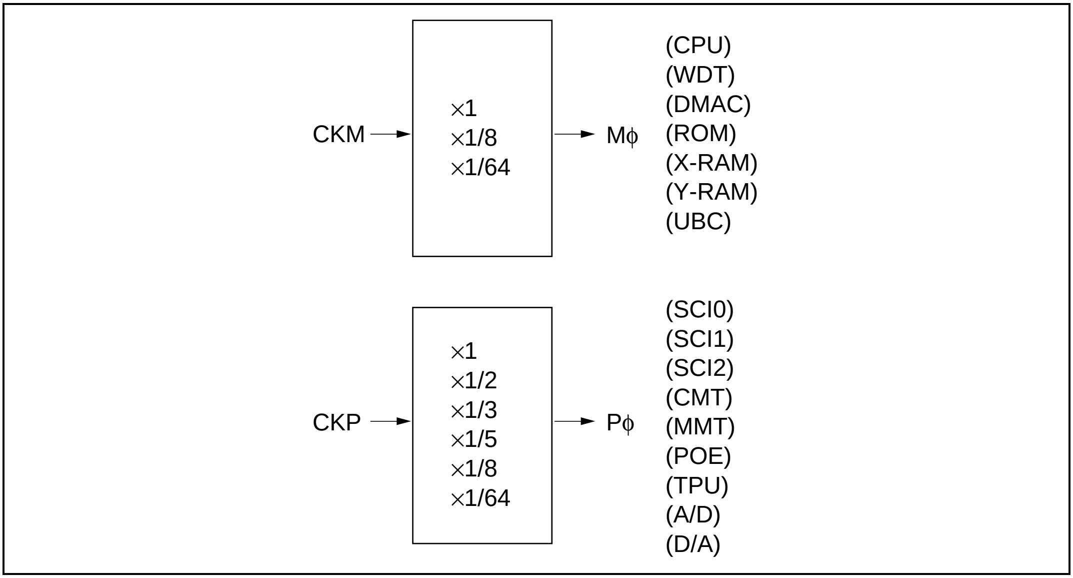


Figure 4.9 Divided Clocks and Corresponding Modules

4.14.2 Transition to Module Clock Division Function

Setting MCLK bits in module clock control registers 1 to 5 (MCLKCR1 to MCLKCR5) supplies a clock obtained by further division of the master clock (CKM) or peripheral clock (CKP) set in the frequency control register (FRQCR) of the clock pulse generator (CPG) to the corresponding module. Use of this function allows power consumption to be reduced during normal operation.

The correspondence between the MCLK bits and on-chip peripheral modules is shown in table 4.19.

Table 4.19 MCLK Bits and Corresponding On-Chip Peripheral Modules

Bit[*1]	Description	Maximum Operating Frequency
MCLK191–190	CPU[*2]	60 MHz
MCLK181–180	—	—
MCLK171–170	—	—
MCLK161–160	—	—
MCLK152–150	Serial communication interface (SCI) channel 0	30 MHz
MCLK142–140	Serial communication interface (SCI) channel 1	30 MHz
MCLK132–130	Serial communication interface (SCI) channel 2	30 MHz
MCLK122–120	—	—
MCLK112–110	Compare match timer (CMT)	30 MHz
MCLK102–100	—	—
MCLK092–090	Motor management timer (MMT)	30 MHz
MCLK082–080	Port output enable (POE)	30 MHz
MCLK072–070	Timer pulse unit (TPU)	30 MHz
MCLK062–060	A/D converter (A/D)	20 MHz (clock select CKS = 1) 30 MHz (clock select CKS = 0)
MCLK052–050	D/A converter (D/A)	30 MHz
MCLK042–040	—	—
MCLK032–030	—	—
MCLK022–020	—	—
MCLK012–010	—	—
MCLK002–000	—	—

Notes: 1. Bits to which a module is not assigned must be written with their initial value.

2. Including the DMAC, ROM, X-RAM, Y-RAM, UBC, and WDT.

4.14.3 Exit from Module Clock Division Function

The module clock division function is exited by setting the MCLK bits.

4.14.4 Notes on Use of Module Clock Division Function

1. The module clock division ratio is changed by writing the required value in the MCLK bits in the MCLKCR register.

   The write to MCLKCR must be executed by a program in on-chip RAM or on-chip ROM. Also note that the DMAC must not be used to access MCLKCR.

   If the frequency ratio of Mϕ (the clock resulting from master clock (CKM) division) to CKE (the external bus clock) changes as a result of the frequency change, after the change the MCLKCR5 register must be read before
   - an external space access, or
   - a transition to sleep mode.

   (The MCLKCR5 register value read at this time will be undefined.)

   When changing Pϕ (the clock resulting from peripheral clock (CKP) division), after the change a register in the module corresponding to the changed Pϕ must be read before
   - accessing a register in the module corresponding to the changed Pϕ,
   - entering the module standby state for the module corresponding to the changed Pϕ,
   - changing the changed Pϕ again, or
   - entering software standby mode.

   (The register value read at this time will be undefined.)

2. Ensure that CKM, CKP, CKE, and Mϕ and Pϕ supplied to the modules, do not exceed their maximum frequency while the setting is being made.

3. Immediately after the value of the MCLK bits is changed, the module corresponding to the changed Mϕ or Pϕ will temporarily enter the module standby state. Therefore, when an MCLK bit value corresponding to the SCI or A/D converter is changed, the SCI or A/D converter registers are initialized. However, there is no temporary transition to the module standby state if the same value is written to the MCLK bits.

4. Do not set a combination that gives a division ratio of Mϕ:CKE = 1/8:1/4 (taking the clock input to dividers 1 to 4 in the CPG as 1); that is, a combination giving a CPG division setting of CKM:CKE = 1:1/4, and a module clock division setting of CKM:Mϕ = 1:1/8.

4.15 Note on Initialization

To reduce current dissipation, the following instructions should be executed in application software initialization.

```
PCLR A0       ; Clear A0 register to 0
PSHA #5, A0   ; Left-shift 5 bits
```

As there are nodes that are not initialized by a power-on reset after powering on, current dissipation may increase by approximately 30 mA if the above instructions are not executed.

Section 5 Exception Handling

5.1 Overview

5.1.1 Exception Handling Types and Priority

As table 5.1 indicates, exception handling may be caused by a reset, address error, interrupt, or instruction. Exception handling is prioritized as shown in table 5.1. If two or more exceptions occur simultaneously, they are accepted and processed in order of priority.

Table 5.1 Exception Types and Priority

	Exception Handling		Priority
Reset	Power-on reset		High
Address errors	CPU address error		
	DMAC address error		
Interrupts	NMI		
	User break		
	External interrupt (IRQ/IRL)		
	On-chip peripheral modules	Direct memory access controller (DMAC)	
		Bus state controller (BSC)	
		Watchdog timer (WDT)	
		Timer pulse unit (TPU)	
		Serial communication interface (SCI)	
		Compare match timer (CMT)	
		A/D converter (A/D)	
		Motor management timer (MMT)	
Instructions	Trap instruction (TRAPA instruction)		
	General illegal instruction (undefined code)		
	Slot illegal instruction (undefined code or instruction that modifies PC*1 located immediately after delayed branch instruction*2)		Low

Notes: 1. Instructions that modify PC: JMP, JSR, BRA, BSR, RTS, RTE, BT, BF, TRAPA, BF/S, BT/S, BSRF, BRAF

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

5.1.2 Timing of Exception Source Detection and Start of Exception Handling

Table 5.2 shows the timing of detection and the start of exception handling for each exception source.

Table 5.2 Exception Source Detection and Exception Handling Start Timing

Exception Type		Source Detection and Start of Exception Handling
Reset	Power-on reset	Started immediately after low-to-high transition at $\overline{\text{RES}}$ pin
Address error		Detected when instruction is decoded; exception handling is started after completion of currently executing instruction
Interrupt		
Instruction	Trap instruction	Started by execution of TRAPA instruction
	General illegal instruction	Started when undefined code not in a delayed branch instruction (delay slot) is decoded
	Slot illegal instruction	Started when undefined code or instruction that modifies PC located in a delayed branch instruction (delay slot) is decoded

When exception handling is initiated, the CPU operates as follows.

Power-On Reset Exception Handling: The initial values of the program counter (PC) and stack pointer (SP) are fetched from exception vector table addresses H'00000000 and H'00000004, respectively. See section 5.1.3, Exception Vector Table, for details of the exception vector table. Next, the vector base register (VBR) is cleared to 0 and the interrupt mask bits (I3 to I0) in the status register (SR) are set to 1111. Program execution starts from the PC address fetched from the exception vector table.

Address Error, Interrupt, or Instruction Exception Handling: SR and PC are saved on the stack indicated by R15. In the case of interrupt exception handling, the interrupt priority level is written to the interrupt mask bits (I3 to I0) in SR. In address error and instruction exception handling, bits I3 to I0 are not affected. Next, the start address is fetched from the exception vector table and program execution starts from that address.

5.1.3 Exception Vector Table

Before exception handling is executed, the exception vector table must have been set up in memory. The exception vector table holds the start addresses of the exception service routines (the reset exception handling table holds the initial values of PC and SP.

A different vector number and vector table address offset are assigned to each exception source. The vector table address is calculated from the corresponding vector number and vector table address offset. In exception handling, the start address of the exception service routine is fetched from the exception vector table entry indicated by this vector table address.

The vector numbers and vector table address offsets are shown in table 5.3, and the method of calculating the vector table address in table 5.4.

Table 5.3 Exception Vector Table

Exception Source		Vector Number	Vector Table Address Offset
Power-on reset	PC	0	H'00000000–H'00000003
	SP	1	H'00000004–H'00000007
(Reserved for system)	PC	2	H'00000008–H'0000000B
	SP	3	H'0000000C–H'0000000F
General illegal instruction		4	H'00000010–H'00000013
(Reserved for system)		5	H'00000014–H'00000017
Slot illegal instruction		6	H'00000018–H'0000001B
(Reserved for system)		7	H'0000001C–H'0000001F
		8	H'00000020–H'00000023
CPU address error		9	H'00000024–H'00000027
DMAC address error		10	H'00000028–H'0000002B
Interrupt	(Reserved for system)	11	H'0000002C–H'0000002F
	NMI	12	H'00000030–H'00000033
	User break	13	H'00000034–H'00000037
(Reserved for system)		14 to 31	H'00000038–H'0000003B to H'0000007C–H'0000007F
Trap instruction (user vector)		32 to 63	H'00000080–H'00000083 to H'000000FC–H'000000FF

Exception Source		Vector Number	Vector Table Address Offset
Interrupt	IRQ0	64	H'00000100–H'00000103
	IRQ1, IRL1	65	H'00000104–H'00000107
	IRQ2, IRL2	66	H'00000108–H'0000010B
	IRQ3, IRL3	67	H'0000010C–H'0000010F
	IRQ4	80	H'00000140–H'00000143
	IRQ5	81	H'00000144–H'00000147
	IRQ6	82	H'00000148–H'0000014B
	IRQ7	83	H'0000014C–H'0000014F
	(Reserved for system)	84	H'00000150–H'00000153
		85	H'00000154–H'00000157
		86	H'00000158–H'0000015B
		87	H'0000015C–H'0000015F
		88	H'00000160–H'00000163
		89	H'00000164–H'00000167
		90	H'00000168–H'0000016B
		91	H'0000016C–H'0000016F
	IRL4 to IRL15[*1]	68 to 79	H'00000110–H'00000113 to H'0000013C–H'0000013F
(Reserved for system)		92 to 127	H'00000170–H'00000173 to H'000001FC–H'000001FF
On-chip peripheral module[*2]		128 to 255	H'00000200–H'00000203 to H'000003FC–H'000003FF

Notes: 1. For the vector numbers and vector table offsets of external interrupts IRL4 and IRL5, see table 6.3, IRL Interrupt Priority Levels and Vector Numbers, in section 6, Interrupt Controller (INTC).

2. For the vector numbers and vector table offsets of on-chip peripheral module interrupts, see table 6.6, On-Chip Peripheral Module Interrupt Exception Vectors and Priority Order, in section 6, Interrupt Controller (INTC).

Table 5.4 Exception Vector Table Address Calculation

Exception Source	Vector Table Address Calculation
Reset	Vector table address = (vector table address offset) = (vector number) × 4
Address error, interrupt, instruction	Vector table address = VBR + (vector table address offset) = VBR + (vector number) × 4

Notes: VBR: Vector base register

Vector table address offset: See table 5.3.

Vector number: See table 5.3.

5.2 Power-on Reset

When the $\overline{\text{RES}}$ pin is driven low, the SH7065 enters the power-on reset state. To ensure that the SH7065 is properly reset, the $\overline{\text{RES}}$ pin must be held low for at least the oscillation settling time when powering on or when in standby mode (when the clock is stopped), or at least 40 tcyc (of the slowest module clock) when the clock is running. In the power-on reset state, the internal state of the CPU and all on-chip peripheral module registers are initialized. See appendix B, Pin States, for the pin states in the power-on reset state.

When the $\overline{\text{RES}}$ pin is driven high after being held low for the necessary time in the power-on reset state, power-on reset exception handling is started. CPU operations are as follows.

1. The initial value of the program counter (PC) (i.e. the execution start address) is fetched from the exception vector table.
2. The initial value of the stack pointer (SP) is fetched from the exception vector table.
3. The vector base register (VBR) is cleared to H'00000000, and the interrupt mask bits (I3 to I0) in the status register (SR) are set to H'F (1111).
4. The values fetched from the exception vector table are set in the program counter (PC) and stack pointer (SP), and program execution is started.

Power-on reset processing must always be executed when the system is powered on.

5.3 Address Errors

5.3.1 Address Error Sources

Address errors occur in instruction fetches and data read/write accesses, as shown in table 5.5.

Table 5.5 Bus Cycles and Address Errors

Bus Cycle			
Type	**Bus Master**	**Bus Cycle Operation**	**Address Error Occurrence**
Instruction fetch	CPU	Instruction fetched from even address	No error (normal)
		Instruction fetched from odd address	Address error
		Instruction fetched from other than on-chip peripheral module space*	No error (normal)
		Instruction fetched from on-chip peripheral module space*	Address error
		Instruction fetched from external memory space in single-chip mode	Address error
Data read/write	CPU or DMAC	Word data accessed from even address	No error (normal)
		Word data accessed from odd address	Address error
		Longword data accessed from longword boundary	No error (normal)
		Longword data accessed from other than longword boundary	Address error
		Word data or byte data accessed in on-chip peripheral module space*	No error (normal)
		Longword data accessed in 16-bit on-chip peripheral module space*	No error (normal)
		Longword data accessed in 8-bit on-chip peripheral module space*	No error (normal)
		External memory space accessed in single-chip mode	Address error

Note: * For details of the on-chip peripheral module space, see section 8, Bus State Controller (BSC).

5.3.2 Address Error Exception Handling

When an address error occurs, the CPU starts address error exception handling as shown below after the end of the bus cycle in which the error occurred and completion of the currently executing instruction.

1. The status register (SR) is saved on the stack.
2. The program counter (PC) is saved on the stack. The PC value saved is the start address of the instruction following the last instruction executed.
3. The exception service routine start address is fetched from the exception vector table entry corresponding to the address error, and program execution starts from that address. The jump in this case is not a delayed branch.

5.4 Interrupts

5.4.1 Interrupt Sources

Interrupt exception handling can be initiated by NMI, a user break, IRQ, or an on-chip peripheral module, as shown in table 5.6.

Table 5.6 Interrupt Sources

Type	Request Source
NMI	NMI pin (external input)
User break	User break controller
IRQ, IRL	Pins $\overline{\text{IRQ0}}$ to $\overline{\text{IRQ7}}$ (external input)
On-chip peripheral module	Direct memory access controller
	Timer pulse unit
	Compare match timer
	A/D converter
	Serial communication interface
	Watchdog timer
	Bus state controller
	Motor management timer

Each interrupt source is assigned a different vector number and vector table offset. For details of vector numbers and vector table address offsets, see section 6.2.5, Interrupt Exception Vectors and Priority Order.

5.4.2 Interrupt Priority

Interrupt sources are assigned priority levels. If a number of interrupts occur simultaneously (multiple interruption), the priority order is determined by the interrupt controller (INTC) and exception handling is initiated accordingly.

Interrupt source priority levels are expressed as values from 0 to 16, with 0 representing the lowest priority level and 16 as the highest. The NMI interrupt is the highest-priority interrupt at level 16; it cannot be masked and is always accepted. The user break interrupt is assigned priority level 15. The priority level of IRQ interrupts and on-chip peripheral module interrupts can be set as desired in the INTC's interrupt priority registers A to L (IPRA to IPRL) (see table 5.7). Priority levels 0 to 15, but not 16, can be set. For details of IPRA to IPRL, see section 6.3.1, Interrupt Priority Registers A to L (IPRA to IPRL).

Table 5.7 Interrupt Priority Levels

Type	Priority Level	Notes
NMI	16	Fixed priority level, not maskable
User break	15	Fixed priority level
IRQ, IRL	0 to 15	Can be set in interrupt priority registers A to L (IPRA to IPRL)
On-chip peripheral module		

5.4.3 Interrupt Exception Handling

When an interrupt occurs, its priority is determined by the interrupt controller (INTC). NMI is always accepted, but other interrupts are only accepted if their priority level is higher than the priority level set in the interrupt mask bits (I3 to I0) in the status register (SR).

When an interrupt is accepted, interrupt exception handling is started. In interrupt exception handling, the CPU saves SR and the program counter (PC) on the stack and writes the priority level of the accepted interrupt to bits I3 to I0 in SR. In the case of NMI, however, although its priority level is 16, H'F (level 15) is written to bits I3 to I0. Next, the CPU fetches the exception service routine start address from the exception vector table entry corresponding to the accepted interrupt, jumps to that address, and starts executing the exception service routine. For details of interrupt exception handling, see section 6.4, Operation.

5.5 Instruction Exceptions

5.5.1 Types of Instruction Exception

There are three kinds of instruction that can initiate exception handling: the TRAP instruction, slot illegal instructions, and general illegal instructions, These are summarized in table 5.8.

Table 5.8 Instruction Exception Types

Type	Source Instructions	Notes
Trap instruction	TRAPA	
Slot illegal instruction	Undefined code or instruction that modifies PC located immediately after delayed branch instruction (in delay slot)	Delayed branch instructions: JMP, JSR, BRA, BSR, RTS, RTE, BF/S, BT/S, BSRF, BRAF Instructions that modify PC: JMP, JSR, BRA, BSR, RTS, RTE, BT, BF, TRAPA, BF/S, BT/S, BSRF, BRAF
General illegal instruction	Undefined code other than in delay slot	

5.5.2 Trap Instruction

When a TRAPA instruction is executed, trap instruction exception handling is started. The CPU operates as follows.

1. The status register (SR) is saved on the stack.
2. The program counter (PC) is saved on the stack. The PC value saved is the start address of the instruction following the trap instruction.
3. The exception service routine start address is fetched from the exception vector table entry corresponding to the vector number specified by the TRAPA instruction, a jump is made to that address, and program execution starts from that point. The jump in this case is not a delayed branch.

5.5.3 Slot Illegal Instructions

An instruction located immediately after a delayed branch instruction is said to be located in the delay slot. If the instruction in the delay slot is undefined code, slot illegal instruction exception handling is started when that undefined code is decoded. Also, if the instruction in the delay slot is one that modifies the program counter (PC), slot illegal instruction exception handling is started when that instruction is decoded. CPU operations in slot illegal instruction exception handling are as follows.

1. The status register (SR) is saved on the stack.
2. The program counter (PC) is saved on the stack. The PC value saved is the jump destination address of the delayed branch instruction immediately preceding the undefined code or PC-modifying instruction.
3. The exception service routine start address is fetched from the exception vector table entry corresponding to the generated exception, a jump is made to that address, and program execution starts from that point. The jump in this case is not a delayed branch.

5.5.4 General Illegal Instructions

When undefined code located other than immediately after a delayed branch instruction (in a delay slot) is decoded, general illegal instruction exception handling is started. The CPU follows the same procedure as in the case of slot illegal instruction exception handling, except that the program counter (PC) value saved is the start address of the undefined code.

5.6 Cases in Which Exceptions Are Not Accepted

There are cases, as shown in table 5.9, in which, if an address error or interrupt occurs after a delayed branch instruction or an interrupt for which interruption is prohibited, the exception is not accepted immediately, but is held pending. In such cases, the address error or interrupt will be accepted when an instruction for which exception acceptance is permitted is decoded.

Table 5.9 Exception Occurrence: Special Cases

	Exception Source	
Point of Occurrence	**Address Error**	**Interrupt**
Immediately after a delayed branch instruction*1	Not accepted	Not accepted
Immediately after an instruction for which interruption is prohibited*2	Accepted	Not accepted
Repeat loop comprising up to three instructions (instruction fetch cycle not generated) First instruction or last three instructions in a repeat loop containing four or more instructions	Not accepted	Not accepted
Fourth from last instruction in a repeat loop containing four or more instructions	Accepted	Not accepted

Notes: 1. Delayed branch instructions: JMP, JSR, BRA, BSR, RTS, RTE, BF/S, BT/S, BSRF, BRAF

2. Instructions for which interruption is prohibited: LDC, LDC.L, STC, STC.L, LDS, LDS.L, STS, STS.L

5.6.1 After a Delayed Branch Instruction

When an instruction located immediately after a delayed branch instruction (i.e. in the delay slot) is decoded, neither an address error nor an interrupt is accepted. As a delayed branch instruction and the instruction located immediately after it (in the delay slot) are always executed consecutively, exception handling is not initiated during this period.

5.6.2 After an Instruction for Which Interruption Is Prohibited

When the instruction immediately following an instruction for which interruption is prohibited is decoded, an interrupt is not accepted. However, an address error exception is accepted.

5.6.3 Instructions in Repeat Loops

If a repeat loop comprises up to three instructions, neither exceptions nor interrupts are accepted. If a repeat loop contains four or more instructions, neither exceptions nor interrupts are accepted during the execution cycle of the first instruction or the last three instructions. If a repeat loop contains four or more instructions, address errors only are accepted during the execution cycle of the fourth from last instruction. For more information, see the *SH-1/SH-2/SH-DSP Software Manual*.

A. All interrupts and address errors are accepted.
B. Address errors only are accepted.
C. No interrupts or address errors are accepted.

When RC > = 1;
Operation depends on the number of instructions in the repeat loop, as follows.

1. One instruction

```
                    ← A
              instr0
                    ← B
Start (End):  instr1
                    ← C
              instr2
                    ← A
```

2. Two instructions

```
                ← A
          instr0
                ← B
Start:    instr1
                ← C
End:      instr2
                ← C
          instr3
                ← A
```

3. Three instructions

```
                ← A
          instr0
                ← B
Start:    instr1
                ← C
          instr2
                ← C
End:      instr3
                ← C
          instr4
                ← A
```

4. Four instructions

```
                ← A
        instr0
                ← A or C*1
Start:  instr1
                ← B
        instr2
                ← C
        instr3
                ← C
End:    instr4
                ← C
        instr5
                ← A
```

5. Five or more instructions

```
                    ← A
        instr0
                    ← A or C*2
Start:  instr1
                    ← A
        :
                    :
        :
                    ← A
        instr n-3
                    ← B
        instr n-2
                    ← C
        instr n-1
                    ← C
End:    instr n
                    ← C
        instr n+1
                    ← A
```

Notes: 1. On return from instr 4
2. On return from instr n

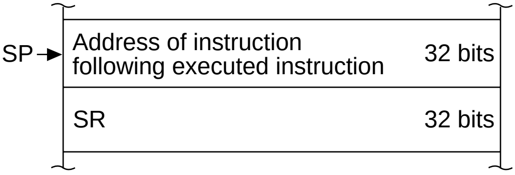
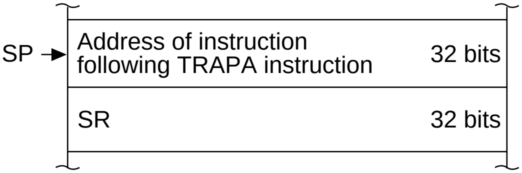
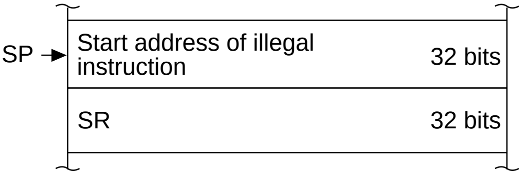
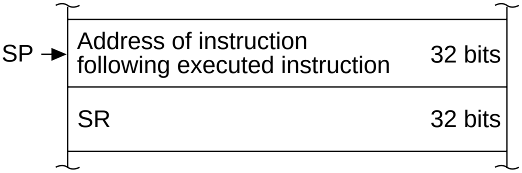
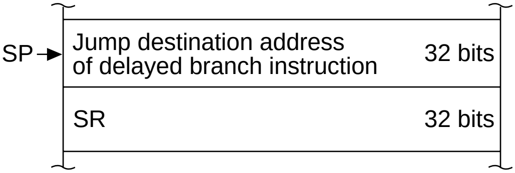
When RC = 0;
All interrupts and address errors are accepted.

Figure 5.1 Restrictions on Interrupt Acceptance in Repeat Mode

5.7 Stack Status after Exception Handling

Table 5.10 shows the stack after completion of exception handling.

Table 5.10 Stack Status after Exception Handling

Type	Stack Status
Address error	 SP → Address of instruction following executed instruction 32 bits SR 32 bits
TRAP instruction	 SP → Address of instruction following TRAPA instruction 32 bits SR 32 bits
General illegal instruction	 SP → Start address of illegal instruction 32 bits SR 32 bits
Interrupt	 SP → Address of instruction following executed instruction 32 bits SR 32 bits
Slot illegal instruction	 SP → Jump destination address of delayed branch instruction 32 bits SR 32 bits

5.8 Usage Notes

5.8.1 Stack Pointer (SP) Value

Ensure that the stack pointer (SP) value is a multiple of 4. If it is not, an address error will be caused when the stack is accessed in exception handling.

5.8.2 Vector Base Register (VBR) Value

Ensure that the vector base register (VBR) value is a multiple of 4. If it is not, an address error will be caused when the stack is accessed in exception handling.

5.8.3 Address Errors Occurring in Address Error Exception Handling Stacking

If the stack pointer (SP) value is not a multiple of 4, an address error will occur in exception handling (interrupt, etc.) stacking, and after the exception handling is completed, address error exception handling will be started. An address error will also occur in stacking in the address error exception handling, but this address error will not be accepted in order to prevent endless stacking due to address errors. This enables program control to be switched to the address error exception service routine, and error handling to be carried out.

When an address error occurs in exception handling stacking, the stacking bus cycle (write) is executed. In status register (SR) and program counter (PC) stacking, SP is decremented by 4 in each case, and therefore the SP value is not a multiple of 4 after stacking is completed. Also, the address value output in stacking is the SP value, and the actual address at which the error occurred is output. In this case, the stacked write data is undefined.

Section 6 Interrupt Controller (INTC)

6.1 Overview

The interrupt controller (INTC) ascertains the priority of interrupt sources and controls interrupt requests to the CPU. The INTC registers set the order of priority of each interrupt, allowing the user to handle interrupt requests according to a user-set priority order.

6.1.1 Features

The INTC has the following features.

- Two external interrupt modes
  - IRQ mode

    Eight external signals comprise independent interrupt sources ($\overline{\text{IRQ7}}$ to $\overline{\text{IRQ0}}$). Each interrupt source has an interrupt vector, and can be assigned a priority level.
  - IRL mode

    Four external interrupt signals ($\overline{\text{IRQ3}}$ to $\overline{\text{IRQ0}}$) can be assigned a priority level from 1 to 15. External interrupt signals $\overline{\text{IRQ4}}$ to $\overline{\text{IRQ7}}$ function as independent interrupt sources.
- 16 interrupt priority levels

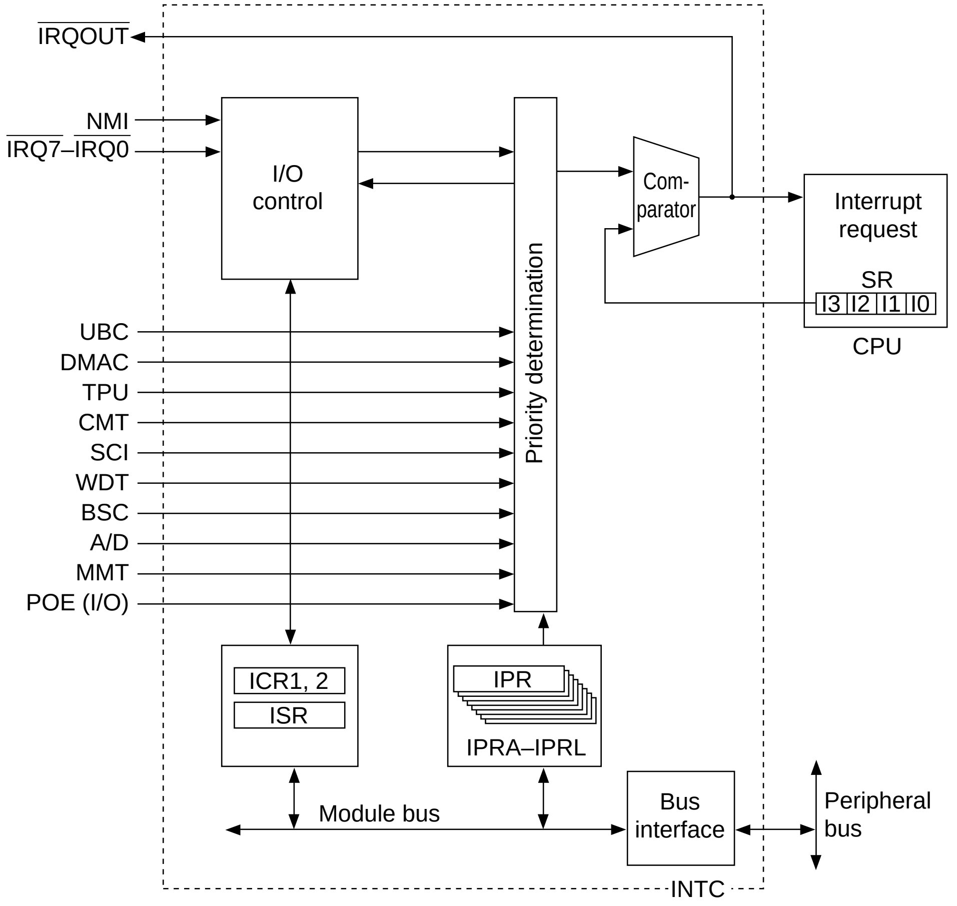
  Using 12 interrupt priority registers, one of 15 priority levels can be assigned to each IRQ interrupt and on-chip peripheral module interrupt source. Priority level 16 is automatically assigned to the NMI interrupt.
- NMI noise canceler function

  An NMI input level bit is provided to indicate the NMI pin state. The pin state can be checked by reading this bit in the interrupt exception service routine, enabling it to be used as a noise canceler.
- Interrupt occurrence can be reported externally ($\overline{\text{IRQOUT}}$ pin)

  When the SH7065 has released the bus, for example, this function can be used to report interrupt generation to an external bus master, and request the bus.

6.1.2 Block Diagram

Figure 6.1 shows a block diagram of the INTC.



Legend:

UBC:	User break controller
DMAC:	Direct memory access controller
TPU:	Timer pulse unit
CMT:	Compare match timer
SCI:	Serial communication interface
WDT:	Watchdog timer
BSC:	Bus state controller (DRAM refresh control unit)
A/D:	A/D converter
MMT:	Motor management timer
ICR1, ICR2:	Interrupt control registers 1 and 2
ISR:	IRQ status register
IPRA to IPRL:	Interrupt priority registers A to L
SR:	Status register
POE (I/O):	Port output enable

Figure 6.1 Block Diagram of INTC

6.1.3 Pin Configuration

Table 6.1 shows the INTC pin configuration.

Table 6.1 INTC Pins

Pin Name	I/O	Function
NMI	Input	Input of nonmaskable interrupt request signal
$\overline{\text{IRQ7}}$–$\overline{\text{IRQ0}}$	Input	Input of maskable interrupt request signals
$\overline{\text{IRQOUT}}$	Output	Output of signal indicating interrupt source occurrence

6.1.4 Register Configuration

The INTC has the 15 registers shown in table 6.2. The functions of these registers include interrupt priority level setting and control of external interrupt input signal detection.

Table 6.2 INTC Registers

Name	Abbreviation	R/W	Initial Value	Address	Access Size
Interrupt priority register A	IPRA	R/W	H'0000	H'FFFF 1050	8, 16, 32
Interrupt priority register B	IPRB	R/W	H'0000	H'FFFF 1052	8, 16, 32
Interrupt priority register C	IPRC	R/W	H'0000	H'FFFF 1054	8, 16, 32
Interrupt priority register D	IPRD	R/W	H'0000	H'FFFF 1056	8, 16, 32
Interrupt priority register E	IPRE	R/W	H'0000	H'FFFF 1058	8, 16, 32
Interrupt priority register F	IPRF	R/W	H'0000	H'FFFF 105A	8, 16, 32
Interrupt priority register G	IPRG	R/W	H'0000	H'FFFF 105C	8, 16, 32
Interrupt priority register H	IPRH	R/W	H'0000	H'FFFF 105E	8, 16, 32
Interrupt priority register I	IPRI	R/W	H'0000	H'FFFF 1060	8, 16, 32
Interrupt priority register J	IPRJ	R/W	H'0000	H'FFFF 1062	8, 16, 32
Interrupt priority register K	IPRK	R/W	H'0000	H'FFFF 1064	8, 16, 32
Interrupt priority register L	IPRL	R/W	H'0000	H'FFFF 1066	8, 16, 32
Interrupt control register 1	ICR1	R/W	*1	H'FFFF 106E	8, 16, 32
Interrupt control register 2	ICR2	R/W	H'0000	H'FFFF 1070	8, 16, 32
IRQ status register	ISR	R/(W)*2	H'0000	H'FFFF 1072	8, 16, 32

Notes: 1. Bit 15 (NMIL) indicates the level of the signal being input to the NMI pin. For details, see section 6.3.2, Interrupt Control Register 1 (ICR1).

2. See section 6.3.4, IRQ Status Register (ISR), for details.

6.2 Interrupt Sources

There are four types of interrupt sources: NMI, user break, IRQ/IRL, and on-chip peripheral modules. Each interrupt has a priority level (0 to 16), with level 0 as the lowest and level 16 as the highest. If level 0 is set, the interrupt is masked.

6.2.1 NMI Interrupt

The NMI interrupt has the highest priority level of 16, and is always accepted. Input from the NMI pin is edge-detected. The NMI edge select bit (NMIE) in the interrupt control register 1 (ICR1) is used to select either rising or falling edge detection.

NMI interrupt exception handling sets the interrupt mask bits (I3 to I0) in the status register (SR) to 15. The NMI pin level is set in the NMIL bit in the interrupt control register 1 (ICR1). Interrupt requests resulting from erroneous edge detection due to noise can be avoided by referencing the NMIL bit in the interrupt exception service routine.

6.2.2 User Break Interrupt

The user break interrupt is requested when a break condition set in the user break controller (UBC) occurs. Its priority level is 15. A user break interrupt is edge-detected, and is retained until acknowledged. User break exception handling sets the interrupt mask bits (I3 to I0) in the status register (SR) to 15. For details of user breaks, see section 7, User Break Controller (UBC).

6.2.3 External Interrupts

IRQ interrupt mode (initial setting) or IRL interrupt mode can be selected for external interrupts. The selection is made with the EXIMD bit in interrupt control register 1 (ICR1).

IRQ Interrupts

IRQ interrupts correspond to pins $\overline{\text{IRL7}}$ to $\overline{\text{IRL0}}$. Low-level detection or falling edge detection can be selected independently for each pin with the IRQ sense select bits (IRQ7S to IRQ0S) in the interrupt control register 2 (ICR2), and a priority level of 0 to 15 can be selected independently for each pin by means of interrupt priority registers A and B. IRQ interrupt exception handling sets the interrupt mask bits (I3 to I0) in SR to the priority level of the accepted IRQ interrupt.

IRL Interrupts

IRL interrupts are requested by input at external pins $\overline{\text{IRQ3}}$ to $\overline{\text{IRQ0}}$. Fifteen interrupts, IRL15 to IRL1, can be input from an external source by means of pins $\overline{\text{IRQ3}}$ to $\overline{\text{IRQ0}}$. Interrupts IRL15 to IRL1 have priority levels of 15 to 1, respectively, and are assigned vector numbers 79 to 64. In IRL interrupt mode, external interrupts $\overline{\text{IRQ4}}$ to $\overline{\text{IRQ7}}$ are independent interrupt sources. The priority levels of $\overline{\text{IRQ4}}$ to $\overline{\text{IRQ7}}$ can be set in interrupt priority register B (IPRB), as in IRQ interrupt mode. (An example of interrupt connection is shown in figure 6.2.)

Table 6.3 IRL Interrupt Priority Levels and Vector Numbers

	Signals					
Interrupt	**$\overline{\text{IRQ3}}$**	**$\overline{\text{IRQ2}}$**	**$\overline{\text{IRQ1}}$**	**$\overline{\text{IRQ0}}$**	**Priority Level**	**Vector Number**
IRL15	0	0	0	0	15	79
IRL14	0	0	0	1	14	78
IRL13	0	0	1	0	13	77
IRL12	0	0	1	1	12	76
IRL11	0	1	0	0	11	75
IRL10	0	1	0	1	10	74
IRL9	0	1	1	0	9	73
IRL8	0	1	1	1	8	72
IRL7	1	0	0	0	7	71
IRL6	1	0	0	1	6	70
IRL5	1	0	1	0	5	69
IRL4	1	0	1	1	4	68
IRL3	1	1	0	0	3	67
IRL2	1	1	0	1	2	66
IRL1	1	1	1	0	1	65
—	1	1	1	1	0 (no interrupt)	64

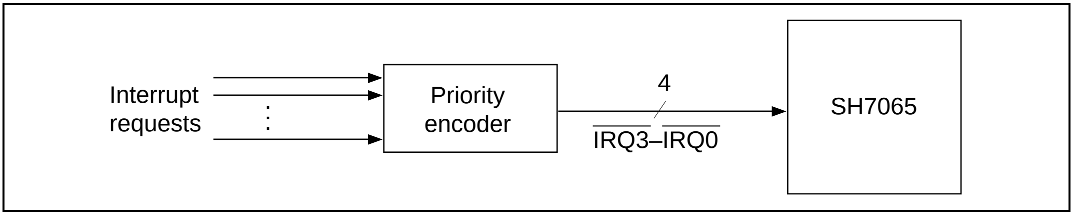


Figure 6.2 Example of IRL Mode Interrupt Connection

6.2.4 On-Chip Peripheral Module Interrupts

On-chip peripheral module interrupts are generated by the following modules:

- Direct memory access controller (DMAC)
- Watchdog timer (WDT)
- Bus state controller (BSC)
- Timer pulse unit (TPU)
- Serial communication interface (SCI)
- Compare match timer (CMT)
- Motor management timer (MMT)
- A/D converter (A/D)
- Port output enable (POE (I/O))

As a different vector number is assigned to each source, it is not necessary to determine the source in the exception service routine. A priority level in the range 0 to 15 can be set for each module with interrupt priority registers E to L (IPRE to IPRL). In on-chip peripheral module interrupt exception handling, the interrupt mask bits (I3 to I0) in the status register (SR) are set to the priority level of the accepted on-chip peripheral module interrupt.

6.2.5 Interrupt Exception Vectors and Priority Order

Tables 6.4 to 6.6 show interrupt sources, vector numbers, vector table addresses, and default interrupt priorities.

Each interrupt source is assigned a different vector number and vector table address offset. The vector table address is calculated from the vector number and vector table address offset. In interrupt exception handling, the start address of the exception service routine is fetched from the vector table entry indicated by this vector table address. For the method of calculating the vector table address, see table 5.4, Exception Vector Table Address Calculation, in section 5, Exception Handling.

In IRQ mode, an interrupt priority level of 0 to 15 can be assigned to the IRQ interrupts using interrupt priority registers A and B (IPRA, IPRB).

In IRL mode, IRL interrupts IRL15 to IRL1 are assigned interrupt priority levels 15 to 1, respectively. The vectors shown in tables 6.3 to 6.5 can be used for the vector numbers of IRQ and IRL interrupts.

The priority of on-chip peripheral module interrupts can be set to any level from 0 to 15 for each module using interrupt priority registers E to L (IPRE to IPRL). The "Priority within IPR Setting" column in table 6.6 shows the relative priority of interrupts sharing the same IPR field. This priority order cannot be changed. In a power-on reset, IRQ interrupts and on-chip peripheral module interrupts are set to priority level 0. If two or more interrupt sources assigned the same priority level occur simultaneously, they are handled according to the default priority order shown in tables 6.4 to 6.6.

Table 6.4 IRQ Mode Interrupt Exception Vectors and Priority Order

Interrupt Source	Interrupt Priority (Initial Value)	IPR (Bit Numbers)	Vector		Default Priority
			Vector Number	Vector Table Offset	
NMI	16	—	12	H'0000 0030	High
User break	15	—	13	H'0000 0034	
IRQ0	0–15 (0)	IPRA (15–12)	64	H'0000 0100	
IRQ1	0–15 (0)	IPRA (11–8)	65	H'0000 0104	
IRQ2	0–15 (0)	IPRA (7–4)	66	H'0000 0108	
IRQ3	0–15 (0)	IPRA (3–0)	67	H'0000 010C	
IRQ4	0–15 (0)	IPRB (15–12)	80	H'0000 0140	
IRQ5	0–15 (0)	IPRB (11–8)	81	H'0000 0144	
IRQ6	0–15 (0)	IPRB (7–4)	82	H'0000 0148	
IRQ7	0–15 (0)	IPRB (3–0)	83	H'0000 014C	Low

Table 6.5 IRL Mode Interrupt Exception Vectors and Priority Order

Interrupt Source	Interrupt Priority (Initial Value)	IPR (Bit Numbers)	Vector		Default Priority
			Vector Number	Vector Table Offset	
NMI	16	—	12	H'0000 0030	High
User break	15	—	13	H'0000 0034	
IRL15	15	—	79	H'0000 013C	
IRL14	14	—	78	H'0000 0138	
IRL13	13	—	77	H'0000 0134	
IRL12	12	—	76	H'0000 0130	
IRL11	11	—	75	H'0000 012C	
IRL10	10	—	74	H'0000 0128	
IRL9	9	—	73	H'0000 0124	
IRL8	8	—	72	H'0000 0120	
IRL7	7	—	71	H'0000 011C	
IRL6	6	—	70	H'0000 0118	
IRL5	5	—	69	H'0000 0114	
IRL4	4	—	68	H'0000 0110	
IRL3	3	—	67	H'0000 010C	
IRL2	2	—	66	H'0000 0108	
IRL1	1	—	65	H'0000 0104	Low

Table 6.6 On-Chip Peripheral Module Interrupt Exception Vectors and Priority Order

Interrupt Source		Interrupt Priority (Initial Value)	IPR (Bit Numbers)	Priority within IPR Setting	Vector Number	Vector Table Offset	Default Priority
DMAC0	DEI0	0–15 (0)	IPRE (15–12)	—	128	H'0000 0200	High
DMAC1	DEI1	0–15 (0)	IPRE (11–8)	—	129	H'0000 0204	↕
DMAC2	DEI2	0–15 (0)	IPRE (7–4)	—	130	H'0000 0208	
DMAC3	DEI3	0–15 (0)	IPRE (3–0)	—	131	H'0000 020C	
Reserved		0–15 (0)	IPRF (15–12)	—	132	H'0000 0210	
		0–15 (0)	IPRF (11–8)	—	133	H'0000 0214	
		0–15 (0)	IPRF (7–4)	—	134	H'0000 0218	
		0–15 (0)	IPRF (3–0)	—	135	H'0000 021C	
BSC	CMI	0–15 (0)	IPRG (15–12)	—	136	H'0000 0220	
	OVI	0–15 (0)	IPRG (11–8)	—	137	H'0000 0224	
WDT	ITI	0–15 (0)	IPRG (7–4)	—	138	H'0000 0228	
Reserved		0–15 (0)	IPRG (3–0)	—	139	H'0000 022C	
TPU0	TGI0A	0–15 (0)	IPRH (15–12)	High	140	H'0000 0230	
	TGI0B			↕	141	H'0000 0234	
	TGI0C				142	H'0000 0238	
	TGI0D			Low	143	H'0000 023C	
	TCI0V	0–15 (0)	IPRH (11–8)	High	144	H'0000 0240	
	Reserved			↕	145	H'0000 0244	
	Reserved				146	H'0000 0248	
	Reserved			Low	147	H'0000 024C	
TPU1	TGI1A	0–15 (0)	IPRH (7–4)	High	148	H'0000 0250	
	TGI1B			↕	149	H'0000 0254	
	Reserved				150	H'0000 0258	
	Reserved			Low	151	H'0000 025C	
	TCI1V	0–15 (0)	IPRH (3–0)	High	152	H'0000 0260	
	TCI1U			↕	153	H'0000 0264	
	Reserved				154	H'0000 0268	
	Reserved			Low	155	H'0000 026C	Low

Interrupt Source		Interrupt Priority (Initial Value)	IPR (Bit Numbers)	Priority within IPR Setting	Vector Number	Vector Table Offset	Default Priority
TPU2	TGI2A	0–15 (0)	IPRI (15–12)	High	156	H'0000 0270	High
	TGI2B			↑	157	H'0000 0274	↑
	Reserved			↓	158	H'0000 0278	
	Reserved			Low	159	H'0000 027C	
	TCI2V	0–15 (0)	IPRI (11–8)	High	160	H'0000 0280	
	TCI2U			↑	161	H'0000 0284	
	Reserved			↓	162	H'0000 0288	
	Reserved			Low	163	H'0000 028C	
TPU3	TGI3A	0–15 (0)	IPRI (7–4)	High	164	H'0000 0290	
	TGI3B			↑	165	H'0000 0294	
	TGI3C			↓	166	H'0000 0298	
	TGI3D			Low	167	H'0000 029C	
	TCI3V	0–15 (0)	IPRI (3–0)	High	168	H'0000 02A0	
	Reserved			↑	169	H'0000 02A4	
	Reserved			↓	170	H'0000 02A8	
	Reserved			Low	171	H'0000 02AC	
TPU4	TGI4A	0–15 (0)	IPRJ(15–12)	High	172	H'0000 02B0	
	TGI4B			↑	173	H'0000 02B4	
	Reserved			↓	174	H'0000 02B8	
	Reserved			Low	175	H'0000 02BC	
	TCI4V	0–15 (0)	IPRJ(11–8)	High	176	H'0000 02C0	
	TCI4U			↑	177	H'0000 02C4	
	Reserved			↓	178	H'0000 02C8	↓
	Reserved			Low	179	H'0000 02CC	Low

Interrupt Source		Interrupt Priority (Initial Value)	IPR (Bit Numbers)	Priority within IPR Setting	Vector Number	Vector Table Offset	Default Priority
TPU5	TGI5A	0–15 (0)	IPRJ (7–4)	High	180	H'0000 02D0	High
	TGI5B			↑	181	H'0000 02D4	↑
	Reserved			↓	182	H'0000 02D8	
	Reserved			Low	183	H'0000 02DC	
	TCI5V	0–15 (0)	IPRJ (3–0)	High	184	H'0000 02E0	
	TCI5U			↑	185	H'0000 02E4	
	Reserved			↓	186	H'0000 02E8	
	Reserved			Low	187	H'0000 02EC	
SCI0	ERI0	0–15 (0)	IPRK (15–12)	High	188	H'0000 02F0	
	RXI0			↑	189	H'0000 02F4	
	TXI0			↓	190	H'0000 02F8	
	TEI0			Low	191	H'0000 02FC	
SCI1	ERI1	0–15 (0)	IPRK (11–8)	High	192	H'0000 0300	
	RXI1			↑	193	H'0000 0304	
	TXI1			↓	194	H'0000 0308	
	TEI1			Low	195	H'0000 030C	
SCI2	ERI2	0–15 (0)	IPRK (7–4)	High	196	H'0000 0310	
	RXI2			↑	197	H'0000 0314	
	TXI2			↓	198	H'0000 0318	
	TEI2			Low	199	H'0000 031C	
Reserved	Reserved	0–15 (0)	IPRK (3–0)	High	200	H'0000 0320	
	Reserved			↑	201	H'0000 0324	
	Reserved			↓	202	H'0000 0328	
	Reserved			Low	203	H'0000 032C	
CMT	CMI0	0–15 (0)	IPRL (15–12)	High	204	H'0000 0330	
	CMI1			↑	205	H'0000 0334	
	Reserved			↓	206	H'0000 0338	↓
	Reserved			Low	207	H'0000 033C	Low

Interrupt Source		Interrupt Priority (Initial Value)	IPR (Bit Numbers)	Priority within IPR Setting	Vector Number	Vector Table Offset	Default Priority
A/D	ADI0	0–15 (0)	IPRL (11–8)	High	208	H'0000 0340	High
	ADI1				209	H'0000 0344	
	Reserved				210	H'0000 0348	
	Reserved			Low	211	H'0000 034C	
MMT	TGIM	0–15 (0)	IPRL (7–4)	High	212	H'0000 0350	
	TGIN				213	H'0000 0354	
	Reserved				214	H'0000 0358	
	Reserved			Low	215	H'0000 035C	
POE (I/O)	OEI	0–15 (0)	IPRL (3–0)	High	216	H'0000 0360	
	Reserved				217	H'0000 0364	
	Reserved				218	H'0000 0368	
	Reserved			Low	219	H'0000 036C	Low

6.3 Register Descriptions

6.3.1 Interrupt Priority Registers A to L (IPRA to IPRL)

Bit:	15	14	13	12	11	10	9	8
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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

Interrupt priority registers A to L (IPRA to IPRL) are 16-bit readable/writable registers that set priority levels from 0 to 15 for IRQ interrupts and on-chip peripheral module interrupts. Table 6.7 shows the relationship between the interrupt request sources and bits in registers IPRA to IPRL.

Table 6.7 Interrupt Request Sources and Registers IPRA to IPRL

	Bits			
Register	**15–12**	**11–8**	**7–4**	**3–0**
Interrupt priority register A	$\overline{\text{IRQ0}}$	$\overline{\text{IRQ1}}$	$\overline{\text{IRQ2}}$	$\overline{\text{IRQ3}}$
Interrupt priority register B	$\overline{\text{IRQ4}}$	$\overline{\text{IRQ5}}$	$\overline{\text{IRQ6}}$	$\overline{\text{IRQ7}}$
Interrupt priority register C	Reserved	Reserved	Reserved	Reserved
Interrupt priority register D	Reserved	Reserved	Reserved	Reserved
Interrupt priority register E	DMAC0	DMAC1	DMAC2	DMAC3
Interrupt priority register F	Reserved	Reserved	Reserved	Reserved
Interrupt priority register G	BSC	BSC	WDT	Reserved
Interrupt priority register H	TPU0	TPU0	TPU1	TPU1
Interrupt priority register I	TPU2	TPU2	TPU3	TPU3
Interrupt priority register J	TPU4	TPU4	TPU5	TPU5
Interrupt priority register K	SCI0	SCI1	SCI2	Reserved
Interrupt priority register L	CMT	A/D	MMT	POE (I/O)

Four IRQ pins or four on-chip peripheral modules are assigned to one register. Interrupt priority levels are established by setting a value from H'0 (0000) to H'F (1111) in each of the four-bit groups: 15 to 12, 11 to 8, 7 to 4, and 3 to 0. Setting H'0 designates priority level 0 (the lowest level), and setting H'F designates priority level 15 (the highest level).

Registers IPRA to IPRL are initialized to H'0000 by a power-on reset. They are not initialized in standby mode. Reserved bits are always read as 0, and should only be written with 0.

6.3.2 Interrupt Control Register 1 (ICR1)

Bit:	15	14	13	12	11	10	9	8
	NMIL*	—	—	—	—	EXIMD	—	NMIE
Initial value:	—	0	0	0	0	0	0	0
R/W:	R	—	—	—	—	R/W	—	R/W

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

Note: * 1 when NMI pin input is high, 0 when low.

Interrupt control register 1 (ICR1) is a 16-bit register that sets the input signal detection mode for external interrupt input pins NMI and $\overline{\text{IRQ7}}$ to $\overline{\text{IRQ0}}$, and indicates the input level at the NMI pin.

ICR1 is initialized to H'0000 or H'8000 by a power-on reset. It is not initialized in standby mode.

Bit 15—NMI Input Level (NMIL): The level of the signal input to the NMI pin is set in this bit. This bit can be read to determine the NMI pin level. It cannot be modified.

Bit 15: NMIL	Description
0	NMI pin input level is low
1	NMI pin input level is high

Bits 14 to 11—Reserved: These bits are always read as 0 and cannot be modified.

Bit 10—External Interrupt Vector Mode Select (EXIMD): This bit selects IRQ mode or IRL mode. In IRQ mode, each of signals $\overline{\text{IRQ7}}$ to $\overline{\text{IRQ0}}$ functions as a separate interrupt source. In IRL mode, signals $\overline{\text{IRQ3}}$ to $\overline{\text{IRQ0}}$ specify an interrupt priority level from 1 to 15, and each of signals $\overline{\text{IRQ7}}$ to $\overline{\text{IRQ4}}$ functions as a separate interrupt source.

Bit 10: EXIMD	Description	
0	IRQ mode	(Initial value)
1	IRL mode	

Bit 9—Reserved: This bit is always read as 0 and cannot be modified.

Bit 8—NMI Edge Select (NMIE): Specifies whether an interrupt request is detected at the falling or rising edge of NMI input.

Bit 8: NMIE	Description	
0	Interrupt request detected at falling edge of NMI input	(Initial value)
1	Interrupt request detected at rising edge of NMI input	

Bits 7 to 0—Reserved: These bits are always read as 0 and cannot be modified.

6.3.3 Interrupt Control Register 2 (ICR2)

Bit:	15	14	13	12	11	10	9	8
	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	—	—	—	—	—	—	—	—

Bit:	7	6	5	4	3	2	1	0
	IRQ7S	IRQ6S	IRQ5S	IRQ4S	IRQ3S	IRQ2S	IRQ1S	IRQ0S
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

Interrupt control register 2 (ICR2) is a 16-bit register that sets the input signal detection mode for $\overline{\text{IRQ7}}$ to $\overline{\text{IRQ0}}$.

ICR2 is initialized to H'0000 by a power-on reset. It is not initialized in standby mode.

Bits 15 to 8—Reserved: These bits are always read as 0 and cannot be modified.

Bits 7 to 0—IRQ7 to IRQ0 Sense Select (IRQ7S to IRQ0S): These bits set the IRQ detection mode for IRQ7 to IRQ0 interrupt requests.

Bits 7 to 0: IRQ7S to IRQ0S	Description	
0	Interrupt request detected at low level of IRQ input	(Initial value)
1	Interrupt request detected at falling edge of IRQ input	

6.3.4 IRQ Status Register (ISR)

Bit:	15	14	13	12	11	10	9	8
	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	—	—	—	—	—	—	—	—

Bit:	7	6	5	4	3	2	1	0
	IRQ7F	IRQ6F	IRQ5F	IRQ4F	IRQ3F	IRQ2F	IRQ1F	IRQ0F
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

The IRQ status register (ISR) is a 16-bit register that indicates the interrupt request status of external interrupt input pins $\overline{\text{IRQ7}}$ to $\overline{\text{IRQ0}}$. When edge detection is set for an IRQ interrupt, an interrupt request being retained can be cleared by reading IRQnF while set to 1 and then writing 0 to IRQnF.

ISR is initialized to H'0000 by a power-on reset. It is not initialized in standby mode.

Bits 15 to 8—Reserved: These bits are always read as 0 and cannot be modified.

Bits 7 to 0—IRQ0 to IRQ7 Flags (IRQ0F to IRQ7F): These bits indicate the IRQ7 to IRQ0 interrupt request status.

Bits 7 to 0: **IRQ0F to IRQ7F**	**Detection Setting**	**Description**
0	Level detection	There is no IRQn interrupt request [Clearing condition] When $\overline{\text{IRQn}}$ input is high
	Edge detection	An IRQn interrupt request has not been detected [Clearing conditions] • When 0 is written to IRQnF after reading IRQnF = 1 • When IRQn interrupt exception handling is executed
1	Level detection	There is no IRQn interrupt request [Setting condition] When $\overline{\text{IRQn}}$ input is low
	Edge detection	An IRQn interrupt request has been detected [Setting condition] When a falling edge occurs in $\overline{\text{IRQn}}$ input

The edge detection circuit operates at all times, even when level detection is set. Note, therefore, that IRQnF may be set when a switch is made to edge detection after level detection operation. To cancel an interrupt request (clear IRQnF) when using edge detection, read IRQnF and then write 0 to it. Figure 6.3 shows the interrupt control circuit.

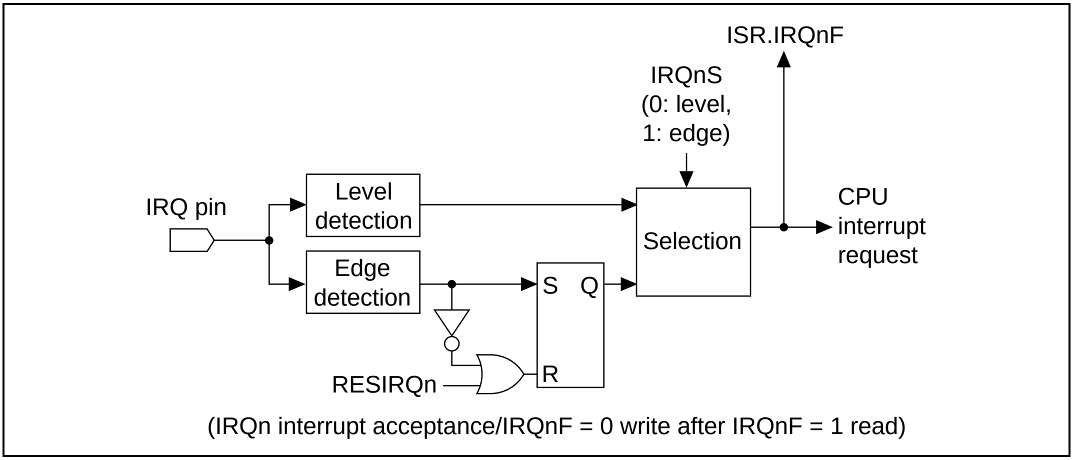


Figure 6.3 External Interrupt Process

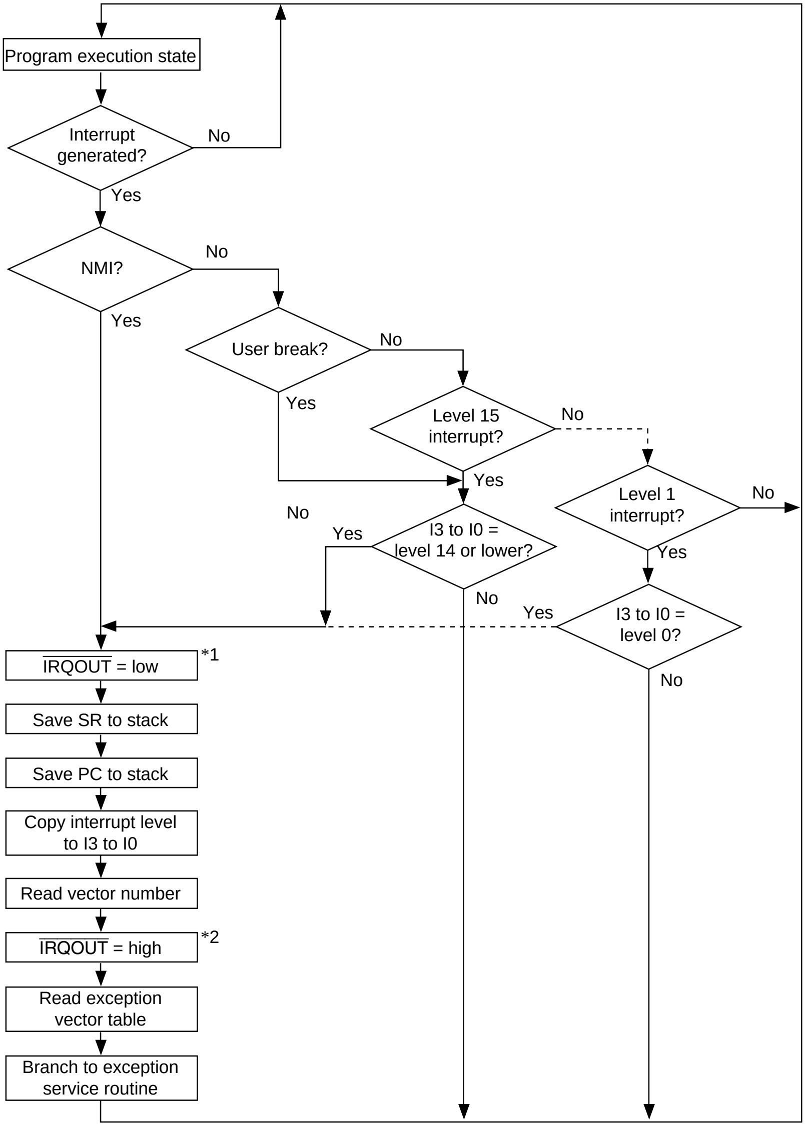
6.4 Operation

6.4.1 Interrupt Operation Sequence

The sequence of operations when an interrupt is requested is described below. Figure 6.4 shows a flowchart of the operations.

1. 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 interrupt priority registers A to L (IPRA to IPRL). Lower-priority interrupts are ignored*. If two of these interrupts have the same priority level, or if multiple interrupts occur within the same module, the interrupt with the highest priority according to the “Default Priority” and “Priority within IPR Setting” entries in table 6.6 is selected. An ignored interrupt is accepted after completion of the higher-level interrupt handling, or can be cleared before the end of the higher-level interrupt handling.
3. The priority level of the interrupt selected by the interrupt controller is compared with the interrupt mask bits (I3 to I0) in the CPU’s status register (SR). An interrupt with a priority level equal to or lower than that set in bits I3 to I0 will be ignored. If the interrupt priority level is higher that the level in bits I3 to I0, the interrupt controller accepts the interrupt and sends an interrupt request signal to the CPU.
4. When the interrupt controller accepts an interrupt, a low-level signal is output from the $\overline{\text{IRQOUT}}$ pin.
5. The interrupt request sent from the interrupt controller is detected when the instruction about to be executed by the CPU is decoded, and execution of that instruction is replaced by interrupt exception handling (see figure 6.5).
6. The status register (SR) and program counter (PC) are saved to the stack.
7. The priority level of the accepted interrupt is written to bits I3 to I0 in SR.
8. If the accepted interrupt is level-detected, or is from an on-chip peripheral module, a high-level signal is output from the $\overline{\text{IRQOUT}}$ pin. If the accepted interrupt is edge-detected, a high-level signal is output from the $\overline{\text{IRQOUT}}$ pin at the point at which the instruction about to be executed by the CPU is replaced by interrupt exception handling in (5). However, if the interrupt controller accepts another interrupt of a higher level than the interrupt in the process of being accepted, the $\overline{\text{IRQOUT}}$ pin remains low.
9. The start address of the exception service routine is fetched form the exception vector table entry corresponding to the accepted interrupt, a jump is made to that address, and program execution starts from that point. The jump in this case is not a delayed branch.

Note: * An interrupt request for which edge detection has been set will be held pending until it can be acknowledged. However, an IRQ interrupt can be cleared by accessing the IRQ status register (ISR). For details, see section 6.2.3, External Interrupts.



Notes: I3 to I0: Interrupt mask bits in the CPU's status register (SR).

1. IRQOUT is the same signal as the interrupt request signal sent to the CPU (see figure 6.1), and so is output in the event of an interrupt request with a higher priority level than that set in bits I3 to I0 in SR.
2. If the accepted interrupt is edge-detected, IRQOUT goes high at the point at which the instruction about to be executed by the CPU is replaced by interrupt exception handling (before SR is saved to the stack). If the interrupt controller is accepting another interrupt with a higher priority level and an interrupt request is being output to the CPU, the IRQOUT pin remains low.

Figure 6.4 Interrupt Operation Flowchart

6.4.2 Interrupt Response Time

The time from generation of an interrupt request until interrupt exception handling is performed and fetching of the first instruction of the exception service routine is started (the interrupt response time) is shown in table 6.8. Figure 6.5 shows an example of pipeline operation when an IRQ interrupt is accepted.

Table 6.8 Interrupt Response Time

Item		Number of States		Notes
		NMI, peripheral Modules	**IRQ**	
Time for priority decision and SR mask bit comparison		2	3	
Wait time until end of sequence being executed by CPU		X (≥ 0)	X (≥ 0)	The longest sequence is for interrupt or address error exception handling (X = 4 + m1 + m2 + m3 + m4). However, the sequence may be even longer if an interrupt-masking instruction follows.
Time from interrupt exception handling until fetch of first instruction of exception service routine is started		5 + m1 + m2 + m3	5 + m1 + m2 + m3	SR and PC save and vector address fetch are performed.
Response time	Total	7 + m1 + m2 + m3	8 + m1 + m2 + m3	
	Minimum case	10	11	At 60-MHz operation: 0.17 to 0.18 μs
	Maximum case	11 + 2 (m1 + m2 + m3) + m4	12 + 2 (m1 + m2 + m3) + m4	At 60-MHz operation: 0.30 to 0.32 μs*

Legend: m1 to m4 are the number of states required for the following memory accesses.

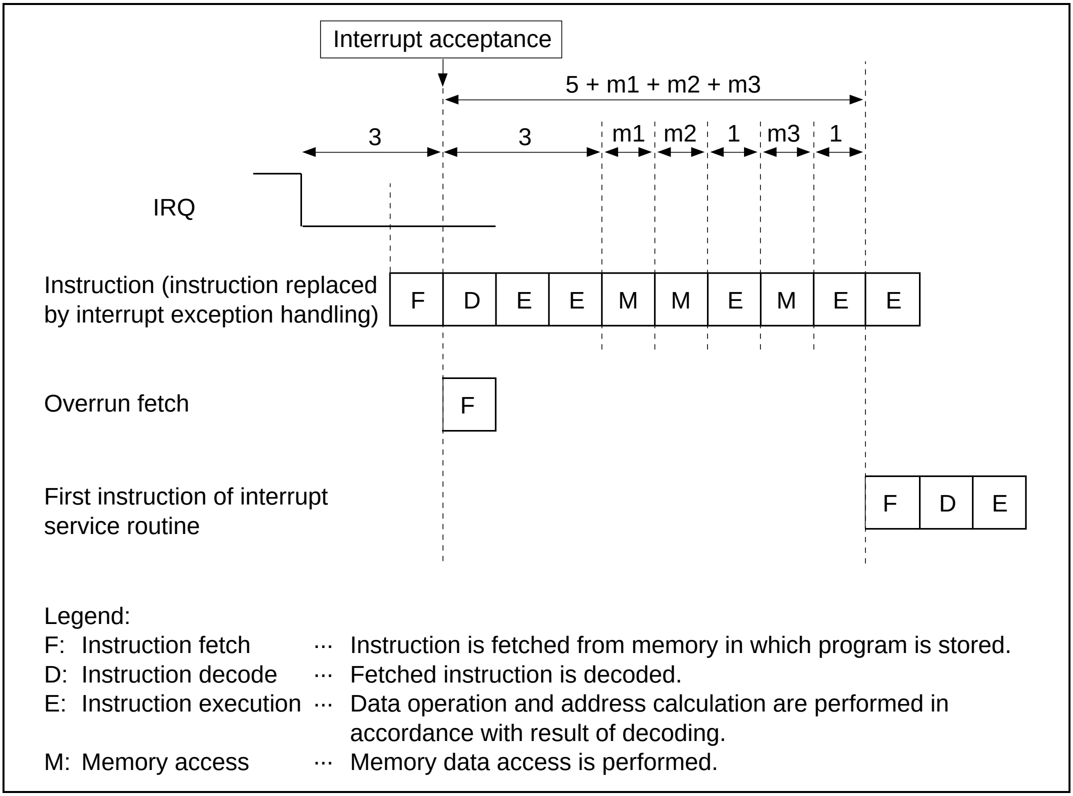
m1: SR save (longword write)

m2: PC save (longword write)

m3: Vector address read (longword read)

m4: Fetch of first instruction of interrupt service routine

Note: * When m1 = m2 = m3 = m4 = 1

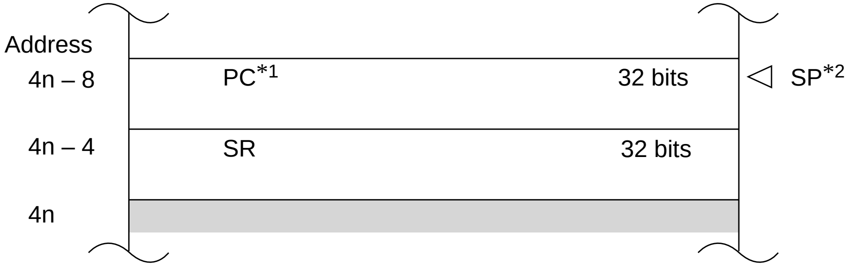

Interrupt acceptance
5 + m1 + m2 + m3
3
3
m1
m2
1
m3
1
IRQ
Instruction (instruction replaced by interrupt exception handling)
F
D
E
E
M
M
E
M
E
E
Overrun fetch
F
First instruction of interrupt service routine
F
D
E

Legend:
F: Instruction fetch ··· Instruction is fetched from memory in which program is stored.
D: Instruction decode ··· Fetched instruction is decoded.
E: Instruction execution ··· Data operation and address calculation are performed in accordance with result of decoding.
M: Memory access ··· Memory data access is performed.

Figure 6.5 Example of Pipeline Operations when IRQ Interrupt is Accepted

6.4.3 Stack Status after Interrupt Exception Handling

Figure 6.6 shows the stack after completion of interrupt exception handling.



Notes: 1. PC: Start address of instruction following executed instruction (i.e. return destination instruction)
2. Ensure that the SP value is a multiple of 4.

Figure 6.6 Stack Status after Interrupt Exception Handling

6.5 Sampling of Signals IRQ3 to IRQ0 in IRL Mode

In IRL mode, interrupt request signals IRQ3 to IRQ0 pass through a noise canceler before being sent by the interrupt controller to the CPU as an interrupt request. The noise canceler eliminates minute width variations in the signals. The CPU samples interrupts between executing instructions. During this period, the noise canceler varies its output according to the signal level after noise cancellation, so the signal level must be held until the CPU completes its sampling operation. Therefore, interrupt source clearing should normally be carried out after making the transition to the interrupt routine.

Figure 6.7 shows a block diagram of the interrupt response mechanism, and figure 6.8 shows the interrupt response timing.

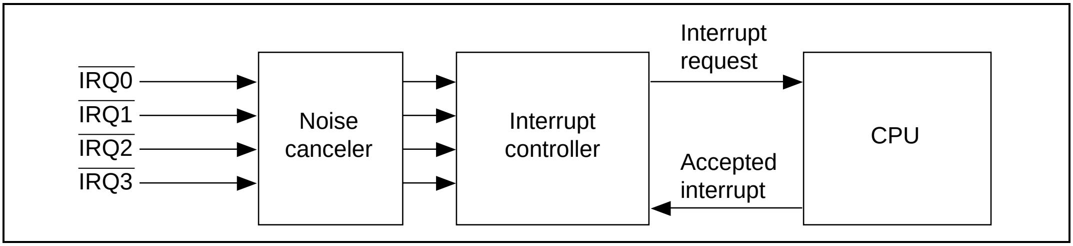


Figure 6.7 Interrupt Response Block Diagram

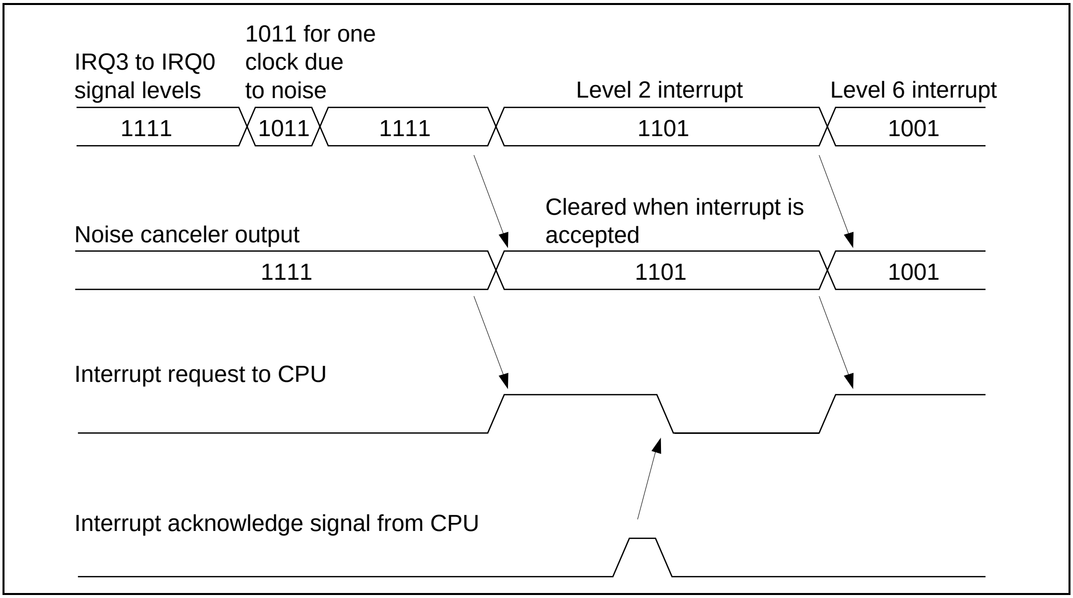

IRQ3 to IRQ0 signal levels
1011 for one clock due to noise
Level 2 interrupt
Level 6 interrupt
1111
1011
1111
1101
1001
Cleared when interrupt is accepted
Noise canceler output
1111
1101
1001
Interrupt request to CPU
Interrupt acknowledge signal from CPU

Figure 6.8 Interrupt Response Timing Chart

6.6 Data Transfer by Means of Interrupt Request Signal

The DMAC can be activated, and data transfer performed, by means of an interrupt request signal. Interrupt sources designated as DMAC activation sources are masked, and not input to the INTC. The mask condition is as follows:

Mask condition = DME · (DE0 · source selection 0 + DE1 · source selection 1 + DE2 · source selection 2 + DE3 · source selection 3)

A control block diagram is shown in figure 6.9.

DME is bit 0 of the DMAC's DMAOR register, and DEn (n = 0 to 3) is bit 0 of DMAC registers CHR0 to CHR3. For details see section 9, Direct Memory Access Controller (DMAC).

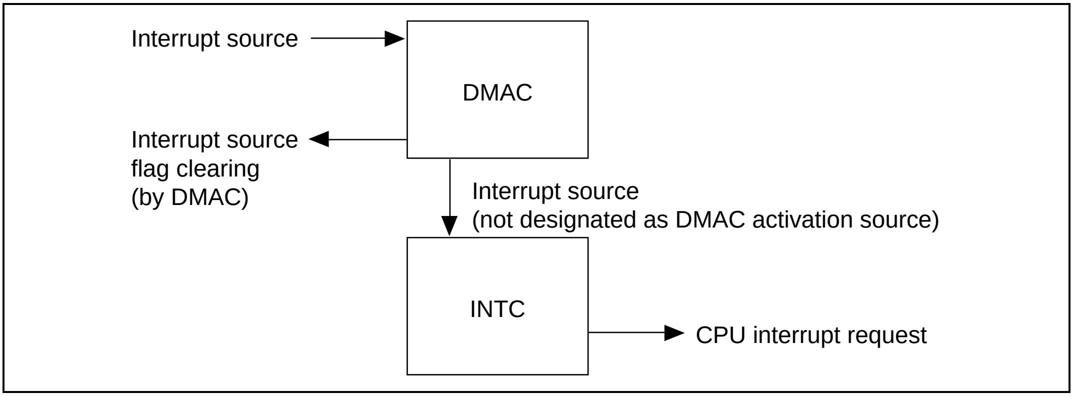

Interrupt source
DMAC
Interrupt source flag clearing (by DMAC)
Interrupt source (not designated as DMAC activation source)
INTC
CPU interrupt request

Figure 6.9 Interrupt Control Block Diagram

6.6.1 To Designate a Source as a DMAC Activation Source, Not a CPU Interrupt Source

1. Select the source in the DMAC, and set DE = 1, DME = 1. The CPU interrupt source is masked regardless of the interrupt priority register settings.
2. When the interrupt is generated, the activation source is sent to the DMAC.
3. The DMAC clears the activation source when it performs transfer.

6.6.2 To Designate a Source as a CPU Interrupt Source, Not a DMAC Activation Source

1. Do not select the source in the DMAC, or clear the DME bit to 0. If the source has been selected in the DMAC, clear the DE bit for the relevant DMAC channel to 0.
2. When the interrupt is generated, an interrupt request is sent to the CPU.
3. The CPU clears the interrupt source and performs the necessary processing in the interrupt service routine.

6.7 Usage Notes

6.7.1 IRQ3 to IRQ0 Sampling and Interrupt Source Determination in IRL Interrupt Mode

Low level sensing or falling edge sensing can be set as the sampling method for each of pins IRQ3 to IRQ0 by means of IRQ sense select bits 3 to 0 in interrupt control register 2 (ICR2).

In IRL interrupt mode, the same sampling method must be set for all four pins, IRQ3 to IRQ0.

When low level sensing is set for IRQ3 to IRQ0, on acceptance of an interrupt request the interrupt source (IRL1 to IRL15) is determined by the levels of IRQ3 to IRQ0.

When falling edge sensing is set for IRQ3 to IRQ0, the falling edge detection results for IRQ3 to IRQ0 are retained. When an interrupt request is accepted, the interrupt source (IRL1 to IRL15) is determined by the retained detection results.

For example, if level 3 (IRQ[3:0] = H'1100) input is not accepted, and this is followed by level 4 (IRQ[3:0] = H'1011) input without clearing the retained detection results, a level 7 (IRQ[3:0] = H'1000) interrupt request will be judged to have been issued on the basis of the detection results retained up to that point. In this example, in order to end up with a level 4 interrupt request, the level 3 detection results must be cleared before the level 4 interrupt signal is input. Detection results can be cleared by reading 1 from bits IRQ3F to IRQ0F in the IRQ status register (ISR), then writing 0 to these bits.

6.7.2 IRQ Pin Noise Cancellation Function

Signals IRQ7 to IRQ0 are sent to the interrupt controller via a noise canceler that eliminates noise of one state or less in duration. Therefore, when edge detection is set for the IRQ pins, the IRQ input must be at least 2.5 states in duration.

Section 7 User Break Controller (UBC)

7.1 Overview

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 conditions of the bus cycle generated by the CPU or on-chip DMAC. This function makes it easy to design an effective self-monitoring debugger, enabling programs to be debugged with the chip alone, without using a large-scale in-circuit emulator.

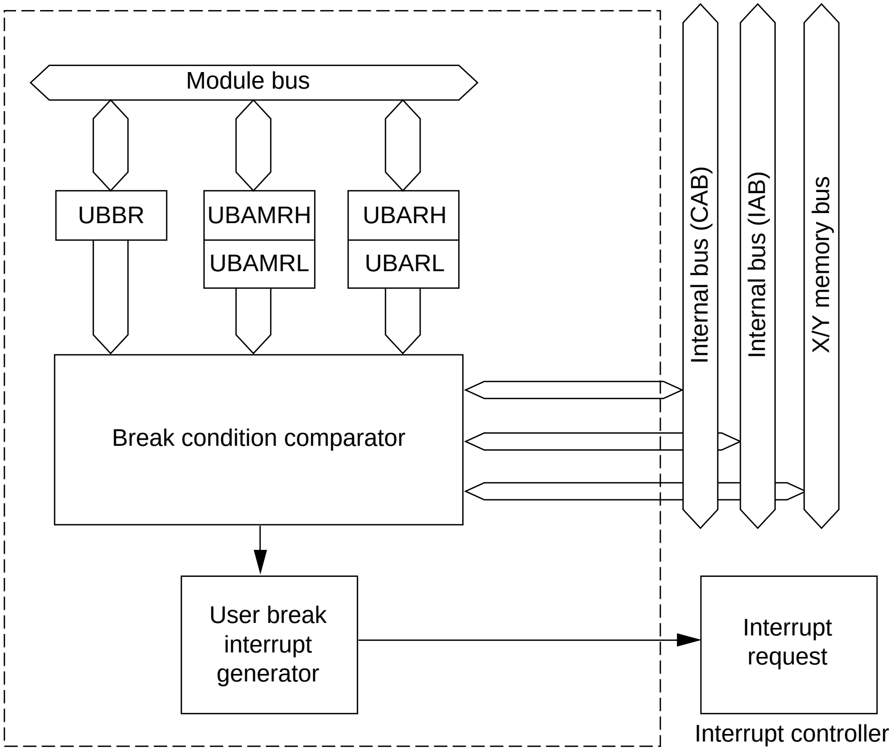
7.1.1 Features

The UBC has the following features:

- The following break conditions can be set:
  - Address (bit masking possible)
    Internal address bus (CAB)/internal address bus (IAB)/X memory address bus (XAB)/Y memory address bus (YAB)
  - Bus master
    CPU cycle/DMA cycle
  - Bus cycle
    Instruction fetch/data access
  - Read/write
  - Operand size
    Byte/word/longword
- User break interrupt generation on occurrence of break condition
  A user-written user break interrupt exception routine can be executed.
- When a user break is set for a CPU instruction fetch, the break is effected before execution of the next instruction (post-execution break).

7.1.2 Block Diagram

Figure 7.1 shows a block diagram of the UBC.



Legend:
UBARH, UBARL: User break address registers H and L
UBAMRH, UBAMRL: User break address mask registers H and L
UBBR: User break bus cycle register

Figure 7.1 Block Diagram of UBC

7.1.3 Register Configuration

The UBC has the five registers shown in table 7.1. These registers are used to set the break conditions.

Table 7.1 UBC Registers

Name	Abbre- viation	R/W	Initial Value	Address	Access Size
User break address register H	UBARH	R/W	H'0000	H'FFFF0C80	16, 32
User break address register L	UBARL	R/W	H'0000	H'FFFF0C82	16, 32
User break address mask register H	UBAMRH	R/W	H'0000	H'FFFF0C84	16, 32
User break address mask register L	UBAMRL	R/W	H'0000	H'FFFF0C86	16, 32
User break bus cycle register	UBBR	R/W	H'0000	H'FFFF0C88	16, 32

7.2 Register Descriptions

7.2.1 User Break Address Register (UBAR)

UBARH

Bit:	15	14	13	12	11	10	9	8
	UBA31	UBA30	UBA29	UBA28	UBA27	UBA26	UBA25	UBA24
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
	UBA23	UBA22	UBA21	UBA20	UBA19	UBA18	UBA17	UBA16
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

UBARL

Bit:	15	14	13	12	11	10	9	8
	UBA15	UBA14	UBA13	UBA12	UBA11	UBA10	UBA9	UBA8
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
	UBA7	UBA6	UBA5	UBA4	UBA3	UBA2	UBA1	UBA0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

The user break address register (UBAR) consists of two 16-bit readable/writable registers: user break address register H (UBARH) and user break address register L (UBARL).

Control bits XYE and XYS in the user break bus cycle register (UBBR) select the break condition address bus. When XYE is 0, the break address is specified on the CAB internal address bus or IAB internal address bus. In this case, UBARH specifies the upper half (bits 31 to 16) of the address used as the break condition, and UBARL specifies the lower half (bits 15 to 0). When XYE is 1, UBARH specifies the break address on X memory address bus XAB (bits 15 to 1), and UBARL specifies the break address on Y memory address bus YAB (bits 15 to 1). As XAB and YAB have only 15 bits, 0 must be set as the least significant bit. When XYE is 1, either XAB or YAB must be selected with the XYS bit in UBBR.

UBARH and UBARL are initialized to H'0000 by a power-on reset and in hardware standby mode. They are not initialized by the module standby function or in software standby mode.

XYE	UBARH	UBARL
0	CAB31–CAB16/IAB31–IAB16	CAB15–CAB0/IAB15–IAB0
1	XAB15–XAB1 (XYS = 0)	YAB15–YAB1 (XYS = 1)

7.2.2 User Break Address Mask Register (UBAMR)

UBAMRH

Bit:	15	14	13	12	11	10	9	8
	UBM31	UBM30	UBM29	UBM28	UBM27	UBM26	UBM25	UBM24
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
	UBM23	UBM22	UBM21	UBM20	UBM19	UBM18	UBM17	UBM16
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

UBAMRL

Bit:	15	14	13	12	11	10	9	8
	UBM15	UBM14	UBM13	UBM12	UBM11	UBM10	UBM9	UBM8
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
	UBM7	UBM6	UBM5	UBM4	UBM3	UBM2	UBM1	UBM0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

The user break address mask register (UBAMR) consists of two 16-bit readable/writable registers: user break address mask register H (UBAMRH) and user break mask address register L (UBAMRL).

Control bits XYE and XYS in the user break bus cycle register (UBBR) select the break condition address bus. When XYE is 0, UBAR specifies a break address on the CAB internal address bus or IAB internal address bus. In this case, UBAMRH specifies which bits of the break address set in UBARH are to be masked, and UBAMRL specifies which bits of the break address set in UBARL are to be masked. When XYE is 1, UBAMRH specifies which bits of the break address on XAB (bits 15 to 1) set in UBARH are to be masked, and UBAMRL specifies which bits of the break address on YAB (bits 15 to 1) set in UBARL are to be masked. As XAB and YAB have only 15 bits, the setting of the least significant bit of UBAMRH and UBAMRL is invalid. When XYE is 1, either XAB or YAB must be selected with the XYS bit in UBBR.

UBAMRH and UBAMRL are initialized to H'0000 by a power-on reset and in hardware standby mode. They are not initialized by the module standby function or in software standby mode.

XYE	UBAMRH	UBAMRL
0	CAB31–16/IAB31–16 masked	CAB15–0/IAB15–0 masked
1	XAB15–1 masked (XYS = 0)	YAB15–1 masked (XYS = 1)

Bits 15 to 0: UBMn	Description	
0	User break address UBAn is included in break conditions	(Initial value)
1	User break address UBAn is not included in break conditions	

Note: n = 31 to 0

7.2.3 User Break Bus Cycle Register (UBBR)

Bit:	15	14	13	12	11	10	9	8
	UBIE	—	—	—	—	—	XYE	XYS
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R	R	R	R	R	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
	CP1	CP0	ID1	ID0	RW1	RW0	SZ1	SZ0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

The user break bus cycle register (UBBR) is a 16-bit readable/writable register that sets five conditions—(1) internal bus (C-bus) or internal bus (I-bus)/X memory bus or Y memory bus, (2) CPU cycle/DMA cycle, (3) instruction fetch/data access, (4) read/write, and (5) operand size (byte/word/longword)—and selects whether or not a user break interrupt is to generated when a condition is matched. UBBR is initialized to H'0000 by a power-on reset and in hardware standby mode. It is not initialized by the module standby function or in software standby mode.

Bit 15—User Break Interrupt Enable (UBIE): Specifies whether or not a user break interrupt is to be generated when the set break condition occurs.

Bit 15: UBIE	Description
0	User break interrupt is not generated when break condition occurs (Initial value)
1	User break interrupt is generated when break condition occurs

Bits 14 to 10—Reserved: These bits are always read as 0, and should only be written with 0.

Bit 9—X/Y Memory Bus Enable (XYE): Selects the C-bus/I-bus or the X/Y memory bus as the break condition bus.

Bit 9: XYE	Description
0	C-bus or I-bus is selected as break condition (Initial value)
1	X memory bus or Y memory bus is selected as break condition

Bit 8—X Memory Bus/Y Memory Bus Select (XYS): Selects the X memory bus or Y memory bus as the break condition bus.

Bit 8: XYS	Description
0	X memory bus is selected as break condition (Initial value)
1	Y memory bus is selected as break condition

Note: When XYE = 0, the setting of bit 8 is ignored.

Bits 7 and 6—CPU Cycle/DMA Cycle Select (CP1, CP0): These bits select the CPU or DMA as the break condition bus master.

Bit 7: CP1	Bit 6: CP0	Description
0	0	User break interrupt is not generated (Initial value)
	1	CPU cycle is selected as break condition
1	0	DMA cycle is selected as break condition
	1	Both CPU cycle and DMA cycle are selected as break conditions

Bits 5 and 4—Instruction Fetch/Data Access Select (ID1, ID0): These bits select an instruction fetch cycle or data access cycle as the break condition bus cycle.

Bit 5: ID1	Bit 4: ID0	Description
0	0	User break interrupt is not generated (Initial value)
	1	Instruction fetch cycle is selected as break condition
1	0	Data access cycle is selected as break condition
	1	Both instruction fetch cycle and data access cycle are selected as break conditions

Bits 3 and 2—Read/Write Select (RW1, RW0): These bits select a read cycle or write cycle as the break condition access.

Bit 3: RW1	Bit 2: RW0	Description
0	0	User break interrupt is not generated (Initial value)
	1	Read cycle is selected as break condition
1	0	Write cycle is selected as break condition
	1	Both read cycle and write cycle are selected as break conditions

Bits 1 and 0—Operand Size Select (SZ1, SZ0): These bits select the operand size of the break condition bus cycle.

Bit 1: SZ1	Bit 0: SZ0	Description
0	0	Operand size is not included in break conditions (Initial value)
	1	Byte access is selected as break condition
1	0	Word access is selected as break condition
	1	Longword access is selected as break condition

7.3 Operation

7.3.1 User Break Operation Sequence

The sequence of operations from setting of break conditions to user break interrupt exception handling is described below.

1. As break conditions, set the user break address in the user break address register (UBAR), the address bits to be masked in the user break address mask register (UBAMR), and the type of bus cycle on which a break is to be executed in the user break bus cycle register (UBBR). If any pair of bits from among the CPU cycle/DMA cycle select bits (CP1, CP0), instruction fetch/data access select bits (ID1, ID0), or read/write select bits (RW1, RW0) in UBBR is set to 00 (user break interrupt not generated), a user break interrupt will not be generated even if other conditions are satisfied. If the user break interrupt is to be used, a condition must be set in all of these bit pairs.

2. If the user break interrupt enable bit (UBIE) in the user break bus cycle register (UBBR) is set to 1 when a break condition occurs, the UBC sends a user break interrupt request signal to the interrupt controller (INTC).

3. When the INTC receives the user break interrupt request signal, it determines its priority. As the priority level of the user break interrupt is 15, it is accepted if the level set in the interrupt mask bits (I3 to I0) in the status register (SR) is 14 or less. If the level set in bits I3 to I0 is 15, the user break interrupt is not accepted, but is held pending until it can be. As the setting of bits I3 to I0 is 15 during NMI exception handling, a user break interrupt is not accepted during execution of the NMI exception service routine. However, changing the setting of bits I3 to I0 to level 14 or below at the start of the NMI exception service routine will enable subsequent user break interrupts to be accepted. For details of priority determination, see section 6, Interrupt Controller (INTC).

4. The INTC sends a user break interrupt request signal to the CPU. On receiving this signal, the CPU begins user break interrupt exception handling. For details of interrupt exception handling, see section 5, Exception Handling, and section 6.4, Operation.

7.3.2 Instruction Fetch Cycle Break

When an internal bus (C-bus)/CPU/instruction fetch/read setting is made in the user break bus cycle register (UBBR), a CPU instruction fetch cycle can be selected as a user break condition. In this case, an operand size setting is not necessary.

When a user break condition occurs, the instruction set in the user break conditions is executed, and an interrupt is generated before execution of the next instruction. Therefore, a user break cannot be specified for an instruction that is fetched but not executed, such as an overrun fetch instruction. However, if a user break condition is set for a delayed branch instruction or an interrupt-disabled instruction such as LDC, an interrupt will be generated before execution of the next instruction at which the interrupt can be accepted.

When an instruction fetch cycle is set as a user break condition, the start address at which that instruction is located should be set as the user break address. A user break will not occur if a different address is set. Therefore, if the address of the lower word of a 32-bit instruction is set as a user break condition, a user break will not occur.

7.3.3 Data Access Cycle Break

Memory cycles subject to a CPU data access break are memory cycles due to instructions and stack operations and vector reads when exception handling is executed. Table 7.2 shows the bits of the user break address register and the address bus that are compared for each operand size to determine whether a break condition has been matched.

Table 7.2 Data Access Cycle Address and Operand Size Comparison Conditions

Access Size	Compared Address Bits
Longword	Bits 31–2 of break address register compared with bits 31–2 of address bus
Word	Bits 31–1 of break address register compared with bits 31–1 of address bus
Byte	Bits 31–0 of break address register compared with bits 31–0 of address bus

This means, for example, that if address H'00001003 is set without specifying an operand size condition (i.e. the operand size select bits in the user break bus cycle register are set to 00), bus cycles that satisfy the break conditions include the following(assuming that all other conditions are satisfied):

- Longword access at H'00001000
- Word access at H'00001002
- Byte access at H'00001003

7.3.4 X Memory Bus or Y Memory Bus Cycle Break

When XYE is set to 1 in UBBR, break addresses on the X memory bus or Y memory bus are selected. Either the X memory bus or the Y memory bus must be selected with the XYS bit in UBBR; the X and Y memory buses cannot both be included in the break conditions at the same time. The break conditions are applied to X memory bus cycles or Y memory bus cycles by specifying the CPU bus master, data access cycle, read or write access, and word operand size or operand size not included.

When the X memory address bus is selected as a break condition, specify the X memory address in UBARH and UBAMRH; when the Y memory address bus is selected, specify the Y memory address in UBARL and UBAMRL.

7.3.5 Program Counter (PC) Value Saved

When Instruction Fetch is Set as User Break Condition

The program counter (PC) value saved in user break interrupt exception handling is the address of the instruction to be executed after the instruction at which the user break condition was satisfied. The instruction at which the break condition was satisfied is executed, and a user break interrupt is generated before execution of the next instruction. However, when a user break condition is set for a delayed branch instruction, the delay slot instruction is executed, and the user break interrupt is generated before execution of the branch instruction. In this case, the PC value is the address of the branch destination instruction.

When Data Access (CPU/DMA) is Set as User Break Condition

The address saved is the start address of the instruction following the instruction for which execution has been completed at the point at which user break exception handling starts. When a data access (CPU/DMA) is set as a user break condition, it is not possible to specify where the break will occur. The break will be effected at an instruction about to be fetched close to where the break data access occurred.

7.4 Examples of Use

CPU Instruction Fetch Cycle Break Condition Settings

Example of Valid Settings:

- Register settings

  UBARH = H'0000, UBARL = H'0404

  UBAMRH = H'0000, UBAMRL = H'0000

  UBBR = H'8054
- Set conditions

  Address: H'00000404; address mask: H'00000000

  Bus cycle: CPU, instruction fetch, read (operand size not included)

A user break interrupt is generated after execution of the instruction at address H'00000404.

Example of Invalid Settings:

- Register settings

  UBARH = H'0015, UBARL = H'389C

  UBAMRH = H'0000, UBAMRL = H'0000

  UBBR = H'8058
- Set conditions

  Address: H'0015389C; address mask: H'00000000

  Bus cycle: CPU, instruction fetch, write (operand size not included)

As an instruction fetch cycle is not a write cycle, a user break interrupt is not generated.

CPU Data Access Cycle (Internal Bus (C-Bus) Cycle) Break Condition Settings

Example of Valid Settings (1):

- Register settings

  UBARH = H'0012, UBARL = H'3456

  UBAMRH = H'0000, UBAMRL = H'0000

  UBBR = H'806A
- Set conditions

  Address: H'00123456; address mask: H'00000000

  Bus cycle: CPU, data access, write, word

A user break interrupt is generated when word data is written to address H'00123456.

Example of Valid Settings (2):

- Register settings

  UBARH = H'00A8, UBARL = H'3901

  UBAMRH = H'0000, UBAMRL = H'0000

  UBBR = H'8066
- Set conditions

  Address: H'00A80391; address mask: H'00000000

  Bus cycle: CPU, data access, read, word

As a word access is performed on an even address, user break interrupt exception handling is performed after address error exception handling.

Example of Invalid Settings:

- Register settings

  UBARH = H'0034, UBARL = H'5024

  UBAMRH = H'0000, UBAMRL = H'0000

  UBBR = H'8062
- Set conditions

  Address: H'00345024; address mask: H'00000000

  Bus cycle: CPU, data access, —, word

As the access type has not been set as either read or write, a user break interrupt is not generated.

DMA Cycle Break Condition Settings

Example of Valid Settings:

- Register settings

  UBARH = H'0076, UBARL = H'BCDC

  UBAMRH = H'0000, UBAMRL = H'0000

  UBBR = H'80A7
- Set conditions

  Address: H'0076BCDC; address mask: H'00000000

  Bus cycle: DMA, data access, longword

A user break interrupt is generated when longword data is read from address H'0076BCDC.

Example of Invalid Settings:

- Register settings

  UBARH = H'0023, UBARL = H'45C8

  UBAMRH = H'0000, UBAMRL = H'0000

  UBBR = H'8094
- Set conditions

  Address: H'002345C8; address mask: H'00000000

  Bus cycle: DMA, instruction fetch, write (operand size not included)

As an instruction fetch is not performed in a DMA cycle, a user break interrupt is not generated.

CPU Data Access Cycle (X/Y Memory Bus Cycle) Break Condition Settings

Example of Valid Settings:

- Register settings

  UBARH = H'8000, UBARL = H'0000

  UBAMRH = H'0000, UBAMRL = H'0000

  UBBR = H'826A
- Set conditions

  Address: H'FFFF8000; address mask: H'00000000

  Bus cycle: CPU, data access (X memory access using X memory bus), write, word

A user break access is generated when word data is written to address H'FFFF8000 in X memory space using the X memory bus.

Example of Invalid Settings:

- Register settings

  UBARH = H'A000, UBARL = H'0000

  UBAMRH = H'0000, UBAMRL = H'0000

  UBBR = H'826B
- Set conditions

  Address: H'FFFFA000; address mask: H'00000000

  Bus cycle: CPU, data access (Y memory access using Y memory bus), write, byte

As byte access cannot be performed in a data access cycle using the X/Y memory bus, a user break interrupt is not generated.

7.5 Usage Notes

7.5.1 Changes to UBC Register Settings

UBC register reads and writes are executed in the MA (memory access) stage of the instruction pipeline, and checks are not carried out based on new user break conditions until register changes have been completed. A user break based on new break conditions will not occur until register setting changes have been completed. Thus, if the check stage of the succeeding instruction occurs before new user break conditions are written to the UBC registers, a user break will not occur even if the checked address matches the user break condition.

7.5.2 Repeat Condition Breaks

If repeated execution of a repeat instruction is included as a break condition, note that a user break will not occur if a user break condition is set for an instruction being executed repeatedly during execution of a repeat loop consisting of no more than three instructions.

Section 8 Bus State Controller (BSC)

8.1 Overview

The functions of the bus state controller (BSC) include division of the address space and output of control signals for various types of memory. The BSC functions allow DRAM, EDO DRAM, SRAM, ROM, etc., to be connected directly to the SH7065 without the use of external circuitry.

8.1.1 Features

The BSC has the following features:

- Address space is managed as six independent areas
  - CS0 space: Maximum linear 48 Mbytes in on-chip ROM enabled modes, and maximum 64 Mbytes in on-chip ROM disabled modes
  - CS1 to CS3 spaces: Maximum linear 64 Mbytes for each
  - CS4, CS5 spaces: Dedicated DRAM spaces, maximum linear 64 Mbytes
  - Memory types (DRAM, EDO DRAM, SRAM, ROM, etc.) can be specified individually for each space
  - Bus width (8, 16, or 32 bits) can be selected for each space

    (Settable by external pin for CS0 space only)
  - Wait states can be inserted by software for each space
  - Wait states can be inserted by the $\overline{\text{WAIT}}$ pin when accessing external memory space
  - Output of appropriate control signals for memory connected to each space
  - Automatic wait cycle insertion to prevent data bus collisions in case of consecutive memory accesses to different CS spaces, or a read access followed by a write access to the same area
  - Big-endian or little-endian mode can be set for each space
- Direct DRAM interface
  - Row address/column address multiplexed output according to DRAM capacity
  - Burst operation supported (fast page mode, EDO mode, RAS down mode)
  - Precharge cycle generated to secure RAS precharge interval
- Access control for various kinds of memory and peripheral chips
  - Address/data multiplexing

- Refresh functions
  - Supports CAS-before-RAS refreshing and self-refreshing
  - Supports refresh operation immediately after self-refresh operation in low-power DRAM by means of refresh counter overflow interrupt function
  - Up to 8 consecutive CAS-before-RAS refreshes
- Refresh counter can be used as interval timer
  - Interrupt request generated by compare match
  - Interrupt request generated by refresh counter overflow

8.1.2 Block Diagram

Figure 8.1 shows a block diagram of the BSC.

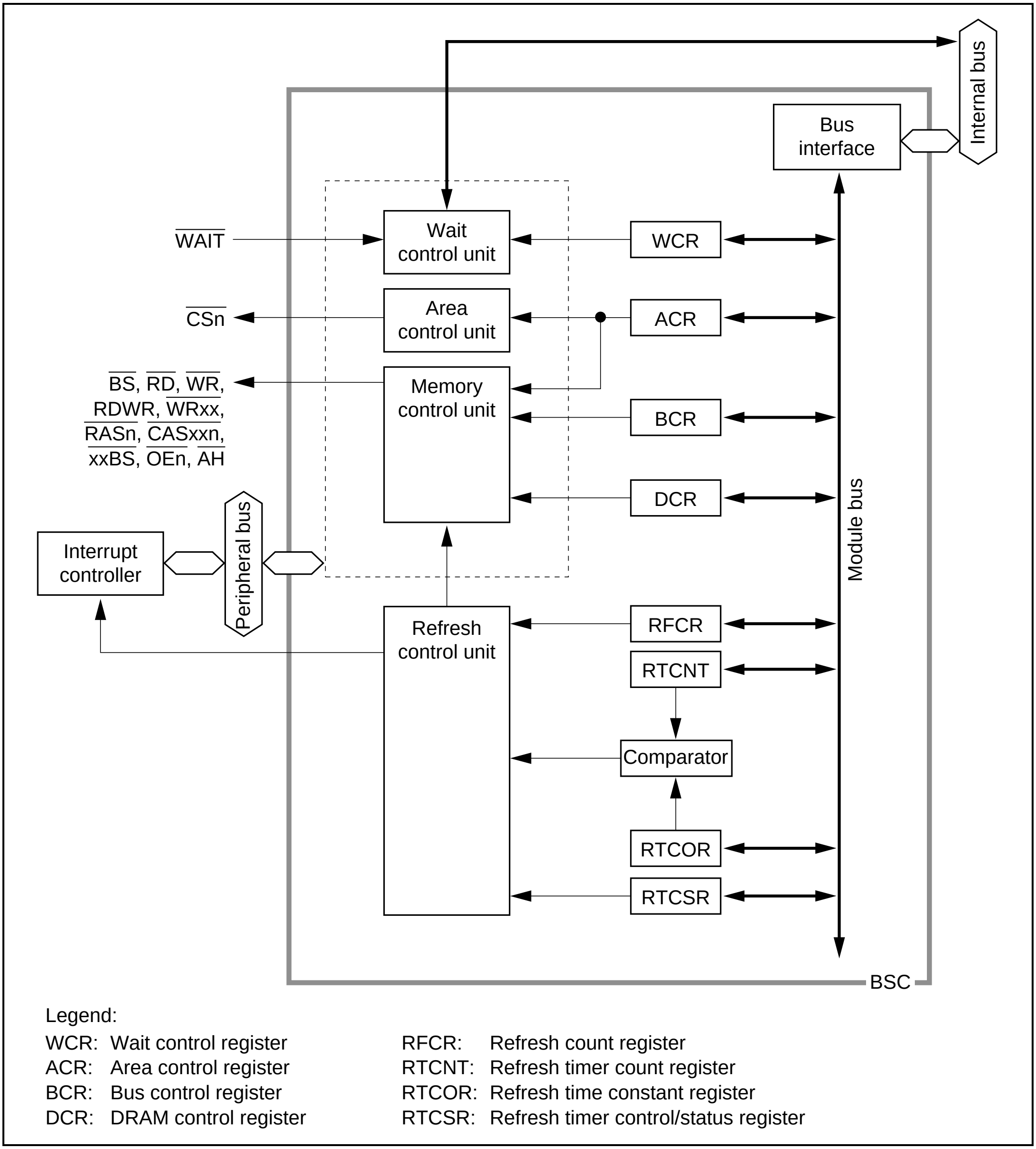

Internal bus
Bus interface
Wait control unit
Area control unit
Memory control unit
Refresh control unit
WAIT
CSn
BS, RD, WR, RDWR, WRxx, RASn, CASxxn, xxBS, OEn, AH
Interrupt controller
Peripheral bus
WCR
ACR
BCR
DCR
RFCR
RTCNT
Comparator
RTCOR
RTCSR
Module bus
BSC
Legend:
WCR: Wait control register
ACR: Area control register
BCR: Bus control register
DCR: DRAM control register
RFCR: Refresh count register
RTCNT: Refresh timer count register
RTCOR: Refresh time constant register
RTCSR: Refresh timer control/status register

Figure 8.1 Block Diagram of BSC

8.1.3 Pin Configuration

Table 8.1 shows the BSC pin configuration.

Table 8.1 BSC Pins

Name	Signals	I/O	Function
Address bus	A25–A0	Output	Address output
Data bus	D31–D0	I/O	Data input/output
Bus cycle start	$\overline{\text{BS}}$	Output	Signal that indicates the start of a bus cycle. In burst transfer, asserted each data cycle.
Chip select	$\overline{\text{CS5}}$–$\overline{\text{CS0}}$	Output	Chip select signals indicating the area being accessed
Read	$\overline{\text{RD}}$	Output	Strobe signal indicating a read cycle
Write LL	$\overline{\text{WRLL}}$	Output	Strobe signal indicating a D7–D0 write cycle
Write LH	$\overline{\text{WRLH}}$	Output	Strobe signal indicating a D15–D8 write cycle
Write HL	$\overline{\text{WRHL}}$	Output	Strobe signal indicating a D23–D16 write cycle
Write HH	$\overline{\text{WRHH}}$	Output	Strobe signal indicating a D31–D24 write cycle
Write strobe LL	$\overline{\text{LLBS}}$	Output	Strobe signal indicating access to D7–D0
Write strobe LH	$\overline{\text{LHBS}}$	Output	Strobe signal indicating access to D15–D8
Write strobe HL	$\overline{\text{HLBS}}$	Output	Strobe signal indicating access to D23–D16
Write strobe HH	$\overline{\text{HHBS}}$	Output	Strobe signal indicating access to D31–D24
Write	$\overline{\text{WR}}$	Output	Data bus input/output direction designation signal. Used as write designation signal for byte-strobe memory.
Read/write	RDWR	Output	DRAM/EDO DRAM write designation signal
Row address strobe	$\overline{\text{RAS1}}$, $\overline{\text{RAS0}}$	Output	RAS signals for DRAM connected to areas 5 and 4
Column address strobe LL	$\overline{\text{CASLL1}}$, $\overline{\text{CASLL0}}$	Output	D7–D0 CAS signals for DRAM connected to areas 5 and 4
Column address strobe LH	$\overline{\text{CASLH1}}$, $\overline{\text{CASLH0}}$	Output	D15–D8 CAS signals for DRAM connected to areas 5 and 4
Column address strobe HL	$\overline{\text{CASHL1}}$, $\overline{\text{CASHL0}}$	Output	D23–D16 CAS signals for DRAM connected to areas 5 and 4
Column address strobe HH	$\overline{\text{CASHH1}}$, $\overline{\text{CASHH0}}$	Output	D31–D24 CAS signals for DRAM connected to areas 5 and 4

Name	Signals	I/O	Function
Output enable	$\overline{\text{OE1}}$, $\overline{\text{OE0}}$	Output	Output enable signals for EDO DRAM connected to areas 5 and 4. Used for access in RAS down mode.
Address hold	$\overline{\text{AH}}$	Output	Signal for holding address in address/data multiplexing
Wait	$\overline{\text{WAIT}}$	Input	Wait state request signal
Bus release request	$\overline{\text{BREQ}}$	Input	Bus release request signal
Bus request acknowledge	$\overline{\text{BACK}}$	Output	Signal granting use of the bus

8.1.4 Register Configuration

The BSC has the 18 registers shown in table 8.2. The functions of these registers include control of direct interfaces to various types of memory, wait states, and refreshing.

Table 8.2 BSC Registers

Name	Abbre- viation	R/W	Initial Value	Address	Access Size
Bus control register	BCR	R/W	H'0000	H'FFFF 0C00	8, 16, 32
Area control register 1 (for area 0)	ACR1_0	R/W	H'07FF	H'FFFF 0C10	8, 16, 32
Area control register 1 (for area 1)	ACR1_1	R/W	H'07FF	H'FFFF 0C12	8, 16, 32
Area control register 1 (for area 2)	ACR1_2	R/W	H'07FF	H'FFFF 0C14	8, 16, 32
Area control register 1 (for area 3)	ACR1_3	R/W	H'07FF	H'FFFF 0C16	8, 16, 32
Area control register 1 (for area 4)	ACR1_4	R/W	H'0000	H'FFFF 0C20	8, 16, 32
Area control register 1 (for area 5)	ACR1_5	R/W	H'0000	H'FFFF 0C22	8, 16, 32
Wait control register (for area 0)	WCR_0	R/W	H'FFFE	H'FFFF 0C30	8, 16, 32
Wait control register (for area 1)	WCR_1	R/W	H'FFFE	H'FFFF 0C32	8, 16, 32
Wait control register (for area 2)	WCR_2	R/W	H'FFFE	H'FFFF 0C34	8, 16, 32
Wait control register (for area 3)	WCR_3	R/W	H'FFFE	H'FFFF 0C36	8, 16, 32
DRAM control register 1	DCR1	R/W	H'0000	H'FFFF 0C40	8, 16, 32
DRAM control register 2	DCR2	R/W	H'1FE0	H'FFFF 0C42	8, 16, 32
DRAM control register 3	DCR3	R/W	H'1800	H'FFFF 0C44	8, 16, 32
Refresh timer control/status register	RTCSR	R/W	H'0000	H'FFFF 0C68	8, 16, 32
Refresh timer counter	RTCNT	R/W	H'0000	H'FFFF 0C6A	8, 16, 32
Refresh time constant counter	RTCOR	R/W	H'0000	H'FFFF 0C6C	8, 16, 32
Refresh count register	RFCR	R/W	H'0000	H'FFFF 0C6E	8, 16, 32

8.1.5 Address Format

Figure 8.2 shows the address format used in the SH7065.

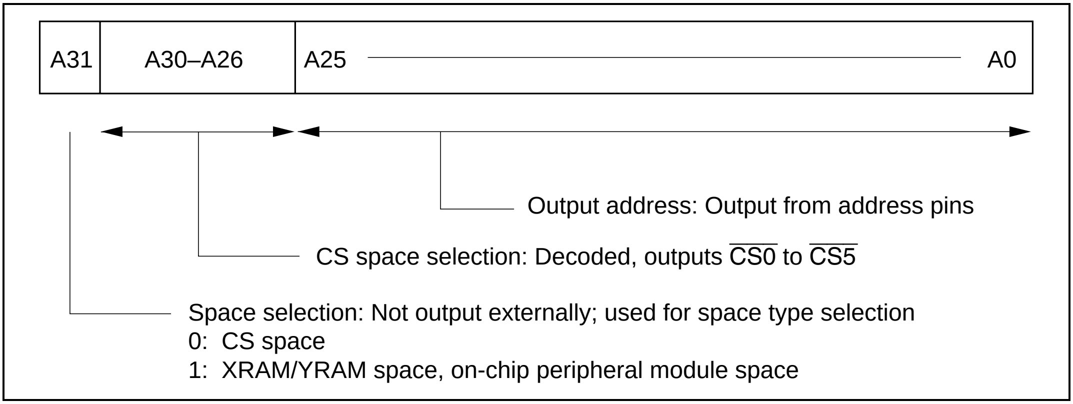

A31
A30–A26
A25
A0
Output address: Output from address pins
CS space selection: Decoded, outputs $\overline{\text{CS0}}$ to $\overline{\text{CS5}}$
Space selection: Not output externally; used for space type selection
0: CS space
1: XRAM/YRAM space, on-chip peripheral module space

Figure 8.2 Address Format

The SH7065 uses 32-bit addresses.

Bit A31 is used to select the type of space. This signal is not output externally.

Bits A30 to A26 are decoded to provide the chip select signal ($\overline{\text{CS0}}$ to $\overline{\text{CS5}}$) for the area, which is output.

Bits A25 to A0 are output externally.

Table 8.3 shows the address maps when the maximum range is set for each space.

Table 8.3 Address Maps

- In on-chip ROM disabled modes

Addresses	Type of Space	Type of Memory	Size	Bus Width
H'0000 0000 to H'03FF FFFF	CS0 space	Normal space	64 MB	8/16/32 bits
H'0400 0000 to H'07FF FFFF	CS1 space	Normal space/ multiplexed I/O space	64 MB	8/16/32 bits
H'0800 0000 to H'0BFF FFFF	CS2 space		64 MB	8/16/32 bits
H'0C00 0000 to H'0FFF FFFF	CS3 space		64 MB	8/16/32 bits
H'1000 0000 to H'3FFF FFFF	Reserved	Reserved		
H'4000 0000 to H'43FF FFFF	CS4 space	DRAM	64 MB	8/16/32 bits
H'4400 0000 to H'47FF FFFF	CS5 space		64 MB	8/16/32 bits
H'4800 0000 to H'57FF FFFF	Reserved	Reserved		
H'5800 0000 to H'5803 FFFF	On-chip ROM*	On-chip ROM*	256 kB	32 bits
H'5804 0000 to H'FFFE FFFF	Reserved	Reserved		
H'FFFF 0000 to H'FFFF 13FF	On-chip peripheral module	On-chip peripheral module	5 kB	8/16 bits
H'FFFF 1400 to H'FFFF 7FFF	Reserved	Reserved		
H'FFFF 8000 to H'FFFF 8FFF	XRAM	XRAM	4 kB	32 bits
H'FFFF 9000 to H'FFFF 9FFF	Reserved	Reserved		
H'FFFF A000 to H'FFFF AFFF	YRAM	YRAM	4 kB	32 bits
H'FFFF B000 to H'FFFF FFFF	Reserved	Reserved		

Note: * In this mode, the power-on reset vector table is located in the CS0 space (external space). Also, addresses H'5800 0000 to H'5803 FFFF can be used as on-chip ROM.

- In on-chip ROM enabled modes

Addresses	Type of Space	Type of Memory	Size	Bus Width
H'0000 0000 to H'0003 FFFF	On-chip ROM	On-chip ROM	256 kB	32 bits
H'0004 0000 to H'00FF FFFF	Reserved	Reserved		
H'0100 0000 to H'03FF FFFF	CS0 space	Normal space	48 MB	8/16/32 bits
H'0400 0000 to H'07FF FFFF	CS1 space	Normal space/ multiplexed I/O space	64 MB	8/16/32 bits
H'0800 0000 to H'0BFF FFFF	CS2 space		64 MB	8/16/32 bits
H'0C00 0000 to H'0FFF FFFF	CS3 space		64 MB	8/16/32 bits
H'1000 0000 to H'3FFF FFFF	Reserved	Reserved		
H'4000 0000 to H'43FF FFFF	CS4 space	DRAM	64 MB	8/16/32 bits
H'4400 0000 to H'47FF FFFF	CS5 space		64 MB	8/16/32 bits
H'4800 0000 to H'57FF FFFF	Reserved	Reserved		
H'5800 0000 to H'5803 FFFF	On-chip ROM*	On-chip ROM*	256 kB	32 bits
H'5804 0000 to H'FFFE FFFF	Reserved	Reserved		
H'FFFF 0000 to H'FFFF 13FF	On-chip peripheral module	On-chip peripheral module	5 kB	8/16 bits
H'FFFF 1400 to H'FFFF 7FFF	Reserved	Reserved		
H'FFFF 8000 to H'FFFF 8FFF	XRAM	XRAM	4 kB	32 bits
H'FFFF 9000 to H'FFFF 9FFF	Reserved	Reserved		
H'FFFF A000 to H'FFFF AFFF	YRAM	YRAM	4 kB	32 bits
H'FFFF B000 to H'FFFF FFFF	Reserved	Reserved		

Note: * The same data as in on-chip ROM addresses H'0000 0000 to H'0003 FFFF can be read.

8.2 Register Descriptions

8.2.1 Bus Control Register (BCR)

The bus control register (BCR) is a 16-bit readable/writable register that specifies bus settings common to all areas.

BCR is initialized to H'0000 by a power-on reset, but is not initialized in standby mode.

Bit:	15	14	13	12	11	10	9	8
	BRQE	BAS	HIZCNT	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R	R	R	R	R

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

Bit 15—BREQ Enable (BRQE): Enables or disables acceptance of the bus release request (BREQ).

Bit 15: BRQE	Description	
0	Bus release request (BREQ) is not accepted	(Initial value)
1	Bus release request (BREQ) is accepted	

Bit 14—Byte Access Specification (BAS): Specifies the byte access control signals.

Bit 14: BAS	Description	
0	Access by $\overline{\text{WRHH}}$, $\overline{\text{WRHL}}$, $\overline{\text{WRLH}}$, and $\overline{\text{WRLL}}$ signals	(Initial value)
1	Access by $\overline{\text{WR}}$, $\overline{\text{HHBS}}$, $\overline{\text{HLBS}}$, $\overline{\text{LHBS}}$, and $\overline{\text{LLBS}}$ signals	

Bit 13—High-Impedance Control (HIZCNT): Specifies the state of the $\overline{\text{RAS}}$, $\overline{\text{CAS}}$, and $\overline{\text{OE}}$ signals (which control the DRAM self-refresh status) in standby mode and when the bus is released. This enables the DRAM to be kept in the self-refresh state.

Bit 13: HIZCNT	Description
0	The $\overline{\text{RAS}}$, $\overline{\text{CAS}}$, and $\overline{\text{OE}}$ signals go to the high-impedance (Hi-Z) state in standby mode and when the bus is released (Initial value)
1	The $\overline{\text{RAS}}$, $\overline{\text{CAS}}$, and $\overline{\text{OE}}$ signals drive in standby mode and when the bus is released

Bits 12 to 0—Reserved: These bits are always read as 0 and should only be written with 0.

8.2.2 Area Control Registers 1 (ACR1_0 to ACR1_5)

The ACR1 registers are 16-bit readable/writable registers that specify the type of memory to be connected to each area, acceptance of external waits, bus width, number of idle cycles, and number of CS expansion cycles.

The ACR1 registers are initialized to H'07FF (ACR1_0 to ACR1_3 for areas 0 to 3) or H'0000 (ACR1_4 and ACR1_5 for areas 4 and 5) by a power-on reset, but are not initialized in standby mode.

Registers ACR1_0 to ACR1_3 (forAreas 0 to 3)

Bit:	15	14	13	12	11	10	9	8
	ENDIAN	TP1	TP0	EXWE	—	SZ1	SZ0	IW2
Initial value:	0	0	0	0	0	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
	IW1	IW0	SWH2	SWH1	SHW0	SWT2	SWT1	SWT0
Initial value:	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 15—Endian Specification (ENDIAN): Specifies the endian mode for each area.

Bit 15: ENDIAN	Description
0	Big-endian mode (Initial value)
1	Little-endian mode

Bits 14 and 13—Memory Specification (TP1, TP0): These bits specify the type of memory or I/O connected to each area.

Bit 14: TP1	Bit 13: TP0	Description
0	0	Access as normal space (Initial value)
	1	Reserved (Do not set)
1	0	Access as multiplexed address/data I/O space
	1	Reserved (Do not set)

Note: Area 0 is always normal space. For this space, these bits are always read as 0 and should only be written with 0.

Bit 12—External Wait Enable (EXWE): Specifies for each area whether or not wait requests via the external $\overline{\text{WAIT}}$ pin are to be accepted.

Bit 12: EXWE	Description
0	External wait requests are accepted (Initial value)
1	External wait requests are not accepted

Bit 11—Reserved: This bit is always read as 0 and should only be written with 0.

Bits 10 and 9—Bus Width Specification (SZ1, SZ0): These bits specify the bus width of each area.

Bit 10: SZ1	Bit 9: SZ0	Description
0	0	Reserved (Do not set)
	1	8 bits
1	0	16 bits
	1	32 bits (Initial value)

Note: In ROMless expanded mode, the bus width of the CS0 space is set by pins MD0 and MD1. For details, see section 8.3.2, Areas.

Bits 8 to 6—Inter-Cycle Idle Specification (IW2 to IW0): These bits specify the number of idle cycles between bus cycles to be inserted for each area when switching to a different space, or from read access to write access in the same space. The idle cycle specification for the area accessed immediately before is valid. When switching to access to a different space, one idle cycle is inserted automatically in the case of a read cycle, and two idle cycles in the case of a write cycle, even if “No idle cycles” is set. When switching from read cycle to write cycle in the same space, two idle cycles are inserted automatically even if “No idle cycles” is set.

Bit 8: IW2	Bit 7: IW1	Bit 6: IW0	Description
0	0	0	No idle cycles
		1	1 idle cycle inserted
	1	0	2 idle cycles inserted
:	:	:	:
1	1	0	6 idle cycles inserted
		1	7 idle cycles inserted (Initial value)

Bits 5 to 3—Extension Cycles after $\overline{\text{CS}}$ Assertion (SWH2 to SWH0): These bits specify, for each area, the number of cycles to be inserted between assertion of the $\overline{\text{CS}}$ signal and assertion of the $\overline{\text{RD}}$ signal or $\overline{\text{WR}}$ signal.

Bit 5: SWH2	Bit 4: SWH1	Bit 3: SWH0	Description
0	0	0	No extension cycles
		1	1 extension cycle inserted
	1	0	2 extension cycles inserted
:	:	:	:
1	1	0	6 extension cycles inserted
		1	7 extension cycles inserted (Initial value)

Bits 2 to 0—Extension Cycles before $\overline{\text{CS}}$ Negation (SWH2 to SWH0): These bits specify, for each area, the number of cycles to be inserted between negation of the $\overline{\text{RD}}$ signal or $\overline{\text{WR}}$ signal and negation of the $\overline{\text{CS}}$ signal.

Bit 2: SWT2	Bit 1: SWT1	Bit 0: SWT0	Description
0	0	0	No extension cycles
		1	1 extension cycle inserted
	1	0	2 extension cycles inserted
:	:	:	:
1	1	0	6 extension cycles inserted
		1	7 extension cycles inserted (Initial value)

Registers ACR1_4 and ACR1_5 (for Areas 4 and 5)

Bit:	15	14	13	12	11	10	9	8
	ENDIAN	—	—	EXWE	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R	R	R/W	R	R	R	R

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

Bit 15—Endian Specification (ENDIAN): Specifies the endian mode for each area.

Bit 15: ENDIAN	Description
0	Big-endian mode (Initial value)
1	Little-endian mode

Bits 14 and 13—Reserved: These bits are always read as 0 and should only be written with 0.

Bit 12—External Wait Enable (EXWE): Specifies for each area whether or not wait requests via the external $\overline{\text{WAIT}}$ pin are to be accepted.

Bit 12: EXWE	Description
0	External wait requests are accepted (Initial value)
1	External wait requests are not accepted

Bits 11 to 0—Reserved: These bits are always read as 0 and should only be written with 0.

8.2.3 Wait Control Registers (WCR_0 to WCR_3)

The wait control registers (WCR) are 16-bit readable/writable registers that specify the number of wait state cycles to be inserted in areas 0 to 3.

The WCR registers are initialized to H'FFFE by a power-on reset, but are not initialized in standby mode.

Bit:	15	14	13	12	11	10	9	8
	W3	W2	W1	W0	DSWW3	DSWW2	DSWW1	DSWW0
Initial value:	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
	DSWR3	DSWR2	DSWR1	DSWR0	HWW2	HWW1	HWW0	—
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

Bits 15 to 12—Wait State Insertion Cycle Specification (W3 to W0): These bits specify the number of wait states to be inserted in areas 0 to 3.

Bit 15: W3	Bit 14: W2	Bit 13: W1	Bit 12: W0	Description
0	0	0	0	No waits
			1	1 wait
		1	0	2 waits
:	:	:	:	:
1	1	1	0	14 waits
			1	15 waits (Initial value)

Bits 11 to 8—CS0 to CS3 Space DMA Single Address Mode Write Access Wait State Insertion Cycle Specification (DSWW3 to DSWW0): These bits specify the number of wait states to be inserted in writes to spaces CS0 to CS3 in DMA single address mode.

Bit 11: DSWW3	Bit 10: DSWW2	Bit 9: DSWW1	Bit 8: DSWW0	Description
0	0	0	0	No waits
			1	1 wait
		1	0	2 waits
:	:	:	:	:
1	1	1	0	14 waits
			1	15 waits (Initial value)

Bits 7 to 4—CS0 to CS3 Space DMA Single Address Mode Read Access Wait State Insertion Cycle Specification (DSWR3 to DSWR0): These bits specify the number of wait states to be inserted in reads from spaces CS0 to CS3 in DMA single address mode.

Bit 7: DSWR3	Bit 6: DSWR2	Bit 5: DSWR1	Bit 4: DSWR0	Description
0	0	0	0	No waits
			1	1 wait
		1	0	2 waits
:	:	:	:	:
1	1	1	0	14 waits
			1	15 waits (Initial value)

Bits 3 to 1—Wait State Insertion Cycles after External $\overline{\text{WAIT}}$ Pin Negation (HWW2 to HWW0): These bits specify the number of wait states to be inserted after external $\overline{\text{WAIT}}$ pin negation in areas 0 to 3. This specification is valid only when a hard wait is inserted by means of the external $\overline{\text{WAIT}}$ pin. If a hard wait is not inserted, the wait states specified by these bits will not be inserted.

Bit 3: HWW2	Bit 2: HWW1	Bit 1: HWW0	Description
0	0	0	No waits
		1	1 wait
	1	0	2 waits
:	:	:	:
1	1	0	6 waits
		1	7 waits (Initial value)

Bit 0—Reserved: This bit is always read as 0 and should only be written with 0.

8.2.4 DRAM Control Register 1 (DCR1)

DRAM control register 1 (DCR1) is a 16-bit readable/writable register that specifies DRAM control. Control is the same for CS4 and CS5 space accesses.

DCR1 is initialized to H'0000 by a power-on reset, but is not initialized in standby mode.

Bit:	15	14	13	12	11	10	9	8
	TPC1	TPC0	TPCS2	TPCS1	TPCS0	RCD2	RCD1	RCD0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
	—	—	DWW1	DWW0	DWR1	DWR0	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R/W	R/W	R	R

Bits 15 and 14—RAS Precharge Interval Specification (TPC1, TPC0): These bits specify, for DRAM, the minimum number of cycles before RAS is asserted again after being negated.

Bit 15: TPC1	Bit 14: TPC0	Description	
0	0	1 cycle	(Initial value)
	1	2 cycles	
1	0	3 cycles	
	1	4 cycles	

Bits 13 to 11—RAS Precharge Interval Immediately after Self-Refreshing (TPCS2 to TPCS0): These bits specify, for DRAM, the RAS precharge interval immediately after self-refreshing.

Bit 13: TPCS2	Bit 12: TPCS1	Bit 11: TPCS0	Description
0	0	0	Cycles specified by TPC + 0 cycles (Initial value)
		1	Cycles specified by TPC + 1 cycle
	1	0	Cycles specified by TPC + 2 cycles
:	:	:	:
1	1	0	Cycles specified by TPC + 6 cycles
		1	Cycles specified by TPC + 7 cycles

Bits 10 to 8—RAS-CAS Delay specification (RCD2 to RCD0): These bits specify the DRAM RAS-CAS delay time.

			Description	
Bit 10: RCD2	Bit 9: RCD1	Bit 8: RCD0	Normal Access	EDO Access
0	0	0	1 cycle (Initial value)	1 cycle (Initial value)
		1	2 cycles	Do not set
:	:	:	:	:
1	1	0	7 cycles	Do not set
		1	8 cycles	Do not set

Note: Use the one cycle setting for EDO DRAM.

Bits 7 and 6—Reserved: These bits are always read as 0 and should only be written with 0.

Bits 5 and 4—Write Cycle Column Address Output Cycle Interval Specification (DWW1, DWW0): These bits specify the column address output cycle interval in a DRAM write cycle.

		Description		
Bit 5: DWW1	Bit 4: DWW0	In Normal Write Cycle	In EDO Write Cycle	In EDO Burst Write Cycle
0	0	2 cycles (no waits)*	2 cycles (no waits)*	1 cycle (no waits)*
	1	3 cycles (1 wait)	Do not set	Do not set
1	0	4 cycles (2 waits)	Do not set	Do not set
	1	5 cycles (3 waits)	Do not set	Do not set

Note: * Initial value

Use the no wait setting for EDO DRAM.

Bits 3 and 2—Read Cycle Column Address Output Cycle Interval Specification (DWR1, DWR0): These bits specify the column address output cycle interval in a DRAM read cycle.

		Description		
Bit 3: DWR1	**Bit 2: DWR0**	**In Normal Read Cycle**	**In EDO Read Cycle**	**In EDO Burst Read Cycle**
0	0	2 cycles (no waits)*	2 cycles (no waits)*	1 cycle (no waits)*
	1	3 cycles (1 wait)	Do not set	Do not set
1	0	4 cycles (2 waits)	Do not set	Do not set
	1	5 cycles (3 waits)	Do not set	Do not set

Note: * Initial value

Use the no wait setting for EDO DRAM.

Bits 1 and 0—Reserved: These bits are always read as 0 and should only be written with 0.

8.2.5 DRAM Control Register 2 (DCR2)

DRAM control register 2 (DCR2) is a 16-bit readable/writable register that specifies DRAM control. Control is the same for CS4 and CS5 space accesses.

DCR2 is initialized to H'1FE0 by a power-on reset, but is not initialized in standby mode.

Bit:	15	14	13	12	11	10	9	8
	DIW2	DIW1	DIW0	DDWW3	DDWW2	DDWW1	DDWW0	DDWR3
Initial value:	0	0	0	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
	DDWR2	DDWR1	DDWR0	RDW	TCAS	—	—	—
Initial value:	1	1	1	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R	R	R

Bits 15 to 13—Idle Cycles after DRAM Access (DIW2 to DIW0): These bits specify the number of idle cycles to be inserted between bus cycles when access is switched from the CS4 or CS5 space to another space, or from read access to write access within the same CS4 or CS5 space. When switching to access to a different space, one idle cycle is inserted automatically in the case of a read cycle, and two idle cycles in the case of a write cycle, even if "No idle cycles" is set. When switching from a read cycle to a write cycle within the same space, two idle cycles are inserted automatically.

Bit 15: DIW2	Bit 14: DIW1	Bit 13: DIW0	Description	
0	0	0	No idle cycles	(Initial value)
		1	1 idle cycle inserted	
	1	0	2 idle cycles inserted	
:	:	:	:	
1	1	0	6 idle cycles inserted	
		1	7 idle cycles inserted	

Bits 12 to 9—DMA Single Address Mode Write Access Wait State Insertion Cycle Specification (DDWW3 to DDWW0): These bits specify the number of wait states to be inserted in writes to DRAM in DMA single address mode.

Bit 12: DDWW3	Bit 11: DDWW2	Bit 10: DDWW1	Bit 9: DDWW0	Description	
				Normal Access	EDO Access
0	0	0	0	No waits	No waits
			1	1 wait	Do not set
		1	0	2 waits	Do not set
:	:	:	:	:	:
1	1	1	0	14 waits	Do not set
			1	15 waits (Initial value)	Do not set (Initial value)

Note: Use the no wait setting for EDO DRAM.

Bits 8 to 5—DMA Single Address Mode Read Access Wait State Insertion Cycle Specification (DDWR3 to DDWR0): These bits specify the number of wait states to be inserted in reads from DRAM in DMA single address mode.

Bit 8: DDWR3	Bit 7: DDWR2	Bit 6: DDWR1	Bit 5: DDWR0	Description	
				Normal Access	**EDO Access**
0	0	0	0	No waits	No waits
			1	1 wait	Do not set
		1	0	2 waits	Do not set
:	:	:	:	:	:
1	1	1	0	14 waits	Do not set
			1	15 waits (Initial value)	Do not set (Initial value)

Note: Use the no wait setting for EDO DRAM.

Bit 4—Idle Cycle Insertion before Continuous Burst Operation in DMA Single Transfer in RAS Down Mode (RDW): Specifies whether one idle cycle is to be inserted before burst operation when the same DRAM row address is accessed in DMA single mode during RAS down mode. This cycle is inserted only when access is switched from another space to the CS4 space or CS5 space, or from read access to write access within the same CS4 or CS5 space.

Bit 4: RDW	Description	
0	No idle cycle	(Initial value)
1	1 idle cycle inserted	

Bit 3—Write Cycle CAS Assertion Width with Software Wait Setting (TCAS): Specifies the CAS assertion width in a DRAM write cycle.

Bit 3: TCAS	Description	
	Normal Access	**EDO Access**
0	1 cycle (Initial value)	1 cycle (Initial value)
1	2 cycles	Do not set

Note: Use the no wait setting for EDO DRAM.

Bits 2 to 0—Reserved: These bits are always read as 0 and should only be written with 0.

8.2.6 DRAM Control Register 3 (DCR3)

DRAM control register 3 (DCR3) is a 16-bit readable/writable register that specifies DRAM control. Control is the same for CS4 and CS5 space accesses.

DCR3 is initialized to H'1800 by a power-on reset, but is not initialized in standby mode.

Bit:	15	14	13	12	11	10	9	8
	BE	RSD	EDO	DSZ1	DSZ0	AMX2	AMX1	AMX0
Initial value:	0	0	0	1	1	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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

Bit 15—Burst Enable (BE): Specifies whether or not burst access is performed on DRAM.

Bit 15: BE	Description	
0	Burst disabled	(Initial value)
1	Access in fast page mode	

Bit 14—RAS Down Mode (RSD): Specifies whether or not RAS down mode access is performed on DRAM.

Bit 14: RSD	Description	
0	DRAM accessed in RAS up mode	(Initial value)
1	DRAM accessed in RAS down mode	

Bit 13—EDO Mode (EDO): Specifies whether or not EDO mode access is performed on DRAM.

Bit 13: EDO	Description	
0	DRAM accessed in normal mode	(Initial value)
1	DRAM accessed in EDO mode	

Bits 12 and 11—Bus Width Specification (DSZ1, DSZ0): These bits specify the bus width for DRAM.

Bit 12: DSZ1	Bit 11: DSZ0	Description	
0	0	Reserved (Do not set)	
	1	8 bits	
1	0	16 bits	
	1	32 bits	(Initial value)

Bits 10 to 8—Address Multiplexing Specification (AMX2 to AMX0): These bits specify DRAM address multiplexing.

Bit 10: AMX2	Bit 9: AMX1	Bit 8: AMX0	Description	
0	0	0	9 bits	(Initial value)
		1	10 bits	
	1	0	11 bits	
		1	12 bits	
1	0	0	13 bits	
		1	14 bits	
	1	0	15 bits	
		1	16 bits	

Bit 7—Refresh Control (RFSH): Specifies whether or not refreshing is performed for DRAM. When the refresh function is not used, the refresh request cycle generation timer can be used as an interval timer.

Bit 7: RFSH	Description	
0	Refreshing is not performed	(Initial value)
1	Refreshing is performed	

Bit 6—Refresh Mode (RMD): Specifies whether normal refreshing or self-refreshing is performed for DRAM when the RFSH bit is set to 1. When the RFSH bit is 1 and this bit is cleared to 0, CAS-before-RAS refreshing is performed using the cycle set with refresh-related registers RTCNT, RTCOR, and RTCSR. If a refresh request is issued during execution of an external bus cycle, the refresh cycle is executed when the bus cycle ends. When the RFSH bit is 1 and this bit is set to 1, the self-refresh state is set after waiting for the end of any currently executing external bus cycle. All refresh requests for memory in the self-refresh state are ignored.

Bit 6: RMD	Description
0	CAS-before-RAS refreshing is performed (Initial value)
1	Self-refreshing is performed

Bits 5 to 0—Reserved: These bits are always read as 0 and should only be written with 0.

8.2.7 Refresh Timer Control/Status Register (RTCSR)

The refresh timer control/status register (RTCSR) is a 16-bit readable/writable register that specifies the refresh cycle, whether interrupts are to be generated, and if so the interrupt cycle.

RTCSR is initialized to H'0000 by a power-on reset, but is not initialized in standby mode.

Bit:	15	14	13	12	11	10	9	8
	CMF	CMIE	CKS2	CKS1	CKS0	OVF	OVIE	LMTS1
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
	LMTS0	BREF2	BREF1	BREF0	TRAS2	TRAS1	TRAS0	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

Bit 15—Compare Match Flag (CMF): Status flag that indicates a match between the refresh timer counter (RTCNT) and refresh time constant register (RTCOR) values.

Bit 15: CMF	Description
0	RTCNT and RTCOR values do not match (Initial value) [Clearing condition] When 0 is written to CMF after reading RTCSR when CMF = 1, or when refreshing is performed with RFSH = 1 and RMD = 0 (CBR refreshing performed)
1	RTCNT and RTCOR values match [Setting condition] When RTCNT = RTCOR

Bit 14—Compare Match Interrupt Enable (CMIE): Controls generation or suppression of an interrupt request when the CMF flag is set to 1 in RTCSR. Do not set this bit to 1 when CAS-before-RAS refreshing is used.

Bit 14: CMIE	Description
0	Interrupts initiated by CMF are disabled (Initial value)
1	Interrupts initiated by CMF are enabled

Bits 13 to 11—Clock Select Bits (CKS2 to CKS0): These bits select the RTCNT input clock. The base clock is the external bus clock (CKE). The RTCNT count clock is obtained by scaling CKE by the specified factor.

To change the division ratio (scaling factor), first clear bits CKS0 to CKS2 to 0, then write the required value in these bits.

Bit 13: CKS2	Bit 12: CKS1	Bit 11: CKS0	Description
0	0	0	Clock input disabled (Initial value)
		1	External bus clock (CKE) /4
	1	0	External bus clock (CKE) /16
		1	External bus clock (CKE) /64
1	0	0	External bus clock (CKE) /256
		1	External bus clock (CKE) /1024
	1	0	External bus clock (CKE) /2048
		1	External bus clock (CKE) /4096

Bit 10—Refresh Count Overflow Flag (OVF): Status flag that indicates that the number of refresh requests indicated by the refresh count register (RFCR) has exceeded the number specified by the LMTS bits in RTCSR.

Bit 10: OVF	Description
0	RFCR has not overflowed the count limit indicated by the LMTS bits [Clearing condition] When RTCSR is read while OVF = 1, then 0 is written to OVF (Initial value)
1	RFCR has overflowed the count limit indicated by the LMTS bits [Setting condition] When RFCR overflows the count limit indicated by the LMTS bits

Bit 9—Refresh Count Overflow Interrupt Enable (OVIE): Controls generation or suppression of an interrupt request when the OVF flag is set to 1 in RTCSR.

Bit 9: OVIE	Description	
0	Interrupts initiated by OVF are disabled	(Initial value)
1	Interrupts initiated by OVF are enabled	

Bits 8 and 7—Refresh Count Overflow Limit Select (LMTS1, LMTS0): These bits specify the count limit to be compared with the refresh count indicated by the refresh count register (RFCR). If the RFCR register value exceeds the value specified by the LMTS bits, the OVF flag is set to 1.

Bit 8: LMTS1	Bit 7: LMTS0	Description	
0	0	Refresh count limit is 4096	(Initial value)
	1	Refresh count limit is 2048	
1	0	Refresh count limit is 1024	
	1	Refresh count limit is 512	

Bits 6 to 4—Refresh Request Number Select (BREF2 to BREF0): These bits specify the number of consecutive refresh requests requested by a single compare match. The number of CAS-before-RAS refreshes specified by these bits are performed consecutively.

Bit 6: BREF2	Bit 5: BREF1	Bit 4: BREF0	Description
0	0	0	1 CAS-before-RAS refresh is performed (Initial value)
		1	2 consecutive CAS-before-RAS refreshes are performed
	1	0	3 consecutive CAS-before-RAS refreshes are performed
		1	4 consecutive CAS-before-RAS refreshes are performed
1	0	0	5 consecutive CAS-before-RAS refreshes are performed
		1	6 consecutive CAS-before-RAS refreshes are performed
	1	0	7 consecutive CAS-before-RAS refreshes are performed
		1	8 consecutive CAS-before-RAS refreshes are performed

Bits 3 to 1—Refresh RAS Assertion Interval Specification (TRAS2 to TRAS0): These bits specify the refresh interval of the DRAM connected to areas 4 and 5. With DRAM, this is the RAS assertion interval in CAS-before-RAS refreshing.

Bit 3: TRAS2	Bit 2: TRAS1	Bit 1: TRAS0	Description
0	0	0	2 cycles (Initial value)
		1	3 cycles
	1	0	4 cycles
		1	5 cycles
1	0	0	6 cycles
		1	7 cycles
	1	0	8 cycles
		1	9 cycles

Bit 0—Reserved: This bit is always read as 0 and should only be written with 0.

8.2.8 Refresh Timer Counter (RTCNT)

The refresh timer counter (RTCNT) is an 8-bit readable/writable counter that is incremented by the input clock selected by bits CKS2 to CKS0 in the RTCSR register. When the RTCNT counter value matches the RTCOR register value, the CMF bit is set in the RTCSR register and the RTCNT counter is cleared.

RTCNT bits 15 to 8 are reserved; they are always read as 0 and should only be written with 0. RTCNT is initialized to H'00 by a power-on reset. In standby mode, RTCNT is not initialized, and retains its contents.

Bit:	15	14	13	12	11	10	9	8
	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit:	7	6	5	4	3	2	1	0
	RTCNT7	RTCNT6	RTCNT5	RTCNT4	RTCNT3	RTCNT2	RTCNT1	RTCNT0
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

8.2.9 Refresh Time Constant Register (RTCOR)

The refresh time constant register (RTCOR) is a 16-bit readable/writable register that specifies the upper limit of the RTCNT counter. The RTCOR register and RTCNT counter values (lower 8 bits) are constantly compared, and when they match the CMF bit is set in the RTCSR register and the RTCNT counter is cleared to 0. If RFSH has been set to 1 and RMD has been cleared to 0 in DRAM control register 3 (DCR3), CAS-before-RAS refreshing is performed. If the CMIE bit has been set to 1 in RTCSR, a compare match interrupt (CMI) is generated.

RTCOR bits 15 to 8 are reserved; they are always read as 0 and should only be written with 0. RTCOR is initialized to H'0000 by a power-on reset, but is not initialized in standby mode.

Bit:	15	14	13	12	11	10	9	8
	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit:	7	6	5	4	3	2	1	0
	RTCOR7	RTCOR6	RTCOR5	RTCOR4	RTCOR3	RTCOR2	RTCOR1	RTCOR0
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

8.2.10 Refresh Count Register (RFCR)

The refresh count register (RFCR) is a 12-bit readable/writable counter that counts the number of refreshes by being incremented each time the RTCOR register and RTCNT counter values match. If the RFCR register value exceeds the count limit specified by bits LMTS1 and LMTS0 in the RTCSR register, the OVF flag is set in the RTCSR register and the RFCR register is cleared.

RFCR bits 15 to 12 are reserved; they are always read as 0 and should only be written with 0. RFCR is initialized to H'0000 by a power-on reset. In standby mode, RFCR is not initialized, and retains its contents.

Bit:	15	14	13	12	11	10	9	8
	—	—	—	—	RFCR11	RFCR10	RFCR9	RFCR8
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:	7	6	5	4	3	2	1	0
	RFCR7	RFCR6	RFCR5	RFCR4	RFCR3	RFCR2	RFCR1	RFCR0
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

8.3 Operation

8.3.1 Endian/Access Size and Data Alignment

The SH7065 supports both big-endian mode, in which the most significant byte (MSByte) is at the 0 address end in a string of byte data, and little-endian mode, in which the least significant byte (LSByte) is at the 0 address end. The mode is set by means of the ENDIAN bit in area control register 1 (ACR1_0 to ACR1_5).

A data bus width of 8, 16, or 32 bits can be selected for normal memory and DRAM. For multiplexed I/O there is a choice of 8 or 16 bits. Data alignment is carried out according to the data bus width and endian mode of each device. Thus, four read operations are needed to read longword data from an 8-bit device. In the SH7065, data alignment and data length conversion between the different interfaces is performed automatically.

The relationship between the endian mode, device data width, and access unit, is shown in tables 8.4 to 8.9. Instruction codes should be handled as word data. Similarly, with a 32-bit instruction code, handle the A-field and B-field instruction codes as word data.

Table 8.4 32-Bit External Device/Big-Endian Access and Data Alignment

	Data Bus				Strobe Signals			
Operation	**D31–D24**	**D23–D16**	**D15–D8**	**D7–D0**	**WRHH, HHBS, CASHH**	**WRHL, HLBS, CASHL**	**WRLH, LHBS, CASLH**	**WRLL, LLBS, CASLL**
Address 0 byte access	Data 7–0	—	—	—	Asserted			
Address 1 byte access	—	Data 7–0	—	—		Asserted		
Address 2 byte access	—	—	Data 7–0	—			Asserted	
Address 3 byte access	—	—	—	Data 7–0				Asserted
Address 0 word access	Data 15–8	Data 7–0	—	—	Asserted	Asserted		
Address 2 word access	—	—	Data 15–8	Data 7–0			Asserted	Asserted
Address 0 longword access	Data 31–24	Data 23–16	Data 15–8	Data 7–0	Asserted	Asserted	Asserted	Asserted

Table 8.5 16-Bit External Device/Big-Endian Access and Data Alignment

		Data Bus				Strobe Signals			
Operation		**D31–D24**	**D23–D16**	**D15–D8**	**D7–D0**	**WRHH, HHBS, CASHH**	**WRHL, HLBS, CASHL**	**WRLH, LHBS, CASLH**	**WRLL, LLBS, CASLL**
Address 0 byte access		—	—	Data 7–0	—			Asserted	
Address 1 byte access		—	—	—	Data 7–0				Asserted
Address 2 byte access		—	—	Data 7–0	—			Asserted	
Address 3 byte access		—	—	—	Data 7–0				Asserted
Address 0 word access		—	—	Data 15–8	Data 7–0			Asserted	Asserted
Address 2 word access		—	—	Data 15–8	Data 7–0			Asserted	Asserted
Address 0 longword access	1st access (address 0)	—	—	Data 31–24	Data 23–16			Asserted	Asserted
	2nd access (address 2)	—	—	Data 15–8	Data 7–0			Asserted	Asserted

Table 8.6 8-Bit External Device/Big-Endian Access and Data Alignment

		Data Bus				Strobe Signals			
Operation		**D31–D24**	**D23–D16**	**D15–D8**	**D7–D0**	**WRHH, HHBS, CASHH**	**WRHL, HLBS, CASHL**	**WRLH, LHBS, CASLH**	**WRLL, LLBS, CASLL**
Address 0 byte access		—	—	—	Data 7–0				Asserted
Address 1 byte access		—	—	—	Data 7–0				Asserted
Address 2 byte access		—	—	—	Data 7–0				Asserted
Address 3 byte access		—	—	—	Data 7–0				Asserted
Address 0 word access	1st access (address 0)	—	—	—	Data 15–8				Asserted
	2nd access (address 1)	—	—	—	Data 7–0				Asserted
Address 2 word access	1st access (address 2)	—	—	—	Data 15–8				Asserted
	2nd access (address 3)	—	—	—	Data 7–0				Asserted
Address 0 longword access	1st access (address 0)	—	—	—	Data 31–24				Asserted
	2nd access (address 1)	—	—	—	Data 23–16				Asserted
	3rd access (address 2)	—	—	—	Data 15–8				Asserted
	4th access (address 3)	—	—	—	Data 7–0				Asserted

Table 8.7 32-Bit External Device/Little-Endian Access and Data Alignment

	Data Bus				Strobe Signals			
Operation	**D31–D24**	**D23–D16**	**D15–D8**	**D7–D0**	**WRHH, HHBS, CASHH**	**WRHL, HLBS, CASHL**	**WRLH, LHBS, CASLH**	**WRLL, LLBS, CASLL**
Address 0 byte access	—	—	—	Data 7–0				Asserted
Address 1 byte access	—	—	Data 7–0	—			Asserted	
Address 2 byte access	—	Data 7–0	—	—		Asserted		
Address 3 byte access	Data 7–0	—	—	—	Asserted			
Address 0 word access	—	—	Data 15–8	Data 7–0			Asserted	Asserted
Address 2 word access	Data 15–8	Data 7–0	—	—	Asserted	Asserted		
Address 0 longword access	Data 31–24	Data 23–16	Data 15–8	Data 7–0	Asserted	Asserted	Asserted	Asserted

Table 8.8 16-Bit External Device/Little-Endian Access and Data Alignment

		Data Bus				Strobe Signals			
Operation		**D31–D24**	**D23–D16**	**D15–D8**	**D7–D0**	**WRHH, HHBS, CASHH**	**WRHL, HLBS, CASHL**	**WRLH, LHBS, CASLH**	**WRLL, LLBS, CASLL**
Address 0 byte access		—	—	—	Data 7–0				Asserted
Address 1 byte access		—	—	Data 7–0	—			Asserted	
Address 2 byte access		—	—	—	Data 7–0				Asserted
Address 3 byte access		—	—	Data 7–0	—			Asserted	
Address 0 word access		—	—	Data 15–8	Data 7–0			Asserted	Asserted
Address 2 word access		—	—	Data 15–8	Data 7–0			Asserted	Asserted
Address 0 longword access	1st access (address 0)	—	—	Data 15–8	Data 7–0			Asserted	Asserted
	2nd access (address 2)	—	—	Data 31–24	Data 23–16			Asserted	Asserted

Table 8.9 8-Bit External Device/Little-Endian Access and Data Alignment

Operation		Data Bus				Strobe Signals			
		D31–D24	D23–D16	D15–D8	D7–D0	WRHH, HHBS, CASHH	WRHL, HLBS, CASHL	WRLH, LHBS, CASLH	WRLL, LLBS, CASLL
Address 0 byte access		—	—	—	Data 7–0				Asserted
Address 1 byte access		—	—	—	Data 7–0				Asserted
Address 2 byte access		—	—	—	Data 7–0				Asserted
Address 3 byte access		—	—	—	Data 7–0				Asserted
Address 0 word access	1st access (address 0)	—	—	—	Data 7–0				Asserted
	2nd access (address 1)	—	—	—	Data 15–8				Asserted
Address 2 word access	1st access (address 2)	—	—	—	Data 7–0				Asserted
	2nd access (address 3)	—	—	—	Data 15–8				Asserted
Address 0 longword access	1st access (address 0)	—	—	—	Data 7–0				Asserted
	2nd access (address 1)	—	—	—	Data 15–8				Asserted
	3rd access (address 2)	—	—	—	Data 23–16				Asserted
	4th access (address 3)	—	—	—	Data 31–24				Asserted

8.3.2 Areas

Area 0

For area 0, address bits A31 to A26 are 000000. However, in the on-chip ROM enabled modes, the space from H'0000 0000 to H'0003 FFFF is allocated to on-chip ROM. Enabling or disabling of on-chip ROM is selected in a power-on reset by means of external pins MD2, MD1, and MD0.

Normal memory such as SRAM and ROM can be connected to this space. A value of 0 must always be written to bits TP1 and TP0 in the ACR1 register. These bits are always read as 0.

A bus width of 8, 16, or 32 bits can be selected in a power-on reset by means of external pins MD1 and MD0.

When area 0 space is accessed, the $\overline{\text{CS0}}$ signal is asserted. In addition, the $\overline{\text{RD}}$ signal, which can be used as the SRAM and ROM $\overline{\text{OE}}$ signal, and write control signals $\overline{\text{WRHH}}$ to $\overline{\text{WRLL}}$, are asserted.

As regards the number of bus cycles, from 0 to 15 waits can be selected with bits W3 to W0 in the WCR register. In addition, any number of waits can be inserted in each bus cycle by means of the external wait pin ($\overline{\text{WAIT}}$).

Areas 1 to 3

For areas 1 to 3, address bits A31 to A26 are 000001 to 000011.

Normal memory such as SRAM and ROM, and address/data multiplexed I/O devices, can be connected to this space. The kind of memory control to be performed is set with bits TP1 and TP0 in the ACR1 register provided for each area.

A bus width of 8, 16, or 32 bits can be selected in a power-on reset with bits SZ1 and SZ0 in the ACR1 register provided for each area. However, when a multiplexed address/data I/O device is connected, bits SZ1 and SZ0 in the ACR1 register are ignored, and the bus width is 8 bits when address bit A14 is 0, and 16 bits when 1.

When area 1, 2, or 3 space is accessed, the $\overline{\text{CS1}}$, $\overline{\text{CS2}}$, or $\overline{\text{CS3}}$ signal is asserted, respectively. In addition, the $\overline{\text{RD}}$ signal, which can be used as the SRAM and ROM $\overline{\text{OE}}$ signal, and write control signals $\overline{\text{WRHH}}$ to $\overline{\text{WRLL}}$, are asserted.

As regards the number of bus cycles, from 0 to 15 waits can be selected with bits W3 to W0 in the WCR register provided for each area. In addition, any number of waits can be inserted in each bus cycle by means of the external wait pin ($\overline{\text{WAIT}}$).

Areas 4 and 5

For areas 4 and 5, address bits A31 to A26 are 010000 and 010001, respectively.

A bus width of 8, 16, or 32 bits can be selected in a power-on reset with bits DSZ1 and DSZ0 in the DCR3 register.

When area 4 or 5 space is accessed, the $\overline{\text{CS4}}$ or $\overline{\text{CS5}}$ signal is asserted, respectively. The $\overline{\text{RAS}}$ signal, the $\overline{\text{CASHH}}$, $\overline{\text{CASHL}}$, $\overline{\text{CASLH}}$, and $\overline{\text{CASLL}}$ signals, and the RDWR signal are asserted, and address multiplexing is performed. $\overline{\text{RAS}}$, $\overline{\text{CAS}}$, and data timing control, and address multiplexing control, can be set with registers DCR1 to DCR3.

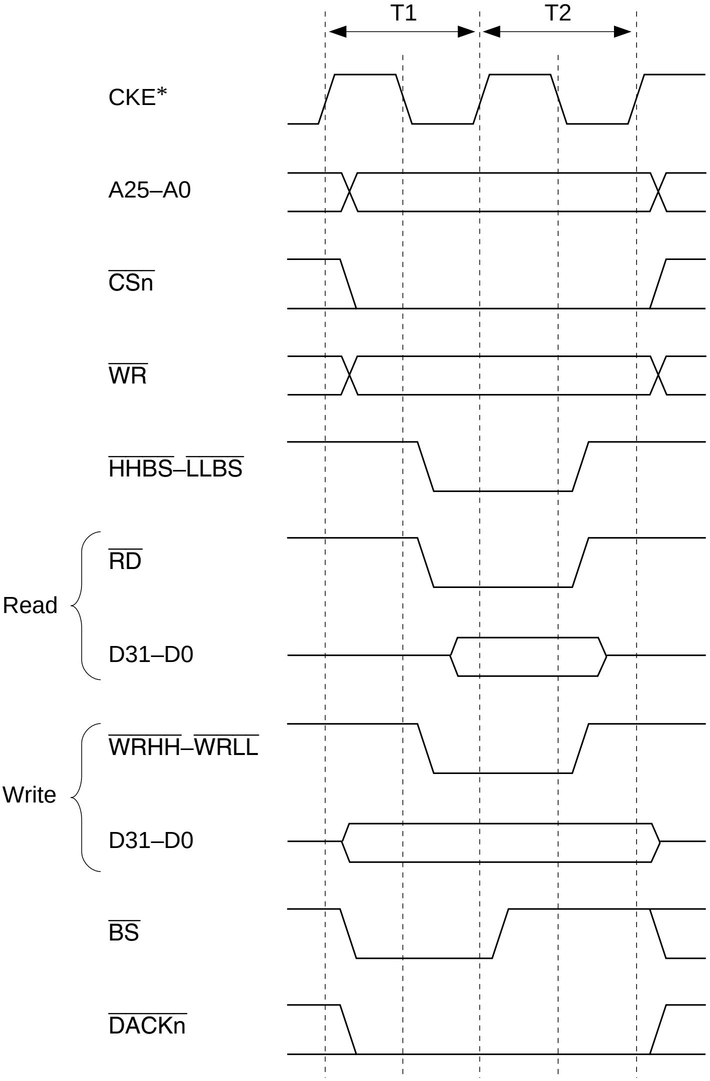
As regards the number of bus cycles, from 0 to 3 waits can be selected with a setting in the DCR1 register. In addition, any number of waits can be inserted in each bus cycle by means of the external wait pin ($\overline{\text{WAIT}}$). However, a wait setting should not be made for EDO DRAM.

8.3.3 Normal Space Access

Basic Timing

The SH7065 uses strobe signal output for normal space access in consideration of the fact that mainly static RAM will be directly connected. Figure 8.3 shows the basic timing of normal space accesses. A no-wait normal access is completed in two cycles. The $\overline{\text{BS}}$ signal is asserted for one cycle to indicate the start of a bus cycle.

There is no access size specification when reading. The correct access start address is output in the least significant bit of the address, but since there is no access size specification, 32 bits are always read in the case of a 32-bit device, and 16 bits in the case of a 16-bit device, using the necessary byte value. When writing, only the $\overline{\text{WRLL}}$ to $\overline{\text{WRHH}}$ signal for the byte to be written, or the $\overline{\text{WR}}$ signal and $\overline{\text{LLBS}}$ to $\overline{\text{HHBS}}$ are asserted, according to the setting of the BAS bit in the BCR register. For details, see section 8.3.1, Endian/Access Size and Data Alignment.



Note: * When the setting CKE = CKIO is made in clock mode 0 to 3, 6, or 7, CKE is identical to CKIO on the timing chart.
In clock modes 4 and 5, the phases of CKE and CKIO do not coincide, but the relative relationship of the AC specifications is the same as in the other clock modes.

Figure 8.3 Basic Timing of Normal Space Access

Figures 8.4, 8.5, and 8.6 show examples of connection to 32-, 16-, and 8-bit data width SRAM.

Figures 8.7 and 8.8 show examples of connection to 32- and 16-bit data width byte-strobe SRAM.

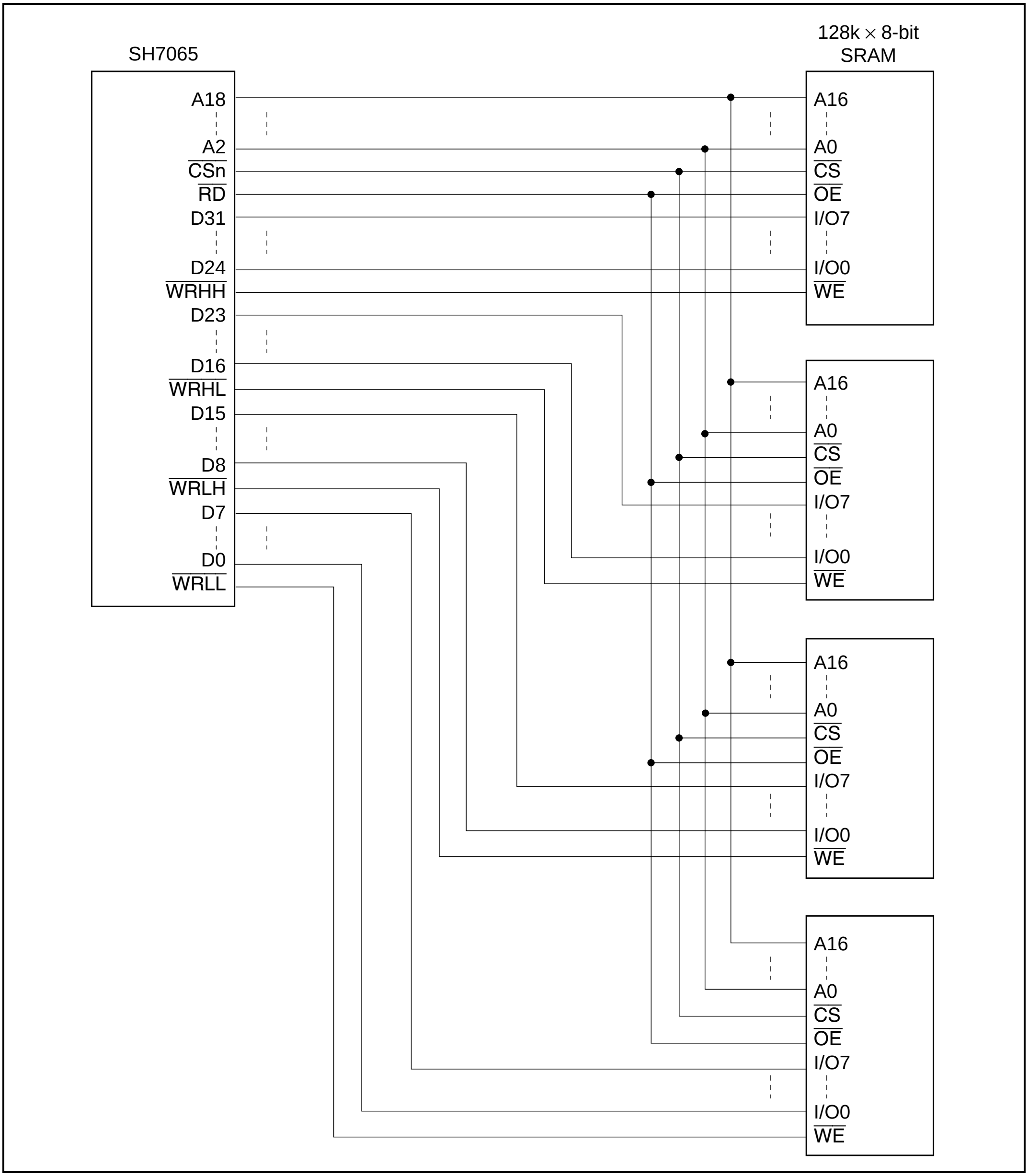


Figure 8.4 Example of 32-Bit Data Width SRAM Connection

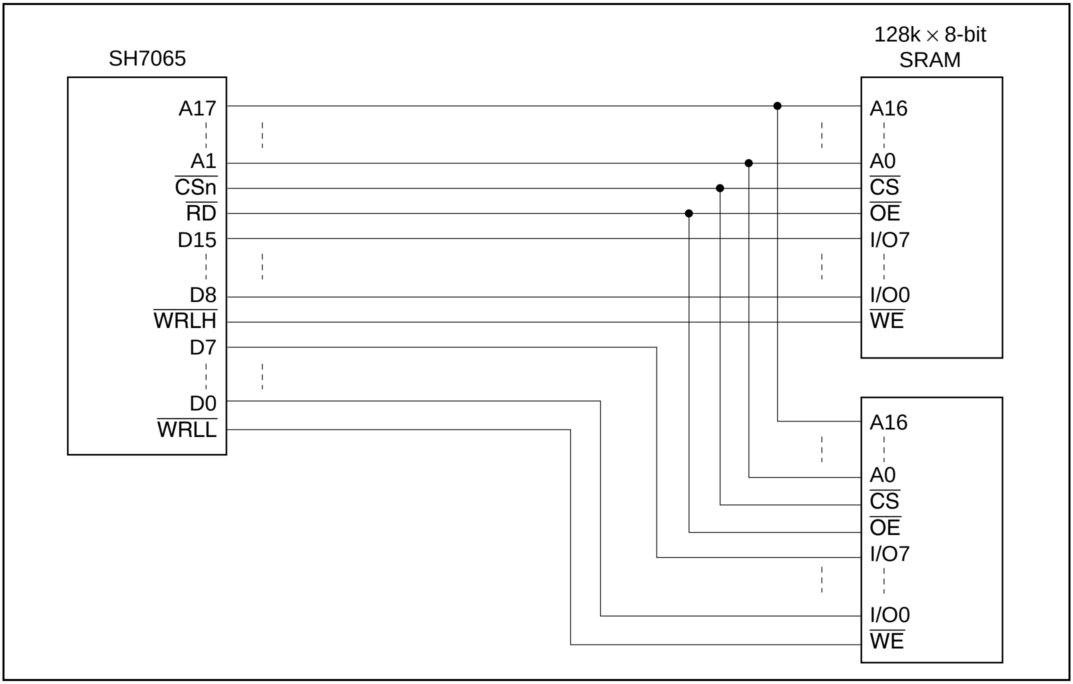


Figure 8.5 Example of 16-Bit Data Width SRAM Connection

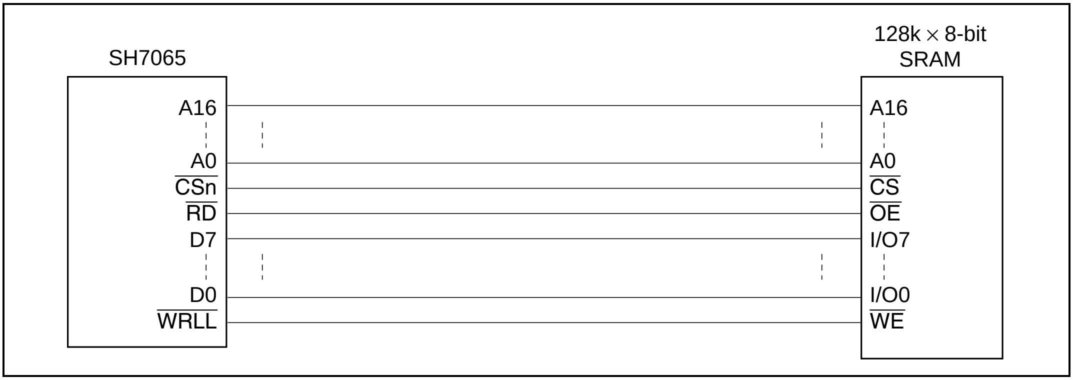


Figure 8.6 Example of 8-Bit Data Width SRAM Connection

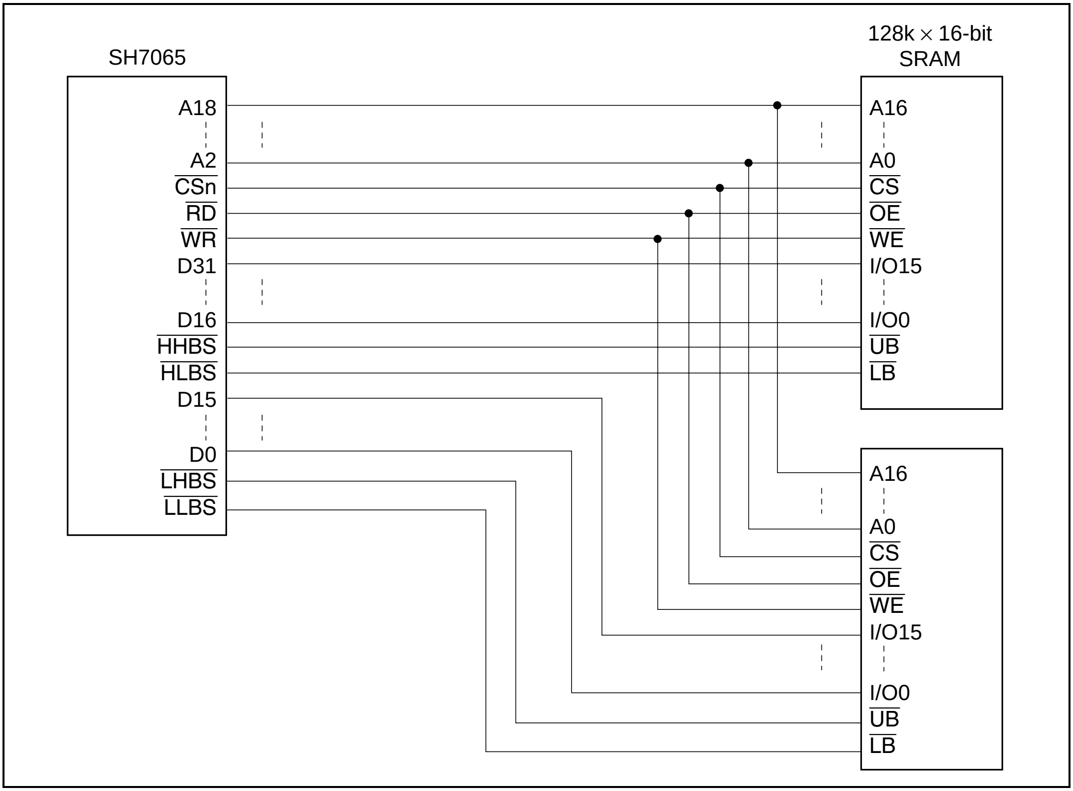


Figure 8.7 Example of 32-Bit Data Width Byte-Strobe SRAM Connection

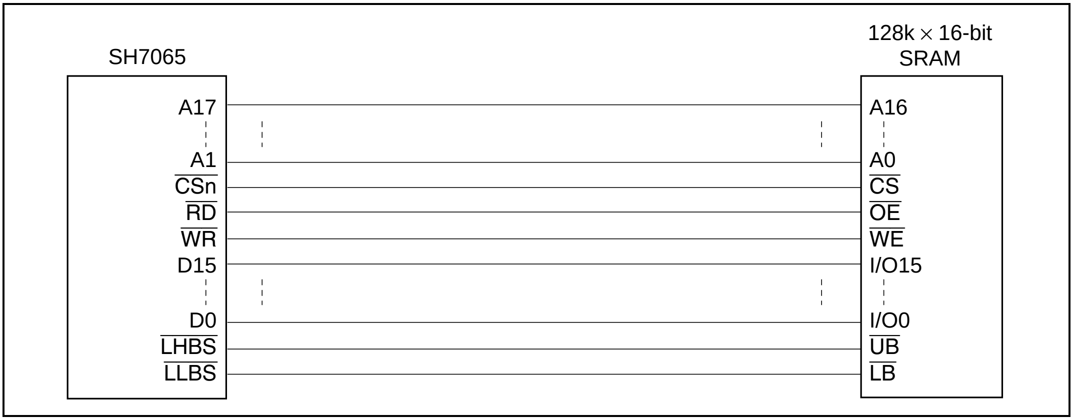
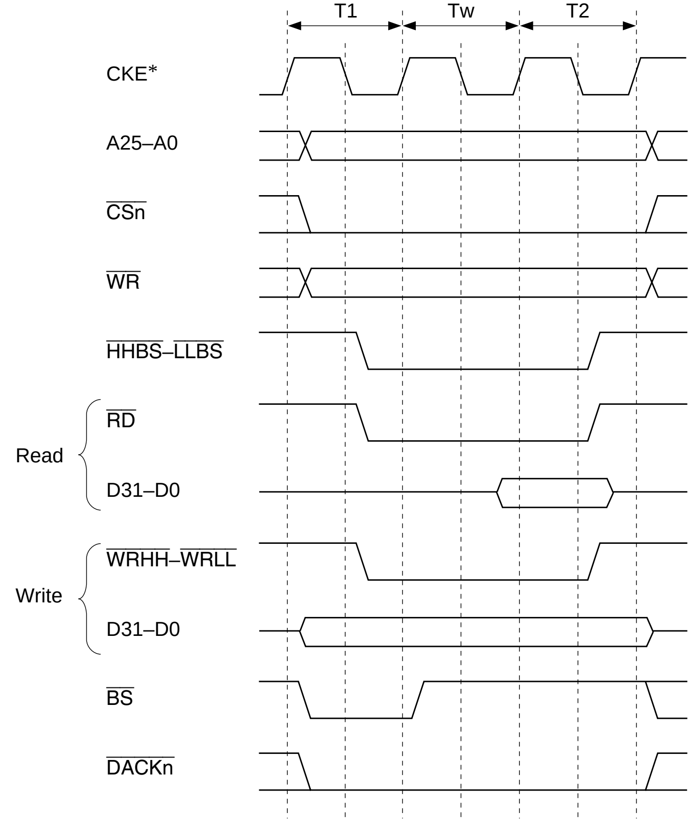


Figure 8.8 Example of 16-Bit Data Width Byte-Strobe SRAM Connection

Wait State Control

Wait state insertion in normal space access can be controlled by means of WCR settings. If the WCR wait specification bits corresponding to a particular area are not zero, a software wait is inserted in accordance with that specification. For details, see section 8.2.3, Wait Control Registers (WCR_0 to WCR_3).

The number of Tw cycles specified in WCR are inserted as wait cycles using the basic interface wait timing shown in figure 8.9.

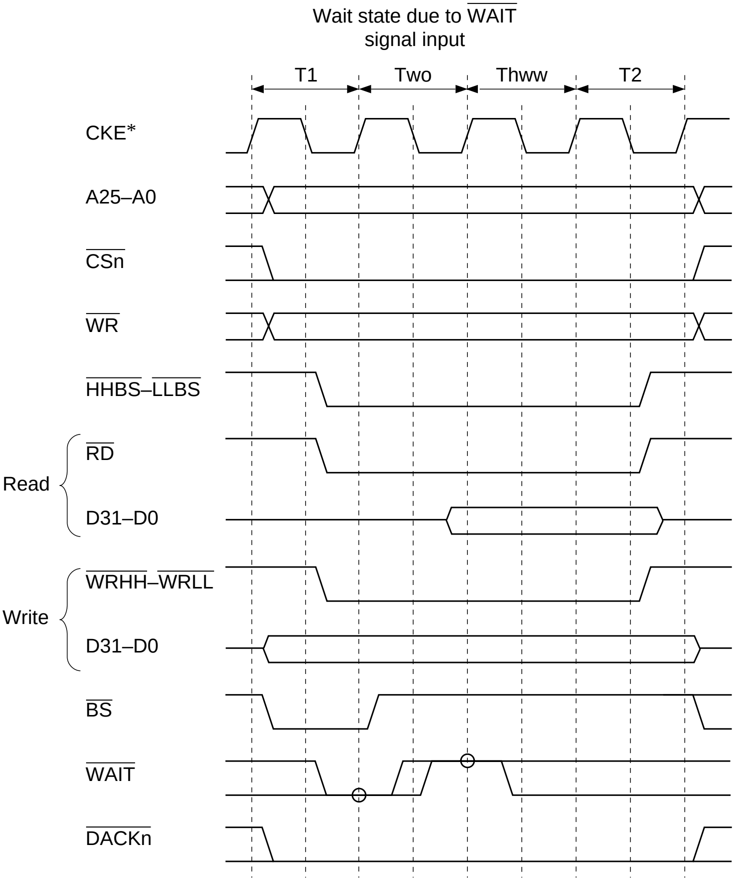


Note: * When the setting CKE = CKIO is made in clock mode 0 to 3, 6, or 7, CKE is identical to CKIO on the timing chart.
In clock modes 4 and 5, the phases of CKE and CKIO do not coincide, but the relative relationship of the AC specifications is the same as in the other clock modes.

Figure 8.9 Wait State Timing for Normal Space Access (One Software Wait State Inserted)

The wait input $\overline{\text{WAIT}}$ signal from an external source can be sampled by making the appropriate setting for the EXWE bit in ACR1. $\overline{\text{WAIT}}$ signal sampling is shown in figure 8.10. The signal is sampled at the rise of the clock in the T2 cycle.

By making a setting in bits HWW2 to HWW0 in WCR, additional software wait states can be inserted after the $\overline{\text{WAIT}}$ signal is negated. The specified number of Thww cycles (see figure 8.10) are inserted as wait cycles after negation of the $\overline{\text{WAIT}}$ signal.


Wait state due to $\overline{\text{WAIT}}$ signal input
T1
Two
Thww
T2
CKE*
A25–A0
$\overline{\text{CSn}}$
$\overline{\text{WR}}$
$\overline{\text{HHBS}}$–$\overline{\text{LLBS}}$
Read
$\overline{\text{RD}}$
D31–D0
Write
$\overline{\text{WRHH}}$–$\overline{\text{WRLL}}$
D31–D0
$\overline{\text{BS}}$
$\overline{\text{WAIT}}$
$\overline{\text{DACKn}}$

Note: * When the setting CKE = CKIO is made in clock mode 0 to 3, 6, or 7, CKE is identical to CKIO on the timing chart.
In clock modes 4 and 5, the phases of CKE and CKIO do not coincide, but the relative relationship of the AC specifications is the same as in the other clock modes.

Figure 8.10 Wait State Timing for Normal Space Access (One Wait Inserted by $\overline{\text{WAIT}}$ Signal, and One Software Wait Inserted after Negation of $\overline{\text{WAIT}}$ Signal)

SH7065F Bus Timing

Timing waveforms when longword access is performed to external word-size memory space are shown in figure 8.11. The SH7065F performs external accesses consecutively. The $\overline{\text{CS}}$ signal is asserted during this time, and remains asserted.

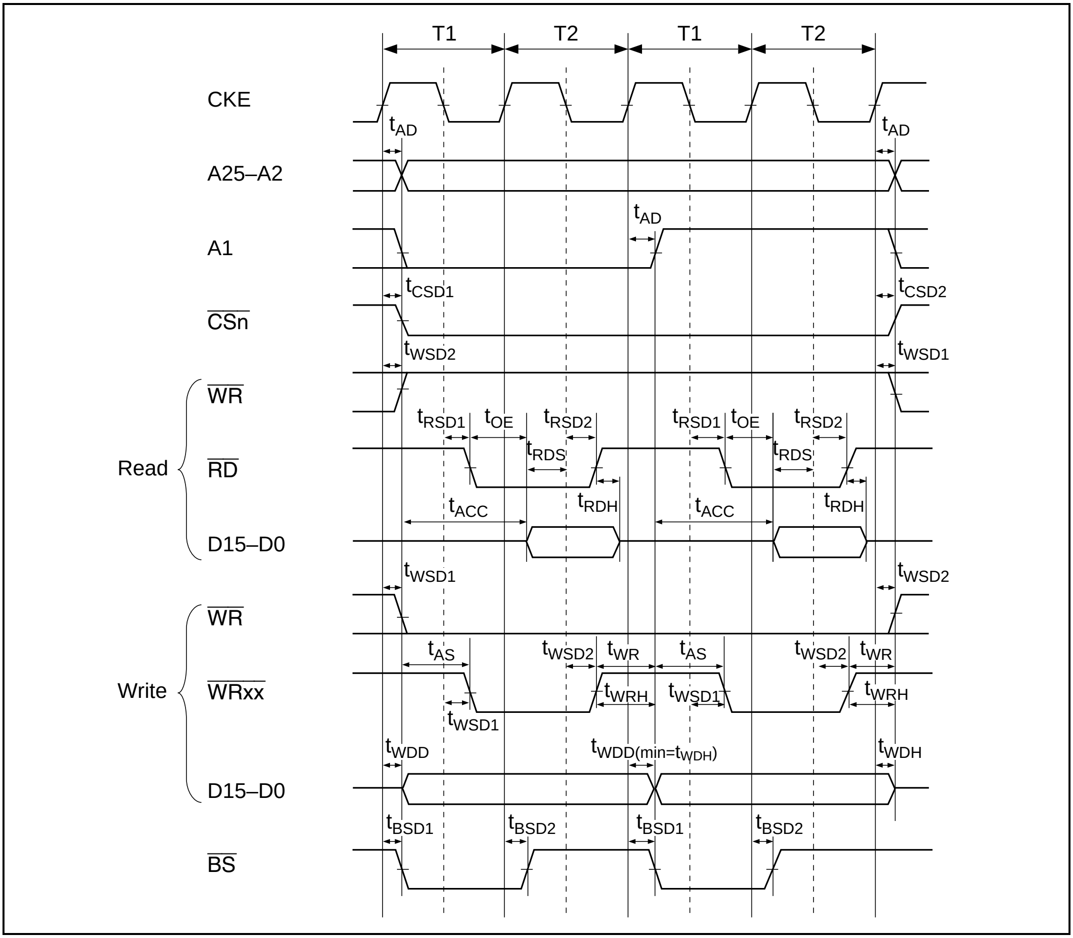
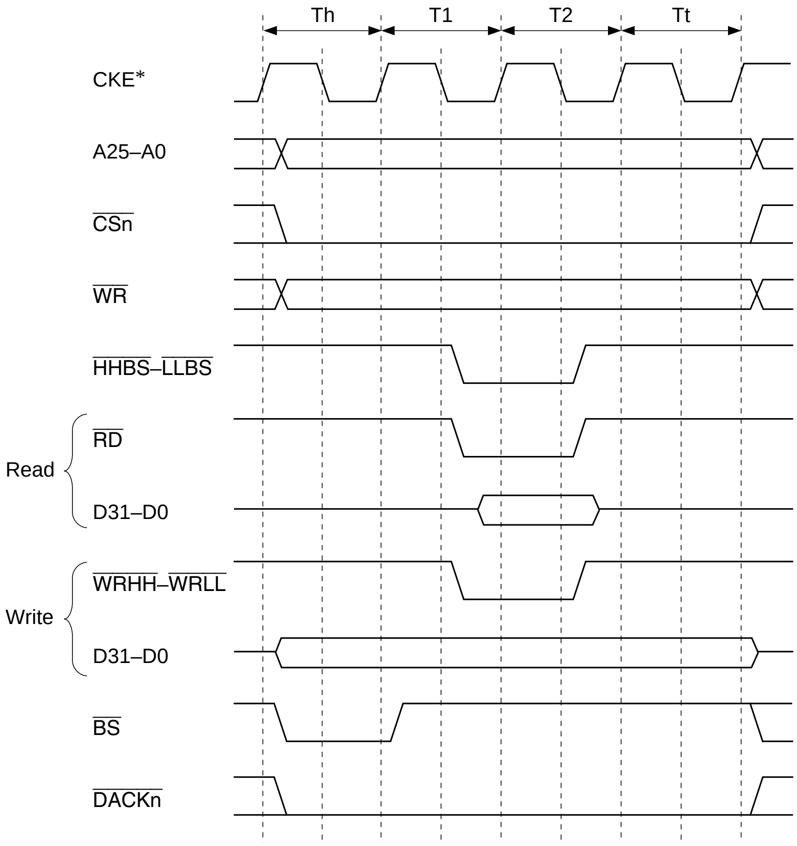


Figure 8.11 SH7065F Bus Timing

Extension of $\overline{\text{CS}}$ Assertion Interval

By making settings in bits SWH2 to SWH0 and SWT2 to SWT0 in ACR1, idle cycles can be inserted to prevent the $\overline{\text{RD}}$ or $\overline{\text{WR}}$ assertion interval from extending beyond the $\overline{\text{CSn}}$ assertion interval. This allows flexible interfacing to external circuitry. The timing is shown in figure 8.12. The Th and Tt cycles are added before and after the normal cycles, respectively. The number of Th cycles is set in bits SWH2 to SWH0, and the number of Tt cycles in bits SWT2 to SWT0. In these cycles, only $\overline{\text{CSn}}$ is asserted; $\overline{\text{RD}}$ and $\overline{\text{WR}}$ are not. Also, since data is extended up to the Tt cycle, this feature is useful for devices with slow write operations, for example.



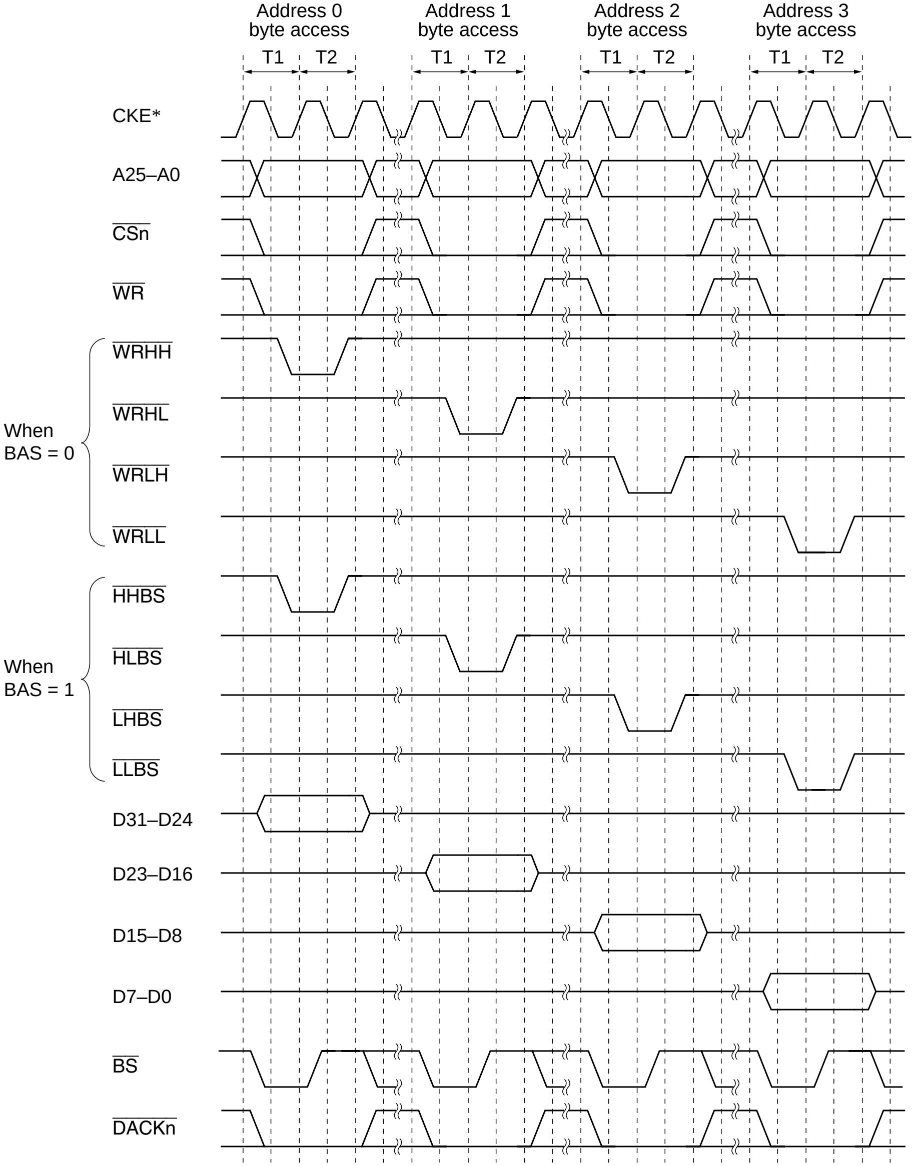
Note: * When the setting CKE = CKIO is made in clock mode 0 to 3, 6, or 7, CKE is identical to CKIO on the timing chart.
In clock modes 4 and 5, the phases of CKE and CKIO do not coincide, but the relative relationship of the AC specifications is the same as in the other clock modes.

Figure 8.12 $\overline{\text{CS}}$ Assertion Interval Extension (SWH = 1, SWT = 1)

Byte Access Control

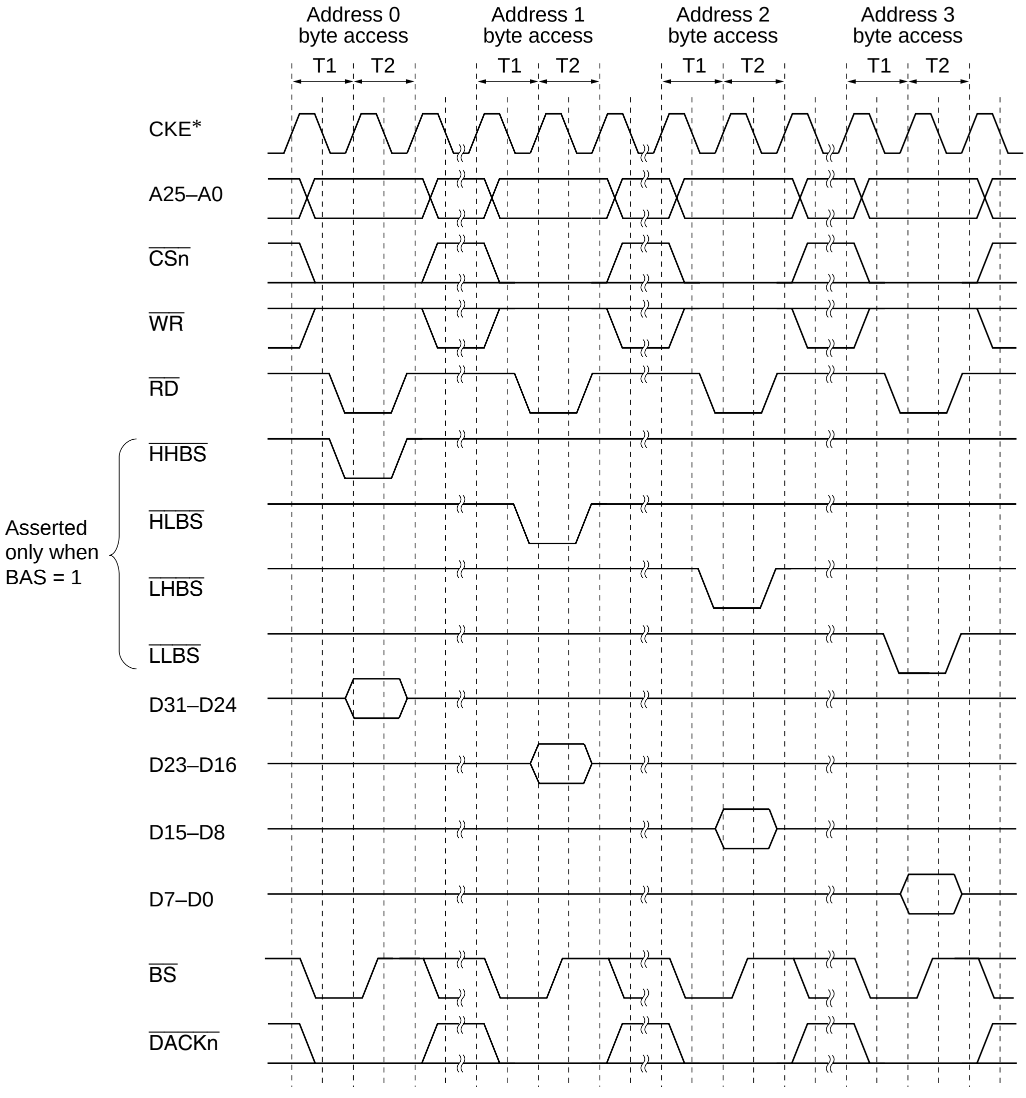
Making the appropriate setting for the BAS bit in BCR enables byte-strobe type 16-bit-width SRAM to be connected directly. When the BAS bit is cleared to 0, access is performed using the $\overline{\text{WRHH}}$, $\overline{\text{WRHL}}$, $\overline{\text{WRLH}}$, and $\overline{\text{WRLL}}$ signals. When the BAS bit is set to 1, access is performed using the $\overline{\text{WR}}$, $\overline{\text{HHBS}}$, $\overline{\text{HLBS}}$, $\overline{\text{LHBS}}$, and $\overline{\text{LLBS}}$ signals. Also, since the $\overline{\text{HHBS}}$, $\overline{\text{HLBS}}$, $\overline{\text{LHBS}}$, and $\overline{\text{LLBS}}$ signals are also asserted in read accesses, it is always possible to know which byte position is being accessed.

Figure 8.13 shows the timing for a 32-bit-bus-width, big-endian, no-wait write cycle, and figure 8.14 shows the timing for a read cycle.



Note: * When the setting CKE = CKIO is made in clock mode 0 to 3, 6, or 7, CKE is identical to CKIO on the timing chart.
In clock modes 4 and 5, the phases of CKE and CKIO do not coincide, but the relative relationship of the AC specifications is the same as in the other clock modes.

**Figure 8.13 Byte Access Control Timing
(32-Bit Bus Width, Big-Endian Mode, No Waits, Write Cycle)**



Note: * When the setting CKE = CKIO is made in clock mode 0 to 3, 6, or 7, CKE is identical to CKIO on the timing chart.
In clock modes 4 and 5, the phases of CKE and CKIO do not coincide, but the relative relationship of the AC specifications is the same as in the other clock modes.

Figure 8.14 Byte Access Control Timing (32-Bit Bus Width, Big-Endian Mode, No Waits, Read Cycle)

8.3.4 DRAM Interface

Direct Connection of DRAM

When area 4 or area 5 space is accessed, the target space is 64-Mbyte DRAM space, and the DRAM interface function can then be used to connect DRAM directly to the SH7065.

As $\overline{\text{CAS}}$ is used to control byte access, 2-$\overline{\text{CAS}}$ type 16-bit-width DRAMs can be connected.

In addition to normal read and write access modes, fast page mode is supported for burst access. EDO mode is similarly supported, enabling one-cycle access in burst mode, in particular.

Address Multiplexing

Address multiplexing is always performed in accesses to DRAM. This enables DRAM, which requires row and column address multiplexing, to be connected directly to the SH7065 without using an external address multiplexer circuit. Any of the eight multiplexing methods shown below can be selected, by setting bits AMX2 to AMX0 in DCR3. The relationship between bits AMX2 to AMX0 and address multiplexing is shown in table 8.10. The address output pins subject to address multiplexing are A15 to A0. The original address signals are output to pins A25 to A16.

Table 8.10 Relationship between Bits AMX2 to AMX0 and Address Multiplexing

AMX2	AMX1	AMX0	Number of Column Address Bits	Output Timing	External Address Pins						
					A0–A9	**A10**	**A11**	**A12**	**A13**	**A14**	**A15**
0	0	0	9 bits	Column address	A0–A9	A10	A11	A12	A13	A14	A15
				Row address	A9–A18	A19	A20	A21	A22	A23	A24
		1	10 bits	Column address	A0–A9	A10	A11	A12	A13	A14	A15
				Row address	A10–A19	A20	A21	A22	A23	A24	A25
	1	0	11 bits	Column address	A0–A9	A10	A11	A12	A13	A14	A15
				Row address	A11–A20	A21	A22	A23	A24	A25	A15
		1	12 bits	Column address	A0–A9	A10	A11	A12	A13	A14	A15
				Row address	A12–A21	A22	A23	A24	A25	A14	A15
1	0	0	13 bits	Column address	A0–A9	A10	A11	A12	A13	A14	A15
				Row address	A13–A22	A23	A24	A25	A13	A14	A15
		1	14 bits	Column address	A0–A9	A10	A11	A12	A13	A14	A15
				Row address	A14–A23	A24	A25	A12	A13	A14	A15
	1	0	15 bits	Column address	A0–A9	A10	A11	A12	A13	A14	A15
				Row address	A15–A24	A25	A11	A12	A13	A14	A15
		1	16 bits	Column address	A0–A9	A10	A11	A12	A13	A14	A15
				Row address	A16–A25	A10	A11	A12	A13	A14	A15

Basic Timing

The basic timing for DRAM access is 3 cycles. This basic timing is shown in figure 8.15. Tr is the $\overline{\text{RAS}}$ assert cycle, Tc1 the $\overline{\text{CAS}}$ assert cycle, and Tc2 the read data latch cycle.

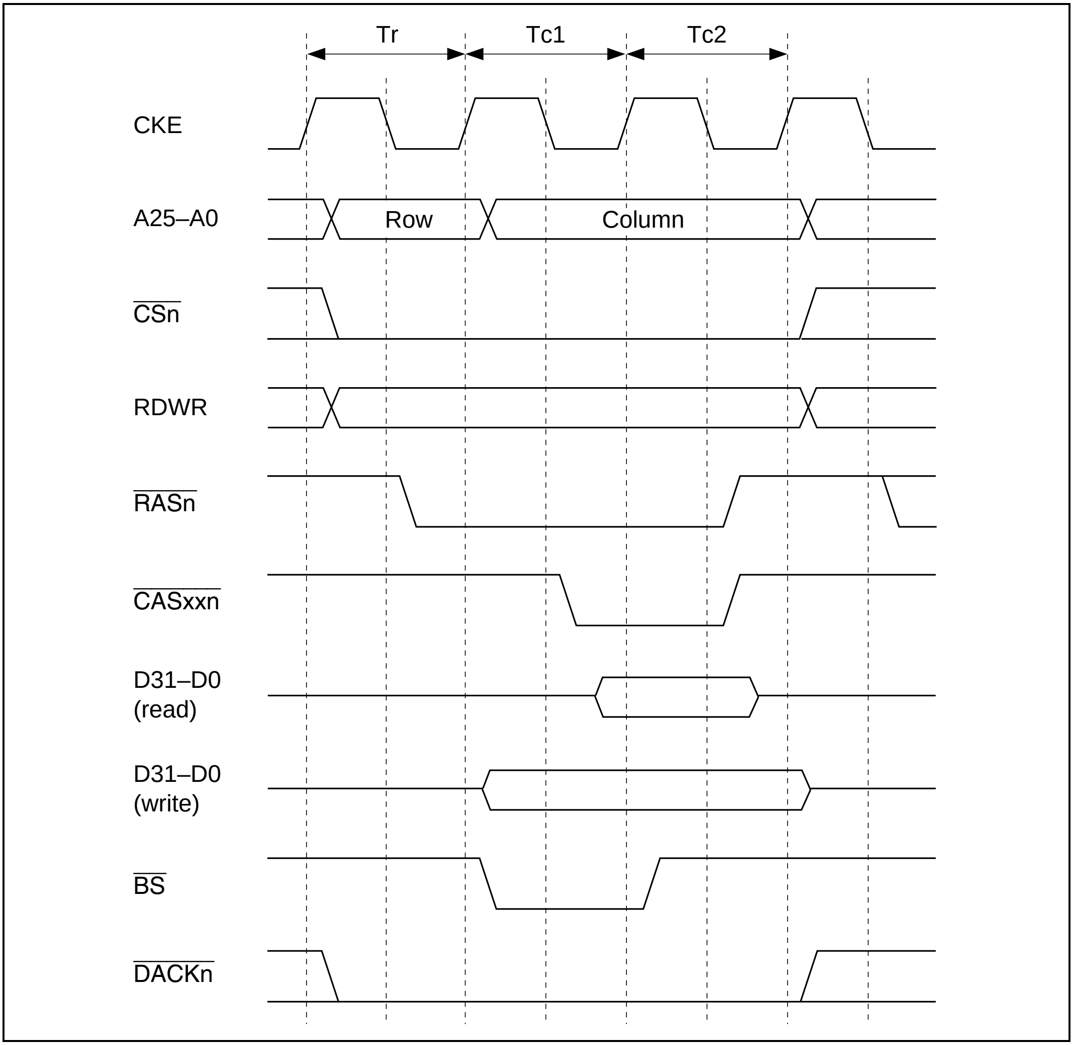


Figure 8.15 DRAM Basic Access Timing

Figures 8.16, 8.17, and 8.18 show examples of connection to 32-, 16-, and 8-bit data width DRAM.

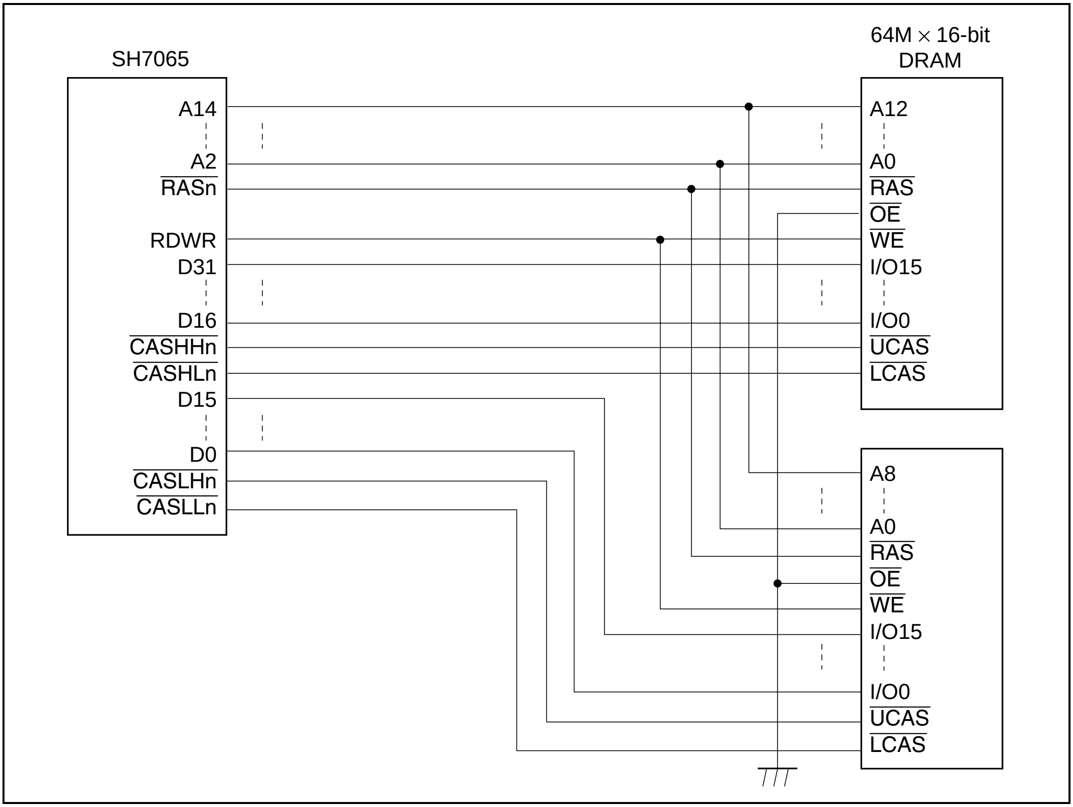


Figure 8.16 Example of 32-Bit Data Width DRAM Connection

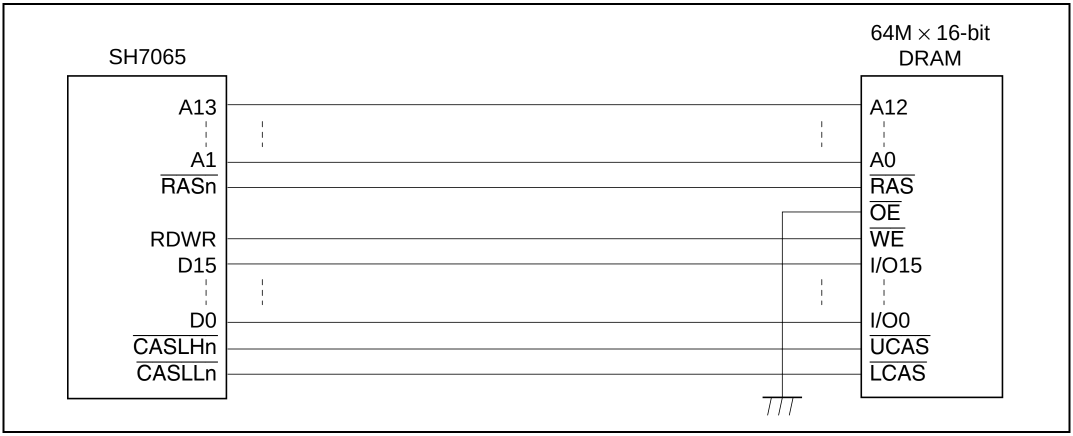


Figure 8.17 Example of 16-Bit Data Width DRAM Connection

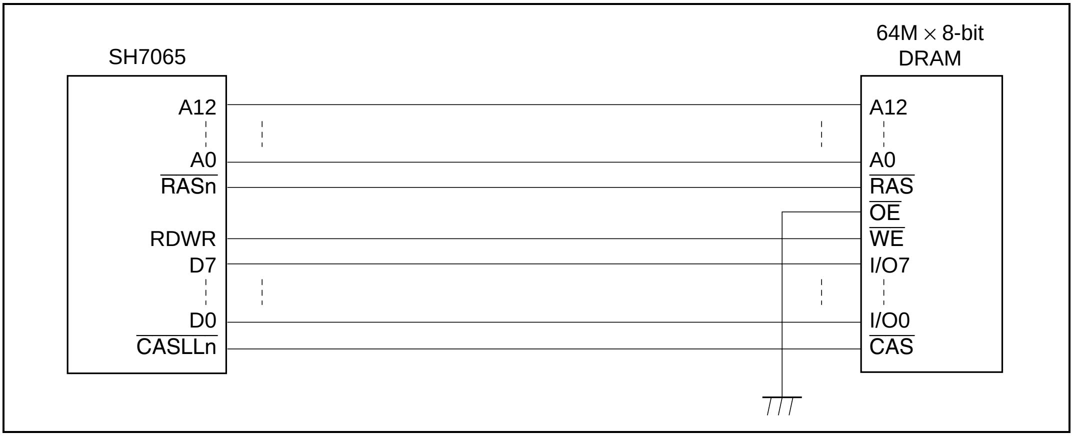


Figure 8.18 Example of 8-Bit Data Width DRAM Connection

Wait State Control

As the clock frequency increases, it becomes impossible to complete all states in one cycle as in the basic cycle. Therefore, provision is made for state extension by using setting bits in the DCR1 and DCR2 registers. The timing with state extension using these settings is shown in figure 8.19. Additional Tpc cycles (used to secure the $\overline{\text{RAS}}$ precharge time) can be inserted by means of the TPC bits in the DCR1 register; from 1 to 4 cycles can be selected. The number of cycles from $\overline{\text{RAS}}$ assertion to $\overline{\text{CAS}}$ assertion can be set to between 1 and 8 by inserting Trw cycles by means of the RCD bits in the DCR1 register. The number of cycles from $\overline{\text{CAS}}$ assertion to the end of the access can be varied, when reading, between 2 and 5 (1 cycle only in EDO mode) with the DWR bits in the DCR1 register, enabling $\overline{\text{CAS}}$ negation to be extended, and when writing, between 2 and 5 (1 cycle only in EDO mode) with the DWW bits in the DCR1 register, enabling $\overline{\text{CAS}}$ assertion to be extended. In a write, a $\overline{\text{CAS}}$ assertion width of 1 or 2 cycles can be set with the TCAS bit in the DCR2 register. Also, when TCAS = 1, the end of the write is extended by 1 cycle.

As with normal space, the wait input $\overline{\text{WAIT}}$ signal from an external source can be sampled by making the appropriate setting for the EXWE bit in ACR1. $\overline{\text{WAIT}}$ signal sampling is shown in figure 8.20. The signal is sampled at the rise of the clock in the Tc1 cycle.

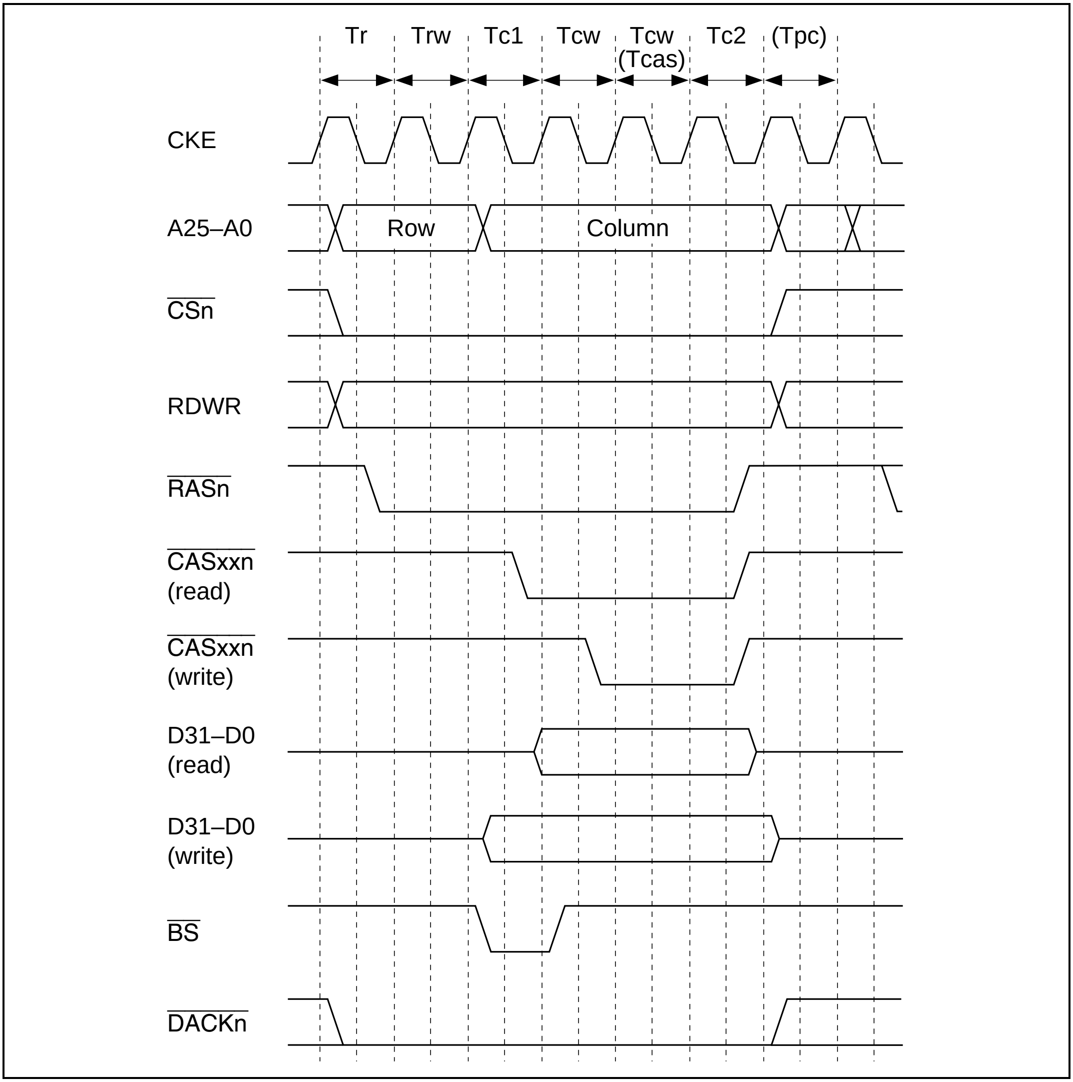


Figure 8.19 DRAM Wait State Timing
(Normal Mode, RCD = 1, TPC = 1, DWR = 2, DWW/TCAS = 1)

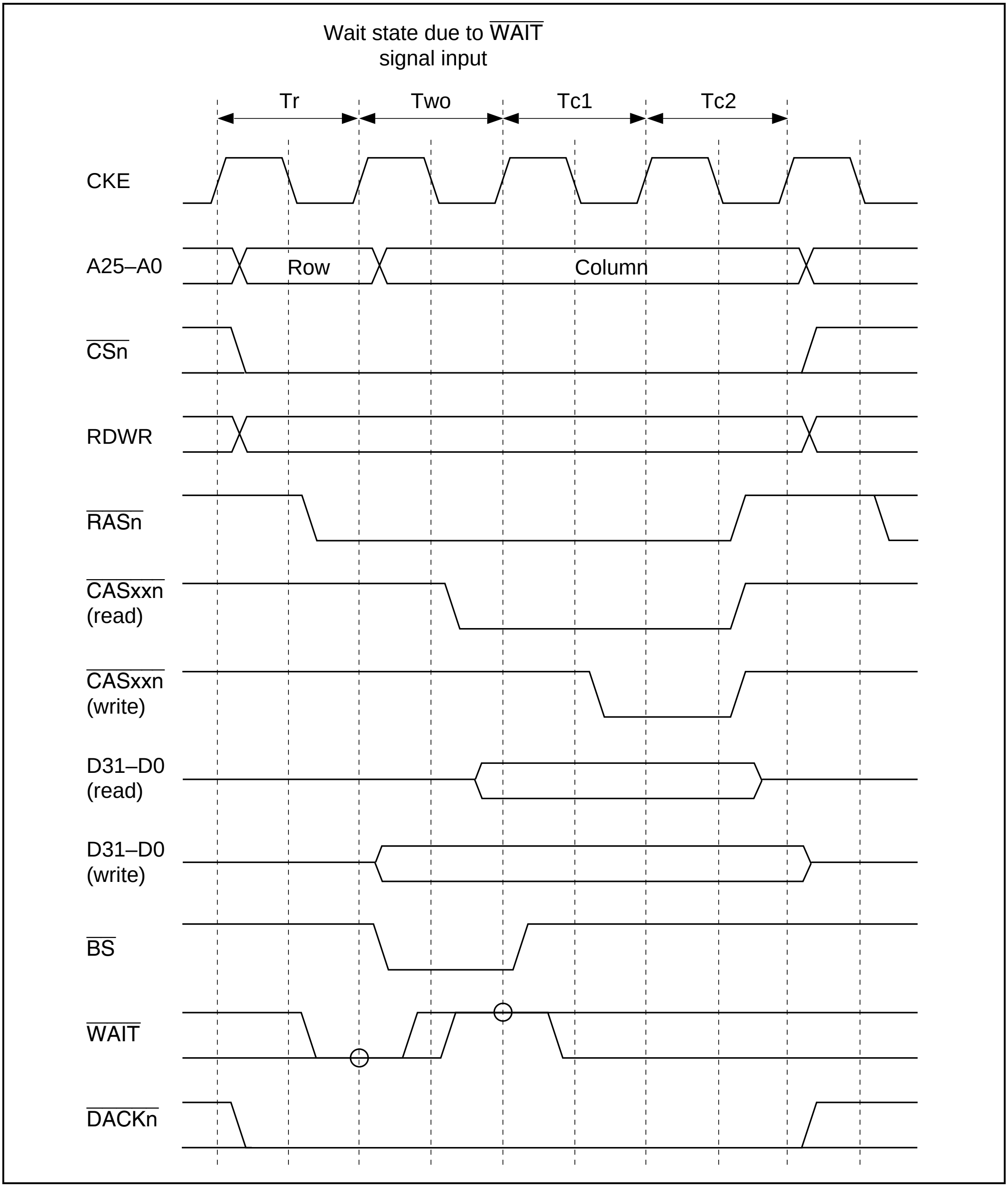


Figure 8.20 DRAM Basic Access Timing (Wait State Inserted by $\overline{\text{WAIT}}$ Signal)

Burst Access

n addition to the normal DRAM access mode in which a row address is output in each data access, a fast page mode is also provided for the case where consecutive accesses are made to the same row. This mode allows fast access to data by outputting the row address only once, then changing only the column address for each subsequent access. Normal access or burst access using fast page mode can be selected by means of the BE bit in DCR3. The timing for burst access using fast page mode is shown in figure 8.21.

Burst transfer is performed when the access width exceeds the bus width, or in single address transfer in burst mode by the DMAC.

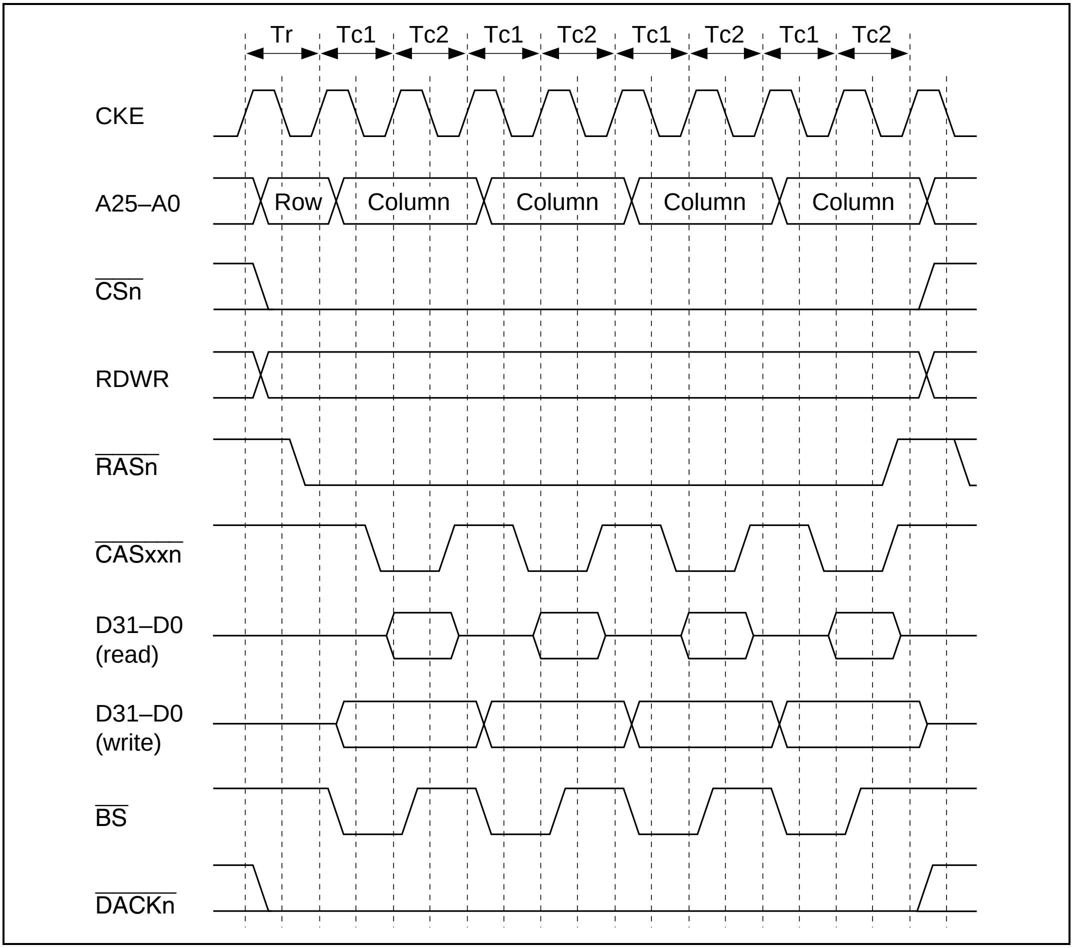


Figure 8.21 Basic Timing of DRAM Burst Access

EDO Mode

With DRAM, in addition to the mode in which data is output to the data bus only while the $\overline{\text{CAS}}$ signal is asserted in a data read cycle, an EDO mode is also provided in which, once the $\overline{\text{CAS}}$ signal is asserted while the $\overline{\text{RAS}}$ signal is asserted, even if the $\overline{\text{CAS}}$ signal is negated, data is output to the data bus until the $\overline{\text{CAS}}$ signal is next asserted. Either normal access/burst access using fast page mode, or EDO mode normal access/burst access, can be selected for DRAM with the EDO bit in DCR3. EDO mode normal access is shown in figure 8.22, and burst access in figure 8.23. In burst access, only one-cycle access is possible only when column addresses are consecutive. No-wait access must be used for EDO DRAM. No-wait access must be used for EDO DRAM, and wait state insertion by means of the $\overline{\text{WAIT}}$ pin must not be used.

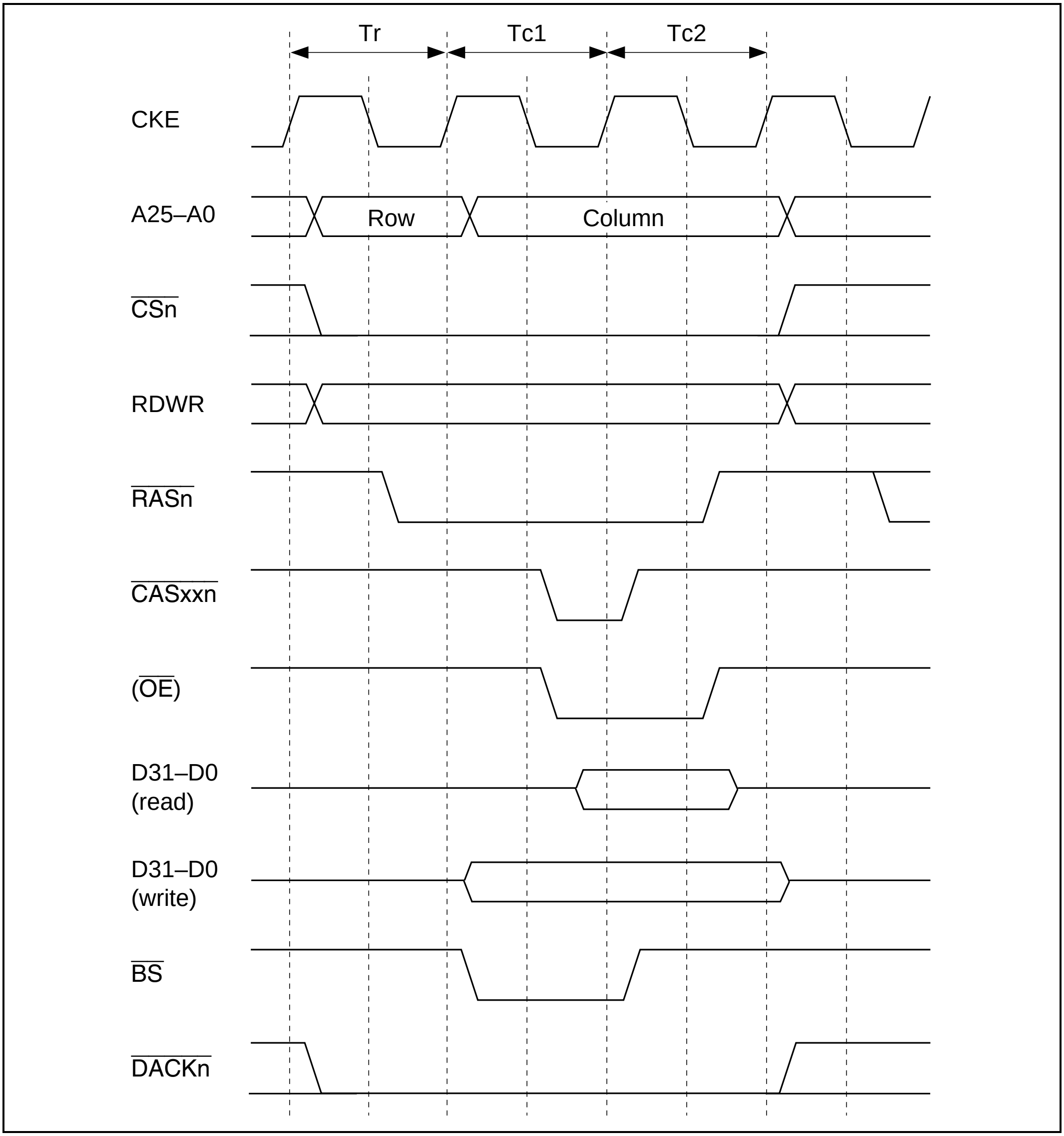


Figure 8.22 DRAM Basic Access Timing in EDO Mode

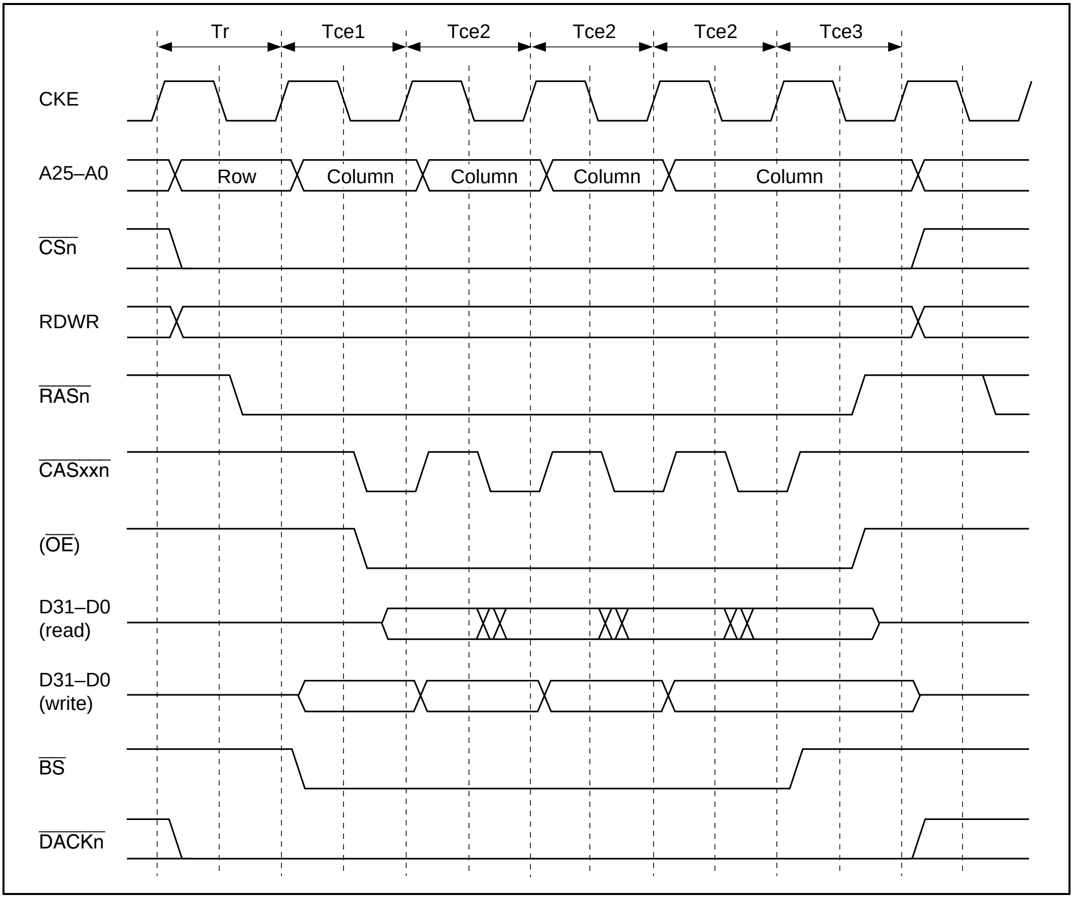


Figure 8.23 DRAM Burst Access Basic Timing in EDO Mode

RAS Down Mode

Even if burst operation is selected, it may happen that DRAM accesses are not consecutive, but are interrupted by an access to a different space. With the normal setting, the $\overline{\text{RAS}}$ signal is temporarily negated while a different space is being accessed, and must be reasserted to restart burst operation when DRAM is next accessed. This is known as RAS up mode. However, it is possible to keep the $\overline{\text{RAS}}$ signal asserted while a different space is being accessed, enabling burst operation to be continued when the same DRAM row address is next accessed. This is known as RAS down mode.

To use RAS down mode, set both BE and RASD to 1 in DCR3. When using RAS down mode to access DRAM in EDO mode, the $\overline{\text{OE}}$ signal must be connected to the SH7065. Figure 8.24 shows the timing in RAS up mode, and figure 8.25 the timing in RAS down mode

Setting the RDW bit in the DCR2 register enables an idle cycle to be inserted before burst operation when the same DRAM row address is accessed in DMA single address mode. The $\overline{\text{DACK}}$ signal is asserted during this idle cycle, facilitating DMA single transfer. Figure 8.26 shows an example of idle cycle insertion in RAS down mode when using EDO mode.

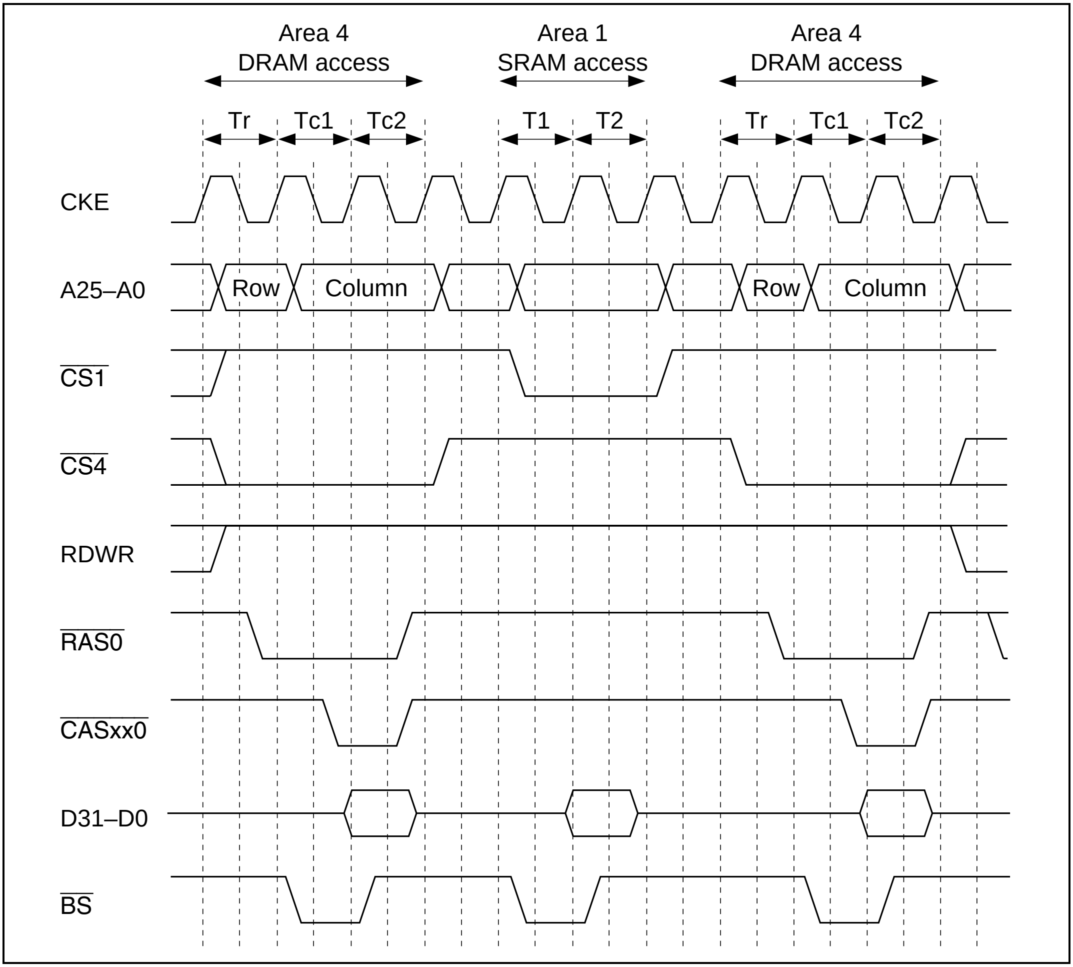


Figure 8.24 RAS Up Mode Basic Timing (Read Cycle)

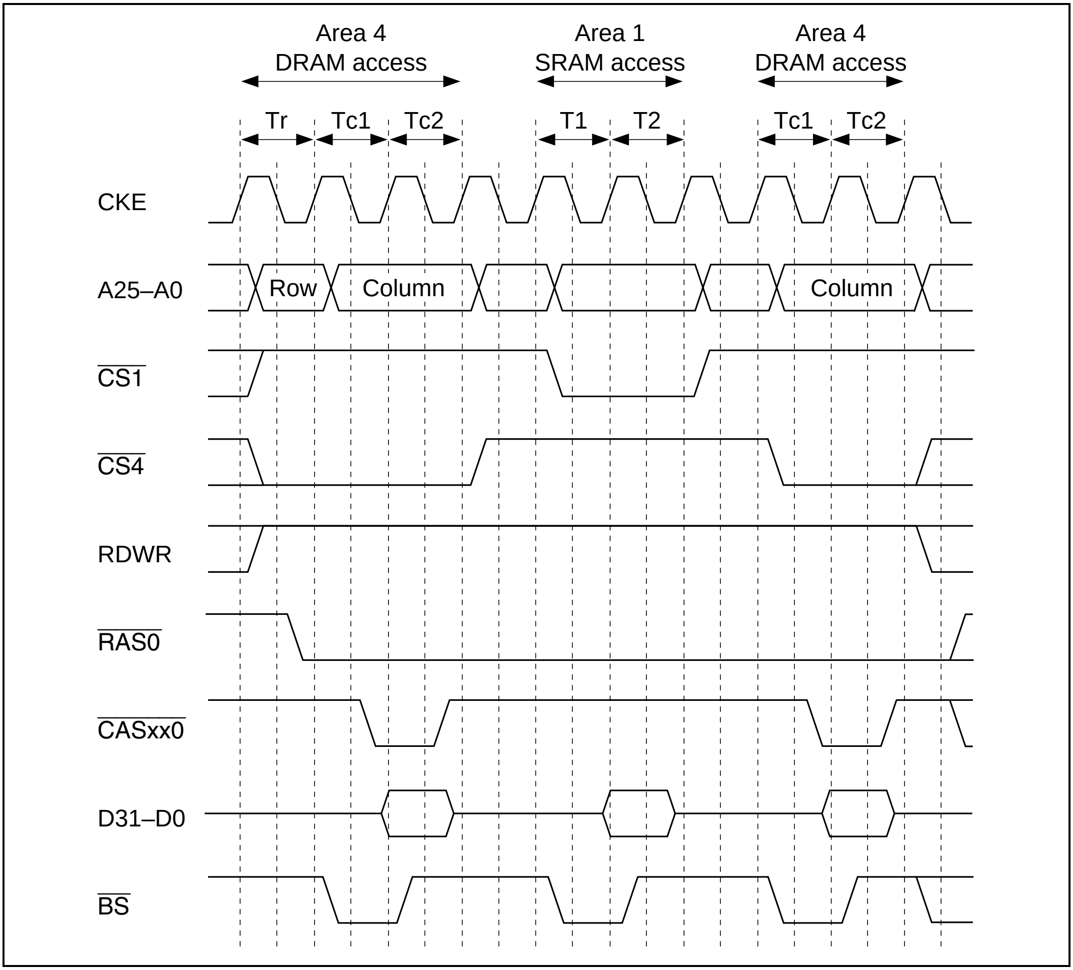


Figure 8.25 RAS Down Mode Basic Timing (Read Cycle)

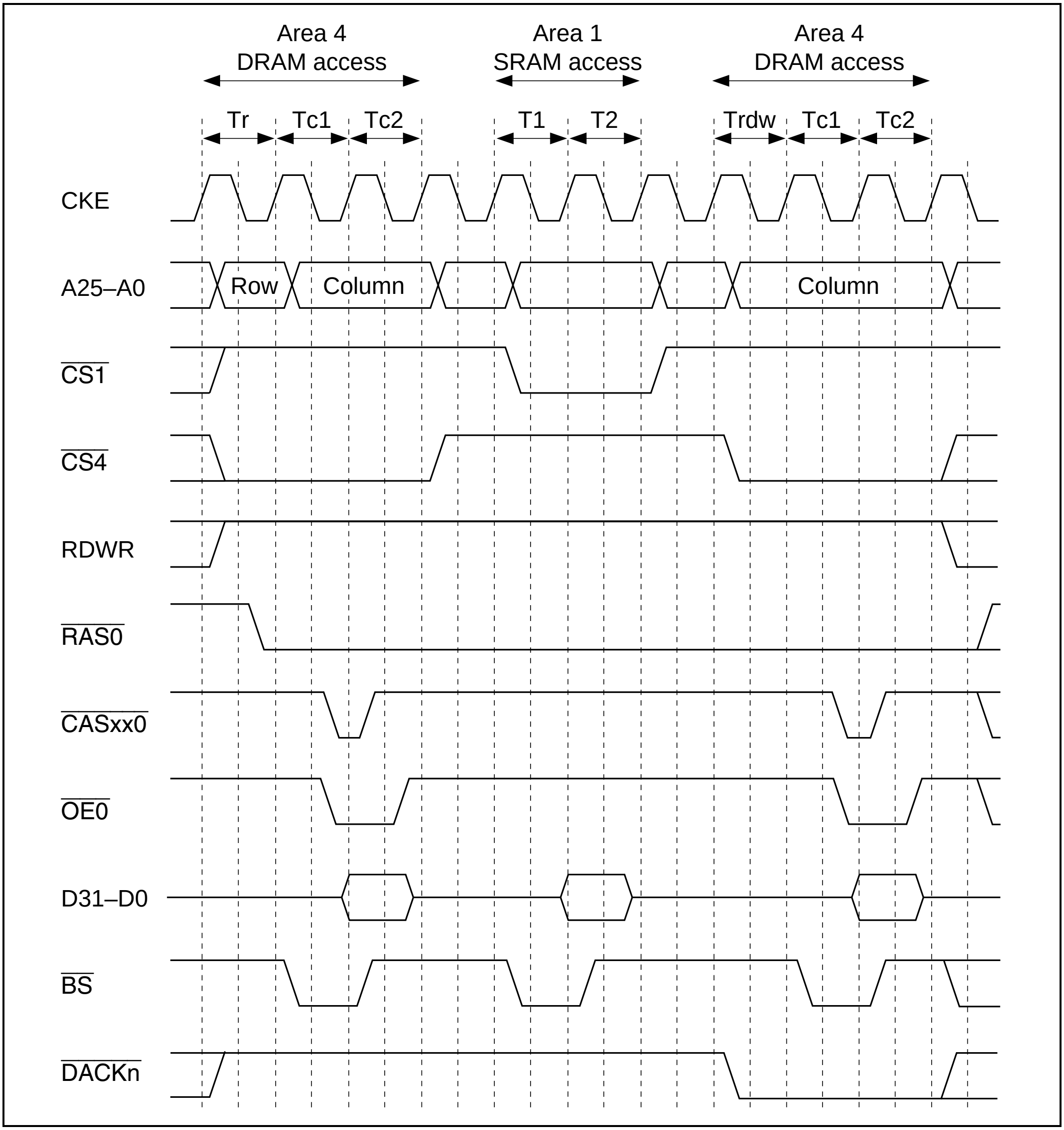


Figure 8.26 Example of RAS Down Mode Wait Timing (EDO Mode, Read Cycle, RDW = 1)

Refresh Timing

The bus state controller includes a function for controlling DRAM refreshing. Distributed refreshing using CAS-before-RAS refresh cycles can be performed for DRAM by clearing the RMD bit to 0 and setting the RFSH bit to 1 in DCR3. Self-refresh mode is also supported.

CAS-before-RAS Refreshing: When CAS-before-RAS refresh cycles are executed, refreshing is performed at intervals determined by the input clock selected by bits CKS2 to CKS0 in RTCSR, and the value set in RTCOR. The value of bits CKS2 to CKS0 in RTCOR should be set so as to satisfy the specification for the DRAM refresh interval. To change the input clock, first clear bits CKS0 to CKS2 to 0, then write the required value in these bits. When the clock is selected by bits 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. At the same time, RTCNT is cleared to zero and the count-up is restarted. After generation of the reference request, if the SH7065's external bus can be used, CAS-before-RAS refreshing is performed. A setting can be made in bits BREF2 to BREF0 in RTCSR to specify execution of from 1 to 8 consecutive CAS-before-RAS refreshes in response to a single refresh request. Figure 8.27 shows the operation of CAS-before-RAS refreshing, and figure 8.28 shows the timing of CMF bit setting.

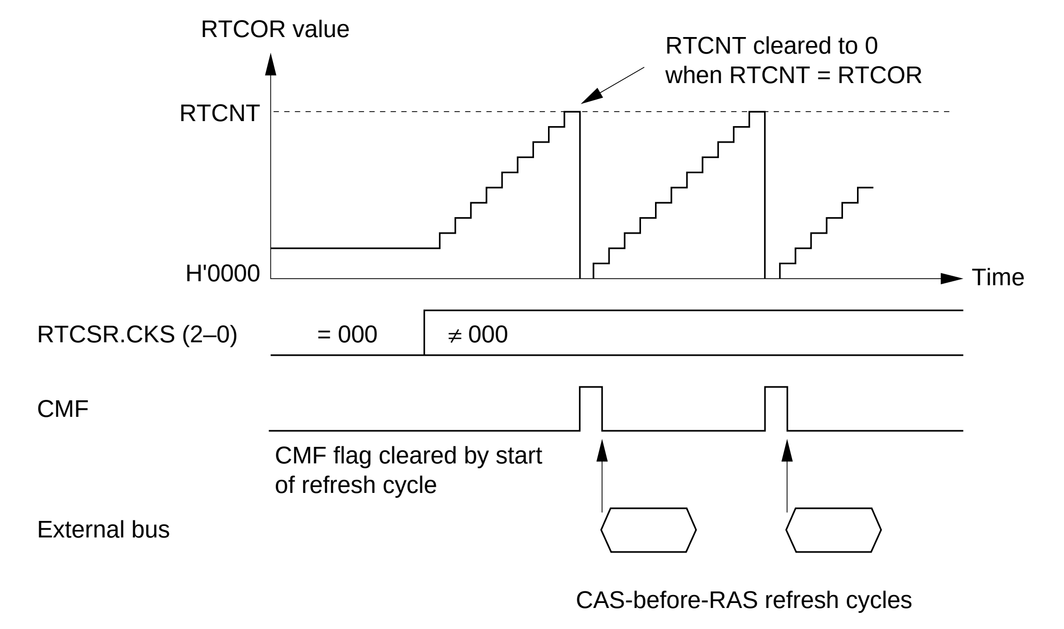


Figure 8.27 CAS-Before-RAS Refresh Operation

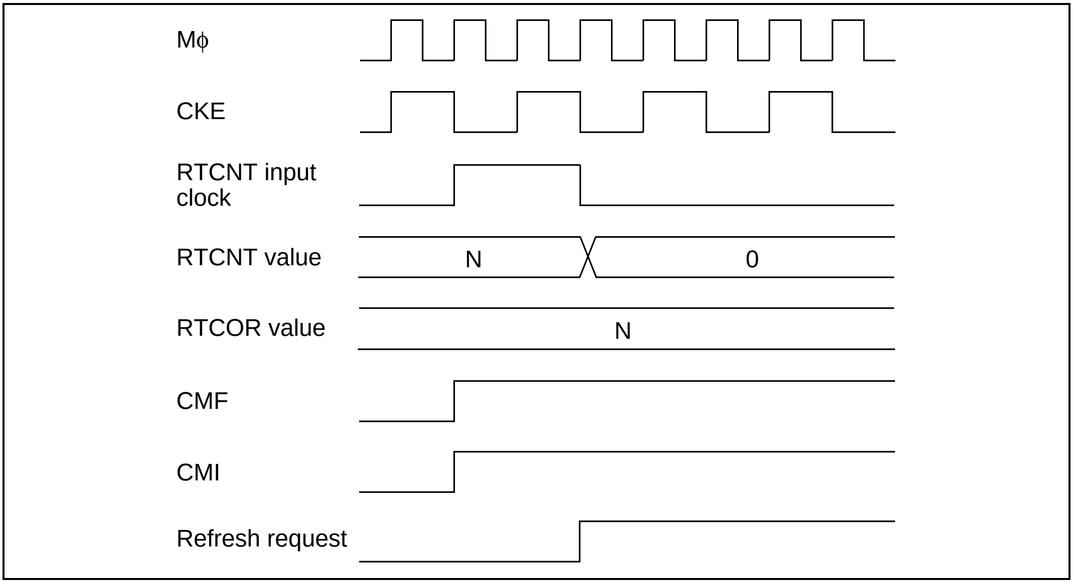


Figure 8.28 Timing of CMF Bit Setting (when M0: CKE = 1:1/2)

Figure 8.29 shows the timing of the CAS-before-RAS refresh cycle.

The number of RAS assert cycles in the refresh cycle is specified by the TRAS bits in RTCSR. The specification of the RAS precharge time in the refresh cycle is determined by the setting of the TPC bits in DCR1.

CAS-before-RAS refreshing is performed in normal operation, in sleep mode.

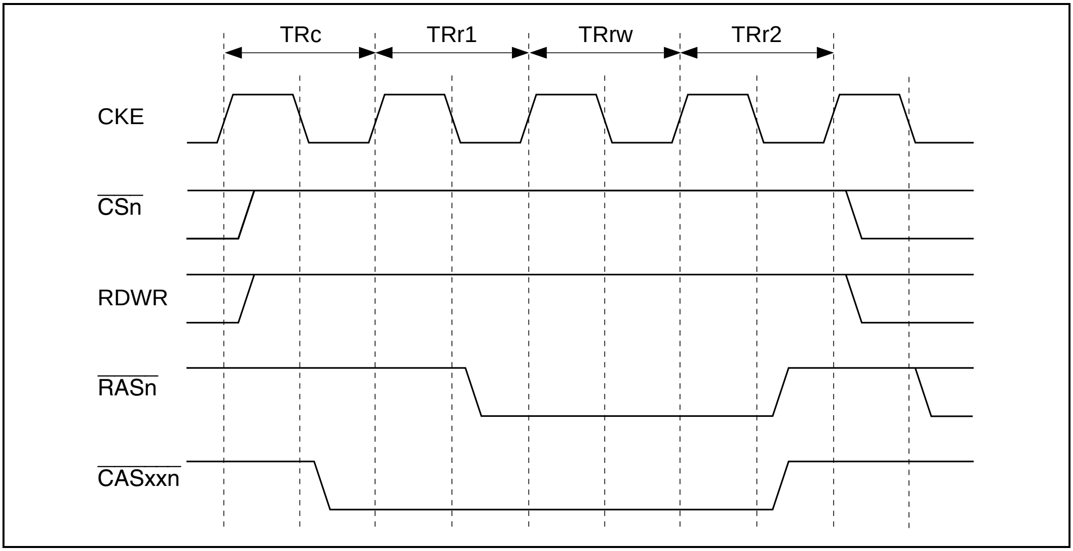


Figure 8.29 Basic Timing of DRAM CAS-Before-RAS Refresh Cycle

Self-Refreshing: The self-refreshing supported by the SH7065 is shown in figure 8.30.

A transition to self-refresh mode is made by setting the RFSH bit and RMD bit to 1 in DCR. Self-refresh mode is exited by clearing the RMD bit to 0 in DCR, than performing CAS-before-RAS refreshing on all row addresses within the time specified for the DRAM. The RAS precharge time immediately after the end of self-refreshing can be set with the TPCS bits in DCR. If there is a delay between clearing self-refreshing and the start of CAS-before-RAS refreshing, this must be taken into consideration when setting the initial RTCNT value. When the RTCNT value is set to the same value as RTCOR, a refresh request is issued immediately.

To protect DRAM data, the DRAM should not be accessed during self-refreshing. If DRAM is to be accessed during self-refreshing, first clear self-refreshing, then perform refreshing of all row addresses before making the access.

DRAMs include low-power products (L versions) with a long refresh cycle time (for example, the HM51W4160AL L version has a refresh cycle of 1024 cycles/128 ms compared with 1024 cycles/16 ms for the normal version). With these DRAMs, however, the same refresh cycle as for the normal version is requested only in the case of refreshing immediately following self-refreshing. To ensure efficient DRAM refreshing, therefore, processing is needed to generate an overflow interrupt and restore the refresh cycle to the proper value, after the necessary CAS-before-RAS refreshing has been performed following self-refreshing of an L-version DRAM, using the OVF, OVIE, and LMTS bits in RTCSR, and the refresh controller’s refresh count register (RFCR). The necessary procedure is as follows.

1. Normally, set the refresh counter count cycle to the optimum value for the L version (e.g. 1024 cycles/128 ms).
2. When a transition is made to self-refreshing:
    a. Provide an interrupt handler to restore the refresh counter count value to the optimum value for the L version (e.g. 1024 cycles/128 ms) when a refresh counter overflow interrupt is generated.
    b. Re-set the refresh counter count cycle to the requested short cycle (e.g. 1024 cycles/16 ms), set refresh controller overflow interruption, and clear the refresh controller's refresh count register (RFCR) to 0.
    c. Set self-refresh mode.

By using this procedure, the refreshing immediately following a self-refresh will be performed in a short cycle, and when refreshing ends, an interrupt is generated and the setting can be restored to the original refresh cycle.

Self-refreshing is performed in normal operation, in sleep mode, and in standby mode.

When the bus has been released in response to a bus arbitration request, or when a transition is made to standby mode, signals generally become high-impedance, but whether the $\overline{\text{RAS}}$ and $\overline{\text{CAS}}$ signals for DRAM in the self-refresh state become high-impedance or continue to be output can be controlled by the HIZCNT bit in BCR. The DRAM can be kept in the self-refresh state when the bus is released and in standby mode by setting the HIZCNT bit to 1. However, in this case, too, the DRAM should not be accessed during self-refreshing. Also, after self-refreshing is set, a bus request, self-refresh clearing, or execution of a SLEEP instruction involving a transition to software standby mode, should only be performed after another CS space has first been accessed.

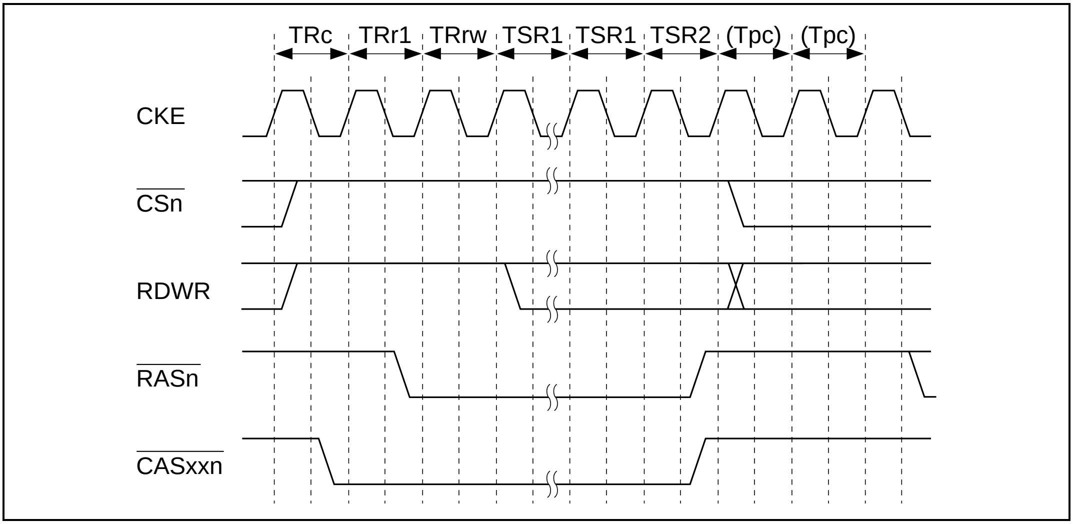


Figure 8.30 DRAM Self-Refresh Cycle Timing

Relationship between Refresh Requests and Bus Cycle Requests: If a refresh request is generated during execution of a bus cycle, execution of the refresh is deferred until the bus cycle is completed. If a refresh request occurs when the bus has been released by the bus arbiter, 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 normally, 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 $\overline{\text{IRQOUT}}$ pin is asserted (driven low). Therefore, normal refreshing can be performed by having the $\overline{\text{IRQOUT}}$ pin monitored by a bus master other than the SH7065 requesting the bus, or the bus arbiter, and returning the bus to the SH7065. When refreshing is started, and if no other interrupt request has been generated, the $\overline{\text{IRQOUT}}$ pin is negated (driven high). For details, see section 17.3.27, Function Control Register (FCR).

Power-On Sequence

Regarding use of DRAM after powering on, it is requested that a wait time (at least 100 µs or 200 µs) during which no access can be performed be provided, followed by the prescribed number (usually 8) or more dummy CAS-before-RAS refresh cycles. As the bus state controller does not perform any special operations for a power-on reset, the necessary power-on sequence must be carried out by the initialization program executed after a power-on reset.

8.3.5 Multiplexed Address/Data I/O Interface

Basic Timing

A function is provided that performs multiplexed input/output of an address and data on pins D15 to D0 when the appropriate setting is made in bits TP1 and TP0 of the ACR1 registers for areas 1 to 3. This allows a peripheral LSI that requires address/data multiplexing to be connected to the SH7065.

The bus width of multiplexed address/data I/O space is selected by the A14 bit. When A14 = 0, the data bus width is 8 bits; the address is output at pins D15 to D0 and data is input/output at pins D7 to D0. When A14 = 1, the address and data are both 16 bits, and address output and data input/output is performed at pins D15 to D0.

In multiplexed address/data I/O space access, normal space type access is carried out after address output has been performed for three cycles (fixed). The basic timing for multiplexed address/data I/O space is shown in figure 8.31.

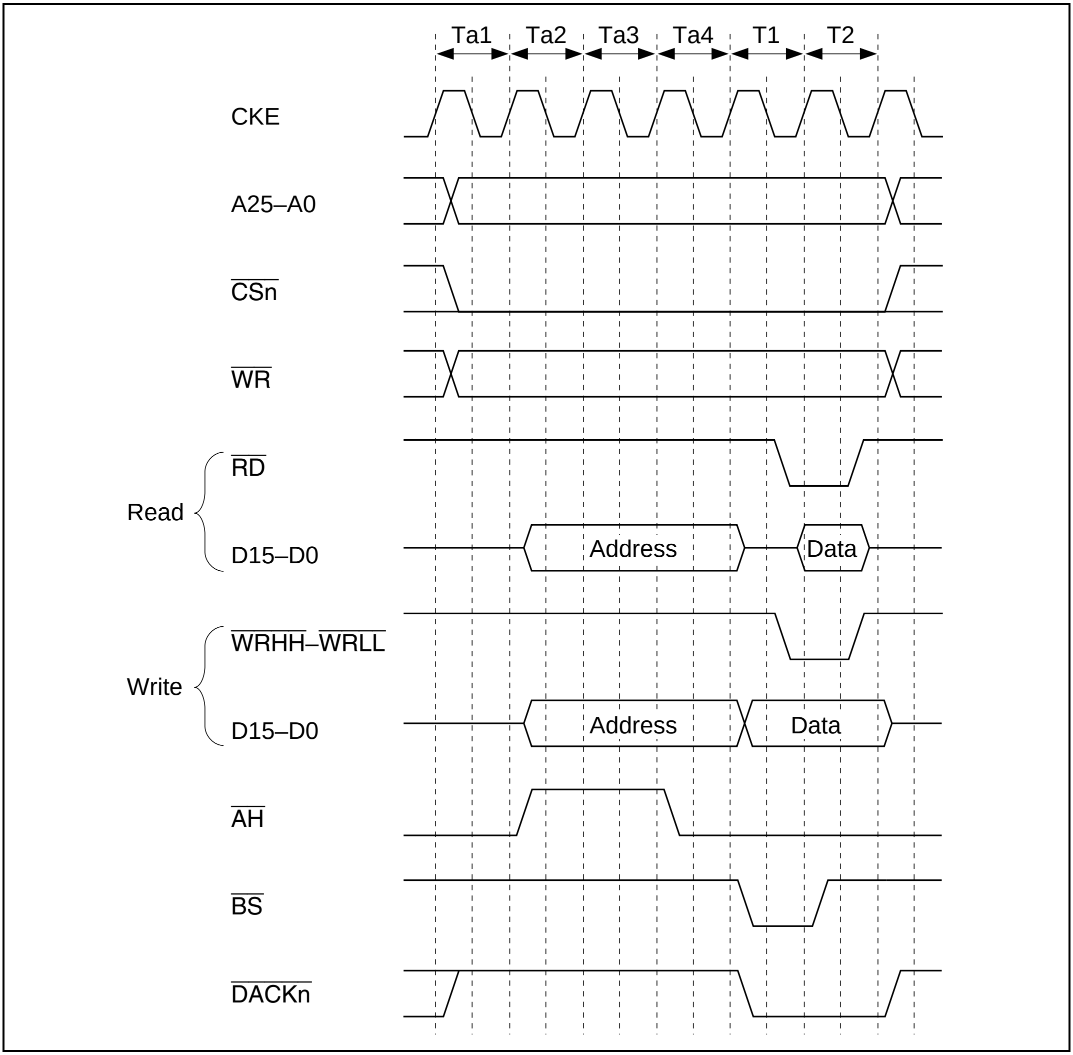


Figure 8.31 Basic Access Timing for Multiplexed Address/Data I/O Space

Wait State Control

Wait control for multiplexed address/data I/O space access is carried out by making the appropriate setting for bits W3 to W0 in WCR and the EXWE bit in ACR1. Software wait and external wait insertion timing is the same as for normal space access. Figure 8.32 the timing for two software wait insertion, and figure 8.33 shows the timing when one external wait is inserted, and then an additional software wait state is inserted after negation of the $\overline{\text{WAIT}}$ signal. Figure 8.34 shows the timing when extension of $\overline{\text{CS}}$ assertion has been set.

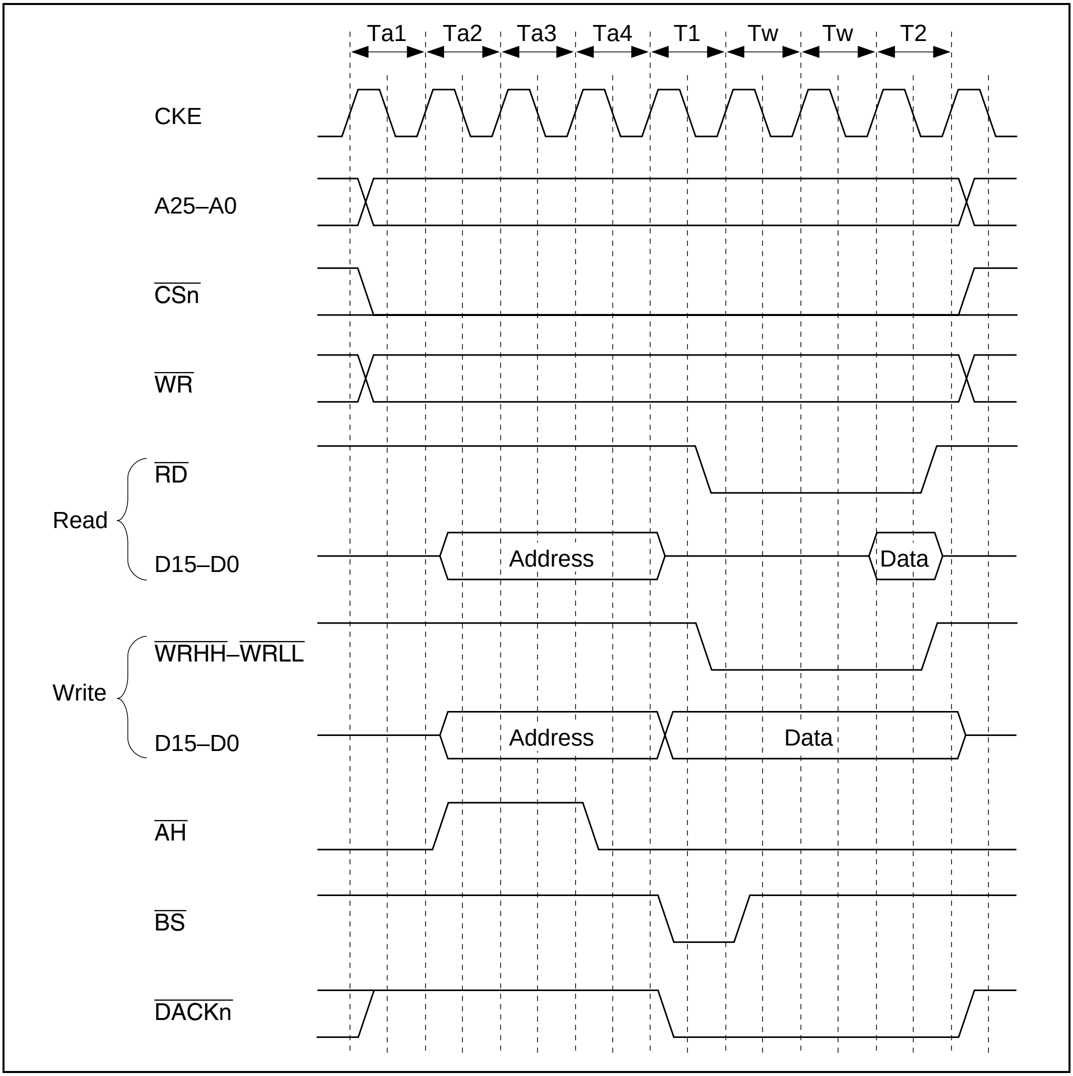


Figure 8.32 Wait State Timing for Multiplexed Address/Data I/O Space (Two Software Waits)

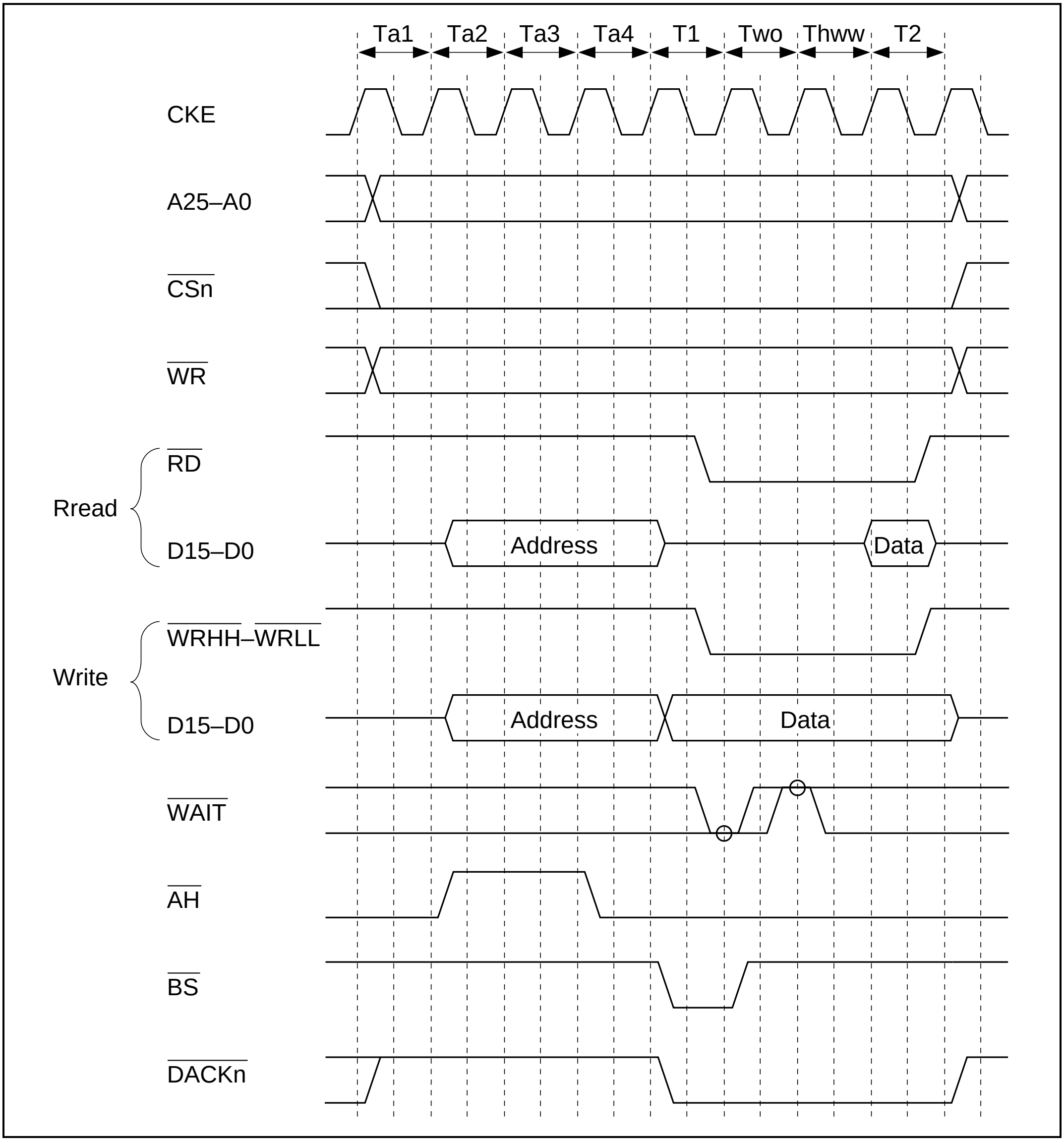


Figure 8.33 Wait State Timing for Multiplexed Address/Data I/O Space (Insertion of One External Wait + One Software Wait after $\overline{\text{WAIT}}$ Pin Negation)

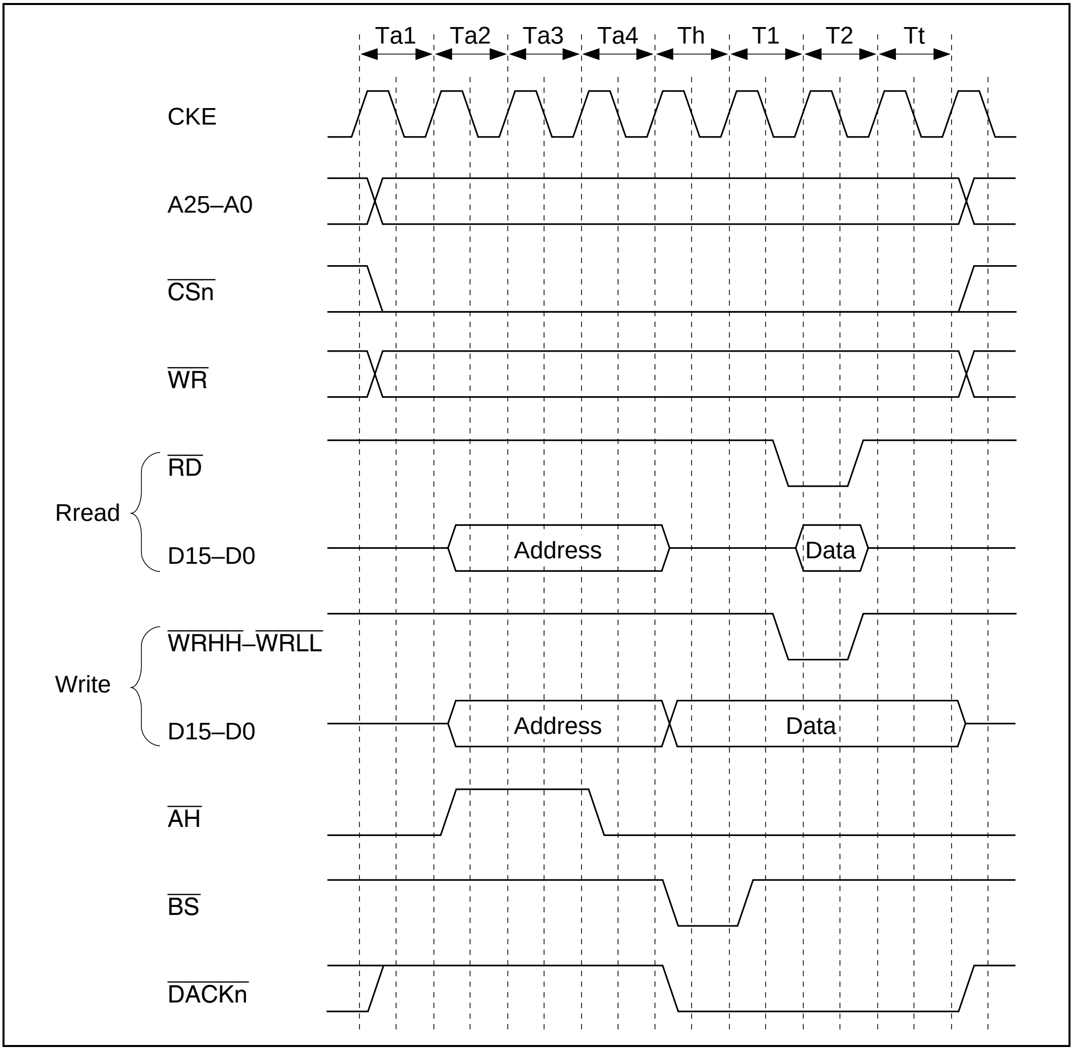


Figure 8.34 Timing when Extension of $\overline{\text{CS}}$ Assertion is Set (SWH = 1, SWT = 1)

8.3.6 Waits between Access Cycles

A problem associated with higher external memory bus operating frequencies is that data buffer turn-off on completion of a read from a low-speed device may be too slow, causing a collision with the data in the next access, and so resulting in lower reliability or incorrect operation. To avoid this problem, a data collision prevention feature has been provided. This memorizes the preceding access area and the kind of read/write, and if there is a possibility of a bus collision when the next access is started, inserts a wait cycle before the access cycle to prevent a data collision.

There are two cases in which wait cycles are inserted: (1) when an access is immediately followed by an access to a different area, and (2) when a read cycle access is immediately followed by a write access from the SH7065. When the SH7065 performs consecutive write cycles, the data transfer direction is fixed (from the SH7065 to other memory) and there is no problem. With read access to the same area, also, in principle data is assumed to be output from the same data buffer, and the set wait cycle insertion is not performed. Figure 8.35 shows the timing of waits between access cycles.

The number of idle cycles to be inserted between access cycles is specified by bits IW2 to IW0 in ACR1 and bits DIW2 to DIW0 in DCR2. If there is space between accesses to begin with, the number of idle cycles inserted is the specified number of idle cycles minus the number of empty cycles. When a write cycle is executed immediately after a read cycle, two wait cycles are inserted automatically between the cycles even if the inter-cycle wait specification is 0. When switching to access to a different space, also, one wait cycle is inserted automatically before a read cycle, and two wait cycles before a write cycle, even if “No idle cycles” is set. In the case of consecutive accesses to the same space, one wait cycle is inserted automatically in the case of a read cycle, and two wait cycles in the case of a write cycle, regardless of the inter-cycle wait setting.

When bus arbitration is performed, empty cycles are inserted for arbitration purposes, and so waits are not inserted between cycles.

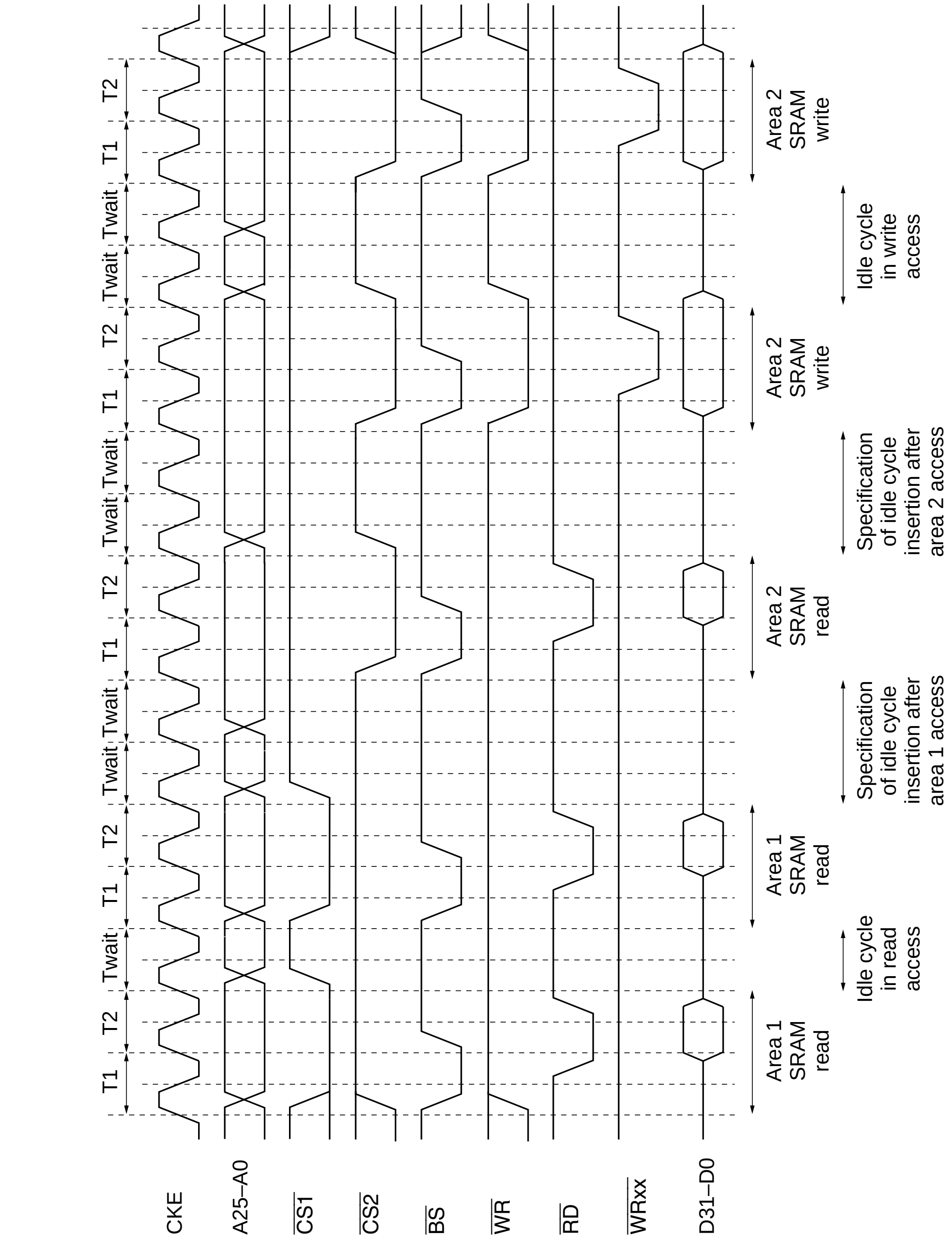


Figure 8.35 Example of Timing of Waits between Access Cycles (No Wait)

8.3.7 Bus Arbitration

When the bus release request signal ($\overline{\text{BREQ}}$) is asserted in accordance with the setting of the BRQE bit in BCR, the SH7065 releases the bus as soon as the currently executing bus cycle ends, and outputs the bus request acknowledge signal ($\overline{\text{BACK}}$). However, bus release is not performed between a read cycle and write cycle during execution of a TAS instruction (unless the destination of TAS instruction execution is on-chip RAM). Also, bus arbitration is not performed between bus cycles generated due to the fact that the data bus width is smaller than the access size, such as when a longword access is made to 8-bit memory. When $\overline{\text{BREQ}}$ is negated, $\overline{\text{BACK}}$ is negated and use of the bus is resumed. See Appendix B.1, Pin States in Reset, Power-Down State, and Bus-Released State, for the pin states when the bus is released.

Sometimes, the SH7065 may want to take back the bus while in the process of releasing it. This happens if a memory refresh request is generated internally, or an interrupt is requested, and the relevant processing must be executed. For this reason, the SH7065 is provided with an $\overline{\text{IRQOUT}}$ pin to output a bus request signal. If the SH7065 needs to take back the bus, it asserts the $\overline{\text{IRQOUT}}$ signal. On receiving this $\overline{\text{IRQOUT}}$ signal assertion, the device that asserted the external bus request negates the $\overline{\text{BREQ}}$ signal in order to release the bus. The bus is thereby returned to the SH7065, which then carries out the necessary processing. Note that if the device that asserted the external bus request does not return the bus within the time specified as the DRAM refresh interval, the SH7065 will not be able to carry out refreshing, and RAM contents may be lost. There are two cases in which the $\overline{\text{IRQOUT}}$ pin is asserted: (1) when a memory refresh request has been issued and the refresh cycle has not yet begun, and (2) when an interrupt source occurs and the interrupt request level is higher than that set in the interrupt mask bits (I3 to I0) in the status register (SR).

The SH7065 has two internal bus masters: the CPU and the DMAC. When DRAM is connected and refresh control is performed, refresh requests constitute a third bus master. In addition to these are bus requests from external devices. If requests occur simultaneously, priority is given, in high-to-low order, to a refresh request, a bus request from an external device, the DMAC, and the CPU. If an external space access request by the CPU or DMAC and a bus request by an external device occur, in that order, during execution of a refresh cycle, acceptance of the bus request by the external device will be delayed until the refresh cycle and external space access have been executed. Similarly, if an external space access request by the CPU or DMAC and a refresh request occur, in that order, execution of the refresh cycle after the SH7065 acquires the bus will be delayed until the external space access has been executed.

Bus requests from off-chip are not accepted in sleep mode.

If $\overline{\text{BREQ}}$ is asserted in sleep mode and the DMAC is subsequently activated, external access by the DMAC is delayed until $\overline{\text{BREQ}}$ is negated.

In the software standby state, external bus address/data/bus control signals (except DRAM signals) go to the high-impedance state, that is, the bus-released state. In the software standby state, the $\overline{\text{BREQ}}$ bus release request input signal is ignored. Note that the following two cases apply to the $\overline{\text{BACK}}$ bus use enable output signal.

1. Transition from bus-released state ($\overline{\text{BREQ}}$ input asserted low) to software standby state

   When the bus release request signal ($\overline{\text{BREQ}}$) is asserted low in the normal state, the $\overline{\text{BACK}}$ pin is set to low output, indicating that the bus has been released. If the software standby state is entered in this state, $\overline{\text{BACK}}$ output goes high, but other address, data, and bus control signals remain in the high-impedance state, that is, the bus-released state. If the software standby state is exited while $\overline{\text{BREQ}}$ input is still asserted, $\overline{\text{BACK}}$ output goes low and the bus-released state is maintained. If software standby is exited while $\overline{\text{BREQ}}$ input is negated, $\overline{\text{BACK}}$ output goes high and the chip returns to the normal state (in which the bus is not released).

2. Transition from normal state ($\overline{\text{BREQ}}$ input negated high) to software standby state

   When a transition is made from the normal state to the software standby state, $\overline{\text{BACK}}$ output goes to the Z (high-impedance) state, and the external bus goes to the high-impedance state, that is, the bus-released state. If this state is exited while $\overline{\text{BREQ}}$ input is negated, $\overline{\text{BACK}}$ output returns to the high level. If $\overline{\text{BREQ}}$ input is in the asserted state when software standby is exited, $\overline{\text{BACK}}$ is output high for 1.5 external clock (CKE) cycles, and then returns to the low level, that is, the bus-released state.

   When DMAC transfer is specified without regard to transfer space or transfer mode during execution of a TAS instruction (unless the destination of TAS instruction execution is on-chip RAM), DMA transfer cycles are inserted between a read and write cycles of the TAS instruction. In this case, if the bus release request signal $\overline{\text{BREQ}}$ is asserted, bus authority is released. All of the DMAC channels should be stopped before the execution of a TAS instruction, when the bus release request can be occurred during execution of the TAS instruction. ($\overline{\text{BREQ}}$ is not accepted during execution of a TAS instruction unless DMA transfer cycles occur during the execution of the TAS instruction.)

8.4 Number of Access Cycles (SH7065A)

External Memory and External I/O

Table 8.11 shows the number of external access cycles for Mϕ:CKE division ratios of 1:1, 1:1/2, and 1:1/4. The CPU regards an external space write as being executed in one cycle, and performs the next processing. However, the write actually takes the number of cycles shown in table 8.11. Therefore, execution of an on-chip register or external access following an external space write by the CPU is delayed until the end of the external space write.

Table 8.12 shows the number of idle cycles. For the number of accesses on a CKE basis, add this number of idle cycles to the number of external bus idle cycles.

Table 8.11 Number of External Access Cycles (Mϕ Basis)

		Bus Master	
Mϕ:CKE Division Ratio	**Read/Write**	**Cycles in Access from CPU**	**Cycles in Access from DMAC**
1:1	Read	Number of external bus cycles + 3	Number of external bus cycles + 1
	Write	Number of external bus cycles + 4	Number of external bus cycles + 2
1:1/2	Read	(Number of external bus cycles) × 2 + (4 or 5)*	(Number of external bus cycles) × 2 + (2 or 3)*
	Write	(Number of external bus cycles) × 2 + (6 or 7)*	(Number of external bus cycles) × 2 + (4 or 5)*
1:1/4	Read	(Number of external bus cycles) × 4 + (5 to 8)*	(Number of external bus cycles) × 4 + (3 to 6)*
	Write	(Number of external bus cycles) × 4 + (9 to 12)*	(Number of external bus cycles) × 4 + (7 to 10)*

Note: * Depends on the phase difference between Mϕ and CKE due to frequency division.

Table 8.12 Number of Idle Cycles in Consecutive Accesses to External Space (CKE Basis)

			Mϕ:CKE = 1:1				Mϕ:CKE = 1:1/2				Mϕ:CKE = 1:1/4			
Type of Access		**Number of Waits Set by Idle Function***	**CPU → CPU**	**CPU → DMAC**	**DMAC → CPU**	**DMAC → DMAC**	**CPU → CPU**	**CPU → DMAC**	**DMAC → CPU**	**DMAC → DMAC**	**CPU → CPU**	**CPU → DMAC**	**DMAC → CPU**	**DMAC → DMAC**
Consecutive accesses to same CS space	Read → Read	Invalid	3	2	2	1	2	2	2	1	2	1	1	1
	Read → Write	0	4	3	3	2	3	3	3	2	3	2	2	2
		1	4	3	3	2	3	3	3	2	3	2	2	2
		2	4	3	3	2	3	3	3	2	3	2	2	2
		3	4	3	3	3	3	3	3	3	3	3	3	3
		4	4	4	4	4	4	4	4	4	4	4	4	4
		5	5	5	5	5	5	5	5	5	5	5	5	5
		6	6	6	6	6	6	6	6	6	6	6	6	6
		7	7	7	7	7	7	7	7	7	7	7	7	7
	Write → Read	Invalid	2	2	2	1	2	2	2	1	1	1	1	1
	Write → Write	Invalid	3	3	3	2	3	3	3	2	2	2	2	2
Consecutive access to other CS space	Read → Read	0	3	2	2	1	2	2	2	1	2	1	1	1
		1	3	2	2	1	2	2	2	1	2	1	1	1
		2	3	2	2	2	2	2	2	2	2	2	2	2
		3	3	3	3	3	3	3	3	3	3	3	3	3
		4	4	4	4	4	4	4	4	4	4	4	4	4
		5	5	5	5	5	5	5	5	5	5	5	5	5
		6	6	6	6	6	6	6	6	6	6	6	6	6
		7	7	7	7	7	7	7	7	7	7	7	7	7
	Read → Write	0	4	3	3	2	3	3	3	2	3	2	2	2
		1	4	3	3	2	3	3	3	2	3	2	2	2
		2	4	3	3	2	3	3	3	2	3	2	2	2
		3	4	3	3	3	3	3	3	3	3	3	3	3
		4	4	4	4	4	4	4	4	4	4	4	4	4
		5	5	5	5	5	5	5	5	5	5	5	5	5
		6	6	6	6	6	6	6	6	6	6	6	6	6
		7	7	7	7	7	7	7	7	7	7	7	7	7
	Write → Read	0	2	2	2	1	2	2	2	1	1	1	1	1
		1	2	2	2	1	2	2	2	1	1	1	1	1
		2	2	2	2	2	2	2	2	2	2	2	2	2
		3	3	3	3	3	3	3	3	3	3	3	3	3
		4	4	4	4	4	4	4	4	4	4	4	4	4
		5	5	5	5	5	5	5	5	5	5	5	5	5
		6	6	6	6	6	6	6	6	6	6	6	6	6
		7	7	7	7	7	7	7	7	7	7	7	7	7
	Write → Write	0	3	3	3	2	3	3	3	2	2	2	2	2
		1	3	3	3	2	3	3	3	2	2	2	2	2
		2	3	3	3	2	3	3	3	2	2	2	2	2
		3	3	3	3	3	3	3	3	3	3	3	3	3
		4	4	4	4	4	4	4	4	4	4	4	4	4
		5	5	5	5	5	5	5	5	5	5	5	5	5
		6	6	6	6	6	6	6	6	6	6	6	6	6
		7	7	7	7	7	7	7	7	7	7	7	7	7

Note: * Number set by bits IW2 to IW0 in ACR1 and bits DIW2 to DIW0 in DCR2

On-Chip Registers

In BSC, UBC, WDT, INTC, CPG, DMAC, PFC, I/O, flash memory related, and power-down related register access: Table 8.13 shows the number of access cycles. The CPU regards an on-chip register write as being executed in one cycle, and performs the next processing. However, the write actually takes the number of cycles shown in table 8.13. When a value written to an on-chip register is to be used by a later instruction, either read the written value or else wait for the number of cycles shown in table 8.13, before executing that later instruction. Execution of an on-chip register or external access following an on-chip register write by the CPU is delayed until the end of the on-chip register write.

Table 8.13 Number of Access Cycles in BSC, UBC, WDT, INTC, CPG, DMA, PFC, I/O, Flash Memory Related, and Power-Down Related Register Access

		Bus Master	
Operand Size	**Read/Write**	**Cycles in Access from CPU**	**Cycles in Access from DMAC**
Word/byte*	Read	5	3
	Write	6	4
Longword*	Read	8	6
	Write	9	7

Note: * Only byte access in the case of flash memory related registers.

A/D, D/A, TPU, MMT, CMT, POE, and SCI internal register access: Table 8.14 shows the number of access cycles for Mϕ:Pϕ division ratios of 1:1, 1:1/2, and 1:1/3. The CPU regards an on-chip register write as being executed in one cycle, and performs the next processing. However, the write actually takes the number of cycles shown in table 8.14. When a value written to an on-chip register is to be used by a later instruction, either read the written value or else wait for the number of cycles shown in table 8.14, before executing that later instruction. Execution of an on-chip register or external access following an on-chip register write by the CPU is delayed until the end of the on-chip register write.

Table 8.14 Number of Access Cycles in A/D, D/A, TPU, MMT, CMT, POE, and SCI Internal Register Access

			Bus Master	
Mϕ:Pϕ Division Ratio	**Operand Size**	**Read/ Write**	**Cycles in Access from CPU**	**Cycles in Access from DMAC**
1:1	Word/byte*1	Read	7	5
		Write	7	5
	Longword*2	Read	9	7
		Write	9	7
1:1/2	Word/byte*1	Read	9, 10*3	7, 8*3
		Write	9, 10*3	7, 8*3
	Longword*2	Read	13, 14*3	11, 12*3
		Write	13, 14*3	11, 12*3
1:1/3	Word/byte*1	Read	12–14*3	10–12*3
		Write	12–14*3	10–12*3
	Longword*2	Read	18–20*3	16–18*3
		Write	18–20*3	16–18*3

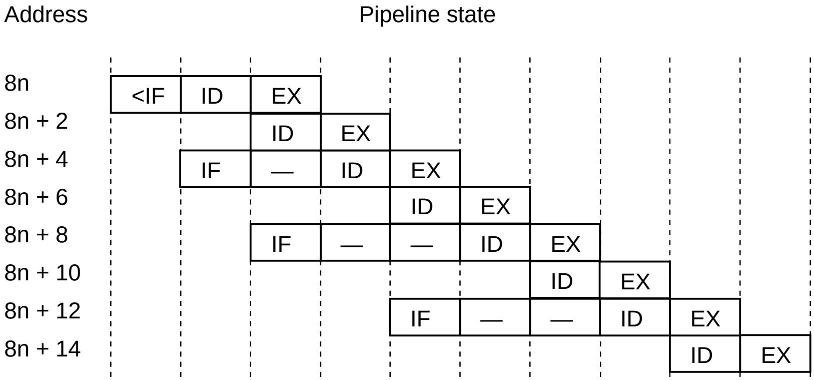
Notes: 1. Only byte access applies in the case of A/D and D/A registers.
2. Only word access applies in the case of A/D and D/A registers.
3. Depends on the phase difference between Mϕ and Pϕ due to frequency division.

On-Chip ROM

- In low-speed mode
  All 2 cycles

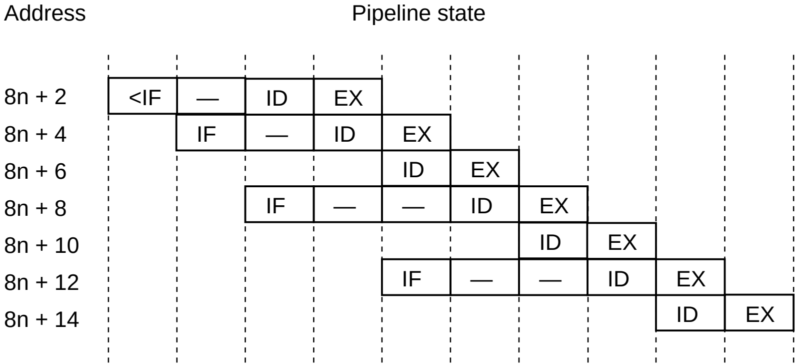
- In high-speed mode
  — Consecutive instruction fetch cycles
    1 cycle (However, in a branch to address 8n+4 or 8n+6, consecutive instruction fetch cycles immediately after the branch instruction fetch cycle comprise two cycles.)
  — Branch instruction fetch cycle
    2 to 3 cycles*
  — Data read cycle
    2 to 3 cycles*

Note: * The number of cycles depends on the state of the CPU pipeline, and buffering between the internal 32-bit data bus (CDB) and the 64-bit internal data ROM bus.
Figures 8.36 to 8.43 show the CPU pipeline state and the number of on-chip ROM access cycles when no on-chip ROM data read cycles are generated.



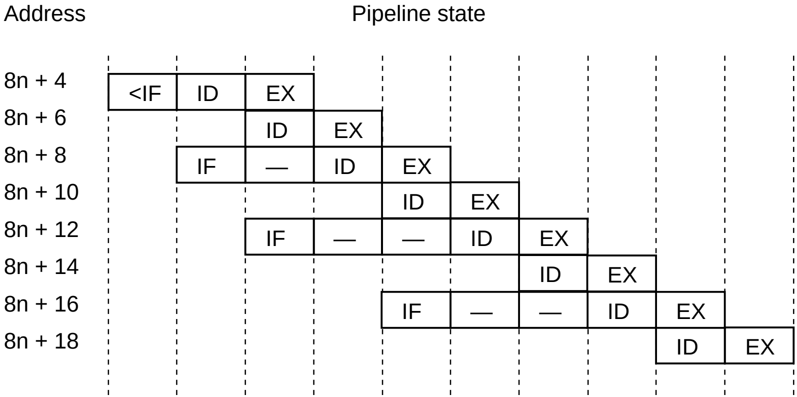
Access		fetch	fetch	fetch	nop	fetch	nop	fetch	nop	fetch	nop
Number of cycles	In low-speed mode	2	2	2	1	2	1	2	1	2	1
	In high-speed mode	2, 3	1	1	1	1	1	1	1	1	1

Figure 8.36 Consecutive Execution of 16-Bit Instructions (In Case of Branch to Address 8n)



Access		fetch	fetch	fetch	nop	fetch	nop	fetch	nop	fetch	nop
Number of cycles	In low-speed mode	2	2	2	1	2	1	2	1	2	1
	In high-speed mode	2, 3	1	1	1	1	1	1	1	1	1

Figure 8.37 Consecutive Execution of 16-Bit Instructions (In Case of Branch to Address 8n + 2)



Access		fetch	fetch	fetch	nop	fetch	nop	fetch	nop	fetch	nop
Number of cycles	In low-speed mode	2	2	2	1	2	1	2	1	2	1
	In high-speed mode	2, 3	2	1	1	1	1	1	1	1	1

Figure 8.38 Consecutive Execution of 16-Bit Instructions (In Case of Branch to Address 8n + 4)

Address	Pipeline state									
8n + 6	<IF	—	ID	EX						
8n + 8		IF	—	ID	EX					
8n + 10					ID	EX				
8n + 12			IF	—	—	ID	EX			
8n + 14							ID	EX		
8n + 16					IF	—	—	ID	EX	
8n + 18									ID	EX

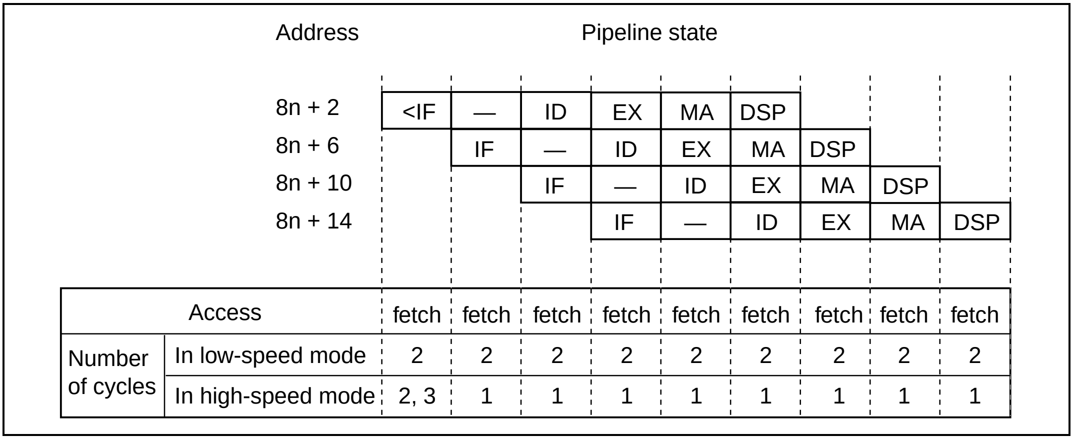
Access		fetch	fetch	fetch	nop	fetch	nop	fetch	nop	fetch	nop
Number of cycles	In low-speed mode	2	2	2	1	2	1	2	1	2	1
	In high-speed mode	2, 3	2	1	1	1	1	1	1	1	1

Figure 8.39 Consecutive Execution of 16-Bit Instructions (In Case of Branch to Address 8n + 6)

Address	Pipeline state							
8n	<IF	ID	EX	MA	DSP			
8n + 4		IF	ID	EX	MA	DSP		
8n + 8			IF	ID	EX	MA	DSP	
8n + 12				IF	ID	EX	MA	DSP

Access		fetch	fetch	fetch	fetch	fetch	fetch	fetch	fetch
Number of cycles	In low-speed mode	2	2	2	2	2	2	2	2
	In high-speed mode	2, 3	1	1	1	1	1	1	1

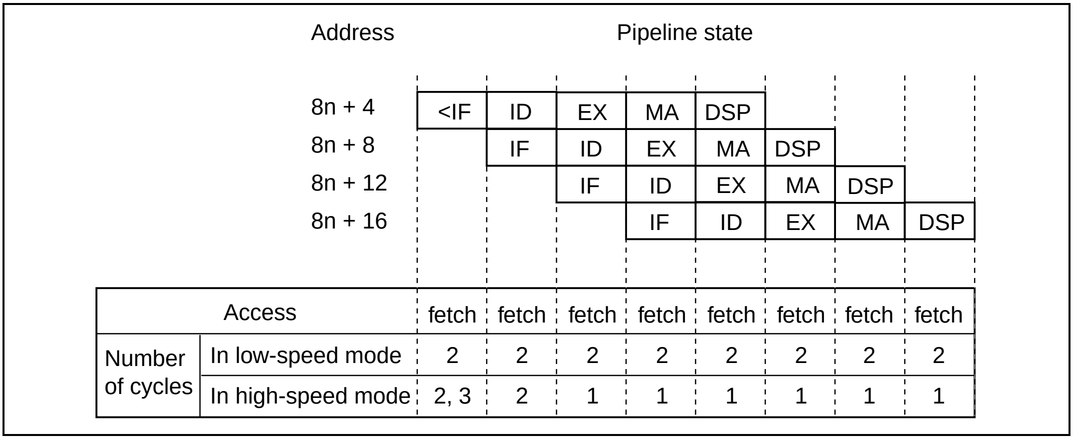
Figure 8.40 Consecutive Execution of 32-Bit Instructions (In Case of Branch to Address 8n)



Address	Pipeline state								
8n + 2	<IF	—	ID	EX	MA	DSP			
8n + 6		IF	—	ID	EX	MA	DSP		
8n + 10			IF	—	ID	EX	MA	DSP	
8n + 14				IF	—	ID	EX	MA	DSP

Access		fetch	fetch	fetch	fetch	fetch	fetch	fetch	fetch	fetch
Number of cycles	In low-speed mode	2	2	2	2	2	2	2	2	2
	In high-speed mode	2, 3	1	1	1	1	1	1	1	1

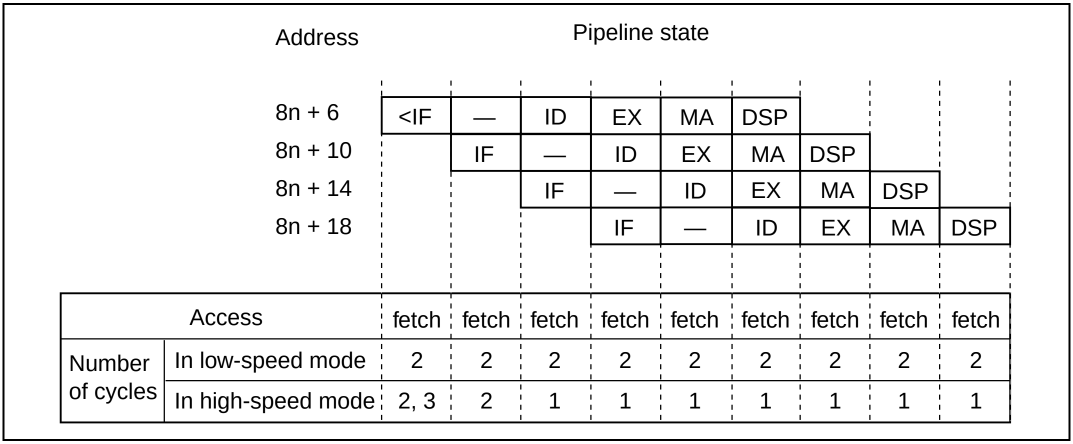
Figure 8.41 Consecutive Execution of 32-Bit Instructions (In Case of Branch to Address 8n + 2)



Address	Pipeline state							
8n + 4	<IF	ID	EX	MA	DSP			
8n + 8		IF	ID	EX	MA	DSP		
8n + 12			IF	ID	EX	MA	DSP	
8n + 16				IF	ID	EX	MA	DSP

Access		fetch	fetch	fetch	fetch	fetch	fetch	fetch	fetch
Number of cycles	In low-speed mode	2	2	2	2	2	2	2	2
	In high-speed mode	2, 3	2	1	1	1	1	1	1

Figure 8.42 Consecutive Execution of 32-Bit Instructions (In Case of Branch to Address 8n + 4)



Address	Pipeline state								
8n + 6	<IF	—	ID	EX	MA	DSP			
8n + 10		IF	—	ID	EX	MA	DSP		
8n + 14			IF	—	ID	EX	MA	DSP	
8n + 18				IF	—	ID	EX	MA	DSP

Access		fetch	fetch	fetch	fetch	fetch	fetch	fetch	fetch	fetch
Number of cycles	In low-speed mode	2	2	2	2	2	2	2	2	2
	In high-speed mode	2, 3	2	1	1	1	1	1	1	1

Figure 8.43 Consecutive Execution of 32-Bit Instructions (In Case of Branch to Address 8n + 6)

8.5 Usage Notes

1. Even if a CAS assertion width of two cycles is set with the TCAS bit in DRAM control register 2 (DCR2), the CAS assertion width will be one cycle in the second and subsequent accesses when the access size exceeds the bus width (for example, accesses to addresses 4n+1/4n+2/4n+3 in the case of longword access to 8-bit-bus-width DRAM).

2. The following restrictions apply when using DRAM/EDO DRAM in RAS down mode.
   - RAS down mode is not supported when Mϕ (the clock obtained after frequency division of the master clock (CKM)) is slower than CKE (the external bus clock).
   - In the event of a row address miss, the CS signal for the next space to be accessed is asserted for one cycle before external bus cycle generation.
   - If the row address value in a CS4 space access is different from the previously accessed CS5 space row address value, RAS1 is negated.
   - In DMAC dual address mode, when the transfer source is CS4/5 space and the transfer destination is CS space or on-chip register space, RAS1 is negated if the bit value corresponding to the transfer destination row address is different from the transfer source row address value.
   - When the DMAC is activated in dual address mode immediately after a CS4/5 space access by the CPU, and the transfer source is a different CS space or on-chip register space, RAS1 is negated if the bit value corresponding to the transfer source row address is different from the row address value in the preceding CS4/5 space access by the CPU. This negation occurs only in the case of a transfer immediately after DMAC activation. It does not occur in the second and subsequent transfers when the DMAC is in burst mode.
   - When the DMAC is activated with a different CS space access in single address mode immediately after a CS4/5 space access by the CPU, RAS1 is negated if the bit value corresponding to the different CS space row address is different from the row address value in the preceding CS4/5 space access by the CPU. This negation occurs only in the case of a transfer immediately after DMAC activation. It does not occur in the second and subsequent transfers when the DMAC is in burst mode.

3. If a TAS instruction for the on-chip RAM space is executed while the bus is released, $\overline{\text{BACK}}$ is first negated, then asserted again after execution is completed.

Section 9 Direct Memory Access Controller (DMAC)

9.1 Overview

The SH7065 includes an on-chip four-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 $\overline{\text{DACK}}$ (transfer request acknowledge signal), external memories, memory-mapped external devices, and on-chip peripheral modules (except the DMAC, BSC, and UBC). Using the DMAC reduces the burden on the CPU and increases the operating efficiency of the entire chip.

Certain usage notes apply to this DMAC: see section 9.6, DMAC Restrictions.

9.1.1 Features

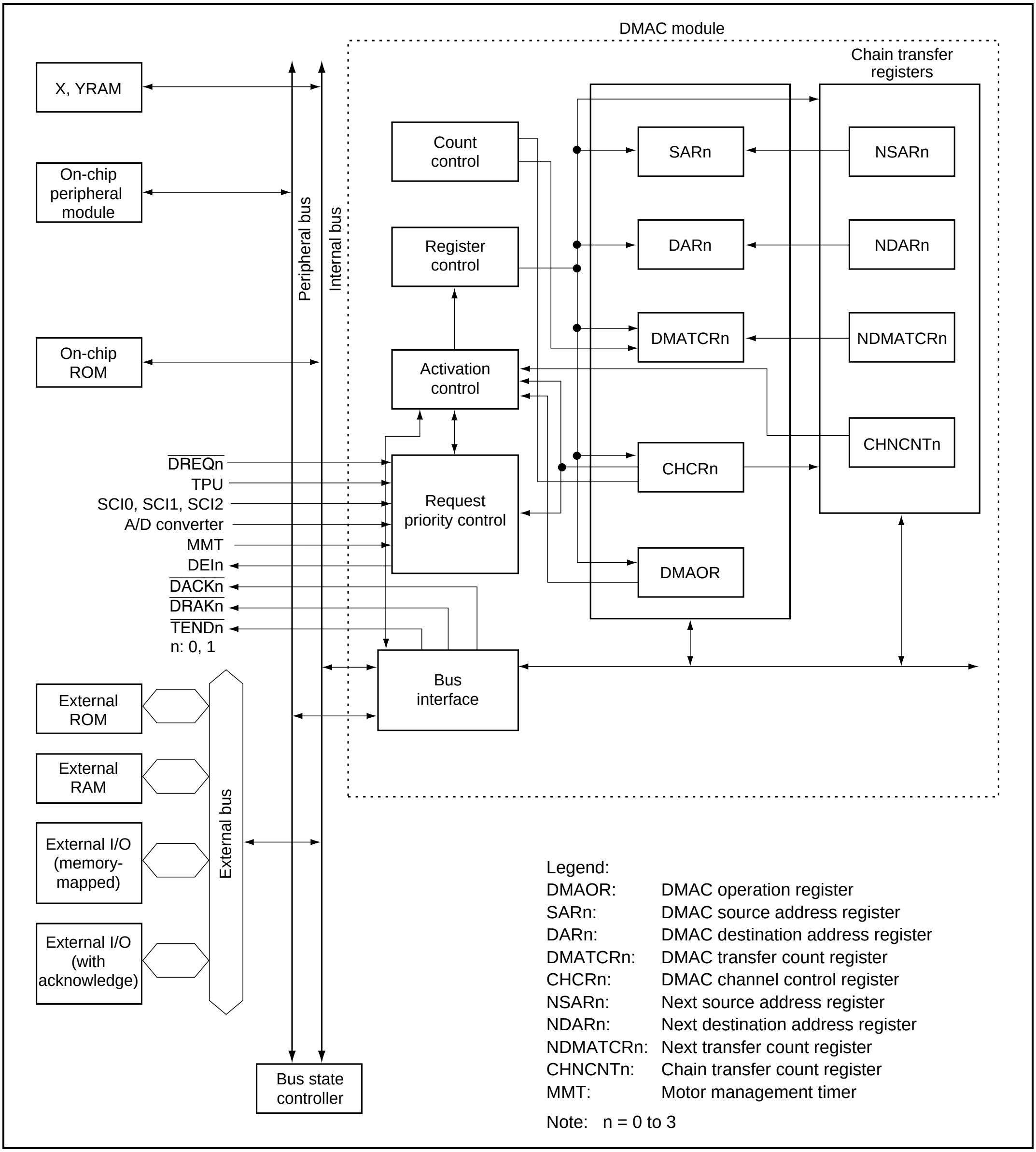
The DMAC has the following features.

- Four channels
- Four Gbytes of address space in the architecture
- Choice of 8-bit, 16-bit, or 32-bit transfer data length
- Maximum of 4G (4,294,967,296) transfers
- Choice of single or dual address mode
  - — Single address mode: Either the transfer source or the transfer destination (peripheral device) is accessed by a $\overline{\text{DACK}}$ signal while the other is accessed by address. One data transfer is completed in one bus cycle.
  - — Dual address mode: Both the transfer source and transfer destination are accessed by address. Values set in DMAC internal registers indicate the accessed address for both the transfer source and the transfer destination. Two bus cycles are required for one data transfer.
- Channel functions: The transfer mode can be set independently for each channel.
- Transfer requests: The following DMAC transfer activation requests are supported.
  - — External request: From two $\overline{\text{DREQ}}$ pins. Either low level detection or falling edge detection can be specified. When low level detection is selected, the sampled $\overline{\text{DREQ}}$ signal is stored in a FIFO. Either a 1-stage or 16-stage FIFO can be selected.
  - — Internal requests: Transfer requests from on-chip modules such as the TPU and SCI.
- Choice of bus mode: cycle steal mode or burst mode

- Two types of DMAC channel priority ranking:
  - Fixed priority mode: Channel priorities are permanently fixed.
  - Round robin mode: Sets the lowest priority for the channel that last received an execution request.
- An interrupt request can be sent to the CPU on completion of the specified number of transfers.
- Chain transfer allows a specified block of data to be transferred consecutively without CPU processing after the end of the current data transfer.
- A transfer end signal ($\overline{\text{TEND}}$) can be output for each channel at the end of DMA transfer.

9.1.2 Block Diagram

Figure 9.1 shows a block diagram of the DMAC.



Legend:

DMAOR:	DMAC operation register
SARn:	DMAC source address register
DARn:	DMAC destination address register
DMATCRn:	DMAC transfer count register
CHCRn:	DMAC channel control register
NSARn:	Next source address register
NDARn:	Next destination address register
NDMATCRn:	Next transfer count register
CHNCNTn:	Chain transfer count register
MMT:	Motor management timer

Note: n = 0 to 3

Figure 9.1 Block Diagram of DMAC

9.1.3 Pin Configuration

Table 9.1 shows the pins provided for each DMAC channel.

Table 9.1 DMAC Pins

Pin Name	Abbreviation	I/O	Function
DMA transfer request	$\overline{\text{DREQn}}$	Input	DMA transfer request input from external device to channel 0 or 1
DMA transfer request acceptance	$\overline{\text{DRAKn}}$	Output	Output of sampling acceptance signal for DMA transfer request input to channel 0 or 1 from external device
DMA transfer strobe	$\overline{\text{DACKn}}$	Output	Strobe output to external I/O in case of DMA transfer request from external device to channel 0 or 1
DMA transfer end	$\overline{\text{TENDn}}$	Output	Output at end of DMA transfer on relevant channel 0 or 1

9.1.4 Register Configuration

Table 9.2 summarizes the DMAC registers. The DMAC has a total of 33 registers. Eight registers are allocated to each channel, and an additional control register is shared by all four channels.

Table 9.2 DMAC Registers

Abbreviation	Name	Chan-nel	Read/ Write	Initial Value	Address	Register Size	Access Size
SAR0	DMA source address register 0	0	R/W	Undefined	H'FFFF1100	32 bits	16, 32
DAR0	DMA destination address register 0	0	R/W	Undefined	H'FFFF1104	32 bits	16, 32
DMATCR0	DMA transfer count register 0	0	R/W	Undefined	H'FFFF1108	32 bits	16, 32
CHCR0	DMA channel control register 0	0	R/W*1	00000000	H'FFFF110C	32 bits	16, 32
NSAR0	Next source address register 0	0	R/W	Undefined	H'FFFF1110	32 bits	16, 32
NDAR0	Next destination address register 0	0	R/W	Undefined	H'FFFF1114	32 bits	16, 32
NDMATCR0	Next transfer count register 0	0	R/W	Undefined	H'FFFF1118	32 bits	16, 32
CHNCNT0	Chain transfer count register 0	0	R/W	Undefined	H'FFFF111C	32 bits	16, 32
SAR1	DMA source address register 1	1	R/W	Undefined	H'FFFF1120	32 bits	16, 32
DAR1	DMA destination address register 1	1	R/W	Undefined	H'FFFF1124	32 bits	16, 32
DMATCR1	DMA transfer count register 1	1	R/W	Undefined	H'FFFF1128	32 bits	16, 32
CHCR1	DMA channel control register 1	1	R/W*1	00000000	H'FFFF112C	32 bits	16, 32
NSAR1	Next source address register 1	1	R/W	Undefined	H'FFFF1130	32 bits	16, 32
NDAR1	Next destination address register 1	1	R/W	Undefined	H'FFFF1134	32 bits	16, 32
NDMATCR1	Next transfer count register 1	1	R/W	Undefined	H'FFFF1138	32 bits	16, 32
CHNCNT1	Chain transfer count register 1	1	R/W	Undefined	H'FFFF113C	32 bits	16, 32
SAR2	DMA source address register 2	2	R/W	Undefined	H'FFFF1140	32 bits	16, 32

Abbreviation	Name	Chan-nel	Read/ Write	Initial Value	Address	Register Size	Access Size*[2]
DAR2	DMA destination address register 2	2	R/W	Undefined	H'FFFF1144	32 bits	16, 32
DMATCR2	DMA transfer count register 2	2	R/W	Undefined	H'FFFF1148	32 bits	16, 32
CHCR2	DMA channel control register 2	2	R/W*[1]	00000000	H'FFFF114C	32 bits	16, 32
NSAR2	Next source address register 2	2	R/W	Undefined	H'FFFF1150	32 bits	16, 32
NDAR2	Next destination address register 2	2	R/W	Undefined	H'FFFF1154	32 bits	16, 32
NDMATCR2	Next transfer count register 2	2	R/W	Undefined	H'FFFF1158	32 bits	16, 32
CHNCNT2	Chain transfer count register 2	2	R/W	Undefined	H'FFFF115C	32 bits	16, 32
SAR3	DMA source address register 3	3	R/W	Undefined	H'FFFF1160	32 bits	16, 32
DAR3	DMA destination address register 3	3	R/W	Undefined	H'FFFF1164	32 bits	16, 32
DMATCR3	DMA transfer count register 3	3	R/W	Undefined	H'FFFF1168	32 bits	16, 32
CHCR3	DMA channel control register 3	3	R/W*[1]	00000000	H'FFFF116C	32 bits	16, 32
NSAR3	Next source address register 3	3	R/W	Undefined	H'FFFF1170	32 bits	16, 32
NDAR3	Next destination address register 3	3	R/W	Undefined	H'FFFF1174	32 bits	16, 32
NDMATCR3	Next transfer count register 3	3	R/W	Undefined	H'FFFF1178	32 bits	16, 32
CHNCNT3	Chain transfer count register 3	3	R/W	Undefined	H'FFFF117C	32 bits	16, 32
DMAOR	DMA operation register	All	R/W*[1]	0000	H'FFFF10F0	16 bits	16

Notes: 1. Bit 1 of CHCR0 to CHCR3 and bits 1 and 2 of DMAOR can only be written with 0 after being read as 1, to clear the flags.

2. When 16-bit access is used on SAR0 to SAR3, DAR0 to DAR3, or CHCR0 to CHCR3, the 16 bits that are not accessed retain their value.

9.2 Register Descriptions

9.2.1 DMA Source Address Registers 0 to 3 (SAR0 to SAR3)

Bit:	31	30	29	28	27	26	25	24	23		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

DMA source address registers 0 to 3 (SAR0 to SAR3) are 32-bit readable/writable registers that specify the source address of a DMA transfer. These registers have a count function, and during a DMA transfer they indicate the next source address. In single address mode, the SAR value is ignored when a device with $\overline{\text{DACK}}$ has been specified as the transfer source.

Specify a 16-bit boundary address in a 16-bit transfer, and a 32-bit boundary address in a 32-bit transfer. Operation cannot be guaranteed if a different address is set.

The value of these registers is undefined after a power-on reset, and in hardware standby mode and software standby mode.

9.2.2 DMA Destination Address Registers 0 to 3 (DAR0 to DAR3)

Bit:	31	30	29	28	27	26	25	24	23		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

DMA destination address registers 0 to 3 (DAR0 to DAR3) are 32-bit readable/writable registers that specify the destination address of a DMA transfer. These registers have a count function, and during a DMA transfer they indicate the next destination address. In single address mode, the DAR value is ignored when a device with $\overline{\text{DACK}}$ has been specified as the transfer destination.

Specify a 16-bit boundary address in a 16-bit transfer, and a 32-bit boundary address in a 32-bit transfer. Operation cannot be guaranteed if a different address is set.

The value of these registers is undefined after a power-on reset, and in hardware standby mode and software standby mode.

9.2.3 DMA Transfer Count Registers 0 to 3 (DMATCR0 to DMATCR3)

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

DMA transfer count registers 0 to 3 (DMATCR0 to DMATCR3) are 32-bit readable/writable registers that specify the transfer count for the channel (number of bytes, words, or longwords). Setting H'00000001 gives a transfer count of 1, while H'00000000 gives the maximum setting of 4,294,967,296 (4G) transfers. During DMAC operation, the remaining number of transfers is shown.

The value of these registers is undefined after a power-on reset, and in hardware standby mode and software standby mode.

9.2.4 DMA Channel Control Registers 0 to 3 (CHCR0 to CHCR3)

Bit:	31	30	29	28	27	26	25	24
	—	—	—	RS4	RS3	RS2	RS1	RS0
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:	23	22	21	20	19	18	17	16
	—	FIFOS	—	—	NDARE	NSARE	FCS	TES
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R/W	R	R	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8
	DM1	DM0	SM1	SM0	CHNE	RL	AM	AL
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
	TEND	DS	TM	TS1	TS0	IE	TE*	DE
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: * The TE bit can only be cleared by writing 0 after it is read as 1.

DMA channel control registers 0 to 3 (CHCR0 to CHCR3) are 32-bit readable/writable registers that specify the operating mode, transfer method, etc., for each channel.

All bits in these registers are initialized to 0 after a power-on reset, and in hardware standby mode and software standby mode.

Bits 31 to 29—Reserved: These bits are always read as 0 and cannot be modified.

Bits 28 to 24—Resource Select 4 to 0 (RS4 to RS0): These bits specify the transfer request source.

Bit 28: RS4	Bit 27: RS3	Bit 26: RS2	Bit 25: RS1	Bit 24: RS0	Description
0	0	0	0	0	External request, dual address mode (Initial value)
				1	(Reserved)
			1	0	External request, single address mode External address space → external device
				1	External request, single address mode External device → external address space
		1	0	0	Auto-request
				1	(Reserved)
			1	0	(Reserved)
				1	(Reserved)
	1	0	0	0	TPU TGI0A
				1	TGI1A
			1	0	TGI2A
				1	TGI3A
		1	0	0	TGI4A
				1	TGI5A
			1	0	A/D ADI 0
				1	ADI 1
1	0	0	0	0	SCI0 TXI0
				1	RXI0
			1	0	SCI1 TXI1
				1	RXI1
		1	0	0	SCI2 TXI2
				1	RXI2
			1	0	(Reserved)
				1	(Reserved)
	1	0	0	0	MMT TGM
				1	MMT TGN
			1	0	(Reserved)
				1	(Reserved)
		1	0	0	(Reserved)
				1	(Reserved)
			1	0	(Reserved)
				1	(Reserved)

Bit 23—Reserved: This bit is always read as 0 and cannot be modified.

Bit 22—FIFO Select (FIFOS): Selects the FIFO to be used for $\overline{\text{DREQ}}$ level detection. This bit is invalid when $\overline{\text{DREQ}}$ falling edge detection is used.

Bit 22: FIFOS	Description
0	1-stage FIFO is used for $\overline{\text{DREQ}}$ level detection (Initial value)
1	16-stage FIFO is used for $\overline{\text{DREQ}}$ level detection

Bits 21 and 20—Reserved: These bits are always read as 0 and cannot be modified.

Bit 19—Next Destination Address Register Enable (NDARE): Selects whether or not the next destination address register value is to be transferred to the destination address register to update the destination address during chain transfer.

Bit 19: NDARE	Description
0	In chain transfer, next destination address register value is not copied to destination address register (Initial value)
1	In chain transfer, next destination address register value is copied to destination address register

Bit 18—Next Source Address Register Enable (NSARE): Selects whether or not the next source address register value is to be transferred to the source address register to update the source address during chain transfer.

Bit 18: NSARE	Description
0	In chain transfer, next source address register value is not copied to source address register (Initial value)
1	In chain transfer, next source address register value is copied to source address register

Bit 17—Flag Clear Timing Select (FCS): When a transfer request by an on-chip module is accepted, the DMAC outputs a signal to clear the transfer request flag of the on-chip module that made the transfer request. This bit selects whether this output is to be performed in the bus cycle in which the transfer count register (DMATCRn) value becomes 0, or in every bus cycle. When this bit is set to 1, the edge detection setting should be made in bit 6 (DREQ Select: DS).

Bit 17: FCS	Description
0	When an on-chip module is the transfer request source, the DMAC outputs the flag clearing signal in the bus cycle in which the transfer count register (DMATCRn) value becomes 0 (Initial value)
1	When an on-chip module is the transfer request source, the DMAC outputs the flag clearing signal in every last bus cycle

Note: When DREQ is edge-detected, FCS can be used to select the edge clearing timing.

Bit 16—Transfer End Setting Select (TES): Specifies whether the transfer end bit (TE) is to be set at the end of all the chain transfers specified in the chain count register (CHNCNT), or at the end of the number of data transfers specified by DMATCRn.

This bit is valid regardless of the setting of bit 11 (Chain Transfer Enable: CHNE). Therefore, when not performing chain transfer, either set this bit to 1 or else set a value of 0 in the CHNCNT. When bit 2 (Interrupt Enable: IE) is set to 1, a transfer end interrupt (DEI) is requested when the transfer end bit is set at the timing specified by this bit.

Bit 16: TES	Description
0	Transfer end bit (TE) is set to 1 when CHNCNTn = 0 and DMATCRn = 0 (Initial value)
1	Transfer end bit (TE) is set to 1 when DMATCRn = 0

Note: With auto-request, this bit is invalid and an interrupt is requested when DMATCRn = 0. When auto-request is selected, TES = 1 operation is used.

Bits 15 and 14—Destination Address Modes 1 and 0 (DM1, DM0): These bits specify incrementing/decrementing of the DMA transfer destination address. The specification of these bits is ignored when data is transferred from address space to an external device in single address mode.

Bit 15: DM1	Bit 14: DM0	Description
0	0	Destination address fixed (Initial value)
	1	Destination address incremented (+1 in 8-bit transfer, +2 in 16-bit transfer, +4 in 32-bit transfer)
1	0	Destination address decremented (–1 in 8-bit transfer, –2 in 16-bit transfer, –4 in 32-bit transfer)
	1	(Use prohibited)

Bits 13 and 12—Source Address Modes 1 and 0 (SM1, SM0): 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 address space in single address mode.

Bit 13: SM1	Bit 12: SM0	Description
0	0	Source address fixed (Initial value)
	1	Source address incremented (+1 in 8-bit transfer, +2 in 16-bit transfer, +4 in 32-bit transfer)
1	0	Source address decremented (–1 in 8-bit transfer, –2 in 16-bit transfer, –4 in 32-bit transfer)
	1	(Use prohibited)

Bit 11—Chain Transfer Enable (CHNE): Selects whether or not chain transfer is performed in DMAC transfer.

Bit 11: CHNE	Description
0	Chain transfer is not performed (Initial value)
1	Chain transfer is performed

Bit 10—Request Check Level (RL): Selects whether the $\overline{\text{DRAK}}$ signal (that notifies an external device of the acceptance of $\overline{\text{DREQ}}$) is an active-high or active-low output. Note that the initial value of this bit is the active-high setting.

Bit 10: RL	Description
0	$\overline{\text{DRAK}}$ is an active-high output (Initial value)
1	$\overline{\text{DRAK}}$ is an active-low output

Bit 9—Acknowledge Mode (AM): In dual address mode, selects whether $\overline{\text{DACK}}$ is output in the data read cycle or write cycle. In single address mode, $\overline{\text{DACK}}$ is always output regardless of the setting of this bit.

Bit 9: AM	Description
0	$\overline{\text{DACK}}$ is output in read cycle (Initial value)
1	$\overline{\text{DACK}}$ is output in write cycle

Bit 8—Acknowledge Level (AL): Specifies the $\overline{\text{DACK}}$ (acknowledge) signal as active-high or active-low. Note that the initial value of this bit is the active-high setting.

Bit 8: AL	Description	
0	Active-high output	(Initial value)
1	Active-low output	

Bit 7—TEND Select (TEND): Selects whether or not the $\overline{\text{TEND}}$ signal (notifying an external device that transfer has ended) is to be output at the end of DMA transfer. When output is selected, $\overline{\text{TEND}}$ is output in synchronization with $\overline{\text{DACK}}$ assertion at the end of transfer.

Bit 7: TEND	Description	
0	$\overline{\text{TEND}}$ is not output at end of DMA transfer	(Initial value)
1	$\overline{\text{TEND}}$ is output at end of DMA transfer	

Bit 6—DREQ Select (DS): Specifies either low level detection or falling edge detection as the sampling method for the $\overline{\text{DREQ}}$ pin and on-chip peripheral module transfer requests used in external request mode.

If auto-request is specified, the specification of this bit is ignored. Edge detection is not used with auto-request.

Bit 6: DS	Description	
0	Low level detection	(Initial value)
1	Falling edge detection	

Bit 5—Transmit Mode (TM): Specifies the bus mode for transfer.

Bit 5: TM	Description	
0	Cycle steal mode	(Initial value)
1	Burst mode	

Bits 4 and 3—Transmit Size 1 and 0 (TS1, TS0): These bits specify the transfer data size.

Bit 4: TS1	Bit 3: TS0	Description
0	0	Byte size (8 bits) (Initial value)
	1	Word size (16 bits)
1	0	Longword size (32 bits)
	1	(Use prohibited)

Bit 2—Interrupt Enable (IE): When this bit is set to 1, an interrupt request is generated after the number of data transfers specified in DMATCR, or after all chain transfers are completed.

Bit 2: IE	Description
0	Interrupt request not generated after number of transfers specified in DMATCR (Initial value)
1	Interrupt request generated after number of transfers specified in DMATCR

Bit 1—Transfer End (TE): This bit is set to 1 on completion of the number of transfers specified in DMATCR, or on completion of all the chain transfers specified in CHNCNT. The timing of TE bit setting is specified by bit 16 (TES). If the IE bit is set to 1 at this time, an interrupt request is generated.

If data transfer ends before TE is set to 1 (for example, due to an NMI interrupt, address error, or clearing of the DE bit or the DME bit in DMAOR), the TE bit is not set to 1. When this bit is 1, data transfer is not enabled even if the DE bit is set to 1.

Bit 1: TE	Description
0	Number of transfers specified in DMATCR not completed (Initial value) [Clearing conditions] • When 0 is written to TE after reading TE = 1 • In a power-on reset, and in standby mode
1	Number of transfers specified in DMATCR completed, or all chain transfers specified in CHNCNT completed

Note: Not initialized in module standby mode.

Bit 0—DMAC Enable (DE): Enables operation of the corresponding channel.

Bit 0: DE	Description	
0	Operation of corresponding channel is disabled	(Initial value)
1	Operation of corresponding channel is enabled	

When auto-request is specified (with RS5 to RS0), transfer is begun when this bit is set to 1. In the case of an external request or on-chip module request, transfer is begun when a transfer request is issued after this bit is set to 1. Transfer can be suspended midway by clearing this bit to 0.

Even if the DE bit has been set, transfer is not enabled when TE is 1, when DME in DMAOR is 0, or when the NMIF or AE bit in DMAOR is 1.

9.2.5 Next Source Address Registers 0 to 3 (NSAR0 to NSAR3)

Bit:	31	30	29	28	27	26	25	24	23		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

Next source address registers 0 to 3 (NSAR0 to NSAR3) are 32-bit readable/writable registers that specify the source address for the next transfer when chain transfer is set. In single address mode, the NSAR value is ignored when a device with $\overline{\text{DACK}}$ has been specified as the transfer destination.

Specify a 16-bit boundary address in a 16-bit transfer, and a 32-bit boundary address in a 32-bit transfer. Operation cannot be guaranteed if a different address is set.

The value of these registers is undefined after a power-on reset, and in hardware standby mode and software standby mode.

9.2.6 Next Destination Address Registers 0 to 3 (NDAR0 to NDAR3)

Bit:	31	30	29	28	27	26	25	24	23		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

Next destination address registers 0 to 3 (NDAR0 to NDAR3) are 32-bit readable/writable registers that specify the destination address for the next transfer when chain transfer is set. In single address mode, the NDAR value is ignored when a device with $\overline{\text{DACK}}$ has been specified as the transfer destination.

Specify a 16-bit boundary address in a 16-bit transfer, and a 32-bit boundary address in a 32-bit transfer. Operation cannot be guaranteed if a different address is set.

The value of these registers is undefined after a power-on reset, and in hardware standby mode and software standby mode.

9.2.7 Next Transfer Count Registers 0 to 3 (NDMATCR0 to NDMATCR3)

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

Next transfer count registers 0 to 3 (NDMATCR0 to NDMATCR3) are 32-bit readable/writable registers that specify the transfer count for the next transfer on the channel (number of bytes, words, or longwords) in chain transfer.

The value of these registers is undefined after a power-on reset, and in hardware standby mode and software standby mode.

9.2.8 Chain Transfer Count Registers 0 to 3 (CHNCNT0 to CHNCNT3)

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

Chain transfer count registers 0 to 3 (CHNCNT0 to CHNCNT3) are 32-bit readable/writable registers that specify the chain transfer count when chain transfer is set.

The value of these registers is undefined after a power-on reset, and in hardware standby mode and software standby mode. If chain transfer is not to be enabled, either initialize these registers to 0 or set bit 16 (TES) in CHCRn to 1 before enabling transfer.

9.2.9 DMA Operation Register (DMAOR)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	RC3	RC2	RC1	RC0	—	—	—	—	—	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/W	R/W	R	R	R	R	R	R/(W)	R/(W)	R/W

Note: The AE and NMIF bits can only be cleared to 0 after being read as 1.

The DMA operation register (DMAOR) is a 16-bit readable/writable register that specifies the DMAC transfer mode.

DMAOR bits are initialized to 0 after a power-on reset, and in hardware standby mode and software standby mode.

Bits 15 to 12—Reserved: These bits are always read as 0 and cannot be modified.

Bits 11 to 8—Round Robin Channel Select 3 to 0 (RC3 to RC0): When there are simultaneous transfer requests for a number of channels, these bits determine the channel priority order for executing the transfers. Bits RC3 to RC0 correspond to channels CH3 to CH0. When a bit is set to 1, the priority of the corresponding channel is determined according to the round robin method.

Bits 11 to 8: **RCn**	**Description**
0	The priority order of corresponding channel CHn (n = 0 to 3) is fixed. When all RC bits are 0, the channel priority order is CH0 > CH1 > CH2 > CH3. (Initial value)
1	The priority order of corresponding channel CHn (n = 0 to 3) is determined according to the round robin method.

Note: When the round robin method is set for the priority order, at least two RC bits should be set to 1. If only one RC bit is set to 1, the inter-channel priority order will be CH0 > CH1 > CH2 > CH3.

When the priority order of two or more channels is determined by the round robin method, channels with consecutive channel numbers must be set (e.g. CH2 and CH3, or CH1, CH2, and CH3). Operation cannot be guaranteed if channels with non-consecutive channel numbers (such as CH0 and CH2) are designated as having their priority order determined by the round robin method.

If round robin priority is specified for CH1, CH2, and CH3, the channel priority relationship with the other channel will be as follows:

$$\text{CH0} > \underbrace{\text{CH1, CH2, CH3}}_{\text{Round Robin}}.$$

Bits 7 to 3—Reserved: These bits are always read as 0 and cannot be modified.

Bit 2—Address Error Flag (AE): Flag that indicates the occurrence of an address error during DMA transfer. If this bit is set during data transfer, transfers on all channels are suspended. The CPU cannot write a 1 to AE. This bit can only be cleared by writing 0 after reading 1.

Bit 2: AE	**Description**
0	No address error, DMA transfer enabled (Initial value) [Clearing condition] When 0 is written to AE after reading AE = 1
1	Address error, DMA transfer disabled [Setting condition] When an address error is caused by the DMAC

Bit 1—NMI Flag (NMIF): Flag that indicates NMI input. Setting of this bit can be performed regardless of whether the DMAC is operating or halted. If this bit is set during data transfer, transfers on all channels are suspended. The CPU cannot write a 1 to NMIF. This bit can only be cleared by writing 0 after reading 1.

Bit 1: NMIF	Description	
0	No NMI input, DMA transfer enabled [Clearing condition] When 0 is written to NMIF after reading NMIF = 1	(Initial value)
1	NMI input, DMA transfer disabled [Setting condition] When an NMI interrupt is generated	

Bit 0—DMAC Master Enable (DME): Enables activation of the entire DMAC. When the DME bit and the DE bit of the CHCR register for the corresponding channel are set to 1, that channel is enabled for transfer. If this bit is cleared during data transfer, transfers on all channels are suspended.

Even if the DME bit has been set, 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.

Bit 0: DME	Description	
0	Operation disabled on all channels	(Initial value)
1	Operation enabled on all channels	

9.3 Operation

When there is a DMA transfer request, 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.

9.3.1 DMA Transfer Procedure

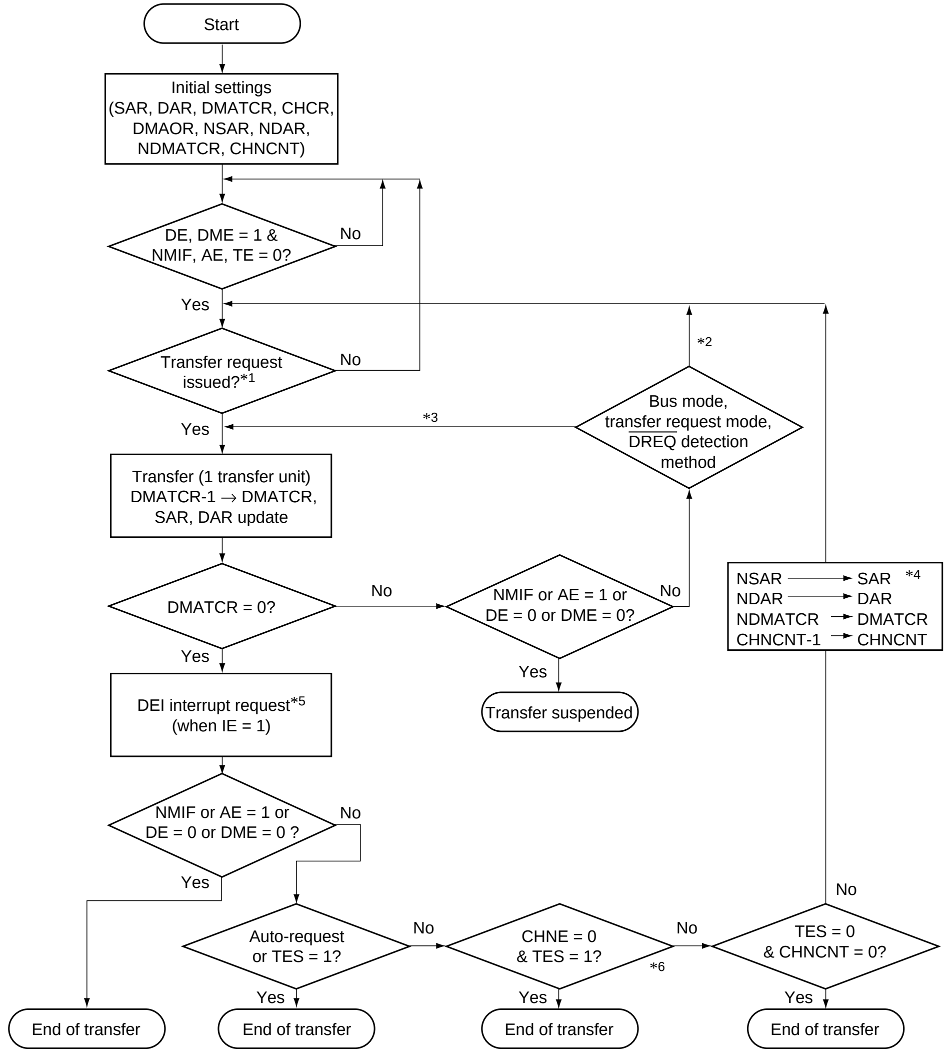
After the desired transfer conditions have been set in the DMA source address register (SAR), DMA destination address register (DAR), DMA transfer count register (DMATCR), DMA channel control register (CHCR), DMA operation register (DMAOR), next source address register (NSAR), next destination address register (NDAR), next transfer count register (NDMATCR), and chain transfer count register (CHNCNT), the DMAC executes data transfer 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 while transfer is enabled, the DMAC transfers one transfer unit of data (determined by the setting of TS0 and TS1). In auto-request mode, the transfer begins automatically when the DE bit and DME bit 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, a DEI interrupt request is sent to the CPU.*
4. When a DMAC address error or NMI interrupt occurs, the transfer is suspended. Transfer is also suspended when the DE bit in CHCR or the DME bit in DMAOR is cleared to 0.
5. In the case of auto-request, or when CHNE = 0 and TES = 1, transfer ends when DMATCRn = 0.

   When the chain transfer enable bit (CHNE) is set to 1, the values in the next source address register (NSAR), next destination address register (NDAR), and next transfer count register (NDMATCR) are copied, respectively, to the DMA source address register (SAR), DMA destination address register (DAR), and DMA transfer count register (DMATCR), and chain transfer is started. Chain transfer ends when the value in the chain transfer count register (CHNCNT) reaches 0.

Note: * If the TES bit in CHCRn is cleared to 0, a DEI interrupt is generated when the CHNCNTn and DMATCRn values both become 0.

Figure 9.2 shows a flowchart of this procedure.


Start
Initial settings (SAR, DAR, DMATCR, CHCR, DMAOR, NSAR, NDAR, NDMATCR, CHNCNT)
DE, DME = 1 & NMIF, AE, TE = 0?
No
Yes
Transfer request issued?*1
No
Yes
*2
*3
Bus mode, transfer request mode, DREQ detection method
Transfer (1 transfer unit) DMATCR-1 → DMATCR, SAR, DAR update
DMATCR = 0?
No
NMIF or AE = 1 or DE = 0 or DME = 0?
No
NSAR → SAR *4
NDAR → DAR
NDMATCR → DMATCR
CHNCNT-1 → CHNCNT
Yes
Yes
Transfer suspended
DEI interrupt request*5 (when IE = 1)
NMIF or AE = 1 or DE = 0 or DME = 0 ?
No
Yes
No
Auto-request or TES = 1?
No
CHNE = 0 & TES = 1?
No
*6
TES = 0 & CHNCNT = 0?
Yes
Yes
Yes
End of transfer
End of transfer
End of transfer
End of 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. DREQ level detection (external request) in burst mode, or cycle steal mode. When edge detection and edge clearing every cycle are set.
3. When transfer requests are edge-detected, and edge clearing is performed at the end of all transfers.
4. Whether or not NSAR → SAR and NDAR → DAR transfer is required can be set with the corresponding bits in CHCR.
5. When the TES bit in CHCR is cleared to 0, the condition for interrupt generation is {CHNCNTn = 0 and DMATCRn = 0}. When the TES bit in CHCR is set to 1, the condition for interrupt generation is {DMATCRn = 0}, and the value of CHNCNTn is immaterial.
6. When clearing CHNE to 0, either set TES to 1 or clear CHNCNT to 0.

Figure 9.2 DMAC Transfer Flowchart

9.3.2 DMA Transfer Requests

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 means of bits RS4 to RS0 in DMA channel control registers 0 to 3 (CHCR0 to CHCR3).

Auto Request Mode

When there is 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, the auto-request mode allows the DMAC to automatically generate a transfer request signal internally. When the RS bit in CHCR0 to CHCR3 is set to auto-request mode, the DE bit is set to 1, and the DME bit in the DMA operation register (DMAOR) is set to 1, the transfer begins (so long as the TE bit in CHCR0 to CHCR3 and the NMIF and AE bits in DMAOR are all 0).

External Request Mode

In this mode a transfer is performed in response to a transfer request signal ($\overline{\text{DREQ}}$) from an external device. One of the modes shown in table 9.3 should be chosen 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 bit in CHCR0 to CHCR3 is used to select either falling edge detection or low level detection for the $\overline{\text{DREQ}}$ signal (level detection when DS = 0, edge detection when DS = 1). When low level detection is used, the FIFO to be used can be selected with the FIFOS bit. When edge detection is used, the edge clearing timing can be selected with the FCS bit.

The source of the transfer request does not have to be the data transfer source or destination.

Table 9.3 Selecting External Request Mode with RS Bits

RS4	RS3	RS2	RS1	RS0	Address Mode	Transfer Source	Transfer Destination
0	0	0	0	0	Dual address mode	Any*	Any*
0	0	0	1	0	Single address mode	External memory or memory-mapped external device	External device with DACK
0	0	0	1	1	Single address mode	External device with DACK	External memory or memory-mapped external device

Note: * External memory, memory-mapped external device, on-chip memory, on-chip peripheral module (except DMAC, UBC, and BSC)

On-Chip Peripheral Module Request Mode

In this mode a transfer is performed in response to a transfer request signal (interrupt request signal) from one of the SH7065's on-chip peripheral modules. As shown in table 9.4, there are a total of 16 transfer request signals: six compare match interrupts or input capture interrupts from the timer pulse unit (TPU), receive-data-full interrupts (RXI) and transmit-data-empty interrupts (TXI) from the three serial communication interface (SCI) channels, A/D conversion end interrupts (ADI) from the two A/D converter channels, and two interrupts from the motor management timer (MMT). If DMA transfer is enabled (DE = 1, DME = 1, TE = 0, NMIF = 0, AE = 0), DMA transfer starts when a transfer request signal is input.

The source of the transfer request does not have to be the data transfer source or destination. However, when the transfer request is set to RXI (transfer request by SCI receive-data-full interrupt), the transfer source must be the SCI's receive FIFO data register (SCFRDR). When the transfer request is set to TXI (transfer request by SCI transmit-data-empty interrupt), the transfer destination must be the SCI's transmit FIFO data register (SCFTDR). When the transfer request is set to ADIn, the transfer source must be the A/D data register (ADDRn).

To output a transfer request from an on-chip peripheral module, set the interrupt enable bit for the module and output an interrupt signal.

When an on-chip peripheral module interrupt request signal is used as a DMA transfer request signal, an interrupt is not issued to the CPU.

The transfer request signals shown in table 9.4 are cleared automatically when the corresponding DMA transfer is performed. In cycle steal mode the signal is cleared for a single transfer; in burst mode, there is a choice of clearance each time a transfer is executed or on execution of the last transfer. The flag clear timing select bit (FCS) in the channel control register (CHCR) is used to select the transfer request signal clearing mode.

Table 9.4 Selecting On-Chip Peripheral Module Request Mode with RS Bits

RS4	RS3	RS2	RS1	RS0	DMAC Transfer Request Source	DMAC Transfer Request Signal	Transfer Source	Transfer Destination	Bus Mode
0	1	0	0	0	TPU	TGI0A interrupt	Any*	Any*	Burst/cycle steal mode
				1	TPU	TGI1A interrupt	Any*	Any*	Burst/cycle steal mode
			1	0	TPU	TGI2A interrupt	Any*	Any*	Burst/cycle steal mode
				1	TPU	TGI3A interrupt	Any*	Any*	Burst/cycle steal mode
		1	0	0	TPU	TGI4A interrupt	Any*	Any*	Burst/cycle steal mode
				1	TPU	TGI5A interrupt	Any*	Any*	Burst/cycle steal mode
			1	0	A/D converter	ADI0 (A/D conversion end interrupt)	ADDR0	Any*	Burst/cycle steal mode
				1	A/D converter	ADI1 (A/D conversion end interrupt)	ADDR1	Any*	Burst/cycle steal mode
1	0	0	0	0	SCI0 transmitter	TXI0 (SCI0 transmit-data-empty transfer request)	Any*	TDR0	Burst/cycle steal mode
				1	SCI0 receiver	RXI0 (SCI0 receive-data-full transfer request)	RDR0	Any*	Burst/cycle steal mode
			1	0	SCI1 transmitter	TXI1 (SCI1 transmit-data-empty transfer request)	Any*	TDR1	Burst/cycle steal mode
				1	SCI1 receiver	RXI1 (SCI1 receive-data-full transfer request)	RDR1	Any*	Burst/cycle steal mode
		1	0	0	SCI2 transmitter	TXI2 (SCI2 transmit-data-empty transfer request)	Any*	TDR2	Burst/cycle steal mode
				1	SCI2 receiver	RXI2 (SCI2 receive-data-full transfer request)	RDR2	Any*	Burst/cycle steal mode
	1	0		0	MMT	TGM	Any*	Any*	Burst/cycle steal mode
				1	MMT	TGN	Any*	Any*	Burst/cycle steal mode

Legend:

TPU:	Timer pulse unit
SCI0, SCI1, SCI2:	Serial communication interface channels 0 to 2

ADDR0, ADDR1:	A/D data registers for A/D converters 0 and 1
TDRn, RDRn:	SCFTDRn and SCFRDRn for SCI channel n (n = 0 to 2)
MMT:	Motor management timer

Note: * External memory, memory-mapped external device, on-chip memory, on-chip peripheral module (except DMAC, BSC, and UBC)

9.3.3 Channel Priorities

If the DMAC receives simultaneous transfer requests on two or more channels, it selects a channel according to a predetermined priority system, either in a fixed mode or round robin mode. The mode is selected with the round robin channel select bits (RC0 to RC3) in the DMA operation register (DMAOR). The fixed mode is selected for the channel priority order by clearing the round robin channel select bits for all channels to 0. When using round robin mode, the round robin bits for the channels to be used are set to 1. The channels specified in this case must have consecutive channel numbers.

Fixed Mode

In the fixed mode, the channel priority order does not change. In this mode, the following initial channel priority order is used.

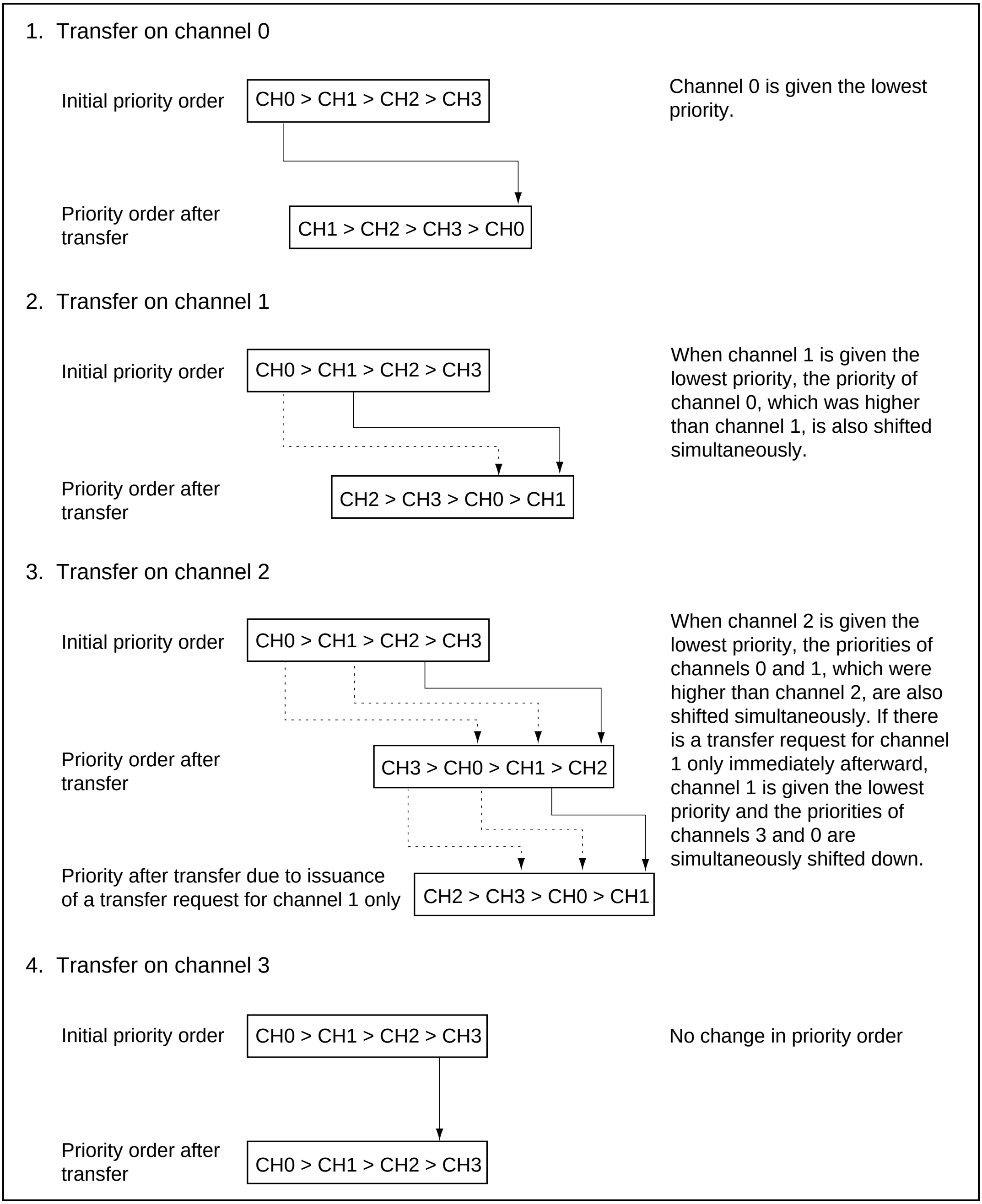
CH0 > CH1 > CH2 > CH3

To perform fixed mode transfer, the round robin channel select bits (RC0 to RC3) in the DMA operation register (DMAOR) must all be cleared to 0.

Round Robin Mode

In round robin mode, each time the transfer of one transfer unit (byte, word, or longword) ends on a given channel, that channel is assigned the lowest priority level. The channels specified by the round robin channel select bits (RC0 to RC3) in the DMA operation register (DMAOR) are subject to round robin control. Only channels with consecutive channel numbers can be specified; operation cannot be guaranteed if non-consecutive channels are specified for round robin priority control. If only one channel is specified, the channel priority order is the same as in the fixed mode.

Figure 9.3 illustrates round robin operation for CH0 to CH3. The order of priority in round robin mode immediately after a reset is the initial priority order: CH0 > CH1 > CH2 > CH3.



Channel 0 is given the lowest priority.

When channel 1 is given the lowest priority, the priority of channel 0, which was higher than channel 1, is also shifted simultaneously.

When channel 2 is given the lowest priority, the priorities of channels 0 and 1, which were higher than channel 2, are also shifted simultaneously. If there is a transfer request for channel 1 only immediately afterward, channel 1 is given the lowest priority and the priorities of channels 3 and 0 are simultaneously shifted down.

No change in priority order

Figure 9.3 Round Robin Mode

Figure 9.4 shows the changes in priority levels when transfer requests are issued simultaneously for channels 0 and 3, and channel 1 receives 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.

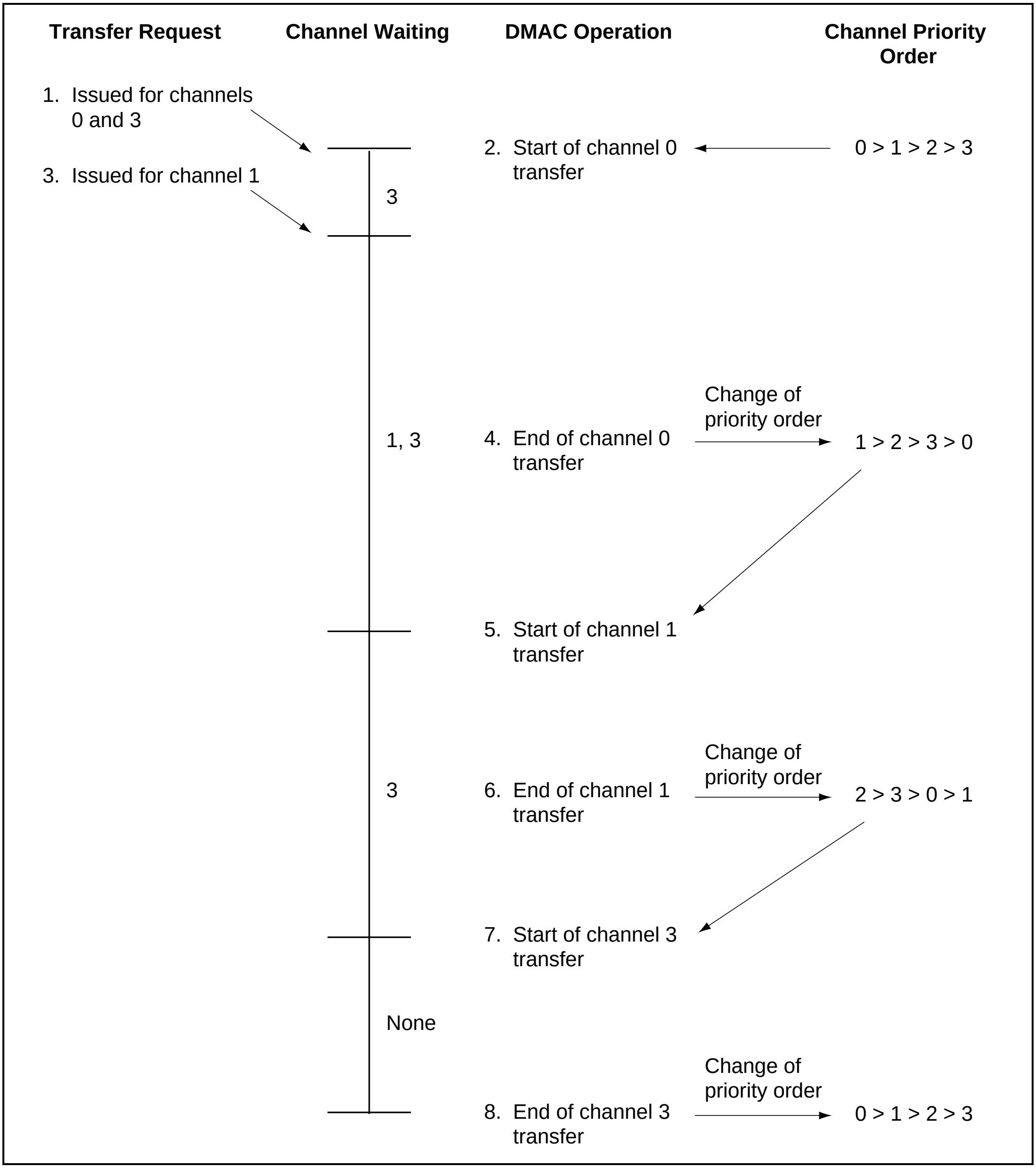


Figure 9.4 Example of Changes in Channel Priority Order in Round Robin Mode

9.3.4 Types of DMA Transfer

The DMAC supports the transfers shown in table 9.5. 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.

Table 9.5 Supported DMA Transfers

	Transfer Destination				
Transfer Source	**External Device with DACK**	**External Memory**	**Memory-Mapped External Device**	**On-Chip Memory**	**On-Chip Peripheral Module**
External device with DACK	Not available	Single address mode	Single address mode	Not available	Not available
External memory	Single address mode	Dual 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	Dual address mode
On-chip memory	Not available	Dual address mode	Dual address mode	Dual address mode	Dual address mode
On-chip peripheral module	Not available	Dual address mode	Dual address mode	Dual address mode	Dual address mode

Address Modes

Single Address Mode: In single address mode, both the transfer source and the transfer destination are external; one is accessed by the $\overline{\text{DACK}}$ 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 I/O strobe signal ($\overline{\text{DACK}}$) 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 9.5 shows an example of a transfer between external memory and an external device with $\overline{\text{DACK}}$ in which the external device outputs data to the data bus while that data is written to external memory in the same bus cycle.

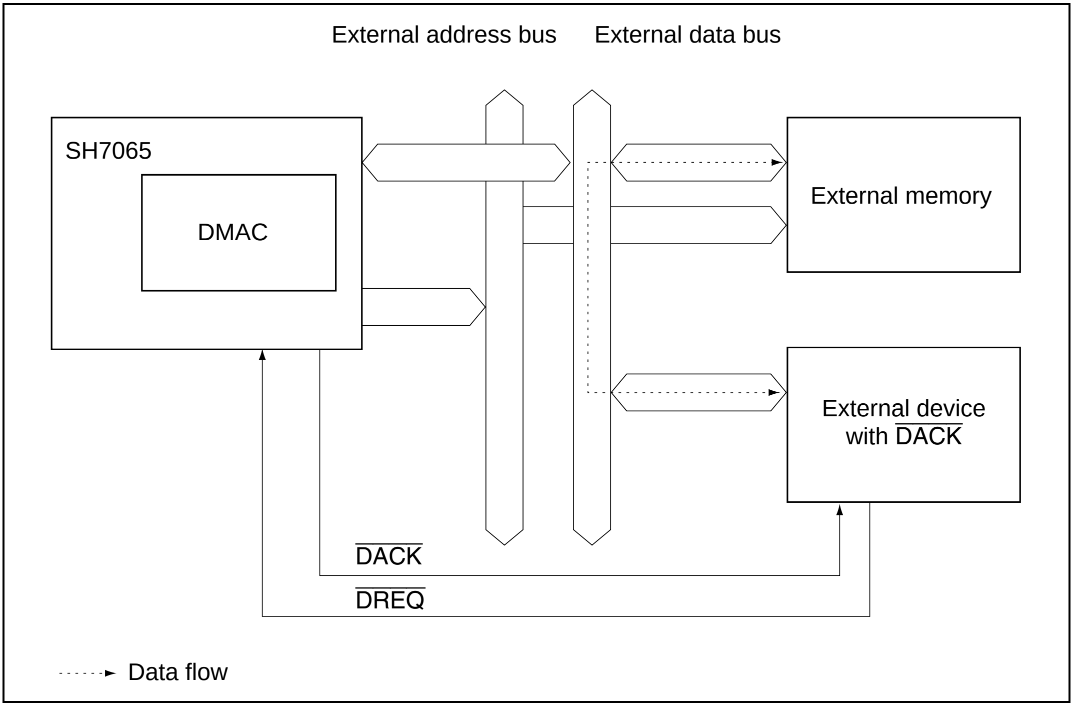


Figure 9.5 Data Flow in Single Address Mode

Two types of transfer are possible in single address mode: (1) transfer between an external device with $\overline{\text{DACK}}$ and a memory-mapped external device, and (2) transfer between an external device with $\overline{\text{DACK}}$ and external memory. Only the external request signal ($\overline{\text{DREQ}}$) is used in both these cases.

Figure 9.6 shows the DMA transfer timing in single address mode.

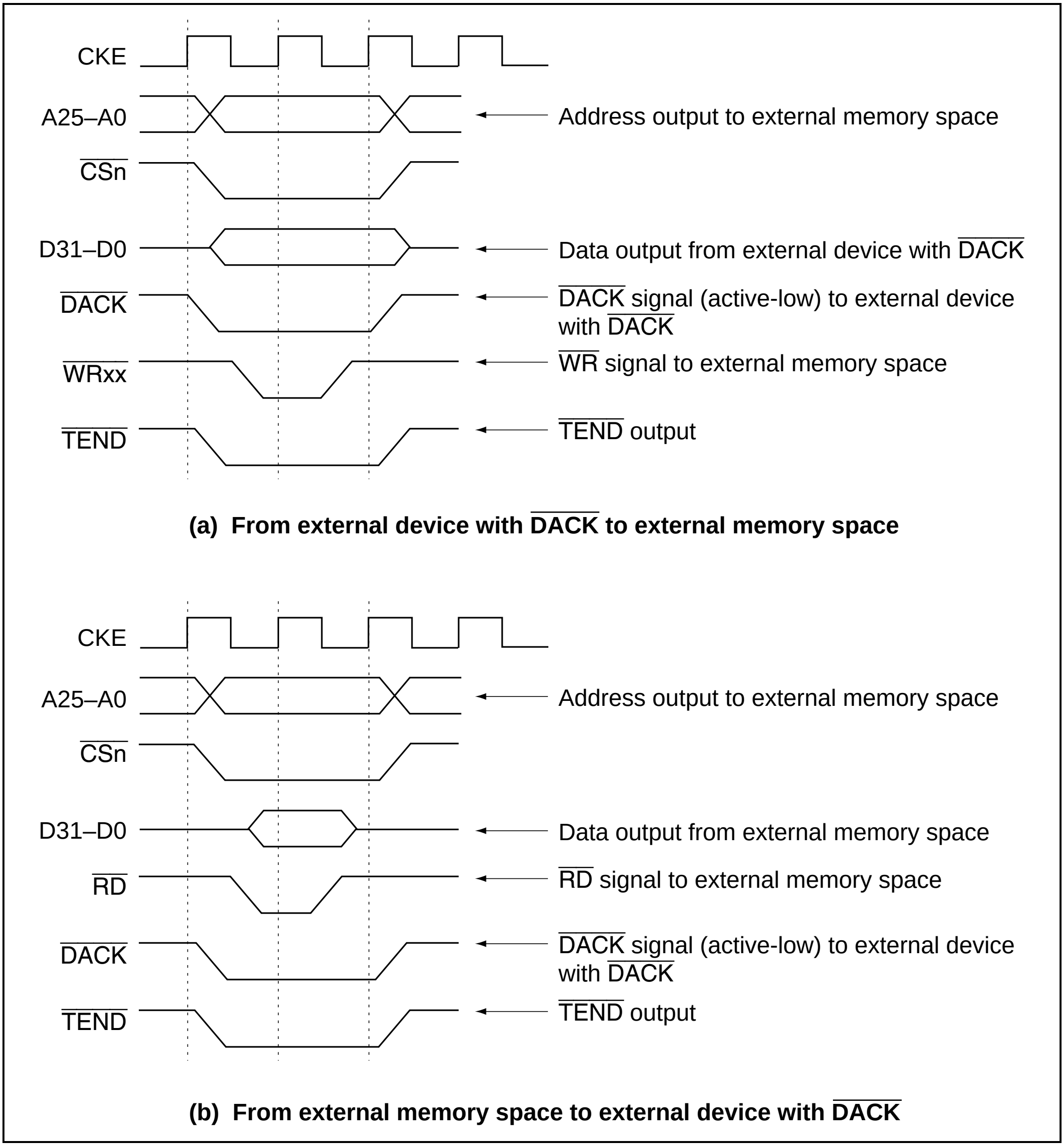
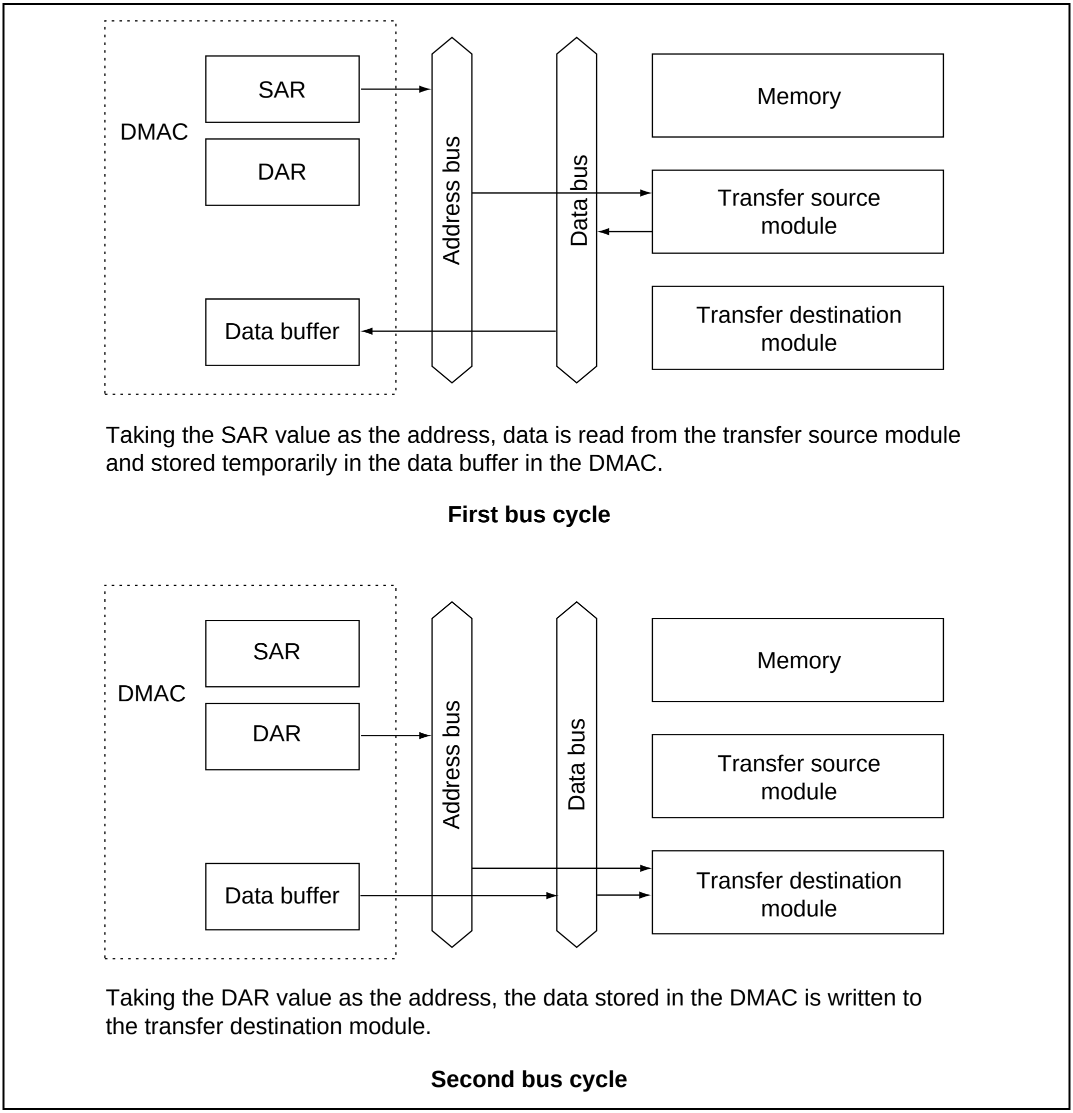

CKE
A25–A0
CSn
D31–D0
DACK
WRxx
TEND
Address output to external memory space
Data output from external device with DACK
DACK signal (active-low) to external device with DACK
WR signal to external memory space
TEND output
(a) From external device with DACK to external memory space
CKE
A25–A0
CSn
D31–D0
RD
DACK
TEND
Address output to external memory space
Data output from external memory space
RD signal to external memory space
DACK signal (active-low) to external device with DACK
TEND output
(b) From external memory space to external device with DACK

Figure 9.6 DMA Transfer Timing in Single Address Mode

Dual 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 accessed either internally or externally.

In dual address mode, data is read from the transfer source in the data read cycle, and written to the transfer destination in the data write cycle, so that the transfer is executed in two bus cycles. The transfer data is temporarily stored in the DMAC. In a transfer between external memories such as that shown in figure 9.7, data is read from external memory into the DMAC in the read cycle, then written to the other external memory in the write cycle. Figure 9.8 shows the timing for this operation.



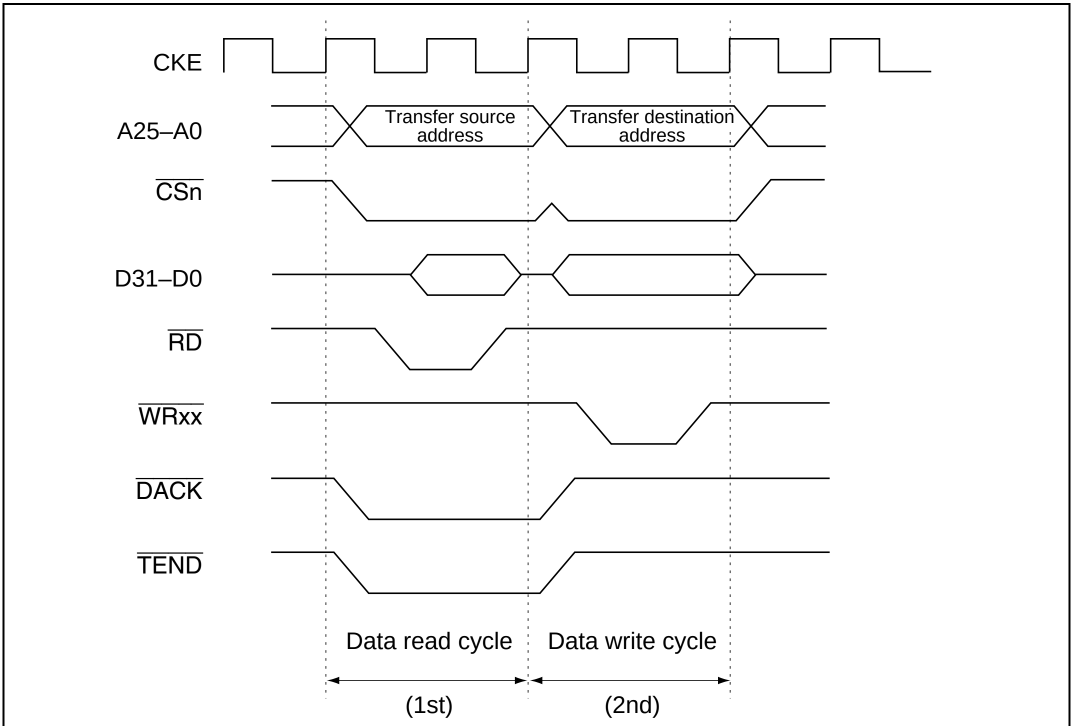
Taking the SAR value as the address, data is read from the transfer source module and stored temporarily in the data buffer in the DMAC.

First bus cycle

Taking the DAR value as the address, the data stored in the DMAC is written to the transfer destination module.

Second bus cycle

Figure 9.7 Direct Address Operation in Dual Address Mode



Transfer from external memory space to external memory space, TEND output enabled, DACK output in read cycle

Figure 9.8 Example of Dual Address Mode Transfer Timing

Bus Modes

There are two bus modes, cycle steal mode and burst mode, selected with the TM bit in CHCR0 to CHCR3.

Cycle Steal Mode: In cycle steal mode, the DMAC gives the bus to another bus master at the end of each transfer-unit (8-bit, 16-bit, or 32-bit) transfer. When the next transfer request is issued, the DMAC reacquires the bus from the other bus master and carries out another transfer-unit transfer. At the end of this transfer, the bus is again given to another bus master. This is repeated until the transfer end condition is satisfied.

Cycle steal mode can be used with all categories of transfer request source, transfer source, and transfer destination.

Figure 9.9 shows an example of DMA transfer timing in cycle steal mode. The transfer conditions in this example are dual address mode and $\overline{\text{DREQ}}$ level detection (using the 16-stage FIFO).

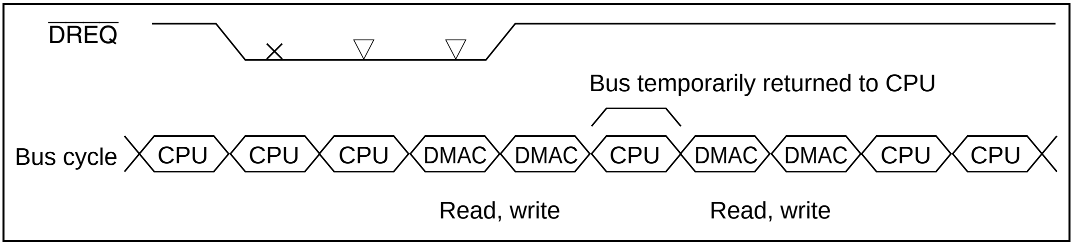


Figure 9.9 Example of DMA Transfer in Cycle Steal Mode

Burst Mode: In burst mode, once the DMAC has acquired the bus it 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 9.10 shows an example of DMA transfer timing in burst mode. The transfer conditions in this example are single address mode and $\overline{\text{DREQ}}$ level detection (using the 16-stage FIFO).

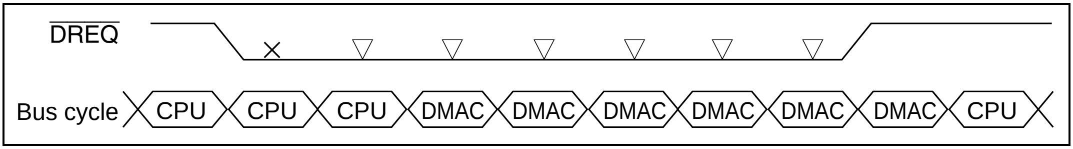


Figure 9.10 Example of DMA Transfer in Burst Mode

Relationship between DMA Transfer Type, Request Mode, and Bus Mode

Table 9.6 shows the relationship between the type of DMA transfer, the request mode, and the bus mode.

Table 9.6 Relationship between DMA Transfer Type, Request Mode, and Bus Mode

Address Mode	Type of Transfer	Request Mode	Bus Mode	Transfer Size (Bits)	Usable Channels
Single	External device with DACK and external memory	External	B/C	8/16/32	0, 1
	External device with DACK and memory-mapped external device	External	B/C	8/16/32	0, 1
Dual	External memory and external memory	Any*1	B/C	8/16/32	0–3
	External memory and memory-mapped external device	Any*1	B/C	8/16/32	0–3
	Memory-mapped external device and memory-mapped external device	Any*1	B/C	8/16/32	0–3
	External memory and on-chip memory	Any*1	B/C	8/16/32	0–3
	External memory and on-chip peripheral module	Any*2	B/C	8/16/32*3	0–3
	Memory-mapped external device and on-chip memory	Any*1	B/C	8/16/32	0–3
	Memory-mapped external device and on-chip peripheral module	Any*2	B/C	8/16/32*3	0–3
	On-chip memory and on-chip memory	Any*1	B/C	8/16/32	0–3
	On-chip memory and on-chip peripheral module	Any*2	B/C	8/16/32*3	0–3
	On-chip peripheral module and on-chip peripheral module	Any*2	B/C	8/16/32*3	0–3

Legend: B: Burst
C: Cycle steal

Notes: 1. External request, auto-request, or on-chip peripheral module request possible. The SCI or A/D converter cannot be specified as the transfer request source in on-chip peripheral module request mode.

2. External request, auto-request, or on-chip peripheral module request possible. If the transfer request source is also the SCI or A/D converter, the transfer source or transfer destination must be the SCI or A/D converter.

3. Access size permitted for register of on-chip peripheral module that is the transfer source or transfer destination.

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 order (CH0 > CH1), or if burst mode is set for channel 0, transfer on channel 1 is continued after transfer on channel 0 is completely finished.

If round robin mode has been set for the priority order, transfer on channel 1 is restarted after one transfer unit of data is transferred on channel 0, regardless of whether cycle steal mode or burst mode is set for channel 0. Bus mastership then alternates in the order: channel 1 → channel 0 → channel 1 → channel 0.

Since channel 1 is in burst mode, the bus is not given to the CPU during this period, regardless of whether fixed mode or round robin mode is set for the priority order.

An example of round robin mode operation is shown in figure 9.11.

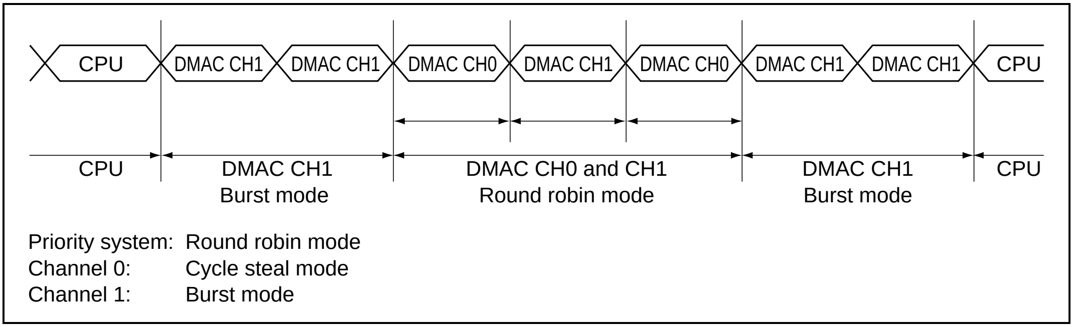


Figure 9.11 Bus Handling with Two DMAC Channels Operating

9.3.5 Number of Bus Cycle States and $\overline{\text{DREQ}}$ Pin Sampling Timing

Number of Bus Cycle States

When the DMAC is the bus master, the number of bus cycle states is controlled by the bus state controller (BSC) in the same way as when the CPU is the bus master. For details, see section 8, Bus State Controller (BSC).

$\overline{\text{DREQ}}$ Pin Sampling and $\overline{\text{DRAK}}$ Signal

In external request mode, the $\overline{\text{DREQ}}$ pin for each channel is sampled using falling edge or low level detection. The $\overline{\text{DREQ}}$ sampling circuit for each channel comprises a noise canceler and edge detection circuit, a 16-stage FIFO, and a 1-stage FIFO.

Regardless of whether $\overline{\text{DREQ}}$ pin falling edge detection or low level detection is used, the signal is sampled via a noise canceler circuit. The noise canceler eliminates noise of one clock cycle or less in duration by ignoring the first clock cycle of $\overline{\text{DREQ}}$ input.

After passing through the noise canceler, an external request signal is sampled by one of three $\overline{\text{DREQ}}$ sampling methods, as selected by the relevant register settings: falling edge detection, low level detection using the 16-stage FIFO, or low level detection using the 1-stage FIFO.

Whichever $\overline{\text{DREQ}}$ sampling method is selected, the $\overline{\text{DRAK}}$ signal is output for one CKE state each time $\overline{\text{DREQ}}$ is sampled. Regardless of whether cycle steal or burst mode is selected, and of the $\overline{\text{DREQ}}$ sampling method, the $\overline{\text{DRAK}}$ signal is output when generation of a DMA transfer cycle in response to the sampled $\overline{\text{DREQ}}$ signal is confirmed. $\overline{\text{DRAK}}$ signal output is synchronized with CKE, but it is not possible to stipulate the output timing relative to the external bus cycle.

- $\overline{\text{DREQ}}$ falling edge detection

  When $\overline{\text{DREQ}}$ falling edge detection is used, $\overline{\text{DREQ}}$ samples are not stored in a FIFO, and DMA transfer is carried out in response to a single falling edge. Since the noise canceler function ignores the first state of $\overline{\text{DREQ}}$ input, the $\overline{\text{DREQ}}$ signal must be input for at least two states. $\overline{\text{DREQ}}$ sampling is performed in the same way regardless of whether single or dual address mode, or cycle steal or burst mode, is selected.

  With $\overline{\text{DREQ}}$ falling edge detection, the number of transfers to be initiated by detection of a single falling edge can be set, regardless of whether single or dual address mode, or cycle steal or burst mode, is selected. The following settings can be made with the Flag Clear Timing Select (FCS) bit in the channel control register (CHCR) for each channel.

- Execution of one transfer in response to one falling edge
- Execution of the number of transfers set in the DMA transfer count register (DMATCR) in response to one falling edge

Regardless of the setting of the Flag Clear Timing Select (FCS) bit, while the transfer initiated by the previously detected falling edge is in progress, there is a period during which the next $\overline{\text{DREQ}}$ falling edge is ignored. Therefore, the next falling edge should not be input until the final $\overline{\text{DRAK}}$ signal has been output for the currently executing transfer.

The $\overline{\text{DRAK}}$ signal is output once only for one falling edge. Regardless of the setting of the Flag Clear Timing Select (FCS) bit, it is output at the same time as, or earlier than, address output in the first DMA transfer cycle.

Figure 9.12 shows an example of the operation when one DMA transfer is performed in response to one falling edge. The example in figure 9.12 is for cycle steal mode, but even if burst mode is selected, the operation still ends after one transfer in response to one falling edge. Figure 9.13 shows an example of the operation when the number of DMA transfers set in the DMA transfer count register (DMATCR) are performed in response to one falling edge.

- $\overline{\text{DREQ}}$ low level detection using 16-stage FIFO

  When $\overline{\text{DREQ}}$ low level detection is selected, $\overline{\text{DREQ}}$ is sampled in every cycle at the rise of CKE (figure 9.14). The noise canceler function prevents sampling of $\overline{\text{DREQ}}$ input at the first rise of CKE. The sampled $\overline{\text{DREQ}}$ signal is stored in the 16-stage FIFO. One DMA transfer is performed for one stored sampling result, and a number of transfers corresponding to the number of stored samples are always carried out. If $\overline{\text{DREQ}}$ is input continuously, samples are taken until the FIFO is full; once the FIFO is full, the $\overline{\text{DREQ}}$ input is no longer sampled (figure 9.15). The FIFO is incremented each time $\overline{\text{DREQ}}$ is sampled, and is decremented when generation of the next DMA transfer cycle is confirmed. It is not always possible to stipulate the FIFO decrement timing relative to the external bus cycle, but in the case of a CKE:CKM frequency division ratio of 1:1, it will be at the start of the bus cycle before the DMA transfer cycle (timings (A), (B), (C), (D), and (E) in figure 9.14). With the 16-stage FIFO, in the event of contention between FIFO incrementing and decrementing, both are performed simultaneously. The FIFO operation in this case is as shown at (A), (B), (C), (D), and (E) in figure 9.14. The $\overline{\text{DRAK}}$ signal is output one state after the FIFO is decremented, and at the same time as, or earlier than, address output in the corresponding DMA transfer cycle. The $\overline{\text{DRAK}}$ signal is output once for each $\overline{\text{DREQ}}$ sample.

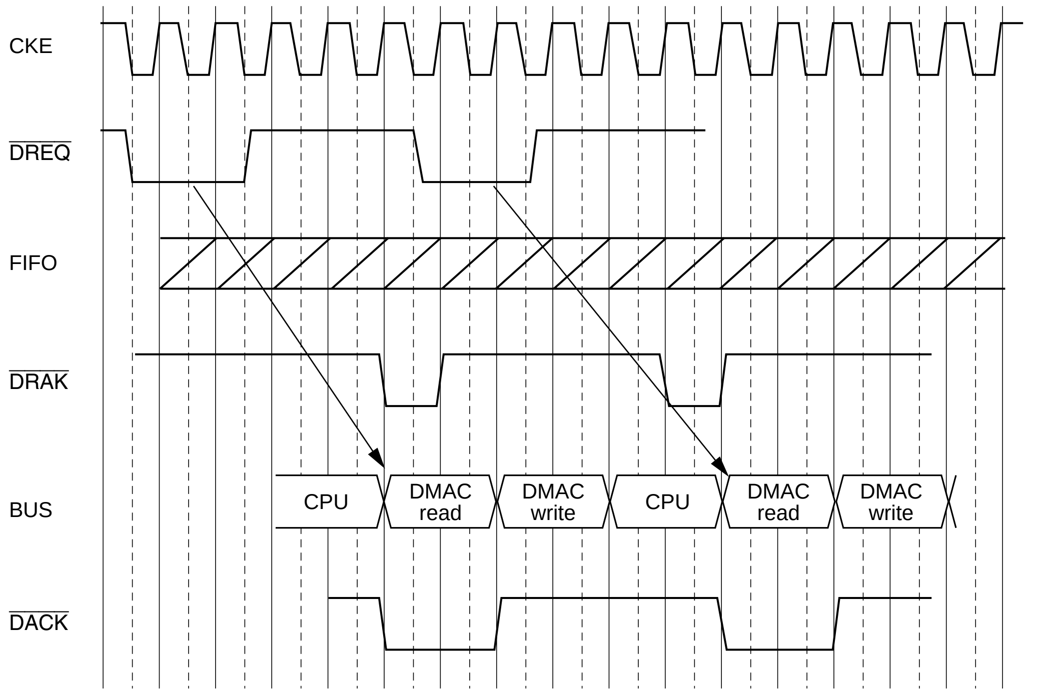
  With this $\overline{\text{DREQ}}$ sampling method, sampling is carried out in the same way regardless of whether single or dual address mode, or cycle steal or burst mode, is selected (figures 9.16 to 9.18).

- $\overline{\text{DREQ}}$ low level detection using 1-stage FIFO

  $\overline{\text{DREQ}}$ low level sampling using the 1-stage FIFO should be selected when external bus cycles are two CKE states or longer in maximum-speed operation. If external bus cycles are two CKE states or longer in maximum-speed operation, when $\overline{\text{DREQ}}$ input is halted upon $\overline{\text{DRAK}}$ signal output, one subsequent DMA transfer will always be performed after the DMA transfer cycle corresponding to this $\overline{\text{DRAK}}$ signal output before transfer is halted (figure 9.19). If low level detection using the 1-stage FIFO is selected when the external bus cycle is one CKE state in maximum-speed operation, the amount of DMA cycle overrun cannot be guaranteed.

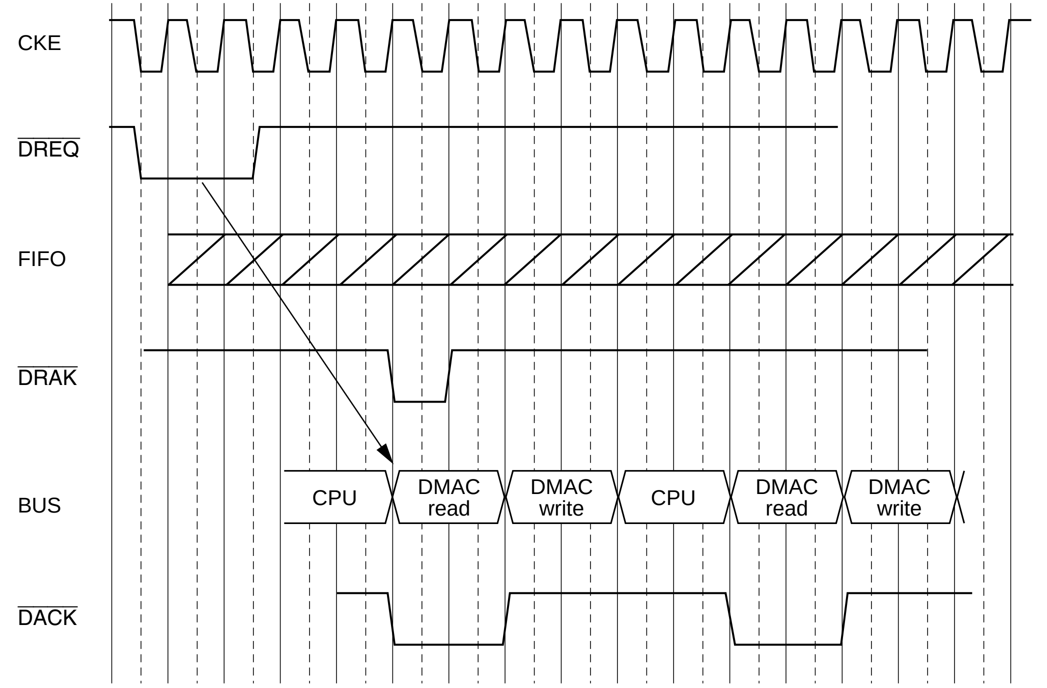
  With the 1-stage FIFO, as with the 16-stage FIFO, $\overline{\text{DREQ}}$ sampling is performed at the rise of CKE. The noise canceler function prevents sampling of $\overline{\text{DREQ}}$ input at the first rise of CKE. The sampled $\overline{\text{DREQ}}$ signal is stored in the 1-stage FIFO, and the FIFO is full after one sample is taken. The $\overline{\text{DRAK}}$ signal is output at the same time as, or earlier than, address output in the corresponding DMA transfer cycle, and is output once for each $\overline{\text{DREQ}}$ sample.

  The sampling conditions for the 1-stage FIFO are different from those for the 16-stage FIFO. With the 16-stage FIFO, FIFO incrementing and decrementing are performed simultaneously (see figure 9.19), but with the 1-stage FIFO, after the FIFO is cleared by decrementing, the next sampling operation is performed. The FIFO decrement timing is the same as the $\overline{\text{DRAK}}$ signal output timing. Figure 9.19 shows an example of the operation when single/burst mode DMA transfer is carried out with $\overline{\text{DREQ}}$ sampling by low level detection using the 1-stage FIFO. In figure 9.19, when $\overline{\text{DREQ}}$ input is halted (B in the figure) at the point at which $\overline{\text{DRAK}}$ is output (A in the figure), transfer is terminated after execution of the DMA transfer cycle following the corresponding DMA cycle. With this $\overline{\text{DREQ}}$ sampling method, sampling is carried out in the same way regardless of whether single or dual address mode, or cycle steal or burst mode, is selected (figures 9.20 to 9.22).



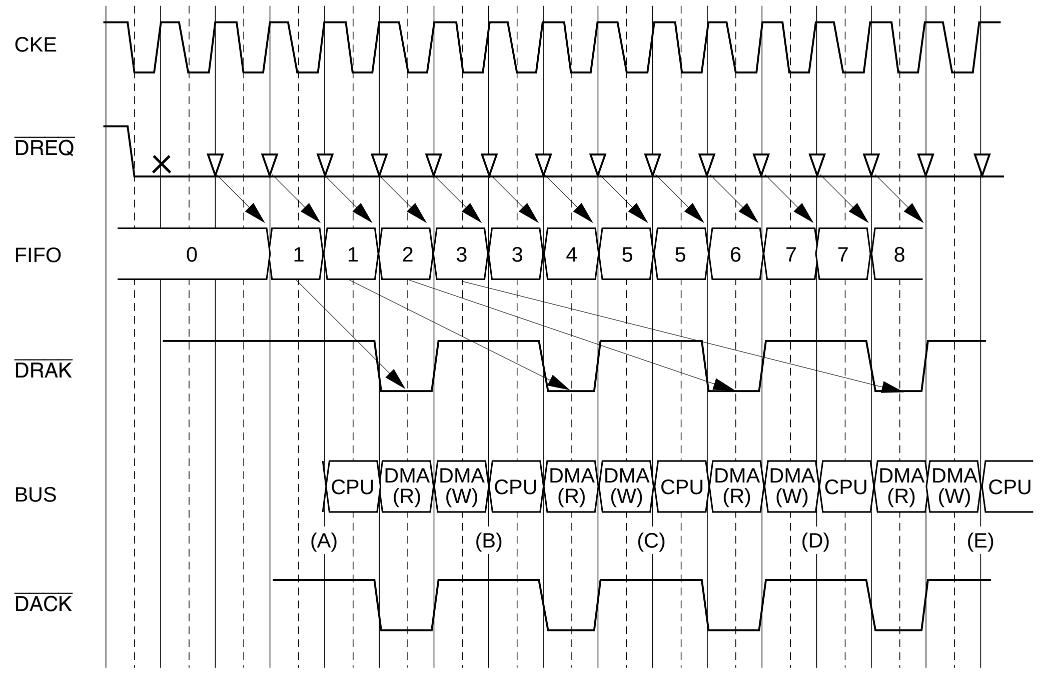
Notes:
1. CKE:CKM = 1:1
2. Cycle steal mode/dual address transfer
3. One transfer is performed for one edge.
4. $\overline{\text{DRAK}}$ and $\overline{\text{DACK}}$ are active-low.

Figure 9.12 Operation Example: CKE = CKM, Edge Detection, One Transfer for One Edge



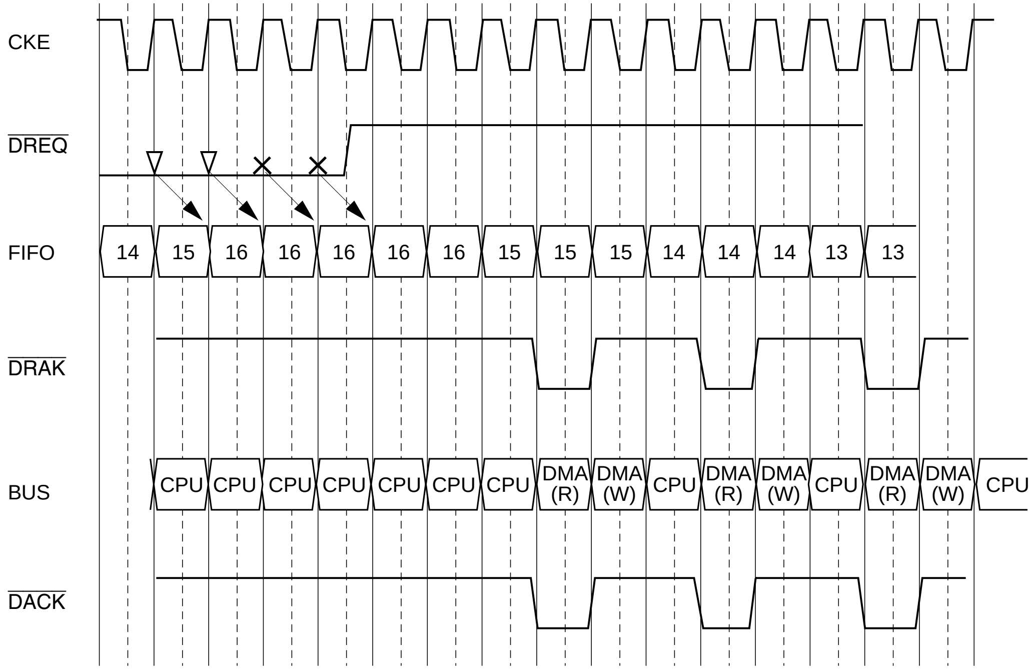
Notes: 1. CKE:CKM = 1:1
2. Cycle steal mode/dual address transfer
3. Number of transfers set in DMATCR are performed for one edge.
4. $\overline{\text{DRAK}}$ and $\overline{\text{DACK}}$ are active-low.

Figure 9.13 Operation Example:
CKE = CKM, Edge Detection, Set Number of Transfers for One Edge



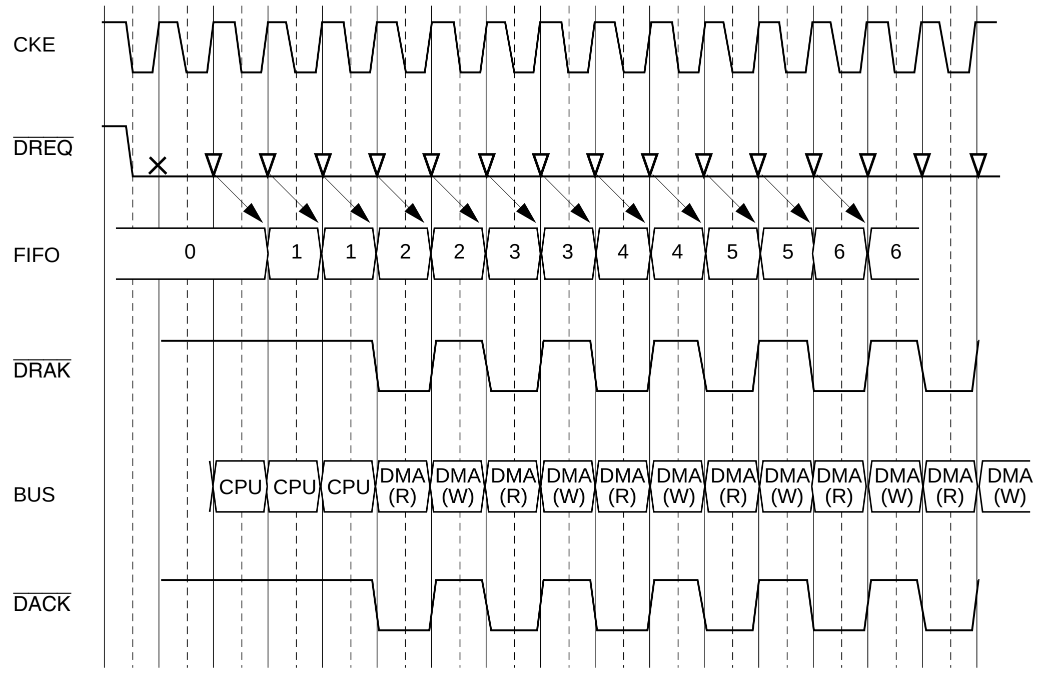
Notes: 1. CKE:CKM = 1:1
2. Cycle steal mode/dual address transfer
3. Low level detection using 16-stage FIFO
4. DRAK and DACK are active-low.

Figure 9.14 Operation Example:
CKE = CKM, Low Level Detection, 16-Stage FIFO Used (1)
(Maximum-Speed Operation In Dual Address/Cycle Steal Mode)



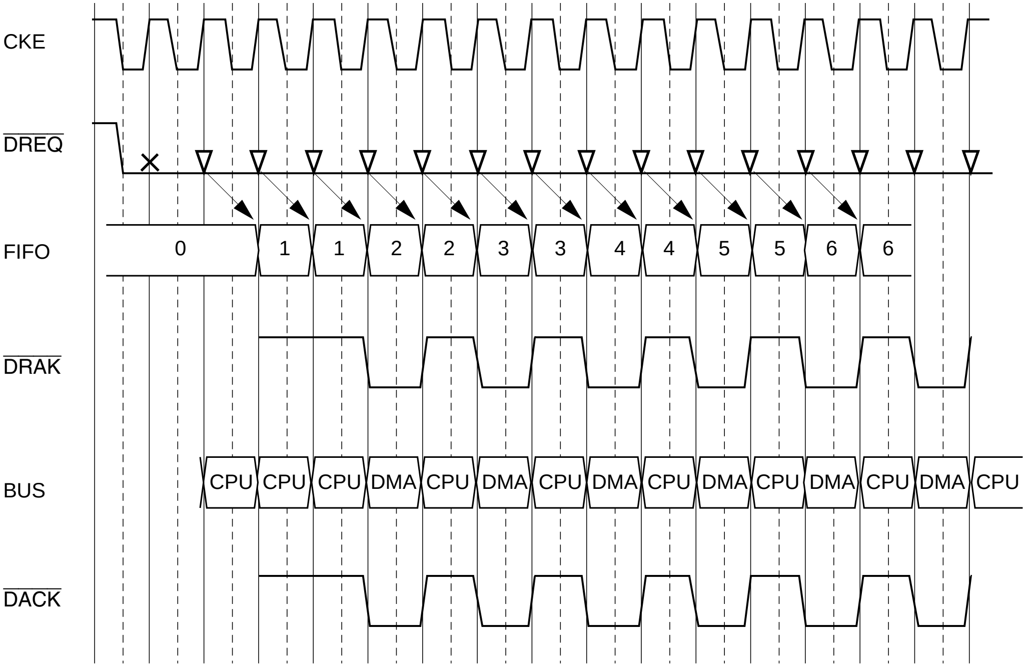
Notes: 1. CKE:CKM = 1:1
2. Cycle steal mode/dual address transfer
3. Low level detection using 16-stage FIFO
4. $\overline{\text{DRAK}}$ and $\overline{\text{DACK}}$ are active-low.

Figure 9.15 Operation Example:
CKE = CKM, Low Level Detection, 16-Stage FIFO Used (2)
Sampling Operation when FIFO = FULL
(Dual Address/Cycle Steal Mode)



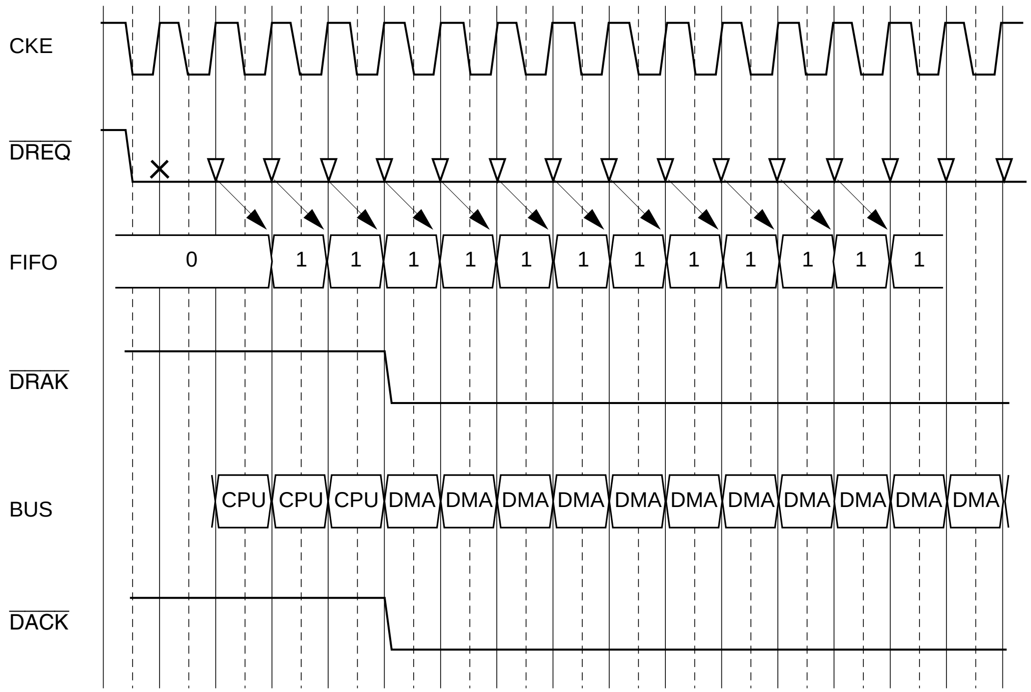
Notes: 1. CKE:CKM = 1:1
2. Burst mode/dual address transfer
3. Low level detection using 16-stage FIFO
4. $\overline{\text{DRAK}}$ and $\overline{\text{DACK}}$ are active-low.

Figure 9.16 Operation Example: CKE = CKM, Low Level Detection, 16-Stage FIFO Used (3) (Dual Address/Burst Mode)



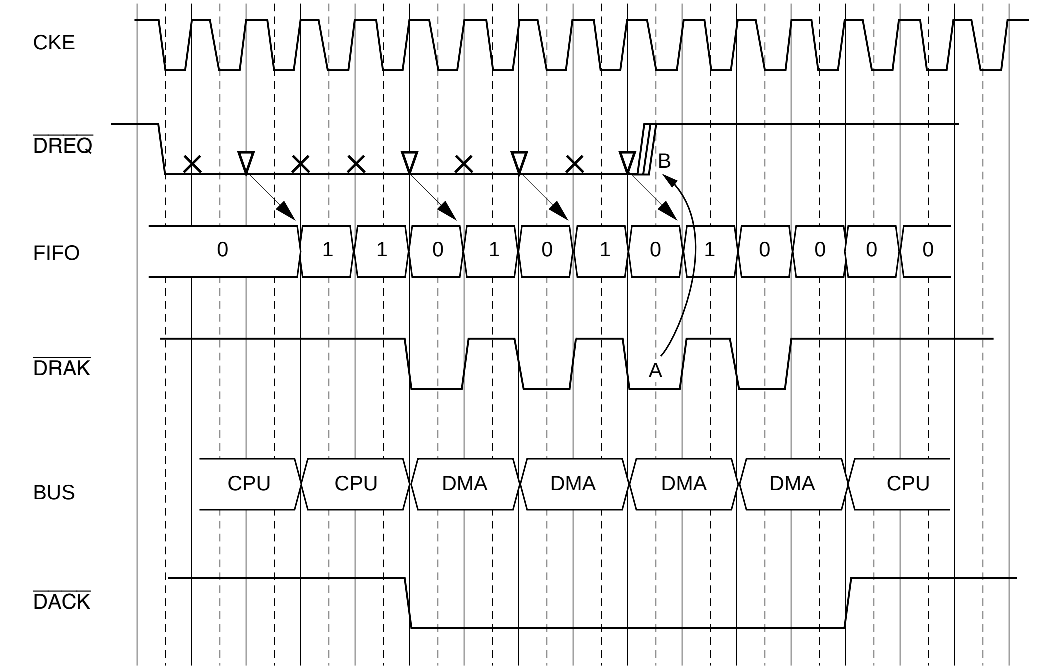
Notes: 1. CKE:CKM = 1:1
2. Cycle steal mode/single address transfer
3. Low level detection using 16-stage FIFO
4. $\overline{\text{DRAK}}$ and $\overline{\text{DACK}}$ are active-low.

Figure 9.17 Operation Example:
CKE = CKM, Low Level Detection, 16-Stage FIFO Used (4)
(Single Address/Cycle Steal Mode)



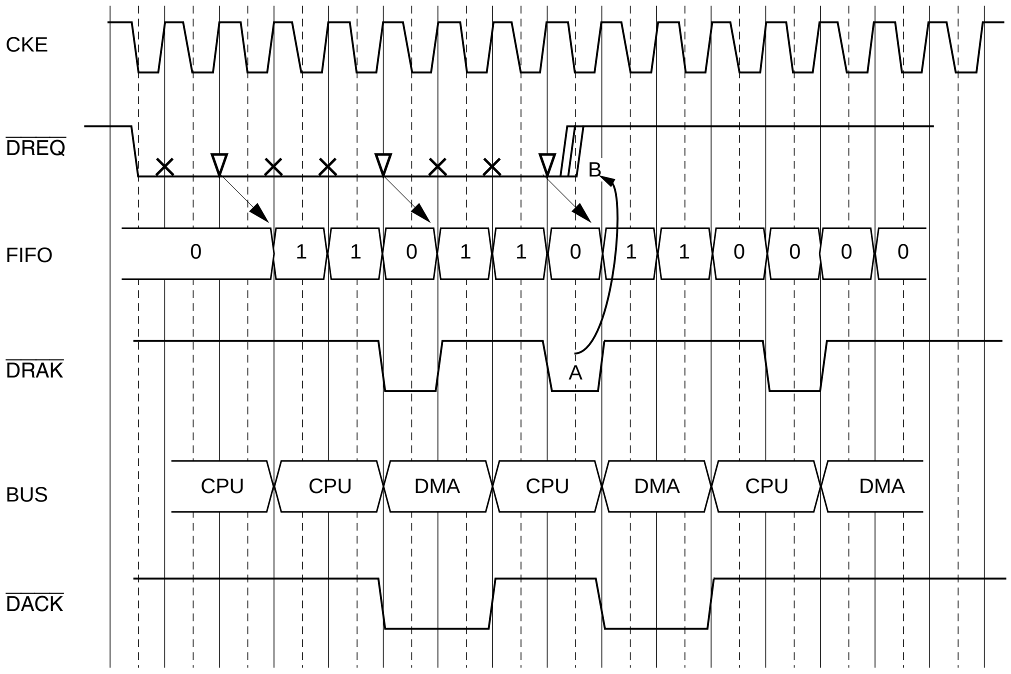
Notes: 1. CKE:CKM = 1:1
2. Burst mode/single address transfer
3. Low level detection using 16-stage FIFO
4. $\overline{\text{DRAK}}$ and $\overline{\text{DACK}}$ are active-low.

Figure 9.18 Operation Example: CKE = CKM, Low Level Detection, 16-Stage FIFO Used (5) (Single Address/Burst Mode)



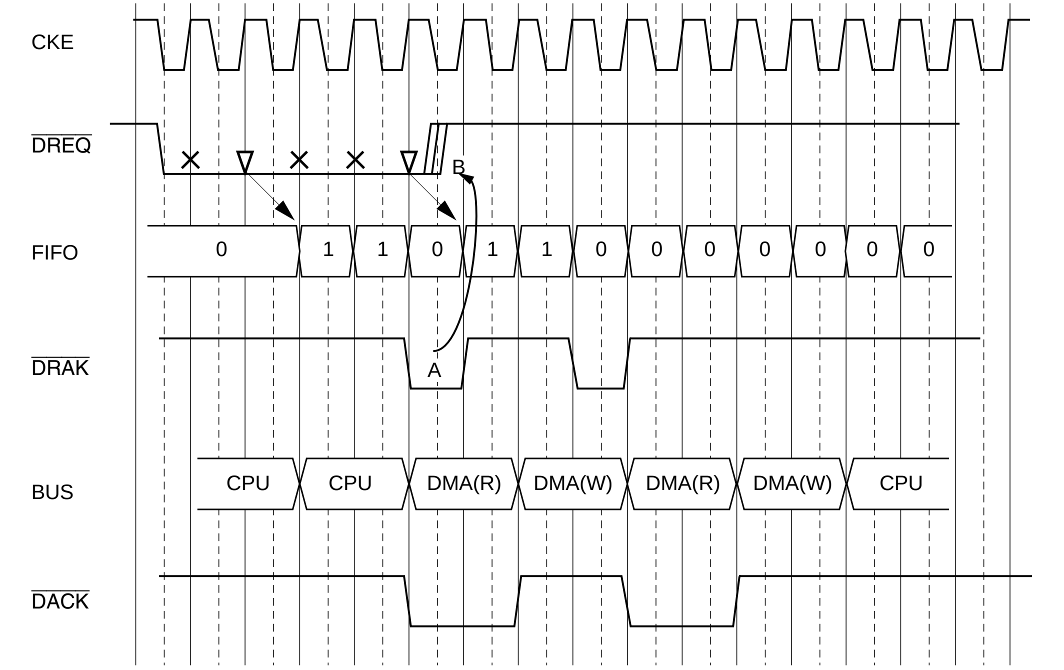
Notes: 1. CKE:CKM = 1:1
2. Burst mode/single address transfer
3. Low level detection using 16-stage FIFO
4. DRAK and DACK are active-low.

**Figure 9.19 Operation Example:
CKE = CKM, Low Level Detection, 1-Stage FIFO Used (1)
(Single Address/Burst Mode/Maximum-Speed Operation)**



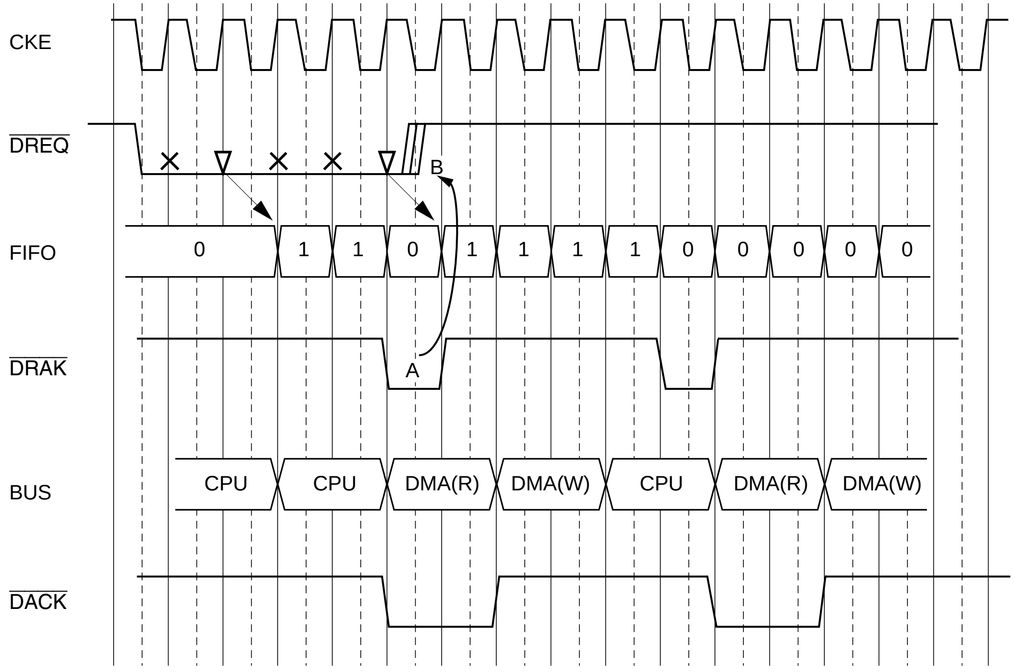
Notes: 1. CKE:CKM = 1:1
2. Cycle steal mode/single address transfer
3. Low level detection using 16-stage FIFO
4. $\overline{\text{DRAK}}$ and $\overline{\text{DACK}}$ are active-low.

**Figure 9.20 Operation Example:
CKE = CKM, Low Level Detection, 1-Stage FIFO Used (2)
(Single Address/Cycle Steal Mode)**



Notes: 1. CKE:CKM = 1:1
2. Burst mode/dual address transfer
3. Low level detection using 16-stage FIFO
4. $\overline{\text{DRAK}}$ and $\overline{\text{DACK}}$ are active-low.

**Figure 9.21 Operation Example:
CKE = CKM, Low Level Detection, 1-Stage FIFO Used (3)
(Dual Address/Burst Mode)**



Notes: 1. CKE:CKM = 1:1
2. Cycle steal mode/dual address transfer
3. Low level detection using 16-stage FIFO
4. $\overline{\text{DRAK}}$ and $\overline{\text{DACK}}$ are active-low.

**Figure 9.22 Operation Example:
CKE = CKM, Low Level Detection, 1-Stage FIFO Used (4)
(Dual Address/Cycle Steal Mode)**

Clock Frequency Division Restrictions Relating to $\overline{\text{DREQ}}$ Sampling

In the SH7065, the frequency of the master clock (CKM) and the external bus clock (CKE) can be set independently by means of settings in the frequency control register (FRQCR) in the clock pulse generator (CPG). When the DMAC is activated by an external request, the frequency of the master clock (CKM) must be set as equal to or higher than that of the external bus clock (CKE). If the master clock (CKM) frequency is set to a value lower than the external bus clock (CKE) frequency, sampling will not be performed correctly with either falling edge detection or low level detection, and it will not be possible to stipulate the number of DMA transfers performed in response to an external request. For details of clock frequency settings, see section 4, Clock Pulse Generator (CPG) and Power-Down Modes.

9.3.6 Parallel Operation of DMA and CPU

The SH7065 has two 32-bit internal buses, the C-bus and I-bus. DMA is never the master on the C-bus, so the CPU can access mask ROM and on-chip flash memory from the C-bus during DMA transfer. However, when the DMA controller is accessing X-RAM or Y-RAM, the CPU cannot simultaneously access the same RAM.

The combinations of access spaces for which parallel DMA and CPU operation is possible are shown in table 9.7.

9.3.7 DMA Transfer When External Bus Is Released

If the DMA transfer source and transfer destination are both on-chip memory or on-chip peripheral modules, DMA transfer cannot be performed while the external bus is released. The combinations of transfer source and transfer destination for which DMA transfer is possible while the external bus is released are shpown in table 9.8.

Table 9.7 Contention between DMA and CPU

Mode	DMA Transfer	External	On-Chip ROM/Flash	On-Chip Memory	On-Chip Peripheral Module
Single	External device with DACK and external memory	X	O	O	X
	External device with DACK and memory-mapped external device	X	O	O	X
Dual	External memory and external memory	X	O	O	X
	External memory and memory-mapped external device	X	O	O	X
	External memory and on-chip memory	X	O	Δ	X
	External memory and on-chip peripheral module	X	O	O	X
	Memory-mapped external device and memory-mapped external device	X	O	O	X
	Memory-mapped external device and on-chip memory	X	O	O	X
	Memory-mapped external device and on-chip peripheral module	X	O	O	X
	On-chip memory and on-chip memory	X	O	Δ	X
	On-chip memory and on-chip peripheral module	X	O	Δ	X
	On-chip peripheral module and on-chip peripheral module	X	O	O	X

Legend:

O: Parallel DMA and CPU operation possible

X: Parallel DMA and CPU operation not possible because of bus contention

Δ: On-chip memory consists of X-RAM and Y-RAM. As X-RAM and Y-RAM can be accessed independently, parallel operation is possible when different RAM is accessed by DMA and by the CPU.

Table 9.8 Contention between DMA and External Bus Release

		Transfer Destination				
		External Device with DACK	**External Memory**	**Memory-Mapped External Device**	**On-Chip Memory**	**On-Chip Peripheral Module**
Transfer source	External device with DACK	DMA transfer not possible	X	X	DMA transfer not possible	DMA transfer not possible
	External memory	X	X	X	X	X
	Memory-mapped external device	X	X	X	X	X
	On-chip memory	DMA transfer not possible	X	X	O	O
	On-chip peripheral module	DMA transfer not possible	X	X	O	O

Legend:

O: DMA transfer possible while external bus is released

X: DMA transfer not possible while external bus is released

9.3.8 Chain Transfer

Use of chain transfer allows a specified block of data to be transferred consecutively without CPU processing after the end of the current data transfer.

To perform chain transfer, it is necessary to set the registers used for chain transfer—the next source address register (NSAR), next destination address register (NDAR), next transfer count register (NDMATCR), and chain transfer count register (CHNCNT)—and to set the chain transfer enable bit (CHNE) to 1 in the channel control register (CHCR). When the number of chain transfers set in the transfer count register (DMATCR) are completed while chain transfer is enabled, in the state following the end of transfer the set values are copied from NSAR into SAR, from NDAR into DAR, and from NDMATCR into DMATCR, and the DMAC waits for the next transfer request (figure 9.2). However, whether or not copying of NSAR and NDAR is necessary can be specified by means of the next source address register enable bit (NSARE) and next destination address register enable bit (NDARE) in CHCR. Register copying is performed the number of times set in the chain transfer count register (CHNCNT), and chain transfer ends when the value in CHNCNT reaches 0. There is no auto-request chain transfer, and transfer always ends on completion of the first transfer.

As an example of chain transfer, table 9.9 shows the settings when the data stored in external memory addresses H'04000000 to H'04001000 is transferred in eight 512-byte transfers to the same address space in XRAM. In this example, DMAC channel 0 is used and the transfer request source is DREQ0 falling edge detection.

Table 9.9 Sample Chain Transfer Settings

Transfer Conditions	Register	Set Value
Transfer source: External memory	SAR0	H'04000000
Transfer destination: XRAM	DAR0	H'FFFF8000
Number of transfers: 128	DMATCR0	H'00000080
Transfer source and destination addresses: Incremented	CHCR0	H'000858F5
Transfer request source: DREQ0 (dual address)		
DREQ0 detection mode: Falling edge detection		
Edge clear timing: End of transfer		
Bus mode: Burst		
Transfer unit: Longword		
Chain transfer enabled		
Transfer source address copying in chain transfer disabled		
Transfer destination address copying in chain transfer enabled		
Interrupt request generated at end of chain transfer		
Transfer source address in chain transfer: Setting unnecessary	NSAR0	Setting unnecessary
Transfer destination address in chain transfer: XRAM	NDAR0	H'FFFF8000
Number of transfers in chain transfer: 128	NDMATCR0	H'00000080
Number of chain transfers: 7	CHNCNT0	H'00000007
Channel priority order: 0 > 1 > 2 > 3	DMA0R	H'0001

9.4 Example of Use

9.4.1 Example of DMA Transfer between On-Chip SCI and External Memory

In the example considered here, on-chip serial communication interface channel 2 (SCI2) receive data is transferred to external memory using DMAC channel 3. DMAC settings must be completed and transfer enabled before inputting a transfer request from the serial communication interface.

Table 9.10 shows the transfer conditions and register set values in this case.

Table 9.10 Example of Use

Transfer Conditions	Register	Set Value
Transfer source: RDR0 of on-chip SCI0	SAR3	H'FFFF0546
Transfer destination: External memory	DAR3	H'04000000
Number of transfers: 8	DMATCR3	H'00000008
Transfer source address: Fixed	CHCR3	H'13024045
Transfer destination address: Incremented		
Transfer request source: SCI0 (RX0)		
Bus mode: Cycle steal		
Transfer unit: Byte		
Interrupt requested at end of transfer		
Channel priority order: 0 > 1 > 2 > 3	DMAOR	H'0001

9.5 Usage Notes

1. Only word (16-bit) access can be used on the DMA operation register (DMAOR). Word (16-bit) or longword (32-bit) access can be used on all other registers.
2. When modifying bits RS0 to RS4 in CHCR0 to CHCR3, first clear the DE bit to 0 (when modifying CHCR, clear the DE bit to 0 beforehand).
3. The NMIF bit in DMAOR is set when an NMI interrupt is input even if the DMAC is not operating.
4. When setting standby mode, first clear the DME bit in DMAOR to 0 and wait until the DMAC has completed processing of all accepted transfer requests.
5. Do not access the DMAC, BSC, or UBC on-chip peripheral modules.

6. When activating the DMAC, make the CHCR setting as the final step. The DMAC may not operate normally if any other register setting is made last.
7. After the DMATCR count reaches 0 and DMA transfer ends normally, always write 0 to DMATCR even when executing the maximum number of transfers on the same channel. The DMAC may not operate normally if this is not done.
8. When using the round robin method to determine the priority order, more than one channel must be specified, and consecutive channel numbers must be specified (e.g. CH1, CH2, CH3). Operation cannot be guaranteed if non-consecutive channel numbers are specified. To change the specified channel, change the DMA operation register (DMAOR) setting when the channel priority order is the initial priority order.
9. When falling edge detection is used for external requests, keep the external request pin high when making DMAC settings.
10. When using the DMAC in single address mode, set an external address as the address. The DMAC may not operate normally if an internal address is set.
11. On-chip ROM space cannot be accessed.
12. The same internal request cannot be set for more than one channel. If it is, the request will only be valid for the channel with the highest default priority.
13. When a transfer request is accepted from an on-chip peripheral module, the relevant interrupt request signal is masked and not input to the INTC. For details of the masking conditions, see section 6, Interrupt Controller (INTC).
14. The DMAC internal registers cannot be accessed while the DMAC is operating. However, it is possible to perform access to DMAOR and CHCRn, and change the DME bit in DMAOR and the TE and DE bits in CHCRn to control DMAC operation. If any other bit in DMAOR or CHCRn is changed, the change of setting may not be reflected in DMA transfer following the change.
15. When performing chain transfer initiated by an on-chip module, the DS bit in CHCRn must be set to 1.
16. When chain transfer is disabled by means of the CHNE bit in CHCRn, either clear CHNCNTn to 0 or set the TES bit to 1 in CHCRn.
17. A transfer request should not be made until the DMAC register settings are completed (figure 9.2).

9.6 DMAC Restrictions

9.6.1 TEND Output

In on-chip DMAC channels 0 and 1 (channels 2 and 3 have no TEND pin and so are not affected), TEND output may not be asserted at the end of DMA transfer.

If a setting is made for a channel using TEND to output $\overline{\text{TEND}}$ in dual address mode write cycles (AM = 1 and TEND = 1), $\overline{\text{TEND}}$ is asserted normally.

9.6.2 Notes on Suspension of Transfer

The following bug occurs regarding transfer suspension in dual address mode.

1. Conditions for occurrence
   (1) The last transfer before suspension when transfer is suspended by DME bit clearance in the CHCR register or DME bit clearance in the DMAOR register of a channel set to dual address mode before that channel completes transfer
   (2) Transfer before suspension due to exception handling when an address error occurs in a write access on a channel set to dual address mode and burst mode
   (3) The last transfer before suspension due to exception handling when an NMI interrupt occurs during transfer on a channel set to dual address mode

2. Description of bug
   In a dual address mode transfer, “read --> write” should be performed, but the operation is “repeated read --> read”.

3. Countermeasures
   In the case of condition (1) above, the problem can be avoided by using the following procedure to perform suspension.
   a. Clear channel transfer requests for all channels set to dual address mode.
   b. Clear the DE bit of all channels set to single address mode.
   c. Perform a dummy read (in external space or on-chip peripheral space) for each channel used.
   d. Clear the DE bit or DME bit of the channel to be suspended.
   (The order of (a) and (b) is immaterial.)
   There is no way of avoiding this problem in the case of above conditions (2) and (3).

4. Detection and corrective measures for transfer restart

   If a channel's settings conform to the following conditions, bug occurrence can be detected.

   (1) With settings to increment/decrement the transfer destination address and increment/decrement the transfer source address

   If the DAR and DMATCR registers are read and there is a discrepancy between the amount of decrement of DMATCR and the amount of change of DAR from the initial setting, or if the DAR and SAR registers are read and there is a discrepancy between the amount of change of SAR and the amount of change of DAR from the initial setting, the bug has occurred.

   Before restarting transfer, write the value of one transfer back to DMATCR and SAR.

   (2) With settings to increment/decrement the transfer destination address and leave the transfer source address unchanged

   If the DAR and DMATCR registers are read and there is a discrepancy between the amount of decrement of DMATCR and the amount of change of DAR from the initial setting, the bug has occurred.

   Before restarting transfer, write the value of one transfer back to DMATCR.

Section 10 16-Bit Timer Pulse Unit (TPU)

10.1 Overview

The SH7065 has an on-chip 16-bit timer pulse unit (TPU) that comprises six 16-bit timer channels.

10.1.1 Features

The TPU has the following features:

- Maximum 16-pulse input/output
  - A total of 16 timer general registers (TGRs) are provided (four each for channels 0 and 3, and two each for channels 1, 2, 4, and 5), each of which can be set independently as an output compare/input capture register
  - TGRC and TGRD for channels 0 and 3 can also be used as buffer registers
- Selection of 8 counter input clocks for each channel
- The following operations can be set for each channel:
  - Waveform output at compare match: Selection of 0, 1, or toggle output
  - Input capture function: Selection of rising edge, falling edge, or both edge detection
  - Counter clear operation: Counter clearing possible by compare match or input capture
  - Synchronous operation: Multiple timer counters (TCNT) can be written to simultaneously
    Simultaneous clearing by compare match and input capture possible
    Register simultaneous input/output possible by counter synchronous operation
  - PWM mode: Any PWM output duty can be set
  - Maximum of 15-phase PWM output possible by combination with synchronous operation
- Buffer operation settable for channels 0 and 3
  - Input capture register double-buffering possible
  - Automatic rewriting of output compare register possible
- Phase counting mode settable independently for each of channels 1, 2, 4, and 5
  - Two-phase encoder pulse up/down-count possible
- Cascaded operation
  - Channel 1 (channel 4) input clock operates as 32-bit counter by setting channel 2 (channel 5) overflow/underflow
- Fast access via internal 16-bit bus
  - Fast access is possible via a 16-bit bus interface

- 26 interrupt sources
  - For channels 0 and 3, four compare match/input capture dual-function interrupts and one overflow interrupt can be requested independently
  - For channels 1, 2, 4, and 5, two compare match/input capture dual-function interrupts, one overflow interrupt, and one underflow interrupt can be requested independently
- Automatic transfer of register data
  - Block transfer, 1-word transfer, and 1-byte transfer possible by DMA controller (DMAC) activation
- A/D converter conversion start trigger can be generated
  - Channel 0 to 5 compare match A/input capture A signal can be used as A/D converter conversion start trigger

Table 10.1 lists the functions of the TPU.

Table 10.1 TPU Functions

Item		Channel 0	Channel 1	Channel 2	Channel 3	Channel 4	Channel 5
Count clock		Pφ/1	Pφ/1	Pφ/1	Pφ/1	Pφ/1	Pφ/1
		Pφ/4	Pφ/4	Pφ/4	Pφ/4	Pφ/4	Pφ/4
		Pφ/16	Pφ/16	Pφ/16	Pφ/16	Pφ/16	Pφ/16
		Pφ/64	Pφ/64	Pφ/64	Pφ/64	Pφ/64	Pφ/64
		TCLKA	Pφ/256	Pφ/1024	Pφ/256	Pφ/1024	Pφ/256
		TCLKB	TCLKA	TCLKA	Pφ/1024	TCLKA	TCLKA
		TCLKC	TCLKB	TCLKB	Pφ/4096	TCLKC	TCLKC
		TCLKD		TCLKC	TCLKA		TCLKD
General registers		TGR0A	TGR1A	TGR2A	TGR3A	TGR4A	TGR5A
		TGR0B	TGR1B	TGR2B	TGR3B	TGR4B	TGR5B
General registers/ buffer registers		TGR0C	—	—	TGR3C	—	—
		TGR0D			TGR3D		
I/O pins		TIOC0A	TIOC1A	TIOC2A	TIOC3A	TIOC4A	TIOC5A
		TIOC0B	TIOC1B	TIOC2B	TIOC3B	TIOC4B	TIOC5B
		TIOC0C			TIOC3C		
		TIOC0D			TIOC3D		
Counter clear function		TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture
Compare match output	0 output	O	O	O	O	O	O
	1 output	O	O	O	O	O	O
	Toggle output	O	O	O	O	O	O
Input capture function		O	O	O	O	O	O
Synchronous operation		O	O	O	O	O	O
PWM mode		O	O	O	O	O	O
Phase counting mode		—	O	O	—	O	O
Buffer operation		O	—	—	O	—	—

Item	Channel 0	Channel 1	Channel 2	Channel 3	Channel 4	Channel 5
DMAC activation	TGR0A compare match or input capture	TGR1A compare match or input capture	TGR2A compare match or input capture	TGR3A compare match or input capture	TGR4A compare match or input capture	TGR5A compare match or input capture
A/D conversion start trigger	TGR0A compare match or input capture	TGR1A compare match or input capture	TGR2A compare match or input capture	TGR3A compare match or input capture	TGR4A compare match or input capture	TGR5A compare match or input capture
Interrupt sources	5 sources • Compare match/ input capture 0A • Compare match/ input capture 0B • Compare match/ input capture 0C • Compare match/ input capture 0D • Overflow	4 sources • Compare match/ input capture 1A • Compare match/ input capture 1B • Overflow • Underflow	4 sources • Compare match/ input capture 2A • Compare match/ input capture 2B • Overflow • Underflow	5 sources • Compare match/ input capture 3A • Compare match/ input capture 3B • Compare match/ input capture 3C • Compare match/ input capture 3D • Overflow	4 sources • Compare match/ input capture 4A • Compare match/ input capture 4B • Overflow • Underflow	4 sources • Compare match/ input capture 5A • Compare match/ input capture 5B • Overflow • Underflow

Legend:

O: Possible

—: Not possible

10.1.2 Block Diagram

Figure 10.1 shows a block diagram of the TPU.

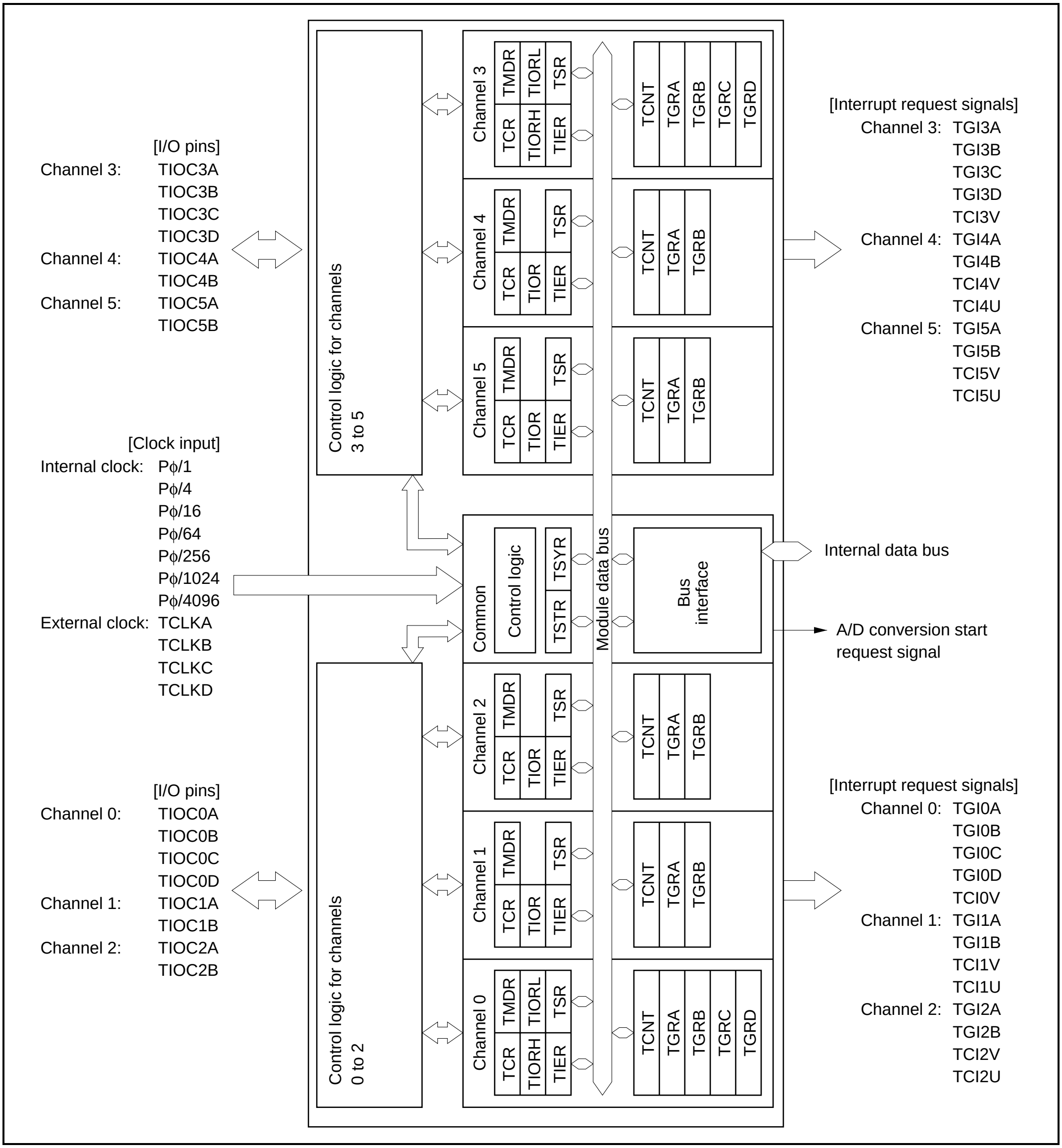


Figure 10.1 Block Diagram of TPU

10.1.3 Pin Configuration

Table 10.2 shows the pin configuration of the TPU.

Table 10.2 TPU Pins

Channel	Name	Abbre- viation	I/O	Function
All	Clock input A	TCLKA	Input	External clock A input pin (Channel 1 and 5 phase counting mode A phase input)
	Clock input B	TCLKB	Input	External clock B input pin (Channel 1 and 5 phase counting mode B phase input)
	Clock input C	TCLKC	Input	External clock C input pin (Channel 2 and 4 phase counting mode A phase input)
	Clock input D	TCLKD	Input	External clock D input pin (Channel 2 and 4 phase counting mode B phase input)
0	Input capture/out compare match 0A	TIOC0A	I/O	TGR0A input capture input/output compare output/PWM output pin
	Input capture/out compare match 0B	TIOC0B	I/O	TGR0B input capture input/output compare output/PWM output pin
	Input capture/out compare match 0C	TIOC0C	I/O	TGR0C input capture input/output compare output/PWM output pin
	Input capture/out compare match 0D	TIOC0D	I/O	TGR0D input capture input/output compare output/PWM output pin
1	Input capture/out compare match 1A	TIOC1A	I/O	TGR1A input capture input/output compare output/PWM output pin
	Input capture/out compare match 1B	TIOC1B	I/O	TGR1B input capture input/output compare output/PWM output pin
2	Input capture/out compare match 2A	TIOC2A	I/O	TGR2A input capture input/output compare output/PWM output pin
	Input capture/out compare match 2B	TIOC2B	I/O	TGR2B input capture input/output compare output/PWM output pin
3	Input capture/out compare match 3A	TIOCA3	I/O	TGR3A input capture input/output compare output/PWM output pin
	Input capture/out compare match 3B	TIOC3B	I/O	TGR3B input capture input/output compare output/PWM output pin
	Input capture/out compare match 3C	TIOC3C	I/O	TGR3C input capture input/output compare output/PWM output pin
	Input capture/out compare match 3D	TIOC3D	I/O	TGR3D input capture input/output compare output/PWM output pin

Channel	Name	Abbre- viation	I/O	Function
4	Input capture/out compare match 4A	TIOC4A	I/O	TGR4A input capture input/output compare output/PWM output pin
	Input capture/out compare match 4B	TIOC4B	I/O	TGR4B input capture input/output compare output/PWM output pin
5	Input capture/out compare match 5A	TIOC5A	I/O	TGR5A input capture input/output compare output/PWM output pin
	Input capture/out compare match 5B	TIOC5B	I/O	TGR5B input capture input/output compare output/PWM output pin

10.1.4 Register Configuration

Table 10.3 summarizes the TPU registers.

Table 10.3 TPU Registers

Channel	Name	Abbre- viation	R/W	Initial Value	Address	Access Size
0	Timer control register 0	TCR0	R/W	H'00	H'FFFF0410	8, 16, 32
	Timer mode register 0	TMDR0	R/W	H'C0	H'FFFF0411	8, 16, 32
	Timer I/O control register 0H	TIOR0H	R/W	H'00	H'FFFF0412	8, 16, 32
	Timer I/O control register 0L	TIOR0L	R/W	H'00	H'FFFF0413	8, 16, 32
	Timer interrupt enable register 0	TIER0	R/W	H'40	H'FFFF0414	8, 16, 32
	Timer status register 0	TSR0	R/(W)*	H'C0	H'FFFF0415	8, 16, 32
	Timer counter 0	TCNT0	R/W	H'0000	H'FFFF0416	16, 32
	Timer general register 0A	TGR0A	R/W	H'FFFF	H'FFFF0418	16, 32
	Timer general register 0B	TGR0B	R/W	H'FFFF	H'FFFF041A	16, 32
	Timer general register 0C	TGR0C	R/W	H'FFFF	H'FFFF041C	16, 32
	Timer general register 0D	TGR0D	R/W	H'FFFF	H'FFFF041E	16, 32
1	Timer control register 1	TCR1	R/W	H'00	H'FFFF0420	8, 16, 32
	Timer mode register 1	TMDR1	R/W	H'C0	H'FFFF0421	8, 16, 32
	Timer I/O control register 1	TIOR1	R/W	H'00	H'FFFF0422	8, 16, 32
	Timer interrupt enable register 1	TIER1	R/W	H'40	H'FFFF0424	8, 16, 32
	Timer status register 1	TSR1	R/(W)*	H'C0	H'FFFF0425	8, 16, 32
	Timer counter 1	TCNT1	R/W	H'0000	H'FFFF0426	16, 32
	Timer general register 1A	TGR1A	R/W	H'FFFF	H'FFFF0428	16, 32
	Timer general register 1B	TGR1B	R/W	H'FFFF	H'FFFF042A	16, 32
2	Timer control register 2	TCR2	R/W	H'00	H'FFFF0430	8, 16, 32
	Timer mode register 2	TMDR2	R/W	H'C0	H'FFFF0431	8, 16, 32
	Timer I/O control register 2	TIOR2	R/W	H'00	H'FFFF0432	8, 16, 32
	Timer interrupt enable register 2	TIER2	R/W	H'40	H'FFFF0434	8, 16, 32
	Timer status register 2	TSR2	R/(W)*	H'C0	H'FFFF0435	8, 16, 32
	Timer counter 2	TCNT2	R/W	H'0000	H'FFFF0436	16, 32
	Timer general register 2A	TGR2A	R/W	H'FFFF	H'FFFF0438	16, 32
	Timer general register 2B	TGR2B	R/W	H'FFFF	H'FFFF043A	16, 32

Channel	Name	Abbre-viation	R/W	Initial Value	Address	Access Size
3	Timer control register 3	TCR3	R/W	H'00	H'FFFF0440	8, 16, 32
	Timer mode register 3	TMDR3	R/W	H'C0	H'FFFF0441	8, 16, 32
	Timer I/O control register 3H	TIOR3H	R/W	H'00	H'FFFF0442	8, 16, 32
	Timer I/O control register 3L	TIOR3L	R/W	H'00	H'FFFF0443	8, 16, 32
	Timer interrupt enable register 3	TIER3	R/W	H'40	H'FFFF0444	8, 16, 32
	Timer status register 3	TSR3	R/(W)*	H'C0	H'FFFF0445	8, 16, 32
	Timer counter 3	TCNT3	R/W	H'0000	H'FFFF0446	16, 32
	Timer general register 3A	TGR3A	R/W	H'FFFF	H'FFFF0448	16, 32
	Timer general register 3B	TGR3B	R/W	H'FFFF	H'FFFF044A	16, 32
	Timer general register 3C	TGR3C	R/W	H'FFFF	H'FFFF044C	16, 32
	Timer general register 3D	TGR3D	R/W	H'FFFF	H'FFFF044E	16, 32
4	Timer control register 4	TCR4	R/W	H'00	H'FFFF0450	8, 16, 32
	Timer mode register 4	TMDR4	R/W	H'C0	H'FFFF0451	8, 16, 32
	Timer I/O control register 4	TIOR4	R/W	H'00	H'FFFF0452	8, 16, 32
	Timer interrupt enable register 4	TIER4	R/W	H'40	H'FFFF0454	8, 16, 32
	Timer status register 4	TSR4	R/(W)*	H'C0	H'FFFF0455	8, 16, 32
	Timer counter 4	TCNT4	R/W	H'0000	H'FFFF0456	16, 32
	Timer general register 4A	TGR4A	R/W	H'FFFF	H'FFFF0458	16, 32
	Timer general register 4B	TGR4B	R/W	H'FFFF	H'FFFF045A	16, 32
5	Timer control register 5	TCR5	R/W	H'00	H'FFFF0460	8, 16, 32
	Timer mode register 5	TMDR5	R/W	H'C0	H'FFFF0461	8, 16, 32
	Timer I/O control register 5	TIOR5	R/W	H'00	H'FFFF0462	8, 16, 32
	Timer interrupt enable register 5	TIER5	R/W	H'40	H'FFFF0464	8, 16, 32
	Timer status register 5	TSR5	R/(W)*	H'C0	H'FFFF0465	8, 16, 32
	Timer counter 5	TCNT5	R/W	H'0000	H'FFFF0466	16, 32
	Timer general register 5A	TGR5A	R/W	H'FFFF	H'FFFF0468	16, 32
	Timer general register 5B	TGR5B	R/W	H'FFFF	H'FFFF046A	16, 32
All	Timer start register	TSTR	R/W	H'00	H'FFFF0400	8, 16, 32
	Timer sync register	TSYR	R/W	H'00	H'FFFF0401	8, 16, 32

Note: * Can only be written with 0 for flag clearing.

10.2 Register Descriptions

10.2.1 Timer Control Registers (TCR)

Channel 0: TCR0
Channel 3: TCR3

Bit:	7	6	5	4	3	2	1	0
	CCLR2	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0
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

Channel 1: TCR1
Channel 2: TCR2
Channel 4: TCR4
Channel 5: TCR5

Bit:	7	6	5	4	3	2	1	0
	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0
Initial value:	0	0	0	0	0	0	0	0
R/W:	—	R/W	R/W	R/W	R/W	R/W	R/W	R/W

The TCR registers are 8-bit registers that control the TCNT channels. The TPU has six TCR registers, one for each of channels 0 to 5. The TCR registers are initialized to H'00 by a reset, and in hardware standby mode and software standby mode. They are not initialized by the module standby function.

TCR settings should only be made when TCNT operation is halted.

Bits 7, 6, and 5—Counter Clear 2 to 0 (CCLR2 to CCLR0): These bits select the TCNT counter clearing source.

Channel	Bit 7: CCLR2	Bit 6: CCLR1	Bit 5: CCLR0	Description
0, 3	0	0	0	TCNT clearing disabled (Initial value)
			1	TCNT cleared by TGRA compare match/input capture
		1	0	TCNT cleared by TGRB compare match/input capture
			1	TCNT cleared by counter clearing for another channel performing synchronous clearing/synchronous operation*1
	1	0	0	TCNT clearing disabled
			1	TCNT cleared by TGRC compare match/input capture*2
		1	0	TCNT cleared by TGRD compare match/input capture*2
			1	TCNT cleared by counter clearing for another channel performing synchronous clearing/synchronous operation*1

Channel	Bit 7: Reserved*3	Bit 6: CCLR1	Bit 5: CCLR0	Description
1, 2, 4, 5	0	0	0	TCNT clearing disabled (Initial value)
			1	TCNT cleared by TGRA compare match/input capture
		1	0	TCNT cleared by TGRB compare match/input capture
			1	TCNT cleared by counter clearing for another channel performing synchronous clearing/synchronous operation*1

Notes: 1. Synchronous operation setting is performed by setting the SYNC bit in TSYR to 1.

2. When TGRC or TGRD is used as a buffer register, TCNT is not cleared because the buffer register setting has priority, and compare match/input capture does not occur.

3. Bit 7 is reserved in channels 1, 2, 4, and 5. It is always read as 0 and cannot be modified.

Bits 4 and 3—Clock Edge 1 and 0 (CKEG1, CKEG0): These bits select the input clock edge. When the internal clock is counted using both edges, the input clock period is halved (e.g. Pϕ/4 both edges = Pϕ/2 rising edge). If phase counting mode is used on channels 1, 2, 4, and 5, this setting is ignored and the phase counting mode setting has priority.

Bit 4: CKEG1	Bit 3: CKEG0	Description
0	0	Count at rising edge (Initial value)
	1	Count at falling edge
1	—	Count at both edges

Note: Internal clock edge selection is valid when the input clock is Pϕ/4 or slower. This setting is ignored if the input clock is Pϕ/1, or when overflow/underflow of another channel is selected.

Bits 2, 1, and 0—Time Prescaler 2 to 0 (TPSC2 to TPSC0): These bits select the TCNT counter clock. The clock source can be selected independently for each channel. Table 10.4 shows the clock sources that can be set for each channel.

Table 10.4 TPU Clock Sources

Channel	Internal Clock							External Clock				Overflow/ Underflow on Another Channel
	Pϕ/1	Pϕ/4	Pϕ/ 16	Pϕ/ 64	Pϕ/ 256	Pϕ/ 1024	Pϕ/ 4096	TCLKA	TCLKB	TCLKC	TCLKD	
0	O	O	O	O				O	O	O	O	
1	O	O	O	O	O			O	O			O
2	O	O	O	O		O		O	O	O		
3	O	O	O	O	O	O	O	O				
4	O	O	O	O		O		O		O		O
5	O	O	O	O	O			O		O	O	

Legend:

O: Setting available

Blank: No setting

Channel	Bit 2: TPSC2	Bit 1: TPSC1	Bit 0: TPSC0	Description
0	0	0	0	Internal clock: Counts on Pϕ/1 (Initial value)
			1	Internal clock: Counts on Pϕ/4
		1	0	Internal clock: Counts on Pϕ/16
			1	Internal clock: Counts on Pϕ/64
	1	0	0	External clock: Counts on TCLKA pin input
			1	External clock: Counts on TCLKB pin input
		1	0	External clock: Counts on TCLKC pin input
			1	External clock: Counts on TCLKD pin input

Channel	Bit 2: TPSC2	Bit 1: TPSC1	Bit 0: TPSC0	Description
1	0	0	0	Internal clock: Counts on Pϕ/1 (Initial value)
			1	Internal clock: Counts on Pϕ/4
		1	0	Internal clock: Counts on Pϕ/16
			1	Internal clock: Counts on Pϕ/64
	1	0	0	External clock: Counts on TCLKA pin input
			1	External clock: Counts on TCLKB pin input
		1	0	Internal clock: Counts on Pϕ/256
			1	Counts on TCNT2 overflow/underflow

Note: This setting is invalid when channel 1 is in phase counting mode.

Channel	Bit 2: TPSC2	Bit 1: TPSC1	Bit 0: TPSC0	Description
2	0	0	0	Internal clock: Counts on Pϕ/1 (Initial value)
			1	Internal clock: Counts on Pϕ/4
		1	0	Internal clock: Counts on Pϕ/16
			1	Internal clock: Counts on Pϕ/64
	1	0	0	External clock: Counts on TCLKA pin input
			1	External clock: Counts on TCLKB pin input
		1	0	External clock: Counts on TCLKC pin input
			1	Internal clock: Counts on Pϕ/1024

Note: This setting is invalid when channel 2 is in phase counting mode.

Channel	Bit 2: TPSC2	Bit 1: TPSC1	Bit 0: TPSC0	Description
3	0	0	0	Internal clock: Counts on Pϕ/1 (Initial value)
			1	Internal clock: Counts on Pϕ/4
		1	0	Internal clock: Counts on Pϕ/16
			1	Internal clock: Counts on Pϕ/64
	1	0	0	External clock: Counts on TCLKA pin input
			1	Internal clock: Counts on Pϕ/1024
		1	0	Internal clock: Counts on Pϕ/256
			1	Internal clock: Counts on Pϕ/4096

Channel	Bit 2: TPSC2	Bit 1: TPSC1	Bit 0: TPSC0	Description
4	0	0	0	Internal clock: Counts on Pϕ/1 (Initial value)
			1	Internal clock: Counts on Pϕ/4
		1	0	Internal clock: Counts on Pϕ/16
			1	Internal clock: Counts on Pϕ/64
	1	0	0	External clock: Counts on TCLKA pin input
			1	External clock: Counts on TCLKC pin input
		1	0	Internal clock: Counts on Pϕ/1024
			1	Counts on TCNT5 overflow/underflow

Note: This setting is invalid when channel 4 is in phase counting mode.

Channel	Bit 2: TPSC2	Bit 1: TPSC1	Bit 0: TPSC0	Description
5	0	0	0	Internal clock: Counts on Pϕ/1 (Initial value)
			1	Internal clock: Counts on Pϕ/4
		1	0	Internal clock: Counts on Pϕ/16
			1	Internal clock: Counts on Pϕ/64
	1	0	0	External clock: Counts on TCLKA pin input
			1	External clock: Counts on TCLKC pin input
		1	0	Internal clock: Counts on Pϕ/256
			1	External clock: Counts on TCLKD pin input

Note: This setting is invalid when channel 5 is in phase counting mode.

10.2.2 Timer Mode Registers (TMDR)

Channel 0: TMDR0
Channel 3: TMDR3

Bit:	7	6	5	4	3	2	1	0
	—	—	BFB	BFA	MD3	MD2	MD1	MD0
Initial value:	1	1	0	0	0	0	0	0
R/W:	—	—	R/W	R/W	R/W	R/W	R/W	R/W

Channel 1: TMDR1
Channel 2: TMDR2
Channel 4: TMDR4
Channel 5: TMDR5

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	MD3	MD2	MD1	MD0
Initial value:	1	1	0	0	0	0	0	0
R/W:	—	—	—	—	R/W	R/W	R/W	R/W

The TMDR registers are 8-bit readable/writable registers that are used to set the operating mode for each channel. The TPU has six TMDR registers, one for each channel. The TMDR registers are initialized to H'C0 by a reset, and in hardware standby mode and software standby mode. They are not initialized by the module standby function.

TMDR settings should only be made when TCNT operation is halted.

Bits 7 and 6—Reserved: These bits are always read as 1 and cannot be modified.

Bit 5—Buffer Operation B (BFB): Specifies whether TGRB is to operate in the normal way, or TGRB and TGRD are to be used together for buffer operation. When TGRD is used as a buffer register, TGRD input capture/output compare is not generated.

In channels 1, 2, 4, and 5, which have no TGRD, bit 5 is reserved. It is always read as 0 and cannot be modified.

Bit 5: BFB	Description	
0	TGRB operates normally	(Initial value)
1	TGRB and TGRD are used together for buffer operation	

Bit 4—Buffer Operation A (BFA): Specifies whether TGRA is to operate in the normal way, or TGRA and TGRC are to be used together for buffer operation. When TGRC is used as a buffer register, TGRC input capture/output compare is not generated.

In channels 1, 2, 4, and 5, which have no TGRC, bit 4 is reserved. It is always read as 0 and cannot be modified.

Bit 4: BFA	Description
0	TGRA operates normally (Initial value)
1	TGRA and TGRC are used together for buffer operation

Bit 3 to 0—Mode 3 to 0 (MD3 to MD0): These bits are used to set the timer operating mode.

Bit 3: MD3[*1]	Bit 2: MD2[*2]	Bit 1: MD1	Bit 0: MD0	Description
0	0	0	0	Normal operation (Initial value)
			1	Reserved
		1	0	PWM mode 1
			1	PWM mode 2
	1	0	0	Phase counting mode 1
			1	Phase counting mode 2
		1	0	Phase counting mode 3
			1	Phase counting mode 4
1	*	*	*	—

Legend:

*: Don’t care

Notes: 1. MD3 is a reserved bit. In a write, it should always be written with 0.

2. Phase counting mode cannot be set for channels 0 and 3. In these channels, 0 should always be written to MD2.

10.2.3 Timer I/O Control Registers (TIOR)

Channel 0: TIOR0H
Channel 1: TIOR1
Channel 2: TIOR2
Channel 3: TIOR3H
Channel 4: TIOR4
Channel 5: TIOR5

Bit:	7	6	5	4	3	2	1	0
	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0
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

Channel 0: TIOR0L
Channel 3: TIOR3L

Bit:	7	6	5	4	3	2	1	0
	IOD3	IOD2	IOD1	IOD0	IOC3	IOC2	IOC1	IOC0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: When TGRC or TGRD is designated for buffer operation, this setting is invalid and the register operates as a buffer register.

The TIOR registers are 8-bit registers that control the TGR registers. The TPU has eight TIOR registers, two each for channels 0 and 3, and one each for channels 1, 2, 4, and 5. The TIOR registers are initialized to H'00 by a reset, and in hardware standby mode and software standby mode. They are not initialized by the module standby function.

Note that the TIOR registers are affected by the TMDR settings.

The initial output specified by TIOR becomes valid when the counter is halted (the CST bit is cleared to 0 in TSTR). In PWM mode 2, the output at the point at which the counter is cleared to 0 is specified.

Bits 7 to 4— I/O Control B3 to B0 (IOB3 to IOB0)
I/O Control D3 to D0 (IOD3 to IOD0):

IOB3 to IOB0 specify the function of TGRB.

IOD3 to IOD0 specify the function of TGRD.

Channel	Bit 7: IOB3	Bit 6: IOB2	Bit 5: IOB1	Bit 4: IOB0	Description		
0	0	0	0	0	TGR0B is output compare register	Output disabled	(Initial value)
				1		Initial output is 0 output	0 output at compare match
			1	0			1 output at compare match
				1			Toggle output at compare match
		1	0	0		Output disabled	
				1		Initial output is 1 output	0 output at compare match
			1	0			1 output at compare match
				1			Toggle output at compare match
	1	0	0	0	TGR0B is input capture register	Capture input source is TIOC0B pin	Input capture at rising edge
				1			Input capture at falling edge
			1	*			Input capture at both edges
		1	*	*		Capture input source is channel 1/count clock	Input capture at TCNT1 count-up/count-down

Legend:

*: Don’t care

Channel	Bit 7: IOD3	Bit 6: IOD2	Bit 5: IOD1	Bit 4: IOD0	Description		
0	0	0	0	0	TGR0D is output compare register*2	Output disabled	(Initial value)
				1		Initial output is 0 output	0 output at compare match
			1	0			1 output at compare match
				1			Toggle output at compare match
		1	0	0		Output disabled	
				1		Initial output is 1 output	0 output at compare match
			1	0			1 output at compare match
				1			Toggle output at compare match
	1	0	0	0	TGR0D is input capture register*2	Capture input source is TIOC0D pin	Input capture at rising edge
				1			Input capture at falling edge
			1	*			Input capture at both edges
		1	*	*		Capture input source is channel 1/count clock	Input capture at TCNT1 count-up/count-down*1

Legend:

*: Don’t care

Notes: 1. When bits TPSC2 to TPSC0 in TCR1 are set to B'000 and $P\phi/1$ is used as the TCNT1 count clock, this setting is invalid and input capture is not generated.

2. When the BFB bit in TMDR0 is set to 1 and TGR0D is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Channel	Bit 7: IOB3	Bit 6: IOB2	Bit 5: IOB1	Bit 4: IOB0	Description		
1	0	0	0	0	TGR1B is output compare register	Output disabled	(Initial value)
				1		Initial output is 0 output	0 output at compare match
			1	0			1 output at compare match
				1			Toggle output at compare match
		1	0	0		Output disabled	
				1		Initial output is 1 output	0 output at compare match
			1	0			1 output at compare match
				1			Toggle output at compare match
	1	0	0	0	TGR1B is input capture register	Capture input source is TIOC1B pin	Input capture at rising edge
				1			Input capture at falling edge
			1	*			Input capture at both edges
		1	*	*		Capture input source is TGR0C compare match/input capture	Input capture at TGR0C compare match/input capture

Legend:

*: Don’t care

Channel	Bit 7: IOB3	Bit 6: IOB2	Bit 5: IOB1	Bit 4: IOB0	Description		
2	0	0	0	0	TGR2B is output compare register	Output disabled	(Initial value)
				1		Initial output is 0 output	0 output at compare match
			1	0			1 output at compare match
				1			Toggle output at compare match
		1	0	0		Output disabled	
				1		Initial output is 1 output	0 output at compare match
			1	0			1 output at compare match
				1			Toggle output at compare match
	1	*	0	0	TGR2B is input capture register	Capture input source is TIOC2B pin	Input capture at rising edge
				1			Input capture at falling edge
			1	*			Input capture at both edges

Legend:
*: Don’t care

Channel	Bit 7: IOB3	Bit 6: IOB2	Bit 5: IOB1	Bit 4: IOB0	Description		
3	0	0	0	0	TGR3B is output compare register	Output disabled	(Initial value)
				1		Initial output is 0 output	0 output at compare match
			1	0			1 output at compare match
				1			Toggle output at compare match
		1	0	0		Output disabled	
				1		Initial output is 1 output	0 output at compare match
			1	0			1 output at compare match
				1			Toggle output at compare match
	1	0	0	0	TGR3B is input capture register	Capture input source is TIOC3B pin	Input capture at rising edge
				1			Input capture at falling edge
			1	*			Input capture at both edges
		1	*	*		Capture input source is channel 4/count clock	Input capture at TCNT4 count-up/count-down

Legend:

*: Don't care

Channel	Bit 7: IOD3	Bit 6: IOD2	Bit 5: IOD1	Bit 4: IOD0	Description		
3	0	0	0	0	TGR3D is output compare register*2	Output disabled	(Initial value)
				1		Initial output is 0 output	0 output at compare match
			1	0			1 output at compare match
				1			Toggle output at compare match
		1	0	0		Output disabled	
				1		Initial output is 1 output	0 output at compare match
			1	0			1 output at compare match
				1			Toggle output at compare match
	1	0	0	0	TGR3D is input capture register*2	Capture input source is TIOC3D pin	Input capture at rising edge
				1			Input capture at falling edge
			1	*			Input capture at both edges
		1	*	*		Capture input source is channel 4/count clock	Input capture at TCNT4 count-up/count-down*1

Legend:

*: Don't care

Notes: 1. When bits TPSC2 to TPSC0 in TCR4 are set to B'000 and Pϕ/1 is used as the TCNT4 count clock, this setting is invalid and input capture is not generated.

2. When the BFB bit in TMDR3 is set to 1 and TGR3D is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Channel	Bit 7: IOB3	Bit 6: IOB2	Bit 5: IOB1	Bit 4: IOB0	Description		
4	0	0	0	0	TGR4B is output compare register	Output disabled	(Initial value)
				1		Initial output is 0 output	0 output at compare match
			1	0			1 output at compare match
				1			Toggle output at compare match
		1	0	0		Output disabled	
				1		Initial output is 1 output	0 output at compare match
			1	0			1 output at compare match
				1			Toggle output at compare match
	1	0	0	0	TGR4B is input capture register	Capture input source is TIOC4B pin	Input capture at rising edge
				1			Input capture at falling edge
			1	*			Input capture at both edges
		1	*	*		Capture input source is TGR3C compare match/input capture	Input capture at generation of TGR3C compare match/input capture

Legend:

*: Don’t care

Channel	Bit 7: IOB3	Bit 6: IOB2	Bit 5: IOB1	Bit 4: IOB0	Description		
5	0	0	0	0	TGR5B is output compare register	Output disabled	(Initial value)
				1		Initial output is 0 output	0 output at compare match
			1	0			1 output at compare match
				1			Toggle output at compare match
		1	0	0		Output disabled	
				1		Initial output is 1 output	0 output at compare match
			1	0			1 output at compare match
				1			Toggle output at compare match
	1	*	0	0	TGR5B is input capture register	Capture input source is TIOC5B pin	Input capture at rising edge
				1			Input capture at falling edge
			1	*			Input capture at both edges

Legend:

*: Don’t care

Bits 3 to 0— I/O Control A3 to A0 (IOA3 to IOA0) I/O Control C3 to C0 (IOC3 to IOC0):

IOA3 to IOA0 specify the function of TGRA.

IOC3 to IOC0 specify the function of TGRC.

Channel	Bit 3: IOA3	Bit 2: IOA2	Bit 1: IOA1	Bit 0: IOA0	Description		
0	0	0	0	0	TGR0A is output compare register	Output disabled	(Initial value)
				1		Initial output is 0 output	0 output at compare match
			1	0			1 output at compare match
				1			Toggle output at compare match
		1	0	0		Output disabled	
				1		Initial output is 1 output	0 output at compare match
			1	0			1 output at compare match
				1			Toggle output at compare match
	1	0	0	0	TGR0A is input capture register	Capture input source is TIOC0A pin	Input capture at rising edge
				1			Input capture at falling edge
			1	*			Input capture at both edges
		1	*	*		Capture input source is channel 1/count clock	Input capture at TCNT1 count-up/count-down

Legend:

*: Don’t care

Channel	Bit 3: IOC3	Bit 2: IOC2	Bit 1: IOC1	Bit 0: IOC0	Description		
0	0	0	0	0	TGR0C is output compare register[*1]	Output disabled	(Initial value)
				1		Initial output is 0 output	0 output at compare match
			1	0			1 output at compare match
				1			Toggle output at compare match
		1	0	0		Output disabled	
				1		Initial output is 1 output	0 output at compare match
			1	0			1 output at compare match
				1			Toggle output at compare match
	1	0	0	0	TGR0C is input capture register[*1]	Capture input source is TIOC0C pin	Input capture at rising edge
				1			Input capture at falling edge
			1	*			Input capture at both edges
		1	*	*		Capture input source is channel 1/count clock	Input capture at TCNT1 count-up/count-down

Legend:

*: Don't care

Note: 1. When the BFA bit in TMDR0 is set to 1 and TGR0C is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Channel	Bit 3: IOA3	Bit 2: IOA2	Bit 1: IOA1	Bit 0: IOA0	Description		
1	0	0	0	0	TGR1A is output compare register	Output disabled (Initial value)	
				1		Initial output is 0 output	0 output at compare match
			1	0			1 output at compare match
				1			Toggle output at compare match
		1	0	0		Output disabled	
				1		Initial output is 1 output	0 output at compare match
			1	0			1 output at compare match
				1			Toggle output at compare match
	1	0	0	0	TGR1A is input capture register	Capture input source is TIOC1A pin	Input capture at rising edge
				1			Input capture at falling edge
			1	*			Input capture at both edges
		1	*	*		Capture input source is TGR0A compare match/input capture	Input capture at generation of channel 0/TGR0A compare match/input capture

Legend:

*: Don’t care

Channel	Bit 3: IOA3	Bit 2: IOA2	Bit 1: IOA1	Bit 0: IOA0	Description		
2	0	0	0	0	TGR2A is output compare register	Output disabled	(Initial value)
				1		Initial output is 0 output	0 output at compare match
			1	0			1 output at compare match
				1			Toggle output at compare match
		1	0	0		Output disabled	
				1		Initial output is 1 output	0 output at compare match
			1	0			1 output at compare match
				1			Toggle output at compare match
	1	*	0	0	TGR2A is input capture register	Capture input source is TIOC2A pin	Input capture at rising edge
				1			Input capture at falling edge
			1	*			Input capture at both edges

Legend:

*: Don’t care

Channel	Bit 3: IOA3	Bit 2: IOA2	Bit 1: IOA1	Bit 0: IOA0	Description		
3	0	0	0	0	TGR3A is output compare register	Output disabled	(Initial value)
				1		Initial output is 0 output	0 output at compare match
			1	0			1 output at compare match
				1			Toggle output at compare match
		1	0	0		Output disabled	
				1		Initial output is 1 output	0 output at compare match
			1	0			1 output at compare match
				1			Toggle output at compare match
	1	0	0	0	TGR3A is input capture register	Capture input source is TIOC3A pin	Input capture at rising edge
				1			Input capture at falling edge
			1	*			Input capture at both edges
		1	*	*		Capture input source is channel 4/count clock	Input capture at TCNT4 count-up/count-down

Legend:

*: Don’t care

Channel	Bit 3: IOC3	Bit 2: IOC2	Bit 1: IOC1	Bit 0: IOC0	Description		
3	0	0	0	0	TGR3C is output compare register*1	Output disabled	(Initial value)
				1		Initial output is 0 output	0 output at compare match
			1	0			1 output at compare match
				1			Toggle output at compare match
		1	0	0		Output disabled	
				1		Initial output is 1 output	0 output at compare match
			1	0			1 output at compare match
				1			Toggle output at compare match
	1	0	0	0	TGR3C is input capture register*1	Capture input source is TIOC3C pin	Input capture at rising edge
				1			Input capture at falling edge
			1	*			Input capture at both edges
		1	*	*		Capture input source is channel 4/count clock	Input capture at TCNT4 count-up/count-down

Legend:

*: Don’t care

Note: 1. When the BFA bit in TMDR3 is set to 1 and TGR3C is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Channel	Bit 3: IOA3	Bit 2: IOA2	Bit 1: IOA1	Bit 0: IOA0	Description		
4	0	0	0	0	TGR4A is output compare register	Output disabled (Initial value)	
				1		Initial output is 0 output	0 output at compare match
			1	0			1 output at compare match
				1			Toggle output at compare match
		1	0	0		Output disabled	
				1		Initial output is 1 output	0 output at compare match
			1	0			1 output at compare match
				1			Toggle output at compare match
	1	0	0	0	TGR4A is input capture register	Capture input source is TIOC4A pin	Input capture at rising edge
				1			Input capture at rising edge
			1	*			Input capture at both edges
		1	*	*		Capture input source is TGR3A compare match/input capture	Input capture at generation of TGR3A compare match/input capture

Legend:

*: Don’t care

Channel	Bit 3: IOA3	Bit 2: IOA2	Bit 1: IOA1	Bit 0: IOA0	Description		
5	0	0	0	0	TGR5A is output compare register	Output disabled	(Initial value)
				1		Initial output is 0 output	0 output at compare match
			1	0			1 output at compare match
				1			Toggle output at compare match
		1	0	0		Output disabled	
				1		Initial output is 1 output	0 output at compare match
			1	0			1 output at compare match
				1			Toggle output at compare match
	1	*	0	0	TGR5A is input capture register	Capture input source is TIOC5A pin	Input capture at rising edge
				1			Input capture at falling edge
			1	*			Input capture at both edges

Legend:

*: Don’t care

10.2.4 Timer Interrupt Enable Registers (TIER)

Channel 0: TIER0
Channel 3: TIER3

Bit:	7	6	5	4	3	2	1	0
	TTGE	—	—	TCIEV	TGIED	TGIEC	TGIEB	TGIEA
Initial value:	0	1	0	0	0	0	0	0
R/W:	R/W	—	—	R/W	R/W	R/W	R/W	R/W

Channel 1: TIER1
Channel 2: TIER2
Channel 4: TIER4
Channel 5: TIER5

Bit:	7	6	5	4	3	2	1	0
	TTGE	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA
Initial value:	0	1	0	0	0	0	0	0
R/W:	R/W	—	R/W	R/W	—	—	R/W	R/W

The TIER registers are 8-bit registers that control enabling or disabling of interrupt requests for each channel. The TPU has six TIER registers, one for each channel. The TIER registers are initialized to H'40 by a reset, and in hardware standby mode and software standby mode. They are not initialized by the module standby function.

Bit 7—A/D Conversion Start Request Enable (TTGE): Enables or disables generation of A/D conversion start requests by TGRA input capture/compare match.

Bit 7: TTGE	Description	
0	A/D conversion start request generation disabled	(Initial value)
1	A/D conversion start request generation enabled	

Bit 6—Reserved: This bit is always read as 1 and cannot be modified.

Bit 5—Underflow Interrupt Enable (TCIEU): Enables or disables interrupt requests (TCIU) by the TCFU flag in TSR when TCFU is set to 1 in channels 1, 2, 4, and 5.

In channels 0 and 3, bit 5 is reserved. It is always read as 0 and cannot be modified.

Bit 5: TCIEU	Description	
0	Interrupt requests (TCIU) by TCFU disabled	(Initial value)
1	Interrupt requests (TCIU) by TCFU enabled	

Bit 4—Overflow Interrupt Enable (TCIEV): Enables or disables interrupt requests (TCIV) by the TCFV flag in TSR when TCFV is set to 1.

Bit 4: TCIEV	Description	
0	Interrupt requests (TCIV) by TCFV disabled	(Initial value)
1	Interrupt requests (TCIV) by TCFV enabled	

Bit 3—TGR Interrupt Enable D (TGIED): Enables or disables interrupt requests (TGID) by the TGFD bit in TSR when TGFD is set to 1 in channels 0 and 3.

In channels 1, 2, 4, and 5, bit 3 is reserved. It is always read as 0 and cannot be modified.

Bit 3: TGIED	Description	
0	Interrupt requests (TGID) by TGFD bit disabled	(Initial value)
1	Interrupt requests (TGID) by TGFD bit enabled	

Bit 2—TGR Interrupt Enable C (TGIEC): Enables or disables interrupt requests (TGIC) by the TGFC bit in TSR when TGFC is set to 1 in channels 0 and 3.

In channels 1, 2, 4, and 5, bit 2 is reserved. It is always read as 0 and cannot be modified.

Bit 2: TGIEC	Description	
0	Interrupt requests (TGIC) by TGFC bit disabled	(Initial value)
1	Interrupt requests (TGIC) by TGFC bit enabled	

Bit 1—TGR Interrupt Enable B (TGIEB): Enables or disables interrupt requests (TGIB) by the TGFB bit in TSR when TGFB is set to 1.

Bit 1: TGIEB	Description	
0	Interrupt requests (TGIB) by TGFB bit disabled	(Initial value)
1	Interrupt requests (TGIB) by TGFB bit enabled	

Bit 0—TGR Interrupt Enable A (TGIEA): Enables or disables interrupt requests (TGIA) by the TGFA bit in TSR when TGFA is set to 1.

Bit 0: TGIEA	Description	
0	Interrupt requests (TGIA) by TGFA bit disabled	(Initial value)
1	Interrupt requests (TGIA) by TGFA bit enabled	

10.2.5 Timer Status Registers (TSR)

Channel 0: TSR0
Channel 3: TSR3

Bit:	7	6	5	4	3	2	1	0
	—	—	—	TCFV	TGFD	TGFC	TGFB	TGFA
Initial value:	1	1	0	0	0	0	0	0
R/W:	—	—	—	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*

Note: * Can only be written with 0 for flag clearing.

Channel 1: TSR1
Channel 2: TSR2
Channel 4: TSR4
Channel 5: TSR5

Bit:	7	6	5	4	3	2	1	0
	TCFD	—	TCFU	TCFV	—	—	TGFB	TGFA
Initial value:	1	1	0	0	0	0	0	0
R/W:	R	—	R/(W)*	R/(W)*	—	—	R/(W)*	R/(W)*

Note: * Can only be written with 0 for flag clearing.

The TSR registers are 8-bit registers that indicate the status of each channel. The TPU has six TSR registers, one for each channel. The TSR registers are initialized to H'C0 by a reset, and in hardware standby mode and software standby mode. They are not initialized by the module standby function.

Bit 7—Count Direction Flag (TCFD): Status flag that shows the direction in which TCNT counts in channels 1, 2, 4, and 5.

In channels 0 and 3, bit 7 is reserved. It is always read as 1 and cannot be modified.

Bit 7: TCFD	Description	
0	TCNT counts down	
1	TCNT counts up	(Initial value)

Bit 6—Reserved: This bit is always read as 1 and cannot be modified.

Bit 5—Underflow Flag (TCFU): Status flag that indicates that TCNT underflow has occurred in channels 1, 2, 4, and 5.

In channels 0 and 3, bit 5 is reserved. It is always read as 0 and cannot be modified.

Bit 5: TCFU	Description	
0	[Clearing condition] When 0 is written to TCFU after reading TCFU = 1	(Initial value)
1	[Setting condition] When the TCNT value underflows (changes from H'0000 to H'FFFF)	

Bit 4—Overflow Flag (TCFV): Status flag that indicates that TCNT overflow has occurred.

Bit 4: TCFV	Description	
0	[Clearing condition] When 0 is written to TCFV after reading TCFV = 1	(Initial value)
1	[Setting condition] When the TCNT value overflows (changes from H'FFFF to H'0000)	

Bit 3—Input Capture/Output Compare Flag D (TGFD): Status flag that indicates the occurrence of TGRD input capture or compare match in channels 0 and 3.

In channels 1, 2, 4, and 5, bit 3 is reserved. It is always read as 0 and cannot be modified.

Bit 3: TGFD	Description
0	[Clearing condition] (Initial value) When 0 is written to TGFD after reading TGFD = 1
1	[Setting conditions] • When TCNT = TGRD while TGRD is functioning as output compare register • When TCNT value is transferred to TGRD by input capture signal while TGRD is functioning as input capture register

Bit 2—Input Capture/Output Compare Flag C (TGFC): Status flag that indicates the occurrence of TGRC input capture or compare match in channels 0 and 3.

In channels 1, 2, 4, and 5, bit 2 is reserved. It is always read as 0 and cannot be modified.

Bit 2: TGFC	Description
0	[Clearing condition] (Initial value) When 0 is written to TGFC after reading TGFC = 1
1	[Setting conditions] • When TCNT = TGRC while TGRC is functioning as output compare register • When TCNT value is transferred to TGRC by input capture signal while TGRC is functioning as input capture register

Bit 1—Input Capture/Output Compare Flag B (TGFB): Status flag that indicates the occurrence of TGRB input capture or compare match.

Bit 1: TGFB	Description
0	[Clearing condition] (Initial value) When 0 is written to TGFB after reading TGFB = 1
1	[Setting conditions] • When TCNT = TGRB while TGRB is functioning as output compare register • When TCNT value is transferred to TGRB by input capture signal while TGRB is functioning as input capture register

Bit 0—Input Capture/Output Compare Flag A (TGFA): Status flag that indicates the occurrence of TGRA input capture or compare match.

Bit 0: TGFA	Description
0	[Clearing condition] (Initial value) When 0 is written to TGFA after reading TGFA = 1
1	[Setting conditions] • When TCNT = TGRA while TGRA is functioning as output compare register • When TCNT value is transferred to TGRA by input capture signal while TGRA is functioning as input capture register

Note: Cleared by DMAC transfer initiated by TGFA.

10.2.6 Timer Counters (TCNT)

Channel 0: TCNT0 (up-counter)
Channel 1: TCNT1 (up/down-counter*)
Channel 2: TCNT2 (up/down-counter*)
Channel 3: TCNT3 (up-counter)
Channel 4: TCNT4 (up/down-counter*)
Channel 5: TCNT5 (up/down-counter*)

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

Note: * These counters can be used as up/down-counters only in phase counting mode (or when counting overflows/underflows on another channel in phase counting mode). In other cases they function as up-counters.

The TCNT registers are 16-bit counters. The TPU has six TCNT counters, one for each channel.

The TCNT counters are initialized to H'0000 by a reset, and in hardware standby mode and software standby mode. They are not initialized by the module standby function.

The TCNT counters cannot be accessed in 8-bit units; they must always be accessed as a 16-bit unit.

10.2.7 Timer General Registers (TGR)

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 TGR registers are 16-bit registers with a dual function as output compare and input capture registers. The TPU has 16 general registers, four each for channels 0 and 3 and two each for channels 1, 2, 4, and 5. TGRC and TGRD for channels 0 and 3 can also be designated for operation as buffer registers*. The TGR registers are initialized to H'FFFF by a reset, and in hardware standby mode and software standby mode. They are not initialized by the module standby function.

The TGR registers cannot be accessed in 8-bit units; they must always be accessed as a 16-bit unit.

Note: * TGR buffer register combinations are TGRA–TGRC and TGRB–TGRD.

10.2.8 Timer Start Register (TSTR)

Bit:	7	6	5	4	3	2	1	0
	—	—	CST5	CST4	CST3	CST2	CST1	CST0
Initial value:	0	0	0	0	0	0	0	0
R/W:	—	—	R/W	R/W	R/W	R/W	R/W	R/W

TSTR is an 8-bit readable/writable register that selects operation/stoppage for channels 0 to 5. TSTR is initialized to H'00 by a reset, and in hardware standby mode and software standby mode. It is not initialized by the module standby function.

Bits 7 and 6—Reserved: These bits are always read as 0 and cannot be modified.

Bits 5 to 0—Counter Start 5 to 0 (CST5 to CST0): These bits select operation or stoppage of the TCNT counters.

Bit n: CSTn	Description	
0	TCNTn count operation is stopped	(Initial value)
1	TCNTn performs count operation	

Notes: n = 0 to 5

If 0 is written to the CST bit during operation with the TIOC pin designated for output, the counter stops but the TIOC pin output compare output level is retained. If TIOR is written to when the CST bit is cleared to 0, the pin output level will be changed to the set initial output value.

10.2.9 Timer Sync Register (TSYR)

Bit:	7	6	5	4	3	2	1	0
	—	—	SYNC5	SYNC4	SYNC3	SYNC2	SYNC1	SYNC0
Initial value:	0	0	0	0	0	0	0	0
R/W:	—	—	R/W	R/W	R/W	R/W	R/W	R/W

TSYR is an 8-bit readable/writable register that selects independent operation or synchronous operation for the channel 0 to 5 TCNT counters. A channel performs synchronous operation when the corresponding bit in TSYR is set to 1.

TSYR is initialized to H'00 by a reset, and in hardware standby mode and software standby mode. It is not initialized by the module standby function.

Bits 7 and 6—Reserved: These bits must always be written with 0.

Bits 5 to 0—Timer Sync 5 to 0 (SYNC5 to SYNC0): These bits select whether operation is independent of or synchronized with other channels.

When synchronous operation is selected, synchronous presetting*1 of multiple TCNT counters or synchronous clearing*2 by counter clearing on another channel is possible.

Notes: 1. To set synchronous operation, the SYNC bits for at least two channels must be set to 1.

2. To set synchronous clearing, in addition to the SYNC bit, the TCNT clearing source must also be set by means of bits CCLR2 to CCLR0 in TCR.

Bit n: SYNCn	Description
0	TCNTn operates independently (TCNT presetting/clearing is unrelated to other channels) (Initial value)
1	TCNTn performs synchronous operation TCNT synchronous presetting/synchronous clearing is possible

Note: n = 5 to 0

10.3 Interface to Bus Master

10.3.1 16-Bit Registers

TCNT and TGR are 16-bit registers. As the data bus to the bus master is 16 bits wide, these registers can be read and written to in 16-bit units.

These registers cannot be read or written to in 8-bit units; 16-bit access must always be used.

An example of 16-bit register access operation is shown in figure 10.2.

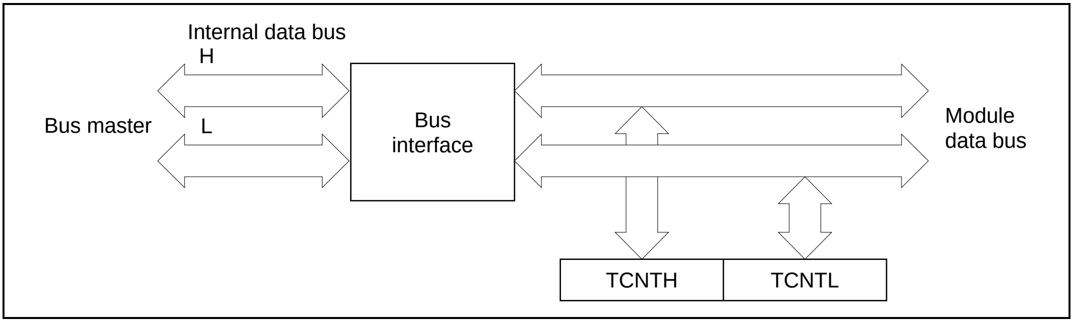


Figure 10.2 16-Bit Register Access Operation (Bus Master ↔ TCNT (16 Bits))

10.3.2 8-Bit Registers

Registers other than TCNT and TGR are 8-bit registers. As the data bus to the CPU is 16 bits wide, these registers can be read and written to in 16-bit units. They can also be written to in 8-bit units.

Examples of 8-bit register access operation are shown in figures 10.3, 10.4, and 10.5.

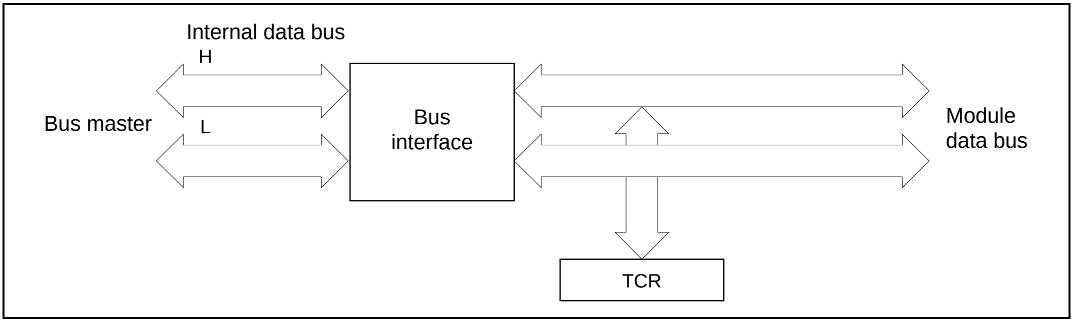


Figure 10.3 8-Bit Register Access Operation (Bus Master ↔ TCR (Upper 8 Bits))

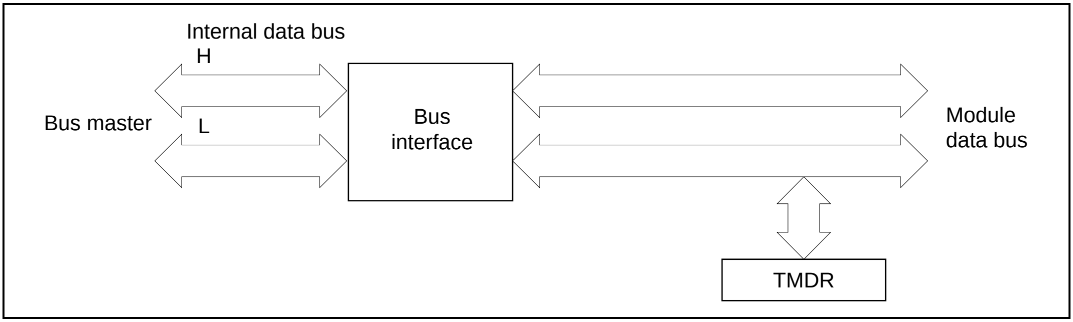


Figure 10.4 8-Bit Register Access Operation (Bus Master ↔ TMDR (Lower 8 Bits))

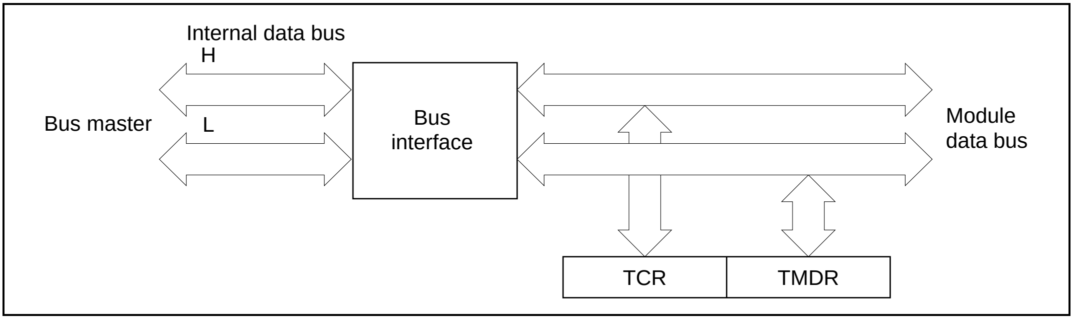


Figure 10.5 8-Bit Register Access Operation (Bus Master ↔ TCR and TMDR (16 Bits))

10.4 Operation

10.4.1 Overview

Operation in each mode is outlined below.

Normal Operation: Each channel has a TCNT and TGR register. TCNT performs up-counting, and is also capable of free-running operation, synchronous counting, and external event counting.

Each TGR can be used as an input capture register or output compare register.

Synchronous Operation: When synchronous operation is designated for a channel, TCNT for that channel performs synchronous presetting. That is, when TCNT for a channel designated for synchronous operation is rewritten, the TCNT counters for the other channels are also rewritten at the same time. Synchronous clearing of the TCNT counters is also possible by setting the timer synchronization bits in TSYR for channels designated for synchronous operation.

Buffer Operation:

- When TGR is an output compare register

  When a compare match occurs, the value in the buffer register for the relevant channel is transferred to TGR.
- When TGR is an input capture register

  When input capture occurs, the value in TCNT is transfer to TGR and the value previously held in TGR is transferred to the buffer register.

Cascaded Operation: The channel 1 counter (TCNT1) and channel 2 counter (TCNT2), or the channel 4 counter (TCNT4) and channel 5 counter (TCNT5), can be connected together to operate as a 32-bit counter.

PWM Mode: In this mode, a PWM waveform is output. The output level can be set in TIOR. A PWM waveform with a duty of between 0% and 100% can be output, according to the setting of each TGR register.

Phase Counting Mode: In this mode, TCNT is incremented or decremented by detecting the phases of two clocks input from the external clock input pins in channels 1, 2, 4, and 5. When phase counting mode is set, the corresponding TCLK pin functions as the clock pin, and TCNT performs up- or down-counting.

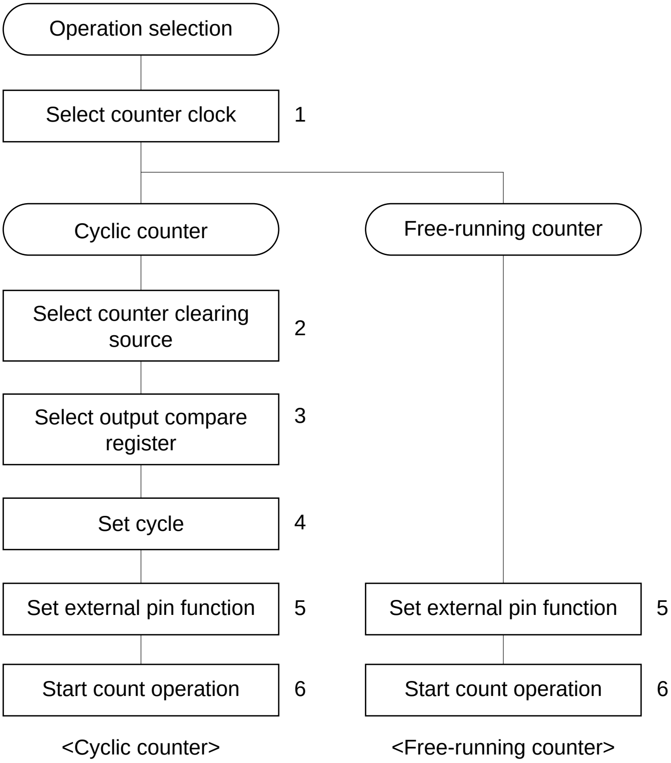
This mode can be used for two-phase encoder pulse input.

10.4.2 Basic Functions

Counter Operation

When one of bits CST0 to CST5 is set to 1 in TSTR, the TCNT counter for the corresponding channel starts counting. TCNT can operate as a free-running counter, a cyclic counter, and so on.

Example of count operation setting procedure: Figure 10.6 shows an example of the count operation setting procedure.



1. Select the counter clock with bits TPSC2 to TPSC0 in TCR. At the same time, select the input clock edge with bits CKEG1 and CKEG0 in TCR.
2. For cyclic counter operation, select the TGR to be used as the TCNT clearing source with bits CCLR2 to CCLR0 in TCR.
3. Designate the TGR selected in 2 as an output compare register by means of TIOR.
4. Set the cyclic counter cycle in the TGR selected in 2.
5. Set the external pin function with the pin function controller (PFC).
6. Set the CST bit in TSTR to 1 to start the count operation.

Figure 10.6 Example of Counter Operation Setting Procedure

Free-running count operation and cyclic count operation: Immediately after a reset, the TPU's TCNT counters are all designated as free-running counters. When the relevant bit in TSTR is set to 1 the corresponding TCNT counter starts up-count operation as a free-running counter. When TCNT overflows (from H'FFFF to H'0000), the TCFV bit in TSR is set to 1. If the value of the corresponding TCIEV bit in TIER is 1 at this point, the TPU requests an interrupt. After overflow, TCNT starts counting up again from H'0000.

Figure 10.7 illustrates free-running counter operation.

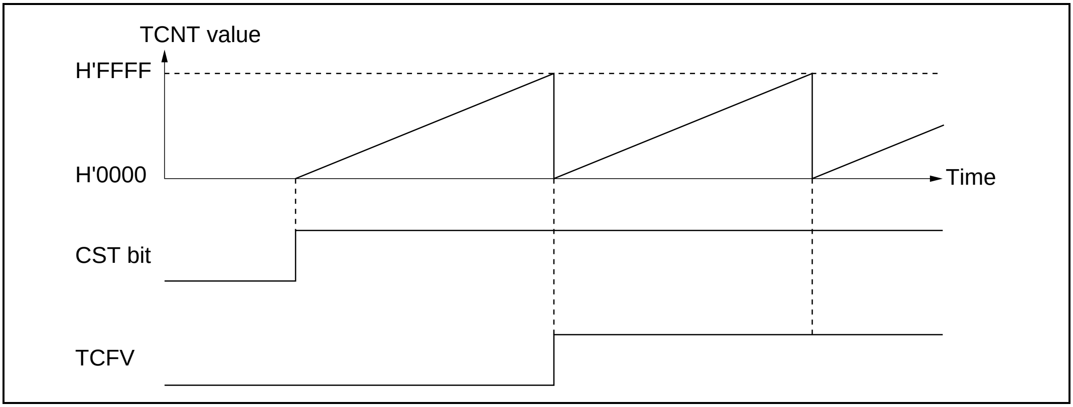


Figure 10.7 Free-Running Counter Operation

When compare match is selected as the TCNT clearing source, the TCNT counter for the relevant channel performs cyclic count operation. The TGR register for setting the cycle is designated as an output compare register, and counter clearing by compare match is selected by means of bits CCLR2 to CCLR0 in TCR. After the settings have been made, TCNT starts up-count operation as a cyclic counter when the corresponding bit in TSTR is set to 1. When the count value matches the value in TGR, the TGF bit in TSR is set to 1 and TCNT is cleared to H'0000.

If the value of the corresponding TGIE bit in TIER is 1 at this point, the TPU requests an interrupt. After a compare match, TCNT starts counting up again from H'0000.

Figure 10.8 illustrates cyclic counter operation.

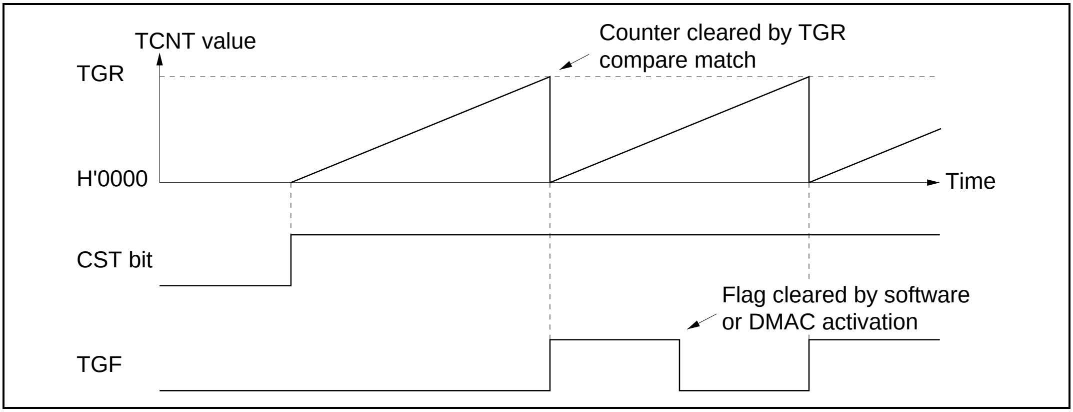
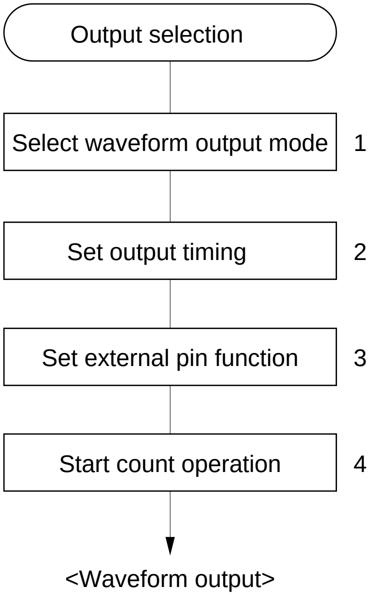

TCNT value
Counter cleared by TGR compare match
TGR
H'0000
Time
CST bit
Flag cleared by software or DMAC activation
TGF

Figure 10.8 Cyclic Counter Operation

Waveform Output by Compare Match

The TPU can perform 0, 1, or toggle output from the corresponding output pin using compare matches.

Example of setting procedure for waveform output by compare match: Figure 10.9 shows an example of the setting procedure for waveform output by compare match.



1. Select initial value 0 output or 1 output and compare match output value 0 output, 1 output, or toggle output by means of TIOR. The set initial value is output at the TIOC pin until the first compare match occurs.
2. Set the timing for compare match generation in TGR.
3. Set the external pin function with the pin function controller (PFC).
4. Set the CST bit in TSTR to 1 to start the count operation.

Figure 10.9 Example of Setting Procedure for Waveform Output by Compare Match

Examples of waveform output operation: Figure 10.10 shows an example of 0 output/1 output.

In this example TCNT has been designated as a free-running counter, and settings have been made so that 1 is output by compare match A, and 0 is output by compare match B. When the set level and the pin level coincide, the pin level does not change.

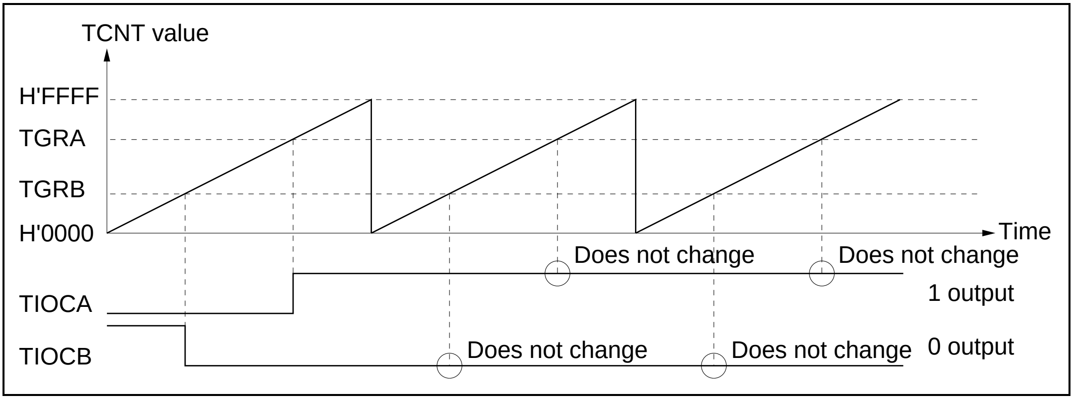


Figure 10.10   Example of 0 Output/1 Output Operation

Figure 10.11 shows an example of toggle output.

In this example TCNT has been designated as a cyclic counter (with counter clearing performed by compare match B), and settings have been made so that output is toggled by both by compare match A and compare match B.

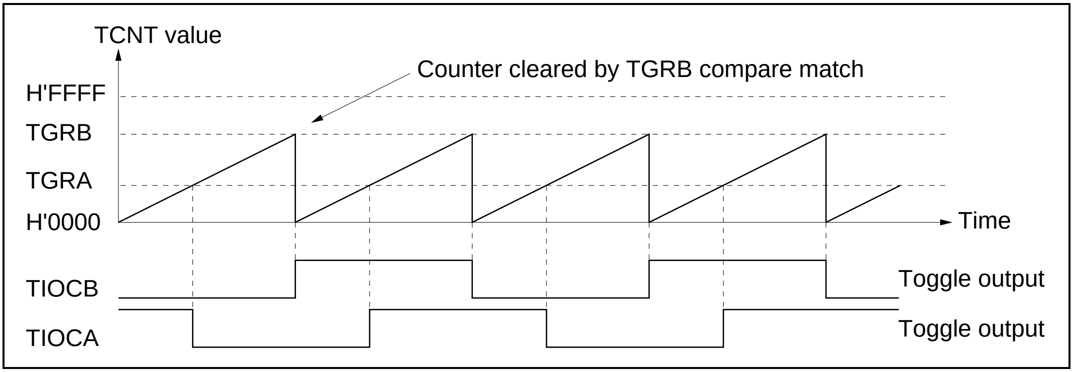


Figure 10.11   Example of Toggle Output Operation

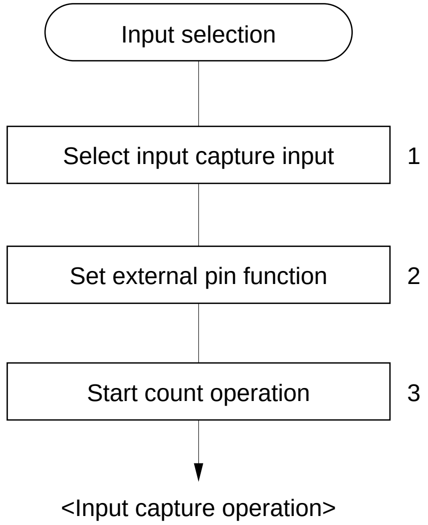
Input Capture Function

The TCNT value can be transferred to TGR on detection of the TIOC pin input edge.

Rising edge, falling edge, or both edges can be selected as the detected edge. For channels 0, 1, 3, and 4, it is also possible to specify another channel's counter input clock or compare match signal as the input capture source.

Note: When another channel's counter input clock is specified as the input capture source in channel 0 or 3, Pϕ/1 must not be selected as the counter input clock used for input capture input. Input capture will not occur if Pϕ/1 is selected.

Example of input capture operation setting procedure: Figure 10.12 shows an example of the input capture operation setting procedure.


Input selection
Select input capture input
1
Set external pin function
2
Start count operation
3
<Input capture operation>

1. Designate TGR as an input capture register by means of TIOR, and select the input signal rising edge, falling edge, or both edges as the input capture source.
2. Set the external pin function with the pin function controller (PFC).
3. Set the CST bit in TSTR to 1 to start the count operation.

Figure 10.12 Example of Input Capture Operation Setting Procedure

Example of input capture operation: Figure 10.13 shows an example of input capture operation.

In this example both rising and falling edges have been selected as the TIOCA pin input capture input edge, falling edge has been selected as the TIOCB pin input capture input edge, and counter clearing by TGRB input capture has been designated for TCNT.

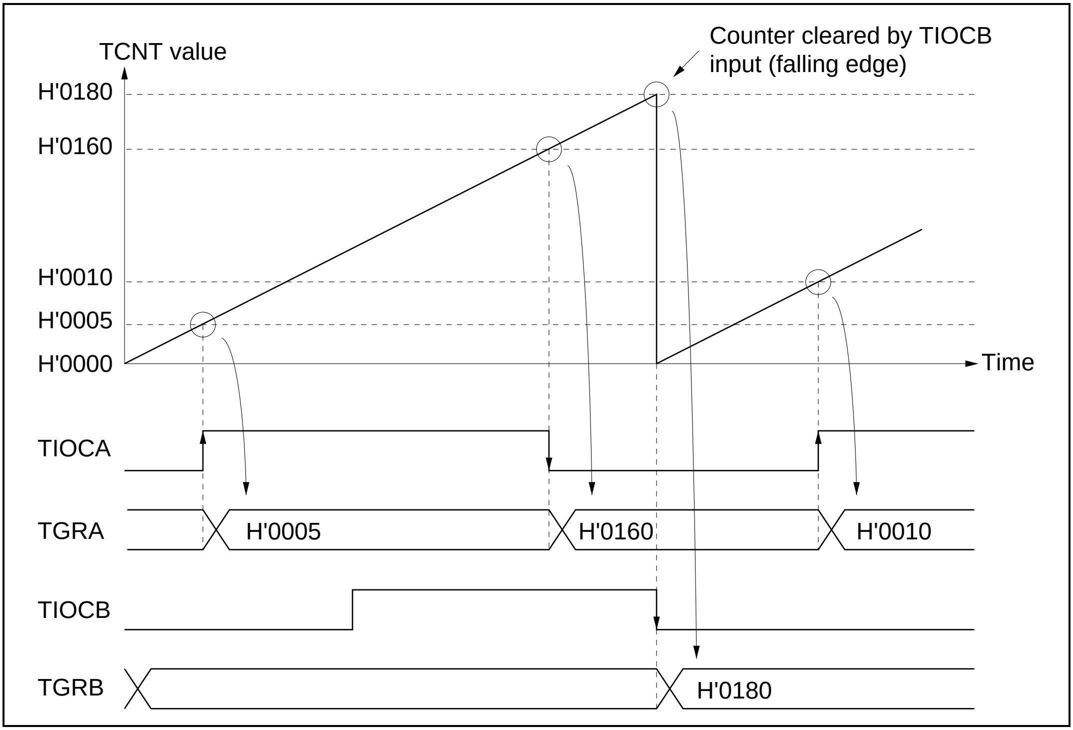


Figure 10.13 Example of Input Capture Operation

10.4.3 Synchronous Operation

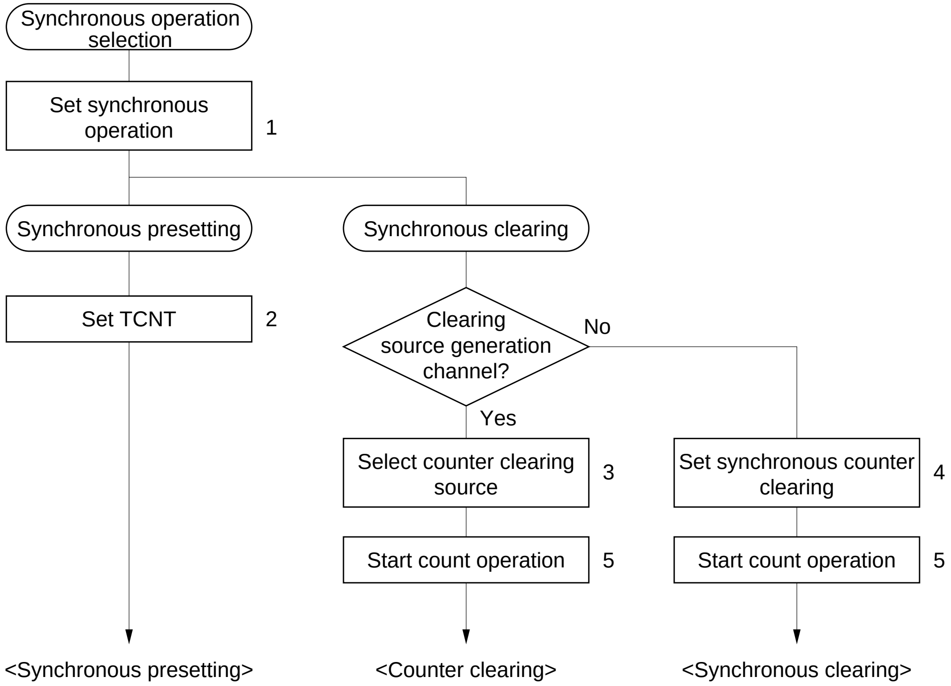
In synchronous operation, the values in a number of TCNT counters can be rewritten simultaneously (synchronous presetting). Also, a number of TCNT counters can be cleared simultaneously by making the appropriate setting in TCR (synchronous clearing).

Synchronous operation enables TGR to be incremented with respect to a single time base.

Channels 0 to 5 can all be designated for synchronous operation.

Example of Synchronous Operation Setting Procedure

Figure 10.14 shows an example of the synchronous operation setting procedure.



1. Set to 1 the SYNC bits in TSYR corresponding to the channels to be designated for synchronous operation.
2. When the TCNT counter of any of the channels designated for synchronous operation is written to, the same value is simultaneously written to the other TCNT counters.
3. Use bits CCLR2 to CCLR0 in TCR to specify TCNT clearing by input capture/output compare, etc.
4. Use bits CCLR2 to CCLR0 in TCR to designate synchronous clearing for the counter clearing source.
5. Set to 1 the CST bits in TSTR for the relevant channels, to start the count operation.

Figure 10.14 Example of Synchronous Operation Setting Procedure

Example of Synchronous Operation

Figure 10.15 shows an example of synchronous operation.

In this example, synchronous operation and PWM mode 1 have been designated for channels 0 to 2, TGR0B compare match has been set as the channel 0 counter clearing source, and synchronous clearing has been set for the channel 1 and 2 counter clearing source.

Three-phase PWM waveforms are output from pins TIOC0A, TIOC1A, and TIOC2A. Synchronous presetting and synchronous clearing by TGR0B compare match is performed for the channel 0 to 2 TCNT counters, and the data set in TGR0B is the PWM cycle.

For details of PWM modes, see section 10.4.6, PWM Modes.

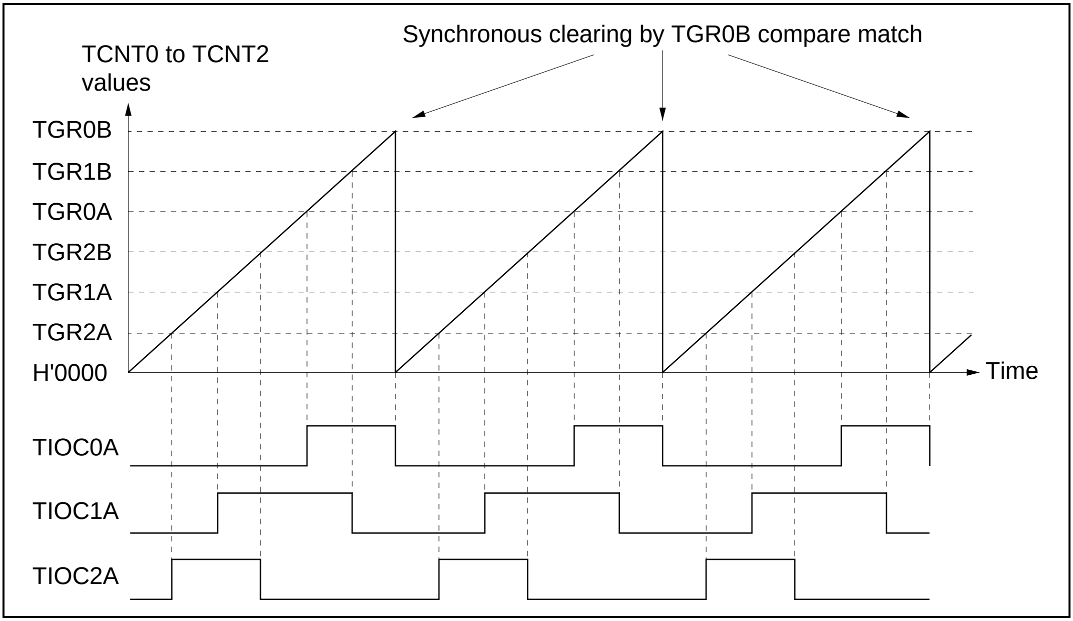


Figure 10.15 Example of Synchronous Operation

10.4.4 Buffer Operation

Buffer operation, provided for channels 0 and 3, enables TGRC and TGRD to be used as buffer registers.

Buffer operation differs depending on whether TGR has been designated as an input capture register or as a compare match register.

Table 10.5 shows the register combinations used in buffer operation.

Table 10.5 Register Combinations in Buffer Operation

Channel	Timer General Register	Buffer Register
0	TGR0A	TGR0C
	TGR0B	TGR0D
3	TGR3A	TGR3C
	TGR3B	TGR3D

- When TGR is an output compare register

  When a compare match occurs, the value in the buffer register for the corresponding channel is transferred to the timer general register.

  This operation is illustrated in figure 10.16.

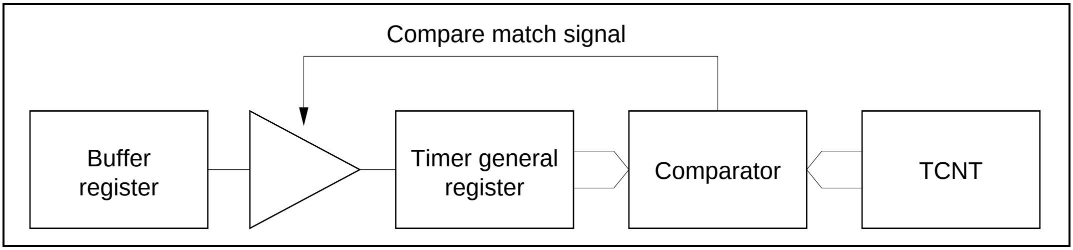


Figure 10.16 Compare Match Buffer Operation

- When TGR is an input capture register

  When input capture occurs, the value in TCNT is transferred to TGR and the value previously held in the timer general register is transferred to the buffer register.

  This operation is illustrated in figure 10.17.

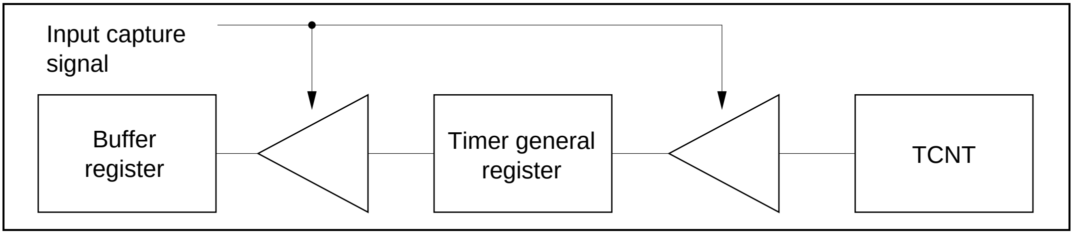
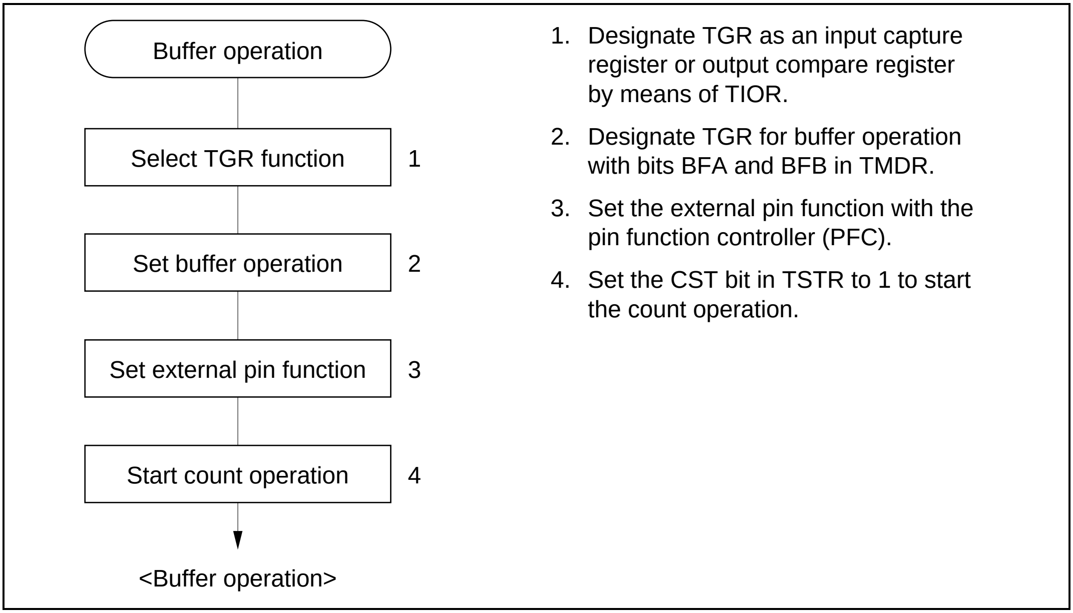


Figure 10.17 Input Capture Buffer Operation

Example of Buffer Operation Setting Procedure

Figure 10.18 shows an example of the buffer operation setting procedure.



1. Designate TGR as an input capture register or output compare register by means of TIOR.
2. Designate TGR for buffer operation with bits BFA and BFB in TMDR.
3. Set the external pin function with the pin function controller (PFC).
4. Set the CST bit in TSTR to 1 to start the count operation.

Figure 10.18 Example of Buffer Operation Setting Procedure

Examples of Buffer Operation

When TGR is an output compare register: Figure 10.19 shows an operation example in which PWM mode 1 has been designated for channel 0, and buffer operation has been designated for TGRA and TGRC. The settings used in this example are TCNT clearing by compare match B, 1 output at compare match A, and 0 output at compare match B.

As buffer operation has been set, when compare match A occurs the output changes and the value in buffer register TGRC is simultaneously transferred to timer general register TGRA. This operation is repeated each time compare match A occurs.

For details of PWM modes, see section 10.4.6, PWM Modes.

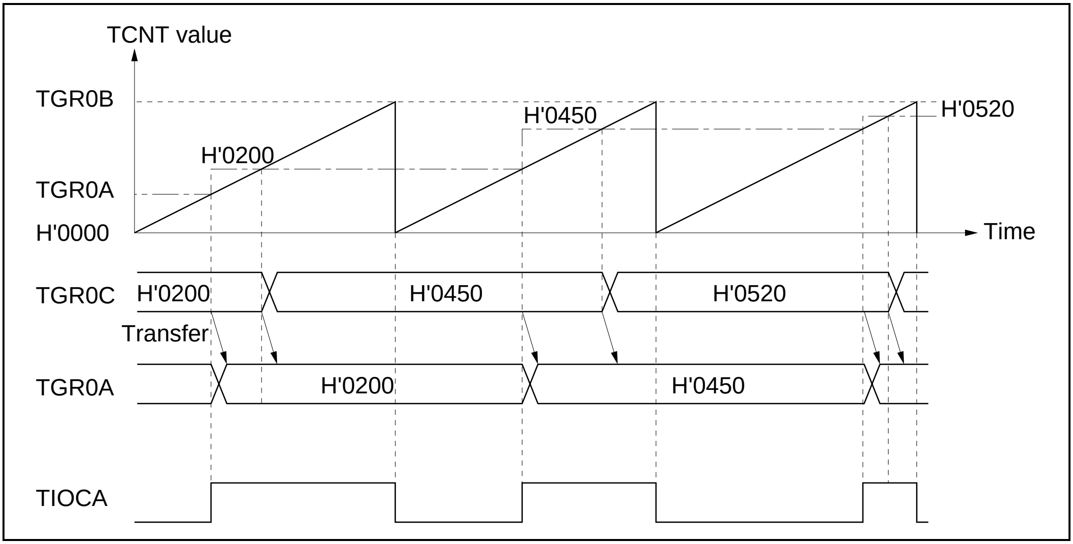


Figure 10.19 Example of Buffer Operation (1)

When TGR is an input capture register: Figure 10.20 shows an operation example in which TGRA has been designated as an input capture register, and buffer operation has been designated for TGRA and TGRC.

Counter clearing by input capture has been set for TCNT, and both rising and falling edges have been selected as the TIOCA pin input capture input edge.

As buffer operation has been set, when the TCNT value is stored in TGRA upon occurrence of input capture A, the value previously stored in TGRA is simultaneously transferred to TGRC.

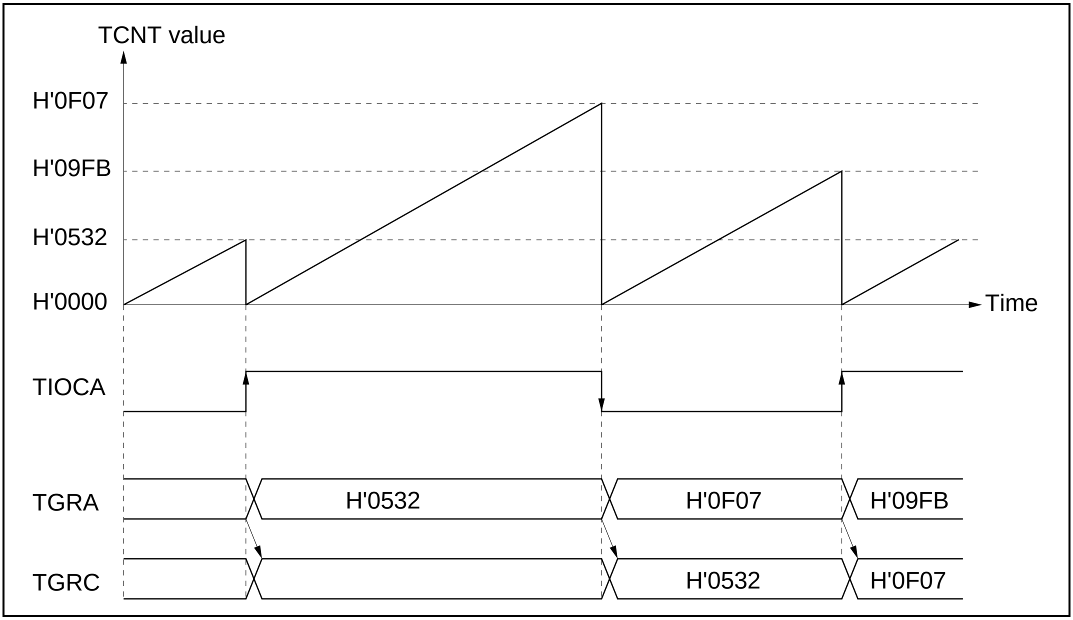


Figure 10.20 Example of Buffer Operation (2)

10.4.5 Cascaded Operation

In cascaded operation, two 16-bit counters for different channels are used together as a 32-bit counter.

This function works by counting the channel 1 (channel 4) counter clock upon overflow/underflow of TCNT2 (TCNT5) as set in bits TPSC2 to TPSC0 in TCR.

Underflow occurs only when the lower-16-bit TCNT is in phase counting mode.

Table 10.6 shows the register combinations used in cascaded operation.

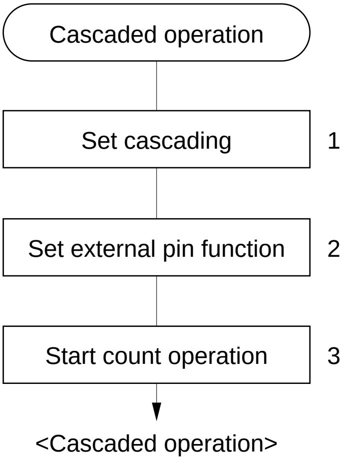
Note: When phase counting mode is set for channel 1 or 4, the counter clock setting is invalid and the counter operates independently in phase counting mode.

Table 10.6 Cascading Combinations

Combination	Upper 16 Bits	Lower 16 Bits
Channels 1 and 2	TCNT1	TCNT2
Channels 4 and 5	TCNT4	TCNT5

Example of Cascaded Operation Setting Procedure

Figure 10.21 shows an example of the setting procedure for cascaded operation.



1. Set bits TPSC2 to TPSC0 in the channel 1 (channel 4) TCR to B'111 to select TCNT2 (TCNT5) overflow/underflow counting.
2. Set the external pin function with the pin function controller (PFC).
3. Set the CST bit in TSTR for the upper and lower channel to 1 to start the count operation.

Figure 10.21 Cascaded Operation Setting Procedure

Examples of Cascaded Operation

Figure 10.22 illustrates the operation when counting upon TCNT2 overflow/underflow has been set for TCNT1, and phase counting mode has been designated for channel 2.

TCNT1 is incremented by TCNT2 overflow and decremented by TCNT2 underflow.

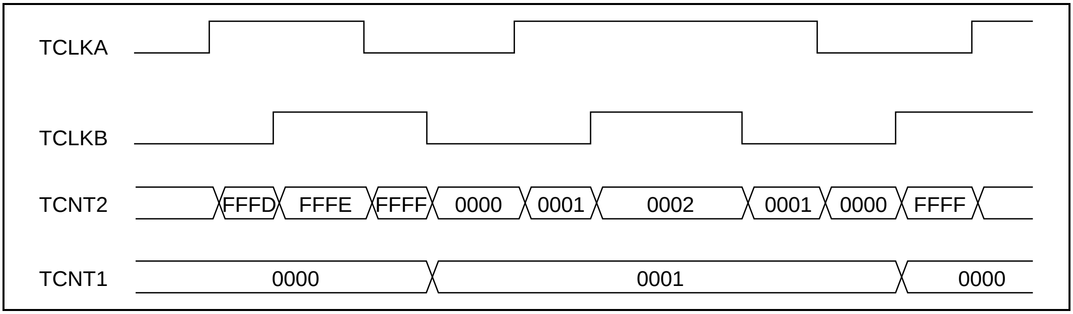


Figure 10.22 Example of Cascaded Operation

10.4.6 PWM Modes

In PWM mode, PWM waveforms are output from the output pins. 0, 1, or toggle output can be selected as the output level in response to compare match of each TGR.

Designating TGR compare match as the counter clearing source enables the cycle to be set in that register. All channels can be designated for PWM mode independently. Synchronous operation is also possible.

There are two PWM modes, as described below.

- PWM mode 1

  PWM output is generated from the TIOCA and TIOCC pins by pairing TGRA with TGRB and TGRC with TGRD. The output specified by TIOR bits IOA3 to IOA0 or IOC3 to IOC0 is performed from the TIOCA or TIOCC pin upon compare match A or C. Also, the output specified by TIOR bits IOB3 to IOB0 or IOD3 to IOD0 is performed upon compare match B or D. The initial output value is the value set in TGRA or TGRC. If the set values of the paired TGR registers are identical, the output value does not change when a compare match occurs.

  In PWM mode 1, a maximum 8-phase PWM output is possible.

- PWM mode 2

  PWM output is generated using one TGR register as the cycle register and the others as duty registers. The output specified by TIOR is performed upon compare match. Upon counter clearing by a cycle register compare match, the output value of each pin is the initial value set in TIOR. If the set values of the cycle and duty registers are identical, the output value does not change when a compare match occurs.

  In PWM mode 2, a maximum 15-phase PWM output is possible by combined use with synchronous operation.

The correspondence between PWM output pins and registers is shown in table 10.7.

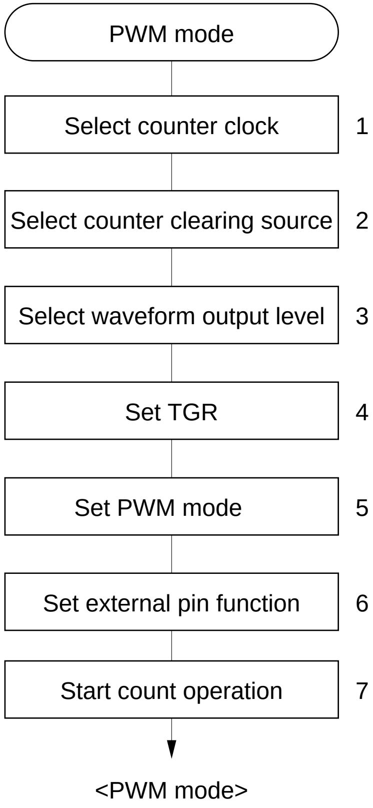
Table 10.7 PWM Output Registers and Output Pins

		Output Pins	
Channel	**Registers**	**PWM Mode 1**	**PWM Mode 2**
0	TGR0A	TIOC0A	TIOC0A
	TGR0B		TIOC0B
	TGR0C	TIOC0C	TIOC0C
	TGR0D		TIOC0D
1	TGR1A	TIOC1A	TIOC1A
	TGR1B		TIOC1B
2	TGR2A	TIOC2A	TIOC2A
	TGR2B		TIOC2B
3	TGR3A	TIOC3A	TIOC3A
	TGR3B		TIOC3B
	TGR3C	TIOC3C	TIOC3C
	TGR3D		TIOC3D
4	TGR4A	TIOC4A	TIOC4A
	TGR4B		TIOC4B
5	TGR5A	TIOC5A	TIOC5A
	TGR5B		TIOC5B

Note: In PWM mode 2, PWM output is not possible for the TGR register in which the cycle is set.

Example of PWM Mode Setting Procedure

Figure 10.23 shows an example of the PWM mode setting procedure.



1. Select the counter clock with bits TPSC2 to TPSC0 in TCR. At the same time, select the input clock edge with bits CKEG1 and CKEG0 in TCR.
2. Use bits CCLR2 to CCLR0 in TCR to select the TGR register to be used as the TCNT clearing source.
3. Use TIOR to designate the output compare register, and select the initial value and output value.
4. Set the cycle in the TGR register selected in 2, and set the duty in the other TGR registers.
5. Select the PWM mode with bits MD3 to MD0 in TMDR.
6. Set the external pin function with the pin function controller (PFC).
7. Set the CST bit in TSTR to 1 to start the count operation.

Figure 10.23 Example of PWM Mode Setting Procedure

Examples of PWM Mode Operation

Figure 10.24 shows an example of PWM mode 1 operation.

In this example, TGRA compare match is set as the TCNT clearing source, 0 is set for the TGRA initial output value and output value, and 1 output is set as the TGRB output value.

In this case, the value set in TGRA is used as the cycle, and the value set in TGRB as the duty.

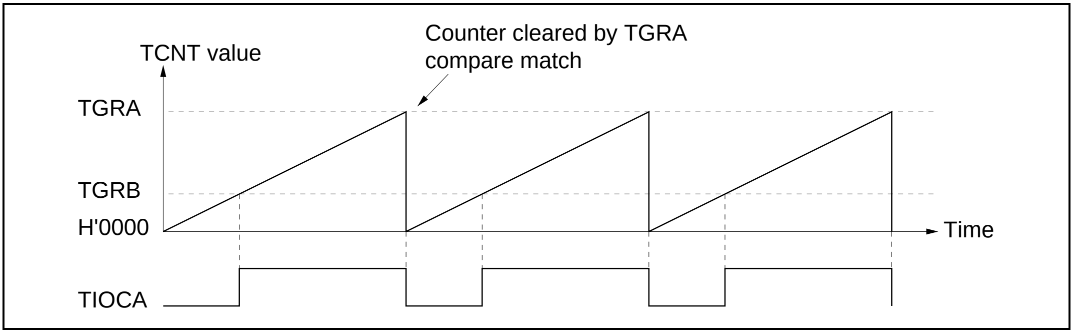


Figure 10.24 Example of PWM Mode Operation (1)

Figure 10.25 shows an example of PWM mode 2 operation.

In this example, synchronous operation is designated for channels 0 and 1, TGR1B compare match is set as the TCNT clearing source, and 0 is set for the initial output value and 1 for the output value of the other TGR registers, to output a 5-phase PWM waveform.

In this case, the value set in TGR1B is used as the cycle, and the values set in the other TGR registers as the duty.

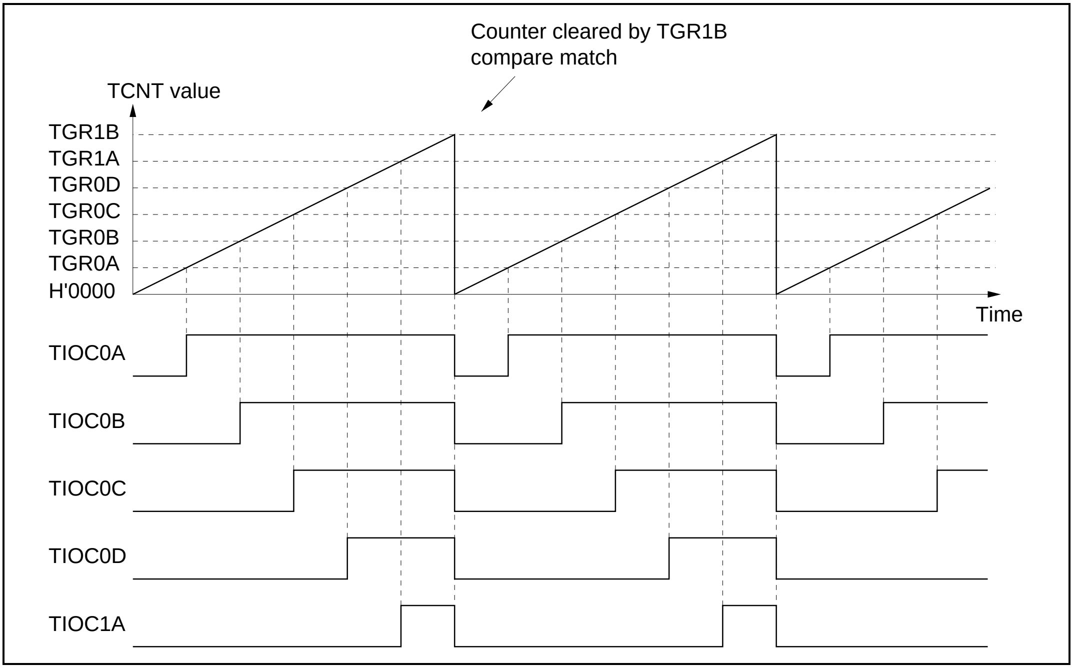


Figure 10.25 Example of PWM Mode Operation (2)

Figure 10.26 shows examples of PWM waveform output with 0% duty and 100% duty in PWM mode.

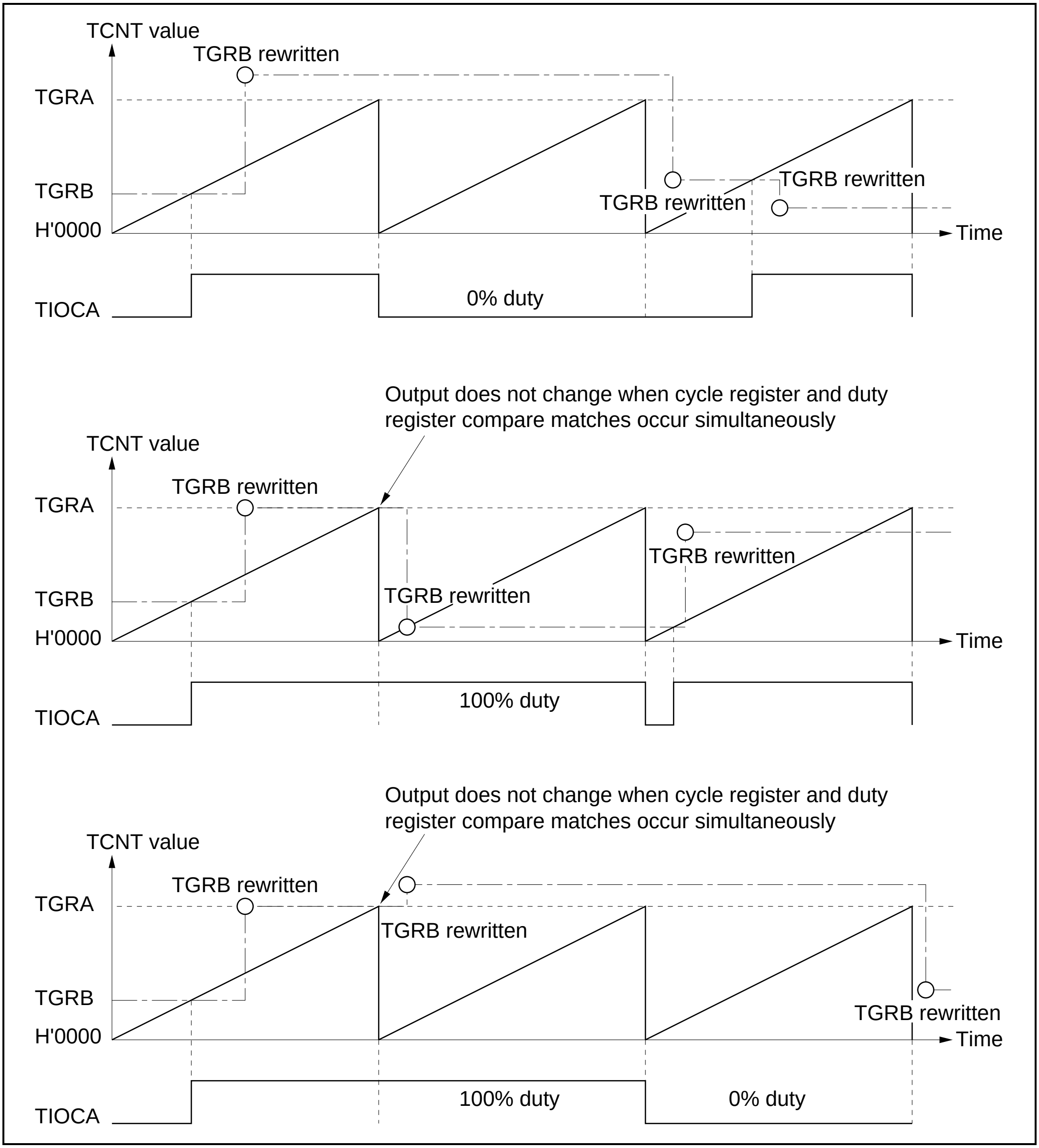


Figure 10.26 Example of PWM Mode Operation (3)

10.4.7 Phase Counting Mode

In phase counting mode, the phase difference between two external clock inputs is detected and TCNT is incremented/decremented accordingly. This mode can be set for channels 1, 2, 4, and 5.

When phase counting mode is set, an external clock is selected as the counter input clock and TCNT operates as an up/down-counter regardless of the setting of bits TPSC2 to TPSC0 and bits CKEG1 and CKEG0 in TCR. However, the functions of bits CCLR1 and CCLR0 in TCR, and of TIOR, TIER, and TGR are valid, and input capture/compare match and interrupt functions can be used.

When overflow occurs while TCNT is counting up, the TCFV flag in TSR is set; when underflow occurs while TCNT is counting down, the TCFU flag is set.

The TCFD bit in TSR is the count direction flag. By reading the TCFD flag it is possible to check whether TCNT is counting up or down.

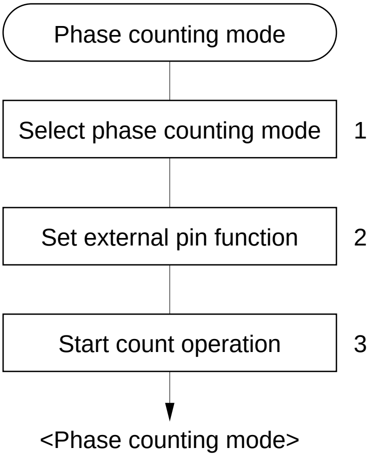
Table 10.8 shows the correspondence between external clock pins and channels.

Table 10.8 Phase Counting Mode Clock Input Pins

	External Clock Pins	
Channel	**A-Phase**	**B-Phase**
When channel 1 or 5 is set to phase counting mode	TCLKA	TCLKB
When channel 2 or 4 is set to phase counting mode	TCLKC	TCLKD

Example of Phase Counting Mode Setting Procedure

Figure 10.27 shows an example of the phase counting mode setting procedure.



1. Select phase counting mode with bits MD3 to MD0 in TMDR.
2. Set the external pin function with the pin function controller (PFC).
3. Set the CST bit in TSTR to 1 to start the count operation.

Figure 10.27 Example of Phase Counting Mode Setting Procedure

Examples of Phase Counting Mode Operation

In phase counting mode, TCNT counts up or down according to the phase difference between two external clocks. There are four modes, according to the count conditions.

Phase counting mode 1: Figure 10.28 shows an example of phase counting mode 1 operation, and table 10.9 summarizes the TCNT up/down-count conditions.

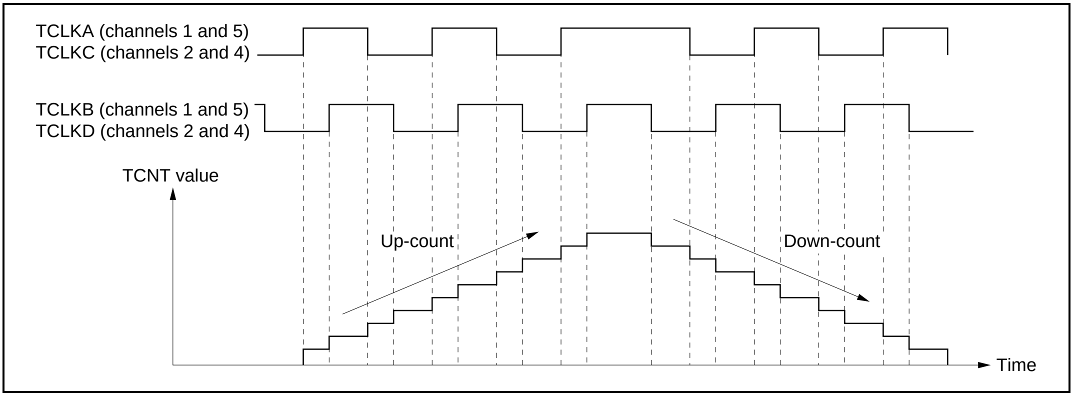


Figure 10.28 Example of Phase Counting Mode 1 Operation

Table 10.9 Up/Down-Count Conditions in Phase Counting Mode 1

TCLKA (Channels 1 and 5) TCLKC (Channels 2 and 4)	TCLKB (Channels 1 and 5) TCLKD (Channels 2 and 4)	Operation
High level	↑	Up-count
Low level	↓	
↑	Low level	
↓	High level	
High level	↓	Down-count
Low level	↑	
↑	High level	
↓	Low level	

Legend:
↑: Rising edge
↓: Falling edge

Phase counting mode 2: Figure 10.29 shows an example of phase counting mode 2 operation, and table 10.10 summarizes the TCNT up/down-count conditions.

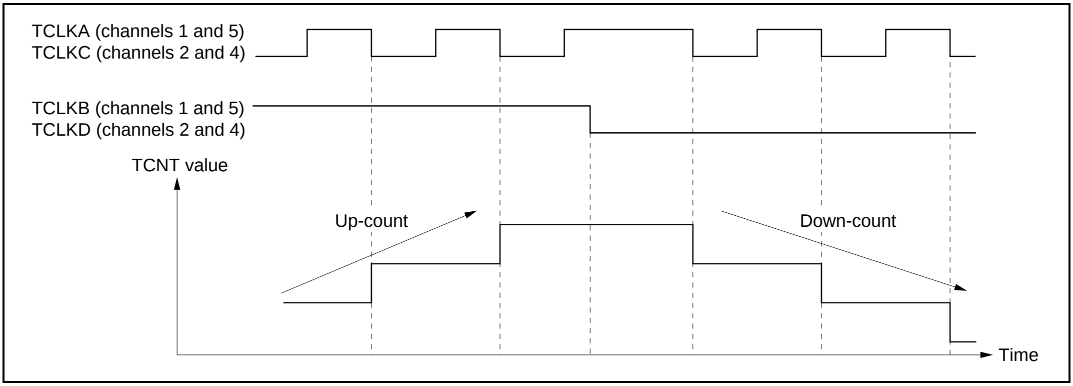


Figure 10.29 Example of Phase Counting Mode 2 Operation

Table 10.10 Up/Down-Count Conditions in Phase Counting Mode 2

TCLKA (Channels 1 and 5) TCLKC (Channels 2 and 4)	TCLKB (Channels 1 and 5) TCLKD (Channels 2 and 4)	Operation
High level	↑	Don’t care
Low level	↓	Don’t care
↑	Low level	Don’t care
↓	High level	Up-count
High level	↓	Don’t care
Low level	↑	Don’t care
↑	High level	Don’t care
↓	Low level	Down-count

Legend:
↑: Rising edge
↓: Falling edge

Phase counting mode 3: Figure 10.30 shows an example of phase counting mode 3 operation, and table 10.11 summarizes the TCNT up/down-count conditions.

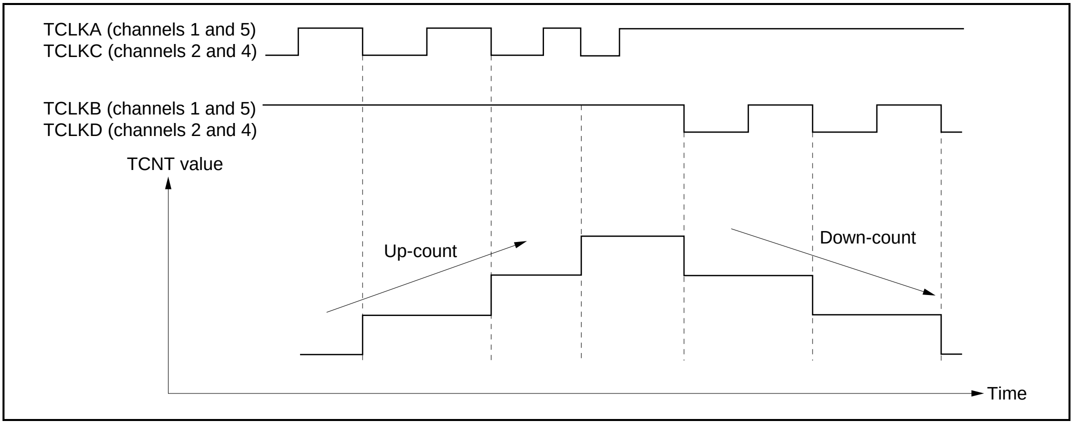


Figure 10.30 Example of Phase Counting Mode 3 Operation

Table 10.11 Up/Down-Count Conditions in Phase Counting Mode 3

TCLKA (Channels 1 and 5) **TCLKC (Channels 2 and 4)**	**TCLKB (Channels 1 and 5)** **TCLKD (Channels 2 and 4)**	**Operation**
High level	↑	Don’t care
Low level	↓	Don’t care
↑	Low level	Don’t care
↓	High level	Up-count
High level	↓	Down-count
Low level	↑	Don’t care
↑	High level	Don’t care
↓	Low level	Don’t care

Legend:
↑: Rising edge
↓: Falling edge

Phase counting mode 4: Figure 10.31 shows an example of phase counting mode 4 operation, and table 10.12 summarizes the TCNT up/down-count conditions.

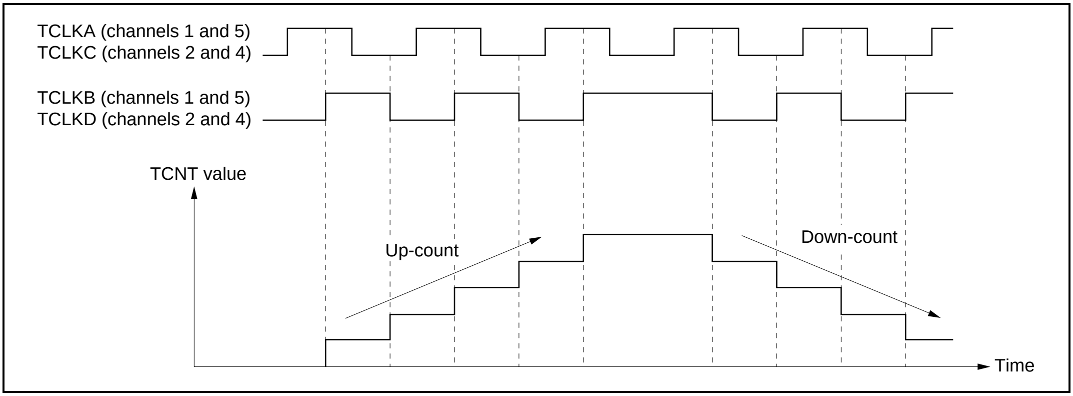


Figure 10.31 Example of Phase Counting Mode 4 Operation

Table 10.12 Up/Down-Count Conditions in Phase Counting Mode 4

TCLKA (Channels 1 and 5) **TCLKC (Channels 2 and 4)**	**TCLKB (Channels 1 and 5)** **TCLKD (Channels 2 and 4)**	**Operation**
High level	↑	Up-count
Low level	↓	
↑	Low level	Don’t care
↓	High level	
High level	↓	Down-count
Low level	↑	
↑	High level	Don’t care
↓	Low level	

Legend:
↑: Rising edge
↓: Falling edge

Phase Counting Mode Application Example

Figure 10.32 shows an example in which phase counting mode is designated for channel 1, and channel 1 is coupled with channel 0 to input servo motor 2-phase encoder pulses in order to detect the position or speed.

Channel 1 is set to phase counting mode 1, and the encoder pulse A-phase and B-phase are input to TCLKA and TCLKB.

Channel 0 operates with TCNT counter clearing by TGR0C compare match; TGR0A and TGR0C are used for the compare match function, and are set with the speed control period and position control period. TGR0B is used for input capture, with TGR0B and TGR0D operating in buffer mode. The channel 1 counter input clock is designated as the TGR0B input capture source, and detection of the pulse width of 2-phase encoder 4-multiplication pulses is performed.

TGR1A and TGR1B for channel 1 are designated for input capture, channel 0 TGR0A and TGR0C compare matches are selected as the input capture source, and these registers store the up/down-counter values for the respective control periods.

This procedure enables accurate position/speed detection to be achieved.

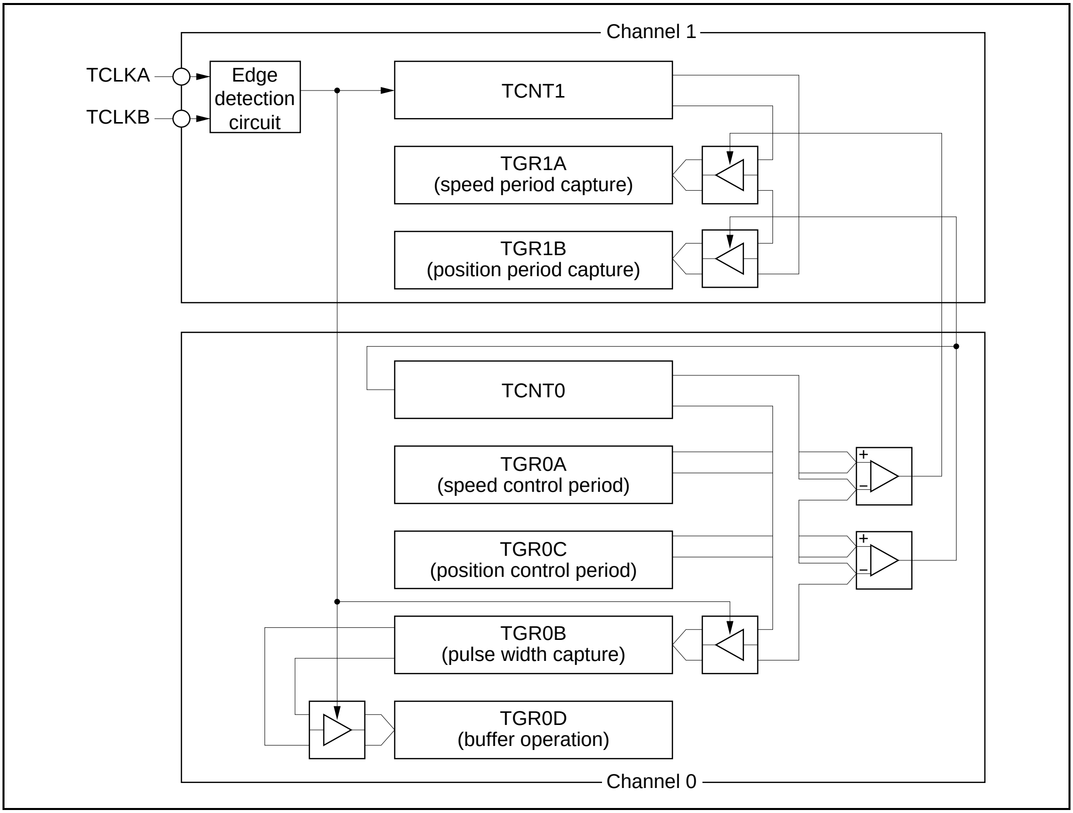

Channel 1
TCLKA
TCLKB
Edge detection circuit
TCNT1
TGR1A (speed period capture)
TGR1B (position period capture)
TCNT0
TGR0A (speed control period)
TGR0C (position control period)
TGR0B (pulse width capture)
TGR0D (buffer operation)
+
−
+
−
Channel 0

Figure 10.32 Phase Counting Mode Application Example

10.5 Interrupts

10.5.1 Interrupt Sources and Priorities

There are three kinds of TPU interrupt source: TGR input capture/compare match, TCNT overflow, and TCNT underflow. Each interrupt source has its own status flag and enable/disable bit, allowing generation of internal reset signals to be enabled or disabled individually.

When an interrupt request is generated, the corresponding status flag in TSR is set to 1. If the corresponding enable/disable bit in TIER is set to 1 at this time, an interrupt is requested. The interrupt request is cleared by clearing the status flag to 0.

Relative channel priorities can be changed by the interrupt controller, but the priority order within a channel is fixed. For details, see section 6, Interrupt Controller (INTC).

Table 10.13 lists the TPU interrupt sources.

Table 10.13 TPU Interrupts

Channel	Interrupt Source	Description	DMAC Activation	Priority
0	TGI0A	TGR0A input capture/compare match	Possible	High
	TGI0B	TGR0B input capture/compare match	Not possible	
	TGI0C	TGR0C input capture/compare match	Not possible	
	TGI0D	TGR0D input capture/compare match	Not possible	
	TCI0V	TCNT0 overflow	Not possible	
1	TGI1A	TGR1A input capture/compare match	Possible	
	TGI1B	TGR1B input capture/compare match	Not possible	
	TCI1V	TCNT1 overflow	Not possible	
	TCI1U	TCNT1 underflow	Not possible	
2	TGI2A	TGR2A input capture/compare match	Possible	
	TGI2B	TGR2B input capture/compare match	Not possible	
	TCI2V	TCNT2 overflow	Not possible	
	TCI2U	TCNT2 underflow	Not possible	
3	TGI3A	TGR3A input capture/compare match	Possible	
	TGI3B	TGR3B input capture/compare match	Not possible	
	TGI3C	TGR3C input capture/compare match	Not possible	
	TGI3D	TGR3D input capture/compare match	Not possible	
	TCI3V	TCNT3 overflow	Not possible	
4	TGI4A	TGR4A input capture/compare match	Possible	
	TGI4B	TGR4B input capture/compare match	Not possible	
	TCI4V	TCNT4 overflow	Not possible	
	TCI4U	TCNT4 underflow	Not possible	
5	TGI5A	TGR5A input capture/compare match	Possible	
	TGI5B	TGR5B input capture/compare match	Not possible	
	TCI5V	TCNT5 overflow	Not possible	
	TCI5U	TCNT5 underflow	Not possible	Low

Note: This table shows the initial state immediately after a reset. The relative channel priorities can be changed by the interrupt controller.

Input Capture/Compare Match Interrupts

An interrupt is requested if the TGIE bit in TIER is set to 1 when the TGF flag in TSR is set to 1 by the occurrence of a TGR input capture/compare match on a particular channel. The interrupt request is cleared by clearing the TGF flag to 0. The TPU has 16 input capture/compare match interrupts, four each for channels 0 and 3, and two each for channels 1, 2, 4, and 5.

Overflow Interrupts

An interrupt is requested if the TCIEV bit in TIER is set to 1 when the TCFV flag in TSR is set to 1 by the occurrence of TCNT overflow on a particular channel. The interrupt request is cleared by clearing the TCFV flag to 0. The TPU has six overflow interrupts, one for each channel.

Underflow Interrupts

An interrupt is requested if the TCIEU bit in TIER is set to 1 when the TCFU flag in TSR is set to 1 by the occurrence of TCNT underflow on a particular channel. The interrupt request is cleared by clearing the TCFU flag to 0. The TPU has four overflow interrupts, one each for channels 1, 2, 4, and 5.

10.5.2 DMAC Activation

The TGRA input capture/compare match interrupt for a channel can be used as a DMAC activation source. For details, see section 9, Direct Memory Access Controller (DMAC).

In the TPU, a total of six TGRA input capture/compare match interrupts can be used as DMAC activation sources, one for each channel.

10.5.3 A/D Converter Activation

The A/D converter can be activated by the TGRA input capture/compare match interrupt for a channel.

If the TTGE bit in TIER is set to 1 when the TFGA flag in TSR is set to 1 by the occurrence of a TGRA input capture/compare match interrupt on a particular channel, a request to start A/D conversion is sent to the A/D converter. If the TPU conversion start trigger has been selected on the A/D converter side at this time, A/D conversion is started.

In the TPU, a total of six TGRA input capture/compare match interrupts can be used as A/D converter conversion start sources, one for each channel.

10.6 Operation Timing

10.6.1 Input/Output Timing

TCNT Count Timing

Figure 10.33 shows TCNT count timing in input clock operation, and figure 10.34 shows TCNT count timing in external clock operation.

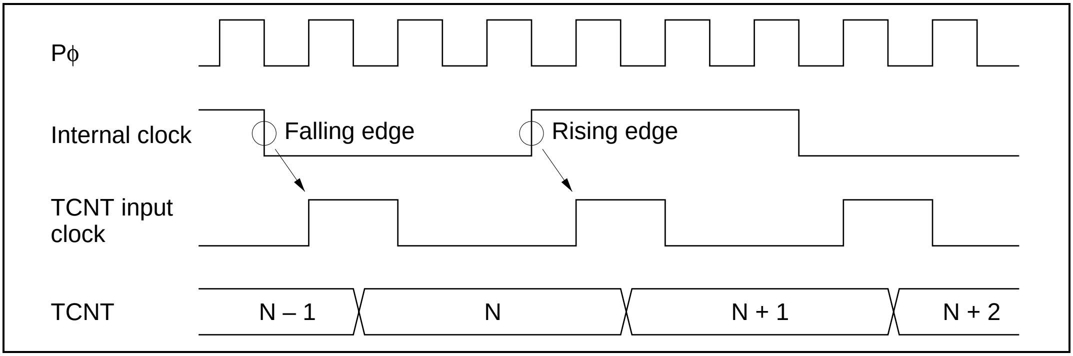


Figure 10.33 Count Timing in Internal Clock Operation

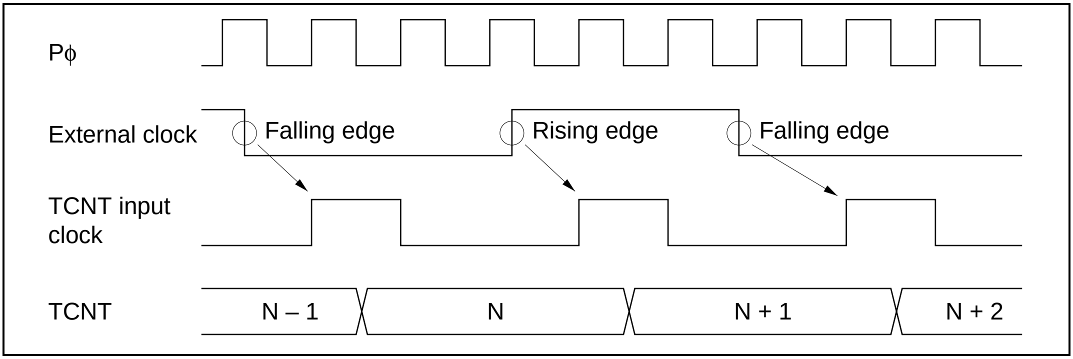


Figure 10.34 Count Timing in External Clock Operation

Output Compare Output Timing

A compare match signal is generated in the final state in which TCNT and TGR match (the point at which the count value matched by TCNT is updated). When a compare match signal is generated, the output value set in TIOR is output at the output compare output pin (the TIOC pin).

After a match between TCNT and TGR, the compare match signal is not generated until the TCNT input clock is generated.

Figure 10.35 shows output compare output timing.

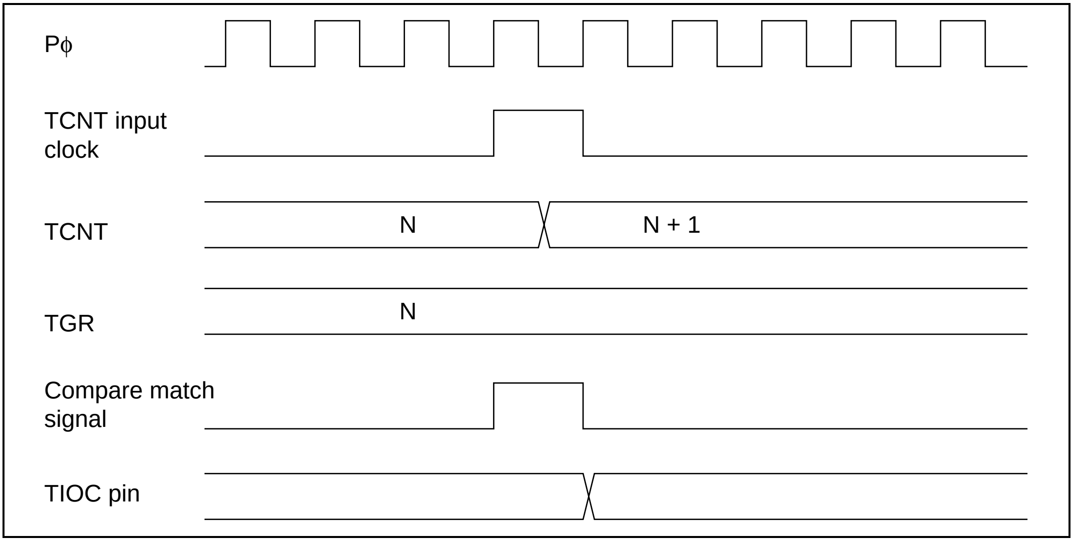


Figure 10.35 Output Compare Output Timing

Input Capture Signal Timing

Figure 10.36 shows input capture signal timing.

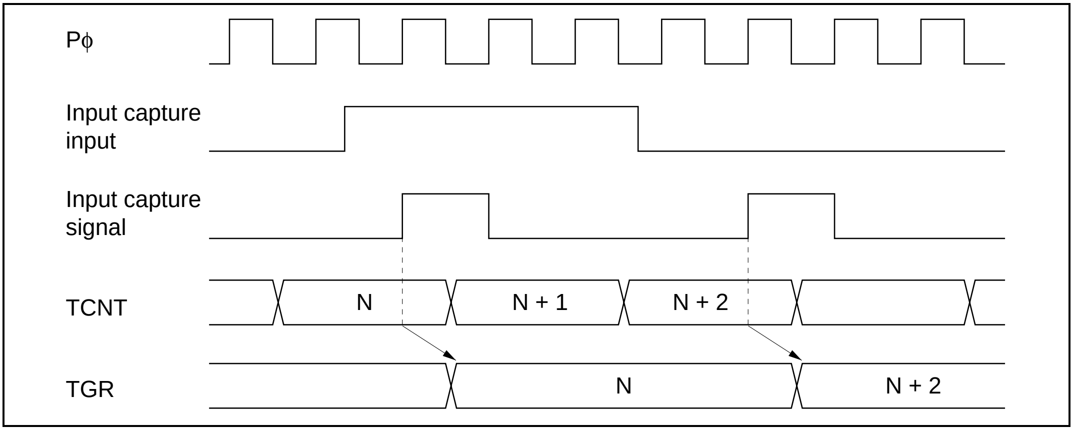


Figure 10.36 Input Capture Input Signal Timing

Timing of Counter Clearing by Compare Match/Input Capture

Figure 10.37 shows the timing when counter clearing by compare match occurrence is specified, and figure 10.38 shows the timing when counter clearing by input capture occurrence is specified.

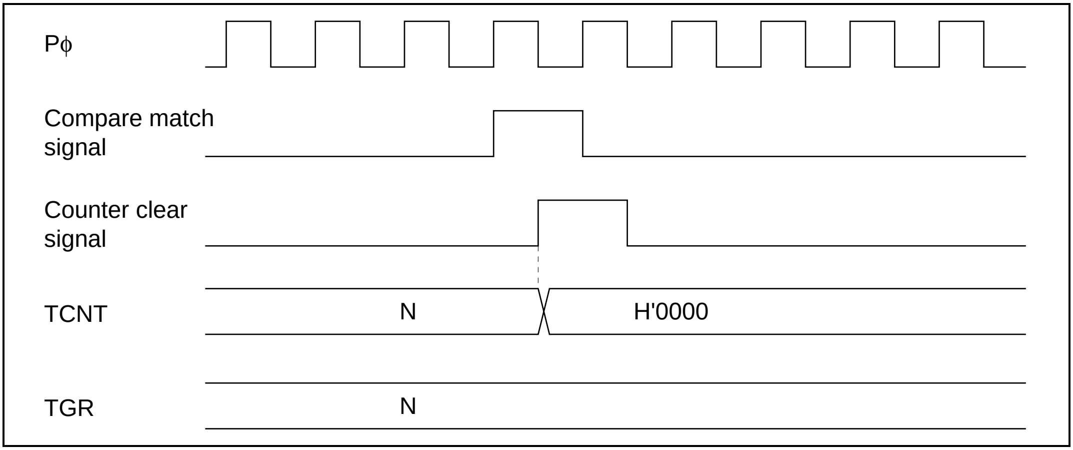


Figure 10.37 Counter Clear Timing (Compare Match)

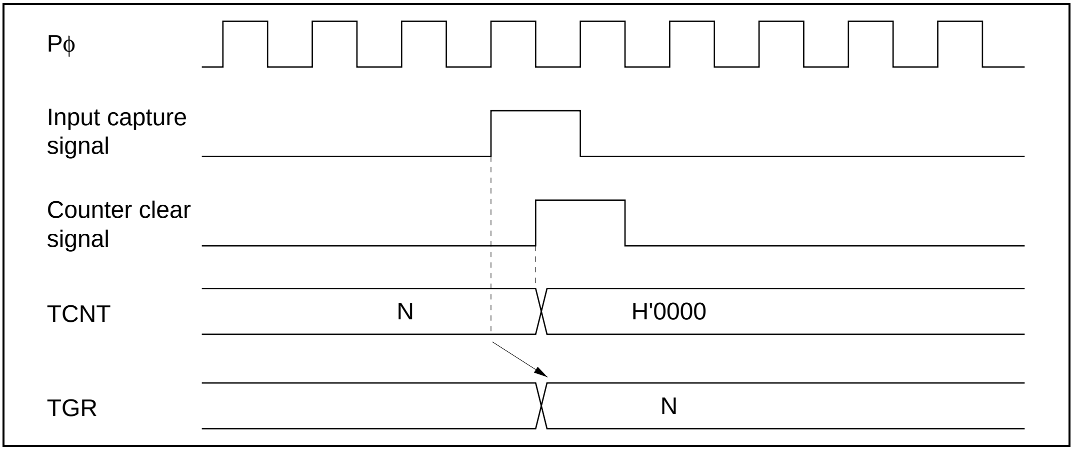


Figure 10.38 Counter Clear Timing (Input Capture)

Buffer Operation Timing

Figures 10.39 and 10.40 show the timing in buffer operation.

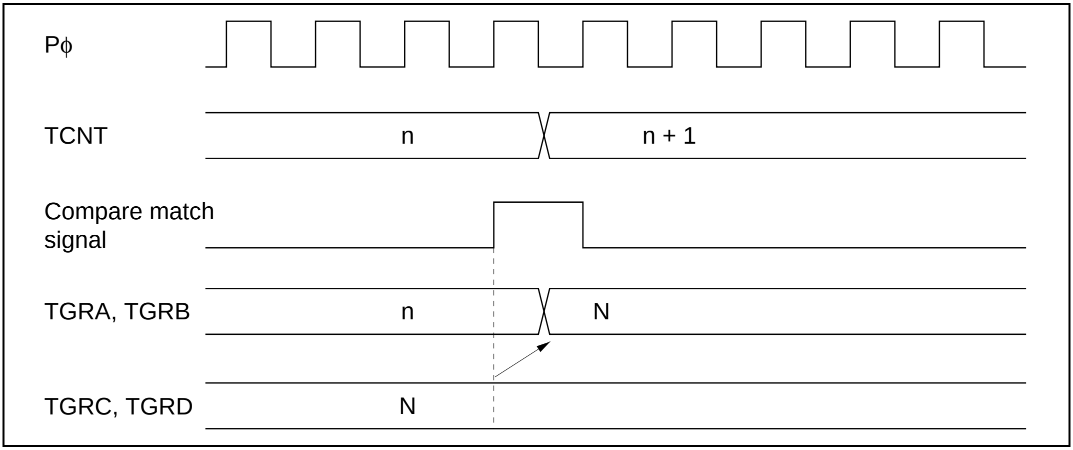


Figure 10.39 Buffer Operation Timing (Compare Match)

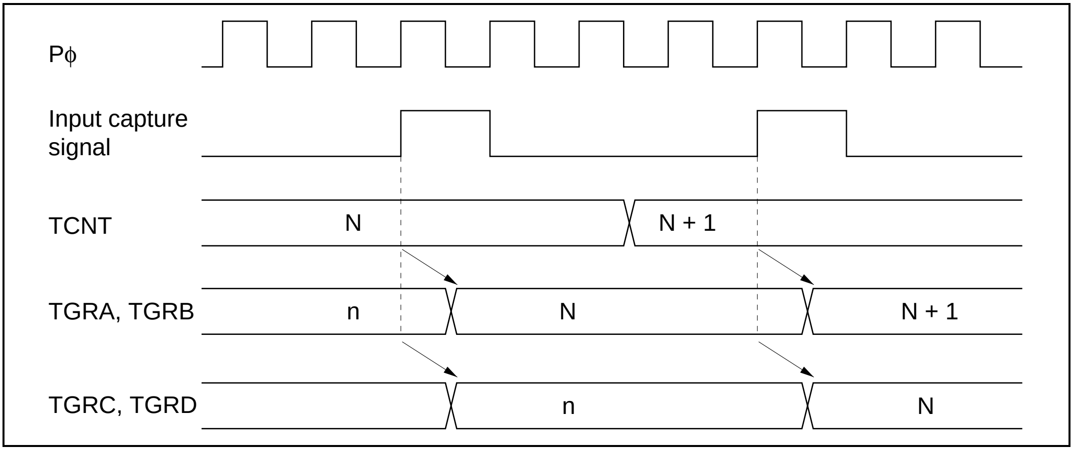


Figure 10.40 Buffer Operation Timing (Input Capture)

10.6.2 Interrupt Signal Timing

TGF Flag Setting Timing in Case of Compare Match

Figure 10.41 shows the timing of setting of the TGF flag in TSR by compare match occurrence, and the TGI interrupt request signal timing.

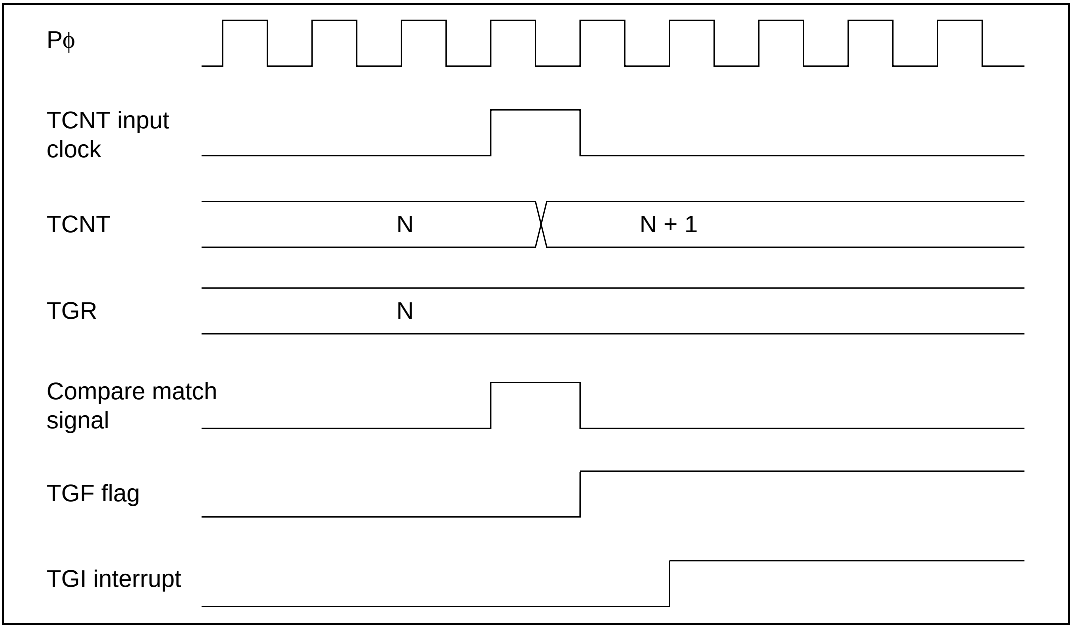


Figure 10.41   TGI Interrupt Timing (Compare Match)

TGF Flag Setting Timing in Case of Input Capture

Figure 10.42 shows the timing of setting of the TGF flag in TSR by input capture occurrence, and the TGI interrupt request signal timing.

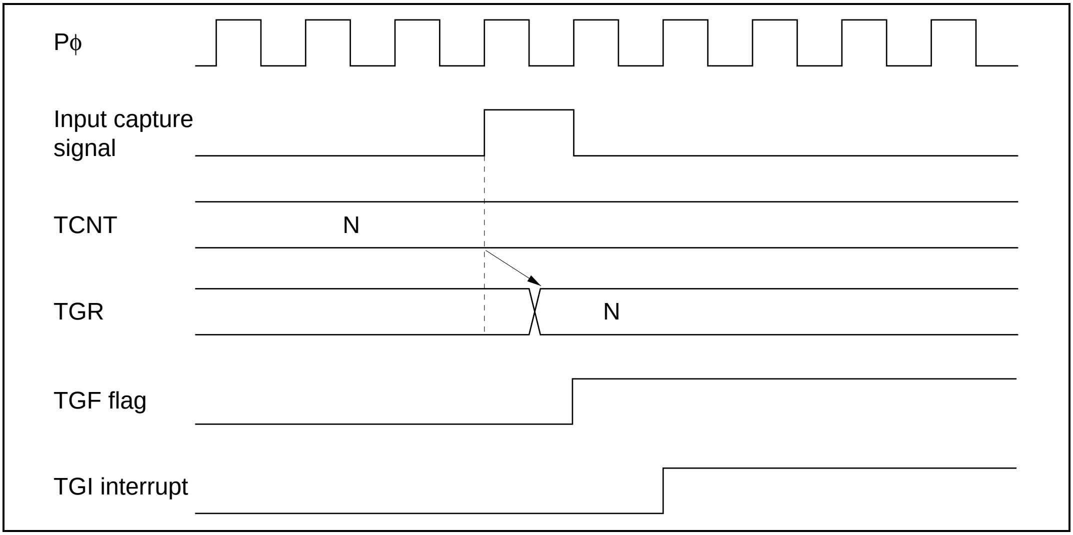


Figure 10.42 TGI Interrupt Timing (Input Capture)

TCFV Flag/TCFU Flag Setting Timing

Figure 10.43 shows the timing of setting of the TCFV flag in TSR by overflow occurrence, and the TCIV interrupt request signal timing.

Figure 10.44 shows the timing of setting of the TCFU flag in TSR by underflow occurrence, and the TCIU interrupt request signal timing.

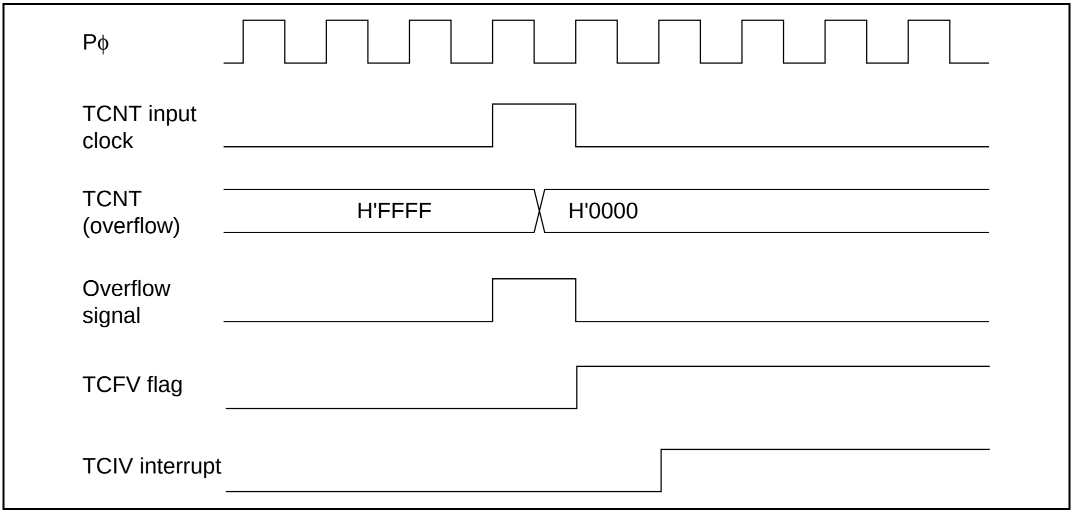


Figure 10.43 TCIV Interrupt Setting Timing

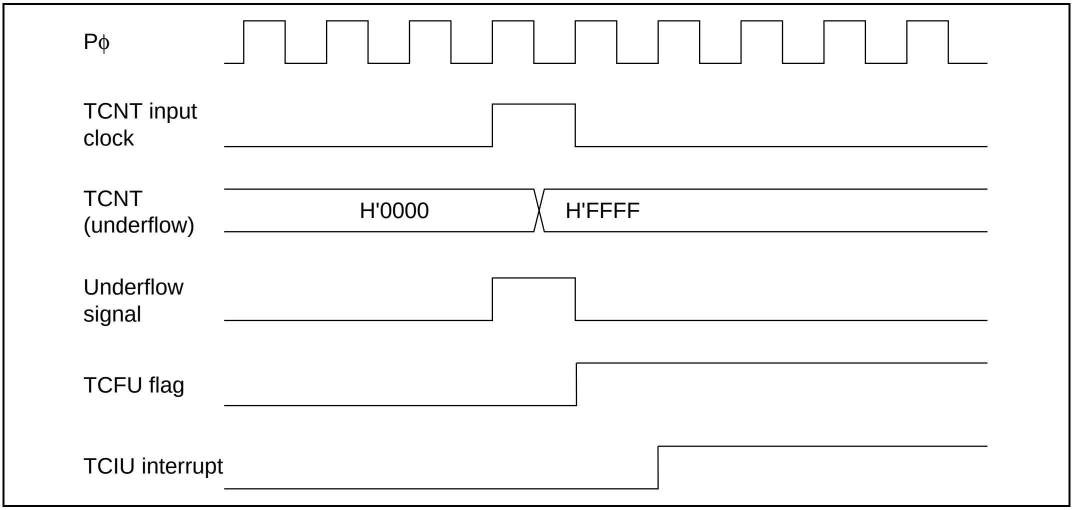


Figure 10.44 TCIU Interrupt Setting Timing

Status Flag Clearing Timing

After a status flag is read as 1 by the CPU, it is cleared by writing 0 to it. When the DMAC is activated, the flag is cleared automatically. Figure 10.45 shows the timing of status flag clearing by the CPU, and figure 10.46 shows the timing of status flag clearing by the DMAC.

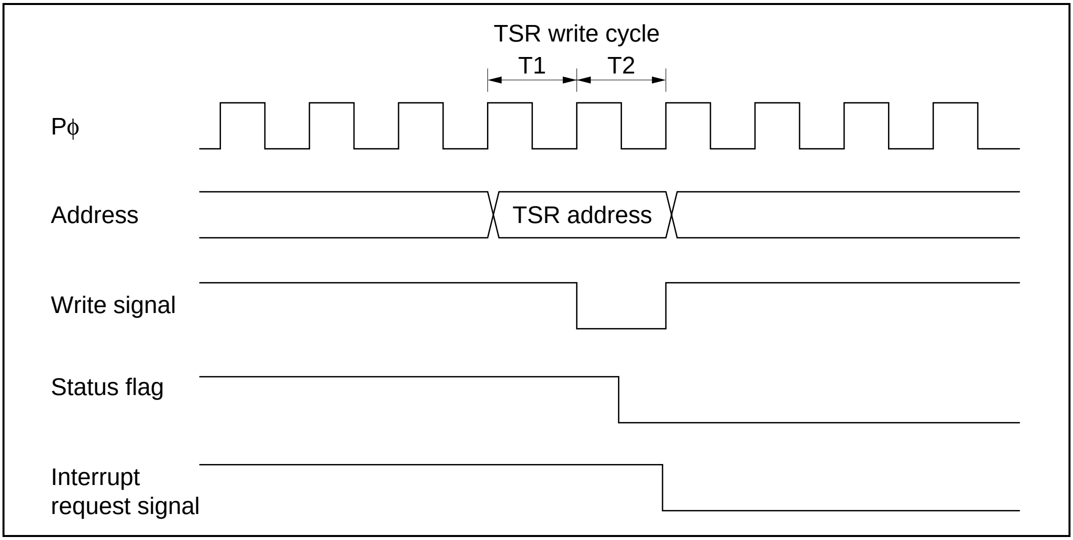


Figure 10.45 Timing of Status Flag Clearing by CPU

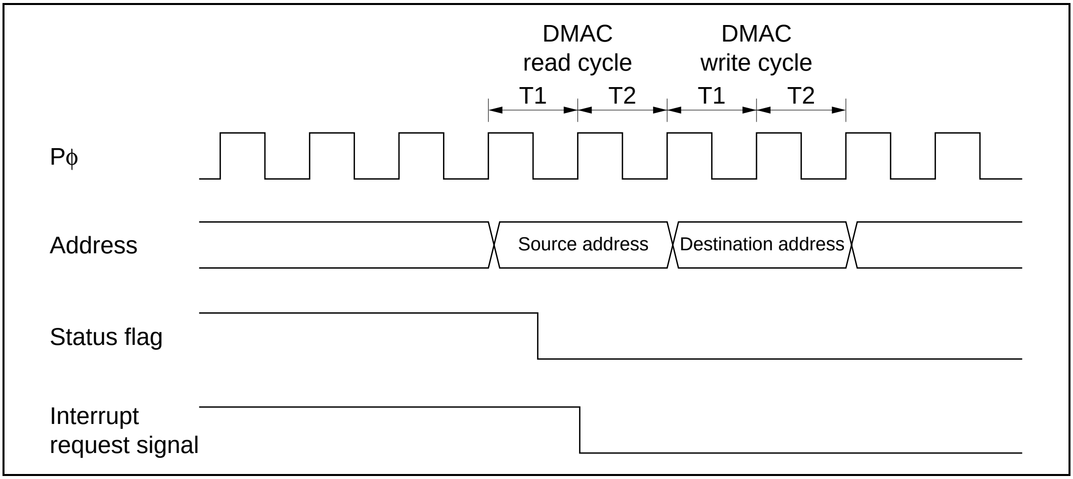


Figure 10.46 Timing of Status Flag Clearing by DMAC Activation

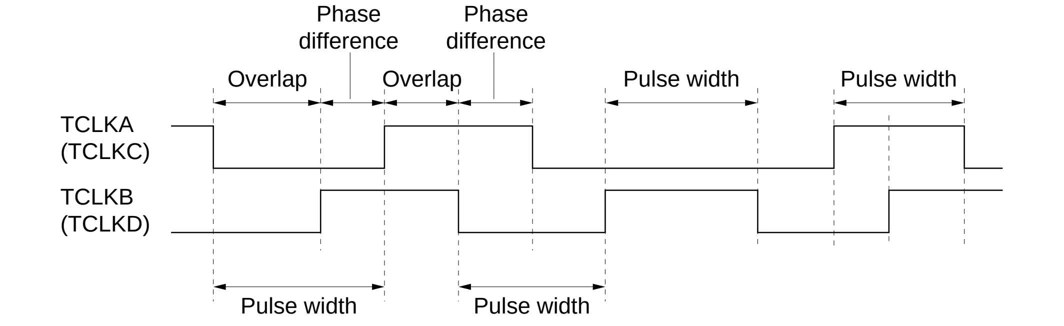
10.7 Usage Notes

Note that the kinds of operation and contention described below occur during TPU operation.

Input Clock Restrictions

The input clock pulse width must be at least 1.5 states in the case of single-edge detection, and at least 2.5 states in the case of both-edge detection. The TPU will not operate properly with a narrower pulse width.

In phase counting mode, the phase difference and overlap between the two input clocks must be at least 1.5 states, and the pulse width must be at least 2.5 states. Figure 10.47 shows the input clock conditions in phase counting mode.



Notes: Phase difference and overlap: 1.5 states or more
Pulse width: 2.5 states or more

Figure 10.47 Phase Difference, Overlap, and Pulse Width in Phase Counting Mode

Caution on Period Setting

When counter clearing by compare match is set, TCNT is cleared in the final state in which it matches the TGR value (the point at which the count value matched by TCNT is updated). Consequently, the actual counter frequency is given by the following formula:

$$f = \frac{P\phi}{(N + 1)}$$

f: Counter frequency
Pϕ: Operating frequency
N: TGR set value

Contention between TCNT Write and Clear Operations

If the counter clear signal is generated in the T2 state of a TCNT write cycle, TCNT clearing takes precedence and the TCNT write is not performed.

Figure 10.48 shows the timing in this case.

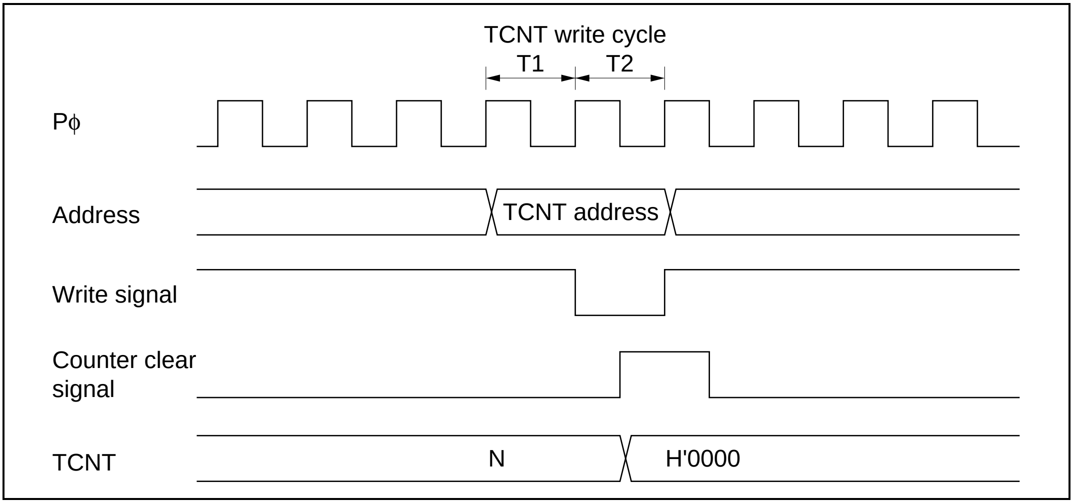


Figure 10.48 Contention between TCNT Write and Clear Operations

Contention between TCNT Write and Increment Operations

If incrementing occurs in the T2 state of a TCNT write cycle, the TCNT write takes precedence and TCNT is not incremented.

Figure 10.49 shows the timing in this case.

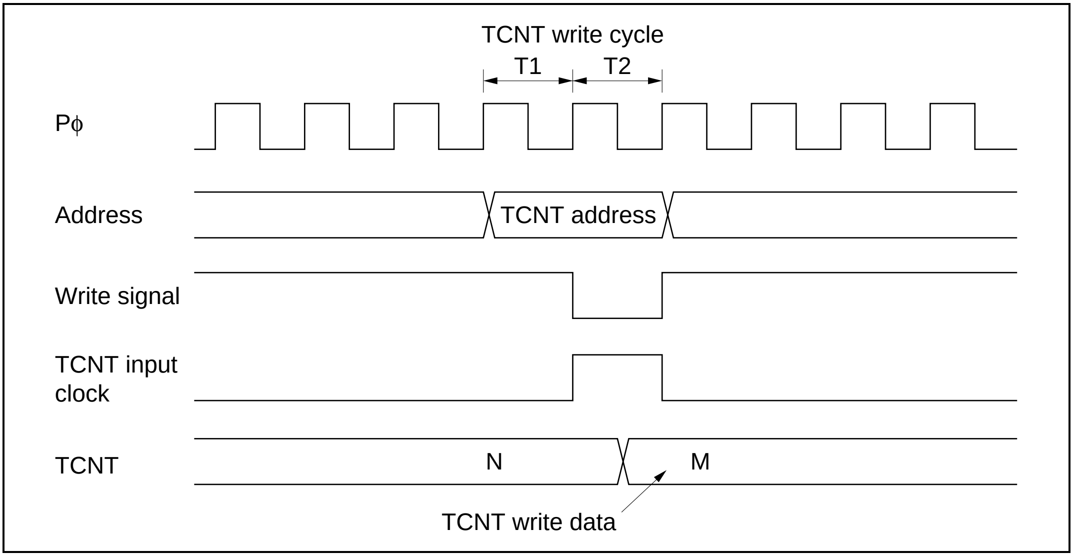


Figure 10.49 Contention between TCNT Write and Increment Operations

Contention between TGR Write and Compare Match

If a compare match occurs in the T2 state of a TGR write cycle, the TGR write takes precedence and the compare match signal is inhibited. A compare match does not occur even if the same value as before is written.

Figure 10.50 shows the timing in this case.

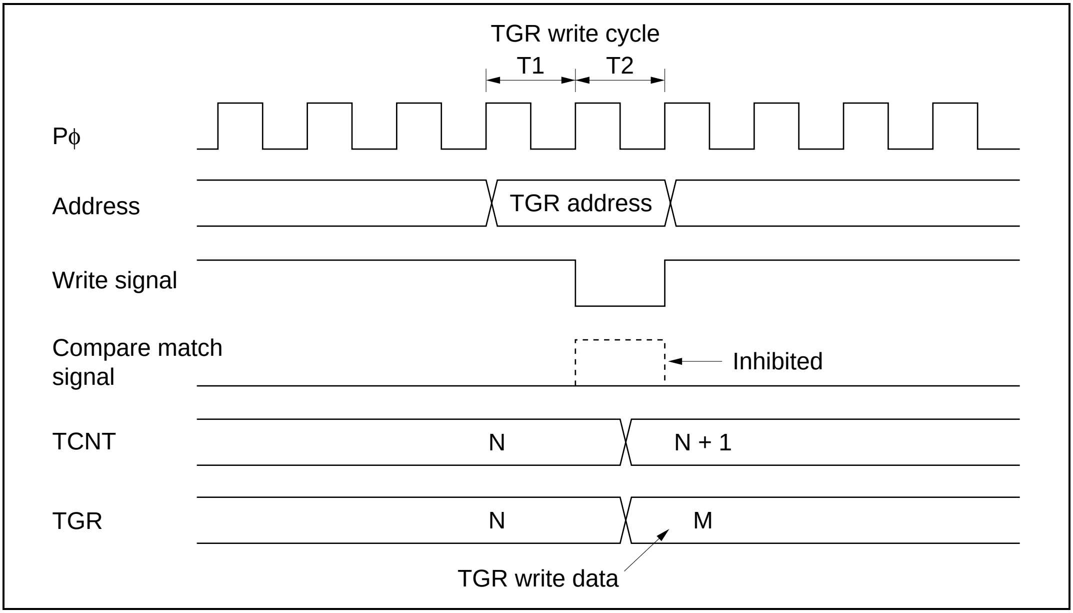


Figure 10.50 Contention between TGR Write and Compare Match

Contention between Buffer Register Write and Compare Match

If a compare match occurs in the T2 state of a TGR write cycle, the data transferred to TGR by the buffer operation will be the write data.

Figure 10.51 shows the timing in this case.

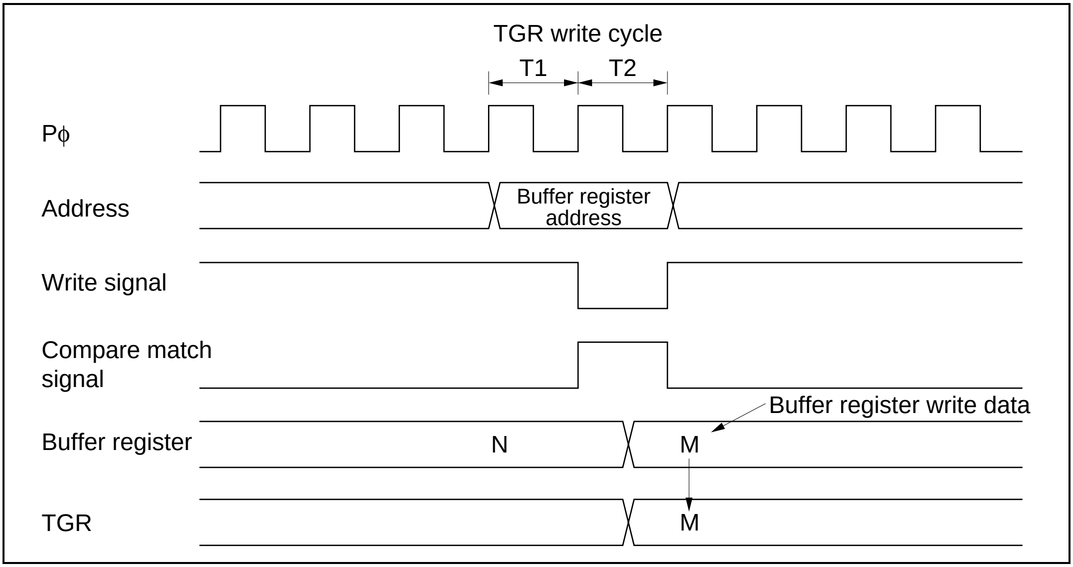


Figure 10.51 Contention between Buffer Register Write and Compare Match

Contention between TGR Read and Input Capture

If the input capture signal is generated in the T1 state of a TGR read cycle, the data that is read will be the data after input capture transfer.

Figure 10.52 shows the timing in this case.

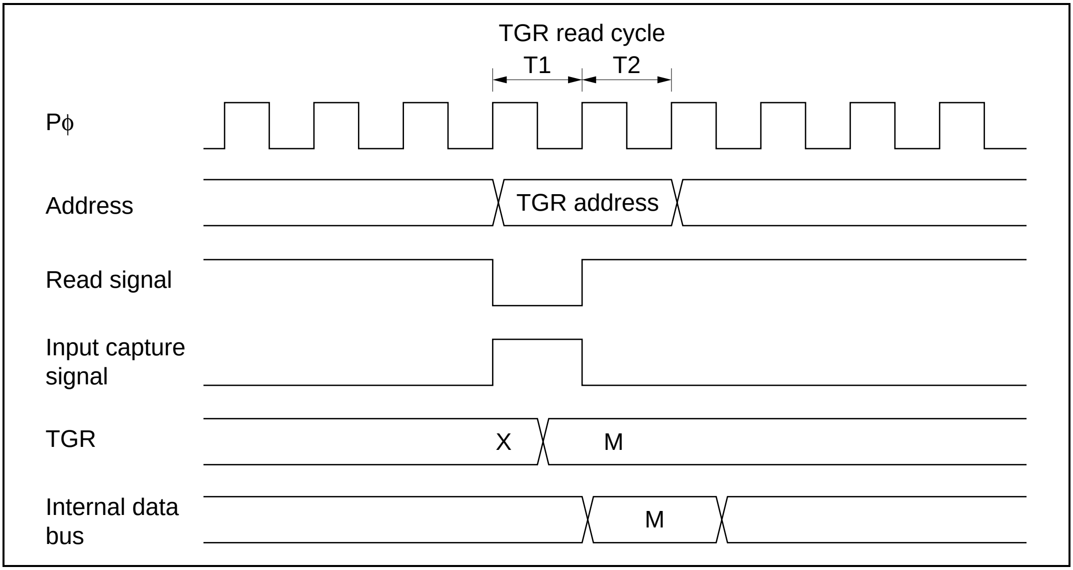


Figure 10.52 Contention between TGR Read and Input Capture

Contention between TGR Write and Input Capture

If the input capture signal is generated in the T2 state of a TGR write cycle, the input capture operation takes precedence and the write to TGR is not performed.

Figure 10.53 shows the timing in this case.

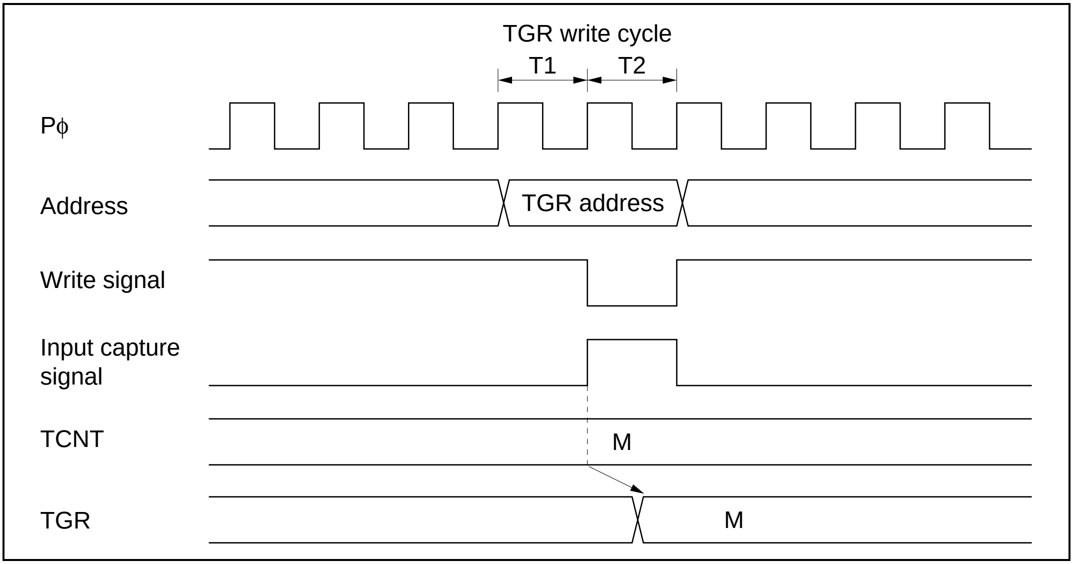


Figure 10.53 Contention between TGR Write and Input Capture

Contention between Buffer Register Write and Input Capture

If the input capture signal is generated in the T2 state of a buffer write cycle, the buffer operation takes precedence and the write to the buffer register is not performed.

Figure 10.54 shows the timing in this case.

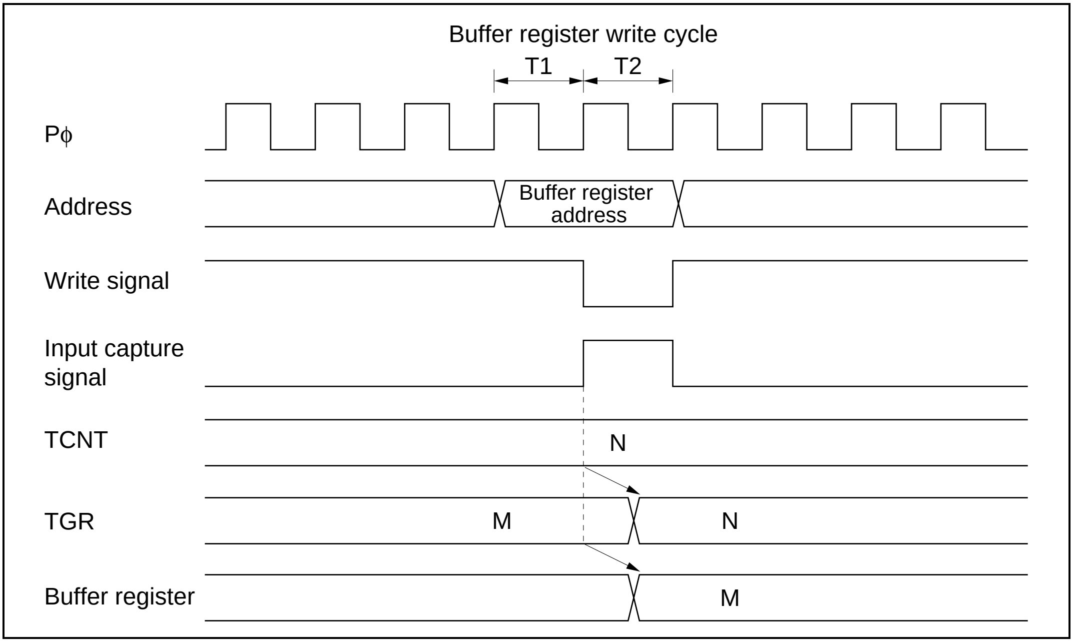


Figure 10.54 Contention between Buffer Register Write and Input Capture

Contention between Overflow/Underflow and Counter Clearing

If overflow/underflow and counter clearing occur simultaneously, TCNT clearing takes precedence and the TCFV/TCFU flag in TSR is not set.

Figure 10.55 shows the operation timing when a TGR compare match is specified as the clearing source, and H'FFFF is set in TGR.

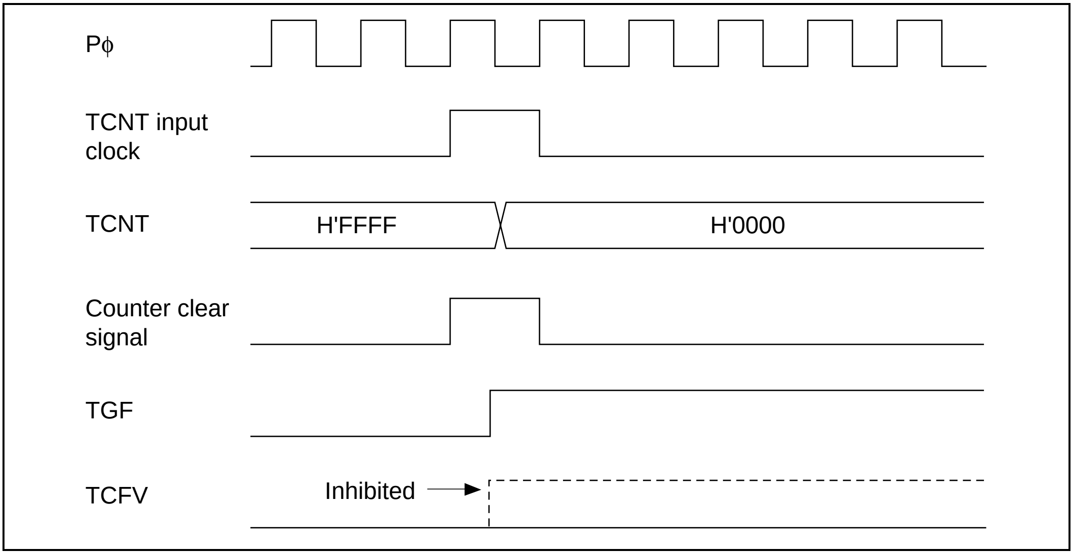


Figure 10.55 Contention between Overflow and Counter Clearing

Contention between TCNT Write and Overflow/Underflow

If there is an up-count or down-count in the T2 state of a TCNT write cycle, and overflow/underflow occurs, the TCNT write takes precedence and the TCFV/TCFU flag in TSR is not set.

Figure 10.56 shows the operation timing in case of contention between a TCNT write and overflow.

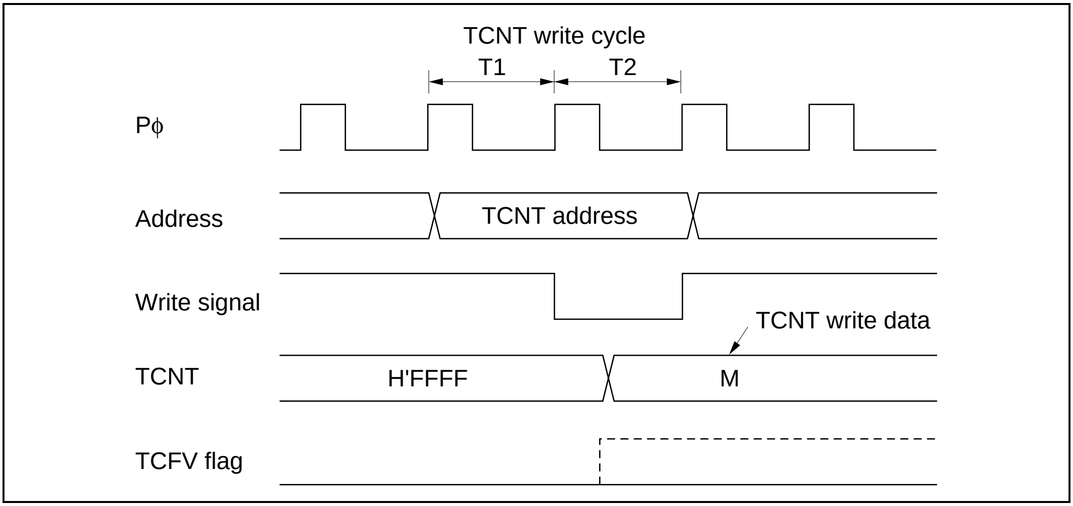


Figure 10.56 Contention between TCNT Write and Overflow

Section 11 Motor Management Timer (MMT)

11.1 Overview

The SH7065 has an on-chip motor management timer (MMT) consisting of a 16-bit timer. The MMT is a single-channel timer capable of outputting 6-phase PWM waveforms with non-overlap times.

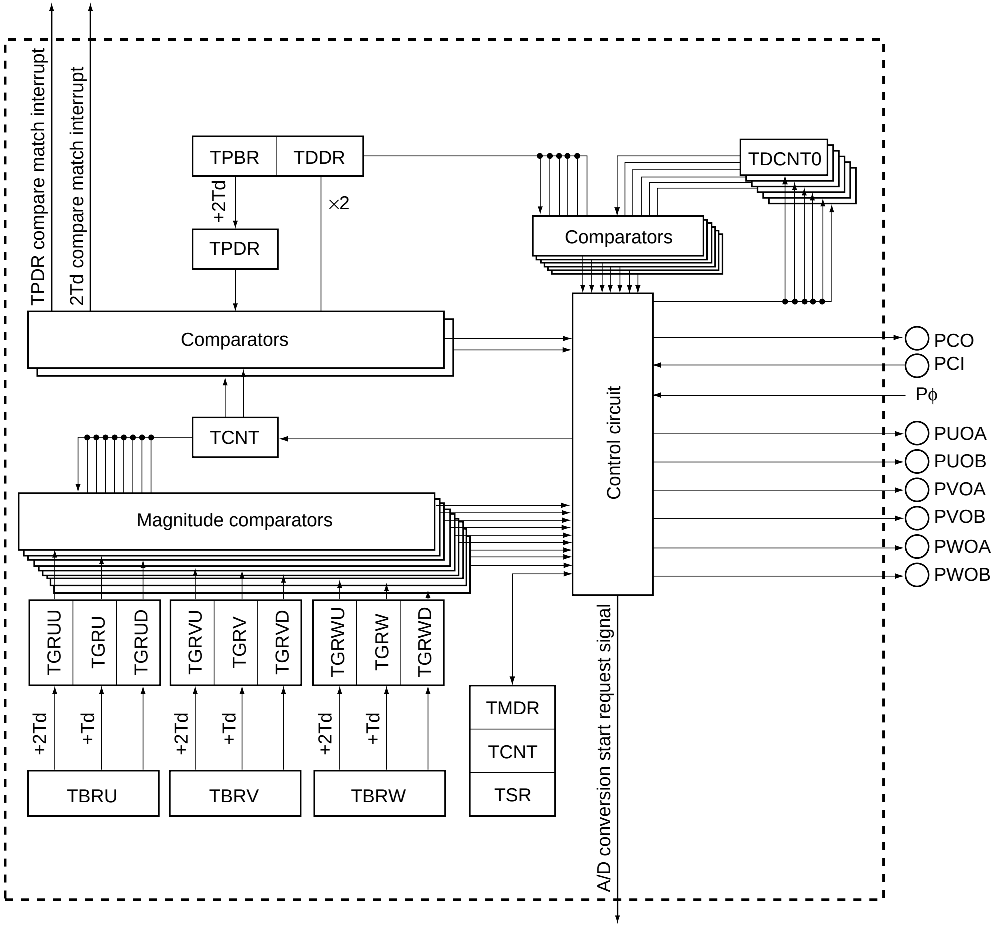
11.1.1 Features

The MMT has the following features:

- Triangular wave comparison type 6-phase PWM waveform output with non-overlap times
  - Non-overlap times generated by timer dead time counters
- Toggle output synchronized with PWM period
- Counter clearing can be performed by an external signal
- Provision for data transfer by means of DMAC activation
- A/D converter conversion start trigger can be generated
  - Compare match signal used as A/D converter conversion start trigger
- Output-off functions
  - PWM output halted by external signal
  - PWM output halted when oscillation stops

11.1.2 Block Diagram

Figure 11.1 shows a block diagram of the MMT. Pϕ is obtained by division of CKP according to a setting in the module clock division setting register.



Legend:

TGR:	Timer general register	TMDR:	Timer mode register
TBR:	Timer buffer register	TCNR:	Timer control register
TDDR:	Timer dead time data register	TSR:	Timer status register
TPDR:	Timer period data register	TCNT:	Timer counter
TPBR:	Timer period buffer register	TDCNT:	Timer dead time counter
Td:	Dead time		

Figure 11.1 Block Diagram of MMT

11.1.3 Pin Configuration

Table 11.1 shows the pin configuration of the MMT.

Table 11.1 MMT Pins

Pin Name	Signal Name	I/O	Function
Counter clear input	PCI	Input	Counter clear signal input
PWM period output	PCO	Output	Toggle output synchronized with PWM period
PWMU phase output A	PUOA	Output	PWMU phase output (positive phase)
PWMU phase output B	PUOB	Output	PWMU phase output (negative phase)
PWMV phase output A	PVOA	Output	PWMV phase output (positive phase)
PWMV phase output B	PVOB	Output	PWMV phase output (negative phase)
PWMW phase output A	PWOA	Output	PWMW phase output (positive phase)
PWMW phase output B	PWOB	Output	PWMW phase output (negative phase)

11.1.4 Register Configuration

Table 11.2 summarizes the MMT registers.

Table 11.2 MMT Registers

Name	Abbre-viation	R/W	Initial Value	Address	Access Size
Timer mode register	TMDR	R/W	H'00	H'FFFF 0480	8, 16, 32
Timer control register	TCNR	R/W	H'00	H'FFFF 0482	8, 16, 32
Timer status register	TSR	R/(W)	H'80	H'FFFF 0484	8, 16, 32
Timer counter	TCNT	R/W	H'0000	H'FFFF 0486	16, 32
Timer buffer register U	TBRU	R/W	H'FFFF	H'FFFF 0490, H'FFFF 049C*	16, 32
Timer buffer register V	TBRV	R/W	H'FFFF	H'FFFF 04A0, H'FFFF 04AC*	16, 32
Timer buffer register W	TBRW	R/W	H'FFFF	H'FFFF 04B0, H'FFFF 04BC*	16, 32
Timer general register UU	TGRUU	R/W	H'FFFF	H'FFFF 0492	16, 32

Name	Abbre- viation	R/W	Initial Value	Address	Access Size
Timer general register VU	TGRVU	R/W	H'FFFF	H'FFFF 04A2	16, 32
Timer general register WU	TGRWU	R/W	H'FFFF	H'FFFF 04B2	16, 32
Timer general register U	TGRU	R/W	H'FFFF	H'FFFF 0494	16, 32
Timer general register V	TGRV	R/W	H'FFFF	H'FFFF 04A4	16, 32
Timer general register W	TGRW	R/W	H'FFFF	H'FFFF 04B4	16, 32
Timer general register UD	TGRUD	R/W	H'FFFF	H'FFFF 0496	16, 32
Timer general register VD	TGRVD	R/W	H'FFFF	H'FFFF 04A6	16, 32
Timer general register WD	TGRWD	R/W	H'FFFF	H'FFFF 04B6	16, 32
Timer dead time counter 0	TDCNT0	R	H'0000	H'FFFF 0498	16, 32
Timer dead time counter 1	TDCNT1	R	H'0000	H'FFFF 049A	16, 32
Timer dead time counter 2	TDCNT2	R	H'0000	H'FFFF 04A8	16, 32
Timer dead time counter 3	TDCNT3	R	H'0000	H'FFFF 04AA	16, 32
Timer dead time counter 4	TDCNT4	R	H'0000	H'FFFF 04B8	16, 32
Timer dead time counter 5	TDCNT5	R	H'0000	H'FFFF 04BA	16, 32
Timer dead time data register	TDDR	R/W	H'FFFF	H'FFFF 048C	16, 32
Timer period buffer register	TPBR	R/W	H'FFFF	H'FFFF 048A	16, 32
Timer period data register	TPDR	R/W	H'FFFF	H'FFFF 0488	16, 32

Note: Registers TBRU to TBRW each have two addresses, a buffer operation address (shown first) and a free operation address (shown second). A value written to the buffer operation address is transferred to the corresponding TGR at the timing set in bits MD1 and MD0 in the timer mode register (TMDR). A value set in the free operation address is transferred to the corresponding TGR immediately.

11.2 Register Descriptions

11.2.1 Timer Mode Register (TMDR)

The timer mode register (TMDR) is an 8-bit readable/writable register that sets the operating mode and selects the PWM output level.

TMDR is initialized to H'00 by a power-on reset and in standby mode. It is not initialized in module standby mode.

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	OLSN	OLSP	MD1	MD0
Initial value:	0	0	0	0	0	0	0	0
R/W:	—	—	—	—	R/W	R/W	R/W	R/W

Bits 7 to 4—Reserved: These bits are always read as 0 and should only be written with 0.

Bit 3—Output Level Select N (OLSN): Selects the negative phase output level in the operating modes.

Bit 3: OLSN	Description	
0	Active level is low	(Initial value)
1	Active level is high	

Bit 2—Output Level Select P (OLSP): Selects the positive phase output level in the operating modes.

Bit 2: OLSP	Description	
0	Active level is low	(Initial value)
1	Active level is high	

Bits 1 and 0—Mode 1 and 0 (MD1, MD0): These bits set the timer operating mode.

Bit 1: MD1	Bit 0: MD0	Description	
0	0	Operation halted	(Initial value)
	1	Operating mode 1 (transfer at crest)	
1	0	Operating mode 2 (transfer at trough)	
	1	Operating mode 3 (transfer at crest and trough)	

11.2.2 Timer Control Register (TCNR)

The timer control register (TCNR) is an 8-bit readable/writable register that controls enabling or disabling of interrupt requests, selects enabling or disabling of register access, selects counter operation or halting, and controls enabling or disabling of toggle output synchronized with the PWM period.

TCNR is initialized to H'00 by a power-on reset and in standby mode. It is not initialized in module standby mode.

Bit:	7	6	5	4	3	2	1	0
	TTGE	CST	RPRO	—	—	—	TGIEN	TGIEM
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	—	—	—	R/W	R/W

Bit 7—A/D Conversion Start Request Enable (TTGE): Enables or disables generation of A/D conversion start requests by a compare match between TCNT and the TPDR register, and by a compare match between TCNT and 2Td (Td: dead time).

Bit 7: TTGE	Description	
0	A/D conversion start request generation disabled	(Initial value)
1	A/D conversion start request generation enabled	

The A/D conversion start timing in each operating mode is shown in table 11.3.

Table 11.3 Conversion Start Timing in Each Operating Mode

Operating Mode	A/D Conversion Start Timing
Operating mode 1 (transfer at crest)	A/D conversion starts at trough
Operating mode 2 (transfer at trough)	A/D conversion starts at crest
Operating mode 3 (transfer at crest and trough)	A/D conversion starts at crest or trough

Bit 6—Timer Counter Start (CST): Selects operation or halting of the timer counter (TCNT) and timer dead time counter (TDCNT).

Bit 6: CST	Description	
0	TCNT and TDCNT operation is halted	(Initial value)
1	TCNT and TDCNT perform count operations	

Bit 5—Register Protect (RPRO): Enables or disables reading of registers other than TSR and writes to registers other than TBRU to TBRW, TPBR, and TSR. Writes to TCNR itself are also disabled. Note that reset input is necessary in order to write to these registers again.

Bit 5: RPRO	Description	
0	Register access enabled	(Initial value)
1	Register access disabled	

Bits 4 to 2—Reserved: These bits are always read as 0 and should only be written with 0.

Bit 1—TGR Interrupt Enable N (TGIEN): Enables or disables interrupt requests by the TGFN bit in the TSR register when TGFN is set to 1.

Bit 1: TGIEN	Description	
0	Interrupt requests (TGIN) by TGFN bit disabled	(Initial value)
1	Interrupt requests (TGIN) by TGFN bit enabled	

Bit 0—TGR Interrupt Enable M (TGIEM): Enables or disables interrupt requests by the TGFM bit in the TSR register when TGFM is set to 1.

Bit 0: TGIEM	Description	
0	Interrupt requests (TGIM) by TGFM bit disabled	(Initial value)
1	Interrupt requests (TGIM) by TGFM bit enabled	

11.2.3 Timer Status Register (TSR)

The timer status register (TSR) is an 8-bit register containing status flags.

TSR is initialized to H'80 by a power-on reset and in standby mode. It is not initialized in module standby mode.

Bit:	7	6	5	4	3	2	1	0
	TCFD	—	—	—	—	—	TGFN	TGFM
Initial value:	1	0	0	0	0	0	0	0
R/W:	R	—	—	—	—	—	R/(W)*	R/(W)*

Note: * Can only be written with 0 for flag clearing.

Bit 7—Count Direction Flag (TCFD): Status flag that indicates the count direction of the TCNT counter.

Bit 7: TCFD	Description	
0	TCNT counts down	
1	TCNT counts up	(Initial value)

Bits 6 to 2—Reserved: These bits are always read as 0 and should only be written with 0.

Bit 1—Output Compare Flag N (TGFN): Status flag that indicates the occurrence of a compare match between TCNT and 2Td (Td: TDDR value).

Bit 1: TGFN	Description	
0	[Clearing condition] When 0 is written to TGFN after reading TGFN = 1	(Initial value)
1	[Setting condition] When TCNT = 2Td	

Bit 0—Output Compare Flag M (TGFM): Status flag that indicates the occurrence of a compare match between TCNT and the TPDR register.

Bit 0: TGFM	Description	
0	[Clearing condition] When 0 is written to TGFM after reading TGFM = 1	(Initial value)
1	[Setting condition] When TCNT = TGRM	

11.2.4 Timer Counter (TCNT)

The timer counter (TCNT) is a 16-bit counter.

TCNT is initialized to H'0000 by a power-on reset and in standby mode. It is not initialized in module standby mode. Only 16-bit access can be used on TCNT; 8-bit access is not possible.

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

11.2.5 Timer Buffer Registers (TBR)

The timer buffer registers (TBR) function as 16-bit buffer registers. The MMT has three TBR registers. The TBR value is transferred to the TGR register at the timing set in the TMDR register (except in the case of a write to the TBR’s free operation address, in which case the value is transferred to the TGR register immediately).

The TBR registers are initialized to H'FFFF by a power-on reset and in standby mode. They are not initialized in module standby mode. Only 16-bit access can be used on the TBR registers; 8-bit access is not possible.

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

11.2.6 Timer General Registers (TGR)

The timer general registers (TGR) function as 16-bit compare registers. The MMT has nine TGR registers, which are compared with the TCNT counter in the operating modes.

The TGR registers are initialized to H'FFFF by a power-on reset and in standby mode. They are not initialized in module standby mode. Only 16-bit access can be used on the TGR registers; 8-bit access is not possible.

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

11.2.7 Timer Dead Time Counters (TDCNT)

The timer dead time counters (TDCNT) are 16-bit read-only counters.

The TDCNT counters are initialized to H'0000 by a power-on reset and in standby mode. They are not initialized in module standby mode. Only 16-bit access can be used on the TDCNT counters; 8-bit access is not possible.

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

11.2.8 Timer Dead Time Data Register (TDDR)

The timer dead time data register (TDDR) is a 16-bit register that sets the positive phase and negative phase non-overlap time (dead time).

TDDR is initialized to H'FFFF by a power-on reset and in standby mode. It is not initialized in module standby mode. Only 16-bit access can be used on TDDR ; 8-bit access is not possible.

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

11.2.9 Timer Period Buffer Register (TPBR)

The timer period buffer register (TPBR) is a 16-bit register that functions as a buffer register for the TPDR register. A value of 1/2 the PWM carrier period should be set as the TPBR value. The TPBR value is transferred to the TPDR register at the transfer timing set in the TMDR register.

TPBR is initialized to H'FFFF by a power-on reset and in standby mode. It is not initialized in module standby mode. Only 16-bit access can be used on TPBR ; 8-bit access is not possible.

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

11.2.10 Timer Period Data Register (TPDR)

The timer period data register (TPDR) functions as a 16-bit compare register. In the operating modes, the TPDR register value is constantly compared with the TCNT counter value, and when they match the TCNT counter changes its count direction from up to down.

TPDR is initialized to H'FFFF by a power-on reset and in standby mode. It is not initialized in module standby mode. Only 16-bit access can be used on TPDR ; 8-bit access is not possible.

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

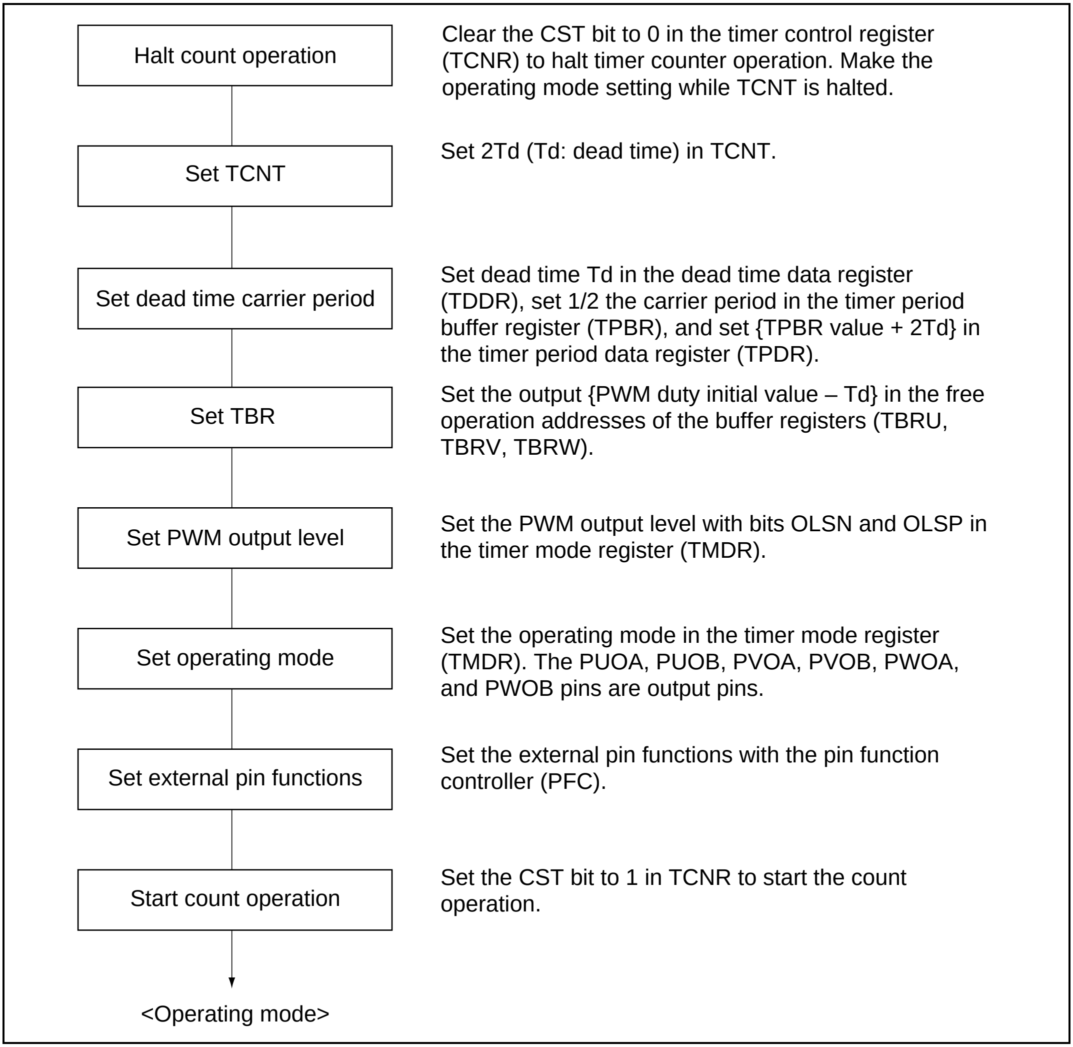
11.3 Operation

When the operating mode is selected, 3-phase PWM waveform output is performed with a non-overlap relationship between the positive and negative phases.

The PUOA, PUOB, PVOA, PVOB, PWOA, and PWOB pins are PWM output pins, and the PCIO pin functions as a toggle output synchronized with the PWM waveform, or as the counter clear signal input. The TCNT counter performs up- and down-count operations, while the TDCNT counters perform up-count operations.

11.3.1 Sample Setting Procedure

An example of the operating mode setting procedure is shown in figure 11.2.



Clear the CST bit to 0 in the timer control register (TCNR) to halt timer counter operation. Make the operating mode setting while TCNT is halted.

Set 2Td (Td: dead time) in TCNT.

Set dead time Td in the dead time data register (TDDR), set 1/2 the carrier period in the timer period buffer register (TPBR), and set {TPBR value + 2Td} in the timer period data register (TPDR).

Set the output {PWM duty initial value – Td} in the free operation addresses of the buffer registers (TBRU, TBRV, TBRW).

Set the PWM output level with bits OLSN and OLSP in the timer mode register (TMDR).

Set the operating mode in the timer mode register (TMDR). The PUOA, PUOB, PVOA, PVOB, PWOA, and PWOB pins are output pins.

Set the external pin functions with the pin function controller (PFC).

Set the CST bit to 1 in TCNR to start the count operation.

Figure 11.2 Sample Operating Mode Setting Procedure

11.3.2 Overview of Operation

Count Operation

Set 2Td (Td: value set in TDDR) as the initial value of the TCNT counter when the setting of the CST bit in TCNR is 0.

When the CST bit is set to 1, TCNT counts up to {value set in TPBR + 2Td}, and then starts counting down. When TCNT reaches 2Td, it starts counting up again, and continues in this way.

TCNT is constantly compared with TGRU, TGRV, and TGRW. In addition, it is compared with TGRUU, TGRVU, TGRWU, and TPDR when counting up, and with TGRUD, TGRVD, TGRWD, and 2Td when counting down.

TDCNT0 to TDCNT5 are read-only counters. It is not necessary to set their initial values.

TDCNT0, TDCNT2, and TDCNT4 start counting up at the falling edge of positive phase compare match output when TCNT is counting up, and when they match TDDR they are cleared to 0 and halt.

TDCNT1, TDCNT3, and TDCNT5 start counting up at the falling edge of negative phase compare match output when TCNT is counting up, and when they match TDDR they are cleared to 0 and halt.

TDCNT0 to TDCNT5 are compared with TDDR only while a count operation is in progress. No count operation is performed when the TDDR value is 0.

Figure 11.3 shows an example of the TCNT count operation.

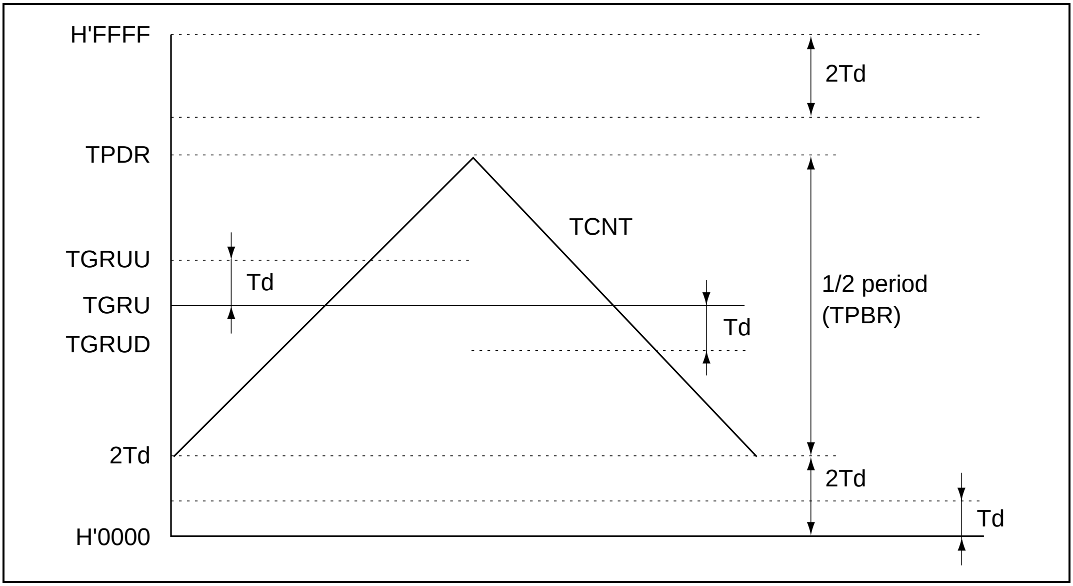


Figure 11.3 Example of TCNT Count Operation

Register Operation

In the operating modes, four buffer registers and ten compare registers are used.

The registers constantly compared with the TCNT counter are TGRU, TGRV, and TGRW. In addition, TGRUU, TGRVU, TGRWU, and TPDR are compared with TCNT when it is when counting up, and TGRUD, TGRVD, TGRWD are compared with TCNT when it is counting down. The buffer register for TPDR is TPBR; the buffer register for TGRUU, TGRU, and TGRUD is TBRU; the buffer register for TGRVU, TGRV, and TGRVD is TBRV; and the buffer register for TGRWU, TGRW, and TGRWD is TBRW.

To change compare register data, the new data should be written to the corresponding buffer register. The buffer registers can be read and written to at all times. Data written to TPBR and to the buffer operation addresses for and TBRU to TBRW is transferred at the timing specified by bits MD1 and MD0 in the timer mode register (TMDR). Data written to the free operation addresses for TBRU to TBRW is transferred immediately.

After data transfer is completed, the relationship between the compare registers and buffer registers is as follows.

TGRU (TGRV, TGRW) value = TBRU (TBRV, TBRW) value + Td (Td: value set in TDDR)
TGRUU (TGRVU, TGRWU) value = TBRU (TBRV, TBRW) value + 2Td
TGRUD (TGRVD, TGRWD) value = TBRU (TBRV, TBRW) value
TPDR value = TPBR value + 2Td

The values of TBRU to TBRW should always be set in the range H'0000 to H'FFFF – 2Td, and the value of TPBR should always be set in the range H'0000 to H'FFFF – 4Td.

Figure 11.4 shows examples of counter and register operations.

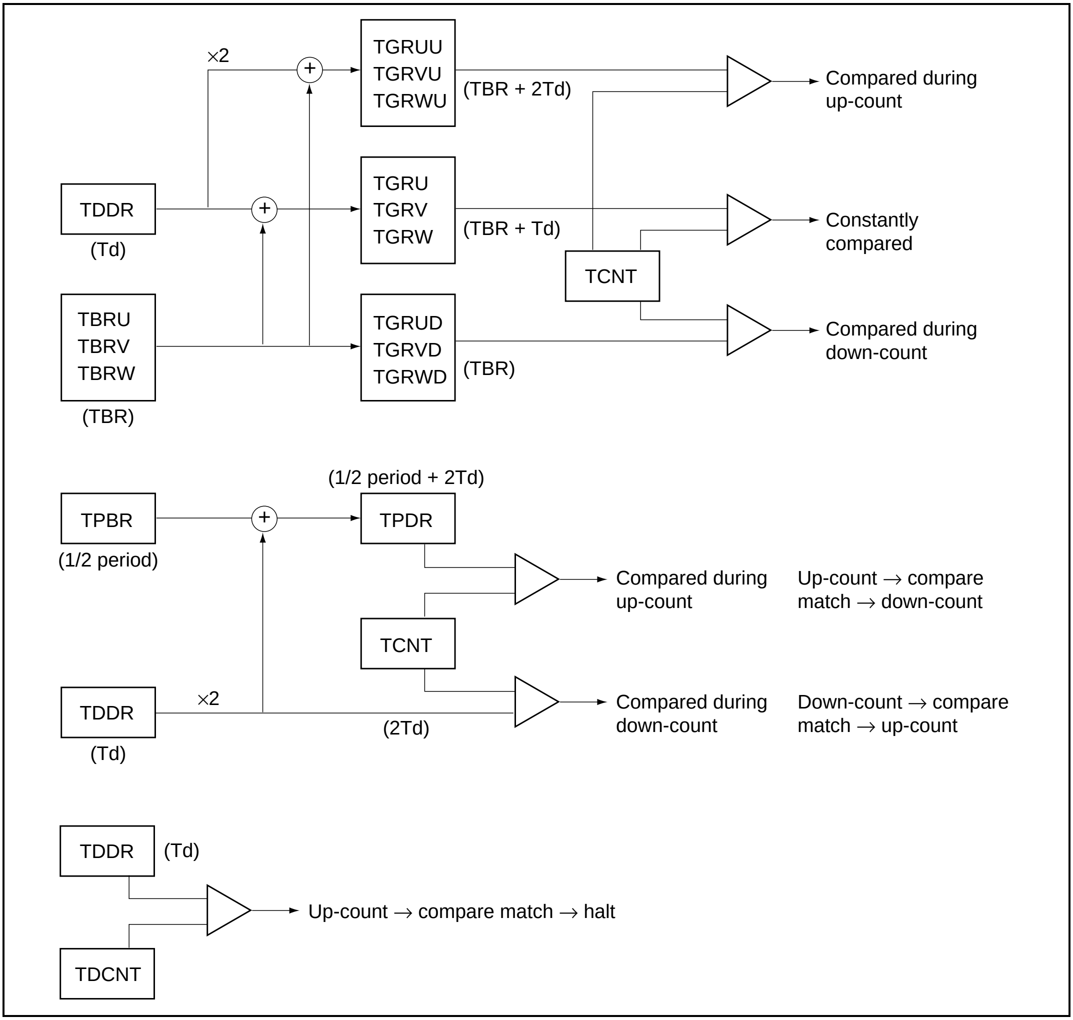


Figure 11.4 Examples of Counter and Register Operations

Initial Settings

In the operating modes, there are five registers that require initial settings.

Make the following register settings before setting the operating mode with bits MD1 and MD0 in the timer mode register (TMDR).

Set 1/2 the PWM carrier period in the timer period buffer register (TPBR), dead time Td in the timer dead time data register (TDDR) (when outputting an ideal waveform, Td = H'0000), and {TPBR value + 2Td} in the timer period data register (TPDR).

Set {PWM duty initial value – Td} in the free write operation addresses for TBRU to TBRW.

The values of TBRU to TBRW should always be set in the range H'0000 to H'FFFF – 2Td, and the value of TPBR should always be set in the range H'0000 to H'FFFF – 4Td.

PWM Output Active Level Setting

In the operating modes, the active level of PWM pulses is set with bits OLSN and OLSP in the timer mode register (TMDR).

The output level can be set for the three positive phases and the three negative phases of 6-phase output. The operating mode must be exited before setting or changing the output level.

Dead time Setting

In the operating modes, PWM pulses are output with a non-overlap relationship between the positive and negative phases. This non-overlap time is known as the dead time. The non-overlap time is set in the timer dead time data register (TDDR). The dead time generation waveform is generated by comparing the value set in TDDR with the timer dead time counters (TDCNT) for each phase. The operating mode must be exited before changing the contents of TDDR.

PWM Period Setting

In the operating modes, 1/2 the PWM pulse period is set in the TPBR register. The TPBR value should always be set in the range H'0000 to H'FFFF – 4Td. The value set in TPBR is transferred to TPDR at the timing selected with bits MD1 and MD0 in the timer mode register (TMDR). After the transfer, the value in TPDR is {TPBR value + 2Td}.

The new PWM period is effective from the next period when data updating is performed at the TCNT counter crest, and from the same period when data updating is performed at the trough.

Register Updating

In the operating modes, buffer registers are used to update compare register data. Update data can be written to a buffer register at all times. The buffer register value is transferred to the compare register at the timing set by bits MD1 and MD0 in the timer mode register (TMDR) (except in the case of a write to the free operation address for TBRU to TBRW, in which case the value is transferred to the compare register immediately).

Initial Output in Operating Modes

The initial output in the operating modes is determined by the initial value of TBRU to TBRW.

Table 11.4 shows the relationship between the initial value of TBRU to TBRW and the initial output.

Table 11.4 Initial Values of TBRU to TBRW and Initial Output

	Initial Output	
Initial Value of TBRU to TBRW	**OLSP = 1, OLSN = 1**	**OLSP = 0, OLSN = 0**
TBR = H'0000	Positive phase: 1 Negative phase: 0	Positive phase: 0 Negative phase: 1
H'0000 < TBR ≤ Td	Positive phase: 0 Negative phase: 0	Positive phase: 1 Negative phase: 1
Td < TBR ≤ H'FFFF – 2Td	Positive phase: 0 Negative phase: 1	Positive phase: 1 Negative phase: 0

PWM Output Generation in Operating Modes

In the operating modes, 3-phase PWM waveform output is performed with a non-overlap relationship between the positive and negative phases. This non-overlap time is called the dead time.

The PWM waveform is generated from an output generation waveform generated by ANDing the compare output waveform with the dead time generation waveform. Waveform generation for one phase (the U-phase) is shown here. The V-phase and W-phase waveforms are generated in the same way.

Compare Output Waveform: The compare output waveform is generated by comparing the values in the TCNT counter and the TGR registers.

For compare output waveform U phase A (CMOUA), 0 is output if TGRUU > TCNT in the T1 interval (when TCNT is counting up), and 1 is output if TGRUU ≤ TCNT. In the T2 interval (when TCNT is counting down), 0 is output if TGRU > TCNT, and 1 is output if TGRU ≤ TCNT.

For compare output waveform U phase B (CMOUB), 1 is output if TGRU > TCNT in the T1 interval, and 0 is output if TGRU ≤ TCNT. In the T2 interval, 1 is output if TGRUD > TCNT, and 0 is output if TGRUD ≤ TCNT.

Dead Time Generation Waveform: For dead time generation waveform U phase A (DTGUA) and B (DTGUB), 1 is output as the initial value.

TDCNT0 starts counting at the falling edge of CMOUA. DTGUA outputs 0 while TDCNT0 is counting, and 1 otherwise.

TDCNT1 starts counting at the falling edge of CMOUB. DTGUB outputs 0 while TDCNT1 is counting, and 1 otherwise.

Output Generation Waveform: Output generation waveform U phase A (OGUA) is generated by ANDing CMOUA and DTGUB, and output generation waveform U phase B (OGUB) is generated by ANDing CMOUB and DTGUA.

PWM Waveform: The PWM waveform is generated by converting the output generation waveform to the output level set in bits OLSN and OLSP in the timer mode register (TMDR).

Figure 11.5 shows an example of PWM waveform generation (operating mode 3, OLSN = 1, OLSP = 1).

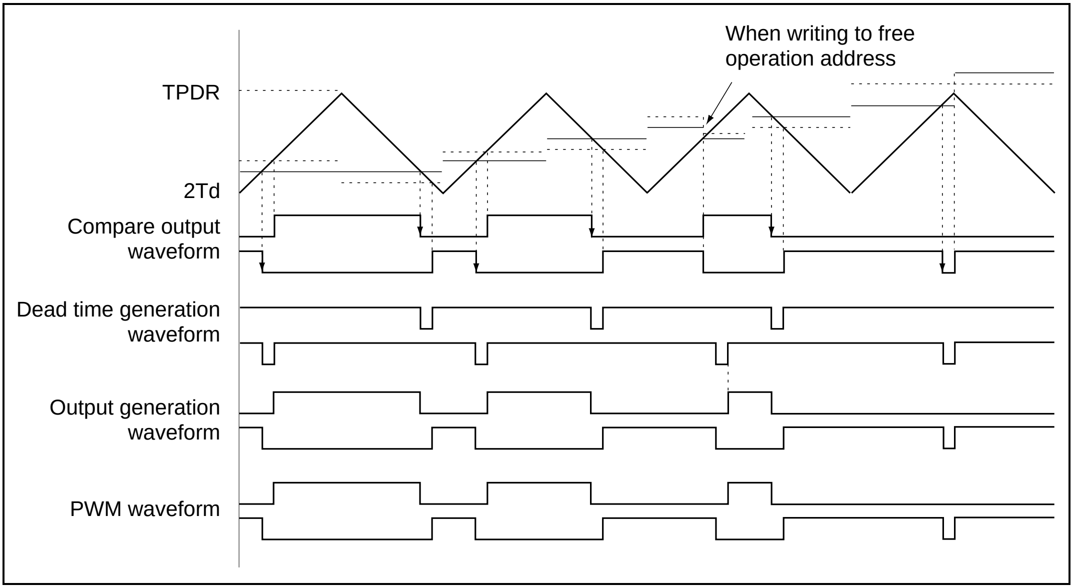

When writing to free operation address
TPDR
2Td
Compare output waveform
Dead time generation waveform
Output generation waveform
PWM waveform

Figure 11.5 Example of PWM Waveform Generation

0% to 100% Duty Output

In the operating modes, PWM waveforms with any duty from 0% to 100% can be output. The output PWM duty is set by means of the buffer registers (TBRU to TBRW).

100% duty output is performed when the buffer register (TBRU to TBRW) value is set to H'0000. The waveform in this case has the positive phase in the 100% on state. 0% duty output is performed when a value greater than the TPDR value is set as the buffer register (TBRU to TBRW) value. The waveform in this case has the positive phase in the 100% off state.

External Counter Clear Function

In the operating modes, the TCNT counter can be cleared from an external source. When using the counter clear function, the PCIO pin function should be set to input with the pin function controller (PFC).

At the falling edge of PCIO, the TCNT counter is cleared to 2Td (initial set value), counts up until it reaches the TPDR value, and then starts counting down. When the count reaches 2Td, TCNT starts counting up again, and this sequence is repeated. An example of counter clearing is shown in figure 11.6.

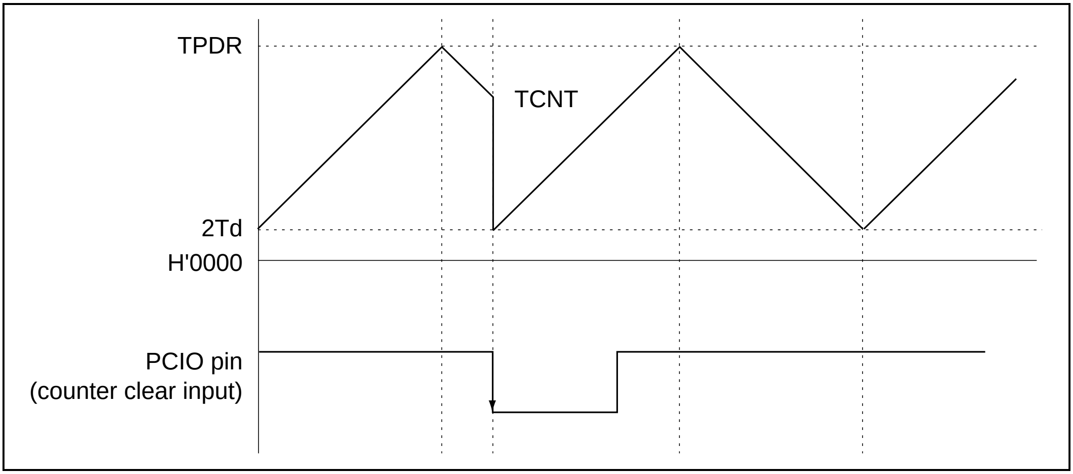


Figure 11.6 Example of TCNT Counter Clearing

Toggle Output Synchronized with PWM Period

In the operating modes, output can be toggled in synchronization with the PWM carrier period. When outputting the PWM period, the PCIO pin function should be set to output with the pin function controller (PFC). An example of the toggle output waveform is shown in figure 11.7.

PWM output is toggled according to the TCNT count direction. The toggle output pin is PCIO. PCIO outputs 1 when TCNT is counting up, and 0 when counting down.

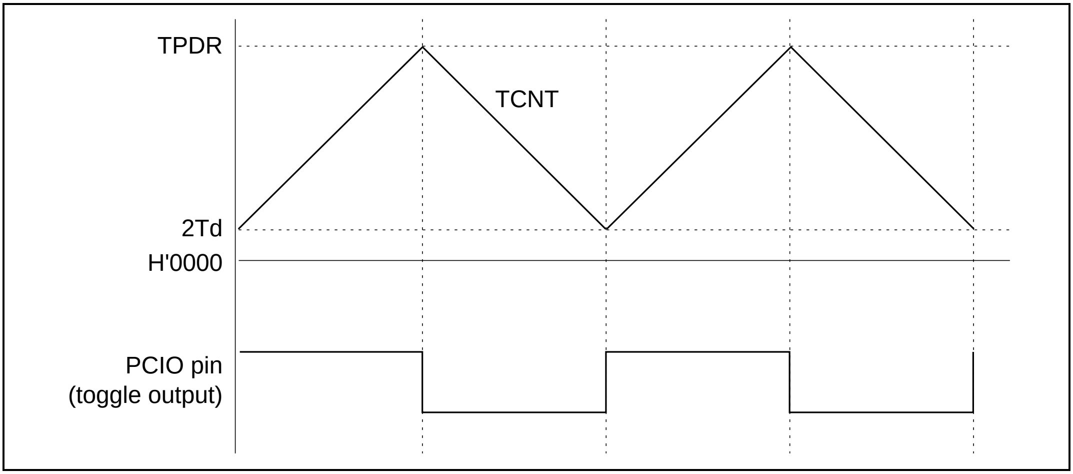


Figure 11.7 Example of Toggle Output Waveform Synchronized with PWM Period

A/D Conversion Start Request Setting

An A/D conversion start request can be made using a compare match between TCNT and TPDR or between TCNT and 2Td. When a start request using a compare match between TCNT and TPDR is set, A/D conversion can be started in the middle of a PWM pulse (at the TCNT counter crest). When a start request using a compare match between TCNT and 2Td is set, A/D conversion can be started at the edge of a PWM pulse (at the TCNT counter trough).

A/D conversion start requests can be set by setting the TTGE bit to 1 in the timer control register (TCNR).

11.3.3 Output Protection Functions

Operating mode output has the following protection functions.

- Halting PWM output by external signal

  The 6-phase PWM output pins can be placed in the high-impedance state automatically by inputting a specified external signal. There are four external signal input pins. For details, see section 11.7, Port Output Enable (POE).

- Halting PWM output when oscillation stops

  The 6-phase PWM output pins are placed in the high-impedance state automatically when stoppage of the clock input to the SH7065 is detected. However, pin states are not guaranteed when the clock is restarted.

11.4 Interrupts

11.4.1 Compare Match Interrupts

When the TGFM (TGFN) flag is set to 1 in the timer status register (TSR) by a compare match between TCNT and the TPDR register (2Td), if the setting of the TGIEM (TGIEN) bit in the timer control register (TCNR) is 1 an interrupt is requested. The interrupt request is cleared by clearing the TGF flag to 0.

11.4.2 DMA Controller Activation

The on-chip DMA controller can be activated by a compare match between TCNT and TPDR or between TCNT and 2Td.

11.4.3 A/D Converter Activation

The on-chip A/D converter can be activated by a compare match between TCNT and TPDR or between TCNT and 2Td. When the TGF flag is set to 1 in the timer status register (TSR) by either of these compare matches, an A/D conversion start request is sent to the A/D converter. If the MMT start trigger has been selected on the A/D converter side, A/D conversion begins.

11.5 Operation Timing

11.5.1 Input/Output Timing

TCNT and TDCNT Count Timing

Figure 11.8 shows the TCNT and TDCNT count timing.

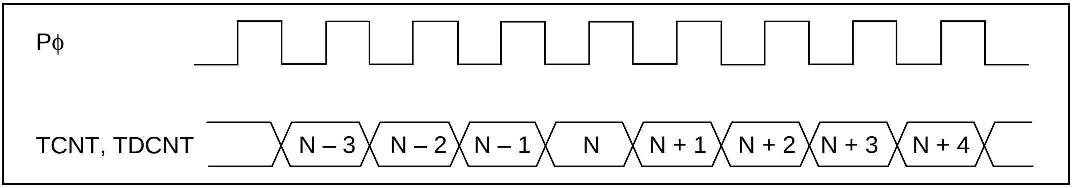


Figure 11.8 TCNT and TDCNT Count Timing

TCNT Counter Clearing Timing

Figure 11.9 shows the timing of TCNT counter clearing by an external signal.

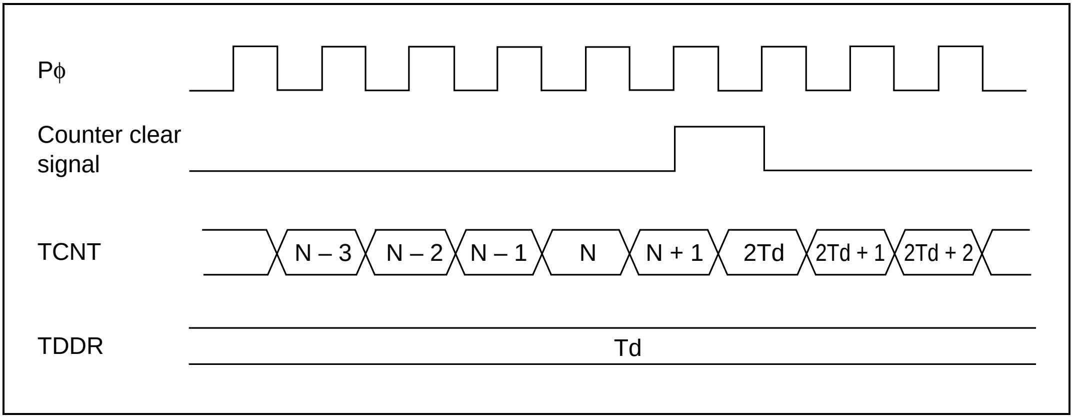


Figure 11.9 TCNT Counter Clearing Timing

TDCNT Operation Timing

Figure 11.10 shows the TDCNT operation timing.

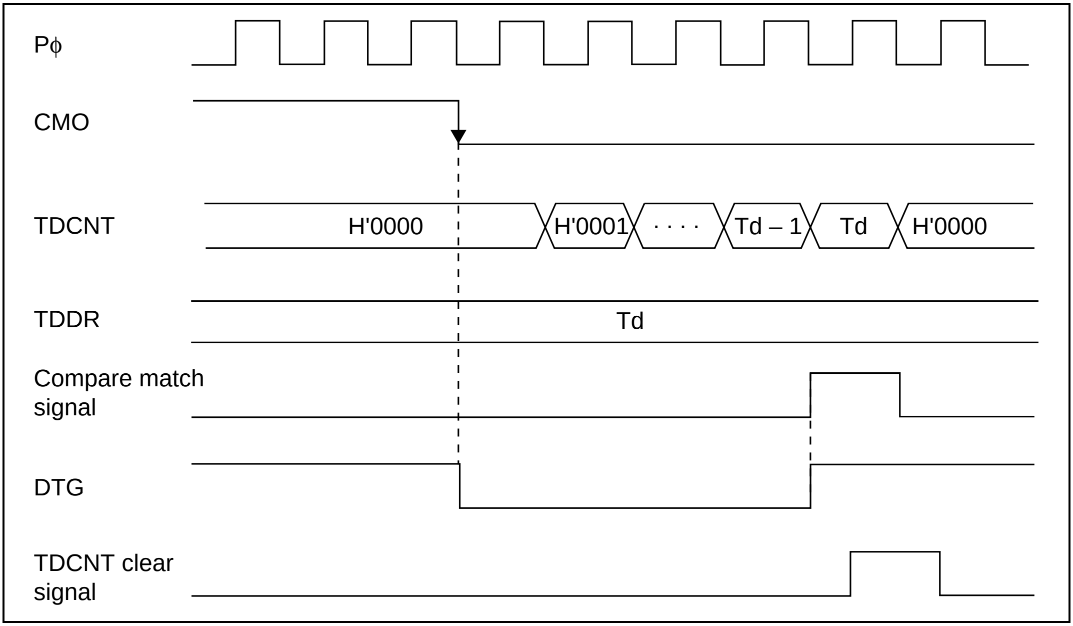


Figure 11.10 TDCNT Operation Timing

Buffer Operation Timing

Figure 11.11 shows the compare match buffer operation timing.

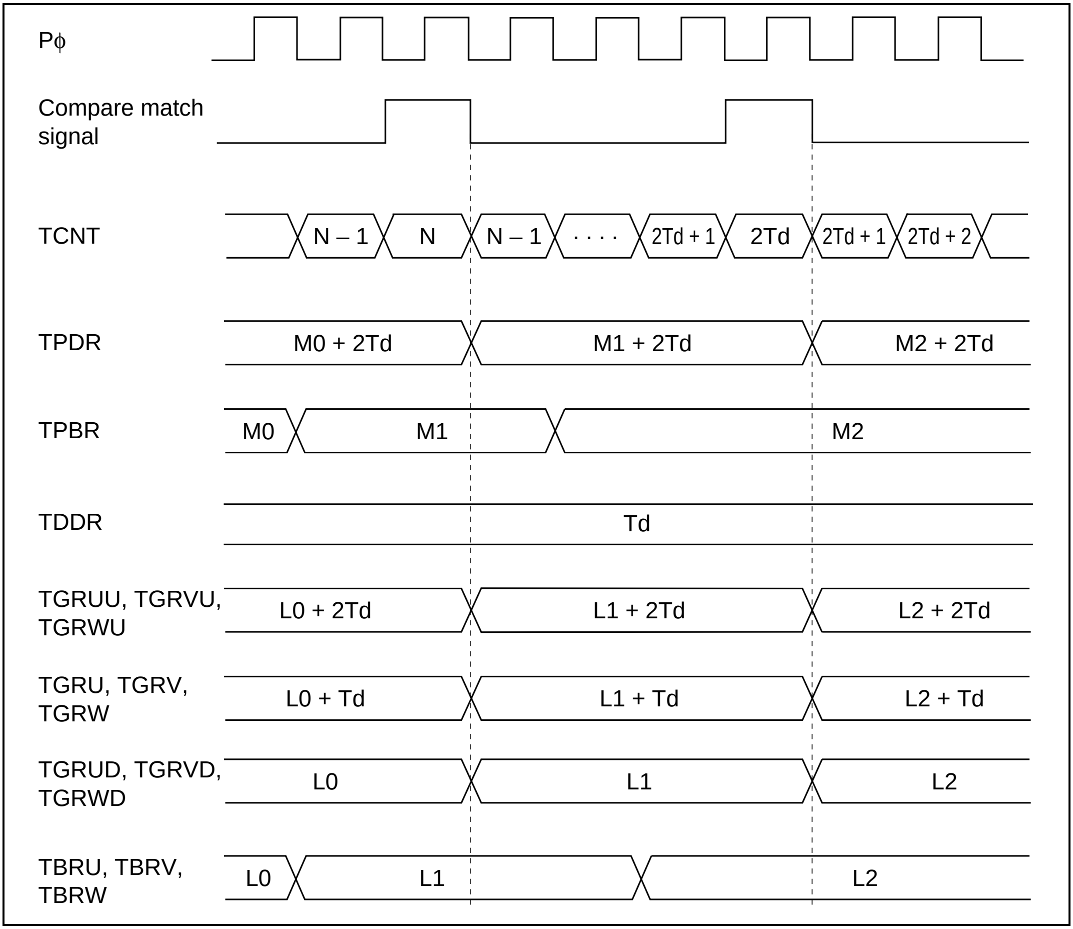


Figure 11.11 Buffer Operation Timing

11.5.2 Interrupt Signal Timing

Timing of TGF Flag Setting by Compare Match

Figure 11.12 shows the timing of setting of the TGF flag in the timer status register (TSR) by a compare match between TCNT and TPDR, and the timing of the TGI interrupt request signal. The timing is the same for a compare match between TCNT and 2Td.

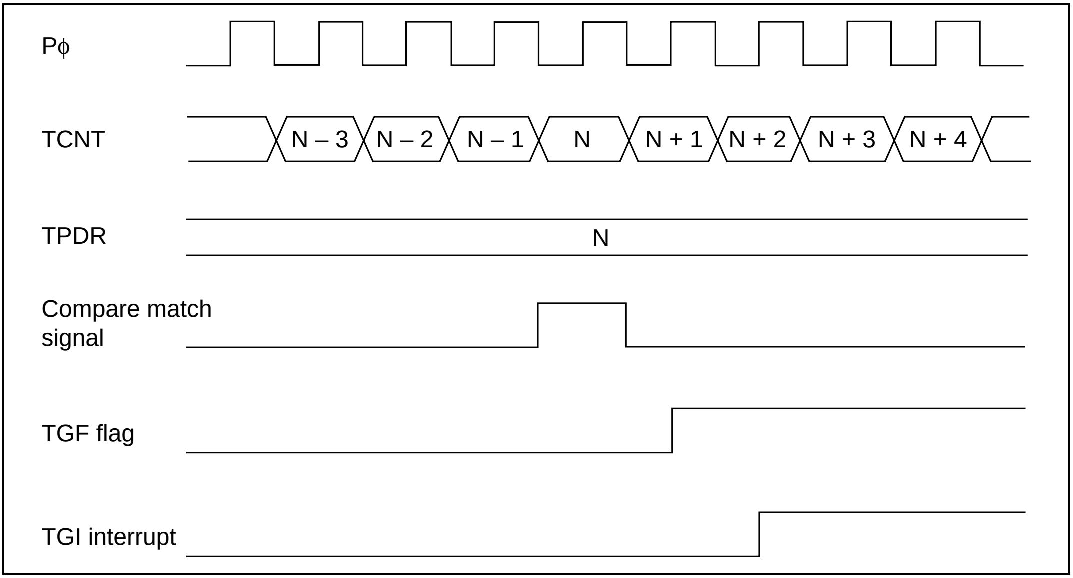


Figure 11.12 TGI Interrupt Timing

Status Flag Clearing Timing

A status flag is cleared when the CPU reads 1 from the flag, then writes 0 to it. When the DMA controller is activated, the flag is cleared automatically. Figure 11.13 shows the timing of status flag clearing by the CPU, and figure 11.14 shows the timing of status flag clearing by the DMA controller.

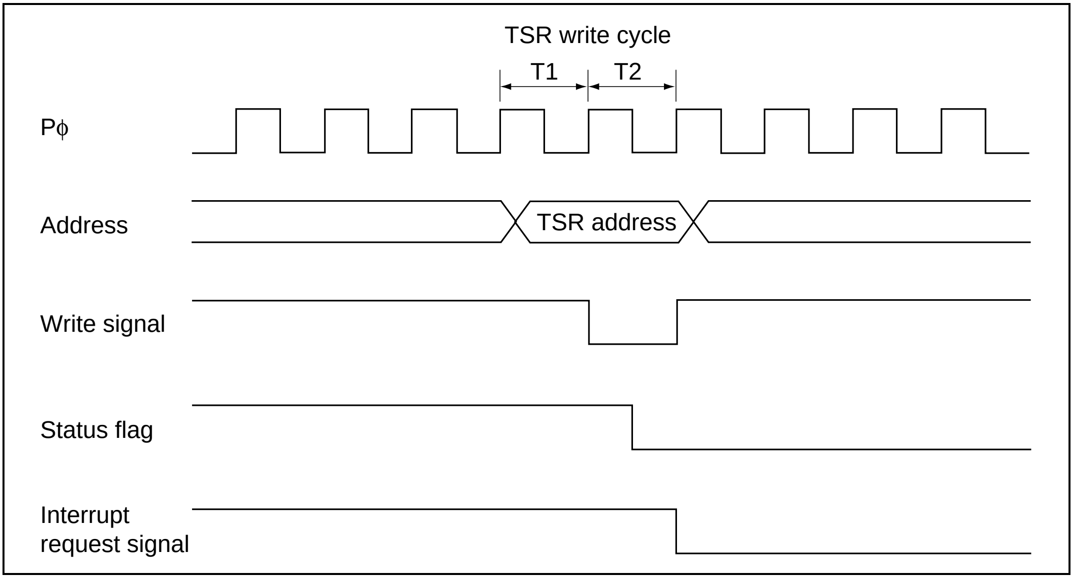


Figure 11.13 Timing of Status Flag Clearing by CPU

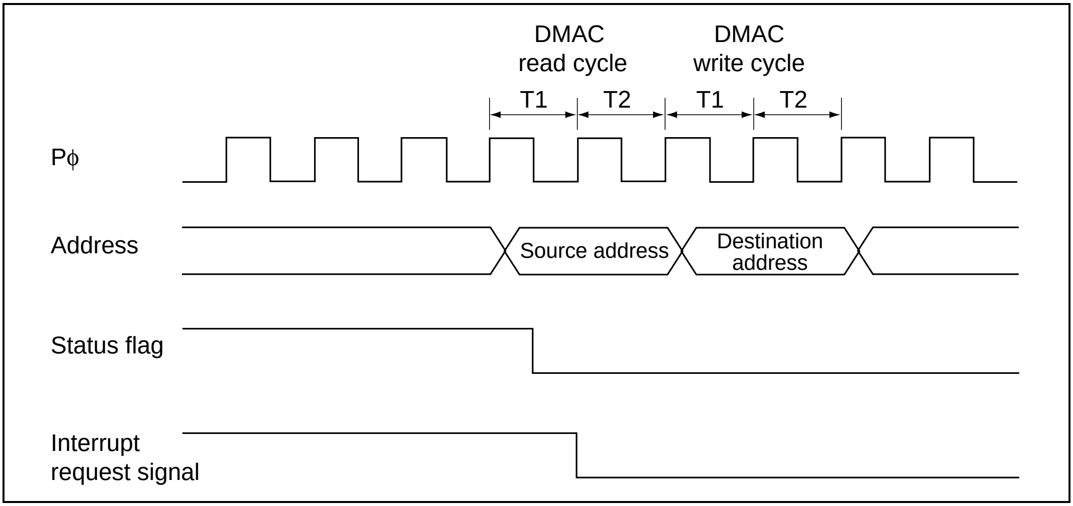


Figure 11.14 Timing of Status Flag Clearing by DMA Controller

11.6 Usage Notes

Note that the kinds of operation and contention described below occur during MMT operation.

Contention between Buffer Register Write and Compare Match

If a compare match occurs in the T2 state of a buffer register (TBRU, TBRV, TBRW, or TPBR) write cycle, data is transferred from the buffer register to the compare register (TGR or TPDR) by means of a buffer operation. The data transferred is the buffer register write data.

Figure 11.15 shows the timing in this case.

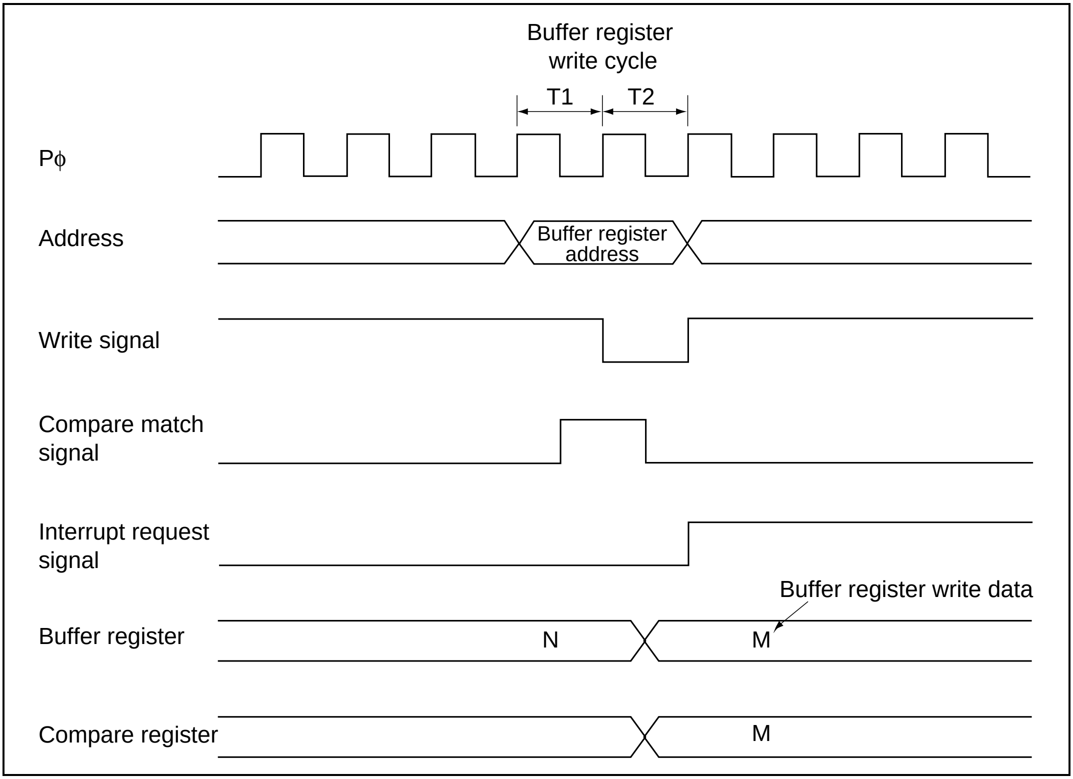


Figure 11.15 Contention between Buffer Register Write and Compare Match

Contention between Compare Register Write and Compare Match

If a compare match occurs in the T2 state of a compare register (TGR or TPDR) write cycle, the compare register write is not performed, and data is transferred from the buffer register (TBRU, TBRV, TBRW, or TPBR) to the compare register (TGR or TPDR) by means of a buffer operation.

Figure 11.16 shows the timing in this case.

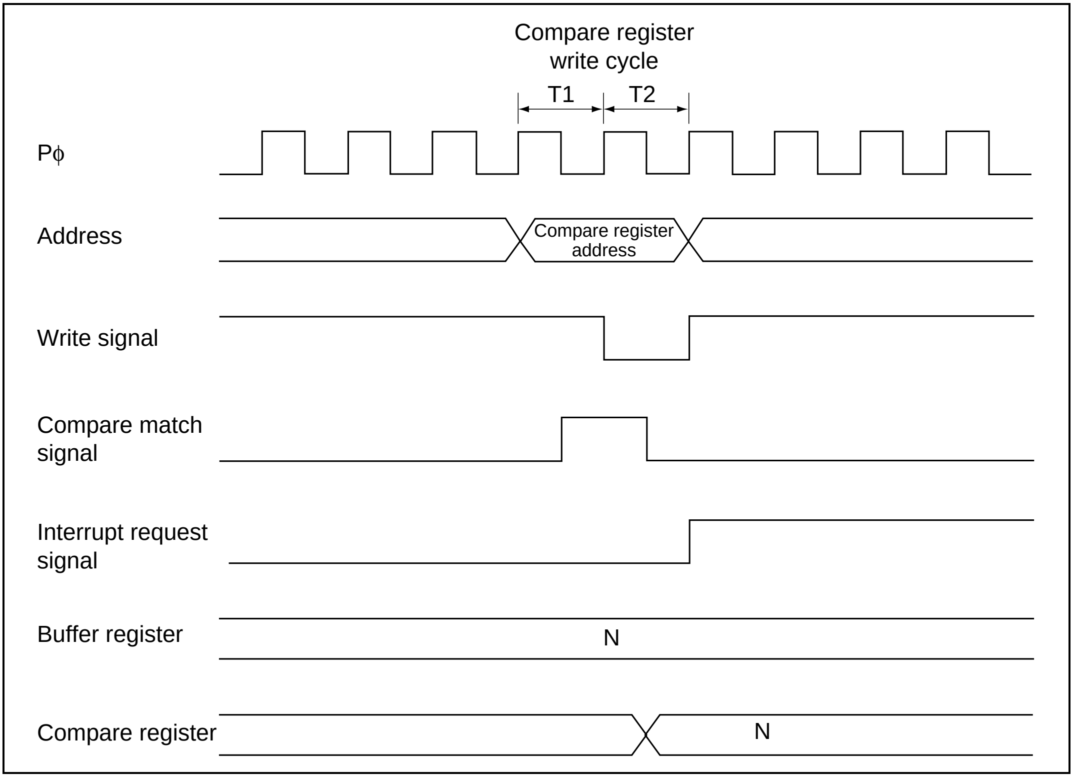


Figure 11.16 Contention between Compare Register Write and Compare Match

Pay Attention to the Notices Below, When a Value Is Written into the Timer General Register U (TGRU), Timer General Register V (TGRV), Timer General Register W (TGRW), and in Case of Written into Free Operation Address (*):

- In case of counting up: Do not write a value {Previous value of TGRU + Td} into TGRU.
- In case of counting down: Do not write a value {Previous value of TGRU – Td} into TGRU.

In the same manner to TGRV and TGRW. When a value {Previous value of TGRU + Td} is written (in case of counting down {Previous value of TGRU – Td}), the output of PUOA/PUOB, PVOA/PVOB, PWOA/PWOB (corresponding to U, V, W phase) may not be output for 1 cycle. Figure 11.17 shows the error case. When writing into the buffer operation address, these notes are not relevant.

Note: * When addresses, H'FFFF049C, H'FFFF04AC, H'FFFF04BC are used as register address for TBRU, TBRV, TBRW, respectively.

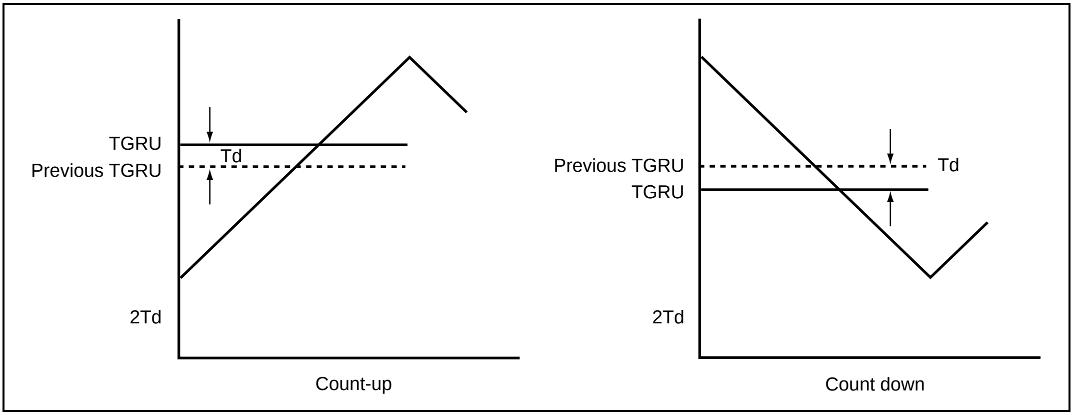


Figure 11.17 Writing into Timer General Registers (When One Cycle Is Not Output)

Writing Operation into Timer Period Data Register (TPDR) and Timer Dead Time Data Register (TDDR) When MMT Is Operating:

- Do not revise TPDR register when MMT is operating. Always use a buffer-write operation through TPBR register.
- Do not revise TDDR register once an operation of MMT is invoked. When TDDR is revised, a wave may not be output for as much as 1 cycle (full count period of 16 bits in TDCNT), because a value cannot be written into TDCNT, which is compared to a value set in TDDR.

Notes on Halting TCNT Counter Operation: If TCNT counter operation is halted, a PCM waveform may be output with dead time (non-overlap time) shorter than the value set in the timer dead time register (MMT_TDDR) or no dead time at all (value of 0). To prevent this, use one of the following methods.

- Set the CST bit in the timer control register (TCNR) to 1 and do not clear it to 0 after MMT counter operation starts. If the CST bit is cleared to 0, do not set it to 1 again.
- When setting, clearing, and then resetting the CST bit, use the following procedure for clearing and then resetting.

  (1) Use the pin function controller (PFC) to set the PWM output pin as a general input port.

  (2) Set the free operation addresses for all the buffer registers (TBRU, TBRV, and TBRW) to H'0000.

  (3) After the specified dead time duration has elapsed, set TCNR to H'00 and clear the CST bit to 0.

  (4) Once again, set the CST bit to 1.
- When setting, clearing, and then resetting the CST bit, use the following procedure for clearing and then resetting.

  (1) Clear the CST bit in TCNR to 0 to halt counter operation.

  (2) Use the pin function controller to set the PWM output pin as a general input port.

  (3) Clear the MSTP9 bit in module standby control register 1 (MSTCR1) to 0 to transition to module standby mode, and initialize the internal status of MMT.

  (4) Immediately set the MSTP9 bit to 1 to transition back from module standby mode, and reset MMT and the pin to their initial settings.

  (5) Set the CST bit in TCNR to 1 to restart counter operation.

11.7 Port Output Enable (POE)

11.7.1 Overview

The port output enable (POE) circuit enables the MMT's 6-phase output pins (POUA, POUB, POVA, POVB, POWA, and POWB) to be placed in the high-impedance state by varying the input at pins $\overline{\text{POE0}}$ to $\overline{\text{POE3}}$. An interrupt can also be requested at the same time.

Separately from this function, the MMT's 6-phase output pins go to the high-impedance state when the oscillator halts. For details, see section 4, Clock Pulse Generator (CPG) and Power-Down Modes.

Features

The POE circuit has the following features:

- Falling edge, Pφ/8 × 16 times, Pφ/16 × 16 times, or Pφ/128 × 16 times low-level sampling settings can be made for each of input pins $\overline{\text{POE0}}$ to $\overline{\text{POE3}}$.
- The MMT's 6-phase output pins can be placed in the high-impedance state on sampling of a falling edge or low level at pins $\overline{\text{POE0}}$ to $\overline{\text{POE3}}$.
- An interrupt request can be initiated by input level sampling.

Block Diagram

Figure 11.17 shows a block diagram of the POE circuit.

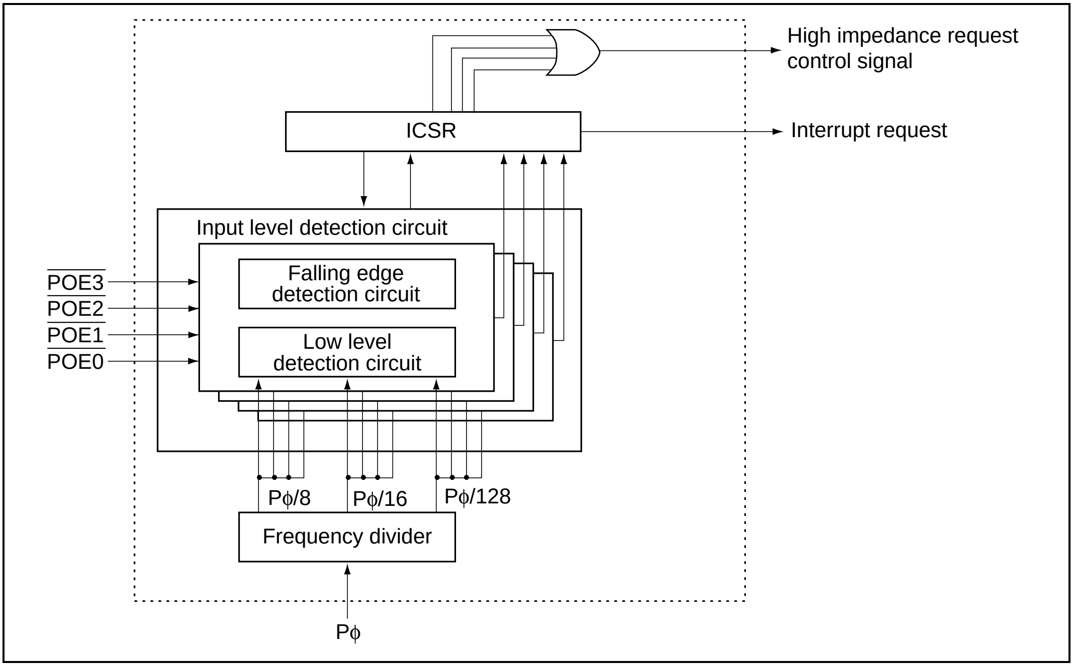


Figure 11.17 Block Diagram of POE

Pin Configuration

Table 11.5 shows the pin configuration of the POE circuit.

Table 11.5 POE Pins

Name	Abbreviation	I/O	Function
Port output enable input pins	$\overline{\text{POE0}}$–$\overline{\text{POE3}}$	Input	Input request signals for placing MMT's 6-phase output pins in high-impedance state

Register Configuration

The POE circuit has the single register summarized in table 11.6. The input level control/status register (ICSR) controls detection of the input signals on pins $\overline{\mathrm{POE0}}$ to $\overline{\mathrm{POE3}}$, and interrupt requests.

Table 11.6 POE Register

Name	Abbre-viation	R/W	Initial Value	Address	Access Size
Input level control/status register	ICSR	R/(W)*	H'0000	H'FFFF 04E0 H'FFFF 04E1	8, 16, 32

Note: * Bits 15 to 12 can only be written with 0, to clear the flags.

11.7.2 Register Description

Input Level Control/Status Register (ICSR)

The input level control/status register (ICSR) is a 16-bit readable/writable register that selects the input mode for pins $\overline{\mathrm{POE0}}$ to $\overline{\mathrm{POE3}}$, controls enabling or disabling of interrupts, and gives status indications.

ICSR is initialized to H'0000 by an external power-on reset, but is not initialized, and retains its prior data, in a WDT reset, in standby mode, and in sleep mode.

Bit:	15	14	13	12	11	10	9	8
	POE3F	POE2F	POE1F	POE0F	—	—	—	PIE
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	—	—	—	R/W

Bit:	7	6	5	4	3	2	1	0
	POE3 M1	POE3 M0	POE2 M1	POE2 M0	POE1 M1	POE1 M0	POE0 M1	POE0 M0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: * Only 0 can be written, for flag clearing.

Bit 15—POE3 Flag (POE3F): Indicates that a high impedance request has been input to the $\overline{\text{POE3}}$ pin.

Bit 15: POE3F	Description
0	[Clearing condition] (Initial value) When 0 is written to POE3F after reading POE3F = 1
1	[Setting condition] When the input set by bits 7 and 6 of ICSR occurs at the $\overline{\text{POE3}}$ pin

Bit 14—POE2 Flag (POE2F): Indicates that a high impedance request has been input to the $\overline{\text{POE2}}$ pin.

Bit 14: POE2F	Description
0	[Clearing condition] (Initial value) When 0 is written to POE2F after reading POE2F = 1
1	[Setting condition] When the input set by bits 5 and 4 of ICSR occurs at the $\overline{\text{POE2}}$ pin

Bit 13—POE1 Flag (POE1F): Indicates that a high impedance request has been input to the $\overline{\text{POE1}}$ pin.

Bit 13: POE1F	Description
0	[Clearing condition] (Initial value) When 0 is written to POE1F after reading POE1F = 1
1	[Setting condition] When the input set by bits 3 and 2 of ICSR occurs at the $\overline{\text{POE1}}$ pin

Bit 12—POE0 Flag (POE0F): Indicates that a high impedance request has been input to the $\overline{\text{POE0}}$ pin.

Bit 12: POE0F	Description
0	[Clearing condition] (Initial value) When 0 is written to POE0F after reading POE0F = 1
1	[Setting condition] When the input set by bits 1 and 0 of ICSR occurs at the $\overline{\text{POE0}}$ pin

Bits 11 to 9—Reserved: These bits are always read as 0 and should only be written with 0.

Bit 8—Port Interrupt Enable (PIE): Enables or disables an interrupt request when 1 is set in any of bits POE0F to POE3F in ICSR.

Bit 8: PIE	Description
0	Interrupt request disabled (Initial value)
1	Interrupt request enabled

Bits 7 and 6—POE3 Mode 1 and 0 (POE3M1, POE3M0): These bits select the input mode of the $\overline{\text{POE3}}$ pin.

Bit 7: POE3M1	Bit 6: POE3M0	Description
0	0	Request accepted at falling edge of $\overline{\text{POE3}}$ input (Initial value)
	1	$\overline{\text{POE3}}$ input is sampled for low level 16 times every Pϕ/8 clock, and request is accepted when all samples are low level
1	0	$\overline{\text{POE3}}$ input is sampled for low level 16 times every Pϕ/16 clock, and request is accepted when all samples are low level
	1	$\overline{\text{POE3}}$ input is sampled for low level 16 times every Pϕ/128 clock, and request is accepted when all samples are low level

Bits 5 and 4—POE2 Mode 1 and 0 (POE2M1, POE2M0): These bits select the input mode of the $\overline{\text{POE2}}$ pin.

Bit 5: POE2M1	Bit 4: POE2M0	Description
0	0	Request accepted at falling edge of $\overline{\text{POE2}}$ input (Initial value)
	1	$\overline{\text{POE2}}$ input is sampled for low level 16 times every Pϕ/8 clock, and request is accepted when all samples are low level
1	0	$\overline{\text{POE2}}$ input is sampled for low level 16 times every Pϕ/16 clock, and request is accepted when all samples are low level
	1	$\overline{\text{POE2}}$ input is sampled for low level 16 times every Pϕ/128 clock, and request is accepted when all samples are low level

Bits 3 and 2—POE1 Mode 1 and 0 (POE1M1, POE1M0): These bits select the input mode of the $\overline{\text{POE1}}$ pin.

Bit 3: POE1M1	Bit 2: POE1M0	Description
0	0	Request accepted at falling edge of $\overline{\text{POE1}}$ input (Initial value)
	1	$\overline{\text{POE1}}$ input is sampled for low level 16 times every Pϕ/8 clock, and request is accepted when all samples are low level
1	0	$\overline{\text{POE1}}$ input is sampled for low level 16 times every Pϕ/16 clock, and request is accepted when all samples are low level
	1	$\overline{\text{POE1}}$ input is sampled for low level 16 times every Pϕ/128 clock, and request is accepted when all samples are low level

Bits 1 and 0—POE0 Mode 1 and 0 (POE0M1, POE0M0): These bits select the input mode of the $\overline{\text{POE0}}$ pin.

Bit 1: POE0M1	Bit 0: POE0M0	Description
0	0	Request accepted at falling edge of $\overline{\text{POE0}}$ input (Initial value)
	1	$\overline{\text{POE0}}$ input is sampled for low level 16 times every Pϕ/8 clock, and request is accepted when all samples are low level
1	0	$\overline{\text{POE0}}$ input is sampled for low level 16 times every Pϕ/16 clock, and request is accepted when all samples are low level
	1	$\overline{\text{POE0}}$ input is sampled for low level 16 times every Pϕ/128 clock, and request is accepted when all samples are low level

11.7.3 Operation

Input Level Detection

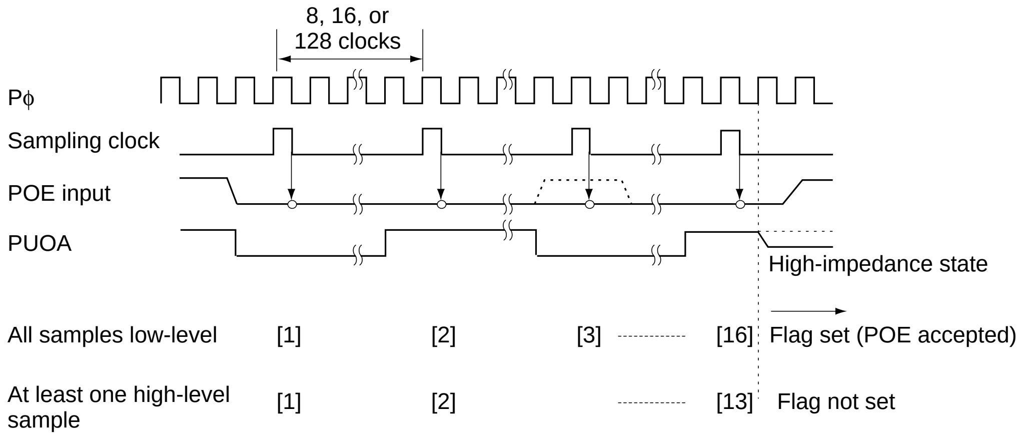
When the input condition set in ICSR occurs on any one of the $\overline{\text{POE}}$ pins, the MMT's 6-phase output pins go to the high-impedance state.

- Pins placed in high-impedance state (MMT 6-phase output pins)

  The 12 MMT (motor management timer) pins PD26/D26/PWOB/RxD3, PD25/D25/PVOB/TxD3, PD24/D24/PUOB/SCK3, PD22/D22/PWOA/SCK0, PD21/D21/PVOA/$\overline{\text{IRQ7}}$, PD20/D20/PUOA/$\overline{\text{IRQ6}}$, PE23/$\overline{\text{IRQ7}}$/PWOB, PE22/$\overline{\text{IRQ6}}$/PVOB, PE21/$\overline{\text{IRQ5}}$/PUOB, PE19/$\overline{\text{IRQ3}}$/PWOA, PE18/$\overline{\text{IRQ2}}$/PVOA, and PE17/$\overline{\text{IRQ1}}$/PUOA/SCK0 are placed in the high-impedance state.

Falling edge detection: When a translation from high-to low-level input occurs on a $\overline{\text{POE}}$ pin.

Low level detection: Figure 11.18 shows the low level detection operation. Low level sampling is performed 16 times in succession using the sampling clock set in ICSR. The input is not accepted if a high level is detected even once among these samples.
The timing of entry of the MMT's 6-phase output pins into the high-impedance state from the sampling clock is the same for falling edge detection and low level detection.



Note: The other MMT 6-phase output pins also go to the high-impedance state at the same timing.

Figure 11.18 Low Level Detection Operation

Exiting High-Impedance State

MMT 6-phase output pins that have entered the high-impedance state as the result of input level detection are released from this state by restoring them to their initial states by means of a power-on reset, or by clearing all the POE flags in ICSR (POE0F to POE3F: bits 12 to 15).

Section 12 Compare Match Timer (CMT)

12.1 Overview

The SH7065 has an on-chip compare match timer (CMT) consisting of a two-channel 16-bit timer. The CMT has a 16-bit counter, and can generate interrupts at set intervals.

12.1.1 Features

The CMT has the following features:

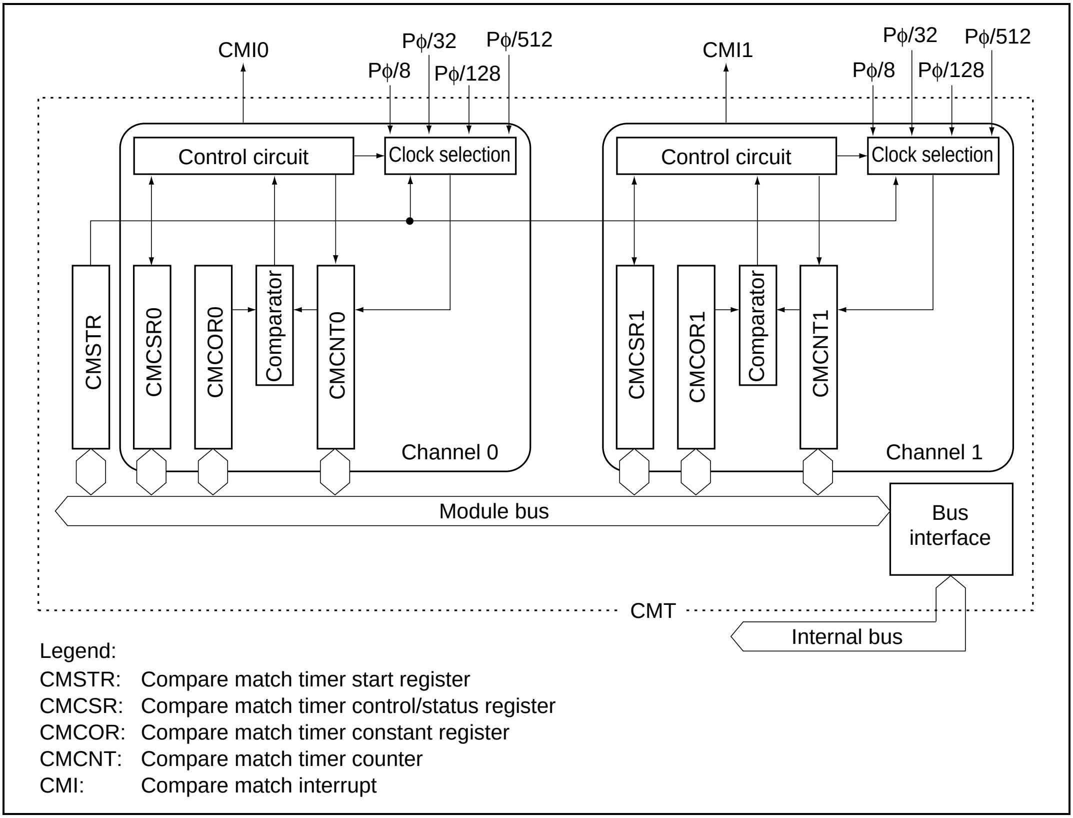
- Choice of four counter input clocks

  Any of four internal clocks (Pϕ/8, Pϕ/32, Pϕ/128, Pϕ/512) can be selected independently for each channel (where Pϕ is the clock input to the CMT). The CMT's input clock is obtained by frequency division of an external clock.
- Interrupt sources

  A compare match interrupt can be requested independently for each channel.

12.1.2 Block Diagram

Figure 12.1 shows a block diagram of the CMT.



Legend:

CMSTR: Compare match timer start register

CMCSR: Compare match timer control/status register

CMCOR: Compare match timer constant register

CMCNT: Compare match timer counter

CMI: Compare match interrupt

Figure 12.1 Block Diagram of CMT

12.1.3 Register Configuration

Table 12.1 summarizes the CMT registers.

Table 12.1 CMT Registers

Channel	Name	Abbre-viation	R/W	Initial Value	Address	Access Size
Both	Compare match timer start register	CMSTR	R/W	H'0000	H'FFFF04C0	8, 16, 32
0	Compare match timer control/status register 0	CMCSR0	R/(W)*	H'0000	H'FFFF04C2	8, 16, 32
	Compare match timer counter 0	CMCNT0	R/W	H'0000	H'FFFF04C4	8, 16, 32
	Compare match timer constant register 0	CMCOR0	R/W	H'FFFF	H'FFFF04C6	8, 16, 32
1	Compare match timer control/status register 1	CMCSR1	R/(W)*	H'0000	H'FFFF04C8	8, 16, 32
	Compare match timer counter 1	CMCNT1	R/W	H'0000	H'FFFF04CA	8, 16, 32
	Compare match timer constant register 1	CMCOR1	R/W	H'FFFF	H'FFFF04CC	8, 16, 32

Note: * The CMF bit in CMCSR0 and CMCSR1 can only be written with 0, to clear the flag.

12.2 Register Descriptions

12.2.1 Compare Match Timer Start Register (CMSTR)

The compare match timer start register (CMSTR) is a 16-bit register that specifies operation or stoppage of the counters (CMCNT) in channels 0 and 1.

CMSTR is initialized by a power-on reset, and in hardware standby mode and software standby mode. It is not initialized in module standby mode.

Bit:	15	14	13	12	11	10	9	8
	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	—	—	—	—	—	—	—	—

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	STR1	STR0
Initial value:	0	0	0	0	0	0	0	0
R/W:	—	—	—	—	—	—	R/W	R/W

Bits 15 to 2—Reserved: These bits are always read as 0 and cannot be modified.

Bit 1—Count Start 1 (STR1): Selects operation or stoppage of compare match timer counter 1.

Bit 1: STR1	Description	
0	CMCNT1 count operation is stopped	(Initial value)
1	CMCNT1 performs count operation	

Bit 0—Count Start 0 (STR0): Selects operation or stoppage of compare match timer counter 0.

Bit 0: STR0	Description	
0	CMCNT0 count operation is stopped	(Initial value)
1	CMCNT0 performs count operation	

12.2.2 Compare Match Timer Control/Status Registers 0 and 1 (CMCSR0, CMCSR1)

The compare match timer control/status registers (CMCSR0, CMCSR1) are 16-bit registers that indicate compare match occurrence, enable or disable interrupts, and select the clock to be used for the up-count.

The CMCSR registers are initialized by a power-on reset, and in hardware standby mode and software standby mode. They are not initialized in module standby mode.

Bit:	15	14	13	12	11	10	9	8
	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	—	—	—	—	—	—	—	—

Bit:	7	6	5	4	3	2	1	0
	CMF	CMIE	—	—	—	—	CKS1	CKS0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/(W)*	R/W	—	—	—	—	R/W	R/W

Note: * Only 0 can be written, to clear the flag.

Bits 15 to 8—Reserved: These bits are always read as 0 and cannot be modified.

Bit 7—Compare Match Flag (CMF): Indicates a match between the values of CMCNT and CMCOR.

Bit 7: CMF	Description	
0	CMCNT and CMCOR values do not match [Clearing condition] When 0 is written to CMF after reading CMF = 1	(Initial value)
1	CMCNT and CMCOR values match	

Bit 6—Compare Match Interrupt Enable (CMIE): Enables or disables generation of a compare match interrupt (CMTI) when the CMCNT and CMCOR values match (CMF = 1).

Bit 6: CMIE	Description	
0	Compare match interrupt (CMI) is disabled	(Initial value)
1	Compare match interrupt (CMI) is enabled	

Bits 5 to 2—Reserved: These bits are always read as 0 and cannot be modified.

Bits 1 and 0—Clock Select 1 and 0 (CKS1, CKS0): These bits select the clock to be input to CMCNT from four internal clocks obtained by frequency division of Pϕ. When the STR bit is set to 1 in CMSTR, CMCNT starts counting up on the clock selected by CKS1 and CKS0.

Bit 1: CKS1	Bit 0: CKS0	Description	
0	0	Pϕ/8	(Initial value)
	1	Pϕ/32	
1	0	Pϕ/128	
	1	Pϕ/512	

12.2.3 Compare Match Counters 0 and 1 (CMCNT0, CMCNT1)

The compare match counters (CMCNT0, CMCNT1) are 16-bit registers used as up-counters to generate interrupt requests.

When an internal clock is selected with bits CKS1 and CKS0 in CMCSR, CMCNT starts counting up on that clock. When the CMCNT value matches the value in the compare match constant register (CMCOR), CMCNT is cleared to H'0000 and the CMF flag is set to 1 in CMCSR. If the setting of the CMIE bit in CMCSR is 1 at this time, a compare match interrupt (CMI0 or CMI1) is requested.

The CMCNT registers are initialized by a power-on reset, and in hardware standby mode and software standby mode.

Bit:	15	14	13	12	11	10	9	8
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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

12.2.4 Compare Match Constant Registers 0 and 1 (CMCOR0, CMCOR1)

The compare match constant registers (CMCOR0, CMCOR1) are 16-bit registers that set the compare match cycle.

The CMCOR registers are initialized by a power-on reset, and in hardware standby mode and software standby mode.

Bit:	15	14	13	12	11	10	9	8
Initial value:	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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

12.3 Operation

12.3.1 Cyclic Count Operation

When an internal clock is selected with bits CKS1 and CKS0 in CMCSR, and the STR bit is set to 1 in CMSTR, CMCNT starts counting up on that clock. When the CMCNT value matches the value in the compare match constant register (CMCOR), CMCNT is cleared to H'0000 and the CMF flag is set to 1 in CMCSR. If the setting of the CMIE bit in CMCSR is 1 at this time, a compare match interrupt (CMT1) is requested. CMCNT then starts counting up from H'0000 again.

Figure 12.2 shows the operation of the compare match counter.

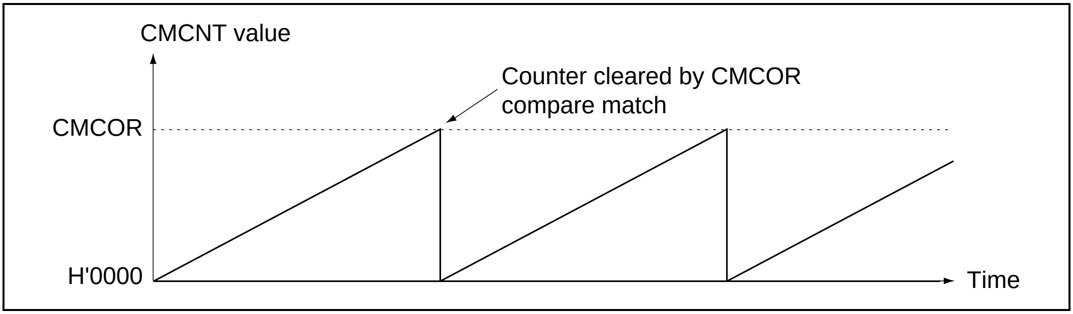


Figure 12.2 Counter Operation

12.3.2 CMCNT Count Timing

One of four clocks (Pϕ/8, Pϕ/32, Pϕ/128, or Pϕ/512) scaled from Pϕ can be selected with bits CKS1 and CKS0 in CMCSR.

Figure 12.3 shows the count timing.

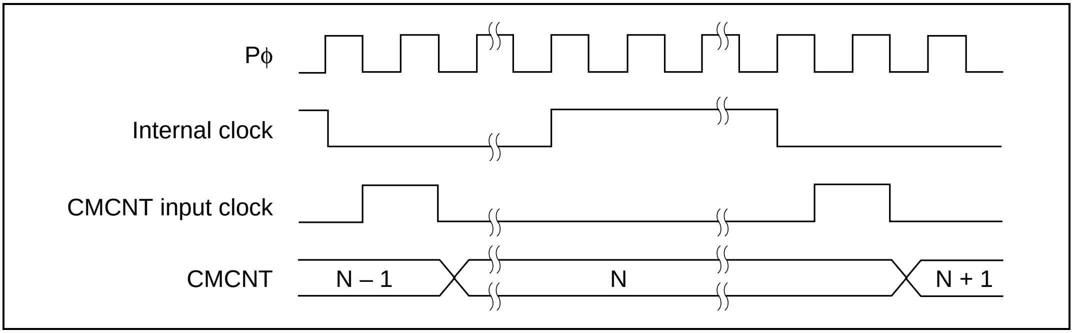


Figure 12.3 Count Timing

12.4 Interrupts

12.4.1 Interrupt Sources

The CMT has a compare match interrupt for each channel, each assigned a different vector address. When interrupt request flag CMF is set to 1, and of interrupt enable bit CMIE is also 1, the corresponding interrupt request is output.

When a CPU interrupt is initiated by an interrupt request, the relative channel priorities can be changed by means of an interrupt controller setting. For details, see section 6, Interrupt Controller (INTC).

12.4.2 Timing of Compare Match Flag Setting

The CMF bit is CMCSR is set to 1 by a compare match signal generated when the CMCOR and CMCNT values match. The compare match signal is generated in the last state in which the match is true (when the value at which the CMCNT match occurred is about to be updated). Therefore, after a match between CMCNT and CMCOR, the compare match signal is not generated until the next CMCNT counter input clock pulse.

Figure 12.4 shows the timing of CMF bit setting.

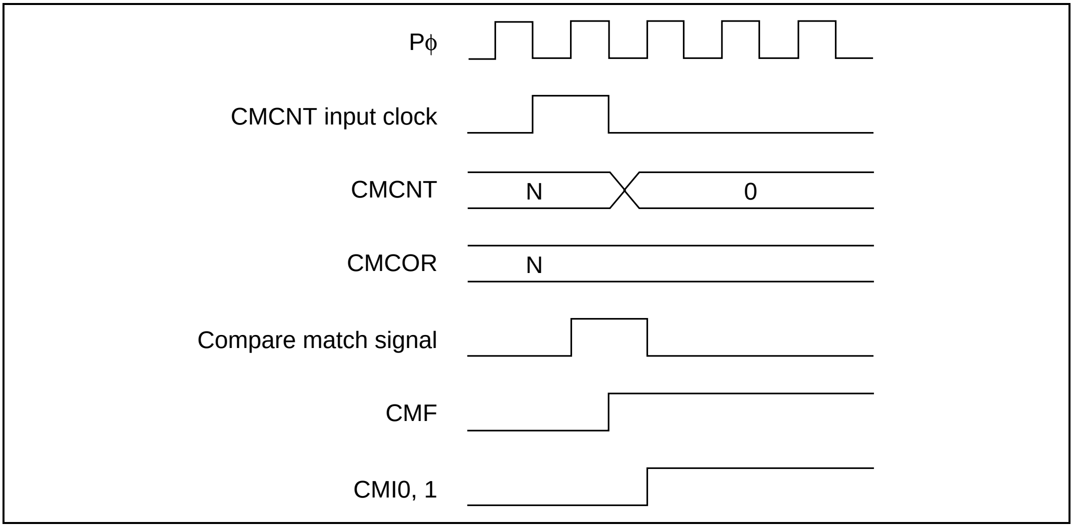


Figure 12.4 Timing of CMF Setting

12.4.3 Timing of Compare Match Flag Clearing

The CMF bit in CMCSR is cleared by reading the bit when it is set to 1, then writing 0 to it.

Figure 12.5 shows the timing of CMF bit clearing.

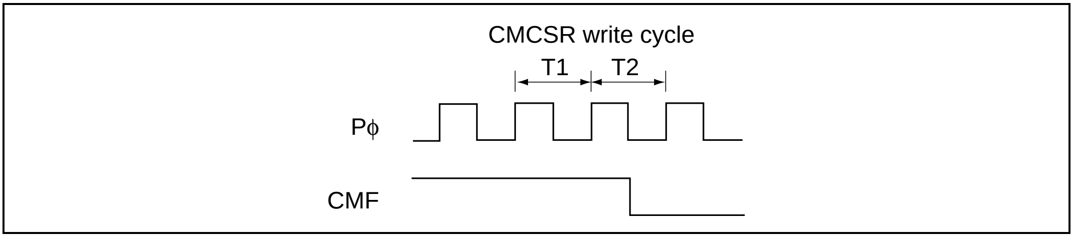


Figure 12.5 Timing of CMF Clearing

12.5 Usage Notes

Note that the kinds of operation and contention described below occur during CMT operation.

Contention between CMCNT Write and Compare Match

If a compare match occurs in the T2 state of a CMCNT write cycle, the CMCNT clearing takes precedence and the write to CMCNT is not performed.

Figure 12.6 shows the timing in this case.

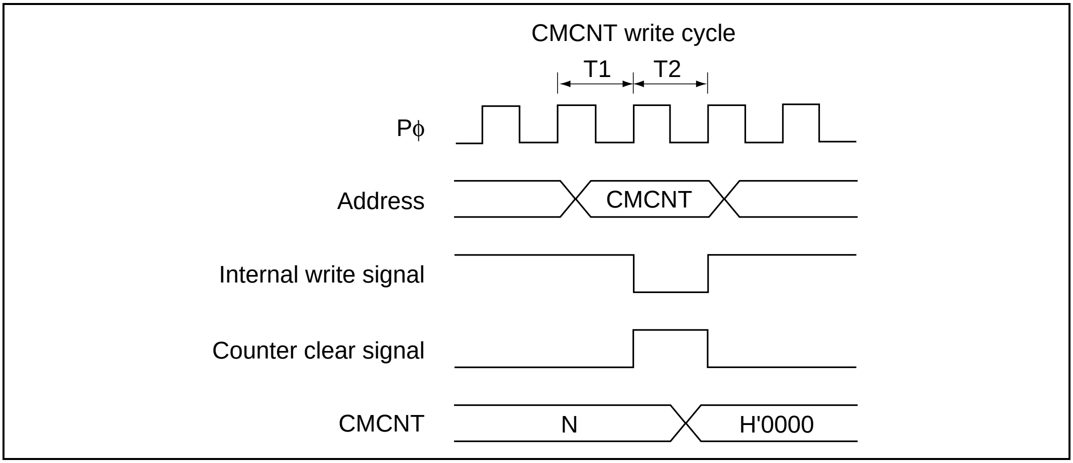


Figure 12.6 Contention between CMCNT Write and Compare Match

Contention between CMCNT Word Write and Increment

If an increment pulse occurs in the T2 state of a CMCNT word write cycle, the counter write takes precedence and counter is not incremented.

Figure 12.7 shows the timing in this case.

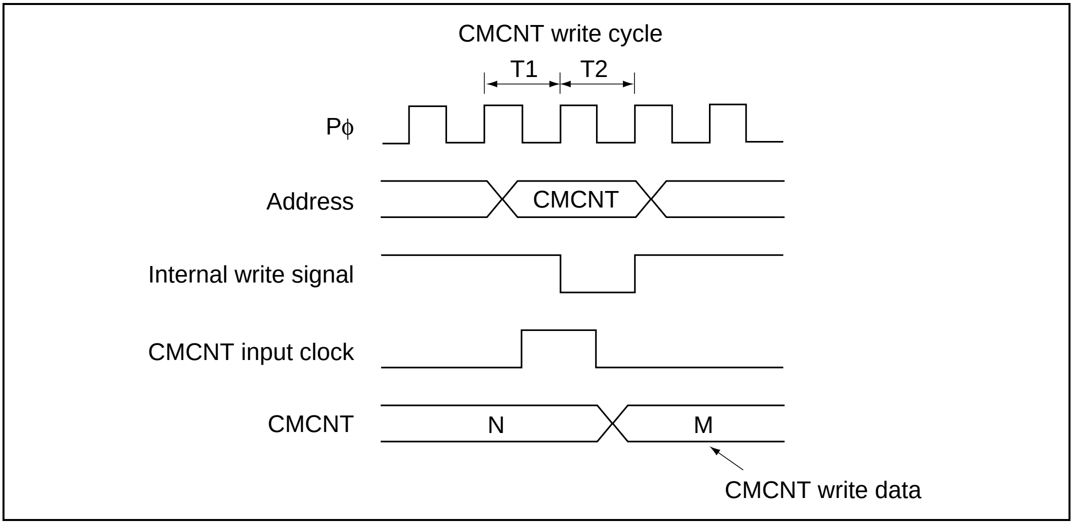


Figure 12.7 Contention between CMCNT Word Write and Increment

Contention between CMCNT Byte Write and Increment

If an increment pulse occurs in the T2 state of a CMCNT byte write cycle, the counter write takes precedence and the byte data for which the write was performed is not incremented. The byte data for which a write was not performed is not incremented either, and retains its previous value.

Figure 12.8 shows the timing when an increment pulse occurs in the T2 state of a CMCNTH write cycle.

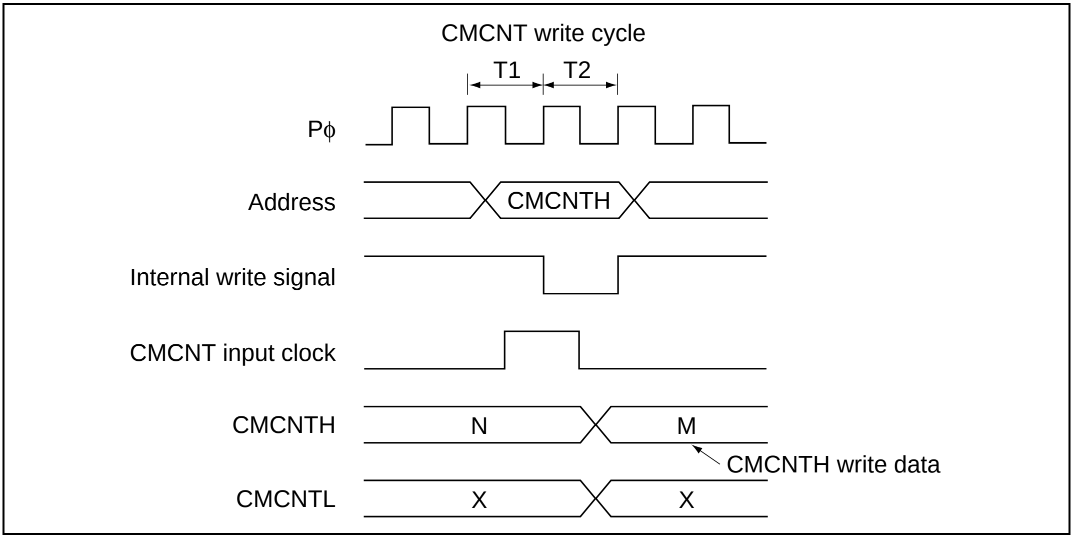


Figure 12.8 Contention between CMCNT Byte Write and Increment

Section 13 Watchdog Timer

13.1 Overview

The SH7065 has a single-channel on-chip watchdog timer (WDT) for monitoring system operation. The WDT outputs an overflow signal ($\overline{\text{WDTOVF}}$) externally if a system crash prevents the CPU from writing to the timer counter, allowing it to overflow. At the same time, the WDT can also generate an internal reset signal for the SH7065.

When this watchdog function is not needed, the WDT can be used as an interval timer. In interval timer operation, an interval timer interrupt is requested each time the counter overflows. The WDT is also in exiting software standby mode.

13.1.1 Features

The WDT has the following features:

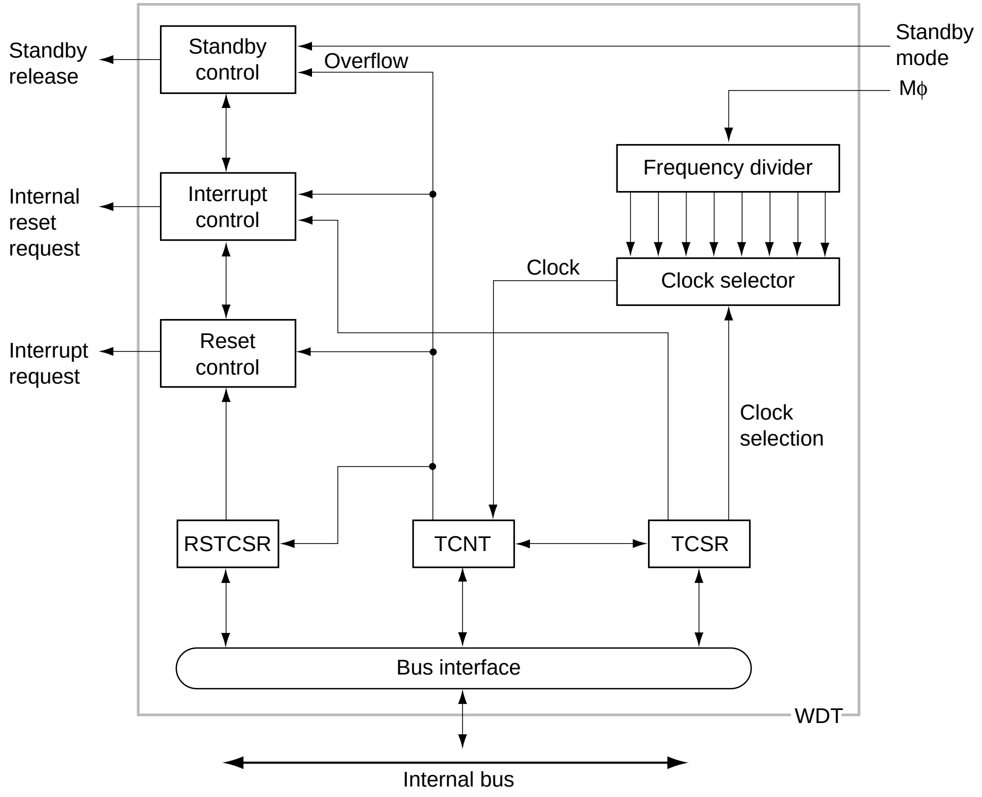
- Switchable between watchdog timer mode and interval timer mode
- $\overline{\text{WDTOVF}}$ output when in watchdog timer mode

  If the counter overflows, the WDT outputs $\overline{\text{WDTOVF}}$ externally. It is possible to select whether or not the SH7065 is reset internally at the same time.
- Interrupt generation when in interval timer mode

  If the counter overflows, the WDT generates an interval timer interrupt.
- Used when exiting software standby mode
- Choice of eight counter input clocks

13.1.2 Block Diagram

Figure 13.1 shows a block diagram of the WDT.



Legend:

TCSR:	Timer control/status register
TCNT:	Timer counter
RSTCSR:	Reset control/status register
Mϕ:	Clock further scaled from the fastest master clock by means of the module clock control register (see section 4, Clock Pulse Generator (CPG) and Power-Down Modes).

Figure 13.1 Block Diagram of WDT

13.1.3 Pin Configuration

Table 13.1 shows details of the WDT output pin.

Table 13.1 WDT Pin

Name	Abbreviation	I/O	Function
Watchdog timer overflow	$\overline{\text{WDTOVF}}$	Output	Outputs counter overflow signal in watchdog timer mode

13.1.4 Register Configuration

The WDT has the three registers shown in table 13.2. These registers control clock selection, WDT mode switching, and the reset signal.

Table 13.2 WDT Registers

Name	Abbre-viation	R/W	Initial Value	Address	
				Write*1	Read*2
Timer control/status register	TCSR	R/(W)*3	H'18	H'FFFF 1000	H'FFFF 1000
Timer counter	TCNT	R/W	H'00	H'FFFF 1000	H'FFFF 1001
Reset control/status register	RSTCSR	R/(W)*3	H'1F	H'FFFF 1002	H'FFFF 1003

Notes: 1. Use word writes; byte and longword writes cannot be used.

2. Use byte reads; a word or longword read will not return the correct value.

3. Only 0 can be written to bit 7, to clear the flag.

13.2 Register Descriptions

13.2.1 Timer Counter (TCNT)

Bit:	7	6	5	4	3	2	1	0
	TCNT7	TCNT6	TCNT5	TCNT4	TCNT3	TCNT2	TCNT1	TCNT0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

The timer counter (TCNT) is an 8-bit readable/writable up-counter. When the timer enable bit (TME) bit is set to 1 in the timer control/status register (TCSR), TCNT starts counting up on the internal clock selected with bits CKS2 to CKS0 in TCSR. When the count overflows (changes from H'FF to H'00), either the watchdog timer overflow signal ($\overline{\text{WDTOVF}}$) or an interval timer interrupt (ITI) is generated, depending on the mode selected with the WT/$\overline{\text{IT}}$ bit in TCSR.

TCNT is initialized to H'00 by a power-on reset, or when the TME bit is cleared to 0. It is not initialized in standby mode.

Note: The method of writing to TCNT is different from that for general registers to prevent inadvertent overwriting. For details see section 13.2.4, Notes on Register Access.

13.2.2 Timer Control/Status Register (TCSR)

Bit:	7	6	5	4	3	2	1	0
	OVF	WT/$\overline{\text{IT}}$	TME	—	—	CKS2	CKS1	CKS0
Initial value:	0	0	0	1	1	0	0	0
R/W:	R/(W)*	R/W	R/W	R	R	R/W	R/W	R/W

Note: * Only 0 can be written to bit 7, to clear the flag.

The timer control/status register (TCSR) is an 8-bit readable/writable register whose functions include selecting the clock source to be input to the timer counter (TCNT), and the timer mode.

Bits 7 to 5 are initialized to 000 by a power-on reset and in standby mode. Bits 2 to 0 are initialized to 000 by a power-on reset, but are not initialized in standby mode.

Note: The method of writing to TCSR is different from that for general registers to prevent inadvertent overwriting. For details see section 13.2.4, Notes on Register Access.

Bit 7—Overflow Flag (OVF): Indicates that TCNT has overflowed from H'FF to H'00 when in interval timer mode. This flag is not set in watchdog timer (WDT) mode.

Bit 7: OVF	Description
0	No TCNT overflow in interval timer mode (Initial value)
1	TCNT overflow has occurred in interval timer mode [Clearing condition] Cleared by reading OVF, then writing 0 to OVF

Bit 6—Timer Mode Select (WT/$\overline{\text{IT}}$): Selects whether the WDT is used as a watchdog timer or interval timer. If used as an interval timer, the WDT generates an interval timer interrupt request (ITI) when TCNT overflows. If used as a watchdog timer, the WDT generates the $\overline{\text{WDTOVF}}$ signal when TCNT overflows.

Bit 6: WT/$\overline{\text{IT}}$	Description
0	Interval timer mode: Interval timer interrupt (ITI) request is sent to CPU when TCNT overflows (Initial value)
1	Watchdog timer mode: $\overline{\text{WDTOVF}}$ signal is output externally when TCNT overflows

Note: For details of what happens when TCNT overflows during watchdog timer operation, see section 13.2.3, Reset Control/Status Register (RSTCSR).

Bit 5—Timer Enable (TME): Selects whether the timer runs or is halted.

Bit 5: TME	Description
0	Timer disable: TCNT is initialized to H'00 and halted (Initial value)
1	Timer enable: TCNT counts up

Bits 4 and 3—Reserved: These bits are always read as 1 and should only be written with 1.

Bits 2 to 0—Clock Select 2 to 0 (CKS2 to CKS0): These bits select one of eight clocks, obtained by dividing the Mϕ clock, for input to TCNT.

			Description	
Bit 2: CKS2	**Bit 1: CKS1**	**Bit 0: CKS0**	**Clock Division Ratio**	**Overflow Period (When Mϕ = 60 MHz)**
0	0	0	1/2 (Initial value)	8.5 µs
		1	1/4	17.1 µs
	1	0	1/8	34.1 µs
		1	1/32	136.5 µs
1	0	0	1/256	1.1 ms
		1	1/1024	4.4 ms
	1	0	1/2048	8.7 ms
		1	1/4096	17.5 ms

Note: The overflow period is the time from when TCNT starts counting up from H'00 until overflow occurs. Mϕ is a clock further scaled from the master clock by means of the module clock control register. For details see section 4, Clock Pulse Generator (CPG) and Power-Down Modes.

13.2.3 Reset Control/Status Register (RSTCSR)

Bit:	7	6	5	4	3	2	1	0
	WOVF	RSTE	—	—	—	—	—	—
Initial value:	0	0	0	1	1	1	1	1
R/W:	R/(W)*	R/W	R	R	R	R	R	R

Note: * Only 0 can be written to bit 7, to clear the flag.

The reset control/status register (RSTCSR) is an 8-bit readable/writable register that controls generation of the internal reset signal when the timer counter (TCNT) overflows.

RSTCSR is initialized to H'1F by a reset signal from the $\overline{\text{RES}}$ pin, but is not initialized by the internal reset signal caused by WDT overflow. It is also initialized to H'1F in standby mode.

Note: The method of writing to RSTCSR is different from that for general registers to prevent inadvertent overwriting. For details see section 13.2.4, Notes on Register Access.

Bit 7—Watchdog Overflow Flag (WOVF): Indicates that TCNT has overflowed (changed from H'FF to H'00) in watchdog timer mode. This bit is not set in interval timer mode.

Bit 7: WOVF	Description	
0	No TCNT overflow in watchdog timer mode	(Initial value)
1	TCNT overflow has occurred in watchdog timer mode [Clearing condition] Cleared by reading WOVF, then writing 0 to WOVF	

Bit 6—Reset Enable (RSTE): Specifies whether or not a reset signal is to be generated in the SH7065 if TCNT overflows in watchdog timer mode.

Bit 6: RSTE	Description	
0	Internal reset is not performed if TCNT overflows	(Initial value)
1	Internal reset is performed if TCNT overflows	

Note: The modules in the SH7065 are not reset, but TCNT and TCSR within the WDT are reset.

Bit 5—Reserved: This bit is always read as 0 and should only be written with 0.

Bits 4 to 0—Reserved: These bits are always read as 1 and should only be written with 1.

13.2.4 Notes on Register Access

The method of writing to the watchdog timer's timer counter (TCNT), timer control/status register (TCSR), and reset control/status register (RSTCSR) differs from that for other registers to prevent inadvertent overwriting. The procedures for writing to and reading these registers are given below.

Writing to TCNT and TCSR

A word transfer instruction must be used to write to TCNT and TCSR. They cannot be written to with a byte transfer instruction.

Figure 13.2 shows the format of data written to TCNT and TCSR. TCNT and TCSR both have the same write address. For a write to TCNT, the upper byte of the written word must contain H'5A and the lower byte must contain the write data. For a write to TCSR, the upper byte of the written word must contain H'A5 and the lower byte must contain the write data. This transfers the write data from the lower byte to TCNT or TCSR.

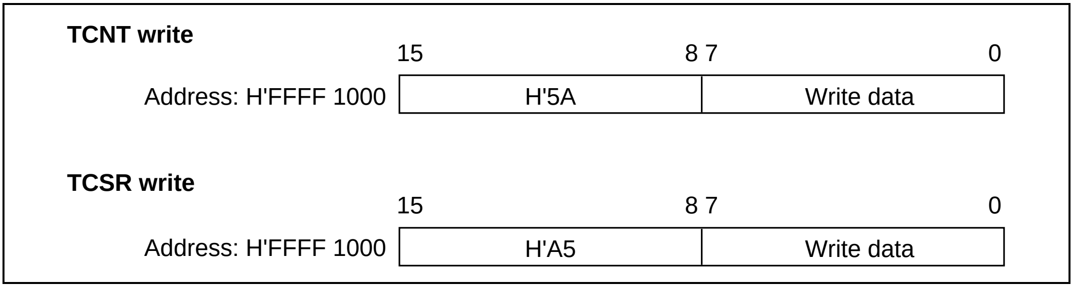


Figure 13.2 Writing to TCNT and TCSR

Writing to RSTCSR

To write to RSTCSR, a word transfer must be made to address H'FFFF1002. RSTCSR cannot be written to with a byte transfer instruction.

Figure 13.3 shows the format of data written to RSTCSR. The method of writing 0 to the WOVF bit (bit 7) differs from that for writing to the RSTE bit (bit 6).

To write 0 to the WOVF bit, the write data must have H'A5 in the upper byte and H'00 in the lower byte. This clears the WOVF bit to 0, but has no effect on the RSTE bit. To write to the RSTE bit, the upper byte must contain H'5A and the lower byte must contain the write data. This writes the value in bit 6 of the lower byte into the RSTE bit, but has no effect on the WOVF bit.

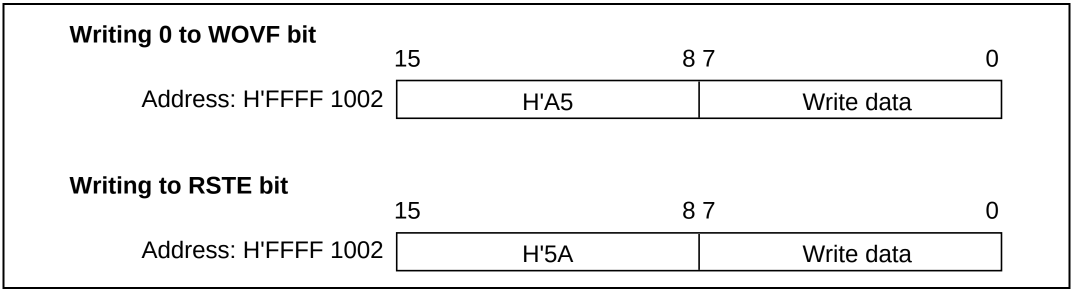


Figure 13.3 Writing to RSTCSR

Reading TCNT, TCSR, and RSTCSR

These registers are read in the same way as other registers. The read addresses are H'FFFF1000 for TCSR, H'FFFF1001 for TCNT, and H'FFFF1003 for RSTCSR. Byte transfer instructions must be used to read these registers.

13.3 Operation

13.3.1 Operation in Watchdog Timer Mode

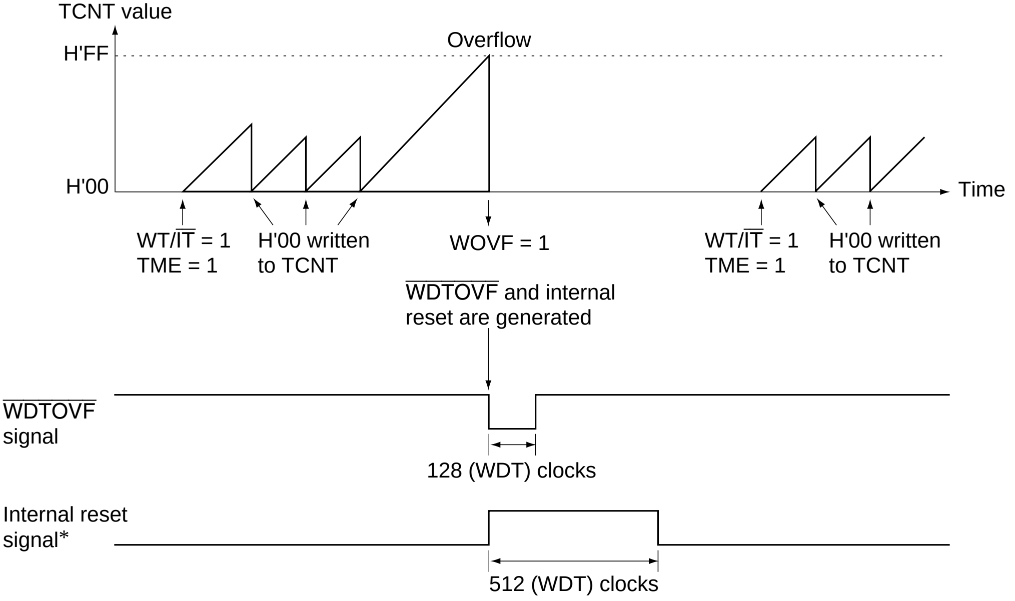
Figure 13.4 illustrates WDT operation in watchdog timer mode. To use the WDT as a watchdog timer, set the WT/$\overline{\text{IT}}$ and TME bits to 1 in the timer control/status register (TCSR). Software must prevent TCNT overflows by rewriting the TCNT value (normally be writing H'00) before overflows occurs. This ensures that TCNT does not overflow while the system is operating normally. If TCNT overflows without being rewritten because of a system crash or other error, the $\overline{\text{WDTOVF}}$ signal is output. This $\overline{\text{WDTOVF}}$ signal can be used to reset the system.

The $\overline{\text{WDTOVF}}$ signal is output for 128 (WDT) clock cycles. This (WDT) clock is a clock further scaled from the internal clock by means of the module clock control register (see section 4, Clock Pulse Generator (CPG) and Power-Down Modes).

If TCNT overflows when 1 is set in the RSTE bit in the reset control/status register (RSTCSR), a signal that resets the SH7065 internally is generated at the same time as the $\overline{\text{WDTOVF}}$ signal. The internal reset signal is output for 512 (WDT) clock cycles.

If a reset caused by a signal input to the $\overline{\text{RES}}$ pin occurs at the same time as a reset caused by a WDT overflow, the $\overline{\text{RES}}$ pin reset has priority and the WOVF bit in RSTCSR is cleared to 0.

The following registers are not initialized by the WDT reset signal: (1) the MMT's POE (port output enable) function registers, (2) pin function controller (PFC) registers, (3) I/O port registers. These registers are initialized only by a power-on reset from off-chip.



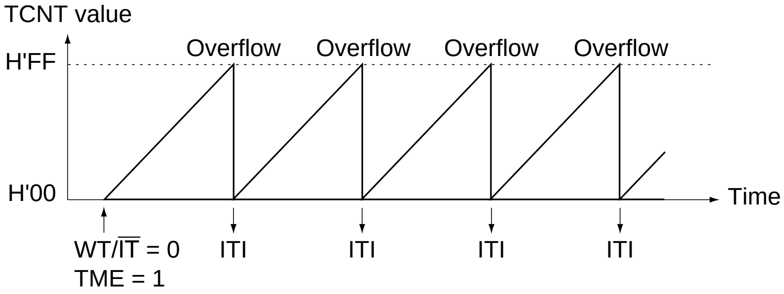
Legend:
WT/$\overline{\text{IT}}$: Timer mode select bit
TME: Timer enable bit

Note: * The internal reset signal is generated only if the RSTE bit is set to 1.

Figure 13.4 Operation in Watchdog Timer Mode

13.3.2 Operation in Interval Timer Mode

Figure 13.5 illustrates WDT operation in interval timer mode. To use the WDT as an interval timer, clear the $WT/\overline{IT}$ bit in TCSR to 0 and set the TME bit to 1. When the WDT is operating as an interval timer, an interval timer interrupt (ITI) is generated each time TCNT overflows. This function can be used to generate interrupt requests at regular intervals.



Legend:
ITI: Interval timer interrupt request generation

Figure 13.5 Operation in Interval Timer Mode

13.3.3 Operation When Clearing Software Standby Mode

The WDT is used when software standby mode is cleared by an NMI interrupt. When software standby mode is used, the WDT should be set as described in 1 below.

1. Settings before transition to software standby mode

   Before making a transition to software standby mode, the WDT must be halted by clearing the TME bit to 0 in the timer control/status register (TCSR). A transition to software standby mode cannot be made while the TME bit is set to 1. Also set bits CKS2 to CKS0 in TCSR so that the timer counter (TCNT) overflow period is at least as long as the oscillation settling time (see section 22.3, AC Characteristics Test Conditions).

2. Operation when software standby mode is cleared

   When an NMI interrupt is generated in software standby mode, the oscillator starts operating and TCNT begins counting up on the clock selected with bits CKS2 to CKS0 prior to the transition to software standby mode.

   When TCNT overflows (from H'FF to H'00), the clock is judged to be stable and ready for use, and clocks are supplied throughout the chip. This clears software standby mode.

Section 14 Serial Communication Interface (SCI)

14.1 Overview

The SH7065 is equipped with a three-channel serial communication interface with built-in FIFO buffers (SCI: SCI with FIFO). The SCI can handle both asynchronous and synchronous serial communication. A function is also provided for serial communication between processors (multiprocessor communication function).

An on-chip Infrared Data Association (IrDA) interface based on the IrDA 1.0 system is also provided, enabling infrared communication.

Sixteen-stage FIFO registers are provided for both transmission and reception, enabling fast, efficient, and continuous communication.

14.1.1 Features

The SCI has the following features:

- Choice of synchronous or asynchronous serial communication mode
    - Asynchronous 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). A multiprocessor communication function is also provided that enables serial data communication with a number of processors.

      There is a choice of 12 serial data transfer formats.

        - Data length: 7 or 8 bits
        - Stop bit length: 1 or 2 bits
        - Parity: Even/odd/none
        - Multiprocessor bit: 1 or 0
        - Receive error detection: Parity, overrun, and framing errors
        - Auto break detection: A break can be detected automatically.
    - Synchronous mode

      Serial data communication is synchronized with a clock. Serial data communication can be carried out with other chips that have a synchronous communication function.

      There is a single serial data transfer format.

  - Data length: 8 bits
  - Receive error detection: Overrun errors
- IrDA 1.0 compliance
- Full-duplex communication capability

  The transmitter and receiver are mutually independent, enabling transmission and reception to be executed simultaneously. Double-buffering is used in both the transmitter and the receiver, enabling continuous transmission and continuous reception of serial data.

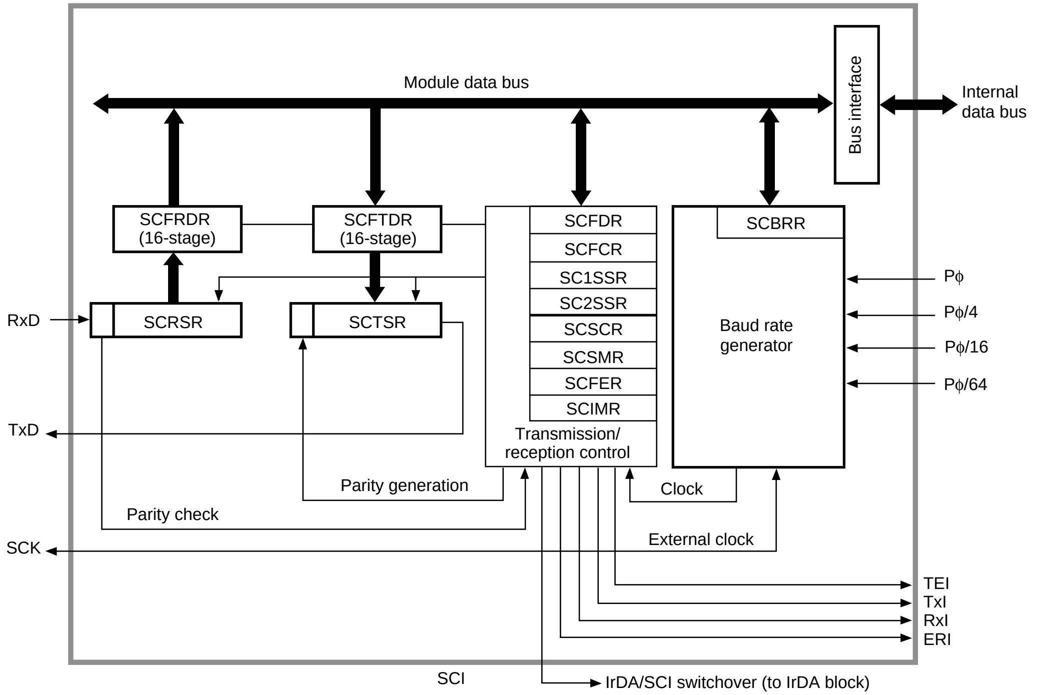
  In addition, the transmitter and receiver both have a 16-stage FIFO buffer structure, enabling continuous serial data transmission and reception.

  (However, IrDA communication is carried out in half-duplex mode.)
- Built-in 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 SCK pin
- Four interrupt sources

  There are four interrupt sources—transmit-FIFO-data-empty, transmit-end, receive-FIFO-data-full, and receive-error—that can issue requests independently. The transmit-FIFO-data-empty and receive-FIFO-data-full interrupts can activate the on-chip DMAC to execute data transfer
- When not in use, the SCI can be stopped by halting its clock supply to reduce power consumption.
- Choice of LSB-first or MSB-first mode
- In asynchronous mode, operation can be selected on a base clock of 4, 8, or 16 times the bit rate.

14.1.2 Block Diagrams

A block diagram of the SCI is shown in figure 14.1, and a diagram of the IrDA block in figure 14.2.



Legend:

SCRSR:	Receive shift register	SC1SSR:	Serial status 1 register
SCFRDR:	Receive FIFO data register	SC2SSR:	Serial status 2 register
SCTSR:	Transmit shift register	SCBRR:	Bit rate register
SCFTDR:	Transmit FIFO data register	SCFCR:	FIFO control register
SCSMR:	Serial mode register	SCFDR:	FIFO data count register
SCSCR:	Serial control register	SCFER:	FIFO error register
		SCIMR:	IrDA mode register

Figure 14.1 Block Diagram of SCI

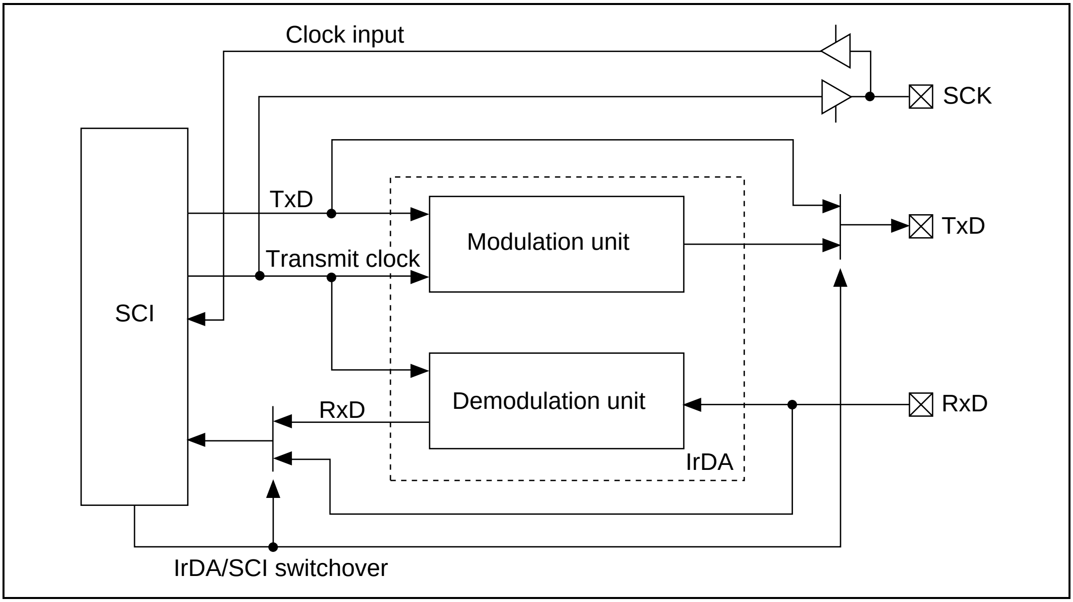


Figure 14.2 Diagram of IrDA Block

14.1.3 Pin Configuration

The SCI has the serial pins shown in table 14.1 for each channel.

Table 14.1 SCI Pins

Channel	Name	Abbreviation	I/O	Function
0–2	Serial clock pin	SCK0–2	I/O	Clock input/output
	Receive data pin	RxD0–2	Input	Receive data input
	Transmit data pin	TxD0–2	Output	Transmit data output

14.1.4 Register Configuration

The SCI has the internal registers shown in table 14.2. These registers are used to specify asynchronous mode/synchronous mode and the IrDA communication mode, the data format and the bit rate, and to perform transmitter/receiver control.

Table 14.2 SCI Registers

Channel	Name	Abbre- viation	R/W	Initial Value	Address	Access Size
0	Serial mode register	SCSMR0	R/W	H'00	H'FFFF0500	8
	Bit rate register	SCBRR0	R/W	H'FF	H'FFFF0502	8
	Serial control register	SCSCR0	R/W	H'00	H'FFFF0504	8
	Transmit FIFO data register	SCFTDR0	W	—	H'FFFF0506	8
	Serial status 1 register	SC1SSR0	R/(W)*	H'84	H'FFFF0508	16
	Serial status 2 register	SC2SSR0	R/(W)*	H'20	H'FFFF050A	8
	Receive FIFO data register	SCFRDR0	R	Undefined	H'FFFF050C	8
	FIFO control register	SCFCR0	R/W	H'00	H'FFFF050E	8
	FIFO data count register	SCFDR0	R	H'00	H'FFFF0510	16
	FIFO error register	SCFER0	R	H'00	H'FFFF0512	16
	IrDA mode register	SCIMR0	R/W	H'00	H'FFFF0514	8
1	Serial mode register	SCSMR1	R/W	H'00	H'FFFF0520	8
	Bit rate register	SCBRR1	R/W	H'FF	H'FFFF0522	8
	Serial control register	SCSCR1	R/W	H'00	H'FFFF0524	8
	Transmit FIFO data register	SCFTDR1	W	—	H'FFFF0526	8
	Serial status 1 register	SC1SSR1	R/(W)*	H'84	H'FFFF0528	16
	Serial status 2 register	SC2SSR1	R/(W)*	H'20	H'FFFF052A	8
	Receive FIFO data register	SCFRDR1	R	Undefined	H'FFFF052C	8
	FIFO control register	SCFCR1	R/W	H'00	H'FFFF052E	8
	FIFO data count register	SCFDR1	R	H'00	H'FFFF0530	16
	FIFO error register	SCFER1	R	H'00	H'FFFF0532	16
	IrDA mode register	SCIMR1	R/W	H'00	H'FFFF0534	8

Channel	Name	Abbre- viation	R/W	Initial Value	Address	Access Size
2	Serial mode register	SCSMR2	R/W	H'00	H'FFFF0540	8
	Bit rate register	SCBRR2	R/W	H'FF	H'FFFF0542	8
	Serial control register	SCSCR2	R/W	H'00	H'FFFF0544	8
	Transmit FIFO data register	SCFTDR2	W	—	H'FFFF0546	8
	Serial status 1 register	SC1SSR2	R/(W)*	H'84	H'FFFF0548	16
	Serial status 2 register	SC2SSR2	R/(W)*	H'20	H'FFFF054A	8
	Receive FIFO data register	SCFRDR2	R	Undefined	H'FFFF054C	8
	FIFO control register	SCFCR2	R/W	H'00	H'FFFF054E	8
	FIFO data count register	SCFDR2	R	H'00	H'FFFF0550	16
	FIFO error register	SCFER2	R	H'00	H'FFFF0552	16
	IrDA mode register	SCIMR2	R/W	H'00	H'FFFF0554	8

Note: * Only 0 can be written, to clear flags. Use byte access on registers with an access size of 8, and word access on registers with an access size of 16.

14.2 Register Descriptions

With the exception of the IrDA mode register (SCIMR) and bits 6 to 3 (ICK3 to ICK0) of the serial mode register (SCSMR), IrDA communication mode settings are the same as for asynchronous mode.

14.2.1 Receive Shift Register (SCRSR)

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

The receive shift register (SCRSR) is the register used to receive serial data.

The SCI sets serial data input from the RxD pin in SCRSR in the order received, starting with the LSB (bit 0) or MSB (bit 7), and converts it to parallel data. When one byte of data has been received, it is transferred to the receive FIFO data register, SCFRDR, automatically.

SCRSR cannot be read or written to directly.

14.2.2 Receive FIFO Data Register (SCFRDR)

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

The receive FIFO data register (SCFRDR) is a 16-stage FIFO register (8 bits per stage) that stores received serial data.

When the SCI 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 the receive FIFO data register is full (16 data bytes).

SCFRDR is a read-only register, and cannot be written to.

If a read is performed when there is no receive data in the receive FIFO data register, an undefined value will be returned. When the receive FIFO data register is full of receive data, subsequent receive data is lost.

14.2.3 Transmit Shift Register (SCTSR)

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

The transmit shift register (SCTSR) is the register used to transmit serial data.

To perform serial data transmission, the SCI first transfers transmit data from SCFTDR to SCTSR, then sends the data to the TxD pin starting with the LSB (bit 0) or MSB (bit 7).

When transmission of one byte is completed, the next transmit data is transferred from SCFTDR to SCTSR, and transmission started, automatically.

SCTSR cannot be read or written to directly.

14.2.4 Transmit FIFO Data Register (SCFTDR)

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

The transmit FIFO data register (SCFTDR) is a 16-stage FIFO register (8 bits per stage) that stores data for serial transmission.

If SCTSR is empty when transmit data has been written to SCFTDR, the SCI transfers the transmit data written in SCFTDR to SCTSR and starts serial transmission.

SCFTDR is a write-only register, and cannot be read.

The next data cannot be written when SCFTDR is filled with 16 bytes of transmit data. Data written in this case is ignored.

14.2.5 Serial Mode Register (SCSMR)

Bit:	7	6	5	4	3	2	1	0
	C/$\overline{\text{A}}$	CHR/ ICK3	PE/ICK2	O/$\overline{\text{E}}$/ICK1	STOP/ ICK0	MP	CKS1	CKS0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

The serial mode register (SCSMR) is an 8-bit register used to set the SCI's serial transfer format and select the baud rate generator clock source. In IrDA communication mode, it is used to select the output pulse width.

SCSMR can be read or written to by the CPU at all times.

SCSMR is initialized to H'00 by a reset, by the module standby function, and in standby mode.

Bit 7—Communication Mode (C/$\overline{\text{A}}$): Selects asynchronous mode or synchronous mode as the SCI operating mode. In IrDA communication mode, this bit must be cleared to 0.

Bit 7: C/$\overline{\text{A}}$	Description	
0	Asynchronous mode	(Initial value)
1	Synchronous mode	

Bit 6—Character Length (CHR)/IrDA Clock Select 3 (ICK3): Selects 7 or 8 bits as the data length in asynchronous mode. In synchronous mode, a fixed data length of 8 bits is used regardless of the CHR setting,

Bit 6: CHR	Description	
0	8-bit data	(Initial value)
1	7-bit data*	

Note: * When 7-bit data is selected, some bits of the transmit FIFO data register (SCFTDR) are not transmitted according to the selection of either LSB-first or MSB-first mode in data transmission by the TLM (bit 7) of the serial status 2 register (SC2SSR):

(1) When TLM = 0 (transmission in LSB-first mode): MSB (bit 7) is not transmitted.

(2) When TLM = 1 (transmission in MSB-first mode): LSB (bit 0) is not transmitted.

In IrDA communication mode, bit 6 is the IrDA clock select 3 (ICK3) bit, enabling appropriate clock pulses to be generated according to its setting. See, Pulse Width Selection, in section 14.3.6, Operation in IrDA Mode, for details.

Bit 5—Parity Enable (PE)/IrDA Clock Select 2 (ICK2): In asynchronous mode, selects whether or not parity bit addition is performed in transmission, and parity bit checking in reception. In synchronous mode, parity bit addition and checking is not performed, regardless of the PE bit setting.

Bit 5: PE	Description	
0	Parity bit addition and checking disabled	(Initial value)
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.

In IrDA communication mode, bit 5 is the IrDA clock select 2 (ICK2) bit, enabling appropriate clock pulses to be generated according to its setting. See, Pulse Width Selection, in section 14.3.6, Operation in IrDA Mode, for details.

Bit 4—Parity Mode (O/$\overline{\text{E}}$)/IrDA Clock Select 1 (ICK1): Selects either even or odd parity for use in parity addition and checking. The O/$\overline{\text{E}}$ bit setting is only valid when the PE bit is set to 1, enabling parity bit addition and checking, in asynchronous mode. The O/$\overline{\text{E}}$ bit setting is invalid in synchronous mode, and when parity addition and checking is disabled in asynchronous mode.

Bit 4: O/$\overline{\text{E}}$	Description	
0	Even parity*1	(Initial value)
1	Odd parity*2	

Notes: 1. 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.

2. 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.

In IrDA communication mode, bit 4 is the IrDA clock select 1 (ICK1) bit, enabling appropriate clock pulses to be generated according to its setting. See, Pulse Width Selection, in section 14.3.6, Operation in IrDA Mode, for details.

Bit 3—Stop Bit Length (STOP)/IrDA Clock Select 0 (ICK0): Selects 1 or 2 bits as the stop bit length in asynchronous mode. The STOP bit setting is only valid in asynchronous mode. If synchronous mode is set, the STOP bit setting is invalid since stop bits are not added.

Bit 3: STOP	Description	
0	1 stop bit*1	(Initial value)
1	2 stop bits*2	

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.

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.

In IrDA communication mode, bit 3 is the IrDA clock select 0 (ICK0) bit, enabling appropriate clock pulses to be generated according to its setting. See, Pulse Width Selection, in section 14.3.6, Operation in IrDA Mode, for details.

Bit 2—Multiprocessor Mode (MP): Selects a multiprocessor format. When a multiprocessor format is selected, the PE bit and $O/\overline{E}$ bit parity settings are invalid. The MP bit setting is only valid in asynchronous mode; it is invalid in synchronous mode and IrDA mode.

For details of the multiprocessor communication function, see section 14.3.3, Multiprocessor Communication Function.

Bit 2: MP	Description	
0	Multiprocessor function disabled	(Initial value)
1	Multiprocessor format selected	

Bits 1 and 0—Clock Select 1 and 0 (CKS1, CKS0): These bits select the clock source for the built-in baud rate generator. The clock source can be selected from Pφ, Pφ/4, Pφ/16, and Pφ/64, according to the setting of bits CKS1 and CKS0.

For the relationship between the clock source, the bit rate register setting, and the baud rate, see section 14.2.9, Bit Rate Register (SCBRR).

Bit 1: CKS1	Bit 0: CKS0	Description	
0	0	Pφ clock	(Initial value)
	1	Pφ/4 clock	
1	0	Pφ/16 clock	
	1	Pφ/64 clock	

Note: Pφ (SCI) is a clock scaled from the CKP peripheral clock according to the setting in the module clock control register. For details see section 4, Clock Pulse Generator (CPG) and Power-Down Modes.

14.2.6 Serial Control Register (SCSCR)

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

The serial control register (SCSCR) performs enabling or disabling of SCI transmit/receive operations, and interrupt requests, and selection of the transmit/receive clock source.

SCSCR can be read or written to by the CPU at all times.

SCSCR is initialized to H'00 by a reset, by the module standby function, and in standby mode.

Bit 7—Transmit Interrupt Enable (TIE): Enables or disables transmit-FIFO-data-empty interrupt (TXI) request generation when, after serial transmit data is transferred from the transmit FIFO data register (SCFTDR) to the transmit shift register (SCTSR), the number of data bytes in SCFTDR falls to or below the transmit trigger set number, and the TDFE flag is set to 1 in the serial status 1 register (SC1SSR).

Bit 7: TIE	Description
0	Transmit-FIFO-data-empty interrupt (TXI) request disabled* (Initial value)
1	Transmit-FIFO-data-empty interrupt (TXI) request enabled

Note: * TXI interrupt requests can be cleared by writing transmit data exceeding the transmit trigger set number to SCFTDR, reading 1 from the TDFE flag, then clearing it to 0, or by clearing the TIE bit to 0. When transmit data is written to SCFTDR using the on-chip DMAC, the TDFE flag is cleared automatically.

Bit 6—Receive Interrupt Enable (RIE): Enables or disables generation of a receive-FIFO-data-full interrupt (RXI) request and receive-error interrupt (ERI) request when, after serial receive data is transferred from the receive shift register (SCRSR) to the receive FIFO data register (SCFRDR), the number of data bytes in SCFRDR reaches or exceeds the receive trigger set number, and the RDF flag is set to 1 in SC1SSR.

Bit 6: RIE	Description
0	Receive-FIFO-data-full interrupt (RXI) request and receive-error interrupt (ERI) request disabled* (Initial value)
1	Receive-FIFO-data-full interrupt (RXI) request and receive-error interrupt (ERI) request enabled

Note: * RXI and ERI interrupt requests can be cleared by reading 1 from the RDF flag, or the ORER, BRK, DR, or ER flag, then clearing the flag to 0, or by clearing the RIE bit to 0. When ORER occurs, read at least the receive trigger set number of receive data bytes from SCFRDR, then read 1 from the ORER flag and clear it to 0.

Bit 5—Transmit Enable (TE): Enables or disables the start of serial transmission by the SCI.

Bit 5: TE	Description	
0	Transmission disabled*1	(Initial value)
1	Transmission enabled*2	

Notes: 1. The TDFE flag in SC1SSR is not affected by clearing TE to 0, and the TxD pin is fixed high.

2. Serial transmission is started when transmit data is written to SCFTDR in this state.

Serial mode register (SCSMR) and FIFO control register (SCFCR) settings must be made, the transmission format decided, and the transmit FIFO reset, before the TE bit is set to 1.

Bit 4—Receive Enable (RE): Enables or disables the start of serial reception by the SCI.

Bit 4: RE	Description	
0	Reception disabled*1	(Initial value)
1	Reception enabled*2	

Notes: 1. Clearing the RE bit to 0 does not affect the RDF, FER, PER, ORER, DR, and BRK flags, which retain their states.

2. Serial reception is started in this state when a start bit is detected in asynchronous mode or serial clock input is detected in synchronous mode. When a setting is made to output a serial clock in synchronous mode, if TE = 0, the serial clock is output and reception is started as soon as RE is set to 1.

When TE = 1 and RE = 1, serial data is received simultaneously with the transmit operation.

SCSMR setting must be made to decide the reception format before setting the RE bit to 1.

Bit 3—Multiprocessor Interrupt Enable (MPIE): Enables or disables multiprocessor interrupts. The MPIE bit setting is only valid in asynchronous mode when the MP bit in SCSMR is set to 1.

The MPIE bit setting is invalid in synchronous mode and IrDA mode, and when the MP bit is cleared to 0.

Bit 3: MPIE	Description
0	Multiprocessor interrupts disabled (normal reception performed) (Initial value) [Clearing conditions] • When the MPIE bit is cleared to 0 • When data with MPB = 1 is received
1	Multiprocessor interrupts enabled* Receive interrupt (RXI) requests, receive-error interrupt (ERI) requests, and setting of the RDF, ORER, and FER flags in SC1SSR are disabled until data with the multiprocessor bit set to 1 is received.

Note: * Receive data transfer from SCRSR to SCFRDR, receive error detection, and setting of the RDF, FER, and ORER flags in SC1SSR, is not performed. When receive data including MPB = 1 is received, the MPB FLAG in SC1SSR is set to 1, the MPIE bit is cleared to 0 automatically, and generation of RXI and ERI interrupts (when the RIE bit in SCSCR are set to 1) and FER and ORER flag setting is enabled.

Bit 2—Transmit-End interrupt Enable (TEIE): Enables or disables transmit-end interrupt (TEI) request generation when there is no valid transmit data in SCFTDR when the last bit of the transmit data is sent.

Bit 2: TEIE	Description
0	Transmit-end interrupt (TEI) request disabled* (Initial value)
1	Transmit-end interrupt (TEI) request enabled*

Note: * TEI interrupt requests can be cleared by writing data to SCFTDR and clearing the TEND flag to 0 in SC1SSR, or by clearing the TEIE bit to 0.

Bits 1 and 0—Clock Enable 1 and 0 (CKE1, CKE0): These bits are used to select the SCI clock source and enable or disable clock output from the SCK pin. The combination of the CKE1 and CKE0 bits determines whether the SCK pin functions as the serial clock output pin or the serial clock input pin.

The setting of the CKE0 bit, however, is only valid for internal clock operation (CKE1 = 0) in asynchronous mode. The CKE0 bit setting is invalid in synchronous mode and in the case of external clock operation (CKE1 = 1). The CKE1 and CKE0 bits must be set before determining the SCI's operating mode with SCSMR.

For details of clock source selection, see table 14.9 in section 14.3, Operation.

Bit 1: CKE1	Bit 0: CKE0	Description	
0	0	Asynchronous mode	Internal clock/SCK pin functions as input pin (input signal ignored)*1
		Synchronous mode	Internal clock/SCK pin functions as serial clock output*1
	1	Asynchronous mode	Internal clock/SCK pin functions as clock output*2
		Synchronous mode	Internal clock/SCK pin functions as serial clock output
1	*	Asynchronous mode	External clock/SCK pin functions as clock input*3
		Synchronous mode	External clock/SCK pin functions as serial clock input

Legend:

*: Don't care

Notes: 1. Initial value

2. Outputs a clock with a frequency of 16/8/4 times the bit rate.

3. Inputs a clock with a frequency 16/8/4 times the bit rate.

4. Don't care

14.2.7 Serial Status 1 Register (SC1SSR)

Bit:	15	14	13	12	11	10	9	8
	PER3	PER2	PER1	PER0	FER3	FER2	FER1	FER0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit:	7	6	5	4	3	2	1	0
	TDFE	RDF	ORER	FER	PER	TEND	MPB	MPBT
Initial value:	1	0	0	0	0	1	0	0
R/W:	R/(W)*	R/(W)*	R/(W)*	R	R	R	R	R/W

Note: * Only 0 can be written, to clear the flag.

The serial status 1 register (SC1SSR) is a 16-bit register in which the lower 8 bits consist of status flags that indicate the operating status of the SCI plus the multiprocessor bit, and the upper 8 bits indicate the number of receive errors in the data in the receive FIFO register.

SC1SSR can be read or written to at all times. However, 1 cannot be written to the TDFE, RDF, ORER, FER, PER, and TEND flags. Also note that in order to clear the TDFE, RDF, and ORER flags to 0, they must first be read as 1. The FER, PER, TEND, and MPB flags are read-only and cannot be modified.

SC1SSR is initialized to H'0084 by a reset, in module standby mode, and in standby mode.

Bits 15 to 12—Number of Parity Errors (PER3 to PER0): These bits indicate the number of data bytes in which a parity error occurred in the receive data in the receive FIFO data register.

These bits are cleared by reading all the receive data in the receive FIFO data register or setting the RFRST bit to 1 in SCFCR to reset the receive FIFO data register to the empty state.

Bits 11 to 8—Number of Framing Errors (FER3 to FER0): These bits indicate the number of data bytes in which a framing error occurred in the receive data in the receive FIFO data register.

These bits are cleared by reading all the receive data in the receive FIFO data register or setting the RFRST bit to 1 in SCFCR to reset the receive FIFO data register to the empty state.

Bit 7—Transmit FIFO Data Register Empty (TDFE): Indicates that data has been transferred from the transmit FIFO data register (SCFTDR) to the transmit shift register (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 the FIFO control register (SCFCR), and transmit data can be written to SCFTDR.

Bit 7: TDFE	Description
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, and 0 is written to TDFE after reading TDFE = 1 • When transmit data exceeding the transmit trigger set number is written to SCFTDR by the on-chip DMAC
1	The number of transmit data bytes in SCFTDR does not exceed the transmit trigger set number (Initial value) [Setting conditions] • In a reset, in standby mode • When the number of SCFTDR transmit data bytes falls to or below the transmit trigger set number as the result of a transmit operation*

Note: * As SCFTDR is a 16-byte FIFO register, the maximum number of bytes that can be written when TDFE = 0 is {16 – (transmit trigger set number)}. Data written in excess of this will be ignored. The number of data bytes in SCFTDR is indicated by the upper 8 bits of SCFDR.

Bit 6—Receive FIFO Data Register Full (RDF): Indicates that the received data has been transferred to the receive FIFO data register (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 the FIFO control register (SCFCR).

Bit 6: RDF	Description
0	The number of receive data bytes in SCFRDR is less than the receive trigger set number (Initial value) [Clearing conditions] • In a reset or in standby mode • When SCFRDR is read until the number of receive data bytes in SCFRDR falls below the receive trigger set number, and 0 is written to RDF after reading RDF = 1 • When SCFRDR is read by the on-chip 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*

Note: * SCFRDR is a 16-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 the lower 8 bits of SCFDR.

Bit 5—Overrun Error (ORER): Indicates that an overrun error occurred during reception, causing abnormal termination.

Bit 5: ORER	Description
0	Reception in progress, or reception has ended normally*1 (Initial value) [Clearing conditions] • In a reset or in standby mode • When 0 is written to ORER after reading ORER = 1
1	An overrun error occurred during reception*2 [Setting condition] When the next serial receive operation is completed while there are 16 receive data bytes in SCFRDR

Notes: 1. The ORER flag is not affected and retains its previous state when the RE bit in SCSCR is cleared to 0.

2. 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. Also, serial transmission cannot be continued in synchronous mode.

Bit 4—Framing Error (FER): Indicates a framing error in the data read from the receive FIFO data register (SCFRDR).

Bit 4: FER	Description
0	There is no framing error in the receive data read from SCFRDR (Initial value) [Clearing conditions] • In a reset or in standby mode • When there is no framing error in SCFRDR read data
1	There is a framing error in the receive data read from SCFRDR [Setting condition] When there is a framing error in SCFRDR read data

Bit 3—Parity Error (PER): In asynchronous mode, indicates a parity error in the data read from the receive FIFO data register (SCFRDR).

Bit 3: PER	Description
0	There is no parity error in the receive data read from SCFRDR (Initial value) [Clearing conditions] • In a reset or in standby mode • When there is no parity error in SCFRDR read data
1	There is a parity error in the receive data read from SCFRDR [Setting condition] When there is a parity error in SCFRDR read data

Bit 2—Transmit End (TEND): Indicates that there is no valid data in SCFTDR when the last bit of the transmit character is sent, and transmission has been ended.

Bit 2: TEND	Description
0	Transmission is in progress [Clearing condition] When data is written to SCFTDR while TE = 1
1	Transmission has been ended (Initial value) [Setting conditions] • In a reset or in standby mode • 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

Bit 1—Multiprocessor bit (MPB): When reception is performed using a multiprocessor format in asynchronous mode, MPB stores the multiprocessor bit in the receive data.

The MPB flag is read-only and cannot be modified.

Bit 1: MPB	Description
0	Data with a 0 multiprocessor bit has been received* (Initial value)
1	Data with a 1 multiprocessor bit has been received

Note: * Retains its previous state when the RE bit is cleared to 0 while using a multiprocessor format.

Bit 0—Multiprocessor Bit Transfer (MPBT): When transmission is performed using a multiprocessor format in asynchronous mode, MPBT stores the multiprocessor bit to be added to the transmit data.

The MPBT bit setting is invalid in synchronous mode and IrDA mode, when a multiprocessor format is not used, and when the operation is not transmission.

Bit 0: MPBT	Description	
0	Data with a 0 multiprocessor bit is transmitted	(Initial value)
1	Data with a 1 multiprocessor bit is transmitted	

14.2.8 Serial Status 2 Register (SC2SSR)

Bit:	7	6	5	4	3	2	1	0
	TLM	RLM	N1	N0	BRK	DR	EI	ER
Initial value:	0	0	1	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/(W)*	R/(W)*	R/W	R/(W)*

Note: * Only 0 can be written, to clear the flag.

The serial status 2 register (SC2SSR) is an 8-bit register.

SC2SSR can be read or written to at all times. However, 1 cannot be written to the BRK, DR, and ER flags. Also note that in order to clear these flags to 0, they must first be read as 1. SC2SSR is initialized to H'20 by a reset, in module standby mode, and in standby mode.

Bit 7—Transmit LSB/MSB-First Select (TLM): Selects LSB-first or MSB-first mode in data transmission.

Bit 7: TLM	Description	
0	LSB-first transmission	(Initial value)
1	MSB-first transmission	

Note: When data is transmitted by 7-bit data length in asynchronous mode, MSB (bit 7) of the data is not transmitted in LSB-first transmission mode, and LSB (bit 0) of the data is not transmitted in MSB-first transmission mode.

Bit 6—Receive LSB/MSB-First Select (RLM): Selects LSB-first or MSB-first mode in data reception.

Bit 6: RLM	Description
0	LSB-first reception (Initial value)
1	MSB-first reception

Note: When data is received by 7-bit data length in asynchronous mode, MSB (bit 7) of the received data is 0 in LSB-first reception mode, and LSB (bit 0) of the received data is 0 in MSB-first reception mode.

Bits 5 and 4—Clock Bit Rate Ratio (N1, N0): These bits select the ratio of the base clock to the bit rate.

Bit 5: N1	Bit 4: N0	Description
0	0	SCI operates on base clock of 4 times the bit rate
	1	SCI operates on base clock of 8 times the bit rate
1	0	SCI operates on base clock of 16 times the bit rate (Initial value)
	1	Setting prohibited

Bit 3—Break Detect (BRK): Indicates that a receive data break signal has been detected.

Bit 3: BRK	Description
0	A break signal has not been received (Initial value) [Clearing conditions] • In a reset or in standby mode • When 0 is written to BRK after reading BRK = 1
1	A break signal has been received* [Setting condition] When data with a framing error is received, and a framing error also occurs in the next receive data (all space “0”)

Note: * 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.

Bit 2—Receive Data Ready (DR): Indicates that there are fewer than the receive trigger set number of data bytes in the receive FIFO data register (SCFRDR), and no further data has arrived for at least 15 etu after the stop bit of the last data received.

Bit 2: DR	Description
0	Reception is in progress or has ended normally and there is no receive data left in SCFRDR (Initial value) [Clearing conditions] • In a reset or in standby mode • When 0 is written to DR after reading all remaining receive data and the state of DR = 1*1
1	No further receive data has arrived, and SCFRDR contains fewer than the receive trigger set number of data bytes [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*2

Notes: 1. All remaining receive data should be read before clearing the DR flag.
2. Equivalent to 1.5 frames with an 8-bit, 1-stop-bit format.
etu: Elementary time unit = sec/bit

Bit 1—Receive Data Error Ignore Enable (EI): Selects whether or not the receive operation is to be continued when a framing error or parity error occurs in receive data (ER = 1).

Bit 1: EI	Description
0	Receive operation is halted when framing error or parity error occurs during reception (ER = 1) (Initial value)
1	Receive operation is continued when framing error or parity error occurs during reception (ER = 1)

Note: When EI = 0, only the last data in SCFRDR is treated as data containing an error. When EI = 1, receive data is sent to SCFRDR even if it contains an error.

Bit 0—Receive Error (ER): Indicates that a framing error, parity error, or overrun error occurred during reception.

Bit 0: ER	Description
0	Reception in progress, or reception has ended normally*[1] (Initial value) [Clearing conditions] • In a reset or in standby mode • When 0 is written to ER after reading ER = 1
1	A framing error, parity error, or overrun error occurred during reception [Setting conditions] • When the SCI 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 the serial mode register (SCSMR) • When the next serial receive operation is completed while there are 16 receive data bytes in SCFRDR

Notes: 1. The ER flag is not affected and retains its previous state when the RE bit in SCSCR is cleared to 0. When a framing error or parity error occurs, the receive data is still transferred to SCFRDR, and reception is then halted or continued according to the setting of the EI bit. When an overrun error occurs, the receive data is not transferred to SCFRDR and reception cannot be continued.

2. In 2-stop-bit mode, only the first stop bit is checked for a value of 1; the second bit is not checked.

14.2.9 Bit Rate Register (SCBRR)

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

The bit rate register (SCBRR) is an 8-bit register that sets the serial transmit/receive bit rate in accordance with the baud rate generator operating clock selected by bits CKS1 and CKS0 in the serial mode register (SCSMR).

SCBRR can be read or written to by the CPU at all times.

SCBRR is initialized to H'FF by a reset, by the module standby function, and in software standby mode and hardware standby mode.

The SCBRR setting is found from the following equations.

Asynchronous mode:

$$N = \frac{P\phi}{64 \times 2^{2n-1} \times B} \times 10^6 - 1 \quad \text{(When operating on a base clock of 16 times the bit rate)}$$

$$N = \frac{P\phi}{32 \times 2^{2n-1} \times B} \times 10^6 - 1 \quad \text{(When operating on a base clock of 8 times the bit rate)}$$

$$N = \frac{P\phi}{16 \times 2^{2n-1} \times B} \times 10^6 - 1 \quad \text{(When operating on a base clock of 4 times the bit rate)}$$

Synchronous mode:

$$N = \frac{P\phi}{8 \times 2^{2n-1} \times B} \times 10^6 - 1$$

Where B: Bit rate (bits/s)
N: SCBRR setting for baud rate generator ($0 \leq N \leq 255$)
Pϕ: Peripheral module operating frequency (MHz)
n: Baud rate generator input clock (n = 0 to 3)
(See the table below for the relation between n and the clock.)

		SCSMR Settings	
n	**Clock**	**CKS1**	**CKS0**
0	Pϕ	0	0
1	Pϕ/4	0	1
2	Pϕ/16	1	0
3	Pϕ/64	1	1

The bit rate error in asynchronous mode is found from the following equations:

$$\text{Error (\%)} = \left\{ \frac{P\phi \times 10^6}{(N + 1) \times B \times 64 \times 2^{2n-1}} - 1 \right\} \times 100$$

(When operating on a base clock of 16 times the bit rate)

$$\text{Error (\%)} = \left\{ \frac{P\phi \times 10^6}{(N + 1) \times B \times 32 \times 2^{2n-1}} - 1 \right\} \times 100$$

(When operating on a base clock of 8 times the bit rate)

$$\text{Error (\%)} = \left\{ \frac{P\phi \times 10^6}{(N + 1) \times B \times 16 \times 2^{2n-1}} - 1 \right\} \times 100$$

(When operating on a base clock of 4 times the bit rate)

Table 14.3 shows sample SCBRR settings in asynchronous mode, and table 14.4 shows sample SCBRR settings in synchronous mode.

Table 14.3 Examples of Bit Rates and SCBRR Settings in Asynchronous Mode

	Pφ (MHz)											
	2			2.097152			2.4576			3		
Bit Rate (Bits/s)	**n**	**N**	**Error (%)**	**n**	**N**	**Error (%)**	**n**	**N**	**Error (%)**	**n**	**N**	**Error (%)**
110	1	141	0.03	1	148	–0.04	1	174	–0.26	1	212	0.03
150	1	103	0.16	1	108	0.21	1	127	0.00	1	155	0.16
300	0	207	0.16	0	217	0.21	0	255	0.00	1	77	0.16
600	0	103	0.16	0	108	0.21	0	127	0.00	0	155	0.16
1200	0	51	0.16	0	54	–0.70	0	63	0.00	0	77	0.16
2400	0	25	0.16	0	26	1.14	0	31	0.00	0	38	0.16
4800	0	12	0.16	0	13	–2.48	0	15	0.00	0	19	–2.34
9600	0	6	–6.99	0	6	–2.48	0	7	0.00	0	9	–2.34
19200	0	2	8.51	0	2	13.78	0	3	0.00	0	4	–2.34
31250	0	1	0.00	0	1	4.86	0	1	22.88	0	2	0.00
38400	0	1	–18.62	0	1	–14.67	0	1	0.00	—	—	—

	Pφ (MHz)											
	3.6864			4			4.9152			5		
Bit Rate (Bits/s)	**n**	**N**	**Error (%)**	**n**	**N**	**Error (%)**	**n**	**N**	**Error (%)**	**n**	**N**	**Error (%)**
110	2	64	0.70	2	70	0.03	2	86	0.31	2	88	–0.25
150	1	191	0.00	1	207	0.16	1	255	0.00	2	64	0.16
300	1	95	0.00	1	103	0.16	1	127	0.00	1	129	0.16
600	0	191	0.00	0	207	0.16	0	255	0.00	1	64	0.16
1200	0	95	0.00	0	103	0.16	0	127	0.00	0	129	0.16
2400	0	47	0.00	0	51	0.16	0	63	0.00	0	64	0.16
4800	0	23	0.00	0	25	0.16	0	31	0.00	0	32	–1.36
9600	0	11	0.00	0	12	0.16	0	15	0.00	0	15	1.73
19200	0	5	0.00	0	6	–6.99	0	7	0.00	0	7	1.73
31250	—	—	—	0	3	0.00	0	4	–1.70	0	4	0.00
38400	0	2	0.00	0	2	8.51	0	3	0.00	0	3	1.73

	Pφ (MHz)											
	6			6.144			7.37288			8		
Bit Rate (Bits/s)	**n**	**N**	**Error (%)**	**n**	**N**	**Error (%)**	**n**	**N**	**Error (%)**	**n**	**N**	**Error (%)**
110	2	106	–0.44	2	108	0.08	2	130	–0.07	2	141	0.03
150	2	77	0.16	2	79	0.00	2	95	0.00	2	103	0.16
300	1	155	0.16	1	159	0.00	1	191	0.00	1	207	0.16
600	1	77	0.16	1	79	0.00	1	95	0.00	1	103	0.16
1200	0	155	0.16	0	159	0.00	0	191	0.00	0	207	0.16
2400	0	77	0.16	0	79	0.00	0	95	0.00	0	103	0.16
4800	0	38	0.16	0	39	0.00	0	47	0.00	0	51	0.16
9600	0	19	–2.34	0	19	0.00	0	23	0.00	0	25	0.16
19200	0	9	–2.34	0	9	0.00	0	11	0.00	0	12	0.16
31250	0	5	0.00	0	5	2.40	0	6	5.33	0	7	0.00
38400	0	4	–2.34	0	4	0.00	0	5	0.00	0	6	–6.99

	Pφ (MHz)											
	9.8304			10			12			12.288		
Bit Rate (Bits/s)	**n**	**N**	**Error (%)**	**n**	**N**	**Error (%)**	**n**	**N**	**Error (%)**	**n**	**N**	**Error (%)**
110	2	174	–0.26	2	177	–0.25	2	212	0.03	2	217	0.08
150	2	127	0.00	2	129	0.16	2	155	0.16	2	159	0.00
300	1	255	0.00	2	64	0.16	2	77	0.16	2	79	0.00
600	1	127	0.00	1	129	0.16	1	155	0.16	1	159	0.00
1200	0	255	0.00	1	64	0.16	1	77	0.16	1	79	0.00
2400	0	127	0.00	0	129	0.16	0	155	0.16	0	159	0.00
4800	0	63	0.00	0	64	0.16	0	77	0.16	0	79	0.00
9600	0	31	0.00	0	32	–1.36	0	38	0.16	0	39	0.00
19200	0	15	0.00	0	15	1.73	0	19	0.16	0	19	0.00
31250	0	9	–1.70	0	9	0.00	0	11	0.00	0	11	2.40
38400	0	7	0.00	0	7	1.73	0	9	–2.34	0	9	0.00

	Pφ (MHz)								
	14.7456			**16**			**30**		
Bit Rate (Bits/s)	**n**	**N**	**Error (%)**	**n**	**N**	**Error (%)**	**n**	**N**	**Error (%)**
110	3	64	0.70	3	70	0.03	3	132	0.13
150	2	191	0.00	2	207	0.16	3	97	–0.35
300	2	95	0.00	2	103	0.16	2	194	0.16
600	1	191	0.00	1	207	0.16	2	97	–0.35
1200	1	95	0.00	1	103	0.16	1	194	0.16
2400	0	191	0.00	0	207	0.16	1	97	–0.35
4800	0	95	0.00	0	103	0.16	0	197	0.16
9600	0	47	0.00	0	51	0.16	0	97	–0.35
19200	0	23	0.00	0	25	0.16	0	48	–0.35
31250	0	14	–1.70	0	15	0.00	0	29	0.00
38400	0	11	0.00	0	12	0.16	0	23	1.73

Table 14.4 Examples of Bit Rates and SCBRR Settings in Synchronous Mode

	Pϕ (MHz)					
	4		**8**		**16**	
Bit Rate (Bits/s)	**n**	**N**	**n**	**N**	**n**	**N**
110	—	—	—	—	—	—
250	2	249	3	124	3	249
500	2	124	2	249	3	124
1 k	1	249	2	124	2	249
2.5 k	1	99	1	199	2	99
5 k	0	199	1	99	1	199
10 k	0	99	0	199	1	99
25 k	0	39	0	79	0	159
50 k	0	19	0	39	0	79
100 k	0	9	0	19	0	39
250 k	0	3	0	7	0	15
500 k	0	1	0	3	0	7
1 M	0	0*	0	1	0	3
2 M			0	0*	0	1

Legend:

Blank: No setting is available.

—: A setting is available but error occurs.

Notes: As far as possible, the setting should be made so that the error is within 1%.

* Continuous transmission/reception is not possible.

Table 14.5 shows the maximum bit rate for various frequencies in asynchronous mode when using the baud rate generator. Tables 14.6 and 14.7 show the maximum bit rates when using external clock input.

Table 14.5 Maximum Bit Rate for Various Frequencies with Baud Rate Generator (Asynchronous Mode)

		Settings	
Pϕ (MHz)	**Maximum Bit Rate (Bits/s)**	**n**	**N**
2	62500	0	0
2.097152	65536	0	0
2.4576	76800	0	0
3	93750	0	0
3.6864	115200	0	0
4	125000	0	0
4.9152	153600	0	0
8	250000	0	0
9.8304	307200	0	0
12	375000	0	0
14.7456	460800	0	0
16	500000	0	0
19.66080	614400	0	0
20	625000	0	0
24	750000	0	0
24.57600	768000	0	0
28	896875	0	0
30	937500	0	0

Table 14.6 Maximum Bit Rate with External Clock Input (Asynchronous Mode)

Pϕ (MHz)	External Input Clock (MHz)	Maximum Bit Rate (Bits/s)
2	0.5000	31250
2.097152	0.5243	32768
2.4576	0.6144	38400
3	0.7500	46875
3.6864	0.9216	57600
4	1.0000	62500
4.9152	1.2288	76800
8	2.0000	125000
9.8304	2.4576	153600
12	3.0000	187500
14.7456	3.6864	230400
16	4.0000	250000
30	7.5000	468750

Table 14.7 Maximum Bit Rate with External Clock Input (Synchronous Mode)

Pϕ (MHz)	External Input Clock (MHz)	Maximum Bit Rate (Bits/s)
8	1.3333	1333333.3
16	2.6667	2666666.7
30	5.0	5000000.0

14.2.10 FIFO Control Register (SCFCR)

Bit:	7	6	5	4	3	2	1	0
	RTRG1	RTRG0	TTRG1	TTRG0	—	TFRST	RFRST	LOOP
Initial value:	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

The FIFO control register (SCFCR) performs data count resetting and trigger data number setting for the transmit and receive FIFO registers, and also contains a loopback test enable bit.

SCFCR can be read or written to at all times.

SCFCR is initialized to H'00 by a reset, by the module standby function, and in software standby mode and hardware standby mode.

Bits 7 and 6—Receive FIFO Data Number Trigger (RTRG1, RTRG0): These bits are used to set the number of receive data bytes that sets the receive data full (RDF) flag in the serial status 1 register (SC1SSR).

The RDF flag is set when the number of receive data bytes in the receive FIFO data register (SCFRDR) is equal to or greater than the trigger set number shown in the following table.

Bit 7: RTRG1	Bit 6: RTRG0	Receive Trigger Number	
0	0	1	(Initial value)
	1	4	
1	0	8	
	1	14	

Bits 5 and 4—Transmit FIFO Data Number Trigger (TTRG1, TTRG0): These bits are used to set the number of remaining transmit data bytes that sets the transmit FIFO data register empty (TDFE) flag in the serial status 1 register (SC1SSR).

The TDFE flag is set when the number of transmit data bytes in the transmit FIFO data register (SCFTDR) is equal to or less than the trigger set number shown in the following table.

Bit 5: TTRG1	Bit 4: TTRG0	Transmit Trigger Number
0	0	8 (8)*
	1	4 (12)
1	0	2 (14)
	1	1 (15)

Note: * Initial value. Figures in parentheses are the number of empty bytes in SCFTDR when the flag is set.

Bit 3—Reserved: This bit is always read as 0 and should only be written with 0.

Bit 2—Transmit FIFO Data Register Reset (TFRST): Invalidates the transmit data in the transmit FIFO data register and resets it to the empty state.

Bit 2: TFRST	Description	
0	Reset operation disabled*	(Initial value)
1	Reset operation enabled	

Note: * A reset operation is performed in the event of a reset or in standby mode.

Bit 1—Receive FIFO Data Register Reset (RFRST): Invalidates the receive data in the receive FIFO data register and resets it to the empty state.

Bit 1: RFRST	Description	
0	Reset operation disabled*	(Initial value)
1	Reset operation enabled	

Note: * A reset operation is performed in the event of a reset or in standby mode.

Bit 0—Loopback Test (LOOP): Internally connects the transmit output pin (TxD) and receive output pin (RxD), enabling loopback testing.

Bit 0: LOOP	Description	
0	Loopback test disabled	(Initial value)
1	Loopback test enabled	

14.2.11 FIFO Data Count Register (SCFDR)

The FIFO data count register (SCFDR) is a 16-bit register that indicates the number of data bytes stored in the transmit FIFO data register (SCFTDR) and receive FIFO data register (SCFRDR).

The upper 8 bits show the number of transmit data bytes in SCFTDR, and the lower 8 bits show the number of receive data bytes in SCFRDR. SCFDR is initialized to H'00 by a reset, in module standby mode, and in standby mode. It is also initialized to H'00 by setting the TFRST and RFRST bits to 1 in SCFCR to reset SCFTDR and SCFRDR to the empty state.

SCFDR can be read by the CPU at all times.

Upper 8 bits:	7	6	5	4	3	2	1	0
	—	—	—	T4	T3	T2	T1	T0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bits 15 to 13—Reserved: These bits are always read as 0 and should only be written with 0.

Bits 12 to 8—Transmit FIFO Data Count (T4 to T0): These bits show the number of untransmitted data bytes in SCFTDR. A value of H'00 indicates that there is no transmit data, and a value of H'10 indicates that SCFTDR is full of transmit data. The value is cleared to H'00 by transmitting all the data, as well as by the above initialization conditions.

Lower 8 bits:	7	6	5	4	3	2	1	0
	—	—	—	R4	R3	R2	R1	R0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bits 7 to 5—Reserved: These bits are always read as 0 and should only be written with 0.

Bits 4 to 0—Receive FIFO Data Count (R4 to R0): These bits show the number of receive data bytes in SCFRDR. A value of H'00 indicates that there is no receive data, and a value of H'10 indicates that SCFRDR is full of receive data. The value is cleared to H'00 by reading all the receive data from SCFRDR, as well as by the above initialization conditions.

14.2.12 FIFO Error Register (SCFER)

The FIFO error register (SCFER) indicates the data location at which a parity error or framing error occurred in receive data stored in the receive FIFO data register (SCFRDR).

SCFER can be read at all times.

Upper 8 bits:	15	14	13	12	11	10	9	8
	ED15	ED14	ED13	ED12	ED11	ED10	ED9	ED8
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Lower 8 bits:	7	6	5	4	3	2	1	0
	ED7	ED6	ED5	ED4	ED3	ED2	ED1	ED0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bits 15 to 0—Error Data Flags (ED15 to ED0): These flags indicate the data location in the receive FIFO data register at which an error occurred. When data in the nth stage of the buffer contains an error, the nth bit is set to 1. Note that this register is not cleared by setting the RFRST bit to 1 in SCFCR. To clear this register, read all the receive data in which the error occurred from the SCFRDR register before setting the RFRST bit to 1 in SCFCR to clear SCFRDR.

Bits 15 to 0: ED15 to ED0	Description
0	No parity or framing error in data in corresponding stage of register FIFO (Initial value)
1	Parity or framing error present in data in corresponding stage of register FIFO

Note: A reset operation is performed in the event of a reset, when the module standby function is initiated, or in standby mode. Also, these flags are cleared by reading the data in which the parity error or framing error occurred from SCFRDR.

14.2.13 IrDA Mode Register (SCIMR)

The IrDA mode register (SCIMR) allows selection of the IrDA mode and the IrDA output pulse width, and inversion of the IrDA receive data polarity.

SCIMR can be read and written to at all times.

SCIMR is initialized to H'00 by a reset, by the module standby function, and in software standby mode and hardware standby mode.

Bit:	7	6	5	4	3	2	1	0
	IRMOD	PSEL	RIVS	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R	R	R	R	R

Bit 7—IrDA Mode (IRMOD): Selects operation as an IrDA serial communication interface.

Bit 7: IRMOD	Description	
0	Operation as SCI is selected	(Initial value)
1	Operation as IrDA is selected*	

Note: * When operation as an IrDA interface is selected, bit 7 ($C/\overline{A}$) of the serial mode register (SCSMR) must be cleared to 0.

Bit 6—Output Pulse Width Select (PSEL): Selects either 3/16 of the bit length set by bits ICK3 to ICK0 in the serial status 1 register (SC1SSR), or 3/16 of the bit length corresponding to the selected bit rate, as the IrDA output pulse width. The setting is shown together with bits 6 to 3 (ICK3 to ICK0) of the serial mode register (SCSMR).

Serial Mode Register (SCSMR)				SCIMR	
Bit 6: ICK3	**Bit 5: ICK2**	**BIT 4: ICK1**	**Bit 3: ICK0**	**Bit 2: PSEL**	**Description**
ICK3	ICK2	ICK1	ICK0	1	Pulse width: 3/16 of bit length set in bits ICK3 to ICK0
Don’t care	Don’t care	Don’t care	Don’t care	0	Pulse width: 3/16 of bit length set in SCBRR

Note: A fixed clock pulse signal, IRCLK, must be generated by multiplying the Pϕ clock by 1/(2N + 2) (where N is determined by the value set in ICK3 to ICK0). For details, see Pulse Width Selection in section 14.3.6, Operation in IrDA Mode.

Bit 5—IrDA Receive Data Inverse (RIVS): Allows inversion of the receive data polarity to be selected in IrDA communication.

Bit 5: RIVS	Description	
0	Receive data polarity inverted in reception	(Initial value)
1	Receive data polarity not inverted in reception	

Note: Make the selection according to the characteristics of the IrDA modulation/demodulation module.

Bits 4 to 0—Reserved: These bits are always read as 0 and should only be written with 0.

14.3 Operation

14.3.1 Overview

The SCI can carry out serial communication in two modes: asynchronous mode in which synchronization is achieved character by character, and synchronous mode in which synchronization is achieved with clock pulses.

An IrDA block is also provided, enabling infrared communication conforming to IrDA 1.0 to be executed by connecting an infrared transmission/reception unit.

Sixteen-stage FIFO buffers are provided for both transmission and reception, reducing the CPU overhead and enabling fast, continuous communication to be performed.

Selection of asynchronous, synchronous, or IrDA mode and the transmission format is made by means of the serial mode register (SCSMR) and IrDA mode register (SCIMR) as shown in table 14.8. The SCI clock source is determined by a combination of the $C/\overline{A}$ bit in SCSMR, the IRMOD bit in SCIMR, and the CKE1 and CKE0 bits in the serial control register (SCSCR), as shown in table 14.9.

- Asynchronous Mode
  - Data length: Choice of 7 or 8 bits
  - Choice of parity addition, multiprocessor bit addition, and addition of 1 or 2 stop bits (the combination of these parameters determines the transmit/receive format and character length)
  - Detection of framing, parity, and overrun errors, receive FIFO data full and receive data ready conditions, and breaks, during reception
  - Detection of transmit FIFO data empty condition during transmission
  - Choice of internal or external clock as SCI clock source

    When internal clock is selected: The SCI operates on a clock with a frequency of 16, 8, or 4 times the bit rate of the baud rate generator, and can output this operating clock.

    When external clock is selected: A clock with a frequency of 16, 8, or 4 times the bit rate must be input (the built-in baud rate generator is not used).

- Synchronous Mode
  - Transfer format: Fixed 8-bit data
  - Detection of overrun errors during reception
  - Choice of internal or external clock as SCI clock source

    When internal clock is selected: The SCI operates on the baud rate generator clock and can output a serial clock to external devices.

    When external clock is selected: The on-chip baud rate generator is not used, and the SCI operates on the input serial clock.

- IrDA Mode
  - IrDA 1.0 compliance
  - Data length: 8 bits
  - Stop bit length: 1 bit
  - Protection function to prevent receiver being affected during transmission
  - Clock source: Internal clock

Table 14.8 SCSMR and SCIMR Settings for Serial Transmit/Receive Format Selection

SCIMR	SCSMR Settings						SCI Transmit/Receive Format			
Bit 7: IRMOD	**Bit 7: C/$\overline{A}$**	**Bit 6: CHR**	**Bit 2: MP**	**Bit 5: PE**	**Bit 3: STOP**	**Mode**	**Data Length**	**MP Bit**	**Parity Bit**	**Stop Bit Length**
0	0	0	0	0	0	Asynchronous mode	8-bit data	Absent	Absent	1 bit
					1					2 bits
				1	0				Present	1 bit
					1					2 bits
		1		0	0		7-bit data		Absent	1 bit
					1					2 bits
				1	0				Present	1 bit
					1					2 bits
		0	1	*	0	Asynchronous mode (multi-processor format)	8-bit data	Present	Absent	1 bit
				*	1					2 bits
		1		*	0		7-bit data			1 bit
				*	1					2 bits
0	1	*	*	*	*	Synchronous mode	8-bit data	Absent	Absent	None
1	0	ICK3	ICK2	ICK1	ICK0	IrDA mode	8-bit data	Absent	Absent	1 bit
	1	*	*	*	*	Setting prohibited	—	—	—	—

Legend:

*: Don’t care

Table 14.9 SCSMR and SCSCR Settings for SCI Clock Source Selection

SCSMR	SCSCR Setting			SCI Transmit/Receive Clock	
Bit 7: C/$\overline{A}$	Bit 1: CKE1	Bit 0: CKE0	Mode	Clock Source	SCK Pin Function
0	0	0	Asynchronous mode	Internal	SCI does not use SCK pin
		1			Outputs clock with frequency of 16/8/4 times bit rate
	1	0		External	Inputs clock with frequency of 16/8/4 times bit rate
		1			
1	0	0	Synchronous mode	Internal	Outputs serial clock
		1			
	1	0		External	Inputs serial clock
		1			

14.3.2 Operation in Asynchronous Mode

In asynchronous mode, characters are sent or received, each preceded by a start bit indicating the start of communication and followed by one or two stop bits indicating the end of communication. Serial communication is thus carried out with synchronization established on a character-by-character basis.

Inside the SCI, the transmitter and receiver are independent units, enabling full-duplex communication. Both the transmitter and the receiver also have a 16-stage FIFO buffer structure, so that data can be read or written during transmission or reception, enabling continuous data transfer.

Figure 14.3 shows the general format for asynchronous serial communication.

In asynchronous serial communication, the communication line is usually held in the mark state (high level). The SCI monitors the line, and when it goes to the space state (low level), recognizes a start bit and starts serial communication.

One serial communication character consists of a start bit (low level), followed by data (LSB-first or MSB-first order selectable), a parity bit (high or low level), and finally one or two stop bits (high level).

In asynchronous mode, the SCI performs synchronization at the falling edge of the start bit in reception. The SCI samples the data on the eighth, fourth, or second pulse of a clock with a frequency of 16, 8, or 4 times the length of one bit, so that the transfer data is latched at the center of each bit.

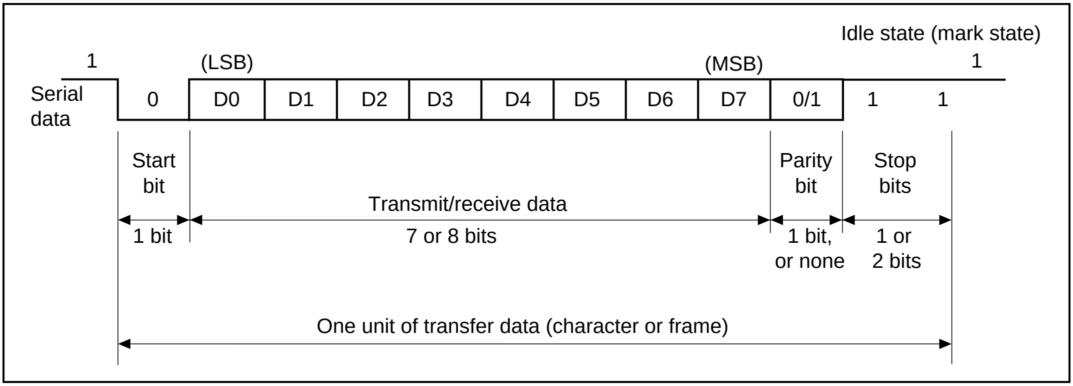


Figure 14.3 Data Format in Asynchronous Communication (Example with 8-Bit Data, Parity, Two Stop Bits, LSB-First Transfer)

Transmit/Receive Format

Table 14.10 shows the transmit/receive formats that can be used in asynchronous mode. Any of 12 transmit/receive formats can be selected by means of settings in the serial mode register (SCSMR).

Table 14.10 Serial Transmit/Receive Formats (Asynchronous Mode)

SCSMR Settings				Serial Transmit/Receive Format and Frame Length											
CHR	**PE**	**MP**	**STOP**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
0	0	0	0	S	8-bit data								STOP		
0	0	0	1	S	8-bit data								STOP	STOP	
0	1	0	0	S	8-bit data								P	STOP	
0	1	0	1	S	8-bit data								P	STOP	STOP
1	0	0	0	S	7-bit data							STOP			
1	0	0	1	S	7-bit data							STOP	STOP		
1	1	0	0	S	7-bit data							P	STOP		
1	1	0	1	S	7-bit data							P	STOP	STOP	
0	*	1	0	S	8-bit data								MPB	STOP	
0	*	1	1	S	8-bit data								MPB	STOP	STOP
1	*	1	0	S	7-bit data							MPB	STOP		
1	*	1	1	S	7-bit data							MPB	STOP	STOP	

Legend:
S: Start bit
STOP: Stop bit
P: Parity bit
MPB: Multiprocessor bit
*: Don't care

Clock

Either an internal clock generated by the built-in baud rate generator or an external clock input at the SCK pin can be selected as the SCI's serial clock, according to the setting of the $C/\overline{A}$ bit in SCSMR and the CKE1 and CKE0 bits in SCSCR. For details of SCI clock source selection, see table 14.9.

When an external clock is input at the SCK pin, the clock frequency should be 16, 8, or 4 times the bit rate used.

When the SCI is operated on an internal clock, the clock can be output from the SCK pin. The frequency of the clock output in this case is 16, 8, or 4 times the bit rate.

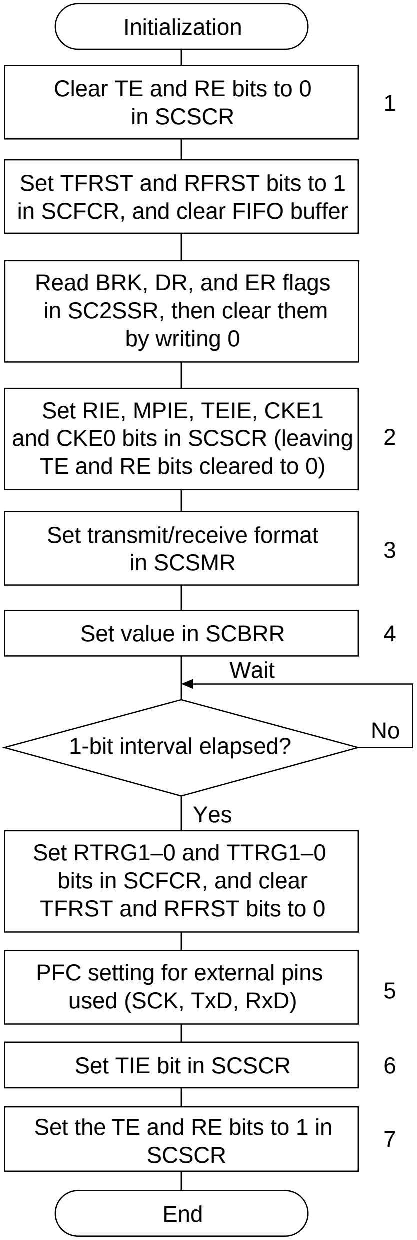
Data Transmit/Receive Operations

SCI Initialization (Asynchronous Mode): Before transmitting and receiving data, it is necessary to clear the TE and RE bits to 0 in SCSCR, then initialize the SCI as described below.

When the operating mode, communication format, etc., is changed, the TE and RE bits must be cleared to 0 before making the change using the following procedure. When the TE bit is cleared to 0, the transmit shift register (SCTSR) is initialized. Note that clearing the TE and RE bits to 0 does not change the contents of the serial status 1 register (SC1SSR), the transmit FIFO data register (SCFTDR), or the receive FIFO data register (SCFRDR). The TE bit should not be cleared to 0 until all transmit data has been transmitted and the TEND flag has been set in SC1SSR. It is possible to clear the TE bit to 0 during transmission, but the data being transmitted will go to the high-impedance state after TE is cleared. Also, before starting transmission by setting TE again, the TFRST bit should first be set to 1 in SCFCR to reset SCFTDR.

When an external clock is used the clock should not be stopped during operation, including initialization, since operation will be unreliable in this case.

Figure 14.4 shows a sample SCI initialization flowchart.



Initialization
Clear TE and RE bits to 0 in SCSCR
1
Set TFRST and RFRST bits to 1 in SCFCR, and clear FIFO buffer
Read BRK, DR, and ER flags in SC2SSR, then clear them by writing 0
Set RIE, MPIE, TEIE, CKE1 and CKE0 bits in SCSCR (leaving TE and RE bits cleared to 0)
2
Set transmit/receive format in SCSMR
3
Set value in SCBRR
4
Wait
1-bit interval elapsed?
No
Yes
Set RTRG1–0 and TTRG1–0 bits in SCFCR, and clear TFRST and RFRST bits to 0
PFC setting for external pins used (SCK, TxD, RxD)
5
Set TIE bit in SCSCR
6
Set the TE and RE bits to 1 in SCSCR
7
End

1. Clear bits TE and RE to 0 before end of initialization.
2. Set disabling/enabling of RXI, and TEI interrupt requests. When enabling an interrupt request, also make a setting in the IPRK register of the INTC.
3. Set the transmit/receive format in SCSMR.

   When using IrDA mode, also set SCIMR.
4. Write a value corresponding to the bit rate to the bit rate register (SCBRR). Not necessary if an external clock is used. Wait for at least one bit interval after making this setting.
5. Make PFC settings for the external pins to be used.

   Make a setting for RxD input when receiving, and for TxD output when transmitting. Make an SCK input/output setting according to the setting of bits CKE1 and CKE0.

   An SCK pin setting is not necessary when CKE1 and CKE0 are cleared to 0 in asynchronous mode.

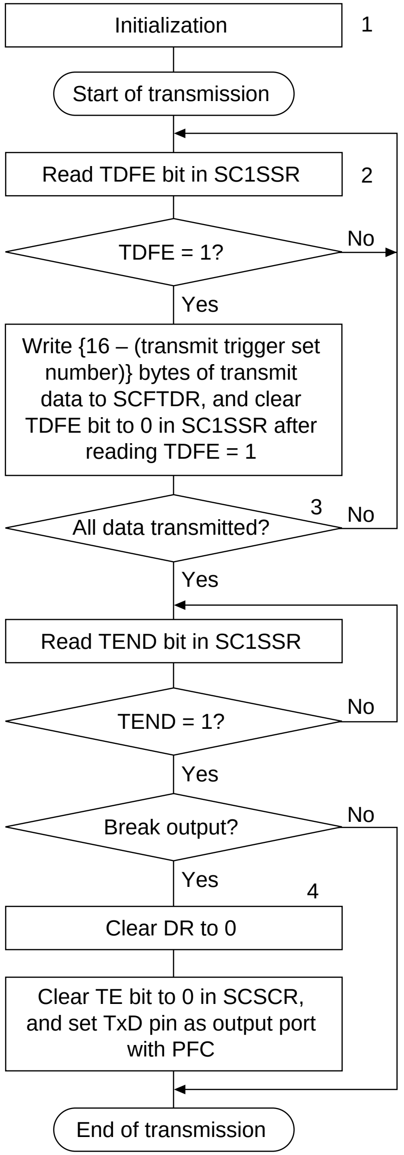
   When serial clock output is set, clock output from the SCK pin begins at this point.
6. When the TXI interrupt is used, first clear TFRST and RFRST in SCFCR to 0. Even if TE = 0, a TXI interrupt will be generated as soon as the TIE bit in SCSCR is set to 1.
7. Set the TE bit or RE bit in SCSCR to 1.

   Setting the TE and RE bits enables the TxD, RxD and SCK pins to be used. When transmitting, the TxD pin will go to the mark state; when receiving, RxD pin will go to the idle state, waiting for a start bit.

Figure 14.4 Sample SCI Initialization Flowchart

Serial Data Transmission (Asynchronous Mode): Figure 14.5 shows a sample flowchart for serial transmission.

Use the following procedure for serial data transmission after enabling the SCI for transmission.


Initialization
1
Start of transmission
Read TDFE bit in SC1SSR
2
TDFE = 1?
No
Yes
Write {16 – (transmit trigger set number)} bytes of transmit data to SCFTDR, and clear TDFE bit to 0 in SC1SSR after reading TDFE = 1
3
All data transmitted?
No
Yes
Read TEND bit in SC1SSR
TEND = 1?
No
Yes
Break output?
No
Yes
4
Clear DR to 0
Clear TE bit to 0 in SCSCR, and set TxD pin as output port with PFC
End of transmission

1. SCI initialization:

   See figure 14.4, Sample SCI Initialization Flowchart.

2. SCI status check and transmit data write:

   Read the serial status 1 register (SC1SSR) and check that the TDFE bit is set to 1, then write transmit data to the transmit FIFO data register (SCFTDR) and clear the TDFE bit to 0 after reading TDFE = 1. The TEND bit is cleared automatically when transmission is started by writing transmit data.

   The number of data bytes that can be written is {16 – (transmit trigger set number)}.

3. Serial transmission continuation procedure:

   To continue serial transmission, read 1 from the TDFE bit to confirm that writing is possible, then write data to SCFTDR, and then clear the TDFE bit to 0. (Checking and clearing of the TDFE bit is automatic when the DMAC is activated by a transmit-FIFO-data-empty interrupt (TXI) request, and data is written to SCFTDR.)

4. Break output at the end of serial transmission:

   To output a break in serial transmission, clear the port data register (DR) to 0, then clear the TE bit to 0 in SCSCR, and set the TxD pin as an output port with the PFC.

In steps 2 and 3, the number of transmit data bytes that can be written can be ascertained from the number of transmit data bytes in SCFTDR indicated in the upper 8 bits of the FIFO data count register (SCFTDR).

Figure 14.5 Sample Serial Transmission Flowchart

In serial transmission, the SCI operates as described below.

1. When data is written to the transmit FIFO data register (SCFTDR), the SCI transfers the data to the transmit shift register (SCTSR), and starts transmitting. Check that the TDFE flag is set to 1 in the serial status 1 register (SC1SSR) before writing transmit data to SCFTDR. The number of data bytes that can be written is at least {16 – (transmit trigger set number)}.

2. When data is transferred from SCFTDR to SCTSR and transmission is started, transmit operations are performed continually until there is no transmit data left in SCFTDR. If the number of data bytes in SCFTDR falls to or below the transmit trigger number set in the FIFO control register (SCFCR) during transmission, the TDFE flag is set. If the TIE bit setting in the serial control register (SCSCR) is 1 at this time, a transmit-FIFO-data-empty interrupt (TXI) is requested.

   The serial transmit data is sent from the 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 or MSB-first order according to the setting of the TLM bit in SC2SSR.
   c. Parity bit or multiprocessor bit: One parity bit (even or odd parity), or one multiprocessor bit is output. (A format in which neither a parity bit nor a multiprocessor bit is 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 SCI checks for transmit data in SCFTDR at the timing for sending the stop bit. If there is data in SCFTDR, it is transferred to SCTSR, the stop bit is sent, and then serial transmission of the next frame is started.

   If there is no transmit data in SCFTDR, the TEND flag is set to 1 in the serial status 1 register (SC1SSR), the stop bit is sent, and then the line goes to the mark state in which 1 is output continuously. If the TEIE bit setting in SCSCR is 1 at this time, a TEI interrupt is requested.

Figure 14.6 shows an example of the operation for transmission in asynchronous mode.

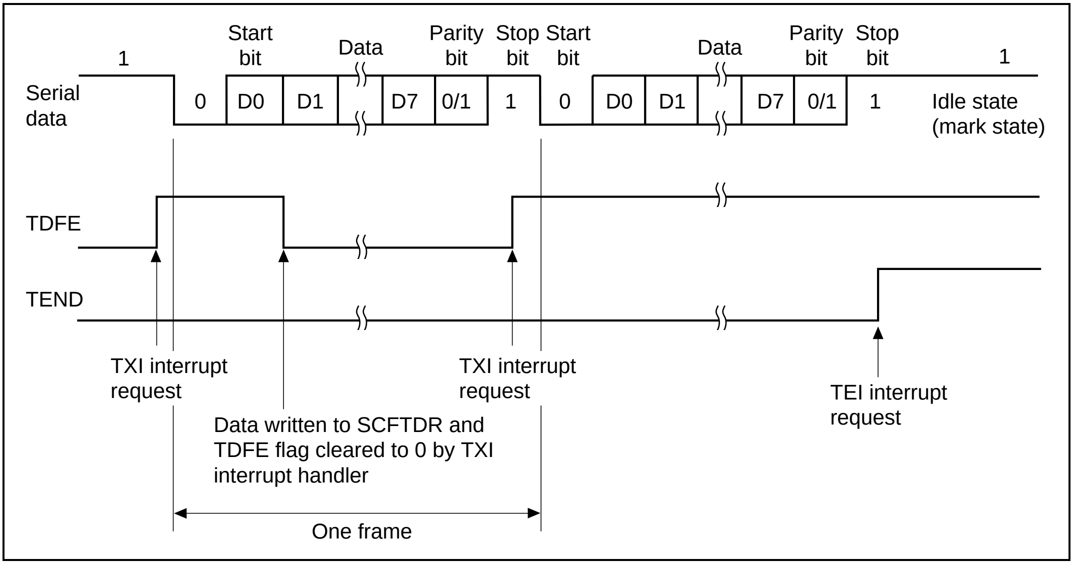
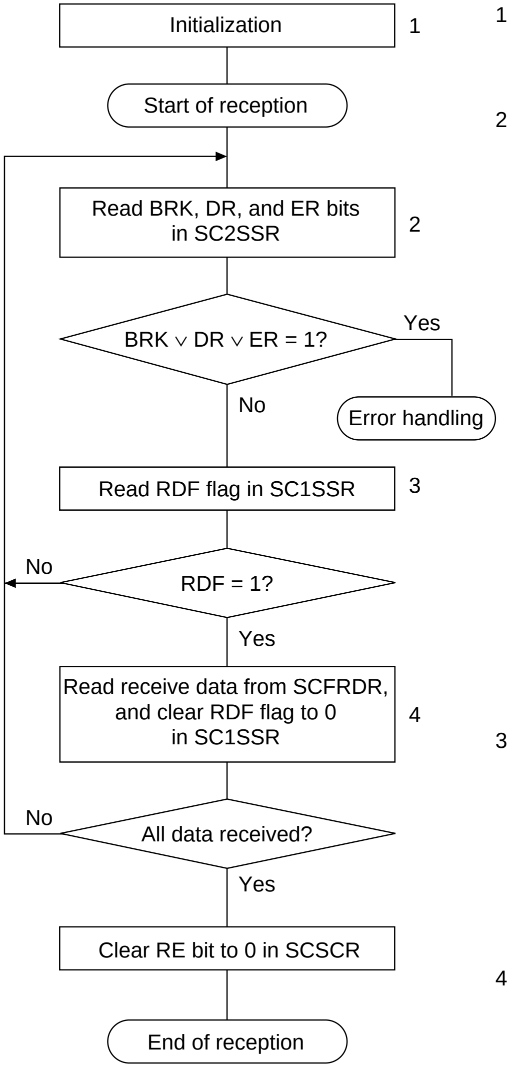


Figure 14.6 Example of Transmit Operation in Asynchronous Mode (Example with 8-Bit Data, Parity, One Stop Bit, LSB-First Transfer)

Serial Data Reception (Asynchronous Mode): Figures 14.7 and 14.8 show a sample flowchart for serial reception.

Use the following procedure for serial data reception after enabling the SCI for reception.



1. SCI initialization:

   See figure 14.4, Sample SCI Initialization Flowchart.

2. Receive error handling and break detection:

   Read the BRK, DR, and ER bits in SC2SSR to check whether a receive error has occurred.

   If a receive error occurs, read the ORER, PER3 to 0, and FER3 to 0 flags in SC1SSR, and the DR and BRK flags in SC2SSR to identify the error. After performing the appropriate error handling, ensure that the ORER, BRK, DR, and ER bits are all cleared to 0.

   Reception cannot be resumed if the ORER bit is set to 1. The setting of the EI bit in SC2SSR determines whether reception is continued or halted when either PER3 to 0 or FER3 to 0 is set to 1.

   In the case of a framing error, a break can be detected by reading the value of the RxD pin.

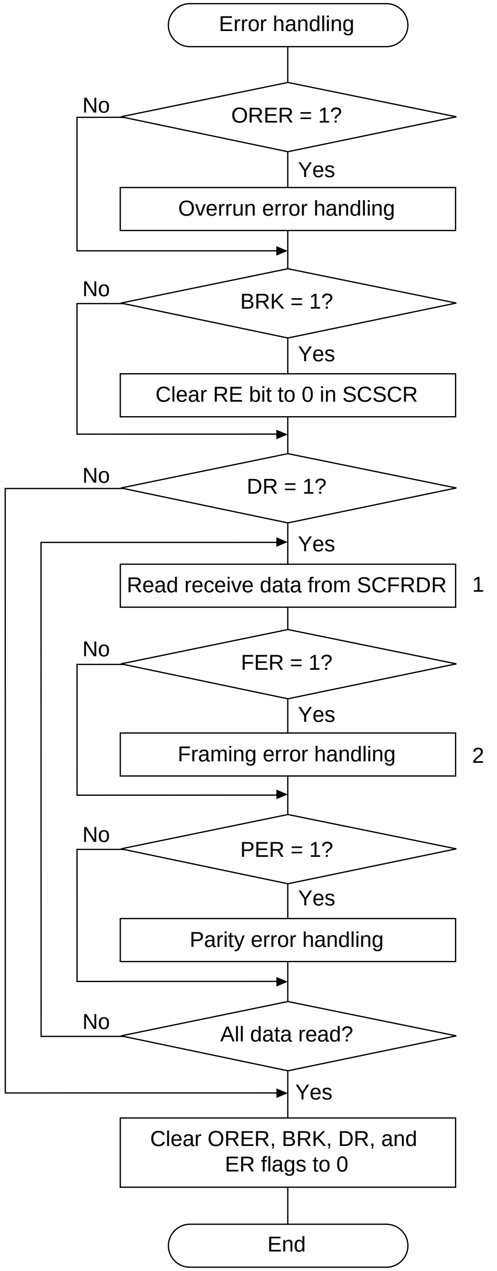
3. SCI status check and receive data read :

   Read the serial status 1 register (SC1SSR) and check that RDF = 1, then read receive data from the receive FIFO data register (SCFRDR) and clear the RDF bit to 0. Transition of the RDF bit from 0 to 1 can also be identified by means of an RXI interrupt.

4. Serial reception continuation procedure:

   To continue serial reception, read at least the receive trigger set number of data bytes from SCFRDR, and write 0 to the RDF flag after reading 1 from it. The number of receive data bytes in SCFRDR can be ascertained by reading the lower 8 bits of the FIFO data count register (SCFDR). (The RDF bit is cleared automatically when the DMAC is activated by an RXI interrupt and the SCFRDR value is read.)

Figure 14.7 Sample Serial Reception Flowchart (1)



1. Whether a framing error or parity error has occurred in the receive data read from SCFRDR can be ascertained from the FER and PER bits in SC1SSR.
2. When a break signal is received, receive data is not transferred to SCFRDR while the BRK flag is set. However, note that the last data in SCFRDR is H'00 and the break data in which a framing error occurred is stored.

Figure 14.8 Sample Serial Reception Flowchart (2)

In serial reception, the SCI operates as described below.

1. The SCI monitors the communication 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 or MSB-to-LSB order according to the setting of the RLM bit in SC2SSR.

3. The parity bit and stop bit are received.

   After receiving these bits, the SCI carries out the following checks.

   a. Parity check: The SCI checks whether the number of 1-bits in the receive data agrees with the parity (even or odd) set in the O/$\overline{\text{E}}$ bit in the serial mode register (SCSMR).
   b. Stop bit check: The SCI checks whether the stop bit is 1. If there are two stop bits, only the first is checked.
   c. Status check: The SCI checks whether receive data can be transferred from the receive shift register (SCRSR) to SCFRDR.
   d. Break check: The SCI checks that the BRK flag is 0, indicating no break.

   If all the above checks are passed, the receive data is stored in SCFRDR.

   If a receive error is detected in the error check, the operation is as shown in table 14.11.

Note: No further receive operations can be performed when an overrun error has occurred. The setting of the EI bit in SC2SSR determines whether reception is continued or halted when a framing error or parity error occurs.
Also, as the RDF flag is not set to 1 when receiving, the error flags must be cleared to 0.

4. If the RIE bit setting in SCSCR is 1 when the RDF flag changes to 1, a receive-FIFO-data-full interrupt (RXI) is requested.

   If the RIE bit setting in SCSCR is 1 when the ORER, PER, FER, or DR flag changes to 1, a receive-error interrupt (ERI) is requested.

Table 14.11 Receive Error Conditions

Receive Error	Abbreviation	Condition	Data Transfer
Overrun error	ORER	Next serial receive operation is completed while there are 16 receive data bytes in SCFRDR	Receive data is not transferred from SCRSR to SCFRDR
Framing error	FER	Stop bit is 0	Receive data is transferred from SCRSR to SCFRDR
Parity error	PER	Received data parity differs from that (even or odd) set in SCSMR	Receive data is transferred from SCRSR to SCFRDR

Figure 14.9 shows an example of the operation for reception in asynchronous mode.

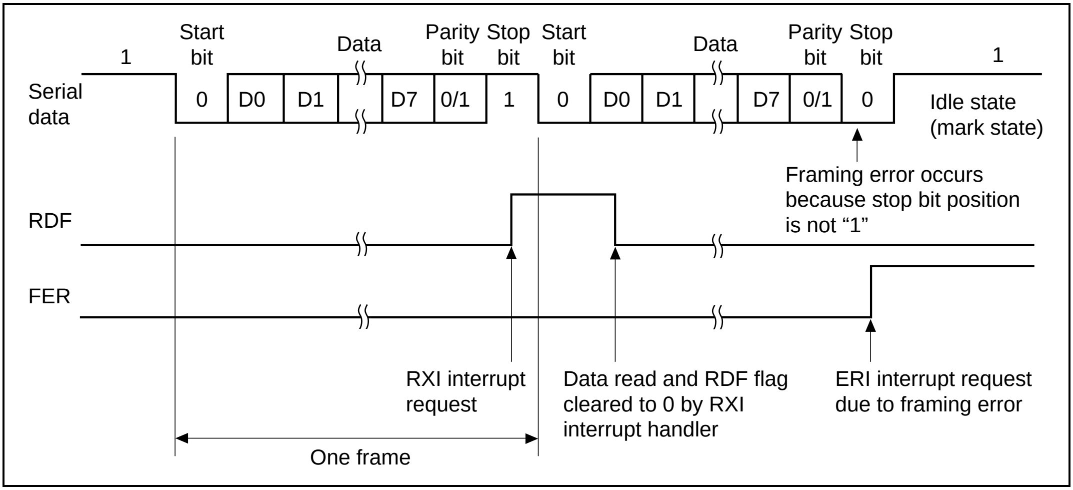

1
Start bit
Data
Parity bit
Stop bit
Start bit
Data
Parity bit
Stop bit
1
Serial data
0
D0
D1
D7
0/1
1
0
D0
D1
D7
0/1
0
Idle state (mark state)
Framing error occurs because stop bit position is not "1"
RDF
FER
RXI interrupt request
Data read and RDF flag cleared to 0 by RXI interrupt handler
ERI interrupt request due to framing error
One frame

Figure 14.9 Example of SCI Receive Operation (Example with 8-Bit Data, Parity, One Stop Bit, LSB-First Transfer)

14.3.3 Multiprocessor Communication Function

The multiprocessor communication function performs serial communication using a multiprocessor format, in which a multiprocessor bit is added to the transfer data, in asynchronous mode. Use of this function enables data transfer to be performed among a number of processors sharing a serial communication line.

When multiprocessor communication is carried out, each receiving station is addressed by a unique ID code.

The serial communication cycle consists of two cycles: an ID transmission cycle which specifies the receiving station, and a data transmission cycle. The multiprocessor bit is used to differentiate between the ID transmission cycle and the data transmission cycle.

The transmitting station first sends the ID of the receiving station with which it wants to perform serial communication as data with a 1 multiprocessor bit added. It then sends transmit data as data with a 0 multiprocessor bit added.

The receiving stations skip the data until data with a 1 multiprocessor bit is sent. When data with a 1 multiprocessor bit is received, each receiving stations compares that data with its own ID. The station whose ID matches then receives the data sent next. Stations whose ID does not match continue to skip the data until data with a 1 multiprocessor bit is again received. In this way, data communication is carried out among a number of processors.

Figure 14.10 shows an example of inter-processor communication using a multiprocessor format.

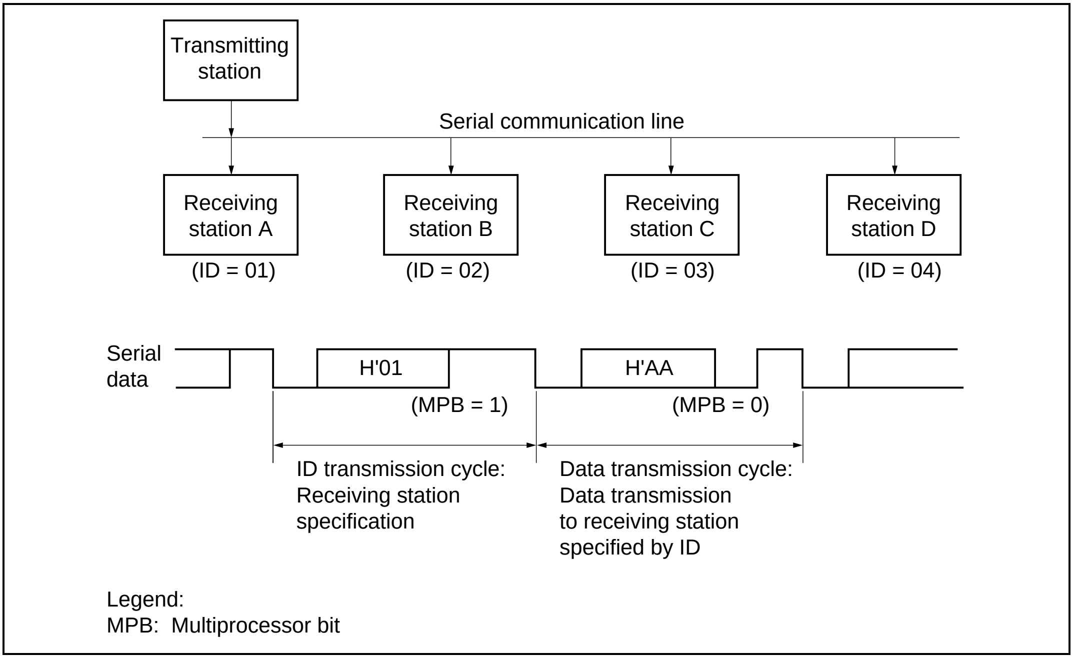


Figure 14.10 Example of Inter-Processor Communication Using Multiprocessor Format (Transmission of Data H'AA to Receiving Station A)

Transmit/Receive Formats

There are four transmit/receive formats. When the multiprocessor format is specified, the parity bit specification is invalid. For details, see table 14.10.

Clock

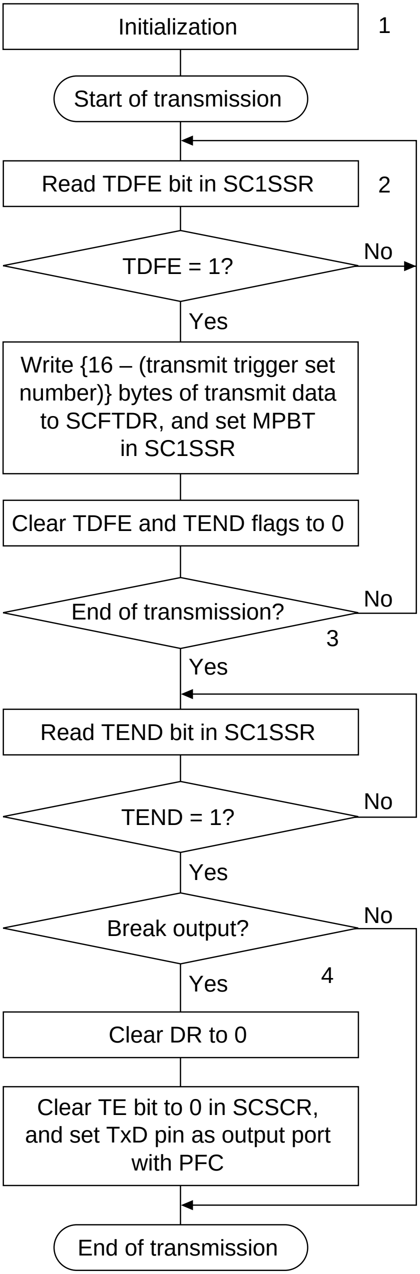
See the section on asynchronous mode.

Data Transmit/Receive Operations

SCI Initialization: See the section on asynchronous mode.

Multiprocessor Serial Data Transmission: Figure 14.11 shows a sample flowchart for multiprocessor serial data transmission.

Use the following procedure for multiprocessor serial data transmission after enabling the SCI for transmission.


Initialization
1
Start of transmission
Read TDFE bit in SC1SSR
2
TDFE = 1?
No
Yes
Write {16 – (transmit trigger set number)} bytes of transmit data to SCFTDR, and set MPBT in SC1SSR
Clear TDFE and TEND flags to 0
End of transmission?
No
3
Yes
Read TEND bit in SC1SSR
TEND = 1?
No
Yes
Break output?
No
Yes
4
Clear DR to 0
Clear TE bit to 0 in SCSCR, and set TxD pin as output port with PFC
End of transmission

1. SCI initialization:

   See figure 14.4, Sample SCI Initialization Flowchart.

2. SCI status check and transmit data write:

   First, set the MPBT bit in SC1SSR to 0 or 1.

   Read the serial status 1 register (SC1SSR) and check that the TDFE bit is set to 1, then write transmit data to the transmit FIFO data register (SCFTDR). Finally, clear the TDFE and TEND flags to 0 after reading 1 from them.

   The number of data bytes that can be written is {16 – (transmit trigger set number)}.

3. Serial transmission continuation procedure:

   To continue serial transmission, read 1 from the TDFE bit to confirm that writing is possible, then write data to SCFTDR, and then clear the TDFE bit to 0. (Checking and clearing of the TDFE bit is automatic when the DMAC is activated by a transmit-FIFO-data-empty interrupt (TXI) request, and data is written to SCFTDR.)

4. Break output at the end of serial transmission:

   To output a break in serial transmission, clear the port data register (DR) to 0, then clear the TE bit to 0 in SCSCR, and set the TxD pin as an output port with the PFC.

In steps 2 and 3, the number of transmit data bytes that can be written can be ascertained from the number of transmit data bytes in SCFTDR indicated in the upper 8 bits of the FIFO data count register (SCFDR).

Figure 14.11 Sample Multiprocessor Serial Transmission Flowchart

In serial transmission, the SCI operates as described below.

1. When data is written to SCFTDR, the SCI transfers the data to SCTSR and starts transmitting. Check that the TDFE flag is set to 1 in SC1SSR before writing transmit data to SCFTDR. The number of data bytes that can be written is at least {16 – (transmit trigger set number)}.

2. When data is transferred from SCFTDR to SCTSR and transmission is started, transmit operations are performed continually until there is no transmit data left in SCFTDR. If the number of data bytes in SCFTDR falls to or below the transmit trigger number set in SCFCR during transmission, the TDFE flag is set to 1. If the TIE bit setting in SCSCR is 1 at this time, a transmit-FIFO-data-empty interrupt (TXI) is requested.

   The serial transmit data is sent from the 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 or MSB-first order according to the setting of the TLM bit in SC2SSR.

   c. Multiprocessor bit: One multiprocessor bit (MPBT value) is output.

   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 SCI checks for transmit data in SCFTDR at the timing for sending the stop bit. If there is data in SCFTDR, it is transferred to SCTSR, the stop bit is sent, and then serial transmission of the next frame is started.

   If there is no transmit data in SCFTDR, the TEND flag is set to 1 in SC1SSR, the stop bit is sent, and then the line goes to the mark state in which 1 is output continuously. If the TEIE bit setting in SCSCR is 1 at this time, a transmit-end interrupt (TEI) is requested.

Figure 14.12 shows an example of SCI operation for transmission using a multiprocessor format.

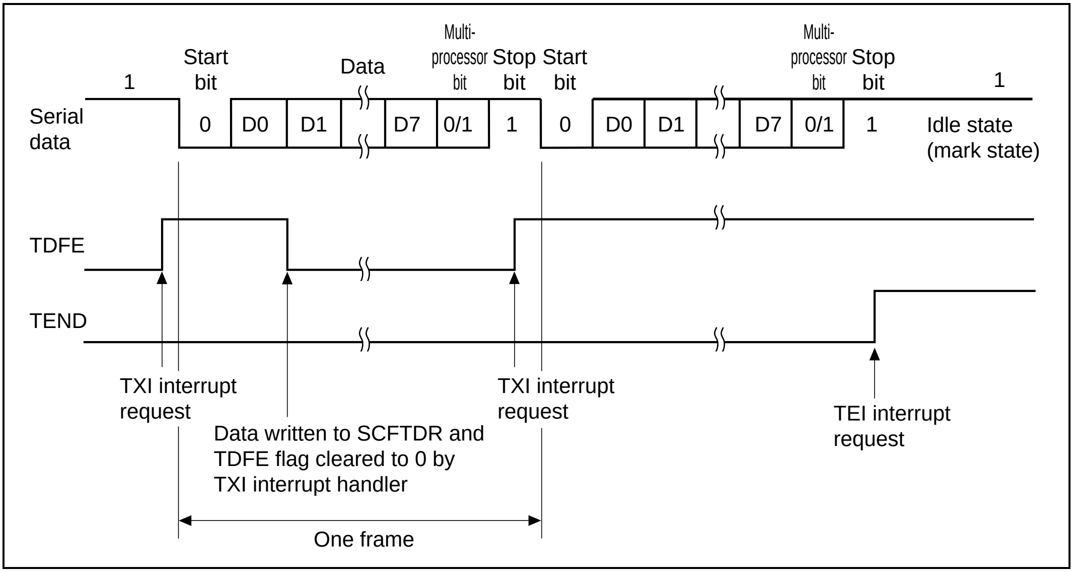
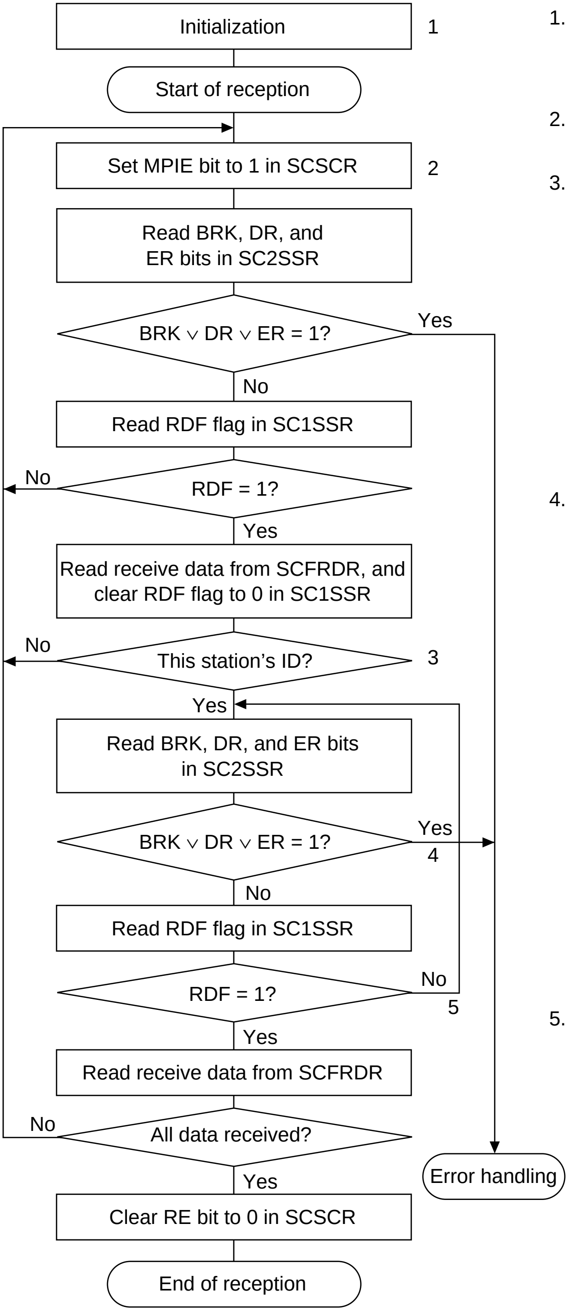


Figure 14.12 Example of SCI Transmit Operation (Example with 8-Bit Data, Multiprocessor Bit, One Stop Bit, LSB-First Transfer)

Multiprocessor serial data reception: Figures 14.13 and 14.14 show a sample flowchart for multiprocessor serial reception.

Use the following procedure for multiprocessor serial data reception after enabling the SCI for reception.



1. SCI initialization:

   See figure 14.4, Sample SCI Initialization Flowchart.

2. ID reception cycle: Set the MPIE bit to 1 in SCSCR.

3. SCI status check, ID reception and comparison:

   Read SC1SSR and check that the RDF bit is set to 1, then read the receive data in the receive FIFO data register (SCFRDR) and compare it with this station's ID.

   If the data is not this station's ID, set the MPIE bit to 1 again, and clear the RDF bit to 0. If the data is this station's ID, clear the RDF bit to 0.

4. Receive error handling and break detection:

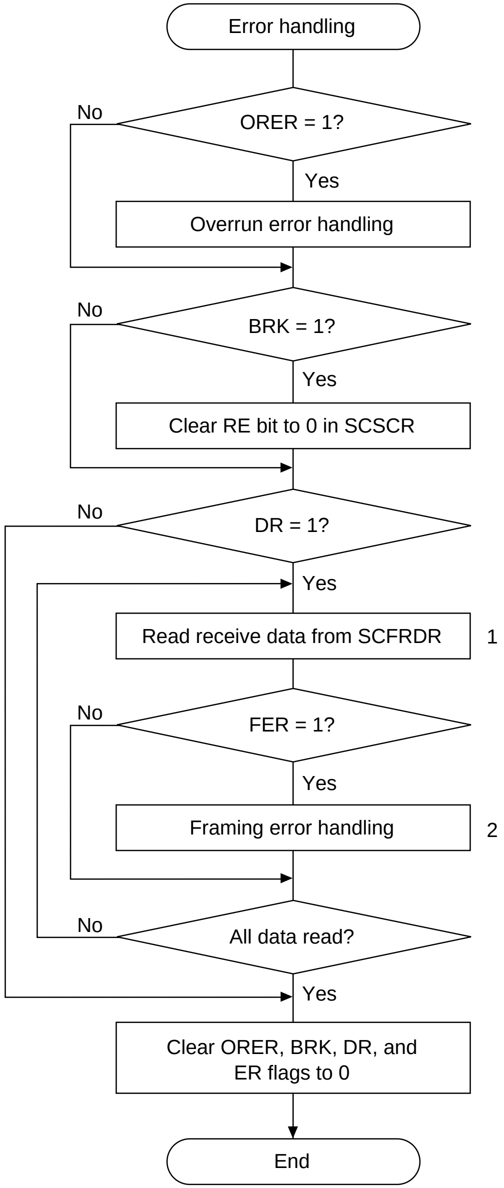
   Read the BRK, DR, and ER bits in SC2SSR to check whether a receive error has occurred.

   If a receive error occurs, read the ORER and FER3 to 0 flags in SC1SSR, and the BRK, DR, and ER flags in SC2SSR to identify the error. After performing the appropriate error handling, ensure that the ORER, BRK, DR, and ER bits are all cleared to 0. The setting of the EI bit in SC2SSR determines whether reception is continued or halted when the ORER bit is set to 1. In the case of a framing error, a break can be detected by reading the value of the RxD pin.

5. SCI status check and receive data read:

   Read the serial status 1 register (SC1SSR) and check that RDF = 1, then read receive data from the receive FIFO data register (SCFRDR).

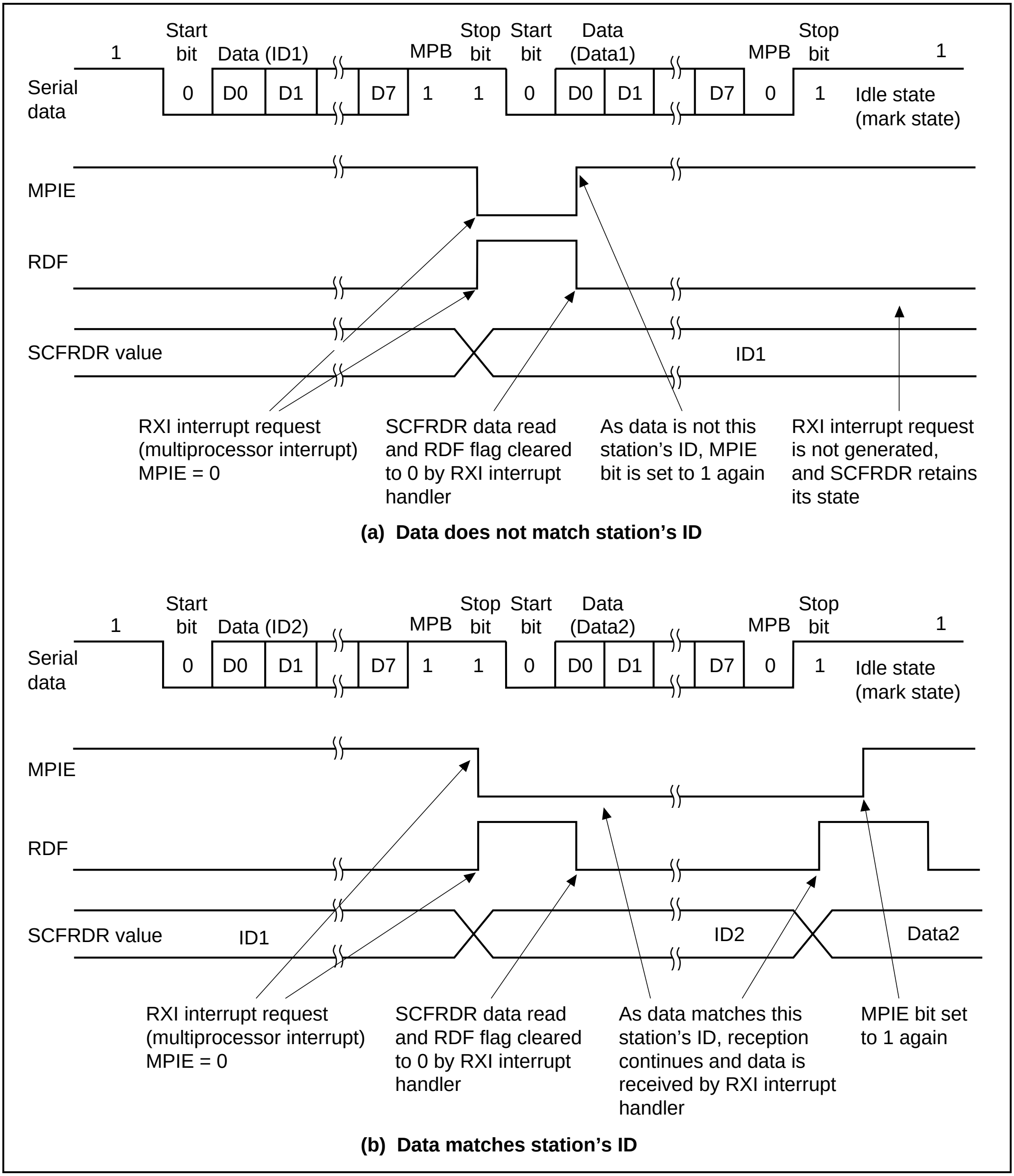
Figure 14.13 Sample Multiprocessor Serial Reception Flowchart (1)



1. Whether a framing error has occurred in the receive data read from SCFRDR can be ascertained from the FER bit in SC1SSR.
2. When a break signal is received, receive data is not transferred to SCFRDR while the BRK flag is set. However, note that the last data in SCFRDR is H'00 and the break data in which a framing error occurred is stored.

Figure 14.14 Sample Multiprocessor Serial Reception Flowchart (2)

Figure 14.15 shows an example of SCI operation for multiprocessor format reception.



1
Start bit
Data (ID1)
MPB
Stop bit
Start bit
Data (Data1)
MPB
Stop bit
1
Serial data
0
D0
D1
D7
1
1
0
D0
D1
D7
0
1
Idle state (mark state)
MPIE
RDF
SCFRDR value
ID1
RXI interrupt request (multiprocessor interrupt) MPIE = 0
SCFRDR data read and RDF flag cleared to 0 by RXI interrupt handler
As data is not this station's ID, MPIE bit is set to 1 again
RXI interrupt request is not generated, and SCFRDR retains its state
(a) Data does not match station's ID
1
Start bit
Data (ID2)
MPB
Stop bit
Start bit
Data (Data2)
MPB
Stop bit
1
Serial data
0
D0
D1
D7
1
1
0
D0
D1
D7
0
1
Idle state (mark state)
MPIE
RDF
SCFRDR value
ID1
ID2
Data2
RXI interrupt request (multiprocessor interrupt) MPIE = 0
SCFRDR data read and RDF flag cleared to 0 by RXI interrupt handler
As data matches this station's ID, reception continues and data is received by RXI interrupt handler
MPIE bit set to 1 again
(b) Data matches station's ID

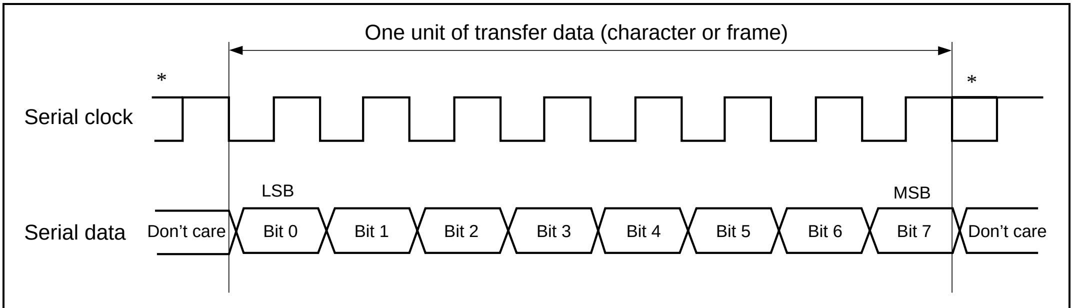
Figure 14.15 Example of SCI Receive Operation
(Example with 8-Bit Data, Multiprocessor Bit, One Stop Bit, LSB-First Transfer)

14.3.4 Operation in Synchronous Mode

In synchronous mode, data is transmitted or received in synchronization with clock pulses, making it suitable for high-speed serial communication.

Inside the SCI, the transmitter and receiver are independent units, enabling full-duplex communication using a common clock. Both the transmitter and the receiver also have a 16-stage FIFO buffer structure, so that data can be read or written during transmission or reception, enabling continuous data transfer.

Figure 14.16 shows the general format for synchronous serial communication.



Note: * High except in continuous transmission/reception

Figure 14.16 Data Format in Synchronous Communication

In synchronous serial communication, data on the communication line is output from one fall of the serial clock to the next. Data is guaranteed valid at the rise of the serial clock.

In serial communication, each character is output starting with the LSB and ending with the MSB, or vice versa, according to the setting of the TLM bit in the serial status 2 register (SC2SSR). After the last data is output, the communication line remains in the state of the last data.

In synchronous mode, the SCI receives data in synchronization with the rise of the serial clock.

Transmit/Receive Format

A fixed 8-bit data format is used. No parity or multiprocessor bits are added.

Clock

Either an internal clock generated by the built-in baud rate generator or an external serial clock input at the SCK pin can be selected, according to the setting of the $C/\overline{A}$ bit in SCSMR and the CKE1 and CKE0 bits in SCSCR. For details of SCI clock source selection, see table 14.9.

When the SCI is operated on an internal clock, the serial clock is output from the SCK pin.

Eight serial clock pulses are output in the transfer of one character, and when no transmission/reception is performed the clock is fixed high. Note that, in receive-only operation, the serial clock continues to be output and reception continues until the FIFO buffer is full and an overrun error occurs. In this case, 8 x (16 + 1) = 136 serial clock pulses are output. To perform reception of n characters, select an external clock as the clock source. If an internal clock is used, set RE = 1 and TE = 1, and perform reception of n characters simultaneously with transmission of n characters of dummy data.

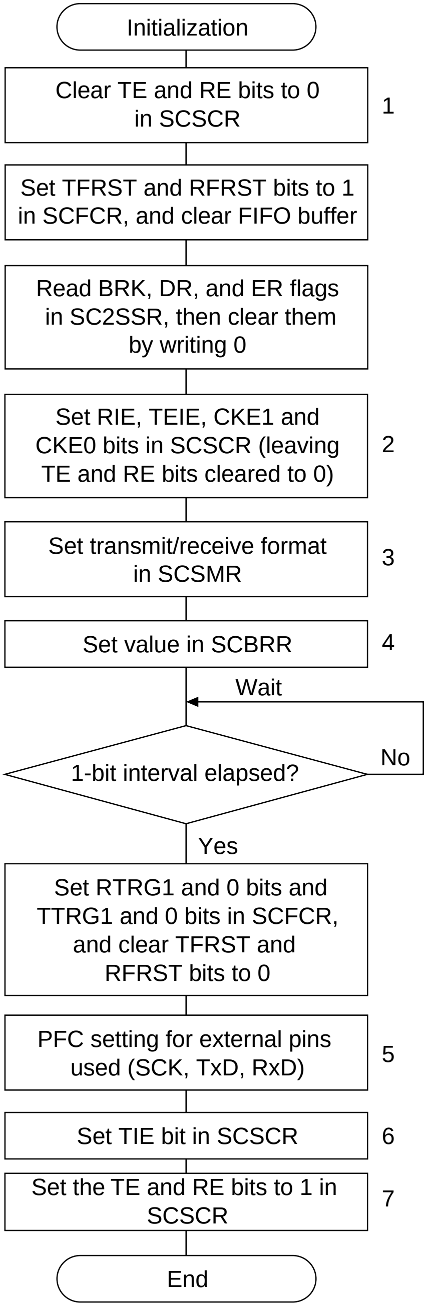
Transmit/Receive Operations

SCI Initialization (Synchronous Mode): Before transmitting and receiving data, it is necessary to clear the TE and RE bits to 0 in the serial control register (SCSCR), then initialize the SCI as described below.

When the operating mode, communication format, etc., is changed, clear the TE and RE bits to 0 and perform initialization. When the TE bit is cleared to 0, the transmit shift register (SCTSR) is initialized.

Note that clearing the RE bit to 0 does not change the contents of the RDF, PER, FER, and ORER flags, or the receive data register (SCRDR).

Figure 14.17 shows a sample SCI initialization flowchart.


Initialization
Clear TE and RE bits to 0 in SCSCR
1
Set TFRST and RFRST bits to 1 in SCFCR, and clear FIFO buffer
Read BRK, DR, and ER flags in SC2SSR, then clear them by writing 0
Set RIE, TEIE, CKE1 and CKE0 bits in SCSCR (leaving TE and RE bits cleared to 0)
2
Set transmit/receive format in SCSMR
3
Set value in SCBRR
4
Wait
1-bit interval elapsed?
No
Yes
Set RTRG1 and 0 bits and TTRG1 and 0 bits in SCFCR, and clear TFRST and RFRST bits to 0
PFC setting for external pins used (SCK, TxD, RxD)
5
Set TIE bit in SCSCR
6
Set the TE and RE bits to 1 in SCSCR
7
End

1. Clear bits TE and RE to 0 before end of initialization.
2. Set disabling/enabling of RXI and TEI interrupt requests. When enabling an interrupt request, also make a setting in the IPRK register of the INTC.
3. Set the transmit/receive format in SCSMR.

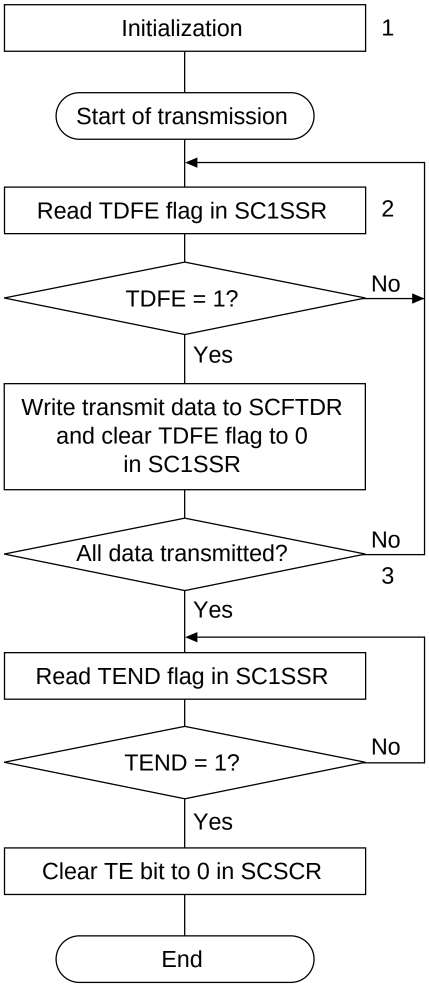
   When using IrDA mode, also set SCIMR.
4. Write a value corresponding to the bit rate to the bit rate register (SCBRR). Not necessary if an external clock is used. Wait for at least one bit interval after making this setting.
5. Make PFC settings for the external pins to be used.

   Make a setting for RxD input when receiving, and for TxD output when transmitting. Make an SCK input/output setting according to the setting of bits CKE1 and CKE0.
6. When the TXI interrupt is used, first clear TFRST and RFRST in SCFCR to 0. Even if TE = 0, a TXI interrupt will be generated as soon as the TIE bit in SCSCR is set to 1.
7. Set the TE bit or RE bit to 1 in SCSCR. The TxD or RxD pin and the SCK pin become available for use at this point. When transmitting, the TxD pin goes to the mark state. When receiving in synchronous mode with serial clock output (clock master) set, clock output from the SCK pin begins at this point.

Figure 14.17 Sample SCI Initialization Flowchart

Serial Data Transmission (Synchronous Mode): Figure 14.18 shows a sample flowchart for serial transmission.

Use the following procedure for serial data transmission after enabling the SCI for transmission.



1. SCI initialization:

   See figure 14.17, Sample SCI Initialization Flowchart.

2. SCI status check and transmit data write:

   Read SC1SSR and check that the TDFE =1, then write transmit data to the transmit FIFO data register (SCFTDR) and clear the TDFE flag to 0.

3. 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.

Figure 14.18 Sample Serial Transmission Flowchart

In serial transmission, the SCI operates as described below.

1. When data is written to the transmit FIFO data register (SCFTDR), the SCI transfers the data from SCFTDR to the transmit shift register (SCTSR), and starts transmitting. Check that the TDFE flag is set to 1 in the serial status 1 register (SC1SSR) before writing transmit data to SCFTDR. The number of data bytes that can be written is at least {16 – (transmit trigger set number)}.

2. When data is transferred from SCFTDR to SCTSR and transmission is started, transmit operations are performed continually until there is no transmit data left in SCFTDR. If the number of data bytes in SCFTDR falls to or below the transmit trigger number set in the FIFO control register (SCFCR) during transmission, the TDFE flag is set. If the TIE bit setting in the serial control register (SCSCR) is 1 at this time, a transmit-FIFO-data-empty interrupt (TXI) is requested.

   When clock output mode has been set, the SCI outputs 8 serial clock pulses for one unit of data.

   When use of an external clock has been specified, data is output in synchronization with the input clock.

   The serial transmit data is sent from the TxD pin starting with the LSB (bit 0) or MSB (bit 7) according to the setting of the TLM bit in the serial status 2 register (SC2SSR).

3. The SCI checks for transmit data in SCFTDR at the timing for sending the last bit. If there is data in SCFTDR, it is transferred to SCTSR and then serial transmission of the next frame is started. If there is no transmit data in SCFTDR, the TEND flag is set to 1 in the serial status 1 register (SC1SSR), the last bit is sent, and then the transmit data pin (TxD) holds its state.

   If the transmit-end interrupt enable bit (TEIE) setting in SCSCR is 1 at this time, a transmit-end interrupt (TEI) is requested.

4. After completion of serial transmission, the SCK pin is fixed high.

Figure 14.19 shows an example of SCI operation in transmission.

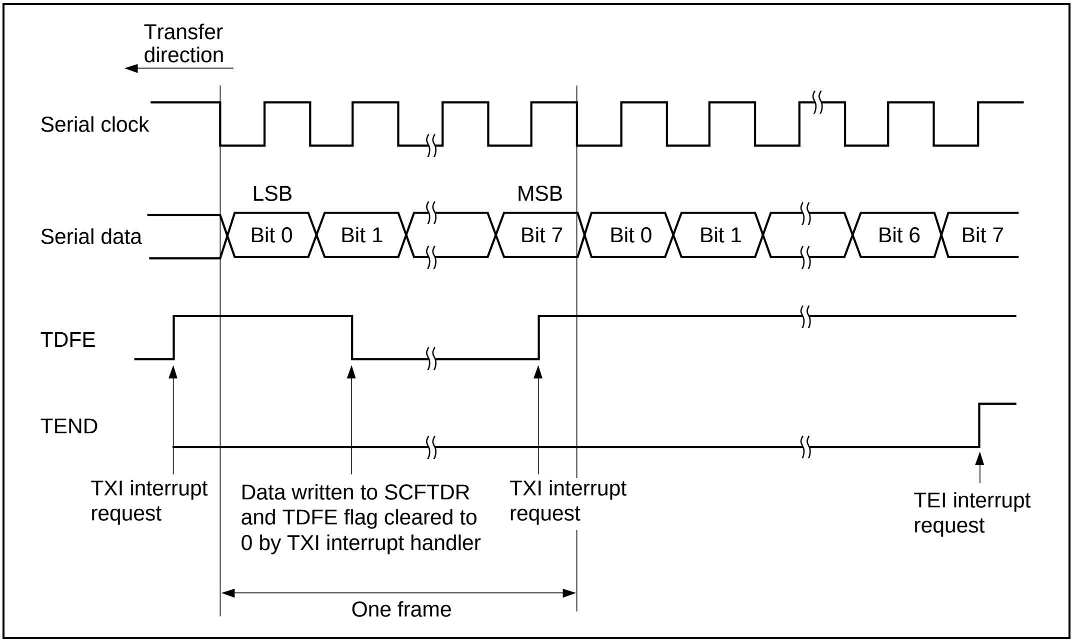
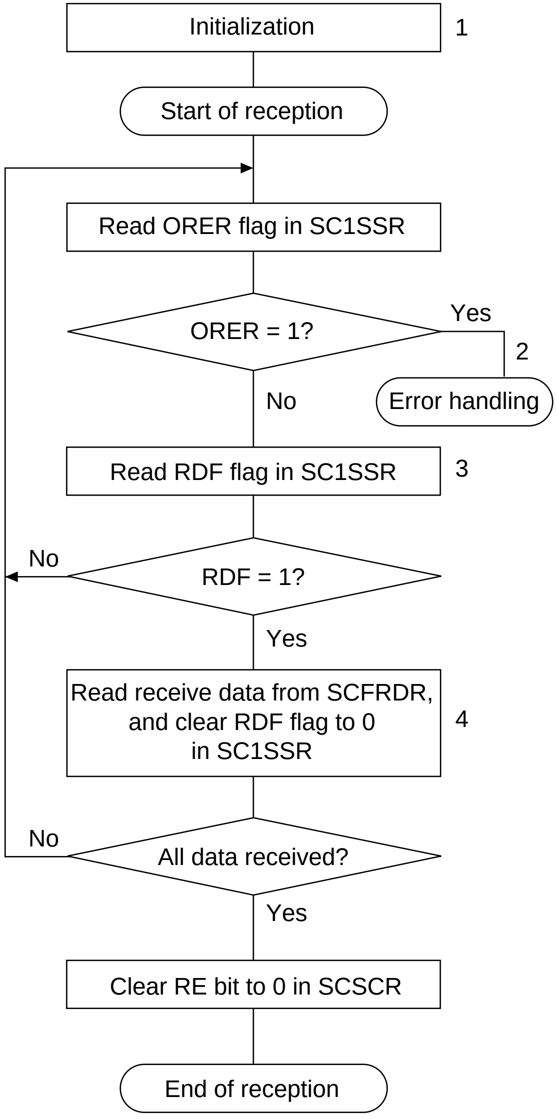


Figure 14.19 Example of SCI Transmit Operation

Serial Data Reception (Synchronous Mode): Figures 14.20 and 14.21 show a sample flowchart for serial reception.

Use the following procedure for serial data reception after enabling the SCI for reception.

When changing the operating mode from asynchronous to synchronous without resetting SCFRDR and SCFTDR by means of SCI initialization, be sure to check that the ORER, PER3 to PER0, and FER3 to FER0 flags are all cleared to 0. The RDF flag will not be set if any of flags FER3 to FER0 or PER3 to PER0 are set to 1, and neither transmit nor receive operations will be possible.



1. SCI initialization:

   See figure 14.17, Sample SCI Initialization Flowchart.

2. Receive error handling:

   If a receive error occurs, read the ORER flag in SC1SSR , and after performing the appropriate error handling, clear the ORER flag to 0. Transmission/reception cannot be resumed if the ORER flag is set to 1.

3. SCI status check and receive data read:

   Read the serial status 1 register (SC1SSR) and check that the RDF flag is set to 1, then read receive data from the receive FIFO data register (SCFRDR) and clear the RDF flag to 0. Transition of the RDF flag from 0 to 1 can also be identified by an RXI interrupt.

4. Serial reception continuation procedure:

   To continue serial reception, read at least the receive trigger set number of data bytes from SCFRDR, and write 0 to the RDF flag after reading 1 from it. The number of receive data bytes in SCFRDR can be ascertained by reading the lower 8 bits of the FIFO data count register (SCFDR). (The RDF bit is cleared automatically when the DMAC is activated by an RXI interrupt and the SCFRDR value is read.)

Figure 14.20 Sample Serial Reception Flowchart (1)

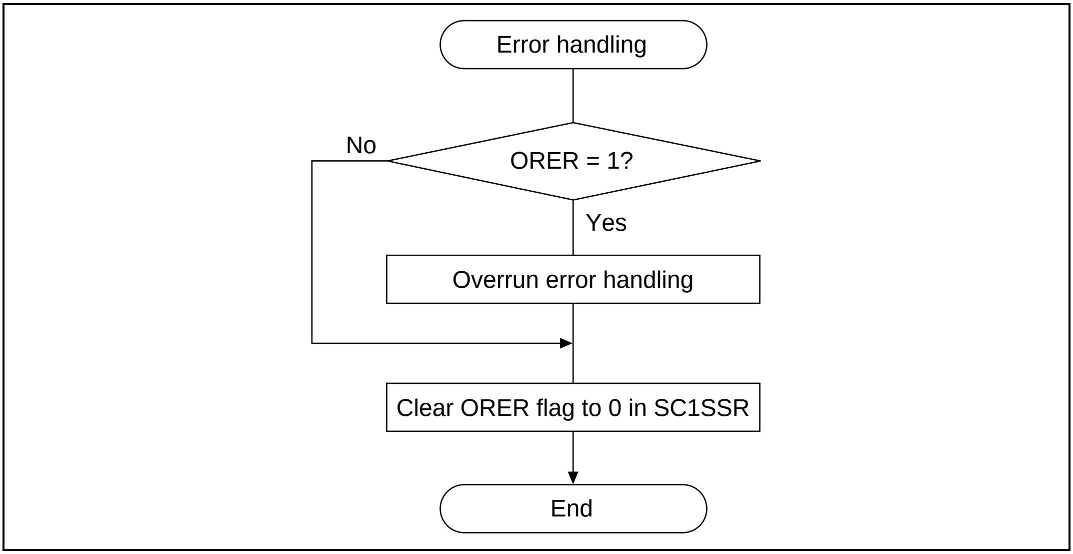


Figure 14.21 Sample Serial Reception Flowchart (2)

In serial reception, the SCI operates as described below.

1. The SCI performs internal initialization in synchronization with serial clock input or output.

2. The received data is stored in the receive shift register (SCRSR) in LSB-to-MSB order or MSB-to-LSB order according to the setting of the RLM bit in SC2SSR.

   After reception, the SCI checks whether the receive data can be transferred from SCRSR to the receive FIFO data register (SCFRDR). If this check is passed, the receive data is stored in SCFRDR.

   If a receive error is detected in the error check, the operation is as shown in table 14.11. Neither transmit nor receive operations can be performed subsequently when a receive error has been found in the error check.

   Also, as the RDF flag is not set to 1 when receiving, the ORER flag must be cleared to 0.

3. If the RIE bit setting in the serial control register (SCSCR) is 1 when the RDF flag changes to 1, a receive-FIFO-data-full interrupt (RXI) is requested. If the RIE bit setting in SCRSR is 1 when the ORER flag changes to 1, a receive-error interrupt (ERI) is requested.

Figure 14.22 shows an example of SCI operation in reception.

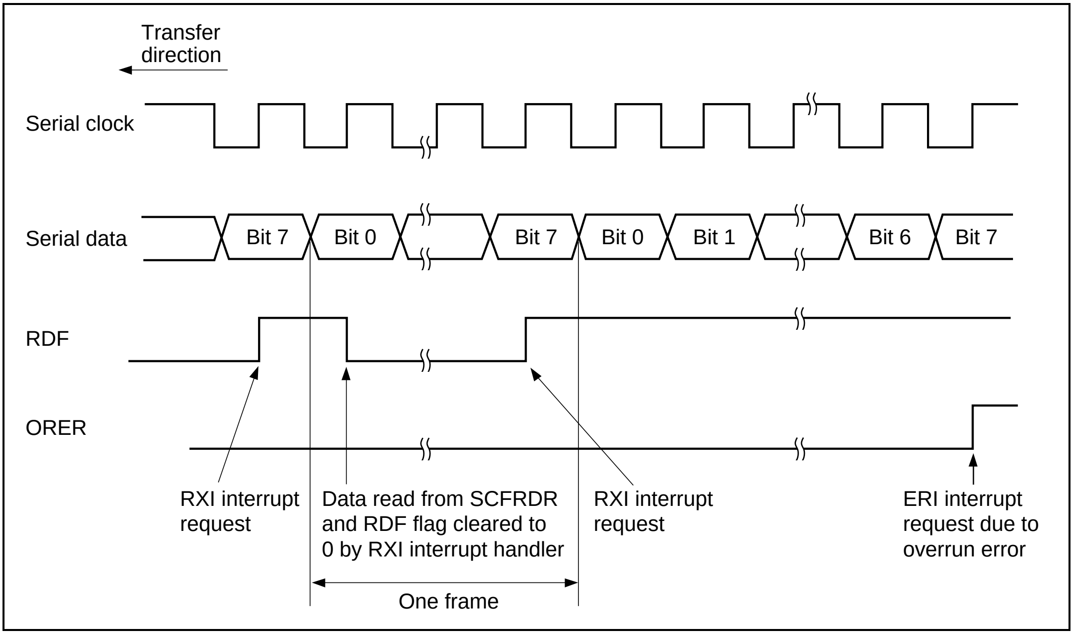
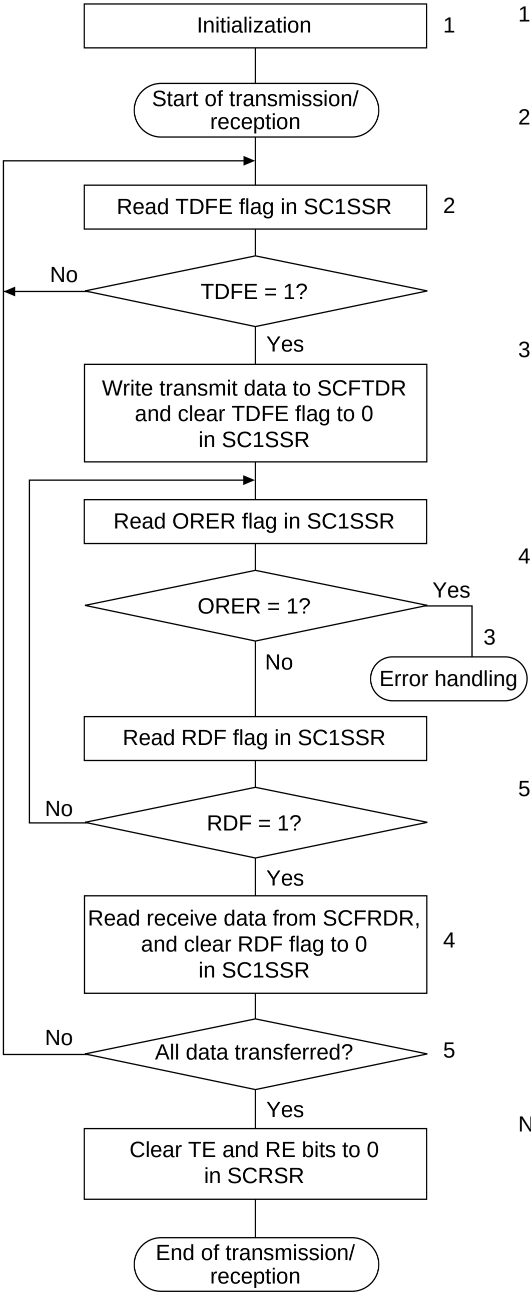


Figure 14.22 Example of SCI Receive Operation

Simultaneous Serial Data Transmission and Reception (Synchronous Mode): Figure 14.23 shows a sample flowchart for simultaneous serial transmit and receive operations.

Use the following procedure for simultaneous serial data transmit and receive operations after enabling the SCI for transmission and reception.


Initialization
1
Start of transmission/ reception
Read TDFE flag in SC1SSR
2
No
TDFE = 1?
Yes
Write transmit data to SCFTDR and clear TDFE flag to 0 in SC1SSR
Read ORER flag in SC1SSR
ORER = 1?
Yes
3
No
Error handling
Read RDF flag in SC1SSR
No
RDF = 1?
Yes
Read receive data from SCFRDR, and clear RDF flag to 0 in SC1SSR
4
No
All data transferred?
5
Yes
Clear TE and RE bits to 0 in SCRSR
End of transmission/ reception

1. SCI initialization:

   See figure 14.17, Sample SCI Initialization Flowchart.

2. SCI status check and transmit data write:

   Read SC1SSR and check that the TDFE flag is set to 1, then write transmit data to SCFTDR and clear the TDFE flag to 0. Transition of the TDFE flag from 0 to 1 can also be identified by a TXI interrupt.

3. Receive error handling:

   If a receive error occurs, read the ORER flag in SC1SSR , and after performing the appropriate error handling, clear the ORER flag to 0. Transmission/reception cannot be resumed if the ORER flag is set to 1.

4. SCI status check and receive data read:

   Read SC1SSR and check that the RDF flag is set to 1, then read receive data from SCFRDR and clear the RDF flag to 0. Transition of the RDF flag from 0 to 1 can also be identified by an RXI interrupt.

5. Serial transmission/reception continuation procedure:

   To continue serial transmission/ reception, finish reading the RDF flag, reading SCFRDR, and clearing the RDF flag to 0, before the last bit of the current frame is received. Also, before the last bit of the current frame is transmitted, read 1 from the TDFE flag to confirm that writing is possible, then write data to SCFTDR and clear the TDFE flag to 0.

Note: When switching from transmitting or receiving to simultaneous transmitting and receiving, first clear the TE bit and RE bit to 0, then set the TE bit and RE bit to 1 simultaneously.

Figure 14.23 Sample Flowchart for Serial Transmission and Reception

14.3.5 Use of Transmit/Receive FIFO Buffers

The SCI has independent 16-stage FIFO buffers for transmission and reception. The configuration of these buffers is shown in figure 14.24.

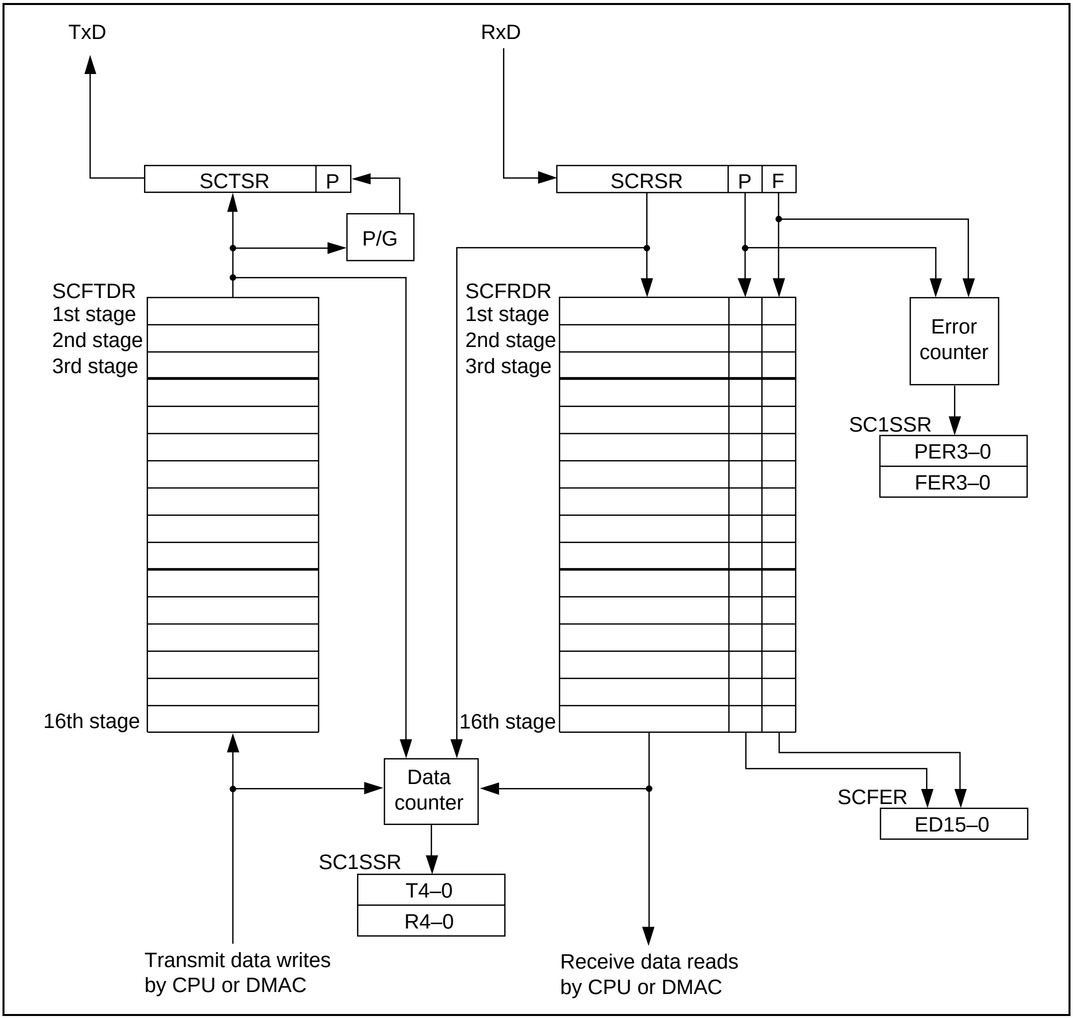


Figure 14.24 Transmit/Receive FIFO Configuration

In Serial Data Transmit Operations

In transmission, when transmit data is written to the transmit FIFO by the CPU or DMAC and the TE bit is set to 1 in the serial control register (SCSCR), the data is first transferred to the transmit shift register (SCTSR) in the order of writing to the transmit FIFO, a parity bit is added by the parity generator (P/G), and then serial data is transmitted from the TxD pin.

Each time data is written into the transmit FIFO, the value in bits T4 to T0 in the FIFO data count register (SCFDR) is incremented, and each time data is transferred to SCTSR the value in bits T4 to T0 is decremented. The current number of data bytes in the transmit FIFO can thus be found by reading bits T4 to T0 in SCFDR.

A value of H'10 in bits T4 to T0 means that data has been written into all 16 stages of the transmit FIFO. If additional data is written to the FIFO in this state, bits T4 to T0 will not be incremented and the written data will be lost.

When the transmit trigger number is set and transmit data is written to the FIFO by the DMAC, if bit 17 (Flag Clear Timing Select (FCS)) in the DMAC's DMA channel control register (CHCRn) is 0 and bit 6 (DREQ Select (DS)) is 1, even though TDFE in serial status register 1 (SCISSR) is cleared to 0 by execution of the DMAC transfer, the DMAC will continue to transfer data to the FIFO until the value in the DMA transfer count register reaches 0. In this case, therefore, care must be taken not to write data exceeding the number of empty bytes in SCFTDR indicated by the FIFO control register (SCFCR) (see section 14.2.10, FIFO Control Register (SCFCR)).

In Serial Data Receive Operations

In reception, serial data input from the RxD pin is first captured in the receive shift register (SCRSR) in the order specified by the RLM bit in the serial status 2 register (SC2SSR). A parity bit check is carried out, and if there is a parity error the P (parity error) flag for that data is set to 1. A stop bit check is also performed, and if a framing error is found the F (framing error) flag for that data is set to 1. The receive FIFO buffer has a 10-bit configuration, with the P and F flags for each 8-bit data unit stored together with that data.

Receive FIFO Control in Normal Operation: Receive data held in the receive FIFO buffer is read by the CPU or DMAC.

Each time data is transferred from SCRSR to the receive FIFO, the value in bits R4 to R0 in SCFDR is incremented, and each time the CPU or DMAC reads receive data from the receive FIFO, the value in bits R4 to R0 is decremented. The current number of data bytes in the receive FIFO can thus be found by reading bits R4 to R0 in SCFDR.

A value of H'10 in bits R4 to R0 means that receive data has been transferred to all 16 stages of the receive FIFO. If the next serial receive operation is completed before the CPU or DMAC reads data from the FIFO, an overrun error will result and the serial data will be lost. If receive FIFO data is read when the value of bits R4 to R0 is H'00, an undefined value will be returned.

Receive FIFO Control in Error Data Reception: When data is transferred from SCRSR to the receive FIFO, the P and F flags are also transferred. If either of these flags is set to 1, the error counter is incremented and the corresponding bit (PER3 to PER0, FER3 to FER0) is updated in the serial status 1 register (SC1SSR). The error counter is also incremented if the P or F flag is 1 when data in the receive FIFO is read by the CPU or DMAC. The settings of the P and F flags for the read receive data are also reflected in the PER and FER flags in SC1SSR. PER and FER are set when data containing a parity error or framing error is read from the receive FIFO; they are not set when serial data containing a parity error or framing error is received from the RxD pin. PER and FER are cleared when data with no parity error or framing error is read from the receive FIFO.

This data is transferred to the receive FIFO even if it contains a parity error or framing error. Whether or not the receive operation is to be continued at this point can be specified with the EI bit in SC2SSR. If the EI bit is set to 1, specifying continuation of the receive operation, receive data is still transferred sequentially to the receive FIFO after an error occurs. The stage of the 16-stage FIFO buffer in which the data with the error is located can be determined by reading bits ED15 to ED0 in the FIFO error register (SCFER).

When the receive trigger number is set and receive data is read from the receive FIFO by the DMAC, care must be taken not to read data exceeding the receive trigger number indicated by the FIFO control register (SCFCR) (see section 14.2.10, FIFO Control Register (SCFCR)).

FIFO Control by DR Flag: When a number of data bytes equal to or exceeding the receive trigger number have been received, a receive data read request is issued to the CPU or DMAC by means of an RXI interrupt. However, an RXI interrupt is not requested if all reception has been completed with fewer than the receive trigger number of data bytes having been received. In this case, the DR flag is set and an ERI interrupt is requested 15 etu after reception of the last data is completed. The CPU should therefore read bits R4 to R0 in SCFDR to find the number of data bytes left in the receive FIFO, and read all the data in the FIFO.

Note: etu: Elementary time unit = s/bit
A 15 etu is equivalent to 1.5 frames with an 8-bit, 1-stop-bit format.

14.3.6 Operation in IrDA Mode

In IrDA mode, the waveform of TxD/RxD transmit/receive data is modified to comply with the IrDA 1.0 infrared communication specification. This makes it possible to carry out infrared transmission and reception conforming to the IrDA standard by connecting an infrared transmission/reception transceiver/receiver.

In the IrDA 1.0 specification, communication is initially executed at 9600 bps, and then the transfer rate can be changed as required. However, the communication speed is not changed automatically in this module. When executing communication, therefore, it is necessary to check the communication speed and have the appropriate speed set in this module by software.

Note: In IrDA mode, reception is not possible when the TE bit is set to 1 (enabling communication) in the serial control register (SCSCR). When performing reception, the TE bit in SCSCR must be cleared to 0.

Transmission

In the case of a serial output signal (UART frame) from the SCI, the waveform is corrected and the signal is converted to an IR frame serial output signal by the IrDA module as shown in figure 14.25.

When the serial data is 0, if the PSEL bit is 0 in the IrDA mode register (SCIMR) a pulse of 3/16 the IR frame bit width is generated and output, and if the PSEL bit is 1 a pulse of 3/16 the bit width of the bit rate set in bits ICK3 to 0 in the serial mode register (SCSMR) is generated and output. When the serial data is 1, a pulse is not output.

An infrared LED is driven by a signal demodulated to a 3/16 width.

Reception

Received pulses of 3/16 the IR frame bit width are converted to UART frames after demodulation as shown in figure 14.25.

When RIVS = 0 in the SCIMR register, demodulation to 0 is executed for pulse output and demodulation to 1 when there is no pulse output.

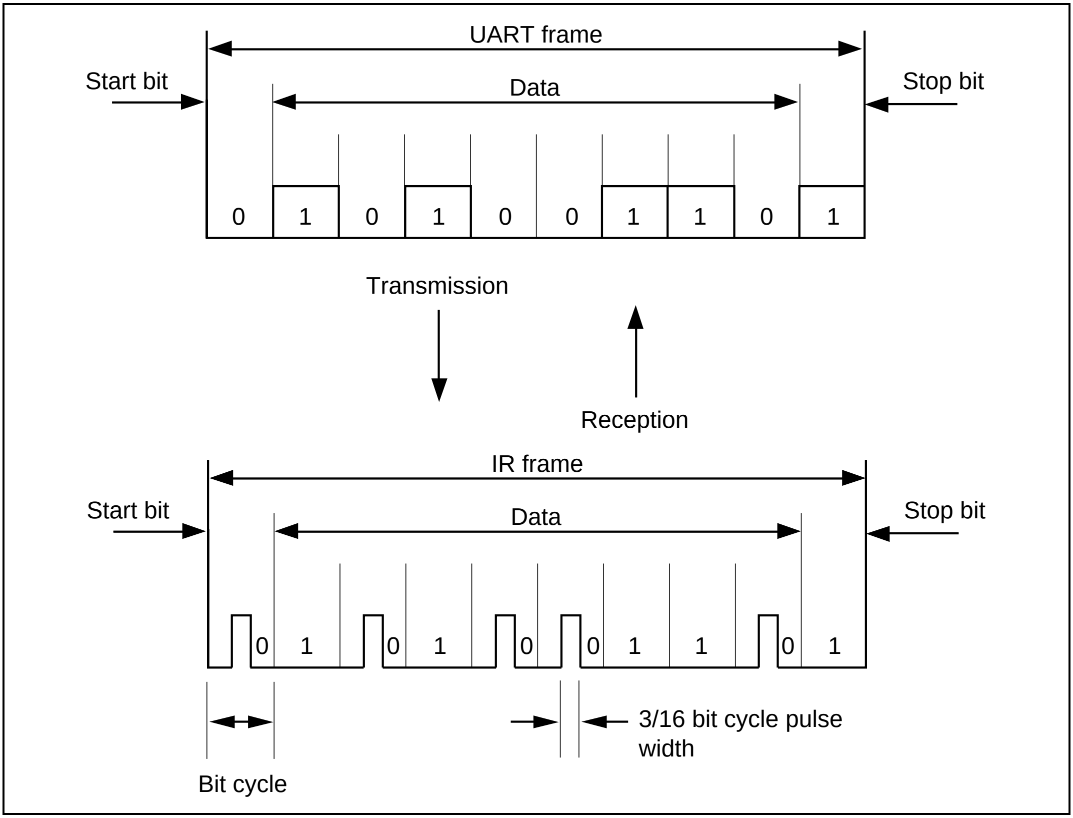


Figure 14.25 IrDA Mode Transmit/Receive Operations

Pulse Width Selection

In transition, the IR frame pulse width can be selected as either 3/16 of the transmission bit rate or a smaller pulse width by means of the PSEL bit in the IrDA mode register (SCIMR).

The SCI includes a baud rate generator that generates the transmit frame bit rate and a baud rate generator that generates the IRCLK signal for varying the pulse width.

When the PSEL bit is cleared to 0 in SCIMR, a width of 3/16 the bit rate set in the bit rate register (SCBRR) is output as the IR frame pulse width. As the pulse width is the direct infrared emission time; if the user wishes to minimize the pulse width in order to reduce power consumption, the PSEL bit should be set to 1 in SCIMR and a setting should also be made in bits ICK3 to ICK0 in the serial mode register (SCSMR) to generate the IRCLK signal, resulting in output with the minimum settable pulse width.

The minimum IR frame pulse width must be 3/16 of the 115.2 kbps bit rate (= 1.63 μs). With this minimum pulse width, IRCLK = 921.6 kHz, and so the setting for bits ICK3 to ICK0 to give the minimum settable pulse width is given by the following equation.

Pϕ: Operating clock frequency

IRCLK: 921.6 kHz (fixed)
N: Set value of ICK3 to ICK0 ($0 \le N \le 15$)

$$N \geq \frac{P\phi}{2 \times IRCLK} - 1$$

For example, when Pϕ = 20 MHz, N = 10.

Table 14.12 shows the settings of bits ICK3 to ICK0 that can be used to obtain the minimum pulse width for various operating frequencies.

Table 14.12 Bits ICK3 to ICK0 and Operating Frequencies in IrDA mode(When PSEL = 1)

Operating Frequency	Setting of Bits ICK3 to ICK0			
Pϕ (MHz)	ICK3	ICK2	ICK1	ICK0
2	0	0	0	0
3				1
5			1	0
6				1
8		1	0	0
10				1
12			1	0
14				1
16	1	0	0	0
18				1
20			1	0
21				1
22				1
23		1	0	0
24				1
25				1
26			1	0
27				0
28				1

14.4 SCI Interrupt Sources and the DMAC

The SCI has four interrupt sources: the transmit-end interrupt (TEI) request, receive-error interrupt (ERI) request, receive-FIFO-data-full interrupt (RXI) request, and transmit-FIFO-data-empty interrupt (TXI) request.

Table 14.13 shows the interrupt sources and their relative priorities. Individual interrupt sources can be enabled or disabled with the TIE, RIE, and TEIE bits in SCSCR. Each kind of interrupt request is sent to the interrupt controller independently.

When the TDFE flag is set to 1 in the serial status register (SC1SSR), a TXI interrupt is requested. A TXI interrupt request can activate the on-chip DMAC to perform data transfer. The TDFE bit is cleared to 0 automatically when all transmit FIFO data register (SCFTDR) writes by the DMAC are completed.

When the RDF flag is set to 1 in SC1SSR, an RXI interrupt is requested. An RXI interrupt request can activate the on-chip DMAC to perform data transfer. The RDF bit is cleared to 0 automatically when all receive FIFO data register (SCFRDR) reads by the DMAC are completed.

When the ER or DR flag is set to 1 in SC2SSR, an ERI interrupt is requested. The on-chip DMAC cannot be activated by an ERI interrupt request.

When the TEND flag is set to 1 in SC1SSR, a TEI interrupt is requested. The on-chip DMAC cannot be activated by a TEI interrupt request.

A TXI interrupt indicates that transmit data can be written, and an RXI interrupt indicates that there is receive data in SCFRDR. The TEI interrupt indicates that the transmit operation has ended.

Table 14.13 SCI Interrupt Sources

Interrupt Source	Description	DMAC Activation	Priority on Reset Release
ERI	Receive error (ER or DR)	Not possible	High
RXI	Receive FIFO data register full (RDF)	Possible	↑
TXI	Transmit FIFO data register empty (TDFE)	Possible	↓
TEI	Transmit end (TEND)	Not possible	Low

14.5 Usage Notes

The following points should be noted when using the SCI.

SCFTDR Writing and the TDFE Flag

The TDFE flag in the serial status 1 register (SC1SSR) is set when the number of transmit data bytes written in the transmit FIFO data register (SCFTDR) has fallen to or below the transmit trigger number set by bits TTRG1 and TTRG0 in the FIFO control register (SCFCR). After TDFE is set, transmit data up to the number of empty bytes in SCFTDR can be written, allowing efficient continuous transmission.

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 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 the upper 8 bits of the FIFO data count register (SCFDR).

Simultaneous Multiple Receive Errors

If a number of receive errors occur at the same time, the state of the status flags in SC1SSR is as shown in table 14.14. If there is an overrun error, data is not transferred from the receive shift register (SCRSR) to the receive FIFO data register (SCFRDR), and the receive data is lost.

Table 14.14 SC1SSR Status Flags and Transfer of Receive Data

	SC1SSR Status Flags				Receive Data Transfer
Receive Errors	**RDF**	**ORER**	**FER**	**PER**	**SCRSR → SCFRDR**
Overrun error	1	1	0	0	×
Framing error	0	0	1	0	O
Parity error	0	0	0	1	O
Overrun error + framing error	1	1	1	0	×
Overrun error + parity error	1	1	0	1	×
Framing error + parity error	0	0	1	1	O
Overrun error + framing error + parity error	1	1	1	1	×

Legend:

O: Receive data is transferred from SCRSR to SCFRDR.

×: Receive data is not transferred from SCRSR to SCFRDR.

Break Detection and Processing

Break signals can be detected by reading the RxD pin directly when a framing error (FER) is detected. In the break state the input from the RxD pin consists of all 0s, so the FER flag is set and the parity error flag (PER) may also be set. Note that although the SCI stops transferring receive data to SCFRDR after receiving a break, the receive operation continues, so if the FER and BRK flags are cleared to 0 they will be set to 1 again.

Sending a Break Signal

The TxD pin is a general I/O pin whose input/output direction and level are determined by the I/O port data register (DR) and the control register (CR) of the pin function controller (PFC). This fact can be used to send a break signal.

The DR value substitutes for the mark state until the PFC setting is made. The initial setting should therefore be as an output port outputting 1.

To send a break signal during serial transmission, clear DR to 0, then set the TxD pin as an output port with the PFC.

When the TE bit is cleared to 0, the transmitter is initialized regardless of the current transmission state.

Receive Error Flags and Transmit Operations (Synchronous Mode Only)

Transmission cannot be started when a receive error flag (ORER, PER3 to 0, or FER3 to 0) is set to 1, even if the TE bit is set to 1. Be sure to clear the receive error flags to 0 before starting transmission.

Note also that the receive error flags are not cleared to 0 by clearing the RE bit to 0.

Receive Data Sampling Timing and Receive Margin in Asynchronous Mode

The SCI operates on a base clock with a frequency of 16, 8, or 4 times the transfer rate.

In reception, the SCI 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, fourth, or second base clock pulse. The timing is shown in figure 14.26.

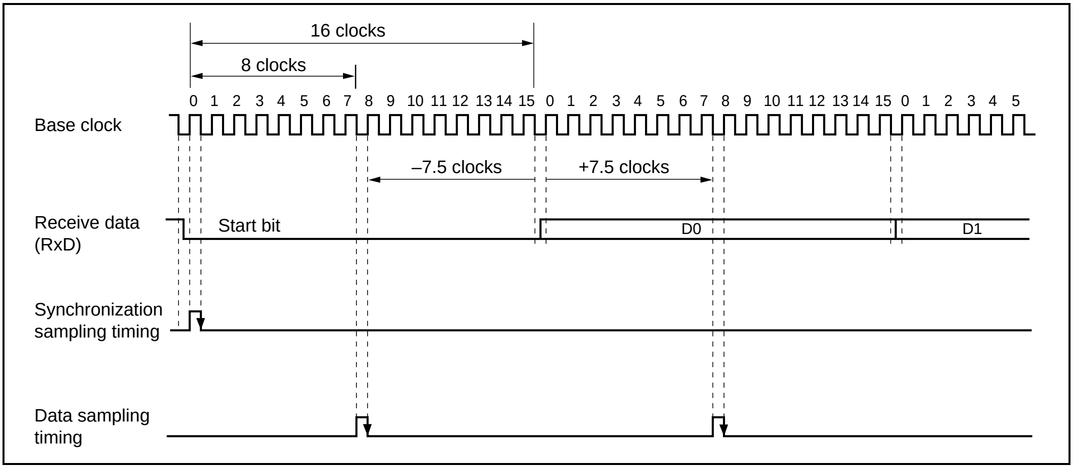


Figure 14.26 Receive Data Sampling Timing in Asynchronous Mode (Using base clock with frequency of 16 times the transfer rate, sampled in 8th clock cycle)

The receive margin in asynchronous mode can therefore be expressed as shown in equation (1).

$$M = \left| \left(0.5 - \frac{1}{2N}\right) - (L - 0.5)\,F - \frac{|D - 0.5|}{N}\,(1 + F) \right| \times 100\% \quad \text{.........................} \quad (1)$$

M: Receive margin (%)
N: Ratio of clock frequency to bit rate (N = 16, 8, or 4)
D: Clock duty cycle (D = 0 to 1.0)
L: Frame length (L = 9 to 12)
F: Absolute deviation of clock frequency

From equation (1), if F = 0 and D = 0.5, the receive margin is 46.875%, as given by equation (2).

When D = 0.5, F = 0, and N = 16:

$$\begin{aligned} M &= (0.5 - 1/(2 \times 16)) \times 100\% \\ &= 46.875\% \quad \text{..} \quad (2) \end{aligned}$$

This is a theoretical value. A reasonable margin to allow in system designs is 20% to 30%.

When Using Synchronous External Clock Mode

- Do not set TE or RE to 1 until at least 4 peripheral operating clock cycles after external clock SCK has changed from 0 to 1.
- Only set both TE and RE to 1 when external clock SCK is 1.
- In reception, note that if RE is cleared to 0 from 2.3 to 3.5 peripheral operating clock cycles after the rising edge of the RxD D7 bit SCK input, RDF will be set to 1 but copying to SCFRDR will not be possible.

When Using Synchronous Internal Clock Mode

In reception, note that if RE is cleared to 0 1.5 peripheral operating clock cycles after the rising edge of the RxD D7 bit SCK output, RDF will be set to 1 but copying to SCFRDR will not be possible.

When Using the DMAC

When an external clock source is used as the serial clock, the transmit clock should not be input until at least 5 Pϕ clock cycles after SCFTDR is updated by the DMAC. Incorrect operation may result if the transmit clock is input within 4 Pϕ cycles after SCFTDR is updated. (See figure 14.27.)

When performing SCFRDR reads by the DMAC, be sure to set the relevant SCI receive-FIFO-data-full interrupt (RXI) as an activation source.

Also, it is recommended that the FCS bit be set to 1 (flag clearing performed every bus cycle) and the DS bit be set to 1 (falling edge detection) in the DMAC's CHCRn register, as in section 9.4.1, Example of DMA Transfer between On-Chip SCI and External Memory.

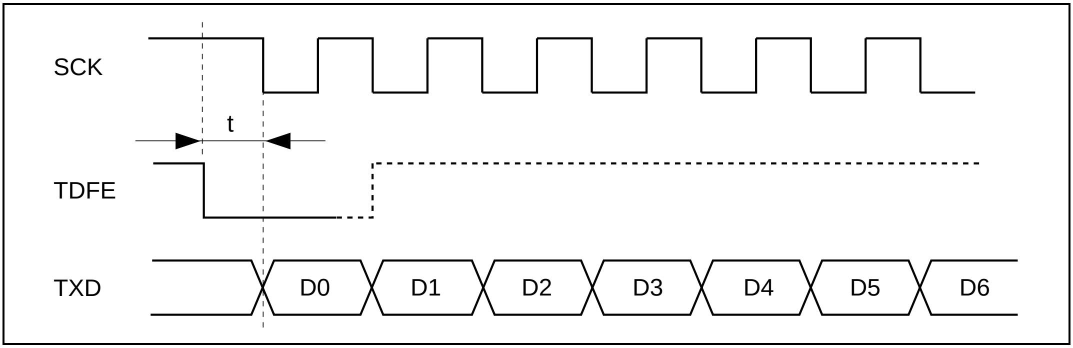


Figure 14.27 Example of Synchronous Transmission by DMAC

SCFRDR Reading and the RDF Flag

The RDF flag in the serial status 1 register (SC1SSR) is set when the number of receive data bytes in the receive FIFO data register (SCFRDR) has become equal to or greater than the receive trigger number set by bits RTRG1 and RTRG0 in the FIFO control register (SCFCR). After RDF is set, receive data equivalent to the trigger number can be read from SCFRDR, allowing efficient continuous reception.

If the number of data bytes in SCFRDR is equal to or greater than the trigger number, the RDF flag will be set to 1 again if it is cleared to 0. RDF should therefore be cleared to 0 after being read as 1 after receive data has been read to reduce the number of data bytes in SCFRDR to less than the trigger number.

The number of receive data bytes in SCFRDR can be found from the lower 8 bits of the FIFO data count register (SCFDR).

Section 15 A/D Converter

15.1 Overview

The on-chip A/D converter has 10-bit resolution and allows selection of up to 8 analog input channels.

The A/D converter is composed of two independent modules, A/D0 and A/D1.

15.1.1 Features

The A/D converter has the following features:

- 10-bit resolution
- Eight input channels (4 channels × 2)
- Conversion time

  Minimum conversion time (per channel): 6.7 µs (20 MHz, CKS = 1)

  Operating frequency: Pϕ > 20 MHz, CKS = 0
  Pϕ ≤ 20 MHz, CKS = 0, 1
- Choice of conversion mode

  Single mode or multi mode can be selected.

  Conversion can be carried out simultaneously on two channels.
- Three conversion start methods

  Software, timer conversion start trigger (MMT), TPU or $\overline{\text{ADTRG}}$ pin can be selected.
- Eight A/D data registers

  Conversion results are held in 16-bit data registers for each channel.
- Sample and hold function
- A/D conversion end interrupt

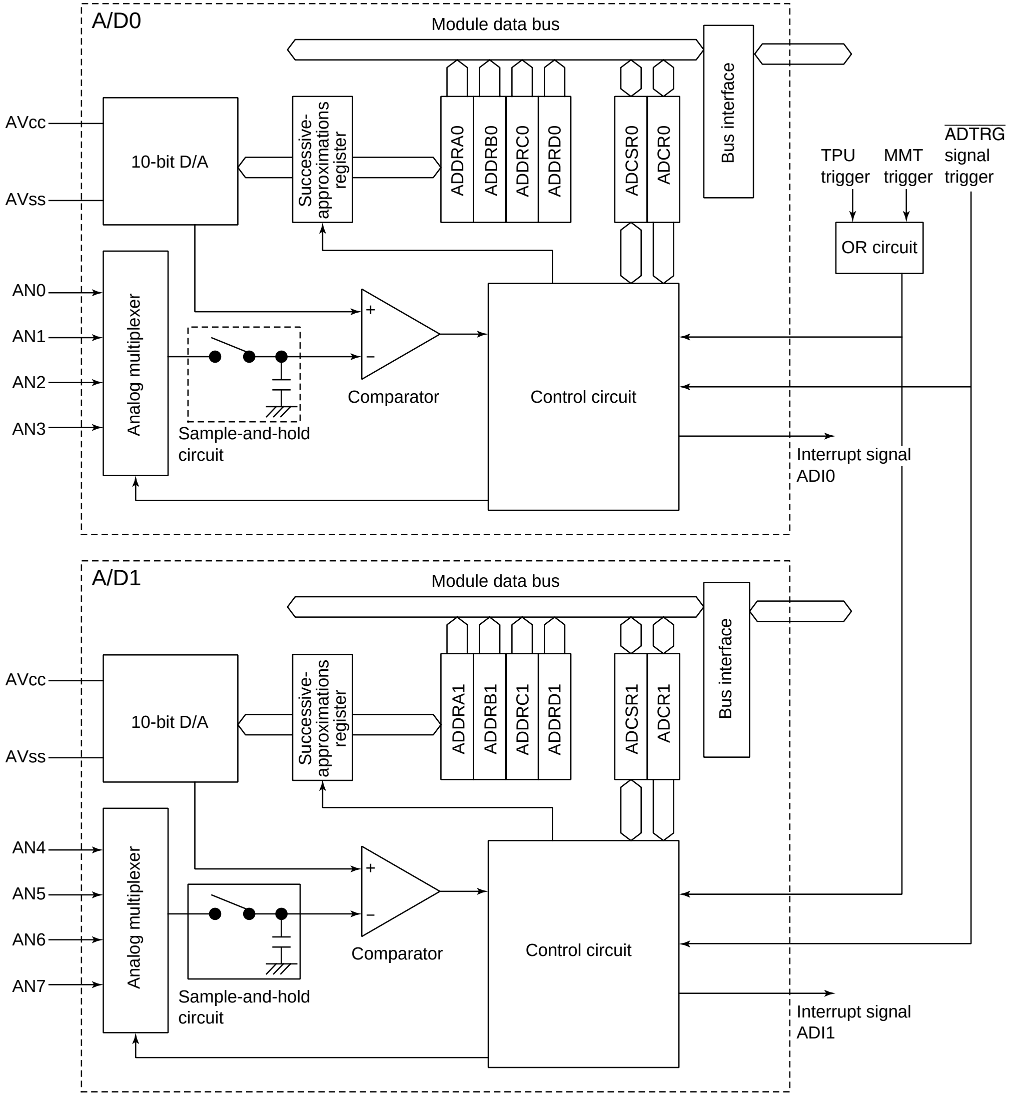
  An A/D conversion end interrupt (ADI) can be requested on completion of A/D conversion.

  The DMAC can be activated by ADI0 (A/D0 interrupt request) and ADI1 (A/D1 interrupt request).

15.1.2 Block Diagram

Figure 15.1 shows a block diagram of the A/D converter.

AV_{CC} and AV_{SS} for both A/D modules are common pins in the chip.



Legend:

ADCR0: A/D0 control register
ADCSR0: A/D0 control/status register
ADDRA0: A/D0 data register A
ADDRB0: A/D0 data register B
ADDRC0: A/D0 data register C
ADDRD0: A/D0 data register D

ADCR1: A/D1 control register
ADCSR1: A/D1 control/status register
ADDRA1: A/D1 data register A
ADDRB1: A/D1 data register B
ADDRC1: A/D1 data register C
ADDRD1: A/D1 data register D

Figure 15.1 Block Diagram of A/D Converter

15.1.3 Pin Configuration

Figure 15.1 shows the pins used by the A/D converter.

AV_{CC} and AV_{SS} are power supply pins for the analog circuits in the A/D converter.

Table 15.1 A/D Converter Pins

Pin Name		Abbreviation	I/O	Function
Analog power supply		AV_{CC}	Input	Analog circuit power supply and A/D conversion reference voltage
Analog ground		AV_{SS}	Input	Analog circuit ground and A/D conversion reference ground
A/D0	Analog input 0	AN0	Input	Analog input channel 0
	Analog input 1	AN1	Input	Analog input channel 1
	Analog input 2	AN2	Input	Analog input channel 2
	Analog input 3	AN3	Input	Analog input channel 3
A/D1	Analog input 4	AN4	Input	Analog input channel 4
	Analog input 5	AN5	Input	Analog input channel 5
	Analog input 6	AN6	Input	Analog input channel 6
	Analog input 7	AN7	Input	Analog input channel 7
A/D external trigger input		ADTRG	Input	External trigger for starting A/D conversion

15.1.4 Register Configuration

Table 15.2 summarizes the A/D converter's registers.

Table 15.2 A/D Converter Registers

Name	Abbreviation	R/W	Initial Value	Address	Access Size
A/D0 data register AH	ADDRA0H	R	H'00	H'FFFF0080	8, 16
A/D0 data register AL	ADDRA0L	R	H'00	H'FFFF0081	8
A/D0 data register BH	ADDRB0H	R	H'00	H'FFFF0082	8, 16
A/D0 data register BL	ADDRB0L	R	H'00	H'FFFF0083	8
A/D0 data register CH	ADDRC0H	R	H'00	H'FFFF0084	8, 16
A/D0 data register CL	ADDRC0L	R	H'00	H'FFFF0085	8
A/D0 data register DH	ADDRD0H	R	H'00	H'FFFF0086	8, 16
A/D0 data register DL	ADDRD0L	R	H'00	H'FFFF0087	8
A/D0 control/status register	ADCSR0	R/(W)*	H'00	H'FFFF0098	8, 16
A/D0 control register	ADCR0	R/W	H'3F	H'FFFF0099	8
A/D1 data register AH	ADDRA1H	R	H'00	H'FFFF00A0	8, 16
A/D1 data register AL	ADDRA1L	R	H'00	H'FFFF00A1	8
A/D1 data register BH	ADDRB1H	R	H'00	H'FFFF00A2	8, 16
A/D1 data register BL	ADDRB1L	R	H'00	H'FFFF00A3	8
A/D1 data register CH	ADDRC1H	R	H'00	H'FFFF00A4	8, 16
A/D1 data register CL	ADDRC1L	R	H'00	H'FFFF00A5	8
A/D1 data register DH	ADDRD1H	R	H'00	H'FFFF00A6	8, 16
A/D1 data register DL	ADDRD1L	R	H'00	H'FFFF00A7	8
A/D1 control/status register	ADCSR1	R/(W)*	H'00	H'FFFF00B8	8, 16
A/D1 control register	ADCR1	R/W	H'3F	H'FFFF00B9	8

Note: * Bit 7 can only be written with 0, to clear the flag.

15.2 Register Descriptions

15.2.1 A/D Data Registers A to D (ADDRA0 to ADDRD0, ADDRA1 to ADDRD1)

Bit:	15	14	13	12	11	10	9	8
ADDRn	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit:	7	6	5	4	3	2	1	0
ADDRn	AD1	AD0	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Note: n = A to D

The A/D data registers (ADDR) are 16-bit read-only registers that store the results of A/D conversion. There are eight registers, ADDRA0 to ADDRD0, ADDRA1 to ADDRD1.

The result of A/D conversion is 10-bit data which is transferred to and stored in the ADDR register for the selected channel. The upper 8 bits of the converted data correspond to the upper byte of ADDR, and the lower 2 bits correspond to the lower byte. Bits 5 to 0 of the lower byte of ADDR are reserved, and are always read as 0. Table 15.3 shows the correspondence between the analog input channels and the A/D data registers.

The ADDR registers can be read by the CPU at all times. The upper byte is read directly, but the lower byte data is transferred via a temporary register (TEMP). For details, see section 15.3, CPU Interface.

The ADDR registers are initialized to H'0000 by a power-on reset and in standby mode.

Table 15.3 Analog Input Channels and A/D Data Registers

Analog Input Channel	A/D Data Register	Module
AN0	ADDRA0	A/D0
AN1	ADDRB0	
AN2	ADDRC0	
AN3	ADDRD0	
AN4	ADDRA1	A/D1
AN5	ADDRB1	
AN6	ADDRC1	
AN7	ADDRD1	

15.2.2 A/D Control/Status Registers (ADCSR0, ADCSR1)

Bit:	7	6	5	4	3	2	1	0
	ADF	ADIE	ADST	MULTI	CKS	—	CH1	CH0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/(W)*	R/W	R/W	R/W	R/W	R	R/W	R/W

Note: * Only 0 can be written, to clear the flag.

The A/D control/status registers (ADCSR0, ADCSR1) are 8-bit readable/writable registers that perform A/D converter operation control, including selection of the A/D conversion mode. ADCSR0 is used for A/D0, and ADCSR1 for A/D1.

The ADCSR registers are initialized to H'00 by a power-on reset and in standby mode.

Bit 7—A/D End Flag (ADF): Indicates the end of A/D conversion.

Bit 7: ADF	Description
0	[Clearing conditions] (Initial value) • When 0 is written to ADF after reading ADF = 1 • When the DMAC is activated by and ADI interrupt, and an A/D converter register is accessed
1	[Setting conditions] • Single mode: When A/D conversion ends • Multi mode: When A/D conversion ends on all selected channels

Bit 6—A/D Interrupt Enable (ADIE): Enables or disables the interrupt request (ADI) at the end of A/D conversion.

Bit 6: ADIE	Description	
0	A/D end interrupt request (ADI) is disabled	(Initial value)
1	A/D end interrupt request (ADI) is enabled	

Bit 5—A/D Start (ADST): Starts or stops A/D conversion. The ADST bit remains set to 1 during A/D conversion. It can also be set to 1 by a conversion start trigger from a timer (MMT, TPU), and by means of the A/D conversion trigger input pin ($\overline{\text{ADTRG}}$).

Bit 5: ADST	Description	
0	A/D conversion is stopped	(Initial value)
1	• Single mode: A/D conversion is started. ADST is automatically cleared to 0 when A/D conversion ends on the specified channel. • Multi mode: A/D conversion is started. Conversion continues, once on each channel in turn, until ADST is cleared to 0 by software.	

Bit 4—Multi Mode (MULTI): Selects single mode or multi mode as the A/D conversion mode. For details of the operation in single mode and multi mode, see section 15.4, Operation. Change the mode only when ADST = 0.

Bit 4: MULTI	Description	
0	Single mode	(initial value)
1	Multi mode	

Bit 3—Clock Select (CKS): Sets the A/D conversion time. Change the conversion time only when ADST = 0.

Bit 3: CKS	Description	
0	Conversion time = 266 states (max.)	(Initial value)
1	Conversion time = 134 states (max.)	

Bit 2—Reserved: This bit is always read as 0 and should only be written with 0.

Bits 1 and 0—Channel Select 1 and 0 (CH1, CH0): These bits, together with the multi mode bit, select the analog input channels. Change the channel selection only when ADST = 0.

Channel Select Bits		Description			
		Single Mode		Multi Mode	
CH1	**CH0**	**A/D0**	**A/D1**	**A/D0**	**A/D1**
0	0	AN0 (Initial value)	AN4 (Initial value)	AN0	AN4
	1	AN1	AN5	AN0, AN1	AN4, AN5
1	0	AN2	AN6	AN0–AN2	AN4–AN6
	1	AN3	AN7	AN0–AN3	AN4–AN7

15.2.3 A/D Control Registers (ADCR0, ADCR1)

Bit:	7	6	5	4	3	2	1	0
	TRGE1	TRGE0	—	—	—	—	—	—
Initial value:	0	0	1	1	1	1	1	1
R/W:	R/W	R/W	R	R	R	R	R	R

The A/D control registers (ADCR0, ADCR1) are 8-bit readable/writable registers that enable or disable starting of A/D conversion by external trigger input. ADCR0 is used for A/D0, and ADCR1 for A/D1.

The ADCR registers are initialized to H'3F by a power-on reset and in standby mode.

Bits 7 and 6—Trigger Enable (TRGE1, TRGE0): These bits enable or disable starting of A/D conversion by input of an $\overline{\text{ADTRG}}$ pin or an MMT or TPU trigger.

Bit 7: TRGE1	Bit 6: TRGE0	Description
0	0	Start of A/D conversion by $\overline{\text{ADTRG}}$ pin or MTT/TPU trigger input is disabled (Initial value)
	1	A/D conversion is started at MMT or TPU trigger input
1	0	Start of A/D conversion by $\overline{\text{ADTRG}}$ pin or MTT/TPU trigger input is enabled
	1	A/D conversion is started at falling edge of $\overline{\text{ADTRG}}$ pin

The $\overline{\text{ADTRG}}$ pin and the MMT/TPU trigger are shared by A/D0 and A/D1. The settings for A/D0 and A/D1 are ORed.

Bits 5 to 0—Reserved: These bits are always read as 1 and should only be written with 1.

15.3 CPU Interface

The A/D data registers (ADDRA0 to ADDRD0, ADDRA1 to ADDRD1) are 16-bit registers, but they are connected to the CPU by an 8-bit data bus. Therefore, the upper and lower bytes of these registers must be read separately.

To prevent the data being changed between the reads of the upper and lower bytes of an A/D data register, the lower byte is read via a temporary register (TEMP). The upper byte can be read directly.

Data is read from an A/D data register as follows. When the upper byte is read, the upper-byte value is transferred directly to the CPU and the lower-byte value is transferred into TEMP. Next, when the lower byte is read, the TEMP contents are transferred to the CPU.

When performing byte-size reads on an A/D data register, always read the upper byte before the lower byte. It is possible to read only the upper byte, but if only the lower byte is read, incorrect data may be obtained. If a word-size read is performed on an A/D data register, reading is performed in upper byte, lower byte order automatically.

Figure 15.2 shows the data flow when reading an A/D data register.

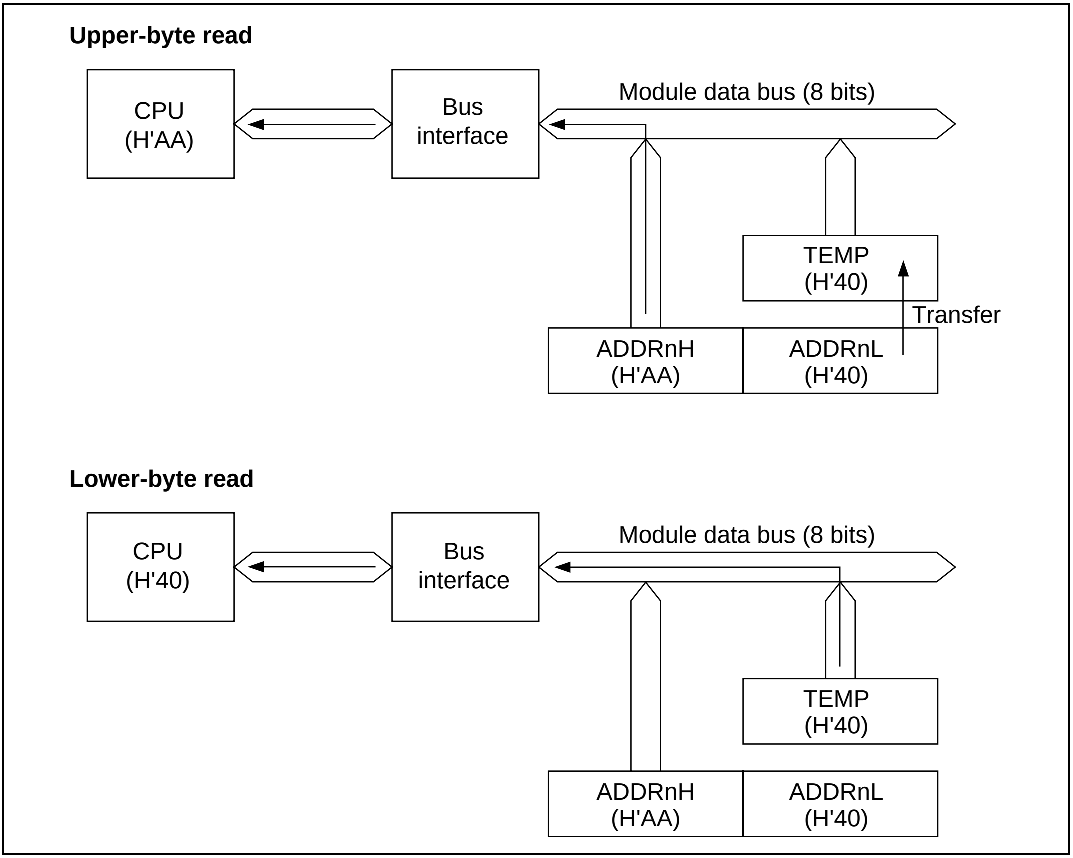


Figure 15.2 A/D Data Register Access Operation (Reading H'AA40)

15.4 Operation

The A/D converter operates by successive approximations with 10-bit resolution. It has two operating modes: single mode and multi mode. The operation in these two modes is described below.

15.4.1 Single Mode (MULTI = 0)

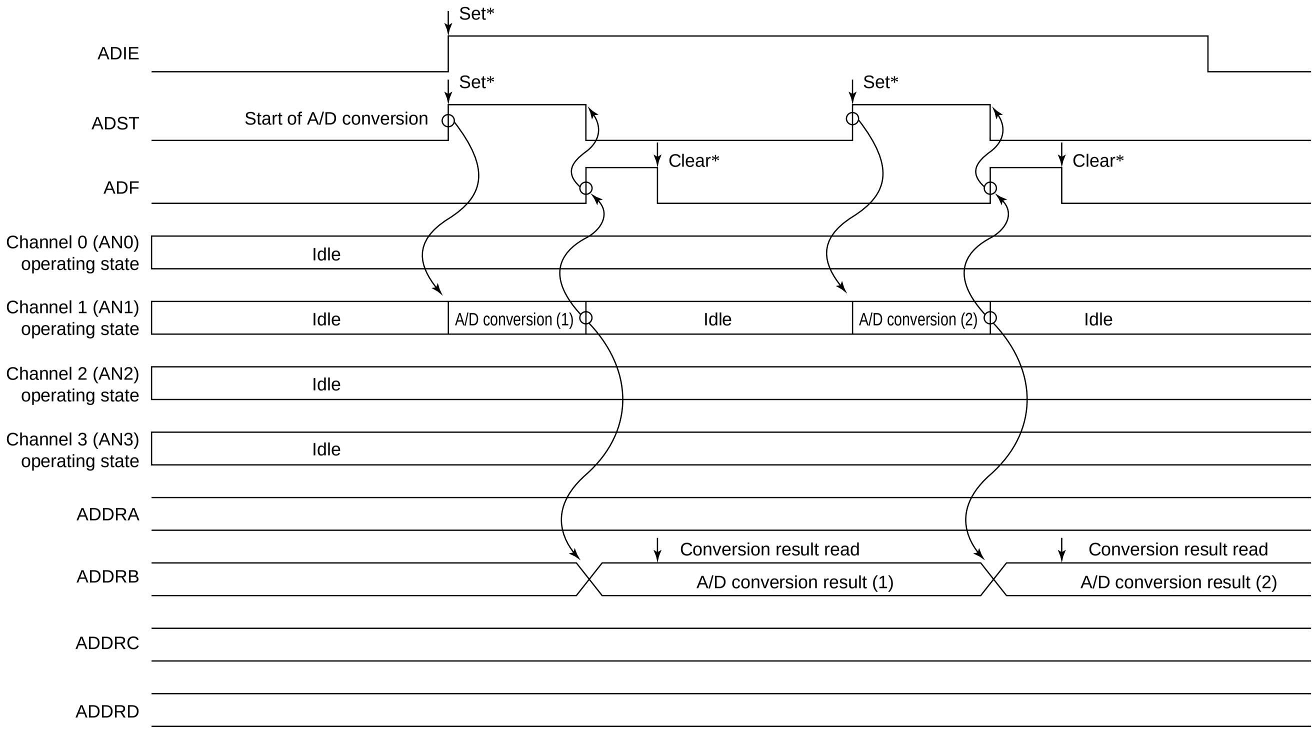
Single mode should be selected for A/D conversion on only one channel. A/D conversion starts when the ADST bit in the A/D control/status register (ADCSR) is set to 1 by software or external trigger input. The ADST bit remains set to 1 during A/D conversion, and is automatically cleared to 0 when conversion ends.

When conversion ends, the ADF bit in ADCSR is set to 1. If the ADIE bit in ADCSR is also 1, an ADI interrupt is requested. To clear the ADF bit, first read ADF when set to 1, then write 0 to ADF.

To prevent incorrect operation, A/D conversion should be halted by clearing the ADST bit to 0 before changing the mode or analog input channel. After the change is made, A/D conversion is restarted by setting the ADST bit to 1 (the mode or channel change and setting of the ADST bit can be carried out simultaneously).

An example of the operation when channel 1 (AN1) is selected and A/D conversion is performed in single mode is described below. Figure 15.3 shows a timing diagram for this example (bit specifications in the operation example refer to the ADCSR0 register).

1. Single mode is selected (MULTI = 0), input channel AN1 is selected (CH1 = 0, CH0 = 1), the A/D interrupt request is enabled (ADIE = 1), and A/D conversion is started (ADST = 1).
2. When A/D conversion ends, the result is transferred to ADDRB0. At the same time ADF is set to 1, ADST is cleared to 0, and the A/D converter becomes idle.
3. Since ADF = 1 and ADIE = 1, an ADI interrupt is requested.
4. The A/D interrupt service routine is started.
5. The routine reads ADF set to 1, then writes 0 to ADF.
6. The routine reads and processes the conversion result (ADDRB0).
7. Execution of the A/D interrupt service routine ends. After this, if the ADST bit is set to 1, A/D conversion starts again and steps (2) to (7) are repeated.



Note: * Vertical arrows (↓) indicate instructions executed by software.

Figure 15.3 Example of A/D Converter Operation (Single Mode, Channel 1 Selected)

15.4.2 Multi Mode

Multi mode is useful for monitoring analog inputs in a group of one or more channels. When the ADST bit in the A/D control/status register (ADCSR) is set to 1 by software or external trigger input, A/D conversion starts on the first channel in the group (AN0 in A/D0, or AN4 in A/D1).

When more than one channel has been selected, A/D conversion starts on the second channel (AN1 or AN5) as soon as conversion ends on the first channel.

After A/D conversion has been performed once on each of the selected channels, the ADST bit is cleared to 0 automatically. The conversion results are transferred to and stored in the ADDR register for each channel.

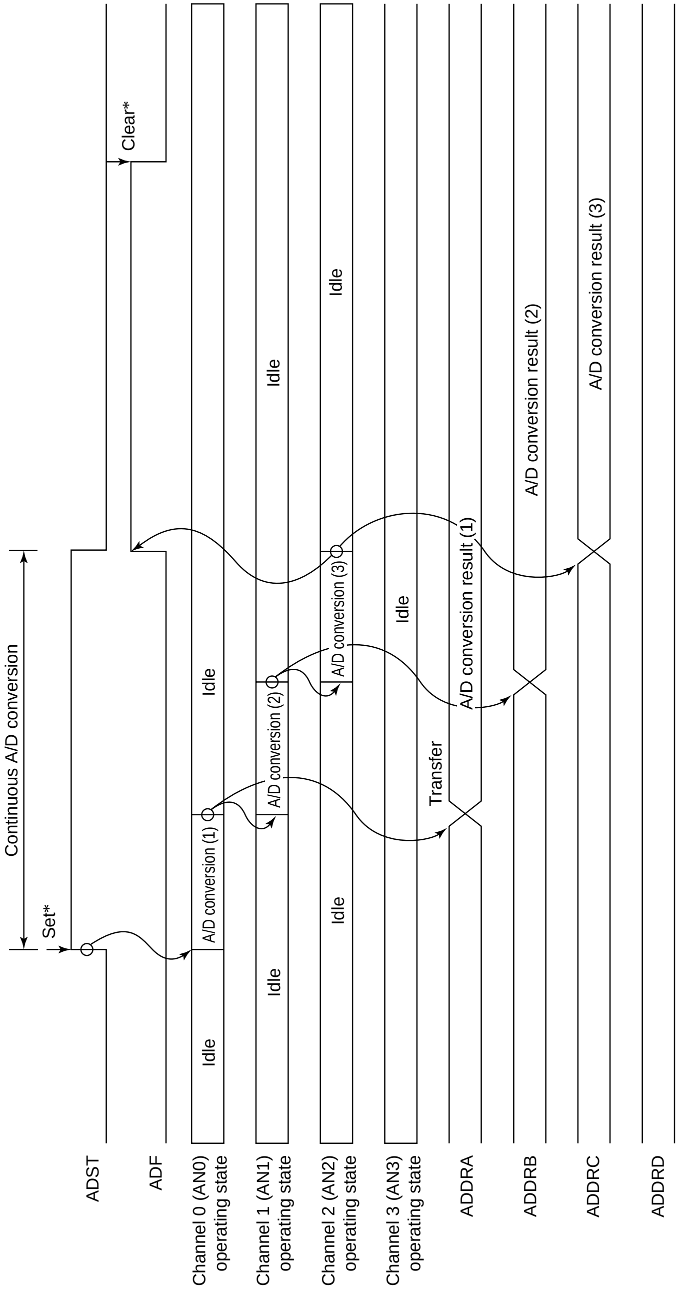
To prevent incorrect operation, A/D conversion should be halted by clearing the ADST bit to 0 before changing the mode or analog input channels. After the change is made, the first channel is selected and A/D conversion is restarted by setting the ADST bit to 1 (the mode or channel change and setting of the ADST bit can be carried out simultaneously).

An example of the A/D conversion operation in multi mode when three channels (AN0 to AN2) in group 0 are selected is described below. Figure 15.4 shows a timing diagram for this example (bit specifications in the operation example refer to the ADCSR0 register).

1. Multi mode is selected (MULTI = 1), analog input channels AN0 to AN2 are selected (CH1 = 1, CH0 = 0), and A/D conversion is started (ADST = 1).
2. A/D conversion starts on the first channel (AN0), and when completed, the result is transferred to ADDRA0. Conversion then starts automatically on the second channel (AN1).
3. Conversion proceeds in the same way through the third channel (AN2).
4. When conversion is completed for all the selected channels (AN0 to AN2), ADF is set to 1 ADST is cleared to 0, and conversion stops.

   If the ADIE bit is 1, an ADI interrupt is requested when conversion ends.

   When the ADST bit is cleared to 0, A/D conversion stops.
5. ADF is read while set to 1, then written with 0. After this, if the ADST bit is set to 1, A/D conversion starts again from the first channel (AN0).


ADST
ADF
Channel 0 (AN0) operating state
Channel 1 (AN1) operating state
Channel 2 (AN2) operating state
Channel 3 (AN3) operating state
ADDRA
ADDRB
ADDRC
ADDRD
Continuous A/D conversion
Set*
Clear*
Idle
A/D conversion (1)
A/D conversion (2)
A/D conversion (3)
Transfer
A/D conversion result (1)
A/D conversion result (2)
A/D conversion result (3)

Note: * Vertical arrows (↓) indicate instructions executed by software.

Figure 15.4 Example of A/D Converter Operation (Multi Mode, Channels AN0 to AN2 Selected)

15.4.3 Input Sampling and A/D Conversion Time

The A/D converter has a built-in sample and hold circuit. The A/D converter samples the analog input at time t_D after access to the A/D control/status register (ADCSR) is started, then starts conversion. Figure 15.5 shows the A/D conversion timing, and table 15.4 shows A/D conversion times.

As shown in figure 15.5, A/D conversion time t_{CONV} includes A/D conversion start delay time t_D and analog input sampling time t_{SPL}. The length of t_D is not fixed, but is determined by the timing of the write to ADSCR. The total conversion time therefore varies within the ranges shown in table 15.4.

In multi mode, the t_{CONV} values given in table 15.4 apply to the first conversion. In the second and subsequent conversions, t_{CONV} is fixed at 266 states (Pϕ) when CKS = 0, or 134 states (Pϕ) when CKS = 1.

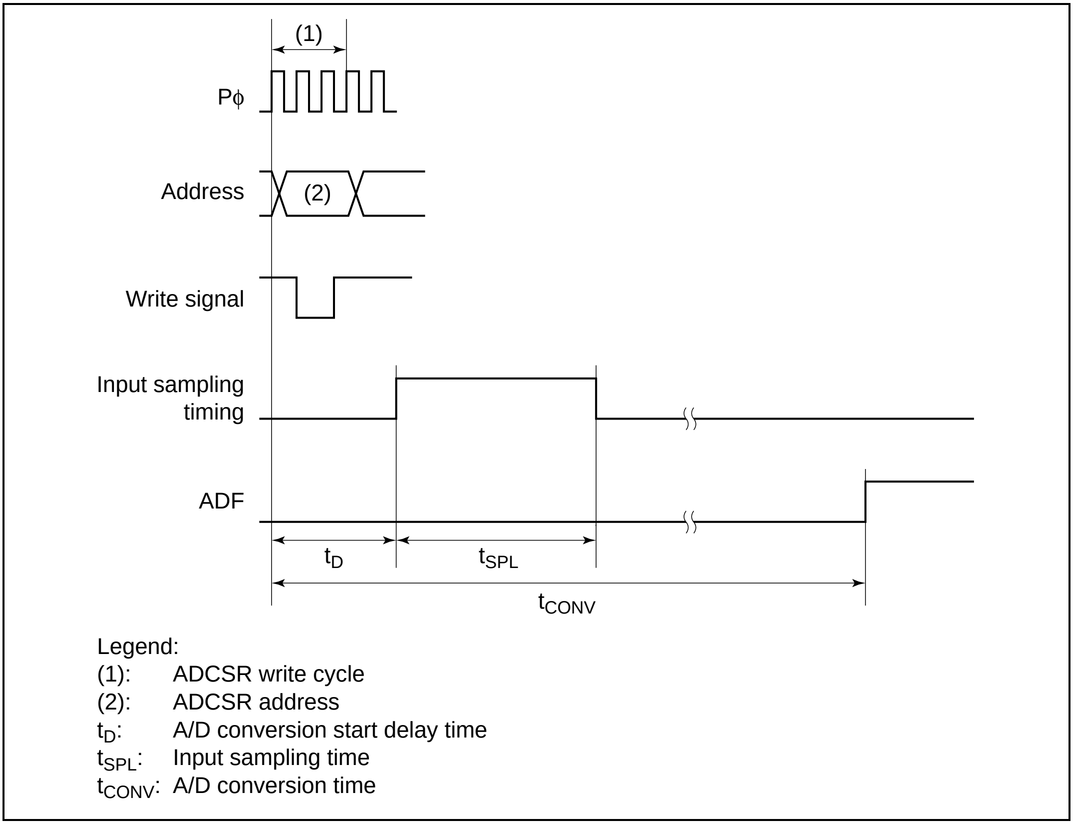


Figure 15.5 A/D Conversion Timing

Table 15.4 A/D Conversion Times (Single Mode)

		CKS = 0			CKS = 1		
	Symbol	Min	Typ	Max	Min	Typ	Max
A/D conversion start delay time	t_D	10	—	17	6	—	9
Input sampling time	t_{SPL}	—	64	—	—	32	—
A/D conversion time	t_{CONV}	259	—	266	131	—	134

Note: Unit: states (Pϕ)

15.4.4 External Trigger Input Timing

A/D conversion can also be started by external trigger input. When the TRGE bit is set to 1 in the A/D control register (ADCR), an external trigger is input from the $\overline{\text{ADTRG}}$ pin, or from the MMT or TPU.

When a falling edge on the $\overline{\text{ADTRG}}$ input pin or an MMT trigger is detected, the ADST bit is set to 1 in the A/D control/status register (ADCSR), and A/D conversion starts.

Other operations, for both single mode and multi mode, are the same as when the ADST bit is set to 1 by software.

Figure 15.6 shows the timing for external trigger input.

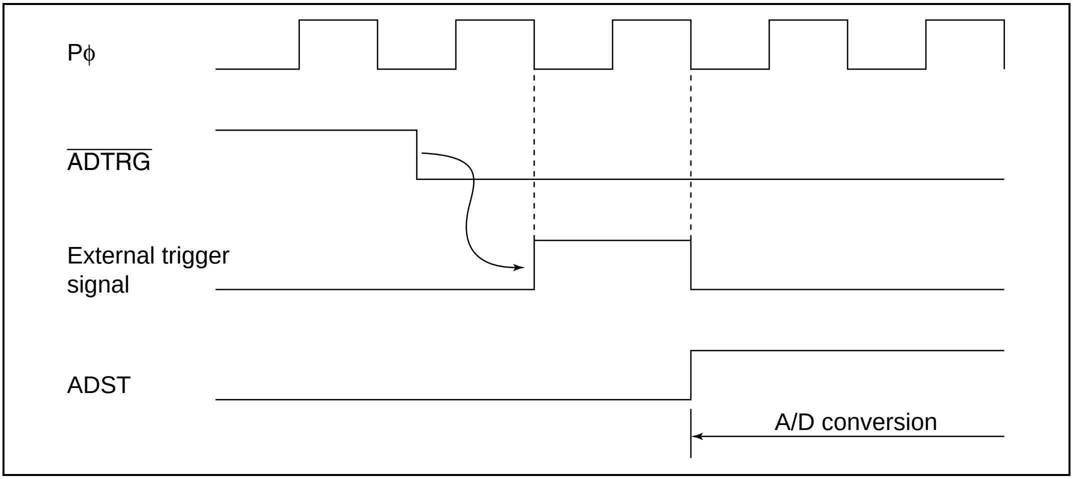


Figure 15.6 Timing of External Trigger Input by Means of $\overline{\text{ADTRG}}$ Pin

15.5 Interrupt Sources and DMA Transfer Requests

The A/D converter generates an A/D conversion end interrupt (ADI) on completion of A/D conversion. The ADI interrupt can be enabled or disabled with the ADIE bit in ADCSR. DMA transfer can be initiated by an ADI interrupt.

When the DMAC is activated by an ADI interrupt, the ADF bit in the A/D control/status register (ADCSR) is automatically cleared to 0 when an A/D register is accessed.

15.6 A/D Conversion Accuracy Definitions

The A/D converter converts analog values input from analog input channels to 10-bit digital values by comparing them with an analog reference voltage. In this operation, the absolute accuracy of the A/D conversion (i.e. the deviation between the input analog value and the output digital value) includes the following kinds of error.

1. Offset error
2. Full-scale error
3. Quantization error
4. Nonlinearity error

The above four kinds of error are described below with reference to figure 15.7. For the sake of clarity, this figure shows 3-bit A/D conversion rather than 10-bit A/D conversion. Offset error (see figure 15.7 (1)) is the deviation between the actual A/D conversion characteristic and the ideal A/D conversion characteristic when the digital output value changes from the minimum value (zero voltage) of 0000000000 (000 in the figure) to 0000000001 (001 in the figure). Full-scale error (see figure 15.7 (2)) is the deviation between the actual A/D conversion characteristic and the ideal A/D conversion characteristic when the digital output value changes from 1111111110 (110 in the figure) to the maximum value (full-scale voltage) of 1111111111 (111 in the figure). Quantization error is the deviation inherent in the A/D converter, given by 1/2 LSB (see figure 15.7 (3)). Nonlinearity error is the deviation between the actual A/D conversion characteristic and the ideal A/D conversion characteristic from zero voltage to full-scale voltage (see figure 15.7 (4)). This does not include offset error, full-scale error, and quantization error.

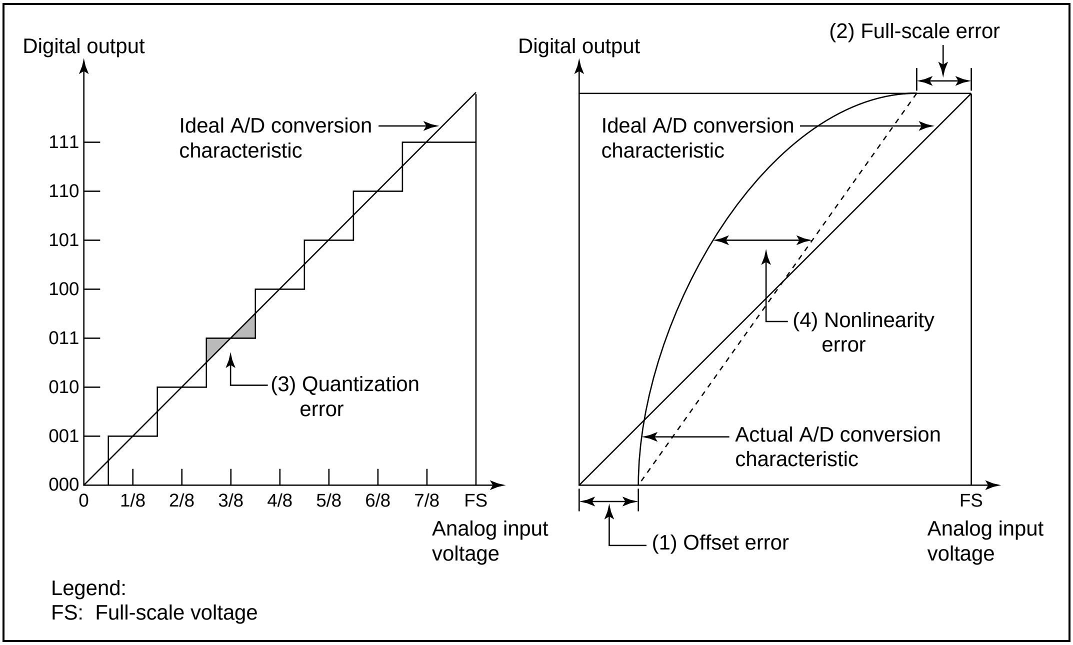


Figure 15.7 A/D Conversion Accuracy Definitions

15.7 Usage Notes

The following points should be noted when using the A/D converter.

15.7.1 Analog Voltage Settings

1. Analog input voltage range

   The voltage applied to analog input pins during A/D conversion should be in the range $AV_{SS} \leq ANn \leq AV_{CC}$ (n = 0 to 7).

2. AV_{CC} and AV_{SS} input voltages

   For the AV_{CC} and AV_{SS} input voltages, set $AV_{CC} = V_{CC} \pm 10\%$, and $AV_{SS} = V_{SS}$. When the A/D converter is not used, set $AV_{CC} = V_{CC}$ and $AV_{SS} = V_{SS}$.

3. AV_{CC} must be connected to the power supply (V_{CC}) even when the A/D converter is not used, and in standby mode.

15.7.2 Handling of Analog Input Pins

To prevent damage from surges and other abnormal voltages at the analog input pins (AN0-AN7), a protection circuit such as that shown in figure 15.8 should be connected. This circuit also includes a CR filter function that suppresses error due to noise. The circuit shown here is only a design example; circuit constants must be decided on the basis of the actual operating conditions.

Figure 15.9 shows an equivalent circuit for the analog input pins, and table 15.5 summarizes the analog input pin specifications.

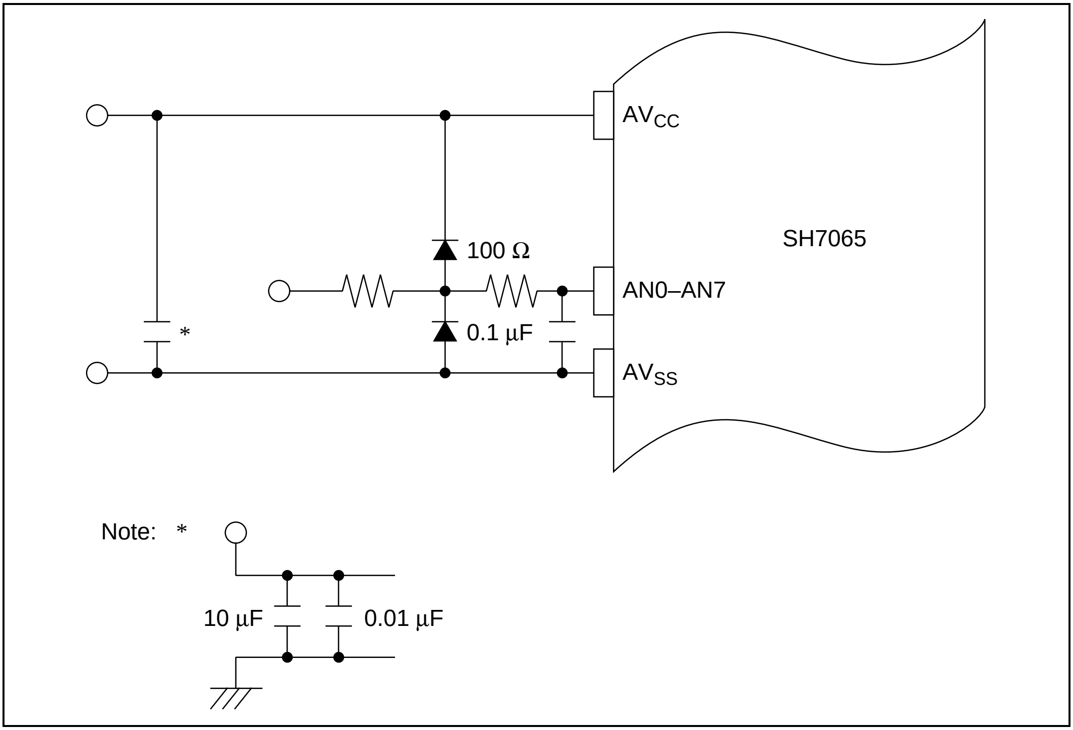
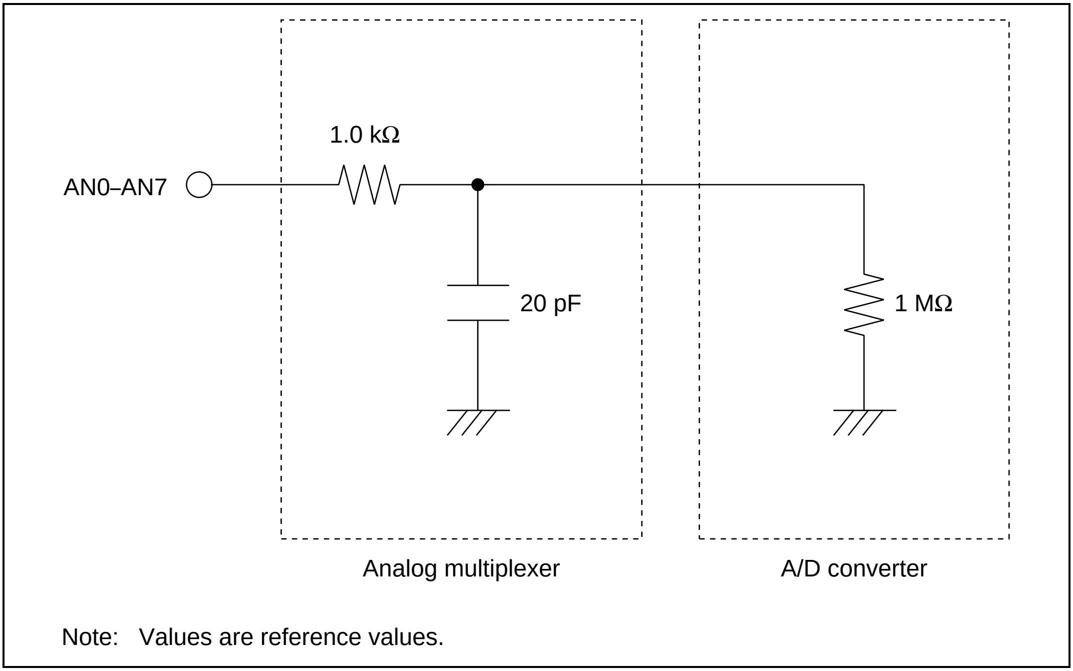


Figure 15.8 Example of Analog Input Pin Protection Circuit



Note: Values are reference values.

Figure 15.9 Analog Input Pin Equivalent Circuit

Table 15.5 Analog Input Pin Specifications

Item	Min	Max	Unit
Analog input capacitance	—	20	pF
Permissible signal source impedance	—	1	kΩ

15.7.3 Note on PH0 and PH1 Output

The PH0 and PH1 outputs must not be changed during A/D conversion, as the conversion accuracy cannot be guaranteed in this case.

15.7.4 Port I PFC Settings

Function switching for port I pins, which are used as A/D converter analog input pins, is performed automatically when A/D conversion is started, so no PFC settings are necessary for port I.

15.7.5 Simultaneous A/D and D/A Conversion

If A/D conversion is started during D/A conversion and analog voltage output while the DAE bit is cleared to 0 in the D/A converter's D/A control register (DACR), the analog power supply current increases sharply, and noise may occur in the analog output of the D/A converter. When using the A/D converter and D/A converter simultaneously, noise in the analog output can be prevented by setting the DAE bit in DACR to 1 beforehand.

However, when the DAE bit is set to 1, even if bits DAOE0 and DAOE1 in DACR and the ADST bit in ADCSR are cleared to 0, the same current is drawn from the analog power supply as during simultaneous A/D and D/A conversion.

Section 16 D/A Converter

16.1 Overview

The SH7065 includes a two-channel D/A converter.

16.1.1 Features

The D/A converter has the following features:

- 8-bit resolution
- Two output channels
- Conversion time: maximum 10 µs (with 20 pF capacitive load)
- Output voltage: 0 V to AV_{CC}

16.1.2 Block Diagram

Figure 16.1 shows a block diagram of the D/A converter.

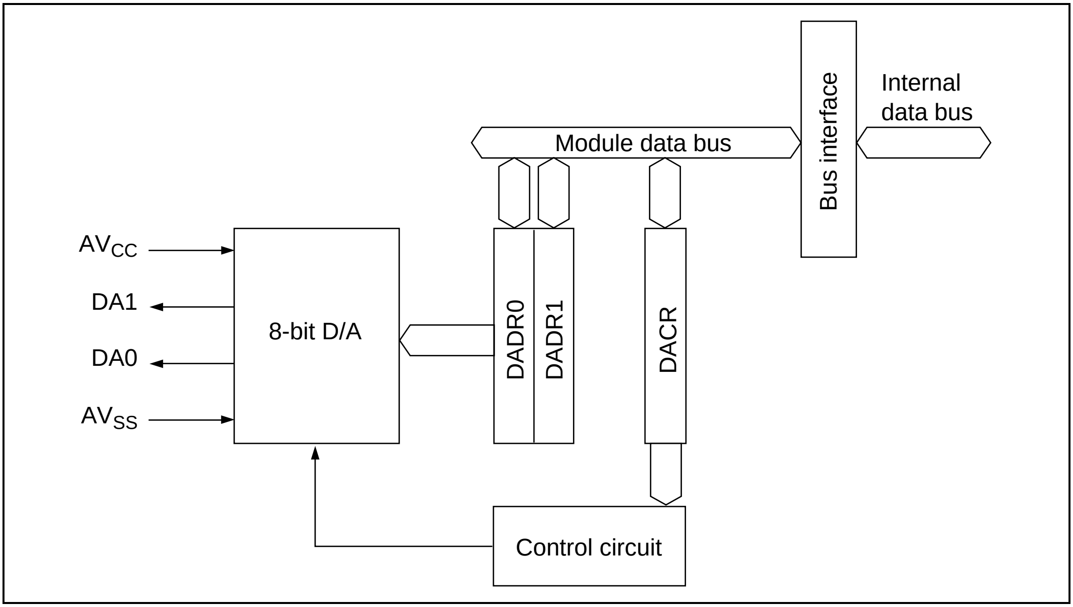


Figure 16.1 Block Diagram of D/A Converter

16.1.3 Pin Configuration

Table 16.1 summarizes the input and output pins of the D/A converter.

Table 16.1 D/A Converter Pins

Pin Name	Abbreviation	I/O	Function
Analog power supply pin	AV_{CC}	Input	Analog power supply
Analog ground pin	AV_{SS}	Input	Analog ground and reference voltage
Analog output pin 0	DA0	Output	Channel 0 analog output
Analog output pin 1	DA1	Output	Channel 1 analog output

16.1.4 Register Configuration

Table 16.2 summarizes the registers of the D/A converter.

Table 16.2 D/A Converter Registers

Name	Abbreviation	R/W	Initial Value	Address	Access Size
D/A data register 0	DADR0	R/W	H'00	H'FFFF00C0	8, 16
D/A data register 1	DADR1	R/W	H'00	H'FFFF00C1	8, 16
D/A control register	DACR	R/W	H'1F	H'FFFF00C2	8, 16

16.2 Register Descriptions

16.2.1 D/A Data Registers 0 and 1 (DADR0, DADR1)

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

D/A data registers 0 and 1 (DADR0 and DADR1) are 8-bit readable/writable registers that store the data to be converted. When analog output is enabled, the values in DADR0 and DADR1 are constantly converted and output at the analog output pins.

The D/A data registers are initialized to H'00 by a reset and in standby mode.

16.2.2 D/A Control Register (DACR)

Bit:	7	6	5	4	3	2	1	0
	DAOE1	DAOE0	DAE	—	—	—	—	—
Initial value:	0	0	0	1	1	1	1	1
R/W:	R/W	R/W	R/W	—	—	—	—	—

The D/A control register (DACR) is an 8-bit readable/writable register that controls the operation of the D/A converter.

DACR is initialized to H'1F by a reset and in standby mode.

Bit 7—D/A Output Enable 1 (DAOE1): Controls D/A conversion and analog output.

Bit 7: DAOE1	Description	
0	DA1 analog output is disabled	(Initial value)
1	Channel 1 D/A conversion and DA1 analog output are enabled	

Bit 6—D/A Output Enable 0 (DAOE0): Controls D/A conversion and analog output.

Bit 6: DAOE0	Description	
0	DA0 analog output is disabled	(Initial value)
1	Channel 0 D/A conversion and DA0 analog output are enabled	

Bit 5—D/A Enable (DAE): Controls D/A conversion, together with bits DAOE0 and DAOE1. When the DAE bit is cleared to 0, D/A conversion is controlled independently in channels 0 and 1.

			Description			
			Channel 1		Channel 0	
Bit 7: DAOE1	Bit 6: DAOE0	Bit 5: DAE	D/A Conversion	Analog Output	D/A Conversion	Analog Output
0	0	0	Halted	Halted	Halted	Halted
		1	Executed		Executed	
	1	0	Halted			Executed
		1	Executed			
1	0	0	Executed	Executed	Halted	Halted
		1			Executed	
	1	0				Executed
		1				

When using the A/D converter and D/A converter simultaneously, set the DAE bit to 1. This will prevent the noise associated with the start of A/D converter operation.

When the DAE bit is set to 1, even if bits DAOE0 and DAOE1 in DACR and the ADST bit in ADCSR are cleared to 0, the same current is drawn from the analog power supply as during simultaneous A/D and D/A conversion.

Bits 4 to 0—Reserved: Read-only bits, always read as 1.

16.3 Operation

The D/A converter has two built-in D/A conversion circuits that can perform conversion independently.

D/A conversion is performed constantly while enabled in DACR. If the DADR0 or DADR1 value is modified, conversion of the new data begins immediately. The conversion results are output when bits DAOE0 and DAOE1 are set to 1.

An example of D/A conversion on channel 0 is given below. The timing is shown in figure 16.2.

1. Data to be converted is written in DADR0.
2. Bit DAOE0 is set to 1 in DACR. D/A conversion starts and DA0 becomes an output pin. The conversion result is output after the conversion time. The output value is (DADR0 contents/256) × AV_{CC}. Output of this conversion result continues until the value in DADR0 is modified or the DAOE0 bit is cleared to 0.
3. If the DADR0 value is modified, conversion starts immediately, and the result is output after the conversion time.
4. When the DAOE0 bit is cleared to 0, DA0 becomes an input pin.

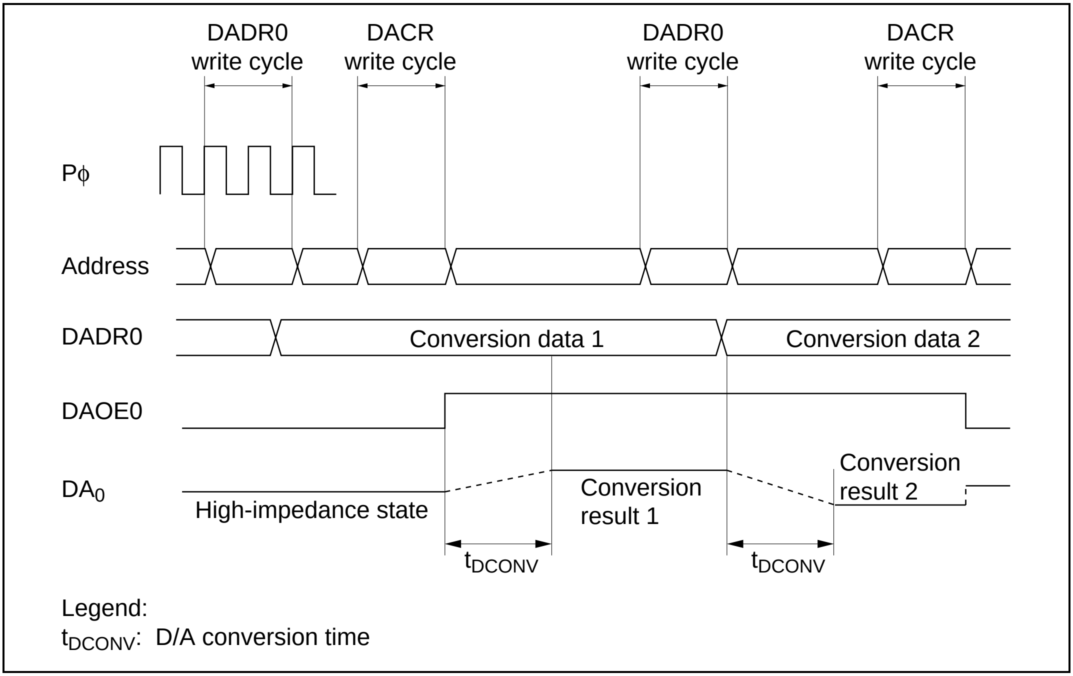


Figure 16.2 Example of D/A Converter Operation

16.4 Usage Note

If the digital output at the PH1 pin is changed during analog output from the DA0 pin, noise may be introduced into the DA0 analog output. Similarly, if the digital output at the PH0 pin is changed during analog output from the DA1 pin, noise may be introduced into the DA1 analog output. Caution is therefore necessary if the PH0 or PH1 output level is changed during analog output.

DA0 and DA1 pin settings should be made with the port H PFC. Noise may occur in the analog output if the A/D converter is used at the same time. This can be prevented by setting the DAE bit to 1 in DACR. For details see section 15.7.5, Simultaneous A/D and D/A Conversion.

Section 17 Pin Function Controller (PFC)

17.1 Overview

The pin function controller (PFC) consists of registers for selecting multiplex pin functions and their input/output direction. Tables 17.1 to 17.9 show the SH7065's multiplex pins. The functions of the multiplex pins are determined by the operating mode. Table 17.10 shows the pin functions in each operating mode, together with the initial functions.

Usage notes apply to this PFC, so the contents of section 17.4, PFC Restrictions, should be noted before using the PFC.

Table 17.1 Multiplex Pins (Port A)

Port	Function 1 (Related Module)	Function 2 (Related Module)	Function 3 (Related Module)	Function 4 (Related Module)	Function 5 (Related Module)
A	PA25 I/O (port)	$\overline{\text{CS5}}$ output (BSC)	—	—	—
	PA24 I/O (port)	$\overline{\text{CS4}}$ output (BSC)	—	—	—
	PA23 I/O (port)	$\overline{\text{CS3}}$ output (BSC)	—	—	—
	PA22 I/O (port)	$\overline{\text{CS2}}$ output (BSC)	—	—	—
	PA21 I/O (port)	$\overline{\text{CS1}}$ output (BSC)	—	—	—
	PA20 I/O (port)	$\overline{\text{CS0}}$ output (BSC)	—	—	—
	PA19 I/O (port)	$\overline{\text{BS}}$ output (BSC)	—	—	—
	PA18 I/O (port)	$\overline{\text{RD}}$ output (BSC)	—	—	—
	PA17 I/O (port)	$\overline{\text{WR}}$ output (BSC)	—	—	—
	PA16 I/O (port)	$\overline{\text{WRHH}}$ output (BSC)	$\overline{\text{HHBS}}$ output (BSC)	TCLKC input (TPU)	TIOC3A I/O (TPU)
	PA15 I/O (port)	$\overline{\text{WRHL}}$ output (BSC)	$\overline{\text{HLBS}}$ output (BSC)	TCLKD input (TPU)	TIOC3B I/O (TPU)
	PA14 I/O (port)	$\overline{\text{WRLH}}$ output (BSC)	$\overline{\text{LHBS}}$ output (BSC)	—	—
	PA13 I/O (port)	$\overline{\text{WRLL}}$ output (BSC)	$\overline{\text{LLBS}}$ output (BSC)	—	—
	PA12 I/O (port)	$\overline{\text{WAIT}}$ input (BSC)	—	—	—
	PA9 I/O (port)	$\overline{\text{RAS1}}$ output (BSC)	—	—	—
	PA8 I/O (port)	$\overline{\text{RAS0}}$ output (BSC)	—	—	—
	PA1 I/O (port)	$\overline{\text{OE1}}$ output (BSC)	—	—	—
	PA0 I/O (port)	$\overline{\text{OE0}}$ output (BSC)	—	—	—

Table 17.2 Multiplex Pins (Port B)

Port	Function 1 (Related Module)	Function 2 (Related Module)	Function 3 (Related Module)	Function 4 (Related Module)	Function 5 (Related Module)
B	PB23 I/O (port)	$\overline{\text{CASHH1}}$ output (BSC)	TxD1 output (SCI)	$\overline{\text{TEND0}}$ output (DMAC)	—
	PB22 I/O (port)	$\overline{\text{CASHL1}}$ output (BSC)	RxD1 input (SCI)	$\overline{\text{TEND1}}$ output (DMAC)	—
	PB21 I/O (port)	$\overline{\text{CASLH1}}$ output (BSC)	—	—	—
	PB20 I/O (port)	$\overline{\text{CASLL1}}$ output (BSC)	—	—	—
	PB19 I/O (port)	$\overline{\text{CASHH0}}$ output (BSC)	TxD0 output (SCI)	—	—
	PB18 I/O (port)	$\overline{\text{CASHL0}}$ output (BSC)	RxD0 input (SCI)	—	—
	PB17 I/O (port)	$\overline{\text{CASLH0}}$ output (BSC)	—	—	—
	PB16 I/O (port)	$\overline{\text{CASLL0}}$ output (BSC)	—	—	—
	PB13 I/O (port)	RDWR output (BSC)	—	—	—
	PB7 I/O (port)	$\overline{\text{BACK}}$ output (BSC)	—	—	—
	PB6 I/O (port)	$\overline{\text{BREQ}}$ input (BSC)	—	—	—

Table 17.3 Multiplex Pins (Port C)

Port	Function 1 (Related Module)	Function 2 (Related Module)	Function 3 (Related Module)	Function 4 (Related Module)
C	PC25 I/O (port)	A25 output (BSC)	TIOC3B I/O (TPU)	TCLKD input (TPU)
	PC24 I/O (port)	A24 output (BSC)	TIOC3A I/O (TPU)	TCLKC input (TPU)
	PC23 I/O (port)	A23 output (BSC)	TIOC1B I/O (TPU)	TCLKB input (TPU)
	PC22 I/O (port)	A22 output (BSC)	TIOC1A I/O (TPU)	TCLKA input (TPU)
	PC21 I/O (port)	A21 output (BSC)	TIOC5B I/O (TPU)	—
	PC20 I/O (port)	A20 output (BSC)	TIOC5A I/O (TPU)	—
	PC19 I/O (port)	A19 output (BSC)	TIOC4B I/O (TPU)	—
	PC18 I/O (port)	A18 output (BSC)	TIOC4A I/O (TPU)	—
	PC17 I/O (port)	A17 output (BSC)	TIOC3B I/O (TPU)	—
	PC16 I/O (port)	A16 output (BSC)	TIOC3A I/O (TPU)	—
	PC15 I/O (port)	A15 output (BSC)	TIOC3D I/O (TPU)	—
	PC14 I/O (port)	A14 output (BSC)	TIOC3C I/O (TPU)	—
	PC13 I/O (port)	A13 output (BSC)	—	—
	PC12 I/O (port)	A12 output (BSC)	—	—
	PC11 I/O (port)	A11 output (BSC)	—	—
	PC10 I/O (port)	A10 output (BSC)	—	—
	PC9 I/O (port)	A9 output (BSC)	—	—
	PC8 I/O (port)	A8 output (BSC)	—	—
	PC7 I/O (port)	A7 output (BSC)	—	—
	PC6 I/O (port)	A6 output (BSC)	—	—
	PC5 I/O (port)	A5 output (BSC)	—	—
	PC4 I/O (port)	A4 output (BSC)	—	—
	PC3 I/O (port)	A3 output (BSC)	—	—
	PC2 I/O (port)	A2 output (BSC)	—	—
	PC1 I/O (port)	A1 output (BSC)	—	—
	PC0 I/O (port)	A0 output (BSC)	—	—

Table 17.4 Multiplex Pins (Port D)

Port	Function 1 (Related Module)	Function 2 (Related Module)	Function 3 (Related Module)	Function 4 (Related Module)	Function 5 (RelatedModule)
D	PD31 I/O (port)	D31 I/O (BSC)	RxD2 input (SCI)	TIOC5A I/O (TPU)	—
	PD30 I/O (port)	D30 I/O (BSC)	TxD2 output (SCI)	TIOC4B I/O (TPU)	—
	PD29 I/O (port)	D29 I/O (BSC)	SCK2 I/O (SCI)	TIOC4A I/O (TPU)	—
	PD28 I/O (port)	D28 I/O (BSC)	TCLKB input (TPU)	TIOC3D I/O (TPU)	—
	PD27 I/O (port)	D27 I/O (BSC)	TCLKA input (TPU)	TIOC3C I/O (TPU)	—
	PD26 I/O (port)	D26 I/O (BSC)	PWOB output (MMT)	—	—
	PD25 I/O (port)	D25 I/O (BSC)	PVOB output (MMT)	—	—
	PD24 I/O (port)	D24 I/O (BSC)	PUOB output (MMT)	—	—
	PD23 I/O (port)	D23 I/O (BSC)	PCO output (MMT)	PCI input (MMT)	SCK1 I/O (SCI)
	PD22 I/O (port)	D22 I/O (BSC)	PWOA output (MMT)	SCK0 I/O (SCI)	—
	PD21 I/O (port)	D21 I/O (BSC)	PVOA output (MMT)	$\overline{\text{IRQ7}}$ input (INTC)	—
	PD20 I/O (port)	D20 I/O (BSC)	PUOA output (MMT)	$\overline{\text{IRQ6}}$ input (INTC)	—
	PD19 I/O (port)	D19 I/O (BSC)	$\overline{\text{POE3}}$ input (MMT)	$\overline{\text{IRQ5}}$ input (INTC)	—
	PD18 I/O (port)	D18 I/O (BSC)	$\overline{\text{POE2}}$ input (MMT)	$\overline{\text{IRQ4}}$ input (INTC)	—
	PD17 I/O (port)	D17 I/O (BSC)	$\overline{\text{POE1}}$ input (MMT)	$\overline{\text{ADTRG}}$ input (A/D)	—
	PD16 I/O (port)	D16 I/O (BSC)	$\overline{\text{POE0}}$ input (MMT)	—	—
	PD15 I/O (port)	D15 I/O (BSC)	TIOC5B I/O (TPU)	—	—
	PD14 I/O (port)	D14 I/O (BSC)	TIOC5A I/O (TPU)	—	—
	PD13 I/O (port)	D13 I/O (BSC)	TIOC4B I/O (TPU)	—	—
	PD12 I/O (port)	D12 I/O (BSC)	TIOC4A I/O (TPU)	—	—
	PD11 I/O (port)	D11 I/O (BSC)	TIOC2B I/O (TPU)	—	—
	PD10 I/O (port)	D10 I/O (BSC)	TIOC2A I/O (TPU)	—	—
	PD9 I/O (port)	D9 I/O (BSC)	TIOC1B I/O (TPU)	—	—
	PD8 I/O (port)	D8 I/O (BSC)	TIOC1A I/O (TPU)	—	—
	PD7 I/O (port)	D7 I/O (BSC)	—	—	—
	PD6 I/O (port)	D6 I/O (BSC)	—	—	—
	PD5 I/O (port)	D5 I/O (BSC)	—	—	—
	PD4 I/O (port)	D4 I/O (BSC)	—	—	—
	PD3 I/O (port)	D3 I/O (BSC)	—	—	—
	PD2 I/O (port)	D2 I/O (BSC)	—	—	—
	PD1 I/O (port)	D1 I/O (BSC)	—	—	—
	PD0 I/O (port)	D0 I/O (BSC)	—	—	—

Table 17.5 Multiplex Pins (Port E)

Port	Function 1 (Related Module)	Function 2 (Related Module)	Function 3 (Related Module)	Function 4 (Related Module)
E	PE23 I/O (port)	$\overline{\text{IRQ7}}$ input (INTC)	PWOB output (MMT)	—
	PE22 I/O (port)	$\overline{\text{IRQ6}}$ input (INTC)	PVOB output (MMT)	—
	PE21 I/O (port)	$\overline{\text{IRQ5}}$ input (INTC)	PUOB output (MMT)	—
	PE20 I/O (port)	$\overline{\text{IRQ4}}$ input (INTC)	PCO output (MMT)	PCI input (MMT)
	PE19 I/O (port)	$\overline{\text{IRQ3}}$ input (INTC)	PWOA output (MMT)	—
	PE18 I/O (port)	$\overline{\text{IRQ2}}$ input (INTC)	PVOA output (MMT)	—
	PE17 I/O (port)	$\overline{\text{IRQ1}}$ input (INTC)	PUOA output (MMT)	SCK0 I/O (SCI)
	PE16 I/O (port)	$\overline{\text{IRQ0}}$ input (INTC)	SCK1 I/O (SCI)	$\overline{\text{AH}}$ output (BSC)
	PE15 I/O (port)	$\overline{\text{IRQ7}}$ input (INTC)	—	—
	PE14 I/O (port)	$\overline{\text{IRQ6}}$ input (INTC)	—	—
	PE13 I/O (port)	$\overline{\text{IRQ5}}$ input (INTC)	—	—
	PE12 I/O (port)	$\overline{\text{IRQ4}}$ input (INTC)	—	—

Table 17.6 Multiplex Pins (Port F)

Port	Function 1 (Related Module)	Function 2 (Related Module)	Function 3 (Related Module)	Function 4 (Related Module)
F	PF7 I/O (port)	$\overline{\text{DREQ1}}$ input (DMAC)	$\overline{\text{IRQOUT}}$ output (INTC)	TIOC0D I/O (TPU)
	PF6 I/O (port)	$\overline{\text{DRAK1}}$ output (DMAC)	TxD1 output (SCI)	TIOC2A I/O (TPU)
	PF5 I/O (port)	$\overline{\text{DACK1}}$ output (DMAC)	RxD1 input (SCI)	TIOC2B I/O (TPU)
	PF3 I/O (port)	$\overline{\text{DREQ0}}$ input (DMAC)	TIOC0A I/O (TPU)	—
	PF2 I/O (port)	$\overline{\text{DRAK0}}$ output (DMAC)	TIOC0C I/O (TPU)	—
	PF1 I/O (port)	$\overline{\text{DACK0}}$ output (DMAC)	TIOC0B I/O (TPU)	—

Table 17.7 Multiplex Pins (Port G)

Port	Function 1 (Related Module)	Function 2 (Related Module)	Function 3 (Related Module)	Function 4 (Related Module)
G	PG31 I/O (port)	RxD2 input (SCI)	—	—
	PG30 I/O (port)	TxD2 output (SCI)	—	—
	PG29 I/O (port)	SCK2 I/O (SCI)	—	—

Table 17.8 Multiplex Pins (Port H)

Port	Function 1 (Related Module)	Function 2 (Related Module)	Function 3 (Related Module)	Function 4 (Related Module)
H	PH1 input/output (port)	DA1 output (D/A)	—	—
	PH0 input/output (port)	DA0 output (D/A)	—	—

Table 17.9 Multiplex Pins (Port I)

Port	Function 1 (Related Module)	Function 2 (Related Module)	Function 3 (Related Module)	Function 4 (Related Module)
I	PI7 input (port)	AN7 input (A/D)	—	—
	PI6 input (port)	AN6 input (A/D)	—	—
	PI5 input (port)	AN5 input (A/D)	—	—
	PI4 input (port)	AN4 input (A/D)	—	—
	PI3 input (port)	AN3 input (A/D)	—	—
	PI2 input (port)	AN2 input (A/D)	—	—
	PI1 input (port)	AN1 input (A/D)	—	—
	PI0 input (port)	AN0 input (A/D)	—	—

Note: Switching to port I function 2 (AN7 to AN0 input) is performed automatically when the A/D converter is started, and switching to function 1 (port input) is performed automatically when A/D conversion ends.

Table 17.10 Pin Functions in Each Operating Mode

	Pin Name									
	On-Chip ROM Disabled						On-Chip ROM Enabled			
	MCU Mode 4		MCU Mode 3		MCU Mode 2		MCU Mode 1		Single-Chip Mode	
Pin No.	Initial Function	Functions Settable by PFC	Initial Function	Functions Settable by PFC	Initial Function	Functions Settable by PFC	Initial Function	Functions Settable by PFC	Initial Function	Functions Settable by PFC
17, 26, 39, 48, 58, 70, 79, 92, 105, 118, 126, 140	V_{CC}	V_{CC}	V_{CC}	V_{CC}	V_{CC}	V_{CC}	V_{CC}	V_{CC}	V_{CC}	V_{CC}
10, 13, 21, 25, 31, 38, 45, 54, 64, 76, 88, 89, 101, 110, 113, 124, 128, 133, 135, 147	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}
5, 160, 173	PV_{CC}	PV_{CC}	PV_{CC}	PV_{CC}	PV_{CC}	PV_{CC}	PV_{CC}	PV_{CC}	PV_{CC}	PV_{CC}
1, 166	PV_{SS}	PV_{SS}	PV_{SS}	PV_{SS}	PV_{SS}	PV_{SS}	PV_{SS}	PV_{SS}	PV_{SS}	PV_{SS}
129	$PLLV_{CC}$	$PLLV_{CC}$	$PLLV_{CC}$	$PLLV_{CC}$	$PLLV_{CC}$	$PLLV_{CC}$	$PLLV_{CC}$	$PLLV_{CC}$	$PLLV_{CC}$	$PLLV_{CC}$
132	$PLLV_{SS}$	$PLLV_{SS}$	$PLLV_{SS}$	$PLLV_{SS}$	$PLLV_{SS}$	$PLLV_{SS}$	$PLLV_{SS}$	$PLLV_{SS}$	$PLLV_{SS}$	$PLLV_{SS}$
130	PLLCAP1	PLLCAP1	PLLCAP1	PLLCAP1	PLLCAP1	PLLCAP1	PLLCAP1	PLLCAP1	PLLCAP1	PLLCAP1
131	PLLCAP2	PLLCAP2	PLLCAP2	PLLCAP2	PLLCAP2	PLLCAP2	PLLCAP2	PLLCAP2	PLLCAP2	PLLCAP2
159	AV_{CC}	AV_{CC}	AV_{CC}	AV_{CC}	AV_{CC}	AV_{CC}	AV_{CC}	AV_{CC}	AV_{CC}	AV_{CC}
148	AV_{SS}	AV_{SS}	AV_{SS}	AV_{SS}	AV_{SS}	AV_{SS}	AV_{SS}	AV_{SS}	AV_{SS}	AV_{SS}
114	EXTAL	EXTAL	EXTAL	EXTAL	EXTAL	EXTAL	EXTAL	EXTAL	EXTAL	EXTAL
112	XTAL	XTAL	XTAL	XTAL	XTAL	XTAL	XTAL	XTAL	XTAL	XTAL
127	CKIO	CKIO	CKIO	CKIO	CKIO	CKIO	CKIO	CKIO	CKIO	CKIO
134	CK	CK	CK	CK	CK	CK	CK	CK	CK	CK
123	$\overline{RES}$	$\overline{RES}$	$\overline{RES}$	$\overline{RES}$	$\overline{RES}$	$\overline{RES}$	$\overline{RES}$	$\overline{RES}$	$\overline{RES}$	$\overline{RES}$
146	$\overline{WDTOVF}$	$\overline{WDTOVF}$	$\overline{WDTOVF}$	$\overline{WDTOVF}$	$\overline{WDTOVF}$	$\overline{WDTOVF}$	$\overline{WDTOVF}$	$\overline{WDTOVF}$	$\overline{WDTOVF}$	$\overline{WDTOVF}$
125	$\overline{HSTBY}$	$\overline{HSTBY}$	$\overline{HSTBY}$	$\overline{HSTBY}$	$\overline{HSTBY}$	$\overline{HSTBY}$	$\overline{HSTBY}$	$\overline{HSTBY}$	$\overline{HSTBY}$	$\overline{HSTBY}$
121	MD5	MD5	MD5	MD5	MD5	MD5	MD5	MD5	MD5	MD5
119	MD4	MD4	MD4	MD4	MD4	MD4	MD4	MD4	MD4	MD4
117	MD3	MD3	MD3	MD3	MD3	MD3	MD3	MD3	MD3	MD3
116	MD2	MD2	MD2	MD2	MD2	MD2	MD2	MD2	MD2	MD2

	Pin Name									
	On-Chip ROM Disabled						On-Chip ROM Enabled			
	MCU Mode 4		MCU Mode 3		MCU Mode 2		MCU Mode 1		Single-Chip Mode	
Pin No.	Initial Function	Functions Settable by PFC	Initial Function	Functions Settable by PFC	Initial Function	Functions Settable by PFC	Initial Function	Functions Settable by PFC	Initial Function	Functions Settable by PFC
115	MD1	MD1	MD1	MD1	MD1	MD1	MD1	MD1	MD1	MD1
111	MD0	MD0	MD0	MD0	MD0	MD0	MD0	MD0	MD0	MD0
122	NMI	NMI	NMI	NMI	NMI	NMI	NMI	NMI	NMI	NMI
120	FWE	FWE	FWE	FWE	FWE	FWE	FWE	FWE	FWE	FWE
41	PA25	PA25/$\overline{\text{CS5}}$	PA25	PA25/$\overline{\text{CS5}}$	PA25	PA25/$\overline{\text{CS5}}$	PA25	PA25/$\overline{\text{CS5}}$	PA25	PA25
42	PA24	PA24/$\overline{\text{CS4}}$	PA24	PA24/$\overline{\text{CS4}}$	PA24	PA24/$\overline{\text{CS4}}$	PA24	PA24/$\overline{\text{CS4}}$	PA24	PA24
43	PA23	PA23/$\overline{\text{CS3}}$	PA23	PA23/$\overline{\text{CS3}}$	PA23	PA23/$\overline{\text{CS3}}$	PA23	PA23/$\overline{\text{CS3}}$	PA23	PA23
44	PA22	PA22/$\overline{\text{CS2}}$	PA22	PA22/$\overline{\text{CS2}}$	PA22	PA22/$\overline{\text{CS2}}$	PA22	PA22/$\overline{\text{CS2}}$	PA22	PA22
46	PA21	PA21/$\overline{\text{CS1}}$	PA21	PA21/$\overline{\text{CS1}}$	PA21	PA21/$\overline{\text{CS1}}$	PA21	PA21/$\overline{\text{CS1}}$	PA21	PA21
47	$\overline{\text{CS0}}$	$\overline{\text{CS0}}$	$\overline{\text{CS0}}$	$\overline{\text{CS0}}$	$\overline{\text{CS0}}$	$\overline{\text{CS0}}$	PA20	PA20/$\overline{\text{CS0}}$	PA20	PA20
142	$\overline{\text{BS}}$	$\overline{\text{BS}}$	$\overline{\text{BS}}$	$\overline{\text{BS}}$	$\overline{\text{BS}}$	$\overline{\text{BS}}$	PA19	PA19/$\overline{\text{BS}}$	PA19	PA19
49	$\overline{\text{RD}}$	$\overline{\text{RD}}$	$\overline{\text{RD}}$	$\overline{\text{RD}}$	$\overline{\text{RD}}$	$\overline{\text{RD}}$	PA18	PA18/$\overline{\text{RD}}$	PA18	PA18
50	PA17	PA17/$\overline{\text{WR}}$	PA17	PA17/$\overline{\text{WR}}$	PA17	PA17/$\overline{\text{WR}}$	PA17	PA17/$\overline{\text{WR}}$	PA17	PA17
51	$\overline{\text{WRHH}}$	$\overline{\text{WRHH}}$/ $\overline{\text{HHBS}}$/ TCLKC/ TIOC3A	$\overline{\text{WRHH}}$	$\overline{\text{WRHH}}$/ $\overline{\text{HHBS}}$/ TCLKC/ TIOC3A	$\overline{\text{WRHH}}$	$\overline{\text{WRHH}}$/ $\overline{\text{HHBS}}$/ TCLKC/ TIOC3A	PA16	PA16/ $\overline{\text{WRHH}}$/ $\overline{\text{HHBS}}$/ TCLKC/ TIOC3A	PA16	PA16/ TCLKC/ TIOC3A
52	$\overline{\text{WRHL}}$	$\overline{\text{WRHL}}$/ $\overline{\text{HLBS}}$/ TCLKD/ TIOC3B	$\overline{\text{WRHL}}$	$\overline{\text{WRHL}}$/ $\overline{\text{HLBS}}$/ TCLKD/ TIOC3B	$\overline{\text{WRHL}}$	$\overline{\text{WRHL}}$/ $\overline{\text{HLBS}}$/ TCLKD/ TIOC3B	PA15	PA15/ $\overline{\text{WRHL}}$/ $\overline{\text{HLBS}}$/ TCLKD/ TIOC3B	PA15	PA15/ TCLKD/ TIOC3B
53	$\overline{\text{WRLH}}$	$\overline{\text{WRLH}}$/ $\overline{\text{LHBS}}$	$\overline{\text{WRLH}}$	$\overline{\text{WRLH}}$/ $\overline{\text{LHBS}}$	$\overline{\text{WRLH}}$	$\overline{\text{WRLH}}$/ $\overline{\text{LHBS}}$	PA14	PA14/ $\overline{\text{WRLH}}$/ $\overline{\text{LHBS}}$	PA14	PA14
55	$\overline{\text{WRLL}}$	$\overline{\text{WRLL}}$/ $\overline{\text{LLBS}}$	$\overline{\text{WRLL}}$	$\overline{\text{WRLL}}$/ $\overline{\text{LLBS}}$	$\overline{\text{WRLL}}$	$\overline{\text{WRLL}}$/ $\overline{\text{LLBS}}$	PA13	PA13/ $\overline{\text{WRLL}}$/ $\overline{\text{LLBS}}$	PA13	PA13
56	PA12	PA12/ $\overline{\text{WAIT}}$	PA12	PA12/ $\overline{\text{WAIT}}$	PA12	PA12/ $\overline{\text{WAIT}}$	PA12	PA12/ $\overline{\text{WAIT}}$	PA12	PA12
57	PA9	PA9/$\overline{\text{RAS1}}$	PA9	PA9/$\overline{\text{RAS1}}$	PA9	PA9/$\overline{\text{RAS1}}$	PA9	PA9/$\overline{\text{RAS1}}$	PA9	PA9
59	PA8	PA8/$\overline{\text{RAS0}}$	PA8	PA8/$\overline{\text{RAS0}}$	PA8	PA8/$\overline{\text{RAS0}}$	PA8	PA8/$\overline{\text{RAS0}}$	PA8	PA8
136	PA1	PA1/$\overline{\text{OE1}}$	PA1	PA1/$\overline{\text{OE1}}$	PA1	PA1/$\overline{\text{OE1}}$	PA1	PA1/$\overline{\text{OE1}}$	PA1	PA1
137	PA0	PA0/$\overline{\text{OE0}}$	PA0	PA0/$\overline{\text{OE0}}$	PA0	PA0/$\overline{\text{OE0}}$	PA0	PA0/$\overline{\text{OE0}}$	PA0	PA0

	Pin Name									
	On-Chip ROM Disabled						On-Chip ROM Enabled			
	MCU Mode 4		MCU Mode 3		MCU Mode 2		MCU Mode 1		Single-Chip Mode	
Pin No.	Initial Function	Functions Settable by PFC	Initial Function	Functions Settable by PFC	Initial Function	Functions Settable by PFC	Initial Function	Functions Settable by PFC	Initial Function	Functions Settable by PFC
60	PB23	PB23/ $\overline{\text{CASHH1}}$/ TxD1/ $\overline{\text{TEND0}}$	PB23	PB23/ $\overline{\text{CASHH1}}$/ TxD1/ $\overline{\text{TEND0}}$	PB23	PB23/ $\overline{\text{CASHH1}}$/ TxD1/ $\overline{\text{TEND0}}$	PB23	PB23/ $\overline{\text{CASHH1}}$/ TxD1/ $\overline{\text{TEND0}}$	PB23	PB23/ TxD1
61	PB22	PB22/ $\overline{\text{CASHL1}}$/ RxD1/ $\overline{\text{TEND1}}$	PB22	PB22/ $\overline{\text{CASHL1}}$/ RxD1/ $\overline{\text{TEND1}}$	PB22	PB22/ $\overline{\text{CASHL1}}$/ RxD1/ $\overline{\text{TEND1}}$	PB22	PB22/ $\overline{\text{CASHL1}}$/ RxD1/ $\overline{\text{TEND1}}$	PB22	PB22/ RxD1
62	PB21	PB21/ $\overline{\text{CASLH1}}$	PB21	PB21/ $\overline{\text{CASLH1}}$	PB21	PB21/ $\overline{\text{CASLH1}}$	PB21	PB21/ $\overline{\text{CASLH1}}$	PB21	PB21
63	PB20	PB20/ $\overline{\text{CASLL1}}$	PB20	PB20/ $\overline{\text{CASLL1}}$	PB20	PB20/ $\overline{\text{CASLL1}}$	PB20	PB20/ $\overline{\text{CASLL1}}$	PB20	PB20
65	PB19	PB19/ $\overline{\text{CASHH0}}$/ TxD0	PB19	PB19/ $\overline{\text{CASHH0}}$/ TxD0	PB19	PB19/ $\overline{\text{CASHH0}}$/ TxD0	PB19	PB19/ $\overline{\text{CASHH0}}$/ TxD0	PB19	PB19/ TxD0
66	PB18	PB18/ $\overline{\text{CASHL0}}$/ RxD0	PB18	PB18/ $\overline{\text{CASHL0}}$/ RxD0	PB18	PB18/ $\overline{\text{CASHL0}}$/ RxD0	PB18	PB18/ $\overline{\text{CASHL0}}$/ RxD0	PB18	PB18/ RxD0
67	PB17	PB17/ $\overline{\text{CASLH0}}$	PB17	PB17/ $\overline{\text{CASLH0}}$	PB17	PB17/ $\overline{\text{CASLH0}}$	PB17	PB17/ $\overline{\text{CASLH0}}$	PB17	PB17
68	PB16	PB16/ $\overline{\text{CASLL0}}$	PB16	PB16/ $\overline{\text{CASLL0}}$	PB16	PB16/ $\overline{\text{CASLL0}}$	PB16	PB16/ $\overline{\text{CASLL0}}$	PB16	PB16
69	PB13	PB13/ RDWR	PB13	PB13/ RDWR	PB13	PB13/ RDWR	PB13	PB13/ RDWR	PB13	PB13
164	PB7	PB7/ $\overline{\text{BACK}}$	PB7	PB7 $\overline{\text{/BACK}}$	PB7	PB7/ $\overline{\text{BACK}}$	PB7	PB7/ $\overline{\text{BACK}}$	PB7	PB7
165	PB6	PB6/ $\overline{\text{BREQ}}$	PB6	PB6/ $\overline{\text{BREQ}}$	PB6	PB6/ $\overline{\text{BREQ}}$	PB6	PB6/ $\overline{\text{BREQ}}$	PB6	PB6
40	A25	A25/ TIOC3B/ TCLKD	A25	A25/ TIOC3B/ TCLKD	A25	A25/ TIOC3B/ TCLKD	PC25	PC25/A25/ TIOC3B/ TCLKD	PC25	PC25/ TIOC3B/ TCLKD
37	A24	A24/ TIOC3A/ TCLKC	A24	A24/ TIOC3A/ TCLKC	A24	A24/ TIOC3A/ TCLKC	PC24	PC24/A24/ TIOC3A/ TCLKC	PC24	PC24/ TIOC3A/ TCLKC
36	A23	A23/ TIOC1B/ TCLKB	A23	A23/ TIOC1B/ TCLKB	A23	A23/ TIOC1B/ TCLKB	PC23	PC23/A23/ TIOC1B/ TCLKB	PC23	PC23/ TIOC1B/ TCLKB
35	A22	A22/ TIOC1A/ TCLKA	A22	A22/ TIOC1A/ TCLKA	A22	A22/ TIOC1A/ TCLKA	PC22	PC22/A22/ TIOC1A/ TCLKA	PC22	PC22/ TIOC1A/ TCLKA

	Pin Name									
	On-Chip ROM Disabled						On-Chip ROM Enabled			
	MCU Mode 4		MCU Mode 3		MCU Mode 2		MCU Mode 1		Single-Chip Mode	
Pin No.	**Initial Function**	**Functions Settable by PFC**	**Initial Function**	**Functions Settable by PFC**	**Initial Function**	**Functions Settable by PFC**	**Initial Function**	**Functions Settable by PFC**	**Initial Function**	**Functions Settable by PFC**
34	A21	A21/ TIOC5B	A21	A21/ TIOC5B	A21	A21/ TIOC5B	PC21	PC21/A21/ TIOC5B	PC21	PC21/ TIOC5B
33	A20	A20/ TIOC5A	A20	A20/ TIOC5A	A20	A20/ TIOC5A	PC20	PC20/A20/ TIOC5A	PC20	PC20/ TIOC5A
32	A19	A19/ TIOC4B	A19	A19/ TIOC4B	A19	A19/ TIOC4B	PC19	PC19/A19/ TIOC4B	PC19	PC19/ TIOC4B
30	A18	A18/ TIOC4A	A18	A18/ TIOC4A	A18	A18/ TIOC4A	PC18	PC18/A18/ TIOC4A	PC18	PC18/ TIOC4A
29	A17	A17/ TIOC3B	A17	A17/ TIOC3B	A17	A17/ TIOC3B	PC17	PC17/A17/ TIOC3B	PC17	PC17/ TIOC3B
28	A16	A16/ TIOC3A	A16	A16/ TIOC3A	A16	A16/ TIOC3A	PC16	PC16/A16/ TIOC3A	PC16	PC16/ TIOC3A
27	A15	A15/ TIOC3D	A15	A15/ TIOC3D	A15	A15/ TIOC3D	PC15	PC15/A15/ TIOC3D	PC15	PC15/ TIOC3D
24	A14	A14/ TIOC3C	A14	A14/ TIOC3C	A14	A14/ TIOC3C	PC14	PC14/A14/ TIOC3C	PC14	PC14/ TIOC3C
23	A13	A13	A13	A13	A13	A13	PC13	PC13/A13	PC13	PC13
22	A12	A12	A12	A12	A12	A12	PC12	PC12/A12	PC12	PC12
20	A11	A11	A11	A11	A11	A11	PC11	PC11/A11	PC11	PC11
19	A10	A10	A10	A10	A10	A10	PC10	PC10/A10	PC10	PC10
18	A9	A9	A9	A9	A9	A9	PC9	PC9/A9	PC9	PC9
16	A8	A8	A8	A8	A8	A8	PC8	PC8/A8	PC8	PC8
15	A7	A7	A7	A7	A7	A7	PC7	PC7/A7	PC7	PC7
14	A6	A6	A6	A6	A6	A6	PC6	PC6/A6	PC6	PC6
12	A5	A5	A5	A5	A5	A5	PC5	PC5/A5	PC5	PC5
11	A4	A4	A4	A4	A4	A4	PC4	PC4/A4	PC4	PC4
9	A3	A3	A3	A3	A3	A3	PC3	PC3/A3	PC3	PC3
8	A2	A2	A2	A2	A2	A2	PC2	PC2/A2	PC2	PC2
7	A1	A1	A1	A1	A1	A1	PC1	PC1/A1	PC1	PC1
6	A0	A0	A0	A0	A0	A0	PC0	PC0/A0	PC0	PC0
71	PD31	PD31/D31/ RxD2/ TIOC5A	PD31	PD31/D31/ RxD2/ TIOC5A	D31	D31/ RxD2/ TIOC5A	PD31	PD31/D31/ RxD2/ TIOC5A	PD31	PD31/ RxD2/ TIOC5A
72	PD30	PD30/D30/ TxD2/ TIOC4B	PD30	PD30/D30/ TxD2/ TIOC4B	D30	D30/ TxD2/ TIOC4B	PD30	PD30/D30/ TxD2/ TIOC4B	PD30	PD30/ TxD2/ TIOC4B

	Pin Name									
	On-Chip ROM Disabled						On-Chip ROM Enabled			
	MCU Mode 4		MCU Mode 3		MCU Mode 2		MCU Mode 1		Single-Chip Mode	
Pin No.	Initial Function	Functions Settable by PFC	Initial Function	Functions Settable by PFC	Initial Function	Functions Settable by PFC	Initial Function	Functions Settable by PFC	Initial Function	Functions Settable by PFC
73	PD29	PD29/D29/SCK2/TIOC4A	PD29	PD29/D29/SCK2/TIOC4A	D29	D29/SCK2/TIOC4A	PD29	PD29/D29/SCK2/TIOC4A	PD29	PD29/SCK2/TIOC4A
74	PD28	PD28/D28/TCLKB/TIOC3D	PD28	PD28/D28/TCLKB/TIOC3D	D28	D28/TCLKB/TIOC3D	PD28	PD28/D28/TCLKB/TIOC3D	PD28	PD28/TCLKB/TIOC3D
75	PD27	PD27/D27/TCLKA/TIOC3C	PD27	PD27/D27/TCLKA/TIOC3C	D27	D27/TCLKA/TIOC3C	PD27	PD27/D27/TCLKA/TIOC3C	PD27	PD27/TCLKA/TIOC3C
77	PD26	PD26/D26/PWOB	PD26	PD26/D26/PWOB	D26	D26/PWOB	PD26	PD26/D26/PWOB	PD26	PD26/PWOB
78	PD25	PD25/D25/PVOB	PD25	PD25/D25/PVOB	D25	D25/PVOB	PD25	PD25/D25/PVOB	PD25	PD25/PVOB
80	PD24	PD24/D24/PUOB	PD24	PD24/D24/PUOB	D24	D24/PUOB	PD24	PD24/D24/PUOB	PD24	PD24/PUOB
81	PD23	PD23/D23/PCO/PCI/SCK1	PD23	PD23/D23/PCO/PCI/SCK1	D23	D23/PCO/PCI/SCK1	PD23	PD23/D23/PCO/PCI/SCK1	PD23	PD23/PCO/PCI/SCK1
82	PD22	PD22/D22/PWOA/SCK0	PD22	PD22/D22/PWOA/SCK0	D22	D22/PWOA/SCK0	PD22	PD22/D22/PWOA/SCK0	PD22	PD22/PWOA/SCK0
83	PD21	PD21/D21/PVOA/$\overline{\text{IRQ7}}$	PD21	PD21/D21/PVOA/$\overline{\text{IRQ7}}$	D21	D21/PVOA/$\overline{\text{IRQ7}}$	PD21	PD21/D21/PVOA/$\overline{\text{IRQ7}}$	PD21	PD21/PVOA/$\overline{\text{IRQ7}}$
84	PD20	PD20/D20/PUOA/$\overline{\text{IRQ6}}$	PD20	PD20/D20/PUOA/$\overline{\text{IRQ6}}$	D20	D20/PUOA/$\overline{\text{IRQ6}}$	PD20	PD20/D20/PUOA/$\overline{\text{IRQ6}}$	PD20	PD20/PUOA/$\overline{\text{IRQ6}}$
85	PD19	PD19/D19/$\overline{\text{POE3}}$/$\overline{\text{IRQ5}}$	PD19	PD19/D19/$\overline{\text{POE3}}$/$\overline{\text{IRQ5}}$	D19	D19/$\overline{\text{POE3}}$/$\overline{\text{IRQ5}}$	PD19	PD19/D19/$\overline{\text{POE3}}$/$\overline{\text{IRQ5}}$	PD19	PD19/$\overline{\text{POE3}}$/$\overline{\text{IRQ5}}$
86	PD18	PD18/D18/$\overline{\text{POE2}}$/$\overline{\text{IRQ4}}$	PD18	PD18/D18/$\overline{\text{POE2}}$/$\overline{\text{IRQ4}}$	D18	D18/$\overline{\text{POE2}}$/$\overline{\text{IRQ4}}$	PD18	PD18/D18/$\overline{\text{POE2}}$/$\overline{\text{IRQ4}}$	PD18	PD18/$\overline{\text{POE2}}$/$\overline{\text{IRQ4}}$
87	PD17	PD17/D17/$\overline{\text{POE1}}$/$\overline{\text{ADTRG}}$	PD17	PD17/D17/$\overline{\text{POE1}}$/$\overline{\text{ADTRG}}$	D17	D17/$\overline{\text{POE1}}$/$\overline{\text{ADTRG}}$	PD17	PD17/D17/$\overline{\text{POE1}}$/$\overline{\text{ADTRG}}$	PD17	PD17/$\overline{\text{POE1}}$/$\overline{\text{ADTRG}}$
90	PD16	PD16/D16/$\overline{\text{POE0}}$	PD16	PD16/D16/$\overline{\text{POE0}}$	D16	D16/$\overline{\text{POE0}}$	PD16	PD16/D16/$\overline{\text{POE0}}$	PD16	PD16/$\overline{\text{POE0}}$
91	PD15	D15/TIOC5B	D15	D15/TIOC5B	D15	D15/TIOC5B	PD15	PD15/D15/TIOC5B	PD15	PD15/TIOC5B

	Pin Name									
	On-Chip ROM Disabled						On-Chip ROM Enabled			
	MCU Mode 4		MCU Mode 3		MCU Mode 2		MCU Mode 1		Single-Chip Mode	
Pin No.	Initial Function	Functions Settable by PFC	Initial Function	Functions Settable by PFC	Initial Function	Functions Settable by PFC	Initial Function	Functions Settable by PFC	Initial Function	Functions Settable by PFC
93	PD14	D14/TIOC5A	D14	D14/TIOC5A	D14	D14/TIOC5A	PD14	PD14/D14/TIOC5A	PD14	PD14/TIOC5A
94	PD13	D13/TIOC4B	D13	D13/TIOC4B	D13	D13/TIOC4B	PD13	PD13/D13/TIOC4B	PD13	PD13/TIOC4B
95	PD12	D12/TIOC4A	D12	D12/TIOC4A	D12	D12/TIOC4A	PD12	PD12/D12/TIOC4A	PD12	PD12/TIOC4A
96	PD11	D11/TIOC2B	D11	D11/TIOC2B	D11	D11/TIOC2B	PD11	PD11/D11/TIOC2B	PD11	PD11/TIOC2B
97	PD10	D10/TIOC2A	D10	D10/TIOC2A	D10	D10/TIOC2A	PD10	PD10/D10/TIOC2A	PD10	PD10/TIOC2A
98	PD9	D9/TIOC1B	D9	D9/TIOC1B	D9	D9/TIOC1B	PD9	PD9/D9/TIOC1B	PD9	PD9/TIOC1B
99	PD8	D8/TIOC1A	D8	D8/TIOC1A	D8	D8/TIOC1A	PD8	PD8/D8/TIOC1A	PD8	PD8/TIOC1A
100	D7	D7	D7	D7	D7	D7	PD7	PD7/D7	PD7	PD7
102	D6	D6	D6	D6	D6	D6	PD6	PD6/D6	PD6	PD6
103	D5	D5	D5	D5	D5	D5	PD5	PD5/D5	PD5	PD5
104	D4	D4	D4	D4	D4	D4	PD4	PD4/D4	PD4	PD4
106	D3	D3	D3	D3	D3	D3	PD3	PD3/D3	PD3	PD3
107	D2	D2	D2	D2	D2	D2	PD2	PD2/D2	PD2	PD2
108	D1	D1	D1	D1	D1	D1	PD1	PD1/D1	PD1	PD1
109	D0	D0	D0	D0	D0	D0	PD0	PD0/D0	PD0	PD0
167	PE23	PE23/$\overline{\text{IRQ7}}$/PWOB	PE23	PE23/$\overline{\text{IRQ7}}$/PWOB	PE23	PE23/$\overline{\text{IRQ7}}$/PWOB	PE23	PE23/$\overline{\text{IRQ7}}$/PWOB	PE23	PE23/$\overline{\text{IRQ7}}$/PWOB
168	PE22	PE22/$\overline{\text{IRQ6}}$/PVOB	PE22	PE22/$\overline{\text{IRQ6}}$/PVOB	PE22	PE22/$\overline{\text{IRQ6}}$/PVOB	PE22	PE22/$\overline{\text{IRQ6}}$/PVOB	PE22	PE22/$\overline{\text{IRQ6}}$/PVOB
169	PE21	PE21/$\overline{\text{IRQ5}}$/PUOB	PE21	PE21/$\overline{\text{IRQ5}}$/PUOB	PE21	PE21/$\overline{\text{IRQ5}}$/PUOB	PE21	PE21/$\overline{\text{IRQ5}}$/PUOB	PE21	PE21/$\overline{\text{IRQ5}}$/PUOB
170	PE20	PE20/$\overline{\text{IRQ4}}$/PCO/PCI	PE20	PE20/$\overline{\text{IRQ4}}$/PCO/PCI	PE20	PE20/$\overline{\text{IRQ4}}$/PCO/PCI	PE20	PE20/$\overline{\text{IRQ4}}$/PCO/PCI	PE20	PE20/$\overline{\text{IRQ4}}$/PCO/PCI
171	PE19	PE19/$\overline{\text{IRQ3}}$/PWOA	PE19	PE19/$\overline{\text{IRQ3}}$/PWOA	PE19	PE19/$\overline{\text{IRQ3}}$/PWOA	PE19	PE19/$\overline{\text{IRQ3}}$/PWOA	PE19	PE19/$\overline{\text{IRQ3}}$/PWOA

	Pin Name									
	On-Chip ROM Disabled						On-Chip ROM Enabled			
	MCU Mode 4		MCU Mode 3		MCU Mode 2		MCU Mode 1		Single-Chip Mode	
Pin No.	Initial Function	Functions Settable by PFC	Initial Function	Functions Settable by PFC	Initial Function	Functions Settable by PFC	Initial Function	Functions Settable by PFC	Initial Function	Functions Settable by PFC
172	PE18	PE18/ $\overline{\text{IRQ2}}$/ PVOA	PE18	PE18/ $\overline{\text{IRQ2}}$/ PVOA	PE18	PE18/ $\overline{\text{IRQ2}}$/ PVOA	PE18	PE18/ IRQ2/ PVOA	PE18	PE18/ $\overline{\text{IRQ2}}$/ PVOA
174	PE17	PE17/ $\overline{\text{IRQ1}}$/ PUOA/ SCK0	PE17	PE17/ $\overline{\text{IRQ1}}$/ PUOA/ SCK0	PE17	PE17/ $\overline{\text{IRQ1}}$/ PUOA/ SCK0	PE17	PE17/ $\overline{\text{IRQ1}}$/ PUOA/ SCK0	PE17	PE17/ $\overline{\text{IRQ1}}$/ PUOA/ SCK0
175	PE16	PE16/ $\overline{\text{IRQ0}}$/ SCK1/$\overline{\text{AH}}$	PE16	PE16/ $\overline{\text{IRQ0}}$/ SCK1/$\overline{\text{AH}}$	PE16	PE16/ $\overline{\text{IRQ0}}$/ SCK1/$\overline{\text{AH}}$	PE16	PE16/ $\overline{\text{IRQ0}}$/ SCK1/$\overline{\text{AH}}$	PE16	PE16/ $\overline{\text{IRQ0}}$/ SCK1
176	PE15	PE15/ $\overline{\text{IRQ7}}$	PE15	PE15/ $\overline{\text{IRQ7}}$	PE15	PE15/ $\overline{\text{IRQ7}}$	PE15	PE15/ $\overline{\text{IRQ7}}$	PE15	PE15/ $\overline{\text{IRQ7}}$
2	PE14	PE14/ $\overline{\text{IRQ6}}$	PE14	PE14/ $\overline{\text{IRQ6}}$	PE14	PE14/ $\overline{\text{IRQ6}}$	PE14	PE14/ $\overline{\text{IRQ6}}$	PE14	PE14/ $\overline{\text{IRQ6}}$
3	PE13	PE13/ $\overline{\text{IRQ5}}$	PE13	PE13/ $\overline{\text{IRQ5}}$	PE13	PE13/ $\overline{\text{IRQ5}}$	PE13	PE13/ $\overline{\text{IRQ5}}$	PE13	PE13/ $\overline{\text{IRQ5}}$
4	PE12	PE12/ $\overline{\text{IRQ4}}$	PE12	PE12/ $\overline{\text{IRQ4}}$	PE12	PE12/ $\overline{\text{IRQ4}}$	PE12	PE12/ $\overline{\text{IRQ4}}$	PE12	PE12/ $\overline{\text{IRQ4}}$
145	PF7	PF7/ $\overline{\text{DREQ1}}$/ $\overline{\text{IRQOUT}}$/ TIOC0D	PF7	PF7/ $\overline{\text{DREQ1}}$/ $\overline{\text{IRQOUT}}$/ TIOC0D	PF7	PF7/ $\overline{\text{DREQ1}}$/ $\overline{\text{IRQOUT}}$/ TIOC0D	PF7	PF7/ $\overline{\text{DREQ1}}$/ $\overline{\text{IRQOUT}}$/ TIOC0D	PF7	PF7/ $\overline{\text{IRQOUT}}$/ TIOC0D
144	PF6	PF6/ $\overline{\text{DRAK1}}$/ TxD1/ TIOC2A	PF6	PF6/ $\overline{\text{DRAK1}}$/ TxD1/ TIOC2A	PF6	PF6/ $\overline{\text{DRAK1}}$/ TxD1/ TIOC2A	PF6	PF6/ $\overline{\text{DRAK1}}$/ TxD1/ TIOC2A	PF6	PF6/ TxD1/ TIOC2A
143	PF5	PF5/ $\overline{\text{DACK1}}$/ RxD1/ TIOC2B	PF5	PF5/ $\overline{\text{DACK1}}$/ RxD1/ TIOC2B	PF5	PF5/ $\overline{\text{DACK1}}$/ RxD1/ TIOC2B	PF5	PF5/ $\overline{\text{DACK1}}$/ RxD1/ TIOC2B	PF5	PF5/ RxD1/ TIOC2B
138	PF3	PF3/ $\overline{\text{DREQ0}}$/ TIOC0A	PF3	PF3/ $\overline{\text{DREQ0}}$/ TIOC0A	PF3	PF3/ $\overline{\text{DREQ0}}$/ TIOC0A	PF3	PF3/ $\overline{\text{DREQ0}}$/ TIOC0A	PF3	PF3/ TIOC0A
139	PF2	PF2/ $\overline{\text{DRAK0}}$/ TIOC0C	PF2	PF2/ $\overline{\text{DRAK0}}$/ TIOC0C	PF2	PF2/ $\overline{\text{DRAK0}}$/ TIOC0C	PF2	PF2/ $\overline{\text{DRAK0}}$/ TIOC0C	PF2	PF2/ TIOC0C
141	PF1	PF1/ $\overline{\text{DACK0}}$/ TIOC0B	PF1	PF1/ $\overline{\text{DACK0}}$/ TIOC0B	PF1	PF1/ $\overline{\text{DACK0}}$/ TIOC0B	PF1	PF1/ $\overline{\text{DACK0}}$/ TIOC0B	PF1	PF1/ TIOC0B
161	PG31	PG31/ RxD2	PG31	PG31/ RxD2	PG31	PG31/ RxD2	PG31	PG31/ RxD2	PG31	PG31/ RxD2

	Pin Name									
	On-Chip ROM Disabled						On-Chip ROM Enabled			
	MCU Mode 4		MCU Mode 3		MCU Mode 2		MCU Mode 1		Single-Chip Mode	
Pin No.	Initial Function	Functions Settable by PFC	Initial Function	Functions Settable by PFC	Initial Function	Functions Settable by PFC	Initial Function	Functions Settable by PFC	Initial Function	Functions Settable by PFC
162	PG30	PG30/ TxD2	PG30	PG30/ TxD2	PG30	PG30/ TxD2	PG30	PG30/ TxD2	PG30	PG30/ TxD2
163	PG29	PG29/ SCK2	PG29	PG29/ SCK2	PG29	PG29/ SCK2	PG29	PG29/ SCK2	PG29	PG29/ SCK2
150	PH1	PH1/DA1	PH1	PH1/DA1	PH1	PH1/DA1	PH1	PH1/DA1	PH1	PH1/DA1
149	PH0	PH0/DA0	PH0	PH0/DA0	PH0	PH0/DA0	PH0	PH0/DA0	PH0	PH0/DA0
158*	PI7	PI7/AN7	PI7	PI7/AN7	PI7	PI7/AN7	PI7	PI7/AN7	PI7	PI7/AN7
157*	PI6	PI6/AN6	PI6	PI6/AN6	PI6	PI6/AN6	PI6	PI6/AN6	PI6	PI6/AN6
156*	PI5	PI5/AN5	PI5	PI5/AN5	PI5	PI5/AN5	PI5	PI5/AN5	PI5	PI5/AN5
155*	PI4	PI4/AN4	PI4	PI4/AN4	PI4	PI4/AN4	PI4	PI4/AN4	PI4	PI4/AN4
154*	PI3	PI3/AN3	PI3	PI3/AN3	PI3	PI3/AN3	PI3	PI3/AN3	PI3	PI3/AN3
153*	PI2	PI2/AN2	PI2	PI2/AN2	PI2	PI2/AN2	PI2	PI2/AN2	PI2	PI2/AN2
152*	PI1	PI1/AN1	PI1	PI1/AN1	PI1	PI1/AN1	PI1	PI1/AN1	PI1	PI1/AN1
151*	PI0	PI0/AN0	PI0	PI0/AN0	PI0	PI0/AN0	PI0	PI0/AN0	PI0	PI0/AN0

Note: * Since switching to the port I ANn (A/D converter analog input) function is performed automatically when A/D conversion is started, and the PIn (port input) function is restored when A/D conversion ends, port I has no register for PFC settings.

17.2 Register Configuration

PFC registers are listed in table 17.11.

Table 17.11 PFC Registers

Name	Abbreviation	R/W	Initial Value	Address	Access Size
Port A IO register H	PAIORH	R/(W)	H'0000	H'FFFF1204	8, 16, 32
Port A IO register L	PAIORL	R/(W)	H'0000	H'FFFF1206	8, 16, 32
Port A control register H1	PACRH1	R/(W)	H'0000	H'FFFF1208	8, 16, 32
Port A control register H2	PACRH2	R/(W)	H'0000	H'FFFF120A	8, 16, 32
Port A control register L1	PACRL1	R/(W)	H'0000	H'FFFF120C	8, 16, 32
Port A control register L2	PACRL2	R/(W)	H'0000	H'FFFF120E	8, 16, 32
Port B IO register H	PBIORH	R/(W)	H'0000	H'FFFF1214	8, 16, 32
Port B IO register L	PBIORL	R/(W)	H'0000	H'FFFF1216	8, 16, 32
Port B control register H2	PBCRH2	R/(W)	H'0000	H'FFFF121A	8, 16, 32
Port B control register L1	PBCRL1	R/(W)	H'0000	H'FFFF121C	8, 16, 32
Port B control register L2	PBCRL2	R/(W)	H'0000	H'FFFF121E	8, 16, 32
Port C IO register H	PCIORH	R/(W)	H'0000	H'FFFF1224	8, 16, 32
Port C IO register L	PCIORL	R/(W)	H'0000	H'FFFF1226	8, 16, 32
Port C control register H1	PCCRH1	R/(W)	H'0000	H'FFFF1228	8, 16, 32
Port C control register H2	PCCRH2	R/(W)	H'0000	H'FFFF122A	8, 16, 32
Port C control register L1	PCCRL1	R/(W)	H'0000	H'FFFF122C	8, 16, 32
Port C control register L2	PCCRL2	R/(W)	H'0000	H'FFFF122E	8, 16, 32
Port D IO register H	PDIORH	R/(W)	H'0000	H'FFFF1234	8, 16, 32
Port D IO register L	PDIORL	R/(W)	H'0000	H'FFFF1236	8, 16, 32
Port D control register H1	PDCRH1	R/(W)	H'0000	H'FFFF1238	8, 16, 32
Port D control register H2	PDCRH2	R/(W)	H'0000	H'FFFF123A	8, 16, 32
Port D control register L1	PDCRL1	R/(W)	H'0000	H'FFFF123C	8, 16, 32
Port D control register L2	PDCRL2	R/(W)	H'0000	H'FFFF123E	8, 16, 32
Port E IO register H	PEIORH	R/(W)	H'0000	H'FFFF1244	8, 16, 32
Port E IO register L	PEIORL	R/(W)	H'0000	H'FFFF1246	8, 16, 32
Port E control register H2	PECRH2	R/(W)	H'0000	H'FFFF124A	8, 16, 32
Port E control register L	PECRL	R/(W)	H'0000	H'FFFF124C	8, 16, 32

Name	Abbreviation	R/W	Initial Value	Address	Access Size
Port F IO register L	PFIORL	R/(W)	H'0000	H'FFFF1266	8, 16, 32
Port F control register L2	PFCRL2	R/(W)	H'0000	H'FFFF126E	8, 16, 32
Port G IO register	PGIOR	R/(W)	H'0000	H'FFFF1274	8, 16, 32
Port G control register H1	PGCRH1	R/(W)	H'0000	H'FFFF1278	8, 16, 32
Port H IO register	PHIOR	R/(W)	H'0000	H'FFFF1286	8, 16, 32
Port H control register	PHCR	R/(W)	H'0000	H'FFFF128E	8, 16, 32
Function control register	FCR	R/(W)	H'0000	H'FFFF1250	8, 16, 32

17.3 Register Descriptions

17.3.1 Port A IO Register H (PAIORH)

Bit:	15	14	13	12	11	10	9	8
	—	—	—	—	—	—	PA25IOR	PA24IOR
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
	PA23IOR	PA22IOR	PA21IOR	PA20IOR	PA19IOR	PA18IOR	PA17IOR	PA16IOR
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

Port A IO register H (PAIORH) is a 16-bit readable/writable register that selects the input/output direction of pins in port A. Bits PA25IOR to PA16IOR correspond to pins PA25/$\overline{\text{CS5}}$ to PA16/$\overline{\text{WRHH}}$/$\overline{\text{HHBS}}$/TCLKC/TIOC3A. PAIORH is enabled when port A pins function as general input/output pins (PA25 to PA16) or PA16 functions as a TPU TIOC pin, and disabled otherwise.

When port A pins function as PA25 to PA16, or PA16 functions as a TPU TIOC pin, a pin becomes an output when the corresponding bit in PAIORH is set to 1, and an input when the bit is cleared to 0.

PAIORH is initialized to H'0000 by an external power-on reset, but is not initialized by a WDT reset, in standby mode, or in sleep mode.

17.3.2 Port A IO Register L (PAIORL)

Bit:	15	14	13	12	11	10	9	8
	PA15IOR	PA14IOR	PA13IOR	PA12IOR	—	—	PA9IOR	PA8IOR
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R	R	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
	—	—	—	–	—	—	PA1IOR	PA0IOR
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W

Port A IO register L (PAIORL) is a 16-bit readable/writable register that selects the input/output direction of pins in port A. Bits PA15IOR to PA0IOR correspond to pins PA15/$\overline{\text{WRHL}}$/$\overline{\text{HLBS}}$/TCLKD/TIOC3B to PA0/$\overline{\text{OE0}}$. PAIORL is enabled when port A pins function as general input/output pins (PA15 to PA0) or a TPU TIOC pin, and disabled otherwise.

When port A pins function as PA15 to PA0, or PA15 functions as a TPU TIOC pin, a pin becomes an output when the corresponding bit in PAIORL is set to 1, and an input when the bit is cleared to 0.

PAIORL is initialized to H'0000 by an external power-on reset, but is not initialized by a WDT reset, in standby mode, or in sleep mode.

17.3.3 Port A Control Registers H1 and H2 (PACRH1, PACRH2)

Port A control registers H1 and H2 (PACRH1, PACRH2) are 16-bit readable/writable registers that select the functions of the upper 16 multiplex pins in port A.

PACRH1 selects the functions of port A pins PA25/$\overline{\text{CS5}}$ to PA24/$\overline{\text{CS4}}$, and PACRH2 selects the functions of port A pins PA23/$\overline{\text{CS3}}$ to PA16/$\overline{\text{WRHH}}$/$\overline{\text{HHBS}}$/TCLKC/TIOC3A.

Port A includes bus control signals ($\overline{\text{CS0}}$ to $\overline{\text{CS5}}$, $\overline{\text{BS}}$, $\overline{\text{RD}}$, $\overline{\text{WR}}$, $\overline{\text{WRHH}}$, and $\overline{\text{HHBS}}$), but register settings relating to the selection of these pin functions may not be valid in all operating modes. For details, see table 17.10, Pin Functions in Each Operating Mode.

PACRH1 and PACRH2 are initialized to H'0000 by an external power-on reset, but are not initialized by a WDT reset, in standby mode, or in sleep mode.

Port A Control Register H1 (PACRH1)

Bit:	15	14	13	12	11	10	9	8
	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	—	PA25MD	—	PA24MD
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R	R/W

Bits 15 to 3—Reserved: These bits are always read as 0 and should only be written with 0.

Bit 2—PA25 Mode (PA25MD): Selects the function of the PA25/$\overline{\text{CS5}}$ pin.

Bit 2: PA25MD	Description	
0	General input/output (PA25)	(Initial value)
1	Chip select output ($\overline{\text{CS5}}$) (PA25 in single-chip mode)	

Bit 1—Reserved: This bit is always read as 0 and should only be written with 0.

Bit 0—PA24 Mode (PA24MD): Selects the function of the PA24/$\overline{\text{CS4}}$ pin.

Bit 0: PA24MD	Description	
0	General input/output (PA24)	(Initial value)
1	Chip select output ($\overline{\text{CS4}}$) (PA24 in single-chip mode)	

Port A Control Register H2 (PACRH2)

Bit:	15	14	13	12	11	10	9	8
	—	PA23 MD	—	PA22 MD	—	PA21 MD	—	PA20 MD
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R/W	R	R/W	R	R/W	R	R/W

Bit:	7	6	5	4	3	2	1	0
	—	PA19 MD	—	PA18 MD	—	PA17 MD	PA16 MD1	PA16 MD0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R/W	R	R/W	R	R/W	R/W	R/W

Bit 15—Reserved: This bit is always read as 0 and should only be written with 0.

Bit 14—PA23 Mode (PA23MD): Selects the function of the PA23/$\overline{\text{CS3}}$ pin.

Bit 14: PA23MD	Description	
0	General input/output (PA23)	(Initial value)
1	Chip select output ($\overline{\text{CS3}}$) (PA23 in single-chip mode)	

Bit 13—Reserved: This bit is always read as 0 and should only be written with 0.

Bit 12—PA22 Mode (PA22MD): Selects the function of the PA22/$\overline{\text{CS2}}$ pin.

Bit 12: PA22MD	Description	
0	General input/output (PA22)	(Initial value)
1	Chip select output ($\overline{\text{CS2}}$) (PA22 in single-chip mode)	

Bit 11—Reserved: This bit is always read as 0 and should only be written with 0.

Bit 10—PA21 Mode (PA21MD): Selects the function of the PA21/$\overline{\text{CS1}}$ pin.

Bit 10: PA21MD	Description	
0	General input/output (PA21)	(Initial value)
1	Chip select output ($\overline{\text{CS1}}$) (PA21 in single-chip mode)	

Bit 9—Reserved: This bit is always read as 0 and should only be written with 0.

Bit 8—PA20 Mode (PA20MD): Selects the function of the PA20/$\overline{\text{CS0}}$ pin.

Bit 8: PA20MD	Description
0	General input/output (PA20) ($\overline{\text{CS0}}$ in on-chip ROM disabled modes) (Initial value)
1	Chip select output ($\overline{\text{CS0}}$) (PA20 in single-chip mode)

Bit 7—Reserved: This bit is always read as 0 and should only be written with 0.

Bit 6—PA19 Mode (PA19MD): Selects the function of the PA19/$\overline{\text{BS}}$ pin.

Bit 6: PA19MD	Description
0	General input/output (PA19) ($\overline{\text{BS}}$ in on-chip ROM disabled modes) (Initial value)
1	Bus start output ($\overline{\text{BS}}$) (PA19 in single-chip mode)

Bit 5—Reserved: This bit is always read as 0 and should only be written with 0.

Bit 4—PA18 Mode (PA18MD): Selects the function of the PA18/$\overline{\text{RD}}$ pin.

Bit 4: PA18MD	Description
0	General input/output (PA18) ($\overline{\text{RD}}$ in on-chip ROM disabled modes) (Initial value)
1	Read output ($\overline{\text{RD}}$) (PA18 in single-chip mode)

Bit 3—Reserved: This bit is always read as 0 and should only be written with 0.

Bit 2—PA17 Mode (PA17MD): Selects the function of the PA17/$\overline{\text{WR}}$ pin.

Bit 2: PA17MD	Description
0	General input/output (PA17) (Initial value)
1	Write output ($\overline{\text{WR}}$) (PA17 in single-chip mode)

Bits 1 and 0—PA16 Mode 1 and 0 (PA16MD1, PA16MD0): These bits select the function of the PA16/$\overline{\text{WRHH}}$/$\overline{\text{HHBS}}$/TCLKC/TIOC3A pin.

Bit 1: PA16MD1	Bit 0: PA16MD0	Description
0	0	General input/output (PA16) (Initial value) ($\overline{\text{WRHH}}$ or $\overline{\text{HHBS}}$ in on-chip ROM disabled modes)
	1	Byte write output ($\overline{\text{WRHH}}$) or byte strobe output ($\overline{\text{HHBS}}$) (PA16 in single-chip mode)
1	0	TPU clock input (TCLKC)
	1	TPU input capture input/output compare output (TIOC3A)

17.3.4 Port A Control Registers L1 and L2 (PACRL1, PACRL2)

Port A control registers L1 and L2 (PACRL1, PACRL2) are 16-bit readable/writable registers that select the functions of pins in port A.

PACRL1 selects the functions of port A pins PA15/$\overline{\text{WRHL}}$/$\overline{\text{HLBS}}$/TCLKD/TIOC3B to PA8/$\overline{\text{RAS0}}$, and PACRL2 selects the functions of port A pins PA1/$\overline{\text{OE1}}$ and PA0/$\overline{\text{OE0}}$.

Port A includes bus control signals ($\overline{\text{WRHL}}$, $\overline{\text{WRLH}}$, $\overline{\text{WRLL}}$, $\overline{\text{HLBS}}$, $\overline{\text{LHBS}}$, $\overline{\text{LLBS}}$, $\overline{\text{WAIT}}$, $\overline{\text{RAS0}}$, $\overline{\text{RAS1}}$, $\overline{\text{OE0}}$, and $\overline{\text{OE1}}$), but register settings relating to the selection of these pin functions may not be valid in all operating modes. For details, see table 17.10, Pin Functions in Each Operating Mode.

PACRL1 and PACRL2 are initialized to H'0000 by an external power-on reset, but are not initialized by a WDT reset, in standby mode, or in sleep mode.

Port A Control Register L1 (PACRL1)

Bit:	15	14	13	12	11	10	9	8
	PA15 MD1	PA15 MD0	—	PA14 MD	—	PA13 MD	—	PA12 MD
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R	R/W	R	R/W	R	R/W

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	—	PA9 MD	—	PA8 MD
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R	R/W

Bits 15 and 14—PA15 Mode 1 and 0 (PA15MD1, PA15MD0): These bits select the function of the PA15/$\overline{\text{WRHL}}$/$\overline{\text{HLBS}}$/TCLKD/TIOC3B pin.

Bit 15: PA15MD1	Bit 14: PA15MD0	Description
0	0	General input/output (PA15) (Initial value) ($\overline{\text{WRHL}}$ or $\overline{\text{HLBS}}$ in on-chip ROM disabled modes)
	1	Byte write output ($\overline{\text{WRHL}}$) or byte strobe output ($\overline{\text{HLBS}}$) (PA15 in single-chip mode)
1	0	TPU clock input (TCLKD)
	1	TPU input capture input/output compare output (TIOC3B)

Bit 13—Reserved: This bit is always read as 0 and should only be written with 0.

Bit 12—PA14 Mode (PA14MD): Selects the function of the PA14/$\overline{\text{WRLH}}$/$\overline{\text{LHBS}}$ pin.

Bit 12: PA14MD	Description
0	General input/output (PA14) (Initial value) ($\overline{\text{WRLH}}$ or $\overline{\text{LHBS}}$ in on-chip ROM disabled modes)
1	Byte write output ($\overline{\text{WRLH}}$) or byte strobe output ($\overline{\text{LHBS}}$) (PA14 in single-chip mode)

Bit 11—Reserved: This bit is always read as 0 and should only be written with 0.

Bit 10—PA13 Mode (PA13MD): Selects the function of the PA13/$\overline{\text{WRLL}}$/$\overline{\text{LLBS}}$ pin.

Bit 10: PA13MD	Description	
0	General input/output (PA13) ($\overline{\text{WRLL}}$ or $\overline{\text{LLBS}}$ in on-chip ROM disabled modes)	(Initial value)
1	Byte write output ($\overline{\text{WRLL}}$) or byte strobe output ($\overline{\text{LLBS}}$) (PA13 in single-chip mode)	

Bit 9—Reserved: This bit is always read as 0 and should only be written with 0.

Bit 8—PA12 Mode (PA12MD): Selects the function of the PA12/$\overline{\text{WAIT}}$ pin.

Bit 8: PA12MD	Description	
0	General input/output (PA12)	(Initial value)
1	Wait request input ($\overline{\text{WAIT}}$) (PA12 in single-chip mode)	

Bits 7 to 3—Reserved: These bits are always read as 0 and should only be written with 0.

Bit 2—PA9 Mode (PA9MD): Selects the function of the PA9/$\overline{\text{RAS1}}$ pin.

Bit 2: PA9MD	Description	
0	General input/output (PA9)	(Initial value)
1	Row address strobe output ($\overline{\text{RAS1}}$) (PA9 in single-chip mode)	

Bit 1—Reserved: This bit is always read as 0 and should only be written with 0.

Bit 0—PA8 Mode (PA8MD): Selects the function of the PA8/$\overline{\text{RAS0}}$ pin.

Bit 0: PA8MD	Description	
0	General input/output (PA8)	(Initial value)
1	Row address strobe output ($\overline{\text{RAS0}}$) (PA8 in single-chip mode)	

Port A Control Register L2 (PACRL2)

Bit:	15	14	13	12	11	10	9	8
	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	—	PA1MD	—	PA0MD
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R	R/W

Bits 15 to 3—Reserved: These bits are always read as 0 and should only be written with 0.

Bit 2—PA1 Mode (PA1MD): Selects the function of the PA1/$\overline{\text{OE1}}$ pin.

Bit 2: PA1MD	Description	
0	General input/output (PA1)	(Initial value)
1	Output enable output ($\overline{\text{OE1}}$) (PA1 in single-chip mode)	

Bit 1—Reserved: This bit is always read as 0 and should only be written with 0.

Bit 0—PA0 Mode (PA0MD): Selects the function of the PA0/$\overline{\text{OE0}}$ pin.

Bit 0: PA0MD	Description	
0	General input/output (PA0)	(Initial value)
1	Output enable output ($\overline{\text{OE0}}$) (PA0 in single-chip mode)	

17.3.5 Port B IO Register H (PBIORH)

Bit:	15	14	13	12	11	10	9	8
	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit:	7	6	5	4	3	2	1	0
	PB23IOR	PB22IOR	PB21IOR	PB20IOR	PB19IOR	PB18IOR	PB17IOR	PB16IOR
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

Port B IO register H (PBIORH) is a 16-bit readable/writable register that selects the input/output direction of pins in port B. Bits PB23IOR to PB16IOR correspond to pins PB23/$\overline{\text{CASHH1}}$/TxD1/$\overline{\text{TEND0}}$ to PB16/$\overline{\text{CASLL0}}$. PBIORH is enabled when port B pins function as general input/output pins (PB23 to PB16), and disabled otherwise.

When port B pins function as PB23 to PB16, a pin becomes an output when the corresponding bit in PBIORH is set to 1, and an input when the bit is cleared to 0.

PBIORH is initialized to H'0000 by an external power-on reset, but is not initialized by a WDT reset, in standby mode, or in sleep mode.

17.3.6 Port B IO Register L (PBIORL)

Bit:	15	14	13	12	11	10	9	8
	—	—	PB13IOR	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R	R	R	R	R

Bit:	7	6	5	4	3	2	1	0
	PB7IOR	PB6IOR	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R	R	R	R	R	R

Port B IO register L (PBIORL) is a 16-bit readable/writable register that selects the input/output direction of pins in port B. Bits PB13IOR to PB6IOR correspond to pins PB13/RDWR to

PB6/$\overline{\text{BREQ}}$. PBIORL is enabled when port B pins function as general input/output pins (PB13 to PB6), and disabled otherwise.

When port B pins function as PB13 to PB6, a pin becomes an output when the corresponding bit in PBIORL is set to 1, and an input when the bit is cleared to 0.

PBIORL is initialized to H'0000 by an external power-on reset, but is not initialized by a WDT reset, in standby mode, or in sleep mode.

17.3.7 Port B Control Register H2 (PBCRH2)

Port B control register H2 (PBCRH2) is a 16-bit readable/writable register that selects the functions of the upper 8 multiplex pins in port B.

PBCRH2 selects the functions of port B pins PB23/$\overline{\text{CASHH1}}$/TxD1/$\overline{\text{TEND0}}$ to PB16/$\overline{\text{CASLL0}}$.

Port B includes bus control signals ($\overline{\text{CASLL0}}$, $\overline{\text{CASLL1}}$, $\overline{\text{CASLH0}}$, $\overline{\text{CASLH1}}$, $\overline{\text{CASHL0}}$, $\overline{\text{CASHL1}}$, $\overline{\text{CASHH0}}$, and $\overline{\text{CASHH1}}$) and DMAC control signals ($\overline{\text{TEND0}}$ and $\overline{\text{TEND1}}$), but register settings relating to the selection of these pin functions may not be valid in all operating modes. For details, see table 17.10, Pin Functions in Each Operating Mode.

PBCRH2 is initialized to H'0000 by an external power-on reset, but is not initialized by a WDT reset, in standby mode, or in sleep mode.

Port B Control Register H2 (PBCRH2)

Bit:	15	14	13	12	11	10	9	8
	PB23 MD1	PB23 MD0	PB22 MD1	PB22 MD0	—	PB21 MD	—	PB20 MD
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R	R/W	R	R/W

Bit:	7	6	5	4	3	2	1	0
	PB19 MD1	PB19 MD0	PB18 MD1	PB18 MD0	—	PB17 MD	—	PB16 MD
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R	R/W	R	R/W

Bits 15 and 14—PB23 Mode 1 and 0 (PB23MD1, PB23MD0): These bits select the function of the PB23/$\overline{\text{CASHH1}}$/TxD1/$\overline{\text{TEND0}}$ pin.

Bit 15: PB23MD1	Bit 14: PB23MD0	Description
0	0	General input/output (PB23) (Initial value)
	1	Column address strobe output ($\overline{\text{CASHH1}}$) (PB23 in single-chip mode)
1	0	SCI transmit data output (TxD1)
	1	DMAC transfer end output ($\overline{\text{TEND0}}$) (PB23 in single-chip mode)

Bits 13 and 12—PB22 Mode 1 and 0 (PB22MD1, PB22MD0): These bits select the function of the PB22/$\overline{\text{CASHL1}}$/RxD1/$\overline{\text{TEND1}}$ pin.

Bit 13: PB22MD1	Bit 12: PB22MD0	Description
0	0	General input/output (PB22) (Initial value)
	1	Column address strobe output ($\overline{\text{CASHL1}}$) (PB22 in single-chip mode)
1	0	SCI receive data input (RxD1)
	1	DMAC transfer end output ($\overline{\text{TEND1}}$) (PB22 in single-chip mode)

Bit 11—Reserved: This bit is always read as 0 and should only be written with 0.

Bit 10—PB21 Mode (PB21MD): Selects the function of the PB21/$\overline{\text{CASLH1}}$ pin.

Bit 10: PB21MD	Description
0	General input/output (PB21) (Initial value)
1	Column address strobe output ($\overline{\text{CASLH1}}$) (PB21 in single-chip mode)

Bit 9—Reserved: This bit is always read as 0 and should only be written with 0.

Bit 8—PB20 Mode (PB20MD): Selects the function of the PB20/$\overline{\text{CASLL1}}$ pin.

Bit 8: PB20MD	Description
0	General input/output (PB20) (Initial value)
1	Column address strobe output ($\overline{\text{CASLL1}}$) (PB20 in single-chip mode)

Bits 7 and 6—PB19 Mode 1 and 0 (PB19MD1, PB19MD0): These bits select the function of the PB19/$\overline{\text{CASHH0}}$/TxD0 pin.

Bit 7: PB19MD1	Bit 6: PB19MD0	Description
0	0	General input/output (PB19) (Initial value)
	1	Column address strobe output ($\overline{\text{CASHH0}}$) (PB19 in single-chip mode)
1	0	SCI transmit data output (TxD0)
	1	Reserved (Do not set)

Bits 5 and 4—PB18 Mode 1 and 0 (PB18MD1, PB18MD0): These bits select the function of the PB18/$\overline{\text{CASHL0}}$/RxD0 pin.

Bit 5: PB18MD1	Bit 4: PB18MD0	Description
0	0	General input/output (PB18) (Initial value)
	1	Column address strobe output ($\overline{\text{CASHL0}}$) (PB18 in single-chip mode)
1	0	SCI receive data input (RxD0)
	1	Reserved (Do not set)

Bit 3—Reserved: This bit is always read as 0 and should only be written with 0.

Bit 2—PB17 Mode (PB17MD): Selects the function of the PB17/$\overline{\text{CASLH0}}$ pin.

Bit 2: PB17MD	Description
0	General input/output (PB17) (Initial value)
1	Column address strobe output ($\overline{\text{CASLH0}}$) (PB17 in single-chip mode)

Bit 1—Reserved: This bit is always read as 0 and should only be written with 0.

Bit 0—PB16 Mode (PB16MD): Selects the function of the PB16/$\overline{\text{CASLL0}}$ pin.

Bit 0: PB16MD	Description
0	General input/output (PB16) (Initial value)
1	Column address strobe output ($\overline{\text{CASLL0}}$) (PB16 in single-chip mode)

17.3.8 Port B Control Registers L1 and L2 (PBCRL1, PBCRL2)

Port B control registers L1 and L2 (PBCRL1, PBCRL2) are 16-bit readable/writable registers that select the functions of pins in port B.

PBCRL1 selects the functions of port B pin PB13/RDWR, and PBCRL2 selects the functions of port B pins PB7/$\overline{\text{BACK}}$ to PB6/$\overline{\text{BREQ}}$.

Port B includes bus control signals (RDWR, $\overline{\text{BACK}}$, and $\overline{\text{BREQ}}$), but register settings relating to the selection of these pin functions may not be valid in all operating modes. For details, see table 17.10, Pin Functions in Each Operating Mode.

PBCRL1 and PBCRL2 are initialized to H'0000 by an external power-on reset, but are not initialized by a WDT reset, in standby mode, or in sleep mode.

Bit:	15	14	13	12	11	10	9	8
	—	—	—	—	—	PB13MD	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R	R

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

Bits 15 to 11—Reserved: These bits are always read as 0 and should only be written with 0.

Bit 10—PB13 Mode (PB13MD): Selects the function of the PB13/RDWR pin.

Bit 10: PB13MD	Description	
0	General input/output (PB13)	(Initial value)
1	Read/write output (RDWR) (PB13 in single-chip mode)	

Bits 9 to 0—Reserved: These bits are always read as 0 and should only be written with 0.

Port B Control Register L2 (PBCRL2)

Bit:	15	14	13	12	11	10	9	8
	—	PB7MD	—	PB6MD	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R/W	R	R/W	R	R	R	R

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

Bit 15—Reserved: This bit is always read as 0 and should only be written with 0.

Bit 14—PB7 Mode (PB7MD): Selects the function of the PB7/$\overline{\text{BACK}}$ pin.

Bit 14: PB7MD	Description	
0	General input/output (PB7)	(Initial value)
1	BUS request acknowledge output ($\overline{\text{BACK}}$) (PB7 in single-chip mode)	

Bit 13—Reserved: This bit is always read as 0 and should only be written with 0.

Bit 12—PB6 Mode (PB6MD): Selects the function of the PB6/$\overline{\text{BREQ}}$ pin.

Bit 12: PB6MD	Description	
0	General input/output (PB6)	(Initial value)
1	Bus release request input ($\overline{\text{BREQ}}$) (PB6 in single-chip mode)	

Bits 11 to 0—Reserved: These bits are always read as 0 and should only be written with 0.

17.3.9 Port C IO Register H (PCIORH)

Bit:	15	14	13	12	11	10	9	8
	—	—	—	—	—	—	PC25IOR	PC24IOR
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
	PC23IOR	PC22IOR	PC21IOR	PC20IOR	PC19IOR	PC18IOR	PC17IOR	PC16IOR
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

Port C IO register H (PCIORH) is a 16-bit readable/writable register that selects the input/output direction of pins in port C. Bits PC25IOR to PC16IOR correspond to pins PC25/A25/TIOC3B/TCLKD to PC16/A16/TIOC3A. PCIORH is enabled when port C pins function as general input/output pins (PC25 to PC16) or TPU TIOC pins, and disabled otherwise.

When port C pins function as PC25 to PC16 or TPU TIOC pins, a pin becomes an output when the corresponding bit in PCIORH is set to 1, and an input when the bit is cleared to 0.

PCIORH is initialized to H'0000 by an external power-on reset, but is not initialized by a WDT reset, in standby mode, or in sleep mode.

17.3.10 Port C IO Register L (PCIORL)

Bit:	15	14	13	12	11	10	9	8
	PC15IOR	PC14IOR	PC13IOR	PC12IOR	PC11IOR	PC10IOR	PC9IOR	PC8IOR
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
	PC7IOR	PC6IOR	PC5IOR	PC4IOR	PC3IOR	PC2IOR	PC1IOR	PC0IOR
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

Port C IO register L (PCIORL) is a 16-bit readable/writable register that selects the input/output direction of pins in port C. Bits PC15IOR to PC0IOR correspond to pins PC15/A15/TIOC3D to

PC0/A0. PCIORL is enabled when port C pins function as general input/output pins (PC15 to PC0) or TPU TIOC pins, and disabled otherwise.

When port C pins function as PC15 to PC0 or TPU TIOC pins, a pin becomes an output when the corresponding bit in PCIORL is set to 1, and an input when the bit is cleared to 0.

PCIORL is initialized to H'0000 by an external power-on reset, but is not initialized by a WDT reset, in standby mode, or in sleep mode.

17.3.11 Port C Control Registers H1 and H2 (PCCRH1, PCCRH2)

Port C control registers H1 and H2 (PCCRH1, PCCRH2) are 16-bit readable/writable registers that select the functions of pins in port C.

PCCRH1 selects the functions of port C pins PC25/A25/TIOC3B/TCLKD and PC24/A24/TIOC3A/TCLKC, and PCCRH2 selects the functions of port C pins PC23/A23/TIOC1B/TCLKB to PC16/A16/TIOC3A.

Port C includes address outputs (A16 to A25), but register settings relating to the selection of these pin functions may not be valid in all operating modes. For details, see table 17.10, Pin Functions in Each Operating Mode.

PCCRH1 and PCCRH2 are initialized to H'0000 by an external power-on reset, but are not initialized by a WDT reset, in standby mode, or in sleep mode.

Port C Control Register H1 (PCCRH1)

Bit:	15	14	13	12	11	10	9	8
	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	PC25 MD1	PC25 MD0	PC24 MD1	PC24 MD0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W	R/W

Bits 15 to 4—Reserved: These bits are always read as 0 and should only be written with 0.

Bits 3 and 2—PC25 Mode 1 and 0 (PC25MD1, PC25MD0): These bits select the function of the PC25/A25/TIOC3B/TCLKD pin.

Bit 3: PC25MD1	Bit 2: PC25MD0	Description
0	0	General input/output (PC25) (Initial value) (A25 in on-chip ROM disabled modes)
	1	Address output (A25) (PC25 in single-chip mode)
1	0	TPU input capture input/output compare output (TIOC3B)
	1	TPU clock input (TCLKD)

Bits 1 and 0—PC24 Mode 1 and 0 (PC24MD1, PC24MD0): These bits select the function of the PC24/A24/TIOC3A/TCLKC pin.

Bit 1: PC24MD1	Bit 0: PC24MD0	Description
0	0	General input/output (PC24) (Initial value) (A24 in on-chip ROM disabled modes)
	1	Address output (A24) (PC24 in single-chip mode)
1	0	TPU input capture input/output compare output (TIOC3A)
	1	TPU clock input (TCLKC)

Port C Control Register H2 (PCCRH2)

Bit:	15	14	13	12	11	10	9	8
	PC23 MD1	PC23 MD0	PC22 MD1	PC22 MD0	PC21 MD1	PC21 MD0	PC20 MD1	PC20 MD0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
	PC19 MD1	PC19 MD0	PC18 MD1	PC18 MD0	PC17 MD1	PC17 MD0	PC16 MD1	PC16 MD0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bits 15 and 14—PC23 Mode 1 and 0 (PC23MD1, PC23MD0): These bits select the function of the PC23/A23/TIOC1B/TCLKB pin.

Bit 15: PC23MD1	Bit 14: PC23MD0	Description
0	0	General input/output (PC23) (Initial value) (A23 in on-chip ROM disabled modes)
	1	Address output (A23) (PC23 in single-chip mode)
1	0	TPU input capture input/output compare output (TIOC1B)
	1	TPU clock input (TCLKB)

Bits 13 and 12—PC22 Mode 1 and 0 (PC22MD1, PC22MD0): These bits select the function of the PC22/A22/TIOC1A/TCLKA pin.

Bit 13: PC22MD1	Bit 12: PC22MD0	Description
0	0	General input/output (PC22) (Initial value) (A22 in on-chip ROM disabled modes)
	1	Address output (A22) (PC22 in single-chip mode)
1	0	TPU input capture input/output compare output (TIOC1A)
	1	TPU clock input (TCLKA)

Bits 11 and 10—PC21 Mode 1 and 0 (PC21MD1, PC21MD0): These bits select the function of the PC21/A21/TIOC5B pin.

Bit 11: PC21MD1	Bit 10: PC21MD0	Description
0	0	General input/output (PC21) (Initial value) (A21 in on-chip ROM disabled modes)
	1	Address output (A21) (PC21 in single-chip mode)
1	0	TPU input capture input/output compare output (TIOC5B)
	1	Reserved (Do not set)

Bits 9 and 8—PC20 Mode 1 and 0 (PC20MD1, PC20MD0): These bits select the function of the PC20/A20/TIOC5A pin.

Bit 9: PC20MD1	Bit 8: PC20MD0	Description
0	0	General input/output (PC20) (Initial value) (A20 in on-chip ROM disabled modes)
	1	Address output (A20) (PC20 in single-chip mode)
1	0	TPU input capture input/output compare output (TIOC5A)
	1	Reserved (Do not set)

Bits 7 and 6—PC19 Mode 1 and 0 (PC19MD1, PC19MD0): These bits select the function of the PC19/A19/TIOC4B pin.

Bit 7: PC19MD1	Bit 6: PC19MD0	Description
0	0	General input/output (PC19) (Initial value) (A19 in on-chip ROM disabled modes)
	1	Address output (A19) (PC19 in single-chip mode)
1	0	TPU input capture input/output compare output (TIOC4B)
	1	Reserved (Do not set)

Bits 5 and 4—PC18 Mode 1 and 0 (PC18MD1, PC18MD0): These bits select the function of the PC18/A18/TIOC4A pin.

Bit 5: PC18MD1	Bit 4: PC18MD0	Description
0	0	General input/output (PC18) (Initial value) (A18 in on-chip ROM disabled modes)
	1	Address output (A18) (PC18 in single-chip mode)
1	0	TPU input capture input/output compare output (TIOC4A)
	1	Reserved (Do not set)

Bits 3 and 2—PC17 Mode 1 and 0 (PC17MD1, PC17MD0): These bits select the function of the PC17/A17/TIOC3B pin.

Bit 3: PC17MD1	Bit 2: PC17MD0	Description
0	0	General input/output (PC17) (Initial value) (A17 in on-chip ROM disabled modes)
	1	Address output (A17) (PC17 in single-chip mode)
1	0	TPU input capture input/output compare output (TIOC3B)
	1	Reserved (Do not set)

Bits 1 and 0—PC16 Mode 1 and 0 (PC16MD1, PC16MD0): These bits select the function of the PC16/A16/TIOC3A pin.

Bit 1: PC16MD1	Bit 0: PC16MD0	Description
0	0	General input/output (PC16) (Initial value) (A16 in on-chip ROM disabled modes)
	1	Address output (A16) (PC16 in single-chip mode)
1	0	TPU input capture input/output compare output (TIOC3A)
	1	Reserved (Do not set)

17.3.12 Port C Control Registers L1 and L2 (PCCRL1, PCCRL2)

Port C control registers L1 and L2 (PCCRL1, PCCRL2) are 16-bit readable/writable registers that select the functions of pins in port C.

PCCRL1 selects the functions of port C pins PC15/A15/TIOC3D to PC8/A8, and PCCRL2 selects the functions of port C pins PC7/A7 to PC0/A0.

Port C includes address outputs (A0 to A15), but register settings relating to the selection of these pin functions may not be valid in all operating modes. For details, see table 17.10, Pin Functions in Each Operating Mode.

PCCRL1 and PCCRL2 are initialized to H'0000 by an external power-on reset, but are not initialized by a WDT reset, in standby mode, or in sleep mode.

Port C Control Register L1 (PCCRL1)

Bit:	15	14	13	12	11	10	9	8
	PC15 MD1	PC15 MD0	PC14 MD1	PC14 MD0	—	PC13 MD	—	PC12 MD
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R	R/W	R	R/W

Bit:	7	6	5	4	3	2	1	0
	—	PC11 MD	—	PC10 MD	—	PC9 MD	—	PC8 MD
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R/W	R	R/W	R	R/W	R	R/W

Bits 15 and 14—PC15 Mode 1 and 0 (PC15MD1, PC15MD0): These bits select the function of the PC15/A15/TIOC3D pin.

Bit 15: PC15MD1	Bit 14: PC15MD0	Description
0	0	General input/output (PC15) (Initial value) (A15 in on-chip ROM disabled modes)
	1	Address output (A15) (PC15 in single-chip mode)
1	0	TPU input capture input/output compare output (TIOC3D)
	1	Reserved (Do not set)

Bits 13 and 12—PC14 Mode 1 and 0 (PC14MD1, PC14MD0): These bits select the function of the PC14/A14/TIOC3C pin.

Bit 13: PC14MD1	Bit 12: PC14MD0	Description
0	0	General input/output (PC14) (Initial value) (A14 in on-chip ROM disabled modes)
	1	Address output (A14) (PC14 in single-chip mode)
1	0	TPU input capture input/output compare output (TIOC3C)
	1	Reserved (Do not set)

Bit 11—Reserved: This bit is always read as 0 and should only be written with 0.

Bit 10—PC13 Mode (PC13MD): Selects the function of the PC13/A13 pin.

Bit 10: PC13MD	Description	
0	General input/output (PC13) (A13 in on-chip ROM disabled modes)	(Initial value)
1	Address output (A13) (PC13 in single-chip mode)	

Bit 9—Reserved: This bit is always read as 0 and should only be written with 0.

Bit 8—PC12 Mode (PC12MD): Selects the function of the PC12/A12 pin.

Bit 8: PC12MD	Description	
0	General input/output (PC12) (A12 in on-chip ROM disabled modes)	(Initial value)
1	Address output (A12) (PC12 in single-chip mode)	

Bit 7—Reserved: This bit is always read as 0 and should only be written with 0.

Bit 6—PC11 Mode (PC11MD): Selects the function of the PC11/A11 pin.

Bit 6: PC11MD	Description	
0	General input/output (PC11) (A11 in on-chip ROM disabled modes)	(Initial value)
1	Address output (A11) (PC11 in single-chip mode)	

Bit 5—Reserved: This bit is always read as 0 and should only be written with 0.

Bit 4—PC10 Mode (PC10MD): Selects the function of the PC10/A10 pin.

Bit 4: PC10MD	Description	
0	General input/output (PC10) (A10 in on-chip ROM disabled modes)	(Initial value)
1	Address output (A10) (PC10 in single-chip mode)	

Bit 3—Reserved: This bit is always read as 0 and should only be written with 0.

Bit 2—PC9 Mode (PC9MD): Selects the function of the PC9/A9 pin.

Bit 2: PC9MD	Description	
0	General input/output (PC9) (A9 in on-chip ROM disabled modes)	(Initial value)
1	Address output (A9) (PC9 in single-chip mode)	

Bit 1—Reserved: This bit is always read as 0 and should only be written with 0.

Bit 0—PC8 Mode (PC8MD): Selects the function of the PC8/A8 pin.

Bit 0: PC8MD	Description	
0	General input/output (PC8) (A8 in on-chip ROM disabled modes)	(Initial value)
1	Address output (A8) (PC8 in single-chip mode)	

Port C Control Register L2 (PCCRL2)

Bit:	15	14	13	12	11	10	9	8
	—	PC7MD	—	PC6MD	—	PC5MD	—	PC4MD
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R/W	R	R/W	R	R/W	R	R/W

Bit:	7	6	5	4	3	2	1	0
	—	PC3MD	—	PC2MD	—	PC1MD	—	PC0MD
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R/W	R	R/W	R	R/W	R	R/W

Bit 15—Reserved: This bit is always read as 0 and should only be written with 0.

Bit 14—PC7 Mode (PC7MD): Selects the function of the PC7/A7 pin.

Bit 14: PC7MD	Description	
0	General input/output (PC7) (A7 in on-chip ROM disabled modes)	(Initial value)
1	Address output (A7) (PC7 in single-chip mode)	

Bit 13—Reserved: This bit is always read as 0 and should only be written with 0.

Bit 12—PC6 Mode (PC6MD): Selects the function of the PC6/A6 pin.

Bit 12: PC6MD	Description	
0	General input/output (PC6) (A6 in on-chip ROM disabled modes)	(Initial value)
1	Address output (A6) (PC6 in single-chip mode)	

Bit 11—Reserved: This bit is always read as 0 and should only be written with 0.

Bit 10—PC5 Mode (PC5MD): Selects the function of the PC5/A5 pin.

Bit 10: PC5MD	Description	
0	General input/output (PC5) (A5 in on-chip ROM disabled modes)	(Initial value)
1	Address output (A5) (PC5 in single-chip mode)	

Bit 9—Reserved: This bit is always read as 0 and should only be written with 0.

Bit 8—PC4 Mode (PC4MD): Selects the function of the PC4/A4 pin.

Bit 8: PC4MD	Description	
0	General input/output (PC4) (A4 in on-chip ROM disabled modes)	(Initial value)
1	Address output (A4) (PC4 in single-chip mode)	

Bit 7—Reserved: This bit is always read as 0 and should only be written with 0.

Bit 6—PC3 Mode (PC3MD): Selects the function of the PC3/A3 pin.

Bit 6: PC3MD	Description	
0	General input/output (PC3) (A3 in on-chip ROM disabled modes)	(Initial value)
1	Address output (A3) (PC3 in single-chip mode)	

Bit 5—Reserved: This bit is always read as 0 and should only be written with 0.

Bit 4—PC2 Mode (PC2MD): Selects the function of the PC2/A2 pin.

Bit 4: PC2MD	Description	
0	General input/output (PC2) (A2 in on-chip ROM disabled modes)	(Initial value)
1	Address output (A2) (PC2 in single-chip mode)	

Bit 3—Reserved: This bit is always read as 0 and should only be written with 0.

Bit 2—PC1 Mode (PC1MD): Selects the function of the PC1/A1 pin.

Bit 2: PC1MD	Description	
0	General input/output (PC1) (A1 in on-chip ROM disabled modes)	(Initial value)
1	Address output (A1) (PC1 in single-chip mode)	

Bit 1—Reserved: This bit is always read as 0 and should only be written with 0.

Bit 0—PC0 Mode (PC0MD): Selects the function of the PC0/A0 pin.

Bit 0: PC0MD	Description	
0	General input/output (PC0) (A0 in on-chip ROM disabled modes)	(Initial value)
1	Address output (A0) (PC0 in single-chip mode)	

17.3.13 Port D IO Register H (PDIORH)

Bit:	15	14	13	12	11	10	9	8
	PD31IOR	PD30IOR	PD29IOR	PD28IOR	PD27IOR	PD26IOR	PD25IOR	PD24IOR
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
	PD23IOR	PD22IOR	PD21IOR	PD20IOR	PD19IOR	PD18IOR	PD17IOR	PD16IOR
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

Port D IO register H (PDIORH) is a 16-bit readable/writable register that selects the input/output direction of pins in port D. Bits PD31IOR to PD16IOR correspond to pins PD31/D31/RxD2/TIOC5A to PD16/D16/$\overline{\text{POE0}}$. PDIORH is enabled when port D pins function as general input/output pins (PD31 to PD16), SCI SCK pins, or TPU TIOC pins, or when PD23 functions as the MMT PCIO pin, and disabled otherwise.

When port D pins function as PD31 to PD16, SCI SCK pins, or TPU TIOC pins, or when PD23 functions as the MMT PCIO pin, a pin becomes an output when the corresponding bit in PDIORH is set to 1, and an input when the bit is cleared to 0.

PDIORH is initialized to H'0000 by an external power-on reset, but is not initialized by a WDT reset, in standby mode, or in sleep mode.

17.3.14 Port D IO Register L (PDIORL)

Bit:	15	14	13	12	11	10	9	8
	PD15IOR	PD14IOR	PD13IOR	PD12IOR	PD11IOR	PD10IOR	PD9IOR	PD8IOR
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
	PD7IOR	PD6IOR	PD5IOR	PD4IOR	PD3IOR	PD2IOR	PD1IOR	PD0IOR
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

Port D IO register L (PDIORL) is a 16-bit readable/writable register that selects the input/output direction of pins in port D. Bits PD15IOR to PD0IOR correspond to pins PD15/D15/TIOC5B to PD0/D0. PDIORL is enabled when port D pins function as general input/output pins (PD15 to PD0) or TPU TIOC pins, and disabled otherwise.

When port D pins function as PD15 to PD0 or TPU TIOC pins, a pin becomes an output when the corresponding bit in PDIORL is set to 1, and an input when the bit is cleared to 0.

PDIORL is initialized to H'0000 by an external power-on reset, but is not initialized by a WDT reset, in standby mode, or in sleep mode.

17.3.15 Port D Control Registers H1 and H2 (PDCRH1, PDCRH2)

Port D control registers H1 and H2 (PDCRH1, PDCRH2) are 16-bit readable/writable registers that select the functions of pins in port D.

PDCRH1 selects the functions of port D pins PD31/D31/RxD2/TIOC5A to PD24/D24/PUOB, and PDCRH2 selects the functions of port D pins PD23/D23/PCIO/SCK1 to PD16/D16/$\overline{\text{POE}}$0.

Port D includes data input/output functions (D16 to D31), but register settings relating to the selection of these pin functions may not be valid in all operating modes. For details, see table 17.10, Pin Functions in Each Operating Mode.

PDCRH1 and PDCRH2 are initialized to H'0000 by an external power-on reset, but are not initialized by a WDT reset, in standby mode, or in sleep mode.

Port D Control Register H1 (PDCRH1)

Bit:	15	14	13	12	11	10	9	8
	PD31 MD1	PD31 MD0	PD30 MD1	PD30 MD0	PD29 MD1	PD29 MD0	PD28 MD1	PD28 MD0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
	PD27 MD1	PD27 MD0	PD26 MD1	PD26 MD0	PD25 MD1	PD25 MD0	PD24 MD1	PD24 MD0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bits 15 and 14—PD31 Mode 1 and 0 (PD31MD1, PD31MD0): These bits select the function of the PD31/D31/RxD2/TIOC5A pin.

Bit 15: PD31MD1	Bit 14: PD31MD0	Description
0	0	General input/output (PD31) (Initial value) (D31 in on-chip ROM disabled modes with 32-bit CS0 bus width)
	1	Data input/output (D31) (PD31 in single-chip mode)
1	0	SCI receive data input (RxD2)
	1	TPU input capture input/output compare output (TIOC5A)

Bits 13 and 12—PD30 Mode 1 and 0 (PD30MD1, PD30MD0): These bits select the function of the PD30/D30/TxD2/TIOC4B pin.

Bit 13: PD30MD1	Bit 12: PD30MD0	Description
0	0	General input/output (PD30) (Initial value) (D30 in on-chip ROM disabled modes with 32-bit CS0 bus width)
	1	Data input/output (D30) (PD30 in single-chip mode)
1	0	SCI transmit data output (TxD2)
	1	TPU input capture input/output compare output (TIOC4B)

Bits 11 and 10—PD29 Mode 1 and 0 (PD29MD1, PD29MD0): These bits select the function of the PD29/D29/SCK2/TIOC4A pin.

Bit 11: PD29MD1	Bit 10: PD29MD0	Description
0	0	General input/output (PD29) (Initial value) (D29 in on-chip ROM disabled modes with 32-bit CS0 bus width)
	1	Data input/output (D29) (PD29 in single-chip mode)
1	0	SCI clock input/output (SCK2)
	1	TPU input capture input/output compare output (TIOC4A)

Bits 9 and 8—PD28 Mode 1 and 0 (PD28MD1, PD28MD0): These bits select the function of the PD28/D28/TCLKB/TIOC3D pin.

Bit 9: PD28MD1	Bit 8: PD28MD0	Description
0	0	General input/output (PD28) (Initial value) (D28 in on-chip ROM disabled modes with 32-bit CS0 bus width)
	1	Data input/output (D28) (PD28 in single-chip mode)
1	0	TPU clock input (TCLKB)
	1	TPU input capture input/output compare output (TIOC3D)

Bits 7 and 6—PD27 Mode 1 and 0 (PD27MD1, PD27MD0): These bits select the function of the PD27/D27/TCLKA/TIOC3C pin.

Bit 7: PD27MD1	Bit 6: PD27MD0	Description
0	0	General input/output (PD27) (Initial value) (D27 in on-chip ROM disabled modes with 32-bit CS0 bus width)
	1	Data input/output (D27) (PD27 in single-chip mode)
1	0	TPU clock input (TCLKA)
	1	TPU input capture input/output compare output (TIOC3C)

Bits 5 and 4—PD26 Mode 1 and 0 (PD26MD1, PD26MD0): These bits select the function of the PD26/D26/PWOB pin.

Bit 5: PD26MD1	Bit 4: PD26MD0	Description
0	0	General input/output (PD26) (Initial value) (D26 in on-chip ROM disabled modes with 32-bit CS0 bus width)
	1	Data input/output (D26) (PD26 in single-chip mode)
1	0	MMT PWM W-phase output (PWOB)
	1	Reserved (Do not set)

Bits 3 and 2—PD25 Mode 1 and 0 (PD25MD1, PD25MD0): These bits select the function of the PD25/D25/PVOB pin.

Bit 3: PD25MD1	Bit 2: PD25MD0	Description
0	0	General input/output (PD25) (Initial value) (D25 in on-chip ROM disabled modes with 32-bit CS0 bus width)
	1	Data input/output (D25) (PD25 in single-chip mode)
1	0	MMT PWM V-phase output (PVOB)
	1	Reserved (Do not set)

Bits 1 and 0—PD24 Mode 1 and 0 (PD24MD1, PD24MD0): These bits select the function of the PD24/D24/PUOB pin.

Bit 1: PD24MD1	Bit 0: PD24MD0	Description
0	0	General input/output (PD24) (Initial value) (D24 in on-chip ROM disabled modes with 32-bit CS0 bus width)
	1	Data input/output (D24) (PD24 in single-chip mode)
1	0	MMT PWM U-phase output (PUOB)
	1	Reserved (Do not set)

Port D Control Register H2 (PDCRH2)

Bit:	15	14	13	12	11	10	9	8
	PD23 MD1	PD23 MD0	PD22 MD1	PD22 MD0	PD21 MD1	PD21 MD0	PD20 MD1	PD20 MD0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
	PD19 MD1	PD19 MD0	PD18 MD1	PD18 MD0	PD17 MD1	PD17 MD0	PD16 MD1	PD16 MD0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bits 15 and 14—PD23 Mode 1 and 0 (PD23MD1, PD23MD0): These bits select the function of the PD23/D23/PCIO/SCK1 pin.

Bit 15: PD23MD1	Bit 14: PD23MD0	Description
0	0	General input/output (PD23) (Initial value) (D23 in on-chip ROM disabled modes with 32-bit CS0 bus width)
	1	Data input/output (D23) (PD23 in single-chip mode)
1	0	MMT PWM cycle output (PCO)/counter clear input (PCI)
	1	SCI clock input/output (SCK1)

Bits 13 and 12—PD22 Mode 1 and 0 (PD22MD1, PD22MD0): These bits select the function of the PD22/D22/PWOA/SCK0 pin.

Bit 13: PD22MD1	Bit 12: PD22MD0	Description
0	0	General input/output (PD22) (Initial value) (D22 in on-chip ROM disabled modes with 32-bit CS0 bus width)
	1	Data input/output (D22) (PD22 in single-chip mode)
1	0	MMT PWM W-phase output (PWOA)
	1	SCI clock input/output (SCK0)

Bits 11 and 10—PD21 Mode 1 and 0 (PD21MD1, PD21MD0): These bits select the function of the PD21/D21/PVOA/$\overline{\text{IRQ7}}$ pin.

Bit 11: PD21MD1	Bit 10: PD21MD0	Description
0	0	General input/output (PD21) (Initial value) (D21 in on-chip ROM disabled modes with 32-bit CS0 bus width)
	1	Data input/output (D21) (PD21 in single-chip mode)
1	0	MMT PWM V-phase output (PVOA)
	1	External interrupt request input ($\overline{\text{IRQ7}}$)

Bits 9 and 8—PD20 Mode 1 and 0 (PD20MD1, PD20MD0): These bits select the function of the PD20/D20/PUOA/$\overline{\text{IRQ6}}$ pin.

Bit 9: PD20MD1	Bit 8: PD20MD0	Description
0	0	General input/output (PD20) (Initial value) (D20 in on-chip ROM disabled modes with 32-bit CS0 bus width)
	1	Data input/output (D20) (PD20 in single-chip mode)
1	0	MMT PWM U-phase output (PUOA)
	1	External interrupt request input ($\overline{\text{IRQ6}}$)

Bits 7 and 6—PD19 Mode 1 and 0 (PD19MD1, PD19MD0): These bits select the function of the PD19/D19/$\overline{\text{POE3}}$/$\overline{\text{IRQ5}}$ pin.

Bit 7: PD19MD1	Bit 6: PD19MD0	Description
0	0	General input/output (PD19) (Initial value) (D19 in on-chip ROM disabled modes with 32-bit CS0 bus width)
	1	Data input/output (D19) (PD19 in single-chip mode)
1	0	MMT port output enable input ($\overline{\text{POE3}}$)
	1	External interrupt request input ($\overline{\text{IRQ5}}$)

Bits 5 and 4—PD18 Mode 1 and 0 (PD18MD1, PD18MD0): These bits select the function of the PD18/D18/$\overline{\text{POE2}}$/$\overline{\text{IRQ4}}$ pin.

Bit 5: PD18MD1	Bit 4: PD18MD0	Description
0	0	General input/output (PD18) (Initial value) (D18 in on-chip ROM disabled modes with 32-bit CS0 bus width)
	1	Data input/output (D18) (PD18 in single-chip mode)
1	0	MMT port output enable input ($\overline{\text{POE2}}$)
	1	External interrupt request input ($\overline{\text{IRQ4}}$)

Bits 3 and 2—PD17 Mode 1 and 0 (PD17MD1, PD17MD0): These bits select the function of the PD17/D17/$\overline{\text{POE1}}$/$\overline{\text{ADTRG}}$ pin.

Bit 3: PD17MD1	Bit 2: PD17MD0	Description
0	0	General input/output (PD17) (Initial value) (D17 in on-chip ROM disabled modes with 32-bit CS0 bus width)
	1	Data input/output (D17) (PD17 in single-chip mode)
1	0	MMT port output enable input ($\overline{\text{POE1}}$)
	1	A/D conversion trigger input ($\overline{\text{ADTRG}}$)

Bits 1 and 0—PD16 Mode 1 and 0 (PD16MD1, PD16MD0): These bits select the function of the PD16/D16/$\overline{\text{POE0}}$ pin.

Bit 1: PD16MD1	Bit 0: PD16MD0	Description
0	0	General input/output (PD16) (Initial value) (D16 in on-chip ROM disabled modes with 32-bit CS0 bus width)
	1	Data input/output (D16) (PD16 in single-chip mode)
1	0	MMT port output enable input ($\overline{\text{POE0}}$)
	1	Reserved (Do not set)

17.3.16 Port D Control Registers L1 and L2 (PDCRL1, PDCRL2)

Port D control registers L1 and L2 (PDCRL1, PDCRL2) are 16-bit readable/writable registers that select the functions of pins in port D.

PDCRL1 selects the functions of port D pins PD15/D15/TIOC5B to PD8/D8/TIOC1A, and PDCRL2 selects the functions of port D pins PD7/D7 to PD0/D0.

Port D includes data input/output functions (D0 to D15), but register settings relating to the selection of these pin functions may not be valid in all operating modes. For details, see table 17.10, Pin Functions in Each Operating Mode.

PDCRL1 and PDCRL2 are initialized to H'0000 by an external power-on reset, but are not initialized by a WDT reset, in standby mode, or in sleep mode.

Port D Control Register L1 (PDCRL1)

Bit:	15	14	13	12	11	10	9	8
	PD15 MD1	PD15 MD0	PD14 MD1	PD14 MD0	PD13 MD1	PD13 MD0	PD12 MD1	PD12 MD0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
	PD11 MD1	PD11 MD0	PD10 MD1	PD10 MD0	PD9 MD1	PD9 MD0	PD8 MD1	PD8 MD0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bits 15 and 14—PD15 Mode 1 and 0 (PD15MD1, PD15MD0): These bits select the function of the PD15/D15/TIOC5B pin.

Bit 15: PD15MD1	Bit 14: PD15MD0	Description
0	0	General input/output (PD15) (Initial value) (D15 in on-chip ROM disabled modes with 32-bit/16-bit CS0 bus width)
	1	Data input/output (D15) (PD15 in single-chip mode)
1	0	TPU input capture input/output compare output (TIOC5B)
	1	Reserved (Do not set)

Bits 13 and 12—PD14 Mode 1 and 0 (PD14MD1, PD14MD0): These bits select the function of the PD14/D14/TIOC5A pin.

Bit 13: PD14MD1	Bit 12: PD14MD0	Description
0	0	General input/output (PD14) (Initial value) (D14 in on-chip ROM disabled modes with 32-bit/16-bit CS0 bus width)
	1	Data input/output (D14) (PD14 in single-chip mode)
1	0	TPU input capture input/output compare output (TIOC5A)
	1	Reserved (Do not set)

Bits 11 and 10—PD13 Mode 1 and 0 (PD13MD1, PD13MD0): These bits select the function of the PD13/D13/TIOC4B pin.

Bit 11: PD13MD1	Bit 10: PD13MD0	Description
0	0	General input/output (PD13) (Initial value) (D13 in on-chip ROM disabled modes with 32-bit/16-bit CS0 bus width)
	1	Data input/output (D13) (PD13 in single-chip mode)
1	0	TPU input capture input/output compare output (TIOC4B)
	1	Reserved (Do not set)

Bits 9 and 8—PD12 Mode 1 and 0 (PD12MD1, PD12MD0): These bits select the function of the PD12/D12/TIOC4A pin.

Bit 9: PD12MD1	Bit 8: PD12MD0	Description
0	0	General input/output (PD12) (Initial value) (D12 in on-chip ROM disabled modes with 32-bit/16-bit CS0 bus width)
	1	Data input/output (D12) (PD12 in single-chip mode)
1	0	TPU input capture input/output compare output (TIOC4A)
	1	Reserved (Do not set)

Bits 7 and 6—PD11 Mode 1 and 0 (PD11MD1, PD11MD0): These bits select the function of the PD11/D11/TIOC2B pin.

Bit 7: PD11MD1	Bit 6: PD11MD0	Description
0	0	General input/output (PD11) (Initial value) (D11 in on-chip ROM disabled modes with 32-bit/16-bit CS0 bus width)
	1	Data input/output (D11) (PD11 in single-chip mode)
1	0	TPU input capture input/output compare output (TIOC2B)
	1	Reserved (Do not set)

Bits 5 and 4—PD10 Mode 1 and 0 (PD10MD1, PD10MD0): These bits select the function of the PD10/D10/TIOC2A pin.

Bit 5: PD10MD1	Bit 4: PD10MD0	Description
0	0	General input/output (PD10) (Initial value) (D10 in on-chip ROM disabled modes with 32-bit/16-bit CS0 bus width)
	1	Data input/output (D10) (PD10 in single-chip mode)
1	0	TPU input capture input/output compare output (TIOC2A)
	1	Reserved (Do not set)

Bits 3 and 2—PD9 Mode 1 and 0 (PD9MD1, PD9MD0): These bits select the function of the PD9/D9/TIOC1B pin.

Bit 3: PD9MD1	Bit 2: PD9MD0	Description
0	0	General input/output (PD9) (Initial value) (D9 in on-chip ROM disabled modes with 32-bit/16-bit CS0 bus width)
	1	Data input/output (D9) (PD9 in single-chip mode)
1	0	TPU input capture input/output compare output (TIOC1B)
	1	Reserved (Do not set)

Bits 1 and 0—PD8 Mode 1 and 0 (PD8MD1, PD8MD0): These bits select the function of the PD8/D8/TIOC1A pin.

Bit 1: PD8MD1	Bit 0: PD8MD0	Description
0	0	General input/output (PD8) (Initial value) (D8 in on-chip ROM disabled modes with 32-bit/16-bit CS0 bus width)
	1	Data input/output (D8) (PD8 in single-chip mode)
1	0	TPU input capture input/output compare output (TIOC1A)
	1	Reserved (Do not set)

Port D Control Register L2 (PDCRL2)

Bit:	15	14	13	12	11	10	9	8
	—	PD7MD	—	PD6MD	—	PD5MD	—	PD4MD
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R/W	R	R/W	R	R/W	R	R/W

Bit:	7	6	5	4	3	2	1	0
	—	PD3MD	—	PD2MD	—	PD1MD	—	PD0MD
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R/W	R	R/W	R	R/W	R	R/W

Bit 15—Reserved: This bit is always read as 0 and should only be written with 0.

Bit 14—PD7 Mode (PD7MD): Selects the function of the PD7/D7 pin.

Bit 14: PD7MD	Description	
0	General input/output (PD7) (D7 in on-chip ROM disabled modes)	(Initial value)
1	Data input/output (D7) (PD7 in single-chip mode)	

Bit 13—Reserved: This bit is always read as 0 and should only be written with 0.

Bit 12—PD6 Mode (PD6MD): Selects the function of the PD6/D6 pin.

Bit 12: PD6MD	Description	
0	General input/output (PD6) (D6 in on-chip ROM disabled modes)	(Initial value)
1	Data input/output (D6) (PD6 in single-chip mode)	

Bit 11—Reserved: This bit is always read as 0 and should only be written with 0.

Bit 10—PD5 Mode (PD5MD): Selects the function of the PD5/D5 pin.

Bit 10: PD5MD	Description	
0	General input/output (PD5) (D5 in on-chip ROM disabled modes)	(Initial value)
1	Data input/output (D5) (PD5 in single-chip mode)	

Bit 9—Reserved: This bit is always read as 0 and should only be written with 0.

Bit 8—PD4 Mode (PD4MD): Selects the function of the PD4/D4 pin.

Bit 8: PD4MD	Description	
0	General input/output (PD4) (D4 in on-chip ROM disabled modes)	(Initial value)
1	Data input/output (D4) (PD4 in single-chip mode)	

Bit 7—Reserved: This bit is always read as 0 and should only be written with 0.

Bit 6—PD3 Mode (PD3MD): Selects the function of the PD3/D3 pin.

Bit 6: PD3MD	Description	
0	General input/output (PD3) (D3 in on-chip ROM disabled modes)	(Initial value)
1	Data input/output (D3) (PD3 in single-chip mode)	

Bit 5—Reserved: This bit is always read as 0 and should only be written with 0.

Bit 4—PD2 Mode (PD2MD): Selects the function of the PD2/D2 pin.

Bit 4: PD2MD	Description	
0	General input/output (PD2) (D2 in on-chip ROM disabled modes)	(Initial value)
1	Data input/output (D2) (PD2 in single-chip mode)	

Bit 3—Reserved: This bit is always read as 0 and should only be written with 0.

Bit 2—PD1 Mode (PD1MD): Selects the function of the PD1/D1 pin.

Bit 2: PD1MD	Description	
0	General input/output (PD1) (D1 in on-chip ROM disabled modes)	(Initial value)
1	Data input/output (D1) (PD1 in single-chip mode)	

Bit 1—Reserved: This bit is always read as 0 and should only be written with 0.

Bit 0—PD0 Mode (PD0MD): Selects the function of the PD0/D0 pin.

Bit 0: PD0MD	Description	
0	General input/output (PD0) (D0 in on-chip ROM disabled modes)	(Initial value)
1	Data input/output (D0) (PD0 in single-chip mode)	

17.3.17 Port E IO Register H (PEIORH)

Bit:	15	14	13	12	11	10	9	8
	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit:	7	6	5	4	3	2	1	0
	PE23IOR	PE22IOR	PE21IOR	PE20IOR	PE19IOR	PE18IOR	PE17IOR	PE16IOR
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

Port E IO register H (PEIORH) is a 16-bit readable/writable register that selects the input/output direction of pins in port E. Bits PE23IOR to PE16IOR correspond to pins PE23/$\overline{\text{IRQ7}}$/PWOB to PE16/$\overline{\text{IRQ0}}$/SCK1/$\overline{\text{AH}}$. PEIORH is enabled when port E pins function as general input/output pins (PE23 to PE16) or as SCI SCK pins, or when PE20 functions as the MMT PCIO pin, and disabled otherwise.

When port E pins function as PE23 to PE16 or as SCI SCK pins, or when PE20 functions as the MMT PCIO pin, a pin becomes an output when the corresponding bit in PEIORH is set to 1, and an input when the bit is cleared to 0.

PEIORH is initialized to H'0000 by an external power-on reset, but is not initialized by a WDT reset, in standby mode, or in sleep mode.

17.3.18 Port E IO Register L (PEIORL)

Bit:	15	14	13	12	11	10	9	8
	PE15IOR	PE14IOR	PE13IOR	PE12IOR	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R	R	R	R

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

Port E IO register L (PEIORL) is a 16-bit readable/writable register that selects the input/output direction of pins in port E. Bits PE15IOR to PE12IOR correspond to pins PE15/$\overline{\text{IRQ7}}$ to PE12/$\overline{\text{IRQ4}}$. PEIORL is enabled when port E pins function as general input/output pins (PE15 to PE12), and disabled otherwise.

When port E pins function as PE15 to PE12, a pin becomes an output when the corresponding bit in PEIORL is set to 1, and an input when the bit is cleared to 0.

PEIORL is initialized to H'0000 by an external power-on reset, but is not initialized by a WDT reset, in standby mode, or in sleep mode.

17.3.19 Port E Control Register H2 (PECRH2)

Bit:	15	14	13	12	11	10	9	8
	PE23 MD1	PE23 MD0	PE22 MD1	PE22 MD0	PE21 MD1	PE21 MD0	PE20 MD1	PE20 MD0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
	PE19 MD1	PE19 MD0	PE18 MD1	PE18 MD0	PE17 MD1	PE17 MD0	PE16 MD1	PE16 MD0
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

Port E control register H2 (PECRH2) is a 16-bit readable/writable register that selects the functions of pins in port E.

PECRH2 selects the functions of port E pins PE23/$\overline{\text{IRQ7}}$/TIOC0C to PE16/$\overline{\text{IRQ0}}$/SCK1/$\overline{\text{AH}}$.

Port E includes a bus control signal ($\overline{\text{AH}}$), but register settings relating to the selection of this pin function may not be valid in all operating modes. For details, see table 17.10, Pin Functions in Each Operating Mode.

PECRH2 is initialized to H'0000 by an external power-on reset, but is not initialized by a WDT reset, in standby mode, or in sleep mode.

Bits 15 and 14—PE23 Mode 1 and 0 (PE23MD1, PE23MD0): These bits select the function of the PE23/$\overline{\text{IRQ7}}$/PWOB pin.

Bit 15: PE23MD1	Bit 14: PE23MD0	Description
0	0	General input/output (PE23) (Initial value)
	1	External interrupt request input ($\overline{\text{IRQ7}}$)
1	0	MMT PWM W-phase output (PWOB)
	1	Reserved (Do not set)

Bits 13 and 12—PE22 Mode 1 and 0 (PE22MD1, PE22MD0): These bits select the function of the PE22/$\overline{\text{IRQ6}}$/PVOB pin.

Bit 13: PE22MD1	Bit 12: PE22MD0	Description
0	0	General input/output (PE22) (Initial value)
	1	External interrupt request input ($\overline{\text{IRQ6}}$)
1	0	MMT PWM V-phase output (PVOB)
	1	Reserved (Do not set)

Bits 11 and 10—PE21 Mode 1 and 0 (PE21MD1, PE21MD0): These bits select the function of the PE21/$\overline{\text{IRQ5}}$/PUOB pin.

Bit 11: PE21MD1	Bit 10: PE21MD0	Description
0	0	General input/output (PE21) (Initial value)
	1	External interrupt request input ($\overline{\text{IRQ5}}$)
1	0	MMT PWM U-phase output (PUOB)
	1	Reserved (Do not set)

Bits 9 and 8—PE20 Mode 1 and 0 (PE20MD1, PE20MD0): These bits select the function of the PE20/$\overline{\text{IRQ4}}$/PCIO pin.

Bit 9: PE20MD1	Bit 8: PE20MD0	Description
0	0	General input/output (PE20) (Initial value)
	1	External interrupt request input ($\overline{\text{IRQ4}}$)
1	0	MMT PWM cycle output (PCO)/counter clear input (PCI)
	1	Reserved (Do not set)

Bits 7 and 6—PE19 Mode 1 and 0 (PE19MD1, PE19MD0): These bits select the function of the PE19/$\overline{\text{IRQ3}}$/PWOA pin.

Bit 7: PE19MD1	Bit 6: PE19MD0	Description
0	0	General input/output (PE19) (Initial value)
	1	External interrupt request input ($\overline{\text{IRQ3}}$)
1	0	MMT PWM W-phase output (PWOA)
	1	Reserved (Do not set)

Bits 5 and 4—PE18 Mode 1 and 0 (PE18MD1, PE18MD0): These bits select the function of the PE18/$\overline{\text{IRQ2}}$/PVOA pin.

Bit 5: PE18MD1	Bit 4: PE18MD0	Description
0	0	General input/output (PE18) (Initial value)
	1	External interrupt request input ($\overline{\text{IRQ2}}$)
1	0	MMT PWM V-phase output (PVOA)
	1	Reserved (Do not set)

Bits 3 and 2—PE17 Mode 1 and 0 (PE17MD1, PE17MD0): These bits select the function of the PE17/$\overline{\text{IRQ1}}$/PUOA/SCK0 pin.

Bit 3: PE17MD1	Bit 2: PE17MD0	Description
0	0	General input/output (PE17) (Initial value)
	1	External interrupt request input ($\overline{\text{IRQ1}}$)
1	0	MMT PWM U-phase output (PUOA)
	1	SCI clock input/output (SCK0)

Bits 1 and 0—PE16 Mode 1 and 0 (PE16MD1, PE16MD0): These bits select the function of the PE16/$\overline{\text{IRQ0}}$/SCK1/$\overline{\text{AH}}$ pin.

Bit 1: PE16MD1	Bit 0: PE16MD0	Description
0	0	General input/output (PE16) (Initial value)
	1	External interrupt request input ($\overline{\text{IRQ0}}$)
1	0	SCI clock input/output (SCK1)
	1	Address hold output ($\overline{\text{AH}}$) (PE16 in single-chip mode)

17.3.20 Port E Control Register L (PECRL)

Bit:	15	14	13	12	11	10	9	8
	—	PE15MD	—	PE14MD	—	PE13MD	—	PE12MD
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R/W	R	R/W	R	R/W	R	R/W

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

Port E control register L (PECRL) is a 16-bit readable/writable register that selects the functions of pins in port E.

PECRL selects the functions of port E pins PE15/$\overline{\text{IRQ7}}$ to PE12/$\overline{\text{IRQ4}}$.

PECRL is initialized to H'0000 by an external power-on reset, but is not initialized by a WDT reset, in standby mode, or in sleep mode.

Bit 15—Reserved: This bit is always read as 0 and should only be written with 0.

Bit 14—PE15 Mode (PE15MD): Selects the function of the PE15/$\overline{\text{IRQ7}}$ pin.

Bit 14: PE15MD	Description
0	General input/output (PE15) (Initial value)
1	External interrupt request input ($\overline{\text{IRQ7}}$)

Bit 13—Reserved: This bit is always read as 0 and should only be written with 0.

Bit 12—PE14 Mode (PE14MD): Selects the function of the PE14/$\overline{\text{IRQ6}}$ pin.

Bit 12: PE14MD	Description	
0	General input/output (PE14)	(Initial value)
1	External interrupt request input ($\overline{\text{IRQ6}}$)	

Bit 11—Reserved: This bit is always read as 0 and should only be written with 0.

Bit 10—PE13 Mode (PE13MD): Selects the function of the PE13/$\overline{\text{IRQ5}}$ pin.

Bit 10: PE13MD	Description	
0	General input/output (PE13)	(Initial value)
1	External interrupt request input ($\overline{\text{IRQ5}}$)	

Bit 9—Reserved: This bit is always read as 0 and should only be written with 0.

Bit 8—PE12 Mode (PE12MD): Selects the function of the PE12/$\overline{\text{IRQ4}}$ pin.

Bit 8: PE12MD	Description	
0	General input/output (PE12)	(Initial value)
1	External interrupt request input ($\overline{\text{IRQ4}}$)	

Bits 7 to 0—Reserved: These bits are always read as 0 and should only be written with 0.

17.3.21 Port F IO Register L (PFIORL)

Bit:	15	14	13	12	11	10	9	8
	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit:	7	6	5	4	3	2	1	0
	PF7IOR	PF6IOR	PF5IOR	—	PF3IOR	PF2IOR	PF1IOR	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R	R/W	R/W	R/W	R

Port F IO register L (PFIORL) is a 16-bit readable/writable register that selects the input/output direction of pins in port F. Bits PF7IOR to PF1IOR correspond to pins PF7/$\overline{\text{DREQ1}}$/$\overline{\text{IRQOUT}}$/TIOC0D to PF1/$\overline{\text{DACK0}}$/TIOC0B. PFIORL is enabled when port F pins function as general input/output pins (PF7 to PF1) or TPU TIOC pins, and disabled otherwise.

When port F pins function as PF7 to PF1 or TPU TIOC pins, a pin becomes an output when the corresponding bit in PFIORL is set to 1, and an input when the bit is cleared to 0.

PFIORL is initialized to H'0000 by an external power-on reset, but is not initialized by a WDT reset, in standby mode, or in sleep mode.

17.3.22 Port F Control Register L2 (PFCRL2)

Bit:	15	14	13	12	11	10	9	8
	PF7 MD1	PF7 MD0	PF6 MD1	PF6 MD0	PF5 MD1	PF5 MD0	—	—
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:	7	6	5	4	3	2	1	0
	PF3 MD1	PF3 MD0	PF2 MD1	PF2 MD0	PF1 MD1	PF1 MD0	—	—
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

Port F control register L2 (PFCRL2) is a 16-bit readable/writable register that selects the functions of pins in port F.

PFCRL2 selects the functions of port F pins PF7/$\overline{\text{DREQ1}}$/$\overline{\text{IRQOUT}}$/TIOC0D to PF1/$\overline{\text{DACK0}}$/TIOC0B.

Port F includes DMAC control signals ($\overline{\text{DREQ0}}$, $\overline{\text{DREQ1}}$, $\overline{\text{DRAK0}}$, $\overline{\text{DRAK1}}$, $\overline{\text{DACK0}}$, and $\overline{\text{DACK1}}$), but register settings relating to the selection of these pin functions may not be valid in all operating modes. For details, see table 17.10, Pin Functions in Each Operating Mode.

PFCRL2 is initialized to H'0000 by an external power-on reset, but is not initialized by a WDT reset, in standby mode, or in sleep mode.

Bits 15 and 14—PF7 Mode 1 and 0 (PF7MD1, PF7MD0): These bits select the function of the PF7/$\overline{\text{DREQ1}}$/$\overline{\text{IRQOUT}}$/TIOC0D pin.

Bit 15: PF7MD1	Bit 14: PF7MD0	Description
0	0	General input/output (PF7) (Initial value)
	1	DMA transfer request input ($\overline{\text{DREQ1}}$) (PF7 in single-chip mode)
1	0	Interrupt request acknowledge output ($\overline{\text{IRQOUT}}$)
	1	TPU input capture input/output compare output (TIOC0D)

Bits 13 and 12—PF6 Mode 1 and 0 (PF6MD1, PF6MD0): These bits select the function of the PF6/$\overline{\text{DRAK1}}$/TxD1/TIOC2A pin.

Bit 13: PF6MD1	Bit 12: PF6MD0	Description
0	0	General input/output (PF6) (Initial value)
	1	DMA transfer request sampling output($\overline{\text{DRAK1}}$) (PF6 in single-chip mode)
1	0	SCI transmit data output (TxD1)
	1	TPU input capture input/output compare output (TIOC2A)

Bits 11 and 10—PF5 Mode 1 and 0 (PF5MD1, PF5MD0): These bits select the function of the PF5/$\overline{\text{DACK1}}$/RxD1/TIOC2B pin.

Bit 11: PF5MD1	Bit 10: PF5MD0	Description
0	0	General input/output (PF5) (Initial value)
	1	DMA transfer request acknowledge output($\overline{\text{DACK1}}$) (PF5 in single-chip mode)
1	0	SCI receive data input (RxD1)
	1	TPU input capture input/output compare output (TIOC2B)

Bits 9 and 8—Reserved: These bits are always read as 0 and should only be written with 0.

Bits 7 and 6—PF3 Mode 1 and 0 (PF3MD1, PF3MD0): These bits select the function of the PF3/$\overline{\text{DREQ0}}$/TIOC0A pin.

Bit 7: PF3MD1	Bit 6: PF3MD0	Description	
0	0	General input/output (PF3)	(Initial value)
	1	DMA transfer request input ($\overline{\text{DREQ0}}$)	
1	0	TPU input capture input/output compare output (TIOC0A)	
	1	Reserved (Do not set)	

Bits 5 and 4—PF2 Mode 1 and 0 (PF2MD1, PF2MD0): These bits select the function of the PF2/$\overline{\text{DRAK0}}$/TIOC0C pin.

Bit 5: PF2MD1	Bit 4: PF2MD0	Description	
0	0	General input/output (PF2)	(Initial value)
	1	DMA transfer request sampling output ($\overline{\text{DRAK0}}$)	
1	0	TPU input capture input/output compare output (TIOC0C)	
	1	Reserved (Do not set)	

Bits 3 and 2—PF1 Mode 1 and 0 (PF1MD1, PF1MD0): These bits select the function of the PF1/$\overline{\text{DACK0}}$/TIOC0B pin.

Bit 3: PF1MD1	Bit 2: PF1MD0	Description	
0	0	General input/output (PF1)	(Initial value)
	1	DMA transfer request acknowledge output ($\overline{\text{DACK0}}$)	
1	0	TPU input capture input/output compare output (TIOC0B)	
	1	Reserved (Do not set)	

Bits 1 and 0—Reserved: These bits are always read as 0 and should only be written with 0.

17.3.23 Port G IO Register (PGIOR)

Bit:	15	14	13	12	11	10	9	8
	PG31IOR	PG30IOR	PG29IOR	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R	R	R	R	R

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

The port G IO register (PGIOR) is a 16-bit readable/writable register that selects the input/output direction of pins in port G. Bits PG31IOR to PG29IOR correspond to pins PG31/RxD2 to PG29/SCK2. PGIOR is enabled when port G pins function as general input/output pins (PG31 to PG29) or when PG29 functions as an SCI SCK pin, and disabled otherwise.

When port G pins function as PG31 to PG29 or when PG29 functions as an SCI SCK pin, a pin becomes an output when the corresponding bit in PGIOR is set to 1, and an input when the bit is cleared to 0.

PGIOR is initialized to H'0000 by an external power-on reset, but is not initialized by a WDT reset, in standby mode, or in sleep mode.

17.3.24 Port G Control Register H1 (PGCRH1)

Bit:	15	14	13	12	11	10	9	8
	—	PG31MD	—	PG30MD	—	PG29MD	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R/W	R	R/W	R	R/W	R	R

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

Port G control register H1 (PGCRH1) is a 16-bit readable/writable register that selects the functions of pins in port G.

PGCRH1 selects the functions of port G pins PG31/RxD2 to PG29/SCK2.

PGCRH1 is initialized to H'0000 by an external power-on reset, but is not initialized by a WDT reset, in standby mode, or in sleep mode.

Bit 15—Reserved: This bit is always read as 0 and should only be written with 0.

Bit 14—PG31 Mode (PG31MD): Selects the function of the PG31/RxD2 pin.

Bit 14: PG31MD	Description	
0	General input/output (PG31)	(Initial value)
1	SCI receive data input (RxD2)	

Bit 13—Reserved: This bit is always read as 0 and should only be written with 0.

Bit 12—PG30 Mode (PG30MD): Selects the function of the PG30/TxD2 pin.

Bit 12: PG30MD	Description	
0	General input/output (PG30)	(Initial value)
1	SCI transmit data output (TxD2)	

Bit 11—Reserved: This bit is always read as 0 and should only be written with 0.

Bit 10—PG29 Mode (PG29MD): Selects the function of the PG29/SCK2 pin.

Bit 10: PG29MD	Description	
0	General input/output (PG29)	(Initial value)
1	SCI clock input/output (SCK2)	

Bits 9 to 0—Reserved: These bits are always read as 0 and should only be written with 0.

17.3.25 Port H IO Register (PHIOR)

Bit:	15	14	13	12	11	10	9	8
	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	PH1IOR	PH0IOR
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W

The port H IO register (PHIOR) is a 16-bit readable/writable register that selects the input/output direction of pins in port H. Bits PH1IOR and PH0IOR correspond to pins PH1/DA1 to PH0/DA0. PHIOR is enabled when port H pins function as general input/output pins (PH1 and PH0), and disabled otherwise.

When port H pins function as PH1 and PH0, a pin becomes an output when the corresponding bit in PHIOR is set to 1, and an input when the bit is cleared to 0.

PHIOR is initialized to H'0000 by an external power-on reset, but is not initialized by a WDT reset, in standby mode, or in sleep mode.

17.3.26 Port H Control Register (PHCR)

Bit:	15	14	13	12	11	10	9	8
	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	—	PH1MD	—	PH0MD
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R	R/W

The port H control register (PHCR) is a 16-bit readable/writable register that selects the functions of pins in port H. PHCR selects the functions of port H pins PH1/DA1 and PH0/DA0.

PHCR is initialized to H'0000 by an external power-on reset, but is not initialized by a WDT reset, in standby mode, or in sleep mode.

Bits 15 to 3—Reserved: These bits are always read as 0 and should only be written with 0.

Bit 2—PH1 Mode (PH1MD): Selects the function of the PH1/DA1 pin.

Bit 2: PH1MD	Description	
0	General input/output (PH1)	(Initial value)
1	D/A converter output (DA1)	

Bit 1—Reserved: This bit is always read as 0 and should only be written with 0.

Bit 0—PH0 Mode (PH0MD): Selects the function of the PH0/DA0 pin.

Bit 0: PH0MD	Description	
0	General input/output (PH0)	(Initial value)
1	D/A converter output (DA0)	

17.3.27 Function Control Register (FCR)

Bit:	15	14	13	12	11	10	9	8
	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	—	SCIMD	IRQMD1	IRQMD0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W

The function control register (FCR) is a 16-bit readable/writable register that is sued to control the $\overline{\text{IRQOUT}}$ output and SCI outputs (SCK0 to SCK2 and TxD0 to TxD2). If the port control register settings specify a function other than $\overline{\text{IRQOUT}}$ or SCI output, the settings in this register do not affect the pin functions.

FCR is initialized to H'0000 by an external power-on reset, but is not initialized by a WDT reset, in standby mode, or in sleep mode.

Bits 15 to 3—Reserved: These bits are always read as 0 and should only be written with 0.

Bit 2—SCI Output Mode (SCIMD): Selects the function of the SCI output pins.

Bit 2: SCIMD	Description	
0	Output by normal CMOS circuit	(Initial value)
1	Output by open-drain circuit	

Bits 1 and 0—IRQOUT Mode 1 and 0 (IRQMD1, IRQMD0): These bits select the function of the $\overline{\text{IRQOUT}}$ pin.

Bit 1: IRQMD1	Bit 0: IRQMD0	Description
0	0	Interrupt request acknowledge output (Initial value)
	1	Refresh signal output
1	0	Interrupt request acknowledge or refresh signal output (depending on the current operating state)
	1	Always high-level output

17.4 PFC Restrictions

Note that the following bug may occur in the pin function controller (PFC).

Bug description and applicable pins

With the pins listed in table 17.12, if a PFC switch is made to the output mode of a specific function and then a PFC switch is made again to other function output, the pin may constantly be fixed at low output.
However, there are no restrictions on PFC switching in input mode selection for any functions.
Also, the output of the specific function involved operates normally even if output is fixed low.

Table 17.12 Applicable Pins and Related Functions

Applicable Pins	Functions Susceptible to Bug Due to PFC Switch	Functions Susceptible to Bug Due to Further PFC Switch after Selection of Function at Left
PC25 to PC14	TPU output compare output	General port output and address output
PD31	TPU output compare output	General port output and data output
PD30, PD29	TPU output compare output	General port output and data output
	SCI TxD2 output and SCK2 output	
PD28, PD27	TPU output compare output	General port output and data output
PD26 to PD24	MMT PWM output	General port output and data output
PD23, PD22	SCI SCK1 output and SCK0 output	General port output and data output
	MMT PWM output and toggle output	
PD21, PD20	TPU output compare output	General port output and data output
PD15 to PD8	TPU output compare output	General port output and data output

Conditions for Occurrence

In the selection of an output function (TPU, MMT, or SCI) susceptible to this bug, as shown in table 17.12, if a switch is made to another output function when a pin output value is low, the pin may become fixed at low output.

Usage Notes

With the applicable pins listed in table 17.12, do not switch to general port output or an address/data output function after TPU, MMT, or SCI output function selection.
Also, if a break is to be transmitted by means of a general port output function when performing serial data transmission from SCI channel 2, execute the break transmission while serial data transmission is not being performed (the TxD2 pin goes to the idle state (high output) while serial data data is not being transmitted, so this bug does not occur).

Section 18 I/O Ports (I/O)

18.1 Overview

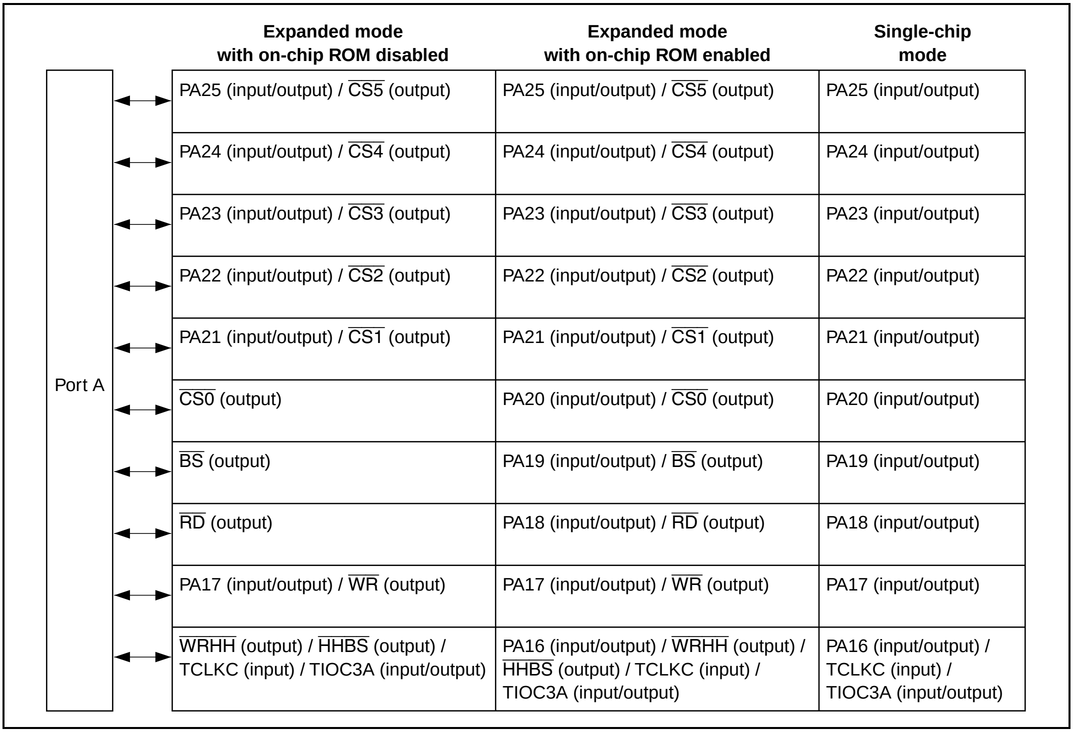
The SH7065 has nine ports: A, B, C, D, E, F, G, H, and I.

All the port pins are multiplexed as general input/output pins (general input pins in the case of port I) and special function pins. The functions of the multiplex pins are selected by means of the pin function controller (PFC). Each port is provided with a data register for storing the pin data.

The initial state of each pin after a power-on reset depends on the operating mode. For details, see table 17.10, Pin Functions in Each Operating Mode.

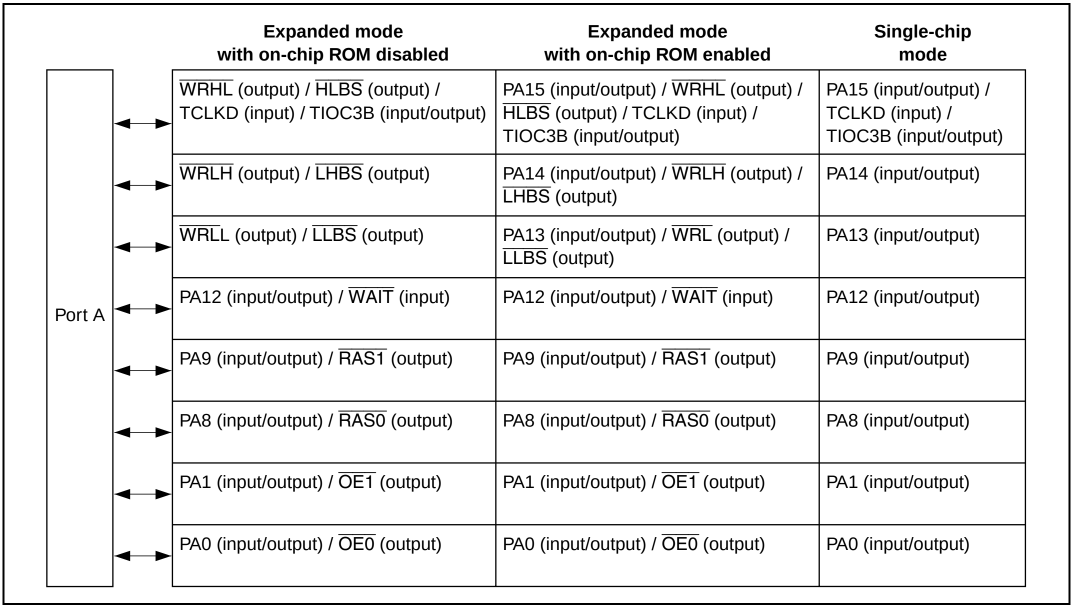
18.2 Port A

Port A is an input/output port with the 18 pins shown in figures 18.1 and 18.2.



Expanded mode with on-chip ROM disabled	Expanded mode with on-chip ROM enabled	Single-chip mode
PA25 (input/output) / $\overline{\text{CS5}}$ (output)	PA25 (input/output) / $\overline{\text{CS5}}$ (output)	PA25 (input/output)
PA24 (input/output) / $\overline{\text{CS4}}$ (output)	PA24 (input/output) / $\overline{\text{CS4}}$ (output)	PA24 (input/output)
PA23 (input/output) / $\overline{\text{CS3}}$ (output)	PA23 (input/output) / $\overline{\text{CS3}}$ (output)	PA23 (input/output)
PA22 (input/output) / $\overline{\text{CS2}}$ (output)	PA22 (input/output) / $\overline{\text{CS2}}$ (output)	PA22 (input/output)
PA21 (input/output) / $\overline{\text{CS1}}$ (output)	PA21 (input/output) / $\overline{\text{CS1}}$ (output)	PA21 (input/output)
$\overline{\text{CS0}}$ (output)	PA20 (input/output) / $\overline{\text{CS0}}$ (output)	PA20 (input/output)
$\overline{\text{BS}}$ (output)	PA19 (input/output) / $\overline{\text{BS}}$ (output)	PA19 (input/output)
$\overline{\text{RD}}$ (output)	PA18 (input/output) / $\overline{\text{RD}}$ (output)	PA18 (input/output)
PA17 (input/output) / $\overline{\text{WR}}$ (output)	PA17 (input/output) / $\overline{\text{WR}}$ (output)	PA17 (input/output)
$\overline{\text{WRHH}}$ (output) / $\overline{\text{HHBS}}$ (output) / TCLKC (input) / TIOC3A (input/output)	PA16 (input/output) / $\overline{\text{WRHH}}$ (output) / $\overline{\text{HHBS}}$ (output) / TCLKC (input) / TIOC3A (input/output)	PA16 (input/output) / TCLKC (input) / TIOC3A (input/output)

Figure 18.1 Port A (PA25 to PA16)



	Expanded mode with on-chip ROM disabled	Expanded mode with on-chip ROM enabled	Single-chip mode
Port A	$\overline{\text{WRHL}}$ (output) / $\overline{\text{HLBS}}$ (output) / TCLKD (input) / TIOC3B (input/output)	PA15 (input/output) / $\overline{\text{WRHL}}$ (output) / $\overline{\text{HLBS}}$ (output) / TCLKD (input) / TIOC3B (input/output)	PA15 (input/output) / TCLKD (input) / TIOC3B (input/output)
	$\overline{\text{WRLH}}$ (output) / $\overline{\text{LHBS}}$ (output)	PA14 (input/output) / $\overline{\text{WRLH}}$ (output) / $\overline{\text{LHBS}}$ (output)	PA14 (input/output)
	$\overline{\text{WRL}}$L (output) / $\overline{\text{LLBS}}$ (output)	PA13 (input/output) / $\overline{\text{WRL}}$ (output) / $\overline{\text{LLBS}}$ (output)	PA13 (input/output)
	PA12 (input/output) / $\overline{\text{WAIT}}$ (input)	PA12 (input/output) / $\overline{\text{WAIT}}$ (input)	PA12 (input/output)
	PA9 (input/output) / $\overline{\text{RAS1}}$ (output)	PA9 (input/output) / $\overline{\text{RAS1}}$ (output)	PA9 (input/output)
	PA8 (input/output) / $\overline{\text{RAS0}}$ (output)	PA8 (input/output) / $\overline{\text{RAS0}}$ (output)	PA8 (input/output)
	PA1 (input/output) / $\overline{\text{OE1}}$ (output)	PA1 (input/output) / $\overline{\text{OE1}}$ (output)	PA1 (input/output)
	PA0 (input/output) / $\overline{\text{OE0}}$ (output)	PA0 (input/output) / $\overline{\text{OE0}}$ (output)	PA0 (input/output)

Figure 18.2 Port A (PA15 to PA0)

18.2.1 Register Configuration

The port A registers are shown in table 18.1.

Table 18.1 Port A Registers

Name	Abbreviation	R/W	Initial Value	Address	Access Size
Port A data register H	PADRH	R/W	H'0000	H'FFFF 1200	8, 16, 32
Port A data register L	PADRL	R/W	H'0000	H'FFFF 1202	8, 16, 32

18.2.2 Port A Data Register H (PADRH)

Bit:	15	14	13	12	11	10	9	8
	—	—	—	—	—	—	PA25DR	PA24DR
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
	PA23DR	PA22DR	PA21DR	PA20DR	PA19DR	PA18DR	PA17DR	PA16DR
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

Port A data register H (PADRH) is a 16-bit readable/writable register that stores port A data. Bits PA25DR to PA16DR correspond to pins PA25/$\overline{\text{CS5}}$ to PA16/$\overline{\text{WRHH}}$/$\overline{\text{HHBS}}$/TCLKC/TIOC3A.

When a pin functions as a general output, if a value is written to PADRH, that value is output directly from the pin, and if PADRH is read, the register value is returned directly regardless of the pin state.

When a pin functions as a general input, if PADRH is read the pin state, not the register value, is returned directly. If a value is written to PADRH, although that value is written into PADRH it does not affect the pin state. Table 18.2 summarizes port A data register read/write operations.

PADRH is initialized by an external power-on reset, but is not initialized by a WDT reset or in standby mode or sleep mode.

18.2.3 Port A Data Register L (PADRL)

Bit:	15	14	13	12	11	10	9	8
	PA15DR	PA14DR	PA13DR	PA12DR	—	—	PA9DR	PA8DR
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R	R	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	PA1DR	PA0DR
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W

Port A data register L (PADRL) is a 16-bit readable/writable register that stores port A data. Bits PA15DR to PA0DR correspond to pins PA15/$\overline{\text{WRHL}}$/$\overline{\text{HLBS}}$/TCLKD/TIOC3B to PA0/$\overline{\text{OE0}}$.

When a pin functions as a general output, if a value is written to PADRL, that value is output directly from the pin, and if PADRL is read, the register value is returned directly regardless of the pin state.

When a pin functions as a general input, if PADRL is read the pin state, not the register value, is returned directly. If a value is written to PADRL, although that value is written into PADRL it does not affect the pin state. Table 18.2 summarizes port A data register read/write operations.

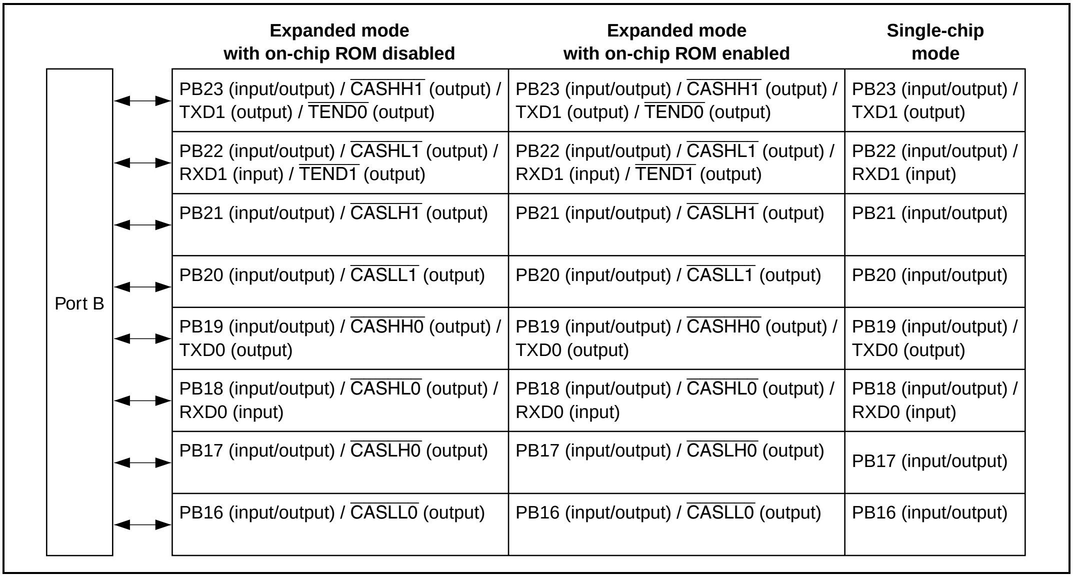
PADRL is initialized by an external power-on reset, but is not initialized by a WDT reset or in standby mode or sleep mode.

Table 18.2 Port A Data Register (PADR) Read/Write Operations

PAIOR	Pin Function	Read	Write
0	General input	Pin state	Value is written to PADR, but does not affect pin state
	Other than general input	Undefined	Value is written to PADR, but does not affect pin state
1	General output	PADR value	Write value is output from pin
	Other than general output	PADR value	Value is written to PADR, but does not affect pin state

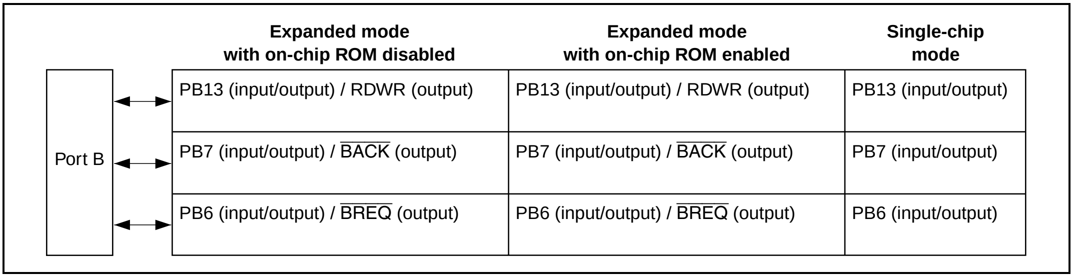
18.3 Port B

Port B is an input/output port with the 11 pins shown in figures 18.3 and 18.4.



	Expanded mode with on-chip ROM disabled	Expanded mode with on-chip ROM enabled	Single-chip mode
Port B	PB23 (input/output) / $\overline{\text{CASHH1}}$ (output) / TXD1 (output) / $\overline{\text{TEND0}}$ (output)	PB23 (input/output) / $\overline{\text{CASHH1}}$ (output) / TXD1 (output) / $\overline{\text{TEND0}}$ (output)	PB23 (input/output) / TXD1 (output)
	PB22 (input/output) / $\overline{\text{CASHL1}}$ (output) / RXD1 (input) / $\overline{\text{TEND1}}$ (output)	PB22 (input/output) / $\overline{\text{CASHL1}}$ (output) / RXD1 (input) / $\overline{\text{TEND1}}$ (output)	PB22 (input/output) / RXD1 (input)
	PB21 (input/output) / $\overline{\text{CASLH1}}$ (output)	PB21 (input/output) / $\overline{\text{CASLH1}}$ (output)	PB21 (input/output)
	PB20 (input/output) / $\overline{\text{CASLL1}}$ (output)	PB20 (input/output) / $\overline{\text{CASLL1}}$ (output)	PB20 (input/output)
	PB19 (input/output) / $\overline{\text{CASHH0}}$ (output) / TXD0 (output)	PB19 (input/output) / $\overline{\text{CASHH0}}$ (output) / TXD0 (output)	PB19 (input/output) / TXD0 (output)
	PB18 (input/output) / $\overline{\text{CASHL0}}$ (output) / RXD0 (input)	PB18 (input/output) / $\overline{\text{CASHL0}}$ (output) / RXD0 (input)	PB18 (input/output) / RXD0 (input)
	PB17 (input/output) / $\overline{\text{CASLH0}}$ (output)	PB17 (input/output) / $\overline{\text{CASLH0}}$ (output)	PB17 (input/output)
	PB16 (input/output) / $\overline{\text{CASLL0}}$ (output)	PB16 (input/output) / $\overline{\text{CASLL0}}$ (output)	PB16 (input/output)

Figure 18.3 Port B (PB23 to PB16)



	Expanded mode with on-chip ROM disabled	Expanded mode with on-chip ROM enabled	Single-chip mode
Port B	PB13 (input/output) / RDWR (output)	PB13 (input/output) / RDWR (output)	PB13 (input/output)
	PB7 (input/output) / $\overline{\text{BACK}}$ (output)	PB7 (input/output) / $\overline{\text{BACK}}$ (output)	PB7 (input/output)
	PB6 (input/output) / $\overline{\text{BREQ}}$ (output)	PB6 (input/output) / $\overline{\text{BREQ}}$ (output)	PB6 (input/output)

Figure 18.4 Port B (PB13, PB7, and PB6)

18.3.1 Register Configuration

The port B registers are shown in table 18.3.

Table 18.3 Port B Registers

Name	Abbreviation	R/W	Initial Value	Address	Access Size
Port B data register H	PBDRH	R/W	H'0000	H'FFFF 1210	8, 16, 32
Port B data register L	PBDRL	R/W	H'0000	H'FFFF 1212	8, 16, 32

18.3.2 Port B Data Register H (PBDRH)

Bit:	15	14	13	12	11	10	9	8
	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit:	7	6	5	4	3	2	1	0
	PB23DR	PB22DR	PB21DR	PB20DR	PB19DR	PB18DR	PB17DR	PB16DR
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

Port B data register H (PBDRH) is a 16-bit readable/writable register that stores port B data. Bits PB23DR to PB16DR correspond to pins PB23/$\overline{\text{CASHH1}}$/TXD1/$\overline{\text{TEND0}}$ to PB16/$\overline{\text{CASLL0}}$.

When a pin functions as a general output, if a value is written to PBDRH, that value is output directly from the pin, and if PBDRH is read, the register value is returned directly regardless of the pin state.

When a pin functions as a general input, if PBDRH is read the pin state, not the register value, is returned directly. If a value is written to PBDRH, although that value is written into PBDRH it does not affect the pin state. Table 18.4 summarizes port B data register read/write operations.

PBDRH is initialized by an external power-on reset, but is not initialized by a WDT reset or in standby mode or sleep mode.

18.3.3 Port B Data Register L (PBDRL)

Bit:	15	14	13	12	11	10	9	8
	—	—	PB13DR	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R	R	R	R	R

Bit:	7	6	5	4	3	2	1	0
	PB7DR	PB6DR	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R	R	R	R	R	R

Port B data register L (PBDRL) is a 16-bit readable/writable register that stores port B data. Bits PB13DR to PB6DR correspond to pins PB13/RDWR to PB6/$\overline{\text{BREQ}}$.

When a pin functions as a general output, if a value is written to PBDRL, that value is output directly from the pin, and if PBDRL is read, the register value is returned directly regardless of the pin state.

When a pin functions as a general input, if PBDRL is read the pin state, not the register value, is returned directly. If a value is written to PBDRL, although that value is written into PBDRL it does not affect the pin state. Table 18.4 summarizes port B data register read/write operations.

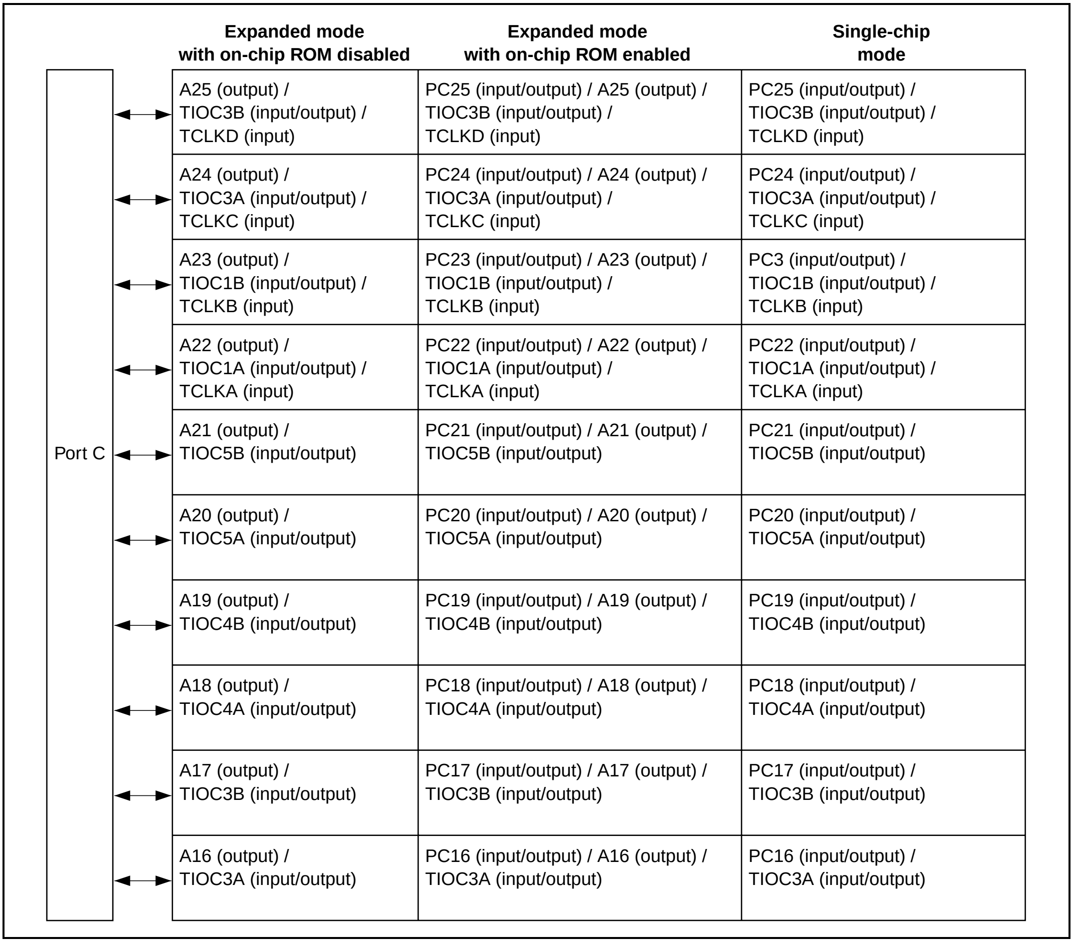
PBDRL is initialized by an external power-on reset, but is not initialized by a WDT reset or in standby mode or sleep mode.

Table 18.4 Port B Data Register (PBDR) Read/Write Operations

PBIOR	Pin Function	Read	Write
0	General input	Pin state	Value is written to PBDR, but does not affect pin state
	Other than general input	Undefined	Value is written to PBDR, but does not affect pin state
1	General output	PBDR value	Write value is output from pin
	Other than general output	PBDR value	Value is written to PBDR, but does not affect pin state

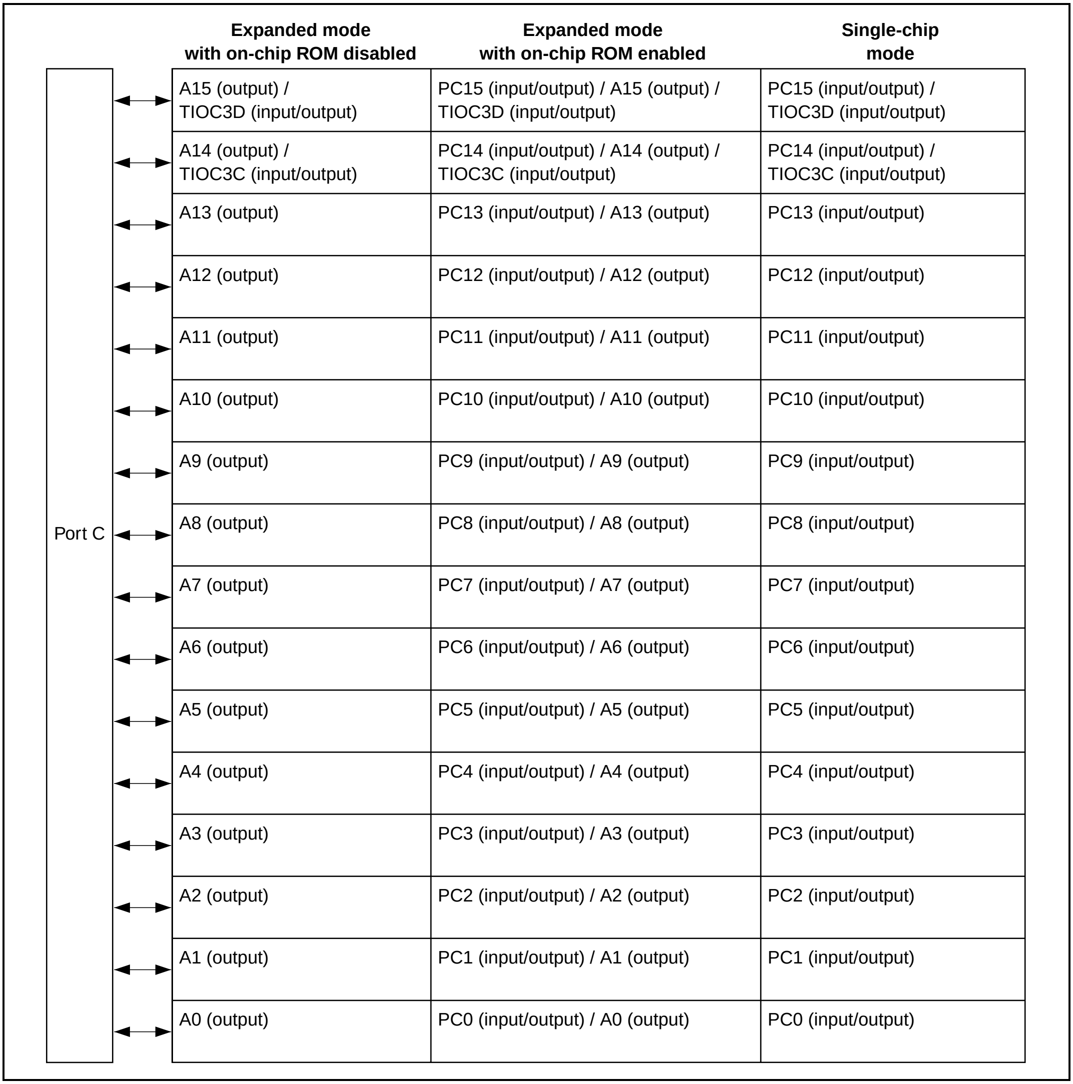
18.4 Port C

Port C is an input/output port with the 26 pins shown in figures 18.5 and 18.6.



	Expanded mode with on-chip ROM disabled	Expanded mode with on-chip ROM enabled	Single-chip mode
Port C	A25 (output) / TIOC3B (input/output) / TCLKD (input)	PC25 (input/output) / A25 (output) / TIOC3B (input/output) / TCLKD (input)	PC25 (input/output) / TIOC3B (input/output) / TCLKD (input)
	A24 (output) / TIOC3A (input/output) / TCLKC (input)	PC24 (input/output) / A24 (output) / TIOC3A (input/output) / TCLKC (input)	PC24 (input/output) / TIOC3A (input/output) / TCLKC (input)
	A23 (output) / TIOC1B (input/output) / TCLKB (input)	PC23 (input/output) / A23 (output) / TIOC1B (input/output) / TCLKB (input)	PC3 (input/output) / TIOC1B (input/output) / TCLKB (input)
	A22 (output) / TIOC1A (input/output) / TCLKA (input)	PC22 (input/output) / A22 (output) / TIOC1A (input/output) / TCLKA (input)	PC22 (input/output) / TIOC1A (input/output) / TCLKA (input)
	A21 (output) / TIOC5B (input/output)	PC21 (input/output) / A21 (output) / TIOC5B (input/output)	PC21 (input/output) / TIOC5B (input/output)
	A20 (output) / TIOC5A (input/output)	PC20 (input/output) / A20 (output) / TIOC5A (input/output)	PC20 (input/output) / TIOC5A (input/output)
	A19 (output) / TIOC4B (input/output)	PC19 (input/output) / A19 (output) / TIOC4B (input/output)	PC19 (input/output) / TIOC4B (input/output)
	A18 (output) / TIOC4A (input/output)	PC18 (input/output) / A18 (output) / TIOC4A (input/output)	PC18 (input/output) / TIOC4A (input/output)
	A17 (output) / TIOC3B (input/output)	PC17 (input/output) / A17 (output) / TIOC3B (input/output)	PC17 (input/output) / TIOC3B (input/output)
	A16 (output) / TIOC3A (input/output)	PC16 (input/output) / A16 (output) / TIOC3A (input/output)	PC16 (input/output) / TIOC3A (input/output)

Figure 18.5 Port C (PC25 to PC16)



Expanded mode with on-chip ROM disabled	Expanded mode with on-chip ROM enabled	Single-chip mode
A15 (output) / TIOC3D (input/output)	PC15 (input/output) / A15 (output) / TIOC3D (input/output)	PC15 (input/output) / TIOC3D (input/output)
A14 (output) / TIOC3C (input/output)	PC14 (input/output) / A14 (output) / TIOC3C (input/output)	PC14 (input/output) / TIOC3C (input/output)
A13 (output)	PC13 (input/output) / A13 (output)	PC13 (input/output)
A12 (output)	PC12 (input/output) / A12 (output)	PC12 (input/output)
A11 (output)	PC11 (input/output) / A11 (output)	PC11 (input/output)
A10 (output)	PC10 (input/output) / A10 (output)	PC10 (input/output)
A9 (output)	PC9 (input/output) / A9 (output)	PC9 (input/output)
A8 (output)	PC8 (input/output) / A8 (output)	PC8 (input/output)
A7 (output)	PC7 (input/output) / A7 (output)	PC7 (input/output)
A6 (output)	PC6 (input/output) / A6 (output)	PC6 (input/output)
A5 (output)	PC5 (input/output) / A5 (output)	PC5 (input/output)
A4 (output)	PC4 (input/output) / A4 (output)	PC4 (input/output)
A3 (output)	PC3 (input/output) / A3 (output)	PC3 (input/output)
A2 (output)	PC2 (input/output) / A2 (output)	PC2 (input/output)
A1 (output)	PC1 (input/output) / A1 (output)	PC1 (input/output)
A0 (output)	PC0 (input/output) / A0 (output)	PC0 (input/output)

Figure 18.6 Port C (PC15 to PC0)

18.4.1 Register Configuration

The port C registers are shown in table 18.5.

Table 18.5 Port C Registers

Name	Abbreviation	R/W	Initial Value	Address	Access Size
Port C data register H	PCDRH	R/W	H'0000	H'FFFF 1220	8, 16, 32
Port C data register L	PCDRL	R/W	H'0000	H'FFFF 1222	8, 16, 32

18.4.2 Port C Data Register H (PCDRH)

Bit:	15	14	13	12	11	10	9	8
	—	—	—	—	—	—	PC25DR	PC24DR
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
	PC23DR	PC22DR	PC21DR	PC20DR	PC19DR	PC18DR	PC17DR	PC16DR
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

Port C data register H (PCDRH) is a 16-bit readable/writable register that stores port C data. Bits PC25DR to PC16DR correspond to pins PC25/A25/TIOC3B/TCLKD to PC16/A16/TIOC3A.

When a pin functions as a general output, if a value is written to PCDRH, that value is output directly from the pin, and if PCDRH is read, the register value is returned directly regardless of the pin state.

When a pin functions as a general input, if PCDRH is read the pin state, not the register value, is returned directly. If a value is written to PCDRH, although that value is written into PCDRH it does not affect the pin state. Table 18.6 summarizes port C data register read/write operations.

PCDRH is initialized by an external power-on reset, but is not initialized by a WDT reset or in standby mode or sleep mode.

18.4.3 Port C Data Register L (PCDRL)

Bit:	15	14	13	12	11	10	9	8
	PC15DR	PC14DR	PC13DR	PC12DR	PC11DR	PC10DR	PC9DR	PC8DR
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
	PC7DR	PC6DR	PC5DR	PC4DR	PC3DR	PC2DR	PC1DR	PC0DR
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

Port C data register L (PCDRL) is a 16-bit readable/writable register that stores port C data. Bits PC15DR to PC0DR correspond to pins PC15/A15/TIOC3D to PC0/A0.

When a pin functions as a general output, if a value is written to PCDRL, that value is output directly from the pin, and if PCDRL is read, the register value is returned directly regardless of the pin state.

When a pin functions as a general input, if PCDRL is read the pin state, not the register value, is returned directly. If a value is written to PCDRL, although that value is written into PCDRL it does not affect the pin state. Table 18.6 summarizes port C data register read/write operations.

PCDRL is initialized by an external power-on reset, but is not initialized by a WDT reset or in standby mode or sleep mode.

Table 18.6 Port C Data Register (PCDR) Read/Write Operations

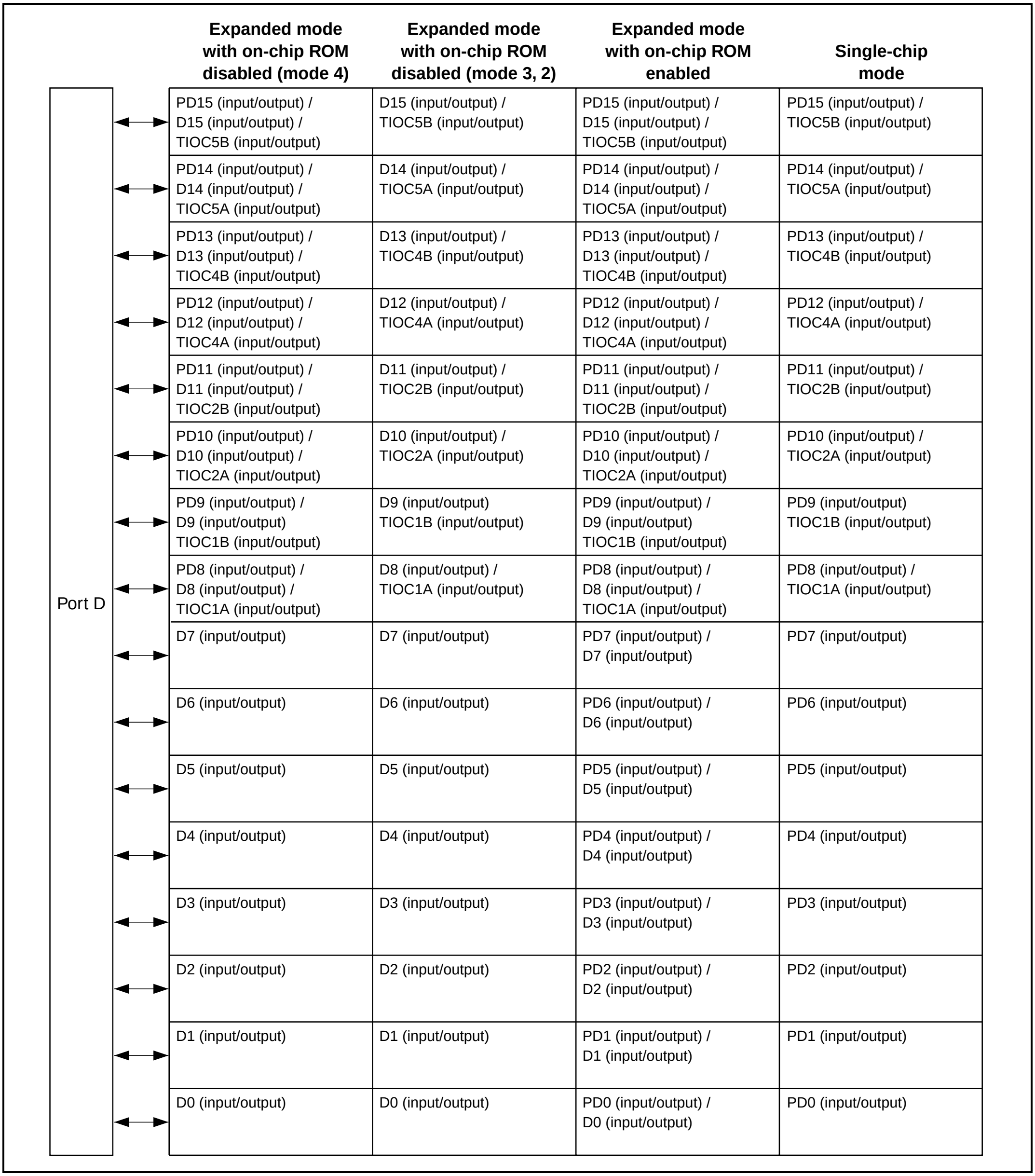
PCIOR	Pin Function	Read	Write
0	General input	Pin state	Value is written to PCDR, but does not affect pin state
	Other than general input	Undefined	Value is written to PCDR, but does not affect pin state
1	General output	PCDR value	Write value is output from pin
	Other than general output	PCDR value	Value is written to PCDR, but does not affect pin state

18.5 Port D

Port D is an input/output port with the 32 pins shown in figures 18.7 and 18.8.

	Expanded mode with on-chip ROM disabled (mode 3, 4)	Expanded mode with on-chip ROM disabled (mode 2)	Expanded mode with on-chip ROM enabled	Single-chip mode
Port D	PD31 (input/output) / D31 (input/output) / RXD2 (input) / TIOC5A (input/output)	PD31 (input/output) / RXD2 (input) / TIOC5A (input/output)	PD31 (input/output) / D31 (input/output) / RXD2 (input) / TIOC5A (input/output)	PD31 (input/output) / RXD2 (input) / TIOC5A (input/output)
	PD30 (input/output) / D30 (input/output) / TXD2 (output) / TIOC4B (input/output)	D30 (input/output) / TXD2 (output) / TIOC4B (input/output)	PD30 (input/output) / D30 (input/output) / TXD2 (output) / TIOC4B (input/output)	PD30 (input/output) / TXD2 (output) / TIOC4B (input/output)
	PD29 (input/output) / D29 (input/output) / SCK2 (input/output) / TIOC4A (input/output)	D29 (input/output) / SCK2 (input/output) / TIOC4A (input/output)	PD29 (input/output) / D29 (input/output) / SCK2 (input/output) / TIOC4A (input/output)	PD29 (input/output) / SCK2 (input/output) / TIOC4A (input/output)
	PD28 (input/output) / D28 (input/output) / TCLKB (input) / TIOC3D (input/output)	D28 (input/output) / TCLKB (input) / TIOC3D (input/output)	PD28 (input/output) / D28 (input/output) / TCLKB (input) / TIOC3D (input/output)	PD28 (input/output) / TCLKB (input) / TIOC3D (input/output)
	PD27 (input/output) / D27 (input/output) / TCLKA (input) / TIOC3C (input/output)	D27 (input/output) / TCLKA (input) / TIOC3C (input/output)	PD27 (input/output) / D27 (input/output) / TCLKA (input) / TIOC3C (input/output)	PD27 (input/output) / TCLKA (input) / TIOC3C (input/output)
	PD26 (input/output) / D26 (input/output) / PWOB (output)	D26 (input/output) / PWOB (input)	PD26 (input/output) / D26 (input/output) / PWOB (output)	PD26 (input/output) / PWOB (input)
	PD25 (input/output) / D25 (input/output) / PVOB (output)	D25 (input/output) / PVOB (output)	PD25 (input/output) / D25 (input/output) / PVOB (output)	PD25 (input/output) / PVOB (output)
	PD24 (input/output) / D24 (input/output) / PUOB (output)	D24 (input/output) / PUOB (output)	PD24 (input/output) / D24 (input/output) / PUOB (output)	PD24 (input/output) / PUOB (output)
	PD23 (input/output) / D23 (input/output) / PCIO (input/output) / SCK1 (input/output)	D23 (input/output) / PCIO (input/output) / SCK1 (input/output)	PD23 (input/output) / D23 (input/output) / PCIO (input/output) / SCK1 (input/output)	PD23 (input/output) / PCIO (input/output) / SCK1 (input/output)
	PD22 (input/output) / D22 (input/output) / PWOA (output) / SCK0 (input/output)	D22 (input/output) / PWOA (output) / SCK0 (input/output)	PD22 (input/output) / D22 (input/output) / PWOA (output) / SCK0 (input/output)	PD22 (input/output) / PWOA (output) / SCK0 (input/output)
	PD21 (input/output) / D21 (input/output) / PVOA (output) / $\overline{\text{IRQ7}}$ (input)	D21 (input/output) / PVOA (output) / $\overline{\text{IRQ7}}$ (input)	PD21 (input/output) / D21 (input/output) / PVOA (output) / $\overline{\text{IRQ7}}$ (input)	PD21 (input/output) / PVOA (output) / $\overline{\text{IRQ7}}$ (input)
	PD20 (input/output) / D20 (input/output) / PUOA (output) / $\overline{\text{IRQ6}}$ (input)	D20 (input/output) / PUOA (output) / $\overline{\text{IRQ6}}$ (input)	PD20 (input/output) / D20 (input/output) / PUOA (output) / $\overline{\text{IRQ6}}$ (input)	PD20 (input/output) / PUOA (output) / $\overline{\text{IRQ6}}$ (input)
	PD19 (input/output) / D19 (input/output) / $\overline{\text{POE3}}$ (input) / $\overline{\text{IRQ5}}$ (input)	D19 (input/output) / $\overline{\text{POE3}}$ (input) / $\overline{\text{IRQ5}}$ (input)	PD19 (input/output) / D19 (input/output) / $\overline{\text{POE3}}$ (input) / $\overline{\text{IRQ5}}$ (input)	PD19 (input/output) / $\overline{\text{POE3}}$ (input) / $\overline{\text{IRQ5}}$ (input)
	PD18 (input/output) / D18 (input/output) / $\overline{\text{POE2}}$ (input) / $\overline{\text{IRQ4}}$ (input)	D18 (input/output) / $\overline{\text{POE2}}$ (input) / $\overline{\text{IRQ4}}$ (input)	PD18 (input/output) / D18 (input/output) / $\overline{\text{POE2}}$ (input) / $\overline{\text{IRQ4}}$ (input)	PD18 (input/output) / $\overline{\text{POE2}}$ (input) / $\overline{\text{IRQ4}}$ (input)
	PD17 (input/output) / D17 (input/output) / $\overline{\text{POE1}}$ (input) / $\overline{\text{ADTRG}}$ (input)	D17 (input/output) / $\overline{\text{POE1}}$ (input) / $\overline{\text{ADTRG}}$ (input)	PD17 (input/output) / D17 (input/output) / $\overline{\text{POE1}}$ (input) / $\overline{\text{ADTRG}}$ (input)	PD17 (input/output) / $\overline{\text{POE1}}$ (input) / $\overline{\text{ADTRG}}$ (input)
	PD16 (input/output) / D16 (input/output) / $\overline{\text{POE0}}$ (input)	D16 (input/output) / $\overline{\text{POE0}}$ (input)	PD16 (input/output) / D16 (input/output) / $\overline{\text{POE0}}$ (input)	PD16 (input/output) / $\overline{\text{POE0}}$ (input)

Figure 18.7 Port D (PD31 to PD16)



	Expanded mode with on-chip ROM disabled (mode 4)	Expanded mode with on-chip ROM disabled (mode 3, 2)	Expanded mode with on-chip ROM enabled	Single-chip mode
Port D	PD15 (input/output) / D15 (input/output) / TIOC5B (input/output)	D15 (input/output) / TIOC5B (input/output)	PD15 (input/output) / D15 (input/output) / TIOC5B (input/output)	PD15 (input/output) / TIOC5B (input/output)
	PD14 (input/output) / D14 (input/output) / TIOC5A (input/output)	D14 (input/output) / TIOC5A (input/output)	PD14 (input/output) / D14 (input/output) / TIOC5A (input/output)	PD14 (input/output) / TIOC5A (input/output)
	PD13 (input/output) / D13 (input/output) / TIOC4B (input/output)	D13 (input/output) / TIOC4B (input/output)	PD13 (input/output) / D13 (input/output) / TIOC4B (input/output)	PD13 (input/output) / TIOC4B (input/output)
	PD12 (input/output) / D12 (input/output) / TIOC4A (input/output)	D12 (input/output) / TIOC4A (input/output)	PD12 (input/output) / D12 (input/output) / TIOC4A (input/output)	PD12 (input/output) / TIOC4A (input/output)
	PD11 (input/output) / D11 (input/output) / TIOC2B (input/output)	D11 (input/output) / TIOC2B (input/output)	PD11 (input/output) / D11 (input/output) / TIOC2B (input/output)	PD11 (input/output) / TIOC2B (input/output)
	PD10 (input/output) / D10 (input/output) / TIOC2A (input/output)	D10 (input/output) / TIOC2A (input/output)	PD10 (input/output) / D10 (input/output) / TIOC2A (input/output)	PD10 (input/output) / TIOC2A (input/output)
	PD9 (input/output) / D9 (input/output) TIOC1B (input/output)	D9 (input/output) TIOC1B (input/output)	PD9 (input/output) / D9 (input/output) TIOC1B (input/output)	PD9 (input/output) TIOC1B (input/output)
	PD8 (input/output) / D8 (input/output) / TIOC1A (input/output)	D8 (input/output) / TIOC1A (input/output)	PD8 (input/output) / D8 (input/output) / TIOC1A (input/output)	PD8 (input/output) / TIOC1A (input/output)
	D7 (input/output)	D7 (input/output)	PD7 (input/output) / D7 (input/output)	PD7 (input/output)
	D6 (input/output)	D6 (input/output)	PD6 (input/output) / D6 (input/output)	PD6 (input/output)
	D5 (input/output)	D5 (input/output)	PD5 (input/output) / D5 (input/output)	PD5 (input/output)
	D4 (input/output)	D4 (input/output)	PD4 (input/output) / D4 (input/output)	PD4 (input/output)
	D3 (input/output)	D3 (input/output)	PD3 (input/output) / D3 (input/output)	PD3 (input/output)
	D2 (input/output)	D2 (input/output)	PD2 (input/output) / D2 (input/output)	PD2 (input/output)
	D1 (input/output)	D1 (input/output)	PD1 (input/output) / D1 (input/output)	PD1 (input/output)
	D0 (input/output)	D0 (input/output)	PD0 (input/output) / D0 (input/output)	PD0 (input/output)

Figure 18.8 Port D (PD15 to PD0)

18.5.1 Register Configuration

The port D registers are shown in table 18.7.

Table 18.7 Port D Registers

Name	Abbreviation	R/W	Initial Value	Address	Access Size
Port D data register H	PDDRH	R/W	H'0000	H'FFFF 1230	8, 16, 32
Port D data register L	PDDRL	R/W	H'0000	H'FFFF 1232	8, 16, 32

18.5.2 Port D Data Register H (PDDRH)

Bit:	15	14	13	12	11	10	9	8
	PD31DR	PD30DR	PD29DR	PD28DR	PD27DR	PD26DR	PD25DR	PD24DR
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
	PD23DR	PD22DR	PD21DR	PD20DR	PD19DR	PD18DR	PD17DR	PD16DR
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

Port D data register H (PDDRH) is a 16-bit readable/writable register that stores port D data. Bits PD31DR to PD16DR correspond to pins PD31/D31/RXD2/TIOC5A to PD16/D16/$\overline{\text{POE0}}$.

When a pin functions as a general output, if a value is written to PDDRH, that value is output directly from the pin, and if PDDRH is read, the register value is returned directly regardless of the pin state.

When a pin functions as a general input, if PDDRH is read the pin state, not the register value, is returned directly. If a value is written to PDDRH, although that value is written into PDDRH it does not affect the pin state. Table 18.8 summarizes port D data register read/write operations.

PDDRH is initialized by an external power-on reset, but is not initialized by a WDT reset or in standby mode or sleep mode.

18.5.3 Port D Data Register L (PDDRL)

Bit:	15	14	13	12	11	10	9	8
	PD15DR	PD14DR	PD13DR	PD12DR	PD11DR	PD10DR	PD9DR	PD8DR
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
	PD7DR	PD6DR	PD5DR	PD4DR	PD3DR	PD2DR	PD1DR	PD0DR
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

Port D data register L (PDDRL) is a 16-bit readable/writable register that stores port D data. Bits PD15DR to PD0DR correspond to pins PD15/D15/TIOC5B to PD0/D0.

When a pin functions as a general output, if a value is written to PDDRL, that value is output directly from the pin, and if PDDRL is read, the register value is returned directly regardless of the pin state.

When a pin functions as a general input, if PDDRL is read the pin state, not the register value, is returned directly. If a value is written to PDDRL, although that value is written into PDDRL it does not affect the pin state. Table 18.8 summarizes port D data register read/write operations.

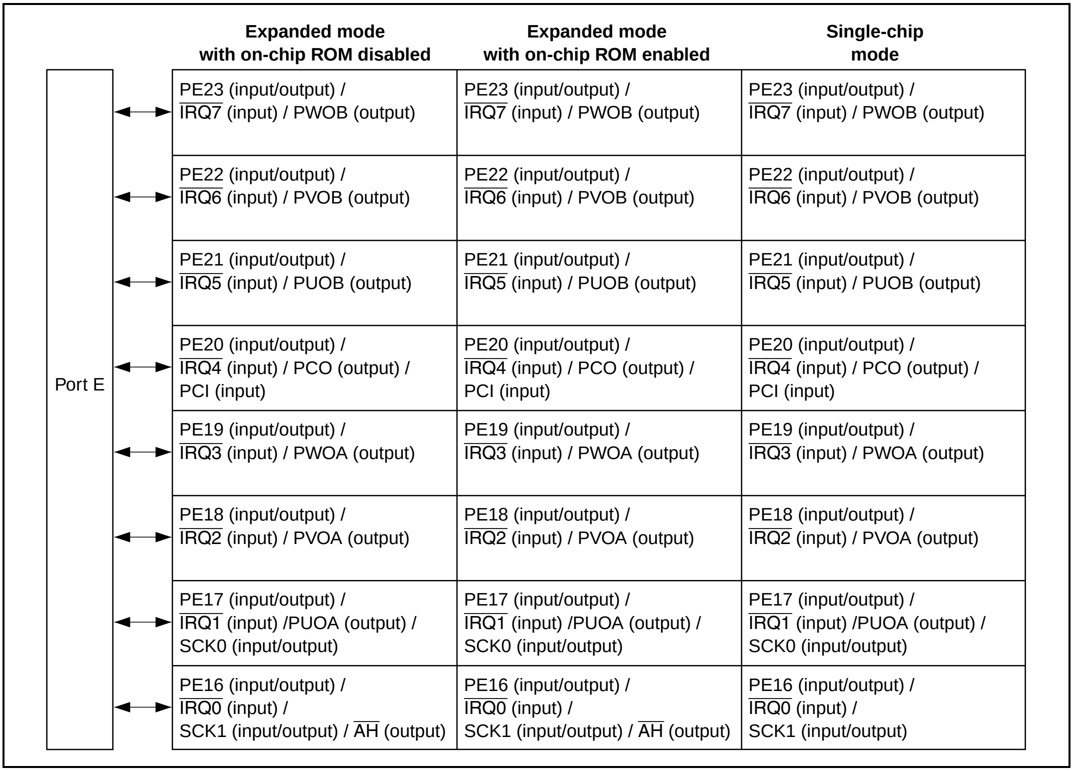
PDDRL is initialized by an external power-on reset, but is not initialized by a WDT reset or in standby mode or sleep mode.

Table 18.8 Port D Data Register (PDDR) Read/Write Operations

PDIOR	Pin Function	Read	Write
0	General input	Pin state	Value is written to PDDR, but does not affect pin state
	Other than general input	Undefined	Value is written to PDDR, but does not affect pin state
1	General output	PDDR value	Write value is output from pin
	Other than general output	PDDR value	Value is written to PDDR, but does not affect pin state

18.6 Port E

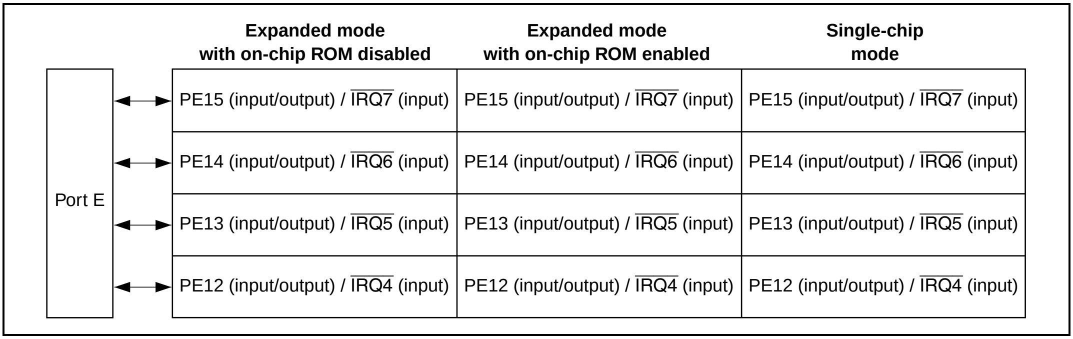
Port E is an input/output port with the 12 pins shown in figures 18.9 and 18.10.



Port E

Expanded mode with on-chip ROM disabled	Expanded mode with on-chip ROM enabled	Single-chip mode
PE23 (input/output) / $\overline{\text{IRQ7}}$ (input) / PWOB (output)	PE23 (input/output) / $\overline{\text{IRQ7}}$ (input) / PWOB (output)	PE23 (input/output) / $\overline{\text{IRQ7}}$ (input) / PWOB (output)
PE22 (input/output) / $\overline{\text{IRQ6}}$ (input) / PVOB (output)	PE22 (input/output) / $\overline{\text{IRQ6}}$ (input) / PVOB (output)	PE22 (input/output) / $\overline{\text{IRQ6}}$ (input) / PVOB (output)
PE21 (input/output) / $\overline{\text{IRQ5}}$ (input) / PUOB (output)	PE21 (input/output) / $\overline{\text{IRQ5}}$ (input) / PUOB (output)	PE21 (input/output) / $\overline{\text{IRQ5}}$ (input) / PUOB (output)
PE20 (input/output) / $\overline{\text{IRQ4}}$ (input) / PCO (output) / PCI (input)	PE20 (input/output) / $\overline{\text{IRQ4}}$ (input) / PCO (output) / PCI (input)	PE20 (input/output) / $\overline{\text{IRQ4}}$ (input) / PCO (output) / PCI (input)
PE19 (input/output) / $\overline{\text{IRQ3}}$ (input) / PWOA (output)	PE19 (input/output) / $\overline{\text{IRQ3}}$ (input) / PWOA (output)	PE19 (input/output) / $\overline{\text{IRQ3}}$ (input) / PWOA (output)
PE18 (input/output) / $\overline{\text{IRQ2}}$ (input) / PVOA (output)	PE18 (input/output) / $\overline{\text{IRQ2}}$ (input) / PVOA (output)	PE18 (input/output) / $\overline{\text{IRQ2}}$ (input) / PVOA (output)
PE17 (input/output) / $\overline{\text{IRQ1}}$ (input) /PUOA (output) / SCK0 (input/output)	PE17 (input/output) / $\overline{\text{IRQ1}}$ (input) /PUOA (output) / SCK0 (input/output)	PE17 (input/output) / $\overline{\text{IRQ1}}$ (input) /PUOA (output) / SCK0 (input/output)
PE16 (input/output) / $\overline{\text{IRQ0}}$ (input) / SCK1 (input/output) / $\overline{\text{AH}}$ (output)	PE16 (input/output) / $\overline{\text{IRQ0}}$ (input) / SCK1 (input/output) / $\overline{\text{AH}}$ (output)	PE16 (input/output) / $\overline{\text{IRQ0}}$ (input) / SCK1 (input/output)

Figure 18.9 Port E (PE23 to PE16)



Port E

Expanded mode with on-chip ROM disabled	Expanded mode with on-chip ROM enabled	Single-chip mode
PE15 (input/output) / $\overline{\text{IRQ7}}$ (input)	PE15 (input/output) / $\overline{\text{IRQ7}}$ (input)	PE15 (input/output) / $\overline{\text{IRQ7}}$ (input)
PE14 (input/output) / $\overline{\text{IRQ6}}$ (input)	PE14 (input/output) / $\overline{\text{IRQ6}}$ (input)	PE14 (input/output) / $\overline{\text{IRQ6}}$ (input)
PE13 (input/output) / $\overline{\text{IRQ5}}$ (input)	PE13 (input/output) / $\overline{\text{IRQ5}}$ (input)	PE13 (input/output) / $\overline{\text{IRQ5}}$ (input)
PE12 (input/output) / $\overline{\text{IRQ4}}$ (input)	PE12 (input/output) / $\overline{\text{IRQ4}}$ (input)	PE12 (input/output) / $\overline{\text{IRQ4}}$ (input)

Figure 18.10 Port E (PE15 to PE12)

18.6.1 Register Configuration

The port E registers are shown in table 18.9.

Table 18.9 Port E Registers

Name	Abbreviation	R/W	Initial Value	Address	Access Size
Port E data register H	PEDRH	R/W	H'0000	H'FFFF 1240	8, 16, 32
Port E data register L	PEDRL	R/W	H'0000	H'FFFF 1242	8, 16, 32

18.6.2 Port E Data Register H (PEDRH)

Bit:	15	14	13	12	11	10	9	8
	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit:	7	6	5	4	3	2	1	0
	PE23DR	PE22DR	PE21DR	PE20DR	PE19DR	PE18DR	PE17DR	PE16DR
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

Port E data register H (PEDRH) is a 16-bit readable/writable register that stores port E data. Bits PE23DR to PE16DR correspond to pins PE23/$\overline{\text{IRQ7}}$/PWOB to PE16/$\overline{\text{IRQ0}}$/SCK0/$\overline{\text{AH}}$.

When a pin functions as a general output, if a value is written to PEDRH, that value is output directly from the pin, and if PEDRH is read, the register value is returned directly regardless of the pin state.

When a pin functions as a general input, if PEDRH is read the pin state, not the register value, is returned directly. If a value is written to PEDRH, although that value is written into PEDRH it does not affect the pin state. Table 18.10 summarizes port E data register read/write operations.

PEDRH is initialized by an external power-on reset, but is not initialized by a WDT reset or in standby mode or sleep mode.

18.6.3 Port E Data Register L (PEDRL)

Bit:	15	14	13	12	11	10	9	8
	PE15DR	PE14DR	PE13DR	PE12DR	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R	R	R	R

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

Port E data register L (PEDRL) is a 16-bit readable/writable register that stores port E data. Bits PE15DR to PE12DR correspond to pins PE15/$\overline{\text{IRQ7}}$ to PE12/$\overline{\text{IRQ4}}$.

When a pin functions as a general output, if a value is written to PEDRL, that value is output directly from the pin, and if PEDRL is read, the register value is returned directly regardless of the pin state.

When a pin functions as a general input, if PEDRL is read the pin state, not the register value, is returned directly. If a value is written to PEDRL, although that value is written into PEDRL it does not affect the pin state. Table 18.10 summarizes port E data register read/write operations.

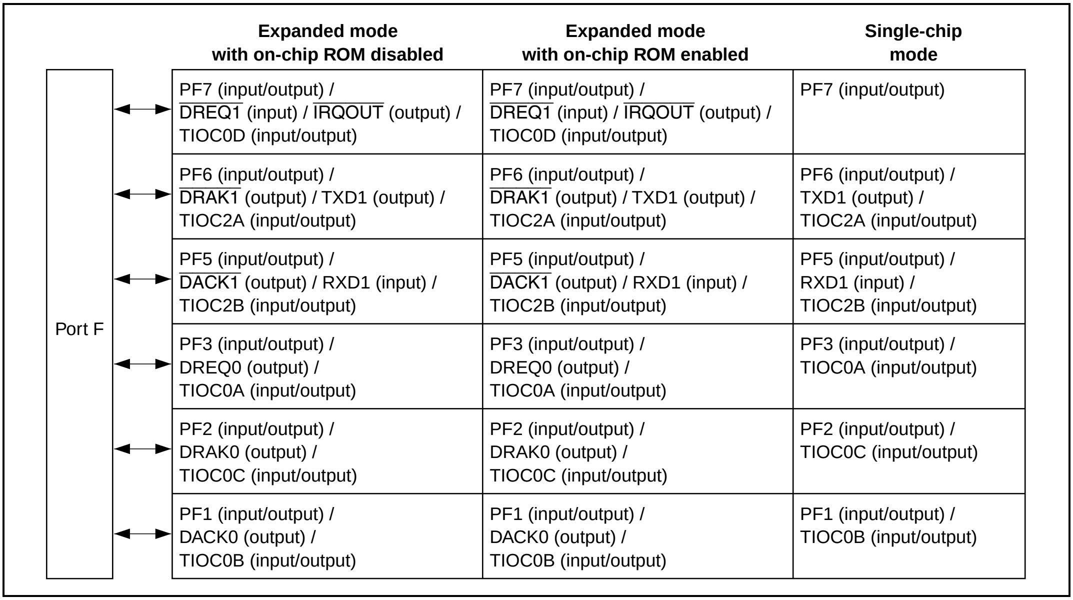
PEDRL is initialized by an external power-on reset, but is not initialized by a WDT reset or in standby mode or sleep mode.

Table 18.10 Port E Data Register (PEDR) Read/Write Operations

PEIOR	Pin Function	Read	Write
0	General input	Pin state	Value is written to PEDR, but does not affect pin state
	Other than general input	Undefined	Value is written to PEDR, but does not affect pin state
1	General output	PEDR value	Write value is output from pin
	Other than general output	PEDR value	Value is written to PEDR, but does not affect pin state

18.7 Port F

Port F is an input/output port with the 6 pins shown in figure 18.11.



	Expanded mode with on-chip ROM disabled	Expanded mode with on-chip ROM enabled	Single-chip mode
Port F	PF7 (input/output) / $\overline{\text{DREQ1}}$ (input) / $\overline{\text{IRQOUT}}$ (output) / TIOC0D (input/output)	PF7 (input/output) / $\overline{\text{DREQ1}}$ (input) / $\overline{\text{IRQOUT}}$ (output) / TIOC0D (input/output)	PF7 (input/output)
	PF6 (input/output) / $\overline{\text{DRAK1}}$ (output) / TXD1 (output) / TIOC2A (input/output)	PF6 (input/output) / $\overline{\text{DRAK1}}$ (output) / TXD1 (output) / TIOC2A (input/output)	PF6 (input/output) / TXD1 (output) / TIOC2A (input/output)
	PF5 (input/output) / $\overline{\text{DACK1}}$ (output) / RXD1 (input) / TIOC2B (input/output)	PF5 (input/output) / $\overline{\text{DACK1}}$ (output) / RXD1 (input) / TIOC2B (input/output)	PF5 (input/output) / RXD1 (input) / TIOC2B (input/output)
	PF3 (input/output) / DREQ0 (output) / TIOC0A (input/output)	PF3 (input/output) / DREQ0 (output) / TIOC0A (input/output)	PF3 (input/output) / TIOC0A (input/output)
	PF2 (input/output) / DRAK0 (output) / TIOC0C (input/output)	PF2 (input/output) / DRAK0 (output) / TIOC0C (input/output)	PF2 (input/output) / TIOC0C (input/output)
	PF1 (input/output) / DACK0 (output) / TIOC0B (input/output)	PF1 (input/output) / DACK0 (output) / TIOC0B (input/output)	PF1 (input/output) / TIOC0B (input/output)

Figure 18.11 Port F (PF7 to PF1)

18.7.1 Register Configuration

The port F register is shown in table 18.11.

Table 18.11 Port F Register

Name	Abbreviation	R/W	Initial Value	Address	Access Size
Port F data register L	PFDRL	R/W	H'0000	H'FFFF 1262	8, 16, 32

18.7.2 Port F Data Register L (PFDRL)

Bit:	15	14	13	12	11	10	9	8
	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit:	7	6	5	4	3	2	1	0
	PF7DR	PF6DR	PF5DR	—	PF3DR	PF2DR	PF1DR	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R	R/W	R/W	R/W	R

Port F data register L (PFDRL) is a 16-bit readable/writable register that stores port F data. Bits PF7DR to PF1DR correspond to pins PF7/$\overline{\text{DREQ1}}$/$\overline{\text{IRQOUT}}$/TIOC0D to PF1/$\overline{\text{DACK0}}$/TIOC0B.

When a pin functions as a general output, if a value is written to PFDRL, that value is output directly from the pin, and if PFDRL is read, the register value is returned directly regardless of the pin state.

When a pin functions as a general input, if PFDRL is read the pin state, not the register value, is returned directly. If a value is written to PFDRL, although that value is written into PFDRL it does not affect the pin state. Table 18.12 summarizes port F data register read/write operations.

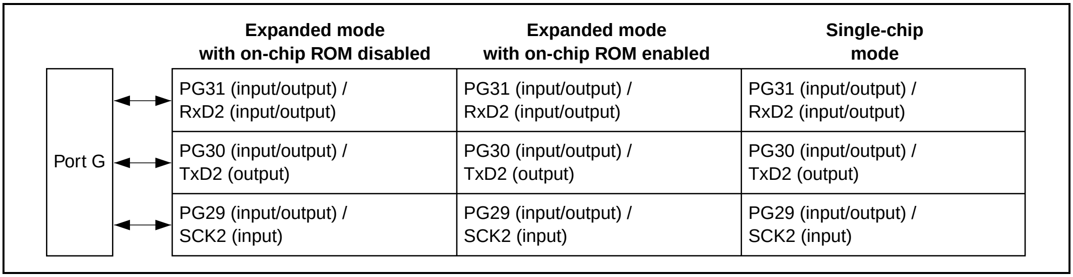
PFDRL is initialized by an external power-on reset, but is not initialized by a WDT reset or in standby mode or sleep mode.

Table 18.12 Port F Data Register (PFDR) Read/Write Operations

PFIOR	Pin Function	Read	Write
0	General input	Pin state	Value is written to PFDR, but does not affect pin state
	Other than general input	Undefined	Value is written to PFDR, but does not affect pin state
1	General output	PFDR value	Write value is output from pin
	Other than general output	PFDR value	Value is written to PFDR, but does not affect pin state

18.8 Port G

Port G is an input/output port with the 3 pins shown in figure 18.12.



	Expanded mode with on-chip ROM disabled	Expanded mode with on-chip ROM enabled	Single-chip mode
Port G	PG31 (input/output) / RxD2 (input/output)	PG31 (input/output) / RxD2 (input/output)	PG31 (input/output) / RxD2 (input/output)
	PG30 (input/output) / TxD2 (output)	PG30 (input/output) / TxD2 (output)	PG30 (input/output) / TxD2 (output)
	PG29 (input/output) / SCK2 (input)	PG29 (input/output) / SCK2 (input)	PG29 (input/output) / SCK2 (input)

Figure 18.12 Port G (PG31 to PG29)

18.8.1 Register Configuration

The port G register is shown in table 18.13.

Table 18.13 Port G Register

Name	Abbreviation	R/W	Initial Value	Address	Access Size
Port G data register H	PGDRH	R/W	H'0000	H'FFFF 1270	8, 16, 32

18.8.2 Port G Data Register H (PGDRH)

Bit:	15	14	13	12	11	10	9	8
	PG31DR	PG30DR	PG29DR	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R	R	R	R	R

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

Port G data register H (PGDRH) is a 16-bit readable/writable register that stores port G data. Bits PG31DR to PG29DR correspond to pins PG31/RxD2 to PG29/SCK1.

When a pin functions as a general output, if a value is written to PGDRH, that value is output directly from the pin, and if PGDRH is read, the register value is returned directly regardless of the pin state.

When a pin functions as a general input, if PGDRH is read the pin state, not the register value, is returned directly. If a value is written to PGDRH, although that value is written into PGDRH it does not affect the pin state. Table 18.14 summarizes port G data register read/write operations.

PGDRH is initialized by an external power-on reset, but is not initialized by a WDT reset or in standby mode or sleep mode.

Table 18.14 Port G Data Register (PGDR) Read/Write Operations

PGIOR	Pin Function	Read	Write
0	General input	Pin state	Value is written to PGDR, but does not affect pin state
	Other than general input	Undefined	Value is written to PGDR, but does not affect pin state
1	General output	PGDR value	Write value is output from pin
	Other than general output	PGDR value	Value is written to PGDR, but does not affect pin state

18.9 Port H

Port H is an input/output port with the 2 pins shown in figure 18.13.

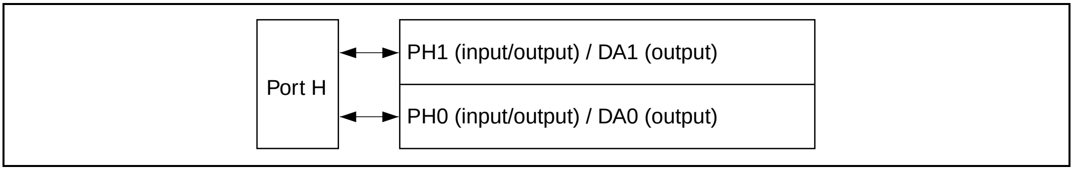

Port H
PH1 (input/output) / DA1 (output)
PH0 (input/output) / DA0 (output)

Figure 18.13 Port H (PH1 and PH0)

18.9.1 Register Configuration

The port H register is shown in table 18.15.

Table 18.15 Port H Register

Name	Abbreviation	R/W	Initial Value	Address	Access Size
Port H data register	PHDR	R/W	H'0000	H'FFFF 1282	8

18.9.2 Port H Data Register (PHDR)

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	PH1DR	PH0DR
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W

The port H data register (PHDR) is an 8-bit readable/writable register that stores port H data. Bits PH1DR and PH0DR correspond to pins PH1/DA1 and PH0/DA0.

When a pin functions as a general output, if a value is written to PHDR, that value is output directly from the pin, and if PHDR is read, the register value is returned directly regardless of the pin state.

When a pin functions as a general input, if PHDR is read the pin state, not the register value, is returned directly. If a value is written to PHDR, although that value is written into PHDR it does not affect the pin state. Table 18.16 summarizes port H data register read/write operations.

PHDR is initialized by an external power-on reset, but is not initialized by a WDT reset or in standby mode or sleep mode.

Table 18.16 Port H Data Register (PHDR) Read/Write Operations

PHIOR	Pin Function	Read	Write
0	General input	Pin state	Value is written to PHDR, but does not affect pin state
	Other than general input	Undefined	Value is written to PHDR, but does not affect pin state
1	General output	PHDR value	Write value is output from pin
	Other than general output	PHDR value	Value is written to PHDR, but does not affect pin state

18.10 Port I

Port I is an input port with the 8 pins shown in figure 18.14.

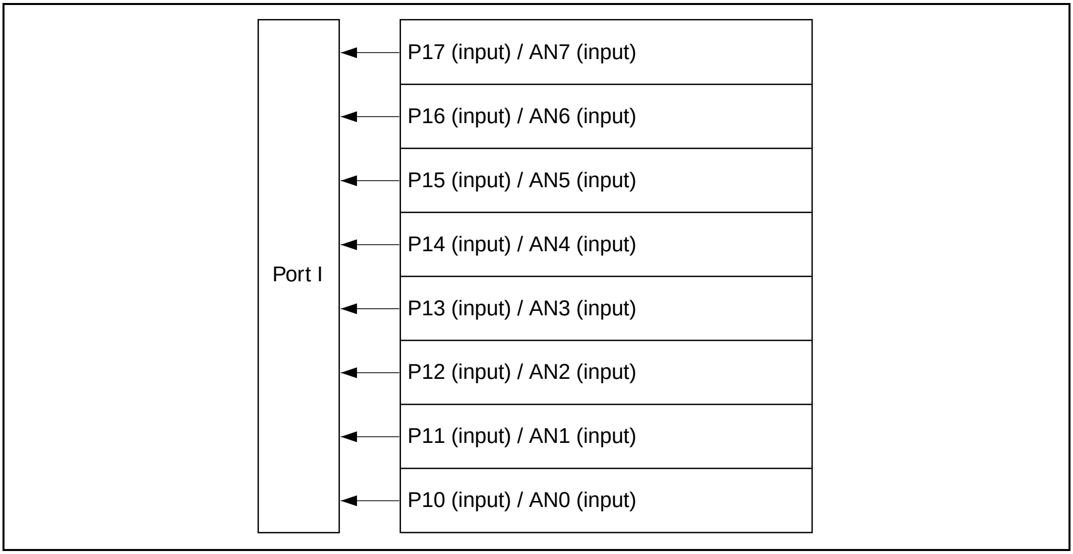

Port I
P17 (input) / AN7 (input)
P16 (input) / AN6 (input)
P15 (input) / AN5 (input)
P14 (input) / AN4 (input)
P13 (input) / AN3 (input)
P12 (input) / AN2 (input)
P11 (input) / AN1 (input)
P10 (input) / AN0 (input)

Figure 18.14 Port I (PI7 to PI0)

18.10.1 Register Configuration

The port I register is shown in table 18.17.

Table 18.17 Port I Register

Name	Abbreviation	R/W	Initial Value*	Address	Access Size
Port I data register	PIDR	R	Depends on external pins	H'FFFF 1290	8

Note: * Initial value depends on the pin state when a read is performed.

18.10.2 Port I Data Register (PIDR)

Bit:	7	6	5	4	3	2	1	0
	PI7DR	PI6DR	PI5DR	PI4DR	PI3DR	PI2DR	PI1DR	PI0DR
Initial value:	*	*	*	*	*	*	*	*
R/W:	R	R	R	R	R	R	R	R

Note: * Initial value depends on the pin state when a read is performed.

The port I data register (PIDR) is an 8-bit read-only register that stores port I data. Bits PI7DR to PI0DR correspond to pins PI7/AN7 to PI0/An0.

Writes to these bits are ignored, and do not affect the pin states. When these bits are read the pin state, not the register value, is returned directly. However, 1 will be returned while A/D converter analog input is being sampled. Table 18.18 summarizes port I data register read/write operations.

PIDR is not initialized by a power-on or WDT reset, or in standby mode or sleep mode. (The bits always reflect the pin states.)

Table 18.18 Port I Data Register (PIDR) Read/Write Operations

Pin Input/Output	Pin Function	Read	Write
Input	General input	Pin state is read	Ignored (does not affect pin state)
	ANn	1 is read	Ignored (does not affect pin state)

Legend:
ANn: Analog input

Section 19 256 kB Flash Memory (F-ZTAT)

19.1 Features

The SH7065 has 256 kbytes of on-chip flash memory. The flash memory has the following features:

- Four flash memory operating modes
  - Program modeg
  - Erase mode
  - Program-verify mode
  - Erase-verify mode
- Programming/erase methods

  The flash memory is programmed 128 bytes at a time. Erasing is performed in block units (to erase the entire memory, individual blocks must be erased successively). Block erasing can be performed as required on 4 kB, 32 kB, and 64 kB blocks.
- Programming/erase times

  The flash memory programming time is 15 ms (typ.) for simultaneous 128-byte programming, equivalent to 117 µs (typ.) per byte. The erase time is 10 ms (typ.).
- Reprogramming capability

  The flash memory can be reprogrammed up to 100 times.
- On-board programming modes

  There are two modes in which flash memory can be programmed/erased/verified on-board:
  - Boot mode
  - User program mode
- Automatic bit rate adjustment

  With data transfer in boot mode, the SH7065's bit rate can be automatically adjusted to match the transfer bit rate of the host.
- Flash memory emulation in on-chip RAM

  Flash memory programming can be emulated in real time by overlapping a part of on-chip RAM onto flash memory.
- Protect modes

  There are two protect modes, hardware and software, which allow protected status to be designated for flash memory program/erase/verify operations

- Programmer mode

  Flash memory can be programmed/erased in programmer mode, using a PROM programmer, as well as in on-board programming mode.

19.2 Overview

19.2.1 Block Diagram

Figure 19.1 shows a block diagram of the flash memory.

The on-chip ROM is 64 bits wide, and can be accessed in two cycles. The on-chip ROM is connected to the internal data bus (C-bus) via a buffer, and has a basic access capability of 32 bits per cycle.

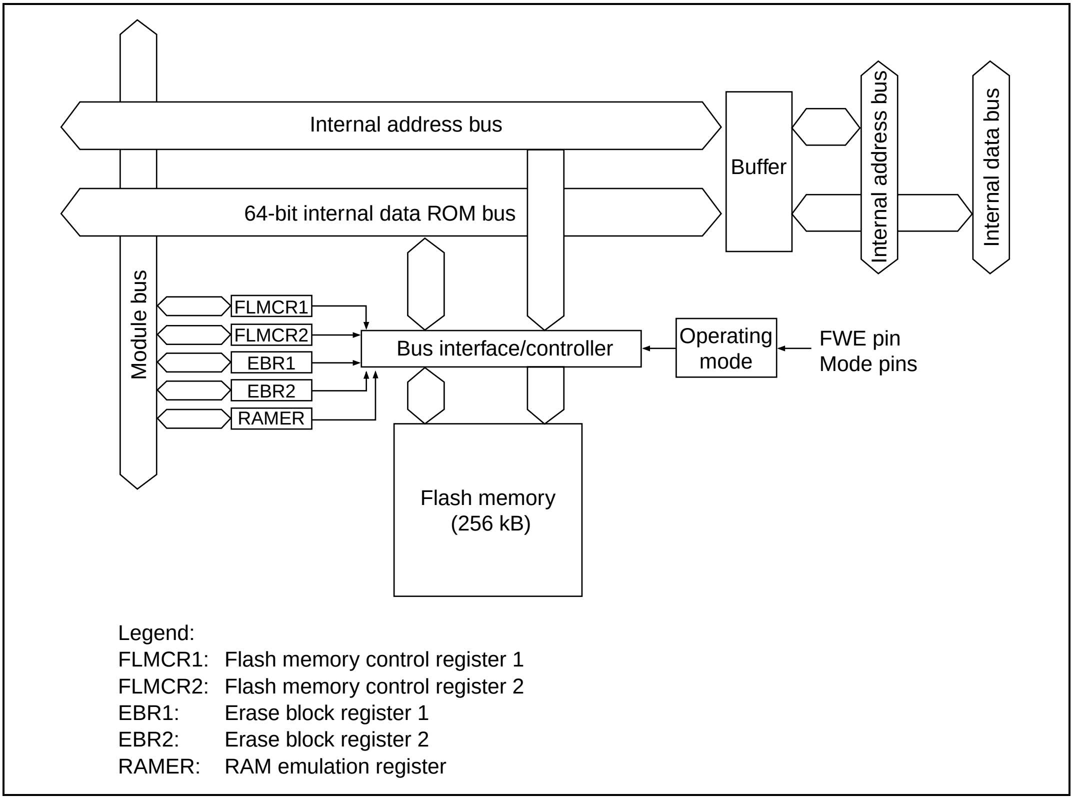
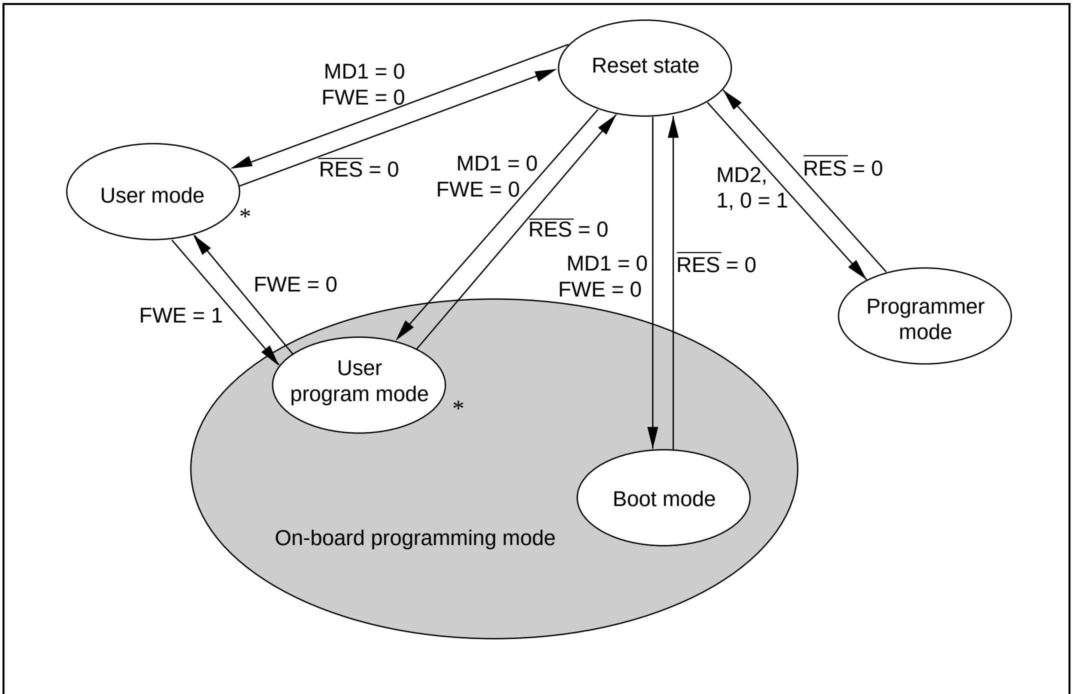

Internal address bus
64-bit internal data ROM bus
Buffer
Internal address bus
Internal data bus
Module bus
FLMCR1
FLMCR2
EBR1
EBR2
RAMER
Bus interface/controller
Operating mode
FWE pin
Mode pins
Flash memory (256 kB)
Legend:
FLMCR1: Flash memory control register 1
FLMCR2: Flash memory control register 2
EBR1: Erase block register 1
EBR2: Erase block register 2
RAMER: RAM emulation register

Figure 19.1 Block Diagram of Flash Memory

19.2.2 Mode Transitions

When the mode pins and the FWE pin are set in the reset state and a reset start is executed, the SH7065 enters one of the operating modes shown in figure 19.2. In user mode, flash memory can be read but not programmed or erased.

Flash memory can be programmed and erased in boot mode, user program mode, and programmer mode.



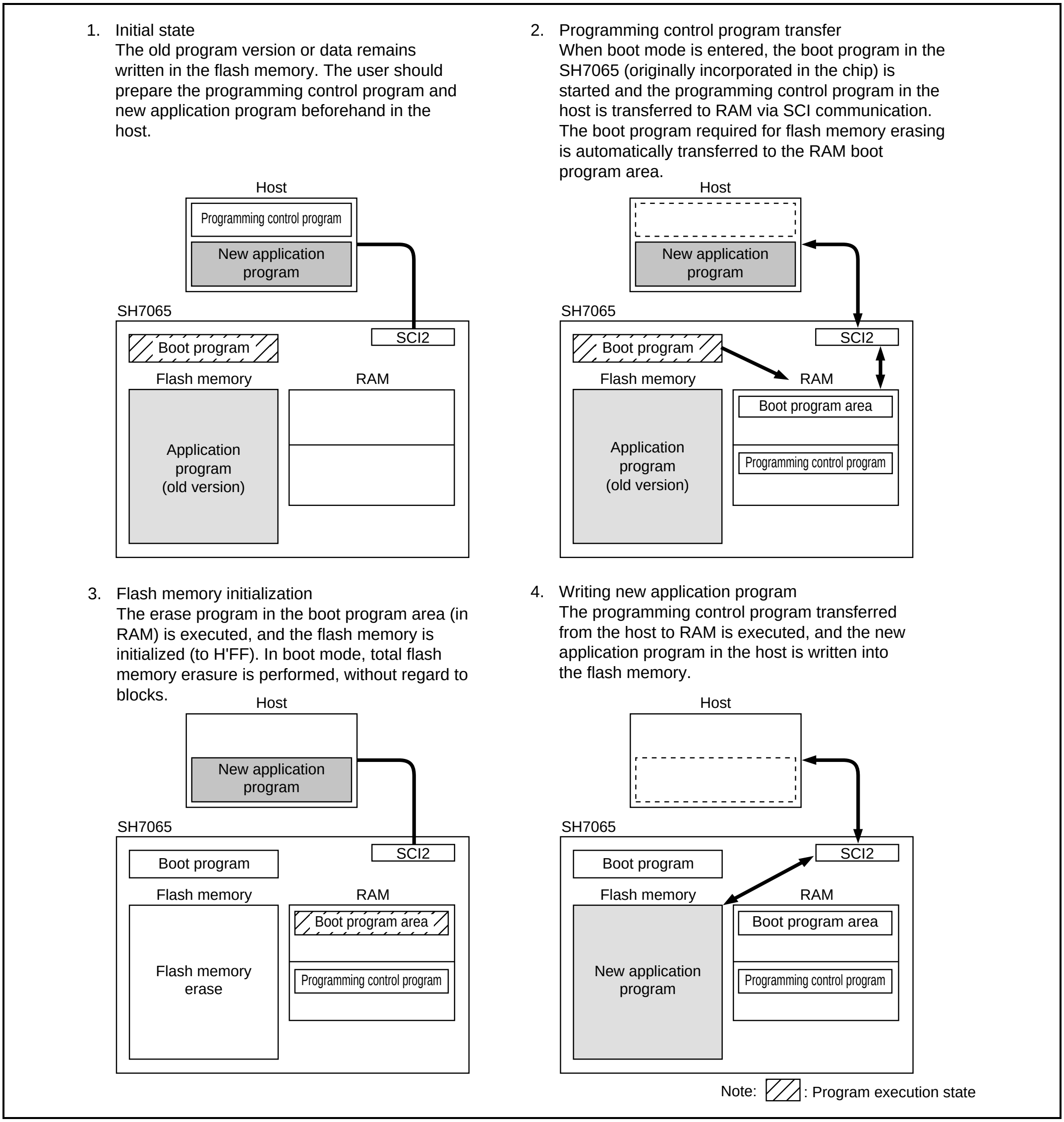
Note: * RAM emulation possible

Figure 19.2   Flash Memory Mode Transitions

19.2.3 On-Board Programming Modes

Boot Mode

Figure 19.3 shows programming operation in boot mode. For details of this mode, see section 19.6.1, Boot Mode.



1. Initial state
The old program version or data remains written in the flash memory. The user should prepare the programming control program and new application program beforehand in the host.

2. Programming control program transfer
When boot mode is entered, the boot program in the SH7065 (originally incorporated in the chip) is started and the programming control program in the host is transferred to RAM via SCI communication. The boot program required for flash memory erasing is automatically transferred to the RAM boot program area.

3. Flash memory initialization
The erase program in the boot program area (in RAM) is executed, and the flash memory is initialized (to H'FF). In boot mode, total flash memory erasure is performed, without regard to blocks.

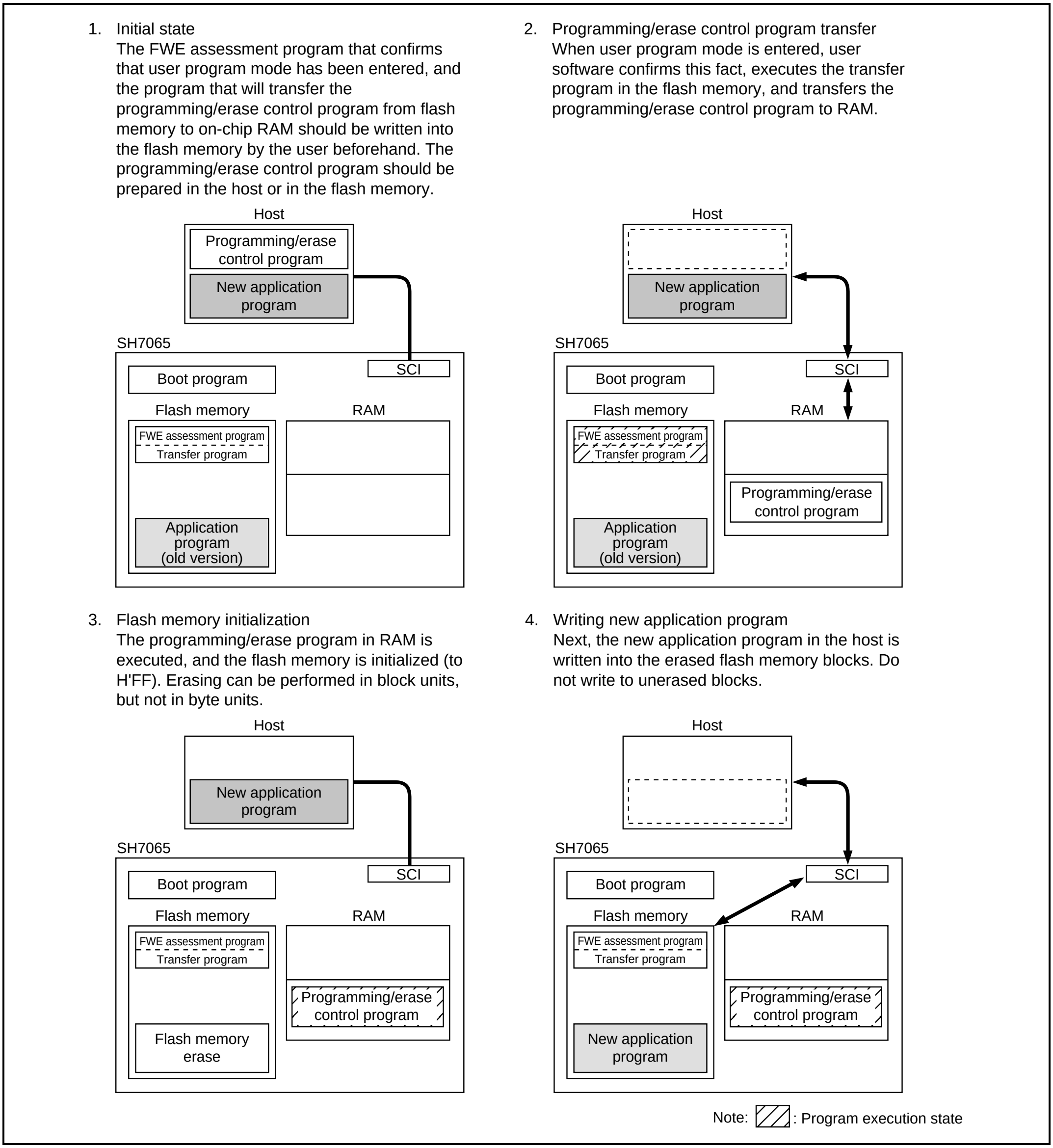
4. Writing new application program
The programming control program transferred from the host to RAM is executed, and the new application program in the host is written into the flash memory.

Note: ▨ : Program execution state

Figure 19.3 Programming Operation in Boot Mode

User Program Mode

Figure 19.4 shows an example of programming in user program mode. For details of this mode, see section 19.6.2, User Program Mode.



1. Initial state
   The FWE assessment program that confirms that user program mode has been entered, and the program that will transfer the programming/erase control program from flash memory to on-chip RAM should be written into the flash memory by the user beforehand. The programming/erase control program should be prepared in the host or in the flash memory.

2. Programming/erase control program transfer
   When user program mode is entered, user software confirms this fact, executes the transfer program in the flash memory, and transfers the programming/erase control program to RAM.

3. Flash memory initialization
   The programming/erase program in RAM is executed, and the flash memory is initialized (to H'FF). Erasing can be performed in block units, but not in byte units.

4. Writing new application program
   Next, the new application program in the host is written into the erased flash memory blocks. Do not write to unerased blocks.

Note: ▨: Program execution state

Figure 19.4 Example of Programming Operation in User Program Mode

19.2.4 Flash Memory Emulation in RAM

Emulation should be performed in user mode or user program mode. When the emulation block set in RAMER is accessed while the emulation function is being executed, read and write accesses are performed in the overlap RAM.

- User Mode
- User Program Mode

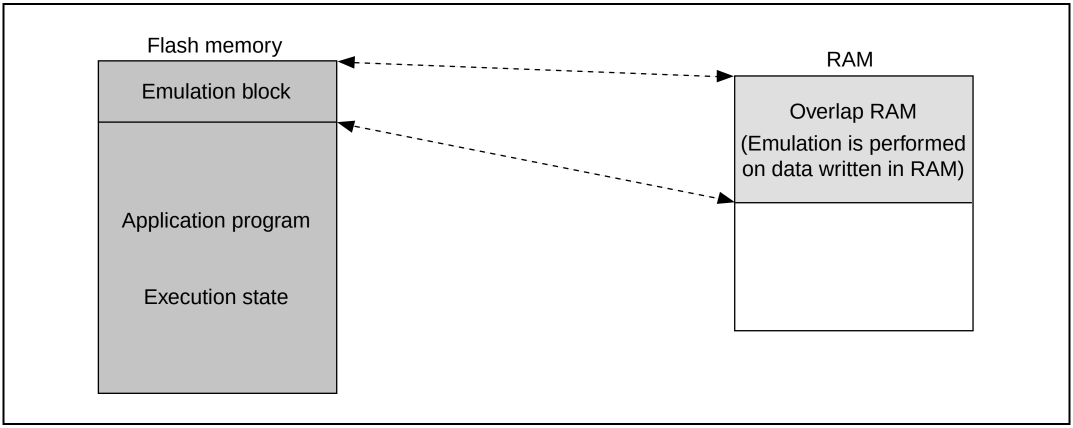


Figure 19.5 RAM Emulation (RAM Overlap)

When overlap RAM data is confirmed, clear the RAMS bit to cancel RAM overlap, and actually perform writes to the flash memory in user program mode.

When the programming control program is transferred to RAM, ensure that the transfer destination and the overlap RAM do not overlap, as this will cause data in the overlap RAM to be rewritten.

- User Program Mode

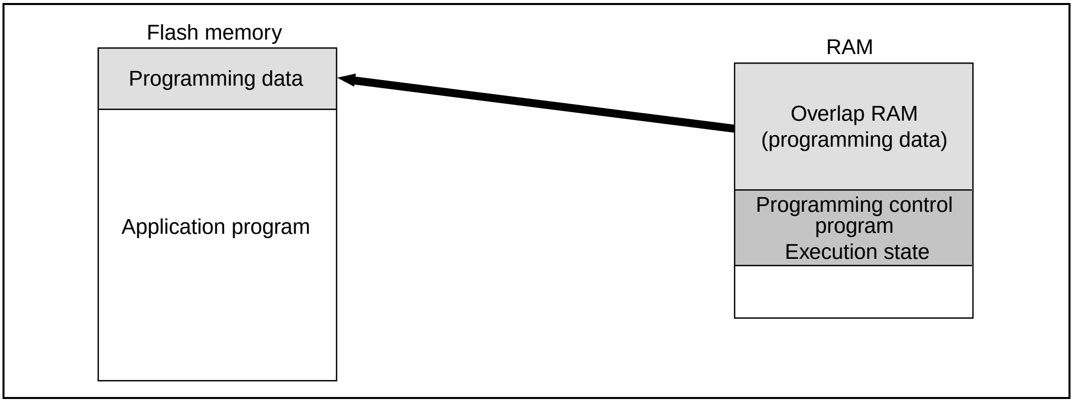


Figure 19.6 RAM Emulation (Flash Memory Programming)

19.2.5 Differences between Boot Mode and User Program Mode

Table 19.1 Differences between Boot Mode and User Program Mode

	Boot Mode	User Program Mode
Total erase	Yes	Yes
Block erase	No	Yes
Programming control program*	(2)	(1) (2) (3)

(1) Erase/erase-verify

(2) Program/program-verify

(3) Emulation

Note: * To be provided by the user, in accordance with the recommended algorithm.

19.2.6 Block Configuration

The flash memory is divided into eight 4 kB blocks, one 32 kB block, and three 64 kB blocks. In user program mode, erasing can be carried out in block units.

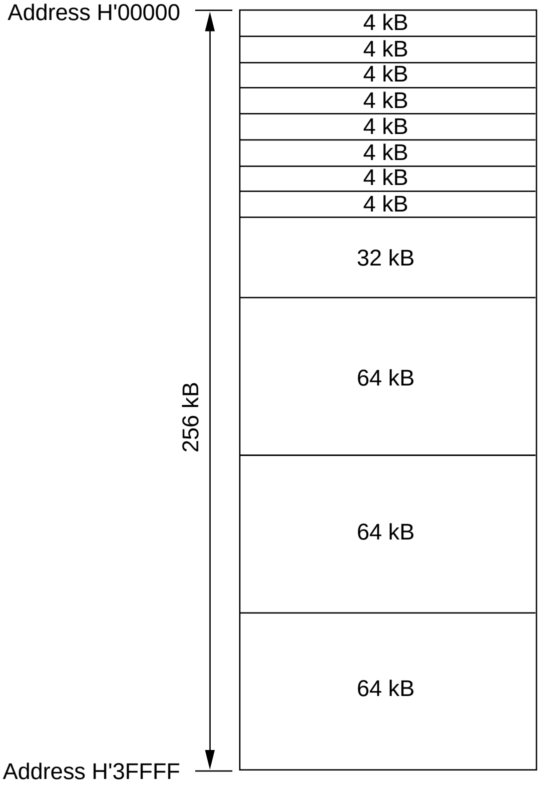


Figure 19.7 Erase Area Block Divisions

19.3 Pin Configuration

The flash memory is controlled by means of the pins shown in table 19.2.

Table 19.2 Flash Memory Pins

Pin Name	Abbreviation	I/O	Function
Reset	$\overline{\text{RES}}$	Input	Reset
Flash write enable	FWE	Input	Program/erase protection by hardware
Mode 5	MD5	Input	Sets SH7065 clock operating mode
Mode 4	MD4	Input	Sets SH7065 clock operating mode
Mode 3	MD3	Input	Sets SH7065 clock operating mode
Mode 2	MD2	Input	Sets SH7065 operating mode
Mode 1	MD1	Input	Sets SH7065 operating mode
Mode 0	MD0	Input	Sets SH7065 operating mode
Transmit data	TxD2 (PG30)	Output	Serial transmit data output
Receive data	RxD2 (PG31)	Input	Serial receive data input

19.4 Register Configuration

The registers used to control the on-chip flash memory when enabled are shown in table 19.3.

Table 19.3 Flash Memory Registers

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Flash memory control register 1	FLMCR1	R/W*[1]	H'00*[2]	FFFF0800	8
Flash memory control register 2	FLMCR2	R	H'00	FFFF0801	8
Erase block register 1	EBR1	R/W*[1]	H'00*[3]	FFFF0802	8
Erase block register 2	EBR2	R/W*[1]	H'00*[3]	FFFF0803	8
RAM emulation register	RAMER	R/W	H'0000	FFFF0C70	8, 16, 32

Notes: FLMCR1, FLMCR2, EBR1, and EBR2 are 8-bit registers, and RAMER is a 16-bit register.

Only byte accesses are valid for FLMCR1, FLMCR2, EBR1, and EBR2, the access requiring 3 cycles. Three cycles are required for a byte or word access to RAMER, and 6 cycles for a longword access.

When a longword write is performed on RAMER, 0 must always be written to the lower word (address H'FFFF0C72). Operation is not guaranteed if a nonzero value is written.

1. In the on-chip ROM disabled modes (MCU modes 2, 3, and 4), a read will return H'00, and writes are invalid. Writes are also disabled when the FWE bit is not set to 1 in FLMCR1.
2. When a high level is input to the FWE pin, the initial value is H'80.
3. When a low level is input to the FWE pin, or if a high level is input and the SWE bit in FLMCR1 is not set, these registers are initialized to H'00.

19.5 Register Descriptions

19.5.1 Flash Memory Control Register 1 (FLMCR1)

FLMCR1 is an 8-bit register used for flash memory operating mode control.

Program-verify mode or erase-verify mode is entered by setting SWE to 1 when FWE = 1, then setting the EV or PV bit.

Program mode is entered by setting SWE to 1 when FWE = 1, then setting the PSU bit, and finally setting the P bit.

Erase mode is entered by setting SWE to 1 when FWE = 1, then setting the ESU bit, and finally setting the E bit.

FLMCR1 is initialized by a reset, and in hardware standby mode and software standby mode. Its initial value is H'80 when a high level is input to the FWE pin, and H'00 when a low level is input. In the on-chip ROM disabled modes (MCU modes 2, 3, and 4), a read will return H'00, and writes are invalid.

Writes to bits ESU, PSU, EV, and PV in FLMCR1 are enabled only when FWE = 1 and SWE = 1; writes to the E bit only when FWE = 1, SWE = 1, and ESU = 1; and writes to the P bit only when FWE = 1, SWE = 1, and PSU = 1.

Bit:	7	6	5	4	3	2	1	0
	FWE	SWE	ESU	PSU	EV	PV	E	P
Initial value:	1/0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7—Flash Write Enable (FWE): Sets hardware protection against flash memory programming/erasing.

Bit 7: FWE	Description
0	When a low level is input to the FWE pin (hardware-protected state)
1	When a high level is input to the FWE pin

Bit 6—Software Write Enable (SWE): Enables or disables the flash memory. This bit should be set when setting bits 5 to 0, EBR1 bits 7 to 0, and EBR2 bits 3 to 0.

When SWE = 1, flash memory can only be read in program-verify or erase-verify mode.

Bit 6: SWE	Description	
0	Programming/erasing disabled	(Initial value)
1	Programming/erasing enabled [Setting condition] When FWE = 1	

Bit 5—Erase Setup (ESU): Prepares for a transition to erase mode. Do not set the SWE, PSU, EV, PV, E, or P bit at the same time.

Bit 5: ESU	Description	
0	Erase setup cleared	(Initial value)
1	Erase setup [Setting condition] When FWE = 1 and SWE = 1	

Bit 4—Program Setup (PSU): Prepares for a transition to program mode. Do not set the SWE, ESU, EV, PV, E, or P bit at the same time.

Bit 4: PSU	Description	
0	Program setup cleared	(Initial value)
1	Program setup [Setting condition] When FWE = 1 and SWE = 1	

Bit 3—Erase-Verify (EV): Selects erase-verify mode transition or clearing. Do not set the SWE, ESU, PSU, PV, E, or P bit at the same time.

Bit 3: EV	Description	
0	Erase-verify mode cleared	(Initial value)
1	Transition to erase-verify mode [Setting condition] When FWE = 1 and SWE = 1	

Bit 2—Program-Verify (PV): Selects program-verify mode transition or clearing. Do not set the SWE, ESU, PSU, EV, E, or P bit at the same time.

Bit 2: PV	Description	
0	Program-verify mode cleared	(Initial value)
1	Transition to program-verify mode [Setting condition] When FWE = 1 and SWE = 1	

Bit 1—Erase (E): Selects erase mode transition or clearing. Do not set the SWE, ESU, PSU, EV, PV, or P bit at the same time.

Bit 1: E	Description	
0	Erase mode cleared	(Initial value)
1	Transition to erase mode [Setting condition] When FWE = 1, SWE = 1, and ESU = 1	

Bit 0—Program (P): Selects program mode transition or clearing. Do not set the SWE, ESU, PSU, EV, PV, or E bit at the same time.

Bit 0: P	Description	
0	Program mode cleared	(Initial value)
1	Transition to program mode [Setting condition] When FWE = 1, SWE = 1, and PSU = 1	

19.5.2 Flash Memory Control Register 2 (FLMCR2)

FLMCR2 is an 8-bit register used to monitor enabling or disabling of flash memory program/erase protection (error protection).

FLMCR2 is initialized by a reset and in hardware standby mode. In the on-chip ROM disabled modes (MCU modes 2, 3, and 4), a read will return H'00, and writes are invalid.

Note: FLMCR2 is a read-only register, and should not be written to.

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

Bit 7—Flash Memory Error (FLER): Indicates that an error has occurred during an operation on flash memory (programming or erasing). When FLER is set to 1, flash memory goes to the error-protection state.

Bit 7: FLER	Description
0	Flash memory is operating normally Flash memory program/erase protection (error protection) is disabled [Clearing condition] Reset or hardware standby mode (Initial value)
1	An error has occurred during flash memory programming/erasing Flash memory program/erase protection (error protection) is enabled [Setting condition] See section 19.8.3, Error Protection

Bits 6 to 0—Reserved: These bits are always read as 0.

19.5.3 Erase Block Register 1 (EBR1)

EBR1 is an 8-bit register that specifies the flash memory erase area block by block.

EBR1 is initialized to H'00 by a reset, in hardware standby mode and software standby mode, when a low level is input to the FWE pin, and when a high level is input to the FWE pin and the SWE bit in FLMCR1 is not set. When a bit in EBR1 is set to 1, the corresponding block can be erased. Other blocks are erase-protected. Only one bit can be set in EBR1 and EBR2 together; do not set more than two bits. If more than two bits are set, EBR1 and EBR2 will both be cleared to H'00. In the on-chip ROM disabled modes (MCU modes 2, 3, and 4), a read will return H'00, and writes are invalid.

The flash memory block configuration is shown in table 19.4.

Bit:	7	6	5	4	3	2	1	0
	EB7	EB6	EB5	EB4	EB3	EB2	EB1	EB0
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

19.5.4 Erase Block Register 2 (EBR2)

EBR2 is an 8-bit register that specifies the flash memory erase area block by block.

EBR2 is initialized to H'00 by a reset, in hardware standby mode and software standby mode, when a low level is input to the FWE pin, and when a high level is input to the FWE pin and the SWE bit in FLMCR1 is not set. When a bit in EBR2 is set to 1, the corresponding block can be erased. Other blocks are erase-protected. In the on-chip ROM disabled modes (MCU modes 2, 3, and 4), a read will return H'00, and writes are invalid.

The flash memory block configuration is shown in table 19.4.

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	EB11	EB10	EB9	EB8
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W	R/W

Bits 7 to 4—Reserved: These bits are always read as 0.

Table 19.4 Flash Memory Erase Blocks

Block (Size)	Addresses
EB0 (4 kB)	H'000000–H'000FFF
EB1 (4 kB)	H'001000–H'001FFF
EB2 (4 kB)	H'002000–H'002FFF
EB3 (4 kB)	H'003000–H'003FFF
EB4 (4 kB)	H'004000–H'004FFF
EB5 (4 kB)	H'005000–H'005FFF
EB6 (4 kB)	H'006000–H'006FFF
EB7 (4 kB)	H'007000–H'007FFF
EB8 (32 kB)	H'008000–H'00FFFF
EB9 (64 kB)	H'010000–H'01FFFF
EB10 (64 kB)	H'020000–H'02FFFF
EB11 (64 kB)	H'030000–H'03FFFF

19.5.5 RAM Emulation Register (RAMER)

RAMER specifies the area of flash memory to be overlapped with part of RAM when emulating real-time flash memory programming. RAMER is initialized to H'0000 by a reset and in hardware standby mode. It is not initialized in software standby mode. RAMER settings should be made in user mode or user program mode.

Flash memory area divisions are shown in table 19.5. To ensure correct operation of the emulation function, the ROM for which RAM emulation is performed should not be accessed immediately after this register has been modified. Normal execution of an access immediately after register modification is not guaranteed.

Bit:	15	14	13	12	11	10	9	8
	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit:	7	6	5	4	3	2	1	0
	—	—	—	RAMAS	RAMS	RAM2	RAM1	RAM0
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

Bits 15 to 5—Reserved: These bits are always read as 0.

Bit 4—RAM Address Select (RAMAS): Selects the RAM addresses to be used for flash memory emulation. This bit is ignored in the on-chip ROM disabled modes (MCU modes 2, 3, and 4).

Bit 4: RAMAS	Description
0	RAM addresses H'FFFF8000 to H'FFFF8FFF are used for emulation (Initial value)
1	RAM addresses H'FFFFA000 to H'FFFFAFFF are used for emulation

Bit 3—RAM Select (RAMS): Specifies selection or non-selection of flash memory emulation in RAM. When RAMS = 1, all flash memory block are program/erase-protected. This bit is ignored in the on-chip ROM disabled modes (MCU modes 2, 3, and 4).

Bit 3: RAMS	Description
0	Emulation not selected Program/erase-protection of all flash memory blocks is disabled(Initial value)
1	Emulation selected Program/erase-protection of all flash memory blocks is enabled

Bits 2 to 0—Flash Memory Area Selection (RAM2 to RAM0): These bits are used together with bit 3 to select the flash memory area to be overlapped with RAM. (See table 19.5.)

Table 19.5 Flash Memory Area Divisions

Addresses	Block Name	RAMS	RAM2	RAM1	RAM0
Addresses selected by RAMAS bit	4 kB RAM area	0	*	*	*
H'000000–H'000FFF	EB0 (4kB)	1	0	0	0
H'001000–H'001FFF	EB1 (4kB)	1	0	0	1
H'002000–H'002FFF	EB2 (4kB)	1	0	1	0
H'003000–H'003FFF	EB3 (4kB)	1	0	1	1
H'004000–H'004FFF	EB4 (4kB)	1	1	0	0
H'005000–H'005FFF	EB5 (4kB)	1	1	0	1
H'006000–H'006FFF	EB6 (4kB)	1	1	1	0
H'007000–H'007FFF	EB7 (4kB)	1	1	1	1

Legend:
*: Don't care

19.6 On-Board Programming Modes

When pins are set to on-board programming mode and a reset start is executed, a transition is made to the on-board programming state in which program/erase/verify operations can be performed on the on-chip flash memory. There are two on-board programming modes: boot mode and user program mode. The pin settings for transition to each of these modes are shown in table 19.6. For a diagram of the transitions to the various flash memory modes, see figure 19.2.

Table 19.6 Setting On-Board Programming Modes

Mode		FWE	MD5	MD4	MD3	MD2	MD1	MD0
Boot mode	Expanded mode	1	See section 4, Clock Pulse Generator (CPG) and Power-Down Modes, for the clock modes.			0	1	1
	Single-chip mode					0	1	0
User program mode	Expanded mode	1				0	0	1
	Single-chip mode					0	0	0

19.6.1 Boot Mode

When boot mode is used, the flash memory programming control program must be prepared in the host beforehand. The SCI channel to be used is set to asynchronous mode.

When a reset start is executed after the SH7065 pins have been set to boot mode, the boot program built into the SH7065 is started and the programming control program prepared in the host is serially transmitted to the SH7065 via SCI channel 2 (RxD2 (PG31), TxD2 (PG30)). In the SH7065, the user program received via SCI channel 2 is written into the user program area in on-chip RAM. After the transfer is completed, control branches to the start address of the user program area and the user program execution state is entered (flash memory programming is performed). The transferred user program must therefore include coding that follows the programming algorithm given later.

The system configuration in boot mode is shown in figure 19.8, and the boot program mode execution procedure in figure 19.9.

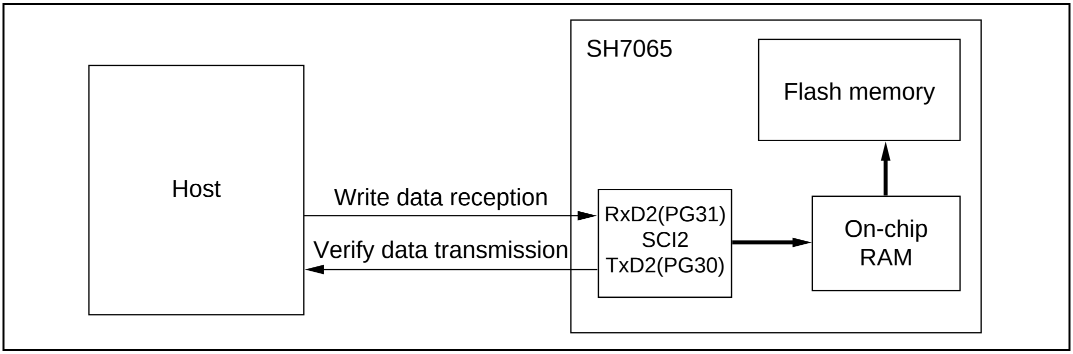
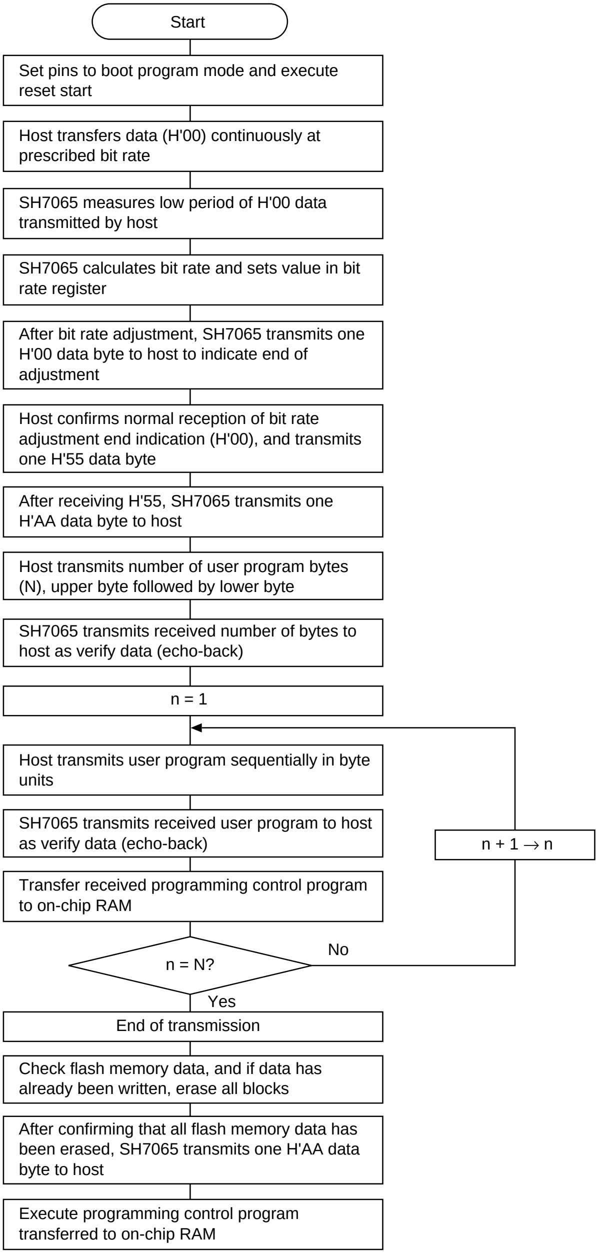


Figure 19.8 System Configuration in Boot Mode


Start
Set pins to boot program mode and execute reset start
Host transfers data (H'00) continuously at prescribed bit rate
SH7065 measures low period of H'00 data transmitted by host
SH7065 calculates bit rate and sets value in bit rate register
After bit rate adjustment, SH7065 transmits one H'00 data byte to host to indicate end of adjustment
Host confirms normal reception of bit rate adjustment end indication (H'00), and transmits one H'55 data byte
After receiving H'55, SH7065 transmits one H'AA data byte to host
Host transmits number of user program bytes (N), upper byte followed by lower byte
SH7065 transmits received number of bytes to host as verify data (echo-back)
n = 1
Host transmits user program sequentially in byte units
SH7065 transmits received user program to host as verify data (echo-back)
n + 1 → n
Transfer received programming control program to on-chip RAM
n = N?
No
Yes
End of transmission
Check flash memory data, and if data has already been written, erase all blocks
After confirming that all flash memory data has been erased, SH7065 transmits one H'AA data byte to host
Execute programming control program transferred to on-chip RAM

Note: If a memory cell does not operate normally and cannot be erased, one H'FF byte is transmitted as an erase error, and the erase operation and subsequent operations are halted.

Figure 19.9 Boot Mode Execution Procedure

Automatic SCI Bit Rate Adjustment

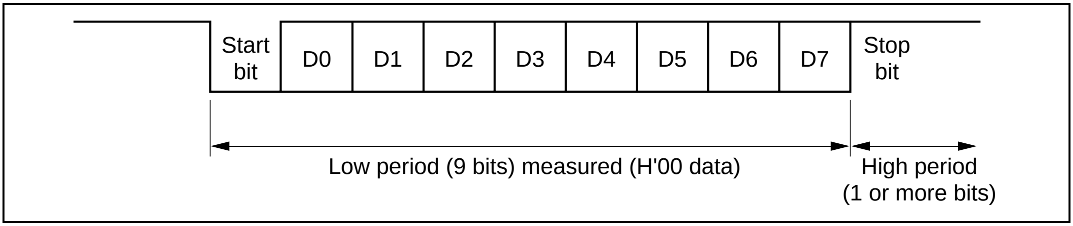

Start bit
D0
D1
D2
D3
D4
D5
D6
D7
Stop bit
Low period (9 bits) measured (H'00 data)
High period (1 or more bits)

Figure 19.10 Automatic SCI Bit Rate Adjustment

When boot mode is initiated, the SH7065 measures the low period of the asynchronous SCI communication data (H'00) transmitted continuously from the host. The SCI transmit/receive format should be set as follows: 8-bit data, 1 stop bit, no parity. The SH7065 calculates the bit rate of the transmission from the host from the measured low period, and transmits one H'00 byte to the host to indicate the end of bit rate adjustment. The host should confirm that this adjustment end indication (H'00) has been received normally, and transmit one H'55 byte to the SH7065. If reception cannot be performed normally, initiate boot mode again (reset), and repeat the above operations.

Table 19.7 shows host transfer bit rates and peripheral clock frequencies for which automatic adjustment of the SH7065 bit rate is possible. The boot program should set the initial settings for the clock division ratios (in clock mode 7 only, the peripheral clock is set to 1/2 the input), no module clock division, and 4 times the bit rate for the SCI2 base clock. The boot program should be executed with a bit rate for which adjustment is possible according to the peripheral clock frequency.

Table 19.7 Peripheral Clock Frequencies Enabling Automatic Adjustment of SH7065 Bit Rate

Host Bit Rate	Peripheral Clock Frequency Enabling Automatic Adjustment of SH7065 Bit Rate
19200 bps	4–30 MHz
9600 bps	2–30 MHz

On-Chip RAM Area Divisions in Boot Mode

In boot mode, the RAM area is divided into an area used by the boot program and an area to which the programming control program is transferred via the SCI, as shown in figure 19.11. The boot program area cannot be used until the execution state in boot mode switches to the programming control program transferred from the host.

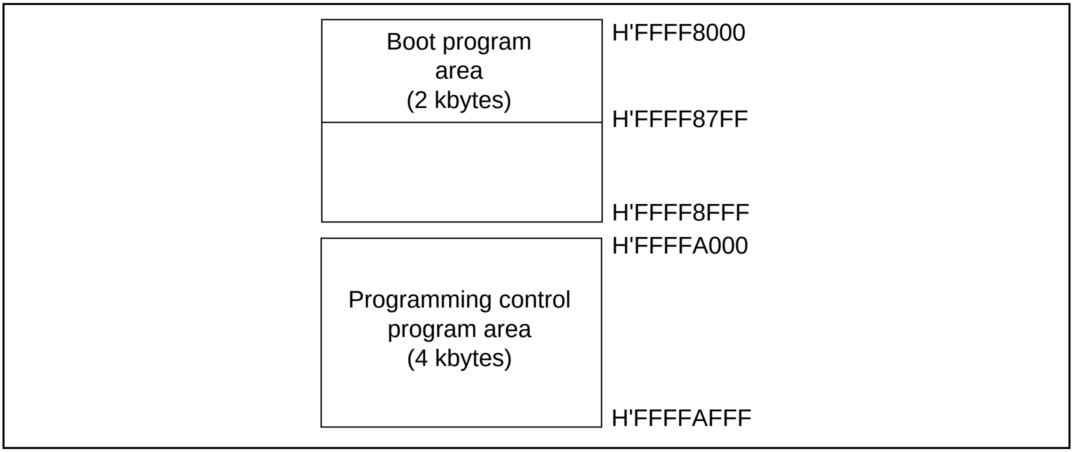

Boot program area (2 kbytes)
H'FFFF8000
H'FFFF87FF
H'FFFF8FFF
H'FFFFA000
Programming control program area (4 kbytes)
H'FFFFAFFF

Figure 19.11 RAM Areas in Boot Mode

Note: The boot program area cannot be used until a transition is made to the execution state for the programming control program transferred to RAM. Note also that the boot program remains in this area of the on-chip RAM even after control branches to the programming control program.

19.6.2 User Program Mode

After setting the FWE pin, the programming/erase control program prepared by the user beforehand should be branched to and executed.

As the flash memory itself cannot be read while flash memory programming/erasing is being executed, the control program that performs programming and erasing should be run in on-chip RAM or external memory.

Use the following procedure (figure 19.12) to execute the programming control program that writes to flash memory (when transferred to RAM).

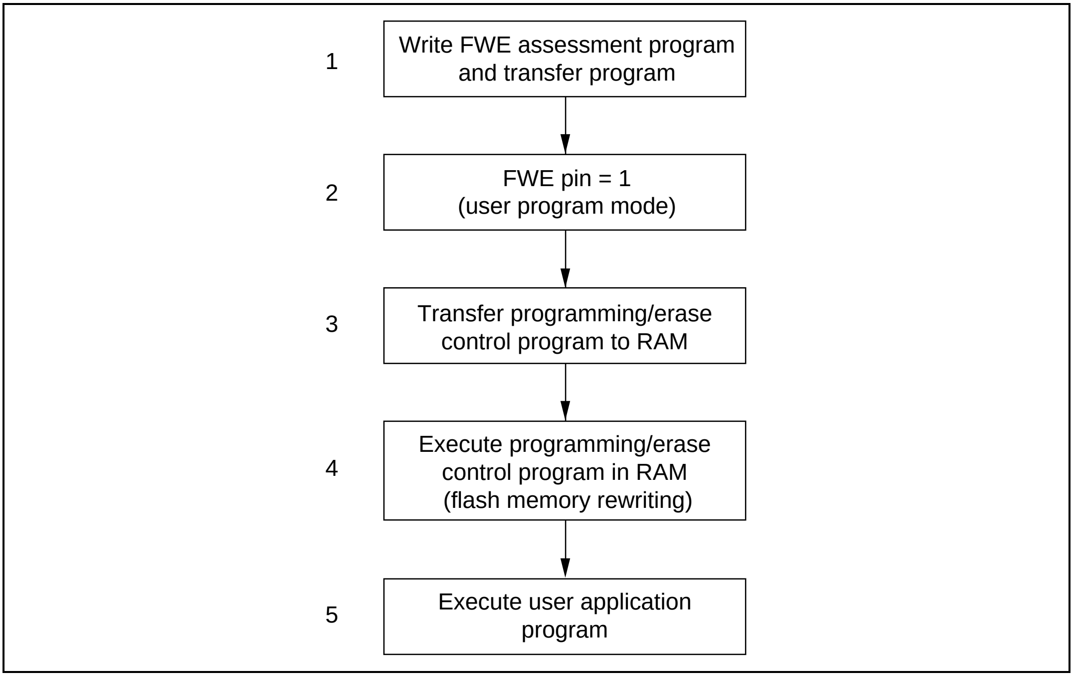

1
Write FWE assessment program and transfer program
2
FWE pin = 1 (user program mode)
3
Transfer programming/erase control program to RAM
4
Execute programming/erase control program in RAM (flash memory rewriting)
5
Execute user application program

Figure 19.12 User Program Mode Execution Procedure

Notes: 1. When programming and erasing, start the watchdog timer so that measures can be taken to prevent program runaway, etc. Memory cells may not operate normally if overprogrammed or overerased due to program runaway.

2. If an address at which a flash memory register resides is read in the mask ROM version, the value will be undefined. If a flash memory version program is used in the mask ROM version, the state of the FWE pin cannot be determined. A modification must therefore be made to prevent operation of the flash memory rewrite program.

19.7 Programming/Erasing Flash Memory

A software method, using the CPU, is employed to program and erase flash memory in the on-board programming modes. There are four flash memory operating modes: program mode, erase mode, program-verify mode, and erase-verify mode. Transitions to these modes are made by setting the PSU, ESU, P, E, PV, and EV bits in FLMCR1.

The flash memory cannot be read while being programmed or erased. Therefore, the program (user program) that controls flash memory programming/erasing should be located and executed in on-chip RAM or external memory.

Notes: 1. Operation is not guaranteed if setting or clearing of the SWE, ESU, PSU, EV, PV, E, and P bits in FLMCR1 is executed by a program in flash memory.

2. When programming or erasing, drive the FWE pin high (programming/erasing will not be executed if the FWE pin is low).

3. Programming must be executed in the erased state. Do not perform additional programming on addresses that have already been programmed.

19.7.1 Program Mode

When writing data or programs to flash memory, the program/program-verify flowchart shown in figure 19.13 should be followed. Performing program operations according to this flowchart will enable data or programs to be written to flash memory without subjecting the device to voltage stress or sacrificing program data reliability. Programming should be carried out 128 bytes at a time.

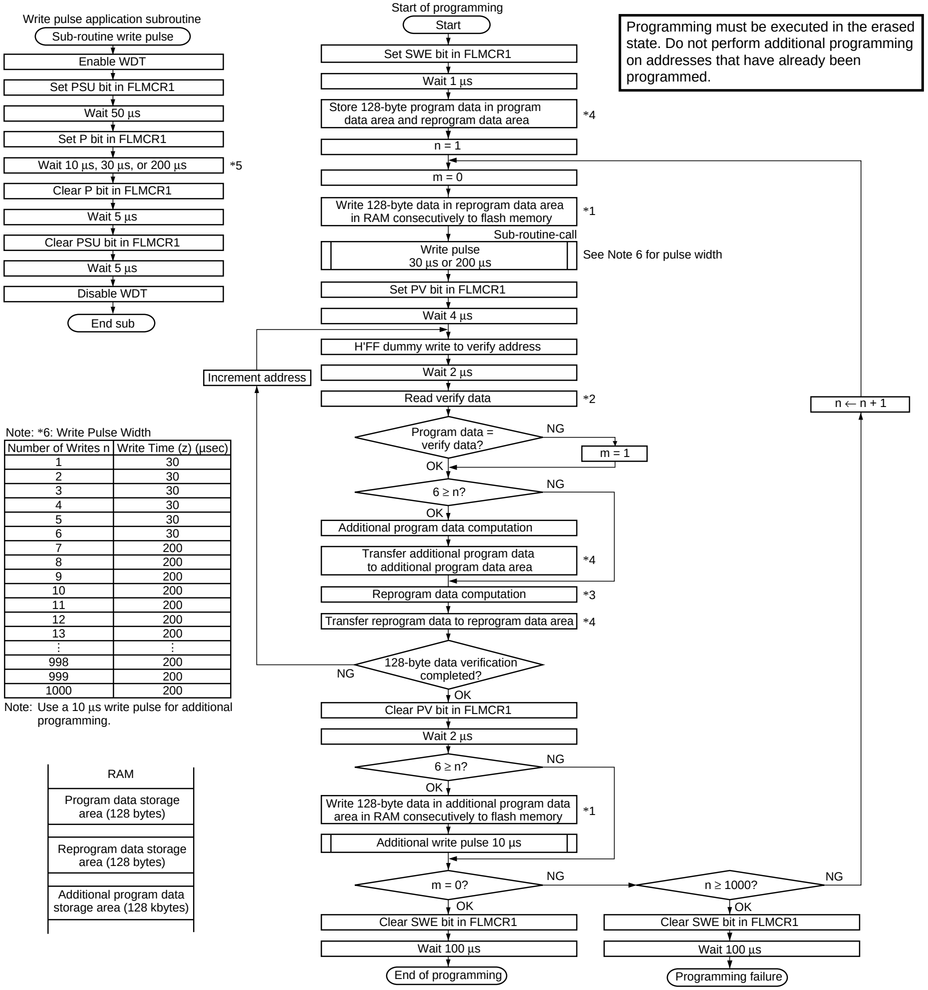
Following the elapse of 1 µs or more after the SWE bit is set to 1 in flash memory control register 1 (FLMCR1), 128-byte data is written consecutively to the write addresses (the lower 8 bits of the first address written to must be H'00 or H'80). 128 consecutive byte data transfers are performed. The program address and program data are latched in the flash memory. A 128-byte data transfer must be performed even if writing fewer than 128 bytes; in this case, H'FF data must be written to the extra addresses.

Next, the watchdog timer is set to prevent overprogramming in the event of program runaway, etc. After this, preparation for program mode (program setup) is carried out by setting the PSU bit in FLMCR1, and after the elapse of 50 µs or more, the operating mode is switched to program mode by setting the P bit in FLMCR1. The time during which the P bit is set is the flash memory programming time. See the table in the programming flowchart for the programming time.

19.7.2 Program-Verify Mode

In program-verify mode, the data written in program mode is read to check whether it has been correctly written in the flash memory.

After the elapse of a given programming time, the programming mode is exited (the P bit in FLMCR1 is cleared, then the PSU bit is cleared at least 5 µs later). After the elapse of 5 µs or more, the watchdog timer is cleared, and the operating mode is switched to program-verify mode by setting the PV bit in FLMCR1. In program-verify mode, a dummy write of H'FF data should be made to the addresses to be read. The dummy write should be executed after the elapse of 4 µs or more. When the flash memory is read in this state (verify data is read in 32-bit units), the data at the latched address is read. Wait at least 2 µs after the dummy write before performing this read operation. Next, the originally written data is compared with the verify data, and reprogram data is computed (see figure 19.13) and transferred to RAM. After 128 bytes of data have been verified, exit program-verify mode, wait for at least 2 µs, then clear the SWE bit in FLMCR1. If reprogramming is necessary, set program mode again, and repeat the program/program-verify sequence as before. However, ensure that the program/program-verify sequence is not repeated more than 1000 times on the same bits.


Write pulse application subroutine
Sub-routine write pulse
Enable WDT
Set PSU bit in FLMCR1
Wait 50 µs
Set P bit in FLMCR1
Wait 10 µs, 30 µs, or 200 µs
*5
Clear P bit in FLMCR1
Wait 5 µs
Clear PSU bit in FLMCR1
Wait 5 µs
Disable WDT
End sub
Start of programming
Start
Set SWE bit in FLMCR1
Wait 1 µs
Store 128-byte program data in program data area and reprogram data area
*4
n = 1
m = 0
Write 128-byte data in reprogram data area in RAM consecutively to flash memory
*1
Sub-routine-call
Write pulse 30 µs or 200 µs
See Note 6 for pulse width
Set PV bit in FLMCR1
Wait 4 µs
H'FF dummy write to verify address
Increment address
Wait 2 µs
Read verify data
*2
Program data = verify data?
NG
m = 1
OK
6 ≥ n?
NG
OK
Additional program data computation
Transfer additional program data to additional program data area
*4
Reprogram data computation
*3
Transfer reprogram data to reprogram data area
*4
128-byte data verification completed?
NG
OK
Clear PV bit in FLMCR1
Wait 2 µs
6 ≥ n?
NG
OK
Write 128-byte data in additional program data area in RAM consecutively to flash memory
*1
Additional write pulse 10 µs
m = 0?
NG
OK
n ≥ 1000?
NG
OK
n ← n + 1
Clear SWE bit in FLMCR1
Wait 100 µs
End of programming
Clear SWE bit in FLMCR1
Wait 100 µs
Programming failure
Programming must be executed in the erased state. Do not perform additional programming on addresses that have already been programmed.
RAM
Program data storage area (128 bytes)
Reprogram data storage area (128 bytes)
Additional program data storage area (128 kbytes)

Note: *6: Write Pulse Width

Number of Writes n	Write Time (z) (µsec)
1	30
2	30
3	30
4	30
5	30
6	30
7	200
8	200
9	200
10	200
11	200
12	200
13	200
⋮	⋮
998	200
999	200
1000	200

Note: Use a 10 µs write pulse for additional programming.

Notes: 1. Data transfer is performed by byte transfer. The lower 8 bits of the first address written to must be H'00 or H'80. A 128-byte data transfer must be performed even if writing fewer than 128 bytes; in this case, H'FF data must be written to the extra addresses.
2. Verify data is read in 32-bit (longword) units.
3. Even bits for which programming has been completed in the 128-byte programming loop will be subjected to additional programming if they fail the subsequent verify operation.
4. A 128-byte area for storing program data, a 128-byte area for storing reprogram data, and a 128-byte area for storing additional program data must be provided in RAM. The reprogram and additional program data contents are modified as programming proceeds.
5. The write pulse of 30 µs or 200 µs is applied according to the progress of the programming operation. See Note *6 for the pulse widths. When writing of additional program data is executed, a 10 µs write pulse should be applied. Reprogram data X' means reprogram data when the write pulse is applied.

Reprogram Data Computation Chart

Original Data (D)	Verify Data (V)	Reprogram Data (X)	Comments
0	0	1	Programming completed
	1	0	Programming incomplete; reprogram
1	0	1	
	1	1	Still in erased state; no action

Additional Program Data Computation Chart

Reprogram Data (X')	Verify Data (V)	Additional Program Data (Y)	Comments
0	0	0	Additional programming executed
	1	1	Additional programming not executed
1	0	1	Additional programming not executed
	1	1	Additional programming not executed

Figure 19.13 Program/Program-Verify Flowchart

Sample 128-Byte Programming Program

The wait time set values (number of loops) are for the case where f = 60 MHz. For other frequencies, the set value is given by {wait time (µs) × f (MHz) ÷ 4}.

Registers Used

R11 (input):	Program data storage address
R12 (input):	Programming destination address
R13 (output):	OK (normal) or NG (error)
R0–10, 14:	Work registers

```
FLMCR1          .EQU  H'00
OK              .EQU  H'0
NG              .EQU  H'1
WAIT_X          .EQU  15            ; 1 µs
WAIT_Y          .EQU  750           ; 50 µs
WAIT_Z1         .EQU  450           ; 30 µs (1st to 6th time)
WAIT_Z5         .EQU  3000          ; 200 µs (7th to 1000th time)
WAIT_ZA         .EQU  150           ; 10 µs (Additional write)
WAIT_A          .EQU  75            ; 5 µs
WAIT_B          .EQU  75            ; 5 µs
WAIT_C          .EQU  60            ; 4 µs
WAIT_D          .EQU  30            ; 2 µs
WAIT_E          .EQU  30            ; 2 µs
WAIT_F          .EQU  1500          ; 100 µs
WDT_TCSR        .EQU  H'FFFF1000
WDT_1M          .EQU  H'A57C
SWESET          .EQU  H'40          ; B'01000000
PSUSET          .EQU  H'50          ; B'01010000
PSET            .EQU  H'51          ; B'01010001
PCLEAR          .EQU  H'50          ; B'01010000
PSUCLEAR        .EQU  H'40          ; B'01000000
PVSET           .EQU  H'44          ; B'01000100
PVCLEAR         .EQU  H'40          ; B'01000000
SWECLEAR        .EQU  H'00          ; B'00000000
MAXVERIFY       .EQU  1000
;
```

```
FLASHPROGRAM    .EQU  $
        MOV     R11,R3              ; Save program data to
        MOV.L   #RDATABUFF,R0       ; work area
        MOV.L   #ADATABUFF,R2
        MOV     #32,R6
COPY_LOOP1      .EQU  $
        MOV.L   @R3+,R1
        MOV.L   R1,@R0
        MOV.L   R1,@R2
        ADD     #4,R0
        ADD     #4,R2
        DT      R6
        BF      COPY_LOOP1
        MOV.L   #H'FFFF0800,R0      ; Initialize GBR
        LDC     R0,GBR
;
        MOV.L   #WAIT_X,R2
        MOV     #SWESET,R0
        MOV.B   R0,@(FLMCR1,GBR)    ; Set SWE
WAIT_1  DT      R2                  ; Wait 1 µs
        BF      WAIT_1
;
        MOV     #1,R14              ; Initialize N (R14) to 1
PROGRAM_LOOP    .EQU  $
        MOV     #0,R5               ; Initialize M (R5) to 0
        MOV.L   #128,R2             ; Write 128-byte data consecutively
        MOV.L   #RDATABUFF,R3
        MOV     R12,R6
WRITE_LOOP1     .EQU  $
        MOV.B   @R3+,R1
        MOV.B   R1,@R6
        ADD     #1,R6
        DT      R2
        BF      WRITE_LOOP1
;
```

```
        MOV.L   #WDT_TCSR,R0          ; Enable WDT
        MOV.L   #WDT_1M,R1            ; 1.1 ms cycle
        MOV.W   R1,@R0
;
        MOV.L   #WAIT_Y,R2
        MOV     #PSUSET,R0            ; Set PSU
        MOV.B   R0,@(FLMCR1,GBR)
WAIT_2  DT      R2                    ; Wait 50 µs
        BF      WAIT_2
;
        MOV.L   #WAIT_Z1,R2           ; 1st to 6th time
        MOV     #6,R3
        CMP/GE  R14,R3
        BT      UNDER7
        MOV.L   #WAIT_Z5,R2           ; 7th to 1000th time
UNDER7  MOV     #PSET,R0              ; Set P
        MOV.B   R0,@(FLMCR1,GBR)
WAIT_3  DT      R2                    ; Wait 30 µs or 200 µs
        BF      WAIT_3
;
        MOV.L   #WAIT_A,R2
        MOV     #PCLEAR,R0            ; Clear P
        MOV.B   R0,@(FLMCR1,GBR)
WAIT_4  DT      R2                    ; Wait 5 µs
        BF      WAIT_4
;
        MOV.L   #WAIT_B,R2
        MOV     #PSUCLEAR,R0          ; Clear PSU
        MOV.B   R0,@(FLMCR1,GBR)
WAIT_5  DT      R2                    ; Wait 5 µs
        BF      WAIT_5
;
        MOV.L   #WDT_TCSR,R0          ; Disable WDT
        MOV.W   #H'A55F,R1
        MOV.W   R1,@R0
```

```
;
        MOV.L   #WAIT_C,R2
        MOV     #PVSET,R0         ; Set PV
        MOV.B   R0,@(FLMCR1,GBR)
WAIT_6  DT      R2                ; Wait 4 µs
        BF      WAIT_6
;
        MOV.L   #ADATABUFF,R9
        MOV.L   #RDATABUFF,R7
        MOV     R11,R1
        MOV     R12,R3
        MOV     #32,R6
        MOV.L   #H'FFFFFFFF,R4
;
VERIFYLOOP      .EQU  $
        MOV.L   R4,@R3            ; Write H'FF to verify address
        MOV.L   R4,@R9            ; Additional program data RAM (ADATABUFF) initialization
        MOV.L   #WAIT_D,R2
WAIT_7  DT      R2                ; Wait 2 µs
        BF      WAIT_7
;
        MOV.L   @R3+,R2           ; Read verify data
        MOV.L   @R1+,R0           ; Read program data (source data)
        CMP/EQ  R2,R0             ; Verify check
        BT      VERIFY_OK
        MOV     #1,R5             ; If verify NG, assign 1 to M
;
VERIFY_OK       .EQU  $
        MOV     #6,R8             ; 6 or more writes?
        CMP/GE  R14,R8
        BF      NO_ADWRT
        MOV.L   @R7,R10           ; Read reprogram data
        OR      R2,R10            ; Additional program data operation
        MOV.L   R10,@R9           ; Store in additional program data RAM (ADATABUFF)
;
```

```
NO_ADWRT        .EQU  $
        MOV.L   R4,@R7                ; Reprogram data RAM (RDATABUFF) initialization
        NOT     R2,R2                 ; Reprogram data computation
        OR      R2,R0
        MOV.L   R0,@R7                ; Store in reprogram data RAM (RDATABUFF)
;
        ADD     #4,R7
        ADD     #4,R9
        DT      R6
        BF      VERIFYLOOP
;
        MOV.L   #WAIT_E,R2
        MOV     #PVCLEAR,R0           ; Clear PV
        MOV.B   R0,@(FLMCR1,GBR)
WAIT_8  DT      R2                    ; Wait 2 µs
        BF      WAIT_8
;
        MOV     #6,R8                 ; 6 or more writes?
        CMP/GE  R14,R8
        BF      NO_ADWRT2
;
        MOV.L   #128,R2               ; Consecutively write 128-byte date to
        MOV.L   #ADATABUFF,R3         ; additional program data RAM (ADATABUFF)
        MOV     R12,R6
WRITE_LOOP2     .EQU  $
        MOV.B   @R3+,R1
        MOV.B   R1,@R6
        ADD     #1,R6
        DT      R2
        BF      WRITE_LOOP2
;
        MOV.L   #WDT_TCSR,R0          ; Enable WDT
        MOV.L   #WDT_1M,R1            ; 1.1 ms cycle
        MOV.W   R1,@R0
;
```

```
        MOV.L   #WAIT_Y,R2
        MOV     #PSUSET,R0                ; Set PSU
        MOV.B   R0,@(FLMCR1,GBR)
WAIT_9  DT      R2                        ; Wait 50 µs
        BF      WAIT_9
;
        MOV.L   #WAIT_ZA,R2               ; 10 µs additional write
        MOV     #PSET,R0                  ; Set P
        MOV.B   R0,@(FLMCR1,GBR)
WAIT_10 DT      R2                        ; Wait 10 µs
        BF      WAIT_10
;
        MOV.L   #WAIT_A,R2
        MOV     #PCLEAR,R0                ; Clear P
        MOV.B   R0,@(FLMCR1,GBR)
WAIT_11 DT      R2                        ; Wait 5 µs
        BF      WAIT_11
;
        MOV.L   #WAIT_B,R2
        MOV     #PSUCLEAR,R0              ; Clear PSU
        MOV.B   R0,@(FLMCR1,GBR)
WAIT_12 DT      R2                        ; Wait 5 µs
        BF      WAIT_12
;
        MOV.L   #WDT_TCSR,R0              ; Disable WDT
        MOV.W   #H'A55F,R1
        MOV.W   R1,@R0
;
NO_ADWRT2       .EQU  $
        CMP/PL  R5                        ; If M = 0, end of programming
        BF      PROGRAM_OK
        ADD     #1,R14
        MOV     #NG,R13                   ; Move NG (return value) to R13
        MOV.L   #MAXVERIFY,R3             ; If N ≥ 1000, programming error
        CMP/GT  R14,R3
        BF      PROGRAM_END
;
        BRA     PROGRAM_LOOP
        NOP
;
PROGRAM_OK      .EQU  $
        MOV     #OK,R13                 ; Move OK (return value) to R13
PROGRAM_END     .EQU  $
        MOV     #SWECLEAR,R0            ; Clear SWE
        MOV.B   R0,@(FLMCR1,GBR)
;
        MOV.L   #WAIT_F,R2
WAIT_13 DT      R2                      ; Wait 100 µs
        BF      WAIT_13
;
        RTS
        NOP
;
ADATABUFF       .RES.B 128              ; Additional programming RAM area
RDATABUFF       .RES.B 128              ; Reprogramming RAM area
```

19.7.3 Erase Mode (n = 1 for Addresses H'00000 to H'07FFF, n = 2 for Addresses H'08000 to H'3FFFF)

When erasing flash memory, the erase/erase-verify flowchart (single-block erase) shown in figure 19.14 should be followed for each block.

To perform data or program erasure, set the flash memory area to be erased in erase block register n (EBRn) at least 1 µs after setting the SWE bit to 1 in flash memory control register 1 (FLMCR1). Next, the watchdog timer is set to prevent overprogramming in the event of program runaway, etc. After this, preparation for erase mode (erase setup) is carried out by setting the ESU bit in FLMCR1, and after the elapse of 100 µs or more, the operating mode is switched to erase mode by setting the E bit in FLMCR1. The time during which the E bit is set is the flash memory erase time. Ensure that the erase time does not exceed 10 ms.

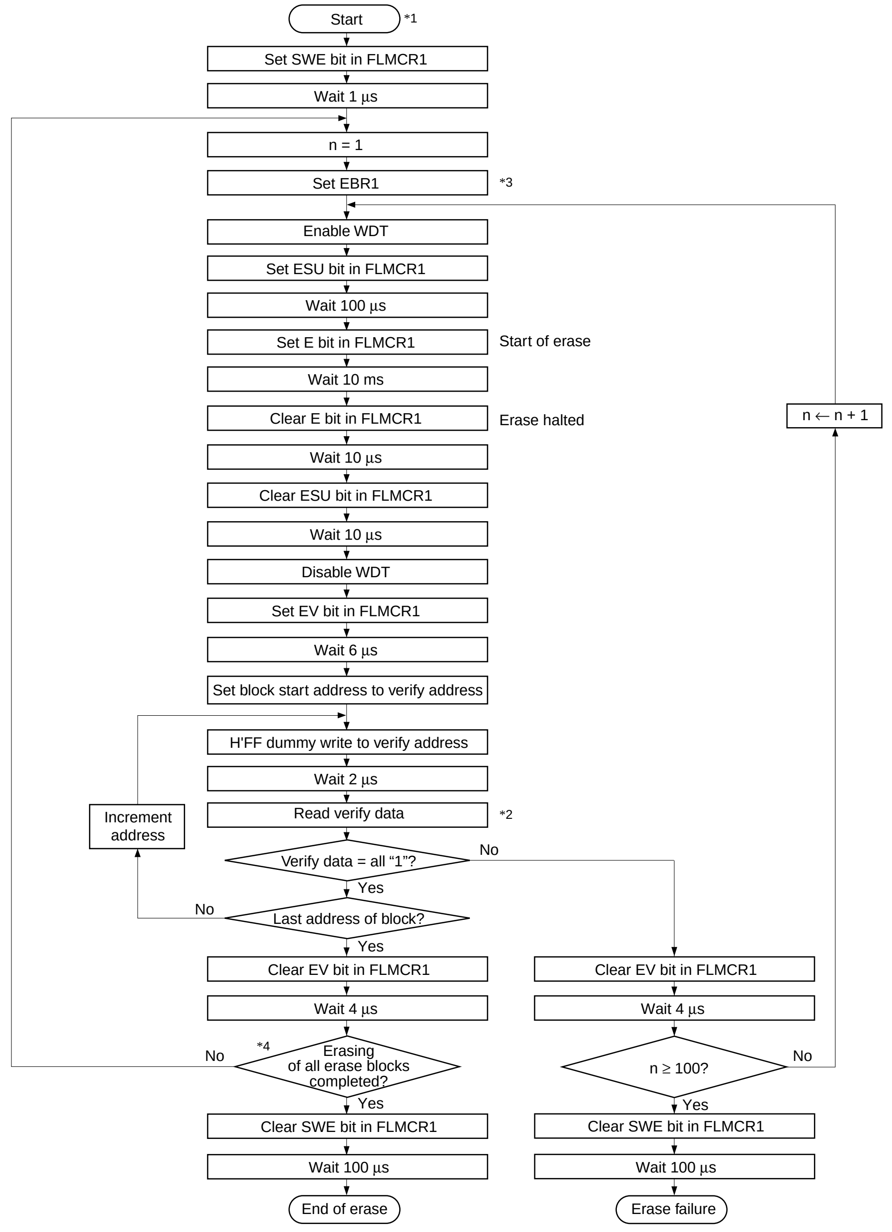
Note: With flash memory erasing, preprogramming (setting all memory data in the memory to be erased to all “0”) is not necessary before starting the erase procedure.

19.7.4 Erase-Verify Mode (n = 1 for Addresses H'00000 to H'07FFF, n = 2 for Addresses H'08000 to H'3FFFF)

In erase-verify mode, data is read after memory has been erased to check whether it has been correctly erased.

After the elapse of the erase time, erase mode is exited (the E bit in FLMCR1 is cleared, then the ESU bit is cleared at least 10 µs later). After the elapse of 10 µs or more, the watchdog timer is cleared, and the operating mode is switched to erase-verify mode by setting the EV bit in FLMCR1. Before reading in erase-verify mode, a dummy write of H'FF data should be made to the addresses to be read. The dummy write should be executed after the elapse of 6 µs or more. When the flash memory is read in this state (verify data is read in 32-bit units), the data at the latched address is read. Wait at least 2 µs after the dummy write before performing this read operation. If the read data has been erased (all “1”), a dummy write is performed to the next address, and erase-verify is performed.

If the read data has not been erased, set erase mode again and repeat the erase/erase-verify sequence as before. However, ensure that this operation is not repeated more than 100 times. When verification is completed, exit erase-verify mode, and wait for at least 4 µs. When erasure has been completed on all the erase blocks, clear the SWE bit in FLMCR1. If there are any unerased blocks, make a 1-bit setting for the flash memory area to be erased, and repeat the erase/erase-verify sequence as before.



Notes: 1. Preprogramming (setting erase block data to all "0") is not necessary.
2. Verify data is read in 32-bit (longword) units.
3. Set only one bit in the erase block register (EBR). Two or more bits must not be set.
4. Erasing is performed in block units. For a multiple-block erase, the individual blocks must be erased sequentially.

Figure 19.14 Erase/Erase-Verify Flowchart (Single-Block Erase)

Sample Single-Block Erase Program

The wait time set values (number of loops) are for the case where f = 60 MHz. For other frequencies, the set value is given by {wait time (μs) × f (MHz) ÷ 4}.

Registers Used
R5 (input): Memory block table pointer
R7 (output): OK (normal) or NG (error)
R0–3, 6, 8–9: Work registers

```
FLMCR1                 .EQU              H'00
EBR1                   .EQU              H'02
OK                     .EQU              H'0
NG                     .EQU              H'1
EWait_X                .EQU              15
EWait_Y                .EQU              1500
EWait_Z                .EQU              150000
EWait_a                .EQU              150
EWait_b                .EQU              150
EWait_c                .EQU              90
EWait_d                .EQU              30
EWait_e                .EQU              60
EWait_f                .EQU              1500
WDT_TCSR                EQU              H'FFFF1000
WDT_4m                  EQU              H'A57D
SWESET                 .EQU              B'01000000
ESUSET                 .EQU              B'00100000
ESET                   .EQU              B'00000010
ECLEAR                 .EQU              B'11111101
ESUCLEAR               .EQU              B'11011111
EVSET                  .EQU              B'00001000
EVCLEAR                .EQU              B'11110111
SWECLEAR               .EQU              B'10111111
MAXErase               .EQU              100
;
FlashErase             .EQU              $
             MOV.L     #H'FFFF0800,R0
             LDC       R0,GBR                     ; Initialize GBR
             MOV.L     #1,R2
;
             MOV,L     #EWait_X,R3
             MOV.L     #FLMCR1,R0
             OR.B      #SWESET,@(R0,GBR)          ; Set SWE
EWait_1      SUBC      R2,R3                      ; Wait 1 μs
             BF        EWait_1
;
             MOV.L     #0,R9                      ; Initialize n (R9) to 0
;
             MOV.W     @(6,R5),R0
             MOV.W     R0,@(EBR1,GBR)             ; Erase memory block (EBR1/2) setting
             MOV.L     @R5,R6                     ; Erase memory block start address →
                                                  ; R6 setting
;
EraseLoop              .EQU              $
             MOV.L     #WDT_TCSR,R1               ; Enable WDT
             MOV.W     #WDT_4m,R3                 ; 4.4 ms cycle
             MOV.W     R3,@R1
;
             MOV.L     #EWait_Y,R3
             MOV.L     #FLMCR1,R0
             OR.B      #ESUSET,@(R0,GBR)          ; Set ESU
EWait_2      SUBC      R2,R3                      ; Wait 100 μs
             BF        EWait_2
;
             MOV.L     #EWait_Z,R3
             OR.B      #ESET,@(R0,GBR)            ; Set E
EWait_3      SUBC      R2,R3                      ; Wait 10 ms
             BF        EWait_3
;
             MOV.L     #EWait_a,R3
             AND.B     #ECLEAR,@(R0,GBR)          ; Clear E
EWait_4      SUBC      R2,R3                      ; Wait 10 μs
```

```
             BF         EWait_4
;
             MOV.L      #EWait_b,R3
             AND.B      #ESUCLEAR,@(R0,GBR)    ; Clear ESU
EWait_5      SUBC       R2,R3                  ; Wait 10 µs
             BF         EWait_5
;
             MOV.L      #WDT_TCSR,R1           ; Disable WDT
             MOV.W      #H'A55F,R3
             MOV.W      R3,@R1
;
             MOV.L      #EWait_c,R3
             OR.B       #EVSET,@(R0,GBR)       ; Set EV
EWait_6      SUBC       R2,R3                  ; Wait 6 µs
             BF         EWait_6
;
BlockVerify_1           .EQU               $   ; Erase-verify
             MOV.L      #H'FFFFFFFF,R8
             MOV.L      R8,@R6                 ; H'FF dummy write
             MOV.L      #EWait_d,R3
EWait_7      SUBC       R2,R3                  ; Wait 2 µs
             BF         EWait_7
;
             MOV.L      @R6+,R1                ; Read verify data
             CMP/EQ     R8,R1
             BF         BlockVerify_NG
             MOV.L      @(8,R5),R7
             CMP/EQ     R6,R7                  ; Check for last address of memory block
             BF         BlockVerify_1
             MOV.L      #EWait_e,R3
             AND.B      #EVCLEAR,@(R0,GBR)     ; Clear EV
EWait_8      SUBC       R2,R3                  ; Wait 4 µs
             BF         EWait_8
;
             MOV.L      #OK,R7                 ; Move OK (return value) to R7
             BRA       FlashErase_end          ; Verify OK
             NOP
;
BlockVerify_NG         .EQU              $
             ADD.L     #1,R9                   ; If verify NG, assign n+1 to n
             ADD.L     #-4,R6                  ; Next verify address
             MOV.L     #EWait_e,R3
             AND.B     #EVCLEAR,@(R0,GBR)      ; Clear EV
EWait_9      SUBC      R2,R3                   ; Wait 4 µs
             BF        EWait_9
;
             MOV.L     #MAXErase,R7            ; If N > 100, erase error
             CMP/EQ    R7,R9
             BF        EraseLoop
             MOV.L     #NG,R7                  ; Move NG (return value) to R7
FlashErase_end         .EQU              $
             MOV.L     #FLMCR1,R0
             AND.B     #SWECLEAR,@(R0,GBR)     ; Clear SWE
             MOV.L     #Ewait_f,R3
Ewait_10     SUBC      R2,R3                   ; Wait 100 µs
             BF        Ewait_10
;
             RTS
             NOP
;
;Memory block table        Memory block start address: EBR value
             .ALIGN     4
Flash_BlockData         .EQU              $
EB0          .DATA.L    H'00000000,H'00000100
EB1          .DATA.L    H'00001000,H'00000200
EB2          .DATA.L    H'00002000,H'00000400
EB3          .DATA.L    H'00003000,H'00000800
EB4          .DATA.L    H'00004000,H'00001000
EB5          .DATA.L    H'00005000,H'00002000
EB6          .DATA.L    H'00006000,H'00004000
```

```
  EB7          .DATA.L    H'00007000,H'00008000
  EB8          .DATA.L    H'00008000,H'00000001
  EB9          .DATA.L    H'00010000,H'00000002
  EB10         .DATA.L    H'00020000,H'00000004
  EB11         .DATA.L    H'00030000,H'00000008
  Dummy        .DATA.L    H'00040000
```

19.7.5 Wait Time Widths in Programming/Erasing

Various wait time widths in the program/erase control program provided by the user should be within the specifications shown below.

Table 19.8 Programming/Erasing-Related Wait Width Specifications

Flow Section	Item	Symbol	Min	Typ	Max	Unit	Notes
Programming-related	Wait time after PSU bit setting	tspsu	50	50	—	µs	
	Wait time after P bit setting (10 µs)	tsp10	8	10	12	µs	Additional-programming time wait
	Wait time after P bit setting (30 µs)	tsp30	28	30	32	µs	Programming time wait
	Wait time after P bit setting (200 µs)	tsp200	198	200	202	µs	Programming time wait
	Wait time after P bit clearing	tcp	5	5	—	µs	
	Wait time after PSU bit clearing	tcpsu	5	5	—	µs	
	Wait time after PV bit setting	tspv	4	4	—	µs	
	Wait time after dummy write	tspvr	2	2	—	µs	
	Wait time after PV bit clearing	tcpv	2	2	—	µs	
Erase-related	Wait time after ESU bit setting	tsesu	100	100	—	µs	
	Wait time after E bit setting	tse	10	10	100	ms	Erase time wait
	Wait time after E bit clearing	tce	10	10	—	µs	
	Wait time after ESU bit clearing	tcesu	10	10	—	µs	
	Wait time after EV bit setting	tsev	6	6	—	µs	
	Wait time after dummy write	tsevr	2	2	—	µs	
	Wait time after EV bit clearing	tcev	4	4	—	µs	
Other (common)	Wait time after SWE bit setting	tsswe	1	1	—	µs	
	Wait time after SWE bit clearing	tcswe	100	100	—	µs	

19.8 Protection

There are two kinds of flash memory program/erase protection, hardware protection and software protection.

19.8.1 Hardware Protection

Hardware protection refers to a state in which programming/erasing of flash memory is forcibly disabled or aborted. Hardware protection is set by settings in flash memory control register 1 (FLMCR1), erase block register 1 (EBR1), and erase block register 2 (EBR2). The FLMCR1, EBR1, and EBR2 settings are retained in the error protection state. (See table 19.9.)

Table 19.9 Hardware Protection

		Functions	
Item	**Description**	**Program**	**Erase**
FWE pin protection	• When a low level is input to the FWE pin, FLMCR1, EBR1, and EBR2 are initialized, and the program/erase-protected state is entered.	Yes	Yes
Reset/standby protection	• In a reset (including a WDT overflow reset) and in standby mode, FLMCR1, EBR1, and EBR2 are initialized, and the program/erase-protected state is entered. • In a reset via the $\overline{\text{RES}}$ pin, the reset state is not entered unless the $\overline{\text{RES}}$ pin is held low until oscillation stabilizes after powering on. In the case of a reset during operation, hold the $\overline{\text{RES}}$ pin low for the $\overline{\text{RES}}$ pulse width specified in the AC Characteristics section.	Yes	Yes

Note: "Yes" indicates a function for which protection is enabled.

19.8.2 Software Protection

Software protection can be implemented by setting the SWE bit in flash memory control register 1 (FLMCR1), erase block register 1 (EBR1), erase block register 2 (EBR2), and the RAMS bit in the RAM emulation register (RAMER). When software protection is in effect, setting the P or E bit in flash memory control register 1 (FLMCR1) does not cause a transition to program mode or erase mode. (See table 19.10.)

Table 19.10 Software Protection

		Functions	
Item	**Description**	**Program**	**Erase**
SWE bit protection	• Clearing the SWE bit to 0 in FLMCR1 sets the program/erase-protected state for all blocks. (Execute in on-chip RAM or external memory.)	Yes	Yes
Block specification protection	• Erase protection can be set for individual blocks by settings in erase block register 1 (EBR1) and erase block register 2 (EBR2). • Setting EBR1 and EBR2 to H'00 places all blocks in the erase-protected state.	—	Yes
Emulation protection	• Setting the RAMS bit to 1 in the RAM emulation register (RAMER) places all blocks in the program/erase-protected state.	Yes	Yes

Note: "Yes" indicates a function for which protection is enabled.

19.8.3 Error Protection

In error protection, an error is detected when SH7065 runaway occurs during flash memory programming/erasing, or operation is not performed in accordance with the program/erase algorithm, and the program/erase operation is aborted. Aborting the program/erase operation prevents damage to the flash memory due to overprogramming or overerasing.

If SH7065 malfunctions during flash memory programming/erasing, the FLER bit is set to 1 in FLMCR2 and the error protection state is entered. The FLMCR1, EBR1, and EBR2 settings are retained, but program mode or erase mode is aborted at the point at which the error occurred. Program mode or erase mode cannot be re-entered by re-setting the P or E bit. However, PV and EV bit setting is enabled, and a transition can be made to verify mode.

FLER bit setting conditions are as follows:

1. When flash memory is read during programming/erasing (including a vector read or instruction fetch)
2. Immediately after the start of exception handling (excluding a reset and hardware standby mode) during programming/erasing
3. When a SLEEP instruction (including software standby) is executed during programming/erasing
4. When the bus is released during programming/erasing

Error protection is released only by a reset or in hardware standby mode.

Figure 19.15 shows the flash memory state transition diagram.

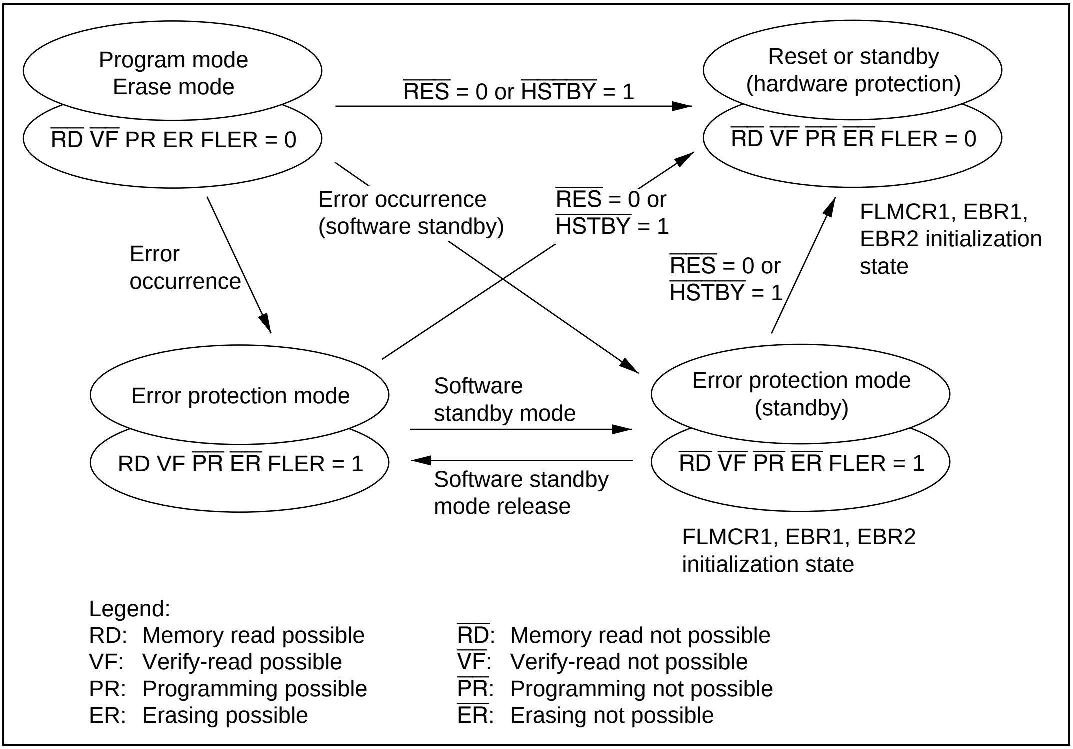


Figure 19.15 Flash Memory State Transitions

19.9 Flash Memory Emulation in RAM

Making a setting in the RAM emulation register (RAMER) enables part of RAM to be overlapped onto the flash memory area so that data to be written to flash memory can be emulated in RAM in real time. After the RAMER setting has been made, accesses can be made from the flash memory area or the RAM area overlapping flash memory. Emulation can be performed in user mode and user program mode. Figure 19.16 shows an example of emulation of real-time flash memory programming.

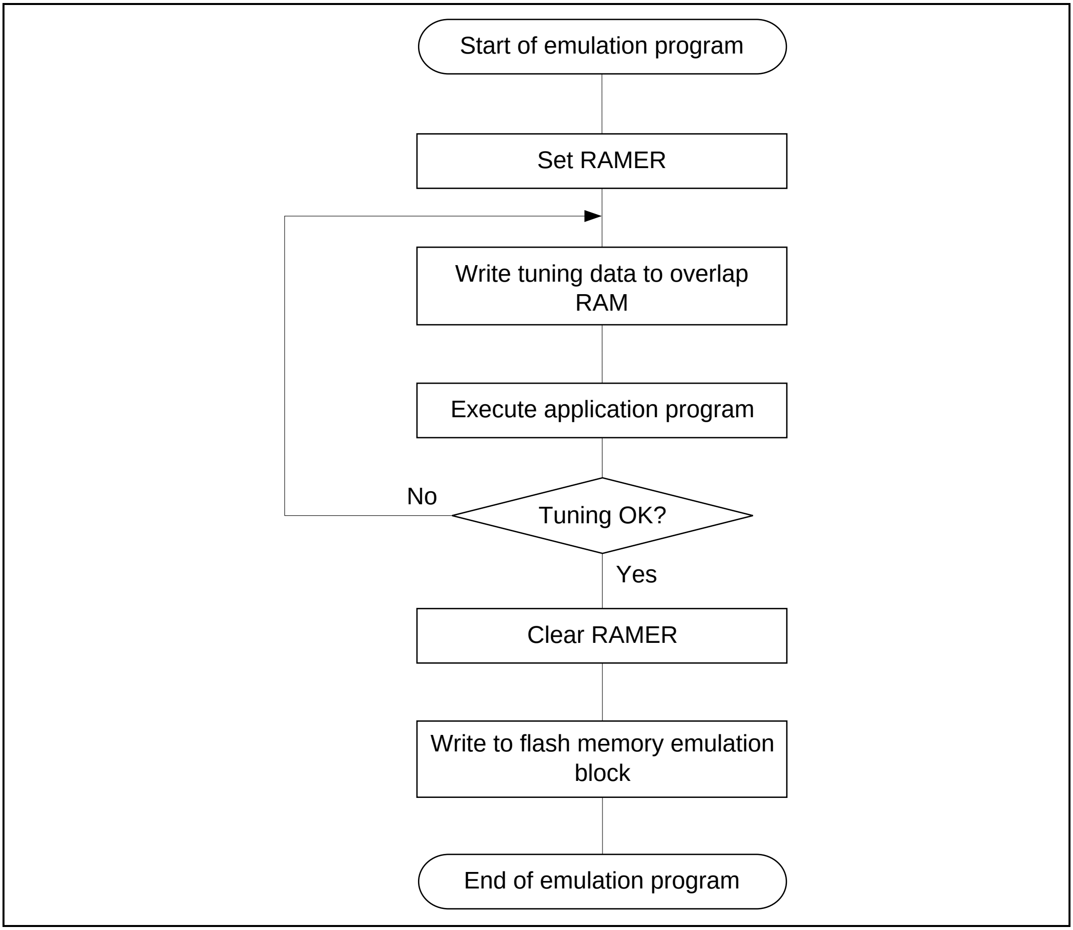

Start of emulation program
Set RAMER
Write tuning data to overlap RAM
Execute application program
No
Tuning OK?
Yes
Clear RAMER
Write to flash memory emulation block
End of emulation program

Figure 19.16 Flowchart of Flash Memory Emulation in RAM

Note: The number of execution cycles in emulation with RAM differs from the number of cycles when using flash memory.

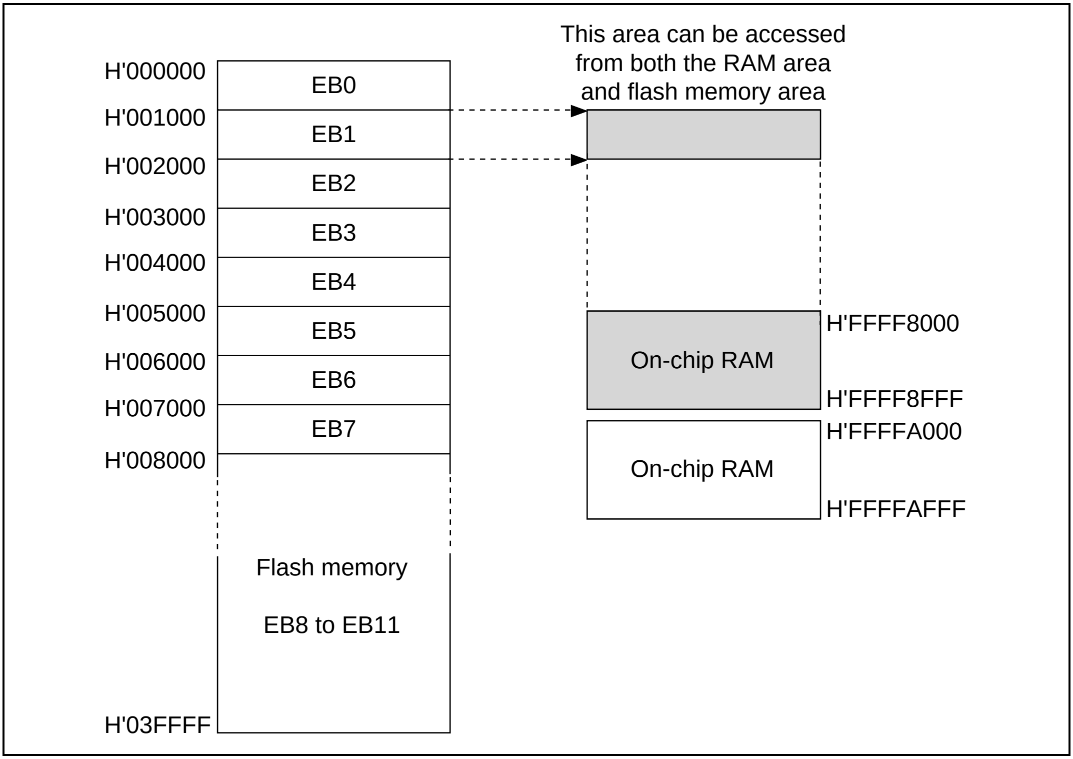


Figure 19.17 Example of RAM Overlap Operation

Example of Flash Memory Block Area EB1 Overlapping:

1. Set bits RAMAS, RAMS, and RAM2 to RAM0 in RAMER to 0,1, 0, 0, 1, to overlap RAM (H'FFFF8000 to H'FFFF8FFF) onto the area (EB1) for which real-time programming is required.
2. Real-time programming is performed using the overlapping RAM.
3. After the program data has been confirmed, the RAMS bit is cleared, releasing RAM overlap.
4. The data written in the overlapping RAM is written into the flash memory space (EB1).

Notes: 1. When the RAMS bit is set to 1, program/erase protection is enabled for all blocks regardless of the value of RAM2 to RAM0 (emulation protection). In this state, setting the P or E bit in flash memory control register 1 (FLMCR1) will not cause a transition to program mode or erase mode. When actually programming or erasing a flash memory area, the RAMS bit should be cleared to 0.

2. A RAM area cannot be erased by execution of software in accordance with the erase algorithm while flash memory emulation in RAM is being used.

19.10 Note on Flash Memory Programming/Erasing

In the on-board programming modes (boot mode and user program mode), NMI input should be disabled to give top priority to the program/erase operations (including RAM emulation).

19.11 Flash Memory Programmer Mode

Programs and data can be written and erased in programmer mode as well as in the on-board programming modes. In programmer mode, flash memory read mode, auto-program mode, auto-erase mode, and status read mode are supported. In auto-program mode, auto-erase mode, and status read mode, a status polling procedure is used, and in status read mode, detailed internal signals are output after execution of an auto-program or auto-erase operation.

In programmer mode, set the mode pins to crystal oscillation/PLLx4 mode (see table 19.11) and use a 6 MHz crystal oscillator, so that the SH7065 runs at 24 MHz.

Table 19.11 shows the pin settings for programmer mode. For the pin names in programmer mode, see figure 19.19.

Table 19.11 Programmer Mode Pin Settings

Pin Names	Settings
Clock pins: MD5 to MD3	MD5 = 1, MD4 = 0, MD3 = 1 (crystal oscillator, PLL2×4)
Mode pins: MD2 to MD0	MD2 = 1, MD1 = 1, MD0 = 1
FWE pin	High level input (in auto-program and auto-erase modes)
$\overline{\text{RES}}$ pin	Power-on reset circuit
XTAL, EXTAL, PLLV$_{CC}$, PLLCAP2, PLLV$_{SS}$ pins	Oscillator circuit

19.11.1 Socket Adapter Pin Correspondence Diagram

Connect the socket adapter to the chip as shown in figure 19.19. This will enable conversion to a 32-pin arrangement. The on-chip ROM memory map is shown in figure 19.18, and the socket adapter pin correspondence diagram in figure 19.19.

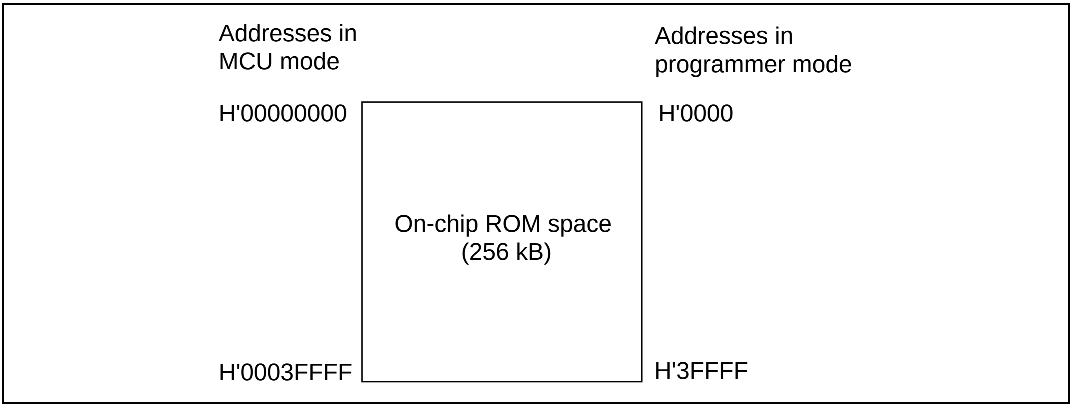
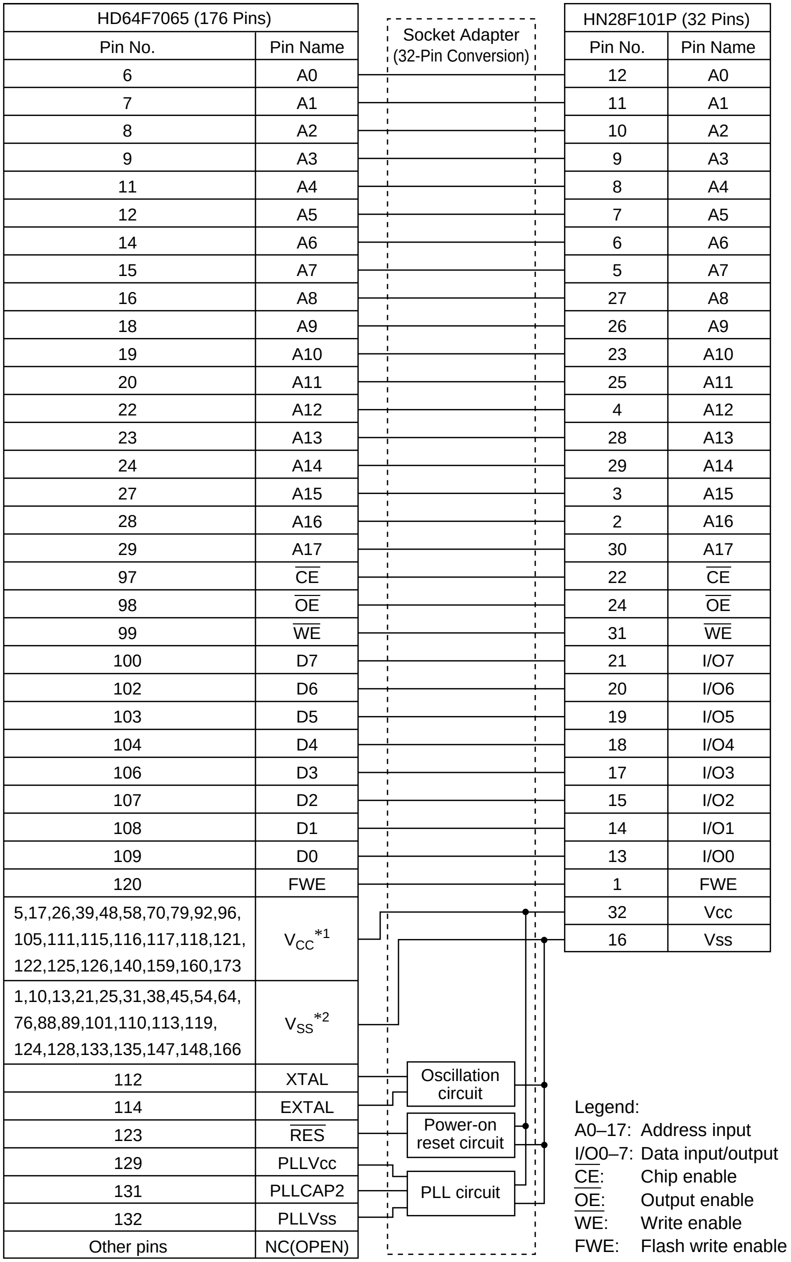


Figure 19.18 On-Chip ROM Memory Map



HD64F7065 (176 Pins)		HN28F101P (32 Pins)	
Pin No.	Pin Name	Pin No.	Pin Name
6	A0	12	A0
7	A1	11	A1
8	A2	10	A2
9	A3	9	A3
11	A4	8	A4
12	A5	7	A5
14	A6	6	A6
15	A7	5	A7
16	A8	27	A8
18	A9	26	A9
19	A10	23	A10
20	A11	25	A11
22	A12	4	A12
23	A13	28	A13
24	A14	29	A14
27	A15	3	A15
28	A16	2	A16
29	A17	30	A17
97	$\overline{\text{CE}}$	22	$\overline{\text{CE}}$
98	$\overline{\text{OE}}$	24	$\overline{\text{OE}}$
99	$\overline{\text{WE}}$	31	$\overline{\text{WE}}$
100	D7	21	I/O7
102	D6	20	I/O6
103	D5	19	I/O5
104	D4	18	I/O4
106	D3	17	I/O3
107	D2	15	I/O2
108	D1	14	I/O1
109	D0	13	I/O0
120	FWE	1	FWE
5,17,26,39,48,58,70,79,92,96, 105,111,115,116,117,118,121, 122,125,126,140,159,160,173	V_{CC}*1	32	Vcc
1,10,13,21,25,31,38,45,54,64, 76,88,89,101,110,113,119, 124,128,133,135,147,148,166	V_{SS}*2	16	Vss
112	XTAL		
114	EXTAL		
123	$\overline{\text{RES}}$		
129	PLLVcc		
131	PLLCAP2		
132	PLLVss		
Other pins	NC(OPEN)		

Legend:	
A0–17:	Address input
I/O0–7:	Data input/output
$\overline{\text{CE}}$:	Chip enable
$\overline{\text{OE}}$:	Output enable
$\overline{\text{WE}}$:	Write enable
FWE:	Flash write enable

Notes: 1. Including NMI, $\overline{\text{HSTBY}}$, MD0, MD1, MD2, MD3, and MD5
2. Including MD4

Figure 19.19 Socket Adapter Pin Correspondence Diagram

19.11.2 Operation in Programmer Mode

Table 19.12 shows how the different operating modes are set when using programmer mode, and table 19.13 lists the commands used in programmer mode. Details of each mode are given below.

- Memory Read Mode

  Memory read mode supports byte reads.

- Auto-Program Mode

  Auto-program mode supports programming of 128 bytes at a time. Status polling is used to confirm the end of auto-programming.

- Auto-Erase Mode

  Auto-erase mode supports automatic erasing of the entire flash memory mat. Status polling is used to confirm the end of auto-erasing.

- Status Read Mode

  Status polling is used for auto-programming and auto-erasing, and normal termination can be confirmed by reading the I/O6 signal. In status read mode, error information is output if an error occurs.

Table 19.12 Operating Mode Settings in Programmer Mode

	Pin Names					
Mode	**FWE**	**$\overline{CE}$**	**$\overline{OE}$**	**$\overline{WE}$**	**I/O7–0**	**A17–0**
Read	H or L	L	L	H	Data output	Ain
Output disable	H or L	L	H	H	Hi-Z	Ain
Command write	H or L	L	H	L	Data input	*Ain
Chip disable	H or L	H	X	X	Hi-Z	Ain

Notes: 1. Chip disable is not a standby state; internally, it is an operation state.

2. *Ain indicates that there is also address input in auto-program mode.

3. For command writes in auto-program and auto-erase modes, input a high level to the FWE pin.

Table 19.13 Programmer Mode Commands

Command Name	Number of Cycles	1st Cycle Mode	1st Cycle Address	1st Cycle Data	2nd Cycle Mode	2nd Cycle Address	2nd Cycle Data
Memory read mode	1 + n	Write	X	H'00	Read	RA	Dout
Auto-program mode	129	Write	X	H'40	Write	WA	Din
Auto-erase mode	2	Write	X	H'20	Write	X	H'20
Status read mode	2	Write	X	H'71	Write	X	H'71

Notes: 1. In auto-program mode, 129 cycles are required for command writing by a simultaneous 128-byte write.

2. In memory read mode, the number of cycles depends on the number of address write cycles (n).

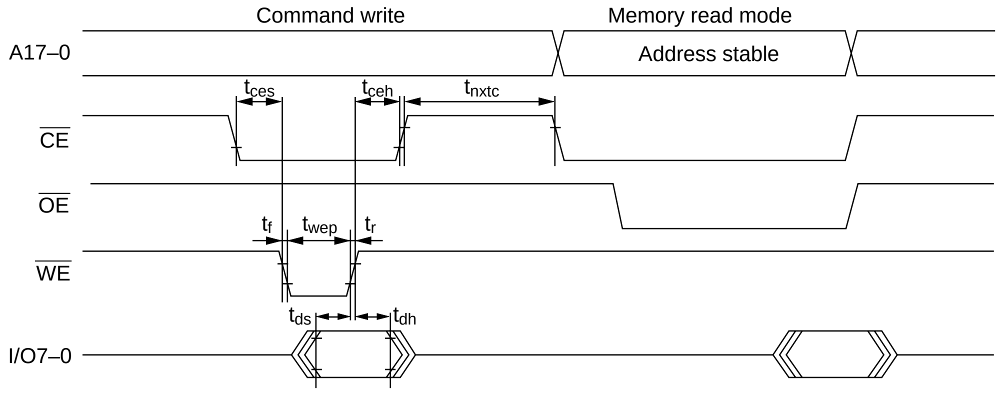
19.11.3 Memory Read Mode

1. After completion of auto-program/auto-erase/status read operations, a transition is made to the command wait state. When reading memory contents, a transition to memory read mode must first be made with a command write, after which the memory contents are read.
2. In memory read mode, command writes can be performed in the same way as in the command wait state.
3. Once memory read mode has been entered, consecutive reads can be performed.
4. After powering on, memory read mode is entered.

Table 19.14 AC Characteristics in Memory Read Mode

Conditions: V_{CC} = 3.3 V ±0.3 V, V_{SS} = 0 V, T_a = 25°C ±5°C

Item	Symbol	Min	Max	Unit	Notes
Command write cycle	t_{nxtc}	20	—	µs	
$\overline{\text{CE}}$ hold time	t_{ceh}	0	—	ns	
$\overline{\text{CE}}$ setup time	t_{ces}	0	—	ns	
Data hold time	t_{dh}	50	—	ns	
Data setup time	t_{ds}	50	—	ns	
Write pulse width	t_{wep}	70	—	ns	
$\overline{\text{WE}}$ rise time	t_r	—	30	ns	
$\overline{\text{WE}}$ fall time	t_f	—	30	ns	



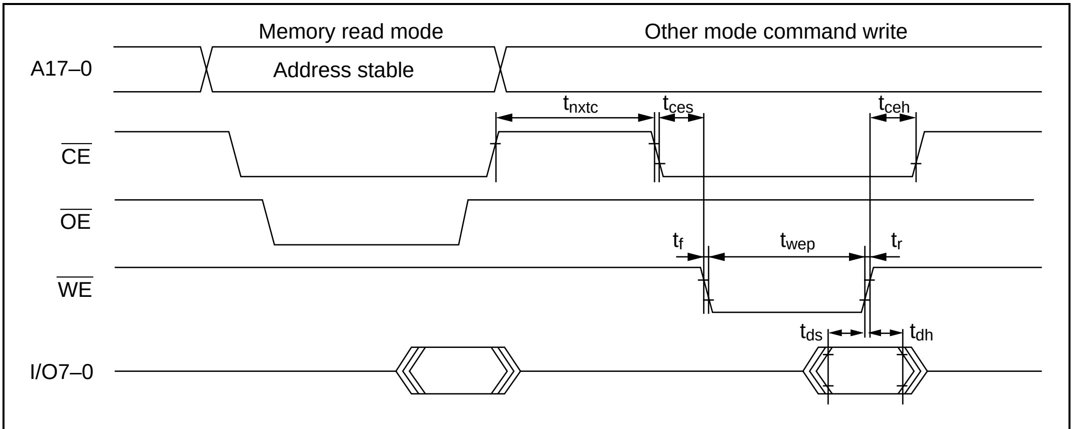
Note: Data is latched at the rising edge of $\overline{\text{WE}}$.

Figure 19.20 Memory Read Mode Timing Waveforms after Command Write

Table 19.15 AC Characteristics in Transition from Memory Read Mode to Another Mode

Conditions: V_{CC} = 3.3 V ±0.3 V, V_{SS} = 0 V, T_a = 25°C ±5°C

Item	Symbol	Min	Max	Unit	Notes
Command write cycle	t_{nxtc}	20	—	µs	
$\overline{\text{CE}}$ hold time	t_{ceh}	0	—	ns	
$\overline{\text{CE}}$ setup time	t_{ces}	0	—	ns	
Data hold time	t_{dh}	50	—	ns	
Data setup time	t_{ds}	50	—	ns	
Write pulse width	t_{wep}	70	—	ns	
$\overline{\text{WE}}$ rise time	t_r	—	30	ns	
$\overline{\text{WE}}$ fall time	t_f	—	30	ns	



Note: Do not enable $\overline{\text{CE}}$ and $\overline{\text{OE}}$ simultaneously.

Figure 19.21 Timing Waveforms in Transition from Memory Read Mode to Another Mode

Table 19.16 AC Characteristics in Memory Read Mode

Conditions: V_{CC} = 3.3 V ±0.3 V, V_{SS} = 0 V, T_a = 25°C ±5°C

Item	Symbol	Min	Max	Unit	Notes
Access time	t_{acc}	—	20	µs	
$\overline{CE}$ output delay time	t_{ce}	—	150	ns	
$\overline{OE}$ output delay time	t_{oe}	—	150	ns	
Output disable delay time	t_{df}	—	100	ns	
Data output hold time	t_{oh}	5	—	ns	

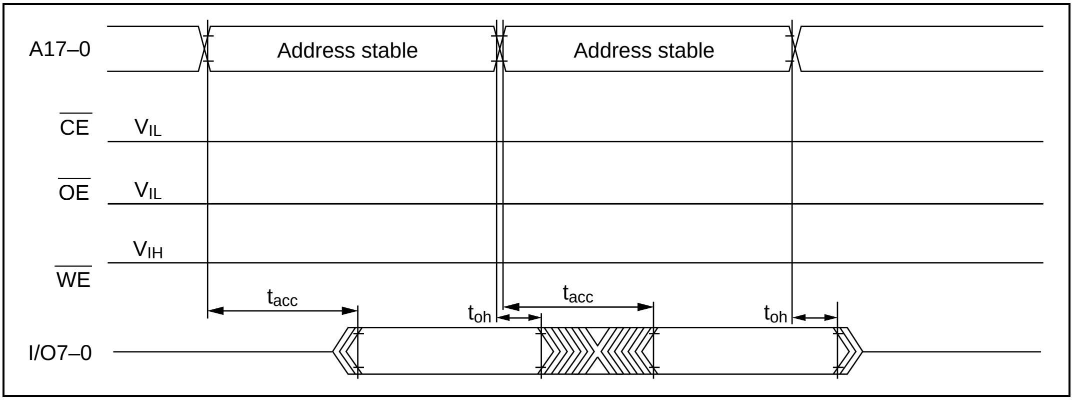


Figure 19.22 Timing Waveforms in $\overline{CE}$ and $\overline{OE}$ Enable State Read

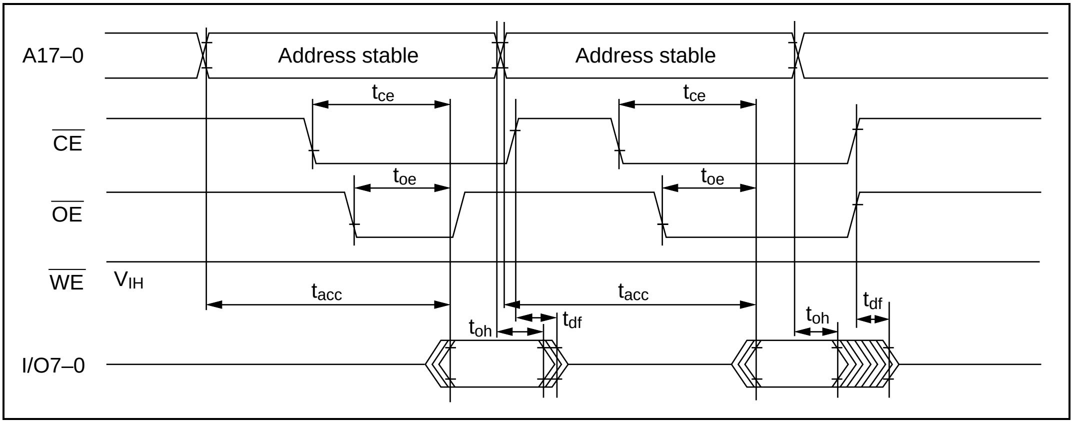


Figure 19.23 Timing Waveforms in $\overline{CE}$ and $\overline{OE}$ Clock System Read

19.11.4 Auto-Program Mode

1. In auto-program mode, 128 bytes are programmed simultaneously. This should be carried out by executing 128 consecutive byte transfers.
2. A 128-byte data transfer is necessary even when programming fewer than 128 bytes. In this case, H'FF data must be written to the extra addresses.
3. The lower 8 bits of the transfer address must be H'00 or H'80. If a value other than an effective address is input, processing will switch to a memory write operation but a write error will be flagged.
4. Memory address transfer is performed in the second cycle (figure 19.24). Do not perform transfer after the second cycle.
5. Do not perform a command write during a programming operation.
6. Perform one auto-programming operation for a 128-byte block for each address. One or more additional programming operations cannot be carried out on address blocks that have already been programmed.
7. Confirm normal end of auto-programming by checking I/O6. Alternatively, status read mode can also be used for this purpose (I/O7 status polling uses the auto-program operation end identification pin).
8. Status polling I/O6 and I/O7 pin information is retained until the next command write. As long as the next command write has not been performed, reading is possible by enabling $\overline{\text{CE}}$ and $\overline{\text{OE}}$.

Table 19.17 AC Characteristics in Auto-Program Mode

Conditions: V_{CC} = 3.3 V ±0.3 V, V_{SS} = 0 V, T_a = 25°C ±5°C

Item	Symbol	Min	Max	Unit	Notes
Command write cycle	t_{nxtc}	20	—	µs	
$\overline{\text{CE}}$ hold time	t_{ceh}	0	—	ns	
$\overline{\text{CE}}$ setup time	t_{ces}	0	—	ns	
Data hold time	t_{dh}	50	—	ns	
Data setup time	t_{ds}	50	—	ns	
Write pulse width	t_{wep}	70	—	ns	
Status polling start time	t_{wsts}	1	—	ms	
Status polling access time	t_{spa}	—	150	ns	
Address setup time	t_{as}	0	—	ns	
Address hold time	t_{ah}	60	—	ns	
Memory write time	t_{write}	1	3000	ms	
Write setup time	t_{pns}	100	—	ns	
Write end setup time	t_{pnh}	100	—	ns	
$\overline{\text{WE}}$ rise time	t_r	—	30	ns	
$\overline{\text{WE}}$ fall time	t_f	—	30	ns	

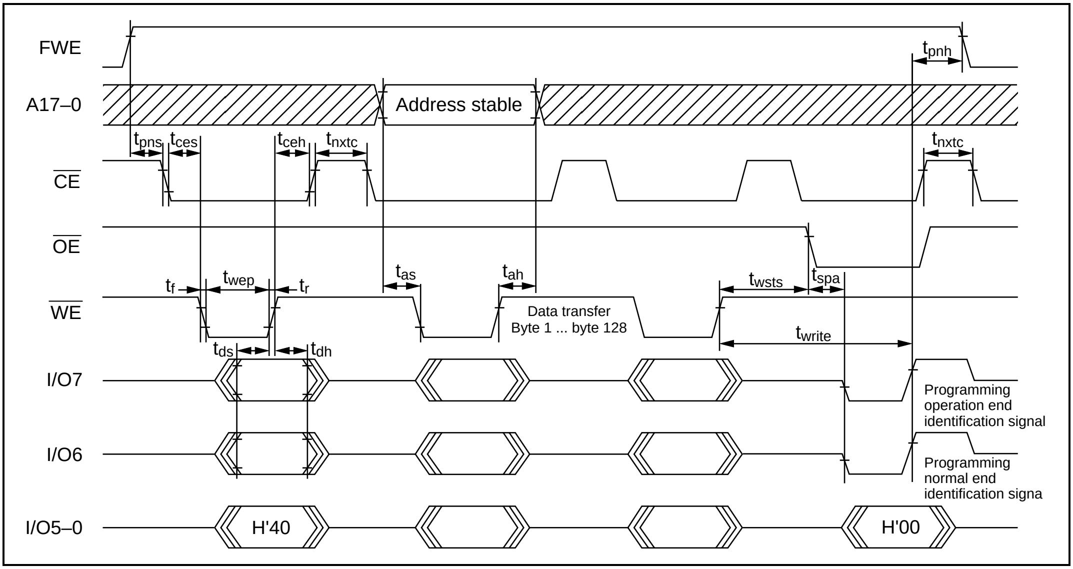


Figure 19.24 Auto-Program Mode Timing Waveforms

19.11.5 Auto-Erase Mode

1. Auto-erase mode supports only total memory erasing.
2. Do not perform a command write during auto-erasing.
3. Confirm normal end of auto-erasing by checking I/O6. Alternatively, status read mode can also be used for this purpose (I/O7 status output is used to identify the end of an auto-erase operation).
4. Status polling I/O6 and I/O7 pin information is retained until the next command write. As long as the next command write has not been performed, reading is possible by enabling $\overline{\text{CE}}$ and $\overline{\text{OE}}$.

Table 19.18 AC Characteristics in Auto-Erase Mode

Conditions: V_{CC} = 3.3 V ±0.3 V, V_{SS} = 0 V, T_a = 25°C ±5°C

Item	Symbol	Min	Max	Unit	Notes
Command write cycle	t_{nxtc}	20	—	µs	
$\overline{\text{CE}}$ hold time	t_{ceh}	0	—	ns	
$\overline{\text{CE}}$ setup time	t_{ces}	0	—	ns	
Data hold time	t_{dh}	50	—	ns	
Data setup time	t_{ds}	50	—	ns	
Write pulse width	t_{wep}	70	—	ns	
Status polling start time	t_{wsts}	1	—	ms	
Status polling access time	t_{spa}	—	150	ns	
Memory erase time	t_{erase}	100	40000	ms	
Erase setup time	t_{ens}	100	—	ns	
Erase end setup time	t_{enh}	100	—	ns	
$\overline{\text{WE}}$ rise time	t_r	—	30	ns	
$\overline{\text{WE}}$ fall time	t_f	—	30	ns	

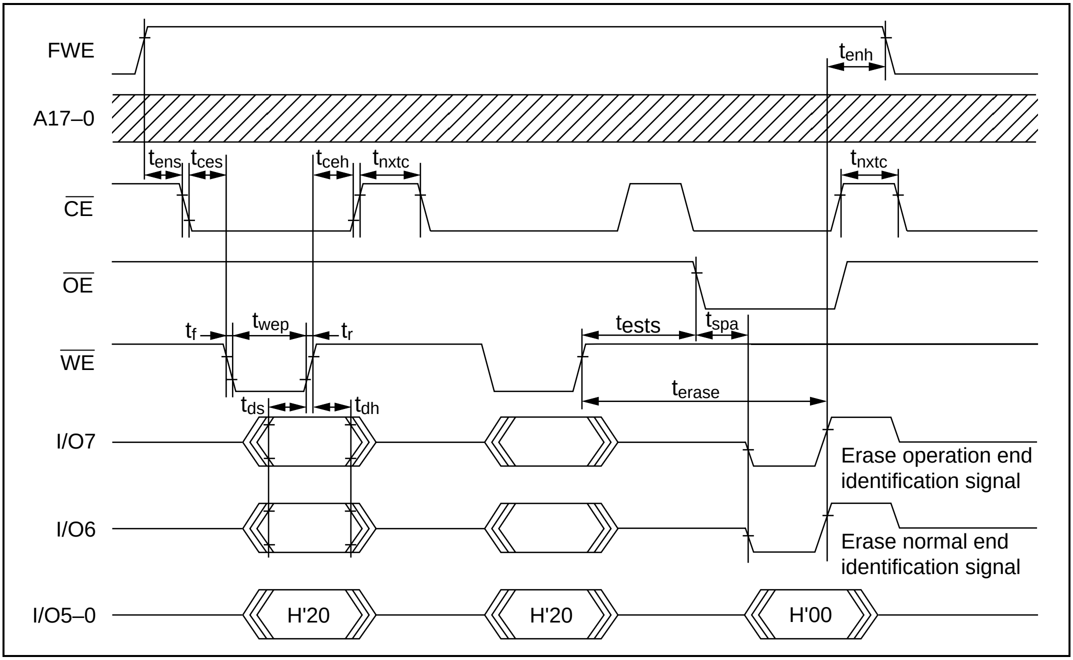


Figure 19.25   Auto-Erase Mode Timing Waveforms

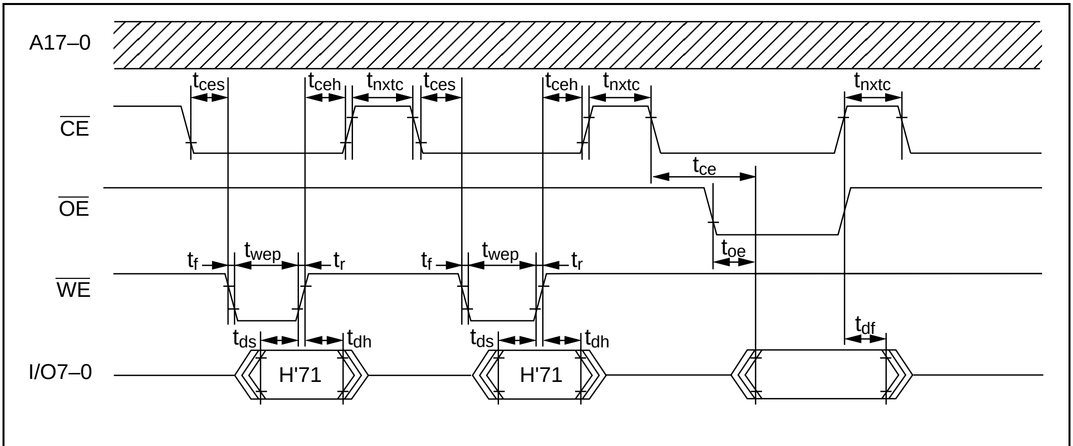
19.11.6 Status Read Mode

1. Status read mode is provided to indicate the type of an abnormal end. It should be used if an error occurs in auto-program or auto-erase mode.
2. The return code is retained until a command write is performed in a mode other than status read mode.

Table 19.19 AC Characteristics in Status Read Mode

Conditions: V_{CC} = 3.3 V ±0.3 V, V_{SS} = 0 V, T_a = 25°C ±5°C

Item	Symbol	Min	Max	Unit	Notes
Read time after command write	t_{nxtc}	20	—	µs	
$\overline{CE}$ hold time	t_{ceh}	0	—	ns	
$\overline{CE}$ setup time	t_{ces}	0	—	ns	
Data hold time	t_{dh}	50	—	ns	
Data setup time	t_{ds}	50	—	ns	
Write pulse width	t_{wep}	70	—	ns	
$\overline{OE}$ output delay time	t_{oe}	—	150	ns	
Disable delay time	t_{df}	—	100	ns	
$\overline{CE}$ output delay time	t_{ce}	—	150	ns	
$\overline{WE}$ rise time	t_r	—	30	ns	
$\overline{WE}$ fall time	t_f	—	30	ns	



Note: I/O3 and I/O2 are undefined.

Figure 19.26 Timing Waveforms in Status Read Mode

Table 19.20 Status Read Mode Return Commands

Pin Name	I/O7	I/O6	I/O5	I/O4	I/O3	I/O2	I/O1	I/O0
Attribute	Normal end identification	Command error	Programming error	Erase error	—	—	Programming or erase count exceeded	Effective address error
Initial value	0	0	0	0	0	0	0	0
Indications	Normal end: 0 Abnormal end: 1	Command error: 1 Otherwise: 0	Programming error: 1 Otherwise: 0	Erasing error: 1 Otherwise: 0	—	—	Count exceeded: 1 Otherwise: 0	Effective address error: 1 Otherwise: 0

Note: I/O3 and I/O2 are undefined.

19.11.7 Status Polling

1. I/O7 status polling is a flag that indicates the operating status in auto-program/auto-erase mode.
2. I/O6 status polling is a flag that indicates a normal or abnormal end in auto-program/auto-erase mode.

Table 19.21 Status Polling Output Truth Table

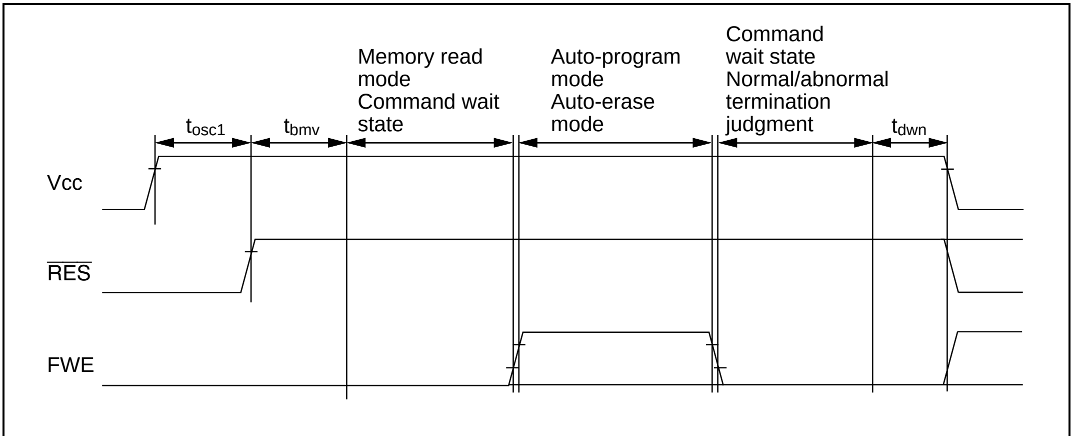
Pin Name	During Internal Operation	Abnormal End	—	Normal End
I/O7	0	1	0	1
I/O6	0	0	1	1
I/O5–0	0	0	0	0

19.11.8 Programmer Mode Transition Time

Commands cannot be accepted during the oscillation settling period or the programmer mode setup period. After the programmer mode setup time, a transition is made to memory read mode.

Table 19.22 Stipulated Transition Times to Command Wait State

Item	Symbol	Min	Max	Unit	Notes
Standby release (oscillation settling time)	t_{osc1}	30	—	ms	
Programmer mode setup time	t_{bmv}	10	—	ms	
V_{CC} hold time	t_{dwn}	0	—	ms	


Memory read mode Command wait state
Auto-program mode Auto-erase mode
Command wait state Normal/abnormal termination judgment
t_{osc1}
t_{bmv}
t_{dwn}
Vcc
$\overline{RES}$
FWE

Note: Except in auto-program mode and auto-erase mode, the FWE input pin should be driven low.

Figure 19.27 Oscillation Settling Time and Boot Program Transfer Time

19.11.9 Cautions Concerning Memory Programming

1. When programming addresses which have previously been programmed, carry out auto-erasing before auto-programming.
2. When performing programming using programmer mode on a chip that has been programmed/erased in an on-board programming mode, auto-erasing is recommended before carrying out auto-programming.

Notes: 1. The flash memory is initially in the erased state when the device is shipped by Renesas. For other chips for which the erasure history is unknown, it is recommended that auto-erasing be executed to check and supplement the initialization (erase) level.

2. Auto-programming should be performed once only on the same address block. Additional programming cannot be performed on previously programmed address blocks.

19.12 Usage Notes

1. Do not release the FWE pin or clear the SWE bit during programming, erasing, or verifying (FWE = 1, SWE = 1).
2. If the reset or HSTBY signal is input during programming, erasing, or verifying (FWE = 1, SWE = 1), the signal must be input continuously for at least 100 µs.
3. When reading flash memory (including a reset start) immediately after a transition from program/erase/verify mode (FWE = 1, SWE = 1) to normal mode (FWE = 0, SWE = 0), allow a wait of at least 100 µs.
4. When recovery from the flash memory module's module standby state is performed by clearing the MSTP bit (see section 4.13, Module Standby Function), wait at least 100 µs before reading the flash memory.

19.13 Cautions on Transition from F-ZTAT to Mask ROM Version

When switching from the F-ZTAT version to a mask ROM version product, care is needed when using F-ZTAT application software.

If an address at which a flash memory register resides (see table 19.3, Flash Memory Registers) is read in the mask ROM version, the read value will be undefined. If F-ZTAT version application software is used in a mask ROM version product, the state of the FWE pin cannot be determined. Program modifications must be made to ensure that the flash memory programming (erase/write) sections and RAM emulation section are not activated. Also ensure that mode pin (FWE, MD2 to MD0) settings are not made for user program mode, boot mode, or PROM mode (programmer mode) in the mask ROM version.

Note: The above applies to all F-ZTAT version products and mask ROM version products with different ROM sizes within the same series.

Section 20 256 kB Mask ROM

20.1 Overview

The SH7065 has 256 kbytes of on-chip mask ROM. The on-chip ROM is connected to the CPU and direct memory access controller (DMAC) by a 32-bit-wide data bus (figure 20.1). The CPU and DMAC can use 8-, 16-, or 32-bit access to the on-chip ROM.

The on-chip ROM is 64 bits wide, and is accessed in two cycles. It is connected to the internal data bus (C-bus) via an on-chip ROM buffer, and has a basic access capability of 32 bits per cycle.

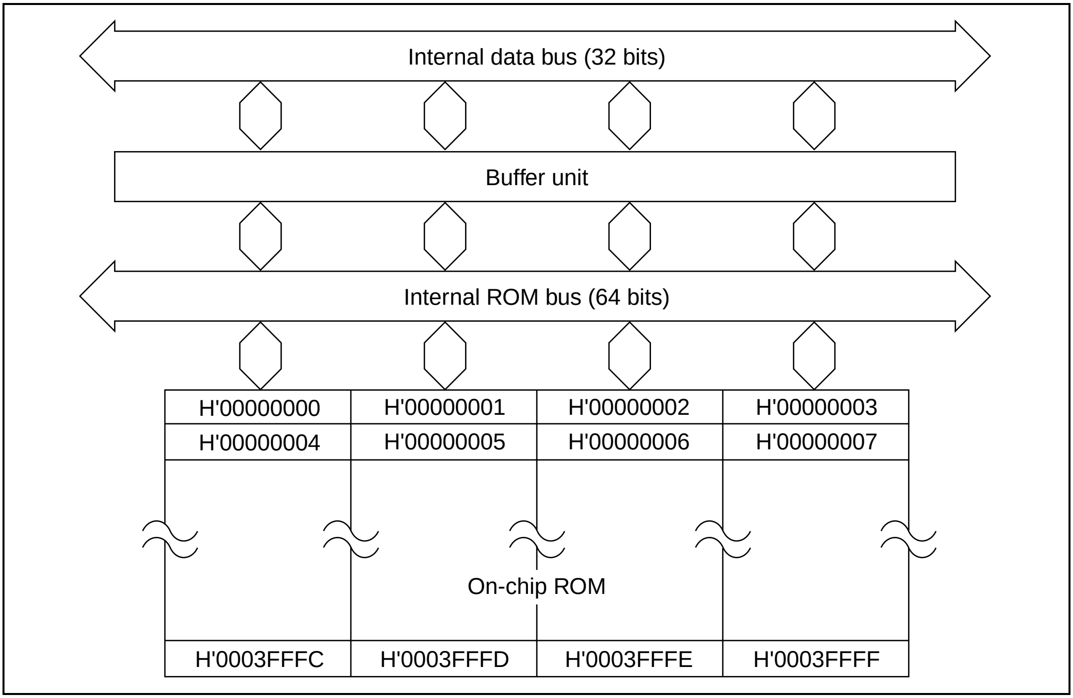


Figure 20.1 Block Diagram of On-Chip ROM (256 KB Version)

The on-chip ROM is enabled or disabled depending on the operating mode. For details of the operating modes, see section 3, Operating Modes. The mode is selected by mode setting pins MD3 to MD0, as shown in figure 20.1. Select mode 0 or 1 when on-chip ROM is used, and mode 2, 3, or 4 when it is not used. The on-chip ROM is allocated to addresses H'00000000 to H'0003FFFF (256 kB version) in memory area 0.

Section 21 XRAM and YRAM

21.1 Overview

The SH7065 has 4 kbytes each of XRAM and YRAM. The XRAM and YRAM are connected to the CPU and DSP by a 16-bit X-bus and Y-bus, respectively, (figures 21.1 and 21.2), allowing 16-bit-wide data exchange with the CPU and DSP. The XRAM and YRAM are also connected to the CPU by a 32-bit-wide internal data bus (CDB) and to the direct memory access controller (DMAC) by a 32-bit internal data bus (IDB) (figures 21.1 and 21.2), allowing 8-, 16-, or 32-bit-wide access to the XRAM and YRAM.

The XRAM and YRAM can be accessed in parallel using the X-bus and Y-bus, enabling two data transfer instructions to be executed simultaneously. XRAM and YRAM data can always be accessed in one state, making this memory suitable for use in areas requiring high-speed access. The contents of XRAM and YRAM are retained in sleep mode and standby mode.

The XRAM and YRAM are allocated to addresses H'FFFF8000 to H'FFFF8FFF and H'FFFFA000 to H'FFFFAFFF, respectively.

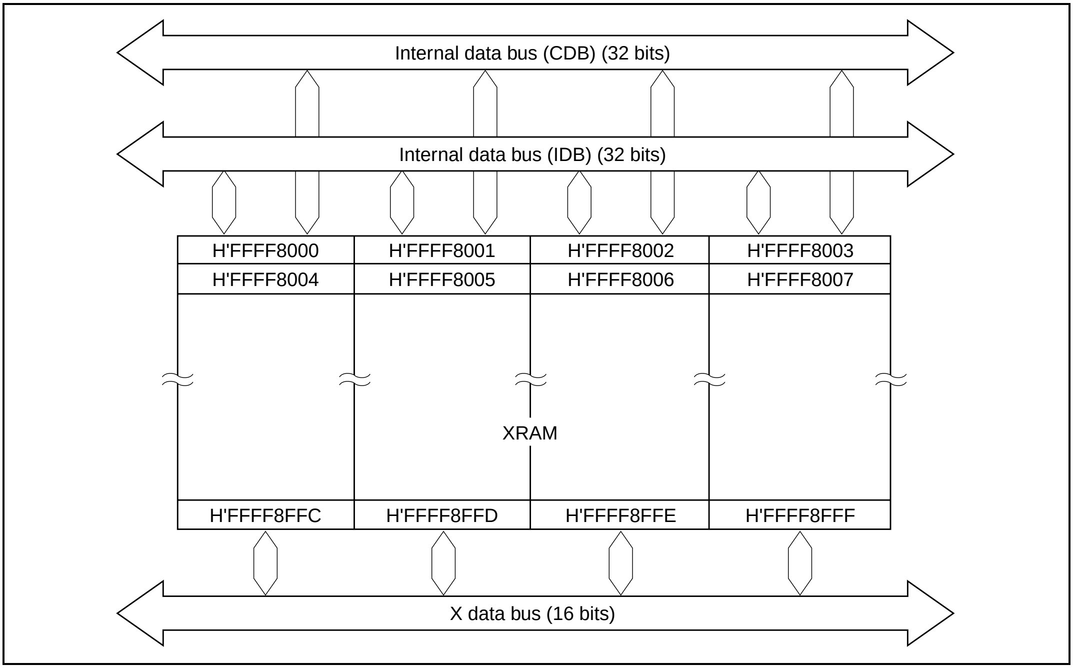


Figure 21.1 Block Diagram of XRAM

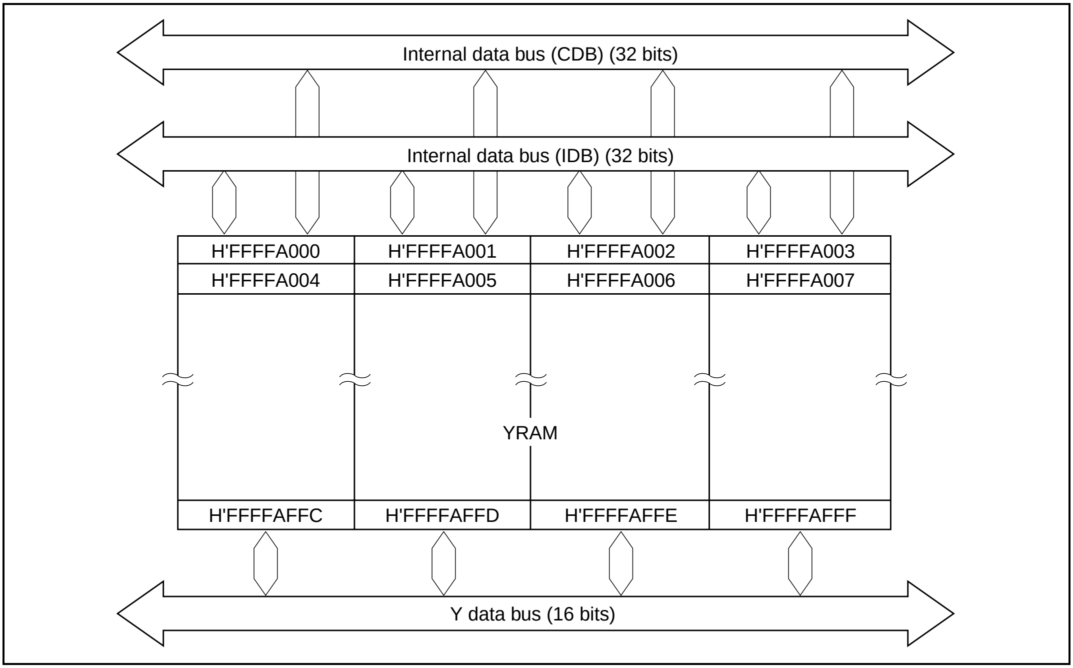


Figure 21.2 Block Diagram of YRAM

21.2 Operation

When addresses H'FFFF8000 to H'FFFF8FFF are accessed, XRAM is accessed, and when addresses H'FFFFA000 to H'FFFFAFFF are accessed, YRAM is accessed.

Section 22 Electrical Characteristics

22.1 Absolute Maximum Ratings

Table 22.1 Absolute Maximum Ratings

Item	Symbol	Value	Unit
Power supply voltage	V_{CC}	–0.3 to +4.3	V
	PV_{CC}	–0.3 to +7.0	V
Input voltage (except A/D ports)	V_{in}	–0.3 to V_{CC} +0.3	V
Input voltage (A/D ports)	V_{in}	–0.3 to AV_{CC} +0.3	V
Analog power supply voltage	AV_{CC}	–0.3 to +4.3	V
Analog input voltage	V_{AN}	–0.3 to +AV_{CC} +0.3	V
Operating temperature	T_{opr}	–20 to +75	°C
Write/erase temperature (F-ZTAT version only)	T_{we}	0 to 50	°C
Storage temperature	T_{stg}	–55 to +125	°C

Usage Note: Permanent damage to the chip may result if the maximum ratings are exceeded.

22.2 Electrical Characteristics

22.2.1 DC Characteristics (1)

Table 22.2 DC Characteristics

Conditions: V_{CC} = $PLLV_{CC}$ = 3.3 V ±0.3 V, PV_{CC} = 5.0 V ±0.5 V/3.3 V ±0.3 V, $PV_{CC} \geq V_{CC}$, AV_{CC} = 3.3 V ±0.3 V, V_{SS} = $PLLV_{SS}$ = PV_{SS} = AV_{SS} = 0 V, Ta = –20°C to +75°C

Item		Symbol	Min	Typ	Max	Unit	Test Conditions
Input high-level voltage*1	$\overline{RES}$, NMI, $\overline{HSTBY}$	V_{IH}	V_{CC} –0.3	—	V_{CC} +0.3	V	
	EXTAL, CKIO		V_{CC} × 0.9	—	V_{CC} +0.3	V	
	A/D ports		2.6	—	AV_{CC} +0.3	V	
	3 V/5 V dual-function pins		2.6	—	PV_{CC} +0.3	V	
	Other input pins		2.6	—	V_{CC} +0.3	V	
Input low-level voltage*1	$\overline{RES}$, NMI, $\overline{HSTBY}$	V_{IL}	–0.3	—	0.5	V	
	Other input pins		–0.3	—	0.8	V	
Schmitt-trigger input characteristics	TIOC0A–TIOC5A, TIOC0B–TIOC5B, TIOC0C, TIOC3C, TIOC0D, TIOC3D, TCLKA, TCLKB, TCLKC, TCLKD, SCK0, SCK1, SCK2, RxD0, RxD1, RxD2, PCI	V_T^+	$V_T^+ \geq$ 4.0 V (min): 3 V/5 V dual-function pins, PV_{CC} = 5 V ±0.5 V $V_T^+ \geq$ 2.6 V (min): Other cases				
		V_T^-	$V_T^- \leq$ 0.8 V (max)				
		$V_T^+ - V_T^-$	0.2	—	—	V	
Input leakage current	$\overline{RES}$, NMI, $\overline{HSTBY}$	| Iin |	—	—	1.0	µA	
	A/D ports		—	—	1.0	µA	
	Other input pins		—	—	1.0	µA	
Three-state leakage current (off state)	A25–A0, D31–D0, $\overline{CS5}$–$\overline{CS0}$, RDWR, $\overline{WRxx}$, $\overline{RD}$, $\overline{WR}$, $\overline{RASx}$, $\overline{CASxxx}$	| I_{TS} |	—	—	1.0	µA	

Item		Symbol	Min	Typ	Max	Unit	Test Conditions
Output high-level voltage	3 V/5 V dual-function pins	V_{OH}	PV_{CC} –0.7	—	—	V	I_{OH} = –200 μA
	Other output pins		V_{CC} –0.7	—	—	V	I_{OH} = –200 μA
Output low-level voltage	3 V/5 V dual-function pins	V_{OL}	—	—	0.6	V	I_{OL} = 1.6 mA
	Other output pins		—	—	0.6	V	I_{OL} = 1.6 mA
	PE17–PE19, PE21–PE23		—	—	1.5	V	I_{OL} = 15 mA
Input capacitance	$\overline{\text{RES}}$	C_{in}	—	—	60	pF	V_{in} = 0 V f = 1 MHz T_a = 25°C
	NMI		—	—	30	pF	
	All other input pins		—	—	25	pF	
Current dissipation (HD64F7065S, HD64F7065A) F-ZTAT version	Normal operation	I_{CC}	—	230	290	mA	
	Sleep mode		—	160	260	mA	
	Standby mode		—	5	100	μA	T_a ≤ 50°C
			—	—	800	μA	50°C < T_a
Current dissipation (HD6437065A) Mask ROM version	Normal operation	I_{CC}	—	170	250	mA	
	Sleep mode		—	150	240	mA	
	Standby mode		—	5	100	μA	T_a ≤ 50°C
			—	—	800	μA	50°C < T_a
Port power supply current	During operation	PI_{cc}	—	5	10	mA	PV_{cc} = 3.6 V
			—	7	15	mA	PV_{cc} = 5.5 v
	Standby mode		—	0.2	0.25	mA	
Analog power supply current		AI_{cc}	—	2.0	5.0	mA	During A/D and D/A conversion*2
			—	0.1	0.15	mA	A/D and D/A halted*2
RAM standby voltage		V_{RAM}	2.0	—	—	V	

Notes: 1. Except Schmitt-trigger inputs

2. AI_{CC} during A/D and D/A conversion is the analog power supply current value when the A/D converter's DAE bit is set to 1.

"Halted" refers to the state in which A/D converter bit ADST is 0 and D/A converter bits DAOE1, DAOE0, and DAE are 0.

Usage Notes:

1. If the A/D converter is not used (including standby mode), do not leave the AV_{CC} and AV_{SS} pins open. Connect the AV_{CC} pin to V_{CC}, and the AV_{SS} pin to V_{SS}.
2. Current dissipation values are for V_{IH} min = V_{CC} –0.5 V and V_{IL} max = 0.5 V with all output pins unloaded. (For standby current specification conditions, values are for V_{IH} = V_{CC}/PV_{CC} and V_{IL} = 0 V with all output pins unloaded.)
3. The F-ZTAT and mask ROM versions have the same functions, and the electrical characteristics of both are within specification, but characteristic-related performance values, operating margins, noise margins, noise emission, etc., are different. Caution is therefore required in carrying out system design, and when switching between F-ZTAT and mask ROM versions.

22.2.2 DC Characteristics (2)

Table 22.3 Permissible Output Current Values

Conditions: V_{CC} = $PLLV_{CC}$ = 3.3 V ±0.3 V, PV_{CC} = 5.0 V ±0.5 V/3.3 V ±0.3 V, $PV_{CC} \geq V_{CC}$,
AV_{CC} = 3.3 V ±0.3 V, V_{SS} = $PLLV_{SS}$ = PV_{SS} = AV_{SS} = 0 V, Ta = –20°C to +75°C

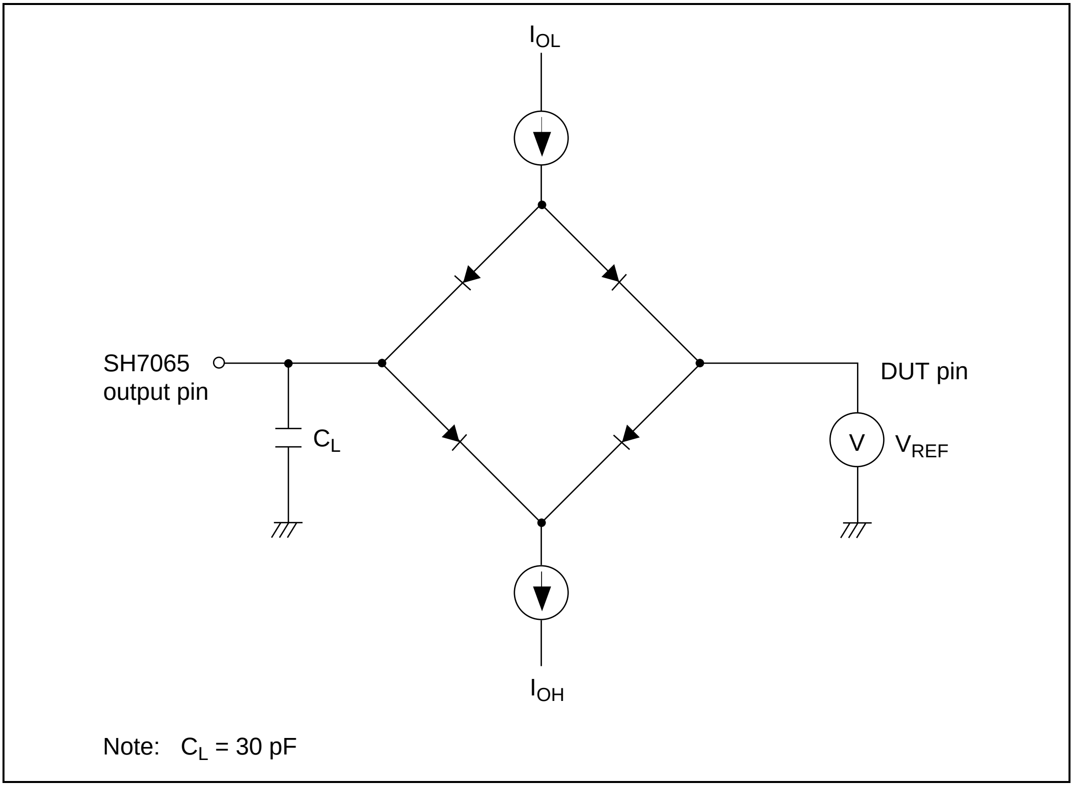
Item	Symbol	Min	Typ	Max	Unit
Output low-level permissible current (per pin)	I_{OL}	—	—	2.0*	mA
Output low-level permissible current (total)	ΣI_{OL}	—	—	80	mA
Output high-level permissible current (per pin)	$-I_{OH}$	—	—	2.0	mA
Output high-level permissible current (total)	$\Sigma(-I_{OH})$	—	—	25	mA

Note: * For PE17 to PE19 and PE21 to PE23, I_{OL} = 15 mA (max). However, an I_{OL} current exceeding 2.0 mA should not be applied to more than three of these pins simultaneously.

Usage Note: To protect chip reliability, do not exceed the output current values in table 22.3.

22.3 AC Characteristics Test Conditions

Input reference levels: High level: V_{CC} –0.3 V, low level: 0.5 V
Output reference levels: High level: 2.0 V, low level: 0.8 V



Note: C_L = 30 pF

Figure 22.1 Output Load Circuit

22.3.1 Clock Timing

Table 22.4 Clock Timing

Conditions: V_{CC} = PLLV_{CC} = 3.3 V ±0.3 V, PV_{CC} = 5.0 V ±0.5 V/3.3 V ±0.3 V, PV_{CC} ≥ V_{CC}, AV_{CC} = 3.3 V ±0.3 V, V_{SS} = PLLV_{SS} = PV_{SS} = AV_{SS} = 0 V, Ta = –20°C to +75°C

Item	Symbol	Min	Max	Unit	Figure
Operating frequency (master clock)	f_{OP}	20	60	MHz	Figure 22.2
Clock cycle time	t_{cyc}	16.7	50	ns	
Clock low-level pulse width	t_{CL}	4.4	—	ns	
Clock high-level pulse width	t_{CH}	4.4	—	ns	
Clock rise time	t_{CR}	—	4	ns	
Clock fall time	t_{CF}	—	4	ns	
EXTAL/CKIO clock input frequency	f_{EX}	5	30	MHz	Figure 22.3
EXTAL/CKIO clock input cycle time	t_{EXcyc}	33.3	200	ns	
EXTAL/CKIO clock input low-level pulse width	t_{EXL}	11.6	—	ns	
EXTAL/CKIO clock input high-level pulse width	t_{EXH}	11.6	—	ns	
EXTAL/CKIO clock input rise time	t_{EXR}	—	5	ns	
EXTAL/CKIO clock input fall time	t_{EXF}	—	5	ns	
Reset oscillation settling time	t_{OSC1}	10	—	ms	Figure 22.4
Standby recovery oscillation settling time	t_{OSC2}	10	—	ms	

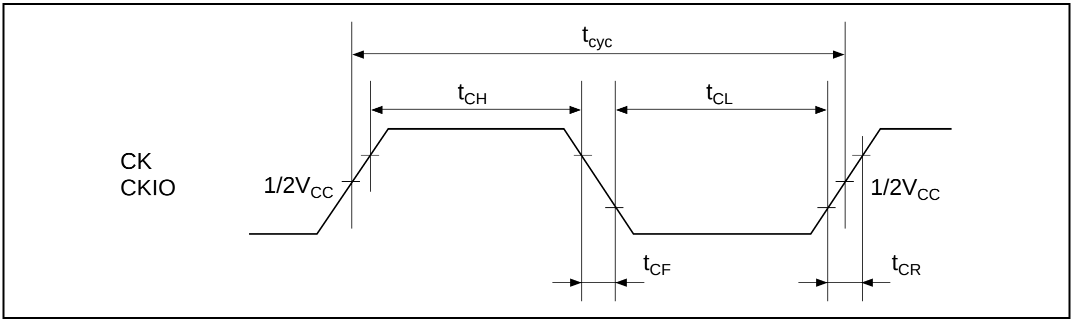


Figure 22.2 System Clock Timing

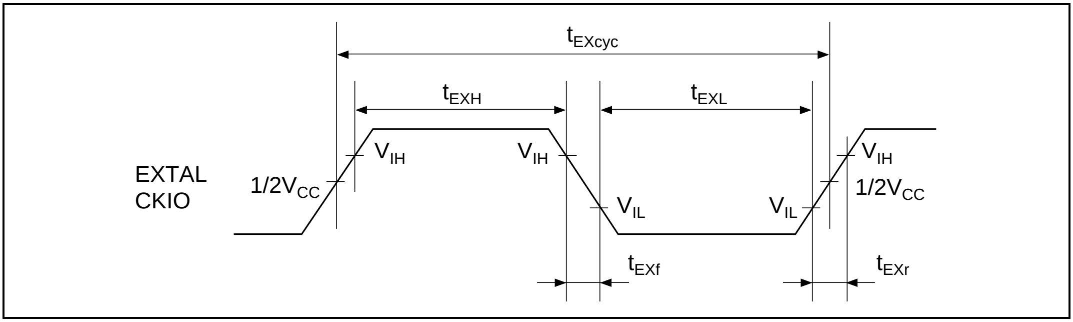


Figure 22.3 EXTAL Clock Input Timing

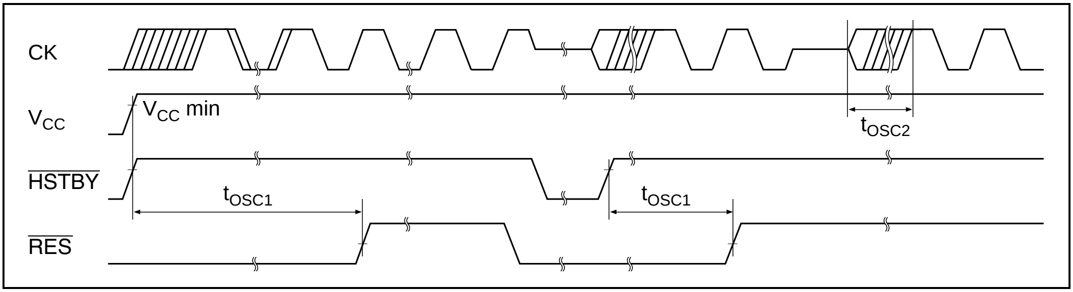


Figure 22.4 Oscillation Settling Time

22.3.2 Control Signal Timing

Table 22.5 Control Signal Timing

Conditions: V_{CC} = $PLLV_{CC}$ = 3.3 V ±0.3 V, PV_{CC} = 5.0 V ±0.5 V/3.3 V ±0.3 V, $PV_{CC} \geq V_{CC}$, AV_{CC} = 3.3 V ±0.3 V, V_{SS} = $PLLV_{SS}$ = PV_{SS} = AV_{SS} = 0 V, Ta = –20°C to +75°C

Item	Symbol	Min	Max	Unit	Figure
$\overline{\text{RES}}$ rise and fall	t_{RESr}, t_{RESf}	—	200	ns	Figure 22.5
$\overline{\text{RES}}$ pulse width	t_{RESW}	40	—	t_{cyc}*1	
NMI rise and fall	t_{NMIr}, t_{NMIf}	—	200	ns	Figure 22.6
NMI pulse width	t_{NMIW}	2.5	—	t_{cyc}*2	
$\overline{\text{IRQ}}$ pulse width	t_{IRQW}	2.5	—	t_{cyc}*2	
$\overline{\text{IRQOUT}}$ output delay time	t_{IRQOD}	—	35	ns	Figure 22.7
Bus request setup time	t_{BRQS}	35	—	ns	Figure 22.8
Bus acknowledge delay time 1	t_{BACKD1}	—	35	ns	
Bus acknowledge delay time 2	t_{BACKD2}	—	35	ns	
Bus three-state delay time	t_{BZD}	—	35	ns	

Notes: 1. Slowest module clock
2. Slower of Mϕ and CKE clocks

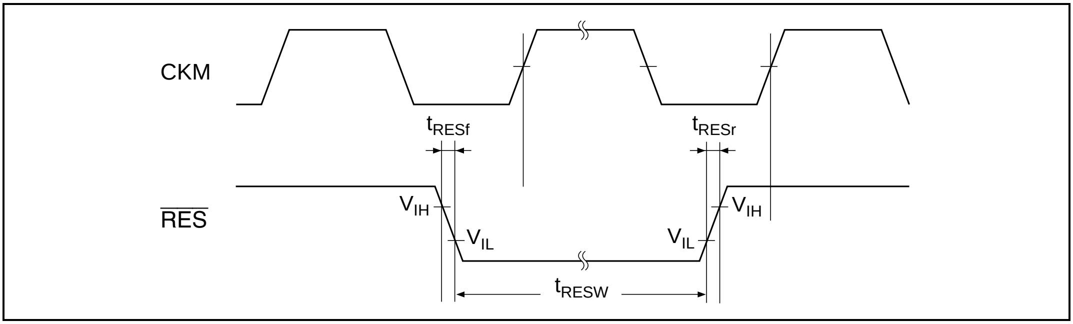


Figure 22.5 Reset Input Timing

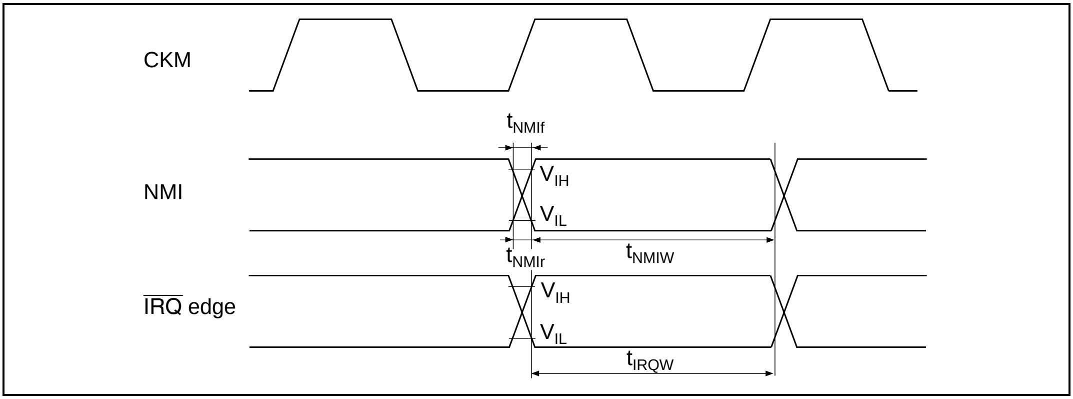


Figure 22.6 Interrupt Signal Input Timing

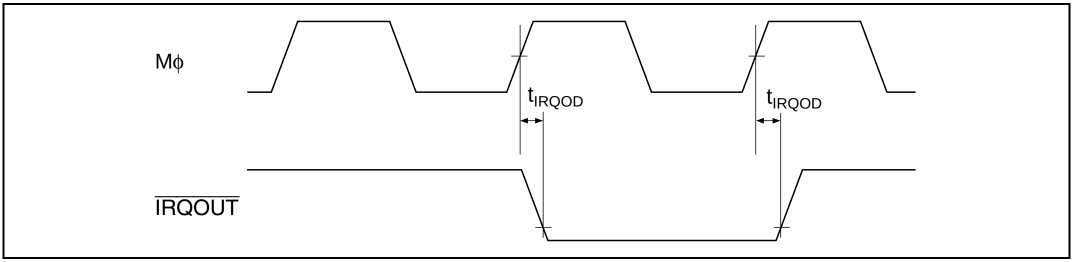
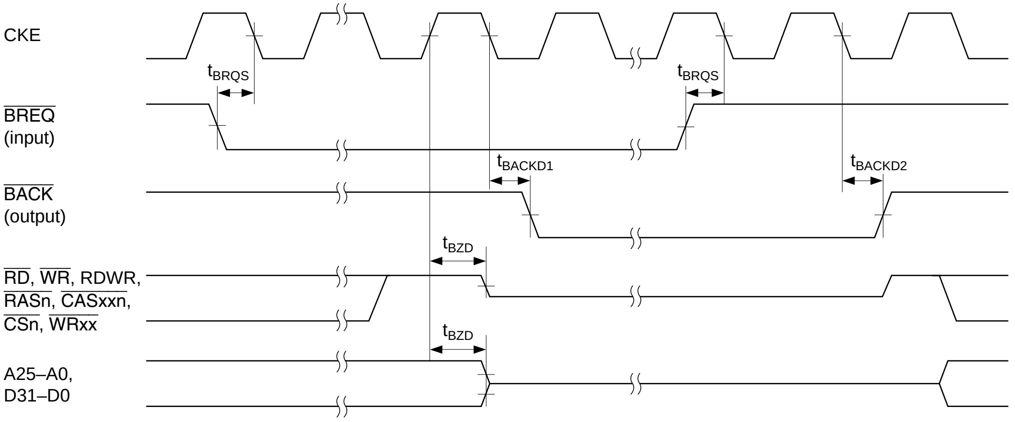


Figure 22.7 Interrupt Signal Output Timing


CKE
t_{BRQS}
t_{BRQS}
$\overline{BREQ}$ (input)
t_{BACKD1}
t_{BACKD2}
$\overline{BACK}$ (output)
t_{BZD}
$\overline{RD}$, $\overline{WR}$, RDWR, $\overline{RASn}$, $\overline{CASxxn}$, $\overline{CSn}$, $\overline{WRxx}$
t_{BZD}
A25–A0, D31–D0

Notes: 1. $\overline{\text{RASn}}$, $\overline{\text{CASxxn}}$, and RDWR are output when the bus is released during self-refreshing.
2. When a TAS instruction is executed during the bus release, BACK is temporarily negated, then asserted again after execution is completed.

Figure 22.8 Bus Release Timing

22.3.3 Bus Timing

Table 22.6 Bus Timing

Conditions: V_{CC} = $PLLV_{CC}$ = 3.0 V to 3.6 V, AV_{CC} = 3.0 V to 3.6 V, AV_{CC} = V_{CC} ±10%,
V_{SS} = $PLLV_{SS}$ = PV_{SS} = AV_{SS} = 0 V, Ta = –20°C to +75°C

Item	Symbol	Min	Max	Unit	Figure
Address delay time	t_{AD}	2^{*3}	25	ns	Figure 22.9, Figure 22.10, Figure 22.12, Figure 22.13, Figure 22.14, Figure 22.15, Figure 22.16, Figure 22.19, Figure 22.20
$\overline{\text{CS}}$ delay time 1	t_{CSD1}	2^{*3}	25	ns	
$\overline{\text{CS}}$ delay time 2	t_{CSD2}	2^{*3}	25	ns	
Read strobe delay time 1	t_{RSD1}	2^{*3}	25	ns	Figure 22.9, Figure 22.10, Figure 22.19, Figure 22.20
Read strobe delay time 2	t_{RSD2}	2^{*3}	25	ns	
Read data setup time	t_{RDS}^{*4}	25	—	ns	Figure 22.9, Figure 22.10, Figure 22.12, Figure 22.13, Figure 22.14, Figure 22.15, Figure 22.16, Figure 22.19, Figure 22.20
Read data hold time	t_{RDH}	0	—	ns	
Write strobe delay time 1	t_{WSD1}	2^{*3}	25	ns	Figure 22.9, Figure 22.10, Figure 22.19, Figure 22.20
Write strobe delay time 2	t_{WSD2}	2^{*3}	25	ns	
Write data delay time	t_{WDD}	—	25	ns	Figure 22.9, Figure 22.10, Figure 22.12, Figure 22.13, Figure 22.14, Figure 22.15, Figure 22.16, Figure 22.19, Figure 22.20
Write data hold time	t_{WDH}	0	25^{*2}	ns	
$\overline{\text{WAIT}}$ setup time	t_{WTS}	25	—	ns	Figure 22.11, Figure 22.13, Figure 22.20
$\overline{\text{WAIT}}$ hold time	t_{WTH}	0	—	ns	
$\overline{\text{RAS}}$ delay time 1	t_{RASD1}	2^{*3}	25	ns	Figure 22.12, Figure 22.13, Figure 22.14, Figure 22.15, Figure 22.16, Figure 22.17, Figure 22.18
$\overline{\text{RAS}}$ delay time 2	t_{RASD2}	2^{*3}	25	ns	

Item	Symbol	Min	Max	Unit	Figure
$\overline{\text{CAS}}$ delay time 1	t_{CASD1}	2^{*3}	25	ns	Figure 22.12, Figure 22.13, Figure 22.14, Figure 22.15, Figure 22.16, Figure 22.17, Figure 22.18
$\overline{\text{CAS}}$ delay time 2	t_{CASD2}	2^{*3}	25	ns	
Read data access time	t_{ACC}^{*1}	$t_{cyc} \times (n+1.5) - 33$	—	ns	Figure 22.9, Figure 22.10
Access time from read strobe	t_{OE}^{*1}	$t_{cyc} \times (n+1) - 33$	—	ns	
Access time from column address	t_{AA}^{*1}	$t_{cyc} \times (n+1.5) - 33$	—	ns	Figure 22.12, Figure 22.14, Figure 22.15
Access time from $\overline{\text{RAS}}$	t_{RAC}^{*1}	$t_{cyc} \times (n+RCD+2) - 33$	—	ns	
Access time from $\overline{\text{CAS}}$	t_{CAC}^{*1}	$t_{cyc} \times (n+1) - 33$	—	ns	
Row address hold time	t_{RAH}	$t_{cyc} \times (RCD+0.5) - 25$	—	ns	Figure 22.12, Figure 22.15
Row address setup time	t_{ASR}	$t_{cyc} \times 0.5 - 16.6$	—	ns	
Data input setup time	t_{DS}	$t_{cyc} \times (m+0.5) - 25$	—	ns	Figure 22.12, Figure 22.14, Figure 22.15
Data input hold time	t_{DH}	t_{cyc}	—	ns	

Notes: n is the number of waits. m is 0 when the number of DRAM write cycle waits is 0, and 1 otherwise. RCD is the set value of the RCD bit in DCR1.

1. The t_{RDS} specification need not be met as long as the access time specification is met.
2. t_{WDH} (max) is a reference value.
3. The minimum (Min) values for delay times are reference values.
4. t_{RDS} is a reference value.

Item	Symbol	Min	Max	Unit	Figure
Read/write strobe delay time 1	t_{RWD1}	2*	25	ns	Figure 22.12, Figure 22.13, Figure 22.14, Figure 22.15
Read/write strobe delay time 2	t_{RWD2}	2*	25	ns	
Fast page mode $\overline{\text{CAS}}$ precharge time	t_{CP}	t_{cyc} – 25	—	ns	Figure 22.15
$\overline{\text{RAS}}$ precharge time	t_{RP}	$t_{cyc} \times$ (TPC +1.0) – 25	—	ns	Figure 22.12, Figure 22.15
$\overline{\text{CAS}}$ setup time	t_{CSR}	10	—	ns	Figure 22.17, Figure 22.18
$\overline{\text{AH}}$ delay time 1	t_{AHD1}	2*	25	ns	Figure 22.19, Figure 22.20
$\overline{\text{AH}}$ delay time 2	t_{AHD2}	2*	25	ns	
Multiplex address delay time	t_{MAD}	2*	25	ns	
Multiplex address hold time	t_{MAH}	0	—	ns	
$\overline{\text{DACK}}$ delay time 1	t_{DACKD1}	2*	25	ns	Figure 22.9, Figure 22.10, Figure 22.12, Figure 22.13, Figure 22.14, Figure 22.15, Figure 22.16, Figure 22.19, Figure 22.20
An setup time with respect to $\overline{\text{WRxx}}$ fall	t_{AS}	0	—	ns	Figure 22.9, Figure 22.10
An hold time with respect to $\overline{\text{WRxx}}$ rise	t_{WR}	5	—	ns	
Dn hold time with respect to $\overline{\text{WRxx}}$ rise	t_{WRH}	0	—	ns	

Item	Symbol	Min	Max	Unit	Figure
Bus start delay time 1	t_{BSD1}	2*	25	ns	Figure 22.9, Figure 22.10, Figure 22.12, Figure 22.13, Figure 22.14, Figure 22.16, Figure 22.19, Figure 22.20
Bus start delay time 2	t_{BSD2}	2*	25	ns	
Byte strobe delay time 1	t_{XXBSD1}	2*	25	ns	Figure 22.9, Figure 22.10
Byte strobe delay time 2	t_{XXBSD2}	2*	25	ns	
Output enable delay time 1	t_{OED1}	2*	25	ns	Figure 22.14, Figure 22.16
Output enable delay time 2	t_{OED2}	2*	25	ns	
Column address hold time (read)	t_{CAH}	$t_{cyc} \times (w + 1.5) - 15$	—	ns	Figure 22.12, Figure 22.13, Figure 22.14, Figure 22.15
Column address hold time (write)	t_{CAH}	$t_{cyc} \times (TCAS + 0.5) - 15$	—	ns	
Column address setup time (read)	t_{ASC}	$t_{cyc} \times 0.5 - 17.5$	—	ns	
Column address setup time (write)	t_{ASC}	$t_{cyc} \times (w + 0.5) - 17.5$	—	ns	

Notes: TPC is the set value of the TPC bit in DCR1.

TCAS is the set value of the TCAS bit in DCR2.

W is the DWW set number in a write from the CPU, the DDWW set number in a DMAC single address write, or the number of waits via the WAIT pin.

t_{cyc} is the CKE cycle (min. 33.3 ns).

* The minimum (Min) values for delay times are reference values.

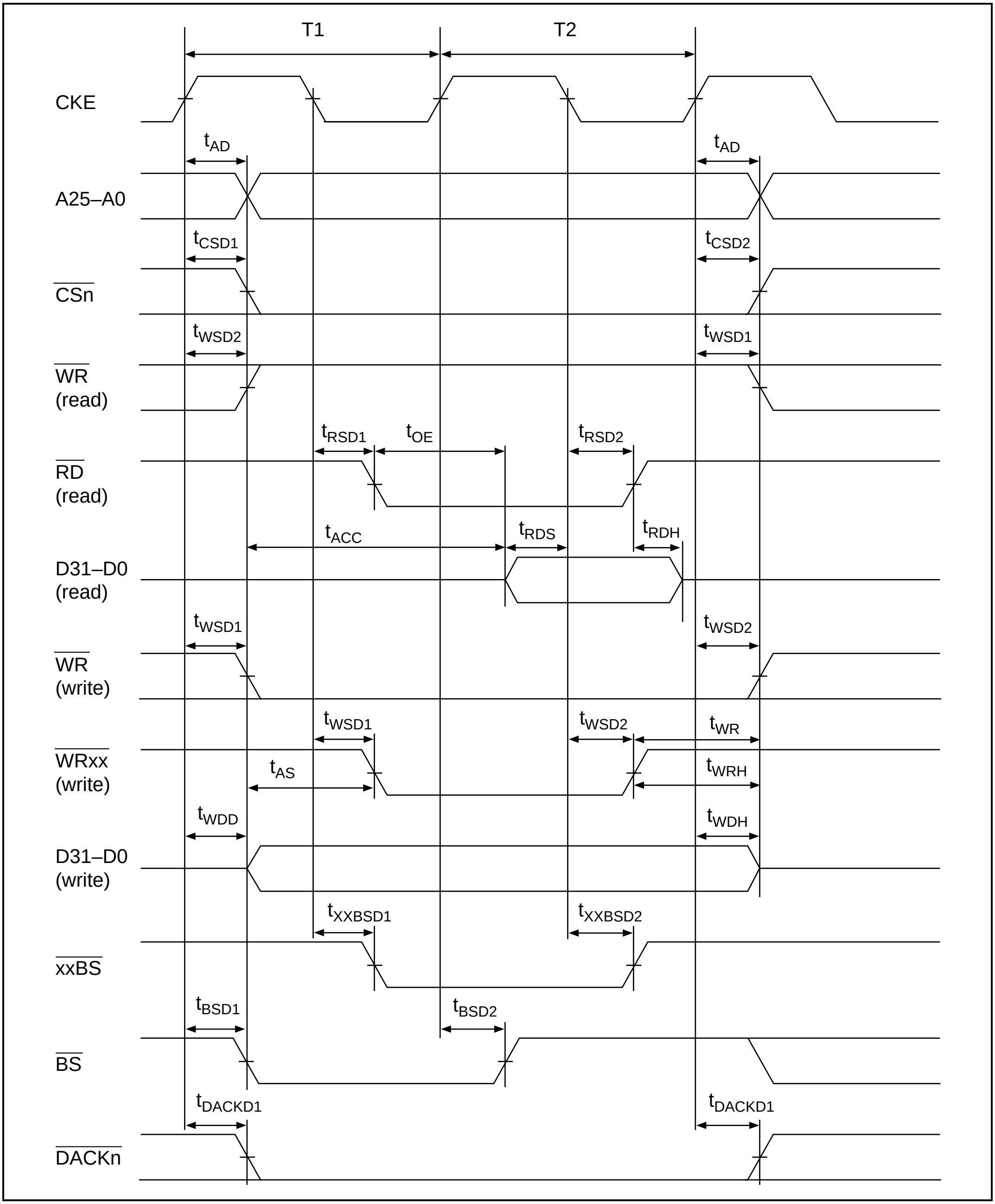


Figure 22.9 Basic Cycle (No Wait)

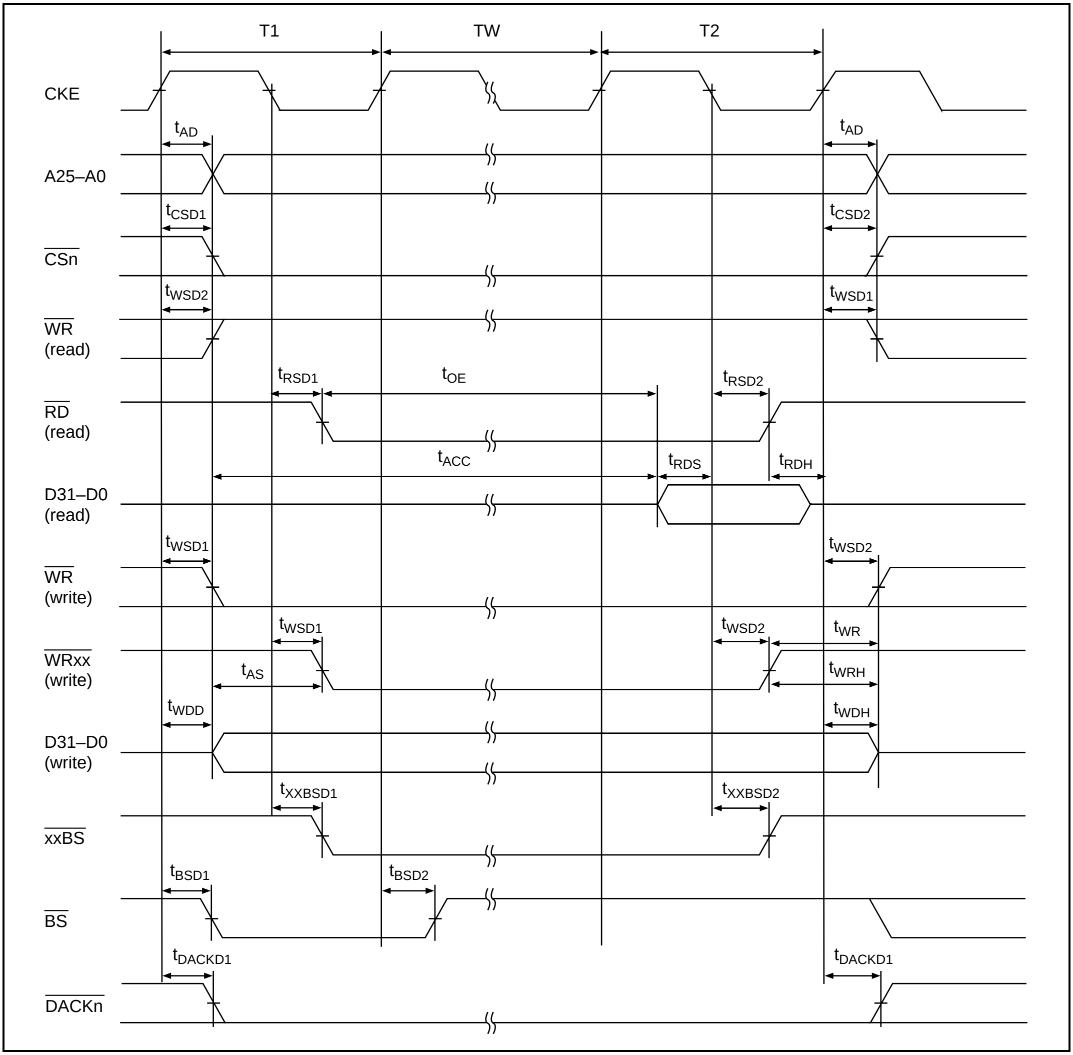


Figure 22.10 Basic Cycle (Software Wait)

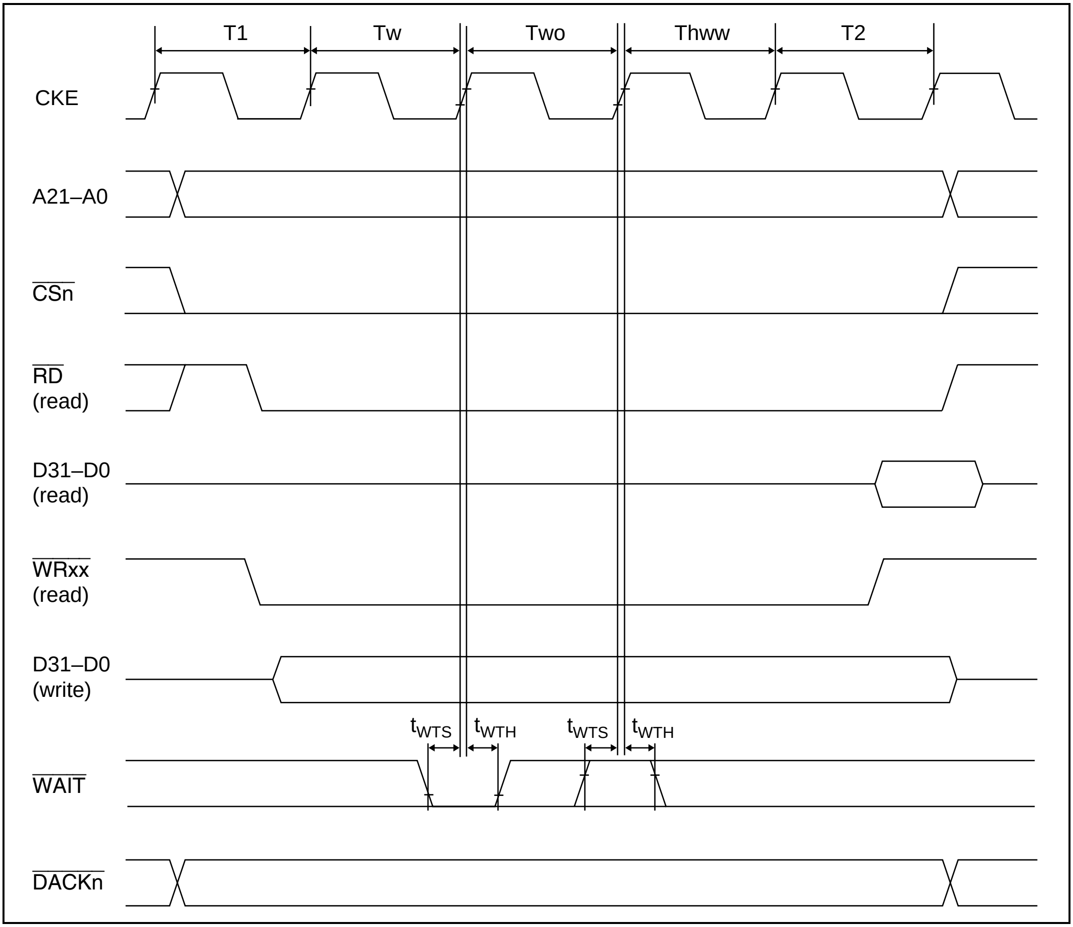
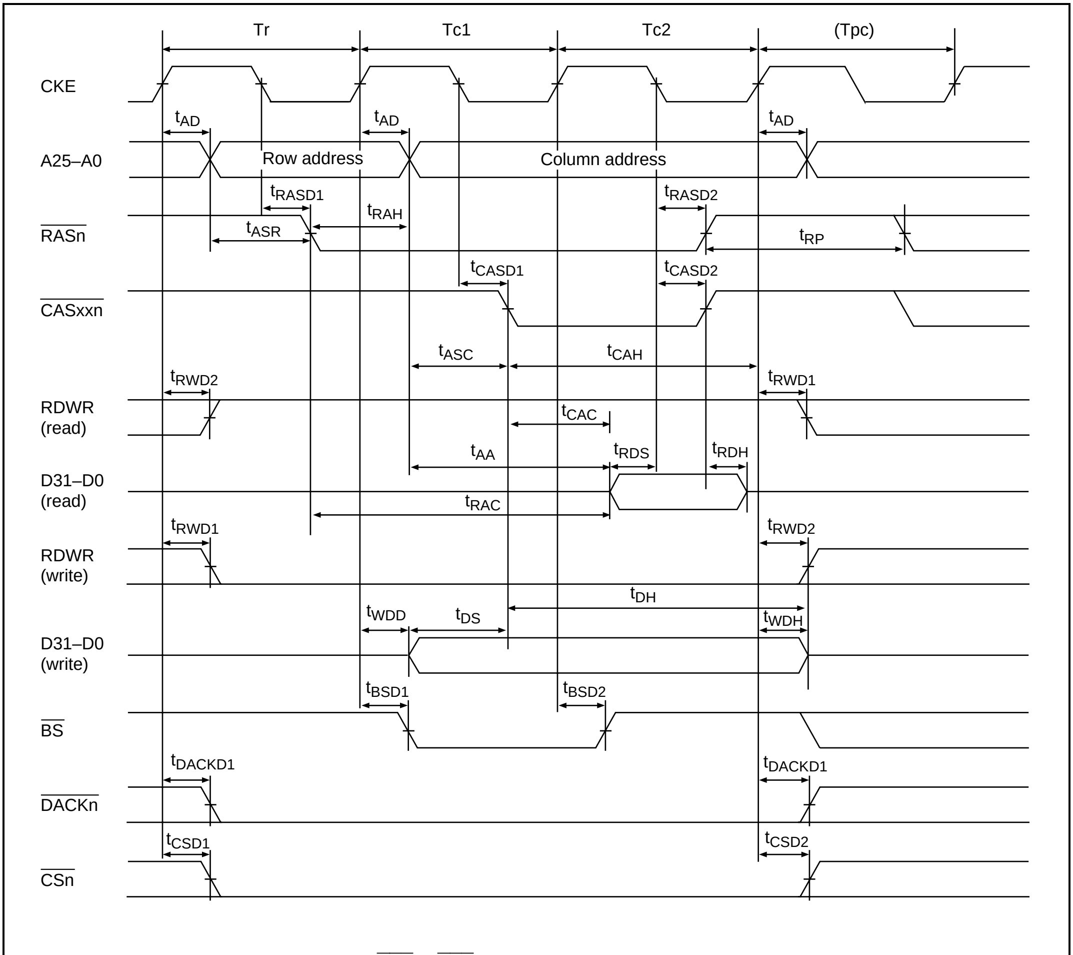
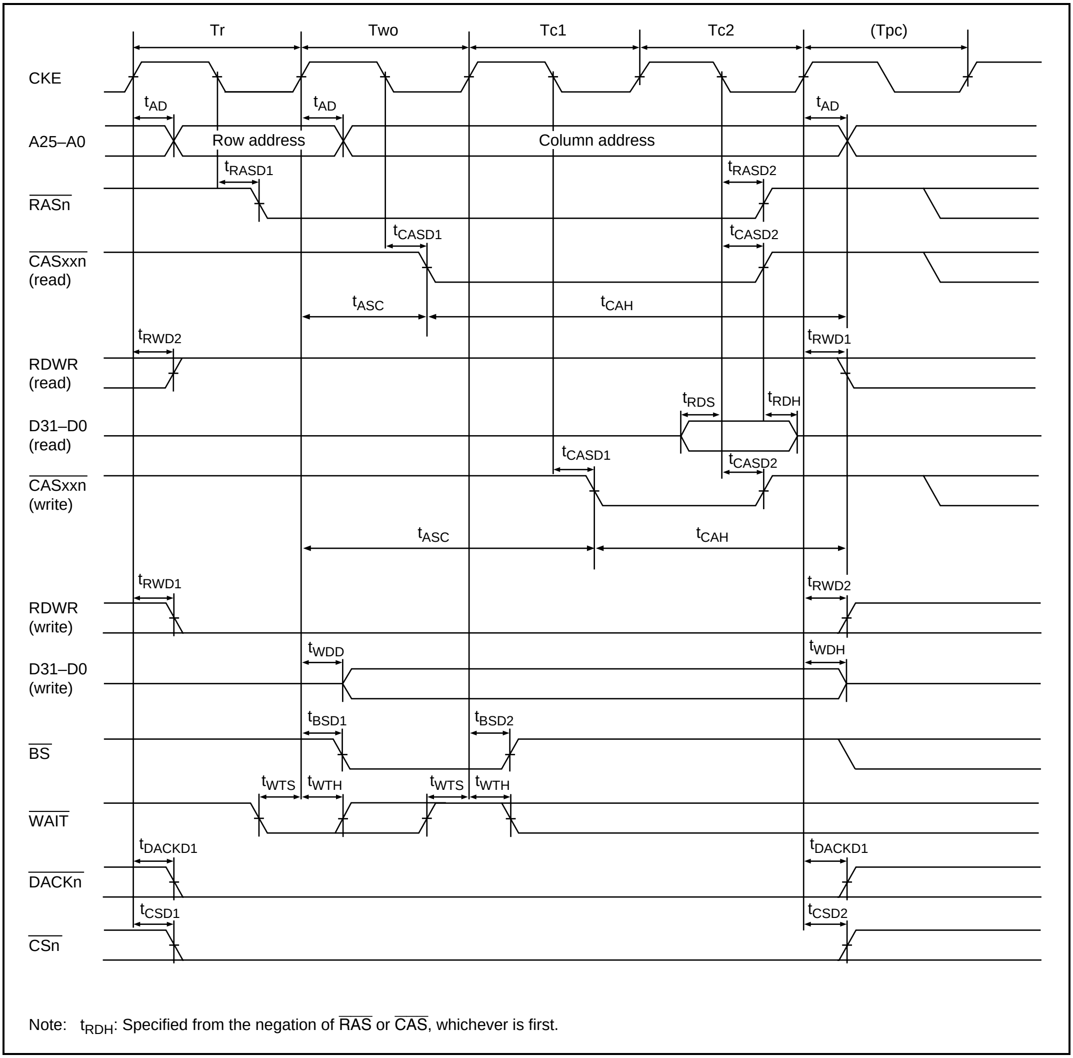


Figure 22.11 Basic Cycle (One Software Wait + $\overline{\text{WAIT}}$ Signal Wait, One Software Wait after $\overline{\text{WAIT}}$ Signal Negation)



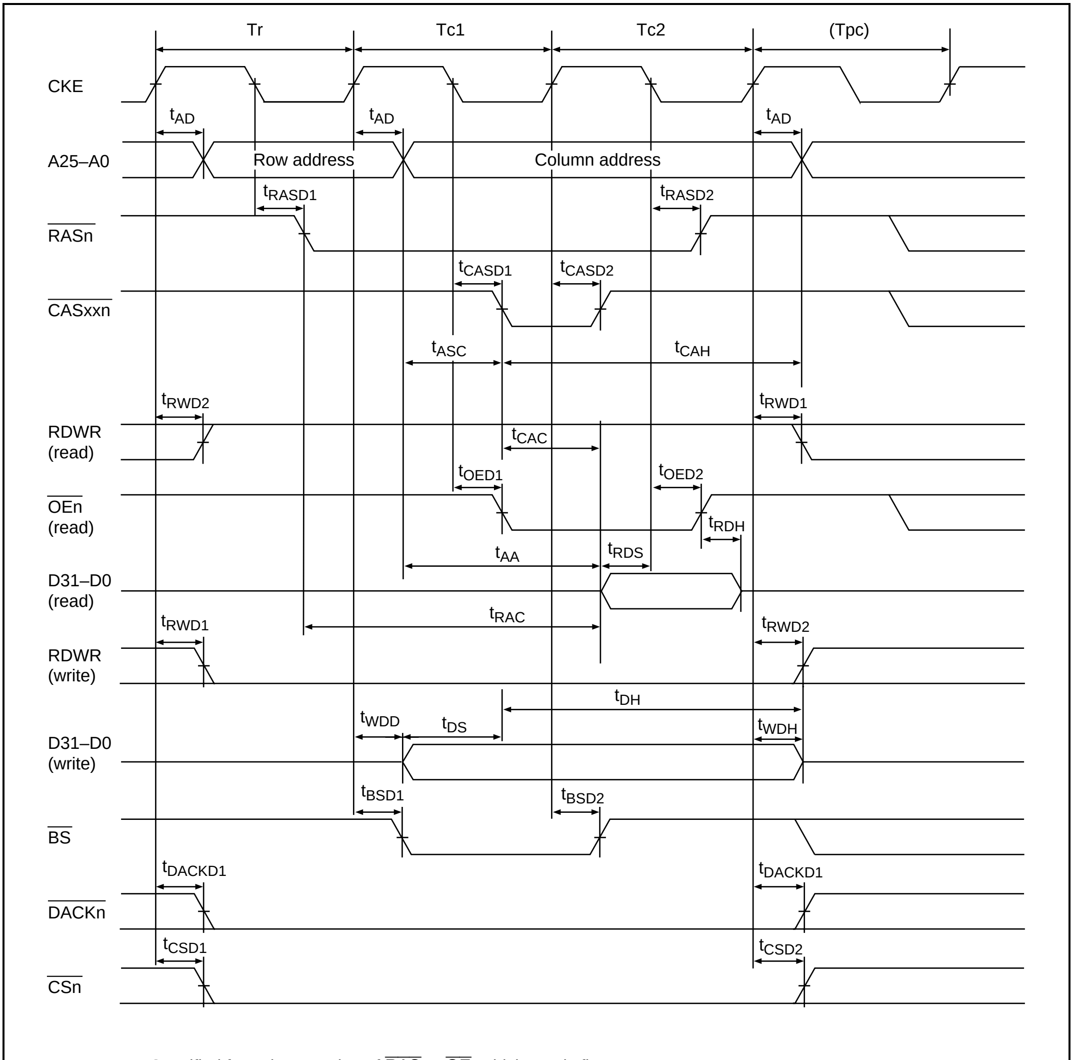
Note: t_{RDH}: Specified from the negation of $\overline{\text{RAS}}$ or $\overline{\text{CAS}}$, whichever is first.

Figure 22.12 DRAM Cycle (Normal Mode, No Wait, TPC = 0, RCD = 0)



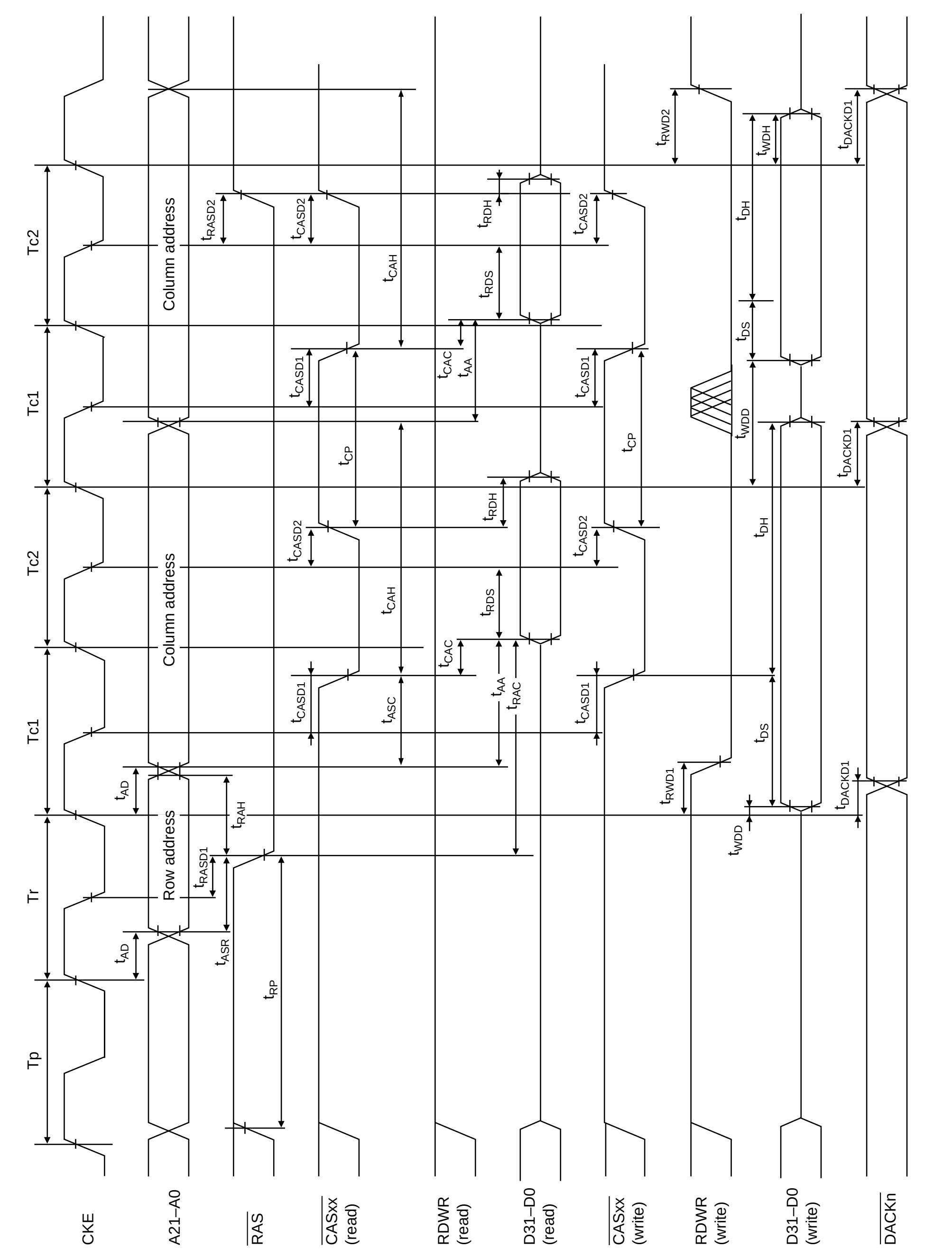
Note: t_{RDH}: Specified from the negation of $\overline{RAS}$ or $\overline{CAS}$, whichever is first.

Figure 22.13 DRAM Cycle (Normal Mode, $\overline{\text{WAIT}}$ Signal Wait, TPC = 0, RCD = 0)



Note: t_{RDH}: Specified from the negation of $\overline{\text{RAS}}$ or $\overline{\text{OE}}$, whichever is first.

Figure 22.14 DRAM Cycle (EDO Mode)



Note: tRDH: Specified from the negation of RAS or CAS, whichever is first.

Figure 22.15 DRAM Cycle (Fast Page Mode)

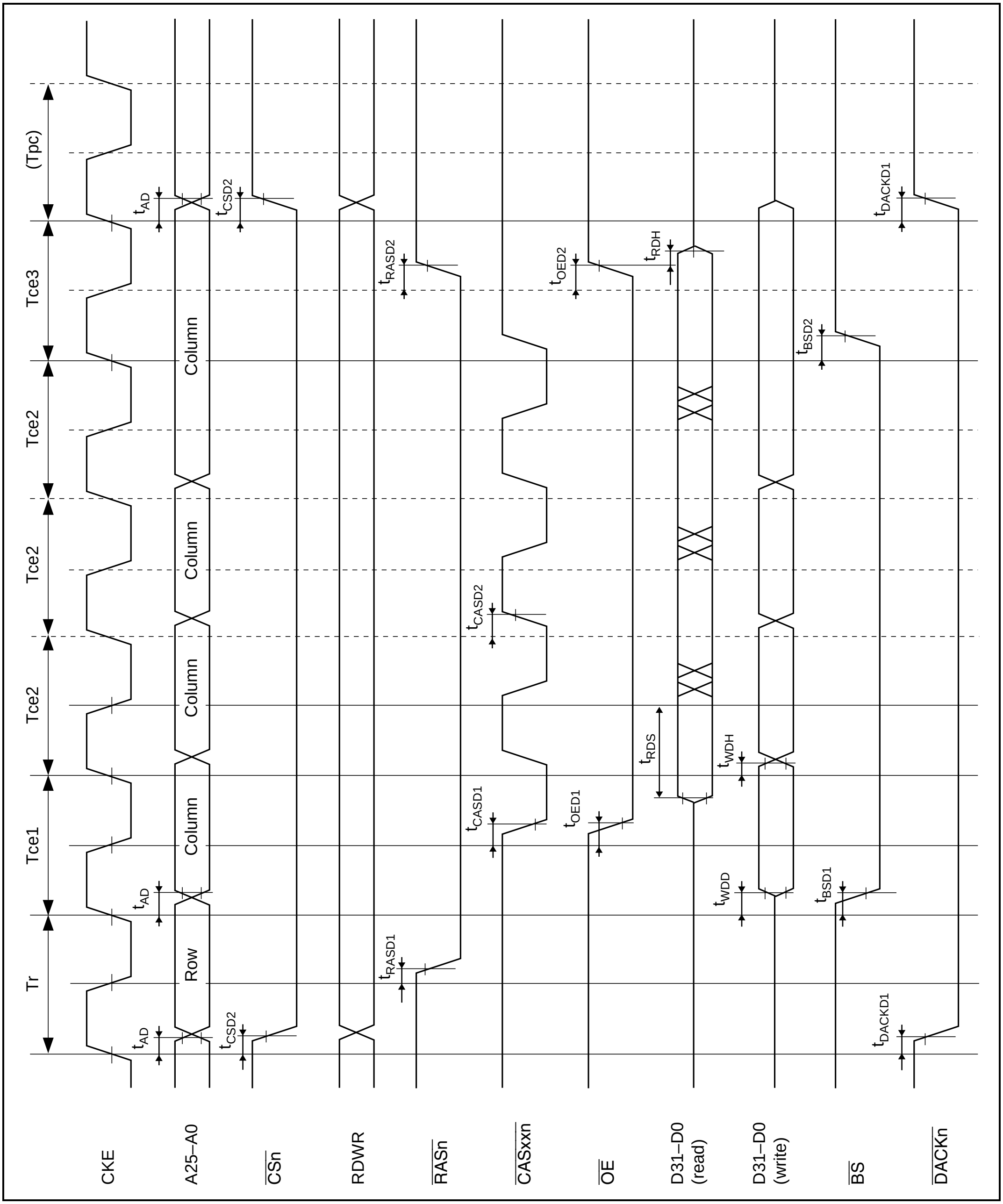


Figure 22.16 DRAM Cycle (EDO Mode, Burst Access)

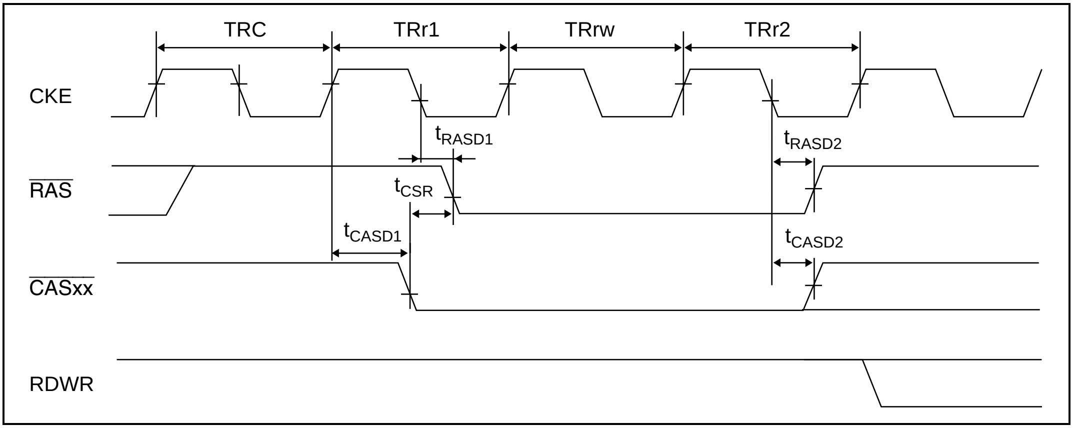


Figure 22.17 CAS-before-RAS Refresh (TRAS1, TRAS0 = 0, 0)

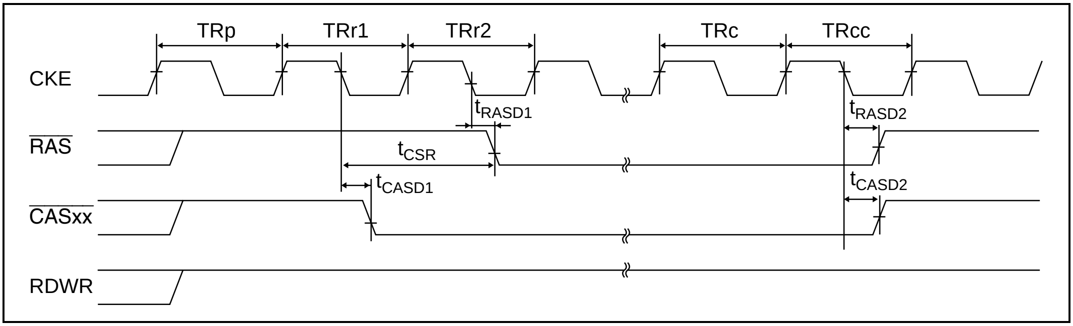


Figure 22.18 Self-Refresh

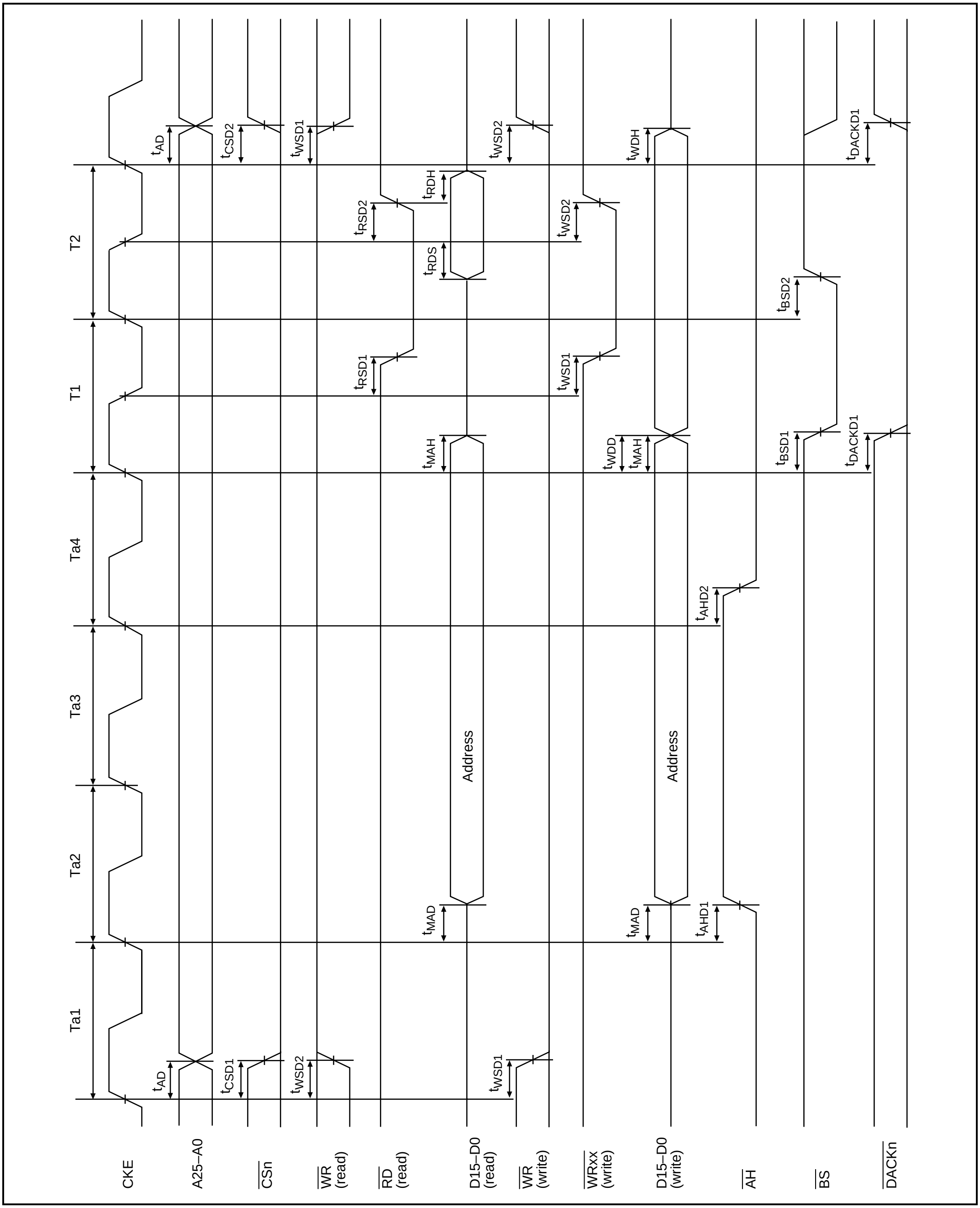


Figure 22.19 Multiplexed Address/Data I/O Space Cycle (One Software Wait + One External Wait)

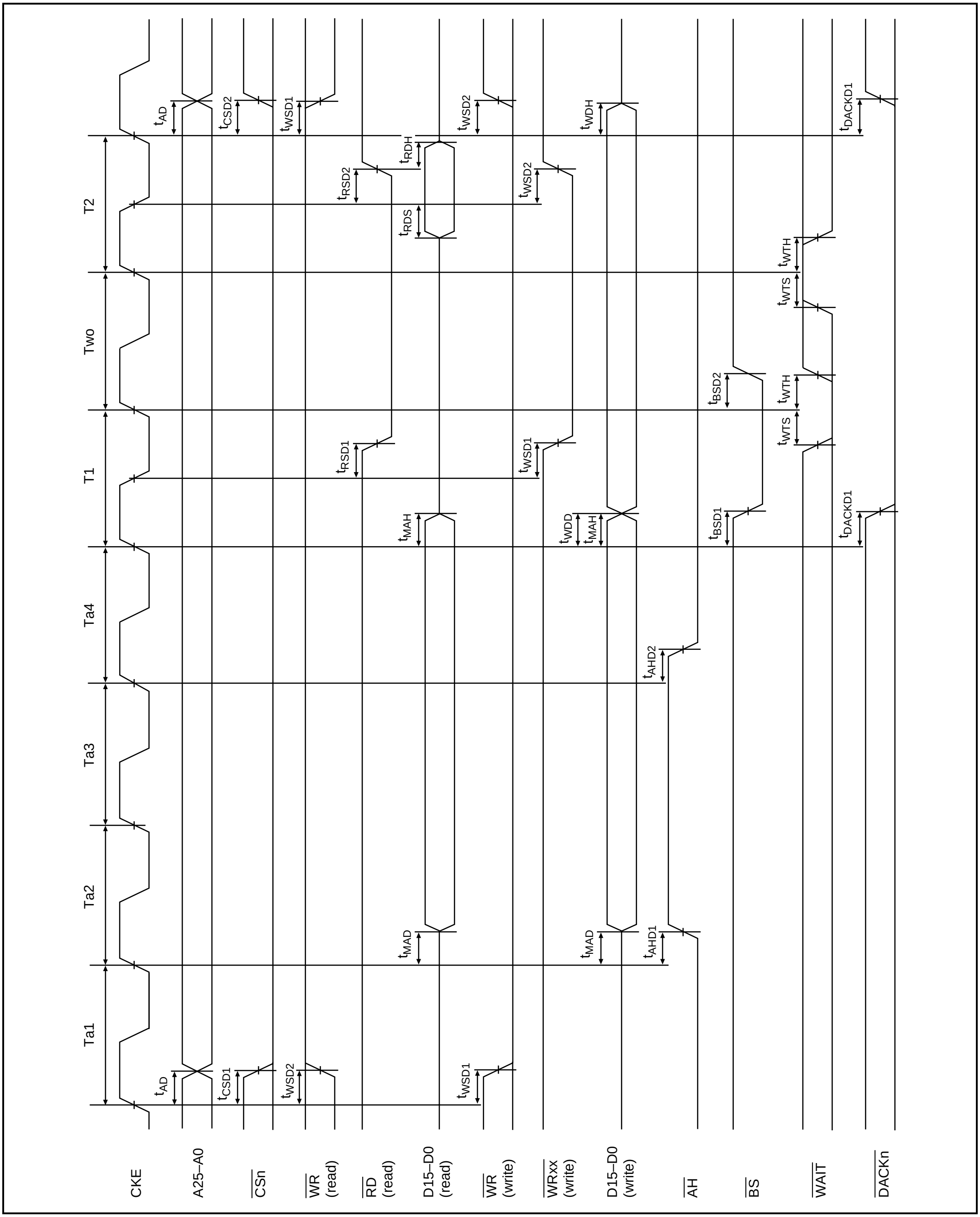


Figure 22.20 Multiplexed Address/Data I/O Space Cycle (WAIT Signal Wait)

22.3.4 Direct Memory Access Controller Timing

Table 22.7 Direct Memory Access Controller Timing

Conditions: V_{CC} = PLLV_{CC} = 3.0 to 3.6 V, AV_{CC} = 3.0 to 3.6 V, AV_{CC} = V_{CC} ±10%,
V_{SS} = PLLV_{SS} = PV_{SS} = AV_{SS} = 0 V, Ta = –20 to +75°C

Item	Symbol	Min	Max	Unit	Figure
$\overline{\text{DREQ0}}$, $\overline{\text{DREQ1}}$ setup time	t_{DRQS}	16	—	ns	Figure 22.21, Figure 22.23
$\overline{\text{DREQ0}}$, $\overline{\text{DREQ1}}$ hold time	t_{DRQH}	16	—	ns	Figure 22.21
$\overline{\text{DREQ0}}$, $\overline{\text{DREQ1}}$ pulse time	t_{DRQW}	2.5	—	tcyc	Figure 22.22
$\overline{\text{DRAK}}$ output delay time	t_{DRAKD}	—	16	ns	Figure 22.23, Figure 22.24
$\overline{\text{TEND}}$ output delay time	t_{TED}	—	16	ns	Figure 22.25

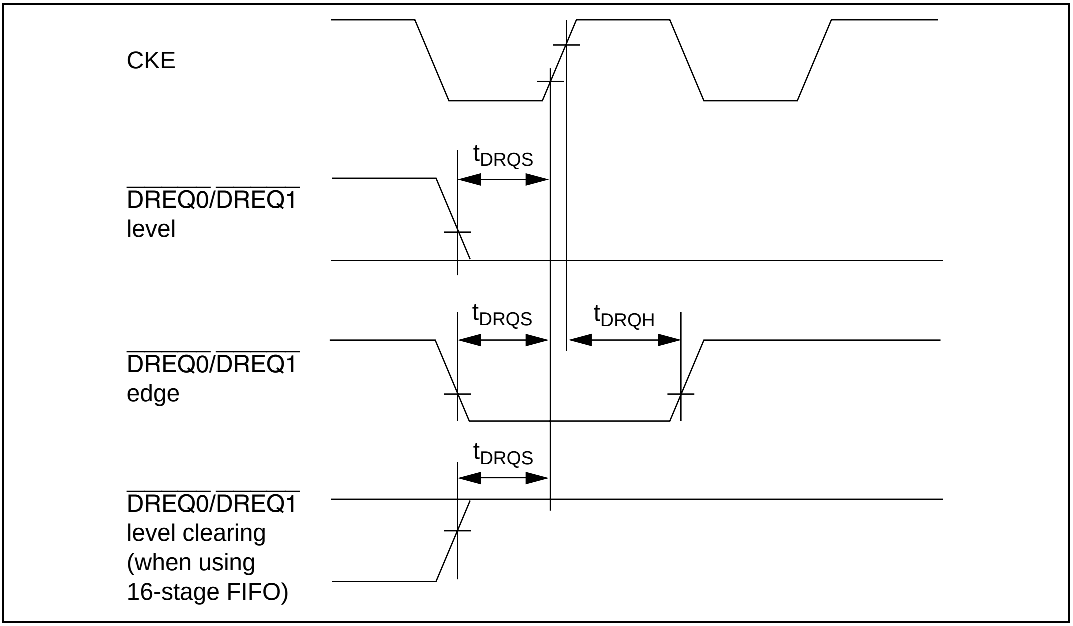

CKE
t_{DRQS}
$\overline{\text{DREQ0}}$/$\overline{\text{DREQ1}}$
level
t_{DRQS}
t_{DRQH}
$\overline{\text{DREQ0}}$/$\overline{\text{DREQ1}}$
edge
t_{DRQS}
$\overline{\text{DREQ0}}$/$\overline{\text{DREQ1}}$
level clearing
(when using
16-stage FIFO)

Figure 22.21 $\overline{\text{DREQ0}}$ and $\overline{\text{DREQ1}}$ Input Timing (1)

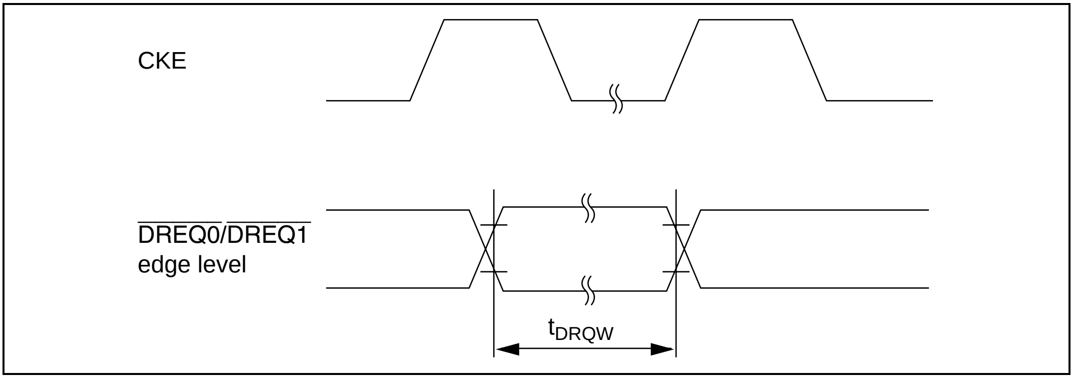


Figure 22.22 $\overline{\text{DREQ0}}$ and $\overline{\text{DREQ1}}$ Input Timing (2)

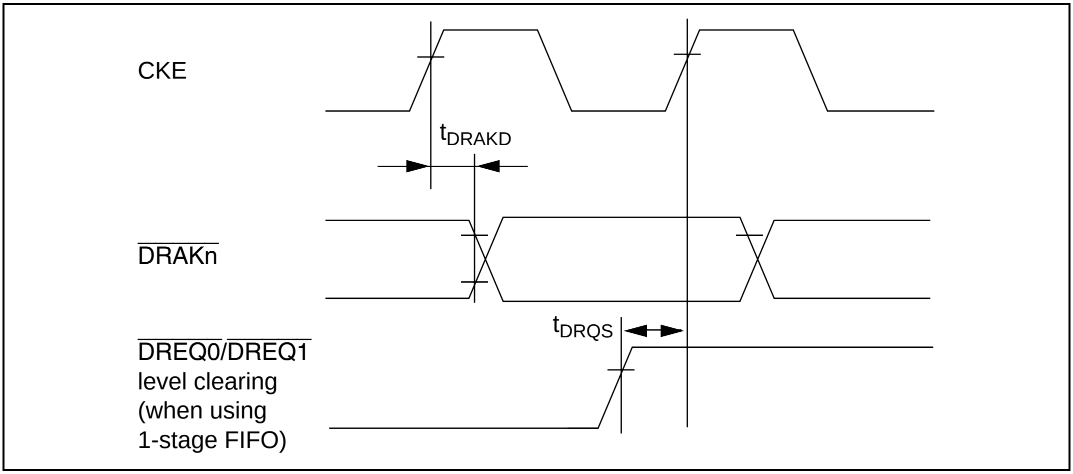

CKE
t_{DRAKD}
$\overline{\text{DRAKn}}$
t_{DRQS}
$\overline{\text{DREQ0}}/\overline{\text{DREQ1}}$
level clearing
(when using
1-stage FIFO)

Figure 22.23 $\overline{\text{DREQ0}}$ and $\overline{\text{DREQ1}}$ Input Timing (3)

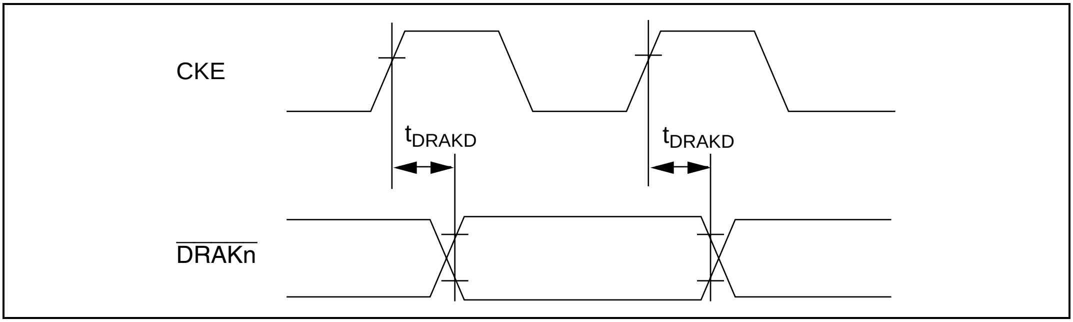


Figure 22.24 $\overline{\text{DRAK}}$ Output Delay Time

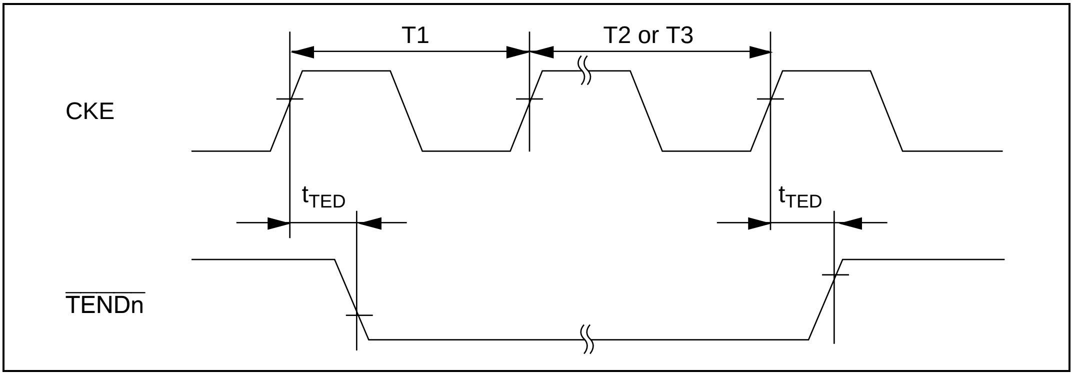


Figure 22.25 $\overline{\text{TEND}}$ Output Delay Time

22.3.5 16-Bit Timer Pulse Unit (TPU) Timing

Table 22.8 16-Bit Timer Pulse Unit (TPU) Timing

Conditions: V_{CC} = PLLV_{CC} = 3.0 to 3.6 V, AV_{CC} = 3.0 to 3.6 V, AV_{CC} = V_{CC} ±10%,
V_{SS} = PLLV_{SS} = PV_{SS} = AV_{SS} = 0 V, Ta = –20 to +75°C

Item	Symbol	Min	Max	Unit	Figure
Output compare output delay time	t_{TOCD}	—	100	ns	Figure 22.26
Input capture input setup time	t_{TICS}	35	—	ns	
Timer input setup time	t_{TCKS}	35	—	ns	Figure 22.27
Timer clock pulse width (single-edge specification)	$t_{TCKWH/L}$	1.5	—	t_{cyc}	
Timer clock pulse width (both-edge specification)	$t_{TCKWH/L}$	2.5	—	t_{cyc}	
Timer clock pulse width (phase counting mode)	$t_{TCKWH/L}$	2.5	—	t_{cyc}	

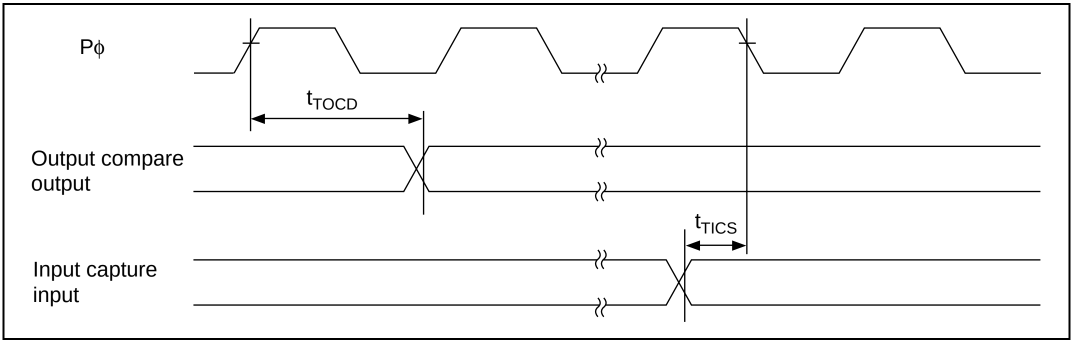


Figure 22.26 TPU Input/Output Timing

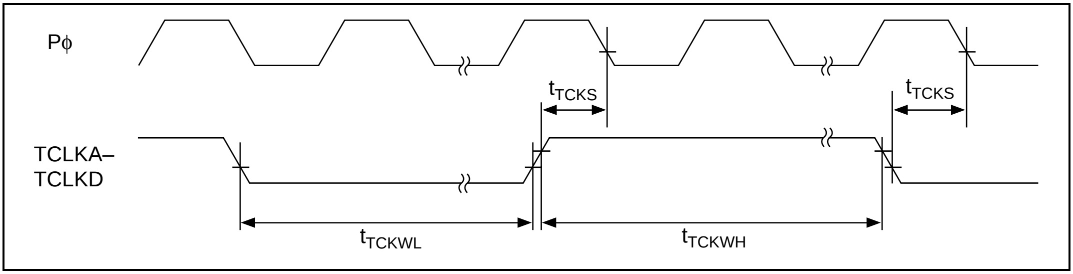


Figure 22.27 TPU Clock Input Timing

22.3.6 Motor Management Timer (MMT) Timing

Table 22.9 Motor Management Timer (MMT) Timing

Conditions: V_{CC} = $PLLV_{CC}$ = 3.0 to 3.6 V, AV_{CC} = 3.0 to 3.6 V, AV_{CC} = V_{CC} ±10%,
V_{SS} = $PLLV_{SS}$ = PV_{SS} = AV_{SS} = 0 V, Ta = –20 to +75°C

Item	Symbol	Min	Max	Unit	Figure
MMT output delay time	t_{MTOD}	—	100	ns	Figure 22.28
PCI input setup time	t_{PCIS}	35	—	ns	
PCI input pulse width	t_{PCIW}	1.5	—	t_{cyc}	

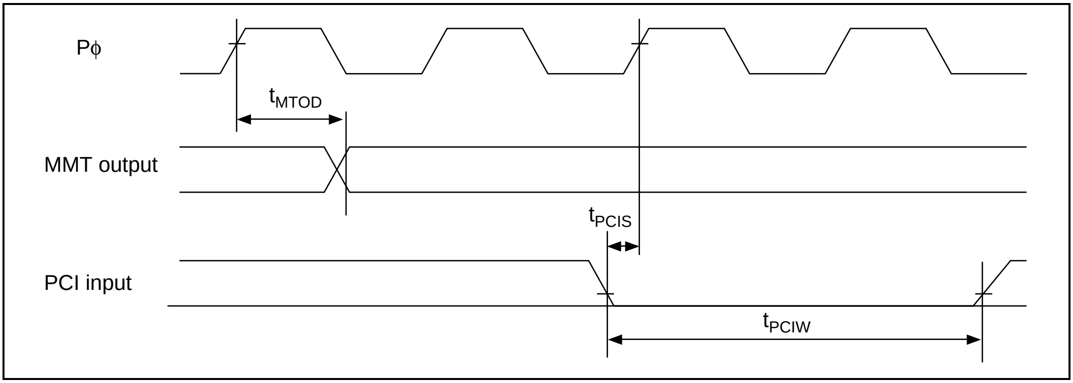


Figure 22.28 MMT Input/Output Timing

22.3.7 Port Output Enable (POE) Timing

Table 22.10 Port Output Enable (POE) Timing

Conditions: V_{CC} = $PLLV_{CC}$ = 3.0 to 3.6 V, AV_{CC} = 3.0 to 3.6 V, AV_{CC} = V_{CC} ±10%,
V_{SS} = $PLLV_{SS}$ = PV_{SS} = AV_{SS} = 0 V, Ta = –20 to +75°C

Item	Symbol	Min	Max	Unit	Figure
$\overline{\text{POE}}$ input setup time	t_{POES}	35	—	ns	Figure 22.29
$\overline{\text{POE}}$ input pulse width	t_{POEW}	1.5	—	tcyc	

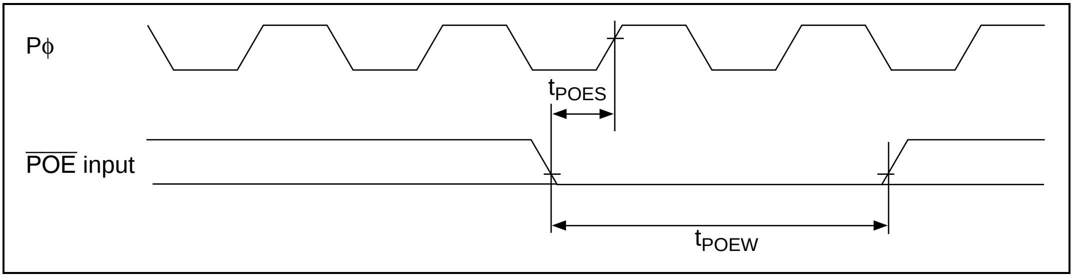


Figure 22.29 $\overline{\text{POE}}$ Input/Output Timing

22.3.8 I/O Port Timing

Table 22.11 I/O Port Timing

Conditions: V_{CC} = PLLV_{CC} = 3.0 to 3.6 V, AV_{CC} = 3.0 to 3.6 V, AV_{CC} = V_{CC} ±10%,
V_{SS} = PLLV_{SS} = PV_{SS} = AV_{SS} = 0 V, Ta = –20 to +75°C

Item	Symbol	Min	Max	Unit	Figure
Port output data delay time	t_{PWD}	—	100	ns	Figure 22.30
Port input hold time	t_{PRH}	100	—	ns	
Port input setup time	t_{PRS}	100	—	ns	

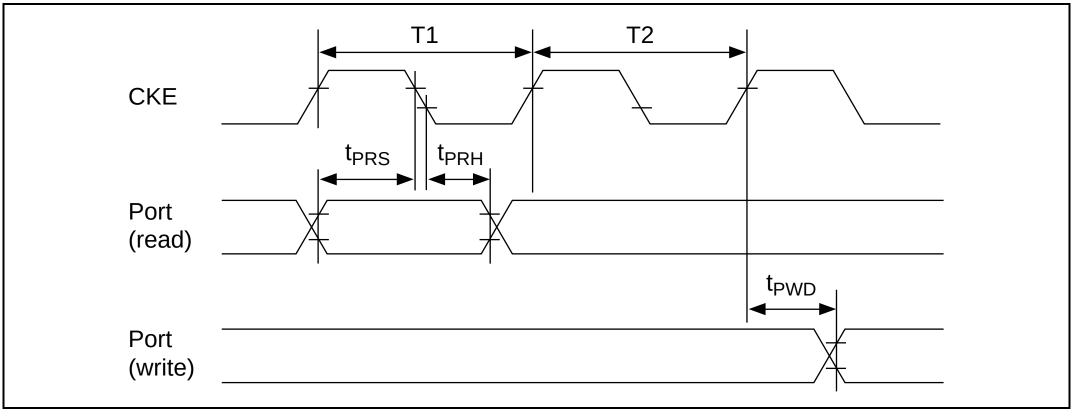


Figure 22.30 I/O Port Input/Output Timing

22.3.9 Watchdog Timer Timing

Table 22.12 Watchdog Timer Timing

Conditions: V_{CC} = $PLLV_{CC}$ = 3.0 to 3.6 V, AV_{CC} = 3.0 to 3.6 V, AV_{CC} = V_{CC} ±10%,
V_{SS} = $PLLV_{SS}$ = PV_{SS} = AV_{SS} = 0 V, Ta = –20 to +75°C

Item	Symbol	Min	Max	Unit	Figure
$\overline{\text{WDTOVF}}$ delay time	t_{WOVD}	—	100	ns	Figure 22.31

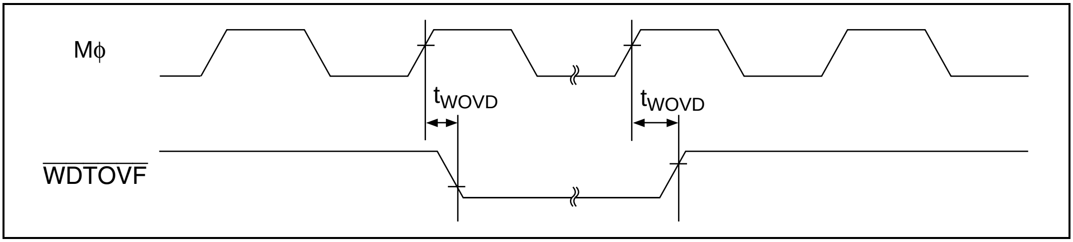


Figure 22.31 Watchdog Timer Timing

22.3.10 Serial Communication Interface Timing

Table 22.13 Serial Communication Interface Timing

Conditions: V_{CC} = $PLLV_{CC}$ = 3.0 to 3.6 V, AV_{CC} = 3.0 to 3.6 V, AV_{CC} = V_{CC} ±10%,
V_{SS} = $PLLV_{SS}$ = PV_{SS} = AV_{SS} = 0 V, Ta = –20 to +75°C

Item	Symbol	Min	Max	Unit	Figure
Input clock cycle	t_{scyc}	4	—	Pϕ	Figure 22.32, Figure 22.33
Input clock cycle (synchronous)	t_{scyc}	6	—	Pϕ	
Input clock pulse width	t_{sckw}	0.4	0.6	t_{scyc}	
Input clock rise time	t_{sckr}	—	1.5	Pϕ	
Input clock fall time	t_{sckf}	—	1.5	Pϕ	
Transmit data delay time (synchronous)	t_{TXD}	—	100	ns	Figure 22.33
Receive data setup time (synchronous)	t_{RXS}	100	—	ns	
Receive data hold time (synchronous)	t_{RXH}	100	—	ns	

Note: When the SCI output pin is set as an open-drain output, the characteristics depend on the pull-up resistance.

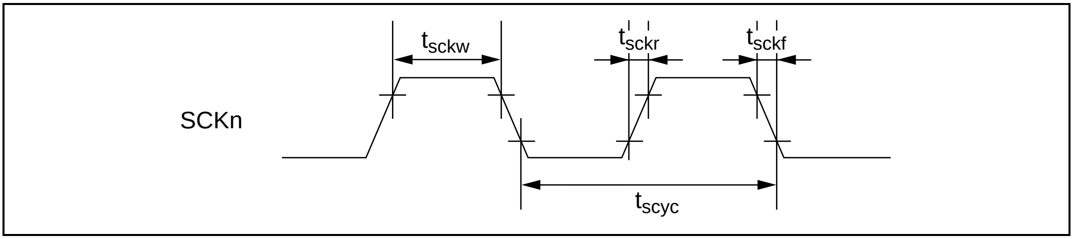


Figure 22.32 Input Clock Timing

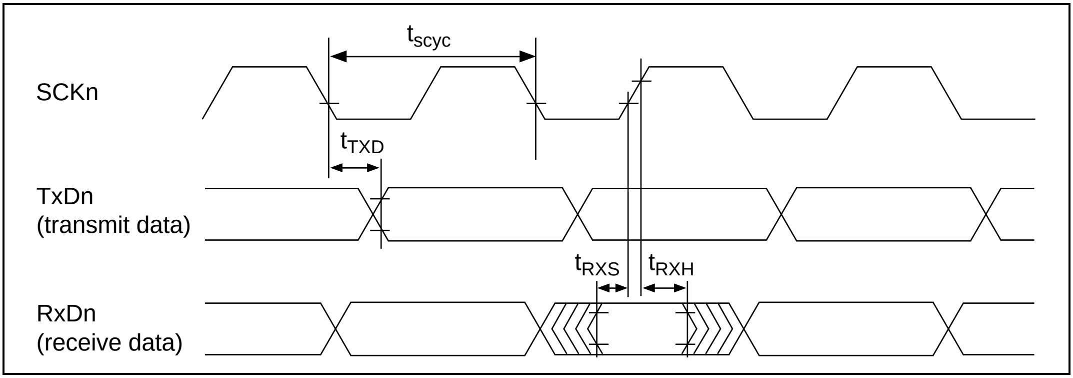


Figure 22.33 SCI Input/Output Timing (Synchronous Mode)

22.3.11 A/D Conversion Timing

Table 22.14 A/D Conversion Timing

Conditions: V_{CC} = PLLV_{CC} = 3.0 to 3.6 V, AV_{CC} = 3.0 to 3.6 V, AV_{CC} = V_{CC} ±10%,
V_{SS} = PLLV_{SS} = PV_{SS} = AV_{SS} = 0 V, Ta = –20 to +75°C

Item		Symbol	Min	Typ	Max	Unit	Figure
External trigger input pulse width		t_{TRGW}	2	—	—	Pϕ	Figure 22.34
External trigger input start delay time		t_{TRGS}	50	—	—	ns	
A/D conversion start delay time	CKS = 0	t_D	10	—	17	Pϕ	Figure 22.35
	CKS = 1		6	—	9	Pϕ	
Input sampling time	CKS = 0	t_{SPL}	—	64	—	Pϕ	
	CKS = 1		—	32	—	Pϕ	
A/D conversion time	CKS = 0	t_{CONV}	259	—	266	Pϕ	
	CKS = 1		131	—	134	Pϕ	

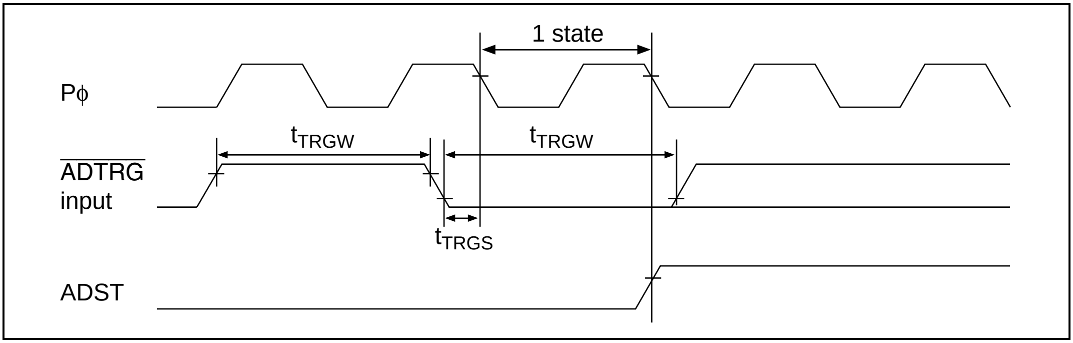
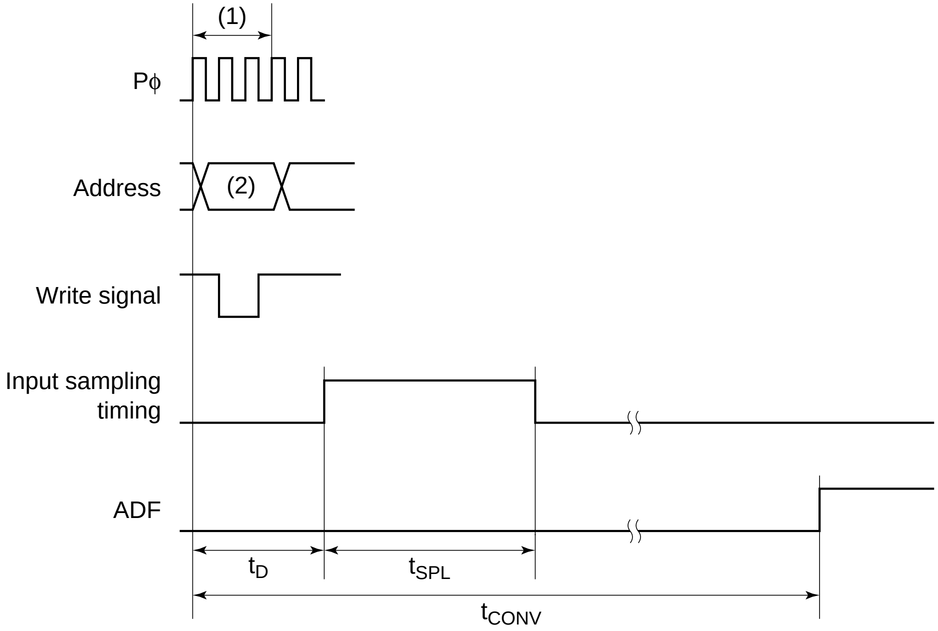


Figure 22.34 External Trigger Input Timing



Legend:

(1): ADCSR write cycle
(2): ADCSR address
t_D: A/D conversion start delay time
t_{SPL}: Input sampling time
t_{CONV}: A/D conversion time

Figure 22.35 A/D Conversion Timing

22.3.12 A/D Conversion Characteristics

Table 22.15 A/D Conversion Characteristics

Conditions: V_{CC} = $PLLV_{CC}$ = 3.0 to 3.6 V, AV_{CC} = 3.0 to 3.6 V, AV_{CC} = V_{CC} ±10%,
V_{SS} = $PLLV_{SS}$ = PV_{SS} = AV_{SS} = 0 V, Ta = –20 to +75°C

	20 MHz			30 MHz			
Item	**Min**	**Typ**	**Max**	**Min**	**Typ**	**Max**	**Unit**
Resolution	10	10	10	10	10	10	Bits
Conversion time[*2]	—	—	6.7 (CKS = 1)	—	—	8.9 (CKS = 0)	μs
Analog input capacitance	—	—	20	—	—	20	pF
Permissible signal source impedance	—	—	1	—	—	1	kΩ
Nonlinearity error[*2]	—	—	±3.5[*1]	—	—	±3.5[*1]	LSB
Offset error[*2]	—	—	±3.5[*1]	—	—	±3.5[*1]	LSB
Full-scale error[*2]	—	—	±3.5[*1]	—	—	±3.5[*1]	LSB
Quantization error[*2]	—	±0.5[*1]	—	—	±0.5[*1]	—	LSB
Absolute accuracy	—	—	±4	—	—	±4	LSB

Notes: 1. Reference values
2. Pϕ ≤ 20 MHz: CKS can be set to 0 or 1.
Pϕ > 20 MHz: CKS can only be set to 0.

22.3.13 D/A Conversion Characteristics

Table 22.16 D/A Conversion Characteristics

Conditions: V_{CC} = PLLV_{CC} = 3.0 to 3.6 V, AV_{CC} = 3.0 to 3.6 V, AV_{CC} = V_{CC} ±10%,
V_{SS} = PLLV_{SS} = PV_{SS} = AV_{SS} = 0 V, Ta = –20 to +75°C

	30 MHz			15 MHz				
Item	**Min**	**Typ**	**Max**	**Min**	**Typ**	**Max**	**Unit**	**Test Conditions**
Resolution	8	8	8	8	8	8	Bits	
Conversion time	—	—	10	—	—	10	µs	20 pF load capacitance
Absolute accuracy	—	±2	±4	—	±2	±4	LSB	2 MΩ load resistance
	—	—	±3	—	—	±3	LSB	4 MΩ load resistance

Appendix A On-Chip Peripheral Module Registers

Table A.1 On-Chip Peripheral Module Registers

Address	Register Name	Bit Names								Module
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FFFF 0080	ADDRA0H	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	A/D
H'FFFF 0081	ADDRA0L	AD1	AD0	—	—	—	—	—	—	
H'FFFF 0082	ADDRB0H	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
H'FFFF 0083	ADDRB0L	AD1	AD0	—	—	—	—	—	—	
H'FFFF 0084	ADDRC0H	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
H'FFFF 0085	ADDRC0L	AD1	AD0	—	—	—	—	—	—	
H'FFFF 0086	ADDRD0H	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
H'FFFF 0087	ADDRD0L	AD1	AD0	—	—	—	—	—	—	
H'FFFF 0088 to H'FFFF 0097	—	—	—	—	—	—	—	—	—	
H'FFFF 0098	ADCSR0	ADF	ADIE	ADST	MULTI	CKS	—	CH1	CH0	
H'FFFF 0099	ADCR0	TRGE1	TRGE0	—	—	—	—	—	—	
H'FFFF 009A to H'FFFF 009F	—	—	—	—	—	—	—	—	—	
H'FFFF 00A0	ADDRA1H	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
H'FFFF 00A1	ADDRA1L	AD1	AD0	—	—	—	—	—	—	
H'FFFF 00A2	ADDRB1H	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
H'FFFF 00A3	ADDRB1L	AD1	AD0	—	—	—	—	—	—	
H'FFFF 00A4	ADDRC1H	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
H'FFFF 00A5	ADDRC1L	AD1	AD0	—	—	—	—	—	—	
H'FFFF 00A6	ADDRD1H	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
H'FFFF 00A7	ADDRD1L	AD1	AD0	—	—	—	—	—	—	
H'FFFF 00A8 to H'FFFF 00B7	—	—	—	—	—	—	—	—	—	
H'FFFF 00B8	ADCSR1	ADF	ADIE	ADST	MULTI	CKS	—	CH1	CH0	
H'FFFF 00B9	ADCR1	TRGE1	TRGE0	—	—	—	—	—	—	
H'FFFF 00BA to H'FFFF 00BF	—	—	—	—	—	—	—	—	—	

Address	Register Name	Bit Names								Module
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FFFF 00C0	DADR0									D/A
H'FFFF 00C1	DADR1									
H'FFFF 00C2	DACR	DAOE1	DAOE0	DAE	—	—	—	—	—	
H'FFFF 00C3 to H'FFFF 03FF	—	—	—	—	—	—	—	—	—	
H'FFFF 0400	TSTR	—	—	CST5	CST4	CST3	CST2	CST1	CST0	TPU
H'FFFF 0401	TSYR	—	—	SYNC5	SYNC4	SYNC3	SYNC2	SYNC1	SYNC0	
H'FFFF 0402 to H'FFFF 040F	—	—	—	—	—	—	—	—	—	
H'FFFF 0410	TCR0	CCLR2	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	
H'FFFF 0411	TMDR0	—	—	BFB	BFA	MD3	MD2	MD1	MD0	
H'FFFF 0412	TIOR0H	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0	
H'FFFF 0413	TIOR0L	IOD3	IOD2	IOD1	IOD0	IOC3	IOC2	IOC1	IOC0	
H'FFFF 0414	TIER0	TTGE	—	—	TCIEV	TGIED	TGIEC	TGIEB	TGIEA	
H'FFFF 0415	TSR0	—	—	—	TCFV	TGFD	TGFC	TGFB	TGFA	
H'FFFF 0416	TCNT0									
H'FFFF 0417										
H'FFFF 0418	TGR0A									
H'FFFF 0419										
H'FFFF 041A	TGR0B									
H'FFFF 041B										
H'FFFF 041C	TGR0C									
H'FFFF 041D										
H'FFFF 041E	TGR0D									
H'FFFF 041F										
H'FFFF 0420	TCR1	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	
H'FFFF 0421	TMDR1	—	—	—	—	MD3	MD2	MD1	MD0	
H'FFFF 0422	TIOR1	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0	
H'FFFF 0423	—	—	—	—	—	—	—	—	—	
H'FFFF 0424	TIER1	TTGE	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA	
H'FFFF 0425	TSR1	TCFD	—	TCFU	TCFV	—	—	TGFB	TGFA	
H'FFFF 0426	TCNT1									
H'FFFF 0427										
H'FFFF 0428	TGR1A									
H'FFFF 0429										

Address	Register Name	Bit Names								Module
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FFFF 042A	TGR1B									TPU
H'FFFF 042B										
H'FFFF 042C to H'FFFF 042F	—	—	—	—	—	—	—	—	—	
H'FFFF 0430	TCR2	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	
H'FFFF 0431	TMDR2	—	—	—	—	MD3	MD2	MD1	MD0	
H'FFFF 0432	TIOR2	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0	
H'FFFF 0433	—	—	—	—	—	—	—	—	—	
H'FFFF 0434	TIER2	TTGE	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA	
H'FFFF 0435	TSR2	TCFD	—	TCFU	TCFV	—	—	TGFB	TGFA	
H'FFFF 0436	TCNT2									
H'FFFF 0437										
H'FFFF 0438	TGR2A									
H'FFFF 0439										
H'FFFF 043A	TGR2B									
H'FFFF 043B										
H'FFFF 043C to H'FFFF 043F	—	—	—	—	—	—	—	—	—	
H'FFFF 0440	TCR3	CCLR2	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	
H'FFFF 0441	TMDR3	—	—	BFB	BFA	MD3	MD2	MD1	MD0	
H'FFFF 0442	TIOR3H	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0	
H'FFFF 0443	TIOR3L	IOD3	IOD2	IOD1	IOD0	IOC3	IOC2	IOC1	IOC0	
H'FFFF 0444	TIER3	TTGE	—	—	TCIEV	TGIED	TGIEC	TGIEB	TGIEA	
H'FFFF 0445	TSR3	—	—	—	TCFV	TGFD	TGFC	TGFB	TGFA	
H'FFFF 0446	TCNT3									
H'FFFF 0447										
H'FFFF 0448	TGR3A									
H'FFFF 0449										
H'FFFF 044A	TGR3B									
H'FFFF 044B										
H'FFFF 044C	TGR3C									
H'FFFF 044D										
H'FFFF 044E	TGR3D									
H'FFFF 044F										
H'FFFF 0450	TCR4	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	

Address	Register Name	Bit Names								Module
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FFFF 0451	TMDR4	—	—	—	—	MD3	MD2	MD1	MD0	TPU
H'FFFF 0452	TIOR4	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0	
H'FFFF 0453	—	—	—	—	—	—	—	—	—	
H'FFFF 0454	TIER4	TTGE	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA	
H'FFFF 0455	TSR4	TCFD	—	TCFU	TCFV	—	—	TGFB	TGFA	
H'FFFF 0456	TCNT4									
H'FFFF 0457										
H'FFFF 0458	TGR4A									
H'FFFF 0459										
H'FFFF 045A	TGR4B									
H'FFFF 045B										
H'FFFF 045C to H'FFFF 045F	—	—	—	—	—	—	—	—	—	
H'FFFF 0460	TCR5	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	
H'FFFF 0461	TMDR5	—	—	—	—	MD3	MD2	MD1	MD0	
H'FFFF 0462	TIOR5	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0	
H'FFFF 0463	—	—	—	—	—	—	—	—	—	
H'FFFF 0464	TIER5	TTGE	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA	
H'FFFF 0465	TSR5	TCFD	—	TCFU	TCFV	—	—	TGFB	TGFA	
H'FFFF 0466	TCNT5									
H'FFFF 0467										
H'FFFF 0468	TGR5A									
H'FFFF 0469										
H'FFFF 046A	TGR5B									
H'FFFF 046B										
H'FFFF 046C to H'FFFF 047F	—	—	—	—	—	—	—	—	—	
H'FFFF 0480	TMDR	—	—	—	—	OLSN	OLSP	MD1	MD0	MMT
H'FFFF 0481	—	—	—	—	—	—	—	—	—	
H'FFFF 0482	TCNR	TTGE	CST	RPRO	—	—	—	TGIEN	TGIEM	
H'FFFF 0483	—	—	—	—	—	—	—	—	—	
H'FFFF 0484	TSR	TCFD	—	—	—	—	—	TGFN	TGFM	
H'FFFF 0485	—	—	—	—	—	—	—	—	—	
H'FFFF 0486	TCNT									
H'FFFF 0487										

Address	Register Name	Bit Names								Module
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FFFF 0488	TPDR									MMT
H'FFFF 0489										
H'FFFF 048A	TPBR									
H'FFFF 048B										
H'FFFF 048C	TDDR									
H'FFFF 048D										
H'FFFF 048E	—	—	—	—	—	—	—	—	—	
H'FFFF 048F	—	—	—	—	—	—	—	—	—	
H'FFFF 0490	TBRU									
H'FFFF 0491										
H'FFFF 0492	TGRUU									
H'FFFF 0493										
H'FFFF 0494	TGRU									
H'FFFF 0495										
H'FFFF 0496	TGRUD									
H'FFFF 0497										
H'FFFF 0498	TDCNT0									
H'FFFF 0499										
H'FFFF 049A	TDCNT1									
H'FFFF 049B										
H'FFFF 049C	TBRU									
H'FFFF 049D										
H'FFFF 049E	—	—	—	—	—	—	—	—	—	
H'FFFF 049F	—	—	—	—	—	—	—	—	—	
H'FFFF 04A0	TBRV									
H'FFFF 04A1										
H'FFFF 04A2	TGRVU									
H'FFFF 04A3										
H'FFFF 04A4	TGRV									
H'FFFF 04A5										
H'FFFF 04A6	TGRVD									
H'FFFF 04A7										
H'FFFF 04A8	TDCNT2									
H'FFFF 04A9										
H'FFFF 04AA	TDCNT3									
H'FFFF 04AB										

Address	Register Name	Bit Names								Module
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FFFF 04AC	TBRV									MMT
H'FFFF 04AD										
H'FFFF 04AE	—	—	—	—	—	—	—	—	—	
H'FFFF 04AF	—	—	—	—	—	—	—	—	—	
H'FFFF 04B0	TBRW									
H'FFFF 04B1										
H'FFFF 04B2	TGRWU									
H'FFFF 04B3										
H'FFFF 04B4	TGRW									
H'FFFF 04B5										
H'FFFF 04B6	TGRWD									
H'FFFF 04B7										
H'FFFF 04B8	TDCNT4									
H'FFFF 04B9										
H'FFFF 04BA	TDCNT5									
H'FFFF 04BB										
H'FFFF 04BC	TBRW									
H'FFFF 04BD										
H'FFFF 04BE	—	—	—	—	—	—	—	—	—	
H'FFFF 04BF	—	—	—	—	—	—	—	—	—	
H'FFFF 04C0	CMSTR	—	—	—	—	—	—	—	—	CMT
H'FFFF 04C1		—	—	—	—	—	—	STR1	STR0	
H'FFFF 04C2	CMCSR0	—	—	—	—	—	—	—	—	
H'FFFF 04C3		CMF	CMIE	—	—	—	—	CKS1	CKS0	
H'FFFF 04C4	CMCNT0									
H'FFFF 04C5										
H'FFFF 04C6	CMCOR0									
H'FFFF 04C7										
H'FFFF 04C8	CMCSR1	—	—	—	—	—	—	—	—	
H'FFFF 04C9		CMF	CMIE	—	—	—	—	CKS1	CKS0	
H'FFFF 04CA	CMCNT1									
H'FFFF 04CB										
H'FFFF 04CC	CMCOR1									
H'FFFF 04CD										
H'FFFF 04CE to H'FFFF 04DF	—	—	—	—	—	—	—	—	—	

Address	Register Name	Bit Names								Module
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FFFF 04E0	ICSR	POE3F	POE2F	POE1F	POE0F	—	—	—	PIE	POE
H'FFFF 04E1		POE3M1	POE3M0	POE2M1	POE2M0	POE1M1	POE1M0	POE0M1	POE0M0	
H'FFFF 04E2 to H'FFFF 04FF	—	—	—	—	—	—	—	—	—	
H'FFFF 0500	SCSMR0	C/A	CHR/ ICK3	PE/ICK2	O/E/ICK1	STOP/ ICK0	MP	CKS1	CKS0	SCI
H'FFFF 0501	—	—	—	—	—	—	—	—	—	
H'FFFF 0502	SCBRR0									
H'FFFF 0503	—	—	—	—	—	—	—	—	—	
H'FFFF 0504	SCSCR0	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0	
H'FFFF 0505	—	—	—	—	—	—	—	—	—	
H'FFFF 0506	SCFTDR0									
H'FFFF 0507	—	—	—	—	—	—	—	—	—	
H'FFFF 0508	SC1SSR0	PER3	PER2	PER1	PER0	FER3	FER2	FER1	FER0	
H'FFFF 0509		TDFE	RDF	ORER	FER	PER	TEND	MPB	MPBT	
H'FFFF 050A	SC2SSR0	TLM	RLM	N1	N0	BRK	DR	EI	ER	
H'FFFF 050B	—	—	—	—	—	—	—	—	—	
H'FFFF 050C	SCFRDR0									
H'FFFF 050D	—	—	—	—	—	—	—	—	—	
H'FFFF 050E	SCFCR0	RTRG1	RTRG0	TTRG1	TTRG0	—	TFRST	RFRST	LOOP	
H'FFFF 050F	—	—	—	—	—	—	—	—	—	
H'FFFF 0510	SCFDR0	—	—	—	T4	T3	T2	T1	T0	
H'FFFF 0511		—	—	—	R4	R3	R2	R1	R0	
H'FFFF 0512	SCFER0	ED15	ED14	ED13	ED12	ED11	ED10	ED9	ED8	
H'FFFF 0513		ED7	ED6	ED5	ED4	ED3	ED2	ED1	ED0	
H'FFFF 0514	SCIMR0	IRMOD	PSEL	RIVS	—	—	—	—	—	
H'FFFF 0515 to H'FFFF 051F	—	—	—	—	—	—	—	—	—	
H'FFFF 0520	SCSMR1	C/A	CHR/ ICK3	PE/ICK2	O/E/ICK1	STOP/ ICK0	MP	CKS1	CKS0	
H'FFFF 0521	—	—	—	—	—	—	—	—	—	
H'FFFF 0522	SCBRR1									
H'FFFF 0523	—	—	—	—	—	—	—	—	—	
H'FFFF 0524	SCSCR1	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0	
H'FFFF 0525	—	—	—	—	—	—	—	—	—	
H'FFFF 0526	SCFTDR1									

Address	Register Name	Bit Names								Module
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FFFF 0527	—	—	—	—	—	—	—	—	—	SCI
H'FFFF 0528	SC1SSR1	PER3	PER2	PER1	PER0	FER3	FER2	FER1	FER0	
H'FFFF 0529		TDFE	RDF	ORER	FER	PER	TEND	MPB	MPBT	
H'FFFF 052A	SC2SSR1	TLM	RLM	N1	N0	BRK	DR	EI	ER	
H'FFFF 052B	—	—	—	—	—	—	—	—	—	
H'FFFF 052C	SCFRDR1									
H'FFFF 052D	—	—	—	—	—	—	—	—	—	
H'FFFF 052E	SCFCR1	RTRG1	RTRG0	TTRG1	TTRG0	—	TFRST	RFRST	LOOP	
H'FFFF 052F	—	—	—	—	—	—	—	—	—	
H'FFFF 0530	SCFDR1	—	—	—	T4	T3	T2	T1	T0	
H'FFFF 0531		—	—	—	R4	R3	R2	R1	R0	
H'FFFF 0532	SCFER1	ED15	ED14	ED13	ED12	ED11	ED10	ED9	ED8	
H'FFFF 0533		ED7	ED6	ED5	ED4	ED3	ED2	ED1	ED0	
H'FFFF 0534	SCIMR1	IRMOD	PSEL	RIVS	—	—	—	—	—	
H'FFFF 0535 to H'FFFF 053F	—	—	—	—	—	—	—	—	—	
H'FFFF 0540	SCSMR2	C/A	CHR/ ICK3	PE/ICK2	O/E/ICK1	STOP/ ICK0	MP	CKS1	CKS0	
H'FFFF 0541	—	—	—	—	—	—	—	—	—	
H'FFFF 0542	SCBRR2									
H'FFFF 0543	—	—	—	—	—	—	—	—	—	
H'FFFF 0544	SCSCR2	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0	
H'FFFF 0545	—	—	—	—	—	—	—	—	—	
H'FFFF 0546	SCFTDR2									
H'FFFF 0547	—	—	—	—	—	—	—	—	—	
H'FFFF 0548	SC1SSR2	PER3	PER2	PER1	PER0	FER3	FER2	FER1	FER0	
H'FFFF 0549		TDFE	RDF	ORER	FER	PER	TEND	MPB	MPBT	
H'FFFF 054A	SC2SSR2	TLM	RLM	N1	N0	BRK	DR	EI	ER	
H'FFFF 054B	—	—	—	—	—	—	—	—	—	
H'FFFF 054C	SCFRDR2									
H'FFFF 054D	—	—	—	—	—	—	—	—	—	
H'FFFF 054E	SCFDR2	RTRG1	RTRG0	TTRG1	TTRG0	—	TFRST	RFRST	LOOP	
H'FFFF 054F	—	—	—	—	—	—	—	—	—	
H'FFFF 0550	SCFDR2	—	—	—	T4	T3	T2	T1	T0	
H'FFFF 0551		—	—	—	R4	R3	R2	R1	R0	

Address	Register Name	Bit Names								Module
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FFFF 0552	SCFER2	ED15	ED14	ED13	ED12	ED11	ED10	ED9	ED8	SCI
H'FFFF 0553		ED7	ED6	ED5	ED4	ED3	ED2	ED1	ED0	
H'FFFF 0554	SCIMR2	IRMOD	PSEL	RIVS	—	—	—	—	—	
H'FFFF 0555 to H'FFFF 07FF	—	—	—	—	—	—	—	—	—	
H'FFFF 0800	FLMCR1	FWE	SWE	ESU	PSU	EV	PV	E	P	FLASH
H'FFFF 0801	FLMCR2	FLER	—	—	—	—	—	—	—	
H'FFFF 0802	EBR1	EB7	EB6	EB5	EB4	EB3	EB2	EB1	EB0	
H'FFFF 0803	EBR2	—	—	—	—	EB11	EB10	EB9	EB8	
H'FFFF 0804 to H'FFFF 0BFF	—	—	—	—	—	—	—	—	—	
H'FFFF 0C00	BCR	BRQE	BAS	HIZCNT	—	—	—	—	—	BSC
H'FFFF 0C01		—	—	—	—	—	—	—	—	
H'FFFF 0C02 to H'FFFF 0C0F	—	—	—	—	—	—	—	—	—	
H'FFFF 0C10	ACR1_0	ENDIAN	TP1	TP0	EXWE	—	SZ1	SZ0	IW2	
H'FFFF 0C11		IW1	IW0	SWH2	SWH1	SWH0	SWT2	SWT1	SWT0	
H'FFFF 0C12	ACR1_1	ENDIAN	TP1	TP0	EXWE	—	SZ1	SZ0	IW2	
H'FFFF 0C13		IW1	IW0	SWH2	SWH1	SWH0	SWT2	SWT1	SWT0	
H'FFFF 0C14	ACR1_2	ENDIAN	TP1	TP0	EXWE	—	SZ1	SZ0	IW2	
H'FFFF 0C15		IW1	IW0	SWH2	SWH1	SWH0	SWT2	SWT1	SWT0	
H'FFFF 0C16	ACR1_3	ENDIAN	TP1	TP0	EXWE	—	SZ1	SZ0	IW2	
H'FFFF 0C17		IW1	IW0	SWH2	SWH1	SWH0	SWT2	SWT1	SWT0	
H'FFFF 0C18 to H'FFFF 0C1F	—	—	—	—	—	—	—	—	—	
H'FFFF 0C20	ACR1_4	ENDIAN	—	—	EXWE	—	—	—	—	
H'FFFF 0C21		—	—	—	—	—	—	—	—	
H'FFFF 0C22	ACR1_5	ENDIAN	—	—	EXWE	—	—	—	—	
H'FFFF 0C23		—	—	—	—	—	—	—	—	
H'FFFF 0C24 to H'FFFF 0C2F	—	—	—	—	—	—	—	—	—	
H'FFFF 0C30	WCR_0	W3	W2	W1	W0	DSWW3	DSWW2	DSWW1	DSWW0	
H'FFFF 0C31		DSWR3	DSWR2	DSWR1	DSWR0	HWW2	HWW1	HWW0	—	

Address	Register Name	Bit Names								Module
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FFFF 0C32	WCR_1	W3	W2	W1	W0	DSWW3	DSWW2	DSWW1	DSWW0	BSC
H'FFFF 0C33		DSWR3	DSWR2	DSWR1	DSWR0	HWW2	HWW1	HWW0	—	
H'FFFF 0C34	WCR_2	W3	W2	W1	W0	DSWW3	DSWW2	DSWW1	DSWW0	
H'FFFF 0C35		DSWR3	DSWR2	DSWR1	DSWR0	HWW2	HWW1	HWW0	—	
H'FFFF 0C36	WCR_3	W3	W2	W1	W0	DSWW3	DSWW2	DSWW1	DSWW0	
H'FFFF 0C37		DSWR3	DSWR2	DSWR1	DSWR0	HWW2	HWW1	HWW0	—	
H'FFFF 0C38 to H'FFFF 0C3F	—	—	—	—	—	—	—	—	—	
H'FFFF 0C40	DCR1	TPC1	TPC0	TPCS2	TPCS1	TPCS0	RCD2	RCD1	RCD0	
H'FFFF 0C41		—	—	DWW1	DWW0	DWR1	DWR0	—	—	
H'FFFF 0C42	DCR2	DIW2	DIW1	DIW0	DDWW3	DDWW2	DDWW1	DDWW0	DDWR3	
H'FFFF 0C43		DDWR2	DDWR1	DDWR0	RDW	TCAS	—	—	—	
H'FFFF 0C44	DCR3	BE	RSD	EDO	DSZ1	DSZ0	AMX2	AMX1	AMX0	
H'FFFF 0C45		RFSH	RMD	—	—	—	—	—	—	
H'FFFF 0C46 to H'FFFF 0C67	—	—	—	—	—	—	—	—	—	
H'FFFF 0C68	RTCSR	CMF	CMIE	CKS2	CKS1	CKS0	OVF	OVIE	LMTS1	
H'FFFF 0C69		LMTS0	BREF2	BREF1	BREF0	TRAS2	TRAS1	TRAS0	—	
H'FFFF 0C6A	RTCNT	—	—	—	—	—	—	—	—	
H'FFFF 0C6B		RTCNT7	RTCNT6	RTCNT5	RTCNT4	RTCNT3	RTCNT2	RTCNT1	RTCNT0	
H'FFFF 0C6C	RTCOR	—	—	—	—	—	—	—	—	
H'FFFF 0C6D		RTCOR7	RTCOR6	RTCOR5	RTCOR4	RTCOR3	RTCOR2	RTCOR1	RTCOR0	
H'FFFF 0C6E	RFCR	—	—	—	—	RFCR11	RFCR10	RFCR9	RFCR8	
H'FFFF 0C6F		RFCR7	RFCR6	RFCR5	RFCR4	RFCR3	RFCR2	RFCR1	RFCR0	
H'FFFF 0C70	RAMER	—	—	—	—	—	—	—	—	FLASH
H'FFFF 0C71		—	—	—	RAMAS	RAMS	RAM2	RAM1	RAM0	
H'FFFF 0C72 to H'FFFF 0C7F	—	—	—	—	—	—	—	—	—	
H'FFFF 0C80	UBARH	UBA31	UBA30	UBA29	UBA28	UBA27	UBA26	UBA25	UBA24	UBC
H'FFFF 0C81		UBA23	UBA22	UBA21	UBA20	UBA19	UBA18	UBA17	UBA16	
H'FFFF 0C82	UBARL	UBA15	UBA14	UBA13	UBA12	UBA11	UBA10	UBA9	UBA8	
H'FFFF 0C83		UBA7	UBA6	UBA5	UBA4	UBA3	UBA2	UBA1	UBA0	
H'FFFF 0C84	UBAMRH	UBM31	UBM30	UBM29	UBM28	UBM27	UBM26	UBM25	UBM24	
H'FFFF 0C85		UBM23	UBM22	UBM21	UBM20	UBM19	UBM18	UBM17	UBM16	

Address	Register Name	Bit Names								Module
		Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**	
H'FFFF 0C86	UBAMRL	UBM15	UBM14	UBM13	UBM12	UBM11	UBM10	UBM9	UBM8	UBC
H'FFFF 0C87		UBM7	UBM6	UBM5	UBM4	UBM3	UBM2	UBM1	UBM0	
H'FFFF 0C88	UBBR	UBIE	—	—	—	—	—	XYE	XYS	
H'FFFF 0C89		CP1	CP0	ID1	ID0	RW1	RW0	SZ1	SZ0	
H'FFFF 0C8A to H'FFFF 0FFF	—	—	—	—	—	—	—	—	—	
H'FFFF 1000	TCSR	OVF	WT/$\overline{\text{IT}}$	TME	—	—	CKS2	CKS1	CKS0	WDT
H'FFFF 1001	TCNT*1	TCNT7	TCNT6	TCNT5	TCNT4	TCNT3	TCNT2	TCNT1	TCNT0	
H'FFFF 1002	RSTCSR*2	WOVF	RSTE	—	—	—	—	—	—	
H'FFFF 1003	RSTCSR*3	WOVF	RSTE	—	—	—	—	—	—	
H'FFFF 1004	SBYCR	SBY	HIZ	—	—	—	—	—	—	
H'FFFF 1005 to H'FFFF 101F	—	—	—	—	—	—	—	—	—	
H'FFFF 1020	MSR	—	—	—	—	—	—	—	—	SYS
H'FFFF 1021		—	—	MD5	MD4	MD3	MD2	MD1	MD0	
H'FFFF 1022 to H'FFFF 1027	—	—	—	—	—	—	—	—	—	
H'FFFF 1028	FRQCR	CKIOOE	CKOE	—	—	—	—	—	—	
H'FFFF 1029		FR7	FR6	FR5	FR4	FR3	FR2	FR1	FR0	
H'FFFF 102A	MODECR	—	—	—	—	—	—	—	—	
H'FFFF 102B		—	—	—	ROMMD	—	—	—	—	
H'FFFF 102C to H'FFFF 102F	—	—	—	—	—	—	—	—	—	
H'FFFF 1030	MSTPCR1	MSTP15	MSTP14	MSTP13	MSTP12	MSTP11	MSTP10	MSTP9	MSTP8	
H'FFFF 1031		MSTP7	MSTP6	MSTP5	MSTP4	MSTP3	MSTP2	MSTP1	MSTP0	
H'FFFF 1032	MSTPCR2	MSTP31	MSTP30	MSTP29	MSTP28	MSTP27	MSTP26	MSTP25	MSTP24	
H'FFFF 1033		MSTP23	MSTP22	MSTP21	MSTP20	MSTP19	MSTP18	MSTP17	MSTP16	
H'FFFF 1034	MCLKCR1	—	MCLK032	MCLK031	MCLK030	—	MCLK022	MCLK021	MCLK020	
H'FFFF 1035		—	MCLK012	MCLK011	MCLK010	—	MCLK002	MCLK001	MCLK000	
H'FFFF 1036	MCLKCR2	—	MCLK072	MCLK071	MCLK070	—	MCLK062	MCLK061	MCLK060	
H'FFFF 1037		—	MCLK052	MCLK051	MCLK050	—	MCLK042	MCLK041	MCLK040	
H'FFFF 1038	MCLKCR3	—	MCLK112	MCLK111	MCLK110	—	MCLK102	MCLK101	MCLK100	
H'FFFF 1039		—	MCLK092	MCLK091	MCLK090	—	MCLK082	MCLK081	MCLK080	

		Bit Names								
Address	Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
H'FFFF 103A	MCLKCR4	—	MCLK152	MCLK151	MCLK150	—	MCLK142	MCLK141	MCLK140	SYS
H'FFFF 103B		—	MCLK132	MCLK131	MCLK130	—	MCLK122	MCLK121	MCLK120	
H'FFFF 103C	MCLKCR5	—	—	MCLK191	MCLK190	—	—	MCLK181	MCLK180	
H'FFFF 103D		—	—	MCLK171	MCLK170	—	—	MCLK161	MCLK160	
H'FFFF 103E to H'FFFF 104F	—	—	—	—	—	—	—	—	—	
H'FFFF 1050	IPRA	IRQ0	IRQ0	IRQ0	IRQ0	IRQ1	IRQ1	IRQ1	IRQ1	INTC
H'FFFF 1051		IRQ2	IRQ2	IRQ2	IRQ2	IRQ3	IRQ3	IRQ3	IRQ3	
H'FFFF 1052	IPRB	IRQ4	IRQ4	IRQ4	IRQ4	IRQ5	IRQ5	IRQ5	IRQ5	
H'FFFF 1053		IRQ6	IRQ6	IRQ6	IRQ6	IRQ7	IRQ7	IRQ7	IRQ7	
H'FFFF 1054	IPRC	—	—	—	—	—	—	—	—	
H'FFFF 1055		—	—	—	—	—	—	—	—	
H'FFFF 1056	IPRD	—	—	—	—	—	—	—	—	
H'FFFF 1057		—	—	—	—	—	—	—	—	
H'FFFF 1058	IPRE	DMAC0	DMAC0	DMAC0	DMAC0	DMAC1	DMAC1	DMAC1	DMAC1	
H'FFFF 1059		DMAC2	DMAC2	DMAC2	DMAC2	DMAC3	DMAC3	DMAC3	DMAC3	
H'FFFF 105A	IPRF	—	—	—	—	—	—	—	—	
H'FFFF 105B		—	—	—	—	—	—	—	—	
H'FFFF 105C	IPRG	BSC	BSC	BSC	BSC	BSC	BSC	BSC	BSC	
H'FFFF 105D		WDT	WDT	WDT	WDT	—	—	—	—	
H'FFFF 105E	IPRH	TPU0	TPU0	TPU0	TPU0	TPU0	TPU0	TPU0	TPU0	
H'FFFF 105F		TPU1	TPU1	TPU1	TPU1	TPU1	TPU1	TPU1	TPU1	
H'FFFF 1060	IPRI	TPU2	TPU2	TPU2	TPU2	TPU2	TPU2	TPU2	TPU2	
H'FFFF 1061		TPU3	TPU3	TPU3	TPU3	TPU3	TPU3	TPU3	TPU3	
H'FFFF 1062	IPRJ	TPU4	TPU4	TPU4	TPU4	TPU4	TPU4	TPU4	TPU4	
H'FFFF 1063		TPU5	TPU5	TPU5	TPU5	TPU5	TPU5	TPU5	TPU5	
H'FFFF 1064	IPRK	SCI0	SCI0	SCI0	SCI0	SCI1	SCI1	SCI1	SCI1	
H'FFFF 1065		SCI2	SCI2	SCI2	SCI2	—	—	—	—	
H'FFFF 1066	IPRL	CMT	CMT	CMT	CMT	A/D	A/D	A/D	A/D	
H'FFFF 1067		MMT	MMT	MMT	MMT	POE	POE	POE	POE	
H'FFFF 1068 to H'FFFF 106D	—	—	—	—	—	—	—	—	—	
H'FFFF 106E	ICR1	NMIL	—	—	—	—	EXIMD	—	NMIE	
H'FFFF 106F		—	—	—	—	—	—	—	—	

Address	Register Name	Bit Names								Module
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FFFF 1070	ICR2	—	—	—	—	—	—	—	—	INTC
H'FFFF 1071		IRQ7S	IRQ6S	IRQ5S	IRQ4S	IRQ3S	IRQ2S	IRQ1S	IRQ0S	
H'FFFF 1072	ISR	—	—	—	—	—	—	—	—	
H'FFFF 1073		IRQ7F	IRQ6F	IRQ5F	IRQ4F	IRQ3F	IRQ2F	IRQ1F	IRQ0F	
H'FFFF 1074 to H'FFFF 10EF	—	—	—	—	—	—	—	—	—	
H'FFFF 10F0	DMAOR	—	—	—	—	RC3	RC2	RC1	RC0	DMA
H'FFFF 10F1		—	—	—	—	—	AE	NMIF	DME	
H'FFFF 10F2 to H'FFFF 10FF	—	—	—	—	—	—	—	—	—	
H'FFFF 1100	SAR0									
H'FFFF 1101										
H'FFFF 1102										
H'FFFF 1103										
H'FFFF 1104	DAR0									
H'FFFF 1105										
H'FFFF 1106										
H'FFFF 1107										
H'FFFF 1108	DMATCR0									
H'FFFF 1109										
H'FFFF 110A										
H'FFFF 110B										
H'FFFF 110C	CHCR0	—	—	—	RS4	RS3	RS2	RS1	RS0	
H'FFFF 110D		—	FIFOS	—	—	NDARE	NSARE	FCS	TES	
H'FFFF 110E		DM1	DM0	SM1	SM0	CHNE	RL	AM	AL	
H'FFFF 110F		TEND	DS	TM	TS1	TS0	IE	TE	DE	
H'FFFF 1110	NSAR0									
H'FFFF 1111										
H'FFFF 1112										
H'FFFF 1113										
H'FFFF 1114	NDAR0									
H'FFFF 1115										
H'FFFF 1116										
H'FFFF 1117										

Address	Register Name	Bit Names								Module
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FFFF 1118	NDMATCR0									DMA
H'FFFF 1119										
H'FFFF 111A										
H'FFFF 111B										
H'FFFF 111C	CHNCNT0									
H'FFFF 111D										
H'FFFF 111E										
H'FFFF 111F										
H'FFFF 1120	SAR1									
H'FFFF 1121										
H'FFFF 1122										
H'FFFF 1123										
H'FFFF 1124	DAR1									
H'FFFF 1125										
H'FFFF 1126										
H'FFFF 1127										
H'FFFF 1128	DMATCR1									
H'FFFF 1129										
H'FFFF 112A										
H'FFFF 112B										
H'FFFF 112C	CHCR1	—	—	—	RS4	RS3	RS2	RS1	RS0	
H'FFFF 112D		—	FIFOS	—	—	NDARE	NSARE	FCS	TES	
H'FFFF 112E		DM1	DM0	SM1	SM0	CHNE	RL	AM	AL	
H'FFFF 112F		TEND	DS	TM	TS1	TS0	IE	TE	DE	
H'FFFF 1130	NSAR1									
H'FFFF 1131										
H'FFFF 1132										
H'FFFF 1133										
H'FFFF 1134	NDAR1									
H'FFFF 1135										
H'FFFF 1136										
H'FFFF 1137										
H'FFFF 1138	NDMATCR1									
H'FFFF 1139										
H'FFFF 113A										
H'FFFF 113B										

Address	Register Name	Bit Names								Module
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FFFF 113C	CHNCNT1									DMA
H'FFFF 113D										
H'FFFF 113E										
H'FFFF 113F										
H'FFFF 1140	SAR2									
H'FFFF 1141										
H'FFFF 1142										
H'FFFF 1143										
H'FFFF 1144	DAR2									
H'FFFF 1145										
H'FFFF 1146										
H'FFFF 1147										
H'FFFF 1148	DMATCR2									
H'FFFF 1149										
H'FFFF 114A										
H'FFFF 114B										
H'FFFF 114C	CHCR2	—	—	—	RS4	RS3	RS2	RS1	RS0	
H'FFFF 114D		—	FIFOS	—	—	NDARE	NSARE	FCS	TES	
H'FFFF 114E		DM1	DM0	SM1	SM0	CHNE	RL	AM	AL	
H'FFFF 114F		TEND	DS	TM	TS1	TS0	IE	TE	DE	
H'FFFF 1150	NSAR2									
H'FFFF 1151										
H'FFFF 1152										
H'FFFF 1153										
H'FFFF 1154	NDAR2									
H'FFFF 1155										
H'FFFF 1156										
H'FFFF 1157										
H'FFFF 1158	NDMATCR2									
H'FFFF 1159										
H'FFFF 115A										
H'FFFF 115B										
H'FFFF 115C	CHNCNT2									
H'FFFF 115D										
H'FFFF 115E										
H'FFFF 115F										

Address	Register Name	Bit Names								Module
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FFFF 1160	SAR3									DMA
H'FFFF 1161										
H'FFFF 1162										
H'FFFF 1163										
H'FFFF 1164	DAR3									
H'FFFF 1165										
H'FFFF 1166										
H'FFFF 1167										
H'FFFF 1168	DMATCR3									
H'FFFF 1169										
H'FFFF 116A										
H'FFFF 116B										
H'FFFF 116C	CHCR3	—	—	—	RS4	RS3	RS2	RS1	RS0	
H'FFFF 116D		—	FIFOS	—	—	NDARE	NSARE	FCS	TES	
H'FFFF 116E		DM1	DM0	SM1	SM0	CHNE	RL	AM	AL	
H'FFFF 116F		TEND	DS	TM	TS1	TS0	IE	TE	DE	
H'FFFF 1170	NSAR3									
H'FFFF 1171										
H'FFFF 1172										
H'FFFF 1173										
H'FFFF 1174	NDAR3									
H'FFFF 1175										
H'FFFF 1176										
H'FFFF 1177										
H'FFFF 1178	NDMATCR3									
H'FFFF 1179										
H'FFFF 117A										
H'FFFF 117B										
H'FFFF 117C	CHNCNT3									
H'FFFF 117D										
H'FFFF 117E										
H'FFFF 117F										
H'FFFF 1180 to H'FFFF 11FF	—	—	—	—	—	—	—	—	—	

Address	Register Name	Bit Names								Module
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FFFF 1200	PADRH	—	—	—	—	—	—	PA25DR	PA24DR	PORT
H'FFFF 1201		PA23DR	PA22DR	PA21DR	PA20DR	PA19DR	PA18DR	PA17DR	PA16DR	
H'FFFF 1202	PADRL	PA15DR	PA14DR	PA13DR	PA12DR	—	—	PA9DR	PA8DR	
H'FFFF 1203		—	—	—	—	—	—	PA1DR	PA0DR	
H'FFFF 1204	PAIORH	—	—	—	—	—	—	PA25IOR	PA24IOR	
H'FFFF 1205		PA23IOR	PA22IOR	PA21IOR	PA20IOR	PA19IOR	PA18IOR	PA17IOR	PA16IOR	
H'FFFF 1206	PAIORL	PA15IOR	PA14IOR	PA13IOR	PA12IOR	—	—	PA9IOR	PA8IOR	
H'FFFF 1207		—	—	—	—	—	—	PA1IOR	PA0IOR	
H'FFFF 1208	PACRH1	—	—	—	—	—	—	—	—	
H'FFFF 1209		—	—	—	—	—	PA25MD	—	PA24MD	
H'FFFF 120A	PACRH2	—	PA23MD	—	PA22MD	—	PA21MD	—	PA20MD	
H'FFFF 120B		—	PA19MD	—	PA18MD	—	PA17MD	PA16MD1	PA16MD0	
H'FFFF 120C	PACRL1	PA15MD1	PA15MD0	—	PA14MD	—	PA13MD	—	PA12MD	
H'FFFF 120D		—	—	—	—	—	PA9MD	—	PA8MD	
H'FFFF 120E	PACRL2	—	—	—	—	—	—	—	—	
H'FFFF 120F		—	—	—	—	—	PA1MD	—	PA0MD	
H'FFFF 1210	PBDRH	—	—	—	—	—	—	—	—	
H'FFFF 1211		PB23DR	PB22DR	PB21DR	PB20DR	PB19DR	PB18DR	PB17DR	PB16DR	
H'FFFF 1212	PBDRL	—	—	PB13DR	—	—	—	—	—	
H'FFFF 1213		PB7DR	PB6DR	—	—	—	—	—	—	
H'FFFF 1214	PBIORH	—	—	—	—	—	—	—	—	
H'FFFF 1215		PB23IOR	PB22IOR	PB21IOR	PB20IOR	PB19IOR	PB18IOR	PB17IOR	PB16IOR	
H'FFFF 1216	PBIORL	—	—	PB13IOR	—	—	—	—	—	
H'FFFF 1217		PB7IOR	PB6IOR	—	—	—	—	—	—	
H'FFFF 1218	PBCRH1	—	—	—	—	—	—	—	—	
H'FFFF 1219		—	—	—	—	—	—	—	—	
H'FFFF 121A	PBCRH2	PB23MD1	PB23MD0	PB22MD1	PB22MD0	—	PB21MD	—	PB20MD	
H'FFFF 121B		PB19MD1	PB19MD0	PB18MD1	PB18MD0	—	PB17MD	—	PB16MD	
H'FFFF 121C	PBCRL1	—	—	—	—	—	PB13MD	—	—	
H'FFFF 121D		—	—	—	—	—	—	—	—	
H'FFFF 121E	PBCRL2	—	PB7MD	—	PB6MD	—	—	—	—	
H'FFFF 121F		—	—	—	—	—	—	—	—	
H'FFFF 1220	PCDRH	—	—	—	—	—	—	PC25DR	PC24DR	
H'FFFF 1221		PC23DR	PC22DR	PC21DR	PC20DR	PC19DR	PC18DR	PC17DR	PC16DR	
H'FFFF 1222	PCDRL	PC15DR	PC14DR	PC13DR	PC12DR	PC11DR	PC10DR	PC9DR	PC8DR	
H'FFFF 1223		PC7DR	PC6DR	PC5DR	PC4DR	PC3DR	PC2DR	PC1DR	PC0DR	

Address	Register Name	Bit Names								Module
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FFFF 1224	PCIORH	—	—	—	—	—	—	PC25IOR	PC24IOR	PORT
H'FFFF 1225		PC23IOR	PC22IOR	PC21IOR	PC20IOR	PC19IOR	PC18IOR	PC17IOR	PC16IOR	
H'FFFF 1226	PCIORL	PC15IOR	PC14IOR	PC13IOR	PC12IOR	PC11IOR	PC10IOR	PC9IOR	PC8IOR	
H'FFFF 1227		PC7IOR	PC6IOR	PC5IOR	PC4IOR	PC3IOR	PC2IOR	PC1IOR	PC0IOR	
H'FFFF 1228	PCCRH1	—	—	—	—	—	—	—	—	
H'FFFF 1229		—	—	—	—	PC25MD1	PC25MD0	PC24MD1	PC24MD0	
H'FFFF 122A	PCCRH2	PC23MD1	PC23MD0	PC22MD1	PC22MD0	PC21MD1	PC21MD0	PC20MD1	PC20MD0	
H'FFFF 122B		PC19MD1	PC19MD0	PC18MD1	PC18MD0	PC17MD1	PC17MD0	PC16MD1	PC16MD0	
H'FFFF 122C	PCCRL1	PC15MD1	PC15MD0	PC14MD1	PC14MD0	—	PC13MD	—	PC12MD	
H'FFFF 122D		—	PC11MD	—	PC10MD	—	PC9MD	—	PC8MD	
H'FFFF 122E	PCCRL2	—	PC7MD	—	PC6MD	—	PC5MD	—	PC4MD	
H'FFFF 122F		—	PC3MD	—	PC2MD	—	PC1MD	—	PC0MD	
H'FFFF 1230	PDDRH	PD31DR	PD30DR	PD29DR	PD28DR	PD27DR	PD26DR	PD25DR	PD24DR	
H'FFFF 1231		PD23DR	PD22DR	PD21DR	PD20DR	PD19DR	PD18DR	PD17DR	PD16DR	
H'FFFF 1232	PDDRL	PD15DR	PD14DR	PD13DR	PD12DR	PD11DR	PD10DR	PD9DR	PD8DR	
H'FFFF 1233		PD7DR	PD6DR	PD5DR	PD4DR	PD3DR	PD2DR	PD1DR	PD0DR	
H'FFFF 1234	PDIORH	PD31IOR	PD30IOR	PD29IOR	PD28IOR	PD27IOR	PD26IOR	PD25IOR	PD24IOR	
H'FFFF 1235		PD23IOR	PD22IOR	PD21IOR	PD20IOR	PD19IOR	PD18IOR	PD17IOR	PD16IOR	
H'FFFF 1236	PDIORL	PD15IOR	PD14IOR	PD13IOR	PD12IOR	PD11IOR	PD10IOR	PD9IOR	PD8IOR	
H'FFFF 1237		PD7IOR	PD6IOR	PD5IOR	PD4IOR	PD3IOR	PD2IOR	PD1IOR	PD0IOR	
H'FFFF 1238	PDCRH1	PD31MD1	PD31MD0	PD30MD1	PD30MD0	PD29MD1	PD29MD0	PD28MD1	PD28MD0	
H'FFFF 1239		PD27MD1	PD27MD0	PD26MD1	PD26MD0	PD25MD1	PD25MD0	PD24MD1	PD24MD0	
H'FFFF 123A	PDCRH2	PD23MD1	PD23MD0	PD22MD1	PD22MD0	PD21MD1	PD21MD0	PD20MD1	PD20MD0	
H'FFFF 123B		PD19MD1	PD19MD0	PD18MD1	PD18MD0	PD17MD1	PD17MD0	PD16MD1	PD16MD0	
H'FFFF 123C	PDCRL1	PD15MD1	PD15MD0	PD14MD1	PD14MD0	PD13MD1	PD13MD0	PD12MD1	PD12MD0	
H'FFFF 123D		PD11MD1	PD11MD0	PD10MD1	PD10MD0	PD9MD1	PD9MD0	PD8MD1	PD8MD0	
H'FFFF 123E	PDCRL2	—	PD7MD	—	PD6MD	—	PD5MD	—	PD4MD	
H'FFFF 123F		—	PD3MD	—	PD2MD	—	PD1MD	—	PD0MD	
H'FFFF 1240	PEDRH	—	—	—	—	—	—	—	—	
H'FFFF 1241		PE23DR	PE22DR	PE21DR	PE20DR	PE19DR	PE18DR	PE17DR	PE16DR	
H'FFFF 1242	PEDRL	PE15DR	PE14DR	PE13DR	PE12DR	—	—	—	—	
H'FFFF 1243		—	—	—	—	—	—	—	—	
H'FFFF 1244	PEIORH	—	—	—	—	—	—	—	—	
H'FFFF 1245		PE23IOR	PE22IOR	PE21IOR	PE20IOR	PE19IOR	PE18IOR	PE17IOR	PE16IOR	
H'FFFF 1246	PEIORL	PE15IOR	PE14IOR	PE13IOR	PE12IOR	—	—	—	—	
H'FFFF 1247		—	—	—	—	—	—	—	—	

Address	Register Name	Bit Names								Module
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FFFF 1248	PECRH1	—	—	—	—	—	—	—	—	PORT
H'FFFF 1249		—	—	—	—	—	—	—	—	
H'FFFF 124A	PECRH2	PE23MD1	PE23MD0	PE22MD1	PE22MD0	PE21MD1	PE21MD0	PE20MD1	PE20MD0	
H'FFFF 124B		PE19MD1	PE19MD0	PE18MD1	PE18MD0	PE17MD1	PE17MD0	PE16MD1	PE16MD0	
H'FFFF 124C	PECRL	—	PE15MD	—	PE14MD	—	PE13MD	—	PE12MD	
H'FFFF 124D		—	—	—	—	—	—	—	—	
H'FFFF 124E	—	—	—	—	—	—	—	—	—	
H'FFFF 124F	—	—	—	—	—	—	—	—	—	
H'FFFF 1250	FCR	—	—	—	—	—	—	—	—	
H'FFFF 1251		—	—	—	—	—	SCIMD	IRQMD1	IRQMD0	
H'FFFF 1252 to H'FFFF 1261	—	—	—	—	—	—	—	—	—	
H'FFFF 1262	PFDRL	—	—	—	—	—	—	—	—	
H'FFFF 1263		PF7DR	PF6DR	PF5DR	—	PF3DR	PF2DR	PF1DR	—	
H'FFFF 1264	—	—	—	—	—	—	—	—	—	
H'FFFF 1265	—	—	—	—	—	—	—	—	—	
H'FFFF 1266	PFIORL	—	—	—	—	—	—	—	—	
H'FFFF 1267		PF7IOR	PF6IOR	PF5IOR	—	PF3IOR	PF2IOR	PF1IOR	—	
H'FFFF 1268 to H'FFFF 126B	—	—	—	—	—	—	—	—	—	
H'FFFF 126C	PFCRL1	—	—	—	—	—	—	—	—	
H'FFFF 126D		—	—	—	—	—	—	—	—	
H'FFFF 126E	PFCRL2	PF7MD1	PF7MD0	PF6MD1	PF6MD0	PF5MD1	PF5MD0	—	—	
H'FFFF 126F		PF3MD1	PF3MD0	PF2MD1	PF2MD0	PF1MD1	PF1MD0	—	—	
H'FFFF 1270	PGDR	PG31DR	PG30DR	PG29DR	—	—	—	—	—	
H'FFFF 1271		—	—	—	—	—	—	—	—	
H'FFFF 1272	—	—	—	—	—	—	—	—	—	
H'FFFF 1273	—	—	—	—	—	—	—	—	—	
H'FFFF 1274	PGIOR	PG31IOR	PG30IOR	PG29IOR	—	—	—	—	—	
H'FFFF 1275		—	—	—	—	—	—	—	—	
H'FFFF 1276	—	—	—	—	—	—	—	—	—	
H'FFFF 1277	—	—	—	—	—	—	—	—	—	
H'FFFF 1278	PGCRH1	—	PG31MD	—	PG30MD	—	PG29MD	—	—	
H'FFFF 1279		—	—	—	—	—	—	—	—	
H'FFFF 127A	PGCRH2	—	—	—	—	—	—	—	—	
H'FFFF 127B		—	—	—	—	—	—	—	—	

Address	Register Name	Bit Names								Module
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FFFF 127C to H'FFFF 1281	—	—	—	—	—	—	—	—	—	PORT
H'FFFF 1282	PHDR	—	—	—	—	—	—	PH1DR	PH0DR	
H'FFFF 1283 to H'FFFF 1285	—	—	—	—	—	—	—	—	—	
H'FFFF 1286	PHIOR	—	—	—	—	—	—	—	—	
H'FFFF 1287		—	—	—	—	—	—	PH1IOR	PH0IOR	
H'FFFF 1288 to H'FFFF 128D	—	—	—	—	—	—	—	—	—	
H'FFFF 128E	PHCR	—	—	—	—	—	—	—	—	
H'FFFF 128F		—	—	—	—	—	PH1MD	—	PH0MD	
H'FFFF 1290	PIDR	PI7DR	PI6DR	PI5DR	PI4DR	PI3DR	PI2DR	PI1DR	PI0DR	
H'FFFF 1291 to H'FFFF 12FF	—	—	—	—	—	—	—	—	—	

Notes: 1. Write address is H'FFFF1000; read address is H'FFFF1001.

2. Write address is H'FFFF1002.

3. Read address is H'FFFF1003.

Appendix B Pin States

B.1 Pin States in Reset, Power-Down State, and Bus-Released State

Table B.1 Pin States in Reset, Power-Down State, and Bus-Released State

Pin Function		Pin State						
			Power-Down State					
Type	Pin Name	Reset State	Software Standby	Hardware Standby	Sleep	Bus-Released State	Software Standby in Bus-Released State	Hardware Standby in Bus-Released State
Clock	CKIO	I/O/Z*1 *2	I/L/Z*1 *2	I/L/Z*1 *2	I/O/Z*1 *2	I/O/Z*1 *2	I/L/Z*1 *2	I/L/Z*1 *2
	EXTAL	I*1	I*1	I*1	I*1	I*1	I*1	I*1
	XTAL	O*1	O*1	O*1	O*1	O*1	O*1	O*1
	CK	O/Z*1 *2	L/Z*1 *2	L/Z*1 *2	O/Z*1 *2	O/Z*1 *2	L/Z*1 *2	L/Z*1 *2
	PLLCAP1–PLLCAP2	I	I	I	I	I	I	I
System control	$\overline{\text{RES}}$	I	I	I	I	I	I	I
	$\overline{\text{WDTOVF}}$	H	H	Z	O	O	H	Z
	$\overline{\text{BREQ}}$	Z	Z	Z	I	I	Z	Z
	$\overline{\text{BACK}}$	Z	Z	Z	H	L	H	Z
	$\overline{\text{HSTBY}}$	I	I	I	I	I	I	I
Operating mode control	MD0–MD5	I	I	I	I	I	I	I
	FWE	I	I	I	I	I	I	I
Interrupt	NMI	I	I	I	I	I	I	I
	$\overline{\text{IRQ0}}$ to $\overline{\text{IRQ7}}$ of PE23 to PE21 and PE19 to PE17	Z	Z	Z	I	I	Z	Z
	$\overline{\text{IRQ0}}$ to $\overline{\text{IRQ7}}$ other than above	Z	I	Z	I	I	I	Z
	$\overline{\text{IRQOUT}}$	Z	H*4	Z	O	O	H*4	Z

Pin Function		Pin State						
			Power-Down State					
Type	Pin Name	Reset State	Software Standby	Hardware Standby	Sleep	Bus-Released State	Software Standby in Bus-Released State	Hardware Standby in Bus-Released State
Address bus	A0–25	O	Z	Z	O	Z	Z	Z
Data bus	D0–31	Z	Z	Z	I/O	Z	Z	Z
Bus control	$\overline{\text{BS}}$	H*5	Z	Z	O	Z	Z	Z
	$\overline{\text{CS0}}$	H*5	Z	Z	O	Z	Z	Z
	$\overline{\text{CS1}}$–$\overline{\text{CS5}}$	Z	Z	Z	O	Z	Z	Z
	$\overline{\text{RD}}$	H*5	Z	Z	O	Z	Z	Z
	RDWR	Z	Z	Z	O	Z	Z	Z
	$\overline{\text{WRLL}}$–$\overline{\text{HH}}$/ $\overline{\text{LLBS}}$–$\overline{\text{HHBS}}$	H*5	Z	Z	O	Z	Z	Z
	$\overline{\text{WAIT}}$	Z	Z	Z	I	Z	Z	Z
	$\overline{\text{WR}}$	Z	Z	Z	O	Z	Z	Z
	$\overline{\text{RAS0}}$–$\overline{\text{RAS1}}$	Z	Z/O*3	Z	O	Z/O*3	Z/O*3	Z
	$\overline{\text{CASHH0}}$, $\overline{\text{CASHL0}}$, $\overline{\text{CASLH0}}$, $\overline{\text{CASLL0}}$, $\overline{\text{CASHH1}}$, $\overline{\text{CASHL1}}$, $\overline{\text{CASLH1}}$, $\overline{\text{CASLL1}}$	Z	Z/O*3	Z	O	Z/O*3	Z/O*3	Z
	$\overline{\text{OE0}}$–$\overline{\text{OE1}}$	Z	Z/O*3	Z	O	Z/O*3	Z/O*3	Z
	$\overline{\text{AH}}$	Z	Z	Z	O	Z	Z	Z
DMAC	$\overline{\text{DREQ0}}$–$\overline{\text{DREQ1}}$	Z	Z	Z	I	I	Z	Z
	$\overline{\text{DRAK0}}$–$\overline{\text{DRAK1}}$	Z	Z	Z	O	O	Z	Z
	$\overline{\text{DACK0}}$–$\overline{\text{DACK1}}$	Z	O*4	Z	O	O	O*4	Z
	$\overline{\text{TEND0}}$–$\overline{\text{TEND1}}$	Z	O*4	Z	O	O	O*4	Z

Pin Function		Pin State						
			Power-Down State					
Type	Pin Name	Reset State	Software Standby	Hardware Standby	Sleep	Bus-Released State	Software Standby in Bus-Released State	Hardware Standby in Bus-Released State
TPU	TCLKA–TCLKD	Z	Z	Z	I	I	Z	Z
	TIOC0A–TIOC0D	Z	K^{*4}	Z	I/O	I/O	K^{*4}	Z
	TIOC1A–TIOC1B	Z	K^{*4}	Z	I/O	I/O	K^{*4}	Z
	TIOC2A–TIOC2B	Z	K^{*4}	Z	I/O	I/O	K^{*4}	Z
	TIOC3A–TIOC3D	Z	K^{*4}	Z	I/O	I/O	K^{*4}	Z
	TIOC4A–TIOC4B	Z	K^{*4}	Z	I/O	I/O	K^{*4}	Z
	TIOC5A–TIOC5B	Z	K^{*4}	Z	I/O	I/O	K^{*4}	Z
MMT	PCIO	Z	K^{*4}	Z	I/O	I/O	K^{*4}	Z
	PUOA(PD20)–PUOB(PD24)	Z	O^{*4}	Z	O	O	O^{*4}	Z
	PVOA(PD21)–PVOB(PD25)	Z	O^{*4}	Z	O	O	O^{*4}	Z
	PWOA(PD22)–PWOB(PD26)	Z	O^{*4}	Z	O	O	O^{*4}	Z
	PUOA(PE17)–PUOB(PE21)	Z	Z^{*6}	Z	O	O	Z^{*6}	Z
	PVOA(PE18)–PVOB(PE22)	Z	Z^{*6}	Z	O	O	Z^{*6}	Z
	PWOA(PE19)–PWOB(PE23)	Z	Z^{*6}	Z	O	O	Z^{*6}	Z
	$\overline{\text{POE0}}$–$\overline{\text{POE3}}$	Z	Z	Z	I	I	Z	Z
SCI	TxD0–TxD2	Z	O^{*4}	Z	O	O	O^{*4}	Z
	RxD0–RxD2	Z	Z	Z	I	I	Z	Z
	SCK0–SCK2	Z	Z	Z	I/O	I/O	Z	Z
A/D converter	AN0–AN7	Z	Z	Z	I	I	Z	Z
	$\overline{\text{ADTRG}}$	Z	Z	Z	I	I	Z	Z
D/A converter	DA0–DA1	Z	Z	Z	O	O	Z	Z

Pin Function		Pin State						
			Power-Down State			Bus-Released State	Software Standby in Bus-Released State	Hardware Standby in Bus-Released State
Type	Pin Name	Reset State	Software Standby	Hardware Standby	Sleep			
I/O ports	PAn	Z	K[*4]	Z	K	I/O	K[*4]	Z
	PBn	Z	K[*4]	Z	K	I/O	K[*4]	Z
	PCn	Z	K[*4]	Z	K	I/O	K[*4]	Z
	PDn	Z	K[*4]	Z	K	I/O	K[*4]	Z
	PE23–PE21, PE19–PE17	Z	Z	Z	K	I/O	Z	Z
	PEn other than above	Z	K[*4]	Z	K	I/O	K[*4]	Z
	PFn	Z	K[*4]	Z	K	I/O	K[*4]	Z
	PGn	Z	K[*4]	Z	K	I/O	K[*4]	Z
	PHn	Z	K[*4]	Z	K	I/O	K[*4]	Z
	PIn	Z	Z	Z	I	I	Z	Z

Legend:

I:	Input
O:	Output
H:	High-level output
L:	Low-level output
Z:	High-impedance state
K:	Input pins are in the high-impedance state; output pins maintain their previous state.

Notes:
1. Depends on the clock mode.
2. Z or O depending on frequency control register (FRQCR) setting.
3. Z or O depending on bus control register (BCR) setting.
4. Z or O depending on standby control register (SBYCR) setting.
5. Z in on-chip ROM enabled modes and single-chip mode.
6. Z for all pins when PUOA, PVOA, PWOA, PUOB, PVOB, and PWOB are multiplexed.

B.2 Bus-Related Signal Pin States

Table B.2 Bus-Related Signal Pin States

				On-Chip Supporting Modules			
					16-Bit Space		
Pin		On-Chip ROM Space	On-Chip RAM Space	8-Bit Space	Upper Byte	Lower Byte	Word/ Longword
$\overline{\text{CS0}}$–$\overline{\text{CS5}}$		H	H	H	H	H	H
$\overline{\text{RAS0}}$, $\overline{\text{RAS1}}$*1		H	H	H	H	H	H
$\overline{\text{CASHH0}}$, $\overline{\text{CASHH1}}$*2		H	H	H	H	H	H
$\overline{\text{CASHL0}}$, $\overline{\text{CASHL1}}$*2		H	H	H	H	H	H
$\overline{\text{CASLH0}}$, $\overline{\text{CASLH1}}$*2		H	H	H	H	H	H
$\overline{\text{CASLL0}}$, $\overline{\text{CASLL1}}$*2		H	H	H	H	H	H
$\overline{\text{OE0}}$, $\overline{\text{OE1}}$		H	H	H	H	H	H
RDWR	R	H	H	H	H	H	H
	W	H	H	H	H	H	H
$\overline{\text{AH}}$		L	L	L	L	L	L
$\overline{\text{BS}}$		H	H	H	H	H	H
$\overline{\text{RD}}$	R	H	H	H	H	H	H
	W	—	H	H	H	H	H
$\overline{\text{WR}}$	R	H	H	H	H	H	H
	W	—	H	H	H	H	H
$\overline{\text{WRHH}}$	R	H	H	H	H	H	H
	W	—	H	H	H	H	H
$\overline{\text{WRHL}}$	R	H	H	H	H	H	H
	W	—	H	H	H	H	H
$\overline{\text{WRLH}}$	R	H	H	H	H	H	H
	W	—	H	H	H	H	H
$\overline{\text{WRLL}}$	R	H	H	H	H	H	H
	W	—	H	H	H	H	H
$\overline{\text{HHBS}}$	R	H	H	H	H	H	H
	W	—	H	H	H	H	H
$\overline{\text{HLBS}}$	R	H	H	H	H	H	H
	W	—	H	H	H	H	H
$\overline{\text{LHBS}}$	R	H	H	H	H	H	H
	W	—	H	H	H	H	H
$\overline{\text{LLBS}}$	R	H	H	H	H	H	H
	W	—	H	H	H	H	H
A25–A0		Address*3	Address*3	Address*3	Address*3	Address*3	Address*3
D31–D24		Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z
D23–D16		Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z
D15–D8		Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z
D7–D0		Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z

		External Normal Space					
			16-Bit Space				
			Big-Endian		Little-Endian		Word/
Pin		**8-Bit Space**	**Upper Byte**	**Lower Byte**	**Upper Byte**	**Lower Byte**	**Longword**
$\overline{\text{CS0}}$–$\overline{\text{CS3}}$		Enabled[*4]	Enabled[*4]	Enabled[*4]	Enabled[*4]	Enabled[*4]	Enabled[*4]
$\overline{\text{CS4}}$, $\overline{\text{CS5}}$		H	H	H	H	H	H
$\overline{\text{RAS0}}$, $\overline{\text{RAS1}}$[*1]		H	H	H	H	H	H
$\overline{\text{CASHH0}}$, $\overline{\text{CASHH1}}$[*2]		H	H	H	H	H	H
$\overline{\text{CASHL0}}$, $\overline{\text{CASHL1}}$[*2]		H	H	H	H	H	H
$\overline{\text{CASLH0}}$, $\overline{\text{CASLH1}}$[*2]		H	H	H	H	H	H
$\overline{\text{CASLL0}}$, $\overline{\text{CASLL1}}$[*2]		H	H	H	H	H	H
$\overline{\text{OE0}}$, $\overline{\text{OE1}}$		H	H	H	H	H	H
RDWR	R	H	H	H	H	H	H
	W	H	H	H	H	H	H
$\overline{\text{AH}}$		L	L	L	L	L	L
$\overline{\text{BS}}$		Enabled[*5]	Enabled[*5]	Enabled[*5]	Enabled[*5]	Enabled[*5]	Enabled[*5]
$\overline{\text{RD}}$	R	L	L	L	L	L	L
	W	H	H	H	H	H	H
$\overline{\text{WR}}$	R	H	H	H	H	H	H
	W	L	L	L	L	L	L
$\overline{\text{WRHH}}$	R	H	H	H	H	H	H
	W	H	H	H	H	H	H
$\overline{\text{WRHL}}$	R	H	H	H	H	H	H
	W	H	H	H	H	H	H
$\overline{\text{WRLH}}$	R	H	H	H	H	H	H
	W	H	L	H	H	L	L
$\overline{\text{WRLL}}$	R	H	H	H	H	H	H
	W	L	H	L	L	H	L
$\overline{\text{HHBS}}$	R	H	H	H	H	H	H
	W	H	H	H	H	H	H
$\overline{\text{HLBS}}$	R	H	H	H	H	H	H
	W	H	H	H	H	H	H
$\overline{\text{LHBS}}$	R	H	L	H	H	L	L
	W	H	L	H	H	L	L
$\overline{\text{LLBS}}$	R	L	H	L	L	H	L
	W	L	H	L	L	H	L
A25–A0		Address	Address	Address	Address	Address	Address
D31–D24		Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z
D23–D16		Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z
D15–D8		Hi-Z	Data	Hi-Z	Hi-Z	Data	Data
D7–D0		Data	Hi-Z	Data	Data	Hi-Z	Data

		External Normal Space							
		32-Bit Space							
		Big-Endian				Little-Endian			
Pin		Most Significant Byte	Byte 1	Byte 2	Least Significant Byte	Most Significant Byte	Byte 1	Byte 2	Least Significant Byte
CS0–CS3		Enabled*4	Enabled*4	Enabled*4	Enabled*4	Enabled*4	Enabled*4	Enabled*4	Enabled*4
CS4, CS5		H	H	H	H	H	H	H	H
RAS0, RAS1*1		H	H	H	H	H	H	H	H
CASHH0, CASHH1*2		H	H	H	H	H	H	H	H
CASHL0, CASHL1*2		H	H	H	H	H	H	H	H
CASLH0, CASLH1*2		H	H	H	H	H	H	H	H
CASLL0, CASLL1*2		H	H	H	H	H	H	H	H
OE0, OE1		H	H	H	H	H	H	H	H
RDWR	R	H	H	H	H	H	H	H	H
	W	H	H	H	H	H	H	H	H
AH		L	L	L	L	L	L	L	L
BS		Enabled*5	Enabled*5	Enabled*5	Enabled*5	Enabled*5	Enabled*5	Enabled*5	Enabled*5
RD	R	L	L	L	L	L	L	L	L
	W	H	H	H	H	H	H	H	H
WR	R	H	H	H	H	H	H	H	H
	W	L	L	L	L	L	L	L	L
WRHH	R	H	H	H	H	H	H	H	H
	W	L	H	H	H	H	H	H	L
WRHL	R	H	H	H	H	H	H	H	H
	W	H	L	H	H	H	H	L	H
WRLH	R	H	H	H	H	H	H	H	H
	W	H	H	L	H	H	L	H	H
WRLL	R	H	H	H	H	H	H	H	H
	W	H	H	H	L	L	H	H	H
HHBS	R	L	H	H	H	H	H	H	L
	W	L	H	H	H	H	H	H	L
HLBS	R	H	L	H	H	H	H	L	H
	W	H	L	H	H	H	H	L	H
LHBS	R	H	H	L	H	H	L	H	H
	W	H	H	L	H	H	L	H	H
LLBS	R	H	H	H	L	L	H	H	H
	W	H	H	H	L	L	H	H	H
A25–A0		Address	Address	Address	Address	Address	Address	Address	Address
D31–D24		Data	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Data
D23–D16		Hi-Z	Data	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Data	Hi-Z
D15–D8		Hi-Z	Hi-Z	Data	Hi-Z	Hi-Z	Data	Hi-Z	Hi-Z
D7–D0		Hi-Z	Hi-Z	Hi-Z	Data	Data	Hi-Z	Hi-Z	Hi-Z

		External Normal Space				
		32-Bit Space				
		Big-Endian		Little-Endian		
Pin		**Upper Byte**	**Lower Byte**	**Upper Byte**	**Lower Byte**	**Longword**
$\overline{\text{CS0}}$–$\overline{\text{CS3}}$		Enabled[*4]	Enabled[*4]	Enabled[*4]	Enabled[*4]	Enabled[*4]
$\overline{\text{CS4}}$, $\overline{\text{CS5}}$		H	H	H	H	H
$\overline{\text{RAS0}}$, $\overline{\text{RAS1}}$[*1]		H	H	H	H	H
$\overline{\text{CASHH0}}$, $\overline{\text{CASHH1}}$[*2]		H	H	H	H	H
$\overline{\text{CASHL0}}$, $\overline{\text{CASHL1}}$[*2]		H	H	H	H	H
$\overline{\text{CASLH0}}$, $\overline{\text{CASLH1}}$[*2]		H	H	H	H	H
$\overline{\text{CASLL0}}$, $\overline{\text{CASLL1}}$[*2]		H	H	H	H	H
$\overline{\text{OE0}}$, $\overline{\text{OE1}}$		H	H	H	H	H
RDWR	R	H	H	H	H	H
	W	H	H	H	H	H
$\overline{\text{AH}}$		L	L	L	L	L
$\overline{\text{BS}}$		Enabled[*5]	Enabled[*5]	Enabled[*5]	Enabled[*5]	Enabled[*5]
$\overline{\text{RD}}$	R	L	L	L	L	L
	W	H	H	H	H	H
$\overline{\text{WR}}$	R	H	H	H	H	H
	W	L	L	L	L	L
$\overline{\text{WRHH}}$	R	H	H	H	H	H
	W	L	H	H	L	L
$\overline{\text{WRHL}}$	R	H	H	H	H	H
	W	L	H	H	L	L
$\overline{\text{WRLH}}$	R	H	H	H	H	H
	W	H	L	L	H	L
$\overline{\text{WRLL}}$	R	H	H	H	H	H
	W	H	L	L	H	L
$\overline{\text{HHBS}}$	R	L	H	H	L	L
	W	L	H	H	L	L
$\overline{\text{HLBS}}$	R	L	H	H	L	L
	W	L	H	H	L	L
$\overline{\text{LHBS}}$	R	H	L	L	H	L
	W	H	L	L	H	L
$\overline{\text{LLBS}}$	R	H	L	L	H	L
	W	H	L	L	H	L
A25–A0		Address	Address	Address	Address	Address
D31–D24		Data	Hi-Z	Hi-Z	Data	Data
D23–D16		Data	Hi-Z	Hi-Z	Data	Data
D15–D8		Hi-Z	Data	Data	Hi-Z	Data
D7–D0		Hi-Z	Data	Data	Hi-Z	Data

		Multiplexed I/O Space					
			16-Bit Space				
			Big-Endian		Little-Endian		Word/ Longword
Pin		8-Bit Space	Upper Byte	Lower Byte	Upper Byte	Lower Byte	
$\overline{\text{CS0}}$		H	H	H	H	H	H
$\overline{\text{CS1}}$–$\overline{\text{CS3}}$		Enabled*4	Enabled*4	Enabled*4	Enabled*4	Enabled*4	Enabled*4
$\overline{\text{CS4}}$, $\overline{\text{CS5}}$		H	H	H	H	H	H
$\overline{\text{RAS0}}$, $\overline{\text{RAS1}}$*1		H	H	H	H	H	H
$\overline{\text{CASHH0}}$, $\overline{\text{CASHH1}}$*2		H	H	H	H	H	H
$\overline{\text{CASHL0}}$, $\overline{\text{CASHL1}}$*2		H	H	H	H	H	H
$\overline{\text{CASLH0}}$, $\overline{\text{CASLH1}}$*2		H	H	H	H	H	H
$\overline{\text{CASLL0}}$, $\overline{\text{CASLL1}}$*2		H	H	H	H	H	H
$\overline{\text{OE0}}$, $\overline{\text{OE1}}$		H	H	H	H	H	H
RDWR	R	H	H	H	H	H	H
	W	H	H	H	H	H	H
$\overline{\text{AH}}$		Enabled*6	Enabled*6	Enabled*6	Enabled*6	Enabled*6	Enabled*6
$\overline{\text{BS}}$		Enabled*5	Enabled*5	Enabled*5	Enabled*5	Enabled*5	Enabled*5
$\overline{\text{RD}}$	R	L	L	L	L	L	L
	W	H	H	H	H	H	H
$\overline{\text{WR}}$	R	H	H	H	H	H	H
	W	L	L	L	L	L	L
$\overline{\text{WRHH}}$	R	H	H	H	H	H	H
	W	H	H	H	H	H	H
$\overline{\text{WRHL}}$	R	H	H	H	H	H	H
	W	H	H	H	H	H	H
$\overline{\text{WRLH}}$	R	H	H	H	H	H	H
	W	H	L	H	H	L	L
$\overline{\text{WRLL}}$	R	H	H	H	H	H	H
	W	L	H	L	L	H	L
$\overline{\text{HHBS}}$	R	H	H	H	H	H	H
	W	H	H	H	H	H	H
$\overline{\text{HLBS}}$	R	H	H	H	H	H	H
	W	H	H	H	H	H	H
$\overline{\text{LHBS}}$	R	H	H	H	H	H	H
	W	H	H	H	H	H	H
$\overline{\text{LLBS}}$	R	H	H	H	H	H	H
	W	H	H	H	H	H	H
A25–A0		Address	Address	Address	Address	Address	Address
D31–D24		Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z
D23–D16		Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z
D15–D8		Hi-Z	Address/Data	Address	Address	Address/Data	Address/Data
D7–D0		Address/Data	Address	Address/Data	Address/Data	Address	Address/Data

		DRAM Space					
			16-Bit Space				
			Big-Endian		Little-Endian		Word/
Pin		8-Bit Space	Upper Byte	Lower Byte	Upper Byte	Lower Byte	Longword
CS0–CS3		H	H	H	H	H	H
CS4, CS5		Enabled*4	Enabled*4	Enabled*4	Enabled*4	Enabled*4	Enabled*4
RAS0, RAS1*1		Enabled*7	Enabled*7	Enabled*7	Enabled*7	Enabled*7	Enabled*7
CASHH0, CASHH1*2		H	H	H	H	H	H
CASHL0, CASHL1*2		H	H	H	H	H	H
CASLH0, CASLH1*2		H	Enabled*7	H	H	Enabled*7	Enabled*7
CASLL0, CASLL1*2		Enabled*7	H	Enabled*7	Enabled*7	H	Enabled*7
OE0, OE1		Enabled*8	Enabled*8	Enabled*8	Enabled*8	Enabled*8	Enabled*8
RDWR	R	H	H	H	H	H	H
	W	L	L	L	L	L	L
AH		L	L	L	L	L	L
BS		Enabled*5	Enabled*5	Enabled*5	Enabled*5	Enabled*5	Enabled*5
RD	R	H	H	H	H	H	H
	W	H	H	H	H	H	H
WR	R	H	H	H	H	H	H
	W	H	H	H	H	H	H
WRHH	R	H	H	H	H	H	H
	W	H	H	H	H	H	H
WRHL	R	H	H	H	H	H	H
	W	H	H	H	H	H	H
WRLH	R	H	H	H	H	H	H
	W	H	H	H	H	H	H
WRLL	R	H	H	H	H	H	H
	W	H	H	H	H	H	H
HHBS	R	H	H	H	H	H	H
	W	H	H	H	H	H	H
HLBS	R	H	H	H	H	H	H
	W	H	H	H	H	H	H
LHBS	R	H	H	H	H	H	H
	W	H	H	H	H	H	H
LLBS	R	H	H	H	H	H	H
	W	H	H	H	H	H	H
A25–A0		Address	Address	Address	Address	Address	Address
D31–D24		Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z
D23–D16		Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z
D15–D8		Hi-Z	Data	Hi-Z	Hi-Z	Data	Data
D7–D0		Data	Hi-Z	Data	Data	Hi-Z	Data

		DRAM Space							
		32-Bit Space							
		Big-Endian				Little-Endian			
Pin		Most Significant Byte	Byte 1	Byte 2	Least Significant Byte	Most Significant Byte	Byte 1	Byte 2	Least Significant Byte
$\overline{\text{CS0}}$–$\overline{\text{CS3}}$		H	H	H	H	H	H	H	H
$\overline{\text{CS4}}$, $\overline{\text{CS5}}$		Enabled*4	Enabled*4	Enabled*4	Enabled*4	Enabled*4	Enabled*4	Enabled*4	Enabled*4
$\overline{\text{RAS0}}$, $\overline{\text{RAS1}}$*1		Enabled*7	Enabled*7	Enabled*7	Enabled*7	Enabled*7	Enabled*7	Enabled*7	Enabled*7
$\overline{\text{CASHH0}}$, $\overline{\text{CASHH1}}$*2		Enabled*7	H	H	H	H	H	H	Enabled*7
$\overline{\text{CASHL0}}$, $\overline{\text{CASHL1}}$*2		H	Enabled*7	H	H	H	H	Enabled*7	H
$\overline{\text{CASLH0}}$, $\overline{\text{CASLH1}}$*2		H	H	Enabled*7	H	H	Enabled*7	H	H
$\overline{\text{CASLL0}}$, $\overline{\text{CASLL1}}$*2		H	H	H	Enabled*7	Enabled*7	H	H	H
$\overline{\text{OE0}}$, $\overline{\text{OE1}}$		Enabled*8	Enabled*8	Enabled*8	Enabled*8	Enabled*8	Enabled*8	Enabled*8	Enabled*8
RDWR	R	H	H	H	H	H	H	H	H
	W	L	L	L	L	L	L	L	L
$\overline{\text{AH}}$		L	L	L	L	L	L	L	L
$\overline{\text{BS}}$		Enabled*5	Enabled*5	Enabled*5	Enabled*5	Enabled*5	Enabled*5	Enabled*5	Enabled*5
$\overline{\text{RD}}$	R	H	H	H	H	H	H	H	H
	W	H	H	H	H	H	H	H	H
$\overline{\text{WR}}$	R	H	H	H	H	H	H	H	H
	W	H	H	H	H	H	H	H	H
$\overline{\text{WRHH}}$	R	H	H	H	H	H	H	H	H
	W	H	H	H	H	H	H	H	H
$\overline{\text{WRHL}}$	R	H	H	H	H	H	H	H	H
	W	H	H	H	H	H	H	H	H
$\overline{\text{WRLH}}$	R	H	H	H	H	H	H	H	H
	W	H	H	H	H	H	H	H	H
$\overline{\text{WRLL}}$	R	H	H	H	H	H	H	H	H
	W	H	H	H	H	H	H	H	H
$\overline{\text{HHBS}}$	R	H	H	H	H	H	H	H	H
	W	H	H	H	H	H	H	H	H
$\overline{\text{HLBS}}$	R	H	H	H	H	H	H	H	H
	W	H	H	H	H	H	H	H	H
$\overline{\text{LHBS}}$	R	H	H	H	H	H	H	H	H
	W	H	H	H	H	H	H	H	H
$\overline{\text{LLBS}}$	R	H	H	H	H	H	H	H	H
	W	H	H	H	H	H	H	H	H
A25–A0		Address	Address	Address	Address	Address	Address	Address	Address
D31–D24		Data	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Data
D23–D16		Hi-Z	Data	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Data	Hi-Z
D15–D8		Hi-Z	Hi-Z	Data	Hi-Z	Hi-Z	Data	Hi-Z	Hi-Z
D7–D0		Hi-Z	Hi-Z	Hi-Z	Data	Data	Hi-Z	Hi-Z	Hi-Z

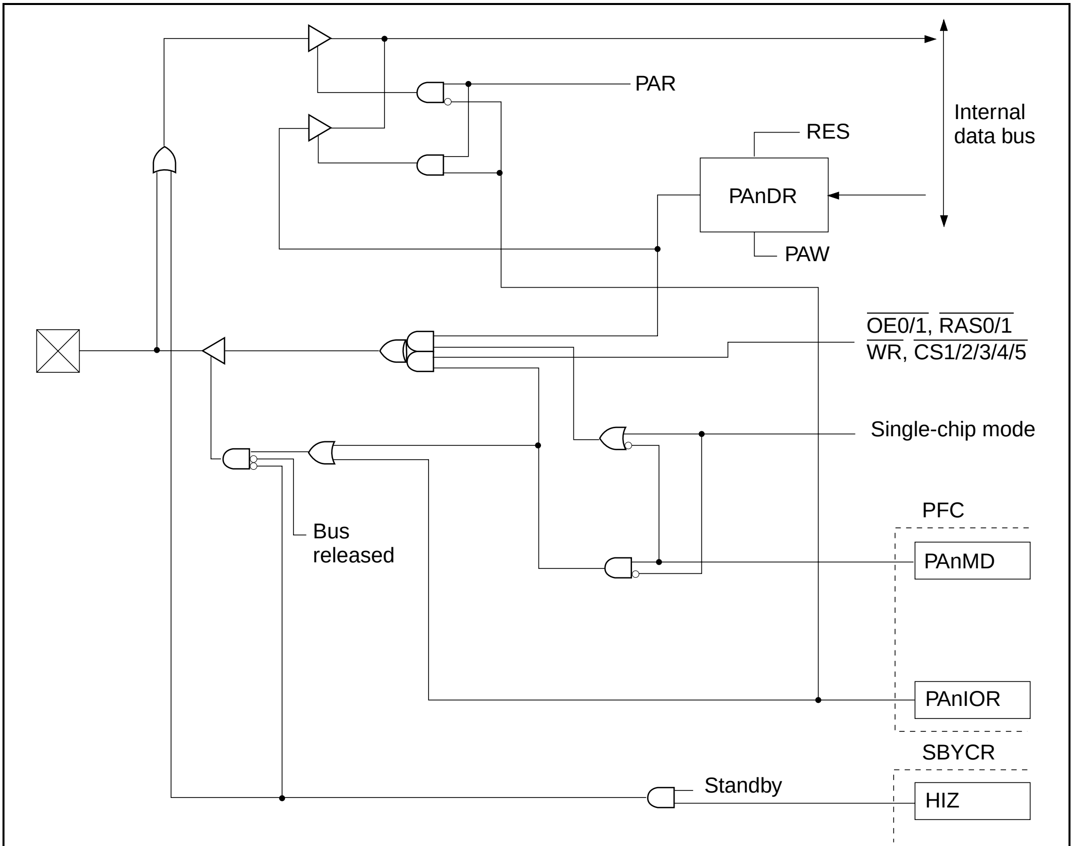
		DRAM Space				
		32-Bit Space				
		Big-Endian		Little-Endian		
Pin		Upper Byte	Lower Byte	Upper Byte	Lower Byte	Longword
$\overline{\text{CS0}}$–$\overline{\text{CS3}}$		H	H	H	H	H
$\overline{\text{CS4}}$, $\overline{\text{CS5}}$		Enabled[*4]	Enabled[*4]	Enabled[*4]	Enabled[*4]	Enabled[*4]
$\overline{\text{RAS0}}$, $\overline{\text{RAS1}}$[*1]		Enabled[*7]	Enabled[*7]	Enabled[*7]	Enabled[*7]	Enabled[*7]
$\overline{\text{CASHH0}}$, $\overline{\text{CASHH1}}$[*2]		Enabled[*7]	H	H	Enabled[*7]	Enabled[*7]
$\overline{\text{CASHL0}}$, $\overline{\text{CASHL1}}$[*2]		Enabled[*7]	H	H	Enabled[*7]	Enabled[*7]
$\overline{\text{CASLH0}}$, $\overline{\text{CASLH1}}$[*2]		H	Enabled[*7]	Enabled[*7]	H	Enabled[*7]
$\overline{\text{CASLL0}}$, $\overline{\text{CASLL1}}$[*2]		H	Enabled[*7]	Enabled[*7]	H	Enabled[*7]
$\overline{\text{OE0}}$, $\overline{\text{OE1}}$		Enabled[*8]	Enabled[*8]	Enabled[*8]	Enabled[*8]	Enabled[*8]
RDWR	R	H	H	H	H	H
	W	L	L	L	L	L
$\overline{\text{AH}}$		L	L	L	L	L
$\overline{\text{BS}}$		Enabled[*5]	Enabled[*5]	Enabled[*5]	Enabled[*5]	Enabled[*5]
$\overline{\text{RD}}$	R	H	H	H	H	H
	W	H	H	H	H	H
$\overline{\text{WR}}$	R	H	H	H	H	H
	W	H	H	H	H	H
$\overline{\text{WRHH}}$	R	H	H	H	H	H
	W	H	H	H	H	H
$\overline{\text{WRHL}}$	R	H	H	H	H	H
	W	H	H	H	H	H
$\overline{\text{WRLH}}$	R	H	H	H	H	H
	W	H	H	H	H	H
$\overline{\text{WRLL}}$	R	H	H	H	H	H
	W	H	H	H	H	H
$\overline{\text{HHBS}}$	R	H	H	H	H	H
	W	H	H	H	H	H
$\overline{\text{HLBS}}$	R	H	H	H	H	H
	W	H	H	H	H	H
$\overline{\text{LHBS}}$	R	H	H	H	H	H
	W	H	H	H	H	H
$\overline{\text{LLBS}}$	R	H	H	H	H	H
	W	H	H	H	H	H
A25–A0		Address	Address	Address	Address	Address
D31–D24		Data	Hi-Z	Hi-Z	Data	Data
D23–D16		Data	Hi-Z	Hi-Z	Data	Data
D15–D8		Hi-Z	Data	Data	Hi-Z	Data
D7–D0		Hi-Z	Data	Data	Hi-Z	Data

Legend:
R: Read
W: Write

Notes: 1. Asserted low in RAS down or refresh state.
2. Asserted low in refresh state.
3. Previously accessed external space address value is retained.

4. Chip select signal for accessed area is low, other chip select signals are high.
5. Asserted low in accordance with BS timing.
6. Output at high level in accordance with AH timing.
7. Signal for accessed area is asserted low, and other signals go high, at timing in accordance with DRAM access strobe waveform.
8. In EDO mode only, asserted low in accordance with OE timing.

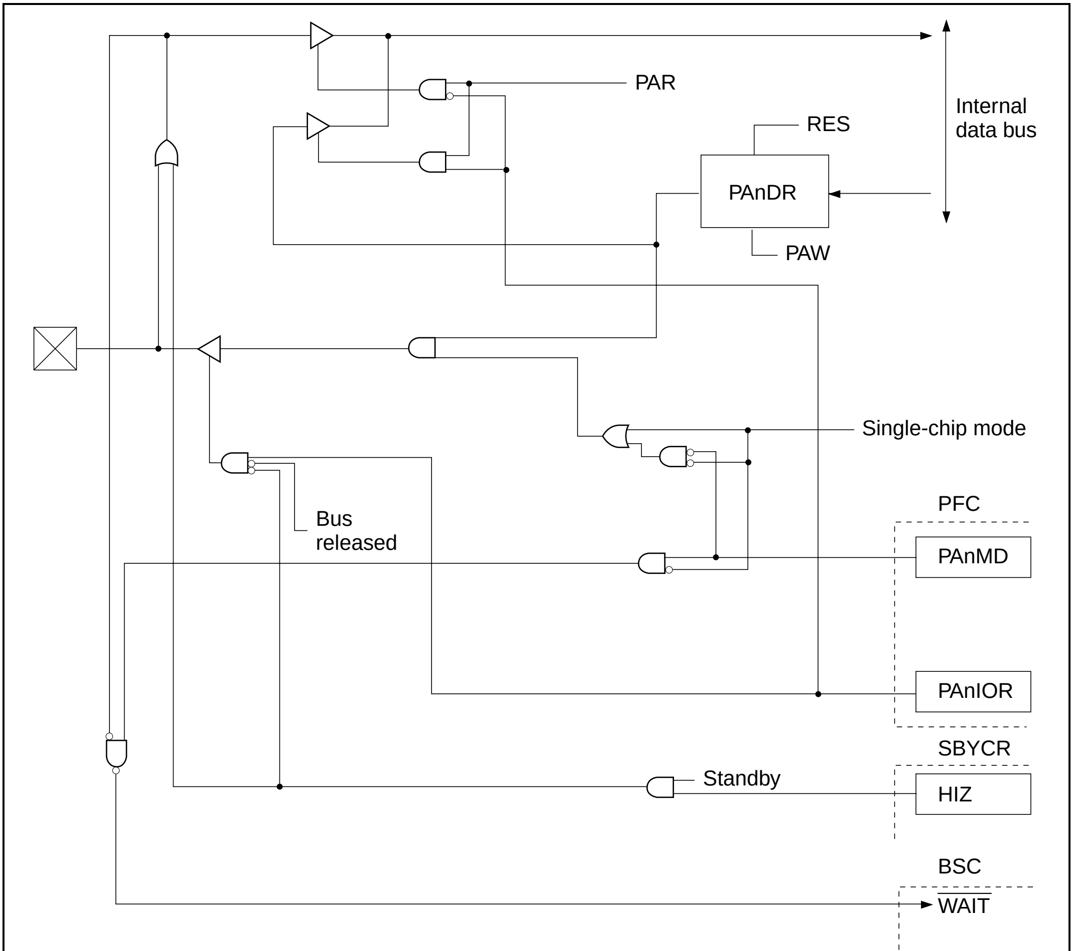
Appendix C I/O Port Block Diagrams


PAR
RES
Internal data bus
PAnDR
PAW
$\overline{\text{OE0/1}}$, $\overline{\text{RAS0/1}}$
$\overline{\text{WR}}$, $\overline{\text{CS1/2/3/4/5}}$
Single-chip mode
PFC
PAnMD
PAnIOR
SBYCR
HIZ
Standby
Bus released

Legend:
PAR: Port A read signal
PAW: Port A write signal
RES: Reset signal

Note: n = 0, 1, 8, 9, 17, or 21 to 25

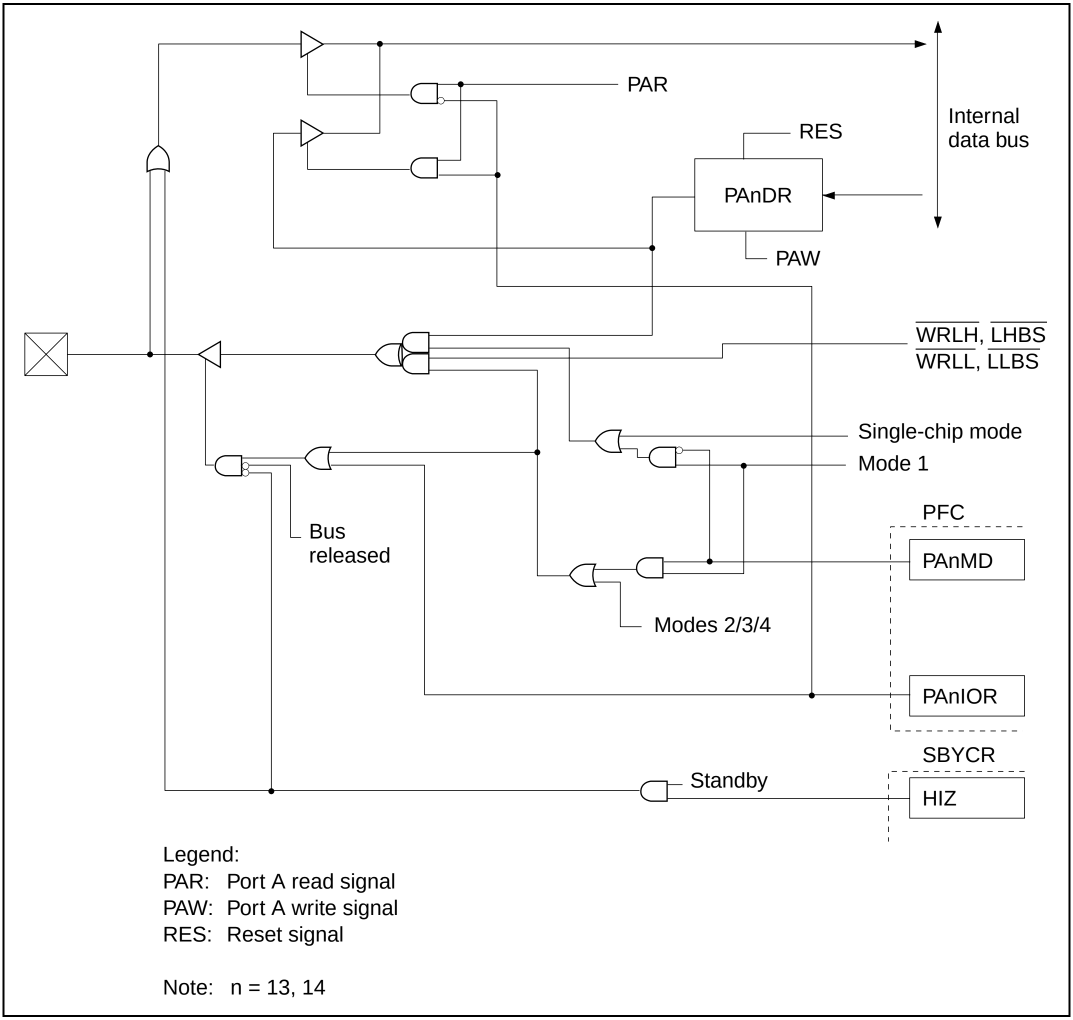
Figure C.1 PAn/XXXX Block Diagram



Legend:
PAR: Port A read signal
PAW: Port A write signal
RES: Reset signal

Note: n = 12

Figure C.2 PA12/$\overline{\text{WAIT}}$ Block Diagram



Legend:
PAR: Port A read signal
PAW: Port A write signal
RES: Reset signal

Note: n = 13, 14

Figure C.3 PA13/$\overline{\text{WRLL}}$/$\overline{\text{LLBS}}$, PA14/$\overline{\text{WRLH}}$/$\overline{\text{LHBS}}$ Block Diagram

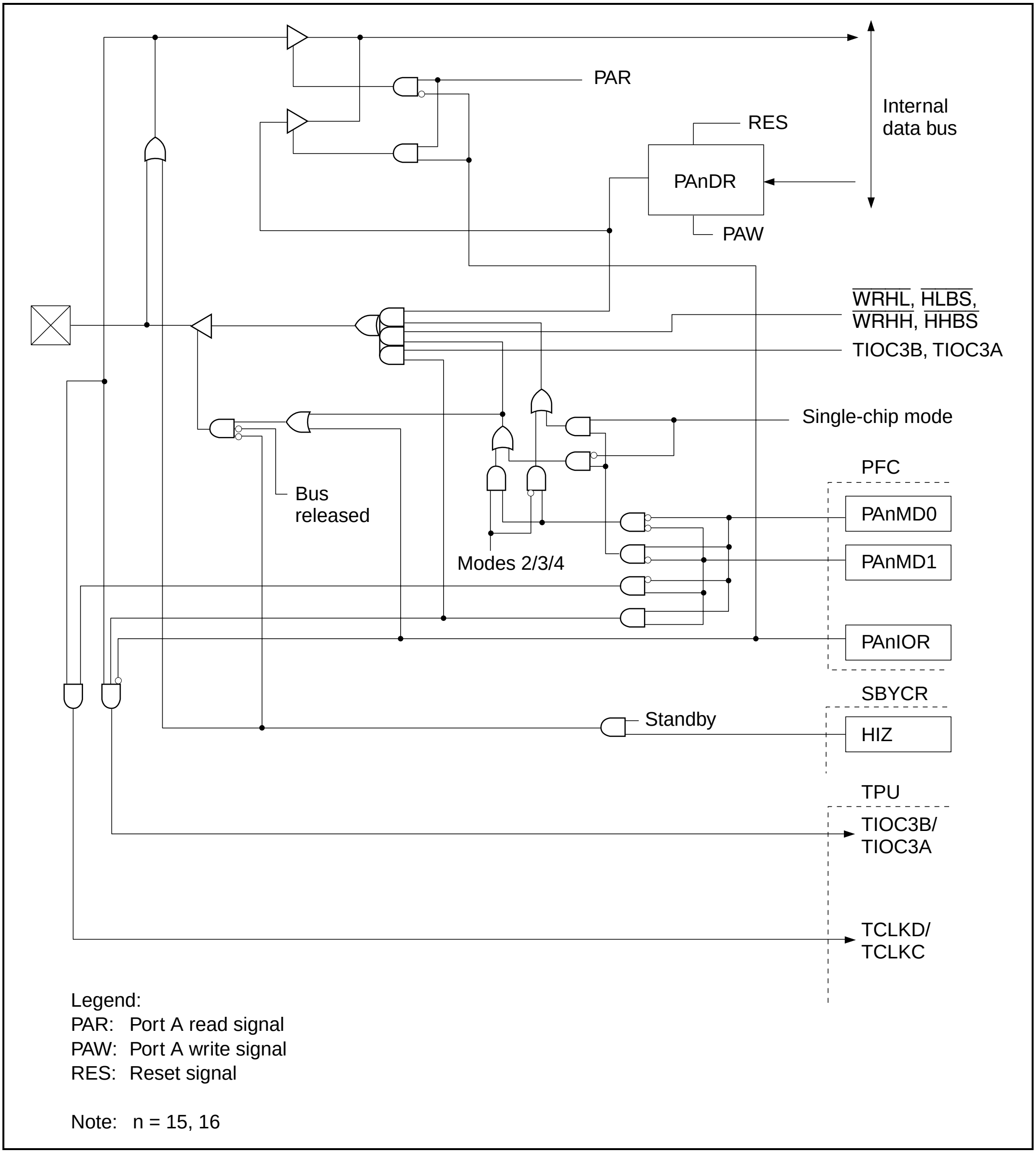
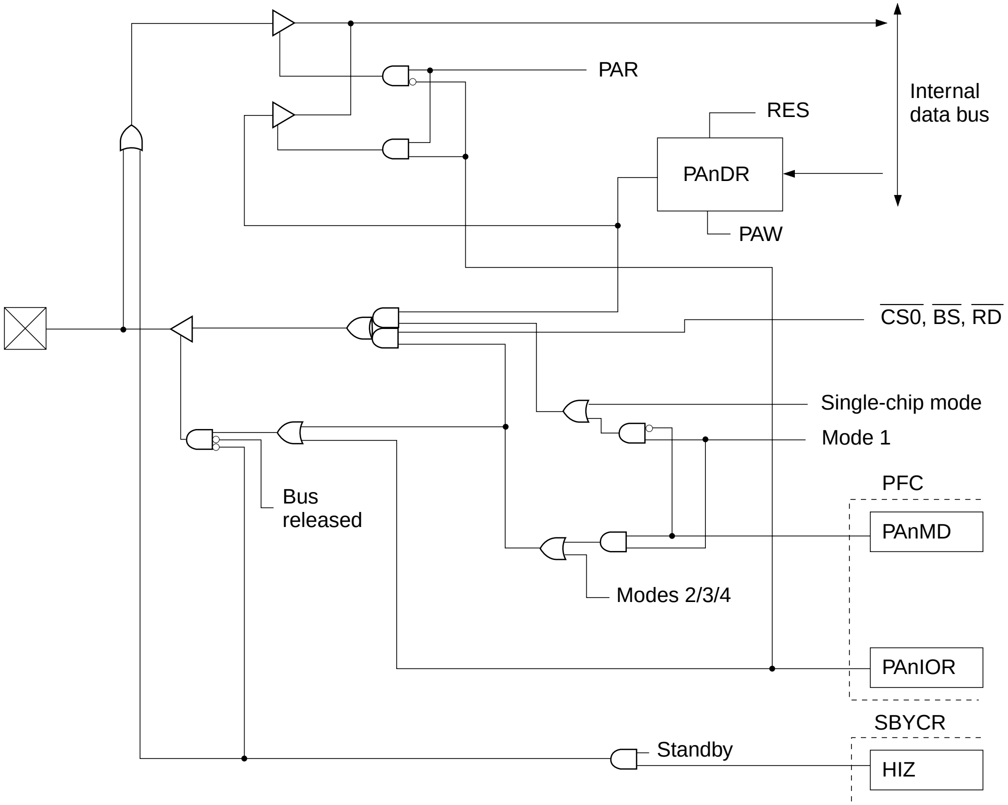


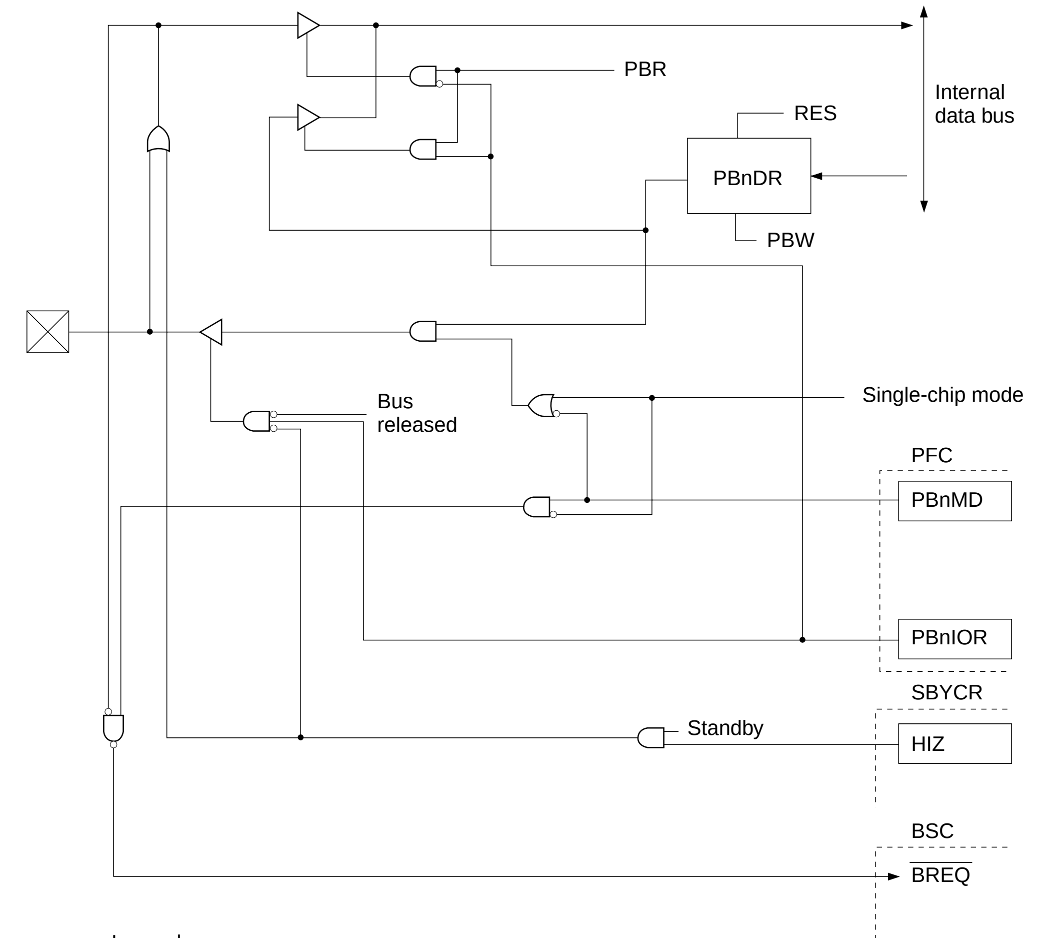
Figure C.4 PA15/$\overline{\text{WRHL}}$/$\overline{\text{HLBS}}$/TCLKD/TIOC3B, PA16/$\overline{\text{WRHH}}$/$\overline{\text{HHBS}}$/TCLKC/TIOC3A Block Diagram



Legend:
PAR: Port A read signal
PAW: Port A write signal
RES: Reset signal

Note: n = 18 to 20

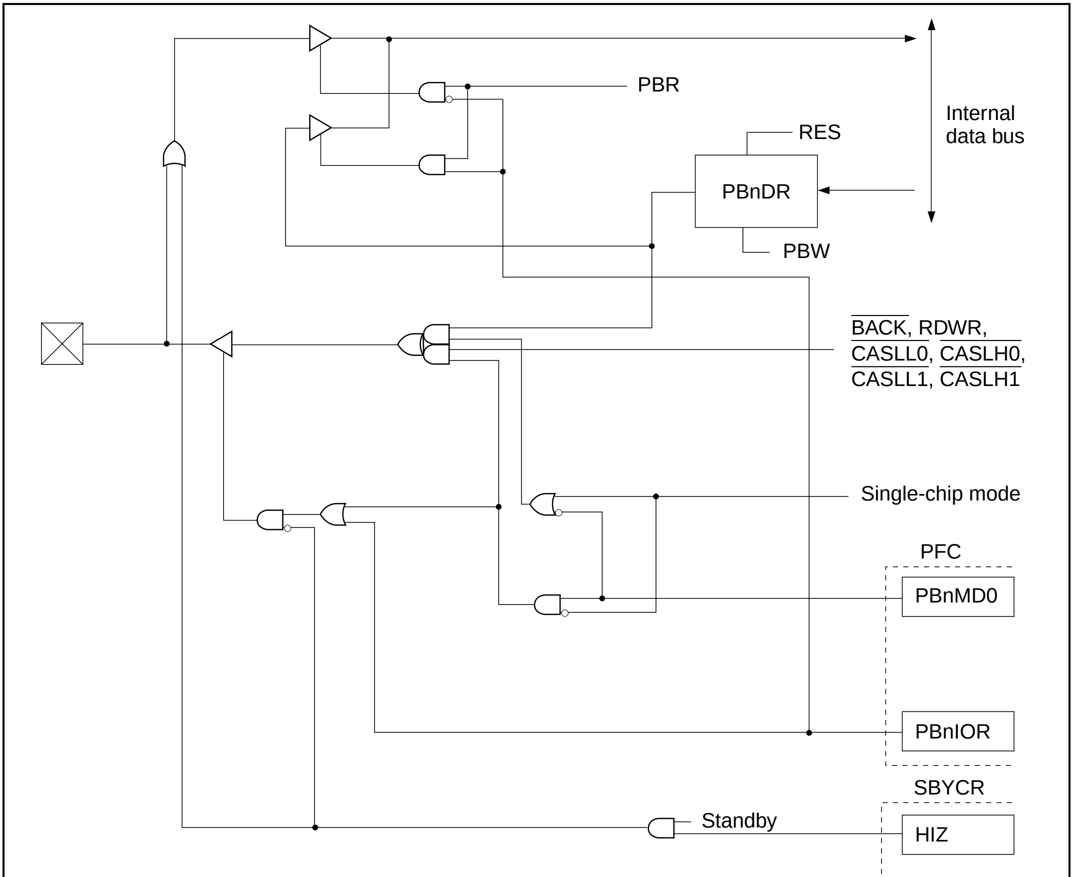
Figure C.5 PA18/$\overline{\text{RD}}$, PA19/$\overline{\text{BS}}$, PA20/$\overline{\text{CS0}}$ Block Diagram



Legend:
PBR: Port B read signal
PBW: Port B write signal
RES: Reset signal

Note: n = 6

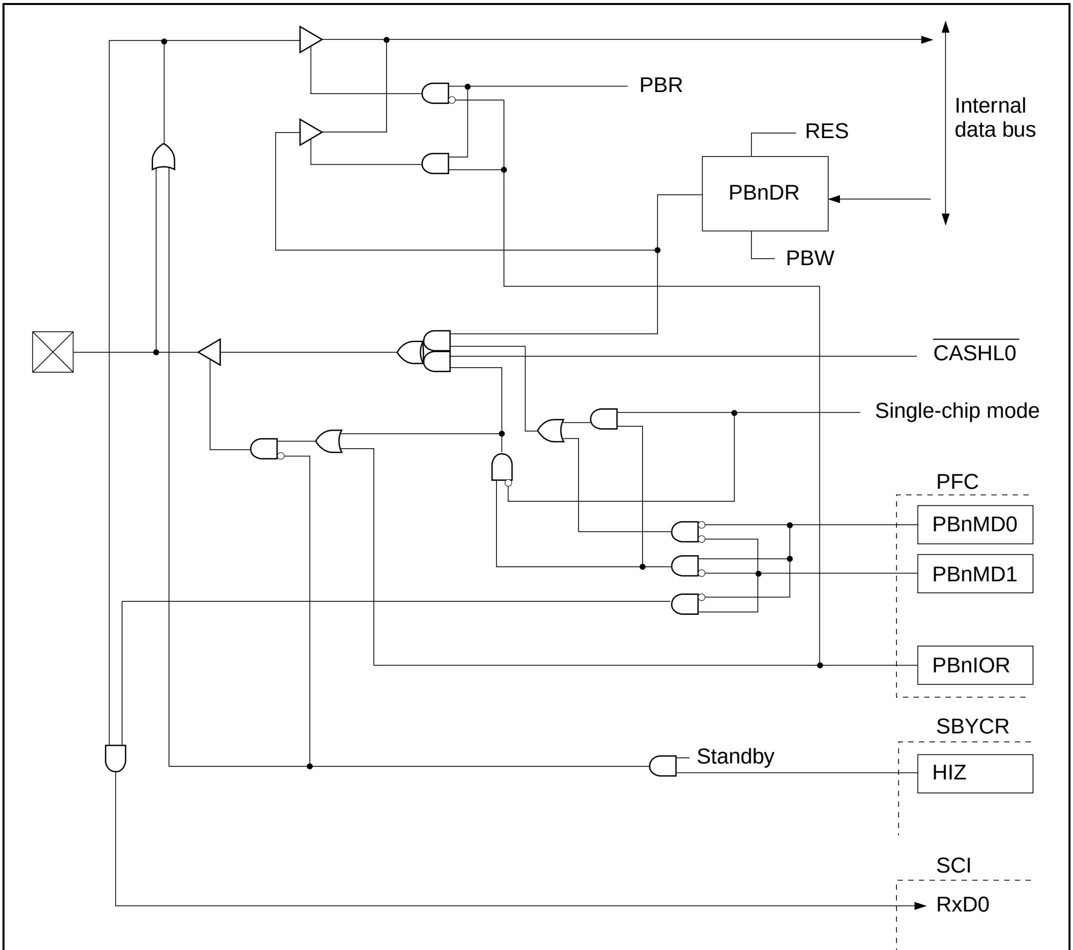
Figure C.6 PB6/$\overline{\text{BREQ}}$ Block Diagram



Legend:
PBR: Port B read signal
PBW: Port B write signal
RES: Reset signal

Note: n = 7, 13, 16, 17, 20, 21

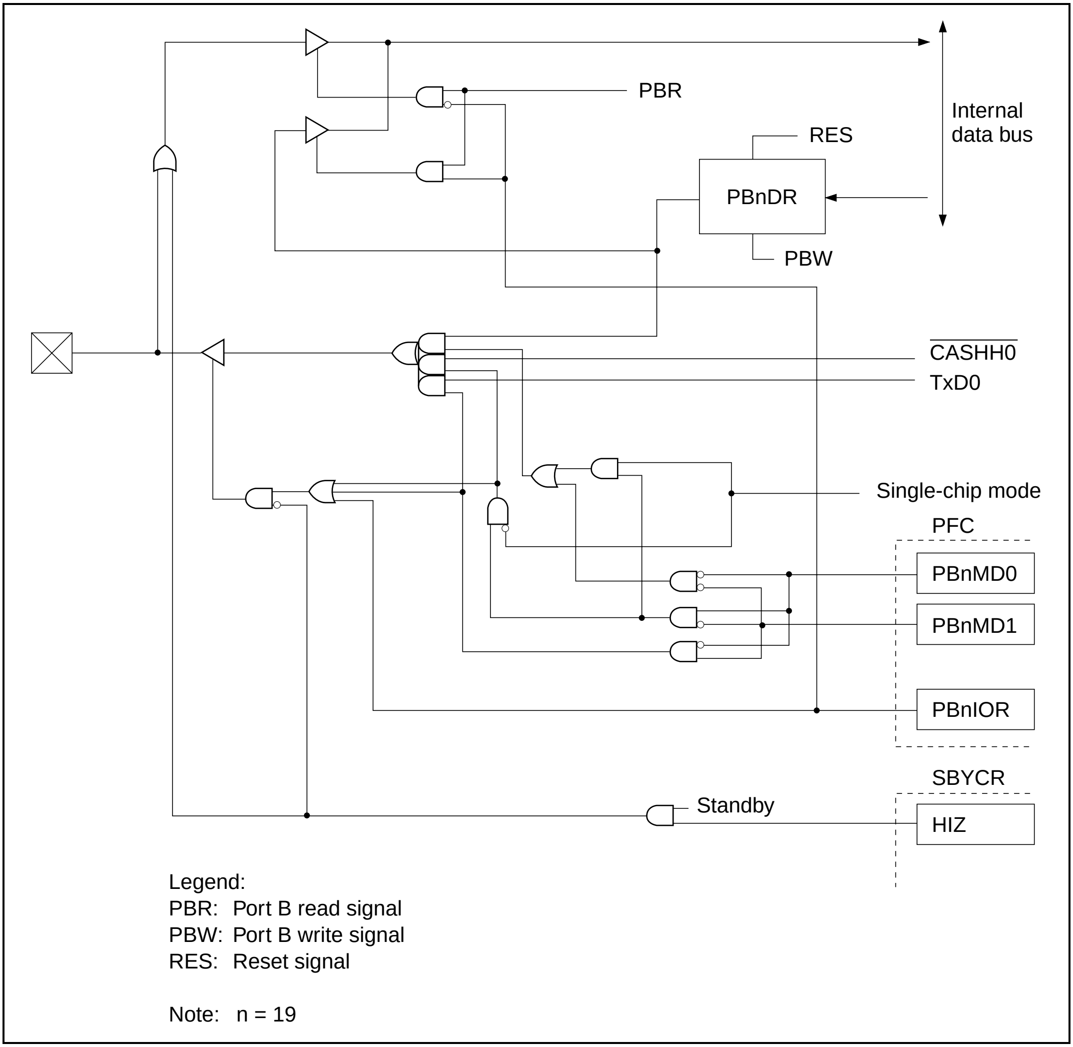
Figure C.7 PBn/XXXX Block Diagram



Legend:
PBR: Port B read signal
PBW: Port B write signal
RES: Reset signal

Note: n = 18

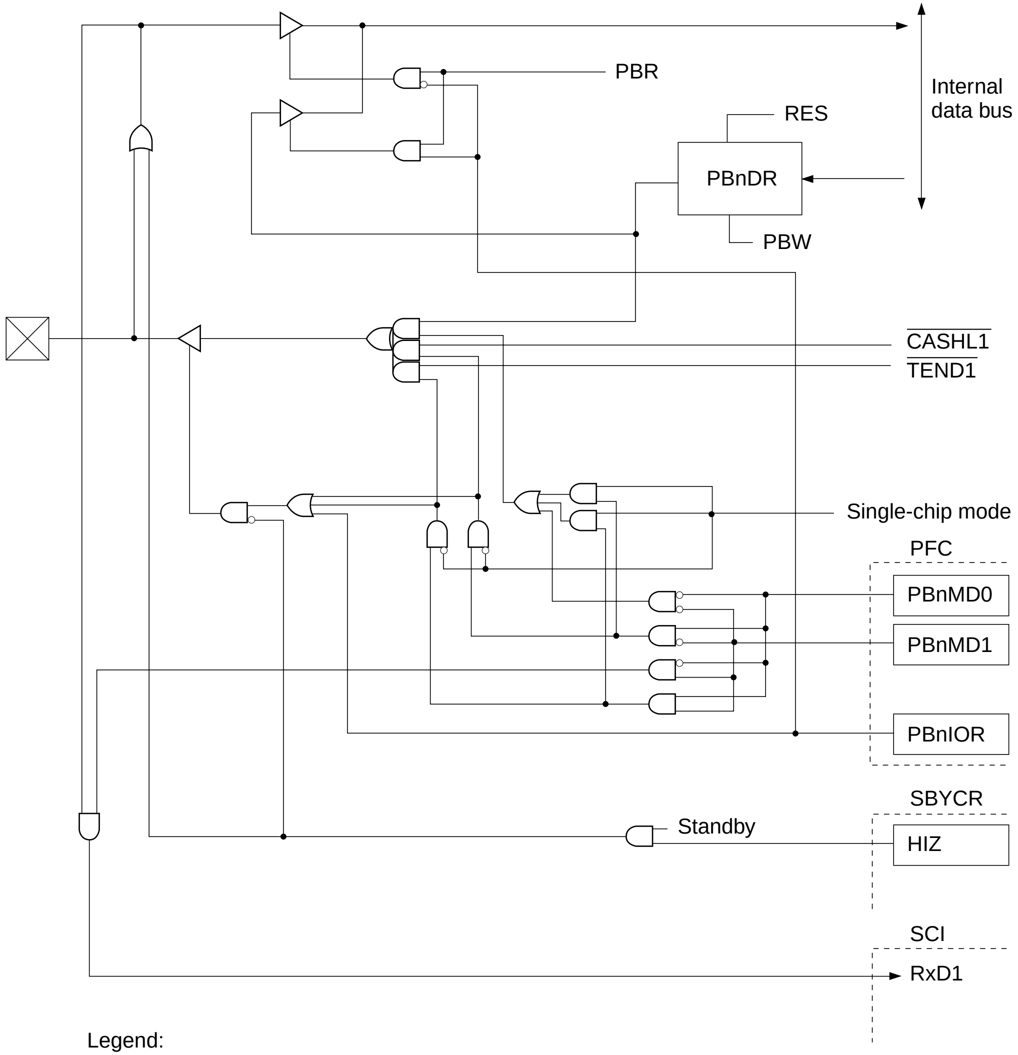
Figure C.8 PB18/$\overline{\text{CASHL0}}$/RxD0 Block Diagram



Legend:
PBR: Port B read signal
PBW: Port B write signal
RES: Reset signal

Note: n = 19

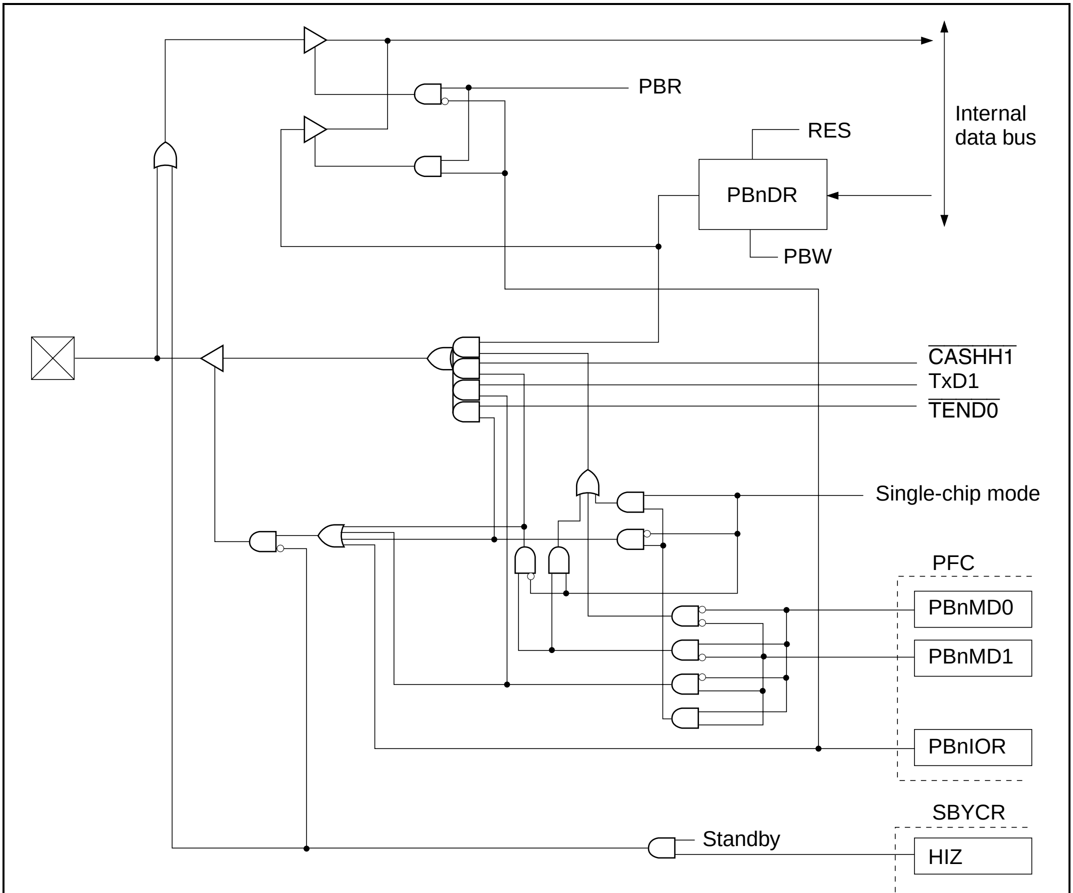
Figure C.9 PB19/$\overline{\text{CASHH0}}$/TxD0 Block Diagram



Legend:
PBR: Port B read signal
PBW: Port B write signal
RES: Reset signal

Note: n = 22

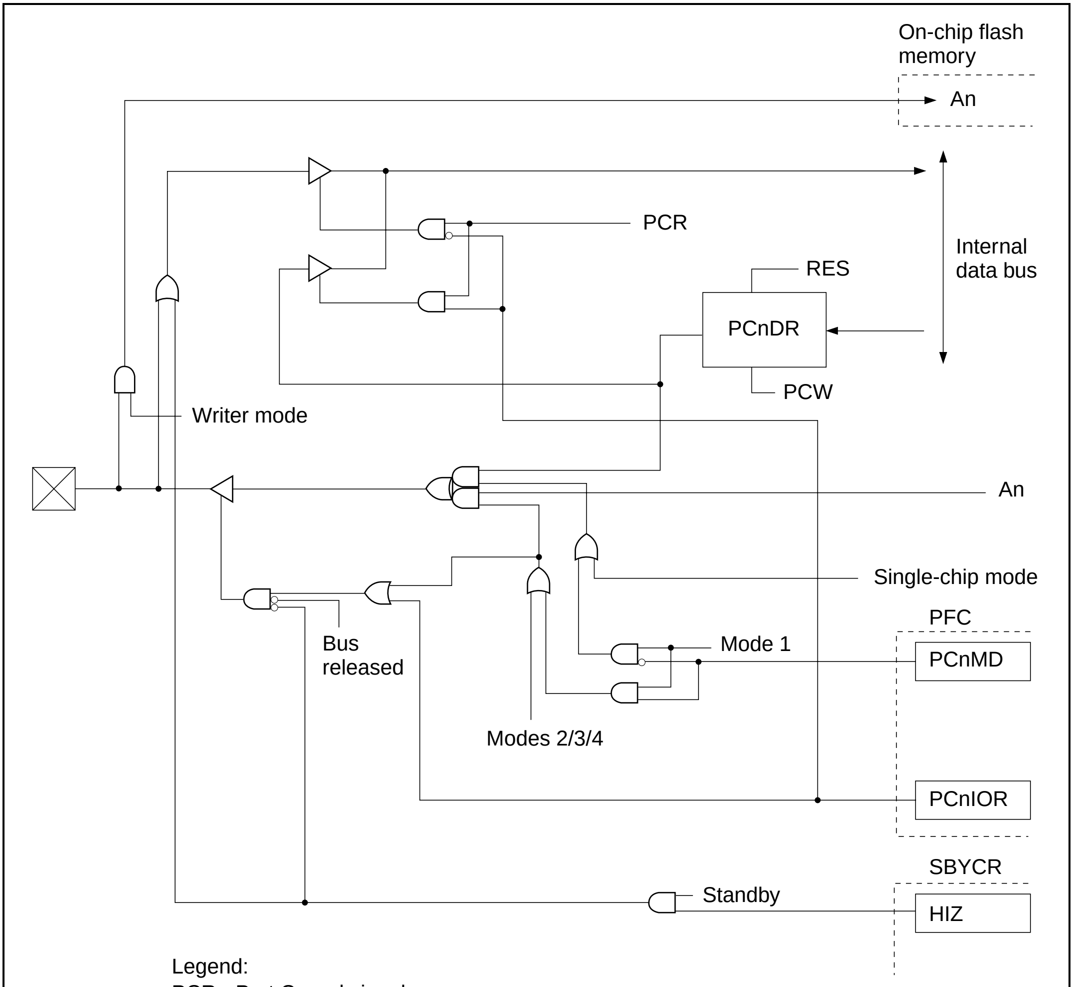
Figure C.10 PB22/$\overline{\text{CASHL1}}$/RxD1/$\overline{\text{TEND1}}$ Block Diagram



Legend:
PBR: Port B read signal
PBW: Port B write signal
RES: Reset signal

Note: n = 23

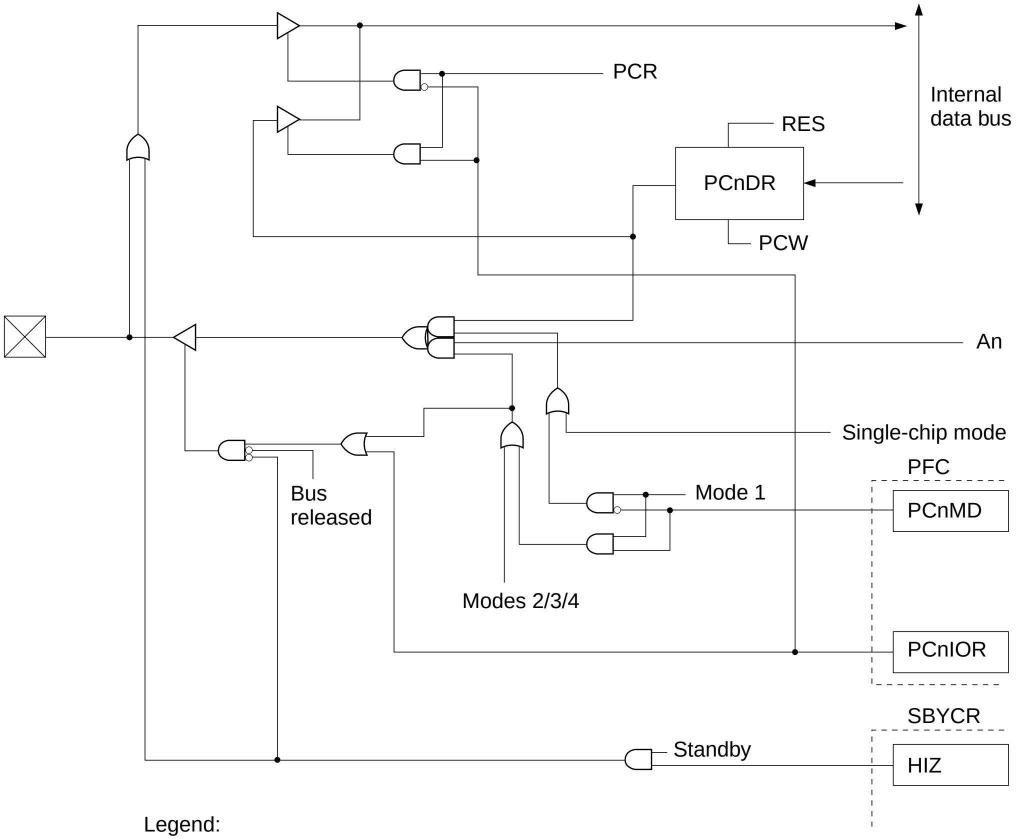
Figure C.11 PB23/$\overline{\text{CASHH1}}$/TxD1/$\overline{\text{TEND0}}$ Block Diagram



Legend:
PCR: Port C read signal
PCW: Port C write signal
RES: Reset signal

Note: n = 0 to 13

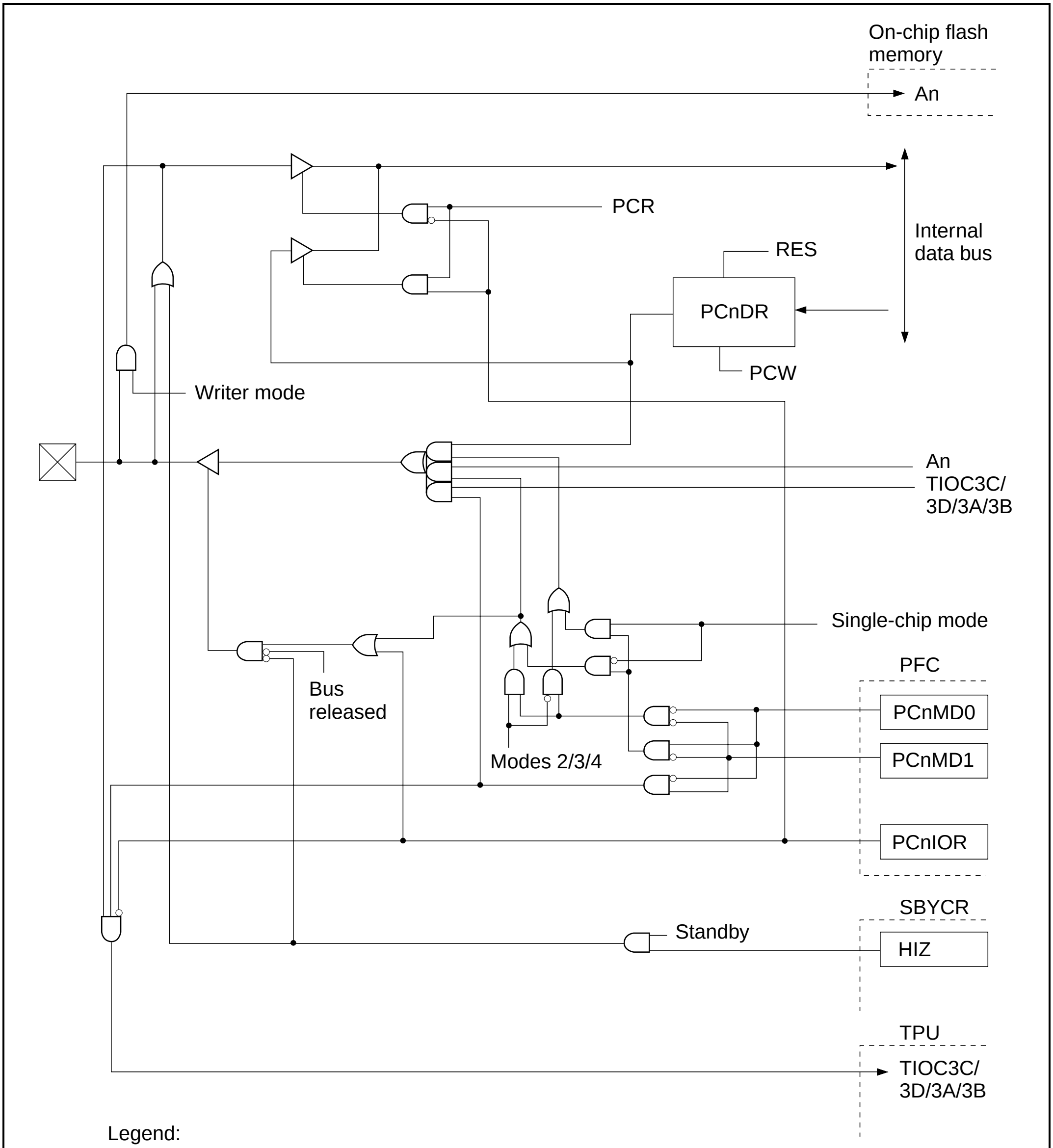
Figure C.12 PCn/An Block Diagram (for F-ZTAT version)



Legend:
PCR: Port C read signal
PCW: Port C write signal
RES: Reset signal

Note: n = 0 to 13

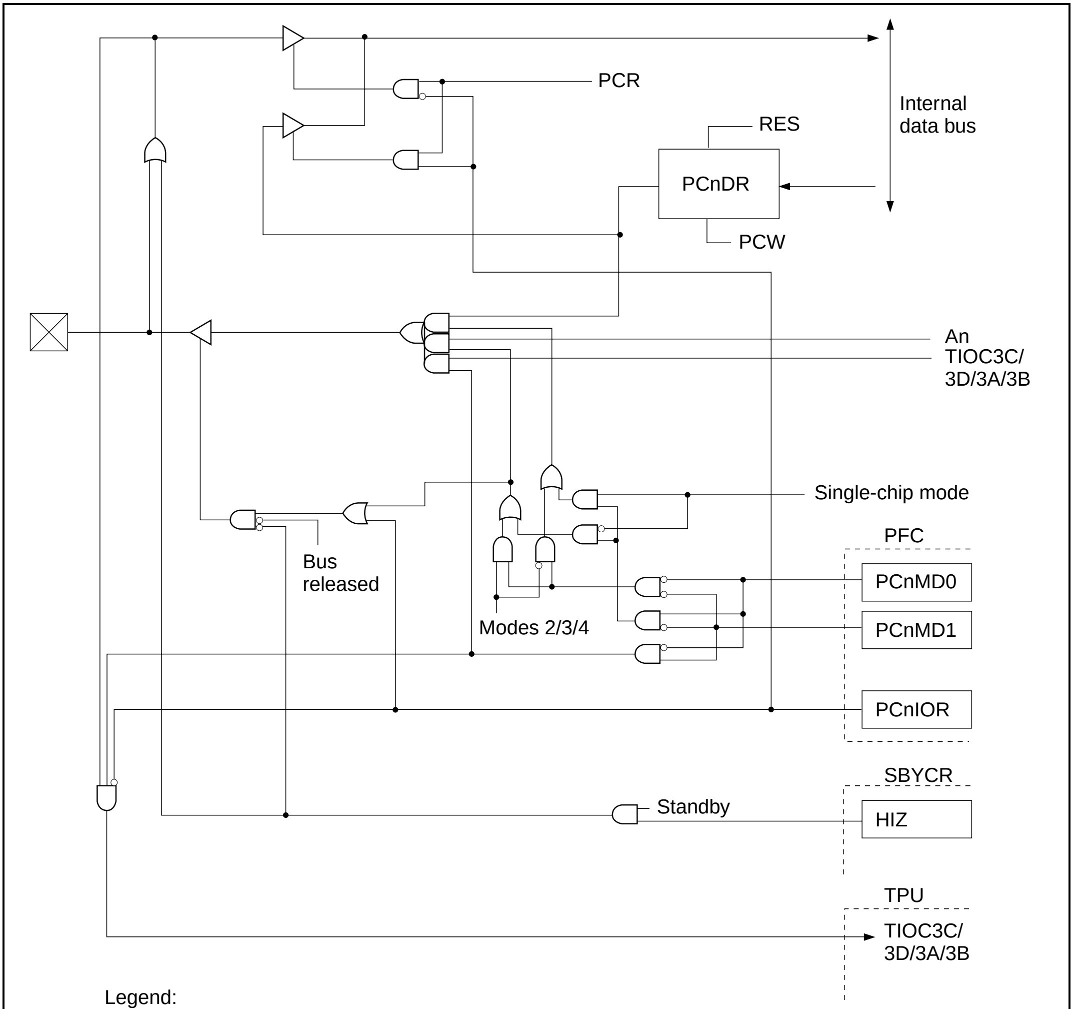
Figure C.13 PCn/An Block Diagram (for Mask ROM version)



Legend:
PCR: Port C read signal
PCW: Port C write signal
RES: Reset signal

Note: n = 14 to 17

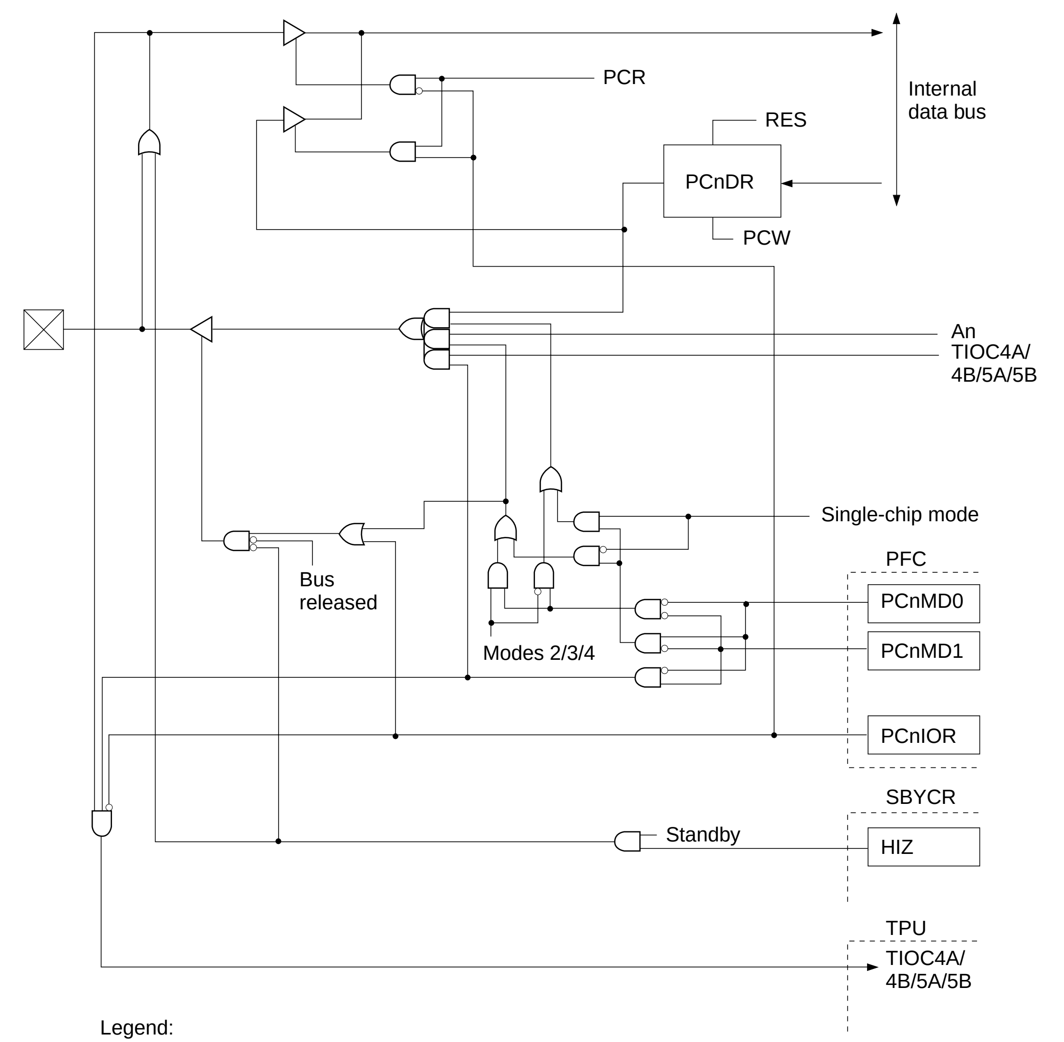
Figure C.14 PC14/A14/TIOC3C, PC15/A15/TIOC3D, PC16/A16/TIOC3A, PC17/A17/TIOC3B Block Diagram (for F-ZTAT version)



Legend:
PCR: Port C read signal
PCW: Port C write signal
RES: Reset signal

Note: n = 14 to 17

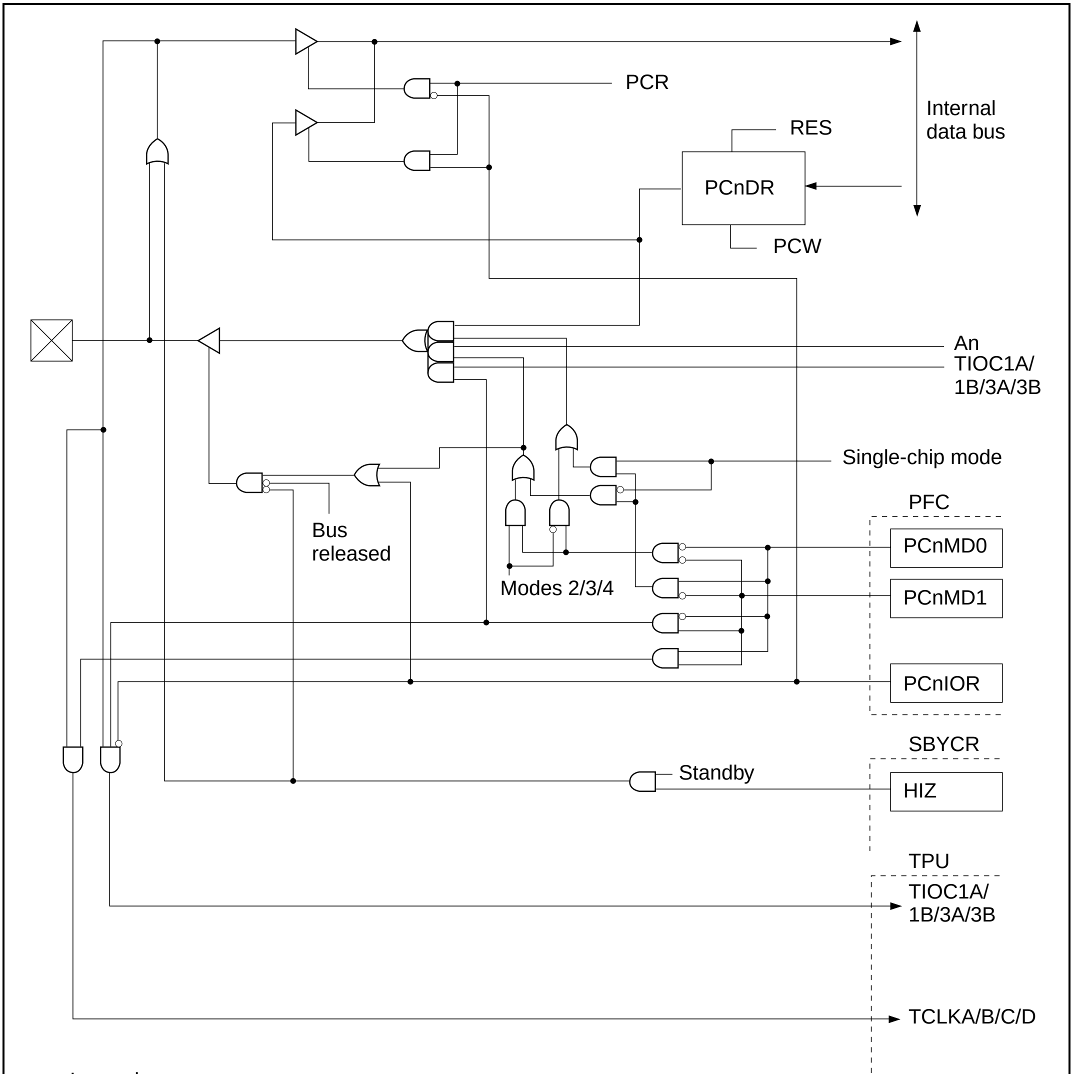
Figure C.15   PC14/A14/TIOC3C, PC15/A15/TIOC3D, PC16/A16/TIOC3A, PC17/A17/TIOC3B Block Diagram (for Mask ROM version)



Legend:
PCR: Port C read signal
PCW: Port C write signal
RES: Reset signal

Note: n = 18 to 21

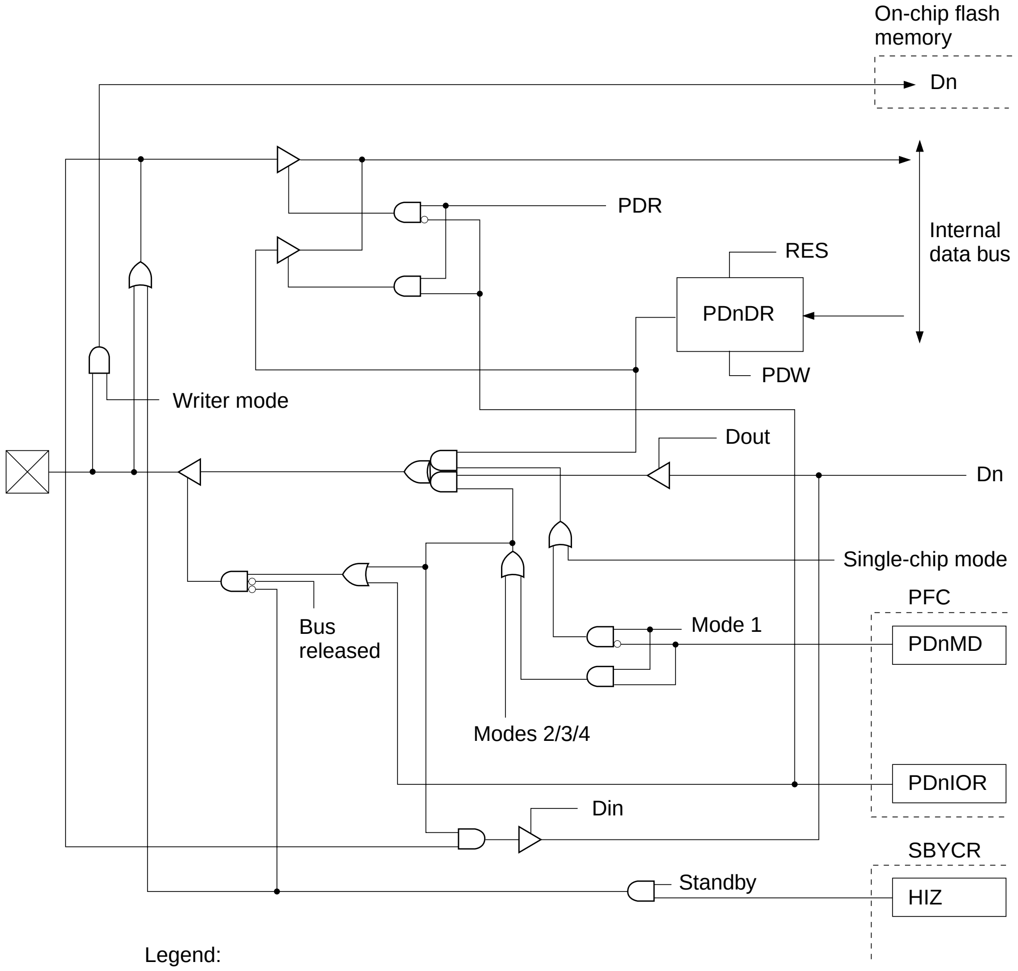
Figure C.16 PC18/A18/TIOC4A, PC19/A19/TIOC4B, PC20/A20/TIOC5A, PC21/A21/TIOC5B Block Diagram



Legend:
PCR: Port C read signal
PCW: Port C write signal
RES: Reset signal

Note: n = 22 to 25

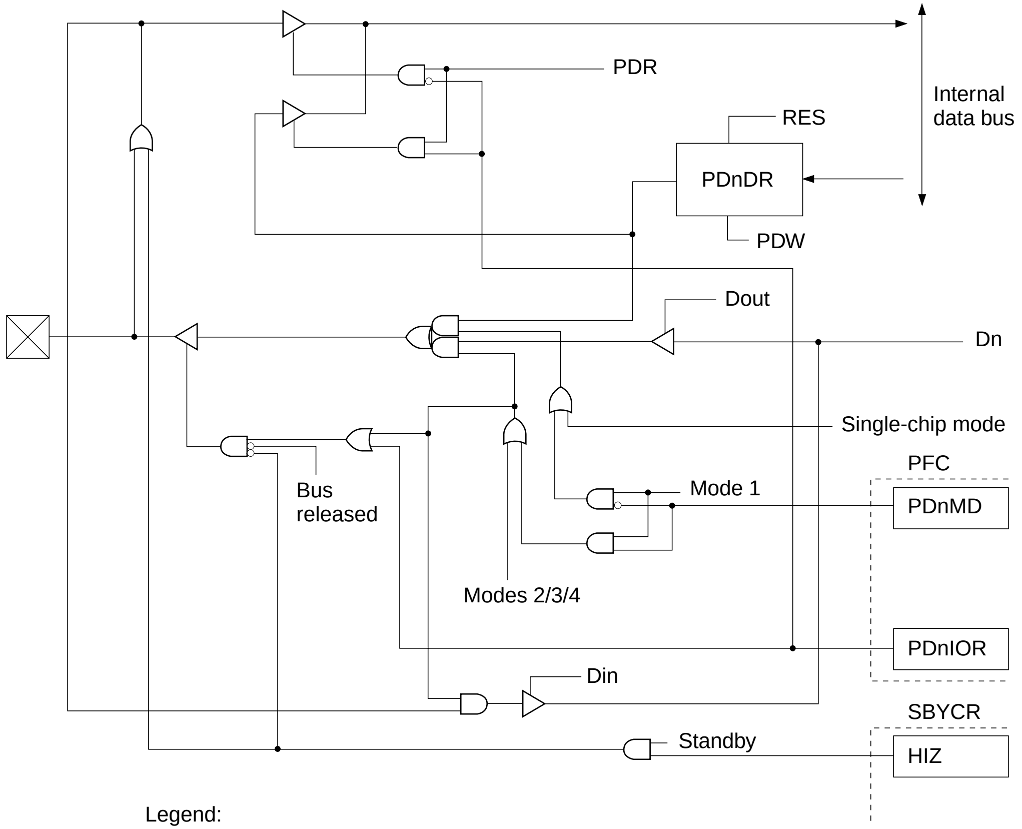
Figure C.17 PC22/A22/TIOC1A/TCLKA, PC23/A23/TIOC1B/TCLKB, PC24/A24/TIOC3A/TCLKC, PC25/A25/TIOC3B/TCLKD Block Diagram



Legend:
PDR: Port D read signal
PDW: Port D write signal
RES: Reset signal
Dout: Data bus output timing signal
Din: Data bus input timing signal

Note: n = 0 to 7

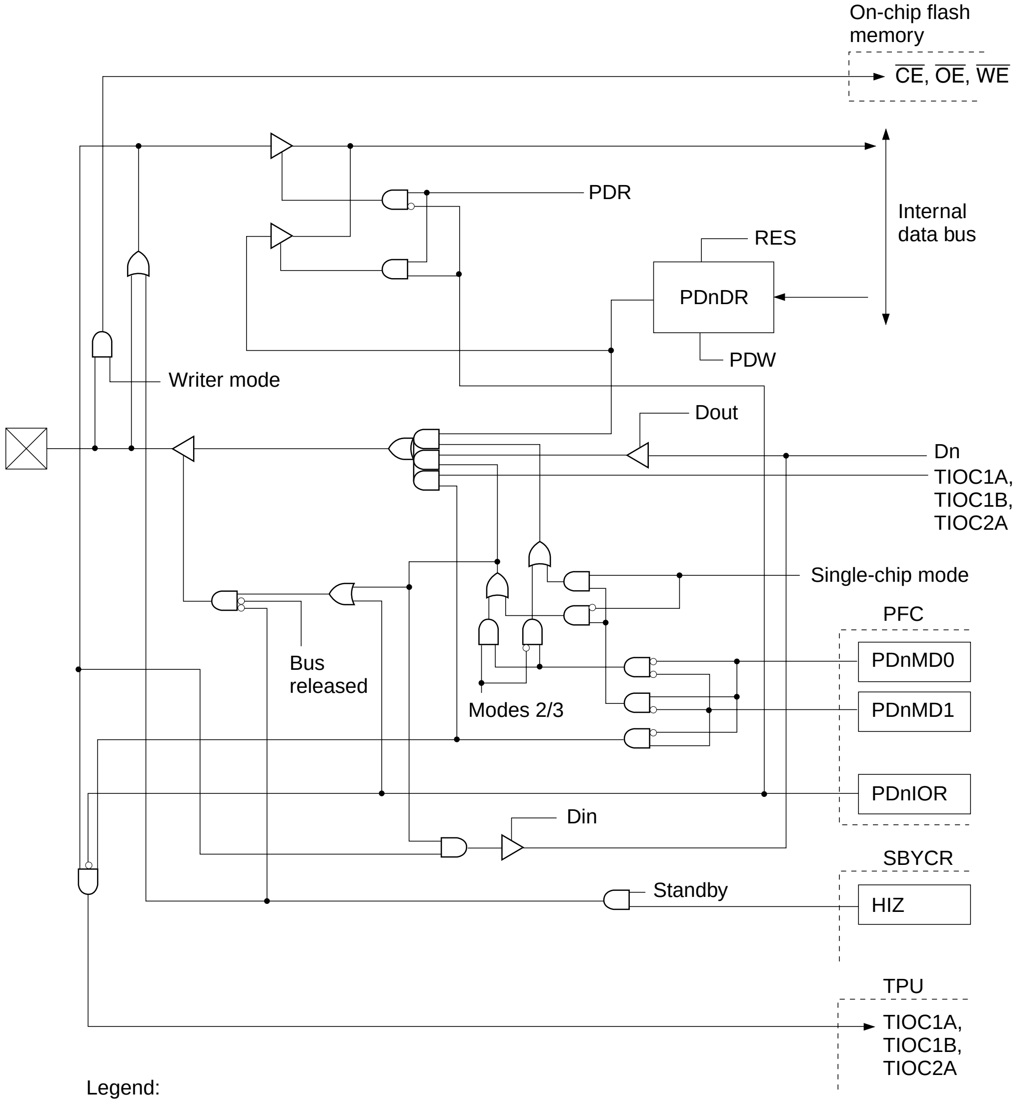
Figure C.18 PDn/Dn Block Diagram (for F-ZTAT version)



Legend:
PDR: Port D read signal
PDW: Port D write signal
RES: Reset signal
Dout: Data bus output timing signal
Din: Data bus input timing signal

Note: n = 0 to 7

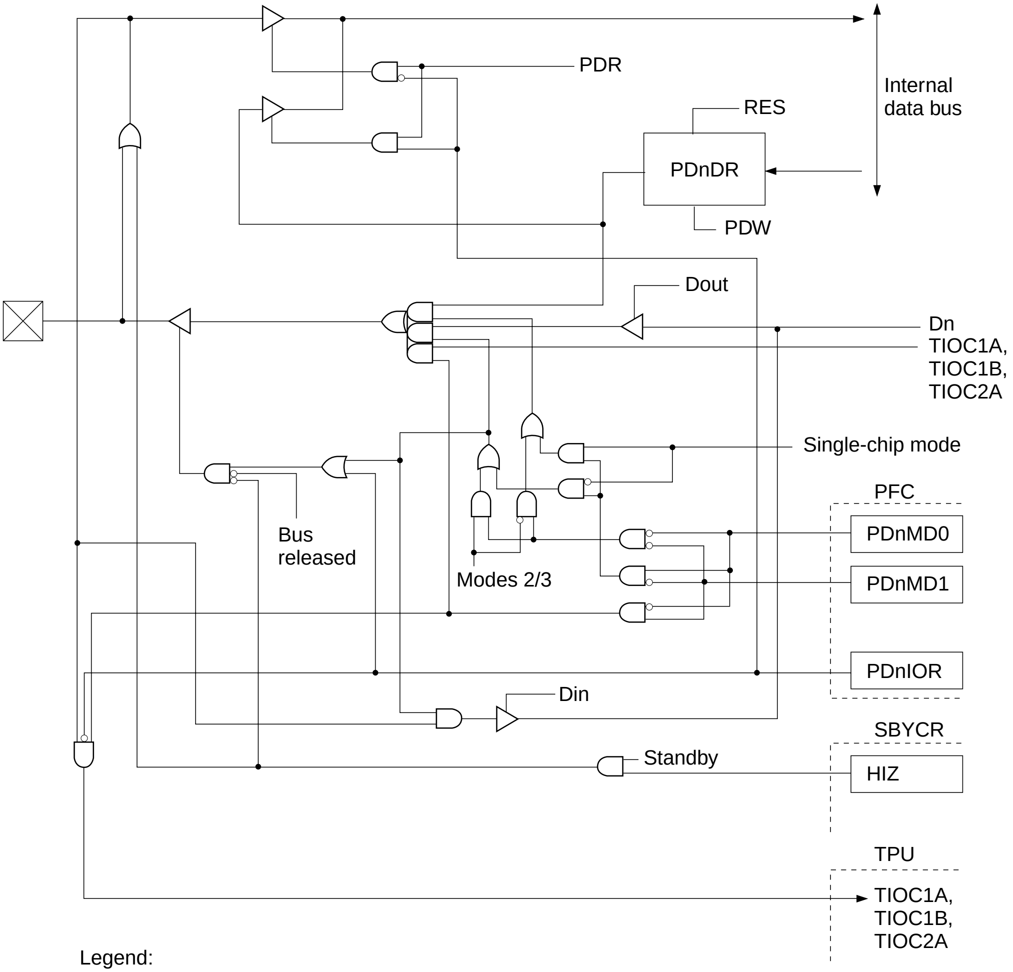
Figure C.19 PDn/Dn Block Diagram (for Mask ROM version)



Legend:
PDR: Port D read signal
PDW: Port D write signal
RES: Reset signal
Dout: Data bus output timing signal
Din: Data bus input timing signal

Note: n = 8 to 10

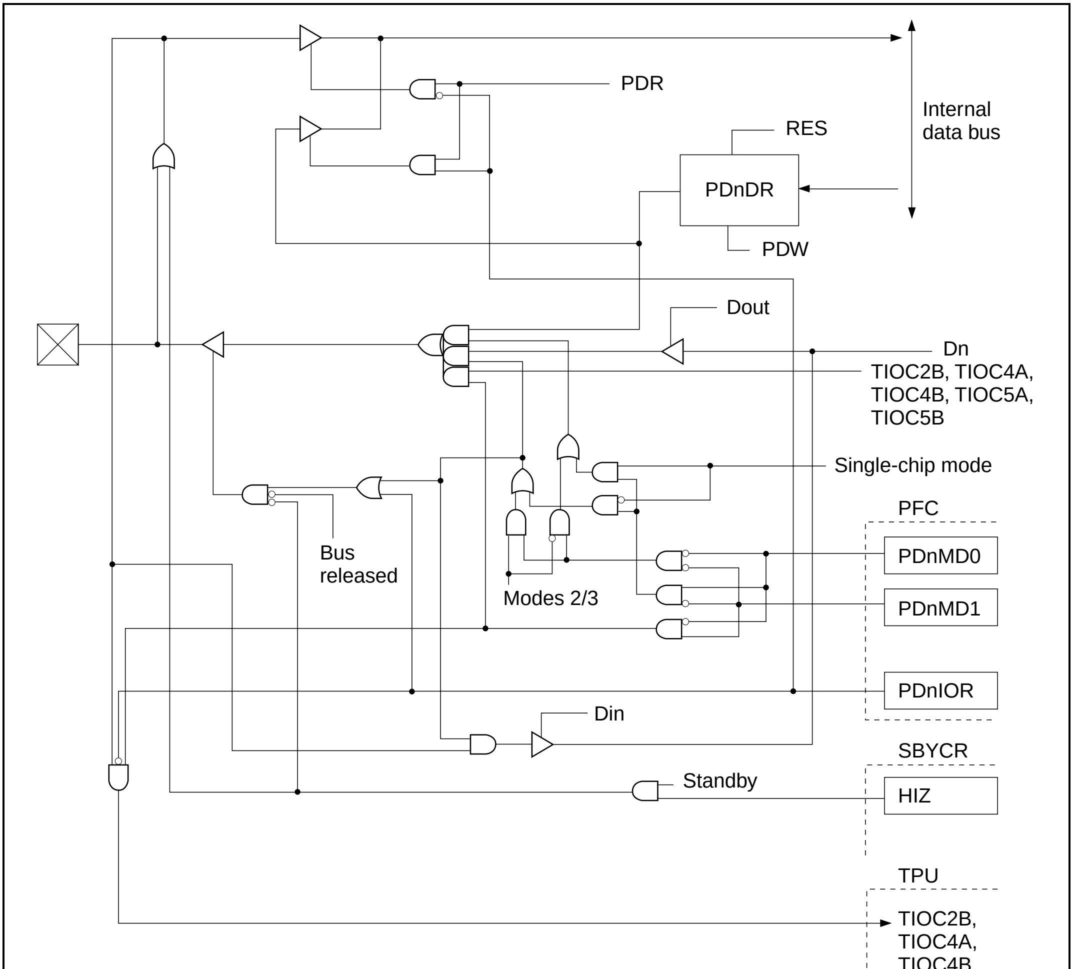
Figure C.20 PD8/D8/TIOC1A, PD9/D9/TIOC1B, PD10/D10/TIOC2A Block Diagram (for F-ZTAT version)



Legend:
PDR: Port D read signal
PDW: Port D write signal
RES: Reset signal
Dout: Data bus output timing signal
Din: Data bus input timing signal

Note: n = 8 to 10

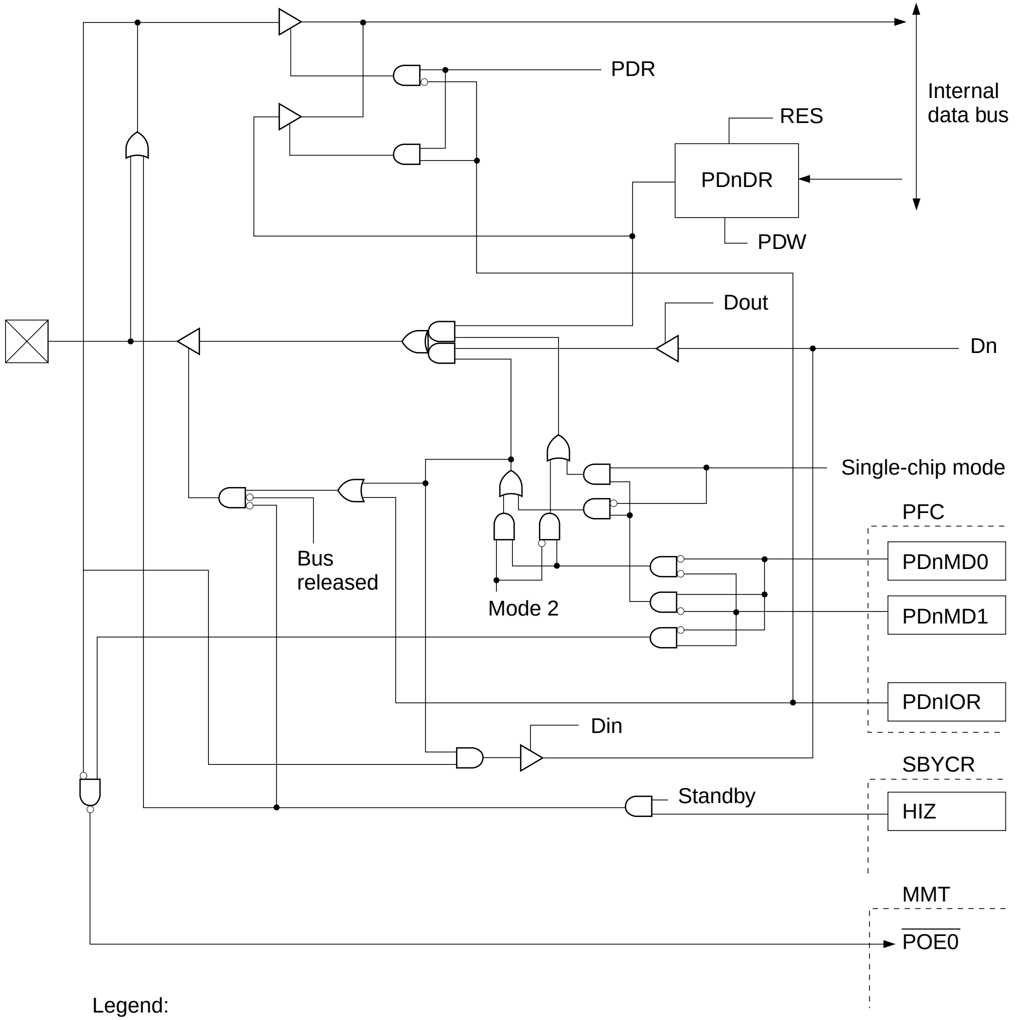
Figure C.21 PD8/D8/TIOC1A, PD9/D9/TIOC1B, PD10/D10/TIOC2A Block Diagram (for Mask ROM version)



Legend:
PDR: Port D read signal
PDW: Port D write signal
RES: Reset signal
Dout: Data bus output timing signal
Din: Data bus input timing signal

Note: n = 11 to 15

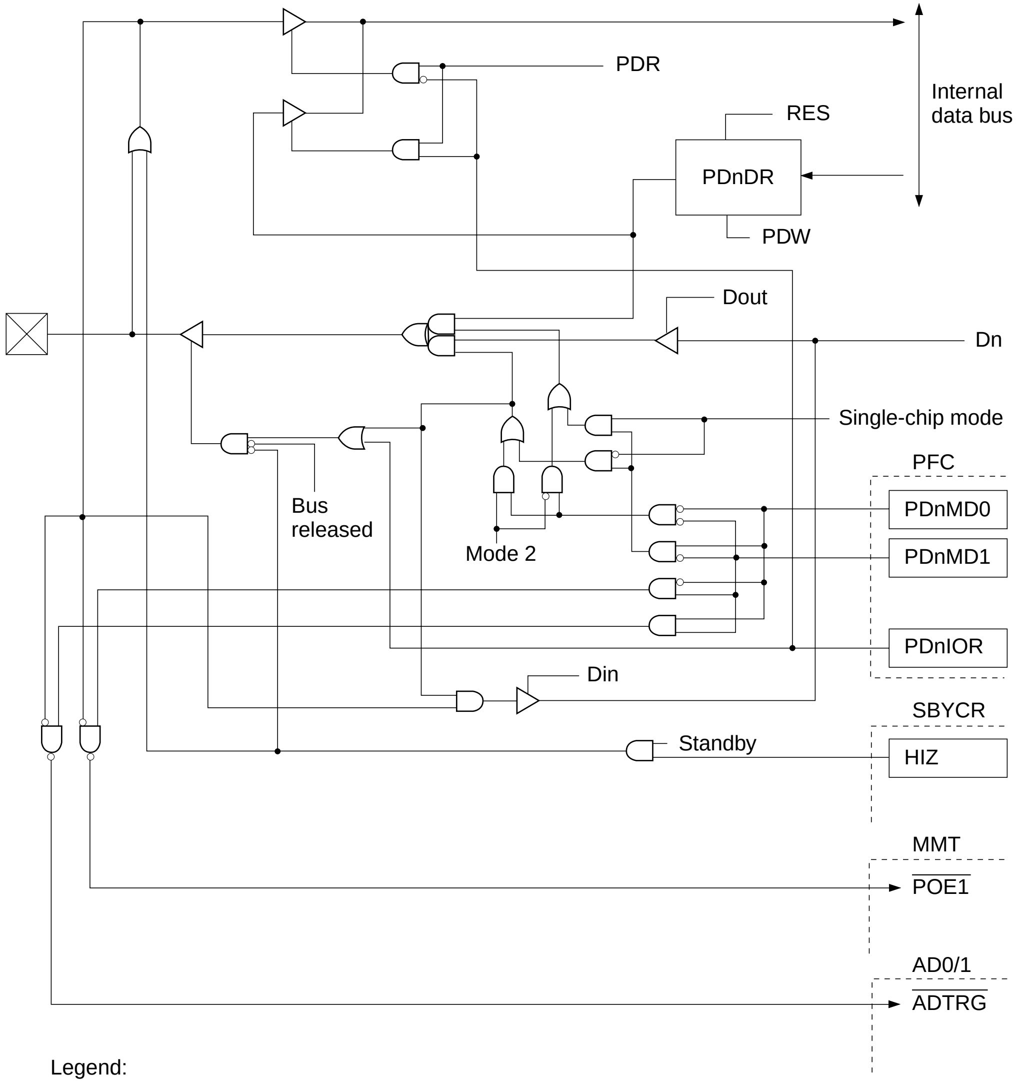
Figure C.22 PD11/D11/TIOC2B, PD12/D12/TIOC4A, PD13/D13/TIOC4B, PD14/D14/TIOC5A, PD15/D15/TIOC5B Block Diagram



Legend:
PDR: Port D read signal
PDW: Port D write signal
RES: Reset signal
Dout: Data bus output timing signal
Din: Data bus input timing signal

Note: n = 16

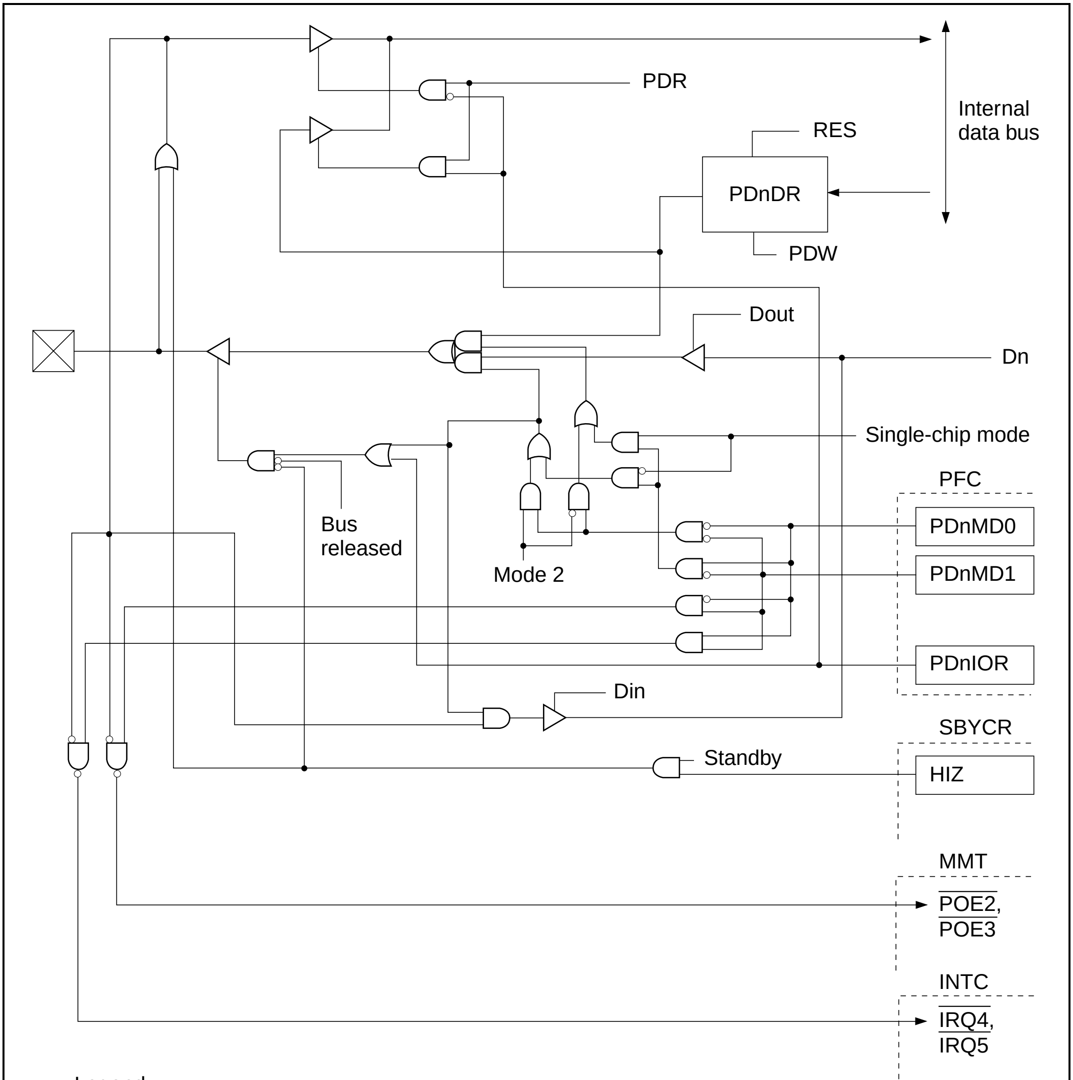
Figure C.23 PD16/D16/$\overline{\text{POE0}}$ Block Diagram



Legend:
PDR: Port D read signal
PDW: Port D write signal
RES: Reset signal
Dout: Data bus output timing signal
Din: Data bus input timing signal

Note: n = 17

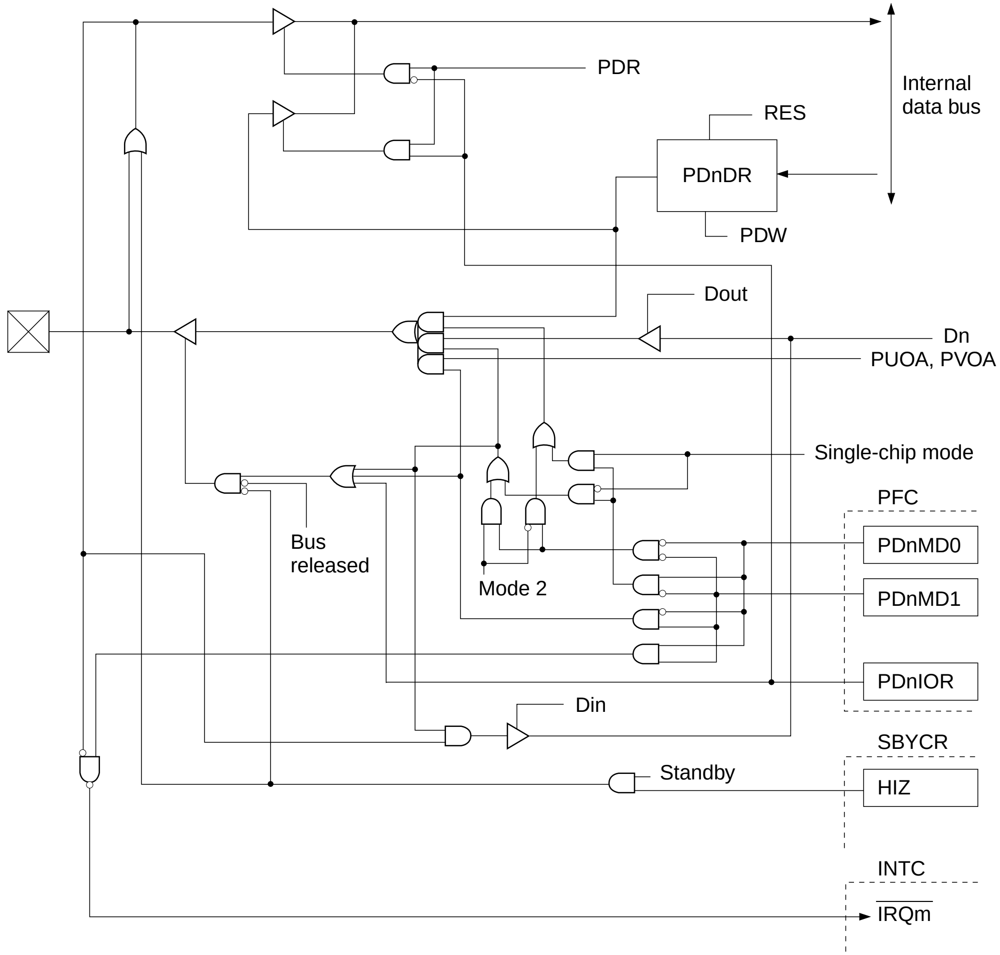
Figure C.24 PD17/D17/$\overline{\text{POE1}}$/$\overline{\text{ADTRG}}$ Block Diagram



Legend:
PDR: Port D read signal
PDW: Port D write signal
RES: Reset signal
Dout: Data bus output timing signal
Din: Data bus input timing signal

Note: n = 18, 19

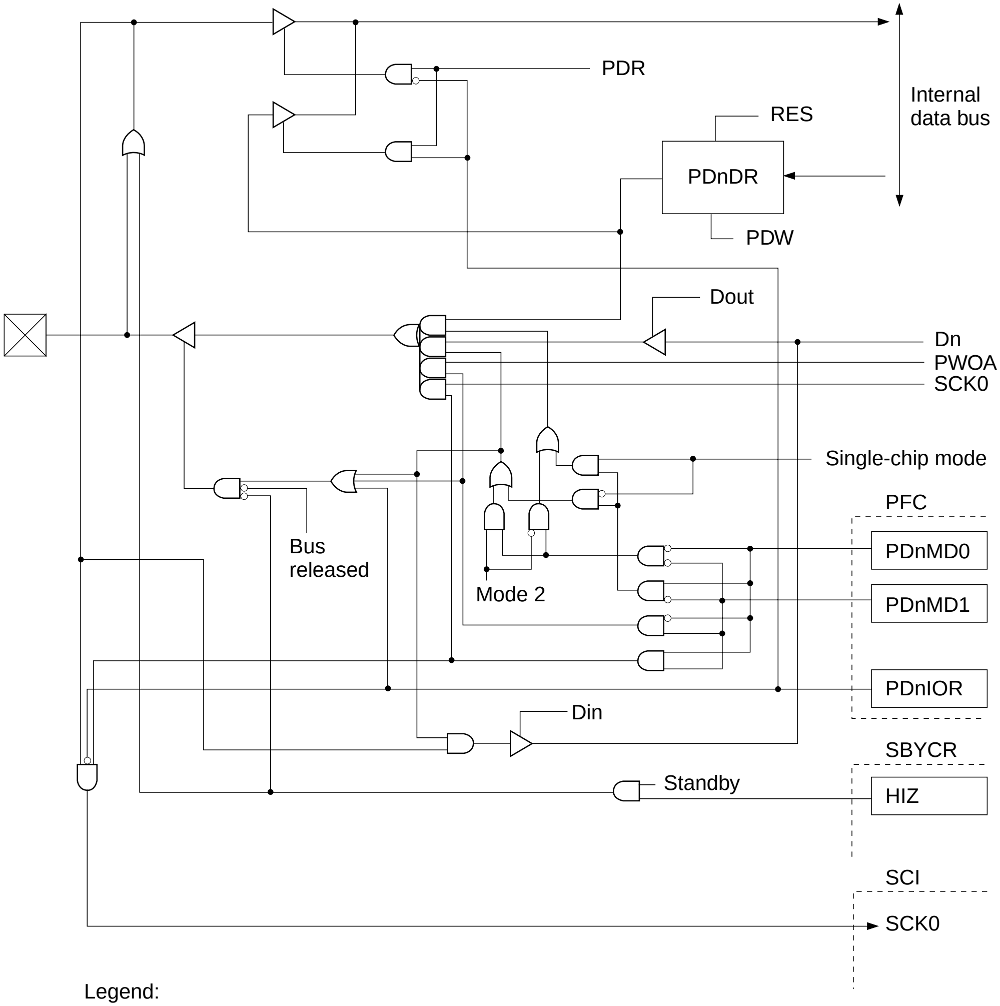
Figure C.25 PD18/D18/$\overline{\text{POE2}}$/$\overline{\text{IRQ4}}$, PD19/D19/$\overline{\text{POE3}}$/$\overline{\text{IRQ5}}$ Block Diagram



Legend:
PDR: Port D read signal
PDW: Port D write signal
RES: Reset signal
Dout: Data bus output timing signal
Din: Data bus input timing signal

Note: n = 20, 21
m = 6, 7

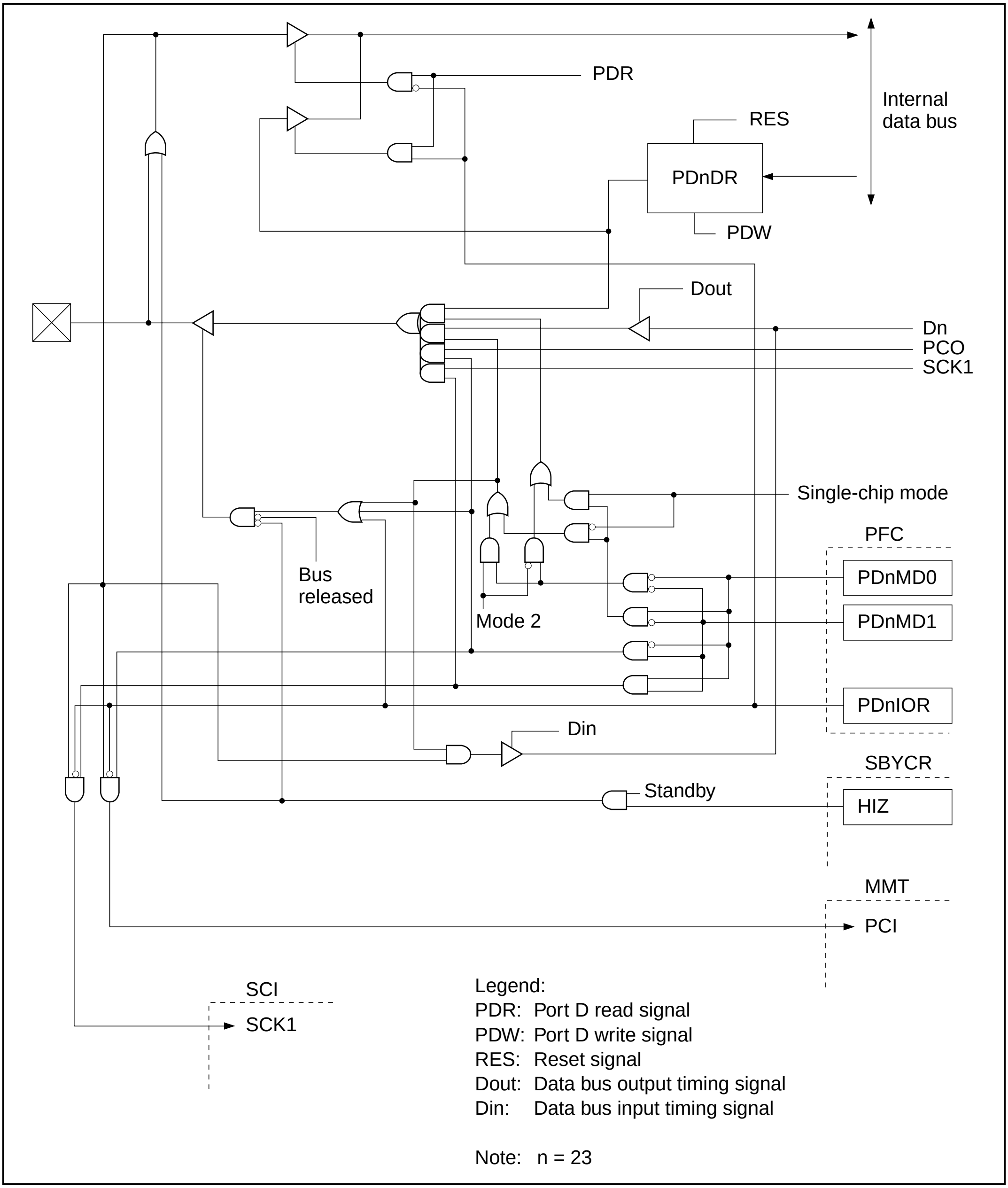
Figure C.26 PD20/D20/PUOA/$\overline{\text{IRQ6}}$, PD21/D21/PVOA/$\overline{\text{IRQ7}}$ Block Diagram



Legend:
PDR: Port D read signal
PDW: Port D write signal
RES: Reset signal
Dout: Data bus output timing signal
Din: Data bus input timing signal

Note: n = 22

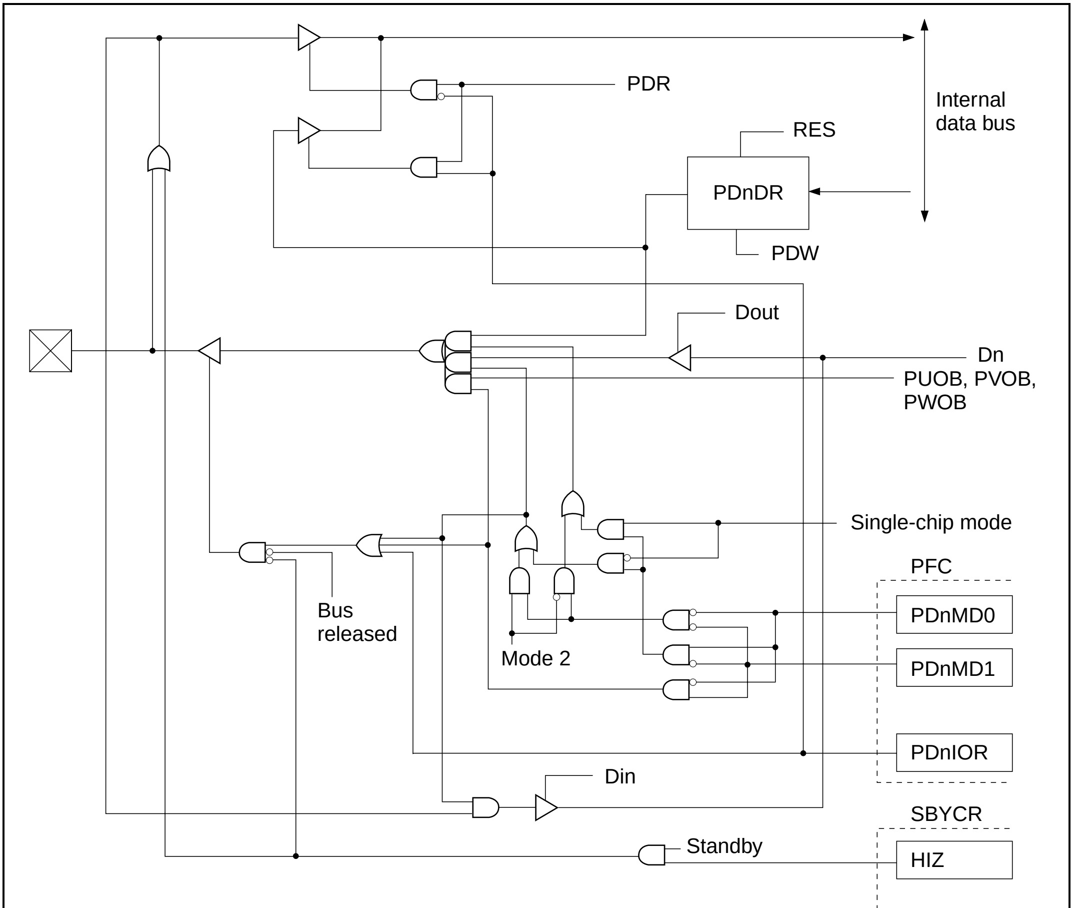
Figure C.27 PD22/D22/PWOA/SCK0 Block Diagram



Legend:
PDR: Port D read signal
PDW: Port D write signal
RES: Reset signal
Dout: Data bus output timing signal
Din: Data bus input timing signal

Note: n = 23

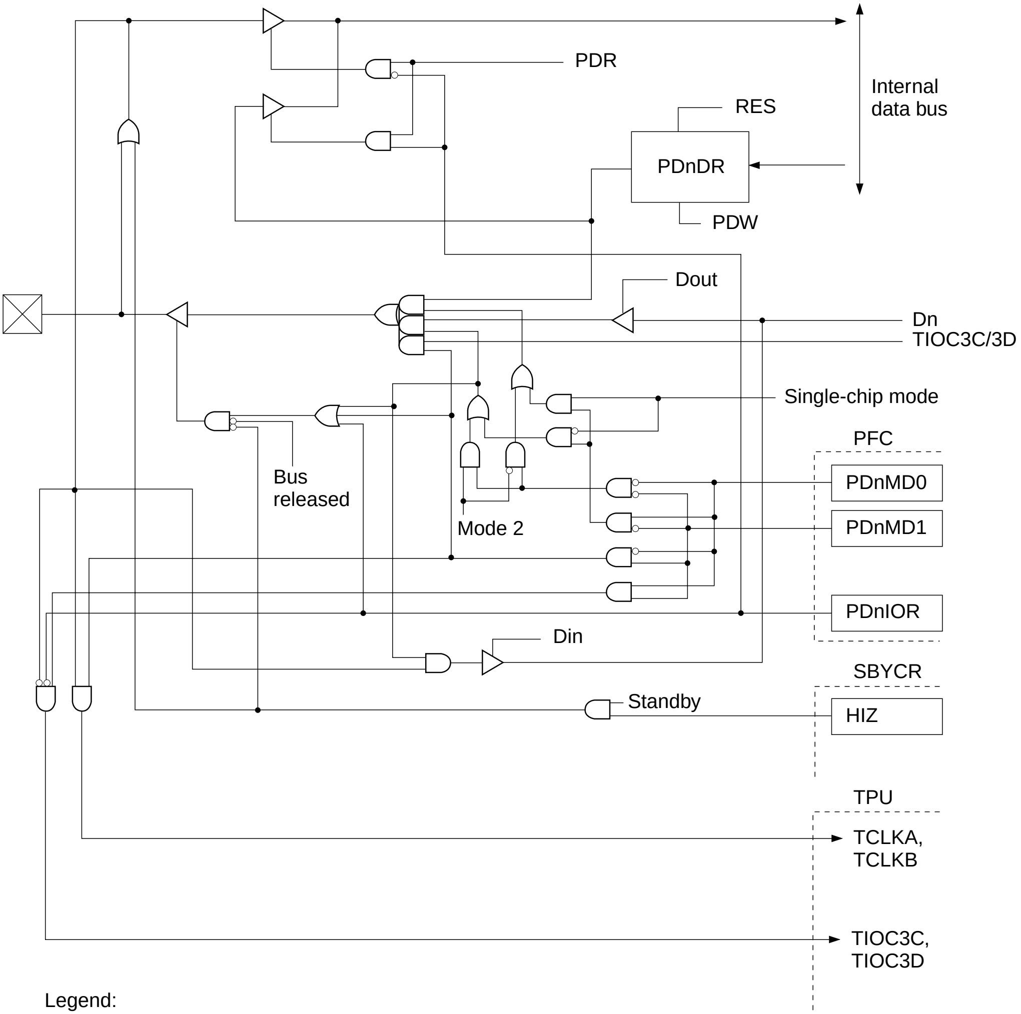
Figure C.28 PD23/D23/PCO/PCI/SCK1 Block Diagram



Legend:
PDR: Port D read signal
PDW: Port D write signal
RES: Reset signal
Dout: Data bus output timing signal
Din: Data bus input timing signal

Note: n = 24 to 26

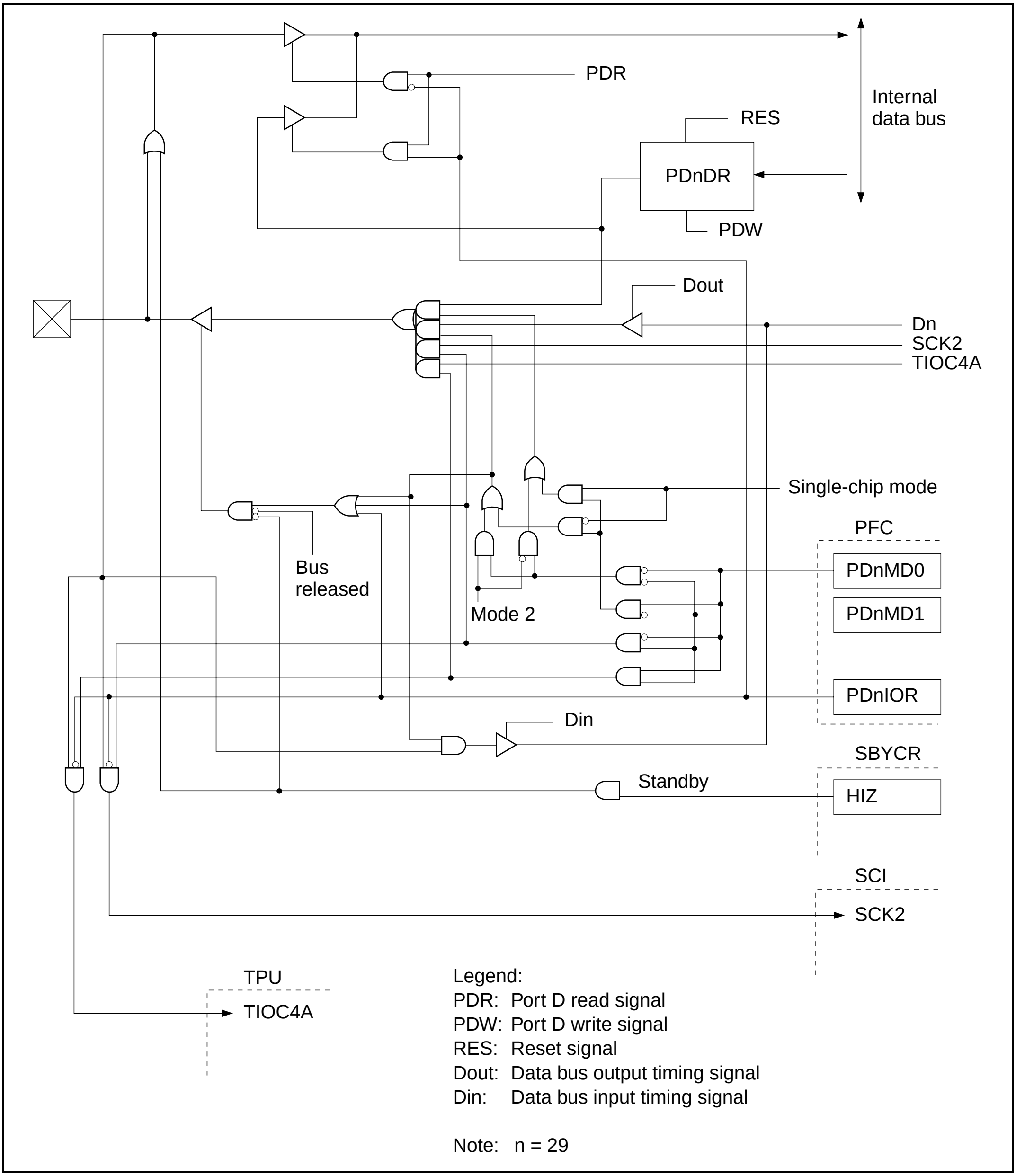
Figure C.29 PD24/D24/PUOB, PD25/D25/PVOB, PD26/D26/PWOB Block Diagram



Legend:
PDR: Port D read signal
PDW: Port D write signal
RES: Reset signal
Dout: Data bus output timing signal
Din: Data bus input timing signal

Note: n = 27, 28

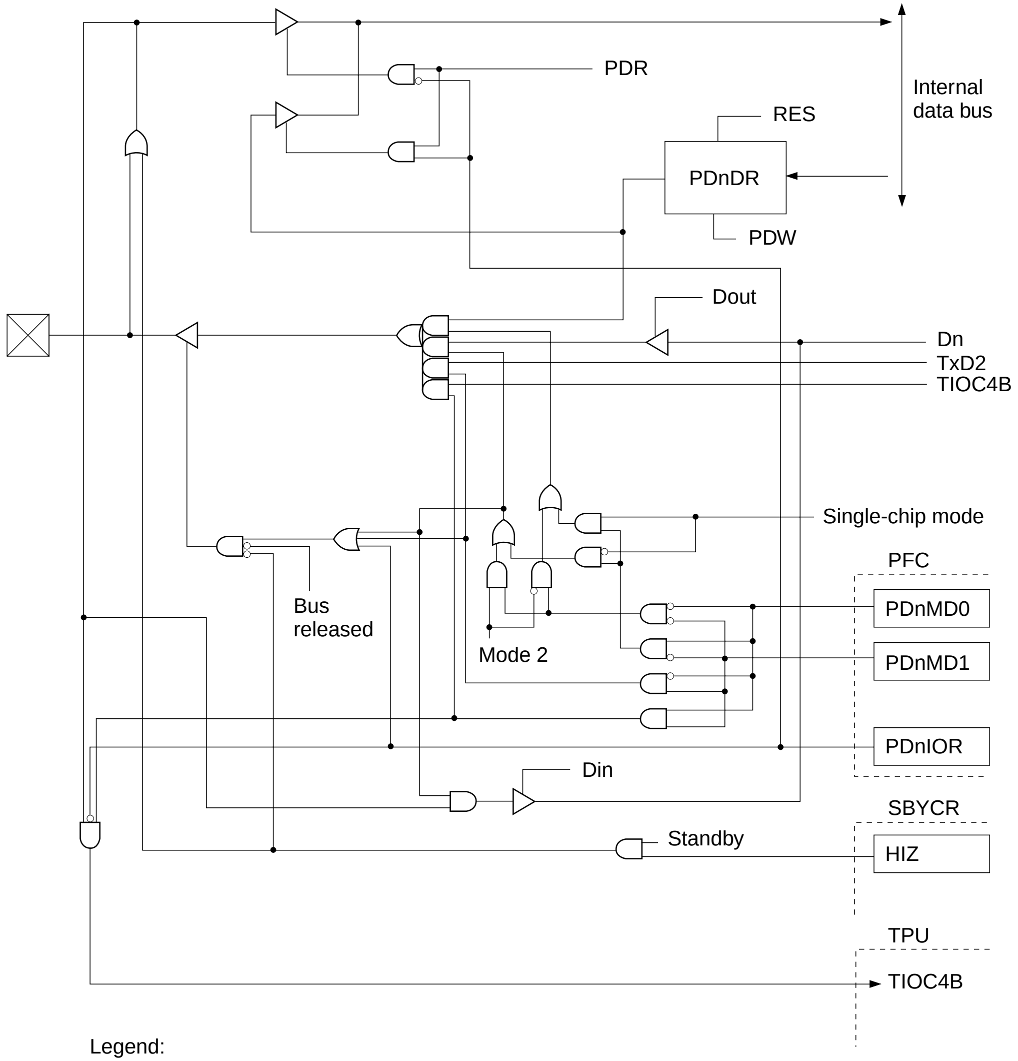
Figure C.30 PD27/D27/TCLKA/TIOC3C, PD28/D28/TCLKB/TIOC3D Block Diagram



Legend:
PDR: Port D read signal
PDW: Port D write signal
RES: Reset signal
Dout: Data bus output timing signal
Din: Data bus input timing signal

Note: n = 29

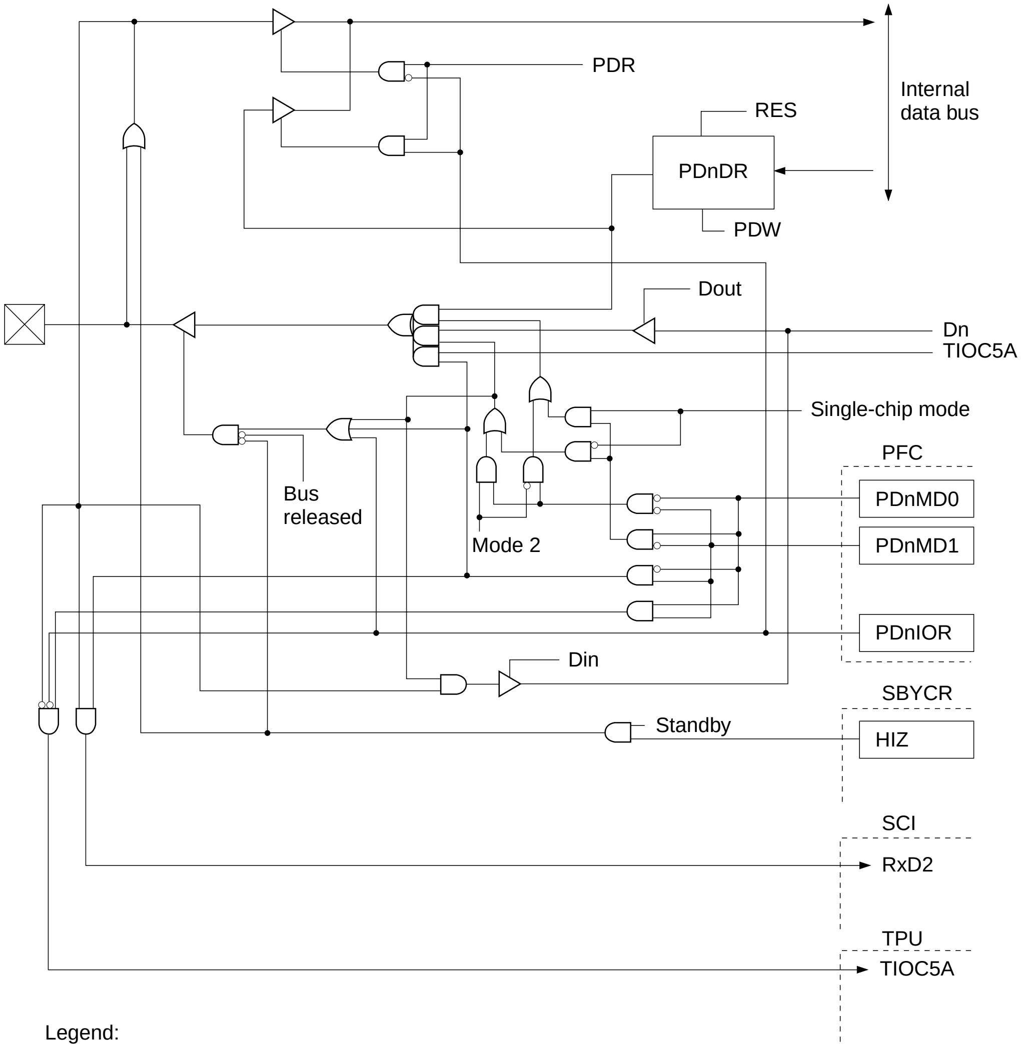
Figure C.31 PD29/D29/SCK2/TIOC4A Block Diagram



Legend:
PDR: Port D read signal
PDW: Port D write signal
RES: Reset signal
Dout: Data bus output timing signal
Din: Data bus input timing signal

Note: n = 30

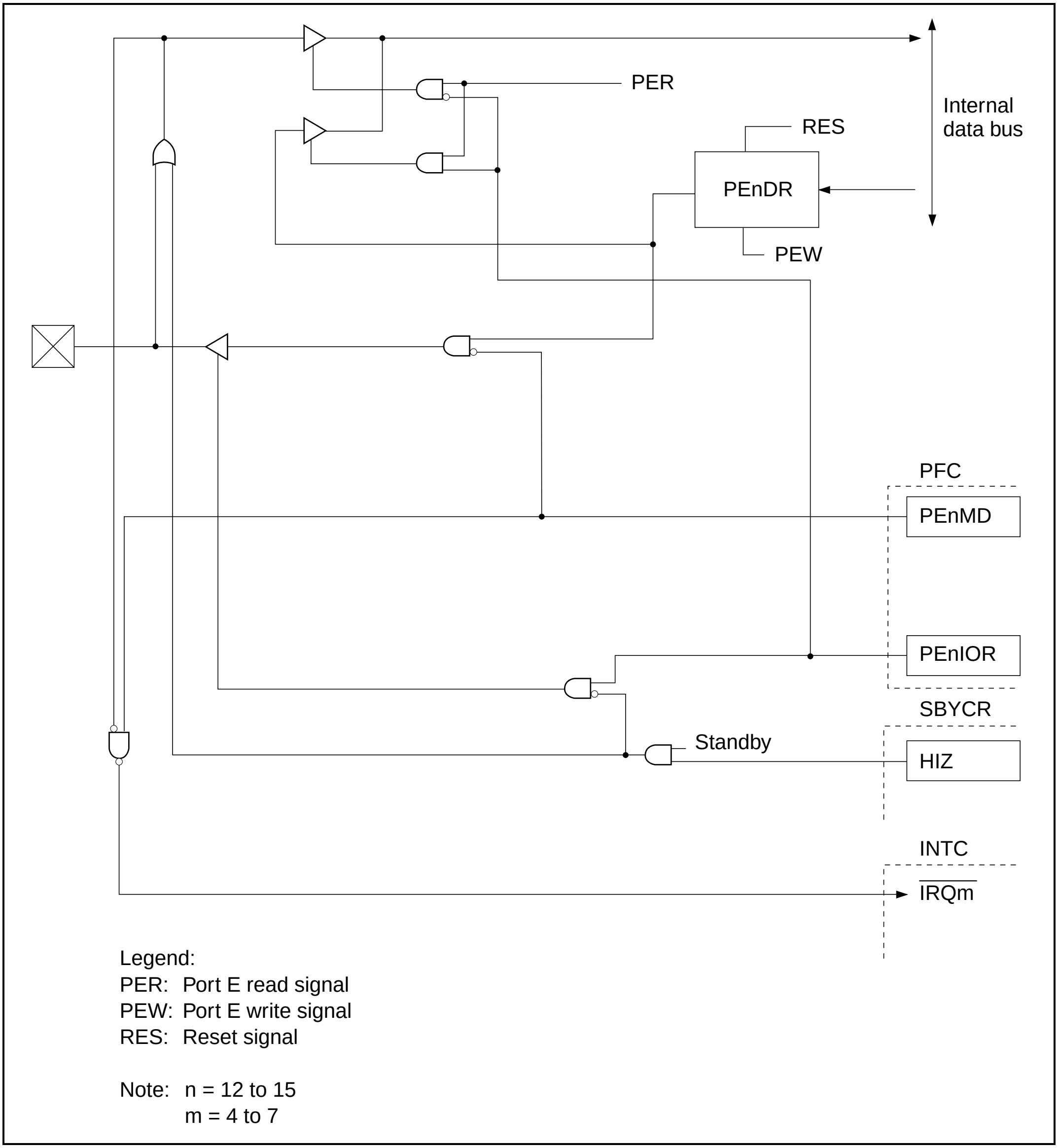
Figure C.32 PD30/D30/TxD2/TIOC4B Block Diagram



Legend:
PDR: Port D read signal
PDW: Port D write signal
RES: Reset signal
Dout: Data bus output timing signal
Din: Data bus input timing signal

Note: n = 31

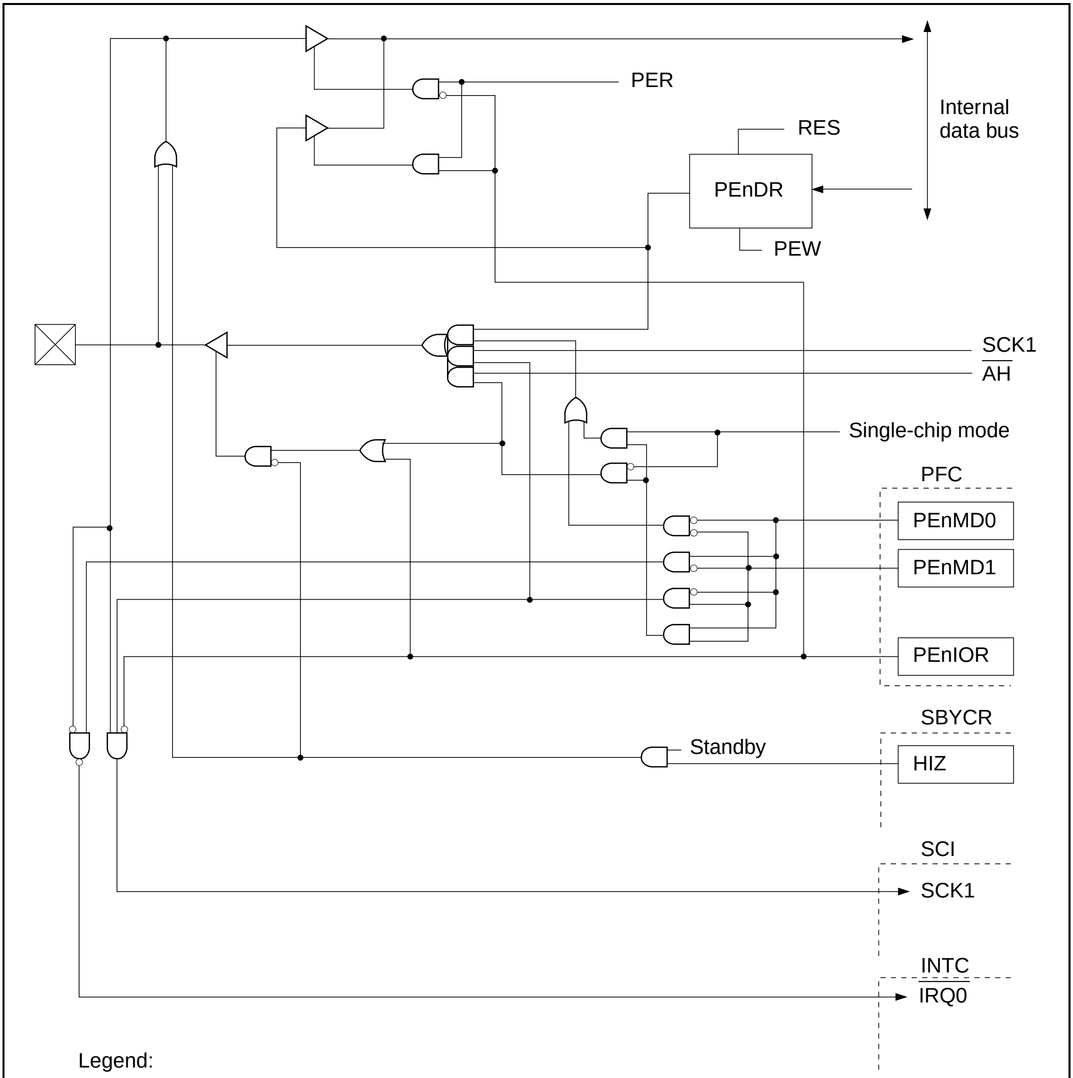
Figure C.33 PD31/D31/RxD2/TIOC5A Block Diagram



Legend:
PER: Port E read signal
PEW: Port E write signal
RES: Reset signal

Note: n = 12 to 15
m = 4 to 7

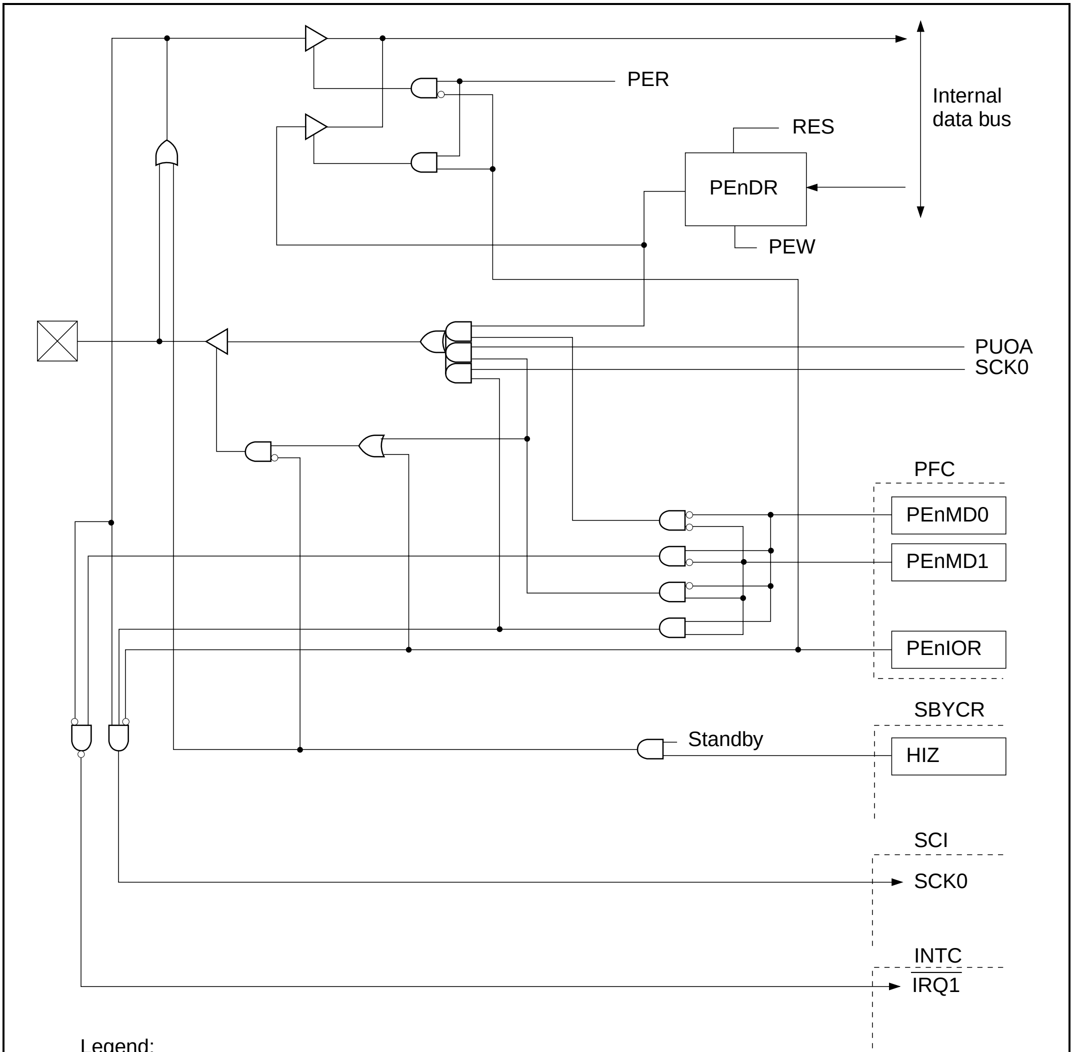
Figure C.34 PEn/$\overline{\text{IRQm}}$ Block Diagram



Legend:
PER: Port E read signal
PEW: Port E write signal
RES: Reset signal

Note: n = 16

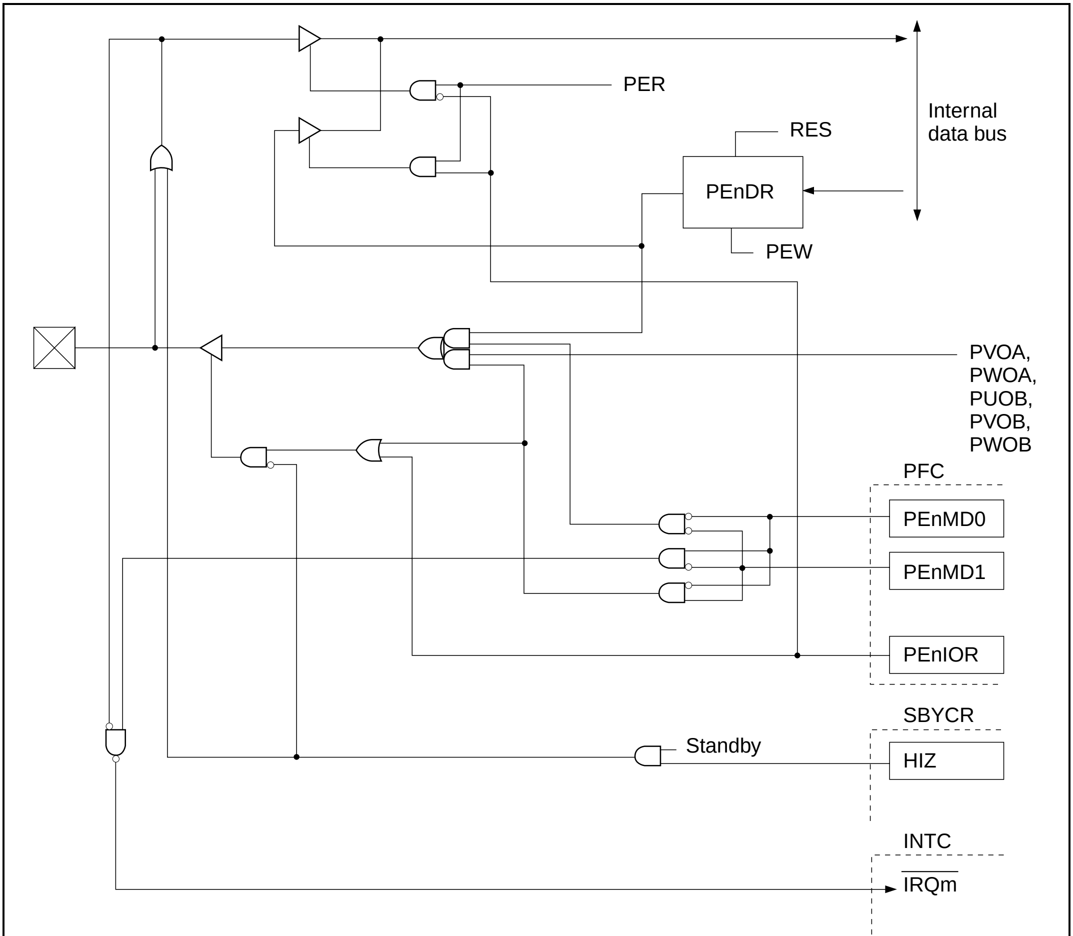
Figure C.35 PE16/$\overline{\text{IRQ0}}$/SCK1/$\overline{\text{AH}}$ Block Diagram



Legend:
PER: Port E read signal
PEW: Port E write signal
RES: Reset signal

Note: n = 17

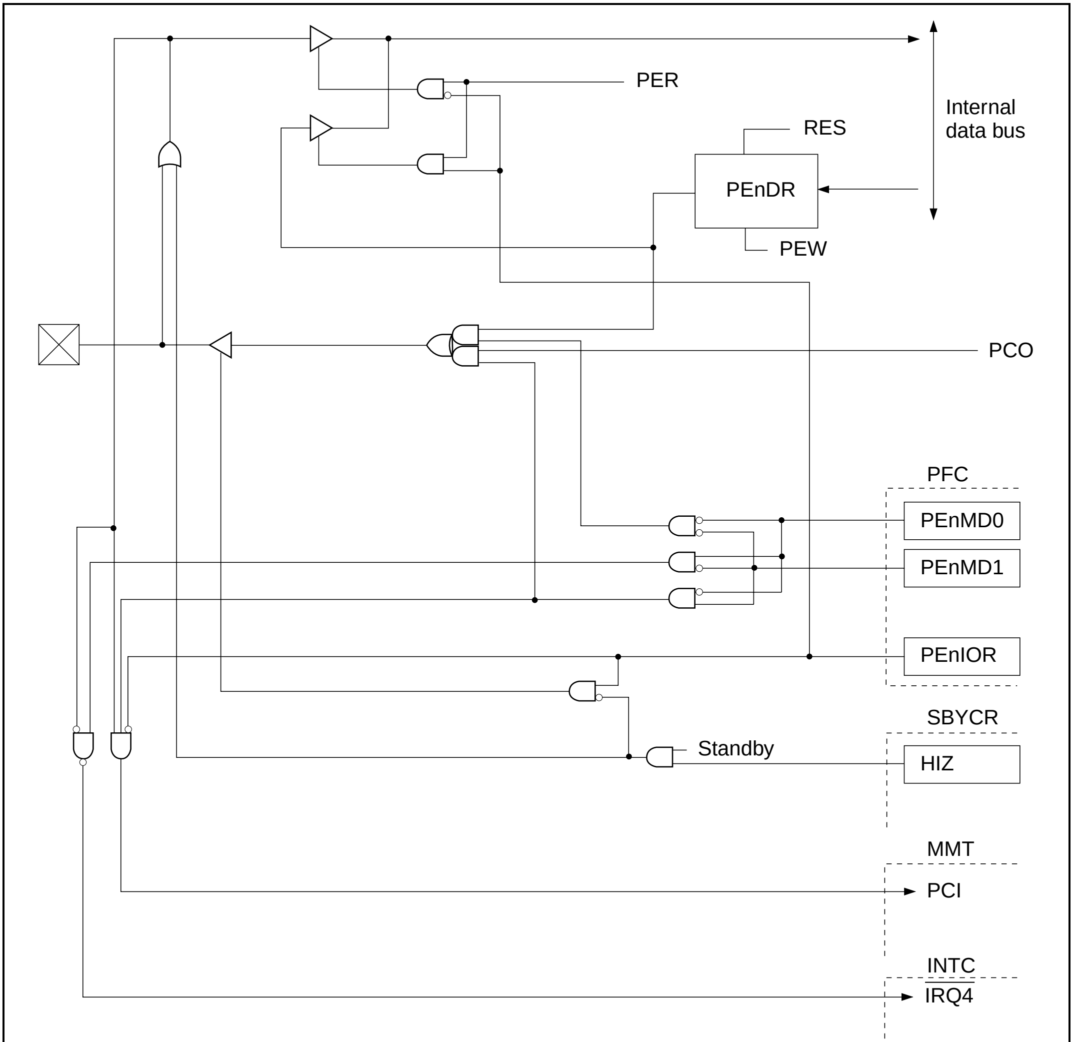
Figure C.36 PE17/$\overline{\text{IRQ1}}$/PUOA/SCK0 Block Diagram



Legend:
PER: Port E read signal
PEW: Port E write signal
RES: Reset signal

Note: n = 18, 19, or 21 to 23
m = 2, 3, or 5 to 7

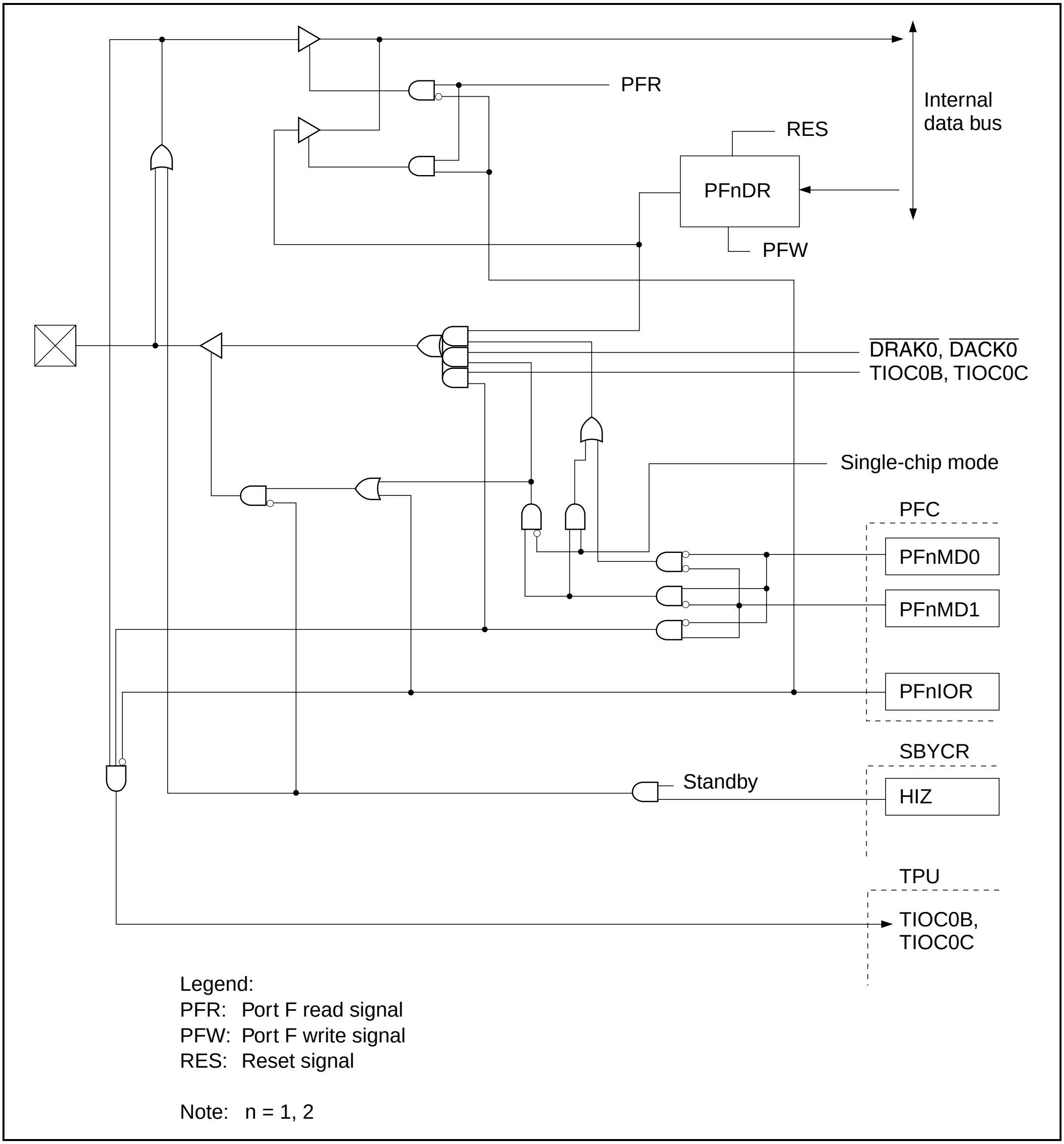
Figure C.37 PE18/$\overline{\text{IRQ2}}$/PVOA, PE19/$\overline{\text{IRQ3}}$/PWOA, PE21/$\overline{\text{IRQ5}}$/PUOB, PE22/$\overline{\text{IRQ6}}$/PVOB, PE23/$\overline{\text{IRQ7}}$/PWOB Block Diagram



Legend:
PER: Port E read signal
PEW: Port E write signal
RES: Reset signal

Note: n = 20

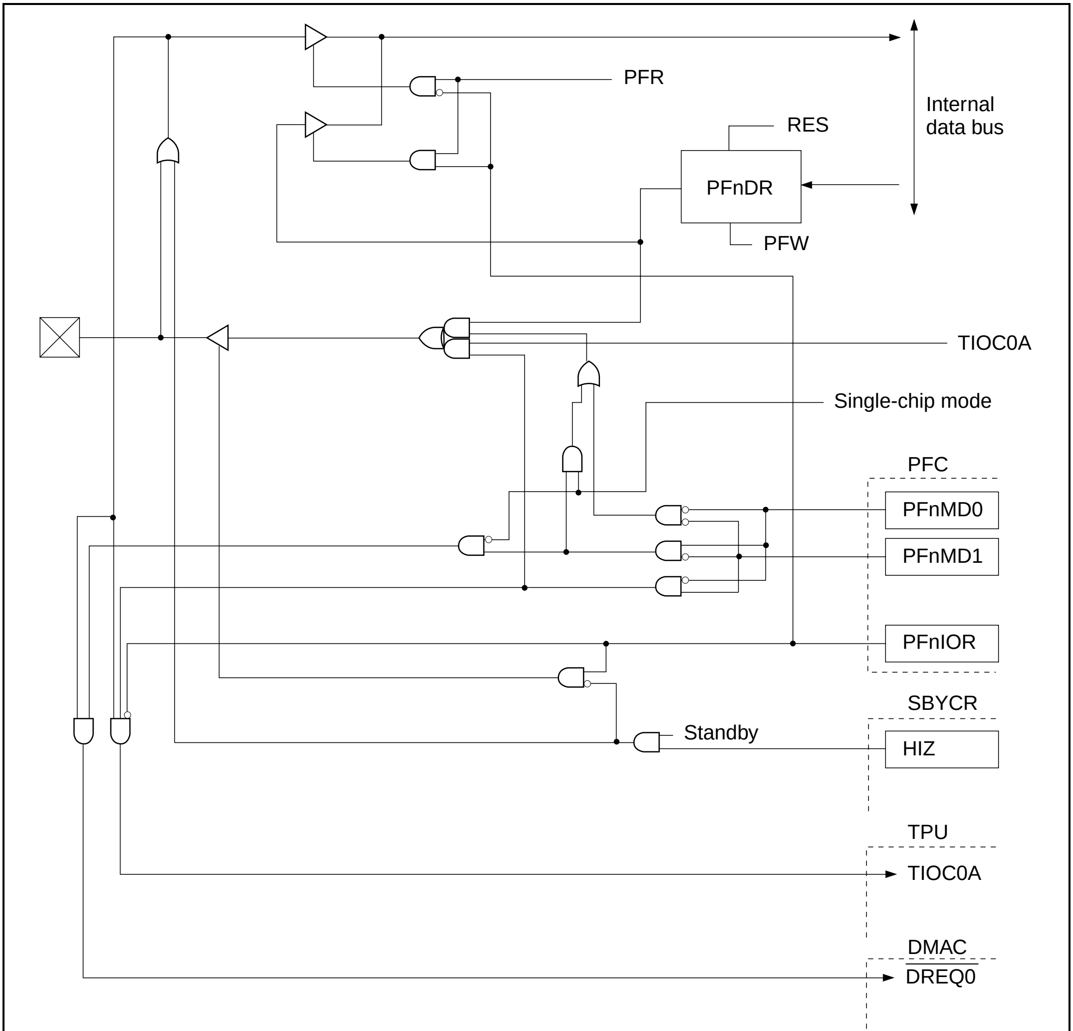
Figure C.38 PE20/$\overline{\text{IRQ4}}$/PCO/PCI Block Diagram



Legend:
PFR: Port F read signal
PFW: Port F write signal
RES: Reset signal

Note: n = 1, 2

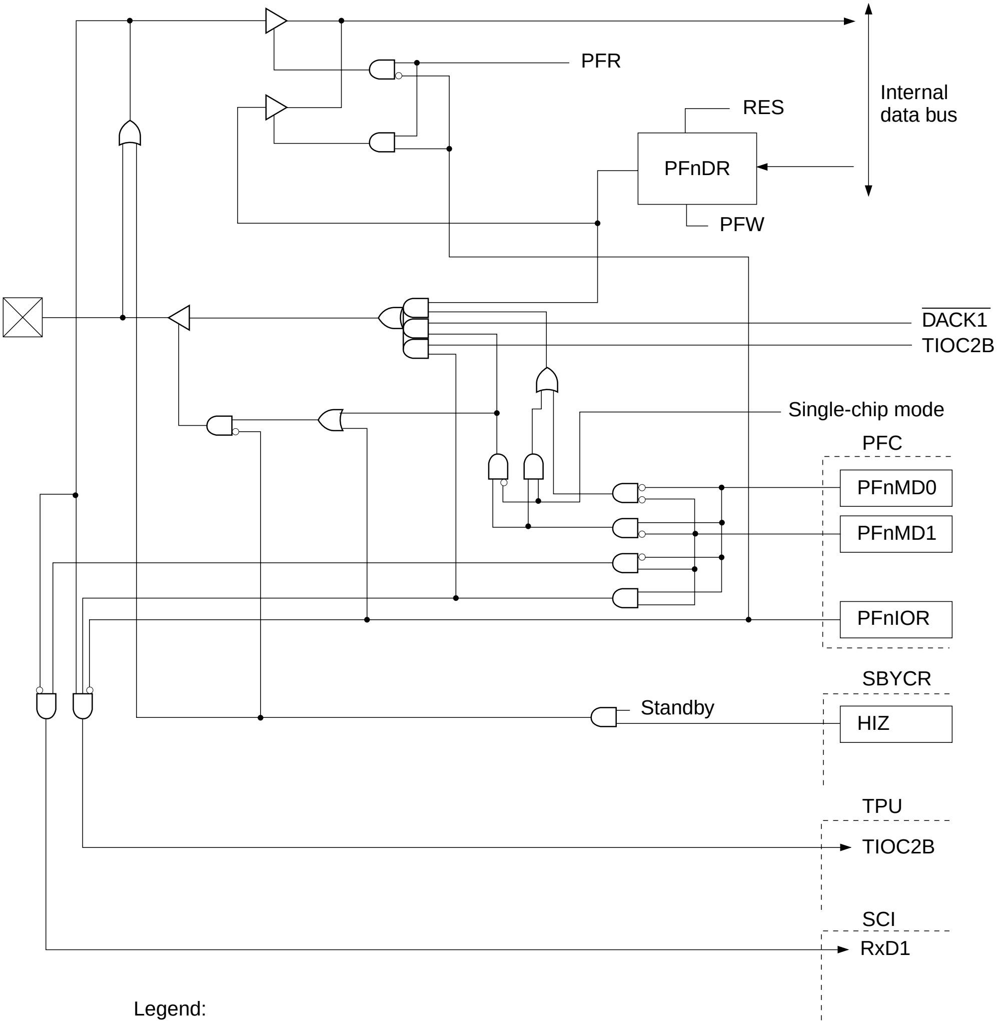
Figure C.39 PF1/$\overline{\text{DACK0}}$/TIOC0B, PF2/$\overline{\text{DRAK0}}$/TIOC0C Block Diagram



Legend:
PFR: Port F read signal
PFW: Port F write signal
RES: Reset signal

Note: n = 3

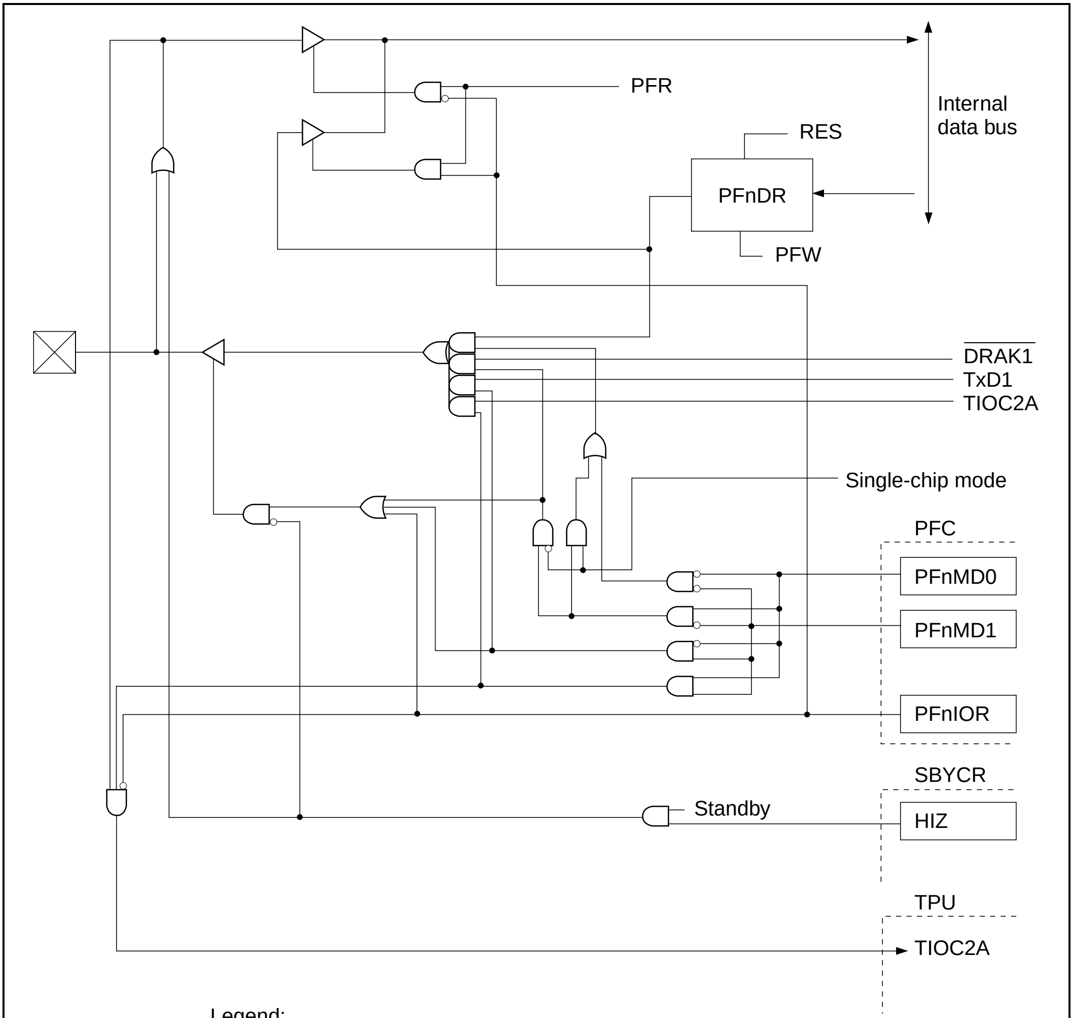
Figure C.40 PF3/$\overline{\text{DREQ0}}$/TIOC0A Block Diagram



Legend:
PFR: Port F read signal
PFW: Port F write signal
RES: Reset signal

Note: n = 5

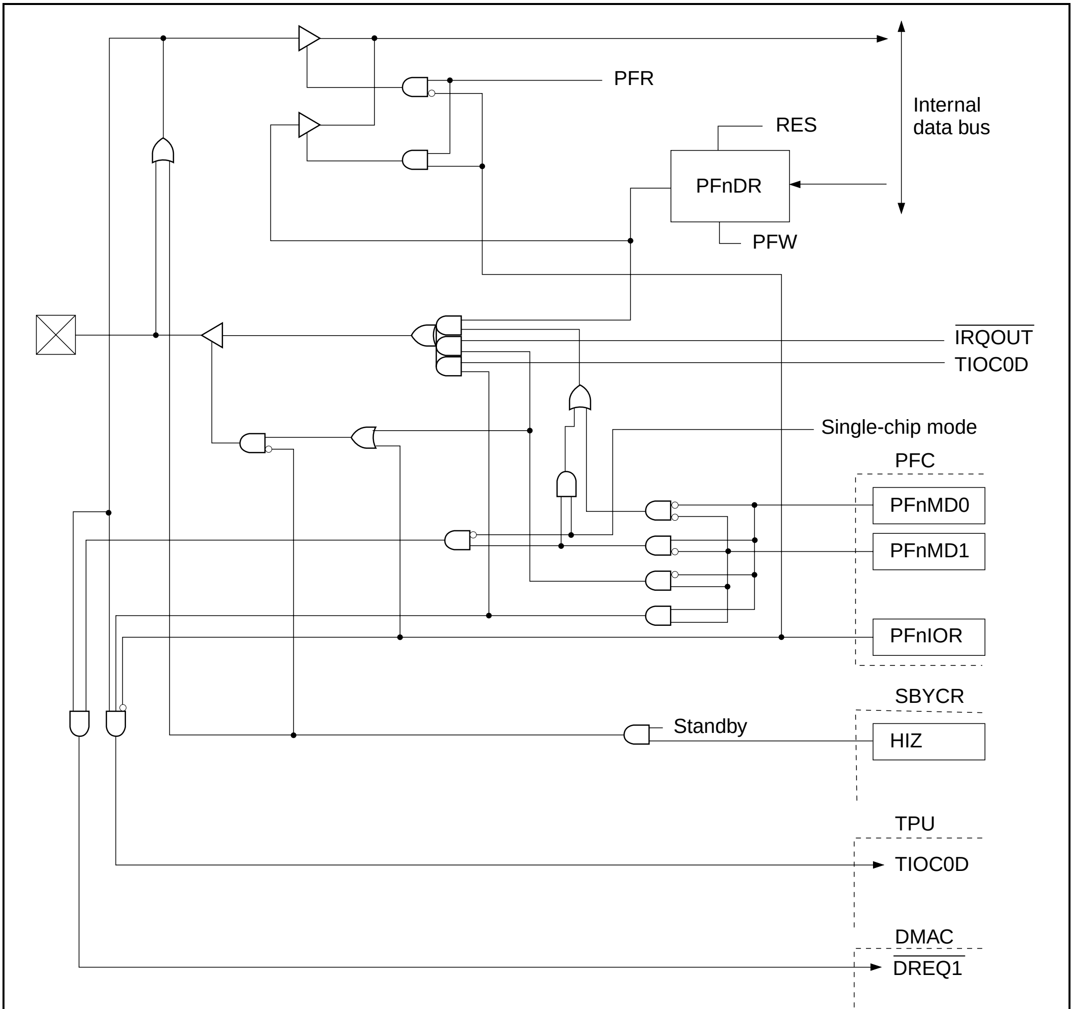
Figure C.41 PF5/$\overline{\text{DACK1}}$/RxD1/TIOC2B Block Diagram



Legend:
PFR: Port F read signal
PFW: Port F write signal
RES: Reset signal

Note: n = 6

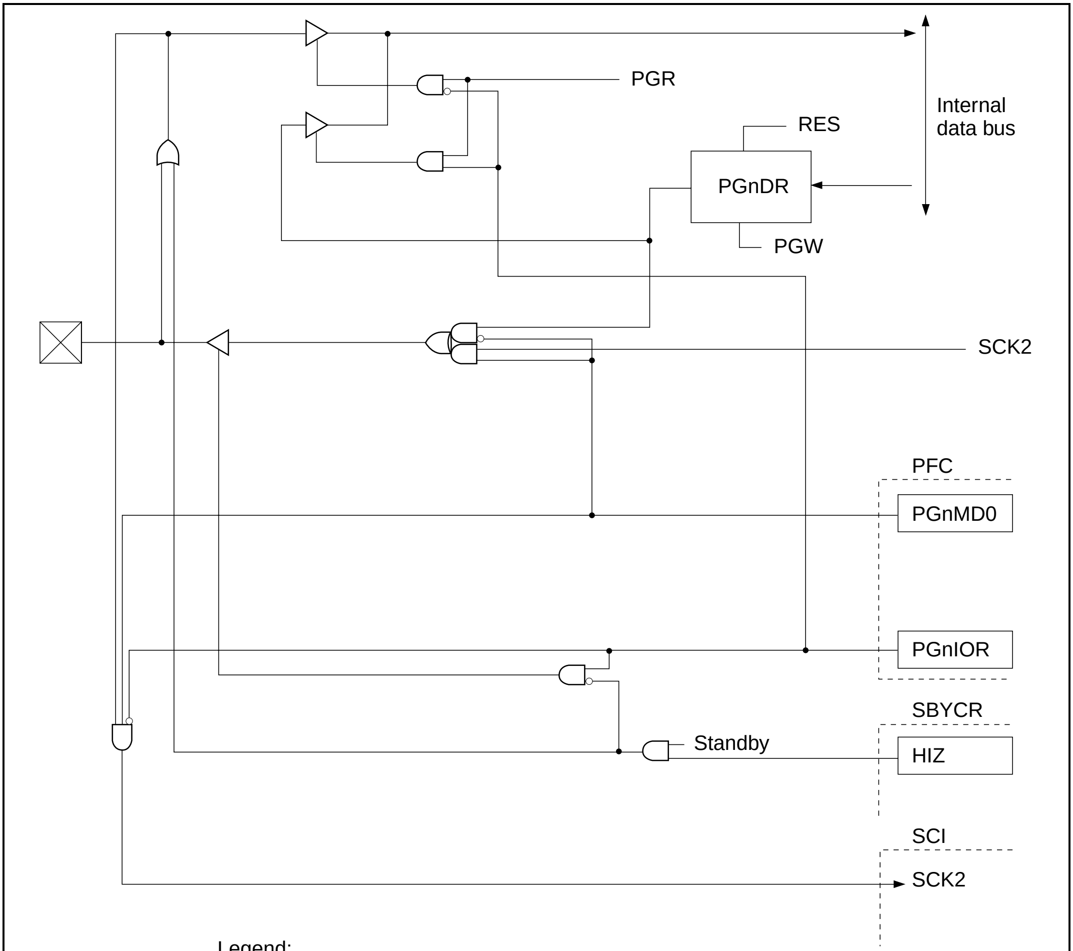
Figure C.42 PF6/$\overline{\text{DRAK1}}$/TxD1/TIOC2A Block Diagram



Legend:
PFR: Port F read signal
PFW: Port F write signal
RES: Reset signal

Note: n = 7

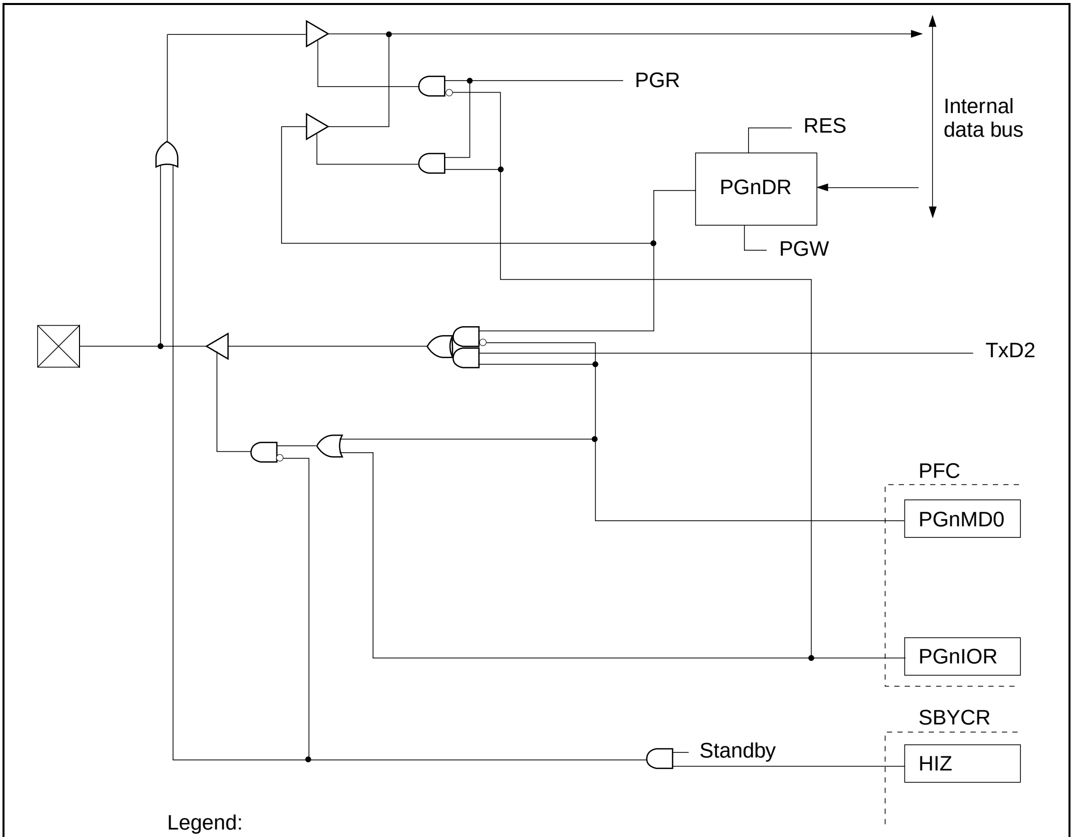
Figure C.43 PF7/$\overline{\text{DREQ1}}$/$\overline{\text{IRQOUT}}$/TIOC0D Block Diagram



Legend:
PGR: Port G read signal
PGW: Port G write signal
RES: Reset signal

Note: n = 29

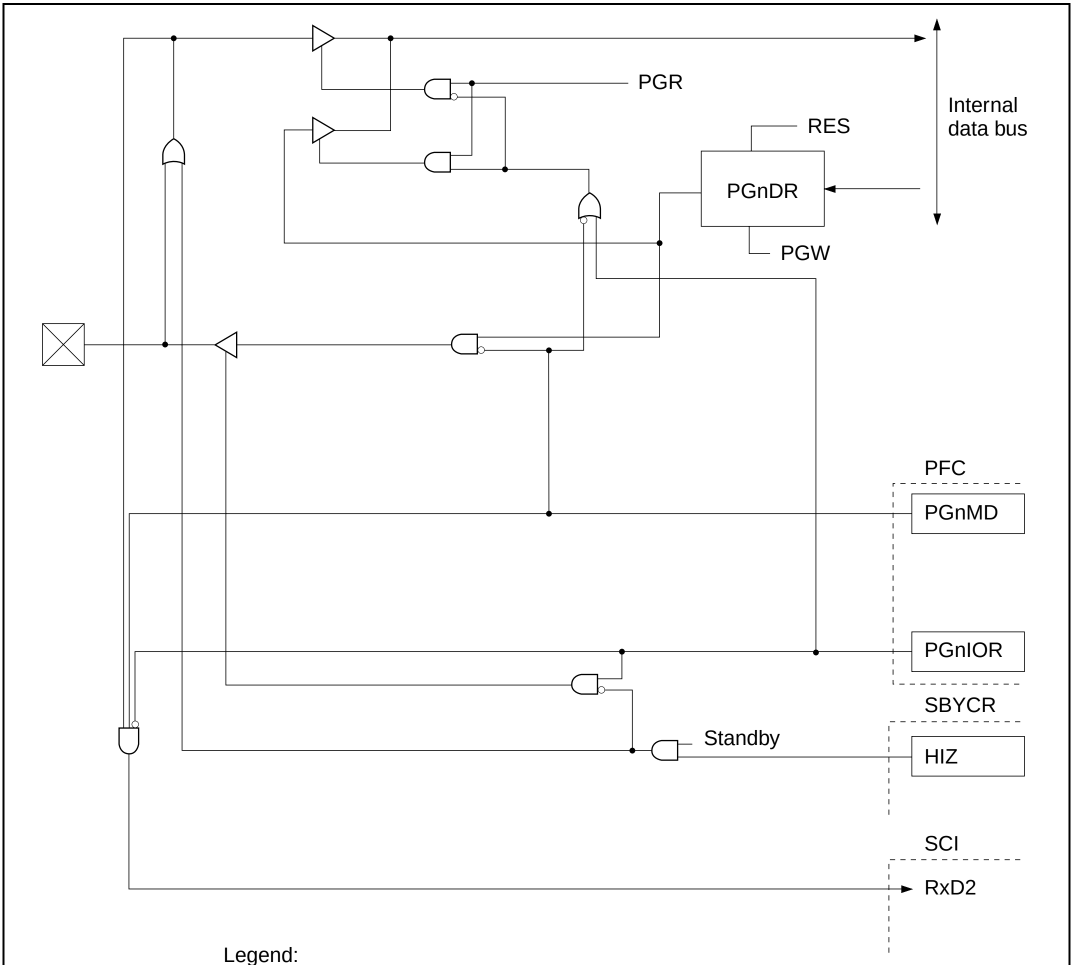
Figure C.44   PG29/SCK2 Block Diagram



Legend:
PGR: Port G read signal
PGW: Port G write signal
RES: Reset signal

Note: n = 30

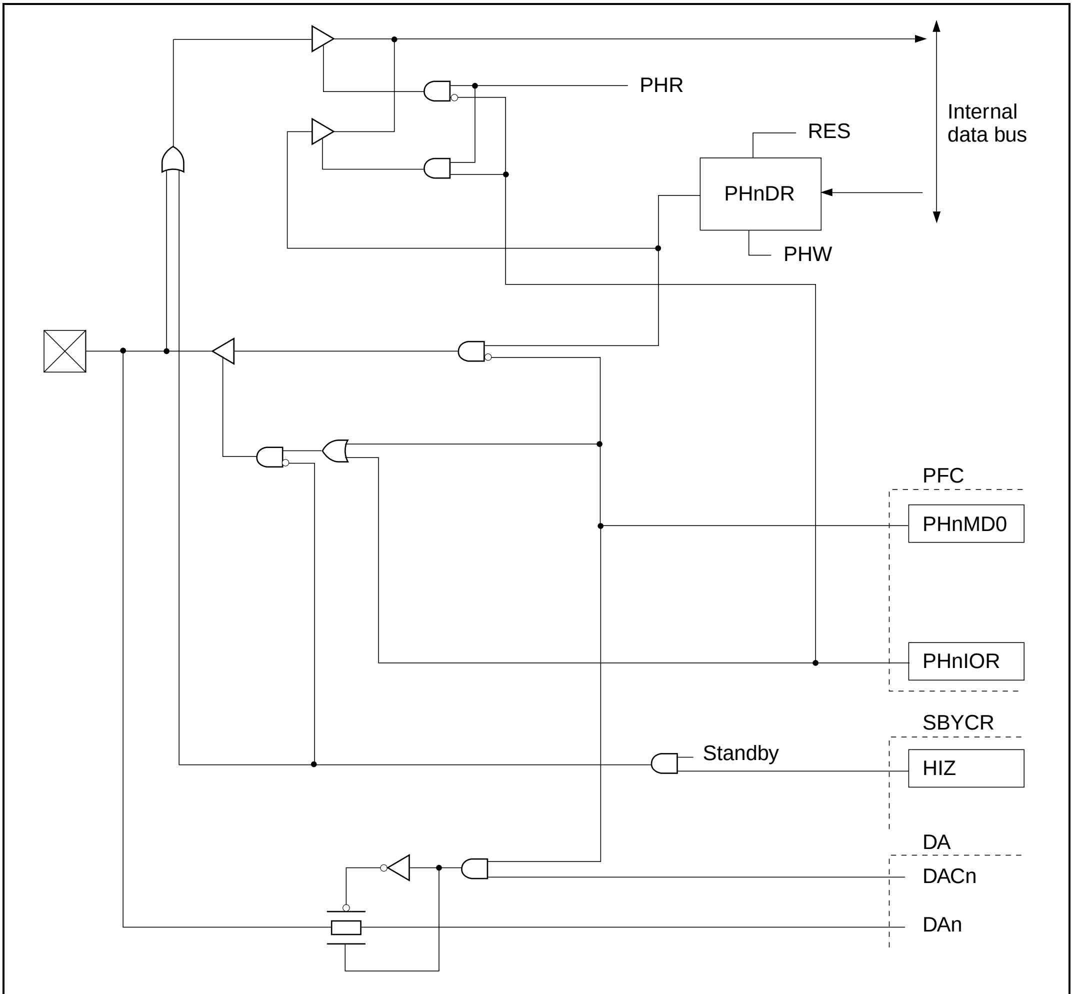
Figure C.45 PG30/TxD2 Block Diagram



Legend:
PGR: Port G read signal
PGW: Port G write signal
RES: Reset signal

Note: n = 31

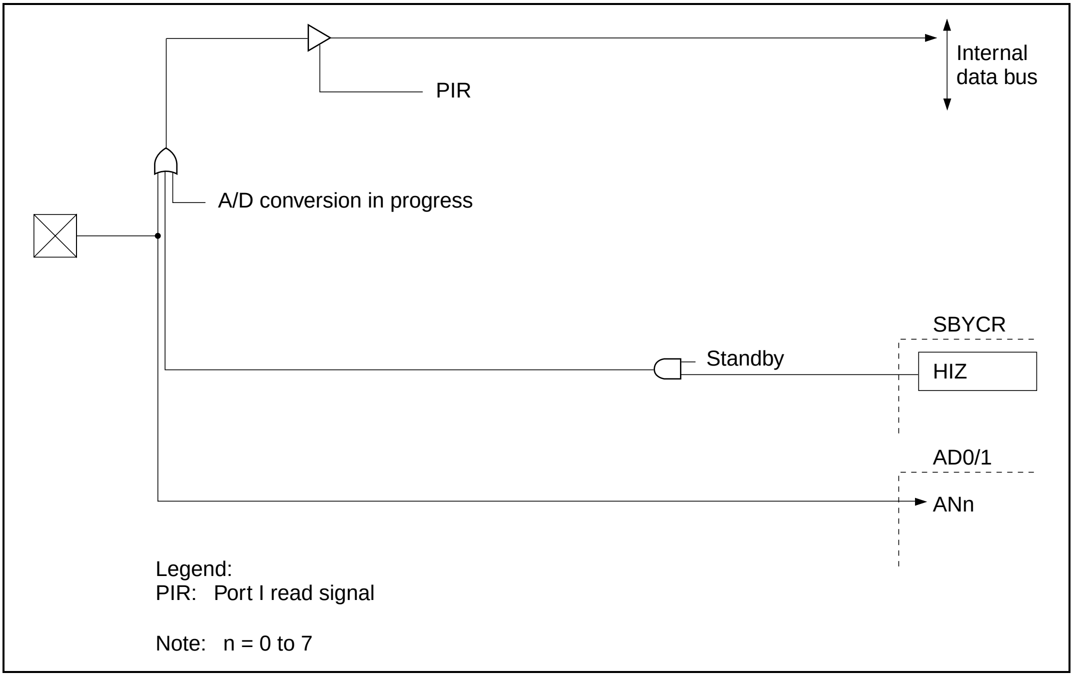
Figure C.46 PG31/RxD2 Block Diagram



Legend:
PHR: Port H read signal
PHW: Port H write signal
RES: Reset signal

Note: n = 0, 1

Figure C.47 PHn/DAn Block Diagram



Legend:
PIR: Port I read signal

Note: n = 0 to 7

Figure C.48 PIn/ANn Block Diagram

Appendix D Restrictions and Caution on HD64F7065S (and HD64F7065A Lots Prior to “1D5”)

In addition to the Usage Notes given at the end of each section, the following restrictions and caution also apply to the HD64F7065S. Items D.5 and D.6 apply to all HD64F7065S lots and also to HD64F7065A lots prior to “TD5”. Items D.1 to D.4 apply only to all HD64F7065S lots.

D.1 BSC Restrictions

1. EDO DRAM burst operation is not supported when Mϕ (the clock obtained after frequency division of the master clock (CKM)) is faster than CKE (the external bus clock).
2. When wait cycle control is performed by means of the external WAIT pin, the following restrictions apply only when an 8-bit-bus width is set.
   - The access size for external 8-bit space must be byte or word.
   - Use two word accesses to access 32-bit data in external 8-bit space.
3. If the master clock (CKM) frequency exceeds 30 MHz, the inter-cycle idle number specification by bits IW2 to IW0 in the area control register (ACR) is invalid. The inter-cycle idle number in this frequency band is an indeterminate number between 0 and 7 cycles. The following automatically inserted idle cycles are valid.
   - Two idle cycles when switching from a read cycle to a write cycle in the same space
   - One idle cycle when switching from a read cycle to a read cycle in a different space
   - Two idle cycles when switching from a read cycle or write cycle to a write cycle in a different space

D.2 Restrictions in Case of Contention between DSP Instruction and DMAC Transfer

In on-chip ROM high-speed mode, a bug may occur under the following conditions during program execution in on-chip ROM.

1. Contention between MOVX instruction execution and on-chip XRAM access by the DMAC
2. Contention between MOVY instruction execution and on-chip YRAM access by the DMAC

This bug may result in an incorrect result of MOVX or MOVY instruction execution.

On-chip XRAM access/on-chip YRAM access here refers to cases where either the transfer source or transfer destination is on-chip XRAM/on-chip YRAM. The DMAC transfer mode (cycle-steal/burst) is irrelevant.

D.3 Pin State Related Restrictions

The states of pins PF7 and PF6 in software standby mode are as follows.

1. PF7 has PF7/$\overline{\text{IRQOUT}}$/TIOC0D functions. It goes to the high-impedance state in software standby mode, regardless of which function is selected.
2. PF6 has PF6/TxD1/TIOC2A functions. This pin, too, basically goes to the high-impedance state in software standby mode, as with the PF7 pin. However, when the PF6 output function or TxD1 output function is selected and the pin state is high-level output, the high-level output state is retained when a transition is made to software standby mode.

D.4 Caution Concerning Electrical Characteristics

In external space access, noise with a peak value of 1.5 V max. may occur while the write signal ($\overline{\text{WR}}$) is low.

D.5 DMAC Restrictions

The DMA transfer overrun may occur when using the four on-chip DMAC channels (channel 0 to channel 3).

In external request mode and on-chip peripheral module request mode, once DMA transfer is executed, DMA transfer may continue to be executed until the contents of DMAC transfer count register (DMATCRn) and next transfer count register (NDMATCRn) become 0. This does not apply to auto request mode.

Conditions under which the DMAC stops for certain are shown below:

- When chain transfer is not performed (chain transfer enable bit (CHNE) = 0): until DMATCRn = 0
- When chain transfer is performed (chain transfer enable bit (CHNE) = 1): until DMATCRn = 0 and NDMATCRn = 0

There are some restrictions about TEND output. Refer to 9.6 DMAC Restrictions.

D.6 Restrictions about Changing the Saturation Operation Mode during the Execution of Multiply/Multiply and Accumulate, or DSP Instructions

When instruction execution is stalled by the occurrence of multiplier contention between Multiply/Multiply and Accumulate instructions, or register contention by the continuous execution of DSP instructions, pay attention to the following:

If the S bit (saturation operation bit) of SR (status register) is changed directly after the execution of Multiply/Multiply and Accumulate or DSP instructions, the sequences of these instructions can be reversed. AS a result, an instruction that should be executed before the change of the S bit can be executed after the change of the S bit, and the result of the instruction may be wrong.

Instructions that can be affected by the change of the S bit

- Multiply and Accumulate instructions: MAC.W, MAC.L
- DSP instructions: ALU Arithmetic operation instructions, Fixed-point multiply instruction, Arithmetic shift operation instructions

Examples for the wrong operation cases are shown below:

1. In case of Multiply/Multiply and Accumulate instructions

(a)	DMULU.L	R4,R10	MUL.L, DMULS.L, DMULU.L, and MAC.L can be applied.
(b)	MAC.L	@R5+,@R5+	MAC.W and MAC.L can be applied. Instruction execution is stalled by the occurrence of multiplier contention.
(c)	LDC	R0,SR	Saturation operation mode is changed.

Multiplier contention is occurred between (a) DMULU.L instruction and (b) MAC.L instruction, and the execution of (b) MAC.L instruction is stalled. The changing of the S bit by (c) instruction is executed in the CPU before the execution of (b) MAC.L instruction due to the pipeline control. As a result, the sequence of (b) and (c) instructions is reversed, and the result of the MAC.L instruction becomes to be wrong.

2. In case of DSP instructions

```
(a)  PSHA   #1,A1
(b)  PINC   X0,A0    MOVX.W   A1,@R5    Instruction execution is stalled by the
                                        occurrence of register contention.
(c)  LDC    R0,SR                       Saturation operation mode is changed.
```

As the operation result of DSP instruction is stored directly after the execution of the DSP instruction, register contention is occurred between (a) PSHA and (b) MOVX instructions, and the execution of (b) PINC instruction is stalled. The changing of the S bit by (c) instruction is executed in the CPU before the execution of (b) PINC instruction due to the pipeline control. As a result, the sequence of (b) and (c) instructions is reversed, and the operation result of the PINC instruction becomes to be wrong.

In order to avoid the restriction; select one of the followings:

(1) Do not access SR register directly after the Multiply and Accumulate or DSP instructions.

(2) Insert NOP instruction directly before LDC Rn, SR instruction.

(3) Arrange the instruction sequences so that neither multiplier contention nor DSP register contention can be occurred (no instruction is stalled).

Appendix E Product Lineup

Table E.1 SH7065 Product Lineup

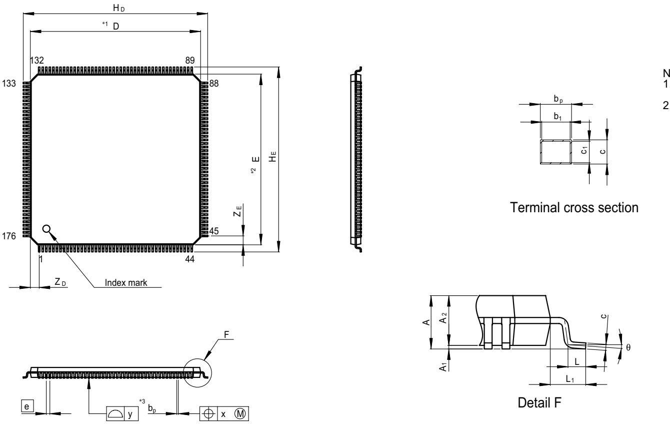
	Product Type	Model	Model Marking	Package
SH7065	Mask ROM version	HD6437065A	HD6437065A(***)F	176-pin QFP (FP-176C)
	F-ZTAT version	HD64F7065S	HD64F7065SF60	176-pin QFP (FP-176)
		HD64F7065A	HD64F7065AF60	176-pin QFP (FP-176)

Note: (***) is the ROM code.

Appendix F Package Dimensions

The package dimension that is shown in the Renesas Semiconductor Package Data Book has priority.

JEITA Package Code	RENESAS Code	Previous Code	MASS[Typ.]
P-LQFP176-24x24-0.50	PLQP0176KC-A	FP-176/FP-176V	1.9g

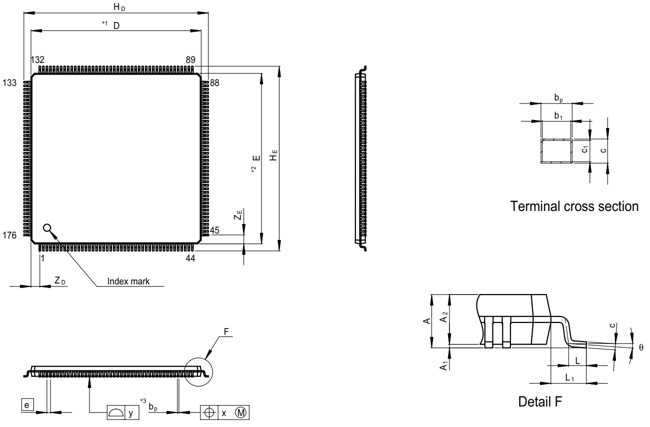


NOTE)
1. DIMENSIONS"*1"AND"*2"
DO NOT INCLUDE MOLD FLASH
2. DIMENSION"*3"DOES NOT
INCLUDE TRIM OFFSET.

Reference Symbol	Dimension in Millimeters		
	Min	Nom	Max
D	—	24	—
E	—	24	—
A_2	—	1.40	—
H_D	25.8	26.0	26.2
H_E	25.8	26.0	26.2
A	—	—	1.70
A_1	0.05	0.10	0.15
b_p	0.17	0.22	0.27
b_1	—	0.20	—
c	0.12	0.17	0.22
c_1	—	0.15	—
θ	0°	—	8°
$\boxed{e}$	—	0.5	—
x	—	—	0.10
y	—	—	0.10
Z_D	—	1.25	—
Z_E	—	1.25	—
L	0.4	0.5	0.6
L_1	—	1.0	—

Figure F.1 Package Dimensions (FP-176)

JEITA Package Code	RENESAS Code	Previous Code	MASS[Typ.]
P-LQFP176-24x24-0.50	PLQP0176KD-A	FP-176C/FP-176CV	1.9g



NOTE)
1. DIMENSIONS"*1"AND"*2"
   DO NOT INCLUDE MOLD FLASH
2. DIMENSION"*3"DOES NOT
   INCLUDE TRIM OFFSET.

Reference Symbol	Dimension in Millimeters		
	Min	Nom	Max
D	—	24	—
E	—	24	—
A_2	—	1.40	—
H_D	25.8	26.0	26.2
H_E	25.8	26.0	26.2
A	—	—	1.70
A_1	0.05	0.10	0.15
b_p	0.17	0.22	0.27
b_1	—	0.20	—
c	0.12	0.17	0.22
c_1	—	0.15	—
θ	0°	—	8°
$\boxed{e}$	—	0.5	—
x	—	—	0.08
y	—	—	0.08
Z_D	—	1.25	—
Z_E	—	1.25	—
L	0.4	0.5	0.6
L_1	—	1.0	—

Figure F.2 Package Dimensions (FP-176C)

Renesas 32-Bit RISC Microcomputer
Hardware Manual
SH7065

Publication Date: 1st Edition, August 1999
Rev.5.00, September 11, 2006
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

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) 6887-7898

Renesas Technology Hong Kong Ltd.
7th Floor, North Tower, World Finance Centre, Harbour City, 1 Canton Road, Tsimshatsui, Kowloon, Hong Kong
Tel: <852> 2265-6688, Fax: <852> 2730-6071

Renesas Technology Taiwan Co., Ltd.
10th Floor, No.99, Fushing North Road, Taipei, Taiwan
Tel: <886> (2) 2715-2888, Fax: <886> (2) 2713-2999

Renesas Technology Singapore Pte. Ltd.
1 Harbour Front Avenue, #06-10, Keppel Bay Tower, Singapore 098632
Tel: <65> 6213-0200, Fax: <65> 6278-8001

Renesas Technology Korea Co., Ltd.
Kukje Center Bldg. 18th Fl., 191, 2-ka, Hangang-ro, Yongsan-ku, Seoul 140-702, Korea
Tel: <82> (2) 796-3115, Fax: <82> (2) 796-2145

Renesas Technology Malaysia Sdn. Bhd
Unit 906, Block B, Menara Amcorp, Amcorp Trade Centre, No.18, Jalan Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: <603> 7955-9390, Fax: <603> 7955-9510

Colophon 6.0

SH7065
Hardware Manual

RENESAS

Renesas Technology Corp.
2-6-2, Ote-machi, Chiyoda-ku, Tokyo, 100-0004, Japan